The grid codes generation for solving problems of the cosmic plasma hydrodynamics on supercomputers

Anna Sapetina¹, Ivan Ulyanichev² and Boris Glinskiy¹

¹ Institute of Computational Mathematics and Mathematical Geophysics SB RAS, 6 Akad. Lavrentieva pr., Novosibirsk, Russia
² Novosibirsk State Technical University, 20 K. Marksa pr., Novosibirsk, Russia

E-mail: afsapetina@gmail.com

Abstract. In this paper an attempt is made to create a system for generating the parallel codes to simulate some astrophysical phenomena on modern computing architectures. A code construction scheme for the numerical solution of various problems of the cosmic plasma hydrodynamics is proposed using a knowledge base with an ontological representation of a given mathematical simulation area. The ontology of numerical methods and parallel algorithms and the ontology of parallel architectures and technologies together with the expert (inference) rules allow selecting an efficient numerical method, a parallel algorithm and a computational architecture for solving a necessary problem from the outlined domain. The corresponding program units that implement various stages of solving the problem on a chosen computational architecture are substituted into scheme of solving the problem constructed in this way. Program units can be based on both the author’s implementation and on calling various parallel libraries.

1. Introduction

It is known that more than 90% of the visible matter in the universe consists of plasma, so the plasma-astrophysics discipline has numerous fields of application. For example, plasma is a component of stars, nebulae, interplanetary, interstellar and intergalactic media. In the formation and dynamics of astrophysical objects, the magnetic field can fundamental affect the plasma dynamics.

The study of many astrophysical problems (for example, the formation of jets and magnetized disks around the newborn black holes, etc [1]) requires the investigation of the hydrodynamics of magnetized liquids.

An crucial part in studying of astrophysical processes is played by mathematical modeling, for which a large number of parallel codes have been developed. Among them, we can distinguish the codes based on the SPH method [2]-[4], the grid codes [5]-[7], codes using the adaptive [8]-[10] and the moving [11]-[13] meshes. Most of them are focused on the classical supercomputer architectures usage, but there are also representatives focused on graphics accelerators [14]-[16] and Intel Xeon Phi accelerators [17].

Most of the existing codes for simulating the cosmic plasma hydrodynamics are aimed at solving certain types of problems, but there are also universal codes [18]-[21] with a modular structure that support the variability of the problems being solved and the methods used. For example, the MPI-AMRVAC code has a certain versatility for solving the problems of stellar
evolution [22], but, nevertheless, it has a certain restrictions on solving the problems of galactic evolution or the problems of supernova explosions.

There are also attempts to create systems for generating astrophysical codes. For example, a code generation system using the EXCALC package has been created at the University of Costa Rica. This system transforms MHD equations to the Cartesian coordinates, followed by discretizing the equations by a method selected [23]. Similar works are carried out in Russia. Thus, the Institute of Astronomy of RAS, under the leadership of D. V. Bisikalo, Corresponding member of RAS, has created a code generation system for the evolution of the close binary stars [NSCF, 2013]. Still, all these systems are not completely universal.

This paper proposes a concept for creating a system of generating the grid parallel codes for the numerical solution of various hydrodynamic problems of the cosmic plasma based on a generalized representation of the magnetic hydrodynamics equations using the Godunov method. Using the ontological knowledge of the area in question and the modular structure of the code make it easier to prepare the necessary blocks for its generation.

2. A generalized description of the problem statement and solution method

The basic equations of the gravitational magnetic hydrodynamics for the vector of conservative variables $U$ have the following form:

$$\frac{\partial U}{\partial t} + \nabla \cdot F(U) = Q,$$

(1)

$$\Delta \Phi = P(U),$$

(2)

where $F(U)$ is the flow vector of conservative variables, $Q$ is the right-hand side vector (a source), $\Phi$ is the gravitational potential, $P(U)$ is the gravity source.

The operator splitting approach allows us to represent this system as a combination of a hyperbolic system $\frac{\partial U}{\partial t} + \nabla \cdot F(U) = 0$, an elliptic system $\Delta \Phi = P(U)$ and a right-hand side $\frac{\partial U}{\partial t} = Q$. Equation (1) can also include a parabolic term describing, for example, the heat conduction effect, but at this stage of our study this was not considered.

