

Article

# A New Parallel Code Based on a Simple Piecewise Parabolic Method for Numerical Modeling of Colliding Flows in Relativistic Hydrodynamics

Igor Kulikov <sup>\*</sup>, Igor Chernykh, Dmitry Karavaev, Vladimir Prigarin, Anna Sapetina, Ivan Ulyanichev and Oleg Zavyalov

Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk 630090, Russia; chernykh@parbz.sccc.ru (I.C.); kda@opg.sccc.ru (D.K.); vovkaprigarin@sccc.ru (V.P.); sapetina@sccc.ru (A.S.); wmzonacomvn@sccc.ru (I.U.); o.romanowi4@sccc.ru (O.Z.)

\* Correspondence: kulikov@ssd.sccc.ru

**Abstract:** A new parallel code based on models of special relativistic hydrodynamics is presented for describing interacting flows. A new highly accurate numerical method is considered and verified. A parallel implementation of the method by means of Coarray Fortran technology and its efficiency are described in detail. The code scalability is 92% on a cluster with Intel Xeon 6248R NKS-1P with 192 Coarray Fortran images. Different interacting relativistic flows are considered as astrophysical applications.

**Keywords:** computational astrophysics; relativistic hydrodynamics; high performance computing

**MSC:** 68W10



**Citation:** Kulikov, I.; Chernykh, I.; Karavaev, D.; Prigarin, V.; Sapetina, A.; Ulyanichev, I.; Zavyalov, O. A New Parallel Code Based on a Simple Piecewise Parabolic Method for Numerical Modeling of Colliding Flows in Relativistic Hydrodynamics. *Mathematics* **2022**, *10*, 1865. <https://doi.org/10.3390/math10111865>

Academic Editors: Leonid B. Sokolinsky and Mikhail Zymbler

Received: 4 May 2022  
Accepted: 26 May 2022  
Published: 30 May 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Many astrophysical phenomena proceed with relativistic speeds. Examples are relativistic jets in active galactic nuclei [1,2] or Herbig–Haro objects [3], collapse of massive stars [4], black holes [5–7], neutron stars [8–10], and gravitational waves [11,12]. Of particular interest are interacting relativistic flows of matter [13–15]. Such processes are observed well with modern instruments [16–19], and observations are used to construct models [20] and perform laboratory experiments [21,22]. Important tools for research are provided by mathematical modeling with supercomputers [23].

Godunov's method, which was developed almost 70 years ago [24], is used in most software codes to simulate hydrodynamic (in particular, relativistic) flows. It has been studied in detail for convergence [25] and the behavior of entropy [26,27]. It has been modified to take into account the transfer of matter [28] and extended to solve equations of relativistic hydrodynamics [29], in particular for magnetic field flows [30,31]. Almost immediately after the invention of Godunov's method, works began to appear on simplified solutions of the Riemann problem (for instance, Rusanov's scheme [32]) and decreased dissipation methods (for instance, Kolgan's scheme [33]). In 2000, a scheme called MUSCL was proposed [34]. It combines the advantages of the Rusanov and Kolgan schemes. The next step in decreasing the dissipation of Godunov-type numerical methods was the use of piecewise parabolic [35,36] and piecewise cubic [37] presentations of solutions. A piecewise cubic presentation of the solution was extended by the authors to HLL-type methods [38] and an operator separation method [39]. The separation of operators and the use of a combination of Godunov- and Rusanov-type schemes made it possible to formulate efficient numerical methods [40,41] and solve the problem of sound point in Godunov's method with linearized discontinuity breakdown [42]. There is a tendency towards increasing the order of the polynomial for presenting the physical variables to be

reconstructed in Godunov's scheme [43]). Nevertheless, there are many original works on using Kolgan's approach to decrease the dissipation in solving the Riemann problem [44,45]. The present paper is essentially an extension of paper [42], using a new numerical method for solving the equations of special relativistic hydrodynamics and adding the degenerate Rusanov method with velocities limited to the speed of sound [46]. This approach will allow simulating relativistic flows in highly rarefied zones, which are often present in relativistic jets [47].

One of the major advantages of the MPI 3.0 standard is an efficient implementation of one-way communications, which allows using one of the most promising parallel programming models, called Partitioned Global Address Space (PGAS). Computational experiments with continuum mechanics models have shown [48–50] that the code obtained by the authors using Coarray Fortran is as efficient as the MPI code. At the same time, this code is much less complicated, which allows the development of new libraries [51] and language extensions [52]. Note that the multidimensional decomposition of computations in the code has practically no effect on its efficiency [53,54].

The limitations of existing solutions in terms of reproducing individual relativistic gas flows, which are primarily associated with the numerical technique used, as well as the lack of codes that use all the advantages of one-way network communications implemented in MPI 3.0 and especially in Coarray Fortran, show us the need to develop a new numerical code for modeling relativistic gas flows. The implementation priority is Coarray Fortran for us, since its use significantly simplifies the code in terms of parallel implementation, which will simplify its implementation. Section 2 provides a description of a new numerical method for solving the equations of special relativistic hydrodynamics. Section 3 is devoted to a parallel implementation of the method. In Section 4, the numerical method is verified on standard tests. Section 5 presents two astrophysical applications. Section 6 provides conclusions to the paper.

## 2. Numerical Method

Let us first introduce the physical variables:  $\rho$ —density,  $\vec{v}$ —velocity vector, and  $p$ —pressure. For definiteness, we take the speed of light  $c \equiv 1$ . In this case the Lorentz factor  $\Gamma$  is defined by the formula

$$\Gamma = \frac{1}{\sqrt{1 - (v/c)^2}} = \frac{1}{\sqrt{1 - v^2}}. \quad (1)$$

The condition imposed on the speed of light ensures that the velocity modulus is limited by unity. This fact will be used in the construction of the numerical scheme. In the code, we will use an ideal gas model whose equation of state is written in the form of an equation for special enthalpy  $h$ :

$$h = 1 + \frac{\gamma}{\gamma - 1} \frac{p}{\rho}, \quad (2)$$

where  $\gamma$  is the adiabatic index. The speed of sound  $c_s$  is defined by the formula

$$c_s^2 = \frac{\gamma p}{\rho h}. \quad (3)$$

To write the equations of special relativistic hydrodynamics, we introduce the following conservative variables: relativistic density  $D = \Gamma\rho$ , relativistic momentum  $M_j = \Gamma^2\rho hv_j$ , where  $v_j$  are components of the velocity vector  $\vec{v}$  at  $j = x, y, z$ , and total relativistic energy  $E = \Gamma^2\rho h - p$ . We write the system of equations in vector conservative form from classic book [55]:

$$\frac{\partial}{\partial t} \begin{pmatrix} \Gamma\rho \\ \Gamma^2\rho hv_j \\ \Gamma^2\rho h - p \end{pmatrix} + \sum_{k=1}^3 \frac{\partial}{\partial x_k} \begin{pmatrix} \rho\Gamma v_k \\ \rho h\Gamma^2 v_j v_k + p\delta_{jk} \\ (\Gamma^2\rho h - p)v_k + pv_k \end{pmatrix} = 0, \tag{4}$$

where  $\delta_{jk}$  is the Kronecker symbol. The conservative variables are easily calculated from the physical variables, but for the inverse transformation the following nonlinear equation has to be solved:

$$f(p) = \Gamma^2\rho h - p - E = 0, \tag{5}$$

For this, Newton’s iterative method is used:

$$p_{m+1} = p_m - \frac{f(p_m)}{f'(p_m)}, \tag{6}$$

where the derivative of the function  $f$  is

$$f'(p) = \frac{\gamma}{\gamma - 1} \Gamma^2 - \frac{M^2 \Gamma^3}{(E + p)^3} \left( D + 2 \frac{\gamma}{\gamma - 1} p \Gamma \right) - 1. \tag{7}$$

When a given accuracy is reached, the iterative process is terminated. The value from the previous time-step is used as the initial pressure approximation.

The system of Equation (4) can be rewritten in vector form as follows:

$$\frac{\partial U}{\partial t} + \sum_{k=1}^3 \frac{\partial F_k}{\partial x_k} = 0, \tag{8}$$

Then for any cell, Godunov’s scheme can be written as

$$\begin{aligned} & \frac{U_{i+\frac{1}{2},k+\frac{1}{2},l+\frac{1}{2}}^{n+1} - U_{i+\frac{1}{2},k+\frac{1}{2},l+\frac{1}{2}}^n}{\tau} + \frac{F_{x,i+1,k+\frac{1}{2},l+\frac{1}{2}}^* - F_{x,i,k+\frac{1}{2},l+\frac{1}{2}}^*}{\Delta_x} + \\ & + \frac{F_{y,i+\frac{1}{2},k+1,l+\frac{1}{2}}^* - F_{y,i+\frac{1}{2},k,l+\frac{1}{2}}^*}{\Delta_y} + \frac{F_{z,i+\frac{1}{2},k+\frac{1}{2},l+1}^* - F_{z,i+\frac{1}{2},k+\frac{1}{2},l}^*}{\Delta_z} = 0, \end{aligned} \tag{9}$$

where  $\Delta_{x,y,z}$  are the grid spacings along the corresponding coordinates,  $\tau$  is a time-step, and  $F^*$  are the fluxes of the corresponding quantities through the cell boundaries obtained by solving the Riemann problem. Let us describe a method for constructing a numerical scheme for solving the Riemann problem.

