Thomas Penick 452 80 6040

Computer Assignment 5

beginning on page 124

Chapter 6, section 1, problem 1a: Orthogonality

Matlab Input:

```
A = eye(4) - 0.5*ones(4)
B = cyclic(4)
C = [54 -82 18 6;54 18 -82 6; 18 6 6 -98;62 54 54 18]/100

Atest = A'*A
Btest = B'*B
Ctest = C'*C
```

Matlab Output:

```
A =
    0.5000   -0.5000   -0.5000   -0.5000
   -0.5000    0.5000   -0.5000   -0.5000
   -0.5000   -0.5000    0.5000   -0.5000
   -0.5000   -0.5000   -0.5000    0.5000
B =
     0     1     0     0
     0     0     1     0
     0     0     0     1
     1     0     0     0
C =
    0.5400   -0.8200    0.1800    0.0600
    0.5400    0.1800   -0.8200    0.0600
    0.1800    0.0600    0.0600   -0.9800
    0.6200    0.5400    0.5400    0.1800
Atest =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
Btest =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
Ctest =
    1.0000         0         0    0.0000
         0    1.0000    0.0000    0.0000
         0    0.0000    1.0000    0.0000
    0.0000    0.0000    0.0000    1.0000
```

Therefore *A*, *B*, and *C*, are orthogonal matrices since $A^{T}A$, $B^{T}B$ and $C^{T}C$, all result in the identity function.

Thomas Penick 452 80 6040
M 340L-C March 5, 1998

Chapter 6, section 1, problem 1b: Orthogonality

Matlab Input:

```
A = eye(4) - 0.5*ones(4)
B = cyclic(4)
C = [54 -82 18 6;54 18 -82 6; 18 6 6 -98;62 54 54 18]/100

Ainv = inv(A)
Binv = inv(B)
Cinv = inv(C)
```

Matlab Output:

```
A =
    0.5000   -0.5000   -0.5000   -0.5000
   -0.5000    0.5000   -0.5000   -0.5000
   -0.5000   -0.5000    0.5000   -0.5000
   -0.5000   -0.5000   -0.5000    0.5000
B =
     0     1     0     0
     0     0     1     0
     0     0     0     1
     1     0     0     0
C =
    0.5400   -0.8200    0.1800    0.0600
    0.5400    0.1800   -0.8200    0.0600
    0.1800    0.0600    0.0600   -0.9800
    0.6200    0.5400    0.5400    0.1800
Ainv =
    0.5000   -0.5000   -0.5000   -0.5000
   -0.5000    0.5000   -0.5000   -0.5000
   -0.5000   -0.5000    0.5000   -0.5000
   -0.5000   -0.5000   -0.5000    0.5000
Binv =
     0     0     0     1
     1     0     0     0
     0     1     0     0
     0     0     1     0
Cinv =
    0.5400    0.5400    0.1800    0.6200
   -0.8200    0.1800    0.0600    0.5400
    0.1800   -0.8200    0.0600    0.5400
    0.0600    0.0600   -0.9800    0.1800
```

A, *B*, and *C* are all orthogonal matrices. The inverses are the transposes of the original matrices.

Thomas Penick 452 80 6040
M 340L-C March 5, 1998

Chapter 6, section 1, problem 1c: Orthogonality

Matlab Input:

```
A = eye(4) - 0.5*ones(4)
B = cyclic(4)
C = [54 -82 18 6;54 18 -82 6; 18 6 6 -98;62 54 54 18]/100

ABprod = A*B
ABtest = ABprod'*ABprod
BCprod = B*C
BCtest = BCprod'*BCprod
ACprod = A*C
ACtest = ACprod'*ACprod
```

