CS207: Systems Development for Computational Science

https://iacs-cs-207.github.io/cs207-F18/

Instructor: David Sondak
TFs: Timothy Lee, Bernard Klenyhans, and Shiyun Qiu

Harvard University
Institute for Applied Computational Science

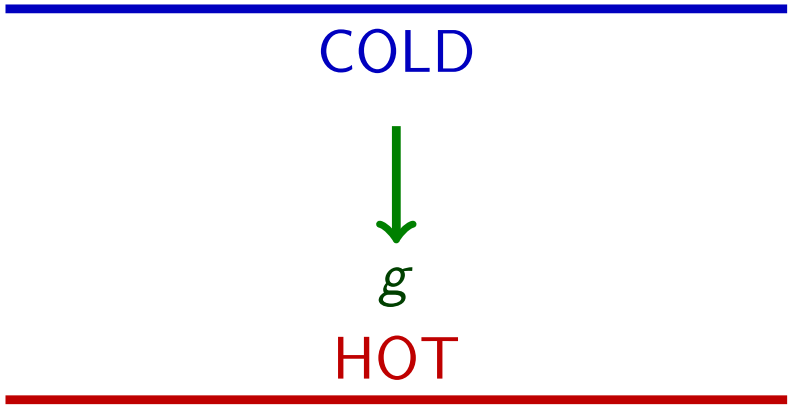
9/4/2018

Motivation: Thermal Convection and the Geodynamo

Thermal convection drives most fluid flows in the universe

Motivation: Thermal Convection and the Geodynamo

Thermal convection drives most fluid flows in the universe

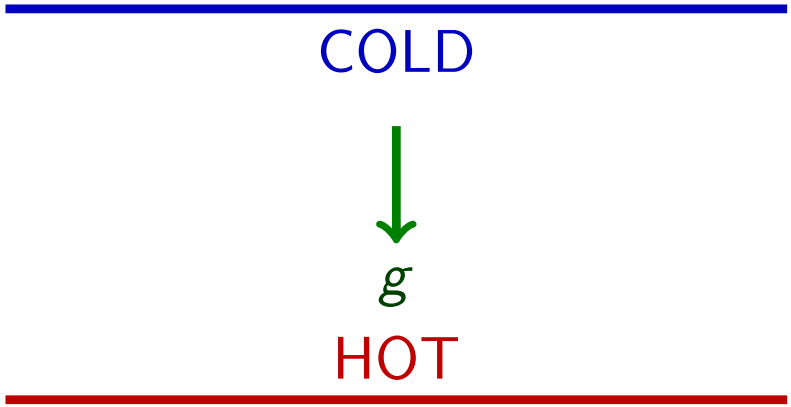


Cold fluid falls, hot fluid rises

▸ Plate Tectonics Video

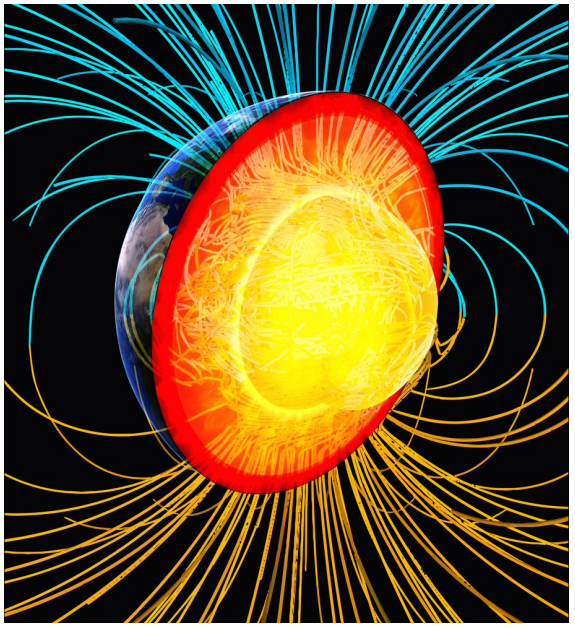
Motivation: Thermal Convection and the Geodynamo

