Mikhail Gaerlan
23 January 2018
MAT 226B Freund

Problem 1

Let $0 \le \alpha \le 1$ and $A_\alpha = [a_{jk}^\alpha]$. Then

$$ee^T = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}.$$

Thus, $a_{jk}^\alpha = \alpha a_{jk} + \dfrac{1}{n}(1-\alpha]$. Also, $\alpha_{jk}^\alpha \ge 0$ since $0 \le \alpha \le 1$, and so

$$\sum_{k=1}^n a_{jk}^\alpha = \sum_{k=1}^n \left(\alpha a_{jk} + \frac{1}{n}(1-\alpha)\right) = n\frac{1}{n}(1-\alpha) + \alpha \sum_{k=1}^n a_{jk} = 1 - \alpha + \alpha = 1.$$

Therefore, A_α is row-stochastic.

Problem 2

The PageRank vector x is given by the formula $A^T x = x$ where A is given by the following matrix:

$$A = \begin{bmatrix}
0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 & 1/3 & 0 \\
1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\
1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 \\
1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/10 \\
0 & 0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 0 & 1/4 & 0 \\
0 & 1/5 & 0 & 1/5 & 0 & 0 & 1/5 & 1/5 & 0 & 1/5 \\
1/5 & 0 & 1/5 & 1/5 & 0 & 0 & 1/5 & 0 & 1/5 & 0
\end{bmatrix}.$$

The following MATLAB code was used to solve for x and rank the eigenvalues.

```
format long e;
load matrices.mat A;
[V,D]=eig(A');l = diag(D);
[absl,c]=sort(abs(l));
x = abs(V(:,c(end):c(end)));
[~,b]=sort(x,'descend');
matrix2latex(x,'x.tex','format','%-.15e')
matrix2latex(b','sort.tex')
```

$$x = \begin{bmatrix} \texttt{1.611685764370067e-01} \\ \texttt{1.785251923609917e-01} \\ \texttt{2.057998745272545e-01} \\ \texttt{2.717395180598953e-01} \\ \texttt{1.931698370468546e-01} \\ \texttt{1.720939462820155e-01} \\ \texttt{7.479616674742428e-01} \\ \texttt{1.785251923609919e-01} \\ \texttt{3.296982176632033e-01} \\ \texttt{2.429151380432770e-01} \end{bmatrix}$$

Thus the ranking of the websites from most to least important is given by the row vector:

$$b = \begin{bmatrix} 7 & 9 & 4 & 10 & 3 & 5 & 8 & 2 & 6 & 1 \end{bmatrix}.$$

Problem 3

(a)

Let $x \in \mathbb{R}^n$. Then,

$$\begin{aligned} \left\|A^T x\right\|_1 &= \sum_{k=1}^n \left|\sum_{j=1}^n a_{jk} x_j\right| \\ &\leq \sum_{k=1}^n \sum_{j=1}^n |a_{jk}|\,|x_j| \\ &= \sum_{j=1}^n |x_j| \sum_{k=1}^n |a_{jk}| \\ &= \sum_{j=1}^n |x_j| \\ &= \|x\|_1 \end{aligned}$$

Since, $\left\|A^T x\right\|_1 \leq \|x\|_1\ \forall\, x \in \mathbb{R}^n$, then if λ is an eigenvalue of A^T, then $|\lambda| \leq 1$. Since A and A^T have the same eigenvalues, then λ is an eigenvalue of A and $|\lambda| \leq 1$.

(b)

Since $a_{jk}, x_j^{(0)} \geq 0$, then $x^{(i)} \geq 0$ by closure of the nonnegative real numbers under addition and multiplication. Also, since $x^{(0)} \geq 0$, then the inequality from 3.(a) becomes an equality, so $\sum_{j=1}^n x_j^{(i)} = \sum_{j=1}^n x_j^{(0)}\ \forall\, i \geq 0$.

