

Introduction to PLUTO

JONATAN MARTÍN*

Industrial Training - Master in Numerical Methods for Engineering (UPC)

jonatanmrtn@gmail.com

1. Introduction

I carried out my training in the Institute of Space Sciences (Campus UAB, Bellaterra) under the supervision of Prof. Dr. Diego F. Torres, the director of the institute. During this time we agreed to make the following tasks:

1. To install, understand the basics and be able to run PLUTO,
2. To configure some examples of shocks (e.g., a simple non-relativistic strong shock) and apply this to some astrophysical problems,
3. To document all the process to facilitate continuity into this line of research.

In the next section, I summarize my working plan. As it was part of my training, in the subsequent sections I describe briefly the PLUTO code and its installation, basic set-up, visualization tools and physical problems that can be faced with it. I also explain two particular astrophysical examples performed with the code and, finally, I enumerate all the material in terms of code and documentation produced and introduce the forthcoming project in which I am involved as a fruit of this work.

2. Working plan

My internship took place from May 17th 2016 to August 9th 2016. I was there for 60 working days, 7.5 hours per day, making a total of 450 hours, the number of hours required to pass the subject. The approximate timeline of my training was the following:

1. **From May 17th to May 20th:** installation of the PLUTO and the libraries required.
2. **From May 23rd to June 10th:** read and study the PLUTO manual, understand the most important physics modules and their parameters. Make a detailed document for the installation of PLUTO and the required libraries.
3. **From June 13th to July 1st:** read and understand the paper of Truelove & McKee 1999 about simulating non-radiative shocks in Supernova Remnants (SNRs) and write the code necessary in PLUTO to reproduce the trajectories of the forward and reverse shocks in a SNR.
4. **From July 4th to July 15th:** perform the simulations and plot the graphs of the trajectories varying different parameters (physical and non-physical) to see the behaviour of the code and document all the tests done.
5. **From July 18th to August 9th:** read the last papers in the literature about RMHD simulations for Pulsar Wind Nebulae (PWNe), understand the problem set-up (initial and boundary conditions) and the equations to calculate from the RMHD solution the synchrotron radiation emitted by these objects, and start a new code relying on PLUTO to perform a simulation of this kind.

*Online student

3. The PLUTO code

PLUTO is a freely distributed software for the numerical solution of mixed hyperbolic/parabolic systems of partial differential equations (PDE conservation laws) targeting high Mach number flows in astrophysical fluid dynamics (Mignone et al. 2007, 2012). The software is developed at the Dipartimento di Fisica, Torino University in a joint collaboration with the Istituto Nazionale di Astrofisica (INAF), Osservatorio Astronomico di Torino and the Supercomputing Applications and Innovation (SCAI) Department of CINECA.

The code is designed with a modular and flexible structure whereby different numerical algorithms can be separately combined to solve systems of conservation laws using the finite volume or finite difference approach based on Godunov-type schemes. Equations are discretized and solved on a structured mesh that can be either static or adaptive. The code is written in C while the adaptive mesh refinement (AMR) interface requires also C++ and Fortran.

PLUTO can be run from a single workstation up to several thousands processors using the Message Passing Interface (MPI).

3.1. Installation

In order to work with PLUTO in all its capability we need to install supplementary libraries which allow to manage the produced data and run the code in parallel.

The minimal prerequisites to install and run the code are a C compiler, the GNU make and a Python compiler. In order to make possible the parallel running, we use the MPI libraries provided by OpenMPI. Finally, to use AMR the code is supported on the Chombo libraries, which require a C++ compiler, a Fortran compiler and HDF5 libraries which allows to store and manage data.

All these tools were installed in a Mac OS X El Capitan and a Linux Ubuntu 14.04 computers successfully.

3.2. Basic physics modules

PLUTO can solve the fluid equations under 4 different regimes: Hydrodynamics (HD), Magneto-hydrodynamics (MHD), Relativistic Hydrodynamics (RHD) and Relativistic Magneto-hydrodynamics (RMHD).

