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

# Computer Assignment 2

## Chapter 2, section 1, problem 3

**Matlab commands:**

```
Apes = [70 50;30 60]
Diet = [25 40 20;15 30 25]
Result = Apes * Diet
```

**Matlab results:**

```
Apes =
    70    50
    30    60
Diet =
    25    40    20
    15    30    25
Result =
        2500        4300        2650
        1650        3000        2100
```

**The meaning of the calculation:**

$$\begin{array}{cl} \begin{array}{cc} \text{young} & \text{old} \end{array} & \\ \begin{bmatrix} 70 & 50 \\ 30 & 60 \end{bmatrix} & \begin{array}{l} \text{chimpanzees} \\ \text{gibbons} \end{array} \end{array}$$

$$\times \begin{array}{cl} \begin{array}{ccc} \text{prot.} & \text{carbo.} & \text{fat} \end{array} & \\ \begin{bmatrix} 25 & 40 & 20 \\ 15 & 30 & 25 \end{bmatrix} & \begin{array}{l} \text{young} \\ \text{old} \end{array} \end{array}$$

---

$$= \begin{array}{cl} \begin{array}{ccc} \text{prot.} & \text{carbo.} & \text{fat} \end{array} & \\ \begin{bmatrix} 2500 & 4300 & 2650 \\ 1650 & 3000 & 2100 \end{bmatrix} & \begin{array}{l} \text{young and old chimpanzees} \\ \text{young and old gibbons} \end{array} \end{array}$$

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

**Chapter 2, section 1, problem 10a**

**Matlab commands:**

```
A = randint(3,4,5,2)
% RANDINT(m,n,k,r) is an m by n matrix of rank r
% with integer entries in the interval [-k:k].
% If less than three arguments are used the default
% value of k is taken to be 9.
% If only one input argument is used then it is assumed
% that the matrix is square.
% If the last argument is left off, no attempt is made
% to determine the rank.
rref(A)
```

**Matlab results:**

```
A =
     0     3     0     3
    -3    -2     3     1
    -2     1     2     3
ans =
     1     0    -1    -1
     0     1     0     1
     0     0     0     0
```

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

```
x1 = [4;-2;2;2];
x2 = [5;-2;3;2];
a = 3; b = -2;
x3 = a * x1 + b * x2
Product1 = A*x1
Product2 = A*x2
Product3 = A*x3
```

**Matlab results:**

```
x3 =
     2
    -2
     0
     2
Product1 =
     0
     0
     0
Product2 =
     0
     0
     0
Product3 =
     0
     0
     0
```

***x1*** and ***x2*** are two non-zero non-multiple solutions of ***Ax*** = 0. Two values for *a* and *b* were selected at random and ***x3*** was formed by multiplying ***x1*** by *a* and adding to the product of ***x2*** and *b*. Then the products of ***A********x1***, ***A********x2***, and ***A********x3*** were all found to be zero confirming that ***x1***, ***x2***, and ***x3*** are all solutions of ***Ax*** = 0.

Therefore it appears that any sum of multiples of solutions of a homogeneous linear system is also a solution of that system.

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

**<u>Chapter 2, section 1, problem 10b</u>**

**Matlab commands:**

```
p = randint(4,1,5)
% RANDINT(m,n,k,r) is an m by n matrix of rank r
% with integer entries in the interval [-k:k].
% If less than three arguments are used the default
% value of k is taken to be 9.
% If only one input argument is used then it is assumed
% that the matrix is square.
% If the last argument is left off, no attempt is made
% to determine the rank.
B = A*p
C = p + 3*x1 + (-2)*x2
B2 = A*C
```

**Matlab results:**

```
p =
     4
     3
     0
    -5
B =
    -6
   -23
   -20
C =
     6
     1
     0
    -3
B2 =
    -6
   -23
   -20
```

A random vector ***p*** is created and ***B*** = ***A*** * ***p*** is computed so that ***p*** is a solution of the nonhomogeneous system ***Ax*** = ***B***. Using values from problem 10a, we compute ***p*** + ***a*x1*** + ***b*x2*** which is stored in ***C***.

To check to see if ***C*** is also a solution of ***Ax*** = ***B***, we compute ***B2*** = ***A*** * ***C***. It is found that ***B2*** = ***B*** so therefore ***p*** + ***a*x1*** + ***b*x2*** is also a solution of ***Ax*** = ***B***.

Therefore it appears that any sum of multiples of solutions of a nonhomogeneous linear system is also a solution of that system.

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

## Chapter 2, section 1, problem 14a

### Matlab commands:

```
A = [3 2;4 3]
B = [3 -2;-4 3]
DeterminantA = det(A)
DeterminantB = det(B)
AB = A*B
BA = B*A
InverseA = inv(A)
InverseB = inv(B)
```

### Matlab results:

```
A =
     3     2
     4     3
B =
     3    -2
    -4     3
DeterminantA =
     1
DeterminantB =
     1
AB =
     1     0
     0     1
BA =
     1     0
     0     1
InverseA =
     3    -2
    -4     3
InverseB =
     3     2
     4     3
```

The definition of an inverse matrix states that "if the transformation $\boldsymbol{y} = \boldsymbol{Ax}$ is invertible, its inverse is $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{y}$." It follows that:

$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{y}$ and $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{A}\boldsymbol{x}$

$\boldsymbol{y}/\boldsymbol{y} = \boldsymbol{A}\boldsymbol{A}^{-1}$ $\quad$ $\boldsymbol{x}/\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{A}$

$1 = \boldsymbol{A}\boldsymbol{A}^{-1}$ $\quad$ $1 = \boldsymbol{A}^{-1}\boldsymbol{A}$

and if $\boldsymbol{A}^{-1} = \boldsymbol{B}$ then:

$1 = \boldsymbol{AB}$ and $1 = \boldsymbol{BA}$

Both ***A*** and ***B*** are nonsingular and therefore invertable since their determinants are not equal to zero. The product of ***A********B*** and ***B********A*** both yield the identity matrix. So it is seen that ***A*** is the inverse matrix of ***B*** and that ***B*** is the inverse matrix of ***A***. This is further confirmed by using the Matlab command inv() to find the inverse.

