AC290r

Extreme Computing

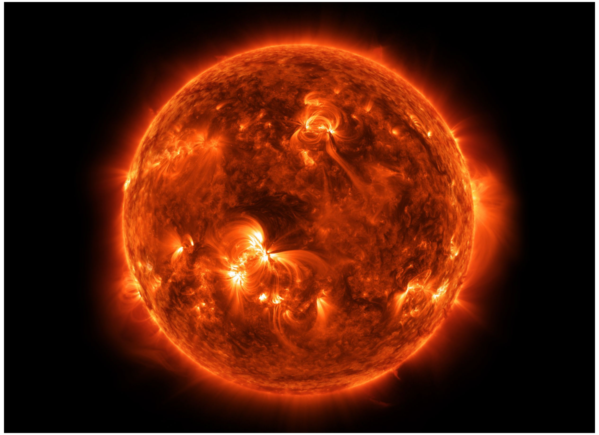
https://harvard-iacs.github.io/2019-AC290/

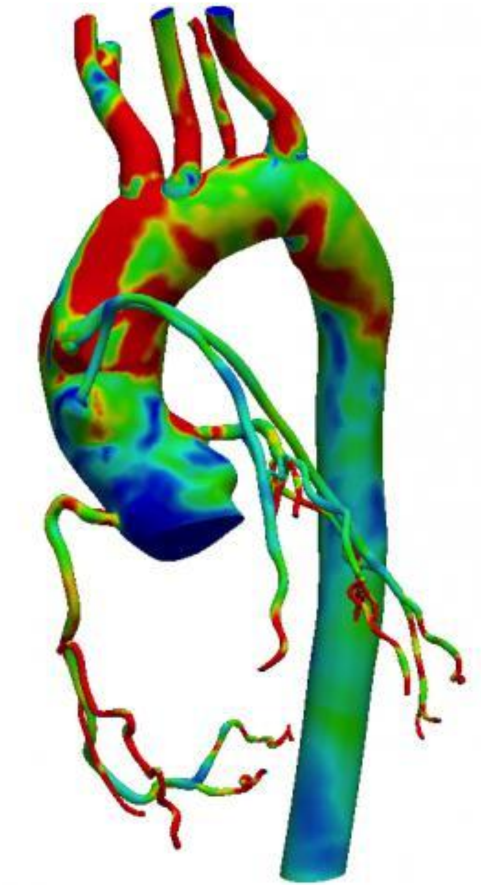
David Sondak

Objective

Explore the techniques, infrastructure, and algorithms used for extreme computing.

Motivation





- Simulate real-world physical processes to enable scientific and engineering predictions
- Requires large scale parallel simulations at the exascale and beyond
- Cutting edge extreme scale simulations use ***millions*** of compute cores

Who should take this class?

- Scientists who want to run large-scale simulations
  - MPI (distributed across many CPU nodes)
  - GPU (multiple GPU nodes)
- People interested in the end-to-end simulation process
  - Building a large, scalable code base
  - Submitting jobs on supercomputers
  - Post-processing and data analysis to interpret simulation results
- Students curious about fluid mechanics and numerical methods
  - Classical incompressible fluid mechanics
    - Simulations with the finite element method
  - Hemodynamics
    - Simulations with the Lattice Boltzmann method

Sample Topics

- MPI and GPU computing
- Basic fluid mechanics
- Numerical methods and algorithms including
  - Finite Element Method
  - Lattice Boltzmann Method
  - Linear algebra solvers
- Building large codes on Linux systems (Odyssey)
- Submitting and managing large jobs
- Processing and analyzing large data sets

Course Structure

- AC290r is a project-based course
- Two 3 hour meetings per week
    - Tuesdays, Thursdays
    - 9 AM - 11:45 AM
- Tuesdays: Lectures and hands-on activities
- Thursdays: Coding and hack-a-thons
- Two modules
    - Computational fluid dynamics with finite element code from Sandia National Labs (1st half)
    - Lattice Boltzmann simulations for hemodynamics (2nd half)
- Homework assignments due every other week
- Guest lectures from scientists at Sandia National Laboratories, Sapienza University, and the Institute for Calculus Applications in Italy

