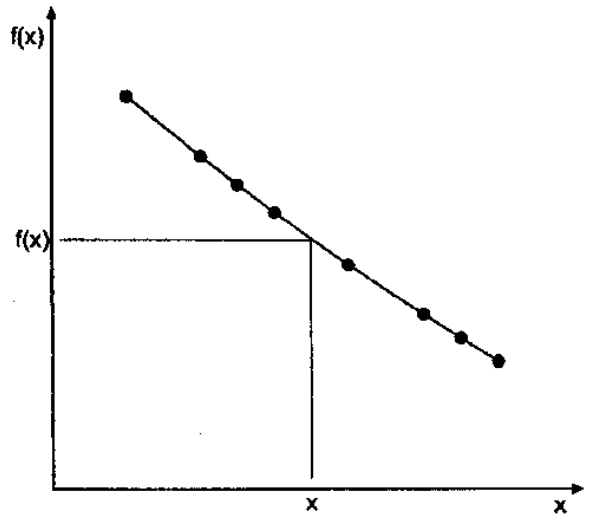
Chapter 3

Interpolation and Curve Fitting

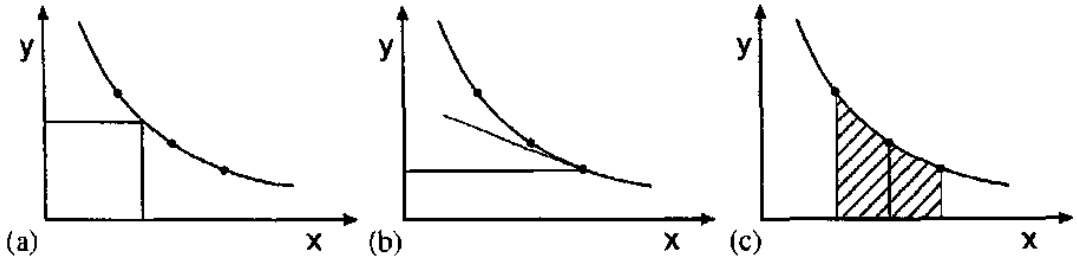
3.1 Introduction

The Figure below illustrates a set of tabular data in the form of a set of **[x, f(x)]** pairs. The function ***f(x)*** is known at a finite set (actually eight) of discrete values of ***x***. The value of the function can be determined at any of the eight values of ***x*** simply by a table lookup. However, a problem arises when the value of the function is needed at any value of ***x*** between the discrete values in the table. The actual function is not known and cannot be determined form the tabular values. However, the actual function can be approximated by some known function, and the value of the approximating function can be determined at any desired value of ***x***. This process, which is called ***interpolatio***n, is the subject of this Chapter. The discrete data of the figure below are actually values of the function $f(x) = 1/x$.

In many applications, the values of the discrete data at the specific points are not all that is needed. Values of the function at points other than the known discrete points may be needed (i.e. ***interpolation***). The derivative of the function may be required (i.e. ***differentiation***). The integral of the function may be of interest (i.e. ***integration***). These processes are performed by ***fitting*** an approximating function to the set of discrete data and performing the desired process on the approximating function.



x	f(x)
3.20	0.312500
3.30	0.303030
3.35	0.298507
3.40	0.294118
3.50	0.285714
3.60	0.277778
3.65	0.273973
3.70	0.270270



a) interpolation ***b) differentiation*** ***c) integration***

Many types of approximating functions exist. In fact, any analytical function can be used as an approximating function. Three of the more common approximating functions are:

1. Polynomials.
2. Trigonometric functions.
3. Exponential functions.

Approximating functions should have the following properties:

