

# Arbitrarily High Order Transport Method of the Characteristic Type for Tetrahedral Grids

Y. Y. Azmy  
Oak Ridge National Laboratory  
P.O. Box 2008, MS-6363  
Oak Ridge, Tennessee 37831-6363  
[azmyyy@ornl.gov](mailto:azmyyy@ornl.gov)

D. A. Barnett  
Lignum Computing, Inc.  
2240 Ennis Rd.  
Pattersonville, NY 12137  
[allen@lignumcomputing.com](mailto:allen@lignumcomputing.com)

Prepared for:

2001 ANS International Meeting on  
Mathematical Methods for Nuclear Applications  
September 9-13, 2001  
Salt Lake City, Utah  
USA

The submitted manuscript has been authored by a contractor of the U.S. Government under contract No. DE-AC05-00OR22725. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.



# ARBITRARILY HIGH ORDER TRANSPORT METHOD OF THE CHARACTERISTIC TYPE FOR TETRAHEDRAL GRIDS

**Y. Y. Azmy**

Oak Ridge National Laboratory  
P. O. Box 2008, MS 6363  
Oak Ridge, Tennessee 37831-6363  
azmyyy@ornl.gov

**D. A. Barnett**

Lignum Computing, Inc.  
2240 Ennis Rd.  
Pattersonville, NY 12137  
allen@lignumcomputing.com

**Keywords:** Discrete-ordinates, Arbitrarily High Order Transport, Characteristic Method, Tetrahedral cells.

## ABSTRACT

A formalism is derived for the Arbitrarily High Order Transport (AHOT) method of the Characteristic type (AHOT-C) in three-dimensional geometry for unstructured grids (UG). The resulting equations are implemented in a computer code, AHOT-C-UG, in the C language. The transport solution on the unstructured grid is stored as two inter-linked lists of cell and face flux moments. This arrangement allows the transport sweep to select the order of evaluation dynamically so that the typical recursive ordering of the discrete ordinate's mesh sweep is maintained without the need to store a precomputed order for each ordinate. The dynamic cell sweep order thus reduces the memory demand without excessively increasing execution time. Comparison of AHOT-C-UG's solutions to fine mesh TORT solutions illustrate high accuracy of the new method. In particular, large half a million cell numerical tests illustrate a convergence rate for the error as  $O(h)$ , where  $h$  is a measure of the longest edge in the tetrahedral grid. Execution time on a 700 MHz Intel Pentium III running Linux 2.4.0 is less than 0.2 ms per cell-angle sweep operation. Also the total memory requirement is of the order of 240 bytes per tetrahedral cell, where 64-bit arithmetic precision is employed throughout.

## 1 INTRODUCTION

Discrete ordinates methods have been developed primarily for one-, two-, and three-dimensional orthogonal coordinate systems. This requires approximating the geometry of the physical system which seldom coincides with an orthogonal coordinate system in practical applications. Furthermore, in curvilinear coordinates an additional approximation is introduced to deal with the so-called *redistribution* term, which is not completely understood, and whose effect on the solution accuracy is not easily accounted for. These two shortcomings of discrete ordinates codes are greatly reduced by employing a three-dimensional irregular, or unstructured, grid as has been accomplished in recent years by a few research groups. This permits piecewise plane representation of curved interfaces, rather than staircasing, resulting in better geometric approximation, elimination of the redistribution term, and potentially a reduction in the number of computational cells. Nevertheless, several difficulties have to be overcome to reach this objective. First, the spatial discretization procedure over cells whose boundaries do not coincide with constant surfaces in an orthogonal coordinate system must be developed. While this has been done in the framework of finite element methods in other fields, it is still rather new in the neutron transport field. Based on existing experience it is clear that the Characteristic method approach is most likely to succeed as it does not require taking transverse moments of the transport operator, like Nodal methods, which is not easy to generalize to unstructured grids. Second, it is highly desirable to develop higher order spatial approximation methods. Arbitrarily High Order Transport Characteristic (AHOT-C) methods have been developed for two-dimensional Cartesian grids (Azmy, 1992). Here we extend this formalism to unstructured, three dimensional grids. Earlier studies have demonstrated the superiority of high order methods in terms of providing higher pointwise accuracy on a given mesh, ensuring solution positivity in highly absorbing materials, and improving computational efficiency, both in terms of execution time and memory requirement.

The so-derived single cell AHOT-C equations must then be implemented in a mesh sweep algorithm to solve multi-cell problems (Suslov, 1997). In an early attempt to accomplish this goal the sweep algorithm started by determining the order in which the tetrahedral cells are to be swept in a given angular direction so that all incoming face angular flux moments would be available from sweeping previous cells. It then processed each tetrahedral cell in this order using the single cell routines feeding the outgoing face angular flux from one cell to incoming faces of adjacent tetrahedrons. The conclusion of that research project was that AHOT-C is feasible and valuable in cases where accuracy of the geometric description of the problem configuration is important. However, the increase in CPU time and memory requirement with increasing problem size and spatial approximation order was too large, casting serious doubts on the practical utility of this method. In this

work we describe a new algorithm for sweeping 3-Dimensional Unstructured Grids as implemented in the AHOT-C-UG (AHOT-C on Unstructured Grids) code that is particularly focused on high computational efficiency, both in terms of execution time and memory requirement. We illustrate these features by performing a few numerical experiments, and comparing the solutions to those obtained by TORT (Rhoades and Simpson, 1997) thus establishing the accuracy and efficiency of the new method and algorithm.

