

CHAPTER 11

Matrices

Matrices are rectangular arrays of numbers, the position and value of each number being important. A matrix is usually, but not always, shown in brackets thus:

$$\begin{bmatrix} 2 & -3 & 6 \\ -4 & -4 & 5 \end{bmatrix}$$

The size of a matrix is given by the number of rows and columns. We say that a matrix is an M by N matrix if it has M rows and N columns. Thus the example above is a 2 by 3 matrix. Here are some more examples of matrices.

$$\begin{bmatrix} 8 & 12 & 0 \\ 1 & 1 & 8 \\ 8 & 12 & 1 \end{bmatrix}$$

A 3 by 3 matrix

$$\begin{bmatrix} 9 & 12 \\ 0 & 1 \\ 8 & -2 \end{bmatrix}$$

A 3 by 2 matrix

$$\begin{bmatrix} 9 & 12 \\ 3 & 2 \end{bmatrix}$$

A 2 by 2 matrix

If the number of rows in a matrix equals the number of columns in the matrix then we say that it is a square matrix.

Matrices are a useful concept for holding information in a concise way. For instance, suppose that you have calculated the amount spent in your household on three types of fuel each quarter during a year. You could display the information in a table as follows:

Quarter	Gas	Electric	Solid fuel
1	30	28	5
2	27	19	4
3	25	15	0
4	32	27	3

The data can be stored in a 4 by 3 matrix.

$$\begin{bmatrix} 30 & 28 & 5 \\ 27 & 19 & 4 \\ 25 & 15 & 0 \\ 32 & 27 & 3 \end{bmatrix}$$

You can store matrices in your computer by using two-dimensional arrays such as A(I,J). For instance, the matrix of fuel expenditure could be stored by making the following assignments.

$$\begin{array}{lll} A(1,1) = 30 & A(1,2) = 28 & A(1,3) = 5 \\ A(2,1) = 27 & A(2,2) = 19 & A(2,3) = 4 \\ A(3,1) = 25 & A(3,2) = 15 & A(3,3) = 0 \\ A(4,1) = 32 & A(4,2) = 28 & A(4,3) = 3 \end{array}$$

The information would, of course, be entered by using READ statements.

```
FOR I=1 TO 4:FOR J=1 TO 3:READ A(I,J):NEXT:NEXT
DATA 30,28,5,27,19,4,25,15,0,32,27,3
```

The following short program illustrates the data being read in and then displayed.

Listing 11.1

LIST

```
10 CLS:@%=4:DIM A(4,3):PRINT '
20 FOR I=1 TO 4:FOR J=1 TO 3:READ A(I
,J):NEXT:NEXT
30 FOR I=1 TO 4
40 FOR J=1 TO 3:PRINT A(I,J),;:NEXT
50 PRINT
60 NEXT
70 DATA 30,28,5,27,19,4,25,15,0,32,27
,3
```

RUN

```
30 28 5
27 19 4
25 15 0
32 27 3
```

Note that to use the memory space of the computer more efficiently we should have started the I and J counters at 0 instead of 1 since arrays begin at 0. However, conceptually the above is easier.

Adding matrices

Two matrices may be added together provided that they are of the same size. In other words two matrices may be added together provided that they both have the same number of rows and the same number of columns. This addition is performed term-by-term of the corresponding terms in the corresponding positions. For example

$$\begin{bmatrix} 1 & 5 & 4 \\ -2 & 3 & 1 \end{bmatrix} + \begin{bmatrix} 2 & 0 & 5 \\ 3 & 2 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1+2 & 5+0 & 4+5 \\ -2+3 & 3+2 & 1+1 \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 5 & 9 \\ 1 & 5 & 2 \end{bmatrix}$$

Why add matrices?