To construct the numerical scheme, we consider the one-dimensional equations:

$$\frac{\partial}{\partial t} \begin{pmatrix} \Gamma\rho \\ \Gamma^2\rho hv_x \\ \Gamma^2\rho hv_y \\ \Gamma^2\rho hv_z \\ \Gamma^2\rho h - p \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho\Gamma v_x \\ \rho h\Gamma^2 v_x v_x + p \\ \rho h\Gamma^2 v_y v_x \\ \rho h\Gamma^2 v_z v_x \\ (\Gamma^2\rho h - p)v_x + pv_x \end{pmatrix} = 0, \tag{10}$$

With the operator separation scheme, the system of Equation (10) is divided into the Eulerian stage with the pressure forces at work:

$$\frac{\partial}{\partial t} \begin{pmatrix} \Gamma\rho \\ \Gamma^2\rho hv_x \\ \Gamma^2\rho hv_y \\ \Gamma^2\rho hv_z \\ \Gamma^2\rho h - p \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} 0 \\ p \\ 0 \\ 0 \\ pv_x \end{pmatrix} = 0 \tag{11}$$

and the Lagrangian stage at which the advective transport takes place:

$$\frac{\partial}{\partial t} \begin{pmatrix} \Gamma \rho \\ \Gamma^2 \rho h v_x \\ \Gamma^2 \rho h v_y \\ \Gamma^2 \rho h v_z \\ \Gamma^2 \rho h - p \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho \Gamma v_x \\ \rho h \Gamma^2 v_x v_x \\ \rho h \Gamma^2 v_y v_x \\ \rho h \Gamma^2 v_z v_x \\ (\Gamma^2 \rho h - p) v_x \end{pmatrix} = 0. \tag{12}$$

For the systems *O* (Original)—(10), *E* (Eulerian)—(11), and *A* (Lagrangian)—(12), we will use a Rusanov-type scheme, which can be written for an arbitrary cell in a one-dimensional statement in the following form:

$$\frac{U_{i+\frac{1}{2}}^{n+1} - U_{i+\frac{1}{2}}^n}{\tau} + \frac{F_{i+1}^{O,E,A} - F_i^{O,E,A}}{\Delta} = 0, \tag{13}$$

where  $\Delta$  is the grid spacing,  $\tau$  is a time-step, and  $F_i^{O,E,A}$  are the solutions of Riemann problems for equations *O*, *E*, and *A* on the interface between cell  $i - \frac{1}{2}$  and cell  $i + \frac{1}{2}$ . We have

$$F^{E,A} = \frac{F_L^{E,A} + F_R^{E,A}}{2} + \frac{\lambda}{2} (U_L - U_R), \tag{14}$$

where *L* denotes the left cell and *R* the right one. Let us solve the system at each stage of the method.

To solve system (11), we find the eigenvalues of the Jacobian. Omitting long but obvious manipulation, we obtain

$$\lambda_R = \max \left( v_{n,R} \frac{(1 - c_{s,R}^2) + c_{s,R} \Gamma_R^{-1} \omega_R}{1 - c_{s,R}^2 v_R^2} - v_{n,R}, 0 \right), \tag{15}$$

$$\lambda_L = \min \left( v_{n,L} \frac{(1 - c_{s,L}^2) - c_{s,L} \Gamma_L^{-1} \omega_L}{1 - c_{s,L}^2 v_L^2} - v_{n,L}, 0 \right), \tag{16}$$

where  $\omega = \sqrt{1 - v_n^2 - c_s(v^2 - v_n^2)}$ , and  $v_n$  is the normal velocity component. To implement Rusanov’s scheme, at the Eulerian stage we use the maximum characteristic angle:

$$\lambda^E = \max(|\lambda_R|, |\lambda_L|). \tag{17}$$

The scheme for calculating the flux is

$$F^E = \frac{F_L^E + F_R^E}{2} + \frac{\lambda^E}{2} (U_L - U_R), \tag{18}$$

Note that the characteristics are limited by the angle associated with the speed of light, that is, the condition  $\lambda \leq 1$  is satisfied. In this way we can considerably simplify the numerical scheme.

When solving the system of Equation (12), the eigenvalue of the Jacobian matrix is the numerical advection speed calculated from the equation

$$\lambda^A = \frac{\sqrt{\rho_L} v_{n,L} + \sqrt{\rho_R} v_{n,R}}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \tag{19}$$

where, taking into account Rusanov’s scheme (14) and the form of Equation (12), we obtain the following simple scheme:

$$F^A = \lambda^A \begin{cases} U_L, \lambda^A \geq 0 \\ U_R, \lambda^A < 0 \end{cases}. \tag{20}$$

Thus, we have obtained fluxes for the Eulerian ( $E$ ) and Lagrangian ( $A$ ) stages. The flux for solving the full system of equations of special relativistic hydrodynamics is the sum of these fluxes:

$$F^* = F^O = \lambda^A \begin{cases} U_L, \lambda^A \geq 0 \\ U_R, \lambda^A < 0 \end{cases} + \frac{F_L^E + F_R^E}{2} + \frac{\lambda^E}{2}(U_L - U_R). \tag{21}$$

Taking into account that the velocity is naturally limited by the speed of light, the flux in trivial form may be written as follows:

$$F^* = \frac{F_L + F_R}{2} + \frac{U_L - U_R}{2}, \tag{22}$$

To reduce the dissipation of the numerical method, we will use a piecewise parabolic reconstruction of the physical variables. For definiteness, a piecewise parabolic function of a physical variable  $q(x)$  will be constructed on a regular grid with spacing  $\Delta$  on the interval  $[x_i, x_{i+1}]$ . For simplicity, we will use subscript  $i$ . In general form, the parabola may be written as follows:

$$q(x) = q_i^L + \xi \left( \Delta q_i + q_i^{(6)}(1 - \xi) \right),$$

where  $q_i$  is the value in the cell center,  $\xi = (x - x_i)\Delta^{-1}$ ,  $\Delta q_i = q_i^L - q_i^R$ , and  $q_i^{(6)} = 6(q_i - 1/2(q_i^L + q_i^R))$ ; the conservation condition is

$$q_i = \Delta^{-1} \int_{x_i}^{x_{i+1}} q(x) dx.$$

Let us describe the construction of the parabola step by step.

At the first step, we construct  $\delta q_i = 1/2(q_{i+1} - q_{i-1})$ . To avoid the extrema of the functions, we use the formula

$$\delta_m q_i = \begin{cases} \min(|\delta q_i|, 2|q_{i+1} - q_i|, 2|q_i - q_{i-1}|) \text{sign}(\delta q_i), \\ \quad (q_{i+1} - q_i)(q_i - q_{i-1}) > 0 \\ 0, (q_{i+1} - q_i)(q_i - q_{i-1}) \leq 0 \end{cases}.$$

Then, we recalculate the values at the boundary using the fourth-order interpolant

$$q_i^R = q_{i+1}^L = q_{i+1/2} = 1/2(q_i + q_{i+1}) - 1/6(\delta_m q_{i+1} - \delta_m q_i).$$

At the second step, we begin to construct the local parabola itself using the formula

$$\Delta q_i = q_i^L - q_i^R, \quad q_i^{(6)} = 6(q_i - 1/2(q_i^L + q_i^R)).$$

If the local parabola is nonmonotone (this may take place at discontinuities), we rearrange the values at the boundaries  $q_i^L, q_i^R$  by the formulas

$$\begin{aligned} q_i^L &= q_i, q_i^R = q_i, (q_i^L - q_i)(q_i - q_i^R) \leq 0, \\ q_i^L &= 3q_i - 2q_i^R, \Delta q_i q_i^{(6)} > (\Delta q_i)^2, \\ q_i^R &= 3q_i - 2q_i^L, \Delta q_i q_i^{(6)} < -(\Delta q_i)^2. \end{aligned}$$

Thus, the boundary values satisfy the monotonicity conditions.

At the third step, the parameters of the parabola are reconstructed taking into account the new values at the cell boundaries:

$$\begin{aligned} \Delta q_i &= q_i^L - q_i^R, \\ q_i^{(6)} &= 6(q_i - 1/2(q_i^L + q_i^R)). \end{aligned}$$

Recall that the local parabolas are used as important parts of the Riemann problem, since there may be discontinuities at the interfaces.

At the fourth step, the parabola is finally reconstructed, taking into account the new values at the cell boundaries:

$$\Delta q_i = q_i^L - q_i^R,$$

$$q_i^{(6)} = 6(q_i - 1/2(q_i^L + q_i^R)).$$

As a result, we obtain a local parabola in each cell  $[x_{i-1/2}, x_{i+1/2}]$ .

The left and right values of the physical variables are recalculated using the formulas

$$q_L(-\lambda_L\tau) = q_i^R - \frac{\lambda_L\tau}{2\Delta} \left( \Delta q_i - q_i^{(6)} \left( 1 - \frac{2\lambda_L\tau}{3\Delta} \right) \right),$$

otherwise

$$q_R(\lambda_R\tau) = q_i^L + \frac{\lambda_R\tau}{2\Delta} \left( \Delta q_i + q_i^{(6)} \left( 1 - \frac{2\lambda_R\tau}{3\Delta} \right) \right),$$

where  $\lambda_{L,R}$  is the corresponding eigenvalue at each step.

The Courant–Friedrichs–Lewy (CFL) condition is a necessary condition for the stability of an explicit Godunov-type scheme. From this condition, we determine the time-step. Completing the description of the numerical method, we again use the limitation of velocities by the light speed to limit the time-step. Then, the CFL condition can be formulated as follows:

$$\frac{\tau}{\Delta} \leq CFL, \tag{23}$$

where *CFL* is the Courant–Friedrichs–Lewy number.

