The collision probability method in 1D – part 2

Alain Hébert
alain.hebert@polymtl.ca

Institut de génie nucléaire
École Polytechnique de Montréal
Calculation of 1D collision probabilities
- plane 2D geometry
- cylindrical 1D geometry
- spherical 1D geometry
The collision probability method

- Consider an infinite lattice of unit cells, each of them represented as $\bigcup_i V_i$
- $V_i^\infty$ represents the infinite set of regions $V_i$ belonging to all the cells in the lattice
- The sources of secondary neutrons are uniform and equal to $Q_i$ on each region $V_i$.

The collision probability $P_{ij,g}$ is the probability for a neutron born uniformly and isotropically in any of the regions $V_i$ of the lattice to undergo its first collision in region $V_j$ of a unit cell or assembly.

If the total cross section $\Sigma(r)$ is constant and equal to $\Sigma_j$ in region $V_j$, reduced CPs are

$$p_{ij} = \frac{P_{ij}}{\Sigma_j} = \frac{1}{4\pi V_i} \int_{V_i^\infty} d^3r' \int_{V_j} d^3r \frac{e^{-\tau(s)}}{s^2}$$

(1)

where the optical path $\tau(s)$ is given by

$$\tau(s) = \int_0^s ds' \Sigma(r - s' \Omega).$$

(2)
We will now study the case of a 2D geometry defined in the $x$–$y$ plane and homogeneous along the $z$–axis. A cylinder or tube of infinite length is an example of such a geometry. In this case, it is possible to integrate analytically the integral transport equation along the axial angle $\theta$. This angle is used to define the direction cosine of the particle relative to the $z$–axis. The direction of the particle is

$$\Omega = \sin \theta \cos \epsilon \mathbf{i} + \sin \theta \sin \epsilon \mathbf{j} + \cos \theta \mathbf{k}.$$  

Integration over $\theta$ is possible by taking the projection of each particle free path on the $x$–$y$ plane. We write

$$d^2 \Omega = d\theta \, d\epsilon \, \sin \theta$$

$$\tau(\rho) = \tau(s) \, \sin \theta$$

$$d\rho = ds \, \sin \theta$$

(3)

where $\rho$ is the projection of $s$ on the $x$–$y$ plane.
Plane 2D geometry

We will limit ourself on the integration of the integral transport equation, defined on the domain of a unique unit cell. We use the relations $r = r' + s \Omega$ and $d^3r = s^2 d^2\Omega ds$ and obtain

$$p_{ij} = \frac{1}{4\pi V_i} \int_{V_i} d^3r' \int_{4\pi} d^2\Omega \int_{I_j} ds e^{-\tau(s)}$$

where the quantity $I_j$ is the set of points belonging simultaneously
- to the half straight line of origin $r$ and direction $-\Omega$;
- to volume $V_j$ of the unit cell.

With the help of figure, Eq. (4) can be rewritten as

$$p_{ij} = \frac{1}{4\pi V_i} \int_0^{2\pi} d\epsilon \int_{V_i} d^2r' \int_0^\pi d\theta \int_{I_j} d\rho e^{-\frac{\tau(\rho)}{\sin \theta}}$$

where $V_i$ now represents a surface in the $x-y$ plane and $I_j$ is the set of points belonging simultaneously
- to the half straight line of origin $r'$ and direction $\epsilon$;
- to surface $V_j$ of the unit cell.
Bickley functions are defined as

\[ K_i^n(x) = \int_0^{\pi/2} d\theta \sin^{n-1} \theta e^{-\frac{x}{\sin \theta}} = \int_0^{\pi/2} d\theta \cos^{n-1} \theta e^{-\frac{x}{\cos \theta}} \]

with \( n \geq 1 \).

The following equations are relating Bickley functions of different orders:

\[ \int_x^{x'} du \ K_i^n(u) = K_{i+1}(x) - K_{i+1}(x') \]

and

\[ \frac{d}{dx} K_i^n(x) = -K_{i-1}(x) \]

They can be evaluated efficiently using the matlab script \text{akin}. 
Equation (5) can be simplified by introducing the Bickley functions. We obtain

\[
p_{ij} = \frac{1}{2\pi V_i} \int_0^{2\pi} d\epsilon \int_{V_i} d^2r' \int_{I_j} d\rho K_i [\tau(\rho)].
\] (7)

