Monte Carlo Methods and Simulations in Nuclear Technology

Introduction

Jan Dufek

2022

KTH Royal Institute of Technology

Contents

Introduction

Course structure

Group project topics

Introduction

What is the Monte Carlo method?

What is the Monte Carlo method?

- The Monte Carlo method is a way of solving a deterministic problem by a stochastic approach using a random number generator. The Monte Carlo method is commonly applied on deterministic problems that are difficult to solve by deterministic methods.
- A number *n* of independent observations (e.g. neutron histories) are collected during the calculation.
- Results are derived from the averaged observation.

Advantages and disadvantages of the Monte Carlo method

Advantages

- Simplicity of the method (e.g. the transport equation does not have to be formulated to find the neutron flux in the reactor).
- Applicable on complex problems without simplifications (e.g. accurate geometry, continuous-energy cross sections).
- Precision of results can be easily improved by collecting more samples.
- Efficiency in complex problems (the statistical error generally decays as $1/\sqrt{n}$).
- Can be parallelised effectively.

Disadvantages

- Computational cost (depending on the complexity of the problem) for non-integral quantities (like the power distribution).
- Results with statistical errors (not necessarily a disadvantage though).

Application of the Monte Carlo method

Application in physics

Mainly for simulation of transport of various particles, like neutrons, photons, electrons and other charged particles.

Application in other fields

The method is nowadays also used in mathematics (e.g.multidimensional integrals), chemistry (e.g. gas kinetics), economics (analysis of stock market), biology (e.g. genetic research) and other fields.

Application of the Monte Carlo method

Application in nuclear engineering

- In research and development (in industry and academia), for:
  - shielding calculations,
  - criticality calculations (multiplication factor, neutron flux, reaction rates, group cross sections, etc.),
  - depletion calculations (simulation of fuel cycles).
- Reference solutions for benchmarks of deterministic codes.

Major Monte Carlo codes for reactor physics calculations

- SERPENT (VTT, Finland) - used in this course
- OpenMC (MIT, USA)
- MCNP - Monte Carlo N-Particle Transport Code (Los Alamos National Laboratory, USA)
- TRIPOLI (CEA, France)
- KENO (Oak Ridge National Laboratory, USA)

Course Objectives

Important remark

The course topics focus on **principle problems of Monte Carlo calculations**, and not on topics that are covered by code manuals (creating the input file, etc).

After completion of the course, you will be able to:

- By the Monte Carlo method (code) model accurately the geometry of nuclear reactors and carry out shielding, criticality and depletion calculations.
- Optimise free parameters in criticality calculations to achieve a good source convergence and computing efficiency.
- Optimise free parameters in depletion calculations to ensure computing stability and efficiency.
- Describe specific variance reduction techniques used in criticality and shielding problems.
- Choose an appropriate parallel-computing scheme for best efficiency.

Course structure

Course structure

Course activities

- 11 flipped-classroom lectures.
- 8 home assignments, 3 points max for each.
- 9 quizzes integrated in Canvas, 3 points max for each.
- Group project (groups by two students):
  - Project presentation, 4 points max for each student.
  - Project report (written by two students), 15 points max for each student.
- Oral exam, 30 points max.

Grade

Total score	Grade
> 90	A
81-90	B
71-80	C
61-70	D
51-60	E
41-50	Fx
< 41	F

Course structure

What does the flipped classroom mean?

- That means that you'll primarily learn about the topics outside the classroom from short video tutorials (on YouTube) and you will learn how to use Serpent from the code manual.
- You have to prepare for each lecture beforehand.

What shall we do in the classroom?

- 1st hour:
  - A very brief summary of the topic (since all my lectures are on YouTube, I will not repeat them in the classroom).
  - I will be available for answering your questions that you may have to the topics that you learned.
  - We will discuss the solution to the previous home assignment (a randomly selected student presents the solution).
  - A new assignment will be given with a deadline for the next lecture.
- 2nd hour:
  - Canvas quiz (10 minutes) comes right after the break.
  - We will then discuss progress on your group projects.