An example of the expenditure of fuel for the four quarters of a year was given earlier on. Suppose that during the next year the corresponding matrix is as follows:

$$\begin{bmatrix} 34 & 27 & 4 \\ 30 & 20 & 3 \\ 20 & 16 & 1 \\ 35 & 29 & 5 \end{bmatrix}$$

By adding the two matrices together we can obtain the amount spent on the three fuels during a particular quarter in the two years. The result is the following matrix:

$$\begin{bmatrix} 64 & 55 & 9 \\ 57 & 39 & 7 \\ 45 & 31 & 1 \\ 67 & 56 & 8 \end{bmatrix}$$

Your computer can add matrices, or two-dimensional arrays, quite easily. Suppose that we have two arrays A(I,J) and B(I,J) where in each case I ranges from 1 to 4 and J ranges from 1 to 3. Addition of these two arrays will produce a third array C(I,J) in the following way:

```
FOR I= 1 TO 4:FOR J=1 TO 3
C(I,J) = A(I,J) + B(I,J)
NEXT: NEXT
```

Subtraction of matrices uses the same rules as addition of matrices.

Matrix multiplication

Matrix multiplication is a quite subtle concept. In order that two matrices may be multiplied it is required that the number of columns in the first matrix is equal to the number of rows in the second matrix. Thus a 4 by 3 matrix can be multiplied with a 3 by 2 matrix. But a 4 by 3 matrix cannot be multiplied with a 2 by 3 matrix.

The product of an M by N matrix with an N by P matrix will be an M by P matrix. The actual process is best illustrated with an example, the explanation being given afterwards. Suppose we want to multiply the following matrices:

$$\begin{bmatrix} 2 & -3 & 6 \\ -4 & 4 & 5 \end{bmatrix} \quad \text{First matrix, a 2 by 3 matrix}$$

$$\begin{bmatrix} 9 & 12 \\ 0 & 1 \\ 8 & -2 \end{bmatrix} \quad \text{Second matrix, a 3 by 2 matrix}$$

The number of columns in the first matrix equals the number of rows in the second and so we may multiply the first by the second.

$$\begin{bmatrix} 2 & -3 & 6 \\ -4 & 4 & 5 \end{bmatrix} * \begin{bmatrix} 9 & 12 \\ 0 & 1 \\ 8 & -2 \end{bmatrix}$$

$$= \begin{bmatrix} 2*9 + -3*0 + 6*8 & 2*12 + -3*1 + 6*-2 \\ -4*9 + 4*0 + 5*8 & -4*12 + 4*1 + 4*-2 \end{bmatrix}$$

$$= \begin{bmatrix} 66 & 9 \\ 4 & -52 \end{bmatrix}$$

To perform the multiplication first look at each row of the first matrix and each column of the second matrix. The number of elements in each of these is the same. For each of these rows and columns we multiply the first of each together. This produces a number for the product matrix.

On your computer you can form the product of the M by N matrix A(I,J) with the N by P matrix B(I,J) in the following way. The result is an M by P matrix C(I,J).

```

FOR I=1 TO M
FOR J=1 TO P
C(I,J) = 0
FOR K=1 TO N : C(I,J) = C(I,J) + A(I,K)*B(K,J) : NEXT
NEXT
NEXT
    
```

Note that, if A and B are two matrices and if you can form the product A*B, you may not be able to form the product B*A. Even if the product B*A exists it is not necessarily equal to A*B. Find some simple examples!

Why multiply matrices?

Let's look at our household fuel costs example once again. Recall the original details.

Quarter	Gas	Electric	Solid fuel	
1		30	28	5
2		27	19	4
3		25	15	0
4		32	27	3

Suppose that there have been two estimates of the likely increases in the costs of these fuels.

	Estimate 1	Estimate 2
Gas	10%	5%
Electric	5%	10%
Solid fuel	10%	10%

Essential Maths on the BBC and Electron Computers

These increases are given as decimals in the next table.

