Homework 3

Math 3607, Summer 2021

Spenser Smith

Table of Contents

Problem 1. 1
Problem 2. 2
Problem 3. 6
Problem 4. 7

Problem 1.

This problem asks for hand-written solutions to a variety of problems that mostly deal with matrix size.

a) i-th row : $[r_i w_1 \;\; r_i w_2 \;\; \cdots \;\; r_i w_n]$

size : $n \times n$ matrix

b) j-th element : $\sum_i b_j^T [A]_{ij} = \sum_i b_i a_{ij}$

Size: vector of length m

c) if $C = AB$, $c_{ij} = \sum_k a_{ik} b_{kj}$

size: $m \times p$ matrix

d) if $D = B^T A^T$, $d_{ij} = \sum_k b_{jk} a_{ki}$

size: $p \times m$ matrix

e) since we know $(B^T A^T) = (AB)^T$

we have $d_{ij} = \sum_k b_{jk} a_{ki}$

and $c_{ij} = \sum_k a_{ik} b_{kj}$

Thus $c_{ji} = c_{ij}^T = \sum_k (a_{ik} b_{kj})^T = \sum_k b_{jk} a_{ki} = d_{ij}$

f) $A^T A = \sum_i a_{ji} a_{ij}$

size: $n \times n$

g) $AA^T = \sum_i a_{ij} a_{ji}$

size: $m \times m$

h) $\sum_i a_{ij} a_{ji} b_j$

size: vector with length m

Problem 2.