Sets of integration lines, referred as tracks, are drawn over the complete domain. Each set is characterized by a given angle \(\epsilon\) and contains parallel tracks covering the domain. Tracks in a set can be separated by a constant distance \(\Delta h\) or can be placed at optimal locations, as depicted in sub-figures (a) and (b), respectively. Each track is used forward and backward, corresponding to angles \(\epsilon\) and \(-\epsilon\).
In the case of convex volumes $i$ and $j$, Eq. (7) can be rewritten as

\begin{equation}
 p_{ij} = \frac{1}{2\pi V_i} \int_0^{2\pi} d\epsilon \int_{h_{\text{min}}}^{h_{\text{max}}} dh \int_0^\ell_i d\ell' \int_0^\ell_j d\ell \, \text{Ki}_1(\tau_{ij} + \Sigma_i \ell' + \Sigma_j \ell) \quad \text{if } i \neq j
\end{equation}

where $\tau_{ij}$ is the optical path of the materials between regions $i$ and $j$ and

\begin{equation}
 p_{ii} = \frac{1}{2\pi V_i} \int_0^{2\pi} d\epsilon \int_{h_{\text{min}}}^{h_{\text{max}}} dh \int_0^\ell_i d\ell' \int_0^\ell_i d\ell \, \text{Ki}_1[\Sigma_i (\ell - \ell')] .
\end{equation}
In the case of 1D cylindrical geometry, the tracks are identical for any value of the angle $\epsilon$.

The tubular volumes are concave, making extra terms to appear. The corresponding geometry is depicted in figure. Two tracks are represented, both contributing to collision probability components $p_{ij}$ with $i < j$ and $p_{ii}$. Track ① corresponds to the integration domain where region $i$ is concave and track ② corresponds to the integration domain where it is convex. In this figure, $\tau_{ij}$ and $\tau_{ii}$ are the optical paths of the materials located between regions $i$ and $j$ or between two instances of region $i$. 
In this case, Eq. (8) can be rewritten as

\[
p_{ij} = \frac{2}{V_i} \left\{ \int_0^{r_{i-1/2}} dh \int_0^{\ell_i} d\ell' \int_0^{\ell_j} d\ell \left[ K_{i1} \left( (\tau_{ij} + \tau_{ii} + \Sigma_i \ell_i) + \Sigma_i \ell' + \Sigma_j \ell \right) \right. \right.
\]
\[
+ K_{i1} (\tau_{ij} + \Sigma_i \ell' + \Sigma_j \ell) \right\} \int_0^{r_{i+1/2}} dh \int_0^{\ell_i} d\ell' \int_0^{\ell_j} d\ell \left[ K_{i1} (\tau_{ij} + \Sigma_i \ell' + \Sigma_j \ell) \right] \}
\]

if \( i < j \)

where \( r_{i\pm1/2} \) are the radii bounding region \( i \) and Eq. (9) can be rewritten as

\[
p_{ii} = \frac{2}{V_i} \left\{ \int_0^{r_{i-1/2}} dh \int_0^{\ell_i} d\ell' \left[ 2 \int_0^{\ell_i} d\ell \left[ K_{i1} \left( \Sigma_i (\ell - \ell') \right) \right] + \int_0^{\ell_i} d\ell \left[ K_{i1} \left( \tau_{ii} + \Sigma_i (\ell' + \ell) \right) \right] \right) \right. \}
\]
\[
+ \int_0^{r_{i+1/2}} dh \int_0^{\ell_i} d\ell' \int_0^{\ell_i} d\ell \left[ K_{i1} \left( \Sigma_i (\ell - \ell') \right) \right] \}
\]

where the first and second terms are the contributions from tracks 1 and 2, respectively.
These equations can be simplified by introducing the following definitions:

\begin{equation}
\mathcal{C}_{ij}(\tau_0) = \int_0^{\ell_i} d\ell' \int_0^{\ell_j} d\ell \, K_1(\tau_0 + \Sigma_i \ell' + \Sigma_j \ell)
\end{equation}

and

\begin{equation}
\mathcal{D}_i = \int_0^{\ell_i} d\ell' \int_0^{\ell_i} d\ell \, K_1[\Sigma_i (\ell - \ell')]
\end{equation}

so that

\begin{equation}
p_{ij} = \frac{2}{V_i} \left\{ \int_0^{r_i-1/2} dh \left[ \mathcal{C}_{ij}(\tau_{ij} + \tau_{ii} + \Sigma_i \ell_i) + \mathcal{C}_{ij}(\tau_{ij}) \right] + \int_{r_i-1/2}^{r_i+1/2} dh \, \mathcal{C}_{ij}(\tau_{ij}) \right\}
\end{equation}

if \( i < j \), and

\begin{equation}
p_{ii} = \frac{2}{V_i} \left\{ \int_0^{r_i-1/2} dh \left[ 2\mathcal{D}_i + \mathcal{C}_{ii}(\tau_{ii}) \right] + \int_{r_i-1/2}^{r_i+1/2} dh \, \mathcal{D}_i \right\}.
\end{equation}
Integration in $\ell$ and $\ell'$ is next performed, leading to the following relations:

