

Computational Visualization of Two-Dimensional Wave Interference Using Python

Abstract

Wave interference is a fundamental phenomenon in classical physics arising from the principle of superposition. While the underlying mathematics is well established, conceptual understanding often remains abstract for students and early researchers. In this work, a two-dimensional computational model of wave interference is developed using Python. By simulating two coherent point sources emitting harmonic waves, clear regions of constructive and destructive interference are visualized through intensity patterns. The model demonstrates how path difference and phase relationships give rise to interference fringes analogous to those observed in classical optics experiments such as Young's double-slit experiment. This study highlights the effectiveness of computational visualization as a pedagogical and analytical tool in physics.

Keywords

Wave interference, Superposition principle, Computational physics, Python simulation, Classical waves

1. Introduction

Interference is one of the most striking consequences of the wave nature of physical phenomena. It appears in diverse contexts, including water waves, sound waves, and electromagnetic radiation. The phenomenon arises when two or more coherent waves overlap in space, producing regions of enhanced or diminished amplitude depending on their relative phase.

Traditionally, interference is introduced analytically through trigonometric expressions or experimentally through demonstrations such as Young's double-slit experiment. However, these approaches may obscure the spatial structure of interference patterns, particularly in two dimensions. Computational methods offer an effective means to bridge this gap by enabling direct visualization of wave superposition.

In this paper, a two-dimensional numerical simulation of wave interference is presented using Python. The objective is not experimental verification, but conceptual clarity: to show how simple mathematical expressions, when implemented computationally, naturally generate interference fringes.

2. Theoretical Framework

2.1 Principle of Superposition

For linear wave systems, the principle of superposition states that when multiple waves overlap, the resultant displacement at any point is the algebraic sum of the individual displacements. If two waves are described by functions $y_1(x, y, t)$ and $y_2(x, y, t)$, the total wave is given by:

$$y(x, y, t) = y_1(x, y, t) + y_2(x, y, t)$$

Interference effects arise entirely from this linear addition.

2.2 Two-Dimensional Harmonic Waves

Consider two coherent point sources located at positions S_1 and S_2 in a two-dimensional plane. Each source emits a monochromatic wave of wavelength λ and angular frequency ω . The wave from each source can be expressed as:

$$y_i(x, y, t) = A \sin(kr_i - \omega t)$$

where r_i is the distance from the i -th source to the observation point, $k = \frac{2\pi}{\lambda}$ is the wave number, and A is the amplitude.

The resultant wave is obtained by summing the contributions from both sources. Regions of constructive interference occur when the path difference $\Delta r = r_2 - r_1$ satisfies $\Delta r = n\lambda$, while destructive interference occurs for $\Delta r = (n + 1/2)\lambda$, where n is an integer.

2.3 Intensity Representation

In many physical contexts, detectors measure not the instantaneous displacement but the time-averaged intensity, which is proportional to the square of the wave amplitude. Accordingly, the intensity is defined as:

$$I(x, y) \propto y^2(x, y, t)$$

This quantity is used to visualize the interference pattern.

3. Computational Methodology

The simulation is implemented using Python with the NumPy and Matplotlib libraries. A two-dimensional spatial grid is constructed, and two coherent point sources are placed symmetrically along the horizontal axis. For each grid point, the distances to the sources are computed, and the corresponding wave amplitudes are evaluated using the harmonic wave equation.

The resultant wave is obtained through superposition, and the intensity distribution is calculated as the square of the resultant amplitude. The final interference pattern is visualized using a color map, where brighter regions correspond to higher intensity.

3.1 Python Implementation

The following Python code was used to generate the interference patterns presented in this work:

```
import numpy as np
import matplotlib.pyplot as plt
```

```

# Create spatial grid
x = np.linspace(-10, 10, 500)
y = np.linspace(-10, 10, 500)
X, Y = np.meshgrid(x, y)

# Wave parameters
wavelength = 1.0
k = 2 * np.pi / wavelength
omega = 2 * np.pi
t = 0

# Source positions
s1 = (-2, 0)
s2 = (2, 0)

# Distances from sources
r1 = np.sqrt((X - s1[0])**2 + (Y - s1[1])**2)
r2 = np.sqrt((X - s2[0])**2 + (Y - s2[1])**2)

# Wave functions
wave1 = np.sin(k * r1 - omega * t)
wave2 = np.sin(k * r2 - omega * t)

# Superposition and intensity
resultant = wave1 + wave2
intensity = resultant**2

# Visualization
plt.imshow(intensity, extent=[-10, 10, -10, 10], origin='lower')
plt.colorbar(label='Intensity')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Two-Dimensional Wave Interference Pattern')
plt.show()

```

This implementation computes the time-independent intensity pattern corresponding to the superposition of two coherent waves, which represents the physically observable interference distribution.

4. Results

The simulation produces a clear two-dimensional interference pattern characterized by alternating bright and dark fringes. The pattern exhibits symmetry about the perpendicular bisector of the line joining the two sources. Regions of high intensity correspond to constructive interference, while nodal lines of near-zero intensity correspond to destructive interference.

The spatial distribution of the fringes qualitatively matches theoretical predictions and experimental observations from classical wave interference setups. The results confirm that interference patterns emerge naturally from the superposition principle without requiring additional assumptions.

5. Discussion

The computational model captures the essential physics of two-source interference while remaining simple and transparent. Although the model neglects factors such as damping, finite source size, and boundary effects, it successfully illustrates the role of path difference and phase coherence.

Such simulations are particularly valuable in educational contexts, where they can complement analytical derivations and laboratory demonstrations. Moreover, the approach can be readily extended to include more sources, different geometries, or time-dependent visualizations.

6. Limitations

The present model assumes ideal point sources emitting perfectly coherent waves in a homogeneous medium. Effects such as attenuation, noise, and nonlinearity are not included. Consequently, the simulation is intended for conceptual and illustrative purposes rather than precise experimental prediction.

7. Conclusion

A two-dimensional computational visualization of wave interference has been presented using a simple Python-based model. By directly implementing the principle of superposition, the simulation reproduces characteristic interference fringes associated with constructive and destructive interference. This work demonstrates how computational tools can enhance understanding of fundamental physical phenomena and serve as an accessible entry point into computational physics.

References

1. Hecht, E. *Optics*. Addison-Wesley, 4th ed. (2002).
2. Young, T. "The Bakerian Lecture: Experiments and Calculations Relative to Physical Optics." *Philosophical Transactions of the Royal Society of London* **94**, 1–16 (1804).
3. Newman, M. E. J. *Computational Physics*. CreateSpace Independent Publishing (2012).