This problem asks for a script that uses the Gram-Schmidt method to factor a matrix for large values of m and n. Then, check how accurate it is compared to the built-in MATLAB function qr.

```
m = input('Choose the number of rows: ');
n = input('Choose the number of columns: ');
A = randi(100,[m,n]);

[Q, R] = qr(A); %calling each function
[S, T] = gs(A);

ans1a = (Q'*Q) - eye(m); %check if each function gives zero matrix
fprintf('The QR approx for equation 1 is: \n')
```

```
The QR approx for equation 1 is:
```

```
disp(ans1a)
```

```
  1.0e-15 *
 Columns 1 through 18
  -0.1110   -0.0075    0.0305   -0.0112   -0.0271   -0.0102    0.0186    0.0055   -0.0322    0.0103   -0.0135    0
  -0.0075   -0.4441   -0.0654   -0.0751   -0.0386    0.1148    0.0400    0.0806   -0.0080   -0.0037    0.0587   -0
   0.0305   -0.0654         0   -0.0205    0.0908   -0.0106   -0.0653   -0.0455    0.0018    0.0373    0.1064   -0
  -0.0112   -0.0751   -0.0205    0.4441    0.0446   -0.0235   -0.0637   -0.0790   -0.0144    0.0410    0.1051    0
  -0.0271   -0.0386    0.0908    0.0446         0   -0.0357    0.0099   -0.0025   -0.0659   -0.0986    0.1366    0
  -0.0102    0.1148   -0.0106   -0.0235   -0.0357    0.4441   -0.1027   -0.0137    0.0427   -0.0799    0.0591    0
   0.0186    0.0400   -0.0653   -0.0637    0.0099   -0.1027    0.2220    0.0636   -0.0921    0.0612   -0.0356   -0
   0.0055    0.0806   -0.0455   -0.0790   -0.0025   -0.0137    0.0636    0.4441   -0.0430    0.0332   -0.0414   -0
  -0.0322   -0.0080    0.0018   -0.0144   -0.0659    0.0427   -0.0921   -0.0430    0.2220   -0.0451    0.0824    0
   0.0103   -0.0037    0.0373    0.0410   -0.0986   -0.0799    0.0612    0.0332   -0.0451    0.4441   -0.0521    0
  -0.0135    0.0587    0.1064    0.1051    0.1366    0.0591   -0.0356   -0.0414    0.0824   -0.0521   -0.2220    0
   0.0786   -0.0747   -0.0901    0.0899    0.0643    0.0641   -0.0351   -0.0192    0.1275    0.0064    0.0836    0
  -0.0409    0.0002    0.0736   -0.0432   -0.0282    0.0505   -0.1143    0.0903   -0.0141    0.0308    0.0022   -0
```

```
    0.0177    0.0445    0.0037    0.0350    0.1132    0.0198    0.0811   -0.0245   -0.0126   -0.0629   -0.0249   -0
    0.0400    0.0305    0.0619    0.0023    0.0419   -0.0323   -0.1868   -0.0866    0.0164    0.0383    0.0102   -0
    0.0068   -0.0050    0.0253    0.0283    0.1386    0.0256   -0.0381   -0.0231   -0.0680   -0.0193    0.0240   -0
   -0.0137    0.0189   -0.0224    0.0073   -0.0340    0.0875    0.0792   -0.0291    0.0370    0.0319   -0.0003   -0
    0.0195   -0.1583   -0.0537    0.0146    0.0054    0.1313   -0.0055    0.0276    0.0546    0.0258    0.0598   -0
    0.0047   -0.0730    0.0248   -0.1226    0.0018   -0.0232    0.0019   -0.0479   -0.0711   -0.0777   -0.0066   -0
    0.0010    0.0828    0.0669   -0.0481   -0.0238    0.0321    0.0644    0.0178   -0.0409    0.0196    0.0335   -0
    0.0427    0.1166    0.0768   -0.0403   -0.0590   -0.0584   -0.0590   -0.0132    0.1291    0.0636    0.1074   -0
   -0.0082    0.0588   -0.0293   -0.0824   -0.0169   -0.0522    0.0230    0.0143    0.1796   -0.0016   -0.0969    0
   -0.0085    0.1494    0.0771    0.0302    0.0097   -0.0651    0.0467   -0.0553   -0.0549    0.0167    0.0601    0
    0.0258   -0.0562   -0.0438    0.0530    0.0578   -0.0185   -0.0630    0.0149    0.0211   -0.0550   -0.0730   -0
   -0.0340    0.0571    0.0157   -0.0099   -0.0516    0.0233   -0.0564   -0.0034   -0.0925   -0.0285   -0.0190   -0
   -0.0052   -0.0628   -0.0546   -0.0083   -0.1007    0.0342    0.0228   -0.0297   -0.1487   -0.0448   -0.0881   -0
   -0.0392    0.0146   -0.0518   -0.0530   -0.1095    0.0560   -0.0437    0.0147   -0.0159   -0.1581    0.1681    0
   -0.0255    0.0426    0.0538   -0.1093   -0.0512   -0.0345   -0.0660    0.0189    0.1018   -0.0323    0.1069    0
   -0.0597   -0.1798   -0.1274    0.0341   -0.0390    0.0195    0.0868   -0.0997    0.0533    0.0472   -0.0709    0
    0.0490   -0.0043    0.0915    0.0422    0.0018   -0.0183   -0.0270   -0.0216    0.1186   -0.0176    0.1051   -0
  Columns 19 through 30
    0.0047    0.0010    0.0427   -0.0082   -0.0085    0.0258   -0.0340   -0.0052   -0.0392   -0.0255   -0.0597    0
   -0.0730    0.0828    0.1166    0.0588    0.1494   -0.0562    0.0571   -0.0628    0.0146    0.0426   -0.1798   -0
    0.0248    0.0669    0.0768   -0.0293    0.0771   -0.0438    0.0157   -0.0546   -0.0518    0.0538   -0.1274    0
   -0.1226   -0.0481   -0.0403   -0.0824    0.0302    0.0530   -0.0099   -0.0083   -0.0530   -0.1093    0.0341    0
    0.0018   -0.0238   -0.0590   -0.0169    0.0097    0.0578   -0.0516   -0.1007   -0.1095   -0.0512   -0.0390    0
   -0.0232    0.0321   -0.0584   -0.0522   -0.0651   -0.0185    0.0233    0.0342    0.0560   -0.0345    0.0195   -0
    0.0019    0.0644   -0.0590    0.0230    0.0467   -0.0630   -0.0564    0.0228   -0.0437   -0.0660    0.0868   -0
   -0.0479    0.0178   -0.0132    0.0143   -0.0553    0.0149   -0.0034   -0.0297    0.0147    0.0189   -0.0997   -0
   -0.0711   -0.0409    0.1291    0.1796   -0.0549    0.0211   -0.0925   -0.1487   -0.0159    0.1018    0.0533    0
   -0.0777    0.0196    0.0636   -0.0016    0.0167   -0.0550   -0.0285   -0.0448   -0.1581   -0.0323    0.0472   -0
