Gerald’s Column
by Gerald Fitton

This month my article is about the behaviour of the Floating Point Emulator which is supplied with the RISC OS 4 ROM chips. In particular I shall describe how changes in the way that it works has affected some spreadsheet applications.

Problems

Let’s cut to the ‘bottom line’. These are three examples from many.

Michael Ben-Gershon upgraded to RISC OS 4 and found that one of his Fireworkz custom functions failed with the new Floating Point Emulator (FPE). The computer “hung up”! His custom function includes an iteration which continues until a value close to zero is achieved. Michael had chosen a value of 1e−18 for his ‘close to zero’ value.

Derek McCulloch has a similar problem with PipeDream. He says: “The main practical problem is with any calculation which ‘homes in’ on a solution by successive approximations . . . can fail to finish, and ends up continuously ‘hunting’.”

Eugéne R Dahmen had a different but related problem. Fireworkz uses an iterative process to find the Internal Rate of Return. Eugéne found that, after his upgrade to RISC OS 4, the Fireworkz irr(guess,range) function sometimes didn’t work—it returned an error message.

Others have written to me recounting similar problems with Schema and Eureka.

RISC OS 4

Although this is an article about the Floating Point Emulator, just for a moment let’s talk about the CDFS shipped with RISC OS 4.

We supply our GoldLine User Group CD in CD ROM XA (Mode 2) (Form 1) format. I shall call this Mode 2 for short.

Back in 1997 we decided on this format after running trials for the new CD based User Group. At that time most CDs used on the RISC OS platform were in CD ROM Mode 1 format. I shall call this format Mode 1 for short. On the Windows platform many of the newer CDs included sessions in Mode 2 format. The CD Extra format used for multi media and Kodak PhotoCD format contain sessions in Mode 2 format. In addition PhotoCD is in multi-session format.

The version of the CDFS built into our RISC OS 3.7 chips is V 2.29.

We checked our Mode 2 GoldLine CDs thoroughly, not only on our equipment but also with other selected customers. Undoubtedly CDFS V 2.29 supports Mode 2. What is more Acorn issued a most informative document describing many different CD formats (including Audio and CD Extra) all of which were supported by V 2.29.

We believed we were ‘forward looking’ in choosing Mode 2 for our GoldLine CD.

The version of the CDFS built into the RISC OS 4 ROM chips is version 2.37. This new CDFS would not read Mode 2 format properly. When reading a file into some but not all packages the first 12 bytes are ‘lost’. Acorn’s Edit loads files in a different way from most packages and the file is not corrupted. However, archives in zip format were rejected by SparkFS because the first 12 bytes of the header are missing.

From our point of view the most annoying part of this discovery was that no organisation would accept responsibility. Many of our customers questioned the quality of our discs suggesting corrupt discs and low quality media. That was bad; however worse was to come. RiscOS Ltd refused to accept that there was anything wrong with the chips they were supplying and insisted that the cause of the problem was something we were doing.

It was only when Kodak PhotoCDs users found their discs unreadable that RiscOS Ltd accepted that CDFS 2.29 did have problems. Although many patches have been issued for RISC OS 4 by RiscOS Ltd the ‘best’ solution is still the ‘OldCdfs!! fix’ by Rex Palmer. This ‘fix’ has been mentioned before in Archive; it will have a place on out web site until such time as a definitive solution is released by RiscOS Ltd. The fix by Rex unplugs the CDFS supplied with the RISC OS 4 chips, V 2.37, and installs an earlier version of the CDFS, V 2.29, which will handle Mode 2 CDs.

There are rumours that a new version of RISC OS is in the pipeline. I do hope that those involved in its release will ensure that it is thoroughly tested before releasing it. As I said in last month’s article I believe that too many users upgrade too soon just because it’s the latest piece of kit. Don’t upgrade unless you know that you have need for the new facilities—and don’t upgrade until you are sure that the new kit will do what you want. My advice particularly applies to the next version of the operating system.

