**Scientific Computing 2**

**Mid Exam**

Jaya Harsha Thippana

113422600

**Problem 1:**  To show that PCA is optimal there are 2 key steps in this problem:

Step 1: When given a k dimensional space we need to find a best approximation for x(k) where x(k) belongs to Sk

Step 2: Find the specific subspace sk such that the maximum variance is present. If we are able to explain 90% variance of the data then we can approximate it to the entire data.

Lets suppose there’s a matrix H which is of dimension m\*k and this matrix is Symmetric and Idempotent.

Let x(k) = H\*alpha where alpha belong to Rk

By using Orthogonal projection theorem we can say that alpha = HT \* X

Now we can compute each column of alpha individually alphai = hTi \*X where I is between 1 and k that is 1<= i<= k.

Here we are trying to minimize the amount of data we are using by not considering all the columns.

By reducing this we will have some error within the process which we can explain by e = x – x(h)

E = (Im – H\*HT)\*x

**Problem 2:**

**Dataset 1:** Annual U.S. suicide rate (per 100,1000) from 1920 to 1969:

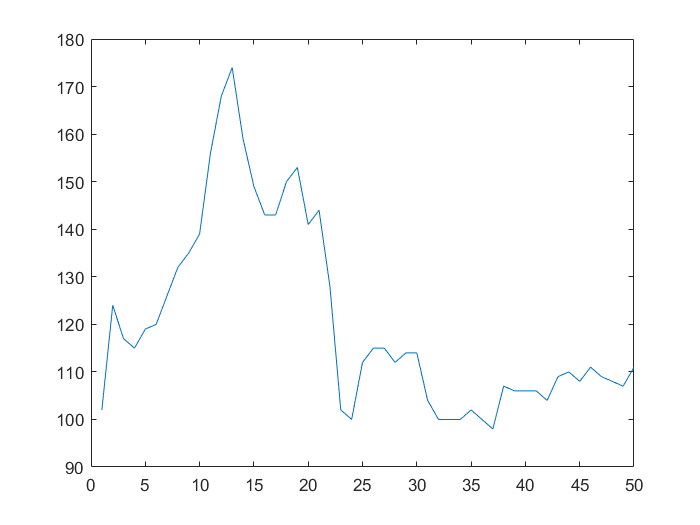
disp(Dataset1)

X = table2array(Dataset1)

X = 50×2

1920 102  
 1921 124  
 1922 117  
 1923 115  
 1924 119  
 1925 120  
 1926 126  
 1927 132  
 1928 135  
 1929 139

plot(X(:,2))



m = 15

for j = 1:n-m+1

for i =1:m

Y(i,j) = X(j+i-1,2)

end

end

secmom = Y \* Y'

secmom =

1.1708 1.1670 1.1590 1.1528 1.1474 1.1419 1.1360 1.1293 1.1213 1.1109 1.0992 1.0841 1.0671 1.0505 1.0380  
 1.1670 1.1692 1.1627 1.1559 1.1501 1.1439 1.1378 1.1316 1.1239 1.1149 1.1043 1.0888 1.0710 1.0526 1.0398  
 1.1590 1.1627 1.1613 1.1564 1.1501 1.1433 1.1364 1.1298 1.1224 1.1136 1.1042 1.0890 1.0702 1.0507 1.0369  
 1.1528 1.1559 1.1564 1.1564 1.1519 1.1447 1.1373 1.1300 1.1223 1.1137 1.1046 1.0908 1.0725 1.0522 1.0370  
 1.1474 1.1501 1.1501 1.1519 1.1524 1.1470 1.1392 1.1314 1.1230 1.1141 1.1053 1.0918 1.0751 1.0552 1.0392  
 1.1419 1.1439 1.1433 1.1447 1.1470 1.1466 1.1405 1.1323 1.1233 1.1137 1.1046 1.0913 1.0747 1.0563 1.0409  
 1.1360 1.1378 1.1364 1.1373 1.1392 1.1405 1.1394 1.1330 1.1235 1.1134 1.1035 1.0898 1.0734 1.0552 1.0413  
 1.1293 1.1316 1.1298 1.1300 1.1314 1.1323 1.1330 1.1314 1.1237 1.1130 1.1025 1.0879 1.0710 1.0529 1.0393  
 1.1213 1.1239 1.1224 1.1223 1.1230 1.1233 1.1235 1.1237 1.1208 1.1118 1.1007 1.0853 1.0673 1.0486 1.0353  
 1.1109 1.1149 1.1136 1.1137 1.1141 1.1137 1.1134 1.1130 1.1118 1.1076 1.0982 1.0822 1.0633 1.0435 1.0297

eigen = eig(secmom)

eigen =

0.7804  
 1.9954  
 4.1725  
 9.9011  
 11.2593  
 11.3675  
 18.8249  
 31.2548  
 37.1625  
 42.6535

[eigvec,eigval] = eig(secmom)

eigvec =

-0.0062 0.0926 0.1288 -0.1519 0.0314 0.0055 0.3320 0.2867 0.0298 0.4708 0.3052 -0.2804 0.3478 0.4164 0.2638  
 0.0960 -0.2172 -0.3051 0.3944 0.0045 -0.0574 -0.5232 -0.1486 -0.2172 -0.0936 0.0247 -0.1343 0.3261 0.3808 0.2644  
 -0.2544 0.2166 0.3717 -0.4226 -0.1564 -0.0006 -0.0423 -0.2161 -0.1079 -0.4652 -0.2197 0.1044 0.1515 0.3446 0.2639  
 0.3378 -0.1096 -0.3861 0.0207 0.1249 0.2684 0.4082 0.1496 0.2620 -0.2598 -0.2328 0.3283 -0.0710 0.2738 0.2638  
 -0.3458 -0.0844 0.3121 0.3479 0.2171 -0.2973 -0.1284 0.0740 0.3574 0.2048 -0.0468 0.3764 -0.2839 0.1837 0.2637  
 0.3097 0.2897 -0.1997 -0.2087 -0.4100 -0.1252 -0.1518 -0.2916 -0.0643 0.3472 0.1580 0.1928 -0.4241 0.0938 0.2633  
 -0.1800 -0.4063 0.1586 0.0382 0.0976 0.4624 0.1450 -0.1220 -0.4472 0.0039 0.2081 -0.1107 -0.4383 0.0078 0.2625  
 0.0277 0.4334 -0.1490 0.0451 0.3734 -0.3200 0.0038 0.2985 -0.1917 -0.3098 0.0467 -0.3678 -0.3361 -0.0688 0.2616  
 0.1120 -0.4253 0.1220 -0.0860 -0.4743 -0.1657 -0.0660 0.2461 0.3242 -0.1113 -0.2073 -0.4447 -0.1502 -0.1358 0.2602  
 -0.2527 0.3468 -0.1346 0.0888 0.0750 0.4904 -0.1335 -0.1921 0.2703 0.2615 -0.3966 -0.2856 0.0464 -0.1994 0.2585

for i = 1:m

U1 = Y' \* (eigvec(:,i))/ sqrt(eigval(i,i))