**a)** $\Sigma_i \neq 0$ and $\Sigma_j \neq 0$:

$$C_{ij}(\tau_0) = \frac{1}{\Sigma_i \Sigma_j} \left[ K_i(\tau_0) - K_i(\tau_0 + \ell_i) - K_i(\tau_0 + \ell_j) + K_i(\tau_0 + \Sigma_i \ell_i + \Sigma_j \ell_j) \right]$$

**b)** $\Sigma_i = 0$ and $\Sigma_j \neq 0$:

$$C_{ij}(\tau_0) = \frac{\ell_i}{\Sigma_j} \left[ K_{i2}(\tau_0) - K_{i2}(\tau_0 + \Sigma_j \ell_j) \right]$$

**c)** $\Sigma_i \neq 0$ and $\Sigma_j = 0$:

$$C_{ij}(\tau_0) = \frac{\ell_j}{\Sigma_i} \left[ K_{i2}(\tau_0) - K_{i2}(\tau_0 + \Sigma_i \ell_i) \right]$$

**d)** $\Sigma_i = \Sigma_j = 0$:

$$C_{ij}(\tau_0) = \ell_i \ell_j \ K_1(\tau_0)$$
Cylindrical 1D geometry

\( e \) \( \sum_i \neq 0 \) :

\[ D_i = \frac{\ell_i}{\sum_i} - \frac{1}{\sum_i^2} \left[ K_{i3}(0) - K_{i3}(\sum_i \ell_i) \right] \]

\( f \) \( \sum_i = 0 \) :

\[ D_i = \frac{\pi \ell_i^2}{4} \]
The above relations are implemented in Matlab using the following two scripts:

```matlab
function f=cij_f(tau0,sigi,sigj,segmenti,segmentj)
    if sigi ~= 0 && sigj ~= 0
        f=(akin(3,tau0)-akin(3,tau0+sigi*segmenti)- ...
            akin(3,tau0+sigj*segmentj)+ ...  
            akin(3,tau0+sigi*segmenti+sigj*segmentj))/(sigi*sigj) ;
    elseif sigi == 0 && sigj ~= 0
        f=(akin(2,tau0)-akin(2,tau0+sigj*segmentj))*segmenti/sigj ;
    elseif sigi ~= 0 && sigj == 0
        f=(akin(2,tau0)-akin(2,tau0+sigi*segmenti))*segmentj/sigi ;
    else
        f=akin(1,tau0)*segmenti*segmentj ;
    end

function f=di_f(sig,segment)
    if sig ~= 0
        f=segment/sig-(akin(3,0)-akin(3,sig*segment))/sig^2 ;
    else
        f=pi*segment^2/4 ;
    end
```

The collision probability method in 1D – part 2 – 14/25
The tracking of 1D cylindrical and spherical geometries is similar and can be generated with the same algorithm.

A matlab script \( f = \text{sytb1d}(\text{rad}, \text{lgsp}, \text{ngauss}) \) is presented to produce a tracking object in these cases.

The figure presents an example with \( I = 2 \) regions and \( K = 2 \) tracks per regions. Tracks \( 1 \) and \( 2 \) have two segments, identified as \( \ell_1 \) and \( \ell_2 \). Tracks \( 3 \) and \( 4 \) have only one segment, identified as \( \ell_2 \).
A $K$ point Gauss-Jacobi quadrature is used.

The four tracks are obtained using

```plaintext
rad=[0.075 0.15] ;
track=sybtld(rad,false,2) ;
```

`track(i,j)` is a 2D cell array containing $I \times K$ elements, one per track. The $i$ index refers to the tracks located in $r_{i-1/2} < h < r_{i+1/2}$. The $(i,j)$–th cell is also a cell array containing three elements:

- the weight $w_k$ of the track,
- the product $w_k \cos \phi$,
- a 1D array of dimension $I - i + 1$ containing the track segments.

