Applications of Logarithmic and Exponential Functions

Mathematical Models and Python Visualizations

February 3, 2026

Introduction

Exponential and logarithmic functions are fundamental in modeling natural and human-made systems. In this document, each real-world scenario is explained mathematically and visualized using Python's `matplotlib` library. Proper axis labels and titles are used to emphasize physical interpretation.

—

1 Population Growth

Mathematical Model

$$P(t) = 50{,}000e^{0.03t}$$

- t: time in years
- $P(t)$: population

Problem

Find the time required for the population to reach 100,000.

$$t = \frac{\ln(2)}{0.03}$$

Python Visualization

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

t = np.linspace(0, 30, 300)
P = 50000 * np.exp(0.03 * t)

plt.plot(t, P, label="Population Growth")
plt.axhline(100000, linestyle="--", label="Target Population")

plt.xlabel("Time (years)")
plt.ylabel("Population")
plt.title("Exponential Population Growth")
plt.legend()
plt.grid(True)
plt.show()
```

Listing 1: Population growth using an exponential model

Code Explanation

The time array is generated using `numpy.linspace`. The population is calculated using the exponential growth equation. A horizontal line indicates the target population, and labeled axes provide physical meaning.

—

2 Radioactive Decay

Mathematical Model

$$N(t) = 1000e^{-0.2t}$$

- t: time in hours
- $N(t)$: remaining radioactive material

Python Visualization

```
t = np.linspace(0, 20, 300)
N = 1000 * np.exp(-0.2 * t)

plt.plot(t, N, label="Radioactive Decay")
plt.axhline(200, linestyle="--", label="200 units")

plt.xlabel("Time (hours)")
plt.ylabel("Remaining Amount")
plt.title("Exponential Radioactive Decay")
plt.legend()
plt.grid(True)
plt.show()
```

Listing 2: Radioactive decay over time

Code Explanation

This plot shows exponential decay. The logarithm is later used to determine decay time for a given remaining amount.

—

3 Sound Intensity and Decibel Scale

Mathematical Model

$$L = 10\log_{10}\left(\frac{I}{I_0}\right)$$

Python Visualization

```
I0 = 1e-12
I = np.logspace(-12, -2, 300)
L = 10 * np.log10(I / I0)

plt.plot(I, L)
plt.xscale("log")

plt.xlabel("Sound Intensity (W/m$^2$)")
```

```
plt.ylabel("Sound Level (dB)")
plt.title("Logarithmic Sound Intensity Scale")
plt.grid(True)
plt.show()
```

Listing 3: Logarithmic sound intensity scale

Code Explanation

Sound intensity spans many orders of magnitude, so a logarithmic x-axis is used. This reflects how humans perceive sound.

—

4 Drug Concentration in Blood

Mathematical Model

$$C(t) = 80e^{-0.15t}$$

Python Visualization

```
t = np.linspace(0, 30, 300)
C = 80 * np.exp(-0.15 * t)

plt.plot(t, C, label="Drug Concentration")
plt.axhline(20, linestyle="--", label="Minimum Effective Level")

plt.xlabel("Time (hours)")
plt.ylabel("Concentration (mg/L)")
plt.title("Drug Elimination Over Time")
plt.legend()
plt.grid(True)
plt.show()
```

Listing 4: Drug concentration decay in bloodstream

Code Explanation

The exponential decay curve models how drugs are metabolized. Logarithms determine safe re-dosing intervals.

—

5 Earthquake Magnitude

Mathematical Model

$$E = 10^M$$

Python Visualization

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

M = np.linspace(0, 9, 300)
E = 10 ** M

plt.plot(M, E)

plt.xlabel("Earthquake Magnitude")
plt.ylabel("Relative Energy Released")
plt.title("Logarithmic Nature of Earthquake Magnitude")
plt.grid(True)
plt.show()
```

Listing 5: Energy released vs earthquake magnitude

Code Explanation

Each unit increase in magnitude corresponds to ten times more energy. The exponential curve explains why large earthquakes are vastly more destructive.

—

Conclusion

Python visualizations reinforce the interpretation of exponential and logarithmic models. Combined with mathematics, they provide insight into real-world systems involving growth, decay, perception, and scaling.