HydroBox3D: Parallel & Distributed Hydrodynamical Code for Numerical Simulation of Supernova Ia

Igor Kulikov(✉), Igor Chernykh, Dmitry Karavaev, Evgeny Berendeev, and Viktor Protasov

Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Lavrentjeva 6, 630090 Novosibirsk, Russia
kulikov@ssd.sscc.ru

Abstract. In the paper a new parallel & distributed hydrodynamical code HydroBox3D for numerical simulation of supernovae Ia type explosion was described. The HydroBox3D code is created on basis of combination the adaptive nested mesh for hydrodynamical simulation of supernovae explosion and the regular mesh is second level of nested mesh for hydrodynamical simulation of nuclear reaction. The adaptive nested mesh code for shared memory architecture with using Intel Optane technology was developed. The second level of nested mesh code for Intel Xeon Phi KNL supercomputer was developed. The HydroBox3D code analysis is described. The results of numerical simulation of supernova Ia explosions on massive parallel supercomputers by means HydroBox3D code are presented.

1 Introduction

Supernovas are major sources of "life" elements—from carbon to iron. Type Ia supernovas (SNIa) are very bright and, therefore, they are used as "standard candles" to determine distances to galaxies and the expansion rate of the Universe. A major scenario [1] of supernova explosion is based on the merging of two degenerate white dwarfs with subsequent collapse of a new star when it reaches the Chandrasekhar mass, ignition of the carbon burning process, and type Ia supernova explosion. The goal of this paper is to determine the role of the ignition point in nuclear fuel burning and in the dynamics of the remnants of a degenerate dwarf explosion.

Numerical simulations plays a key role in the modern astrophysics. Perhaps, it is the only universal approach to study the nonlinear evolutional processes in the Universe. One of the main problems of astrophysics simulation is the scale ratio. By example, a typical galaxy can have the mass of 10^{13} Solar masses and the size of 10^4 parsecs, resulting in 13 order gap for the mass and 14 order gap for the size in comparison to the Sun. Therefore it is necessary to use best available supercomputers in order to simulate complex astrophysical processes with high resolution.

V. Malyshkin (Ed.): PaCT 2019, LNCS 11657, pp. 187–198, 2019.
https://doi.org/10.1007/978-3-030-25636-4_15

Nine of the top ten supercomputers listed in the 2018 November version of the Top 500 list are equipped with graphic accelerators and Intel Xeon Phi/Sunway accelerators. Most likely, the first ExaScale performance supercomputer will be built based on the hybrid approach. The code development for the hybrid supercomputers is not a solely technical problem, but an individual complex scientific problem, requiring co-design of algorithms during all stages of problem solving – from physical statement to development tools.

The problem of Mind the Gap of reproducing the nuclear front of heavy elements burning thin relatively to the star size, remains even when using top-level supercomputers when solving problems SNIa. One possible solution to such problems is the use of multi-level nested grids. The approach is to use adaptive nested grids to simulate hydrodynamics of the SNIa explosion and the dynamics of residuals. The next level of nesting of grids allows to reproduce the burning front more correctly. Using the resources of SSCC, we were able to partially solve the Mind the Gap problem by reproducing seven orders of magnitude. We hope that regular access to more productive supercomputers will allow us to advance several orders of magnitude. Following is a short review of codes, that allow you to use a high resolution.

AREPO [2]. The code is based on the technology of moving mesh based on Voronoi and Delaunay triangulation with Lloyd's regularization [3]. This approach allows you to adapt the mesh for the solution. In this case, unlike the SPH methods, the method is based on the Eulerian approach. With all the advantages of such an approach, it is rather difficult in terms of computational costs. The question remains about the quality of the solution in the areas described by less detailed grid cells. Nevertheless, the AREPO code is one of the most used in the World at the moment.

BETHE-HYDRO [4]. This code is based on an ALE-approach combining advantages of the Euler and Lagrange approaches. The equations of hydrodynamics are solved on an unstructured grid in nonconservative Lagrangian form. The numerical method is based on an operator approach which makes it possible to construct (and this is done in the present paper) balanced schemes to approximate the gradient and divergence operators. To solve the Poisson equation in one-dimensional statement, the tridiagonal matrix algorithm (or the Thomas method) is used. In two-dimensional statement the Poisson equation is solved by a conjugate gradient method. Then the potential is corrected to conserve the total energy (the sum of the kinetic, internal, and potential energies) of the system. It should be noted that the total energy of the system is not exactly conserved, but the error in the collapse problem is insignificant, about 10^{-2} per cent. Unfortunately the approach has not been extended to the three-dimensional case.

