# Unit 4. Partial differential equations (PDEs) in physics

**Lecture 401: simulation in physics**

**Reference book:**

**“Introduction to Computational Astrophysical Hydrodynamics” by Zingale.**
**https://open-astrophysics-bookshelf.github.io/numerical_exercises/**

**W. Banda-Barragán, 2025**

# What is a physics simulation?

Simulation enables us to build a model of a system and allows us to do virtual experiments to understand how this system reacts to a range of conditions and assumptions.

Simulations are not designed to “match reality”, but to understand the underlying physical process that lead to specific observed realities.

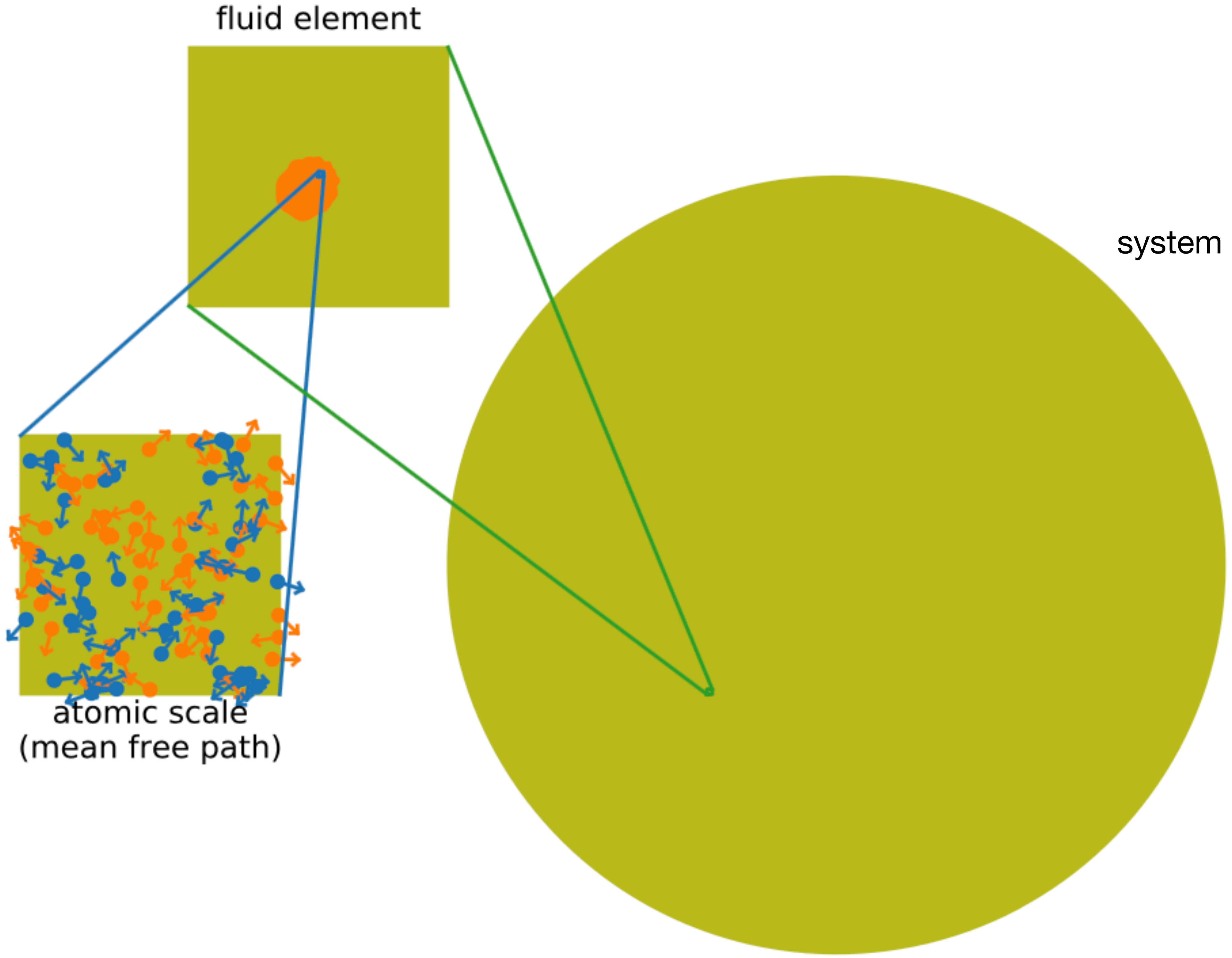
It is tempting to think that one can download a simulation code, set a few parameters, maybe edit some initial conditions, run, and then have a virtual realisation of some physical system that you are interested in. Just like that. In practice, it is not this simple.