Thermal convection drives most fluid flows in the universe



Cold fluid falls, hot fluid rises

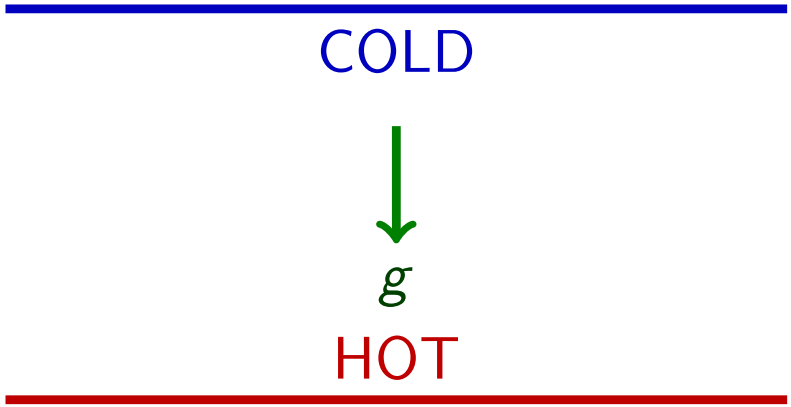
▸ Plate Tectonics Video



DESY

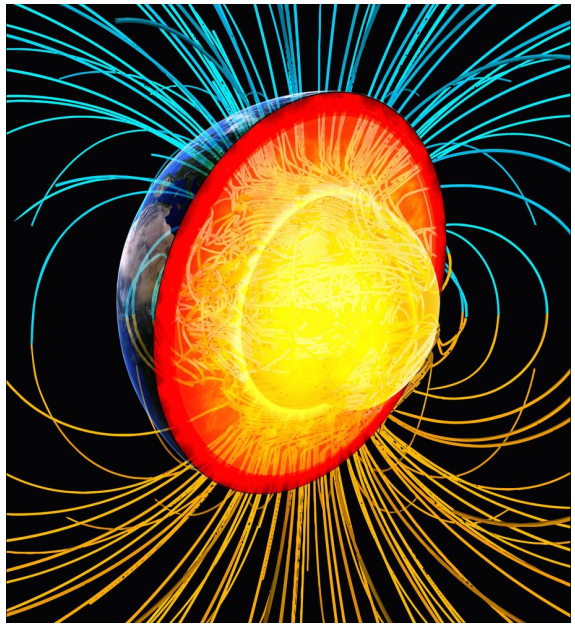
Motivation: Thermal Convection and the Geodynamo

