# A study of the 2D wave equation

Colin Munro, Maddie Rodgriguez, Matt Smith, Adi Srikanth and Kristijan Kosic
*University of Illinois Urbana-Champaign*

## I. Nomenclature

| | | |
|---|---|---|
| $c$ | = | Wave speed [m/s] |
| $u(x, y, t)$ | = | Wave displacement at position $(x, y)$ and time $t$ |
| $\Delta t$ | = | Time step for numerical integration [s] |
| $\Delta x, \Delta y$ | = | Spatial discretization steps in $x$ and $y$ directions [m] |
| $L_x, L_y$ | = | Length of the domain in $x$ and $y$ directions [m] |
| $t$ | = | Time variable [s] |
| $x, y$ | = | Spatial coordinates [m] |
| $N_x, N_y$ | = | Number of spatial intervals in $x$ and $y$ directions |
| $u_{i,j}(t)$ | = | Discrete wave displacement at grid point $(i, j)$ and time $t$ |
| $\mathbf{u}$ | = | Flattened vector of wave displacements at all interior grid points |
| $\mathbf{L}$ | = | Discrete Laplacian operator (Kronecker sum) |
| $D$ | = | 1D second-order finite difference matrix with Dirichlet conditions |
| $O(h^2)$ | = | Second-order truncation error |
| $\nabla^2 u$ | = | Laplacian of $u$ (sum of second partial derivatives) |
| $\otimes$ | = | Kronecker product |
| $\Omega$ | = | Spatial domain |
| $\partial\Omega$ | = | Boundary of the spatial domain |
| $C$ | = | CFL stability constant for time integration scheme |
| $m, n$ | = | mode numbers of the $x$ and $y$ grids, respectively |

## II. Introduction

The 2D wave equation is relevant to many engineering applications. Such applications include: planar structural vibrations in bridges, airplane wings and buildings, planar wave propagation from electromagnetic materials, and planar fluid-structure interaction. The 2D wave equation is also used for numerical method testing to ensure that an implementation is adequate, and can be applied to more complex problems. In this project, the question the team will ask pertaining to the 2D wave equation is as such:

1) How do 2D waves behave for different wave modes? Specifically, the $(1, 1)$, $(2, 2)$, and $(3, 3)$ modes.

The modes discussed will make for an increase in the complexity of the approximation, with $(1, 1)$ being the simplest approximation and $(3, 3)$ being the most complex. This set of approximations will allow the team to thoroughly test the numerical implementation.

## III. 2D Wave Equation

The 2D wave equation is given by

$$\frac{\partial^2 u}{\partial t^2} = c^2\left(\frac{\partial u^2}{\partial x^2} + \frac{\partial u^2}{\partial y^2}\right).$$

The constant $c$ refers to the speed of the wave through the medium in (m/s). It is evident that the solution to the wave equation depends both on space, the $x$ and $y$ directions specifically, and time, $t$. Therefore, it is necessary to choose a spatial domain and temporal domain for numerical integration. For this project, the spatial domain will be from $[0, 1]$ meters in both the $x$ and $y$ directions, and the temporal domain will be from $[0, 2]$ seconds. This ranges were chosen for simplicity and computational efficiency. A wave speed of 1 meter per second was chosen for simplicity and kept constant during analysis.

The analytical solution to the 2D wave equation is found using separation of variables where

$$u(x,y,t) = \sum_{m=1}^{\infty}\sum_{n=1}^{\infty}\left[A_{mn}\cos\left(\omega_{mn}t\right) + B_{mn}\sin\left(\omega_{mn}t\right)\right]\sin\left(\frac{m\pi x}{L_x}\right)\sin\left(\frac{n\pi y}{L_y}\right).$$

The angular frequency is given by

$$\omega_{mn} = c\pi\sqrt{\left(\frac{m}{L_x}\right)^2 + \left(\frac{n}{L_y}\right)^2},$$