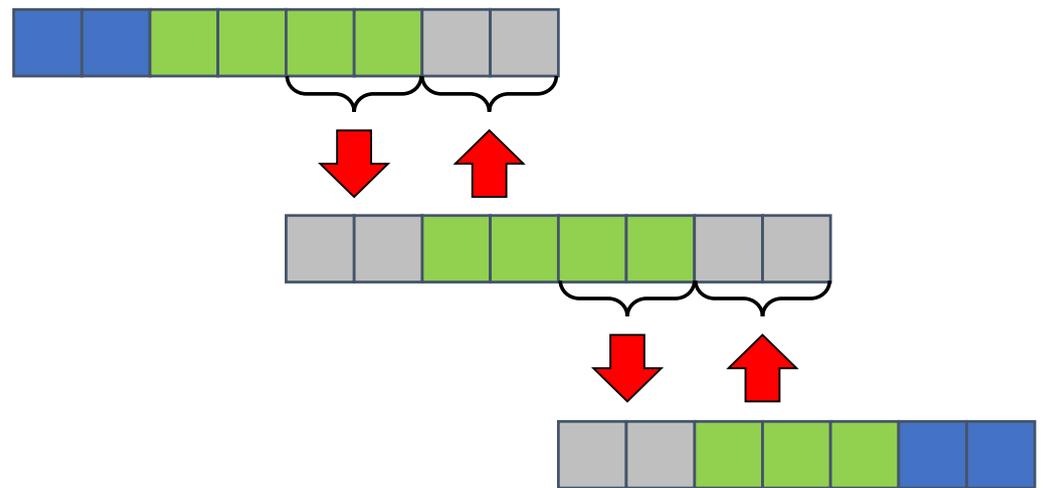
### 3. Parallel Code

When using the  $256^3$  computational grid, one time-step is calculated in approximately a minute; a complete calculation takes approximately a day.

Of course, it is necessary to develop a parallel implementation of the code based on the geometric decomposition of the computational domain with overlapping subdomains. Decomposition occurs on the images of the Coarray Fortran program. The description and study of the parallel implementation will be described below. In this section, we consider the structure of the parallel code. For simplicity, we will describe in detail a one-dimensional version of the code. However, all computational experiments and efficiency studies will be made using a three-dimensional version (see <https://gitflic.ru/project/igorkulikov/rhd3dhllk>, accessed on 29 May 2022).

The parallel implementation is based on a one-dimensional geometric decomposition of the computational domain with overlapping over two cells. This is due to the use of a five-point stencil for calculating the parabolas. The interprocess communications are performed by using Coarray Fortran based on one-way communications implemented in MPI-3.0 standard. Note that the interprocess communication tools are standards of the language, not of an external library.

This implies that the software of the code being developed is portable. A schematic diagram of the geometric decomposition of calculations and the one-way communications using, as an example, a one-dimensional implementation, are shown in Figure 1, where the blue color shows the cells for the boundary conditions, the green color denotes the calculation cells, and the gray color is the overlapping of subdomain cells. Note that in one-way interaction the exchange process is always initiated by the left process (image for Coarray Fortran programs). This is due to the fact that reading and writing data of a right image is always in the first cells. Therefore, the information about the local computational domain size contained in each image need not be stored in each program image. Nevertheless, this organization of data exchanges ensures data integrity when calculating the next time-step. Implementation details of the Coarray Fortran code are given in Appendix A.



**Figure 1.** Geometric decomposition of the calculation domain and scheme of communications.

To study the parallel implementation, we use two nodes of an NKS-1P cluster, each with two 24-core Intel Xeon 6248R processors with Hyper-Threading support. Let us perform a preliminary study of the program speedup on a  $256^3$  grid with various numbers of nodes, sockets, cores, and threads, which, upon multiplication, forms the number of images.

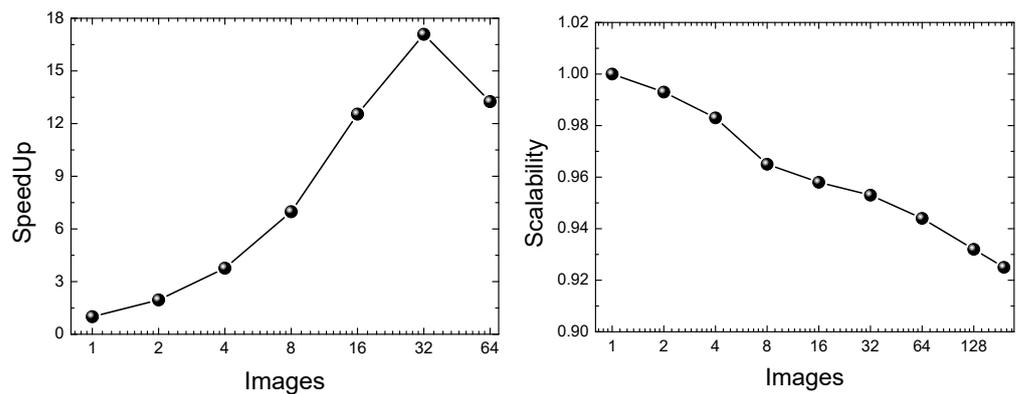
A study of the calculation time shows that to increase the program images optimally, the following scheme should be used: increase the number of nodes, threads, processors, and cores, as shown in Table 1.

**Table 1.** Calculation time for one time-step when using a  $256^3$  grid.

| Nodes | Sockets | Cores | Threads | Images | Time (s) |
|-------|---------|-------|---------|--------|----------|
| 1     | 1       | 1     | 1       | 1      | 27.542   |
| 1     | 1       | 2     | 1       | 2      | 14.724   |
| 1     | 1       | 1     | 2       | 2      | 14.443   |
| 1     | 2       | 1     | 1       | 2      | 14.122   |
| 2     | 1       | 1     | 1       | 2      | 14.072   |
| 1     | 1       | 2     | 2       | 4      | 7.345    |
| 1     | 1       | 4     | 1       | 4      | 7.485    |
| 1     | 2       | 1     | 2       | 4      | 7.395    |
| 2     | 1       | 2     | 1       | 4      | 7.398    |
| 2     | 1       | 1     | 2       | 4      | 7.318    |
| 2     | 2       | 1     | 1       | 4      | 7.378    |
| 1     | 1       | 8     | 1       | 8      | 3.952    |
| 1     | 1       | 4     | 2       | 8      | 4.045    |
| 1     | 2       | 2     | 2       | 8      | 4.051    |
| 1     | 2       | 4     | 1       | 8      | 3.985    |
| 2     | 2       | 1     | 2       | 8      | 3.945    |
| 2     | 1       | 2     | 2       | 8      | 3.946    |
| 2     | 2       | 2     | 1       | 8      | 3.971    |

To study the speedup of the software implementation, we use a  $256^3$  calculation grid with different numbers of Coarray Fortran program images, and to study the scalability, a  $128^3$  grid per one program image.

A 17-fold speedup was achieved with 32 images and 92% efficiency when using 192 Coarray Fortran images, results are shown in Figure 2.



**Figure 2.** Speedup (left) and scalability (right) of the parallel program. A 17-fold speedup was achieved on a  $256^3$  grid using 32 images of the Coarray Fortran program, and 92% efficiency was achieved using 192 Coarray Fortran images when each image had a grid of  $128^3$ .

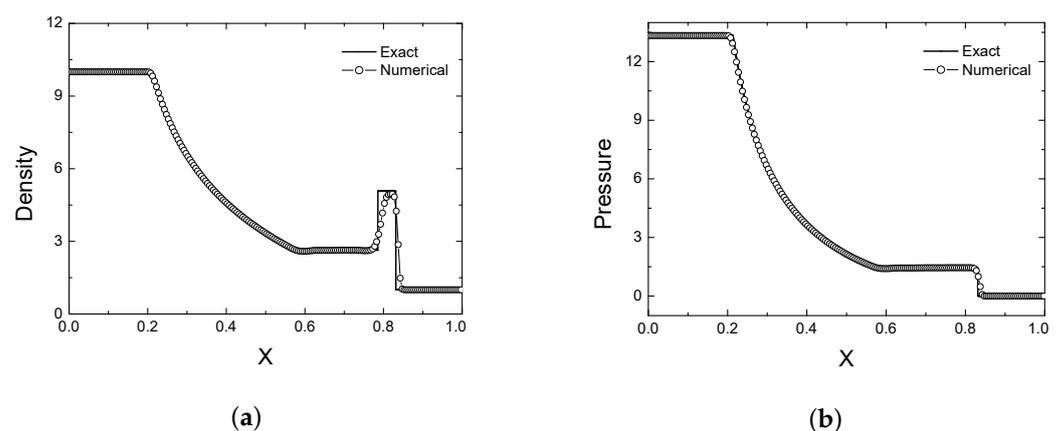
**4. Verification**

For verification, we formulate three one-dimensional tests to consider the dynamics of a relativistic gas on the interval  $[0, 1]$  up to  $t = 0.4$ . At point  $x_0 = 0.5$  from the initial time the gas has a discontinuity. The adiabatic index  $\gamma = 5/3$ . We use non-reflecting boundary conditions. In the simulation, 200 cells for all one-dimensional tests are used. In addition to comparing a numerical solution with an analytical one, we will analyze the order of convergence of the method in  $L_1$  norm:

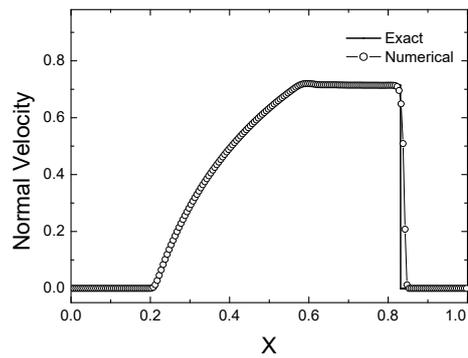
$$L_1 = \sum_i h |u_i - u(x_i)|,$$

where  $u(x_i)$  is the exact solution at point  $x_i$ ,  $u_i$  is the numerical result, and  $h$  is the uniform grid spacing.

First, consider a one-dimensional shock wave test with the following parameters: pressure  $p^L = 40/3$ , density  $\rho^L = 10$ , and velocity  $v^L = 0$ . The parameters of the gas to the right of the discontinuity are: pressure  $p^R = 10^{-8}$ , density  $\rho^R = 1$ , and velocity  $v^R = 0$ . The simulation results are shown in Figures 3 and 4.