U = [U,U1]

end

U =

0.2275 -0.2689 -0.0432 0.1302 -0.0356 -0.0375 -0.3844 0.1434 -0.2257 -0.2068 0.0693 0.1954 0.2487 -0.3446 0.1826  
 -0.2891 0.1629 -0.1398 -0.0112 0.0237 0.0168 0.2205 0.1029 -0.2606 0.2818 -0.0771 0.0471 0.2216 -0.3138 0.1865  
 0.2911 -0.1006 0.2462 -0.0921 0.0064 -0.0239 0.0953 -0.1590 0.0272 0.2676 -0.2258 -0.0945 0.0795 -0.3050 0.1883  
 -0.0684 0.3392 -0.0757 0.1925 -0.0165 -0.0025 0.0046 -0.1096 0.3133 0.0053 -0.1768 -0.2366 -0.0642 -0.2762 0.1914  
 -0.1367 -0.3716 0.0301 -0.0341 0.0186 -0.0285 0.0942 0.0352 0.2330 -0.2742 0.0054 -0.2618 -0.1976 -0.2352 0.1948  
 -0.0118 0.1415 -0.2433 -0.3043 0.1291 0.0940 -0.1813 0.1025 -0.1497 -0.2384 0.1146 -0.1141 -0.3091 -0.1794 0.1970  
 -0.0316 0.0023 0.3248 0.2446 -0.1920 0.1217 -0.0825 -0.0554 -0.2620 0.0198 0.1418 0.0869 -0.3199 -0.1196 0.1992  
 -0.1791 -0.1996 -0.4775 -0.0794 -0.0298 -0.2357 0.0187 -0.0639 -0.0500 0.2626 -0.0291 0.2541 -0.2776 -0.0447 0.1995  
 0.0525 -0.1204 0.2479 -0.2929 0.3188 0.1175 -0.2076 0.2177 0.0553 0.2103 -0.3517 0.2808 -0.2141 0.0530 0.1971  
 0.0247 0.2273 -0.1272 0.2330 -0.3188 0.2480 -0.0131 0.1913 -0.0885 -0.0783 -0.5276 0.0919 -0.1214 0.1499 0.1942

%PCA Analysis Code%

p = sum(eigen)

p = 1.6624e+05

a = eigen

a =

0.7804  
 1.9954  
 4.1725  
 9.9011  
 11.2593  
 11.3675  
 18.8249  
 31.2548  
 37.1625  
 42.6535

prompt = 'Enter the beta value'

prompt = 'Enter the beta value'

beta = double(input(prompt))

beta = 0.9500

co = 0

co = 0

estimated = 1.00

estimated = 1

while estimated > beta

[a , estimated] = function23(a, p)

co = co + 1

if estimated <= beta

disp(a)

break

end

end

y = 0.9953

a =

0.0115  
 1.6430

estimated = 0.9953

co = 13

index = 1

a = 1.6430e+05

newa = 1.6430e+05

newsum = 1.6430e+05

y = 0.9884

a = 1.6430e+05

estimated = 0.9884

co = 14

index = 1

co = 15

PNoise = zeros(36,15)

for i= 1:14

%index = find(eigen == a(i))

p1 = sqrt(eigval(i,i))\*U(:,i)\*eigvec(:,i)'

PNoise = PNoise + p1

end

PNoise =

-5.1021 -2.0351 -2.9929 -3.2665 -2.6936 -2.5184 -1.6160 -0.6966 -0.1716 0.5225 3.0693 5.0012 6.1477 4.3567 3.1891  
 -2.4041 -3.4393 -3.6893 -3.1141 -2.9653 -2.0824 -1.1788 -0.6822 -0.0136 2.5217 4.3644 5.4528 3.6357 2.5588 1.9621  
 -3.5895 -3.9181 -3.3192 -3.1682 -2.3123 -1.4290 -0.9491 -0.3104 2.1977 4.0272 5.0227 3.1436 2.0367 1.5287 1.7830  
 -4.1981 -3.6789 -3.5038 -2.6455 -1.7894 -1.3298 -0.7076 1.7708 3.5734 4.5565 2.5846 1.4165 0.8802 1.2263 2.4747  
 -4.0043 -3.9102 -3.0274 -2.1689 -1.7370 -1.1353 1.3264 3.0990 4.0551 2.0707 0.8087 0.2107 0.5287 1.8710 2.5584  
 -4.0878 -3.2871 -2.4039 -1.9696 -1.3961 1.0444 2.7996 3.7245 1.7119 0.4361 -0.2586 -0.0053 1.3060 2.0864 0.6553  
 -3.4805 -2.6803 -2.2209 -1.6451 0.7669 2.5007 3.4081 1.3641 0.0598 -0.6487 -0.4932 0.7530 1.5023 0.1655 0.8587  
 -2.6684 -2.2926 -1.6918 0.7225 2.4275 3.3128 1.2504 -0.0864 -0.8248 -0.6845 0.4613 1.1422 -0.2293 0.5558 -1.4374  
 -1.9841 -1.4662 0.9726 2.6797 3.5360 1.4511 0.0951 -0.6770 -0.5682 0.5603 1.1385 -0.3051 0.4409 -1.4654 -4.8762  
 -1.1041 1.2530 2.9842 3.8426 1.7290 0.3506 -0.4406 -0.3653 0.7318 1.2926 -0.2528 0.4211 -1.5248 -4.8506 -4.8712

P = sqrt(eigval(15,15))\*U(:,15)\*eigvec(:,15)'

P =

19.5270 19.5714 19.5392 19.5299 19.5227 19.4890 19.4351 19.3642 19.2635 19.1350 18.9924 18.7576 18.4596 18.1293 17.8827  
 19.9403 19.9856 19.9527 19.9433 19.9359 19.9015 19.8464 19.7740 19.6711 19.5400 19.3944 19.1546 18.8502 18.5130 18.2612  
 20.1358 20.1815 20.1484 20.1388 20.1314 20.0966 20.0410 19.9679 19.8640 19.7316 19.5846 19.3424 19.0351 18.6945 18.4402  
 20.4616 20.5080 20.4743 20.4646 20.4570 20.4217 20.3652 20.2909 20.1854 20.0508 19.9014 19.6553 19.3430 18.9969 18.7385  
 20.8335 20.8808 20.8465 20.8365 20.8288 20.7929 20.7354 20.6597 20.5522 20.4152 20.2631 20.0125 19.6946 19.3422 19.0791  
 21.0584 21.1062 21.0715 21.0615 21.0537 21.0174 20.9592 20.8828 20.7741 20.6356 20.4819 20.2286 19.9072 19.5510 19.2851  
 21.2995 21.3479 21.3128 21.3027 21.2948 21.2580 21.1992 21.1219 21.0120 20.8720 20.7164 20.4602 20.1352 19.7749 19.5059  
 21.3361 21.3845 21.3494 21.3392 21.3313 21.2945 21.2356 21.1581 21.0480 20.9077 20.7519 20.4953 20.1697 19.8088 19.5394  
 21.0760 21.1238 21.0891 21.0791 21.0713 21.0349 20.9767 20.9002 20.7915 20.6529 20.4990 20.2455 19.9238 19.5674 19.3012  
 20.7617 20.8088 20.7746 20.7647 20.7570 20.7212 20.6639 20.5885 20.4814 20.3449 20.1933 19.9435 19.6267 19.2755 19.0133

