AstroBEAR: AMR for Astrophysical Applications - I: Methods

Poludnenko, A.1, Varnière, P.1, Cunningham, A.1, Frank, A.1, Mitran, S.2

1 University of Rochester varni, ajc4, a frank@pas.rochester.edu
2 University of North Carolina at Chapel Hill mitran@math.unc.edu

1 Introduction

Adaptive Mesh Refinement methods combined with modern shock-capturing methods hold great promise for the simulation of many astrophysical fluid dynamics environments. This is particularly true of problems involving strong shocks followed by efficient cooling via radiative losses. The high degrees of compression achieved in such circumstances require high resolutions that can tax the computational resources of many research groups. The need to accurately compute various micro-physical processes (non-equilibrium ionization, radiation transport, etc.) place further constraints on acceptable resolution for a simulation.

In this contribution we present a new AMR code for astrophysical applications called AstroBEAR. The code is designed on the BFLCIW framework (Boundary Embedded Adaptive Mesh Refinement for Conservation Laws) [M04] and it offers generalized adaptive facilities including mesh refinement in a grid-based formalism [BL98]. The code is flexible and efficient being designed specifically for multi-physics applications in which processes operating under different time and length scales can be simultaneously simulated.

Our design philosophy in the construction of AstroBEAR was to develop a code that would perform well for a targeted list of astrophysical applications. These include: the interaction of shocks and winds with heterogeneous environments; astrophysical jets; stellar wind blown nebulae; accretion disk physics including the accretion disk wind outflows and the interaction of disks with an embedded proto-planet. These applications are accessed via pre-defined modules. Various modules providing handling of physical processes (ionization dynamics, chemistry, radiative cooling, radiation pressure, central gravity, etc.) can be accessed without code re-compilation and can be switched on and off at runtime by the user.

In this paper we describe the general numerical method used including aspects of the code which are still under development (i.e. the formalism for treating flux conservation in MHD).
where the $\tilde{F}$ are the correction fluxes which will be defined in terms of the waves $\mathcal{W}_p^\nu$ and their speeds $\lambda_\nu^p$ arising from the i-th Riemann problem at the interface between cells $i$ and $i+1$. The $A^+$ and $A^-$ terms describe the fluctuation splitting and they are defined as

$$A^{\pm} \Delta q = \sum_{p=1}^{M} (\lambda^p)^{\pm} \mathcal{W}_p^\nu,$$

where $\lambda^- = \min(\lambda, 0)$ and $\lambda^+ = \max(\lambda, 0)$. Without limiters the correction fluxes take the form

$$\tilde{F}_i = \frac{1}{2} \sum_{p=1}^{M} |\lambda_p^2| \left(1 - \frac{\Delta t}{\Delta x} |\lambda_p^2| \right) \mathcal{W}_p^\nu$$

Flux, slope or wave limiters are introduced into the method described above to reduce oscillations resulting from discretization. The functions $\phi$ adjust the fluxes, carried by the waves into and out of the cell, to reduce high frequency oscillations associated with the Gibbs effect. BEARCLAW provides the user with choices of a variety of standard limiters such as MinMod and Superbee (see [L97] and references therein).

An important component of the wave propagation algorithm is the implementation of transverse waves for multi-dimensional problems. For the example of 2-dimensional problems the governing equations include fluxes in both the $x$ and $y$ directions $F_{i,j}$ and $G_{i,j}$. Since the waves resulting from the solution of a Riemann problem in the normal direction and originating at interfaces should propagate in a multi-dimensional manner one should include their effect on other neighboring cell averages. Transverse fluctuations may be defined by decomposing each fluctuation $A^\nu \Delta q$ into the “up-going” $B^+ A^\nu \Delta q_{i,j}$ and the “down-going” $B^- A^\nu \Delta q_{i,j}$ components. Here $*$ refers to the left or right propagating fluctuation. In practice the transverse fluctuations are found by determining the Jacobian matrix $B$ of the fluxes $G$ in the transverse dimension. This can be an exact Jacobian matrix as for the case of an exact Riemann solver or an approximate one as for the case of the linearized Riemann solvers. Subsequently eigenvalues $\mu^\nu$ and eigenvectors $w^\nu$ of $B$ are determined and the latter can be used to decompose $A^\nu \Delta q$ into their linear combination. Based on this we finally get for the transverse fluctuations

$$B^\pm A^\nu \Delta q_{i,j} = \Sigma_i (\mu_i^\nu)^\pm \beta_i^\nu w_i^\nu.$$ 

Thus when updating the cells $(i,j)$ and $(i+1,j)$ after solving the Riemann problem at the interface between them via an $x$ pass, the transverse fluctuations are used to modify the four neighboring $y$ fluxes below and above these two cells, namely $G_{i,j}$, $G_{i,j-1}$, $G_{i+1,j}$ and $G_{i+1,j-1}$. 

\( \rho = \frac{E_1^2 + E_2^2}{2}; \quad \rho u_1 = \frac{E_1^2 + E_2^2}{2} \); \( \rho u(2,3) = \frac{E_1^2 + E_2^2}{2} \). \quad (7)

The extent of the diffused region then depends on a quantity
\[ \epsilon = \eta_{CD} \text{MIN}(|u_l - u^*|, |u_r - u^*|), \quad (8) \]

where \( u_l \) (\( u_r \)), depending on the nature of the left (right) non-linear wave, are either the speed of the left (right) shock or the tail of the left (right) rarefaction and \( \eta_{CD} \) is a user-specified quantity between 0 and 1. Using this formalism we were able to eliminate the carbuncles while limiting numerical diffusion.

