A TUTORIAL ON COMPUTER SPREADSHEETS AND THEIR APPLICATIONS IN ENGINEERING ANALYSIS/COMPUTATIONS*

BY

A. O. F. WILLIAMS
DEPARTMENT OF CHEMICAL ENGINEERING
LAGOS STATE POLYTECHNIC
P. M. B. 21606, IKEJA, LAGOS

ABSTRACT

Traditionally, scientists and engineers use computers for problem solving, mainly through programming in such languages as FORTRAN, BASIC, Pascal, or more recently, C. While this is the only approach for very complex and intensive computations, other tools with which one may quickly solve a variety of other problems (without much programming and debugging that is characteristic of the programming language approach) now exist. One of these, which has become very popular is the spreadsheet software. Given its verstility and the fact that it is simple to learn, and to use, it is clearly desirable that students, scientists and engineers become familiar with this powerful analytical and graphical tool. Thus this tutorial paper hopes to provide a brief but functional overview of spreadsheet software, with some illustrations of their applications in scientific an engineering computations. In particular, the paper answers the question - what are spreadsheets? and discusses the categories of allowable data types, and how to enter these into a spreadsheet software. Three examples which include detailed step-by-step procedure for setting up the computer spreadsheet solution are used for illustrations. The paper should be of interest to all science, technology and engineering students; as well as professional scientists, engineers and technologists.

1. Introduction

The past two decades or so have witnessed rapid changes in the computing resources available for scientific, engineering analysis and computation or information processing: from the log tables/slide rule to calculators - ordinary, scientific and programmable scientific types - and then computers - mainframes, minicomputers and most recently, microcomputers (also called personal computers, Pcs). Because they are relative cheap to buy and maintain (when compared with mainframes and minicomputers), microcomputers have bceome an increasingly important tool in a variety of engineering and other applications (Williams, 1994). In the past, the scope of applications of microcomputers was limited by available system's memory and speed. This has howeever, changed drastically. The recent breakthroughs in large-scale integration (LSI) and very large-scale integration (VLSI) has led to the development and maufacture of microcomputers with speed and memory capacities near, or surpassing those of minicomputers and mainframes of yesteryear (see Chorafas, 1990). The result of this has been an increase in the use of microcomputers in many science, engineering and other related disciplines.

Traditionally, scientists and engineers use computers for problem solving, mainly through programming in such languages as FORTRAN, BASIC, Pascal, or more recently, C. While this the only approach for very complex and intensive computations, other tools with which one may quickly solve a variety of other problems (without much programming and debugging that is characteristic of the programming language approach) now exist. One of these, which has become very popular is spreadsheet. As Ravella (1993) points out, spreadsheets suit the occasional user who does not have the time and resources of a large programming computer facility to help formulate and debug a source code written in any of the previously mentioned programming languages. Spreadsheets are simple to learn and to use, and after a few hours, one can almost immediately begin to produce valuable results.

Given its versatility for a variety of problems, it is important and desirable that scientists, engineers and even students become familiar with this powerful analytical and graphical tool. As part of the general objective of getting students, professional scientists

*Paper first submitted for publcation: Oct., 1994
Reviewers' comments received: 7 Oct., 1995;
Revised Manufscript submitted: 31st October, 1995.

and engineers become familiar with available, modern computing resources, an earlier paper (Williams, 1993) was devoted to programmable calculators and their applications in scientific and engineering applications. When compared with programmable calculators, however, Sowa (1987) notes that a personal-computer spreadsheet provides better readability, more intermediate results, easier outputs (including graphics), and more versatile storage. It is also easier to setup a spreadsheet than to program such a calculator. On the other hand, spreadsheets may be too slow and too inflexible for rigorous modelling of engineering systems. Such problems typically require simultaneous solution of several equations either by matrix methods, or by trial and error, and can be done better with specialized programs. The purpose of this tutorial paper is to present a brief, but functional overview of spreadsheets, together with illustration of their application/use in scientific and engineering computations. Section 2 of the paper answers the question What are spreadsheets? Section 3 discusses the categories of allowable data types and how to enter these into a spreadsheet. Section 4 presents three examples which are used to illustrate the application of spreadsheets in engineering analysis and computations. Section 5 concludes the paper.

2. What are Spreadsheets?

Some of the presentation here and in the next section is a revision/update on that initially given by Williams (1992).

A spreadsheet is a computer program which may be regarded as the elctronic version of accountants' ledger books, which are used to record figures and other information. However, unlike the ledger book, spreadsheets are powerful analytical and graphical tools which can be used to perform a wide variety of tasks which include: preparing budgests, developing business plans, performing standard accounting functions, cost estimating, financial analysis, presentation graphics, maintaining data on employees or clients, managing inventory, linear regression analysis, matrix multiplication and inversion, etc.

Although originally introduced as tools for accountants, spreadsheets have a nunmber of unique advantages which makes them very suitable for scientific, and other engineering applications. These advantages include:

(i) **"Unlimited" worksheet size:** Theoretically speaking, the size of the worksheet is only limited by the computer memory. Thus, what is displayed on the computer screen is actually only a fraction of the entire spreadsheet. It is possible, though, to scroll the spreadsheet or use the GOTO command to display different areas.

(ii) **Automatic calculation:** One can enter formulas in a worksheet to automatically calculate values; for examples, to total the values in a column, or to calculate the average, standard deviation, the variance, the minimum and the maximum in a range of entries, etc. Built-in business functions include present value analysis, internal rate of return, the payment you would have to make for a particular loan etc. Scientific functions include square roots, abosolute value, integer, fraction, sine, cosine, tangent etc. Many of the built-in functions allow one to perform complex calculations without extensive programming and debugging as is conventional with any of the programming languages. Spreadsheet operations may be completely automated through the use of macros (A macro may be regarded as a user's program to carry out a task or solve a problem, and consists of the recorded sequence of strokes and/or commands necessary to achieve this. The macro may then be executed with a single command, commands that can be invoked with a single command).