Thermal convection drives most fluid flows in the universe



Cold fluid falls, hot fluid rises

▸ Plate Tectonics Video



DESY

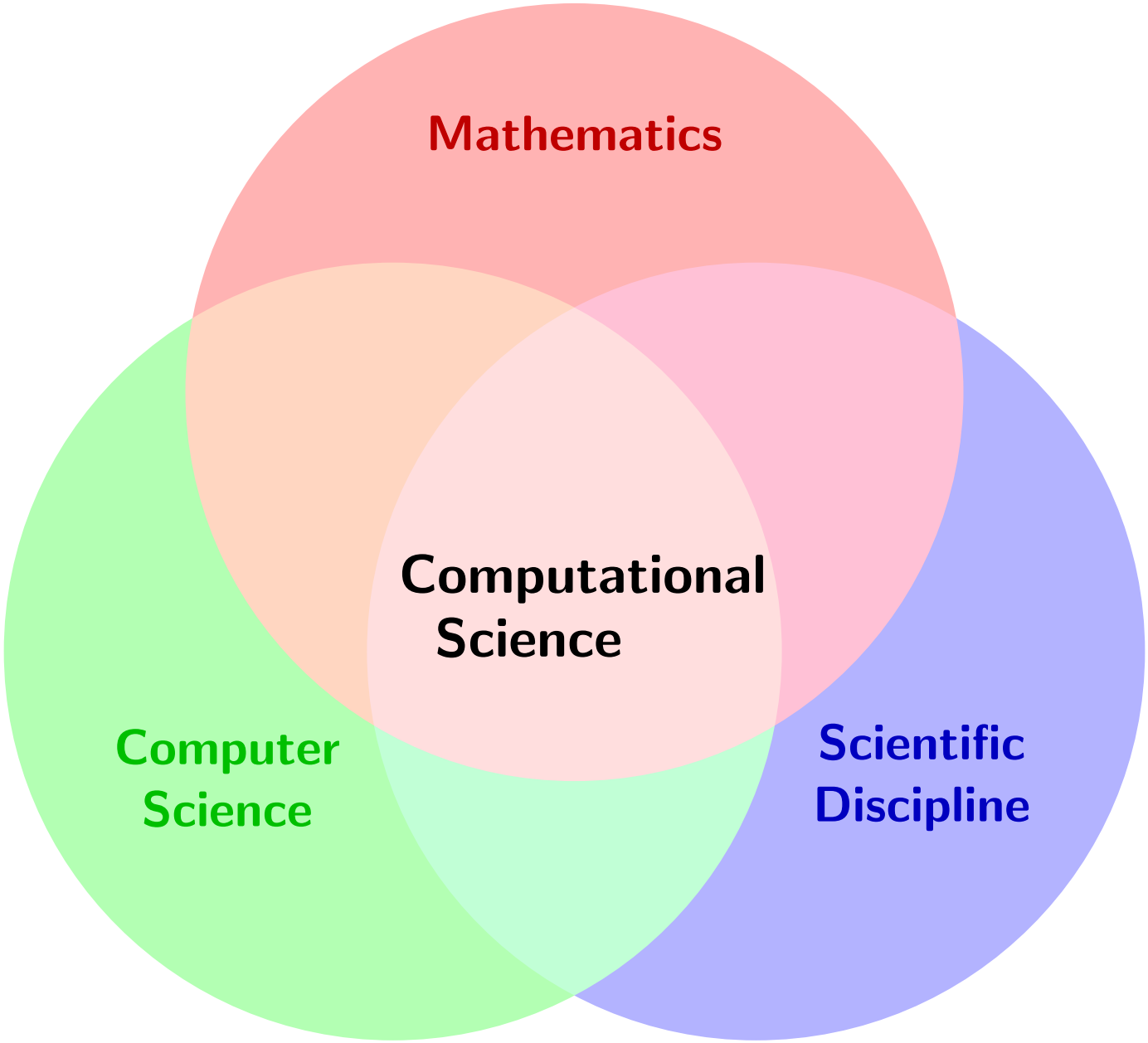
$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u}\, T) = k \nabla^2 T$$

- Ignoring $\nabla \cdot (\mathbf{u}\, T)$ gives the usual heat conduction equation!

Motivation: The Pillars of Science

Motivation: The Pillars of Science

Computational Science



Why take this class?

- Scientific software is complex
- Your code needs to be:
  - Reuseable
  - Portable
  - Robust
- Must go beyond "scripting"

Why take this class?

- Scientific software is complex
- Your code needs to be:
  - Reuseable
  - Portable
  - Robust
- Must go beyond "scripting"

Why take this class?

- Scientific software is complex
- Your code needs to be:
  - Reuseable
  - Portable
  - Robust
- Must go beyond "scripting"

CS207 Objectives

To give students who may not have a traditional computer science background the knowledge and tools to develop and maintain effective software for computational science applications.

Why take this class?

- Scientific software is complex
- Your code needs to be:
  - Reuseable
  - Portable
  - Robust
- Must go beyond "scripting"

CS207 Objectives

To give students who may not have a traditional computer science background the knowledge and tools to develop and maintain effective software for computational science applications.

Who should take this class?

- Any kind of scientist is welcome to take this class!
- This course is computer science for people who aren't computer scientists:
  - Data scientists
  - Biologists
  - Chemists
  - Engineers
  - Physicists
  - Mathematicians
  - Economists
  - ⋮
- It is also for computer scientists who want to develop scientific software
- CS207 is for students who need to know effective and modern software practices for their career

Sample Topics

A few selected topics to be covered:

- Version control
- Python (basics)
- How Python works
- Software documentation
- Software testing
- Object-oriented programming
- Data structures
- Databases

