Simulations in Science and Engineering for University Education

Ferdinand Ceballos-Bejarano[1], Edison Ceballos-Bejarano[1], Asencio Huaita-Bedregal[1], Ariosto Carita-Choquecahua[1], Segundo Ortiz-Cansaya[1], Franyelit Suárez-Carreño[2], José Calizaya-López[1*]

[1]*Universidad Nacional de San Agustín de Arequipa. Arequipa. Perú; Email: jcalizayal@unsa.edu.pe*

[2]*Universidad de las Américas, Facultad de Ingeniería y Ciencias Aplicadas, Carrera de Ingeniería Industrial*

Abstracts: *In this paper, the strategies and tests that guarantee a reliable solution of partial differential equations (PDE) solved with the finite difference method are proposed. The elements to be considered are presented so that the solution of equations in partial derivatives is the closest to reality. The experience of the proponents and previous work play an important role in validating simulations in science and engineering problems. A series of tests are proposed that must be carried out to validate and make the simulation results reliable, due to the fact that the selection of the finite difference scheme, the initial data, the boundary conditions and the restrictions in the time step, the stability and convergence of the solution cannot be guaranteed a priori. There are situations in which the solution is stable but converges to values that have no physical meaning. To illustrate the strategies and tests inherent in a simulation, such that the solutions can be trusted, the 3D wave equation (three spatial dimensions and one temporal dimension) is solved numerically, by implementing a code in FORTRAN 90, implementing a scheme finite difference to second order approximation. The stability and convergence of the solution are studied according to the schemes proposed in this research, in such a way that the results are reliable, and guarantee the possible practical implementation of the solutions. Finally, the evolution of the scalar field for different times is shown, validated with the principle of conservation of energy. The strategies implemented in this work to guarantee the reliability of the simulations are also valid when solving EDP problems with neural networks, genetic algorithms and other intelligent computing techniques.*

Keywords: Finite Differences, Simulations, Partial Differential Equations, Stability, Convergence.

1. INTRODUCTION

The use of numerical methods to solve problems in science and engineering is becoming more and more frequent. That is due to the emergence of increasingly powerful computers capable of performing simulations that require much computational power or bring together many variables that need to be determined. However, the power of the hardware is not enough; it requires the application of numerical methods widely used in physics, mechanics, biology, economics, and other branches. Explicit and implicit schemes, when used in finite difference to solve PDEs, introduce errors that can be measured by convergence and stability tests. It shows that the realizations of these tests help find the numerical solution closest to the actual solution and improve computational accuracy [1-2].

In problems associated with science and engineering, it is common for phenomena to be described by PDE solutions. The solution and simulation of these equations require as much dedication as an experimentalist in a research laboratory. The person who solves PDEs becomes an experimentalist due to the demands of these simulations and the time required to produce satisfactory results. Numerical experimentation is part of the simulation, as will be noted later [3].

Numerical analysis is a very efficient alternative for solving algebraic equations (polynomials) and transcendental. Numerical analysis has a significant advantage over other methods, such as the repetition of logical instructions (iterations). This procedure allows for improving the initial values considered as a solution. Since it is always the same logical operation, it is very relevant to use computational resources to perform this task. However, it is necessary to clarify that numerical analysis alone does not solve the problem. In the solution of PDEs, numerical methods give approximate results, subject to an error. Although it can be as small as the numerical methods and calculation resources allow, this error is always present, and its handling must be considered in

developing the solutions. This research shows a different alternative to simple iterations to find solutions close to reality [4].

However, errors may occur during the execution of the algorithm. These errors can occur in the handling of the numerical data or in the nature of the chosen model itself. In these cases, a random behavior of the errors between approximations is usually observed. The robustness of a numerical method lies in its convergence and stability. It is possible to use methods whose convergence test indicates the validity of their application. However, during its application, instabilities are obtained that have repercussions on the number of iterations and, consequently, on the time invested in the solution. The idea is to apply convergent and stable methods [5].

In the different PDE scenarios, problems with analytical solutions are unknown, particularly those related to hyperbolic PDEs. Finite differences appear as an ideal numerical method to find an approximate solution to these problems. When applying the finite difference method to hyperbolic partial differential equations, reducing the error arising from the approximations is essential. In practice, when using the finite difference method in hyperbolic partial differential equations, some authors do not consider the solution's stability, consistency, and convergence. In this paper, we delve into the possible errors of the method and the so-called convergence criteria when the solutions are not known [6-7-8].

Convergence is the most challenging property to demonstrate by conventional methods, so it is complemented by looking for the stability of the solution or using the convergence criteria. The latter must be analyzed in a discrete grid with a finite number of points, accompanied by a fundamental condition, for which the Courant-Friedrichs-Levy (CFL) condition will help quickly to demonstrate the stability, where it is only applicable in hyperbolic partial differential equations. It ensures that the numerical solution obtained approximates the proper solution well enough. However, in most publications, stability and convergence are not adequately mentioned. This work's main objectives are to develop and establish completeness and validation strategies for simulations involving Hyperbolic PDEs [9-10].