In the HD module, PLUTO solves the Euler equations for a classical fluid. These equations are written using a set of conservative variables: the mass density ρ , the momentum density $\mathbf{m} = \rho \mathbf{v}$ and the total energy density E . To close the system of equations we include the equation of state (EOS) such that $\rho e = \rho e(p, \rho)$, where e is the internal energy density and p the thermal pressure. There are already implemented different forms for the EOS (isothermal, ideal, PVTE-law, Taub-Matthews law or tabulated). The initial conditions are given using a set of primitive variables, in this case, the initial velocity \mathbf{v} , pressure p and density ρ profiles.

In the MHD module, it is introduced a continuity equation for the magnetic field \mathbf{B} , and additional terms coupled with \mathbf{B} in the HD equations. The initial magnetic field can be provided in terms of the potential vector \mathbf{A} , such that $\mathbf{B} = \nabla \times \mathbf{A}$. When we solve this equations is important to control the null divergence of the magnetic field condition. PLUTO provides three different algorithms to accomplish this condition at each time step (eight-wave formulation, hyperbolic divergence cleaning or constrained transport). The user can choose the most accurate one depending on the problem it is facing.

Finally, the RHD and RMHD modules implement the relativistic versions of the latter two. The equation are written in the laboratory frame (observer's frame) and the relation between the conservative and primitive variables differ due to Doppler boosting.

3.3. Problem header: definitions.h

The file definitions.h is created by the PLUTO setup script and it contains basic information about the characteristics of the problem that we want to solve and advanced parameters related with the nu-

merical method that we want to use. The header is structured in 4 main blocks: the basic options, where we introduce the information about the physics module (HD, MHD, RHD or RMHD), dimensions and components of our problem, geometry (cartesian, cylindrical or spherical), body forces included (described by a scalar potential, a vector or both), cooling effects due to optically thin thermal losses (power law model, tabulated or other implemented specific models), spatial reconstruction (flat, linear, WENO3, LimO3 and parabolic), time stepping (1st order explicit Euler algorithm, RK2, RK3, Hancock), dimensional splitting, number of tracers (scalar parameters obeying simple advection equations) and user-defined parameters; the physics-dependent options as the background magnetic fields (only available in MHD and RMHD modules), divergence of the magnetic field control and equation of state (already described in the previous section), entropy switch (add the equation of the entropy in the system), resistivity (only for RHD and RMHD), rotating frame around the z-axis, thermal conduction and viscosity effects; the user-defined constants, as for example, the characteristic magnitudes of the problem; and finally, other used options as smoothing, level of verbosity, enable a vector potential to describe the magnetic field, etc.

3.4. Initialization file: `pluto.ini`

The initialization file `pluto.ini` contains the run-time information necessary for the integration. It is divided in 8 blocks: the grid block, where we describe the physical domain and the resolution of the grid; the Chombo refinement block, where we define the levels of refinement, refinement ratios between each level, maximum grid size, fill ratio and other parameters used when PLUTO is compiled with the Chombo libraries; the time block, where we specify the CFL (Courant-Friedrichs-Lewy) number, the initial time step and the final time of the integration; the solver block, where we choose the Riemann solver that we want to use (two-shock approach, linearized Roe solver, Advection Upstream Splitting Method, Harten-Lax-van Leer (HLL), HLL with

contact discontinuity (HLLC), HLL with contact discontinuity and Alfvén mode (HLLD) or Lax-Friedrichs scheme); the boundary block, where we define the boundary conditions (outflow, reflective, axisymmetric, equatorial symmetric, periodic, shearingbox or user-defined); the static grid and Chombo HDF5 output blocks, where we specify the format of the outputs and the frequency depending on the time step or the computational time; and the parameters block, where we write the values for the user-defined parameters included in `definitions.h`.