Upgrading to RISC OS 4

Some spreadsheet users have had problems after upgrading to RISC OS 4. These problems have arisen when they have upgraded from a non StrongArm machine to a RISC OS 4 machine and not upgraded their spreadsheet software to match. StrongArm has a ‘cache’. Software designed for a ‘pre-cache’ machine will not run on a StrongArm machine unless you turn OFF the ‘cache’. As a rule a spreadsheet which runs on a StrongArm machine with RISC OS 3 will run under RISC OS 4.

Precision

I have RISC OS 3.7 which includes Version 4.06 of the FPE. The RISC OS 4 chips contain version 4.09 of the Floating Point Emulator.

The older FPE had greater precision than the ‘new’ version. In the ‘old’ version all calculations were performed in extended double precision (a special Acorn mode) before truncating to the ‘standard’ IEEE ‘double precision’. The working precision of the ‘old’ FPE was 80 bits comprising 64 bits for the mantissa, a 15 bit exponent and a sign bit.

It has been stated that in the ‘new’ version, calculations are performed in ‘double precision’ mode. The working precision of this mode is 64 bits comprising 52 bits for the mantissa, an 11 bit exponent and a sign bit.

I Don’t Believe It

I don’t believe that the precision of the ‘new’ FPE corresponds to Double Precision. The IEEE definition of ‘double precision’ implies a maximum exponent of 10^340 and 17 decimal digits of precision. Certainly tests I have carried out with (much) earlier versions of Acorn’s FPE have confirmed this. Let me add that Acorn’s Extended Double Precision and their Packed Decimal gave an extra couple of decimal places.

In many ot the tests I have carried out using the ‘new’ V 4.09 FPE I find that the precision appears to have been reduced to about 9 or 10 decimal digits. This corresponds to the definition given by the IEEE for Single Precision and not the Double Precision claimed for the FPE shipped with RISC OS 4. Some calculations using the new FPE do ‘work’ in Double Precision but some seem to use Single Precision!

One of the consequences of this reduction from about 17 decimal places of precision (or 19 with Extended Double Precision) to about 9 or 10 decimal places is the failure of some spreadsheet applications. The failures are not limited to PipeDream and Fireworkz but they also occur with Schema and Eureka. The failures finally appear during a subtraction when the difference between two numbers is (roughly) less than a billionth of the two numbers. I know a billionth sounds as if it is a small number but many iterative procedures built into spreadsheets have been set up to ‘close’ (exit from the iterative loop) when the difference is about 10 ^ −18; this latter number requires double the precision required for a billionth. Indeed it requires Double Precision rather than Single Precision.

Those spreadsheets which rely on detecting small numbers of the order of 10 ^ −18 in order to exit from the iterative loop will ‘hunt’ around what should be the final answer instead of converging. The spreadsheet appears to ‘hang up’.

Michael Ben-Gershon, like Derek, had a problem with the convergence of an iterative procedure. His problem was solved by changing one line of a custom function. Initially his test for “both these numbers are ‘near enough’ the same” was that the difference was less than 1e−18. He changed this value to 1e−10 and the iteration process then ‘closed’.

I have said that the failures “finally appear” during a subtraction. However, as we shall see later, the subtraction itself is not the problem. The problem arises because there is a loss of precision in calculations preceding the final subtraction.

Variance

I think that the Internal Rate of Return function, irr(guess,array), is one which best demonstrates this problem. However it is my experience that ‘IRR’ is not an easy concept to explain so I have chosen another mathematical function as my primary example. On the monthly disc you will find an IRR example together with a custom function which gives a more accurate answer when used with less precise calculating equipment.

I suppose that over the years the word ‘Variance’ has fallen out of favour and has been replaced by Standard Deviation. The Standard Deviation (SD) of a set of numbers is the square root of the Variance of those numbers—so the two are related.

The units in which the Standard Deviation is measured is the same as the mean of a set of numbers so that if you have, lets say, a set of measured lengths in mm then the SD is also measured in mm.

As a rule of thumb, if you have a set of data (say heights of people) then about 95% of the measurements will be within a couple of SD of the mean height.