(c)

By induction, $x^{(i)} = X\Lambda^i X^{-1} x^{(0)}$ where Λ, Λ^i are given by the following matrices:

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{bmatrix} \text{ and } \Lambda^i = \begin{bmatrix} \lambda_1^i & 0 & \cdots & 0 \\ 0 & \lambda_2^i & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n^i \end{bmatrix}$$

Therefore,

$$\begin{aligned} \lim_{i\to\infty} x^{(i)} &= \lim_{i\to\infty} X\Lambda^i X^{-1} x^{(0)} \\ &= X\left(\lim_{i\to\infty} \Lambda^i\right) X^{-1} x^{(0)} \end{aligned}$$

Since $|\lambda_j| < 1 \ \forall 2 \leq j \leq n$, then $\lim_{i\to\infty} \lambda_j^i = \begin{cases} 1 & \text{if } j = 1 \\ 0 & \text{if } j > 1 \end{cases}$. Thus $\lim_{i\to\infty} \Lambda^i$ converges. Since $x^i \geq 0 \ \forall i$, then $x \geq 0$ by the properties of limits. Also, since $\sum_{j=1}^{n} x_j^{(i)} = \sum_{j=1}^{n} x_j^{(0)} \ \forall i \geq 0$, then $x^{(0)} \neq 0 \implies x \neq 0$.

Problem 4

(a)

For the case $n = 10$ from Problem 2, the code

```
load 'matrices.mat' A
n = 10;
Jv = [3;6;7];
Q = A;
for j = Jv'
    Q(j,:) = zeros(1,n);
end
matrix2latex(graphx(n,Q,Jv,ee),'graphy.tex','format','%-.15e')
matrix2latex(graphax(n,Q,Jv,0.5,ee),'graphy05.tex','format','%-.15e')
matrix2latex(graphax(n,Q,Jv,0.85,ee),'graphy085.tex','format','%-.15e')
```

where the functions **graphx** and **graphax** are defined as the following

```
function y = graphx(n,Q,Jv,x)
%Multiply the matrix representation of the connectivity of a directed graph.
%   input   - n     size of vector
%           - Q     initial connections
%           - Jv    indices of vectors with dj = 0
%           - x     column vector
%   output - A'*x   where A = Q + v*e'/n
A = Q;
y = A'*x + sum(x(Jv'))/n;
end
```

```
function y = graphax(n,Q,Jv,a,x)
%Multiply the matrix representation of the connectivity of a directed graph.
%   input   - n     size of vector
%           - Q     initial connections
%           - Jv    indices of vectors with dj = 0
%           - a     parameter between 0 and 1
%           - x     column vector
%   output - A'*x   where A = Q + v*e'/n
A = Q;
y = a*A'*x + a*sum(x(Jv'))/n + (1-a)*sum(x)/n;
end
```

has the following output

$$y = \begin{bmatrix} 5.000000000000000\text{e-}01 \\ 5.000000000000000\text{e-}01 \\ 7.500000000000000\text{e-}01 \\ 9.500000000000000\text{e-}01 \\ 8.000000000000000\text{e-}01 \\ 6.333333333333333\text{e-}01 \\ 3.116666666666666\text{e+}00 \\ 5.000000000000000\text{e-}01 \\ 1.416666666666667\text{e+}00 \\ 8.333333333333333\text{e-}01 \end{bmatrix}$$

$$y_{0.5} = \begin{bmatrix} 7.500000000000000\text{e-}01 \\ 7.500000000000000\text{e-}01 \\ 8.750000000000000\text{e-}01 \\ 9.750000000000000\text{e-}01 \\ 9.000000000000000\text{e-}01 \\ 8.166666666666667\text{e-}01 \\ 2.058333333333334\text{e+}00 \\ 7.500000000000000\text{e-}01 \\ 1.208333333333333\text{e+}00 \\ 9.166666666666666\text{e-}01 \end{bmatrix} \qquad y_{0.85} = \begin{bmatrix} 5.750000000000001\text{e-}01 \\ 5.750000000000001\text{e-}01 \\ 7.875000000000000\text{e-}01 \\ 9.575000000000000\text{e-}01 \\ 8.300000000000000\text{e-}01 \\ 6.883333333333334\text{e-}01 \\ 2.799166666666666\text{e+}00 \\ 5.750000000000001\text{e-}01 \\ 1.354166666666667\text{e+}00 \\ 8.583333333333334\text{e-}01 \end{bmatrix}$$

