SOLVING NONLINEAR EQUATIONS USING *EXCEL*

Jay Villanueva
Mathematics Department
Florida Memorial University, Miami, FL 33054
jvillanu@fmuniv.edu
International Conference on Technology in Collegiate Mathematics, 2014

I.   Introduction
   A. Nonlinear equations
   B. The built-in features of *Excel*

II.   Examples of solutions of nonlinear equations by *Excel*
   A. Ship design
   B. Physics
   C. Open channel hydraulics
   D. General mathematics
   E. Calculus
   F. Calculus
   G. What can go wrong?

III.   Conclusions
   A. Ease of solving nonlinear equations using *Excel*
   B. Significance of this technique to other forms of equations

I, Introduction

As researchers and teachers of mathematics, we often find the need to solve equations, in particular, nonlinear equations. A *linear equation* in the *n* variables is one that can be expressed in the form:

where   and *b* are constants. The variables are usually called the *unknowns*. The following are linear equations:

We note that a linear equation does not involve any products or roots of variables. All variables occur only to the first power and do not appear as arguments for trigonometric, logarithmic, or exponential functions. The following are *nonlinear* equations:

The harder types of nonlinear equations are what we may call the *implicit* type, such as the two equations on the left in the last examples above. There are a number of traditional methods of solution, but often, these methods require some programming to implement them.

Excel has definite advantages over the traditional methods in that it is readily available, and it has built-in features – *Goal Seek* and *Solver* – which allow us to perform iterative solutions without programming. Both tools allow us to iteratively find a target value in a target cell by changing a value in some other related cell, and both are simple to use, although with certain key differences.

Let us take a specific example: To solve the implicit equation

for the frictional drag coefficient, as a function of the Reynolds number, for some ship resistance calculations. This equation cannot be written in an explicit form and we have to resort to some iterative method to find corresponding to some given value of So, first, we rewrite the equation in the form

.

Now, we can solve the equation using *Goal Seek* or *Solver* to find iteratively a value for given a value for So enter the equation as a spreadsheet formula, as in Figure 1.

To use *Goal Seek* select … from the main menu bar and enter the three parameters: Set cell (cell C5, which contains the formula above); To value (0, the target value for
the formula above); and By changing cell (cell C4, which contains the friction coefficient, , as in Figure 2. When we press the OK button, Goal Seek will vary the value in C4 until the formula in C5 is approximately 0. The result in this example is given in Figure 1, where the residual value in C5 is In this particular example, where the initial value given to was for we know that this should be a very small value, but we could not assign 0 initially because it would result to a divide-by-zero error, as is clear from the formula.
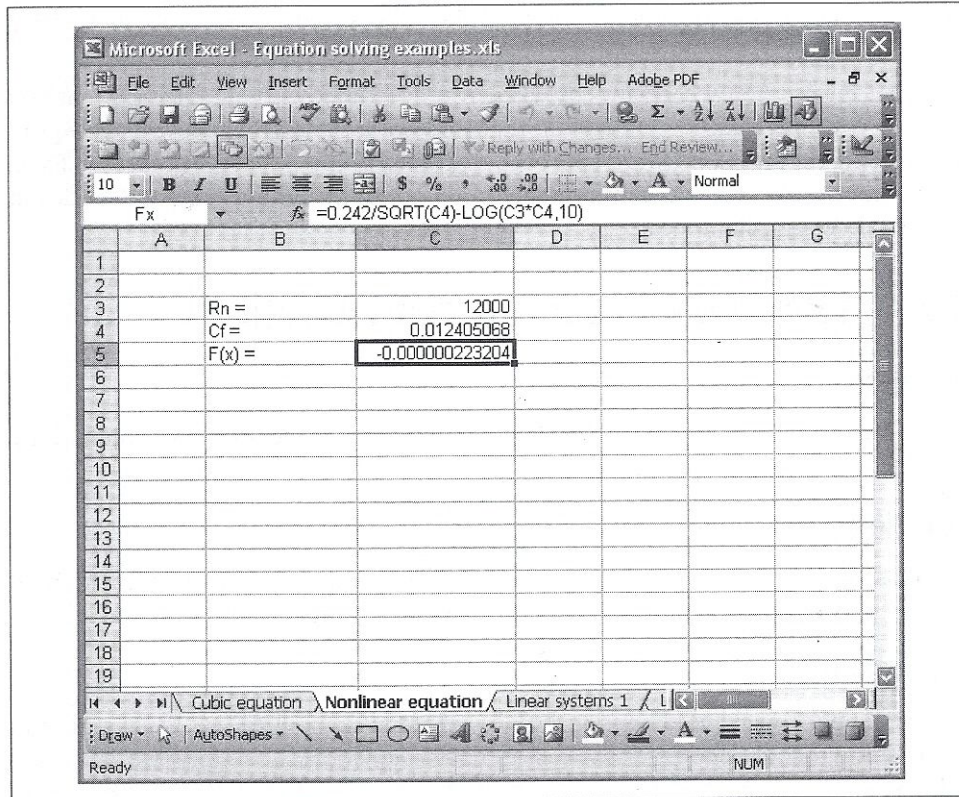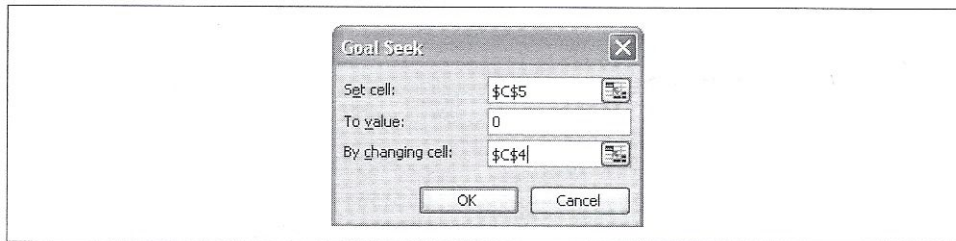
Figure 1. Nonlinear example