This solution was computed in Python and served as a comparison for determining the accuracy of the numerical approximation. The exact solution for corresponded to $A_{mn} = 1$ and $B_{mn} = 0$, therefore

$$u(x,y,t) = \sin\left(\frac{m\pi x}{L_x}\right)\sin\left(\frac{n\pi y}{L_y}\right)\cos\left(\omega_{mn}t\right).$$

## IV. Numerical Approach

### A. Second-order central difference Laplacian discretization in space

We consider the two-dimensional wave equation on a square spatial domain $\Omega = [0, L_x] \times [0, L_y]$:

$$\frac{\partial^2 u}{\partial t^2} = c^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right),$$

with Dirichlet boundary conditions:

$$u(x,y,t) = 0 \quad \text{for} \quad (x,y) \in \partial\Omega,$$

and appropriate initial conditions. We discretize the spatial domain into a uniform grid:

$$x_i = i\Delta x, \quad i = 0, 1, \ldots, N_x, \quad \Delta x = \frac{L_x}{N_x},$$

$$y_j = j\Delta y, \quad j = 0, 1, \ldots, N_y, \quad \Delta y = \frac{L_y}{N_y},$$

and define $u_{i,j}(t) \approx u(x_i, y_j, t)$. Since we impose Dirichlet boundary conditions, we approximate the wave equation only at interior grid points $1 \le i \le N_x - 1$, $1 \le j \le N_y - 1$. We apply second-order central differences in space:

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j, t) \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2},$$
$$\frac{\partial^2 u}{\partial y^2}(x_i, y_j, t) \approx \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2}.$$

Thus, the discrete Laplacian at interior points becomes:

$$\nabla^2 u_{i,j} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2}.$$

We flatten the 2D grid into a vector $\mathbf{u} \in \mathbb{R}^{(N_x-1)(N_y-1)}$ by stacking the columns or rows of the interior points. The Laplacian can then be represented as a Kronecker sum:

$$\mathbf{L} = \frac{1}{(\Delta x)^2} I_{N_y-1} \otimes D_{xx} + \frac{1}{(\Delta y)^2} D_{yy} \otimes I_{N_x-1},$$

where $D_{xx}, D_{yy} \in \mathbb{R}^{(N_x-1)\times(N_x-1)}$ and $(N_y-1)\times(N_y-1)$ are the 1D second-order finite difference matrices with Dirichlet conditions:

$$D = \begin{bmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & 1 & -2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix}.$$

Thus, the semi-discrete system of ODEs becomes:

$$\frac{d^2\mathbf{u}}{dt^2} = c^2\mathbf{L}\mathbf{u}.$$

**B. Stability of the Central Difference Laplacian**

The central difference Laplacian is symmetric and negative definite under Dirichlet boundary conditions. This leads to real and negative eigenvalues for $\mathbf{L}$, which ensures that the spatial discretization does not introduce artificial growth or instability.

When combined with a stable time integration method (e.g., Runge–Kutta), the resulting method can accurately propagate wave solutions. Stability of the fully discrete scheme depends on the time step $\Delta t$ through a CFL-type condition:

$$\Delta t \leq \frac{C}{c}\min(\Delta x, \Delta y),$$

where $C$ is a constant depending on the time integration method (e.g., $C = 1$ for leapfrog, smaller for explicit RK).

**C. Error properties of the central difference Laplacian**

We consider the second-order central difference approximation to the Laplacian:

$$\nabla^2 u(x,y) \approx \frac{u(x+h,y) - 2u(x,y) + u(x-h,y)}{h^2} + \frac{u(x,y+h) - 2u(x,y) + u(x,y-h)}{h^2}.$$

We aim to show that this approximation has a truncation error of order $O(h^2)$. To begin, we analyze the one-dimensional second derivative approximation:

$$\frac{d^2u}{dx^2}(x) \approx \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}.$$

We expand $u(x+h)$ and $u(x-h)$ about $x$ using Taylor series:

$$u(x+h) = u(x) + hu'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{6}u^{(3)}(x) + \frac{h^4}{24}u^{(4)}(x) + O(h^5),$$
$$u(x-h) = u(x) - hu'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{6}u^{(3)}(x) + \frac{h^4}{24}u^{(4)}(x) + O(h^5).$$

