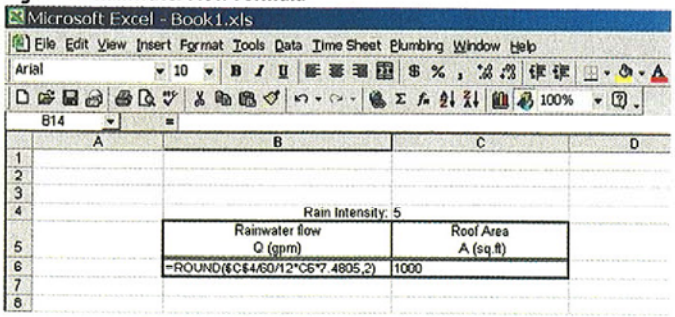
Using Excel Functions in Plumbing Engineering

Anjian Lu, CPD

Plumbing engineers often must use formulas such as the Hazen-Williams Formula, Manning's Formula, and the Hardy Cross method to determine piping friction loss and flow capacity, check water pressure and size pumps, convert fixture unit to flow rate, calculate equivalent length, hot/cold water mixing, and fuel gas friction loss, among many other calculations.

Figure 1. Rainwater Flow Formula



Before the advent of the computer era, scientists and engineers found laws or relations between relevant factors, expressed those relations with formulas, and established many tables, charts, and other aids for use in daily engineering tasks. But these charts are not always practicable. For example, to determine a pipe's friction loss you can consult a thick manual containing different pipe sizes and flow rate increments. But the manual only provides one Hazen-Williams C factor of 100 because a table incorporating small increments of every parameter would be too big and clumsy for use.

Luckily, today you can perform all of these calculations using computer programs. Microsoft Excel is one of the most powerful software programs for this purpose because of its inherent functions, User-Defined Functions (UDFs), and other capabilities.

Functions are the fundamental components of computer-aided computations. Many functions come pre-loaded on Excel, and users can write functions based on their own needs. Functions can be very simple procedures or quite long programs. In this article I use a simple function as an example, but the concepts work for longer functions as well.

Excel's Inherent Functions

In Excel 2000 and earlier versions, the Paste Function icon (*fx*) is located on the standard toolbar. Clicking this icon brings up a form containing common functions. Excel 2002 and 2003 do not have the Paste Function icon on the standard toolbar. You can click the Σ icon on the standard toolbar to enter the SUM() function or click the arrow beside it to show the pull-down menu with other functions. If you select the last item (More Functions), a form called Insert Function appears. You can either type a brief description of what you want to do and then click Go or select a category in the second form to show more functions. If you have UDFs, the last category will be User Defined. All of these are self-explanatory.

If you use Excel 2000 or earlier versions, in the Paste Function form, a function's variables and an explanation appear in the lower left corner when that function is selected. Whether in a cell or in the formula bar at the top of the workbook, functions should be preceded by an equal sign. Otherwise, Excel treats the entered numbers and functions as a text string. You can use as many levels of functions as you wish, such as: =ROUND(ABS(SIN(PI()/4)),2). This formula returns the value 0.71.

If you use Excel 2002 or 2003 and select a function from the Insert Function form and click OK, a Function Arguments form appears. You then can enter data for each argument and get the result directly. (As plumbing engineers, we use the term *variable* instead of *argument*.) But in most cases you can enter a cell address instead of a number, so the function reacts as the value in a cell is changing.

Entering a Formula

Let's calculate rainwater flow as an example of entering formulas. As rainwater flow is the product of rain intensity and collection area, the calculation is as follows:

$$Q = H/(60 \times 12) \times A \times 7.4805$$

where

Q = rainwater flow (gallons per minute)
H = rain intensity (inches/hour)
A = roof area (square feet)
60, 12, and 7.4805 = factors to convert the final unit to gpm

If the rain intensity is five inches/hour and the roof area is 1,000 square feet, and we enter them in cells C4 and C6 respectively, then the Excel function is:

=ROUND(C4/60/12*C6*7.4805,2)

ROUND is an inherent function, which has the format ROUND(number, digit). The first argument, number, is a number to be rounded and the second, digit, is for specifying the rounded number's decimal points.