CHOLLA [5]. The software package is designed for GPU computational experiments and is based on a CTU (Corner Transport Upwind) method. The method is used to extend the upwind scheme to the multidimensional case [6,7]. A cell structure containing all hydrodynamic parameters is used to store the calculation

grid on the GPU. Such data locality allows more efficient use of the graphics card global memory. Calculations of a time step are performed on graphic accelerators with the use of CUDA extensions. All numerical methods being used are described in detail in [5].

ENZO [8]. The software package is based on the solution of the equations of magnetic gas dynamics with allowance for cosmological expansion. An N-body model is used to simulate the collisionless component. The code includes a large number of subgrid processes: primordium chemical kinetics, cooling/heating functions, radiation transport, as well as star formation processes and effects resulting from supernova explosions. Several solvers are used to solve the hydrodynamic equations: PPM (implemented only for the equations of gas dynamics), MUSCL, and a finite difference method. An algorithm based on the fast Fourier transform is used to solve the Poisson equation. A so-called structured adaptive grid is also used. Here the basic idea is that the calculation grid has a minimum difference between the neighboring cells. This structure allows using regular trees where a subdomain is divided not more than two times, which increases the efficiency of using such calculation grids.

GADGET2 [9]. The code uses an SPH method as a basic method of solution. At present this is the most widely used code based on the SPH approach. However, the number of codes based on the SPH method decreases, and a major tendency is to use Lagrange–Euler approaches in combination with grid methods. A passage along a Peano–Hilbert curve is used to distribute the particles between the processes. Now it is a standard approach for the parallel implementation of SPH methods.

GAMER [10]. The code contains a solution of the gas dynamics equation using an AMR approach on graphic accelerators. A TVD approach is used to solve the gas dynamics equations, and a combination of a method based on the fast Fourier transform and a method of successive upper relaxation is used to solve the Poisson equation. It seems that a major peculiarity of this complex is the implementation of the AMR approach on graphics cards. In this way a regular structure of the grid is naturally projected onto the GPU architecture, whereas a tree structure needs special approaches. This approach is in using "octets" to define the grid by projecting onto a specific graphics card flow. A major problem here is the formation of fictitious cells for the octet, which takes about 63% of the time. However, this procedure can be performed for each of the octets independently.

GIZMO [11]. For this software code, a new mesh-free approach to solving equations of gravitational gas dynamics has been developed and implemented. The approach is based on a combination of classical grid methods and an SPH method. This method is in using the gas dynamics equations in Euler coordinates which, according to the variational Galerkin principle, are multiplied by test functions. A peculiarity of these functions is that they are linked not to the calculation grid, as in paper [12], but to individual particles [13] which are

similar to SPH particles. To determine the values at the domain boundaries, a solution of the Riemann problem using the MUSCL scheme is used.

RAMSES [14]. The code employs a numerical solution of the gravitational gas dynamics equations using an AMR approach based a division into octets. A combination of a method based on the fast Fourier transform and the Gauss–Seidel method is used to solve the Poisson equation. Simple 5-point finite difference approximation is used to solve the Poisson equation. It was replaced by a more efficient 19-point approximation implemented in the form of an extension of the RAMSES code for the case of nonclassical gravitation (MOND) [15].

In the Sect. 2, we describe the concept of co-design, within which the computational model SNIa was developed. We also briefly summarize the information about numerical methods that was used. The Sect. 3 will be devoted to the parallel implementation of the HydroBox3D code. In the Sect. 4 the results of mathematical modeling of the SNIa noncentral explosion will be presented. The conclusion is given in the Sect. 5.

2 The Co-design of Numerical Model

As mentioned in the introduction, the development of software for supercomputers is a complicated scientific problem and it requires the co-design at all stages of the numerical model creation. We outline six co-design stages of numerical modeling Fig. 1. The main difference between the co-design and the classic design of the computational model is the possibility of returning to the previous development stage with the constraints at the current stage. This makes it possible to build in a short time an effective computational model that takes into account all the developments.

