

DIF3D-VARIANT 11.0: A Decade of Updates

Nuclear Engineering Division

About Argonne National Laboratory

Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

DOCUMENT AVAILABILITY

Online Access: U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via DOE's SciTech Connect (<http://www.osti.gov/scitech/>)

Reports not in digital format may be purchased by the public from the National Technical Information Service (NTIS):

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312
www.ntis.gov
Phone: (800) 553-NTIS (6847) or (703) 605-6000
Fax: (703) 605-6900
Email: **orders@ntis.gov**

Reports not in digital format are available to DOE and DOE contractors from the Office of Scientific and Technical Information (OSTI):

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
www.osti.gov
Phone: (865) 576-8401
Fax: (865) 576-5728
Email: **reports@osti.gov**

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 UChicago Argonne, LLC, nor any of their employees or officers, 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 document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

DIF3D-VARIANT 11.0: A Decade of Updates

prepared by
M. A. Smith, E. E. Lewis, E. R. Shemon
Nuclear Engineering Division, Argonne National Laboratory

January 16, 2014

SUMMARY

The DIF3D code has been a workhorse of fast reactor analysis work at Argonne National Laboratory for over 30 years. In 1995, a transport option called VARIANT was added to DIF3D to improve the flux solutions for fast reactor problems. VARIANT performs nodal neutron transport calculations using P_N or SP_N theory in Cartesian and hexagonal two- and three-dimensional geometries. Because of the computing capabilities at that time, VARIANT was restricted to 33 group P_3 flux approximations with P_1 scattering. Clearly computer memory capabilities have increased since then and thus large space-angle-energy approximations are possible. This manuscript serves as an update to the theory section of the original manual and details more than a decade worth of changes made to DIF3D to make DIF3D 11.0.

The primary focus of this work was to extend the space-angle approximations available in DIF3D-VARIANT such that the error due to transport approximations could be resolved. This work was started and completed in 2002 and marked the official version of DIF3D 10.0. Unfortunately, those higher order approximations could not be used due to the memory constraints of the BPOINTER software (limited to 2 GB). In DIF3D 11.0, BPOINTER was circumvented for the largest arrays by introducing a Fortran 90 module called LMA (Large Memory Array). This seamlessly replaces all of the functionality of the previous software concept except the use of 64 bit addressing allows any given array to be considerably larger than 2 GB. It is now common for DIF3D-VARIANT jobs to consume 50 GB of memory on modern workstations when using high order space-angle approximations and a large number of groups.

Over the course of making these changes, numerous minor bugs were discovered that had to be fixed. Many of these bugs were found during the process of creating the perturbation and sensitivity code PERSENT and thus VARIANT was updated accordingly. All of the changes are discussed in this manuscript along with the impacts upon performance and accuracy. A detailed set of verification problems are performed demonstrating the new capabilities and we refer users to the PERSENT manual for further details on the steady state fixed source capabilities.

Unfortunately, the coarse mesh rebalance & fission source extrapolation, the acceleration scheme for the outers, were removed in DIF3D 11.0 as they were not very effective in hexagonal geometries and were not working for the adjoint flux problems. Tchebychev acceleration was inserted to replace it which we note is typically less effective. This change can increase the number of outer iterations relative to DIF3D 10.0, but it was not uncommon for coarse mesh rebalance to lead to false convergence or no convergence at all which itself might have been associated with the various bugs that were fixed in DIF3D 11.0.

In conclusion, the new version of VARIANT incorporated in DIF3D-11 allows users to fully validate their calculations by performing detailed space-angle refinement and resolves numerous minor bugs. These improvements facilitate existing users of DIF3D to study ever more difficult problems and are a testament to the importance of the VARIANT option for existing research being carried out at Argonne National Laboratory. This manuscript serves as an addendum manual to the original VARIANT manual covering the numerous changes made over the past decade and the impact that they have had on the performance and accuracy.

TABLE OF CONTENTS

Summary	i
Table of Contents	ii
List of Figures	iv
List of Tables	v
1 Introduction	6
2 Variational Nodal Transport Theory	8
2.1 The within-group transport equation	8
2.2 The even-parity form of the transport equation	9
2.3 The variational nodal functional	10
2.4 The variational treatment	12
2.5 Obtaining the response matrix form	14
2.6 Anisotropic Scattering Considerations	18
3 The Angular Approximation Implemented in VARIANT	23
3.1 Introduction	23
3.2 Angular approximation	23
3.3 Boundary conditions for the response matrix method	25
3.4 Corrections to the angular approximation	33
3.5 The angular integrals	35
4 The Spatial Approximations Implemented in VARIANT	42
4.1 Introduction	42
4.2 The RITZ approximation	42
5 Non-Zero Storage Implementation	47
6 Large Memory Array Implementation	49
6.1 Introduction	49
6.2 Implementation of LMA	49
6.3 Impact of LMA	50
7 Additional DIF3D-11.0 Algorithm Changes	51
7.1 Fission source error measures and acceleration	51
7.2 Modification to the DIF3D outer iteration history	55
7.3 Inclusion of an external P_N based fixed source file	61
7.4 Anisotropic scattering bug in SPN	61
7.5 Errors with “Omega” Acceleration	61
7.6 Errors with Hexagonal 60 and 120 Periodic Geometries	65
8 Results	73
8.1 Introduction	73
8.2 TAKEDA-2 Benchmark	73

8.3 Takeda-4 Benchmark	74
8.4 EBR-II Benchmark	75
8.5 ABR Benchmark.....	76
8.6 MONJU Benchmark	78
8.7 Higher Order Scattering Kernel	80
9 Conclusions	82
9.1 Acknowledgements.....	83
10 References	84
11 Appendix A. Modifications to DIF3D Input Descriptions.....	86

LIST OF FIGURES

Figure 2.1 The Splitting of the Energy Range into Energy Groups.....	8
Figure 2.2 An Example Three Dimensional Cartesian Nodal Geometry	11
Figure 3.1. A Three-Dimensional Cartesian Node	25
Figure 3.2. Direction Cosines and Numbered Octants	26
Figure 3.3. Simplified 1-D Diagram of the Reflected Boundary Condition.....	26
Figure 3.4. Even Parity Interface Trial Functions $p(\hat{V})$	29
Figure 3.5. Odd Parity Interface Trial Functions $k(\hat{V})$	30
Figure 3.6. Cartesian and Hexagonal Angular Surface Function Orientations.....	39
Figure 4.1. Gramm-Schmidt Procedure for Obtaining Orthonormal Functions.....	43
Figure 4.2. Fortran Approach to Performing Gramm-Schmidt Procedure.	44
Figure 7.1. Scatter Load Example for Hex Geometry and Fast Flux Profile.....	53
Figure 7.2. Example DIF3D 8.0 through 10.0 Outer Iteration Information without Acceleration	57
Figure 7.3. Example DIF3D 10.0 Outer Iteration Information with Acceleration	58
Figure 7.4. Example DIF3D 11.0 Outer Iteration Information.....	59
Figure 7.5. Example DIF3D 11.0 Optical Thickness Convergence Problem.....	60
Figure 7.6. Hexagonal 60 Degree Periodic Geometry and Current Rotations.....	66
Figure 7.7. Hexagonal 120 Degree Periodic Geometry and Current Rotations.....	66
Figure 7.8. Four Current Rotations Necessary in Periodic Hexagonal Geometries	66
Figure 7.9. Example Current Alignment in Hexagonal Geometry	67
Figure 7.10. Full Core 3D Results Proving that the R_B and R_C Mappings are Necessary.....	70
Figure 7.11. 30 Degree Symmetric Global Surface IDs for j^\pm Partial Currents.	71
Figure 7.12. 60 Degree Periodic Global Surface IDs for j^\pm Partial Currents.....	71
Figure 7.13. 120 Degree Periodic Global Surface IDs for j^\pm Partial Currents.....	72
Figure 11.1. Updated A.DIF3D File Description	86
Figure 11.2. Description of New VARSRC Fixed Source File Added to DIF3D-11.0.....	88

LIST OF TABLES

Table 3.1. Angular Rotations for Spherical Harmonic Trial Functions.....	39
Table 3.2. Changes to the DIF3D Input Description on Card 12.	41
Table 4.1. Changes to the DIF3D BCD Input Description on Card 12.	46
Table 7.1. Bare, Homogeneous Hexagonal Reactor Outer Iteration Count in Diffusion Theory	52
Table 7.2. Twelve Ring Scatter Core Outer Iteration Count in Diffusion Theory	53
Table 7.3. Sixteen Ring Scatter Core Outer Iteration Count in Diffusion Theory	54
Table 7.4. Forward and Adjoint Eigenvalues and PCM Errors for the Test Problems.....	64
Table 7.5. Outer Iteration Count for the Test Problems	65
Table 8.1. Eigenvalue and Performance for the Takeda 2 Benchmark	73
Table 8.2. Eigenvalue and Performance for the Takeda-4 Benchmark	75
Table 8.3. Eigenvalue and Performance for the EBR-II Benchmark	75
Table 8.4. Eigenvalue and Performance for the ABR Benchmark	77
Table 8.5. Eigenvalue and Performance for the 9g MONJU Simplified Benchmark.....	78
Table 8.6. Eigenvalue and Performance for the 33g MONJU Simplified Benchmark.....	79
Table 8.7. Eigenvalue and Performance for the 70g MONJU Simplified Benchmark.....	79
Table 8.8. Eigenvalue and Performance for the 230g MONJU Simplified Benchmark.....	80
Table 8.9. Eigenvalue and Performance for the MONJU Benchmark	81
Table 11.1. Changes to the DIF3D Input Description on Card 12.	87
Table 11.2. Changes to the DIF3D BCD Input Description on Card 12.	87

1 Introduction

The DIF3D code has been a workhorse of fast reactor analysis work at Argonne National Laboratory for over 30 years. In 1995, a transport option called VARIANT [1-11] was added to DIF3D to improve the flux solutions for fast reactor problems. VARIANT performs nodal neutron transport calculations using P_N or SP_N theory in Cartesian and hexagonal two- and three-dimensional geometries. At its inception, VARIANT was limited because of the computational effort required to solve the transport equation and the limited computer memory. The typical job was restricted to 33 group P_3 flux approximations with P_1 scattering. Clearly computer memory and performance have increased considerably since then and larger space-angle-energy approximations are possible.

In 2002, there was a considerable desire to assess the error due to the space-angle approximations made in the VARIANT code. At issue was that VARIANT only allowed a maximum P_5 flux approximation with a P_5 scattering kernel to be combined with a 6th order flux (4th order in the axial direction) and quadratic leakage approximations. Thus to enable higher order space-angle approximations, we had to modify the parts of the code that restricted it to the original capabilities. In this manuscript we show how an arbitrary order space-angle approximation was implemented consistent with the underlying VARIANT methodology. The anisotropic scattering treatment was also extended to match the flux approximation order although we have not extensively checked it. The coupling coefficient data statements originally implemented within the VARIANT code were completely removed and Fortran 90 data structures were added to avert having to rewrite substantial portions of the original coding. This was accomplished by treating Fortran 90 modules as common blocks. The conclusion of this work marked the creation of DIF3D 10.0.

While this version was relatively easy to create, higher order approximations could not be used due to the memory constraints of the BPOINTER software (limited to 2 GB). Given workstation computer memory today is at 132 GB and likely will increase more in the future, having an artificial 2 GB memory restriction is quite difficult to justify. BPOINTER was circumvented for the largest arrays by introducing a Fortran 90 module called LMA (Large Memory Array). This module seamlessly replaces all of the functionality of the BPOINTER software concept except it uses 64 bit addressing allowing any given array to be considerably larger than 2 GB. It is now common for DIF3D-VARIANT jobs to consume 50 GB of memory on modern workstations when using high order space-angle approximations and a large number of groups. This marked the creation of DIF3D 11.0. This manuscript revisits the methodology behind removing the data statements and the way in which the new coding is implemented into VARIANT covering all of the changes to make DIF3D 10.0 and 11.0.

The primary motivating force behind these changes was the desire to run large energy group problems with P_5 anisotropic scattering. At present, we have modified the code well enough that P_9 flux approximations with P_9 anisotropic scattering are tractable at 230 groups although not advisable because the total run time for a full core reactor problem can be considerable. The results of several benchmark problems are detailed along with computational timing comparisons between DIF3D 9.0, 10.0 and 11.0. Section 2 gives a brief review of the variational nodal theory to correct some of the notation mistakes found in the VARIANT

manual. In Sections 3 and 4, the new methodology of obtaining the angular and spatial approximations is detailed along with corrections to the nomenclature given in the VARIANT manual. Most of sections 2 through 4 were taken directly from reference [7]. In Sections 5 and 6, we briefly cover some of the changes that had to be implemented to the data structuring of VARIANT. Section 7 discusses the changes in the algorithm which improved the functionality in the code. Finally, in Section 8 we present numerical results contrasting the new version of VARIANT with the existing production version.

2 Variational Nodal Transport Theory

2.1 The within-group transport equation

The formulation starts with the time independent form of the Boltzmann transport equation taken from Lewis [15]

$$\hat{W} \cdot \hat{\nabla} y(\vec{r}, \hat{W}, E) + S_t(\vec{r}, E) y(\vec{r}, \hat{W}, E) = \int \int S_s(\vec{r}, \hat{W}' \rightarrow \hat{W}, E' \rightarrow E) y(\vec{r}, \hat{W}', E') dW' dE' + S(\vec{r}, \hat{W}, E) \quad (1)$$

$y(\vec{r}, \hat{W}, E)$ represents the neutron angular flux and is a function of three space variables (x, y, z in \vec{r}), two angular variables (q and f in \hat{W}), and one energy variable (E). The total cross section, $S_t(\vec{r}, E)$, represents the sum of all possible neutron reaction probabilities with energy E at the point \vec{r} . Similarly, the scattering kernel $S_s(\vec{r}, \hat{W}' \rightarrow \hat{W}, E' \rightarrow E) dW' dE'$ represents the probability that a particle at \vec{r} with energy E' traveling in the direction \hat{W}' is scattered into energy dE about E with direction dW about \hat{W} . Finally, $S(\vec{r}, \hat{W}, E)$ is a generic neutron source that includes fission sources as well as fixed and external sources. Overall, equation 1 conserves neutrons over the entire problem domain. Using equation 1, a deterministic solution of the neutron angular flux is sought using a variational principle.

For now, only isotropic scattering is implemented to simplify the system of equations (anisotropic scattering is discussed in Section 2.6). The substitution of an isotropic scattering kernel in place of the anisotropic scattering kernel simplifies equation 1 to

$$\hat{W} \cdot \hat{\nabla} y(\vec{r}, \hat{W}, E) + S_t(\vec{r}, E) y(\vec{r}, \hat{W}, E) = \int S_s(\vec{r}, E' \rightarrow E) f(\vec{r}, E') dE' + S(\vec{r}, \hat{W}, E), \quad (2)$$

where the scalar flux relationship $f(\vec{r}, E) = \int y(\vec{r}, \hat{W}, E) dW$ has been employed.

The first step in a deterministic formulation is to apply a multigroup approach [1] to the energy dependence of the neutron flux. This approach divides the energy range of interest into G intervals with an upper energy cutoff, E_G , and lower energy cutoff, E_0 , as seen in Figure 2.1.

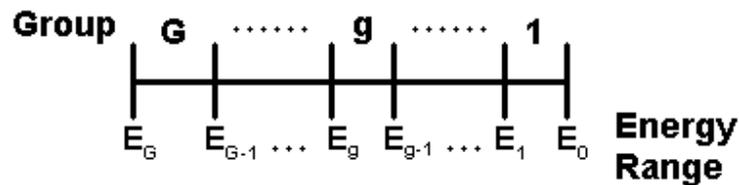


Figure 2.1 The Splitting of the Energy Range into Energy Groups

The desire is to develop G equations based upon the group angular fluxes $y_g(\vec{r}, \hat{W})$ and the group sources $S_g(\vec{r}, \hat{W})$ defined by equations 3 and 4.

$$y_g(\vec{r}, \hat{W}) = \int_{E_g}^{E_{g-1}} y(\vec{r}, \hat{W}, E) dE = \int_{E_g} y(\vec{r}, \hat{W}, E) dE \quad g = 1, \dots, G \quad (3)$$

$$S_g(\vec{r}, \hat{W}) = \int_{E_g} S(\vec{r}, \hat{W}, E) dE \quad g = 1, \dots, G \quad (4)$$

To obtain these G equations, equation 2 is integrated over each energy group g and the following relations are defined for the group cross sections.

$$S_{t,g}(\vec{r}) f_g(\vec{r}) = \int_{E_g} S_{t,g}(\vec{r}, E) f(\vec{r}, E) dE \quad (5)$$

$$S_{s,g \otimes g}(\vec{r}) f_g(\vec{r}) = \int_{E_g} \int_{E'_g} S_s(\vec{r}, E' \otimes E) f(\vec{r}, E') dE' dE \quad (6)$$

Using these relationships the multigroup Boltzmann transport equation is given as

$$\hat{W} \cdot \nabla y_g(\vec{r}, \hat{W}) + S_{t,g}(\vec{r}) y_g(\vec{r}, \hat{W}) = \sum_{g'=1}^G S_{s,g \otimes g'}(\vec{r}) f_{g'}(\vec{r}) + S_g(\vec{r}, \hat{W}). \quad g = 1, \dots, G \quad (7)$$

To simplify this expression all but the within group scattering ($g = g'$) can be lumped into the group source thus arriving at the following within group form of the transport equation

$$\hat{W} \cdot \nabla y_g(\vec{r}, \hat{W}) + S_{t,g}(\vec{r}) y_g(\vec{r}, \hat{W}) = S_{s,g \otimes g}(\vec{r}) f_g(\vec{r}) + S_g(\vec{r}, \hat{W}). \quad g = 1, \dots, G \quad (8)$$

2.2 The even-parity form of the transport equation

The next step is to transform this equation into an even parity form, which lends itself better to the variational treatment. First the angular flux is split up into even and odd parity components given by

$$y_g(\vec{r}, \hat{W}) = y_g^+(\vec{r}, \hat{W}) + y_g^-(\vec{r}, \hat{W}), \quad (9)$$

where + denotes even parity and - denotes odd parity. The even and odd parity components of the flux have the following properties, where the function $f_g(\vec{r})$ represents the group scalar flux.

Even	Odd	
$y_g^+(\vec{r}, \hat{W}) = y_g^+(\vec{r}, -\hat{W})$	$y_g^-(\vec{r}, \hat{W}) = -y_g^-(\vec{r}, -\hat{W})$	(10)
$\int_{\hat{\Omega}} y_g^+(\vec{r}, \hat{W}) dW = f_g(\vec{r})$	$\int_{\hat{\Omega}} y_g^-(\vec{r}, \hat{W}) dW = 0$	

Inserting equation 9 into equation 8 gives an expression of the form

$$\hat{W} \cdot \nabla y_g^+(\vec{r}, \hat{W}) + y_g^-(\vec{r}, \hat{W}) \int_{\hat{\Omega}} S_{t,g}(\vec{r}) dW + S_{t,g}(\vec{r}) y_g^+(\vec{r}, \hat{W}) \int_{\hat{\Omega}} dW - \int_{\hat{\Omega}} S_{s,g \otimes g}(\vec{r}) dW y_g^+(\vec{r}, \hat{W}) + y_g^-(\vec{r}, \hat{W}) \int_{\hat{\Omega}} S_{s,g \otimes g}(\vec{r}) dW + S_g(\vec{r}, \hat{W}) \int_{\hat{\Omega}} dW = 0 \quad (11)$$

Equation 11 can be evaluated at $-\hat{\Omega}$ to get

$$-\hat{W} \cdot \nabla y_g^+(\vec{r}, -\hat{W}) + y_g^-(\vec{r}, -\hat{W}) \int_{\hat{\Omega}} S_{t,g}(\vec{r}) dW + S_{t,g}(\vec{r}) y_g^-(\vec{r}, -\hat{W}) \int_{\hat{\Omega}} dW - \int_{\hat{\Omega}} S_{s,g \otimes g}(\vec{r}) dW y_g^-(\vec{r}, -\hat{W}) + y_g^+(\vec{r}, -\hat{W}) \int_{\hat{\Omega}} S_{s,g \otimes g}(\vec{r}) dW + S_g(\vec{r}, -\hat{W}) \int_{\hat{\Omega}} dW = 0 \quad (12)$$

Upon adding equation 11 to equation 12 and using the even and odd parity flux definitions in equation 10 to simplify the new equation, the even parity form of the transport equation, given

by equation 13, is obtained. In a similar fashion, subtraction of equation 12 from equation 11 will lead to the odd parity form of the transport equation, given in equation 14.

Even Parity

$$2\hat{\mathbf{N}}\hat{\mathbf{N}}^{\dagger}y_g^-(\mathbf{r},\hat{\mathbf{W}})+2\mathcal{S}_{t,g}(\mathbf{r})y_g^+(\mathbf{r},\hat{\mathbf{W}})=2\mathcal{S}_{s,g\otimes g}(\mathbf{r})f_g(\mathbf{r})+S_g(\mathbf{r},\hat{\mathbf{W}})+S_g(\mathbf{r},-\hat{\mathbf{W}}) \quad (13)$$

Odd Parity

$$2\hat{\mathbf{N}}\hat{\mathbf{N}}^{\dagger}y_g^+(\mathbf{r},\hat{\mathbf{W}})+2\mathcal{S}_{t,g}(\mathbf{r})y_g^-(\mathbf{r},\hat{\mathbf{W}})=S_g(\mathbf{r},\hat{\mathbf{W}})-S_g(\mathbf{r},-\hat{\mathbf{W}}). \quad (14)$$

To further simplify the equations, an isotropic source will be used in addition to the use of an isotropic scattering kernel (an anisotropic source is discussed in section 2.6). This simplifies equations 13 and 14 to equations 15 and 16, respectively.

Even Parity

$$\hat{\mathbf{W}}\hat{\mathbf{N}}\hat{\mathbf{N}}^{\dagger}y_g^-(\mathbf{r},\hat{\mathbf{W}})+\mathcal{S}_{t,g}(\mathbf{r})y_g^+(\mathbf{r},\hat{\mathbf{W}})=\mathcal{S}_{s,g\otimes g}(\mathbf{r})f_g(\mathbf{r})+S_g(\mathbf{r}) \quad (15)$$

Odd Parity

$$\hat{\mathbf{W}}\hat{\mathbf{N}}\hat{\mathbf{N}}^{\dagger}y_g^+(\mathbf{r},\hat{\mathbf{W}})+\mathcal{S}_{t,g}(\mathbf{r})y_g^-(\mathbf{r},\hat{\mathbf{W}})=0 \quad (16)$$

Equation 16 can now be solved for $y_g^-(\mathbf{r},\hat{\mathbf{W}})$ and substituted into equation 15 to obtain equation 17, the second order form of the even parity transport equation.

$$\hat{\mathbf{W}}\hat{\mathbf{N}}\hat{\mathbf{N}}^{\dagger}\frac{\partial}{\partial r}y_g^+(\mathbf{r},\hat{\mathbf{W}})+\mathcal{S}_{t,g}(\mathbf{r})y_g^+(\mathbf{r},\hat{\mathbf{W}})=\mathcal{S}_{s,g\otimes g}(\mathbf{r})f_g(\mathbf{r})+S_g(\mathbf{r}) \quad (17)$$

This differential equation is accompanied with boundary conditions that constrain the angular flux along the boundary of the problem domain. These boundary conditions are discussed in detail in Section 3.3.

2.3 The variational nodal functional

The even-parity transport equation can be formulated as a variational principle [1, 15]. That is, this equation can be derived as the necessary condition for a functional to become extremum, and its solution can be found by looking for the function that makes the functional extremum. The functional to which equation 17 is the necessary condition for the extremum is given in equation 18. The $y_g^-(\mathbf{r},\hat{\mathbf{W}})$ variable in equation 18 results from the boundary conditions, and is thus known along the problem domain boundary.

$$\begin{aligned} F &= \int_V \hat{\mathbf{N}}\hat{\mathbf{N}}^{\dagger}y_g^+(\mathbf{r},\hat{\mathbf{W}})\frac{\partial}{\partial r}y_g^+(\mathbf{r},\hat{\mathbf{W}})+\mathcal{S}_{t,g}(\mathbf{r})y_g^+(\mathbf{r},\hat{\mathbf{W}})-\mathcal{S}_{s,g\otimes g}(\mathbf{r})f_g(\mathbf{r})-2y_g^+(\mathbf{r},\hat{\mathbf{W}})S_g(\mathbf{r}) \\ &+ 2\hat{\mathbf{N}}\hat{\mathbf{N}}^{\dagger}(\hat{\mathbf{W}}\hat{\mathbf{n}})y_g^+(\mathbf{r},\hat{\mathbf{W}})y_g^-(\mathbf{r},\hat{\mathbf{W}})d\mathbf{W}dV \end{aligned} \quad (18)$$

The functional in equation 18 is defined over the entire problem domain. In a nodal method, the problem domain is split into several nodes and a functional is defined for each node. Figure 2.2 gives an example three-dimensional Cartesian nodal geometry. Given the creation of the nodal geometry, a local or nodal functional F_v can be defined such that it is dependent only upon each node's even parity flux $y_g^+(r, \hat{W})$ and its odd parity boundary or interface flux $c_g^-(r, \hat{W})$ as seen in equation 19. It is important to note that the change from $y_g^-(r, \hat{W})$ to $c_g^-(r, \hat{W})$ was made to indicate that the odd parity flux along each node interface is unknown and that only the odd parity flux along the boundary of the problem domain is known via the boundary conditions.

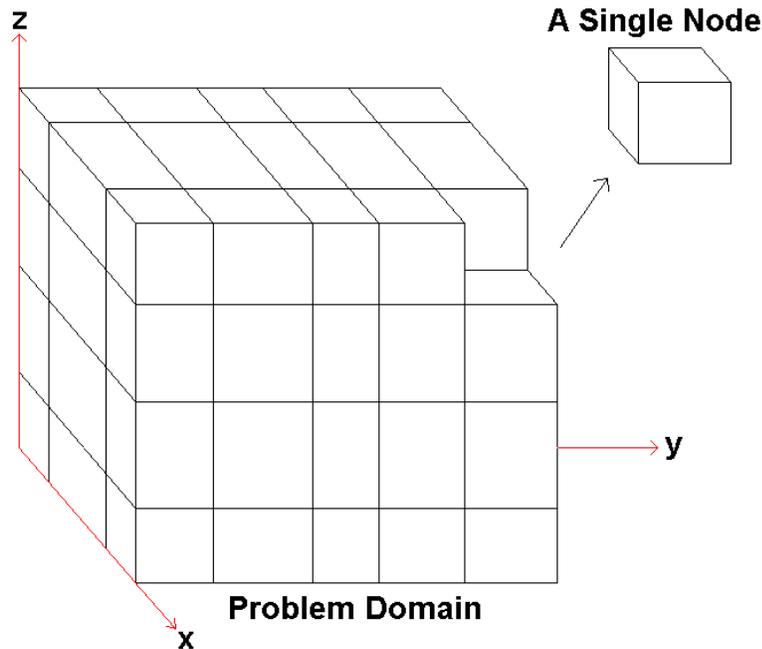


Figure 2.2 An Example Three Dimensional Cartesian Nodal Geometry

$$\begin{aligned}
 F_v(y_g^+(r, \hat{W}), c_g^-(r, \hat{W})) = & \int_V \left[\frac{1}{2} \sum_{t,g} \left(y_g^+(r, \hat{W}) \right)^2 + \sum_{t,g} S_{t,g}(r) y_g^+(r, \hat{W}) \right. \\
 & + \sum_{t,g} S_{t,g}(r) y_g^+(r, \hat{W}) + \sum_{t,g} S_{s,g}(r) f_g(r) \\
 & - 2 \sum_{t,g} S_{t,g}(r) y_g^+(r, \hat{W}) \\
 & \left. + 2 \sum_{t,g} \hat{n}_g \cdot \hat{W} y_g^+(r, \hat{W}) c_g^-(r, \hat{W}) \right] dV
 \end{aligned} \tag{19}$$

The functional from equation 18 (the entire problem domain) can now be expressed as a sum of all of the nodal functionals in the nodal geometry given by

$$F(y_g^+, c_g^-) = \sum_{nodes} F_v(y_g^+, c_g^-) \tag{20}$$

2.4 The variational treatment

The even-parity transport equation within each node and the continuity of even- and odd-parity fluxes at the boundary of adjoining nodes can be derived by requiring the functional in equation 19 to be stationary with respect to variations in $y_g^+(r, \hat{W})$ and $c_g^-(r, \hat{W})$. Suppose that $y_g^+(r, \hat{W})$ is varied from a reference flux $y_g^{0+}(r, \hat{W})$ by an arbitrary amount $dy_g^+(r, \hat{W})$, as shown in equation 21.

$$y_g^+(r, \hat{W}) = y_g^{0+}(r, \hat{W}) + dy_g^+(r, \hat{W}) \quad (21)$$

Then, the first variation of F_v with respect to the arbitrary shift in $y_g^+(r, \hat{W})$ is

$$\begin{aligned} dF_v = & \int_V \left[\frac{\delta F_v}{\delta y_g^+(r, \hat{W})} dy_g^+(r, \hat{W}) + \frac{\delta F_v}{\delta c_g^-(r, \hat{W})} dc_g^-(r, \hat{W}) \right] dV \\ & + \int_V \left[\frac{\delta F_v}{\delta S_{t,g}(r)} \delta S_{t,g}(r) + \frac{\delta F_v}{\delta S_{s,g}(r)} \delta S_{s,g}(r) + \frac{\delta F_v}{\delta S_g(r)} \delta S_g(r) \right] dV \\ & + \int_G \left[\frac{\delta F_v}{\delta (\hat{n}_g \cdot \hat{W})} \delta (\hat{n}_g \cdot \hat{W}) + \frac{\delta F_v}{\delta c_g^-(r, \hat{W})} dc_g^-(r, \hat{W}) \right] dG \end{aligned} \quad (22)$$

The term $\int_V \frac{\delta F_v}{\delta y_g^+(r, \hat{W})} dy_g^+(r, \hat{W}) dV$ can be rewritten using integration by parts to obtain

$$\begin{aligned} \int_V \frac{\delta F_v}{\delta y_g^+(r, \hat{W})} dy_g^+(r, \hat{W}) dV = & \int_V \left[\frac{\delta F_v}{\delta S_{t,g}(r)} \delta S_{t,g}(r) + \frac{\delta F_v}{\delta S_{s,g}(r)} \delta S_{s,g}(r) \right] dV \\ & - \int_V \frac{\delta F_v}{\delta S_g(r)} \delta S_g(r) dV \end{aligned} \quad (23)$$

Use of the divergence theorem can transform equation 23 to

$$\begin{aligned} \int_V \frac{\delta F_v}{\delta y_g^+(r, \hat{W})} dy_g^+(r, \hat{W}) dV = & \int_V \left[\frac{\delta F_v}{\delta S_{t,g}(r)} \delta S_{t,g}(r) + \frac{\delta F_v}{\delta S_{s,g}(r)} \delta S_{s,g}(r) \right] dV \\ & - \int_G \frac{\delta F_v}{\delta S_g(r)} \delta S_g(r) dG \end{aligned} \quad (24)$$

2.5 Obtaining the response matrix form

As mentioned above, in the variational approach, an approximate solution of equation 17 is determined by looking for the function that makes the functional in equation 19 extremum in a pre-selected function space. Given this fact, the focus now turns to the details of the flux approximation. To start the response matrix formulation the following approximations are made to the group even parity flux, the group odd parity flux, and the group source, where the subscripts i, j, m, n , and g imply summation. The g index is used to indicate that the surface of each node is split up into several components.

$$y_g^+(r, \hat{W}) = f_i(r) g_m(\hat{W}) z_{g,i,m} \quad (30)$$

$$c_{g,g}^-(r, \hat{W}) = h_{g,j}(r) k_{g,n}(\hat{W}) c_{g,g,j,n} \quad (31)$$

$$S_g(r) = f_i(r) s_{g,i} \quad (32)$$

$z_{g,i,m}$ and $c_{g,g,j,n}$ represent the unknown coefficients that come with the known spatial trial functions $f_i(r)$ and $h_{g,j}(r)$ and the known angular trial functions $g_m(\hat{W})$ and $k_{g,n}(\hat{W})$.