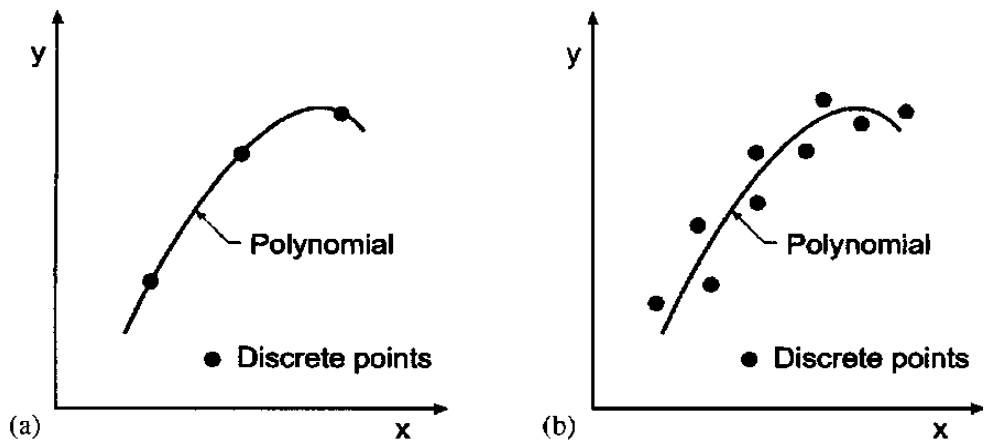
1. The approximating function should be easy to determine.
2. It should be easy to evaluate.
3. It should be easy to differentiate.
4. It should be easy to integrate.

Polynomials satisfy all four of these properties. Consequently, polynomial approximating functions are used here to fit sets of discrete data for interpolation, differentiation, and integration.

There are tow fundamentally different ways to fit a polynomial to a set of discrete data:

1. Exact fits.
2. Approximate fits.

An ***exact fit*** yields a polynomial that passes exactly through all of the discrete points, as illustrated in the figure below. This type of fit is useful for small sets of smooth data. An ***approximate fit*** yields a polynomial that passes through the set of data in the best manner possible, without being required to pass exactly through any of the data points. Approximate fits are useful for large sets of smooth data and small or large sets of rough data.



Polynomial approximation. (a) Exact fit. (b) Approximate fit.

3.2 Direct Fit Polynomials

First let's consider a completely general procedure for fitting a polynomial to a set of equally spaced or unequally spaced data. Given ***n+1*** sets of data $[x_o, f(x_o)], [x_1, f(x_1)], ..., [x_n, f(x_n)]$, which will be written as (x_o, f_o), $(x_1, f_1), ..., (x_n, f_n)$, determine the unique ***nth-degree*** polynomial $\boldsymbol{P_n(x)}$ that passes exactly through the ***n+1*** points:

$$P_n(x) = a_o + a_1 x + a_2 x^2 + ...+ a_n x^n \quad ..... \quad (1)$$

For simplicity of notation, let $f(x_i) = f_i$. Substituting each data point into Eq. (1) yields ***n+1*** equations:

$$f_o = a_o + a_1 x_o + a_2 x_o^2 + ...+ a_n x_o^n \quad ..... \quad (2.0)$$

$$f_1 = a_o + a_1 x_1 + a_2 x_1^2 + ...+ a_n x_1^n \quad ..... \quad (2.1)$$

.....

$$f_n = a_o + a_1 x_n + a_2 x_n^2 + ...+ a_n x_n^n \quad ..... \quad (2.n)$$

There are ***n+1*** linear equations containing the ***n+1*** coefficients $\boldsymbol{a_0}$ to $\boldsymbol{a_n}$. Equation (2) can be solved for $\boldsymbol{a_o}$ to $\boldsymbol{a_n}$ by Gauss elimination. The resulting polynomial is the unique ***nth-degree*** polynomial that passes exactly through the ***n+1*** data points. The direct fit polynomial procedure work for both equally spaced data and unequally spaced data.

Example: Direct fit polynomials.

Solution: To illustrate interpolation by a direct fit polynomial, consider the simple function $y = f(x) = 1/x$, and construct the following set of six significant figure data:

x	***f(x)***
3.35	0.298507
3.40	0.294118
3.50	0.285714
3.60	0.277778

Let's interpolate for ***y*** at ***x = 3.44*** using linear, quadratic, and cubic interpolation. The exact value is

$$y(3.44) = f(3.44) = \frac{1}{3.44} = 0.290698...$$

Let's illustrate the procedure in detail for a quadratic polynomial:

$$P_2(x) = a + b\,x + c\,x^2$$

To center data around ***x = 3.44***, the first three points are used. Applying ***P₂(x)*** at each of these data points gives the following three equations:

$$0.298507 = a + b(3.35) + c(3.35)^2$$
$$0.294118 = a + b(3.40) + c(3.40)^2$$
$$0.285714 = a + b(3.50) + c(3.50)^2$$

Solving theses three equations for ***a***, ***b***, and ***c*** by Gauss elimination without scaling or pivoting yields

$$P_2(x) = 0.876561 - 0.256080x + 0.0249333x^2$$

Substituting ***x = 3.44*** in the polynomial gives

$$P_2(3.44) = 0.876561 - 0.256080(3.44) + 0.0249333(3.44)^2 = 0.290697$$

The error is $Error(3.44) = P_2(3.44) - f(3.44) = 0.290697 - 0.290698 = -0.000001$.