This work shows the main aspects that must be considered when performing a simulation. Before solving a hyperbolic PDE, which describes a specific physical phenomenon, it is necessary to know the problem in-depth, select the boundary conditions, and make a stability and convergence study to know which finite difference scheme is appropriate or gives less error. These and other aspects are implemented in this work.

This work is structured as follows: in section II, theoretical aspects used and applied during the development of the work are shown. Section III shows the aspects inherent to the simulation, with two examples of science and engineering problems. Section IV discusses the results. Finally, the conclusions and primary references are presented for the research development.

2. DEVELOPMENT

Before any PDEs simulation or solution that describes a physical phenomenon, it is necessary to understand and know the Problem. It involves the physical model, mathematical model, and the procedure to build an algorithm. Once the above is fixed, it is necessary to know which finite difference scheme is appropriate depending on whether the equation to be solved is elliptic, hyperbolic, or parabolic [11]. In addition, which are the initial and boundary conditions. Having clear these aspects, intrinsic to the simulation comes the consideration of possible errors, comparing the numerical solutions found with the analytical solution, which is not always known.

Once a problem has been defined, the immediate thing to do is to construct an algorithm of the possible solution. The procedure for constructing an algorithm is shown below:

- An accurate description of the physical system.
- The differential equations describe the physical system, including an appropriate choice of dependent and independent variables and initial and boundary conditions.

- Discrete operators approximate the derivatives in the differential equations describing the system. For example, the partial derivative operator ∂ s approximated by a DΔ. A consistent discretization implies that lím Δ → 0 is satisfied D f = ∂f for any sufficiently smooth function f.

- Describe the problem, discretization strategies, solution strategies, possible solutions, and if possible, solve it algebraically and discretely for a finite value of ∆.

- In the very thin discretization limit, Δ → 0 a numerical solution approximating the analytical solution must be obtained.

- Once a consistent finite difference approximation has been selected, stability is a necessary and sufficient condition for the numerical solution to converge to the analytical solution.

2.1. Stability, convergence and norms

For any simulation, stability and convergence tests are necessary because of the following [4-9-10-11]:

- When simulations involving numerical solutions of partial differential equations are performed, apart from the rounding and truncation errors of the series, the finite difference expressions are derived from a Taylor series development up to second order, where a calculation error accumulates that must be controlled.

- There are errors inherent to the limitations of fitting the physical model to accurately describe the natural physical system and the limitations of the boundary conditions. Convergence and stability tests ensure that the finite difference approximations implemented in the discretization are correct.

- There are ways to calculate the convergence of a code when the analytical solutions are known. In some cases, appropriate parameters are assumed to have a fine enough mesh and obtain a convergent system in as few iterations as possible. For this, the CFL condition ensures that the meshing allows a solution closer to reality.

- A significant case is when progression schemes are used in the numerical solutions. In many simulations, the time step has to be less than a characteristic value directly related to the spatial step. Otherwise, the code may be unstable and not yield the correct results. The CFL condition is commonly referred to as the numerical propagation velocity.

- To study stability, it is sufficient that the solutions of the differential equations do not show oscillations. Let the dependent variables evolve for a long time to verify that they do not show oscillations or jagged peaks.

- Convergence consists of checking how close the numerical solution obtained is to a known analytical solution. In some cases, when the analytical solution is unknown, one way to perform convergence is to take the numerical value of the program's first iteration as the analytical solution and compare it with the numerical value of the second iteration. Then the value of the second iteration is taken as the analytical solution and compared with the numerical value of the third iteration [12]. An alternative method is convergence criteria, which depend on the problem itself.

Convergence and stability tests for any simulation ensure that the finite-difference approximations used in the discretization are correct. A convergent code is a fundamental tool for simulations to be reliable. The results analyze the stability of the computational code with the CFL condition, which states that the time interval for the numerical evolution is restricted by: $\Delta u \leq k\Delta r\Delta^2$, where k is the unit order number and Δ is the angular mesh spacing. For each value of Δ, a value of Δu is selected that satisfies the above condition and is found experimentally when the numerical evolution is stable.

One way to find the convergence of a code is to proceed as follows: because the finite difference approximation method used in Taylor series development is to second order, it is postulated that the error is also of second order in ∆x:

$$\varepsilon = (\Delta x)^2 \qquad (1)$$

From where

$$\log(\varepsilon) = 2\log(\Delta x) \qquad (2)$$

Equation (2) has the form of a straight line of slope equal to two. If log(ε) vs. log(∆x) is plotted, the error between a known analytical solution and the numerical solution found is obtained. A slope roughly equal to two implies that the numerical solution converges to the analytical solution. In this case of this research, for the convergence, general expressions for the convergence calculation given by the L_1 and L_2 were used [7-11-13].

