The numerical modeling of the collapse of molecular
cloud on adaptive nested mesh

Igor Kulikov

1 Institute of Computational Mathematics and Mathematical Geophysics SB RAS, 6-th
Lavrentyev avenue, Novosibirsk, Russian Federation
E-mail: kulikov@ssd.sscc.ru

Abstract. In paper, a numerical modeling of the collapse of molecular cloud is presented.
The numerical method for solving the equations of hydrodynamics is based on the extension
of HLL method for using of nested adaptive mesh technologies. In case of nested meshes, if
neighboring cells have the same size as the one considered cell, the Rieman problem is trivial.
If the neighboring cell is larger than a considered cell, then a uniform distribution of the parameters
over the large cell is assumed. Then a new subcell is allocated in a larger cell corresponding
to a size of the cell considered and being adjacent to it, after which the necessary data for a
computing. If a neighboring cell is less than a considered cell, then values around a small cell
are averaged. The energy behavior of the rotating molecular cloud is in quantitative agreement
with the results of other codes before the moment of collapse, and after the moment of collapse.
It should be noted that collapse occurs at exactly the same moment in time.

1. Introduction
The processes of collapse of molecular clouds and stars are being actively studied today in
connection with the appearance of a large number of observational data. The phenomenon of
collapse takes place both at the initial stage of the evolution of the star, and at the final stage of
a supernova explosion with a collapsing core. Mathematical modeling plays a more important
role in the theoretical study of such processes. One of the main problems in such problems is
the ratio of scales: when the size of the star is 10^9 meters the collapsed core can reach a size of
10^4 meter. Even with the use of powerful supercomputers, it is impossible to adequately model
such processes.

One of approaches to solving the collapse problems is adaptive grids, which allow the local
grinding of the grid. Here is an overview of the most known codes based on this approach.

ART [1]. The code is based on the technology of adaptive grid trees. Thus, the classical
adaptive mesh (AMR) is represented in the form of a regular (octet) tree structure, which
naturally lies on the classical data structures, and therefore all classical tree parallel algorithms
are applicable to them. To solve the Poisson equation, the relaxation method is used when the
Poisson equation is reformulated as:

\[ \Delta \Phi = 4\pi G \rho \rightarrow \lim_{\tau \to \infty} \left( \frac{\partial \Phi}{\partial \tau} = \Delta \Phi - 4\pi G \rho \right) \]

to solve the last equation is used the method of successive upper relaxation. The MUSCL scheme
is used to solve gas-dynamic equations.
CASTRO [2]. The program complex is based on the MUSCL and PPM (piecewise-parabolic method) methods using the AMR approach. To solve the Poisson equation, a multigrid method is used, and at each stage of the cycle the Gauss-Seidel method is used. An important feature of the code from the point of view of computational methods is the use of an approach based on the construction of a three-dimensional piecewise parabolic function [3], which is important for the invariance of the numerical method. In addition, more "smooth" symmetric constraints were used [4, 5].

ENZO [6]. At the heart of the software complex is the solution of the equations of magnetic gas dynamics with allowance for cosmological expansion. To simulate a collisionless component, the N-body model is used. The code includes a large number of subgrid processes: primordial chemical kinetics, cooling / heating functions, radiation transfer, as well as star formation processes and supernova explosion effects. Several solvers are used to solve hydrodynamic equations: PPM (implemented only for gas dynamics equations), MUSCL and finite-difference method. To solve the Poisson equation, an algorithm based on a fast Fourier transform is used. The so-called structured adaptive grid is used in the complex, the basic idea is to minimize the difference between the computational grid between neighboring cells. Such a structure allows the use of regular trees, where the subdomain is divided no more than twice, which increases the efficiency of using such computed grids.

GAMER [7]. The code implements the solution of the equation of gas dynamics using the AMR approach on graphic accelerators. To solve the equations of gas dynamics, the TVD approach is used, the combination of the method based on the fast Fourier transform and the method of successive upper relaxation is used to solve the Poisson equation. Perhaps the main feature of this complex is the implementation of the AMR approach on graphics cards. So the regular grid structure is naturally projected onto the GPU architecture, while the tree structure requires special approaches. This approach consists in using octets to define a grid, which is projected onto a separate stream of the graphics card. The main problem here is the formation of dummy cells for an octet, which is about 63 % of the time, but this procedure can be implemented independently for each octet.

