Excel for Algebra [1]

Lesson 4: More About Goal Seek

Don't we know about Goal Seek already?

Yes, we do. We've used Goal Seek in earlier lessons, to find values for checking equations, and to find values that solve consumer loan problems. This lesson just focuses a bit more closely on Goal Seek in particular.

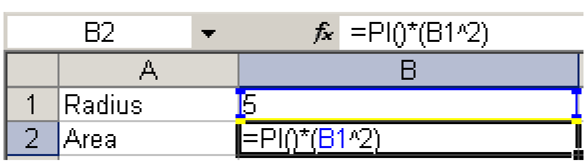
What does "Goal Seek" mean?

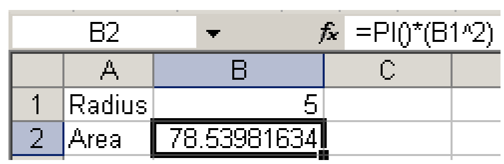
Imagine that you have a spreadsheet set up with a formula, or a chain of formulas, that computes a single result value from a single input value. Now suppose you want to run the calculation in reverse: you know the desired result, and you want to know the input value that will produce it. "Goal Seek" is a command in Excel that will automatically solve this problem.

Can you give me an example?

Sure. Suppose you have a formula for the area of a circle: $A = \pi r^2$ (area = pi times the radius squared). In Excel, we might have a spreadsheet that looks like this:

(Shown as formulas, using ctrl-` , then as values)





Notice that we have a formula in cell B2 that computes the area based on the radius that is stored in cell B1. (Excel's built-in function PI() gives the value of π to as much accuracy as Excel can handle.)

We plug the value 5 into cell B1 (radius), and cell B2 (area) automatically gets recalculated as 78.53981634.

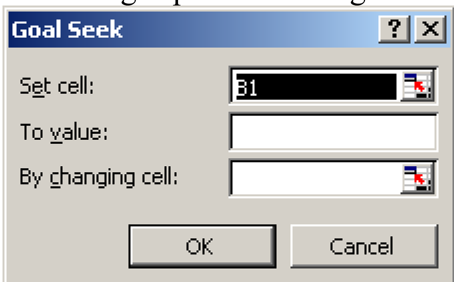
But what if we want to know the radius that produces an area of 100?

One approach is to attack with algebra, but we haven't gotten there yet in class.

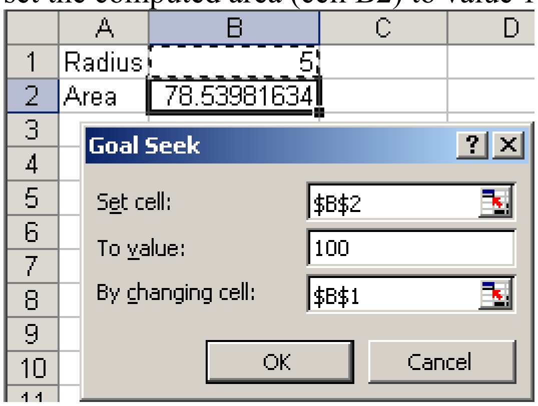
Another one is to let Goal Seek compute the number for us. Let's do Goal Seek.

In Excel's menu system, click on Tools > Goal Seek... (or the Excel 2007 equivalent, Data > What-If Analysis > Goal Seek...).

That brings up a little dialog like this:



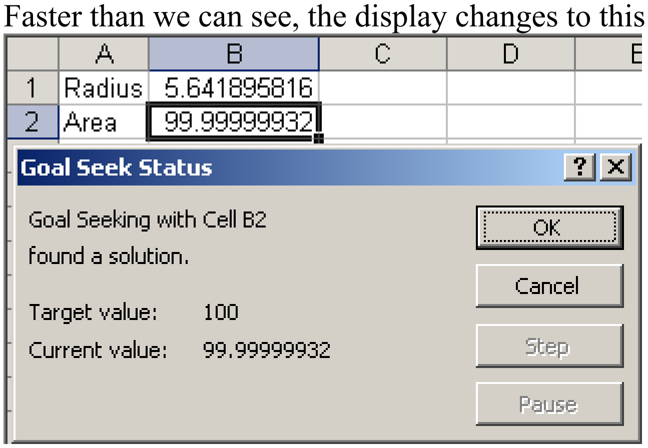
We specify the three requested pieces of information, so that we're telling Goal Seek to set the computed area (cell B2) to value 100, by changing the radius (cell B1).