%Actual Matrix = P after removing the noise. Noise of the matrix is PNOISE

%computed with beta = 0.05

%Problem 2.c Identifying Noise Component Based on PCA Analysis by taking Beta value as 0.05%

newm = zeros(36,15)

for i = 1:m

newm = newm + eigval(i,i)\*U(:,i)\*eigvec(:,i)'

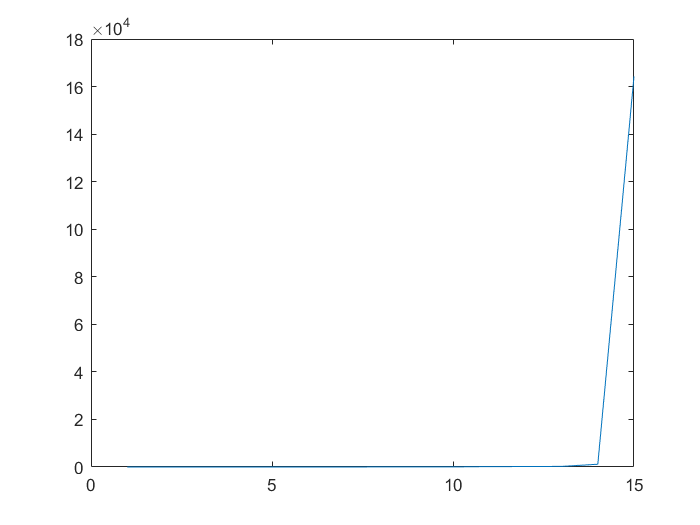
end

newm =

7.7587 7.8040 7.7978 7.8107 7.8333 7.8422 7.8477 7.8476 7.8339 7.8182 7.8070 7.7422 7.6291 7.4746 7.3609  
 7.9527 7.9784 7.9698 7.9836 8.0012 8.0123 8.0182 8.0168 8.0086 7.9973 7.9752 7.8926 7.7583 7.6034 7.4928  
 8.0210 8.0526 8.0516 8.0652 8.0869 8.1002 8.1067 8.1109 8.1105 8.0928 8.0527 7.9483 7.8093 7.6593 7.5532  
 8.1589 8.1903 8.1874 8.2036 8.2269 8.2416 8.2534 8.2647 8.2567 8.2193 8.1554 8.0454 7.9118 7.7697 7.6700  
 8.3247 8.3494 8.3478 8.3646 8.3888 8.4089 8.4269 8.4305 8.4020 8.3395 8.2694 8.1643 8.0395 7.9041 7.8035  
 8.4300 8.4570 8.4572 8.4754 8.5041 8.5283 8.5365 8.5197 8.4658 8.3972 8.3321 8.2352 8.1171 7.9773 7.8635  
 8.5496 8.5763 8.5768 8.5989 8.6317 8.6467 8.6354 8.5929 8.5327 8.4686 8.4111 8.3212 8.1991 8.0492 7.9354  
 8.5920 8.6189 8.6267 8.6543 8.6770 8.6706 8.6313 8.5822 8.5266 8.4713 8.4195 8.3240 8.1903 8.0387 7.9121  
 8.5221 8.5576 8.5751 8.5944 8.5942 8.5565 8.5075 8.4624 8.4157 8.3672 8.3081 8.1989 8.0614 7.8948 7.7610  
 8.4445 8.4870 8.4971 8.4940 8.4626 8.4162 8.3719 8.3348 8.2951 8.2393 8.1642 8.0501 7.8965 7.7279 7.6157

plot(eigen)

Eigen Values VS I graph



%Problem 2.d identifying different Components%

trendcomponent = eigval(14,14)\*U(:,14)\*eigvec(:,14)' + eigval(15,15)\*U(:,15)\*eigvec(:,15)'

trendcomponent =

7.7498 7.7819 7.7833 7.8076 7.8405 7.8625 7.8748 7.8765 7.8623 7.8355 7.8020 7.7195 7.6002 7.4573 7.3507  
 7.9321 7.9633 7.9631 7.9849 8.0145 8.0330 8.0418 8.0402 8.0227 7.9925 7.9557 7.8700 7.7480 7.6031 7.4950  
 8.0156 8.0466 8.0459 8.0669 8.0956 8.1131 8.1208 8.1181 8.0995 8.0682 8.0301 7.9432 7.8199 7.6739 7.5650  
 8.1615 8.1916 8.1895 8.2081 8.2337 8.2480 8.2525 8.2467 8.2253 8.1910 8.1499 8.0603 7.9349 7.7874 7.6774  
 8.3319 8.3607 8.3566 8.3718 8.3931 8.4028 8.4029 8.3930 8.3676 8.3293 8.2842 8.1913 8.0634 7.9145 7.8033  
 8.4498 8.4765 8.4700 8.4805 8.4960 8.4999 8.4941 8.4790 8.4488 8.4058 8.3561 8.2601 8.1306 7.9815 7.8703  
 8.5763 8.6008 8.5915 8.5972 8.6064 8.6039 8.5919 8.5711 8.5358 8.4878 8.4332 8.3338 8.2025 8.0534 7.9421  
 8.6270 8.6485 8.6361 8.6356 8.6371 8.6268 8.6073 8.5799 8.5387 8.4851 8.4251 8.3227 8.1909 8.0435 7.9334  
 8.5685 8.5857 8.5694 8.5610 8.5524 8.5321 8.5033 8.4676 8.4194 8.3594 8.2932 8.1885 8.0579 7.9148 7.8079  
 8.4875 8.5005 8.4804 8.4642 8.4455 8.4154 8.3773 8.3336 8.2786 8.2123 8.1402 8.0335 7.9044 7.7660 7.6626

Oscilatorycomponent = eigval(12,12)\*U(:,12)\*eigvec(:,12)' + eigval(13,13)\*U(:,13)\*eigvec(:,13)'+ eigval(11,11)\*U(:,11)\*eigvec(:,11)'

Oscilatorycomponent =

14.7287 15.6502 10.9063 5.2754 -4.0977 -17.4639 -29.2562 -34.4406 -28.6370 -12.4847 7.6322 20.7624 23.3891 17.9331 13.7059  
 12.9444 16.6759 12.2447 2.0253 -11.8202 -23.9668 -28.1732 -22.7329 -9.7826 5.0884 15.2221 16.0364 10.2249 3.9377 3.8416  
 0.5961 8.0627 9.4088 1.3459 -10.8095 -17.9841 -14.6037 -1.7707 12.9348 21.1014 17.7433 5.7223 -7.4320 -13.9998 -11.5306  
 -1.8004 0.1393 -0.7516 -7.0156 -11.2592 -6.6269 6.0274 20.9085 28.9177 24.0349 7.5095 -10.2221 -20.5594 -19.4506 -12.5612  
 -2.8460 -9.4214 -13.0383 -13.3086 -4.9734 11.4996 27.6339 35.4235 29.8336 11.8064 -10.3489 -23.5068 -23.3504 -13.1145 -5.0181  
 -15.0696 -21.9729 -18.3155 -6.1731 12.9072 31.8089 40.5774 35.2061 17.5646 -4.8901 -21.8684 -24.9802 -16.9025 -6.2718 -4.1772  
 -25.5482 -27.9792 -15.6629 5.7417 28.1202 41.1924 38.4122 22.0241 -0.2738 -17.9338 -23.3411 -16.2630 -5.0569 0.1851 -5.4598  
 -39.6108 -29.5461 -4.4009 22.2421 38.6005 38.3976 24.2185 5.2093 -10.3506 -15.3669 -9.8186 -0.9450 2.4133 -4.9668 -18.2858  
 -51.9326 -26.3518 10.3966 35.3400 38.5270 24.1003 5.4265 -4.5867 -3.8945 5.2533 13.8634 12.6992 -0.9252 -22.5913 -39.2753  
 -42.5050 -14.5397 16.5705 28.5077 19.5002 2.4924 -6.8828 -0.3695 14.9376 28.4224 28.6472 13.0012 -12.1091 -35.3576 -45.2587

