PHYS 6260 — Computational Physics (with Python)

Prof. John Wise

Logistics

- **Instructor:** Prof. John Wise, Office: Boggs 1-90D,
  - Email: jwise@gatech.edu
  - **Office Hours:** Thurs 1-2pm
- Bring your laptop to every class
- Today we will use it to install Python and the necessary software stack on your laptop
- **Teaching Assistant:** Matteo Reynoso
  - **Office hours:** TBA
  - Email: mreynoso@gatech.edu

How to reach me

- In-person office hours: Thurs 1-2
- MS Teams chat
  - Faster: Mateo and I will answer your questions as soon as we see them
  - Efficient: Many students have the same concerns and questions
  - DM me if you want privacy
- E-mail: jwise@gatech.edu

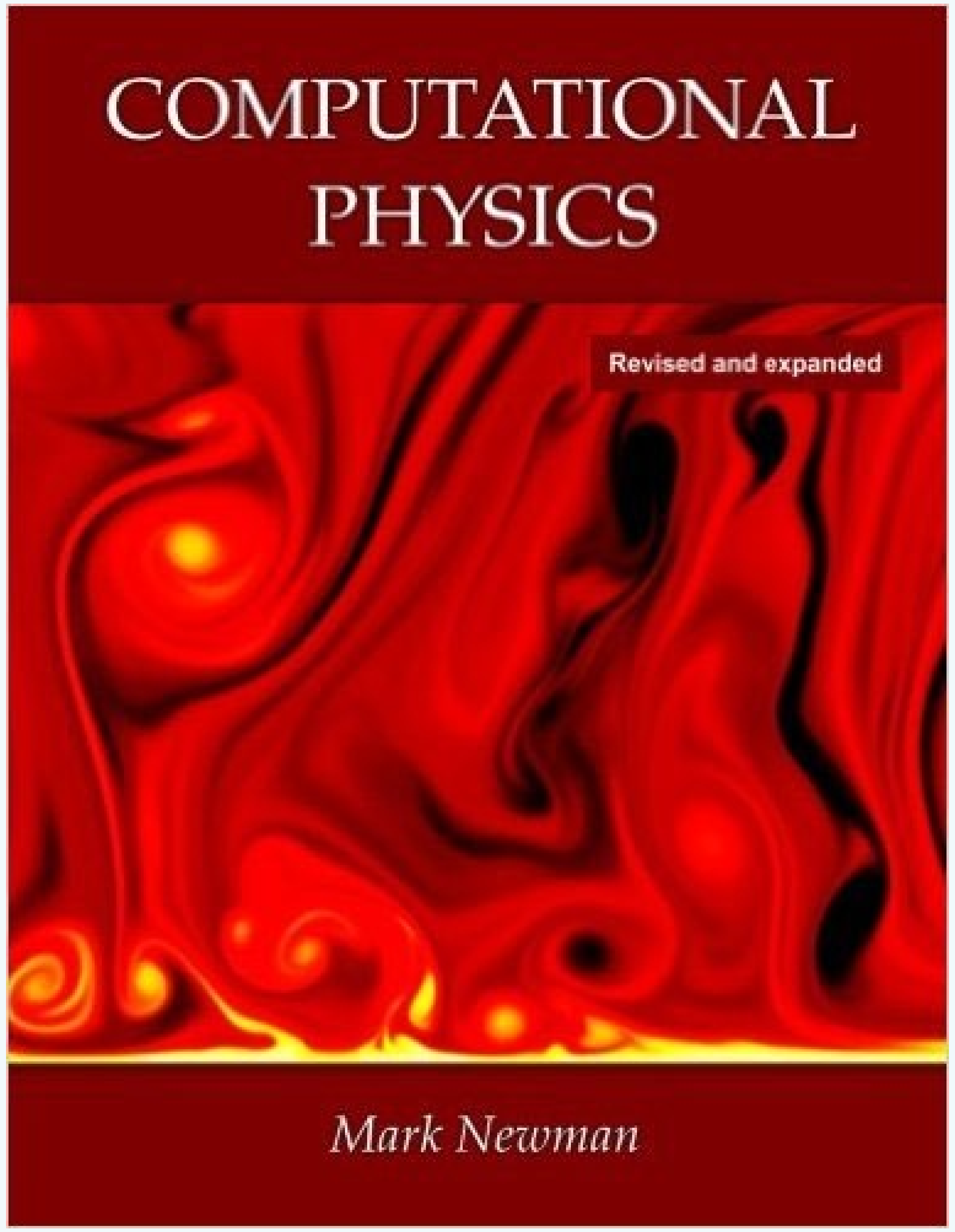
Textbooks

- ***First 4 weeks:*** *Computational Physics* (2013, “revised and expanded”) by M. Newman

- We will cover the highlights throughout the book.

