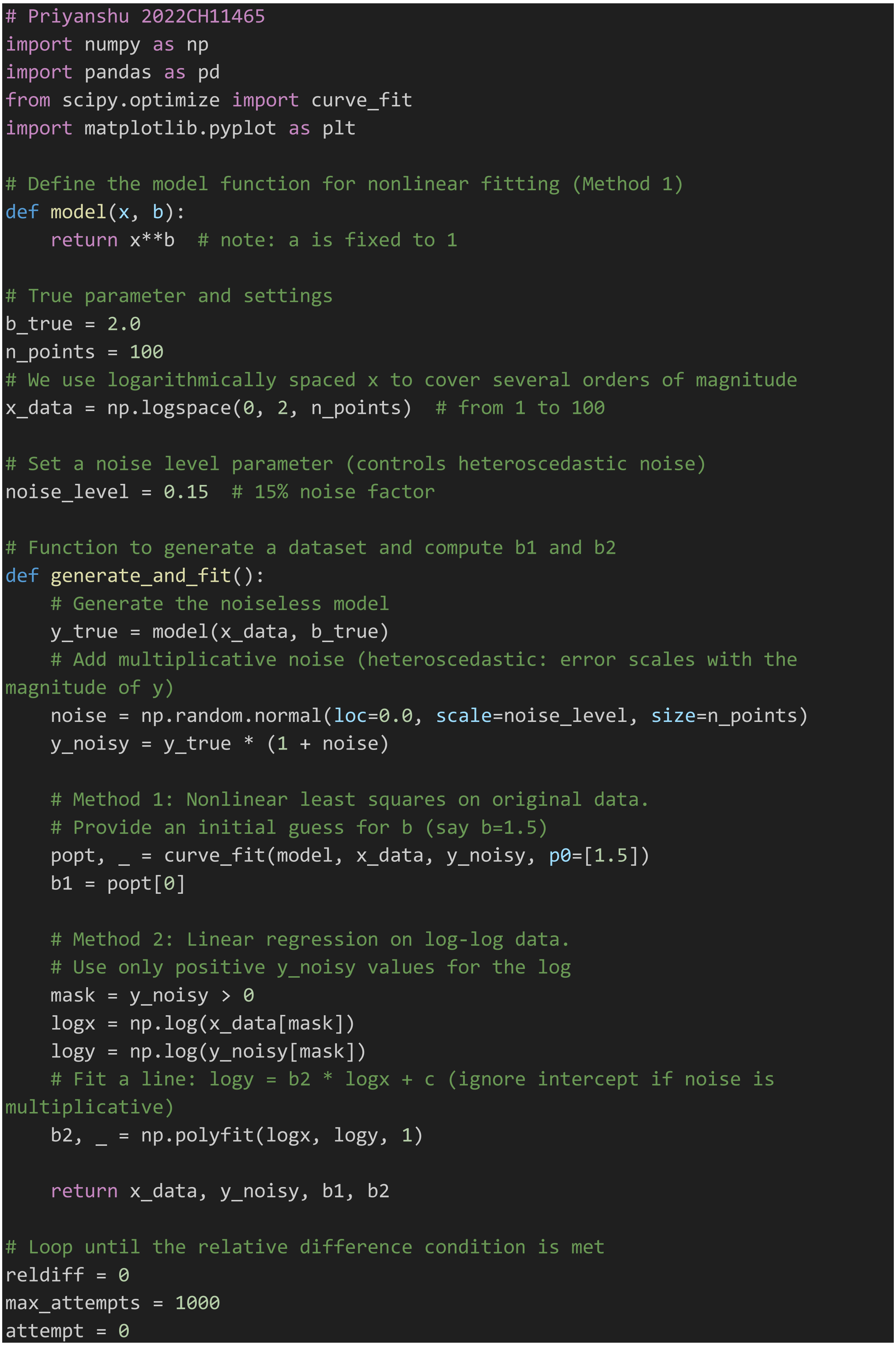

```python
# Priyanshu 2022CH11465
import numpy as np
import pandas as pd
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt

# Define the model function for nonlinear fitting (Method 1)
def model(x, b):
    return x**b  # note: a is fixed to 1

# True parameter and settings
b_true = 2.0
n_points = 100
# We use logarithmically spaced x to cover several orders of magnitude
x_data = np.logspace(0, 2, n_points)  # from 1 to 100

# Set a noise level parameter (controls heteroscedastic noise)
noise_level = 0.15  # 15% noise factor

# Function to generate a dataset and compute b1 and b2
def generate_and_fit():
    # Generate the noiseless model
    y_true = model(x_data, b_true)
    # Add multiplicative noise (heteroscedastic: error scales with the
magnitude of y)
    noise = np.random.normal(loc=0.0, scale=noise_level, size=n_points)
    y_noisy = y_true * (1 + noise)

    # Method 1: Nonlinear least squares on original data.
    # Provide an initial guess for b (say b=1.5)
    popt, _ = curve_fit(model, x_data, y_noisy, p0=[1.5])
    b1 = popt[0]

    # Method 2: Linear regression on log-log data.
    # Use only positive y_noisy values for the log
    mask = y_noisy > 0
    logx = np.log(x_data[mask])
    logy = np.log(y_noisy[mask])
    # Fit a line: logy = b2 * logx + c (ignore intercept if noise is
multiplicative)
    b2, _ = np.polyfit(logx, logy, 1)

    return x_data, y_noisy, b1, b2

# Loop until the relative difference condition is met
reldiff = 0
max_attempts = 1000
attempt = 0
```

```python
while attempt < max_attempts:
    attempt += 1
    x_data, y_noisy, b1, b2 = generate_and_fit()
    reldiff = abs(b1 - b2) / (abs(b1) + abs(b2))
    if reldiff >= 0.05:
        break

print(f"Dataset accepted after {attempt} attempt(s)")
print(f"Method 1 (nonlinear fit) estimated b1: {b1:.4f}")
print(f"Method 2 (log-log linear fit) estimated b2: {b2:.4f}")
print(f"Relative difference: {reldiff:.4f}")

# Save the dataset to a CSV file
df = pd.DataFrame({'x': x_data, 'y': y_noisy})
df.to_csv("fitted_dataset.csv", index=False)
print("Dataset saved to 'fitted_dataset.csv'.")
```