For the case $n = 685230$, the following code

```
load('www0.mat');load('x0.mat');
n = 685230;
y = graphx(n,Q,Jv,x0);
y85 = graphax(n,Q,Jv,0.85,x0);
matrix2latex([y(2);y(222222);y(300000);y(400000)],'www0graphy.tex','
    alignment','r','format','%-.15e')
matrix2latex([y85(2);y85(222222);y85(300000);y85(400000)],'www0graphy85.
    tex','alignment','r','format','%-.15e')
```

resulted in the output

$$\begin{bmatrix} y_2 \\ y_{222222} \\ y_{300000} \\ y_{400000} \end{bmatrix} = \begin{bmatrix} \texttt{-4.833020050815520e-02} \\ \texttt{1.715159442482511e-01} \\ \texttt{9.711221038679012e-03} \\ \texttt{-7.835267507349471e-02} \end{bmatrix}, \quad \begin{bmatrix} y_{0.85,2} \\ y_{0.85,222222} \\ y_{0.85,300000} \\ y_{0.85,400000} \end{bmatrix} = \begin{bmatrix} \texttt{-4.108411686447251e-02} \\ \texttt{1.457851061784729e-01} \\ \texttt{8.251091450336577e-03} \\ \texttt{-6.660322024501109e-02} \end{bmatrix}$$

(b)

```
function [lambda,j,relresid,x] = powerm(n,Q,Jv,x0,eps,kmax)
%Find the dominant eigenvalue of a matrix A'
%   input   - A          n x n matrix
%           - x0         starting vector
%           - eps        tolerance
%           - kmax       maximum iterations
%   output - lambda     dominant eigenvalue of A'
%           - j          iteration reached
%           - relresid   relative residual
x = x0;
[~,I]=max(abs(x));lambda = x0(I);
for k = 1:kmax
    j = k; oldlambda = lambda;
    x = graphx(n,Q,Jv,x);
    [~,I]=max(abs(x));
    lambda=x(I); x = x/lambda;
    relresid=abs(lambda-oldlambda)/abs(oldlambda);
    if (relresid <= eps)
        break;
    end
end
end
```

(c)

The following code was used to compute the dominant eigenvalue λ, the iteration count k, the *relative eigenvalue residual* ϵ, and the corresponding eigenvector x:

```
load matrices.mat A;eps=1e-15;kmax=1000;n = 10;
Jv = [3;6;7];Q = A;
for j = Jv'
    Q(j,:) = zeros(1,n);
end

x10 = ones(10,1);
[lambda,k,releps,x]=powerm(n,Q,Jv,x10,eps,kmax);
matrix2latex([lambda,k,releps]','powerm.tex','alignment','r','format','%-.15e')
matrix2latex(x,'powermx.tex','alignment','r','format','%-.15e')
[~,b]=sort(x,'descend');
matrix2latex(b','sortp.tex')

x20 = [1;-1;1;-1;1;-1;1;-1;1;-1];
[lambda,k,releps,x]=powerm(n,Q,Jv,x20,eps,kmax);
matrix2latex([lambda,k,releps]','powerm2.tex','alignment','r','format','%-.15e')
matrix2latex(x,'powerm2x.tex','alignment','r','format','%-.15e')
[~,b]=sort(x,'descend');
matrix2latex(b','sortp2.tex')
```

with the output given by

$$\begin{bmatrix} \lambda_1 \\ k_1 \\ \epsilon_1 \end{bmatrix} = \begin{bmatrix} 9.999999999999998e{-}01 \\ 4.600000000000000e{+}01 \\ 8.881784197001246e{-}16 \end{bmatrix} \qquad \begin{bmatrix} \lambda_2 \\ k_2 \\ \epsilon_2 \end{bmatrix} = \begin{bmatrix} 9.999999999999998e{-}01 \\ 1.010000000000000e{+}02 \\ 6.661338147750936e{-}16 \end{bmatrix}$$