With these approximations the following matrices can be defined for each node.

$$P_{i,j,g}^{k,l} = \int_{S_{t,g}(r)} \tilde{N}_k f_i(r) \tilde{N}_l f_j(r) dV \quad (33)$$

$$F_{i,j} = \int f_i(r) f_j(r) dV \quad (34)$$

$$F_{i,j,g}^t = \int S_{t,g}(r) f_i(r) f_j(r) dV \quad (35)$$

$$F_{i,j,g}^s = \int S_{s,g} \otimes_g(r) f_i(r) f_j(r) dV \quad (36)$$

$$D_{i,j}^g = \int f_i(r) h_{g,j}(r) dG \quad (37)$$

$$H_{m,n}^{k,l} = \int \hat{W}_k \hat{W}_l g_m(\hat{W}) g_n(\hat{W}) dW \quad (38)$$

$$I_{m,n} = \int g_m(\hat{W}) g_n(\hat{W}) dW \quad (39)$$

$$J_{m,n} = \int g_m(\hat{W}) d_{n,l} dW \quad (40)$$

$$E_{m,n}^g = \int (\hat{W} \otimes_g) g_m(\hat{W}) k_{g,n}(\hat{W}) dW \quad (41)$$

The functional from equation 17 can be rewritten as a tensor product (denoted by the symbol \otimes) of the matrices defined in equations 33 through 41 resulting in

$$F_v = z_g^T \begin{pmatrix} \hat{e}_0 \\ \hat{e}^{K,L} \end{pmatrix} \bar{P}_g^{K,L} \bar{H}^{K,L} \begin{pmatrix} \hat{u} \\ \hat{u} \end{pmatrix} + z_g^T \begin{pmatrix} \hat{e}_g^t \\ \hat{e}_g^s \end{pmatrix} \bar{I}_g \begin{pmatrix} \hat{u} \\ \hat{u} \end{pmatrix} - z_g^T \begin{pmatrix} \hat{e}_g^s \\ \hat{e}_g^t \end{pmatrix} \bar{J}_g \begin{pmatrix} \hat{u} \\ \hat{u} \end{pmatrix} - 2 z_g^T \begin{pmatrix} \hat{e}_g^t \\ \hat{e}_g^s \end{pmatrix} \bar{J}_g \begin{pmatrix} \hat{u} \\ \hat{u} \end{pmatrix} + 2 z_g^T \hat{a}_g (\bar{D}^g \otimes \bar{E}^g) c_{g,g} \quad (42)$$

Equation 42 can be written in a more compact form by using the following nodal matrix definitions.

$$\bar{A}_g = \overset{\circ}{\mathbf{a}}_{K,L} \bar{P}_g^{K,L} \bar{H}^{K,L} + \bar{F}_g^T \bar{I} - \bar{F}_g^S \bar{J} \quad (43)$$

$$S_g = \bar{F} \bar{J} S_g \quad (44)$$

$$\bar{M}_g = \bar{D}_g \bar{E}_g \quad (45)$$

With these matrix definitions, the compact form of the functional is given as

$$F_v = \mathbf{z}_g^T \bar{A}_g \mathbf{z}_g - 2 \mathbf{z}_g^T S_g + 2 \mathbf{z}_g^T \overset{\circ}{\mathbf{a}}_g \bar{M}_g \mathbf{c}_{g,g} \quad (46)$$

Although not shown here [1,7], the variation of this functional with respect to \mathbf{z}_g results in

$$\bar{A}_g \mathbf{z}_g - S_g + \overset{\circ}{\mathbf{a}}_g \bar{M}_g \mathbf{c}_{g,g} = 0 \quad (47)$$

Solving for the internal flux coefficients \mathbf{z}_g , equation 47 can be written as

$$\mathbf{z}_g = \bar{A}_g^{-1} S_g - \bar{A}_g^{-1} \overset{\circ}{\mathbf{a}}_g \bar{M}_g \mathbf{c}_{g,g} \quad (48)$$

Equation 48 gives the solution of the nodal even parity flux within each node in terms of the nodal odd parity boundary flux and the internal source. In order to use a nodal response matrix approach an additional equation needs to be defined such that feedback for the nodal odd parity boundary flux based upon the nodal even parity flux is obtained. This new equation must allow the even and odd parity flux between nodes to be coupled together. The method used in VARIANT to obtain the new equation is to define the even parity function in equation 49 along the boundary of the node and project the nodal even parity flux onto it.

$$j_{g,g}(\hat{\mathbf{r}}, \hat{\mathbf{W}}) = h_{g,j}(\hat{\mathbf{r}}) j_{g,g,j}(\hat{\mathbf{W}}) = h_{g,j}(\hat{\mathbf{r}}) p_{g,n}(\hat{\mathbf{W}}) j_{g,g,j,n} \quad (49)$$

The projection of the spatial approximation takes the following form

$$\oint h_{g,j}(\hat{\mathbf{r}}) j_{g,g}(\hat{\mathbf{r}}, \hat{\mathbf{W}}) dG = \oint h_{g,j}(\hat{\mathbf{r}}) y_g^+(\hat{\mathbf{r}}) \hat{\mathbf{G}} \hat{\mathbf{W}} dG \quad (50)$$

Equation 51 is obtained after substitution of the spatial trial functions $h_{g,j}(\hat{\mathbf{r}})$ used along the boundary of the node.

$$\oint h_{g,j}(\hat{\mathbf{r}}) h_{g,k}(\hat{\mathbf{r}}) j_{g,g,k}(\hat{\mathbf{W}}) dG = \oint h_{g,j}(\hat{\mathbf{r}}) f_i(\hat{\mathbf{r}}) y_{g,i}^+(\hat{\mathbf{W}}) dG \quad (51)$$

If the $h_{g,j}(\hat{\mathbf{r}})$ trial functions are defined to be orthogonal polynomials, the left hand side of equation 51 becomes the identity matrix \bar{I} and the right hand side becomes the spatial interface matrix \bar{D}_g from equation 37. Using this approach, equation 51 is expressed as

$$\bar{I} \mathbf{x}_{g,g}(\hat{\mathbf{W}}) = \bar{D}_g^T \mathbf{x}_g^+(\hat{\mathbf{W}}) \quad (52)$$

Given this definition of the spatial approximation along the node boundary, the angular approximation along the node boundary can now be considered.

The even parity functions $p_{g,n}(\hat{\mathbf{W}})$ could be defined using the same nodal even parity trial functions $g_m(\hat{\mathbf{W}})$ from equation 30; however, using such an approach makes it difficult to satisfy

the boundary and continuity conditions. The method used in VARIANT is to make use of the existing odd parity trial functions multiplied by an additional odd parity function to make them even parity, as seen in equation 53.

$$P_{g,n}(\hat{W}) = (\hat{W} \chi_g) k_{g,n}(\hat{W}). \quad (53)$$

This approach allows for a straightforward treatment of the boundary and continuity conditions and satisfies the underlying Rumyanstev boundary conditions [13-14]. Using equation 53 in equation 52 the following relation is obtained for the projection

$$I \bar{A} \bar{Q} (\hat{W} \chi_g) k_{g,n}(\hat{W}) j_{g,g}(\hat{W}) dW = D_g^T \bar{A} \bar{Q} (\hat{W} \chi_g) k_{g,n}(\hat{W}) y_g^+(\hat{W}) dW. \quad (54)$$

The right hand side of equation 54 will lead directly back to the earlier definition of the \bar{M}_g matrix in equation 45, except that it is transposed here. The angular integral on the left hand side of equation 54 can be written as the following matrix

$$Q_{g,n,m} = \bar{Q} (\hat{W} \chi_g) k_{g,n}(\hat{W}) (\hat{W} \chi_g) k_{g,m}(\hat{W}) dW, \quad (55)$$

which is not an identity matrix. Rather than inverting this matrix to solve for the $j_{g,g}$ coefficients, the set of coefficients $y_{g,g}$ defined by equation 56 are used.

$$y_{g,g} = (I \bar{A} Q) j_{g,g} = M_g^T z_g. \quad (56)$$

This is appropriate because the continuity condition for the transport equation is exactly preserved with the set of coefficients in equation 56 [1,14-15]. After substitution of equation 48 into equation 56, equation 57 is obtained.

$$y_{g,g} = M_g^T z_g = \bar{M}_g^T \bar{A}_g^{-1} \bar{S}_g - \bar{M}_g^T \bar{A}_g^{-1} \mathbf{a}_{g'} \bar{M}_g \mathbf{c}_{g',g} \quad (57)$$

As can be seen, equation 57 represents the even parity boundary flux as a function of the internal source and odd parity boundary fluxes. Because equation 57 does not provide direct feedback to the odd parity boundary flux used to get z_g in equation 48, both the even and odd parity boundary fluxes must be solved for simultaneously. To do this a change of variable to j_g^+ and j_g^- is carried out where j_g^+ and j_g^- are analogous to partial currents across the node interface [5].

$$j_{g,g}^{\pm} = \frac{1}{4} y_{g,g} \pm \frac{1}{2} c_{g,g} \quad (58)$$

Upon rearrangement of equation 58, the following definitions for $y_{g,g}$ and $c_{g,g}$ in terms of j_g^+ and j_g^- are obtained.

$$y_{g,g} = 2(j_{g,g}^+ + j_{g,g}^-) \quad (59)$$

$$c_{g,g} = j_{g,g}^+ - j_{g,g}^- \quad (60)$$

Substitution of these expressions into equation 57 results in

$$2(j_{g,g}^+ + j_{g,g}^-) = \bar{M}_g^T \bar{A}_g^{-1} \bar{S}_g - \bar{M}_g^T \bar{A}_g^{-1} \mathbf{a}_{g'} \bar{M}_g (j_{g',g}^+ - j_{g',g}^-). \quad (61)$$

Equation 61 can be simplified by defining the matrices in equations 62 and 63.

$$\bar{G}_{g,g',g} = \frac{1}{2} \bar{M}_g^T \bar{A}_g^{-1} \bar{M}_{g'} \quad (62)$$

$$\bar{C}_{g,g} = \frac{1}{2} \bar{M}_g^T \bar{A}_g^{-1} \quad (63)$$

Using these matrices equation 61 is written as

$$j_{g,g}^+ + j_{g,g}^- = \bar{C}_{g,g} \bar{S}_g - \underset{g'}{\mathbf{a}} \bar{G}_{g,g',g} (j_{g',g}^+ - j_{g',g}^-). \quad (64)$$

At this point equation 64 is partitioned [4] with respect to the surface vectors j_g^+ and j_g^- , such that the following nodal interface vectors and matrices are created.

$$j^+ = [j_1^+, j_2^+, j_3^+, \dots, j_g^+]^T \quad (65)$$

$$j^- = [j_1^-, j_2^-, j_3^-, \dots, j_g^-]^T \quad (66)$$

$$\bar{G}_g = \begin{matrix} \begin{matrix} \bar{G}_{1,1,g} & \bar{G}_{1,2,g} & L & \bar{G}_{1,g,g} \\ \bar{G}_{2,1,g} & \bar{G}_{2,2,g} & L & M \\ M & M & O & M \\ \bar{G}_{g,1,g} & L & L & \bar{G}_{g,g,g} \end{matrix} \end{matrix} \quad (67)$$

$$\bar{C}_g = [\bar{C}_{1,g}, \bar{C}_{2,g}, \bar{C}_{3,g}, \dots, \bar{C}_{g,g}]^T \quad (68)$$

Rewriting equation 64 using these nodal vectors and matrices yields

$$j_g^+ + j_g^- = \bar{C}_g \bar{S}_g - \bar{G}_g (j_g^+ - j_g^-). \quad (69)$$

Rearranging equation 69 to solve for j^+ leads to equations 70 and 71, where \bar{I} represents an identity matrix.

$$(\bar{G}_g + \bar{I}) j_g^+ = \bar{C}_g \bar{S}_g + (\bar{G}_g - \bar{I}) j_g^- \quad (70)$$

$$j_g^+ = (\bar{G}_g + \bar{I})^{-1} \bar{C}_g \bar{S}_g + (\bar{G}_g + \bar{I})^{-1} (\bar{G}_g - \bar{I}) j_g^- \quad (71)$$

Equation 71 can be written compactly by defining the matrices in equations 72 and 73.

$$\bar{R}_g = (\bar{G}_g + \bar{I})^{-1} (\bar{G}_g - \bar{I}) \quad (72)$$

$$\bar{B}_g = (\bar{G}_g + \bar{I})^{-1} \bar{C}_g \quad (73)$$

Insertion of equations 72 and 73 into 71 leads to the final nodal response matrix form in equation 74.

$$j_g^+ = \bar{B}_g \bar{S}_g + \bar{R}_g j_g^- \quad (74)$$

The final step in the nodal approach is to substitute the partial current definition from equation 58 for the odd parity boundary flux in equation 48, reiterated as equation 75 below, to arrive at equation 76.

$$z_g = \bar{A}_g^{-1} \bar{S}_g - \bar{A}_g^{-1} \underset{g}{\mathbf{a}} \bar{M}_g c_{g,g} \quad (75)$$

$$z_g = \bar{A}_g^{-1} \bar{S}_g - 2 \bar{C}_g^T (j_g^+ - j_g^-) \quad (76)$$

The production code VARIANT is based upon the iterative solution of equations 74 and 76. The changes made in this work are focused mainly on the definition and evaluation of the spatial matrices in equations 33-37 and on the definition of the angular matrices in equations 38-41. The details of the solution procedure and convergence methods can be found in the VARIANT manual [1].

2.6 Anisotropic Scattering Considerations

Anisotropic scattering is made included in this section and the response matrix derivation above is repeated. The starting point is taken as the multigroup even and odd parity transport equations given by equations 77 and 78 taken from equations 13 and 14.

Even Parity

$$\hat{\mathbf{W}} \times \hat{\mathbf{N}} y_g^-(r, \hat{\mathbf{W}}) + S_{t,g}(r) y_g^+(r, \hat{\mathbf{W}}) = \int_0^{\infty} dW \hat{\mathbf{S}}_{s,g@g}^+(r, \hat{\mathbf{W}} \times \hat{\mathbf{W}}) y_g^+(r, \hat{\mathbf{W}}) + S_g^+(r, \hat{\mathbf{W}}) \quad (77)$$

Odd Parity

$$\hat{\mathbf{W}} \times \hat{\mathbf{N}} y_g^+(r, \hat{\mathbf{W}}) + S_{t,g}(r) y_g^-(r, \hat{\mathbf{W}}) = \int_0^{\infty} dW \hat{\mathbf{S}}_{s,g@g}^-(r, \hat{\mathbf{W}} \times \hat{\mathbf{W}}) y_g^-(r, \hat{\mathbf{W}}) + S_g^-(r, \hat{\mathbf{W}}) \quad (78)$$

where the terms $\hat{\mathbf{S}}_{s,g@g}^+(r, \hat{\mathbf{W}} \times \hat{\mathbf{W}})$, $\hat{\mathbf{S}}_{s,g@g}^-(r, \hat{\mathbf{W}} \times \hat{\mathbf{W}})$, $S_g^+(r, \hat{\mathbf{W}})$, and $S_g^-(r, \hat{\mathbf{W}})$ are defined by

$$S_{s,g@g}^+(r, \hat{\mathbf{W}} \times \hat{\mathbf{W}}) = \int_0^N \hat{\mathbf{a}} \hat{\mathbf{a}} S_{l,s,g@g}^+(r) Y_{l,m}^e(\hat{\mathbf{W}}) Y_{l,m}^e(\hat{\mathbf{W}}), \quad (79)$$

$l=0,2,\dots, m$

$$S_{s,g@g}^-(r, \hat{\mathbf{W}} \times \hat{\mathbf{W}}) = \int_0^N \hat{\mathbf{a}} \hat{\mathbf{a}} S_{l,s,g@g}^-(r) Y_{l,m}^o(\hat{\mathbf{W}}) Y_{l,m}^o(\hat{\mathbf{W}}), \quad (80)$$

$l=1,3,\dots, m$

$$S_g^+(r, \hat{\mathbf{W}}) = S_g(r, \hat{\mathbf{W}}) + S_g(r, -\hat{\mathbf{W}}), \quad (81)$$

and

$$S_g^-(r, \hat{\mathbf{W}}) = S_g(r, \hat{\mathbf{W}}) - S_g(r, -\hat{\mathbf{W}}). \quad (82)$$

Solving equation 78 for the odd parity flux gives

$$y_g^-(r, \hat{\mathbf{W}}) = \frac{1}{S_{t,g}(r)} \int_0^{\infty} dW \hat{\mathbf{S}}_{s,g@g}^-(r, \hat{\mathbf{W}} \times \hat{\mathbf{W}}) y_g^-(r, \hat{\mathbf{W}}) + S_g^-(r, \hat{\mathbf{W}}) \quad (83)$$

Each component of the odd parity spherical harmonic expansion in equation 80 can be extracted from the odd parity flux in equation 83 by multiplying by the spherical harmonic and integrating over the angular domain as shown in equation 84.

$$\int_0^{\infty} dW \hat{\mathbf{S}}_{l,m}^o(\hat{\mathbf{W}}) y_g^-(r, \hat{\mathbf{W}}) = \frac{1}{S_{t,g}(r)} \int_0^{\infty} dW \hat{\mathbf{S}}_{l,m}^o(\hat{\mathbf{W}}) \int_0^{\infty} dW \hat{\mathbf{S}}_{s,g@g}^-(r, \hat{\mathbf{W}} \times \hat{\mathbf{W}}) y_g^-(r, \hat{\mathbf{W}}) + S_g^-(r, \hat{\mathbf{W}}) \quad (84)$$

$$+ \frac{1}{S_{t,g}(r)} \int_0^{\infty} dW \hat{\mathbf{S}}_{l,m}^o(\hat{\mathbf{W}}) \int_0^{\infty} dW \hat{\mathbf{S}}_{s,g@g}^-(r, \hat{\mathbf{W}} \times \hat{\mathbf{W}}) y_g^-(r, \hat{\mathbf{W}})$$

Substitution of the cross section definition of equation 80 into the last term of equation 84 allows for the simplification of this term as seen in equation 85.

$$\begin{aligned} & \frac{\partial}{\partial \hat{W}} dW_{l,m}^{s,g}(\hat{W}) \frac{\partial}{\partial \hat{W}} dW_{s,g}^{s,g}(\hat{W}) y_g^-(r, \hat{W}) \frac{\partial}{\partial \hat{W}} \\ & \frac{\partial}{\partial \hat{W}} dW_{l,m}^{s,g}(\hat{W}) \frac{\partial}{\partial \hat{W}} \sum_{l=1,3,\dots}^N \sum_{m'}^m \hat{a}_{l,s,g}^{s,g}(\hat{W}) y_{l,m'}^o(\hat{W}) y_g^-(r, \hat{W}) \frac{\partial}{\partial \hat{W}} \end{aligned} \quad (85)$$

Substitution of equation 85 into 84 yields equation 86.

$$\begin{aligned} & \frac{\partial}{\partial \hat{W}} dW_{l,m}^{s,g}(\hat{W}) y_g^-(r, \hat{W}) \frac{\partial}{\partial \hat{W}} + \frac{1}{S_{t,g}(r)} \frac{\partial}{\partial \hat{W}} dW_{l,m}^{s,g}(\hat{W}) \hat{W} y_g^+(r, \hat{W}) + S_g^-(r, \hat{W}) \frac{\partial}{\partial \hat{W}} \\ & + \frac{S_{l,s,g}^{s,g}(r)}{S_{t,g}(r)} \frac{\partial}{\partial \hat{W}} dW_{l,m}^{s,g}(\hat{W}) y_g^-(r, \hat{W}) \frac{\partial}{\partial \hat{W}} \end{aligned} \quad (86)$$

Collecting on similar terms, equation 86 is simplified to equation 87.

$$\begin{aligned} & \frac{\partial}{\partial \hat{W}} dW_{l,m}^{s,g}(\hat{W}) y_g^-(r, \hat{W}) \frac{\partial}{\partial \hat{W}} \\ & \frac{1}{(S_{t,g}(r) - S_{l,s,g}^{s,g}(r))} \frac{\partial}{\partial \hat{W}} dW_{l,m}^{s,g}(\hat{W}) \hat{W} y_g^+(r, \hat{W}) + S_g^-(r, \hat{W}) \frac{\partial}{\partial \hat{W}} \end{aligned} \quad (87)$$

The odd parity flux in equation 83 can be eliminated from the right hand side by substitution of equation 87 yielding equation 88.

$$\begin{aligned} y_g^-(r, \hat{W}) &= \frac{1}{S_{t,g}(r)} \frac{\partial}{\partial \hat{W}} dW_{l,m}^{s,g}(\hat{W}) \hat{W} y_g^+(r, \hat{W}) + S_g^-(r, \hat{W}) \frac{\partial}{\partial \hat{W}} \\ & + \frac{\partial}{\partial \hat{W}} dW_{l,m}^{s,g}(\hat{W}) \frac{\partial}{\partial \hat{W}} \frac{S_{l,s,g}^{s,g}(r)}{S_{t,g}(r)} \hat{W} y_g^+(r, \hat{W}) + S_g^-(r, \hat{W}) \frac{\partial}{\partial \hat{W}} \end{aligned} \quad (88)$$

The cross section $S_{l,s,g}^{s,g}(r, \hat{W})$ is defined by equation 89.

$$S_{l,s,g}^{s,g}(r, \hat{W}) = \sum_{l=1,3,\dots}^N \sum_{m'}^m \hat{a}_{l,s,g}^{s,g}(\hat{W}) \frac{S_{l,s,g}^{s,g}(r)}{S_{t,g}(r)} \frac{\partial}{\partial \hat{W}} dW_{l,m}^{s,g}(\hat{W}) \quad (89)$$

Substituting equation 88 into the even parity transport equation (equation 77) results in equation 90.

$$\begin{aligned} & \hat{W} y_g^-(r, \hat{W}) \frac{1}{S_{t,g}(r)} \hat{W} y_g^+(r, \hat{W}) + \hat{W} y_g^-(r, \hat{W}) \frac{1}{S_{t,g}(r)} \frac{\partial}{\partial \hat{W}} dW_{l,m}^{s,g}(\hat{W}) \frac{\partial}{\partial \hat{W}} \frac{S_{l,s,g}^{s,g}(r)}{S_{t,g}(r)} \hat{W} y_g^+(r, \hat{W}) \\ & + S_{t,g}(r) y_g^+(r, \hat{W}) = \\ & + \frac{\partial}{\partial \hat{W}} dW_{l,m}^{s,g}(\hat{W}) y_g^+(r, \hat{W}) \frac{\partial}{\partial \hat{W}} + S_g^+(r, \hat{W}) \\ & - \hat{W} y_g^-(r, \hat{W}) \frac{1}{S_{t,g}(r)} \frac{\partial}{\partial \hat{W}} dW_{l,m}^{s,g}(\hat{W}) \frac{\partial}{\partial \hat{W}} \frac{S_{l,s,g}^{s,g}(r)}{S_{t,g}(r)} S_g^-(r, \hat{W}) - \hat{W} y_g^-(r, \hat{W}) \frac{1}{S_{t,g}(r)} S_g^-(r, \hat{W}) \end{aligned} \quad (90)$$

The corresponding functional for equation 90 is given in equation 91.

$$\begin{aligned}
 F_v \delta y^+, c^- \delta y^- &= \delta \int dV \delta \int dW \left[\hat{W}^+ N y^+ - \frac{1}{S_t} \hat{W}^+ N y^+ + \hat{W}^+ N y^+ \frac{1}{S_t} \delta \int dW \frac{\delta \hat{W}^+}{S_t} \hat{W}^+ N y^+ \right. \\
 &\quad \left. - y^+ S_t y^+ - y^+ \delta \int dW \frac{\delta \hat{W}^+}{S_t} y^+ \right] 2y^+ S^+ \\
 &\quad - \delta \int dV \delta \int dW \left[\hat{W}^+ N y^+ \frac{1}{S_t} \delta \int dW \frac{\delta \hat{W}^+}{S_t} S^- \frac{\delta \hat{W}^+}{S_t} + 2 \hat{W}^+ N y^+ \frac{1}{S_t} S^- \right. \\
 &\quad \left. + 2 \delta \int dV \delta \int dW \hat{W}^+ N y^+ c^- \delta y^- \right]
 \end{aligned} \tag{91}$$

Performing the variation with respect to the even parity flux in equation 91 gives equation 90 as the Euler-Lagrange equation. Performing the variation with respect to the odd parity flux in equation 91 leads to the same continuity conditions found with the isotropic scattering functional.

To simplify the matrix notation and allow the use of tensor notation as done before, the spherical harmonics for the anisotropic scattering treatment are arranged into a single indexed vector as seen in equation 92 (two-dimensional version). The anisotropic scattering cross section moments must also be aligned to match $N^\pm(\hat{W})$, the details of which can be found elsewhere [1,7].

$$N^+(\hat{W}) = \begin{pmatrix} Y_{0,0}(\hat{W}) \\ Y_{2,0}(\hat{W}) \\ Y_{2,1}(\hat{W}) \\ \vdots \\ M \\ \vdots \\ Y_{l=even,m}(\hat{W}) \end{pmatrix} \quad N^-(\hat{W}) = \begin{pmatrix} Y_{1,0}(\hat{W}) \\ Y_{1,1}(\hat{W}) \\ Y_{3,0}(\hat{W}) \\ \vdots \\ M \\ \vdots \\ Y_{l=odd,m}(\hat{W}) \end{pmatrix} \tag{92}$$

Implementation of the spatial and angular approximations leads to the definition of the following angular and spatial matrices.

$${}^t P_{i,j}^{K,L} = \delta \int \frac{1}{S_t(r)} \tilde{N}^K f_i(r) \tilde{N}^L f_j(r) dV \tag{93}$$

$$\begin{aligned}
 {}^{st} P_{i,j,l}^{K,L} &= \delta \int \frac{1}{S_t(r)} \frac{\delta \hat{W}^+}{S_t(r)} \tilde{N}^K f_i(r) \tilde{N}^L f_j(r) dV \\
 &= \delta \int \frac{1}{S_t(r)} \frac{S_{s,l}^-(r)}{(S_t(r) - S_{s,l}^-(r))} \tilde{N}^K f_i(r) \tilde{N}^L f_j(r) dV
 \end{aligned} \tag{94}$$

$$U_{i,j,l}^K = \delta \int \frac{1}{S_t(r) - S_{s,l}^-(r)} f_i(r) \tilde{N}^K f_j(r) dV \tag{95}$$

$$F_{i,j} = \delta \int f_i(r) f_j(r) dV \tag{96}$$

$$F_{i,j}^t = \delta \int S_t(r) f_i(r) f_j(r) dV \tag{97}$$

$$F_{i,j,l}^s = \delta \int S_{s,l}^+(r) f_i(r) f_j(r) dV \tag{98}$$

$$D_{i,j}^g = \int \delta f_i(\mathbf{r}) h_{g,j}(\mathbf{r}) dG \quad (99)$$

$$H_{m,n}^{k,l} = \int \delta \hat{W}_k \hat{W}_l g_m(\hat{W}) g_n(\hat{W}) dW \quad (100)$$

$$V_{i,l}^k = \int \delta \hat{W}_k g_i(\hat{W}) N_l^-(\hat{W}) dW \quad (101)$$

$$I_{m,n} = \int \delta g_m(\hat{W}) g_n(\hat{W}) dW \quad (102)$$

$$J_{i,l} = \int \delta g_i(\hat{W}) N_l^+(\hat{W}) dW \quad (103)$$

$$E_{m,n}^g = \int \delta (\hat{W} \hat{\kappa}_g) g_m(\hat{W}) k_{g,n}(\hat{W}) dW \quad (104)$$

Upon substitution of equations 93-104 into the functional in equation 91, one obtains the discretized functional of equation 105.

$$F_v[z, c] = z^T \int \delta \hat{H}_{K,L}^{K,L} \delta \bar{P}^{K,L} + \int \delta \bar{V}_{K,l} \bar{V}_{L,l}^{T} \delta \bar{P}_l^{K,L} + \bar{T} \delta \bar{F}^t - \int \delta \bar{J}_l^T \delta \bar{F}_l^s - 2z^T \int \delta \bar{J}_l^T \delta \bar{F} S_l^+ - 2z^T \int \delta \bar{V}_{K,l} \delta \bar{U}_l^K S_l^- + 2z^T \int \delta (\bar{D}^g \delta \bar{E}^g) c_g \quad (105)$$

It is important to note that the matrices in equations 101 and 103 are really vectors for each spherical harmonic moment of the anisotropic scattering expansion. As was done for the isotropic functional, equation 105 can be simplified using the following matrix definitions.

$$\bar{A} = \int \delta \bar{H}_{K,L}^{K,L} \delta \bar{P}^{K,L} + \int \delta \bar{V}_{K,l} \bar{V}_{L,l}^{T} \delta \bar{P}_l^{K,L} + \bar{T} \delta \bar{F}^t - \int \delta \bar{J}_l^T \delta \bar{F}_l^s \quad (106)$$

$$\bar{T}_l = \int \delta \bar{V}_{K,l} \delta \bar{U}_l^K \quad (107)$$

$$\bar{W}_l = \int \delta \bar{J}_l^T \delta \bar{F} \quad (108)$$

$$\bar{M}_g = \bar{D}^g \delta \bar{E}^g \quad (109)$$

Substituting equations 106-109 into equation 105 and partitioning the \bar{T}_l matrix and the even and odd parity source components, yields equation 110.

$$F_v[z, c] = z^T \bar{A} z - 2z^T \bar{W} S^+ - 2z^T \bar{T} S^- + 2z^T \bar{M} c \quad (110)$$

Performing the variation with respect to the even parity flux coefficients in equation 110 and solving the resulting equation for the even parity flux coefficients leads to equation 111.

$$z = \bar{A}^{-1} \bar{W} S^+ + \bar{A}^{-1} \bar{T} S^- - \bar{A}^{-1} \bar{M} c \quad (111)$$

Following the same response matrix approach used for the isotropic functional, the following response matrix equations are obtained.

$$j^+ = \bar{B}^+ S^+ + \bar{B}^- S^- + \bar{R} c \quad (112)$$

$$\bar{G} = \frac{1}{2} \bar{M}^T \bar{A}^{-1} \bar{M} \quad (113)$$

$$\bar{C} = \frac{1}{2} \bar{M}^T \bar{A}^{-1} \quad (114)$$

$$\bar{B}^+ = (\bar{G} + \bar{I})^{-1} \bar{C} \quad (115)$$

$$\bar{B}^- = \bar{B}^+ \bar{T} \quad (116)$$

$$\bar{R} = (\bar{G} + \bar{I})^{-1} (\bar{G} - \bar{I}) \quad (117)$$

$$\mathbf{z} = \bar{A}^{-1} \bar{W} S^+ + \bar{A}^{-1} \bar{T} S^- - 2\bar{C}^T \mathbf{c} \quad (118)$$

The details on how to obtain the even and odd parity source moments are given in the VARIANT manual. This concludes the description of the theory implemented in the VARIANT code. Section 3 deals with the angular approximations implemented in VARIANT while Section 4 deals with the spatial approximations.

3 The Angular Approximation Implemented in VARIANT

3.1 Introduction

In Section 2, the response matrix equations were derived with the assumption that the flux would be approximated with a set of spatial and angular trial functions. In this section the angular trial functions are displayed. In the production version of VARIANT, the angular and spatial integrals were evaluated analytically using MATHEMATICA. The results of these integrations were placed into FORTRAN data statements and placed into the coding during compilation. In this work the angular trial functions and matrices are obtained numerically during VARIANT code execution for a user specified order. In this section, the procedure used to obtain the angular trial functions and the method used to obtain the angular matrices and to implement them into VARIANT is detailed.

3.2 Angular approximation

The production version of VARIANT can use either spherical harmonics or simplified spherical harmonics to approximate the even parity trial functions, $g(\hat{\mathbf{W}})$, and the odd parity trial functions, $k_g(\hat{\mathbf{W}})$. In this report only the spherical harmonics formulation will be derived; the derivation of the simplified spherical harmonics approximation can be found elsewhere [1].

The spherical harmonic functions [15] are defined as

$$Y_l^m(\hat{\mathbf{W}}) = \sqrt{C_l^m} \times P_l^m(m) \times e^{imf}, \quad (119)$$

where $P_l^m(m)$ represents the associated Legendre function ($m = \cos(\theta)$) and the normalization constant is given by

$$C_l^m = \frac{(2l+1)(l-m)!}{(l+m)!}. \quad (120)$$

The spherical harmonic functions can be separated into sine and cosine series as shown in equations 121-123.

$$Y_l^m(\hat{\mathbf{W}}) = Y_{l,m}^e(\hat{\mathbf{W}}) + i Y_{l,m}^o(\hat{\mathbf{W}}) \quad (121)$$

$$Y_{l,m}^c(\hat{\mathbf{W}}) = \sqrt{C_l^m} \times P_l^m(m) \times \cos(mf) \quad (122)$$

$$Y_{l,m}^s(\hat{\mathbf{W}}) = \sqrt{C_l^m} \times P_l^m(m) \times \sin(mf) \quad (123)$$

What is unique about equations 122 and 123 is that the functions $Y_{l,m}^c(\hat{\mathbf{W}})$ are symmetric with respect to f and the functions $Y_{l,m}^s(\hat{\mathbf{W}})$ are not. Also important is that the functions with even l are even parity and odd l are odd parity. These properties of the spherical harmonic functions are quite beneficial when defining the functions $g(\hat{\mathbf{W}})$ and $k_g(\hat{\mathbf{W}})$ as will be shown shortly.

