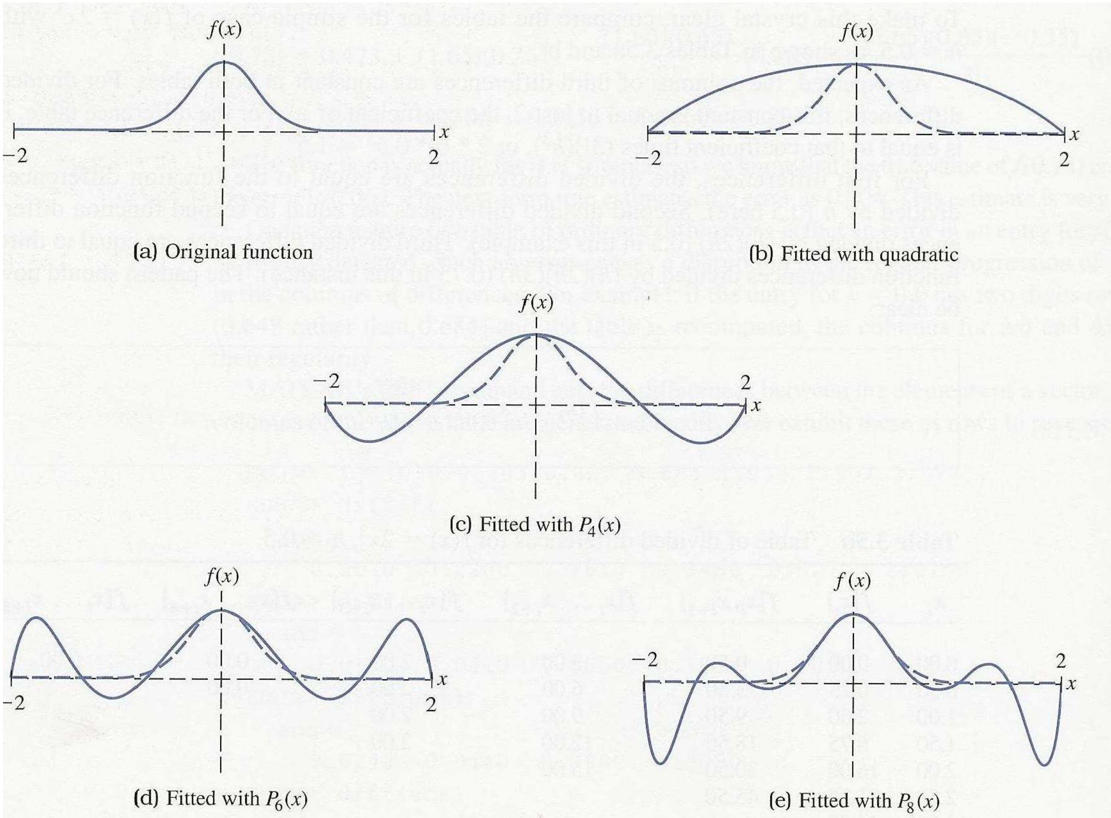


Figure 1: Fitting with different degrees of the polynomial.

# 1 Spline Curves

- There are times when fitting an interpolating polynomial to data points is very difficult. Figure 1a is plot of $f(x) = cos^{10}(x)$ on the interval $[-2, 2]$. It is a nice, smooth curve but has a pronounced maximum at $x = 0$ and is near to the $x$-axis for $|x| > 1$. The curves of Figure 1b,c, d, and e are for polynomials of degrees $-2, -4, -6$, and $-8$ that match the function at evenly spaced points. None of the polynomials is a good representation of the function.

- One might think that a solution to the problem would be to break up the interval $[-2, 2]$ into subintervals and fit separate polynomials to the function in these smaller intervals. Figure 2 shows a much better fit if we use a quadratic between $x = -0.65$ and $x = 0.65$, with $P(x) = 0$ outside that interval. That is better but there are discontinuities in the slope where the separate polynomials join.

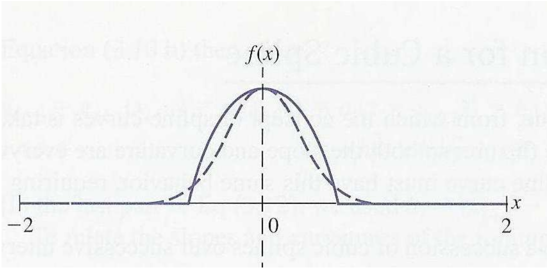


Figure 2: Fitting with quadratic in subinterval.