For the numerical solution of system (1-2), various discretization methods can be used. In this paper, we consider a finite difference approach, which is more suitable for modeling media without significant curvilinear inclusions, in contrast to the finite volume method or finite element method. To discretize the computational domain, we introduce a uniform grid with a step $h$, at which the conservative variables are determined at the cell centers using fractional indices, and the conservative variables flows – at the centers of the interfaces between the cells. Then the Godunov scheme for equation (1) in the general form can be written down as follows:

$$U^{n+1}_{i+\frac{1}{2}, j, k} = U^n_{i+\frac{1}{2}, j, k} + \tau \left( F_{x}^n_{i+1, j, k} - F_{x}^n_{i, j, k} + \frac{F_{x}^n_{i+\frac{1}{2}, j, k+\frac{1}{2}} - F_{x}^n_{i+\frac{1}{2}, j, k-\frac{1}{2}}}{h} + \frac{F_{y}^n_{i, j+1, k} - F_{y}^n_{i, j, k}}{h} \right) + \frac{F_{z}^n_{i, j, k+\frac{1}{2}} - F_{z}^n_{i, j, k-\frac{1}{2}}}{h} = Q^n_{i+\frac{1}{2}, j, k},$$

where $\tau$ is the time step, calculated from the CFL condition, $F_{x}, F_{y}, F_{z}$ are the conservative variables flows in terms of $x, y, z$ coordinates.

To implement such a computational scheme, it is necessary to find the flow values at the interfaces between the cells, that is, to solve the Riemann problem. For this, various solvers can be applied, including high-order ones with different ways of representing a solution, for example, such as: MUSCL schemes, piecewise-linear and parabolic approximations, WENO schemes, HLL method, etc.

To numerically solve the Poisson equation (2), one can use any known grid method for the elliptic systems, for example, using the fast Fourier transform, various iterative methods, etc.
Such above-mentioned generalized description of the equations system and the numerical method for solving problems of magnetohydrodynamics allows us to compose a unified upper algorithm level suitable for solving a wide class of astrophysical problems and to propose a code generation scheme based on it.

3. A code generation scheme

Let us describe in greater detail the proposed structure of the generated code and the necessary modules from which it is assembled.

The general code organization scheme for the numerical solution of equations of the forms of (1-2) is shown in Figure 1. The first step is to initialize the necessary grid arrays for a selected data structure, which is determined by the specificity of solving equations and the effect on the parallelization efficiency. The second step is to upload the information about a medium under consideration. Further, in the loop, while the time condition is fulfilled, the following actions are sequentially performed: recalculating the current time step to fulfill the CFL condition, fulfilling boundary conditions for primitive variables, reconstructing primitive variables based on the selected local template, fulfilling boundary conditions for reconstructed variables, starting the non-blocking exchange of boundary values between the adjacent nodes (parallelization by decomposition of the computational domain), solving the Riemann problem, computing the solution to equation (1) according to the Godunov scheme, solving the Poisson equation (2), verification of the exchanges completion, saving the solution at the current time step. Figure 1 also lists the necessary modules for assembling the final code, which, together with the universal function main, based on the described scheme, compound the finished version of the code (MPI-program). Let us note that as specific modules one can use both the library solutions and author's implementations. Moreover, specific solving equations determine only a set of conservative and primitive variables; the general solution scheme remains universal. The choice of specific methods to solving the arising sub problem and the corresponding modules is up to the researchers. Therewith, modules that implement solutions to the Riemann problem and calculations according to the Godunov scheme assume a parallel implementation, for example, using OpenMP technology and vector extensions for existing Intel processors and co-processors or using the CUDA core-function on the threads grid for calculations on graphic accelerators. The main structure can be expanded in the case of using the operator splitting approach or adding the calculation of chemical reactions, adaptive grids. However in any case the main framework and all the above modules will remain the same.

