PLOS ONE

RESEARCH ARTICLE

A direct Primitive Variable Recovery Scheme for hyperbolic conservative equations: The case of relativistic hydrodynamics

A. Aguayo-Ortiz[1]*, S. Mendoza[1], D. Olvera[2]

1 Instituto de Astronomía, Universidad Nacional Autónoma de México, AP 70-264, Ciudad de México 04510, México, **2** School of Mathematics, University of Bristol, Bristol BS8 1TW, United Kingdom

* aaguayo@astro.unam.mx

OPEN ACCESS

Citation: Aguayo-Ortiz A, Mendoza S, Olvera D (2018) A direct Primitive Variable Recovery Scheme for hyperbolic conservative equations: The case of relativistic hydrodynamics. PLoS ONE 13(4): e0195494. https://doi.org/10.1371/journal.pone.0195494

Editor: Iratxe Puebla, Public Library of Science, UNITED KINGDOM

Received: April 3, 2017

Accepted: March 23, 2018

Published: April 16, 2018

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This work was supported by Direccion General de Asuntos del Personal Academico - Universidad Nacional Autonoma de Mexico (DGAPA-UNAM project number IN112616, http://dgapa.unam.mx) and Consejo Nacional de Ciencia y Tecnologia (CONACyT project ID:CB-2014-1 #240512 http://www.conacyt.gob.mx). AAO, SM and DO acknowledge economic support from

Abstract

In this article we develop a Primitive Variable Recovery Scheme (PVRS) to solve any system of coupled differential conservative equations. This method obtains directly the primitive variables applying the chain rule to the time term of the conservative equations. With this, a traditional finite volume method for the flux is applied in order avoid violation of both, the entropy and "Rankine-Hugoniot" jump conditions. The time evolution is then computed using a forward finite difference scheme. This numerical technique evades the recovery of the primitive vector by solving an algebraic system of equations as it is often used and so, it generalises standard techniques to solve these kind of coupled systems. The article is presented bearing in mind special relativistic hydrodynamic numerical schemes with an added pedagogical view in the appendix section in order to easily comprehend the PVRS. We present the convergence of the method for standard shock-tube problems of special relativistic hydrodynamics and a graphical visualisation of the errors using the fluctuations of the numerical values with respect to exact analytic solutions. The PVRS circumvents the sometimes arduous computation that arises from standard numerical methods techniques, which obtain the desired primitive vector solution through an algebraic polynomial of the charges.

Introduction

The use of numerical methods to solve differential equations has constituted a substantial amount of work since the conception of approximate solutions to a given set of equations. In the last few decades, digital computers have been a great help to heavily iterate complicated partial differential equations using extensive numerical, parallel and adaptive mesh techniques in personal computers and large clusters.

Physical laws are often written in a set of conservative differential equations, for which there are many well established convergent numerical techniques to obtain accurate solutions. In spite of this, there is an intermediate step that is often, depending on the nature of the problem, extremely cumbersome to deal with. This appears since the general solution to the problem is obtained as a set of vector charges $\boldsymbol{q}$ at every point or cell on a given domain of space at

CONACyT (788898, 26344, 255602). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

a particular time in the iteration. However, physical phenomena are described and measured by means of a set of vector primitive variables $\boldsymbol{u}$. Depending on the nature of the physical problem, the function $\boldsymbol{u}(\boldsymbol{q})$ may not have an analytic form and so, at every point or cell of the integration space a cumbersome technique requires to be performed for each time step. No matter how fast this routine may be, it introduces an extra computational time that can heavily grow when the space-time resolution increases. In problems of special relativistic hydrodynamics this fact appears and, at each time step, a 10th degree algebraic polynomial has to be solved for a unique given value of each component of the vector $\boldsymbol{u}$ (see e.g., for an excellent account on this, [1]).

To make things even more complicated, for each particular physical problem it is necessary to have either an analytic solution $\boldsymbol{u}(\boldsymbol{q})$ or a specific numerical technique to obtain it.

In this article we show how it is possible to construct a general numerical iteration method, using a combination of finite differences and finite volume integration techniques for the time and spatial evolutions respectively, to directly find the solutions $\boldsymbol{u}$ avoiding any middle cumbersome step such as the ones mentioned above. This technique is so general that requires no analytical knowledge whatsoever of $\boldsymbol{u}(\boldsymbol{q})$. The method developed is general and valid to any set of coupled conservative equations. We also show how this method can be applied in the particular case of 1D special relativistic hydrodynamics (1DRHD). For this particular case, we construct convergence tests.

The article is organised as follows. In the Appendix, we briefly mention some (mostly used in relativistic hydrodynamics for shock capturing) of the traditional methods to solve a set of conservative equations. In Section 1 we construct our "*Primitive Variable Recovery Scheme* (PVRS)" which can directly obtain the primitive variables from quite a standard numerical procedure. Section 4 deals with different convergence relativistic Sod [2] shock-tube tests and error estimates are given using a standard L_1-norm. Also, the errors are graphically interpreted using the fluctuations of the solution with respect to analytical known values is presented. Finally, in Section 5 we discuss and conclude our results.

1 Primitive Variable Recovery Scheme (PVRS)

In the appendix we discuss some of the standard techniques for discretising any set of scalar and coupled conservative equations. This is done in order to easy understand the further developments of the article for the less expert reader, and not to interrupt the experienced one with such well known methods. However, we note that in the appendix and in what follows Einstein's summation convention will be used throughout the equations displayed in this article, something that does not usually appear in the literature.

The usual way to solve a system of hyperbolic equations (cf. Eq (15)):

$$\frac{\partial \boldsymbol{q}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{q})}{\partial x} = 0, \tag{1}$$

is by implementing Finite Difference and Finite Volume Methods (FDM & FVM) in order to obtain solutions for the conservative charges $\boldsymbol{q}$. In the particular case of relativistic and non-relativistic hydrodynamics, these charges are the linear momentum along the three dimensions S^i, the energy τ and the particle density D. In order to compare the numerical solution with experiments and/or observations, a set of primitive physical measurable variables u needs to be constructed. For this particular case, this primitive variable set is given by by the pressure p, the velocity along three spacial dimensions v^i and, the particle number density n. Some authors prefer to find the particle mass density ρ rather than the particle number density n. For most practical proposes, both variables are related by $\rho = mn$ where m is the average mass per

particle. In here and in what follows all thermodynamical quantities (pressure p, particle number density n and energy density e and so on, are measured on its proper reference frame following the convention in [3, 4]). The explicit dependences $\boldsymbol{q} = \boldsymbol{q}(\boldsymbol{u}(x, t))$ and $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{u}(x, t))$ for 1D flow in the special relativistic case are given by (see e.g. [3, 4]):

$$q_1 = D = \frac{n}{\sqrt{1 - v^2}} \quad \text{and} \quad f_1 = v\frac{n}{\sqrt{1 - v^2}}, \tag{2}$$

$$q_2 = S^x = v\frac{e + p}{1 - v^2} \quad \text{and} \quad f_2 = v^2\frac{e + p}{1 - v^2} + p, \tag{3}$$

$$q_3 = \tau = \frac{e + v^2 p}{1 - v^2} \quad \text{and} \quad f_3 = v\frac{e + p}{1 - v^2}. \tag{4}$$

where e is the total (rest plus internal) proper energy per unit volume which can be related with the density and pressure via a state equation $e = e(n, p)$ like the one derived by Tooper [5] for a polytropic relativistic gas:

$$e = nm + \frac{p}{\kappa - 1}, \tag{5}$$

where κ is the polytropic index. In the previous equations and in what follows we choose a system of units in which the velocity of light is set to unity.

As we can see from Eqs (2–4), obtaining the inverse function $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{q}(x, t))$ results in quite a completed algebraic problem. In fact, the solution to this problem leads to a system of transcendental algebraic equations that have been deeply studied by Riccardi [1]. One way of solving this system is by using a Newton-Raphson method (cf. [6]) but this or any other numerical solution to obtain $\boldsymbol{u}(\boldsymbol{q}(x, t))$ will carry an extra error besides the proper numerical error of the FDM or FVM. This procedure also adds a bit of computational processing time since an iteration loop to find the solution needs to be carried out at each cell every time step. In order to avoid this cumbersome task, we show now how it is possible to obtain a direct numerical solution of the primitive variables, which is valid for all conservative equation systems (cf. Eq (15)).