(Remember that the dollar signs just mean it's an absolute cell reference. B2 is the same as B2. You can type in B2 if you like. I just clicked on the cell that I wanted, and the dialog automatically filled in B2 for me.)

Once the two cells and the value have been specified, we click OK.

Faster than we can see, the display changes to this:



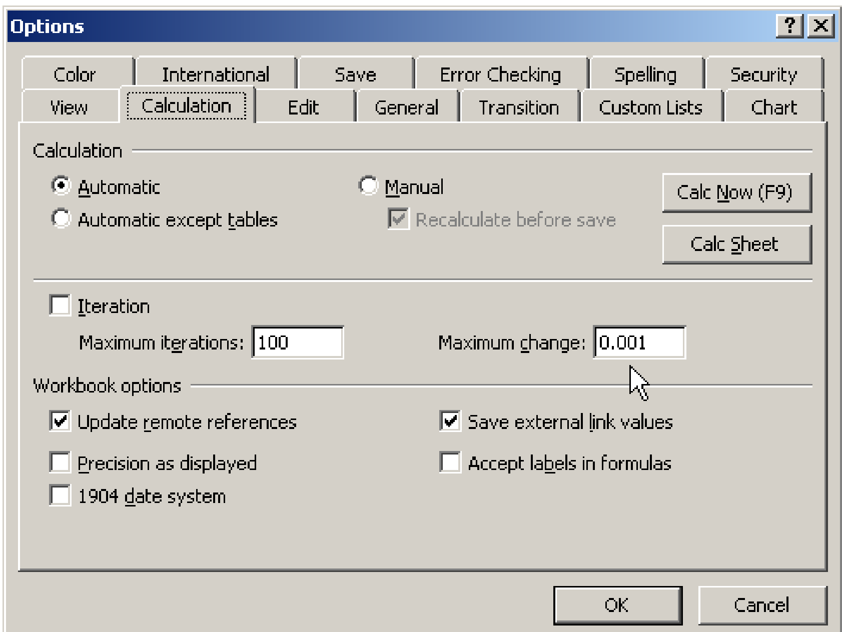
So just like that, Goal Seek has automatically determined (given the formula we specified), that it takes a radius of 5.641895816 to produce a circle with area 99.99999932 — equal to 100 for most practical purposes.

That's pretty accurate! But can I get the answer even closer?

Yes, you can. Goal Seek always gives "approximate" answers. Sometimes its answer happens to be exact, but really that's just good luck. However, there is a way to tell it to work harder and make the answer more accurate.

That's done on the "Calculations" tab of the "Options" dialog. In Excel 2003, you get that dialog through the menu system by clicking on Tools > Options..., then clicking on the Calculations tab.

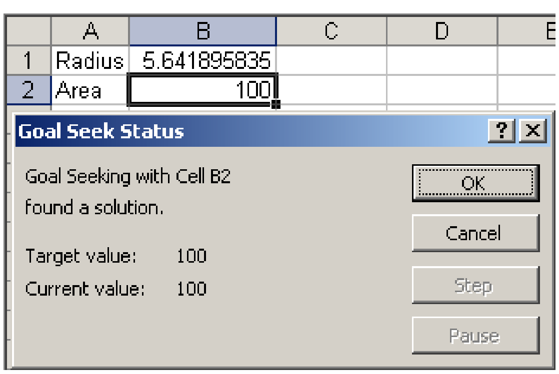
Notice the value labeled "Maximum change". The default value is 0.001, which lets Goal Seek be kind of sloppy. To make Goal Seek be a lot more precise, just insert some more zeros to make the number be a lot smaller, something like 0.00000000001 . (You won't be able to see the entire number at once, but that's OK.)



becomes 

.

Now when you run Goal Seek as before (starting from Radius = 5), it gives this:



That's pretty cool! But what if my problem is more complicated?

Goal Seek is a very simple and robust tool that can solve pretty much any formula or chain of formulas in which there's one input and one output. [2]

With some complicated formulas, you might have to provide some reasonable guess as a place for Goal Seek to start. Normally, you will have tested your formulas by plugging in some input values and checking to be sure that the corresponding outputs were correct. Whichever of your test inputs gives an output closest to the desired value is a good guess for Goal Seek to start from.

