

# Forty years of spreadsheets

John R Hudson\*

11 June 2019

Though spreadsheet-like programs had existed on mainframes, [VisiCalc](#), the first spreadsheet for a micro-computer, created by Dan Bricklin and Bob Franckston in 1979 for the Apple II computer, was the first to present the idea in the form of a single interactive interface. It was quickly ported to other micro-computers and to the IBM PC when it arrived in 1981 for which it became the ‘killer app’ generating sales of the IBM PC.

Unlike word-processors, which have taken over from typewriters, and databases, which have taken over from card indexes, spreadsheets had not previously existed in an alternative form.

## What do spreadsheets do?

Spreadsheets allow you to break down a very long or very complicated series of calculations into hundreds of simple ones and display the results almost ‘instantly.’ Partial entries are allowed so that you can see how things are going part way through a long calculation and a change to a single calculation can be reflected in a completely new set of results almost ‘instantly.’

This ability to handle large quantities of routine calculations and display the results ‘instantly’ offered the possibility of ‘What if?’ calculations — ‘what if’ we change our prices by this percentage or interest goes up by this percentage?

## Spreadsheet concepts

**Cell:** each sheet in a spreadsheet is like a set of pigeonholes. By convention the leftmost column of pigeonholes is called A, the next one B and so on with column 27 called AA, column 28 AB and so on; the topmost row of pigeonholes is called 1, the next row 2 and so on. Each ‘cell’ or pigeonhole is then named by the column and row in which it falls, for example, A1, G5, L16, AP255.

---

\*Thanks to members of the [Bradford Linux Users Group](#) for their comments on a presentation drawing on this paper.

	A	B	C	D	E	O
1:	PERSONAL ACCOUNTS					
2:		on 10/ 5/1987	January	February	March	Total
3:						
4:	Income		670.00	670.00	670.00	8190.00
5:	Other Receipts		21.00	31.00	47.00	522.00
6:						
7:			691.00	701.00	717.00	8712.00
8:						
9:	Mortgage		103.24	103.24	103.24	1199.68
10:	Rates					469.12
11:	Building		23.76	67.89	32.09	569.84
12:	Fuel			303.25		860.22
13:	Food		240.00	240.00	240.00	2880.00
14:	Clothing		40.00	40.00	40.00	480.00
15:	Car		150.00	150.00	150.00	1800.00
16:	Leisure		30.00	30.00	30.00	360.00
17:						
18:			587.00	934.38	595.33	8618.86
19:						
20:	Balance		104.00	-233.38	121.67	93.14

Figure 1: Using built-in formulae

Unlike a relational database table, which also has columns and rows, each cell may contain *either* a data item *or* a formula/function. Normally, each cell, as seen by the user, displays the data item or the results of the calculation defined by the formula/function in the cell.

However, most spreadsheets can display the formulae in their respective cells for checking or analysis. Figure 1 illustrates the use of formulae and a function, the date. Figure 2 shows the formulae that underlie the spreadsheet.

By using cells to hold the many separate calculations that might go into writing a conventional program, the user can break down a ‘program’ into manageable units, can test the ‘program’ easily and can amend the ‘program’ one cell at a time, normally without having to amend any other parts of the ‘program.’

**Data types:** like databases, spreadsheets distinguish between the various types of data entry a user might make; all can handle *numeric* data to be used in calculations and *alphanumeric* data (also known as ‘character strings’, ‘labels’, ‘strings’ or ‘text’) but most spreadsheets now accept a range of other types of data, such as *dates*, on which it is possible to perform calculations such as the number of unpaid bills in the last three months.

Most spreadsheets assume that you will be entering *numeric* data unless you specify

	:A:	:B:	:C:	:D:	:E:	:O:
1:	PERSONAL ACCOUNTS					
2:		on TODAY	January	February	March	Total
3:						
4:	Income					SUM(C4:N4)
5:	Other Receipts					SUM(C5:N5)
6:						
7:			C4+C5	D4+D5	E4+E5	O4+O5
8:						
9:	Mortgage					SUM(C9:N9)
10:	Rates					SUM(C10:N1)
11:	Building					SUM(C11:N1)
12:	Fuel					SUM(C12:N1)
13:	Food					SUM(C13:N1)
14:	Clothing					SUM(C14:N1)
15:	Car					SUM(C15:N1)
16:	Leisure					SUM(C16:N1)
17:						
18:			SUM(C9:C16)	SUM(D9:D16)	SUM(E9:E16)	SUM(O9:O16)
19:						
20:	Balance		C7-C18	D7-D18	E7-E18	O7-O18

Figure 2: The built-in formulae displayed

otherwise.

