UNIC Code: Algorithmic Specification of the Method of Long Characteristics

Technical Memorandum ANL/MCS-TM-301, June 2008
Division of Mathematics and Computer Science
Argonne National Laboratory

Stephen F. Siegel
Department of Computer and Information Sciences
103 Smith Hall
University of Delaware
Newark, DE 19716, USA
siegel@cis.udel.edu

Andrew R. Siegel
Mathematics and Computer Science Division
Argonne National Laboratory
9700 S. Cass Avenue
Argonne, IL 60439, USA
siegela@mcs.anl.gov

Cristian Rabiti
Nuclear Engineering Division
Argonne National Laboratory
9700 S. Cass Avenue
Argonne, IL 60439, USA
crabiti@anl.gov

This page intentionally left blank.

Contents

Abstract 1
1. The Steady State Neutron Transport Equation 2
1.1. Fundamental Equations 2
1.2. Energy Discretization 2
2. Geometry 3
2.1. Spatial Discretization 3
2.2. Integral Approximations 4
2.3. Boundary Condition 5
3. Outer Algorithm 5
4. Middle Algorithms 6
5. Inner Algorithm 8
6. Ray-tracing Algorithm 9
7. Conformance 12
References 12

Abstract

The purpose of this document is to specify the method of long characteristics algorithm used in the UNIC neutron transport code.

1. The Steady State Neutron Transport Equation

1.1. **Fundamental Equations.** In this section we briefly review the fundamental integro-differential equation that we wish to solve. It is beyond the scope of this paper to describe the equation and its physical significance in detail. Such details are covered e.g. in [1].

Given σ, σ^s, σ^f, χ, and ν, we wish to solve the steady state neutron transport equation for K_{eff} and ψ:

$$\begin{aligned}\big[\hat{\Omega}\cdot\vec{\nabla}+\sigma(\vec{r},E)\big]\psi(\vec{r},\hat{\Omega},E) &= \int dE'\int d\hat{\Omega}'\,\sigma^s(\vec{r},E'\to E,\hat{\Omega}'\cdot\hat{\Omega})\psi(\vec{r},\hat{\Omega}',E') \\ &\quad+\frac{\chi(E)}{4\pi K_{\mathit{eff}}}\int dE'\,\nu\sigma^f(\vec{r},E')\int d\hat{\Omega}'\,\psi(\vec{r},\hat{\Omega}',E')\end{aligned} \tag{1}$$

The notation is summarized in Figure 1. Notice that both sides of (1) are functions of $\vec{r}$, $\hat{\Omega}$, and E. The integral over energy (E') runs from 0 to ∞; the integral over angle $\hat{\Omega}'$ is an integral over the surface of the unit sphere S^2 (which has area 4π).

For simplicity in this version of the specification we assume *isotropic scattering*: i.e., σ^s is independent of μ. Note that when modeling nuclear reactor cores (the principle application domain of UNIC) this is not necessarily a good assumption and instead we will require an expansion of the flux to at least several P_n modes. This and other features will be included in a subsequent version of the specification.

For the isotropic scattering case we can write $\sigma^s(\vec{r},E'\to E)$ for $\sigma^s(\vec{r},E'\to E,\mu)$ for any μ. Then (1) becomes

$$\begin{aligned}\big[\hat{\Omega}\cdot\vec{\nabla}+\sigma(\vec{r},E)\big]\psi(\vec{r},\hat{\Omega},E) &= \int dE'\,\sigma^s(\vec{r},E'\to E)\phi(\vec{r},E') \\ &\quad+\frac{\chi(E)}{K_{\mathit{eff}}}\int dE'\,\nu\sigma^f(\vec{r},E')\phi(\vec{r},E'),\end{aligned} \tag{2}$$

where

$$\phi(\vec{r},E)=\frac{1}{4\pi}\int_{S^2}\psi(\vec{r},\hat{\Omega},E)\,d\hat{\Omega}$$

is the *normalized isotropic scalar flux.* Notice that in this case the right hand side of (2) is independent of $\hat{\Omega}$.

