PYTHON FOR ASTROPHYSICS

Lecture 1

Wladimir E. Banda-Barragán
Universidad Yachay Tech

2025

Lecture 1 goals:

1. Tools for research in astronomy and astrophysics.

2. Why use free, open-source OS and languages?

3. Master Google Colab for Linux.

What do you need for the practicals?

- A PC/laptop with any OS.

- Internet access.

- A Google/gmail account.

- A GitHub account (desirable, not strictly needed).

A few words on Git

Git is an **Open Source Distributed Version Control System** for tracking changes in source code or any other set of software files.

- **Control System:** Git is a content tracker.
- **Version Control System:** Code is constantly changing. Many developers can add code in parallel. Keeps history of what changes have been implemented.
- **Branching/Forks:** Also, Git provides features like branches and merges.
- **Distributed Version Control System:** Git has a remote repository stored in a server and a local repository which is stored in the computer of each developer.

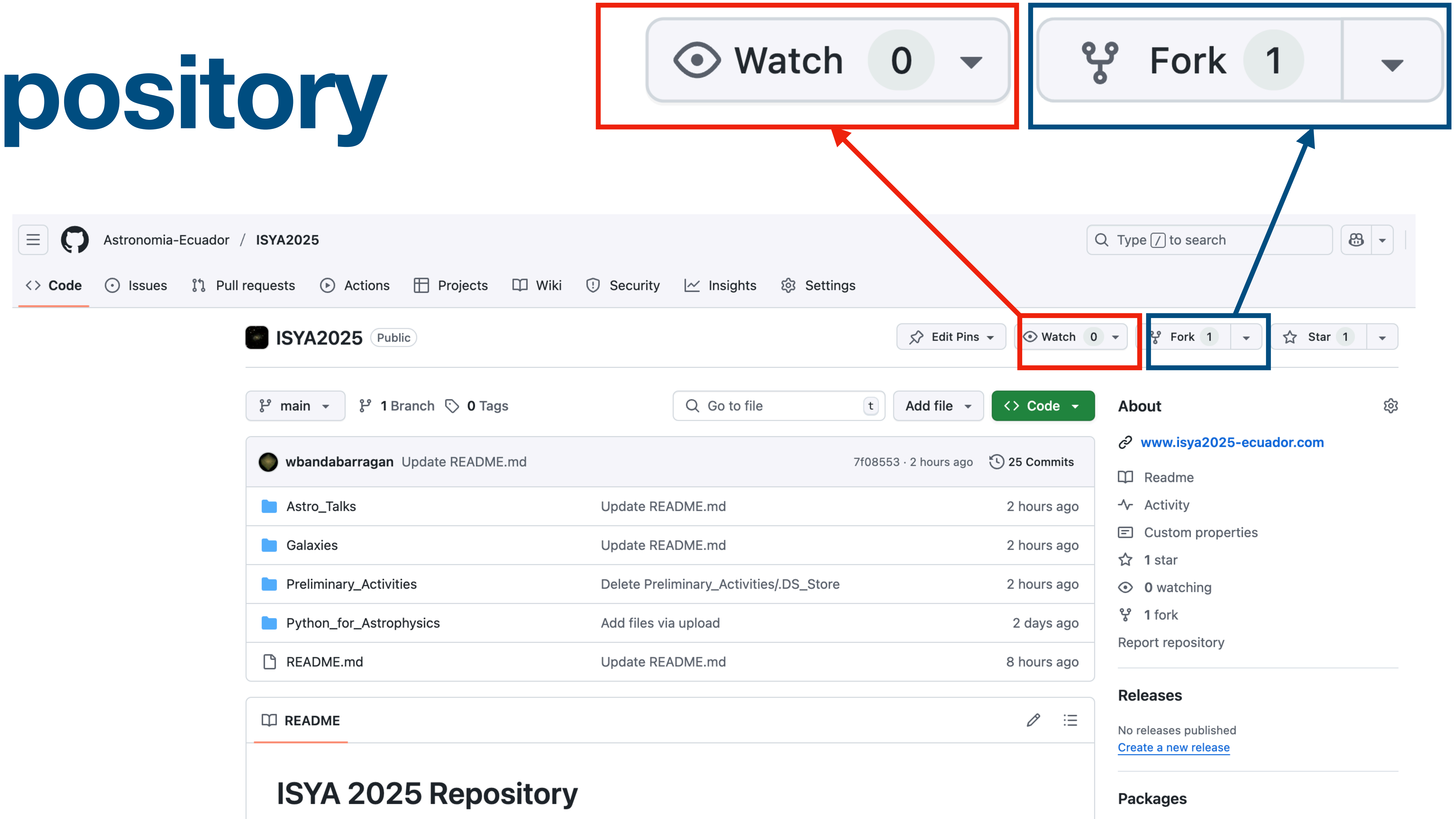
About GitHub

GitHub is a web-based platform that provides hosting for software development and version control using Git.

It is an ideal platform to share code with colleagues.

ISYA GitHub Repository

https://github.com/Astronomia-Ecuador/ISYA2025



Linux and the need for open source software

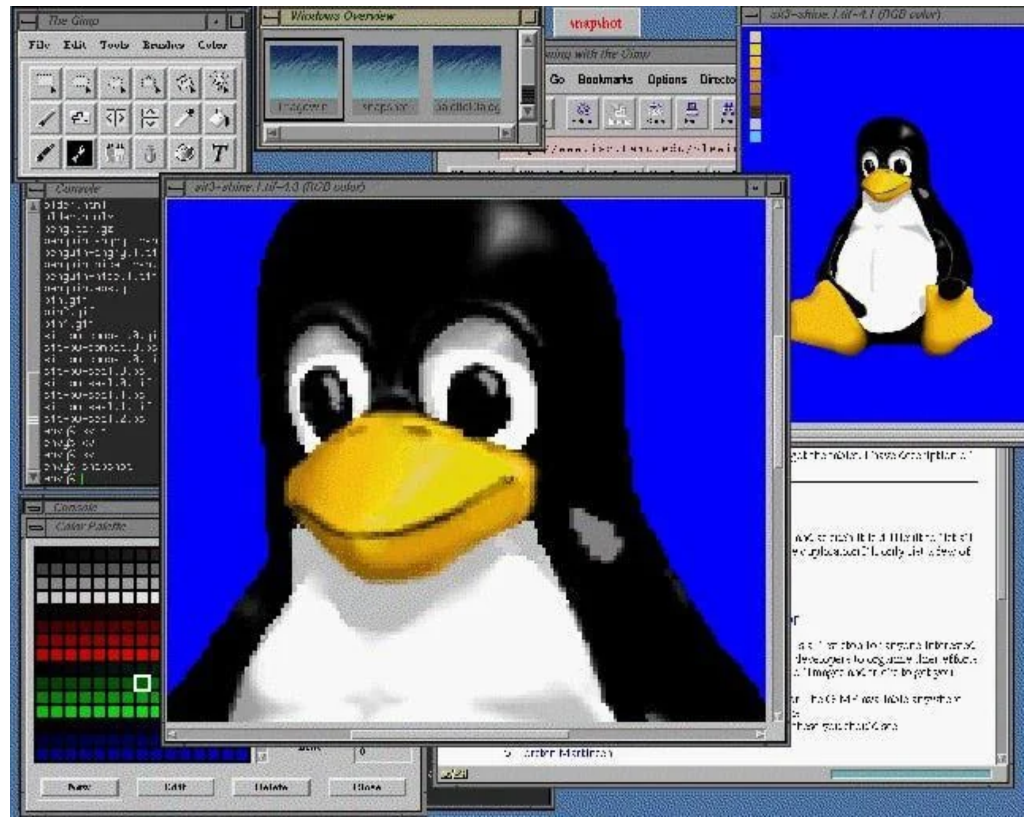
It is an operating system based on Unix, which was developed by Ken Thompson and Dennis Ritchie (at AT&T Bell Laboratories) during the 60's/70's.

Unix was highly portable, so it was adopted, copied, and modified by many companies and universities.

The source code was available, but modification and redistribution were restricted and its commercial version was too expensive.

Finnish student Linus Torvalds decides to create a new free operating system kernel called Linux.

Linux released its first version in 1991.



https://www.reddit.com/r/aggies/comments/vintc1/tamu_linux_released_in_1992_it_was_the_first/

Reference: https://www.greenteapress.com/thinkpython/thinkCSpy/html/chap01.html

Advantages of Linux OS

Linux OS are free and open source.

You can modify the source code and adapt it to your applications at will.

Linux provides security (much harder to hack, nearly no viruses).

https://www.geeksforgeeks.org/blogs/kde-vs-gnome/

Linux distributions come with an in-built platform to do programming.

There are two types of desktops:
KDE and GNOME.

Flavours of Linux OS

You have many options, the most popular ones in physics are:

1. Ubuntu
2. Fedora
3. Debian
4. CentOS

Linux can run on virtual machines / co-exist with other OS.

Linux is installed in (pretty much all) large-scale, high-performance supercomputers.

Linux is the OS of cloud servers (Google Colab).

The backend of GitHub relies heavily on **Linux**.

Basic Linux OS commands

1. man — offline manual, get help about any commands
2. which — find out where a command is defined.
3. <command> --help — Find help on any command
4. cd — Change the current directory (folder)
5. ls — List files in a directory
6. mkdir — Make/create a new directory
7. pwd — Print current directory
8. cp — Copy files and directories
9. rm — Delete files and directories
10. cat file.txt — see contents of file.
11. head file.txt — see the first 10 lines of a file
12. tail file.txt— see the last 10 lines of a file.
13. chmod — change permissions of a file or directory for 3 user groups: user (owner) permission, group permission, and other permission.
14. diff file1.txt file2.txt— show differences between two files
15. file — show the type of a file
16. less — browse the contents of a file, exit with q
17. locate — find files with names matching a pattern
18. touch — Create a new file or update an existing one
19. top — See what is going on, what processes are running, exit with q
20. ping server — check to see if a server is alive
21. df — show free disk space
22. du — show disk space usage
23. uname -a — information on Linux kernel
24. uptime — how long the system has been running
25. date — show current date/time

Thanks to modern coding tools, there is no need to install Linux /Python.