- An answer to the dilemma is to use *spline curves.* Spline curves may be of varying degrees. Suppose that we have a set of $n+1$ points (which do not have to be evenly spaced):

$$(x_i, y_i),\ \ with\ i = 0, 1, 2, \ldots, n.$$

- A spline fits a set of $n^{th}$-degree polynomials, $g_i(x)$, between each pair of points, from $x_i$ to $x_{i+1}$. The points at which the splines join are called knots.

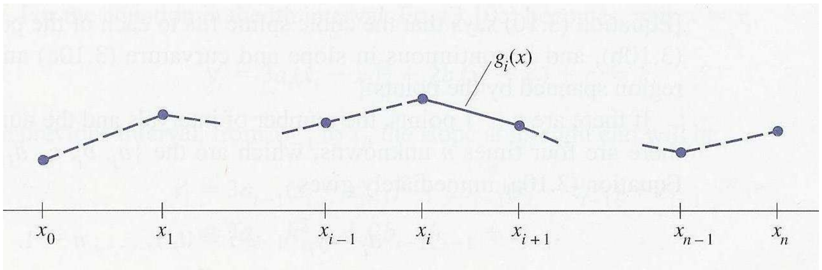


Figure 3: Linear spline.

- If the polynomials are all of degree-1, we have a *linear spline* and the curve would appear as in the Fig. 3. The slopes are discontinuous where the segments join.

## 1.1 The Equation for a Cubic Spline

- We will create a succession of cubic splines over successive intervals of the data (See Fig. 4). Each spline must join with its neighboring cubic polynomials at the knots where they join with the same slope and curvature.

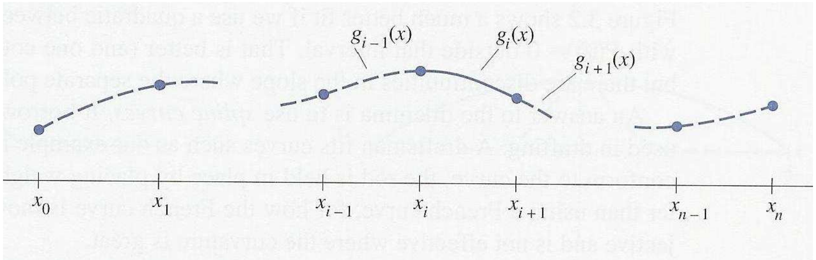


Figure 4: Cubic spline.

- We write the equation for a cubic polynomial, $g_i(x)$, in the $i$th interval, between points $(x_i, y_i), (x_{i+1}, y_{i+1})$. It looks like the solid curve shown here. The dashed curves are other cubic spline polynomials. It has this equation:

$$g_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i$$

Thus, the cubic spline function we want is of the form

$$g(x) = g_i(x) \textit{ on the interval} [x_i, x_{i+1}], \textit{ for } i = 0, 1, \ldots, n-1$$

and meets these conditions:

$$g_i(x_i) = y_i, \ i = 0, 1, \ldots, n-1 \textit{ and } g_{n-1}(x_n) = y_n \quad (1)$$

$$g_i(x_{i+1}) = g_{i+1}(x_{i+1}), \ i = 0, 1, \ldots, n-2 \quad (2)$$

$$g_i^{'}(x_{i+1}) = g_{i+1}^{'}(x_{i+1}), \ i = 0, 1, \ldots, n-2 \quad (3)$$

$$g_i^{''}(x_{i+1}) = g_{i+1}^{''}(x_{i+1}), \ i = 0, 1, \ldots, n-2 \quad (4)$$

- Equations say that the cubic spline fits to each of the points Eq. 1, is continuous Eq. 2, and is continuous in slope and curvature Eq. 3 and Eq. 4, throughout the region spanned by the points.

# 2 Least-Squares Approximations

- Until now, in this chapter we have assumed that the data are accurate, but when these values are derived from an experiment, there is some error in the measurements.