(iii) **Easy editing:** It is easy and straight forward to make changes to desired entires in the spreadsheet, just as it is easy to make entries. Other entires that are affected by the change(s) are automatically updated. Also, a spreadsheet has flexible data entry format (Sowa, 1987). Labels and explanations can be typed in wherever they are needed, and input values or results (in the desired number format such as fixed, scientific, etc.) can easily be arranged in the required order.

(iv) **Powrerful commands:** By hitting a few keys, one can issue powerful commands to do a variety of tasks. Some commands deal with the entire spreadsheet/worksheet or just a range or part of the worksheet. These commands include copy, move, save, graph, print, insert/delete rows or columns, change the width of columns, display numbers in desired format such as fixed, scientific, currency, financial etc.

(v) **Instant graphs:** One can instantly create and print high quality business graphs such as line, bar, pie etc. for a pictorial presentation of a spreadsheet data.

Spreadsheet programs are ideal for applications that involve a lot of tabular calculations and comarison, or those which can be jso structured. As Cheremisinoff (1987) notes, spreadsheet programs also provide a convenient format for archiving data,

organizing and previewing reasonably large databases, and performing simple (and not too complex) mathematical manipulations. Sowa (1987) also points out that spreadsheets can save much time in calculations that are cumbersome to perform repetitively on a hand held calculator, and yet too trivial to handle with a specialized program.

2.1 Dynamic Use of Spreadsheets

As Stair (1986) notes, the real advantage of spreadsheets is the ability to make changes and to have these changes immediately and automatically reflected in the entire worksheet. Thus, spreadsheet models are not static. They are not designed to be used once. This is very important as one can use this unique feature to quickly carry out parametric sensitivity studies.

2.2 Using Templates

A template is a worksheet that contains all the necessary formulas but none of the values that are to be used. You can develop your own worksheet templates, for examples you might develop all the equations to use in statistical analysis of a grouped data, a beam fixed at both ends under distributed load, etc. Then when you actually want to make this analysis, you enter the appropriate data into the worksheet, and the formula do the rest.

2.3 Types of Spreadsheet Software

There are a number of commercial spreadsheet software available in the market. Among these are FREELANCE, SUPERCALC, EXCEL, LOTUS 1-2-3, QUATTRO etc., with the last two being the most popular. All these software packages are very similar in operation and scope of applications. If one understands how to use one, only little or nothing will be required to understand how to use another. For an introduction to the basic procedure on how to use a typical spreadsheet software such as QUATTRO, including how to save and retrieve spreadsheet files, printing and ending a work session, etc., see Williams (1992).

3. Data Types and Entering/Editing Data in Spreadsheets

3.1 Data Types

There are two types of data that can be entered into a spreadsheet: labels and values. A spreadsheet program determines the data type of a cell by the first character that you type.

3.1.1 Labels

A label is a text cell entry usually used for descriptive information, such as name or address. It can begin with any letter or punctuation mark except the following:
/ + - $ (@ #.

When you enter a label into a cell, it is positioned according to the spreadsheet's default alignment, which is initially left. This can, however, be changed to right or centered, as desired.

To align a label differently from the default, one will have to precede the label with a label-prefix character as follows:

Label - prefix character	Alignment
' (apostrophe)	Left
" (quotation mark)	Right
^ (caret)	Centered

Labels can be up to 240 characters long. When you enter a label that exceeds the width of the cell, the spreadsheet program automatically displays the "spill over" text in blank cells to the right.

3.2 Values

A value is any numeric data, entered either as a number, a date, or a formula that calculates a number. Formulas usually are numbers you have entered in other cells of rows and columns to calculate values. Unlike labels, the alignment of values data types cannot be changed. All value entries are automatically aligned with the right side of the cell.

(i) **Numbers:** A number entry basically contains only numerals (0.9), although you may include a leading negative or positive sign (- or +), or follow a number with a percent sign (%). You can also include a single decimal point. One may also use scientific notation (eg. 8.471E+5) to enter a number. If the calculated number fits in the cell (and the display format makes it possible), it will be displayed in full.

A number can be up to 240 characters long. Unlike labels, however, if a number doesn't fit in the current cell, it won't run into adjacent cells. Instead, it displays either in scientific notation or as a row of asterisks (***), depending upon the cell's display format. To correct this, one will have to adjust the column width.

(ii) **Dates:** This is another type of value entry for entering dates. For more information on how to enter

this type of data value into a spreadsheet software such as QUATTRO, see Williams (1992)

(iii) Formulas: A formula is another way to enter and display the value in a cell. Formulas allow the user take advantage of one of the most powerful and time-saving functions of spreadsheets: automatic calculation.

A spreadsheet software can instantly calculate all kinds of information. Using data entered in the spreadsheet, it can total columns of figures, calculate average, standard deviation etc. More interestingly, if the figures involved are changed, the program automatically recalculates all the formulas or cells.

Formulas combine values and operators e.g. If +A10+A12 is the formula in cell B14, the spreadsheet will add the values in cells: A10 and A12, and put the result in cell B14.

A formula must begin with any of the following characters:

0 1 2 3 4 5 6 7 8 9 . + - (@ # & *

To begin a formula with a letter, such as for a cell address or text entry, precede it with a plus sign (+). Otherwise, the spreadsheet software interprets the entry as a label instead of a value and does not calculate the formula.

In general, formulas can be up to 240 characters long. You can include spaces, between operators and values, but they will be deleted automatically.

(a) Operators: Formulas use operators, or mathematical symbols, to express a relationship between two or more values e.g. F5-D10. Often, formulas contain several operators e.g. +F10-E20+0.24*A6.

