

Numerical Treatment of Interfaces for Second-Order Wave Equations

**Florencia Parisi, Mariana Cécere, Mirta
Iriondo & Oscar Reula**

Journal of Scientific Computing

ISSN 0885-7474

J Sci Comput

DOI 10.1007/s10915-014-9880-7



Your article is protected by copyright and all rights are held exclusively by Springer Science +Business Media New York. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at link.springer.com".

Numerical Treatment of Interfaces for Second-Order Wave Equations

Florencia Parisi · Mariana Cécere · Mirta Iriondo · Oscar Reula

Received: 14 August 2013 / Revised: 8 April 2014 / Accepted: 12 June 2014
© Springer Science+Business Media New York 2014

Abstract In this article we develop a numerical scheme to deal with interfaces between touching numerical grids when solving the second-order wave equation. We show that it is possible to implement an interface scheme of “penalty” type for the second-order wave equation, similar to the ones used for first-order hyperbolic and parabolic equations, and the second-order scheme used by Mattsson et al. 2008. These schemes, known as SAT schemes for finite difference approximations and penalties for spectral ones, and ours share similar properties but in our case one needs to pass at the interface a smaller amount of data than previously known schemes. This is important for multi-block parallelizations in several dimensions, for it implies that one obtains the same solution quality while sharing among different computational grids only a fraction of the data one would need for a comparable (in accuracy) SAT or Mattsson et al.’s scheme. The semi-discrete approximation used here preserves the norm and uses standard finite-difference operators satisfying summation by parts. For the time integrator we use a semi-implicit IMEX Runge–Kutta method. This is crucial, since the explicit Runge–Kutta method would be impractical given the severe restrictions that arise from the stiff parts of the equations.

Keywords Finite-difference methods · Partial differential equations · Ordinary and partial differential equations · Boundary value problems

F. Parisi · M. Cécere · M. Iriondo · O. Reula (✉)
FaMAF-Universidad Nacional de Córdoba, Instituto de Física Enrique Gaviola, CONICET,
Ciudad Universitaria, 5000 Córdoba, Argentina
e-mail: reula@famaf.unc.edu.ar

F. Parisi
e-mail: fparisi@famaf.unc.edu.ar

M. Iriondo
e-mail: mirta@famaf.unc.edu.ar

M. Cécere
Instituto de Astronomía Teórica y Experimental, CONICET-UNC, Observatorio Astronómico de Córdoba,
X5000BGR Córdoba, Argentina
e-mail: cecere@famaf.unc.edu.ar

1 Introduction

In many situations modern simulations of physical models based on systems of partial differential equations require the use of several grid patches. This could be because the topology of the underlying space is not trivial and so cannot be described by a unique chart (as in the case of codes that evolve fields on a sphere) or it could be because the problem is too large for a single computer and so it has to be partitioned to be solved parallelly in clusters of computers, or because some areas of the integration domain need more resolution than others. Information needs to be passed among these grids in a dynamical and synchronized fashion. Therefore it is important to devise methods that guarantee the stability of the global solution and that require the minimal amount of information to be transferred at grid interfaces, preserving this way a given accuracy.

For some years now, several numerical techniques have been available to deal with these interface problems when solving first-order hyperbolic or parabolic equations. Some of them use interpolation between overlapping regions, while others use penalties which modify the system at boundary grid points by including information from the same space points at neighboring grids [2]. This last method is preferable in many situations, for it has very nice properties. The most interesting one is the fact that it is constructed so that the resulting semi-discrete system preserves the corresponding continuum energy estimate of its constant-coefficient linear system. Thus we can ensure that, at least for linear constant-coefficient systems, the scheme is stable.

Another property that makes these schemes attractive is that the amount of information one has to pass from one grid to the next is minimal. Thus these schemes are optimal for massive parallel computations, where calculations must be split among several CPU/GPU's, and communication among them usually would add a non-trivial overhead to the computation of the solution.

Furthermore, the accuracy of the method is only determined by the precision order of the time and space operators. Hence the amount of data transferred is constant. This is not the case though when interpolation is used since in order to increase the accuracy of that scheme one has to take more points on each of the grids and use higher-order interpolants, leading to a larger amount of data that needs to be transmitted.

For a second-order hyperbolic equation these methods can be used, but with the drawback that one must first rewrite it as a first-order system to be able to apply the approach, by this creating many more variables and constraints, all of which have to be monitored during the evolution. This not only increases the required machine memory, but also increases the amount of traffic among processors, due to the corresponding increase of information that needs to be passed along the boundaries.

In 2008 Mattsson et al. [1] developed a method to evolve second-order hyperbolic equations in the same spirit of the standard SAT scheme without the need to reduce it to first-order. In addition, one of us [3] recently found a way to implement similar techniques for Schrödinger's equation. We show in this paper that the same underlying idea in [3] can be extended to also deal with the second-order wave equation. In this scheme only one field needs to be passed across the boundaries, namely the time derivative of the field, making this a very efficient and simple algorithm, while in the one developed by Mattsson et al., penalties are imposed on the field and its spatial and time derivatives and are present in the evolution of the internal grid points.

The idea of our scheme is to consider the boundary as a repeater which absorbs the incoming wave on one side of the interface and creates an outgoing wave on the other, with a non-increasing total energy.

This paper is organized as follows: In Sect. 2 we describe the new numerical scheme, and derive the interface terms that need to be added to the equations for this method to work. There are two different types of interface terms: one, including only the values of the fields at the same side of the interface (necessary for cancelling the usual boundary term from the elliptic part of the operator in the energy estimate) and another one, that can be regarded as an interaction between the fields at both sides of the interface, and can be considered a penalty, in the sense that it depends on the difference of fields on both sides of the grids and that it drives the difference exponentially to zero. The latter is highly stiff, so a semi-implicit method is needed in order to avoid paying a huge price in the time step. For that purpose, we use IMEX Runge–Kutta methods. Furthermore we present, adapted to our case, the method developed by Mattsson et al. [1].

In Sect. 3 we present some numerical results using the new scheme herein presented. We compare the results of evolving a one-dimensional system on a circle, first with periodic boundary conditions in a single grid and then using the interface scheme between the first and final point of the grid. We also compare our scheme with the usual SAT approach, in which the system is treated in its first-order form and with the second order method developed by Mattsson et al. In addition, we study an implementation of our scheme on a two-dimensional torus. We analyze convergence, accuracy and stability of the proposed method for different scenarios.

In Sect. 4 we present two applications of our method in much more demanding situations, namely, an equation with variable coefficients both in space and time which is often used in general relativity as a standard test, and the propagation of a wave on the surface of 2-sphere which has been partitioned into six square grids.

Finally, in Sect. 5 we present our conclusions.

2 Numerical Scheme

Let us consider, for simplicity, a one dimensional problem; the generalization to more dimensions being trivial. Consider a field $\Phi(x, t) : S^1 \times \mathfrak{R} \rightarrow \mathfrak{R}$ satisfying the wave equation:

$$\partial_t^2 \Phi = \partial_x^2 \Phi, \tag{1}$$

and assume sufficiently smooth initial data is given at $t = 0$: $\Phi(x, 0) = \Phi_0(x)$, $\partial_t \Phi(x, 0) = \Pi_0(x)$.

As mentioned, the traditional way to solve this equation when interfaces are present is by reducing it to a first-order form (hereafter *FO-scheme*) by introducing the variables $\Pi := \partial_t \Phi$ and $\Psi := \partial_x \Phi$. Then equation (1) is equivalent to the system

$$\begin{aligned} \partial_t \Phi &= \Pi, \\ \partial_t \Psi &= \partial_x \Pi, \\ \partial_t \Pi &= \partial_x \Psi. \end{aligned} \tag{2}$$

This way of solving the equation has the previously mentioned disadvantage of introducing auxiliary variables, something that can be very expensive in terms of memory, especially when considering systems of wave equations in many dimensions, as is often the case, for instance, in General Relativity.

Furthermore, the use of first-order systems results in less accurate numerical approximations, since the phase error is larger than when using schemes based on the second-order version of the systems as per [4].

We are interested here in solving Eq. (1) in second-order form for the spatial operators, although we shall keep the first-order form for the time integration, since we will use either a Runge–Kutta or IMEX scheme to advance the fields in time. We therefore consider the system

$$\begin{aligned} \partial_t \Phi &= \Pi, \\ \partial_t \Pi &= \partial_x^2 \Phi, \end{aligned} \tag{3}$$

and develop a numerical method for solving it when interfaces are present. From here on we deal exclusively with interfaces of the type called *conforming grids*, in which for each interface point of a grid there corresponds another point from a neighboring grid representing the same spatial point.