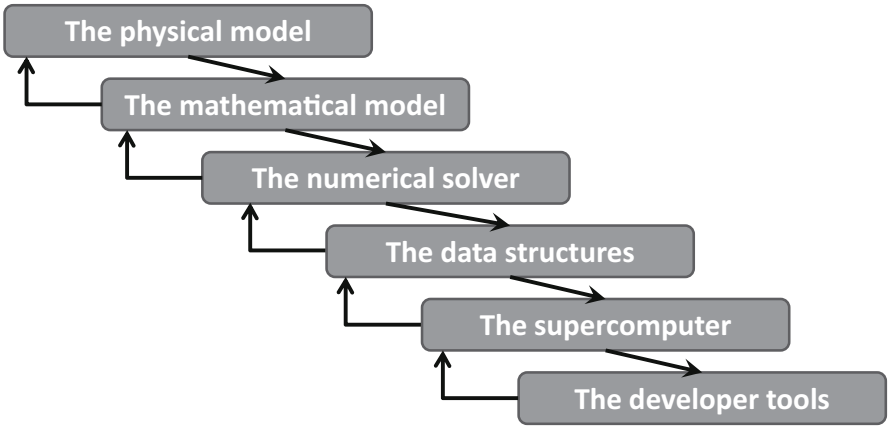

Fig. 1. The co-design conception of astrophysical problem solution method

The problem statement is studying the SNIa explosion during the perturbation of an individual white dwarf, which occurs before the merger of two white dwarfs. In this case, the SNIa explosion occurs at the periphery of the star.

The source of the perturbation is a companion, which is introduced into the physical model by a white dwarf perturbation displaced from the center. For the transition from deflagration to detonation, it is necessary to carefully take into account the combustion front at which nuclear combustion of carbon takes place (we will dwell on it in present study as the most energy efficient source of the explosion). The size of such a front is not resolvable for present day architectures, so we will focus on use of hydrodynamic modeling on multilevel nested grids. Next, we describe the organization of calculations, and then give a briefly description of mathematical model and numerical methods that are used.

2.1 The Parallel & Distributed Computing

The hydrodynamic numerical simulation of SNIa is performed on architecture with shared memory on adaptive nested meshes and is distributed using OpenMP tools within a single process. In our computational experiments we used an Intel Optane node which has 700 GB RAM for a single process. The nuclear reaction hydrodynamics of SNIa is performed on an architecture with distributed memory, with a software implementation based on a one-dimensional geometrical decomposition of a regular calculation domain by MPI tools and subsequent decomposition of the calculations into threads using OpenMP tools within a single process. A diagram of calculations organization is shown in Fig. 2. Regular grids at the second level of adaptive nested mesh are used to calculate hydrodynamic turbulence, which begins with a uniform density distribution corresponding to the cell. For a characteristic time step, one should not expect a

