Nuclear Mathematical and Computational Sciences: A Century in Review, A Century Anew
Gatlinburg, Tennessee, April 6-11, 2003, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2003)

Progress on Adaptive Discrete-Ordinates Algorithms and Strategies

Joseph C. Stone and Marvin L. Adams
Texas A&M University
3133 TAMU, College Station, TX 77843-3133
hapstone@tamu.edu; mladams@tamu.edu

ABSTRACT

We develop strategies for adaptively refining discrete-ordinates angular quadrature sets in particle transport calculations. Our intent is for quadrature sets to adapt locally in position and direction so that they resolve angular variations in the angular flux by efficient placement of quadrature points. We describe our strategies and our initial algorithms for implementing those strategies. We demonstrate the effectiveness of these simple algorithms on test problems that present significant challenges to angular discretizations. We study various results from these test problems in considerable detail. We conclude that our approach is promising and merits continued development.

Key Words: Adaptive, Discrete Ordinates, Angular Discretization, Ray Effect, Local Refinement

1.   INTRODUCTION

Deterministic solutions of the particle transport equation require discretization of each independent variable. The usual independent variables are position, energy, direction, and time. Because a direction is described by angles relative to coordinate axes, we often refer to the direction variable as the *angular* variable. The angular variable is usually discretized either by function expansion (spherical harmonics or finite elements) or by replacing angular integrals with quadrature sums (discrete-ordinates method). The most commonly used angular discretization is the discrete-ordinates (D-O) method.

We address the following one-group fixed-source steady-state transport problem:

$$\mathbf{\Omega}\bullet\nabla\psi(\mathbf{r},\mathbf{\Omega})+\sigma_t(\mathbf{r})\psi(\mathbf{r},\mathbf{\Omega})=q(\mathbf{r},\mathbf{\Omega})+\int_{4\pi}d\Omega'\sigma_s(\mathbf{r},\mathbf{\Omega}'\rightarrow\mathbf{\Omega})\psi(\mathbf{r},\mathbf{\Omega}'). \tag{1}$$

The transport equation is notoriously difficult to discretize accurately, largely because the solution can be discontinuous in every variable. At any given spatial position **r**, for example, the transport solution ψ can be discontinuous at an arbitrary number of arbitrary locations in the direction domain. Even if the solution is continuous in angle it is often devoid of smoothness. This makes the direction variable extremely difficult to discretize accurately.

We illustrate this behavior with a variant of a classical “ray-effect” problem, modified to make it relatively easy to solve analytically. This problem illustrates properties of the transport solution that make angular discretization challenging. It also offers some motivation for the approach that we describe here, which we hope will improve the state of the art in D-O calculations.

Figure 1a depicts a two-dimensional problem that contains a circular isotropic source centered at the origin in a non-scattering material of uniform composition. Figure 1.b is a plot of the angular flux at four spatial points, as a function of the azimuthal direction (angle) variable, which we call γ. (We have chosen a fixed polar angle, θ, of 30 degrees from the *z* axis. γ is the angle between the *y* axis and the direction of particle motion, with clockwise arbitrarily chosen to be positive.)

Figure 1 illustrates several important points about "ray effects" – unphysical oscillations in the *scalar flux* – which have plagued the D-O method since its creation decades ago. The *scalar flux* and its D-O approximation are defined as follows.

$$\phi(\mathbf{r},E) \equiv \int_{4\pi} \psi(\mathbf{r},E,\mathbf{\Omega})\,d\Omega \xrightarrow[D-O]{} \sum_{m=1}^{M} w_m \psi(\mathbf{r},E,\mathbf{\Omega}_m)\,. \qquad (2)$$

Here the $\{w_m\}$ and $\{\mathbf{\Omega}_m\}$ are the quadrature *weights* and *points*, respectively.

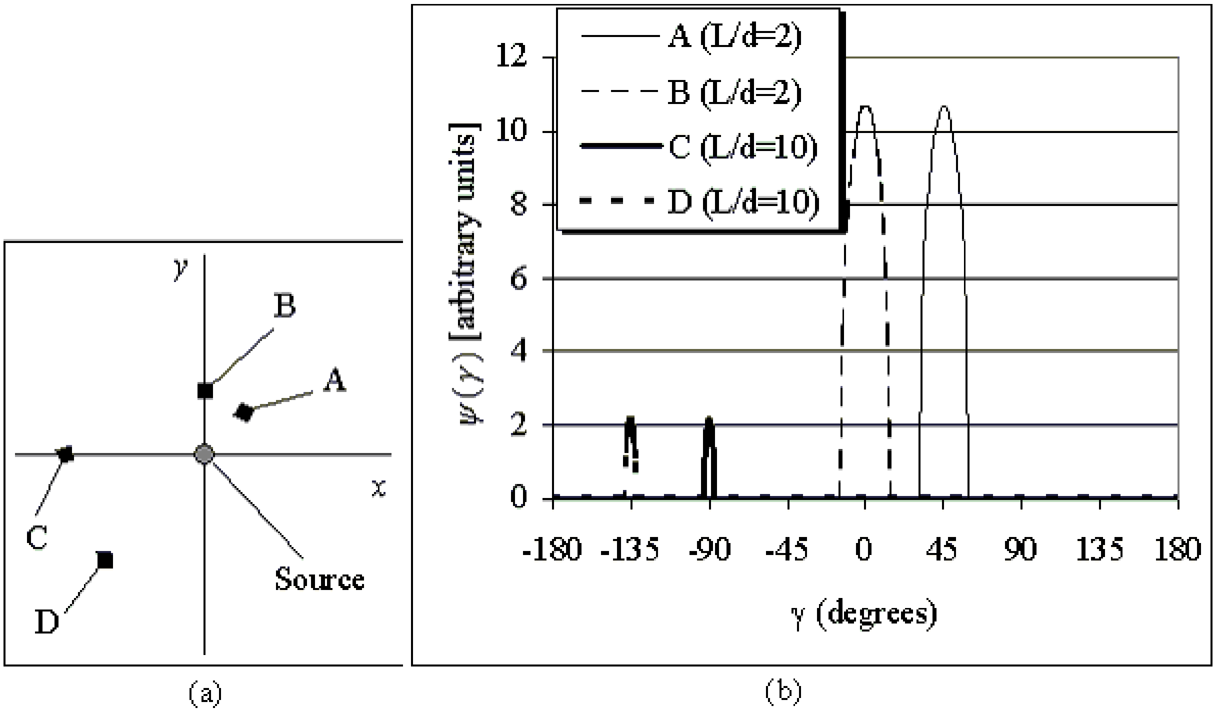


Figure 1. "Ray-effect" example. (a) Problem geometry (not to scale). (b) Solution as a function of the azimuthal direction variable, at four different spatial points. *L* is distance from origin; *d* is diameter of source.

The exact scalar flux in our example problem is the same at points A and B, for A & B are equidistant from the source and thus their curves bound the same area in Fig. 1.b. The same statement holds for points C and D, and in fact for all points on any circle centered on the source.

Consider now a very crude quadrature set: the S_N level-symmetric quadrature set with N=2. (N is the number of unique direction cosines along each coordinate axis in a level-symmetric set. The number of distinct directions, M, is $N(N+2)$ in three dimensional Cartesian problems and half that in two dimensions.) The S_2 quadrature set has 4 directions in our 2D problem; the associated γ angles are –135, –45, 45, and 135 degrees. The quadrature weights, w_m, all have the same value, which is $4\pi/4 = \pi$. Consider now the S_2 approximations of the scalar fluxes at points A-D in Fig. 1. These will be *zero* for points B and C, because at these spatial points the angular flux, ψ, is zero in all four quadrature directions. Conversely, the D-O approximation of the scalar fluxes will be much too large for points A and D, because the narrow peaks in the associated ψ functions are along quadrature directions.

