# Simulation of PDEs in Polar Coordinates

## 1. Introduction

This document explains the simulation of Partial Differential Equations (PDEs) in polar coordinates. It focuses on implementing both the Heat Equation and the Wave Equation in a circular domain using finite difference methods. The code is modularized into five Python scripts for clarity and pedagogy.

## 2. Theory

### 2.1 Heat Equation

The heat equation in polar coordinates (r, θ) without angular dependency simplifies to:  
  
∂u/∂t = c² [∂²u/∂r² + (1/r) ∂u/∂r + (1/r²) ∂²u/∂θ²]  
  
where u is the temperature distribution and c is the thermal diffusivity.

### 2.2 Wave Equation

The wave equation in polar coordinates is given by:  
  
∂²u/∂t² = c² [∂²u/∂r² + (1/r) ∂u/∂r + (1/r²) ∂²u/∂θ²]  
  
where u is the displacement and c is the wave speed.

## 3. Implementation Overview

To simulate these equations, we discretize the domain into radial and angular steps using finite difference methods. The simulation is split into five Python files for clarity:

1. geometry\_and\_initial\_conditions.py – Defines the grid and initial conditions.

2. laplacian\_in\_polar.py – Computes the Laplacian operator in polar coordinates.

3. boundary\_and\_plotting.py – Applies boundary conditions and sets up the 3D plotting.

4. simulate\_heat\_wave.py – Implements the numerical simulation for Heat and Wave equations.

5. main\_simulation\_driver.py – The driver script to configure and run the simulation.

## 4. Code Skeleton

### 4.1 geometry\_and\_initial\_conditions.py

Contains definitions of initial functions and grid creation.

### 4.2 laplacian\_in\_polar.py

Implements the polar Laplacian for 2D functions.

### 4.3 boundary\_and\_plotting.py

Sets boundary conditions and plotting axes.

### 4.4 simulate\_heat\_wave.py

Core simulation logic for Heat and Wave PDEs using finite difference.

### 4.5 main\_simulation\_driver.py

Brings all modules together and calls the simulation function with appropriate parameters.

## 5. How to Run

1. Ensure all five Python files are in the same directory.  
2. Install required libraries using:  
 pip install numpy matplotlib  
3. Run the main driver script using:  
 python main\_simulation\_driver.py