NE 155/255, Fall 2019
Solution Approaches
September 25, 2019

Last class we talked about the reactor problem types and physics involved, as well as how that necessarily impacts the solution strategies we use. Today we'll talk about approaches and what kinds of methods we use when.

Computational Methods

Inside the box of computational methods, we have two main categories:

- **Deterministic**: discretize all independent variables, obtain a set of coupled, linear, algebraic equations and develop a numerical method to solve them.
- **Probabilistic**: follow the history of each relevant particle based on the underlying probabilities for various types of interactions.

Everything we've done so far applies to both methods (we've only simplified the equation). Each method has its own benefits and challenges. We'll talk a bit about **deterministic** methods first. Here, the discretization methods are a big part of the strategy, followed by parallelization.

The size of the problems (which impacts memory and parallelization performance) is governed by discretization. The quality and behavior of solutions is also governed by discretization. To get more accurate fluxes, typical transport problems today are three-dimensional, have up to thousands $\times$ thousands $\times$ thousands of mesh points, use up to $\sim$ 150 energy groups, include accurate expansions of scattering terms, and are solved over many directions.

Table 1: Meaning and Range of Indices Used in Transport Discretization

Variable	Symbol	First	Last	Low	High
Energy	g	0 or 1	$G-1$ or G	2	100s
Solid Angle	a	1	n	S_2	S_{16}
Space	i, j, k	0,0,0	I, J, K	1?	1e9
Legendre moment (P_N)	ℓ	0	N	0	9

E.g. In 2012, a Pressurized Water Reactor (PWR)-900 with 44 groups, a 578 $\times$ 578 $\times$ 700 mesh, using S_{16} level-symmetric quadrature, and with P_0 scattering was solved: 1.7 trillion unknowns.

With **Monte Carlo** methods, we sample the physics of every single interaction of every single particle until we have enough samples to assert that something is statistically valid. This requires lots and lots of samples, but is typically fairly easy to parallelize. Monte Carlo methods require strategies and rules for sampling the physics, which can be complicated.

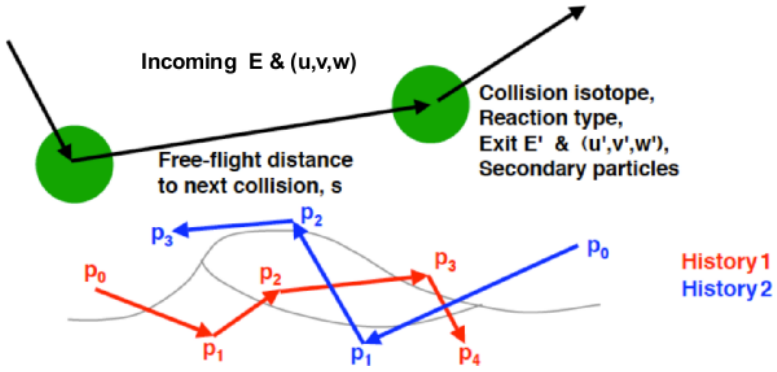

Figure 1: Monte Carlo particle tracking

Comparison

In order of increasing accuracy *and* increasing runtime:

1. Diffusion Theory
    - discretized and homogenized space
    - linearly anisotropic direction
    - discretized energy (few-group)
2. Deterministic
    - discretized space
    - discretized direction (discrete ordinates [S_N]) or functional expansion of direction (spherical harmonics)
    - discretized energy (multi-group)
3. Monte Carlo
    - continuous spatial resolution
    - continuous direction representation
    - continuous energy representation

	Monte Carlo	Deterministic
Strengths	* General geometry * Continuous energy * Continuous in engle * Inherently 3-D * Easy to parallelize on CPUs	* Fast * Global solution * Solution is of same quality everywhere * Inputs can be relatively simple
Weaknesses	* Slow * Might be memory intensive * Solutions have statistical error * Local solutions only * Must adequately sample phase space * Need efficient variance reduction (VR) * Input can be extremely complicated	* Discretization governs solution quality * Might be memory intensive * Solution contains truncation error * Constrained by what you can mesh * Ray effects * Can be complicated to parallelize

We tend to think of Monte Carlo methods as "benchmark quality" and highly flexible.

What is one major hazard of Monte Carlo methods?

What is the biggest challenge of using MC methods for detailed design?