NENG 685 - Fall 2017
Methods for Neutral Particle Transport

Instructor: Capt James Bevins
Office: Bldg 640, Rm 331A
E-mail: jbevins1@afit.edu
Office phone: (937) 255-3636 x4767

Class: MW, 0800-1000, Bldg 640, Rm 222
Office hours: T/Th, 0900-1000 & by appt

Course Description

This course covers the principal methods used for deterministically solving the Boltzmann transport equation for neutral particles (neutrons and photons). This course presents the fundamental mathematical and computational methods using discretizations in space, energy, and angle. Iterative methods for efficient solution of the transport problems are explored and analyzed. Monte Carlo and Discrete Ordinates methods are explicitly developed and applied to shielding and criticality problems of interest. The course will include both code development and use of existing codes for solving criticality and shielding problems of interest in nuclear engineering.

Requisites

NENG 651 - Introduction to Nuclear Physics

Textbooks

FORTRAN 95/2003 For Scientists and Engineers, 3d Ed, Steven Chapman, McGraw-Hill, 2008.
A Primer on Scientific Programming with Python, Hans Petter Langtangen, 3rd Ed., Springer, 2012.
Effective Computation in Physics, Anthony Scopatz and Kathryn Huff, O'Reilly, 2015.

Additional References

- Course notes and handouts: `https://github.com/jamesbevins/NENG685`
- Free programming lessons: `https://software-carpentry.org/lessons/`
- Reproducible Research: `https://www.practicereproducibleresearch.org/`
- Helpful guides: `https://github.com/jamesbevins/SurvivingAFIT`
- Jupyter (the awesome thing formerly known as iPython): http://jupyter.org/
- KAERI: http://atom.kaeri.re.kr/
- U.S. Nuclear Data http://www.nndc.bnl.gov/
- E. E. Lewis and W. E. Miller Jr., "Computational Methods of Neutron Transport," J. Wiley & Sons, New York (1993).
- J. J. Duderstadt and L. J. Hamilton, "Nuclear Reactor Analysis," Wiley (1976)

Course Objectives

At the end of the course, students should:

- Understand program design, structure, and procedures
- Develop code to solve problems of interest in nuclear engineering
- Be able to numerically solve ODEs, PDEs, and systems of equations
- Be able to perform numerical integration and differentiation
- Understand the different methods of neutron transport and their associated benefits and drawbacks
- Understand Boltzmann Transport Equation, Diffusion Equation, and the methods used to solve them
- Understand Monte Carlo transport techniques and associated concepts
- Be familiar with deterministic transport (PENTRAN) and stochastic transport (MCNP) codes
- Be comfortable operating in the remote high performance computing environments available at AFIT

Grading

Grading will consist of a variety of metrics: homeworks, ”pre-flight” quizzes, and a final project. The grading structure reflects that programming and algorithm development can only be achieved through practice. The pre-flight quizzes are meant to enhance learning by instilling key foundational ideas prior to use in class.

- Homework: 45%[1]
- Pre-flight Quizzes: 25%[2]
- Final Project (code, paper, presentation): 30%

It is not anticipated that there will be a curve applied. It is my belief that the grades will sort themselves out if the course assessment material is properly done. I have no set distribution of grades and have no reservation about everyone earning “As” or any other grade on the spectrum. Rarely are there enough statistics (i.e. students) in this class to try to impose a normal distribution on the grades and in the AF (and grad school) 90% of the people are above average.

Extra points - specified in each assignment - may be available for following ”best practices” that are above and beyond what is required to complete the assignment.

Programming Language

My philosophy is that it is best to know one language really well because different organizations, communities, projects, etc. will require different languages, and it is impossible and often counterproductive to learn them all. If you know one language well, you can pick up other languages when the need arises.

Consistent with this philosophy, this course will primarily be taught using Python. The goal is to get you proficient in *a language* by the end of the course. However, you are free to use whatever language you feel comfortable with and works for your workflow. The languages I will be able to help with in order of decreasing knowledge and/or increasing time since last use: Python, Matlab, FORTRAN, C++, and Java. There are other resources in the department, online, and in my library for the various languages as well that you are free to use.

