Independent Study Report: Understanding and Simulating Standing Waves

Mehak Dhawan
Jumeirah College
mehak.dhawan0206@gmail.com

Abstract

These working notes study an exploration of standing waves, from basic wave theory to its applications and computational simulations that incorporate realistic noise modeling. The work uses a structured three-phase approach: (1) foundational research, (2) mathematical derivations, and (3) Python-based simulations. The foundational research broadly includes the mathematical principles behind wave interference, how stationary patterns form through superposition, and their importance across multiple areas of science such as musical acoustics, material science (phonons in crystals), quantum mechanics (electron wave functions in atomic orbitals), advanced optics (laser cavity modes and LiDAR), and spectroscopy. Next, the mathematical analysis shows the boundary conditions that govern wave confinement. Finally, we use computational python code enhanced with Gaussian noise to create realistic visualizations that connect theory with physical behavior.

Introduction

Waves are one of the most fundamental natural phenomena, appearing across nearly every branch of science and engineering. A wave can be defined as a disturbance or oscillation that moves through a medium or space, transferring energy without carrying matter along with it. Waves can take many forms, including mechanical waves (such as sound and water waves), electromagnetic waves (such as light and radio waves), and matter waves described by quantum mechanics. This study focuses on standing waves, which represent a unique type of wave pattern that forms when two identical waves travel in opposite directions and interfere with each other.

Standing waves or stationary waves occur when two waves of the same amplitude and frequency, travel in opposite directions through the same medium. Their interference creates nodes (points of zero amplitude) and antinodes (points of maximum amplitude). This generates a oscillatory pattern instead of propagating through space. Unlike traveling waves that move energy forward, standing waves represent an ongoing exchange between kinetic and potential energy at fixed points. Thus, they are important for understanding resonance, vibration modes, and wave confinement in systems ranging from musical instruments to quantum mechanical orbitals.

In this study, we do a fundamental understanding of standing waves and then we look at some computational simulations where we see how the noise affects the presence of standing waves. We see that in standing waves with gaussian noise, we can force the nodes to be fixed and this gives us standing waves with gaussian noise.

Theoretical Background

Wave Fundamentals

Wave motion is the process by which energy is transferred through oscillating disturbances in a medium or field. Despite their differences, all types of waves share similar mathematical descriptions defined by the following wave parameters:

- **Amplitude (A)**: Maximum displacement from equilibrium position, directly proportional to energy content
- **Wavelength (λ)**: Spatial distance between consecutive identical points in the wave pattern
- **Frequency (f)**: Number of complete oscillations per unit time, measured in Hertz (Hz)
- **Period (T)**: Time required for one complete oscillation, where $T = 1/f$
- **Wave speed (v)**: Rate of wave pattern propagation, given by the fundamental relationship $v = f\lambda$
- **Phase**: Position within the oscillation cycle at any given time and location

The basic wave propagation equation represents the fundamental partial differential equation:

$$\partial^2 u/\partial t^2 = v^2\, \partial^2 u/\partial x^2$$

where $u(x,t)$ describes the displacement at position x and time t, and v represents the wave speed in the medium.

Standing Wave

Standing waves are formed by the superposition of two waves moving in opposite directions through the same medium with the same amplitude, frequency, and wavelength. This interference creates a distinct stationary pattern.

The mathematical description starts with two sinusoidal waves as follows:

Wave 1 (rightward propagation): $u_1(x,t) = A \sin(kx - \omega t)$
Wave 2 (leftward propagation): $u_2(x,t) = A \sin(kx + \omega t)$

where,

- A = amplitude of each wave
- $k = 2\pi/\lambda$ = wave number
- $\omega = 2\pi f$ = angular frequency
- x = spatial position
- t = time

Applying the superposition principle:

$$u(x,t) = u_1(x,t) + u_2(x,t) = A \sin(kx - \omega t) + A \sin(kx + \omega t)$$

Using the trigonometric identity $\sin(\alpha) + \sin(\beta) = 2\sin[(\alpha+\beta)/2]\cos[(\alpha-\beta)/2]$:

$$u(x,t) = 2A \sin(kx) \cos(\omega t)$$

This equation describes the fundamental characteristic of standing waves: complete separation of spatial and temporal dependencies. The spatial function $2A \sin(kx)$ determines the amplitude envelope, and $\cos(\omega t)$ governs the temporal oscillation.

- *Nodes:* Positions where $\sin(kx) = 0$, occurring at $x = n\pi/k = n\lambda/2$ (where n = 0, 1, 2, ...). These points maintain zero displacement for all time due to the destructive interference.

- *Antinodes:* Positions where $|\sin(kx)| = 1$, occurring at $x = (2n+1)\pi/2k = (2n+1)\lambda/4$. These points show maximum amplitude oscillation due to the constructive interference.

- The spatial separation between adjacent nodes is $\lambda/2$. Similarly, the spatial separation between adjacent antinodes is $\lambda/2$.

Standing waves often occur in systems with boundaries such as a string fixed at both ends. These boundaries cause restrictions that lead to quantized wavelengths and frequencies.