1.2. **Energy Discretization.** We now assume the energy dimension has been discretized by specifying $G+1$ real numbers $E_G<E_{G-1}<\ldots<E_1<E_0$. For $1\le g\le G$, *energy group* g is defined to be the interval of real numbers $[E_g,E_{g-1}]$. We let

$$\psi_g(\vec{r},\hat{\Omega})=\int_g dE\,\psi(\vec{r},\hat{\Omega},E) \tag{3}$$

$$\phi_g(\vec{r})=\frac{1}{4\pi}\int d\hat{\Omega}\,\psi_g(\vec{r},\hat{\Omega}). \tag{4}$$

Recasting (2) in terms of these discrete energy groups requires the calculation of *multigroup cross sections.* The simplest way to do this is to assume that for each g, there is a function $f(E)$ defined on $[E_g,E_{g-1}]$ such that

$$\psi(\vec{r},E,\hat{\Omega})\approx f(E)\psi_g(\vec{r},\hat{\Omega}).$$

symbol	meaning
$\vec{r}$	position vector
v	neutron speed
E	neutron energy ($\frac{1}{2}mv^2$)
$\hat{\Omega}$	direction of neutron motion (θ, ϕ)
$\psi(\vec{r}, \hat{\Omega}, E)$	neutron angular flux
$\phi(\vec{r}, E)$	neutron scalar flux: $\frac{1}{4\pi}\int d\hat{\Omega}\, \psi(\vec{r}, \hat{\Omega}, E)$
$\sigma(\vec{r}, E)$	total interaction cross section
μ	cosine of scatter angle
$\sigma^s(\vec{r}, E' \to E, \mu)$	macroscopic scattering cross section
$\sigma^s(\vec{r}, E' \to E)$	macroscopic scattering cross section (isotropic case)
$\sigma^f(\vec{r}, E)$	macroscopic fission cross section
$\chi(E)$	energy distribution for fissioned neutron evaluated at E
ν	average number of neutrons emitted per fission
K_{eff}	a generalized eigenvalue

FIGURE 1. Symbols used in describing the linear Boltzman equation

Then define

$$\sigma_g(\vec{r}) = \int_g dE\, \sigma(\vec{r}, E) f(E) \tag{5}$$

$$\sigma_g^f(\vec{r}) = \int_g dE\, \sigma^f(\vec{r}, E) f(E) \tag{6}$$

$$\sigma_{gg'}^s(\vec{r}, \mu) = \int_g dE \int_{g'} dE'\, \sigma^s(\vec{r}, E' \to E, \mu) f(E') \tag{7}$$

To simplify the presentation, we shall just assume that suitably calculated multigroup cross sections are provided. For details, see [1, 2-2].

Discretizing (2), we arrive at one equation for each group g:

$$\left[\hat{\Omega} \cdot \vec{\nabla} + \sigma_g(\vec{r})\right] \psi_g(\vec{r}, \hat{\Omega}) = \sum_{g'} \sigma_{gg'}^s(\vec{r}) \phi_{g'}(\vec{r}) + \frac{\chi_g}{K_{\mathit{eff}}} \sum_{g'} \nu \sigma_{g'}^f(\vec{r}) \phi_{g'}(\vec{r}). \tag{8}$$

2. GEOMETRY

2.1. **Spatial Discretization.** We assume the domain is partitioned into N distinct regions V_n $(1 \le n \le N)$, called *elements*. The volume of V_n is $vol[n]$. The *average scalar flux* for group g in V_n is defined to be

$$\phi_g[n] = \frac{1}{vol[n]} \int_{V_n} \phi_g(\vec{r})\, dV.$$

The angular space, which can be identified with the unit sphere S^2, is partitioned into I distinct "angular regions." For each i, $\hat{\Omega}_i$ will denote some fixed angle in angular region i. We let $\Delta\hat{\Omega}_i$ denote the quotient of the area (on the unit sphere) of the i^{th} region by 4π.

Since 4π is the area of S^2, we have

$$\sum_{i=1}^{I} \Delta\hat{\Omega}_i = 1.$$

Moreover, the integral of a function f defined on S^2 can be approximated as follows:

$$\frac{1}{4\pi} \int_{S^2} f(\hat{\Omega})\, d\hat{\Omega} \approx \sum_{i=1}^{I} f_i \Delta\hat{\Omega}_i,$$

where f_i is the representative value for f in angular region i.

For each i, we are given J_i *trajectories* (i.e., rays, also referred to as "tracks") in the direction $\hat{\Omega}_i$. The trajectories are numbered $1, \ldots, J_i$. Each trajectory has associated to it a *trajectory cylinder* with cross-sectional area $\Delta S_{i,j}$. One way to think of this is to associate to each i a plane P such that lines perpendicular to P are oriented in the direction of $\hat{\Omega}_i$. Then a finite region of P can be partitioned into J_i disjoint subregions, with one starting point per subregion, and $\Delta S_{i,j}$ is the area of the j^{th} such subregion. In any case, the volume of a "slice" of cylidner j for angular region i is the product of $\Delta S_{i,j}$ and the length of that slice.