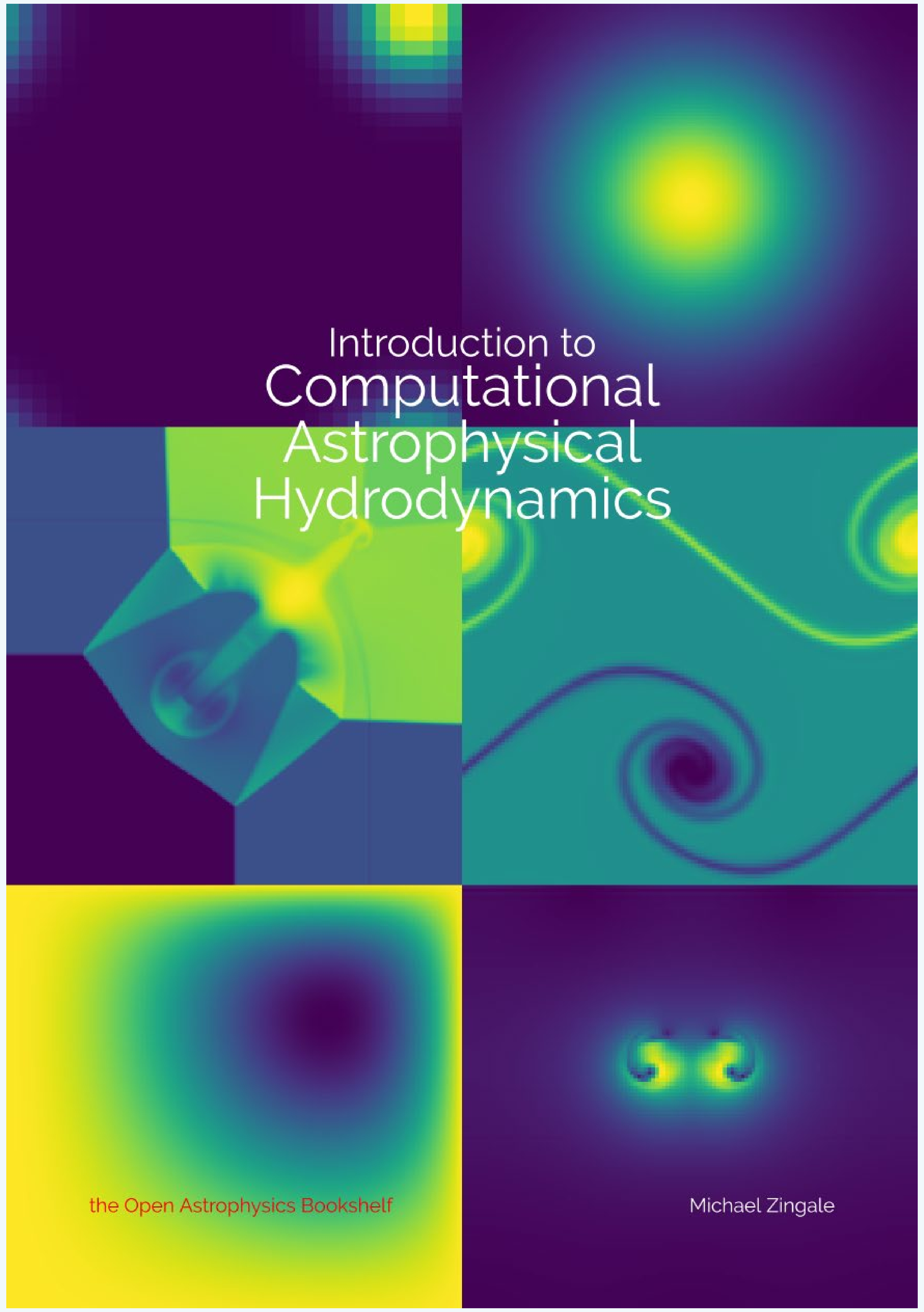
- All lecture notes (sometimes in the form of notebooks) and recording will be posted on Canvas after class.

- Slides will be posted before class, so you can write notes on them.



