

PLUTO: A NUMERICAL CODE FOR COMPUTATIONAL ASTROPHYSICS

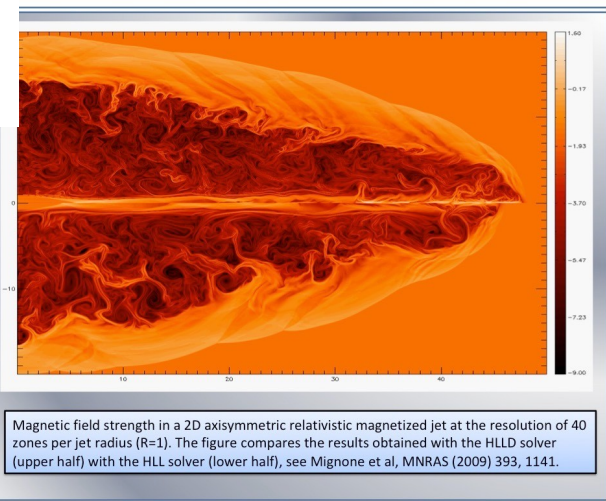
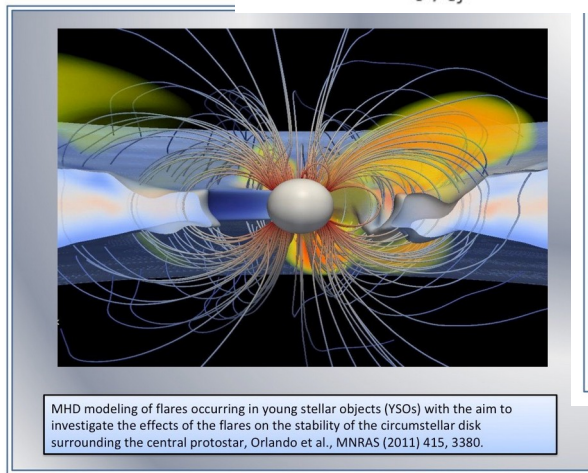
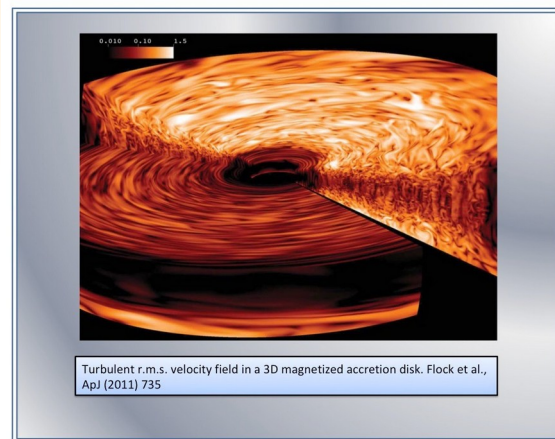
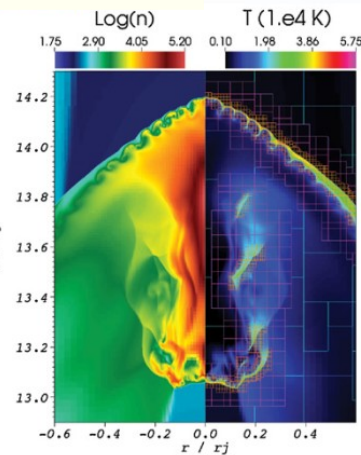
Somayeh Sheikhezami

Institute for Advanced Studies in Basic Sciences (IASBS)

Workshop on computational astrophysical fluids

IPM – April 2025 (Ordibehesht 1404)

Radiative shocks



PLUTO: A NUMERICAL CODE FOR COMPUTATIONAL ASTROPHYSICS

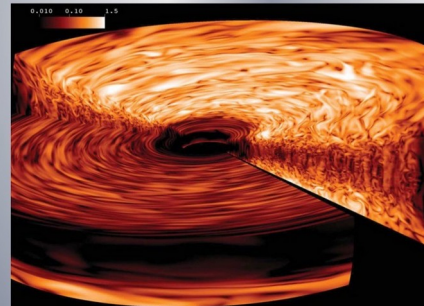
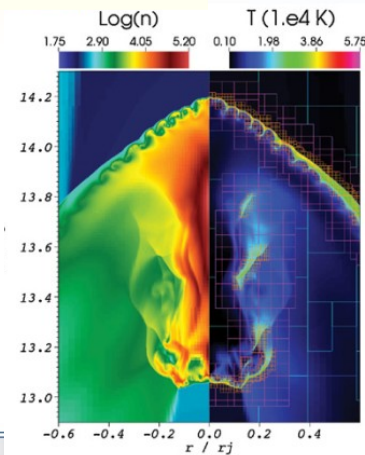
PLUTO v. 4.4-patch3 (September 2024)

