

Discontinuous Finite Element S_N Methods on Three-Dimensional Unstructured Grids

Todd A. Wareing, John M. McGhee, Jim E. Morel, and Shawn D. Pautz

*Los Alamos National Laboratory, Transport Methods Group, XTM
P.O. Box 1663, MS D409, Los Alamos, New Mexico 87545*

Received January 26, 2000

Accepted January 15, 2001

Abstract—*Discontinuous finite element methods for the S_N equations on three-dimensional unstructured tetrahedral and hexahedral meshes are presented. Solution techniques including source iteration and diffusion-synthetic acceleration are described. Numerical results are presented that demonstrate the accuracy and efficiency of these methods.*

I. INTRODUCTION

The purpose of this paper is to present methods for solving the S_N equations on three-dimensional unstructured tetrahedral and hexahedral meshes using the discontinuous finite element method (DFEM) for the spatial discretization. The majority of DFEMs developed within the reactor physics community have been rectangular mesh methods. Notable exceptions are the methods used in the TRIPLET (Ref. 1), TRIDENT (Ref. 2), and ZEPHYR (Ref. 3) S_N codes. TRIPLET and TRIDENT are two-dimensional triangular mesh codes. Their meshes are actually semistructured rather than fully unstructured because the triangles are located on bands. The ZEPHYR code has a fully unstructured mesh consisting of arbitrary combinations of quadrilaterals and triangles. There are two significant complications encountered when applying DFEMs to the S_N equations on two-dimensional unstructured meshes that are not encountered when applying them on two-dimensional rectangular meshes:

1. For quadrilaterals, the finite element matrix elements must be evaluated by quadrature because they cannot be analytically integrated.

2. The lower triangular ordering of the angular flux unknowns required to solve the source iteration equations via the back-substitution or sweeping technique is mesh dependent. Thus, this ordering must be explicitly determined for each direction via a computational algorithm. On rectangular meshes, this ordering depends on direction, but it is mesh independent and trivial to recognize.

It has been observed that a lower triangular ordering exists on all two-dimensional unstructured meshes as long as the mesh is non-reentrant as a whole, and each element (spatial cell) within the mesh is non-reentrant.

Further complications are encountered when DFEMs are applied on three-dimensional unstructured meshes:

1. Because the faces of general hexahedra can be nonplanar, a given direction can be incident on one part of an element face and exiting on the other part. This causes the finite element definition of the angular flux to change on the interior of element faces, whereas such changes never occur with flat faces. These face-interior changes require separate matrix-element integrations within the incident and exiting regions of the face because each region has a different basis function representation for the flux. Because standard finite element quadratures are intended to integrate a single polynomial flux representation over a whole face, they can be very inaccurate for performing these double-representation integrations. Furthermore, having directions that are both incident and exiting on the same face make a lower triangular ordering impossible because the two elements sharing that face become mutually dependent.

2. We initially assumed that if a three-dimensional unstructured mesh were non-reentrant as a whole, and if all of the elements in the mesh were non-reentrant, a lower triangular ordering of the angular flux unknowns would exist. However, we have found that this is not necessarily true. It is possible for “rings” of mutual

dependency to form on three-dimensional unstructured meshes even if the previously described mesh criteria are met. When this occurs, a lower triangular ordering of the angular flux unknowns does not exist.

Thus, if one is to develop a DFEM S_N method for three-dimensional unstructured meshes, one must deal with the complications just described. Our strategy for dealing with them relies on two facts:

1. These complications will rarely occur on well-shaped, i.e., not highly skewed, meshes.

2. Directions that are both incident and exiting on a face are essentially parallel to the face. Fluxes exactly parallel to a flat face yield a zero solution within the element. Thus, we expect that such approximations made for directions nearly parallel to the faces will only weakly affect the overall solution.

The full details are given later, but a high-level description of our strategy can be given as follows: We deal with a direction that is both incident and exiting on the same face by defining the direction to be either incident or exiting over the whole face via a single average face normal. The finite element flux representation is then uniquely defined over the entire face. Furthermore, this solves the local problem of mutual coupling between the elements that share the face, but global dependency rings can still form. We use graph theory to identify mutually dependent rings of elements in the sweep ordering process. The mutual dependence is effectively broken by assuming that the incoming fluxes for one element in the ring are known. This allows the ordering process to continue even though the resulting system of source iteration equations is not lower triangular. Because the equations are not lower triangular, an exact solution is not obtained after each sweep is performed.

The remainder of this paper is organized as follows: in Sec. II, we describe the DFEM spatial differencing of the S_N equations on three-dimensional unstructured meshes; in Sec. III, we describe the solution method for solving these DFEM S_N equations; in Sec. IV, we present some numerical results; and in Sec. V we present some conclusions.

II. DFEM SPATIAL DIFFERENCING

We first consider the multigroup S_N equations for domain V and boundary δV and an isotropic fixed source:

$$\underline{\Omega}_m \cdot \underline{\nabla} \Psi_{m,g}(\underline{r}) + \sigma_{t,g}(\underline{r}) \Psi_{m,g}(\underline{r}) = S_{m,g}(\underline{r}) + Q_{m,g}(\underline{r}) , \quad (1)$$

$$S_{m,g}(\underline{r}) = \sum_{g'=1}^G \sum_{l=1}^L (2l+1) \sigma_{s,g' \rightarrow g}^l(\underline{r}) \times \sum_{n=-l}^l Y_{l,n}(\underline{\Omega}_m) \phi_{l,g'}^n(\underline{r}) , \quad (2)$$

and

$$\Psi_{m,g}(\underline{r}) = F_{m,g}(\underline{r}) , \quad \underline{n} \cdot \underline{\Omega}_m < 0 , \quad (3)$$

where

m, g = angle and group indices

Ψ = discrete-ordinates angular flux

F = incident angular flux

σ_t = macroscopic total cross section

σ_s = macroscopic scattering cross section

S = scattering source represented as an expansion of spherical harmonics ($Y_{l,n}$)

Q = fixed source

\underline{n} = outward directed unit normal vector of δV .

We need only consider isotropic scattering in the development of the DFEM equations, but this is not a limitation. Expanding the $\underline{\nabla}$ operator using the Einstein summation convention (i.e., a repeated index in the same multiplicative term implies a summation) and suppressing the m and g subscripts, Eq. (1) for a single energy group in three-dimensional Cartesian geometry becomes

$$\Omega_i \frac{\partial}{\partial r_i} \Psi(\underline{r}) + \sigma_t(\underline{r}) \Psi(\underline{r}) = \sigma_s(\underline{r}) \Phi(\underline{r}) + Q(\underline{r}) , \quad (4)$$

where the individual components of $\underline{\Omega}$ have been written as Ω_i and the individual components of $\underline{\nabla}$ have been written as $\partial/\partial r_i$, and where $r_1 = x$, $r_2 = y$, and $r_3 = z$. The indices run from one to the number of spatial dimensions. Unless otherwise noted, this summation convention will be used in all the equations that follow.