When the analytical solution is known, one way to check the validity of the numerical solutions and to verify that there are no errors in the programming, or the elaboration of the code is to define the error (1) by the quantities widely used in numerical analysis:

$$L_1 = \sum |f_{ij} - F_{ij}| \qquad (3)$$

$$L_2 = \sum \sqrt{(f_{ij} - F_{ij})^2} \qquad (4)$$

where F_{ij} is the numerical solution and f_{ij} is the analytical solution. The idea is to vary the number of points in the computational grid. For example, suppose that Laplace's equation is solved numerically. In that case, the idea is to compute the electric potential by varying the number of points in the mesh, which involves varying the Δ. The basic idea is to vary the number of points, for example, N=100, …,100000 which implies obtaining different field values for each point. A straight line with slope two should be obtained when plotting $\log(L_1)$ Vs $Log(\Delta)$. It allows us to find the range of stability and convergence of the code or program. The critical point is that codes are not stable for any number of points. The convergence test is also used to study the validity range of the code.

2.2. Computational Mesh, Initial And Boundary Conditions.

In general, there are no problems calculating the numerical derivatives at the internal points of the computational mesh. If the boundary conditions are not correctly handled, problems may appear at the border points, which implies setting strategies for those points. A necessary step for a simulation to be considered acceptable is to perform mesh refinement. As the number of points in a mesh increases, the Δ becomes smaller, and the numerical solution becomes closer to the authentic or natural solution. Mesh refinement is limited by the machine being used because the computation time is longer for very fine meshes.

There are problems where the boundary conditions are chosen according to the simulated system and lead to a reliable solution. On the other hand, there are many problems where the choice of boundary conditions is not easy, and they are necessarily "put by hand." The most commonly used simulations are i) Dirichlet boundary conditions, where only the values of the function to be determined at the edges and the start at t=0 is required. ii) Newmann boundary conditions, where the values of the function's derivatives are specified at the edges and the start at t=0. Combinations of them are also used. For example, in problems where the heat transfer equation is solved, it is common to use a combination of both boundary conditions [4].

2.3. Convergence criteria

It is not feasible to apply the convergence calculation based on the above norms when the analytical solutions are unknown. In this case, convergence criteria are used, i.e., looking for elements inherent to the problem being

solved that allow the user to identify when the solution is reliable. For example, in some cases, the energy conservation principle is used [3].

2.3.1 Proposal

The elements mentioned above inherent to the numerical solutions are put into practice with a simple example to find the solution of an ordinary differential equation. It allows for the proposal and establishes all the elements that must be considered when solving differential equations or carrying out simulations of problems in science and engineering.

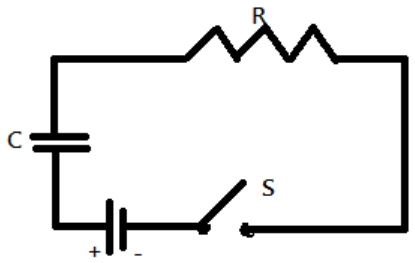

Figure 1. RC circuit with DC voltage source. The initial condition is that the switch closes at t=0.

i) Analitical solution

The RC circuit with continuous source is shown in Figure 1. The switch closes at t=0. The differential equation whose solution describes the behavior of the load:

$$\varepsilon - iR - \frac{q}{C} = 0 \qquad (8)$$

with

$$i = \frac{dq}{dt}$$

it has that

$$\int_0^q \frac{dq}{q - \varepsilon C} = \int_0^t \frac{dt}{RC}$$

whose analytical solution is

$$q = \varepsilon C \left(1 - e^{\frac{t}{RC}}\right) \qquad (9)$$

which describes the behavior of the electric charge for any time.

ii) Numerical Solution

- The differential equation must be written in a convenient form.

$$\frac{dq}{dt} = \frac{q}{RC} + \frac{\varepsilon}{R} \qquad (10)$$

- The differential equation is discretized. Several finite difference discretization schemes have been tested and verified to be stable depending on whether the differential equation is elliptic, parabolic, or hyperbolic:

- Using forward finite differences and substituting them into the differential equation.

$$\frac{q^{n+1}-q^n}{\Delta t}=\frac{q}{RC}+\frac{\varepsilon}{R}$$

$$q^{n+1}=q^n(1-RC)+\frac{\varepsilon}{R}\Delta t \qquad (11)$$

- Numerical implementation

- Pseudocode: for clarity in programming and coding the numerical solution, pseudocode is established. For this example, the pseudocode is:

program capacitor

constants R, C, ε, N

DO $n = 0 \ldots N$

$\Delta t = \frac{1}{N}$

$q^{n-1} = q^n(1-RC)+\frac{\varepsilon}{R}\Delta t$

$q = \varepsilon C\left(1-e^{\frac{t}{RC}}\right)$

$q_{numerical} = q^{n+1}$

$q_{analtical} = q$

$L_1 = \sum |f_{ij} - F_{ij}|$

write()

plot ()

end program

iii) Modeling

- Modeling involves the following:

- Mesh refinement. At least consider coarse, medium, and fine mesh. When varying, the number of points varies, in the same way, Δ, which will allow finding the stability range of the code.