   -0.0066    0.0335    0.1074   -0.0969    0.0601   -0.0730   -0.0190   -0.0881    0.1681    0.1069   -0.0709    0
   -0.0283   -0.0472   -0.0227    0.0991    0.1182   -0.0351   -0.0625   -0.1035    0.0271    0.0057    0.0974   -0
    0.0522    0.1385   -0.0150   -0.1405    0.0427    0.1173   -0.0262   -0.0139    0.0919    0.0837    0.0189    0
    0.0078    0.0662   -0.1249   -0.0737    0.0888    0.1265   -0.0738   -0.0186    0.0166    0.1461    0.0848    0
   -0.0254   -0.0515    0.0333    0.0167    0.0259   -0.0566    0.0842   -0.1227   -0.1252   -0.1440    0.1112   -0
    0.1112    0.0077   -0.0515   -0.0632   -0.0543    0.0297   -0.0708   -0.0338    0.1459    0.0404   -0.0983    0
    0.0401    0.0106   -0.1132    0.1848    0.0677   -0.0102   -0.0072    0.0705    0.2520   -0.0786   -0.1148    0
   -0.0147    0.0321    0.0172    0.1390    0.0581   -0.0764   -0.1485    0.0067    0.0051   -0.0342   -0.0783    0
   -0.2220    0.0597   -0.0547    0.1876    0.0008    0.0293    0.0663   -0.0231    0.0224    0.0376   -0.0527   -0
    0.0597   -0.3331   -0.0828   -0.0986   -0.0862    0.0102    0.0311   -0.0279    0.0587   -0.2030   -0.1094    0
   -0.0547   -0.0828    0.4441    0.0635   -0.1135    0.1172   -0.1404   -0.0495   -0.1482    0.0256    0.0892    0
    0.1876   -0.0986    0.0635         0   -0.1059    0.0103   -0.1190   -0.0399   -0.0048   -0.0211    0.1397   -0
    0.0008   -0.0862   -0.1135   -0.1059    0.2220    0.0812   -0.1194   -0.0202   -0.0638    0.1477    0.0464   -0
    0.0293    0.0102    0.1172    0.0103    0.0812         0    0.0143   -0.0306    0.0633   -0.0450   -0.0633    0
    0.0663    0.0311   -0.1404   -0.1190   -0.1194    0.0143   -0.2220    0.0949    0.0208    0.0756    0.0271    0
   -0.0231   -0.0279   -0.0495   -0.0399   -0.0202   -0.0306    0.0949         0    0.0117    0.0112    0.0118   -0
    0.0224    0.0587   -0.1482   -0.0048   -0.0638    0.0633    0.0208    0.0117    0.2220   -0.1688    0.0528   -0
    0.0376   -0.2030    0.0256   -0.0211    0.1477   -0.0450    0.0756    0.0112   -0.1688   -0.5551   -0.0177   -0
   -0.0527   -0.1094    0.0892    0.1397    0.0464   -0.0633    0.0271    0.0118    0.0528   -0.0177   -0.4441   -0
   -0.0204    0.0103    0.0035   -0.0129   -0.0297    0.0306    0.0680   -0.0537   -0.1389   -0.1569   -0.0015    0
```

```
ans1b = (S'*S) - eye(n);
fprintf('The Gram-Schmidt approx for equation 1 is: \n')
```

```
The Gram-Schmidt approx for equation 1 is:
```

```
disp(ans1b)
```

```
  Columns 1 through 18
   -0.0000    0.0000    0.5471   -0.2597    0.8587   -0.7470    0.8639   -0.7774    0.8299   -0.8538    0.8469   -0
    0.0000    0.0000   -0.0000   -0.5216    0.2806   -0.2580    0.2722   -0.2940    0.3611   -0.1726    0.3784   -0
    0.5471   -0.0000         0   -0.5080    0.6475   -0.6136    0.6667   -0.5323    0.6494   -0.7260    0.6030   -0
   -0.2597   -0.5216   -0.5080   -0.0000   -0.3854    0.1266   -0.3284    0.1492   -0.4201    0.3767   -0.4916    0
    0.8587    0.2806    0.6475   -0.3854   -0.0000   -0.8407    0.9687   -0.8773    0.9524   -0.9233    0.9463   -0
```

```
   -0.7470   -0.2580   -0.6136    0.1266   -0.8407         0   -0.8553    0.8092   -0.8314    0.7904   -0.7574    0
    0.8639    0.2722    0.6667   -0.3284    0.9687   -0.8553   -0.0000   -0.9154    0.9469   -0.9375    0.9402   -0
   -0.7774   -0.2940   -0.5323    0.1492   -0.8773    0.8092   -0.9154   -0.0000   -0.8206    0.7922   -0.8195    0
    0.8299    0.3611    0.6494   -0.4201    0.9524   -0.8314    0.9469   -0.8206         0   -0.9362    0.9490   -0
   -0.8538   -0.1726   -0.7260    0.3767   -0.9233    0.7904   -0.9375    0.7922   -0.9362         0   -0.8871    0
    0.8469    0.3784    0.6030   -0.4916    0.9463   -0.7574    0.9402   -0.8195    0.9490   -0.8871   -0.0000   -0
   -0.7870   -0.3931   -0.5812    0.4376   -0.8853    0.7413   -0.9105    0.8195   -0.9028    0.8176   -0.9296    0
    0.8404    0.3128    0.5992   -0.4299    0.9238   -0.7491    0.9133   -0.7881    0.9490   -0.8808    0.9566   -0
   -0.7145   -0.2851   -0.5960    0.3898   -0.8571    0.7311   -0.7945    0.6736   -0.8696    0.8059   -0.8411    0
    0.8445    0.4074    0.4983   -0.4489    0.8989   -0.7379    0.9089   -0.7759    0.9313   -0.8532    0.9524   -0
   -0.8022   -0.3360   -0.5193    0.4222   -0.8517    0.6930   -0.8683    0.7350   -0.8469    0.8023   -0.9128    0
    0.8281    0.3864    0.5298   -0.4056    0.8769   -0.7469    0.8885   -0.7766    0.9292   -0.8537    0.9180   -0
   -0.7697   -0.3012   -0.4451    0.3659   -0.7887    0.6535   -0.8019    0.6436   -0.8680    0.8090   -0.8349    0
    0.8153    0.3577    0.6034   -0.4268    0.9101   -0.7343    0.9092   -0.8150    0.9540   -0.8791    0.9393   -0
   -0.7284   -0.3952   -0.5327    0.3642   -0.8302    0.6855   -0.8579    0.7670   -0.8948    0.7971   -0.8897    0
  Columns 19 through 20
    0.8153   -0.7284
    0.3577   -0.3952
    0.6034   -0.5327
   -0.4268    0.3642
    0.9101   -0.8302
   -0.7343    0.6855
    0.9092   -0.8579
   -0.8150    0.7670
    0.9540   -0.8948
   -0.8791    0.7971
    0.9393   -0.8897
   -0.8733    0.8429
    0.9465   -0.8857
   -0.8560    0.7538
    0.9147   -0.8798
   -0.8240    0.8150
    0.9424   -0.9207
   -0.8397    0.8072
         0   -0.9548
   -0.9548         0
```

```
ans2a = (Q*R) - A;
