

# 1

# Microsoft Excel Spreadsheet

The two computational essentials presented in this text are the Excel spreadsheet and the programmed procedures that, when written in the VBA (Visual Basic for Applications) macro programming language and attached to a workbook, can handle any programming chore. This text will concentrate on the engineering applications of structures and soils and their interaction. Application to any other area of specialization will be evident. The text will approach the teaching not inductively, with instructions in general theory, but deductively, with examples showing the specific aspect being addressed. The book is a learning tool as much as it is a reference guide, so it should be used in conjunction with the enclosed programs using the data in the examples, or other data created by the user to solve problems on the computer. To begin, the Excel spreadsheet will be covered in this chapter and the VBA programming language in Chapter 2.

## 1.1 History of Spreadsheet Development

Excel is a spreadsheet program developed by Microsoft and is part of Microsoft's Office suite. The earliest spreadsheets were VisiCalc and SuperCalc, which were first distributed in 1979 and 1980. Microsoft marketed its Multiplan in 1982 on CP/M-based systems. Shortly after, Lotus 1-2-3 became popular, eventually dominating on the newer MS-DOS systems. Borland's spreadsheet program, Quattro Pro, was also a big seller.

The Microsoft Windows operating system was introduced in 1985. Microsoft's first Windows-based spreadsheet program was introduced in 1987, renamed Excel, and called version 2.0. Eventually, Excel supplanted Lotus 1-2-3 as the industry standard for spreadsheet programs, and continues as such today. The 1993 version of Excel was the first to include VBA. Excel 95 was the first to support a 32-bit operating system. Excel 2007 introduced the Ribbon, a complete revamp of the user interface (UI) which proved to be a contentious change at first.

Other office suites that still feature a spreadsheet continue to be offered: Open Office.org (OOo – formerly StarOffice developed by Sun Microsystems), IBM's Lotus Symphony (which since 2010 has incorporated code from OpenOffice), and

LibreOffice, along with Google Docs, the online office suite. These spreadsheets are not a true substitute for the Excel spreadsheet in Microsoft Office, as support for VBA is not found in them, and neither is full compatibility with Microsoft's .doc(x) and .xls(x) formats. That said, many programs are capable of both reading and writing .xls and .xlsx files. The strength and reasonable price of Microsoft Excel software make it the only choice.

## 1.2 Excel 2010

Beginning with Windows XP, Microsoft provided the choice of either a 32- or 64-bit version for the operating system. It was not until Microsoft Office 2010 that both 32- and 64-bit versions of Excel were offered. The 32-bit version enables the use of solutions built in previous versions of Microsoft Office without modification. CPUs sold today from AMD and Intel are 64/32-bit capable and will run either 32- or 64-bit Windows.

The 64-bit version of Microsoft Office 2010 requires use of the 64-bit Windows 7 operating system on a computer with a 64-bit capable processor. The 64-bit version can address a much larger memory than the 4 GB that a 32-bit processor can. By default the 32-bit version of Microsoft Office 2010 is installed unless a custom installation is performed. Such a version is supported by Microsoft and can handle files created by older versions as well as work on 64-bit operating systems such as Windows 7.

The word spreadsheet is the more common description for what Microsoft Excel refers to as a worksheet. Excel uses the word workbook to refer to a collection of worksheets. If only one worksheet is in a workbook it could be denoted as a worksheet or a workbook. All the worksheets and associated VBA macro procedures in this text have been developed in Excel 2010 but will run in Excel 2003 to 2010. Office 2003 does not read or write the default Office formats of the 2007 or 2010 versions, which is .xlsx for Excel, but the latter versions can readily read files created in the 1997–2003 versions and save them in the old format. Additionally, Microsoft has a Compatibility Pack for Office 2003, available from its Download Center, which allows the older Office suite to read and write the newer formats. The features in Excel 2007 and Excel 2010 are not very different other than the change to offering both 32- and 64-bit versions.

### *1.2.1 File Conversion and Compatibility*

One of the greatest advantages of the Excel–VBA package offered by Microsoft is its backwards and forwards compatibility with previous and new versions. As a transition from a 32-bit Windows XP to a 64-bit Windows 7 operating system is made, the ability to run programs created with the Excel–VBA package still remains. In this transition many users of programs created in one environment find that they cannot be run in the other and are forced to reuse their older computer because the drivers may have issues with the new program.