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

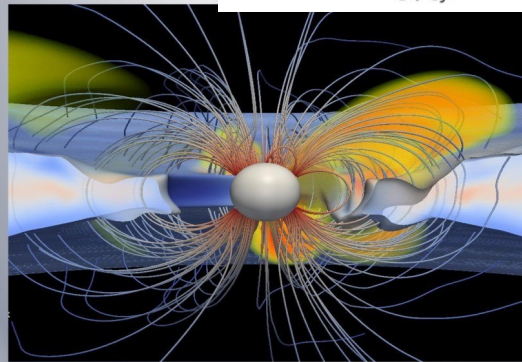
Mignone et al. ApJS (2007), 170, 228-242;

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

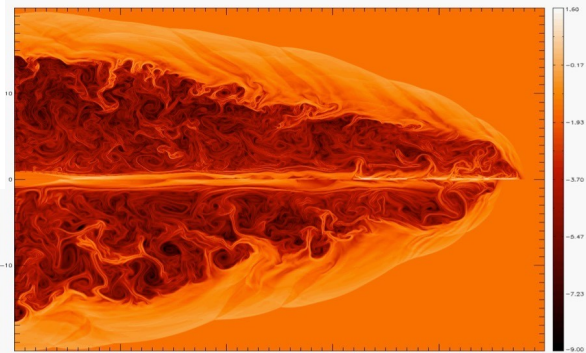
Radiative shocks



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



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.



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.

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

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

- 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:

Files

main

Go to file

1st-day

2nd-day

PLUTO_MATERIALS

KHI_testrun.zip

Rayleigh_Taylor_test.zip

ReadMe

Simple_ShockTube.zip

modified_pluto.zip

Pencil-material

fargo3d_material

ramses-materials

README.md

3rd-day

PLUTO

README.md

hydro-workshop / 2nd-day / PLUTO_MATERIALS /

Add file

...

snezami1404 Add files via upload 588950f · 1 hour ago History

Name	Last commit message	Last commit da...
..		
KHI_testrun.zip	Add files via upload	1 hour ago
Rayleigh_Taylor_test.zip	Add files via upload	1 hour ago
ReadMe	Update ReadMe	1 hour ago
Simple_ShockTube.zip	Add files via upload	1 hour ago
modified_pluto.zip	Update modified_pluto.zip	1 hour ago

ReadMe

This folder contains PLUTO-related materials, including the source code, the new PyPLUTO package, and test runs.