Noisecomponent = zeros(36,15)

Noisecomponent =

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

for i = 1:10

Noisecomponent = Noisecomponent + eigval(i,i)\*U(:,i)\*eigvec(:,i)'

end

Noisecomponent =

-5.8036 6.4861 3.6349 -2.1387 -3.1265 -2.8473 2.0903 5.4937 0.2196 -4.7988 -2.6086 1.9517 5.5165 -0.6477 -3.5178  
 7.6528 -1.6375 -5.5012 -3.2898 -1.4645 3.2587 4.5140 -0.6360 -4.3428 -0.2957 4.3596 6.5536 0.0845 -3.6450 -6.0323  
 4.8256 -2.0800 -3.7656 -3.0921 2.1385 5.0772 0.5097 -5.3682 -1.9590 3.4991 4.8287 -0.6505 -3.2011 -0.6022 -0.2755  
 -0.7736 -1.3900 -1.3584 2.5536 4.4375 0.2404 -5.0500 -2.9595 2.4967 4.2853 -2.0176 -4.6514 -2.5114 1.7704 5.1924  
 -4.3387 -1.8787 4.2160 6.1243 0.7167 -5.4552 -3.5970 2.1226 4.5640 -1.5858 -4.4727 -3.5222 -0.5642 2.7677 5.2739  
 -4.7618 2.4733 5.5137 1.0338 -4.7831 -3.4079 1.8088 5.4978 -0.5098 -3.7201 -2.1611 0.1272 3.4372 2.0469 -2.5756  
 -1.1573 3.5553 0.9220 -4.0450 -2.8423 1.5923 5.0827 -0.2607 -2.8259 -1.3138 1.2453 3.6929 1.6331 -4.3174 -1.1461  
 4.5621 -0.0084 -4.9973 -3.5633 1.3471 5.4134 -0.2616 -2.8916 -1.7818 1.5195 4.2049 2.2178 -3.0919 0.1699 -3.0548  
 5.6090 -1.7162 -4.7145 -1.9817 3.2780 0.2931 -1.1691 -0.6310 0.1588 2.5492 1.0335 -2.3110 4.4355 2.5627 -7.6244  
 -0.5121 1.0652 0.1081 1.3173 -2.4109 -1.7739 1.4617 1.5875 1.5751 -1.3363 -4.6950 3.6430 4.2391 -2.7559 -1.5895

function [a] = rmvfunction(a,leas)

index = find(a==leas)

a(index) = []

end

function [newa ,y] = function23(a, p)

newa = rmvfunction(a, min(a))

newsum = sum(newa)

y = newsum/p

end

**Dataset 2:** Total annual rainfall (in inches), London, England, 1813 to 1912

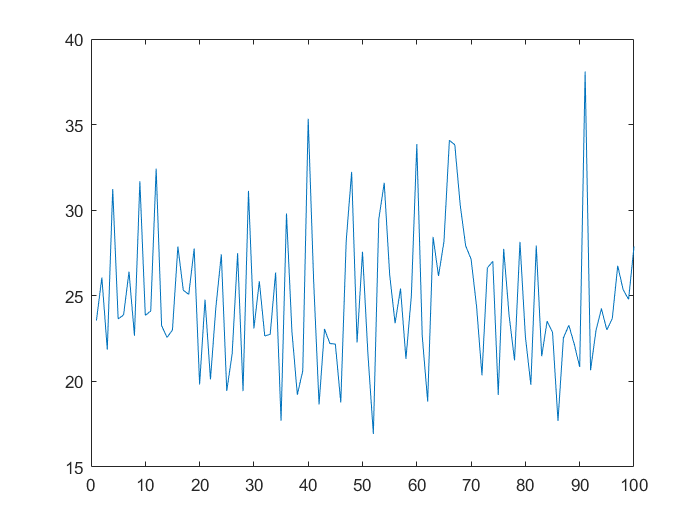
X = table2array(Dataset2)

X = 100×2

103 ×

1.8130 0.0236  
 1.8140 0.0261  
 1.8150 0.0219  
 1.8160 0.0312  
 1.8170 0.0236  
 1.8180 0.0239  
 1.8190 0.0264  
 1.8200 0.0227  
 1.8210 0.0317  
 1.8220 0.0239

plot(X(:,2))



m = 25

m = 25

for j = 1:n-m+1

for i =1:m

Y(i,j) = X(j+i-1,2)

end

end

secmom = Y \* Y'

eigen = eig(secmom)

eigen =

2.4741  
 7.4789  
 7.7391  
 7.8573  
 8.7943  
 9.2656  
 9.4269  
 9.5394  
 9.6925  
 10.3214

[eigvec,eigval] = eig(secmom)

eigvec =

-0.1373 0.4916 0.0555 -0.1371 0.2523 -0.0278 0.0397 -0.0273 -0.2135 -0.1675 -0.2115 0.0448 0.2085 0.1047 0.2459 -0.1309 0.3227 -0.0166 0.1511 0.0452 0.2989 0.2631 -0.2067 0.1646 0.2022  
 0.1779 -0.0711 -0.2097 0.1349 0.2398 0.1426 -0.1061 0.3346 0.2910 -0.0106 -0.3182 -0.0113 0.3118 0.1276 -0.1020 -0.0185 -0.2142 0.2591 -0.1706 -0.1335 -0.0758 0.3407 -0.1005 -0.2386 0.2022  
 -0.1769 0.1228 0.1089 0.1180 -0.2246 0.3746 -0.4039 -0.0222 0.0059 0.0094 -0.0708 0.0546 0.0544 -0.4027 -0.0355 0.2700 0.1936 -0.0538 0.2042 0.0546 -0.2281 0.2482 0.2921 -0.0362 0.2017  
 0.2053 -0.4181 0.0457 0.1442 0.1324 0.0377 -0.2762 -0.3361 0.0309 -0.0489 0.0113 0.0555 -0.1453 0.0949 0.2443 -0.1731 0.3397 0.2113 -0.0257 -0.3991 -0.0291 -0.0101 -0.0048 0.2526 0.2025  
 -0.2317 0.0588 0.2559 -0.2930 -0.1679 -0.1301 0.0060 -0.3744 0.2528 -0.1443 0.0196 -0.0217 0.0943 0.2242 -0.1649 0.2056 0.0518 0.4294 -0.1436 0.1808 -0.2311 -0.0675 -0.1998 -0.1147 0.2016  
 0.2627 -0.0759 -0.3093 -0.1788 0.1529 -0.0072 0.3342 -0.3128 0.1299 0.1289 -0.0261 -0.0413 0.0363 -0.2857 -0.1435 -0.0700 0.0293 0.1874 0.5152 0.1419 0.1072 -0.0267 0.1637 -0.1389 0.2012  
 -0.2608 0.0751 -0.2102 0.1412 -0.2945 0.2316 0.3533 0.0482 0.2211 -0.0568 0.1832 -0.1062 -0.1817 0.0024 0.1778 0.2167 -0.1126 0.2051 -0.1269 -0.2397 0.4053 0.0932 0.0882 0.1910 0.2016  
 0.2902 -0.1145 0.0704 0.3363 -0.2084 0.1371 0.2746 -0.1015 -0.3845 0.0674 0.0271 -0.1183 -0.1407 0.1477 -0.2728 0.0245 0.1195 -0.0485 -0.1245 0.3069 -0.0628 0.3534 -0.2350 0.0459 0.2012  
 -0.2765 -0.2393 0.3479 0.0222 -0.1184 -0.2104 0.2903 0.1523 -0.1449 -0.2711 -0.0588 -0.0695 -0.0537 -0.1137 0.1118 -0.3201 -0.1822 -0.0241 0.2578 -0.2535 -0.1958 0.1902 0.0582 -0.2503 0.2012  
 0.2682 -0.0259 0.1196 -0.3696 -0.0300 -0.2319 0.2032 0.2411 -0.2065 0.1875 -0.3018 0.1238 -0.1097 0.0157 0.0741 0.4531 0.1208 0.0096 -0.1672 -0.1916 -0.0779 -0.0178 0.2864 0.0845 0.2003