To satisfy the continuity conditions along nodal boundaries, Romyantsev boundary conditions [13] are implemented in VARIANT. Romyantsev has shown that the integrals for the total flux in equation 124 must be continuous across interfaces for the spherical harmonics approximation.

$$\oint (\hat{\mathbf{W}} \times \hat{\mathbf{n}}_g) Y_l^m (\hat{\mathbf{W}}) y^{\pm} (r \hat{\mathbf{I}} \cdot \mathbf{G} \hat{\mathbf{W}}) dW |l| = 0, 1, 2, \dots, N \quad |m| = 0, 1, 2, \dots, l \quad (124)$$

Separating the angular flux along the boundary into even and odd parity components yields equations 125 and 126. It is important to note that the range of l has been adjusted in these equations to compensate for the parity dependence upon the associated Legendre function, which forces some of the terms automatically to zero [15].

$$\oint (\hat{\mathbf{W}} \times \hat{\mathbf{n}}_g) Y_l^m (\hat{\mathbf{W}}) y^+ (\hat{\mathbf{W}}) dW \quad l = 1, 3, 5, \dots, N \quad |m| = 0, 1, 2, \dots, l \quad (125)$$

$$\oint (\hat{\mathbf{W}} \times \hat{\mathbf{n}}_g) Y_l^m (\hat{\mathbf{W}}) y^- (\hat{\mathbf{W}}) dW \quad l = 0, 2, 4, \dots, N-1 \quad |m| = 0, 1, 2, \dots, l \quad (126)$$

The even and odd parity flux can be expanded in spherical harmonics as shown in equations 127 and 128.

$$y^+ (\hat{\mathbf{W}}) = Y_l^m (\hat{\mathbf{W}}) y_{l,m}^+ \quad l = 0, 2, 4, \dots, N-1 \quad |m| = 0, 1, 2, \dots, l \quad (127)$$

$$y^- (\hat{\mathbf{W}}) = Y_l^m (\hat{\mathbf{W}}) y_{l,m}^- \quad l = 1, 3, 5, \dots, N \quad |m| = 0, 1, 2, \dots, l \quad (128)$$

Upon insertion of equations 127 and 128 into equations 125 and 126, equations 129 and 130 are obtained.

$$\oint (\hat{\mathbf{W}} \times \hat{\mathbf{n}}_g) Y_l^m (\hat{\mathbf{W}}) Y_{l',m'} (\hat{\mathbf{W}}) y_{l',m'}^+ dW \quad \begin{matrix} l = 1, 3, 5, \dots, N & |m| = 0, 1, 2, \dots, l \\ l' = 0, 2, 4, \dots, N-1 & |m'| = 0, 1, 2, \dots, l' \end{matrix} \quad (129)$$

$$\oint (\hat{\mathbf{W}} \times \hat{\mathbf{n}}_g) Y_l^m (\hat{\mathbf{W}}) Y_{l',m'} (\hat{\mathbf{W}}) y_{l',m'}^- dW \quad \begin{matrix} l = 0, 2, 4, \dots, N-1 & |m| = 0, 1, 2, \dots, l \\ l' = 1, 3, 5, \dots, N & |m'| = 0, 1, 2, \dots, l' \end{matrix} \quad (130)$$

Upon inspection of equations 129 and 130, the matrix defined in equation 131 is found to exist for any arbitrary surface $\hat{\mathbf{n}}_g$.

$$E_{l',m'}^{l,m} = \oint (\hat{\mathbf{W}} \times \hat{\mathbf{n}}_g) Y_l^m (\hat{\mathbf{W}}) Y_{l',m'} (\hat{\mathbf{W}}) dW \quad \begin{matrix} l = 0, 2, 4, \dots, N-1 & |m| = 0, 1, 2, \dots, l \\ l' = 1, 3, 5, \dots, N & |m'| = 0, 1, 2, \dots, l' \end{matrix} \quad (131)$$

In terms of the spherical harmonics expansion defined by equations 127 and 128, the Rumyantsev boundary conditions can be interpreted as a requirement that $\bar{E}^T y^+$ and $\bar{E} y^-$ be continuous across node interfaces. From the application of the variation in Section 2, equation 29, it is known that the $\bar{E} y^-$ condition is satisfied. The $\bar{E}^T y^+$ condition is also satisfied because of the special even parity interface functions chosen in Section 2, equation 53. The use of a different set of even parity angular interface trial functions for the projection, such as $g(\hat{\mathbf{W}})$, would not meet the continuity requirements outlined by equation 131.

With a spherical harmonics expansion, the \bar{E} matrix of equation 104 (or equation 131) is not a square matrix; there are clearly more odd parity expansion terms than even parity for a selected order N (N is an odd number). For the response matrix formulation derived in section 2, this matrix must be square, which means that either some odd parity functions must be eliminated or some even parity functions must be added. It has been shown [1-4,7,14] that some of the odd parity trial functions in an N order expansion are linearly dependent in the projection of \bar{E} and thus they can be eliminated. The method currently implemented in VARIANT is to remove the $Y_l^{\pm 1}$ terms from the odd parity expansion (l is odd) and the required square \bar{E}

matrix is obtained. The resulting angular trial functions for two-dimensional problems are found to be

$$g(\hat{W}) = \{Y_{0,0}^c, Y_{2,2}^c, Y_{2,1}^c, Y_{2,0}^c, Y_{4,4}^c, Y_{4,3}^c, Y_{4,2}^c, \dots\} \text{ and} \quad (132)$$

$$k_g(\hat{W}) = \{Y_{1,0}^c, Y_{3,2}^c, Y_{3,1}^c, Y_{3,0}^c, Y_{5,4}^c, Y_{5,3}^c, Y_{5,2}^c, \dots\}. \quad (133)$$

For three-dimensional problems the angular trial functions are found to be

$$g(\hat{W}) = \{Y_{0,0}^c, Y_{2,-2}^s, Y_{2,-1}^s, Y_{2,0}^c, Y_{2,2}^c, Y_{2,1}^c, Y_{4,-4}^s, Y_{4,-3}^s, Y_{4,-2}^s, \dots\} \text{ and} \quad (134)$$

$$k_g(\hat{W}) = \{Y_{1,0}^c, Y_{3,-2}^s, Y_{3,-1}^s, Y_{3,0}^c, Y_{3,2}^c, Y_{3,1}^c, Y_{5,-4}^s, Y_{5,-3}^s, Y_{5,-2}^s, \dots\}. \quad (135)$$

Note that the g index is included to denote that the $k_g(\hat{W})$ trial functions must be rotated to align the polar angle with the outward normal.

With these definitions of the functions $g(\hat{W})$ and $k_g(\hat{W})$, the angular matrices derived during the response matrix formulation can either be evaluated analytically or numerically. For the matrices in equations 100-104, equation 102 reduces to

$$I_{m,m'} = d_{m,m'}. \quad (136)$$

Equation 103 has a similar result since the spherical harmonic set used for the anisotropic scattering is a subset of the even parity flux moments. The matrices in equations 100 and 101 can be obtained analytically. The matrices in equation 104 require the surface functions to be rotated, which makes the analytical integration exceedingly difficult, especially for hexagonal geometries. Note that we could also employ the generic rotation matrices, but we have found this to be quite difficult in practice because of the non-standard orientations applied to the surface functions in VARIANT. Given this problem, exact numerical integration techniques (within the precision of the machine) were employed to perform the angular integrals in equations 100, 101, and 104. The details of this numerical integration and its implementation into VARIANT are given in Section 3.5.

3.3 Boundary conditions for the response matrix method

With the angular trial functions defined, the new angular boundary conditions must be explored. For this analysis a three-dimensional Cartesian nodal geometry, as seen in Figure 3.1, will be used. The two types of boundary conditions that will be applied in this work are the reflective and vacuum boundary conditions. To reduce the complexity of the boundary condition expressions, the direction cosines defined by equation 137, shown in Figure 3.2, are employed.

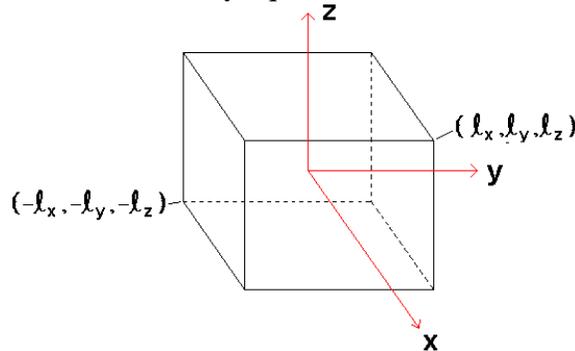


Figure 3.1. A Three-Dimensional Cartesian Node

$$\hat{W} = m\hat{i} + h\hat{j} + z\hat{k} = \cos(q)\hat{i} + \cos(f)\sin(q)\hat{j} + \sin(f)\sin(q)\hat{k} \quad (137)$$

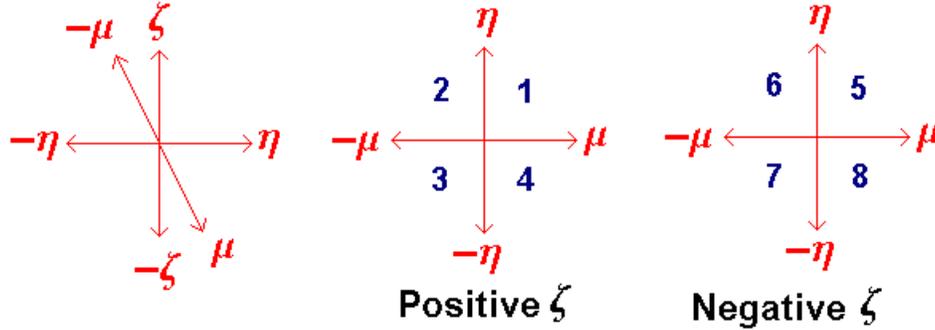


Figure 3.2. Direction Cosines and Numbered Octants

First the reflective boundary condition given by equation 138 is considered. Equation 138 can be graphically represented as seen in Figure 3.3, where \hat{W} represents the direction of reflection corresponding to the incident angle \hat{W} .

$$\begin{aligned} y_g(\vec{r}, \hat{W}) &= y_g(\vec{r}, \hat{W}) \\ y_g(\vec{r}, m\hat{h}, z) &= y_g(\vec{r}, m, h', z') \end{aligned} \quad \vec{r} \hat{=} G \quad (138)$$

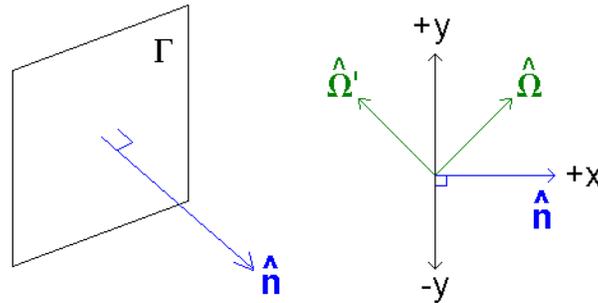


Figure 3.3. Simplified 1-D Diagram of the Reflected Boundary Condition

In Figure 3.3, the positive x side of the problem domain is used as an example for which the reflected boundary condition is given by

$$y_g(\vec{r}, m\hat{h}, z) = y_g(\vec{r}, -m\hat{h}, z). \quad (139)$$

Expanding the angular flux in equation 139 into even and odd parity flux components leads to equation 140.

$$y_g^+(\vec{r}, m\hat{h}, z) + y_g^-(\vec{r}, m\hat{h}, z) = y_g^+(\vec{r}, -m\hat{h}, z) + y_g^-(\vec{r}, -m\hat{h}, z) \quad (140)$$

To develop unique expressions for the even and odd parity components, equation 140 is evaluated at $-\hat{W}$ to get equation 141.

$$y_g^+(\vec{r}, -m\hat{h}, -z) + y_g^-(\vec{r}, -m\hat{h}, -z) = y_g^+(\vec{r}, m\hat{h}, -z) + y_g^-(\vec{r}, m\hat{h}, -z) \quad (141)$$

Equations 140 and 141 are then added and subtracted to yield equations 142 and 143 in standard form (left) and octant form (right).

$$\begin{aligned}
y_g^+(r, m, h, z) &= y_g^+(r, -m, h, z) \\
&= y_g^+(r, -m, h, z) y_g^+ \{1\} = y_g^+ \{2\} \\
&= y_g^+(r, m, h, z) = y_g^+ \{7\} \\
&= y_g^+(r, q, f + p) = y_g^+ \{8\} \\
&= y_g^+(r, q, f)
\end{aligned}
\tag{142}$$

$$\begin{aligned}
y_g^-(r, m, h, z) &= -y_g^-(r, -m, h, z) \\
&= -y_g^-(r, -m, h, z) y_g^- \{1\} = -y_g^- \{2\} \\
&= -y_g^-(r, m, h, z) = -y_g^- \{7\} \\
&= -y_g^-(r, q, f + p) = -y_g^- \{8\} \\
&= -y_g^-(r, q, f)
\end{aligned}
\tag{143}$$

These relations can be implemented into the nodal functional and one finds that only the boundary term is affected, the result of which is given in equation 144.

$$\begin{aligned}
&\int_0^{2p} dG \int_0^p df \int_0^{\hat{V}} \sin(q) dq \hat{V}_g y_g^+(r, q, f) y_g^-(r, q, f) \dot{U} \\
&- \int_0^{2p} dG \int_0^p df \int_0^{\hat{V}} \sin(q) dq \hat{V}_g y_g^+(r, q, f + p) y_g^-(r, q, f + p) \dot{U}
\end{aligned}
\tag{144}$$

Defining the variable $f' = f + p$, equation 144 can be simplified to equation 145, which can only be satisfied if the boundary term is forced to zero.

$$\begin{aligned}
&\int_0^{2p} dG \int_0^p df \int_0^{\hat{V}} \sin(q) dq \hat{V}_g y_g^+(r, q, f) y_g^-(r, q, f) \dot{U} \\
&- \int_0^{2p} dG \int_0^p df' \int_0^{\hat{V}} \sin(q) dq \hat{V}_g y_g^+(r, q, f') y_g^-(r, q, f') \dot{U} = 0
\end{aligned}
\tag{145}$$

It has been shown in earlier work [1-4,7] that in order to satisfy the above restriction the expansion coefficients for even parity trial functions $p_g(\hat{V})$ with odd order m and the expansion coefficients for odd parity trial functions $k_g(\hat{V})$ with even order m must be forced to zero.

Although this can be shown mathematically, a visual analysis based on the octant form of equations 146 and 147 can more easily clarify why these terms must be forced to zero. For this visual analysis a two-dimensional Cartesian P_3 approximation is used as an example. The P_3 angular trial functions for surface 1 are given in equations 146 and 147 and plotted in Figures 3.4 and 3.5 for two different perspectives in 2π geometry ($0 \leq q \leq p$ and $0 \leq f \leq p$).

$$p(\hat{W}) = \sum_{l=0}^{\infty} \sum_{m=-l}^l a_{l,m}^e Y_{l,m}^e(\hat{W}) \quad g=1 \quad (146)$$

$$k(\hat{W}) = \sum_{l=0}^{\infty} \sum_{m=-l}^l b_{l,m}^e Y_{l,m}^e(\hat{W}) \quad g=1 \quad (147)$$

From the boundary condition in equation 142 one can see that only the first, second, and fourth even parity P_3 functions obey the reflected boundary condition. The coefficient for the third even parity P_3 function, corresponding to $a_{3,1}^e(\hat{W})$, must therefore be forced to zero. In the same fashion, inspection of the higher order angular trial functions shows that only the even m even parity trial functions match the reflective boundary condition specified by equation 142.

A similar behavior can be observed for the odd parity P_3 functions shown in Figure 3.5. From the boundary condition described by equation 143 one can see that only the third odd parity P_3 function, corresponding to $b_{3,1}^e(\hat{W})$, obeys the reflected boundary condition. Therefore, the first, second, and fourth odd parity expansion coefficients must all be forced to zero to match the reflected boundary condition. Similar to the even parity trial functions, further inspection of higher order angular trial functions shows that only the odd m odd parity trial functions obey the reflective boundary condition specified in equation 143.

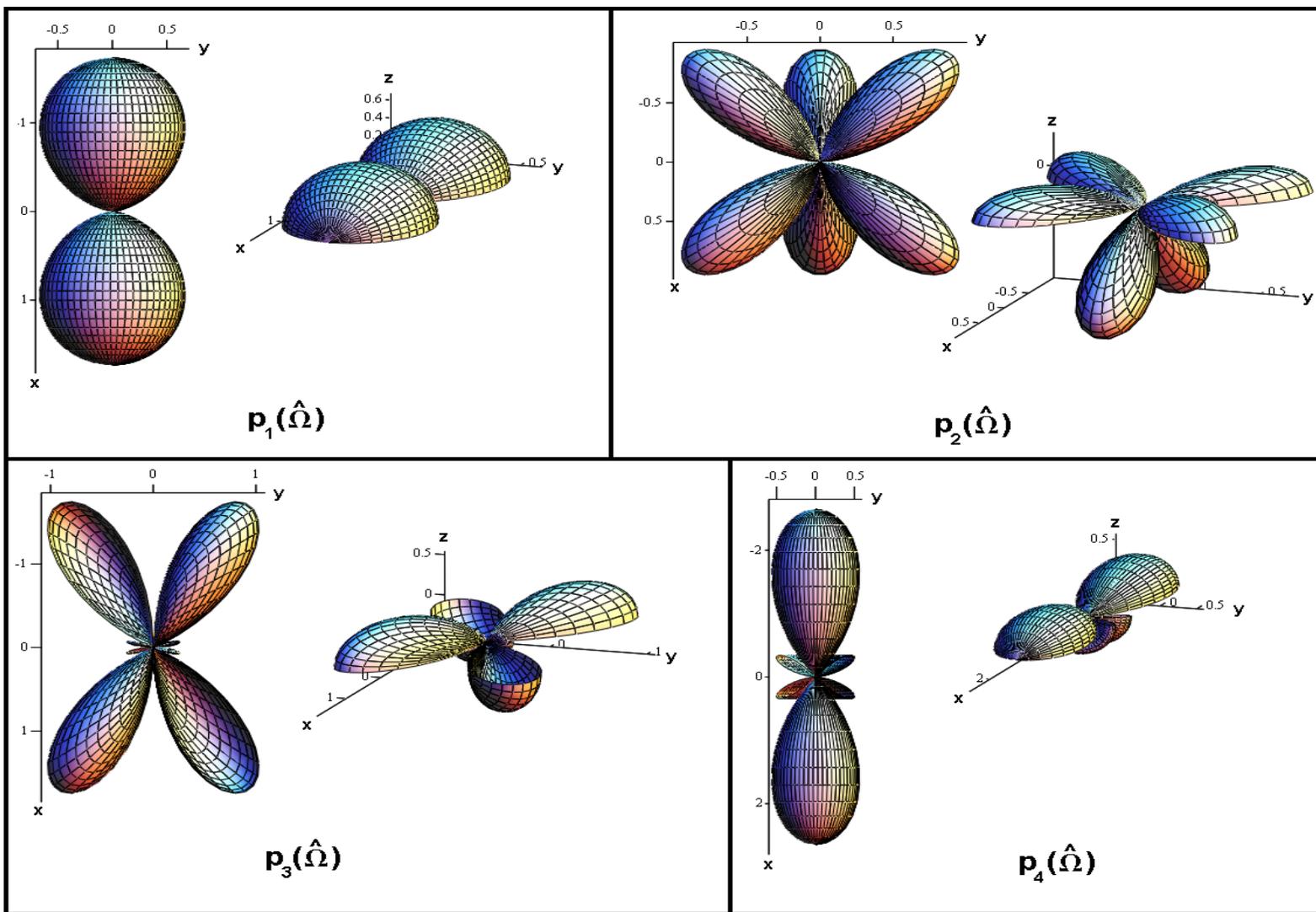
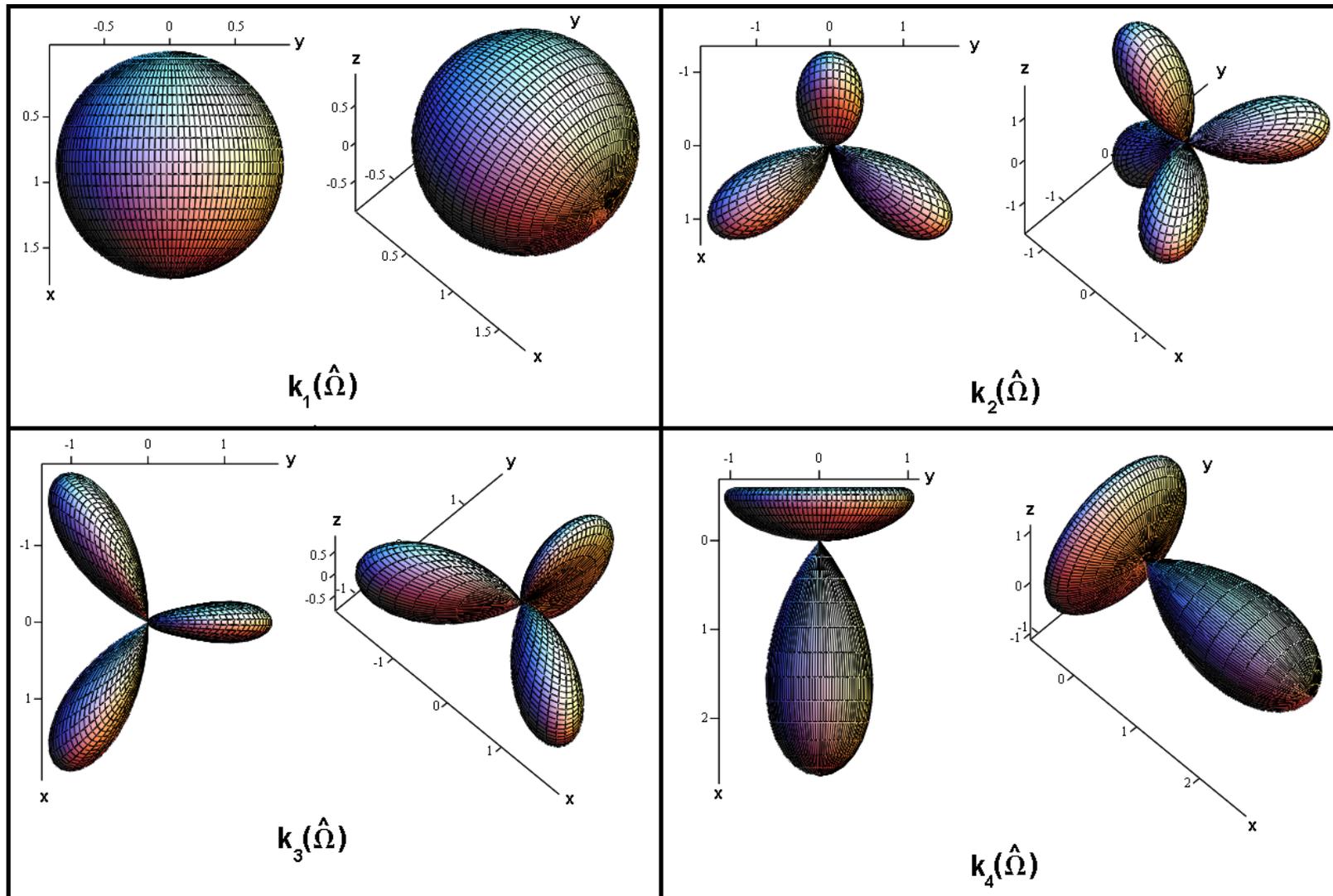


Figure 3.4. Even Parity Interface Trial Functions $p(\hat{\Omega})$

Figure 3.5. Odd Parity Interface Trial Functions $k(\hat{\Omega})$

equations 152 and 153. Both equation 152 and 153 must be obeyed in order to satisfy the vacuum boundary condition.

$$\hat{W}\hat{\chi}^{\dagger}y_g^+(r,\hat{W})-S_{t,g}(r)y_g^+(r,\hat{W})=0 \quad r \hat{=} G, \quad \hat{W}\hat{\chi} < 0 \quad (152)$$

$$\hat{W}\hat{\chi}^{\dagger}y_g^+(r,\hat{W})+S_{t,g}(r)y_g^+(r,\hat{W})=0 \quad r \hat{=} G, \quad \hat{W}\hat{\chi} > 0 \quad (153)$$

The vacuum boundary condition is imposed as a natural boundary condition by altering the formulation of the boundary term $\delta \int \hat{W}\hat{\chi}^{\dagger}y_g^+(r,\hat{W})y_g^-(r,\hat{W})dWdG$ in the functional. Using equations 152 and 153, the new boundary term shown in equation 154 can be obtained.

$$\delta \int \hat{W}\hat{\chi}^{\dagger}y_g^+(r,\hat{W})y_g^-(r,\hat{W})dWdG \rightarrow \delta \int \hat{W}\hat{\chi}^{\dagger}y_g^+(r,\hat{W})y_g^-(r,\hat{W})dWdG \quad (154)$$

To implement the vacuum boundary condition, the present boundary term in the functional in equation 19 must be replaced by the new boundary term in equation 154. A direct substitution will lead to response matrices that are dependent upon the boundary conditions, which is undesirable since this can increase the number of unique nodes that have to be defined (this can significantly impact the time required to obtain the response matrices). To obviate this problem, a new global even parity function $y_g^+(r,\hat{W})$ is defined along the boundary of the problem domain as seen in equation 155.

$$F \int \hat{W}\hat{\chi}^{\dagger}y_g^+(r,\hat{W})y_g^-(r,\hat{W})dWdG \rightarrow F \int \hat{W}\hat{\chi}^{\dagger}y_g^+(r,\hat{W})c_g^-(r,\hat{W})dWdG - 2 \delta \int \hat{W}\hat{\chi}^{\dagger}y_g^+(r,\hat{W})c_g^-(r,\hat{W})dWdG + \delta \int \hat{W}\hat{\chi}^{\dagger}y_g^+(r,\hat{W})y_g^+(r,\hat{W})dWdG \quad (155)$$

The variation of this new functional with respect to $y_g^+(r,\hat{W})$ requires that the following constraint be obeyed along the vacuum boundary.

$$- \delta \int \hat{W}\hat{\chi}^{\dagger}c_g^-(r,\hat{W})dWdG + \delta \int \hat{W}\hat{\chi}^{\dagger}y_g^+(r,\hat{W})y_g^+(r,\hat{W})dWdG = 0 \quad (156)$$

Since the boundary condition is not defined to be spatially dependent along the selected interface, the spatial variation can be ignored given that each spatial expansion coefficient is treated identically. Also, since the even and odd parity interface trial functions are rotated at each interface, the problem can further be reduced to a one-dimensional analysis ($g = 1$).

The angular expansions for the nodal even and odd parity functions in equation 156 for a selected spatial coefficient take the following forms.

$$y_g^+(\hat{W}) = g_m(\hat{W})z_{g,m} \quad (157)$$

$$c_g^-(\hat{W}) = k_n(\hat{W})c_{g,n} \quad g = 1 \quad (158)$$

Substituting these relations into equation 155 and taking the variation with respect to the angular expansion coefficients leads to equation 159.

$$\delta \int \hat{W}\hat{\chi}^{\dagger}g_m(\hat{W})k_n(\hat{W})c_{g,n}dW = \delta \int \hat{W}\hat{\chi}^{\dagger}g_m(\hat{W})g_q(\hat{W})z_{g,q}dW \quad (159)$$

Equation 159 can be expressed in a matrix form, like that done for the reflected boundary condition, via the definition of the angular matrices in equations 160 and 161.

$$E_{m,n} = \hat{\mathbf{O}} \left(\hat{\mathbf{W}} \hat{\mathbf{x}} \hat{\mathbf{n}} \right) g_m(\hat{\mathbf{W}}) k_n(\hat{\mathbf{W}}) d\mathbf{W} \quad (160)$$

$$L_{m,q} = \hat{\mathbf{O}} \left| \hat{\mathbf{W}} \hat{\mathbf{x}} \hat{\mathbf{n}} \right| g_m(\hat{\mathbf{W}}) g_q(\hat{\mathbf{W}}) d\mathbf{W} \quad (161)$$

Substitution of equations 160 and 161 into equation 159 yields equation 162.

$$\bar{\mathbf{E}} \mathbf{C}_g = \bar{\mathbf{L}} \hat{\mathbf{Z}}_g \quad (162)$$

Solving equation 162 for $\hat{\mathbf{Z}}_g$ gives

$$\hat{\mathbf{Z}}_g = \bar{\mathbf{L}}^{-1} \bar{\mathbf{E}} \mathbf{C}_g \quad (163)$$

At this point the even parity angular flux can be projected to the node boundary (as done earlier in Section 2, equations 49-57) such that the following relationship is obtained.

$$\hat{\mathbf{Y}}_g = \bar{\mathbf{E}}^T \bar{\mathbf{L}}^{-1} \bar{\mathbf{E}} \mathbf{C}_g \quad (164)$$

Using the definitions of the partial currents from equations 59 and 60 in Section 2, equation 164 can be rearranged to obtain the following final form of the vacuum boundary condition for any specified node surface.

$$\hat{\mathbf{j}}_g = \left(\frac{1}{2} \bar{\mathbf{E}}^T \bar{\mathbf{L}}^{-1} \bar{\mathbf{E}} + \bar{\mathbf{I}} \right)^{-1} \left(\frac{1}{2} \bar{\mathbf{E}}^T \bar{\mathbf{L}}^{-1} \bar{\mathbf{E}} - \bar{\mathbf{I}} \right) \hat{\mathbf{j}}_g^+ \quad (165)$$

Equation 165 has the same form as that seen in equation 149, thus the application of the vacuum boundary condition simply requires the definition of a different \mathfrak{R}^i matrix.

3.4 Corrections to the angular approximation

The previous sections have thus far given only a brief review of the theory implemented into the present VARIANT code. What follows is the discussion on the corrections made to the angular approximation resulting from the recent work of Won Sik Yang in a paper on Romyantsev boundary conditions [14]. Although the details of that paper are omitted here (about 30-40 pages) the conclusions are given.

Before these new trial functions are given, a correction to the notation used up to this point for the spherical harmonics must be made. Although a $-m$ is valid for the spherical harmonics defined in equation 119, it doesn't apply to the even and odd parity series defined by equations 122 and 123. Instead, in this report and that of the VARIANT manual, m has been used to denote the transition from cosine of the azimuthal angle (ϕ) to sine of the azimuthal angle. The correct notation for the even and odd parity functions utilized in VARIANT is given below. Note that they do not follow the standard convention, but instead rely upon a linear combination of $|L|$ and $|M|$ functions.

$$Y_{1,m}^e(\hat{\mathbf{W}}) = \begin{cases} \sqrt{C_1^{|m|}} \mathbf{x} P_1^{|m|}(m) \mathbf{x} \cos(|m|f) & m \geq 0 \\ G_1^m \sqrt{C_1^{|m|}} \mathbf{x} P_1^{|m|}(m) \mathbf{x} \sin(|m|f) & m < 0 \end{cases} \quad (166)$$

$$Y_{1,m}^o(\hat{\mathbf{W}}) = \begin{cases} \sqrt{C_1^{|m|}} \mathbf{x} P_1^{|m|}(m) \mathbf{x} \cos(|m|f) & m \geq 0 \\ G_1^m \sqrt{C_1^{|m|}} \mathbf{x} P_1^{|m|}(m) \mathbf{x} \sin(|m|f) & m < 0 \end{cases} \quad (167)$$

$$G_l^m = (-1)^{|m|} \frac{(l-|m|)!}{(l+|m|)!} \quad (168)$$

$$C_l^m = \frac{(2l+1)(l-m)!}{(l+m)!} \quad (169)$$

The factor G_l^m comes from the Legendre polynomial relation.

