Numerical Solution to the 2D Wave Equation on a Complex Domain

MACM 416 - Final Project Presentation

Claire Curry, Thahn Do

Simon Fraser University

Fall 2024

The Wave Equation:

$$u_{tt} = c^2 \Delta u, \ \vec{x} \in \Omega$$

The Wave Equation:

$$u_{tt} = c^2 \Delta u, \ \vec{x} \in \Omega$$

In 2D:

$$u_{tt} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \ (x, y) \in \Omega$$

The Wave Equation

Properties

The constant c represents the “wave speed”.

Figure: Water droplet and propagating waves

Boundary conditions determine wave interactions at the boundary $\partial\Omega$

$$u \mid_{\partial\Omega} = 0$$

Homogeneous Dirichlet conditions, wave inverts at the boundary.

$$\nabla u \cdot \hat{n} \mid_{\partial\Omega} = 0$$

Homogeneous Neumann conditions, wave reflects without inverting.

$$u_{tt} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \ (x, y) \in \Omega$$

Boundary conditions

$$\nabla u \cdot \hat{n} \mid_{\partial\Omega} = 0$$

Initial conditions

$$u(x, y, 0) = \exp\left(-(x-a)^2 - (y-b)^2\right)$$
$$u_t(x, y, 0) = 0$$

The Wave Equation

Domain Ω

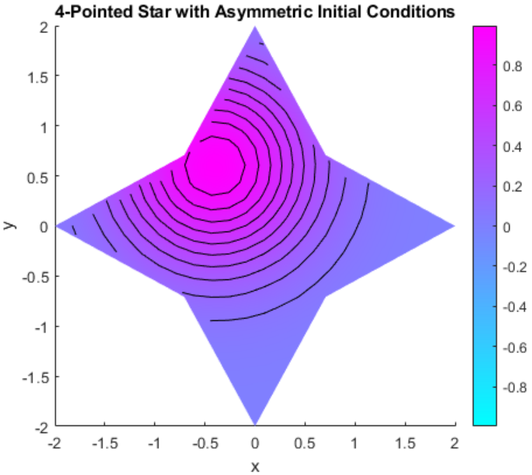


Figure: Asymmetric initial conditions for the wave equation on a 4-point sta

The Test Problem

Forced Wave in 1-D

$$u_{tt} = u_{xx} + \sin(t)$$

Boundary conditions

$$u(x,0) = \frac{\partial}{\partial t}u(x,0) = 0$$

Initial conditions

$$u(0,t) = u(l,t) = 0$$

Exact solution using Fourier series

$$u(x,t) = \sum_{n=1}^{\infty} 2\left(\frac{1-(-1)^n}{\pi n(\pi^2 n^2 - 1)}\right)\left(\sin(t) - \frac{1}{\pi n}\sin(n\pi t)\right)\sin(n\pi x)$$

Numerical Approximation Methods

Time Discretization

$$\frac{\partial^2 u}{\partial t^2}\Big|_{t=t_n} = \frac{u(t_{n+1}) - 2u(t_n) + u(t_{n-1})}{\Delta t^2} + O(\Delta t^2).$$

- Explicit time stepping
- Low computational cost allowing for larger time ranges
- Order Δt^2 accuracy
- We take $\Delta t = 10^{-3}$ for optimal range of stable resolutions

Linear finite elements

Find U^{n+1} such that for all $v \in \mathcal{V}$

$$\int_\Omega \frac{1}{\Delta t^2} v U^{n+1} dx + \int_\Omega -\frac{2}{\Delta t^2} v U^n dx + \frac{1}{\Delta t^2} v U^{n-1} dx - \nabla v \cdot \nabla U^n dx - v f dx = 0.$$

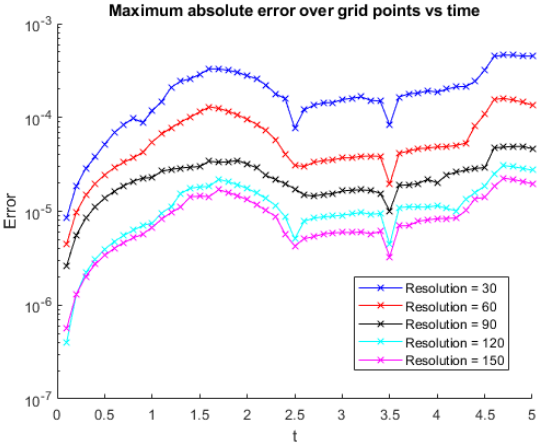
Where $U^{n+1} \approx u(x, y, t_{n+1})$