Figure 2. *Goal Seek* parameters

A residual value of is not bad. But if higher accuracy is required, we can go to the Options window and change two things: Maximum number of iterations (default, 100), and Maximum change between iterations (default, 0.001), see Figure 3.
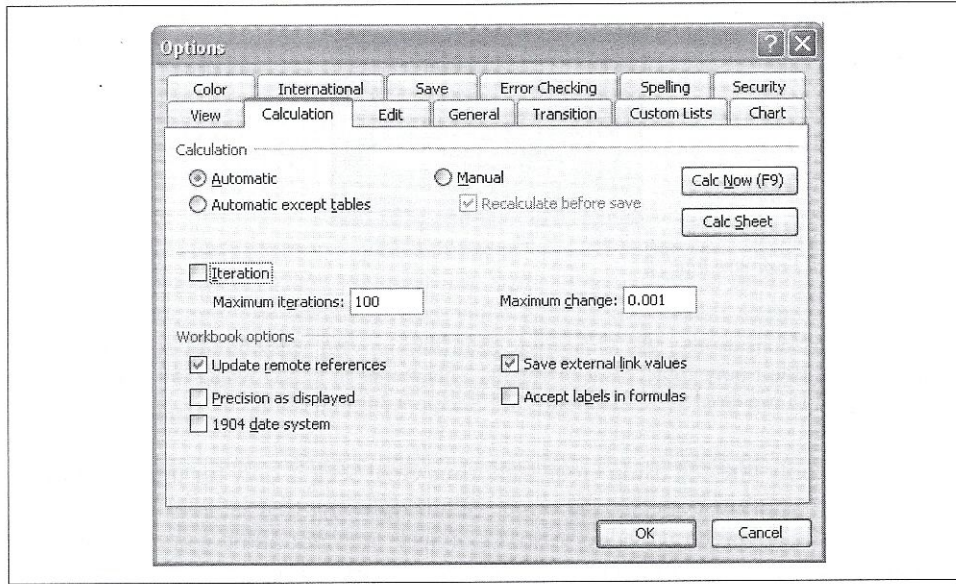
Figure 3. Options window

For simple iterative problems, *Goal Seek* is quite capable and it is quick and easy to use. For more complex problems, *Solver* is the tool of choice.

Solver is similar to *Goal Seek* in that you can iteratively find a target value in a target cell by changing values in an independent cell, Figure 4.
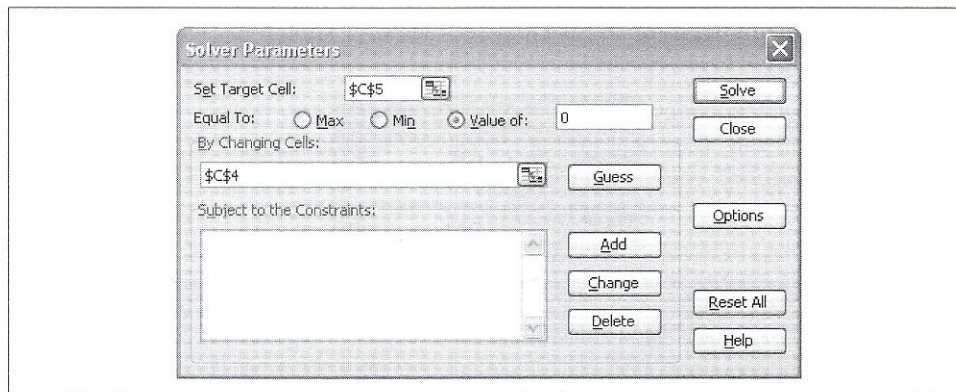