For a string of length L with fixed ends:

- Boundary conditions: $u(0,t) = 0$ and $u(L,t) = 0$
- These requirements demand $\sin(kL) = 0$
- Solutions exist when $kL = n\pi$ (where n = 1, 2, 3, ...)

This quantization condition yields:

- Allowed wavelengths: $\lambda_n = 2L/n$
- Allowed frequencies: $f_n = nv/2L = nf_1$, where $f_1 = v/2L$ represents the fundamental frequency.

Thus, the quantized standing wave solutions can be expressed as:

$$u_n(x,t) = 2A \sin(n\pi x/L) \cos(\omega_n t)$$

Here, each mode n:

- Contains n-1 internal nodes (points of zero displacement).
- Contains n antinodes (points of maximum displacement).
- Produces unique spatial patterns that correspond to harmonic frequencies.

Applications of Standing Waves

Musical Instruments:

The physics of standing waves is important to the sound produced by musical instruments. It governs both **pitch** as well as **tone quality**.

String Instruments

Instruments like guitars, violins, and pianos rely on vibrating strings that form standing wave patterns. Here, nodes occur at the fixed ends of the string, and antinodes occur along the string's length, depending on the harmonic mode. The fundamental frequency (the lowest pitch a string can produce) is given by:

$$f_1 = (1/2L)\sqrt{(T/\mu)}$$

where,

- T = string tension
- μ = linear mass density (mass per unit length)
- L = string length

Musicians can change pitch in several ways:

- *Fretting:* Shortening the string length (L), which increases frequency.
- *Tuning:* Adjusting the tension (T) of the string.
- *String choice:* Using different materials or thicknesses (μ) for varied pitch ranges.

Wind Instruments

Air columns inside flutes, clarinets, and organ pipes also sustain standing waves, but the boundary conditions depend on whether the pipe is open or closed:

- Closed pipes (one end closed): Fundamental frequency $f_1 = v/4L$
- Open pipes (both ends open): Fundamental frequency $f_1 = v/2L$

where v is the speed of sound in air (~343 m/s at room temperature).

Harmonics and Timbre
Higher-order standing waves (harmonics) enrich the sound. The **relative strength of harmonics** gives each instrument its characteristic **timbre**. For instance, a violin and a flute may play the same fundamental frequency, but their harmonic content makes them sound distinct.

Material Science and Phonons:

In crystalline solids, atoms vibrate in well-ordered patterns that form **standing waves also** known as **phonons**. These phonons play an important role in determining the **thermal, electrical, and mechanical properties** of materials.

Thermal Conductivity
Phonons are the primary carriers of heat in **insulators** and **semiconductors**. By understanding and controlling phonon standing waves, scientists can:

- **Design thermoelectric materials**
- **Develop thermal barrier coatings**
- Improve heat spreaders

Mechanical Properties
Phonon interactions also impact the following mechanical properties:

- **Elastic constants:** how a crystal responds to applied stress.
- **Thermal expansion:** how a material's dimensions change with temperature.
- **Sound velocity:** the speed of acoustic waves traveling through the material.

Quantum Mechanics:

Standing waves are extremely important to understand the **matter waves** and **atomic structure**.

Atomic Orbitals

Electrons in atoms behave like standing waves such that their behavior is described by the solutions to the Schrödinger equation. The wave function ψ(r) determines electron probability distributions, with $|\psi(r)|^2$ representing probability density. For hydrogen-like atoms, the radial wave functions exhibit standing wave characteristics with:

- Nodes: Positions where ψ(r) = 0, representing zero electron probability
- Antinodes: Regions of maximum $|\psi(r)|^2$, where the electron is most likely to be found.

Because the wave function must satisfy boundary conditions (approaching zero at infinity), electrons occupy only certain energy levels:

$$E_n = -13.6 \text{ eV}/n^2 \text{ (for hydrogen)}$$

where n is the principal quantum number. This explains the discrete energy levels observed in atomic spectra.

Quantum Wells, Wires, and Dots

Engineered nanostructures confine electrons, creating artificial standing wave systems:

- ***Quantum wells:*** One-dimensional confinement, widely used in **laser diodes** and **LEDs**.
- ***Quantum wires:*** Two-dimensional confinement that enhances electrical conduction.
- ***Quantum dots:*** Three-dimensional confinement that acts like “artificial atoms” with **tunable electronic and optical properties**.

Laser Physics and Optical Cavities:

The functioning of a laser relies on the creation of **standing electromagnetic waves** within an **optical cavity**, which is typically formed by two reflecting mirrors. These standing waves set the allowed modes of oscillation and, therefore, the laser’s possible frequencies.

For a cavity of length L, resonance occurs when the round trip produces constructive interference, which requires $L=m\lambda/2$, where m represents the longitudinal mode number. The corresponding frequency of the m-th mode is given by: $\nu_m = mc/2nL$

where,

- c = speed of light
- n = refractive index of the medium inside the cavity

The separation between adjacent longitudinal modes, known as the free spectral range (FSR), is given by $\Delta\nu = c/2nL$. For instance, in a typical He–Ne laser (L ≈ 30 cm) filled with air (n ≈ 1), this spacing is $\Delta\nu = (3.00\times10^8 \text{ m/s})/(2\times1\times0.30 \text{ m}) = 500$ MHz.