The remainder of this paper is organized as follows. In Sec. 2 we develop the previously unpublished AHOT-C formalism for a single, generic tetrahedral cell. The dynamic-ordering sweep algorithm used to extend these equations to multi-cell unstructured grids is described in Sec. 3. A variety of numerical tests and comparison of the new method's results to traditional methods, i.e. TORT, are presented in Sec. 4. Our conclusions are summarized in Sec. 5.

## 2 THE AHOT-C FORMALISM IN TETRAHEDRAL GEOMETRY

Consider the arbitrary tetrahedron with vertices located at  $v^\ell$ ,  $\ell = 0, \dots, 3$ , where  $v^\ell \equiv \{x_j^\ell, j = 1, 2, 3\}$  with respect to the global coordinate system  $x_1, x_2, x_3$  centered at  $O$ . The four faces of the tetrahedral cell are  $P^k$ ,  $k = 0, \dots, 3$ , whose plane equations,  $\alpha_j^k x_j = \beta^k$ , are determined from the vertices  $v^k$ ; note using the summation convention. The unit vector along the direction of motion of a neutron is  $\hat{\Omega} \equiv \{\mu_j \hat{x}_j, j = 1, 2, 3\}$  where  $\hat{x}_j$  is the unit vector along the  $x_j$ -axis, and  $\mu_j$  is the direction cosine of  $\hat{\Omega}$  with respect to the  $x_j$ -axis. The discrete ordinates approximation of the neutron transport equation over the tetrahedral cell is given by,

$$\hat{\Omega} \cdot \nabla \psi + \sigma_T \psi = \sigma_S \phi + S, \quad (1)$$

where we have employed standard notation. To develop an AHOT for Eq. (1) we utilize the set of polynomial functions,

$$\lambda_{\vec{i}} \equiv \prod_{j=1}^3 x_j^{i_j}, \quad 0 \leq i_j \leq \Lambda, \quad \vec{i} \equiv \{i_j, j = 1, 2, 3\}, \quad (2)$$

for the order  $\Lambda$  method.

The conservation of neutrons to all computed orders over the tetrahedron is enforced by multiplying Eq. (1) by  $\lambda_{\vec{i}}$  then integrating over the cell's volume. This results in the relation,

$$\sum_{k=0}^3 (\hat{\Omega} \cdot \hat{\alpha}^k) \psi_{\vec{i}}^k - \sum_{j=1}^3 i_j \mu_j \bar{\psi}_{\vec{i}-\vec{j}} + \sigma_T \bar{\psi}_{\vec{i}} = \sigma_S \bar{\phi}_{\vec{i}} + \bar{S}_{\vec{i}}, \quad (3)$$

where we have defined the face-moments of the angular flux by,

$$\psi_{\vec{i}}^k \equiv \int_{P^k} \lambda_{\vec{i}} \psi dP^k, \quad (4)$$

and the cell-moments of the angular flux, scalar flux, and fixed source by,

$$\bar{q}_{\vec{i}} \equiv \int_V \lambda_{\vec{i}} q dV, \quad q \equiv \psi, \phi, \text{ and } S, \quad (5)$$

respectively, and  $V$  is the cell volume. The scalar flux moments are related to the angular flux moments by the standard discrete ordinates relationship,

$$\bar{\phi}_{\vec{i}} = \sum_{\text{discrete ordinates}} w \bar{\psi}_{\vec{i}}. \quad (6)$$

Clearly if  $\hat{\Omega}$  lies within the plane of face  $k$  the first term in Eq. (3) vanishes for this  $k$ ; also if  $i_j = 0$  for any  $1 \leq j \leq 3$ , the contribution to the summation in the second term of Eq. (3) from this  $j$  vanishes.

Typically in the case of an orthogonal grid superimposed on an orthogonal coordinate system the face moment of the angular flux, Eq. (4), comprises a double integral over two dimensions while the expansion function in the third dimension evaluates to a constant on the given face. In tetrahedral geometry this is generally not possible. Instead one has to replace one of the expansion functions, say  $x_3^{i_3}$ , by its value on face  $k$  in Eq. (4) to obtain,

$$\psi_{\vec{i}}^k = \sum_{m_1=0}^{i_3} \sum_{m_2=0}^{i_3-m_1} \gamma_{i_3, m_1, m_2}^k \psi_{(i_1+m_1)\hat{i}_1 + (i_2+m_2)\hat{i}_2 + 0\hat{i}_3}^k, \quad \alpha_3^k \neq 0, \quad (7)$$

where the coefficients are given by,

$$\gamma_{i_3, m_1, m_2}^k \equiv \frac{i_3! (\beta^k)^{i_3-m_1-m_2} (-\alpha_1^k)^{m_1} (-\alpha_2^k)^{m_2}}{m_1! m_2! (i_3 - m_1 - m_2)! (\alpha_3^k)^{m_2}}, \quad \alpha_3^k \neq 0. \quad (8)$$

Equation (7) essentially replaces 3-variable moments over a face of a tetrahedron by a linear combination of 2-variable moments over the same face, which we denote  $\psi_{\vec{i}}^k$ , where  $\vec{i}$  is a two dimensional vector. This greatly simplifies the derivation of the relationship between the angular flux moments over the incoming and outgoing faces using the characteristic method. It must be emphasized that the choice to eliminate the  $x_3$ -moments in Eqs. (7) is arbitrary; in general Eq. (7) is reformulated by eliminating the  $x_j$ -moment for any  $j$  satisfying the condition  $\alpha_j^k \neq 0$ . It has also been observed that it is necessary to compute  $\psi_{\vec{i}}^k$  to order  $2\Lambda$  in order for the conversion from two-variable moments to three-variable moments to retain complete fidelity of the three-dimensional solution.

