A new way to conserve total energy for Eulerian hydrodynamic simulations with self-gravity

Yan-Fei Jiang *, Mikhail Belyaev, Jeremy Goodman, James M. Stone

Department of Astrophysical Sciences, Princeton University, Princeton, NJ 08544, USA

HIGHLIGHTS

▶ Energy is conserved with our new method for Eulerian simulations with self-gravity.
▶ The new algorithm is stable, second order accurate and usually not more expensive.
▶ The new algorithm is important when energy error can severely affect the dynamics.

ARTICLE INFO

ABSTRACT

We propose a new method to conserve the total energy to round-off error in grid-based codes for hydrodynamic simulations with self-gravity. A formula for the energy flux due to the work done by the self-gravitational force is given, so the change in total energy can be written in conservative form. Numerical experiments with the code Athena show that the total energy is indeed conserved with our new algorithm and the new algorithm is second order accurate. We have performed a set of tests that show the numerical errors in the traditional, non-conservative algorithm can affect the dynamics of the system. The new algorithm only requires one extra solution of the Poisson equation, as compared to the traditional algorithm which includes self-gravity as a source term. If the Poisson solver takes a negligible fraction of the total simulation time, such as when FFTs are used, the new algorithm is almost as efficient as the original method. This new algorithm is useful in Eulerian hydrodynamic simulations with self-gravity, especially when results are sensitive to small energy errors, as for radiation pressure dominated flow.

1. Introduction

Self-gravity plays an important role in many astrophysical flows. For instance, it is central to massive star formation (e.g., McKee and Ostriker, 2007; Krumholz et al., 2009), supernova explosions (e.g., Nordhaus et al., 2010), planet formation in protoplanetary disks (e.g., Armitage, 2011), and structure formation in the early universe (e.g., Bertschinger, 1998). Numerical modeling of these systems requires that self-gravity be implemented correctly in hydrodynamical simulations, especially in problems for which energy balance is important. For example, the fate of a collapsing cloud depends on the total energy which is the sum of the potential, internal, and kinetic energies. For some systems, e.g. those having a relativistic equation of state with adiabatic gamma, $\gamma \approx 4/3$, the total energy can be much smaller than either the potential or internal energies in hydrostatic equilibrium. Thus, a

small numerical error in the computation of the potential energy can lead to a large fractional error in the total energy, potentially causing a bound system to become unbound (e.g., Jiang and Goodman, 2011) or vice versa. Energy conservation in a grid-based code with self-gravity is usually not guaranteed. The goal of this paper is to propose a new algorithm for grid-based codes, which conserves the sum of the internal, kinetic, and potential energies to round-off error, allowing accurate simulations of self-gravitating systems to be performed.

The change in the momentum and the energy due to self-gravity are typically added to the equations of hydrodynamics as source terms. In this case, the Euler equations read

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{1}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v} + \mathbf{P}) = -\rho \nabla \phi, \tag{2}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot [(E + P)\mathbf{v}] = -\rho \mathbf{v} \cdot \nabla \phi, \tag{3}$$

$$\nabla^2 \phi = 4\pi G \rho. \tag{4}$$

* Corresponding author.
  E-mail address: yanfei@astro.princeton.edu (Y.-F. Jiang).

Here ρ, $\mathbf{v}$, P are density, velocity, and pressure respectively, and E is the sum of the internal and kinetic energies. G is the gravitational constant, and ϕ is the gravitational potential, which is related to the density via the Poisson equation (4). We can also define the total energy as

$$E_{tot} \equiv E + \frac{1}{2}\rho\phi. \tag{5}$$

E_{tot} is a *globally* conserved quantity so that

$$\frac{\partial}{\partial t}\int d^3\mathbf{x}E_{tot} = 0 \tag{6}$$

for any isolated self-gravitating system. Expression (6) can be derived by integrating Eq. (3) over all of space and applying the divergence theorem (see e.g., Binney and Tremaine, 2008, Section 2.1).

Eqs. (2) and (3) are written in non-conservative form. Therefore, momentum and energy are usually not conserved to roundoff error when they are solved numerically. It is well known that if momentum is not conserved for a self-gravitating system, then the shape of the object cannot be kept very well when it moves across the simulation box (e.g., Edgar et al., 2005; Tasker et al., 2008). In order to conserve momentum, it is now common practice to write the momentum equation in conservative form as

$$\frac{\partial(\rho\mathbf{v})}{\partial t} + \nabla\cdot(\rho\mathbf{vv} + \mathbf{P} + \mathbf{T_g}) = 0, \tag{7}$$

where the gravitational tensor $\mathbf{T_g}$ is

$$\mathbf{T_g} = \frac{1}{4\pi G}\left[\nabla\phi\nabla\phi - \frac{1}{2}(\nabla\phi)\cdot(\nabla\phi)\mathbf{I}\right], \tag{8}$$

and $\mathbf{I}$ is the identity tensor. This equation is mathematically equivalent to the original momentum Eq. (2) with use of Eq. (4). However Eq. (8) can be solved in a completely different way numerically. Momentum change due to gravitational acceleration is no longer included as a source term. Instead, it is included as a flux term via $\mathbf{T_g}$, which means the total momentum can be conserved to round-off error numerically. This has already been implemented in many grid-based codes, such as Athena.

The improvement we propose in this paper is to also write the energy equation with self-gravity (3) in conservative form. Using this form of the equation, we devise a numerical algorithm that conserves the total energy with self-gravity to round-off error.

2. Formula for the energy flux

We begin by rewriting the energy Eq. (3) in the following form

$$\frac{\partial}{\partial t}\left(E + \frac{1}{2}\rho\phi\right) + \nabla\cdot\left[(E+P)\mathbf{v} + \mathbf{F_g}\right] = 0. \tag{9}$$

As long as the flux due to self-gravity $\mathbf{F_g}$ falls off faster than r^{-2} at large distances and the system is spatially bounded, Eq. (9) automatically implies that Eq. (6) is satisfied. We now solve for the form of $\mathbf{F_g}$.