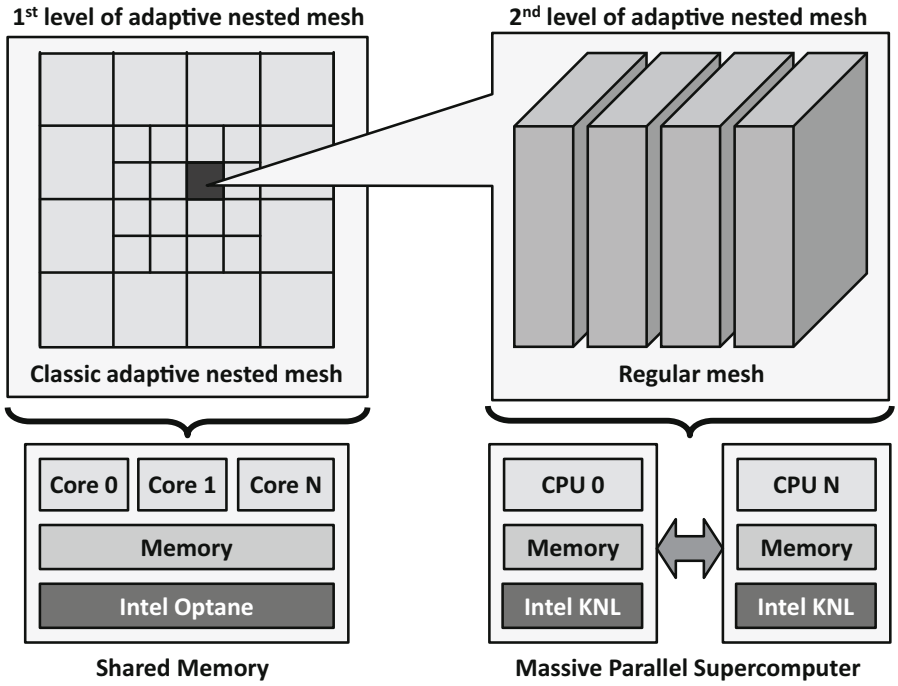

Fig. 2. The organization of parallel and distributed computing in HydroBox3d code

local increase in density by several orders of magnitude. Therefore, the use of regular grids on the second level is fully justified.

2.2 The Numerical Model

Consider the conservative form of the equations of gravitational gas dynamics of conservation of masses

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

conservation of momentum

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}\boldsymbol{u}) = -\nabla p - \rho \nabla \Phi, \tag{2}$$

and conservation of total mechanical energy

$$\frac{\partial}{\partial t}\left[E + \rho \frac{\boldsymbol{u}^2}{2}\right] + \nabla \cdot \left(\left[E + \rho \frac{\boldsymbol{u}^2}{2}\right]\boldsymbol{u}\right) = -\nabla \cdot (p\boldsymbol{u}) - (\rho \nabla \Phi, \boldsymbol{u}) + Q, \tag{3}$$

supplemented by the Poisson equation for the gravitational potential

$$\Delta \Phi = 4\pi G \rho, \tag{4}$$

where ρ is the density, $\boldsymbol{u}$ is the velocity, p is the pressure, Φ is the gravitational potential, E is the internal energy of the gas, G is the gravitational constant, and Q is a source of energy due to nuclear reactions.

The equation of state for stars consists of the pressure of a nondegenerate hot gas and the pressure due to radiation and a degenerate gas [16]. In the case of a degenerate gas, both relativistic and nonrelativistic regimes are considered. The equation of state $p = (\rho, T)$ is sought for as the sum of four components:

$$p = p_{rad} + p_{ion} + p_{deg,nrel} + p_{deg,rel}, \tag{5}$$

where T is the temperature, p_{rad} is the pressure of radiation, p_{ion} is the pressure of a nondegenerate hot gas (ions), $p_{deg,nrel}$ is the pressure of a degenerate nonrelativistic gas, and $p_{deg,rel}$ is the pressure of a degenerate relativistic gas.

As nuclear carbon burning we first consider a nuclear reaction responsible for the bombardment of carbon by carbon yielding natrium and proton $12C\,(12C, p)\,23Na$, where $Q = 2.24$ MeV is the energy released during the nuclear reaction. Assume that the nuclear reaction rate $k_{12C(12C,p)23Na}$ is known from the literature [17].

2.3 The Hydrodynamical Solver

The numerical method to solve the equations of hydrodynamics is based on a combination of Godunov's method for conservation laws by calculating fluxes through the boundaries [18], an operator splitting method to construct a scheme that is invariant with respect to rotation to approximate the advection terms

[19–21], and Rusanov's method to solve Riemann problems [22] for determining the fluxes with vectorization of the calculations [23]. A compact scheme for a piecewise-parabolic representation of the solution in each of the directions is used to solve the Riemann problems [24–26].

To solve the hydrodynamic equations, a modification of an original numerical method based on a combination of an operator splitting method, Godunov's method, and a Rusanov-type scheme is used. This method has all advantages of the above methods and a high degree of parallelization. The numerical scheme is considered in detail in paper [23]. The main idea of the method is in writing the equations of hydrodynamics in vector form:

$$\frac{\partial v}{\partial t} + \bigtriangledown \cdot f(v) = 0, \tag{6}$$

where v is the vector of conservative variables. For Eq. (6) we use the following numerical scheme in one of the directions:

$$\frac{v_i^{n+1} - v_i^n}{\tau} + \frac{F_{i+1/2} - F_{i-1/2}}{h} = 0, \tag{7}$$

where F is the solution to a Riemann problem. Omitting the details of derivation of the numerical scheme, which is based on adjoint equations and an operator splitting method, we have the final form of the solution to the Riemann problem:

$$F = \frac{f(v_L) + f(v_R)}{2} + \frac{c + \|\boldsymbol{u}\|}{2} (v_L - v_R). \tag{8}$$