- Use different discretization schemes: implicit and explicit,

- Use different finite difference schemes: forward, centered, and backward.

- Change initial and boundary conditions. Study the sensitivity of the code to these variations.

- Convergence and stability tests.

- Convergence criteria: in this case, a convergence criterion that allows calibrating the constructed code is taking the energy conservation in the form: the energy supplied by the battery is the sum of the energy dissipated by the Joule effect in the resistance and the energy stored in the capacitor.

$$E_b = I^2R + \frac{1}{2}C\varepsilon^2 \qquad (12)$$

Finally, where possible, perform experimental measurements for validation.

3. RESULTS

As an example of the application of the aspects inherent to a simulation, in the sense of the associated elements that give certainty that the results are reliable, the 3D wave equation for the scalar field in the plane case is solved [14]:

$$2(r\phi)_{,ur} - \left((r\phi)_{,r}\right)_{,r} + \eth\,\bar{\eth}\Phi = 0 \qquad (13)$$

where the comma represents the derivative concerning the indicated coordinate in the equation, the angular part of the wave equation is written in terms of the eth operator (ð), which allows precise second-order finite difference expressions for angular derivatives.

The numerical implementation of the method of null characteristics consists of evolving a given field over a family of null hypersurfaces of a discrete sequence of time increments. For the solution of equation (13), a three-dimensional code is developed in FORTRAN 90, stable and convergent to second order. The numerical implementation of the method of null characteristics consists of evolving a given field (scalar, electromagnetic or gravitational) over a family of null hypersurfaces of a discrete sequence of time increments [15]. This code requires data on the initial null hypersurface.

Among the advantages of this formulation is the compactness of the spatial coordinate, which allows transforming an infinite interval into a finite computational grid of points. The compactified radial coordinate is $x = \frac{r}{r+1}$, Such that infinity is described by a null hypersurface x = 1. A field in space-time can now be defined globally in the interval (0 ≤ x ≤ 1). Transcribing the field equations in this new coordinate is necessary before numerical implementation and provides us with an evolution equation for all radii, including x = 1.

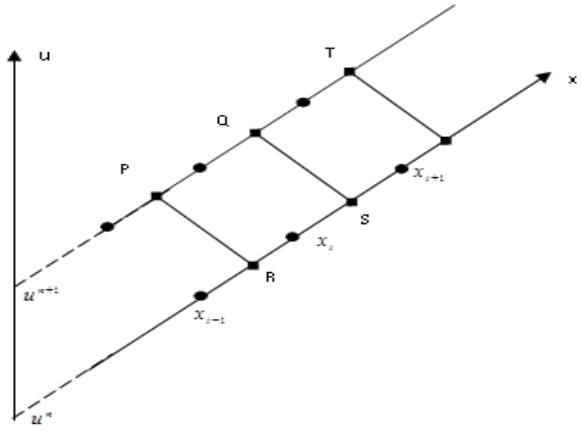

Figure 2. The physical system represents compact support data for the scalar field Φ=0 at the first hypersurface u=0 is directed towards the origin. Also, it allows finding the scattered scalars at the future null infinity, the associated energy, and the power radiated by the scalar field.

A discrete version of the eth formalism was used to find the gravitational radiation emitted by a quasi-spherical source (14). This formalism allows the numerical implementation (discretization) of the angular coordinates, which has been applied to different problems. It covers the sphere with two patches of stereographic coordinates tangent to the northern and southern hemispheres of a topological sphere, with a so-called overlap zone between the patches. The patches are defined by:

$$\zeta_N = tg\left(\frac{\theta}{2}\right)e^{i\Phi} \qquad (14)$$

excluding θ = π

$$\zeta_S = Cotg\left(\frac{\theta}{2}\right)e^{-i\Phi} \qquad (15)$$

excluding θ = 0

Both patches are related to each other

$$\zeta_S = \frac{1}{\zeta_N} = q + ip \qquad (16)$$

In such a way, a mesh of points (qk, pl) can be constructed in a square region of width $N_q\Delta$ and height $N_p\Delta$, where $N_{q,p}$ is the number of points of the angular mesh. The angular part of the equation (13) describes the Laplacian operator and is written in stereographic coordinates in the form:

$$\eth\bar{\eth}\Phi = -\frac{1}{4}(1+q^2+q^2)^2\left[\frac{\partial^2\Phi}{\partial q^2}+\frac{\partial^2\Phi}{\partial p^2}\right] \quad (17)$$

which is easier to implement numerically using second-order finite differences:

$$\frac{\partial^2\Phi}{\partial q^2} = \frac{\Phi_{k+1,l}+\Phi_{k,l}+\Phi_{k-1,l}}{\Delta^2} + O(\Delta^2) \quad (18)$$

$$\frac{\partial^2\Phi}{\partial p^2} = \frac{\Phi_{k,l+1}+\Phi_{k,l}+\Phi_{k,l-1}}{\Delta^2} + O(\Delta^2) \quad (19)$$

What is an initiative known differential equation in spherical coordinates (θ, φ), is implemented in stereographic coordinates (q, p) straightforwardly. The eth calculation simplifies the equations, avoids singularities in the angular coordinates, and is computationally efficient for discretizing the angular derivatives.