Using Eq. (3), Eq. (9) can be written as

$$\frac{\partial}{\partial t}\left(\frac{1}{2}\rho\phi\right) + \nabla\cdot\mathbf{F_g} = \rho\mathbf{v}\cdot\nabla\phi. \tag{10}$$

Using the continuity equation, this becomes

$$\nabla\cdot\mathbf{F_g} = \frac{1}{2}\nabla\cdot(\rho\mathbf{v})\phi + \rho\mathbf{v}\cdot\nabla\phi - \frac{1}{2}\rho\dot{\phi}. \tag{11}$$

Here $\dot{\phi} \equiv \partial\phi/\partial t$ is the time rate of change of the potential. Differentiating the Poisson equation with respect to time yields

$$\nabla^2\dot{\phi} = 4\pi G\dot{\rho} = -4\pi G\nabla\cdot(\rho\mathbf{v}). \tag{12}$$

Substituting this in Eq. (11) gives

$$\nabla\cdot\mathbf{F_g} = \nabla\cdot(\rho\mathbf{v})\phi + \rho\mathbf{v}\cdot\nabla\phi + \frac{1}{8\pi G}\left(\phi\nabla^2\dot{\phi} - \dot{\phi}\nabla^2\phi\right)$$
$$= \nabla\cdot\left[\rho\mathbf{v}\phi + \frac{1}{8\pi G}\left(\phi\nabla\dot{\phi} - \dot{\phi}\nabla\phi\right)\right], \tag{13}$$

so one form of the energy flux due to self-gravity is

$$\mathbf{F_g} = \frac{1}{8\pi G}\left(\phi\nabla\dot{\phi} - \dot{\phi}\nabla\phi\right) + \rho\mathbf{v}\phi. \tag{14}$$

Note that this form for the energy flux is not unique, and as discussed in Appendix A, alternative forms are available, which also satisfy Eq. (13). However, only the divergence of the gravitational tensor and energy flux are used to evolve the system. The gravitational tensor and energy flux themselves are not used directly. Thus, as long as equation (13) is satisfied, different forms of the energy flux are mathematically equivalent. We note that a similar formula for energy flux due to self-gravity has been proposed by Pen (1998).

In order to calculate the energy flux, we need to solve Eq. (12) for $\dot{\phi}$. Note that this equation has the same form as the Poisson equation, which means it can be solved with the same numerical technique (such as FFT for periodic boundary conditions). Details on how to calculate the energy flux $\mathbf{F_g}$ are given in the next section.

3. Numerical implementation

We implement the equations of hydrodynamics with self-gravity in conservative form using Athena (e.g., Stone et al., 2008), which is a MHD code that uses an unsplit, higher order Godunov scheme. Since the implementation of self-gravity is unchanged by the addition of a magnetic field, we focus on the hydrodynamic case here. With self-gravity, the following steps are needed in addition to the original algorithm described in Stone et al. (2008).

First, at time step n we need to compute the gravitational potential ϕ^n by solving the Poisson equation using the density distribution. For periodic boundary conditions, this can be done efficiently using the Fast Fourier Transform (FFT).

Second, after we get the left and right states at the cell interface for each direction (step 1 of 6.1 in Stone et al., 2008), we need to add the change due to self-gravity to the left and right states at the half time step $\delta t/2$. If primitive variables (density, velocity and pressure) are used for the reconstruction, we only need to add the gravitational acceleration $-\nabla\phi$ to the left and right velocity. No gravitational source terms need to be added to the pressure.

Third, after we get the fluxes at the interfaces from the Riemann solvers (step 7 in Stone et al., 2008), we solve Eq. (12) for $\dot{\phi}^{n+1/2}$, and the same numerical technique as for solving the Poisson equation can be used. We just need to replace ρ on the right hand side of the Poisson equation with $-\nabla\cdot(\rho\mathbf{v})$, where $\rho\mathbf{v}$ is the density flux from the Riemann solvers at each interface. The $\dot{\phi}^{n+1/2}$ we get is at the cell centers and is the time-averaged value for time step n.

Fourth, we update the density from time step n to $n+1$ and calculate the new potential ϕ^{n+1} based on the updated density distribution. Then, we use the averaged potential $(\phi^n + \phi^{n+1})/2$ and $\dot{\phi}^{n+1/2}$ to calculate the gravitational tensor $\mathbf{T_g}$ and the energy flux $\mathbf{F_g}$. The cell-centered potential ϕ is spatially-averaged to get the potential at cell interfaces.

Finally, we update the momentum and the total energy with $\mathbf{T_g}$ and $\mathbf{F_g}$. At each cell (i,j,k), if the cell size is $\delta x \times \delta y \times \delta z$, then the momentum change (take ρv_x for example) due to self-gravity is

$$(\rho v_x)_{i,j,k}^{n+1} = (\rho v_x)_{i,j,k}^n - \frac{\delta t}{\delta x}\left(\mathbf{T}_{\mathbf{g}xxi+1/2,j,k}^{n+1/2} - \mathbf{T}_{\mathbf{g}xxi-1/2,j,k}^{n+1/2}\right)$$
$$- \frac{\delta t}{\delta y}\left(\mathbf{T}_{\mathbf{g}yxi,j+1/2,k}^{n+1/2} - \mathbf{T}_{\mathbf{g}yxi,j-1/2,k}^{n+1/2}\right) - \frac{\delta t}{\delta z}\left(\mathbf{T}_{\mathbf{g}zxi,j,k+1/2}^{n+1/2} - \mathbf{T}_{\mathbf{g}zxi,j,k-1/2}^{n+1/2}\right)$$
$$\tag{15}$$

and the change in the total energy is

$$
\begin{aligned}
E_{i,j,k}^{n+1} = E_{i,j,k}^n &+ \left[\frac{1}{2}(\rho\phi)_{i,j,k}^n - \frac{1}{2}(\rho\phi)_{i,j,k}^{n+1}\right] \\
&- \frac{\delta t}{\delta x}\left(\mathbf{F}_{g_{xi+1/2,j,k}}^{n+1/2} - \mathbf{F}_{g_{xi-1/2,j,k}}^{n+1/2}\right) - \frac{\delta t}{\delta y}\left(\mathbf{F}_{g_{yi,j+1/2,k}}^{n+1/2} - \mathbf{F}_{g_{yi,j-1/2,k}}^{n+1/2}\right) \\
&- \frac{\delta t}{\delta z}\left(\mathbf{F}_{g_{zi,j,k+1/2}}^{n+1/2} - \mathbf{F}_{g_{zi,j,k-1/2}}^{n+1/2}\right).
\end{aligned}
\tag{16}
$$

Note that the definition of E does not include potential energy.

