Homework 3

Nima Leclerc
Cornell University
CS 4220
Numerical Analysis: Linear and Nonlinear Problems

March 30, 2018

Question 1:Assume that we are given A

(a) Here one seeks to show that given an a matrix $A \in R^{n\times n}$ with an initial guess of an eigenvector $v^{(0}$ such that $v^{(0)T}v_1 \neq 0$ and $v^{(0)T}v_2 \neq 0$ simultaneously, it can be shown that $v^(0) \in span\{v_1, v_2\}$.
Here, one can define $v^{(0)} = \sum_{i=1}^{n} \alpha_i v_i$ and the kth iterate of v as

$$v^{(k)} = c_k A^k \sum_{i=1}^{n} \alpha_i v_i = c_k \lambda_1^k \sum_{i=1}^{n} \alpha_i (\frac{\lambda_i}{\lambda_1})^k v_i =$$

The above can now be written as,

$$v^{(k)} = c_k \lambda_1^k (\alpha_1 (\frac{\lambda_1}{\lambda_1})^k v_1 + \alpha_2 (\frac{\lambda_2}{\lambda_1})^k v_2 + \sum_{i=3}^{n} \alpha_i (\frac{\lambda_i}{\lambda_1})^k v_i)$$

$$= c_k \lambda_1^k (\alpha_1 (\frac{\lambda_1}{\lambda_1})^k v_1 + \alpha_2 (\frac{\lambda_1}{\lambda_1})^k v_2 + \sum_{i=3}^{n} \alpha_i (\frac{\lambda_i}{\lambda_1})^k v_i) = c_k \lambda_1^k (\alpha_1 (1)^k v_1 + \alpha_2 (1)^k v_2 + \sum_{i=3}^{n} \alpha_i (\frac{\lambda_i}{\lambda_1})^k v_i) =$$

From the nature of this method, it is known that the higher order terms in this expansion ($i > 3$) approach 0, since we have λ_1 as the dominant eigenvalue in the denominator. Hence the above expression reduces to the following,

$$v^{(k)} = c_k \lambda_1^k (\alpha_1 v_1 + \alpha_2 v_2) = \gamma_{1,k} (\alpha_1 v_1 + \alpha_2 v_2) = \beta_1 v_1 + \beta_2 v_2$$

Hence, the above explicitly indicates that the iterate $v^{(k)}$ can be written as a linear combination v_1 and v_2 which suggests that $v^{(k)}$ lies in the span of v_1 and v_2 .

(b)

In proving this convergence, one can simplify as follows

$$||[v^{(k)}]^T [v_1 v_2]||^2 = \frac{c_1^2 \lambda_1^{2k} + c_2^2 \lambda_2^{2k}}{\sum_{i=1}^{n} c_i^2 \lambda_i^{2k}} = \frac{\lambda_1^{2k}(c_1^2 + c_2^2)}{\lambda_1^{2k}(c_1^2 + c_2^2 + O(|\frac{\lambda_3}{\lambda_2}|^{2k}}) = \frac{\beta}{\beta + O(|\frac{\lambda_3}{\lambda_1})|^{2k}} \sim \frac{1}{1 + O(|\frac{\lambda_3}{\lambda_1})|^{2k}} = 1 - O(|\frac{\lambda_3}{\lambda_1})|^{2k}$$

Hence,

$$||1 - [v^{(k)}]^T [v_1 v_2]|| = O(|\frac{\lambda_3}{\lambda_1})|^{2k}$$

This is the rate of convergence. (c) Here, one considers the case finding the eigenvalues of A using Rayleigh Quotient iteration. Considering the case where λ_1 and λ_2 are unique one can achieve the following (considering the two dominant eigenvectors).

$$\lambda_k = \frac{v^{(k)T} A v^{(k)}}{v^{(k)T} v^{(k)}} = \frac{(\beta_1 v_1 + \beta_2 v_2)^T A (\beta_1 v_1 + \beta_2 v_2)}{(\beta_1 v_1 + \beta_2 v_2)^T (\beta_1 v_1 + \beta_2 v_2)}$$

$$\lambda_k = \frac{v^{(k)T} A v^{(k)}}{v^{(k)T} v^{(k)}} = \frac{(\beta_1 v_1 + \beta_2 v_2)^T (\beta_1 A v_1 + \beta_2 A v_2)}{(\beta_1 v_1 + \beta_2 v_2)^T (\beta_1 v_1 + \beta_2 v_2)} = \frac{(\beta_1 v_1 + \beta_2 v_2)^T (\beta_1 \lambda_1 v_1 + \beta_2 \lambda_2 v_2)}{(\beta_1 v_1 + \beta_2 v_2)^T (\beta_1 v_1 + \beta_2 v_2)}$$

