

Linear Interpolation Using Excel*

David B. Thompson, Ph.D., P.E., P.H.
Thompson Hydrologics
2076 Hawaii Circle
Carson City, Nevada 89701

Draft: 17 October 2018

1. Background

A common problem in engineering and the sciences is estimation of an unknown value of a dependent variable from a known value of an independent variable and a table of corresponding values of the underlying function. The assumption is that the function behaves linearly (or close enough) between tabulated values. Before the development of handheld calculators and desktop computers, this was the common way of estimating values from logarithmic and trigonometric tables for subsequent calculations. The calculations to accomplish linear interpolation were all done by hand.

Over the course of my career there has been a distinct move to use of spreadsheet software for engineering computations. What we used to do with pencil, paper, and a hand calculator (or slide rule) is now done at the computer with general purpose software. But, there are still times when estimating a value between entries in a table (however those entries are derived) is still necessary. A common example is routing a hydrograph through a reservoir using Modified Puls routing. This requires many computations of values between elements of a couple of tables and linear interpolation is the preferred approach.

Furthermore, I have encountered a number of spreadsheets in which linear interpolation was needed and done by others. In every case the functions were hard-coded and had to be manually entered and adjusted for each value interpolated in the table. This is both inefficient and prone to errors. I wanted a general approach, or an approach that could be generalized to reduce multiplication of effort and the opportunity for hand-editing to introduce coding errors.

*Copyright © 2018 David B. Thompson, all rights reserved. Comments are welcome; please direct comments to dave@drdbthompson.net.

2. Linear Interpolation

Let y be the dependent variable and x be the independent variable. The target independent variable will be called \hat{x} and the resulting dependent variable \hat{y} . Bounding pairs of the target independent and dependent variables are (x_i, y_i) and (x_{i+1}, y_{i+1}) . With these declarations, the linear interpolation function is based on the linear equation, $y = mx + b$ and is given by:

$$\hat{y} = \frac{(y_{i+1} - y_i)}{(x_{i+1} - x_i)}(\hat{x} - x_i) + y_i. \quad (1)$$

The slope of the linear interpolation equation over the interpolation interval is $\Delta y / \Delta x$ in Equation 1. The first term on the RHS of Equation 1 is the fraction of the distance between the dependent variables y_i and y_{i+1} associated with the target independent variable \hat{x} over the interpolation range of the independent variables x_i and x_{i+1} . The second term on the RHS of Equation 1 is the base value of the dependent value of the interpolation interval.

The trick is to turn Equation 1 into code suitable for use in Microsoft Excel. This requires investigation into one of the functions available in that software. Other spreadsheet programs probably offer an direct substitution, but I am motivating my discussion.

3. INDEX and MATCH

In Excel, Microsoft provides several table lookup functions. Two of them have general application. The first is `MATCH`, which scans a sorted table¹ and returns the row number of the table that contains the last value less than or equal to the target². The second useful function, `INDEX`, will search a table for a particular cell. The combination of `MATCH` and `INDEX` provides the ability to look up values in a sorted table that bound a target value so that Equation 1 can be applied to the values extracted from the table.

A named range can be used for the table, which makes the function call to `INDEX` much cleaner. In addition, the table must include at least the range of the independent variable. The process will fail if the lookup table does not cover the range of the independent variable.

4. Lookup Table

Construct a table of values of interest to interpolation. An example table is provided in Figure 1. This is from an actual application I constructed for this process. The table could be arranged in a variety of ways, but I chose to list Storage and Elevation.

¹Actually, `MATCH` will scan an unsorted table, but that is not of interest to this problem.

²`MATCH` is actually more flexible than this, but again that flexibility is not of interest to this problem.

