

Gerald's Column *by Gerald Fitton*

In many of the letters I've received since the last Archive was published I've been asked to expand greatly on the machine code programs which I used last month to do sums to nineteen significant figures on the Archimedes. This hidden ability of the Archimedes to handle numbers of a greater range and to a greater precision than the ubiquitous Double Precision (used in all the spreadsheets I know of) has come as a great surprise to many of you. An even greater surprise to all (including me) has been that the earlier versions of the Floating Point Emulator can output numbers to a greater precision than the later ones.

I considered interrupting my series on Numbers to expand on my brief introduction to Extended Double Precision but, when I came to start on the article, I realised that I still had too much work to do before going to print.

My other large pile of mail anticipates my forthcoming article on Complex numbers. It seems to me that many of my correspondents regard Complex numbers as nothing more than a pair of Real numbers rather than a different sort of number in its own right. I want to dispel this notion, hence I've decided to digress and tell you something about groups of numbers, in particular Arrays and Matrices and how spreadsheets usually handle them.

Groups of Numbers

There are many different 'group of numbers' concepts in Mathematics.

The most common these days is Matrices. The greatest single use to which these are put is the solution of simultaneous equations – but this is not their only use. In this context I must express my thanks to Bob Ardler who sent me some BASIC programs which solve sets of simultaneous equations – Bob does not take too much personal credit for these programs but congratulates the pupils he taught! After a little persuasion from me Bob converted the BASIC equations to a PipeDream and then a Fireworkz spreadsheet. My problem is that Bob didn't send me an article describing exactly how to use the routines he sent me because he assumed (quite wrongly) that I was so brilliant that I could see exactly what to do with them! I have asked for his help.

There will be many of you who know about Vectors. A Vector is a group of numbers which contains information about magnitude and direction. An example is the velocity of an object such as a projectile. Not only has the projectile a speed (a positive Real number) but also a direction. Four dimensional Vectors are often used to describe and to solve problems in the field of Special Relativity.

At this point I can not resist mentioning one of my heroes, Oliver Heaviside (1850–1925) who died as a penniless recluse in Torquay (Devon, England). When I was at school (in Tavistock, Devon) there was a layer in the sky named after him. Now the Heaviside Layer is called the ionosphere – he predicted its existence and postulated that it would reflect radio waves and thus enable long range radio communication. It was Oliver Heaviside who developed Vectors as a simplification of a mathematical concept called Quaternions (see below). It was he who converted Maxwell's electromagnetic equations to Vector

format and developed the mathematics of Vector Algebra by introducing such Vector operators as *Div*, *Curl*, and *Del*. These operators are differential operators (as in Calculus) and so they do not appear in any but the most specialised digital computer spreadsheets. However during my time at Vickers I constructed analogue computer circuitry which implemented not only these differential Vector operators but also other Vector operators which carry out integration.

Oliver Heaviside used Complex number functions to study transmission line theory. He was a telegraphist by profession (a telegraphist was a morse code operator in the days before speech could be transmitted by wire) and he suggested that by introducing more inductance into the telegraph (now telephone) lines the attenuation would be reduced and the need for so many powerful amplifiers would be removed. The Engineer-in-Chief at the Post Office (the PO ran the telegraph system at that time – BT did not exist) called Sir William Henry Preece (1834–1913) backed an experiment suggest by Heaviside which worked! I hope to use Heaviside's transmission line equations as one example of a Complex number spreadsheet in a later article.

Quaternions were used extensively to solve many problems in three dimensional geometry. Essentially they are groups of four numbers. Quaternions were discovered by Sir William Rowan Hamilton (1805–65) an Irish Mathematician born in Dublin. He later became a Professor of Astronomy at Dublin University. Quaternions were taught in all University Mathematics courses (even in the 1940s and early 1950s) as the standard method of solving what would now be classed as Vector Algebra problems. Vectors are much easier to understand than Quaternions and are taught now as part of the A Level Maths Course.