As stated previously, the reading and writing of Excel 2007 or later file formats using Excel 2003 is possible assuming that the Microsoft Compatibility Pack has been

downloaded and installed. The format for an Excel 2007 workbook file is \*.xlsx and \*.xlsm for a macro-enabled workbook. Office 2007 can open Office 2003 files and then convert and upgrade the \*.xls file to either a \*.xlsx or \*.xlsm file format. Changing from Excel 2003 to Excel 2007 (Office 2007) does not require any change or upgrade to the operating system, such as from Windows XP to Windows 7.

Excel 2010 provides a compatibility check under the File > Check for Issues button. One difference between the 2007 and 2010 versions of Excel is in defining the names for cells. For example, previously Rad2 was an acceptable name, but with the 2010 version it is not. This feature will be covered in more detail in Section 1.6.4 on naming cells.

### 1.3 Transmitting Cell Values Not Formulas

When sending a spreadsheet to others it may be beneficial to keep the calculations, and specifically the formulas in the cells, private, transmitting only the values in the spreadsheet cells. To copy values in cells that do not contain drawing objects or formulas from one worksheet to another using Excel 2003 requires the steps of selecting the cells and using Paste Special to paste the cell values. Alternatively, Adobe Acrobat, but not the free Adobe Reader, can be used to convert Excel 2003 spreadsheets to PDF files. Such a file allows one to send read-only versions of documents.

In Excel 2007, Microsoft created its own competing version of PDF denoted as XPS. An XPS reader, necessary to open such a file, comes with Windows Vista and Excel 2007, and can be downloaded for free from Microsoft.

Excel 2010 makes copying the values in a spreadsheet simple. Any portion of a worksheet that is set as a Print Area can be saved directly to a PDF file. Alternatively, the values of any portion of a worksheet can be copied using the Copy as Picture feature adjacent to the menu copy button in Excel 2010. By copying in this manner, cell formulas and structures are not copied.

### 1.4 Accuracy

Excel stores and calculates with 15 significant digits of precision and uses scientific notation for numbers ranging from E-307 to E+307. This degree of accuracy is the same as variables denoted with Double Precision in the VBA language macros, to be introduced in Chapter 2. Variables denoted as Single Precision in VBA have only seven digits of accuracy. A comparison of this 15-digit accuracy from the same calculation in an Excel worksheet and a VBA macro is presented in the workbook **eCalc**. All workbooks and their worksheets are contained in the DVD that is a part of this text.

Rounding numbers for currency calculations in a spreadsheet to a specific number of decimal places uses the ROUND() function. To round a value to two decimal places, the following statement is used:

```
=ROUND(H18*IntYr/PpY, 2)
```

An example of this rounding is shown in the worksheet **LoanPmtSchd** of **LoanProg** in Chapter 24.

## 1.5 Saving

The best way to save spreadsheets and attached VBA macro programs that are changed or updated is to use the same filename for all those performing a similar function and append with a revision number (1), (2) or -r1, -r2, and so on. The last one or two revisions should be saved in order to return to them, if needed.

When saving an older version file in the new format of Excel 2010, the ranges for the data selected for charting are not transferred and the data ranges must be reselected.

## 1.6 Implementation of Excel Features

### 1.6.1 General Tips

Using columns or rows to provide a type of format spacing is not good practice. Formatting of individual cells should be kept to a minimum.

It is best to use only one page orientation, portrait or landscape, between the worksheets of a workbook. Try to keep a consistent print size, scale, and page orientation so information can readily be transferred between cell locations in different worksheets.

Rather than making extensive use of headers and footers it is better to put such information directly in the spreadsheet.

When an analysis style has been developed, stick with it and use it as a template for other analytical spreadsheets. Much of the information does not change between systems being analyzed.

Hiding rows or columns in a worksheet results in less clarity and more confusion.

If the value in cell E24 were set equal to  $2 * \text{the value in cell E23}$ , and then cell E23 was replaced by cutting and pasting information from another cell, then cell E24 would become #REF!. The same result would not occur if the value in cell E23 were replaced by copying (and not cutting) and pasting information from another cell.

Absolute cell references, that are permanent if copied or cut, are differentiated by using the \$ symbol in front of either the row or column reference, or both. Relative cell references that change when moved do not have these constant row or column references.

When copying a table in Excel to a Microsoft Word 2010 text document, and then converting the Table to Text, either left or right click on the table and almost magically a Table Tools with Design and Layout sub-tabs appears under the Home tab (even if your eye was never up there looking for it). Under the Layout tab is a Convert to Text tab that will allow you to convert the table.