for i = 1:m

U1 = Y' \* (eigvec(:,i))/ sqrt(eigval(i,i))

U = [U,U1]

end

U =

0.0313 0.0029 0.0460 0.1014 -0.0581 -0.0210 -0.0367 0.0380 -0.1281 -0.0045 0.0284 -0.0244 -0.0552 0.0988 0.0104 -0.1304 -0.0594 0.1485 0.0946 -0.2366 -0.0225 -0.0703 -0.0043 -0.0266 0.1151  
 -0.0175 0.0500 -0.0962 0.0595 -0.0130 0.0687 0.0354 0.1018 -0.0492 0.0105 0.1017 -0.0210 -0.1307 -0.0187 -0.0059 0.0203 0.1824 -0.0205 0.0885 0.1657 -0.1717 0.0933 0.0091 0.0384 0.1147  
 0.0010 -0.0261 -0.0615 -0.0778 -0.0521 -0.0733 0.1093 0.0860 0.0688 0.1321 -0.0423 -0.0622 0.0400 -0.1235 0.0341 0.1133 -0.0312 0.0971 -0.2400 -0.1143 0.0284 0.0784 0.0140 -0.0317 0.1150  
 0.0403 -0.0081 0.0442 -0.0860 0.0569 -0.1177 0.1010 -0.0551 -0.0500 -0.1219 0.0588 -0.0288 0.1081 0.0274 -0.0785 -0.0684 0.0173 -0.0404 0.2612 -0.1271 0.0581 0.1169 0.0081 0.0280 0.1145  
 -0.1261 -0.1501 0.0683 0.0160 -0.1486 0.0400 -0.0135 -0.1032 -0.0951 0.1098 0.0593 0.0127 0.0376 0.0844 0.1045 0.0547 -0.0174 0.0990 -0.0552 0.2658 -0.0162 0.0772 -0.0330 -0.0550 0.1145  
 0.1744 -0.0331 0.0124 0.0702 -0.0224 0.1414 0.0246 0.0430 0.0843 -0.0993 0.1355 0.1205 -0.0280 -0.0963 -0.0773 -0.0401 0.0625 0.1383 -0.0819 -0.1845 0.1463 -0.0643 -0.0526 0.0421 0.1144  
 -0.1774 -0.0169 -0.0631 0.0403 0.0271 0.0178 0.0475 -0.0393 0.0515 0.1936 -0.1401 0.0924 0.0699 -0.0053 0.0286 0.1506 0.0827 -0.0213 0.2523 0.0654 -0.0582 -0.0296 0.0649 0.0409 0.1148  
 0.2016 0.0837 0.0262 -0.0387 -0.0531 -0.0794 0.0203 -0.0140 0.0852 -0.2234 -0.0563 0.0936 0.0922 -0.0467 0.0384 -0.0620 -0.0927 0.1764 -0.1553 0.0256 -0.1530 0.0154 0.0171 -0.0707 0.1141  
 -0.2104 -0.0445 -0.0404 -0.0098 0.0566 0.0270 -0.0196 -0.1077 -0.1124 0.0375 -0.1323 0.0718 -0.0647 0.1532 -0.0932 -0.1015 0.2276 -0.0827 0.0058 -0.1493 0.0296 0.0800 -0.0456 0.0261 0.1141  
 0.1459 -0.0323 0.0453 -0.0269 -0.1294 0.1224 -0.0438 0.0294 0.0518 -0.0921 -0.0332 -0.0060 -0.0668 -0.1092 0.2143 0.2048 -0.0687 -0.0941 0.0402 0.0568 -0.0339 0.1130 0.0133 0.0188 0.1131

%PCA Analysis Code%

p = sum(eigen)

p = 1.2048e+04

a = eigen

a =

2.4741  
 7.4789  
 7.7391  
 7.8573  
 8.7943  
 9.2656  
 9.4269  
 9.5394  
 9.6925  
 10.3214

prompt = 'Enter the beta value'

prompt = 'Enter the beta value'

beta = double(input(prompt))

beta = 0.9500

co = 0

co = 0

estimated = 1.00

estimated = 1

while estimated > beta

[a , estimated] = function23(a, p)

co = co + 1

if estimated <= beta

disp(a)

break

end

end

a =

0.0033  
 1.1702

newa =

0.0033  
 1.1702

newsum = 1.1735e+04

y = 0.9740

a =

0.0033  
 1.1702

estimated = 0.9740

co = 23

index = 1

a = 1.1702e+04

newa = 1.1702e+04

newsum = 1.1702e+04

y = 0.9713

a = 1.1702e+04

estimated = 0.9713

co = 24

co = 25

for i= 1:23

%index = find(eigen == a(i))

p1 = sqrt(eigval(i,i))\*U(:,i)\*eigvec(:,i)'