Linear Interpolation in Excel

	A	B	C	D	E
	Storage Volume (ac-ft)	Elevation		Volume (af)	Stage (ft)
1					
2	0.00	4911		1000	4915.8
3	0.03	4,912			
4	0.13	4,913			
5	40.1	4,914			
6	397.1	4,915			
7	1143.2	4,916			
8	1994.8	4,917			
9	2904.7	4,918			
10	3891.4	4,919			
11	4985.1	4,920			
12	6206.0	4,921			
13	7582.2	4,922			
14	9131.2	4,923			
15	10835.3	4,924			
16	12709.0	4,925			
17	14770.0	4,926			
18					

Figure 1: Screen capture of a lookup table, independent variable, and result from an implementation of this process in Excel.

In this example, I defined a name for the interpolation table, `tablexy`. There are a variety of ways this could be done. Mine is just one example.

The next step is to develop `MATCH` and `INDEX` calls to return the values necessary to compute Equation 1. These calls are presented in Table 1. This indirection is needed to return bounding values of the independent variable, \hat{x} . An important point is that the calls in Table 1 assumes that the dependent variable is tabulated in values separated by one. If the lookup table is tabulated with a Δy different than that, then the calls must be adjusted. Furthermore, I recommended tabulating the values in the lookup table with a constant Δy . The process documented herein relies on that requirement to generalize the approach.

Table 1: `INDEX` and `MATCH` calls to return values needed for linear interpolation.

Variable	Call
x_i	<code>INDEX(tablexy,MATCH(\hat{x},\$A2:\$A17),1)</code>
x_{i+1}	<code>INDEX(tablexy,MATCH(\hat{x},\$A2:\$A17)+1,1)</code>
y_i	<code>INDEX(tablexy,MATCH(\hat{x},\$A2:\$A17),2)</code>
y_{i+1}	<code>INDEX(tablexy,MATCH(\hat{x},\$A2:\$A17)+1,2)</code>

5. Putting It Together

The final step is to assemble the various parts into a line of code in an Excel cell. The Excel function looks like:

```
=(INDEX(TableXY,MATCH(D3,A2:A17)+1,2)-INDEX(TableXY,MATCH(D3,A2:A17),2)) /  
  (INDEX(TableXY,MATCH(D3,A2:A17)+1,1)-INDEX(TableXY,MATCH(D3,A2:A17),1))*  
  (D3-INDEX(TableXY,MATCH(D3,A2:A17),1))+INDEX(TableXY,MATCH(D3,A2:A17),2)
```

This is an actual code snippet from a cell in an Excel worksheet³. In this code snippet, E2 is the target value for \hat{x} . Each function call to INDEX returns a table entry according to Table 1. The cell then contains the estimate \hat{y} from the table of function pairs (x, y) based on linear interpolation. There is an assumption embedded in this particular instance of linear interpolation — the difference between x values is one⁴. The equation represented in the code snippet looks complex, but reference between the snippet and the values presented in Table 1 should provide some clarity.

An example of an implementation I recently completed is presented in Figure 1. In the figure, the lookup table is shown on the left. The independent variable (storage) is duplicated in Columns A and C. The dependent variable (stage) is in Column B. The target independent variable, \hat{x} is in Column E and the function and result in Column F.

This implementation of linear interpolation uses a concept called indirect referencing. That is, a calculated value (the nested reference to INDEX and MATCH) is used to reference a variable. It is a technique I learned by programming hand calculators back in the 1980s for manipulating register (storage) locations for variables I wanted to save.

6. Final Thoughts

I developed this approach when teaching reservoir routing to young civil engineers a couple of decades ago. I then lost my notes and had to redevelop the code for a recent application. This time I decided to document my work so that it could be recovered relatively easily.

There are many pages on the Internet that describe linear interpolation in Excel using the INDEX and MATCH functions. In reviewing them for this technical note, I found them not well documented and difficult to understand and implement. Therefore, I decided to write my notes and post them on my webpage. Perhaps others in search of a relatively simple mechanic to implement linear interpolation in a table using Microsoft Excel will find these notes useful.

³A screen capture of the worksheet is presented in Figure 1.

⁴That is because my application often involves relations between reservoir stage in feet (calculated at an increment of one foot) and storage or surface area. Other values are possible and are easily implemented once the code and process is understood.