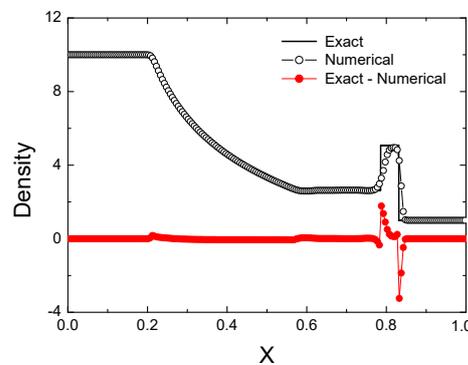


**Figure 3.** Cont.

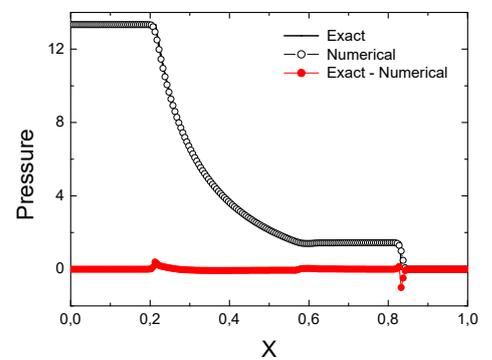


(c)

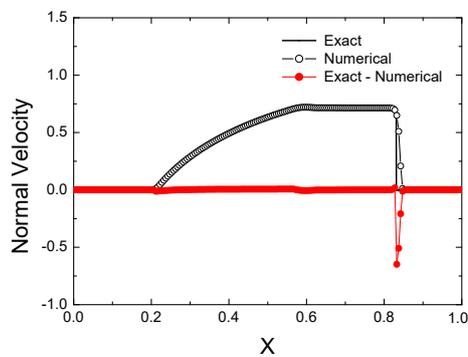
**Figure 3.** One-dimensional shock wave problem. Exact solution (solid line), numerical solution (circles) for physical variables: (a) density, (b) pressure, and (c) normal velocity component.



(a)



(b)



(c)

**Figure 4.** One-dimensional shock wave problem. Exact solution (solid line), numerical solution (circles), and difference between exact and numerical solution (red circles) for physical variables: (a) density, (b) pressure, and (c) normal velocity component.

Figure 5 shows the behavior of the numerical solution error and the order of convergence of the method at grid refinement.

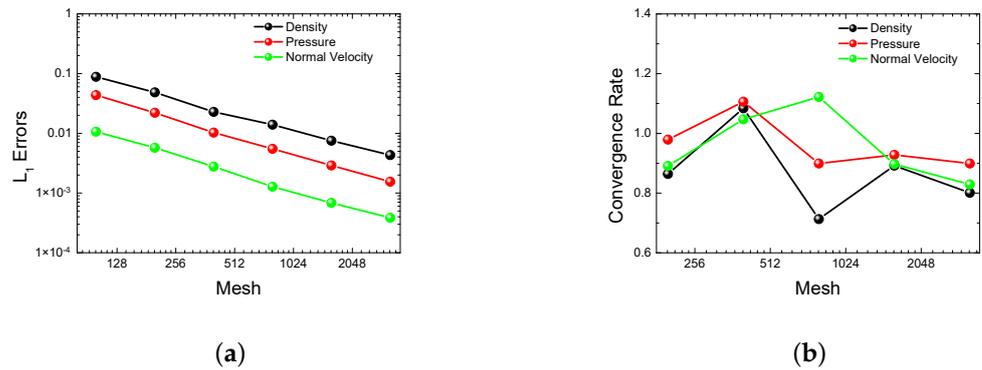


Figure 5. One-dimensional shock wave problem. Behavior of errors (a) and order of convergence (b).

The method correctly reproduces all components of the solution. The shock wave dissipates only over two cells, which is due to the use of piecewise parabolic methods. The shock wave peak in the density function is reproduced correctly. No additional dissipation is observed on the rarefaction wave. Note that there are no entropy traces in the pressure and velocity functions. Although the error curves decrease rather smoothly, for small grids there are jumps in the order of convergence curves. With significant grid refinement, the density and velocity functions converge to 0.8, whereas the pressure function converges to 0.9.

As a second test, consider a stronger shock wave with the following parameters: pressure  $p^L = 1000$ , density  $\rho^L = 1$ , and velocity  $v^L = 0$ . The parameters of the gas to the right of the discontinuity are: pressure  $p^R = 10^{-2}$ , density  $\rho^R = 1$ , and velocity  $v^R = 0$ . The simulation results are shown in Figures 6 and 7.

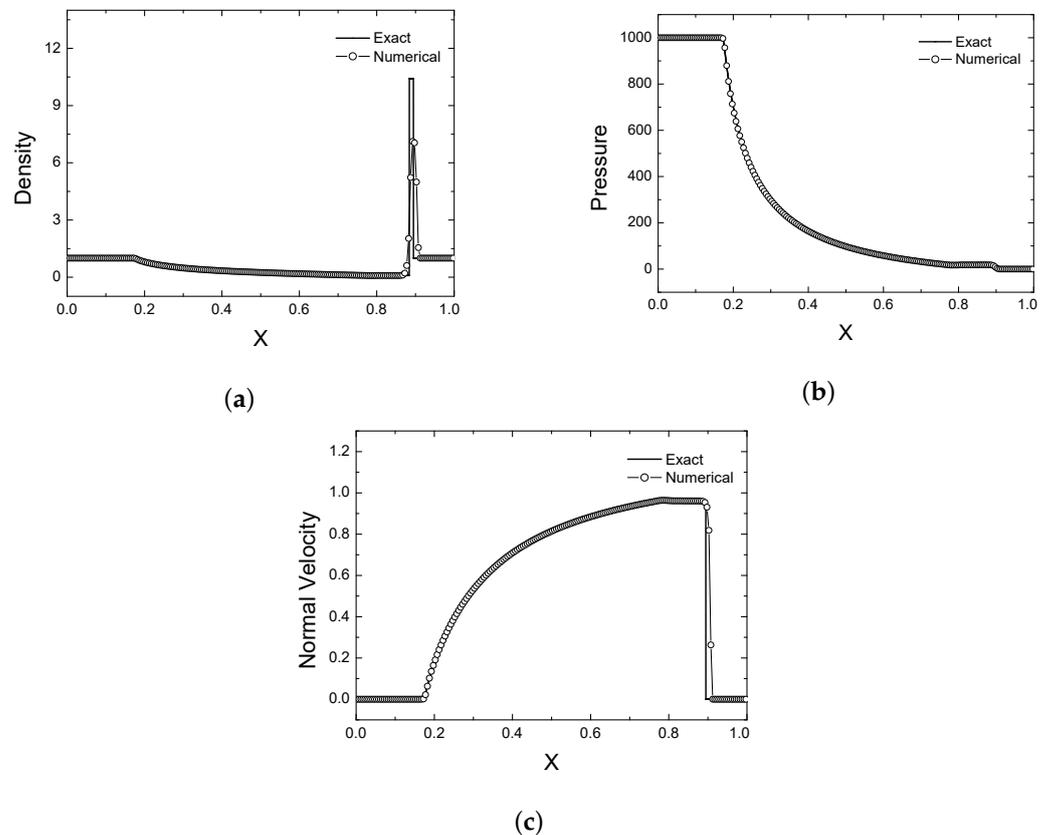
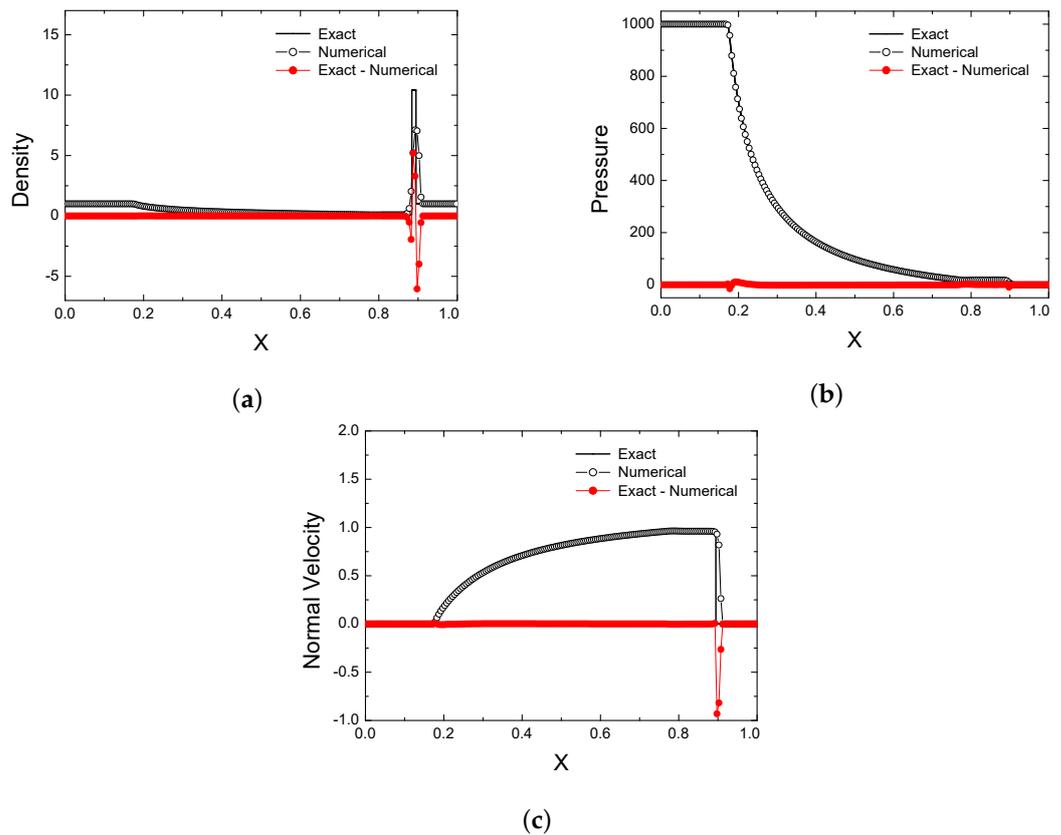
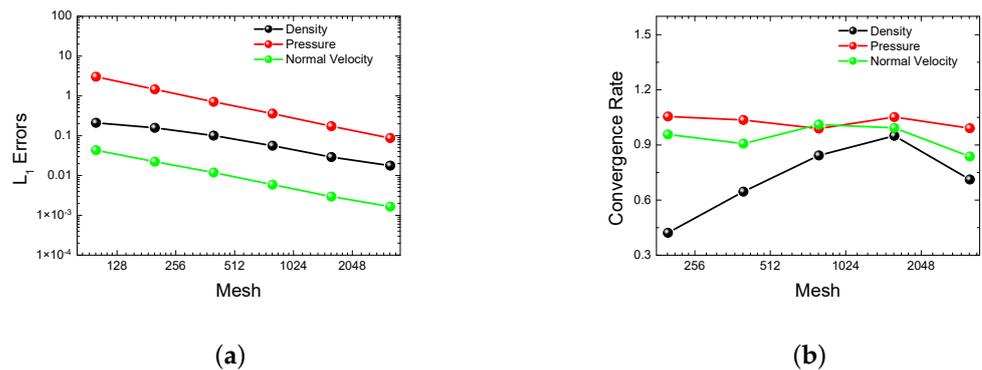


Figure 6. One-dimensional strong shock wave problem. Exact solution (solid line) and numerical solution (circles) for physical variables: (a) density, (b) pressure, and (c) normal velocity component.



