Excel VBA Programming: Functions

(Dr. Tom Co, 9/14/2008)

Introductory items:

1. A **function** a group of statements (or code) that yields a value.
2. There are Excel-supplied functions, such as **log()**, **sin()**, **linest()**, **mmult()**, etc. Sometimes the functions have to be programmed by the user for specific needs, these are often referred by Excel as a **UDF** (User Defined Functions).

Guidelines for when to create a function:

1. The cell formulas are too complicated to be written in one line.
2. The required calculations require several iterations.
3. The created function will be needed in multiple applications.
4. Grouping the calculation with a name will improve understanding of the spreadsheet.
5. The calculations need to be secured with passwords.

Disadvantages

1. Complicated and require accuracy checks → needs multiple case studies.
2. Can be time-consuming to program → needs to balance with time-savings from multiple applications.
3. Requires the management of a VBA modules.

Standard Format of Function

Function *function_name* (*argument_1* ,*argument_2*, …)
⋮
statements and calculations
⋮
function_name = ⋯
End Function

Example 1. Create a function to calculate the Reynolds number given by

$$N_{Re} = \frac{Dv\rho}{\mu}$$

where D is the diameter of the pipe, v is the mean velocity, ρ is the density and μ is the viscosity of the fluid.

Step 1. In an Excel spreadsheet, click **[Alt-F11]** to invoke the VBA editor.

Step 2. Open a module (you may need to create one by using **[Insert]→[module]** menu item).

Step 3. Type the following function

```
Function Nre(diameter, velocity, density, viscosity)

    Nre = (diameter * velocity * density) / viscosity

End Function
```

Step 4. Click [Alt-F11] to go back to the Excel spreadsheet.

Step 5. Try out the function. Select a cell and type: **=Nre(0.5,11,62.4,7.6e-4)** .

Remarks:

1. Unfortunately, in Excel 2007, the tooltips for the functions do not show up. To remember the arguments, do the following:
   a. In the spreadsheet, click on the [**Insert Function]** button as shown in Figure 1.

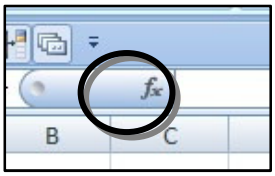

Figure 1. Insert function button.

   b. A window will open. Select **[User defined]** category, then scroll and select **Nre**, and click **[OK]** as shown in Figure 2.

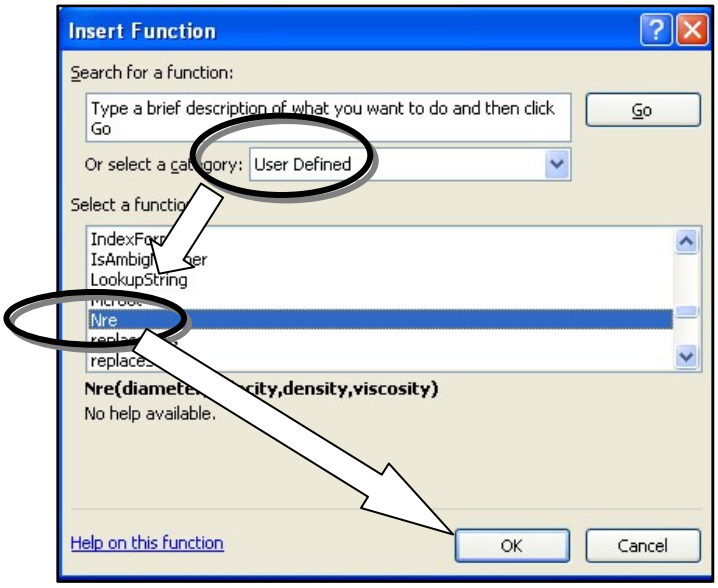

Figure 2. Insert function window.

c. Another window should now pop-up to help with entry of the arguments.