Each trajectory enters the domain, passes through a sequence of elements, and then exits the domain. (We will assume the domain being modeled is convex so that it is not possible to exit and re-enter the domain.) We let $K_{i,j}$ denote the number of elements encountered by trajectory j of angular region i, and we let $\iota(i,j,k)$ $(1 \leq k \leq K_{i,j})$ denote the index n of the k^{th} element intersected. The length of the intersection of the trajectory with V_n is denoted $\ell(i,j,k)$. The point at which the ray enters V_n is denoted $\mathbf{p}_{i,j,k}$. We also let $\mathbf{p}_{i,j,K_{i,j}+1}$ be the point at which the ray exits the domain.

We let

$$\psi_g[i,j,k] = \psi_g(\mathbf{p}_{i,j,k}, \hat{\Omega}_i),$$

though we note that the algorithms only need to store these quantities for $k = 1$. For this reason we set

$$\psi_g[i,j] = \psi_g[i,j,1].$$

This is the angular flux in direction $\hat{\Omega}_i$ at the point at which the j^{th} trajectory for $\hat{\Omega}_i$ first intersects the domain.

2.2. **Integral Approximations.** If the trajectories have been chosen appropriately then for any n and any i, we can consider all trajectories in direction $\hat{\Omega}_i$ that intersect V_n, and we can form the slice of each trajectory cyclinder whose axis is the intersection of the trajectory with V_n. These cylinder slices should roughly partition V_n. This allows us to approximate the integral of a function f defined on V_n as follows:

$$\int_{V_n} f(\vec{r}, \hat{\Omega})\, dV \approx \sum_{j=1}^{J_i} \sum_{\substack{k \\ 1 \leq k \leq K_{i,j} \\ \iota(i,j,k)=n}} \Delta S_{i,j} \int_0^{\ell(i,j,k)} f(\mathbf{p}_{i,j,k} + t\hat{\Omega})\, dt \tag{9}$$

Note that, for any i and j, the sum over k consists of at most 1 term; if it consists of 0 terms it is considered to be 0.

If we apply this to the function $f \equiv 1$, the left hand side of (9) is precisely $vol[n]$, and we define the right hand side to be $vol'[i,n]$, i.e.,

$$vol[n] \approx \sum_{j=1}^{J_i} \sum_{\substack{k \\ 1 \le k \le K_{i,j} \\ \iota(i,j,k)=n}} \ell(i,j,k)\,\Delta S_{i,j} \equiv vol'[i,n]. \tag{10}$$

In fact, we can use (10) to improve the accuracy of the approximation made in (9), as follows:

$$\int_{V_n} f(\vec{r},\hat{\Omega})\,dV \approx \frac{vol[n]}{vol'[i,n]} \sum_{j=1}^{J_i} \sum_{\substack{k \\ 1 \le k \le K_{i,j} \\ \iota(i,j,k)=n}} \Delta S_{i,j} \int_0^{\ell(i,j,k)} f(\mathbf{p}_{i,j,k} + t\hat{\Omega})\,dt \tag{11}$$

In particular, approximation (11) is precise for $f \equiv 1$.

2.3. **Boundary Condition.** The behavior at the boundary of the modeled domain is specified by real numbers $\rho(i,j;i',j')$ $(1 \le i,i' \le I,\ 1 \le j \le J_i,\ 1 \le j' \le J_{i'})$. These numbers specify how neutrons exiting the domain along trajectory j of angular region i re-enter the domain. A neutron traveling along a path within $\Delta\hat{\Omega}_i\,\Delta S_{i,j}$ may re-enter along a path within $\Delta\hat{\Omega}_{i'}\,\Delta S_{i',j'}$ with probability

$$\rho(i,j;i',j').$$

If the boundary is purely reflective (no neutrons are lost), then for any i,j,

$$\sum_{i',j'} \rho(i,j;i',j') = 1.$$

If the boundary is a vaccum (neutrons never return), then $\rho \equiv 0$.

3. Outer Algorithm

We now turn to the detailed description of the MOC algorithm for solving equation (2). The parameters that are considered inputs to the algorithm and are not modified at any time by the algorithm are listed in Figure 2.