3.5. Initial and boundary conditions set-up: `init.c`

We define the initial and boundary conditions of our problem in the file `init.c`. In this file there is a set of subroutines to implement this information. The user can also include additional subroutines if it is needed, but the main ones are the following: `Init()`, where we set the initial conditions as a function of the spatial coordinates; `UserDefBoundary()`, where we set the boundary conditions; `Analysis()`, where we create specific outputs on additional parameters that we are interested in; `BodyForceVector()` and `BodyForcePotential()`, where we define the vector components of an acceleration vector and/or a scalar potential; and `BackgroundField()`, where we set a background magnetic field. The initial and boundary conditions are fixed in a labeled array of primitive variables.

3.6. Higher-order finite difference schemes

As I already commented in section 3.3, we can use some higher-order algorithms already implemented in PLUTO. The available schemes are the LimO3, which is a third order limiter function proposed in Čada & Torrilhon 2009; WENO3 and WENO5, which are third and fifth-order schemes improved and based on the essentially non-oscillatory (ENO) schemes described in Shu & Osher 1989; and the MP5 scheme based on the monotonicity preserving scheme proposed in Suresh & Huynh 1997. All these schemes are not available for all the

physics modules and all their possible configurations. For instance, they are not available for RHD and RMHD modules, the time stepping has to be RK3 always and only cartesian coordinates are supported.

3.7. Output and visualization tools

PLUTO provides many different formats for the output files. The available formats are:

1. ***.dbl**: double-precision binary data (serial/parallel);
2. ***.flt**: single-precision binary data (serial/parallel);
3. ***.dbl.h5**: double-precision HDF5 data (serial/parallel);
4. ***.flt.h5**: single-precision HDF5 data (serial/parallel);
5. ***.vtk**: VTK file format (serial/parallel);
6. ***.tab**: tabulated multi-column ascii format (serial only);
7. ***.ppm**: portable pixmap color images of 2D data slices (serial/parallel);
8. ***.png**: portable network graphics color images of 2D data slices (serial/parallel).

The output files contain information about the density, pressure, velocity, magnetic field profiles, tracers and other analysis parameters introduced by the user.

Regarding the visualization tools, the data can be represented by using different programs as Gnuplot, IDL, Mathematica, VisIt or Paraview. PLUTO includes also its own Python modules to visualize the data with a graphical interface called pyPLUTO. In this training we used VisIt. VisIt works easily with *.vtk, *.h5 and *.hdf5 files and makes possible to perform many additional mathematical operations and analysis using the data without having to create more code or use additional programs.

3.8. Adaptive mesh refinement

PLUTO relies the AMR functionality in the Chombo libraries. Chombo provides a distributed infrastructure for parallel calculations over block-structured, adaptively refined grids. This implementation is compatible with all the physics modules and all coordinate systems. Regarding the input and output files, they are provided by the Hierarchical Data Format (HDF5), designed to store and organize large amounts of numerical data. The implementation method is explained in Mignone et al. 2012.

4. Simulations performed with PLUTO

4.1. 1D shock model for SNRs

A supernova remnant (SNR) is a nebula produced after the supernova explosion of an old massive star (see an example in the left picture of figure 1). Most of them are detected in radio wavelengths, but if they are young enough (up to 20 kyr, with some exceptions) most of the energy is emitted in X-rays as thermal radiation. Due to pion decay, they can be observed also at γ -rays and very high energies (VHE). We can model these nebulae as an spherical outflow with a constant density core and a decreasing density power law profile (such that $\rho(r) \sim \rho_0 r^{-n}$, being n the SNR density index) envelope that expand with an uniform velocity profile which depends only on the radius (1D approach). The pressure inside the SNR is neglected in comparison with the pressure of the expanding shocks. The most important shocks produced during the explosion are the forward shock and the reverse shock. The forward shock is expanding outwards from the center of the SNR and the reverse shock is just behind and moves inwards.