When the cavity contains different sections, each with its own refractive index, the effective optical length becomes $n_1L_1 + n_2L_2$. For example, in a composite cavity made of a 10 cm laser rod (L_1 = 10 cm, n_1 = 1.82) and a 20 cm air gap (L_2 = 20 cm, n_2 = 1.00), the effective optical length is: $n_1L_1 + n_2L_2 = (1.82)(0.10) + (1.00)(0.20) = 0.382$ m. This leads to a reduced mode spacing of $\Delta v = (3.00\times10^8)/(2\times0.382) = 393$ MHz. This simple calculation illustrates how cavity design directly shapes the frequency spectrum and operating characteristics of a laser.

LiDAR Remote Sensing:

Light Detection and Ranging (LiDAR) uses pulsed laser light to make precise distance measurements and create detailed environmental maps. While LiDAR primarily operates with **traveling pulses**, standing wave principles are crucial in **laser design** and **signal processing**.

System Operation: LiDAR emits short laser pulses and measures return times from reflected surfaces:

- *Time-of-flight measurement:* Distance = (speed of light × time)/2
- *Multiple returns*: Single pulses can reflect from different heights (vegetation canopy, understory, ground)
- *Waveform analysis*: Return signal shape provides information about reflecting surface characteristics

Applications:

- *Topographic mapping:* High-resolution elevation models with centimeter accuracy
- *Forest structure analysis:* Canopy height, biomass estimation, vertical vegetation distribution
- *Atmospheric studies:* Aerosol and cloud particle detection using backscatter measurements
- *Autonomous vehicles:* Real-time obstacle detection and navigation

Standing Wave Relevance: While LiDAR primarily involves traveling electromagnetic pulses, standing wave physics applies to:

- *Laser cavity design:* Ensuring stable, coherent light sources for measurement precision
- *Interference effects:* Multiple reflections creating standing wave patterns affect measurement accuracy
- *Signal processing:* Understanding wave interference helps optimize detection algorithms

Methodology

Computational Simulation and Noise Modeling

The final phase transitioned from theoretical analysis to practical computational implementation, developing Python-based simulations for dynamic visualization and realistic environmental modeling.

Simulation Framework: The implementation utilized key Python libraries:

- **NumPy**: Efficient numerical array operations and mathematical functions
- **Matplotlib**: Dynamic visualization and animation capabilities

Gaussian Noise Implementation:

Realistic environmental effects were incorporated through Gaussian noise modeling because real standing wave systems experience perturbations. These sources can be from Thermal fluctuations causing microscopic displacement variations, Environmental vibrations transmitted through mounting systems, Material property variations along the medium, Measurement uncertainties and instrumental limitations and Damping effects from air resistance and internal friction.

Implementation Details:

Numerically, we discretized the string along x on the interval [0, L] with 500 points and simulated six fundamental periods using a stable time step equal to one-eight-hundredth of the third-harmonic period. The displacement field was built as a sum of fixed-end normal modes, then perturbed with Gaussian (zero-mean) noise whose local standard deviation scaled with the absolute displacement ($a \times |u(x,t)|$). To mimic slow environmental drift, we added a weak separable exponential correlation in space and time. Light damping was applied with an exponential envelope so that each period reduces amplitude by a fixed fraction, emulating a finite quality factor (Q). For validation, node locations were obtained from the spatial RMS minima of the displacement and matched theory within a few millimetres, while spectral peaks from an antinode time series appeared at integer multiples of the fundamental frequency (which equals v divided by 2L).