The result of a formula depends on the order in which the arithmetic operations are performed. A spreadsheet program determines the order of performing the operations in a formula based on its assigned precedence to each operator. Some of the most important operators and their precedence are as follows:

Operator	Description	Precedence
=, < >	Equal, not equal	3
<, >	Less than, Greater than	3
<=	Less than or equal	3
>=	Greater than or equal	3
* /	Multiplication, Division	4
-,+	Subtraction, Addition	5
-,+	Negative, Positive	6
^	To the power of (exponentiation)	7

It is possitble to override the precedence of operations by including parenthesis in your formula. Any oeperation surrounded by parenthesis is given highest priority. Parenthesis can be nested inside other parenthesis; innermost part is calculated first.

(b) Types of formulas: Spreadsheets work with three types of formulas, namely arithmetic, text and logical.

Arithmetic formulas are the most commonly used, and these calculate numeric values. They can use most of the @ functions (these are for more advanced calculations e.g. @SUM is used to total values of specified cells), and accept any of the following operators:

-+ * / ^ = < > <= >= <> #ND# #OR# #NOT#.

Text formulas are any formulas that result in a textual entry, while, logical formulas are true/false statements concerning value in other cells.

(c) @Functions: Spreadsheets provide many built-in functions called @functionss (because they are always preceded by an "at" sign - that perform advanced calculations) which you enter in spreadsheet cells, either alone or within formulas. They perform advanced calculations and provide the resulting value.

Spreadsheets provide several @functions which can be divided into six categories, namely: Mathematical, Aggregation and counting, Text (or String), Miscellaneous, Logical, Financial Date and Time and Database Aggregation. The syntax and arguments of the functions provided by spreadsheets can be displayed on the screen through the on-line help facility (F1) or by pressing the FUNCTIONS Key (Alt-F3 in QUATTRO).

A few examples of mathematical functions; aggregation and counting; and logical/string functions are given below:

Mathematical Functions:

Function	Returns
@ABS(X)	The absolute value of X
@EXP(X)	e raised to the power of X
@LN(X)	The log base e of X
@ROUND(X,N)	X rounded to the number of digits specified (up to 15)
@SQRT(X)	The square root of X

Aggregation and Counting Functions

Function	Returns
@AVG(List)	The average of list
@COUNT(List)	A value equal to the non blank cells in (list)
@MAX(List)	The maximum value in (list)
@SUM(List)	The sum of value in (list)

Logical/String Functions

Function	Returns
@IF(Cond,X,Y)	X if Cond is true, Y if Cond is false
@STRING(X,N)	Numeric value of X as a string, with N Decimal places

(d) Referencing: Spreadsheets allow two types of referencing of values in other cells viz: relative and absolute referencing.

In relative referencing, copying a cell or block of cells that includes a formula and the cells referenced in the fomula to a new location results in the formula being updated to reference the new cell locations. In the new location, the spreadsheet will now refer to the addresses of referenced cells relative to their location in the copied cell. For examples

* If the cell D8 contains the formula +D5*C8/(1+(D5-1)*C8) and this is copied to cell E10, then the spreadsheet will adjust the formula in cell E10 to read +E7*D10/(1+(E7-1)*D10)

* If the cell C2 contains the formula +A2+B2 and this is copied to cell D2, the spreadsheet will adjust the formula in cell D2 to read +B2+C2. On the other hand, if the formula in cell C2 is copied to cell F10, the spreadsheet will adjust the formula in cell F10 to read +D10+E10.

In absolute referencing, one uses the $ symbol before the cell address to indicate that when copying a cell or block of cells that includes a formula to a new location, the cell reference(s) with the $ symbol must remain the same. One may make part or all of an address absolute as in the following examples:

$D$5 makes the entire address absolute
$D5 makes only the column absolute
D$5 makes only the row absolute

* If the cell D8 contains the formula +$D$5*C8/(1+($D$5-1)*C8) and this is copied to cell E10, the spreadsheet will adjust the forumla in cell E10 to read +$D$5*D10/(1+($D$5-1)*D10)

* If the cell D8 contains the formula +$D5*C8/(1+($D5-1)*C8) and this is copied to cell E10, the spreadsheet will adjust the formula in cell E10 to read +$D7*D10/(1+($D7-1)*D10).

* If the cell D8 contains the formula +D$5*C8/(1+(D$5-1)*C8) and this is copied to cell E10, the spreadsheet will adjust the formula in cell E10 to read +E$5*D10/(1+(E$5-1)*D10).

Given the foregoing, one has to be particularly careful in referencing cells in formulas which may be copied to other locations.

3.3 Entering/Editing Data

3.3.1 The Worksheet Screen

The worksheet is the core of a spreadsheet software. It is in the worksheet that you enter, display and organize information. The worksheet screen has four distinct areas: the worksheet window, in input line at the top, and the descriptor and status lines at the bottom. Only the worksheet window is of concern to us here.

The worksheet window takes up most of the display screen. It is through here that one views and makes changes to the spreadsheet software. As stated earlier on, only part of the entire worksheet is shown in the window at once. To view other areas, one has to scroll the worksheet, using the directional arrow keys, or, using the GOTO commands.

Typically, the worksheet itself is a rectangular grid made up of numbered rows and columns identified by letters. Each intersection of a row and column makes up a cell, in which data can be stored. The worksheet is made up of the following essential components:

(i) Horizontal boarder: This is a row of letters that identifies each column of cells in the worksheet. Columns are lettered A-Z, then AA-AZ, BA-BZ and so on, up to IV. The column containing the cell selector is usually shown in reverse video, or a different colour.

(ii) Vertical boarder: This is a column of numbers (1-8192) that identifies each row of cells in the worksheet. The row containing the cell selector is ususlly shown in reverse video, or a different colour.

