PHYS 141/241

Lecture 01: Introduction

Javier Duarte — April 3, 2023

Welcome to PHYS 141/241

- Fill out the pre-course survey: https://forms.gle/gFZq2HqnETdfeKwW9
- Let's review the syllabus
- Instructor: Javier Duarte (jduarte@ucsd.edu), Office hours after lecture
- TA: Yi Guo, Office hours during last hour of Lab
- Learning outcomes:
  - Design computer programs to numerically solve physics problems, like the N-body problem.
  - Consider multiple approaches and compare their computational performance, accuracy, and fidelity to physical laws.
  - Find and choose the best tool or programming language for the task
  - Visualize the solutions
  - Collaborate with peers to tackle complex, realistic problems
  - Present findings

Assignment breakdown

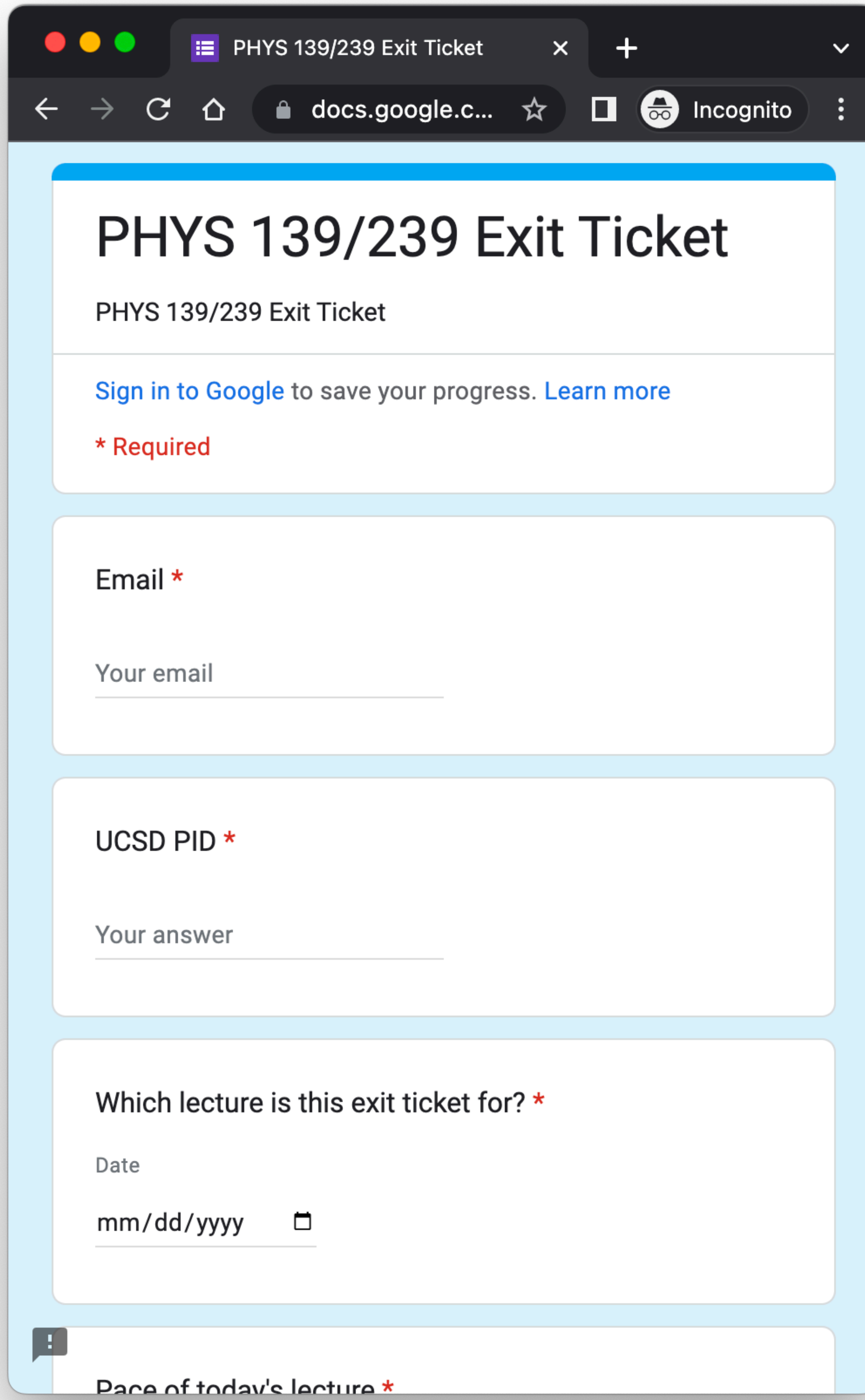
- 40% Homework
- 10% Quizzes
- 5% Participation in class/via Slack and completion of exit tickets
- 20% Midterm project
- 25% Final project