The high-level pseudcode for the MOC algorithm is given in Figure 3. This is called the *outer* algorithm because its convergence loop is the outer-most loop in the MOC algorithm. The outer algorithm relies on subroutines, described in subsequent sections, which rely on their own convergence loops.

The outer algorithm is provided with all of the global constants, together with initial guesses for K_{eff}, the values of ψ on the boundary of the domain, and the values of ϕ throughout the domain. These guesses determine values for the fission component of the source, i.e., for

$$Q_g^f(\vec{r}) = \frac{\chi_g}{K_{\mathit{eff}}} \sum_{g'} \nu\sigma_{g'}^f(\vec{r})\phi_{g'}(\vec{r}).$$

The Q_g^f are integrated over space and energy to yield a single scalar quantity called the *total fission*. Now, equation (8) can be written

$$\left[\hat{\Omega}\cdot\vec{\nabla} + \sigma_g(\vec{r})\right]\psi_g(\vec{r},\hat{\Omega}) = \sum_{g'} \sigma_{gg'}^s(\vec{r})\phi_{g'}(\vec{r}) + Q_g^f(\vec{r}). \tag{12}$$

symbol	meaning
N	the number of finite elements
$vol[n]$	the volume of element n $(1 \le n \le N)$
I	number of angles
$\Delta\hat{\Omega}_i$	weight of the i^{th} angle, divided by 4π
J_i	number of trajectories for angle i $(1 \le i \le I)$
$\Delta S_{i,j}$	the cross-sectional area of trajectory j for angle i
$K_{i,j}$	number of elements intersected by trajectory j of angle i $(1 \le i \le I,\ 1 \le j \le J_i)$
$\iota(i,j,k)$	index of k^{th} element intersected by trajectory j of angle i $(1 \le i \le I,\ 1 \le j \le J_i,\ 1 \le k \le K_{i,j})$
$\ell(i,j,k)$	length of intersecting segment above
$vol'[i,n]$	$\displaystyle\sum_{j=1}^{J_i} \sum_{\substack{k \\ 1 \le k \le K_{i,j} \\ \iota(i,j,k)=n}} \ell(i,j,k)\,\Delta S_{i,j}$
$\rho(i,j;i',j')$	reflective boundary condition coefficients
G	the number of energy groups
H	beginning of up-scattering region
$\sigma_g[n]$	total macroscopic cross section for element n, group g
$\sigma^s_{gg'}[n]$	macroscopic scattering cross section from g' to g for element n
$\sigma^f_g[n]$	macroscopic fission cross section for element n, group g
χ_g	discretized energy distribution for fission at group g
ν	average number of neutrons emitted per fission
ϵ_K	tolerance for K_{eff}
ϵ_ϕ	tolerance for scalar flux in outer algorithm
ϵ'_ϕ	tolerance for scalar flux in up-down-scatter algorithm
ϵ''_ϕ	tolerance for scalar flux in inner algorithm

FIGURE 2. Global parameters used by all algorithms

Considering the Q^f_g to be fixed, we arrive at a system of G equations that must be solved simultaneously for ψ (on the boundary) and ϕ (throughout). We will discuss shortly the procedure for solving this system. Once an approximate solution has been achieved, the new values for ϕ are used to calculate new values for the Q^f_g, and the total fission is re-computed. The ratio between the new and old values for the total fission is used to update the estimate for K_{eff}, and we proceed to solve the new simultaneous system. Iteration stops when the rates of change of K_{eff} and of ϕ fall below specified tolerances.

4. Middle Algorithms

We now turn to the "middle" algorithms that are used to solve the simultaneous systems described above. Typically, the scattering matrix $(\sigma^s_{gg'})$ is "largely" lower triangular. That is, there is some H $(1 \le H \le G)$, usually close to G, such that $\sigma_{gg'} \equiv 0$ whenever $g < H$ and $g' > g$. (This reflects the fact that, for the most part, the neutrons tend to lose energy after scattering, except perhaps in the very low end of the energy spectrum.) In this "down-scattering only" region, the system of equations can be solved sequentially: equation (12)

In : *all global parameters (Fig. 2)*
InOut: K_{eff}, $\phi_g[n]$ $(1 \le g \le G,\ 1 \le n \le N)$
$\psi_g[i,j]$ $(1 \le g \le G,\ 1 \le i \le I,\ 1 \le j \le J_i)$
Data : $Q^{in}_g[n]$ $(1 \le g \le G,\ 1 \le n \le N)$
$\phi'_g[n]$ $(1 \le g \le G,\ 1 \le n \le N)$
$\mathit{fission}[n]$ $(1 \le n \le N)$