Course structure: Lecture topics

Date	Topic	Study videos
Nov 3	1. **Introduction**	-
Nov 11	2. **Fundamentals of prob. theory and stat.**	1.2-1.5
Nov 17	3. **RNG, sampling procedures, simple sampling**	2.1-2.6
Nov 24	4. **General variance reduction techniques**	3.1-3.5
Dec 8	5. **Analog Monte Carlo criticality simulations**	4.1-4.7
Dec 15	6. **Introduction to SERPENT and XS Libraries**	5.1-5.9 + 5.X
Jan 19	7. **Convergence of the Fission Source**	6.1-6.3
Jan 26	8. **Monte Carlo Burnup Calculations**	7.1-7.5
Feb 2	9. **Geometry and Model Representation, Tracking**	8.1-8.4
Feb 9	10. **Non-Analog Monte Carlo Simulations**	9.1-9.5
Feb 16	11. **Variance Red. Techniques and Parallel Sim.**	X.1-X.4
March 2	12. **Group Project Presentations**	

Before each lecture, study the YouTube videos with the numbers in the right column.

Course structure

Study materials

- Youtube channel with voice commented video slides
- A Monte Carlo Primer (textbook, downloadable via Canvas link)
- Serpent manual
- Few more books on Canvas homepage downloadable via KTH library.

Home assignment reports

Format of HA reports

- Upload the report in the pdf format in landscape page-orientation (projector-friendly).
- Required content:
    - Cover page with title, name, date.
    - Description of the task.
    - Methods (brief description of methods employed).
    - Results.
    - Discussion of results and conclusions.
- Be ready to present your results upon request during the lecture.

Grading rules (see a detail description in Canvas Files/grading rules)

- 1pt max for **results and discussion**. (You get a point when the results are reasonable and commented.)
- 1pt max for **content** (above).
- 1pt max for **style** (see the file "Acronyms for errors in technical reports").
- Submissions after deadline get 50% of the above points. One revision possible, but it is graded as a late submission then.

Quizzes

Quizzes

- The quiz will be always about the new topics, so you have to prepare in advance from the pre-recorded lectures.
- The quiz will be closed 10 minutes after it starts. It will start at the beginning of the second hour.
- Correct solutions and your grade will be available as soon as the quiz closes.
- You can't re-take the quiz if you miss it (since the solutions are published).
- You can still take the quiz even when you can't attend the lecture.
- If you have questions about the topics of the lecture then sort them out with me before the quiz starts.

Group projects

Group projects rules

- Each project group has two students, randomly chosen by Canvas.
- Each group chooses or gets a project topic early at the beginning of the course.
- You need to discuss the progress of the work with me during the course.
- You'll give a project presentation at the end of the course.
- Don't forget to upload your presentation to Canvas.
- Only students who actively participate in the presentations will collect the points.

Group projects

Group project report rules

- The final project report has the usual format of a technical report, written in Word or Latex, saved as pdf.
- Preferably format the text into two columns in a form of a journal or conference paper.
- The report must NOT exceed 5 pages!
- The report can be submitted at any time during the course, with the possibility of one revision before final grading.

The report will be graded according to rules specified in the file "Report grading rules.pdf" in Canvas Files/grading rules.

Group project topics

A) Monte Carlo simulation of a xenon oscillation in PWR

Purpose

The purpose is to produce of a series of animated and regular plots (1-D and 2-D) of xenon oscillations in PWR for demonstrations in the reactor physics course.

Process

- Create a 3-D full core model of a very large PWR in Serpent. The larger the core is the easier it will be to demonstrate the xenon oscillations. Set the diameter of the active core to e.g. 4 m or more.
- Add a thick reflector around the core.
- In the model, each fuel assembly must have independently defined fuel material, so that the xenon and iodine concentration can develop uniquely in each assembly according to the local flux. Define a single fuel material in each assembly (do not model the axial variation).
- Set up a series of quick test burnup simulations to debug the model and confirm the presence of xenon oscillations. (Make sure that you **do not** use equilibrium xenon option in the burnup simulations otherwise you couldn't simulate the xenon oscillations.)

A) Monte Carlo simulation of a xenon oscillation in PWR - cont.

Process - cont.

- Set up a final burnup simulation with very short time steps (e.g. 10 minutes), the total time period of few weeks, and fairly good statistics per time step (to eliminate as much noise from the plots as possible).
- Generate the following plots from your results:
  - 2-D regular plots: neutron flux (power) distribution at a number of time steps. (Other plots for xenon concentrations.)
  - 2-D animated plot (video): neutron flux (power) distribution oscillating over the full time period (few weeks). Generate similar plots for xenon concentration.