II.A. Discontinuous Finite Element Formulation

We begin the development by assuming that the problem domain has been divided into an unstructured spatial grid of volume elements (spatial cells). The elements shapes are unspecified for now, but we do require that the vertices be connected by straight lines. The material properties within each element are assumed to be constant. The DFEM derivation begins by considering element k , with volume V_k and surface δV_k . We define approximate angular and scalar flux functions within element k in terms of a linearly independent set of basis functions $[\gamma_p(\underline{r})]$, $1 \leq p \leq P_k$, where p is the vertex index and P_k is the number of vertices per element. These approximate functions take the following form:

$$\Psi(\underline{r}) \cong \psi(\underline{r}) = \underline{\Theta}^T(\underline{r}) \underline{\psi} \quad (5)$$

and

$$\Phi(\underline{r}) \cong \phi(\underline{r}) = \underline{\Theta}^T(\underline{r}) \underline{\phi} , \quad (6)$$

where

$$\underline{\Theta} = [\gamma_1(r), \dots, \gamma_{P_k}(r)]^T, \quad (7)$$

$$\underline{\psi} = [\psi_1, \dots, \psi_{P_k}]^T, \quad (8)$$

and

$$\underline{\phi} = [\phi_1, \dots, \phi_{P_k}]^T. \quad (9)$$

The basis functions are chosen to give appropriate vertex displacements when the coordinates of corresponding vertices p are inserted into Eqs. (5) and (6). That is, $\gamma_p(r) = 1$ at $r = (x_p, y_p, z_p)$ and zero at the other vertices. Here $\underline{\psi}$ and $\underline{\phi}$ are column vectors of vertex angular and scalar fluxes, respectively. Next, we apply the Galerkin method, which consists of inserting Eqs. (5) and (6) into Eq. (4), multiplying by $\underline{\Theta}$ and integrating over V_k . This operation guarantees the orthogonality of the residuals to the space spanned by the spatial basis functions, thus minimizing, in a certain sense, the error introduced by the approximation introduced in Eq. (5). Carrying out this operation and using the divergence theorem, we obtain:

$$\begin{aligned} & \int_{\delta V_k} \Omega_i n_i \underline{\Theta} \psi^s(r) d\delta V - \int_{V_k} \Omega_i \frac{\partial \underline{\Theta}}{\partial r_i} \underline{\Theta}^T \underline{\psi} dV \\ & + \int_{V_k} \underline{\Theta} \{ \sigma_t \underline{\Theta}^T \underline{\psi} - \sigma_s \underline{\Theta}^T \underline{\phi} - q(r) \} dV = 0. \end{aligned} \quad (10)$$

Here, $n_i(r)$ is the i 'th component of the outward directed unit normal vector to δV and $\psi^s(r)$ is the angular flux on the element boundary. For each element,

$$\delta V_k = \sum_{l=1}^{N_{faces}} \delta V_{k,l}, \quad (11)$$

where l represents the face number and N_{faces} is the total number of faces enclosing element k , for example, $N_{faces} = 4$ for tetrahedral elements and $N_{faces} = 6$ for hexahedral elements.

To complete the derivation we need to define the element boundary angular fluxes. Here we assign angular flux functions on the upstream side of the boundary as follows:

$$\psi^s(r) = \underline{\Theta}^T(r) \underline{\psi}^{s,l}, \quad (12)$$

where for the l 'th face,

$$\underline{\psi}^{s,l} = \begin{cases} \underline{\psi} & \Omega_i n_i^{av,l} > 0 \\ \underline{\psi}^{inc} & \Omega_i n_i^{av,l} < 0 \end{cases}. \quad (13)$$

Here, $\underline{\psi}^{inc}$ is the corresponding column vector of vertex angular flux values of the element that shares the l 'th face of element k and $\underline{n}^{av,l}$ is the average unit normal vector of the l 'th face, defined by

$$\underline{n}^{av,l} = \frac{\underline{A}^l}{\|\underline{A}^l\|} \quad (14)$$

where

$$\underline{A}^l = \int_{\delta V_{k,l}} \underline{n}^l d\delta V. \quad (15)$$

Equation (13) requires some explanation. For elements with planar faces (tetrahedra) these definitions are consistent with the finite element formalism. For nonplanar faces, the unit normal vector to the face is not constant across the face, and a given direction can be both incident and exiting to the face. This results in a mutual dependency between the elements that share the face and thereby makes a lower triangular ordering impossible. If the unit normal vector to a face does not change sign, then the use of an average normal for defining incident and exiting fluxes remains fully consistent with the finite element formalism. If the unit normal vector to a face does change sign, we define each given direction to either be incident or exiting across the entire face based on the average outward directed unit normal vector to the face. This is not consistent with the finite element formalism and therefore represents an additional approximation.

The DFEM equations for each element are then given by

$$\sum_{l=1}^{N_{faces}} \Omega_i \underline{U}_i^l \underline{\psi}^{s,l} + (-\Omega_i \underline{K}_i + \sigma_t \underline{M}) \underline{\psi} = \sigma_s \underline{M} \underline{\phi} + \underline{q}, \quad (16)$$

where

$$\underline{U}_i^l = \int_{\delta V_{k,l}} n_i \underline{\Theta} \underline{\Theta}^T d\delta V, \quad (17)$$

$$\underline{K}_i = \int_{V_k} \frac{\partial \underline{\Theta}}{\partial r_i} \underline{\Theta}^T dV, \quad (18)$$

$$\underline{M} = \int_{V_k} \underline{\Theta} \underline{\Theta}^T dV, \quad (19)$$

and

$$\underline{q} = \int_{V_k} \underline{\Theta} q(r) dV. \quad (20)$$

We note that for our implementation of the methods, we have only used linear basis functions for tetrahedral elements and trilinear basis functions for hexahedral elements. This leads to four spatial unknowns within each tetrahedron and eight unknowns in each hexahedron. Thus, hexahedral elements will be proportionally more costly than tetrahedral elements in terms of both CPU time and memory requirements. However, for many

problems, fewer hexahedral elements will be required to model the geometry than tetrahedral elements.

II.B. Spatial Derivatives and Integrals

In general, the volume elements will be nonorthogonal, and it becomes necessary to transform from global coordinates $\underline{r} = (x, y, z)$, with difficult limits of integration, to local coordinates $\underline{\tilde{r}} = (\tilde{x}, \tilde{y}, \tilde{z})$, with simple limits of integration. The local coordinate system is defined such that the physical cell takes on a simple shape and size. For example, every hexahedron is a unit cube in the local coordinate system. The most convenient method of establishing the coordinate transformations is to use the local basis functions to represent the variation of the unknown coordinate. If, for instance, we write for each element:

$$x(\underline{\tilde{r}}) = \underline{\Theta}^T(\underline{\tilde{r}}) \begin{bmatrix} x_1 \\ \vdots \\ x_{P_k} \end{bmatrix} = \underline{\Theta}^T(\underline{\tilde{r}}) \underline{x} , \quad (21)$$

$$y(\underline{\tilde{r}}) = \underline{\Theta}^T(\underline{\tilde{r}}) \begin{bmatrix} y_1 \\ \vdots \\ y_{P_k} \end{bmatrix} = \underline{\Theta}^T(\underline{\tilde{r}}) \underline{y} , \quad (22)$$

and

$$z(\underline{\tilde{r}}) = \underline{\Theta}^T(\underline{\tilde{r}}) \begin{bmatrix} z_1 \\ \vdots \\ z_{P_k} \end{bmatrix} = \underline{\Theta}^T(\underline{\tilde{r}}) \underline{z} , \quad (23)$$

where, \underline{x} , \underline{y} , and \underline{z} are column vectors of the vertex coordinates, immediately a relationship of the required form is available. It is necessary to have some means of expressing the global spatial derivatives in terms of local derivatives. This is done in the literature⁴ and only the final result is given. Thus,

$$dV = dx dy dz = \det \underline{\underline{\mathcal{J}}}(\underline{\tilde{r}}) d\tilde{x} d\tilde{y} d\tilde{z} = \det \underline{\underline{\mathcal{J}}}(\underline{\tilde{r}}) d\tilde{V} \quad (24)$$

and

$$\frac{\partial}{\partial r_i} = \mathcal{J}_{i,j}^{-1}(\underline{\tilde{r}}) \frac{\partial}{\partial \tilde{r}_j} , \quad (25)$$