The formulas derived so far and in the remainder of this report are relative to a local coordinate system for each tetrahedral cell, e.g. its center of mass. The cell-moments defined by Eq. (5) are not used outside of the cell in which they are defined. In contrast the face-moments, Eq. (4), depend on the local origin implying they are, in general, not continuous across an interface. Sweeping the mesh along a given discrete ordinate, however, requires establishing a relationship between the flux moments in cells 1 and 2 whose origins are related by,

$$\vec{O}_2 = \vec{O}_1 + \sum_{j=1}^3 \zeta_j \hat{x}_j. \quad (9)$$

Introducing Eq. (9) into Eq. (4) yields,

$$[\psi_i^k]_{O_2} = \sum_{m_1=0}^{i_1} \sum_{m_2=0}^{i_2} \Gamma_{m_1, m_2} [\psi_{\vec{m}}^k]_{O_1}. \quad (10)$$

where the coefficients are defined by,

$$\Gamma_{m_1, m_2} \equiv \frac{i_1! i_2! (-\zeta_1)^{(i_1-m_1)} (-\zeta_2)^{(i_2-m_2)}}{m_1! m_2! (i_1 - m_1)! (i_2 - m_2)!}. \quad (11)$$

Equation (10) relates the face-moments of the angular flux incoming to the tetrahedron centered at  $O_2$  to the face-moments of the angular flux exiting the tetrahedron centered at  $O_1$ .

The balance equation, Eq. (3), relates face- and cell-moments over the entire area and volume, respectively, of the angular flux and fixed neutron source. In contrast the characteristic relations, derived shortly, are expressed in terms of local expansion coefficients within tetrahedral cells and on their faces because in general they do not cover these in their entirety in a continuous manner. Traditionally spatial moments are preferable to compute because they relate to *total* quantities; for example  $\bar{\phi}_{\vec{0}}$  is easily related to the total reaction rate over a cell. Hence we derive expressions for the local expansion coefficients in terms of the corresponding moments. Suppose  $\sum_{\vec{i}=0}^{\bar{\Lambda}} \Psi_{\vec{i}}^k \lambda_{\vec{i}}^k$  is the local expansion of the angular flux on face  $k$ . The local face-moment is given by,

$$\psi_{\vec{m}}^k = \sum_{\vec{i}=0}^{\bar{\Lambda}} Y_{\vec{i}+\vec{m}}^k \Psi_{\vec{i}}^k, \quad (12a)$$

$$Y_{\vec{i}+\vec{m}}^k \equiv \int_{P^k} dP \lambda_{\vec{i}}^k \lambda_{\vec{m}}^k. \quad (12b)$$

Had the basis functions been selected to be mutually orthogonal, as indeed is the case in Cartesian geometry, the coefficients  $Y_{\vec{n}}^k$  would have comprised a diagonal matrix.

This is not possible here though due to the complexity of constructing an orthogonal basis for a general tetrahedron. Hence we write  $Y_{\bar{n}}^k$  as a linear combination of three subintegrals from the  $x_2$ -axis, for example, to the three edges defining face  $k$  on the  $x_1, x_2$  plane,

$$Y_{\bar{n}}^k = y_{\bar{n}}^{k,(v_0,v_1)} + y_{\bar{n}}^{k,(v_1,v_2)} + y_{\bar{n}}^{k,(v_2,v_0)}, \quad (13)$$

where, for example, for the edge connecting vertex  $v_0$  to vertex  $v_1$ ,

$$y_{\bar{n}}^{k,(v_0,v_1)} = \frac{1}{\alpha_3^k} \sum_{n=0}^{n_1+1} \frac{n_1 (\delta_{v_0,v_1})^n}{n! (n_1 + 1 - n)! (n_2 + 1 + n)} \\ \times (x_1^0 - \delta_{v_0,v_1} x_2^0)^{(n_1+1-n)} [(x_2^1)^{n_2+n+1} - (x_2^0)^{n_2+n+1}], \quad (14)$$

and the gradient of this edge is,

$$\delta_{v_0,v_1} \equiv \frac{x_1^1 - x_1^0}{x_2^1 - x_2^0}. \quad (15)$$

It is worth noting that the next to last term in Eq. (14) is invariant under switching the vertices order, while the last term switches sign. It follows that

$$y_{\bar{n}}^{k,(v_0,v_1)} = -y_{\bar{n}}^{k,(v_1,v_0)}. \quad (16)$$

The proper combination of the  $y_{\bar{n}}^k$  components in Eq. (13) must be performed in a cyclic fashion, the sense of which only determines the sign of  $Y_{\bar{n}}^k$ . This cyclic sense is arbitrary; thus we select it to produce a positive area of face  $k$ , i.e.  $Y_0^k > 0$ .

