Math 1080: Spring 2011

Homework #9 (due April 8)

Problem 1:

Calculate the Rayleigh quotients $r_1 = r(\mathbf{x}_1)$ and $r_2 = r(\mathbf{x}_2)$ for the following matrix **A** and vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. How close are r_1 and r_2 to an eigenvalue of **A** ?

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & 1 \\ 4 & 6 & 4 \\ 1 & 4 & 1 \end{bmatrix} \qquad \mathbf{x}_1 = \begin{bmatrix} 1.5 \\ 2 \\ 1 \end{bmatrix} \qquad \mathbf{x}_2 = \begin{bmatrix} 1 \\ 2.1 \\ 1 \end{bmatrix}$$

SOLUTION:

The Rayleigh quotient of a matrix **A** and vector **x** is equal to $r(\mathbf{x}) = \dfrac{\mathbf{x}^T \mathbf{A}\mathbf{x}}{\mathbf{x}^T \mathbf{x}}$.

Here

$$\mathbf{x}_1^T \mathbf{x}_1 = \frac{29}{4}, \quad \mathbf{x}_1^T \mathbf{A}\mathbf{x}_1 = \begin{bmatrix} 21/2 & 22 & 21/2 \end{bmatrix} \begin{bmatrix} 1.5 \\ 2 \\ 1 \end{bmatrix} = \frac{281}{4}, \quad r_1 = \frac{\mathbf{x}_1^T \mathbf{A}\mathbf{x}_1}{\mathbf{x}_1^T \mathbf{x}_1} = \frac{281}{29} = 9.6897$$

$$\mathbf{x}_2^T \mathbf{x}_2 = 6.41, \quad \mathbf{x}_1^T \mathbf{A}\mathbf{x}_1 = \begin{bmatrix} 10.4 & 20.6 & 10.4 \end{bmatrix} \begin{bmatrix} 1 \\ 2.1 \\ 1 \end{bmatrix} = 64.06, \quad r_1 = \frac{\mathbf{x}_1^T \mathbf{A}\mathbf{x}_1}{\mathbf{x}_1^T \mathbf{x}_1} = \frac{64.06}{6.41} = 9.9938$$

The eigenvalues of **A** are given by

$$0 = \det(\mathbf{A} - \lambda \mathbf{I}) = \begin{bmatrix} 1-\lambda & 4 & 1 \\ 4 & 6-\lambda & 4 \\ 1 & 4 & 1-\lambda \end{bmatrix} = (1-\lambda)^2 (6-\lambda) + 32 - (6-\lambda) - 32(1-\lambda) =$$

$$= -\lambda^3 + 8\lambda^2 + 20\lambda = -\lambda(\lambda - 10)(\lambda + 2)$$

The eigenvalue closest to r_1 and r_2 is 10. The differences are $|r_1 - 10| = 0.3103$ and $|r_2 - 10| = 0.0062$.

Problem 2:

Determine one eigenvalue of the following matrix using Rayleigh Quotient iteration, starting with initial guess $\mathbf{v}^{(0)} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$. Terminate iteration after 3 steps, i.e., after you obtain $\lambda^{(3)}$. What is the approximate eigenvector $\mathbf{v}^{(3)}$? What is the error of $\lambda^{(3)}$?

$$\mathbf{A} = \begin{bmatrix} 3 & -2 \\ -2 & 6 \end{bmatrix}$$

SOLUTION:

$$\mathbf{v}^{(0)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \lambda^{(0)} = (\mathbf{v}^{(0)})^{\mathrm{T}} \mathbf{A}\mathbf{v}^{(0)} = 6$$

First iteration:

$$\text{The solution of } (\mathbf{A} - \lambda^{(0)}\mathbf{I})\mathbf{w} = \begin{bmatrix} -3 & -2 \\ -2 & 0 \end{bmatrix}\mathbf{w} = \mathbf{v}^{(0)} \quad \text{is} \quad \mathbf{w} = \begin{bmatrix} -1/2 \\ 3/4 \end{bmatrix}$$

$$\mathbf{v}^{(1)} = \frac{\mathbf{w}}{\|\mathbf{w}\|} = \frac{1}{\sqrt{13}}\begin{bmatrix} -2 \\ 3 \end{bmatrix} = \begin{bmatrix} -0.5547 \\ 0.8321 \end{bmatrix}$$

$$\lambda^{(1)} = (\mathbf{v}^{(1)})^{\mathrm{T}} \mathbf{A}\mathbf{v}^{(1)} = \frac{90}{13} = 6.9231$$