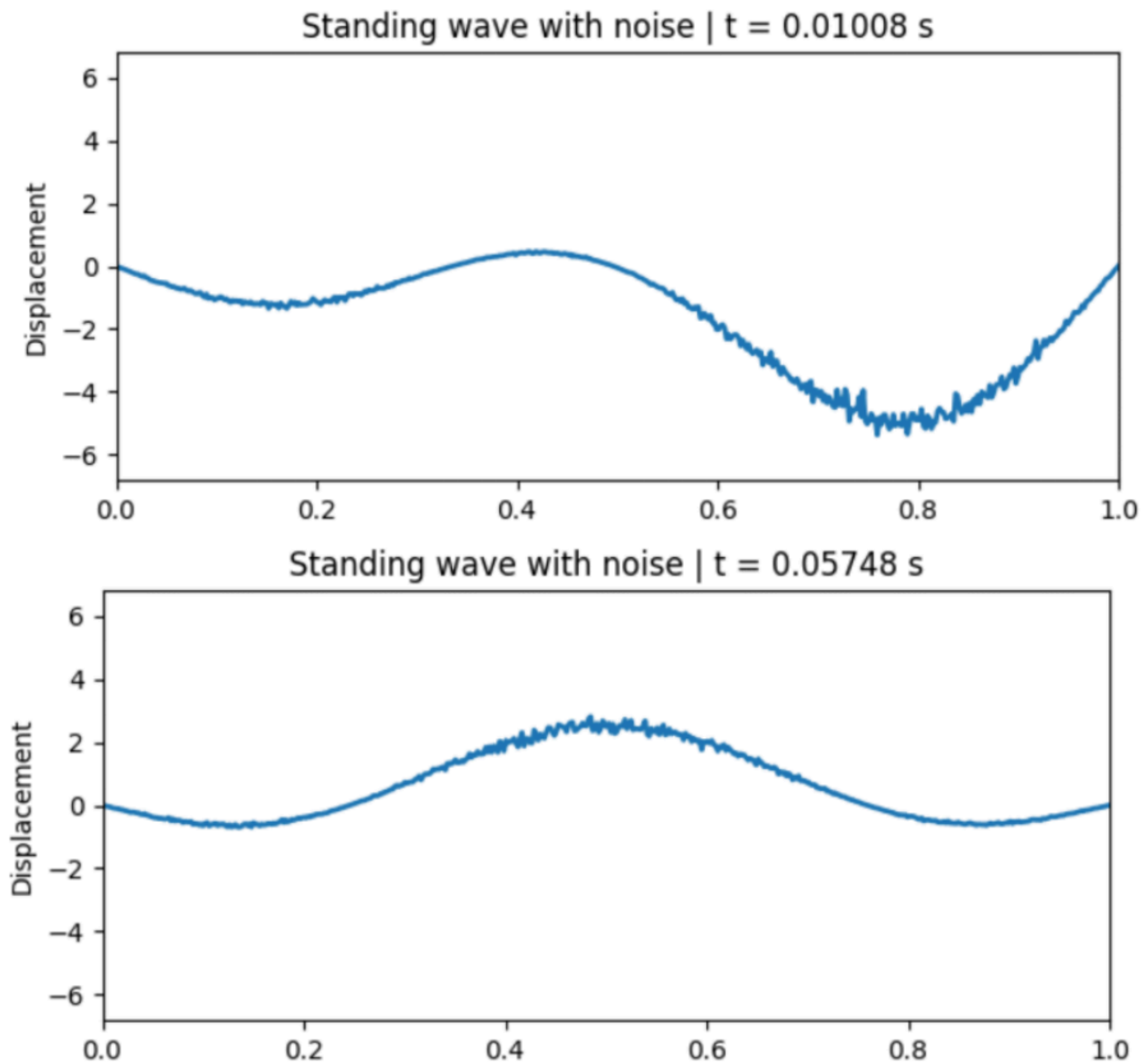

Figure 1: This shows the temporal evolution of the wave with noise. The nodes remain constant but there is noise superimposed in it. In the following figures we see for t=0.010s and t=0.057s.