For a linear polynomial, use ***x = 3.4*** and ***3.5*** to center that data around ***x = 3.44***. The resulting linear polynomial is

$$P_1(x) = 0.579854 - 0.0840400\,x$$

Substituting ***x = 3.44*** in the polynomial gives $P_1(3.44) = 0.290756$

For cubic polynomial, all four points must be used. The resulting cubic polynomial is

$$P_3(x) = 1.121066 - 0.470839x + 0.0878000x^2 - 0.00613333x^3$$

Substituting ***x = 3.44*** in the polynomial gives $P_3(3.44) = 0.290698$.

The results are summarized below, where the results of linear, quadratic, and cubic interpolation and the errors, $Error(3.44) = p_n(3.44) - 0.290698$, are tabulated. The advantages of higher-degree interpolation are obvious.

P(3.44) = 0.290756	***Linear interpolation***	Error = 0.000058
=0.290697	***Quadratic interpolation***	=-0.000001
=0.290698	***Cubic interpolation***	=0.000000

The main advantage of direct fit polynomials is that the explicit form of the approximating function is obtained, and the interpolation at several values of ***x*** can be accomplished simply by evaluating the polynomial at each value of ***x***. The work required to obtain the polynomial does not have to be redone for each value of ***x***. A second advantage is that the data can be unequally spaced.

The main disadvantage of direct fit polynomials is that each time the degree of the polynomial is changed, all of the work required to fit the new polynomial must be redone.

3.2.1 Direct Multivariate Polynomial Approximation

Many problems arise in engineering and science when the dependant variable is a function of two or more independent variables, for example, ***z = f(x.y)*** is a two variable or bivariate, function. Such functions in general called multivariate functions. When multivariate function is given by tabular data, multivariate approximation is required for interpolation, differentiation, and integration.

Consider the bivariate function, ***z = f(x.y)***, and the set of tabular data presented in the following table. The tabular data can be fit be a multivariate polynomial of the form

$$z = f(x, y) = a + bx + cy + dxy + ex^2 + fy^2 + gx^2y + hxy^2 + ix^3 + jy^3 + ...$$

The number of data points must equal the number of coefficients in the polynomial. A linear bivariate polynomial in ***x*** and ***y*** is obtained by including the first four terms in the equation. A quadratic bivariate polynomial in ***x*** and ***y*** is obtained by including the first eight terms in the equation. The number of terms in approximating polynomial increases rapidly as the degree of approximations increases. This leads to ill-conditioned systems of linear equations for determining the coefficients. Consequently, multivariate high-degree approximation must be used with caution.

Example: Direct multivariate linear interpolation.

Solution: Consider the following values, use direct multivariate linear interpolation to calculate ***z(x,y) = z(1100, 1225)***,

	x		
y	800	1000	1200
1150	1380.4	1500.2	1614.5
1200	1377.7	1499.0	1613.6
1250	1375.2	1497.1	1612.6

The form of the approximating polynomial is

$$z = f(x, y) = a + bx + cy + dxy$$

Substituting the four data points that bracket ***x = 1100*** and ***y = 1225*** into the polynomial gives

$$1499.0 = a + (1000)b + (1200)c + (1000)(1200)d$$
$$1497.1 = a + (1000)b + (1250)c + (1000)(1250)d$$
$$1613.6 = a + (1200)b + (1200)c + (1200)(1200)d$$
$$1612.6 = a + (1200)b + (1250)c + (1200)(1250)d$$

Solving for ***a***, ***b***, ***c***, and ***d*** by Gauss elimination yields

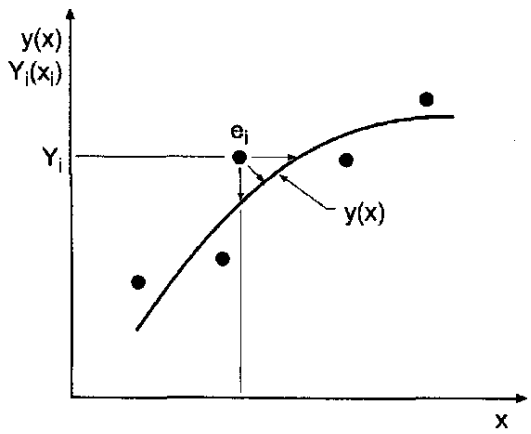
$$z = f(x, y) = 1079.6 + 0.4650x - 0.1280y + 0.0900 \times 10^{-3} xy$$