$$x = \begin{bmatrix} 2.154770537656687e{-}01 \\ 2.386822749404331e{-}01 \\ 2.751476225007769e{-}01 \\ 3.633067440174039e{-}01 \\ 2.582616803066404e{-}01 \\ 2.300839117372837e{-}01 \\ 1.000000000000000e{+}00 \\ 2.386822749404331e{-}01 \\ 4.407956075831346e{-}01 \\ 3.247695017093132e{-}01 \end{bmatrix} \qquad x = \begin{bmatrix} 2.154770537656687e{-}01 \\ 2.386822749404331e{-}01 \\ 2.751476225007770e{-}01 \\ 3.633067440174039e{-}01 \\ 2.582616803066405e{-}01 \\ 2.300839117372837e{-}01 \\ 1.000000000000000e{+}00 \\ 2.386822749404331e{-}01 \\ 4.407956075831347e{-}01 \\ 3.247695017093132e{-}01 \end{bmatrix}.$$

Thus the ranking of the websites from most to least important is given by the row vectors:

$$\begin{aligned} b_1 &= \begin{bmatrix} 7 & 9 & 4 & 10 & 3 & 5 & 2 & 8 & 6 & 1 \end{bmatrix}. \\ b_2 &= \begin{bmatrix} 7 & 9 & 4 & 10 & 3 & 5 & 2 & 8 & 6 & 1 \end{bmatrix}. \end{aligned}$$

(d)

The following code was used

```
load('www0.mat');
eps=1e-12;kmax=10000;
n=685230;x10 = ones(n,1);

[lambda,k,releps,~]=powerm(n,Q,Jv,x10,eps,kmax);
matrix2latex([lambda,k,releps]','www0power.tex','alignment','r','format','
    %-.15e')

[lambda,k,releps,~]=powerm(n,Q,Jv,x0,eps,kmax);
matrix2latex([lambda,k,releps]','www0power0.tex','alignment','r','format',
    '%-.15e')

a = 0.85;
[lambda,k,releps,x]=poweram(n,Q,Jv,a,x10,eps,kmax);
matrix2latex([lambda,k,releps]','www0powera.tex','alignment','r','format',
    '%-.15e')
[~,b]=sort(x,'descend');
matrix2latex(b(1:10)','www0sorta.tex')

[lambda,k,releps,x]=poweram(n,Q,Jv,a,x0,eps,kmax);
matrix2latex([lambda,k,releps]','www0powera0.tex','alignment','r','format'
    ,'%-.15e')
[~,b]=sort(x,'descend');
matrix2latex(b(1:10)','www0sorta0.tex')
```

where the function `poweram` is given by

```
function [lambda,j,relresid,x] = poweram(n,Q,Jv,a,x0,eps,kmax)
%Find the dominant eigenvalue of a matrix A_a'
%   input   - A         n x n matrix
%           - x0        starting vector
%           - eps       tolerance
%           - kmax      maximum iterations
%   output  - lambda    dominant eigenvalue of A_a'
%           - j         iteration reached
%           - relresid  relative residual
x = x0;
[~,I]=max(abs(x));lambda = x0(I);
for k = 1:kmax
    j = k; oldlambda = lambda;
    x = graphax(n,Q,Jv,a,x);
    [~,I]=max(abs(x));
    lambda=x(I); x = x/lambda;
    relresid=abs(lambda-oldlambda)/abs(oldlambda);
    if (relresid <= eps)
        break;
    end
end
end
```

For $x_1 = e$ and $x_2 = x_0$, the results are

$$\begin{bmatrix} \lambda_1 \\ k_1 \\ \varepsilon_1 \end{bmatrix} = \begin{bmatrix} \texttt{1.000000895682723e+00} \\ \texttt{1.000000000000000e+04} \\ \texttt{1.097372320320294e-10} \end{bmatrix} \qquad \begin{bmatrix} \lambda_2 \\ k_2 \\ \epsilon_2 \end{bmatrix} = \begin{bmatrix} \texttt{8.896461299962105e-01} \\ \texttt{1.000000000000000e+04} \\ \texttt{2.085300260832659e-01} \end{bmatrix}$$