2 PVRS attempts with finite difference methods

Let us begin by writing the system of m hyperbolic equations showing the explicit dependence on m primitive variables, i.e.:

$$\frac{\partial q_a(u_1, \ldots, u_m)}{\partial t} + \frac{\partial f_a(u_1, \ldots, u_m)}{\partial x} = 0. \tag{6}$$

A necessary and sufficient condition for the existence of the solution $u_1, \ldots, u_m$ is that $a = 1, \ldots, m$. Now, using the chain rule, the above equation can be written in the following quasi-linear form:

$$\frac{\partial q_a}{\partial u_b}\frac{\partial u_b}{\partial t} + \frac{\partial f_a}{\partial u_c}\frac{\partial u_c}{\partial x} = 0, \tag{7}$$

where $\partial\boldsymbol{q}/\partial\boldsymbol{u}$ and $\partial\boldsymbol{f}/\partial\boldsymbol{u}$ are the Jacobian matrixes of the vectors $\boldsymbol{q}$ and $\boldsymbol{f}$ respectively. Multiplying the previous equation by the inverse matrix $(\partial\boldsymbol{q}/\partial\boldsymbol{u})^{-1}$ we get

$$\frac{\partial u_a}{\partial t} + M_{ab}\frac{\partial u_b}{\partial x} = 0, \tag{8}$$

where

$$M_{ab} \coloneqq \left(\frac{\partial q_c}{\partial u_a}\right)^{-1}\left(\frac{\partial f_c}{\partial u_b}\right). \tag{9}$$

If we perform a discretisation of Eq (8) using a FDM (see e.g. Section Finite differences approach of the appendix), we obtain the following numerical expression:

$$u_a(x_i, t_{n+1}) = u_a(x_i, t_n) - \frac{\Delta t}{2\Delta x} M_{ab}[u_b(x_{i+1}, t_n) - u_b(x_{i-1}, t_n)]. \tag{10}$$

No matter how complicated the functional representations of $\boldsymbol{q}(\boldsymbol{u})$ and $\boldsymbol{f}(\boldsymbol{u})$, it is possible (if not by hand, using a Computer Algebra System) to compute the matrix M_{ab} only once before implementing a discretisation scheme. In what follows we show how to implement a numerical scheme to find directly the primitive variables $\boldsymbol{u}$ solving Eq (8). By doing this, the cumbersome step of recovering $\boldsymbol{u}$ from $\boldsymbol{q}$ at every cell for each time step is not needed anymore.

The discretisation Eq (10) is accurate to the first-order and yields quite good results on smooth solutions. When the solution contains a shock wave, the method is stable but not consistent and so no convergent. This could be understood because Eq (10) is mathematically similar to Eq (16) of the appendix [7] with the substitution of the vector $\boldsymbol{u}$ instead of $\boldsymbol{q}$. Furthermore, Eq (10) is written in a non-conservative form and so, the entropy and Rankine-Hugoniot jump conditions are not satisfied across the shock waves. Due to this fact, the obtained solution converges to a different weak solution as compared to the one obtained by a conservative method (see e.g. [8]). In other words, this FDM scheme does not work and the approach to follow is to consider flux contributions as in standard FVM.

3 Primitive Variable Recovery Scheme using combined FDM and FVM

We now show how to implement a Primitive Variable Recovery Scheme (PVRS) using both a FDM and a FVM schemes for the time and spatial evolution of the equations. As mentioned at the end of the previous section, the fluxes contribution in the method must not be altered because the entropy and Rankine-Hugoniot jump conditions must be accomplished. To do so, the spatial derivative term must be evolved using a Godunov-type method (e.g. an HLL-type Riemann solver).

In the appendix it is shown that the conservative set of Eq (1) can be discretised in the form of relation Eq (48), which can be written in a semi-discrete form as:

$$\frac{\partial q_a(x_i)}{\partial t} = -\frac{1}{\Delta x}([F_a^{HLL}]_{i+1/2}^n - [F_a^{HLL}]_{i-1/2}^n), \tag{11}$$

where $\boldsymbol{F}^{HLL}$ stands for the HLL-type Riemann solver approximation for the spatial fluxes (see appendix). Using the chain rule on the left hand side of the previous equation, it follows that:

$$\frac{\partial u_a(x_i)}{\partial t} = -\frac{\mathcal{A}_{ab}(x_i, t_n)}{\Delta x}([F_a^{HLL}]_{i+1/2}^n - [F_a^{HLL}]_{i-1/2}^n). \tag{12}$$

where $\mathcal{A} = (\partial q/\partial u)^{-1}$. By applying a forward-difference formula scheme on the left hand side of Eq (12), we get

$$u_a(x_i, t_{n+1}) = u_a(x_i, t_n) - \frac{\Delta t}{\Delta x}\mathcal{A}_{ab}(x_i, t_n)([F_b^{HLL}]_{i+1/2}^n - [F_b^{HLL}]_{i-1/2}^n). \tag{13}$$

In Eq (13), we take a numerical flux approach as in standard FVM and a finite difference of the time derivative over the primitive variables $\boldsymbol{u}$. The approximate solution to the Riemann problem, where Rankine-Hugoniot's condition take place, is the same as the one presented in the appendix HLL Riemann solver section. Furthermore, the characteristic velocities used in the HLL solver which correspond to the the eigenvalues of the Jacobian $\partial \boldsymbol{f}/\partial \boldsymbol{q}$, can be computed either from matrix M_{ab} Eq (9) or from $\partial f_a/\partial q_b$ since both matrixes are similar [7]. All matrixes and vectors $(M_{ab}, \mathcal{A}_{ab}, f_a, q_a)$ are computed using a piecewise reconstruction $\tilde{u}$ of the primitive variables, except for matrix A_{ab} which is evaluated on the midpoint x_i of the cell C_i.

It is important to note that in Eqs (8) and (13) the second term on the right hand side has an implicit sum over the repeated index a.

Note that, although it seems that the PVRS discretisation Eq (13) arises directly from discretising the hybrid quasilinear equation $\partial \boldsymbol{u}/\partial t + \mathcal{A}\,\partial \boldsymbol{f}/\partial x = 0$ –which can be directly obtained by using the chain rule on Eq (1), it is impossible to obtain the PVRS discretisation shown in Eq (13) using a standard conservative FVM as presented in the appendix, and which satisfies the entropy and Rankine-Hugoniot jump conditions.

By using Eq (13) on a numerical code, it would no longer be a concern to recover the primitive variables from the computed conservative charges; they would instead be solved directly! Therefore, it would not be necessary to create a module in the code to obtain the final required solution $\boldsymbol{u}(x, t)$. In general terms, this procedure works out for any kind of conservative system in which $\boldsymbol{q}(\boldsymbol{u}(x, t))$ and $\boldsymbol{f}(\boldsymbol{u}(x, t))$ are at least given at some initial time.

The time step evolution of the Eq (13) that we use for our numerical simulations is given by the Method of Lines (MoL):

$$\frac{\partial \boldsymbol{u}(x_i)}{\partial t} = -\boldsymbol{L}(\boldsymbol{u}(x_i)), \tag{14}$$

where $\boldsymbol{L}(\boldsymbol{u}(x_i))$ is the right hand side of Eq (13) (see e.g. [9]), which can be further implemented with a Runge-Kutta integration.

4 Convergence test for PVRS in relativistic hydrodynamics

In this section we are going to show how this new method handles the evolution of a relativistic gas in a particular Riemann problem namely the *shock tube* (see e.g. [9]). This relativistic Sod [2] shock tube problem is a standard test that any code must fulfil for its validation. It has an exact analytical solution for both special relativistic and non-relativistic hydrodynamics and it is used for comparisons with numerical methods.

We calculated the numerical solution using PVRS discretisation Eq (13) with an approximate HLL Riemann solver, a *minmod* limiter for the reconstruction $\tilde{u}$ and a 4th order Runge-Kutta Method of Lines (MoL-RK4) for the integration. The problem was solved in the domain [0, 1] with $N = 800$ identical grid cells. We made three relativistic Sod tests with the initial discontinuity located at $x = 0.5$ and with initial states shown on Table 1. Furthermore, we compared the numerical results with the exact solution [9]. Also, we have estimated the usual L_1-norm error for the following different resolutions: $\Delta x_1 = 1/200$, $\Delta x_2 = 1/400$, $\Delta x_3 = 1/800$, $\Delta x_4 = 1/1600$, $\Delta x_5 = 3200$ and $\Delta x_6 = 1/6400$.

The time-step condition used in this method is different from the commonly used by many authors (cf. [10]). A general CFL-condition applied to this numerical scheme was constructed by us and used in the set of examples presented. The exact condition and its derivation is a subject beyond the scope of this article and will be published elsewhere. For practical purposes, the time step interval can be chosen as a sufficiently smaller number than the corresponding

Table 1. Initial parameters used for the relativistic Sod [2] shock tube tests described in the article. κ stands for the polytropic index.

Test	p_L	v_L	n_L	p_R	v_R	n_L	κ
1	1.0	0.0	1.0	0.1	0.0	0.125	4/3
2	13.33	0.0	10.00	0.1	0.0	1.0	4/3
3	1000	0.0	1.0	0.01	0.0	1.0	5/3

https://doi.org/10.1371/journal.pone.0195494.t001

CFL condition (cf. Eq (40)). For the examples presented below, we have chosen a fixed time step for each simulation.

4.0.1 Test 1: **Weak relativistic blast wave**

The first test corresponds to a lowly relativistic blast wave explosion. The results can be seen in Fig 1, where we compare the numerical solution (points) with the exact solution (lines). It is clear that for both, smooth parts and discontinuities, the numerical solution converges quite well to the exact one.

4.0.2 Test 2: **Mildly relativistic blast wave**

The second test corresponds to a mildly relativistic blast wave explosion. The results can be seen in Fig 2, where we compare the numerical solution (points) with the exact one (lines). The importance of this test is to see if, with a relative high difference in pressure between both states, the numerical method is capable of solving the density function at the contact discontinuity.

4.0.3 Test 3: **Strong relativistic blast wave**

Finally, the last test corresponds to a strongly relativistic blast wave explosion. In this case, the density discontinuity is produced by a a 5 orders of magnitude difference between right and left initial detonation pressure, creating a thin shell which numerically is harder to resolve at low resolutions. However, with a relatively small number of cells and a weak variable reconstruction, the results shown on Fig 3 are as good as the ones obtained by other codes (cf. [10, 11]).

