PLUTO: A NUMERICAL CODE FOR COMPUTATIONAL ASTROPHYSICS

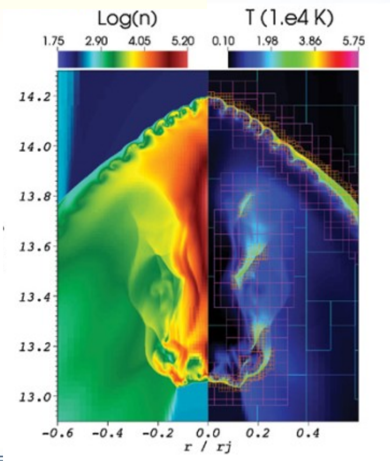
Somayeh Sheikhnezami

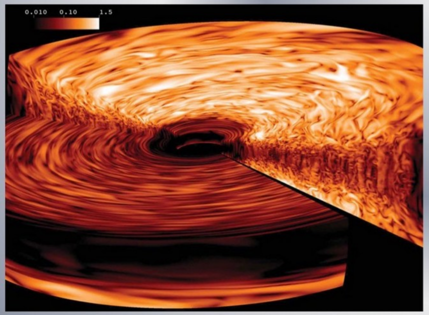
Institute for Advanced Studies in Basic Sciences (IASBS)

Workshop on computational astrophysical fluids

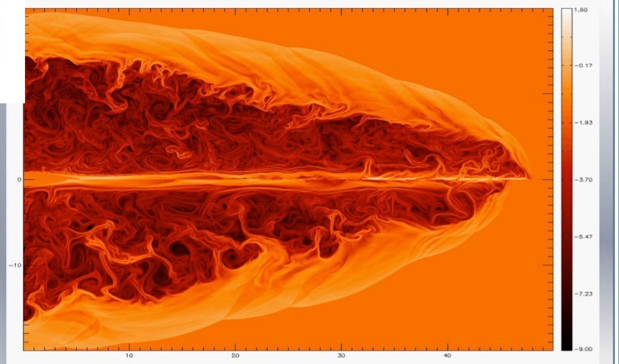
IPM – April 2025 (Ordibehesht 1404)

Radiative shocks

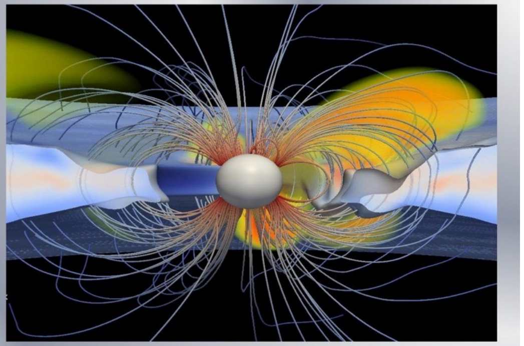




Turbulent r.m.s. velocity field in a 3D magnetized accretion disk. Flock et al., ApJ (2011) 735



Magnetic field strength in a 2D axisymmetric relativistic magnetized jet at the resolution of 40 zones per jet radius (R=1). The figure compares the results obtained with the HLLD solver (upper half) with the HLL solver (lower half), see Mignone et al, MNRAS (2009) 393, 1141.

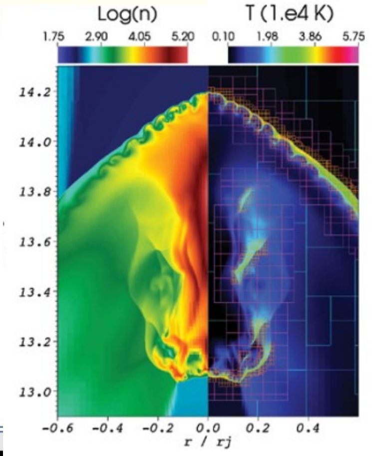


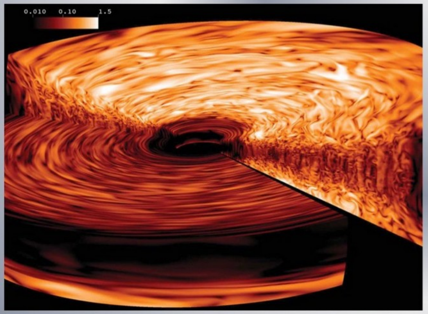
MHD modeling of flares occurring in young stellar objects (YSOs) with the aim to investigate the effects of the flares on the stability of the circumstellar disk surrounding the central protostar, Orlando et al., MNRAS (2011) 415, 3380.