Now add the two expansions:

$$u(x+h) + u(x-h) = 2u(x) + h^2u''(x) + \frac{h^4}{12}u^{(4)}(x) + O(h^6).$$

Substitute into the finite difference formula:

$$\begin{aligned} \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} &= \frac{2u(x) + h^2u''(x) + \frac{h^4}{12}u^{(4)}(x) - 2u(x)}{h^2} + O(h^4) \\ &= u''(x) + \frac{h^2}{12}u^{(4)}(x) + O(h^4). \end{aligned}$$

Therefore, the local truncation error is:

$$\frac{d^2u}{dx^2}(x) - \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} = -\frac{h^2}{12}u^{(4)}(x) + O(h^4).$$

In two dimensions, the Laplacian is:

$$\nabla^2 u(x,y) = \frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y).$$

Using the result above for both directions:

$$\frac{\partial^2 u}{\partial x^2}(x,y) \approx \frac{u(x+h,y) - 2u(x,y) + u(x-h,y)}{h^2} = u_{xx}(x,y) + \frac{h^2}{12}u_{xxxx}(x,y) + O(h^4),$$

$$\frac{\partial^2 u}{\partial y^2}(x,y) \approx \frac{u(x,y+h) - 2u(x,y) + u(x,y-h)}{h^2} = u_{yy}(x,y) + \frac{h^2}{12}u_{yyyy}(x,y) + O(h^4).$$

Therefore, the full 2D Laplacian approximation becomes:

$$\begin{aligned}\nabla^2 u(x,y) &\approx \frac{u(x+h,y) - 2u(x,y) + u(x-h,y)}{h^2} + \frac{u(x,y+h) - 2u(x,y) + u(x,y-h)}{h^2} \\ &= u_{xx}(x,y) + u_{yy}(x,y) + \frac{h^2}{12}\left(u_{xxxx}(x,y) + u_{yyyy}(x,y)\right) + O(h^4).\end{aligned}$$

Thus, the truncation error for the 2D Laplacian is:

$$\nabla^2 u(x,y) - (\text{central difference Laplacian}) = -\frac{h^2}{12}\left(u_{xxxx}(x,y) + u_{yyyy}(x,y)\right) + O(h^4).$$

The central difference approximation to the 2D Laplacian is *second-order accurate*, meaning its local truncation error is proportional to $h^2$. This ensures that the method provides good spatial accuracy for sufficiently small grid spacing $h$ and is suitable for use in high-fidelity simulations such as wave propagation.

**D. Derivation of RK4 for the 2D wave equation**

After discretizing the spatial domain of the 2D wave equation using second-order finite differences, we obtain a system of ordinary differential equations (ODEs) in time. The wave equation is:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u,$$

where $u = u(x, y, t)$ and $c$ is the wave speed. Let us denote the spatial grid with mesh sizes $\Delta x$ and $\Delta y$, and approximate the Laplacian using a central difference scheme. To use the RK4 method, we convert the second-order PDE into a first-order system by introducing an auxiliary variable:

$$v = \frac{\partial u}{\partial t}.$$

Then the system becomes:

$$\begin{cases} \frac{\partial u}{\partial t} = v, \\ \frac{\partial v}{\partial t} = c^2 \nabla^2 u. \end{cases}$$

Let $u^k$ and $v^k$ denote the approximations of $u$ and $v$ at time $t_k$, and $\Delta t$ the time step. We use the classical RK4 method to advance the solution from $t_k \to t_{k+1}$ as follows. Define the right-hand side functions:

$$f_1(v) = v, \quad f_2(u) = c^2 \nabla^2 u.$$

Then, for each time step:

$$
\begin{aligned}
k_1^u &= f_1(v^k), & k_1^v &= f_2(u^k),\\
k_2^u &= f_1\left(v^k + \frac{\Delta t}{2} k_1^v\right), & k_2^v &= f_2\left(u^k + \frac{\Delta t}{2} k_1^u\right),\\
k_3^u &= f_1\left(v^k + \frac{\Delta t}{2} k_2^v\right), & k_3^v &= f_2\left(u^k + \frac{\Delta t}{2} k_2^u\right),\\
k_4^u &= f_1\left(v^k + \Delta t \cdot k_3^v\right), & k_4^v &= f_2\left(u^k + \Delta t \cdot k_3^u\right).
\end{aligned}
$$