fprintf('The QR approx for equation 2 is: \n')
```

```
The QR approx for equation 2 is:
```

```
disp(ans2a)
```

```
   1.0e-13 *
  Columns 1 through 18
    0.2842   -0.2309    0.2132    0.3553   -0.1243    0.3553    0.4263    0.2842    0.1421    0.6395   -0.0355    0
    0.0711    0.1421   -0.2842   -0.2487   -0.0711   -0.2842   -0.2809   -0.2132   -0.5329   -0.4263   -0.1776   -0
    0.0044         0         0    0.0711    0.2842    0.2842    0.0355   -0.0711   -0.0711   -0.1421    0.4263   -0
    0.1421   -0.0355    0.0622    0.0355    0.4263    0.1421         0   -0.1421   -0.0888   -0.1243    0.4263    0
    0.1421         0    0.0178   -0.0178   -0.1421   -0.1421   -0.1421   -0.2842   -0.0711   -0.2842         0    0
    0.1421   -0.0711         0         0   -0.0711    0.3553    0.1421   -0.1421         0   -0.3197         0    0
    0.0089         0         0   -0.0178    0.1421    0.0355    0.0711    0.2842   -0.0711    0.1421    0.1421
    0.0711         0   -0.1421   -0.0711   -0.1421   -0.1421   -0.1243         0   -0.2842   -0.1421   -0.2132    0
    0.0355   -0.0711         0         0    0.0711    0.1421         0         0   -0.0888   -0.1421         0    0
    0.0711         0   -0.1066         0   -0.1066   -0.1066    0.0711         0         0         0   -0.2132   -0
    0.1421         0    0.1421    0.1421         0    0.1421    0.1421    0.1421   -0.0711   -0.0711         0    0
    0.1421   -0.1421    0.0711         0         0    0.0355   -0.0711    0.0711         0   -0.2842   -0.1421   -0
    0.0711   -0.1421   -0.0711   -0.1421         0   -0.1421   -0.1421    0.1421         0   -0.1421         0    0
    0.0711         0         0   -0.0711         0         0    0.0444   -0.1421   -0.1421   -0.3908   -0.1776   -0
    0.1421         0         0         0   -0.1243    0.1421    0.0355   -0.0711   -0.1421         0   -0.1421   -0
    0.0178         0    0.0711         0   -0.0711    0.0711    0.1421   -0.0089         0         0   -0.1776   -0
```

```
    0.0355         0         0         0         0         0   -0.0355         0   -0.2132    0.2842    0.0355    0.
    0.0355         0   -0.0089   -0.0266         0   -0.0866         0   -0.1421   -0.1421   -0.2132    0.1421   -0.
    0.1421   -0.0711   -0.0355    0.1421   -0.0355         0    0.0711         0    0.0355    0.1421   -0.2487   -0.
    0.0178         0   -0.2132   -0.1421   -0.0888   -0.1776         0         0   -0.1421         0   -0.1421   -0.
    0.0355   -0.0178         0         0         0         0         0    0.1421    0.0711   -0.1421   -0.0178    0.
    0.0711         0    0.1421    0.1421    0.0355    0.1421    0.1243    0.1421    0.1421    0.2842   -0.0711    0.
    0.0355         0   -0.0178    0.0711   -0.0711   -0.0711    0.0355   -0.0711         0    0.0711   -0.2487   -0.
    0.1421   -0.0711    0.0711    0.1066   -0.0711   -0.2842    0.0711    0.1421    0.0711    0.2132         0
    0.1421   -0.0711    0.1421    0.1421         0    0.0622    0.1421    0.1421    0.1421   -0.0711   -0.0555    0.
    0.0711   -0.0711   -0.1066   -0.0711         0   -0.1421    0.0711         0   -0.0711   -0.0711    0.1421    0.
    0.0089   -0.0711   -0.1421   -0.1421   -0.2842   -0.1066   -0.1421   -0.1421         0   -0.1421   -0.2132   -0.
    0.0711         0    0.1421    0.1421   -0.1421         0         0    0.1421   -0.1421    0.0711   -0.0711   -0.
    0.1421   -0.1421   -0.1599         0   -0.0711   -0.1421         0   -0.1044         0         0   -0.0711   -0.
    0.1421   -0.0711         0    0.1421   -0.0711    0.2132    0.1421    0.1421    0.1066    0.0888    0.1421    0.
  Columns 19 through 20
    0.2842    0.2842
   -0.5684   -0.5684
    0.1421    0.0711
    0.0711    0.4263
   -0.0355         0
   -0.0711         0
   -0.0888   -0.0711
   -0.2132   -0.5684
    0.1421         0
         0   -0.1954
    0.1421    0.1421
   -0.1421         0
   -0.4263   -0.2842
   -0.1421   -0.1421
         0    0.5329
   -0.2842         0
    0.1421   -0.1066
   -0.1421         0
         0         0
   -0.2132   -0.2842
         0   -0.0355
    0.2132         0
    0.1421   -0.2132
    0.2842         0
    0.1421         0
   -0.0711    0.0711
   -0.3553   -0.2842
         0    0.2842
   -0.0711   -0.1421
   -0.1421    0.2842
```

```
ans2b = (S*T) - A;
fprintf('The Gram-Schmidt approx for equation 2 is: \n')
```

```
The Gram-Schmidt approx for equation 2 is:
```

```
disp(ans2b)
```

```
   1.0e-12 *
  Columns 1 through 18
         0   -0.0018         0         0         0   -0.0071         0         0    0.0071   -0.0071    0.0018   -0
         0         0         0         0         0         0    0.0040         0    0.0036         0         0
         0    0.0071         0         0         0         0   -0.0089         0   -0.0071         0         0
         0    0.0036         0         0         0         0    0.0142   -0.0027   -0.0036         0         0
         0         0         0    0.0027   -0.0142         0         0         0         0         0         0
         0         0         0         0    0.0071    0.0036   -0.0142         0   -0.0142         0   -0.0142
         0   -0.0071   -0.0142    0.0036         0         0    0.0071         0         0         0         0
```

