2.13 MATRIX ALGEBRA AND RELATED FUNCTIONS

Matrix notation is widely used in algebra, as it provides a compact form of expression for systems of similarly structured equations. Operations with matrices look surprisingly like ordinary algebraic operations, but in particular multiplication of matrices is more complex. Excel contains useful matrix functions, in the Math category, which require a little background knowledge of how matrices behave to get full benefit from them. The following sections explain matrix notation, and describe the operations of transposing, adding, multiplying and inverting matrices. The examples illustrating these operations are in the MatDef sheet of the AMFEXCEL workbook. If you are conversant with matrices, you may wish to jump straight to the summary of matrix functions (section 2.13.7).

2.13.1 Introduction to Matrices

In algebra, rectangular arrays of numbers are referred to as matrices. A single column matrix is usually called a column vector; similarly a single row matrix is called a row vector. In Excel, rectangular blocks of cells are called arrays. All the following blocks of numbers can be considered as matrices:

$$\begin{vmatrix} 2 \\ 4 \end{vmatrix} \qquad \begin{vmatrix} 6 & 7 \end{vmatrix} \qquad \begin{vmatrix} -3 & 2 & 7 \\ 2 & 20 & 19 \\ 7 & 9 & 21 \end{vmatrix} \qquad \begin{vmatrix} -3 & 2 & 7 \\ 2 & 20 & 19 \\ 7 & 9 & 21 \\ 0 & 13 & 3 \end{vmatrix}$$

where the brackets | | are merely notational. Calling these matrices **x**, **y**, **A**, and **B** respectively, **x** is a column vector and **y** a row vector. Matrix **A** has three rows and three columns and hence is a square matrix. **B** is not square since it has four rows and three columns, i.e. **B** is a 4 by 3 matrix. The numbers of rows, r, and of columns, c, give the dimensions of a matrix sometimes written as $(r \times c)$. For example, if:

$$\mathbf{x} = \begin{vmatrix} 2 \\ 4 \end{vmatrix} \text{ and } \mathbf{y} = \begin{vmatrix} 6 & 7 \end{vmatrix}$$

then **x** has dimensions (2×1) whereas **y** has dimensions (1×2).

2.13.2 Transposing a Matrix

Transposition of a matrix converts rows into columns (and vice versa). Clearly the transpose of column vector **x** will be a row vector, denoted as $\mathbf{x}^\mathbf{T}$. The spreadsheet extract in Figure 2.32 shows the transposes of column vector **x** and row vector **y**.

The TRANSPOSE function applied to the cells of an array returns its transpose. For example, the transpose of the 2 by 1 vector **x** in cells C4:C5 will have dimensions (1×2). To use the TRANSPOSE function, select the cell range I4:J4 and key in the formula:

=TRANSPOSE(C4:C5)

finishing with Ctrl+Shift+Enter pressed simultaneously. The result is shown in Figure 2.32.

	A	B	C	D	E	F	G	H	I	J	K
1	**Array Manipulation:**										
2									**Transposes**		
3	**array:**	**dim**					**array:**	**dim**			
4	x	(2x1)	2				x^T	(1x2)	2	4	
5			4								
6											
7	y	(1x2)	6	7			y^T	(2x1)	6		
8									7		
9		Array Multiplication									
10	xy	(2x2)	12	14			$(xy)^T$	(2x2)	12	24	
11			24	28					14	28	
12											
13	yx	(1x1)	40				$(yx)^T$	(1x1)	40		

Figure 2.32 Matrix operations illustrated in the MatDef sheet

2.13.3 Adding Matrices

Adding two matrices involves adding their corresponding entries. For this to make sense, the arrays being added must have the same dimensions. Whereas **x** and **y** cannot be added, **x** and $\mathbf{y}^\mathbf{T}$ do have the same dimensions, 2 by 1, and therefore they can be added, the result being:

$$\mathbf{x}+\mathbf{y}^\mathbf{T} = \begin{vmatrix} 2 \\ 4 \end{vmatrix} + \begin{vmatrix} 6 \\ 7 \end{vmatrix} = \begin{vmatrix} 8 \\ 11 \end{vmatrix} = \mathbf{z} \text{ say}$$

To multiply vector **y** by 10 say, every entry of **y** is multiplied by 10. Thus:

$$10\mathbf{y} = 10 * |6 \quad 7| = |60 \quad 70|$$

This is comparable to adding **y** to itself 10 times.

