IT Services

University of Reading

Microsoft Excel 2007™ Goal Seek and Solver (Level 3)

Contents

Introduction

Microsoft Excel can be used to solve mathematical equations and optimization problems (ie where a value is calculated from more than one independent cell). Simple problems can be solved using *Goal Seek*; more complex ones by using *Solver*.

Begin by typing in some data:

1. Start up Excel as usual (or move to an empty worksheet)
2. In cell *A1* type `x` press the **<*right arrow*>** key then (in cell *B1*) type `y`
3. Move to cell *A2* and type `-5` then press the **<*down arrow*>** key
4. In cell *A3* type `-4` then select *A2* and *A3* and drag the *range handle* (the small black square at the bottom right of *A3*) down to cell *A12* - you should now have values from -5 to 5
5. Move to cell *B2*, type the formula `=A2*A2-2*A2-3` (press **<Ctrl Enter>** to stay in *B2*) then *double click* on the *cell handle* to fill down the values - you should have values from 32 to 12

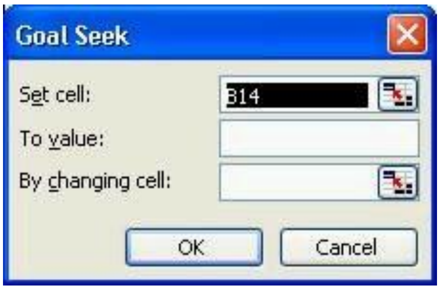
To see exactly what's happening, plot the data on a graph:

6. Press **<Ctrl a>** to select all the data (or drag through cells *A1* to *B12*)
7. Now click on the **Insert** tab, then click on **[Scatter]** in the *Charts* group and choose the second subtype (**Scatter with Smooth Lines and Markers**)
8. Select the subtype **Scatter with data points connected by smoothed Lines** then click on **[Finish]** - your chart will appear on the current worksheet

Goal Seek

It's clear both from the graph and from the data values that the roots of (ie solutions to) the equation $y = x^2 - 2x - 3$ are $x = 3$ and $x = -1$ (this is when the value of y is 0). You could have used the *Goal Seek* command to find these values without having calculated any values or plotting the graph:

1. Move to cell *B12*, **[Copy]** the formula, then move to *B14* and press **<Enter>** for paste
2. Next, on the **Data** tab, click on **[What-If Anaylsis]** then choose **Goal Seek...** - a dialog box appears:



3. Press **<Tab>** then in *To value:* type `0` - press **<Tab>** again
4. In *By changing cell:* type **`A14`** then press **<Enter>** for **[OK]**

You should find that a value now appears in cell *A14* which gives one of the solutions to the equation. The result is shown as *-0.99992*, whereas the actual value is *-1.0*. *Goal Seek* stops when it finds a solution within its pre-set limits. These limits are set in *Excel Options*. To get greater accuracy:

5. First, press **<Enter>** for **[OK]** to close the *Goal Seek Status* window
6. Click on the **[Office Button]** then on **[Excel Options]**
7. Move to the **Formulas** options and, in the top *Calculation options*, change **Maximum Change** to 0.00001 (or the accuracy you require) – sometimes you may also need to change *Maximum Iterations*
8. Press **<Enter>** for **[OK]** then repeat steps **2** to **5** – you should now have a more accurate solution

The algorithm usually finds the nearest root to the starting value (which was 0). To find the other solution:

9. Change *A14* to `5` (press ***<right arrow>*** to move to cell *B14*) then repeat steps **2** to **5**

You should now get the other root, *x=3*. *Goal Seek* can be used to find any target value. If you want to find the value of *x* for a given value of *y* then you just type that into the *To value:* box:

10. Repeat steps **2** to **5** but set *To value:* as `1.8`

Again this is one of two solutions. Use different starting values in *A14* to find the other.

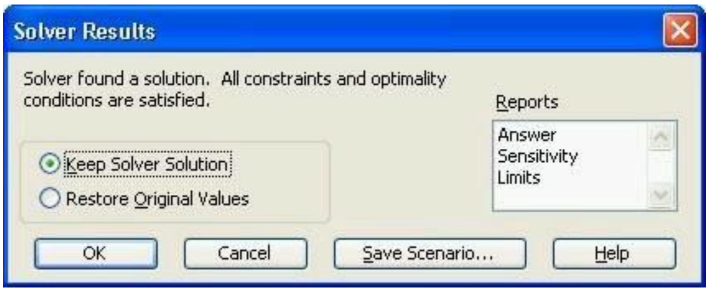
Solver

A more powerful version of *Goal Seek* is provided by *Solver*. Whereas *Goal Seek* has a single target cell and changing cell, *Solver* can work out solutions to problems involving several changing cells (ie cells storing or calculating independent values). In it, you can also set up constraint criteria (eg that cells must hold positive values or be whole numbers). To find the roots of the equation:

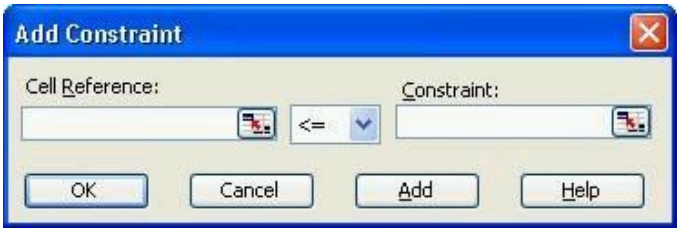
1. On the **Data** tab, click on **[Solver]** in the *Analysis* group on the right - the following dialog box appears:

Note: If this button is missing from your *Ribbon*, you will need to load the **Solver Add-in** from the **Add-In** tools in **[Excel Options]** via the **[OfficeButton]**.

2. Check the *Target Cell:* is **B14** then set *Equal To:* to **Value of:** and leave the value as **0**
3. Set *By Changing Cells:* to `A14` then press **<Enter>** or click on **[Solve]** - one root is found



4. Press **<Enter>** for **[OK]** to keep the Solver solution then click on **[Solver]** again (note that the previous settings are held, unlike *Goal Seek*)
5. Next, click on the **[Add]** button under the *Subject to the Constraints:* heading:



6. Set *Cell Reference:* to `A14` and *Constraint:* to `0`, leaving the relationship as <=
7. Press **<Enter>** for **[OK]** to add the constraint to the *Solver Parameters* (the **[Add]** button is used to add further constraints)
8. Finally press **<Enter>** for **[Solve]** - *Solver* finds the negative root of the equation (press **<Enter>** for **[OK]** to accept the solution)

As well as being able to find a given solution to a formula, *Solver* can find the *Maximum* or *Minimum* value. This is not only of use in mathematics but is widely used in commerce to find maximum profit or minimum costs. Try finding the minimum value of the equation (there isn't a sensible maximum - the values rise to infinity).

First, remove the constraint in case the solution is positive

9. On the **Data** tab, click on **[Solver]**
10. Select *A14 <= 0* in the *Subject to the Constraints:* box then click on **[Delete]**
11. Set *Equal To:* to **Min** then press **<Enter>** for **[Solve]** - the minimum found is *-4* at *x=1* (press **<Enter>** for **[OK]** to accept the solution)

Solving Two Equations

Solver can also be used to find where two graphs cross. Begin by entering a new formula in column *C* then plot the new data on the chart:

1. Move to cell *C1* and type `z` - press **<Enter>**
2. In cell *C2* type `=8+2.5*A2-0.5*A2*A2` then press **<Ctrl Enter>** to stay in cell *C2*
3. Next, *double click* on the *cell handle* to fill the column - you should get values from -17 to 8
4. Select the range *C1:C12* and **[Copy]** the values
5. Now click on the graph and press **<Enter>** to paste in the new series

6. Finally, move to cell *C12*, **[Copy]** the formula then move down to *C14* and press **<Enter>** to paste it

To find where the two curves cross you need to set up a constraint that the *y* and *z* values should be equal:

7. On the **Data** tab, click on **[Solver]** - check that *Equal To:* is set to **Min** or **Max**
8. Click on the **[Add]** button and set a constraint of `B14 = C14` - press **<Enter>** for **[OK]**
9. Press **<Enter>** for **[Solve]** then again for **[OK]** to close the *Solver Results* window

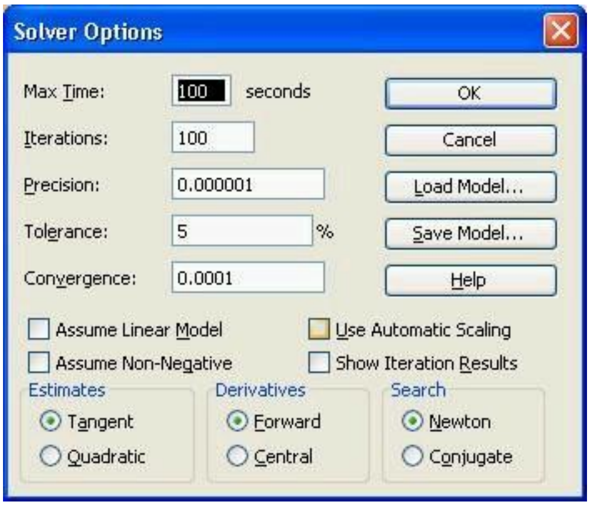
The same solution is found whether you choose *Max* or *Min*, even though there are two values of *A14* (ie *x*) which satisfy the *Solver Parameters*. Sometimes, *Solver* fails to find a solution even when there should be one. To find the alternative one here limit *x* to positive values:

10. Repeat steps **6** to **8** but at step **7** set a second constraint of `A14 >= 0`
11. Change the starting value in *A14* to **10** and run *Solver* again

This time, the second solution is found. Whether this happens or not depends on the values the algorithm chooses to reach the solution. If you are certain that a solution does exist, then try different starting values, as above, to see whether it makes any difference.