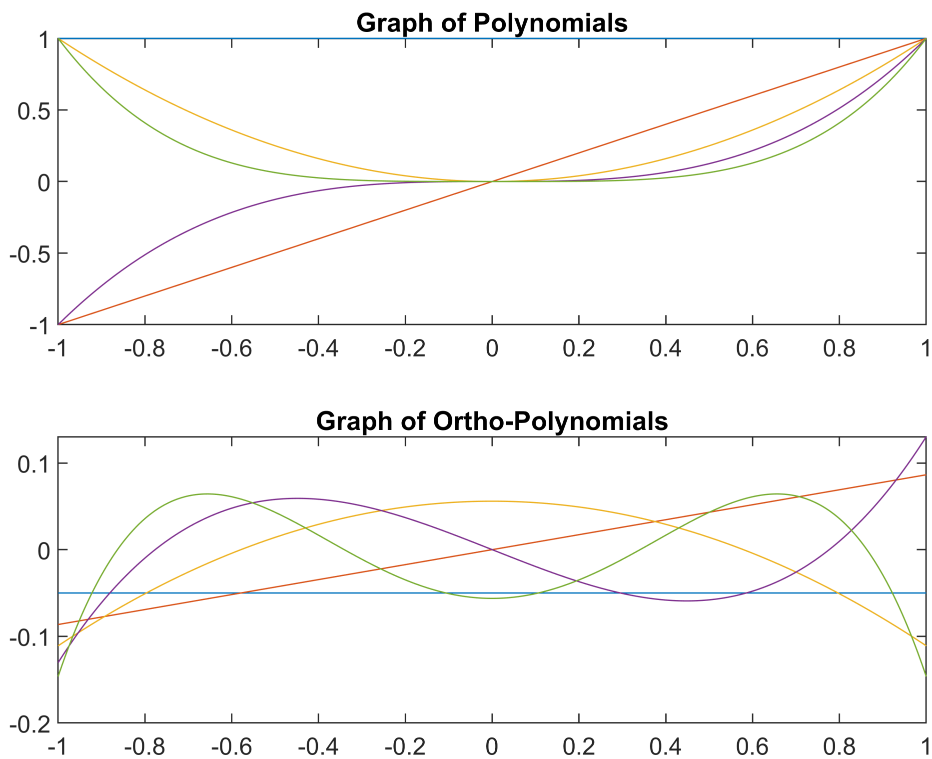
```
         0         0         0         0         0         0   -0.0089         0         0         0         0
         0         0         0         0   -0.0071         0    0.0071   -0.0142         0         0   -0.0036
         0         0         0         0         0   -0.0071   -0.0071   -0.0071         0         0    0.0213    0.
         0         0         0         0         0         0         0         0   -0.0071   -0.0142         0    0.
         0         0    0.0071         0    0.0071    0.0036   -0.0213   -0.0071    0.0142   -0.0071         0
         0         0         0         0    0.0071         0         0         0    0.0071         0   -0.0142    0.
         0         0         0         0   -0.0071         0    0.0044         0         0    0.0107         0
         0         0         0         0    0.0071         0   -0.0071    0.0071         0         0    0.0142    0.
         0         0   -0.0071         0   -0.0142         0   -0.0142   -0.0098    0.0142         0    0.0036   -0.
         0         0         0   -0.0018    0.0071         0   -0.0036         0   -0.0071         0   -0.0036
         0         0    0.0009   -0.0027   -0.0142    0.0016   -0.0071         0    0.0142         0    0.0071
         0         0         0         0   -0.0071         0   -0.0142   -0.0053    0.0036    0.0071   -0.0027
         0    0.0142         0         0    0.0022         0         0   -0.0071    0.0071         0   -0.0036   -0.
         0         0         0         0         0    0.0036   -0.0071         0    0.0018         0         0
         0         0         0         0         0         0   -0.0080         0         0   -0.0142   -0.0053
         0         0         0         0         0   -0.0071   -0.0036   -0.0071         0         0   -0.0071   -0.
         0         0         0         0         0         0    0.0071         0    0.0071   -0.0071    0.0142
         0         0         0         0    0.0142   -0.0115         0         0    0.0142   -0.0071    0.0049   -0.
         0         0         0         0         0         0         0         0   -0.0071         0         0   -0.
         0         0         0         0         0   -0.0036   -0.0142         0   -0.0284         0         0
         0         0         0         0   -0.0142         0   -0.0284   -0.0142         0         0   -0.0213    0.
         0         0    0.0018         0   -0.0071         0   -0.0142   -0.0087    0.0071         0    0.0071    0.
         0         0         0    0.0036         0         0         0         0   -0.0071    0.0089         0    0.
  Columns 19 through 20
         0    0.1137
         0         0
         0         0
         0   -0.0568
         0         0
   -0.0284         0
    0.0284         0
   -0.0284    0.0568
         0   -0.0568
   -0.0284    0.0568
   -0.0568    0.0568
   -0.1137         0
   -0.0568         0
         0         0
    0.0568         0
    0.0568    0.0568
    0.0284   -0.0568
    0.0142   -0.0284
   -0.0284         0
         0         0
   -0.0284    0.0568
    0.0568    0.1137
         0   -0.0568
   -0.0284         0
    0.0568         0
         0    0.0568
    0.0568         0
         0         0
         0         0
         0         0
```