where $\underline{\underline{\mathcal{J}}}$ is the Jacobian matrix defined by

$$\underline{\underline{\mathcal{J}}}(\underline{\tilde{r}}) = \begin{bmatrix} \left(\frac{\partial \underline{\Theta}^T}{\partial \tilde{x}} \underline{x} \right) & \left(\frac{\partial \underline{\Theta}^T}{\partial \tilde{x}} \underline{y} \right) & \left(\frac{\partial \underline{\Theta}^T}{\partial \tilde{x}} \underline{z} \right) \\ \left(\frac{\partial \underline{\Theta}^T}{\partial \tilde{y}} \underline{x} \right) & \left(\frac{\partial \underline{\Theta}^T}{\partial \tilde{y}} \underline{y} \right) & \left(\frac{\partial \underline{\Theta}^T}{\partial \tilde{y}} \underline{z} \right) \\ \left(\frac{\partial \underline{\Theta}^T}{\partial \tilde{z}} \underline{x} \right) & \left(\frac{\partial \underline{\Theta}^T}{\partial \tilde{z}} \underline{y} \right) & \left(\frac{\partial \underline{\Theta}^T}{\partial \tilde{z}} \underline{z} \right) \end{bmatrix} . \quad (26)$$

For the boundaries we have

$$d\delta V = |\underline{\underline{\mathcal{J}}}(\underline{\tilde{r}}_s)| d\tilde{x}_s d\tilde{y}_s = |\underline{\underline{\mathcal{J}}}(\underline{\tilde{r}}_s)| d\delta \tilde{V} , \quad (27)$$

where the \tilde{x}_s and \tilde{y}_s coordinates are assumed to be the local coordinates for the element face and $|\underline{\underline{\mathcal{J}}}(\underline{\tilde{r}}_s)|$ is the magnitude of the surface Jacobian. This Jacobian is given by

$$\underline{\underline{\mathcal{J}}}(\underline{\tilde{r}}_s) = \begin{bmatrix} \left(\frac{\partial \underline{\Theta}^T}{\partial \tilde{x}_s} \underline{x} \right) \\ \left(\frac{\partial \underline{\Theta}^T}{\partial \tilde{x}_s} \underline{y} \right) \\ \left(\frac{\partial \underline{\Theta}^T}{\partial \tilde{x}_s} \underline{z} \right) \end{bmatrix} \times \begin{bmatrix} \left(\frac{\partial \underline{\Theta}^T}{\partial \tilde{y}_s} \underline{x} \right) \\ \left(\frac{\partial \underline{\Theta}^T}{\partial \tilde{y}_s} \underline{y} \right) \\ \left(\frac{\partial \underline{\Theta}^T}{\partial \tilde{y}_s} \underline{z} \right) \end{bmatrix} . \quad (28)$$

Equations (17) to (20) become

$$\underline{\underline{U}}_i^l = \int_{\delta \tilde{V}_{k,l}} n_i \underline{\Theta} \underline{\Theta}^T |\underline{\underline{\mathcal{J}}}(\underline{\tilde{r}}_s)|_l d\delta \tilde{V} , \quad (29)$$

$$\underline{\underline{K}}_i = \int_{\tilde{V}_k} \mathcal{J}_{i,j}^{-1}(\underline{\tilde{r}}) \frac{\partial \underline{\Theta}}{\partial \tilde{r}_j} \underline{\Theta}^T \det \underline{\underline{\mathcal{J}}}(\underline{\tilde{r}}) d\tilde{V} , \quad (30)$$

$$\underline{\underline{M}} = \int_{\tilde{V}_k} \underline{\Theta} \underline{\Theta}^T \det \underline{\underline{\mathcal{J}}}(\underline{\tilde{r}}) d\tilde{V} , \quad (31)$$

and

$$\underline{\underline{q}} = \int_{\tilde{V}_k} \underline{\Theta} q(\underline{\tilde{r}}) \det \underline{\underline{\mathcal{J}}}(\underline{\tilde{r}}) d\tilde{V} . \quad (32)$$

Because the limits of integration are fixed, we can perform these integrals with sufficient accuracy using Gauss quadrature.

III. SOLUTION TECHNIQUE

The DFEM equations are solved using source iteration in conjunction with diffusion-synthetic acceleration (DSA), as described by the following equations

$$\underline{\underline{\Omega}} \cdot \underline{\underline{\nabla}} \Psi^{(\ell+1/2)} + \sigma_t \Psi^{(\ell+1/2)} = \sigma_s \Phi^{(\ell)} + \underline{\underline{Q}} , \quad (33)$$

$$-\underline{\underline{\nabla}} \cdot \frac{1}{3\sigma_t} \underline{\underline{\nabla}} \delta \Phi^{(\ell+1)} + \sigma_a \delta \Phi^{(\ell+1)} = \sigma_s (\Phi^{(\ell+1/2)} - \Phi^{(\ell)}) , \quad (34)$$

and

$$\Phi^{(\ell+1)} = \Phi^{(\ell+1/2)} + \delta \Phi^{(\ell+1)} . \quad (35)$$

Here, ℓ is the iteration index. Equation (33) is the sweep equation, and Eq. (34) is the DSA equation.

III.A. Source Iteration and Transport Sweeps

The discretization of Eq. (33) in angle and space yields, for each angle in the discrete set, a system of equations whose associated matrix can be written in block lower triangular form (the sweep matrix), where each block is usually associated with the unknowns in a single spatial element. The dependencies between the elements for any angle can also be expressed in terms of a directed graph. If this sweep graph is acyclic, then each of the blocks in the matrix is associated with a single element; if the graph is cyclic then at least one of the blocks is associated with more than one element. In the latter case, the source iteration equations can no longer be exactly solved using a sweep.

Each system of DFEM S_N equations is solved by the method of sweeping. The spatial elements are placed into an ordered list in which the order results in a block lower triangular matrix. Each block is solved in succession. Physically this solution order resembles a wave front propagating through the mesh roughly in the direction of the associated angle. In the case of a cyclic sweep graph, a depth-first search algorithm⁵ is used to determine the identities of the elements in each cycle (associated with a large block in the matrix). These cycles are "broken" by approximating one or more unknown incoming face fluxes with the values from the previous source iteration; this approximation yields a modified sweep matrix in which each block refers to the unknowns of only one element. The sweeping order is established once and stored for subsequent sweeps.

Although most of the smooth meshes we have generated and examined have yielded only directed acyclic sweep graphs, we have observed cyclic graphs in both tetrahedral and hexahedral meshes. The use of previous iterate information has not noticeably degraded the performance or stability of the source iteration or DSA iterations. Presently, our algorithm for breaking the cycles has not been optimized; thus, if many cycles are present, a dramatic increase in run time can be expected because of the cost of searching for and breaking the cycles.

III.B. Acceleration Equations

Specific details of DSA are widely available in the literature.^{6,7} One very important and well-known fact about DSA is that the discretization of Eq. (34) must be consistent (or nearly consistent) with the discretization of Eq. (33). The four-step method⁷ is one way to obtain completely consistent differencing of Eq. (34) for all differencing schemes, but for advanced differencing schemes such as DFEMs in multidimensions, the resulting P_1 system of equations cannot be collapsed into a single discretized diffusion equation. Adams and Martin have proposed a modified four-step DSA method⁸ for DFEMs, where the diffusion differencing is obtained by applying a DFEM method to the diffusion equation. Their method leads to a single discretized diffusion equation,

which is both nonstandard and nonsymmetric. Wareing, Larsen, and Adams⁹ have developed yet another approach for obtaining nearly consistent DSA equations for DFEMs. This approach leads to a discretized diffusion equation that can be solved very efficiently; however, although the method is always stable, the effectiveness is significantly degraded for skewed and high-aspect ratio elements. We use an adaptation of the Wareing, Larsen, and Adams method.