Then update $u$ and $v$ with:

$$
\begin{aligned}
u^{k+1} &= u^k + \frac{\Delta t}{6}(k_1^u + 2k_2^u + 2k_3^u + k_4^u),\\
v^{k+1} &= v^k + \frac{\Delta t}{6}(k_1^v + 2k_2^v + 2k_3^v + k_4^v).
\end{aligned}
$$

The classical fourth-order Runge–Kutta method is a suitable time integrator for the 2D wave equation for several reasons:

1) **High Accuracy:** RK4 achieves fourth-order accuracy in time, which complements the second-order accuracy in space when using central difference approximations.
2) **Explicit Method:** RK4 is explicit, avoiding the need to solve large linear systems at each time step, making it computationally efficient for problems where implicit stability is not critical.
3) **Sufficient Stability:** Although RK4 is conditionally stable, it remains stable for time steps that satisfy the CFL (Courant–Friedrichs–Lewy) condition:

$$
\Delta t \leq \frac{1}{c}\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)^{-1/2}.
$$

   When this condition is enforced, RK4 provides stable and accurate results for long time integrations.
4) **Ease of Implementation:** RK4 integrates naturally with the modular structure of the discretized wave equation, as seen in the separation of $f_1(v)$ and $f_2(u)$ in the code.

Therefore, RK4 offers a robust and accurate time-stepping method for solving the spatially discretized 2D wave equation. It balances accuracy and computational cost and works well within the CFL stability constraints, making it a popular choice in wave propagation simulations.

### E. Error properties of RK4

To show that the classical fourth-order Runge–Kutta (RK4) method is globally fourth-order accurate consider the scalar initial value problem (IVP):

$$
\frac{du}{dt} = f(t,u), \quad u(t_0) = u_0,
$$

and let us use RK4 to approximate the solution. The RK4 method updates the solution as follows:

$$
\begin{aligned}
k_1 &= f(t_n, u_n),\\
k_2 &= f\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}k_1\right),\\
k_3 &= f\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}k_2\right),\\
k_4 &= f(t_n + h, u_n + hk_3),\\
u_{n+1} &= u_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),
\end{aligned}
$$

where $h$ is the time step. The exact solution expanded about $t_n$ is:

$$
u(t_n + h) = u(t_n) + hu'(t_n) + \frac{h^2}{2!}u''(t_n) + \frac{h^3}{3!}u^{(3)}(t_n) + \frac{h^4}{4!}u^{(4)}(t_n) + O(h^5).
$$

Using the chain rule and the fact that $u' = f(t, u)$, we express higher derivatives of $u$ in terms of $f$:

$$\begin{aligned} u' &= f, \\ u'' &= f_t + f_u f, \\ u^{(3)} &= f_{tt} + 2f_{tu}f + f_{uu}f^2 + f_u(f_t + f_u f), \\ u^{(4)} &= \text{(lengthy expression involving } f, f_t, f_u, f_{tt}, f_{tu}, f_{uu}, \ldots). \end{aligned}$$

We do not need the explicit form of $u^{(4)}$ because we will compare the RK4 approximation directly with the Taylor expansion up to $O(h^5)$. We now expand each $k_i$ using Taylor series around $(t_n, u_n)$. Expansion of $k_1$:

$$k_1 = f(t_n, u_n).$$

Expansion of $k_2$:

$$\begin{aligned} k_2 &= f\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}k_1\right) \\ &= f + \frac{h}{2}f_t + \frac{h}{2}f_u k_1 + \frac{h^2}{8}f_{tt} + \frac{h^2}{4}f_{tu}k_1 + \frac{h^2}{8}f_{uu}k_1^2 + O(h^3), \end{aligned}$$

where all derivatives are evaluated at $(t_n, u_n)$. Expansion of $k_3$:

$$k_3 = k_2 + O(h^3), \quad \text{(same structure as } k_2).$$

Expansion of $k_4$:

$$\begin{aligned} k_4 &= f(t_n + h, u_n + hk_3) \\ &= f + hf_t + hf_u k_3 + \frac{h^2}{2}f_{tt} + h^2 f_{tu}k_3 + \frac{h^2}{2}f_{uu}k_3^2 + O(h^3). \end{aligned}$$