The convergence of the power method for this case is much slower than the $n = 10$ case. The algorithm reached the maximum iterations instead of reaching the residual epsilon. For $A_{0.85}^T$, the output is given by

$$\begin{bmatrix} \lambda_1 \\ k_1 \\ \varepsilon_1 \end{bmatrix} = \begin{bmatrix} \texttt{9.999999999951747e-01} \\ \texttt{1.220000000000000e+02} \\ \texttt{9.277023593821179e-13} \end{bmatrix} \qquad \begin{bmatrix} \lambda_2 \\ k_2 \\ \epsilon_2 \end{bmatrix} = \begin{bmatrix} \texttt{1.000000000004541e+00} \\ \texttt{1.630000000000000e+02} \\ \texttt{9.168221737304508e-13} \end{bmatrix}$$

Thus the ranking of the top 10 websites from most to least important is given by the row vectors:

$$\begin{array}{l} b_1 = \begin{bmatrix} \texttt{629103} & \texttt{328995} & \texttt{176090} & \texttt{5397} & \texttt{609117} & \texttt{539881} & \texttt{63085} & \texttt{351516} & \texttt{529968} & \texttt{363332} \end{bmatrix} \\ b_2 = \begin{bmatrix} \texttt{629103} & \texttt{328995} & \texttt{176090} & \texttt{5397} & \texttt{609117} & \texttt{539881} & \texttt{63085} & \texttt{351516} & \texttt{529968} & \texttt{363332} \end{bmatrix}. \end{array}$$

Problem 5

(a)

The eigenvalues of C can be found by `conj(fft(c'))` by the following formulas.

$$\begin{aligned} \left[Fc^T\right]_j &= \sum_{k=0}^{n-1} c_k \exp\left(\frac{-2\pi i}{n} jk\right), \quad \forall\, 0 \le j \le n-1 \\ \overline{\left[Fc^T\right]_j} &= \sum_{k=0}^{n-1} c_k \exp\left(\frac{2\pi i}{n} jk\right), \quad \forall\, 0 \le j \le n-1 \end{aligned}$$

(b)

Let

$$\Lambda = \begin{bmatrix} \lambda_0 & 0 & \cdots & 0 \\ 0 & \lambda_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_{n-1} \end{bmatrix}$$

and `lambda` $=$ `conj(fft(c'))`. Then $\Lambda F x =$ `lambda.*fft(x)`. Also,

$$\begin{aligned} \left[\overline{F}x\right]_{j=0,1,\ldots,n-1} &= \left[\sum_{k=0}^{n-1} x_k \exp\left(\frac{2\pi i}{n} jk\right)\right]_{j,k=0,1,\ldots,n-1} \\ &= \overline{\left[\sum_{k=0}^{n-1} \overline{x_k} \exp\left(\frac{-2\pi i}{n} jk\right)\right]_{j=0,1,\ldots,n-1}} \\ &= \overline{\left[F\overline{x}\right]}_{j=0,1,\ldots,n-1} \\ &= \texttt{conj(fft(conj(x)))} \end{aligned}$$

Therefore, $y = Cx = \frac{1}{n}\overline{F}\Lambda F x =$ `conj(fft(conj(fft(c’)).*fft(x))))/length(x)`. Since y is real, then $y =$ `fft(conj(fft(c’)).*fft(x)))/length(x)`. The following MATLAB code was written to compare the two methods of multiplying a circulant matrix:

```
format long e;
c = [2,-1,0,-3];
C = [2,-1,0,-3;...
    -3,2,-1,0;...
    0,-3,2,-1;...
    -1,0,-3,2];
x = [-1,2,1,4]';
matrix2latex(C*x,'Cx.tex','alignment','r','format','%-.15e')
matrix2latex(circulant(c,x),'circulantx.tex','alignment','r','format','%-.15e')
```