There are many other mathematical 'Algebras' which handle what I have called 'groups of numbers' in many different ways. The variety is far too numerous for me to list so I have restricted myself to the most popular. All these Algebras have one thing in common, every number in the 'group of numbers' can be a Complex number. Using Complex numbers rather than limiting the type to Real numbers often simplifies rather than complicates the Algebra of number groups.

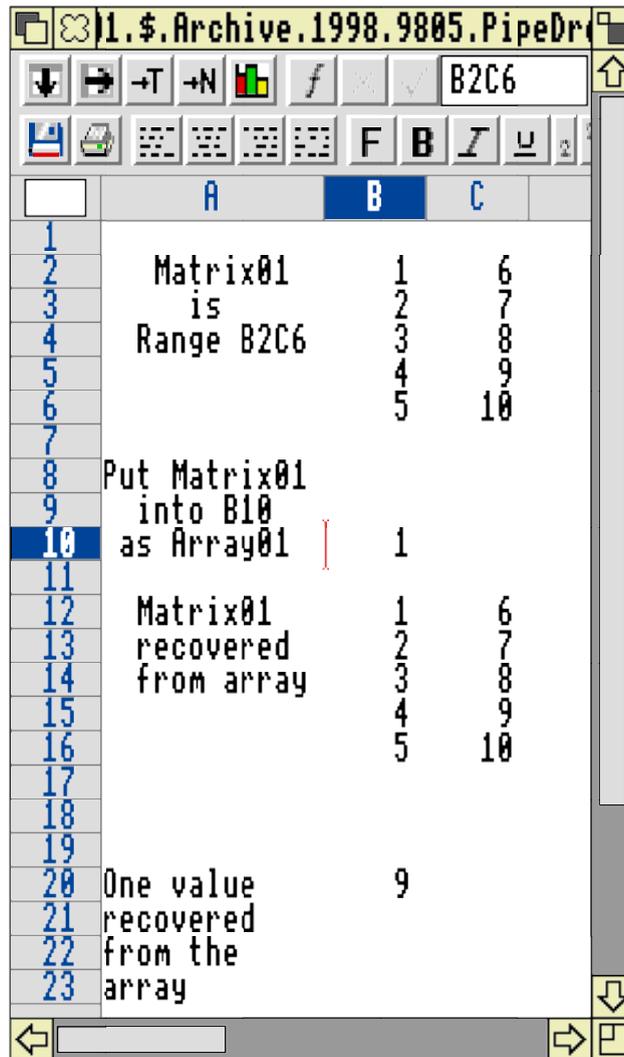
Range – Array – Matrix

In the paragraphs above I haven't mentioned Arrays because (at least as far as I know) they are not a concept introduced by Pure mathematicians but a concept invented by programmers of spreadsheets. Arrays are more general than 'groups of numbers' because they can include textual information as well as numbers.

User Guides for different spreadsheet packages use the word Arrays to mean slightly different things and you must read the fine print to determine what is an Array and what is not. In what follows (unless I say otherwise) I shall use the word Range to mean a block of cells, the word Array to mean one single cell containing a block of data and the word Matrix to mean either a Range or an Array.

From a Range to an Array

Have a look at the screenshot below of the PipeDream file [Array01]. The range [Array01]B2C6 contains the ten numbers (1 to 10) in two columns and five rows.

