Interpolation

Often it's necessary to interpolate between values in a table. You can use simple linear interpolation, which uses a straight line relationship between two adjacent values. Linear interpolation can be adequate if the table values are close together, as in Figure 5-6. Most often, though, an interpolation formula that fits a curve through several data points is necessary; cubic interpolation, in which four data points are used for interpolation, is common. The following sections describe methods for performing linear interpolation or cubic interpolation.

Linear Interpolation in a Table by Means of Worksheet Formulas

To find the value of y at a point x that is intermediate between the table values x_0, y_0 and x_1, y_1, use the equation for simple linear interpolation (equation 5-1).

$$y_x = y_0 + \frac{(x - x_0)}{(x_1 - x_0)}(y_1 - y_0) \qquad (5\text{-}1)$$

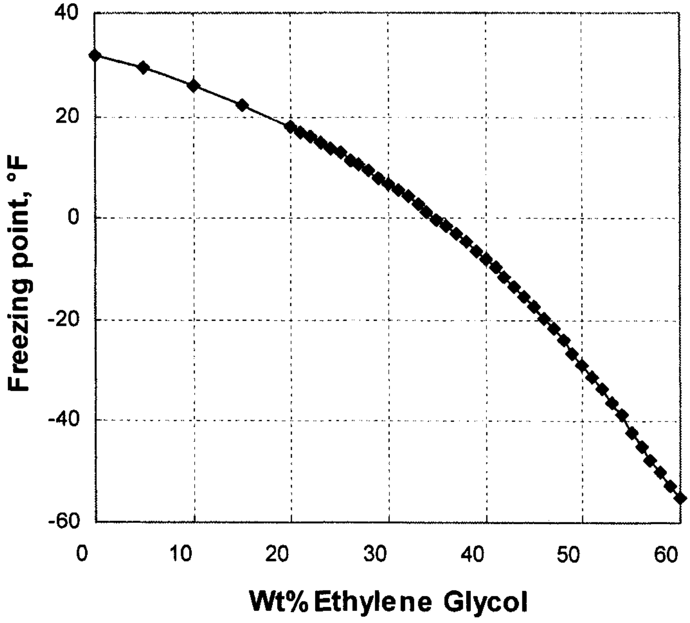


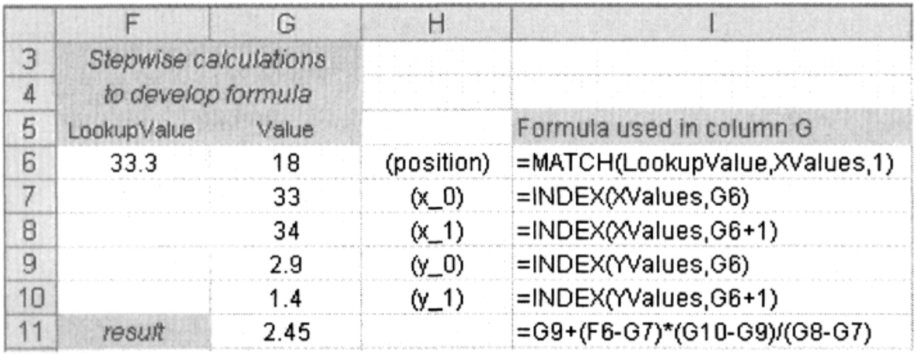
Figure 5-6. Freezing point of ethylene glycol solutions (data from Figure 5-1).
(folder 'Chapter 05 Interpolation', workbook 'Interpolation I', sheet 'Linear Interpolation')

In the following example, we'll assume that values of the independent variable x in the table are in ascending order, as in Figure 5-1, where the independent variable is wt% ethylene glycol. We want to find the freezing point for certain wt% values. Figure 5-2 shows the data (see folder 'Chapter 05 Interpolation', workbook 'Interpolation I', sheet 'Linear Interpolation'); it's clear that, since most of the points are close together, we can use linear interpolation without introducing too much error.

You can create a linear interpolation formula using Excel's MATCH and INDEX functions. If *match_type_num* = 1, MATCH returns the position of the largest array value that is less than or equal to *lookup_value*. The array must be in ascending order. Use this value in the INDEX function to return the values of x_0, y_0, x_1 and y_1, as shown in the following:

position	=MATCH(lookup_value,known_x´s,1)
x_0	=INDEX(known_x´s,position)
x_1	=INDEX(known_x´s,position+1)
y_0	=INDEX(known_y´s,position)
y_1	=INDEX(known_y´s,position+1)

The preceding formulas were applied to the data shown in Figure 5-1 to find the freezing point of a 33.3 wt% solution of ethylene glycol. The following named ranges were used in the calculations: known_x's ($A$3:$A$47), known_y's ($B$3:$B$47), lookup_value ($F$6), position ($G$6). The intermediate calculations and the final interpolated value are shown in Figure 5-7.