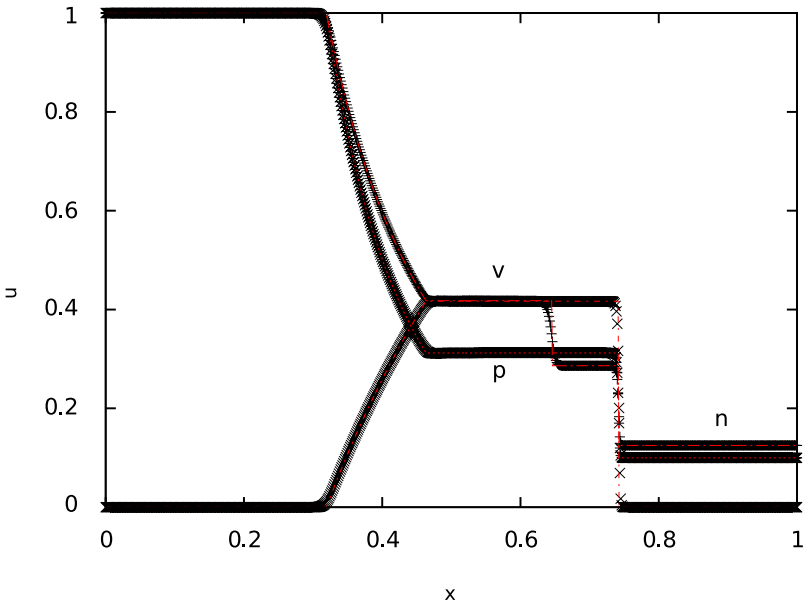

Fig 1. Test 1. The figure shows the result of the simulation of a weak relativistic (Sod shock tube) blast wave explosion at t = 0.35 for the particle number density n, pressure p and velocity v. The time step used for the simulation was 0.001.

https://doi.org/10.1371/journal.pone.0195494.g001

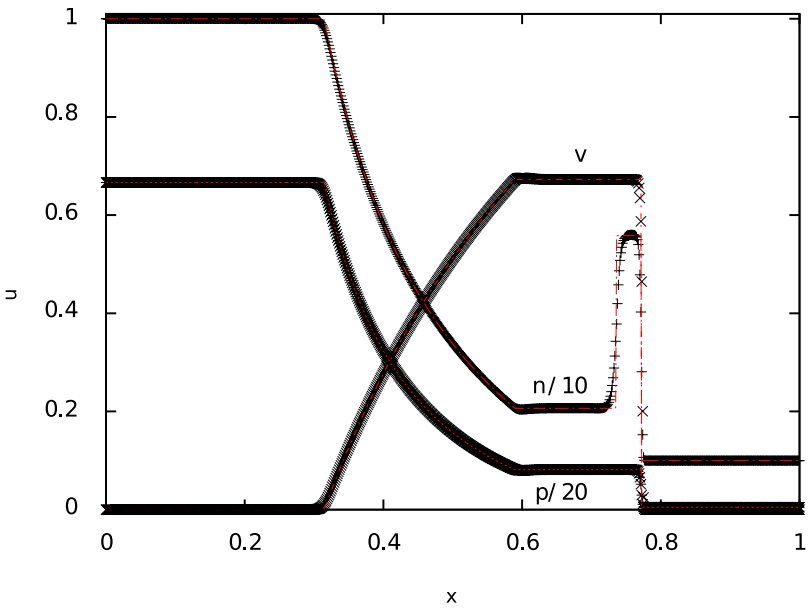

Fig 2. Test 2. The figure shows the result of a mildly relativistic (Sod shock tube) blast wave explosion at t = 0.35 for particle number density n, pressure p and velocity v. The time step used for the simulation was 0.001.

https://doi.org/10.1371/journal.pone.0195494.g002

4.1 Error estimates

We have calculated the error of each test using the traditional L_1—norm value. The convergence order of this test is given by $\log(error_i/error_{i-1})/log(1/2)$, where $error_j$ is the L_1—norm of the Δx_j resolution. As we can see from Table 2, the error decreases when the resolution increases, as expected. Also, we obtain first order convergence for all test in at least one

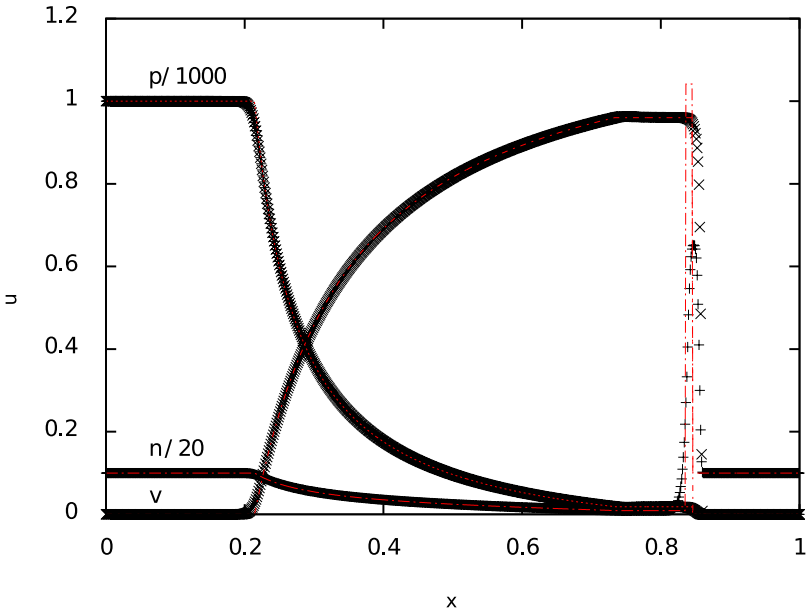

Fig 3. Test 3. The figure shows the result of a strong relativistic (Sod shock tube) blast wave explosion at t = 0.35 for particle number density n, pressure p and velocity v. The time step used for the simulation was 0.0001.

https://doi.org/10.1371/journal.pone.0195494.g003

Table 2. The L_1-norm for the error in the numerical density for the *minmod* limiter with different numerical resolutions. The L_1-norm is computed for all shock-tube and Gaussian tests at time $t = 0.35$. We also show the order of convergence between different resolutions. Since the error decreases when the resolution increases, the PVRS constructed in the article is stable and converges to the exact solution. The data of this is presented in S1 File.

	Error				Order of Convergence			
Resolution	**Test 1**	**Test 2**	**Test 3**	**Smooth**	**Test 1**	**Test 2**	**Test 3**	**Smooth**
Δx_1	3.83e-3	9.00e-2	1.93e-1	4.74e-4	-	-	-	-
Δx_2	2.12e-3	5.04e-2	1.60e-1	1.28e-4	0.85	0.84	0.27	1.89
Δx_3	1.21e-3	2.61e-2	1.21e-1	0.34e-4	0.81	0.95	0.40	1.91
Δx_4	6.68e-4	1.51e-2	8.03e-2	0.09e-4	0.85	0.79	0.59	1.92
Δx_5	4.01e-4	1.02e-2	4.56e-2	0.02e-4	0.74	0.57	0.81	2.17
Δx_6	2.22e-4	5.07e-3	2.62e-2	-	0.83	1.00	1.05	-

https://doi.org/10.1371/journal.pone.0195494.t002

resolution. Additionally, we made an experiment following [6] of a static Gaussian curve in order to estimate the order of convergence of a smooth static profile which, for this case, reaches a convergence value of about 2 in all the tested resolutions for a fixed time step of 0.01. As expected, this means that the important error of the relativistic Sod shock tube test relays on the discontinuities. This is the reason as to why we consider that taking the L_1—norm is not a clear indicator of the "real" error at the shock waves, so we propose a more relevant useful visual interpretation of this estimation as follows.

In Fig 4 we show both exact (red dashed-line) and numerical (blue dashed-line) solution vs. the fluctuation $|u_{\text{num}} - u_{\text{exact}}|/u_{\text{exact}}$ at each point (black line), for the density in Test 3 at every resolution. We can see how the Full Width at Half Maximum (FWHM) of the fluctuation