Links to other workbooks can be viewed and deleted by using Edit > Links > Source > Break. It is not advisable to link workbooks.

A series of values can be filled in by dragging the handle at the right of the cell using the right mouse button.

### 1.6.2 Fonts

Changing the font of a worksheet through Format > Cells > Font rather than through Format > Style > Modify Font yields two different-sized worksheets, even if the same font is used on each.

The Sans Serif Arial font is used for cells since it is more compact both horizontally and vertically than other similar sans-serif fonts like Calibri (a standard in Excel 2010) or Gill Sans MT.

The column width of cells in Excel is governed by the Standard Format font width in the Tools > Style setup page for the spreadsheet.

### 1.6.3 =IF Statements

In Excel there is no Go To branching, For statements or any kind of sequential steps – instead the entire spreadsheet is computed at once. The =IF statement is used to replace these logical statements that are not present. In Chapter 9, program **Class** has two versions. One version just uses a spreadsheet and the other version employs a VBA macro. The same types of IF statements are used in both in determining the result:

```
=IF(condition, action if True, action if False)
```

Logical functions can be made conditional by the statements AND and OR:

```
=AND (test1, test2) for testing to see if ALL of the
conditions are True.
```

```
=OR (test1, test2) for testing to see if ANY of the
conditions are True.
```

An example of both IF and AND is in the statement

```
=IF(AND(H721>F721,H721>F720), "OK", "Not OK")
```

The statement

```
=IF (B16=1, E16, IF(B16=4,E16,IF(B16=5, E16, 0)))
```

may be written as

```
=IF (OR (B16=1, B16=4, B16=5), E16,0)
```

The statement

```
=IF(AND(B118=1,C118=1),DenA-0.0624, IF(AND(B118=1, C118=0), DenA,
IF(AND(B118=2,C118=1),DenB-0.0624, =IF(AND(B118=2, C118=0)DenB,0))))
```

can alternatively be written as a nested IF statement:

```
=IF(B118=1.AND. C118=1, DenA-0.0624,IF(B118=1.AND.C118,DenA,
IF(B118=2,.AND.C118=1, DenB-0.0624, IF(B118=2.AND.C118=0, DenB, 0))))
=IF(C89=1,0,IF(D89=1,0,-6*E*I)
```

Examples of nested IF statements are as follows:

```
=IF(N67<0.3,"rigid",IF(N67>1.3,"flexible","intermediate"))
```

```
=IF(C89=1.0,IF(D89=1,0,-6*e*i))
```

Rather than nest several IF statements, it is often easier and clearer to separate the nested operations into a series of worksheet cells with a specific single operation being performed in each cell.

### 1.6.4 Naming Cells

It is often useful to “name” a cell so that a name rather than a cell location can be referred to in a formula. Doing so makes formulas and equations much easier to understand. One of the differences between the 2007 and 2010 versions of Excel occurs in defining the names for cells. Previously Rad2 and ks1 were acceptable names. With the 2010 version they are not. Now it is necessary to change the variable names Rad2 and ks1 to Radi2 and k. The names Rad2, R2, RR2, Ra1, ks1, and k1 are all unacceptable. IRR cannot be used as a variable since it is the Internal Rate of Return function. Now all numbered names must have at least four characters. In lieu of names containing numbers, an alternative naming convention would be to change the numbers to the letters a, b, and so on. The simplest way to make this change is to use the Name Manager and edit the name. Alternatively, in older versions of Excel, all Rad2 can be found and replaced with Radi2 and then the Rad2 name definition deleted, and Radi2 defined as a name.

### 1.6.5 Functions

Functions require the use of some simple VBA programming statements so their introduction in this chapter is cursory. Functions that are implemented by Procedures that store VBA code consist of the following two types:

- **Sub Procedures:** These will be covered in Chapter 2. Such procedures do not return a value to a cell unless the code specifically contains such a step.
- **Function Procedures:** Such procedures written in VBA return a value to the cell of a worksheet through the name of the function. Worksheet variable values are transferred to the VBA procedure through variable names enclosed in parentheses. These procedures are not identified as a macro when listing macros but when saving they must be in a macro-enabled workbook.

```
Function C(A as Single, B as Single), as Single
C = SQR(A + B)
End Function
```

In a worksheet enter the following in a cell

```
= Function C(3, 4)
```

to return a value for C (the value is 5), to that cell. This is different from a Sub Procedure that does not return value. VBA code is entered into a module where they

are stored. A good example of a Function Procedure is in the **LoanProg** program presented in Chapter 24. This Function Procedure employs an iterative routine to calculate interest. There will be more discussion of this topic in Chapter 2. In that chapter VBA macro programming will be integrated with Excel worksheets to show how they function as the palette for data input and output as well as program control.