RAMSES [8]. The code implements a numerical solution of the equations of gravitational gas dynamics using the AMR approach, based on splitting into octets. To solve the Poisson equation, a combination of the method based on the fast Fourier transform and the Gauss-Seidel method is used. To approximate the Poisson equation, a simple five-point finite-difference approximation is used, which was replaced by a more efficient 19-point approximation, which was implemented as an extension of the RAMSES code to the case of nonclassical gravity (MOND) [9].

However, adaptive meshes are based on grinding the grid no more than twice in neighboring cells, which is due to the peculiarities of software implementations and numerical methods. This approach actually does not allow reproducing multiscale processes.

A powerful alternative to adaptive grids is the nested grid approach, when in the cell a complete task with a sufficiently detailed grid is actually solved. In fact, the use of nested grids allows to reproduce any currents of the multi-scale type. The development of such an approach will allow, for example, to eliminate the gap problem between modeling the evolution of galaxies and the process of star formation, and also to reproduce in sufficient detail the turbulent flows and processes of nuclear collapse. The original approach to creating a computational model of the hydrodynamics of astrophysical objects based on the technology of nested grids will be presented in the article. In this article, we are not concerned with the issues of parallel implementation, although it is clear that the use of nested grids requires the balancing of processor load.

In the second section, a numerical scheme for solving the equations of gas dynamics and the Poisson equation will be described in detail, and an algorithm for rebuilding the embedded grid based on the mass criterion will also be presented. The third section will be devoted to modeling the process of cloud collapse. The conclusion is given in the fourth section.
2. The mathematical model
In a few papers in the field of computational fluid dynamics, instead of a non-conservative equation for internal energy, an equation for entropy is used, written in a conservative form [10, 11, 12, 13]. The undoubted advantage of this approach is the guarantee of non-decreasing entropy and the effect of negative pressure, as well as the possibility of correctly describing supersonic flows. There are also some limitations on the use of this approach (see discussion at work [14]).

\[
\frac{\partial}{\partial t} \begin{pmatrix}
\rho \\
\rho \vec{u}
\end{pmatrix} + \nabla \cdot \begin{pmatrix}
\rho \vec{u} \\
\rho S \\
\rho E
\end{pmatrix} = \begin{pmatrix}
0 \\
-\rho \nabla \Phi \\
0
\end{pmatrix}
\] (1)

where \(\rho\) – gas density, \(\vec{u}\) – velocity vector, \(S\) – entropy, \(p = S\rho\gamma\) – pressure, \(E = p/(\gamma - 1) + \rho \vec{u}^2/2\) – total mechanical energy, \(\gamma\) – adiabatic index, \(\nabla \Phi\) – the gradient of the potential, is determined from the Poisson equation.

\[\Delta \Phi = 4\pi G \rho\] (2)

To solve equations (1) and (2) was used the calculation schemes shown in the figure 1. In the following subsections, a more detailed description of the numerical schemes will be given.

2.1. Solution of the equations of gravitational hydrodynamics
To solve the hydrodynamic equations, a modification of the original numerical method based on a combination of the method of operator separation, the Godunov method and the HLL scheme was used. This method combines all the advantages of the above methods and has a high degree of parallelism. The details of the numerical scheme are described in [15]. The main idea of the method is to use the vector notation of the equations of hydrodynamics:

\[
\frac{\partial v}{\partial t} + \nabla \cdot f(v) = 0
\] (3)

where \(v\) – is a vector of conservative variables. For equation (3) the numerical scheme in one direction is written in the form:

\[
\frac{v_i^{n+1} - v_i^n}{\tau} + \frac{F_{i+1/2} - F_{i-1/2}}{h} = 0
\] (4)

Figure 1. The scheme of the organization of calculations
where \( F \) – solution of the Riemann problem. We omit the derivation of the numerical scheme on the basis of the use of conjugate equations and the method of separating operators and formulate the final form of the solution of the Riemann problem:

\[
F = \frac{f(v_L) + f(v_R)}{2} + \frac{c + \|\vec{u}\|}{2}(v_L - v_R)
\] (5)