You may enter the formula in the formula bar or directly in cell B6 (see Figure 1). Hit the enter key or move the cursor out of cell B6, and the value 51.95 appears. Figure 1 is in Formula

Figure 2. Rainwater Flow Calculation Sheet

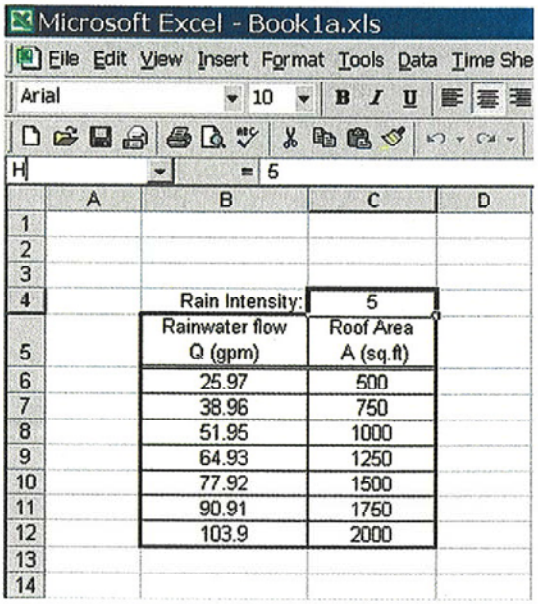
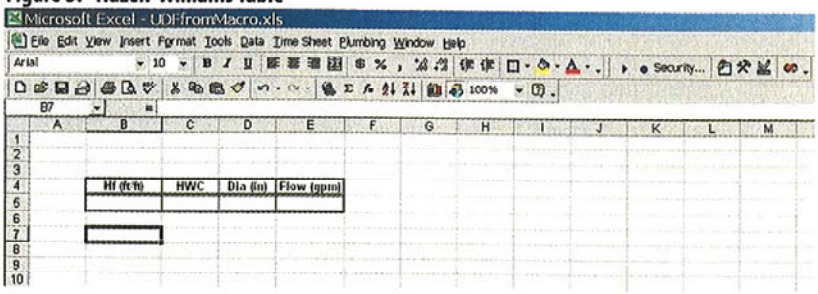


Figure 3. Hazen-Williams Table



view (on the Worksheet Menu Bar follow Tools→Options and check Formulas under View options). The dollar sign keeps the cell fixed in the same position when using the Copy and Paste functions.

Now you can enter other roof areas in column C and copy the formula in column B without needing to calculate one flow rate after another.

Using Names

Using too many cell addresses, especially for functions with more than three variables, may be confusing. The solution is to label cells with particular names. For example, using the previous example, select cell C4. In the name box to the left of the formula bar, change C4 to H and hit enter. Now cell C4 is named H and the formula becomes the following:

=ROUND(7.4805/60/12*H*C6,2) or
=ROUND(0.01039*H*C6,2)

This formula is more meaningful with H representing rain intensity, which is a constant in a given location. Do not replace cell C6 with a name because roof area is a variable.

Figure 2 shows a simple calculation sheet using the above formula. Name cell C4 as H and enter roof areas in cells C6 through C12. Enter the formula in cell B6, copy and paste it in cells B7 through B12, and the flow rate values appear instantly.

Using Macro Recorder

Excel's macro recorder can record the work done over a workbook or worksheet. Macros are actually procedures in Visual Basic automatically created by Excel.

Follow these steps to record a macro:

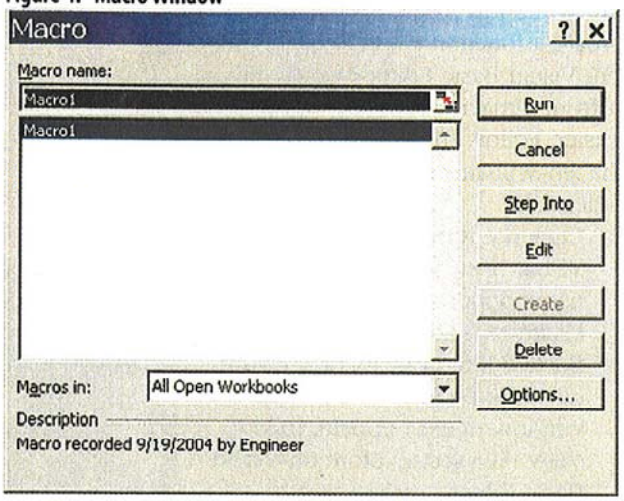
1. Create a new workbook and save it as UDFfromMacro.xls.
2. Create a table on sheet 1 as shown in Figure 3.
3. Bring the Visual Basic toolbar to view by following View→Toolbar→Visual Basic on the Worksheet Menu Bar. The solid circle is the Record Macro tool and the triangle is the Run Macro tool.
4. Click the solid circle and the Record Macro window pops up. You may change the macro name, storage place, and description if you like. In this example we leave the name as Macro1. Click OK to start recording Macro1. The solid circle in the Visual Basic toolbar is now a solid square, or Stop Recording.
5. Name cell B5 as S and enter the following formula:

   =10.4315*(Q/Ch)^1.85/D^4.8655
6. The formula appears in the formula bar. #DIV/0! appears in cell B5 but do not be concerned with that for now. Name cell C5 as Ch, cell D5 as D, and cell E5 as Q. (Ch stands for the Hazen-Williams Coefficient because C already is preserved by Excel.)
7. Click the solid square icon to stop recording. The icon changes back to a solid circle. Clear the contents in cells B5 through E5 and save and close the workbook.

Once you have recorded a macro, you can use it again and again. Simply open the workbook UDFfromMacro.xls and click the Run Macro icon (the solid triangle) on the Visual Basic toolbar. In the Macro window that appears, select Macro1 and click the Run button (see Figure 4).

The Hazen-Williams Formula should be in cell B5. Type 100, 4, and 100 in cells C5, D5, and E5 respectively.

Figure 4. Macro Window



Cell B5 now should show the value 0.012275076 (see Figure 5). This is the friction loss in feet/feet when Ch = 100, D = 4, and Q = 100. You can type any other reasonable numbers in cells C5 through E5, and a new friction loss will appear in cell B5. If for some reason the formula in cell B5 is erased, run Macro1 again and follow the above-mentioned steps to restart the calculation.

Figure 5. Calculated Hazen-Williams Formula

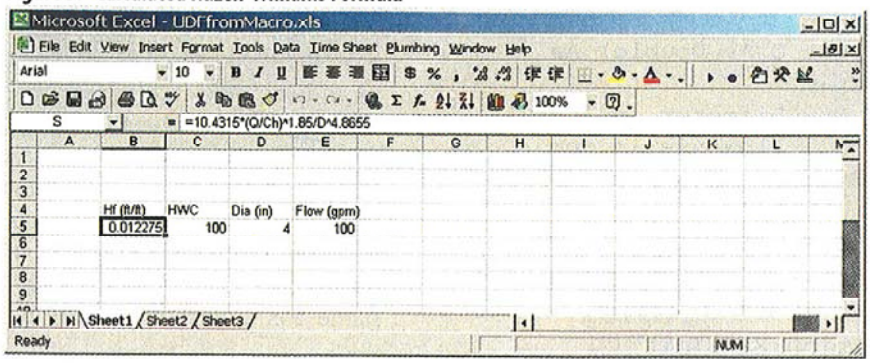
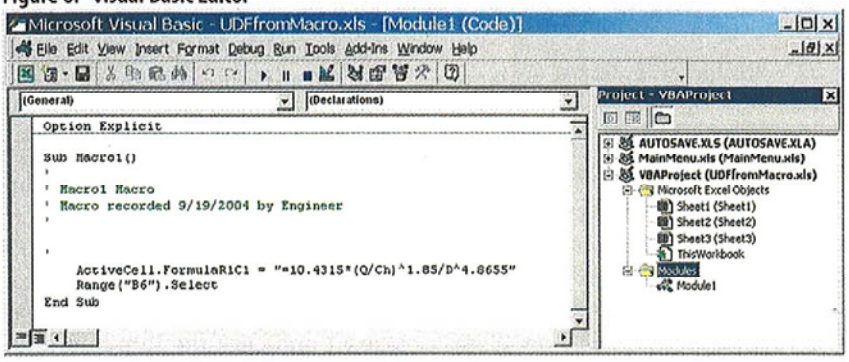


Figure 6. Visual Basic Editor



Creating UDFs from Macros

In addition to the pre-loaded functions, users can create their own functions. Both macros and functions are written in Visual Basic. You can create a function using Excel's inherent Visual Basic Editor or converting it from a macro. The latter method is easier. Follow these steps to convert the Hazen-Williams Formula function from Macro1:

1. Click the Run Macro tool on the Visual Basic toolbar. In the Macro name window, select Macro1 and click the Edit button. The Visual Basic Editor appears (see Figure 6). If the Project – VBAProject window doesn't appear, follow View→Project Explore on Visual Basic Editor's main menu.
2. Right click on VBAProject (UDFfromMacro.xls). A pop-up menu appears. Move the cursor to Insert and select Module. A new module, Module2, is added under Module1.
3. Copy all the contents in Macro1 and paste them in Module2. If two Option Explicit lines appear, delete one of them. Revise the pasted contents to read as follows:

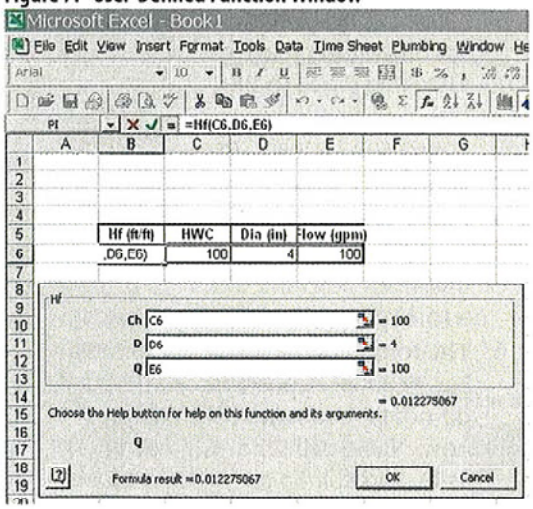
```
Option Explicit
Function Hf(ByVal Ch As
Double, ByVal D As Double, _
ByVal Q As Double) As
Double
' D - Diameter in inch
' Q - Flowrate in gpm
Hf=10.4315*(Q/Ch)^1.85/
D^4.8655
End Function
```

4. If you use the metric system, revise the function as follows:

```
Option Explicit
Function Hf(ByVal Ch As
Double, ByVal D As Double, _
ByVal Q As Double) As
Double
' D - Diameter in mm
' Q - Flowrate in L/s
Hf=1.18478*10^10*(Q/Ch)^1.85/
D^4.8655
End Function
```

Save the UDFfromMacro.xls file. You can rename Module2 by selecting the Properties Window from View on Visual Basic Editor's main menu.

Figure 7. User-Defined Function Window



You cannot use the UDF you just created quite yet. Follow these two steps to make the UDF available for use in other workbooks.

1. Save the workbook UDFfromMacro as an Excel Add-In file. Select Save As from the Worksheet Menu Bar. Select Microsoft Excel Add-In (*.xla) from the Save as type: drop-down box at the bottom of the Save As window. The file name UDFfromMacro.xla should be in the File name box. Choose the folder in which you want to save the UDF and click Save.
2. Add UDFfromMacro.xla to the Add-Ins list. From the Worksheet Menu Bar follow Tools→Add-Ins. The Add-Ins window appears. Click the Browse button to find the file you just saved and click OK. If you check the Add-Ins window again you see that a new item, UDFfromMacro, is added to the list and that the box to its left is checked. Now the Hazen-Williams Formula is loaded in Excel, and you can use it in any worksheet.

Putting UDFs to Work

In a new Excel workbook, bring the Paste Function on Insert Function window on screen as previously described. Click User Defined in the Function category box. In the Function name box select Hf, which is the function we just created, and click OK. The UDF appears as shown in Figure 7 when using Excel 2000 or earlier version. Version 2002 and 2003 redesigned the form and added a title for it as Function Arguments.

Link the argument boxes to the related cells (Ch→C6, D→D6, Q→E6) and click OK. The friction loss (Hf) in feet/feet is shown in the selected cell. In this example it is 0.012275 when C6 = 100, D6 = 4, and E6 = 100. If you know the function's name, you can type it directly in a cell preceded by an equal sign. You also can type values for the arguments in the parentheses, such as =Hf(100,4,100).

Functions are very important because they are the fundamental components of computer-aided computations. Using these guidelines you can create more UDFs to aid in plumbing engineering calculations. ■

Anjian Lu is an engineer with Vanderweil – NCR in Alexandria, Va. For further information, contact him at **luaj@yahoo.com.**