Matlab Output:

```
A =
    0.5000   -0.5000   -0.5000   -0.5000
   -0.5000    0.5000   -0.5000   -0.5000
   -0.5000   -0.5000    0.5000   -0.5000
   -0.5000   -0.5000   -0.5000    0.5000
B =
     0     1     0     0
     0     0     1     0
     0     0     0     1
     1     0     0     0
C =
    0.5400   -0.8200    0.1800    0.0600
    0.5400    0.1800   -0.8200    0.0600
    0.1800    0.0600    0.0600   -0.9800
    0.6200    0.5400    0.5400    0.1800
ABprod =
   -0.5000    0.5000   -0.5000   -0.5000
   -0.5000   -0.5000    0.5000   -0.5000
   -0.5000   -0.5000   -0.5000    0.5000
    0.5000   -0.5000   -0.5000   -0.5000
ABtest =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
BCprod =
    0.5400    0.1800   -0.8200    0.0600
    0.1800    0.0600    0.0600   -0.9800
    0.6200    0.5400    0.5400    0.1800
    0.5400   -0.8200    0.1800    0.0600
BCtest =
    1.0000    0.0000         0    0.0000
    0.0000    1.0000    0.0000    0.0000
         0    0.0000    1.0000    0.0000
    0.0000    0.0000    0.0000    1.0000
ACprod =
   -0.4000   -0.8000    0.2000    0.4000
   -0.4000    0.2000   -0.8000    0.4000
   -0.7600    0.0800    0.0800   -0.6400
   -0.3200    0.5600    0.5600    0.5200
ACtest =
    1.0000    0.0000    0.0000    0.0000
    0.0000    1.0000         0    0.0000
    0.0000         0    1.0000    0.0000
    0.0000    0.0000    0.0000    1.0000
```

The products of the orthogonal matrices *A*, *B*, and *C* (*AB*, *BC*, and *AC*) are also orthogonal.

Thomas Penick 452 80 6040
M 340L-C March 5, 1998

Chapter 6, section 1, problem 1d: Orthogonality

Matlab Input:

```
A = eye(4) - 0.5*ones(4)
B = cyclic(4)
C = [54 -82 18 6;54 18 -82 6; 18 6 6 -98;62 54 54 18]/100
x = randint(4,1)

Length_x = norm(x)
Length_Ax = norm(A*x)
Length_Bx = norm(B*x)
Length_Cx = norm(C*x)
```

Matlab Output:

```
A =
    0.5000   -0.5000   -0.5000   -0.5000
   -0.5000    0.5000   -0.5000   -0.5000
   -0.5000   -0.5000    0.5000   -0.5000
   -0.5000   -0.5000   -0.5000    0.5000
B =
     0     1     0     0
     0     0     1     0
     0     0     0     1
     1     0     0     0
C =
    0.5400   -0.8200    0.1800    0.0600
    0.5400    0.1800   -0.8200    0.0600
    0.1800    0.0600    0.0600   -0.9800
    0.6200    0.5400    0.5400    0.1800
x =
     9
    -5
     2
     0
Length_x =
   10.4881
Length_Ax =
   10.4881
Length_Bx =
   10.4881
Length_Cx =
   10.4881
```

The Matlab command `norm()` computes the length of a matrix. The lengths of **x**, *A***x**, *B***x**, and *C***x** are all equal.

Thomas Penick 452 80 6040
M 340L-C March 5, 1998

Chapter 6, section 1, problem 2a: Computing Orthonormal Bases

Matlab Input:

```
v1 = [-149 537 -27 122]'
v2 = [-50 180 -9 41]'
v3 = [-154 546 -25 129]'
V = rot90([v3';v2';v1'],-1)
U = orth(V)
Test_U = U'*U
Test_RUeqRV = rank(V) == rank([V,U])
```

Matlab Output:

```
v1 =
  -149
   537
   -27
   122
v2 =
   -50
   180
    -9
    41
v3 =
  -154
   546
   -25
   129
V =
  -149   -50  -154
   537   180   546
   -27    -9   -25
   122    41   129
U =
   -0.2626   -0.3550    0.6868
    0.9390   -0.2608    0.2243
   -0.0452    0.4320    0.6914
    0.2175    0.7870    0.0045
Test_U =
    1.0000         0    0.0000
         0    1.0000    0.0000
    0.0000    0.0000    1.0000
Test_RUeqRV =
     1
```

The Matlab command **`Test_RUeqRV = rank(V) == rank([V,U])`** compares the rank of V with the rank of a matrix composed of the columns of V followed by the columns of U and returns a 1 if true. If the two column spaces were not equal, the addition of the column spaces in U to those in V would result in an increase in rank of that combination. Thus the two ranks in the expression above would be unequal and the command would return a 0.