For discretization using nested meshes, we introduce in the three-dimensional domain of the solution a uniform cubic root grid with the coordinates of the cell centers \( x_i = i \times h - h/2, \)
\( i = 1, \ldots, I_{\text{max}}, \)
\( y_k = k \times h - h/2, \)
\( k = 1, \ldots, K_{\text{max}}, \)
\( z_l = l \times h - h/2, \)
\( l = 1, \ldots, L_{\text{max}}, \)
where \( h \) – is the step of the root grid,
\( I_{\text{max}}, K_{\text{max}}, L_{\text{max}} \) – is the number of cells in the directions \( x, y, z. \) In the implementation, for convenience of organizing calculations and without losing the generality of the code \( I_{\text{max}} = K_{\text{max}} = L_{\text{max}} = N \) was used. In the cell with the number \((i, k, l)\) we introduce an embedded cubic grid with the coordinates of the cell centers \( x_{i,\text{nested}} = i \times h_{\text{nested}} - h_{\text{nested}}/2, \)
\( i = 1, \ldots, M, \)
\( y_{k,\text{nested}} = k \times h_{\text{nested}} - h_{\text{nested}}/2, \)
\( k = 1, \ldots, M, \)
\( z_{l,\text{nested}} = l \times h_{\text{nested}} - h_{\text{nested}}/2, \)
\( l = 1, \ldots, M, \)
h_{\text{nested}} – is the nested grid grid,
\( M \) – is the number of cells in the nested grid by directions \( x, y, z. \) The equations of hydrodynamics will be calculated for the values of the root and nested meshes placed in the cells. The Poisson equation for the root grid will also be calculated in cells, after which the solution will be projected onto the boundary nodes of the nested grid. To solve the Poisson equation on an embedded grid, the values of the potential (and density) will be located at the nodes of the nested grid. The detailed arrangement of quantities for the organization of calculations is given in the figure 2.

The solution of the equations of hydrodynamics (finding the solution of Riemann’s problems) occurs in two stages:

(i) solution of Riemann problems on all boundaries of a nested grid,
(ii) solving Riemann problems on all internal interfaces of a nested grid.

If the second part of finding the solution of Riemann’s problems is fairly trivial, then the first part requires a different way of computing, depending on the cell sizes of two adjacent nested
Figure 3. Three relations between the cells of adjacent nested grids. The cell for which the Riemann problem is solved is indicated by a pink color, the cell of a neighboring nested grid is indicated by a blue color.

meshes. In total, only three methods of the relationship between the cells of neighboring grids are possible (see figure 3). In the case of equal cell sizes (figure 3 (in the middle)) the solution of the Riemann problem is analogous to the solution of Riemann problems on internal interfaces of a nested grid and is trivial. In the case where the cell of an adjacent nested grid is larger than the one considered (figure 3 (left)), then a uniform distribution of the quantities over the blue cell is assumed, and even for the interface between the reduced blue cell and the pink cell, the Riemann problem is solved. In the case where the considered pink cell borders on several cells of a neighboring embedded grid (figure 3 (right)), then using the assumption of a uniform distribution of hydrodynamic quantities over the pink cell, Riemann problems are solved on all interfaces, averaged.

2.2. Solution of the Poisson equation

The solution of the Poisson equation is carried out in two stages:

(i) Solution of the Poisson equation on the root grid using the fast Fourier transform.

(ii) Solution of the Poisson equation on an embedded grid using the SOR method.

The detailed arrangement of quantities for the organization of calculations is shown in the figure 2. Also, we do not dwell on the solution of the Poisson equation in the first stage (a detailed description of the solution method can be found in [14]).

The Successive Over-Relaxation (SOR) method is an iterative process of finding the potential on an embedded grid for given initial and boundary conditions obtained from solving the Poisson equation on the root grid. The iterative process is organized as follows.

(i) On the internal nodes of the nested grid a numerical scheme for solving the Poisson equation is written:

\[
\frac{\phi_{i+1,k,l} - 2\phi_{i,k,l} + \phi_{i-1,k,l}}{h_{nested}^2} + \frac{\phi_{i,k+1,l} - 2\phi_{i,k,l} + \phi_{i,k-1,l}}{h_{nested}^2} + \frac{\phi_{i,k,l+1} - 2\phi_{i,k,l} + \phi_{i,k,l-1}}{h_{nested}^2} = 4\pi G \rho_{i,k,l}
\]

(ii) At each iteration a discrepancy is defined:

\[
R_{i,k,l} = \left( \frac{\phi_{i-1,k,l}^{old} + \phi_{i+1,k,l}^{old} + \phi_{i,k+1,l}^{old} + \phi_{i,k-1,l}^{old} + \phi_{i,k,l+1}^{old} + \phi_{i,k,l-1}^{old} - 6\phi_{i,k,l}^{old}}{6} \right) - 4\pi G \rho_{i,k,l} h_{nested}^2
\]

(iii) But we approximate the solution by the equation:

\[
\phi_{i,k,l}^{new} = \phi_{i,k,l}^{old} + \frac{1}{6} \omega R_{i,k,l}
\]
where $\omega$ – relaxation parameter.