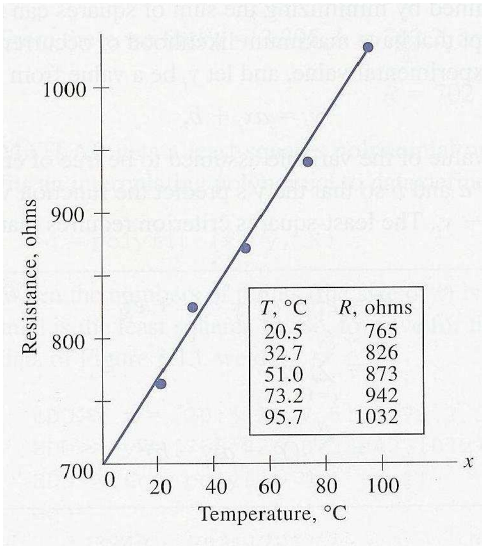


Figure 5: Resistance vs Temperature graph for the Least-Squares Approximation.

- Some students are assigned to find the effect of temperature on the resistance of a metal wire. They have recorded the temperature and resistance values in a table and have plotted their findings, as seen in Fig. 5. The graph suggest a linear relationship.

$$R = aT + b$$

values for the parameters, $a$ and $b$, can be obtained from the plot.

- If someone else were given the data and asked to draw the line, it is not likely that they would draw exactly the same line and they would get different values for $a$ and $b$.

- A way of fitting a line to experimental data that is to minimize the deviations of the points from the line. The usual method for doing this is called the *least-squares method*. The deviations are determined by the distances between the points and the line.

- In analyzing the data, we will assume that the temperature values are accurate and that the errors are only in the resistance numbers; we then will use the vertical distances.

- We might first suppose we could minimize the deviations by making their sum a minimum, but this is not an adequate criterion. Consider

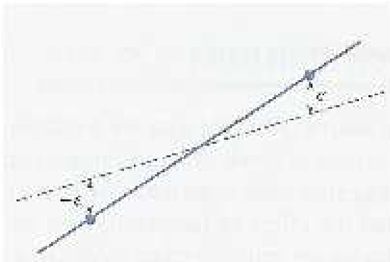

Figure 6: Minimizing the deviations by making the sum a minimum.

the case of only two points (See Fig. 6). Obviously, the best line passes through each point, but any line that passes through the midpoint of the segment connecting them has a sum of errors equal to zero.

- We might accept the criterion that we make the magnitude of the maximum error a minimum (the so-called *minimax* criterion, but for the problem at hand this is rarely done.
- The usual criterion is to minimize the sum of the *squares* of the errors, the *least-squares* principle.
- In addition to giving a unique result for a given set of data, the least-squares method is also in accord with the *maximum-likelihood* principle of statistics. If the measurement errors have a so-called normal distribution and if the standard deviation is constant for all the data, the line determined by minimizing the sum of squares can be shown to have values of slope and intercept that have maximum likelihood of occurrence.
- Let $Y_i$ represent an experimental value, and let $y_i$ be a value from the equation

$$y_i = ax_i + b$$

where $x_i$ is a particular value of the variable assumed to be free of error. We wish to determine the best values for $a$ and $b$ so that the $y$'s predict the function values that correspond to $x$-values. Let $e_i = Y_i - y_i$. The least-squares criterion requires that

$$\begin{aligned} S &= e_1^2 + e_2^2 + \ldots + e_n^2 = \sum_{i=1}^N e_i^2 \\ &= \sum_{i=1}^N (Y_i - ax_i - b)^2 \end{aligned}$$

be a minimum. $N$ is the number of $(x, Y)$-pairs. We reach the minimum by proper choice of the parameters $a$ and $b$, so they are the *variables* of the problem. At a minimum for $S$, the two partial derivatives $\partial S/\partial u$ and $\partial S/\partial b$ will both be zero. Remembering that the $x_i$ and $Y_i$ are data points unaffected by our choice our values for $a$ and $b$, we have

$$\begin{aligned} \frac{\partial S}{\partial a} &= 0 = \sum_{i=1}^{N} 2(Y_i - ax_i - b)(-x_i) \\ \frac{\partial S}{\partial b} &= 0 = \sum_{i=1}^{N} 2(Y_i - ax_i - b)(-1) \end{aligned}$$

Dividing each of these equations by $-2$ and expanding the summation, we get the so-called *normal equations*

$$\begin{aligned} a\sum x_i^2 + b\sum x_i &= \sum x_i Y_i \\ a\sum x_i + bN &= \sum Y_i \end{aligned}$$

All the summations are from $i = 1$ to $i = N$. Solving these equations simultaneously gives the values for slope and intercept $a$ and $b$.