Substitute ***x = 1100*** and ***y = 1225*** in the approximating polynomial gives ***z(1100, 1225)= 1555.5***. The error in this result is $Error = 1555.5 - 1556.0 = -0.5$. The advantage of this approach is that we can use the same approximating polynomial to evaluate other values of ***z(x,y)***,if required, without reevaluating the polynomial coefficients.

3.3 Least Squares Approximation

3.3.1 Introduction

An approximate fit yields a polynomial that passes through the set of points in the best possible manner without being required to pass exactly through any of the points. Several definitions of best possible manner exist. Consider the set of discrete points, $[x_i, Y(x_i)] = (x_i, Y_i)$, and the approximate polynomial ***y(x)*** chosen to represent the set of discrete points, as illustrated in the figure below. The discrete points do not fall on the approximating polynomial. The deviations (i.e. distances) of the points from the approximating function must be minimized in some manner.

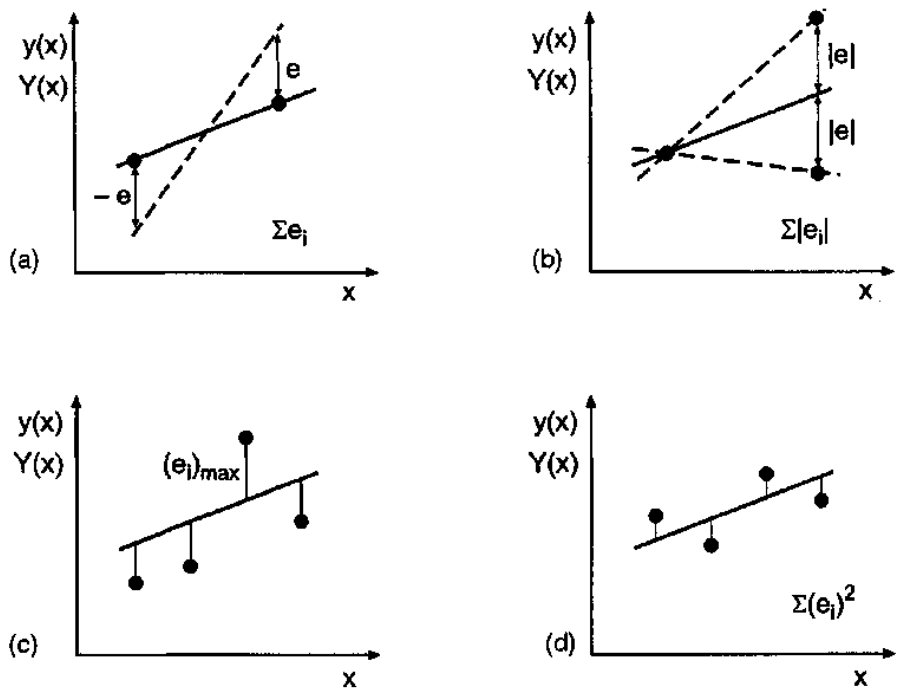


If the values of the independent variable x_i are considered exact, then all the deviation is assigned to the dependant variable Y_i, and the deviation e_i is the vertical distance between Y_i and $y_i=f(x_i)$. Thus,

$$e_i = Y_i - y_i$$

It is certainly possible that the values of Y_i are quite accurate, but the corresponding values of x_i are in error. In that case, the deviation would be measured by the horizontal distance illustrated in the above figure. If x_i and Y_i both have uncertainties in their values, then the perpendicular distance between a point and the approximating function would be the deviation. The usual approach in the approximate fitting of tabular data is to assume that the deviation is the vertical distance between a point and the approximating function, as specified by the above equation.

Several best criteria are illustrated in the next figure for a straight line approximation. From the figure we can see that the least squares criteria, in which the sum of the squares of the deviations is minimized. The least squares procedure yields a good compromise criterion for the best fit approximation.