### 1.6.5.1 Predefined Functions

Excel also provides functions that are predefined formulas or equations. They may be in Math and Trig or Financial or various other fields. The functions contained in the Analysis Toolpak and the Analysis Toolpak –VBA are automatically enabled in Excel versions beginning with 2007.

Matrix algebra operations such as transpose, multiply, and inversion cover arrays defined by a range of rows and columns. When using the math functions MMULT or MINVERSE from the Insert > Function > Math and Trig menu bar in Excel for multiplication or inversion of matrices, do not enter braces { }. First highlight the destination matrix and then enter the =MMULT or =MINVERSE function with the appropriate arrays. This function is entered by pressing Ctrl+Shift+Enter simultaneously. The braces will automatically be entered during the CtrlShiftEnter step. They indicate an array formula.

Examples that show the use of the MINVERSE and MMULT functions available in Excel are Examples 6.1a, 6.2a, 7.1a, 16.2b and 18.7b for a laterally loaded pile. These examples show two different ways of solving a problem: the usual way using program **PFrame** and using the MINVERSE and MMULT functions available in Excel. This alternative workbook approach sets up the constitutive equations and solves them directly using the Excel functions.

### 1.6.6 Drawing

In creating a drawing in Excel, always use the drawing menu items rather than relying on Format > Cells > Border to create a “line” image since cell spacing can change and border information may not then properly fit. To simultaneously delete many drawing objects such as lines and arrows, in Excel 2010 use Home > Find & Select > Select Objects > a rectangle to enclose and select all objects being deleted and then a delete key sequence.

It seems there is a slight compatibility problem that exists between the 2003 and 2007 or 2010 versions of Excel with worksheets that contain drawings. Various users have noted that the drawings can be stripped when a file contains them. This usually occurs when drawings are created in two different versions. This problem was not apparent in Excel 2003 worksheets.

### 1.6.7 Charting

The chart type used exclusively in this book is XY (Scatter). The Chart feature of Excel does not fit all situations. It cannot be readily used to graph or plot a series of lines such as when trying to plot a structure’s configuration. For each line segment the

endpoints must be given and one ends up with as many data sets defined as there are number of line segments. Therefore it is impossible to set up the plot generally for any structure comprising a variable number of line segments. In Chapter 2 a method will be presented whereby such lines can be readily plotted to the Excel worksheet using a VBA macro program.

The fitting of a curve through the plotted data can be performed using the Add Trendline feature of Excel Charts. Right click on the curve and the option of an Add Trendline appears with the following choices:

Linear  $y = ax + b$

Logarithmic  $y = a \ln x + b$

Polynomial  $y = ax^n + b$

Power  $y = ax^b$

Exponential  $y = a e^{bx}$

Moving Average no equation, an  $n$  period = 2 – 19.

The equation can be plotted on the chart using the Options box under Add Trendline.

#### **Keyboard Shortcuts**

F1 – Help

F2 – Activates the information in a cell for Editing

F3 – Paste name from workbook

F5 – Go To

F7 – Spelling and Grammar

F11 – New chart worksheet

F12 – Save As

#### **File Operation Shortcuts – initiated by Ctrl key plus letter (simultaneously):**

Ctrl N – New

Ctrl O – Open

Ctrl P – Print

Ctrl S – Save

#### **Cell Editing Shortcuts**

Ctrl ~ – Displays formulas in every cell

Ctrl 1 – Format cell

Ctrl C – Copy

Ctrl F – Find

Ctrl G – Go To

Ctrl H – Replace

Ctrl V – Paste

Ctrl X – Cut

Ctrl Y – Redo last action

Ctrl Z – Undo last action





### **Related Workbook on DVD**

eCalc with VBA macro stored in Module1.

## **Further Readings**

Bloch, S. (2000) *Excel for Engineers and Scientists*, John Wiley & Sons, Inc.

Gottfried, B.S. (2003) *Spreadsheet Tools for Engineers Using Excel*, McGraw-Hill.