3.1 Numerical Implementation

The discretization of the compactified radial coordinate is $x_i = x_0 + (i-1)\Delta x$ to $i = 1 \ldots N$ with $\Delta x = (1 - x_0)/(Nx - 1)$, points $(Nx = 1)$ coincide with null infinity. The stereographic coordinates $\xi = q + ip$ are discretized $q_j = -1 + (j-3)\Delta q$, $p_k = -1 + (k-3)\Delta p$ to $j, k = 1 \ldots N_q$ y $\Delta = \Delta_q = \Delta_p = 2/(N_p - 5)$. Evolution requires that the time step Δu, subject to the analytical domain of dependence being contained in the numerical domain of dependence, the CFL condition is a necessary condition for convergence and is treated with great care when carrying out the simulation.

The initial datum is given at the first hypersurface at the initial time $u = u_0$ and the scalar field φ is zero at the initial cone. Once the field in the first hypersurface is determined, one moves to the next hypersurface N_{du} at the same time $u = u_0 + du$, where one has the evolution of the scalar field φ.

The initial characteristic value problem is formulated in the space-time region between the origin and infinity I_+, with initial datum φ on the initial hypersurface. The null infinity represents a finite mesh of points using the compactified Penrose coordinate. Given an initial datum on the first hypersurface to solve the 3D wave equation numerically, it allows for evolution (points P, R, S, and Q, of the parallelogram) from Figure 1 to the next hypersurface, and the evolution provides the shape of the scalar wave scattered at infinity from an initial pulse:

$$\Phi(u=0,r,q,p)=\lambda(r-R_a)^4(r-R_b)^4(q^2+p^2-0.3)^4\,. \qquad (20)$$

3.2 Stability Tests

To find the stability range of the program, the program needed to perform several iterations according to the meshes indicated in Table 1. The processing times depend directly on the mesh to be fixed. It is observed that the code is stable for the ratio between the radial and angular meshes 2:1, regulated by the CFL condition.

Table 1. Stability validity range for the code built in FORTRAN 90.

N_x	N_q	N_p	Stability
80	20	20	Unstable
320	40	40	Unstable
80	40	40	Stable
120	60	60	Stable
160	80	80	Stable
200	100	100	Stable

It should be noted that for angular meshes larger than 100x100 points, the code takes too long to produce results, although increasing the number of points in the radial mesh yielded results in a reasonable finite time.

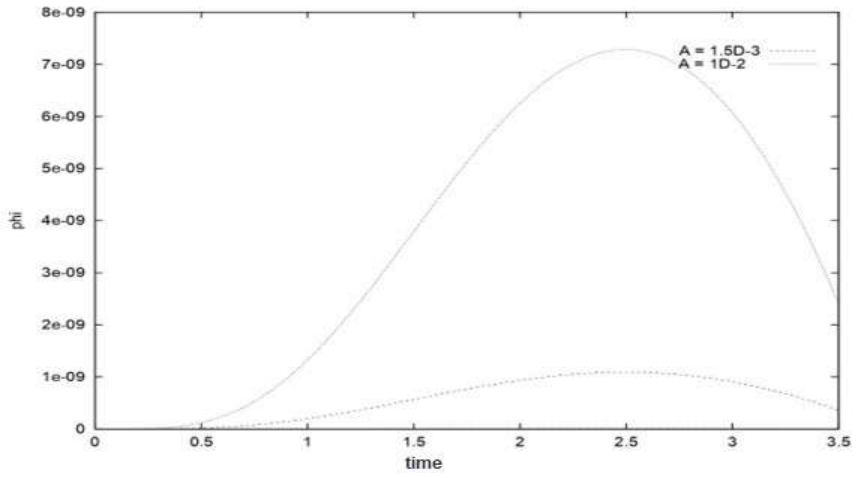

Figure 3. The scalar field as a function of time is shown for the initial data amplitudes 10^{-2} y $1.x10^{-3}$. The smoothness of the evolution is observed (no peaks or oscillations appear), demonstrating the stability of the solutions.