Analogously we derive the relationship of the cell-moments to the local expansion,

$$q_{\bar{m}} = \sum_{\bar{i}=0}^{\Lambda} Z_{\bar{i}+\bar{m}} Q_{\bar{i}}, \quad q \equiv \psi, \phi, S, \quad (17)$$

where the coefficients  $Z_{\bar{i}}$  are written as the linear combination,

$$Z_{\bar{n}} = \sum_{k=0}^3 \left[ z_{\bar{n}}^{k,(v_0,v_1)} + z_{\bar{n}}^{k,(v_1,v_2)} + z_{\bar{n}}^{k,(v_2,v_0)} \right], \quad (18)$$

and, for example,

$$z_{\bar{n}}^{k,(v_0,v_1)} = \frac{1}{(\alpha_3^k)^{n_3+1}} \sum_{n=0}^{n_3+1} \sum_{n'=0}^{n_3+1-n} \sum_{n''=0}^{n_3+1+n} \\ \times \left[ \frac{n_3! (n_1 + n)! (-\alpha_1^k)^n (-\alpha_2^k)^{n'} (\beta^k)^{n_3+1-n-n'} (\delta_{v_0,v_1})^{n''}}{n! n'! n''! (n_3 + 1 - n - n')! (n_1 + 1 + n - n'')! (n_2 + 1 + n' + n'')} \right] \\ \times (x_1^0 - \delta_{v_0,v_1} x_2^0)^{n_1+1+n-n''} [(x_2^1)^{n_2+n'+n''+1} - (x_2^0)^{n_2+n'+n''+1}]. \quad (19)$$

Here also,

$$z_{\bar{n}}^{k,(v_0,v_1)} = -z_{\bar{n}}^{k,(v_1,v_0)}. \quad (20)$$

The proper order of the vertices in Eq. (18) is obtained by selecting a cyclic order on one face,  $k$ , and computing  $z_{\bar{n}}^{k,(v_0,v_1)}, \dots$ . Then the cyclic sense on each edge of face  $k$  is reversed and  $z_{\bar{n}}^{k',(v_1,v_0)}, \dots$ , is computed for  $k' \neq k$ . The resulting twelve values are summed as in Eq. (18) to obtain  $Z_{\bar{n}}$ , then all its components' signs are adjusted so that  $Z_{\bar{0}} > 0$ .

To close the AHOT-C's set of discrete-variable equations additional relations among the face-moments of the angular flux are needed. These are obtained by integrating Eq. (1) along its characteristic curves, the straight lines defined by,

$$t - t^{+k} = \frac{x_j - x_j^{+k}}{\mu_j}, \quad j = 1, 2, 3, \quad (21)$$

where  $t$  is the characteristic variable,  $t^{+k}$ , and  $x_j^{+k}$  are the characteristic, and independent variables, respectively, evaluated on the incoming face  $k$ . [Analogous expressions can be written relative to the outgoing face  $k$  by replacing  $+k$  with  $-k$  in Eq. (21)]. This results in,

$$\psi(t) = \psi(t^{+k}) e^{-\sigma_T(t-t^{+k})} + \int_{t^{+k}}^t e^{-\sigma_T(t-t')} s(t') dt', \quad (22)$$

where we have denoted the present iterate of the scattering source by  $s(t') \equiv \sigma_s \phi(t') + S(t')$ , and  $t$  is an arbitrary point in the tetrahedron. Evaluating Eq. (22) on the outgoing face then taking the  $\bar{i}$  moment over this face yields,

$$\psi_{\bar{i}}^{-k} = \chi_{\bar{i}} + \xi_{\bar{i}}, \quad (23)$$

where  $\chi_{\bar{i}}$ , and  $\xi_{\bar{i}}$  are the incoming face, and volumetric source contributions, respectively.

Clearly since the angular flux spatial moments on the incoming face(s) are not necessarily the same, the first term on the RHS of Eq. (23) must be evaluated once for each **characteristic tetrahedron** (CT). The CT components of the original tetrahedron are defined for a given discrete ordinate  $\hat{\Omega}$  by the subtetrahedra which possess one edge parallel to  $\hat{\Omega}$  so that only one face in a CT can be incoming. For a CT the incoming face flux contribution to the outgoing face flux moments is expressed in the form,

$$\chi_{\bar{i}} = \left( \frac{\hat{\alpha}^{+k} \cdot \hat{\Omega}}{\hat{\alpha}^{-k} \cdot \hat{\Omega}} \right) e^{B_{\bar{3}}^{+k}} \sum_{m_{31}=0}^{2\Lambda} \sum_{m_{32}=0}^{2\Lambda} f_{m_{31},m_{32}} \Psi_{m_{31},m_{32}}^{+k}. \quad (24)$$

If more than one CT make up the present tetrahedron the contribution of each is computed via Eq. (24) then summed. In Eq. (24) the coefficients  $f_{m_{31}, m_{32}}$  are written as linear combinations analogous to Eq. (13),

$$\begin{aligned}
f_{m_{31}, m_{32}} &\equiv \sum_{(v_0, v_1)} \sum_{m_{11}=0}^{i_1} \sum_{m_{12}=0}^{i_1 - m_{11}} \sum_{m_{21}=0}^{i_2} \sum_{m_{22}=0}^{i_2 - m_{21}} F_{m_{31}, m_{32}}^{k, (v_0, v_1)}(A_{31}^{+k}, A_{32}^{+k}) \\
&\times i_1! i_2! (B_1^{+k})^{i_1 - m_{11} - m_{12}} (B_2^{+k})^{i_2 - m_{21} - m_{22}} \\
&\times \frac{(A_{11}^{+k})^{m_{11}} (A_{12}^{+k})^{m_{12}} (A_{21}^{+k})^{m_{21}} (A_{22}^{+k})^{m_{22}}}{m_{11}! m_{12}! m_{21}! m_{22}! (i_1 - m_{11} - m_{12})! (i_2 - m_{21} - m_{22})!}, \quad (25)
\end{aligned}$$