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

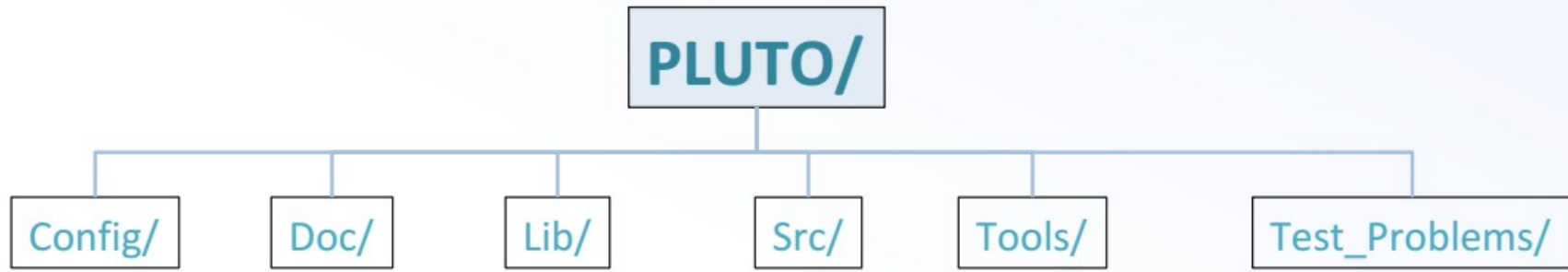
```
cd $PLUTO_DIR
```

```
~/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)

Cdshock_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 &

To Install mpi library

sudo apt update

sudo apt-get install libcr-dev mpich2 mpich2-doc

Running a simple shock-tube problem

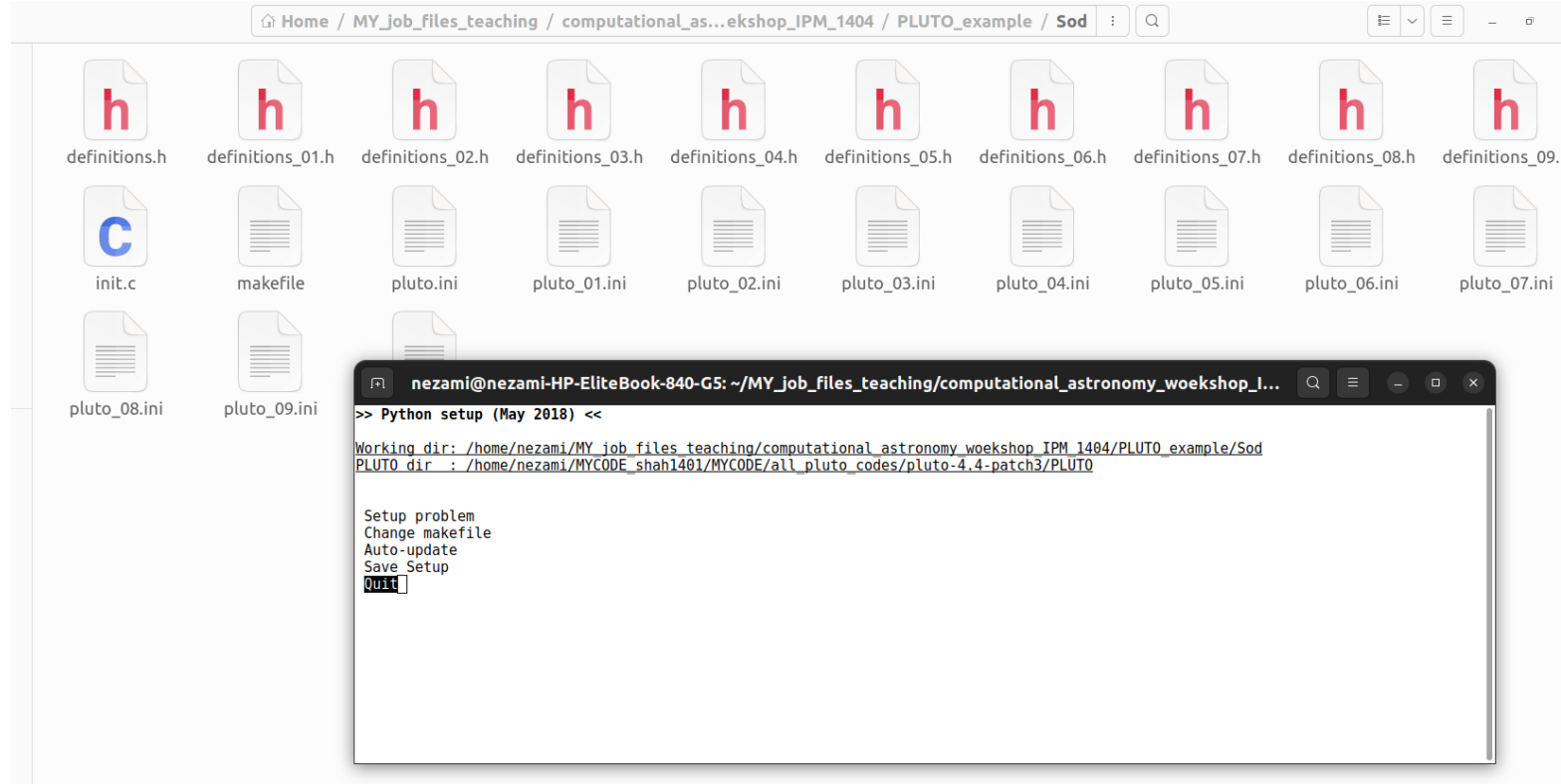
↵ Change directory to the test problem (sod)

`Cdsod`

↵ Run the Python script using

↵ `~....sod> python $PLUTO_DIR/setup.py`

↵





definitions.h



definitions_01.h



definitions_02.h



definitions_03.h



definitions_04.h



definitions_05.h



definitions_06.h



definitions_07.h



definitions_08.h



definitions_09.h



init.c



pluto.ini



pluto_01.ini



pluto_02.ini



pluto_03.ini



pluto_04.ini



pluto_05.ini



pluto_06.ini



pluto_07.ini



pluto_08.ini



pluto_09.ini



sysconf.out

nezami@nezami-HP-EliteBook-840-G5: ~/MY_job_files_teaching/computational_astronomy_workshop_I...



>> Setup problem <<

PHYSICS	HD
DIMENSIONS	1
GEOMETRY	CARTESIAN
BODY_FORCE	NO
COOLING	NO
RECONSTRUCTION	LINEAR
TIME_STEPPING	RK2
NTRACER	0
PARTICLES	NO
USER_DEF_PARAMETERS	1



definitions.h



definitions_01.h



definitions_02.h



definitions_03.h



definitions_04.h



definitions_05.h



definitions_06.h



definitions_07.h



definitions_08.h



definitions_09.h



init.c



pluto.ini



pluto_01.ini



pluto_02.ini



pluto_03.ini



pluto_04.ini



pluto_05.ini



pluto_06.ini



pluto_07.ini



pluto_08.ini



pluto_09.ini



sysconf.out

nezami@nezami-HP-EliteBook-840-G5: ~/MY_job_files_teaching/computational_astronomy_workshop_I...



>> HD Menu <<

DUST_FLUID	NO
EOS	IDEAL
ENTROPY_SWITCH	NO
INCLUDE_LES	NO
THERMAL_CONDUCTION	NO
VISCOSITY	NO
RADIATION	NO
ROTATING_FRAME	NO



definitions.h



definitions_01.h



definitions_02.h



definitions_03.h



definitions_04.h



definitions_05.h



definitions_06.h



definitions_07.h



definitions_08.h



definitions_09.h



init.c



makefile



pluto.ini



pluto_01.ini



pluto_02.ini



pluto_03.ini



pluto_04.ini



pluto_05.ini



pluto_06.ini



pluto_07.ini



pluto_08.ini



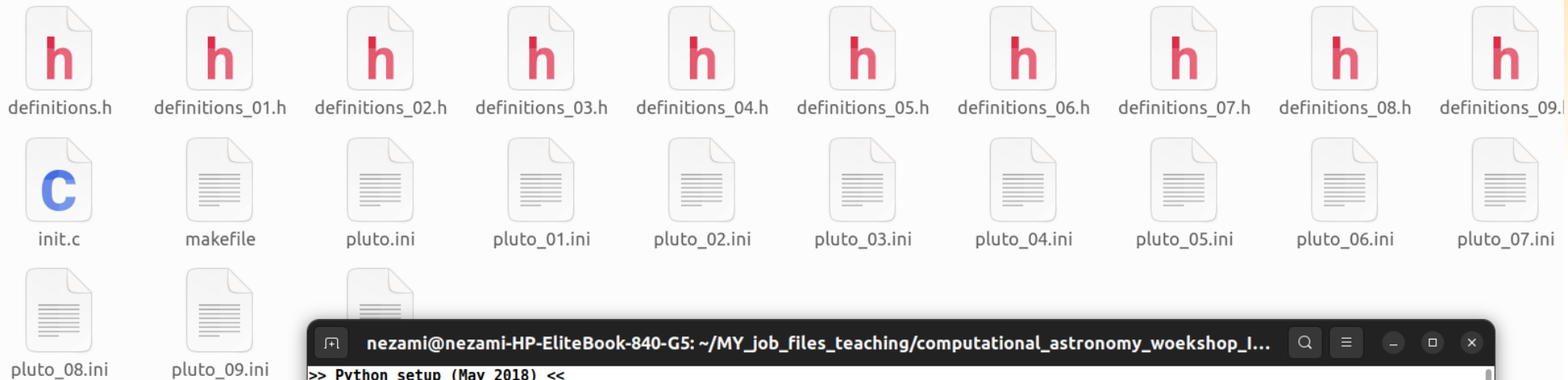
pluto_09.ini

nezami@nezami-HP-EliteBook-840-G5: ~/MY_job_files_teaching/computational_astronomy_workshop_I...



>> Change makefile <<