Textbooks

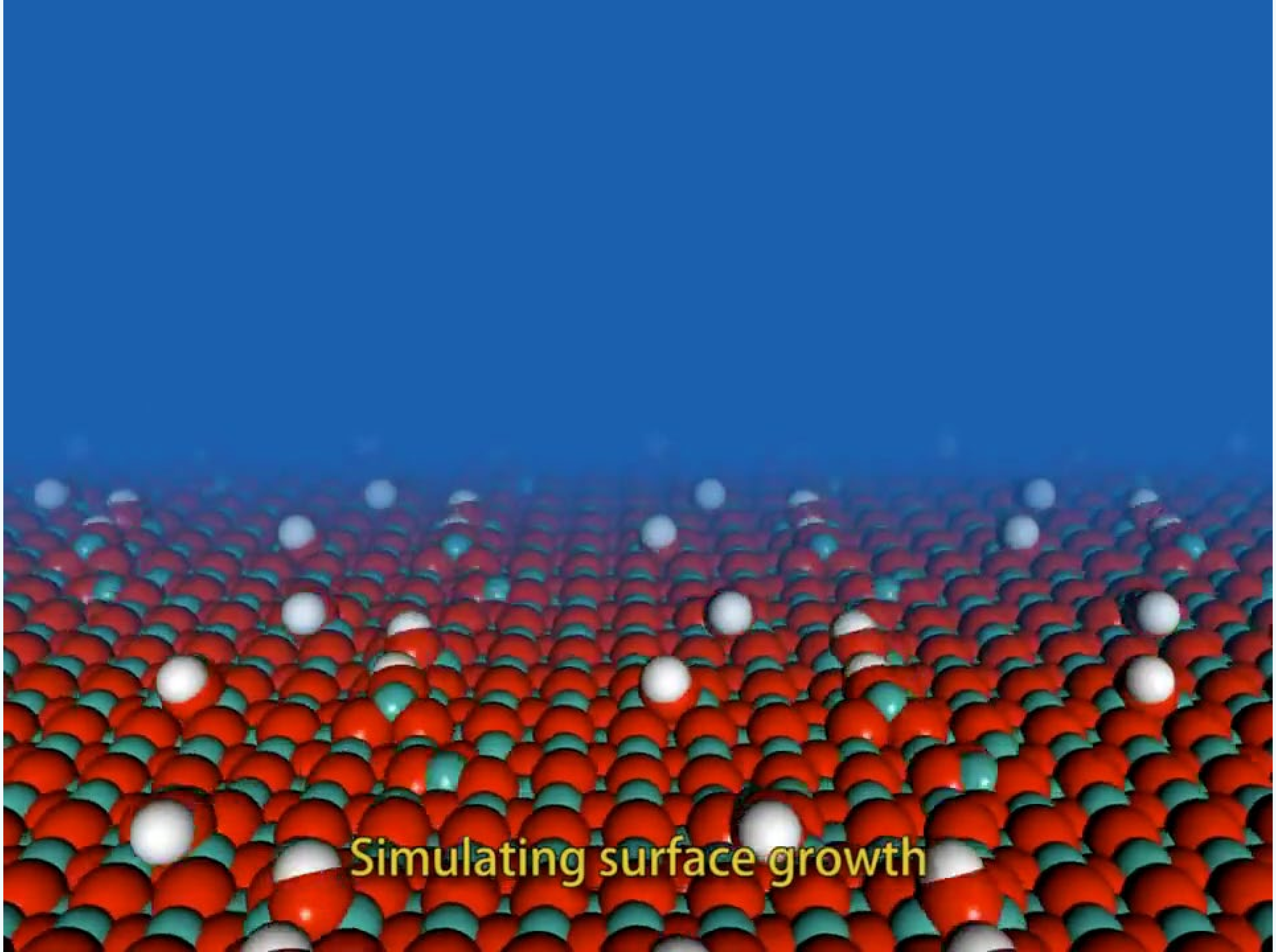
- : *Computational Hydrodynamics for A***Hydrodynamics module***strophysics* by M. Zingale (github; PDF)
- All other material will come my own lecture notes.
- Optional to get up to speed with Python: CodeAcademy



Course Goals

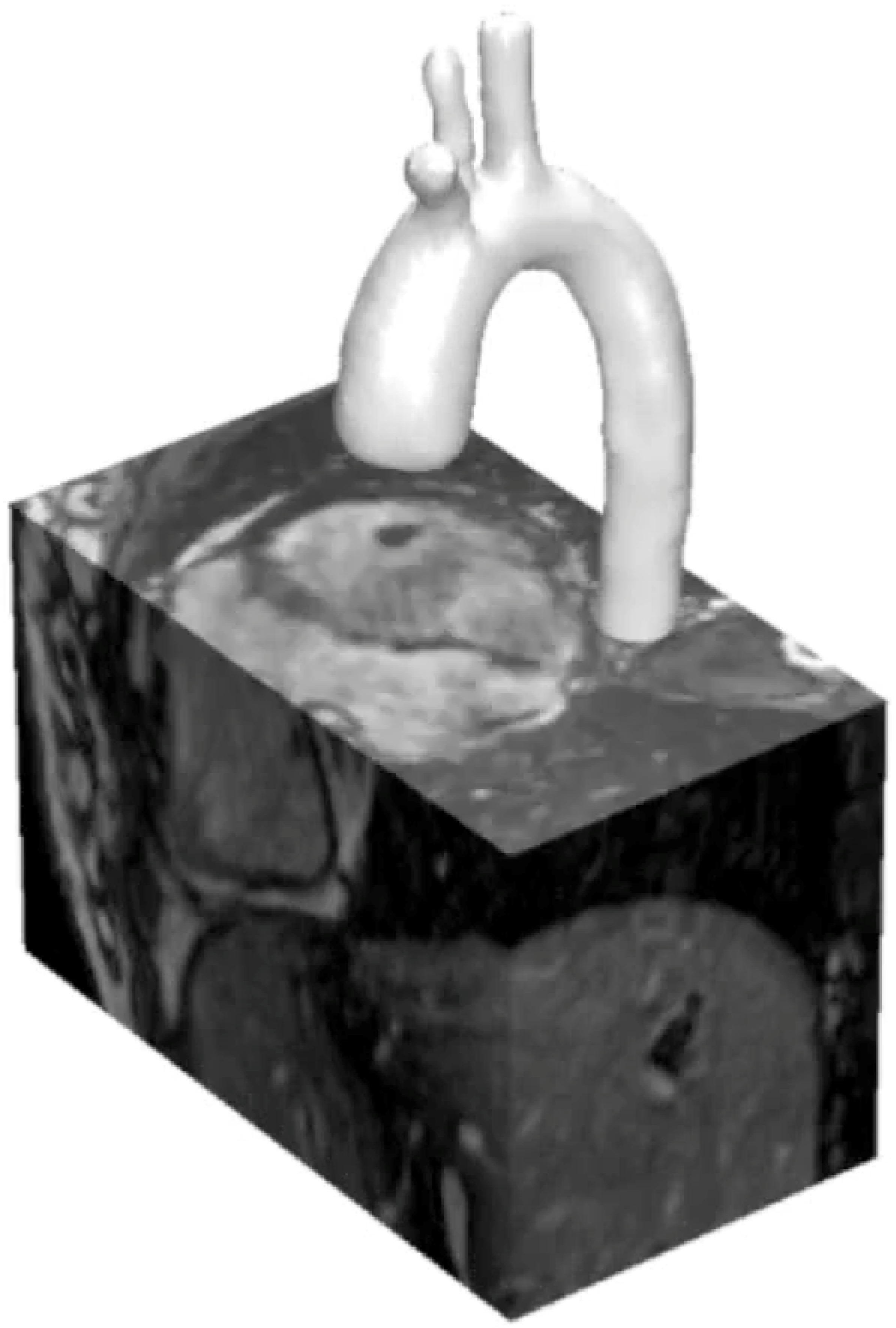
- The purpose of this course is to introduce various numerical methods that are used in solving physics problems
- Class time will be dedicated to covering theoretical aspects of a numerical method and discussing its potential applications
- The homework sets will apply these methods to physics problems
- While the class is not a programming class, you will be exposed to modern programming techniques and expected to reproduce them in exercises and projects
- Example applications include topics such as quantum chromodynamics, classical mechanics, hydrodynamics, astrophysics, biophysics, and material science.