Second iteration:

$$\text{The solution of } (\mathbf{A} - \lambda^{(1)}\mathbf{I})\mathbf{w} = \begin{bmatrix} -51/13 & -2 \\ -2 & -12/13 \end{bmatrix}\mathbf{w} = \mathbf{v}^{(1)} \quad \text{is} \quad \mathbf{w} = \frac{1}{64\sqrt{13}}\begin{bmatrix} -1326 \\ 2665 \end{bmatrix}$$

$$\mathbf{v}^{(1)} = \frac{\mathbf{w}}{\|\mathbf{w}\|} = \frac{1}{13\sqrt{52429}}\begin{bmatrix} -1326 \\ 2665 \end{bmatrix} = \begin{bmatrix} -0.4455 \\ 0.8953 \end{bmatrix}$$

$$\lambda^{(2)} = (\mathbf{v}^{(2)})^{\mathrm{T}} \mathbf{A}\mathbf{v}^{(2)} = \frac{367002}{52429} = 6.9999809$$

Third iteration:

$$\text{Solution of } (\mathbf{A} - \lambda^{(2)}\mathbf{I})\mathbf{w} = \begin{bmatrix} -3.9999809 & -2 \\ -2 & -0.9999809 \end{bmatrix}\mathbf{w} = \mathbf{v}^{(2)} \quad \text{is} \quad \mathbf{w} = \begin{bmatrix} -23446.917 \\ 46893.834 \end{bmatrix}$$

$$\mathbf{v}^{(3)} = \frac{\mathbf{w}}{\|\mathbf{w}\|} = \begin{bmatrix} -0.44721360 \\ 0.89442719 \end{bmatrix}$$

$$\lambda^{(3)} = (\mathbf{v}^{(3)})^{\mathrm{T}} \mathbf{A}\mathbf{v}^{(3)} = 7$$

The eigenvalues of **A** are solutions of

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \begin{vmatrix} 3-\lambda & -2 \\ -2 & 6-\lambda \end{vmatrix} = \lambda^2 - 9\lambda + 14 = (\lambda - 7)(\lambda - 2) = 0$$

i.e., $\lambda_1 = 7$, $\lambda_2 = 2$. The error of $\lambda^{(3)}$ is smaller than the machine accuracy $\varepsilon_{machine} = 10^{-16}$.

Problem 3:

Perform the first three steps of the QR algorithm (i.e., compute $\mathbf{A}^{(2)}$ and $\tilde{\mathbf{Q}}^{(2)}$) for the following matrix. How close are the diagonal elements of $\mathbf{A}^{(2)}$ to the eigenvalues of **A**?

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

SOLUTION:

Put $\mathbf{A}^{(0)} = \mathbf{A}$.

Compute QR factorization of $\mathbf{A}^{(0)}$: $\mathbf{Q}^{(1)}\mathbf{R}^{(1)} = \mathbf{A}^{(0)}$,

$$\mathbf{Q}^{(1)} = \begin{bmatrix} 0.8944 & 0.3586 & 0.2673 \\ -0.4472 & 0.7171 & 0.5345 \\ 0 & -0.5976 & 0.8018 \end{bmatrix}, \quad \mathbf{R}^{(1)} = \begin{bmatrix} 2.2361 & -1.7889 & 0.4472 \\ 0 & 1.6733 & -1.9124 \\ 0 & 0 & 1.0690 \end{bmatrix}$$

$$\mathbf{A}^{(1)} = \mathbf{R}^{(1)}\mathbf{Q}^{(1)} = \begin{bmatrix} 2.8 & -0.7843 & 0 \\ -0.7843 & 2.3429 & -0.6389 \\ 0 & -0.6389 & 0.8571 \end{bmatrix}$$