Thomas Penick 452 80 6040
M 340L-C March 5, 1998

Chapter 6, section 1, problem 2b: Computing Orthonormal Bases

Matlab Input:

```
v1 = [-149 537 -27 122]';
v2 = [-50 180 -9 41]';
v3 = [-154 546 -25 129]';
V = rot90([v3';v2';v1'],-1)
r = rank(V)
[Q,R] = qr(V)
Q = Q(:,1:r)
Test_Q = Q'*Q
Test_RQeqRV = rank(V) == rank([V,Q])
```

Matlab Output:

```
V =
  -149   -50  -154
   537   180   546
   -27    -9   -25
   122    41   129
r =
     3
Q =
   -0.2609   -0.3733   -0.6777    0.5774
    0.9403   -0.2617   -0.2178    0.0000
   -0.0473    0.4144   -0.7019   -0.5774
    0.2136    0.7877   -0.0242    0.5774
R =
  571.1243  191.4732  582.2918
         0    0.1253    5.8538
         0         0   -0.1452
         0         0         0
Q =
   -0.2609   -0.3733   -0.6777
    0.9403   -0.2617   -0.2178
   -0.0473    0.4144   -0.7019
    0.2136    0.7877   -0.0242
Test_Q =
    1.0000    0.0000    0.0000
    0.0000    1.0000    0.0000
    0.0000    0.0000    1.0000
Test_RQeqRV =
     1
```

The Matlab command returns an orthogonal matrix Q and an upper triangular matrix R such that $A = QR$. In this exercise, the column vectors of Q are tested to see if they form an orthonormal set. We also check to see that $R(Q) = R(V)$.

Thomas Penick 452 80 6040
M 340L-C March 5, 1998

Chapter 6, section 1, problem 2c: Computing Orthonormal Bases

Matlab Input:

```
v1 = [-149 537 -27 122]';
v2 = [-50 180 -9 41]';
v3 = [-154 546 -25 129]';
V = rot90([v3';v2';v1'],-1)
r = rank(V); [Q,R] = qr(V); Q = Q(:,1:r)
G = gschmidt(V)
S = gschmidt(V,1)
format long
UtU = U'*U
QtQ = Q'*Q
GtG = G'*G
StS = S'*S
```

Matlab Output:

```
V =
  -149   -50  -154
   537   180   546
   -27    -9   -25
   122    41   129
Q =
   -0.2609   -0.3733   -0.6777
    0.9403   -0.2617   -0.2178
   -0.0473    0.4144   -0.7019
    0.2136    0.7877   -0.0242
G =
   -0.2609   -0.3733    0.6777
    0.9403   -0.2617    0.2178
   -0.0473    0.4144    0.7019
    0.2136    0.7877    0.0242
S =
   -0.2609   -0.3733    0.6777
    0.9403   -0.2617    0.2178
   -0.0473    0.4144    0.7019
    0.2136    0.7877    0.0242
UtU =
   1.00000000000000                   0   0.00000000000000
                  0   1.00000000000000   0.00000000000000
   0.00000000000000   0.00000000000000   1.00000000000000
QtQ =
   1.00000000000000   0.00000000000000   0.00000000000000
   0.00000000000000   1.00000000000000   0.00000000000000
   0.00000000000000   0.00000000000000   1.00000000000000
GtG =
   1.00000000000000   0.00000000000013  -0.00000000000458
   0.00000000000013   1.00000000000000   0.00000000000002
  -0.00000000000458   0.00000000000002   1.00000000000000
StS =
   1.00000000000000   0.00000000000013  -0.00000000000468
   0.00000000000013   1.00000000000000  -0.00000000051921
  -0.00000000000468  -0.00000000051921   1.00000000000000
```

The first two methods do the best job of producing an orthonormal basis. The first method uses the Matlab `orth()` command and the second uses the Matlab `[Q,R] = qr(V)` , where `r` is the rank of `V`, to return an orthogonal matrix *Q*.

The classical Gram-Schmidt process, used in the last example, does the worst job. This is produced by the Matlab command `gschmidt(V,1)`.