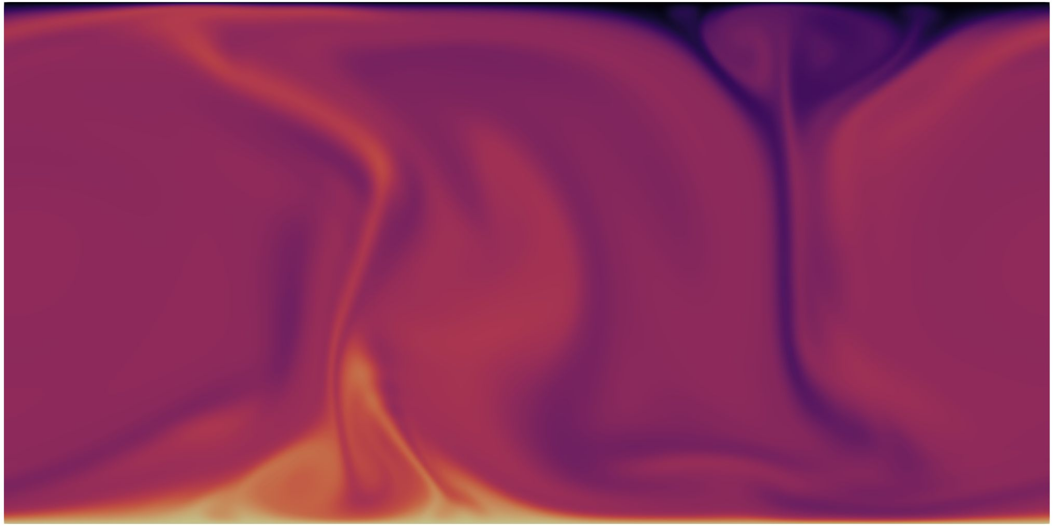
A Few Key Learning Outcomes

By the end of the course you should:

- Understand why scientific and engineering problems need extreme computing
- Understand the end-to-end mathematical and computing infrastructure required in large scale problems
- Understand and execute the end-to-end development process required in large scale problems
- Feel comfortable working with large scale, production-quality scientific software in real supercomputing environments

Module 1: Computational Fluid Dynamics

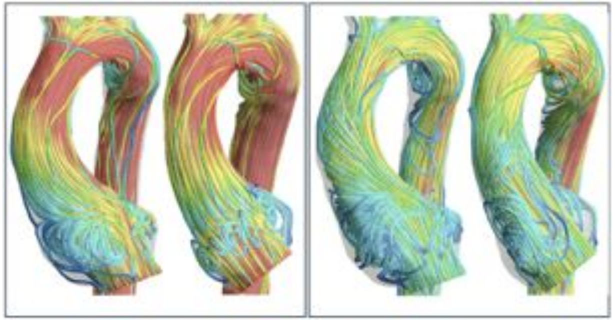
- Simulate fluid flows
- Understand and execute the build process for a large-scale, high performance finite element code
- Submit jobs on the Harvard supercomputer Odyssey
    - We'll use as many cores as we can
    - Duct flow (scaling studies)
    - Thermal convection
- Post process results and perform basic visualizations

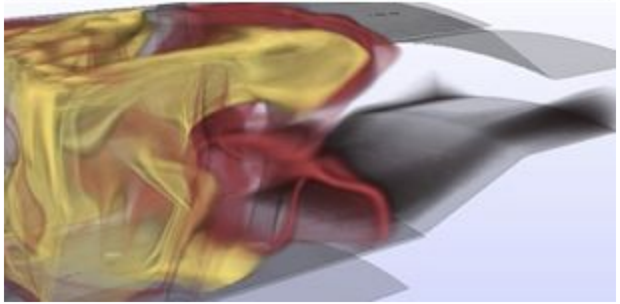


Module 2: Biofluidics

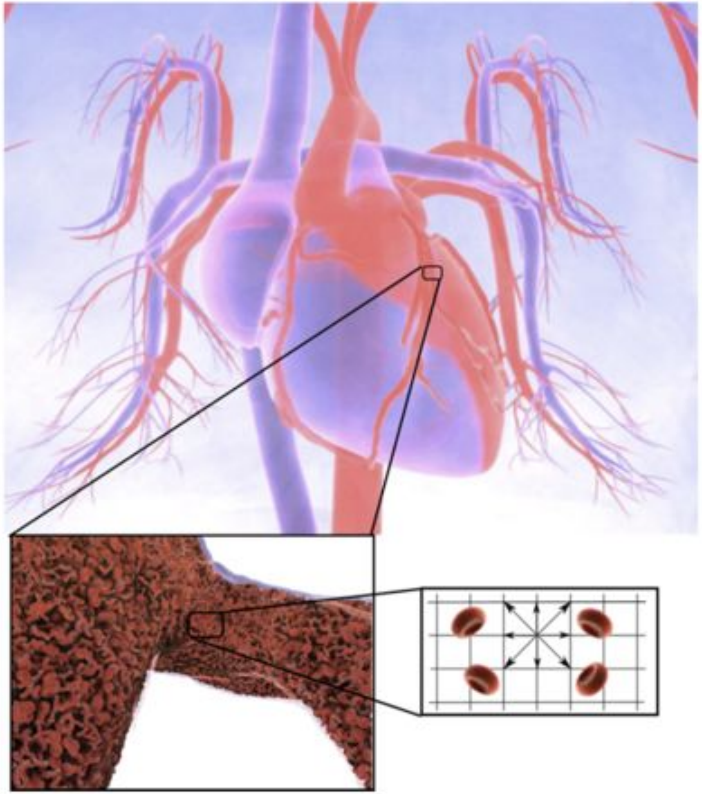
- Learn to simulate on powerful supercomputers
- Interrogate data with advanced visualization methods

- Prep. & simulate physiological systems
- Use Lattice Boltzmann to run Newtonian flows in complex anatomies





Cellular Scale Hemodynamics



- Setup hemodynamic simulations in realistic vessels and in presence of blood cells
- Control the mechanical properties of vessels and blood cells
- See how flow and structures interplay in real environments
- See multiscale simulations at work!