Compute QR factorization of $\mathbf{A}^{(1)}$: $\mathbf{Q}^{(2)}\mathbf{R}^{(2)} = \mathbf{A}^{(1)}$,

$$\mathbf{Q}^{(2)} = \begin{bmatrix} 0.9661 & 0.2467 & 0.0761 \\ -0.2582 & 0.9231 & 0.2849 \\ 0 & -0.2949 & 0.9555 \end{bmatrix}, \quad \mathbf{R}^{(2)} = \begin{bmatrix} 2.8983 & -1.3279 & 0.1650 \\ 0 & 2.1665 & -0.8425 \\ 0 & 0 & 0.6370 \end{bmatrix}$$

$$\mathbf{A}^{(2)} = \mathbf{R}^{(2)}\mathbf{Q}^{(2)} = \begin{bmatrix} 3.1429 & -0.5594 & 0 \\ -0.5594 & 2.2484 & -0.1878 \\ 0 & -0.1878 & 0.6087 \end{bmatrix}$$

Compute QR factorization of $\mathbf{A}^{(2)}$: $\mathbf{Q}^{(3)}\mathbf{R}^{(3)} = \mathbf{A}^{(2)}$,

$$\mathbf{Q}^{(3)} = \begin{bmatrix} 0.9845 & 0.1745 & 0.0155 \\ -0.1752 & 0.9807 & 0.0871 \\ 0 & -0.0884 & 0.9961 \end{bmatrix}, \quad \mathbf{R}^{(3)} = \begin{bmatrix} 3.1923 & -0.9447 & 0.0329 \\ 0 & 2.1240 & -0.2381 \\ 0 & 0 & 0.5900 \end{bmatrix}$$

$$\mathbf{A}^{(2)} = \mathbf{R}^{(2)}\mathbf{Q}^{(2)} = \begin{bmatrix} 3.3084 & -0.3722 & 0 \\ -0.3722 & 2.1039 & -0.0522 \\ 0 & -0.0522 & 0.5876 \end{bmatrix}$$

The matrix $\tilde{\mathbf{Q}}^{(2)}$ is given by

$$\tilde{\mathbf{Q}}^{(3)} = \mathbf{Q}^{(1)}\mathbf{Q}^{(2)}\mathbf{Q}^{(3)} = \begin{bmatrix} 0.6767 & 0.5607 & 0.4771 \\ -0.6767 & 0.2185 & 0.7031 \\ 0.2900 & -0.7986 & 0.5273 \end{bmatrix}$$

The eigenvalues of $\mathbf{A}$ are solutions of

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \begin{vmatrix} 2-\lambda & -1 & 0 \\ -1 & 2-\lambda & -1 \\ 0 & -1 & 2-\lambda \end{vmatrix} = -\lambda^3 + 6\lambda^2 - 10\lambda + 4 = -(\lambda - 2)(\lambda^2 - 4\lambda + 2) = 0$$

i.e., $\lambda_1 = 2 + \sqrt{2} = 3.4142$, $\lambda_2 = 2$, $\lambda_2 = 2 - \sqrt{2} = 0.5858$. The difference between the diagonal elements of $\mathbf{A}^{(2)}$ and the eigenvalues of $\mathbf{A}$ is $\mathrm{diag}(\mathbf{A}^{(2)}) - [\lambda_1\ \lambda_2\ \lambda_3] = [-0.1058\ \ 0.1039\ \ 0.0019]$.

Computer Assignment 6:

a) Write a MATLAB function **`[lambda,v]=rayleigh(A,v0,mu,eps)`** that computes one eigenvalue of a square, symmetric matrix **A** using *Rayleigh Quotient iteration* with the initial guess **v0** and initial eigenvalue estimate **`mu`**. The program should terminate when the difference between two consecutive estimates, $\left|\lambda^{(k+1)} - \lambda^{(k)}\right|$, is smaller than the tolerance **`eps`** . The program should also print out $\lambda^{(k)}$ and $\mathbf{v}^{(k)}$ for each step of the iteration.

b) Calculate at least two different eigenvalues of the following matrix the **`rayleigh`** algorithm starting with two different initial guesses **`mu`** and using tolerance 10^{-6}. Then decrease the tolerance to 10^{-12} and note how many more steps the algorithm requires to converge. Compare this number with the number of steps needed for inverse iteration.

$$\mathbf{A} = \begin{bmatrix} 2 & 6 & 4 & -4 & -5 & -10 \\ 6 & 12 & -2 & 9 & 5 & 9 \\ 4 & -2 & 0 & -1 & -3 & 14 \\ -4 & 9 & -1 & 14 & -6 & 8 \\ -5 & 5 & -3 & -6 & 2 & 8 \\ -10 & 9 & 14 & 8 & 8 & 8 \end{bmatrix}$$