A browser and Colab is all we need!

Tutorial Time

Tutorial Time

1. Please log into your gmail accounts:

2. Open this lecture on GitHub:

   https://github.com/wbandabarragan/ISYA2025/blob/main/Python_for_Astrophysics/1_tutorial_py4astro.ipynb

3. Click on the "**Open in Colab**" icon and you are ready to code!

PYTHON FOR ASTROPHYSICS

Lecture 2

Wladimir E. Banda-Barragán
Universidad Yachay Tech

2025

Lecture 2 goals:

1. High-level vs low-level programming languages

2. Why is Python optimal for astrophysics?

3. Master Google Colab for Python.

What do you need for the practicals?

- A PC/laptop with any OS.

- Internet access.

- A Google/gmail account.

- A GitHub account (desirable, not strictly needed).

High-level vs. Low-level languages

Human
High-level

Python, Julia, IDL/GDL,
Mathematica, Matlab

C++, java

C89/90, Fortran77/90

Computer
Low-level

Binary (01001000)

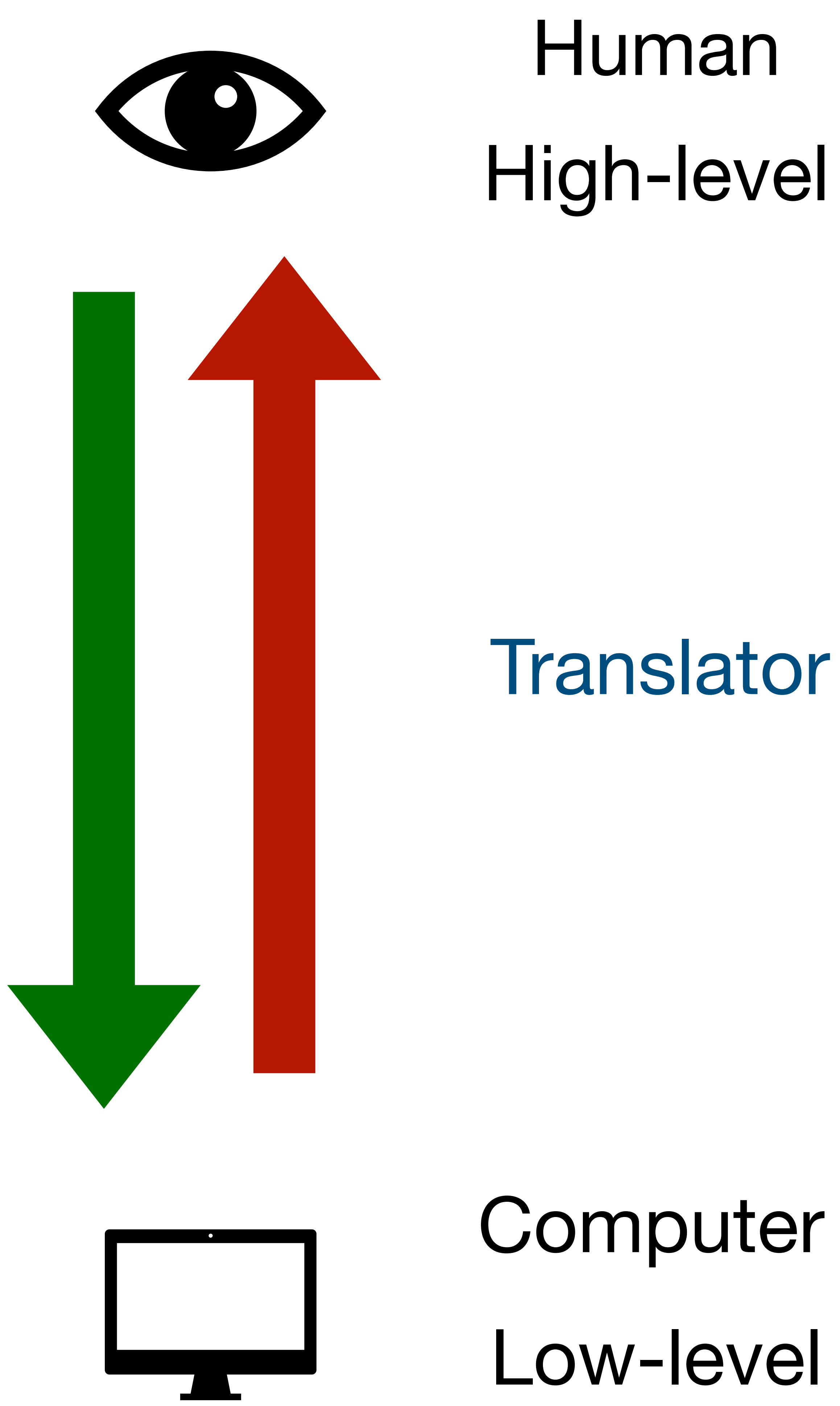
High-level vs. Low-level languages

Programs written in a high-level language **have to be processed** before they can run.

This **extra processing takes some time**, which is a small disadvantage of high-level languages.

Low-level programs can run on only one kind of computer and **have to be rewritten to run on another.**

Roughly speaking, computers **can only execute programs written in low-level languages.**

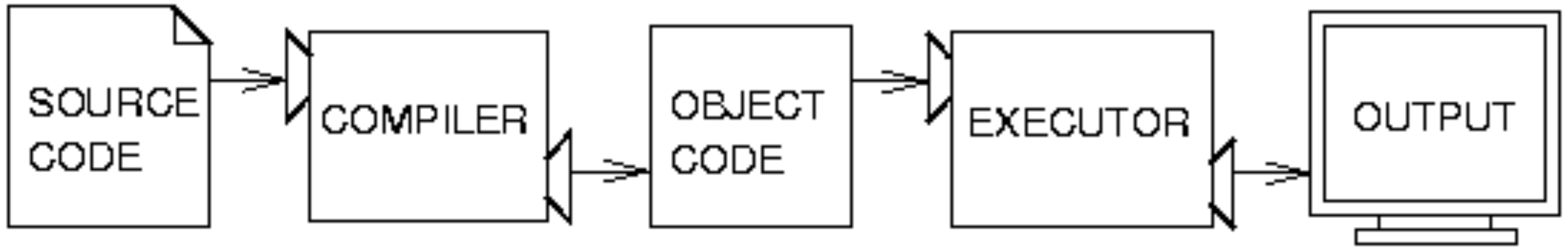


Interpreter vs. Compiler

Two kinds of programs process high-level languages into low-level languages: **interpreters** and **compilers**.

An **interpreter** reads a high-level program and executes it. It processes the program alternately reading lines and performing computations.

A **compiler** reads the program and translates it completely before the program starts running. The program is the **source code**, and the translated program is called the **object code** or the **executable**. Once compiled, you can execute it repeatedly without further translation.



Why python?

Python is considered an interpreted language because Python programs are executed by an interpreter.

The advantages of high-level languages are enormous:

- It is much easier to program in a high-level language.
- Programs written in a high-level language take less time to write.
- They are shorter and easier to read, and they are more likely to be correct.
- They are portable, meaning that they can run on different kinds of computers with few or no modifications.

Low-level languages are used only for a few specialised applications.

Package Managers

pip (python-specific package installer)

conda (language-agnostic package and environment manager)

- **Anaconda** (Full version)

- **Miniconda** (Reduced version)

source installation

- **Python Executable:**
  /usr/bin/python3

- **Standard Library & Third-Party Packages** (`site-packages`)
  /usr/lib/python3.x

Programming / IDE Tools

An IDE (Integrated Development Environment) is an application that provides comprehensive tools for software development.

An IDE consists of a source code editor, build automation tools (like compilers and interpreters), and a debugger.

Popular IDE choices are:

- Visual Studio Code (VS Code)
- Sublime Text
- Spyder
- Android Studio
- Xcode (Apple)
- **Google Colab** (cloud-based Jupyter notebook environment with IDE-like features)

Tutorial

Tutorial Time

1. Please log into your gmail accounts:

2. Open this lecture on GitHub:

   https://github.com/wbandabarragan/ISYA2025/blob/main/Python_for_Astrophysics/2_tutorial_py4astro.ipynb

3. Click on the "**Open in Colab**" icon and you are ready to code!

Open in Colab

INTERNATIONAL SCHOOL FOR YOUNG ASTRONOMERS

UNIVERSIDAD YACHAY TECH

PYTHON FOR ASTROPHYSICS

Lecture 3

Wladimir E. Banda-Barragán
Universidad Yachay Tech

2025

Lecture 3 goals:

1. Plotting tools for astronomy and astrophysics

2. Master Plotting on Google Colab

3. Hydrostatic Equilibrium of isothermal gas slabs (**stratified atmospheres**)

What do you need for the practicals?

- A PC/laptop with any OS.

- Internet access.

- A Google/gmail account.

- A GitHub account (desirable, not strictly needed).