You may be able to improve the efficiency of the *Solver* by altering the *Options*. To see these:

12. On the **Data** tab, click on **[Solver]** then click on the **[Options]** button - the following window appears:



Many of the options are for the real experts, who understand precisely how the *Solver* works. Sadly, quick help on what each of these do is not available here, but you can find full details in the main Help system - search for `solver options`. One option which may be of interest is *Show Iteration Results*. This lets you step through the iterations when you run *Solver* so that you can see whether the algorithm gets stuck (and fails) or whether it genuinely can't find a solution after 100 iterations.

13. Don't change any of the options here – press **<Esc>** or click on **[Cancel]**

Further Examples

The examples so far have been very scientific and based on a single cell (*A14*). In this next section, non-scientific problems using multiple cells are tackled.

Example 1 - Buying Chocolate

You have a £2.00 voucher to spend on chocolate. A small (100g) bar costs 17p, a large (250g) one is 32p. However, there's a special deal on the small bars whereby if you buy 3 you get one free. You want to buy as much chocolate as possible - ideally, you want to spend every penny as the shopkeeper doesn't give change for a voucher. How many of each sort should you buy?

1. Move to a new sheet and in cell *A2* type **`Large`** (press **<Enter>**)
2. In cell *A3* type **`Small`** (press **<Enter>**) and in *A4* type **`Offer`** (press **<Enter>**)
3. In cell *A5* type **`Total Small`** then move to cell *B1* and type **`Number`** (press ***<right arrow>***)
4. In cell *C1* type **`Weight`** (press ***<right arrow>***) then in *D1* type **`Cost`** (press **<Enter>**)
5. Move to cell *B2* and type **`1`** (press ***<right arrow>***)
6. In cell *C2* type **`=B2*250`** (press ***<right arrow>***) then in *D2* type **`=B2*32`** (press **<Enter>**)
7. Into *B3* to *D3* type **`1`** , **`=B3*100`** and **`=B3*17`** , respectively
8. Into *B4* to *D4* type **`1`** , **`=B4*400`** and **`=B4*51`** , respectively
9. In *B5* type **`=B3+4*B4`** (press ***<right arrow>***)
10. In *C5* click on the **[Sum]** button on the **Home** tab then press **<Enter>** to accept the formula
11. Repeat step **10** in cell *D5*
12. Finally, in cell *A7* type **`Budget`** and in *B7* the value **`200`** (press **<Enter>**)

You now have the basic data required for *Solver*.

13. Move to the **Data** tab and click on **[Solver]**
14. Set the *Target Cell:* to **`C5`**, *Equal To:* to **`Max`** and *By Changing Cells:* to **`B2:B4`**
15. Next, click on the **[Add]** button to add the following constraints (note that you can use ranges to define them):
    `D5 = B7` (total cost must equal the budget - click on **[Add]**)
    `B2:B4 >= 0` (you can't have negative numbers - click on **[Add]**)
    `B2:B4 int` (you can't cut a bar up into pieces - press **<Enter>** for **[OK]**)
16. Finally, press **<Enter>** again for **[Solve]** then again to *Keep Solver Solution*

A solution is found (buy 2 large bars and 10 small ones (8 plus 2 free) giving 1500g in total) but is that the best one? Examine the constraints to check they are correct. You must have whole numbers of each bar and you can't have a negative number but do you have to spend all the money? What happens if you relax this slightly:

17. On the **Data** tab, click on **[Solver]**
18. Select the condition *D5=B7* then click on **[Change]**
19. Edit the condition so that it reads **D5 <= B7** then press **<Enter>** for **[OK]**
20. Finally, press **<Enter>** again for **[Solve]** then again to *Keep Solver Solution*

The solution found (3 large and 8 small bars) gives you an extra 50g for only £1.98. You should opt for this solution and tell the shopkeeper to keep the change!