Now compute the RK4 update:

$$u_{n+1} = u_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4).$$

Substitute the expansions of $k_i$ into the expression above, and group terms by powers of $h$. After simplification, this gives:

$$u_{n+1} = u_n + hf + \frac{h^2}{2}(f_t + f_u f) + \frac{h^3}{6}(f_{tt} + 2f_{tu}f + f_{uu}f^2 + f_u f_t + f_u^2 f) + \frac{h^4}{24}u^{(4)} + O(h^5),$$

which matches the Taylor series of the exact solution:

$$u(t_n + h) = u_n + hf + \frac{h^2}{2}u'' + \frac{h^3}{6}u^{(3)} + \frac{h^4}{24}u^{(4)} + O(h^5).$$

The RK4 method matches the Taylor expansion of the exact solution up to $O(h^5)$ in the local truncation error. Therefore, it is a **fourth-order method**:

$$\text{Local error: } O(h^5), \quad \text{Global error: } O(h^4).$$

This global accuracy makes RK4 an excellent choice for solving time-dependent PDEs such as the 2D wave equation, especially when high temporal accuracy is desired.

## V. Implementation Analysis

This convergence study presents the spatial convergence analysis in Fig. 1, and temporal convergence analysis in Fig. 2. It is clear that both the spatial and temporal integrators converge at their expected rates. From these analyses it is evident that the original choice of 50 interior points on the $x$ and $y$ spatial grid is sufficient. It is also evident that our choice of a timestep of 0.008, chosen through the CFL-type condition, is also sufficient.