(iii) Cell: This is a box in the worksheet in which one can enter data. Each cell is identified by an address determined by the row and column that contain

it. For example, the cell in the upper-left corner of the worksheet is named A1. The letter and number that make up the cell address is called the cell's coordinate.

(iv) Cell selector: This is the highlighted rectangle that indicates the current cell. This is the cell that will be affected by one's next action, such as entering a value. One can move the selector to any cell in the spreadsheet.

(v) Cell block: This is a rectangular group of cells identified by the cell addresses of two opposite corners: usually the upper-left and bottom-right cells. Blocks are used in commands and formulas to act on several cells at once. For examples:

* A11..D11 is a cell block that includes the cells A11, B11, C11, and D11.
* C10..C15 is a cell block that includes the cells C10, C11, C12, C13, C14 and C15
* A10..C12 is a cell block that includes cells A10, B10, C10, A11, B11, C11, A12, B12 and C12

3.3.2 Entering Data

To enter data in a worksheet, use the direction or arrow keys to move the cell selector to it and type the entrty. The characters you type appear on the input line above the worksheet. The input line, located at the top of the worksheet screen, displays various types of information, depending on what you are doing such as entering or editing data in a cell, using a menu or executing a command). Use the Backspace key to erase mistakes as you type. When you finish entry, you press Enter or one of the direction keys; the data is then written into the highlighted cell.

A spreadsheet software will automatically go into Edit Mode if it discovers a problem with your entry such as illegal character. It won't accept the entry until you have corrected the problem. To cancel entry, press the Esc key.

3.3.3 Editing Entries

Once an entry has been written into a cell (by pressing Enter) you can erase it simply by moving the selector to it and pressing Del. You can change its contents in two ways.

1. Move the cell selector to the cell and enter a new value. When you press Enter (or any of the arrow keys), the new value replaces the old.
2. Make changes to the existing entry using the Edit mode. This is invoked by pressing the EDIT key (F2 in QUATTRO or LOTUS 1-2-3). In the Edit Mode, you can insert or delete characters in an entry without retyping the whole thing. Many functions are provided in the edit mode. Time and space limitations will, however, not permit us to discuss these here. (In QUATTRO, you can obtain information on the Edit mode by pressing F2, followed by F1 the Help key).

4. Applications of Spreadsheets

Spreadsheet software have been employed for a wide range of applications in the literature, for examples: Goldfarb (1985), Julian (1989), Ferral et al. (1986), Cheremisinoff (1987), Sowa (1987), El-Shayal (1988), Heck (1989), Knudsen (1991), Hughes (1991), Rioux (1991), Ryswyk (1991), Rosenthal (1991), Kuye (1992,1993), etc.

To illustrate the use of spreadsheets in engineering analysis and computations, only three examples developed (by the author) using the QUATTRO Ver. 1 software are presented here.

4.1 Example 1

Solving the quadratic equation:

$$ax^2 + bx + c = 0 \quad (a = 0) \quad \text{.....(1)}$$

is a very common problem.

The solution (roots) of the quadratic equation may be real or complex.

When the roots are real i.e. $b^2 - 4ac \geq 0$ they can be computed by the equations:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{2}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \tag{3}$$

However, if $b^2 - 4ac < 0$, the two roots are complex and can be divided into their real and imaginary parts as follows:

$$x_1 = R + jI \tag{4}$$

$$x_2 = R - jI \tag{5}$$

where

$$R = b/2a \tag{6}$$

$$j = \sqrt{-1} \tag{7}$$

$$I = \sqrt{4ac - b^2}/2a \tag{8}$$

Obviously, this simple problem can be solved on an ordinary scientific calculator. However, imagine a situation where one has to solve several quadratic equations (say 40 or more) for the two roots. Manually calculating the roots then becomes tedious and prone to errors. A spreadsheet solution which avoids these problems is as follows:

Start with a clean worksheet screen. Type the title "Table 1 Solution of quadratic equation", and "ax^2 + bx + c = 0' ' in cells A3 and A4, respectively.

Type "a", "b" and "c" in cells A6, B6 and C6, respectively. Type "ROOT1" and "ROOT2", "d" and "b"/2a" in cells D6, E6, F6, and G6, respectively. For each quadratic equation to be solved, enter the values of a,b,c starting from cells A7, B7 and C7, respectively. The entry should then appear as in Table 2.

Now move to cell F7 and type in the formula: +B7^2-4*A7*C7 to calculate the discriminant, d. Move to cell G7 and type in the formula: +B7/(2*A7) to calculate b/2a. Then enter the following formulas:

@IF(F7>=0,G7+@SQRT(F7)/(2*A7),@STRING(G7,2)&"-j"&@STRING(@SQRT(-F7)/(2*A7),2))

@IF(F7>=0,G7-@SQRT(F7)/(2*A7),@STRING(G7,2)&"-j"&@STRING(@SQRT(-F7)/(2*A7),2))

in cells D7 and E7, respectively. These formulas calculate the two roots for the first entry of a,b,c in cells A7..C7. To calculate the roots for the other entries of a,b,c in cells A8.. C46, use the /BC (/ Block Copy) command sequence to copy the source cells: D7..G7 to the destination cells: D8..G46. The spreadsheet program will automatically calculate the roots for the remaining equations. These roots are displayed in columns D and E of the worksheet under the headings ROOT1 and ROOT2. As a remark, it is interesting to note how we have employed the use of the @STRING functuion to handle the case of complex roots. A partial print out from the spreadsheet solution is shown in Table 2.

Having setup the spreadsheet solution as above, if you change or enter new value(s) for a,b,c, the spreadsheet program will immediately and automatically calculate the new roots.