Institute for Advanced Studies in Basic Sciences
Gava Zang, Zanjan, Iran

PLUTO: A NUMERICAL CODE FOR COMPUTATIONAL ASTROPHYSICS

Radiative shocks

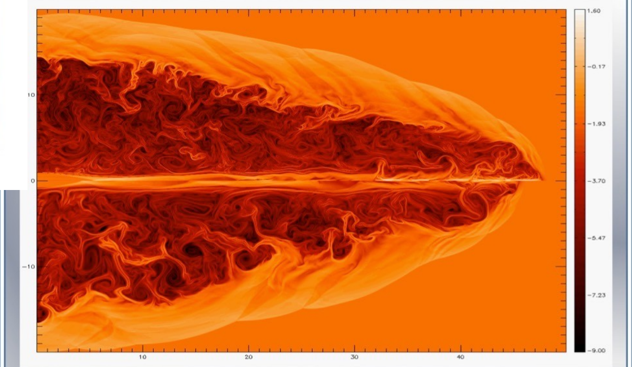




Turbulent r.m.s. velocity field in a 3D magnetized accretion disk. Flock et al., ApJ (2011) 735

PLUTO v. 4.4-patch3 (September 2024)

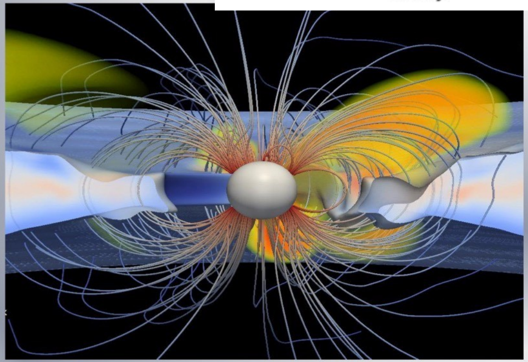
User's Guide https://plutocode.ph.unito.it/



Magnetic field strength in a 2D axisymmetric relativistic magnetized jet at the resolution of 40 zones per jet radius (R=1). The figure compares the results obtained with the HLLD solver (upper half) with the HLL solver (lower half), see Mignone et al, MNRAS (2009) 393, 1141.

Mignone et al. ApJS (2007), 170, 228-242;

Mignone et al, ApJS (2012), 198, 7



MHD modeling of flares occurring in young stellar objects (YSOs) with the aim to investigate the effects of the flares on the stability of the circumstellar disk surrounding the central protostar, Orlando et al., MNRAS (2011) 415, 3380.

Pluto Code: A high-resolution shock-capturing (HRSC) code

- PLUTO is a **finite-volume, shock-capturing** code designed to integrate a system of conservation laws

$$\frac{\partial U}{\partial t} + \nabla . T = S(U)$$

- The code is particularly suitable for time-dependent, explicit computations of **highly supersonic flows in the presence of strong discontinuities**, and it can be employed under different regimes, i.e., classical, relativistic unmagnetized, and magnetized flows.

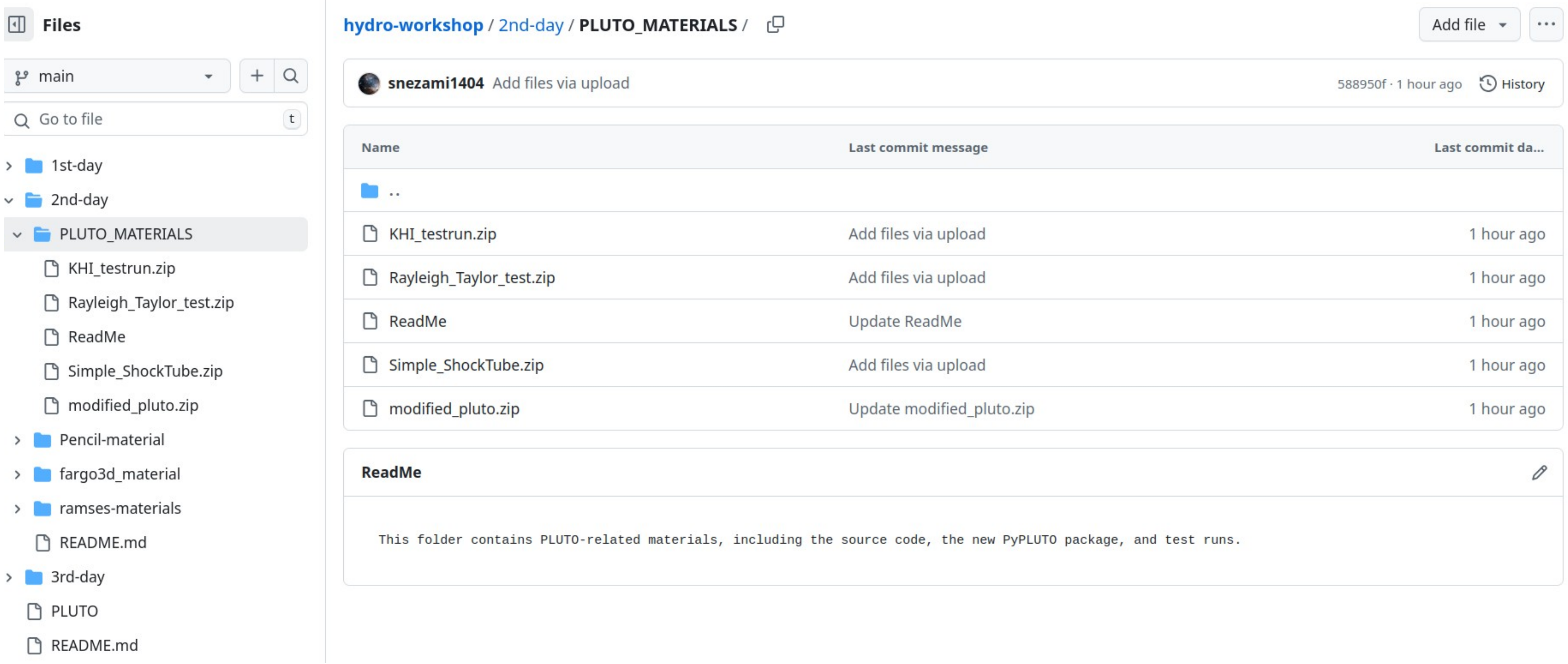
- PLUTO is written in C (~80,000 lines) and C++ (12,000 lines);