$$= \frac{\beta_1^2 \lambda_1 v_1^T v_1 + \beta_2^2 \lambda_2 v_2^T v_2 + \beta_1 \beta_2 \lambda_1 v_2^T v_1 v_1 + \beta_1 \beta_2 \lambda_2 v_1^T v_2}{\beta_1^2 v_1^T v_1 + \beta_2^2 v_2^T v_2 + \beta_1 \beta_2 v_2^T v_1 + \beta_1 \beta_2 v_1^T v_2}$$

Since v_1 and v_2 are orthonormal to one another, the above reduces to the following

$$= \frac{\beta_1^2 \lambda_1 + \beta_2^2 \lambda_2}{\beta_1^2 + \beta_2^2}$$

It is clear that convergence is not achieved in this case, when λ_1 and λ_2 differ. Hence, for this general case convergence cannot be achieved via Rayleigh iteration. However, one can take the case where $\lambda_1 = \lambda_2$. In this case, the above expression reduces to the following

$$\frac{\beta_1^2 \lambda_1 + \beta_2^2 \lambda_1}{\beta_1^2 + \beta_2^2} = \frac{\lambda_1(\beta_1^2 + \beta_2^2)}{\beta_1^2 + \beta_2^2} = \lambda_1$$

This suggests that convergence is possible in when v_1 and v_2 share the same dominant eigenvalue.

2.)

Here, one considers applying the QR algorithm to seek the eigenvalues of the following matrix,

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

The QR algorithm was implemented and the code is shown below,

```
  %%QR_EIG, QR Eigenvalue Solver
%%In: n x n matrix A
%%Out: n x n matrix A1, with eigenvalues along the matrix diagonal
function [A1]=QR_EIG(A)
maxiter=1000; %%Maximum number of iterations
for i=1:maxiter
    [q,r]=qr(A);
    A=r*q;
end
  A1= A;
   end
```

The output of the code applied to matrix A yields the following matrix,

$$A1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

This upper Hessenberg matrix applied with the QR algorithm retrieves the same matrix with 0 along the diagonal. This suggests that this method fails to find the eigenvalues of this type of matrix. Hence, one must seek a different algorithm. Using a shift of $A^{(k)}_{n,n}$ cannot be applied here to yield better results since such a shift is a 0 shift (diagonal elements are 0). The algorithm used for this is implemented below,

```
%%QR_EIG, SHIFTED QR Eigenvalue Solver
%%In: n x n matrix A, shift mu
%%Out: n x n matrix A1, with eigenvalues along the matrix diagonal
function [A1]=QRSHIFT_EIG(A0,mu)
maxiter=1000; %%Maximum number of iterations
n=size(A0);
for i=1:maxiter
    [q,r]=qr(A0-mu*eye(n));
    A0=r*q+mu*eye(n);
end
  A1= A0;
   end
```

3.)

Here, one hopes to show the similarity of the iterates $A^{(k)}$ to its successive iterate in the shifted QR Algorithm. This is achieved in the following way. One defines the $A^{(k)}$ iterate in QR as follows, assuming a shift of $\mu^{(k)}$,

$$A^{(k)} - \mu^{(k)} I = Q^{(k)} R^{(k)}$$

Similarly for the $k+1$th iterate,

$$R^{(k)} Q^{(k)} = A^{(k+1)} - \mu^{(k)} I$$

Hence,

$$R^{(k)} = (A^{(k+1)} - \mu^{(k)} I) Q^{(k)T}$$

Then substituting this into the previous expression,

$$A^{(k)} - \mu^{(k)} I = Q^{(k)}(A^{(k+1)} - \mu^{(k)} I) Q^{(k)T}$$

$$= Q^{(k)} A^{(k+1)} Q^{(k)T} - Q^{(k)} \mu^{(k)} I Q^{(k)T} = Q^{(k)} A^{(k+1)} Q^{(k)T} - \mu^{(k)} I$$

This results in the following,

$$A^{(k)} - \mu^{(k)} I = Q^{(k)} A^{(k+1)} Q^{(k)T} - \mu^{(k)} I$$

$$A^{(k)} = Q^{(k)} A^{(k+1)} Q^{(k)T} = Q^{(k)} A^{(k+1)} Q^{(k)-1} =$$

The above expression defines the similarity of $A^{(k+1)}$ to $A^{(k)}$ Q.E.D.