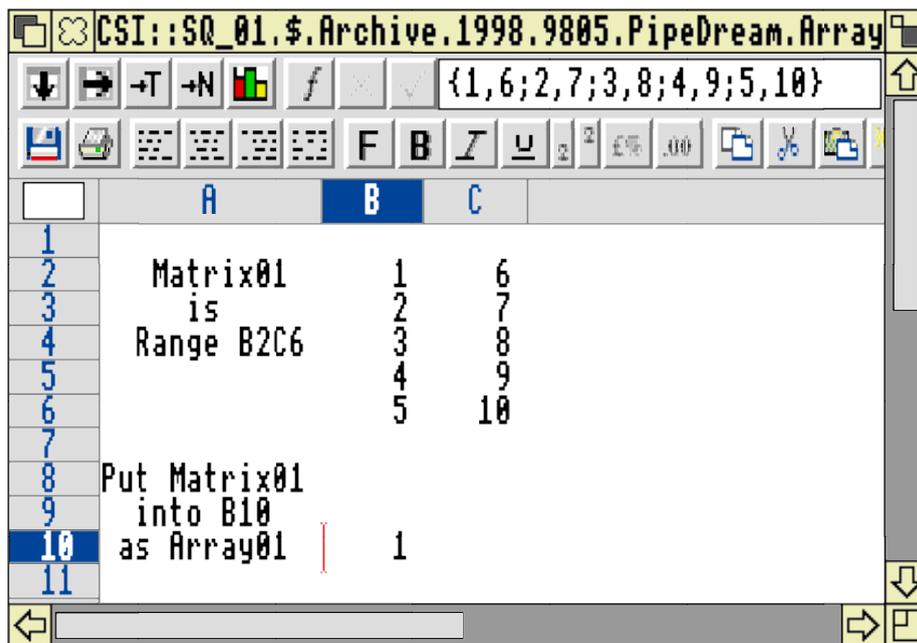


The screenshot of the file [Array01] shows what I believe to be the easiest way to compress a Range into to a single cell. You will see that the cursor is in cell B10 and the formula line shows B2C6, the Range. In the spreadsheet itself in B10 you will see only the top left number; the remaining nine numbers are in there but hidden from view.

Taking a Snapshot of an Array

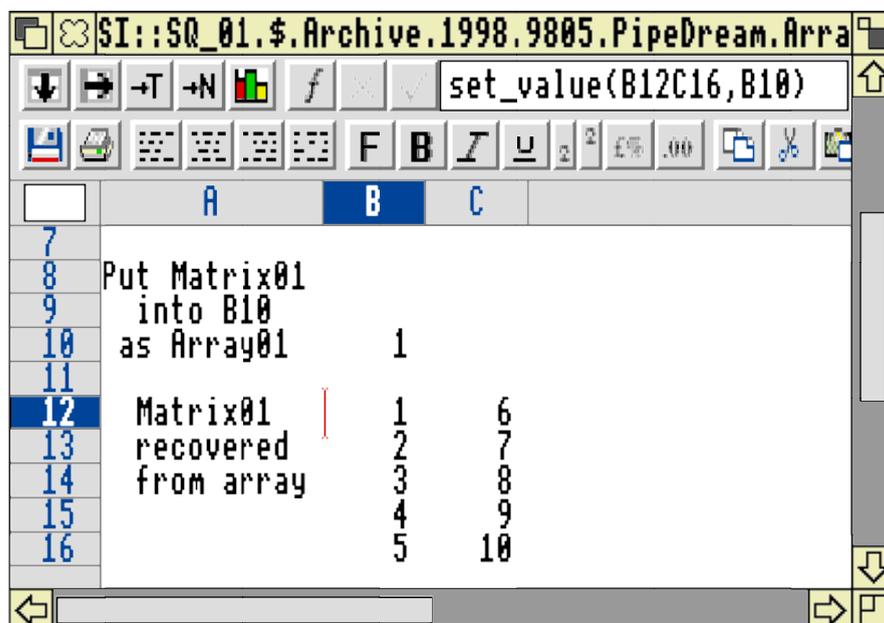
In my Sheets series of a couple of months ago I explained how to convert a formula to a value; in PipeDream the cursor is placed in B10 and the command <Ctrl BSS> (Block SnapShot) is executed.

In the screenshot below I have converted the formula in [Array01]B10 to a value. Have a look at the formula line and you'll see that curly brackets, {}, surround the Array and that commas and semicolons separate the individual values. This is common practice in most Archimedes and Windows spreadsheets which support Arrays.