$\mathit{totalFission} \leftarrow 0$;
for $n \leftarrow 1$ **to** N **do**
  $\mathit{fission}[n] \leftarrow \sum_g \phi_g[n]\nu\sigma^f_g[n]$;
  $\mathit{totalFission} \leftarrow \mathit{totalFission} + \mathit{vol}[n]\mathit{fission}[n]$;
  foreach g **do** $Q^{in}_g[n] \leftarrow \sum_{g' \neq g} \phi_{g'}[n]\sigma^s_{gg'}[n] + \chi_g\, \mathit{fission}[n]/K_{\mathit{eff}}$;
end
repeat
  $\mathit{downScatterOnly}(\phi_{1\ldots G}; \psi_{1\ldots H-1}, Q^{in}_{1\ldots G}; \phi'_{1\ldots H-1})$;
  $\mathit{upAndDownScatter}(\phi_{H\ldots G}; \psi_{H\ldots G}, Q^{in}_{H\ldots G}; \phi'_{H\ldots G})$;
  $\mathit{err}_\phi \leftarrow ||\phi' - \phi||$;
  $\mathit{totalFission}' \leftarrow \sum_{g,n} \mathit{vol}[n]\phi'_g[n]\nu\sigma^f_g[n]$;
  $K'_{\mathit{eff}} \leftarrow K_{\mathit{eff}}\, \mathit{totalFission}'/\mathit{totalFission}$;
  $\mathit{err}_{K_{\mathit{eff}}} \leftarrow |K'_{\mathit{eff}} - K_{\mathit{eff}}|/K'_{\mathit{eff}}$;
  for $n \leftarrow 1$ **to** N **do**
    $t \leftarrow \mathit{fission}[n]$;
    $\mathit{fission}[n] \leftarrow \sum_g \phi'_g[n]\nu\sigma^f_g[n]$;
    foreach g **do** $Q^{in}_g[n] \leftarrow Q^{in}_g[n] + \chi_g\, \mathit{fission}[n]/K'_{\mathit{eff}} - \chi_g t/K_{\mathit{eff}}$;
  end
  $K_{\mathit{eff}} \leftarrow K'_{\mathit{eff}}$;
  $\mathit{totalFission} \leftarrow \mathit{totalFission}'$;
  foreach g, n **do** $\phi_g[n] \leftarrow \phi'_g[n]$;
until $\mathit{err}_{K_{\mathit{eff}}} \le \epsilon_K \wedge \mathit{err}_\phi \le \epsilon_\phi$;

FIGURE 3. Outer algorithm

In : $\phi_g[n]$ $(1 \le g \le G,\ 1 \le n \le N)$
InOut: $\psi_h[i,j]$ $(1 \le h < H,\ 1 \le i \le I,\ 1 \le j \le J_i)$
$Q^{in}_g[n]$ $(1 \le g \le G,\ 1 \le n \le N)$
Out : $\phi'_g[n]$ $(1 \le g < H,\ 1 \le n \le N)$

for $h \leftarrow 1$ **to** $H-1$ **do**
  $\mathit{inner}(h, \phi_h, Q^{in}_h; \psi_h; \phi'_h)$;
  for $g \leftarrow h+1$ **to** G **do**
    foreach n **do** $Q^{in}_g[n] \leftarrow Q^{in}_g[n] + (\phi'_h[n] - \phi_h[n])\sigma^s_{gh}[n]$;
  end
end

FIGURE 4. *downScatterOnly*

In : $\phi_g[n]$ $(H \le g \le G,\ 1 \le n \le N)$
InOut: $\psi_g[i,j]$, $Q_g^{in}[n]$ $(H \le g \le G,\ 1 \le i \le I,\ 1 \le j \le J_i,\ 1 \le n \le N)$
Out : $\phi'_g[i]$ $(H \le g \le G,\ 1 \le i \le N)$

foreach $g \in \{H, \ldots, G\}$, $n \in \{1, \ldots, N\}$ **do** $\phi'_g[n] \leftarrow \phi_g[n]$;
repeat
  $err_{\phi'} \leftarrow 0$;
  for $g \leftarrow H$ **to** G **do**
    $inner(g, \phi'_g, Q_g^{in}; \psi_g; \phi'')$;
    foreach $g' \in \{H, \ldots, G\} \setminus \{g\}$ **do**
      foreach n **do** $Q_{g'}^{in}[n] \leftarrow Q_{g'}^{in}[n] + (\phi''[n] - \phi'_g[n])\sigma_{g'g}^s[n]$;
    end
    $err_{\phi'} \leftarrow err_{\phi'} + ||\phi'' - \phi'_g||$;
    foreach $n \in \{1, \ldots, N\}$ **do** $\phi'_g[n] \leftarrow \phi''[n]$;
  end
until $err_{\phi'} \le \epsilon'_\phi$;

