

## APPM 4600 Homework 12

5 December 2024

1. (a) We have the system

$$\begin{bmatrix} 6 & 2 & 2 \\ 2 & \frac{2}{3} & \frac{1}{3} \\ 1 & 2 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \\ 0 \end{bmatrix}.$$

Observe that the system is solved by

$$\begin{bmatrix} 6 & 2 & 2 \\ 2 & \frac{2}{3} & \frac{1}{3} \\ 1 & 2 & -1 \end{bmatrix} \begin{bmatrix} 2.6 \\ -3.8 \\ -5 \end{bmatrix} = \begin{bmatrix} -2 \\ 1 \\ 0 \end{bmatrix} \implies \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2.6 \\ -3.8 \\ -5 \end{bmatrix}.$$

- (b) We reduce the system without pivoting in 4 digit arithmetic,

$$\left[ \begin{array}{ccc|c} 6 & 2 & 2 & -2 \\ 2 & \frac{2}{3} & \frac{1}{3} & 1 \\ 1 & 2 & -1 & 0 \end{array} \right] = \left[ \begin{array}{ccc|c} 6.000 & 2.000 & 2.000 & -2.000 \\ 2.000 & 0.6667 & 0.3333 & 1.000 \\ 1.000 & 2.000 & -1.000 & 0.000 \end{array} \right] \implies \left[ \begin{array}{ccc|c} 6.000 & 2.000 & 2.000 & -2.000 \\ 0.000 & 0.000 & -0.3334 & 1.667 \\ 0.000 & 1.667 & -1.333 & 0.3333 \end{array} \right].$$

Reducing this system further without pivoting will result in overflows (we need to divide by the 0 in the  $y$  position of the second equation).

- (c) If we solve the same system with pivoting, we get

$$\begin{aligned} \left[ \begin{array}{ccc|c} 6 & 2 & 2 & -2 \\ 2 & \frac{2}{3} & \frac{1}{3} & 1 \\ 1 & 2 & -1 & 0 \end{array} \right] &= \left[ \begin{array}{ccc|c} 6.000 & 2.000 & 2.000 & -2.000 \\ 2.000 & 0.6667 & 0.3333 & 1.000 \\ 1.000 & 2.000 & -1.000 & 0.000 \end{array} \right] \implies \left[ \begin{array}{ccc|c} 6.000 & 2.000 & 2.000 & -2.000 \\ 0.000 & 0.000 & -0.3334 & 1.667 \\ 0.000 & 1.667 & -1.333 & 0.3333 \end{array} \right] \\ &\implies \left[ \begin{array}{ccc|c} 6.000 & 2.000 & 2.000 & -2.000 \\ 0.000 & 0.000 & -0.3334 & 1.667 \\ 0.000 & 1.667 & 0.000 & -6.332 \end{array} \right] \implies \left[ \begin{array}{ccc|c} 6.000 & 0.000 & 0.000 & 15.60 \\ 0.000 & 0.000 & -0.3334 & 1.667 \\ 0.000 & 1.667 & 0.000 & -6.332 \end{array} \right] \\ &\implies \left[ \begin{array}{ccc|c} 1.000 & 0.000 & 0.000 & 2.600 \\ 0.000 & 0.000 & 1 & -5.000 \\ 0.000 & 1.000 & 0.000 & -3.798 \end{array} \right], \end{aligned}$$

which is the solution  $x = 2.6, y = -3.798, z = -5$ .

- (d) Pivoting is much more stable.

2. We have the matrix

$$A = \begin{bmatrix} 12 & 10 & 4 \\ 10 & 8 & -5 \\ 4 & -5 & 3 \end{bmatrix}.$$

We compute

$$\alpha = \sqrt{10^2 + 4^2} = \sqrt{116}$$

and

$$r = \sqrt{\frac{1}{2}(\alpha^2 - 10\alpha)} = \sqrt{58 - 5\sqrt{116}}.$$

We want to zero the last entry of the first column, so we pick

$$\mathbf{u}_1 = \begin{bmatrix} 0 \\ \frac{10 - \sqrt{116}}{2r} \\ \frac{4}{2r} \end{bmatrix}$$

and form the Householder matrix

$$P_1 = I - 2\mathbf{u}_1\mathbf{u}_1^T \approx \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.9285 & 0.3714 \\ 0.3714 & -0.9285 & \end{bmatrix}.$$

