Monte Carlo Methods and Simulations in Nuclear Technology

Geometry and Model Representation, Tracking in Phase Space

Jan Dufek

2023

KTH Royal Institute of Technology

Contents

Geometry and Model Representation

Geometry and Model Representation

Precise geometry representation

In Monte Carlo neutron transport calculations (unlike in deterministic calculations), the physical model can be represented as accurately as needed.

Surfaces

Typically, the model is subdivided into elementary space regions (cells) that are bounded by surfaces of various orders. Various surfaces can be described by the surface equation

$$f(x, y, z) = 0$$

Space regions

In most MC codes, space regions are associated with positive or negative surface senses (sides). When a region is associated with a negative surface sense then

$$f(x, y, z) < 0$$

holds for all points (x, y, z) in this region. In case of closed surfaces, like spheres, the inside volume has a negative sense to the surfaces.

Similarly,

$$f(x, y, z) > 0$$

denotes a region associated with a positive surface senses.

Equations for basic surfaces

Surface type	Equation	Parameter
Plane	$ax + by + cz + d = 0$	**a, b, c, d**
Sphere	$(x-a)^2 + (y-b)^2 + (z-c)^2 - R^2 = 0$	**a, b, c, R**
Cylinder **Parallel to x-axis** **Parallel to y-axis** **Parallel to z-axis**	 $(y-b)^2 + (z-c)^2 - R^2 = 0$ $(x-a)^2 + (z-c)^2 - R^2 = 0$ $(x-a)^2 + (y-b)^2 - R^2 = 0$	 **b, c, R** **a, c, R** **a, b, R**
Cone **Parallel to x-axis** **Parallel to y-axis** **Parallel to z-axis**	 $\sqrt{(y-b)^2 + (z-c)^2} - t(x-a) = 0$ $\sqrt{(x-a)^2 + (z-c)^2} - t(y-b) = 0$ $\sqrt{(x-a)^2 + (y-b)^2} - t(z-c) = 0$	**a, b, c, t**
General ellipsoid, hyperboloid, paraboloid	$b(x-a)^2 + d(y-c)^2 + e(z-f)^2 +$ $2g(x-h) + 2i(y-j) + 2k(z-l) + m = 0$	**a, b, c, d, e, f, g, h, i, j, k, l, m**
Torus (elliptic & circular) **Parallel to x-axis** **Parallel to y-axis** **Parallel to z-axis**	 $\frac{(x-a)^2}{B^2} + \frac{\left(\sqrt{(y-b)^2 + (z-c)^2} - A\right)^2}{C^2} - 1 = 0$ $\frac{(y-b)^2}{B^2} + \frac{\left(\sqrt{(x-a)^2 + (z-c)^2} - A\right)^2}{C^2} - 1 = 0$ $\frac{(z-c)^2}{B^2} + \frac{\left(\sqrt{(x-a)^2 + (y-b)^2} - A\right)^2}{C^2} - 1 = 0$	**a, b, c, A, B, C**

Geometry and Model Representation

Physical model

The physical model is then created composed of cells that are created by "intersection" and "union" operations on regions associated with positive and negative sides of defined surfaces.

Identification of the cell of the collision

- The code has to evaluate the side (sense) of all surfaces on what the collision takes place.
- Then the code has to go through a list of cells and find the cell that shares the same combination of surface sides.

Note

Although a lot of optimisation is possible here, the conventional MC codes spend most of CPU time with this procedure!

Geometry and Model Representation

Various boundary conditions can be applied on the model

- void: particles crossing the surface will be killed if $\vec{n} \cdot \vec{\Omega} > 0$, where $\vec{n}$ is the normal vector to the surface at the crossing point,
- reflective: particle direction will be changed from $\vec{\Omega}$ to $\vec{\Omega}'$ so that the equation $\vec{n} \cdot \vec{\Omega} = -\vec{n} \cdot \vec{\Omega}'$ is satisfied and $\vec{\Omega}$, $\vec{\Omega}'$ and $\vec{n}$ can all be parallel to one plane,
- albedo: a fraction, α, of particles is reflected, the rest leaks out (get terminated).

Conventional Tracking in Phase Space

Conventional Tracking in Phase Space

Conventional tracking

In order to sample the neutron history, we need to know the material properties (macroscopic cross section) at the collision points. Remember, the distance between collisions is sampled as

$$l = -\frac{1}{\Sigma_t(E)} \ln(\xi)$$

where $\Sigma_t(E)$ is different for different cells. Therefore we need to know what space region (cell) the collisions takes place in.

Distance to the nearest surface

In the conventional sampling, the code has to calculate **the distance to the nearest surface** in the direction of flight in order to decide whether $\Sigma_t(E)$ has changed during the neutron flight. (If it did then the neutron must be moved to the boundary and the distance to next collision must be recalculated with the new cell material.)

Conventional Tracking in Phase Space

Distance to the surface

To determine the distance of a particle to a surface, we need to find the smallest positive root of the equation

$$f(x_0 + s\omega_x, y_0 + s\omega_y, z_0 + s\omega_z) = 0$$

where s is the distance to the surface, (x_0, y_0, z_0) is the particle position, $(\omega_x, \omega_y, \omega_z) = \vec{\Omega}$, and $f(x, y, z) = 0$ is satisfied for all (x, y, z) on the surface.

Note

In complex geometries this procedure needs to be done several times to find the next collision point, which is expensive in terms of CPU time.

Delta Tracking in Phase Space

Delta Tracking in Phase Space

Delta tracking

The conventional tracking is very complicate to implement in the MC codes and very CPU demanding during the simulations. Some MC codes solve this problem via the "Delta tracking".

Virtual collision

To describe the delta tracking technique, we need to define the so-called "virtual collision". The virtual collision is not a real collision; it is a collision that does not change the direction or energy or weight. It is associated with the virtual macroscopic collision cross section Σ_v that characterizes the virtual collision probability per path length traveled in the medium.

Delta Tracking in Phase Space

Note

Obviously, we can increase the real total macroscopic cross section by an arbitrary virtual cross sections, and it will not affect the particle transport simulations in any way. (Many additional virtual collisions may be simulated, but these will not affect the particles at all.)

Idea of the delta tracking

If the total macroscopic cross section was equal in all cells then we would not need keep track on the distance to the surface and on the cell number since the kernel would be sampled from the same distribution. Only one cross section would be used to sample the transition kernel. Well, we can do this - we can find what cell has the largest mac. tot. cross section, and we can increase cross sections in other cells using the virtual cross section to match the largest one.

Delta Tracking in Phase Space

Advantages of the delta tracking

- Considerable simplification of the MC code
- FOM of the calculation is substantially increased (i.e., the efficiency is improved) if the system doesn't contain strong absorbers

Disadvantages of the delta tracking

- FOM of the calculation may be decreased if the system contains small cells with strong absorbers. This is because a large number of virtual collisions will be simulated in a majority of cells, taking too much of CPU time. (At each virtual collision, the code still needs to figure out what material the cell contains and load the cross sections corresponding to the neutron energy.)
- Surface flux estimators cannot be implemented easily.