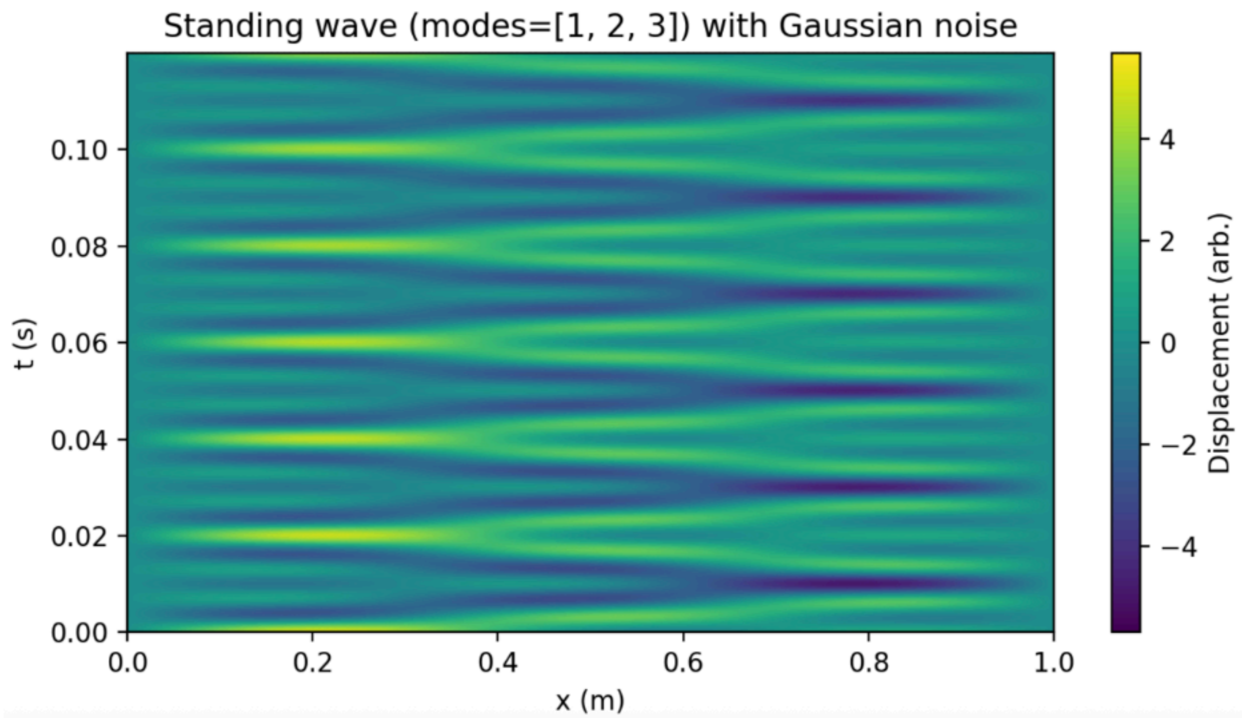

Figure 2: Space–time heatmap of displacement for a fixed string with modes n = 1–3 (x on the horizontal axis, time on the vertical). Dark bands mark nodes (persistent at x = 0 and x = 1 and internally from modal superposition); bright bands mark antinodes. Gaussian, amplitude-proportional noise adds mild jitter without obscuring the standing-wave pattern.

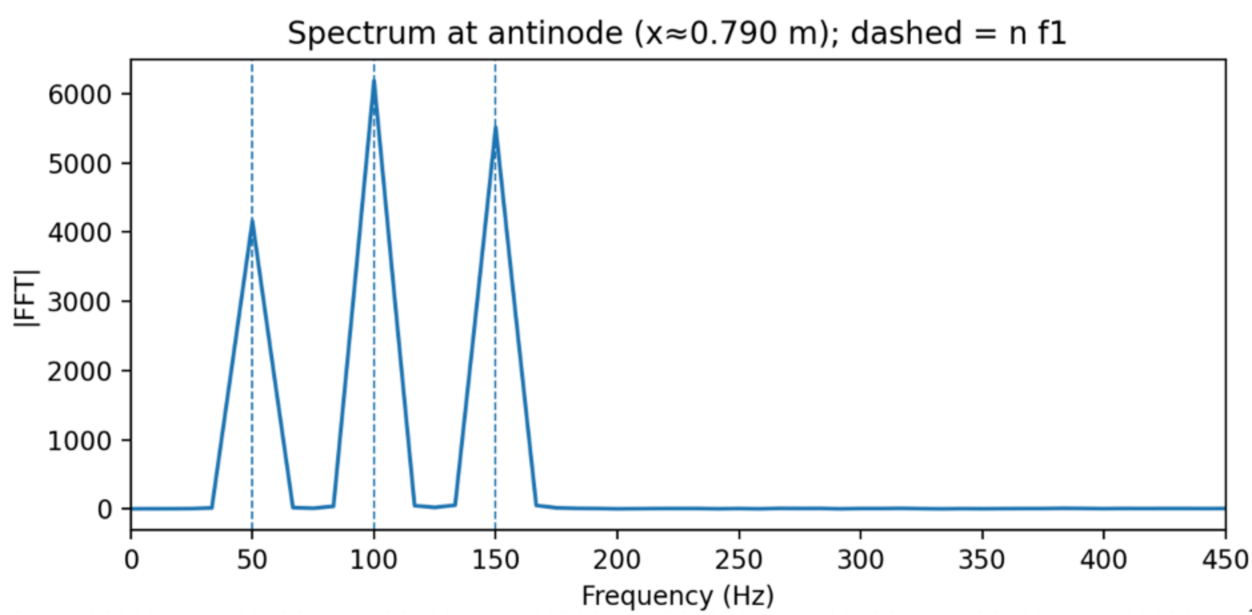

Figure 3: Amplitude spectrum at an antinode (x ≈ 0.79 m). Peaks at 50, 100, and 150 Hz align with the predicted harmonics $n \times f_1$ ($f_1 = v/2L = 50$ Hz); dashed lines mark the theoretical frequencies. The low broadband floor reflects added Gaussian noise.

Conclusion

This study provided a thorough understanding of standing wave phenomena. The three-phase methodology worked efficiently as the foundational research provided a conceptual framework. Next, the mathematical derivations established clear analytical understanding. Finally, the computational simulations linked theory with observable behavior. Also, adding Gaussian noise modeling brought important realism and prepared for experimental applications, where ideal conditions do not exist.

Overall, standing waves play a crucial role in understanding resonance, vibration modes, wave confinement, and energy distribution. This study clearly shows that standing waves support various technologies, including musical instruments, laser systems, quantum devices, and advanced materials. This knowledge builds a strong foundation for further study in physics, engineering, and materials science. It also demonstrates the interconnected nature of scientific knowledge and the importance of theoretical foundations for technological progress.

References:

1. Reif, F. & Larkin, J. *Standing Waves*, MISN-0-433, Project PHYSNET (2002).
2. Bécherrawy, T. *Mechanical and Electromagnetic Vibrations and Waves*. Wiley (2013).
3. Shirahatti, U.S. & Crocker, M.J. “Standing Waves,” in M.J. Crocker (ed.), *Handbook of Acoustics*. Wiley-Interscience (1998).
4. White, H.E. & White, D.H. *Physics and Music: The Science of Musical Sound*. Dover/Courier (2014).
5. Taylor, G.I. (1953) (Proc. Roy. Soc. A 218, 44–59).
6. Hess, P. (2002) (*Physics Today* 55(3), 42–47).
7. Salomon, C. et al. (1987) (PRL 59, 1659).