The $\ell_1$ segment length of track ② is

```plaintext
l2=track{1,2}{3}(1)
```

leading to $l_2 = 0.07186585$.

Any length `track{ik,il}{3}(1)` corresponding to a segment that cross the $h$ axis must be multiplied by 2.
The following script will perform a numerical integration of $p_{11}$ in a 1D cylindrical tube of radius $R \equiv \text{rad}$ and macroscopic total cross section $\Sigma \equiv \text{sigt}$. The formula is

$$p_{11} = \frac{2}{V_1} \int_0^R dh \, D_1 = \frac{1}{V_1} \sum_{k=1}^K \omega_k \text{di}_f(\Sigma, \ell_k)$$

function f=p11_cyl(rad,sigt)
% compute the p11 component in 1D cylindrical geometry
track=sybt1d(rad,false,5) ;
val=0 ;
for i=1:5
    segment=2*track{1,i}{3}(1) ;
    if sigt ~= 0
        d=segment/sigt-(akin(3,0)-akin(3,sigt*segment))/sigt^2 ;
    else
        d=pi*segment^2/4 ;
    end
    val=val+track{1,i}{1}*d ;
end
f=val/(pi*rad^2) ;
Boundary conditions

The external boundary condition of a 1D cylindrical geometry is introduced by choosing an albedo $\beta^+$ set to zero for representing a voided boundary, or set to one for representing reflection of particles.

The voided boundary condition is similar to the vacuum condition used for 1D slab geometry.

The reflecting boundary condition is implemented in the context of the Wigner-Seitz approximation for lattice calculations in reactor physics. The Wigner-Seitz approximation consists of replacing the exact boundary with an equivalent cylindrical boundary, taking care to conserve the amount of moderator present in the cell. The correct condition in this case is the white boundary condition in which the particles are reflected with an isotropic angular distribution.
Boundary conditions

The application of a white boundary condition is based on the transformation of the reduced collision probability matrix $\mathbb{P} = \{p_{ij} \, , \, i = 1, I \text{ and } j = 1, I\}$ corresponding to a vacuum boundary condition. We first compute the escape probability vector $P_{iS} = \{P_{iS} \, , \, i = 1, I\}$, a column vector whose components are obtained from

\begin{equation}
P_{iS} = 1 - \sum_{j=1}^{I} p_{ij} \Sigma_j
\end{equation}

representing the probability for a neutron born in region $i$ to escape from the unit cell without collision.

The probability for a neutron entering in the unit cell by its boundary, with an isotropic angular distribution, to first collide in region $i$ is represented by the component $P_{Si}$. It is possible to show that

\begin{equation}
p_{Si} = \frac{P_{Si}}{\Sigma_i} = \frac{4V_i}{S} P_{iS}
\end{equation}

where $S$ is the surface of the boundary.
Boundary conditions

The transmission probability $P_{SS}$ is therefore given as

\[
P_{SS} = 1 - \sum_{i=1}^{I} p_{Si} \Sigma_i .
\]  

(18)

The reduced collision probability matrix $\tilde{P} = \{\tilde{p}_{ij} , i = 1, I \text{ and } j = 1, I\}$ corresponding to a white boundary condition can now be obtained using

\[
\tilde{P} = P + P_{iS} \left[ \beta^+ + (\beta^+)^2 P_{SS} + (\beta^+)^3 P_{SS}^2 + (\beta^+)^4 P_{SS}^3 + \ldots \right] p_{Sj}^T
\]

or, using a geometric progression, as

\[
\tilde{P} = P + \frac{\beta^+}{1 - \beta^+ P_{SS}} P_{iS} p_{Sj}^T.
\]  

(19)

The collision probability matrix $\tilde{P}$ can be scattering-reduced and used in the matrix flux equation of the CP method.
Spherical 1D geometry

- Computing the tracking and collision probabilities in 1D spherical geometry is similar to the cylindrical geometry case.