In Figure 1 we present two 1-D tests of the hydrodynamic method. The first is a Sod-type shock tube designed to assess the entropy satisfaction property of the code. The second tracks the evolution of a strong shock wave and is intended to assess the overall robustness and accuracy of the numerical scheme. Both tests were taken from [T99]. In both cases we find excellent agreement with the exact result with nominal spreading. 2-D results will be presented in Paper II.

Fig. 1. 1-D tests of the method. Resolution 500 cells, the limiter used was MinMod. Top: Sod-type shock tube; time \( t = 0.2 \); initial discontinuity at \( z = 0.3 \). Bottom: Strong blast wave; time \( t = 0.012 \); initial discontinuity at \( z = 0.5 \).

MHD Solvers: To solve the MHD equations we used the Riemann solver developed by J.Rosmanith [R02] in conjunction with the Wave Propaga-
The equation for the divergence of \( B \) now becomes an advection equation. The non-vanishing divergence term will be advected with the flow instead of remaining at the location of its creation. This procedure can work well only with outflow boundary conditions. It does have the great advantage of being inexpensive compared to other methods. However, it may also create incorrect jump condition across strong shocks because we are not solving the "right" MHD equations.

3.2 The GLM Method

This method, introduced by Dedner et al. [D01] is a generalization of the 8 waves method of Powell. It introduces a generic set of equations for dealing with the divergence of \( B \). These equations can be hyperbolic, elliptic or parabolic. We have coded a form of the advection-diffusion equation which advects the divergence of \( B \) error at the fastest speed in the system while also diffusing it away. Once again the system of equations that is solved is not the original MHD set and thus has the same problem as the 8 waves method around strong shocks.

3.3 The Projection Method

A number of tests were carried out with a projection method suitable for AMR computations. The full details of the method may be found in [MVF04]; a brief description is given here. It was shown by Tóth [T00] that the Poisson solver step, required in the projection method and proposed by Brackbill and Barnes [BB80], does not entail excessive computational penalties, especially if fast techniques are used. The problem that arises in applications to AMR is that the grids on different levels are coupled leading to the necessity of solving the Poisson equation on the entire grid hierarchy for each of the time subcycling steps. While this is indeed the case when applying the projection method to equations exhibiting a true elliptic subsystem (e.g., the incompressible Navier-Stokes equations), in the case of MHD the equation set is hyperbolic so a grid-by-grid correction procedure is feasible. The procedure is presented in the context of a 2D cell-centered finite volume method with the cell average magnetic field on a coarse grid cell \( C^2_{i,j} \) with edges of length \( 2 \Delta \) and cell area \( A^2_{i,j} \) defined by

\[
B_{i,j} = \frac{1}{A^2_{i,j}} \int_{C^2_{i,j}} B(x, y) \, dx \, dy. \tag{10}
\]

The embedded fine grid field is respectively

\[
b_{i,j} = \frac{1}{A^2_{i,j}} \int_{C^2_{i,j}} B(x, y) \, dx \, dy. \tag{11}
\]

\(^3\) Here as an example we assume the refinement ratio of 2 though it can be any positive even number.
The natural restriction operator used in AMR computations of conservation laws is simple averaging of embedded fine grid quantities

\[ \tilde{B}_{ij} = \frac{1}{4} (b_{2i-1,2j-1} + b_{2i,2j-1} + b_{2i-1,2j} + b_{2i,2j}) \]  

(20)

This can be shown to maintain \( O(h^3) \) accuracy for the updated coarse grid values at cells away from coarse-fine interfaces since a series expansion at such points gives

\[ \nabla_{2h} \cdot \tilde{B}_{ij} = \frac{\partial B_x}{\partial x} + \frac{\partial B_y}{\partial y} + \frac{h^2}{8} \left[ \frac{\partial^2 B_x}{\partial x^2} + \frac{\partial^2 B_y}{\partial y^2} + \frac{19}{3} \left( \frac{\partial^3 B_x}{\partial x^3} + \frac{\partial^3 B_y}{\partial y^3} \right) \right] + O(h^3). \]  

(21)

However at coarse-fine interfaces (denoted, for instance, by an index pair \( (I, J) \)) the same procedure leads to a divergence

\[ \nabla_{2h} \cdot \tilde{B}_{I,J} = \frac{\partial B_x}{\partial x} + \frac{\partial B_y}{\partial y} - \frac{h}{32} \left[ \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) (B_x + B_y) \right] + O(h^3), \]  

(22)

introducing a 1st-order error in divergence. Further errors are introduced by the conservative fixup procedures typically used in AMR computations. This leads to the necessity of re-establishing a divergence-free field on a coarse grid after time subcycling has been completed and fine grid values along with conservative fixups have been introduced on the coarse grid. It is relatively easy to include the appropriate correction in the sequence of AMR computations:

1. trial time steps taken on a coarse grid determine placement of fine grids; if the initial coarse grid magnetic field is divergence-free to \( O(h^3) \) accuracy so is the field on the newly constructed fine grids;
2. the coarse grid is advanced to \( t + \Delta t \) through some procedure of solving the MHD equations; typically this would introduce divergence errors which remain not corrected for now;
3. time steps are taken on the fine grid; assume that the fine grid is at the finest level of resolution allowed during the computation, then after each time step required for time subcycling a projection correction is applied individually on each fine grid;
4. at the end of fine grid time subcycling, fine grid values are injected onto the coarse grid and the conservative fixup is applied; the projection procedure is now applied on the coarse grid by solving

\[ \Delta \phi = \nabla \cdot \tilde{B}, \]  

(23)

where \( \tilde{B} \) is the initial estimate of the magnetic field at \( t + \Delta t \), and setting

\[ B^{n+1} = \tilde{B} - \nabla \phi. \]  

(24)

The Poisson equation is discretized using a standard 5-point stencil and solved through an FFT-based method.