FIGURE 5. *upAndDownScatter*

for $g = 1$ involves only ψ_1 and ϕ_1, and not any ψ_k or ϕ_k for $k > 1$. (We will discuss shortly the subroutine for solving this equation.) The solutions for $g = 1$ can then be substituted into equation (12) for $g = 2$, which then involves only ψ_2 and ϕ_2, which is then solved, and so on. The precise algorithm dealing with the down-scattering-only region is given in Figure 4.

For the region $g \ge H$, where up and down scattering are possible, an iteration scheme is used. This algorithm is described in Figure 5.

5. Inner Algorithm

In : g, $\phi[n]$, $Q^{in}[n]$ $(1 \le n \le N)$
InOut: $\psi[i,j]$ $(1 \le i \le I,\ 1 \le j \le J_i)$
Out : $\phi'[n]$ $(1 \le n \le N)$
Data : $\phi''[n]$, $Q[n]$ $(1 \le n \le N)$

foreach n **do** $\phi'[n] \leftarrow \phi[n]$;
repeat
  foreach $n \in \{1, \ldots, N\}$ **do**
    $\phi''[n] \leftarrow \phi'[n]$;
    $Q[n] \leftarrow Q^{in}[n] + \sigma_{gg}^s[n]\phi''[n]$;
  end
  $raytrace(g, Q; \psi; \phi')$;
until $||\phi' - \phi''|| \le \epsilon''_\phi$;

FIGURE 6. *inner*

Both of the middle algorithms rely on a subroutine to solve an equation involving only one group g. Let

$$Q_g^{in}(\vec{r}) = \sum_{g' \neq g} \phi_{g'}(\vec{r}) \sigma_{gg'}^s(\vec{r}) + \frac{\chi_g}{K_{eff}} \sum_{g'} \phi_{g'}(\vec{r}) \nu \sigma_{g'}^f(\vec{r}). \tag{13}$$

Then we can re-write (12) as the *in-group equation*

$$\left[\hat{\Omega} \cdot \vec{\nabla} + \sigma_g(\vec{r})\right] \psi_g(\vec{r}, \hat{\Omega}) = \phi_g(\vec{r}) \sigma_{gg}^s(\vec{r}) + Q_g^{in}(\vec{r}). \tag{14}$$

In (14), Q_g^{in} is considered fixed, and the problem is to solve for ψ_g (on the boundary) and ϕ_g (throughout). The algorithm to do this is called the *inner* algorithm, and is described in Figure 6.

The inner algorithm begins with the current best estimate of ϕ_g. Given this, we can compute the *total source*

$$Q_g(\vec{r}) = \phi_g(\vec{r}) \sigma_{gg}^s(\vec{r}) + Q_g^{in}(\vec{r}). \tag{15}$$

Equation (14) then becomes

$$\left[\hat{\Omega} \cdot \vec{\nabla} + \sigma_g(\vec{r})\right] \psi_g(\vec{r}, \hat{\Omega}) = Q_g(\vec{r}). \tag{16}$$

The problem now is to solve (16), treating the total source Q_g as fixed. This is accomplished by the *ray-tracing* algorithm, which is described in detail below. The ray-tracing algorithm takes as input the current best estimate for ψ_g on the boundary and the total source Q_g. It outputs the new boundary values of ψ_g, and the values of ϕ_g throughout the domain, approximately satisfying (16). The new values of $\phi_g(\vec{r})$ are used to update the total source $Q_g(\vec{r})$ using (15), and the ray-tracing technique is applied again. Iteration continues in this way until the relative change in ϕ_g between successive iterations falls below a specified tolerance.

6. Ray-tracing Algorithm

We now turn to the ray tracing algorithm for solving (16), the central technique of the MOC. We will drop the subscript g to simplify the notation. The pseudocode for the algorithm is given in Figure 7.