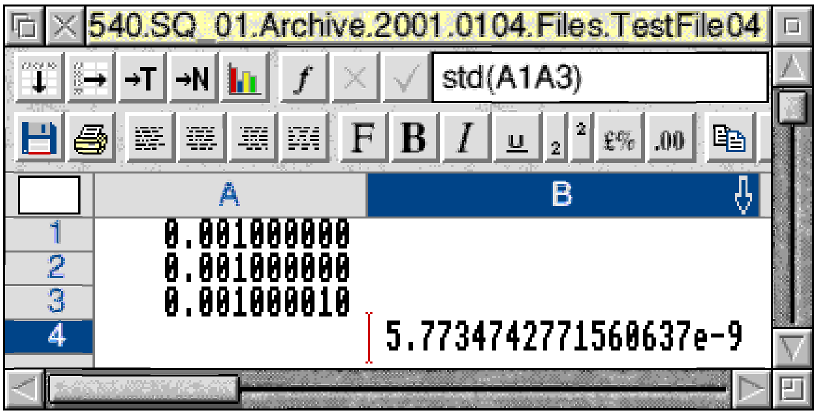
On the Archive monthly disc you will find an article called ‘Teachers’ (which I wrote for Archive in 1997). It contains an interesting worked example which demonstrates the importance of SD. I am sure that if you read it then you will have a better understanding of Standard Deviation (spread or scatter) and how it may be used to analyse data.

The ‘classical’ formulae for finding the Variance of a set of numbers is given in the PipeDream 4 Reference Guide on page 253. It has been well known by ‘Practical Mathematicians’ (a technical term for those who use numerical methods) that this formulae requires machinery with a higher precision than necessary. I use the word “machinery” because the techniques (and knowledge of this ‘precision’ problem) go back over a hundred years. Certainly it is something which was impressed upon me when I did ‘Numerical Methods’ at University a very long time ago.

Finding the SD (or Variance) of a set of numbers is a good test of Precision because the numbers are squared before the difference is taken. Squaring numbers makes them ‘twice as long’—by this I mean that you need double the precision to store a squared number accurately when compared with the original number. If the differences between your numbers is small compared with the size of the numbers then, like Derek McCulloch, you might discover a sudden interest in Precision!

The File [TestFile04]

I received a file similar to [TestFile04] from Derek McCulloch. He has recently upgraded his machinery to include RISC OS 4 and the ‘new’ FPE which comes with it. He says: “The main practical problem is with any calculation which ‘homes in’ on a solution by successive approximations . . . can fail to finish, and ends up continuously ‘hunting’.” Derek (unfairly) blames PipeDream for the lack of Precision built into the ‘new’ FPE shipped with RISC OS 4 and sent me a file similar to [TestFile04] to prove his point.



540.SQ_01.Archive.2001.0104.Files.TestFile04

std(A1A3)

	A	B
1	0.001000000	
2	0.001000000	
3	0.001000010	
4		5.7734742771560637e-9

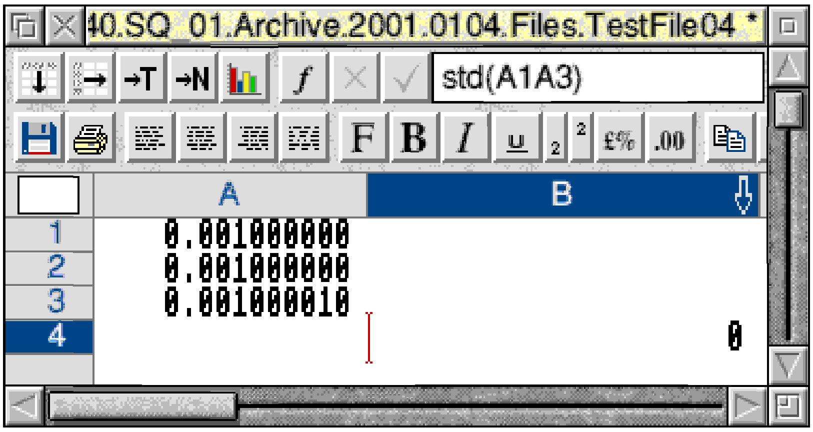
He compares PipeDream 4.13 on an A310 running an earlier version of RISC OS and FPE with PipeDream 4.50 running RISC OS 4.02 and the ‘new’ FPE. Essentially he has three numbers which are approximately 0.001; one of these numbers is 0.00100001 (which has only four zeros between the two 1s). The Precision needed to express these numbers is one part in a hundred thousand which I shall express as a precision of 1e−5. When this third number is squared it becomes 0.000 001 000 020 000 1. This requires a precision of 1e−10.

