Mem. S.A.It. Suppl. Vol. 13, 67

Memorie della
Supplementi

The PLUTO code for astrophysical gasdynamics

A. Mignone[1,2]

[1] Dipartimento di Fisica Generale "Amedeo Avogadro", Università degli Studi di Torino, via Pietro Giuria 1, 10125 Torino, Italy
[2] INAF-Osservatorio Astronomico di Torino, Strada Osservatorio 20, 10025 Pino Torinese, Italy, e-mail: mignone@oato.inaf.it

Abstract. Present numerical codes appeal to a consolidated theory based on finite difference and Godunov-type schemes. In this context we have developed a versatile numerical code, PLUTO, suitable for the solution of high-mach number flow in 1, 2 and 3 spatial dimensions and different systems of coordinates. Different hydrodynamic modules and algorithms may be independently selected 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.

Key words. Methods: numerical - Hydrodynamics - Plasmas - Shock waves

1. Introduction

Astrophysical gases evolve under conditions which are often far from equilibrium and their dynamics is described by complex non-linear interactions of multiple waves which can hardly be solved by analytical methods. For this reason, the modeling of such phenomena has prompted the search for efficient and accurate numerical formulations. In this perspective, the recent advances in numerical modeling of astrophysical flows together with the availability of increasingly more powerful computational resources have made theoretical investigations amenable to numerical simulations.

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.

Finite volume HRSC methods 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, see the books by Toro (1997), LeVeque (1998) and references therein.

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 (Mignone et al. 2007). The code is particularly suitable for the simulation of time-dependent highly supersonic flows in presence of strong discontinuities. The modularity allows to solve different equations, i.e., classical, relativistic unmagnetized, and magnetized flows. The advantage of-

Send offprint requests to: A. Mignone

fered by a multiphysics, multisolver code is to supply the user with the most appropriate algorithms and, at the same time, provide interscheme 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.

2. Code structure

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

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

where $\boldsymbol{U}$ is a state vector of conservative quantities, T is a rank-2 flux tensor and $\boldsymbol{S}$ defines the source terms.

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).

The HD module integrates the Euler equations of classical fluid dynamics, expressing conservation of mass, momentum and energy:

$$\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho\boldsymbol{v} \\ E \end{pmatrix}, \quad \mathsf{T} = \begin{pmatrix} \rho\boldsymbol{v} \\ \rho\boldsymbol{v}\boldsymbol{v} + p\mathsf{I} \\ (E+p)\boldsymbol{v} \end{pmatrix}^{\mathrm{T}}, \tag{2}$$

where ρ, $\boldsymbol{v}$, E and p denote, respectively the density, velocity, energy and pressure of the fluid. The total energy density E and gas pressure p are related through the ideal gas closure:

$$E = \frac{p}{\Gamma - 1} + \frac{|\boldsymbol{m}|^2}{2\rho}. \tag{3}$$

Charged fluids may be described by extending the Euler equations to single fluid ideal magnetohydrodynamics (MHD):

$$\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho\boldsymbol{v} \\ \boldsymbol{B} \\ E \end{pmatrix}, \; \mathsf{T} = \begin{pmatrix} \rho\boldsymbol{v} \\ \rho\boldsymbol{v}\boldsymbol{v} - \boldsymbol{B}\boldsymbol{B} + p_{\mathrm{t}}\mathsf{I} \\ \boldsymbol{v}\boldsymbol{B} - \boldsymbol{B}\boldsymbol{v} \\ (E + p_{\mathrm{t}})\boldsymbol{v} - (\boldsymbol{v}\cdot\boldsymbol{B})\boldsymbol{B} \end{pmatrix}^{\mathrm{T}}, \tag{4}$$

where $\boldsymbol{B}$ is the magnetic induction, $p_t = p + \boldsymbol{B}^2/2$ is the total pressure and $E = E_{\mathrm{HD}} + \boldsymbol{B}^2/2$ is the total energy density. The additional constraint $\nabla \cdot \boldsymbol{B} = 0$ complements the magnetic field evolution. This may be enforced by using 1) the eight wave formulation (Powell 1994) or 2) the constrained transport (CT henceforth) of Balsara & Spicer (1999), and Londrillo & Del Zanna (2004). See Tóth (2000) for a comprehensive review.