2.13.4 Multiplying Matrices

For two matrices to be multiplied they have to have a common dimension, that is, the number of columns for one must equal the number of rows for the other. The shorthand expression for this is ‘dimensional conformity’. For the product **xy** the columns of **x** must match the rows of **y**, (2×1) times (1×2), resulting in a (2×2) matrix as output.

In Figure 2.32, the product **xy** in cells C10:D11 has elements calculated from:

$$\begin{vmatrix} 2 \\ 4 \end{vmatrix} \quad |6 \quad 7| = \begin{vmatrix} 2*6 & 2*7 \\ 4*6 & 4*7 \end{vmatrix} = \begin{vmatrix} 12 & 14 \\ 24 & 28 \end{vmatrix}$$

i.e. the row 1, column 1 element of product **xy** comes from multiplying the individual elements of row 1 of **x** by the elements of column 1 of **y**, etc.

In contrast, the product **yx** has dimensions (1×2) times (2×1), that is (1×1), i.e. it consists of a single element. Looking at product **yx** in cell C13, this element is computed as:

$$|6 \quad 7| \quad \begin{vmatrix} 2 \\ 4 \end{vmatrix} = |6*2+7*4| = |40|$$

These results demonstrate that for matrices, **xy** is not the same as **yx**. The order of multiplication is critical.

The MMULT array function returns the product of two matrices, called array1 and array2. So to get the elements of the (2×2) matrix product **xy**, select the 2 by 2 cell range, C10:D11 and key in or use the Paste Function button and build the expression in the Formula palette:

=MMULT(C4:C5,C7:D7)

remembering to enter it with Ctrl+Shift+Enter.

If ranges C4:C5 and C7:D7 are named **x** and **y** respectively, then the formula to be keyed in simplifies to:

=MMULT(**x**,**y**)

Consider two more arrays:

$$\mathbf{C} = \begin{vmatrix} 12 & 4 \\ 3 & 13 \end{vmatrix} \text{ and } \mathbf{D} = \begin{vmatrix} 16 & 19 & -2 \\ 5 & 12 & 14 \end{vmatrix}$$

The dimensions of **C** and **D** are (2×2) and (2×3) respectively, so since the number of columns in **C** is matched by the number of rows in **D**, the product **CD** can be obtained, its dimensions being (2×3). So:

$$\mathbf{CD} = \begin{vmatrix} (12*16+4*5) & (12*19+4*12) & (-12*2+4*14) \\ (3*16+13*5) & (3*19+13*12) & (-3*2+13*14) \end{vmatrix} = \begin{vmatrix} 212 & 276 & 32 \\ 113 & 213 & 176 \end{vmatrix}$$

However, the product **DC** cannot be formed because of incompatible dimensions (the number of columns in **D** does not equal the number of rows in **C**). In general, the multiplication of matrices is not commutative, so that usually **CD** ≠ **DC**, as in this case.

If **C** and **D** are the names of the 2 by 2 and the 2 by 3 arrays respectively, then the cell formula:

=MMULT(**C**,**D**)

will produce the elements of the 2 by 3 product array.

2.13.5 Matrix Inversion

A square matrix **I** with ones for all its diagonal entries and zeros for all its off-diagonal elements is called an identity matrix. Thus:

$$\mathbf{I} = \begin{vmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1 \end{vmatrix} \quad \text{is an identity matrix}$$

Suppose **D** is the (2×3) matrix used above, and **I** is the (2×2) identity matrix, then:

$$\mathbf{ID} = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} * \begin{vmatrix} 16 & 19 & -2 \\ 5 & 12 & 14 \end{vmatrix} = \begin{vmatrix} 16 & 19 & -2 \\ 5 & 12 & 14 \end{vmatrix} = \mathbf{D}$$

Multiplying any matrix by an identity matrix of appropriate dimension has no effect on the original matrix (and is therefore similar to multiplying by one).