Homework

- Half of grade will be from turning in first version
  - Graded on effort and completeness (for all problems)
  - Solution released shortly afterward
- Half of grade will be from turning in corrected solution
  - Graded on effort and correctness (for all problems)
- Report (pdf file) uploaded to Gradescope
- Code (zip file) uploaded to Gradescope
- First homework will be released later this week (due in Week 3)

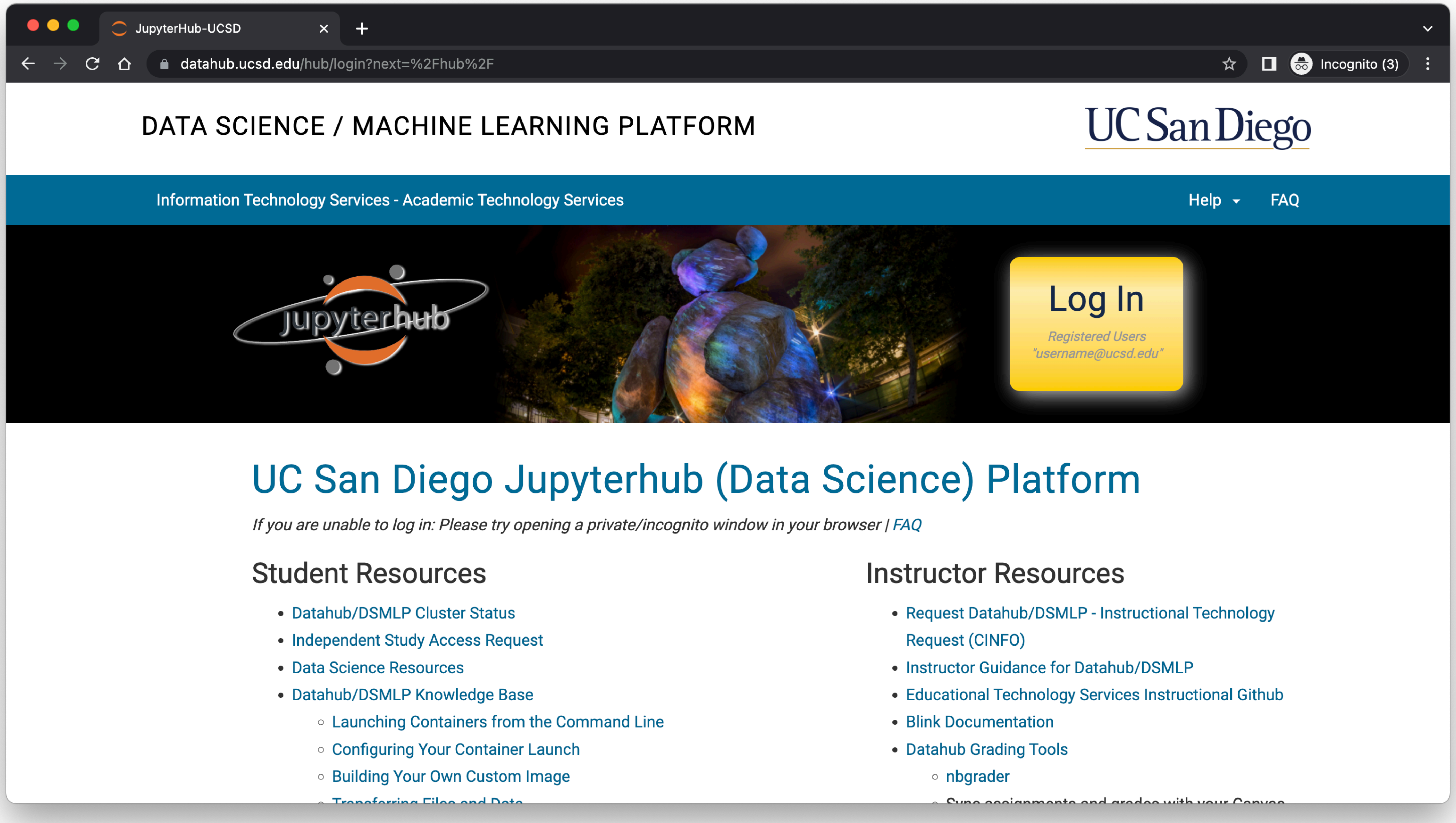
Exit tickets

- Exit tickets: https://forms.gle/b7ZDZRm1czrnHaGBA
    - Designed to see how you felt about the lecture, what you took away, whether you have any further questions or feedback
    - Filling it out will go toward the 10% participation score



DataHub

- We will use DataHub for in-class hands-on portions
  - Recommend to use it for homework, final project, etc.
- Address: datahub.ucsd.edu
- Similar to public, free services Google Colab, but with access to better CPUs and GPUs and run by UCSD
- Provides a “Jupyter notebook” interface (Python-based but interactive coding like MATLAB/Mathematica)