V 4.06 of the FPE has a Precision of about 1e−17 (and even older FPEs had a Precision of 1e−19!) whereas the ‘new’ FPE gives problems when a Precision of 1e−10 is required.

The file [TestFile04] has six columns, A to F. In the screenshots above and below I have included only two of these columns because the spreadsheet is rather wide. The screenshot above was produced using PipeDream 4.50, RISC OS 3.7 and V 4.06 of the FPE.

Derek’s A310 (with an early version of the Floating Point Emulator) returns a value similar to the one shown in the screenshot above. His RiscPC with Version 4.09 of the FPE returns zero as in the screenshot below! The complication for Derek is that his A310 uses PipeDream 4.13 and his RiscPC uses PipeDream 4.5. Naturally Derek was lead to the conclusion that it was PipeDream which was causing the differences in the two values.

If you compare the screenshot above and that below the only difference is the version of the FPE. The conclusion must be that it is the FPE and not PipeDream 4.5 which is causing the loss of precision.



40.SQ_01.Archive.2001.0104.Files.TestFile04 *

std(A1A3)

	A	B
1	0.001000000	
2	0.001000000	
3	0.001000010	
4		0

Even if you don’t have a copy of [TestFile04] it is relatively easy to create the spreadsheet which you see in the screenshot. If you have a zero in B4 then the chances are that you are using the ‘new’ FPE. What I want you to recognise about this example is that the ‘new’ FPE fails when the precision required is 1e−10. This precision corresponds to the IEEE Single Precision and not the IEEE Double Precision standard.

There are other tests which I have done which do indicate that the FPE sometimes operates in Double Precision mode rather than Single Precision mode.

One simple example (which you will find on disc as the Fireworkz file [TestFile02] and the PipeDream file [TestFile05]) is that the addition 1 + 1e−14 returns 1.000 000 000 000 010. This is certainly better than Single precision.

Nevertheless I still maintain that the ‘SD Test’ I have shown above does imply Single and not Double Precision when the calculation involves squaring numbers. The ‘intermittent’ nature of this Single/Double Precision problem does make it difficult to ‘prove’, particularly to those unwilling to listen, that there is a problem with FPE Version 4.09 and not with PipeDream and Fireworkz (and Schema and Eureka).

The Two Versions of the FPE

You can find out which version of the Floating Point Emulator (FPE) you have as follows: Tap <F12> so that an asterisk (the * character) appears in the command line at the bottom of the screen. At the * type in H. FPE. (enter both full stops). The Floating Point Emulator (FPE) fitted on my machine is: FPEmulator 4.06 (14 Mar 1996) (1.07Z).

In this directory I have included two FPE modules. These are versions 4.06 and 4.09, the ‘old’ and the ‘new’ respectively. You can Load either of these FPEs into RAM and force them to be used instead of the version in your ROM chips. You will find these modules in the directories NewFpe and OldFpe.

I must express my thanks to Rex Palmer for these Loaders and for the OldCdfs.

One way of installing either the ‘New’ or the ‘Old’ FPE is to add the NewFpe or OldFpe to your !Boot.Choices.Boot.PreDesk sub directory. If you decide to do this then have a look at the OldCdfs first to see how the ‘Old’ CDFS is installed using Rex Palmer’s utility.

Another method is to double click on the FPE module. This is OK for a test and it does ensure that you can get back to where you were by closing down the machine and starting it up again. If you do use this double click method then I recommend that you Quit PipeDream from the icon bar before you double click on the FPE file, then Load [TestFile04] and see what effect the change in FPE has made.