PNoise = PNoise + p1

end

PNoise =

-0.1361 0.0536 -0.3302 0.6424 -0.1613 -0.1370 0.1607 -0.2299 0.6263 -0.0943 -0.0394 0.7242 -0.1548 -0.1729 -0.1880 0.2657 0.0988 0.0423 0.2536 -0.4578 0.0324 -0.5101 -0.0376 0.3010 -0.5654  
 0.0618 -0.2707 0.6292 -0.2024 -0.0877 0.1757 -0.2763 0.6630 -0.0558 -0.0927 0.7136 -0.0997 -0.1929 -0.2212 0.3556 0.1029 -0.0253 0.2955 -0.4117 -0.0184 -0.4899 0.0372 0.2582 -0.5858 -0.2701  
 -0.2991 0.5658 -0.1501 -0.0841 0.1136 -0.2598 0.6964 -0.1074 -0.1368 0.7673 -0.1154 -0.2663 -0.1804 0.3681 0.0424 -0.0188 0.3534 -0.4797 -0.0517 -0.4206 -0.0044 0.2128 -0.5238 -0.2683 0.2364  
 0.5923 -0.1022 -0.1052 0.0923 -0.2118 0.6991 -0.1422 -0.0885 0.7901 -0.1690 -0.2548 -0.1356 0.3381 0.0303 0.0716 0.3370 -0.5304 0.0048 -0.3974 -0.0485 0.2540 -0.4679 -0.3139 0.2383 -0.4973  
 -0.0875 -0.1915 0.1316 -0.1608 0.6368 -0.1489 -0.0255 0.7663 -0.2448 -0.1969 -0.0996 0.2446 0.0537 0.1342 0.2796 -0.5650 0.1047 -0.4311 -0.1287 0.3429 -0.4609 -0.3980 0.2989 -0.4431 0.5862  
 -0.1542 0.1959 -0.2205 0.6027 -0.0807 -0.0449 0.7021 -0.1745 -0.1729 -0.1991 0.2600 0.1177 0.0658 0.2554 -0.4413 0.0603 -0.5212 -0.0414 0.3617 -0.5498 -0.3363 0.3654 -0.5356 0.5840 -0.1137  
 0.0923 -0.1875 0.6736 -0.1862 -0.0642 0.7785 -0.2212 -0.2508 -0.1393 0.2819 -0.0019 0.0844 0.3258 -0.5443 0.0462 -0.4150 -0.1047 0.2595 -0.4462 -0.3371 0.2434 -0.4504 0.6272 -0.2257 0.1534  
 -0.1617 0.5771 -0.1176 0.0150 0.7095 -0.2123 -0.1534 -0.1636 0.2038 0.0947 0.1383 0.2210 -0.4930 0.1370 -0.4870 -0.1246 0.4037 -0.4848 -0.4199 0.3744 -0.4375 0.5326 -0.1273 0.2312 -0.2693  
 0.6486 -0.0743 -0.0731 0.7065 -0.1445 -0.2049 -0.2167 0.2984 0.0858 0.0233 0.2760 -0.4467 0.0367 -0.4776 0.0055 0.3099 -0.5560 -0.3002 0.3455 -0.5283 0.6362 -0.0969 0.1180 -0.2335 -0.1579  
 -0.1053 -0.0370 0.7796 -0.1776 -0.1961 -0.1456 0.3015 0.0665 0.0735 0.3169 -0.4862 0.0617 -0.4100 -0.0225 0.3300 -0.4757 -0.3029 0.3103 -0.4522 0.6708 -0.1387 0.1877 -0.1778 -0.1907 0.2173

P = sqrt(eigval(25,25))\*U(:,25)\*eigvec(:,25)' + sqrt(eigval(24,24))\*U(:,24)\*eigvec(:,24)'

P =

2.4921 2.5534 2.5162 2.4816 2.5263 2.5250 2.4803 2.4969 2.5427 2.4803 2.4504 2.5188 2.4808 2.4299 2.4880 2.5223 2.4332 2.4657 2.5224 2.4398 2.4456 2.5221 2.4716 2.4410 2.5094  
 2.5452 2.4567 2.4948 2.5674 2.4757 2.4653 2.5433 2.5060 2.4418 2.5037 2.5294 2.4257 2.4499 2.5212 2.4324 2.4291 2.5333 2.4805 2.3937 2.4964 2.5019 2.3968 2.4838 2.5298 2.4331  
 2.4851 2.5582 2.5151 2.4721 2.5274 2.5268 2.4726 2.4934 2.5478 2.4757 2.4414 2.5233 2.4804 2.4199 2.4896 2.5268 2.4226 2.4617 2.5297 2.4326 2.4384 2.5292 2.4678 2.4313 2.5126  
 2.5317 2.4672 2.4932 2.5487 2.4788 2.4699 2.5282 2.4995 2.4529 2.4950 2.5118 2.4356 2.4499 2.5017 2.4364 2.4390 2.5124 2.4732 2.4094 2.4825 2.4880 2.4119 2.4769 2.5107 2.4403  
 2.4525 2.5795 2.5094 2.4278 2.5322 2.5349 2.4365 2.4767 2.5708 2.4539 2.3996 2.5434 2.4783 2.3738 2.4964 2.5470 2.3733 2.4431 2.5627 2.3991 2.4049 2.5610 2.4501 2.3861 2.5268  
 2.5422 2.4451 2.4875 2.5663 2.4667 2.4559 2.5409 2.5005 2.4299 2.4991 2.5280 2.4143 2.4422 2.5206 2.4233 2.4177 2.5332 2.4754 2.3803 2.4938 2.4993 2.3836 2.4786 2.5290 2.4227  
 2.5487 2.4545 2.4954 2.5722 2.4752 2.4645 2.5472 2.5078 2.4393 2.5061 2.5339 2.4236 2.4502 2.5263 2.4318 2.4270 2.5387 2.4825 2.3902 2.5001 2.5056 2.3934 2.4858 2.5347 2.4316  
 2.4287 2.5919 2.5036 2.3960 2.5335 2.5383 2.4104 2.4636 2.5842 2.4373 2.3697 2.5550 2.4750 2.3410 2.4990 2.5586 2.3383 2.4288 2.5829 2.3746 2.3805 2.5804 2.4363 2.3538 2.5343  
 2.5204 2.4603 2.4841 2.5365 2.4705 2.4619 2.5167 2.4896 2.4462 2.4847 2.5000 2.4287 2.4413 2.4896 2.4285 2.4321 2.5000 2.4632 2.4035 2.4713 2.4768 2.4059 2.4670 2.4985 2.4329  
 2.4913 2.4480 2.4634 2.5036 2.4531 2.4456 2.4865 2.4655 2.4345 2.4591 2.4682 2.4163 2.4220 2.4565 2.4120 2.4197 2.4659 2.4387 2.3952 2.4422 2.4477 2.3973 2.4428 2.4657 2.4187

%Actual Matrix = P after removing the noise. Noise of the matrix is PNOISE

%computed with beta = 0.05

%Problem 2.c Identifying Noise Component Based on PCA Analysis by taking Beta value as 0.05%

for i = 1:m

newm = newm + eigval(i,i)\*U(:,i)\*eigvec(:,i)'

