NE 155
Introduction to Numerical Simulations in Radiation Transport

Lecture 1: Introduction

R. N. Slaybaugh

January 18, 2017

Details

- Asst. Prof. Slaybaugh, 4173 Etcheverry Hall
  Email: slaybaugh@berkeley.edu
  Phone: 570-850-3385
- Office hours: F, 2:30 - 3:30 pm
- Prerequisites: Math 53 and 54 (Eng 7 rec)
- Prerequisite knowledge and skills:
  - Solve linear, first, and second order differential equations
  - Linear algebra, vector calculus
  - Computer language knowledge: Python, C, C++, Fortran, or MATLAB /Octave
- Class github page: https://github.com/rachelslaybaugh/NE155
- Many materials will also be on bCourses

References

- Course notes + handouts
- Resources "Page" on bCourses
- The Hacker Within: http://thehackerwithin.github.io/berkeley/
- Software Carpentry: http://software-carpentry.org/lessons.html
- Choose a Python Ebook that fits your needs: http://www.leettips.org/2013/02/top-10-free-python-pdf-ebooks-download.html

Grades and Lab

Grading

Homework	40%
Midterms (2)	15% + 15% = 30%
Final Project	30%
- Late submissions: -20% for each day it is late (max -60%)

Class computer lab accounts

- All students will get class computer lab accounts at Davis Etcheverry Computing Facility (DECF)
- DECF (1171 and 1111 Etcheverry): http://www.decf.berkeley.edu/
- We might use the Serpent Monte Carlo code (http://montecarlo.vtt.fi/)

Course Objectives

- Review systems of linear algebraic equations, linear algebra, eigenvalues and eigenvectors of a matrix, spectral radius of a matrix, numerical differentiation and integration, direct and iterative methods for solving linear systems.
- Introduce the numerical approaches used to solve fixed-source and criticality problems in analysis of neutron transport/diffusion in nuclear systems.
- Introduce solution methods for the point kinetics equation.
- Discuss the basic characteristics of deterministic and Monte Carlo approaches to numerical solution of these problems.

Relevant Courses

- E7 Introduction for Computer Programming for Scientists and Engineers
- CS4 Introduction to Computing for Engineers
- CS9A-H various languages for Programmers
- Math 128A, Numerical Analysis (solution of ordinary differential equations)
- Math 128B, Numerical Analysis (evaluations of eigenvalues and eigenvectors, solution of simple partial differential equations)

SCHEDULE + CAMPUS INFO

Let's look through the syllabus
The schedule is on bCourses under Files

Useful Campus Information:

- Mental health resources: http://www.uhs.berkeley.edu/students/counseling/cps.shtml
- Sexual assault support on campus: http://survivorsupport.berkeley.edu/

Code of Ethics

National Society of Professional Engineers: http://courses.cs.vt.edu/professionalism/WorldCodes/NSPE.code.html

Engineers, in the fulfillment of their professional duties, shall:

1. Hold paramount the safety, health, and welfare of the public in the performance of their professional duties.
2. Perform services only in the areas of their competence.
3. Issue public statements only in an objective and truthful manner.
4. Act in professional matters for each employer or client as faithful agents or trustees.
5. Avoid deceptive acts in the solicitation of professional employment.

Code of Ethics

American Nuclear Society: http://www.ans.org/about/coe/

Fundamental Principle

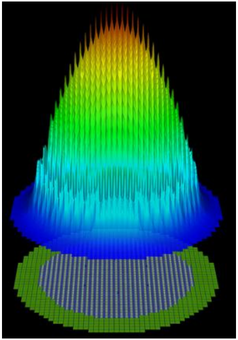
ANS members as professionals are dedicated to improving the understanding of nuclear science and technology, appropriate applications, and potential consequences of their use.

To that end, ANS members uphold and advance the integrity and honor of their professions by using their knowledge and skill for the enhancement of human welfare and the environment; being honest and impartial; serving with fidelity the public, their employers, and their clients; and striving to continuously improve the competence and prestige of their various professions.

Preview

$$[\hat{\Omega} \cdot \nabla + \Sigma(\vec{r}, E)]\psi(\vec{r}, \hat{\Omega}, E) = \chi(E) \int_0^\infty dE' \, \nu\Sigma_f(\vec{r}, E') \int_{4\pi} d\hat{\Omega}' \, \psi(\vec{r}, \hat{\Omega}', E') + \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}' \, \Sigma_s(\vec{r}, E' \to E, \hat{\Omega}' \cdot \hat{\Omega})\psi(\vec{r}, \hat{\Omega}', E')$$

Learn methods to translate equations like the *Boltzmann Transport equation* into computer-generated solutions



Categories

Experiment: Experimental scientists work by observing how nature behaves

Theory: Theoretical scientists use the language of mathematics to explain and predict the behavior of nature

Computation: Computational scientists use theoretical and experimental knowledge to create computer-based models of aspects of nature

Computational Science/Engineering

Computational Science seeks to gain understanding principally through the analysis of mathematical models on high performance computers.