4.2 Example 2

The kinematic equations describing an engine system in which the crank rotates with a constant angular speed of θ rad/s counterclockwise are given by Fletcher and /Shoup (1978) as follows:

$$\text{Displacement}, x = R_1 \cos\theta + \sqrt{R_2^2 - R_1^2 \sin^2\theta} \qquad (9)$$

$$\text{Velocity}, v = \dot{x} = \frac{\dot{\theta} R_1 \sin\theta}{R_1 \cos\theta - x} \qquad (10)$$

$$\text{Acceleration}, a = \ddot{x} = \frac{\dot{x}^2 + 2R_1\dot{x}\dot{\theta}\sin\theta + R_1 x\dot{\theta}^2\cos\theta}{R_1\cos\theta - x} \qquad (11)$$

$$R_2 > R_1$$

It is desired to plot the displacement, velocity and acceleration of the piston for 5-degree positions of the crank.

Fletcher and Shoup (1978) presented a FORTRAN program to calculate x, v and a, while Williams (1993) presented a Casio Fx-502P calculatror program for the same problem. However, both the FORTRAN and the calculator programs are not capable of generating the graphical plots of the computed values. Here, a spreadsheet solution, from which one can directly obtain the graphical plots of the computed values is presented.

Start with a clean worksheet screen. Type the title: "Table 3: Kinematic analysis of an engine system" in cell B3. Move to cell A4 and type "Engine speed, thetadot (rad/s)=", then enter the value of engine speed (15) in cell E4. Move to cell A6 and type "Short link length, R1(cm)=", and then enter the value of R1(1.0) in cell E6.

Type "theta", "x", "v" and "a" in cells A10, B10 and C10, respectively. Now, either manually enter 0.0 to 360.0 in steps of 5 in cells A12..A84, or use the /BF (Block Fill) command sequence to automatically fill the cells. Then type the following formula:

@IF($E$5>$E$6,$E$6*@C0S(A12)+@SQRT($E$5^2-$E$6^2*@SIN(A12)^2,"R2<R1,STOP") in cell B12,

($E$4*$E$6*@SIN(A12))/($E$6*@COS(A12)-B12) in cell C12, and

(C12^2+2*$E$6*C12*$E$4*@SIN(A12)+$E$6*B12*($E$4^2)*@COS(A12))/($E$6*@COS(A12)-B12)

in cell D12, to calculate values of x,x and a, respectively for angle = 0 deg. (A12). To calculate x,v and a for the other values of crank angle, use the /BC (Block Copy) command sequence to copy the source cells: B12..D12 to the destination cells B13..D84. The spreadsheet program will automatically then calculate x, v and a. These values are displayed in columns B, C and D under the headings "x", "v" and "a". A partial print out from the the spreadsheet solution is shown in Table 2.

As pointed out earlier, a unique advantage/ feature of spreadsheet programs is that one can easily generate various graphical plots (i.e. Line, Bar, 3-dimensional bar, XY, Stacked bar, Pie, Area, etc.) from the data in the spreadsheet. An example of how to make a graphical plot from within a spreadsheet is as follows.

To plot the displacement x versus θ ,

* First use /GRG (/ Graph /Reset Graph) command sequence to return the entire graph to its default mode, or menu settings
* Press G to select the Graph type
* Choose the XY plot option from amongst the 10 displayed
* Select X-axis values and enter A12..A84
* Press S to select Series values

* Press 1 to select 1st Series and enter B12..B84
* Press V (view) to display the graphical plot in the screen. Press the Esc key to return to the worksheet
* You may use the customize commands in the Customize Menu to embellish your graph with an array of options, including grids, titles, legends, marker symvbols, etc.
* To obtain a hard copy of the graphical plot, make use of the /GP (/ Graph Print) com mand sequence and respond to the various prompts to set the printer, layout, destination etc. and then finally printing.

The itemized procedures above can be used to similarly generate the plot of v versus θ, and a versus θ. One can save each graph so generated. The three plots generated this way are shown in Figures 1a, b and c.

Having setup the spreadsheet solution as above, if any of the values in cells E4, E5 or E6 i.e. engine speed, long and short link lengths, the spreadsheet will immediately and automatically recalculate the values of x,v and a in cells B12..B84, C12..C84 and D12..D84, respectively. These new changes will also be reflected in the graphical plot the next time these are viewed or plotted.

4.3 Example 3

Finding the Steady-State(s) of a Continuous Stirred Tank Reactor (CSTR).

A frequent task in engineering analysis and design is the numerical computation of root(s) of an equation of the form

$$F(x) = 0 \qquad (12)$$

The most commonly employed numerical techniques for this are the interval bisection method and the Newton-Raphson algorithm (cf. Forsythe et al., 1977; Gerald, 1978).

While one would typically program these algorithms on a digital computer (in FORTRAN, BASIC or any other suitable, higher level language) to solve the nonlinear equation, the graphical capability and the ability to automatically iterate to convergence (without the need for iterative loops) make spreadsheets ideal tools for solving for the root(s) of a nonlinear, or any other trial-and-error procedure/design method, quite easily. This capability shall be illustrated here for the problem of calculating the steady-state concentration in a CSTR. Even though the Newton-Raphson algorithm has a faster convergence than the interval bisection method, lit is sensitive to the choice of the initial guess value, and may therefore diverge for a bad guess. Furthermore, one requires the derivative F'(x) which may not be easily obtained, especially for a complex function. On the other hand, the interval bisection method is slower to converge than the Newton-Raphson ;algorithm. However, it is guaranteed to always converge to a desired root, provided one can determine an interval (a,b) where the root lies, and in which F(a) < 0 and F(b) > 0. This can be easily done on a spreadsheet. All the intervals which meet this requirement can be obtained by computing F(x) in the interval x E (A,B) where all the roots of F(x) are desired, and graphically displaying the plot of F(x) versus x within the spreadsheet program. This approach is used here.