3.3 Code convergence

A convergence criterion of the constructed code was different from the one used in the reference [13]. In this case, the energy and radiated power expressions reported in the reference [14] are used for the flat case M=0 for field energy and radiated power, respectively:

$$E(u)=\frac{1}{2}\int\left[(r\Phi_{,r})^2+\eth\Phi\,\bar{\eth}\Phi\right]drd\Omega \qquad (21)$$

$$P(u) = \frac{1}{2}\int \Phi_{,u}[\Phi_{,u} + \Phi_{,r}]r^2 d\Omega \qquad (22)$$

where $d\Omega$ solid angle element $d\Omega = \frac{4}{P^4} dqdp$. The integrals (21) and (22) are solved numerically and allow us to use the energy conservation principle as a way to verify convergence.

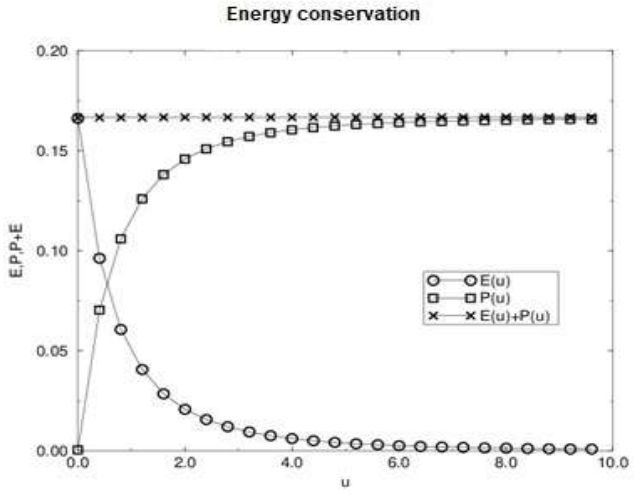

Figure 4. Energy conservation is shown where the sum of the initial scalar field energy and the radiated power is constant.

Another convergence test was performed by solving numerically the angular part of the wave equation, represented by equation (17). For the scalar field Φ, a zero-spin spherical harmonic is chosen for $l = 2$, $m = 2$ [16], such that the equation takes the form.

$$\eth\, \bar{\eth} Y_{22} = 0 \qquad (23)$$

With the application of the rules shown in equations (3) and (4), the results in Figure 5 are obtained.

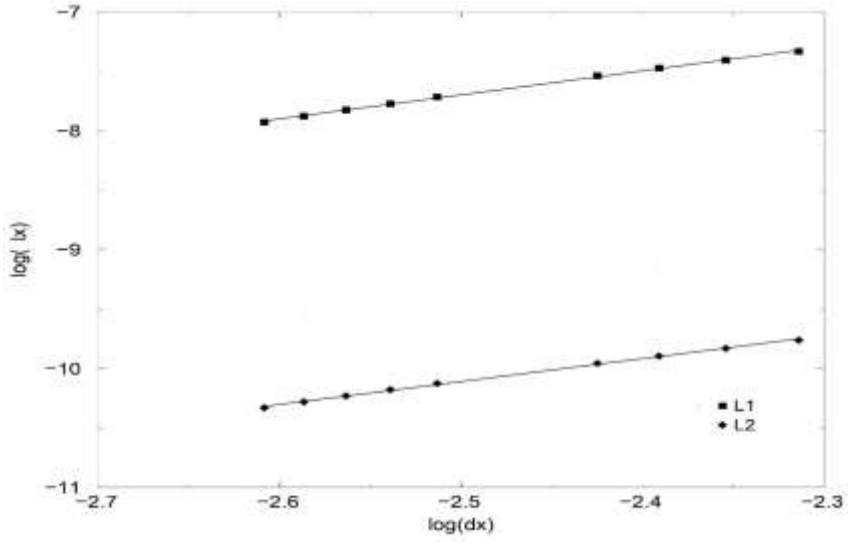

Figure 5. The convergence for the numerical Laplacian is calculated with the spherical harmonic for m=2 y l=2. In this case, L_x, is represented by the norms L_1 and L_2. The slopes are $m = 1.93$ and $m = 1.98$, respectively. According to the finite difference scheme, this result is the expected order for convergence to the second order.

Having demonstrated the stability and convergence of the code, Figure 11 shows the evolution of the scalar field as a function of the stereographic coordinates at four different times. The initial field is the compact support data shown in equation (20). The mesh used in these runs has dimensions: $Np = 80,\ Nq = 80\ and\ Nx = 160,$ represent the points in the angular and radial mesh, respectively.