Quantum Chromodynamics
QCD


Simulating surface growth


16777216 Neurons - 110532997 Compartments - 3968179211 Synapses [PAUSED]
Simulation time: 0.0 ms (Real time factor: 0.0000x recent, 0.0000x avg.)
0.00 FPS (Avg: 0.00) :: Frame render time: 700.78 Mcycles (avg: 645.27 Mcycles)
mPSPs (Thalamus): ON
Retinal Ganglion Cell Activity:

Patient-specific aortic segmentation from MRI data




BIG
DATA

Class Modules

1. Introduction to Python, Jupyter notebooks, and GitHub

2. Introduction to Numerical Methods
   a. Differentiation and integration
   b. ODEs
   c. PDEs
   d. Monte Carlo methods

3. Parallel programming: MPI, Accelerators, HPC

4. Machine Learning: Supervised learning, Deep Learning

5. Applications
   a. Elliptic solvers
   b. Transport phenomena (radiation, neutrinos, etc.)
   c. Molecular dynamics
   d. Fluid dynamics

Course Grades

- The course grade will be entirely determined from your scores on the homework sets, team project, and class participation.
- There will be no tests or final exam as this class is project-based.
- Rounding to the nearest tenth:
  A = >89.5%, B = 79.5-89.4%,
  C = 69.5-79.4%, D = 59.5-69.4%,
  F = <59.4%

- Course grade division:
  - Homework: 50%
  - Term Project: 45%
  - Participation: 5%

Homework (50%)

- There will be seven homework sets during the semester.
- Homework problems will require writing computer programs in Jupyter notebooks or python scripts based on the numerical algorithms discussed in class.
- In addition, there will be one free response question on concepts discussed during the lectures.
- Programs must be written completely from scratch, with the essential steps fully commented in the notebook markup language.
- However, the structure of the program can be based, if necessary, on programs written or discussed by the instructor.

Homework (50%)

- Students are encouraged to work and discuss problems together, but the programming and written work must be your own.
- The instructor reserves the right to request the student to reproduce the results submitted in homework assignments.
- Delays in the submission of the assignment will be penalized 10% per day up to a maximum of 3 days late.
- All homework submissions are encouraged to be in the form of a Jupyter notebook.
- However, for topics like parallel programming and accelerators, python scripts and associated files are acceptable.

Term Project (45%)

- You will formulate, solve, and present a class project. Either an individual or a group project.
- This is an open-ended project in which the teams will pose a research question, devise a plan, and write a program to explore the system.
- Students are free to choose a particular field of physics -- astrophysics, physics of living systems, condensed matter, non-linear dynamics, etc.
- Students must consult with the instructor on the viability and scope of the question before finalizing it.

Term Project (35%)

- Proposal due: **Friday, February 28** (10% of grade)
  - Students must finalize their research question with the instructor before proposal due date
- Progress report due: **Friday, March 28** (15% of grade)
- Poster presentations: **Monday/Wednesday, April 14/16** (25% of grade)
- Final report due: **Tuesday, April 29** (50% of grade)

Class Participation (5%)

- The lecture periods will involve in-lecture problems and associated discussions. *Please have your Jupyter environment ready to go.*
- Every class period, one or two students will present the results of their in-class assignment.
- Your class participation will be computed based on your completion of an in-class presentation and your engagement during such assignments.

Installing Python (and associated development environments)


PYTHON!
YOU'RE FLYING!
HOW?

Installation of a Linux-like environment

Windows

- It's possible to install a Linux distribution natively in Windows with the Windows Subsystem for Linux (WSL). Be sure to use WSL2.
- Follow these instructions
- My distribution preference: Ubuntu (24.04 LTS)
- My terminal preference: Windows Terminal
- X Windows Server (to open Linux GUIs): VcXsrv

Installation of a Linux-like environment

macOS

- macOS has its root in BSD Unix and has a native terminal (spotlight: Terminal)
- You will have to install a Linux-like package manager and X Windows Server
- Package manager: Homebrew
- X Windows Server: Xquartz

Other options: Dual booting and Virtual Machines (personally I don’t like VMs for research)

Installing software dependencies

- When initializing a new system, I usually just install packages as-needed to reduce bloat

- You can use either “apt search …” in Ubuntu (or Debian-based systems) or “brew search” in macOS

- Python: You will need to install a Python environment that’s suitable for scientific computation for this class