The governing equation for the reactor of interest is that given by Carberry (1976)

$$F(x) = \frac{1}{T}(x_o - x) - \frac{kx}{(1+Kx)^2} = 0; \quad T = V/F \qquad (13)$$

where

x_o = Reactant inlet concentration, kmol/m³
V = Reactor volume, m³
F = Feed flow rate, m³/s
x = Reactant exit concentration, kmol/m³
k,K = Kinetic rate constants

Depending on the values of xo, V, F, k and K, the above equation may have one, two or three real roots which are the steady-states of the reactor. When more than one real root exist, the reactor is said to exhibit multiple steady-states. An example data is V = 10m³, F = 0.2m³/s.

Start with a clean worksheet. Use the /DRMM (/ Default, Recalculation, Mode, Manual) to set mode of calculation by the spreadsheet to manual i.e. formulas will only be re-calculated by the spreadsheet when you press the CALC key (F9 in QUATTRO).

Type the title "Table 3: Finding the Steady-States of a CSTR" in cell A3. Type "Inlet concentration, X0, kmol/m³ = ", "Reactor volume, V, m³ = ", "Feed flow rate, F, m³/s = ", "k = ", "K = ", in cells A5, A6, A7, A8 and A9, respectively.

Now, move to cell F5 and enter the value of X0 i.e. 13.00;

Move to cell F6 and enter the value of V i.e. 10.00

Move to cell F7 and enter the value of F i.e. 0.20

Move to cell F8 and enter the value of k i.e. 1.0

Move to cell F9 and enter the value of K i.e. 1.0

Type "x", "F(x)" in cells B14 and C14, respectively. For the given problem data, all the desired, real roots of (13) must lie in the interval [0,xo]*.

* This is based on physical considerations as follows: x being concentration cannot be less than zero; being also reactant exit concentration, x cannot be greater than inlet reactant concentration.

Therefore use the /BF (/ Block Fill) command se-

quence to automatically fill the cells B12..B77, starting from 0, step increment of 0.2, and ending at 13.00. Then type the formula ($E$5-B12) /($E$6/$E$7)-$E$8*B12/(1+$E$9*B12)^2 in cell C12 to calculate F(x) for x = B12 i.e. x = 0. To calculate F(x) for the other values of x in cells B13..B77, use the /BC (/Block Copy) command sequence to copy the source cell C12 to the destination cells C13..C77. The spreadsheet program will automatically compute these values and display them in cells C13..C77.

Now use the /G (/ Graph) options to define the x-axis (B12..B77) and the 1st Series (C12..C77) to an XY graph as described in Example 3, and then press V to display a plot of F(x) versus x on the screen. This plot is shown in Fugure 2.

Note the intervals where F(x) changes sign as [a_1, b_1] = [1,0]; [a_2, b_2] = [1,4]; and [a_3, b_3] = [10,4].

We can see that there are three jreal roots. Now press Esc to move back to the spreadsheet.

Next, type "a", "F(a)", "b", "F(b)", "x1", "x2", "Root, x", and "F(x)" in cells E11, F11, G11, H11, I11, J11, K11 and L11, respectively. Type "i", "ii", and "iii" in cells D12, D13, and D14, respectively. Enter i, 1, and 10 in cells E12, E13 and E14, respectively. Enter 0, 4, 4 in cells G12, G13, and G14, respectively. As it is, cells E12, G12; E13,G13; and E14;G14 now contain the intervals [a,b] for the three real roots. Enter the following in cells F10 and I10, respectively, "Iteration number = ", +K10+1. Next, enter the following formulas in the cells indicated:

$E$5-E12) /($E$6/$E$7)-$E$8*E12/(1+$E$9*E12) -2 in cell F12

($E$5-G12)/($E$6/$E$7)-$E$8*G12/(1+$E$9*G12)-2 in cell H12

@IF($I$10=1,E12,@IF(L12<=0,K12,I12)) in cell I12

@IF($10=1,G12,@IF(L12>=o,K12,J12)) in cell J12

(I12+J12)/2 in cell K12

($E$5-K12)/($E$6/$E$7)-$E$8*K12/(1+$E$9*K12)-2 in cell L12

Then use the /BC (/ Block Copy) command sequence to copy the source cells I12..L12 to the destination cells I13..L14. Now use the /DROC (/ Default, Recalculation, Order, Columnwise) command sequence to set the order of calculation by the spreadsheet to columnwise. Now, press F9 repeatedly to recalculate the formulas in cells:I12..L14. Notice as the spreadsheet program iterate on the values in cells I12..L14. You can continue to press F9 until the desired accuracy in the roots in cells: K12..K14 are attained. Cell I10 then shows the required number of iterations. At the end of the iteration, cells K12..K14 contain the three real roots which are 0.7515, 2.1309 and 8.1175 to the fourth decimal place. These required just 18 iterations. The corresponding values of F(x) evaluated at the three roots in cells K12..K14 are shown in cells L12..L14. A partial print-out from the spreadsheet program (after convergence) is shown in Table 6.

Setting up the problem as described above, takes just about 15 minutes; while the convergence is achieved in about 45 seconds.

5. Conclusions

The aims of this paper were to provide a brief, but functional overview of spreadsheet software, together with illustration of their applications/use in scientific and engineering computations. The paper discussed what spreadsheets are; their unique advantages (such as "unlimited" size, automatic calculation/iteration, easy editing and data entry, powerful commands, graphical capability etc.) which makes them suitable for scientific/engineering computations; types of spreadsheet softwawre; allowable data types and how to enter these into a spreadsheet, etc. Three examples (with detailed, step-by-step procedure of how to set up the spreadsheet solution were used to demonstrate the suitability and versatility of spreadsheet software in scientific and engineering analysis/computations. The examples clearly demonstrate that a spreadsheet software can be used to readily solve a variety of analysis/design computation problems (including thos requiring trial-and error or iterative techniques) that would have typically require extensive programming (using either FORTRAN or BASIC) and debugging. It is hoped that the paper has achieved its aims, and that the enthusiastic student, professional scientiest and engineers will be able to adopt spreadsheets for performing their day-to-day analysis/design computations where complex or very complex computations are not necessary. For such situations, the use of spreadsheet software is often easier, adequate, much less time-consuming, and less frustrating than writing a program in FORTRAN, BASIC or C.