**Formulae:** can use the normal mathematical operators such as plus, minus, multiply, divide and so on, cell references such as A1, G45, AP7, B255 or functions. The following keys are used for the most common mathematical operations:

+ = plus

- = minus

\* = times

/ = divided by

^ = raise to the power of.

The cell to which a reference is made may hold a data item or a formula or function, in which case the result of that formula/function will be used for the calculation. In theory, a spreadsheet could consist of one data item and many thousands of cells containing formulae/functions referring back to that one data item. In practice, most spreadsheets consist of a balance of data items and formulae/functions.

**Function** an operation which changes the nature of a value or simplifies a long formula; for example, the function SUM(G2:G6) avoids having to type G2+G3+G4+G5+G6 as the formula.

Modern spreadsheets provide a plethora of functions which incorporate all the formulae needed for most financial, mathematical, scientific and engineering calculations such as averages, percentages, net present value,  $\pi$ ,  $\epsilon$  and so on.

Virtually all modern spreadsheets require you to precede a formula or a function with =.

**Protection:** because it is so easy to alter the contents of a spreadsheet cell, all spreadsheets have a facility to 'protect' or 'lock' the contents of a cell against interference. Normally, the cell can be easily 'unprotected' or 'opened' but more sophisticated spreadsheets also allow password protection of specific cells, columns, rows or ranges.

**Range** a block of cells normally described by its top left and bottom right cells, for example, C4:F8; a range can often be given a name to which formulae or functions can be applied and which saves you having to remember the cell references.

**Recalculation:** as you enter data, formulae or functions the spreadsheet will automatically recalculate the effect of your last entry on all the other cells in the spreadsheet (unless you turn off automatic recalculation). This was what most entranced the early users of spreadsheets. Sometimes, when you are entering a lot of figures in a large spreadsheet, waiting for automatic recalculation after every entry can slow you down; so you can select manual recalculation, where you tell the spreadsheet when to recalculate, instead.

## The development of spreadsheet programs

Because VisiCalc would only run on a limited number of CP/M computers, [SuperCalc](#) was created in 1981 to fill this void and became the most advanced of all the spreadsheet programs of the 1980s because of its ability to solve circular references and its use of binary maths.

It was also unique in only requiring the user to enter = to indicate a formula or ' to indicate text when there was an ambiguity.<sup>1</sup>

Once the vector graphics program, DR Graph, became available for CP/M, it became possible to export spreadsheet data directly into DR Graph and, when the DOS version of SuperCalc came out, the vector graphing capabilities of DR Graph were integrated into it giving SuperCalc the most sophisticated graphing capabilities of any spreadsheet at the time.

In 1982 Multiplan was produced using a proprietary Microsoft compiler but, in spite of being ported to range of different operating systems, it never acquired much traction in the spreadsheet world.

In 1983, the creation of [Lotus 1-2-3](#) pushed VisiCalc and SuperCalc into the background. It by-passed a lot of DOS functions to interact directly with the hardware, thus enabling it to run much faster than VisiCalc or SuperCalc, and it superseded VisiCalc as the 'killer app' which drove sales of the IBM PC. The company also had to acquire Software Arts, the company then producing VisiCalc, to avoid a patent lawsuit. 1-2-3 referred to the fact that it offered database and graphing capabilities alongside the spreadsheet.

Adam Osborne, creator of the Osborne 'luggable' computer, also created a Lotus 1-2-3 clone called [VP Planner](#) which was the first multi-dimensional spreadsheet; that is, as well as columns and rows, you had sheets and you could reference the equivalent cell in several different sheets. However, the company was found guilty of infringing Lotus 1-2-3's patents in 1990 and VP Planner disappeared though the idea of multi-dimensional spreadsheets did not.

In 1986 Database Software released [Mini-Office II](#), initially for the BBC micro and then for a number of other home computers; this outdid Lotus 1-2-3 in having six modules: word processor, database, spreadsheet, graphics, communications and label printer.

In 1987 Microsoft released [Excel](#), initially for the Apple Macintosh; it was less well featured than its competitors and, unlike its competitors, its macros relied on proprietary software. It made little ground until the arrival of Windows 95 when its early and widespread availability on Windows led to it becoming the dominant spreadsheet for PCs.