Slack

- Join the Slack workspace for the course: https://join.slack.com/t/ucsdphys141/shared_invite/zt-1sqbgv7g0-fdUpFWSIoZiRXkObMDOvsQ

- Tutorial: https://slack.com/help/categories/360000049063

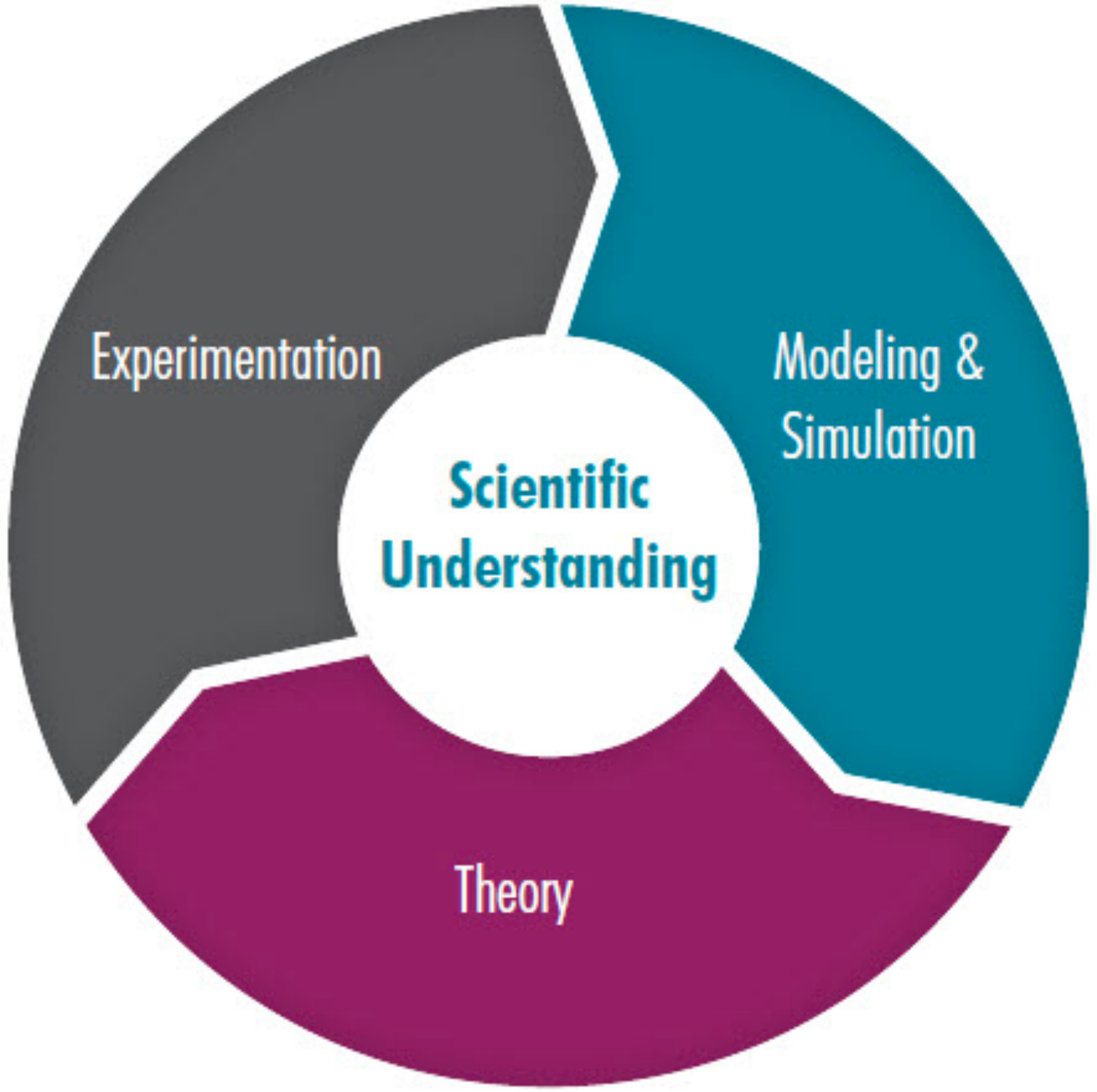
- Feel free to create channels to collaborate with others, etc.