It follows that if we plot the S_2 scalar flux along a circle centered on the source, the plot will oscillate between zero and a value that is much larger than that of the correct scalar flux, whereas the correct scalar flux is constant along such a circle. This is an example of *ray effects* – unphysical oscillations in the scalar flux caused by the finite D-O quadrature set.

It is not difficult to see that any fixed finite quadrature set will suffer from ray effects if the spatial points of interest are far enough from the source. (Compare the width of peaks A & B in Fig. 1.b to the width of peaks C & D. If the peak width is narrower than or comparable to the distance separating quadrature points, there will be pronounced ray effects in this problem.) The ray effects get worse as one looks farther and farther from the source. To summarize, ray effects occur when the D-O quadrature set integrates two functions differently even though one is just a translation (in angle) of the other. Ray effects are most pronounced when the *angular* flux is a “peaky” function *of angle*.

However, suppose that the quadrature set were allowed to conform to the solution at any spatial point of interest. In our simple example problem the nonzero portion of the solution at each spatial point is not a difficult function to integrate over γ; in fact, a three-point Gaussian quadrature integrates this portion with error < 1%, provided the points are properly placed. Obviously, a one-point quadrature exactly integrates each zero portion. Thus, there exists a quadrature set at each spatial point that needs only five azimuthal points to obtain the azimuthal integral to within 1%. Furthermore, the quadrature set at every point that is the same distance (L) from the source is a simple rotation (translation in γ) of the set at any other such point. The solution is continuous and reasonably smooth in the polar angle, θ; thus, the polar integral can be evaluated relatively accurately with a moderate number of Gaussian quadrature points. We conclude that at any spatial point in the example problem, there exists a *product quadrature* set with a reasonably small number of points that would approximate the needed angular integrals. A *product quadrature* set combines two one-dimensional quadrature sets, one in the azimuthal angle and one in the polar angle. A product-quadrature integral can be written as follows.

$$\int_{4\pi} \psi(\mathbf{r},\mathbf{\Omega})\,d\Omega = \int_0^{\pi}\int_0^{2\pi} \psi(\mathbf{r},\gamma,\theta)\,d\gamma \sin\theta d\theta \xrightarrow[\text{quadrature}]{\text{product}} \sum_{m=1}^{M_p} w_{p,m} \sum_{n=1}^{M_a} w_{a,n}\psi(\mathbf{r},\gamma_n,\theta_m)\,. \qquad (3)$$

The sets $\{w_{p,m}\}$ and $\{\theta_m\}$ are the polar quadrature weights and angles; $\{w_{a,n}\}$ and $\{\gamma_n\}$ are the azimuthal quadrature weights and angles.

In this paper we explore strategies and algorithms for adapting the azimuthal quadrature set $\left\{w_{a,n}, \gamma_n\right\}$ so that it can efficiently and accurately integrate the local angular flux. We allow the quadrature set to vary with spatial position and to have non-uniform refinement along the γaxis. That is, we are focusing on a true locally adaptive capability. User input is a starting quadrature set and the choice of refinement strategies and tolerances. We restrict ourselves to two-dimensional Cartesian coordinate systems with rectangular spatial cells. The extension to three dimensions with brick-shaped cells is obvious. The extension to curvilinear coordinates is less obvious but, we believe, relatively straightforward. The extension to unstructured grids will require further research, which will be justified if we succeed in the simpler setting of structured grids. Our intent in this paper is to demonstrate proof-of-principle, not to present a finished product.

Previous researchers have proposed, implemented, and tested quadrature sets designed to overcome ray effects and other quadrature errors in D-O calculations. Longoni and Haghighat have developed “ordinate splitting” techniques that offer substantially improved accuracy for lower computational cost than standard D-O quadrature sets, especially for source-detector type problems [1,2]. This technique allows the user to choose directions around which to place a high concentration of quadrature points. The chosen quadrature set is used for all spatial locations for the duration of the problem. The DORT code has long offered users the option of choosing different quadrature sets in different spatial regions [3]. This can produce large efficiency gains compared to the use of a single quadrature set that is sufficiently accurate for all spatial regions. However, the burden is on the user to define the quadrature sets – the sets do not adapt to the solution as the calculation proceeds. Smedley-Stevenson has explored simple adaptation whereby one quadrature set is applied to the entire spatial domain, and refinement occurs by replacing an entire quadrant or octant of the current quadrature set with a quadrant or octant of a higher-order set [4]. Our work is similar in that it allows the quadrature set to adapt to the solution; however, we allow different quadrature sets in different spatial regions. Furthermore, we focus on product quadrature sets for 2D problems, which we believe offer substantial advantages over level-symmetric sets.

In Section 2 we discuss the algorithms and strategies that we have devised to date for this problem. In Section 3 we describe details of our implementation. In Section 4 we discuss issues that we have yet to address and how we plan to address them. Section 5 contains numerical results from simple test problems, and Section 6 provides concluding remarks and our ideas for the next steps in this research.

2. ALGORITHMS AND STRATEGIES

The main goal of our adaptive D-O effort is to obtain a given accuracy for much lower computational cost, and with less user burden, than can be obtained with fixed quadrature sets. In an adaptive code there is a tradeoff between the reduced number of unknowns needed to produce a given accuracy and the overhead of the adaptive data structures and error tests. We keep this tension in mind when we develop our strategies and design our algorithms.

2.1 Basic Strategies

We begin by dividing the spatial domain into *quadrature regions*. (At this stage in our effort these regions are defined by the user; later we hope to develop algorithms for adaptively selecting these regions to optimize computational efficiency.) Our strategy is to have a single quadrature set for all cells in a given region but to allow different quadrature sets in different regions. In other words, the quadrature set is a piecewise-constant function of position. In a given region the quadrature set should adapt so that it resolves the angular variation of the angular flux in every cell in the region.

We have developed our schemes under the assumption that the problem is solved by iterating on the scattering source, with the mesh being swept in the direction of particle travel during each iteration. In most problems it would be inefficient to invoke the angular adaptation logic during every iteration; it would also be inefficient in most problems to completely converge the scattering source between angular adaptations. We are developing algorithms to optimize the frequency of angular adaptation; we will discuss this in more detail later.

Suppose that it is time to adapt. Our algorithm for making refinement and coarsening choices is best illustrated by example. Figure 2 shows an example spatial domain with 20×20 = 400 spatial cells and 4×4 = 16 quadrature regions of varying sizes. We separately consider each *quadrant* of the direction domain. Quadrant I corresponds to up-and-right directions, quadrant II is up-left, quadrant III is down-left, and quadrant IV is down-right. Note that each quadrant has a different sweep ordering – the order in which cells must be calculated so that incident fluxes for each cell are the most up-to-date exiting fluxes from upstream cells.