Best fit criteria. (a) Minimize $\sum e_i$. **(b)** Minimize $\sum |e_i|$ **. (c)** Minimax. **(d)** Least squares.

3.3.2 The Straight Line Approximation

The simplest polynomial is a linear polynomial, the straight lint. Least squares straight line approximations are an extremely useful and common approximate fit, which is determined as follows. Given ***N*** data points, ***(x$_i$,Y$_i$)***, fit the best straight line through the set data. The approximating function is

$$y = a + b\,x \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

At each value of ***x$_i$***, Eq. (1) gives

$$y_i = a + b\,x_i \qquad (i = 1, \ldots\ldots\ldots, N)$$

The deviation ***e$_i$*** at each value of ***x$_i$*** is

$$e_i = Y_i - y_i \qquad (i = 1, \ldots\ldots\ldots, N)$$

The sum of the squares of the deviations defines the function ***S(a,b)***:

$$S(a,b) = \sum_{i=1}^{N} (e_i)^2 = \sum_{i=1}^{N} (Y_i - a - bx_i)^2$$

The function ***S(a,b)*** is a minimum when $\partial S / \partial a = \partial S / \partial b = 0$. Thus,

$$\left.\begin{aligned} \frac{\partial S}{\partial a} &= \sum_{i=1}^{N} 2(Y_i - a - bx_i)(-1) = 0 \\ \frac{\partial S}{\partial b} &= \sum_{i=1}^{N} 2(Y_i - a - bx_i)(-x_i) = 0 \end{aligned}\right| \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$

Dividing Eq. (2) by 2 and rearranging yields

$$\boxed{\begin{aligned} a\,N + b\sum_{i=1}^{N} x_i &= \sum_{i=1}^{N} Y_i \\ a\sum_{i=1}^{N} x_i + b\sum_{i=1}^{N} x_i^{\,2} &= \sum_{i=1}^{N} x_i\,Y_i \end{aligned}}$$

Which is called the ***normal equations*** of the least squares fit. They can be solved for ***a*** and ***b*** by Gauss elimination.

Example: Least squares straight line approximation.

Solution: Consider the constant pressure specific heat for air at low temperatures presented in the following table, where ***T*** is the temperature and C_p is the specific heat. Determine a least squares straight line approximation for the set of data:

T	300	400	500	600	700	800	900	1000
C_p	1.0045	1.0134	1.0296	1.0507	1.0743	1.0984	1.1212	1.1410

$$C_p = a + bT$$

For this problem the equation becomes

$$8a + b\sum_{i=1}^{8} T_i = \sum_{i=1}^{8} C_{p,i}$$

$$a\sum_{i=1}^{8} T_i + b\sum_{i=1}^{8} T_i^2 = \sum_{i=1}^{N} T_i C_{p,i}$$

Evaluating the summations and substituting into the above equations gives

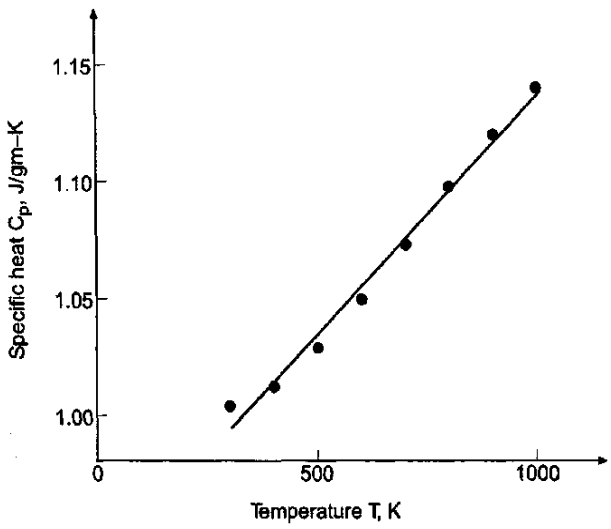
$$8a + 5200b = 8.5331$$
$$5200a + 3{,}800{,}000b = 5632.74$$

Solving for a and b be Gauss elimination without scaling or pivoting yields

$$C_p = 0.933194 + 0.205298 \times 10^{-3} T$$

Substituting the initial values of T into this equation gives the results presented in the next table and figure, which presents the exact data, the least squares straight line approximation, and the percent error. The straight line is not a very good approximation of the data.

