IL NUOVO CIMENTO Vol. 32 C, N. 2 Marzo-Aprile 2009
DOI 10.1393/ncc/i2009-10382-y

Colloquia: CSFI 2008

# The PLUTO code for astrophysical gas dynamics

A. Mignone

*Dipartimento di Fisica Generale "Amedeo Avogadro", Università di Torino*
*via Pietro Giuria 1, 10125 Torino, Italy*

(ricevuto il 15 Maggio 2009; pubblicato online il 3 Settembre 2009)

**Summary.** — We present the PLUTO code for the solution of high-Mach number flows in 1, 2 and 3 spatial dimensions and different systems of coordinates. The code is suitable for astrophysical gas dynamics and embeds different hydrodynamic modules and algorithms to properly describe Newtonian, relativistic, MHD, or relativistic MHD fluids. The modular structure exploits a general framework for integrating a system of conservation laws, built on modern Godunov-type shock-capturing schemes. The code is freely distributed under the GNU public license and it is available for download to the astrophysical community at the URL `http://plutocode.to.astro.it`.

PACS 47.11.-j – Computational methods in fluid dynamics.
PACS 47.40.-x – Compressible flows; shock waves.
PACS 47.65.-d – Magnetohydrodynamics and electrohydrodynamics.
PACS 95.30.Qd – Magnetohydrodynamics and plasmas.

## 1. – Description

Modeling of astrophysical flows has prompted the search for efficient and accurate numerical methods. There is now a strong consensus that the so-called high-resolution shock-capturing (HRSC) schemes provide the necessary tools in developing stable and robust fluid dynamical codes. These schemes are based on a three-step sequence consisting of a piecewise polynomial reconstruction inside each cell, a Riemann solver between discontinous states at zone interfaces and a final update where averaged conserved variables are evolved to the next time level.

Since this sequence of steps is quite general for several systems of conservation laws, we have built a multi-physics, multi-algorithm, high-resolution code, PLUTO [1]. The code is particularly suitable for the simulation of time-dependent highly supersonic flows in the presence of strong discontinuities. The modularity allows to solve different equations, *i.e.*, classical, relativistic unmagnetized, and magnetized flows. The advantage offered by a multiphysics, multisolver code is to supply the user with the most appropriate algorithms and, at the same time, provide inter-scheme comparison for a better verification of the simulation results. PLUTO is entirely written in the C programming language and can run on either single processor or parallel machines, the latter functionality being implemented through the message passing interface (MPI) library.

Finally, user-friendliness has been one of the main goals during the development of PLUTO. A simple user interface based on the Python scripting language is available to setup a physical problem in a quick and self-explanatory way. The interface is conceived to minimize the coding efforts left to the user, allowing to specify all problem-dependent attributes and algorithms, such as number of dimensions, geometry, physics module, reconstruction method, time stepping integration and so forth.

The code together with its documentation is freely distributed and it is available at the Web site `http://plutocode.to.astro.it`.

## 2. – Method of solution

PLUTO is designed to solve an arbitrary system of conservation laws,

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{T} = \mathbf{S}\,, \tag{1}$$

where $\mathbf{U}$ is a state vector of conservative quantities, $\mathbf{T}$ is a rank-2 flux tensor and $\mathbf{S}$ defines the source terms. The explicit form of $\mathbf{U}$, $\mathbf{T}$ and $\mathbf{S}$ depends on the underlying physics being adopted. At the time of this writing, PLUTO supports four independent physics modules, appropriate to describe Newtonian hydro- and magnetohydro-dynamics (HD and MHD) together with their respective relativistic extensions (RHD and RMHD).

For any particular physics module, the conservation law (1) is discretized on a logically rectangular mesh defined by grid coordinates $x^1_i$, $x^2_j$ and $x^3_k$, where $i$, $j$ and $k$ span the entire domain. Grids can be static or adaptive, the latter functionality being provided by the Chombo library available at `http://seesar.lbl.gov/ANAG/chombo/`.