The (special) relativistic extensions of Eq. (2) and (4) are expressed by particle number and energy-momentum conservation (Landau & Lifshitz 1959). The RHD module, for instance, becomes

$$\boldsymbol{U} = \begin{pmatrix} \rho\gamma \\ \rho h\gamma^2\boldsymbol{v} \\ \rho h\gamma^2 - p \end{pmatrix}, \quad \mathsf{T} = \begin{pmatrix} \rho\gamma\boldsymbol{v} \\ \rho h\gamma^2\boldsymbol{v}\boldsymbol{v} + p\mathsf{I} \\ \rho h\gamma^2\boldsymbol{v} \end{pmatrix}^{\mathrm{T}}, \tag{5}$$

where $\gamma = (1 - |\boldsymbol{v}|^2)^{-\frac{1}{2}}$ is the Lorentz factor and h is the specific enthalpy. Despite the apparent similarity to their nonrelativistic limit, Eq. (2) are intrinsically more complex due to the intrinsic coupling between thermal and kinetic terms.

In presence of magnetic fields, the stress energy tensor must include electromagnetic contributions and Maxwell's equations are used to evolve the fields. Under the assumption of infinite conductivity, the electric field can be written as $\boldsymbol{\Omega} = -\boldsymbol{v} \times \boldsymbol{B}$ and the conserved variables become

$$\boldsymbol{U} = \begin{pmatrix} \rho\gamma \\ w_{\mathrm{t}}\gamma^2\boldsymbol{v} - b^0\boldsymbol{b} \\ \boldsymbol{B} \\ w_{\mathrm{t}}\gamma^2 - b^0b^0 - p \end{pmatrix}, \tag{6}$$

where $w_{\mathrm{t}} = \rho h + b^2$ is the total enthalpy while $b^2 \equiv b^\mu b_\mu = |\boldsymbol{B}|^2/\gamma^2 + (\boldsymbol{v}\cdot\boldsymbol{B})^2$ is the norm of the magnetic four-induction $b^\mu = (b^0, \boldsymbol{b})$ with components given by

$$b^0 = \gamma\boldsymbol{v}\cdot\boldsymbol{B}, \quad \boldsymbol{b} = \frac{\boldsymbol{B}}{\gamma} + b^0\boldsymbol{v}. \tag{7}$$

The flux tensor becomes

$$\mathsf{T} = \begin{pmatrix} \rho\gamma\boldsymbol{v} \\ w_t\gamma^2\boldsymbol{v}\boldsymbol{v} - \boldsymbol{b}\boldsymbol{b} + \mathsf{I}p_t \\ \boldsymbol{v}\boldsymbol{B} - \boldsymbol{B}\boldsymbol{v} \\ w_t\gamma^2\boldsymbol{v} - b^0\boldsymbol{b} \end{pmatrix}^{\mathrm{T}} , \tag{8}$$

with $p_t = p + b^2/2$ being the total (thermal + magnetic) pressure.

Note that the transformation between conservative and primitive variables such as $(\rho, \boldsymbol{v}, p, \boldsymbol{B})$ is considerably more involved in the RHD and RMHD modules and requires the solution of nonlinear equations. See, for example Noble et al. (2006); Mignone & Bodo (2006); Mignone & Mc Kinney (2007).

2.1. Method of solution

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

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \Delta t\boldsymbol{\mathcal{L}} , \tag{9}$$

where $\boldsymbol{U}^n$ is the known solution in a given cell (i, j, k) at $t = t^n$ and $\boldsymbol{\mathcal{L}}$ is the flux difference operator

$$\boldsymbol{\mathcal{L}} = \sum_{d=d'}^{d=d''} \frac{\boldsymbol{F}^d_+ - \boldsymbol{F}^d_-}{\Delta x^d} . \tag{10}$$

Here d labels the directional sweep and $\boldsymbol{F}^d_\pm$ 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. (9). 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.

The time increment Δt is limited by the usual Courant-Friedrichs-Lewy (CFL) condition:

$$\Delta t = C_a \frac{\min(\Delta x_d)}{\max\left(|\lambda^d_{\max}|\right)} , \quad C_a < 1 \tag{11}$$

with $\lambda^d_{\max}$ being the largest signal velocity in the d direction.