PROGRAMS:

```
function [r,v] = poweriter(A,v0,eps)
v = v0/norm(v0);
r = 0;
rold = 100;
k = 0;
disp('Iter    lambda        v');
while abs(r-rold) > eps
    rold = r;
    w = A*v;
    k = k + 1;
    v = w/norm(w);
    r = v'*A*v;
    disp([sprintf('%d\t %6.4f\t ',k,r),sprintf('%6.4f   ',v')])
end
```

```
function [r,v] = inverseiter(A,v0,mu,eps)
n = size(A);
v = v0/norm(v0);
r = 0;
rold = 100;
k = 0;
disp('Iter    lambda        v');
while abs(r-rold) > eps
    rold = r;
    w = (A-mu*eye(n))\v;
    k = k + 1;
    v = w/norm(w);
    r = v'*A*v;
    disp([sprintf('%d\t %6.4f\t ',k,r),sprintf('%6.4f   ',v')])
end
```

OUTPUT

```
>> A = [2 6 4 -4 -5 -10; 6 12 -2 9 5 9; 4 -2 0 -1 -3 14; -4 9 -1 14 -6 8; -5 5 -3 -6 2 8;
-10 9 14 8 8 8]

>> [r,v] = rayleigh(A,[1 0 0 0 0 0]',10,1e-6);
Iter    lambda      v
1       9.5596      -0.1920  0.4496  -0.6678  -0.1231  0.4799  -0.2638
2       8.8691      0.6400  0.1970  0.6250  -0.2916  -0.2362  0.1425
3       8.5143      -0.5083  0.0587  -0.7164  0.1185  0.4205  -0.1845
4       8.5070      0.5283  -0.0274  0.7118  -0.1426  -0.4003  0.1817
5       8.5070      -0.5282  0.0275  -0.7118  0.1426  0.4003  -0.1817

>> [r,v] = rayleigh(A,[1 0 0 0 0 0]',-10,1e-6);
Iter    lambda      v
1       -10.3032    -0.4653  0.4071  -0.0850  -0.4239  -0.6513  0.0820
2       -10.3079    0.4722  -0.3914  0.1111  0.4217  0.6505  -0.1031
3       -10.3079    0.4722  -0.3914  0.1111  0.4217  0.6505  -0.1031

>> [r,v] = rayleigh(A,[1 0 0 0 0 0]',10,1e-12);
Iter    lambda      v
1       9.5596      -0.1920  0.4496  -0.6678  -0.1231  0.4799  -0.2638
2       8.8691      0.6400  0.1970  0.6250  -0.2916  -0.2362  0.1425
3       8.5143      -0.5083  0.0587  -0.7164  0.1185  0.4205  -0.1845
4       8.5070      0.5283  -0.0274  0.7118  -0.1426  -0.4003  0.1817
5       8.5070      -0.5282  0.0275  -0.7118  0.1426  0.4003  -0.1817
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 5.677107e-017.
> In rayleigh at 10
6       8.5070      0.5282  -0.0275  0.7118  -0.1426  -0.4003  0.1817

>> [r,v] = rayleigh(A,[1 0 0 0 0 0]',-10,1e-12);
Iter    lambda      v
1       -10.3032    -0.4653  0.4071  -0.0850  -0.4239  -0.6513  0.0820
2       -10.3079    0.4722  -0.3914  0.1111  0.4217  0.6505  -0.1031
3       -10.3079    0.4722  -0.3914  0.1111  0.4217  0.6505  -0.1031
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 2.013324e-017.
> In rayleigh at 10
4       -10.3079    -0.4722  0.3914  -0.1111  -0.4217  -0.6505  0.1031

>> [r,v] = inverseiter(A,[1 0 0 0 0 0]',10,1e-6);
Iter    lambda      v
1       9.5596      -0.1920  0.4496  -0.6678  -0.1231  0.4799  -0.2638
2       9.4450      0.6686  0.3345  0.5252  -0.3770  -0.1059  0.1091
3       9.3769      -0.2390  0.3620  -0.6601  -0.1435  0.5675  -0.1833
4       9.3184      0.6513  0.3011  0.5560  -0.3805  -0.1193  0.1300
5       9.2629      -0.2673  0.3367  -0.6694  -0.1264  0.5640  -0.1818
6       9.2100      0.6477  0.2774  0.5748  -0.3665  -0.1422  0.1363
7       9.1598      -0.2911  0.3150  -0.6793  -0.1062  0.5566  -0.1834
8       9.1123      0.6439  0.2551  0.5913  -0.3517  -0.1645  0.1413
9       9.0674      -0.3130  0.2941  -0.6877  -0.0867  0.5485  -0.1848
10      9.0251      0.6394  0.2338  0.6060  -0.3372  -0.1853  0.1458
11      8.9854      -0.3333  0.2740  -0.6948  -0.0682  0.5402  -0.1859
12      8.9482      0.6344  0.2134  0.6190  -0.3232  -0.2046  0.1498
13      8.9135      -0.3519  0.2549  -0.7007  -0.0508  0.5318  -0.1867
14      8.8811      0.6291  0.1942  0.6305  -0.3097  -0.2225  0.1534
15      8.8510      -0.3689  0.2368  -0.7054  -0.0345  0.5234  -0.1873
16      8.8229      0.6236  0.1761  0.6406  -0.2969  -0.2389  0.1566
17      8.7970      -0.3843  0.2198  -0.7092  -0.0194  0.5152  -0.1877
18      8.7729      0.6179  0.1592  0.6495  -0.2848  -0.2539  0.1594
```