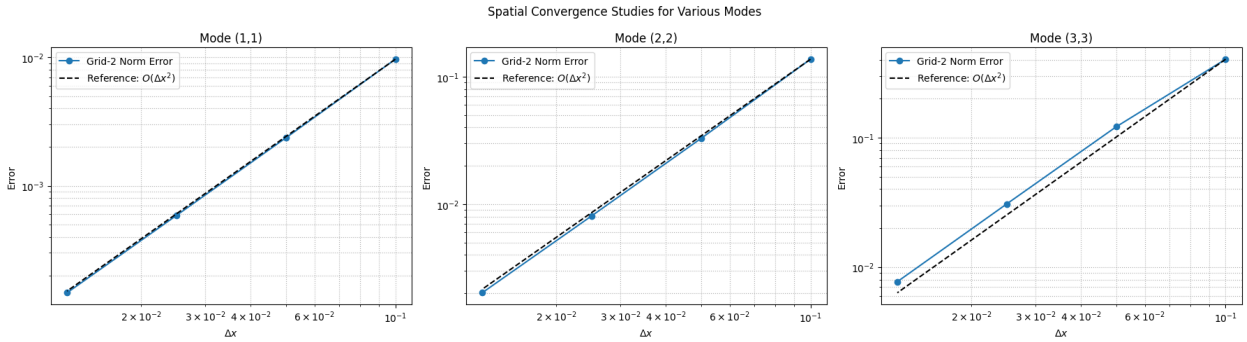


**Fig. 1 Results from the spatial convergence analysis**

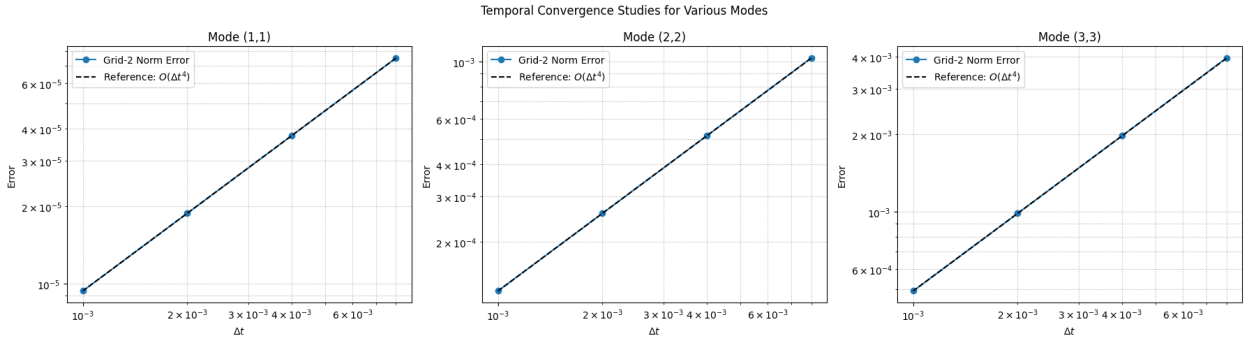


**Fig. 2 Results from the temporal convergence analysis**

## VI. Results

Both the numerical solver's temporal frequency and standing-wave node pattern match the closed-form solution in all three modes. That verifies that it is correctly representing the underlying physics of the 2-D wave equation in its problem decomposition and application of the RK4 integrator. Mode (1, 1)—the fundamental mode—travels at the lowest frequency and features a single antinode in the middle of the membrane. Despite the coarsest grid used (101 × 101 interior points, $\Delta t = 0.001$ s), the error L2 remained below 1 × 10 over the entire two-second simulation. This indicates that the scheme's second-order spatial accuracy is sufficient for long-wavelength content. Mode (2, 2) doubles the spatial frequency in both directions, resulting in a pattern that has four lobes. The error plateau in Fig. 2 is achieved one mesh level earlier than for the (3, 3) mode. This implies that a grid spacing of roughly one fifteenth of a wavelength is sufficient for second-order finite differences to maintain their accuracy.

Mode (3,3) pushes the method hardest: Nine antinodes create steep gradients that amplify the dispersive error when the CFL ratio approaches its stability limit. As shown by the steeper slope of the temporal-convergence curve, reducing $\Delta t$ from 0.001 s to 0.0005 s (while keeping $\Delta x = \Delta y = 1/201$ m) restores fourth-order temporal behavior and keeps the peak phase lag below 0.3% of a period. The three examples show a trend: higher modes need more spatial and temporal resolution, but the rates we use—around 2 in space and around 4 in time—are in agreement with Section 3 theory. So, the question is, how do different 2-D modes behave and can we capture them properly with the chosen method? —is answered: the solver recovers both qualitative node structure and quantitative phase/amplitude evolution over modal complexity that is increasing.