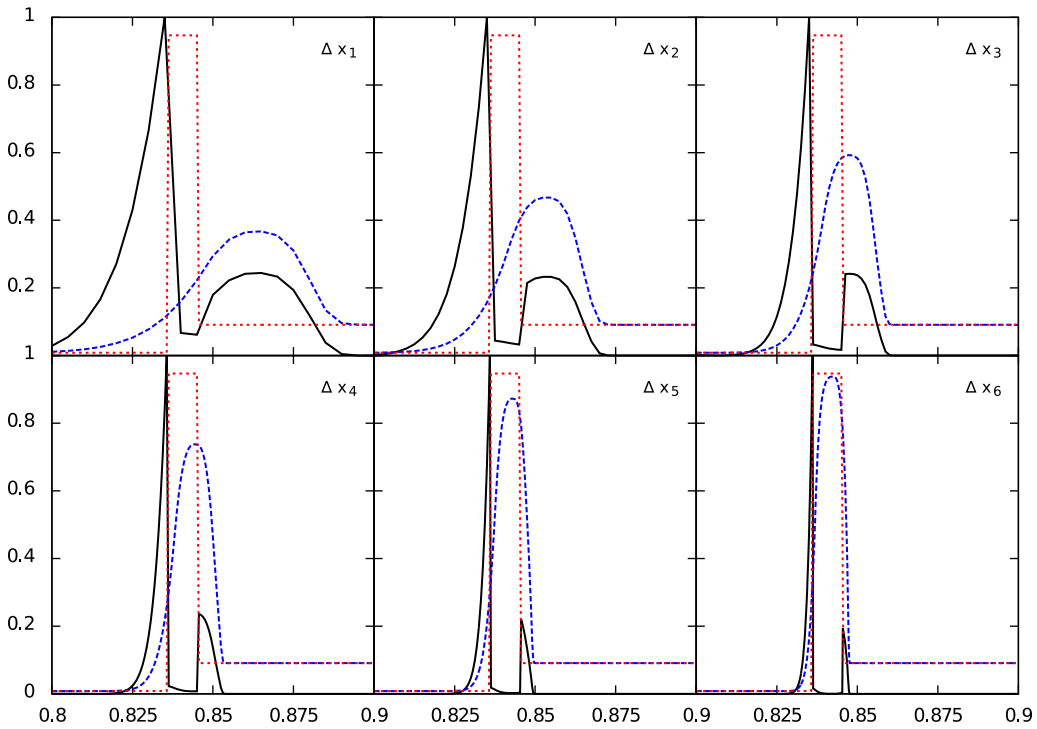

Fig 4. Comparison between the exact (red dashed-line) and the numerical solution (blue dashed-line) of the contact discontinuity in density for the Test 3 vs. the fluctuation $|u_{\text{num}} - u_{\text{exact}}|/u_{\text{exact}}$ (black continuous line) at each point, for all the tested resolutions. Note that as the resolution increases, the width of the fluctuation decreases, showing the convergence in a straightforward manner.

https://doi.org/10.1371/journal.pone.0195494.g004

tends to zero as the resolution increases. Working with the fluctuation of the numerical solution about the exact solution is a much better way to easily see the convergence of a numerical method, rather than the traditional L_1-norm for which smoothing of the errors can be wrongly interpreted as a positive convergence test.

5 Discussion

In this article we have developed a new numerical algorithm to solve any set of coupled differential conservative equations for which the primitive variable vector $\boldsymbol{u}$ is directly obtained. This is a forward step in numerical methods, since it avoids any intermediate step reconstruction of the primitive variable vector from a previously obtained charge vector $\boldsymbol{q}$ at all points or cells in space at each time. In principle, this means that numerical codes can be written in a more direct form. Also, depending on the nature of the physical problem to solve, the computational time may be reduced with this technique.

For practical purposes, we always had in mind special relativistic hydrodynamical problems and for this reason the specific techniques used throughout the article deal with hydrodynamical shock capturing schemes. We demonstrated in the article that the Primitive Variable Recovery Scheme (PVRS) showed good convergence for three shock-tube and one Gaussian tests. Further explorations in other directions, such as a non-static Gaussian test [12] need to be investigated. We will explore more details in future works.

The PVRS presented in this article can be implemented straightforward to any standard hydrodynamical code that already uses HLL Riemann solvers given by Eq (13).

In summary, the PVRS is a numerical maneuver to circumvent the embroiling construction of the primitive vector once the charge vector is obtained from any standard procedure used to solve a set of coupled conservative equations in physical systems.

We are constructing a GNU Public Licensed (GPL) free software (http://www.gnu.org) called "*aztekas*" (http://www.aztekas.org) that deals with relativistic hydrodynamics using this PVRS technique.

Appendix

Traditional approach for numerically solving conservative equations

In this appendix, we deal with traditional well known methods for solving conservative equations. Our intention is to briefly introduce the less versed reader to this topics using Einstein's summation convention.

A system of m conservative equations in one dimension is usually written as:

$$\frac{\partial q_a}{\partial t} + \frac{\partial f_a(q_1, \ldots, q_m)}{\partial x} = 0, \tag{15}$$

where the subindex a takes values from 1 to m, $\boldsymbol{q} \coloneqq \boldsymbol{q}(\boldsymbol{u}(x, t))$ is the vector of *conservative charges* and $\boldsymbol{f} \coloneqq \boldsymbol{f}(\boldsymbol{q}(\boldsymbol{u}(x, t)))$ is the corresponding *flux* vector along the x axis at a given time t. The vector $\boldsymbol{u}$ corresponds to the *primitive variables* for which its number of entries and functional form of $\boldsymbol{q}(\boldsymbol{u})$ depends on the particular problem to solve. From this point onwards, we are going to use $\boldsymbol{f}(x, t)$ instead of the cumbersome notation $\boldsymbol{f}(\boldsymbol{q}(\boldsymbol{u}(x, t)))$, bearing in mind that both, charges and flux vectors, depend on the primitive variables $\boldsymbol{u}(x, t)$. As it is shown in section 1, the fluxes also have an explicit dependence on the primitive variables but are usually written in terms of the conservative charges.

We can rewrite Eq (15) in the *quasilinear* following form

$$\frac{\partial q_a}{\partial t} + J_{ab}\frac{\partial q_b}{\partial x} = 0, \tag{16}$$

where J_{ab} is the *Jacobian* matrix of $\boldsymbol{f}(\boldsymbol{q})$. From now on, we use Einstein implicit sum convention over two repeated subindexes contained in the set $\{a, b, c, d\}$. If the Jacobian matrix satisfies the conditions of having real eigenvalues and a set of independent eigenvectors, then we say that the system Eq (15) is *hyperbolic* (see e.g. [8]).

In the linear cases (when $\boldsymbol{f}$ is a linear function of $\boldsymbol{q}$), there exists an analytical solution for (15), but many physical cases give rise to nonlinear conservative systems which are required to be solved using numerical methods.

In the following subsections we briefly mention two of the main numerical methods used to solve 1D conservative systems such as the one written in Eq (15).

Finite differences approach

The finite differences method (FDM) is one of the most useful and simple numerical methods for solving ordinary and partial differential equations. It consists of an approximation of the derivatives of fluxes and charges based on approximations of their values on sufficiently small intervals of space and time. The space is divided in a grid of N centred points spaced by equal length Δx intervals in which the equation is evaluated.

Using Taylor expansions of the involved quantities, it is possible to work out the finite difference form of Eq (15) to find the value of $\boldsymbol{q}$ in all the grid at time $t + \Delta t =: t_{n+1}$ based on its value at $t =: t_n$:

$$q_a(x_i, t_{n+1}) = q_a(x_i, t_n) - \frac{\Delta t}{2\Delta x}[f_a(x_{i+1}, t_n) - f_a(x_{i-1}, t_n)], \tag{17}$$

where x_i is the i-th point on the grid. This is the Forward Time Central Space (FTCS) Euler method [8]. In Eq (17), the derivative $\partial f_a/\partial x$ at a given time t_n was written using a central approximation value given by $(f_a(x_{i+1}) - f_a(x_{i-1}))/(2\Delta x)$. For the left and right boundary points this derivative can be written using a right or left derivative approximation given by: $(f_a(x_1) - f_a(x_0))/\Delta x$ and $(f_a(x_{N-1}) - f_a(x_N))/\Delta x$ respectively. Unfortunately, Eq (17) leads to numerical unstable solutions [13]. To overcome this problem, many higher order methods have been developed and successfully implemented over time [14].

When a second-order finite differences approximation method is used, additional source *artificial viscosity* terms appear in Eq (17). Those additional terms are either due to the second derivative approximation in Taylor series or to second differences approximation of the first derivatives (see e.g. [14]). The artificial viscosity name was given by von Neumann [13] since it resembles the viscosity term of the Navier-Stokes equation, but has nothing to do with any physical viscosity.

The general form of the artificial viscosity can be written as [14]:

$$\begin{aligned} q_a(x_i, t_{n+1}) &= q_a(x_i, t_n) \\ &- \frac{\Delta t}{2\Delta x}[f_a(x_{i+1}, t_n) - f_a(x_{i-1}, t_n)] \\ &+ \frac{\Delta t}{2\Delta x}[\epsilon_a^+ \Delta q_a^+(x_i, t_n) - \epsilon_a^- \Delta q_a^-(x_i, t_n)], \end{aligned} \tag{18}$$

where $\epsilon_a^{\pm}$ are the *coefficients of second-order explicit artificial viscosity* and

$\Delta q_a^{\pm}(x_i, t_n) = \pm q(x_{i\pm1}, t_n) \mp q(x_i, t_n)$. The choice $\epsilon_a^{\pm} = 2\Delta x/\Delta t$ simplifies the above equation to:

$$\begin{aligned} q_a(x_i, t_{n+1}) = &\frac{1}{2}(q_a(x_{i+1}, t_n) + q_a(x_{i-1}, t_n)) \\ &- \frac{\Delta t}{2\Delta x}[f_a(x_{i+1}, t_n) - f_a(x_{i-1}, t_n)], \end{aligned} \tag{19}$$

which is known as the *Lax-Friedrich method*. Other second-order-two-step methods, such as the *Lax-Wendroff method*, have been developed and successfully implemented in many numerical codes.