```
Darwin.gcc.defs
Darwin.mpicc.defs
Linux.gcc.defs
Linux.mpicc.defs
Template.defs
debug.defs
profile.defs
```



```
nezami@nezami-HP-EliteBook-840-G5: ~/MY_job_files_teaching/computational_astronomy_workshop_I...
>> Python setup (May 2018) <<

Working dir: /home/nezami/MY_job_files_teaching/computational_astronomy_workshop_IPM_1404/PLUTO_example/Sod
PLUTO dir  : /home/nezami/MYCODE_shah1401/MYCODE/all_pluto_codes/pluto-4.4-patch3/PLUTO

Setup problem
Change makefile
Auto-update
Save Setup
Quit
```

Open ▾



pluto.ini

~/MY_job_files_teaching/computational_astrono...404/PLUTO_example/Shock_tube/Simple_ShockTube

Save



```
1 [Grid]
2
3 X1-grid    1    0.0    400    u    1.0
4 X2-grid    1    0.0      1    u    1.0
5 X3-grid    1    0.0      1    u    1.0
6
7 [Chombo Refinement]
8
9 Levels      4
10 Ref_ratio   2  2  2  2  2
11 Regrid_interval 2  2  2  2
12 Refine_thresh 0.3
13 Tag_buffer_size 3
14 Block_factor 4
15 Max_grid_size 32
16 Fill_ratio  0.75
17
18 [Time]
19
20 CFL          0.8
21 CFL_max_var  1.1
22 tstop        1
23 first_dt     1.e-4
24
25 [Solver]
26
27 Solver        roe
28
29 [Boundary]
30
31 X1-beg        outflow
32 X1-end        outflow
33 X2-beg        periodic
34 X2-end        periodic
35 X3-beg        outflow
36 X3-end        outflow
37
38 [Static Grid Output]
39
40 uservar       0
41 dbl           0.2  -1  single_file
42 flt           -1.0 -1  single_file
43 vtk           -1.0 -1  single_file
44 tab           -1.0 -1
```



Open ▾

definitions.h
~/MY_job_files_teaching/computational_astro...4/PLUTO_example/Shock_tube/Simple_ShockTube

Save