4. Intelligent support ontology for solving compute-intensive astrophysics problems

The development of efficiently parallel codes for modern supercomputer systems requires the knowledge about relevant computational methods, parallel architectures and technologies. To simplify the problem of choosing optimal numerical methods and suitable target architectures, it is proposed to present the accumulated knowledge about solving compute-intensive problems of astrophysics in an obvious ontological form. The ontology of mathematical methods and parallel algorithms and the ontology of parallel architectures and technologies, tied together with the rules of inference, allow one to build an optimal scheme for solving the problem. On this basis, the necessary modules for generating a code are determined according to the above described scheme. Such an approach can essentially simplify the process of developing scientific software and make it accessible to a researcher who is poorly skilled in some blocks of knowledge represented in ontology.

Figure 2 shows the top level of the compiled ontology, some basic concepts of which are supplemented by a finite set of objects. The solution to a problem begins with the determination of a Physical Phenomenon under study described by the corresponding Equation system.
Figure 1. The general scheme of code organization and the necessary modules for the numerical solution of MHD-equations.

The Equation System is solved by a Computational Method constructed using the previously mentioned methods. The final Computational Method, in turn, is implemented by a Parallel Algorithm, which is encoded by program Code optimized for a selected Computing Device using assigned Parallel Programming Technologies. In the future, the ontology will be expanded by expert (inference) rules containing the information not explicitly presented in the ontology about the properties of mapping computational methods to computational architectures. The rules will allow one to come to a comparative assessment of the computation time, the size of the memory used, the accuracy of the solution, scalability for a selected bundle: the computational method – the target architecture.

5. Conclusion

This paper presents the code generation concept for solving the problems of cosmic plasma simulation on parallel computing systems using an ontological representation. The generated code structure and the main modules are described, as well as the top-level ontology of the intelligent support for solving compute-intensive problems of astrophysics.

The proposed generation system maintains multipurposeness with respect to the conservation laws of magnetic hydrodynamics and numerical methods in question. The generated code allows one to simulate the dynamics of cosmic plasma for a wide range of problems.
Figure 2. The intelligent support ontology for solving compute-intensive astrophysics problems.

Acknowledgments
This research was conducted within the framework of the budget project No. 0315-2019-0009, for ICMMG SB RAS, and supported in part by the Russian Foundation for Basic Research [grant No. 19-07-00085].

References
[1] Tajima T and Shibata K 2002 Plasma Astrophysics (Boulder:Westview Press)
[2] Volker S 2005 Mon. Not. R. Astron. Soc. 364 1105–34
[3] Wadsley J W, Stadel J and Quinn T 2004 New Astronomy 9 137–58
[4] Steinmetz M 1996 Mon. Not. R. Astron. Soc. 278 1005–17
[5] The Pencil code, http://pencil-code.nordita.org/references.php
[6] Stone J M and Norman M L 1992 ApJS 80 753–90
[7] Stone J M, Gardiner T A, Teuben P, Hawley J F and Simon J B 2008 ApJS 178 137–177
[8] Mignone A, Bodo G, Massaglia S, Matsakos T, Tesileanu O, Zanni C and Ferrari A 2007 ApJS 170 228–42
[9] Bryan G L, Norman M L, OShea B W, at al 2014 ApJS 211 19
[10] Teyssier R 2002 A&A 385 337–64
[11] Hopkins P F 2015 Mon. Not. R. Astron. Soc. 450 53–110
[12] Springel V 2010 Mon. Not. R. Astron. Soc. 401 791–851
[13] Murphy J W and Burrows A 2008 ApJS 179 209–41
[14] Hsi-Yu Schive, Yu-Chih Tsai and Tzihong Chiueh 2010 ApJS 186 457–84
[15] Kulikov I 2014 ApJS 214 12
[16] Schneider E E and Robertson B E 2015 ApJS 217 24
[17] Kulikov I M, Chernykh I G, Snytnikov A V, Glinskiy B M and Tutukov A V 2015 CPC 186 71–80
[18] The Einstein Toolkit, https://einsteintoolkit.org/index.html
[19] The MPI-AMRVAC code, http://amrvac.org/
[20] The Athena++ radiation GRMHD code, https://princetonuniversity.github.io/athena/
[21] The PLUTO Code, http://plutocode.ph.unito.it/
[22] Goedbloed J, Keppens R and Poedts S 2003 Space Science Reviews 107 63
[23] Frutos-Alfaro F and Carboni-Mendez R 2016 Rev. Mat 23 41–61