# What is a physics simulation?

All simulation codes **make approximations**—these start even before one turns to the computer, simply by making a choice of what equations are to be solved.

A simulation has:

- A computational domain
- Physical system parameters, variables
- Initial conditions
- Boundary conditions

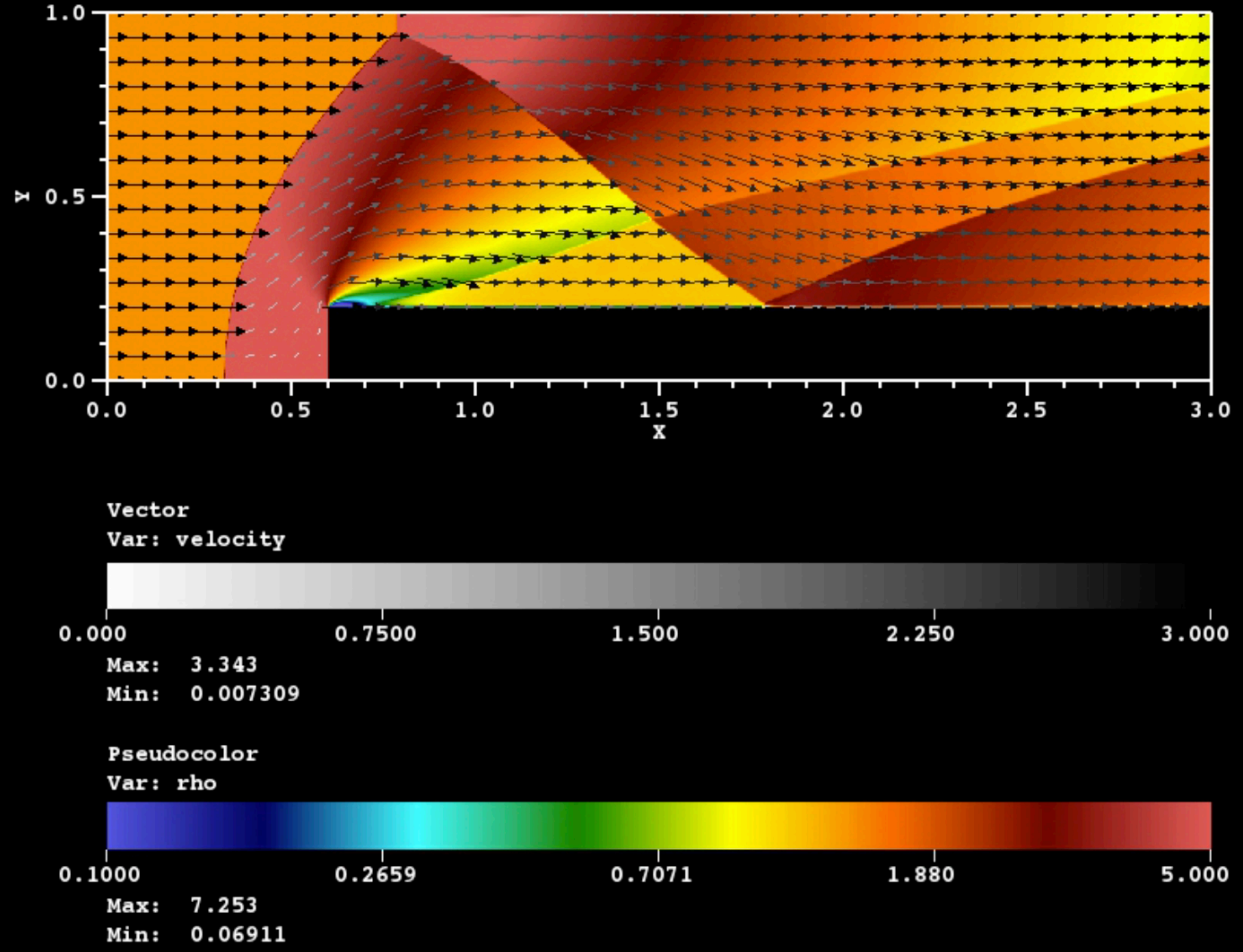


# Hydrodynamic simulations

Hydrodynamics

Fluids

Motion




Wind tunnel

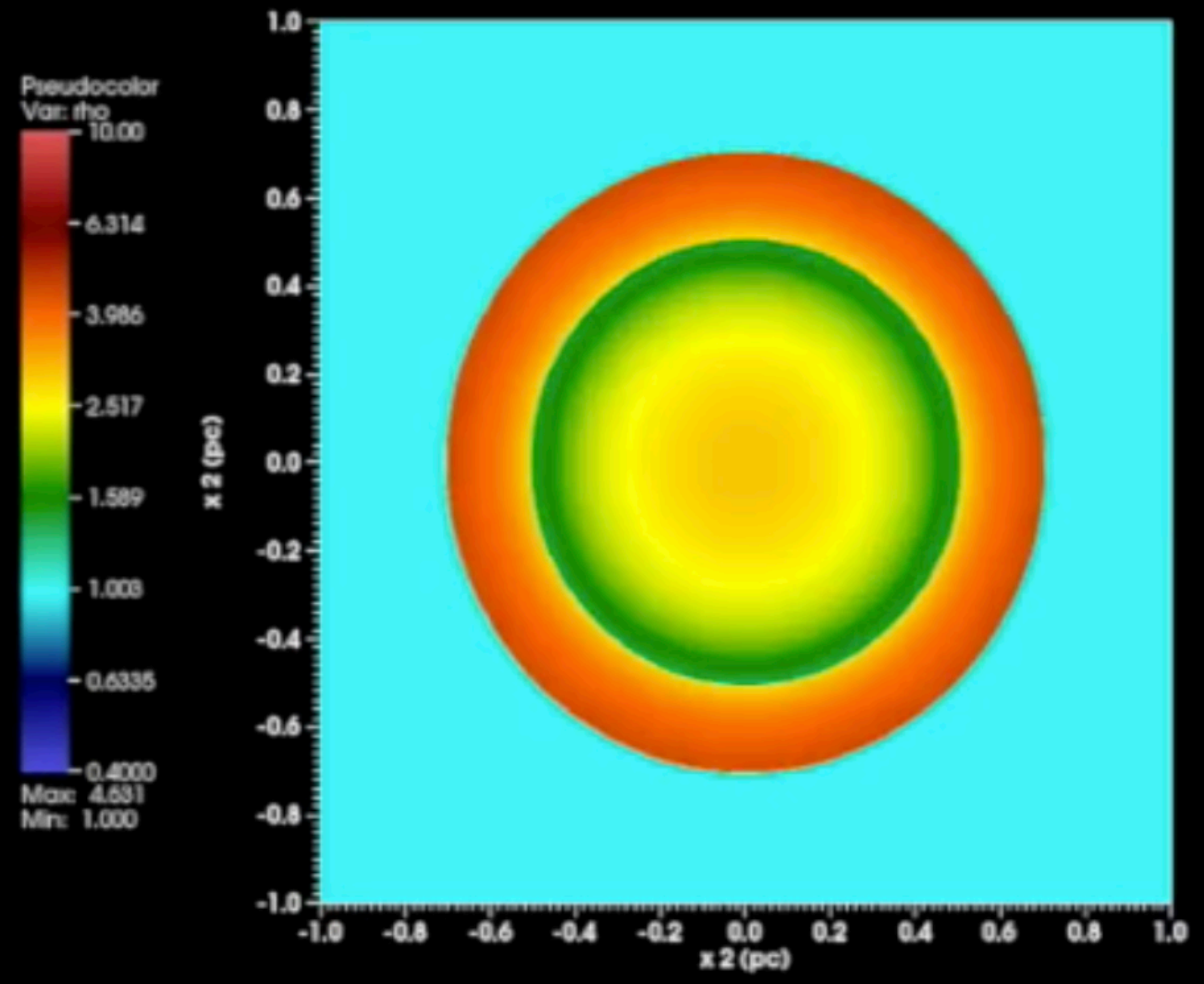