4.)
The codes for the QR, Rayleigh Iteration, and Shifted Inverse eigenvalue solvers are as follows

```
  %%QR_EIG, QR Eigenvalue Solver
%%In: n x n matrix A
%%Out: n x n matrix A1, with eigenvalues along the matrix diagonal
function [A1]=QR_EIG(A)
maxiter=1000; %%Maximum number of iterations
for i=1:maxiter
    [q,r]=qr(A);
    A=r*q;
end
  A1= A;
   end
```

```
%%RAYLEIGH_EIG, Eigenvalue Solver
%%In: n x n matrix A, initial eigenvector guess v0
%%Out: Dominant eigenvalue of A, lam
function [lam]=RAYLEIGH_EIG(A,v0)
maxiter=1000; %%Maximum number of iterations
for i=1:maxiter
    n=size(A);
    lam=(v0')*A*v0;
    B=A-lam*eye(n);
    v=B\v0;
    v0=v/norm(v);
    lam= (v0')*A*v0;
end
   end
```

```
%%SHIFTINV_EIG, QR Eigenvalue Solver
%%In: n x n matrix A, shift alph,initial eigenvector guess v0
%%Out: Dominant eigenvalue of A, lam
function [lam]=SHIFTINV_EIG(A,v0,alph)
maxiter=1000; %%Maximum number of iterations
for i=1:maxiter
    n=size(A);
    B=A-alph*eye(n);
    v=B\v0;
    v0=v/norm(v);
    lam= (v0')*A*v0;
end
   end
```

(a) The QR algorithm above was implemented for a random 500x500 matrix, A such that $A = V^T \Lambda V$ for some random 500 x500 matrix V and random 500x500 diagonal matrix Λ. The distribution obtained for the obtained eigenvalues is attached. The plot suggests a relatively uniform distribution.

(b) The Rayleigh Iteration algorithm was implemented from above. The 500 random eigenvectors were generated from the columns of Q in $A = QR$. The distribution is attached. The histogram suggests that the eigenvalues obtained for various random eigenvectors range between -0.5 and 0.5. This behavior could possibly be explained by the fact that Rayleigh Iteration yields eigenvalues that are bounded by the magnitude of the largest eigenvalue of A. This is observed here clearly, as eigenvalues are bounded between [-0.5 0.5].

(c) Now, one can perform the same with shifted iteration, using a shift of $\mu = 1$. This yields a uniform distribution about 1 as depicted in the attached plot. It seems that the shift value is representative of the dominant eigenvalue of the matrix A. Hence, one must be judiciouss and careful in choosing a shift for a particular application.

5.)

One seeks to show that for splitting with $A = M - N$ to solve a linear system using the following iteration,

$$Mx^{(k+1)} = Nx^{(k)} + b$$

that the spectral radius, $\rho(M^{-1}N) < 1$. This is done as follows,

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b$$

Then,

$$e^{(k+1)} = M^{-1}Ne^{(k)} = (M^{-1}N)...(M^{-1}N)e^{(0)} = (M^{-1}N)^k e^{(0)} = R^k e^{(0)}$$