After the installation and study of PLUTO, my first task was to configure a 1D shock model for a SNR inspired in the isotropic non-radiative model of Truelove and McKee 1999. We need to solve the 1D Euler equations (HD module) for a non-radiative



Figure 1: *Left picture: Supernova Remnant SN 1006. Picture taken from the Astronomical Picture Of the Day July 4th 2008. Right picture: Crab Pulsar Wind Nebula. Picture taken from the NASA Image Gallery.*

and non-magnetic fluid in spherical symmetry and for an ideal monoatomic gas ($\gamma = 5/3$).

We implement the initial and boundary conditions of the problem in the file `init.c` and we also write the analysis tools and subroutines that we need. The initial conditions are described in equations 19, 20 and 21 in Truelove & McKee 1999, with an initial time of $t_0 = 0.01$, given in adimensional units. Another important contribution to this problem was to implement an analysis tool to identify the position of the shocks at each moment. In figure 2, I show two examples of the tests that we did in order to see how the evolution of the shocks is affected by physical and numerical effects, in particular, the interstellar medium density and the CFL number. In the case of the CFL number, we see that for $CFL = 0.5$ we start to have numerical instabilities in the last moments of the simulation. Regarding the other values of the CFL number, the results that we get are very similar as we expected.

The evolution of the shocks for a SNR is important to understand not only the evolution of the morphology of these objects, but also to study with detail the evolution of other objects that evolution inside the SNR, for instance, the pulsar wind nebulae (PWNe) that we describe in the next section.

4.2. 3D RMHD model for an isolated PWN

Pulsar Wind Nebulae (PWNe) are winds of particles injected to the interstellar medium by neutron stars or pulsars. These charged particles are accelerated by the magnetic field generated inside the nebula (pulsar wind) and radiate synchrotron emission from radio to X-rays and inverse Compton (IC) emission from hard X-rays to γ -rays and very high energies (VHE). There are many works in the literature with 1D radiation models capable to describe the main features of the spectra of these objects. Some other works (in a quite smaller quantity) are focused to mimic the complex morphology of the PWNe during its free expansion phase (no interaction with the SNR). The morphology of these nebulae consists in a torus with the powering neutron star in its center and two jets expanding from the poles perpendicularly with respect the torus (see the Crab Nebula, a canonical example in the right figure in figure 1). This morphology is formed due to the formation of a toroidal magnetic field inside the nebula. In order to study this with some detail, we need to solve the equations of magnetohydrodynamics in its relativistic version (RMHD). In the last years, we find some authors that made different approaches in the radiative equations to combine them effectively with the RMHD solution. This is a key factor if we want to understand in detail the formation of spectral features, for example, the wisps in

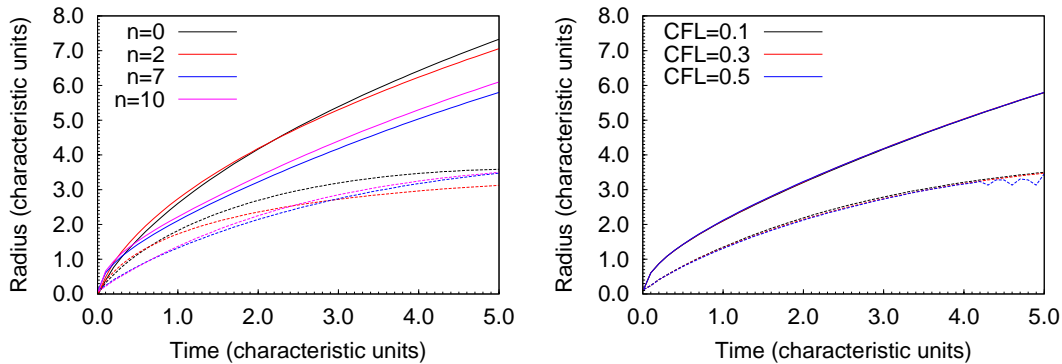


Figure 2: Evolution of the forward (solid lines) and reverse (dashed lines) shocks for a non-radiative isotropic supernova remnant. Left plot: Evolution of the shocks for different values of the interstellar medium density index n . This index is the exponent of a power law that we use to model the surrounding medium density such that $\rho(r) \sim \rho_0 r^{-n}$. Right plot: Evolution of the shocks for different values of the CFL number. The results in this case are very similar and we see only some numerical instabilities at $CFL=0.5$.