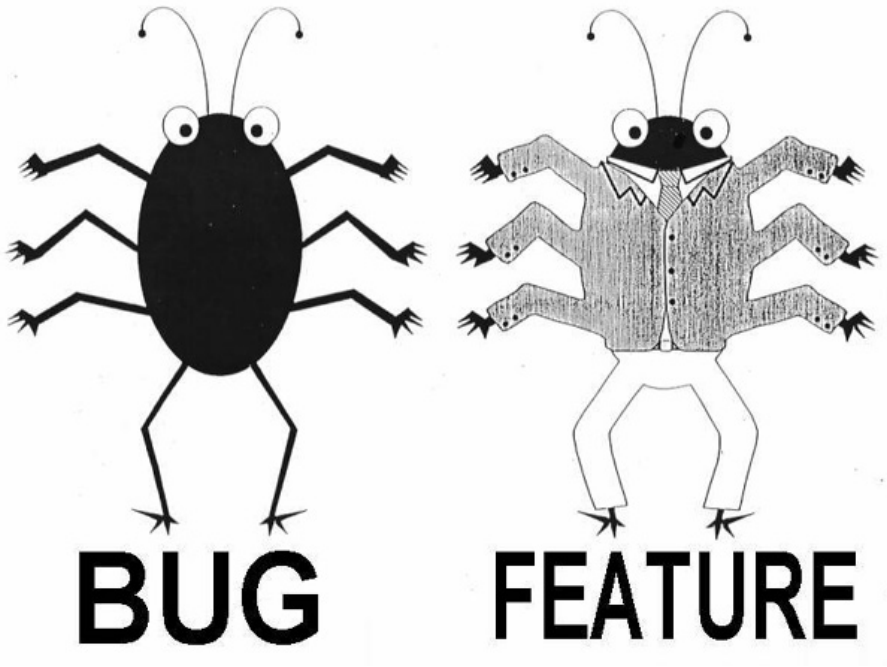
Sample Topics

A few selected topics to be covered:

- Version control
- Python (basics)
- How Python works
- Software documentation
- Software testing
- Object-oriented programming
- Data structures
- Databases

Other potential topics (not guaranteed):

- Debuggers and debugging
- Build systems (Make files, autotools, ...)
- Compiled languages
- Navigating a Unix OS



Course Structure

- CS207 is an application-driven course
- Two, 1.5 hour lectures per week
- Lectures centered around group programming exercises using Jupyter notebooks
- Programming assignments for homework
- Primary deliverable is a software development project
- All course content hosted on GitHub

Course Website:

`https://iacs-cs-207.github.io/cs207-F18/`

Course Project: Overview

- You will work in groups of 3 to 4 people (assigned by teaching staff)
- You will add to your library throughout the semester
- The project consists of two milestones
- For the final project, you will add a non-trivial feature to your library
- A portion of your grade will come from peer-assessment
- Exact details on website

Course Project: The Topic

Automatic differentiation

Course Project: The Topic

Automatic differentiation

What is Automatic Differentiation?

Course Project: The Topic

Automatic differentiation

What is Automatic Differentiation?

- A way to evaluate derivatives of functions and computer programs

Course Project: The Topic

Automatic differentiation

What is Automatic Differentiation?

- A way to evaluate derivatives of functions and computer programs
- Computes derivatives to *machine precision*!

Course Project: The Topic

Automatic differentiation

What is Automatic Differentiation?

- A way to evaluate derivatives of functions and computer programs
- Computes derivatives to *machine precision*!
- Can be very efficient and accurate

Course Project: The Topic

Automatic differentiation

What is Automatic Differentiation?

- A way to evaluate derivatives of functions and computer programs
- Computes derivatives to *machine precision*!
- Can be very efficient and accurate
- Also known as "algorithmic differentiation"

Course Project: The Topic

Automatic differentiation

What is Automatic Differentiation?

- A way to evaluate derivatives of functions and computer programs
- Computes derivatives to *machine precision*!
- Can be very efficient and accurate
- Also known as "algorithmic differentiation"

We will have at least two lectures on automatic differentiation this semester to cover the main points.

Why Automatic Differentiation?

- Encapsulates many ideas in software design
  - Object-oriented programming
  - Operator overloading
  - Datastructures

Why Automatic Differentiation?