Goal Seek has no trouble handling multiple formulas, as long as the formulas all work together to produce one output from one input.

Sometimes you have to do a little rearranging of your problem so that Goal Seek can work on it.

The trick is always to reduce the problem to one input, which produces one output, whose desired value you know.

Often the "desired value" will be an "error value" that tells Goal Seek something about how badly wrong the current input value is. Just set up a formula to compute the error, such that error = 0 indicates that the solution has been found.

For example, suppose you're working one of the mixture problems that we've been working on recently.

> *Joe Grocer wants to mix 50 pounds of nuts that cost $2 per pound with some other nuts that cost $6 per pound, to make a mixture that costs $5 per pound. How many pounds of $6 nuts should he use?*

You have already thought about this problem well enough to attack it by constructing a table, something like this:

Item	**Number of pounds**	**Dollars per pound**	**Dollars**
Type A nuts	50	2	50*2
Type B nuts	x	6	$6x$
mixture	$50 + x$	5	$50*2 + 6x$ (adding down) $(50 + x)*5$ (multiplying across)

To finish solving the problem algebraically, we would write

$$50*2 + 6x = (50 + x)*5$$

and solve that for x. The essence of this approach is to recognize that we have two ways to compute the total dollars for the mixture (adding down, or multiplying across), and those two ways have to produce the same number.

---

[2] The word "robust", as used here, means that Goal Seek is able to handle a wide range of problems without giving strange results and without requiring you to fiddle with how you set up the problem.
The word "solve", as used here, means to find a <u>number</u> that solves a problem. Excel does not know how to do symbolic algebra. Excel will never say "gee, if $A=\pi r^2$, then $r=\text{sqrt}(A/\pi)$". But it can certainly find r such that $100 = \pi r^2$.

To attack with Goal Seek, let's expand the table just a bit to add an "error" value that will just be the difference between the two ways of calculating total dollars for the mixture.

Item	**Number of pounds**	**Dollars per pound**	**Dollars**
Type A nuts	50	2	50*2
Type B nuts	x	6	$6x$
mixture			$50*2 + 6x$ (adding down)
mixture	$50 + x$	5	$(50 + x)*5$ (multiplying across)
			Error = the difference between "adding down" and "multiplying across". If this equals zero, the problem is solved!

Thinking about this algebraically, all that we have done is to turn

$$50*2 + 6x \;=\; (50 + x)*5$$

into $$50*2 + 6x \;-\; (50 + x)*5 \;=\; 0$$

But this is just what we need to make Goal Seek work. Its job will be to make the error value zero, by changing x.

Now, if we construct a spreadsheet to represent this problem, it might look like the picture below (formulas view). In the spreadsheet, there are 5 numbers: the pounds for each of "Type A" and "Type B" nuts, the dollars per pound of those types of nuts, and the dollars per pound of the mixture. The rest of the spreadsheet consists of simple formulas to calculate total pounds of mixture and to calculate the total dollars for each of the ingredients and for the mixture. The total dollars for the mixture is computed two different ways — once by adding the cost of ingredients, and once by multiplying the mixture's weight by the intended cost. Once the problem has been solved, those two ways will give the same number.

<u>Notice that in this spreadsheet, there is no "*x*".</u> That's because, in the spreadsheet, nothing is "unknown". As far as the spreadsheet is concerned, the number of pounds of Type B nuts is just a number, not an unknown. What I've done here is to simply plug in an arbitrary number 3 for the number of pounds of Type B nuts.

	A	B	C	D	E
1	Item	Number of pounds	Dollars per pound	Dollars	
2	Type A nuts	50	2	=B2*C2	
3	Type B nuts	3	6	=B3*C3	
4	mixture			=D3+D2	adding down (to get total $$ for mixture by adding $$ for ingredients)
5	mixture	=B2+B3	5	=B5*C5	multiplying across (to get total $$ for mixture by multiplying total weight of the mixture, times the mixture's cost per pound)
6				=D4-D5	"Error" --- the difference between "adding down" and "multiplying across"

Here is the normal (values) view, with auditing arrows added so that we can see how the numbers flow between formulas.