The idea of the ray-tracing algorithm is to sample the domain with a set of rays that pass through the domain at various angles $\hat{\Omega}$ and that have various starting points. Restricted to a single ray and a single element, (16) becomes a simple first order ordinary differential equation in one variable, which can be solved analytically. One computes the solution along a ray by starting at the point at which the ray enters the boundary, where the value of ψ is given, and propagating forward to compute the value of ψ at successive elements intersected by the ray. As we shall see, the average scalar flux ϕ in an element can be approximated by a sum. Each ray passing through the element contributes one term to the sum. That term is a function of the value of ψ where the ray enters the element and the value where the ray exits the element. After each new value of ψ is computed, the contribution to the scalar flux is computed and added to the variable storing ϕ for that element. The incoming value of ψ can then be forgotten. This is how the algorithm avoids storing ψ at all elements.

The analytical solution is obtained as follows. Assume that

$$\mathbf{p} = (x_0, y_0, z_0) \in \mathbb{R}^3$$

In : g, $Q[n]$ $(1 \le n \le N)$
InOut: $\psi[i,j]$ $(1 \le i \le I,\ 1 \le j \le J_i)$
Out : $\phi'[n]$ $(1 \le n \le N)$
Data : $\psi'[i,j]$ $(1 \le i \le I,\ 1 \le j \le J_i)$
foreach $n \in \{1,\ldots,N\}$ **do** $\phi'[n] \leftarrow 0$;
foreach $i \in \{1,\ldots,I\}$, $j \in \{1,\ldots,J_i\}$ **do** $\psi'[i,j] \leftarrow 0$;
foreach $i \in \{1,\ldots,I\}$, $j \in \{1,\ldots,J_i\}$ **do**
  $\psi_{out} \leftarrow \psi[i,j]$;
  for $k \leftarrow 1$ **to** $K_{i,j}$ **do**
    $\psi_{in} \leftarrow \psi_{out}$;
    $n \leftarrow \iota(i,j,k)$;
    $\gamma \leftarrow e^{-\sigma_g[n]\ell(i,j,k)}$;
    $\beta \leftarrow (1-\gamma)/\sigma_g[n]$;
    $\psi_{out} \leftarrow \psi_{in}\gamma + \beta Q[n]$;
    $\phi'[n] \leftarrow \phi'[n] + \dfrac{\Delta\hat{\Omega}_i\,\Delta S_{i,j}\,(Q[n]\ell(i,j,k) + \psi_{in} - \psi_{out})}{\sigma[n]vol'[i,n]}$;
  end
  foreach i', j' **do** $\psi'[i',j'] \leftarrow \psi'[i',j'] + \dfrac{\Delta\hat{\Omega}_i\,\Delta S_{i,j}}{\Delta\hat{\Omega}_{i'}\,\Delta S_{i',j'}}\rho(i,j;i',j')\psi_{out}$;
end
foreach i, j **do** $\psi[i,j] \leftarrow \psi'[i,j]$;

FIGURE 7. *raytrace*

is the point at which a ray enters an element and that the ray is oriented in direction

$$\hat{\Omega} = (u,v,w) \in S^2.$$

Assume furthermore that we are given ψ_{in}, the value of ψ in direction $\hat{\Omega}$ at $\mathbf{p}$. We wish to compute ψ_{out}, the value of ψ in direction $\hat{\Omega}$ at the point at which the ray exits the element.

Let $\zeta : \mathbb{R} \to \mathbb{R}^3 \times S^2$ be the function defined by

$$\zeta(t) = (\mathbf{p} + \hat{\Omega}t, \hat{\Omega}).$$

Then

$$\zeta(t) = ((x(t), y(t), z(t)), \hat{\Omega}),$$

where

$$x(t) = x_0 + ut, \quad y(t) = y_0 + vt, \quad z(t) = z_0 + wt.$$

Let $f = \psi \circ \zeta$. Note that $f(t)$ is the value of ψ in direction $\hat{\Omega}$ at the point a distance of t along the ray from $\mathbf{p}$. If the length of the segment of the ray intersecting the element is l, then $f(l) = \psi_{out}$.