The term computational scientists has been coined to describe scientists, engineers, and mathematicians who apply high performance computer technology in innovative and essential ways to advance the state of knowledge in their respective disciplines.

Thus, we distinguish it from computer science, which is the study of *computer and computation* and *theory and experiment*, the traditional form of science.

Solving Problems

1. Identify the problem
2. Pose the problem in terms of a mathematical model
3. Identify a computational method for solving the model
4. Implement the computational method on a computer
5. Assess the answer in the context of the
    - Implementation (computer language and architecture)
    - Method (discrete or continuous)
    - Model (symbolic or numerical)

    Using
    - Visualization and interpretation
    - Experimental comparisons
    - Analytical comparisons
    - Engineering judgement

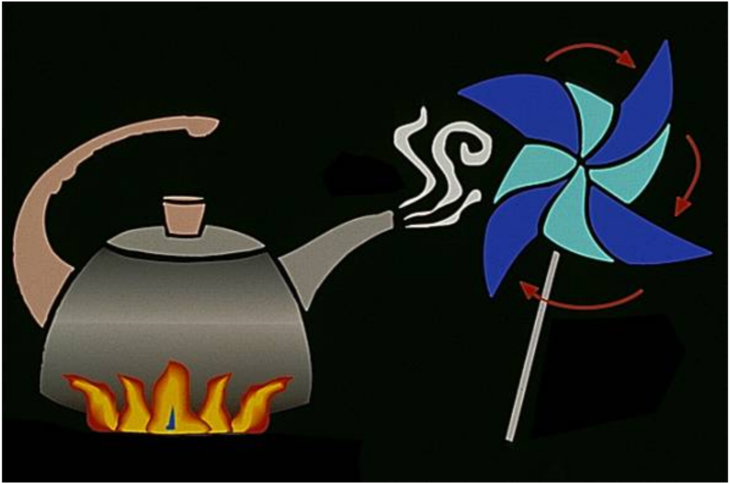
Big Challenges

- Science
  - Global climate modeling
  - Astrophysical modeling
  - Biology: genomics; protein folding; drug design
  - Computational Material Sciences and Nano-sciences
- Engineering
  - Semiconductor design
  - Earthquake and structural modeling
  - Computational fluid dynamics
  - Analysis and design of nuclear reactors
- Business
  - Financial and economic modeling
  - Transaction processing, web services and search engines
- Defense
  - Nuclear weapons – test by simulations
  - Cryptography

What are we trying to accomplish?

- The challenge of designing a nuclear reactor is to make it as **economical** as possible while ensuring its **safety**.
- This is not an easy task!
- The principle of a nuclear reactor is relatively simple:
  - Fission creates heat within the nuclear fuel,
  - The heat is conducted to the fuel cladding surface and to the coolant,
  - The heat is subsequently transported by a coolant through heat exchangers and ultimately to a steam conversion plant.

EASY?



What are we trying to accomplish?

- In order to design economical and safe reactors, one must choose among a vast range of competing designs:
  - What are the best fuels, structure, and coolant materials; what are their appropriate ratios?
  - How does the reactor respond to component failures?
  - How does one balance those choices given competing goals of performance, lifetime, safety, and capital cost?

- Ideally, one would like to base these choices on theory rather than experimental trial and error

- This is where computational science fits in...

Early Days

Before much computer use (e.g., 1943), things took a long time

Project	Est. Time
Density distribution in difficult system	2 weeks
Integral equation for absorption in Al slab	2 weeks
Slowing-down length in H_2O & related calcs	3 weeks
Albedo problems	1 week

Adolescence – Early 1980s

- NE was at the forefront of computer applications (!)
- Major early success story in the computational sciences:
  - Reduced the burden of experiment
  - Contributed greatly to reactor design
- However, modeling was severely constrained
  - Unable to explicitly model the key physical phenomena within a reactor
  - Low-dimensional representation
  - Lumped parameter models
  - Empirical correlations with tunable parameters established largely by experiments

ADOLESCENCE → TODAY'S CHALLENGES

- Computing limitations caused
  - Heavy reliance on expensive and often complicated experiments
  - Inaccuracy resulted in *significant design margins* → negative impact on plant economics
  - Exploration of novel reactor design concepts was greatly constrained

- Many codes developed then are still used

ADOLESCENCE → TODAY'S CHALLENGES

- Computing limitations caused
  - Heavy reliance on expensive and often complicated experiments
  - Inaccuracy resulted in *significant design margins* → negative impact on plant economics
  - Exploration of novel reactor design concepts was greatly constrained

- Many codes developed then are still used
- Can we update these tools?