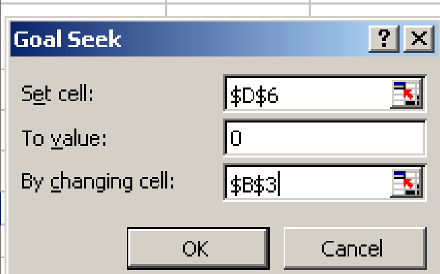
	A	B	C	D	E
1	Item	Number of pounds	Dollars per pound	Dollars	
2	Type A nuts	50	2	100	
3	Type B nuts	3	6	18	
4	mixture			118	adding down (to get total $$ for mixture by adding $$ for ingredients)
5	mixture	53	5	265	multiplying across (to get total $$ for mixture by multiplying total weight of the mixture, times the mixture's cost per pound)
6				-147	"Error" --- the difference between "adding down" and "multiplying across"

By plugging in a few other numbers, we can convince ourselves that the formulas are correct.

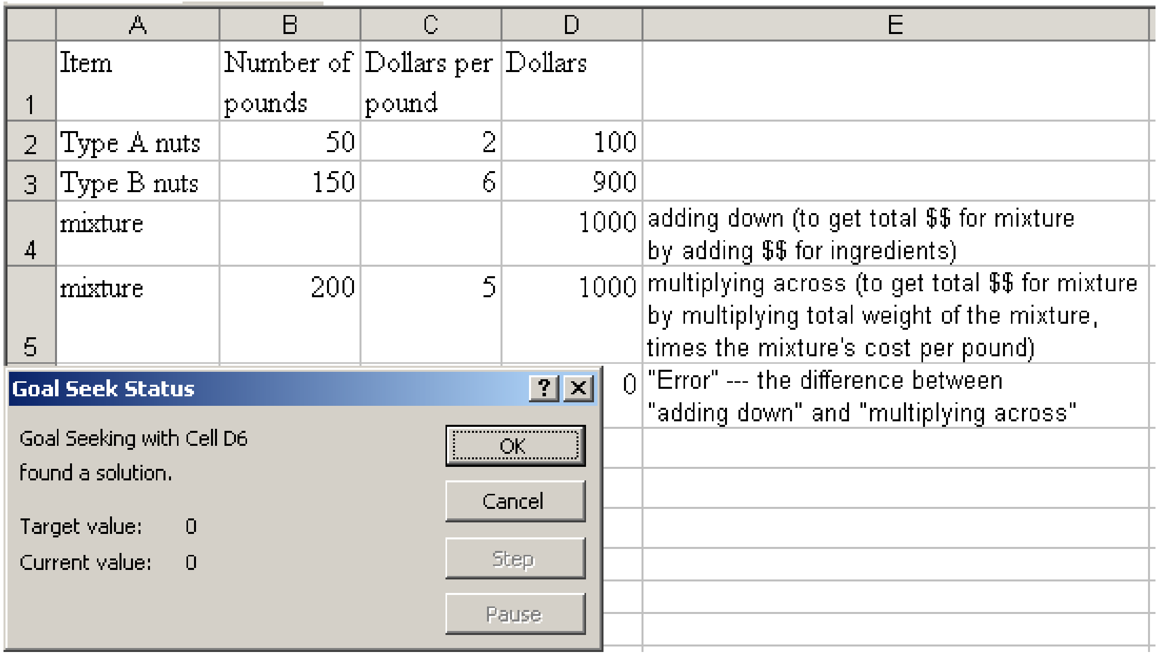
But obviously, we do not yet have the correct value for "number of pounds of $6 nuts", because the current number (3) gives different values for the "adding down" and "multiplying across" ways to compute total dollars for the mixture.

No problem — fire up Goal Seek. It is only at <u>this</u> point that cell B3 becomes an "unknown". We want to know the value for cell B3 that will cause the "adding down" and "multiplying across" values to become the same — at which point the "error" number will become zero.

	A	B	C	D	E
1	Item	Number of pounds	Dollars per pound	Dollars	
2	Type A nuts	50	2	100	
3	Type B nuts	3	6	18	
4	mixture			118	adding down (to get total $$ for mixture by adding $$ for ingredients)
5	mixture	53	5	265	multiplying across (to get total $$ for mixture by multiplying total weight of the mixture, times the mixture's cost per pound)
6				-147	"Error" --- the difference between "adding down" and "multiplying across"
7					
8					
9					
10					
11					
12					



This time, there's a bit more activity on the screen, and then the results settle down.