4. Numerical tests

To show that our algorithm is stable, second order accurate, and conserves the total energy to machine precision, we perform several tests in one, two, and three dimensions. We also compare the results of the new algorithm with a non-conservative one that has self-gravity added as a source term. In all cases, the non-conservative algorithm use the conservative form of the momentum — but not energy — equation. In the non-conservative algorithm, the gravitational potential ϕ^n is first calculated based on density at time step n. After the density is updated, the new potential ϕ^{n+1} at time step $n+1$ is calculated. Then the energy source term is added as $-(\rho\mathbf{v})^{n+1/2} \cdot \nabla(\phi^n + \phi^{n+1})/2$, where $(\rho\mathbf{v})^{n+1/2}$ is the average of the left and right sides of the density flux. In this way, the non-conservative algorithm can achieve second order accurate as long as energy error due to self-gravity is negligible.

4.1. Jeans collapse in 1D and 2D

All of our tests in 1D and 2D are based on the Jeans instability problem, where self-gravity is the dominant force for the dynamics. We initialize the background state to be a self-gravitating, uniform density medium and assume the fluid is adiabatic with sound speed $c_s^2 = \gamma P/\rho$. On top of this background state, we initialize a small-amplitude ($\delta\rho/\rho = 10^{-6}$) normal mode perturbation with wavevector k. The amplitude of the perturbation changes with time as $e^{i\omega t}$, where ω is given by the Jeans dispersion relation

$$
\omega^2 = k^2 c_s^2 - 4\pi G\rho.
\tag{17}
$$

The Jeans length is given by $\lambda_J = \sqrt{\pi c_s^2/G\rho}$, so that a normal mode perturbation with wavelength $\lambda < \lambda_J$ results in a propagating wave, whereas one with $\lambda > \lambda_J$ yields an instability that saturates by forming dense clumps. We focus on the unstable case for comparing the conservative and non-conservative algorithms. Table 1 lists the details of each of our simulations and shows the relevant numerical parameters used.

Table 1
From left to right, the columns designate the simulation label, the number of dimensions, the algorithm used (c for conservative, nc for nonconservative), the number of cells ($N_x \times N_y \times N_z$), the wavelength of the initial perturbation measured in cells (all cells have the same size), the Jeans wavelength measured in cells, the Mach number of the background fluid, and the adiabatic index of the gas. The Courant number for all simulations is 0.8, and the HLLC Riemann solver is used for all simulations.

sim	dim	alg	N	λ	λ_J	M_0	γ
A	1	c	256	256	128	0	5/3
B	1	c	16	16	32	0	5/3
B2	2	c	23×23	16	32	0	5/3
C	1	c	32	32	64	0	5/3
C2	2	c	45×45	32	64	0	5/3
D	1	c	64	64	128	0	5/3
D2	2	c	91×91	64	128	0	5/3
E	1	c	128	128	256	0	5/3
E2	2	c	181×181	128	256	0	5/3
F	1	c	256	256	512	0	5/3
F2	2	c	362×362	256	512	0	5/3
G	1	c	512	512	1024	0	5/3
H	2	nc	22×22	16	8	0	5/3
I	2	nc	45×45	32	16	0	5/3
J	2	nc	91×91	64	32	0	5/3
Jc	2	c	91×91	64	32	0	5/3
K	2	nc	181×181	128	64	0	5/3
L	2	nc	362×362	256	128	0	5/3
M	2	nc	91×91	64	32	0	4/3
N	2	nc	181×181	128	64	0	4/3
O	2	nc	362×362	256	128	0	4/3
P	2	c	91×91	64	32	10	5/3

4.1.1. 1D tests

The first tests we perform are an accuracy and a convergence test in 1D. In Fig. 1, the left panel shows the time-evolution of the square of the perturbed gravitational potential integrated over the simulation domain for simulation A, which is for an unstable mode having $\lambda = 2\lambda_J$. The points show values from the simulations with time measured in units of $(k_J c_s)^{-1}$ and distance measured in units of λ_J. The gravitational potential is thus measured in units of $(\lambda_J k_J c_s)^2 = (2\pi c_s)^2$. The solid line shows the analytical solution obtained from the dispersion relation (17). It is clear that the code accurately captures the exponential growth in the linear phase of the instability through to saturation. The right panel of Fig. 1 shows convergence of the new algorithm with resolution. The initial perturbation has $\lambda = \lambda_J/2$ and corresponds to a stable perturbation. The points are for simulations B-F (the only thing that varies between the simulations is resolution) and the x-axis shows the number of cells per Jeans length. The y-axis shows the error in the $\mathcal{L}_2$ norm between the numerical and analytical solutions for Φ after one period ($t = 2\pi/\omega$). The error in the $\mathcal{L}_2$ norm can be represented as
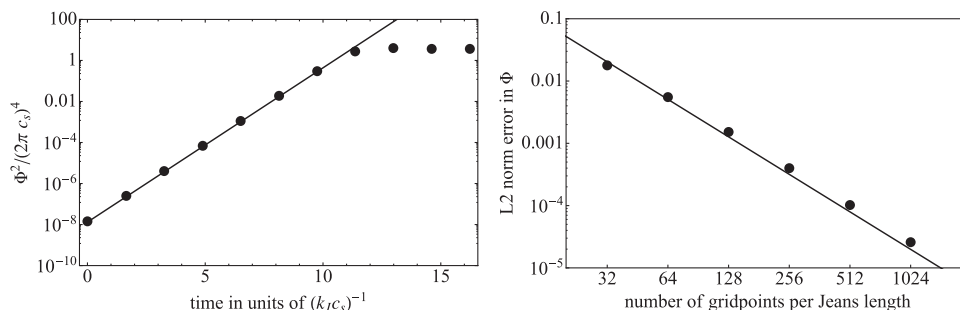

Fig. 1. *Left*: Comparison of the analytical and numerical solutions for simulation A. The analytical solution for the linear stage of growth is shown by the line, and the numerical solution is shown with points. At late times, the numerical solution deviates from the linear analytical solution because the instability becomes non-linear and saturates. *Right*: Convergence study for the conservative algorithm. The points correspond to simulations B − F, and a line with slope indicating second order convergence is plotted for reference.

$$\epsilon(\mathcal{L}_2) = \frac{\sqrt{\sum_i \left(\Phi_i - \tilde{\Phi}_i\right)^2}}{\sqrt{\sum_i \Phi_i^2}}, \tag{18}$$

where Φ_i is the value of the exact solution for the potential at point i, $\tilde{\Phi}_i$ is the value of the numerical solution, and the summation runs over all points in the simulation domain. For reference, we also plot a line whose slope indicates second order convergence, and which confirms that the new algorithm does indeed converge at second order.

