Justin Hexem
AMATH 584

Homework #7

24.1.) a.) <u>Proof:</u> Let λ be an eigenvalue for A and let $\mu \in \mathbb{C}$. Show $\lambda - \mu$ is an eigenvalue of $(A-\mu I)$. We know $Av = \lambda v$ for some $v \in \mathbb{C}^m$, so we see that applying $(A-\mu I)$ to this v gives

$$(A-\mu I)v = Av - \mu I v = \lambda v - \mu v = (\lambda - \mu)v.$$

Thus $(A-\mu I)v = (\lambda-\mu)v$, so $\lambda - \mu$ is an eigenvalue of $(A-\mu I)$. $\boxtimes$

b.) <u>Counterexample:</u> Let $A \in \mathbb{R}^{2\times 2}$ such that $A = I$. We see that

$$\det(A-\lambda I) = 0 \Rightarrow \det(I - \lambda I) = 0 \Rightarrow \det((1-\lambda)I) = 0$$
$$\Rightarrow (1-\lambda)^2 = 0 \Rightarrow \lambda = 1$$

We see that $\lambda = 1$ is an eigenvalue of A, but $-\lambda = -1$ is not.

c.) <u>Proof:</u> Let $A \in \mathbb{R}^{m\times m}$ and let λ be an eigenvalue of A. Show $\bar{\lambda}$ is also an eigenvalue of A. Let us denote the characteristic polynomial of A as p_A. Then p_A is a polynomial with real-valued coefficients, so by the fundamental theorem of algebra p_A can be factored over $\mathbb{R}$ into irreducible linear and quadratic factors. Let us write $\lambda = a + bi$ for $a, b \in \mathbb{R}$. Then either $b = 0$ or $b \neq 0$. If $b = 0$, then $\lambda \in \mathbb{R}$ so $\lambda = \bar{\lambda}$ which means $\bar{\lambda}$ is an eigenvalue of A. If $b \neq 0$, then $\lambda \in \mathbb{C}$ which means λ must be a root of one of the irreducible quadratic factors of p_A. Since the roots of any real-valued quadratic polynomial must always come in complex conjugate pairs, this means $\bar{\lambda}$ is also a root of this quadratic polynomial. This means $\bar{\lambda}$ is a root of p_A, so by theorem 24.1 $\bar{\lambda}$ is also an eigenvalue of A. $\boxtimes$

d.) <u>Proof:</u> Let A be nonsingular and let λ be an eigenvalue of A. Show λ^{-1} is an eigenvalue of A^{-1}. By definition, we know $Av=\lambda v$ for some $v\in\mathbb{C}^m$, so

$$Av=\lambda v \Rightarrow A^{-1}Av=A^{-1}(\lambda v) \Rightarrow Iv=\lambda A^{-1}v$$

$$\Rightarrow v=\lambda A^{-1}v \Rightarrow A^{-1}v=\lambda^{-1}v.$$

Thus $A^{-1}v=\lambda^{-1}v$, so λ^{-1} is an eigenvalue of A^{-1}. $\boxtimes$

e.) <u>Counterexample:</u> Let $A\in\mathbb{R}^{2\times 2}$ be the following:

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

Then A is upper triangular, so the eigenvalues of A are $a_{11}=0$ & $a_{22}=0$. However the a_{12} entry is nonzero, so $A\neq 0$.

f.) <u>Proof:</u> Let A be Hermitian and let λ be an eigenvalue of A. Show $|\lambda|$ is a singular value of A. This follows directly from theorem 5.5. $\boxtimes$

g.) <u>Proof:</u> Let A be diagonalizable and let all eigenvalues of A be equal to λ. Show A is diagonal. Let $A=X^{-1}\Lambda X$ be the diagonalization of A. Then $\Lambda=\lambda I$ since all eigenvalues are equal to λ. This means

$$A=X^{-1}\Lambda X=X^{-1}(\lambda I)X=\lambda X^{-1}IX=\lambda X^{-1}X=\lambda I.$$

Thus $A=\lambda I$, so A is diagonal. $\boxtimes$

24.2.) c.) $|\varepsilon|<1$

$$A=\begin{bmatrix}8&1&0\\1&4&\varepsilon\\0&\varepsilon&1\end{bmatrix} \xRightarrow{\text{By Gerschgorin's theorem}} \begin{aligned}|\lambda_1-8|&\le|1|+|0|=1\\|\lambda_2-4|&\le|1|+|\varepsilon|<1+1=2\\|\lambda_3-1|&\le|0|+|\varepsilon|<1\end{aligned} \Rightarrow \boxed{\begin{aligned}|\lambda_1-8|&\le 1\\|\lambda_2-4|&<2\\|\lambda_3-1|&<1\end{aligned}}$$

d.) By theorem 24.3 A & $X^{-1}AX$ have the same eigenvalues, so

$$X^{-1}AX=\begin{bmatrix}1&0&0\\0&1&0\\0&0&\varepsilon\end{bmatrix}\begin{bmatrix}8&1&0\\1&4&\varepsilon\\0&\varepsilon&1\end{bmatrix}\begin{bmatrix}1&0&0\\0&1&0\\0&0&1/\varepsilon\end{bmatrix}=\begin{bmatrix}8&1&0\\1&4&1\\0&\varepsilon^2&1\end{bmatrix} \xRightarrow{\text{By Gerschgorin's theorem}} |\lambda_3-1|\le|\varepsilon^2|.$$