the torus and knots in the jets that evolve in small periods of time. But all these models only apply for a PWN expanding freely inside the SNR. What happens when the reverse shock of the SNR reaches the PWN? We know that there is a compression of the PWN and this enhances the magnetic field (and consequently there is an enhancement of the synchrotron radiation). After the compression, the PWN re-expands in equilibrium with the shocked medium of the SNR (Sedov phase). How is the morphology affected by this effect? What about the jets? And the wisps and knots, are they still there?

As a very first step in order to study this, I started to write the definitions.h, pluto.ini and init.c files to reproduce some results given in Porth, Komissarov & Keppens 2014 and Bühler & Giomi 2016. In these papers, they make a RMHD model for a radiative PWN. This project is still ongoing and there are no preliminar results to show in a report, but we could already obtain simulations of a PWN with low resolution. The radiative part of the analysis code is still under development, but it is making progress in a very promising way.

5. Codes and documentation

To summarize my work, I could write and comment two codes to simulate HD shocks expanding in low density environments and a RMHD simulation of a PWN without calculating the particle content and the radiation produced. These codes have a total of 1200 lines. Plotting scripts to generate figures and configuration files of the simulations are excluded. Regarding the documentation, I generated new manuals for the installation of PLUTO in Linux and MacOS systems and I reported all the tests that we did using the codes, making a total of ~ 60 pages of documentation useful for future projects and forthcoming users.

6. Ongoing work

Our aim is now to implement an analysis subroutine to calculate the spectrum of particles injected inside the PWN taking into account the main mechanisms of energy loss and to get from this the radiation emitted by these particles and perform a high resolution simulation. In order to do this, we will follow some of the prescriptions given by Bühler & Giomi 2016, Del Zanna, Amato & Bucciantini 2004, Del Zanna et al. 2006, Olmi et al. 2014, 2015

and Porth, Komissarov & Keppens 2014. We want also to search for an efficient and realistic method to compute the broadband spectrum of the PWN (from radio to gamma-rays) by introducing the solution of the energy loss equation for particles inside PLUTO. This would give as a complete picture for a PWN which has been never done before.

References

- [Bühler & Giomi, 2016] Bühler R. & Giomi M., 2016, MNRAS, 462, 2762
- [Čada & Torrilhon, 2009] Čada M. & Torrilhon M., 2009, JCP, 87, 171
- [Del Zanna, Amato & Bucciantini, 2004] Del Zanna L., Amato E. & Bucciantini N., 2004, A&A, 421, 1063
- [Del Zanna et al., 2006] Del Zanna L., Volpi D., Amato E. & Bucciantini N., 2006, A&A, 453, 621
- [Mignone et al., 2007] Mignone A. et al., 2007, ApJS, 170, 280
- [Mignone et al., 2012] Mignone A. et al., 2012, ApJS, 198, 7
- [Olimi et al., 2014] Olmi B., Del Zanna L., Amato E., Bandiera R. & Bucciantini N., 2014, MNRAS, 438, 1518
- [Olimi et al., 2015] Olmi B., Del Zanna L., Amato E. & Bucciantini N., 2015, MNRAS, 449, 3149
- [Porth, Komissarov & Keppens, 2014] Porth O., Komissarov S. S. & Keppens R., 2014, MNRAS, 438, 278
- [Shu & Osher, 1989] Shu C. W. & Osher S., 1989, JCP, 83, 32
- [Suresh & Huynh, 1997] Suresh A. & Huynh H. T., 1997, JCP, 136, 83
- [Truelove & McKee, 1999] Truelove J. K. & McKee C. F., 1999, ApJS, 120, 299