[1] Late submissions: -20% for each day it is late with a maximum of -60%.

[2] Pre-flight quizzes are based on the assigned reading/material, and are due by 2359 the day before the class they are due.

Academic Integrity

Academic Integrity - "Uncompromising adherence to a code of ethics, morality, conduct, scholarship, and other values related to academic activity." (AUI 36-2309)

Academic Freedom - "Academic freedom must be tempered by good judgment to refrain from making offensive remarks, unfounded opinions, or irresponsible statements." (AUI 36-2308)

Non-Attribution - "All guest speakers, students, and permanent-party personnel are prohibited from divulging the identity of any particular speaker, whether a guest speaker, faculty member, or student, for the purpose of attributing to that speaker any specific remarks or statements, including but not limited to offensive remarks and irresponsible statements." (AUI 36-2308)

AFIT provides detailed guidance about these policies in the Student Handbook. Lack of knowledge of these policies is not a reasonable explanation for a violation. Questions related to course assignments and the academic honesty policy should be directed to me.

There are also several great publications on your responsibilities as a scientist/engineer and researcher. I'd recommend at least skimming:

- The National Academies Press's "On Being a Scientist"
- The National Academy of Sciences's "Fostering Integrity in Research"

My policy is that you may work together on homework, but you must **specifically cite with whom you worked and what you did together**.

Attendance

Course credit presumes the student has attended all class sessions in addition to completing all tasks. Thus, attendance is mandatory in all sessions to graduate. If students have an extraordinary reason to be absent (example: a sickness, car breakdown, etc.), the student must contact the instructor (in advance if possible) to evaluate if the absence will be excusable.

Course Schedule

Date	Assigned	Due	Topic	Reading	Calendar Notes
Mon 2 Oct		PF01	Intro to Comp Numerical Error	Scopatz Ch 2[3]	
Wed 4 Oct	HW01	PF02	Interp & Approx	Scopatz Ch 3-4[3]	
Fri 6 Oct	—	—	—	—	Last Day to Add
Mon 9 Oct	—	—	—	—	Columbus Day
Wed 11 Oct		HW01/PF03	Diff & Int	Scopatz Ch 5[3]	
Fri 13 Oct	—	—	—	—	Drop w/o Record
Mon 16 Oct		PF04	ODEs & PDEs	Scopatz Ch 6[3]	
Wed 18 Oct	HW02	PF05	Vectors, Matrices, & Sys of Eqns	Scopatz Ch 9[3]	
Mon 23 Oct			BTE	Course Notes	
Wed 24 Oct	HW03		Diffusion Eqn	Course Notes	
Wed 25 Oct		HW02			
Mon 30 Oct	FP[4]		DE Discretization	Course Notes	
Wed 1 Nov	HW04	HW03/PF06	DE Discretization	Scopaz Ch1	
Mon 6 Nov			PENTRAN Part I	PENTRAN Manual	Dr. Sjoden teaches
Wed 8 Nov			PENTRAN Part II	PENTRAN Manual	Dr. Sjoden teaches
Mon 13 Nov		HW04	Iterative Methods		
Wed 15 Nov	HW05	PF07	Runge-Kutta	PRKE Notes	
Mon 20 Nov				Monte Carlo	
Wed 22 Nov	HW06	HW05	Monte Carlo		
Fri 24 Nov	—	—	—	—	Drop w/o Grade
Mon 27 Nov		PF08	MCNP	TBD	
Wed 29 Nov	HW07	HW06	MCNP		
Mon 4 Dec			ADVANTG		
Wed 6 Dec		HW07	FP Presentation		15 Min Talk
Tue 12 Dec		FP Paper			

[3] Or equivalent for different language
[4] Final Project

Disclaimer: The course instructor retains the right to adjust the course material and grading process while the course is underway as necessary to maximize the learning experience and assess student learning appropriately.

Last updated: October 16, 2017