Standard theorems guarantee the existence of a solution to equation (3) satisfying the energy norm

$$\mathcal{E} := \int \{\Pi^2 + \nabla \Phi \cdot \nabla \Phi\} dV. \tag{4}$$

What we want is to develop a scheme that will preserve the analogous discrete-energy norm, thus guaranteeing stability.

In order to solve this system we consider the domain that consists of the interval $[0, 2]$, where the first and last points are identified, resulting in a circle of length two, with the interface defined in the touching extreme points. For the numerical solution we take a uniformly spaced grid and we write the discrete solution as a vector $\{\Phi_j\}$, $j = 0 \dots N$ corresponding to points $x_j = dx * j$ with the interspace between neighboring grid points $dx := \frac{2}{N}$, so that the last point x_N coincides with the first one x_0 at the interface. With this simplification we focus on the interface treatment avoiding the treatment of boundary conditions.

We introduce the discrete l^2 -norm in the usual fashion,

$$\langle \Psi, \Phi \rangle := dx \sum_{j=0}^N \sigma_j \Psi_j \Phi_j, \tag{5}$$

where $\{\sigma_j\}$ is a set of real-valued weights that depend on the finite-difference operators under consideration.

The semi-discrete system we want to solve at all points except at the interface is then

$$\begin{aligned} \partial_t \Phi_j &= \Pi_j, \\ \partial_t \Pi_j &= (D^2 \Phi)_j \quad j = 1 \dots N - 1, \end{aligned} \tag{6}$$

where D is any finite-difference operator that approximates the derivative operator to some order $q \geq 1$ satisfying the summation by parts property (SBP from now on) [5–7]. That is, it satisfies the discrete counterpart of the integration by parts property

$$\langle \Psi, D\Phi \rangle + \langle D\Psi, \Phi \rangle := \Psi_N \Phi_N - \Psi_0 \Phi_0.$$

Alternatively, instead of D^2 , we could use a second-order operator D_2 approximating the second derivative, which satisfies the corresponding SBP property, i.e. that can be written as in [8]

$$D_2 = H^{-1} \left(-D^T H D + R S \right), \tag{7}$$

being $H = dx \text{diag}(\sigma_0, \dots, \sigma_N)$, D an operator that approximates the first derivative, $R = \text{diag}(-1, 0, \dots, 0, 1)$ and S an operator that approximates the first derivative at the interface. This guarantees that the analogue of the integration by parts property for the second derivative holds, i.e. that D_2 satisfies

$$\langle \Psi, D_2 \Phi \rangle = \Phi_N (D\Psi)_N - \Phi_0 (D\Psi)_0 - \langle D\Psi, D\Phi \rangle. \tag{8}$$

This is a better choice, since the operators have a smaller stencil and preserve the solution phase more accurately. In our scheme, we shall use the narrow diagonal SBP second-order operators obtained in [9].

If one could prove that the linear ODE system (6) has eigenvalues with no positive real part and a complete set of eigenvectors, then one could apply any discrete-time integrator, and as a result obtain a stable numerical evolution for the whole system. For a more detailed description of the theory see, for instance, [7]. A way to check those conditions is to find a energy norm that is either constant or decreases in time. This is the procedure we shall use to implement our scheme.

Using (5), the discretized version of the energy norm (4) becomes

$$\mathcal{E} = \langle \Pi, \Pi \rangle + \langle D\Phi, D\Phi \rangle$$

and we get its time derivative as

$$\begin{aligned} \partial_t \mathcal{E} &= \langle \partial_t \Pi, \Pi \rangle + \langle D\partial_t \Phi, D\Phi \rangle \\ &= \langle D_2 \Phi, \Pi \rangle + \langle D\Pi, D\Phi \rangle \\ &= \Pi_N (D\Phi)_N - \Pi_0 (D\Phi)_0. \end{aligned} \tag{9}$$

Here we have used equations (6) and (8). In order to preserve this norm during evolution we need to cancel these terms, since the contributions in the RHS of this equation come from each side of the interface.

In contrast to this first-order hyperbolic and to the parabolic case, it does not seem possible for second-order systems to control the energy by introducing on each side only terms proportional to the difference of the fields and their normal derivatives at each interface without modifying the standard energy as in Mattsson et al. [1]. Thus, following [3] we introduce our first modification by adding terms at the interface as follows:

$$\partial_t \Pi_j = (D_2 \Phi)_j + \frac{1}{dx \sigma_0} \delta_{j0} (D\Phi)_0 - \frac{1}{dx \sigma_N} \delta_{jN} (D\Phi)_N,$$

where δ_{ij} is the Kronecker delta. With this modification the interface terms are cancelled in Eq. (9) and remaining constant the energy norm, but they introduce no interaction between the two sides of the interface, and so the solution we get would just bounce back at the interface (the energy is conserved and if one interface point can not possibly influence the corresponding point at the other side, the pulse has to bounce back). However, eliminating the interface term means that we can now concentrate on adding terms that, while preserving/decreasing the energy norm, introduce an interaction at the touching points of the grid in such a way that the wave can pass through the interface. We must therefore introduce a term that couples the two sides, namely a penalty term that forces the values at both extremes to coincide. The simplest one that satisfies this requirement is

$$\begin{aligned} \partial_t \Pi_j &= (D_2 \Phi)_j + \frac{1}{dx \sigma_0} \delta_{j0} (D\Phi)_0 - \frac{1}{dx \sigma_N} \delta_{jN} (D\Phi)_N \\ &\quad - L(\Pi_0 - \Pi_N) \left(\frac{1}{dx \sigma_0} \delta_{j0} - \frac{1}{dx \sigma_N} \delta_{jN} \right), \end{aligned} \tag{10}$$

where L , which we call the *interaction factor*, is a positive real constant to be chosen as large as possible in order to make the interaction as strong as possible. In this way we penalize the difference on both sides of the interface and drive them to coincide through a very large exponential factor while keeping the energy bounded, as follows

$$\begin{aligned} \partial_t \mathcal{E} &= \langle \partial_t \Pi, \Pi \rangle + \langle D \partial_t \Phi, D\Phi \rangle \\ &= \langle D_2 \Phi, \Pi \rangle + \langle D\Pi, D\Phi \rangle + \Pi_0 (D\Phi)_0 - \Pi_N (D\Phi)_N - L(\Pi_0 - \Pi_N)^2 \\ &= -L(\Pi_0 - \Pi_N)^2. \end{aligned} \tag{11}$$

The limitation of how large L can be chosen comes from the fact that a too large negative real part would make the system unstable by making a large contribution to the eigenvalues along the negative real axis, thus making explicit time integration schemes fall outside their stability region, or making the needed time step prohibitively small. For explicit schemes, the value of L should not be larger than $L = 1$, since it contributes to the CFL factor as much as the principal part. This value turns out not to be large enough for our scheme, giving unacceptably large errors in the form of bounces at the interface for a resolution that describes appropriately the solution. Thus we use larger factors and resort to a semi-implicit method which would free us from the CFL limitation.

Summarizing, the system of ordinary differential equations described in (6) with the proposed correction introduced in (10), is evolved using a third-order IMEX method [10, 11], specifically the one called IMEX-SSP3(4,3,3) L-stable scheme as per [11]. For the spatial discretization we propose finite-difference operators approximating the second derivative obtained in [9].

In order to confirm the correctness of our approach we compare the results of the method proposed with the corresponding traditional explicit third order Runge–Kutta method verifying the presence of the bounces at the interface with the choice of $L = 1$. In addition, for the spatial discretization, we compare the choice of the spatial operator that approximates the second derivative with the use of first derivative operators [6, 12, 13] applied twice. We do this because in some systems where off-diagonal terms occur in the Laplacian, or lower-order terms are present, one might want to use a single operator for every derivative.

Furthermore we compare the results achieved applying these methods with the traditional evolution obtained using the standard *FO-scheme* [2], which consists of adding penalty terms to the discretization of equation (2) at the interface points, causing the energy to be preserved, namely

$$\begin{aligned} \partial_t \Phi_j &= \Pi_j, \\ \partial_t \Psi_j &= D\Pi_j - \frac{1}{2} \frac{\delta_{j0}}{dx \sigma_0} ((\Pi_0 - \Pi_N) + (\Psi_0 - \Psi_N)) \\ &\quad + \frac{1}{2} \frac{\delta_{jN}}{dx \sigma_N} ((\Pi_N - \Pi_0) - (\Psi_N - \Psi_0)), \\ \partial_t \Pi_j &= D\Psi_j - \frac{1}{2} \frac{\delta_{j0}}{dx \sigma_0} ((\Pi_0 - \Pi_N) + (\Psi_0 - \Psi_N)) \\ &\quad - \frac{1}{2} \frac{\delta_{jN}}{dx \sigma_N} ((\Pi_N - \Pi_0) - (\Psi_N - \Psi_0)). \end{aligned} \tag{12}$$

Finally, as mentioned in the Introduction, Mattsson et al. developed a finite difference second order method that includes a SAT technique to treat the interface and the boundary conditions (hereafter *SO-Mattsson et al.'s scheme*).