8. Harris, L. & Loeb, A.L. *Introduction to Wave Mechanics*. Addison-Wesley (1963).

9. Siegman, A.E. *Lasers*. University Science Books (1986).

10. Svelto, O. *Principles of Lasers*, 5th ed., Springer, 2010.

11. Wasser, L. “The Basics of LiDAR – Light Detection and Ranging.” NEON (tutorial, updated).

12. Cortel, A. (2021) (*The Physics Teacher* 59(6), 462–463).

13. Young, R.W. (1952) (*Am. J. Phys.* 20, 177–183).

14. Slater, J.C. (1958) (*Rev. Mod. Phys.* 30, 197).

15. Wandinger, U. (2005) Weitkamp (ed.), *Lidar: Range-Resolved Optical Remote Sensing of the Atmosphere* (Springer).

16. Radinschi, I. et al. (2017) (*Computer Applications in Engineering Education* 25(3), 521–529; DOI:10.1002/cae.21818).

17. Caetano, T. et al. (2022) (*European Journal of Physics* 43(2), 025801; DOI:10.1088/1361-6404/ac4978).

18. Lou, L.F. (Liang-Fu Lou) (2003) (World Scientific).

19. Dong, P. & Chen, Q. (2017) (CRC Press/Taylor & Francis).

Appendix A: Code

```
# Standing wave simulation with Gaussian noise + diagnostics (1D string, fixed ends)

# Outputs:

#   - standing_wave_spacetime.png

#   - standing_wave_rms_nodes.png
```

```
#   - standing_wave_spectrum.png
#   - standing_wave_animation.gif
#   - standing_wave_freq_analysis.csv
#   - standing_wave_node_analysis.csv

import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation, PillowWriter
from scipy.signal import find_peaks
import pandas as pd

# ---------------------------
# Physics helpers
# ---------------------------
def standing_wave(x, t, n, L, A, v):
    """Ideal standing wave for a string fixed at both ends:
    u_n(x,t) = 2A sin(nπx/L) cos(ω_n t), ω_n = nπ v / L"""
    k = n * np.pi / L
    omega = k * v
    return 2 * A * np.sin(k * x) * np.cos(omega * t)

def add_realistic_noise(y, noise_scale, rng):
    """Gaussian noise proportional to local amplitude: largest at antinodes, ~0 at nodes."""
```

```
    return y + (noise_scale * np.abs(y)) * rng.standard_normal(y.shape)


def add_spatiotemporal_correlated_noise(u, corr_space=0.02, corr_time=0.01, rng=None):
    """
    Weakly correlated noise via separable exponential smoothing in x and t.
    u: (T, X) array used only for sizing.
    """
    if rng is None:
    rng = np.random.default_rng(42)
    T, X = u.shape
    eta = rng.standard_normal((T, X))
    # kernel lengths (odd)
    kx_len = max(3, int(X * corr_space)); kx_len += 1 - kx_len % 2
    kt_len = max(3, int(T * corr_time)); kt_len += 1 - kt_len % 2
    x_idx = np.arange(-(kx_len//2), kx_len//2 + 1)
    t_idx = np.arange(-(kt_len//2), kt_len//2 + 1)
    kx = np.exp(-np.abs(x_idx) / (kx_len/6)); kx /= kx.sum()
    kt = np.exp(-np.abs(t_idx) / (kt_len/6)); kt /= kt.sum()
    # convolve separably
    eta_x = np.apply_along_axis(lambda v: np.convolve(v, kx, mode='same'), axis=1,
arr=eta)
    eta_xt = np.apply_along_axis(lambda v: np.convolve(v, kt, mode='same'), axis=0,
arr=eta_x)
    return eta_xt
```

```
def damp(u, gamma_per_period, periods_elapsed):
    """Exponential amplitude decay ~ (1 - gamma) per period."""
    alpha = -np.log(max(1e-9, 1 - gamma_per_period))
    return u * np.exp(-alpha * periods_elapsed)


# ----------------------------
# Config
# ----------------------------
L = 1.0             # string length (m)
v = 100.0           # wave speed (m/s)
A = 1.0             # single traveling-wave amplitude (arb.)
modes = [1, 2, 3]   # harmonics to superpose
noise_scale = 0.05  # relative noise amplitude
use_correlated_noise = True
corr_strength = 0.15# mix of correlated noise vs proportional noise
gamma_per_period = 0.02 # 2% amplitude loss per period (light damping)

Nx = 500                    # spatial resolution
duration_periods = 6.0              # simulate N fundamental periods
dt_factor = 800             # samples per period of highest mode
rng = np.random.default_rng(7)
```

```
# ----------------------------

# Discretization

# ----------------------------

x = np.linspace(0, L, Nx)

n_ref = max(modes)

f1 = v / (2 * L)                # fundamental frequency

f_ref = n_ref * f1              # highest included frequency

T_ref = 1.0 / f_ref

dt = T_ref / dt_factor

T_total = duration_periods * (1.0 / f1)   # simulate same number of FUNDAMENTAL periods

t = np.arange(0, T_total, dt)

Nt = len(t)


# ----------------------------

# Simulate ideal + noise + damping

# ----------------------------

u_ideal = np.zeros((Nt, Nx), dtype=float)

for n in modes:

        u_mode = np.array([standing_wave(x, ti, n, L, A, v) for ti in t])

        periods_elapsed = t * n * f1    # = t / T_n

        u_ideal += damp(u_mode, gamma_per_period, periods_elapsed[:, None])


# proportional noise per frame
```

```
u_noisy = np.empty_like(u_ideal)
for i in range(Nt):
    u_noisy[i] = add_realistic_noise(u_ideal[i], noise_scale, rng)

# add weak correlated component
if use_correlated_noise:
    eta_xt = add_spatiotemporal_correlated_noise(u_ideal, corr_space=0.02,
corr_time=0.01, rng=rng)
    scale_field = np.maximum(1e-8, np.abs(u_ideal))
    u_noisy += corr_strength * noise_scale * eta_xt * scale_field