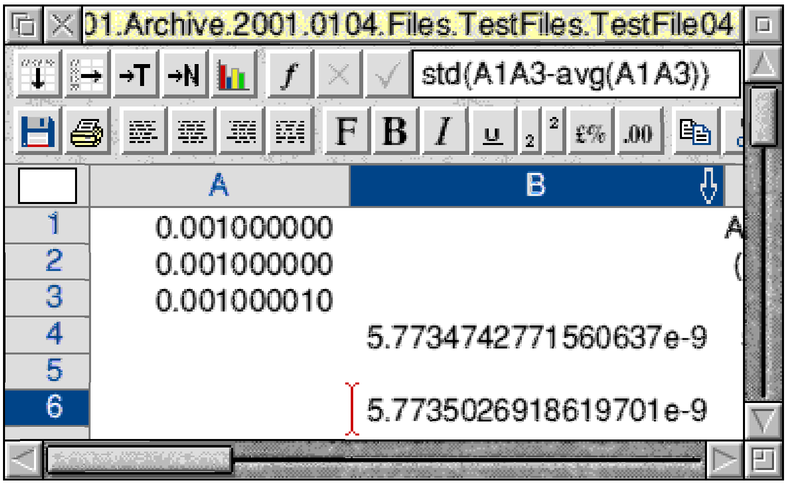
If you have RISC OS 3.7 then be very careful about unplugging any module which is in ROM. You may have to execute a ‘Power on <Delete>’ or RMReInit to recover the version of the FPE built into your machine your machine.

Finding the Standard Deviation

There are other ways of improving accuracy without increasing the precision of calculations. In cell [TestFile04]B6 you will find an improved formulae for the standard deviation of three ‘large’ numbers which have ‘small’ differences.

The most accurate technique for a machine of a given precision is to find the mean (average) of the numbers first. Then calculate the SD of the differences between the numbers and the mean.

The formula in [TestFIle04]B6 (shown in the screenshot below) will return the most accurate answer possible for the precision available in the machine. In this example the ‘usual’ formula loses accuracy in the fifth decimal place. I leave it as an exercise to the reader to determine the accuracy of the value given in B6.



std(A1A3-avg(A1A3))

	A	B
1	0.001000000	
2	0.001000000	
3	0.001000010	
4		5.7734742771560637e-9
5		
6		5.7735026918619701e-9

In the file [TestFile03] (available on the Archive monthly disc) I have calculated the SD in three ways in cells B9, B10 and B11. If you have this file then I suggest that you ‘play with it’ and you will see how the technique I have described improves the accuracy.

Summary

The introduction of RISC OS 4 has produced many ‘problems’.

These include difficulties reading some formats of CD, difficulties with printing, some RAM timing problems and some difficulties with the handling of small numbers by the Floating Point Emulator (FPE). Most of these problems can be overcome by ‘unplugging’ modules supplied with RISC OS 4 and re installing the modules supplied with RISC OS 3.7. Rex Palmer has developed a couple of utilities which unplug the ‘new’ and plug in the ‘old’ CDFS and FPE. Other people have developed similar ‘fixes’ for the printing problems.

I regret that the ‘bits and pieces’ supplied with RISC OS 4 have caused problems for users but I regret more the delay (by those selling RISC OS 4) in accepting that these problems exist. I was particularly upset when the blame was unfairly put on our techniques (our use of Mode 2 CDs) and the PipeDream and Fireworkz software.

Last month I wrote about upgrading and I advised caution. There is a rumour that there is a new version of RISC OS in the pipeline. Whilst I welcome this I would advise the less knowledgeable not to upgrade simply ‘because it is there’. I recommend waiting a while so that those who really need the extra facilities will find all the ‘bugs’ and hopefully will feed back enough information for them to be fixed before general release.

Finally

Thanks for all your letters and emails. Please write to me at the Abacus Training address which you’ll find in Paul’s Fact File at the back of Archive.