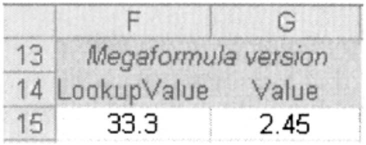


	F	G	H	I
3	Stepwise calculations			
4	to develop formula			
5	LookupValue	Value		Formula used in column G
6	33.3	18	(position)	=MATCH(LookupValue,XValues,1)
7		33	(x_0)	=INDEX(XValues,G6)
8		34	(x_1)	=INDEX(XValues,G6+1)
9		2.9	(y_0)	=INDEX(YValues,G6)
10		1.4	(y_1)	=INDEX(YValues,G6+1)
11	result	2.45		=G9+(F6-G7)*(G10-G9)/(G8-G7)

Figure 5-7. Linear interpolation: intermediate calculations.
(folder 'Chapter 05 Interpolation', workbook 'Interpolation I', sheet 'Linear Interpolation')

The formulas in cells G6:G11 can be combined into a single "megaformula" for linear interpolation, shown below and used in cell G15.

```
=INDEX(YValues,MATCH(LookupValue,XValues,1))+(F15-INDEX(XValues,
MATCH(LookupValue,XValues,1)))*(INDEX(YValues,MATCH(LookupValue,
XValues,1)+1)-INDEX(YValues,MATCH(LookupValue,XValues,1)))/
(INDEX(XValues,MATCH (LookupValue,XValues,1)+1)-INDEX(XValues,
MATCH(LookupValue,XValues,1)))
```



	F	G
13	*Megaformula version*	
14	LookupValue	Value
15	33.3	2.45

Figure 5-8. Linear interpolation: final interpolated value.
(folder 'Chapter 05 Interpolation', workbook 'Interpolation I', sheet 'Linear Interpolation')

If you use the megaformula, the formulas in cells G6:G11 are no longer required.

Linear Interpolation in a Table by Using the TREND Worksheet Function

Excel provides the TREND worksheet function to perform linear interpolation in a table of data by means of a linear least-squares fit to all the data points in the table. But TREND can be used to perform linear interpolation between two adjacent data points.

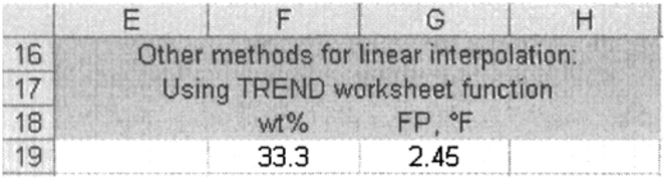
The syntax of the TREND function is

TREND(***known_y's***, *known_x's, new_x's, const*)

where *known_y's* and *known_x's* are one-row or one-column ranges of known values. The argument *new_x's* is a range of cells containing x values for which you want the interpolated value. Use the argument *const* to specify whether the linear relationship $y = mx + b$ has an intercept value; if *const* is set to FALSE or zero, b is set equal to zero.

The TREND worksheet function provides a way to perform linear interpolation between two points without the necessity of creating a worksheet formula. Using the TREND function to perform the linear interpolation calculation that was illustrated in Figure 5-7 is shown in Figure 5-9. Cell G18 contains the formula

```
=TREND(B20:B21,A20:A21,F18,1)
```



	E	F	G	H
16	Other methods for linear interpolation:			
17	Using TREND worksheet function			
18		wt%	FP, °F	
19		33.3	2.45	

Figure 5-9. Using the TREND worksheet function for linear interpolation. (folder 'Chapter 05 Interpolation', workbook 'Interpolation I', sheet 'Linear Interpolation')

Note that although TREND can be used to find the least-squares straight line through a whole set of data points, to perform linear interpolation you must select only two bracketing points, in this example in rows 20 and 21. It should be clear from Figure 5-6 that the least-squares straight line through all the data points will not provide the correct interpolated value.

You can also use TREND for polynomial (e.g., cubic) interpolation by regressing against the same variable raised to different powers (see "Cubic Interpolation in a Table by Using the TREND Worksheet Function" later in this chapter.)

Linear Interpolation in a Table by Means of a Custom Function

The linear interpolation formula can also be easily coded as a custom function, as shown in Figure 5-10.

```
Function InterpL(lookup_value, known_x´s, known_y´s)

Dim pointer As Integer
Dim X0 As Double, Y0 As Double, X1 As Double, Y1 As Double

pointer = Application.Match(lookup_value, known_x´s, 1)
X0 = known_x´s(pointer)
Y0 = known_y´s(pointer)
X1 = known_x´s(pointer + 1)
Y1 = known_y´s(pointer + 1)
InterpL = Y0 + (lookup_value - X0) * (Y1 - Y0) / (X1 - X0)
End Function
```

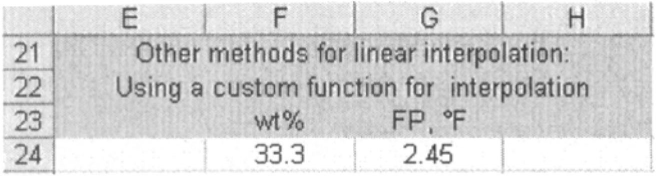
Figure 5-10. Function procedure for linear interpolation. (folder 'Chapter 05 Interpolation', workbook 'Interpolation I', module 'LinearInterpolation')

The syntax of the function is

InterpL(*lookup_value,known_x´s,known_y´s*).

The argument *lookup_value* is the value of the independent variable for which you want the interpolated *y* value; *known_x´s* and *known_y´s* are the arrays of independent and dependent variables, respectively, that comprise the table. The table must be sorted in ascending order of *known_x´s*. Figure 5-11 illustrates the use of the custom function to interpolate values in the table shown in Figure 5-1; cell G24 contains the formula

```
=InterpL(F22,$A$3:$A$54,$B$3:$B$54)
```



	E	F	G	H
21	Other methods for linear interpolation:			
22	Using a custom function for interpolation			
23		wt%	FP, °F	
24		33.3	2.45	

Figure 5-11. Using the InterpL function for linear interpolation.
(folder 'Chapter 05 Interpolation', workbook 'Interpolation I', sheet 'Linear Interpolation')

The custom function can be applied to tables in either vertical or horizontal format.