B) Burnup simulations with OpenMC code

Purpose

The burnup mode has been recently added to the OpenMC code. The purpose of this project is to demonstrate the functionality and options of this mode.

Process

- Download the OpenMC code (open source) and install it on the Serpent server.
- Study the code manual and convert some of the example Serpent files for use with OpenMC.
- Adapt the OpenMC and Serpent input files for burnup, and run burnup simulations with both codes. Use the same cross section library version with both codes.
- Test various burnup options in the OpenMC code, such as the choice of the coupling scheme.
- Compare results between the codes. Plot keff over time, and also plot the concentration of some nuclides (e.g. ^{239}Pu) over time.

C) Development of a Monte Carlo solver

Purpose

The purpose of this project is to develop a simplified Monte Carlo solver for criticality simulations.

Process

- Program a criticality solver with these possible simplifications:
  - Use the tabulated energy-dependent cross sections from the Janis database.
  - You can hardwired a simple 3-D geometry of the model in the code.
  - Use delta tracking for geometry treatment.
  - Assume isotropic scattering of neutrons in the center-of-mass collision system.
- You can use any programming language.
- Create a model of a fissile system with some interesting geometry and calculate the neutron flux distribution and keff of the system, and compare the results to Serpent results.

D) Monte Carlo analysis of higher eigenmodes

Purpose

The purpose of this project is to generate plots of several neutron flux higher eigenfuncions for demonstrations in the reactor physics course.

Process

- In Serpent, create several input files describing very simple systems, such as a large fissile cube, fissile sphere and fissile cylinder with void boundary conditions.
- In the input files, include a card that will instruct Serpent to tally a so-called “fission matrix” of the system. Use sufficiently fine spatial mesh (superimposed over the system) for tallying the fission matrix.
- Using available numerical solvers, such as Matlab or Sage, compute the fundamental and several higher eigenfunctions of the fission matrix (for each system).
- Generate plots of the eigenfunctions in a series of 2-D and 3-D plots for each system.

E) Coupled steady-state Monte Carlo simulations

Purpose

The purpose of this project is to calcule a steady-state power distribution in a thermal reactor with a thermal-hydraulic feedback.

Process

- In Serpent, create a full-core BWR model with void boundary conditions.
- Implement the model in such a way that each fuel assembly can have independently defined average density of its coolant. Assume the simplification that the coolant has uniform temperature and density (axially and radially) within each fuel assembly.
- Write a computer script that will iterate the steady-state conditions. At each step of the iteration, the script will:
  - run the Serpent criticality calculation, and obtain the power of each FA,
  - apply a relaxation scheme with Robbins-Monro algorithm on the obtained power distribution,
  - compute the average coolant density of each fuel assembly corresponding to the relaxed power of the assembly,
  - rewrite the coolant densities in the input file by the new values.
- Run the coupled simulation, and compute the steady-state power.
- Plot the obtained power distribution and compare it to a power distribution in a reactor with a uniform coolant density.

F) Training of artificial neural networks with Monte Carlo simulations

Purpose

The purpose of this project is to develop an artificial neural network that will predict k_∞ for a fuel pin.

Process

- In Serpent, create a simple model of a fuel pin cell with reflected boundary conditions.
- Run a series of simulations, and compute k_∞ of the fuel pin cell with various combinations of free parameters, such as:
  - pin pitch,
  - fuel pellet radius,
  - gap thickness,
  - cladding thickness,
  - fuel temperature,
  - coolant density,
  - coolant temperature,
  - enrichment in ^{235}U
- Design a neural network and train it on the pre-computed data. (Make sure that the random noise in data is small enough.)
- Test the accuracy of the network, and tune the design of the network for best accuracy.

Other possible projects

Other possible projects

- G) Study of the Shannon-entropy source-convergence criterion.
- H) ANN-based Fission Matrix simulations.
- I) MC calculation of a control rod (CR) calibration curve with respect to CR depletion.
- J) Optimisation of free parameters for criticality calculations of a full-core PWR model.
- K) Optimisation of free parameters for burnup calculations of a full-core PWR.