(iv) When a small residual is reached or when the specified number of iterations is exceeded

$$\phi_{i,k,l} = \phi_{i,k,l}^{new}$$

A similar approach to solving the Poisson equation has proved itself in a number of programming codes, for example, in the code GAMER [7].

2.3. Selecting the time step
Step by time $\tau$ is calculated from the Courant condition

$$\frac{\tau \times (c + \|\vec{u}\|)}{h_{min}} = CFL < 1$$

where $CFL$ – Courant number, $c = \sqrt{\gamma p/\rho}$ – sound speed, $h_{min}$ – the size of the minimal cell of the nested grid.

2.4. Regularization of the solution
At the final stage of solving the hydrodynamic equations, the solution is corrected. In the case of a gas-vacuum boundary using the formula:

$$|\vec{u}| = \sqrt{2(E - \epsilon), (E - \vec{u}^2/2)/E} \geq 10^{-3}$$

in the rest of the region there is an adjustment, which guarantees nondecreasing entropy:

$$|\rho \epsilon| = \left(\rho E - \frac{\rho \vec{u}^2}{2}\right), (E - \vec{u}^2/2)/E < 10^{-3}$$

Such a modification provides a detailed balance of energies and guarantees nondecreasing entropy.

2.5. Algorithm for rebuilding the grid
Restructuring of the grid takes place according to the criterion of the cell mass of the root grid. The size of the nested grid is determined from the condition:

$$M = 2^{C_1[log(\rho)]} + C_2$$

where $C_{1,2}$ – scaling constants, which are selected depending on the task (the characteristic density) and the requirements for a minimum resolution. So, for the dimensionless density $\rho$ in the problems of cloud collapse, the equation (8) is rewritten as:

$$M = 2^{2[log(\rho)]} + 3$$

that is, the unit density value corresponds to a nested grid of size $[2^3 \times 2^3 \times 2^3]$, increasing the same density by an order of magnitude leads to a rearrangement of the grid with $[8 \times 8 \times 8]$ on $[32 \times 32 \times 32]$. Verification of the necessity of grid reorganization is carried out at each step in time, therefore the grid changes no more than twice. The grid reconstruction scheme is shown in the figure 4. Figure 4 shows the schemes of projections of conservative quantities (density, angular momentum, entropy density and total energy). After the reorganization of the grid, the nonconservative (primitive in the case of relativistic hydrodynamics) are calculated from conservative variables.
3. Simulation of the collapse of the cloud
We will simulate a gas cloud bounded by a sphere of radius $R_0 = 100$ parsecs, with mass $M_g = 10^7 M_\odot$, with density distribution $\rho(r) \simeq 1/r$ and temperature $T \approx 2000$. The adiabatic exponent corresponds to hydrogen $\gamma = 5/3$. The speed of sound is $c \approx 3.8$ km/s. As the size values, we choose the following values: $L_0 = 100$ parsec, $\rho_0 = 1.2 \cdot 10^{-18}$ /$^3$, $v_0 = 21$ km/s. Then, in dimensionless quantities, the problem is posed as follows: $\rho = 1.0$ – the density of the gas cloud at the center, $p = 2 \times 10^{-2}$ – is the pressure in the gas cloud at the center, $\gamma = 5/3$ – is the adiabatic exponent, $[0; 3.2]^{3}$ – is the computational domain. The behavior of various types of energies using the original approach described in this article, and using AstroPhi [16] code is given in the figure 5. We note that the behavior of the energies is qualitatively close enough for both realizations. However, we note that the simulation shown in Figure 5 with the AstroPhi code was about 20 hours using 32 cores of the resources of the Siberian Supercomputer Center (SCCC), while code based on nested grids allowed to obtain the result in two hours using a single core resources of SSCC. Nested adaptive grids are a powerful tool for modeling hydrodynamic flows that allow you to adapt the computational process to the features of the solution. At the same time, the organization of calculations is quite complex, although it is obvious that the use of the power of many classical processors, graphics accelerators and accelerators Intel Xeon Phi will further improve code performance based on nested grids. In the feature papers, the questions of parallel implementation will be described in detail.