19	8.7507	-0.3984	0.2039	-0.7122	-0.0054	0.5072	-0.1879
20	8.7302	0.6123	0.1434	0.6573	-0.2733	-0.2676	0.1619
21	8.7113	-0.4110	0.1891	-0.7144	0.0076	0.4995	-0.1880
22	8.6939	0.6067	0.1288	0.6640	-0.2627	-0.2801	0.1642
23	8.6778	-0.4225	0.1754	-0.7162	0.0195	0.4921	-0.1879
24	8.6631	0.6013	0.1153	0.6699	-0.2527	-0.2914	0.1661
25	8.6496	-0.4328	0.1627	-0.7174	0.0305	0.4851	-0.1878
26	8.6372	0.5961	0.1029	0.6750	-0.2434	-0.3017	0.1679
27	8.6258	-0.4422	0.1509	-0.7183	0.0405	0.4785	-0.1876
28	8.6154	0.5911	0.0914	0.6795	-0.2349	-0.3110	0.1694
29	8.6059	-0.4506	0.1402	-0.7189	0.0497	0.4722	-0.1874
30	8.5972	0.5863	0.0809	0.6834	-0.2270	-0.3194	0.1707
31	8.5892	-0.4581	0.1303	-0.7192	0.0581	0.4664	-0.1871
32	8.5819	0.5818	0.0713	0.6868	-0.2197	-0.3271	0.1719
33	8.5753	-0.4649	0.1212	-0.7193	0.0657	0.4610	-0.1868
34	8.5692	0.5776	0.0625	0.6897	-0.2130	-0.3340	0.1730
35	8.5637	-0.4710	0.1129	-0.7193	0.0727	0.4560	-0.1866
36	8.5586	0.5736	0.0544	0.6923	-0.2068	-0.3402	0.1739
37	8.5540	-0.4765	0.1053	-0.7192	0.0791	0.4513	-0.1863
38	8.5498	0.5699	0.0471	0.6945	-0.2012	-0.3459	0.1747
39	8.5459	-0.4815	0.0983	-0.7190	0.0848	0.4470	-0.1860
40	8.5424	0.5665	0.0404	0.6965	-0.1960	-0.3510	0.1755
41	8.5393	-0.4860	0.0920	-0.7187	0.0901	0.4430	-0.1857
42	8.5363	0.5633	0.0343	0.6982	-0.1913	-0.3556	0.1761
43	8.5337	-0.4900	0.0862	-0.7184	0.0949	0.4393	-0.1854
44	8.5313	0.5603	0.0287	0.6997	-0.1869	-0.3598	0.1767
45	8.5291	-0.4936	0.0809	-0.7180	0.0992	0.4360	-0.1851
46	8.5271	0.5576	0.0236	0.7010	-0.1830	-0.3636	0.1772
47	8.5253	-0.4969	0.0761	-0.7177	0.1032	0.4329	-0.1849
48	8.5236	0.5551	0.0190	0.7022	-0.1794	-0.3670	0.1776
49	8.5221	-0.4999	0.0717	-0.7173	0.1068	0.4300	-0.1846
50	8.5208	0.5528	0.0148	0.7032	-0.1761	-0.3701	0.1781
51	8.5195	-0.5025	0.0677	-0.7170	0.1101	0.4274	-0.1844
52	8.5184	0.5507	0.0109	0.7041	-0.1731	-0.3729	0.1784
53	8.5173	-0.5050	0.0641	-0.7166	0.1130	0.4250	-0.1842
54	8.5164	0.5487	0.0075	0.7049	-0.1703	-0.3755	0.1787
55	8.5156	-0.5071	0.0608	-0.7163	0.1157	0.4229	-0.1840
56	8.5148	0.5469	0.0043	0.7056	-0.1678	-0.3778	0.1790
57	8.5141	-0.5091	0.0578	-0.7160	0.1182	0.4209	-0.1838
58	8.5134	0.5453	0.0014	0.7063	-0.1656	-0.3799	0.1793
59	8.5128	-0.5109	0.0550	-0.7157	0.1204	0.4190	-0.1836
60	8.5123	0.5438	-0.0012	0.7068	-0.1635	-0.3817	0.1795
61	8.5118	-0.5125	0.0525	-0.7154	0.1224	0.4174	-0.1835
62	8.5114	0.5424	-0.0036	0.7073	-0.1616	-0.3835	0.1797
63	8.5110	-0.5140	0.0503	-0.7151	0.1243	0.4159	-0.1833