# Magneto-hydrodynamic simulations

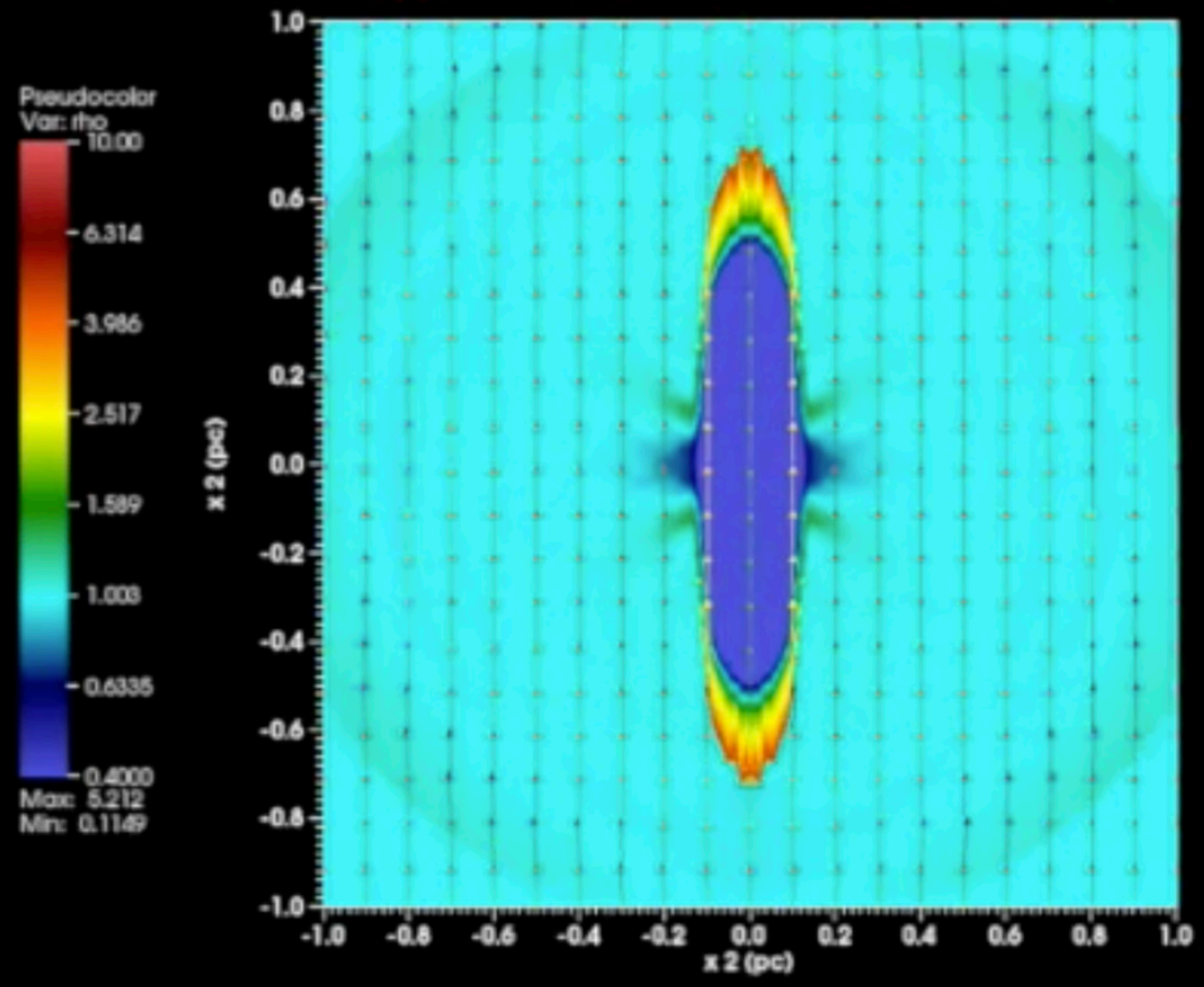
Magnetohydrodynamics

Campos magnéticos

Fluidos

Movimiento

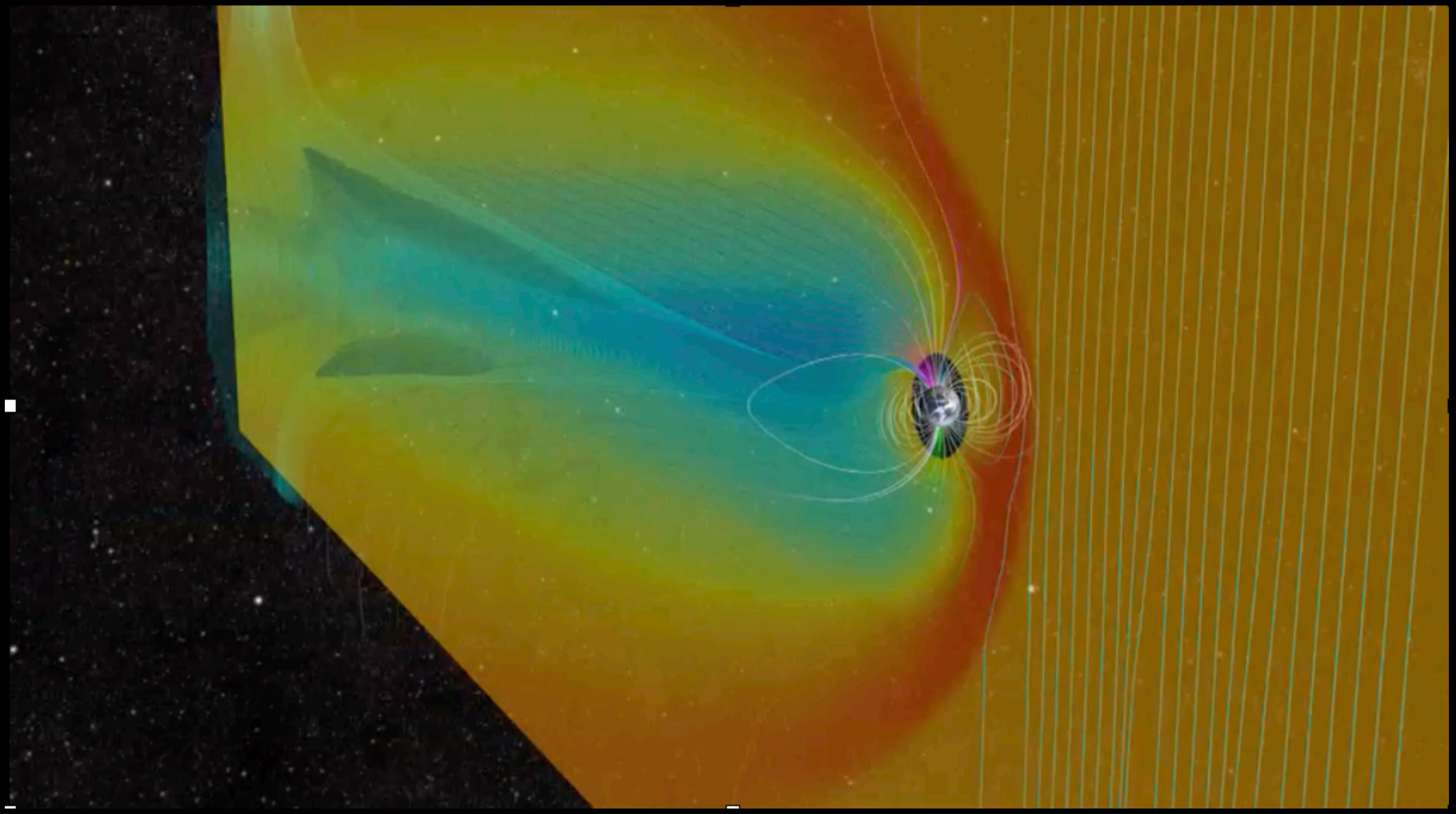



OAQ (2016)

# Magneto-hidrodynamics

**It is the study of the dynamics (i.e. movement) of plasmas in the presence of magnetic fields.