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

def cube_root(n):
    steps = 0
    start_time = time.time()
    x = abs(n)
    guess = x / 3.0
    while True:
        steps += 1
        better_guess = (2 * guess + x / (guess ** 2)) / 3.0
        if abs(guess - better_guess) < 0.000001:
            break
        guess = better_guess
    if n < 0:
        guess = -guess
    end_time = time.time()
    execution_time = end_time - start_time
    return guess, steps, execution_time

def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True

def sum_of_primes(start, end):
    return sum(num for num in range(start, end + 1) if is_prime(num))

n = -27
result, steps, execution_time = cube_root(n)
print(f"The cube root of {n} is approximately {result}")
print(f"Number of steps: {steps}")
print(f"Execution time: {execution_time} seconds")

numbers = [10**i for i in range(1, 10)]
steps_list = []
execution_times = []
for num in numbers:
    _, steps, execution_time = cube_root(num)
    steps_list.append(steps)
    execution_times.append(execution_time)

plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.plot(np.log10(numbers), steps_list, marker='o')
plt.xlabel('Number of digits')
plt.ylabel('Number of steps')
plt.title('Number of steps vs Number of digits')
plt.subplot(1, 2, 2)
plt.plot(np.log10(numbers), execution_times, marker='o')
plt.xlabel('Number of digits')
plt.ylabel('Execution time (seconds)')
plt.title('Execution time vs Number of digits')
plt.tight_layout()
plt.show()

print(f"Sum of prime numbers between 3 and 1000: {sum_of_primes(3, 1000)}")

numbers = [10**i for i in range(1, 5)]
steps_list = []
execution_times = []
for num in numbers:
    start_time = time.time()
    is_prime(num)
    end_time = time.time()
```

```
    execution_time = end_time - start_time
    steps_list.append(int(num**0.5))
    execution_times.append(execution_time)

plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.plot(np.log10(numbers), steps_list, marker='o')
plt.xlabel('Number of digits')
plt.ylabel('Number of steps')
plt.title('Number of steps vs Number of digits')
plt.subplot(1, 2, 2)
plt.plot(np.log10(numbers), execution_times, marker='o')
plt.xlabel('Number of digits')
plt.ylabel('Execution time (seconds)')
plt.title('Execution time vs Number of digits')
plt.tight_layout()
plt.show()
```