28.1.) If we apply the unshifted QR algorithm to an orthogonal matrix, we will just end up getting the same matrix back that we started with. This is because if we take the QR decomposition of $A = QR$, then since A and Q are orthogonal R must be as well. However, R is also upper triangular so this implies that R must be diagonal. This means $QR = RQ$, which means $A^{(k)} = A^{(k-1)}$. In order for theorem 28.4 to apply, A must be a real, symmetric matrix with strictly decreasing eigenvalues (in magnitude) in addition to being orthogonal. This implies that $A = A^T$ (symmetric) and $AA^T = A^TA = I$ (orthogonal), so $A^2 = I$ (or $A = A^{-1}$). Also, this implies that A must be diagonal, and since A is real its diagonal entries must be either 1 or -1. This means all eigenvalues of A have magnitude 1, so it is impossible for an orthogonal matrix to satisfy all of the conditions of theorem 28.4, the theorem that guarantees convergence of the pure QR algorithm.

4.) Proof: Let $A \in \mathbb{R}^{m \times m}$ such that A is diagonalizable as $A = X\Lambda X^{-1}$. Show that $e^{tA} = Xe^{t\Lambda}X^{-1}$. We see that

$$e^{tA} = \sum_{k=0}^{\infty} \frac{(tA)^k}{k!} = \sum_{k=0}^{\infty} \frac{t^k A^k}{k!} = \sum_{k=0}^{\infty} \frac{t^k (X\Lambda X^{-1})^k}{k!} = \sum_{k=0}^{\infty} \frac{t^k X \Lambda^k X^{-1}}{k!} = \sum_{k=0}^{\infty} \frac{X t^k \Lambda^k X^{-1}}{k!}$$

$$= \sum_{k=0}^{\infty} X \frac{(t\Lambda)^k}{k!} X^{-1} = X\left[\sum_{k=0}^{\infty} \frac{(t\Lambda)^k}{k!}\right] X^{-1} = Xe^{t\Lambda}X^{-1}. \boxed{X}$$

5.) a.) Gerschgorin's theorem tells us that all eigenvalues of this matrix are contained in $|\lambda - 2| \le 2$ since for all $k \in \{2, 3, \ldots, m-1\}$