Course overview

- Course overview; galaxy collisions; preview of final projects
- Newtonian mechanics and its numerical methods (Euler, Runge-Kutta, leapfrog)
- Collisionless Boltzmann equation (CBE)
- Plummer's spherical model as analytic solution to the CBE
- Galaxy modeling
- Isothermal CBE models for the bulge of the galaxy
- Jeans equation and virial theorem for full control of the CBE of the galaxy
- Monte Carlo realization of the CBE solution
- Treecode and GyrfalcON implementations

Computer modeling

- Computer modeling plays a very important role in science today
- Physical sciences are characterized by an interplay between experiment and theory
  - Experiment: a system is subjected to measurements, and results, expressed in numeric form, are obtained
  - Theory: a model of the system is constructed, usually in the form of a set of mathematical equations
- Modeling and simulation live at the intersection between (and supplement) theory and experiment
  - But are not a substitute for real-world experimentation



Three components to modeling

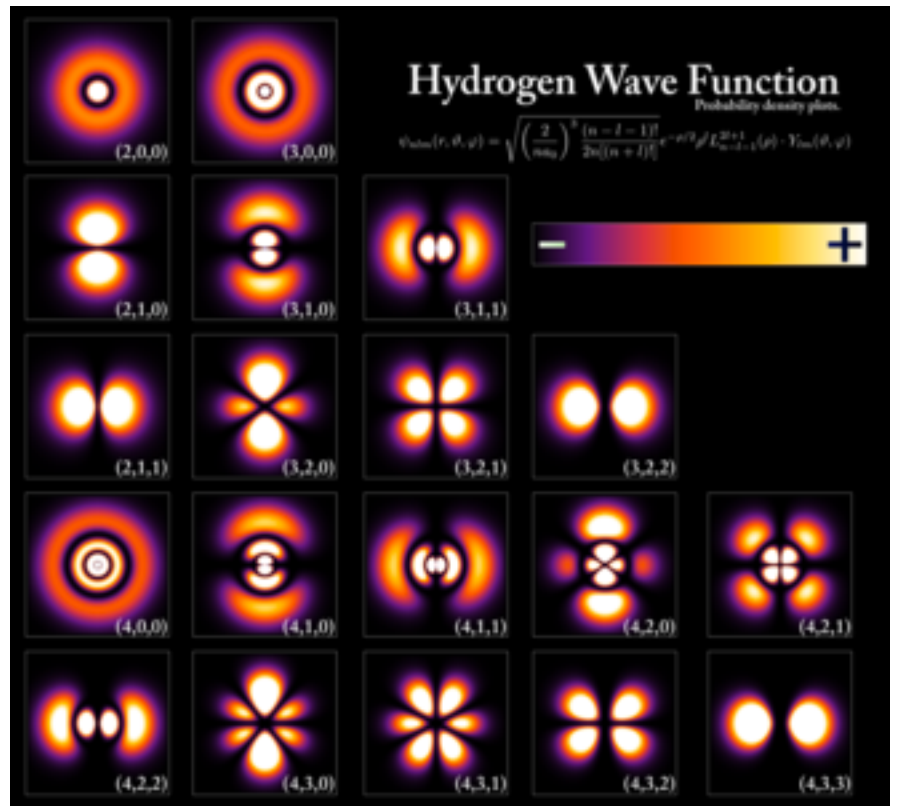
1. The physical problem and its theoretical model

   - Necessary to understand the underlying physics of the problem

   - Only ask the computer to do the things which cannot be done otherwise (e.g. analytically).

   - Development of computer experiments has altered substantially the relationship between theory and experiment, allowing “thought experiments” and more realistic, complex models

   - Creativity is an important component!



Three components to modeling

2. Algorithm and software implementation

    - The advent of high speed computers and “high-level” programming languages, starting with Fortran (1957), C (1972), C++ (1980), Python (1990), Rust (2010), Julia (2012), … made modeling & simulation much more accessible to scientists

Three components to modeling

3. Analysis and visualization

    - Analysis: physical interpretation of the data generated by the computer simulation

    - Visualization tools are indispensable in the interpretation of the results

- For illustration we will discuss the computer modeling of N-body problems with applications to the collisions of galaxies