To be able to compare both schemes, we use the notation that we introduced in this section for summarizing the *SO-Mattsson et al.'s scheme*. We consider a domain that contains an interface at $x = 0$ and we denote by $\Phi^{(1)}$ and $\Phi^{(2)}$ the numerical solutions at the left and right grids respectively. Hence, the conditions that must be imposed on the interface are given by:

$$\begin{aligned} I_1 &\equiv \Phi_N^{(1)} - \Phi_0^{(2)} = 0, \\ I_2 &\equiv \Pi_N^{(1)} - \Pi_0^{(2)} = 0, \\ I_3 &\equiv (RS\Phi^{(1)})_N + (RS\Phi^{(2)})_0 = 0, \end{aligned} \tag{13}$$

where R and S are defined in equation (7). In addition, at the left and right boundaries we should have:

$$(RS\Phi^{(1)})_0 = 0 \quad \text{and} \quad (RS\Phi^{(2)})_N = 0. \tag{14}$$

The semi-discretization of the differential equation, imposing (13) and (14) becomes¹:

$$\begin{aligned} \Phi_{tt}^{(1)} &= D_2\Phi^{(1)} + H^{-1}(\tau\mathbf{e}_N I_1 + \beta(RS)^T \mathbf{e}_N I_1 + \gamma\mathbf{e}_N I_3 + \sigma\mathbf{e}_N I_2) \\ &= D_2\Phi^{(1)} + H^{-1}(\tau\mathbf{e}_N I_1 + \beta S^T \mathbf{e}_N I_1 + \gamma\mathbf{e}_N I_3 + \sigma\mathbf{e}_N I_2), \\ \Phi_{tt}^{(2)} &= D_2\Phi^{(2)} - H^{-1}(\tau\mathbf{e}_0 I_1 + \beta(RS)^T \mathbf{e}_0 I_1 - \gamma\mathbf{e}_0 I_3 + \sigma\mathbf{e}_0 I_2) \\ &= D_2\Phi^{(2)} - H^{-1}(\tau\mathbf{e}_0 I_1 - \beta S^T \mathbf{e}_0 I_1 - \gamma\mathbf{e}_0 I_3 + \sigma\mathbf{e}_0 I_2). \end{aligned} \tag{15}$$

Note that the term containing S^T introduces the penalties in the evolution of the interior points on both sides of the grids.

Using (15), we calculate the energy proposed by Mattsson et al. and verify the correctness of the stability conditions derived in [1], namely

$$\begin{aligned} \gamma &= -\frac{1}{2} = -\beta, \\ \tau &\leq -\frac{1}{2\alpha_M dx}, \\ \sigma &\leq 0, \end{aligned}$$

where the positive constant α_M is given by Lemma 2.3 in [1]. As we compare the schemes, it arises that α_M should satisfy $\alpha_M \leq \sigma_0$.

We report on the findings in the following section.

3 Tests

We test the method by running simulations both in one and two dimensions (1D respectively 2D). For the 1D case all the runs are performed on a circle of length 2 (i.e. the domain is the interval [0, 2], where the last grid point is identified with the first one). For the 2D simulations the domain is a torus and the grid consisted of a 2×2 square with the $x = 0$ face identified with the $x = 2$ face, and similarly for the y coordinate. In this case, for simplicity, one of the interfaces, namely the one corresponding to the y direction, is treated using penalties,

¹ Notice that some signs in (15) are different from those appearing in [1], which contain some typos.

while for the x direction we use periodic operators. In all the runs the number of points and the order of the finite-difference operators employed guarantee a good enough resolution for cases where the solution has a high frequency.²

3.1 Initial-Data Sets

For the purpose of analyzing convergence it is sufficient to choose **smooth** initial data. We therefore choose the following data

– **1D smooth initial data**

$$\begin{aligned}\Phi_0(x) &:= 4^{12} x^{12} (x - 1)^{12}, \\ \Pi_0(x) &:= \partial_x \Phi_0(x).\end{aligned}$$

On the other hand, for the purpose of comparing realistic situations and in order to analyze how the method keeps the phase of the solution, we take the following **rough** and highly variable data:

– **1D rough initial data**

$$\begin{aligned}\Phi_0(x) &:= e^{-8^2(x-0.5)^2} \cos(50\pi x), \\ \Pi_0(x) &:= \partial_x \Phi_0(x),\end{aligned}$$

corresponding to a rough pulse propagating to the left.

– **2D rough initial data**

$$\begin{aligned}\Phi_0(x, y) &:= e^{-8^2((x-1.5)^2+(y-1.5)^2)} \cos(50\pi(y - x)), \\ \Pi_0(x, y) &:= \frac{1}{\sqrt{2}}(\partial_x \Phi_0(x, y) - \partial_y \Phi_0(x, y)).\end{aligned}$$

3.2 Space Discretizations

As mentioned in the Introduction, in this section we perform runs for different schemes and different choices of the space discretization. We compare the results to

- The traditional *FO-scheme*.
- The new second-order formulation presented in this paper (hereafter *SO-interface approximation*) with a second derivative operator D_2 .
- The *SO-interface approximation* where the second derivative is approximated by the first derivative operator D applied twice (i.e. D^2) with and without dissipation. Here we use the same discretization as in *FO-scheme*.
- The *SO-Mattsson et al.'s scheme* with a second derivative operator D_2 .

In all the runs we use a very accurate finite-difference operator, in particular, the first derivative operator is an optimized operator of order eight in the interior and order four at points in the boundary [6, 12, 13]. The second derivative operator used is of order eight in the interior and order six at the boundary [9]; this operator comes also with a first-order companion that is used for the boundary contributions, both of these satisfy SBP with the same norm. In the comparison with *SO-Mattsson et al.'s scheme* we use the finite difference operator of 6th-order accuracy in the interior and 5th-order accuracy at the boundary (see [1])

² Here we aim at an accuracy of about one part in 10^3 for 10 periods. Enough to keep the phase without appreciable error for about 10 crossing times.

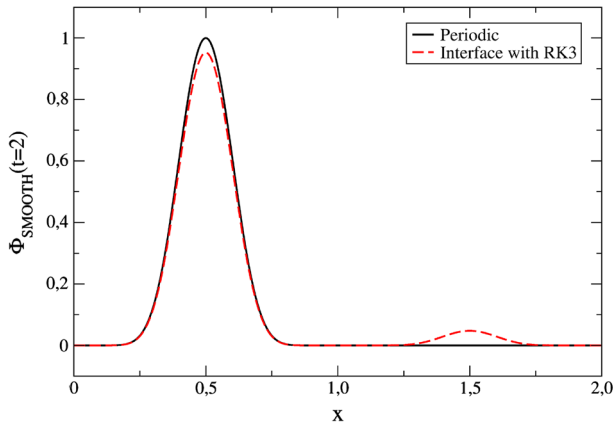


Fig. 1 Comparison of periodic and *SO-interface approximation* runs using the traditional third-order Runge–Kutta method with $L = 10$

Furthermore, the choice of these operators was made in order to preserve the correct phase of the solution on long-time runs, and to be able to test the contribution to the error coming from the interaction term, with the smallest possible interference from the contribution to the error of the space discretization of the derivatives.

3.3 Time Integration

As noted in Sect. 2, we use two time integrators, a traditional Runge–Kutta scheme and an IMEX one. The necessity of an IMEX scheme comes from the fact that the interaction factor has to be very large, hence stiff, in order to achieve good accuracy.

To visualize this, we implement the *SO-interface approximation* with an interaction factor of $L = 1$ and $L = 10$, and we evolve equation (10) using a smooth initial data and a traditional third-order Runge–Kutta. The runs were performed with a resolution of $N = 640$ points and with $dt = 2.5 \times 10^{-5}$ ($CFL = 0.008$). In the plot below, Fig. 1, we show both the periodic solution, i.e. the exact solution of the wave Eq. (1), and the *SO-interface approximation*. The extra bump to the right is the bounce of a fraction of the solution at the interface.

It is possible to reduce the error to a very small amount by enlarging the interaction factor, but at the expense of losing efficiency, since for the traditional Runge–Kutta scheme the time step needed for stability becomes significantly smaller. In fact, we observe that the errors fall to very small values for an interaction factor a thousand times larger if we use, in the traditional Runge–Kutta, a time step a thousand times smaller.