- There are a few package managers (e.g. Anaconda, pip, miniconda) for Python that can create conflicting installations

Installing software dependencies

- My advice is to pick one and keep with it.
  - In the past couple of years, Anaconda has gotten better about keeping track of packages installed with pip and conda.
- If you already use one, just install packages with your chosen manager.
- Personally, I use pip. To install in Ubuntu, “sudo apt install python-pip”
- Useful packages to begin: **numpy**, **scipy, matplotlib**, **Jupyter**, Jupyter-lab, unyt (for later modules: mpi4py, pycuda, scikit-learn)

Development Environments for Python

- Plain text editor and running scripts in the terminal
  - Please don't use Notepad, Wordpad, or TextEdit
  - Suggestions: Sublime Text, Atom, vim, emacs, **VScode** (my preference)
- Web-based editing and execution: Jupyter notebook or JupyterLab (my preference)
  - Test Jupyter out now here!
- Integrated Development Environments (IDE)
  - Spyder seems to be popular
  - https://wiki.python.org/moin/IntegratedDevelopmentEnvironments
  - http://noeticforce.com/best-python-ide-for-programmers-windows-and-mac

OVERVIEW
SETUP
Overview
Linux
macOS
Windows
Raspberry Pi
Network
Additional Components
Enterprise
Uninstall
GET STARTED
USER GUIDE
SOURCE CONTROL
TERMINAL
LANGUAGES
NODE.JS / JAVASCRIPT

Edit

Setting up Visual Studio Code

Getting up and running with Visual Studio Code is quick and easy. It is a small download so you can install in a matter of minutes and give VS Code a try.

Cross platform

VS Code is a free code editor, which runs on the macOS, Linux, and Windows operating systems.

Follow the platform-specific guides below:

- macOS
- Linux
- Windows

VS Code is lightweight and should run on most available hardware and platform versions. You can review the System Requirements to check if your computer configuration is supported.

Update cadence

IN THIS ARTICLE
Cross platform
Update cadence
Insiders nightly build
Portable mode
Additional components
Extensions
Next steps
Common questions

www.anaconda.com/products/individual

Individual Edition

Your data science toolkit

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Download

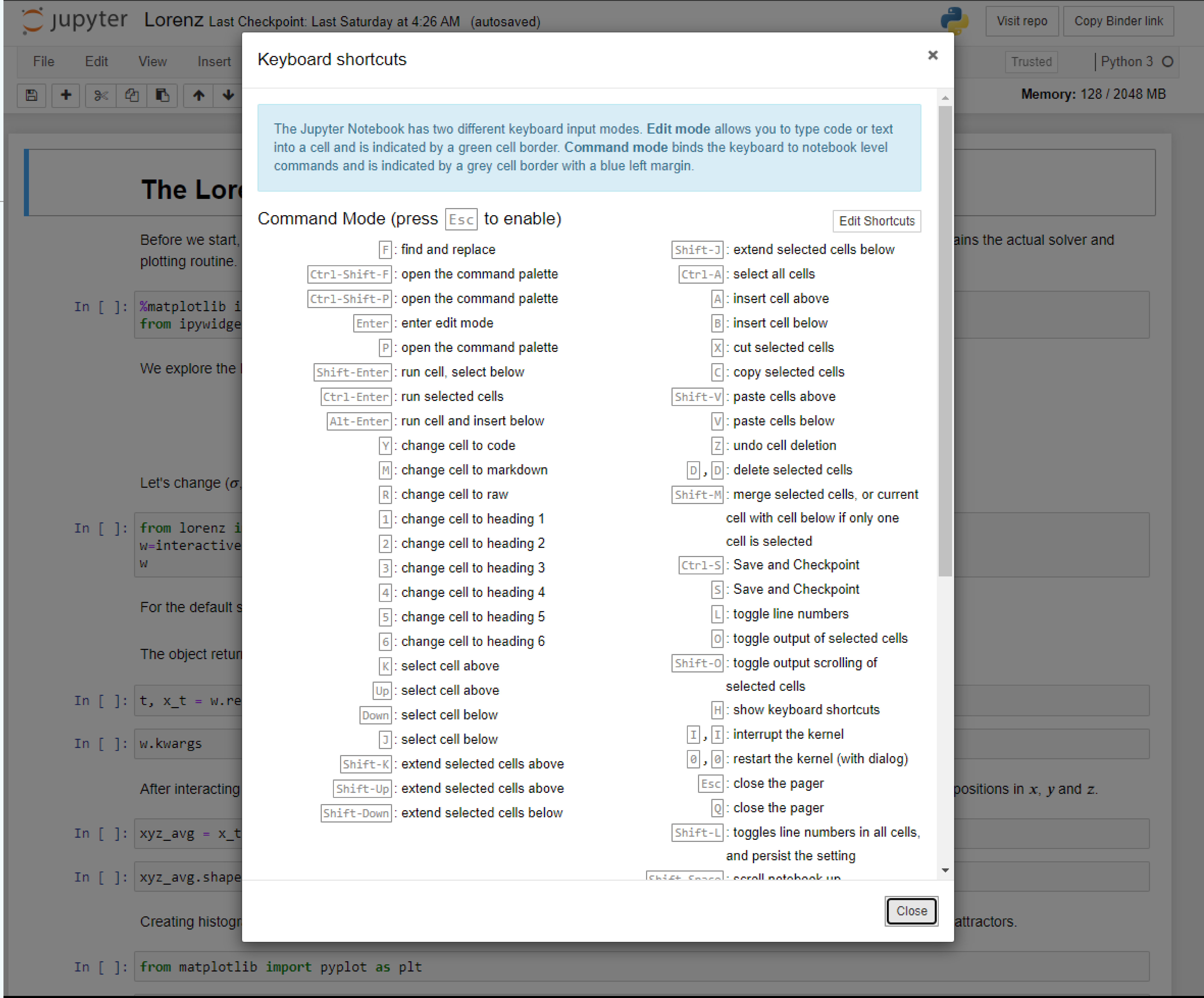
Open Source

Conda Packages

Manage Environments

Keyboard shortcuts are your friends!