64	8.5106	0.5412	-0.0058	0.7078	-0.1599	-0.3850	0.1799
65	8.5103	-0.5153	0.0482	-0.7148	0.1259	0.4145	-0.1832
66	8.5100	0.5400	-0.0077	0.7082	-0.1583	-0.3864	0.1801
67	8.5097	-0.5165	0.0463	-0.7146	0.1274	0.4132	-0.1831
68	8.5095	0.5390	-0.0095	0.7085	-0.1569	-0.3877	0.1802
69	8.5092	-0.5176	0.0446	-0.7144	0.1288	0.4120	-0.1829
70	8.5090	0.5380	-0.0112	0.7089	-0.1556	-0.3888	0.1804
71	8.5088	-0.5185	0.0431	-0.7141	0.1301	0.4110	-0.1828
72	8.5087	0.5371	-0.0127	0.7091	-0.1544	-0.3899	0.1805
73	8.5085	-0.5194	0.0417	-0.7139	0.1312	0.4100	-0.1827
74	8.5084	0.5363	-0.0140	0.7094	-0.1534	-0.3909	0.1806
75	8.5083	-0.5202	0.0404	-0.7138	0.1323	0.4092	-0.1826
76	8.5081	0.5356	-0.0152	0.7096	-0.1524	-0.3917	0.1807
77	8.5080	-0.5210	0.0392	-0.7136	0.1332	0.4084	-0.1826
78	8.5079	0.5349	-0.0163	0.7098	-0.1515	-0.3925	0.1808
79	8.5078	-0.5216	0.0381	-0.7134	0.1340	0.4076	-0.1825
80	8.5078	0.5343	-0.0174	0.7100	-0.1507	-0.3932	0.1809
81	8.5077	-0.5222	0.0372	-0.7133	0.1348	0.4070	-0.1824

```
82      8.5076       0.5338  -0.0183  0.7102  -0.1500  -0.3939  0.1810
83      8.5076       -0.5228  0.0363  -0.7132  0.1355  0.4064  -0.1824
84      8.5075       0.5333  -0.0191  0.7103  -0.1493  -0.3945  0.1810
85      8.5075       -0.5233  0.0355  -0.7131  0.1362  0.4058  -0.1823
86      8.5074       0.5328  -0.0199  0.7105  -0.1487  -0.3950  0.1811
87      8.5074       -0.5237  0.0348  -0.7129  0.1368  0.4053  -0.1822
88      8.5073       0.5324  -0.0206  0.7106  -0.1481  -0.3955  0.1812
89      8.5073       -0.5241  0.0341  -0.7128  0.1373  0.4049  -0.1822
90      8.5073       0.5320  -0.0212  0.7107  -0.1476  -0.3959  0.1812
91      8.5073       -0.5245  0.0335  -0.7127  0.1378  0.4045  -0.1821
92      8.5072       0.5317  -0.0218  0.7108  -0.1472  -0.3963  0.1813
93      8.5072       -0.5249  0.0330  -0.7127  0.1382  0.4041  -0.1821
94      8.5072       0.5314  -0.0223  0.7109  -0.1468  -0.3967  0.1813
95      8.5072       -0.5252  0.0325  -0.7126  0.1386  0.4037  -0.1821
96      8.5071       0.5311  -0.0228  0.7110  -0.1464  -0.3970  0.1813
97      8.5071       -0.5254  0.0320  -0.7125  0.1390  0.4034  -0.1820
98      8.5071       0.5308  -0.0232  0.7111  -0.1460  -0.3973  0.1814
99      8.5071       -0.5257  0.0316  -0.7125  0.1393  0.4032  -0.1820
100     8.5071       0.5306  -0.0236  0.7111  -0.1457  -0.3976  0.1814
101     8.5071       -0.5259  0.0312  -0.7124  0.1396  0.4029  -0.1820
102     8.5071       0.5304  -0.0239  0.7112  -0.1454  -0.3978  0.1814