- Support multi-dimensional parallel (MPI) computations from single processor to a large number of cores (tested up to 262,144);

Downloading and unpacking PLUTO

- PLUTO can be downloaded from the following link **https://plutocode.ph.unito.it/**
- Once downloaded, extract all the files from the archive (pluto-4.4-patch3.tar.gz)

- Or you can download the file from the following repository:



Downloading and unpacking PLUTO

- PLUTO can be downloaded from the following link **https://plutocode.ph.unito.it/**
- Once downloaded, extract all the files from the archive (pluto-4.4-patch3.tar.gz):

~> gunzip pluto-4.4-patch3.tar.gz
~> tar -xvf pluto-4.4-patch3.tar.gz
This will create the folder PLUTO/ in your home directory.

- To get the address of your PLUTO code

Change the directory to PLUTO

~> cd pluto-4.4-patch3/PLUTO
~/pluto-4.4-patch3/PLUTO> **pwd**

- At this point, we advise to set the environment variable PLUTO_DIR to point to your code directory.

Depending on your shell (e.g. tcsh or bash) use either one of the following commands

~> export PLUTO_DIR="/home/user/PLUTO " # If you're using the bash shell;
~> setenv PLUTO_DIR ="/home/user/PLUTO " # If you're using the tcsh shell;

Or you can add the command in your bashrc to be known everywhere:
~> gedit .bashrc