- For the data in Fig. 5 we find that

  $N = 5, \sum T_i = 273.1, \sum T_i^2 = 18,607.27, \sum R_i = 4438, \sum T_i R_i = 254,932.5$

  Our normal equations are then

  $$\begin{aligned} 18,607.27a + 273.1b &= 254,932.5 \\ 273.a + 5b &= 4438 \end{aligned}$$

  From these we find $a = 3.395$, $b = 702.2$, and

  $$R = 702 + 3.39T$$

- MATLAB gets a least-squares polynomial with its *polyfit* command. When the numbers of points (the size of $x$) is greater than the degree plus one, the polynomial is the least squares fit.

```
>> x=[20.5 32.7 51.0 73.2 95.7 ];
>> y=[765 826 873 942 1032];
>> eq=polyfit(x,y,1)
eq= 3.3949 702.1721
```

## 2.1 Nonlinear Data

- In many cases, data from experimental tests are not linear, so we need to fit to them some function other than a first-degree polynomial. Popular forms are the exponential form

  $$y = ax^b$$

  or

  $$y = ae^{bx}$$

- We can develop normal equations to the preceding development for a least-squares line by setting the partial derivatives equal to zero. Such nonlinear simultaneous equations are much more difficult to solve than linear equations. Thus, the exponential forms are usually linearized by taking logarithms before determining the parameters:

$$lny = lna + blnx$$

or

$$lny = lna + bx$$

We now fit the new variable $z = Iny$ as a linear function of $lnx$ or $x$ as described earlier. Here we do not minimize the sum of squares of the deviations of $Y$ from the curve, but rather the deviations of $lnY$.

  - In effect, this amounts to minimizing the squares of the percentage errors, which itself may be a desirable feature.
  - An added advantage of the linearized forms is that plots of the data on either log-log or semilog graph paper show at a glance whether these forms are suitable, by whether a straight line represents the data when so plotted.

- In cases when such linearization of the function is not desirable, or when no method of linearization can be discovered, graphical methods are frequently used; one merely plots the experimental values and sketches in a curve that seems to fit well.

- Transformation of the variables to give near linearity, such as by plotting against $1/x, 1/(ax + b), 1/x^2$, and other polynomial forms of the argument may give curves with gentle enough changes in slope to allow a smooth curve to be drawn. S-shaped curves are not easy to linearize; the relation

$$y = ab^{c^x}$$

is sometimes employed. The constants $a$, $b$, and $c$ are determined by special procedures. Another relation that fits data to an S-shaped curve is

$$\frac{1}{y} = a + be{-}x$$

In awkward cases, subdividing the region of interest into subregions with a piecewise fit in the subregions can be used.

## 2.2 Least-Squares Polynomials

- Because polynomials can be readily manipulated, fitting such functions to data that do not plot linearly is common.
- It will turn out that the normal equations are linear for this situation, which is an added advantage.
- $n$ as the degree of the polynomial and N as the number of data pairs. If $N = n + 1$, the polynomial passes exactly through each point and the methods discussed earlier apply, so we will always have $N > n + l$ in the following. We assume the functional relationship

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n \tag{5}$$

with errors defined by

$$e_i = Y_i - y_i = Y_i - a_0 - a_1 x - a_2 x^2 - \ldots - a_n x^n$$

We again use $Y_i$ to represent the observed or experimental value corresponding to $x_i$, with $x_i$ free of error. We minimize the sum of squares;

$$S = \sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} (Y_i - a_0 - a_1 x - a_2 x^2 - \ldots - a_n x^n)^2$$

At the minimum, all the partial derivatives $\partial S/\partial a_0, \partial S/\partial a_n$ vanish. Writing the equations for these gives $n + 1$ equations:

$$\begin{aligned}
\tfrac{\partial S}{\partial a_0} &= 0 = \textstyle\sum_{i=1}^{N} 2(Y_i - a_0 - a_1 x_i - a_2 x_i^2 - \ldots - a_i x_i^n)(-1) \\
\tfrac{\partial S}{\partial a_1} &= 0 = \textstyle\sum_{i=1}^{N} 2(Y_i - a_0 - a_1 x_i - a_2 x_i^2 - \ldots - a_i x_i^n)(-x_i) \\
&\vdots \\
\tfrac{\partial S}{\partial a_n} &= 0 = \textstyle\sum_{i=1}^{N} 2(Y_i - a_0 - a_1 x_i - a_2 x_i^2 - \ldots - a_i x_i^n)(-x_i^n)
\end{aligned}$$