To determine the quantities $f(v_L)$, $f(v_R)$, v_L, and v_R, we use a piecewise-parabolic representation of the solution. The equations of hydrodynamics for the quantities will be calculated in the cells of the root and nested meshes. The Poisson equation for the root mesh will also be calculated in the cells. Then the solution will be projected onto the boundary nodes of the nested mesh. To solve the Poisson equation on the nested mesh the quantities of the potential (and density) will be arranged at the nodes of the nested mesh.

The equations of hydrodynamics (Riemann problems) are solved in two steps: (1) solving the Riemann problems on all boundaries of the nested mesh, and (2) solving Riemann problems at all internal interfaces of the nested mesh. Whereas the second part of solving the Riemann problems is rather trivial, in the first part the method of calculation depends on the sizes of cells of the two neighboring nested meshes. If the cell sizes are equal, the solution to the Riemann problem is the same as that of the Riemann problems at the internal interfaces of the nested mesh, and it is trivial. If a cell of the neighboring nested mesh is larger than the cell being considered the Riemann problem is solved at the interface between the reduced neighboring cell. If the cell being considered has a common boundary with several cells of the neighboring nested mesh the Riemann problems are solved at all interfaces, and then the fluxes are averaged [27]. To organize the satellite calculations, a regular mesh is used this is equivalent to using a root mesh.

2.4 The Poisson Solver

To solve the Poisson equation we use a combination of method based on the fast Fourier transform (for the root mesh) and method of successive over-relaxation (for nested meshes). The Poisson equation is solved in two steps:

1. Solve the Poisson equation on the root mesh by the fast Fourier transform.
2. Solve the Poisson equation on the nested mesh by the method of successive over-relaxation.

We will not consider the method at the first step of solving the Poisson equation (a detailed description of the method to solve it can be found in paper [26]), which is also used to solve the Poisson equation in the satellite calculations.

The method of successive over-relaxation (SOR) is an iterative process of finding the potential on a nested mesh with given initial and boundary conditions obtained by solving the Poisson equation on the root mesh. A similar approach to solve the Poisson equation has been proved to be efficient is some program codes, for instance, in the GAMER code [10].

3 The Performance Analysis

As noted above, the hydrodynamics numerical simulation of SNIa is made on architecture with shared memory. Therefore, we consider a parallel implementation of the second level of nested meshes based on domain decomposition [21]. The MPI tools are used to perform a one-dimensional geometrical decomposition of the calculation domain. In the case of Intel Xeon Phi processors the OpenMP tools are employed. When using Intel Xeon Phi (KNL) processors the calculations are vectorized with some low-level tools [23,28].

The speedup of the code on a mesh of size 512^3 has been studied. For this, the total numerical method time was measured in seconds at various numbers of threads. The speedup P was calculated as

$$P = \frac{Total_1}{Total_K},$$

where $Total_1$ is the calculation time using one thread, and $Total_K$ is the calculation time on K threads. The actual performance has also been estimated. The results of these investigations of the speedup and performance on the mesh of size 512^3 are shown in Fig. 3. A performance of 173 gigaflops and a 48x speedup are obtained on a single Intel Xeon Phi processor.

The scalability of the code on calculation grid size of a $512p \times 512 \times 512$ was studied using all threads for each of the processors, where p is the number of processors being used. Thus, a subdomain of size of 512^3 was used for each processor. To study the scalability, the total numerical method time was measured in seconds at various numbers of Intel Xeon Phi (KNL) processors. The scalability T was calculated as