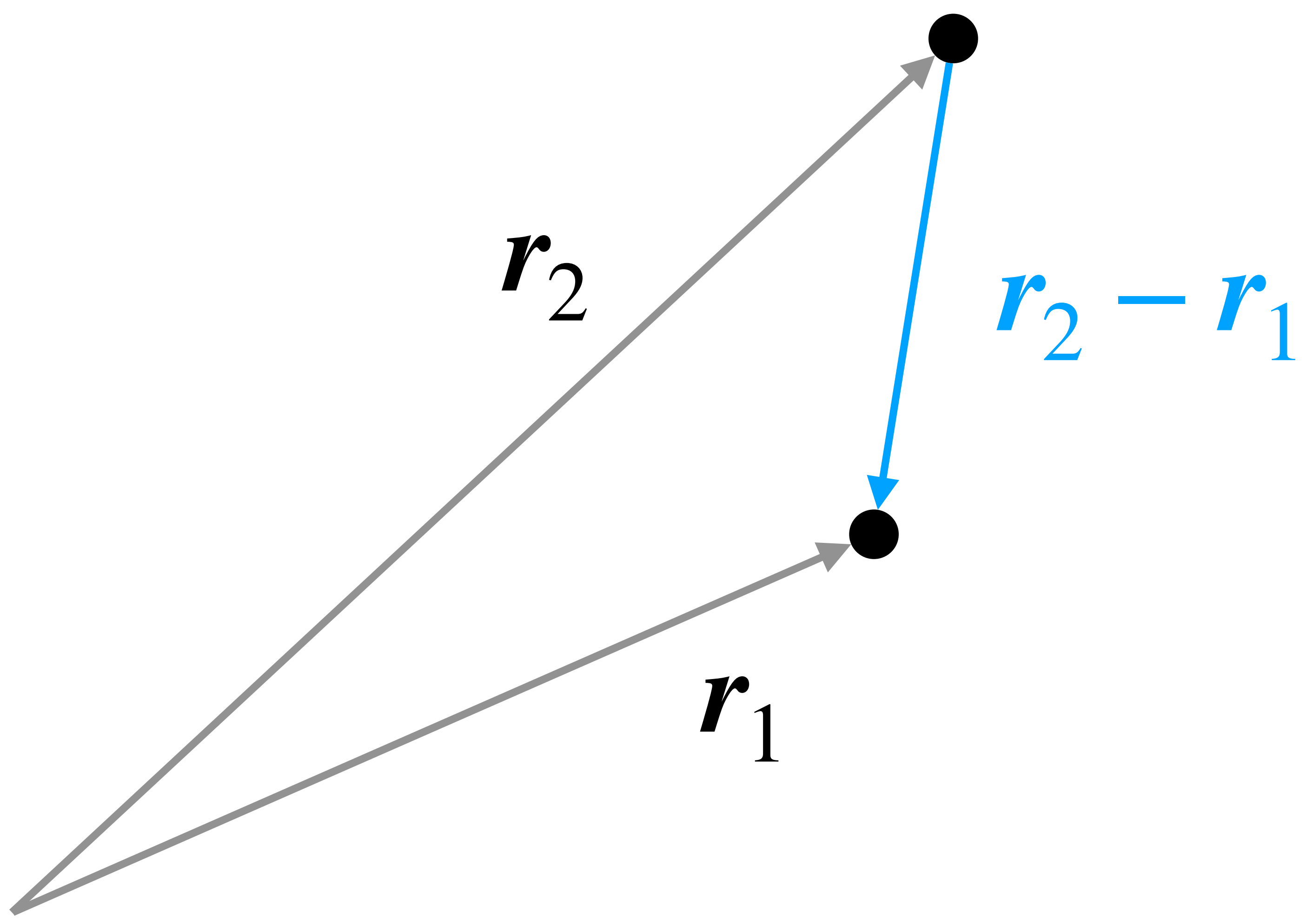
N-body problem

- Any gravitational system may be idealized as a collection of N point-sized bodies, each with mass m_i, 3D position vector $\boldsymbol{r}_i$, and 3D velocity vector $\boldsymbol{v}_i$
- The energy $E = K + V$ for such a system is given by Newtonian mechanics where E depends on all body positions and velocities
- Force on 1 due to 2 is in the direction of $\dfrac{\boldsymbol{r}_2 - \boldsymbol{r}_1}{|\boldsymbol{r}_2 - \boldsymbol{r}_1|}$ with magnitude $\dfrac{Gm_1m_2(\boldsymbol{r}_2 - \boldsymbol{r}_1)}{|\boldsymbol{r}_2 - \boldsymbol{r}_1|^2}$
- Therefore, the force on 1 due to 2 is

$$\boldsymbol{F}_{12} = \frac{Gm_1m_2(\boldsymbol{r}_2 - \boldsymbol{r}_1)}{|\boldsymbol{r}_2 - \boldsymbol{r}_1|^3}$$