In one dimension, the building block of a conservative shock-capturing scheme reads

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \sum_{d=d'}^{d=d''} \frac{\mathbf{F}^d_+ - \mathbf{F}^d_-}{\Delta x^d}\,, \tag{2}$$

where $\mathbf{U}^n$ is the known solution in a given cell $(i, j, k)$ at $t = t^n$, $d$ labels the directional sweep and $\mathbf{F}^d_\pm$ are the fluxes computed at the zone faces orthogonal to the $x^d$ axis. In a dimensionally split method, one simply has $d' = d''$ and the solution consists of sequentially solving one-dimensional problems using eq. (2). On the contrary, in a fully unsplit scheme $d' = 1$ and $d'' = 3$ (in three dimensions) and flux contributions are simultaneously taken from all directions.

Flux computation follows the solution of one-dimensional Riemann problems between discontinuous left and right states at zone interfaces. A number of Riemann solvers is available, such as the Harten-Lax-van Leer (HLL, [2]), HLLC (see [3] and references therein), the Roe scheme [4, 5] and nonlinear two-shock solvers [6, 7].

Left and right states feeding the Riemann solver are provided by suitable piecewise polynomial reconstruction inside each cell. Spurious oscillations are avoided by enforcing monotonicity constraints in proximity of steep gradients or discontinuities, see [6, 1] and references therein. Besides linear (2nd) order reconstruction, other available options include the piecewise parabolic reconstructions [7] and the 5th-order finite-difference WENO scheme of [8].

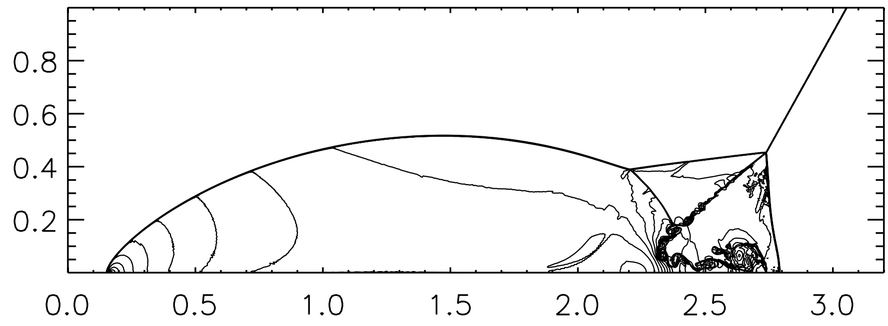


Fig. 1. – Double Mach reflection of a strong shock. Results are shown at $t = 0.2$ on a grid with spacing $1/\Delta x = 960$.

Time stepping can be done using i) method of lines or ii) single step, edge-extrapolated schemes. In the method of lines, spatial discretization is considered separately from the temporal evolution which is left continuous in time. In this framework eq. (1) is discretized as a regular ODE and either the 2nd- or 3rd-order Total Variation Diminishing (TVD) Runge-Kutta schemes of [9]. On the other hand, single-step methods achieve second-order temporal accuracy by computing the fluxes in eq. (2) at $t^n + \Delta t/2$ and the input states to the Riemann solver are estimated using Taylor expansion. This yields the well-known MUSCL-Hancock scheme [6]. A more sophisticated approach is employed, for instance, in the characteristic tracing scheme originally described in the PPM scheme of [10]. The dimensionally unsplit version of this strategy leads to the corner transport upwind (CTU) method of [11]. In this case, an extra correction term accounting for transverse contribution is required. This is the preferred time marching scheme when adaptive mesh refinement is employed, since it requires only one boundary call between adjacent blocks.

## 3. – Code benchmarks

The PLUTO code has been successfully tested on the most severe benchmarks and a number of test problems are given along with the code distribution, see [7, 1] for a comprehensive review. Most tests were specifically designed to deal with highly supersonic flows in the presence of strong discontinuities.

In fig. 1 we show, for example, the density distribution for the double Mach reflection problem [12] at $t = 0.2$ computed with the fifth-order WENO scheme and the third-order Runge Kutta scheme. After the reflection, a complicated flow structure develops with two curved reflected shocks propagating at directions almost orthogonal to each other and a tangential discontinuity separating them. At the wall, a pressure gradient sets up a denser fluid jet propagating along the wall. Kelvin-Helmholtz instability patterns may be identified with the "rolls" developing at the slip line.