To test the conservation properties of the new algorithm in comparison with the old one, we compute the fractional energy error at timestep n to be $|E^n - E^0|/E^0$, where

$$E^n = \sum_i \left(\frac{1}{2}\rho_i^n (v_i^n)^2 + \frac{P_i^n}{\gamma - 1} + \frac{1}{2}\rho_i^n \Phi_i^n\right). \tag{19}$$

Indeed, our new algorithm conserves the total energy to round-off error. However, in 1D, the original algorithm also conserves the total energy to very high precision. Thus, we need to go to a higher number of dimensions to demonstrate the superior energy conservation properties of the new algorithm.

4.1.2. 2D linear wave tests

To see how our new algorithm and the non-conservative algorithm behave in 2D, we again initialize an unstable eigenmode with $\lambda = 2\lambda_J$, just as in the 1D case. Now, however, we choose $k_x/k_y = 1$ so that the wavefronts are not aligned with the grid.

We perform two kinds of tests for the 2D case, which are analogous to the tests in the 1D case. The first is a convergence test of the conservative and non-conservative algorithms, for a stable mode ($\lambda < \lambda_J$). The second, is a test of the energy conservation for the two algorithms for an unstable mode ($\lambda > \lambda_J$).

Fig. 2 shows the results of the convergence test for the conservative algorithm (simulations B2-F2). On a log–log graph, we plot the $\mathcal{L}_2$ norm error in ρ as a function of resolution after one oscillation period for a mode having $\lambda = \lambda_J/2$, and an initial amplitude of $\delta\rho/\rho = 10^{-6}$. The points denote the simulation results, and the slope of the solid line shows second order convergence. It is clear that the conservative algorithm does indeed converge at second order. We have tested that the non-conservative algorithm also converges at second order for this the stable mode. Because the results are essentially identical to the conservative algorithm, we do not plot them.

We now describe tests of energy conservation, using an unstable normal mode. In 2D, in order to make the normal mode collapse to a filament as opposed to a sheet, we increase the density perturbation in the center of the simulation box by 2% relative to the edges. We find that the energy conservation properties of the
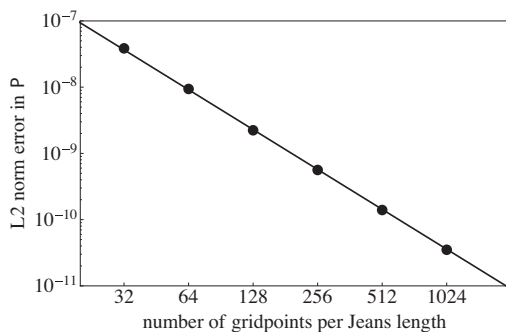

Fig. 2. Convergence study for the conservative algorithm in 2D. The points correspond to simulations $B2 - F2$, and a line with slope indicating second order convergence is plotted for reference. The non-conservative algorithm gives almost the same result for this stable propagation linear mode.

Table 2
The energy error for the non-conservative simulations in 2D as a function of resolution (increasing resolution going down). The error shown is the fractional error in the total energy between the beginning of the simulation and when the perturbation as virialized. The simulations H, I, J, K, L are done with adiabatic index $\gamma = 5/3$ while the simulations M, N, O are done with adiabatic index $\gamma = 4/3$. Note that in the case $\gamma = 4/3$, the initial total energy is twice that of the case $\gamma = 5/3$ (neglecting the energy of the perturbation).

Label	Error	Label	Error
H	.65	–	–
I	.32	–	–
J	.18	M	.09
K	.09	N	.05
L	.05	O	.03

non-conservative algorithm are drastically degraded relative to the 1D case. Table 2 shows the fractional energy error for the non-conservative 2D, $\gamma = 5/3$ simulations (H − L), and the nonconservative 2D, $\gamma = 4/3$ simulations (M − O). It is clear from the table, that in 2D the energy error for the non-conservative algorithm can be of order the total energy, although the error does decrease with increasing resolution. On the other hand, the conservative algorithm still conserves the total energy to machine precision for the same set of tests.

Note that the initial total energy (ignoring the energy of the perturbation) is roughly $E = P/(\gamma - 1)$. Thus, the initial total energy for the $\gamma = 4/3$ case is twice that for the $\gamma = 5/3$ case. This means that although the fractional energy error for the $\gamma = 5/3$ case is roughly double that of the $\gamma = 4/3$ case, the absolute value of the accumulated energy error is roughly the same. We have also made sure that the total energy is conserved for the conservative algorithm if we advect the background medium with respect to the grid at Mach 10 (simulation P).

The main cause of the discrepancy between the 1D and the 2D case is likely to be twofold. First, the wavevector is no longer aligned with the grid, so the 2D case is less symmetric than the 1D case. Second, the collapse proceeds further in the 2D case: to a filament in 2D as opposed to a sheet in 1D. As self-gravitational energy is a global quantity, the energy error caused by the source terms of self-gravity has different dependence on time step and grid size compared with the normal local truncation error. This energy error will be at a maximum when the gravitational potential changes most rapidly. When the total numerical error is dominated by the gravitational energy error for the non-conservative algorithm, Table 2 shows that the error scales roughly as $\mathcal{O}(1/N)$. However, when the energy error from self-gravitational source terms is much smaller than the usual truncation error as in the case of Fig. 2, the non-conservative algorithm shows second order convergence with resolution.

Fig. 3 shows the time-evolution of the energy error for simulation H. Overplotted is the kinetic energy in the simulation, normalized to the maximum kinetic energy. It is evident that most of the error is accumulated in a short interval around the time when the kinetic energy, and hence the infall velocity are large. This is also when the gravitational potential changes most rapidly, making the energy error large for the non-conservative algorithm.

For reference, Fig. 4 shows three snapshots of the density from the 2D simulation which has a resolution of 128 cells per Jeans length. The first snapshot is taken during the linear phase of exponential growth, the second during the non-linear phase when matter is collapsing to a sheet, the third during the phase when matter is collapsing to a filament, and the fourth during the final, virialized state.

