

Excursions INTO EXCEL

PART 4

by Mike Bedford

For the last three months, now, I've presented examples of the types of scientific, mathematical, and engineering tasks that can be performed using Excel. And, by now, a few of you are, hopefully, starting to realise that it's far more than a tool for accountancy and business graphics. If so, it's reasonable to assume that some of you are already using it for technical applications. Before we go too far down this path, though, and before you've created too many Excel masterpieces, there's a topic I really ought to cover – that of documenting Excel workbooks. Let me explain what I mean by this.

Good Programming Practice

For the moment let's forget about Excel and turn our attention to a programming language such as BASIC, C++, or PIC assembler code. If you've ever written a computer program you must surely have had the experience of returning to one of your own creations a few weeks, months or years later, and not being able to figure out how it works. What was obvious when you wrote the program is now as clear as mud. And if the reason for returning to the program is to find a bug or to add some new functionality, not knowing how it works is a major disadvantage. What's more, if even the author of software has difficulty in understanding it at a later date, the chances of a third party being able to get to grips with it are even more remote. This is why students on programming courses are always taught to document their programs.

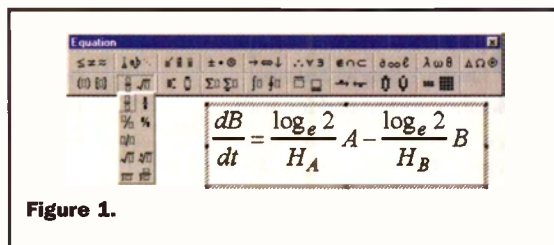


Figure 1.

This involves putting comments into the programs. These textual messages are ignored by the compiler or assembler but explain the purpose of each line or each section of the code, indicate what each variable represents and so forth. Admittedly, understanding an Excel workbook isn't as difficult as understanding a few thousand lines of assembler code, in fact it's blatantly obvious how most work. Even so, the types of application we're looking at in this series are not run-of-the-mill and it's quite feasible that you'll have difficulty in understanding how your workbooks operate when you return to them after a break. Adequate

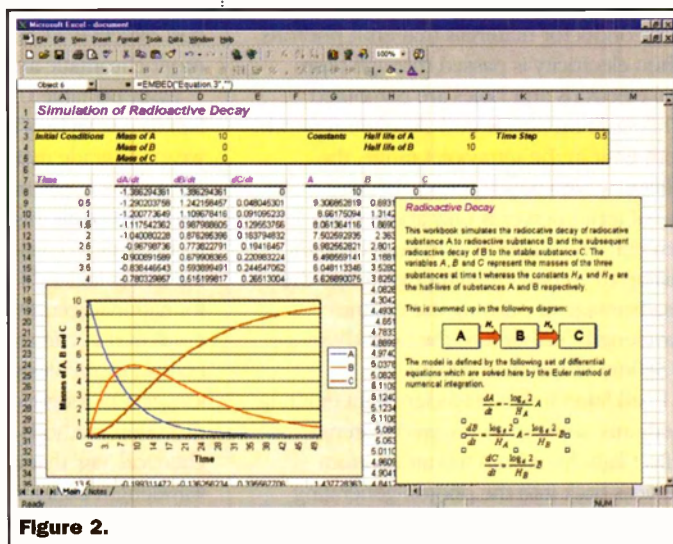


Figure 2.

documentation is, therefore, highly desirable. And if you intend to share your work with others it's a must. In fact, I'd go further than this. A computer program source file is pure ASCII text so the scope for doing anything particularly stylish is limited.

Excel, on the other hand, provides lots of scope for making a workbook look good. My suggestion is, therefore, that you don't just do the minimum necessary to explain how a workbook works but that you format it in a way that makes it pleasant to read.

Formatting Options

OK, that's enough of the soap box stuff – now to some practical suggestions. First of all the obvious ones. Give your workbook a title that appears at the top of the page. Use the formatting options such as the font, the size and the colour to make that title stand out. Give titles to the row and column headings, once again using formatting as appropriate. Include a short textual description which explains what the workbook does. As an alternative to entering the text into a cell, with all the formatting difficulties this brings, you could use a text box that is created from the Drawing toolbar. Text boxes float over the workbook and can be dragged around and resized in the usual way. Alternatively, if you don't want to clutter up the workbook

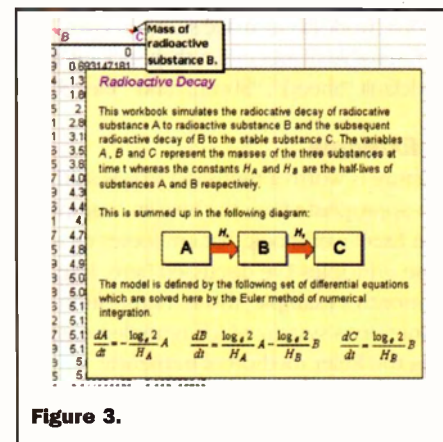


Figure 3.

unduly you could use Excel's comment feature, which causes a comment box to open up, when you move the cursor over the cell to which the comment is attached. The commented cell is highlighted with a red triangle in the corner and typically you'd enter something like "see here for further details" in the cell itself. Since we're dealing with scientific and mathematical applications, adding equations will be a common requirement and, fortunately, Microsoft Office comes with an equation editor which can be used from within any of the applications. To add an equation, select Insert > Object... > Create New and then pick "Microsoft Equation" from the "Object type:" list box. If Microsoft Equation doesn't appear then this option hasn't been installed on your PC and you'll have to install it separately. Like text boxes, equations appear in boxes which float over the text. The boxes also have black outlines by default but these can be removed under Format > Object. And although you can't embed an equation within a text box, you

can float an equation over the top of a text box so that the equation appears to be a part of the text. Finally, you might also want to include line diagrams or even photographs in your workbooks. For simple line drawings you could use the drawing facility which is provided in Excel. However, for more complicated diagrams you might choose to use a separate fully featured drawing package and then import the drawing into Excel.

The documentation we've seen so far is intended to be read by whoever uses the workbook. However, it's also appropriate to add more detailed documentation, perhaps explaining how it's been implemented, to help anyone (including yourself) who might need to modify it in the future. Ideally, though, this isn't the sort of documentation you'd want cluttering up your main worksheet. Fortunately, therefore, Excel workbooks can include multiple worksheets and three are provided by default when you create a new workbook. One option, therefore, is to include all the behind the scenes implementation information in a separate worksheet. If you make use of multiple worksheets, make sure you give them sensible names rather than sticking with the default "Sheet1", "Sheet2" and "Sheet3".

An Example

A picture is worth a thousand words so this is an appropriate time to show an example of an Excel workbook that embodies most of the principles I've discussed here. Like last month's example, this is a simulation of a continuous system which is implemented using the Euler method of numerical integration to solve the differential equations. This should all be familiar to you so you can concentrate on this month's theme, namely the way in which the workbook has been documented. All you need to know about the example, therefore, is that it's the simulation of radioactive decay in which radioactive substance A decays to produce radioactive substance B which, in turn, decays to give the stable substance C. The following equations define the model:

$$\frac{dA}{dt} = -\frac{\log_e 2}{H_A} A$$

$$\frac{dB}{dt} = \frac{\log_e 2}{H_A} A - \frac{\log_e 2}{H_B} B$$

$$\frac{dC}{dt} = \frac{\log_e 2}{H_B} B$$

where A is the mass of substance A, B is the mass of substance B, C is the mass of substance C, H_A is the half life of substance A and H_B is the half life of substance B.