SInce both functions give values that are roughly zero for the respective equation, the Gram-Schmidt algorithm gives a good approximation to the qr function for these two equations.

Problem 3.

This problem asks for a script that plots the ortho-polynomials in MATLAB.

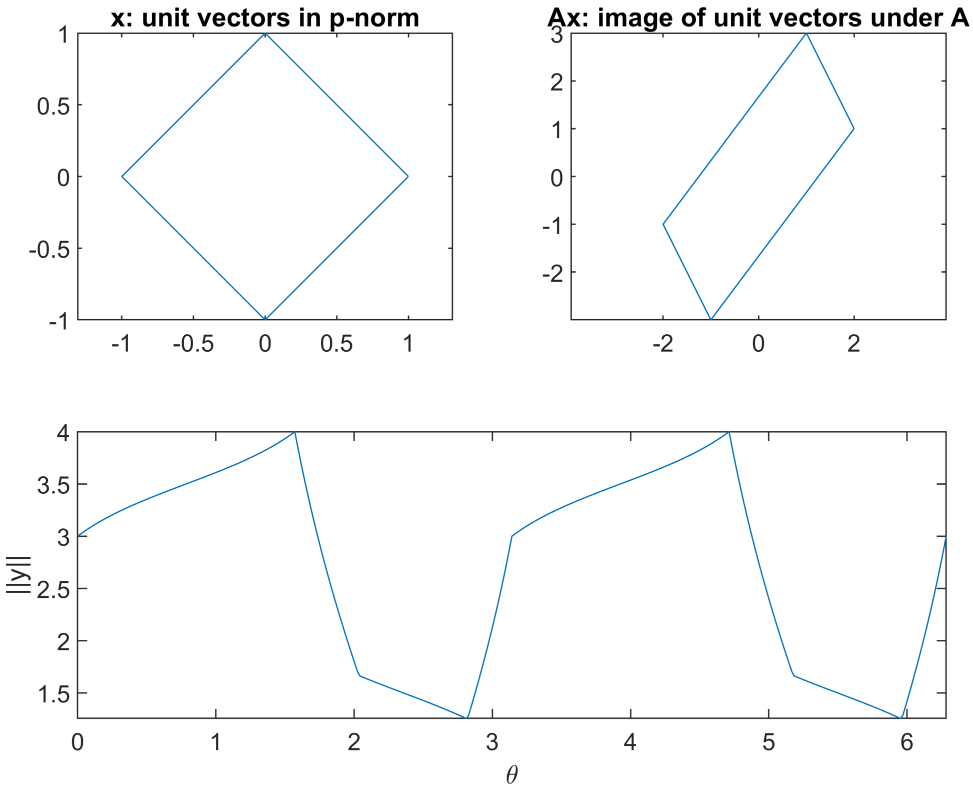
```
clf
x=linspace(-1,1,400)';
V=x.^(0:4);
[Q, ~] = qr(V,0);

subplot(2,1,1)
plot(x,V)
t1=title('Graph of Polynomials');

subplot(2,1,2)
plot(x,Q)
t2=title('Graph of Ortho-Polynomials');
```