Transferring Values from an Array to a Range

Mathematicians always want to ‘do it backwards’ – to be able to execute the reverse of any process. In PipeDream the easiest way I have found of transferring the whole of an Array to a Range is to use the function `set_value(destination,source)`. Some spreadsheets do not contain this most useful function – the `index(,)` function has to be used instead.



Recovering a Single Value from an Array

Sometimes you may want to recover only one value from an Array. In my view the `index(,,)` function is the best way of doing this. In slot [Array01]B20 I have used the index function, `index(B10,2,4)`, to recover the value from column 2, row 4 of [Array01]B10.

Array Arithmetic

To the great surprise of those who come across it for the first time the Array Arithmetic of most spreadsheets does not follow the rules of Matrix Arithmetic.

Have a look at the screenshot below.

The screenshot shows a spreadsheet window with the following content:

	A	B	C	D	E	F
1						
2	Matrix01	1	6	Matrix02	11	16
3	is	2	7	is	12	17
4	Range B2C6	3	8	Range E2F6	13	18
5		4	9		14	19
6		5	10		15	20
7						
8	Put Matrix01			Put Matrix02		
9	into B10			into E10		
10	as Array01	1		as Array02	11	
11						
12						
13						
14						
15						
16						
17						
18	Adding or multiplying arrays is executed element by element					
19						
20	Add the arrays	12		Multiply arrays	11	
21						
22	Expand to	12	22	Expand to	11	96
23	range B22C26	14	24	range E22F26	24	119
24		16	26		39	144
25		18	28		56	171
26		20	30		75	200

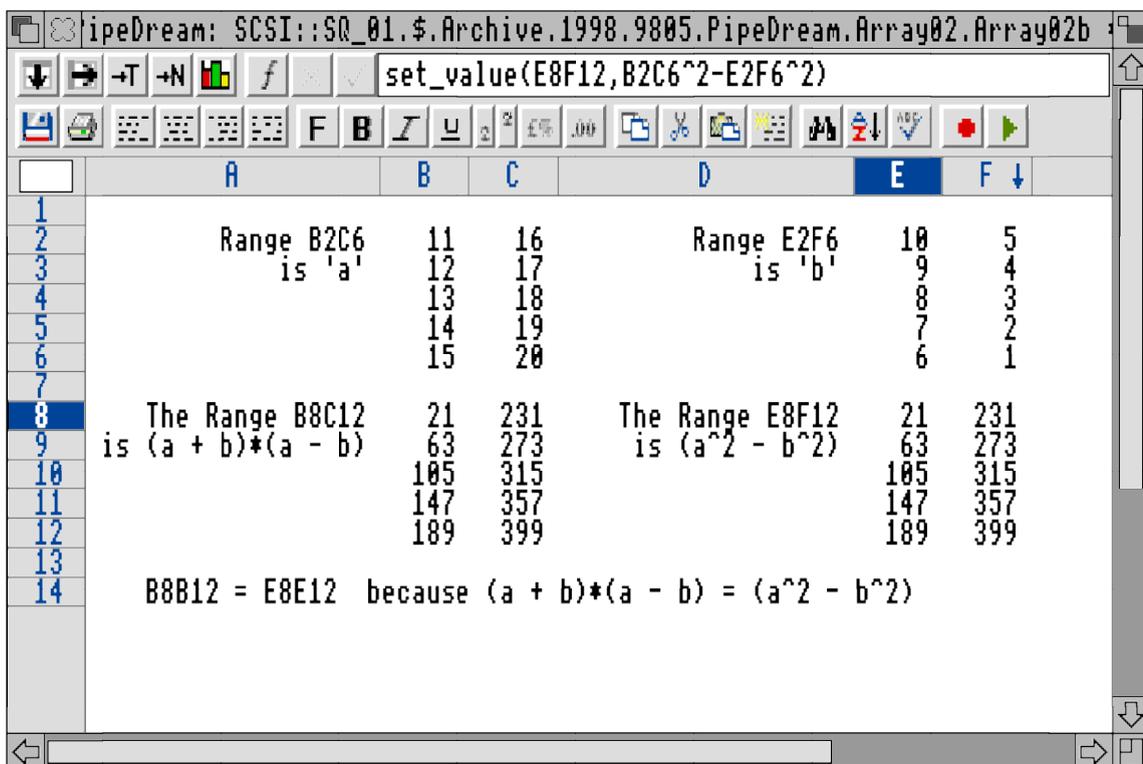
I have created two separate Arrays stored in the cells B10 and and E10. The cursor is in cell E20. Have a look at the formula line and you'll see that the function to be evaluated is `B10*E10`. Forget about Matrix multiplication, concentrate on the expanded versions of the Arrays which are in the Ranges B2C6, E2F6 and E22F26 and you'll see the result of the