- change the directory to $PLUTO_DIR and list the content

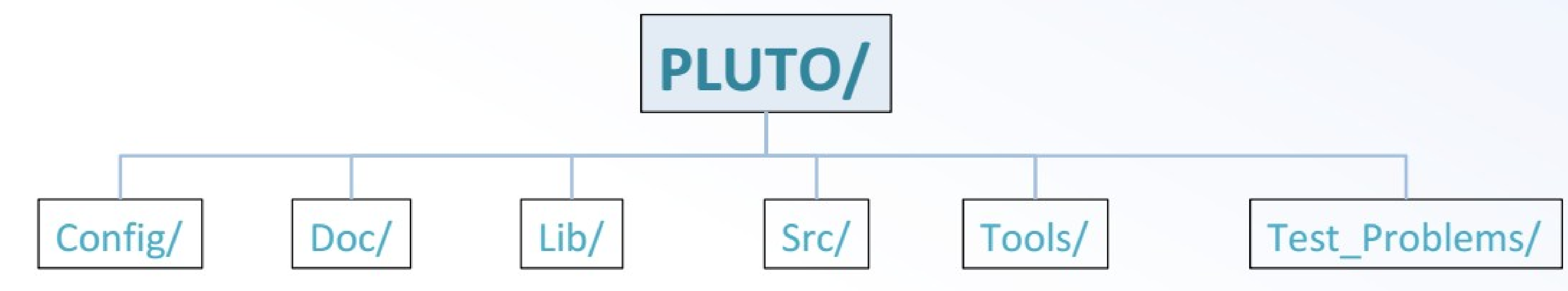
cd $PLUTO_DIR

~/pluto-4.4-patch3/PLUTO$ **ls -l**

- It should look like

```
Total 80

-rwxrwxrwx  1 nezami nezami 22404 Sep 18  2024 CHANGES
drwxrwxr-x  2 nezami nezami  4096 Nov  5 17:29 Config
-rw-r--r--  1 nezami nezami 17982 Apr 18  2011 COPYING
drwxrwxr-x  3 nezami nezami  4096 Feb 25 18:46 Doc
drwxrwxr-x  2 nezami nezami  4096 Nov  5 17:29 Lib
-rw-r--r--  1 nezami nezami  2091 Aug 22  2024 README
-rwxr-xr-x  1 nezami nezami  5304 Jul  7  2021 setup.py
drwxrwxr-x 21 nezami nezami  4096 Nov  5 17:29 Src
drwxrwxr-x  9 nezami nezami  4096 Nov  5 17:29 Test_Problems
drwxrwxr-x  7 nezami nezami  4096 Nov  5 17:29 Tools
```



In order to configure and setup PLUTO for a particular problem, four main steps have to be followed; the resulting configuration will then be stored in 4 different files, part of your local working directory:

definitions.h: header file containing all problem-dependent preprocessor directives required at compilation time (physics module, geometry, dimensions, etc.).

makefile: needed to compile PLUTO and it depends on your system architecture.

pluto.ini: startup initialization file containing run-time parameters (grid size, CFL,…)

init.c: implements initial, boundary conditions, etc.…

Lib/: repository for additional libraries;

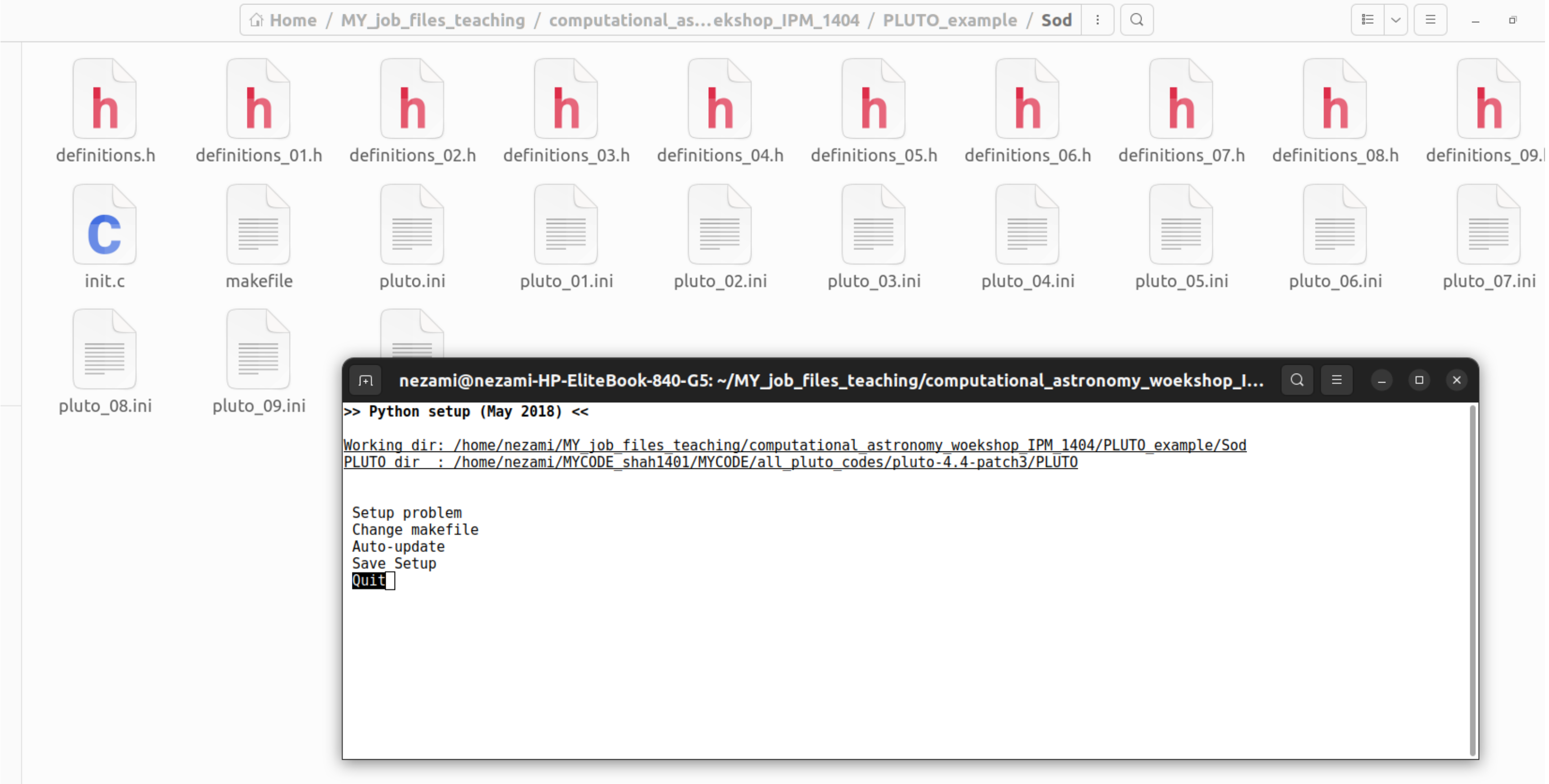
Src/:main repository for all *.c source files with the exception of the init.c file,