To derive the discretization of Eq. (34), using an adaptation of the Wareing, Larsen, and Adams method, we begin by defining approximate scalar flux correction functions within element k in terms of the same linearly independent set of basis functions that we used with the discrete-ordinates equations:

$$\delta\Phi(r) \cong \delta\phi(r) = \underline{\Theta}^T(r) \underline{\delta\phi} . \quad (36)$$

If we multiply Eq. (34) by $\underline{\Theta}$, integrate over the volume V , apply the divergence theorem, and insert Eq. (36) into the resulting equations, we obtain:

$$\begin{aligned} & - \sum_{l=1}^{N_{\text{faces}}} \frac{1}{3\sigma_t} \int_{\delta V_{k,l}} n_i^l \underline{\Theta} \frac{\partial}{\partial r_i} \delta\phi^{s,(\ell+1)}(r) d\delta V \\ & + \frac{1}{3\sigma_t} \int_{V_k} \frac{\partial \underline{\Theta}}{\partial r_i} \frac{\partial \underline{\Theta}^T}{\partial r_i} \underline{\delta\phi}^{(\ell+1)} dV \\ & + \int_{V_k} \underline{\Theta} \{ \sigma_a \underline{\Theta}^T \underline{\delta\phi}^{(\ell+1)} - \sigma_s \underline{\Theta}^T (\underline{\phi}^{(\ell+1/2)} - \underline{\phi}^{(\ell)}) \} dV \\ & = 0 . \end{aligned} \quad (37)$$

All that is needed at this point is the definition of $\delta\phi^{s,(\ell+1)}(r)$, the scalar flux corrections on the boundary of the element. In the modified four-step method of Adams and Martin, one uses the partial current approach to define for the l' th face:

$$\begin{aligned} & - \frac{n_i^l}{3\sigma_t} \frac{\partial}{\partial r_i} \delta\phi^{s,(\ell+1)}(r) \\ & = \left\{ \frac{\underline{\Theta}^T \underline{\delta\phi}^{(\ell+1)}}{4} - \frac{n_i^l}{6\sigma_t} \frac{\partial \underline{\Theta}^T}{\partial r_i} \underline{\delta\phi}^{(\ell+1)} \right\} \\ & - \left\{ \frac{\underline{\Theta}^T \underline{\delta\phi}^{inc,(\ell+1)}}{4} + \frac{n_i^l}{6\sigma_t} \frac{\partial \underline{\Theta}^T}{\partial r_i} \underline{\delta\phi}^{inc,(\ell+1)} \right\} . \end{aligned} \quad (38)$$

Here, $\underline{\delta\phi}^{inc,(l+1)}$ is the corresponding column vector of vertex scalar flux correction values of the element that shares the l' th face of element k . This leads to a nonsymmetric DFEM for the diffusion equation that is nonstandard and cannot be solved efficiently with standard solvers. To simplify these equations, suppose we assume the scalar flux corrections are continuous at the

interelement boundaries and set $\delta\phi^{(\ell+1)} = \delta\phi^{inc,(\ell+1)} = \delta\phi_{cont}^{(\ell+1)}$ in Eq. (38). Then Eq. (37) becomes

$$\begin{aligned} & - \sum_{l=1}^{N_{faces}} \frac{1}{3\sigma_t} \int_{\delta V_{k,l}} n_i^l \underline{\Theta} \frac{\partial \underline{\Theta}^T}{\partial r_i} \delta\phi_{cont}^{(\ell+1)} d\delta V \\ & + \frac{1}{3\sigma_t} \int_{V_k} \frac{\partial \underline{\Theta}}{\partial r_i} \frac{\partial \underline{\Theta}^T}{\partial r_i} \delta\phi_{cont}^{(\ell+1)} dV \\ & + \int_{V_k} \underline{\Theta} \{ \sigma_a \underline{\Theta}^T \delta\phi_{cont}^{(\ell+1)} - \sigma_s \underline{\Theta}^T (\phi^{(\ell+1/2)} - \phi^{(\ell)}) \} dV \\ & = 0 . \end{aligned} \quad (39)$$

These equations represent the contribution from element k to the individual vertices forming element k . A global continuous finite element method (CFEM) matrix is formed for all vertices in the mesh by summing the individual element contributions using the standard finite element technique.⁴ This global matrix is a $N_{vertex} \times N_{vertex}$ symmetric positive-definite matrix, where N_{vertex} is the number of vertices in the mesh. Marshak boundary conditions are used for all boundary vertices. If one uses the CFEM differencing solely for the DSA equations—that is, assign the continuous scalar flux correction to the discontinuous scalar flux correction—the resulting method is stable but becomes ineffective for optically thick cells.

In the method of Wareing, Larsen, and Adams, the CFEM differencing is used in conjunction with a local mapping from continuous scalar flux corrections to discontinuous scalar flux corrections. We use an adaptation of this method, where this mapping is derived from the modified four-step method by assuming for the l 'th face that

$$\begin{aligned} \frac{\underline{\Theta}^T \delta\phi_{cont}^{(\ell+1)}}{2} &= \left\{ \frac{\underline{\Theta}^T \delta\phi^{(\ell+1)}}{4} - \frac{n_i^l}{6\sigma_t} \frac{\partial \underline{\Theta}^T}{\partial r_i} \delta\phi^{(\ell+1)} \right\} \\ &+ \left\{ \frac{\underline{\Theta}^T \delta\phi^{inc,(\ell+1)}}{4} + \frac{n_i^l}{6\sigma_t} \frac{\partial \underline{\Theta}^T}{\partial r_i} \delta\phi^{inc,(\ell+1)} \right\} . \end{aligned} \quad (40)$$

Note that Eq. (40) represents a discontinuous variation on the P_1 identity, $\phi/2 = (\underline{J}^+ + \underline{J}^-)$. Equation (38) then becomes

$$\begin{aligned} & - \frac{n_i^l}{3\sigma_t} \frac{\partial}{\partial r_i} \delta\phi^{s,(\ell+1)}(\underline{r}) \\ & = \left\{ \frac{\underline{\Theta}^T \delta\phi^{(\ell+1)}}{2} - \frac{n_i^l}{3\sigma_t} \frac{\partial \underline{\Theta}^T}{\partial r_i} \delta\phi^{(\ell+1)} \right\} \\ & - \frac{\underline{\Theta}^T \delta\phi_{cont}^{(\ell+1)}}{2} . \end{aligned} \quad (41)$$

Inserting Eq. (41) into Eq. (37) results in the following local set of equations (P_k unknowns) for element k that are decoupled from all other elements:

$$\begin{aligned} & \sum_{l=1}^{N_{faces}} \int_{\delta V_{k,l}} \underline{\Theta} \left(\frac{\underline{\Theta}^T \delta\phi^{(\ell+1)}}{2} - \frac{\underline{\Theta}^T \delta\phi_{cont}^{(\ell+1)}}{2} \right) d\delta V \\ & - \sum_{l=1}^{N_{faces}} \frac{1}{3\sigma_t} \int_{\delta V_{k,l}} n_i^l \underline{\Theta} \frac{\partial \underline{\Theta}^T}{\partial r_i} \delta\phi^{(\ell+1)} d\delta V \\ & + \frac{1}{3\sigma_t} \int_{V_k} \frac{\partial \underline{\Theta}}{\partial r_i} \frac{\partial \underline{\Theta}^T}{\partial r_i} \delta\phi^{(\ell+1)} dV \\ & + \int_{V_k} \underline{\Theta} \{ \sigma_a \underline{\Theta}^T \delta\phi_{cont}^{(\ell+1)} - \sigma_s \underline{\Theta}^T (\phi^{(\ell+1/2)} - \phi^{(\ell)}) \} dV \\ & = 0 . \end{aligned} \quad (42)$$