where the function `circulant` is given by

```
function y = circulant(c,x)
%Multiply a column vector by a circulant matrix using fft.
%   input  - c    row vector that generates a circulant matrix, C
%          - x    column vector
%   output - C*x  matrix multiplication

n = length(x);
y = fft(conj(conj(fft(c')).*fft(x)))/n;
end
```

with the following output

$$\begin{aligned}
\texttt{C*x} &= \begin{bmatrix} \texttt{-1.600000000000000e+01} \\ \texttt{6.000000000000000e+00} \\ \texttt{-8.000000000000000e+00} \\ \texttt{6.000000000000000e+00} \end{bmatrix} \\
\texttt{circulant(c,x)} &= \begin{bmatrix} \texttt{-1.600000000000000e+01} \\ \texttt{6.000000000000000e+00} \\ \texttt{-8.000000000000000e+00} \\ \texttt{6.000000000000000e+00} \end{bmatrix}.
\end{aligned}$$

(c)

If we extend the Toeplitz matrix such that $t_j = t_{j+1-2n}, t_{-j} = t_{2n-1-j}$ for $n < j \leq 2n-1$, then by definition we have a unique, circulant matrix such that $c_j := t_j$.

(d)

The following MATLAB function uses part (c) and part (b) to compute $y = Tx$:

```
function y = toeplitz(t,x)
%Multiply a column vector by a Toeplitz matrix using fft.
%   input  - t    row vector that generates a Toeplitz matrix, T
%          - x    column vector
%   output - T*x  matrix multiplication
n = length(x);
c = zeros(2*n-1,1);
c(1:n) = t(n:end);
c(n+1:end) = t(1:n-1);

d = zeros(2*n-1,1);
d(1:n) = x;
y = circulant(c',d);
y = y(1:n);
end
```

(e)

The following MATLAB code was written to compare the two methods of multiplying a Toeplitz matrix

```
format long e;
t = [6,-5,1,2,-1,4,-3];
T = [2,-1,4,-3;...
    1,2,-1,4;...
    -5,1,2,-1;...
    6,-5,1,2];
x = [-1,2,1,4]';
matrix2latex(T*x,'Tx.tex','alignment','r','format','%-.15e')
matrix2latex(toeplitz(t,x),'toeplitzx.tex','alignment','r','format','%-.15e')
```

and has the following output

$$
\begin{aligned}
\texttt{T*x} &= \begin{bmatrix} \texttt{-1.200000000000000e+01} \\ \texttt{1.800000000000000e+01} \\ \texttt{5.000000000000000e+00} \\ \texttt{-7.000000000000000e+00} \end{bmatrix} \\
\texttt{toeplitz(t,x)} &= \begin{bmatrix} \texttt{-1.200000000000000e+01} \\ \texttt{1.800000000000000e+01} \\ \texttt{5.000000000000000e+00} \\ \texttt{-7.000000000000001e+00} \end{bmatrix}.
\end{aligned}
$$

For the $n = 500000$ case, the code is

```
load('t_and_x.mat')
y = toeplitz(t,x);
matrix2latex([y(1),y(100000),y(200000),y(300000),y(400000),y(500000),sum(y),norm(y,2)]','t_and_x.tex','alignment','r','format','%-.15e')
```

and has the output

$$\begin{bmatrix} y_1 \\ y_{100000} \\ y_{200000} \\ y_{300000} \\ y_{400000} \\ y_{500000} \\ \sum_{j=1}^{n} y_j \\ \|y\|_2 \end{bmatrix} = \begin{bmatrix} \texttt{4.743400116337570e+01-1.897571643350681e-14i} \\ \texttt{-2.044536337994736e+02+3.463356617877540e-14i} \\ \texttt{9.854729389973313e+01-2.893191580611515e-14i} \\ \texttt{-2.273143080393689e+01+4.576858382960006e-14i} \\ \texttt{4.136625635601638e+02-5.480763011145771e-14i} \\ \texttt{2.524304751530100e+02+9.171369375379089e-14i} \\ \texttt{1.030143367304304e+05+7.836054370961132e-13i} \\ \texttt{1.669299439051479e+05} \end{bmatrix}.$$