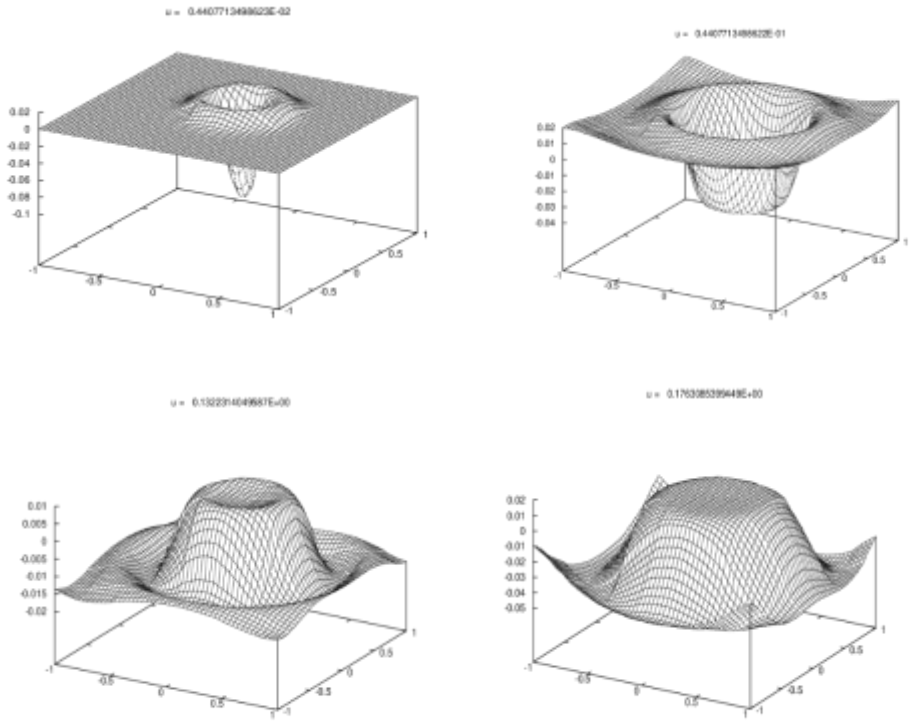

Figure 6. Evolution of the scattered scalar field at infinity (x=1) for an initial compact support data. The amplitude of the scalar field as a function of the stereographic coordinates is shown four times.

4. DISCUSSION

Robust simulation requires a roadmap to validate solutions and models in various environments. It is desirable to have a long-term stability range to control error growth. It is how convergence and stability demonstrate reliable results as several numerical experiments back up the analysis.

It is necessary to know the minimum time unit considering the electrical parameters to have a fine enough mesh to have a convergent system. The CFL condition is commonly represented as the velocity in PDEs [17].

In the cases studied where code instability has been observed, it happens when the CFL condition is violated, which means that the propagation speed, which should be less than the speed of light, is not fulfilled. When applying the finite difference method [17-18-19-20], a criterion must be taken concerning the number of steps in the spatial variable and the number of steps in the temporal variable since the CFL condition must be met for the stability of the method.

Figure 3 shows the scalar field as a function of time for the initial data amplitudes 10^{-2} and $1.x10^{-3}$. The smoothness of the evolution is observed (no peaks or oscillations appear), demonstrating the stability of the solutions.

Figure 4 shows the conservation of energy where the sum of the initial scalar field energy and the radiated power is constant, illustrating a convergence criterion to validate the solutions.

Figure 5 shows the convergence for the numerical Laplacian calculated with the spherical harmonic for m=2 and l=2. The slopes for the standards are $m = 1.93$ and $m = 1.98$, respectively. According to the finite difference scheme, this result is the expected order for convergence to second-order.

Figure 6 shows the evolution of the scattered scalar field at infinity for initial compact support data. It shows the amplitude of the scalar field as a function of the stereographic coordinates four times.

CONCLUSIONS

The 3D hyperbolic wave equation for the scalar field was solved numerically. The tests included, among others, studying the stability and convergence of the code, which allows validating the solutions obtained. The developed code demonstrates that a reliable simulation can be obtained by establishing strategies and criteria when solving science and engineering problems and phenomena involving PDE solutions.

The implementation in the computational code of the methods and techniques that allow solving the 3D wave equation was carried out in Fortran 90. The processing times were high, mainly when the angular mesh was more considerable than 200x200 points.

The finite difference method, together with the method of characteristics and the eth formalism, proved advantageous and powerful, mainly because of the quality of the simulations and the relatively simple computational implementation. Of course, computational limitations for huge meshes come into play.

It is necessary to verify that the finite difference method is consistent and stable to obtain an optimal and convergent approximation to the solution. The CFL condition is a valuable tool in proving the convergence of the solution since it is easier to verify the consistency and stability of a scheme than to prove by definition that the solution of the scheme is convergent. Finalmente:

- Convergence and stability tests ensure that the finite difference approximations used in the discretization and the constructed code are correct.

- There are ways to calculate the convergence of a code when the analytical solutions are known using the convergence criteria.

- Norms L_1 and L_2 are more accurate for fine meshes, but it requires a lot of computational capacity and cost.

The strategies implemented in this work are valid when solving science and engineering problems with neural networks, genetic algorithms and other intelligent computing techniques.