Thus to avoid small time steps, while allowing larger interaction factors, semi-implicit methods are needed. For this reason, we use in our implementation a method among those called IMEX, [10, 11], specifically, the one called IMEX-SSP3(4,3,3) L-stable scheme presented in [11]. This method permits us to explicitly solve stiff parts of the equations while keeping the other terms as usual in traditional Runge–Kutta schemes. In the plot below, Fig. 2, we show the error of the interface (i.e. the difference with the periodic run) using the IMEX method.

From now on, for all runs using *SO-interface approximation*, we present the results evolved with the IMEX-SSP3(4,3,3) L-stable time integrator, while all runs for the *FO-scheme*, all the periodic runs and comparison with *SO-Mattsson et al.’s scheme* (in Sect. 3.8)

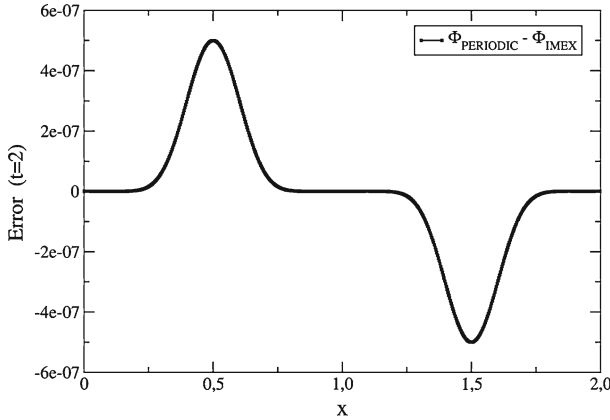


Fig. 2 Error of the interface run (i.e., difference with the periodic run), using IMEX-SSP3(4,3,3) L-stable time integrator with an interaction factor $L = 10^6$

are performed with the traditional third or fourth-order Runge–Kutta integrator. Note that the only term that needs to be treated implicitly with this IMEX method is the term proportional to L , that is, just the last boundary term in Eq. (10).

3.4 Convergence

In the absence of the interaction term, we expect the error to be of the form $e = f_1 dt^p + f_2 dx^q$, where p depends on the time integrator used and q on the space discretization of derivatives.

The convergence rate is calculated as

$$Q = \ln \left(\frac{\|\Phi^{(h)} - \Phi^{(h/2)}\|_{l^2}}{\|\Phi^{(h/2)} - \Phi^{(h/4)}\|_{l^2}} \right) / \ln(2), \tag{16}$$

where $\Phi^{(h_i)}$ is the numerical solution with grid spacing h_i .

In our case, we expect $p \geq 3$ for the IMEX algorithm. The precise value depends on the nature of the solution, in particular the size of the solution near the boundary (where the implicit part of the algorithm is used) in comparison with the size of the solution in the interior of the grid. Since for the space discretization (in the case of D^2) we use a fourth-order accurate operator at the boundary and eighth-order accurate in the interior, $q \geq 5$.

For stability reasons, the CFL condition on the explicit integrator is such that we need to scale dt as dx , so we expect a convergence index of the order of three. Alternatively, we might fix a sufficiently small dt and increase the space resolution, which allows us to study in an independent way space convergence. In this case we would expect a convergence index of the order of five. Any smaller convergence factor must result from the interface treatment. For most of the convergence tests we used very smooth initial data, since the f_1 and f_2 functions depend on high derivatives of the exact solution.

We start by analyzing the convergence of the method for the 1D case with runs of 640, 1,280 and 2,560 points using the smooth data thus in *FO-scheme* as *SO-interface approximation* with D^2 and D_2 operator, and with CFL and dt fixes.

From Fig. 3 we see that, for *FO-scheme* and *SO-interface approximation* with D^2 operator keeping CFL constant (0.08), the convergence factor starts at a value of 3 while the pulse is in the interior, meaning that the main contribution to the error comes from time discretization.

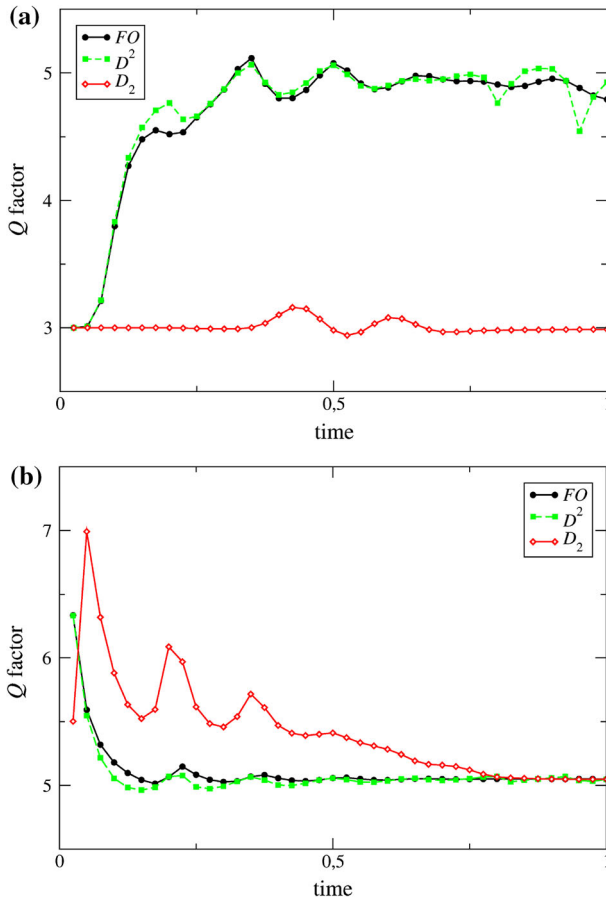


Fig. 3 Comparative plot of the convergence factor for the 1D system, for: first-order formulation FO -scheme, SO -interface approximation using D_2^2 and D_2 respectively, at **a** CFL, and **b** dt fixes

By the time the solution reaches the boundary the Q factor climbs to ~ 5 , which means that there the space discretization is the primary contribution to the error. For SO -interface approximation with D_2 second-order case, however, the convergence remains constant around 3, implying that during the whole run the derivative operator the contribution to the error is negligible.

On the other hand, for fixed $dt = 2.5 \times 10^{-5}$, we observe that during the whole run the error is dominated by the space operators, and the convergence factor starts at a high value, close to 8, corresponding to the time when the pulse has not reached the boundary; and falling to 5 when it does.

For the 2D case we performed runs of 640×640 , $1,280 \times 1,280$ and $2,560 \times 2,560$ points with the rough data.

From Fig. 4, we observe very similar behavior as in the 1D case. Before the wave reaches the boundary, for all discretizations using the third-order Runge–Kutta integrators, both conventional or IMEX, convergence is dominated by the time discretization with a Q factor close to 3, climbing to ~ 5 as the pulse reaches the interface (where the space discretization is the one contributing the most to the error).

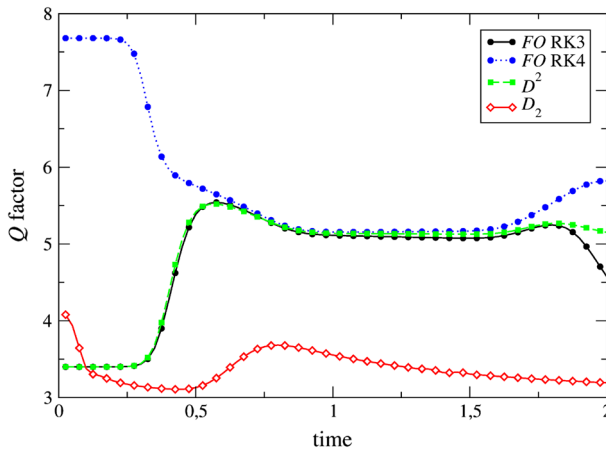


Fig. 4 Comparative plot of the convergence factor for the 2D system, at fixed CFL = 0.08. Plotted are the first-order formulation *FO*-scheme and *SO*-interface approximation using D^2 and D_2 respectively

We also performed a run, for comparison, using a fourth-order Runge–Kutta method for the first-order system. In this case we observe that in the interior the convergence improves and starts close to 8. Here the time integrator is more accurate and hence the space discretization becomes more important. As the pulse reaches the boundary, we again obtain a Q factor of 5.

Convergence alone is not enough to guarantee that we are approaching the correct solution. That is, in principle, the limit of our finite-difference scheme does not need to coincide with the continuum equation (because of the boundary terms which grow with resolution). Thus, it is necessary to analyze convergence against the true solution, which we do in the next section.

3.5 Accuracy

Here we compare methods for realistic data, namely the rough initial data given above, both for the 1D and 2D cases.