Now suppose $\mathbf{A}$ is a square matrix of dimension n, that is an n by n matrix. Then, the square matrix $\mathbf{A}^{-1}$ (also of dimension n) is called the inverse of $\mathbf{A}$ if:

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

For example, if:

$$\mathbf{A} = \begin{vmatrix} -3 & 2 & 7 \\ 2 & 20 & 19 \\ 7 & 9 & 21 \end{vmatrix} \text{ then } \mathbf{A}^{-1} = \begin{vmatrix} -0.175 & -0.015 & 0.072 \\ -0.064 & 0.079 & -0.050 \\ 0.086 & -0.029 & 0.045 \end{vmatrix}$$

and

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

Finding the inverse of a matrix can be a lot of work. Fortunately, the MINVERSE function does this for us. For example, to get the inverse of matrix $\mathbf{A}$ shown in the spreadsheet extract in Figure 2.33, select the 3 by 3 cell range I17:K19 and enter the array formula:

=MINVERSE(C17:E19)

You can check that the result is the inverse of $\mathbf{A}$ by performing the matrix multiplication $\mathbf{A}\mathbf{A}^{-1}$.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
16	**array:**	dim					**array:**	dim									
17	A	(3x3)	-3	2	7		A^{-1}	(3x3)	-0.175	-0.015	0.072		AA^{-1}	1.00	0.00	0.00	
18			2	20	19				-0.064	0.079	-0.050			0.00	1.00	0.00	
19			7	9	21				0.086	-0.029	0.045			0.00	0.00	1.00	
20																	
21	b	(3x1)	20				$x = A^{-1}b$	(3x1)	-3.4355								
22			-5						-1.6772								
23			0						1.8640								
24																	

Figure 2.33 Matrix inversion shown in the MatDef sheet

2.13.6 Solving Systems of Simultaneous Linear Equations

One use for the inverse of a matrix is in solving a set of equations such as the following:

$$-3x_1 + 2x_2 + 7x_3 = 20$$

$$2x_1 + 20x_2 + 19x_3 = -5$$

$$7x_1 + 9x_2 + 21x_3 = 0$$

These can be written in matrix notation as $\mathbf{Ax} = \mathbf{b}$ where:

$$\mathbf{A} = \begin{vmatrix} -3 & 2 & 7 \\ 2 & 20 & 19 \\ 7 & 9 & 21 \end{vmatrix} \quad \mathbf{b} = \begin{vmatrix} 20 \\ -5 \\ 0 \end{vmatrix} \text{ and } \mathbf{x} = \begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix}$$

The solution is given by premultiplying both sides of the equation by the inverse of **A**:

$$\mathbf{A}^{-1}\mathbf{A}\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}, \text{ so } \mathbf{I}\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \quad \text{i.e.} \quad \mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

In Figure 2.33, the solution vector **x** is obtained from the matrix multiplication function in cell range I21:I23 in the form:

=MMULT(I17:K19,C21:C23)

Not every system of linear equations has a solution, and in special cases there may be many solutions. The set $\mathbf{Ax} = \mathbf{b}$ has a unique solution only if the matrix **A** is square and has an inverse $\mathbf{A}^{-1}$. In general, the solution is given by $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.

2.13.7 Summary of Excel's Matrix Functions

In summary, Excel has functions to transpose matrices, to multiply matrices and to invert square matrices. The relevant functions are these:

TRANSPOSE(array)	returns the transpose of an array
MMULT(array1, array2)	returns the matrix product of two arrays
MINVERSE(array)	returns the matrix inverse of an array

Because these functions produce arrays as outputs, the size of the resulting array must be assessed in advance. Having 'selected' an appropriately sized range of cells, the formula is keyed in (or obtained via the Paste Function button and built in the Formula palette). It is entered in the selected cell range with the combination of keys Ctrl+Shift+Enter (instead of simply Enter). If this fails, keep the output range of cells 'selected', press the Edit key (F2), edit the formula if necessary, then press Ctrl+Shift+Enter again.

To consolidate, try the matrix exercises in the sheet MatExs.

We make extensive use of the matrix functions in the Equities part of the book, both for calculations in the spreadsheet and as part of VBA user-defined functions.

SUMMARY

Excel has an extensive range of functions and procedures. These include mathematical, statistical and lookup functions, as well as much-used procedures such as setting up Data Tables and displaying results in XY charts.

Access to the functions is handled through the Paste Function button and the function inputs specified on the Formula palette. The use of range names simplifies the specification of cell ranges, especially when the ranges are sizeable. Range names can be used on the Formula palette.

Facilities on the Auditing toolbar, in particular the Trace Precedents, Trace Dependents and Remove Arrows buttons are invaluable in examining formula cells.

It helps to be familiar with the range of Excel functions because they can easily be incorporated into user-defined functions, economising on the amount of VBA code that has to be written.

Care is required in using array functions. It helps to decide in advance the size of the cell range appropriate for the array results. Then having selected the correct cell range, the formula is entered with the keystroke combination Ctrl+Shift+Enter.