Plotting in Python: Matplotlib

Matplotlib Resources

We will use mainly Matplotlib for plotting: https://matplotlib.org/

Some examples on how to use Matplotlib for plotting cane be found here:

https://matplotlib.org/tutorials/introductory/sample_plots.html
https://matplotlib.org/gallery/index.html

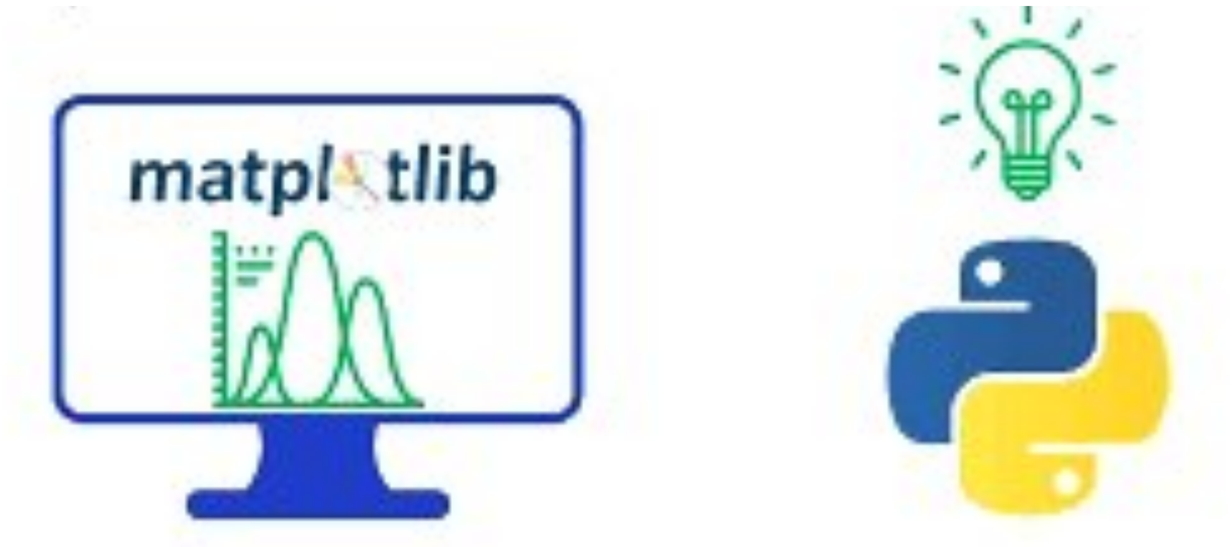
Other examples can be found on forums, e.g. StackOverflow:
https://stackoverflow.com/

Asking questions on Google/AI chatbots is also helpful as there are many links, examples, solutions to problems, etc.

Python is a community-developed language, so you are also welcome to report bugs:
https://bugs.python.org

Pyplot and Plotly

pyplot():



Pyplot is an interface to plot using Matlab- and Mathematica-like plotting sequences.

It facilitates plotting and it is widely used.

Website: https://matplotlib.org/stable/tutorials/pyplot.html

Pyplot and Plotly

Plotly: 3D interactive plotting

Plotly is an open-source graphing library used to create interactive visualisations directly in web browsers.

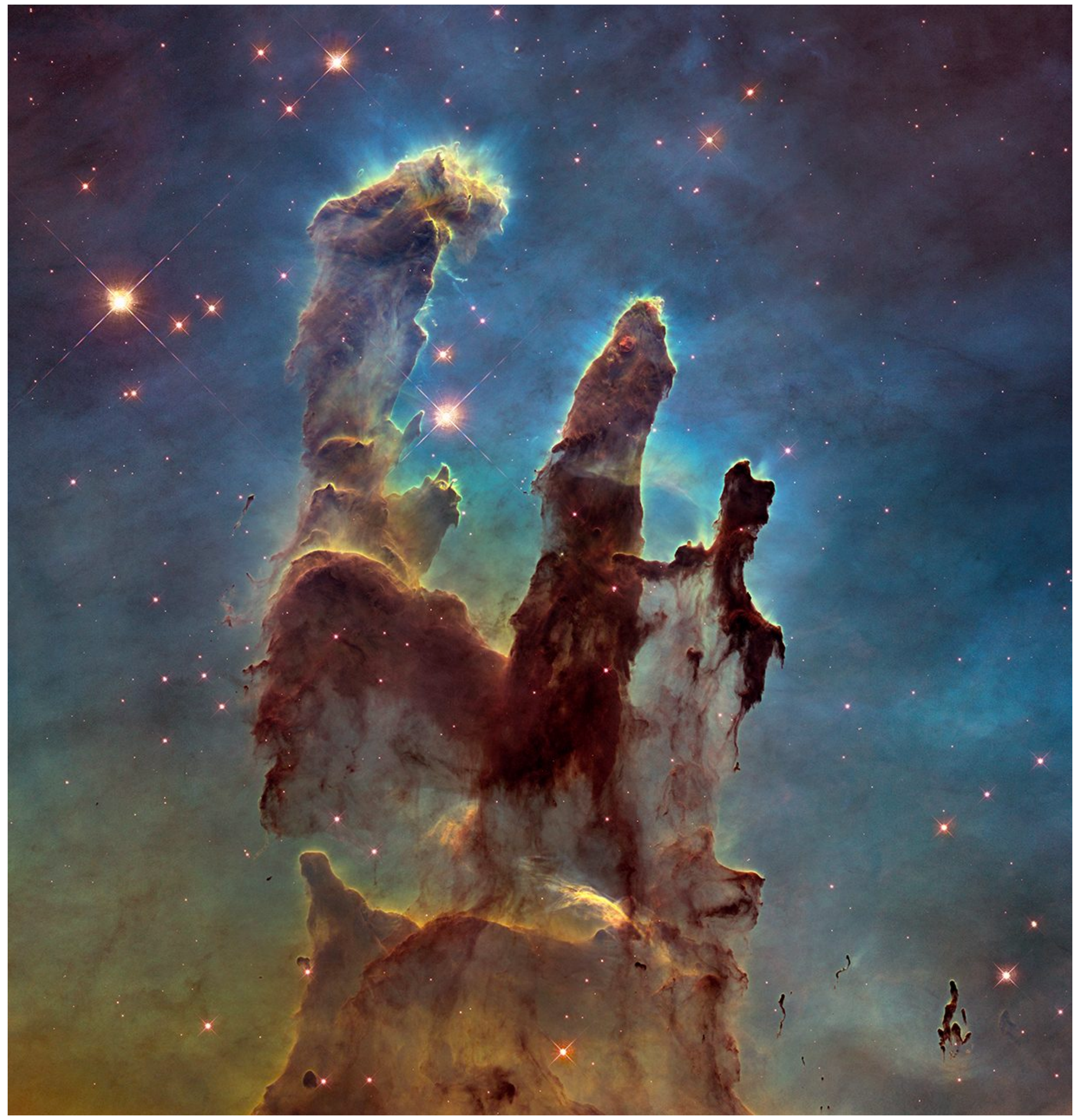
It can be integrated with jupyter notebooks.

Plotly is highly versatile, with APIs available in Python, R, JavaScript, HTML.

Website: https://plotly.com

Interstellar Medium Clouds

Galactic discs



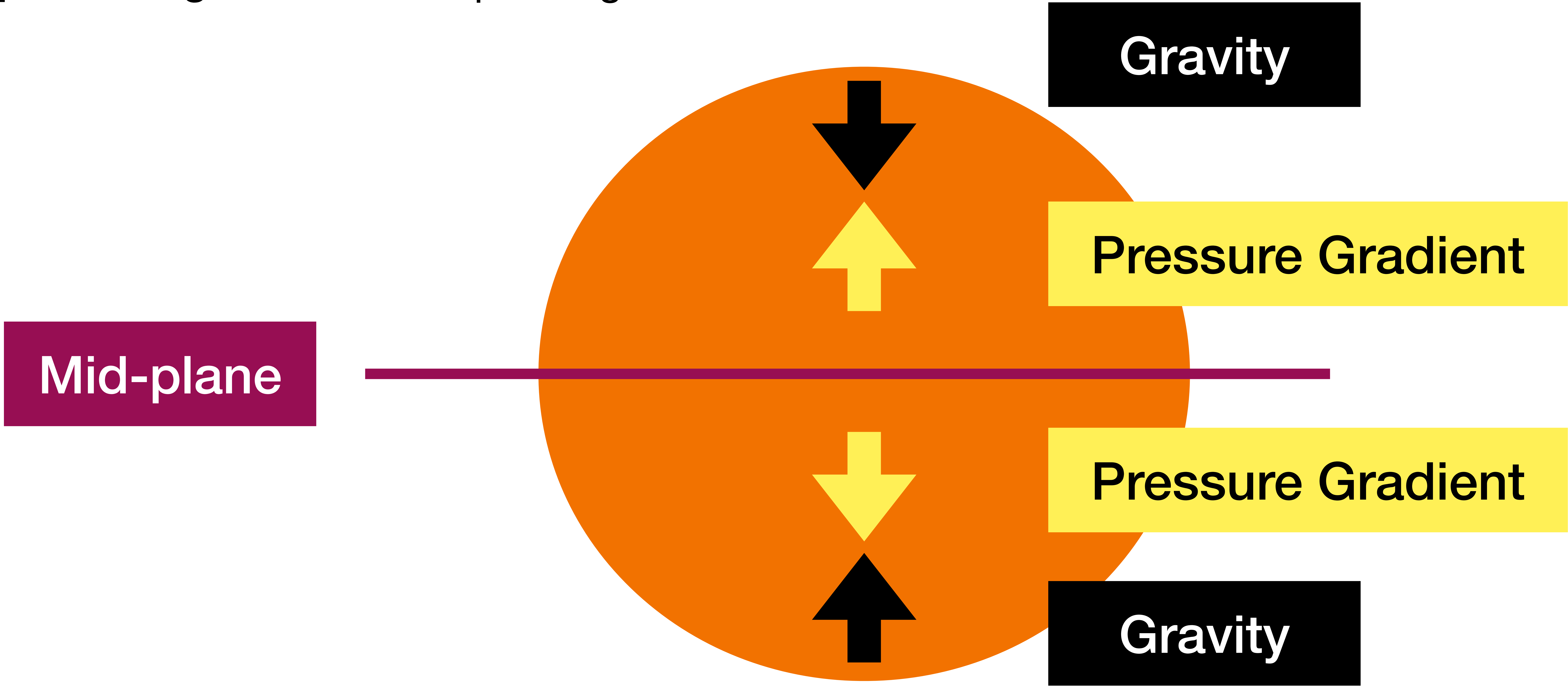
Stratified Astrophysical Atmospheres

Molecular Pillars in the Eagle Nebula (7000 light-years).
Credits: http://hubblesite.org/