T, K	$C_{p,\text{exact}}$	$C_{p,\text{approx.}}$	Error, %
300	1.0045	0.9948	−0.97
400	1.0134	1.0153	0.19
500	1.0296	1.0358	0.61
600	1.0507	1.0564	0.54
700	1.0743	1.0769	0.24
800	1.0984	1.0974	−0.09
900	1.1212	1.1180	−0.29
1000	1.1410	1.1385	−0.22



3.3.3 High-Degree Polynomial Approximation

Given the ***N*** data points, ***(xᵢ ,Yᵢ)***, fit the best ***n***th-degree polynomial through the set of data. Consider the nth-degree polynomial:

$$y = a_o + a_1 x + a_2 x^2 + \dots\dots\dots + a_n x^n$$

The sum of the squares of the deviation is given by

$$S(a_0, a_1, \dots, a_n) = \sum_{i=1}^{N} (e_i)^2 = \sum_{i=1}^{N} (Y_i - a_o - a_1 x_i - \dots - a_n x_i^{\,n})^2$$

The function $S(a_0, a_1, \dots, a_n)$ is a minimum when

$$\frac{\partial S}{\partial a_o} = \sum_{i=1}^{N} 2\,(Y_i - a_o - a_1 x_i - \dots - a_n x_i^{\,n})(-1) = 0$$

$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$

$$\frac{\partial S}{\partial a_n} = \sum_{i=1}^{N} 2\,(Y_i - a_o - a_1 x_i - \dots - a_n x_i^{\,n})(-x_i^{\,n}) = 0$$

Dividing by 2 and rearranging yields the normal equations:

$$\boxed{\begin{aligned} & a_o N + a_1 \sum_{i=1}^{N} x_i + \dots + a_n \sum_{i=1}^{N} x_i^{\,n} = \sum_{i=1}^{N} Y_i \\ & \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ & a_o \sum_{i=1}^{N} x_i^{\,n} + a_1 \sum_{i=1}^{N} x_i^{\,n+1} + \dots + a_n \sum_{i=1}^{N} x_i^{\,2n} = \sum_{i=1}^{N} x_i^{\,n} Y_i \end{aligned}}$$

And this can be solved for ***aₒ*** to ***aₙ*** by Gauss elimination.

Example: Least squares quadratic polynomial approximation.

Solution: Determine a least squares quadratic polynomial approximation for this set of data:

x	1000	1500	2000	2500	3000
Y	1.1410	1.2095	1.2520	1.2782	1.2955

$$y = a + b\,x + c\,x^2$$

For this problem the polynomial becomes

$$a\,N + b\sum x_i + c\sum x_i^{\,2} = \sum Y_i$$
$$a\sum x_i + b\sum x_i^{\,2} + c\sum x_i^{\,3} = \sum x_i\,Y_i$$
$$a\sum x_i^{\,2} + b\sum x_i^{\,3} + c\sum x_i^{\,4} = \sum x_i^{\,2}\,Y_i$$

Or may be represented in matrix form

$$\begin{bmatrix} N & \sum x_i & \sum x_i^{\,2} \\ \sum x_i & \sum x_i^{\,2} & \sum x_i^{\,3} \\ \sum x_i^{\,2} & \sum x_i^{\,3} & \sum x_i^{\,4} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum Y_i \\ \sum x_i\,Y_i \\ \sum x_i^{\,2}\,Y_i \end{bmatrix}$$

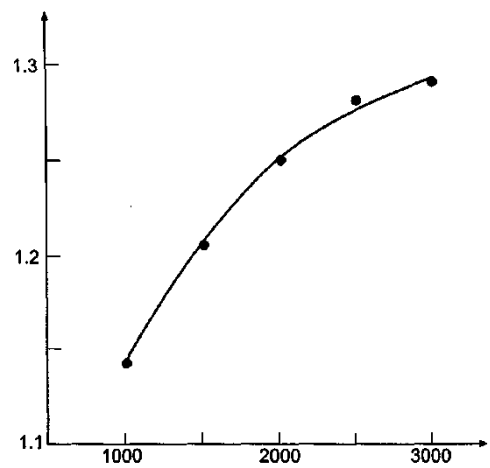
$$\begin{bmatrix} 5 & 10\times10^3 & 22.5\times10^6 \\ 10\times10^3 & 22.5\times10^6 & 55\times10^9 \\ 22.5\times10^6 & 55\times10^9 & 142.125\times10^{12} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 6.1762 \\ 12.5413\times10^3 \\ 288.5186\times10^6 \end{bmatrix}$$

Solving for ***a***, ***b***, and ***c*** by Gauss elimination yields

$$y = 0.965460 + 0.211197\times10^{-3}\,x - 0.0339143\times10^{-6}\,x^2$$

Substituting the initial values of ***x*** into the approximating polynomial gives the results presented in the next table and figure. The quadratic polynomial is a reasonable approximation of the discrete data.