In order to see the possible effect on the dynamics due to the energy error, we compare the density distribution from simulations J and Jc. The two simulations have exactly the same parameters but simulation J uses the non-conservative algorithm,
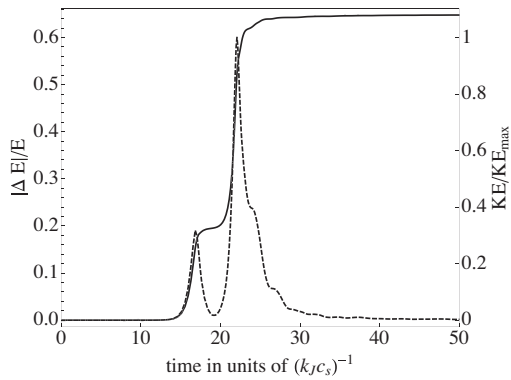
Fig. 3. The solid line shows the fractional error in the total energy as a function of time for simulation H. The dashed line shows the kinetic energy as a fraction of the maximum kinetic energy in the simulation. It is clear that the error accumulates when the kinetic energy is highest. The plateau between $t = 10$ and $t = 20$ corresponds to the phase in which the collapsing object has already finished collapsing to a sheet, but is just starting to collapse to a filament.

whereas simulation Jc uses the conservative one. Fig. 5 shows that the density distributions are different depending on whether the total energy is conserved or not. When the total energy is conserved (right panel of Fig. 5), the formed clump is symmetric, as

it should be, because it is formed from a symmetric filament. However, for the non-conservative algorithm, the symmetry is lost (left panel of Fig. 5). Moreover, rotation is generated with the non-conservative algorithm, which should not be the case, since the initial angular momentum is zero. This is likely because the energy error of the non-conservative algorithm changes the distribution of gas pressure, which causes the rotation of the fluid.

4.2. Collapse of a polytropic sphere in 3D

The test that shows the difference between the conservative and non-conservative algorithms most clearly in 3D is the collapse of a polytropic sphere. This test is also relevant to many astrophysical systems where self-gravity is important, such as star formation and planet formation in protoplanetary disks. In 3D, the adiabatic index $\gamma = 4/3$ is a critical case when the total energy of a hydrostatic self-gravitating sphere becomes zero. The dynamics of the sphere will be sensitive to the energy error made in the numerical solution, which may change the sign of the total energy in the solution. The $\gamma = 4/3$ case is relevant for massive stars when gravity is balanced by a radiation pressure gradient (e.g., Jiang and Goodman, 2011).

We initialize a polytropic sphere with polytropic index $n = 3$ in a periodic domain, so the radial density $\rho(r)$ and pressure $P(r)$ profiles are given by
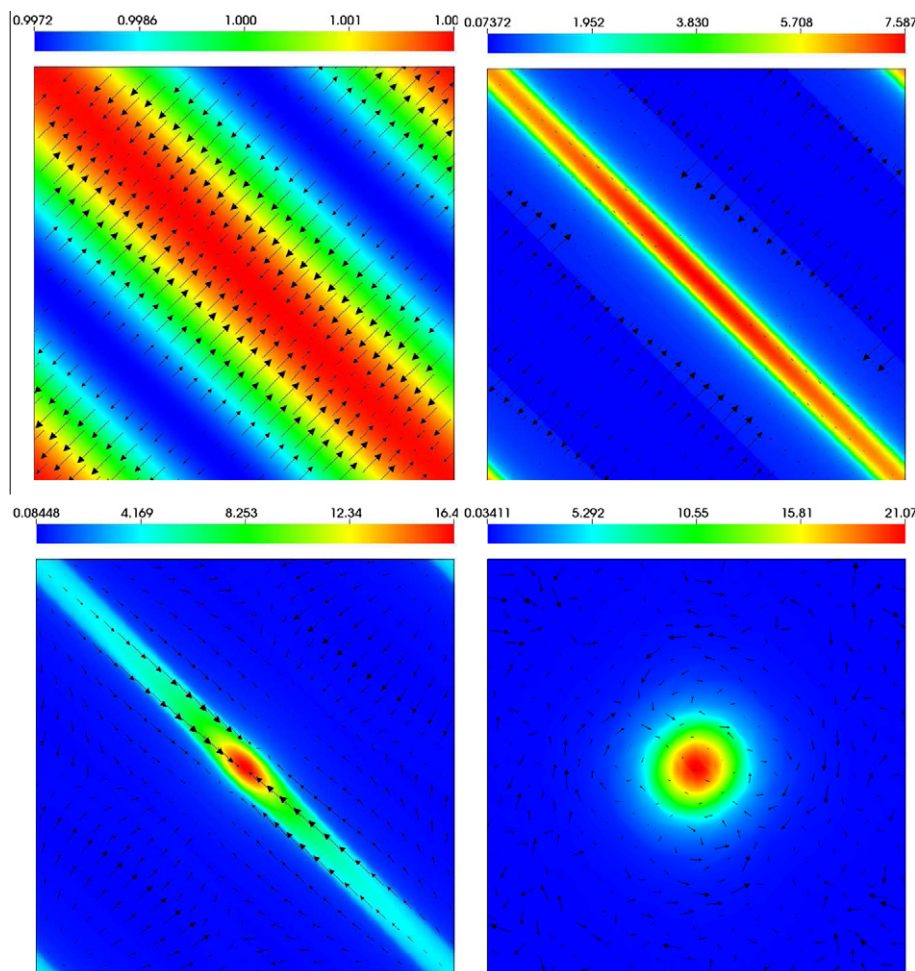