For the 1D case we evolve the solution up to $t = 2.0$, at which point the solution has moved to the left and the pulse has completely passed the interface located at $x = 0$. In the 2D case a pulse is sent in an oblique direction to the interface to check whether the scheme preserves the correct phase in this case and does not introduce, for instance, an excess bounce.

For comparison we performed a run using periodic boundary conditions with eighth-order centered-difference operators with $N = 5,120$ points, or $5,120 \times 5,120$ for the 2D case (referred to as $P_{5,120}$ in both cases). This is used as the reference solution against which we compare all the other runs. For these last simulations, interface conditions are used with $N = 640, 1,280$ and $2,560$ points for both the *FO*-scheme system and the *SO*-interface approximation (denoted by $FO_{640}, FO_{1,280}, FO_{2,560}$ and $D_2\ 640, D_2\ 1,280, D_2\ 2,560$ respectively). In addition, for the *SO*-interface approximation, we perform simulations using both D^2 , and D_2 operators. All the runs are performed with an interaction factor $L = 10^6$ and keeping the CFL factor constant (0.08).

In Fig. 5, we show a comparison of the l^2 -norm of the error for the two different cases under consideration: the standard *FO*-scheme with a third-order Runge–Kutta time integrator, and *SO*-interface approximation with the D_2 operator that uses the IMEX-SSP3(4,3,3) L-stable

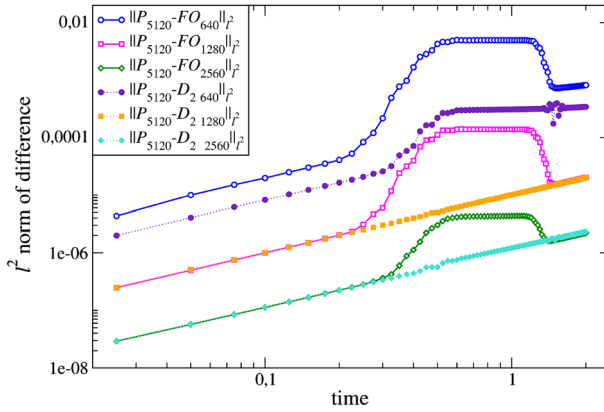


Fig. 5 l^2 -norm of the error for several different 1D runs

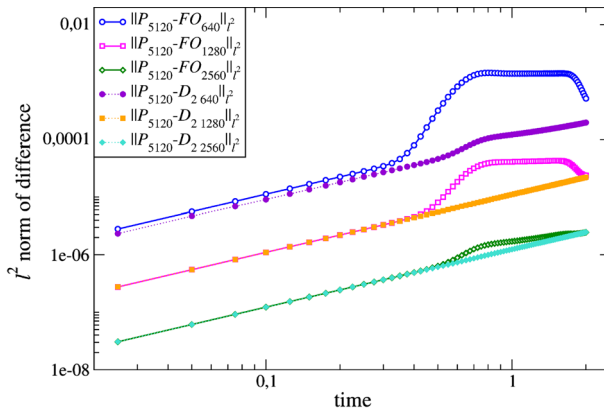


Fig. 6 l^2 -norm of the error for several different 2D runs

scheme. Note that before the pulse has reached the interface, the two methods are comparable, but as soon as the wave reaches and passes through the boundary, the solution obtained using our second-order method improves the accuracy by at least one order of magnitude. This shows that the interface treatment proposed here competes very well with the traditional *FO-scheme*.

We observe the same behavior for the 2D case, displayed in Fig. 6. Again our method and the *FO-scheme* behave similarly in the interior region, but ours is superior to the SAT after the pulse passes the interface.

3.6 Energy Decay

The present scheme is energy-diminishing at the semi-discrete approximation level. This implies that if a stable time integrator is used with a sufficiently small time step the energy given by (4) should decrease only at a rate given by the penalty term, plus, perhaps noticeable, the inherent dissipation of the time integrator. So here we study such a decay, showing that it is indeed very small, as one would infer from the method's accuracy.

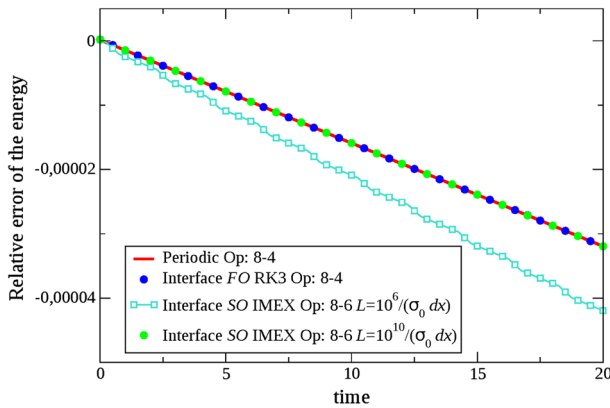


Fig. 7 Relative error of the energy decay for the rough initial data for four different scenarios: periodic boundary conditions, traditional *FO*-scheme, and *SO*-interface approximation using D_2 for two choices of the interaction factor L

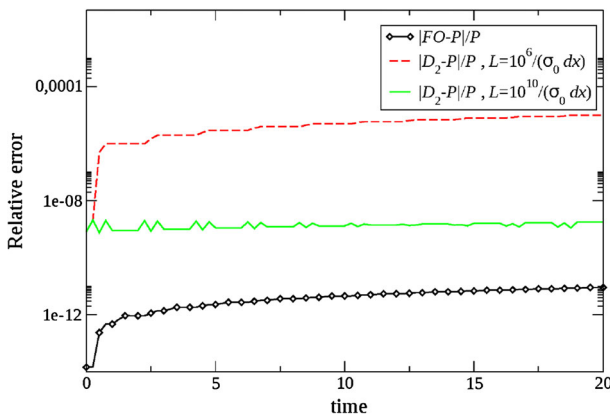


Fig. 8 Energy relative error (compared to the periodic-solution energy) for the rough initial data for three different scenarios: *FO*-scheme and *SO*-interface approximation using D_2 for two choices of the interaction factor L

Figure 7 shows the behavior of the relative error of the energy, i.e. $(E - E_{initial})/E_{initial}$, on longer runs: ten times the previous ones.

These runs were performed with fixed $CFL = 0.08$, and a resolution of 5,120 points. As expected, the decay is very small, and it improves considerably for larger values of the interaction factor. For a value $L = 10^6$, the energy decays at a faster rate than with the first-order SAT scheme, which coincides with the decay given by a periodic treatment. However, if we increase the value of L , the decay approaches the periodic one, and if we take $L = 10^{10}$ the three decays (periodic solution, *FO*-scheme, and *SO*-interface approximation) are indistinguishable. So most of the decay is due to the inherent-Runge-Kutta integrators, and both the standard third-order and the IMEX one seem to have the same dissipation.

Finally, in order to account only for the decay associated to the method, we show in Fig. 8 the relative error of the energy compared to that of the periodic solution, i.e. $(E - E_P)/E_P$

Here we see that the *FO*-scheme is the one that best approximates the periodic energy, while *SO*-interface approximation deviates from it. This difference decreases, however, if

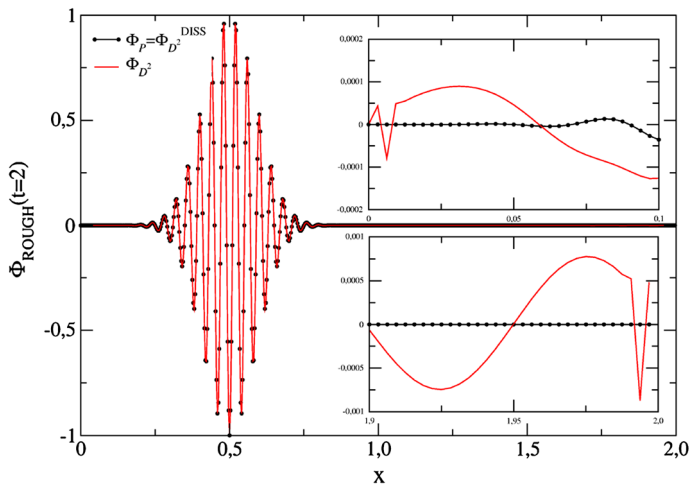


Fig. 9 Comparison of the solution at $t = 2$ using D^2 for the spatial operator with and without dissipation, with the periodic one

we take a larger L , showing, once more, that the larger the interaction factor, the better the proposed method fits the solution.