	Estimate 1	Estimate 2
Gas	0.1	0.05
Electric	0.05	0.1
Solid fuel	0.1	0.1

How much extra would you have to pay for the three fuels during each quarter? The answer depends on which estimate you use. The result may be tabulated as shown:

Quarter	Estimate 1	Estimate 2
1	$30 \cdot 1 + 28 \cdot 0.05 + 5 \cdot 0.1$	$30 \cdot 0.05 + 28 \cdot 0.1 + 5 \cdot 0.1$
2	$27 \cdot 0.1 + 19 \cdot 0.05 + 4 \cdot 0.1$	$27 \cdot 0.05 + 19 \cdot 0.1 + 4 \cdot 0.1$
3	$25 \cdot 0.1 + 15 \cdot 0.05 + 0 \cdot 0.1$	$25 \cdot 0.05 + 15 \cdot 0.1 + 0 \cdot 0.1$
4	$32 \cdot 0.1 + 27 \cdot 0.05 + 3 \cdot 0.1$	$32 \cdot 0.05 + 27 \cdot 0.1 + 3 \cdot 0.1$

This is evaluated to give the following table.

Quarter	Estimate 1	Estimate 2
1	4.9	4.8
2	4.05	3.65
3	3.24	2.75
4	4.85	4.6

You have probably seen that the resulting matrix is the product of the matrix of expenditure with the matrix of estimates. In other words it is the following product of matrices.

$$\begin{bmatrix} 30 & 28 & 5 \\ 27 & 19 & 4 \\ 25 & 15 & 0 \\ 32 & 27 & 3 \end{bmatrix} * \begin{bmatrix} 0.1 & 0.05 \\ 0.05 & 0.1 \\ 0.1 & 0.1 \end{bmatrix}$$

Zeros and ones

Matrices that consist of zeros alone are called zero matrices. Adding a zero matrix to another matrix has no effect. Multiplying a matrix by a zero matrix produces a zero matrix. For instance:

$$\begin{bmatrix} 30 & 28 \\ 27 & 19 \\ 25 & 15 \end{bmatrix} * \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\begin{aligned}
 &= \begin{bmatrix} 30*0 + 28*0 & 30*0 + 28*0 \\ 27*0 + 19*0 & 27*0 + 19*0 \\ 25*0 + 15*0 & 25*0 + 15*0 \end{bmatrix} \\
 &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}
 \end{aligned}$$

The product of two non-zero matrices can be zero. An example is given below.

$$\begin{aligned}
 &\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 1*1 + 1*-1 \\ 1*1 + 1*-1 \end{bmatrix} \quad \begin{bmatrix} 1*-1 + 1*1 \\ 1*-1 + 1*1 \end{bmatrix} \\
 &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}
 \end{aligned}$$

A square matrix that has ones down the diagonal (top left to bottom right) and zeroes elsewhere is called an *identity* matrix. For instance, both of the following are identity matrices.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

A matrix multiplied by an identity matrix equals itself. In other words the identity matrix behaves like a 1 does in ordinary number multiplication. For example:

$$\begin{aligned}
 &\begin{bmatrix} 30 & 28 \\ 27 & 19 \\ 25 & 15 \end{bmatrix} * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 30*1 + 28*0 \\ 27*1 + 19*0 \\ 25*1 + 15*0 \end{bmatrix} \quad \begin{bmatrix} 30*0 + 28*1 \\ 27*0 + 19*1 \\ 25*0 + 15*1 \end{bmatrix}
 \end{aligned}$$

$$= \begin{bmatrix} 30 & 28 \\ 27 & 19 \\ 25 & 15 \end{bmatrix}$$

Inverses of matrices

The inverse of reciprocal of a matrix, if it exists, has the same property as that of the inverse of an ordinary number. The inverse of the number 4 is 0.25 and

$$4 * 0.25 = 1, \quad 0.25 * 4 = 1$$

For a matrix A the inverse (if it exists) is denoted by A^{-1} and

$$A * A^{-1} = I, \quad A^{-1} * A = I$$

where I denotes an identity matrix.