calculation. What has happened is that each element of the first Array (Matrix01) has been multiplied by the corresponding element of the second Array (Matrix02) and the result appears in the corresponding element of the third Array (eg the value in F26 is C6*F6).

Array Arithmetic on Ranges

Those spreadsheets which support Array Arithmetic usually allow Array Arithmetic to be applied to Ranges.

Have a look at the formula in [Array02b]F8 in the screenshot below and you'll see that I have used `set_value(destination,source)` to 'push' an Array formula into the Range E8F12. The Array formula in algebraic form is $(a^2 - b^2)$ where 'a' is always taken from the Range B2C6 and 'b' is the corresponding value from the Range E2E6. For example the value in F12 is $(C6^2 - F6^2)$ or $(20^2 - 1^2) = 400 - 1 = 399$.



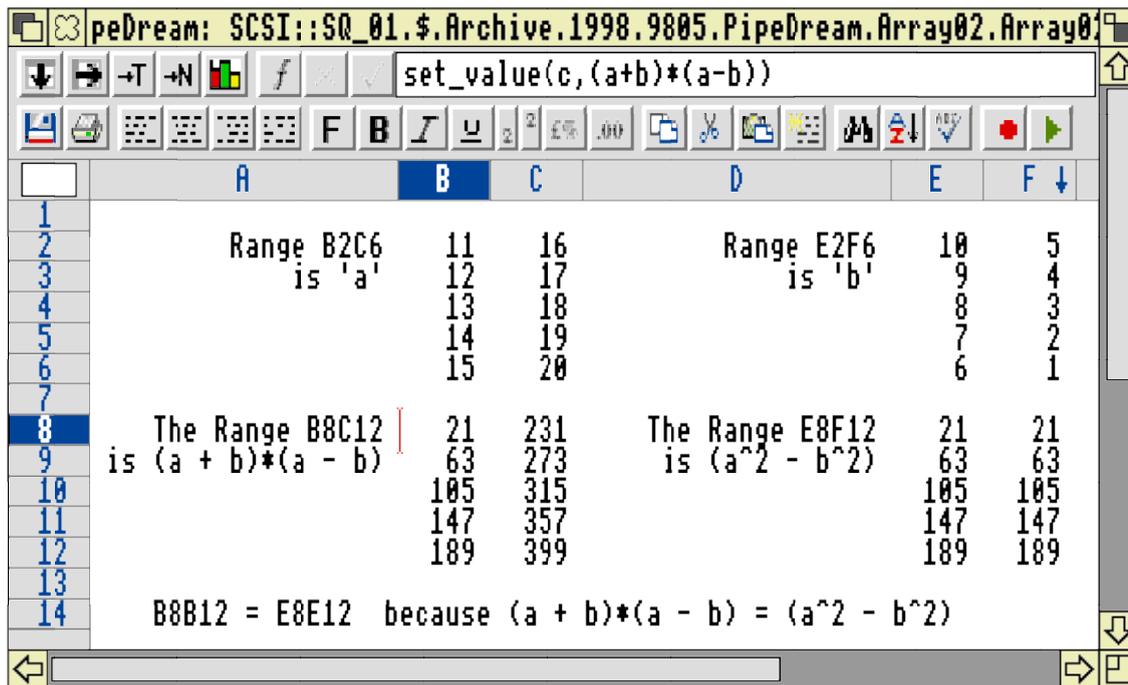
I have used a different `set_value(destination,source)` formula in B8. The Array formula for the Source is $(B2C6 + E2E6)*(B2C6 - E2E6)$ and the `set_value(,)` function 'pushes' the resulting Matrix into the Range B8C12.