**Figure 7.** One-dimensional strong shock wave problem. Exact solution (solid line), numerical solution (circles), and difference between exact and numerical solution (red circles) for physical variables: (a) density, (b) pressure, and (c) normal velocity component.

Figure 8 shows the behavior of the numerical solution error and the order of convergence of the method at grid refinement.

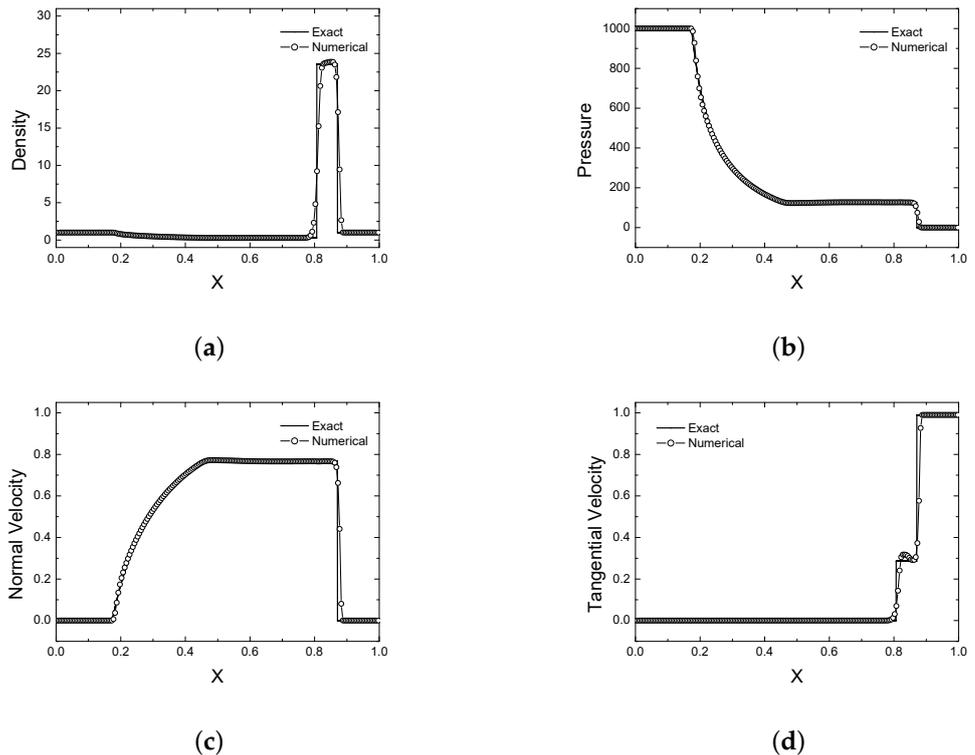


**Figure 8.** One-dimensional strong shock wave problem. Behavior of errors (a) and order of convergence (b).

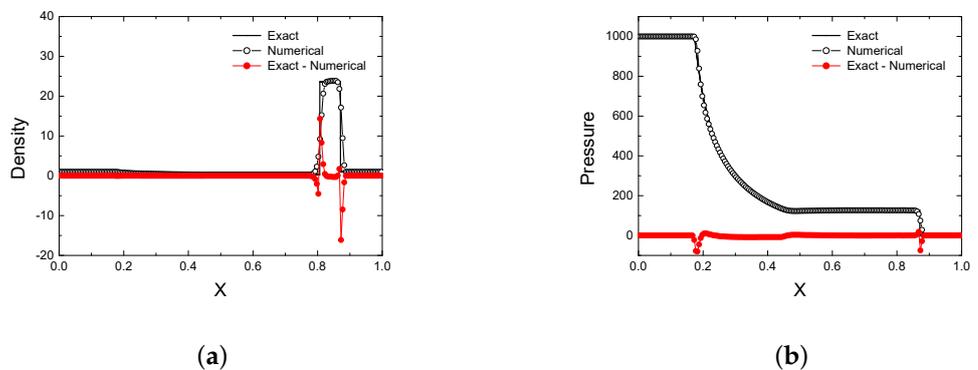
The main difficulty of this test is that behind the shock wave front there is a very thin shell, which covers only two cells in the experiment. Even at this resolution, the numerical method was able to reproduce the shell location, and only 60% of the density peak was reproduced. Note that in the shell zone the solution dissipates over two cells in the direction of the shock wave and over two cells behind the shell front. At the same time, the mass of the dissipation zone corresponds well to the mass of the shell. The other parts of the solution are reproduced correctly. The difficulties in reproducing the shell impose some restrictions on the order of convergence of the density function. Note that on small grids the shell is reproduced with a small, increasing order of convergence to 0.6. At the same time,

the convergence of the velocity and pressure functions is quite stable; they converge to the first-order for the pressure function since the smooth part of the rarefaction wave prevails in the solution, and to 0.8 for the velocity function due to a large shock wave amplitude.

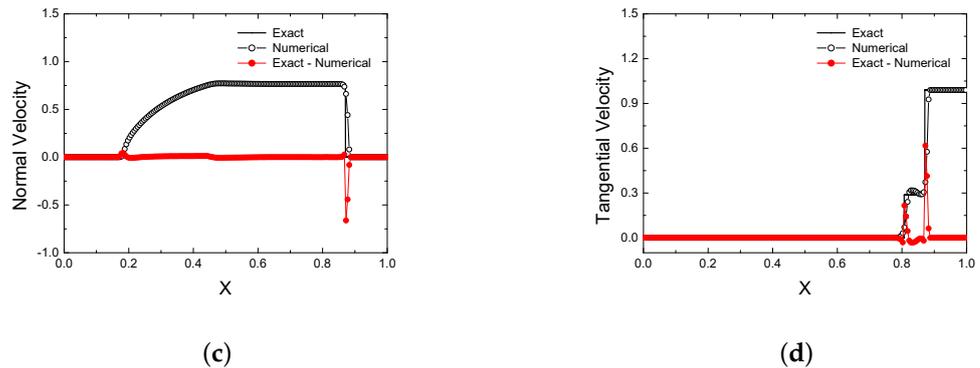
Let us add complexity to the second test by adding a nonzero tangential initial velocity. The parameters of the gas to the left of the discontinuity are: pressure  $p^L = 1000$ , density  $\rho^L = 1$ , normal velocity component  $v_x^L = 0$ , and tangential velocity component  $v_y^L = 0$ . The parameters of the gas to the right of the discontinuity are: pressure  $p^R = 10^{-2}$ , density  $\rho^R = 1$ , normal velocity component  $v_x^R = 0$ , and tangential velocity component  $v_y^R = 0.99$ . The simulation results are shown in Figures 9 and 10.



**Figure 9.** One-dimensional strong shock wave problem with tangential initial velocity. Exact solution (solid line) and numerical solution (circles) for physical variables: (a) density, (b) pressure, (c) normal velocity component, and tangential velocity component (d).

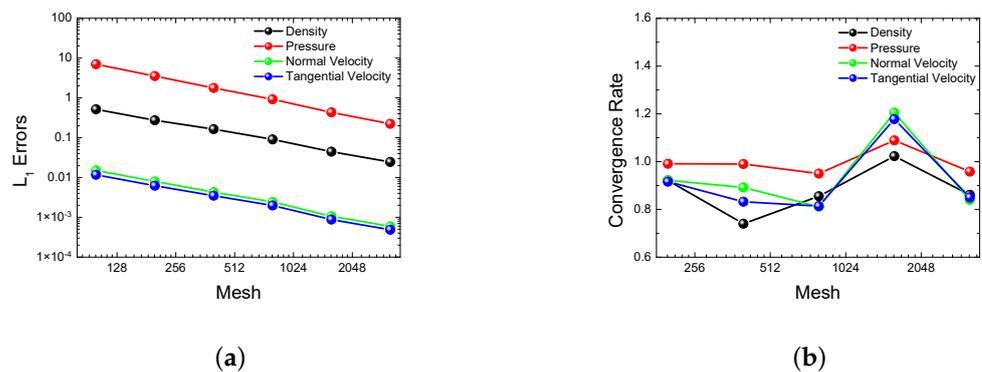


**Figure 10.** Cont.



**Figure 10.** One-dimensional strong shock wave problem with tangential initial velocity. Exact solution (solid line), numerical solution (circles), and difference between exact and numerical solution (red circles) for physical variables: (a) density, (b) pressure, (c) normal velocity component, and tangential velocity component (d).

Figure 11 shows the behavior of the numerical solution error and the order of convergence of the method at grid refinement.