First consider quadrant I (directions up and right). In this quadrant the incident fluxes on the bottom and left edges of the problem are known from boundary conditions; thus, the sweep begins in the lower-left region of the spatial domain. The first step of an adapting sweep in a given region is to ensure that the quadrature set adequately resolves the incident flux's variation in angle, on every incoming cell edge in the region. Our strategy for ensuring adequate resolution is to compare the angular flux evaluated at a set of "test directions" against the angular flux obtained by interpolation from neighboring directions. If the evaluated and interpolated values are within the desired tolerance, the test direction is not needed in the quadrature set; otherwise it is added. The same idea can be applied to coarsening: a direction that is in the quadrature set can be tested the same way, comparing its angular flux against a value interpolated from neighbors. If the two values are within the desired tolerance *everywhere* in the spatial region, then the point can be removed from the region's quadrature set.

Once a region's quadrature set has been adapted to resolve the incident flux on all incident edges, in the quadrant of directions being considered, the sweep proceeds across the region for all the directions in that quadrant. The adaptation logic is then applied to the angular flux on the *exiting* edges of the region. That is, the test/evaluate procedure is used to ensure that the quadrature set adequately resolves the exiting angular fluxes, much like it was used for incident fluxes. In this case coarsening is not appropriate, for if a point is in the quadrature set it must have been needed to resolve an incoming flux; however, refinement could be needed. To evaluate an angular flux in a "test direction" at this stage requires that the region be swept in that direction. This involves returning to the incident edges, interpolating to obtain the incident fluxes (which upon convergence are virtually guaranteed to be within the tolerance of the correct flux), and then sweeping through the region's cells. Again, if the test and evaluated exiting fluxes are not close enough, on any exiting cell edge, the test direction is added to the quadrature set for the region.

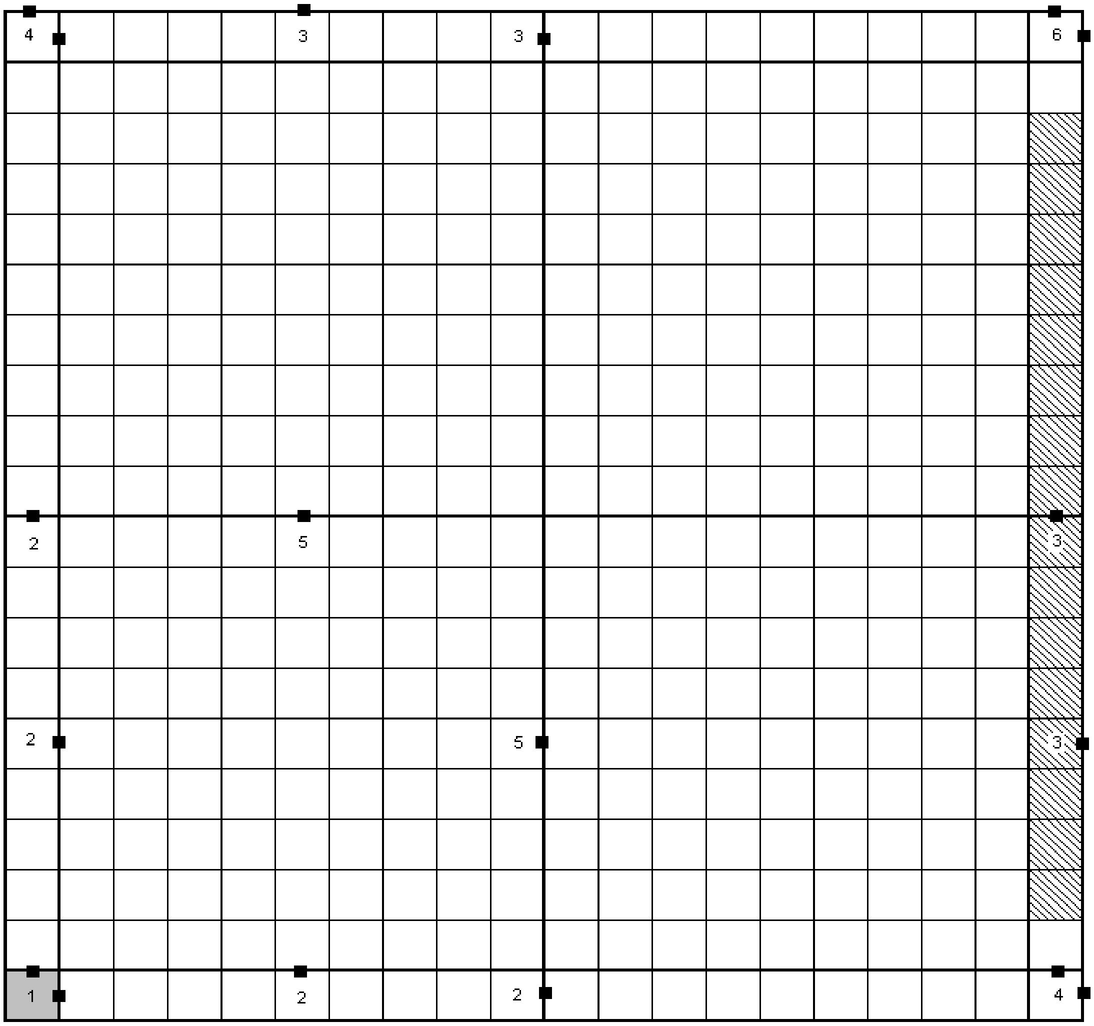


Figure 2. Spatial domain, cells, and quadrature regions in an example problem.

This procedure is repeated region by region and quadrant by quadrant. After marching through each region for each quadrant of directions, we are guaranteed that on every cell edge of every region, each test direction was added to the quadrature set if its angular flux differed enough from an interpolated value. We are not necessarily guaranteed that the angular flux is fully resolved, because our set of test directions was finite.

Our strategy for choosing test directions is to refine by no more than one "level" on a given adapting sweep. Consider a quadrant that contains *N* azimuthal directions in its initial quadrature set. These directions divide the azimuthal range into *N*+1 intervals. We could, for example, place one test direction in the center of each such interval. Alternatively, we could define *N* intervals, each of which spans an existing point, and place one test direction on each side of each existing point, for a total of 2*N* test points. There are many other possibilities as well. Regardless of the details of how we refine by one "level," we leave further refinement for later iterations.

2.2 Detailed Strategies

The preceding subsection defines our basic strategies but ignores some important details. In this subsection we discuss strategies for addressing these details, including interpolation, criteria for determining whether test and evaluated fluxes are "close enough," and the important and potentially difficult issue of how to determine the weights in a refined or coarsened quadrature set.

When we interpolate we assign a functional form to a function whose values are known only at discrete points. Interpolating functions range from low-order (linear between neighboring points) to high-order (high-order polynomials that go through many points, for example). High-order interpolation works well for smooth functions but not otherwise. Low-order interpolation is safer for unsmooth functions but is less accurate for smooth functions. Our ultimate goal is an algorithm that chooses the best kind of interpolation for a given portion of the angular domain in a given spatial region; that is, the interpolating function should also adapt to the local solution. Our first implementation, however, is to simply use linear interpolation.