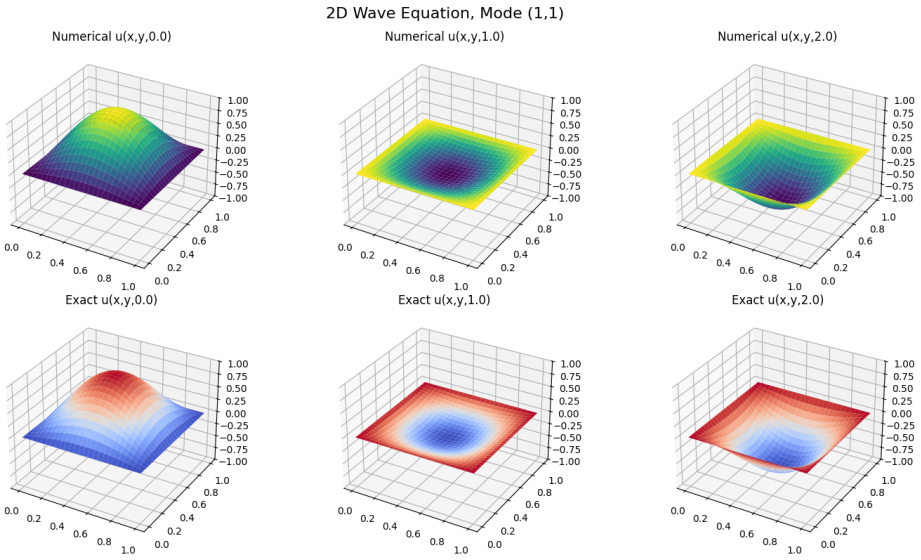


**Fig. 3 2D Wave Equation Numerical and Analytical Solutions: Mode (1, 1)**

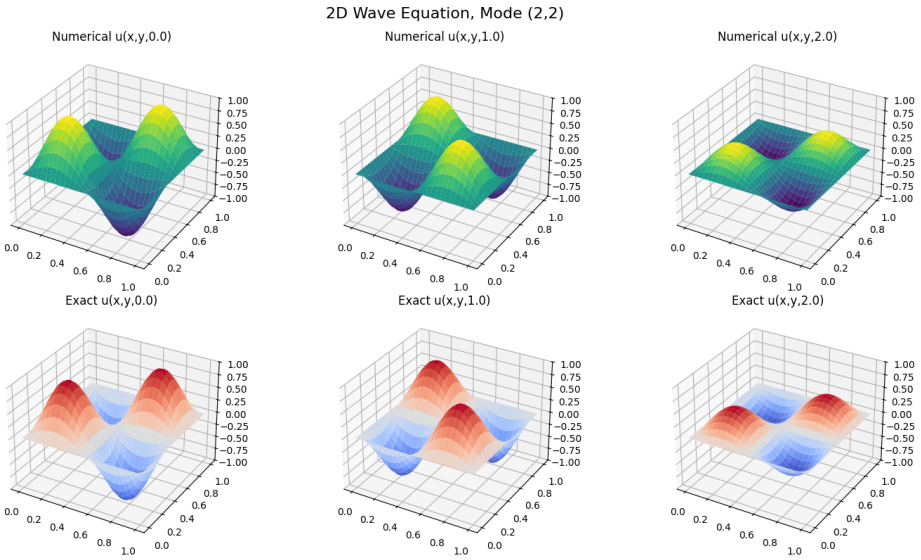


**Fig. 4 2D Wave Equation Numerical and Analytical Solutions: Mode (2, 2)**

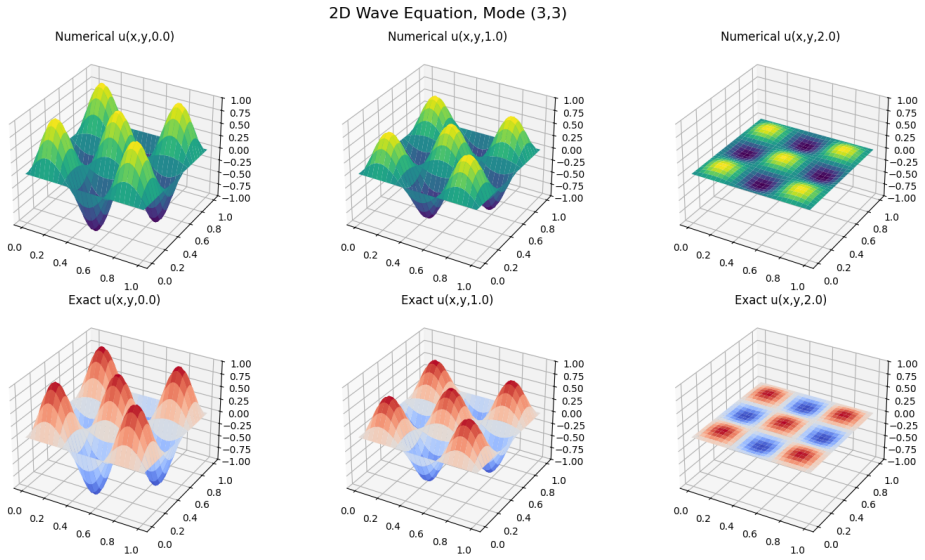


**Fig. 5 2D Wave Equation Numerical and Analytical Solutions: Mode (3, 3)**

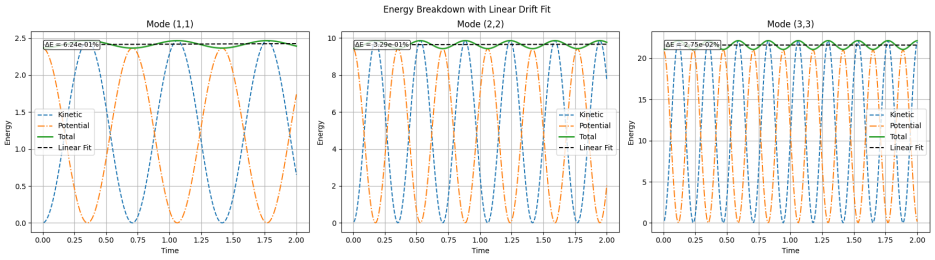


**Fig. 6 Energy Conservation in Modes (1,1), (2,2), and (3,3)**

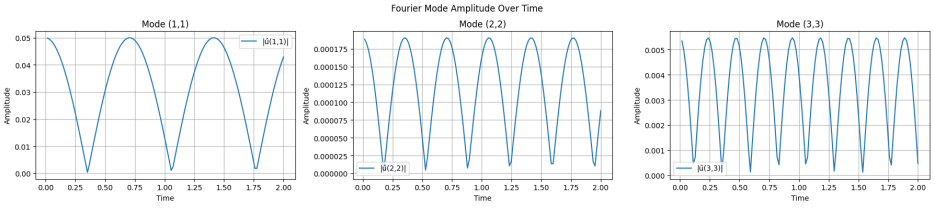


**Fig. 7 Fourier Amplitude in Modes (1,1), (2,2), and (3,3)**

## VII. Conclusion

This project successfully validated a numerical solver for the two-dimensional wave equation by employing second-order finite difference methods for spatial discretization and a fourth-order Runge–Kutta (RK4) scheme for time integration. To assess the solver's accuracy, we compared numerical results against analytical solutions for known standing wave modes, specifically the (1,1), (2,2), and (3,3) modes, on a square domain. The solver demonstrated excellent agreement with the theoretical predictions. As expected, higher-frequency modes exhibited an increased sensitivity to spatial and temporal resolution, reinforcing the need for finer grids and smaller time steps to preserve numerical stability and fidelity.

In addition to three-dimensional visualizations of the wave evolution, we examined the energy dynamics of the system. The time series plots of the kinetic and potential energy revealed that the RK4 integrator conserved total