**



Bow shock

Magnetic reconnection

Animación tomada de: http://svs.gsfc.nasa.gov/11689

# What is a physics simulation?

## Example: Fluid dynamics simulation

The main approximation that we will follow here, is the *fluid approximation*.

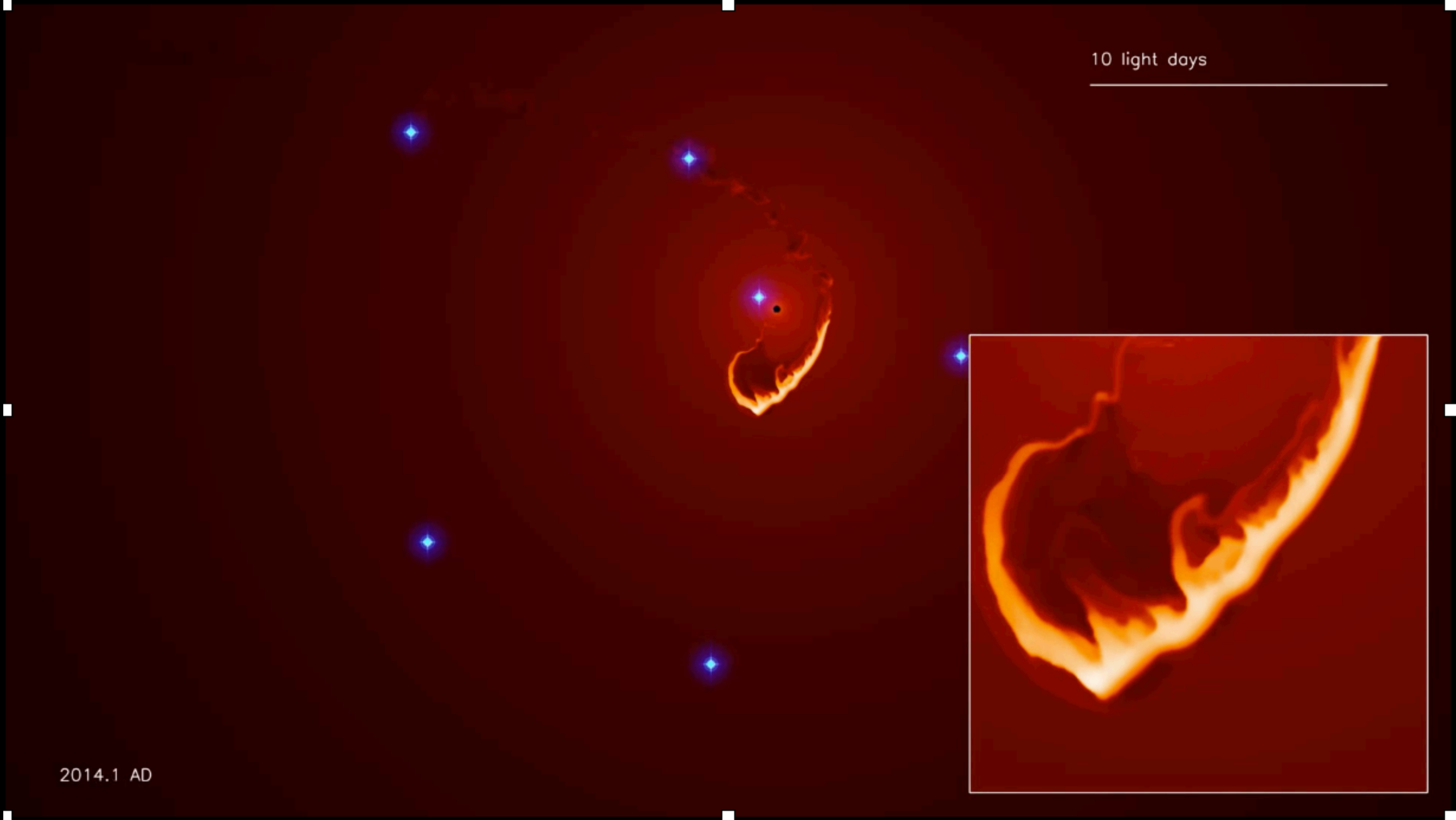
We don't want to focus on the motions of the individual atoms, nuclei, electrons, and photons in our system, so we work on a scale that is much larger than the mean free path of the system.