The work of Yang has shown that the set of odd parity angular trial functions $k(\hat{\Omega})$ chosen in equations 133 and 135, reproduced as equations 170 and 171 below, do not satisfy the Romyantsev boundary conditions. Consequently, the functions $p(\hat{W})$ defined in equation 53 also do not satisfy the Romyantsev boundary conditions.

$$k_g(\hat{W}) = \{Y_{1,0}^e, Y_{3,2}^e, Y_{3,1}^e, Y_{3,0}^e, Y_{5,4}^e, Y_{5,3}^e, Y_{5,2}^e, \dots\} \quad (170)$$

$$k_g(\hat{W}) = \{Y_{1,0}^e, Y_{3,-2}^o, Y_{3,-1}^o, Y_{3,0}^e, Y_{3,2}^e, Y_{3,1}^e, Y_{5,-4}^o, Y_{5,-3}^o, Y_{5,-2}^o, \dots\} \quad (171)$$

Two alternate angular trial function sets were derived by Yang and both were included in the VARIANT coding. However, only the second is feasible with the current acceleration scheme and it is therefore the only one considered here. Equation 172 gives the two-dimensional set of even parity functions while equation 173 gives the three-dimensional set derived by Yang ($\pm m$ denote cosine and sine series as shown earlier).

$$k_g(\hat{W}) = \begin{cases} Y_{l,m}^c(\hat{W}) & l = 1,3,5,K, N; \quad m = 0,2,4,K, l \\ a_{l,m} Y_{l,m}^c(\hat{W}) + a_{l+1,m} Y_{l+2,m}^c(\hat{W}) & l = 1,3,5,K, N-2; \quad m = 1,3,5,K, l \end{cases} \quad (172)$$

$$k_g(\hat{W}) = \begin{cases} Y_{l,m}(\hat{W}) & l = 1,3,5,K, N; \quad |m| = 0,2,4,K, l \\ a_{l,m} Y_{l,m}(\hat{W}) + a_{l+1,m} Y_{l+2,m}(\hat{W}) & l = 1,3,5,K, N-2; \quad |m| = 1,3,5,K, l \end{cases} \quad (173)$$

$$a_{l,m} = \sqrt{\frac{(l+m+1)(l-m+1)}{(2l+1)(2l+3)}} \quad (174)$$

Note that the odd m spherical harmonics are coupled together and no terms are arbitrarily removed. To help clarify the difference between the angular expansion, the two-dimensional P_3 sets of angular trial functions for the new and old approaches are given in equation 175.

$$k_g^{old}(\hat{W}) = \begin{pmatrix} Y_{1,0}^c(\hat{W}) \\ Y_{3,0}^c(\hat{W}) \\ Y_{3,1}^c(\hat{W}) \\ Y_{3,2}^c(\hat{W}) \end{pmatrix} \quad k_g^{new}(\hat{W}) = \begin{pmatrix} Y_{1,0}^c(\hat{W}) \\ Y_{3,0}^c(\hat{W}) \\ Y_{3,2}^c(\hat{W}) \\ \frac{3}{\sqrt{5}} Y_{1,1}^c(\hat{W}) + \frac{2\sqrt{14}}{\sqrt{5}} Y_{3,1}^c(\hat{W}) \end{pmatrix} \quad (175)$$

As can be seen the new approach does pose a significantly different set of angular trial functions and makes the integrations in the E_g matrix of equation 41 considerably more difficult. Fortunately, the new set of trial functions has no significant effects on the boundary conditions since the odd m terms are coupled to each other and the vacuum matrix treatment makes use of the E_1 matrix (compare equation 160 to equation 41).

3.5 The angular integrals

To begin, the angular integral is expressed in terms of the two angular variables as seen in equation 176.

$$\int_{4p} f(W) dW = \int_0^{2p} \int_0^p f(q, f) \sin(q) dq df \quad (176)$$

Since the angular functions used in this work are well-defined polynomials, the best way to numerically perform the integrals is with a Gaussian quadrature. To put equation 176 into a form compatible with Gaussian quadrature integration schemes, the following variable transformations are made.

$$\begin{aligned} m &= \cos(q) & q=0 & m=1 \\ dm &= -\sin(q) dq & q=p & m=-1 \end{aligned} \quad (177)$$

$$\begin{aligned} h &= \cos(f) & f=0 & h=1 \\ dh &= -\sin(f) df & f=p & h=-1 \\ &= -\sqrt{1-\cos^2(f)} df & f=2p & h=1 \\ &= -\sqrt{1-h^2} df \end{aligned} \quad (178)$$

Substituting these transformations into equation 176 and splitting the azimuthal angle into two hemispheres yields equation 179.

$$\begin{aligned} \int_0^{2p} \int_0^p f(q, f) \sin(q) dq df &= \\ \int_{-1}^{-1} \int_{-1}^{-1} f(m, h) \times \frac{1}{\sqrt{1-h^2}} dm dh &+ \int_{-1}^{-1} \int_{-1}^{-1} f(m, h) \times \frac{1}{\sqrt{1-h^2}} dm dh = \\ \int_{-1}^{-1} \int_{-1}^{-1} f(m, h) \times \frac{1}{\sqrt{1-h^2}} dm dh \Bigg|_{\text{upper}} &+ \int_{-1}^{-1} \int_{-1}^{-1} f(m, h) \times \frac{1}{\sqrt{1-h^2}} dm dh \Bigg|_{\text{lower}} \end{aligned} \quad (179)$$

The angular integrations in equation 179 can be handled with the Gauss-Legendre and Gauss-Tchebychev numerical integration treatments defined by equations 180 and 181.

$$\int_{-1}^{-1} f(m) dm = \int_L^L f(m) w_l \quad (180)$$

$$\int_{-1}^{-1} f(h) \frac{1}{\sqrt{1-h^2}} dh = \int_c^c f(h_c) w_c \quad (181)$$

The details of how to obtain the weights and abscissas for the Gaussian quadrature can be found elsewhere [16,17]. The tensor of the two one-dimensional integration schemes produces a set of weights and abscissas that exactly integrates (within the precision of the machine) functions of L order in μ and C order in h .

It is important to note that other numerical integration techniques can be used (such as level-symmetric, equal weight, etc...) the details of which can be found in the literature [17]. In this work, the Legendre-Tchebychev approach is preferable because it guarantees integration of the spherical harmonics and it can easily be extended to very high order N . This is especially true upon the rotation of the angular trial functions where the lack of angular matrix symmetry caused by insufficient integration can cause problems in the inner iteration routine of the VARIANT code.

Application of equations 180 and 181 to the angular integral in equation 179 is difficult because of the use of upper and lower hemisphere. To eliminate this problem with notation, the direction cosines specified in equation 182 are used.

$$\begin{aligned} m &= \cos(q) \\ h &= \sin(q)\cos(f) \\ z &= \sin(q)\sin(f) \end{aligned} \tag{182}$$

Substitution of equations 180-182 into equation 179 gives the final form of the integral seen in equation 183.

$$\int_0^{2\pi} \int_0^{\pi} f(q, f) \sin(q) dq df = \int_P \hat{a} f(m_p, h_p, z_p) w_p \tag{183}$$

To distinguish the upper and lower hemispheres of integration, the z variable takes on positive (upper hemisphere) and negative (lower hemisphere) values. The variable P represents the total number of Legendre-Tchebychev points that are present in the quadrature. For a 4π domain $P=2LC$, for a 2π domain $P=LC$, and for a π domain $P=1/2LC$. This quadrature set is referred to as the ‘‘Square Legendre-Tchebychev’’ or SLC quadrature. Typically, even values of L and C are used, although there are no restrictions requiring even values.

Given the method to numerically perform the integrals, the evaluation of the spherical harmonics functions can be discussed. For this work, the spherical harmonic recursion relationships are used. Equation 184 gives the recursion relationship for the associated Legendre polynomials and equation 185 gives the negative M Legendre polynomial correspondence (defined earlier).

$$P(m)_{L+1,M} = \frac{2L+1}{L-M+1} (m \times P(m)_{L+1,M}) - \frac{L+M}{L-M+1} P(m)_{L-1,M} \tag{184}$$

$$P(m)_{L,-M} = (-1)^{|M|} \frac{(L-|M|)!}{(L+|M|)!} P(m)_{L,M} \tag{185}$$

Equation 186 gives the generating relationship for the associated Legendre polynomials.

$$P(m)_{L,M} = \frac{(-1)^M}{2^L L!} (1-m^2)^{\frac{M}{2}} \frac{d^{L+M}}{dm^{L+M}} (m^2-1)^L \tag{186}$$

Upon forcing $M=L$ in the generating relationship, one finds that the derivative term in equation 186 is a constant and equation 186 reduces to equation 187.

$$P(m)_{L,L} = \frac{(-1)^L}{2^L L!} (1-m^2)^{\frac{L}{2}} (2L)! \quad (187)$$

Using equations 184, 185, and 186 the associated Legendre polynomials can now be evaluated at μ for all possible combinations of L and M . In the VARIANT code the matrix in equation 188 is defined (two-dimensional P_3 example is shown). The P terms are obtained using equation 187 and the p terms are obtained using the recursion relations.

$$Pn2d = \begin{pmatrix} P_{0,0}(m) & 0 & 0 & 0 \\ P_{1,0}(m) & P_{1,1}(m) & 0 & 0 \\ P_{2,0}(m) & P_{2,1}(m) & P_{2,2}(m) & 0 \\ P_{3,0}(m) & P_{3,1}(m) & P_{3,2}(m) & P_{3,3}(m) \end{pmatrix} \quad (188)$$

For three-dimensional applications, the associated Legendre polynomials in equation 188 are duplicated and the negative M correspondence is used as shown in equation 189.

$$Pn3d = \begin{pmatrix} 0 & 0 & 0 \\ P_{1,-1}(m) & 0 & 0 \\ P_{2,-1}(m) & P_{2,-2}(m) & 0 \\ P_{3,-1}(m) & P_{3,-2}(m) & P_{3,-3}(m) \end{pmatrix} \quad (189)$$

To obtain the spherical harmonics, the matrices in equation 188 and 189 are modified to include the normalization factor and the azimuthal angle dependence. Only the modification to equation 189 is shown for brevity.

$$YLM3d = \begin{pmatrix} 0 & 0 & 0 \\ C_1^{-1} P_{1,-1}(m) \sin(f) & 0 & 0 \\ C_2^{-1} P_{2,-1}(m) \sin(f) & C_2^{-2} P_{2,-2}(m) \sin(2f) & 0 \\ C_3^{-1} P_{3,-1}(m) \sin(f) & C_3^{-2} P_{3,-2}(m) \sin(2f) & C_3^{-3} P_{3,-3}(m) \sin(3f) \end{pmatrix} \sqrt{2} \quad (190)$$

As can be seen, the recursion relationships do not make use of the direction cosines from equation 182 (azimuthal angle). This means that the SLC abscissas must be converted to the appropriate variables in order to make use of the recursion relationships, which is a trivial trigonometric operation.

The angular integrals to be performed for the response matrix method in VARIANT are reproduced in equations 191-196 for easy reference.

$$H_{m,n}^{k,l} = \int \hat{W}_k \hat{W}_l g_m(\hat{W}) g_n(\hat{W}) dW \quad (191)$$

$$V_{i,l}^k = \int \hat{W}_k g_i(\hat{W}) N_l^-(\hat{W}) dW \quad (192)$$

$$I_{m,n} = \int g_m(\hat{W}) g_n(\hat{W}) dW \quad (193)$$

$$J_{i,l} = \int g_i(\hat{W}) N_l^+(\hat{W}) dW \quad (194)$$

$$E_{m,n}^g = \oint (\hat{W} \hat{x}_g) g_m(\hat{W}) k_{g,n}(\hat{W}) dW \quad (195)$$

$$L_{m,q} = \oint |\hat{W} \hat{x}_q| g_m(\hat{W}) g_q(\hat{W}) dW \quad (196)$$

Performing the integrations in equations 193 and 194 are unnecessary since they both result in identity matrices. As it turns out, performing the integrations in equation 191 is also unnecessary since they can be obtained from equation 192 as outlined in equation 197, where the V matrices have been partitioned with respect to l .

$$H^{K,L} = V^K (V^L)^T \quad (197)$$

Obtaining the V matrices is straightforward, thus it will be dealt with first. The spherical harmonics are evaluated for each SLC abscissa as demonstrated in equation 190. The results of this matrix are reordered to form a matrix of spherical harmonic trial functions evaluated at each abscissa as shown in equations 198 and 199 (two-dimensional P_3 example shown).

$$EvenPn = \begin{matrix} \begin{matrix} Y_{0,0}(\hat{W}_1) & Y_{2,0}(\hat{W}_1) & Y_{2,1}(\hat{W}_1) & Y_{2,2}(\hat{W}_1) \\ Y_{0,0}(\hat{W}_2) & Y_{2,0}(\hat{W}_2) & Y_{2,1}(\hat{W}_2) & Y_{2,2}(\hat{W}_2) \\ \vdots & \vdots & \vdots & \vdots \\ Y_{0,0}(\hat{W}_P) & Y_{2,0}(\hat{W}_P) & Y_{2,1}(\hat{W}_P) & Y_{2,2}(\hat{W}_P) \end{matrix} \\ \begin{matrix} M & M & M & M \\ \vdots & \vdots & \vdots & \vdots \\ M & M & M & M \end{matrix} \end{matrix} \quad (198)$$

$$OddPn = \begin{matrix} \begin{matrix} Y_{1,0}(\hat{W}_1) & Y_{1,1}(\hat{W}_1) & Y_{3,0}(\hat{W}_1) & Y_{3,1}(\hat{W}_1) & Y_{3,2}(\hat{W}_1) & Y_{3,3}(\hat{W}_1) \\ Y_{1,0}(\hat{W}_2) & Y_{1,1}(\hat{W}_2) & Y_{3,0}(\hat{W}_2) & Y_{3,1}(\hat{W}_2) & Y_{3,2}(\hat{W}_2) & Y_{3,3}(\hat{W}_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Y_{1,0}(\hat{W}_P) & Y_{1,1}(\hat{W}_P) & Y_{3,0}(\hat{W}_P) & Y_{3,1}(\hat{W}_P) & Y_{3,2}(\hat{W}_P) & Y_{3,3}(\hat{W}_P) \end{matrix} \\ \begin{matrix} M & M & M & M & M & M \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ M & M & M & M & M & M \end{matrix} \end{matrix} \quad (199)$$

The final piece of the integral is Ω , evaluated at each of the SLC abscissa points as shown in equation 200. It is important to note that equation 200 is nothing but the SLC abscissa points and that no additional work must be done to obtain equation 200.

$$W = \begin{matrix} \begin{matrix} \eta_1 & h_1 & z_1 \\ \eta_2 & h_2 & z_2 \\ \vdots & \vdots & \vdots \\ \eta_p & h_p & z_p \end{matrix} \\ \begin{matrix} M & M & M \\ \vdots & \vdots & \vdots \\ M & M & M \end{matrix} \end{matrix} \quad (200)$$

To obtain the integrals of the spherical harmonic moments for the V matrix in equation 192, one must simply select the appropriate columns of equations 198, 199, and 200. The product shown in equation 201 is obtained for each SLC abscissa and the sum over all of the P abscissas gives the angular integral from equation 183.

$$V_{i,l,p}^k = \hat{W}_{k,p} g_i(m_p, h_p, z_p) N_l(m_p, h_p, z_p) w_p \quad (201)$$

The order of the SLC quadrature (L and C) necessary to exactly integrate the components of equation 192 is $N+1$, where N is the order of the spherical harmonics expansion. Also, because

of rotational symmetry, the SLC quadrature only needs to be defined over half of the angular domain: 2π for three-dimensional geometries and π for two-dimensional geometries.

For the E matrix integrations the odd parity interface functions must be rotated and reoriented. The reorientation is needed to satisfy geometrical symmetry conditions, the details of which will be given shortly. The angular rotations for the Cartesian and hexagonal geometries are given in Table 3.1 and displayed in Figure 3.6.

Table 3.1. Angular Rotations for Spherical Harmonic Trial Functions

	μ'	η'	ζ'
Cartesian Surface 1	μ	η	ζ
Cartesian Surface 2	η	μ	ζ
Cartesian Surface 3	ζ	η	μ
Hexagonal Surface 1	μ	η	ζ
Hexagonal Surface 2	$\frac{1}{2}\mu + \frac{\sqrt{3}}{2}\eta$	$\frac{\sqrt{3}}{2}\mu - \frac{1}{2}\eta$	ζ
Hexagonal Surface 3	$-\frac{1}{2}\mu + \frac{\sqrt{3}}{2}\eta$	$\frac{\sqrt{3}}{2}\mu + \frac{1}{2}\eta$	ζ
Hexagonal Surface 5	ζ	η	μ

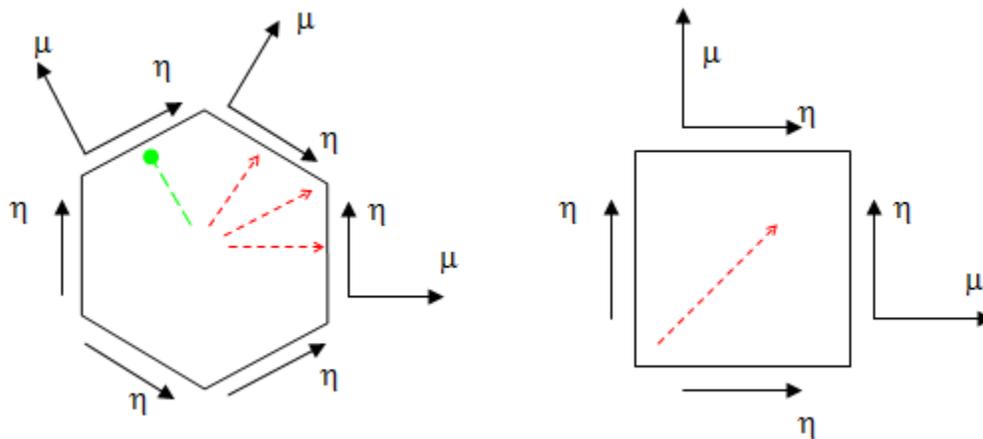


Figure 3.6. Cartesian and Hexagonal Angular Surface Function Orientations.

The most important part of Table 3.1 and Figure 3.6 to recognize is that the “right hand rule” is not followed for all surfaces. The reason for the reorientation is to allow the implementation of 30 and 45 degree geometrical symmetry. In the Cartesian and hexagonal geometries of Figure 3.6, the dashed arrows indicate the lines of symmetry that are viable using direct current matching. The dashed circle-head line indicates a line of symmetry that is not viable with direct current mapping. In short, the direction of the dashed circle-head line fails to result in a symmetrical layout of the current moments (the η variable must be aligned symmetrically on opposite surfaces of the symmetry line). More detail will be given in a later section as this proved to be a formidable bug in the existing coding.

The application of the angular rotations in the E matrix integrations makes the analytical integration difficult since they require variable transformations like $q' = q + 90$ and

$f' = -f - 30$. Fortunately, the numerical integration approach makes the rotation process almost trivial to implement. For the E matrix integrations, the internal even parity trial functions are evaluated the same way in which they were done for the V matrices. The odd parity spherical harmonics along each surface are evaluated using SLC abscissas that are transformed according to the surface angular transformation in Table 3.1. The end result is that an equation of the form seen in equation 199 is obtained for each unique surface of each node. The term $\hat{W}\hat{x}\hat{y}_g$ is also obtained using the SLC abscissas under the transformation. The integrations then follow the same procedure as that done for the V matrix integrations. As was the case for the V matrix integrations, only an $N+1$ order SLC quadrature (L and C) is necessary to exactly integrate the E matrix components. As was the case with the V matrix integrations, the SLC quadrature only needs to be defined over half of the angular domain.

At this point the E matrices are defined using all of the odd parity spherical harmonic moments. To get the E matrices into a square form, as required in the response matrix equations of Section 2, either the old method of moment reduction or the new method proposed by Yang must be applied. In this analysis only the old method is shown for brevity. To carry out the moment reduction, the matrix multiplication shown in equation 202 is applied where the Z matrix is defined in equation 203 for a two-dimensional P_3 expansion according to the moment ordering of equation 199.

$$E'_g = E_g Z \tag{202}$$

$$Z = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \tag{203}$$

The same treatment applies to the Yang approach but in this form the odd m moments are coupled together. The Z matrix is also used to reduce the \mathfrak{R}^i matrix for the reflected boundary condition to one appropriate for the reduced set of angular trial functions.

The last integration to perform is the L matrix integration shown in equation 196. What makes this integration unique is the absolute value of the $|\hat{W}\hat{x}\hat{y}_g|$ term. The standard SLC quadrature used up to this point fails because the μ variable is defined over the entire domain $\{-1,1\}$ and thus smooth polynomials are assumed over that domain. With the absolute value, discontinuities in the slope of the angular function to be integrated exist at 0, thus a double Legendre approximation must be made. In this approximation, smooth Legendre polynomial approximations are made over the half ranges of μ : $\{-1,0\}$ and $\{0,1\}$. This effectively doubles the number of points used in the μ direction. The name of this quadrature set is SDLC and an $N+1$ order expansion (L and C) is needed to exactly evaluate the integrals in equation 196. More detail on the SDLC approach can be found in the literature under the general topic of double Legendre approximations in slab geometry [15,17]. Given that the only change is in the

quadrature used, the mechanics of the integration process are the same as those used in the V and E matrices.

This concludes the new work done on the angular approximation in the VARIANT coding. The above methodology was put into Fortran subroutines using standard Fortran 90 coding to eliminate problems with Fortran 77 memory allocation routines. In the production version of VARIANT, the angular matrices were hardwired at different points in the coding making a single subroutine call to obtain all of the new angular matrices almost impossible. The solution was to write unique subroutines to obtain only the matrices needed at the various points in the VARIANT coding.

The DIF3D input structure also had to be changed to allow the new angular approximations to be used, which affects the third input descriptor on the card type 12 of DIF3D input and the NAPRXA variable in the binary DIF3D file. The major changes to the DIF3D input are shown in Table 3.2 and the modified input and binary descriptions for DIF3D are given in Appendix A. For the new description, two angular trial function sets are possible. The first is the default angular trial function set used in VARIANT since 1992: $H=1$. The second were given by Yang earlier in this section: $H=3$ (30505 for the example in Table 3.2).

Table 3.2. Changes to the DIF3D Input Description on Card 12.

Input Field	$\pm MN$	$\pm HMMNN$ or $\pm MMNN$ or $\pm MN$
\pm	$P_N (+)$ or $SP_N (-)$	$P_N (+)$ or $SP_N (-)$
H	—	Angular Trial Function Set
M	Order of the P_N Flux Expansion	Order of the P_N Flux Expansion
N	Order of the P_N Leakage Expansion	Order of the P_N Leakage Expansion
P_5 example	55	10505 or 0505 or 55
SP_5 example	-55	-10505 or -0505 or -55

Other changes include the removal of the reduced order leakage approximation $MN=31$ and the restriction of a maximum P_5 anisotropic scattering. It was found that the theory behind the reduced order approximation was based upon an older set of angular trial functions (Legendre polynomial expansions) rather than the full spherical harmonics [4]. Recent work by Smith [9] indicates that there are two valid approaches to reducing the angular trial functions on the interface and reduced order options may be reintroduced in the future, but for now they are omitted. As for the limitation on anisotropic scattering, the inclusion of higher order scattering required a significant rework of the cross section setup within DIF3D which was facilitated by using Fortran 90 or just hardwiring the various array dimensions to order P_{101} which are generally beyond the practical limits of existing computing technology.

4 The Spatial Approximations Implemented in VARIANT

4.1 Introduction

With the angular approximation detailed, now the spatial approximation can be displayed. Although this approximation is not original work, it is necessary to clarify the changes made to the production version of VARIANT. The procedure used to obtain the higher order spatial trial functions and the method used to obtain the spatial matrices defined in Section 2 is discussed. Again, the main focus of this section is the removal of data statements from the VARIANT coding.

4.2 The RITZ approximation

The classical Ritz procedure employs orthonormal spatial trial functions. The method used to obtain those functions is the Gram-Schmidt procedure outlined by Figure 4.1. In the production version of VARIANT this procedure was carried out analytically using Mathematica. In Fortran, an analytical approach cannot be used forcing a monomial based approach to be pursued instead. In this approach the analytic integration of an arbitrary monomial ($x^n y^m z^o$) is known over the integration range of the node (volume or surface integration), where the exponents n , m , and o are stored. The procedure of obtaining the orthonormal functions using this approach is outlined in Figure 4.2 and reference 4.

In Figure 4.2, M_i^j represents the j th vector of monomial integrands and f^i represents the monomial coefficients for the i th orthonormal function (a vector). Upon comparison of Figure 4.1 to Figure 4.2, one finds that the major difference is the removal of the integration in step 2 with a vector multiplication. What should be immediately apparent is that the vectors M and f can be partitioned into full matrices, where the M matrix is a sparse symmetric matrix and the f matrix is an upper triangular matrix. The partitioning helps to simplify the storage scheme and speed up the computation of the orthonormal functions.

Given this process to obtain orthonormal functions, the resulting set of functions is dependent upon the ordering of the monomials, which can be setup in any number of ways in Fortran. Consequently, the set of orthonormal functions used in the production version are not identical to the set used here (an arbitrary ordering of the trial functions was implemented for the production form) and the matrices that resulted were not the same for three-dimensional geometries. No impact on the accuracy of the results could be distinguished by using the alternate set of orthonormal functions.

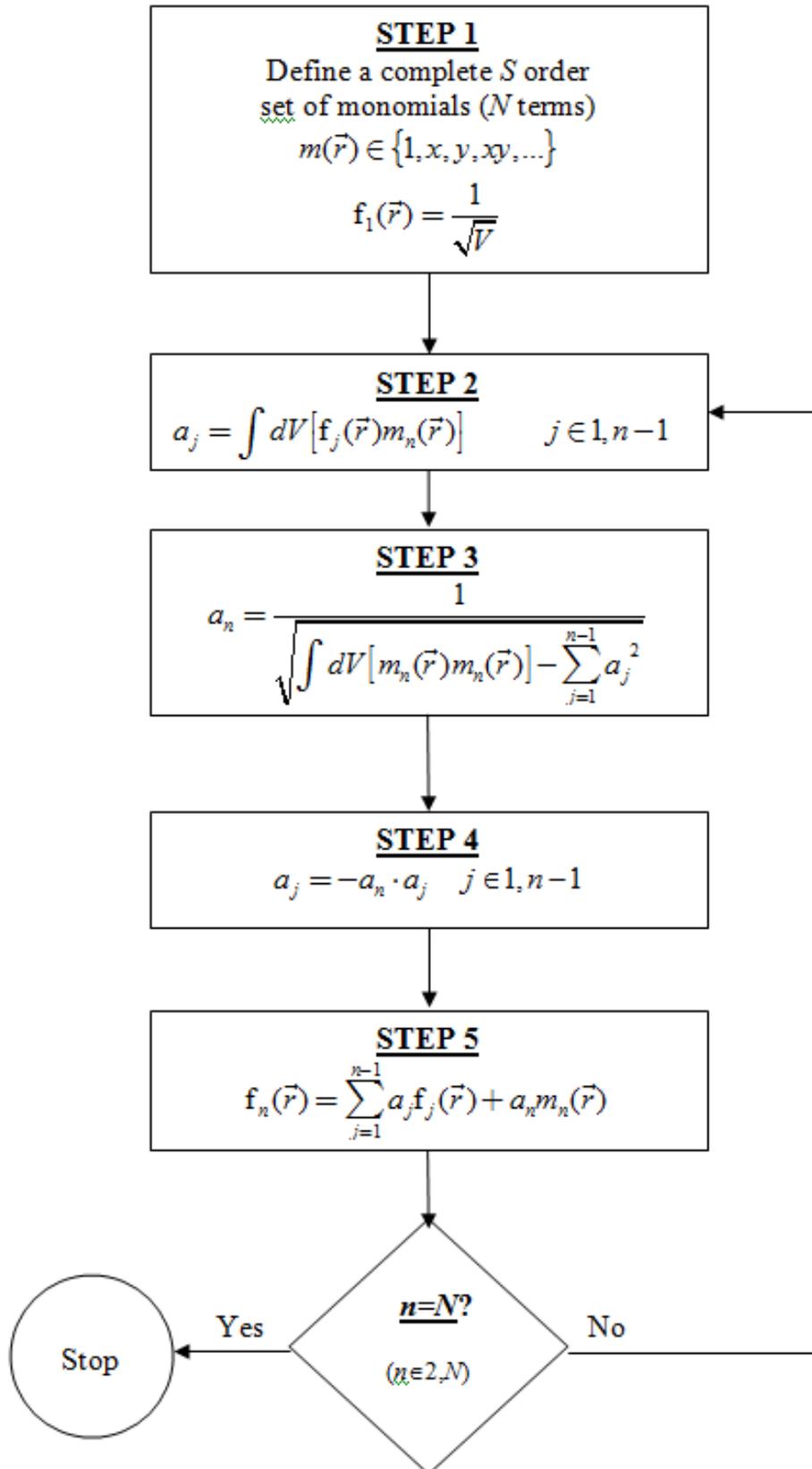


Figure 4.1. Gram-Schmidt Procedure for Obtaining Orthonormal Functions

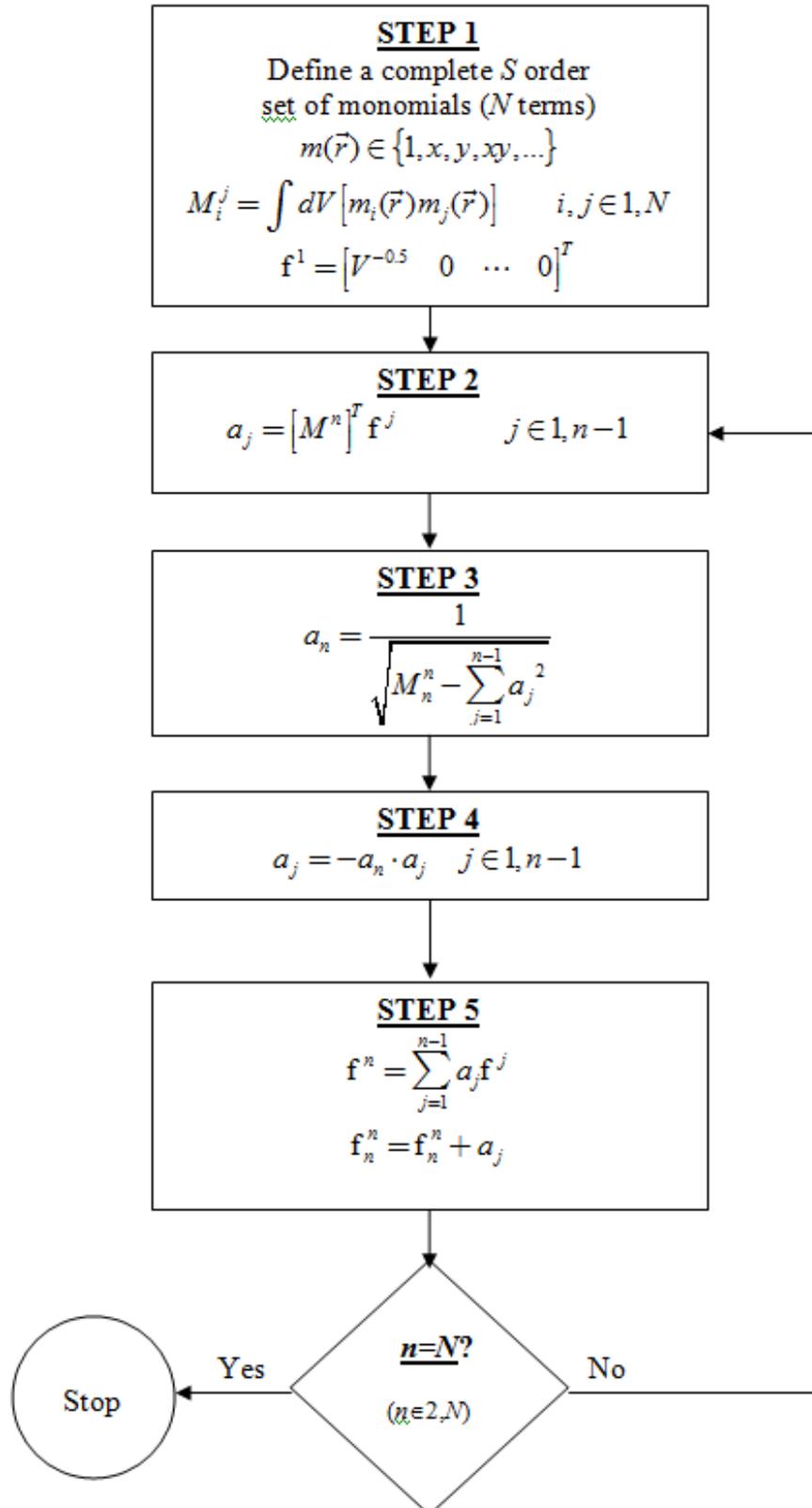


Figure 4.2. Fortran Approach to Performing Gram-Schmidt Procedure.