Fig. 4. Snapshots of the density and the velocity field from simulation L. The scale of the density for each figure is given by the corresponding color bar, and the initial background density is $\rho = 1$. The magnitude scale of the velocity vectors is arbitrary, and varies from figure to figure. The upper left panel is taken at $t = 9.2$ $(k_J c_s)^{-1}$, during the linear stage of the collapse. The upper right panel shows the simulation at $t = 18$ $(k_J c_s)^{-1}$, in the non-linear regime when the density distribution has collapsed to a sheet. The lower left panel shows the collapse of the sheet to a filament at $t = 21$ $(k_J c_s)^{-1}$. The lower right panel shows the final, virialized filament at $t = 73 (k_J c_s)^{-1}$.
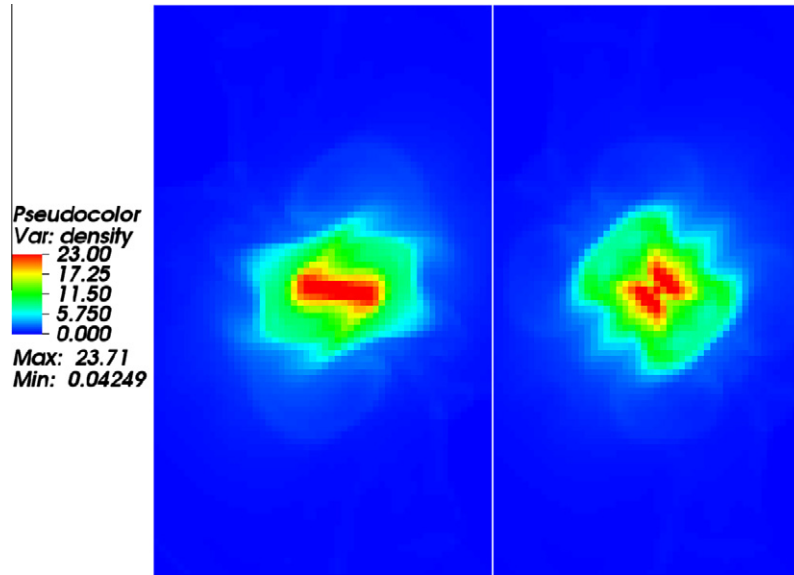
Fig. 5. Comparison of the density distribution for the conservative and non-conservative algorithm for simulations J and Jc in Table 1. The snapshots are taken at time $t = 1.1$, when the collapse stops and the energy error reaches its maximum value for the simulation using the non-conservative algorithm. The left panel is the density distribution from the non-conservative algorithm while the right panel is the result from the conservative algorithm.

$$\frac{1}{r^2}\frac{d}{dr}\left(\frac{r^2}{\rho}\frac{dP}{dr}\right) = -4\pi G\rho, \quad \frac{P}{P_{c,0}} = \left(\frac{\rho}{\rho_{c,0}}\right)^{4/3}. \tag{20}$$

In our units, $4\pi G = 1$, the central density is $\rho_{c,0} = 1$, and the central pressure in hydrostatic equilibrium is $P_{c,0} = 8.09 \times 10^{-5}$. The adiabatic index $\gamma = 1.36$ is chosen to be close to $4/3$ in order to clearly demonstrate the difference between the two algorithms. The background density in the simulation is set to 0.005. The diameter of the sphere is 0.16, which is 20% the size of the simulation box, and the sphere is located at the center of the simulation box. The resolution we use is 256^3.

Initially, we decrease the central pressure to be a fraction of the pressure in hydrostatic equilibrium and let the sphere evolve under its own gravity. Let us estimate how much the size of the polytropic sphere should change due to a decrease in the central pressure. Starting from the virial theorem and using simple scaling relations, it is straightforward to derive that in hydrostatic equilibrium

$$M^{2-\gamma}R^{3\gamma-4} = \alpha(\gamma)P_{c,0}\rho_{c,0}^{-\gamma}. \tag{21}$$

Here $\alpha(\gamma)$ is a constant that depends only on γ, R is the radius of the sphere, and M is the total mass. If we assume homologous adiabatic collapse, then the right hand side of Eq. (21) is a constant during the collapse. Suppose now that we reduce the initial pressure to be a fraction ϵ of the pressure in hydrostatic equilibrium. We can then solve for the final radius, R, in terms of the initial radius R_0 as

$$\frac{R}{R_0} = \epsilon^{\frac{1}{3\gamma-4}}. \tag{22}$$

In our simulations, we use $\gamma = 1.36$ and $\epsilon = .93$, which gives $R/R_0 \approx .4$. Thus, we expect to get a large-amplitude, spherically-symmetric, periodic solution. Moreover, since the fractional radius of the sphere is small compared to the box size, we expect spherical symmetry to be maintained quite well, even though we use periodic boundary conditions.

We start two simulations with exactly the same initial conditions but one uses the non-conservative algorithm whereas the other uses the conservative one. The time-evolution of the central density from the two simulations is shown in Fig. 6. Both simula-

tions give almost the same result for $t < 15$, but after the initial phase of the collapse, they show quite different behaviors with the non-conservative algorithm attaining a much larger peak central density. Snapshots of the density distribution from the two simulations through the plane $y = 0$ at times $t = 460$, $t = 680$, and $t = 900$ are shown in Fig. 7. Radial profiles of the spheres at time $t = 460$ and $t = 900$ through the line $(x = 0, y = 0)$ are examined in Fig. 8. At the early time, $t = 460$, both algorithms are able to keep the spherical symmetry of the object, although non-spherical structures have already appeared with the non-conservative algorithm (top panels of Fig. 7). These non-spherical structures are amplified by time $t = 680$. Eventually, the initial density profile is destroyed for the non-conservative algorithm, and a low density hole is formed at the center of the sphere at time $t = 900$. On the other hand, the conservative algorithm (bottom panels of Fig. 7) yields a periodic solution and keeps the spherical symmetry and initial profile of the sphere much better during the collapse than
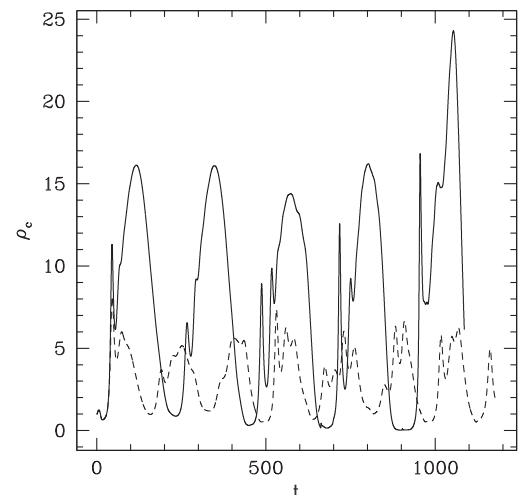

