ASTR 288P
UNIX and Scientific Programming with Applications in Astrophysics

Course Outline
Fall 2018

Constantinos Kalapotharakos
UMCP, NASA GSFC

University of Maryland

About Me

Constantinos Kalapotharakos
constantinos.kalapotharakos@nasa.gov
ckalapotharakos@gmail.com
ckalapotharakos@astro.umd.edu

Use these two (both)

Based at NASA GSFC (so regularly off campus)

You can meet me:
ATL 1243, Fri 1-2pm (before class)

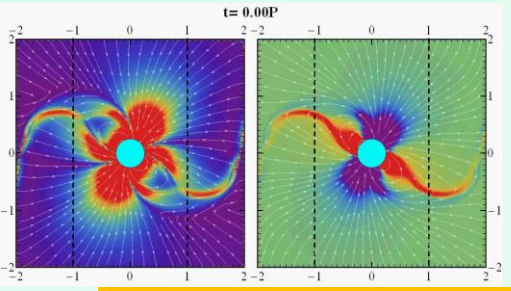
- **Always Confirm**
- **Additional times can be set up by email if needed**

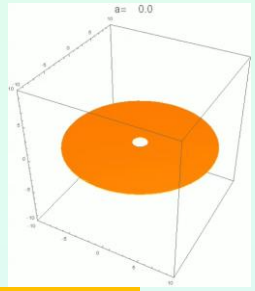
Course Website
https://github.com/ckalapot/astr288p_fall_2018

About Me

High Energy Astrophysics
(Pulsars, Compact Objects)

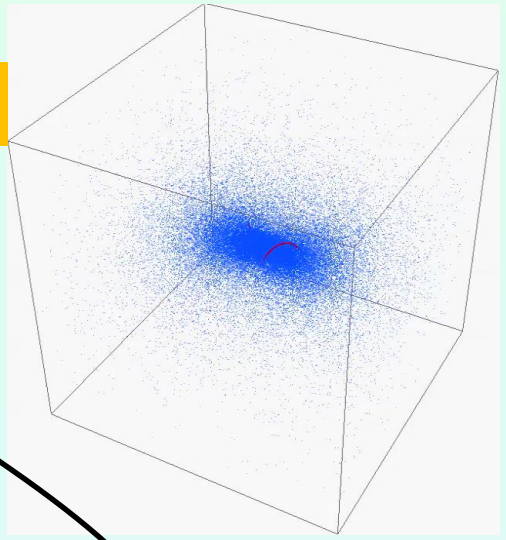
Galactic Dynamics
(Chaos, Secular Evolution)



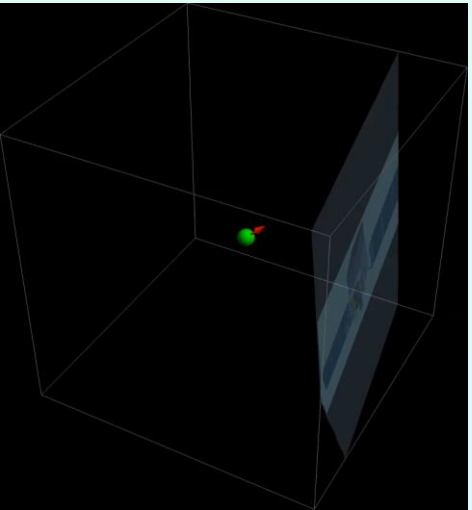


MHD Force-Free code

N-body code



Developer



Computational Astrophysicist

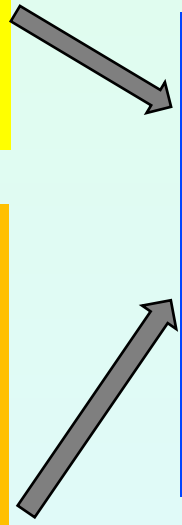
Fortran, C, MPI
Mathematica, Python

Particle-in-cell (PIC) code (C-3PA)

Syllabus

Students will be introduced to the **UNIX environment** and especially to scientific programming (mainly in **Python**). Students will learn how to operate within a UNIX environment (e.g. Linux, macOS), **run commands**, and write **scripts** in **Bash shell**.