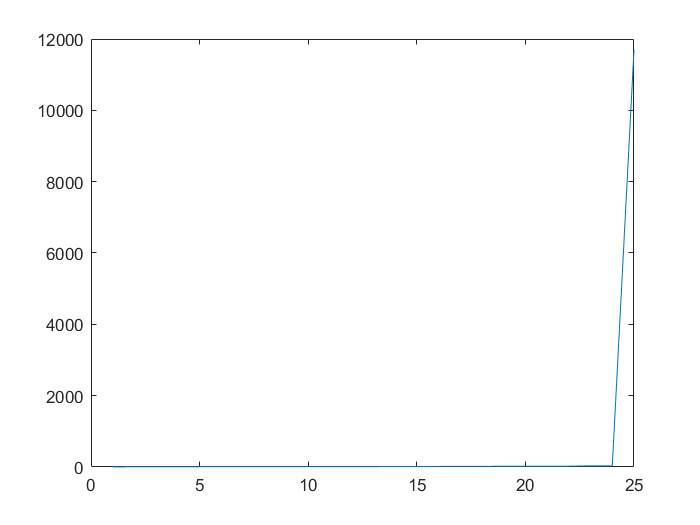
end

newm =

271.5466 272.8527 270.2313 275.1608 270.9776 270.7420 272.0955 269.2637 273.7580 269.3302 268.5764 271.7016 266.5897 266.2807 266.4869 270.3366 268.0361 267.9102 268.8035 264.4777 267.5950 265.8684 268.0003 269.5542 266.1761  
 271.9498 270.0729 273.4946 271.2918 270.4648 270.6843 269.0970 272.8300 269.7071 268.6386 271.1119 266.9299 265.6103 265.9316 267.4020 268.1779 267.3995 268.0735 264.2351 266.2861 264.7966 266.4503 268.6711 265.2319 266.2964  
 270.7773 274.7430 270.9026 272.1647 271.7927 269.1887 273.9147 270.2888 270.3618 272.3284 267.7300 267.5804 266.6019 268.1859 267.0527 269.0468 268.8849 265.1696 267.6743 264.7978 266.8629 268.8657 265.7561 266.8686 268.9616  
 273.6112 270.5803 270.4287 272.0842 268.7085 272.3616 270.0887 269.2070 272.7047 267.6557 266.8266 266.8895 267.3845 266.2982 265.9507 268.8557 264.5811 266.6949 264.4149 265.3590 267.6297 263.8406 265.8500 267.9735 264.8681  
 270.6520 271.0451 270.6410 269.6640 272.8186 269.3432 269.3391 272.6924 268.6180 267.3038 267.0066 268.6924 266.1587 266.0914 266.9061 265.7213 266.6675 264.8738 266.0932 266.8657 263.5979 265.6241 267.7537 264.5260 269.5432  
 270.7853 271.0797 268.6432 273.8608 269.6066 268.6193 273.1598 268.2895 268.4137 267.5369 268.7020 267.3155 266.0143 266.8731 263.6824 267.4492 265.0021 266.0226 267.3633 262.8140 265.2230 266.8026 264.2848 269.3874 266.4823  
 271.9888 270.1533 273.7375 271.5181 270.0644 273.4138 269.7527 269.1999 269.4902 270.3525 268.3481 268.1672 267.8549 264.7631 266.1419 266.8426 267.1102 268.4237 264.3823 265.4962 267.3142 264.5580 270.1017 267.2284 267.6373  
 268.5111 272.8128 269.0595 269.7128 272.2694 267.6564 268.1495 267.8340 270.0571 267.4560 266.8403 267.5560 263.2620 264.7687 263.5963 266.5782 267.1414 263.0179 264.5820 265.1964 262.9002 267.9247 265.6564 266.0310 265.4072  
 272.7936 269.7527 268.8812 273.5155 268.1948 267.6260 268.5304 269.9785 268.7586 267.6734 267.7980 264.6137 264.9314 263.8370 264.6092 268.1697 263.3653 264.7475 265.8981 262.3698 267.7149 264.9504 266.0010 265.4142 265.3597  
 267.5048 267.3322 269.9526 267.0355 265.8758 265.5625 267.9589 266.5257 266.5477 266.2090 262.5978 263.9354 261.5325 262.6252 263.9662 262.3285 262.6005 264.0763 260.7933 264.8450 262.5914 263.4513 263.0687 262.7772 264.2722

plot(eigen)

Scree Plot



%Problem 2.d identifying different Components%

trendcomponent = eigval(25,25)\*U(:,25)\*eigvec(:,25)' + eigval(24,24)\*U(:,24)\*eigvec(:,24)'

trendcomponent =

272.1474 272.5039 271.6328 272.3882 271.4971 270.9763 271.2884 270.8203 271.1660 269.6233 268.7180 268.9003 267.3965 267.0156 267.1435 269.2747 267.7438 267.6779 267.7977 266.6334 267.2472 267.9086 268.2101 268.1141 268.5879  
 271.6259 271.1257 270.6870 272.0526 270.3852 269.8146 270.8257 270.0513 269.7674 268.9399 268.3550 267.5538 266.4097 266.7283 266.0165 267.9264 267.5044 266.9512 266.2514 266.1485 266.7591 266.3807 267.4673 267.8084 267.3380  
 271.8829 272.3065 271.4023 272.1087 271.2799 270.7636 271.0203 270.5768 270.9718 269.3743 268.4448 268.7044 267.1740 266.7384 266.9324 269.0786 267.4620 267.4346 267.6189 266.3722 266.9855 267.7281 267.9675 267.8371 268.3848  
 271.1678 270.8043 270.2980 271.5641 270.0231 269.4615 270.3599 269.6355 269.4520 268.5128 267.8783 267.2345 266.0357 266.2428 265.6655 267.6067 267.0098 266.5349 265.9664 265.6955 266.3054 266.0925 267.0525 267.3237 267.0038  
 270.6124 271.3443 270.2882 270.7697 270.2266 269.7312 269.7331 269.4031 270.0240 268.1760 267.1353 267.7500 266.0975 265.4106 265.9079 268.1228 266.1130 266.2623 266.7427 265.1180 265.7292 266.8447 266.7982 266.5102 267.3971  
 270.9425 270.3925 269.9806 271.3792 269.6696 269.0974 270.1478 269.3570 269.0364 268.2533 267.6889 266.8311 265.7111 266.0706 265.3106 267.2028 266.8477 266.2669 265.5200 265.4805 266.0894 265.6500 266.7807 267.1470 266.6216  
 271.7688 271.2353 270.8130 272.2029 270.5046 269.9318 270.9704 270.1837 269.8755 269.0750 268.5021 267.6625 266.5316 266.8773 266.1328 268.0353 267.6557 267.0834 266.3517 266.2896 266.9004 266.4817 267.5992 267.9573 267.4503  
 269.5485 270.4871 269.3297 269.6595 269.3092 268.8281 268.6587 268.4052 269.1777 267.1625 266.0479 266.9009 265.1675 264.3122 265.0131 267.2725 264.9998 265.2678 265.9470 264.0693 264.6786 266.0440 265.8054 265.4115 266.5259  
 270.1323 269.7936 269.2774 270.5221 269.0081 268.4501 269.3259 268.6127 268.4471 267.4920 266.8512 266.2367 265.0327 265.2201 264.6678 266.6075 265.9827 265.5230 264.9789 264.6802 265.2878 265.1041 266.0390 266.2972 266.0043  
 267.6823 267.4401 266.8816 268.0482 266.6329 266.0855 266.8770 266.2038 266.1083 265.0842 264.4144 263.9128 262.6808 262.7901 262.3345 264.2803 263.5405 263.1381 262.6884 262.2768 262.8791 262.8104 263.6511 263.8593 263.6720

Oscilatorycomponent = eigval(23,23)\*U(:,23)\*eigvec(:,23)' + eigval(22,22)\*U(:,22)\*eigvec(:,22)'+ eigval(21,21)\*U(:,21)\*eigvec(:,21)'