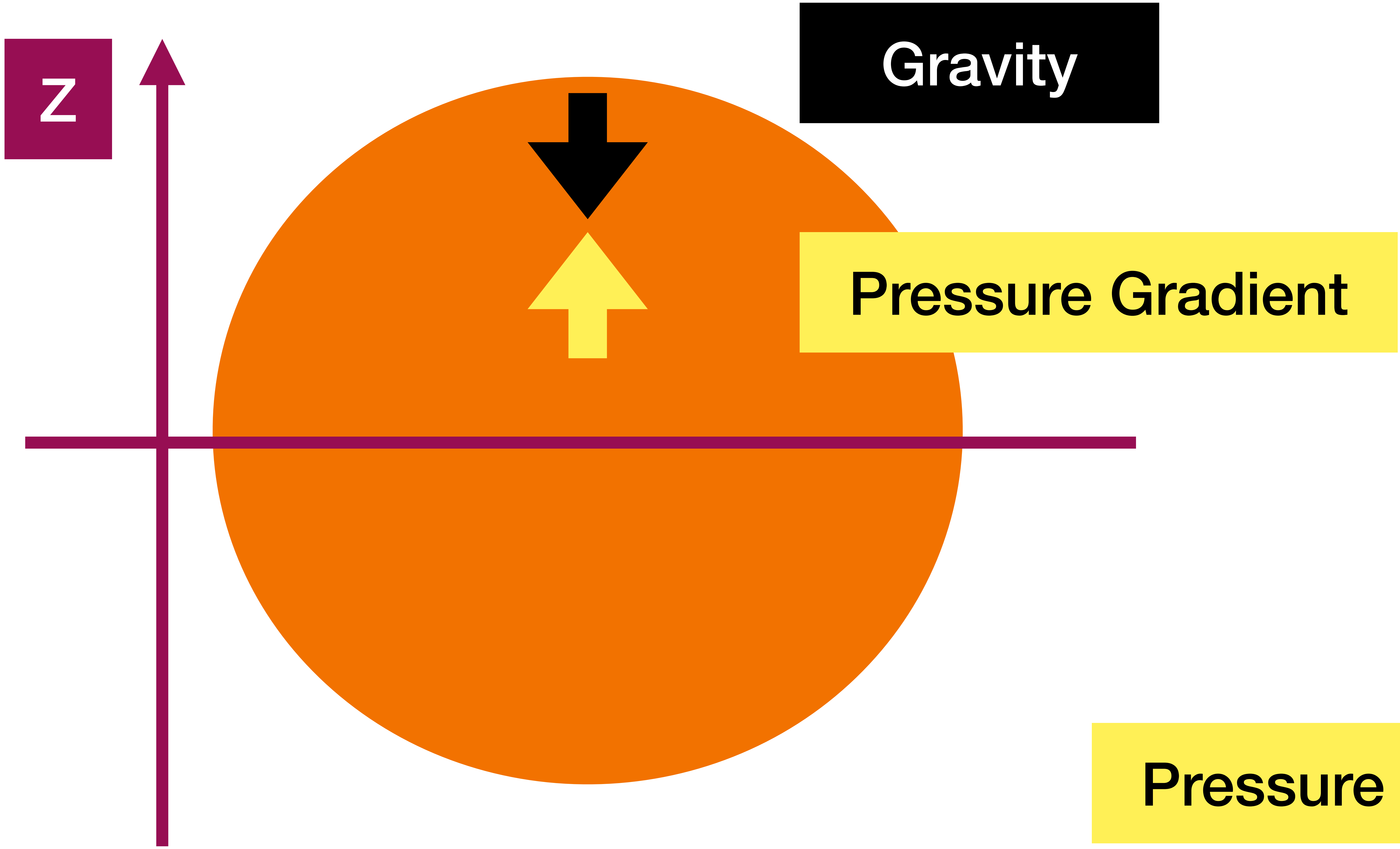
A Hubble image of the spiral disc galaxy NGC3972.
Credits: NASA/Hubble

Hydrostatic equilibrium

The **gravitational force** pulling gas towards the mid-plane is balanced with the **pressure gradient force** pushing outward.



Self-gravitating atmosphere (gas slab)



Vertical gradient of the gravitational potential

$$\frac{dP}{dz} = -\rho(z)\frac{d\Phi}{dz}$$

Pressure Gradient

Mass density

1D self-gravitating atmosphere (gas slab)

The gravitational potential, Φ, is sourced by the gas itself, so we need to solve the Poisson equation:

$$\frac{d^2\Phi}{dz^2} = 4\pi G\rho(z)$$

In addition, we need an equation of state that determines the thermodynamical properties of the gas. The simplest case is to assume the gas is isothermal.

$$P = \rho\, c_s^2$$

These equations combined with the hydrostatic equilibrium equation:

$$\frac{dP}{dz} = -\rho(z)\frac{d\Phi}{dz}$$

Give us in 1D, the following solution:

$$\rho(z) = \rho_0\, \mathrm{sech}^2\left(\frac{z}{H}\right)$$

with: $H = \dfrac{c_s}{\sqrt{2\pi G\rho_0}}$ (scale height)

Reference: https://ui.adsabs.harvard.edu/abs/1942ApJ....95..329S

2D/3D self-gravitating atmosphere (gas slab)

In 2D cylindrical symmetry (e.g., interstellar filaments), the equilibrium density profile is:

$$\rho(r) = \frac{\rho_0}{\left(1 + \frac{r^2}{8H^2}\right)^2}$$

with: $$H = \frac{c_s}{\sqrt{4\pi G\rho 0}}$$ (scale height)

Reference: https://ui.adsabs.harvard.edu/abs/1964ApJ...140.1056O

In 3D solutions require numerical integration. A commonly used analytic approximation to the solution of the isothermal Lane-Emden equation reads:

$$\rho(r) = \frac{\rho_0}{\left(1 + \frac{r^2}{r_c^2}\right)^{3/2}}$$

(~ 7% accurate wrt to the numerical isothermal sphere solution for $r \lesssim 4r_c$, where ρ_0 is the central density and r_c is a characteristic core radius.

Reference: https://ui.adsabs.harvard.edu/abs/2008gady.book.....B

Tutorial Time

1. Please log into your gmail accounts:

2. Open this lecture on GitHub:

   https://github.com/wbandabarragan/ISYA2025/blob/main/Python_for_Astrophysics/3_tutorial_py4astro.ipynb

3. Click on the “**Open in Colab**” icon and you are ready to code!

PYTHON FOR ASTROPHYSICS

Lecture 4

Wladimir E. Banda-Barragán
Universidad Yachay Tech

2025

Lecture 4 goals:

1. Look into some Astropy functions

2. Open FITS files of astronomical images

3. Open TNG simulation files with the YT package

What do you need for the practicals?

- A PC/laptop with any OS.

- Internet access.

- A Google/gmail account.

- A GitHub account (desirable, not strictly needed).

Astropy

Astropy is a toolkit for Astronomical Computing (helpful in observational and theoretical work).

Astropy is an **open-source project** that provides a comprehensive collection of tools designed to make astronomical research and data analysis more **efficient, accurate, and reproducible.**

https://www.astropy.org/

Astropy was created to standardise and centralise these efforts, offering a robust and well-tested framework for astronomical computing.

Astropy functionality

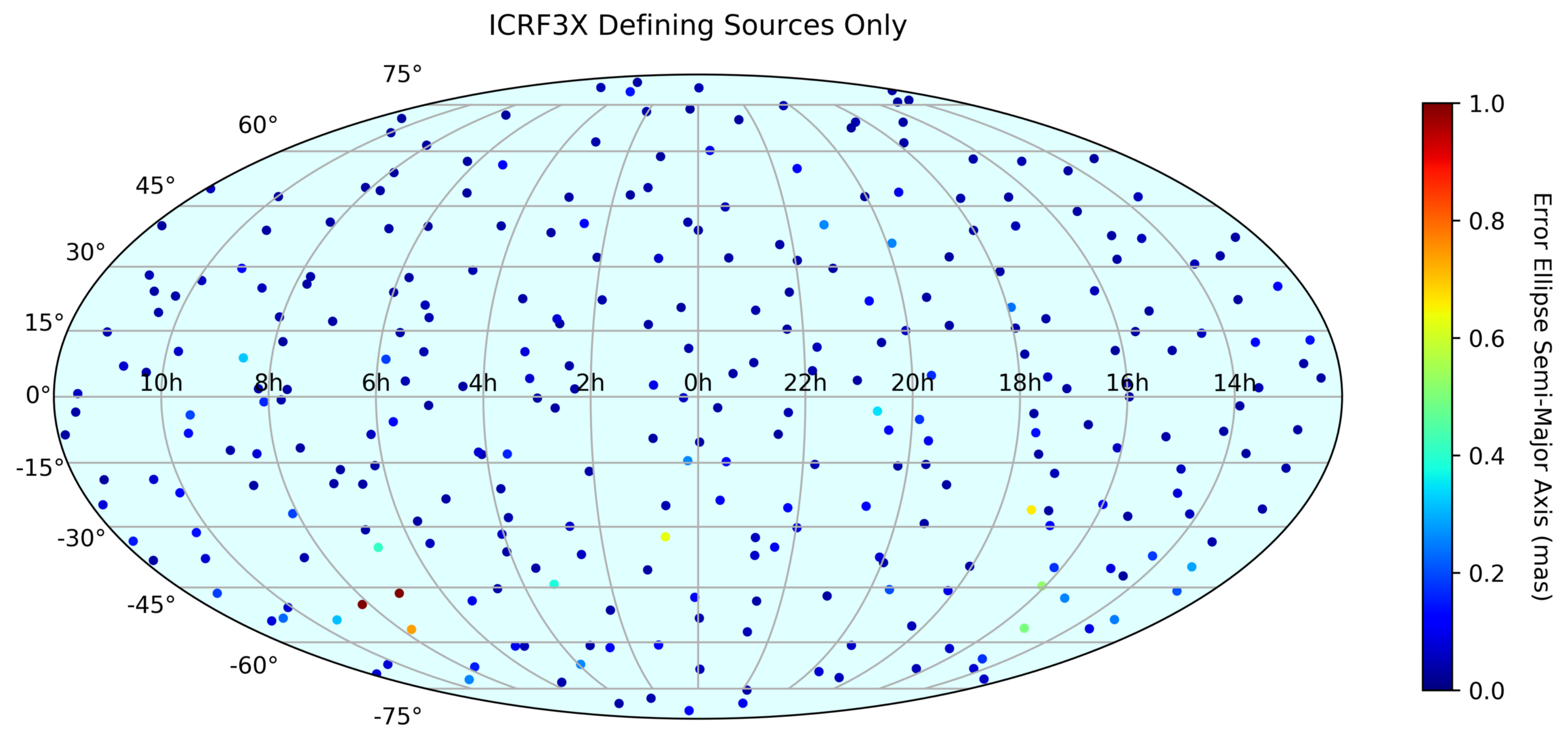
Some of its modules provide functionality for:

- Units and Physical Constants

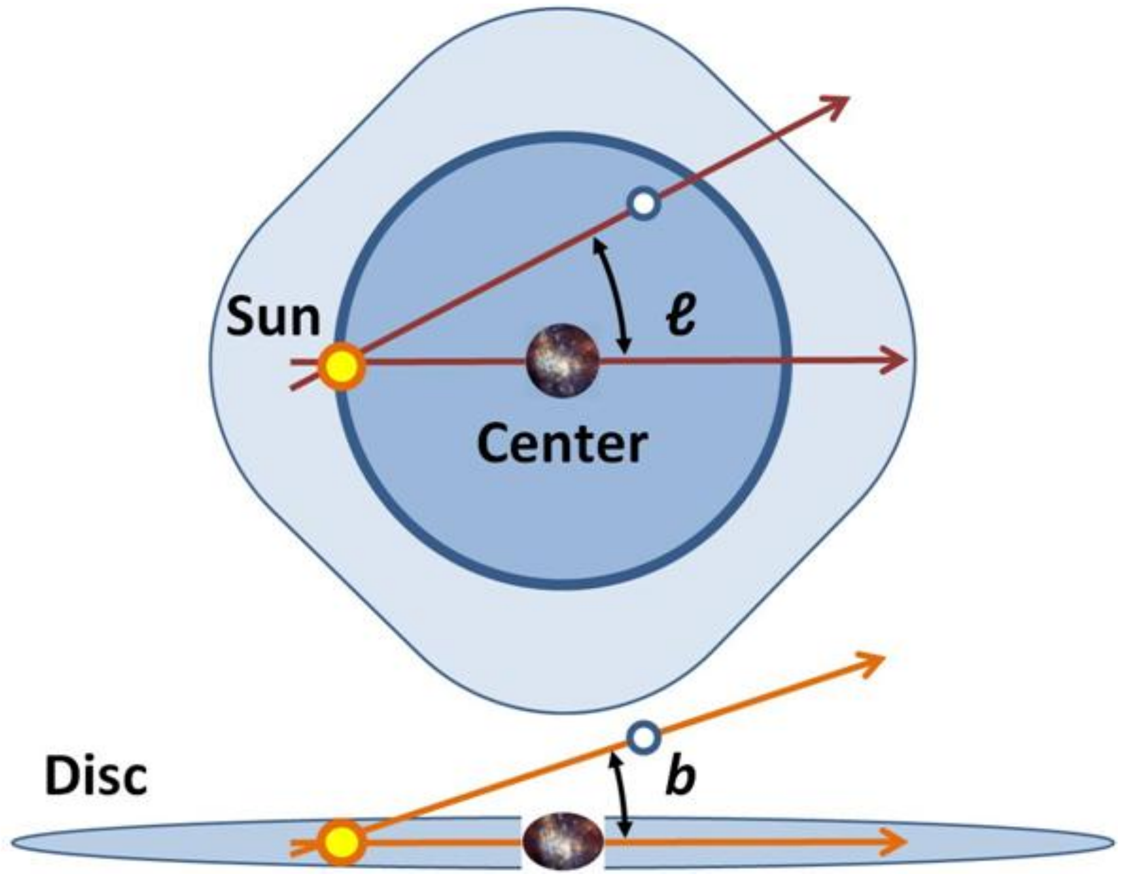
- Celestial Coordinates and Times

- Data Handling (e.g. **FITS format** readers)

- Visualisation tools



International Celestial Reference Frame.
https://en.wikipedia.org/wiki/International_Celestial_Reference_System_and_its_realizations



Galactic Reference Frame.
https://en.wikipedia.org/wiki/Galactic_coordinate_system

FITS data format

Flexible Image Transport System (FITS) format

This is the preferred data format used in **observational** astronomy.

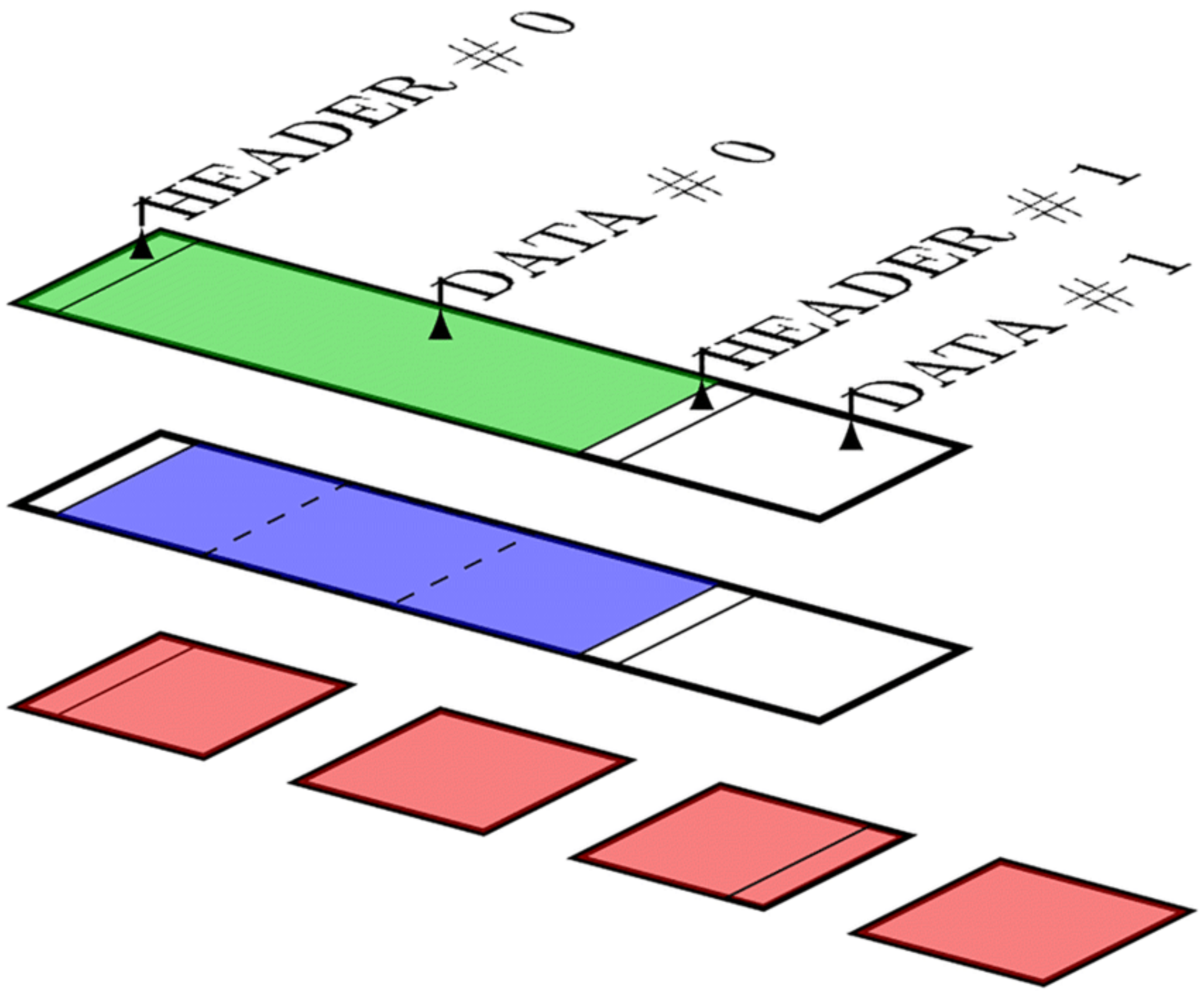
Theorists should learn to use and handle FITS files too.

It is mainly used as a standard format to share astronomical images.

But it can also contain tables or cubes, e.g. position-position-velocity diagrams

It has a header with metadata related to the image.

Header Data Units (HDU)



https://fits.gsfc.nasa.gov/fits_documentation.html

THINGS survey

The HI Nearby Galaxy Survey (THINGS)

This is the preferred data format used in **observational** astronomy.