4. Discussion
In a description of the paper are not three important topics which, although not the subject of this paper has been affected, still require a description.

(i) Using adaptive nested mesh implies the possibility of multiscale modeling. Such an approach can, for example, be used in the problems of star formation in evolution of galaxies. In this case, the evolution or collision of galaxies can be modeled on the root mesh, and the cells of root mesh with a characteristic size and time, corresponding to the evolution of the interstellar medium (ISM), can be used to directly model of the star formation as an individual problem. A similar approach can be used to simulate ISM turbulence with the ability to reproduce protostellar and protoplanetary systems.
Figure 5. The behavior of various types of energy obtained as a result of computational experiments using nested grids (HydroBox3D code) with an effective resolution of $512^3$ and using the AstroPhi code using a regular grid $512^3$.

(ii) Now in the code, multi-core architectures (for example, for Intel Xeon Phi) was developed, which is develop by parallel implementation of iterations of main computational cycles. The main interest is the development for the massively parallel architectures. For such an implementation would use an approach based on the splitting of two groups of processes: processes, that perform calculations on the root mesh, and processes, that dynamically appointed for the nested mesh. The method for a definition such groups requires a priori information about the problem and is adjusted in the course of computing. The distribution of such computations is planned to be described in next papers.

(iii) In the paper, the first order accuracy method was described. Undoubtedly, the use of high-order methods will provide an even better numerical solution. It’s also obvious, that the piecewise-parabolic method on local stencil (PPML) requires a modification for adjacent cells of different sizes to create local parabolas. We need to change two fundamental points in method: to construct a parabola from the average values on the boundary cells (now in PPML method uses a non-compact template), thus avoiding problems with different size cells, and integrate the parabola within the cell, defining the value, arriving at the boundary between cells (similar to the work [13]), which also requires information only of local cell. The description of such a method is planned to be described in next papers.

5. Conclusion
The original approach to the solution of the equations of gravitational hydrodynamics based on the technology of nested grids is presented in the article. The organization of a numerical solution of the equations of gas dynamics and Poisson equations is described in detail. The algorithm of grinding and enlargement of nested meshes on the basis of the mass criterion is given. The process of cloud collapse is modeled. The behavior of various types of energies is shown in the collapse of the cloud, the simulation results are compared using nested grid technology and using the regular detailed grid implemented in AstroPhi code.
Acknowledgments
This work was supported by Russian Science Foundation (project no. 18-11-00044).

References
[1] Kravtsov A, Klypin A and Hoffman Y 2002 The Astrophysical Journal 571 563-575
[2] Almgren A 2010 The Astrophysical Journal 715 1221-1238
[3] Miller G and Colella P 2002 Journal of Computational Physics 183 26-82
[4] Colella P and Sekora M 2008 Journal of Computational Physics 227 7069-7076
[5] McCorquodale P and Colella P 2011 Communications in Applied Mathematics and Computational Science 6 1-25
[6] Bryan G 2014 The Astrophysical Journal Supplement Series 211 (2) 19
[7] Schive H, Tsai Y and Chiuhe T 2010 The Astrophysical Journal 186 457-484
[8] Teyssier R 2002 Astronomy & Astrophysics 385 337-364
[9] Candlish G, Smith R and Fellhauer M 2015 Monthly Notices of the Royal Astronomical Society 446 (1) 1060-1070
[10] Balsara D and Spicer D 1999 Journal of Computational Physics 148 133-148
[11] Ryu D, Ostriker J, Kang H and Cen R 1993 The Astrophysical Journal 414 1-19
[12] Springel V and Hernquist L 2002 Monthly Notices of the Royal Astronomical Society 333 649-664
[13] Godunov S and Kulikov I 2014 Computational Mathematics and Mathematical Physics 54 1012-1024
[14] Kulikov I and Vorobyov E 2016 Journal of Computational Physics 317 318-346
[15] Kulikov I, Chernykh I and Tutukov A 2018 Lobachevskii Journal of Mathematics 39 in print
[16] Kulikov I, Chernykh I, Snytnikov A, Glinskiy B and Tutukov A 2015 Computer Physics Communications 186 71-80