REFERENCES

[1] R. Parovi and S. Tverdyi. Aspects of Numerical Analysis for a Model Nonlinear Fractional Variable Order Equation. Math.Comput. Appl., 2021; 26(55). https:// doi.org/10.3390/mca26030055

[2] J. Saúl. Thermal-mechanical analysis of the briquetting machine segments in steel industries. Rev. Minerva, 2020; 1(1).

[3] L. Yilang, C. Wenbo, Z. Weiwei, and X. Zhenhua. Analysis on numerical stability and convergence of Reynolds averaged Navier–Stokes simulations from the perspective of coupling modes Physics of Fluids 34, 2022. https://doi.org/10.1063/5.0076273.

[4] F. Suárez and L. Rosales. Convergency and Stability of Explicit and Implicit Schemes in the Simulation of the Heat Equation Appl. Sci, 2021; 11, 4468. https://doi.org/10.3390/app11104468

[5] D. Higham and G. Chalmers. Convergence and stability analysis for implicit simulations of stochastic differential equations with random jump magnitudes. Discrete and Continuous Dynamical Systems - Series B, 2008; 9(1): 47-64.

[6] A. Samsudin, N. Rosli and A. Ariffin. Stability Analysis of Explicit and Implicit Stochastic Runge-Kutta Methods for Stochastic Differential Equations. Journal of Physics Conference Series, 2017; 890(1):012084. https://doi.org/10.1088/1742-6596/890/1/012084

[7] E. Lyon and L. Rosales. Fundamentals of finite difference algorithm simulation of the spatial and temporal evolution of heat conduction. Science and Engineering, 2014; 1(1): e003. http://revistas.uniguajira.edu.co/rev/index.php/cei/article/view/3

[8] S. Gedney. Introduction to the finite-difference time-domain (FDTD) method for electromagnetics. Synthesis Lectures on Computational Electromagnetics, 2011; 6(1): 1-250.

[9] Ziauddin, I., Khan, M., Jam, F., & Hijazi, S. (2010). The impacts of employees' job stress on organizational commitment. European Journal of Social Sciences, 13(4), 617-622.

[10] Jam, F. A., Sheikh, R. A., Iqbal, H., Zaidi, B. H., Anis, Y., & Muzaffar, M. (2011). Combined effects of perception of politics and political skill on employee job outcomes. African Journal of Business Management, 5(23), 9896-9904.

[11] S. Marrero, I. González and A. Legra. Analysis of the convergence of the variable integration method applied in the multi-objective optimization of reactive power compensation in electricity supply networks. Dyna, 2015; 82(190): 160-165.

[12] C. Chun, M. Wangy, X. Wisez and Z. Yuex. A positivity-preserving, energy stable and convergent numerical scheme for the Poisson-Nernst-Planck system arXiv:2009.08076. 2020. https://doi.org/10.48550/arXiv.2009.08076

[13] J. Thomas. Numerical Partial Differential Equations: Finite Difference Methods, (Springer-Verlag New York), 1995; 185-200.

[14] L. González. Adaptive algorithms: a bibliographic review. Rev. Athenea, 2021; 2(5).

[15] R. Gomez, L. Lehner, P. Papadopoulos and J. Winicour. Class.Quant.Grav. 1997; 14 977-990. https://doi.org/10.48550/arXiv.gr-qc/9702002.

[16] Jam, F., Donia, M., Raja, U., & Ling, C. (2017). A time-lagged study on the moderating role of overall satisfaction in perceived politics: Job outcomes relationships. Journal of Management & Organization, 23(3), 321-336. doi:10.1017/jmo.2016.13

[17] R. Gómez, W. Barreto and S. Frittelli. Phys.Rev. D76:124029, 2007. https://doi.org/10.1103/PhysRevD.76.124029

[18] W. Barreto, A. Da Silva, R. Gomez, L. Lehner, L. Rosales and J. Winicour J. The 3-dimensional Einstein-Klein-Gordon system in characteristic numerical relativity. Journal-ref: Phys.Rev. D71, 2005.

[19] Y. Zlochower, R. Gómez, S. Husa, L. Lehner, J. Winicour. Mode coupling in the nonlinear response of black holes. Phys.Rev.D 68 084014, gr-qc/0306098, 2003.

[20] Z. Millán, L. De La Torre, L. Oliva and M. Berenguer. Numerical simulation: Diffusion equation. Iberoamerican Journal of Mechanical Engineering, 2011; 15(2): 29-38.

[21] Y. Baba and V. Rakov. Applications of the FDTD method to lightning electromagnetic pulse and surge simulations. IEEE Transactions on Electromagnetic Compatibility, 2014; 56(6): 1506-1521.

[22] A. Elsherbeni and V. Demir V. The finite-difference time-domain method for electromagnetics with MATLAB simulations. The Institution of Engineering and Technology, 2016.

[23] A. Taflove and S. Hagness. Computational Electrodynamics: The Finite Difference Time Domain Method, second edition, Artech House, Boston, 2000.

DOI: https://doi.org/10.15379/ijmst.v10i3.1679