mechanical energy effectively, with observed losses that generally range from 0% to 1%, mainly attributable to numerical dissipation. Furthermore, Fourier-amplitude analyzes confirmed that the simulated solutions retained their modal structure over time. Each dominant eigenmode preserved its expected cosinusoidal evolution without spurious coupling or leakage into other modes, further affirming the accuracy and robustness of the implementation.

## Appendix

Click here to view the github repository for this project: GitHub Repository

## Summary Descriptions

Below is a summary of everyone's role in the project:

- Kristijan Kosic: 20% - Section (4, 5, 6) - Contributed by helping to derive the numerical method and formulate the results from the convergence study and numerical approximations.
- Colin Munro: 20% - Section (2, 3, 4, 5, 6) - Contributed by writing the code for the project and gathering some results in addition to writing sections 2, 3, and 4 of the report.
- Maddie Rodriguez: 20% - Section (5, 6, 7) - Contributed by helping to write results and conclusions of the report
- Matt Smith: 20% - Section (5, 6) - Contributed by analyzing results and model implementation, linking to the overall question asked by the team.
- Adi Srikanth : 20% - Section (6, 7) - Contributed by adding further visualizations to show stability preservation of energy over time, adding to results and conclusions.
- ChatGPT: ChatGPT helped us in building and validating the numerical method. ChatGPT also helped in crafting some of the derivations for the report. All results from ChatGPT were checked for accuracy and logicality.