3.7 Dissipation

It is worth noticing that, for all the runs performed so far, it was not necessary to introduce any artificial dissipation, for we have been considering a linear problem with constant coefficients and smooth data, and therefore there was no noise introduced by high frequency modes. However if we were dealing with a nonlinear equation or one with non-constant coefficients, we might find high frequency oscillations around the correct solution. It is well known that adding dissipation to D^2 we achieve the same result than D_2 . As an example of this, we used the D^2 scheme instead of the D_2 used above, with the rough data.

We see in Fig. 9 that using this operator introduces some numerical noise to the solution, diminishing its quality. We note that the solutions without dissipation are almost indistinguishable except near the interface, where we include a zoomed sector to the right in Fig. 9 to show the disagreement. As known, this noise is removed by using Kreiss-Oliger dissipation [7], that is, by adding to the equations a term proportional to a large power of the Laplacian operator. This term contains a factor that depends on the resolution so that the error produced is of the same or smaller order as the rest of the terms in the approximation.

In particular, we used the one that corresponds to the accuracy we are using for the finite-difference operators [12–14], namely, eighth-derivative dissipation $Q_d = -\sigma_d dx^9 \Delta^4$, where Δ is a finite-difference operator that approximates the Laplacian to first-order accuracy. The runs used for this comparison were performed with a resolution of 640 points at CFL fix (0.08), and $\sigma_d = 100$.

Figure 10 below shows the l^2 -norm of the error using D^2 with and without dissipation, as well as the error for the FO -scheme with dissipation.

We calculated the error by comparing a periodic run with 5120 points against interface runs with 640, 1280 and 2560 points. These were done keeping the CFL factor fix (0.08) and using an interaction factor $L = 10^6$. We see that adding the dissipative term improves the accuracy by one or two orders of magnitude. The errors calculated with dissipation for both FO -scheme and SO -interface approximation systems are almost the same.

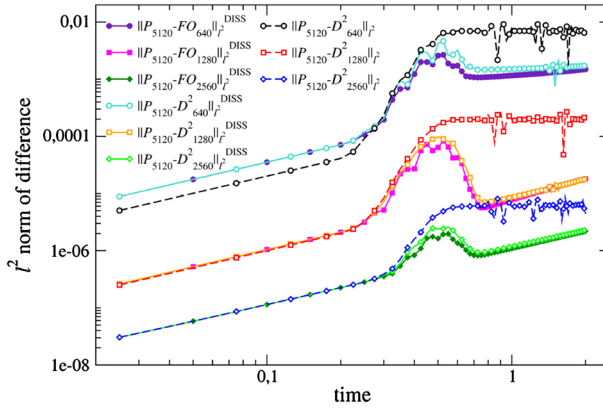


Fig. 10 Behavior of the relative error of the l^2 -norm for runs with operator D^2 with and without dissipation

Also, by comparing with Fig. 6, we see that the errors for the *SO-interface approximation* with D^2 and *FO-scheme* with a dissipation are similar to the error calculated *SO-interface approximation* with the second-order operator D_2 . Thus, as expected, methods that use dissipation are competitive with the D_2 discretization.

3.8 Comparison with Mattsson et al.'s Method

When implementing the *SO-Mattsson et al.'s scheme*, we choose $\sigma = 0$, $\alpha_M = \sigma_0$ and $\tau = -\frac{6.0}{dx\sigma_0}$ and in our *SO-interface approximation* we choose an interaction factor of $L = 10^{10}$. These runs are performed with a fixed factor $CFL = 0.08$.

Furthermore we use the smooth initial data described in Sect. 3.1 and the finite difference operators of second-order accuracy and, of 6th-order accuracy in the interior and 5th-order at the boundary, calculated by Mattsson et al. in [1]. For our scheme we use the operators of second-order accuracy and, of 8th-order accuracy in the interior and 6th-order at the boundary with a first-order companion, calculated in [9].

In Fig. 11a, b we show the error of both method when comparing the numerical solutions obtained using the *SO-Mattsson et al.'s scheme* (Φ_M) and our one (Φ_{D_2}) with the corresponding periodic solutions of same difference operator's accuracy (P_{2560}) for a resolution of 2,560 grid points.

We see that both methods are very similar with a better performance in our scheme when the pulse reaches the interface (see Fig. 11b). Since both methods basically only differ in the treatment of the interface, the difference becomes appreciable in that region. The method developed by Mattsson et al. treats the interface using penalty techniques on Φ , Π and $D\Phi$. These are present not only in the evolution of the terms of the interface, but also in the evolution of the interior point, being the amount of points subjected to the operator accuracy, whilst our method we only needs to communicate the Π field, even though the IMEX scheme requires that this information is passed twice.

4 Applications

In this section, we present two applications of the method developed above, namely, a variable coefficient problem (gauge-wave) and a wave packet propagating on the surface of the 2-sphere.

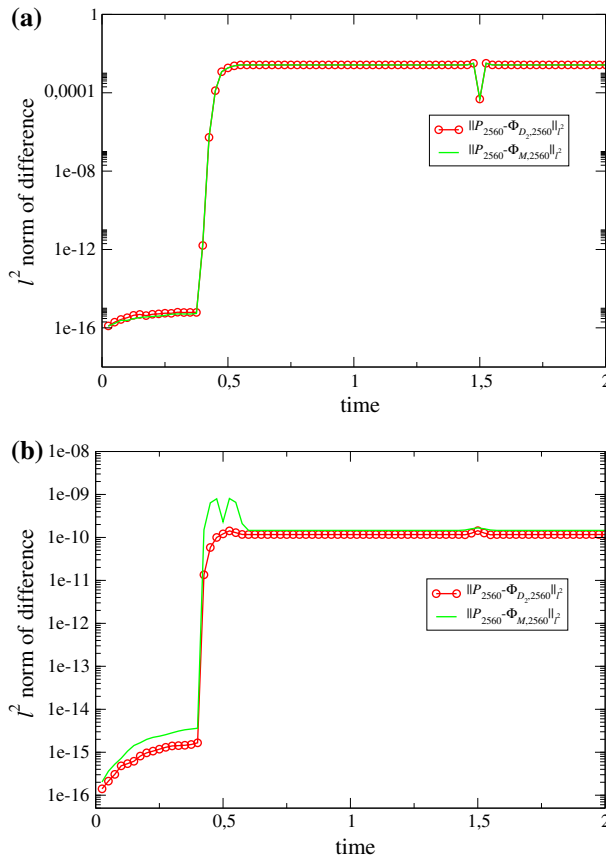


Fig. 11 l^2 -norm of the error corresponding to *SO-Mattsson et al.*'s scheme and to *SO-interface approximation* for $N = 2,560$ with **a** a second-order accuracy difference operator for both scheme and **b** a 6th and 8th-order accuracy difference operator respectively

4.1 Gauge-Wave

As an application of our method involving variable coefficients, we consider a simple 1D test in numerical relativity: a linearized solution of Einstein's equations around a gauge-wave background with line element [15]

$$ds^2 = e^{A \sin(\pi(x-t))} (-dt^2 + dx^2) + dy^2 + dz^2. \quad (17)$$

This background describes flat spacetime, in which a coordinate transformation on the (t, x) plane has been performed, with a sinusoidal dependence along $t - x$. This gauge-wave problem provides us with a simple, yet non trivial, numerical test, for it is linear, the amplitude of the coefficients can be controlled by only adjusting the parameter A , and does not lead to any singularities. This test differs from those of the previous sections since in this case the coefficients depend both on space and time.

There are various papers that deal with this problem [12, 15–17]. Most of them use a method that involves a first-order formulation with periodic boundary conditions, except for [12], which uses a boundary treatment. A second-order scheme with boundary conditions for

this gauge-wave problem was studied in [18,19]. One aspect that these papers show is the exponential growth and loss of convergence displayed by the solution for large amplitudes.

In this section we will apply the method developed above to analyze this problem. We use the same approach as [12] where perturbations of (17) are considered and introduced in Einstein's equations in order to derive the linearized evolution equations for the fields. Here we study the short-time behavior since we are only interested in the stability of the method as the waves go through the interface.

The non trivial variables for this problem are the relevant components of the metric and its time derivative (g_{xx}, K_{xx}) , and the lapse α . We consider, therefore, perturbations of the form

$$g_{xx} = e^{A \sin(\pi(x-t))} + \delta g_{xx}, \tag{18}$$

$$K_{xx} = \frac{A}{2} \cos(\pi(x-t)) e^{\frac{A}{2} \cos(\pi(x-t))} + \delta K_{xx}, \tag{19}$$

$$\alpha = e^{\frac{A}{2} \sin(\pi(x-t))} + \delta \alpha. \tag{20}$$

The resulting equations are

$$\begin{aligned} \partial_t \Phi &= \Pi + A\pi \cos(\pi(x-t)), \\ \partial_t \Pi &= \frac{1}{\hat{\alpha}} \partial_x (\hat{\alpha} \partial_x \Phi) \\ &\quad - \frac{1}{2} \left(A\pi^2 \sin(\pi(x-t)) + \frac{A^2 \pi^2}{2} \cos^2(\pi(x-t)) \right) \Phi \\ &\quad - \frac{1}{2} A\pi \cos(\pi(x-t)) \Pi, \end{aligned} \tag{21}$$

where $\hat{\alpha} = e^{\frac{A}{2} \sin(\pi(x-t))}$, $\Phi = \delta \alpha / \hat{\alpha}$ and $\Pi = \delta K_{xx}$.