```
1 #define PHYSICS HD
2 #define DIMENSIONS 1
3 #define GEOMETRY CARTESIAN
4 #define BODY_FORCE NO
5 #define COOLING NO
6 #define RECONSTRUCTION LINEAR
7 #define TIME_STEPPING RK2
8 #define NTRACER 0
9 #define PARTICLES NO
10 #define USER_DEF_PARAMETERS 1
11
12 /* -- physics dependent declarations -- */
13
14 #define DUST_FLUID NO
15 #define EOS IDEAL
16 #define ENTROPY_SWITCH NO
17 #define INCLUDE_LES NO
18 #define THERMAL_CONDUCTION NO
19 #define VISCOSITY NO
20 #define RADIATION NO
21 #define ROTATING_FRAME NO
22
23 /* -- user-defined parameters (labels) -- */
24
25 #define SCRH 0
26
27 /* [Beg] user-defined constants (do not change this line) */
28
29 #define INTERNAL_BOUNDARY NO
30 #define LIMITER MINMOD_LIM
31 #define CHECK_DIVB_CONDITION NO
32 #define ASSIGN_VECTOR_POTENTIAL NO
33 #define UPDATE_VECTOR_POTENTIAL NO
34 #define WARNING_MESSAGES NO
35 #define PRINT_TO_FILE YES
36 #define MULTIPLE_LOG_FILES NO
37
38 /* [End] user-defined constants (do not change this line) */
```


Open

*init.c

~/MYCODE_shah1401/MYCODE/all_pluto_codes/pluto-4.4-patch3/PLUTO/Src/Templates

Save

```
1
2 /* ////////////////////////////////////// */
3 /*!
4 |file
5 |brief Contains basic functions for problem initialization.
6
7 The init.c file collects most of the user-supplied functions useful
8 for problem configuration.
9 It is automatically searched for by the makefile.
10
11 |author A. Mignone (andrea.mignone@unito.it)
12 |date March 5, 2017
13 */
14 /* ////////////////////////////////////// */
15 #include "pluto.h"
16
17 /* ***** */
18 void Init (double *v, double x1, double x2, double x3)
19 /*!
20 * The Init() function can be used to assign initial conditions as
21 * as a function of spatial position.
22 *
23 * |param [out] v a pointer to a vector of primitive variables
24 * |param [in] x1 coordinate point in the 1st dimension
25 * |param [in] x2 coordinate point in the 2nd dimension
26 * |param [in] x3 coordinate point in the 3rd dimension
27 *
28 * The meaning of x1, x2 and x3 depends on the geometry:
29 * |f[ |begin{array}{cccl}
30 * x_1 & x_2 & x_3 & \mathrm{Geometry} & || \mathrm{noalign}\mathrm{medskip}
31 * \hline
32 * x & y & z & \mathrm{Cartesian} & || \mathrm{noalign}\mathrm{medskip}
33 * R & z & - & \mathrm{cylindrical} & || \mathrm{noalign}\mathrm{medskip}
34 * R & \phi & z & \mathrm{polar} & || \mathrm{noalign}\mathrm{medskip}
35 * r & \theta & \phi & \mathrm{spherical} &
36 * \end{array}
37 * |f]
38 *
39 * Variable names are accessed by means of an index v[nv], where
40 * nv = RHO is density, nv = PRS is pressure, nv = (VX1, VX2, VX3) are
41 * the three components of velocity, and so forth.
42 *
43 ***** */
44 {
```

Visualization with PyPLUTO

- To install PyPLUTO (Python ≥ 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:

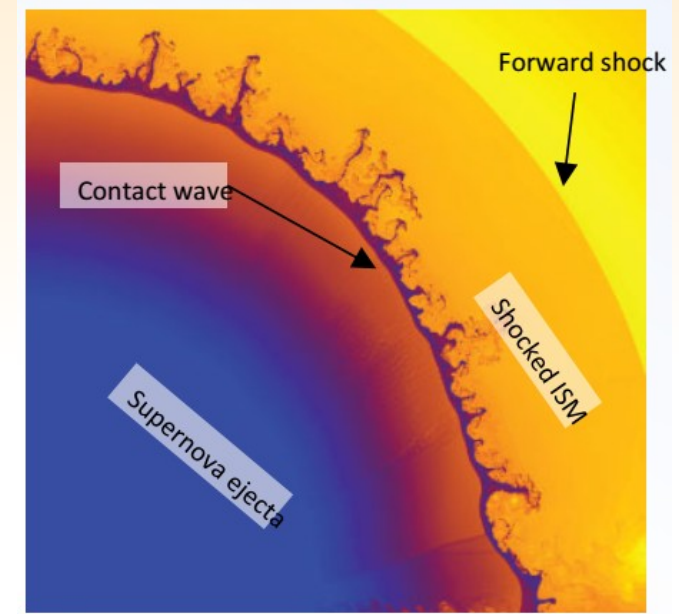
```
import pyPLUTO as pp
D = pp.Load(0)
I = pp.Image().plot(D.x1,D.rho)
pp.show(block = False)
```

- 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()
```