# ----------------------------
# Diagnostics & analysis
# ----------------------------
# Spatial RMS over time → antinode peaks, node minima
rms_space = np.sqrt((u_noisy**2).mean(axis=0))

# Node estimation: minima of RMS profile (use peaks on the negative)
inv = -rms_space.copy()
inv[0] = inv[-1] = inv.mean()  # avoid boundary artifacts
peaks, _ = find_peaks(inv, prominence=np.std(inv) * 0.15, distance=round(Nx * 0.08))
x_nodes_est = x[peaks]
```

```
# Theoretical union of nodes for the included modes (including ends)
theory_nodes = np.unique(np.concatenate([np.linspace(0, L, n+1) for n in modes]))
theory_nodes = np.unique(np.round(theory_nodes, 6))


# Frequency estimate at strongest antinode (max RMS)
ix_antinode = int(np.argmax(rms_space))
signal = u_noisy[:, ix_antinode]
fs = 1.0 / dt
win = np.hanning(len(signal))
fft = np.fft.rfft(signal * win)
freqs = np.fft.rfftfreq(len(signal), d=dt)
mag = np.abs(fft)


pk_idx, _ = find_peaks(mag, prominence=mag.max()*0.02, distance=5)
freq_peaks = freqs[pk_idx]
freq_peaks = freq_peaks[freq_peaks > 0]


theory_freqs = np.array([n * f1 for n in modes])


def nearest(theory, measured):
    if len(measured) == 0: return np.array([], dtype=int)
    return np.argmin(np.abs(theory[:, None] - measured[None, :]), axis=0)
```

```
if len(freq_peaks) > 0:
    idx_map = nearest(theory_freqs, freq_peaks)
    matched = theory_freqs[idx_map]
    freq_df = pd.DataFrame({
    "Measured_peak_Hz": np.round(freq_peaks, 3),
    "Nearest_theory_Hz": np.round(matched, 3),
    "Abs_error_Hz": np.round(np.abs(freq_peaks - matched), 3),
    "Rel_error_%": np.round(100*np.abs(freq_peaks - matched)/(matched+1e-12), 3),
    }).sort_values("Measured_peak_Hz")
else:
    freq_df =
pd.DataFrame(columns=["Measured_peak_Hz","Nearest_theory_Hz","Abs_error_Hz","Rel_error
_%"])


def pair_nodes(est, theo, tol=2e-3):
    rows = []
    for xe in est:
    idx = np.argmin(np.abs(theo - xe))
    rows.append((xe, theo[idx], np.abs(theo[idx]-xe), np.abs(theo[idx]-xe) <= tol))
    return pd.DataFrame(rows,
columns=["Estimated_x","Nearest_theory_x","Abs_error","Within_2mm?"])


node_df = pair_nodes(x_nodes_est, theory_nodes)
```

```
freq_df.to_csv("standing_wave_freq_analysis.csv", index=False)

node_df.to_csv("standing_wave_node_analysis.csv", index=False)


# ----------------------------

# Plots

# ----------------------------

# 1) Space–time heatmap

plt.figure(figsize=(7, 4))

plt.imshow(u_noisy, aspect='auto', origin='lower', extent=[x.min(), x.max(), t.min(), t.max()])

plt.xlabel("x (m)"); plt.ylabel("t (s)")

plt.title(f"Standing wave (modes={modes}) with Gaussian noise")

cbar = plt.colorbar(); cbar.set_label("Displacement (arb.)")

plt.tight_layout(); plt.savefig("standing_wave_spacetime.png", dpi=200); plt.close()


# 2) RMS profile + estimated nodes + theoretical node lines

plt.figure(figsize=(7, 3.5))

plt.plot(x, rms_space, label="RMS amplitude vs x")

if len(peaks) > 0:

    plt.scatter(x_nodes_est, rms_space[peaks], marker='x', s=40, label="Estimated nodes")

for xn in theory_nodes:

    if 0 < xn < L:

    plt.axvline(xn, linestyle='--', linewidth=0.8)

plt.xlabel("x (m)"); plt.ylabel("RMS amplitude")
```

```
plt.title("Nodes from RMS minima (dashed = theoretical union)")
plt.legend(loc="upper right")
plt.tight_layout(); plt.savefig("standing_wave_rms_nodes.png", dpi=200); plt.close()

# 3) Spectrum at an antinode
plt.figure(figsize=(7, 3.5))
plt.plot(freqs, mag)
for fth in theory_freqs:
    plt.axvline(fth, linestyle='--', linewidth=0.8)
plt.xlim(0, theory_freqs.max() * 3.0)
plt.xlabel("Frequency (Hz)"); plt.ylabel("|FFT|")
plt.title(f"Spectrum at antinode (x≈{x[ix_antinode]:.3f} m); dashed = n f1")
plt.tight_layout(); plt.savefig("standing_wave_spectrum.png", dpi=200); plt.close()

# 4) Animation (GIF, ~120 frames)
fig, ax = plt.subplots(figsize=(7, 3))
line, = ax.plot([], [], lw=2)
ax.set_xlim(0, L)
ax.set_ylim(1.2 * u_noisy.min(), 1.2 * u_noisy.max())
ax.set_xlabel("x (m)"); ax.set_ylabel("Displacement")
ax.set_title("Standing wave with noise (snapshot animation)")

frame_idx = np.linspace(0, Nt-1, 120, dtype=int)
```

```
def init(): line.set_data([], []); return (line,)
def update(i):
    y = u_noisy[frame_idx[i]]
    line.set_data(x, y)
    ax.set_title(f"Standing wave with noise | t = {t[frame_idx[i]]:.5f} s")
    return (line,)

anim = FuncAnimation(fig, update, init_func=init, frames=len(frame_idx), blit=True)
anim.save("standing_wave_animation.gif", writer=PillowWriter(fps=30))
plt.close(fig)

# ---------------------------
# Console summary
# ---------------------------
print("Saved:",
    "standing_wave_spacetime.png,",
    "standing_wave_rms_nodes.png,",
    "standing_wave_spectrum.png,",
    "standing_wave_animation.gif,",
    "standing_wave_freq_analysis.csv,",
    "standing_wave_node_analysis.csv")
print("Summary:",
    {"L": L, "v": v, "A": A, "modes": modes, "noise_scale": noise_scale,
```

```
"gamma_per_period": gamma_per_period, "duration_s": float(T_total),
"dt": float(dt), "Nx": Nx, "Nt": Nt})
```