For elements with a constant Jacobian (such as tetrahedra),

$$\begin{aligned} & \frac{1}{3\sigma_t} \int_{V_k} \frac{\partial \underline{\Theta}}{\partial r_i} \frac{\partial \underline{\Theta}^T}{\partial r_i} \delta\phi^{(\ell+1)} dV \\ & - \sum_{l=1}^{N_{faces}} \frac{1}{3\sigma_t} \int_{\delta V_{k,l}} n_i^l \underline{\Theta} \frac{\partial \underline{\Theta}^T}{\partial r_i} \delta\phi^{(\ell+1)} d\delta V = 0 , \end{aligned} \quad (43)$$

and Eq. (42) becomes

$$\begin{aligned} & \sum_{l=1}^{N_{faces}} \int_{\delta V_{k,l}} \underline{\Theta} \left(\frac{\underline{\Theta}^T \delta\phi^{(\ell+1)}}{2} - \frac{\underline{\Theta}^T \delta\phi_{cont}^{(\ell+1)}}{2} \right) d\delta V \\ & + \int_{V_k} \underline{\Theta} \{ \sigma_a \underline{\Theta}^T \delta\phi_{cont}^{(\ell+1)} - \sigma_s \underline{\Theta}^T (\phi^{(\ell+1/2)} - \phi^{(\ell)}) \} dV \\ & = 0 . \end{aligned} \quad (44)$$

Although Eq. (43) does not hold for nonorthogonal hexahedral grids (element with nonconstant Jacobians), we assume that it does. Therefore, given $\delta\phi_{cont}^{(\ell+1)}$ (the CFEM scalar flux corrections), a reasonable approximation of $\delta\phi^{(\ell+1)}$ (the DFEM scalar flux corrections) can be easily obtained from Eq. (44).

We performed a Fourier analysis for our DSA method on orthogonal hexahedral meshes and found that for cube meshes the spectral radius approaches a maximum of 0.83 for optically thick elements with a scattering ratio equal to unity and zero for optically thick elements with scattering ratios less than unity. As in the method of Wareing, Larsen, and Adams, the performance of the DSA method degrades as the aspect ratio of the elements become large. Later in the paper we provide numerical results to demonstrate the effectiveness of the DSA method on unstructured grids.

IV. NUMERICAL RESULTS

IV.A. Mesh Anomalies

In this section we show the number of “rings” of mutual dependency for two different hexahedral meshes, as discussed in the previous sections, as well as the number of faces that are both incoming and exiting for a given angle. The first mesh, as shown in Fig. 1, is generated from a cube of $20 \times 20 \times 20$ orthogonal hexahedral elements. The mesh elements are made nonorthogonal by perturbing the element edge lengths using pseudorandom numbers. In particular, each vertex coordinate, x , y , and z (except the outward-directed coordinates of boundary vertices) is given a perturbed coordinate as follows:

$$\tau_p = \tau_u + 0.15\Delta\tau_u(2R_a - 1), \quad \tau = x, y, z, \quad (45)$$

where

τ_p = perturbed coordinate

τ_u = unperturbed coordinate

$\Delta\tau_u$ = unperturbed element edge length along the coordinate τ

R_a = pseudorandom number in (0,1).

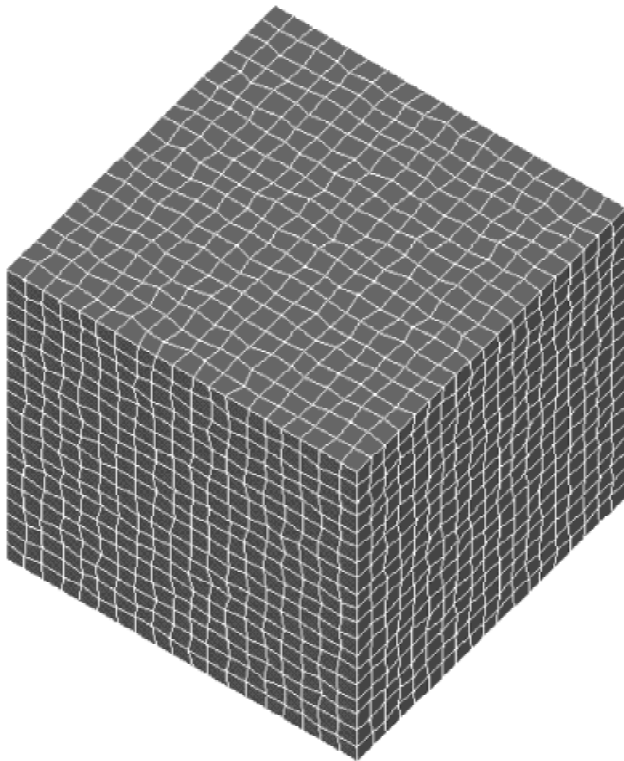


Fig. 1. Cube of random nonorthogonal hexahedral elements.

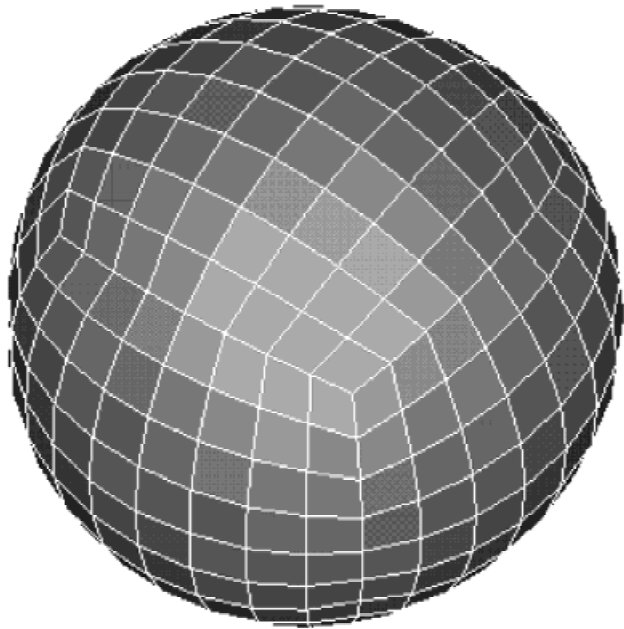


Fig. 2. Sphere of nonorthogonal hexahedral elements.

The average element edge length is approximately the same as that for the orthogonal mesh. This type of mesh is not a smooth mesh and not representative of a “quality” hexahedral mesh.

The second mesh, as shown in Fig. 2, is a sphere comprised of nonorthogonal elements that was generated using ICEM Hexa™, a commercial grid generator. This mesh is smooth and is representative of a “quality” hexahedral mesh.