And there's our answer: 150 pounds of $6 nuts.

Notice that Goal Seek was able to solve our original problem with a lot less work and a lot less chance for error than using classic algebra. In addition, we can easily re-use the same spreadsheet to solve lots of other problems, with different weights, different costs, or even different unknowns (!), at no more than the cost of plugging in different numbers and re-running Goal Seek.

Neat! Are there any problems that Goal Seek can't solve?

Sure, there are lots of them.

The biggest limitation on Goal Seek is that it only works with one input and one output, and it only knows how to make the output equal a specific value.

If your problem has two or more inputs (unknowns), or you want to maximize or minimize some output, instead of setting it to a specified value, then you need a more powerful tool.

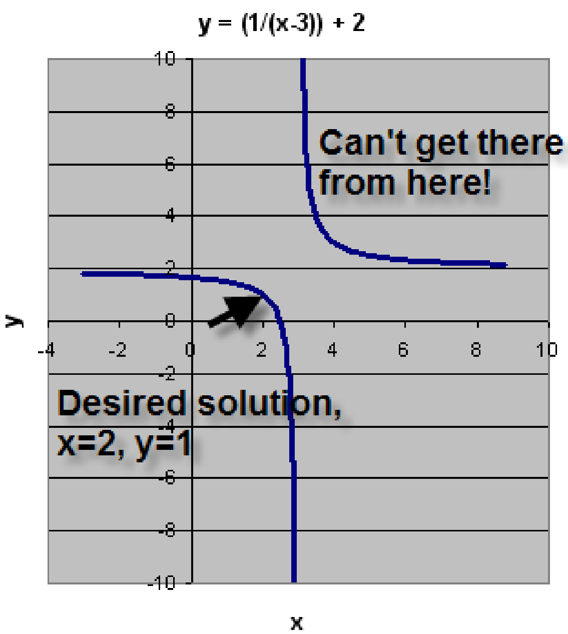
Excel has one of those too. It's called the "Solver", and we'll get to that in another lesson.

But even when there's only one input and one output, and the problem has a solution, Goal Seek might not be able to find it without some help. That's because Goal Seek is really just an automated procedure for "successive approximation". What that means is that Goal Seek takes the current value for the unknown cell as a guess, tests nearby values to see if the guess is too large or too small, adjusts the value to match, and repeats this process. In graphical terms, it's trying to move along a line on a graph until that line reaches the requested value.

For many problems, this procedure is quick and foolproof because the line on the graph rises or falls continuously. For example, Goal Seek can easily solve any linear equation because its graph is a straight line.

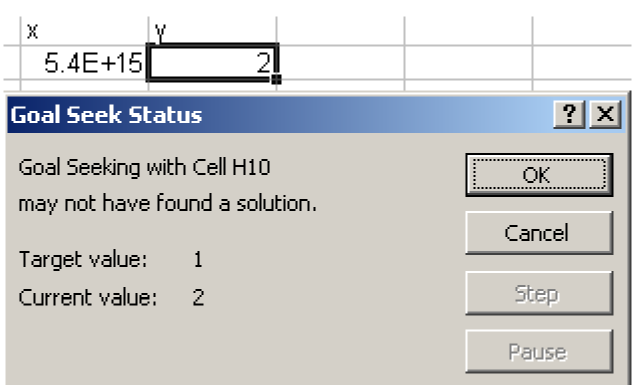
However, Goal Seek can have problems if the line on the graph does <u>not</u> rise or fall continuously. For example, suppose we tell Goal Seek to find x such that $\boxed{\frac{1}{x-3}+2=1}$.

The function $y=\frac{1}{x-3}+2$ has a graph that looks like this:



Goal Seek will have no trouble finding the answer $x = 2$, **<u>if</u>** you start it with any guess that is less than $x = 3$.

But you start it with $x > 3$, then it gets stuck on the upper right part of the curve, pushes x larger and larger in a fruitless attempt to get down to $y = 1$, and ends complaining that it can't find a solution:



Whenever you see something like this happen, you need to look again at what you have asked Goal Seek to do. **The problem has to have a solution, and that solution has to be reachable from your initial guess.**