Running a simple shock-tube problem

- Change directory to the test problem (sod)

Cd   …./sod

- Run the Python script using
- ~…./sod> python $PLUTO_DIR/setup.py



Running a simple shock-tube problem

- Change directory to the test problem (sod)

Cd   …./shock_tube

- Run the Python script using
- ~…./sod> python $PLUTO_DIR/setup.py

- Then you need to compile the code for the current configuration and setup you chose
- ~…./sod> make

- You can now run the code by typing
- ~…./sod> ./pluto &

or

> mpirun -np 4 ./pluto &

- We can run Pluto with gcc or mpicc

Install OpenMPI
sudo apt update
sudo apt install openmpi-bin libopenmpi-dev

Visualization with PyPLUTO

• To install PyPLUTO (Python $\geq$3.10 required):

```
> cd $PLUTO_DIR/Tools/PyPLUTO/Src
> pip install -e ./
```

• This should install all the necessary packages (numpy, matplotlib, scipy, pandas).

• The -e command will prevent from installing the package every time is updated (simply replacing the files will be sufficient)

• If issues arise, try:

```
> cd $PLUTO_DIR/Tools/PyPLUTO/Src
> python setup.py install
```

• For LINUX UBUNTU 24.04 LTS system

You can try:

```
>pip install -e ./ --break-system-packages
```

Visualization with PyPLUTO

- All output files can be loaded with PyPLUTO:

```
> import pyPLUTO as pp
> D = pp.Load(0)
> I = pp.Image().plot(D.x1,D.rho)
> pp.show(block = False)
```

- This will produce a 1D plot (x,y) of the initial condition ('0000').

```
>import pyPLUTO as pp
> D = pp.Load()
> I = pp.Image()
> I.plot(D.x1, D.rho, label = 'rho', legpos = 0)
> I.plot(D.x1, D.prs, xtitle = 'x')
> pp.show()
```

RAYLEIGH-TAYLOR INSTABILITY

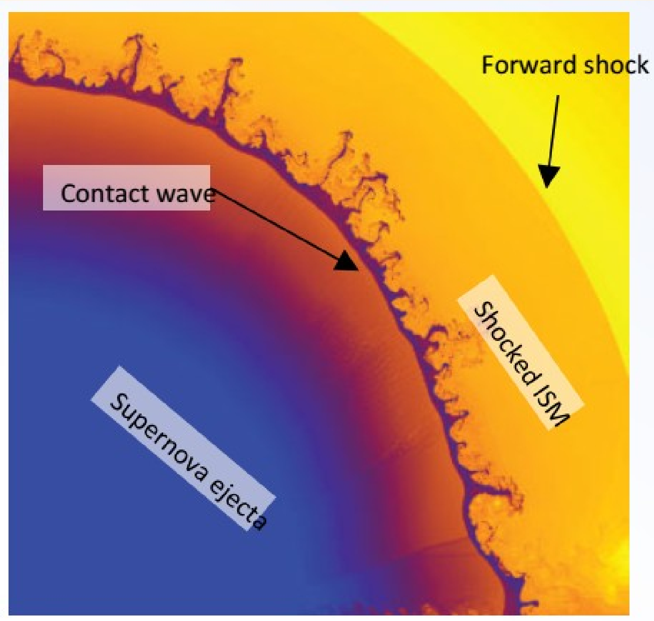
- The Rayleigh–Taylor instability (RTI) is a classical fluid instability that occurs at the interface between two fluids of different densities when the lighter fluid pushes against the heavier one under the influence of gravity or an effective acceleration.

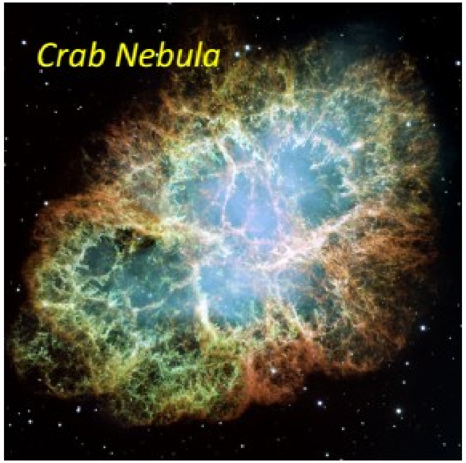
- The amplitude growths and the further upward motion of the lighter fluid assumes the form of rising bubble while the sinking fluid become finger-shaped

- As the instability proceeds, fingers evolve into mushroom-like vortex motion accompanied by secondary shear flow instabilities.