One such favourite two-step method was proposed by MacCormack [15]. It makes a *forward-prediction* of $\boldsymbol{q}$ and with it, a *backward-correction*:

$$\tilde{q}_a(x_i, t_n) \coloneqq q_a(x_i, t_n) - \frac{\Delta t}{\Delta x}[f_a(x_{i+1}, t_n) - f_a(x_i, t_n)], \tag{20}$$

$$q_a(x_i, t_{n+1}) = \frac{1}{2}\left\{ q_a(x_i, t_n) + \tilde{q}_a(x_i, t_n) - \frac{\Delta t}{\Delta x}[\tilde{f}_a(x_i, t_n) - \tilde{f}_a(x_{i-1}, t_n)] \right\}. \tag{21}$$

where $\tilde{\boldsymbol{f}} \coloneqq \boldsymbol{f}(\tilde{\boldsymbol{q}})$. This method has been proved to be consistent, convergent and stable which is the requirement for any numerical method used in a computational code. Nevertheless, in discontinuities and regions with high pressure gradients, such as regions with shock-waves, this algorithm introduces a dispersive error called the Gibbs phenomenon, which consists on the presence of large spurious oscillations near the finite-jump, such as the example shown in Fig 5.

To solve this problem, it is common to apply a *corrective diffusion* in the regions where the non-physical oscillations appear. The correction presented by Book [16] is

$$q_a^*(x_i, t_n) = q_a(x_i, t_n) + \eta[q_a(x_{i+1}, t_n) - 2q_a(x_i, t_n) + q_a(x_{i-1}, t_n)], \tag{22}$$

where η is the *antidiffusion coefficient* at space-time points x_i and t_n:

$$\eta = \begin{cases} \eta_0 \leq 1/4, & \text{if} \quad (\Delta q_a^+)(\Delta q_a^-) < 0, \\ 0, & \text{if} \quad (\Delta q_a^+)(\Delta q_a^-) > 0. \end{cases} \tag{23}$$

Finite volume approach

A more natural way of obtaining the discretisation form of (15) is the Finite Volume Method (FVM) which is based on a subdivision of the spatial domain into intervals (also called *control volumes* or *grid cells*) $C_i \coloneqq [x_{i-1/2}, x_{i+1/2}]$. The integration of Eq (15) over C_i between times t_n and t_{n+1} yields:

$$\int_{t_n}^{t_{n+1}} \int_{C_i} \left[\frac{\partial q_a(x,t)}{\partial t} + \frac{\partial f_a(x,t)}{\partial x} \right] \mathrm{d}x\, \mathrm{d}t = 0. \tag{24}$$

The integral of $\partial_t q$ over time and the integral of $\partial_x f$ over space can be solved exactly and so,

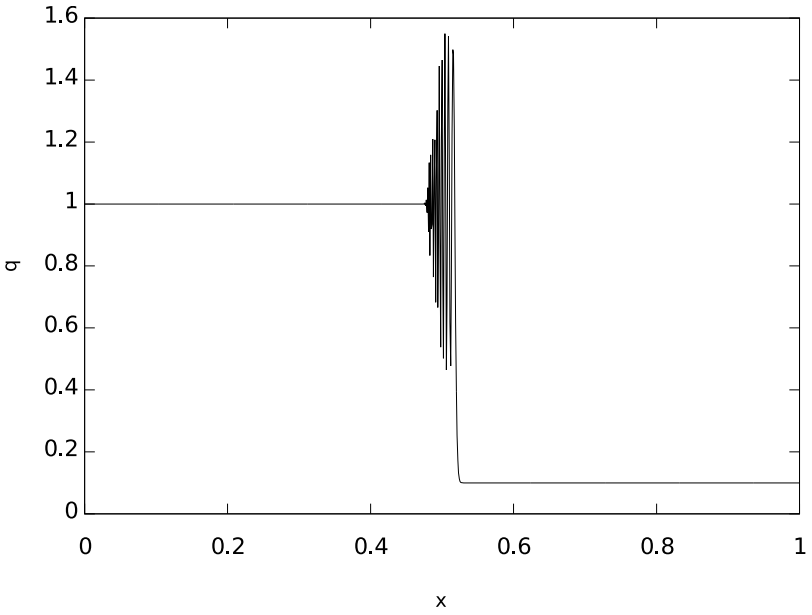

Fig 5. The graph shows the numerical solution of the advection equation: $\partial_t \boldsymbol{q} + \partial_x \boldsymbol{q} = \mathbf{0}$, using exclusively the MacCormack method. The solution shows non-physical oscillations in a finite-jump discontinuity due to the Gibbs phenomenon. At later times, the oscillations grow breaking even more the expected solution. The graph was constructed using the initial conditions of $q = 1.0$ if $x < 0.5$ and $q = 0.125$ elsewhere at a fixed time $t = 0.03$.

https://doi.org/10.1371/journal.pone.0195494.g005

the next integral form of the previous equation is found:

$$\int_{C_i} (q_a(x, t_{n+1}) - q_a(x, t_n))\mathrm{d}x + \int_{t_n}^{t_{n+1}} (f_a(x_{i+1/2}, t) - f_a(x_{i-1/2}, t))\mathrm{d}t = 0. \tag{25}$$

At this point, both integrals in the previous equation cannot be integrated unless we have the exact form of q, which is precisely the solution to the problem. In order to overcome this, we define each integration as a new numerical vector in the following form:

$$[Q_a]_i^n = \frac{1}{\Delta x} \int_{C_i} q_a(x, t_n)\mathrm{d}x, \tag{26}$$

$$[F_a]_{i\pm1/2}^n = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} f_a(x_{i\pm1/2}, t)\mathrm{d}t, \tag{27}$$

where $[Q_a]_i^n$ is the *average charge vector* of q over C_i at time t_n and $[F_a]_{i\pm1/2}^n$ is the *average flux vector* across the boundaries of C_i. From now on, the square brackets notation [ ] around any numerical function is used to denote the corresponding (space or time) average related to that specific numerical function.

If $\boldsymbol{q}(\boldsymbol{u}(x, t))$ is a smooth function, then the integral Eq (26) agrees with the value of $\boldsymbol{q}$ at the midpoint of the interval to $\mathcal{O}(\Delta x^2)$ [8].

The indexes outside the square bracket do not denote the spatial and time evaluation of the average vector, they are just labels that refer to the time and grid positions of the corresponding numerical values.

Substituting the definitions Eqs (26) and (27) in Eq (25) we obtain the main discretisation for the finite volume scheme usually presented in the literature (cf. [8]):

$$[Q_a]_i^{n+1} = [Q_a]_i^n - \frac{\Delta t}{\Delta x}\left([F_a]_{i+1/2}^n - [F_a]_{i-1/2}^n\right). \tag{28}$$

Eq (28) is a numerical recipe of how to compute the mean value $[Q_a]_i^{n+1}$ using the average flux and charge values one time-step backwards for each grid cell C_i. This discretisation has the same exact form as Eq (15) except for the choice of the values Eqs (26) and (27).

The advantage of this method over any finite difference scheme is that the conservative nature of the system is preserved, even across strong discontinuities such as shock waves. This is the reason as to why a finite volume scheme is often used when dealing with the physics of high energy flows where discontinuities may appear.

Numerical flux

The flux $\boldsymbol{f}$ at Eq (27) depends on the value of q at every time. This is why it is impossible to integrate the average flux. Somehow, we have to find a good approximation for this integral. Moreover, the flux $\boldsymbol{f}$ inside the integral is evaluated on the boundaries $x_{i\pm1/2}$ of the grid cell which, numerically speaking, has no sense because we can only approximate the values of the average charges on the midpoint of the finite volume. This set of midpoints can be "safely" considered the ones used in the finite difference mesh mentioned in the Finite differences approach section.

One way to approximate $[F_a]_{i\pm1/2}^n$ is to assume that it can be obtained as a function of the cell average values of $\boldsymbol{q}$ on either side of the interface $x_{i\pm1/2}$, i.e., $[Q_a]_{i\pm1}^n$ and $[Q_a]_i^n$:

$$[F_a]_{i\pm1/2}^n = \mathcal{F}_a([Q_a]_{i\pm1}^n, [Q_a]_i^n). \tag{29}$$

The previous result is expected since in a hyperbolic problem the information of how $\boldsymbol{q}$ change on every cell propagates at a finite characteristic speed (see e.g. [3, 14]). The function $\mathcal{F}_a$ can be thought as a *numerical flux function* for which its functional form will depend on the problem or the particular numerical scheme used to solve it.

Substitution of Eq (29) into Eq (28) yields:

$$[Q_a]_i^{n+1} = [Q_a]_i^n - \frac{\Delta t}{\Delta x}[\mathcal{F}_a([Q_a]_{i+1}^n, [Q_a]_i^n) - \mathcal{F}_a([Q_a]_{i-1}^n, [Q_a]_i^n)]. \tag{30}$$

The numerical flux function is then determined by the evolution of the solution in each interface. A good first guess for the function $\mathcal{F}_a$ is to relate it to the corresponding average flux function of a local (for each cell) Riemann problem [9] with two constant states on each side of the boundary.

In order to obtain an accurate numerical flux function, is important to study the behaviour of the solution based on the form and properties of the governing equation at these particular initial conditions.

Riemann problem

Let us now consider a single conservative equation (i.e. Eq (15) with $a = 1$ only) in which the flux is written as $f(q) = \tilde{u}q$ where $\tilde{u}$ is a constant value:

$$\frac{\partial q}{\partial t} + \tilde{u}\frac{\partial q}{\partial x} = 0. \tag{31}$$

This is the advection equation in which $\tilde{u}$ corresponds to the propagation velocity of q. Note that, since $f'(q) = \tilde{u}$, Eq (31) is also its own quasilinear version.