Flux computation follows the solution of one-dimensional Riemann problems between discontinuous left and right states at zone interfaces. Specializing to one dimension and omitting the d superscript, one has

$$\boldsymbol{F}_+ = \boldsymbol{F}\left(\boldsymbol{U}_{i+\frac{1}{2}}\right) , \tag{12}$$

where $\boldsymbol{U}_{i+\frac{1}{2}}$ is the exact or approximate solution to the original conservation law (1) with initial condition given by

$$\boldsymbol{U}(x, t = 0) = \begin{cases} \boldsymbol{U}_L & \text{for} \quad x < x_{i+\frac{1}{2}} \\ \boldsymbol{U}_R & \text{for} \quad x > x_{i+\frac{1}{2}} \end{cases} \tag{13}$$

Left and right states $\boldsymbol{U}_L$ and $\boldsymbol{U}_R$ are provided by suitable piecewise monotonic interpolation.

The procedure we just sketched out turns out to be general enough for most systems of conservation laws, including those we have previously mentioned. The explicit form of $\boldsymbol{U}$, T and $\boldsymbol{S}$ depends on the underlying physics being adopted. In its minimal form, a physics module collects the set of algorithms required to compute the terms involved in the discretization of the right hand side of Eq. (1). This set should provide one or more Riemann solver(s), mapper routines for the conversion between primitive and conservative variables, a flux routine giving the components of $\mathsf{T}(\boldsymbol{U})$ in each direction, a source term function (if any) and a routine to compute the maximum and minimum characteristic speeds of the Jacobian matrix. Of course, additional features may be easily added by exploiting the independent modularity.

From the user's perspective, a particular configuration can be defined through a friendly interface entirely written in the Python scripting language. The interface allows the user to specify all problem-dependent attributes and algorithms, such as number of dimensions, geometry, physics module, reconstruction method, time stepping integration and so forth.

2.2. Available algorithms

2.2.1. Reconstruction

Piecewise polynomial reconstruction $\mathcal{P}(\boldsymbol{x})$ is used to reconstruct $\boldsymbol{U}$ inside each cell. Spurious oscillations are avoided by enforcing monotonicity constraints in proximity of steep gradients or discontinuities, see Toro (1997), LeVeque (1998). For a second order accurate scheme, for instance, one has

$$\boldsymbol{U}_{\pm} = \boldsymbol{U} \pm \frac{\Delta\boldsymbol{U}}{2}\,, \tag{14}$$

where $\Delta\boldsymbol{U}$ are computed following a limiting procedure applied to primitive or characteristic variables.

Other available options include the third-order convex ENO (Del Zanna & Bucciantini 2002), the piecewise parabolic reconstructions (as in Mignone et al. 2005) and the 5th order finite difference WENO scheme of Jiang & Shu (1996).

2.3. Riemann solver

The exact solution to the Riemann problem (13) involves the decay of a set of non-linear waves and can be a rather cumbersome task to achieve. With the exception of few simple cases, existing Riemann solvers routinely involved in upwind schemes are based on different levels of approximation.

Nonlinear Riemann solvers solve the Rankine-Hugoniot jump conditions across each wave and assuming the outermost waves can be treated as shocks ("two-shock" Riemann solvers). In this case one has

$$\boldsymbol{F}_{+} = \boldsymbol{f}(\boldsymbol{U}^{*})\,. \tag{15}$$

where $\boldsymbol{U}^{*}$ is the solution, for $t^{n} < t < t^{n+1}$ on the $x = x_{i+\frac{1}{2}}$ axis and $\boldsymbol{f} = \hat{\boldsymbol{e}}^{d}\cdot\mathsf{T}(\boldsymbol{U})$ is the projection of the tensor flux on the $\hat{\boldsymbol{e}}^{d}$ unit vector. However, this approach is computationally expensive since it generally involves the solution of highly nonlinear equations. Two-shock solvers are available in PLUTO for the HD and the RHD modules, as described in Colella & Woodard (1984) and Mignone et al. (2005), respectively.