Figure 4. *Solver* parameters

But *Solver* features go well beyond those of *Goal Seek*. It can be used for multivariate problems, up to 200 independent variables; you don't have to specify a target value: you can choose to maximize or minimize the value instead; you can specify constraints on some of the variables, which is good for solving simultaneous equations, and linear or nonlinear programming. For precision, it has a default value of 1E-6 between iterations, up to 100 iterations by default, and with a maximum time of convergence of 32,676 sec (hr). *Solver* stops when one of these limits is reached, but all thresholds can be changed in the Options box, Figure 5.
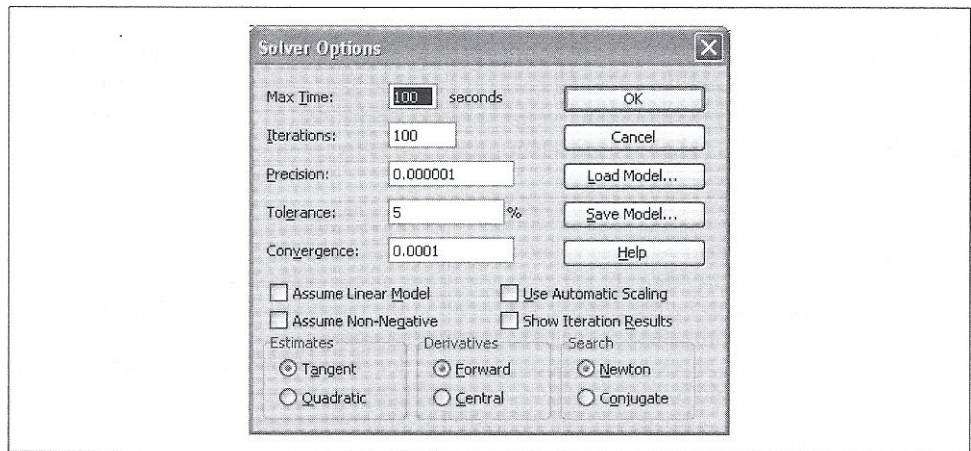
Figure 5. *Solver* options

II. Examples
A. Ship design [Bourg, 2006]

This is the same example given above [Bourg, 2006]. The drag coefficient, is a function of Reynolds number, For a given value of   we vary from an initial value until it converges to 0. We summarize the results (residual value,):

Goal Seek:
Solver:

B. Physics [Resnick et al, 2002]

The velocity *v* of a body under a damped force is given as a function of time, *t*, by:

We want to know for what *t* is $v = 0$, other than the value $t = 0$.

Goal Seek:
Solver:

C. Open channel hydraulics [Giles, 1962; Streeter, 1979]

Flow in an open channel is given by the Powell equation:

where C is the drag coefficient and   is the Reynolds number. We would like to know the value of C for

Goal Seek:
Solver:

D. General Math [Sullivan, 2012]

From the equation   , we would like to get *y* for some value of *x*, say *x* = 1.

Goal Seek:
*Solve*r:

Note that in this example, the equation is quadratic in *y*; thus, we can use the quadratic formula to verify our solution.

E. Calculus [Thomas, 1972]

From the equation   we would like to know y for some value of *x*, say *x* = 1.

Goal Seek:
*Solve*r:

F. Calculus [Ayres, 1964]

From the equation   we would like to find *y* for some value of *x*, say *x* = 1.

Goal Seek:
*Solve*r:

G. What can go wrong?

Although the procedure is straightforward, there are a couple of things to be noted. First, the choice for an initial value of the independent variable is important. For example, in Example A, we know that   is very small. If we set   then we will have an error of divide by zero, and the utility fails. So pick a small but nonzero value, say

Second, if there are several values of the independent variable for the same *y*, then we should specify different initial values appropriate for the problem. For example, in Example D, the equation is quadratic in *y*. Thus,

III. Conclusions

Solving nonlinear equations is by nature difficult because there are no closed-form algorithms one can apply in general. It is inherently numerical because it is iterative. The solution though can be made easy with the use of *Excel* because *Excel* has built-in tools (*Goal Seek* and *Solver*) which can do iterative calculations without the need for programming. The overriding advantage of *Excel* is its ready availability, allowing us to work practically everywhere. The fact that the solution of a difficult problem, like nonlinear equations, can be effected readily adds confidence to our ability to solve any type of equations, not just polynomial equations.

References

Ayres, E, 1964. *Calculus* (Schaum Outline Series). NY: McGraw-Hill.

Bourg,D, 2006. *Excel Scientific & Engineering Cookbook*. CA: OReilly Books.

Giles, R, 1962. *Fluid Mechanics & Hydraulics* (Schaum Outline Series). NY: McGraw-Hill.

Resnick, R, Halliday, D, & Krane, K, 2002. *Physics*. NY: J Wiley.

Streeter, V, 1979. *Fluid Mechanics*. NY: McGraw-Hill.

Sullivan, M, 2012. *Algebra & Trigonometry*. MA: Prentice Hall.

Thomas, G, 1972. *Calculus*. MA: Addison-Wesley