Problem 4.

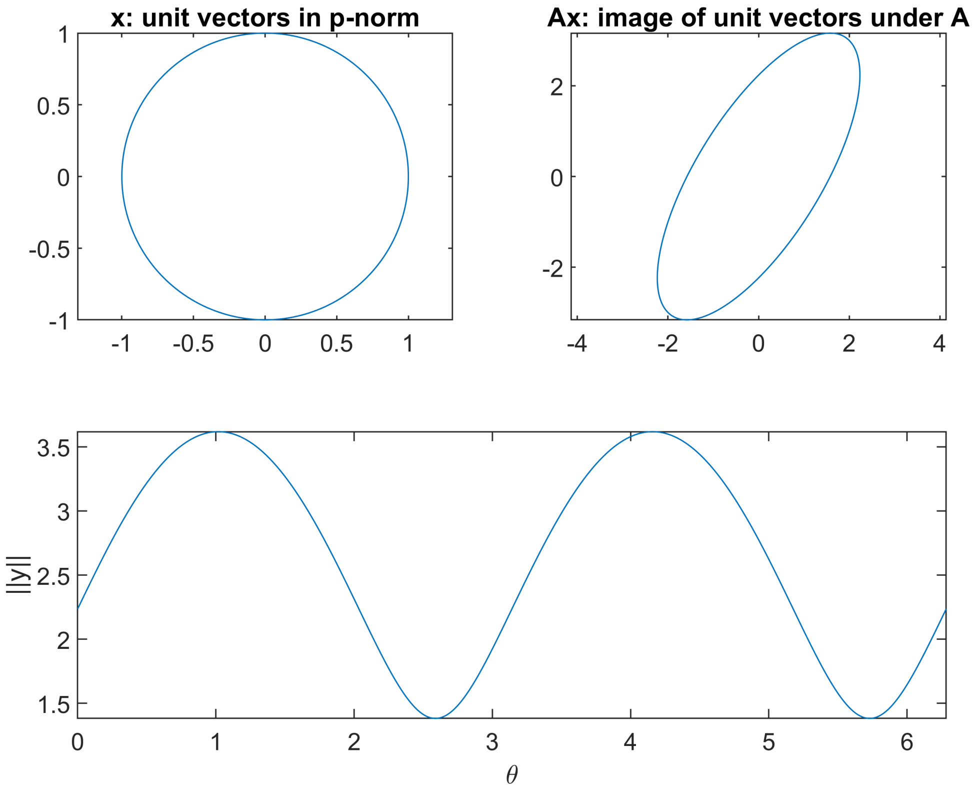
This problem asks for a script that creates a function that graphs the unit vectors in the p-norm and calculates the approximate p-norm while comparing it to the actual p-norm.

```
A = [2 1; 1 3];
visMatrixNorm(A,1)
```