$$T = \frac{Total_1}{Total_p},$$

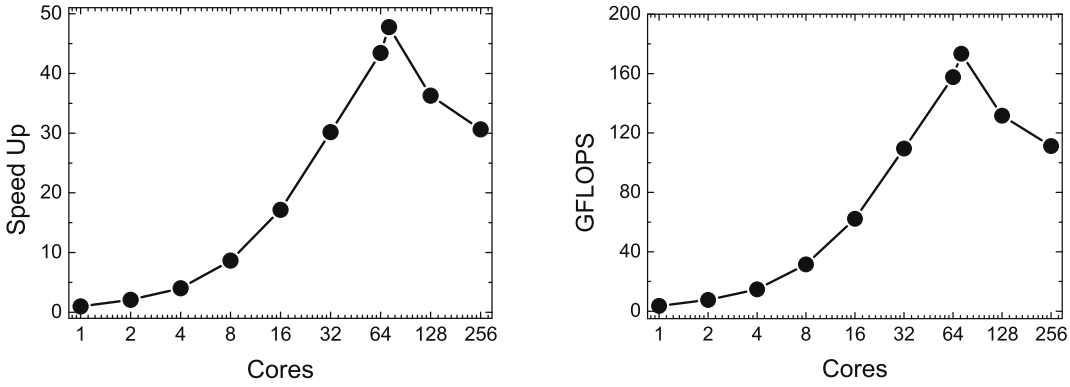

Fig. 3. Speedup and performance of the code on Intel Xeon Phi

where $Total_1$ is the calculation time with the use of one processor, and $Total_p$ is the calculation time with the use of p processors. The results of these investigations of the scalability are shown in Fig. 4. A 97% scalability is reached with 16 processors, which is a rather good result.

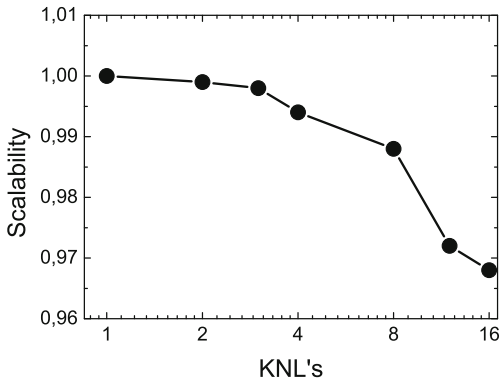

Fig. 4. Scalability of the code

4 The Numerical Simulation

Let's perform simulation of white dwarf with one solar mass and temperature $T = 10^9$ K and a normal distribution of the velocities with a variance of ten percent of the sound speed in the central part of the star. Fig. (5) shows the simulation results: density dynamics from the onset of the explosion to its passage through the bulk of the star. One can see from the simulation results (Fig. 5) that a periphery ignition of the white dwarf takes place when the critical densities for the onset of detonation carbon burning are achieved. As a limiting density for the onset of the process of carbon burning, we use the density of transition from deflagration to detonation from paper [29], which is $\rho_{DDT} = 10^{7.2}$ g cm^{-3}. From the distributed computing it is clear that carbon burning approx 80% complete. This statistics was used to simulate noncentral explosions. However, only the hydrodynamics can show the dynamics of real carbon burning.