Net force on 1

$$F_{12} = m_1\ddot{\boldsymbol{r}}_1$$



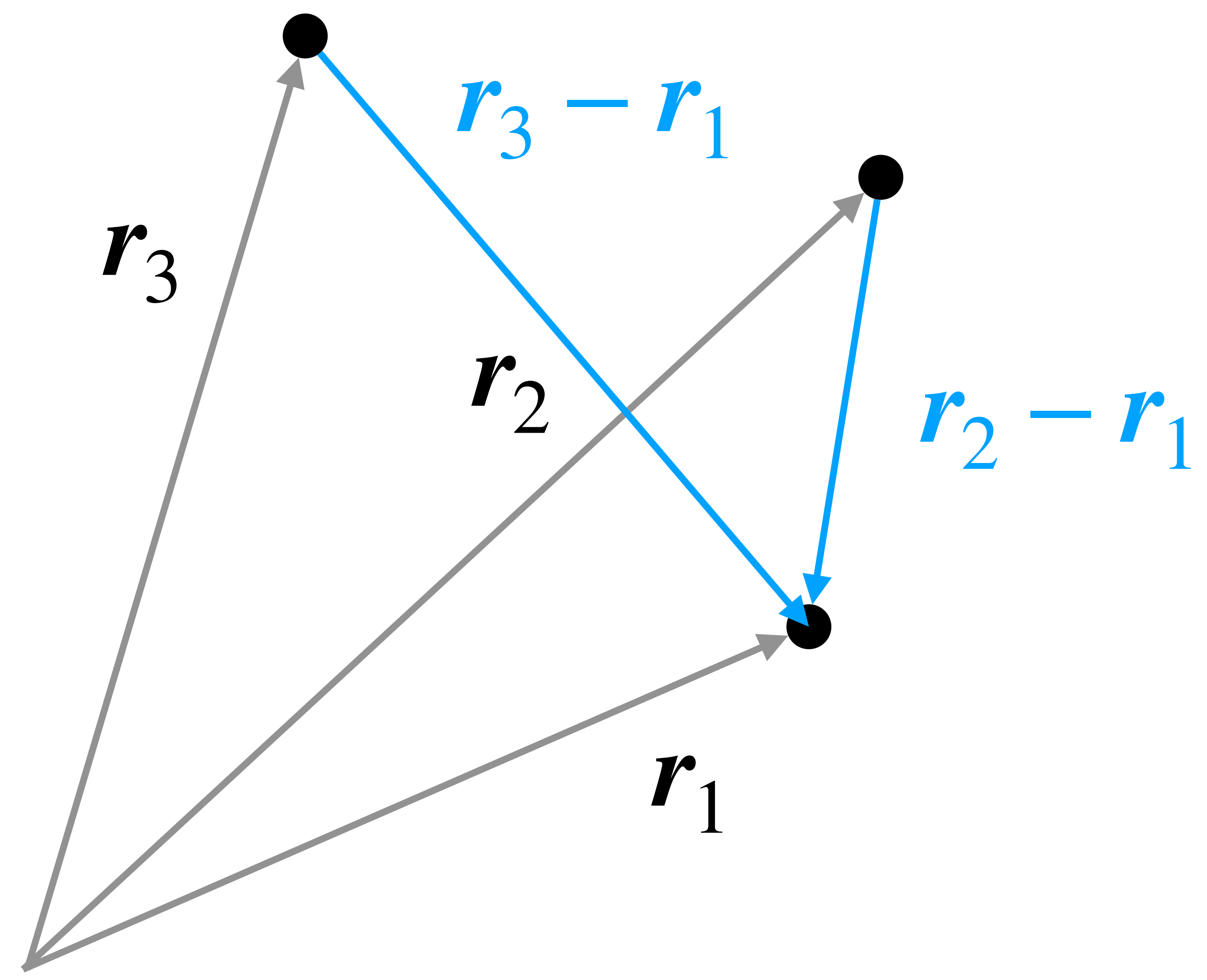
N-body problem

- With 3 bodies, the net force on 1 is

$$\sum_{j=1,\, j\neq 1}^{3} F_{1j} = \sum_{j=1,\, j\neq 1}^{3} \frac{G m_1 m_j (\boldsymbol{r}_j - \boldsymbol{r}_1)}{|\boldsymbol{r}_j - \boldsymbol{r}_1|^3} = m_1 \ddot{\boldsymbol{r}}_1$$

- For N bodies, the equation of motion is

$$\ddot{\boldsymbol{r}}_i = \sum_{j=1,\, j\neq i}^{N} \frac{G m_j (\boldsymbol{r}_j - \boldsymbol{r}_i)}{|\boldsymbol{r}_j - \boldsymbol{r}_i|^3}$$



N-body computational challenge

- For N bodies, the equation of motion is

$$\ddot{\boldsymbol{r}}_i = \sum_{j=1,\, j\neq i}^{N} \frac{G m_j (\boldsymbol{r}_j - \boldsymbol{r}_i)}{|\boldsymbol{r}_j - \boldsymbol{r}_i|^3} \quad \sim \quad \mathcal{O}(N^2) \text{ computations per time step}$$

- If galaxy is $10^{10} - 10^{11}$ points, need $10^{20} - 10^{22}$ FLOP per time step
- Top DOE HPC are just now approaching 1 exaFLOP = 10^{18} FLOP
- Depending on number of time steps, naive simulation would take years!
- Need physical insight

Physical insight

- Lump together 10^6 points and treat lump as a “super point” in Newton’s equations
- Justification will come from collisionless Boltzman equation (CBE)
- Mean field theory trick: Consider galaxy as $f(\boldsymbol{r}, \boldsymbol{v}, t)$ phase space density
- Plummer sphere (assignment 3)
- Treecode: $\mathcal{O}(N^2) \rightarrow \mathcal{O}(N \log N)$
- GyrfalcON: $\mathcal{O}(N \log N) \rightarrow \mathcal{O}(N)$

N-body symmetries

- N-body systems obey conservation laws (simulation should respect these!)

