NE 155/255
Numerical Simulations in Radiation Transport

Lecture 0: Introduction

Kelly L. Rowland

August 28, 2019

Welcome

Let's take a look at the syllabus.

Welcome

Let's take a look at the syllabus.

This is my first time teaching either of these classes, so there's room to adjust class pace accordingly.

Welcome

Let's take a look at the syllabus.

This is my first time teaching either of these classes, so there's room to adjust class pace accordingly.

I give bonus points for

- submitting a pull request to correct course notes
- submitting your homework as a link to a Github repo

References & Resources

- The links in the syllabus are clickable
- There are a number of books listed
  - Some are available electronically
  - My personal favorites are Lewis & Miller and Duderstadt & Hamilton
  - Mostly you'll just use course notes
- The Internet!
  - Chances are that if you're seeing an error message, someone else has seen that same error message

DECF

- DECF Online Help: http://www.decf.berkeley.edu/help/
- How to use SSH to access DECF computers: http://www.decf.berkeley.edu/help/apps/ssh/
- Archipelagos Linux Cluster
  - Access is through SSH ONLY
  - 12 Linux nodes, 26 CPUs
- 1111 Linux Cluster
  - Access is through SSH ONLY
  - 25 Linux nodes, 100 CPUs
- DECF Linux Clusters Status http://www.decf.berkeley.edu/ganglia/

DECF cont'd

- Main Server
  - Kepler (kepler.berkeley.edu) 1.4 Mhz Dell Poweredge 1650 Linux server, 2 GB of real memory
  - Login server for DECF (DO NOT RUN JOBS ON KEPLER)
- 1111 Etcheverry Lab
  - 11 Precision T3500 Workstations (Intel Xeon Quad Core, 6GB RAM)
  - 13 Precision T3400 Workstations (Intel Core 2 Quad, 2GB RAM)
  - 2 HP 4350 black & white laserjet printers
- The 1111 Linux Cluster (*machinename.decf.berkeley.edu*)

boogie	bump	chacha	charleston	fandango
fever	flamenco	foxtrot	freeze	jitterbug
jive	lindy-hop	macarena	mambo	mazurka
merengue	minuet	polka	quickstep	rumba
salsa	sidekick	sock-hop	stomp	

Campus Information

- Mental health resources: http://www.uhs.berkeley.edu/students/counseling/cps.shtml
- Sexual assault support on campus: http://survivorsupport.berkeley.edu/
- Food resources and emergency needs: https://uhs.berkeley.edu/food-resources-emergency-needs

Honor Code

"As a member of the UC Berkeley community, I act with honesty, integrity, and respect for others."

Topics and Schedule

- Computing
- Transport equation
- Deterministic methods
- Monte Carlo methods

Topics and Schedule

- Computing
- Transport equation
- Deterministic methods
- Monte Carlo methods
- Hybrid methods?

Topics and Schedule

- Computing
- Transport equation
- Deterministic methods
- Monte Carlo methods
- Hybrid methods?

I will try to limit repeated content, but please have patience as I need to make sure everyone has a certain set of background information.

Topics and Schedule

- Computing
- Transport equation
- Deterministic methods
- Monte Carlo methods
- Hybrid methods?

I will try to limit repeated content, but please have patience as I need to make sure everyone has a certain set of background information.

Homework will be due approximately every two weeks.

Final Projects

- We'll talk about final projects about halfway through the semester.
- Students enrolled in NE 155 may choose to do either an analysis project or a programming-based project
  - Regardless of project type, the work should be at a course-appropriate level
- Students enrolled in NE 255 will be expected to do a programming-based project
  - Analysis projects are acceptable with compelling justification
- I encourage you to choose a project that is useful to your research
  - At the very least, it should be relevant to your interests
- Keep this in the back of your mind as we go through the course

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

Identify the Problem

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**.

Identify the Problem

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**.
- 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.

Identify the Problem

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**.
- 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.
- Many scales, physics, and systems are involved (read: this is a difficult problem)

Identify the Problem

- 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 predictive computing fits in...

Predictive Computing

The idea behind predictive computing is to have

- a mathematical model that is sufficiently representative
- *and* methods that are sufficiently accurate
- *and* an implementation this is sufficiently robust

Predictive Computing

The idea behind predictive computing is to have

- a mathematical model that is sufficiently representative
- *and* methods that are sufficiently accurate
- *and* an implementation this is sufficiently robust

such that calculations can

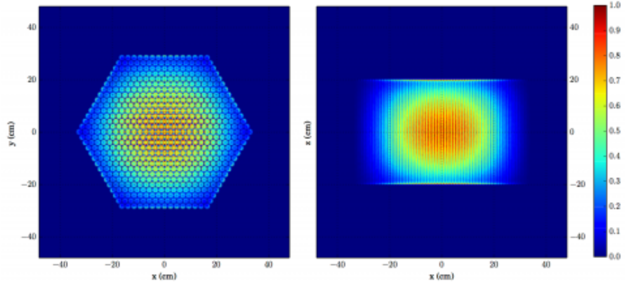
- inform experiment design
- *and* replace experiments
- *and be so reliable that we can make new design choices using only calculations*.