**Figure 11.** One-dimensional strong shock wave problem with tangential initial velocity. Behavior of errors (a) and order of convergence (b).

In contrast to ideal hydrodynamics, in the equations of special relativistic hydrodynamics, the tangential velocity greatly affects the normal velocity component due to the nonlinearity of the Lorentz factor. The test is much more complicated in that the tangential velocity is given in a cold gas. Therefore, the method must correctly reproduce the interaction of the velocity resulting from the shock impact of the hot gas. Note that the method reproduces well the shock wave zone with dissipation over two cells at the front and over four cells behind the shock wave shell zone, as in the previous test. Not only the location but also the density peak amplitude is reproduced correctly. The absence of shell-type solutions causes more-regular behavior of the error and the order of convergence of the method. Thus, for the pressure function, the order of convergence is one (the reason being the same as in the previous test), and for the density and velocity functions it is 0.8.

Let us take up the problem of discontinuity breakdown in a two-dimensional statement. For this, consider gas dynamics in the  $[-1/2; 1/2]^2$  domain up to time  $t = 0.4$ . Let us divide the domain into four parts in geographical directions: NW— $[-1/2; 0] \times [0; 1/2]$ , NE— $[0; 1/2] \times [0; 1/2]$ , SW— $[-1/2; 0] \times [-1/2; 0]$ , SE— $[-1/2; 0] \times [0; 1/2]$ . Let us define the parameters of the gas in each part as follows:

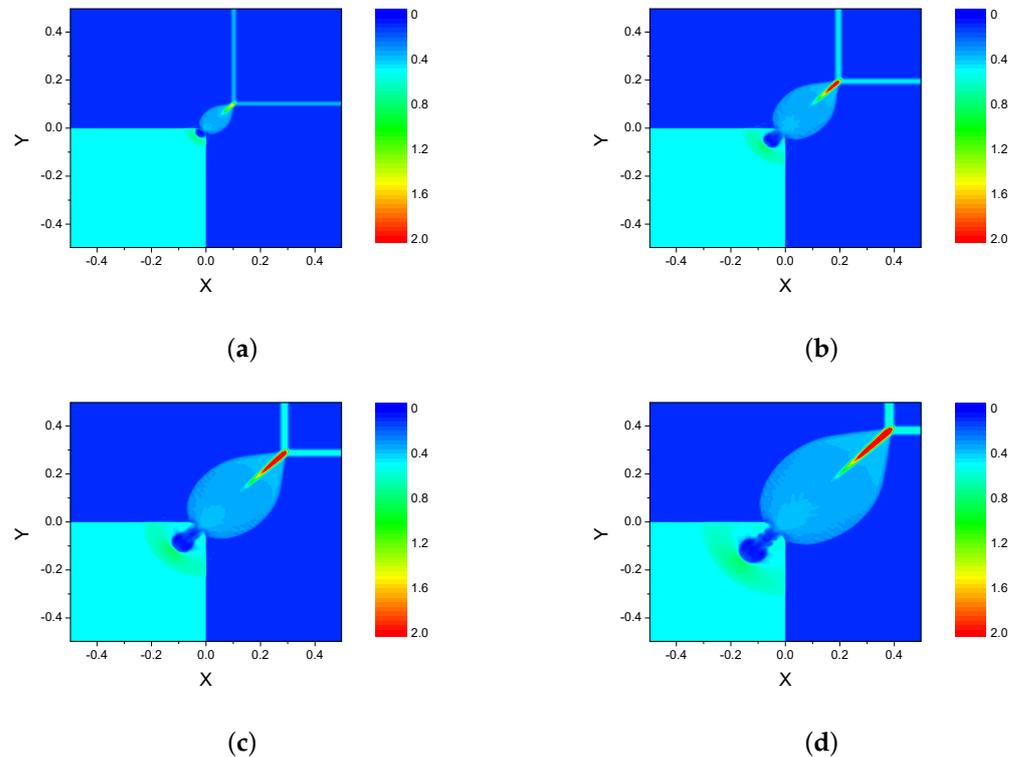
$$(\rho, v_x, v_y, p)^{NE} = (0.1, 0, 0, 0.01),$$

$$(\rho, v_x, v_y, p)^{NW} = (0.1, 0.99, 0, 1),$$

$$(\rho, v_x, v_y, p)^{SE} = (0.1, 0, 0.99, 1),$$

$$(\rho, v_x, v_y, p)^{SW} = (0.5, 0, 0.99, 1),$$

The adiabatic index  $\gamma = 5/3$ . Nonreflecting boundary conditions are set on all boundaries. In the simulation, a grid of  $200^2$  cells is used. The results of the simulation are presented in Figure 12.

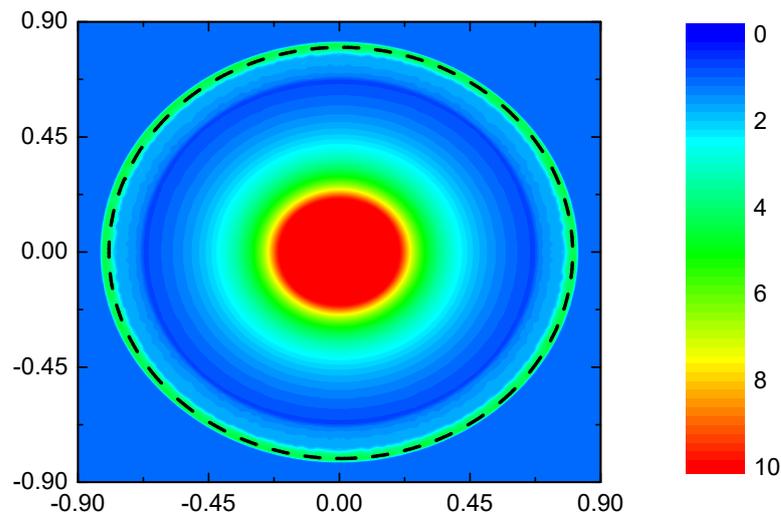


**Figure 12.** Two-dimensional test on discontinuity breakdown. Density function distribution at  $t = 0.1$  (a),  $t = 0.2$  (b),  $t = 0.3$  (c), and  $t = 0.4$  (d). The evolution of the interaction of discontinuity decays in a two-dimensional formulation is shown. The first feature of the test is a numerical artifact due to the interaction of contact discontinuities. The second feature of the test is the interaction of shock waves and the formation of a density jump at the front.

The interaction between domains NW/SW and SE/SW form contact discontinuities with a jump in the tangential velocities. These contact discontinuities result in a numerical artifact that expands over time in domain SW. The formation of two interacting shock waves causes a density jump in domain NE, which has the same amplitude and limited dissipation. In this test, only the interaction of different waves is investigated, since the reproduction of individual waves has already been considered in the previous test problems.

### 5. Numerical Simulation

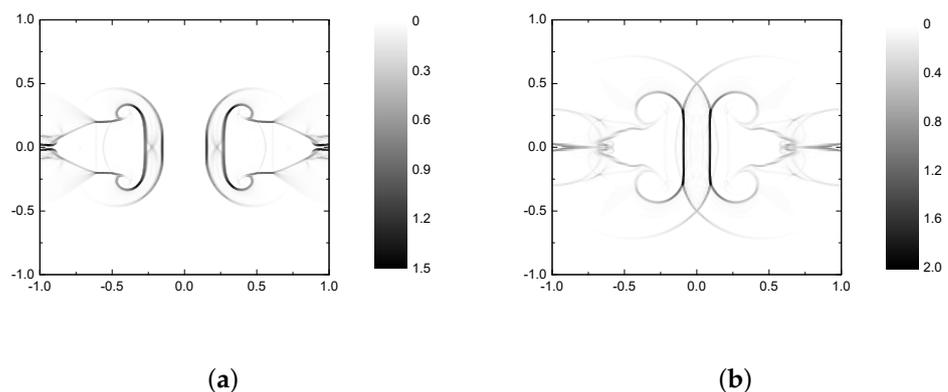
Let us consider the problem of shock wave formation for a gas at rest in a three-dimensional statement. For this, consider gas dynamics in the  $[-1; 1]^3$  domain up to time  $t = 0.4$ . Let the domain have a spherical radius  $r_0 = 0.5$ . In the internal part of the domain, the gas parameters are: pressure  $p_0 = 40/3$  and density  $\rho_0 = 10$ . In the external part of the domain, the gas parameters are: pressure  $p = 10^{-8}$  and density  $\rho = 1$ . The adiabatic index  $\gamma = 5/3$ . Nonreflecting boundary conditions are used. The results of the simulation are presented in Figure 13. A grid of  $400^3$  cells is used in the simulation.



**Figure 13.** Density distribution in a three-dimensional shock wave at  $t = 0.4$ . This test is an extension of the first one on the disintegration of the discontinuity to the three-dimensional case and demonstrates, in addition to the correct reproduction of the shock wave front, the invariance of the solution concerning rotation and the absence of the Carbuncle effect.

According to the results of this test, the solution is invariant with respect to rotation in the three-dimensional case. The density peak is reproduced correctly, and the shock wave is reproduced well (its analytical solution is shown by a dotted line). This problem is considered as a central explosion of a supernova without analyzing the cause of the explosion (thermonuclear reactions or a core collapse) in the model of special relativistic hydrodynamics.

As a second problem, consider interacting relativistic jets. We will simulate two interacting galactic jets with density  $\rho_J = 10^{-3} \text{ cm}^{-3}$  and radius  $R_J = 200 \text{ pc}$  moving with Lorentz factor  $\Gamma = 10$  and having relativistic Mach number  $\mathcal{M} = 8$ . The atmosphere of the galaxy has temperature  $T_A = 10^7 \text{ K}$  and density  $\rho_A = 10^{-2} \text{ cm}^{-3}$ . The adiabatic index  $\gamma = 5/3$ . Figure 14 presents the results of simulating the evolution of the galactic jets.