Given the set of orthogonal functions the integrations of equations 93-99 can be performed. In VARIANT, the cross sections are assumed to be constant within each node (no spatial dependence), thus the spatial integrations can be reduced to those shown in equations 204-208.

$$P_{i,j}^{K,L} = \int \tilde{N}^K f_i(\mathbf{r}) \tilde{N}^L f_j(\mathbf{r}) dV \quad (204)$$

$$U_{i,j}^K = \int f_i(\mathbf{r}) \tilde{N}^K f_j(\mathbf{r}) dV \quad (205)$$

$$F_{i,j} = \int f_i(\mathbf{r}) f_j(\mathbf{r}) dV \quad (206)$$

$$D_{i,j}^g = \int f_i(\mathbf{r}) h_{g,j}(\mathbf{r}) dG \quad (207)$$

The integration in equation 206 does not need to be performed since it is by definition the identity matrix. For this report only the integrations of equation 204 will be detailed with the others briefly described.

To perform the integrations in equation 204, an approach similar to that used to obtain the orthonormal functions is used. Given the set of monomials (from step 1 of Figure 4.2), equation 204 can be written in terms of the monomials to get

$$M_{i,j}^{K,L} = \int \tilde{N}^K m_i(\mathbf{r}) \tilde{N}^L m_j(\mathbf{r}) dV . \quad (208)$$

Since the exponents of the monomials n , m , and o are known, the derivative process in equation 208 becomes a trivial algebraic process (subtraction). To obtain the matrix components in equation 204, the following relation is used.

$$P_{i,j}^{K,L} = \int \tilde{N}^K M^{K,L} \mathbf{f}^j \quad (209)$$

From Figure 4.2, \mathbf{f}^i is the coefficient vector for the i th orthonormal function. An almost identical procedure applies to the U matrix integrations of equation 205, while a slightly modified version must be used for the D matrix integrations in equation 207. In the D matrix integrations, the coefficients for a set of surface functions and volume functions must be created and used (surface functions will appear on the right hand side of the monomial integration matrix in equation 209). The projection of the monomials to the surface of the node is straightforward and thus the M matrix is straightforward for equation 207.

For this work, the Cartesian matrices were all obtained using analytical solutions to the monomial integrations. For the hexagonal geometries, the surface projection monomial integrations were carried out using numerical integration. Although recent work of Yang has given the analytical results for the hexagonal monomials, no effort was made to implement them into the coding. Consequently, the hexagonal spatial expansion is limited to about a 48th order spatial approximation. The impact on the computational efficiency caused by using a numerical approach should be obvious; however, the impact was found to be insignificant for all problems tested thus far (probably adds a couple of milliseconds onto the coupling coefficient time). If at some point in the future, the spatial approximation is to be regenerated for every spatial node, rather than done once at the beginning of the VARIANT code execution, then it is suggested that the analytical form be implemented.

This concludes the new work done on the spatial approximation in the VARIANT coding. The above methodology was put into Fortran subroutines using standard Fortran 90 coding to again eliminate problems with Fortran 77 memory allocation routines. As was the case with the angular approximation, the spatial matrices in the production version of VARIANT were hardwired at different points in the coding making a single subroutine call almost impossible. The solution again was to write unique subroutines to obtain only the matrices needed at the various points in the VARIANT coding.

The DIF3D BCD input description also had to be changed to allow the new spatial approximations to be used. This affects the second input descriptor on the card type 12 of DIF3D BCD input and the NAPRX variable in the binary DIF3D file. The major changes to the BCD input are shown in Table 4.1 and modified descriptions for the DIF3D BCD input are given in Appendix A.

Table 4.1. Changes to the DIF3D BCD Input Description on Card 12.

Input Field	LMN	LLMMXN or LMN
L	Order of the Source	Order of the Source
M	Order of the Flux	Order of the Flux
N	Order of the Leakage	Order of the Leakage
X	--	Order Reduction in the Z Direction
Examples	462	040622 or 462
	442	040402 or 442

As can be seen in Table 4.1, there is a new variable, X, included in the input which is not consistent with that of the preexisting formulation. The reason for the inconsistency is that the production version of VARIANT only allowed a maximum 4th order flux and source approximation in the z direction while the x-y flux approximation was extended to 6th order. To keep this flexibility, the new variable was added into the DIF3D input definition, although its usage is not required for the new input style.

With the flexibility in the input, the question of matrix rank deficiencies must be addressed. Although not detailed here, a check must be performed to guarantee that the spatial projection to the interface does not produce rank deficient response matrices. This check is performed once at the beginning of the code (computational time is insignificant) and an error warning is generated if the matrices are determined to be rank deficient. The reason for using an error warning rather than a stoppage is that the 6th order x-y, 4th order z flux approximation was found to be rank deficient when combined with a quadratic leakage approximation (662 for 3-d Cartesian and hexagonal geometries). As it turns out only severe rank deficiencies will cause problems during the iterative algorithm. Since slightly rank deficient approximations such as “662” do work under most situations, it was decided that a warning would be issued rather than a stoppage. Typical indications of spatial rank deficiencies are slow and unstable convergence of the inner iteration scheme. For severely rank deficient matrices, in which an insufficient number of spatial degrees of freedom have been defined, an error stoppage is issued.

5 Non-Zero Storage Implementation

With the above modifications to the spatial and angular matrix definitions implemented into the VARIANT code, there remains one additional issue to address, the matrix free storage application of the anisotropic scattering matrices. To begin, we identify the main iteration equations from section 2 that VARIANT uses as equations 112 and 118 and reproduce them here as equations 210 and 211.

$$j^+ = \bar{B}^+ S^+ + \bar{B}^- S^- + \bar{R}c \quad (210)$$

$$z = \bar{A}^{-1} \bar{W} S^+ + \bar{A}^{-1} \bar{T} S^- - 2\bar{C}^T c \quad (211)$$

DIF3D-9 formed and stored the matrices: \bar{B}^+ , \bar{B}^- , \bar{R} , \bar{A}^{-1} , $\bar{A}^{-1} \bar{T}$, \bar{T} , and \bar{C}^T . Unlike the previous derivation, the matrix \bar{T} is stored without the group wise cross sections and used in a full matrix-vector operation with the odd-parity source. What is missing from this system is the solution of the odd-parity flux which is derived from equation 88 and given in matrix form as

$$\bar{A}^{-1} F \bar{W} y^- = \sum_{g,K} \mathcal{S}_0 \bar{V}_K^T \bar{U}_K^T y^+ + \sum_{g,K} \mathcal{S}_0 F y^-, \quad (212)$$

where we have dropped the group notation and redefined the matrix

$$U_{i,j,K} = \int_0^1 f_i(r) \bar{N}_K f_j(r) dV, \quad (213)$$

and defined the cross section angular matrix of odd-parity size as

$$\mathcal{S}_0 = \frac{1}{S_t - \bar{S}_-}. \quad (214)$$

Given that F is an identity matrix for the trial functions we can rewrite equation 212 as

$$y^- = T y^+ + \sum_{g,K} \mathcal{S}_0 A I y^-. \quad (215)$$

In this form, one can easily see the real reason for not storing the \bar{T} matrix for each group.

The motivation behind the most recent work is associated with the performance of VARIANT. The primary issue we identified with the production version of VARIANT is a general performance degradation as the number of energy groups increases when anisotropic scattering is being used. After some study, this performance issue was traced directly to the expense of applying the \bar{T} matrix in equations 211 and 212. This matrix also causes severe memory constraints due to its inherent size and the 32 bit storage limitations of BPOINTER. As it turns out, the \bar{T} matrix is ridiculously sparse and the analysis indicated that using non-zero storage of the \bar{T} matrix could not only reduce the overall memory storage of VARIANT, but also eliminate the performance degradation. Rather than just focusing on this change, we also identified that the application of \bar{B}^- and $\bar{A}^{-1} \bar{T}$ could be improved by eliminating them entirely and using the relationship from equation 116, reproduced here as equation 216

$$\bar{B}^- = \bar{B}^+ \bar{T}. \quad (216)$$

Given that we have to operate with \bar{B}^+ and \bar{A}^{-1} we can see that the extra matrices can be completely eliminated assuming we have a means by which to apply \bar{T} . This approach is more efficient because the \bar{T} matrix can be applied in a very cheap manner using the non-zero storage protocol and the previous work of Yang [5] made the computational effort required to apply \bar{B}^+

and \bar{A}^{-1} very cheap. The new version stores the spatial and angular matrices in separate non-zero storage routines and applies the tensor operations using these routines. While this doesn't guarantee the minimal number of multiplications, it does minimize the memory required to apply these routines which should be ideal for performance on large problems. This leaves the primary issue of memory in the formation and inversion of \bar{A} as the remaining inhibitor to using higher order approximations. With time this problem might also be fixed given that \bar{A} is block diagonal and we are using a full matrix storage protocol. Unfortunately this will not significantly alter the computational performance of the underlying algorithm further.

6 Large Memory Array Implementation

6.1 Introduction

Numerous VARIANT subroutines have been modified to increase the efficiency and memory utilization of large VARIANT jobs in DIF3D-11 over previous versions. The changes involve removal of numerous BPOINTER [] calls in favor of LMA, the Large Memory Array module developed for UNIC [] under NEAMS.

6.2 Implementation of LMA

In DIF3D-10 (and preceding versions of DIF3D), the program BPOINTER is used to circumvent the lack of dynamic storage capability in FORTRAN 77, and to move data to and from disk in order to fit large problems on limited computing resources. BPOINTER is called by DIF3D to obtain and/or release large blocks of workspace called “containers”. However, BPOINTER is restricted to 32-bit memory addressing, and so the memory container size is limited to 2 GB regardless of the actual available memory. For large problems, this limitation results in performance degradation due to partitioning the data on disk and repeated calls to perform retrieve/put data on the disk. In several cases, very large problems cannot even be run using DIF3D-10.0 which can leave unresolved questions about some analysis work. From a programming perspective, calls to BPOINTER functions are difficult to understand and maintain due to complex pointer math and outdated naming conventions.

The dynamic storage capabilities in modern FORTRAN make the BPOINTER usage obsolete. A modern memory management program, LMA (Large Memory Array), has been implemented in DIF3D-11. LMA uses 64-bit memory addressing and efficiently utilizes the full available machine memory (has been tested to use over 132 GB of memory). It performs the same tasks as BPOINTER but in a more efficient and intuitive manner, handling data partitioning to disk (if necessary) internally to the module. LMA simplifies the procedure for accessing a particular section of any given large array.

The following examples (taken from vstou2.f in DIF3D-11) show how the group source vector is accessed using BPOINTER, and alternatively, LMA.

```
Example: Using BPOINTER to load external source vector
CALL OPENDF(FSRC,NDFSRC)
CALL BLKGET(NCSRC,KBLKS,NDFSRC,KBLKS)
CALL FINGET(NCSRC,KBLKS,NDFSRC,KBLKS)
```

```
Example: Using LMA to load external source vector
Matrix_Access%Partition_Start(2) = NG
Matrix_Access%Partition_End(2)   = NG
Matrix_Access%Partition_Start(1) = 1
Matrix_Access%Partition_End(1)   = iStorage_iSizeExtSrc
CALL NE_LMA_R64_Get(NOUT,Vector_ExtSrc,Matrix_Access,iOffset_ExtSrc)
```

The LMA Partition_Start and Partition_End points correspond to the index ranges of the desired multidimensional array, in this example, Vector_ExtSrc(1:iStorage_iSizeExtSrc, NG:NG). This assignment tells LMA to retrieve the first iStorage_iSizeExtSrc values stored for group NG. The function NE_LMA_R64_Get does the necessary pointer math and file reading to return the desired data at the physical memory address “iOffset_ExtSrc” within the physical container: Vector_ExtSrc%Storage (this last piece is part of the structure). LMA usage is more transparent than the BPOINTER subroutines OPENDF (opens a file associated with the name “FSRC” if it is not already open), BLKGET (gets the physical 32-bit pointer address, NCSRC, into the memory space BLK for the array named “NDFSRC” which needs to contain KBLKS points of data), and FINGET (loads data from the associated file connected to the BPOINTER array NDFSRC into the provided position).

6.3 Impact of LMA

The primary advantage of DIF3D-11 over previous versions is the ability to utilize the full machine memory efficiently using LMA as opposed to BPOINTER. Users of DIF3D-11 (VARIANT) are able to run larger problems, perform complete space-angle convergence studies, and suffer with faster runtimes. Developers will also find DIF3D-11 easier to maintain due to the replacement of numerous BPOINTER calls with the more intuitive LMA functions. Note that if DIF3D-11 cannot obtain the necessary physical memory, it will attempt to create random access files in the /tmp directory with job specific numbering. We strongly suggest attaching an additional hard drive to the computer mounted as /tmp and adjusting the buffering on files if you intend on running these jobs on small memory machines. If the creation of those files fails, it will attempt to create files in the current working directory which can be disastrous if that is a network mapped file system.

7 Additional DIF3D-11.0 Algorithm Changes

This section describes additional algorithm changes made to VARIANT in DIF3D-11 after the initial version was created to accommodate new features.

7.1 Fission source error measures and acceleration

The subroutines that deal with the fission source computation and convergence tracking have been rearranged considerably from DIF3D-9. Coarse mesh rebalance (CMR) and asymptotic source extrapolation have been permanently disabled (deleted) in DIF3D-11 in anticipation of a better acceleration algorithm being implemented. In the interim, Tchebychev acceleration was added to counter the reduced performance although the ability of DIF3D 11.0 to use larger memory spaces generally overwhelms the performance issues associated with eliminating the previous acceleration.

To understand the troubles with coarse mesh rebalance, we selected a series of problems to display the issues. Before we begin, it is important to note the changes in error norm criteria that cause fundamental differences in the convergence behavior. First, the fission source norm used to estimate the eigenvalue in version 9.0 weighted the fission source with the volume where as version 11.0 does not. In version 9.0, it was observed that large volume fissionable meshes would dominate the eigenvalue convergence such that when using rather high convergence criteria (greater than 10^{-5} on the eigenvalue) the actual solution given by 9.0 was not precisely within the stated error of the fully converged solution. Second, the average source error in version 9.0 was defined using equation 217 while it was modified for version 11.0 to be 218.

$$e = \frac{1}{\overset{\circ}{MESHES}} \frac{\overset{\circ}{fission}_{mesh,flat}^{outer} - \overset{\circ}{fission}_{mesh,flat}^{outer-1}}{\overset{\circ}{fission}_{mesh,flat}^{outer}}. \quad (217)$$

$$e_{11.0} = \sqrt{\frac{\overset{\circ}{(fission}_{mesh,flat}^{outer} - fission_{mesh,flat}^{outer-1})^2}}{\overset{\circ}{fission}_{mesh,flat}^{outer} \times fission_{mesh,flat}^{outer}}}. \quad (218)$$

Given that equation 218 is a much more strict check on the spatial convergence, the user must understand that they can weaken the fission source convergence criteria slightly compared with past usages to get the “same” answer. Unfortunately, because of the change in the eigenvalue norm, one cannot get version 11.0 to “converge” using exactly the same number of outer iterations as version 9.0 in all cases, but they are routinely close.

The first test problem we consider is a bare homogeneous system on which CMR and fission source extrapolation should be ideal. Spatially we chose a 6th order flux, 1st order leakage, and 4th order source and chose to only consider diffusion theory. We constructed multiple reactor sizes and choose only to give the two of relevance in Table 7.1. Note that in both cases (1 meter tall and 2 meter tall), we vary the pitch from 5 cm to 40 cm to cover most realistic cases although both end points are unrealistic. In each problem we give the existing version 9.0 outer iteration result with and without CMR along with the version 11.0 result with Tchebychev acceleration

and the un-accelerated case (almost always identical or one outer iteration off between version 9.0 and 11.0 for this problem).

Starting with the full core results, one can see that the version 9.0 and version 11.0 results are very comparable up to pitch of 8 cm. The dominance ratio above this point transitions from 0.6 to 0.75 at a 14 cm pitch and 0.96 at 40 cm. In this situation, it should be clear that CMR and fission source extrapolation make a considerable difference giving a 71% reduction when using a 16 cm pitch where as the Tchebychev acceleration only yields a 50% reduction in effort. We note that fission source extrapolation accounts for a bulk of the gain.

Table 7.1. Bare, Homogeneous Hexagonal Reactor Outer Iteration Count in Diffusion Theory

Hex Pitch (cm)	Full Core, 10 rings, 1 meter tall				120 periodic, 10 rings, 2 meter tall			
	V9.0	V9.0 no CMR	V11.0	No Accel	V9.0	V9.0 no CMR	V11.0	No Accel
5	18	21	22	29	32	39	45	75
6	16	22	21	27	31	32	47	69
7	17	21	20	26	31	32	41	64
8	18	22	21	27	33	33	41	61
10	18	22	31	33	30	34	40	58
12	18	25	29	41	33	34	42	55
14	18	30	31	51	29	34	32	55
16	18	30	32	63	29	36	40	60
20	21	40	47	89	33	38	47	83
30	35	57	95	179	34	59	94	163
40	56	85	167	298	45	79	158	270

Continuing with the 120 periodic cases which has a 2 meter height, one sees a similar result in that CMR gives almost no improvement in the acceleration for a bulk of the cases. While Tchebychev acceleration is not as good as CMR + fission source extrapolation, it still yields a 33% gain with a 16 cm pitch compared with a 52% gain.

The next problem we consider is a scatter load system the composition assignment for which is depicted in Figure 7.1 along with the fast flux profile. As can be seen, the scatter pattern is relatively regular and not necessarily indicative of a true scatter load reactor, but the flux solution clearly shows no real symmetry pattern which is the underlying goal. We constructed two cores with 12 and 16 rings and show an example of the geometry and fast flux for the 16 ring case in Figure 7.1. We would not expect CMR to work well in either case given the way it was implemented but fission source extrapolation should still yield some improvements. We built hexagonal and Cartesian geometries (16 rings means a 16x16 grid) and only chose to include the full core results in Table 7.2 for each case with a two meter tall geometry. Note that we included 40 cm of axial reflector in each case.

Starting with the hex geometry case, one can easily see the CMR result degrades considerably as the hex pitch size is increased. At 16 cm, the dominance ratio is 0.84 and CMR actually hurts the overall performance although the code still finishes quicker (53% reduction) than the Tchebychev accelerated version 11.0 (36% reduction). Overall, Tchebychev performs

about as well as the old acceleration for the smaller node sizes. At larger node sizes one sees that fission source extrapolation is the best approach. The Cartesian geometry cases show much better improvement with CMR although fission source extrapolation is clearly accounting for a bulk of the acceleration.

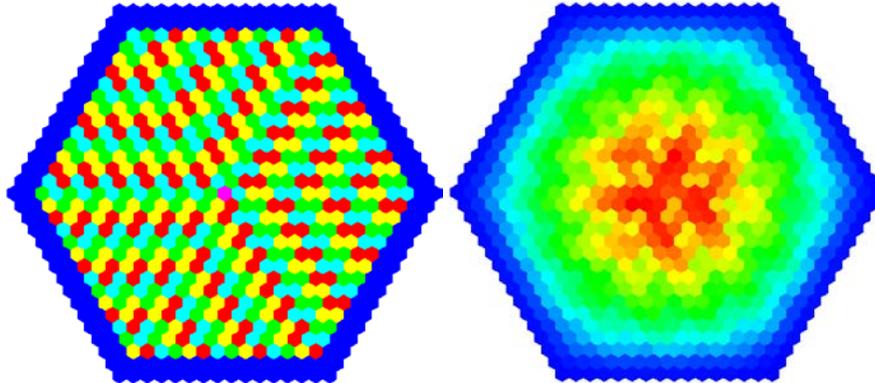


Figure 7.1. Scatter Load Example for Hex Geometry and Fast Flux Profile

Table 7.2. Twelve Ring Scatter Core Outer Iteration Count in Diffusion Theory

Hex Pitch (cm)	Hex Full Core				Cartesian Full Core			
	V9.0	V9.0 no CMR	V11.0	No Accel	V9.0	V9.0 no CMR	V11.0	No Accel
5	22	24	27	30	21	29	43	67
6	18	22	24	31	22	27	42	62
7	18	24	26	33	18	27	35	58
8	22	26	26	36	17	28	34	56
10	23	30	26	49	17	30	34	53
12	29	32	45	65	21	32	35	51
14	33	43	55	84	24	27	38	56
16	49	40	67	106	24	34	43	69
20	62	59	101	162	30	42	64	101
30	112	123	208	386	43	62	119	208
40	152	185	392	500	55	81	205	344

To study “high” dominance ratio problems, we expanded the reactor to 16 rings and found we had to eliminate the large hex pitches due to computational run time in version 9.0. The outer iteration results are tabulated in Table 7.3 for the hexagonal and Cartesian cases where we extended the domain to be 4 meters tall. Assuming the 16 cm pitch would be the target case, this was observed to have a 0.95 dominance ratio. Looking at the hexagonal results, one can see that fission source extrapolation again accounts for a bulk of the acceleration but in this case, CMR leads to net gains in each case with a 28% reduction at a 16 cm pitch. The Cartesian cases are considerably different indicating that CMR clearly makes substantial improvements in the performance (50% over fission source extrapolation and 80% over unaccelerated) which we believe is more due to the number of axial meshes than the increase in radial meshes.

Table 7.3. Sixteen Ring Scatter Core Outer Iteration Count in Diffusion Theory

Hex Pitch (cm)	Hex Full Core				Cartesian Full Core			
	V9.0	V9.0 no CMR	V11.0	No Accel	V9.0	V9.0 no CMR	V11.0	No Accel
5	20	46	58	102	22	71	121	213
6	24	43	48	97	20	70	110	200
7	27	37	52	95	21	70	124	193
8	32	38	54	93	21	66	96	188
10	37	40	54	92	23	64	101	183
12	40	49	68	99	25	74	103	180
14	49	62	71	127	31	75	107	179
16	53	74	105	163	36	73	92	178

We note that all of this work was done in diffusion theory as past experience using transport shows even more minimal gains in CMR acceleration if not negative gains like that observed in Table 7.2. More problematic is that CMR acceleration regularly fails to converge which was the case for all 120 periodic geometries of the scatter load problems and most of the 60 degree periodic geometries. Further, CMR is apparently not implemented correctly when solving the adjoint transport equation as it always fails to converge whether using Cartesian or hexagonal geometries. Given the rather paltry gains over fission source extrapolation, CMR was deleted in version 11.0 assuming that we would replace it with a more effective acceleration algorithm. The asymptotic source extrapolation in all of the preceding problems was found to be quite good, however, these are ideal cases with only a few unique compositions. For a bulk of the actual problems we have executed in the last decade, we have found that extrapolation caused considerable instabilities when using more than 20 groups and it only reduced the number of outer iterations by a few. In the example problem of the next section, a typical user reactor problem, source extrapolation does work and it is responsible for reducing the number of outer iterations by 3 when only engaged twice (coarse mesh rebalance is responsible for reducing the number of outer iterations by 10).

Tchebychev acceleration was expected to be a more reliable scheme than fission source extrapolation on more routine reactor problems, but in order for Tchebychev acceleration to work properly, the dominance ratio estimate must be reasonably stable and accurate. In most problems with large dominance ratios, whether fast or thermal, the dominance ratio is rather stable and Tchebychev works as seen above. However, in some small, transport dominated systems with large amounts of leakage, the dominance ratio estimate effectively becomes a random number between 0.0 and 2.0. Of course the real dominance ratio is typically far less than 0.5 in these cases. Thus whether we use CMR or Tchebychev, the acceleration is going to be minimal as neither of them are effective acceleration schemes as seen with the small problems of Table 7.1. As a consequence, the Tchebychev acceleration is restricted such that it will not engage unless 1) the dominance ratio varies minorly over four consecutive outer iterations and 2) the P_0 based dominance ratio estimate does not deviate far from the actual dominance ratio.

The second criteria is an artifact of the partitioned diffusion VARIANT methodology which is geared to maximize convergence of the diffusive system rather than convergence of the total system. This acceleration works wonderfully when the problem is diffusion dominant, but

in small leaky reactor systems, it can become transport dominant and thus the dominance ratio estimates using just the P_0 components versus the full system can be wildly different. In this case, we cannot use Tchebychev as it is geared to accelerate the full system. It is important to note that CMR was observed to breakdown similarly in these cases.

When following these two criteria, Tchebychev acceleration almost always improves the performance. However, we have observed that on some very small problems where the dominance ratio can be temporarily stable and Tchebychev inappropriately engages leading to considerably longer convergence times. In these cases (i.e. small, leaky, transport dominated reactor systems), we suggest the user disable Tchebychev acceleration to ensure that it is not causing convergence problems.

As a final step, when the dominance ratio is observed to be consistently high and Tchebychev is not activated, an auto adjustment routine was added to increase the number of inners in each outer. This routine is restricted to dominance ratios greater than 0.85 and determines whether increasing the number of inner iterations can reduce or stabilize the dominance ratio. It only makes a few attempts to increase the number of inners when necessary and is hesitant in increasing the number of inners further if an improvement is observed. In effect, we rely upon the user to recognize the problems where the predicted number of inners are insufficient and increase the number of inners appropriately. From experience, this has rarely been necessary.

7.2 Modification to the DIF3D outer iteration history

The DIF3D 10.0 output during the outer iteration convergence was found to be insufficient by most users. In many cases it did not adequately assist either the user or the developer in understanding the specific problem with a particular input deck that was failing or provide an idea of how much time was remaining for convergence. To correct this issue, the output specification was modified to include more detailed information which we describe here.

Figure 7.2 shows the example un-accelerated output from DIF3D 10.0 for a transport dominated problem while Figure 7.3 shows the output when coarse mesh rebalance and fission source extrapolation are enabled and both working. One should note that the convergence criteria are sufficiently low enough that the residual error between the two solutions is within the error criteria and the acceleration saves 15 outer iterations. Starting with Figure 7.2, one can see that the outer iteration number is given along with the relative point error (maximum fission source error), relative sum error (relative flux error), and eigenvalue error (change). These quantities represent the primary error measures used to define convergence of the steady state problem. The additional quantities of fission source extrapolation (FSRC. EXTRAP.), dominance ratio estimate, and rebalance error are associated with the acceleration schemes applied to DIF3D 10.0 which were made defunct in 11.0. The final column of data is the eigenvalue itself which is a common quantity of interest most developers and users wish to see.

Comparison of Figure 7.3 to Figure 7.2 indicates that fission source extrapolation is engaged twice and when disabled we found it increased the number of outer iterations on this problem by 3 (i.e. 25 outer iterations). Disabling the ω acceleration does not change the number of outer iterations, but it has a substantial impact upon the computational effort as the number of

inner iterations are increased by a factor of 2. It is also the reason for the error between the DIF3D 10.0 and DIF3D 11.0 result. Reducing the number of inners to match the accelerated case yields no difference in the solution indicating that the estimation of the number of inners is erroneous in that case. From all of this, it should be clear that coarse mesh rebalance (CMR) is the primary acceleration for this problem and saves 12 outer iterations worth of work.

Figure 7.4 shows the outer iteration history for DIF3D 11.0. To begin, one can see a “=VARIANT” line at each outer iteration. This was done to allow users to grep the convergence history from extensive output files (REBUS) utilizing restart and thus quickly assess convergence problems. The remaining info is similar to the DIF3D 10.0 output, but we note the total time in seconds required for each outer iteration is given. Although present in the previous DIF3D 10.0, it is never listed how many upscattering iterations (Gauss-Siedel iterations over the energy system) are actually used. In fast spectrum systems, only a single Gauss-Siedel iteration is needed and thus “Ups” is reported as 1 for each outer iteration in this case.

The next columns in Figure 7.4 are the eigenvalue (K-effective) followed by its error allowing the user to easily locate and understand how far the eigenvalue is from convergence noting that additional significant digits on the eigenvalue are unnecessary as the error fully informs the user as to the convergence. The “pGS” column indicates whether the partitioned matrix Gauss-Siedel acceleration scheme is being used on the within group equation noting that in outer iteration 16 it is disabled. This is fully consistent with the behavior of DIF3D 10.0 although it only reported the fact that it had done it much later in the as “FULL MATRIX SWEEP.” The maximum fission source error (Fis Max) and the RMS error are then reported which are the replacement norms as discussed. Note that Fis Max is consistent with the relative point error of DIF3D 10.0.

Four columns are dedicated to Tchebychev acceleration which indicate whether it is enabled and the number of back vectors currently being used. The “Dom” and “P0” are the conventional Tchebychev dominance ratio estimates where “P0” only considers the flat diffusive moments. The remaining columns show the breakdown in computational time for the three main subroutines which is the source computation “Src(s),” partial current solution “j+/- (s),” and flux reconstruction equations “Phi(s).” These timing measurements better explain to the user (and developer) what aspect of the input affects the performance of the code. The source computation is the time spent getting the up and down scattering source for each outer. The partial current is the time spent solving the partial current equations in each within group equation. The flux reconstruction is the act of updating the nodal flux values given the updated partial currents. This flux reconstruction component can be observed to change dramatically near the end of a computation will check for unnecessary P_N moments due to the lack of scattering data and skip the reconstruction. To produce a valid flux solution at the end of the computation, the flux moments must be fully constructed which is typically carried out in the very last outer iteration performed. Tchebychev was only engaged for the last 7 iterations and saved 10 outer iterations.

Figure 7.5 shows an example of spectral radius problem with a slightly older version of DIF3D 11.0. In this case, the eigenvalue is observed to converge, but the fission source is clearly diverging after outer iteration 22. This cannot be resolved without altering the geometry as acceleration makes no difference. The typical cause is a nearly void node somewhere in the domain which is the case in this problem as we provide as an example of what to watch for.