This allows us to describe the bulk properties of a fluid element, which in turn is small compared to the system of interest.

Within the fluid approximation, additional approximations are made, both in terms of the physics included and how we represent a continuous fluid in the finite-memory of a computer (the *discretisation* process).

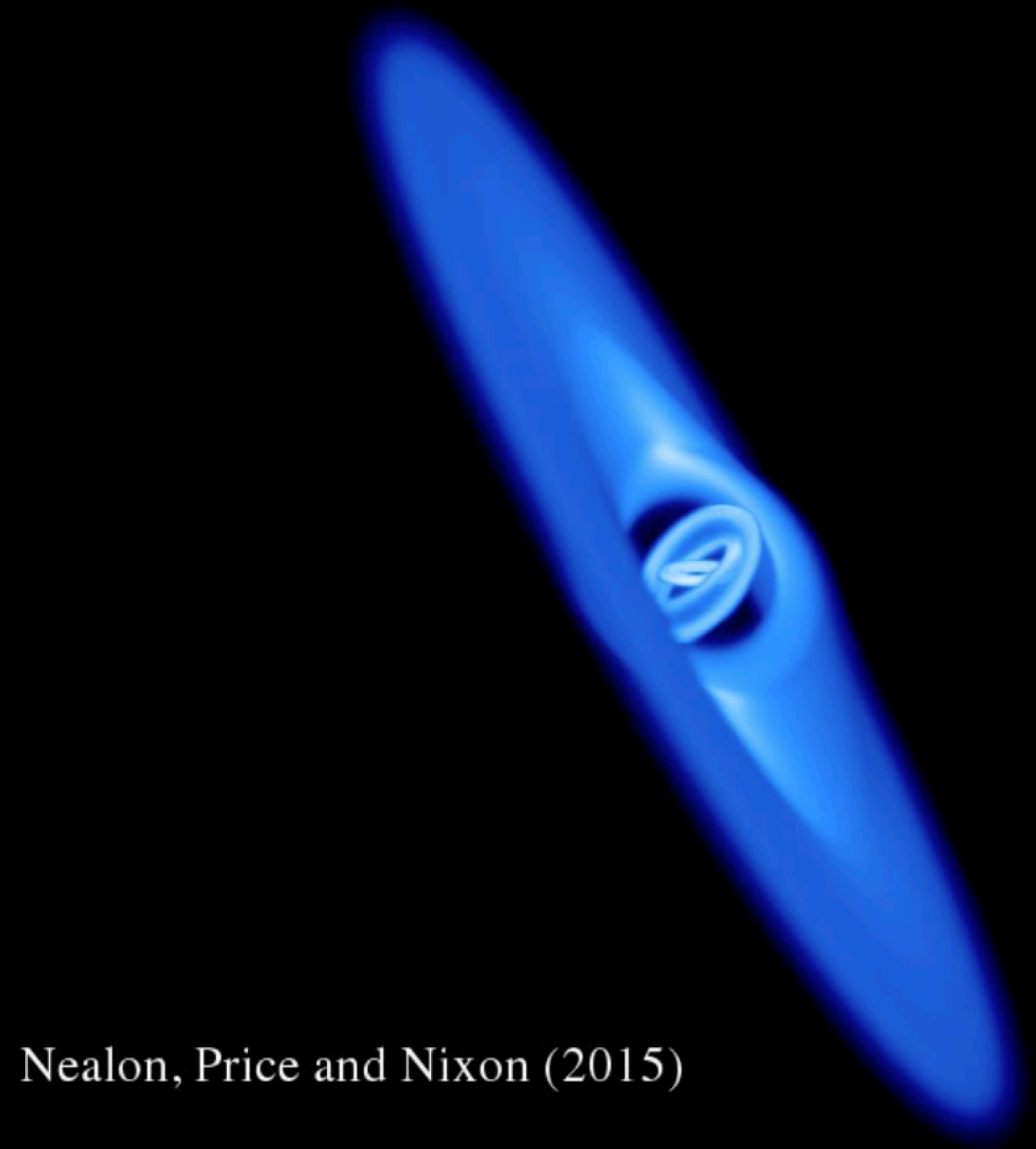
# Grid-based discretisation (Mesh)
# Cloud orbiting a Black Hole.