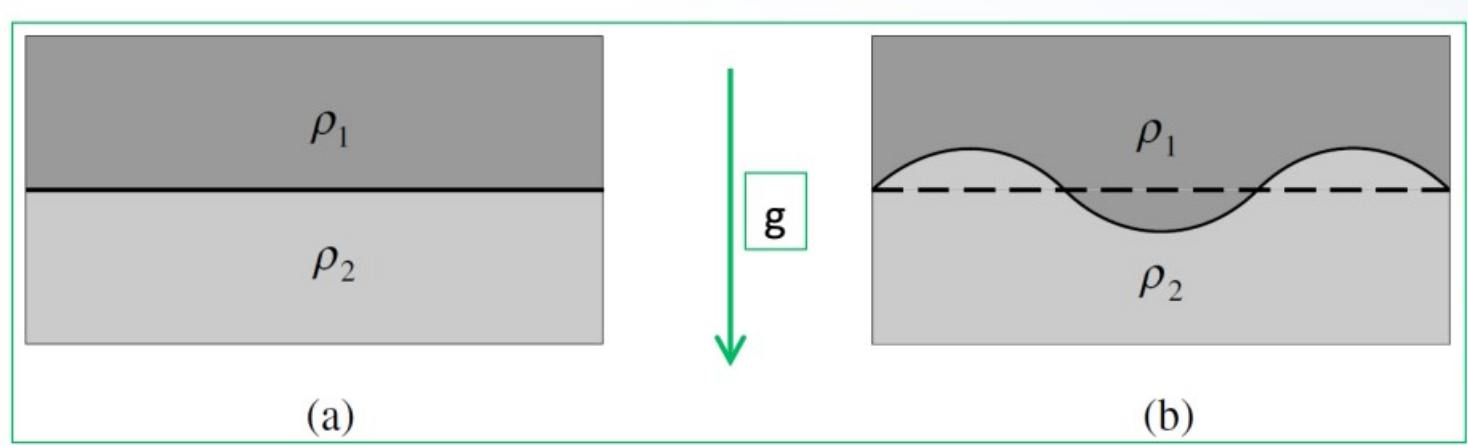
RAYLEIGH-TAYLOR INSTABILITY

- 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 grows and the further upward motion of the lighter fluid assumes the form of rising bubble while the sinking fluid becomes finger-shaped.
- As the instability proceeds, fingers evolve into mushroom-like vortex motion accompanied by secondary shear flow instabilities.
- In a supernova (SN) explosion a large amount of energy is released resulting in the 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)

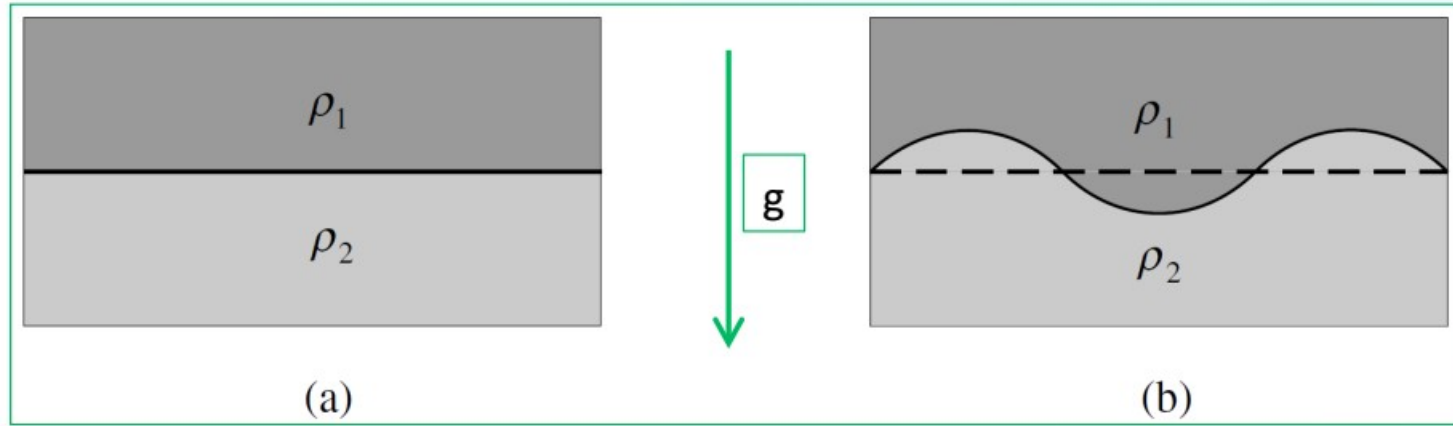
Consider the case where two fluids with densities ρ_1 and ρ_2 are on top of each other .