The function $q(x,t) = \tilde{q}(x - \tilde{u}t)$ satisfies Eq (31) for any function $\tilde{q}$. However, it is more useful for us to describe the problem observing the behaviour of the solution q along *characteristic curves* in the $t - x$ plane. To do so, we perform the time derivative of $\boldsymbol{q}(X(t), t)$ and equate the result to zero, i.e.:

$$\frac{\mathrm{d}}{\mathrm{d}t} q(X(t), t) = \frac{\partial q}{\partial t} + X'(t)\frac{\partial q}{\partial x} = 0. \tag{32}$$

Direct comparison of the above equation with Eq (31), means that that the solution $q(X(t), t)$ is constant all along the ray $X(t) = x_0 + \tilde{u}t$, where x_0 is some initial value. In the most general case, the set of all rays $X(t)$ are called the *characteristics* of the equation.

If we consider the particular case in which the initial conditions of the problem consists on two constant states

$$q(x,0) = \begin{cases} q_l, & \text{if} \quad x < 0, \\ q_r, & \text{if} \quad x > 0, \end{cases} \tag{33}$$

where q_l and q_r are the left and right states respectively, the characteristics $X(t)$ of (31) are then rays with slope $\tilde{u}$ in the $t - x$ plane. With this, the solution can be written as

$$q(x,t) = \begin{cases} q_l, & \text{if} \quad x - \tilde{u}t < 0 \quad \text{or} \quad x/t < \tilde{u}, \\ q_r, & \text{if} \quad x - \tilde{u}t > 0 \quad \text{or} \quad x/t > \tilde{u}. \end{cases} \tag{34}$$

Let us consider now a system of m conservative equations (i.e. $a = 1, 2, \ldots, m$ in Eq (15)), where $f_a = A_{ab}q_b$, i.e.:

$$\frac{\partial q_a}{\partial t} + A_{ab}\frac{\partial q_b}{\partial x} = 0, \tag{35}$$

where A_{ab} is a constant $m \times m$ matrix and so, the system of conservative equations is linear. If A_{ab} is diagonalisable such that:

$$A_{ab} = R_{ac}\Lambda_{cd}R^{-1}_{db}, \tag{36}$$

where R_{ac} is the matrix of eigenvectors, with r^p_a the p-th eigenvector, R^{-1}_{db} its inverse and $\Lambda_{cd} = \mathrm{diag}(\lambda^1, \ldots, \lambda^m)$, for λ^p the p-th eigenvalue. If we define the *characteristic variables* w_a as

$$w_a(x,t) \coloneqq R^{-1}_{ab}q_b(x,t), \tag{37}$$

it is then possible to rewrite Eq (35) as the following system of m advective equations:

$$\frac{\partial w_a}{\partial t} + \Lambda_{ab}\frac{\partial w_b}{\partial x} = 0. \tag{38}$$

In the case of the Riemann problem, the solution for the p-th advective equation is $w_p(x,t) = \tilde{w}_p(x - \lambda^p t, 0)$, and the solution $q_a(x,t)$ is obtained using the definition of w_a:

$$q_a(x,t) = R_{ab}\tilde{w}_b(x,t). \tag{39}$$

In this way one can think that q_a is a superposition of m waves moving with *characteristic velocities* $\lambda^1, \lambda^2, \ldots$ and λ^m, respectively [14].

Another way to see this is by comparing Eq (35) with the time derivative of $q(X(t), t)$ in (32). From this, it follows that the characteristics are curves for which their corresponding slopes are exactly the eigenvalues of the matrix A_{ab}.

In order to obtain a real contribution of one of these waves to the evolution of a contiguous grid cell, the size of the control volume must be larger than the distance travelled by the wave, moving at its characteristic velocity, at a certain fixed time Δt, i.e.,

$$\lambda \frac{\Delta t}{\Delta x} < 1. \tag{40}$$

The quantity $\lambda \Delta t/\Delta x$ is know as the *Courant number* and the fulfilment of Eq (40) is called *Courant-Friedrich-Levy (CFL) condition*. This is a convergence requirement for several numerical methods that solve conservative equations.

The Riemann problem discussed in this subsection, is used to accurately estimate the value of the numerical fluxes at the boundaries of two contiguous grid cells as will be seen in the following section.

Godunov scheme

Godunov in 1959 [17] proposed a numerical scheme for solving conservative equations and this method can be used in terms of the Riemann problem as follows. Consider the single Eq (31). The algorithm proposed by Godunov has the following recipe:

1. Compute the average values of the charges q at the time $t = t_n$ using Eq (26) for $a = 1$ only:

$$[Q]_i^n = \frac{1}{\Delta x} \int_{C_i} q(x, t_n) \mathrm{d}x. \tag{41}$$

2. Reconstruct from $[Q]_i^n$ a polynomial function $\tilde{q}(x, t_n)$ for every value of x. The simplest case for this is to take a constant function:

$$\tilde{q}(x, t_n) \coloneqq [Q]_i^n \quad \text{for} \quad x \in C_i. \tag{42}$$

   In practice [18], the value $[Q]_i^n$ is consider to be q evaluated at the midpoint of the grid cell.

3. Evolve the hyperbolic equation in an exact or approximate way by a time Δt to obtain $\tilde{q}(x, t_{n+1})$.

4. Take the average of $\tilde{q}(x, t_{n+1})$ over C_i to obtain $[Q]_i^{n+1}$.

5. Go back to the first item on the list and iterate until a final time is reached.

As we discuss above, it is impossible to compute exactly the average flux $[F]_{i\pm1/2}^n$ because we do not know the value of q at all times. However, if we consider a Riemann problem in the interface $x_{i\pm1/2}$ between the grid cells C_i and $C_{i\pm1}$ and apply step 3 of Godunov's algorithm, we get that $\tilde{q}(x_{i\pm1/2}, t)$ is constant along the curves that satisfies $(x - x_{i\pm1/2})/t = \text{const}$.

In summary, if we denote by $q^{\downarrow}([Q]_i^n, [Q]_{i\pm1}^n)$ the solution to the Riemann problem at $x_{i\pm1/2}$, the computation of the average fluxes reduces on computing an integral over a constant function [8]. In this way, the Godunov's algorithm can be expressed in terms of average fluxes using the following recipe:

1. Solve the Riemann problem in the interfaces $x_{i\pm1/2}$ of the C_i grid cell in order to obtain $q^{\downarrow}([Q]_i^n, [Q]_{i\pm1}^n)$.

2. Define $\mathcal{F}([Q]_i^n, [Q]_{i\pm1}^n) = f(q^{\downarrow}([Q]_i^n, [Q]_{i\pm1}^n))$.
3. Apply discretisation Eq (30).

The problem with applying Godunov's scheme on non-linear systems and considering wave propagation of characteristic waves on all interfaces, is that the characteristic velocities are not constant at all times and also they change values at different grid cells. For the case of a quasilinear system such as the one of Eq (16), an approximation has to be made. Many methods for obtaining an approximate Riemann solution have been developed and successfully implemented in classical and relativistic magnetohydrodynamic codes (see e.g. [11, 19]).

HLL Riemann solver

One of the most popular approximate Riemann solvers is the called HLL solver [20]. This Godunov's base method considers a Riemann problem with constant states $\boldsymbol{q}^L$ and $\boldsymbol{q}^R$ on each side of the interface in a space-time grid cell $[x_L, x_R] \times [0, T]$ as shown on Fig 6.

Instead of following the solution of all the characteristic variables along their own characteristic velocities, the idea of the HLL approximation consists on considering the larger eigenvalues λ_R and λ_L moving across the interface to the right and left respectively. The region delimited by these characteristic rays is denoted by the state $\boldsymbol{q}^{HLL}$.

Note that, since we are working with a system of m conservative equations, $2m$ characteristic rays will emerge from each interface. The values λ_L and λ_R are to be chosen taking into account all $2m$ characteristic velocities.

The approximate solution to the Riemann problem derived by this scheme has the following form (see e.g. [8] or [21]):

$$q_a(x,t) = \begin{cases} q_a^L, & \text{if} \quad x/t \leq \lambda_L, \\ q_a HLL & \text{if} \quad \lambda_L < x/t < \lambda_R, \\ q_a^R, & \text{if} \quad x/t \geq \lambda_R, \end{cases} \tag{43}$$

where

$$q_a^{HLL} = \frac{\lambda_R q_a^R - \lambda_L q_a^L + f_a^L - f_a^R}{\lambda_R - \lambda_L}, \tag{44}$$

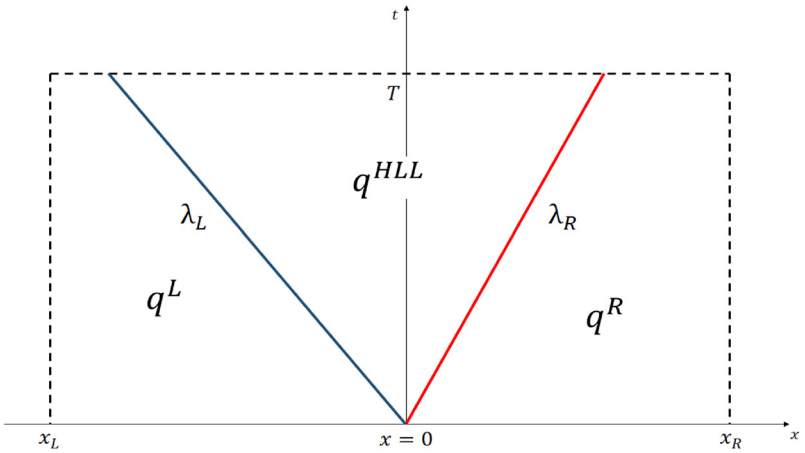

Fig 6. Space-time grid cell $[x_L, x_R] \times [0, T]$. The figure shows the evolution of the 1D conservative equation solution along rays with slope λ_L and λ_R, together with the intermediate state q^{HLL} generated by the HLL solver.