We perform several runs and compare the results for the *SO interface approximation* treated in this paper, using first derivative operator applied twice (D^2) and the second derivative operator (D_2). The semidiscretization of the second derivative term in (21) is thus of the form

$$\partial_x (\hat{\alpha} \partial_x \Phi) \approx D(\hat{\alpha} D \Phi), \tag{22}$$

for the D^2 case.

For the D_2 case, on the other hand, we split the second derivative as

$$\partial_x (\hat{\alpha} \partial_x \Phi) \approx \hat{\alpha} D_2 \Phi + (\partial_x \hat{\alpha}) D_1 \Phi, \tag{23}$$

where the derivative of $\hat{\alpha}$ is calculated analytically, and D_1 and D_2 are fully compatible finite-difference operators approximating the first and second derivative respectively (see [9]). Note that, in certain circumstances, the chain-rule form can experience instability but this is not the case, thus for simplicity, we use the chain-rule to expand de D_2 operator despite there are narrow stencil formulations for this operator.

We run the *SO interface approximation* using $N = 161, 321$ and 641 points and $N = 321, 641$ and $1,281$ points both using D^2 and D_2 operators. In Figs. (12) and (13) we show the convergence factor and the l^2 -norm of the error for these two cases compare with the periodic solution with $N = 5,120$ points, respectively. The runs were performed in the $[-1, 1]$ interval, using an amplitude of 0.5, CFL = 0.01 and the following smooth initial data

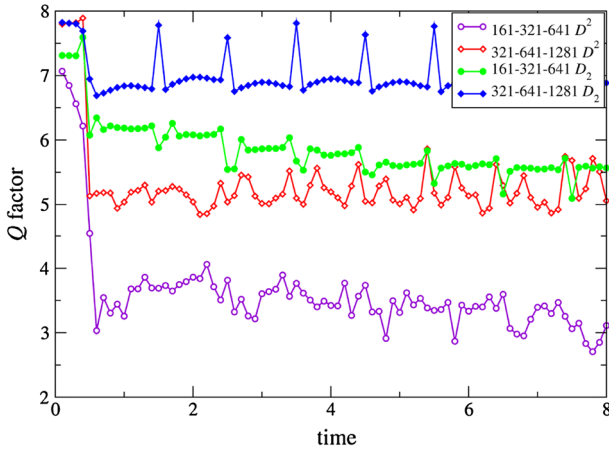


Fig. 12 Convergence factor for two different resolutions with D^2 and D_2 in the SO -interface approximation

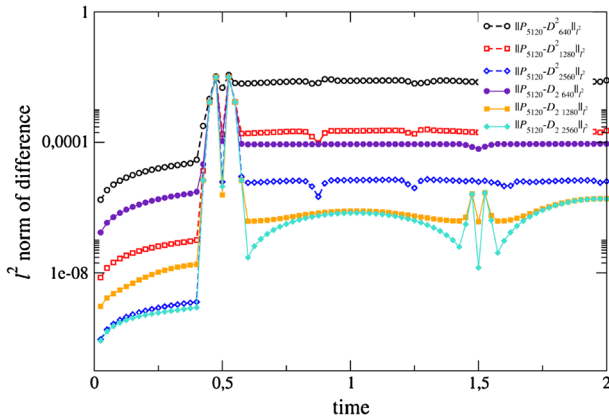


Fig. 13 l^2 -norm of the error for various resolutions for the SO -interface approximation

$$\Phi_0(x) := 100^{12} (x + 0.6)^{12} (x + 0.4)^{12}, \tag{24}$$

$$\Pi_0(x) := \partial_x \Phi_0(x). \tag{25}$$

For all the runs, except for the D^2 case with the lowest resolution, the convergence factors oscillate between the expected values of 8 and 5, consistent with the order of the difference operators used. The l^2 -norm of the error shows again that the D_2 formulation has a lower error compared to the D^2 case for all the resolutions considered. This illustrates the superior performance of the present method even for variable coefficients, which opens a wide range of possible applications.

4.2 Waves in the 2-Sphere

As a second application of our method, consider the wave propagation on the surface of a 2-dimensional sphere. We discretize the domain using six identical square grids, one for each

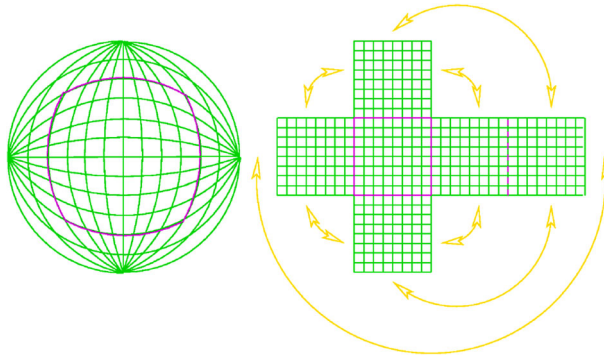


Fig. 14 Configuration of the grid patches used to cover the 2-sphere

of the six patches whose faces are identified in such a way that they cover the sphere (see Fig. 14). The idea is to evolve each patch separately by independent processors and, by imposing the proper interface conditions, obtain the global evolution of the solution.

So, we consider the 2-dimensional wave equation

$$\partial_t^2 u = \frac{1}{\sqrt{g}} \partial_a \left(\sqrt{g} g^{ab} \partial_b u \right), \tag{26}$$

with g^{ab} the metric of the sphere in the coordinates of each patch and g its determinant. We use the initial data on the sphere given by

$$u(\theta, \phi, t = 0) = \begin{cases} 2 \left((\theta^4 - k^4) / k^4 + 0.1 \right) & \text{if } \theta \leq k \\ 0.2 & \text{if } \theta > k, \end{cases} \tag{27}$$

where θ and ϕ are the standard polar angles on the sphere, and $k = 0.5$, and apply the method introduced in Sect. (2) so that each interface is treated using the scheme presented in (10). In Fig. 15 we show the evolution of the solution for several different times (first and third row panels), as well as the difference between this solution and the one obtained treating the interface using the standard SAT method [2] (second and fourth row panels). We see that the wave packet behaves nicely as it passes through the interfaces, and that the difference between the two methods is of order 10^{-4} . We also can see from the second and third row panels that the interface treatment in our method seems to be as good as the SAT scheme, since there is no appreciable difference in the interface, the largest values of the error being the ones in the interior of the patch where the initial bump is located.

5 Conclusions

We have shown that it is possible to implement an interface scheme of “penalty” type for the second-order wave equation similar to the ones used for first-order hyperbolic and parabolic equations and for second-order wave equation with Mattsson et al.’method. Our scheme shares with them similar properties: Only data at interface points need to be passed between grids, and convergence is ensured for linear, constant-coefficient systems.

Our scheme was applied as well to a problem with non-constant coefficients (perturbations of a gauge-wave background) and the wave propagation on the surface of a 2-dimensional sphere. The accuracy of the method seems to be as good as the accuracy of the finite-