of the terms,

$$\begin{aligned}
F_{\bar{n}}^{k, (v_0, v_1)}(A_{31}^{+k}, A_{32}^{+k}) &= \frac{-n_1}{A_{31}^{+k}} F_{\bar{n} - (1, 0)}^{k, (v_0, v_1)}(A_{31}^{+k}, A_{32}^{+k}) - \left( \frac{\bar{\delta}_{n_1, 0}}{A_{31}^{+k} A_{32}^{+k} \alpha_3^{+k}} \right) \\
&\times \left[ (x_2^1)^{n_2} e^{A_{32}^{+k} x_2^1} - (x_2^0)^{n_2} e^{A_{32}^{+k} x_2^0} - n_2 I_{n_2 - 1}^{k, (v_0, v_1)}(A_{32}^{+k}) \right] \\
&+ \frac{1}{A_{31}^{+k} \alpha_3^{+k}} e^{A_{31}^{+k} (x_1^0 - \delta_{v_0, v_1} x_2^0)} \sum_{n=0}^{n_1} \left[ \frac{n_1! (\delta_{v_0, v_1})^n}{n! (n_1 - n)!} \right. \\
&\times \left. (x_1^0 - \delta_{v_0, v_1} x_2^0)^{n_1 - n} I_{n + n_2}^{k, (v_0, v_1)}(A_{32}^{+k} + \delta_{v_0, v_1} A_{31}^{+k}) \right], \quad n_1 > 0, \quad (26)
\end{aligned}$$

which is a recursion relation initialized by,

$$\begin{aligned}
F_{(0, n_2)}^{k, (v_0, v_1)}(A_{31}^{+k}, A_{32}^{+k}) &= \frac{1}{A_{31}^{+k} \alpha_3^{+k}} \left[ e^{A_{31}^{+k} (x_1^0 - \delta_{v_0, v_1} x_2^0)} I_{n_2}^{k, (v_0, v_1)}(A_{32}^{+k} + \delta_{v_0, v_1} A_{31}^{+k}) \right. \\
&\left. - I_{n_2}^{k, (v_0, v_1)}(A_{32}^{+k}) \right], \quad n_2 \geq 0. \quad (27)
\end{aligned}$$

In Eqs. (25), (26), and (27),  $\bar{\delta}_{n_1, 0}$  is the Kronecker delta, and we have defined,

$$I_n^{k, (v_0, v_1)}(A) \equiv \frac{1}{A} \left[ (x_2^1)^n e^{A x_2^1} - (x_2^0)^n e^{A x_2^0} - n I_{n-1}^{k, (v_0, v_1)}(A) \right], \quad n > 0, \quad (28)$$

a recursion formula initialized by,

$$I_0^{k, (v_0, v_1)}(A) \equiv \frac{(x_2^1)^{n+1} - (x_2^0)^{n+1}}{n+1}; \quad (29a)$$

$$A_{11}^{+k} \equiv \frac{\mu_2 \alpha_2^{-k} \alpha_3^{+k} + \mu_1 \alpha_3^{-k} \alpha_1^{+k} + \mu_3 \alpha_3^{-k} \alpha_3^{+k}}{\alpha_3^{+k} (\hat{\alpha}^{-k} \cdot \hat{\Omega})}, \quad (29b)$$

$$A_{12}^{+k} \equiv \frac{\mu_1 (\alpha_3^{-k} \alpha_2^{+k} - \alpha_2^{-k} \alpha_3^{+k})}{\alpha_3^{+k} (\hat{\alpha}^{-k} \cdot \hat{\Omega})}, \quad (29c)$$

$$B_1^{+k} \equiv \frac{\mu_2 (\beta^{-k} \alpha_3^{+k} - \alpha_3^{-k} \beta^{+k})}{\alpha_3^{+k} (\hat{\alpha}^{-k} \cdot \hat{\Omega})}, \quad \alpha_3^{+k} \neq 0, \quad (29d)$$

with analogous expressions for  $A_{n1}^{+k}, A_{n2}^{+k}, B_n^{+k}$ ,  $n = 2, 3$ . The accumulation of  $F_{\vec{n}}$  into  $f_{m_{31}, m_{32}}$  is conducted in the same way and same cyclic sense as for  $Y_{\vec{n}}^k$ .

The volumetric contribution to the outgoing face flux moments is given by,

$$\xi_i = \left( \frac{e^{L_3}}{\hat{\alpha}^{-k} \cdot \hat{\Omega}} \right) \sum_{m_{31}=0}^{2\Lambda} \sum_{m_{32}=0}^{2\Lambda} \sum_{m_{33}=0}^{2\Lambda} w_{m_{31}, m_{32}, m_{33}} s_{m_{31}, m_{32}, m_{33}}, \quad (30)$$

where the coefficients  $w_{m_{31}, m_{32}, m_{33}}$  are computed in an analogous manner to  $Z_{\vec{n}}$  described earlier,

$$\begin{aligned} w_{m_{31}, m_{32}, m_{33}} &\equiv \sum_{k=0}^3 \sum_{(v_0, v_1)} \sum_{m_{11}=0}^{i_1} \sum_{m_{12}=0}^{i_1 - m_{11}} \sum_{m_{13}=0}^{i_1 - m_{11} - m_{12}} \sum_{m_{21}=0}^{i_2} \sum_{m_{22}=0}^{i_2 - m_{21}} \sum_{m_{23}=0}^{i_2 - m_{21} - m_{22}} \\ & W_{m_{31}, m_{32}, m_{33}}^{k, (v_0, v_1)} \frac{i_1! i_2! (L_1)^{i_1 - m_{11} - m_{12} - m_{13}} (L_2)^{i_2 - m_{21} - m_{22} - m_{23}}}{m_{11}! m_{12}! m_{13}! m_{21}! m_{22}! m_{23}!} \\ & \times \frac{(H_{11})^{m_{11}} (H_{12})^{m_{12}} (H_{13})^{m_{13}} (H_{21})^{m_{21}} (H_{22})^{m_{22}} (H_{23})^{m_{23}}}{(i_1 - m_{11} - m_{12} - m_{13})! (i_2 - m_{21} - m_{22} - m_{23})!}, \quad (31) \end{aligned}$$