A gas cloud on its way into the super-



# Particle-based discretisation (SPH)

## Relativistic disc break up around a Black Hole




Bardeen-Peterson effect: Nealon et. al 2015

# What is a physics simulation?

## Example: Fluid dynamics simulation

The fluid approximation is valid when the mean free path ($\lambda$) of the particles in the fluid is much smaller than the characteristic length scale ($L$) of the system or the phenomena being studied.

$$\lambda = \frac{1}{\sqrt{2}n\sigma} \ll L$$

$n$ is the number density of the particles.

$\sigma$ is the collision cross-section of the particles.

Collisions between particles are frequent enough to maintain local thermodynamic equilibrium, allowing macroscopic fluid properties like density, pressure, and temperature to be well-defined and vary smoothly.

# What is a physics simulation?

## Example: Fluid dynamics simulation

Typically, we have a system of PDEs, and we need to convert the continuous functional form of our system into a discrete form that can be represented in the finite memory of a computer. This introduces yet more approximation.

Blindly trusting the numbers that come out of the code is a recipe for disaster. You don't stop being a physicist the moment you execute the code—your job as a computational scientist is to make sure that the code is producing reasonable results, by testing it against known problems and your physical intuition.

## What is a physics simulation?

Because the systems we solve are so nonlinear, small changes in the code or the programming environment (compilers, optimisation, etc.) can produce large differences in the numbers coming out of the code. That's not a reason to panic.

As such it is best not to obsess about precise numbers, but rather the trends our simulations reveal. To really understand the limits of your simulations, you should do **parameter and convergence studies.**

Every algorithm begins with approximations and has limitations. Comparisons between different codes are important and common, and build confidence in the results that we are on the right track.

## What is a physics simulation?

To really understand your simulations, you need to know what the code your are using is doing under the hood. This means understanding the core methods used in physics.

**The best way to learn is to code up these methods for yourself.**

# Commercial vs. OpenSource Packages