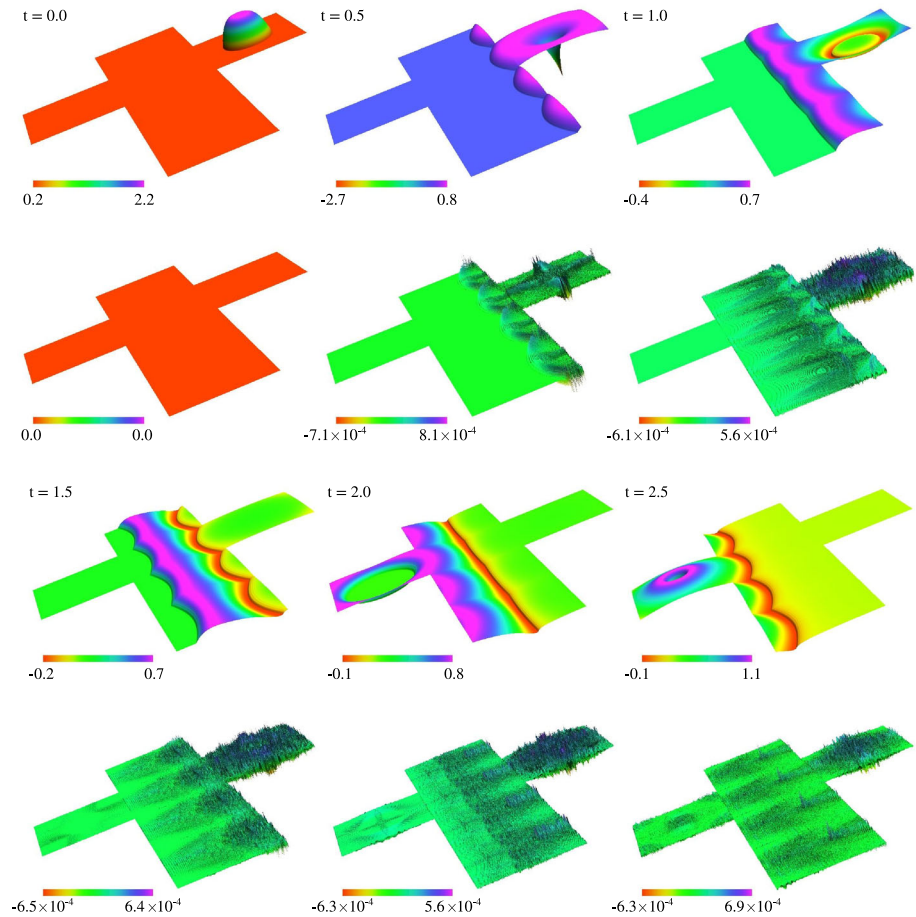


Fig. 15 Evolution of the wave packet in the surface of the 2-sphere with our method (first and third row panels) and the difference between this solution and the one obtained with the SAT technique (second and fourth row panels)

difference operators and of the time integrators used, and competes favorably with both the usual *FO-scheme* and *SO-Mattsson et al.'s scheme* for all the cases we have tried.

Note that for the wave equation in both the *FO-scheme* and *SO-Mattsson et al.'s scheme* one must pass at the boundary many more quantities than in our scheme, namely, in addition to the time derivative fields, either all space derivatives or the normal derivative at the boundary. This fact is important for multi-block parallelizations in several dimensions, for it implies that one obtains the same solution quality while sharing among different computational grids only a small fraction of the data one would need for a comparable (in accuracy) SAT or Mattsson et al.'s method. This will considerably improve the scalability properties of multi-block MPI computations. It might even be advantageous to use it when dividing a grid block into many smaller grids to be dealt by different MPI processes as in binary black hole simulations in General Relativity. In this case the traditional way of doing it is to pass at the boundary the whole stencil needed to compute finite differences using centered operators. The accuracy of

our method implies one could just pass among the neighboring grids the values of the fields, gaining a substantial step on scalability.

Since the information passed along the interface is a time derivative, it behaves as a scalar with respect to coordinate changes in space.³ So, its values at both sides of the grid, namely at two different coordinate patches, should be identified without any change. By contrast, when using *FO-scheme* or *SO-Mattsson et al.'s scheme* and passing space derivatives of the fields, a coordinate transformation is needed in the generic case at which the boundary regions represent different curvilinear coordinate patches. Thus the new scheme requires less coding and less computation.

This new method of dealing with interfaces is not unique to second-order systems, for its underlying ideas can be applied to many cases of interest. In particular, it can be extended to the general case of symmetric hyperbolic first-order systems. This case is under present investigation.

Acknowledgments We thank Luis Lehner and Manuel Tiglio for discussions, and SeCyT-UNC, CONICET, FONCyT and the Partner Group grant of the Max Planck Institute for Gravitational Physics (Albert Einstein Institute) for financial support. O.R. thanks Perimeter Institute for hospitality, where part of this research was carried out.

References

1. Mattsson, K., Ham, F., Iaccarino, G.: Stable and accurate wave-propagation in discontinuous media. *J. Comput. Phys.* **227**, 8753–8767 (2008)
2. Carpenter, M.H., Nordström, J., Gottlieb, D.: A stable and conservative interface treatment of arbitrary spatial accuracy. *J. Comput. Phys.* **148**(2), 341–365 (1999)
3. Reula, O. A.: Numerical treatment of interfaces in Quantum Mechanics. ArXiv e-prints, (2011) <http://adsabs.harvard.edu/abs/2011arXiv1103.5448R>
4. Kreiss, H., Ortiz, O.: Some Mathematical and Numerical Questions Connected with First and Second Order Time-Dependent Systems of Partial Differential Equations. *Lectures Notes in Physics*, vol. 604 (2002)
5. Kreiss, H.-O., Scherer, G.: On the Existence of Energy Estimates for Difference Approximations for Hyperbolic Systems. Tech. Rep. Dept. of Scientific Computing, Uppsala University (1977)
6. Summation by parts for finite difference approximations for d/dx . *J. Comput. Phys.* **110**(1), 47–67 (1994). doi:10.1006/jcph.1994.1005. ISSN 0021–9991
7. Gustafsson, B., Kreiss, H.-O., Olliger, J.: *Time Dependent Problems and Difference Methods*. Wiley, New York (1995)
8. Mattsson, K., Nordström, J.: Summation by parts operators for finite difference approximations of second derivatives. *J. Comput. Phys.* **199**, 503–540 (2004)
9. Mattsson, K., Parisi, F.: Stable and accurate second-order formulation of the shifted wave equation. *Commun. Comput. Phys.* **7**, 103–137 (2010)
10. Ascher, U.M., Ruuth, S.J., Spiteri, R.J.: Implicit–explicit Runge–Kutta methods for time-dependent partial differential equations. *Appl. Numer. Math.* **25**(2–3), 151–167 (1997)
11. Pareschi, L., Russo, G.: Implicit–explicit Runge–Kutta schemes and applications to hyperbolic systems with relaxation. *J. Sci. Comput.* **25**, 129–155 (2005)
12. Lehner, L., Reula, O., Tiglio, M.: Multi-block simulations in general relativity: high order discretizations, numerical stability, and applications. *Class. Quant. Grav.* **22**, 5283–5322 (2005). doi:10.1088/0264-9381/22/24/006
13. Diener, P., Dorband, E.N., Schnetter, E., Tiglio, M.: Optimized high-order derivative and dissipation operators satisfying summation by parts, and applications in three-dimensional multi-block evolutions. *J. Sci. Comput.* **32**, 109–145 (2007)
14. Kreiss, H., Olliger, J.: *Methods for the Approximate Solution of Time Dependent Problems*. GARP Publication Series No. 10. WMO, ICSU, Geneva (1973)

³ This is of course true in the case of scalar quantities. In the case we were dealing with systems of wave equations applied to tensor quantities, some coordinate transformations are unavoidable at interfaces.

15. Tiglio, M., Lehner, L., Neilsen, D.: 3D simulations of Einstein's equations: symmetric hyperbolicity, live gauges, and dynamic control of the constraints. *Phys. Rev. D* **70**(10), 104018 (2004). doi:[10.1103/PhysRevD.70.104018](https://doi.org/10.1103/PhysRevD.70.104018)
16. Calabrese, G., Pullin, J., Sarbach, O., Tiglio, M.: Stability properties of a formulation of Einstein's equations. *Phys. Rev. D* **66**(6), 064011 (2002). doi:[10.1103/PhysRevD.66.064011](https://doi.org/10.1103/PhysRevD.66.064011)
17. Alcubierre, M., Allen, G., Bona, C., Fiske, D., Goodale, T., Guzmán, F.S., Hawke, I., Hawley, S.H., Husa, S., Koppitz, M., Lechner, C., Pollney, D., Rideout, D., Salgado, M., Schnetter, E., Seidel, E., Shinkai, H.-A., Shoemaker, D., Szilágyi, B., Takahashi, R., Winicour, J.: Towards standard testbeds for numerical relativity. *Class. Quantum Gravity* **21**, 589–613 (2004). doi:[10.1088/0264-9381/21/2/019](https://doi.org/10.1088/0264-9381/21/2/019)
18. Szilágyi, B., Kreiss, H.-O., Winicour, J.: Modeling the black hole excision problem. *Phys. Rev. D* **71**(10), 104035 (2005). doi:[10.1103/PhysRevD.71.104035](https://doi.org/10.1103/PhysRevD.71.104035)
19. Babiuc, M.C., Szilágyi, B., Winicour, J.: Harmonic initial-boundary evolution in general relativity. *Phys. Rev. D* **73**(6), 064017 (2006). doi:[10.1103/PhysRevD.73.064017](https://doi.org/10.1103/PhysRevD.73.064017)