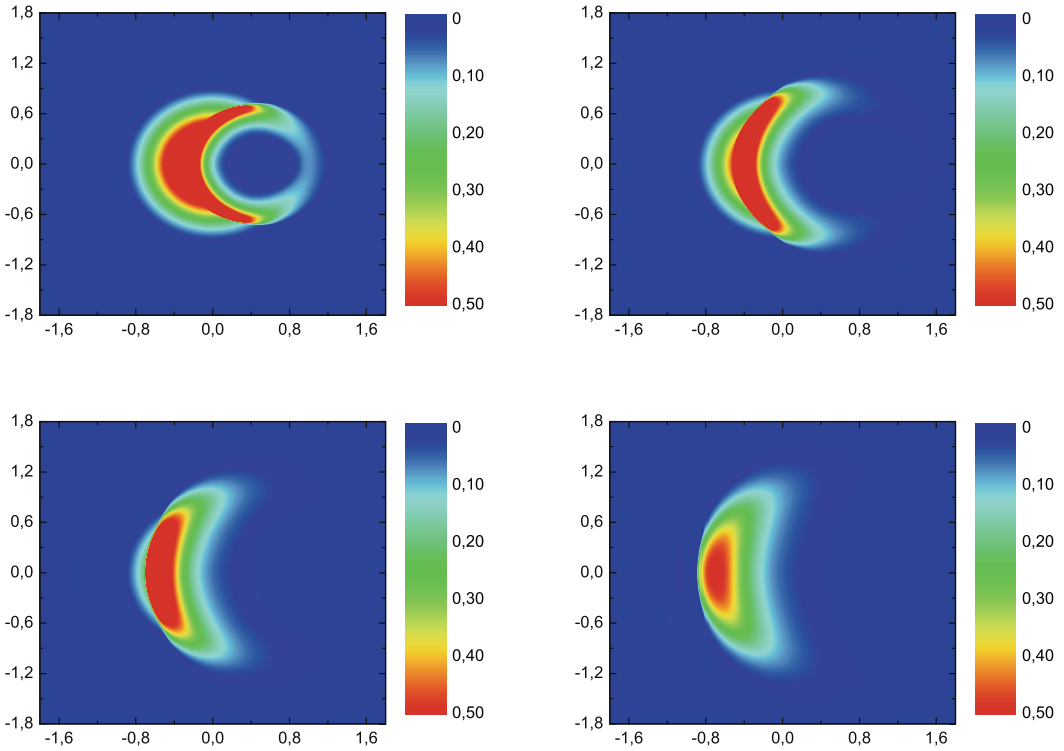

Fig. 5. Relative density distribution from the onset of the explosion to its passage through the bulk of the star

5 Conclusion

The new parallel & distributed hydrodynamical code HydroBox3D for numerical simulation of supernovae Ia type explosion was described in the paper. The HydroBox3D code is developed on the basis of combination of adaptive nested mesh for hydrodynamical simulation of supernovae explosion and regular mesh that is a second level of nested mesh for hydrodynamical simulation of nuclear reaction. A performance of 173 gigaflops and a 48x speedup are obtained on single Intel Xeon Phi processor. A 97% scalability is achieved on 16 processors. Results of numerical simulation of supernova Ia explosions on massive parallel supercomputers obtained with help of the HydroBox3D code are presented.

We developed the HydroBox3D code for a specific problem of supernova of Ia type. Requirements for describing the process of carbon nuclear burning are also was initiated by the features of the problem. However, as the result the technology for solving problems of different-scale gravitational hydrodynamics was developed. So staying within the framework of the implemented hydrodynamic model, we can perform simulation of the star formation process in the interstellar medium in the problems of galaxies collisions and evolution. Also we can perform simulation of the explosion hydrodynamics of supernovae of type II with explosion source – core-collapse, as well as model all the hierarchy of cosmological modeling "observed Universe – cosmic web – clusters of galaxies – and galaxies interaction". The code extension for that hyperbolic models, such as magnetic hydrodynamics, relativistic hydrodynamics and collisionless fluid

dynamics allows one to use program code like a technology to solve a wide class of astrophysics problems. In the future, we plan to use the developed technology for actual problems of astrophysics.

Acknowledgements. The research work was supported by the Grant of the Russian Science Foundation (project 18-11-00044).