The HI Nearby Galaxy Survey (THINGS) is one of the largest programs undertaken at the NRAO Very Large Array (VLA) to perform 21-cm HI observations of nearby galaxies

Technical details:

7” angular

5 km/s velocity resolution

https://www2.mpia-hd.mpg.de/THINGS/Overview.html

https://public.nrao.edu/gallery/this-is-the-vla-wye/

THINGS survey

The HI Nearby Galaxy Survey (THINGS)

The goal of THINGS is to investigate key characteristics related to:

Galaxy morphology

Star formation

Mass distribution

THINGS survey

The HI Nearby Galaxy Survey (THINGS)

A sample of **34 objects** at distances 3<D<15 Mpc have been targeted in THINGS.

The sample covers a wide range of star formation rates, total masses, absolute luminosities, evolutionary stages, and metallicities.

Moment 0 maps (**integrated intensity map**)

2D image showing the **total amount of emission** (e.g., from neutral hydrogen, HI) detected along the line of sight, integrated over the entire velocity range.

THINGS data products can be downloaded here. For detailed information on the data processing see the THINGS \`master' paper (Walter et al. 2008).

Please do not hesitate to contact us (walter@mpia.de) if you intend to use the data for scientific analysis. The following acknowledgement would be appreciated: "This work made use of THINGS, \`The HI Nearby Galaxy Survey' (Walter et al. 2008)"

Acknowledgement of the THINGS project is also appreciated if data are used in public presentations. Thank you.

For a detailed description of the following files, please see the THINGS \`master' paper (Walter et al. 2008). In brief: moment 0: integrated HI map; moment 1: mean intensity-weighted velocity; moment 2: velocity dispersion; cube: HI data cube. All moment maps were derived from the residual-scaled, blanked, primary beam corrected natural (na) and robust (ro) weighted data. The data cubes below are the \`standard' cubes.

	moment 0		moment 1		moment 2		data cube	
DDO 53	na	ro	na	ro	na	ro	na	ro
DDO 154	na	ro	na	ro	na	ro	na	ro
Holmberg I	na	ro	na	ro	na	ro	na	ro
Holmberg II	na	ro	na	ro	na	ro	na	ro
M81 dwarf A	na	ro	na	ro	na	ro	na	ro

https://www2.mpia-hd.mpg.de/THINGS/Data.html

Tutorial Time

1. Please log into your gmail accounts:

2. Open this lecture on GitHub:

   https://github.com/wbandabarragan/ISYA2025/blob/main/Python_for_Astrophysics/4_tutorial_py4astro.ipynb

3. Click on the “**Open in Colab**” icon and you are ready to code!

INTERNATIONAL SCHOOL FOR YOUNG ASTRONOMERS

PYTHON FOR ASTROPHYSICS

Lecture 5

Wladimir E. Banda-Barragán
Universidad Yachay Tech

2025

Lecture 5 goals:

1. Utilise YT to open simulation files.

2. Utilise H5PY to open and visualise simulation files.

3. Visualise simulation files with matplotlib and plotly.

What do you need for the practicals?

- A PC/laptop with any OS.

- Internet access.

- A Google/gmail account.

- A GitHub account (desirable, not strictly needed).

The YT project

The YT package in Python is a versatile and powerful open-source library designed for **analysing and visualising volumetric data.**

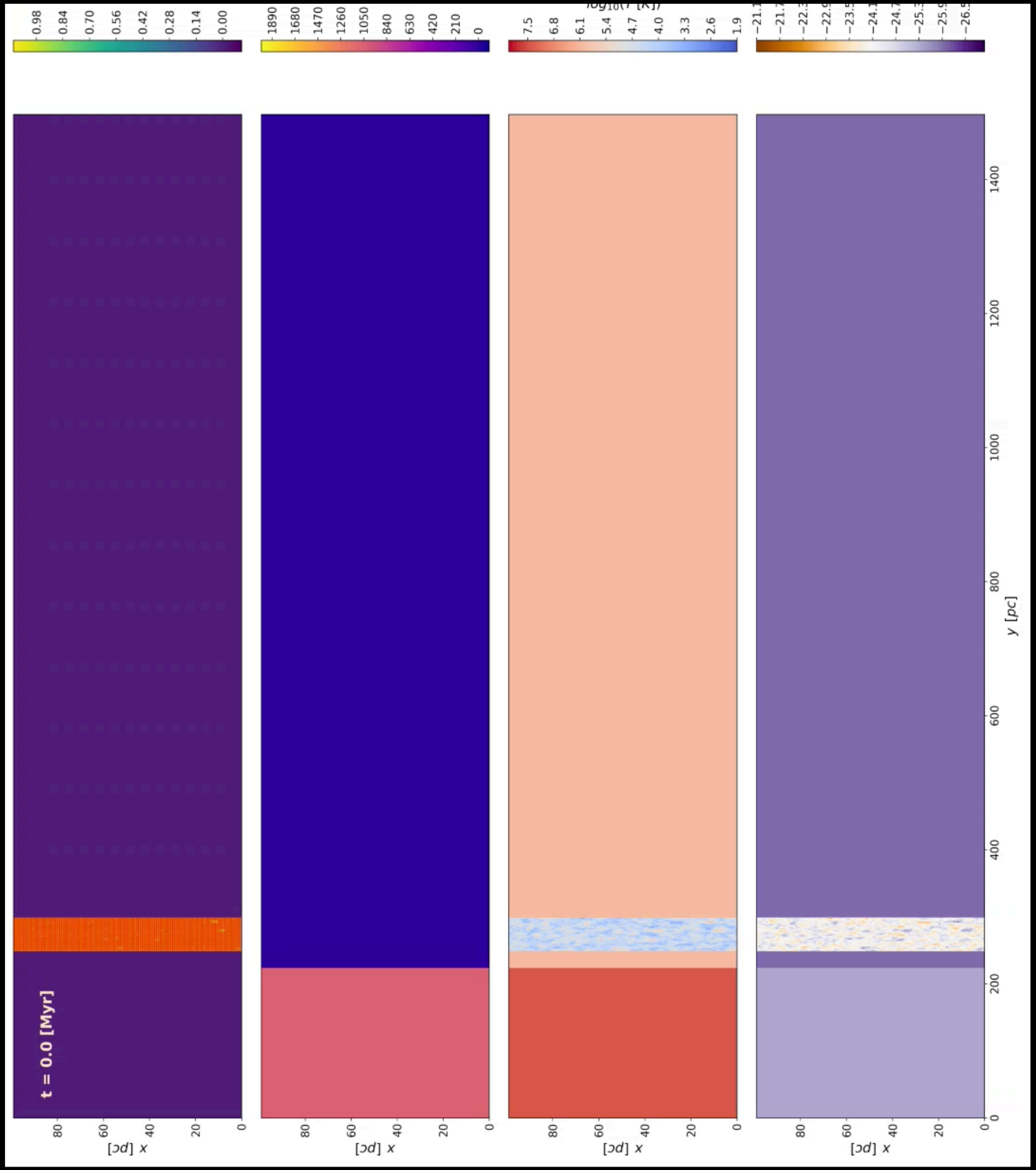
It supports a wide range of data formats and simulation codes, enabling users to perform complex **data analysis** tasks such as creating **projections, slices**, and **profiles**.

YT integrates well with other scientific libraries like **NumPy** and **Matplotlib**.

https://yt-project.org/

Simulation Types

Grid-based discretisation (Mesh)
Shock-cloud models



Antipov, Banda-Barragán et. al. 2025
https://ui.adsabs.harvard.edu/abs/2025MNRAS.540.3798A

Friends-of-friends algorithm

Simulation Types

Particle-based discretisation (SPH)

Relativistic disc break up around a Black Hole

Bardeen-Peterson effect: Nealon et. al 2015
(https://www.youtube.com/watch?v=AaNTY42zgtA)

The HDF5 data format

This is also a self-descriptive format, which is widely used in many physics sub-fields because of its versatility.

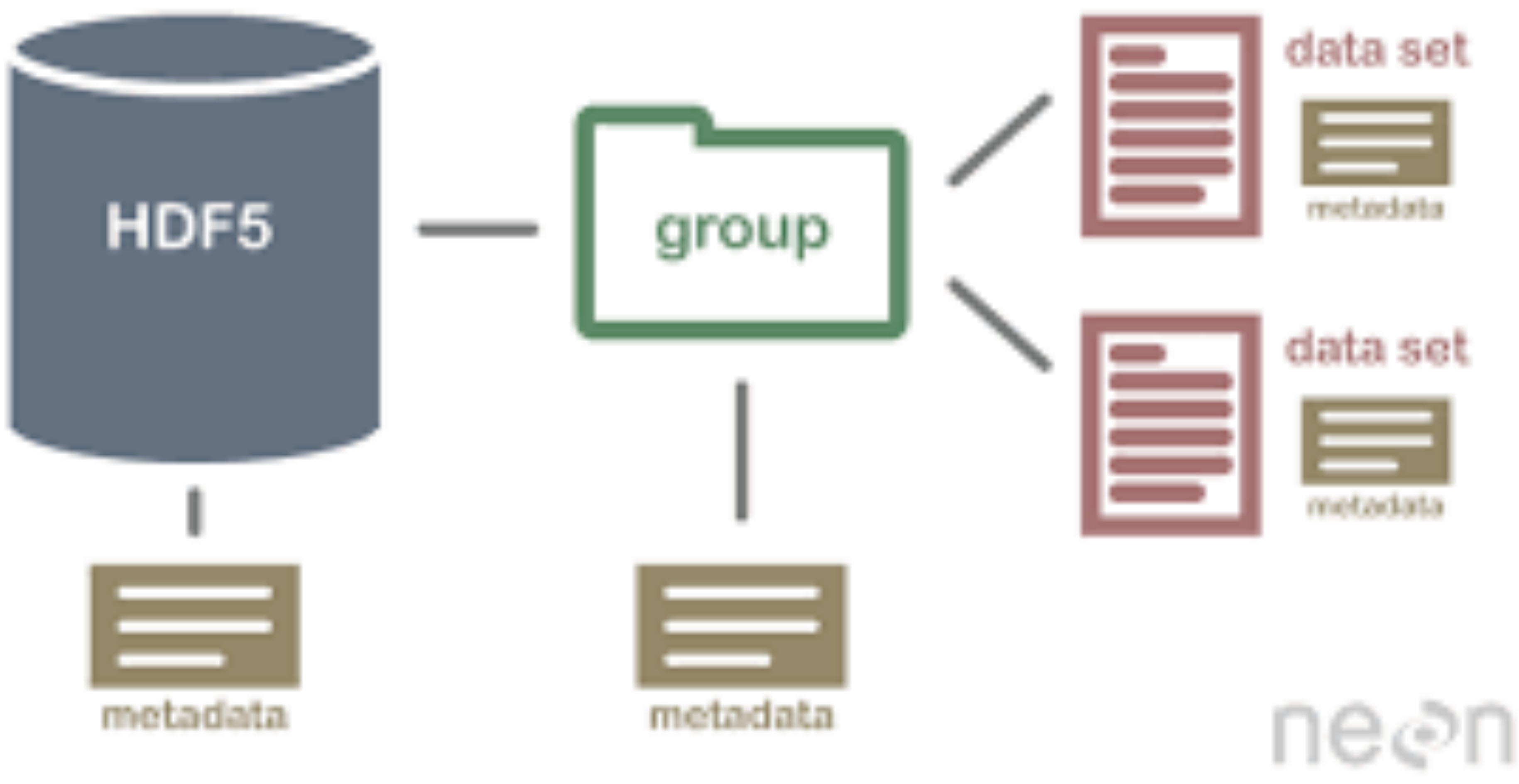
The **data** stored here can be **very large**!

