Applied Computational Science

Homework: Linear Algebra.

Anand Kamble
amk23j@fsu.edu

29th October 2023

3.

In [1]:
```
import numpy as np
```

In [2]:
```
def power_method(A, x0, tol=1e-8, max_iter=1000):
    x = x0
    lambda_prev = 0

    for i in range(max_iter):
        y = np.dot(A, x)
        lambda_max = np.linalg.norm(y, np.inf)
        x = y / lambda_max

        if abs(lambda_max - lambda_prev) < tol:
            break

        lambda_prev = lambda_max

    return lambda_max, x
```

In [3]:
```
def inverse_power_method(A, x0, tol=1e-8, max_iter=1000):
    x = x0
    lambda_prev = 0

    for i in range(max_iter):
        y = np.linalg.solve(A, x)
        lambda_min = np.linalg.norm(y, np.inf)
        x = y / lambda_min

        if abs(lambda_min - lambda_prev) < tol:
            break

        lambda_prev = lambda_min

    return 1 / lambda_min, x
```

In [4]:
```
A = np.array([[4, 4, 0, 0],
              [4, 20, 12, 0],
              [0, 12, 18, 15],
              [0, 0, 15, 61]])

# Initial guess
x0 = np.array([1, 1, 1, 1])

# Power method for largest eigenvalue
lambda_max, eigenvector_max = power_method(A, x0)
```

```
# Inverse power method for smallest eigenvalue
lambda_min, eigenvector_min = inverse_power_method(A, x0)

print(f"Largest Eigenvalue (Power Method): {lambda_max:.4f}")
print(f"Largest Eigenvalue (Exact): 66.0147")
print()
print(f"Smallest Eigenvalue (Inverse Power Method): {lambda_min:.4f}")
print(f"Smallest Eigenvalue (Exact): 1.6655")
```

```
Largest Eigenvalue (Power Method): 66.0147
Largest Eigenvalue (Exact): 66.0147

Smallest Eigenvalue (Inverse Power Method): 1.6655
Smallest Eigenvalue (Exact): 1.6655
```

Condition Number

In [5]:

```
# Calculate the condition number
condition_number = lambda_max / lambda_min

# Display the result
print(f"Condition Number of A: {condition_number:.4f}")
```

```
Condition Number of A: 39.6377
```