**Figure 14.** Schlieren density in the equatorial plane in the evolution of a galactic jet at 1000 years (a) and 2000 years (b). Schlieren density was chosen to demonstrate the interaction of discontinuities in the jet interaction problem. Until the moment of interaction, the structure of the shock wave front and the hotpoint, which is formed in the shell due to the interaction of waves, is visible. After the interaction of jets, an almost vertical shell structure is formed between two jets, while the shock wave fronts do not contain any numerical artifacts that other implementations may have in such interactions.

In both jets, a spherical shock wave propagates forward at the speed of light. Behind the shock front there is a shell with a system of interacting shock waves. Inside the shell

there is a sparse cocoon area. During the interaction of the jets, the hot point zone becomes extended and covers about 1 kpc.

## 6. Discussion and Conclusions

Distinctive features of the code are as follows:

1. To increase the order of accuracy on smooth solutions and decrease dissipation at discontinuities, a piecewise parabolic reconstruction is used for the physical variables since they are responsible for the thermodynamic state of the continuous medium [43].
2. We use a piecewise-parabolic reconstruction of physical variables although the calculations in the scheme are based on the use of vectors of conservative variables and their flows through cell boundaries. This is due to the fact that they determine the thermodynamic state of a continuous medium. This was emphasized in [43] when studying the accuracy of the discontinuous Galerkin method on the Einfeldt problem.
3. When using Rusanov-type methods as part of Godunov's method, no full spectral problem is solved; this is the case, for instance, when using an equation of state of the Taub–Mathews type [56].
4. We plan to extend the model for a multiphase medium, where the equation of state is more complicated [57]. In the present paper, we neglect the right-hand side of the energy equation, although this term can be easily included in our numerical scheme [58].
5. On the basis of the above computational experiments, we will try to avoid using the MPI technology where possible. The Coarray Fortran technology is a good alternative to develop parallel programs for distributed memory architectures.
6. In a number of problems, a significant part of the computational domain is occupied by flows with a high value of the Lorentz factor and low pressure values. In these cases, the energy equation is used in the form:

$$\frac{\partial}{\partial t} (\Gamma^2 \rho h - p - \Gamma \rho) + \sum_{k=1}^3 \frac{\partial}{\partial x_k} (\rho h \Gamma^2 v_k - \rho \Gamma v_k) = 0. \quad (24)$$

The above statements are disputable. However, we believe they are appropriate when discussing the code. To conclude: in the present paper, a new code for describing flows in a model of special relativistic hydrodynamics and implemented with Coarray Fortran parallel programming technology was presented. During the development of the code, a new original numerical method was constructed with high-order accuracy on smooth solutions and low dissipation at discontinuities. The numerical method has been thoroughly verified. A code scalability of 92% is obtained on a cluster with Intel Xeon 6248R NKS-1P with 192 Coarray Fortran images. As an astrophysical application, we considered two interacting relativistic jets.

**Author Contributions:** Methodology, I.K., I.C. and D.K.; Software, I.K., I.C., D.K. and V.P.; Supervision, I.K.; Validation, V.P., A.S. and I.U.; Visualization, O.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Russian Science Foundation (project 18-11-00044).

**Acknowledgments:** This work was supported by the Russian Science Foundation (project 18-11-00044) <https://rscf.ru/project/18-11-00044/> (accessed on 29 May 2022).

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

A typical pattern for the Coarray Fortran program (Listing A1) is as follows:

**Listing A1.** The Pattern for Coarray Fortran Codes.

1. `double precision , allocatable :: array (:)[:]`

```

2.   irank = this_image()
3.   isize = num_images()

4.   Nlocal = N/isize
5.   i_start_index = (N/isize)*(irank-1)
6.   if(irank-1 < mod(N, isize)) then
7.     Nlocal = Nlocal + 1
8.     i_start_index = i_start_index + irank - 1
9.   else
10.    i_start_index = i_start_index + mod(N, isize)
11.  endif

12.  allocate(array(Nlocal+4)[*])

13.  if(irank < isize) then
14.    array(1)[irank+1] = array(Nlocal+1)
array(2)[irank+1] = array(Nlocal+2)
15.    array(Nlocal+3) = array(3)[irank+1]
array(Nlocal+4) = array(4)[irank+1]
16.  endif
17.  sync all

18.  x = (i + i_start_index - 2)*h - h/2.d0
19.  array(i) = f(x)

20.  deallocate(array)

```

Let us consider Listing A1 in detail:

1. definition of dynamic **array** located in shared memory between program images;
2. definition of variable **irank**—Coarray Fortran program image number;
3. definition of variable **isize**—the number of images of Coarray Fortran program;
4. with known calculation grid size **N**, calculate the first approximation of the size of the grid **Nlocal** for the corresponding image of the program;
5. calculate the first approximation of the shift index **i\_start\_index** of the array part;
6. to obtain a uniform distribution of cells over the images, in the first processes the local array size must not exceed the other ones by more than unity;
7. add unity to the calculated local size for all first images;
8. shift the shift index taking into account the fact that the size of the previous images is greater than the first approximation by unity;
9. in the subsequent processes, the first approximation of the grid part size is not recalculated;
10. recalculate the shift index with the local sizes of the first images increased by unity;
11. this completes the calculation of the shift index and local size;
12. allocate memory for **array** taking into account boundary conditions and/or overlapping areas;
13. perform one-way communications with initialization of exchange between the left images of the program;
14. write two right extreme values of the local calculation grid into the first two cells—overlapping areas of the next program image;
15. write the first two cells of the local calculation grid from the next program image into two rightmost cells—overlapping areas;
16. this completes the exchanges between the overlapping areas;
17. synchronize data in all images of the Coarray Fortran program;
18. calculate the coordinates of the center of cell of size  $h$ , taking into account the shift index;
19. assign the function value to the corresponding array element;

20. remove the memory allocated for the array.

The above listing and description of its main items are sufficient for writing a Coarray Fortran program implementing the numerical method.

## References

1. Tyulbashev, S.A.; Chernikov, P.A. Relative variations of the physical parameters of variable extragalactic radio sources. *Astron. Rep.* **2004**, *48*, 716–723. [[CrossRef](#)]
2. Chechetkin, V.M.; Dyachenko, V.F.; Ginzburg, S.L.; Paleichik, V.V.; Fimin, N.N.; Sudarikov, A.L. On the generation mechanism of hard cosmic gamma-ray emission from AGN jets. *Astron. Rep.* **2009**, *53*, 501–509. [[CrossRef](#)]
3. Chernov, S.V. Polarization Properties of Weakly Relativistic Cylindrical Jets. *Astron. Rep.* **2019**, *63*, 910–919. [[CrossRef](#)]
4. Sokolov, V.V.; Bisnovatyi-Kogan, G.S.; Kurt, V.G.; Gnedin, Y.N.; Baryshev, Y.V. Observational constraints on the angular and spectral distributions of photons in gamma-ray burst sources. *Astron. Rep.* **2006**, *50*, 612–625. [[CrossRef](#)]
5. Tutukov, A.V.; Cherepashchuk, A.M. Massive close binary stars and gamma-ray bursts. *Astron. Rep.* **2004**, *48*, 39–44. [[CrossRef](#)]
6. Chechetkin, V.M.; Dyachenko, V.F.; Ginzburg, S.L.; Fimin, N.N. Dynamics of an ultra-relativistic, collisionless astrophysical plasma. *Astron. Rep.* **2012**, *56*, 329–335. [[CrossRef](#)]
7. Tutukov, A.V.; Fedorova, A.V.; Cherepashchuk, A.M. Wolf-Rayet stars with relativistic companions. *Astron. Rep.* **2013**, *57*, 657–668. [[CrossRef](#)]
8. Cherepashchuk, A.M. Wolf-Rayet stars and relativistic objects: Distinctions between the mass distributions in close binary systems. *Astron. Rep.* **2001**, *45*, 120–137. [[CrossRef](#)]
9. Komissarov, S.S. Simulations of the axisymmetric magnetospheres of neutron stars. *Mon. Not. R. Astron. Soc.* **2006**, *367*, 19–31. [[CrossRef](#)]
10. Tutukov, A.V.; Bogomazov, A.I. Radio pulsars in close binaries with neutron stars. *Astron. Rep.* **2008**, *52*, 390–402. [[CrossRef](#)]
11. Fateev, V.F.; Davlatov, R.A. Space-Based Gravitational-Wave Detectors: Development of Ground-Breaking Technologies for Future Space-Based Gravitational Gradiometers. *Astron. Rep.* **2019**, *63*, 699–709. [[CrossRef](#)]
12. Liu, P.; Zhang, C.-M.; Li, D.; Yang, Y.-Y.; Zhang, J.; Zhang, J.-W. The Simulation of Orbit Decay of Double Neutron Star System PSR J1906+0746 by the Gravitational Wave Radiation. *Astron. Rep.* **2019**, *63*, 1090–1094. [[CrossRef](#)]
13. Glushak, A.P. Microquasar jets in the supernova remnant G11.2-0.3. *Astron. Rep.* **2014**, *58*, 6–15. [[CrossRef](#)]
14. Barkov, M.V.; Bisnovatyi-Kogan, G.S. Interaction of a cosmological gamma-ray burst with a dense molecular cloud and the formation of jets. *Astron. Rep.* **2005**, *49*, 24–35. [[CrossRef](#)]
15. Istomin, Y.N.; Komberg, B.V. Gamma-ray bursts as a result of the interaction of a shock from a supernova and a neutron-star companion. *Astron. Rep.* **2002**, *46*, 908–917. [[CrossRef](#)]
16. Artyukh, V.S. Phenomenological model for the evolution of radio galaxies such as Cygnus A. *Astron. Rep.* **2015**, *59*, 520–524. [[CrossRef](#)]
17. Artyukh, V.S. Effect of Aberration on the Estimated Parameters of Relativistic Radio Jets. *Astron. Rep.* **2018**, *62*, 436–439. [[CrossRef](#)]
18. Butuzova, M.S. Search for differences in the velocities and directions of the kiloparsec-scale jets of quasars with and without X-ray emission. *Astron. Rep.* **2016**, *60*, 313–321. [[CrossRef](#)]
19. Butuzova, M.S. The Blazar OJ 287 Jet from Parsec to Kiloparsec Scales. *Astron. Rep.* **2021**, *65*, 635–644. [[CrossRef](#)]
20. Malov, I.F.; Machabeli, G.Z.; Malofeev, V.M. A new model of a magnetar. *Astron. Rep.* **2003**, *47*, 232–239. [[CrossRef](#)]
21. Belyaev, V.S.; Bisnovatyi-Kogan, G.S.; Gromov, A.I.; Zagreev, B.V.; Lobanov, A.V.; Matafonov, A.P.; Moiseenko, S.G.; Toropina, O.D. Numerical Simulations of Magnetized Astrophysical Jets and Comparison with Laboratory Laser Experiments. *Astron. Rep.* **2018**, *62*, 162–182. [[CrossRef](#)]
22. Krauz, V.I.; Mitrofanov, K.N.; Kharrasov, A.M.; Ilyichev, I.V.; Myalton, V.V.; Ananyev, S.S.; Beskin, V.S. Laboratory Modeling of the Rotation of Jets Ejected from Young Stellar Objects at Studies the Azimuthal Structure of an Axial Jet at the PF-3 Facility. *Astron. Rep.* **2021**, *65*, 26–44. [[CrossRef](#)]
23. Kulikov, I. A new code for the numerical simulation of relativistic flows on supercomputers by means of a low-dissipation scheme. *Comput. Phys. Commun.* **2020**, *257*, 107532. [[CrossRef](#)]
24. Godunov, S.K. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Mat. Sb.* **1959**, *47*, 271–306.
25. Godunov, S.K.; Manuzina, Y.D.; Nazar'eva, M.A. Experimental analysis of convergence of the numerical solution to a generalized solution in fluid dynamics. *Comput. Math. Math. Phys.* **2011**, *51*, 88–95. [[CrossRef](#)]
26. Godunov, S.K.; Kulikov, I.M. Computation of Discontinuous Solutions of Fluid Dynamics Equations with Entropy Nondecrease Guarantee. *Comput. Math. Math. Phys.* **2014**, *54*, 1012–1024. [[CrossRef](#)]
27. Godunov, S.K.; Denisenko, V.V.; Klyuchinskii, D.V.; Fortova, S.V.; Shepelev, V.V. Study of Entropy Properties of a Linearized Version of Godunov's Method. *Comput. Math. Math. Phys.* **2020**, *60*, 628–640. [[CrossRef](#)]
28. Godunov, S.K.; Klyuchinskii, D.V.; Fortova, S.V.; Shepelev, V.V. Experimental Studies of Difference Gas Dynamics Models with Shock Waves. *Comput. Math. Math. Phys.* **2018**, *58*, 1201–1216. [[CrossRef](#)]
29. Kulikov, I.M. A Low-Dissipation Numerical Scheme Based on a Piecewise Parabolic Method on a Local Stencil for Mathematical Modeling of Relativistic Hydrodynamic Flows. *Numer. Anal. Appl.* **2020**, *13*, 117–126. [[CrossRef](#)]