Data don't have to be of the same type, you can save **numbers, units, strings, images**, etc, all in the same file.

It uses the structure of a file directory, organising information in **GROUPS**.

GROUPS are then comprised of data fields with their own metadata.

https://www.hdfgroup.org/



https://www.neonscience.org/resources/learning-hub/tutorials/about-hdf5

IllustrisTNG-3 simulation

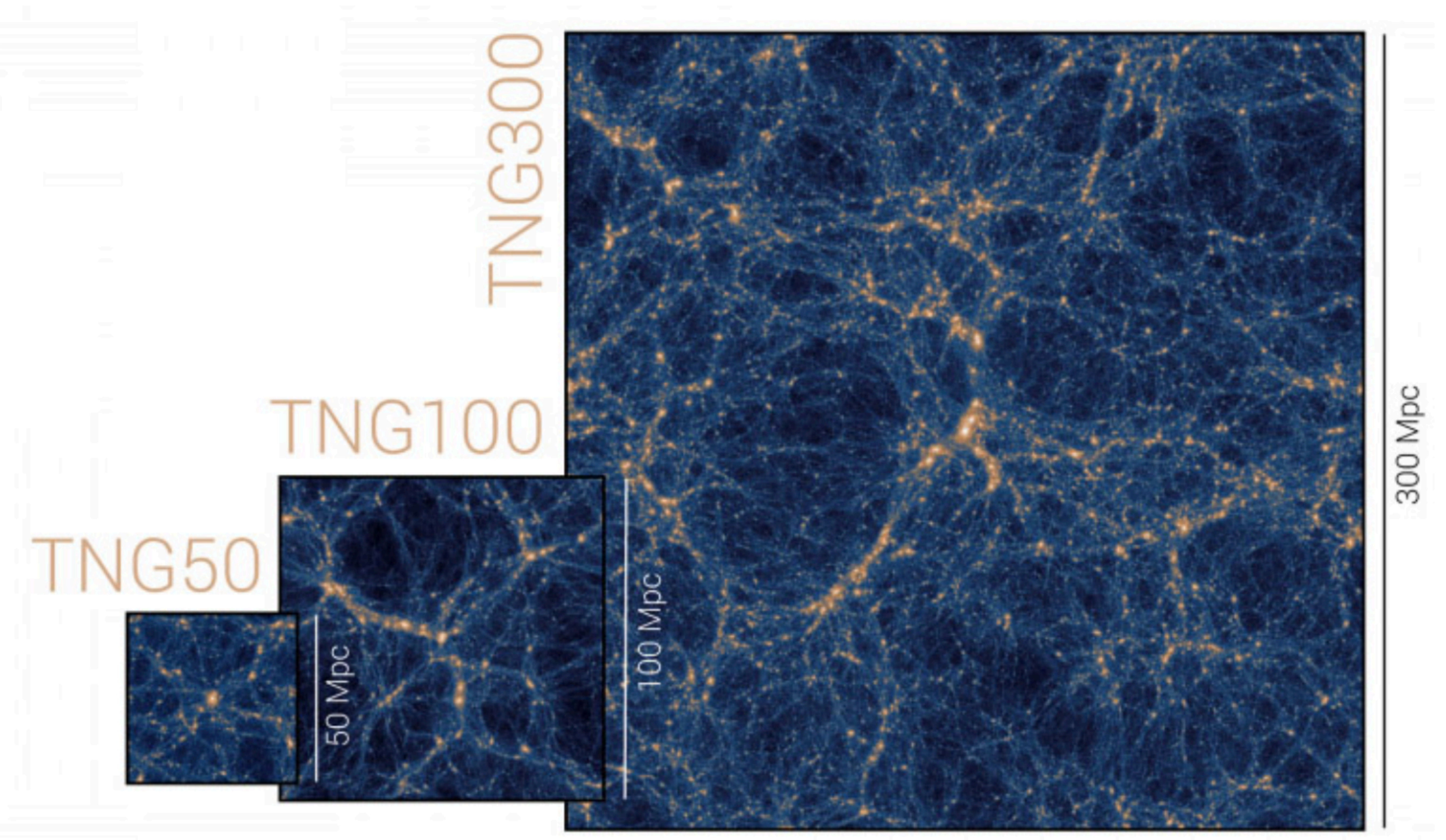
IllustrisTNG-3 is one of the large-scale **cosmological simulations**, used to model galaxy formation & evolution in **ΛCDM.**

TNG-3 simulates a cube of the universe **~302.6 Mpc/h on a side**, evolving

- dark matter
- gas
- stars
- black holes
- from the early universe (z ~ 127) to the present day (z = 0),

Illustrious 3 is publicly available!

https://www.tng-project.org



Tutorial Time

1. Please log into your gmail accounts:

2. Open this lecture on GitHub:

   https://github.com/wbandabarragan/ISYA2025/blob/main/Python_for_Astrophysics/4_tutorial_py4astro.ipynb

3. Click on the “**Open in Colab**” icon and you are ready to code!

ISYA
INTERNATIONAL SCHOOL FOR YOUNG ASTRONOMERS

UNIVERSIDAD YACHAY TECH

PYTHON FOR ASTROPHYSICS

Lecture 6

Wladimir E. Banda-Barragán
Universidad Yachay Tech

2025

Lecture 6 goals:

1. Learn the capabilities of pyvista.

2. Create animations of a full simulation dataset (an MHD vortex).

3. Visualise simulation files with matplotlib and plotly.

What do you need for the practicals?

- A PC/laptop with any OS.
- Internet access.
- A Google/gmail account.
- A GitHub account (desirable, not strictly needed).

The pyvista library

PyVista provides an intuitive Python interface for plotting and mesh analysis, built on top of VTK (Visualization Toolkit).

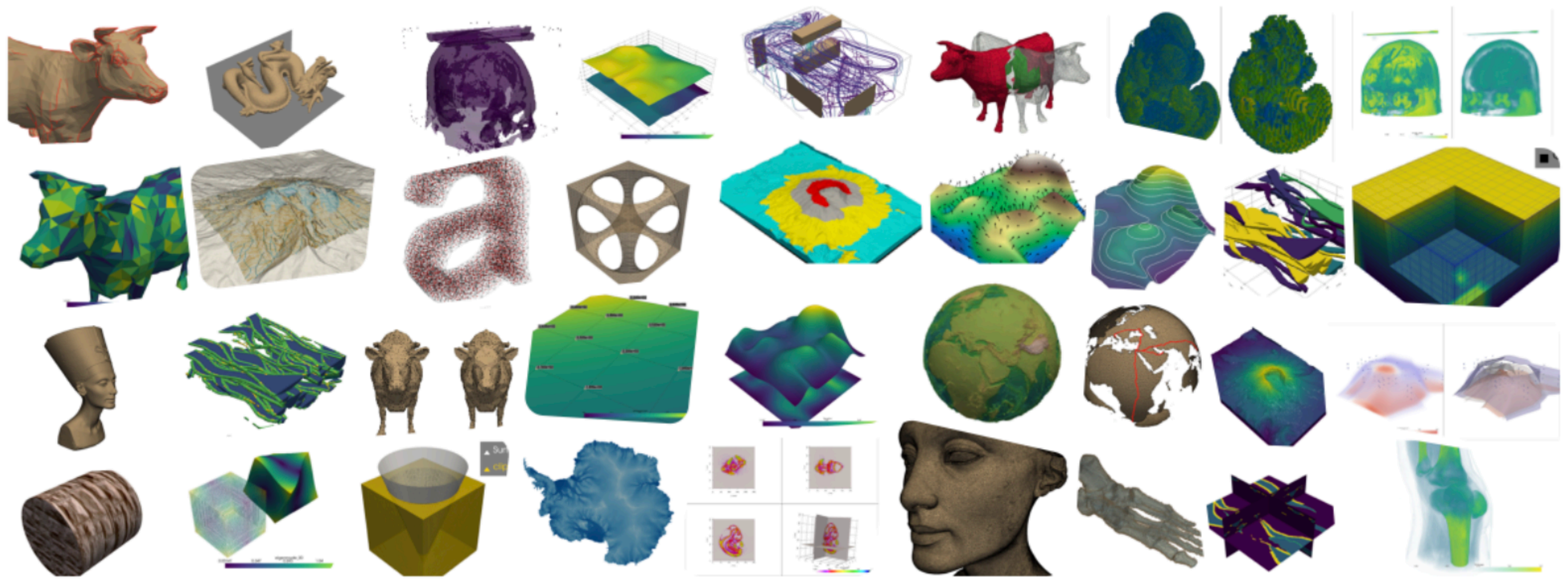
It supports several meshes including structured/unstructured grids, point clouds, and surface meshes with easy manipulation and filtering.