The criterion for determining whether test and evaluation values are "close enough" could take many forms. One very strict possibility is a point-wise relative criterion:

$$\left|\psi_{eval} - \psi_{interp}\right| < \varepsilon_{test} \left|\psi_{eval}\right| \ . \tag{4}$$

One disadvantage of this criterion is that it can cause a lot of points to be added in portions of the angular domain where the angular flux is very small and unimportant. One way to prevent this, and in fact to save a lot of overhead, is to eliminate testing and evaluation in angular intervals whose angular flux is below a threshold value. For example:

If ($\left|\psi_{interp}\right| < \varepsilon_{threshold} \left|\bar{\psi}_{quad}\right|$) do not test for refinement,

where $\bar{\psi}_{quad}$ is the average angular flux in the given quadrant at the given point. Another option is to use the following criterion for "close enough:"

$$\left|\psi_{eval} - \psi_{interp}\right| < \varepsilon_{test} \left|\bar{\psi}_{quad}\right| \ . \tag{5}$$

Another possibility is a combination of criteria (4) and (5):

$$\left|\psi_{eval} - \psi_{interp}\right| < \varepsilon_{test} \left|\psi_{eval}\right| + \varepsilon_{threshold} \left|\bar{\psi}_{quad}\right| \ . \tag{6}$$

One disadvantage of these effort-reducing strategies is that they could under-resolve the angular flux where it has a small value even though those small values could be very important in a downstream region of the spatial domain. In our implementation that we report here, we accept that risk and employ the test shown in Eq. (6).

We mentioned above that in our present stage of development we are using simple linear interpolation for our test points. This is consistent with a trapezoidal-rule quadrature set, which is what we are employing in for the azimuthal integral. We suspect that it is important to maintain consistency between the quadrature set and the interpolation function; for example, if the quadrature set used Simpson's rule, then it would make sense to use piecewise quadratic interpolation. We discuss this issue further in Section 4. For now we note that our choice of the trapezoidal rule makes it easy for us to revise quadrature weights when we refine or coarsen in a given azimuthal integral: a point's weight is always half the sum of the interval widths between the point and left and right neighbors.

We begin analyzing a similar problem to the one shown in Figure 1a. We focus on the upper right quadrant of the figure and assume symmetric conditions in the rest. The problem is divided into 4 by 4 regions, with an infinite linear source in the lower left region. Each region is then subdivided into individual square cells of equal size (throughout the problem) as illustrated in Figure 2. The numbers represent points of interest where angular flux is graphed and the dashed cells are used to compare scalar flux when using different quadrature sets.

3. IMPLEMENTATION

In this section we briefly describe our implementation of our adaptive algorithms in a Fortran-90 transport code that we have written for the purpose of testing these algorithms. Our code solves two-dimensional Cartesian-coordinate problems using rectangular spatial cells.

3.1. Spatial Discretization

We have chosen the step-characteristic (SC) spatial discretization scheme for its simplicity, monotonicity, and positivity. SC has the drawbacks of not being extremely accurate and being somewhat diffusive, but since our focus is on *angular* discretization SC has been adequate to date for our purposes. The SC scheme can be defined as follows.

- The scattering and fixed sources are approximated as constant in each cell, equal to their cell-average values.
- The incident angular flux on each cell edge is approximated as constant, equal to its average value over the edge.
- The angular flux, ψ, is computed analytically in each cell given the approximations described above.
- This analytic flux is averaged over the cell; that average is used in quadrature sums to compute the scalar flux for the scattering source (and to compute other angular moments if scattering is anisotropic).
- The analytic flux is also averaged over each outgoing edge, which produces the incident flux for the downstream cells.

3.2. Implementation of Adaptive Algorithms

In Section 2 we outlined our strategies for adaptive D-O calculations and our present-stage algorithms for implementing those strategies. Here we briefly describe how we have implemented our algorithms in our Fortran-90 code.

Because the interesting and challenging problem in 2D is the variation of the angular flux in the azimuthal direction, this is what we focus on. At this stage we are holding the polar quadrature set fixed as set by the user. We require data structures to which we can add and subtract angular flux values as we refine and coarsen our quadrature sets. Our current implementation uses *linked lists* to store azimuthal directions and weights as well as angular fluxes on region surfaces. A *linked list* employs a derived type statement and pointer declarations to create connections from one data storage allotment to another. We store elements of each list in order of increasing azimuthal angle, which makes it easy to find nearest neighbors for interpolation.

After all angular data structures are initialized, based on user specification of quadrature regions and initial quadrature sets, the code basically functions as follows.

```
For each adaptation step:
    For each scattering iteration (until convergence or until the max specified for this step):
        For each of the four quadrants:
            For each spatial region (in order of particle flow):
                For each direction in this region’s quadrature set in the current quadrant:
                    For each cell in the region (in order of particle flow):
                        Solve for average and exiting angular fluxes
                    End cell loop
                End direction loop
                If first scattering iteration of this adaptation step, then:
                    Create list of test directions
                    For each test direction:
                        For each incident cell edge of the current region:
                            If evaluated psi is not close enough to interpolated, then:
                                make sure test direction is on the “keep” list
                            End if block
                        End incident-edge loop
                        For each cell in the region (in order of particle flow):
                            Solve for average and exiting angular fluxes
                        End cell loop
                        For each exiting cell edge of the current region:
                            If evaluated psi is not close enough to interpolated, then:
                                make sure test direction is on the “keep” list
                            End if block
                        End exit-edge loop
                    End test-direction loop
                    Add “keeper” directions to linked lists for this quadrant, this region
                End if block
            End spatial-region loop
        End quadrant loop
        Test scattering source for convergence.
        If scattering source is converged, exit from scattering-iteration loop
    End scattering-iteration loop
    Test whether adaptation changed any quadrature sets.
    If no changes, exit from the adaptation-step loop
End adaptation-step loop
```

At the present stage of our effort we are not very concerned with efficiency, either of memory or computing time. We are interested first in addressing the question of whether our strategies can produce significantly increased accuracy with significantly fewer unknowns. Our present code allows us to readily explore this question. If the answer is affirmative – and we believe our numerical results in this paper strongly suggest that it is – then we can move on to the next set of questions, including the question of computational efficiency. This could cause us to seek more efficient implementations.

4. ISSUES REMAINING TO BE ADDRESSED

The chief issue that we have not addressed in this paper is the exact integration of spherical-harmonics functions. Our trapezoidal-rule quadrature set does not perfectly integrate very many of these functions. In many problems of practical interest, and in particular in those with highly forward-peaked scattering cross sections, it is well-known that it can be important to accurately integrate as many spherical-harmonics functions as possible [5,6]. Researchers have devoted considerable effort to designing quadrature sets that exactly integrate large numbers of these functions [7,8]. Our current thoughts on this are as follows. A quadrature set that integrates many spherical-harmonics functions can give rise to severe ray effects and other inaccuracies, which simple adaptive quadrature sets can cure. Thus, the ability to integrate these functions does not guarantee an accurate solution, nor does the inability to integrate them perfectly indicate an inaccurate solution. We do believe that it is important to exactly (or at least very accurately) integrate all spherical-harmonics functions that appear in the scattering term. However, in a problem with isotropic or low-order anisotropic scattering, we suspect that exact integration of high-order harmonics is not necessary. What matters most, we believe, is how accurately the quadrature sets perform the integrals that generate the spherical-harmonics moments that appear in the scattering source. Our immediate goal is to expand our library of algorithms to include a range from piecewise low-order to piecewise high-order. It is likely that the lowest-order algorithm we will recommend will be one that exactly integrates spherical-harmonics functions through second order. This entire issue is a subject for considerable further study, which we plan to undertake.

Another issue that we have not fully addressed is how best to choose the sizes and shapes of the "quadrature regions" (spatial regions that have a common quadrature set). In problems with strong localized sources – the kind that give rise to classical ray effects – the best shape for such regions would be wedges emanating outward from the sources. One cannot form such shapes from rectangular cells, and if we try to approximate them we produce regions with re-entrant boundaries, which are not permitted by our current algorithm. One area of future research is quadrature regions of arbitrary shape.