Only square matrices can have inverses. In fact only certain square matrices have inverses.

Several methods exist for finding an inverse of a square matrix. There is a general step-by-step method well suited for your BBC or Electron micro. It consists of performing so-called row operations. Roughly speaking we place an identity matrix next to the matrix whose inverse we want. The row operations are performed on both matrices simultaneously until the original is converted to an identity matrix. The matrix that was the identity is now the inverse matrix.

A program that finds inverses of matrices is given later on. Meanwhile the method is briefly described.

Suppose we want to find the inverse of the following matrix.

$$\begin{bmatrix} -2 & 1 \\ -2 & 4 \end{bmatrix}$$

Place the identity 2 by 2 matrix next to this matrix.

$$\begin{bmatrix} 2 & 1 \\ -2 & 4 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

We want the top left entry to be a 1 and so we divide the whole first row by 2.

$$\begin{bmatrix} 1 & 0.5 \\ -2 & 4 \end{bmatrix} \quad \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}$$

Next we want the bottom left entry to be 0 and so we add twice the first row to the second row.

$$\begin{bmatrix} 1 & 0.5 \\ 0 & 5 \end{bmatrix} \quad \begin{bmatrix} 0.5 & 0 \\ 1 & 1 \end{bmatrix}$$

Now we want the bottom right entry (of the left matrix) to be 1. This is achieved by dividing the second row by 5.

$$\begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0.5 & 0 \\ 0.2 & 0.2 \end{bmatrix}$$

To make the left matrix the identity we subtract half of the second row from the first.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0.4 & -0.1 \\ 0.2 & 0.2 \end{bmatrix}$$

The matrix on the right is the inverse of the matrix we started with. The following calculation verifies this fact.

$$\begin{aligned} & \begin{bmatrix} 2 & 1 \\ -2 & 4 \end{bmatrix} * \begin{bmatrix} 0.4 & -1 \\ 0.2 & 0.2 \end{bmatrix} \\ = & \begin{bmatrix} 2*0.4 + 1*0.2 & 2*-1 + 1*0.2 \\ -2*0.4 + 4*0.2 & -2*-1 + 4*0.2 \end{bmatrix} \\ = & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

The next program finds the inverses of square matrices, that is, if an inverse exists.

Listing 11.2

LIST

```

10 REM Inverses of matrices
20 MODE 1:COLOUR 3:PRINT ' TAB(10);"I
nverses of matrices":@%=10
30 PRINT "This program determines the
inverse of an N by N matrix.'"
40 COLOUR 1:PRINT "Enter the size of
the square matrix."
50 REPEAT
60 INPUT '"Value of N? " N
70 IF N<1 OR N<>INT(N) THEN COLOUR 3