103     8.5071       -0.5261  0.0309  -0.7123  0.1399  0.4027  -0.1820
104     8.5071       0.5302  -0.0243  0.7112  -0.1452  -0.3981  0.1814
105     8.5070       -0.5263  0.0306  -0.7123  0.1401  0.4024  -0.1819
106     8.5070       0.5300  -0.0246  0.7113  -0.1449  -0.3983  0.1815
107     8.5070       -0.5265  0.0303  -0.7122  0.1403  0.4023  -0.1819
108     8.5070       0.5298  -0.0248  0.7113  -0.1447  -0.3985  0.1815
109     8.5070       -0.5266  0.0301  -0.7122  0.1405  0.4021  -0.1819
110     8.5070       0.5297  -0.0251  0.7114  -0.1445  -0.3986  0.1815
111     8.5070       -0.5268  0.0298  -0.7122  0.1407  0.4019  -0.1819
112     8.5070       0.5296  -0.0253  0.7114  -0.1444  -0.3988  0.1815
113     8.5070       -0.5269  0.0296  -0.7121  0.1409  0.4018  -0.1819
114     8.5070       0.5294  -0.0255  0.7115  -0.1442  -0.3989  0.1815
115     8.5070       -0.5270  0.0294  -0.7121  0.1411  0.4016  -0.1818
116     8.5070       0.5293  -0.0257  0.7115  -0.1441  -0.3990  0.1816
117     8.5070       -0.5271  0.0293  -0.7121  0.1412  0.4015  -0.1818
118     8.5070       0.5292  -0.0258  0.7115  -0.1439  -0.3992  0.1816
119     8.5070       -0.5272  0.0291  -0.7120  0.1413  0.4014  -0.1818
120     8.5070       0.5291  -0.0260  0.7115  -0.1438  -0.3993  0.1816
121     8.5070       -0.5273  0.0289  -0.7120  0.1414  0.4013  -0.1818
122     8.5070       0.5290  -0.0261  0.7116  -0.1437  -0.3994  0.1816
123     8.5070       -0.5274  0.0288  -0.7120  0.1415  0.4012  -0.1818
124     8.5070       0.5290  -0.0263  0.7116  -0.1436  -0.3994  0.1816
125     8.5070       -0.5275  0.0287  -0.7120  0.1416  0.4011  -0.1818
126     8.5070       0.5289  -0.0264  0.7116  -0.1435  -0.3995  0.1816

>> [r,v] = inverseiter(A,[1 0 0 0 0 0]',-10,1e-6);
Iter    lambda        v
1       -10.3032     -0.4653  0.4071  -0.0850  -0.4239  -0.6513  0.0820
2       -10.3079     0.4729  -0.3915  0.1107  0.4215  0.6502  -0.1025
3       -10.3079     -0.4722  0.3914  -0.1110  -0.4217  -0.6505  0.1031
4       -10.3079     0.4722  -0.3914  0.1111  0.4217  0.6505  -0.1031
```

Decrease the tolerance to 10^{-12} causes error message because the estimated eigenvalue is already close to the real one - subtraction of mu*I causes the matrix to become close to singular and prevents accurate solution for w. Inverse iteraction efficiency is reasonable for mu = -10 but low for mu = 10. Convergence to the eigenvalue 8.5070 is difficult for that method.