Now have a look at the screenshot below.

The cursor is in cell [Array02c]B8. You will see that the formula in the formula line is deceptively simple, `set_value(c,(a+b)*(a-b))`. The letters a, b and c are PipeDream Names with c referring to the Range B8C12.

It is my hope that you will appreciate that the use of Names and Array functions not only reduces the number of formulae in the spreadsheet (this makes it much faster to Load, Save

and execute) but also it makes the spreadsheet easier to read and understand. A spreadsheet which is easy to interpret is one which can more easily be modified, improved and extended should the need arise.



The Array Function facility simplifies the creation and modification of spreadsheets particularly when used in conjunction with Named Ranges or Arrays.

Matrix Arithmetic

I am about at my limit for space in Archive so I shall defer examples of Matrix Arithmetic on 'groups of numbers' until next month.

Groups of Numbers

Part of the reason why I have dealt at length with Array Arithmetic first in preference to Matrix Arithmetic is because those of you who are used to handling 'groups of numbers' may think that 'number groups' which have a meaning exist only in the form of Matrices or you might believe that there is something special about Matrix Arithmetic. As we shall see next month Matrix Arithmetic is useful. My point this month is that it is not unique. Indeed Array Arithmetic has equally valid rules and conventions and, for many people who use spreadsheets, learning Array Arithmetic will prove more useful to them than learning Matrix Arithmetic.

I have no wish to anticipate my discourse about Complex numbers but I can not help adding that Complex Arithmetic follows different rules again from both Array Arithmetic and Matrix Arithmetic. To those of you who have doubts, let me emphasise that although Complex Arithmetic does not follow the same rules as Matrix Arithmetic that is not a reason for implying that Complex numbers are any less 'real' than those groups which do

follow those rules.

I can not resist adding one more screenshot which demonstrates not only an interesting Complex number calculation but which also demonstrates that the inability of a digital computer to do sums accurately is not limited to sums involving single numbers but also to sums on any number groups.

Have a look at the screenshot below.

	A	B
1		
2		
3	The complex number {0,pi}	0
4		
5	The complex function exp(i*pi)	-1
6		
7	A snapshot of exp(i*pi)	-1

The Complex number in B3 is $(i\pi)$. The value of $\exp(i\pi)$ is calculated in B5 and should be -1 (negative one, a Real number with no Imaginary part). However, you'll see from the screenshot that the Imaginary part as calculated by PipeDream is not zero but, instead, a small number with about 14 zeros after the decimal point. The reason why a small number appears in the answer is because PipeDream (as other spreadsheets) can not find the values of functions to perfection but is limited by the accuracy of the Double Precision arithmetic executed by the Floating Point Emulator.

Next Month

Many spreadsheets support Matrix Arithmetic as well as Array Arithmetic. I shall show you how the rules of Matrix Arithmetic differ from those of Array Arithmetic and use it to do a few simple sums.

As I have said, Complex Arithmetic follows a third set of rules. I do not know of a spreadsheet which has the rules for Quaternion Arithmetic built into it nor, for that matter, rules corresponding to Vector Arithmetic or any other exotic mathematical set of numbers; if you do then please drop me a line.