0	OUTER IT. NO.	REL. POINT ERROR	REL. SUM ERROR	EIGENVALUE CHANGE	FSRC. EXTRAP.	DOM. RATIO ESTIMATED	REBALANCE ERROR	K-EFFECTIVE
0	1	1.138104E+02	5.330136E+00	1.050883E-01	NO	1.693809E+12	9.214395E-01	1.10508834E+00
0	1	8.174757E+00	6.275693E-01	-3.873569E-01	NO	7.895107E+11	0.000000E+00	6.12643118E-01
0	2	2.464780E+02	3.545478E-01	1.967998E-01	NO	1.881540E+00	0.000000E+00	8.09442893E-01
0	3	1.439890E+02	3.876838E-01	1.326190E-01	NO	6.467476E-01	0.000000E+00	9.42061885E-01
0	4	1.349842E+01	1.668780E-01	4.554079E-02	NO	3.494048E-01	0.000000E+00	9.87602679E-01
0	5	1.257865E+00	5.042699E-02	1.066960E-02	NO	2.856158E-01	0.000000E+00	9.98272275E-01
0	6	3.891613E-01	1.057393E-02	1.806735E-03	NO	3.910432E-01	0.000000E+00	1.00007901E+00
0	7	8.130482E-02	2.029900E-03	1.492255E-04	NO	4.166888E-01	0.000000E+00	1.00022824E+00
0	8	2.299323E-02	6.576237E-04	-5.324372E-05	NO	6.463302E-01	0.000000E+00	1.00017499E+00
0	9	1.381967E-02	3.030294E-04	-4.417065E-05	NO	2.905599E-01	0.000000E+00	1.00013082E+00
0	10	7.659400E-03	1.377861E-04	-2.563242E-05	NO	3.171578E-01	0.000000E+00	1.00010519E+00
0	11	4.248676E-03	6.598536E-05	-1.493265E-05	NO	4.757461E-01	0.000000E+00	1.00009026E+00
0	12	2.453879E-03	3.333214E-05	-9.076385E-06	NO	5.361753E-01	0.000000E+00	1.00008118E+00
0	13	1.503240E-03	1.761314E-05	-5.779421E-06	NO	5.702603E-01	0.000000E+00	1.00007540E+00
0	14	9.786356E-04	9.793555E-06	-3.934385E-06	NO	6.501605E-01	0.000000E+00	1.00007147E+00
0	15	6.736367E-04	6.059926E-06	-2.918724E-06	NO	7.299196E-01	0.000000E+00	1.00006855E+00
0	16	4.871206E-04	4.252960E-06	-2.350417E-06	NO	7.918137E-01	0.000000E+00	1.00006620E+00
0	17	3.677257E-04	3.314987E-06	-2.011352E-06	NO	8.413575E-01	0.000000E+00	1.00006419E+00
0	18	2.986403E-04	2.755819E-06	-1.788753E-06	NO	8.779158E-01	0.000000E+00	1.00006240E+00
0	19	2.482796E-04	2.379072E-06	-1.628036E-06	NO	9.018469E-01	0.000000E+00	1.00006077E+00
0	20	2.098784E-04	2.099818E-06	-1.503125E-06	NO	9.166544E-01	0.000000E+00	1.00005927E+00
0	21	1.799468E-04	1.879889E-06	-1.400959E-06	NO	9.259899E-01	0.000000E+00	1.00005786E+00
0	22	1.561543E-04	1.701333E-06	-1.314399E-06	NO	9.322626E-01	0.000000E+00	1.00005655E+00
0	23	1.369064E-04	1.552406E-06	-1.239167E-06	NO	9.367847E-01	0.000000E+00	1.00005531E+00
0	24	1.210879E-04	1.425586E-06	-1.172512E-06	NO	9.402329E-01	0.000000E+00	1.00005414E+00
0	25	1.079031E-04	1.315527E-06	-1.112580E-06	NO	9.429708E-01	0.000000E+00	1.00005303E+00
0	26	9.677332E-05	1.219148E-06	-1.058078E-06	NO	9.452118E-01	0.000000E+00	1.00005197E+00
0	27	8.727108E-05	1.133840E-06	-1.008074E-06	NO	9.470928E-01	0.000000E+00	1.00005096E+00
0	28	7.907548E-05	1.057859E-06	-9.618801E-07	NO	9.487058E-01	0.000000E+00	1.00005000E+00
		OUTER ITERATION WITH FULL MATRIX SWEEP IS PERFORMED						
0	30	1.156758E-04	1.676986E-06	-1.654761E-06	NO	1.067279E+00	0.000000E+00	1.00004671E+00
0	31	9.800049E-05	1.521791E-06	-1.527986E-06	NO	9.379693E-01	0.000000E+00	1.00004518E+00
0	32	8.385475E-05	1.362965E-06	-1.408546E-06	NO	9.186980E-01	0.000000E+00	1.00004377E+00
0	33	7.207775E-05	1.225934E-06	-1.301730E-06	NO	9.176868E-01	0.000000E+00	1.00004247E+00
0	34	6.219366E-05	1.106847E-06	-1.205090E-06	NO	9.193264E-01	0.000000E+00	1.00004127E+00
0	35	5.384640E-05	1.002740E-06	-1.117599E-06	NO	9.212795E-01	0.000000E+00	1.00004015E+00
0	36	4.675516E-05	9.113880E-07	-1.038205E-06	NO	9.232417E-01	0.000000E+00	1.00003911E+00
0	37	4.069871E-05	8.311961E-07	-9.659612E-07	NO	9.250951E-01	0.000000E+00	1.00003814E+00
		MAX ERROR ON POINT FISSION SOURCE OCCURED AT NODE 655 ON PLANE 2						
		0 OUTER ITERATIONS COMPLETED AT ITERATION 37, ITERATIONS HAVE CONVERGED						
0		K-EFFECTIVE = 1.00003814425						

Figure 7.2. Example DIF3D 8.0 through 10.0 Outer Iteration Information without Acceleration

0	OUTER IT. NO.	REL. POINT ERROR	REL. SUM ERROR	EIGENVALUE CHANGE	FSRC. EXTRAP.	DOM. RATIO ESTIMATED	REBALANCE ERROR	K-EFFECTIVE
0	1	1.138104E+02	5.330136E+00	1.050883E-01	NO	1.693809E+12	9.214395E-01	1.10508834E+00
0	2	1.743296E+02	5.058496E-01	-7.648824E-02	NO	2.761199E-01	2.724583E-01	1.02860011E+00
0	3	1.217475E+00	4.562054E-02	-2.176428E-02	NO	2.123212E-01	2.808327E-02	1.00683583E+00
0	4	3.457592E-01	8.301705E-03	-5.221585E-03	NO	1.745165E-01	7.591008E-03	1.00161425E+00
0	5	7.775910E-02	2.582331E-03	-9.431916E-04	NO	2.257507E-01	1.330915E-03	1.00067105E+00
0	6	3.070483E-02	6.635208E-04	-3.695707E-04	NO	3.778580E-01	4.080813E-04	1.00030148E+00
0	7	5.406078E-03	1.874969E-04	-1.319292E-04	NO	5.068078E-01	1.169308E-04	1.00016955E+00
0	8	2.269474E-03	9.650244E-05	-4.901412E-05	NO	4.903609E-01	5.396055E-05	1.00012054E+00
0	9	1.130768E-03	4.837029E-05	-2.231372E-05	YES	4.811589E-01	3.087018E-05	1.00009823E+00
0	10	2.973780E-04	1.119877E-05	-2.557589E-05	NO	3.961954E-01	6.302524E-06	1.00007265E+00
0	11	1.004393E-04	3.955890E-06	-5.896087E-07	NO	1.596102E-01	3.033608E-06	1.00007206E+00
0	12	8.459904E-05	3.040804E-06	-2.283492E-06	NO	1.393198E+00	5.363999E-06	1.00006978E+00
0	13	7.030465E-05	2.639960E-06	-2.196115E-06	NO	8.663072E-01	4.468065E-06	1.00006758E+00
0	14	5.758577E-05	2.348821E-06	-2.026048E-06	NO	8.758791E-01	3.908923E-06	1.00006556E+00
0	15	4.951282E-05	2.108488E-06	-1.837181E-06	NO	9.054297E-01	3.566990E-06	1.00006372E+00
0	16	4.752941E-05	1.907095E-06	-1.685608E-06	NO	9.157423E-01	3.276356E-06	1.00006203E+00
0	17	4.502637E-05	1.733876E-06	-1.559478E-06	NO	9.219312E-01	3.023241E-06	1.00006047E+00
0	18	4.225362E-05	1.585717E-06	-1.453385E-06	YES	9.276983E-01	2.802317E-06	1.00005902E+00
0	19	6.331069E-05	4.556405E-06	-2.114648E-05	NO	6.681148E+00	2.792290E-06	1.00003787E+00
0	20	3.742398E-05	1.094274E-06	1.745596E-06	NO	1.686769E-01	2.698594E-06	1.00003962E+00
0	21	3.487733E-05	8.225915E-07	-1.008528E-07	NO	1.774255E-01	3.959557E-07	1.00003952E+00
		OUTER ITERATION WITH FULL MATRIX SWEEP IS PERFORMED						
0	22	5.509557E-05	1.337599E-06	-5.770602E-07	NO	2.165325E+00	6.910751E-07	1.00003894E+00
		MAX ERROR ON POINT FISSION SOURCE OCCURED AT NODE 573 ON PLANE 15						
0		OUTER ITERATIONS COMPLETED AT ITERATION 22, ITERATIONS HAVE CONVERGED						
0		K-EFFECTIVE = 1.00003894069						

Figure 7.3. Example DIF3D 10.0 Outer Iteration Information with Acceleration

=VARIANT	OUTER	TIME(s)	Ups	K-effective	Error	pGS	Fis Max ,	RMS	TCheby	Vecs	Dom	P0	Src(s)	j+/(s)	Phi(s)
=VARIANT	1	29.2	001	5.863153E-01	7.1E-01	T	8.13E+00, 4.15E-01		T	0	0.41	0.41	1.5	22.0	5.6
=VARIANT	2	29.1	001	7.669153E-01	2.4E-01	T	2.19E+02, 8.71E-01		T	0	2.08	2.10	1.5	21.9	5.6
=VARIANT	3	29.1	001	9.290477E-01	1.7E-01	T	1.43E+03, 3.71E-01		T	0	0.78	0.78	1.5	21.9	5.6
=VARIANT	4	29.1	001	9.823526E-01	5.4E-02	T	4.65E+01, 9.95E-02		T	0	0.37	0.36	1.5	21.9	5.6
=VARIANT	5	29.1	001	9.938115E-01	1.2E-02	T	1.27E+00, 2.84E-02		T	0	0.31	0.31	1.5	21.9	5.6
=VARIANT	6	29.1	001	9.971656E-01	3.4E-03	T	3.73E-01, 1.17E-02		T	0	0.42	0.42	1.5	21.9	5.6
=VARIANT	7	29.1	001	9.989715E-01	1.8E-03	T	6.95E-02, 4.98E-03		T	0	0.43	0.43	1.5	21.9	5.6
=VARIANT	8	29.1	001	9.997894E-01	8.2E-04	T	3.05E-02, 1.88E-03		T	0	0.38	0.38	1.5	21.9	5.6
=VARIANT	9	29.1	001	1.000046E+00	2.6E-04	T	1.59E-02, 6.23E-04		T	0	0.33	0.33	1.5	21.9	5.6
=VARIANT	10	29.1	001	1.000094E+00	4.8E-05	T	8.47E-03, 1.99E-04		T	0	0.32	0.31	1.5	21.9	5.6
=VARIANT	11	29.1	001	1.000090E+00	4.0E-06	T	4.46E-03, 7.60E-05		T	0	0.38	0.38	1.5	21.9	5.6
=VARIANT	12	29.1	001	1.000080E+00	9.6E-06	T	2.43E-03, 3.64E-05		T	0	0.48	0.48	1.5	21.9	5.6
=VARIANT	13	29.1	001	1.000074E+00	6.7E-06	T	1.41E-03, 1.93E-05		T	0	0.53	0.52	1.5	21.9	5.6
=VARIANT	14	29.1	001	1.000069E+00	4.2E-06	T	8.85E-04, 1.14E-05		T	0	0.59	0.56	1.5	21.9	5.6
=VARIANT	15	29.1	001	1.000067E+00	2.9E-06	T	5.93E-04, 7.84E-06		T	0	0.69	0.64	1.5	21.9	5.6
=VARIANT	16	37.1	001	1.000062E+00	4.5E-06	F	7.10E-04, 1.10E-05		T	0	1.40	1.23	1.5	29.9	5.6
=VARIANT	17	37.1	001	1.000059E+00	3.4E-06	F	4.77E-04, 9.35E-06		T	0	0.85	0.86	1.5	29.9	5.6
=VARIANT	18	37.1	001	1.000056E+00	2.6E-06	F	3.38E-04, 7.89E-06		T	0	0.84	0.83	1.5	29.9	5.6
=VARIANT	19	37.1	001	1.000054E+00	2.3E-06	F	2.54E-04, 6.84E-06		T	0	0.87	0.87	1.5	29.9	5.6
=VARIANT	20	37.1	001	1.000052E+00	2.1E-06	F	1.98E-04, 6.00E-06		T	0	0.88	0.88	1.5	29.9	5.6
=VARIANT	21	45.3	001	1.000049E+00	2.9E-06	F	3.94E-04, 7.44E-06		T	1	1.24	1.21	1.5	38.2	5.6
=VARIANT	22	45.3	001	1.000044E+00	4.6E-06	F	2.71E-05, 3.42E-06		T	0	0.46	0.62	1.5	38.1	5.6
=VARIANT	23	45.2	001	1.000044E+00	2.0E-07	F	1.58E-04, 4.30E-06		T	1	1.26	0.78	1.5	38.1	5.6
=VARIANT	24	45.2	001	1.000042E+00	1.9E-06	F	6.94E-05, 3.75E-06		T	0	0.87	1.21	1.5	38.0	5.6
=VARIANT	25	45.0	001	1.000040E+00	2.0E-06	F	1.42E-04, 4.19E-06		T	1	1.12	1.05	1.5	37.9	5.6
=VARIANT	26	44.9	001	1.000038E+00	2.9E-06	F	1.31E-05, 1.95E-06		T	0	0.46	0.57	1.5	37.7	5.6
=VARIANT	27	44.7	001	1.000038E+00	2.0E-07	F	6.67E-05, 2.42E-06		T	1	1.25	0.79	1.5	37.5	5.6

OUTER ITERATIONS COMPLETED AT ITERATION 27, ITERATIONS HAVE CONVERGED
K-EFFECTIVE = 1.00003773

Figure 7.4. Example DIF3D 11.0 Outer Iteration Information

=VARIANT	OUTER	TIME(s)	Ups	K-effective	Error	pGS	Fis Max	RMS	Cheby	Vecs	Dom	Time:	Source	Current	Flux
=VARIANT	1	455.6	001	6.423414E-01	5.6E-01	T	4.07E+00	4.07E-01	F	0	0.41		240.6	57.8	157.2
=VARIANT	2	455.7	001	8.096865E-01	2.1E-01	T	2.07E+00	5.61E-01	F	0	1.15		240.7	57.7	157.2
=VARIANT	3	455.6	001	9.127771E-01	1.1E-01	T	2.55E+00	2.53E-01	F	0	0.68		240.7	57.7	157.1
=VARIANT	4	455.5	001	9.609669E-01	5.0E-02	T	2.33E+00	1.12E-01	F	0	0.55		240.6	57.7	157.2
=VARIANT	5	455.6	001	9.819601E-01	2.1E-02	T	1.65E+00	5.37E-02	F	0	0.53		240.7	57.7	157.2
=VARIANT	6	455.6	001	9.915092E-01	9.6E-03	T	8.77E-01	2.80E-02	F	0	0.54		240.7	57.7	157.1
=VARIANT	7	455.6	001	9.962158E-01	4.7E-03	T	3.51E-01	1.54E-02	F	0	0.56		240.6	57.8	157.2
=VARIANT	8	455.7	001	9.987147E-01	2.5E-03	T	1.29E-01	8.78E-03	F	0	0.58		240.7	57.8	157.2
=VARIANT	9	455.6	001	1.000113E+00	1.4E-03	T	5.13E-02	5.09E-03	F	0	0.58		240.7	57.7	157.1
=VARIANT	10	455.6	001	1.000920E+00	8.1E-04	T	2.37E-02	2.98E-03	F	0	0.59		240.6	57.7	157.2
=VARIANT	11	455.7	001	1.001393E+00	4.7E-04	T	1.21E-02	1.76E-03	F	0	0.59		240.7	57.8	157.2
=VARIANT	12	455.6	001	1.001672E+00	2.8E-04	T	6.63E-03	1.05E-03	F	0	0.59		240.7	57.7	157.1
=VARIANT	13	455.6	001	1.001836E+00	1.6E-04	T	3.81E-03	6.27E-04	F	0	0.60		240.6	57.7	157.2
=VARIANT	14	455.7	001	1.001933E+00	9.7E-05	T	2.37E-03	3.83E-04	F	0	0.61		240.7	57.8	157.2
=VARIANT	15	455.6	001	1.001990E+00	5.7E-05	T	1.50E-03	2.40E-04	F	0	0.63		240.7	57.7	157.1
=VARIANT	16	455.5	001	1.002023E+00	3.3E-05	T	9.70E-04	1.57E-04	F	0	0.65		240.6	57.7	157.2
=VARIANT	17	455.6	001	1.002043E+00	2.0E-05	T	6.50E-04	1.08E-04	F	0	0.69		240.7	57.7	157.2
=VARIANT	18	455.6	001	1.002055E+00	1.2E-05	T	4.39E-04	7.88E-05	F	0	0.73		240.7	57.8	157.1
=VARIANT	19	461.0	001	1.002061E+00	6.7E-06	F	3.02E-04	5.89E-05	F	0	0.75		240.6	63.2	157.2
=VARIANT	20	461.0	001	1.002065E+00	4.1E-06	F	2.16E-04	4.60E-05	F	0	0.78		240.7	63.1	157.2
=VARIANT	21	460.9	001	1.002068E+00	2.4E-06	F	1.64E-04	3.72E-05	F	0	0.81		240.7	63.1	157.1
=VARIANT	22	461.0	001	1.002069E+00	1.5E-06	F	1.76E-04	3.04E-05	F	0	0.82		240.6	63.1	157.2
=VARIANT	23	461.0	001	1.002070E+00	9.3E-07	F	1.91E-04	2.50E-05	F	0	0.82		240.7	63.1	157.2
=VARIANT	24	460.9	001	1.002071E+00	5.8E-07	F	2.10E-04	2.06E-05	F	0	0.83		240.7	63.1	157.1
=VARIANT	25	460.9	001	1.002071E+00	3.7E-07	F	2.34E-04	1.71E-05	F	0	0.83		240.6	63.1	157.2
=VARIANT	26	461.0	001	1.002071E+00	2.4E-07	F	2.65E-04	1.41E-05	F	0	0.83		240.7	63.1	157.2
=VARIANT	27	461.0	001	1.002072E+00	1.5E-07	F	3.04E-04	1.17E-05	F	0	0.83		240.7	63.1	157.1
=VARIANT	28	460.9	001	1.002072E+00	1.0E-07	F	3.53E-04	9.70E-06	F	0	0.83		240.6	63.1	157.2
=VARIANT	29	461.0	001	1.002072E+00	7.0E-08	F	4.13E-04	8.04E-06	F	0	0.83		240.7	63.2	157.2
=VARIANT	30	461.0	001	1.002072E+00	5.0E-08	F	4.88E-04	6.67E-06	F	0	0.83		240.7	63.1	157.1
=VARIANT	31	461.0	001	1.002072E+00	3.8E-08	F	5.80E-04	5.53E-06	F	0	0.83		240.6	63.2	157.2
=VARIANT	32	461.1	001	1.002072E+00	3.1E-08	F	6.95E-04	4.59E-06	F	0	0.83		240.7	63.2	157.2
=VARIANT	33	461.0	001	1.002072E+00	2.7E-08	F	8.39E-04	3.81E-06	F	0	0.83		240.7	63.2	157.1
=VARIANT	34	461.0	001	1.002072E+00	2.5E-08	F	1.02E-03	3.17E-06	F	0	0.83		240.6	63.2	157.2
=VARIANT	35	461.1	001	1.002072E+00	2.5E-08	F	1.25E-03	2.63E-06	F	0	0.83		240.7	63.2	157.2
=VARIANT	36	461.0	001	1.002072E+00	2.7E-08	F	1.53E-03	2.20E-06	F	0	0.84		240.7	63.2	157.1
=VARIANT	37	461.0	001	1.002072E+00	2.9E-08	F	1.90E-03	1.85E-06	F	0	0.84		240.6	63.2	157.2
=VARIANT	38	461.1	001	1.002072E+00	3.2E-08	F	2.37E-03	1.58E-06	F	0	0.85		240.7	63.2	157.2
=VARIANT	39	461.1	001	1.002072E+00	3.6E-08	F	2.97E-03	1.39E-06	F	0	0.88		240.8	63.2	157.1
=VARIANT	40	461.2	001	1.002072E+00	4.1E-08	F	3.74E-03	1.29E-06	F	0	0.93		240.7	63.2	157.3

...

Figure 7.5. Example DIF3D 11.0 Optical Thickness Convergence Problem

7.3 Inclusion of an external P_N based fixed source file

The conventional DIF3D code will build a FIXSRC file given rudimentary input. The file generated by input only allows an isotropic fixed source although the file format itself does allow a spatially dependent source consistent with the VARIANT spatial approximation. To enable the PERSENT sensitivity calculations [25], a new anisotropic fixed source file called VARSRC was added. The format of the binary VARSRC file (Appendix A) is nearly identical to that of NHFLUX_VARIANT and contains both even and odd angular nodal source moments. DIF3D-11 has been modified to check for presence of this new file and use it with priority over FIXSRC.

7.4 Anisotropic scattering bug in SP_N

Another important code bug found in VARIANT is the inaccurate treatment of anisotropic scattering with the SP_N approximation. Although the theory derived was correct, the implementation into the coding improperly defined the odd parity within- group scattering term. In the new coding the problem has been fixed. Given the new approach to generating angular trial functions and the corrections to the notation, the evaluation of the angular integrals are now verified to work properly.

7.5 Errors with “Omega” Acceleration

Over the course of the last ten years of getting users to transition from version 8.0 to version 11.0, we have noticed several outstanding problems with the VARIANT coding. Most important is that the forward and adjoint are occasionally not always consistent. In many cases these errors are easy to reproduce.

Starting with the ω acceleration problem, the discrete matrix-vector equations for VARIANT are typically written on a node-wise basis as:

$$\underline{\psi} = \underline{H} \cdot \underline{s} - \underline{C}^T \cdot \underline{\chi} \quad (219)$$

$$\underline{\varphi} = \underline{C} \cdot \underline{s} - \underline{G} \cdot \underline{\chi} \quad (220)$$

In the conventional formulation, a change of variable is made using partial currents

$$\begin{aligned} \underline{j}^+ &= \frac{1}{4}\underline{\varphi} + \frac{1}{2}\underline{\chi} & \underline{j}^- &= \frac{1}{4}\underline{\varphi} - \frac{1}{2}\underline{\chi} \\ \underline{\varphi} &= 2(\underline{j}^+ + \underline{j}^-) & \underline{\chi} &= \underline{j}^+ - \underline{j}^- \end{aligned} \quad (221)$$

which results in the new governing equations

$$\underline{\psi} = \underline{H} \cdot \underline{s} - \underline{C}^T [\underline{j}^+ - \underline{j}^-] \quad (222)$$

$$\underline{j}^+ = \underline{B} \cdot \underline{s} + \underline{R} \cdot \underline{j}^- \quad (223)$$

The new matrices are found to be

$$\underline{B} = \frac{1}{2} [\frac{1}{2}\underline{G} + \underline{I}]^{-1} \underline{C} \quad (224)$$

$$\underline{R} = [\frac{1}{2}\underline{G} + \underline{I}]^{-1} [\frac{1}{2}\underline{G} - \underline{I}] = \underline{I} - 2[\frac{1}{2}\underline{G} + \underline{I}]^{-1} \quad (225)$$

The “omega” acceleration applied in VARIANT, but in a consistent manner with successive over-relaxation (SOR) makes the fundamental change

$$\begin{aligned}\underline{\hat{j}}^+ &= \frac{1-\omega}{4} \underline{\varphi} + \frac{1+\omega}{2} \underline{\chi} & \underline{\hat{j}}^- &= \frac{1-\omega}{4} \underline{\varphi} - \frac{1+\omega}{2} \underline{\chi} \\ \underline{\varphi} &= \frac{2}{1-\omega} (\underline{\hat{j}}^+ + \underline{\hat{j}}^-) & \underline{\chi} &= \frac{1}{1+\omega} (\underline{\hat{j}}^+ - \underline{\hat{j}}^-)\end{aligned}\quad (226)$$

which leads to the transformation equations between the two definitions of the partial currents as

$$\begin{aligned}\underline{\hat{j}}^+ &= \underline{j}^+ - \omega \underline{j}^- & \underline{\hat{j}}^- &= -\omega \underline{j}^+ + \underline{j}^- \\ \underline{j}^+ &= \frac{\underline{\hat{j}}^+ + \omega \underline{\hat{j}}^-}{(1-\omega)(1+\omega)} & \underline{j}^- &= \frac{\omega \underline{\hat{j}}^+ + \underline{\hat{j}}^-}{(1-\omega)(1+\omega)}\end{aligned}\quad (227)$$

Substituting in the alternative definitions of equation 227 into equations 219 and 220 we obtain the alternate iterative equations and matrices:

$$\underline{\psi} = \underline{H} \cdot \underline{s} - \underline{C}^T \frac{1}{1+\omega} (\underline{\hat{j}}^+ - \underline{\hat{j}}^-) \quad (228)$$

$$\underline{\hat{j}}^+ = \underline{B} \cdot \underline{s} + \underline{R} \cdot \underline{\hat{j}}^- \quad (229)$$

$$\underline{B} = \frac{1-\omega}{2} \left[\frac{1}{2} \frac{1-\omega}{1+\omega} \underline{G} + \underline{I} \right]^{-1} \underline{C} \quad (230)$$

$$\underline{R} = \left[\frac{1}{2} \frac{1-\omega}{1+\omega} \underline{G} + \underline{I} \right]^{-1} \left[\frac{1}{2} \frac{1-\omega}{1+\omega} \underline{G} - \underline{I} \right] = \underline{I} - 2 \left[\frac{1}{2} \frac{1-\omega}{1+\omega} \underline{G} + \underline{I} \right]^{-1} \quad (231)$$

In VARIANT, three quantities are computed and stored because of the repetition in their usage in the above equations

$$O1 = 1 - \omega \quad O2 = \frac{1}{1 + \omega} \quad O3 = \frac{1}{(1 - \omega)(1 + \omega)} \quad (232)$$

In the current implementation, VARIANT only applies these factors to the flat moments leading to numerous differences in the above equations with regard to a standard SOR implementation.

Given that no bug was present, the only rational explanation was a mistake in the formulation of the acceleration. While the acceleration is very effective at reducing the computational effort required to get convergence, the fact that it converges to the wrong solution is quite problematic. Equation 233 shows the conventional Gauss-Seidel and SOR applied to a given matrix vector system.

$$\begin{aligned}Ax &= (L + D + U)x = b \\ \text{Gauss - Siedel} & \quad (D + L)x^{k+1} = b - Ux^k \\ \text{SOR} & \quad (D + \omega L)x^{k+1} = \omega b - (\omega U + (\omega - 1)D)x^k\end{aligned}\quad (233)$$

After studying the VARIANT implementation, it seems likely that the reason for the problem is that the ω acceleration was only applied to the flat moments. With regard the SOR in equation 233, this would be equivalent to applying the acceleration to only a few select rows rather than the entire system. This is obviously not the intention of SOR and this likely alters the diagonal

dominance of the system in some manner. If that is the case, then some of the non-accelerated moments (higher order) may not converge, but oscillate in some manner. If those other moments are key to defining the accuracy of the flat moments, then it could explain why the code converges to the incorrect answer as the convergence criteria in the code only considers the flat moments. While we believe this is the source of the errors observed in what follows, we have not been able to verify it thus far.

To display the issue with ω acceleration, we constructed a series of test problems, the details of which are not entirely relevant, and tabulate the forward and adjoint eigenvalues results in Table 7.4. Note that we consider the case without ω acceleration (Gauss-Seidel) to be the reference solution and that the 10.0 version infers the approach before 11.0. We also note that the convergence criteria was set to 10^{-8} in all quantities and all possible means to eliminate the error (no partitioned matrix, full convergence of inners, etc...) was attempted clearly indicating the ω acceleration as the outstanding problem in many of the test problems. We also note that while the error appears to be zero for several cases, it is in fact never exactly zero unless specifically using Gauss-Seidel.

To begin, in almost all of the problems, the error between the forward and adjoint calculation in Gauss-Seidel, termed “self error,” is within the targeted error criteria. The only exceptions are 120 periodic hexagonal geometry (Hex-120) cases where there is an observable eigenvalue error between the forward and adjoint (self error). This was traced to another bug which is discussed in the following section.

Looking at the errors in the production calculations in Table 7.4, there are clearly identifiable issues with all geometry options when the ω acceleration is enabled leading to 1-2 pcm of error. There is no error on a few problems which can be explained mostly by the fact that those problems are particularly large diffusive reactor problems. The two “restricted” cases infer that the values of ω in equation 228 are restricted to be within the cited range where $\omega=1$ is conventional Gauss-Seidel. Starting with the first restricted case, one can clearly see there is an impact on the computed results noting that the error is reduced by nearly an order of magnitude on most problems. Restricting it further to 0.8 almost eliminates all of the eigenvalue errors back to the base Gauss-Seidel scheme. We obviously tested out other ranges with 0.8 being the most consistent at reducing the error to 0.1 pcm or less. Note that switching the Hex-120 problems typically eliminates all of the errors which is clearly the case with tests 8-10 noting that there were some changes in geometry that result when the full core model is collapsed into 60 and 120 periodic cases.

There is an impact of restricting the value of ω as we are effectively reducing the acceleration and potentially increasing the time required to converge the problem. In all of the calculations in Table 7.4, the number of inners was fixed to be the same for all outer iterations. A simple adjustment routine was added to the code which detects a high dominance ratio (potentially caused by a lack of inner iteration convergence) and attempts to improve the convergence rate by increasing the number of inners. In all of the calculations above except for version 10.0, this adjustment feature was used to guarantee convergence, but was found to be rarely needed. This is consistent with the previous section results indicating that ω acceleration was not making much of a difference in the solution.

Table 7.4. Forward and Adjoint Eigenvalues and PCM Errors for the Test Problems

Test Problem Geometry		Reference Gauss-Seidel		Self Error	Version 10.0	Restrict 0.5 - 1.0	Restrict 0.8 - 1.0			
					Error (pcm)					
		Forward	Adjoint		F	A	F	A	F	A
1	Hex-60 2D small	1.303905	1.303905	0.0	1.2	1.4	0.0	0.2	0.0	0.0
2	Hex-120 2D small	1.303905	1.303905	0.0	1.2	1.4	0.0	0.2	0.0	0.0
3	Hex-60 3D small	1.058513	1.058513	0.0	0.5	0.7	-0.1	0.0	0.0	0.0
4	Hex-120 3D small	1.058513	1.058513	0.0	0.5	0.7	-0.1	0.0	0.0	0.0
5	Hex-60 3D large	1.030034	1.030034	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	Hex-120 3D large	1.019692	1.019605	8.8	3.1	0.9	3.0	0.9	1.6	0.7
7	Hex-60 3D Blanket	0.921557	0.921557	0.0	0.9	0.3	0.9	0.3	0.1	0.0
8	Hex-60 3D channel	1.045978	1.045978	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	Hex-120 3D channel	1.023500	1.023561	-6.0	-0.7	0.4	-0.7	0.4	-0.4	0.0
10	Hex-360 3D channel	1.053337	1.053337	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	Cart 2D tiny	0.969954	0.969954	0.0	1.9	1.1	1.8	1.0	0.1	0.1
12	Cart 2D small	1.303469	1.303469	0.0	3.5	3.8	0.5	0.6	0.0	0.1
13	Cart 3D small	1.058165	1.058165	0.0	2.1	2.3	0.1	0.2	0.0	0.0
14	Cart 3D large	1.016609	1.016609	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 7.5 tabulates the total number of outer iterations required for convergence of the forward and adjoint reference Gauss-Siedel calculation for each test problem. The net change in the number of outers from each reference point are then given for version 10.0 and the two restricted ω cases where negative numbers indicate a gain in performance and positive numbers indicate a loss in performance. In Table 7.5, it should be clear that restricting ω to 0.8 results in a near negligible impact of using the acceleration. The only problems that see any benefit from the acceleration are problems 3, 12, 13 & 14 where the acceleration actually reduces the amount of work required rather than increase it. Relative to version 10.0, one can see there is a substantial impact on problems 3 & 12-14 where the acceleration cuts the overall computational time in nearly half for problems 3 & 13. Of course, the nearly 1 pcm error observed in Table 7.5 on these benchmarks is generally undesirable given that the reference results are identical.

Using Table 7.5 to assess performance, the ω acceleration is either highly effective at getting VARIANT to converge quickly to either the correct solution (just problem 14) or the incorrect solution (problems 1-3 & 12-13) or completely ineffective (problems 2 & 4-11). Given that a majority of the uses for VARIANT target reactor problems similar to tests 6-10 & 14 we can conclude that the acceleration is only useful on benchmark 14 which requires significant flux solution analysis to ensure that the ω acceleration does not have a significant impact on the full solution (i.e. the eigenvalue isn't the only answer to the problem). We note that the adjustment routine did not make any adjustments as the dominance ratio was found to be 0.84 with and without the acceleration applied and the algorithm only makes an adjustment if the dominance ratio is 0.85 or higher.

Table 7.5. Outer Iteration Count for the Test Problems

Test Problem Geometry		Reference Gauss-Seidel		Version 10.0		Restrict 0.5-1.0		Restrict 0.8-1.0	
				Change in Outers					
		F	A	F	A	F	A	F	A
1	Hex-60 2D small	28	37	-1	-10	-1	-8	-1	-5
2	Hex-120 2D small	35	43	0	-8	1	-2	0	0
3	Hex-60 3D small	34	35	-17	-18	-16	-17	-7	-7
4	Hex-120 3D small	27	27	3	2	2	2	1	1
5	Hex-60 3D large	27	27	3	2	2	2	1	1
6	Hex-120 3D large	26	27	3	1	3	1	2	0
7	Hex-60 3D Blanket	103	102	-1	1	1	1	0	1
8	Hex-60 3D channel	36	36	3	3	3	3	1	1
9	Hex-120 3D channel	34	34	3	3	3	3	2	2
10	Hex-360 3D channel	73	74	3	2	3	2	1	1
11	Cart 2D tiny	39	39	1	0	0	0	0	0
12	Cart 2D small	42	67	-11	-25	-12	-31	-3	-11
13	Cart 3D small	43	63	-27	-40	-22	-29	-9	-12
14	Cart 3D large	98	85	-18	-3	-15	-3	-3	-6

After additional studies on these problems and other more practical problems, the ω acceleration value was set to 0.65 to strike a balance between performance and accuracy. In general, this can have an impact on both the eigenvalue and flux solution (the flux is typically impacted in the fourth or fifth significant digit). If this is perceived to be a problem with the desired calculations, then the user should disable the ω acceleration and rely upon Gauss-Seidel. While the adjustment routine can be relied upon in most cases to correct any under predictions in the necessary number of inners required to converge a given problem, the user can still set higher values as desired assuming that they will yield less computational effort. We note that this adjustment routine has rarely been observed to engage and is easy to notice as the time required to complete each outer iteration increases dramatically when it does.