References

Cheremisinoff, N.P. (1987). Statistical regression routines on a spreadsheet, Chem. Eng., Aug. 7, pp. 153 - 157.

Chorafas, D.N. (1990). The New Technology - a survival guide to new material, supercomputers & global communications for the 1990s, Sigma Press, Wilmslow, U.K.

Ferral, J.F. Pappano, A.W. and Jennings C.N. (1986). Process analysis on a spreadsheet Chem. Eng., March 3, pp. 101-10.

El-Shayal, I. (1988). Rigorous engineering calculations on a spreadsheet, Chem. Eng., Jan. 18, pp. 117-119

Fletcher, L.S. and Shoup, T.E. (1978). Introduction to Engineering including FORTRAN Programming, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, pp. 199 154

Forsythe, G.E., Malcolm, M.A. and Moler, C.B. (1977). Computer Methods for Mathemtical Computations, Prentice-Hall, Englewood Cliffs, New Jersey

Gerald, C.F. (1978). Applied Numerical Analysis, second edition, Addison-Wesley Publishing Company, New York

Goldfarb, S.M. (1985). Spreadsheets for chemical engineers: Design calculations, Chem. Eng., April, 15, 1985, pp. 91

Heck, G.F. (1989). Simulating feedback control, Chem. Eng., May 9, 1989, pp. 99-101

Hughes, E. Jr. (1991). A cumulative count method for determining the half-life of Barium-137 and Gallium-68 radioactive isotopes: A spreadsheet application, J. Chem., Educ., Vol. 68, No.2, pp. A41-A42

Julian, F.M. (1985). Flowsheets and spreadsheets, Chem. Eng. Prog., Sept. 1985, pp. 35.

Julian, F.M. (1989). Process modelling on spreadsheets, Chem. Eng. Prog., Oct. 1989, pp. 33

Knudsen, S.K. (1991). Direct solution of Schroedinger equation for vibration via spreadsheet, J Chem. Educ., Vol 68 No. 2, pp. A39 - A41

Kuye, A. (1992). Quality control calculations on a spreadsheet, Proc. 1st Inter. Conf. of Nig. Soc. of Chem. Engrs., Port Harcourt, 4th - 7th, Nov. pp. 100 - 122.

Kuye, A. (1993). A microcomputer model for project evaluation, NSE Tech. Trans., Vol. 28 (2), pp. 55-63

Ravella, A. (1993). Use a spreadsheet for preliminary reactor design, Chem Eng. Prog.., Feb., pp. 68-73

Rioux, F. (1991). Quantum mechanics with a spreadsheet, J. Chem. Educ., Vol. 68, No. 11, pp. A282-A283

Resenthal, J. (1991). Spreadsheet calculations for X-ray powder diffraction patterns, J. Chem. Educ., Vol. 68, No. 11, pp. A285-A286.

Ryswyk, H.V. (1991). Simulation of single-step chronoamperometry via spreadsheet, J. Chem. Educ. Vol. 68, No. 11, pp. A283-A285.

Sowa, C.J. (1987). Engineering calculations on a spreadsheet, Chem. Eng., March 2, pp. 61-63.

Stair, R.M. (1986). Computers in Today's World, IRWIN, Homewood, Illinois.

Williams, A.O.F (1992). An introduction to spreadsheet and application to computation of B.O.Q., invited paper presented at the Short Course on Application of Computer Packages in Engineering, organized by the Civil Engineering Division of the Nigerian Society of Engineers, 16 - 17th. July, 1992, at the National Engineering Centre, Victoria Island, Lagos, pp. 79-94 of Course Handbook.

Williams, A.O.F. (1993). An introduction to programmable calculators and their applications in engineering and scientific computations, paper presented at the 2nd. School of Engineering Staff Seminar Series, 9th. Dec., 1993, Lagos State Ikosi Campus, Lagos.

Williams, A.O.F. (1994). Computer applications in waste management, paper presented at the Workshop on "Effective Waste Management", organized by the Nigerian Society of Engineers, Ikeja Branch, 20th - 22nd. April, at the Nigerian Educational Research Council (NERDC), Central Business District, Alausa, Lagos.

	A	B	C	D	E	F	G
1							
2							
3	Table 1:Solution of quadratic equation						
4	a*x^2 + bx + c = 0; a <> 0						
5							
6	a	b	c	ROOT1	ROOT2	d	b/2a
7	1.00	2.00	3.00	1.00+j1.41	1.00-j1.41	-8.00	1.00
8	3.00	4.00	5.00	0.67+j1.11	0.67-j1.11	-44.00	0.67
9	2.00	3.00	1.00	1.00	0.50	1.00	0.75
10	2.00	1.00	3.00	0.25+j1.20	0.25-j1.20	-23.00	0.25
11	3.00	2.00	1.00	0.33+j0.47	0.33-j0.47	-8.00	0.33
12	4.00	5.00	3.00	0.63+j0.60	0.63-j0.60	-23.00	0.63
13	1.00	5.00	1.00	4.79	0.21	21.00	2.50
14	2.00	2.00	2.00	0.50+j0.87	0.50-j0.87	-12.00	0.50
15	3.00	1.00	3.00	0.17+j0.99	0.17-j0.99	-35.00	0.17
16	1.00	3.00	1.00	2.62	0.38	5.00	1.50
17	1.00	7.00	1.00	6.85	0.15	45.00	3.50
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
46	9.00	21.00	4.00	2.12	0.21	297.00	1.17

