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

# Parameters for the sine wave
sampling_rate = 1000  # Samples per second
frequency = 5         # Frequency of the sine wave in Hz
duration = 1          # Duration in seconds

# Time array
t = np.linspace(0, duration, int(sampling_rate * duration), endpoint=False)

# Create a sine wave signal
signal = np.sin(2 * np.pi * frequency * t)

# Perform Fast Fourier Transform (FFT)
fft_result = np.fft.fft(signal)
fft_magnitude = np.abs(fft_result)  # Magnitude of the FFT
frequencies = np.fft.fftfreq(len(fft_magnitude), 1/sampling_rate)

# Plot the original sine wave signal
plt.figure(figsize=(12, 6))

plt.subplot(2, 1, 1)
plt.plot(t, signal, label='Sine Wave (5 Hz)')
plt.title('Original Sine Wave Signal')
plt.xlabel('Time (s)')
plt.ylabel('Amplitude')
plt.grid()
plt.legend()

# Plot the frequency spectrum
```

```
plt.subplot(2, 1, 2)
plt.plot(frequencies[:len(frequencies)//2], fft_magnitude[:len(frequencies)//2], color='orange',
label='Frequency Spectrum')
plt.title('Frequency Spectrum')
plt.xlabel('Frequency (Hz)')
plt.ylabel('Magnitude')
plt.xlim(0, 50)  # Limit x-axis for better visualization
plt.grid()
plt.legend()

plt.tight_layout()
plt.show()
```

2.import numpy as np

```
from scipy.integrate import quad
def f(x):
   return x**2
a = 0
b = 5
result, error = quad(f, a, b)

print(f"The integral of f(x) = x^2 from {a} to {b} is approximately {result:.4f}")
print(f"Estimated error in the result: {error:.4e}")
```

output:

The integral of f(x) = x^2 from 0 to 5 is approximately 41.6667

Estimated error in the result: 4.6259e-13

3. import numpy as np

```
from scipy.optimize import minimize

# Define the function to minimize
def f(x):
```

```
    return (x - 3)**2 + 2
x0 = 0

# Perform the minimization
result = minimize(f, x0)

# Output the result
printf("The minimum value occurs at x = {result.x[0]:.4f}")
printf("The minimum value of the function is f(x) = {result.fun:.4f}")
```

output:

The minimum value occurs at x = 3.0000

The minimum value of the function is f(x) = 2.0000

4. import numpy as np

```
# Coefficient matrix
A = np.array([[2, 3],
              [4, 1]])

# Right-hand side vector
b = np.array([5, 6])

# Solve the system of equations
solution = np.linalg.solve(A, b)

# Output the result
x, y = solution
printf("The solution is x = {x:.4f}, y = {y:.4f}")
```

output:

The solution is x = 1.3000, y = 0.8000