- The tracking is obtained by the `sybt1d` Matlab script

The expression of reduced collision probabilities is written

\[ p_{ij} = \frac{1}{V_i} \int_{V_i} d^3r' \int_{L_j} ds e^{-\tau(s)} \]
Spherical 1D geometry

which can be rewritten as

\[ p_{ij} = \frac{2\pi}{V_i} \left\{ \int_0^{r_i - 1/2} dh \int_0^{\ell_i} d\ell' \int_0^{\ell_j} d\ell \left[ e^{-[(\tau_{ij} + \tau_{ii} + \Sigma_i \ell_i) + \Sigma_i \ell' + \Sigma_j \ell]} \right. \right. \]

\[ + \left. e^{-[(\tau_{ij} + \Sigma_i \ell' + \Sigma_j \ell)]} \right\} \]

(21)

\[ + \int_{r_i - 1/2}^{r_i + 1/2} dh \int_0^{\ell_i} d\ell' \int_0^{\ell_j} d\ell e^{-[(\tau_{ij} + \Sigma_i \ell' + \Sigma_j \ell)]} \}

if \( i < j \)

where \( r_{i \pm 1/2} \) are the radii bounding region \( i \) and

\[ p_{ii} = \frac{2\pi}{V_i} \left\{ \int_0^{r_i - 1/2} dh \int_0^{\ell_i} d\ell' \left[ 2 \int_{\ell'}^{\ell_i} d\ell e^{-[\Sigma_i (\ell - \ell')] + \int_0^{\ell_i} d\ell e^{-[\tau_{ii} + \Sigma_i (\ell' + \ell)]} \right] \}

(22)

\[ + \int_{r_i - 1/2}^{r_i + 1/2} dh \int_0^{\ell_i} d\ell' \int_{\ell'}^{\ell_i} d\ell e^{-[\Sigma_i (\ell - \ell')] \} \}

where the first and second terms are the contributions from tracks (1) and (2), respectively.
These equations can be simplified by introducing the following definitions:

\begin{equation}
C_{ij}(\tau_0) = \int_0^{\ell_i} d\ell' \int_0^{\ell_j} d\ell' e^{-(\tau_0 + \Sigma_i \ell' + \Sigma_j \ell)}
\end{equation}

and

\begin{equation}
D_i = \int_0^{\ell_i} d\ell' \int_{\ell_i}^{\ell_i} d\ell' e^{-\Sigma_i (\ell - \ell')}
\end{equation}

so that

\begin{equation}
p_{ij} = \frac{2\pi}{V_i} \left\{ \int_0^{r_i-1/2} dh \ h \left[ C_{ij}(\tau_{ij}) + \tau_{ii} + \Sigma_i \ell_i \right] + C_{ij}(\tau_{ij}) \right\} + \int_{r_i-1/2}^{r_i+1/2} dh \ h \ C_{ij}(\tau_{ij})
\end{equation}

if \( i < j \) and

\begin{equation}
p_{ii} = \frac{2\pi}{V_i} \left\{ \int_0^{r_i-1/2} dh \ h \left[ 2D_i + C_{ii}(\tau_{ii}) \right] + \int_{r_i-1/2}^{r_i+1/2} dh \ h \ D_i \right\}.
\end{equation}
Integration in $\ell$ and $\ell'$ is next performed, leading to the following relations:

**a) $\Sigma_i \neq 0$ and $\Sigma_j \neq 0$ :**

$$C_{ij}(\tau_0) = \frac{1}{\Sigma_i \Sigma_j} \left[ e^{-\tau_0} - e^{-\left(\tau_0 + \Sigma_i \ell_i \right)} - e^{-\left(\tau_0 + \Sigma_j \ell_j \right)} + e^{-\left(\tau_0 + \Sigma_i \ell_i + \Sigma_j \ell_j \right)} \right]$$

**b) $\Sigma_i = 0$ and $\Sigma_j \neq 0$ :**

$$C_{ij}(\tau_0) = \frac{\ell_i}{\Sigma_j} \left[ e^{-\tau_0} - e^{-\left(\tau_0 + \Sigma_j \ell_j \right)} \right]$$

**c) $\Sigma_i \neq 0$ and $\Sigma_j = 0$ :**

$$C_{ij}(\tau_0) = \frac{\ell_j}{\Sigma_i} \left[ e^{-\tau_0} - e^{-\left(\tau_0 + \Sigma_i \ell_i \right)} \right]$$

**d) $\Sigma_i = \Sigma_j = 0$ :**

$$C_{ij}(\tau_0) = \ell_i \ell_j e^{-\tau_0}$$
e) $\Sigma_i \neq 0$:

$$D_i = \frac{\ell_i}{\Sigma_i} - \frac{1}{\Sigma_i^2} \left[ 1 - e^{-\Sigma_i \ell_i} \right]$$

f) $\Sigma_i = 0$:

$$D_i = \frac{\ell_i^2}{2}.$$