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

## Chapter 2, section 1, problem 14b

### Matlab commands:

```
A = [1 0 1;1 1 0;-1 2 -2]
B = [-2 2 -1;2 -1 1;3 -2 1]
DeterminantA = det(A)
DeterminantB = det(B)
AB = A*B
BA = B*A
InverseA = inv(A)
InverseB = inv(B)
```

### Matlab results:

```
A =
     1     0     1
     1     1     0
    -1     2    -2
B =
    -2     2    -1
     2    -1     1
     3    -2     1
DeterminantA =
     1
DeterminantB =
     1
AB =
     1     0     0
     0     1     0
     0     0     1
BA =
     1     0     0
     0     1     0
     0     0     1
InverseA =
    -2     2    -1
     2    -1     1
     3    -2     1
InverseB =
    1.0000    0.0000    1.0000
    1.0000    1.0000         0
   -1.0000    2.0000   -2.0000
```

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

**Chapter 2, section 1, problem 14c**

**Matlab commands:**

```
A = [0 0 0 0 1;1 0 0 0 0;0 1 0 0 0;0 0 0 1 0;0 0 1 0 0]
B = [0 1 0 0 0;0 0 1 0 0;0 0 0 0 1;0 0 0 1 0;1 0 0 0 0]
DeterminantA = det(A)
DeterminantB = det(B)
AB = A*B
BA = B*A
InverseA = inv(A)
InverseB = inv(B)
```

**Matlab results:**

```
A =
     0     0     0     0     1
     1     0     0     0     0
     0     1     0     0     0
     0     0     0     1     0
     0     0     1     0     0
B =
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     0     1
     0     0     0     1     0
     1     0     0     0     0
DeterminantA =
    -1
DeterminantB =
    -1
AB =
     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
BA =
     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
InverseA =
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     0     1
     0     0     0     1     0
     1     0     0     0     0
InverseB =
     0     0     0     0     1
     1     0     0     0     0
     0     1     0     0     0
     0     0     0     1     0
     0     0     1     0     0
```

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

**<u>Chapter 2, section 1, problem 14d</u>**

**Matlab commands:**

```
A = [1 1 1;1 1 0]
B = [1 1;-1 0;1 -1]
AB = A*B
InverseA = inv(A)
```

**Matlab results:**

```
A =
     1     1     1
     1     1     0
B =
     1     1
    -1     0
     1    -1
AB =
     1     0
     0     1
 ??? Error using ➔ inv
Matrix must be square.
```

Although ***A*** * ***B*** yields an identity matrix, ***B*** is not the inverse matrix of ***A*** because neither is a square matrix.

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

## Chapter 2, section 1, problem 16a

### Matlab commands:

```
A = [0.2 0.3 0.4;0.4 0.4 0.1;0.5 0.1 0.3]
I = [1 0 0;0 1 0;0 0 1]
IMinusA = I-A
Determinant = det(I-A)
Inverse = inv(I-A)
```

### Matlab results:

```
A =
    0.2000    0.3000    0.4000
    0.4000    0.4000    0.1000
    0.5000    0.1000    0.3000
I =
     1     0     0
     0     1     0
     0     0     1
IMinusA =
    0.8000   -0.3000   -0.4000
   -0.4000    0.6000   -0.1000
   -0.5000   -0.1000    0.7000
Determinant =
    0.0930
Inverse =
    4.4086    2.6882    2.9032
    3.5484    3.8710    2.5806
    3.6559    2.4731    3.8710
```

*I* - *A* is shown to be non-singular, $(I - A)^{-1}$ exists and has all non-negative entries, therefore $\mathbf{p} = (I - A)^{-1}\mathbf{y}$ exists and has all non-negative entries. This means that $(I - A)\mathbf{p} = \mathbf{y}$ has a solution for every possible demand vector **y**.

## Chapter 2, section 1, problem 16b

### Matlab commands:

```
A = [4.5 0.3 0.4;0.4 0.4 0.1;0.5 0.1 0.3]
I = [1 0 0;0 1 0;0 0 1];
IMinusA = I-A
Determinant = det(I-A)
Inverse = inv(I-A)
```

### Matlab results:

```
A =
    4.5000    0.3000    0.4000
    0.4000    0.4000    0.1000
    0.5000    0.1000    0.3000
IMinusA =
   -3.5000   -0.3000   -0.4000
   -0.4000    0.6000   -0.1000
   -0.5000   -0.1000    0.7000
Determinant =
   -1.6700
Inverse =
   -0.2455   -0.1497   -0.1617
   -0.1976    1.5868    0.1138
   -0.2036    0.1198    1.3293
```

With the (1,1) entry of *A* having a value of 4.5, *A* is no longer a realistic consumption matrix because this would mean that the input would be higher than the output.

As shown in Matlab, the determinant of *I* - *A* is not non-negative.