```

Essential Maths on the BBC and Electron Computers

```
:PRINT "Try a sensible number.":COLOUR 1
  80 UNTIL N>0 AND N=INT(N)
  90 PRINT "Enter the matrix, term by
term."
  100 DIM A(N+1,N),B(N,N)
  110 REM Read in the matrix
  120 FOR I=1 TO N
  130 PRINT "  Row ";I
  140 FOR J=1 TO N
  150 PRINT "Column ";J;" ";:INPUT A(I
,J)
  160 NEXT
  170 PRINT
  180 NEXT
  190 COLOUR 2:PRINT "Calculating ";:@%
=&040A
  200 FOR I=1 TO N:B(I,I)=1:NEXT
  210 X=0:GOSUB 300
  220 COLOUR 3:PRINT CHR$(7) ' ' TAB(10);
"Another go? Y or N ";
  230 REPEAT:G$=GET$:UNTIL G$="Y" OR G$=
"N"
  240 IF G$="Y" THEN RUN
  250 CLS:PRINT "Bye for now.":@%=10:EN
D
  300 REM Calculating
  310 REPEAT
  320 X=X+1:Z=X:PRINT "*";
  330 REPEAT
  340 IF A(Z,X)=0 THEN Z=Z+1
  350 UNTIL A(Z,X)<>0 OR Z>N
  360 IF Z>N THEN PRINT '"No inverse!!
":RETURN
  370 IF Z<>X THEN R=1/A(Z,X):I=X:K=Z:P
ROCRoWAdd
  380 IF A(X,X)<>1 THEN R=1/A(X,X):I=X:
PROCRoWTimes
  390 FOR I=1 TO N
```

```
400     IF I=X THEN I=I+1
410     IF I<=N:IF A(I,X)<>0 THEN R=-A(I
,X):K=X:PROCRowAdd
420     NEXT
430 UNTIL X>=N
440 PRINT "'The inverse:"
450 FOR I=1 TO N:FOR J=1 TO N:PRINT B(
I,J),;:NEXT:PRINT:NEXT
460 RETURN
500 REM Row operation add R times row
K to row I
510 DEF PROCRowAdd
520 FOR J=1 TO N
530  A(I,J)=A(I,J)+R*A(K,J):B(I,J)=B(I
,J)+R*B(K,J)
540 NEXT
550 ENDPROC
560 REM Row operation multiply row I b
y R
570 DEF PROCRowTimes
580 FOR J=1 TO N
590  A(I,J)=R*A(I,J):B(I,J)=R*B(I,J)
600 NEXT
610 ENDPROC
RUN
```

Inverses of matrices

This program determines the inverse of an N by N matrix.

Enter the size of the square matrix.

Value of N? 2

Enter the matrix, term by term.

Row 1

Essential Maths on the BBC and Electron Computers

Column 1 ?1

Column 2 ?2

Row 2

Column 1 ?-1

Column 2 ?2

Calculating **

The inverse:

$$\begin{array}{cc} 0.5 & -0.5 \\ 0.25 & 0.25 \end{array}$$

Another go? Y or N

Simultaneous equations

Matrices are useful in solving simultaneous equations. An example of 2 simultaneous equations in 2 unknowns is given below.

$$\begin{array}{l} 3*X + 1*Y = 7 \\ 5*X + 2*Y = 9 \end{array}$$

This is written in matrix form as follows.

$$\begin{bmatrix} 3 & 1 \\ 5 & 2 \end{bmatrix} * \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} 7 \\ 9 \end{bmatrix}$$

One way to solve the problem is to find the inverse of the 2 by 2 matrix on the left and then multiply the above equation by this inverse. In the example the inverse is given by the following matrix:

$$\begin{bmatrix} 2 & -1 \\ -5 & 3 \end{bmatrix}$$

We therefore obtain the following equations.

$$\begin{aligned}
 & \begin{bmatrix} X \\ Y \end{bmatrix} \\
 = & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} X \\ Y \end{bmatrix} \\
 = & \begin{bmatrix} 2 & -1 \\ -5 & 3 \end{bmatrix} * \begin{bmatrix} 3 & 1 \\ 5 & 2 \end{bmatrix} * \begin{bmatrix} X \\ Y \end{bmatrix} \\
 = & \begin{bmatrix} 2 & -1 \\ -5 & 3 \end{bmatrix} * \begin{bmatrix} 7 \\ 9 \end{bmatrix} \\
 = & \begin{bmatrix} 2*7 - 1*9 \\ -5*7 + 3*9 \end{bmatrix} \\
 = & \begin{bmatrix} 5 \\ -8 \end{bmatrix}
 \end{aligned}$$

Thus $X = 5$ and $Y = -8$ solves the simultaneous equations.

The next program Simultaneous Equations essentially goes through this process to solve a set of N simultaneous equations in N unknowns. In fact the process is to write the set of simultaneous equations into two matrices $A(I,J)$ and $B(I)$. The example given earlier on would be written as follows.

$$\begin{bmatrix} 3 & 1 \\ 5 & 2 \end{bmatrix} \begin{bmatrix} 7 \\ 9 \end{bmatrix}$$

Row operations are then performed until the matrix on the left becomes the identity matrix. When this is achieved the matrix on the right becomes the solution to the set of simultaneous equations.