In Tables I and II we provide for the smooth and nonsmooth meshes, respectively, the number of faces that are both incoming and exiting for a given angle (and a percentage of the total face-angle combinations) for a

TABLE I
Dependency Rings and Reentrant Faces for Cube
of Random Hexahedral Elements

S_N Order (Angles)	Reentrant Faces (Percentage)	Dependency Rings
2 (8)	0 (0.00)	0
4 (24)	4 (0.0002)	0
6 (48)	9 084 (0.30)	0
8 (80)	73 372 (1.45)	36
10 (120)	207 432 (2.70)	316
12 (168)	369 540 (3.44)	952
14 (224)	543 980 (3.79)	1790
16 (288)	721 544 (3.91)	2664

TABLE II

Dependency Rings and Reentrant Faces for Sphere of Hexahedral Elements

S_N Order (Angles)	Reentrant Faces (Percentage)	Dependency Rings
2 (8)	928 (0.22)	0
4 (24)	3 884 (0.30)	0
6 (48)	4 492 (0.18)	0
8 (80)	9 970 (0.23)	0
10 (120)	11 458 (0.18)	0
12 (168)	19 080 (0.21)	0
14 (224)	24 192 (0.20)	0
16 (288)	29 282 (0.19)	4

given S_N order as well as the number of rings of mutual dependency. Here we see that the percentage of the faces that are both incoming and exiting for a given angle and the number of rings of mutual dependency is much larger for the nonsmooth mesh than for the smooth mesh. We note that our algorithm for breaking the rings of mutual dependency is not optimized; therefore, for problems with a large number of these rings, the total computational time can be significantly increased as a result of the cost of finding and breaking the cycles.

We do not provide a similar analysis for tetrahedral meshes because there are no reentrant faces, and our experience indicates that rings of mutual dependency are rare.

IV.B. Accuracy

IV.B.1. Test Problem 1

The first test problem is designed to compare the accuracy of the DFEMs on unstructured tetrahedral meshes and hexahedral meshes. This test problem consists of a one-dimensional homogeneous slab of isotropically scattering material with a total length of 1 cm, a total cross section of 2 cm^{-1} , a scattering ratio of 0.5, a spatially constant isotropic homogeneous source of $1 \text{ particle/cm}^3 \cdot \text{s}$, and vacuum boundaries. Obviously, no analytic solution exists for this problem; therefore, we use a highly refined slab geometry S_{16} solution using Gauss-Legendre quadrature as a reference solution. The problem is modeled in three dimensions with reflecting boundaries on four of the six faces such that the solution only varies along the x axis. Triangular Tchebyshev-Legendre S_{16} quadrature is chosen, so that the Legendre angular cosines (which are equal to those of the one-dimensional Gauss-Legendre cosines) correspond to the x axis. In effect, the three-dimensional solution will equal that of the one-dimensional solution in the absence of spatial errors.

TABLE III

Tetrahedral Mesh Description for Test Problem 1

Mesh ID	Tetrahedra	Average Element Edge Length (mfp)
1	192	0.7070
2	1 536	0.3536
3	12 228	0.1770
4	98 304	0.0884

TABLE IV

Hexahedral Mesh Description for Test Problem 1

Mesh ID	Hexahedra	Average Element Edge Length (mfp)
1	125	0.20
2	1 000	0.10
3	8 000	0.05
4	64 000	0.025

Calculations were performed on a sequence of four tetrahedral grids, four orthogonal hexahedral grids, and four nonorthogonal hexahedral grids. The four tetrahedral meshes, as described in Table III, are unstructured but similar in all respects other than number and size of the tetrahedral elements. The four orthogonal and nonorthogonal hexahedral mesh sequences use 5, 10, 20, and 40 elements per side of the cube and are described in Table IV. The nonorthogonal meshes begin with the orthogonal mesh, but the element edge lengths are perturbed using pseudorandom numbers, as discussed in the previous section.

The absolute value of the relative error in the total absorption rate as a function of average element edge lengths (algebraic average of all edges in the mesh) is plotted in Fig. 3 for the tetrahedral, orthogonal hexahedral, and nonorthogonal hexahedral grid solutions. All of the solutions exhibit third-order accuracy, even for nonorthogonal hexahedral meshes that contain several faces that are both incoming and exiting for a given angle. We note that the nonorthogonal mesh with 20 elements per side is identical to the mesh in the previous section that has a considerable number of dependency rings and faces that are both incoming and exiting for a given angle. We also note that these anomalies had no effect on the iterative solution technique.

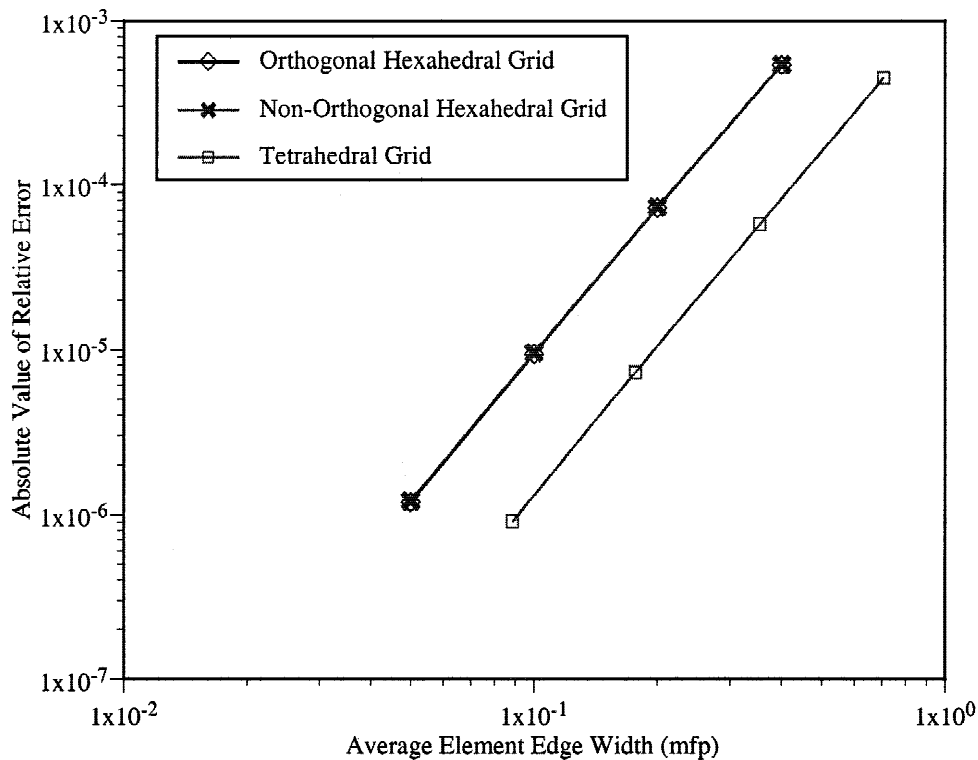


Fig. 3. Absolute error in absorption rate for test problem 1.

IV.C. Acceleration

IV.C.1. Test Problem 2

The second test problem is designed to show the effectiveness of the DSA method with tetrahedral meshes and nonorthogonal hexahedral meshes with and without degenerate elements. A degenerate element is formed when two or more vertices have the same coordinates. The problem consists of a homogeneous sphere with vacuum boundary conditions. We vary the total cross section and the scattering ratio. The tetrahedral mesh consists of 607 tetrahedra with an average element edge length of 0.259 cm. The hexahedral mesh without degeneracies consists of 648 hexahedra with an average element edge length of 0.317 cm. The hexahedral mesh with degeneracies (some of the hexahedra have degenerated into tetrahedra, wedges, and pyramids) consists of 576 hexahedra.

The DSA accelerated spectral radii as a function of total cross section and scattering ratio is given in Tables V, VI, and VII for the tetrahedral mesh, hexahedral mesh without degeneracies, and hexahedral mesh with degeneracies, respectively. The spectral radii for a small total cross section are small because of the large amount of leakage.