A simple improvement that we expect to implement soon is region-dependent refinement criteria. Imagine a problem in which we care mainly about the solution in a small part of the spatial domain (a detector, a piece of sensitive equipment, a person, etc.). There are potentially large gains in efficiency to be achieved by tightly refining along the primary path of the radiation flow from the source to the important region but refining less tightly in other portions of the problem.

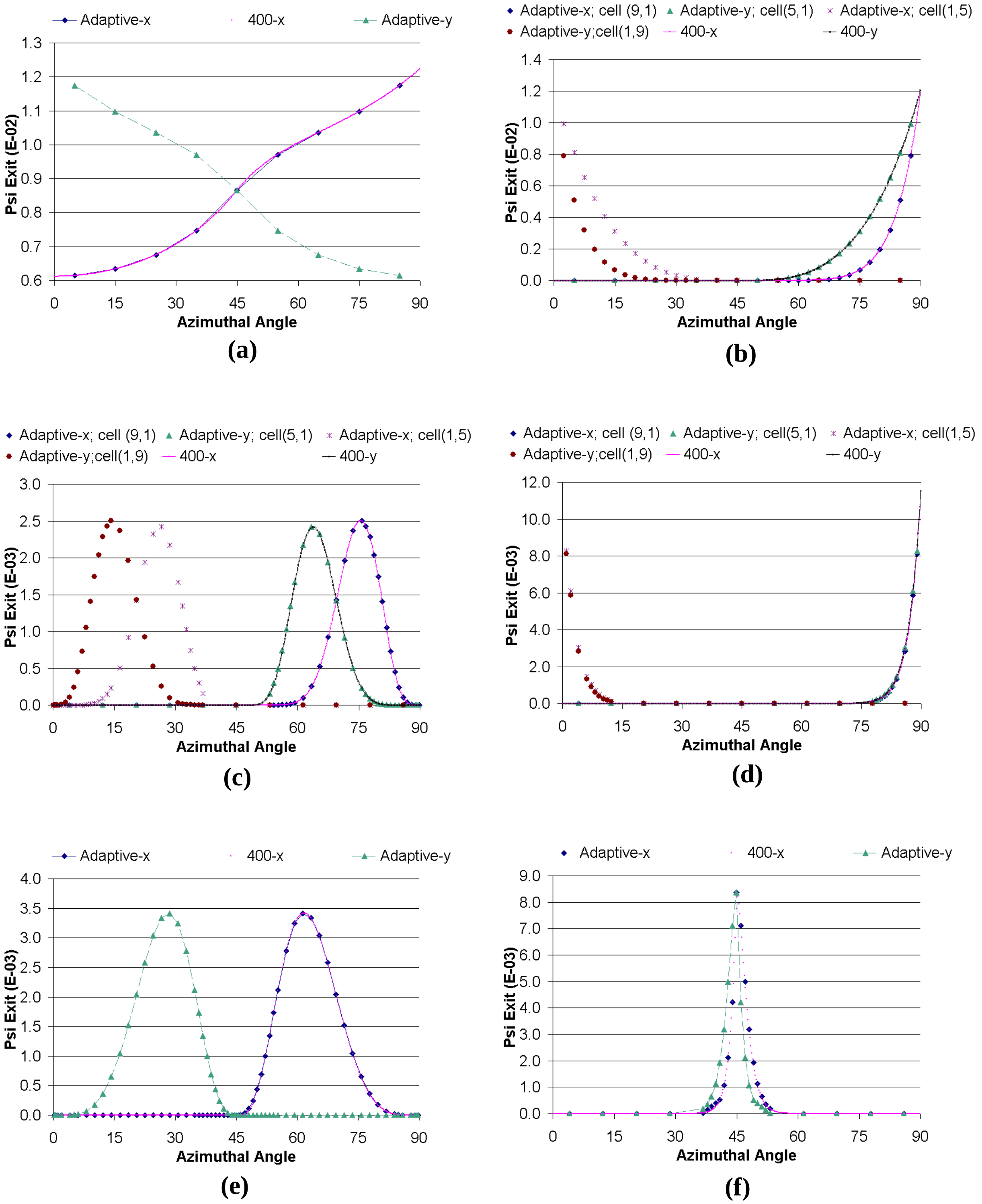
A brief discussion of spatial discretization is in order. A spatial discretization that artificially spreads particle beams is space makes the angular flux artificially smooth in angle, and thus makes it easier for a quadrature set to integrate. This is why calculations made with diffusive spatial discretizations show fewer ray effects than those made with more accurate spatial discretizations, given the same quadrature sets. The step-characteristics spatial discretization used in this paper produces much less of this numerical diffusion than, say, the step-differencing scheme, but significantly more than a higher-order method such as linear characteristics, linear discontinuous, or linear nodal. As we move to more accurate spatial discretizations, our angular-fluxes will lose the artificial smoothness caused by spatial diffusion, and the advantages of our adaptive quadrature sets should become even more pronounced.

5. RESULTS

Because we are focused on the azimuthal refinement problem we have chosen to use only one polar angle in our test problems. We choose one problem with many interesting features and study it very carefully. The problem's spatial domain is shown in Figure 2. It has a 20 x 20 array of square spatial cells, each 0.01 mean-free paths thick, with an isotropic source in the lower left cell. We define a 4 x 4 array of rectangular quadrature regions, as shown by the thicker lines in Figure 2. The central four regions are quite large, which means that their quadrature sets must resolve a wide variety of angular-flux functions. Our problem has no scattering, because scattering smoothes out the angular flux, makes it easier to integrate, and thus reduces ray effects; we prefer a clean, difficult problem for our studies here. Our reference solutions are generated with a quadrature set that has 400 uniformly spaced azimuthal points in each quadrant (1600 total quadrature points).

5.1. Verification of Sanity and Symmetry

We begin with a verification test of our refinement algorithm, which shows that the code refines region by region and quadrant by quadrant just as a careful examination of reference angular fluxes indicates that it should. We use an initial quadrature set with 9 directions per quadrant and set the refinement tolerances to 5% and 0.25% [i.e., ε_{test} = 0.05 and $\varepsilon_{threshold}$ = 0.0025 in Eq. (6)]. In Figures 3 we show plots of angular flux as a function of azimuthal angle in quadrant I, at several different spatial locations. Spatial locations are always cell edges. They are referred to by cell (*i*,*j*) index followed by "x-exit" for the right edge or "y-exit" for the top edge. In Figure 3a we show two sets of three plots; each set should be a mirror image of the other across the 45-degree line because of the symmetry of the problem. We see that this is the case except for one point that was added at 62.5 degrees but not at 27.5. We have determined that this was a case of roundoff error causing the criterion to be barely met in one case but barely missed in the other. Two of the plots in Figure 3a show the reference solution. We examine the adapted quadrature points plotted on top of the solutions and verify that points are indeed added where linear interpolation would miss the correct value by more than the specified tolerance. Figures 3b-f lead to similar conclusions, and the quadrature set is quite obviously different in different regions. (The large number of points shown near 45 degrees in Fig. 3e is needed to resolve the angular flux at other spatial locations in this large region.) Fig. 3f is especially interesting, as it shows no refinement outside of the narrow angular-flux peaks. Recall the ray-effect problem that we discussed in the introduction, and note that Fig. 3f is exactly what we would want to see to efficiently eliminate ray effects.



