Gerald's Column

by Gerald Fitton

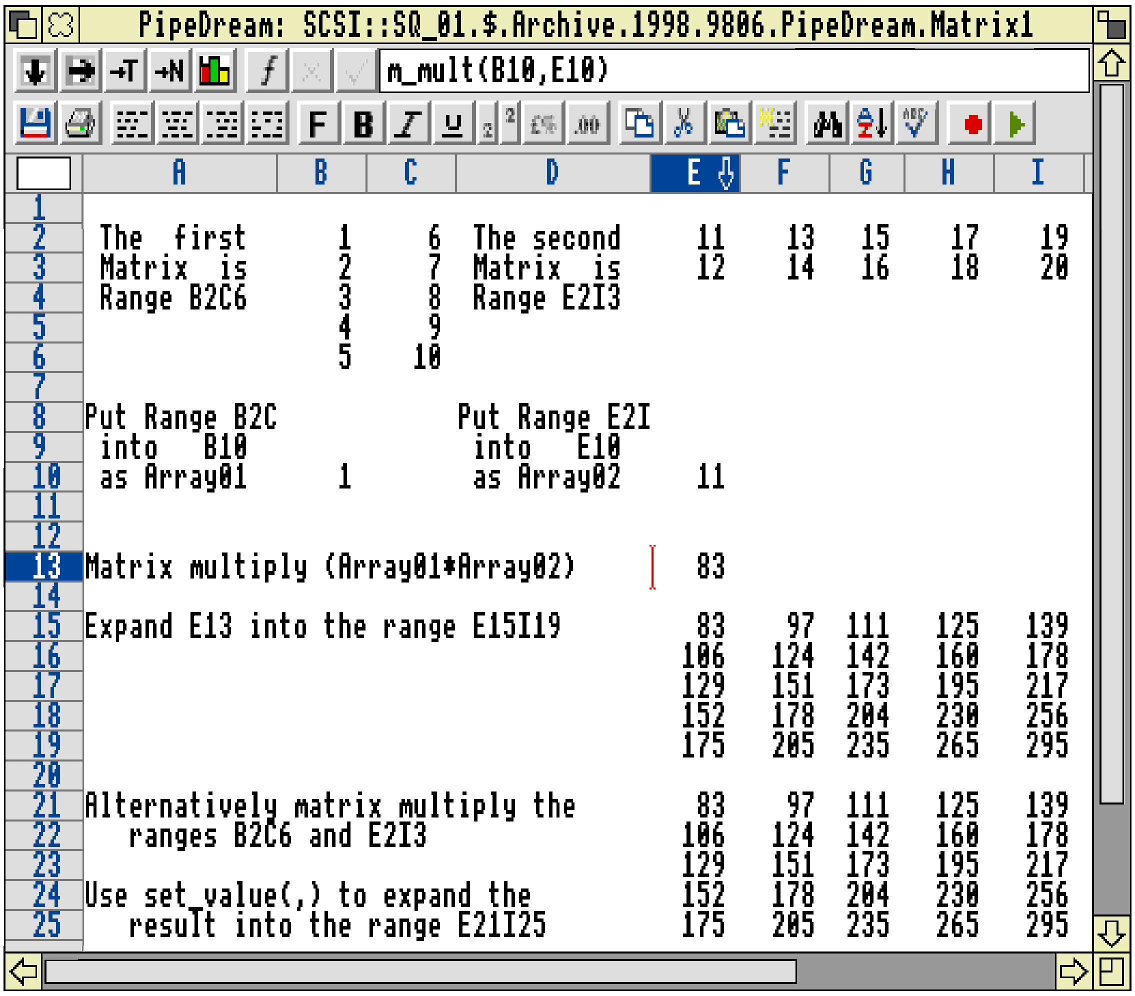
There are many different 'group of numbers' concepts in Mathematics. In last month's article I gave some examples of Array Arithmetic. This month I shall describe one aspect of Matrix arithmetic, Matrix multiplication – we shall deal with the inverse of Matrix multiplication another day.

Matrix Addition and Subtraction

The rules for Matrix addition and subtraction are exactly the same as those of Array arithmetic.

Matrix Multiplication

As we have seen, during Array multiplication each element in the first array is multiplied by the corresponding element of the second array. Both arrays must be the same 'shape'. The rules for Matrix multiplication are different.



PipeDream: SCSI::SQ_01.$.Archive.1998.9806.PipeDream.Matrix1

m_mult(B10,E10)

	A	B	C	D	E	F	G	H	I
1									
2	The first	1	6	The second	11	13	15	17	19
3	Matrix is	2	7	Matrix is	12	14	16	18	20
4	Range B2C6	3	8	Range E2I3					
5		4	9						
6		5	10						
7									
8	Put Range B2C			Put Range E2I					
9	into B10			into E10					
10	as Array01	1		as Array02	11				
11									
12									
13	Matrix multiply (Array01*Array02)				83				
14									
15	Expand E13 into the range E15I19				83	97	111	125	139
16					106	124	142	160	178
17					129	151	173	195	217
18					152	178	204	230	256
19					175	205	235	265	295
20									
21	Alternatively matrix multiply the				83	97	111	125	139
22	ranges B2C6 and E2I3				106	124	142	160	178
23					129	151	173	195	217
24	Use set_value(,) to expand the				152	178	204	230	256
25	result into the range E21I25				175	205	235	265	295

The screenshot above is of a file [Matrix1] which you will find in all the usual places. The input focus is cell E13 and the function m_mult(,) shows in the formula line. This function, m_mult(,) is not a custom function but is one built into PipeDream. Nearly all the spreadsheet packages I know of, including the Windows 95 spreadsheet Excel, have a matrix multiply function built into the package.