7.6 Errors with Hexagonal 60 and 120 Periodic Geometries

As was shown in the previous section there clearly is a problem with hexagonal 120 periodic boundary conditions. A simple 3 group, 2D benchmark was created to magnify the errors and display the problem. In Figure 7.6 and Figure 7.7 we display the rotations of the hexagonal geometry and partial currents for 60 and 120 periodic cases, respectively. The left hand picture shows how the base coordinate system in each hex is rotated due to the periodic boundary condition. The right hand picture shows the orientation of each boundary current in each hex. In both geometries, the “ghost” hexes (represented using dashed boundaries) require the currents from the hexes along the periodic boundary to be remapped. For convenience, we have circled the currents that have to be remapped. Note that the boundary currents are the hexes that are in the solved portion of the domain where the color coding denotes which currents must be connected. A quick survey of both cases shows the remapping of the boundary currents can be summarized by Figure 7.8.

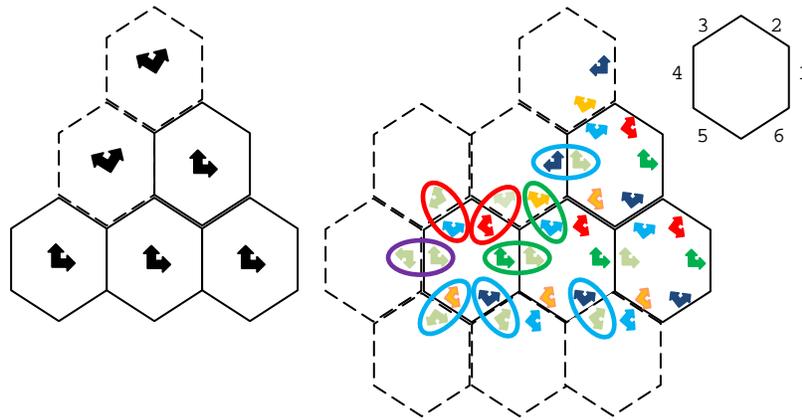


Figure 7.6. Hexagonal 60 Degree Periodic Geometry and Current Rotations

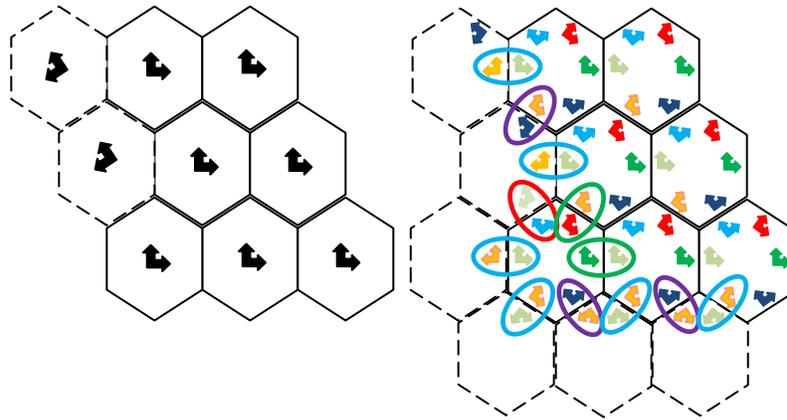


Figure 7.7. Hexagonal 120 Degree Periodic Geometry and Current Rotations

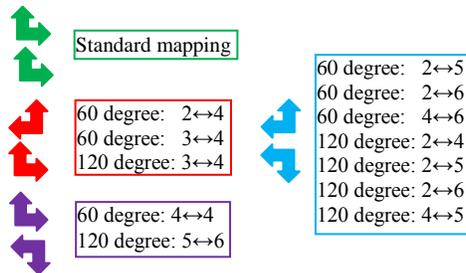


Figure 7.8. Four Current Rotations Necessary in Periodic Hexagonal Geometries

The basis expansion used on the interfaces in DIF3D-VARIANT can roughly be stated as

$$\chi_\gamma(\hat{r}, \hat{\Omega}) = \left\{ h_\gamma(\hat{r} \in \Gamma_\gamma) \otimes Y_\gamma^-(\hat{\Omega}) \right\} \tilde{\chi}_\gamma \quad (234)$$

$$\varphi_\gamma(\hat{r}, \hat{\Omega}) = \left\{ h_\gamma(\hat{r} \in \Gamma_\gamma) \otimes (\hat{\Omega} \cdot \hat{n}_\gamma) Y_\gamma^-(\hat{\Omega}) \right\} \tilde{\varphi}_\gamma,$$

where we are taking a bit of liberty with the definition of $\varphi_\gamma(\hat{r}, \hat{\Omega})$ for illustration purposes. The partial current vectors j_- and j_+ are defined to be:

$$j_\gamma^\pm = \frac{1}{4}\varphi \pm \frac{1}{2}\chi \quad (235)$$

One can expand the partial current notation assuming the space-angle approximations above and get:

$$\begin{aligned} j_\gamma^+(\hat{r}, \hat{\Omega}) &= \left\{ h_\gamma(\hat{r} \in \Gamma_\gamma) \otimes \left[\frac{1}{4}(\hat{\Omega} \cdot \hat{n}_\gamma) Y_\gamma^-(\hat{\Omega}) + \frac{1}{2} Y_\gamma^-(\hat{\Omega}) \right] \right\} j_\gamma^+ \\ j_\gamma^-(\hat{r}, \hat{\Omega}) &= \left\{ h_\gamma(\hat{r} \in \Gamma_\gamma) \otimes \left[\frac{1}{4}(\hat{\Omega} \cdot \hat{n}_\gamma) Y_\gamma^-(\hat{\Omega}) - \frac{1}{2} Y_\gamma^-(\hat{\Omega}) \right] \right\} j_\gamma^- \end{aligned} \quad (236)$$

Across a generic face, we equivalence j^+ to j^- which from equation 236, we can see the only significant impact will be the sign change in $(\hat{\Omega} \cdot \hat{n}_\gamma)$ and the specific change in the definition of $Y_\gamma^-(\hat{\Omega})$. In DIF3D-VARIANT, the $Y_\gamma^-(\hat{\Omega})$ and $h_\gamma(\hat{r} \in \Gamma_\gamma)$ trial functions are carefully made to be consistent across regular connecting interfaces as shown in Figure 7.9.

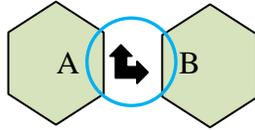


Figure 7.9. Example Current Alignment in Hexagonal Geometry

In the example of Figure 7.9, we find the current equivalence to require:

$$\begin{aligned} j_B^- &\equiv j_A^+ \\ j_B^-(\hat{r}, \hat{\Omega}) &\equiv j_A^+(\hat{r}, \hat{\Omega}) \\ -\frac{1}{4}\mu Y_B^-(\hat{\Omega}) - \frac{1}{2} Y_B^-(\hat{\Omega}) &\equiv \frac{1}{4}\mu Y_A^-(\hat{\Omega}) + \frac{1}{2} Y_A^-(\hat{\Omega}) \end{aligned} \quad (237)$$

In equation 237, the minus sign indicates the need to embed a sign change within the angular basis functions to get the necessary equivalence between the surface currents. This is not what is done in the production coding as the negative signs are simply embedded into the matrices. Looking at the production code, one finds that the negative sign appears by manipulating $\hat{\Omega} \cdot \hat{n}_\gamma$ on both the even-parity basis functions in equation 236 and the odd-parity boundary term in the transport equation. In the example given in equation 237, this minus sign is included for surfaces 4 (B in Figure 7.9), 5, and 6 of the hexagonal geometry shown in Figure 7.6. To expose this in our formalism above, we redefine the meaning of our basis as

$$\chi_\gamma(\hat{r}, \hat{\Omega}) = \left\{ h_\gamma(\hat{r} \in \Gamma_\gamma) \otimes c_\gamma Y_\gamma^-(\hat{\Omega}) \right\} \tilde{\chi}_\gamma \quad (238)$$

$$\varphi_\gamma(\hat{r}, \hat{\Omega}) = \left\{ h_\gamma(\hat{r} \in \Gamma_\gamma) \otimes \hat{\Omega} \cdot \hat{n}_\gamma c_\gamma Y_\gamma^-(\hat{\Omega}) \right\} \tilde{\varphi}_\gamma,$$

$$j_\gamma^\pm(\hat{r}, \hat{\Omega}) = \left\{ h_\gamma(\hat{r} \in \Gamma_\gamma) \otimes \left[\frac{1}{4} \hat{\Omega} \cdot \hat{n}_\gamma c_\gamma Y_\gamma^-(\hat{\Omega}) \pm \frac{1}{2} c_\gamma Y_\gamma^-(\hat{\Omega}) \right] \right\} j_\gamma^\pm. \quad (239)$$

Note that we have dropped the surface notation of the trial functions but have added in the surface dependent sign terms defined as

$$c = (1 \ 1 \ 1 \ -1 \ -1 \ -1)^T. \quad (240)$$

We can again impose the equivalence for Figure 7.8 and find

$$j_B^- \equiv j_A^+ \\ j_B^-(\hat{r}, \hat{\Omega}) \equiv j_A^+(\hat{r}, \hat{\Omega}) \quad (241)$$

$$\frac{1}{4}(-\mu)Y^-(\hat{\Omega}) - \frac{1}{2}Y^-(\hat{\Omega}) \equiv \frac{1}{4}(-1)\mu Y^-(\hat{\Omega}) + \frac{1}{2}(-1)Y^-(\hat{\Omega})$$

At this point we can now consider how the remapping will impact the transfer of data in Figure 7.9.

From Figure 7.8, it should be clear that we need to apply at most three mapping schemes. On each surface, we assume a local coordinate system where the s direction always travels along the surface and the t direction either into or out of the surface as seen in Figure 7.6 and Figure 7.7 (arrows applied near each surface internal to a given hexagon). The remaining coordinate on the surfaces is identical to the z direction in the conventional nodal coordinate system. The spherical harmonics are also re-oriented with the s - t - z coordinate system such that μ travels in the $-t$ direction and η travels in the $+s$ direction. Using that logic, we can define the mapping relationships needed to satisfy the periodic boundary conditions as:

$$R_A \left\{ \begin{array}{c} \leftarrow \uparrow \\ \rightarrow \uparrow \\ \rightarrow \downarrow \end{array} \right\} j^-(s, t, z, \mu, \eta, \zeta) = j^+(-s, t, z, \mu, -\eta, \zeta) \\ R_B \left\{ \begin{array}{c} \leftarrow \uparrow \\ \leftarrow \downarrow \\ \rightarrow \downarrow \end{array} \right\} j^-(s, t, z, \mu, \eta, \zeta) = j^+(s, -t, z, -\mu, \eta, \zeta) \quad (242) \\ R_C \left\{ \begin{array}{c} \leftarrow \uparrow \\ \leftarrow \downarrow \\ \rightarrow \downarrow \end{array} \right\} j^-(s, t, z, \mu, \eta, \zeta) = j^+(-s, -t, z, -\mu, -\eta, \zeta)$$

To investigate the impacts of each remapping, we use a linear approximation in space and P3 approximation in angle defined as

$$h_\gamma(\hat{r} \in \Gamma_\gamma) \square \begin{pmatrix} 1 \\ s \end{pmatrix} Y^-(\hat{\Omega}) \square \begin{pmatrix} \mu \\ \mu^3 - 3\mu \\ \eta(1 + 5\mu^2) \\ \mu(\eta^2 - \zeta^2) \end{pmatrix} \quad (243)$$

Note that equation 243 is just a simplistic representation of the current such that we have dropped the spherical harmonic normalization constants and that the spatial trial functions are not orthogonal. Expanding equation 239 into this basis and noting that μ points outward for surfaces 1, 2, 3 and inward for surfaces 4, 5, 6, we can write

$$j_\gamma^\pm(\hat{r}, \hat{\Omega}) = \left\{ c_\gamma \left(\frac{1}{4} c_\gamma |\mu| \pm \frac{1}{2} \right) \begin{pmatrix} 1 \\ s \end{pmatrix} \otimes \begin{pmatrix} \mu \\ \mu^3 - 3\mu \\ \eta(1 + 5\mu^2) \\ \mu(\eta^2 - \zeta^2) \end{pmatrix} \right\} j_\gamma^\pm \quad (244)$$

Starting with R_A , we have

$$\begin{aligned}
j_2^-(s, t, z, \mu, \eta, \zeta) &= j_4^+(-s, t, z, \mu, -\eta, \zeta) \\
\left\{ \left(\frac{1}{4}|\mu| - \frac{1}{2} \right) \begin{pmatrix} 1 \\ s \end{pmatrix} \otimes \begin{pmatrix} \mu \\ \mu^3 - 3\mu \\ \eta(1 + 5\mu^2) \\ \mu(\eta^2 - \zeta^2) \end{pmatrix} \right\} j_2^- &= \left\{ - \left(-\frac{1}{4}|\mu| + \frac{1}{2} \right) \begin{pmatrix} 1 \\ -s \end{pmatrix} \otimes \begin{pmatrix} \mu \\ \mu^3 - 3\mu \\ -\eta(1 + 5\mu^2) \\ \mu(\eta^2 - \zeta^2) \end{pmatrix} \right\} j_4^+ \quad (245) \\
j_2^- &= \left\{ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \end{pmatrix} \right\} j_4^+
\end{aligned}$$

As can be seen, we only need a simple Boolean vector manipulation where the signs are flipped for certain moments in the expansion. Note that the reverse mapping (i.e. 2→4) will require the same remapping array. Carrying out the same task with the other remapping cases, we find the Boolean results to be:

$$\begin{aligned}
R_A \rightarrow j_2^- &= \left\{ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \end{pmatrix} \right\} j_4^+ \\
R_B \rightarrow j_4^- &= \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \end{pmatrix} \right\} j_5^+ \quad (246) \\
R_C \rightarrow j_5^- &= \left\{ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \right\} j_6^+
\end{aligned}$$

VARIANT was originally written to only include a single current remapping function defined by NPCSEC which is applied to all surfaces that need remapping in Figure 7.6 and Figure 7.7. This was found to be incorrect by Won Sik Yang [24] and corrected to include two remapping arrays which fixed most of the problems, however, it was also shown to be incorrect using the simple 3 group test problem mentioned earlier (included as reference benchmark 34). The reason should be clear from Figure 7.6 and Figure 7.7 as there are clearly mapping errors.

While the error is easy to expose in a 120 hexagonal periodic problem, a similar error is observable in 60 degree periodic geometries so long as it has more than 4 rings and uses at least a linear spatial approximation. To display the rotation problems, we extracted the second active node (ring 2 position 1) and fourth active node (ring 2 position 3) incoming from a full core 3D calculation using a linear leakage, P3 approximation in Figure 7.10. In this case, spatial moments

for each angular moment are bunched together such that moments 1-3 correspond to the isotropic PN moments (there are a total of 6 angular moments). We color the surfaces identically with our earlier designation of A, B, and C and one should immediately see that the remapping in each case requires a sign change that is different. A check of the linear leakage, P5 solution also verified the need for three mapping schemes.

Node	Surface->	1	2	3	4	5	6
2	1	5.715064E+07	8.306824E+07	1.095894E+08	1.293469E+08	1.090745E+08	7.018622E+07
2	2	1.055077E+06	-6.428423E+06	-5.401662E+06	8.495812E+04	-6.563336E+06	-5.935705E+06
2	3	1.363168E+07	2.032730E+07	2.594975E+07	2.995970E+07	2.544448E+07	1.726000E+07
2	4	2.073701E+06	-1.762059E+06	1.596199E+06	4.376635E+06	2.801908E+06	-2.640681E+05
2	5	-2.519730E+02	-1.176633E+06	-4.828176E+05	5.900046E+04	-2.931406E+05	-8.257683E+05
2	6	8.011741E+05	-9.229898E+03	1.764764E+06	2.685410E+06	2.018768E+06	8.448416E+04
2	7	-6.080180E+05	1.392659E+07	1.381873E+07	5.511914E+05	-1.474966E+07	-1.250889E+07
2	8	2.470056E+06	3.936735E+06	1.622043E+06	-2.880375E+06	-3.079589E+06	-1.592088E+06
2	9	-1.867673E+05	3.079515E+06	2.839786E+06	1.582331E+05	-3.079977E+06	-2.724805E+06
2	10	-2.120513E+06	-8.791411E+05	4.043613E+06	5.063297E+06	3.467404E+06	6.257962E+05
2	11	-1.955425E+05	7.000833E+05	-2.129633E+05	-7.654658E+04	-2.167085E+05	3.212808E+05
2	12	-7.390437E+05	-2.711755E+05	2.206082E+06	3.035907E+06	2.211078E+06	7.735461E+04
2	13	1.649151E+07	2.283453E+07	2.983950E+07	-3.551280E+07	-3.006511E+07	-2.032239E+07
2	14	-4.603466E+04	-1.856091E+06	-1.919053E+06	-6.266193E+04	1.880145E+06	1.663515E+06
2	15	-1.140572E+07	-1.430829E+07	-1.820956E+07	2.047509E+07	1.824483E+07	1.218054E+07
2	16	2.354644E+05	-2.167619E+06	-2.188724E+06	-3.282121E+05	-2.612282E+06	-2.178942E+06
2	17	-5.427392E+05	-8.597242E+05	-3.431576E+05	-5.767496E+05	-4.652588E+05	-1.577543E+05
2	18	-1.150573E+05	1.168563E+06	1.292006E+06	9.798754E+04	1.434263E+06	1.023914E+06
Node	Surface->	1	2	3	4	5	6
4	1	1.090745E+08	7.018622E+07	5.715064E+07	8.306824E+07	1.095894E+08	1.293469E+08
4	2	-6.563336E+06	5.935705E+06	-1.055077E+06	-6.428423E+06	5.401662E+06	-8.495812E+04
4	3	2.544448E+07	1.726000E+07	1.363168E+07	2.032730E+07	2.594975E+07	2.995970E+07
4	4	2.801908E+06	-2.640681E+05	2.073701E+06	-1.762059E+06	1.596199E+06	4.376635E+06
4	5	-2.931406E+05	8.257683E+05	2.519730E+02	-1.176633E+06	4.828176E+05	-5.900046E+04
4	6	2.018768E+06	8.448416E+04	8.011741E+05	-9.229898E+03	1.764764E+06	2.685410E+06
4	7	1.474966E+07	-1.250889E+07	6.080180E+05	-1.392659E+07	1.381873E+07	-5.511914E+05
4	8	3.079589E+06	1.592088E+06	2.470056E+06	-3.936735E+06	-1.622043E+06	-2.880375E+06
4	9	3.079977E+06	-2.724805E+06	1.867673E+05	-3.079515E+06	2.839786E+06	-1.582331E+05
4	10	3.467404E+06	6.257962E+05	-2.120513E+06	-8.791411E+05	4.043613E+06	5.063297E+06
4	11	-2.167085E+05	-3.212808E+05	1.955425E+05	7.000833E+05	2.129633E+05	7.654658E+04
4	12	2.211078E+06	7.735461E+04	-7.390437E+05	-2.711755E+05	2.206082E+06	3.035907E+06
4	13	3.006511E+07	2.032239E+07	1.649151E+07	-2.283453E+07	-2.983950E+07	-3.551280E+07
4	14	-1.880145E+06	1.663515E+06	4.603466E+04	1.856091E+06	-1.919053E+06	6.266193E+04
4	15	-1.824483E+07	-1.218054E+07	-1.140572E+07	1.430829E+07	1.820956E+07	2.047509E+07
4	16	-2.612282E+06	2.178942E+06	-2.354644E+05	-2.167619E+06	2.188724E+06	3.282121E+05
4	17	-4.652588E+05	-1.577543E+05	-5.427392E+05	-8.597242E+05	-3.431576E+05	-5.767496E+05
4	18	1.434263E+06	-1.023914E+06	1.150573E+05	1.168563E+06	-1.292006E+06	-9.798754E+04

Figure 7.10. Full Core 3D Results Proving that the R_B and R_C Mappings are Necessary.

There were serious problem with making the necessary changes to DIF3D as the storage was fundamentally broken. The j^+ currents in DIF3D are implicitly defined with respect to the active node number and node surface such that we always have $Nodes \cdot Surfaces$ moments of storage in the problem. For nodes that lie upon the outer domain boundary, we must add additional storage to hold the moments of the size: $DomainSurfaces$. At this point we have no issues, especially for full core problems. For 30 degree symmetric problems, the current implementation numbers the global surface currents as seen in Figure 7.11 while for 60 degree periodic boundary conditions the numbering in Figure 7.12 is used.

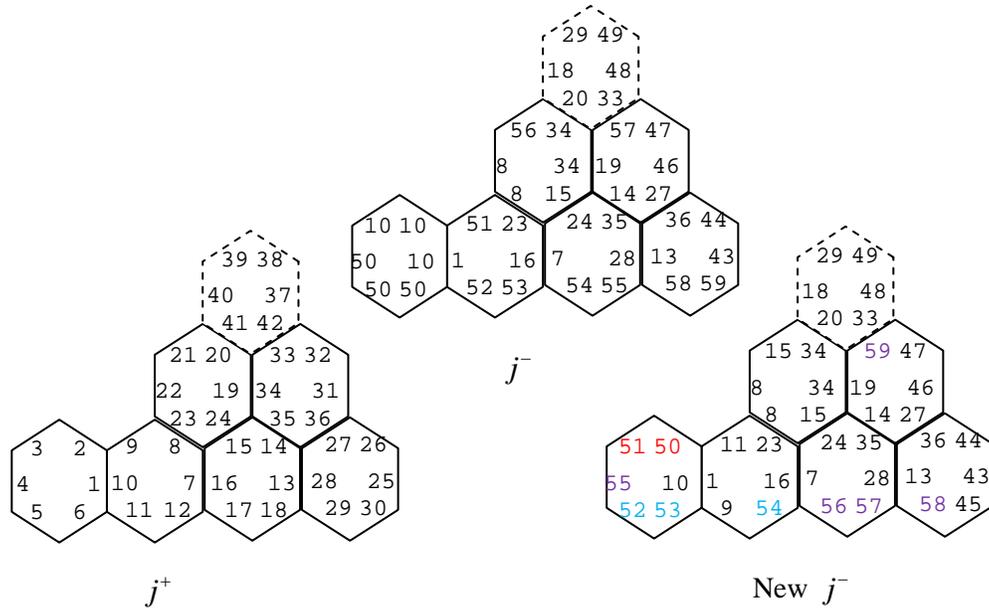


Figure 7.11. 30 Degree Symmetric Global Surface IDs for j^\pm Partial Currents.

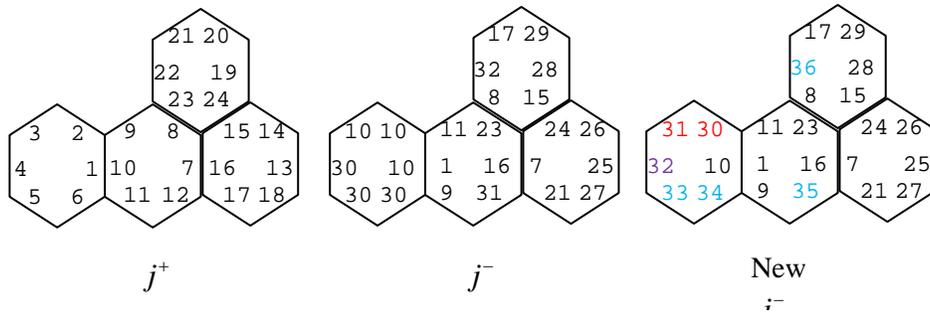


Figure 7.12. 60 Degree Periodic Global Surface IDs for j^\pm Partial Currents.

Starting with Figure 7.11, surfaces 1-24 are conventional outgoing partial currents where surfaces 25-29 are domain boundary conditions and surfaces 30-32 are periodic boundary conditions that need mapping. Of course the issue is that we now know there are an insufficient number of periodic boundary currents because of the necessary mappings. The 30 degree symmetric case is more complicated than the 60 degree periodic, but one can clearly see that the center hex again has insufficient storage. Note that in the 30 degree symmetry case, the last node in the fourth ring is eliminated due to symmetry, but its current storage is not. This is quite confusing in the existing NHFLUX file as the current moments are never copied where as the nodal flux moments are. Also note that in 30 degree symmetry, the moments that need the sign changes are typically eliminated due to the various rotations that must be preserved to impose the symmetry. Finally, the 59th current is improperly listed as a periodic boundary condition that was linked to the 44th surface thereby potentially corrupting the accuracy of the approximation as it should only require a vacuum boundary condition.

For completeness, the 120 degree periodic global surface current numbering is given in Figure 7.13. In this case we see surfaces 1-42 are conventional outgoing partial currents while

surfaces 43-52 are domain boundary conditions, and surfaces 53-58 are periodic boundary conditions that need the mappings.

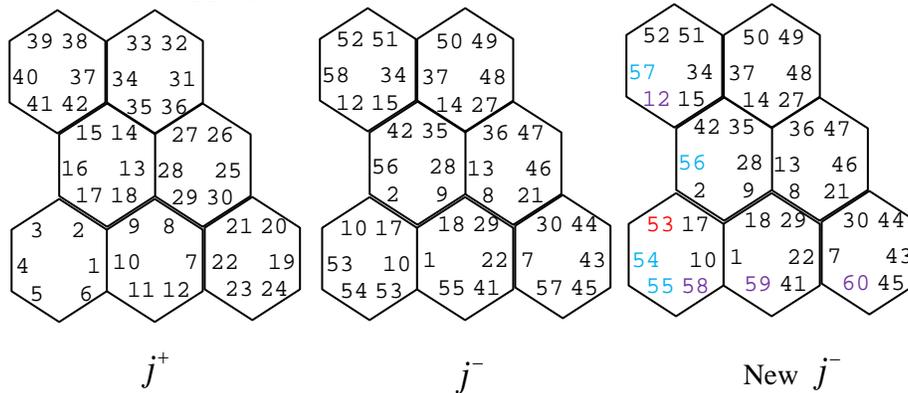


Figure 7.13. 120 Degree Periodic Global Surface IDs for j^\pm Partial Currents.

Based upon the mappings needed in Figure 7.6 and Figure 7.7, we should uniquely define the incoming currents as shown in Figure 7.11 through Figure 7.13 where the coloring indicates the type of rotation necessary. Because of the extra complexity in getting the 30 degree symmetry problem to work, we were not able to complete its implementation into the source code. Note that we were careful to order the periodic surfaces such that similar mappings are collected together.

8 Results

8.1 Introduction

With the above modifications to the VARIANT coding, we now consider the impact of the various changes. First, we note that DIF3D 11.0 was rigorously tested using numerous validation benchmarks, the complete details of which are not shown here for brevity. We also note that the following results do not include the alterations made to DIF3D 11.0 by fixing either the hexagonal periodic boundary conditions or the ω acceleration or adding Tchebychev acceleration. What is important is that those tests not only compare eigenvalue results, but also detailed flux and power distributions. Second, the convergence criteria used in all of the following calculations is well beyond what most users would apply normally converging the eigenvalue to 10^{-9} , the point wise flux to 10^{-7} and the RMS flux to 10^{-8} . As a consequence, one should not assume that the computational burden seen in this report would be common when they execute VARIANT (we can comfortably estimate the average user settings to require at most half of the time reported for any calculation in this report). Finally, to maintain simplicity, we only focus on the eigenvalue comparison in this report (assuming the flux solution matches) and choose to only consider three-dimensional benchmark problems for which the VARIANT code is primarily used. The benchmarks we give results for are the Takeda benchmarks [20], along with assembly homogenized models of the EBR-II reactor [21], the ABR reactor [22], and the MONJU reactor [23].

8.2 TAKEDA-2 Benchmark

The first benchmark considered is a three-dimensional Cartesian geometry benchmark problem specified by Takeda [20]. It provides two-group cross section data with an isotropic scattering kernel and focuses on two different configurations where a control rod position is varied. The details of the geometry can be found in the reference and are not repeated here for brevity, but it is worth mentioning that the problem employs 90 degree radial symmetry. In addition to this, we also only present results for the rodded configuration because the unrodded configuration exhibits a similar behavior. Table 8.1 gives the eigenvalue results obtained with the new and production versions of VARIANT for the unrodded configuration in addition to the performance data.

Table 8.1. Eigenvalue and Performance for the Takeda 2 Benchmark

	Version 9.0		Version 10.0		Version 10.0 & Alt PN	
	eigenvalue	Time (sec)	eigenvalue	Time (sec)	eigenvalue	Time (sec)
6,6,1 P1	0.96913	3.5	0.96913	5.5	0.96913	5.3
6,6,1 P3	0.97349	12	0.97349	13	0.97349	22
6,6,1 P5	0.97364	40	0.97364	41	0.97363	70
6,6,1 P7			0.97369	147		
6,6,1 P9			0.97371	352		
6,8,2 P5			0.97359	223		
6,8,3 P5			0.97358	457		
8,8,3 P5			0.97358	509		

As can be seen there are no significant difference in the eigenvalue results. Using the higher order capabilities of the new version we can also see that further refinement of the spatial flux, source, and leakage amount to a rather minor impact on the eigenvalue as does refinement of the angular approximation past P_5 . The use of the alternate set of trial functions proposed by Yang yielded identical answers to the old functions, but significantly more computational effort was required. This result is consistent with every benchmark problem we include in this report and all those we run but did not include and thus do not include any additional results in this report using those functions. As for an explanation, we suspect the new set of functions degrades the spectral radius of the within group equations and thus always require more computational effort than the original set proposed by Lewis. This does open the possibility that yet another set of functions will produce a spectral radius which is consistently less than the current set and thus improve the existing performance. Finally, there are relatively minor if not insignificant differences in the computational timings reported which is expected given the changes.

Overall we feel we should ignore any timing results for small scale problems such as Takeda unless more care is taken to run the machine under the exact same load and input every time. What this benchmark really tells us is that the new version of VARIANT produces solutions to this benchmark problem that are either consistent with the previous versions or superior to it in a comparable amount of time.

8.3 Takeda-4 Benchmark

The second benchmark considered is a three-dimensional Hexagonal geometry benchmark problem specified by Takeda [20]. It provides four-group cross section data with an isotropic scattering kernel and focuses on two different configurations where a control rod position is varied. The details of the geometry can be found in the reference and are not repeated here for brevity, but it is worth mentioning that the benchmark has 30 degree symmetry and thus only a 60 degree sector is used in the calculation. In addition to this, we only present results for the half-rodded configuration because the rodded and unrodded configurations exhibit a similar behavior to the half-rodded configuration. Table 8.2 gives the eigenvalue results obtained with the new and production versions of VARIANT for the half-rodded configuration in addition to the performance data.

Once again we can see identical results with respect to the eigenvalue solutions and little variability in the machine run times. We can also see the SP_N calculations are consistent between the new and old version, but neither produces the correct transport solutions. This type of behavior is consistent with other researchers who have used SP_N and there is no clear indicator of what the degree of inaccuracy will be when using SP_N . We did not include the results for the alternate set of trial functions since they did not provide any more useful information than that found in the previous benchmark. Similar to the Takeda-2 benchmark, we can conclude that the new version is producing results consistent with the old version and the performance data is similar.

Table 8.2. Eigenvalue and Performance for the Takeda-4 Benchmark

	Version 9.0		Version 10.0	
	eigenvalue	Time (sec)	eigenvalue	Time (sec)
6,6,1 P1	0.96063	2.0	0.96063	1.6
6,6,1 P3	0.98310	3.8	0.98310	3.5
6,6,1 P5	0.98530	16	0.98530	16
6,6,1 P7			0.98589	90
6,6,1 P9			0.98610	386
6,8,2 P5			0.98452	289
6,8,3 P5			0.98431	645
8,8,3 P5			0.98431	648
6,6,1 SP1	0.96063	2.0	0.96063	2.0
6,6,1 SP3	0.98153	1.6	0.98153	1.8
6,6,1 SP5	0.98283	2.0	0.98283	2.3
6,6,1 SP7			0.98313	2.8
6,6,1 SP9			0.98324	3.2

8.4 EBR-II Benchmark

The next benchmark we consider is an assembly homogenized model of the EBR-II benchmark [21]. Unlike the previous benchmarks, this one is much larger consisting of 5.6 times as many nodes as the Takeda 4 benchmark and it uses 9 groups instead of 4 groups. It also defines a P_1 anisotropic scattering kernel unlike the previous benchmarks which used isotropic scattering. Table 8.3 gives the eigenvalue and performance data for the EBR-II benchmark.