Course Head

David Sondak is a lecturer on computational science in the Institute for Applied Computational Science at Harvard University. David's research interests are in fluid mechanics, turbulence, and physics-based, data-driven models and algorithms.

David loves spending time outside with his dog and discussing dogs in general.

Co-Instructor

Simone Melchionna is a researcher in the Institute of Complex Systems in Italy. He studies complex and biological flows with high-performance simulation methods. He is particularly interested in biofluids such as blood.

Teaching Fellow

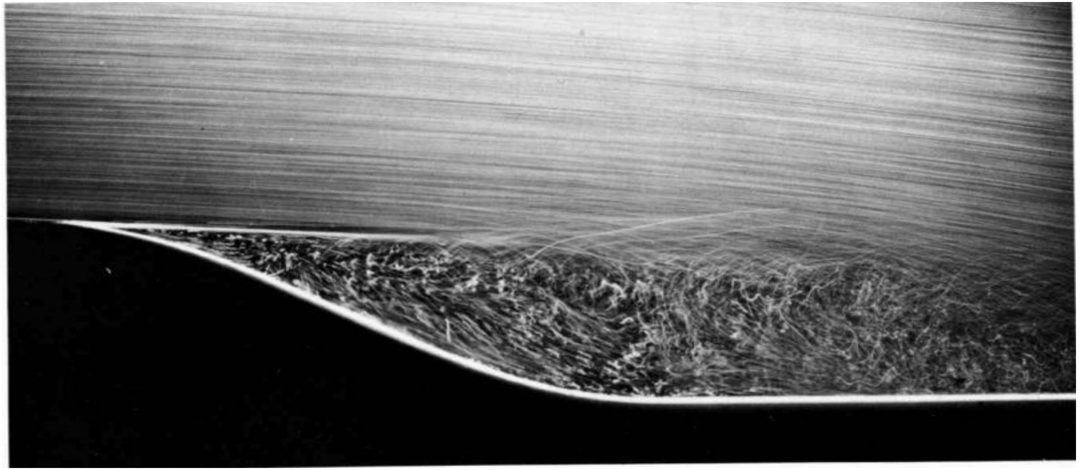
Rui Fang is a second year master's student in the Institute for Applied Computational Science. She works on physics-based machine learning approaches to turbulence modeling.

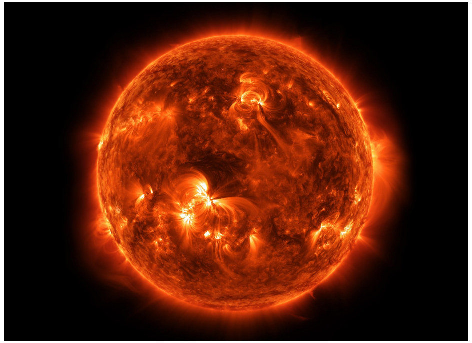
In her free time she likes playing the electric guitar.

Websites and Resources

https://harvard-iacs.github.io/2019-AC290/

Why Fluids?





What Are Fluids?

- This is a really hard question to answer!
  - Continuum description?
  - Kinetic description?
  - Molecular? Quantum?
- In the first half of the class, we’ll take the continuum approach
- We have an intuitive idea of fluid behavior
  - Fluids can be easily set into motion
- But how do we describe this mathematically?

An Observation

A big difference between fluids and solids:

Fluids cannot withstand an imposed shear force!

We relate fluid stresses to the *rate of deformation* rather than the deformation.

$$\textit{stress} \propto \textit{rate of strain}$$

This observation sets the stage for describing fluid behavior.

Some Comments About Fluids

- There are a variety of fluids and fluid flows out there
  - Newtonian, Non-Newtonian (https://www.youtube.com/watch?v=3zoTKXXNQIU)
  - Compressible vs Incompressible flows
  - With and without magnetic fields
  - Chemistry effects
  - …
- Different applications have different requirements
  - Continuum description (e.g. engineering systems)
  - Mesoscale description (e.g. blood flow with cells)
  - Molecular description (e.g. Brownian motion)
- Extremely rich physics

Some Mathematical / Numerical Comments

- The governing equations are (so far) analytically intractable
  - There is no general theoretical approach to solving them
    - Some very simple solutions exist (you'll meet some of these)
  - We barely understand the nature of the equations, let alone techniques for their solution
    - Millennium Prize Problems
- Numerical methods are often fruitful for exploring analytically intractable equations
- When it comes to fluids, even numerical methods struggle
  - Multiscale phenomena
  - Incredibly demanding numerical resolutions required to represent even simple physics
  - Need new algorithms and new / better / more resources

Huge physics problems

+

Epic computational resources

=

Extreme Computing