- n a supernova (SN) explosion a large amount of energyis released resulting in a formation of a large scale SN remnant (SNR)

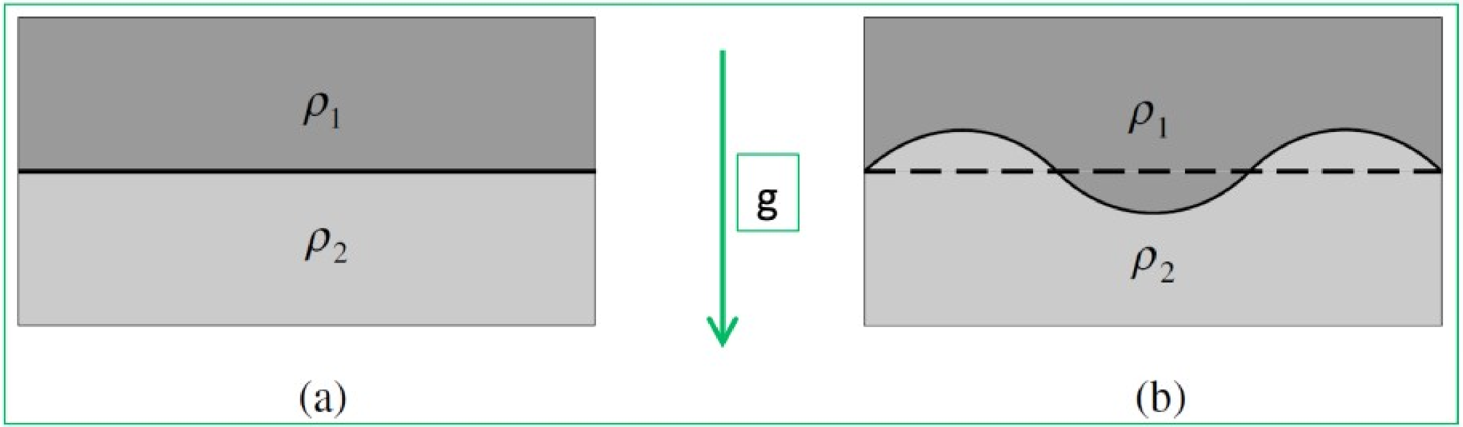
- The dense shell of ejected material decelerates in a rarefied interstellar medium (ISM) and is unstable to RT-type instabilities.





RAYLEIGH-TAYLOR INSTABILITY (RTI:Analysis)

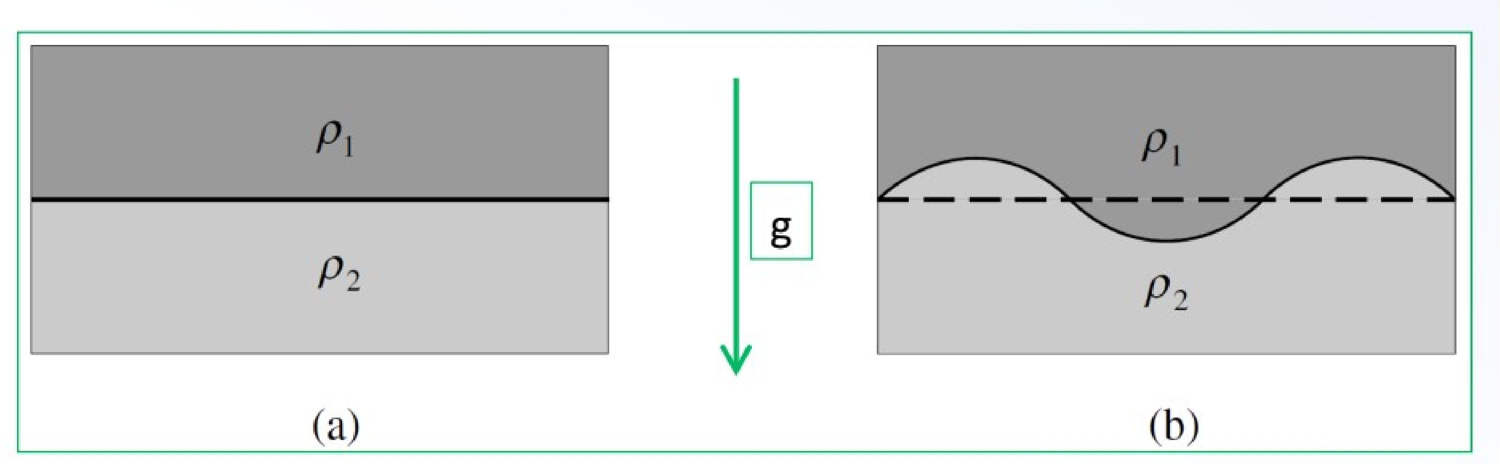
Consider the case where two fluids with densities ρ1 and ρ2 are on top of each other .