Fig. 6. Time evolution of the central density in the polytropic sphere test from two simulations that use the same initial conditions but different numerical algorithms. The solid line is the result using the non-conservative algorithm whereas the dashed line is the result using the conservative one.
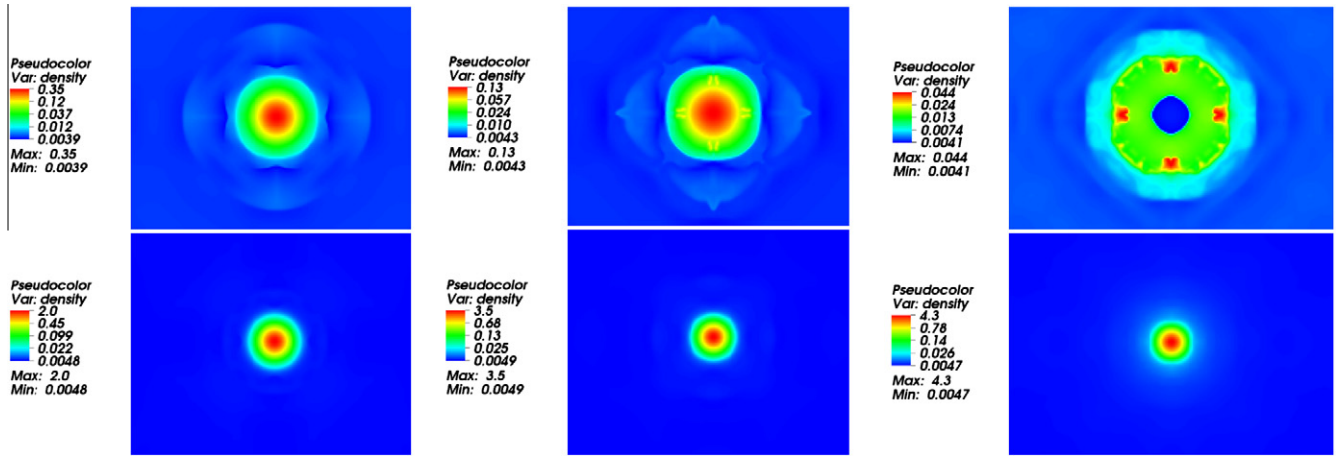
Fig. 7. Snaphsots of the density distributions from the polytropic sphere test taken at different times through the plane $y = 0$ from two simulations using different algorithms. From left to right, the times are $t = 460, 680, 900$, and in each case, only the central region is shown. The two simulations start from exactly the same initial conditions. The top panels correspond to the non-conservative algorithm whereas the bottom panels correspond to the conservative one. A central hole is created with the non-conservative algorithm and the simulation shows strong deviations from spherical symmetry.
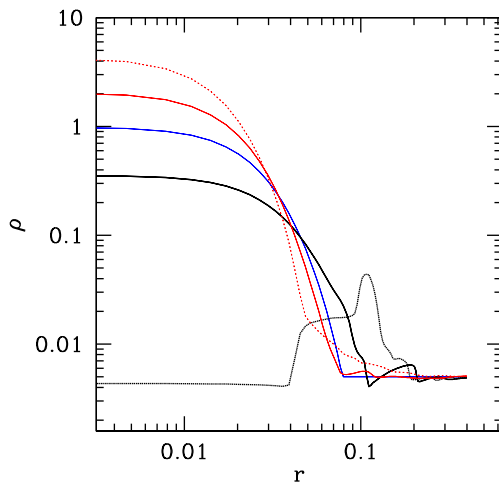

Fig. 8. Density profiles at different times on the line $(x = 0, y = 0)$ from the 3D polytropic sphere test using two different numerical algorithms. The blue line is the initial density profile, which is the same for both simulations. The black lines are results for the simulation using the non-conservative algorithm whereas the red lines are results for the simulation using the conservative algorithm. The solid black and red lines are measured at time $t = 460$ while the dashed black and red lines are measured at time $t = 900$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the nonconservative one. In conclusion, this test shows that for $\gamma \approx 4/3$ in 3D, the dynamics are significantly affected by the choice of algorithm. The conservative algorithm is favored, since it better preserves spherical symmetry throughout the collapse and yields a periodic solution as expected.

5. Performance

In order to calculate the time derivative of the potential, we need to solve one extra Poisson equation compared to the old method. If the Poisson solver takes a significant fraction of the total simulation time, the code will be slowed down by almost a factor of two for this new feature. If the time spent in the Poisson solver is negligible, then the new code is almost as efficient as the old one. For all our simulations, we use periodic boundary conditions and solve the Poisson equation using FFTs. In this case, the new algorithm is almost as efficient as the original method.

6. Summary and discussion

We have developed a new algorithm to conserve total energy to round-off error for Eulerian hydrodynamical simulations incorporating self-gravity. We have implemented this new algorithm in Athena and shown that it conserves total energy to round-off error as expected. By comparing a set of tests in 1D, 2D and 3D with two different numerical algorithms for self-gravity (the new conservative algorithm and the traditional non-conservative algorithm), we have shown that the numerical error made in the traditional algorithm can change the dynamics significantly. From our numerical experiments, we conclude that the conservative algorithm will be important when a small amount of energy error can dramatically affect the dynamics, such as in radiation-dominated systems with an adiabatic index close to 4/3.

We tested our new algorithm by implementing it in Athena, which uses an unsplit Godunov method to solve the equations of hydrodynamics. In principle, this algorithm can also be used in codes that implement an operator split scheme, such as ZEUS (Stone and Norman, 1992). In the application using ZEUS with self-gravity performed by Jiang and Goodman (2011), the energy error using the traditional non-conservative algorithm was positive. If ZEUS is used to study problems of star formation with the non-conservative algorithm, gravitationally bound clumps can acquire a positive total energy due to numerical errors and be destroyed as a result of numerical heating. However, if our conservative algorithm is used, this will not happen, since the total energy is conserved.

The new conservative algorithm is not necessary for all hydrodynamical simulations incorporating self-gravity. We find that for cases when the energy error is not a concern, the two algorithms give almost identical results. For example, for small oscillations of a polytropic sphere with an adiabatic index of $\gamma = 5/3$, the two algorithms yield very similar temporal and spatial behavior. In such cases, the old algorithm is just as good as the new one and is potentially more efficient.

Clancy Rowley for helpful suggestions during the implementation of this algorithm in Athena. J.G. was supported in part by the NSF Center for Magnetic Self-Organization under grant PHY-0821899.