From the chain rule, we have

$$f'(t) = \frac{\partial\psi}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial\psi}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial\psi}{\partial z}\frac{\partial z}{\partial t} = u\frac{\partial\psi}{\partial x} + v\frac{\partial\psi}{\partial y} + w\frac{\partial\psi}{\partial z} = [\hat{\Omega}\cdot\vec{\nabla}\psi](\zeta(t)). \tag{17}$$

Within a single element, we assume that the source Q and cross section σ are constant, so (16) becomes

$$f'(t) + \sigma f(t) = Q. \tag{18}$$

The unique solution to (18) satisfying $f(0) = \psi_{in}$ is

$$f(t) = \psi_{in} e^{-\sigma t} + \frac{Q(1 - e^{-\sigma t})}{\sigma}. \tag{19}$$

Substituting l for t in (19) gives the desired closed-form expression for ψ_{out}:

$$\psi_{out} = \psi_{in} e^{-\sigma l} + \frac{Q(1 - e^{-\sigma l})}{\sigma}. \tag{20}$$

We now turn to the question of the computation of ϕ at all points in the domain. Integrating both sides of (18) from 0 to l, and applying the Fundamental Theorem of Calculus, we obtain

$$\psi_{out} - \psi_{in} + \sigma \int_0^l f(t)\, dt = Ql. \tag{21}$$

Hence

$$\int_0^l f(t)\, dt = \frac{Ql - \psi_{out} + \psi_{in}}{\sigma}. \tag{22}$$

By definition,

$$\begin{aligned} \phi[n] &= \frac{1}{vol[n]} \int_{V_n} \phi(\vec{r})\, dV \\ &= \frac{1}{4\pi\, vol[n]} \int_{V_n} \int_{S^2} \psi(\vec{r}, \hat{\Omega})\, d\hat{\Omega}\, dV \\ &= \frac{1}{4\pi\, vol[n]} \int_{S^2} \int_{V_n} \psi(\vec{r}, \hat{\Omega})\, dV\, d\hat{\Omega} \\ &\approx \frac{1}{vol[n]} \sum_{i=1}^{I} \Delta\hat{\Omega}_i \int_{V_n} \psi(\vec{r}, \hat{\Omega}_i)\, dV. \end{aligned} \tag{23}$$

Now for any i, we can apply (11) to the function $\psi(-, \hat{\Omega}_i)$ restricted to V_n and then use (22) to obtain

$$\begin{aligned} \int_{V_n} \psi(\vec{r}, \hat{\Omega}_i)\, dV &\approx \frac{vol[n]}{vol'[i,n]} \sum_{j=1}^{J_i} \sum_{\substack{k \\ 1 \le k \le K_{i,j} \\ \iota(i,j,k) = n}} \Delta S_{i,j} \int_0^{\ell(i,j,k)} \psi(\mathbf{p}_{i,j,k} + t\hat{\Omega}_i, \hat{\Omega}_i)\, dt \\ &= \frac{vol[n]}{vol'[i,n]} \sum_{j=1}^{J_i} \sum_{\substack{k \\ 1 \le k \le K_{i,j} \\ \iota(i,j,k) = n}} \Delta S_{i,j}\, \frac{Q\ell(i,j,k) - \psi[i,j,k+1] + \psi[i,j,k]}{\sigma} \end{aligned} \tag{24}$$

Substituting (24) into (23) yields

$$\phi[n] \approx \sum_{i=1}^{I} \sum_{j=1}^{J_i} \sum_{\substack{k \\ 1 \le k \le K_{i,j} \\ \iota(i,j,k)=n}} \frac{\Delta S_{i,j}\, \Delta \hat{\Omega}_i}{\sigma\, vol'[i,n]} \left(Q\ell(i,j,k) - \psi[i,j,k+1] + \psi[i,j,k] \right) \tag{25}$$

This is the formula that is used to update ϕ' in line 11 of Figure 7.

7. Conformance

The algorithms keep track of the average scalar flux $\phi_g[n]$ for all elements n, and the angular flux ψ on the boundary. To be precise, we let $\psi_g[i,j]$ denote the angular flux at the point at which trajectory j of angular region i enters the domain being modeled.

The main ("outer") algorithm is given in Figure 3. The input to this algorithm consists of all the quantities listed in Figure 2. This algorithm calls two subroutines, *downScatterOnly* and *upAndDownScatter*, described in Figures 4 and 5, respectively. Each of these subroutines calls *inner*, given in Figure 6, which solves equation (14) for a fixed energy group g.

An implementation is considered conformant with this specification if it performs the same computation as the algorithm of Figure 3. By "same computation," we mean the same if all arithmetic operations were carried out as (infinite precision) operations on real numbers. Hence computed results may differ due to the lack of associativity, commutativity, etc., of floating-point operations.

References

[1] E. E. Lewis and Jr. W. F. Miller. *Computational Methods of Neutron Transport*. American Nuclear Society, Inc., La Grange Park, Illinois, USA, 1993.