Students will learn the fundamentals of computer programming while emphasis will be given to **numerical methods, visualization techniques, and astrophysical applications**. Students will also be made comfortable installing and using **open source** software, which includes collaborating and sharing their own code.



After successfully completing this course you will be able to:

- **Work within UNIX environment**
- **Write shell scripts**
- **Code in Python**
- **Analyze and visualize scientific data**

What will be covered

- **Unix**
  - Shells (e.g bash)
  - File system (/, /etc, $HOME)
  - Basic commands and shell scripting
  - Editors (e.g. emacs, atom, VSCode)
  - Tools (e.g. git, c and fortran compilers -> open source software)

- **Scientific Programming** (python, ipython, jupyter)
  - Fundamentals of programming (e.g. variables, lists, arrays, structures, functions)
  - Numerical methods (e.g. root finding, fitting, numerical integration, integration of differential equations)
  - Data analysis
  - Data visualization (plotting, 2D, 3D, animations)
  - (Astro)physical applications
  - (Astro)physical project

Course Format - Grades

Grades

- Lectures in almost every class
- In-class experimentation and exercises

10%

- Homework (due before each class through submission to ursa)

20%

- Formal assessments (~15mins) open notes, known dates (no surprises)

20%

- Project (among a list of available projects or one of your own) and presentation instead of a final written exam

50%

Course Format - Grades

Final Grade Cutoffs										
+	97.00%	+	87.00%	+	77.00%	+	67.00%			
A	94.00%	B	84.00%	C	74.00%	D	64.00%	F	<60.0%
-	90.00%	-	80.00%	-	70.00%	-	60.00%		

- Lectures in almost every class
- In-class experimentation and exercises

10%

- Homework (due before each class through submission to ursa)

20%

- Formal assessments (~15mins) open notes, known dates (no surprises)

20%

- Project (among a list of available projects or one of your own) and presentation instead of a final written exam

50%

Resources

- Not a specific textbook
- Internet, Google (plethora of information)
  Just a few…
  http://www.ee.surrey.ac.uk/Teaching/Unix/ (unix)
  http://www.stackoverflow.com (any kind)
  https://www.tutorialspoint.com (unix, python…)
  https://docs.python.org/3/ (python)
  https://www.scipy.org/index.html (python)
  https://www.scipy-lectures.org/intro/index.html (python)
- The course website will be updated regularly.

Hardware

- Lab machines
  - Master: ursa.astro.umd.edu
  - Nodes: lab001, lab002,…, lab013
  - Printer: labs.astro.umd.edu

- Your own machine (laptop, desktop)
  - Linux (Ubuntu, Redhat,…)
  - MacOS (Be aware to have Xquartz installed so “ssh -X” works!)
  - Windows (win10 bash and/or ubuntu).
    Putty or CygwinX to make ssh connections
    For X11, you need to install an X-server (e.g. Xming)

- Virtual machines (e.g. **virtualbox**, vmware)

Course Schedule (tentative)

		TOPIC
Fri 8/31	Lecture 1	Course overview and Introduction to Unix
Fri 9/7	Lecture 2	Unix basic commands, remote access, Git
Fri 9/14	Lecture 3	Unix shell scripting, Introduction to Python
Fri 9/21	Lecture 4	Python lists, arrays, tuples
Fri 9/28	Lecture 5	Python scientific programming, numerical methods, visualization
Fri 10/5	Lecture 6	Python scientific programming, numerical methods, visualization
Fri 10/12	Lecture 7	Python scientific programming, numerical methods, visualization
Fri 10/19	Lecture 8	Python scientific programming, numerical methods, visualization
Fri 10/26	Lecture 9	Python scientific programming, numerical methods, visualization
Fri 11/2	Lecture 10	Python, other languages, and open source software
Fri 11/9	Lecture 11	Python scientific programming, numerical methods, visualization
Fri 11/16	Lecture 12	Python scientific programming, numerical methods, visualization
Fri 11/23	Lecture 13	**No Class (Thanksgiving)**
Fri 11/30	Lecture 14	Python scientific programming, numerical methods, visualization
Fri 12/7	Lecture 15	Project Presentations