- Noether's theorem states that each *continuous symmetry* of the Lagrangian L gives rise to a corresponding conservation law

- A continuous symmetry of the Lagrangian $L = K - V$ is a transformation that leaves the physical system unchanged

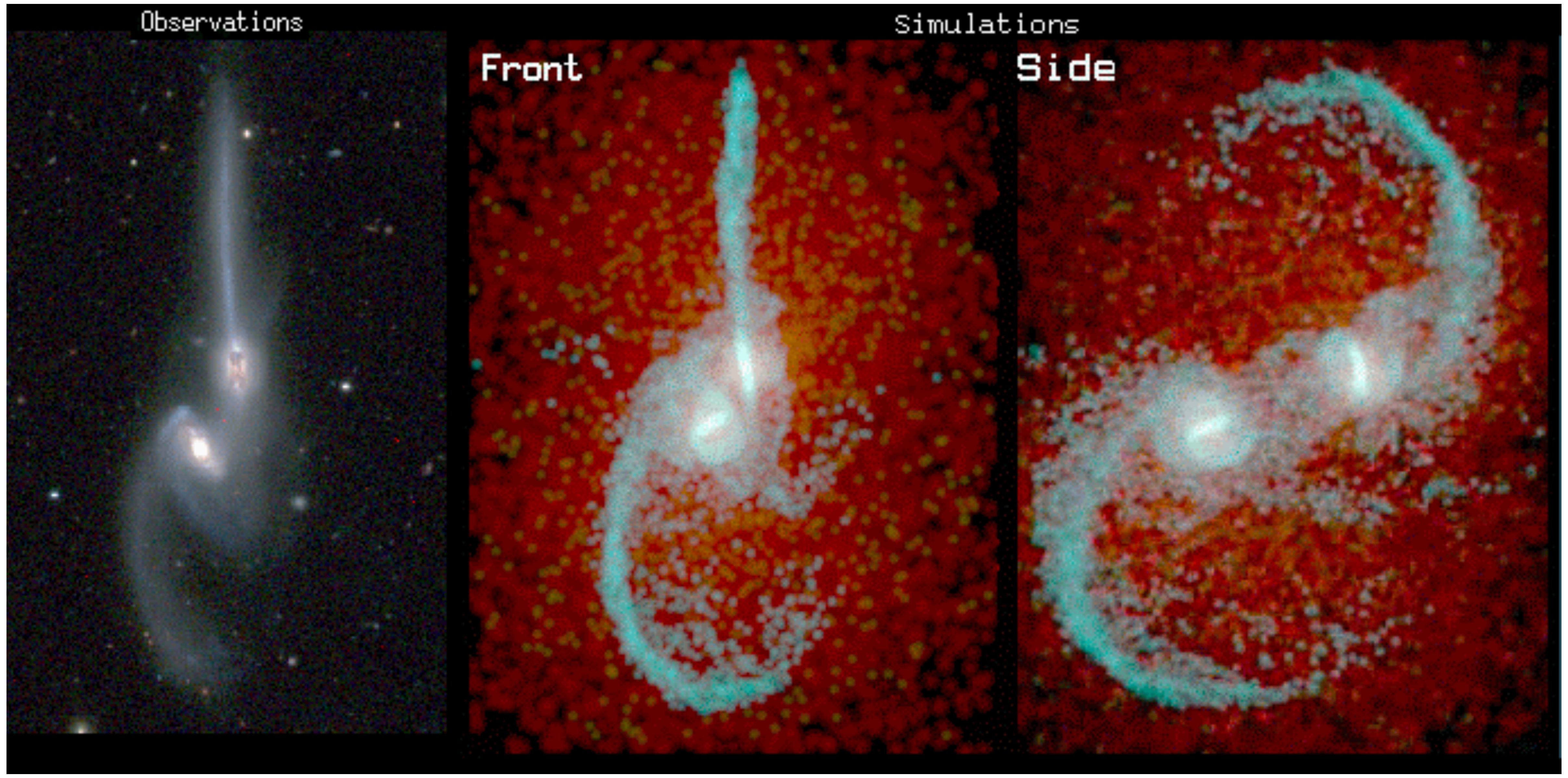
- Translation in time $t \to t + \Delta t$ is a symmetry because L is not an explicit function of time; consequently the total system energy $E = K + U$ is conserved

- Symmetry with respect to translation in space, $\boldsymbol{r} \to \boldsymbol{r} + \Delta \boldsymbol{r}$, implies conservation of total linear momentum

- Symmetry with respect to rotation implies conservation of total angular momentum