Example 2 - Supplying Shops

A wholesaler has 4 depots supplying shops around the country. It receives orders from stores in Brighton, Bristol, Cardiff, Chester, Leeds and Reading respectively for the following quantities of its products: 100, 200, 150, 100, 180 and 120. Stocks in the depots (in Coventry, Liverpool, London and Southampton) stand at 180, 200, 300 and 220. Which depot should supply how much to which stores given the relative costs (transport etc) shown in the table below?

	Brighton	**Bristol**	**Cardiff**	**Chester**	**Leeds**	**Reading**
Coventry	12	7	8	7	10	9
Liverpool	15	10	11	4	8	13
London	5	9	11	13	14	6
So'ton	4	7	8	14	15	6

1. Starting on a new worksheet, type in the above table (using cells *A1* to *G5*)
2. Move to cell *I1* and type `Stocks` (press **<Enter>**)
3. Fill cells *I2* to *I5* with the current stock levels - `180` , `200` , `300` and `220`
4. Move to cell *A7* and type `Orders` then in cells *B7* to *G7* type the order numbers - `100` , `200` , `150` , `100` , `180` and `120`
5. Select cells *A2* to *A5* then **[Copy]** them - move down to cell *A9* and press **<Enter>** for paste
6. Move to cell *I9* and type the formula `=SUM(B9:G9)` (press **<Ctrl Enter>**)
7. Using the cell handle, copy this formula down to cell *I12*
8. Move to cell *A14* and type `Totals` (press **<*right arrow*>**)
9. In cell *B14* type the formula `=SUM(B9:B12)` (press **<Ctrl Enter>**)
10. Using the cell handle, copy this formula across to cell *G14*
11. Move to cell *A16* and type `Costs` (press **<*right arrow*>**)
12. In cell *B16* type the formula `=B2*B9+B3*B10+B4*B11+B5*B12` (press **<Ctrl Enter>**)
13. Copy this formula across to cell *G16* using the cell handle
14. Finally, move to *I16* and type `=SUM(B16:G16)` (press **<Enter>**)

All should now be set up for *Solver*. You need to minimize the total costs (stored in *I16*) by changing the values in *B9* to *G12*, subject to various constraints.

15. On the **Data** tab, click on **[Solver]**
16. Set the *Target Cell:* as `I16` then *Equal To:* to `Min` and *By Changing Cells:* to `B9:G12`
17. Now click on **[Add]** to put in the following constraints:
    `B9:G12 >= 0` (you can't have negative orders - click on **[Add]**)
    `B9:G12 int` (the orders must be whole numbers - click on **[Add]**)
    `I9:I12 <= I2:I5` (the orders must be within stock limits - click on **[Add]**)
    `B7:G7 = B14:G14` (the totals must equal the orders - press **<Enter>** for **[OK]**)
18. Press **<Enter>** for **[Solve]** then again to *Keep Solver Solution*

Cells *B9* to *G12* should now show the amounts that should be shipped from each depot to minimize overall cost.

Saving Solutions

When you save a file in which you have been running *Solver*, the current *Solver Parameters* are stored. Sometimes, however, you may want to save more than one set of parameters. You do this by saving each individually as a *model*.

1. On the **Data** tab, click on **[Solver]**
2. Click on **[Options]** then on **[Save Model...]** - the following screen appears:

You are now asked to select an empty range of cells where *Solver* can store its parameters (the target and changing cells plus any constraints). You only need select a single cell to define the start of the range.

3. Type `K9` as the starting cell for the range then press **<Enter>** for **[OK]**
4. Press **<Enter>** for **[OK]** again to close the *Solver Options* dialog box - note how cells *K9* to *K15* are used to store the parameter values and formulae
5. Now click on **[Add]** to change the current constraints
6. Type in an extra constraint of `B9:G12 <= 100` (the maximum load a lorry can transport is 100 units) then press **<Enter>** for **[OK]**
7. Press **<Enter>** again for **[Solve]** then again to *Keep Solver Solution*
8. Repeat steps **1** to **4**, saving this second model starting in cell `L9`

To rerun the original model:

9. In the *Solver Options* window, click on **[Load Model...]** - a dialog box similar to when saving appears

IMPORTANT: When loading a model, you must specify the range to include *ALL* the cells used to store the parameter values:

10. Type the range as `K9:K15` then press **<Enter>** for **[OK]**

You are now asked whether you want to reset the previous parameters:

11. Press **<Enter>** for **[OK]** then again to close *Solver Options* - you'll find the extra condition typed in at step **6** has been removed
12. Press **<Enter>** for **[Solve]** then again to *Keep Solver Solution*

For practice, load in the other model (ie start at step **9**), specifying the load range as `L9:L16`.

Note: To clear all the current *Solver* parameters, click on **[Reset All]** in the *Solver Parameters* window.

13. **[Close]** your file – there's no need to save it (unless you want to)