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

- Is this a stable equilibrium ?

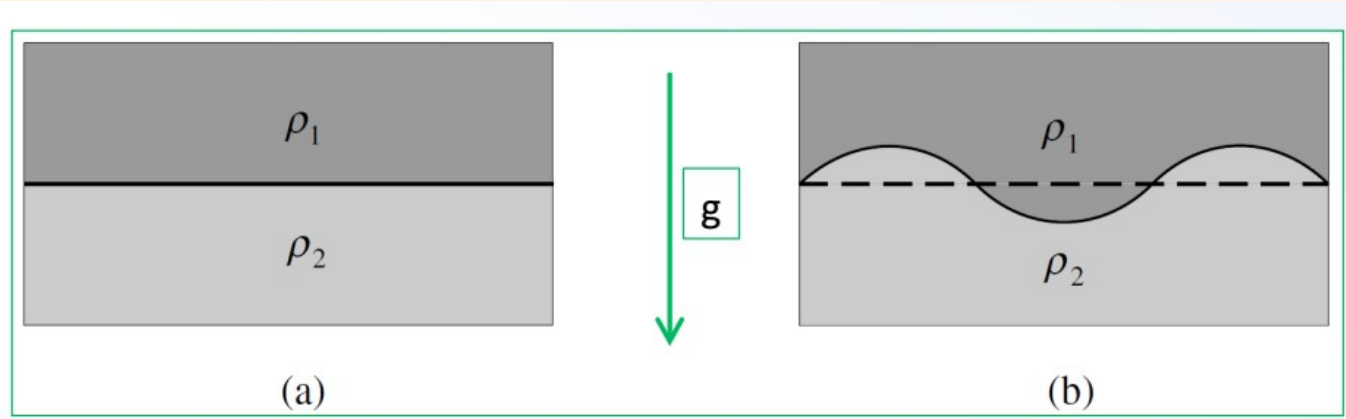
RAYLEIGH-TAYLOR INSTABILITY (RTI:Analysis)



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

- The fluid element of density ρ_1 has moved downwards with consequent loss of gravitational potential energy while the opposite is true of the fluid element of density ρ_2 .
- It is intuitively obvious that the only stable equilibrium is to have the denser fluid supporting the less dense. The instability that arises when $\rho_2 > \rho_1$ is called the Rayleigh–Taylor instability.

RTI: Linear Analysis



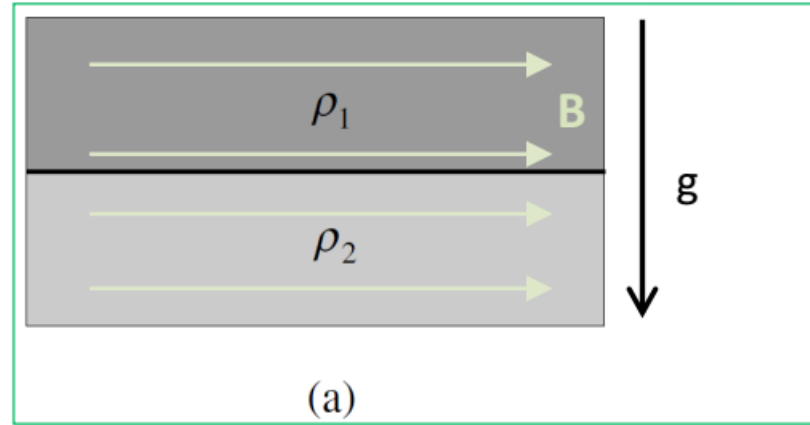
From normal mode analysis it is found that, in absence of magnetic field,

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

- If $\rho_1 < \rho_2$ the equilibrium is stable a surface gravity waves
- If $\rho_1 > \rho_2$ 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 of magnetic field,