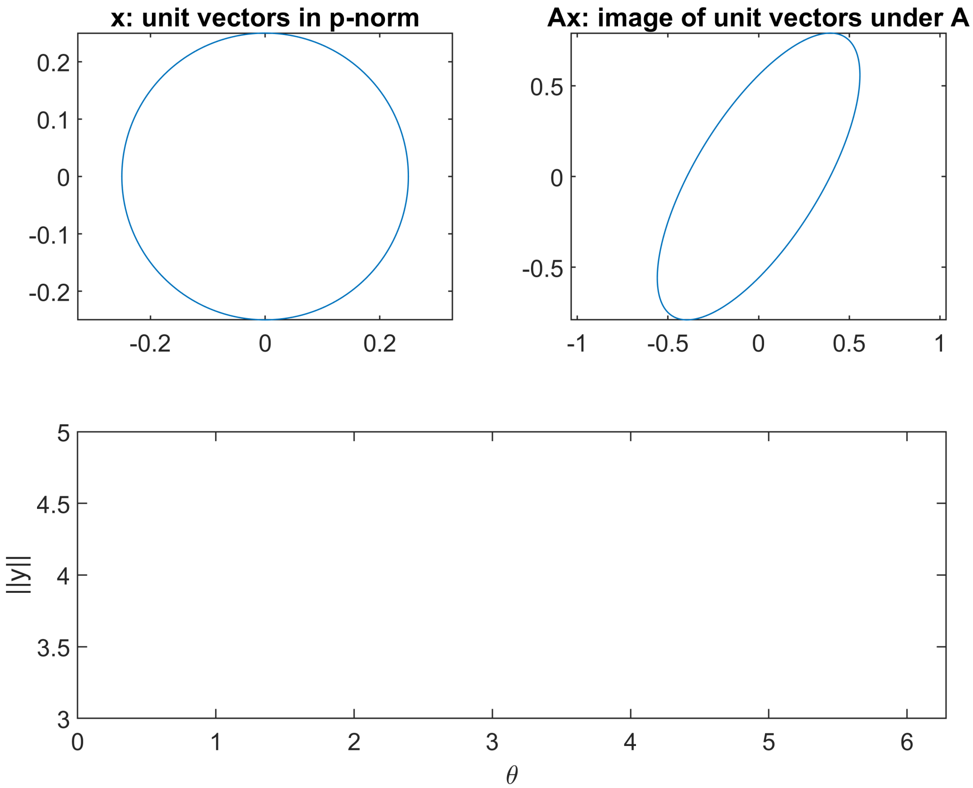

```
p = 1
 approx. norm: 4.0000000000000000
 actual norm: 4.0000000000000000
```

```
visMatrixNorm(A,2)
```




```
p = 2
 approx. norm: 3.6179964204609893
 actual norm: 3.6180339887498953
```

```
visMatrixNorm(A,inf);
```




```
p = Inf
 approx. norm: 4.0000000000000000
 actual norm: 4.0000000000000000
```

```
function g = visMatrixNorm(A,p)
    %input A (2x2 matrix)
    %input p (p-norm where p is 1, 2, or inf)
    theta = linspace(0, 2*pi, 361);
    X = [cos(theta); sin(theta)]; % x: unit vectors in 2-norm
    if p ~= inf
        pnorm = sum(abs(X).^(p)).^(1/p);
        X2 = X ./ pnorm;
        Y2=A*X2;
        norm_Y2=sum(abs(Y2).^(p)).^(1/p);

        % visualization
        clf
        subplot(2,2,1)
        plot(X2(1,:), X2(2,:)), axis equal
        title('x: unit vectors in p-norm')

        subplot(2,2,2)
        plot(Y2(1,:), Y2(2,:)), axis equal
        title('Ax: image of unit vectors under A')

        subplot(2,1,2)
        plot(theta, norm_Y2), axis tight
        xlabel('\theta')
```

```
        ylabel('||y||')

        % matrix norm approximation (and comparison)
        fprintf('p = %d \n',p)
        fprintf(' approx. norm: %18.16f\n', max(norm_Y2))
        fprintf(' actual norm: %18.16f\n', norm(A, p))

    else
        pnorm = max(abs(sum(A)));
        X2 = X ./ pnorm;
        Y2=A*X2;

        % visualization
        clf
        subplot(2,2,1)
        plot(X2(1,:), X2(2,:)), axis equal
        title('x: unit vectors in p-norm')

        subplot(2,2,2)
        plot(Y2(1,:), Y2(2,:)), axis equal
        title('Ax: image of unit vectors under A')

        subplot(2,1,2)
        plot(theta, pnorm), axis tight
        xlabel('\theta')
        ylabel('||y||')

        % matrix norm approximation (and comparison)
        fprintf('p = %d \n',p)
        fprintf(' approx. norm: %18.16f\n', max(pnorm))
        fprintf(' actual norm: %18.16f\n', norm(A, p))
    end
end


function [Q, R] = gs(S)
    %input S (m x n matrix where m>n)
    [m,n] = size(S);
    Q=zeros(m,n);
    R=zeros(n,n);
    B=S;
    for k=1:n %deals with column vectors
        Q(:,k) = B(:,k); %sets original column vectors equal to Q
        if k ~= 1 %if we are not in first column
            R(1:k-1,k) = Q(:,k-1)'*Q(:,k); %R matrix equals Q transpose times Q
            Q(:,k) = Q(:,k) - Q(:,1:k-1)*R(1:k-1,k);
            %next column in Q equals previous
            % minus the projections of new vector
        end
        R(k,k) = norm(Q(:,k)); %find norm of column vectors of Q
        Q(:,k) = Q(:,k)/R(k,k); %divide by norm to get unit vectors
    end
end
```