In order to multiply two matrices together first you need a couple of matrices! These can not be any old matrices but they have to make up a compatible pair. By this I mean that the first matrix must have the same number of columns as the second matrix has rows. Contrast this with Array multiplication where both arrays have to be of the same shape.

Have another look at the screenshot. The first matrix is the range B2C6; it has two columns. The second matrix is the range E2I3; it has two rows. The number of rows of the first matrix and the number of columns in the second matrix are both unimportant; it is just bad luck that I have chosen five for both!

Cell B10 contains the formula B2C6. This compresses the range into the single cell B10. Similarly the second matrix, the range E2I3, is compressed into E10.

The value returned by the function m_mult(B10,E10) is another matrix, similarly compressed into cell E13, which I have expanded into the range E15I19 with the function set_value(E15I19,E13).

Before you go onto the next part of my article please make sure that you know where everything is. The two matrices, B2C6 and E2I3, are multiplied together; the result of the matrix multiplication is in E15I19. Rows 8 to 13 are only a means to this end.

In matrix multiplication you choose a row from the first matrix and a column from the second. Multiply each element of a row from the first matrix with the corresponding element from a column of the second matrix. Add together the results of the multiplication. Here are a few examples from the spreadsheet in the screenshot so that you can see how it works:

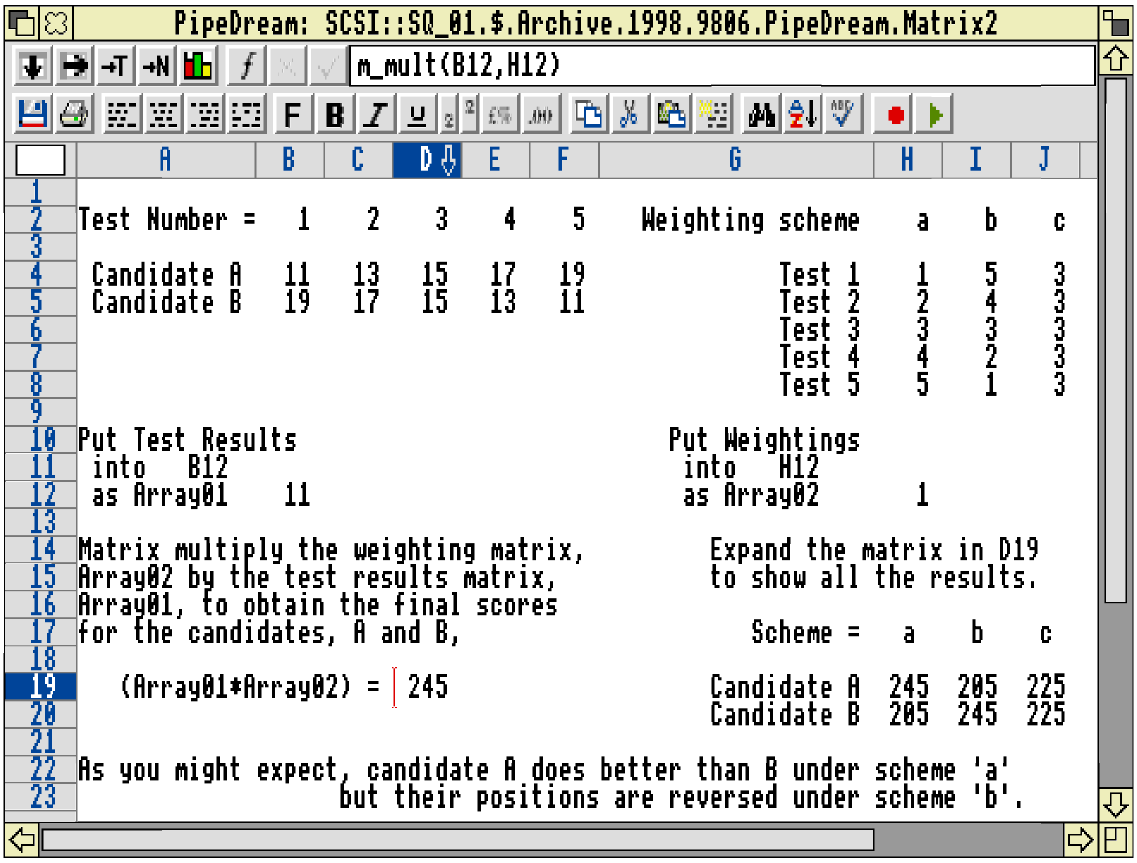
```
E15 = (1*11) + (6*12) =  83
F15 = (1*13) + (6*14) =  97
G15 = (1*15) + (6*16) = 111
E16 = (2*11) + (7*12) = 106
F16 = (2*13) + (7*14) = 124
G16 = (2*15) + (7*16) = 142
E17 = (3*11) + (8*12) = 129
F17 = (3*13) + (8*14) = 151
G17 = (3*15) + (8*16) = 173
```

Try to see this as ‘going along the rows (of the first matrix) and down the columns (of the second matrix)’. Each row/column combination provides one number in the output matrix. If you use your left index finger to run along the top row of the first matrix and the index finger of your right hand to run down the column of the second matrix then it will help you multiply together the correct pair of numbers.