(a)

(b)

(c)

(d)

(e)

(f)

Figures 3. Sanity and symmetry tests.

5.2. Comparison of Angular Resolution of Different Quadrature Sets

We now examine the resolution of the angular flux at three spatial locations, for different uniform quadrature sets, our adaptive set, and the reference 1600-angle set. Figures 4-6 show the plots for this study. In Fig. 4 we see that 10 uniformly distributed points grossly miss the shape of the angular flux in the upper right corner of the problem, 20 is better but still very poor, and even 60 misses the peak. The adaptive set with approximately 25 points is clearly superior and is almost perfect compared to the reference curve.

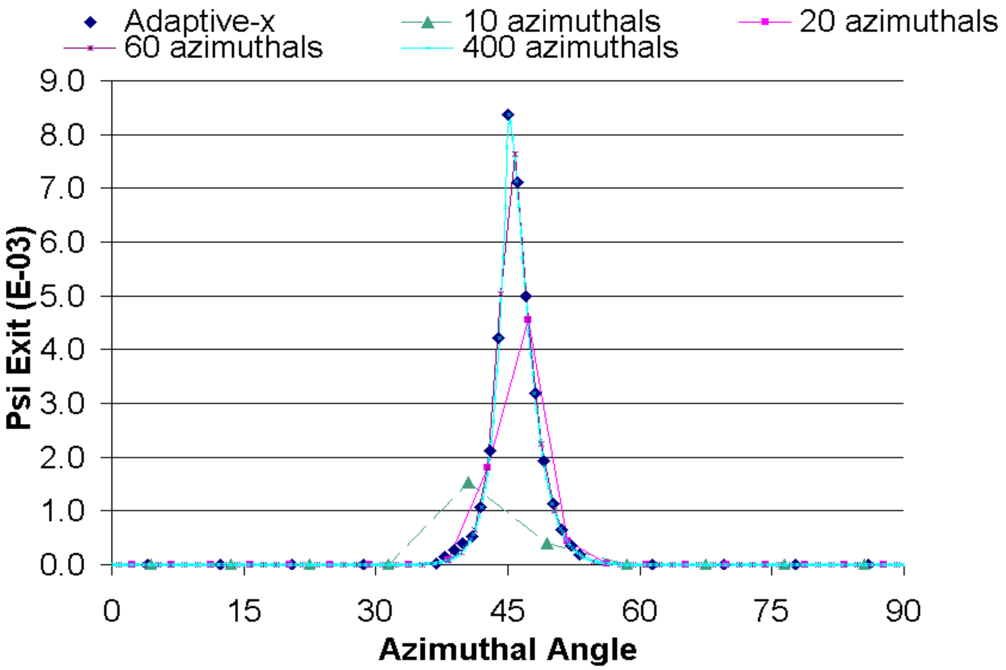


Figure 4. Comparison of angular resolutions on right edge of cell (20,20).

Similar comments hold for Figs. 5 and 6. Even though the angular flux is less peaked at these spatial locations, the adaptive quadrature set is still clearly superior in that it puts points only where they are needed and as a result accurately resolves the angular flux without many unknowns.

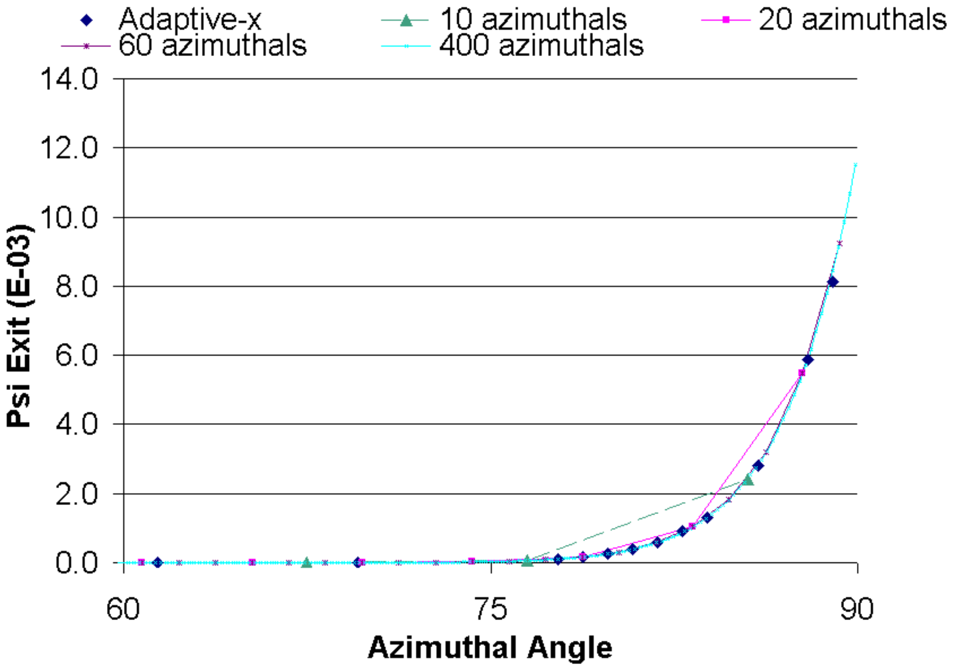


Figure 5. Comparison of angular resolutions on right edge of cell (20,1).

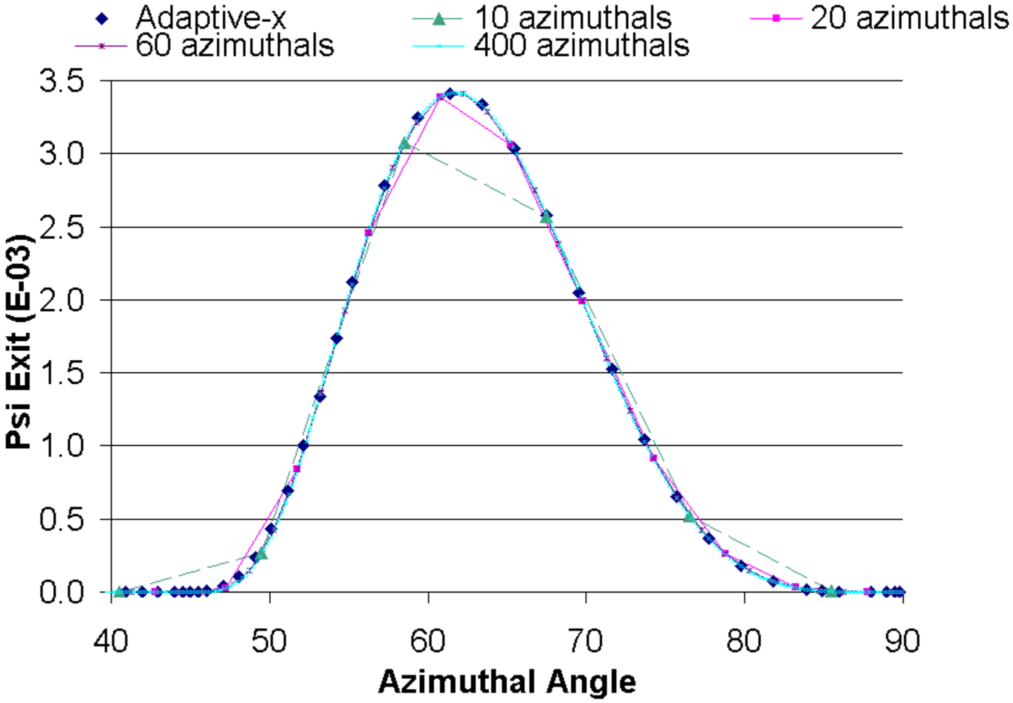


Figure 6. Comparison of angular resolutions on right edge of cell (10,6).