We will only look at one piece needed for predictive computing...

Mathematical Model

We'll solve the steady state Boltzmann Transport Equation

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



Historical Context

- Computing limitations of the past caused
  - Heavy reliance on expensive and often complicated experiments
  - Inaccuracy resulted in *significant design margins* $\rightarrow$ negative impact on plant economics
  - Exploration of novel reactor design concepts was greatly constrained by fundamental limitations in the predictability of the models
- Many codes (methods+implementation) developed then are still used

Historical Context

- Computing limitations of the past caused
  - Heavy reliance on expensive and often complicated experiments
  - Inaccuracy resulted in *significant design margins* $\rightarrow$ negative impact on plant economics
  - Exploration of novel reactor design concepts was greatly constrained by fundamental limitations in the predictability of the models
- Many codes (methods+implementation) developed then are still used
- Can we update these tools or do we need new ones?

Historical Context

- Computing limitations of the past caused
  - Heavy reliance on expensive and often complicated experiments
  - Inaccuracy resulted in *significant design margins* $\rightarrow$ negative impact on plant economics
  - Exploration of novel reactor design concepts was greatly constrained by fundamental limitations in the predictability of the models

- Many codes (methods+implementation) developed then are still used
- Can we update these tools or do we need new ones?
- What *methods* will take us into the future?

Historical Context

- Computing limitations of the past caused
  - Heavy reliance on expensive and often complicated experiments
  - Inaccuracy resulted in *significant design margins* $\rightarrow$ negative impact on plant economics
  - Exploration of novel reactor design concepts was greatly constrained by fundamental limitations in the predictability of the models
- Many codes (methods+implementation) developed then are still used
- Can we update these tools or do we need new ones?
- What *methods* will take us into the future?
- What will the architectures look like?

Historical Context

- Computing limitations of the past caused
  - Heavy reliance on expensive and often complicated experiments
  - Inaccuracy resulted in *significant design margins* $\rightarrow$ negative impact on plant economics
  - Exploration of novel reactor design concepts was greatly constrained by fundamental limitations in the predictability of the models
- Many codes (methods+implementation) developed then are still used
- Can we update these tools or do we need new ones?
- What *methods* will take us into the future?
- What will the architectures look like?
- What and how do we need to include other physics?

This Class

We will focus on

- Understanding the mathematical model (more of that in 250)
- Learning computational methods (most of class)
- (possibly) A little bit of implementation (take a computing class for this)
- Assessing the answer

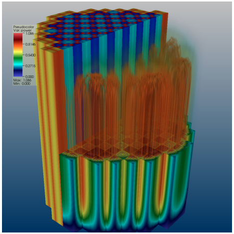
Supercomputing in Research

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

- Summit (ORNL): 2.4M IBM POWER9 Cores (CPU) + 27.6k NVIDIA Volta V100s (GPU); 148,600 teraflops (10E12 flops) = 148.6 petaflops

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, INL

Current State: CASL

2010: the DOE announced *Oak Ridge National Laboratory* won the Nuclear Energy Modeling and Simulation Energy Innovation Hub (awarded 5 more years in 2015), 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's focus is on *currently operating light water reactors*.
- They've developed the Virtual Environment for Reactor Applications, **VERA**, which simulates nuclear reactor physical phenomena using coupled multi-physics models.
- They have LWR-specific challenge problems such as GTRF, CRUD, PCI, DNB, FAD, RPV internals, etc.

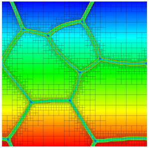
Current State: MOOSE and SHARP

- MOOSE: The Multiphysics Object-Oriented Simulation Environment (MOOSE) is a finite-element, multiphysics framework primarily developed by Idaho National Lab. It provides a high-level interface to some of the most sophisticated nonlinear solver technology on the planet.

Current State: MOOSE and SHARP

- MOOSE: The Multiphysics Object-Oriented Simulation Environment (MOOSE) is a finite-element, multiphysics framework primarily developed by Idaho National Lab. 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 (Argonne National Lab) have developed a set of simulation tools that provide a highly detailed description of the reactor core and the nuclear plant behavior.

Quick Comparison



- MOOSE and SHARP focus more heavily on advanced reactor design.
- They have package that address more types of physics than CASL.
- MOOSE is open source, though many of the "animals" that do the physics are not.
- MOOSE and SHARP are supported by DOE Office of Nuclear Energy, while CASL is the Office of Science.

What Are People Working on Now?

Examples from DOE-NE funding opportunity announcement

- Advanced Reactor Methods Topics
  - Sodium Fast Reactor
  - High Temperature Gas Reactor
  - Molten Salt Reactor
- Reactor Concepts
- Nuclear Energy Advanced Modeling and Simulation (NEAMS): Core Neutronics
- Grand Challenge Problem for Nuclear Energy
- Critical Data Needs for NEAMS

Are You Up To the Challenge?