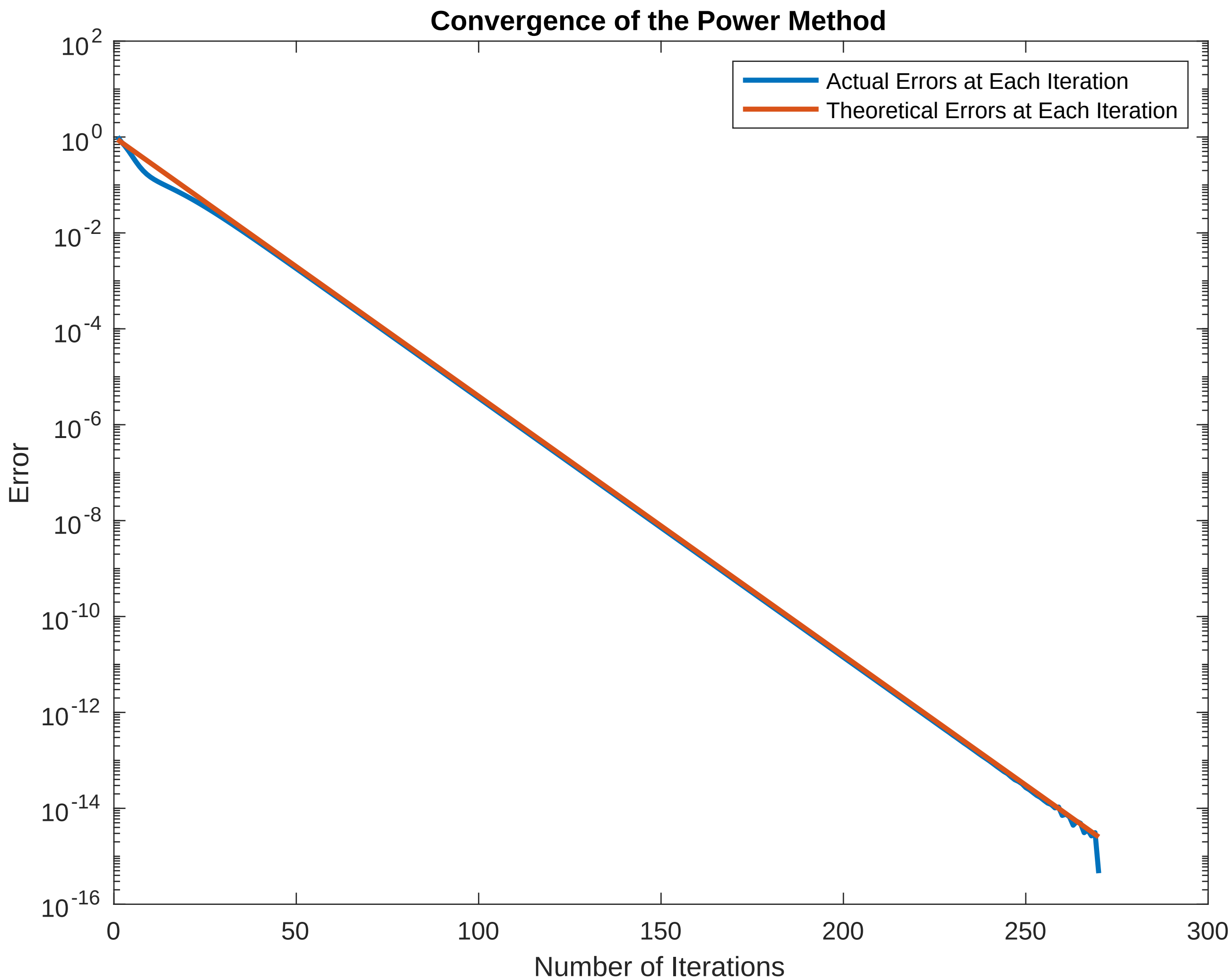
$$|\lambda_k - 2| \le |-1| + |-1| = 1 + 1 = 2$$