Linearized Riemann solvers, such as the Roe solver, are based on a local linearization around some average state. A common way (**?**) to write the flux function is

$$\boldsymbol{F}_{+} = \frac{\boldsymbol{f}_{\rm L} + \boldsymbol{f}_{\rm R}}{2} - \sum_{k}\frac{|\lambda_k|}{2}\left(\boldsymbol{L}_k\cdot\Delta\boldsymbol{U}\right)\boldsymbol{R}_k\,, \tag{16}$$

where $\Delta\boldsymbol{U} = \boldsymbol{U}_R - \boldsymbol{U}_L$, $\boldsymbol{L}$ and $\boldsymbol{R}$ are the left and right eigenvectors of the Jacobian $\partial\boldsymbol{f}(\boldsymbol{U})/\partial\boldsymbol{U}$. Roe type Riemann solvers have been designed for the HD and MHD module, see Toro (1997) and Cargo & Gallice (1997).

By taking λ_k to be the largest local signal velocity independently of k, the diffusion term is maximized and one obtains the Lax-Friedrichs Rusanov flux (Rusanov 1961):

$$\boldsymbol{F}_{+} = \frac{\boldsymbol{f}_{\rm L} + \boldsymbol{f}_{\rm R}}{2} - \frac{|\lambda_{\max}|}{2}\Delta\boldsymbol{U}\,, \tag{17}$$

and $|\lambda_{\max}|$ is the largest local signal velocity. The Lax-Friedrichs Rusanov flux is robust but also the most diffusive solver and is available for all modules.

The Harten-Lax van Leer (HLL) solver computes the solution to the Riemann problem by estimating the smallest (λ_L) and largest (λ_R) signal velocities in the solution. From the consistency condition on the integral average of the Riemann fan, one can find:

$$\boldsymbol{F}_{+} = \begin{cases} \boldsymbol{f}_{\rm L} & \text{if } \lambda_L > 0\,, \\ \boldsymbol{f}_{\rm R} & \text{if } \lambda_R < 0\,, \\ \frac{\lambda_L\lambda_R(\boldsymbol{U}_R-\boldsymbol{U}_L)+\lambda_R\boldsymbol{f}_L-\lambda_L\boldsymbol{f}_R}{\lambda_R-\lambda_L} & \text{otherwise.} \end{cases} \tag{18}$$

The HLL approach can be improved by allowing the existence of intermediate waves inside the Riemann fan. The HLLC solver restores the contact wave λ_c and takes the form

$$\boldsymbol{F}_{+} = \begin{cases} \boldsymbol{f}_{\rm L} & \text{if } \lambda_L > 0\,, \\ \boldsymbol{f}_{\rm L} + \lambda_L\left(\boldsymbol{U}^{*}_{L} - \boldsymbol{U}_L\right) & \text{if } \lambda_L < 0 < \lambda_c\,, \\ \boldsymbol{f}_{\rm R} + \lambda_R\left(\boldsymbol{U}^{*}_{R} - \boldsymbol{U}_R\right) & \text{if } \lambda_c < 0 < \lambda_R\,, \\ \boldsymbol{f}_{\rm R} & \text{if } \lambda_R < 0\,, \end{cases} \tag{19}$$

where $\boldsymbol{U}^{*}_{L}$ and $\boldsymbol{U}^{*}_{R}$ are intermediate states satisfying the jump conditions across the selected waves. Similarly, the HLLD Riemann solver further extends this recipe to the case where rotational discontinuities are included in the solution (Miyoshi & Kusano 2005).

Both HLL and HLLC have been coded, see Toro (1997) for the HD and MHD equations, and Mignone & Bodo (2005, 2006) for their relativistic extensions.

2.4. Temporal evolution

There are two main time stepping strategies available in the PLUTO code.