Notice that  $P_1$  is symmetric and orthogonal, so  $P_1^{-1} = P_1^T = P_1$ . Thus, we can form the similarity transform

$$\begin{aligned} A_2 &= P_1^{-1}AP_1 \\ &= P_1AP_1 \approx \begin{bmatrix} 12 & 10.7703 & 0 \\ 10.7703 & 3.8621 & 5.3448 \\ 0 & 5.3448 & 7.1379 \end{bmatrix}, \end{aligned}$$

which is tridiagonal.

3. (a) The power method was implemented in code shown at the end of this question. The table below gives the number of iterations required to calculate the dominant eigenvalue to  $10^{-10}$  relative accuracy.

$n$	$\lambda_1$	number of iterations
4	1.5002142800557454	5
8	1.695938996910598	7
12	1.7953720595371143	8
16	1.860036442716032	8
20	1.9071347202891533	8

In general, the algorithm converges quickly, likely because the dominant eigenvalue is well separated from the rest.

- (b) We perform the inverse power method on  $n = 16$  and get

$$\lambda_{16} \approx -6.103265343573278 \times 10^{-19}.$$

- (c) The hilbert matrix  $A$  is symmetric, so it has orthogonal diagonalization

$$A = P^{-1}AP = P^TAP.$$

Thus, the Bauer-Fike theorem reduces to

$$\min_{\lambda \in \sigma(A)} |\lambda - \mu| \leq \|P^{-1}\| \|P\| \|E\| \leq \|E\|.$$

If we take  $E$  to be the error introduced by machine epsilon,  $\|E\| \approx 10^{-16}$ , then we expect the error in our eigenvalue to be bounded by  $\|E\|$ .

- (d) The power method assumes that the matrix being considered is diagonalizable. Consider the matrix

$$A = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix},$$

which has eigenvalue  $\lambda = 2$  with algebraic multiplicity 2.

When we apply our power method to  $A$ , it fails to converge to the eigenvalue. This is expected: the power method relies on some spectral gap between the eigenvalues, which doesn't exist for this example.

```
#!/usr/bin/env python3
import numpy as np
import numpy.linalg as la

# APPM 4600 Homework 12
# Power Method
# Edward Wawrzyniek
```

```

# perform the power method on A
# returns (lambda, eigvector, err, iters)
def power_method(A, relative_tol=1e-15, max_iters=100000):
    # select random b
    b = np.random.random((A.shape[0], 1))
    l = (b.T @ A @ b) / (b.T @ b)
    for n in range(max_iters):
        # apply A
        b1 = A.dot(b)
        # normalize resulting vector
        b1 = b1 / la.norm(b1)
        # calculate the Rayleigh quotient
        l1 = (b1.T @ A @ b1) / (b1.T @ b1)
        # check for convergence
        if (np.abs(l1 - l) / np.abs(l1))[0][0] < relative_tol:
            return (l[0][0], b1, False, n)

        l = l1
        b = b1

    return (l[0][0], b, True, max_iters)

# form a hilbert matrix
def hilbert(n):
    h = np.zeros((n, n))
    for i in range(n):
        for j in range(n):
            h[i][j] = 1/((i+1) + (j+1) - 1)

    return h

def driver():
    print("Question_3(a)")
    for n in range(4, 21, 4):
        A = hilbert(n)
        (l, u, err, iters) = power_method(A, 1e-10)
        print(str(n) + "&" + str(l) + "&" + str(iters) + "_\\\\"")

    print("Question_3(b)")
    Ainv = la.inv(hilbert(16))
    (l, u, err, iters) = power_method(Ainv, 1e-16)
    print(1/l)

    print("Question_3(d)")
    A = np.array([[2, 1], [0, 2]])
    (l, u, err, iters) = power_method(A)
    print(err)

driver()

```