T, K	Y_{exact}	$y_{approx.}$	Error, %
1000	1.1410	1.1427	0.15
1500	1.2095	1.2059	−0.29
2000	1.2520	1.2522	0.02
2500	1.2782	1.2815	0.26
3000	1.2955	1.2938	−0.13



3.3.4 Multivariate Polynomial Approximation

Many problems arises in engineering and science where the dependant variable is a function of two or more independent variable, for example, ***z = f(x,y)*** is a two-variable, or bivariate, function.

Given the ***N*** data points, (***x**$_i$*, ***y**$_i$*, ***Z**$_i$*), fit the best linear bivariate polynomial through the set of data. Consider the linear polynomial:

$$z = a + bx + cy$$

The sum of the squares of the deviations is given by

$$S(a,b,c) = \sum (e_i)^2 = \sum (Z_i - a - bx_i - cy_i)^2$$

The function ***S(a,b,c)*** is a minimum when

$$\frac{\partial S}{\partial a} = \sum 2\,(Z_i - a - bx_i - cy_i)(-1) = 0$$
$$\frac{\partial S}{\partial b} = \sum 2\,(Z_i - a - bx_i - cy_i)(-x_i) = 0$$
$$\frac{\partial S}{\partial b} = \sum 2\,(Z_i - a - bx_i - cy_i)(-y_i) = 0$$

Dividing by 2 and rearranging yields the normal equations:

$$a\,N + b\sum_{i=1}^{N} x_i + c\sum_{i=1}^{N} y_i = \sum_{i=1}^{N} Z_i$$
$$a\sum_{i=1}^{N} x_i + b\sum_{i=1}^{N} x_i^{\,2} + c\sum_{i=1}^{N} x_i\, y_i = \sum_{i=1}^{N} x_i\, Z_i$$
$$a\sum_{i=1}^{N} y_i + b\sum_{i=1}^{N} x_i\, y_i + c\sum_{i=1}^{N} y_i^{\,2} = \sum_{i=1}^{N} y_i\, Z_i$$

Which can be solved for a, b, and c by Gauss elimination.

A linear fit to a set of bivariate data may be inadequate. Consider the quadratic bivariate polynomial:

$$z = a + b\,x + c\,y + d\,x^2 + e\,y^2 + f\,xy$$

The sum of the squares of the deviations is given by

$$S(a,b,c,d,e,f) = \sum (e_i)^2 = \sum (Z_i - a - bx_i - cy_i - dx_i^{\,2} - ey_i^{\,2} - f\,x_i y_i)^2$$

The function ***S(a,b,c,d,e,f)*** is a minimum when

$$\frac{\partial S}{\partial a} = \sum 2\,(Z_i - a - bx_i - cy_i - dx_i^{\,2} - ey_i^{\,2} - f\,x_i y_i)(-1) = 0$$

……

$$\frac{\partial S}{\partial f} = \sum 2\,(Z_i - a - bx_i - cy_i - dx_i^{\,2} - ey_i^{\,2} - f\,x_i y_i)(-x_i y_i) = 0$$

Dividing by 2 and rearranging yields the normal equations:

$$a\,N + b\sum x_i + c\sum y_i + d\sum x_i^{\,2} + e\sum y_i^{\,2} + f\sum x_i y_i = \sum Z_i$$
$$a\sum x_i + b\sum x_i^{\,2} + c\sum x_i y_i + d\sum x_i^{\,3} + e\sum x_i y_i^{\,2} + f\sum x_i^{\,2} y_i = \sum x_i\,Z_i$$
$$a\sum y_i + b\sum x_i y_i + c\sum y_i^{\,2} + d\sum x_i^{\,2} y_i + e\sum y_i^{\,3} + f\sum x_i y_i^{\,2} = \sum y_i\,Z_i$$
$$a\sum x_i^{\,2} + b\sum x_i^{\,3} + c\sum x_i^{\,2} y_i + d\sum x_i^{\,4} + e\sum x_i^{\,2} y_i^{\,2} + f\sum x_i^{\,3} y_i = \sum x_i^{\,2}\,Z_i$$
$$a\sum y_i^{\,2} + b\sum x_i y_i^{\,2} + c\sum y_i^{\,3} + d\sum x_i^{\,2} y_i^{\,2} + e\sum y_i^{\,4} + f\sum x_i y_i^{\,3} = \sum y_i^{\,2}\,Z_i$$
$$a\sum x_i y_i + b\sum x_i^{\,2} y_i + c\sum x_i y_i^{\,2} + d\sum x_i^{\,3} y_i + e\sum x_i y_i^{\,3} + f\sum x_i^{\,2} y_i^{\,2} = \sum x_i y_i\,Z_i$$

Which can be solved for ***a*** to ***f*** by Gauss elimination.

Example: Least squares quadratic bivariate polynomial approximation.

Solution: Consider the following values, use Least squares quadratic bivariate polynomial to calculate ***z(x,y) = z(1100, 1225)***,

	x		
y	800	1000	1200
1150	1380.4	1500.2	1614.5
1200	1377.7	1499.0	1613.6
1250	1375.2	1497.1	1612.6

The form of the approximating polynomial is

$$z = a + b\,x + c\,y + d\,x^2 + e\,y^2 + f\,xy$$

$$\begin{bmatrix} N & \sum x_i & \sum y_i & \sum x_i^2 & \sum y_i^2 & \sum x_i y_i \\ \sum x_i & \sum x_i^2 & \sum x_i y_i & \sum x_i^3 & \sum x_i y_i^2 & \sum x_i^2 y_i \\ \sum y_i & \sum x_i y_i & \sum y_i^2 & \sum x_i^2 y_i & \sum y_i^3 & \sum x_i y_i^2 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^2 y_i & \sum x_i^4 & \sum x_i^2 y_i^2 & \sum x_i^3 y_i \\ \sum y_i^2 & \sum x_i y_i^2 & \sum y_i^3 & \sum x_i^2 y_i^2 & \sum y_i^4 & \sum x_i y_i^3 \\ \sum x_i y_i & \sum x_i^2 y_i & \sum x_i y_i^2 & \sum x_i^3 y_i & \sum x_i y_i^3 & \sum x_i^2 y_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} \sum Z_i \\ \sum x_i Z_i \\ \sum y_i Z_i \\ \sum x_i^2 Z_i \\ \sum y_i^2 Z_i \\ \sum x_i y_i Z_i \end{bmatrix}$$

Evaluating the summation and substituting in the matrix yields

$$\begin{bmatrix} 9E0 & 10.800E3 & 9.000E3 & 12.975E6 & 9.240E6 & 10.800E6 \\ 10.800E3 & 12.975E6 & 10.800E6 & 15.606E9 & 11.088E9 & 12.975E9 \\ 9.000E3 & 10.800E6 & 9.240E6 & 12.975E9 & 9.720E9 & 11.088E9 \\ 12.975E6 & 15.606E9 & 12.975E9 & 18.792E12 & 13.321E12 & 15.606E12 \\ 9.240E6 & 11.088E9 & 9.720E9 & 13.321E12 & 10.450E12 & 11.664E12 \\ 10.800E6 & 12.975E9 & 11.088E9 & 15.606E12 & 11.664E12 & 13.321E12 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix}$$

$$= \begin{bmatrix} 13.4703E3 \\ 16.1638E6 \\ 13.6118E6 \\ 19.4185E9 \\ 14.1122E9 \\ 16.3337E9 \end{bmatrix}$$

A problem arises for high-degree polynomials. The coefficients in the matrix are varying over a range of several orders of magnitude, which gives rise to ill-conditioned system. Normalizing each equation helps the situation. Double precision calculations are required.

Each raw in the matrix should be normalized by the exponential term in the first coefficient of each raw. Solving the normalized equations by Gauss eliminations yields

$$z(x,y) = 914.033 + 0.645500x - 0.020500y - 0.0000775x^2 - 0.000040y^2 + 0.0000825xy$$

Evaluating ***z(1100, 1225) = 1556.3***.

The error is, ***Error = 1556.3 - 1556.0 = 0.3***, which is smaller than the error incurred by interpolation using direct multivariate linear approximation.