Table 8.3. Eigenvalue and Performance for the EBR-II Benchmark

	Version 9.0		Version 10.0	
	eigenvalue	Time (sec)	eigenvalue	Time (sec)
6,6,1 P1	1.00378	27	1.00378	12
6,6,1 P3	1.02614	172	1.02614	97
6,6,1 P5	1.02772	939	1.02772	247
6,6,1 P7			1.02821	596
6,6,1 P9			1.02841	1716
6,8,2 P5			1.02737	1086
6,8,3 P5			1.02728	2575
8,8,3 P5			1.02726	3956
6,6,1 SP1	1.00378	29	1.00378	12
6,6,1 SP3	1.02301	62	1.02301	24
6,6,1 SP5	1.02373	80	1.02373	29
6,6,1 SP7			1.02393	40
6,6,1 SP9			1.02402	44

Similar to the previous benchmarks, we again see no differences in the eigenvalue solutions. Note that both the spatial and angular approximations are more important in this benchmark and we should consider using either quadratic or cubic order interface conditions with P_7 , although the error in the eigenvalue is effectively cancelled out in this case (the flux solution might not be equivalent). Further refinement of the source is generally unimportant which is consistent with almost all VARIANT benchmarks where a 4th order source approximation is generally sufficient for eigenvalue convergence.

Unlike the previous benchmarks, we see a substantial and consistent difference in the computational performance which grows as the angular approximation increases. For the P_5 calculation in particular (highlighted in red), we were very careful to run the calculation in the same manner using the new and production versions of VARIANT. This behavior was expected and hoped for given our recent changes to improve the computational performance. The common variance observed during this effort was 10%. As discussed, the new version removes the bulk of the computational effort associated with the anisotropic scattering source operations. It is important to point out that we can expect some difference in the answers between the new and old version because of the changes in the spatial and angular matrices. The use (and elimination) of tolerance statements in the new version can also have an impact in addition to the removal of numerous hard wired stuff and the replacement of single precision math with double precision math. In the end, we believe the new version is more accurate and choose to ignore these differences for now. Only detailed benchmarking will be able to prove whether any version of VARIANT is accurate and we will continue in the future to guarantee both the accuracy and efficiency.

8.5 ABR Benchmark

One of the first benchmarks that the VARIANT code was found to be inadequate for is the ABR benchmark [22]. This benchmark problem is much larger than the EBR-II because it requires the solution of the full core (no symmetry) and has 2.9 times as many nodes as the EBR-II benchmark. The original researchers wanted to investigate the inaccuracies of the cross section data with respect to energy refinement. To do this one increases the number of energy groups which necessitates an improved model of the anisotropic scattering kernel. This type of analysis also requires that the solution is well converged with respect to both space and angle at all energy resolutions to avoid polluting the solution with differing amounts of error due to an inadequately refined space-angle approximation. In this particular case the researchers were attempting to run a 230 group calculation with a P_5 anisotropic scattering kernel. The existing VARIANT code was incapable of producing a solution after several days of computational effort.

Since we cannot run the 230 group problem with the production version and the researchers did not provide us with cross section data other than 33 group, we only consider the 33 group calculations which includes a P_3 anisotropic scattering kernel. Table 8.4 gives the eigenvalue and performance information for the ABR benchmark. We went ahead and included the SP_N results for completeness (i.e. to verify a P_3 anisotropic scattering kernel), but we note that the solutions are fundamentally unimportant.

Table 8.4. Eigenvalue and Performance for the ABR Benchmark

	Version 9.0		Version 10.0	
	eigenvalue	Time (min)	eigenvalue	Time (min)
6,6,1 P1	1.00805	7.2	1.00805	5.5
6,6,1 P3	1.01952	28	1.01962	15
6,6,1 P5	1.02057	161	1.02057	31
6,6,1 P7			1.02087	67
6,6,1 P9			1.02098	258
6,7,2 P5			1.02031	189
6,8,3 P5			1.02021	528
8,8,3 P5			1.02018	702
6,6,1 SP1	1.00805	7.3	1.00805	5.3
6,6,1 SP3	1.01583	8.9	1.01583	4.1
6,6,1 SP5	1.01626	12	1.01626	4.6
6,6,1 SP7			1.01637	5.2
6,6,1 SP9			1.01641	5.8

Once again the eigenvalue solutions are identical between the production and new versions. While this doesn't mean that the solutions are correct given that this work has produced a new version but also corrected the existing production version. The only reliable means to eliminate concerns is to compare the code against another transport code which treats the anisotropic scattering kernel correctly (i.e. not MCNP). Similar to the EBR-II benchmark, the ABR requires higher than a P₅ approximation to guarantee convergence of the eigenvalue. However, similar to the EBR-II benchmark, refining the spatial approximation appears to negate the impact of the refined angular approximation such that we can expect the fully converged eigenvalue to be near that of the P₅ solution using the indicated settings.

The most important thing to note is the improved computational performance of the new version. In the 9 group EBR-II benchmark we can discern a 74% reduction in the computational effort (factor of 3.8) for the P₅ calculation while we achieve an 80% reduction (factor of 5.2) for the 33 group ABR benchmark. This is consistent with our expectations given that the computational effort required to apply the *T* matrix multiplications increases as the number of energy groups increases (not linear because of the number of outer iterations required to converge). Unfortunately these timing numbers are somewhat incorrect because of the machine architecture. As an example, we consider the P₉ calculation. The process id time reported by DIF3D was 258 minutes, but the wall-clock time reported was 372 minutes. After some investigation we found this was the time required to read/write the flux records on the disk which is a constraint placed upon VARIANT by BPOINTER. This was corrected in DIF3D-11 by removing BPOINTER storage of all large arrays within VARIANT as discussed in Section 6.

8.6 MONJU Benchmark

The next benchmark we present results for is derived from the MONJU fast reactor in Japan [23]. Similar to the ABR benchmark, the researchers again wanted to investigate the inaccuracies of the cross section data with respect to energy refinement. The geometry has roughly the same number of nodes as the EBR-II benchmark and the researchers provided cross section data at 9 group, 33 group, 70 group, and 230 group all with a P_5 anisotropic scattering kernel. While some of the 70 group calculations could be executed with DIF3D 9.0 and 10.0, no significant results could be obtained above that resolution.

For brevity, we choose not to execute all of the necessary space-angle calculations. Instead, we just selected a few space-angle resolutions similar to that done in the previous benchmarks which indicate the importance of the various approximations that go into the VARIANT code. We also chose to use a simplified geometry and leave the desired geometry (~10 times larger than the simplified) for the follow on section. We solved all four energy resolutions using the selected space-angle approximations using the new version, but restricted the production version because of the overwhelming computational effort required. The eigenvalue and computational performance information for all four energy resolutions is provided in Tables 8.5 through 8.9. Note that the SP_N calculation option is not included because of its relative lack of importance.

Table 8.5. Eigenvalue and Performance for the 9g MONJU Simplified Benchmark

	Version 9.0		Version 11.0	
	eigenvalue	Time	eigenvalue	Time
6,6,1 P1 K0	0.99787	13 sec	0.99787	16 sec
6,6,1 P1 K1	0.99564	26 sec	0.99564	19 sec
6,6,1 P3 K3	1.00143	10 min	1.00143	1.7 min
6,6,1 P5 K0			1.00432	1.5 min
6,6,1 P5 K1			1.00083	2.5 min
6,6,1 P5 K3			1.00171	5.3 min
6,6,1 P5 K5	1.00172	2.4 hr	1.00172	6.4 min
6,6,1 P7 K5			1.00180	19 min
6,6,1 P9 K5			1.00182	34 min
6,7,2 P5 K5			1.00165	30 min
6,8,3 P5 K5			1.00162	52 min
8,8,3 P5 K5			1.00162	1.7 hr

Once again the eigenvalue solutions are identical between the new and production version. The computational times are again dramatically improved using the new version which is far beyond that observed in the previous benchmarks which is again the main purpose of the new work. Focusing on the P_5 approximation with a P_5 scattering kernel we have an 80% reduction at 9 groups (factor of 9.3), a 95% reduction at 33 groups (factor of 10.3), and a 99% reduction at 70 groups (Factor of 20.4). Similar to the previous calculations we see a dramatic reduction in the computational effort as the number of energy groups goes down. More importantly is that we do not see a large growth in the computational effort as the scattering

kernel size increases. Assuming that the factor of ~33 in computational performance observed in the 33 group and 70 group calculations holds, we can estimate that the 230 group calculation using DIF3D 9.0 would require about 33 days which is obviously not a practical exercise to perform at this point.

Table 8.6. Eigenvalue and Performance for the 33g MONJU Simplified Benchmark

	Version 9.0		Version 11.0	
	eigenvalue	Time	eigenvalue	Time
6,6,1 P1 K0	0.99726	22 sec	0.99726	24 sec
6,6,1 P1 K1	0.99512	1.9 min	0.99512	1.0 min
6,6,1 P3 K3	1.00114	1.3 hr	1.00114	8.9 min
6,6,1 P5 K0			1.00378	5.0 min
6,6,1 P5 K1			1.00059	10.0 min
6,6,1 P5 K3			1.00144	24 min
6,6,1 P5 K5	1.00144	16 hr	1.00144	31 min
6,6,1 P7 K5			1.00153	1.4 hr
6,6,1 P9 K5			1.00155	2.4 hr
6,7,2 P5 K5			1.00138	2.3 hr
6,8,3 P5 K5			1.00134	3.6 hr
8,8,3 P5 K5			1.00131	8.0 hr

Table 8.7. Eigenvalue and Performance for the 70g MONJU Simplified Benchmark

	Version 9.0		Version 11.0	
	eigenvalue	Time	eigenvalue	Time
6,6,1 P1 K0	0.99698	46 sec	0.99698	50 sec
6,6,1 P1 K1	0.99484	5.9 min	0.99484	3.1 min
6,6,1 P3 K3	1.00097	4.6 hr	1.00097	27 min
6,6,1 P5 K0			1.00355	11 min
6,6,1 P5 K1			1.00047	25 min
6,6,1 P5 K3			1.00129	1.1 hr
6,6,1 P5 K5	1.00130	53 hr	1.00130	1.5 hr
6,6,1 P7 K5			1.00139	4.6 hr
6,6,1 P9 K5			1.00143	6.6 hr
6,7,2 P5 K5			1.00123	7.8 hr
6,8,3 P5 K5			1.00119	9.6 hr
8,8,3 P5 K5			1.00105	20 h

Table 8.8. Eigenvalue and Performance for the 230g MONJU Simplified Benchmark

	Version 9.0		Version 11.0	
	eigenvalue	Time	eigenvalue	Time
4,6,1 P1 K0	0.99577	3.6 min	0.99577	4.4 min
4,6,1 P1 K1	0.99387	1.7 hr	0.99387	40 min
4,6,1 P3 K3	1.00009	83 hr	1.00009	8.2 hr
4,6,1 P5 K0			1.00245	37 min
4,6,1 P5 K1			0.99966	3.1 hr
4,6,1 P5 K3			1.00043	16 h
4,6,1 P5 K5			1.00043	24 h
4,6,1 P7 K5			1.00053	36 h
4,6,1 P9 K5			1.00057	43 h

8.7 Higher Order Scattering Kernel

The final benchmark we want to present results for is the only remaining concern left with VARIANT: higher order anisotropic scattering capabilities. In all of the previous calculations we only consider results for up to a P_5 scattering kernel because the production version of VARIANT had a P_5 flux limitation. We removed that restriction in the new version and thus can run scattering kernel approximations as high as the flux approximation. Unfortunately we could not come up with any meaningful benchmark problems because the memory requirements needed to use a P_7 or higher scattering kernel were beyond the capabilities of the BPOINTER limitations in DIF3D-10.0 and earlier. Most of our test problems focused on Cartesian and hexagonal geometries with maximal symmetry and the specific results are neglected here. To summarize them, P_7 anisotropic scattering isn't that important as evident from the preceding MONJU calculations which indicate at most 10 pcm can be gained between P_3 and P_5 .

Table 8.9. Eigenvalue and Performance for the MONJU Benchmark

	9 group		33 group		70 group	
	k-eff	Time	k-eff	Time	k-eff	Time
4,6,1 P1 K1	0.98786	1.5 min	0.98961	8.5 min	0.99002	26 min
4,6,1 P3 K1	0.99377	11 min	0.99638	1.3 hr	0.99686	4.6 hr
4,6,1 P3 K3	0.99454	17 min	0.99648	2.1 hr	0.99696	8.9 hr
4,6,1 P5 K1	0.99421	20 min	0.99702	1.5 hr	0.99751	4.7 hr
4,6,1 P5 K3	0.99514	43 min	0.99709	3.8 hr	0.99759	10.2 hr
4,6,1 P5 K5	0.99514	55 min	0.99710	5.7 hr	0.99759	14.8 hr
4,6,1 P7 K1	0.99433	50 min	0.99721	3.9 hr	0.99771	8.5 hr
4,6,1 P7 K3	0.99528	50 min	0.99726	7.0 hr	0.99777	15.8 hr
4,6,1 P7 K5	0.99530	1.4 hr	0.99727	13.1 hr	0.99777	35.0 hr
4,6,1 P9 K1	0.99438	3.1 hr	0.99729	12.5 hr	0.99780	25.6 hr
4,6,1 P9 K3	0.99534	3.6 hr	0.99733	15.0 hr	0.99784	35.4 hr
4,6,1 P9 K5	0.99536	5.5 hr	*	*	*	*
	116 group		230 group			
	k-eff	Time	k-eff	Time		
4,6,1 P1 K1	0.98961	2.6 hr	0.98939	12.1 hr		
4,6,1 P3 K1	0.99650	14.8 hr	0.99629	47.7 hr		
4,6,1 P3 K3	0.99661	28.6 hr	0.99640	85.5 hr		
4,6,1 P5 K1	0.99716	10.3 hr	0.99695	44.2 hr		
4,6,1 P5 K3	0.99724	30.6 hr	0.99703	106.1 hr		
4,6,1 P5 K5	0.99725	36.4 hr	0.99704	164.5 hr		
4,6,1 P7 K1	0.99737	18.1 hr	0.99715	48.5 hr		
4,6,1 P7 K3	0.99743	42.9 hr	0.99721	124.9 hr		
4,6,1 P7 K5	*	*	0.99722	251.9 hr		

9 Conclusions

The above work shows how modern coding structures (Fortran 90+) were introduced into the DIF3D code to enable improved calculation capabilities. The previous code relied upon hard coded data statements containing the space-angle matrices which were selected based upon the computational limitations in the mid 90s. The new version employs generation routines which allow upwards of 99th order spatial and 99th order angular trial functions to be used. Though these are rather impractical today, they will likely become more realistic with time as higher order approximations can be necessary during the validation process. This work eliminated the poor performance of calculations involving anisotropic scattering which was an artifact of the implementation choice presented by Fortran 77. While the new routine makes a substantial dent in the computational effort required by most routine users of VARIANT (P₅ flux with P₃ scattering), we suspect there is still more room for performance improvements with the ongoing research into using Krylov solvers in a parallel computing environment with the VARIANT methodology. While this is likely not going to be implemented in DIF3D it shows how valuable the VARIANT tool is and could be if the remaining performance issues that limit its use are removed.

The new routines added to DIF3D have been thoroughly tested and validated for Cartesian and hexagonal geometries in both two- and three-dimensions. As discussed, the memory usage of VARIANT was substantially reduced without a significant impact upon the computational effort by employing non-zero storage based algorithms. We presented eigenvalue and computational performance results for higher order angular and spatial approximations which proved the vast improvement in performance that was gained with the recent changes. We also note that the new modifications to the anisotropic scattering treatment allowed problems to be solved which were previously not solvable with VARIANT and difficult to solve with any other existing transport solvers. In all cases the solutions obtained using the new version exactly matched those of the existing production version although it is important to note that the production version (DIF3D-9.0) had to be corrected to guarantee the agreement.

In addition to the improvements, there were considerable errors in the implementation for hexagonal 60 and 120 periodic geometry models that were corrected. Those changes unfortunately have temporarily eliminated the ability to run 30 degree symmetry models, but this should be minor for most problems of interest. In an attempt to improve the performance, the coarse mesh rebalance acceleration and fission source extrapolation were removed due to their inconsistent performance and periodic instabilities. They were temporarily replaced with Tchebychev acceleration which is not as effective, but is more reliable. While the CMR acceleration was clearly the primary acceleration technique, it was not uncommon for users to disable it which might have been a result of the errors found in the hexagonal 60 and 120 periodic boundary conditions. Errors were also identified in the ω acceleration and SPN anisotropic scattering both of which were completely fixed. To compensate for any potential convergence troubles, an algorithm to increase the inner iterations when convergence is stalling was added to the code.

In conclusion, the new version of VARIANT incorporated in DIF3D-11 allows users to fully validate their calculations by removing all concerns of space-angle refinement and resolves numerous bugs. Eighth order flux, eighth order source, and cubic order Lagrange multiplier

spatial approximations were demonstrated. Angular approximations ranging from P_1 through P_{11} and SP_1 through SP_9 were also demonstrated. While these new capabilities are well beyond the original design concept of VARIANT, they have proven to be a valuable addition to the existing suite of reactor analysis tools. Finally, the new improvement to VARIANT facilitates the existing users of DIF3D to study ever more difficult problems and is a testament to the importance of the VARIANT tool for existing research being carried out at Argonne National Laboratory.

9.1 Acknowledgements

The authors of this report would like to acknowledge the contributions made by Giuseppe Palmiotti, Elmer E. Lewis, Won Sik Yang, Keith Derstine, Amr Mohammed, Barbara Chandler, Temitope Taiwo, Cristian Rabiti, and Cyrus Adams for providing frequent consultation, guidance, and discussion during the implementation of this work.

This work is supported by the U.S. Department of Energy, office of Nuclear Engineering, under contract # DE-AC02-06CH11357.

10 References

1. G. Palmiotti, E.E. Lewis, and C.B. Carrico, "VARIANT: VARIational Anisotropic Nodal Transport for Multidimensional Cartesian and Hexagonal Geometry Calculations," Argonne National Laboratory ANL-95/40 (October 1995).
2. E. E. Lewis, C. B. Carrico, G. Palmiotti, "Variational Nodal Formulation for the Spherical Harmonics Equations," *Nucl. Sci. Eng.*, **122**, 194 (1996).
3. C. B. Carrico, E. E. Lewis and G. Palmiotti, "Three-dimensional Variational Nodal Transport Methods for Cartesian, Triangular and Hexagonal Criticality Calculations," *Nucl. Sci. Eng.*, **111**, p 168-179 (1992).
4. C. B. Carrico, E. E. Lewis, and G. Palmiotti, "A Reduced Angular Trial Function Set for the Variational Nodal Method," *Trans. Am. Nucl. Soc.*, **65**, 200 (1992).
5. W. S. Yang, G. Palmiotti, and E. E. Lewis "Numerical Optimization of Computing Algorithms of the Variational Nodal Method Based on Transformations of Variables," *Nucl. Sci. and Eng.*, **139**, 174-185 (2001).
6. M. A. Smith, N. Tsoulfanidis, E. E. Lewis, G. Palmiotti and T. A. Taiwo, "Higher Order Angular Capabilities of the VARIANT Code," *Trans. Am. Nucl. Soc.*, **86**, 321-322 (2002).
7. M. A. Smith, "Implementation of the Finite Element Approximation in a Nodal Three-Dimensional Transport Program to allow for Explicit Node Geometries," PhD dissertation, University of Missouri-Rolla (2001).
8. M. A. Smith, G. Palmiotti, E. E. Lewis, and N. Tsoulfanidis, "An Integral Form of the Variational Nodal Method," *Nucl. Sci. Eng.* **146**, 141 (2004).
9. M. A. Smith, G. Palmiotti, E.E. Lewis, "First-Order Integral Spherical Harmonics Formulation Compatible with the VARIANT Code," *Proc. PHYSOR 2006- Advances in Nuclear Analysis and Simulation*, Vancouver, BC, Canada, Sept 10-14 (2006).
10. E. E. Lewis, M. A. Smith, and G. Palmiotti, "The Variational Nodal Method: Some History and Recent Activity," *International Topical Meeting on Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Applications*, Sept. 12-15, Avignon, France, 2005.
11. Hui Zhang and E. E. Lewis, "Generalization of the Variational Nodal Method to Spherical Harmonics Approximations in R-Z Geometry," *Nucl. Sci. Eng.* **152**, 1-8 (2006).
12. C. Rabiti, Argonne National Laboratory, Personal Communication (Jan. 2006).
13. G. Ya. Rumyantsev, "Boundary Conditions in the Spherical Harmonic Method," *J. Nucl. Energy*, **16**, 111 (1962).
14. W. S. Yang, M. A. Smith, G. Palmiotti, and E. E. Lewis, "Interface Conditions for Spherical Harmonics Methods," *Nucl. Sci. Eng.* **150**, 259, 2005.
15. E.E. Lewis and W.F. Miller Jr., *Computational Methods of Neutron Transport*, John Wiley & Sons, New York (1984).
16. William H. Press, Brian P. Flannery, Saul A. Teukolsky, William T. Vetterling, *Numerical Recipes in Pascal*, Cambridge University Press, New York (1994).
17. K. D. Lathrop, B. G. Carlson, "Discrete Ordinates Angular Quadrature of the Neutron Transport Equation," Los Alamos Scientific Laboratory LA-3186 (1965).
18. C.H. Adams, et al. "The Utility Subroutine Package Used by Applied Physics Division Export Codes," ANL-83-3. Argonne National Laboratory (1983).

19. D. Kaushik, M. Smith, A. Wollaber, B. Smith, A. Siegel, and W. S. Yang, "Enabling High Fidelity Neutron Transport Simulations on Petascale Architectures," Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, vol. 67, Portland, Oregon, ACM (2009).
20. T. Takeda and H. Ikeda, "3-D Neutron Transport Benchmarks," NEACRP-1-300 OECD/NEA, Organization of Economic Cooperation and Development/Nuclear Energy Agency (March 1991).
21. Giuseppe Palmiotti, Argonne National Laboratory, Personal Communication (January 2006).
22. Taek K. Kim, Argonne National Laboratory, Personal Communication (January 2007).
23. Won Sik Yang, Argonne National Laboratory, Personal Communication (August 2009).
24. Won Sik Yang, Argonne National Laboratory, Personal Communication (May 2007).
25. M. A. Smith, C. Adams, W. S. Yang, and E. E. Lewis, "VARI3D & PERSENT: Perturbation and Sensitivity Analysis," ANL/NE-13/8 (2013).

11 Appendix A. Modifications to DIF3D Input Descriptions

```

C-----
CR PARAMETERS FOR VARIATIONAL NODAL OPTION (TYPE 12) -
C -
CL FORMAT-----(I2,4X,11I6) -
C -
CD # COLUMNS CONTENTS...IMPLICATIONS, IF ANY -
CD = =====
CD 1 1-2 12 -
CD -
CD 2 7-12 NODAL SPATIAL APPROXIMATION. -
CD ENTER ONE OF THE FOLLOWING -
CD LMN = CLASSICAL VARIANT -
CD LLMMXN = NEW VARIANT INPUT STYLE -
CD -
CD L IS THE ORDER OF THE POLYNOMIAL APPROXIMATION OF THE -
CD SOURCE WITHIN THE NODE. -
CD M IS THE ORDER OF THE POLYNOMIAL APPROXIMATION OF THE -
CD FLUXES WITHIN THE NODE. -
CD N IS THE ORDER OF THE POLYNOMIAL APPROXIMATION OF THE -
CD LEAKAGES ON THE SURFACES OF THE NODES. -
CD X IS THE ORDER REDUCTION TO BE APPLIED IN THE Z -
CD DIRECTION. FOR A FULLY MTH ORDER APPROXIMATION -
CD USE 0. -
CD -
CD HEXAGONAL AND CARTESIAN GEOMETRY: -
CD L = 1... LINEAR SOURCE APPROXIMATION. -
CD L = 2... QUADRATIC SOURCE APPROXIMATION. -
CD L > 2... LTH ORDER SOURCE APPROXIMATION. -
CD M = 1... LINEAR FLUX APPROXIMATION. -
CD M = 2... QUADRATIC FLUX APPROXIMATION. -
CD M > 2... MTH ORDER FLUX APPROXIMATION. -
CD N = 0... FLAT LEAKAGE APPROXIMATION. -
CD N = 1... LINEAR LEAKAGE APPROXIMATION (DEFAULT). -
CD N > 1... NTH ORDER LEAKAGE APPROXIMATION. -
CD X >=0... ORDER REDUCTION IN THE Z DIRECTION -
CD -
CD LEADING ZEROS ARE IRRELEVANT. -
CD DEFAULT VALUES ARE L=N+1 M=4 N=1, LLMMXN= 20401 -
CD -
CD FOR CLASSICAL VARIANT STYLE 3-D INPUT: -
CD X = 1 WHEN M = 5 AND X = 2 WHEN M > 5 -
CD FOR HEXAGONAL GEOMETRY IT IS SUGGESTED THAT M >= 5 -
CD -
CD 3 13-18 NODAL ANGULAR APPROXIMATION. -
CD ENTER ONE OF THE FOLLOWING THREE OPTIONS -
CD MN = CLASSICAL VARIANT INPUT STYLE -
CD MMNN = UPDATED VARIANT INPUT STYLE -
CD HMMNN = NEW VARIANT INPUT STYLE -
CD -
CD H IS THE PN ANGULAR TRIAL FUNCTION SET TO USE (1 OR 3) -
CD M IS THE ORDER OF THE PN EXPANSION OF THE FLUX. -
CD N IS THE ORDER OF THE PN EXPANSION OF THE LEAKAGE. -
CD -
CD HEXAGONAL AND CARTESIAN GEOMETRIES: -
CD H = 1... STANDARD FUNCTIONS, CIRCA 1992 (DEFAULT). -
CD H = 3... YANG FUNCTIONS, CIRCA 2002. -
CD M = 1... P1 FLUX EXPANSION. -
CD M = 3... P3 FLUX EXPANSION (DEFAULT). -
CD M > 3... PM FLUX EXPANSION, M MUST BE ODD -
CD N = 1... P1 LEAKAGE EXPANSION. -
CD N = 3... P3 LEAKAGE EXPANSION (DEFAULT). -
CD N > 3... PN LEAKAGE EXPANSION, N MUST BE ODD -
CD -
CD M MUST EQUAL N, REDUCED ORDER 31 WAS OBSOLETE. -
CD DEFAULT VALUE IS 10303. DIFFUSION IS 11=0101=10101 -
CD -
CD EXAMPLES: 77=0707=10707 ; 30909 ; -9999 ; 11111 -
CD -
CD IF MN IS NEGATIVE, SIMPLIFIED SPHERICAL HARMONICS(SPN) -
CD ARE USED (EXAMPLE -33=-0303=-10303). -
CD YANG FUNCTIONS DO NOT APPLY TO SPN -
CD -
    
```

Figure 11.1. Updated A.DIF3D File Description

In DIF3D 10.0 and 11.0, the VARIANT input description was modified to facilitate higher order space-angle expansions. There are also now two sets of angular basis functions for the surfaces. The first is the default angular trial function set used in VARIANT since 1992: $H=1$. The second were derived by Yang earlier in Section 3: $H=3$ (30505 for the example in Table A.1). Table 11.1 summarizes the changes made to the angular approximation on the A.DIF3D input.

Table 11.1. Changes to the DIF3D Input Description on Card 12.

Input Field	$\pm MN$	$\pm HMMNN$ or $\pm MMNN$ or $\pm MN$
\pm	$P_N (+)$ or $SP_N (-)$	$P_N (+)$ or $SP_N (-)$
H	—	Angular Trial Function Set
M	Order of the P_N Flux Expansion	Order of the P_N Flux Expansion
N	Order of the P_N Leakage Expansion	Order of the P_N Leakage Expansion
P_5 example	55	10505 or 0505 or 55
SP_5 example	-55	-10505 or -0505 or -55

A similar change in the input description had to be made for the spatial approximation, which is described in Table 11.2. As can be seen in Table 11.2, there is a new variable, X, included in the input which is not consistent with that of the preexisting input. The reason for the inconsistency is that the production version of VARIANT only allowed a maximum 4th order flux and source approximation in the z direction while the x-y flux approximation was extended to 6th order. To keep this flexibility, the new variable was added into the DIF3D input definition, although its usage is not required for the new input style.

Table 11.2. Changes to the DIF3D BCD Input Description on Card 12.

Input Field	LMN	LLMMXN or LMN
L	Order of the Source	Order of the Source
M	Order of the Flux	Order of the Flux
N	Order of the Leakage	Order of the Leakage
X	--	Order Reduction in the Z Direction
Examples	462	040622 or 462
	442	040402 or 442

To facilitate the needs of PERSENT, a new fixed source file was added to DIF3D the details of which are included in Figure 11.2. The NHFLUX and NAFLUX file had to be updated similarly and users will find that the VARSRC file below has a nearly identical setup to the NHFLUX file which was done for convenience internal to PERSENT.

Figure 11.2. Description of New VARSRC Fixed Source File Added to DIF3D-11.0

```

C*****
C
C          Created 03/05/12
C
C          VARSRC FOR DIF3D-VARIANT
C          FIXED SOURCE FILE FORMAT FOR USE WITH VARIANT
C          CONTAINS REGULAR NODAL FLUX-MOMENTS
C          CALCULATIONS IN DIF3D 11.0 AND LATER
C
C          CARD 3D FORMAT SAME AS NHFLUX_VARIANT CARDS 1D,3D
C          WITH IWNHFL=1 AND NSURF=1
C*****

C-----
CS      FILE STRUCTURE
CS
CS      RECORD TYPE          RECORD    PRESENT IF
CS      =====
CS      FILE IDENTIFICATION          ALWAYS
CS      SPECIFICATIONS              1D    ALWAYS
CS
CS      ***** (REPEAT FOR ALL GROUPS)
CS      *      FLUX MOMENTS          3D    ALWAYS
CS      *****
C-----

C-----
C      FILE IDENTIFICATION
C
C      HNAME, (HUSE(I), I=1,2), IVERS
C
C      1+3*MULT=NUMBER OF WORDS
C
C      HNAME          HOLLERITH FILE NAME - NHFLUX - (A6)
C      HUSE(I)        HOLLERITH USER IDENTIFICATION (A6)
C      IVERS          FILE VERSION NUMBER
C      MULT           DOUBLE PRECISION PARAMETER
C                   1- A6 WORD IS SINGLE WORD
C                   2- A6 WORD IS DOUBLE PRECISION WORD
C-----

C-----
C      SPECIFICATIONS      (1D RECORD)
C
C      NDIM,NGROUP,NINTI,NINTJ,NINTK,ITER,EFFK,POWER,NSURF,
C      NMOM,NINTXY,NPCXY,NSCOEF,ITRORD,IAPRX,ILEAK,IAPRXZ,ILEAKZ,
C      IORDER,NPCBDY,NPCSYM,NPCSEC,IWNHFL,NMOMS (IDUM,I=1,6)
C
C      30 =NUMBER OF WORDS
C
C      NDIM          (not used)
C      NGROUP        NUMBER OF ENERGY GROUPS
C      NINTI          (not used)
C      NINTJ          (not used)
C      NINTK          NUMBER OF PLANES
C      ITER          (not used)
C      EFFK          (not used)
C      POWER         (not used)
C      NSURF         NUMBER OF XY-PLANE SURFACES PER NODE
C                   Must = 1 to read with NHFLUX reader
C
C      NMOM          NUMBER OF EVEN-PARITY FLUX MOMENTS IN VARIANT
C      NINTXY        NUMBER OF MESH CELLS (NODES) ON XY-PLANE
C      NPCXY         (not used)
C      NSCOEF        (not used)
C      ITRORD        (not used)
C      IAPRX         (not used)
C      ILEAK         (not used)
C      IAPRXZ        (not used)
C      ILEAKZ        (not used)
C      IORDER        (not used)
C      NPCBDY        (not used)
C      NPCSYM        (not used)
C      NPCSEC        (not used)
C      IWNHFL        NHFLUX CONTENT
C                   Must=1, SIGNIFIES ONLY FLUXES ARE PRESENT
C      NMOMS        NUMBER OF ODD-PARITY FLUX MOMENTS IN VARIANT

```

```
C   IDUM                (not used)                -  
C                                                           -  
C-----  
C-----  
C           REGULAR FLUX MOMENTS   (3D RECORD)      -  
C                                                           -  
C   ((FLUX(I,J),I=1,NMOM+NMOMS),J=1,NINTXY) --SEE STRUCTURE BELOW--- -  
C                                                           -  
C   (NMOM+NMOMS)*NINTXY*MULT = NUMBER OF WORDS      -  
C                                                           -  
C   DO K=1,NINTK                                     -  
C     READ(N)   *LIST AS ABOVE*                     -  
C   END DO                                           -  
C                                                           -  
C   FLUX(I,J) ARE THE REGULAR FLUX MOMENTS BY NODE FOR THE PRESENT -  
C     ENERGY GROUP                                  -  
C                                                           -  
C-----
```




Nuclear Engineering Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439

www.anl.gov



Argonne National Laboratory is a U.S. Department of Energy
laboratory managed by UChicago Argonne, LLC