- Your fingers never have to move to the trackpad, mouse, or screen.
- Once memorized, your productivity can skyrocket as you fly across the screen.



Configuring Access to GT Compute Cluster

Access to PACE (GT Compute Cluster)

- Everyone should have access to the PACE Instructional Compute Environment (PACE-ICE)
    - User's guide & Orientation slides
- Useful for problem sets with parallel computing and accelerators (GPUs)
- The primary benefit will come for your projects if you need more computing power and/or memory

ICE Compute Nodes

- 15GB of backed up storage
- Scratch storage (100 GB) is not backed up
- Access to over 100 computational nodes

Quantity	CPU	Memory	GPU	Local Scratch
30	Dual Xeon Gold 6226 (24 cores/node, 2.70 GHz)	192GB DDR4 2933 MHz		1.6TB NVMe SSD
22	Dual Xeon Gold 6226 (24 cores/node, 2.70 GHz)	192GB DDR4 2933 MHz		1.9TB NVMe SSD
1	Dual Xeon Gold 6226 (24 cores/node, 2.70 GHz)	384GB DDR4 2933 MHz		8TB SAS HDD RAID
5	Dual Xeon Gold 6226 (24 cores/node, 2.70 GHz)	768GB DDR4 2933 MHz		1.6TB NVMe SSD
2	Dual Xeon Gold 6226 (24 cores/node, 2.70 GHz)	768GB DDR4 2933 MHz		1.9TB NVMe SSD
4	Dual Xeon Gold 6248 (40 cores/node, 2.50 GHz)	192GB DDR4 2933 MHz	1x Tesla V100 PCIe 32GB	512GB SATA SSD
4	Dual Xeon Gold 6248 (40 cores/node, 2.50 GHz)	192GB DDR4 2933 MHz	4x Tesla V100 PCIe 32GB	512GB SATA SSD
6	Dual Xeon Gold 6226 (24 cores/node, 2.70 GHz)	192GB DDR4 2933 MHz	4x Quadro Pro RTX6000 24GB	1.6TB NVMe SSD
4	Dual Xeon Gold 6226 (24 cores/node, 2.70 GHz)	384GB DDR4 2933 MHz	4x Quadro Pro RTX6000 24GB	1.9TB NVMe SSD

PACE Cluster Documentation
Home
Getting Started
FAQs
Storage and File Transfer
Scheduler
Hive Cluster Documentation
Phoenix Cluster Documentation
Firebird Cluster Documentation
Buzzard (OSG) Cluster Documentation
ICE Cluster Documentation
Open OnDemand
Open OnDemand Guide
Firebird Software
Phoenix, Hive, and ICE Software
Usage Metrics
Troubleshooting your Jobs