This is an equilibrium situation if the stratification is supported by a pressure gradient:

- Is this a stable equilibrium ?

RAYLEIGH-TAYLOR INSTABILITY (RTI:Analysis)

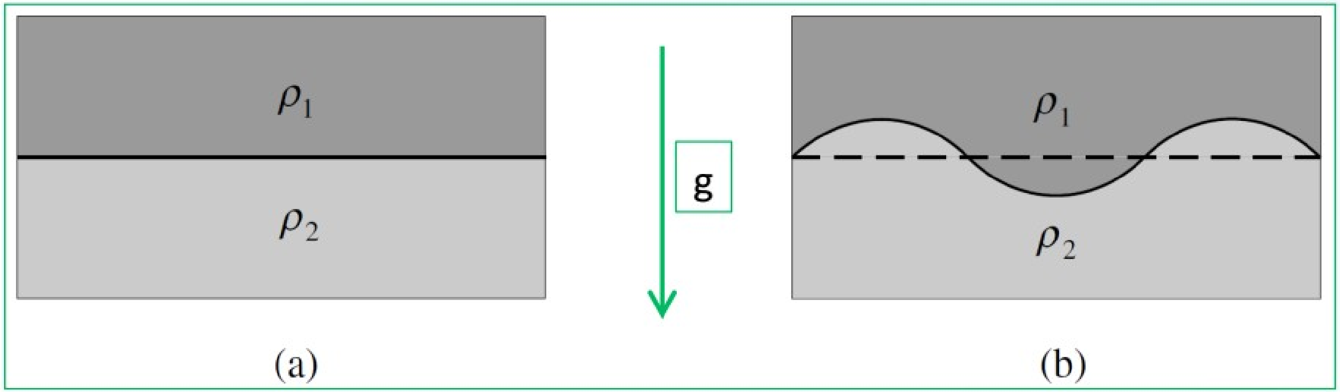


Let’s now perturb the equilibrium by rippling the boundary layer.

- The fluid element of density ρ1 has moved downwards with consequent loss of gravitational potential energy while the opposite is true of the fluid element of density ρ2.

- It is intuitively obvious that the only stable equilibrium is to have the denser fluid supporting the less dense.The instability that arises when ρ2ρ1 is called the Rayleigh–Taylor instability.

RTI: Linear Analysis



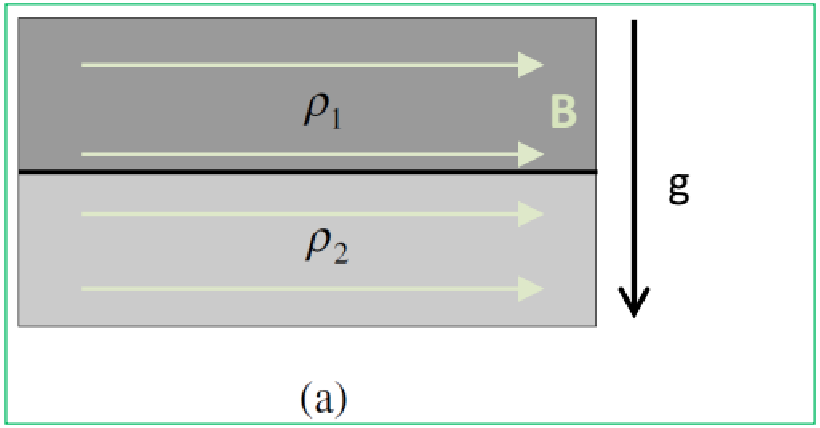
From normal mode analysis it is found that, in absence of magnetic field,

$$\boxed{\omega^2 = -\frac{kg(\rho_1 - \rho_2)}{\rho_1 + \rho_2}}$$

- If ρ1 < ρ2 the equilibrium is stable a surface gravity waves
- If ρ1>ρ2   the equilibrium is unstable a Rayleigh-Taylor instability

RTI: Effects of Magnetic Fields

In a plasma, density and pressure are tied, magnetic field can change pressure but not density. From normal mode analysis it is found that, in absence ofmagnetic field,