- Encapsulates many ideas in software design
  - Object-oriented programming
  - Operator overloading
  - Datastructures
- Pervasive throughout science and gaining steam
  - Neural networks and backpropagation
  - Hamiltonian Monte Carlo methods
  - Full Jacobian calculations
  - Jacobian-free calculations

AD Teaser

Suppose we have a function like

$$y = \exp\left(-\sqrt{x + \cos^2(x)}\right) \sin\left(x \ln\left(1 + x^2\right)\right).$$

AD Teaser

Suppose we have a function like

$$y = \exp\left(-\sqrt{x + \cos^2(x)}\right) \sin\left(x \ln\left(1 + x^2\right)\right).$$

The symbolic derivative is

$$y' = \exp\left(-\sqrt{x + \cos^2(x)}\right) \cos\left(x \ln\left(1 + x^2\right)\right) \left(\frac{2x^2}{1 + x^2} + \ln\left(1 + x^2\right)\right)$$
$$- \exp\left(-\sqrt{x + \cos^2(x)}\right) \frac{1 - 2\cos(x)\sin(x)}{2\sqrt{x + \cos^2(x)}} \sin\left(x \ln\left(1 + x^2\right)\right)$$

AD Teaser

Suppose we have a function like

$$y = \exp\left(-\sqrt{x + \cos^2(x)}\right) \sin\left(x \ln\left(1 + x^2\right)\right).$$

The symbolic derivative is

$$y' = \exp\left(-\sqrt{x + \cos^2(x)}\right) \cos\left(x \ln\left(1 + x^2\right)\right) \left(\frac{2x^2}{1 + x^2} + \ln\left(1 + x^2\right)\right)$$
$$- \exp\left(-\sqrt{x + \cos^2(x)}\right) \frac{1 - 2\cos(x)\sin(x)}{2\sqrt{x + \cos^2(x)}} \sin\left(x \ln\left(1 + x^2\right)\right)$$

And that's only the first derivative!

Next Steps

Go to `https://github.com/IACS-CS-207/cs207-F18/blob/master/lectures/L01/L1.ipynb`.

Set up GitHub and get the course repository

- If you don't already have a GitHub account, go to `https://github.com/` and create an account.
- From your GitHub homepage, find the **New Repository** button:
  1. Click the **New Repository** button.
  2. Name the repository `cs207_firstname_lastname`.
  3. Select **Private** for the repository type.
  4. **Do not**:
     - initialize with a README,
     - add a `.gitignore` file,
     - or choose a license.
  5. Select `Create Repository`
  6. You will see four options. Choose the one at the bottom of the page: "...or import code from another repository" and click the `Import code` button.
  7. Enter https://github.com/IACS-CS-207/cs207-F18.git in the text field under `Your old repository's clone URL`.
  8. Click the **Create Repository** button.
- Congrats! You now have the course repo!

Connecting to the main course repo (1)

You now have all the course content as it currently stands. The problem is, you can't update it!

We will be doing most of our work from the command line.

Mac and Linux Instructions

- Open `Terminal`.
- Type `which git`. If git is installed, you will see a path to git similar to `/usr/bin/git`.
- If git is not installed, then you won't see anything. Follow the instructions here to install it: `https://git-scm.com/book/en/v2/Getting-Started-Installing-Git`.

Windows

You should install `git BASH`: `https://git-for-windows.github.io`

Connecting to the main course repo (2)

- Okay. Now you have git.
- Hopefully you still have your new GitHub repo open. If not, open it.
- Click the `Clone or download` button and copy the text.
- Open your terminal session (or git-BASH session).
- Type `git clone url_to_repo_just_copied`.
- Now you have a local copy of your repo!!

But how do you link your repo to the main course repo?

Connecting to the main course repo (3)

- Inside your local repo, type `git remote -v`. This tells you the remote repos that are known to you.
- You want to add a new remote repo (the main course repo).
- Type `git remote add upstream https://github.com/IACS-CS-207/cs207-F18.git`
- Now type `git remote -v`. See the new repo? It's short name is upstream.
- If you want to get updates from the main course repo, just type `git pull upstream master`. Try it!
- That command says to fetch and merge changes from the master branch of the repo pointed to by upstream.
- Notice that everything is up to date.

Now you have the main course repo on your GitHub page and you know how to get updates from the main repo.

Next time, we'll start to delve into the details of exactly what this all means.