Adolescence → Today's Challenges

- Computing limitations caused
  - Heavy reliance on expensive and often complicated experiments
  - Inaccuracy resulted in *significant design margins* → negative impact on plant economics
  - Exploration of novel reactor design concepts was greatly constrained

- Many codes developed then are still used
- Can we update these tools?
- Do we need to design new tools?

ADOLESCENCE → TODAY'S CHALLENGES

- Computing limitations caused
  - Heavy reliance on expensive and often complicated experiments
  - Inaccuracy resulted in *significant design margins* → negative impact on plant economics
  - Exploration of novel reactor design concepts was greatly constrained

- Many codes developed then are still used
- Can we update these tools?
- Do we need to design new tools?
- What methods will take us into the future?

ADOLESCENCE → TODAY'S CHALLENGES

- Computing limitations caused
    - Heavy reliance on expensive and often complicated experiments
    - Inaccuracy resulted in *significant design margins* → negative impact on plant economics
    - Exploration of novel reactor design concepts was greatly constrained

- Many codes developed then are still used
- Can we update these tools?
- Do we need to design new tools?
- What methods will take us into the future?
- What will the architectures look like?

ADOLESCENCE → TODAY'S CHALLENGES

- Computing limitations caused
  - Heavy reliance on expensive and often complicated experiments
  - Inaccuracy resulted in *significant design margins* → negative impact on plant economics
  - Exploration of novel reactor design concepts was greatly constrained

- Many codes developed then are still used
- Can we update these tools?
- Do we need to design new tools?
- What methods will take us into the future?
- What will the architectures look like?
- How do we successfully navigate that interplay?

Current State

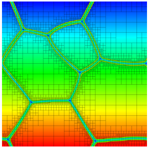
2010: the DOE announced *Oak Ridge National Laboratory* won the Nuclear Energy Modeling and Simulation Energy Innovation Hub (reawarded for 10 years), including:

- Electric Power Research Institute (EPRI), Palo Alto, CA
- Idaho National Laboratory, Idaho Falls, ID
- Los Alamos National Laboratory, Los Alamos, NM
- Massachusetts Institute of Technology, Cambridge, MA
- North Carolina State University, Raleigh, NC
- Sandia National Laboratories, Albuquerque, NM
- Tennessee Valley Authority, Knoxville, TN
- University of Michigan, Ann Arbor, MI
- Westinghouse Electric Company, Pittsburgh, PA

Consortium for Advanced Simulation of Light Water Reactors

CASL was established to provide leading edge modeling and simulation (M& S) capability to improve the performance of *currently operating light water reactors*. Our vision is safer and more productive commercial nuclear power production afforded through comprehensive science-based predictive M& S technology...CASL is developing the Virtual Environment for Reactor Applications, **VERA**. CASL's VERA software simulates nuclear reactor physical phenomena using coupled multi-physics models.

MOOSE AND SHARP



- MOOSE: The Multiphysics Object-Oriented Simulation Environment (MOOSE) is a finite-element, multiphysics framework primarily developed by Idaho National Laboratory. It provides a high-level interface to some of the most sophisticated nonlinear solver technology on the planet.
- SHARP: The Simulation-based High-efficiency Advanced Reactor Prototyping (SHARP) suite of codes enables virtual design and engineering of nuclear plant behavior...researchers have developed a set of simulation tools that provide a highly detailed description of the reactor core and the nuclear plant behavior.

Supercomputing in Research

These kinds of simulations require time on the fastest computers in the world

- Titan (ORNL): 299,008 Opteron Cores (CPU) + 18,688 K21 Keplers (GPU); 27 petaflops
- IBM Sequoia (LLNL): 1,572,864 cores (CPU); 16.32 petaflops

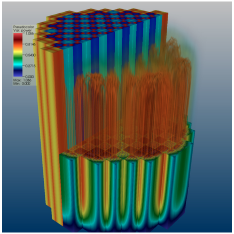
It's Important

". . . At Oak Ridge National Laboratory, they're using supercomputers to get a lot more power out of our nuclear facilities . . . . "

President Obama, 2011 State of the Union
http://www.casl.gov/media/20110127_news.shtml

What Can We Accomplish?

- Predictive simulation
- Model entire facilities at a new level of fidelity
- Coupled multi-physics



What Can We Accomplish?

Integrate

- existing nuclear energy and nuclear national security modeling and simulation capabilities
- and associated expertise
- with high-performance computing

to solve problems that were *previously unthinkable or impractical* in terms of the computing power required to address them.

However, these computer simulations will not completely eliminate the need for *experimental or measurement data* to confirm or “validate” the software.

John Wagner, ORNL

Are You Up To the Challenge?