In 1988 WordPerfect, rapidly becoming the *de facto* word-processor in business after some disastrous mistakes by WordStar, the first business word-processor for micro-computers, released [Quattro Pro](#); this is still available as part of the WordPerfect Office suite.

---

<sup>1</sup>As this paper is based on one originally put together when SuperCalc was around, all the examples use SuperCalc conventions.

In 1985 StarWriter was released in Germany; in 1994 it developed [StarCalc](#) and Star-Base to create StarOffice which was acquired by Sun Microsystems in 1999 allegedly because it was cheaper than buying Microsoft Office licences for its staff. Sun Microsystems released it as [OpenOffice](#) and in 2010 [LibreOffice](#) was forked from it. Both OpenOffice and LibreOffice present themselves as direct alternatives to Microsoft Office.

In 2001 [Gnumeric](#) was released as part of the Gnome Project. Gnumeric has the distinction of being capable of handling spreadsheets sixteen times larger than its nearest competitor and is the true inheritor of SuperCalc's scientific and engineering excellence because, when Microsoft released Excel, it deliberately incorporated in it a number of known bugs in Lotus 1-2-3 so that people using Excel would not get different results from those they would get when using Lotus 1-2-3.<sup>2</sup> It also created confusion by starting its date system from 1 January 1904 in the Macintosh version and from 1 January 1900 in the PC version, compounding the problem by including 29 February 1900 even though 1900 was not a leap year.

Gnumeric also follows SuperCalc in having superior graphing capabilities.

In addition to these, there have been many other spreadsheet programs, mostly as part of suites such as AppleWorks, Microsoft Works and Calligra. Most recently Excel has become available as part of the online Office365 suite while Google offers the online Sheets program with more limited and not always compatible features; for example, LibreOffice allows people to work on Excel spreadsheets containing macros without affecting the macros whereas Sheets does not save them.

## Limitations

With all these advantages, what are the disadvantages? Firstly, all a spreadsheet's calculations have to be done in the computer's workspace. Manipulating a spreadsheet on disk would dramatically slow down its capacity to provide 'instant' results, particularly those involving a large number of interconnected calculations. This means users need to be disciplined about saving the spreadsheet when they wish to save particular sets of results.

Secondly, few spreadsheets can manipulate text as efficiently as numerals. So, calculations involving text, for example, counting all the instances where someone had bought petrol or paid out petty cash for stamps, would slow the spreadsheet down. Calculations involving mathematical functions run fastest.

Thirdly, studies over several decades have shown that around half the spreadsheets used in business contain errors which mean that the results do not reflect what the user thinks they do; in some cases, businesses have lost millions as a result of fairly elementary errors in a spreadsheet.

However, the most notable example occurred in an academic paper (Reinhart and Rogoff, 2010) arguing that 'gross debt (meaning all debt that a government owes, in-

---

<sup>2</sup>This is not unprecedented: nearly all English bibles contain errors from earlier translations which modern translators have been reluctant to correct for fear of upsetting readers used to the erroneous translations.

cluding debt held in government trust funds) exceeding 90 percent of the economy has a significant negative effect on economic growth.’ This paper was used to justify austerity. In a subsequent analysis of the Excel spreadsheet used by Reinhart and Rogoff, Herndon et al. (2014) wrote:

While using RR’s working spreadsheet, we identified coding errors, selective exclusion of available data, and unconventional weighting of summary statistics.

## Designing a spreadsheet

Though a spreadsheet is easy to ‘program’, it is usually helpful to try and ‘design’ it on paper before switching on the computer.

If the spreadsheet contains a lot of cells, it is worth designing the left hand columns of the top 20–40 rows as a summary sheet where inputs can be made and summary outputs shown. Alternatively, multidimensional spreadsheets allow you to make the first sheet a summary sheet, hiding the detail of the calculations in the sheets behind it.

Note that interspersing a spreadsheet with blank rows causes no problems most of the time and is easier on the eye.