A second illustrative example consists in the MHD rotor problem, a rapidly spinning cylinder with higher density embedded in a static background medium with uniform pressure, threaded by a constant horizontal magnetic field, [1]. The left panel in fig. 2 shows density and magnetic pressure contours computed with linear interpolation and Powell's eight wave scheme [13] to control the $\nabla \cdot \mathbf{B} = 0$ condition. Results are shown

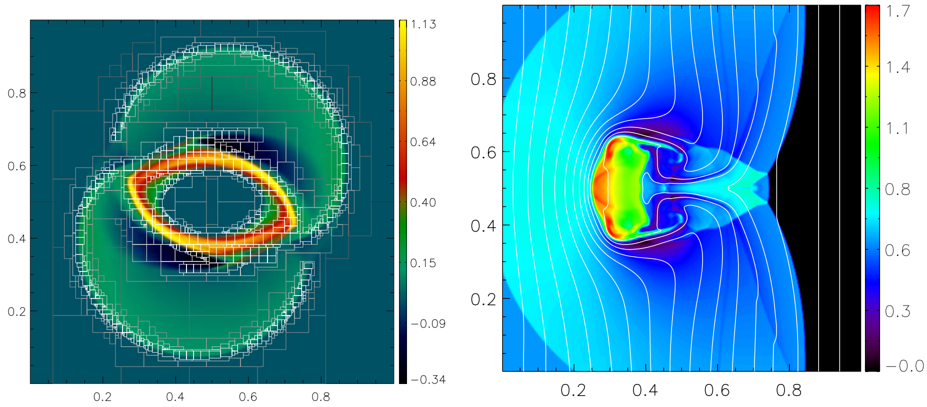


Fig. 2. – Left: density logarithm for the rotor problem at $t = 0.15$. The refined grid patches are enclosed by boxes of different gray shades. Right: interaction of a strong magentized shock with a cloud at $t = 0.06$. Magnetic field lines are over-plotted. The grid has resolution $1/\Delta x = 400$.

at $t = 0.15$ on a $32 \times 32$ Cartesian grid with 5 levels of refinement. As the disk rotates, strong torsional Alfvèn waves form and propagate outward carrying angular momentum from the disk to the ambient.

Last, we show the interaction of a strong magnetized shock with a higher density cloud, right panel in fig. 2. This problem is thoroughly discussed in [14]. In this case the constrained method has been used to evolve the magnetic field and the HLLD scheme of [15] to compute the solution to the Riemann problem. Piecewise parabolic interpolation is used. After the impact, a fast bow shock propagates into the shocked material and a reverse shock is transmitted back into the cloud. By $t = 0.06$ the cloud is entirely wrapped by the incident shock and it becomes a mushroom-shaped shell.

## REFERENCES


[1] Mignone A., Massaglia S., Bodo G., *et al.*, *Astrophys. J. Suppl.*, **170** (2007) 228.
[2] Harten A., Lax P. D. and van Leer B., *SIAM Rev.*, **25** (1983) 61.
[3] Mignone A. and Bodo G., *Mon. Not. R. Astron. Soc.*, **368** (2006) 1040.
[4] Roe P. L., *Annu. Rev. Fluid Mech.*, **18** (1986) 337.
[5] Cargo P. and Gallice G., *J. Comput. Phys.*, **136** (1997) 446.
[6] Toro E., *Riemann Solvers and Numerical Methods for Fluid Dynamics* (Springer) 1999.
[7] Mignone A., Plewa T. and Bodo G., *Astrophys. J. Suppl.*, **160** (2005) 199.
[8] Jiang G.-S. and Shu C.-W., *J. Comput. Phys.*, **126** (1996) 202.
[9] Gottlieb S. and Shu C.-W., *Math. Comput.*, **67** (1998) 73.
[10] Colella P. and Woodward P. R., *J. Comput. Phys.*, **54** (1984) 174.
[11] Saltzman J., *J. Comput. Phys.*, **115** (1994) 153.
[12] Woodward P. R. and Colella P., *J. Comput. Phys*, **54** (1984) 115.
[13] Powell K. G., Roe P. L., Linde T. J., Gombosi T. I. and de Zeeuw D. L., *J. Comput. Phys.*, **153** (1999) 284.
[14] Dai W. and Woodward P. R., *Astrophys. J.*, **436** (1994) 776.
[15] Miyoshi T. and Kusano K., *J. Comput. Phys.*, **208** (2005) 315.