https://doi.org/10.1371/journal.pone.0195494.g006

where $\boldsymbol{f}^{R,L} \coloneqq \boldsymbol{f}(\boldsymbol{q}^{R,L})$. One can work out the approximate solution to the flux through the interface by integrating the hyperbolic equation over the space-time domain outlined in Fig 6 and using the Rankine-Hugoniot jump condition at each characteristic ray ($\lambda_{R,L}$). The final result is that [21]:

$$f_a^{HLL} = \frac{\lambda_R f_a^L - \lambda_L f_a^R + \lambda_R \lambda_L (q_a^R - q_a^L)}{\lambda_R - \lambda_L}. \tag{45}$$

Notice that $\boldsymbol{f}^{HLL} \neq \boldsymbol{f}(\boldsymbol{q}^{HLL})$. The flux Eq (45) can be used along with the Godunov scheme to solve the local Riemann problem of to contiguous grid cells.

Let us now consider the boundary $x_{i-1/2}$ between two control volumes C_i and C_{i-1} and suppose that a constant reconstruction $\tilde{\boldsymbol{q}}$ from the average values of $\boldsymbol{q}$ has been made. With this, let $\tilde{q}_a{}^L(x_{i-1/2}, t_n) \coloneqq [Q_a]_{i-1}^n$ and $\tilde{q}_a{}^R(x_{i-1/2}, t_n) \coloneqq [Q_a]_i^n$ to be the reconstruction points that lay at the interface $x_{i-1/2}$. Note that these values are going to be different if a polynomial reconstruction is made. With this, we can write the numerical flux at $x_{i-1/2}$ used in the Godunov scheme in the following form:

$$[F_a^{HLL}]_{i-1/2}^n = \begin{cases} f_a^L(x_{i-1/2}, t_n), & \text{if} \quad 0 \leq \lambda_L, \\ f_a^{HLL}(x_{i-1/2}, t_n) & \text{if} \quad \lambda_L < 0 < \lambda_R, \\ f_a^R(x_{i-1/2}, t_n), & \text{if} \quad 0 \geq \lambda_L. \end{cases} \tag{46}$$

The flux through $x_{i+1/2}$ is obtained in an analogous way. So, by substituting these numerical fluxes in the discretisation Eq (30), we finally get the numerical solution for the hyperbolic Eq (15) in the finite volume scheme using Godunov's algorithm with a *high resolution* [22] approximate Riemann HLL solver:

$$[Q_a]_i^{n+1} = [Q_a]_i^n - \frac{\Delta t}{\Delta x}\left([F_a^{HLL}]_{i+1/2}^n - [F_a^{HLL}]_{i-1/2}^n\right). \tag{47}$$

A simple way of computing $[Q_a]_i^n$ is by considering that this average value match the magnitude of $\boldsymbol{q}$ evaluated at the midpoint of the grid cell x_i. If $\boldsymbol{q}(x, t)$ is smooth, the error introduced by this approximation is of order $\mathcal{O}(\Delta x^2)$ [8]. In other words:

$$q_a(x_i, t_{n+1}) = q_a(x_i, t_n) - \frac{\Delta t}{\Delta x}\left([F_a^{HLL}]_{i+1/2}^n - [F_a^{HLL}]_{i-1/2}^n\right). \tag{48}$$

Many other HLL-type Riemann solvers have been developed (cf. [21]) and successfully implemented (cf. [19]) but they are beyond the scope of the present article.

Limiters

At first approximation, the reconstruction of q over the grid cell was made considering a constant value $[Q]_i^n$ which is taken as the midpoint value of q of the corresponding control volume C_i. A better way of improving the precision of the above procedure is by considering a piecewise polynomial approximation for this variable.

In the linear case, the reconstruction of q over C_i is given by

$$\tilde{q}(x, t_n) = q(x_i, t_n) + \sigma_i^n (x - x_i), \tag{49}$$

where σ_i^n is the slope of the linear reconstruction. To use the limiters together with a HLL-type Riemann solver, all we need to consider are those points of $\tilde{q}$ in each contiguous grid cells, evaluated at the interfaces $x_{i\pm1/2}$. In this respect, it is not important to do a complete reconstruction of q. The knowledge of q at the boundaries is sufficient for this approximation, and

so the values required to effectively evolve the solution of the hyperbolic equation over the grid cell C_i are:

$$\tilde{q}^L(x_{i-1/2}, t_n) = q(x_{i-1}, t_n) + \frac{1}{2}\sigma^n_{i-1}\Delta x, \tag{50}$$

$$\tilde{q}^R(x_{i-1/2}, t_n) = q(x_i, t_n) - \frac{1}{2}\sigma^n_i\Delta x, \tag{51}$$

$$\tilde{q}^L(x_{i+1/2}, t_n) = q(x_i, t_n) + \frac{1}{2}\sigma^n_i\Delta x, \tag{52}$$

$$\tilde{q}^R(x_{i+1/2}, t_n) = q(x_{i+1}, t_n) - \frac{1}{2}\sigma^n_{i+1}\Delta x. \tag{53}$$

Each pair Eqs (50 and 51) and Eqs (52 and 53), constitute a Riemann problem to be solved at the interface $x_{i-1/2}$ and $x_{i+1/2}$, respectively. The polynomial reconstruction are useful to accurate capture discontinuities such as shock-waves. Eqs (50)–(53) are also valid for each component of the vector $\boldsymbol{q}$ when a coupled system of conservative equations is required.

The usual way of computing σ is by considering some useful function based on finite derivatives of q over C_i. The most used but dissipative reconstruction (also called *limiter* [8]) is the *minmod limiter* (MM) introduced in [23]:

$$\sigma^n_i = \text{minmod}(m_{i-1/2}, m_{i+1/2}), \tag{54}$$

where the function $m_{i\pm1/2}$ is the average slope (or the finite derivative) of q centred at $x_{i\pm1/2}$:

$$m_{i+1/2} = \frac{q(x_{i+1}, t_n) - q(x_i, t_n)}{x_{i+1} - x_i}, \tag{55}$$

$$m_{i-1/2} = \frac{q(x_i, t_n) - q(x_{i-1}, t_n)}{x_i - x_{i-1}}. \tag{56}$$

The *minmod* function of two values a and b stands for:

$$\text{minmod}(a, b) := \begin{cases} 0, & \text{if} \quad ab \leq 0, \\ a, & \text{if} \quad |a| < |b|, \\ b, & \text{if} \quad |b| < |a|. \end{cases} \tag{57}$$

This limiter has been successfully implemented in the case of relativistic hydrodynamics (cf. [11, 18]).

The monotonic centred limiter *MC*, proposed by van Leer [24], has less dissipation than minmod near discontinuities, but has been proved to create spurious oscillations in the strong shock cases [11]. Nevertheless, it produces relatively well damped solutions that capture not too strong shock waves. The slope σ is written as in Eq (54) but the MC function has the

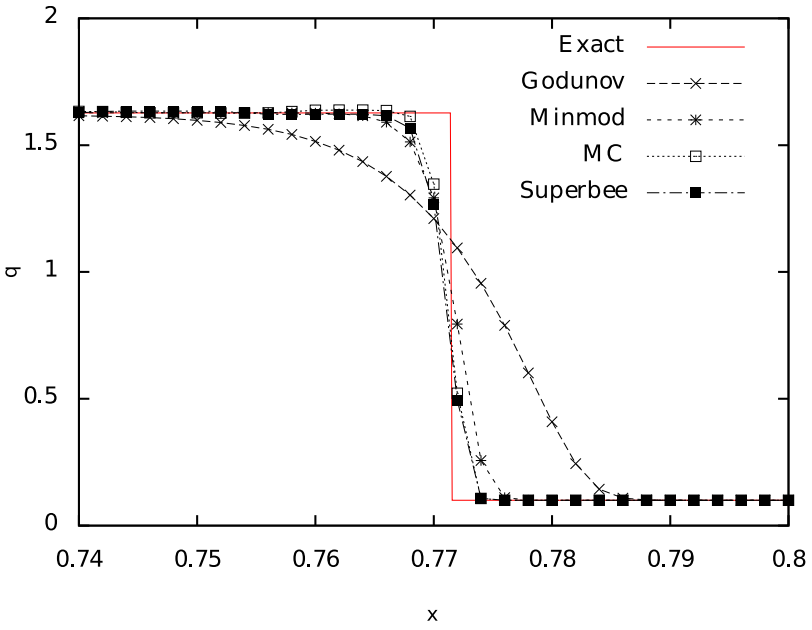

Fig 7. Comparison between the piecewise linear reconstructions (*minmod*, *MC* and *superbee*) with the piecewise constant one (*godunov*). As the complexity of the algorithm grows the shock capture is better, as it is shown in the figure by the superbee simulation. The graph shows the quantity q corresponding to the pressure as a function of the position at a fixed time $t = 0.35$ for a particular Riemann problem in a relativistic Sod shock tube that evolves from the initial value $t = 0$ in such a way that, at this time, $p = 1.69$ for $x < 0.77$ and $p = 0.1$ for $x \geq 0.77$.

https://doi.org/10.1371/journal.pone.0195494.g007

following form:

$$\mathrm{MC}(a,b) \coloneqq \begin{cases} 0, & \text{if} \quad ab \leq 0, \\ 2a, & \text{if} \quad |a| < |b| \quad \text{and} \quad 2|a| < |c|, \\ 2b, & \text{if} \quad |b| < |a| \quad \text{and} \quad 2|b| < |c|, \\ c, & \text{if} \quad |c| < 2|a| \quad \text{and} \quad |c| < 2|b|, \end{cases} \tag{58}$$

where $c \coloneqq (a + b)/2$.