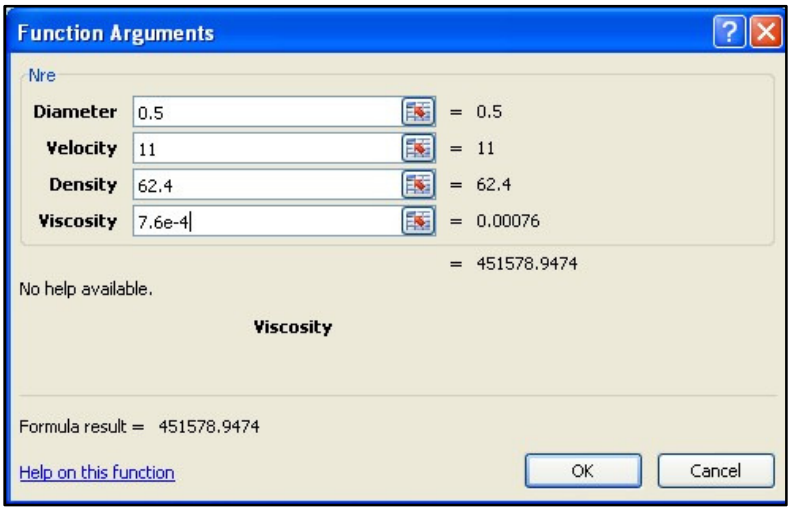

Figure 3. Window for argument input.

2. Tip: Instead of tooltip, one could select the cell, **[right-Click]** then select **[Insert comment]** and type the function name together with the arguments.
3. For longer statements, a continuation mark can be used by typing underscore (_)
4. Comments can be added by using a single quote in front of the line.

Decision statements: If... then...else...

```
If  (condition_1) then
     ...
Elseif (condition_2) then
     ...
Else
     ...
End If
```

Example 2. Create a function to calculate the friction factor f_F given by

$$f_F = \begin{cases} \dfrac{16}{N_{Re}} & \text{if } N_{Re} < 2100 \\ \dfrac{1}{16}\left(\log\left[\dfrac{\epsilon}{3.7}-\dfrac{5.02}{N_{Re}}\log\left(\dfrac{\epsilon}{3.7}+\dfrac{14.5}{N_{Re}}\right)\right]\right)^{-2} & \text{if } N_{Re} > 2100 \end{cases}$$

where ϵ/D is the relative roughness of the pipe and N_{Re} is the Reynold's number.

Then a function can be built for friction factor given by

```
Function FricFac(Nre, Roughness)

    If Nre < 2100 Then
        FricFac = 16 / Nre
    Else
        a = Roughness / 3.7
        b = 14.5 / Nre
        c = 5.02 / Nre
        k = Log(10)
        d = Log(a - c * Log(a + b) / k) / k
        FricFac = (1# / 16#) / (d ^ 2)
    End If

End Function
```

Remarks:

1. The symbol # signifies the number it is attached with is a floating point number instead of integer.

2. The **Log()** function in VBA is, unfortunately, a natural logarithm instead of the logarithm base 10.

Iteration Statements: Do... while

The formula for the pipe velocity based on energy balance is given by

$$v = \sqrt{\frac{g\Delta z + \frac{g_c \Delta P}{\rho}}{\frac{1}{2} - 2\frac{f_F L}{D}}}$$

Using successive substitution, we could use the following functions:

```
Function v_pipe(v_guess, dz, dP, L, D, rho, mu, epsilon)
    g = 32.174
    gc = 32.174
    Roughness = epsilon / D
    L_over_D = L / D
    numerator = g * dz + gc * dP / rho
    abserr = 10
    vnew = v_guess
    Do
        vold = vnew
        Re = Nre(D, vold, rho, mu)
        fF = FricFac(Re, Roughness)
        vnew = (numerator / (0.5 - 2 * L_over_D * fF)) ^ (1 / 2)
        abserr = Abs(vnew - vold)
    Loop While abserr > 0.00000001
    v_pipe = vnew
End Function
```

Remarks:

1. To save the group of functions as a module file, select **[File]→[Export File]** and save as ***.bas** file.
2. The function **v_pipe()** above does not check for convergence. It is strongly advisable to change the function to include a check on the number of iterations to make sure it does not go beyond a prescribed maximum number of iterations.