Here we see that the DSA method is very effective under most conditions. For the tetrahedral mesh and

TABLE V

Spectral Radius for Test Problem 2
Using Tetrahedral Mesh

σ_t	$c = 1.00$	$c = 0.99$	$c = 0.90$
0.1	0.03	0.03	0.03
1.0	0.17	0.17	0.15
10.0	0.58	0.57	0.49
100.0	0.90	0.87	0.61
1000.0	0.95	0.71	0.25

TABLE VI

Spectral Radius for Test Problem 2 Using Hexahedral
Mesh Without Degenerate Elements

σ_t	$c = 1.00$	$c = 0.99$	$c = 0.90$
0.1	0.03	0.03	0.03
1.0	0.17	0.17	0.15
10.0	0.36	0.36	0.32
100.0	0.68	0.64	0.54
1000.0	0.86	0.72	0.35

TABLE VII

Spectral Radius for Test Problem 2 Using Hexahedral Mesh with Degenerate Elements

σ_t	$c = 1.00$	$c = 0.99$	$c = 0.90$
0.1	0.05	0.05	0.05
1.0	0.27	0.27	0.23
10.0	0.75	0.74	0.63
100.0	1.00	0.99	0.87
1000.0	1.00	0.99	0.71

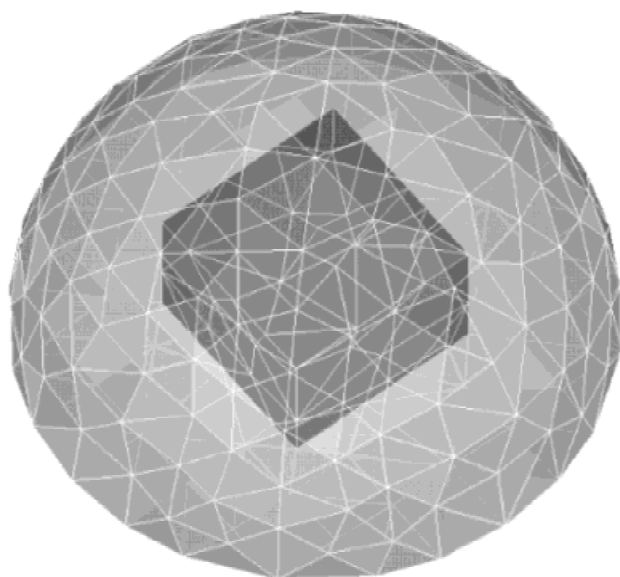


Fig. 4. Tetrahedral mesh for test problem 3.

hexahedral mesh without degeneracies, we see that the spectral radii degrades as $c \rightarrow 1$ and as σ_t becomes large. Clearly this effect is more pronounced for the hexahedral mesh with degeneracies. With increasing amounts of absorption, this degradation in the spectral radii di-

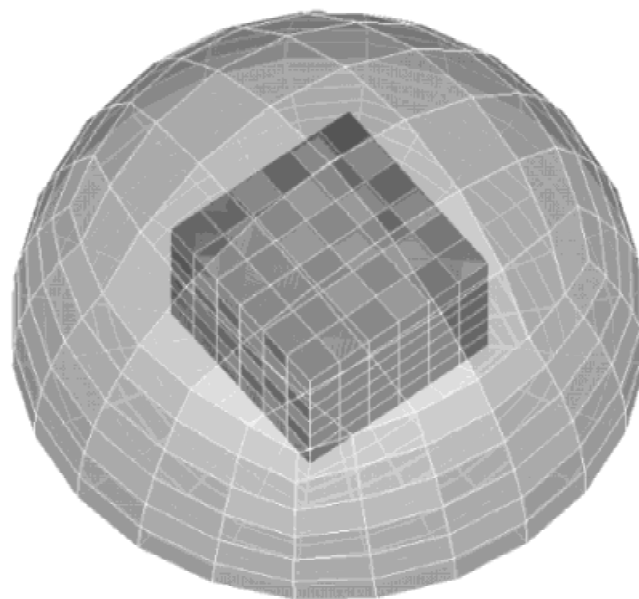


Fig. 5. Hexahedral mesh for test problem 3.

minishes. These results demonstrate that the DSA method should be very effective for most neutronic problems provided the elements are not too skewed, but it may be inadequate for radiative transfer problems.

IV.C.2. Test Problem 3

This problem is designed to show the efficiency of the DSA method on a heterogeneous test problem. The problem is a sphere with a diameter of 2.0 cm containing a 1.0- × 1.0- × 1.0-cm cube in the center. The sphere has a total cross section of 10 cm⁻¹, and the scattering ratio is unity. The total cross section in the box, $\sigma_{t,box}$, is set to 0.01, 1.0, or 10.0 cm⁻¹, and the scattering ratio is 0.9. There is a homogeneous source of strength 1 particle/cm³·s inside the box. The problem is solved on an unstructured tetrahedral mesh and a hexahedral mesh without degeneracies. The tetrahedral mesh contains 1735 elements and is shown in Fig. 4.

TABLE VIII

Tetrahedral Mesh CPU Time and Iteration Counts for Test Problem 3

$\sigma_{t,box}$	Source Iteration		DSA		Absorption Rate (%)
	CPU Time (s)	Iterations	CPU Time (s)	Iterations	
0.1	573.8	157	57.9	15	1.276
1.0	558.0	153	45.8	12	11.78
10.0	496.7	136	35.0	9	62.10

TABLE IX
Hexahedral Mesh CPU Time and Iteration Counts for Test Problem 3

$\sigma_{t,box}$	Source Iteration		DSA		Absorption Rate (%)
	CPU Time (s)	Iterations	CPU Time (s)	Iterations	
0.1	1863.8	157	118.1	9	1.277
1.0	1822.5	153	105.0	8	11.79
10.0	1619.3	137	79.7	6	62.17

The hexahedral mesh contains 2016 elements and is shown in Fig. 5. The problem was solved with S_4 level-symmetric quadrature with a convergence criterion of 10^{-4} .

Tables VIII and IX give the source iteration and DSA CPU time and number of transport iterations for the tet-

rahedral and hexahedral meshes, respectively. The absorption rate percentage is given to verify that the two element mesh types are giving the same answer. Here we see that the DSA method is very effective for this

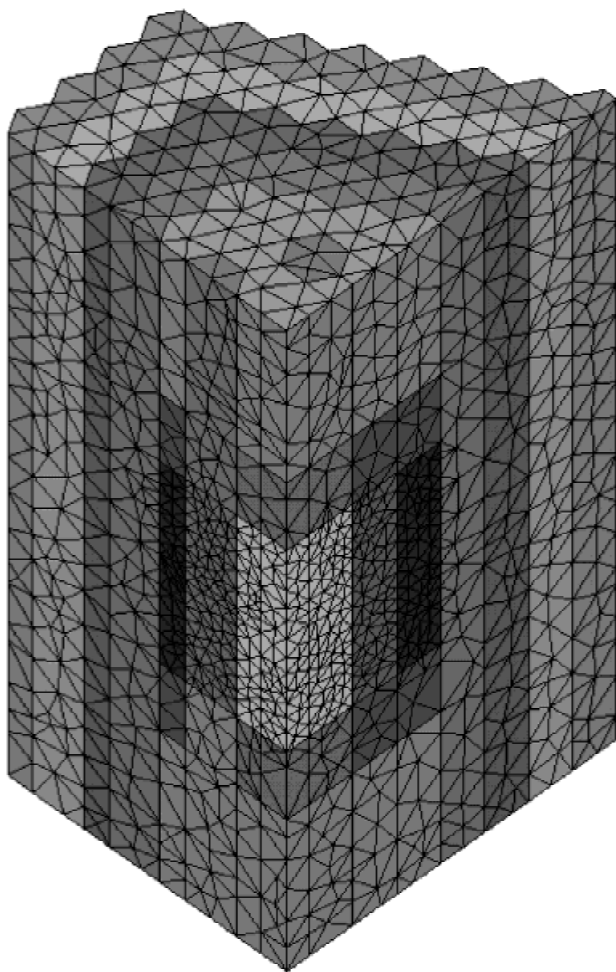


Fig. 6. Tetrahedral mesh for test problem 4.