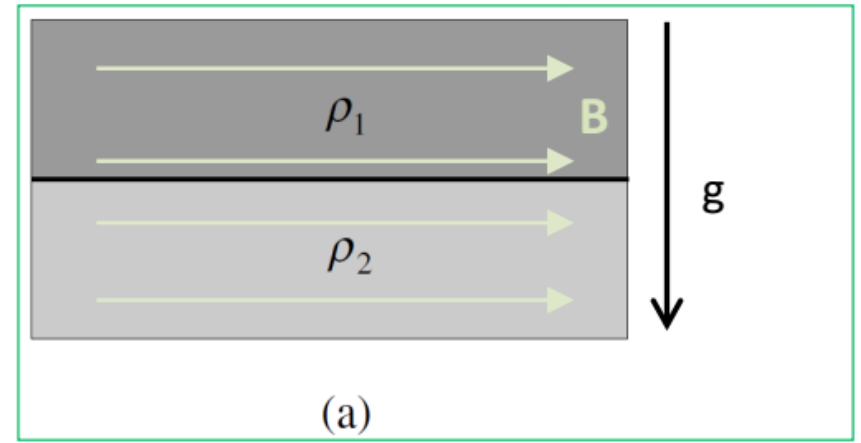


The presence of a uniform magnetic field has the effects of reducing the growth rate of modes parallel to it, although the interface still remains Rayleigh–Taylor unstable in the perpendicular direction.

RTI: Effects of Magnetic Fields

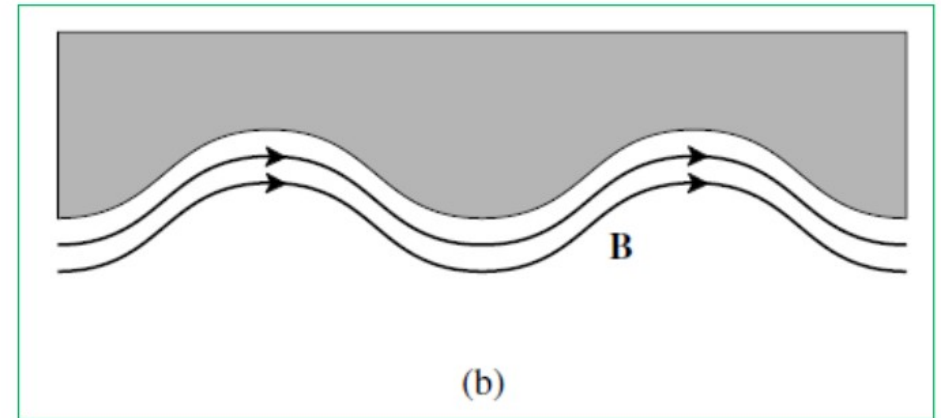
The growth rate now becomes

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

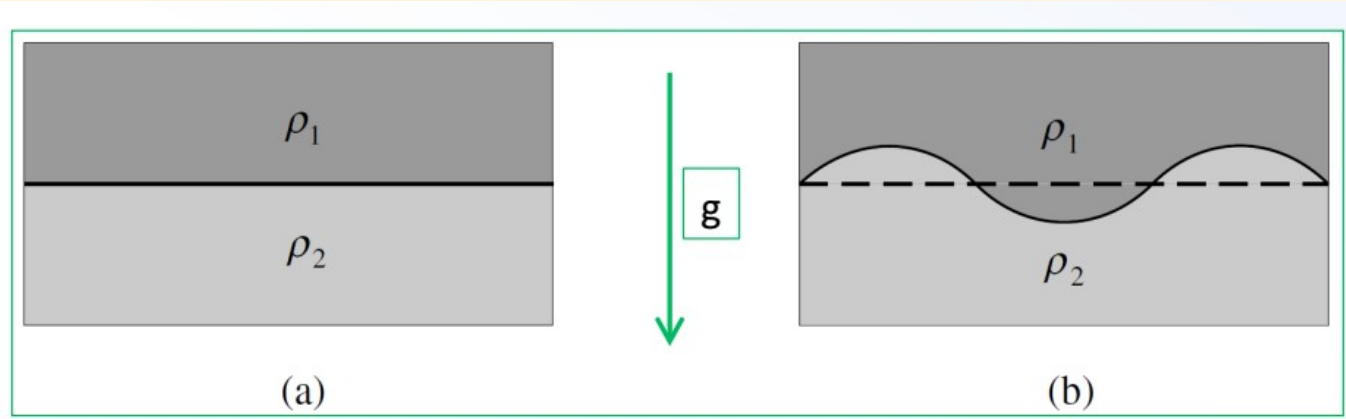


Shorter wave modes such that $k \geq \frac{\mu_0 g (\rho_1 - \rho_2)}{2B_0^2 \cos^2 \theta}$ are stabilized

Bending of field lines resist to the growth of perturbation:



RTI: Linear Analysis



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

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

- If $\rho_1 < \rho_2$ the equilibrium is stable a surface gravity waves
- If $\rho_1 > \rho_2$ the equilibrium is unstable a Rayleigh-Taylor instability

Rayleigh-Taylor instability