30. Godunov, S.K. Thermodynamic formalization of the fluid dynamics equations for a charged dielectric in an electromagnetic field. *Comput. Math. Math. Phys.* **2012**, *52*, 787–799. [[CrossRef](#)]
31. Godunov, S.K. About inclusion of Maxwell's equations in systems relativistic of the invariant equations. *Comput. Math. Math. Phys.* **2013**, *53*, 1179–1182. [[CrossRef](#)]
32. Rusanov, V.V. The calculation of the interaction of non-stationary shock waves with barriers. *Comput. Math. Math. Phys.* **1961**, *1*, 267–279.
33. Kolgan, V.P. Application of the principle of minimizing the derivative to the construction of finite-difference schemes for computing discontinuous gas flows. *Uch. Zap. Tsentr. Aerogidrodin. Inst.* **1972**, *3*, 68–77.
34. Kurganov, A.; Tadmor, E. New High-Resolution Central Schemes for Nonlinear Conservation Laws and Convection-Diffusion Equation. *J. Comput. Phys.* **2000**, *160*, 214–282. [[CrossRef](#)]
35. Popov, M.; Ustyugov, S. Piecewise parabolic method on local stencil for gasdynamic simulations. *Comput. Math. Math. Phys.* **2007**, *47*, 1970–1989. [[CrossRef](#)]
36. Popov, M.; Ustyugov, S. Piecewise parabolic method on a local stencil for ideal magnetohydrodynamics. *Comput. Math. Math. Phys.* **2008**, *48*, 477–499. [[CrossRef](#)]
37. Titarev, V.A.; Toro, E.F. ADER schemes for three-dimensional nonlinear hyperbolic systems. *J. Comput. Phys.* **2005**, *204*, 715–736. [[CrossRef](#)]
38. 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.* **2018**, *39*, 543–551. [[CrossRef](#)]
39. 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.* **2016**, *317*, 318–346. [[CrossRef](#)]
40. Kulikov, I.M.; Chernykh, I.G.; Tutukov, A.V. A New Parallel Intel Xeon Phi Hydrodynamics Code for Massively Parallel Supercomputers. *Lobachevskii J. Math.* **2018**, *39*, 1207–1216. [[CrossRef](#)]
41. Kulikov, I.; Chernykh, I.; Tutukov, A. A New Hydrodynamic Code with Explicit Vectorization Instructions Optimizations that Is Dedicated to the Numerical Simulation of Astrophysical Gas Flow. I. Numerical Method, Tests, and Model Problems. *Astrophys. J. Suppl. Ser.* **2019**, *243*, 4. [[CrossRef](#)]
42. Kulikov, I. Using a Combination of Godunov and Rusanov Solvers based on the Piecewise Parabolic Reconstruction of Primitive Variables for Numerical Simulation of Supernovae Ia Type Explosion. *Lobachevskii J. Math.* **2022**, *in print*.
43. Kriksin, Y.A.; Tishkin, V.F. Variational Entropic Regularization of the Discontinuous Galerkin Method for Gasdynamic Equations. *Math. Model. Comput. Simulations* **2019**, *11*, 1032–1040. [[CrossRef](#)]
44. Tunik, Y.V. Numerical Solution of Test Problems Using a Modified Godunov Scheme. *Comput. Math. Math. Phys.* **2018**, *58*, 1573–1584. [[CrossRef](#)]
45. Kulikov, I.M.; Chernykh, I.G.; Sapetina, A.F.; Lomakin, S.V.; Tutukov, A.V. A New Rusanov-Type Solver with a Local Linear Solution Reconstruction for Numerical Modeling of White Dwarf Mergers by Means Massive Parallel Supercomputers. *Lobachevskii J. Math.* **2020**, *41*, 1485–1491. [[CrossRef](#)]
46. Kulikov, I.M. On a Modification of the Rusanov Solver for the Equations of Special Relativistic Magnetic Hydrodynamics. *J. Appl. Ind. Math.* **2020**, *14*, 524–531. [[CrossRef](#)]
47. Barkov, M.; Bosch-Ramon, V. Relativistic hydrodynamical simulations of the effects of the stellar wind and the orbit on high-mass microquasar jets Get access Arrow. *Mon. Not. R. Astron. Soc.* **2022**, *510*, 3479–3494. [[CrossRef](#)]
48. Reshetova, G.; Cheverda, V.; Khachkova, T. A Comparison of MPI/OpenMP and Coarray Fortran for Digital Rock Physics Application. *Lect. Notes Comput. Sci.* **2019**, *11657*, 232–244.
49. Reshetova, G.; Cheverda, V.; Khachkova, T. Numerical Experiments with Digital Twins of Core Samples for Estimating Effective Elastic Parameters. *Commun. Comput. Inf. Sci.* **2019**, *1129*, 290–301.
50. Reshetova, G.; Cheverda, V.; Koinov, V. Comparative Efficiency Analysis of MPI Blocking and Non-blocking Communications with Coarray Fortran. *Commun. Comput. Inf. Sci.* **2022**, *1510*, 322–336.
51. Wang, Y.; Li, Z. GridFOR: A Domain Specific Language for Parallel Grid-Based Applications. *Int. J. Parallel Program.* **2016**, *44*, 427–448. [[CrossRef](#)]
52. Kataev, N.; Kolganov, A. The experience of using DVM and SAPFOR systems in semi automatic parallelization of an application for 3D modeling in geophysics. *J. Supercomput.* **2019**, *75*, 7833–7843. [[CrossRef](#)]
53. Shterenlikht, A.; Cebamanos, L. MPI vs. Fortran coarrays beyond 100k cores: 3D cellular automata. *Parallel Comput.* **2019**, *84*, 37–49. [[CrossRef](#)]
54. Guo, P.; Wu, J. One-Sided Communication in Coarray Fortran: Performance Tests on TH-1A. *Lect. Notes Comput. Sci.* **2018**, *11337*, 21–33.
55. Landau, L.D.; Lifshitz, E.M. *The Classical Theory of Fields*, 4th ed.; Elsevier: Amsterdam, The Netherlands, 1987; p. 402.
56. Mathews, W. The Hydromagnetic Free Expansion of a Relativistic Gas. *Astrophys. J.* **1971**, *165*, 147–164. [[CrossRef](#)]
57. Perucho, M.; Marti, J.M. A numerical simulation of the evolution and fate of a Fanaroff-Riley type I jet. The case of 3C 31. *Mon. Not. R. Astron. Soc.* **2007**, *382*, 526–542. [[CrossRef](#)]
58. Perucho, M.; Marti, J.M.; Quilis, V. Long-term FR II jet evolution in dense environments. *Mon. Not. R. Astron. Soc.* **2022**, *510*, 2084–2096. [[CrossRef](#)]