Updated 2023-05-08

Open OnDemand Instances for PACE's Clusters

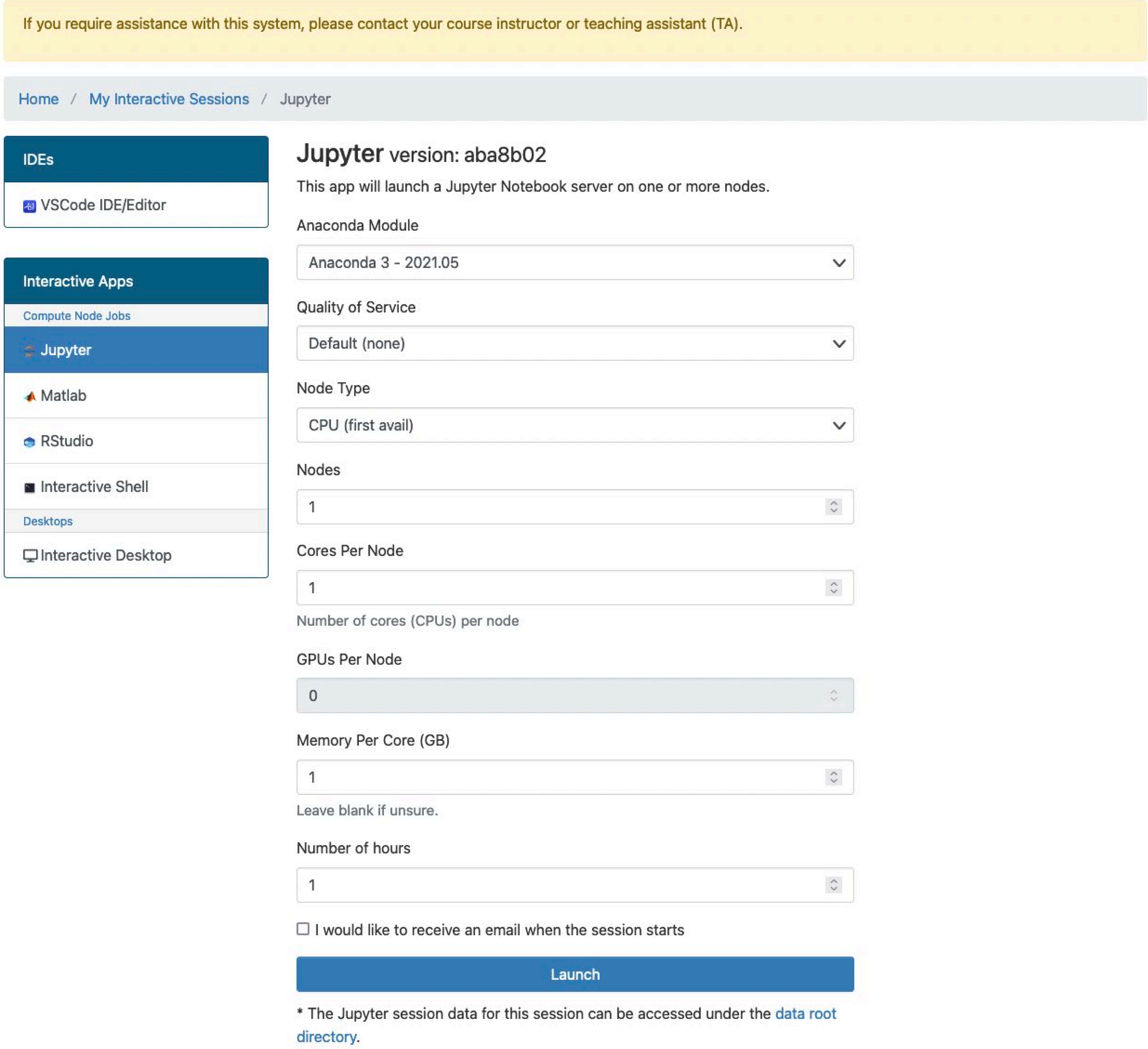
Cluster	Link
Phoenix	Phoenix OnDemand
Hive	Hive OnDemand
ICE	ICE OnDemand

★ Must be connected to GT VPN (even if on eduroam)

Warning

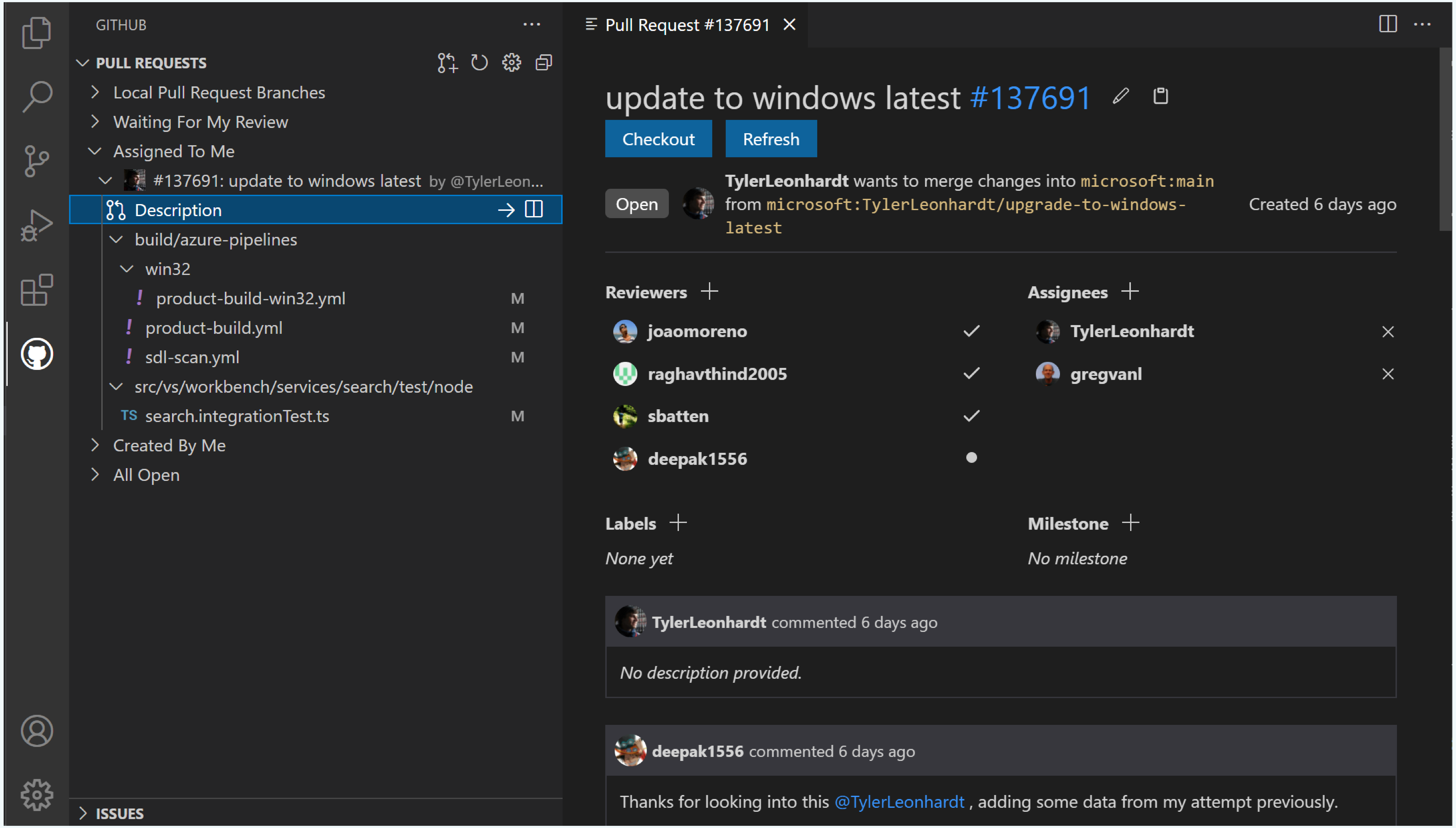
OnDemand does not have IE11 support and does not work well with Safari. Use Chrome, Firefox, or Edge for best performance.