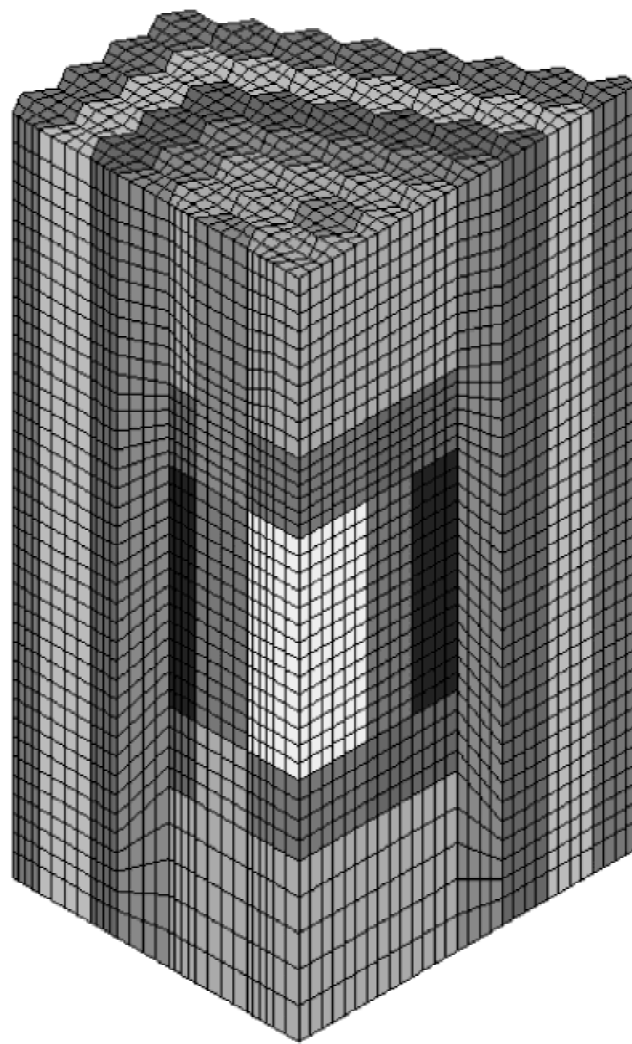


Fig. 7. Hexahedral mesh for test problem 4.

TABLE X
Results for Test Problem 4

Solution	Case 1 k_{eff}	Case 2 k_{eff}
Reference (Monte Carlo)	1.0951 ± 0.0004	0.8799 ± 0.0003
Tetrahedral mesh	1.0948	0.8791
Hexahedral mesh	1.0949	0.8788

problem. The DSA method is also very efficient and only increases the time per iteration by about 6%.

IV.D. Benchmark

IV.D.1. Test Problem 4

This problem is an Organization for Economic Co-operation and Development/Nuclear Energy Agency Committee on Reactor Physics—published benchmark¹⁰ of a small fast breeder reactor (FBR) and used to verify our methods and to compare with other existing methods. Many different Monte Carlo and deterministic solutions are included in Ref. 10. We chose to compare two of the cases specified in the benchmark. Case 1 refers to the controls rods being completely withdrawn, and case 2 refers to the control rods being completely inserted.

We ran this problem using both a tetrahedral and hexahedral mesh using S_4 triangular Tchebyschev-legendre quadrature. The tetrahedral mesh contains 52,092 elements and is shown in Fig. 6. The hexahedral mesh contains 26,364 elements and is shown in Fig. 7. We note that the geometry can be exactly modeled using these unstructured grids. The problems were run both with and without DSA using a relative convergence criterion of 10^{-5} for both k_{eff} and the scalar fluxes. The results are given in Table X. We found substantial speed-ups using DSA. The results for case 1 using the tetrahedral mesh required 2.77 CPU h using DSA and 22.3 CPU h without DSA. The results for case 1 using the hexahedral mesh required 2.27 CPU h using DSA. Due to the length of calculation, the hexahedral mesh was not run without DSA.

The results indicate that the new methods discussed in this paper can be used to accurately model this benchmark problem. We note that our results appear to be much closer to the reference Monte Carlo solutions than the published deterministic solutions (both S_n and P_n) in Ref. 10.

V. CONCLUSIONS

We successfully developed and implemented discontinuous finite element methods for the S_N equations on

three-dimensional unstructured tetrahedral and hexahedral grids. We discussed the difficulties involved in developing these methods and acceptable techniques for dealing with them. We demonstrated that the DFEMs are third-order accurate for integral quantities involving scalar fluxes on both tetrahedral meshes and on nonorthogonal hexahedral meshes even with the presence of faces that are both incoming and exiting for a given angle. The source iterations have successfully been accelerated with a DSA method that is efficient and effective for most types of problems, especially neutronics problems. We are presently investigating other DSA techniques that are effective for all types of problems, especially thermal radiative transfer problems. Finally, we plan to optimize our algorithm for breaking the “rings” of mutual dependency described in previous sections.

ACKNOWLEDGMENT

This work was performed under the auspices of the U.S. Department of Energy.

REFERENCES

1. W. H. REED et al., “TRIPLET: A Two-Dimensional, Multi-group, Triangular-Mesh, Planar Geometry, Explicit Transport Code,” LA-5428-MS, Los Alamos Scientific Laboratory (1973).
2. T. J. SEED et al., “TRIDENT: A Two-Dimensional, Multi-group, Triangular-Mesh, Planar Geometry, Explicit Transport Code,” LA-6735-MS, Los Alamos Scientific Laboratory (1977).
3. M. MORDANT, “Some Efficient Lagrangian Mesh Finite Elements Encoded in ZEPHYR for Two Dimensional Transport Calculations,” *Ann. Nucl. Energy*, **18**, 609 (1981).
4. O. C. ZIENKIEWICZ and R. L. TAYLOR, *The Finite Element Method*, 4th ed., Vol. 1, McGraw-Hill Book Company, New York (1994).
5. A. V. AHO, J. E. HOPCROFT, and J. D. ULLMAN, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Massachusetts (1974).

6. R. E. ALCOUFFE, "Diffusion Synthetic Acceleration Methods for the Diamond-Differenced Discrete-Ordinates Equations," *Nucl. Sci. Eng.*, **64**, 344 (1977).
7. E. W. LARSEN, "Unconditionally Stable Diffusion-Synthetic Acceleration Methods for the Slab Geometry Discrete Ordinates Equations. Part 1: Theory," *Nucl. Sci. Eng.*, **82**, 47 (1982).
8. M. L. ADAMS and W. R. MARTIN, "Diffusion Synthetic Acceleration of Discontinuous Finite Element Transport Iterations," *Nucl. Sci. Eng.*, **111**, 145 (1992).
9. T. A. WAREING, E. W. LARSEN, and M. L. ADAMS, "Diffusion Accelerated Discontinuous Finite Element Schemes for the S_N Equations in Slab and X-Y Geometries," *Proc. Int. Topl. Mtg Advances in Mathematics, Computations, and Reactor Physics*, Pittsburg, Pennsylvania, April 28–May 2, 1991, American Nuclear Society (1991).
10. T. TAKEDA and H. IKEDA, "3-D Neutron Transport Benchmarks," NEACRP-L-330, Organization of Economic Cooperation and Development/National Energy Agency Committee on Reactor Physics (1991).