Oscilatorycomponent =

-0.5361 -0.4925 -0.3258 0.0309 0.2479 -0.0324 -0.3579 -0.5004 -0.2162 0.0262 0.2648 0.3084 -0.2247 -0.3892 -0.1048 0.1058 0.4052 0.5240 0.1457 -0.3059 -0.0878 0.1824 0.3983 0.6699 0.2876  
 -0.6139 0.9725 1.4585 0.0854 0.6559 -0.4059 -1.2812 0.9113 1.1465 0.3363 0.4679 -0.3823 -1.1845 0.0280 1.1229 -0.0726 0.5585 -0.5271 -1.7311 -0.0805 0.1145 0.3038 0.5668 -0.8941 -1.5936  
 0.5576 0.5138 0.4346 -0.0380 -0.3531 0.0930 0.4549 0.4832 0.2446 0.0510 -0.3537 -0.3904 0.3493 0.4314 0.0336 -0.0394 -0.4173 -0.6821 -0.1234 0.4257 0.0305 -0.2213 -0.3615 -0.7799 -0.3685  
 1.0187 0.7820 0.4485 -0.0645 -0.5212 0.1066 0.7787 0.7969 0.2738 -0.0699 -0.5303 -0.5218 0.5356 0.6957 0.0644 -0.1721 -0.7789 -0.9013 -0.0828 0.5689 0.1371 -0.3604 -0.7607 -1.1177 -0.3631  
 0.5769 0.7303 0.2019 -0.0023 0.1766 -0.2594 -0.0732 0.8920 0.3391 -0.3102 0.0603 -0.1435 -0.3600 0.3727 0.5914 -0.4316 -0.4404 -0.1494 -0.5283 -0.0783 0.3745 -0.0531 -0.6314 -0.6010 -0.2816  
 0.9138 -0.5649 -1.5811 -0.0692 -0.2914 0.0985 0.9951 -0.3128 -0.9955 -0.7088 -0.2156 0.4858 0.6566 0.1018 -0.5797 -0.3522 -0.7969 0.8069 1.3383 -0.2533 0.2537 -0.2416 -1.0392 0.7781 1.5998  
 -0.9880 -0.3452 0.7353 0.0333 -0.0849 0.2280 -0.3866 -0.6535 0.2411 0.7132 -0.0076 -0.1890 0.0722 -0.3587 -0.2817 0.6262 0.8100 -0.4108 -0.2132 0.2950 -0.5156 0.1464 1.1474 0.0487 -0.6522  
 -1.0113 0.3138 1.0033 0.0902 0.6297 -0.2729 -1.2588 0.2005 0.7463 0.4104 0.5189 -0.1131 -1.0048 -0.3131 0.6964 0.1457 0.8467 -0.1207 -1.2168 -0.2032 -0.0803 0.3749 0.9172 -0.1984 -1.1110  
 0.9747 0.7181 -0.1282 -0.0298 0.0253 -0.2221 0.2981 0.9488 0.1333 -0.5061 -0.0749 -0.1022 -0.1166 0.5112 0.4632 -0.5544 -0.7696 -0.0849 -0.2090 -0.0567 0.4660 -0.1707 -1.0224 -0.5841 0.0651  
 0.3659 0.8851 0.9295 -0.0066 -0.0900 -0.0761 -0.0201 0.8501 0.6559 0.1353 -0.1659 -0.5120 -0.1125 0.4495 0.4726 -0.1017 -0.2406 -0.8367 -0.7607 0.3654 0.1025 -0.1084 -0.2028 -1.0965 -0.9212

for i = 1:20

Noisecomponent = Noisecomponent + eigval(i,i)\*U(:,i)\*eigvec(:,i)'

end

Noisecomponent =

-0.0647 0.8413 -1.0757 2.7417 -0.7674 -0.2019 1.1650 -1.0563 2.8082 -0.3193 -0.4064 2.4929 -0.5821 -0.3456 -0.5517 0.9560 -0.1129 -0.2917 0.8601 -1.8498 0.4356 -2.2225 -0.6081 0.7702 -2.6993  
 0.9378 -2.0252 1.3491 -0.8462 -0.5762 1.2756 -0.4475 1.8674 -1.2069 -0.6376 2.2890 -0.2416 0.3851 -0.8247 0.2626 0.3241 -0.6634 1.6494 -0.2852 0.2181 -2.0769 -0.2342 0.6371 -1.6823 0.5520  
 -1.6632 1.9227 -0.9343 0.0939 0.8659 -1.6678 2.4395 -0.7712 -0.8546 2.9030 -0.3611 -0.7335 -0.9215 1.0162 0.0868 0.0076 1.8402 -1.5828 0.1788 -2.0000 -0.1532 1.3588 -1.8498 -0.1886 0.9453  
 1.4248 -1.0060 -0.3178 0.5846 -0.7933 2.7935 -1.0498 -1.2255 2.9789 -0.7872 -0.5214 0.1768 0.8132 -0.6403 0.2208 1.4210 -1.6498 1.0613 -1.4687 -0.9053 1.1873 -1.8915 -0.4418 1.7675 -1.7726  
 -0.5372 -1.0294 0.1509 -1.1033 2.4154 -0.1285 -0.3208 2.3973 -1.7452 -0.5620 -0.1890 1.0859 0.4212 0.3081 0.4068 -1.9699 0.9949 -1.2390 -0.1212 1.8260 -2.5058 -1.1675 1.5869 -1.3832 2.4277  
 -1.0709 1.2520 0.2437 2.5508 0.2283 -0.5766 2.0170 -0.7546 0.3729 -0.0076 1.2287 -0.0014 -0.3534 0.7007 -1.0485 0.5987 -1.0486 -1.0512 0.5051 -2.4131 -1.1201 1.3942 -1.4567 1.4623 -1.7391  
 1.2080 -0.7369 2.1891 -0.7180 -0.3553 3.2540 -0.8311 -0.3304 -0.6265 0.5643 -0.1464 0.6937 1.2511 -1.7556 0.2908 -1.8189 -1.3554 1.7510 -1.7561 -1.0883 0.9295 -2.0701 1.3551 -0.7777 0.8393  
 -0.0261 2.0119 -1.2735 -0.0369 2.3305 -0.8988 0.7497 -0.7716 0.1331 -0.1169 0.2735 0.7683 -0.9007 0.7696 -2.1132 -0.8400 1.2949 -2.1292 -0.1482 1.3303 -1.6982 1.5058 -1.0662 0.8180 -0.0077  
 1.6867 -0.7590 -0.2680 3.0233 -0.8386 -0.6019 -1.0936 0.4169 0.1782 0.6875 1.0217 -1.5208 0.0153 -1.8943 -0.5218 2.1166 -1.8479 -0.6906 1.1282 -2.2537 1.9612 0.0170 0.9844 -0.2990 -0.7097  
 -0.5434 -0.9930 2.1415 -1.0062 -0.6672 -0.4469 1.1020 -0.5282 -0.2165 0.9895 -1.6507 0.5346 -1.0358 -0.6144 1.1590 -1.8501 -0.6994 1.7749 -1.1344 2.2028 -0.3903 0.7493 -0.3797 0.0144 1.5213

function [a] = rmvfunction(a,leas)

index = find(a==leas)

a(index) = []

end

function [newa ,y] = function23(a, p)

newa = rmvfunction(a, min(a))

newsum = sum(newa)

y = newsum/p

end