Galaxy collisions: Final project

- We will illustrate the power of solving the N-body equations with the simulation of two colliding galaxies
- At close encounter during the collision, the simulation is a close match to what is observed on the sky for galaxy N4676 which is also known as the Mice galaxies
- The observation on the sky is shown on the left and the simulation results are shown on the right:

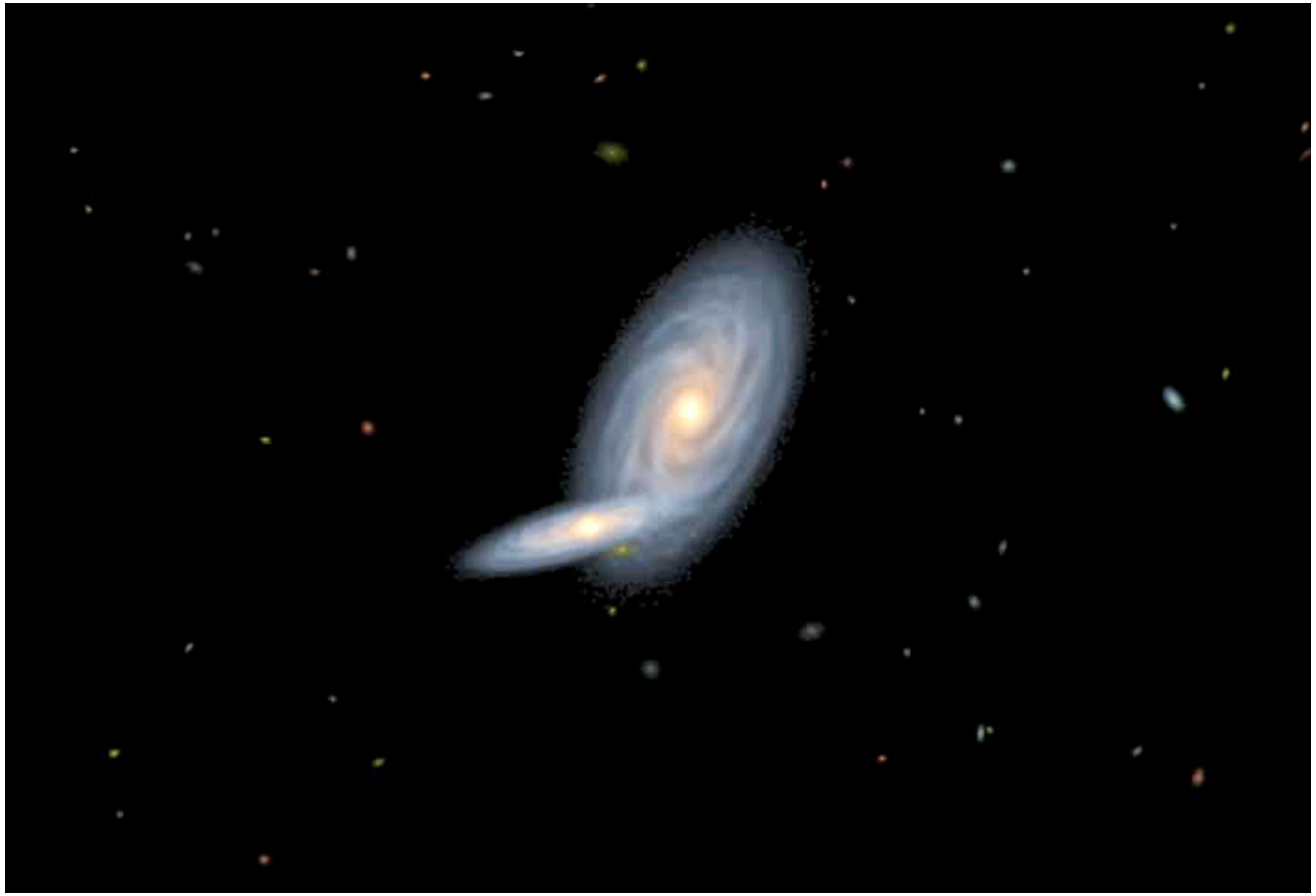


Algorithm and software

- The $6N$ coupled ordinary differential equations have to be solved numerically using finite discretization of the time variable
- Variable discretization schemes are used in the literature, like the leapfrog algorithm, Euler algorithm, or the Runge-Kutta method
- The force calculation is also a challenging problem
  - A naive approach will require $\mathcal{O}(N^2)$ calculations
  - Sophisticated procedures, like treecode reduce the number of operations to $\mathcal{O}(N \log N)$
- We will primarily use C/C++ and Python
- We encourage students to use C/C++ or accelerated Python, e.g. Numba

Analysis & visualization

- Example of N-body simulation (Mice)



Final projects

- Mice
- **Antennae**
- **Cartwheel**