Another piecewise linear reconstruction is the *superbee* limiter, also proposed by Roe in 1986 [23]. This one has a better shock-wave capture than the previous scheme as shown in Fig 7, where comparisons of the superbee limiter with the previous ones and with the piecewise constant reconstruction (*godunov*) is made. For this slope, the function is slightly more complicated than the previous ones and is given by:

$$\sigma = \mathrm{maxmod}\left(\sigma_i^{n(1)}, \sigma_i^{n(2)}\right), \tag{59}$$

where

$$\sigma_i^{n(1)} = \mathrm{minmod}\left(m_{i+1/2}, 2m_{i-1/2}\right), \tag{60}$$

$$\sigma_i^{n(2)} = \mathrm{minmod}\left(2m_{i+1/2}, m_{i-1/2}\right), \tag{61}$$

and

$$\text{maxmod}(a,b) \coloneqq \begin{cases} 0, & \text{if} \quad ab \leq 0, \\ a, & \text{if} \quad |b| < |a|, \\ b, & \text{if} \quad |a| < |b|. \end{cases} \tag{62}$$

Colella in 1984 [25] developed a piecewise parabolic reconstruction (PPM), that have been successfully used by many authors in both relativistic [11] and non-relativistic hydrodynamics (cf. [26]) but for the purposes of this paper, it will not be considered.

Supporting information

S1 File. Numerical vs. exact data. In the file "S1_File.tar.gz", we attached the relevant data of the comparison between numerical simulations and exact solutions used to obtain the L_1—norm.
(GZ)

Acknowledgments

This work was supported by Direccion General de Asuntos del Personal Academico—Universidad Nacional Autonoma de Mexico (DGAPA-UNAM project number IN112616, http://dgapa.unam.mx) and Consejo Nacional de Ciencia y Tecnologia (CONACyT project ID:CB-2014-1 #240512 http://www.conacyt.gob.mx). AAO, SM and DO acknowledge economic support from CONACyT (788898, 26344, 255602). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Author Contributions

Conceptualization: A. Aguayo-Ortiz, S. Mendoza, D. Olvera.

Data curation: A. Aguayo-Ortiz, S. Mendoza, D. Olvera.

Formal analysis: A. Aguayo-Ortiz, S. Mendoza, D. Olvera.

Funding acquisition: A. Aguayo-Ortiz, S. Mendoza, D. Olvera.

Investigation: A. Aguayo-Ortiz, S. Mendoza, D. Olvera.

Methodology: A. Aguayo-Ortiz, S. Mendoza, D. Olvera.

Project administration: A. Aguayo-Ortiz, S. Mendoza, D. Olvera.

Resources: A. Aguayo-Ortiz, S. Mendoza, D. Olvera.

Software: A. Aguayo-Ortiz, S. Mendoza, D. Olvera.

Supervision: A. Aguayo-Ortiz, S. Mendoza, D. Olvera.

Validation: A. Aguayo-Ortiz, S. Mendoza, D. Olvera.

Visualization: A. Aguayo-Ortiz, S. Mendoza, D. Olvera.

Writing – original draft: A. Aguayo-Ortiz, S. Mendoza, D. Olvera.

Writing – review & editing: A. Aguayo-Ortiz, S. Mendoza, D. Olvera.

References

1. Riccardi G, Durante D. Primitive Variable Recovering in Special Relativistic Hydrodynamics Allowing Ultra-Relativistic Flows. International Mathematical Forum. 2008; 42:2081–2111.
2. Sod GA. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. Journal of Computational Physics. 1978 Apr; 27:1–31. https://doi.org/10.1016/0021-9991(78)90023-2
3. Landau LD & Lifshitz EM. Fluid Mechanics. vol. 6 of Course of Theoretical Physics. 2nd ed. London: Pergamon Books.; 1987.
4. Weinberg S. Gravitation and Cosmology: Principles and Applications of the General Theory of Relativity; 1972.
5. Tooper RF. Adiabatic Fluid Spheres in General Relat ivity. Astrophysical Journal. 1965 June; 142:1541–1562. https://doi.org/10.1086/148435
6. Yousaf M, Ghaffar T, Qamar S. Application of Central Upwind Scheme for Solving Special Relativistic Hydrodynamci Equations. Plos One. 2015 Jun;. https://doi.org/10.1371/journal.pone.0128698
7. Font JA, Ibanez JM, Marquina A, Marti JM. Multidimensional relativistic hydrodynamics: Characteristic fields and modern high-resolution shock-capturing schemes. Astronomy and Astrophysics. 1994 Feb; 282:304–314.
8. LeVeque RJ. Finite-Volume Method for Hyperbolic Problems. Cambridge University Press; 2002.
9. Lora-Clavijo FD, Cruz-Pérez JP, F S, González JA. Exact solution of the 1D Riemann Problem in Newtonian and Relativistic Hydrodynamics. Revista Mexicana de Física. 2013 January-June; 59:28–50.
10. Del Zanna L, Bucciantini N. An efficient shock-capturing central-type scheme for multidimensional relativistic flows. I. Hydrodynamics. Astronomy and Astrophysics. 2002 Aug; 390:1177–1186. https://doi.org/10.1051/0004-6361:20020776
11. Lora-Clavijo FD, Cruz-Osorio A, Guzmán FS. CAFE: A New Relativistic MHD Code. ApJS. 2015 Jun; 218:24. https://doi.org/10.1088/0067-0049/218/2/24
12. Radice D, Rezzolla L. THC: a new high-order finite-difference high-resolution shock-capturing code for special-relativistic hydrodynamics. å. 2012; 547:A26. Available from: https://doi.org/10.1051/0004-6361/201219735.
13. von Neumann J, Richtmyer RD. A Method for the Numerical Calculation of Hyd rodynamic Shocks. Journal of Applied Physics. 1950 Mar; 21:232–237. https://doi.org/10.1063/1.1699639
14. Laney CB. Computational Gasdynamics. Cambridge University Press; 1998. Available from: https://books.google.com.mx/books?id=r-bYw-JjKGAC.
15. MacCormack RW, Paullay AJ. Computational efficiencyiy achieved by time splitting of finite differences operators. American Institute of Aeronautics and Astronautics AIAA. 1972;.
16. Book DL, Boris JP, Hain K. Flux-Corrected Transport. II—Generalizations of the method. Journal of Computational Physicis. 1975; 18:248–283. https://doi.org/10.1016/0021-9991(75)90002-9
17. Godunov SK. A Difference Scheme for Numerical Solution of Discontinuos Solution of Hydrodynamic Equation. vol. 47. Mat. Sb. (N.S.); 1959.
18. Rezzolla Lea. Relativistic Hydrodynamics. Oxford University Press; 2013.
19. Miyoshi T, Kusano K. A multi-state HLL approximate Riemann solver for ideal magnetohydrodynaics. Journal of Computational Physics. 2005 Apr; 208:315–344. https://doi.org/10.1016/j.jcp.2005.02.017
20. Harten A, Lax PD, Leer B. On Upstream Differencing and Godunov-Type Conservation Laws. SIAM Review. 1983 04;25. Available from: http://gen.lib.rus.ec/scimag/index.php?s=10.2307/2030019.
21. Toro EF. Riemann Solvers and Numerical Methods for Fluid Dynamics. Springer; 2009.
22. Martí José María, Müller E. Numerical Hydrodynamics in Special Relativity. Living Reviews in Relativity. 2003; 6(7). Available from: http://relativity.livingreviews.org/Articles/lrr-2003-7/. PMID: 28179862
23. Roe PL. Characteristic-based schemes for the Euler equations. Annual Review of Fluid Mechanics. 1986; 18:337–365. https://doi.org/10.1146/annurev.fl.18.010186.002005
24. van Leer B. Towards the ultimate conservative difference scheme. III—Upstream-centered finite-difference schemes for ideal compressible flow. IV—A new approach to numerical convection. Journal of Computational Physics. 1977 Mar; 23:263–299.
25. Colella P, Woodward PR. The Piecewise Parabolic Method (PPM) for Gas Dynamical Simulations. Journal of Computational Physicis. 1975; 54:174–201. https://doi.org/10.1016/0021-9991(84)90143-8
26. St-Cyr A, Jablonowski C, Dennis JM, Tufo HM, Thomas SJ. A Comparison of Two Shallow-Water Models with Nonconforming Adaptive Grids. Monthly Weather Review. 2008; 136:1898. https://doi.org/10.1175/2007MWR2108.1