The ‘Best’ Candidate

From this first example you will see that matrix multiplication is a mixture of multiplication and addition; it is certainly not element by element multiplication as it is for Array multiplication. If you are not familiar with matrix multiplication then you might find it difficult to see any use for it and, as a consequence, the logic behind it will seem obscure. Perhaps a further example will help you see the logic and one of its uses.

For some decades now it has been a popular concept in educational circles to believe that everybody is good at something. Whilst I believe in this general principle I am equally sure that there are a few people who are ‘bad’ at almost everything and a few who are ‘good’ at almost everything they try. These are the exceptions. Generally, when we have a job vacancy we try to find the candidate most suitable for the job. If a candidate is unsuitable for one job they might be best suited to another.



```
PipeDream: SCSI::SQ_01.$.Archive.1998.9806.PipeDream.Matrix2
m_mult(B12,H12)

   A              B    C    D    E    F         G           H    I    J
1
2  Test Number =  1    2    3    4    5    Weighting scheme  a    b    c
3
4   Candidate A  11   13   15   17   19              Test 1  1    5    3
5   Candidate B  19   17   15   13   11              Test 2  2    4    3
6                                                    Test 3  3    3    3
7                                                    Test 4  4    2    3
8                                                    Test 5  5    1    3
9
10 Put Test Results                             Put Weightings
11  into   B12                                   into   H12
12  as Array01    11                             as Array02     1
13
14 Matrix multiply the weighting matrix,           Expand the matrix in D19
15 Array02 by the test results matrix,             to show all the results.
16 Array01, to obtain the final scores
17 for the candidates, A and B,                       Scheme =  a    b    c
18
19    (Array01*Array02) = 245                      Candidate A 245  205  225
20                                                 Candidate B 205  245  225
21
22 As you might expect, candidate A does better than B under scheme 'a'
23                    but their positions are reversed under scheme 'b'.
```

In the file [Matrix2] you will find an example of applying matrix multiplication to the problem of assessing two candidates called A and B. Each candidate takes five tests numbered 1 to 5; these are marked fairly. Let us suppose that the low numbered tests require academic skills and the higher numbered tests require practical skills. Now, for some jobs academic skills are more important and for other jobs it is the other way around. Please don’t write to me criticising me for this oversimplification – I know that nearly all jobs require both sets of skills.

We shall assume that we have three adjudicators, a, b and c, who cannot agree on the relative importance of the five tests so they compromise with three different ‘weighting’ schemes. Scheme ‘a’ concentrates on practical skills, scheme ‘b’ on academic skills and scheme ‘c’ on neither.

Which candidate is best under which scheme?

Matrix multiplication can be used to analyse the assessment results of the two candidates. As you might expect, candidate A does better than B under scheme ‘a’ but their positions are reversed under scheme ‘b’ whereas, under scheme ‘c’ both appear to be of equal merit!

You might like to try other test results and weighting schemes. For example, you might have students who do well in the sciences but badly in humanities or perhaps you have employees who are skilled technicians but are not happy with paperwork. Different grading schemes which weigh sciences more heavily than humanities or which select for promotion (or training) employees who are good at paperwork will give different overall ranking to the students or employees.

As a general comment I would say to you “Do not set a test unless you know exactly what it is that you are testing!” I have run many tests in my time (and I have sat many tests too). I have to confess that often it has been only with hindsight that I have realised what sort of thing I have been testing –often it has not been what I wanted to test.

Matrix Division

The rules of Array division are straightforward. Each element of the first array is divided by the corresponding element of the second array to return an element of the output array.

Matrix division does not exist! At some future date I shall tell you about another concept in mathematics which does a similar job. It is called the ‘Inverse matrix’. Multiplying by this Inverse matrix yields results similar to what you might have expected of Matrix division if it existed.

All those spreadsheets which support Matrix arithmetic have an Inverse matrix function. This Inverse matrix function can be used to solve simultaneous equations.

Groups of Numbers

Both Array arithmetic and Matrix arithmetic operate with groups of numbers but in different ways. There is nothing unique about either. I have been asked if I will give some examples of Quaternion arithmetic – well, maybe at some time in the future. I do not know of any spreadsheet which contains Quaternion arithmetic functions. If you have an example of Quaternion arithmetic to send me then I shall be most interested. I do not need it to be in spreadsheet format.

Complex Arithmetic follows yet another set of rules. We shall look at them next month.

Finally

You can contact me at Abacus Training. The address is at the back of Archive.