References

[1] Reif, Fred, and Jill Larkin. "Standing Waves." (1952).

[2] Bécherrawy, T., 2013. *Mechanical and electromagnetic vibrations and waves*. John Wiley & Sons.

[3] U. S. Shirahatti and Malcolm J. Crocker

[4] Reif, F. and Larkin, J., 1952. Standing Waves.

[5] White, H.E. and White, D.H., 2014. *Physics and music: the science of musical sound*. Courier Corporation.

[6] Shirahatti, U.S. and Crocker, M.J., 1998. Standing waves. *Handbook of Acoustics*, p.73.

[7] Taylor, G.I., 1953. An experimental study of standing waves. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, *218*(1132), pp.44-59.

[8] Hess, P., 2002. Surface acoustic waves in materials science. *Physics Today*, *55*(3), pp.42-47.

[9] Salomon, C., Dalibard, J., Aspect, A., Metcalf, H. and Cohen-Tannoudji, C., 1987. Channeling atoms in a laser standing wave. *Physical review letters*, *59*(15), p.1659.

[10] Close, R.A., 1963. An introduction to wave mechanics.

[11] Arieli, R., 2005. The Laser Adventure.

[12] Wasser, L. (2022). *The Basics of LiDAR - Light Detection and Ranging - Remote Sensing*

[13] Cortel, A., 2021. Nodes and Antinodes in Two-Color Chladni Figures. *The Physics Teacher*, *59*(6), pp.462-463.

[14] Young, R.W., 1952. Modes, Nodes, and Antinodes. *American Journal of Physics*, *20*(3), pp.177-183.

[15] Slater, J.C., 1958. Interaction of waves in crystals. *Reviews of modern physics*, *30*(1), p.197.

[16] Wandinger, U., 2005. Introduction to lidar. In *Lidar: range-resolved optical remote sensing of the atmosphere* (pp. 1-18). New York, NY: Springer New York.

[17] Radinschi, I., Fratiman, V., Ciocan, V. and Cazacu, M.M., 2017. Interactive computer simulations for standing waves. *Computer Applications in Engineering Education*, *25*(3), pp.521-529.

[18] Caetano, T., Junior, M.F.R., da Silva, A.P. and Moreira, C.C., 2022. The physics remote laboratory: Implementation of an experiment on standing waves. *European Journal of Physics*, *43*(2), p.025801.

[19] Lou, L.F., 2003. *Introduction to phonons and electrons*. World Scientific.

[20] Dong, P. and Chen, Q., 2017. *LiDAR remote sensing and applications*. CRC Press.