a linear combination of the terms,

$$\begin{aligned} W_{\vec{n}}^{k, (v_0, v_1)} &= -\frac{n_3}{H_{33}} W_{\vec{n} - (0, 0, 1)}^{k, (v_0, v_1)} - \bar{\delta}_{n_3, 0} \frac{\alpha_3^{+k}}{H_{33}} F_{(n_1, n_2)}^{k, (v_0, v_1)}(H_{31}, H_{32}) \\ & + \frac{(\alpha_3^{+k})^{1-n_3}}{H_{33}} \sum_{n=0}^{n_3} \sum_{n'=0}^{n_3-n} \left[ \frac{n_3! (-\alpha_1^{+k})^n (-\alpha_2^{+k})^{n'} (\beta+k)^{n_3-n-n'} e^{H_{33} \beta+k} / \alpha_3^{+k}}{n! n'! (n_3 - n - n')!} \right] \\ & \times F_{(n_1+n, n_2+n')}^{k, (v_0, v_1)} \left( H_{31} - \frac{H_{33} \alpha_1^{+k}}{\alpha_3^{+k}}, H_{32} - \frac{H_{33} \alpha_2^{+k}}{\alpha_3^{+k}} \right), \quad n_3 > 0, \quad (32) \end{aligned}$$

a recursion relation initialized by,

$$\begin{aligned} W_{(n_1, n_2, 0)}^{k, (v_0, v_1)} &= \frac{n_3}{H_{33}} \left[ e^{H_{33} \beta+k} / \alpha_3^{+k} F_{(n_1, n_2)}^{k, (v_0, v_1)} \left( H_{31} - \frac{H_{33} \alpha_1^{+k}}{\alpha_3^{+k}}, H_{32} - \frac{H_{33} \alpha_2^{+k}}{\alpha_3^{+k}} \right) \right. \\ & \left. - F_{(n_1, n_2)}^{k, (v_0, v_1)}(H_{31}, H_{32}) \right], \quad n_1, n_2 \geq 0. \quad (33) \end{aligned}$$

In Eqs. (31), (32), and (33), we have defined,

$$H_{11} \equiv \frac{\mu_2 \alpha_2^{-k} + \mu_3 \alpha_3^{-k}}{\hat{\alpha}^{-k} \cdot \hat{\Omega}}, \quad (34a)$$

$$H_{12} \equiv -\frac{\mu_1 \alpha_2^{-k}}{\hat{\alpha}^{-k} \cdot \hat{\Omega}}, \quad (34b)$$

$$H_{13} \equiv -\frac{\mu_1 \alpha_3^{-k}}{\hat{\alpha}^{-k} \cdot \hat{\Omega}}, \quad (34c)$$

$$L_1 \equiv -\frac{\mu_1 \beta^{-k}}{\hat{\alpha}^{-k} \cdot \hat{\Omega}}, \quad (34d)$$

with analogous definitions for  $H_{n1}, H_{n2}, H_{n3}, L_n, n = 2, 3$ .

Conditions are also derived to determine the number of CTs an arbitrary tetrahedron is comprised of for a given discrete ordinate, and for computing the four vertices of such CT. A mesh sweep then amounts to taking one tetrahedron at a time where the flux moments on its incoming face(s) are known from boundary conditions or from the previous cell, dividing it into CTs, then applying Eq. (23) to compute the flux moments on its outgoing face(s). This is followed by applying the balance equation, Eq. (3), to compute the cell-moments of the angular flux, then Eqs. (10) and (11) are applied before proceeding to the next tetrahedron. Upon completing the mesh sweep in all employed discrete ordinates the new iterate of the scattering source is compared to the previous iterate before testing the stopping criterion.

### 3 A COMPUTATIONALLY AND MEMORY EFFICIENT SWEEP ALGORITHM

The goals for AHOT-C-UG's sweep algorithm are three-fold: preserve the natural discrete ordinates recursive ordering, minimize the amount of additional computation needed to compute this ordering and limit the amount of memory required. The usual formulation of the discrete ordinates method requires that the particles be tracked through the mesh in the direction of their natural flow. This results in a computationally efficient method since each value used in the recursion is the most recently computed. This is trivial for the finite difference meshes used in most discrete ordinates codes since the ordinates are naturally separated into octants by the regular mesh organization and they can be processed in a strictly predetermined sequence.

On an unstructured mesh, it is not possible to know a priori the exact order in which particles will flow through the cells. For example, a small change in the orientation of a face can change it from an outgoing into an incoming face for a given ordinate and cell. Therefore, each ordinate will have its own optimal mesh sweep order. One approach is to precompute and save the optimal mesh sweep order for each ordinate. This approach results in an efficient mesh sweep, however, it has a high memory requirement, namely the memory required to store an array of cell indices equal to the number of cells times the number of ordinates.

In (Suslov, 1997), the cells of the unstructured mesh were actually reoriented so that a deterministic order could be imposed on the mesh. This approach had two limitations: it depended on a specific organization of the mesh cells and it was computationally expensive to reorient all the cells. For an unstructured mesh representation to be useful, it must not be limited to finite difference-like organizations.