## Some useful terms

**Absolute copy** normally a spreadsheet *adjusts* the formula in a copied cell; so copying cell C7 in Figure 2 to D7 will result in the *relative* adjustment of the formula in cell C7 from C4+C5 to D4+D5 in D7. However, if you wanted D7 to contain C4+D5 (a rather pointless exercise but suggested here purely for illustration), you would make an *absolute copy* of cell C4 and a *relative copy* of cell C5, or ask for ‘no adjustment’ to C4. Entering \$C4+D5 in C7 would have the same effect of ensuring that C4 was always copied without adjustment wherever it was copied to — see [Relative copy](#).

**Cell width** in early spreadsheets, cell width was determined by column width; modern spreadsheets allow cells to span columns; in this case their width is determined by the total width of the columns spanned.

**Character string** the traditional name given to an alphanumeric or text string in which any numerals are to be treated as characters and not as numbers on which mathematical operations will be carried out, such as cheque numbers or invoice numbers. You may have to start a character string with " or ' to let the spreadsheet know what it is — a few are clever enough to work it out anyway.

**Circular reference** a sequence of formulae such that the result of the last one is used by the first one; this can lead to strange effects such as figures suddenly getting bigger or smaller very rapidly

**Column width** a column can be made narrower or wider to suit the layout you want for a spreadsheet; this affects the width of individual cells in a column and cells which span the column.

**CSV** comma separated value: a way of saving the data in spreadsheets for use in another program, sometimes a spreadsheet program with an incompatible format, sometimes a database or other program. Originally developed by IBM in 1972, it was adopted by WordStar, dBASEII and SuperCalc as the standard way of exchanging data between these CP/M programs.

**Date value** most spreadsheets allow you to use date values by assigning a numeric value (unknown to you) to every date; date functions allow you to display dates and do calculations using dates, such as 'What is 60 days after January 13th, 2004?' The spreadsheet knows that it is a leap year and allows for that; another date function can tell you what day of the week it will be,

**Entry line** the line above or below the main spreadsheet area where your entries are displayed before being placed in the appropriate cell and to which your entries are brought if required for amendment later.

**Format** a cell, row, column or whole spreadsheet may be 'formatted' to display text and numerals in particular ways. In the case of text, this usually means flush left, centred or flush right in the column. In the case of numerals, they may be displayed as integers, as floating decimals, as a fixed decimal (where you choose how many decimals should appear after the decimal point), as a percentage, as a currency value with, for example, a £ or a \$ sign appearing in front of the figure, in scientific notation and so on. The range can be enormous.

**Home** cell A1.

**Integer** the format for a cell where you want only whole numbers (no decimals) to be displayed.

**Macro** because the 'program' in a spreadsheet has been broken down into many smaller instructions, it can be tedious to repeat a sequence of instructions. Most spreadsheets allow you to define 'macros', that is, single keystrokes or short sequences of keystrokes that stand for a much longer sequence of spreadsheet commands. Unfortunately, while most spreadsheets of the 1980s which had macros followed the Lotus 1-2-3 approach, Microsoft introduced proprietary macros into Excel which are incompatible with and cannot be exported to any other spreadsheet program.

**Multi dimensional spreadsheet** most spreadsheet programs allow you to add several sets of pigeonholes behind the first one making it easier to combine calculations which would otherwise be in separate spreadsheets. These have more complicated cell referencing but otherwise are no different from normal two dimensional spreadsheets.



**Relative copy** copying a cell so that any cell references in a formula change by how far the cell is copied; for example, if G7 has the formula `SUM(G2:G6)` in it, this will be copied to H7 as `SUM(H2:H6)` and to G14 as `SUM(G9:G13)` — see [Absolute copy](#).

### **Title column**

**Title row** column 1 and row A are known as the ‘titles’ because early spreadsheets could be set to continue displaying these even when the rest of the window was displaying another part of the spreadsheet. Modern spreadsheets allow you to keep more than one row or column fixed while the reminder scroll.

**Value** a simple piece of data, whether a ‘character string’ or a number, held in a cell; also the result of a formula or a function.

**Window** the view of the spreadsheet shown on the screen; more than one ‘window’ can often be displayed at one time allowing you to view different parts of a large spreadsheet at the same time. They can usually be ‘synchronised’ so that they move together or ‘unsynchronised.’

## **References**

Herndon, T., M. Ash, and R. Pollin (2014, March). Does high public debt consistently stifle economic growth? A critique of Reinhart and Rogoff. *Cambridge Journal of Economics* 36(2), 257–279.

Reinhart, C. M. and K. S. Rogoff (2010, May). Growth in a time of debt. *American Economic Review* 100(2), 573–78.