5.3. Mitigation and Elimination of Ray Effects

We turn now to a classical view of ray effects – unphysical *spatial* oscillations in the *scalar* flux – and demonstrate that our adaptive quadrature sets eliminate them in this test problem. We begin by plotting the scalar flux as a function of *y* at the center of the rightmost column of cells. That is, we plot $\phi(x,y)$ at $x = 19.5$ and $y \in (2.5,17.5)$. (The *y* axis in our plots uses cell number instead of *y* coordinate, so it goes from 3 to 18.)

We are using the total number of angular-flux unknowns in the problem as a measure to allow comparison between different kinds of quadrature sets. For example, our reference calculation has 400 cells x 1600 quadrature directions, or 640,000 total cell-averaged angular-flux unknowns. For our adaptive quadrature sets, we similarly compute the number of unknowns region by region and then sum over regions.

Figure 7 displays solutions from relatively crude quadrature sets (those with fewer than 50,000 unknowns) along with the reference solution. The quadrature sets with 2 – 8 azimuthal directions per quadrant exhibit the familiar unphysical oscillations caused by ray effects. Even the quadrature set with 16 azimuthal directions per quadrant, which is similar in resolution to an S_{32} level-symmetric set, displays a noticeable error in cells 17 and 18. In contrast, an adaptive solution with only 20% more unknowns lies essentially right on top of the reference solution. This solution was created by setting $\varepsilon_{test} = 0.05$ and $\varepsilon_{threshold} = 0.0025$ in Eq. (6).

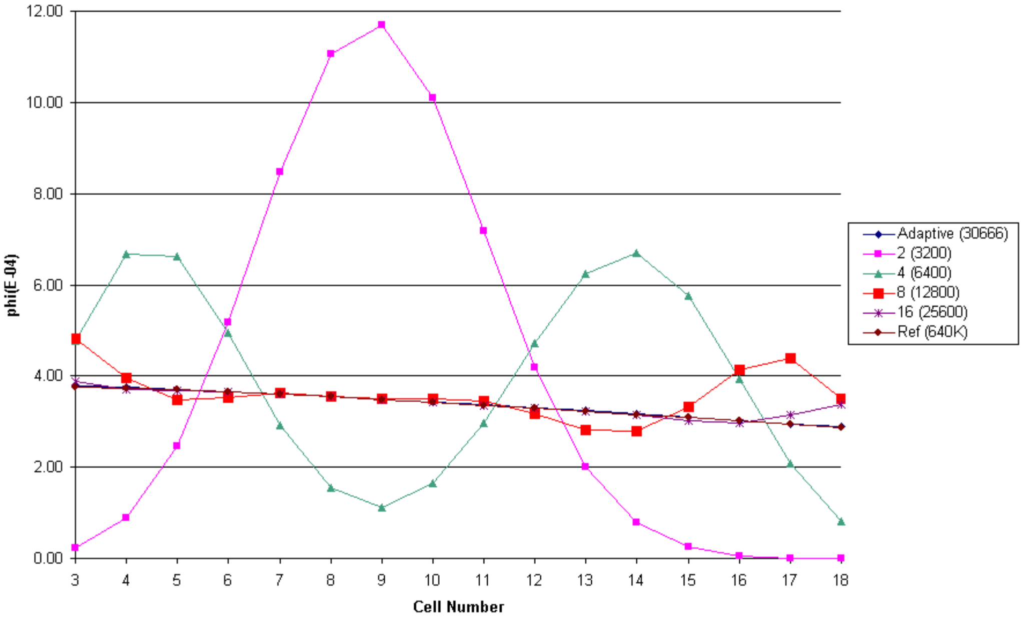


Figure 7. $\phi(x,y)$ at $x = 19.5$, showing ray-effect oscillations. (In parenthesis are total numbers of angular-flux unknowns in each problem.)

Figure 8 shows the same plot for quadrature sets with more unknowns. Consistent with previous studies of ray effects, we see that even the quadrature set with 32 angles per quadrant makes a significant error in cell 18, despite its use of 51,000 unknowns. The adaptive solution with 70,000 unknowns, which was generated by setting ε_{test} = 0.01 and $\varepsilon_{threshold}$ = 0.0001 in Eq. (6), is free from oscillations and appears superior to the 64-angle/quadrant solution that used 100,000 unknowns.

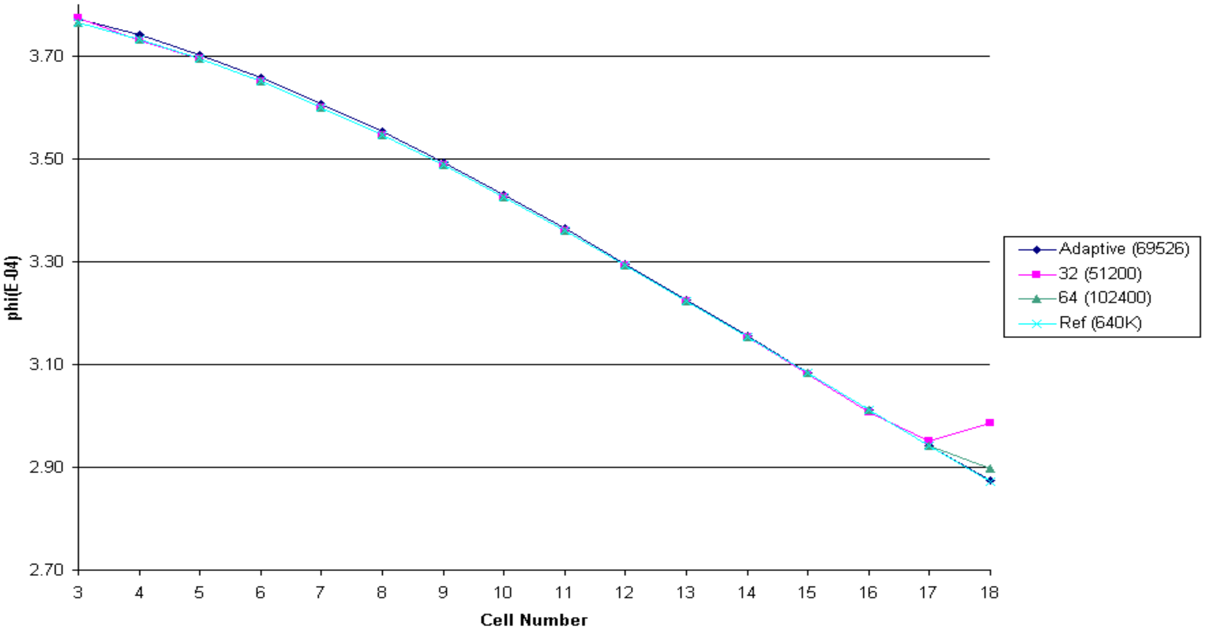


Figure 8. $\phi(x,y)$ at x = 19.5, for higher-order quadratures. (In parenthesis are total numbers of angular-flux unknowns in each problem.)

Instead of relying simply on pictures we have computed RMS errors for the 16 scalar-flux values tested in Figures 7 and 8. We show these in Table I, where we list unknown counts along with RMS errors for uniform quadrature sets as well as the two adaptive sets described in the preceding paragraphs.