Listing 11.3

LIST

```

10 REM Simultaneous equations
20 MODE 1:COLOUR 3:PRINT ' TAB(9); "Si
multaneous equations":@%=10
30 PRINT "This program solves a set o
f N simult- aneous equations in N unkno
wns." '
40 COLOUR 1:PRINT "Enter the number o
f equations."

```

Essential Maths on the BBC and Electron Computers

```
50 REPEAT
60 INPUT "Value of N? " N
70 IF N<1 OR N<>INT(N) THEN COLOUR 3
:PRINT "Try a sensible number.":COLOUR 1
80 UNTIL N>0 AND N=INT(N)
90 PRINT "Enter the coefficients, term by term."
100 DIM A(N+1,N),B(N)
110 REM Read in the matrix of coefficients
120 FOR I=1 TO N
130 PRINT " Row ";I
140 FOR J=1 TO N
150 PRINT "Column ";J;" ";:INPUT A(I,J)
160 NEXT
170 INPUT "Right hand term ";B(I):PRINT
180 NEXT
190 CLS:@%=&040A:PRINT "The matrix of coefficients:"
200 FOR I=1 TO N:FOR J=1 TO N:PRINT A(I,J),,,:NEXT:PRINT B(I):NEXT
210 COLOUR 2:PRINT "Calculating ";
220 X=0:GOSUB 300
230 REM Ending
240 COLOUR 3:PRINT CHR$(7) ' ' TAB(10);
"Another go? Y or N ";
250 REPEAT:G$=GET$:UNTIL G$="Y" OR G$="N"
260 IF G$="Y" THEN RUN
270 CLS:PRINT "Bye for now.":@%=10:END
```

```

350 UNTIL A(Z,X)<>0 OR Z>N
360 IF Z>N THEN PRINT '"No solution!
!"':RETURN
370 IF Z<>X THEN R=1/A(Z,X):I=X:K=Z:P
ROCRoWAdd
380 IF A(X,X)<>1 THEN R=1/A(X,X):I=X:
PROCRoWTimes
390 FOR I=1 TO N
400 IF I=X THEN I=I+1
410 IF I<=N:IF A(I,X)<>0 THEN R=-A(I
,X):K=X:PROCRoWAdd
420 NEXT
430 UNTIL X>=N
440 PRINT '"The solution:"'
450 FOR I=1 TO N:PRINT B(I),;:NEXT:PRI
NT
460 RETURN
500 REM Row operation add R times row
K to row I
510 DEF PROCRoWAdd
520 FOR J=1 TO N
530 A(I,J)=A(I,J)+R*A(K,J)
540 NEXT
550 B(I)=B(I)+R*B(K)
560 ENDPROC
570 REM Row operation multiply row I b
y R
580 DEF PROCRoWTimes
590 FOR J=1 TO N
600 A(I,J)=R*A(I,J)
610 NEXT
620 B(I)=R*B(I)
630 ENDPROC

```

Essential Maths on the BBC and Electron Computers

RUN

Simultaneous equations

This program solves a set of N simultaneous equations in N unknowns.

Enter the number of equations.

Value of N? 3

Enter the coefficients, term by term.

Row 1

Column 1 ?1

Column 2 ?2

Column 3 ?3

Right hand term ?4

Row 2

Column 1 ?-2

Column 2 ?3

Column 3 ?-1

Right hand term ?4

Row 3

Column 1 ?3

Column 2 ?2

Column 3 ?6

Right hand term ?4

The matrix of coefficients:

1	2	3	4
-2	3	-1	4

3 2 6 4

Calculating ***

The solution:

-12 -4 8

Another go? Y or N