Introduction



- Can launch an interactive environment for Juypter notebooks through PACE ICE on-demand
- You may already have access to Phoenix or Hive through your research group
- For this class, you can simply use ICE for all PACE calculations (does not require any payment)

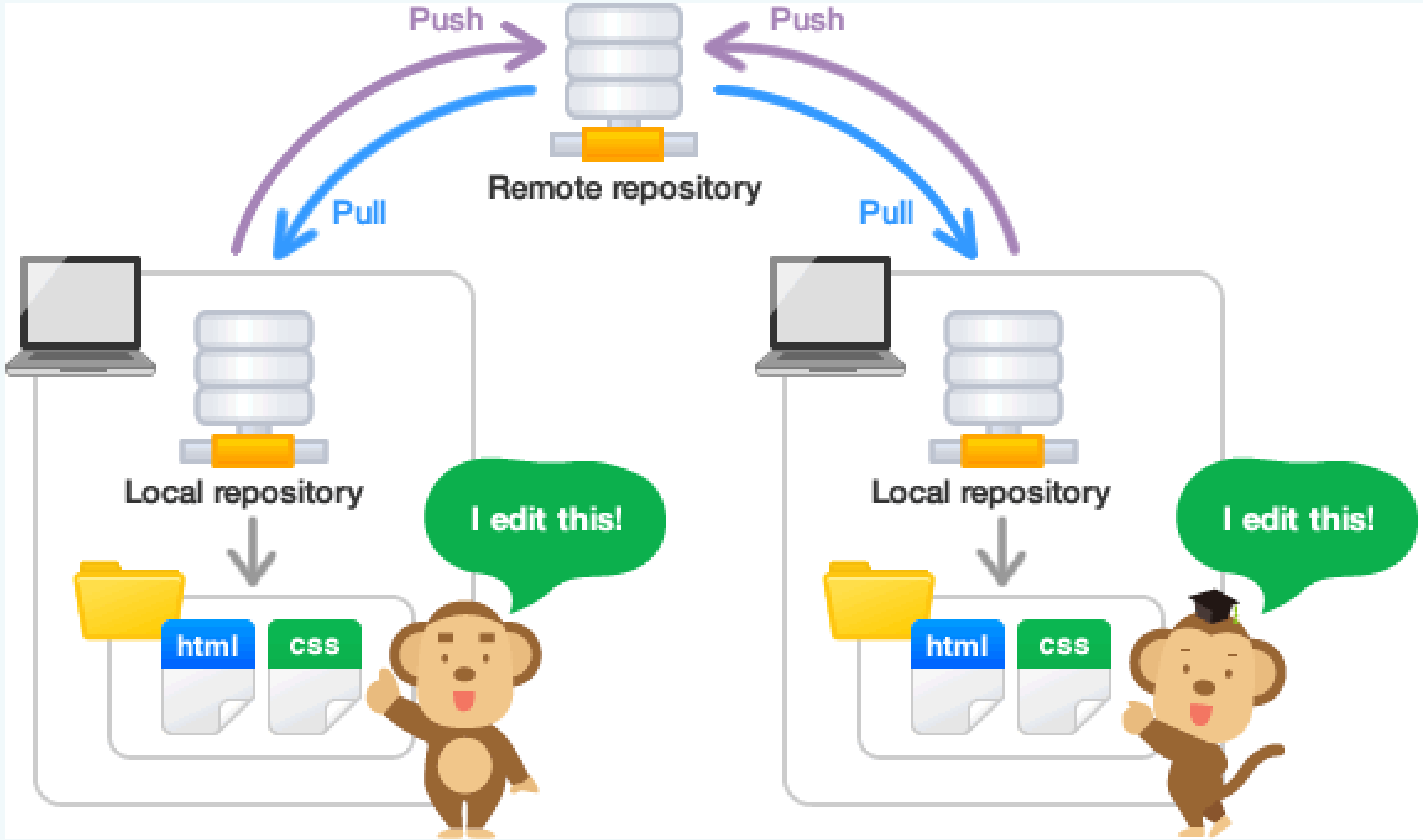
VS Code: Easy to use with PACE & Git



Required use of Github for this course

Distributed version control systems (git)

- You will be required to submit your homework through GitHub Classroom
- You will need to become familiar with git for your homework, and I highly encourage you use it for your project
- This is a distributed version control system
  - There exists a remote server (github.com)
  - These repositories can be cloned onto any machine



For next class ---

- Download the following:
  - VS Code
  - Python environment of your choice (e.g. Jupyter notebook, VS Jupyter extension, spider, etc.)
- We will have an interactive class where we will cover the basics of github and python