It integrates well with other scientific libraries like **NumPy** and **Matplotlib**.

We will use it for I/O of **VTK files.**

VTK = Visualisation Toolkit

3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK)



https://docs.pyvista.org/index.html

https://vtk.org/

VTK data format

VTK: Visualisation ToolKit (VTK) format

This format is an open-source data format, developed by Kitware, and widely used in computational fluid dynamics and computer graphics applications.

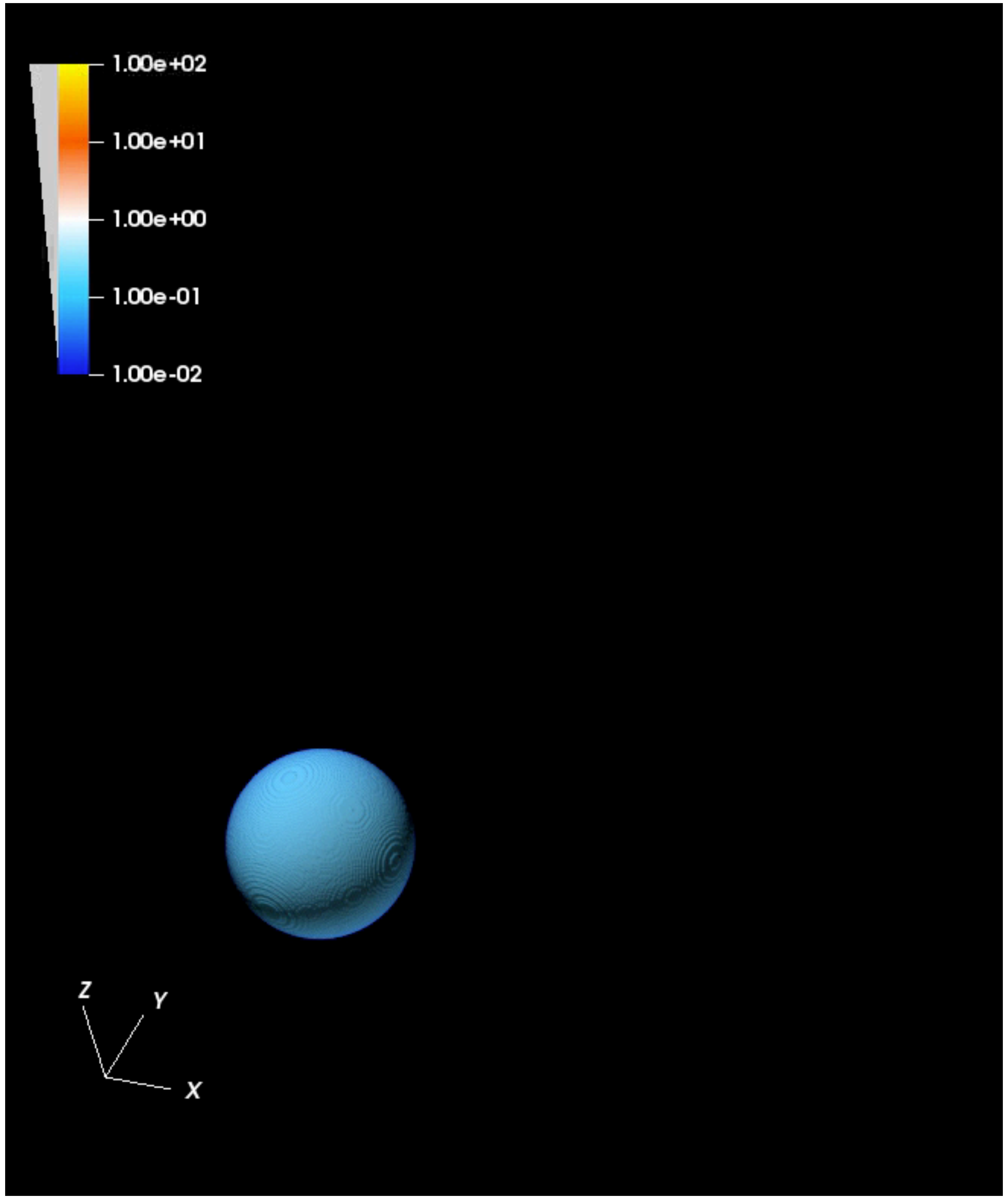
VTK file structure:

- File version and ID.
- Header
- Data type, which can be Binary or ASCII.

STRUCTURED_GRID

UNSTRUCTURED_GRID

RECTILINEAR_GRID



https://ui.adsabs.harvard.edu/abs/2016MNRAS.455.1309B

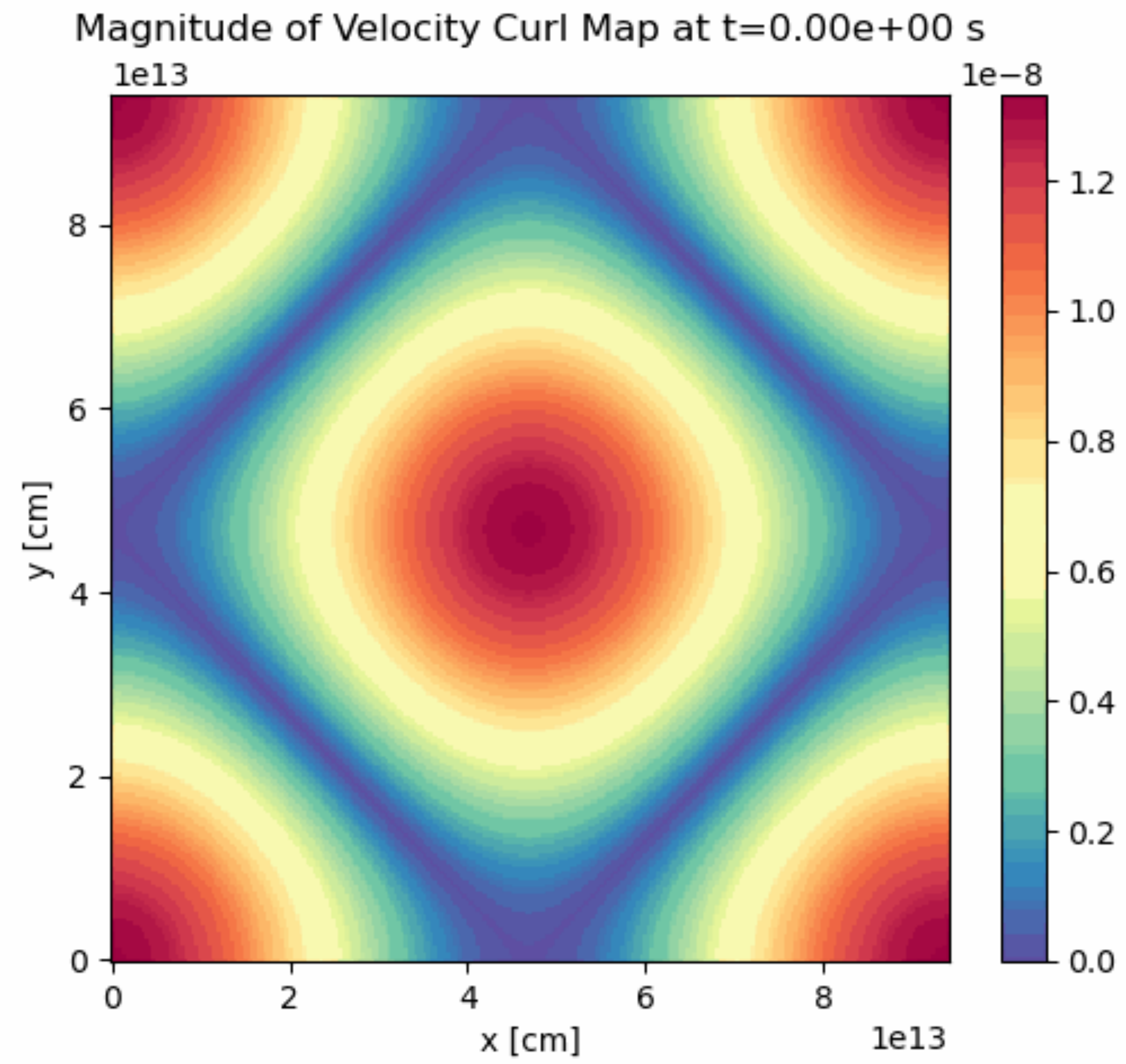
Orszag-Tang Vortex

A **doubly periodic fluid configuration** leading to 2D supersonic **magnetohydrodynamical** (MHD) turbulence

Benchmark for the comparison of MHD numerical solvers.

The computational domain is periodic box with dimensions: $[0,2\pi]^D$ where D=2 is the number of spatial dimensions.

In code units, the initial conditions are given by:



$$\vec{v} = \left(-\sin y,\ \sin x\right)\ , \qquad \overrightarrow{B} = \left(-\sin y,\ \sin 2x\right)\ , \qquad \rho = 25/9\,, \qquad p = 5/3$$

Tutorial Time

1. Please log into your gmail accounts:

2. Open this lecture on GitHub:

   https://github.com/wbandabarragan/ISYA2025/blob/main/Python_for_Astrophysics/4_tutorial_py4astro.ipynb

3. Click on the "**Open in Colab**" icon and you are ready to code!