Instead of precomputing the sweep order, AHOT-C-UG's sweep algorithm

dynamically determines the order for each ordinate on-the-fly throughout the calculation. At the start of the mesh sweep for each ordinate, each cell's faces are marked as to whether they are incoming or outgoing relative to the ordinate. Then, all of the cells in the problem are placed in a circular queue. The queue is continuously traversed and each cell is examined to determine if the angular flux on all of its incoming faces have been computed. When all of a cell's incoming face moments have been calculated, its average moments and outgoing face moments are computed and the outgoing faces are marked as having valid moments. The cell is then removed from the queue. The data structure which holds the face data is shared by the cells which have the face in common, therefore, during the next pass through the queue, the adjoining cell's new incoming flux moments will be available. When the queue is empty, the mesh sweep is complete for that ordinate.

To make the AHOT-C-UG sweep algorithm efficient, the face and cell lists are inter-linked. This results in additional memory requirements above those which are strictly necessary to represent the unstructured mesh. Per cell and per face memory must be used to store both the solution and the bookkeeping information about the mesh. Each face stores a pointer to the next face and a label (these values are only used during the input processing phase, however), the indices of the three vertices which form the face, its  $\alpha$  and  $\beta$  values, the indices of the one or two cells which include this face, a flag stating whether the face moments have been computed and a pointer to the memory where the face moments are stored. A cell data structure includes a pointer to the next cell, the volume of the cell, the index of the material region to which the cell belongs, pointers to the four faces which comprise the cell, the indices of the four vertices which form the cell, four flags which mark each face as incoming or outgoing and pointers to the memory used for the new flux iterate, the old flux iterate and the distributed source. The memory requirements are summarized in Table 1. Some of the bookkeeping values could be recomputed on-the-fly or eliminated after the input processing phase, however, such trade-offs have not been investigated.

**Table 1.** Variable Count in Words by Type for the Lowest Order Approximation in AHOT-C-UG.

per	index(integer) 4 Bytes	boolean 1 Byte	pointer 4 Bytes	floating point 8 Bytes
Face	6	1	2	4
Cell	5	4	8	1

On the machine on which AHOT-C-UG was tested, integers are 4 bytes, booleans are 1 byte, pointers are 4 bytes and floating point numbers are 8 bytes. Thus, the size of the per face structure is 68 bytes (note that the boolean value is

rounded up to 4 bytes for alignment purposes) and the size of the per cell structure is 64 bytes. Noting that the number of face moments goes as the AHOT order plus one squared and the number of cell moments goes as the AHOT order plus one cubed, the total memory requirement of AHOT-C-UG is:

$$\text{Memory (bytes)} = \mathcal{F} \times (68 + (2\Lambda + 1)^2 \times 8) + \mathcal{C} \times (64 + 3 \times (\Lambda + 1)^3 \times 8), \quad (35)$$

where  $\mathcal{F}$ , and  $\mathcal{C}$  are the number of faces, and the number of cells in the tetrahedral mesh.

In the limit of infinitely many tetrahedral cells, there are about twice as many faces as there are cells,  $\mathcal{F} = 2\mathcal{C}$ , so the order of the memory requirement for 0-order is approximately:

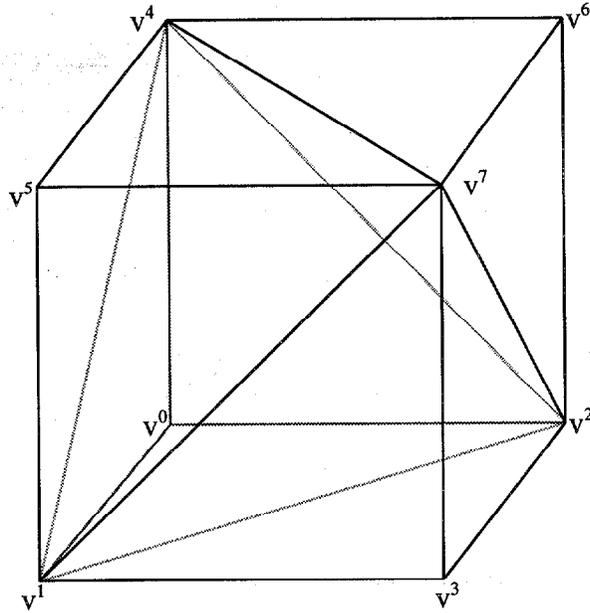
$$\text{Memory} = 2 \times (68 + 1 \times 8) + 1 \times (64 + 3 \times 1 \times 8) = 240 \text{ (bytes/cell)}. \quad (36)$$

#### 4 NUMERICAL TESTS AND RESULTS

The AHOT-C method presented in Sec. 2 and the sweep algorithm described in Sec. 3 were implemented in a program, AHOT-C-UG, written in the C computer language. Although C is not typically used for transport codes, the GNU C compiler (GCC) contains a number of extensions to the ANSI C language which simplify its use in scientific computing. In particular, GCC allows local arrays to be declared with variable dimensions and allows multidimensional arrays to be declared as arguments to subroutines (features which are indispensable in FORTRAN). AHOT-C-UG was compiled with GCC version 2.95.2 and tested on a 700MHz Intel Pentium III system running Linux kernel 2.4.0 and GNU C library version 2.1.3.