Single-step methods achieve second order temporal accuracy by computing the flux at the half time step. In this case the $\boldsymbol{\mathcal{L}}$ operator in (10) is computed at $t^n + \Delta t/2$ and the input state to the Riemann solver are estimated using Taylor expansion:

$$\boldsymbol{U}_{\pm}^{n+\frac{1}{2}} = \boldsymbol{U}_{\pm}^{n} - \frac{\Delta t}{\Delta x}\left[\boldsymbol{f}(\boldsymbol{U}_{+}^{n}) - \boldsymbol{f}(\boldsymbol{U}_{-}^{n})\right] \tag{20}$$

where $\boldsymbol{U}_{\pm}^{n} = \boldsymbol{U}^{n} \pm \Delta\boldsymbol{U}/2$. This yields the well-known MUSCL-Hancock scheme (van Leer 1974; Toro 1997). Eq. (20) may also be written in primitive variables and upwind limiting may be used to select only those characteristics which contribute to the effective left and right states. The latter approach is employed, for instance, in the original PPM advection scheme of Colella & Woodard (1984) and we will refer to as characteristic tracing. The dimensionally unsplit version of this strategy leads to the corner transport upwind (CTU) method of Colella (1990), Saltzman (1994). In this case, an extra correction term is needed in Eq. (20). In two dimensions, for example, the input states for the Riemann problem are modified to

$$\boldsymbol{U}_{\pm}^{n+\frac{1}{2}} \to \boldsymbol{U}_{\pm}^{n+\frac{1}{2}} + \frac{\Delta t}{2}\boldsymbol{\mathcal{L}}^{t,n+\frac{1}{2}}\,, \tag{21}$$

with $\boldsymbol{\mathcal{L}}^{t,n+\frac{1}{2}}$ being the right-hand side operator corresponding to the transverse direction.

The second time stepping strategy is based on the classical method of lines, where the 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. PLUTO implements the 2nd and 3rd order Total Variation Diminishing (TVD) Runge-Kutta schemes of Gottlieb & Shu (1996) consisting, respectively of 2 and 3 stages:

$$\boldsymbol{U}^{*} = \boldsymbol{U}^{n} + \Delta t\boldsymbol{\mathcal{L}}^{n}\,, \tag{22}$$

$$\boldsymbol{U}^{n+1} = \frac{1}{2}\left[\boldsymbol{U}^{n} + \boldsymbol{U}^{*} + \Delta t\boldsymbol{\mathcal{L}}^{*}\right], \tag{23}$$

and

$$\boldsymbol{U}^{*} = \boldsymbol{U}^{n} + \Delta t\boldsymbol{\mathcal{L}}^{n}\,, \tag{24}$$

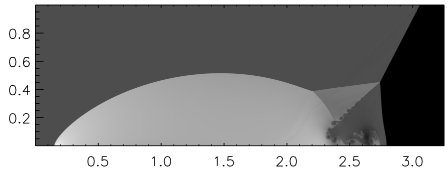

Fig. 1. Double Mach reflection of a strong shock. Results are shown at $t = 0.2$ on a grid with spacing $1/\Delta x = 960$.

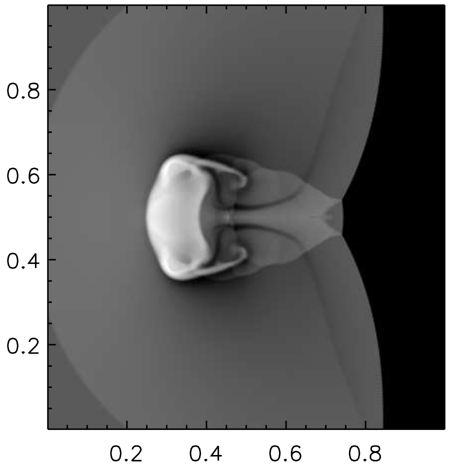

Fig. 2. Interaction of a strong magentized shock with a cloud at $t = 0.06$. The grid has resolution $1/\Delta x = 400$.

$$\boldsymbol{U}^{**} = \frac{1}{4}\left[3\boldsymbol{U}^{n} + \boldsymbol{U}^{*} + \Delta t\boldsymbol{\mathcal{L}}^{*}\right], \tag{25}$$

$$\boldsymbol{U}^{n+1} = \frac{1}{3}\left[\boldsymbol{U}^{n} + 2\boldsymbol{U}^{**} + 2\Delta t\boldsymbol{\mathcal{L}}^{**}\right]. \tag{26}$$

For this class of methods, input states for the Riemann solver are given by the output of the interpolation routine, see §2.2.1.