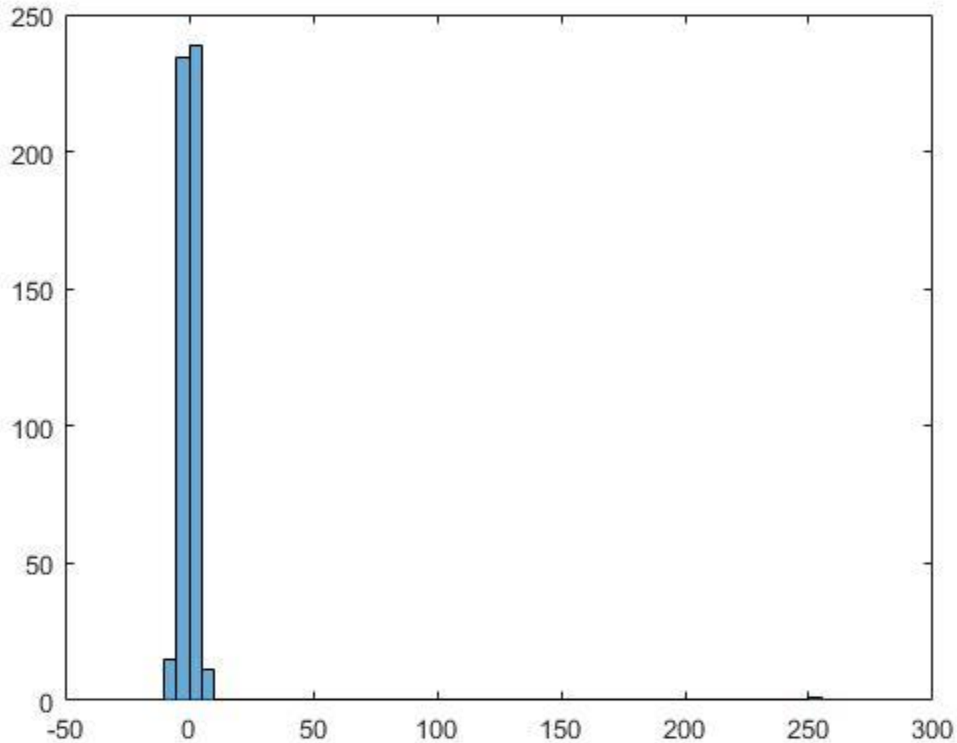
This now sows that as $e^{(k)}$ goes to zero, k goes to infinity if R^k goes to 0. Now one observes that $||R|| < 1$. It can now be shown that the same condition is true for the spectral radius of R. Here, let

$$e^{(0)} = \sum_{i=1}^{n} \beta_i v_i$$

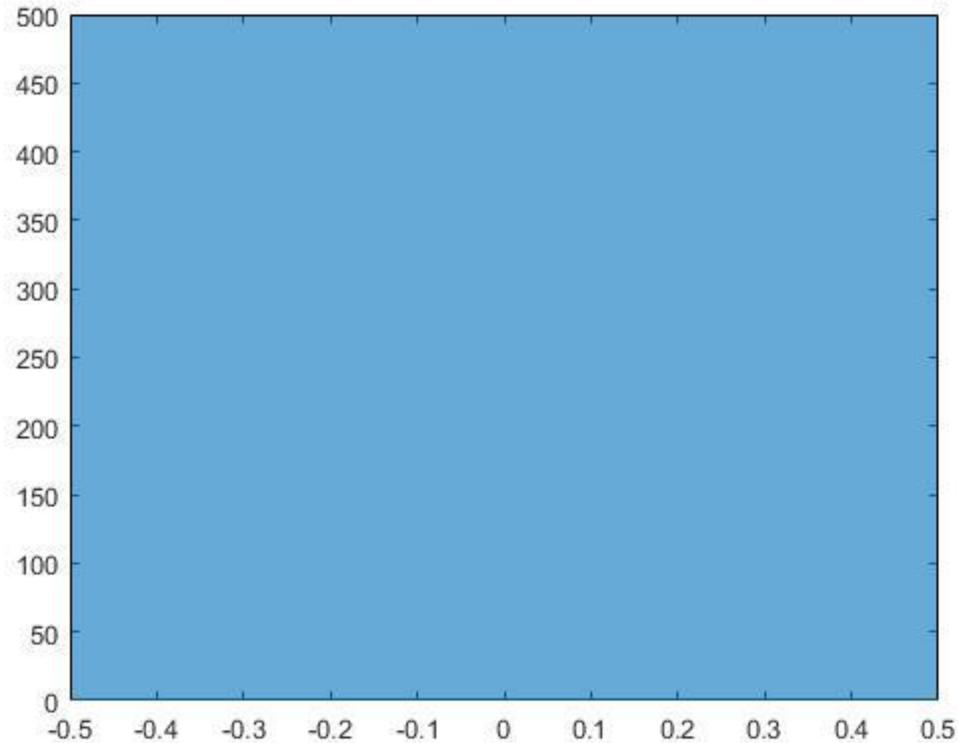
for coefficients β_i and eigenvectors of R, v_i. Now applying R^k to this,

$$R^k e^{(0)} = \sum_{i=1}^{n} \beta_i v_i$$

It is clear that the above expression approaches zero as well. Hence, one must not only put the contraint on $||R||$ but also on the spectral radius of R. Hence, it is enforced that $\rho(R) < 1$.



(a) QR algorithm eigenvalue distribution.



(b) Rayleigh Quotient iteration eigenvalue solver performed with 500 random eigenvectors for a 500x500 eigenvalue.