The test problem solved here is a simple, uniform cube, three mean free paths (MFP) on an edge, with vacuum boundary conditions on all external faces. Scattering is isotropic, an  $S_2$  quadrature is used and the source is uniform throughout the volume. In order to compare to a TORT solution, the cube is broken into subcubes exactly like a finite difference mesh and then the subcubes are tessellated into five tetrahedra. The basic cube tessellation is depicted in Fig. 1. Note that this tessellation has limited symmetry; the tessellation of adjacent cubes must be rotated 90 degrees (along any axis) in order for the faces of adjacent cells to coincide. Note, too, that the interior tetrahedron has twice the volume of the corner tetrahedra.

A series of mesh refinements were studied starting with a  $3 \times 3 \times 3$  discretization into cubes of one MFP edge length. The cubes were then halved in each dimension, resulting in eight new cubes each. The specifics of each test problem are given in Table 2. Note that the characteristic length of the edges of the tetrahedra are halved in each problem.



**Fig. 1.** Tessellation of the Basic Cube Into Five Tetrahedra.

For the largest problems, the wall clock times show that the method requires about 0.2 ms per cell-angle per sweep. The table also shows that there is about 10% growth in the time per cell-angle as the mesh is increased by a factor of eight. This demonstrates that the mesh sweep algorithm is relatively insensitive to the size of the mesh.

Comparison between the AHOT-C-UG solutions and the reference TORT solution for three representative values of the scattering ratio,  $c$ , are presented in Table 3. The TORT configuration had 96 mesh cells along each axis and used theta-weighted differencing scheme (Rhoades and Simpson, 1997). Both the AHOT-C-UG and TORT solutions were integrated on to a one MFP cube mesh for comparison. The results show that, as the characteristic length of the AHOT-C-UG mesh cells are halved in size, the error is also approximately halved, as expected in the zero-order method.

Preliminary testing of the high order approximations implemented in AHOT-C-UG has just started. While a complete evaluation of these is premature at this point, we include in Table 4 results of initial experiments with the linear order approximation for the same test problem described above. These results show a two orders of magnitude reduction in the maximum error on the same mesh compared to the constant approximation. However, the increase in memory requirement, less than a factor of three, and execution time, two orders of magnitude, is rather large and will be examined more closely. It is worth noting here that the arbitrary order imple-

**Table 2.** Specifications for Test Problems, Total Memory Requirement, and Total Execution Time Divided by the Number of Cell-Angles for AHOT-C-UG with the Constant Approximation.

Mesh	Vertices	Cells	Memory (MBytes)	$c = 0.1$	$c = 0.5$	$c = 0.9$
				ms	ms	ms
$3 \times 3 \times 3$	64	135	0.036	0.19	0.20	0.22
$6 \times 6 \times 6$	343	1,080	0.276	0.16	0.15	0.16
$12 \times 12 \times 12$	2,197	8,640	2.139	0.16	0.16	0.16
$24 \times 24 \times 24$	15,625	69,120	16.851	0.17	0.17	0.17
$48 \times 48 \times 48$	117,649	552,960	133.761	0.19	0.18	0.19

**Table 3.** Maximum % Difference Between AHOT-C-UG with the Constant Approximation and the TORT Reference Solution with Theta-Weighted Differencing; Flux Values Integrated Over  $3 \times 3 \times 3$  Cells for Each Mesh.

Mesh	$c = 0.1$	$c = 0.5$	$c = 0.9$
$3 \times 3 \times 3$	10.35	13.52	19.71
$6 \times 6 \times 6$	6.43	8.43	12.41
$12 \times 12 \times 12$	3.73	6.40	7.19
$24 \times 24 \times 24$	2.08	4.88	3.96
$48 \times 48 \times 48$	1.12	2.71	2.10

mentation is punitive to execution time for low order approximations, like constant and linear, because of the latency attributed to loops over order which is less significant in high order approximations. In other words, we conjecture that *fixed* constant- and linear-order-approximation versions of AHOT-C-UG would execute faster than reported here, but would be limited to the specific order implemented.

## 5 CONCLUSION

We have demonstrated the feasibility and accuracy of the AHOT-C method for solving the neutron transport equation on three-dimensional unstructured grids with tetrahedral cells. Comparison of the new method solutions obtained with the new AHOT-C-UG code to those obtained on very fine meshes with the TORT illustrate the high accuracy of our method. The computational resources required by AHOT-C-UG to solve very large problems, i.e. half a million tetrahedra, appears quite reasonable both in terms of execution time and memory size. Preliminary results

**Table 4.** Performance of AHOT-C-UG with the Linear Approximation for the Test Problem.

Mesh	Memory (MBytes)	$c = 0.1$		$c = 0.5$		$c = 0.9$	
		ms	% Error	ms	% Error	ms	% Error
$3 \times 3 \times 3$	0.080	77.0	0.092	77.1	0.082	77.0	0.099
$6 \times 6 \times 6$	0.609	77.2	0.035	77.2	0.034	77.1	0.032

for high order approximations, already implemented in AHOT-C-UG, demonstrate substantial improvement in accuracy in case of the linear approximation.

### References

- Azmy, Y. Y., 1992. Arbitrarily High Order Characteristic Methods for Solving the Neutron Transport Equation. *Annals of Nuclear Energy* **19**, 593.
- Suslov, I. R., 1997. Final Report to the National Research Council's Radioactive Waste Management Program for 1996. Unpublished report.
- Rhoades, W. A., Simpson, D. B., 1997. The TORT Three-Dimensional Discrete Ordinates Neutron/Photon Transport Code, ORNL/TM-13221, Oak Ridge National Laboratory, 1997.
- GCC. GNU Compiler Collection, <http://gcc.gnu.org>.