	A	B	C	D	E	F	G
1							
2							
3	Table 2:Kinematic analysis of an engine system.						
4	Engine speed, thetadot(rad/s) =		15.00				
5	Long link length, R2 (cm) =		2.50				
6	Short link length, R1 (cm) =		1.50				
7							
8							
9							
10	theta	x	v	a			
11							
12	0.00	4.00	0.00	-540.00			
13	5.00	2.47	10.55	52.57			
14	10.00	1.10	5.18	174.67			
15	15.00	1.16	-6.36	192.72			
.	.	.	.	.			
.	.	.	.	.			
.	.	.	.	.			
83	355.00	1.00	0.00	135.00			
84	360.00	1.62	-10.55	244.04			

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2												
3	Table 6:Finding the steady-state of a CSTR											
4												
5	Inlet concentration, X0, kmol/m^3 =				13.00							
6	Reactor volume				= 10.00							
7	Feed flow rate, F, m^3/s =				0.20							
8	k				= 1.00							
9	K				= 1.00							
10						Iteration number		= 18				
11		x	F(x)		a	F(a)	b	F(b)	X1	X2	ROOT,X	F(X)
12		0.00	2.60E-01	i	1.00	-1.00E-02	0.00	2.60E-01	0.75	0.75	0.7515	-1.030E-07
13		0.20	1.17E-01	ii	1.00	-1.00E-02	4.00	2.00E-02	2.13	2.13	2.1309	1.1660E-07
14		0.40	4.79E-02	iii	10.00	-2.26E-02	4.00	2.00E-02	8.12	8.12	8.1175	1.7087E-07
15		0.60	1.36E-02									
16		0.80	-2.9E-03									
17		1.00	-1.0E-02									
18		1.20	-1.2E-02									
19		1.40	-1.1E-02									
20		1.60	-8.7E-03									
.		.	.									
.		.	.									
.		.	.									
76		12.80	-6.3E-02									
77		13.00	-6.6E-02									

Figure Captions

Figure 1 Graphical plots for Example 2. (a) Top: x versus θ; (b) Middle: v versus θ; and (c) Bottom: a versus θ.

Figure 2 Graphical plot of $F(x)$ versus x for Example 3.

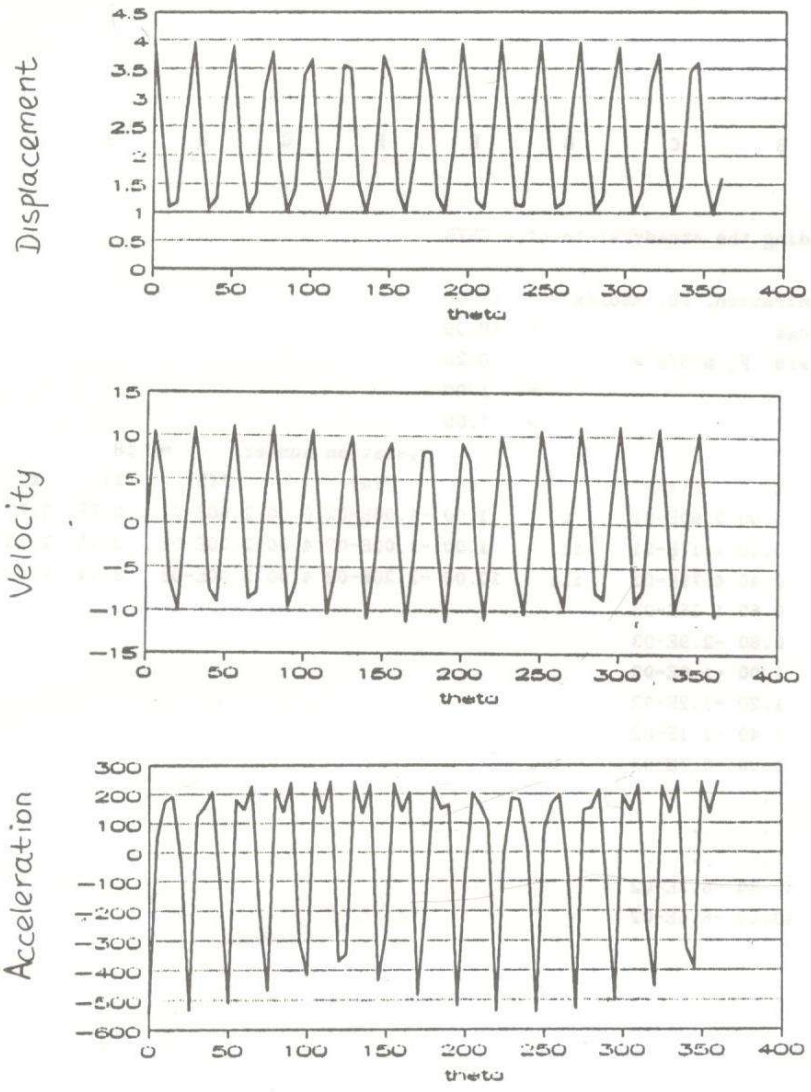

Displacement
4.5
4
3.5
3
2.5
1.5
1
0.5
0
0 50 100 150 200 250 300 350 400
theta
Velocity
15
10
5
0
−5
−10
−15
0 50 100 150 200 250 300 350 400
theta
Acceleration
300
200
100
0
−100
−200
−300
−400
−500
−600
0 50 100 150 200 250 300 350 400
theta
F(x)
0.3
0.25
0.2
0.15
0.1
0.05
0
−0.05
−0.1
0 2 4 6 8 10 12 14
X

Figure 2