Appendix A. Alternate forms

If the only properties required of the gravitational energy flux are that it satisfy Eq. (11) and have the correct physical dimensions, then $\mathbf{F}_g$ is not unique. Energy conservation is not affected by $\mathbf{F}_g \rightarrow \mathbf{F}_g + \mathbf{a}$ if $\nabla \cdot \mathbf{a} = 0$. For example,

$$\mathbf{a} = \frac{1}{4\pi G} \nabla \times (\phi \mathbf{v} \times \nabla\phi).$$

More generally, for arbitrary vector fields $\mathbf{b}$, the replacements

$$\left(\mathbf{F}_g, E_g\right) \rightarrow \left(\mathbf{F}_g + \partial_t \mathbf{b}, \ E_g - \nabla \cdot \mathbf{b}\right)$$

preserve $\partial_t E_g + \nabla \cdot F_g$ and therefore preserve the energy-conservation law. In particular, if one chooses $\mathbf{b} = -(16\pi G)^{-1}\nabla(\phi^2)$, then starting from Eq. (13) for $\mathbf{F}_g$ and $\frac{1}{2}\rho\phi$ for E_g, one obtains the alternate forms

$$\mathbf{F}_g = \rho\phi\mathbf{v} - \frac{1}{4\pi G}\dot{\phi}\nabla\phi, \quad E_g = \rho\phi + \frac{1}{8\pi G}|\nabla\phi|^2. \tag{A1}$$

Because they differ by a divergence, the two forms of E_g yield the same total gravitational energy when integrated over all space.

Additional criteria are clearly needed to select the "correct" forms of E_g and $\mathbf{F}_g$. One such criterion is galilean covariance. The energy density, energy flux, momentum density, and stress tensor of a nonselfgravitating fluid are

$$E = \frac{1}{2}\rho v^2 + u, \quad \mathbf{F} = (E + p)\mathbf{v}, \quad \mathbf{j} = \rho\mathbf{v}, \quad \mathbf{T} = \rho\mathbf{v}\mathbf{v} + p\mathbf{I},$$

where u is internal energy per unit volume. Under an infinitesimal galilean transformation $(\mathbf{v}, \rho, \mathbf{p}, \mathbf{u}) \rightarrow (\mathbf{v} + \delta\mathbf{V}, \rho, \mathbf{p}, \mathbf{u})$, these densities and fluxes transform to first order in $\delta\mathbf{V}$ as

$$E \rightarrow E + \mathbf{j}\cdot\delta\mathbf{V}, \quad \mathbf{j} \rightarrow \mathbf{j} + \rho\delta\mathbf{V},$$
$$\mathbf{F} \rightarrow \mathbf{F} + E\delta\mathbf{V} + \mathbf{T}\cdot\delta\mathbf{V}, \quad \mathbf{T} \rightarrow \mathbf{T} + 2\mathbf{j}\cdot\delta\mathbf{V}. \tag{A2}$$

It seems reasonable to require that the gravitational contributions to $E, \mathbf{F}$, and $\mathbf{T}$ should preserve the form of this transformation. Since the newtonian gravitational mass and momentum densities vanish under any sensible definition, E_g and $\mathbf{T}_g$ should be galilean invariants, while

$$\mathbf{F}_g \rightarrow \mathbf{F}_g + E_g\delta\mathbf{V} + \mathbf{T}_g\cdot\delta\mathbf{V}. \tag{A3}$$

Because $\nabla\phi$ and ϕ are galilean-invariant, $\mathbf{T}_g$ [Eq. (8)] and both forms of E_g are invariant. However, $\dot{\phi} \rightarrow \dot{\phi} - \delta\mathbf{V}\cdot\nabla\phi$, from which one can show that the forms (A1) of E_g and $\mathbf{F}_g$ are compatible with (A3), whereas the original forms $E_g = \rho\phi/2$ and $\mathbf{F}_g$ as given by Eq. (13) are not.

No matter what definitions are used for $(E_g, \mathbf{F}_g, \mathbf{T}_g)$, the same time evolution results if the conservative equations are solved exactly, provided that the definitions are mathematically equivalent to the original equations of motion Eqs. (1)–(4). However, the spatial distribution of gravitational energy and stress will depend upon the inertial frame used, and it is possible that numerical truncation errors may be sensitive to this dependence. The whole point of the present exercise is recast the equations so as to improve energy conservation in finite-difference or finite-volume approximations. Viewed in this light, numerical robustness becomes more important than formal elegance. We mistrust the form (A1) of the gravitational energy density because it is always non-negative in vacuum or near-vacuum regions: thus a small local error in the estimate of $|\nabla\phi|^2/8\pi G$ could cause a large error in the update of the kinetic or internal energies per unit mass. The original form $E_g = \rho\phi/2$ does not have this defect because it tends to zero smoothly with the mass density.

If it is important both that the gravitational terms be locally galilean covariant and that $E_g = \rho\phi/2$, then we may keep Eq. (13) for $\mathbf{F}_g$ but replace the gravitational stress tensor (8) with

$$\tilde{\mathbf{T}}_g = \frac{1}{8\pi G}[(\nabla\phi)(\nabla\phi) - \phi\nabla\nabla\phi] + \frac{1}{2}\rho\phi\mathbf{I}. \tag{A4}$$

The difference between (8) and (A4) can be written as

$$\tilde{\mathbf{T}}_g - \mathbf{T}_g = \frac{1}{16\pi G}\left[\mathbf{I}\nabla^2 - \nabla\nabla\right]\phi^2. \tag{A5}$$

Since the divergence of (A5) vanishes, the gravitational momentum density remains zero if $\mathbf{T}_g$ is replaced by $\tilde{\mathbf{T}}_g$.

References

Armitage, P.J., 2011. ARA&A 49, 195.
Bertschinger, E., 1998. ARA&A 36, 599.
Binney, J., Tremaine, S., 2008. In: Binney, J., Tremaine, S. (Eds.), Galactic dynamics, second ed. Princeton University Press.
Edgar, R.G., Gawryszczak, A., Walch, S., 2005. In: Protostars and Planets V.
Jiang, Y.-F., Goodman, J., 2011. ApJ 730, 45.
Krumholz, M.R., Klein, R.I., McKee, C.F., Offner, S.S.R., Cunningham, A.J., 2009. Science 323, 754.
McKee, C.F., Ostriker, E.C., 2007. ARA&A 45, 565.
Nordhaus, J., Burrows, A., Almgren, A., Bell, J., 2010. ApJ 720, 694.
Pen, U.-L., 1998. ApJS 115, 19.
Stone, J.M., Gardiner, T.A., Teuben, P., Hawley, J.F., Simon, J.B., 2008. ApJS 178, 137.
Stone, J.M., Norman, M.L., 1992. ApJS 80, 753.
Tasker, E.J., Brunino, R., Mitchell, N.L., Michielsen, D., Hopton, S., Pearce, F.R., Bryan, G.L., Theuns, T., 2008. MNRAS 390, 1267.