Table I. Unknowns and RMS values for various quadrature sets.

Method		Number of Unknowns	RMS value
Adaptive	Adaptive (30666)	30666	3.56E-03
fixed	2 (3200)	3200	1.22E+00
fixed	4 (6400)	6400	6.20E-01
fixed	8 (12800)	12800	1.71E-01
fixed	16 (25600)	25600	4.15E-02
Adaptive	Adaptive (69526)	69526	1.51E-03
fixed	32 (51200)	51200	8.41E-03
fixed	64 (102400)	102400	1.83E-03
fixed	Ref (640K)	640000	0

We have run our adaptive algorithm with a variety of refinement criteria, which produce a variety of unknown counts and RMS errors. It is interesting to plot these along with similar data from uniform quadrature sets; we do so in Figure 9. Here we clearly see that the adaptive sets are far better at producing lower errors with fewer unknowns, as they of course should be. We emphasize that our Figure 9 should become more impressive as we learn more about how to implement our adaptation logic more efficiently. In particular, we could certainly reduce our unknown count by using more and smaller spatial regions. In the near future we will investigate the tradeoff between this reduced unknown count and the increased overhead of having more, smaller regions.

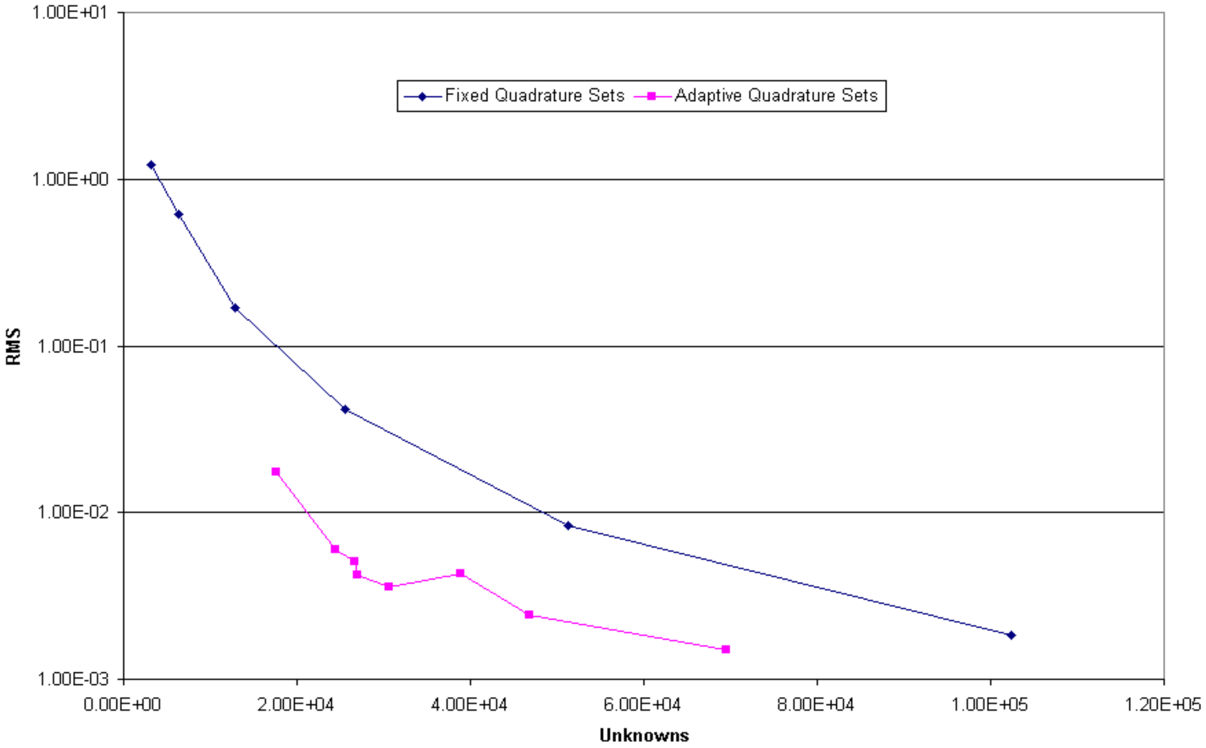


Figure 9. RMS error in $\phi(x,y)$ at x = 19.5, as a function of unknown count, for uniform and adaptive quadrature sets.

6. CONCLUSIONS

We have presented our initial efforts toward an adaptive discrete-ordinates quadrature methodology. We have used very simple algorithms and we have left many questions unanswered, but we find our results to date to be very encouraging. We have shown that it is possible to adapt quadrature sets to resolve local variations in angular fluxes, and we have shown that even very simple adaptation logic combined with very simple linear interpolation and trapezoidal-rule quadrature sets can achieve highly accurate solutions with relatively few unknowns.

There remains a great deal of future work. Our first task is to implement our strategies with higher-order quadrature sets and interpolation functions. This should pay off handsomely in terms of unknowns needed to achieve a given accuracy; the challenge will be to reach this payoff without inordinate overhead costs. We plan to explore strategies for defining spatial regions that share the same quadrature sets, and hopefully devise algorithms for automating these region definitions. We will also test our ever-evolving adaptive capability on more and more kinds of problems as time goes on. We expect to learn a great deal and share what we learn in future communications.

ACKNOWLEDGMENTS

We thank Paul Nelson, Igor Carron, Ed Larsen, Jim Morel, Britton Chang, and Peter Brown for helpful discussions over the years related to adaptive D-O strategies. This work was partially supported by a grant from the DOE Nuclear Engineering Education Research Program.

REFERENCES

1. G. Longoni and A. Haghighat, “Development Of New Quadrature Sets With The “Ordinate Splitting” Technique,” *Proc. International Meeting on Mathematical Methods for Nuclear Applications,* Salt Lake City, UT, September 9-13, 2001, pp.9-13 (2001).
2. G. Longoni and A. Haghighat, “A Spatial Discretization for Solving the Transport Equation on Unstructured Grids of Polyhedra,” *Proc. International Conference on Mathematics and Computation, Reactor Physics and Environmental Analysis in Nuclear Applications*, Madrid, Spain, September 27-30, 1999, Vol. II, pp.1196-1204, Senda Editorial (1999).
3. RSIC Computer Code Collection. TORT-DORT: Two- and Three- Dimensional Discrete Ordinates Transport Version 2.12.14, Radiation Shielding Information Center, Oak Ridge National Laboratory, 1995.
4. R. P. Smedley-Stevenson, “Thermal Radiation Transport on Unstructured Finite Element Meshes,” *Trans. Am. Nucl. Soc.*, **80**, pp.243-246 (2000).
5. S. D. Pautz and M. L. Adams, “An Asymptotic Study of Discretized Transport Equations in the Fokker-Planck Limit,” *Nucl. Sci. Eng*., **140**, pp.51-69 (2002).
6. J. E. Morel, “A Hybrid Collocation-Galerkin-Sn Method for Solving the Boltzmann Transport Equation,” *Nucl. Sci. Eng.*, **101**, pp.72 (1989).
7. B. G. Carlson and K. D. Lathrop, “Discrete Ordinates Angular Quadrature of the Neutron Transport Equation,” Los Alamos Scientific Laboratory Report, LA-3186, February 1965.
8. B. G. Carlson, “Transport Theory: Discrete Ordinates Quadrature over the Unit Sphere,” Los Alamos Scientific Laboratory Report, LA-4554, December 1970.