References

1. Iben, I., Tutukov, A.: On the evolution of close triple stars that produce type Ia supernovae. Astrophys. J. **511**(1), 324–334 (1999)
2. Springel, V.: E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. Mon. Not. Royal Astron. Soc. **401**, 791–851 (2010)
3. Lloyd, S.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
4. Murphy, J., Burrows, A.: BETHE-hydro: an arbitrary Lagrangian-Eulerian multidimensional hydrodynamics code for astrophysical simulations. Astrophys. J. Suppl. Ser. **179**, 209–241 (2008)
5. Schneider, E., Robertson, B.: Cholla: a new massively parallel hydrodynamics code for astrophysical simulation. Astrophys. J. Suppl. Ser. **217**(2), 24 (2015)
6. Collela, P.: Multidimensional upwind methods for hyperbolic conservation laws. J. Comput. Phys. **87**, 171–200 (1990)
7. Gardiner, T., Stone, J.: An unsplit Godunov method for ideal MHD via constrained transport in three dimensions. J. Comput. Phys. **227**, 4123–4141 (2008)
8. Bryan, G., et al.: ENZO: an adaptive mesh refinement code for astrophysics. Astrophys. J. Suppl. Ser. **211**(2), 19 (2014)
9. Springel, V.: The cosmological simulation code GADGET-2. Mon. Not. Royal Astron. Soc. **364**, 1105–1134 (2005)
10. Schive, H., Tsai, Y., Chiueh, T.: GAMER: a GPU-accelerated adaptive-mesh-refinement code for astrophysics. Astrophys. J. **186**, 457–484 (2010)
11. Hopkins, P.: A new class of accurate, mesh-free hydrodynamic simulation methods. Mon. Not. Royal Astron. Soc. **450**(1), 53–110 (2015)
12. Mocz, P., Vogelsberger, M., Sijacki, D., Pakmor, R., Hernquist, L.: A discontinuous Galerkin method for solving the fluid and magnetohydrodynamic equations in astrophysical simulations. Mon. Not. Royal Astron. Soc. **437**(1), 397–414 (2014)
13. Gaburov, E., Nitadori, K.: Astrophysical weighted particle magnetohydrodynamics. Mon. Not. Royal Astron. Soc. **414**(1), 129–154 (2011)
14. Teyssier, R.: Cosmological hydrodynamics with adaptive mesh refinement. A new high resolution code called RAMSES. Astron. Astrophys. **385**, 337–364 (2002)
15. Candlish, G., Smith, R., Fellhauer, M.: RAyMOND: an N-body and hydrodynamics code for MOND. Mon. Not. Royal Astron. Soc. **446**(1), 1060–1070 (2015)
16. Timmes, F.X., Arnett, D.: The accuracy, consistency, and speed of five equations of state for stellar hydrodynamics. Astrophys. J. Suppl. Ser. **125**, 277–294 (1999)
17. Spillane, T., et al.: $^{12}C + {}^{12}C$ fusion reactions near the Gamow energy. Phys. Rev. Lett. **98**, 122501 (2007)
18. Godunov, S., Kulikov, I.: Computation of discontinuous solutions of fluid dynamics equations with entropy nondecrease guarantee. Comput. Math. Math. Phys. **54**, 1012–1024 (2014)

19. Kulikov, I., Chernykh, I., Snytnikov, A., Protasov, V., Tutukov, A., Glinsky, B.: Numerical modelling of astrophysical flow on hybrid architecture supercomputers. In: Tarkov, M. (ed.) Parallel Programming: Practical Aspects, Models and Current Limitations, pp. 71–116 (2014)
20. Vshivkov, V., Lazareva, G., Snytnikov, A., Kulikov, I., Tutukov, A.: Computational methods for ill-posed problems of gravitational gasodynamics. J. Inverse Ill Posed Probl. **19**(1), 151–166 (2011)
21. Kulikov, I., Lazareva, G., Snytnikov, A., Vshivkov, V.: Supercomputer simulation of an astrophysical object collapse by the fluids-in-cell method. In: Malyshkin, V. (ed.) PaCT 2009. LNCS, vol. 5698, pp. 414–422. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03275-2_41
22. Rusanov, V.V.: The calculation of the interaction of non-stationary shock waves with barriers. Comput. Math. Math. Phys. **1**, 267–279 (1961)
23. Kulikov, I.M., Chernykh, I.G., Tutukov, A.V.: A new parallel intel xeon phi hydrodynamics code for massively parallel supercomputers. Lobachevskii J. Math. **39**(9), 1207–1216 (2018)
24. Popov, M., Ustyugov, S.: Piecewise parabolic method on local stencil for gasdynamic simulations. Comput. Math. Math. Phys. **47**(12), 1970–1989 (2007)
25. Popov, M., Ustyugov, S.: Piecewise parabolic method on a local stencil for ideal magnetohydrodynamics. Comput. Math. Math. Phys. **48**(3), 477–499 (2008)
26. Kulikov, I., Vorobyov, E.: Using the PPML approach for constructing a low-dissipation, operator-splitting scheme for numerical simulations of hydrodynamic flows. J. Comput. Phys. **317**, 318–346 (2016)
27. Kulikov, I.: The numerical modeling of the collapse of molecular cloud on adaptive nested mesh. J. Phys. Conf. Ser. **1103**, 012011 (2018)
28. Kulikov, I.M., Chernykh, I.G., Glinskiy, B.M., Protasov, V.A.: An efficient optimization of Hll method for the second generation of Intel Xeon Phi processor. Lobachevskii J. Math. **39**(4), 543–551 (2018)
29. Willcox, D., Townsley, D., Calder, A., Denissenkov, P., Herwig, F.: Type Ia supernova explosions from hybrid carbon - oxygen - neon white dwarf progenitors. Astrophys. J. **832**(1), 13 (2016)