and for $k=1$ and $k=m$ we have

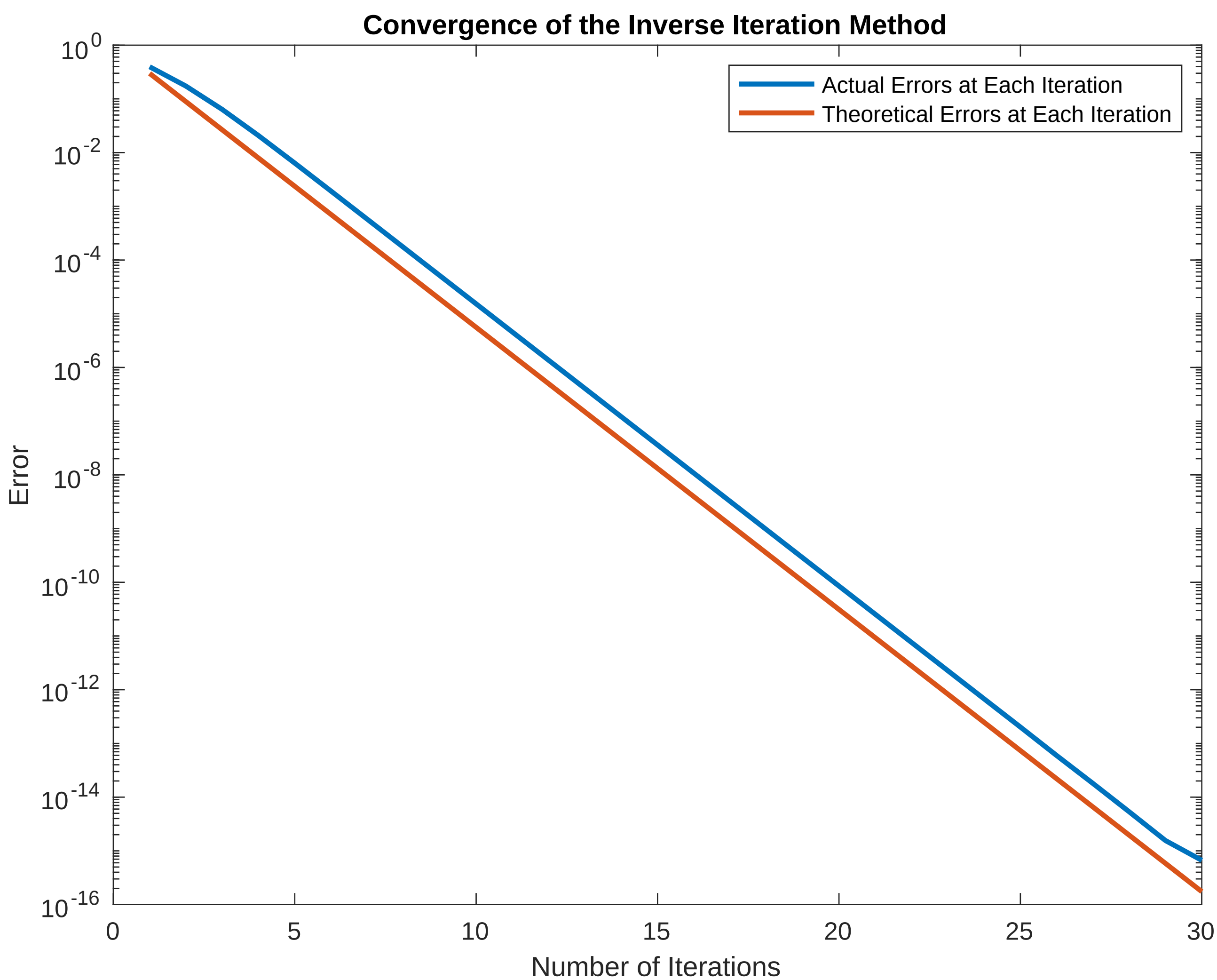
$$|\lambda_k - 2| \le |-1| = 1.$$

b.) The plot seems to closely match the theoretical convergence rate quite well for the power method.

c.) The rate of convergence with $s=1$ as the shift is much faster than the power method from part b. The power method took about 220 iterations to converge and the shifted inverse iteration method took about 30 iterations to converge.



Convergence of the Power Method
Actual Errors at Each Iteration
Theoretical Errors at Each Iteration
Error
Number of Iterations
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
10^{-16}
0
50
100
150
200
250
300



Convergence of the Inverse Iteration Method
Actual Errors at Each Iteration
Theoretical Errors at Each Iteration
Error
Number of Iterations
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
10^{-12}
10^{-14}
10^{-16}
0
5
10
15
20
25
30

```
A = diag(2*ones(1,10)) + diag((-1)*ones(1,9),1) + diag((-1)*ones(1,9),-1);
I = eye(10);

v = rand(10, 1);
v = v / norm(v);
shiftv = v;

lambda = 1;
shiftLambda = 1;

tol = 10e-16;

trueEigs = eig(A);
maxEig = trueEigs(end);
secondEig = trueEigs(end-1);
shiftMaxEig = trueEigs(4);
shiftSecondEig = trueEigs(3);

convFactor = abs(secondEig / maxEig);
counter = 0;
shiftConvFactor = abs((1 - shiftMaxEig) / (1 - shiftSecondEig));
shiftCounter = 0;

errors = [];
shiftErrors = [];
convFactors = [];
shiftConvFactors = [];

while (abs(maxEig - lambda) > tol)
    v = A * v;
    v = v / norm(v);

    lambda = v.' * A * v;

    errors(end+1) = abs(maxEig - lambda);

    counter = counter + 1;
    convFactors(end+1) = convFactor^(2 * counter);
end
```

```
while (abs(shiftMaxEig - shiftLambda) > tol)
    shiftv = (A - I) \ shiftv;
    shiftv = shiftv / norm(shiftv);

    shiftLambda = shiftv.' * A * shiftv;

    shiftErrors(end+1) = abs(shiftMaxEig - shiftLambda);

    shiftCounter = shiftCounter + 1;
    shiftConvFactors(end+1) = shiftConvFactor^(2 * shiftCounter);
end

eigenVec = v
eigenVal = lambda

shiftEigenVec = shiftv
shiftEigenVal = shiftLambda

numErrors = length(errors);
xvals = 1:numErrors;

shiftNumErrors = length(shiftErrors);
shiftxvals = 1:shiftNumErrors;
```

```
%%
semilogy(xvals, errors, "LineWidth", 2)
hold on
semilogy(xvals, convFactors, "Linewidth", 2)
hold off
title("Convergence of the Power Method")
legend("Actual Errors at Each Iteration", "Theoretical Errors at Each Iteration")
xlabel("Number of Iterations")
ylabel("Error")
```

```
%%
semilogy(shiftxvals, shiftErrors, "LineWidth", 2)
hold on
semilogy(shiftxvals, shiftConvFactors, "Linewidth", 2)
hold off
title("Convergence of the Inverse Iteration Method")
legend("Actual Errors at Each Iteration", "Theoretical Errors at Each Iteration")
xlabel("Number of Iterations")
ylabel("Error")
```

```
eigenVec =

   -0.1201
    0.2305
   -0.3223
    0.3879
   -0.4221
    0.4221
   -0.3879
    0.3223
   -0.2305
    0.1201


eigenVal =

    3.9190


shiftEigenVec =

    0.3879
    0.3223
   -0.1201
   -0.4221
   -0.2305
    0.2305
    0.4221
    0.1201
   -0.3223
   -0.3879


shiftEigenVal =

    1.1692
```