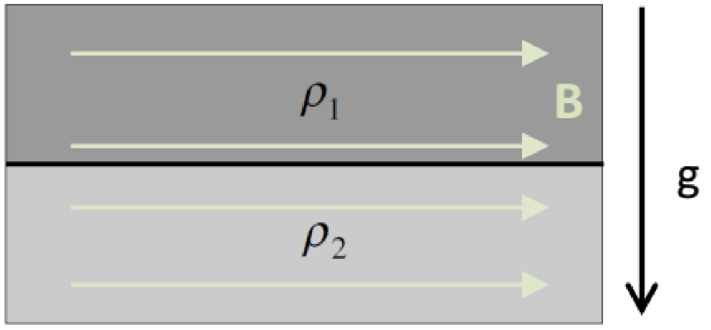


The presence of a uniform magnetic field has the effects of reducing the growth rate of modes parallel to it, although the interface still remains Rayleigh–Taylor unstable in the perpendicular direction.

RTI: Effects of Magnetic Fields

The growth rate now becomes

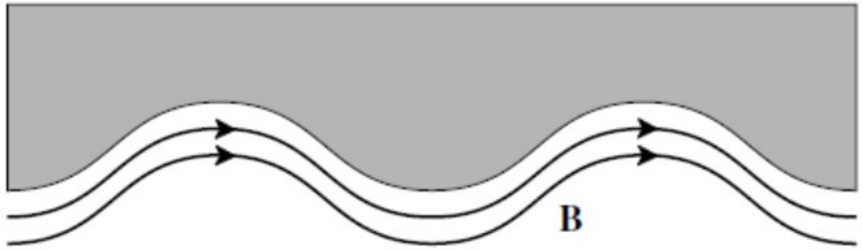
$$\omega^2 = -|\boldsymbol{k}|\, g \frac{\rho_1 - \rho_2}{\rho_1 + \rho_2} + 2\frac{(\boldsymbol{k} \cdot \boldsymbol{B}_0)^2}{4\pi(\rho_1 + \rho_2)}$$



(a)

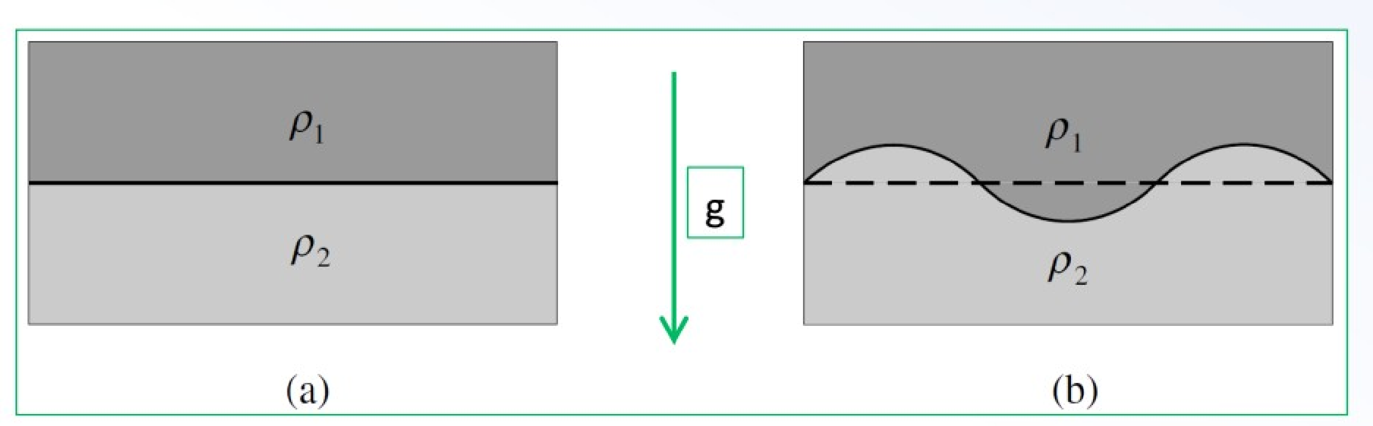
Shorter wave modes such that $k \geq \dfrac{\mu_0 g(\rho_1 - \rho_2)}{2B_0^2 \cos^2\theta}$ are stabilized

Bending of field lines resist to the growth of perturbation:



(b)

RTI: Linear Analysis



From normal mode analysis it is found that, in absence of magnetic field,

$$\boxed{\omega^2 = -\frac{kg(\rho_1 - \rho_2)}{\rho_1 + \rho_2}}$$

- If ρ1 < ρ2 the equilibrium is stable a surface gravity waves
- If ρ1>ρ2   the equilibrium is unstable a Rayleigh-Taylor instability

Rayleigh-Taylor instability