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 Mignone et al. (2005); Mignone et al. (2007) for a comprehensive review. Most tests were specifically designed to deal with highly supersonic flows in presence of strong discontinuities.

In Fig (1) we show, for example, the density distribution for the double Mach reflection problem

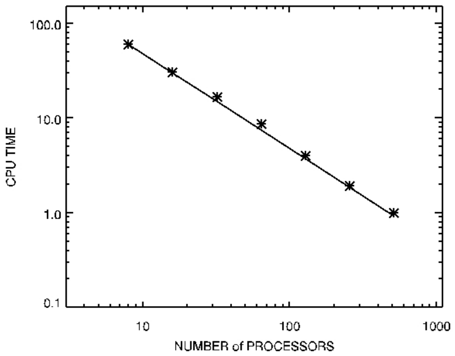

Fig. 3. Scaling of the PLUTO code on the three-dimensional version of the Shock cloud interaction with 512^3 zones. The perfect scaling slope ($\propto 1/N_p$ where N_p is the number of processors) is shown as the solid line.

(Woodward & Colella 1984) at $t = 0.2$ computed with the fifth-order WENO 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 interaction of a strong shock with a higher density cloud, shown in Fig (2). This problem is thoroughly discussed in Dai & Woodward (1994). In this case the constrained method of Balsara & Spicer (1999) has been used to control the divergence of magnetic field and the linearized solver of Cargo & Gallice (1997) to compute the solution to the Riemann problem. After the impact, a bow fast 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.

Scaling tests up to 512 processors have been conducted on the 3-D version of the Shock Cloud problem on a 512^3 grid. Results are very close to the ideal scaling and are shown in Fig. (3).

References

Balsara, D. S., & Spicer, D. S. 1999, J. Comp. Phys., 149, 270
Cargo, P., & Gallice, G. 1997, J. Comp. Phys., 136, 446
Colella, P., & Woodward, P. R. 1984, J. Comp. Phys., 54, 174
Colella, P. 1990, J. Comp. Phys., 87, 171
Dai, W., & Woodward, P. R. 1994, ApJ, 436, 776
Del Zanna, L., & Bucciantini, N. 2002, A&A, 390, 1177
Gottlieb, S., Shu, C.-W. 1996, NASA CR-201591 ICASE Report No. 96-50, 20
Jiang, G.-S., & Shu, C.-W. 1996, J. Comput. Phys., 126, 202
Landau, L. D., & Lifshitz, E. M. 1959, Fluid Mechanics, Pergamon Press
Londrillo, P., & del Zanna, L. 2004, J. Comp. Phys., 195, 17
Mignone, A., Plewa, T., & Bodo, G. 2005, ApJS, 160, 199
Mignone, A., & Bodo, G. 2005, MNRAS, 364, 126
Mignone, A., & Bodo, G. 2006, MNRAS, 368, 1040
Mignone, A., & Mc Kinney, J.C. 2007, MNRAS, 378, 1118
Mignone, A., Massaglia, S., Bodo, G., et al. 2007, ApJS, 170, 228
Miyoshi, T., & Kusano, K. 2005, J. Comp. Phys., 208, 315
Noble, S. C., Gammie, C. F., McKinney, J. C., & Del Zanna, L. 2006, ApJ, 641, 626
Powell, K. G. 1994, ICASE Report No. 94-24, Langley, VA
Roe, P. L. 1986, Ann. Rev. Fluid Mech., 18, 337
Rusanov, V. V. 1961, J. Comp. Math. Phys. USSR, 1, 267
Saltzman, J. 1994, J. Comp. Phys., 115, 153
Toro, E. F., Spruce, M., & Speares, W. 1994, Shock Waves, 4, 25
Toro, E. F. 1997, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer-Verlag, Berlin
Tóth, G. 2000, J. Comp. Phys., 161, 605
van Leer, B. 1974, J. Comp. Phys., 14, 361
LeVeque, R. J., Mihalas, D., Dorfi, E. A., & Müller, E. 1998, Computational Methods for Astrophysical Flow, ed. O. Steiner & A. Gautschy, Springer-Verlag
Woodward, P. R., & Colella, P. 1984, J. Comp. Phys., 54, 115