- Adaptable to arbitrary domains
- Approximately $O(h^2)$ where h is the maximum diameter of the triangular mesh
- Total consistency $O(\Delta t^2 + h^2)$

Test Problem Analysis

Approximations compared to truncated Fourier series

$$u(x,t) = \sum_{n=1}^{10^6} 2\left(\frac{1-(-1)^n}{\pi n(\pi^2 n^2 - 1)}\right)\left(\sin(t) - \frac{1}{\pi n}\sin(n\pi t)\right)\sin(n\pi x)$$



Hard Problem Analysis

How do we test convergence without an exact solution?

- Approximations observe an “energy” principle
- Change in approximations between resolution values decreases as resolution increases

The unforced wave equation

$$u_{tt} = c^2 \Delta u$$

Energy is constant

$$E = \int_\Omega u_{tt}^2 + c^2 |\nabla u|^2 dx$$

We measure the "total energy variation" as our benchmark.

Table: Difference of minimum and maximum energy for the approximation with symmetric initial condition

	Total Energy Variation (Symmetric Initial Conditions)
Resolution 30	1.0000e-05
Resolution 60	3.0000e-05
Resolution 90	4.0000e-05

We measure "Maximum Difference" and "Average Difference" away from the boundary

Table: Maximum difference and average difference between approximations with asymmetric initial conditions at varying resolutions

Resolution	Maximum Difference	Average Difference
$R_1 = 15,\ R_2 = 30$	0.0306	0.0025
$R_1 = 30,\ R_2 = 60$	0.0197	0.0012
$R_1 = 60,\ R_2 = 90$	0.0088	4.6632e-04
$R_1 = 90,\ R_2 = 120$	0.0051	2.6388e-04

Hard Problem Analysis

Efficiency

These approximations can take significant computation times

Table: Computation time taken for asymmetric initial conditions for $t = [0, 1]$

Resolution	Compute Time (seconds)
15	61.84
30	56.94
60	156.22
90	393.22
120	702.47
150	1148.51

Neural Net

An Overview of the Problem

Initial condition:

$$u(x, y, 0) = \phi(x, y) = e^{-x^2-y^2}$$

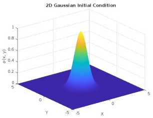


Figure: Gauss Initial condition

Boundary condition:

$$\frac{\partial u}{\partial n} = 0$$

Analytical solution

$$u(x, y, t) = exp[-(\sqrt{x^2 + y^2} - t)^2]$$

Neural Net

First attempt: writing my own Loss function

- Have to use `dlArray` (deep learning array) class
- Must define its own Loss function.
- Must define its own Network Architecture.
- Must manage Numerical Error (rounding) carefully

Neural Net

Second Attempt: using Levenberg-Marquardt algorithm

Produces somewhat decent answer but the training time is painfully long with just 10 nodes and 3 hidden layers, and 50 training iteration.

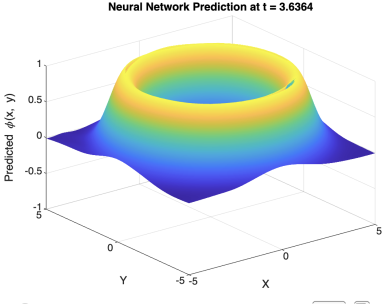


Figure: Prediction

Frame Title

Error

The error is super high to be reliable, it is just there to give intuition how the solution looks like.

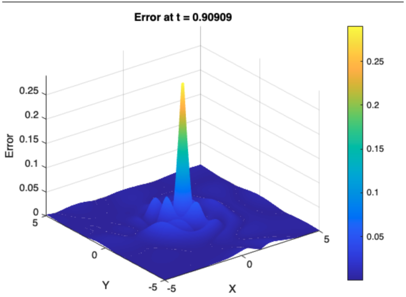


Figure: Error