Dividing each by $-2$ and rearranging gives the $n + 1$ normal equations to be solved simultaneously:

$$\begin{aligned}
a_0 N + a_1 \textstyle\sum x_i + a_2 \sum x_i^2 + \ldots + a_n \sum x_i^n &= \textstyle\sum Y_i \\
a_0 \textstyle\sum x_i + a_1 \sum x_i^2 + a_2 \sum x_i^3 + \ldots + a_n \sum x_i^{n+1} &= \textstyle\sum x_i Y_i \\
a_0 \textstyle\sum x_i^2 + a_1 \sum x_i^3 + a_2 \sum x_i^4 + \ldots + a_n \sum x_i^{n+2} &= \textstyle\sum x_i^2 Y_i \\
&\vdots \\
a_0 \textstyle\sum x_i^n + a_1 \sum x_i^{n+1} + a_2 \sum x_i^{n+2} + \ldots + a_n \sum x_i^{2n} &= \textstyle\sum x_i^n Y_i
\end{aligned} \tag{6}$$

Putting these equations in matrix form shows the coefficient matrix;

$$\begin{bmatrix} N & \sum x_i & \sum x_i^2 & \sum x_i^3 & \ldots & \sum x_i^n \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \ldots & \sum x_i^{n+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \sum x_i^5 & \ldots & \sum x_i^{n+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum x_i^n & \sum x_i^{n+1} & \sum x_i^{n+2} & \sum x_i^{n+3} & \ldots & \sum x_i^{2n} \end{bmatrix} [a] = \begin{bmatrix} \sum Y_i \\ \sum x_i Y_i \\ \sum x_i^2 Y_i \\ \vdots \\ \sum x_i^n Y_i \end{bmatrix} \quad (7)$$

All the summatins in Eqs. 6 and 7 run from 1 to $N$. We will let B stand for the coefficient matrix.

- Equation 7 represents a linear system. However, you need to know that this system is ill-conditioned and round-off errors can distort the solution: the $a$'s of Eq. 5. Up to degree-3 or -4, the problem is not too great. Special methods that use *orthogonal* polynomials are a remedy. Degrees higher than 4 are used very infrequently. It is often better to fit a series of lower-degree polynomials to subsets of the data.

- Matrix $B$ of Eq. 7 is called the *normal matrix* for the least-squares problem. There is another matrix that corresponds to this, called the *design matrix*. It is of the form;

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & \ldots & x_N \\ x_1^2 & x_2^2 & x_3^2 & \ldots & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^n & x_2^n & x_3^n & \ldots & x_N^n \end{bmatrix}$$

$AA^T$ is just the coefficient matrix of Eq. 7. It is easy to see that $Ay$, where $y$ is the column vector of $Y$-values, gives the right-hand side of Eq. 7. We can rewrite Eq. 7 in matrix form, as

$$AA^T a = Ba = Ay$$

- It is illustrated the use of Eqs. 6 to fit a quadratic to the data of Table 1. Figure 7 shows a plot of the data. The data are actually a perturbation of the relation $y = 1 - x + 0.2x^2$. To set up the normal equations, we need the sums tabulated in Table 1. The equations to be solved are:

$$\begin{aligned} 11a_0 + 6.01a_1 + 4.6545a_2 &= 5.905 \\ 6.01a_0 + 4.6545a_1 + 4.1150a_2 &= 2.1839 \\ 4.6545a_0 + 4.1150a_1 + 3.9161a_2 &= 1.3357 \end{aligned}$$

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $x_i$ | 0.05 | 0.11 | 0.15 | 0.31 | 0.46 | 0.52 | 0.70 | 0.74 | 0.82 | 0.98 | 1.171 |
| $Y_i$ | 0.956 | 0.890 | 0.832 | 0.717 | 0.571 | 0.539 | 0.378 | 0.370 | 0.306 | 0.242 | 0.104 |

| | |
|---|---|
| $\Sigma x_i = 6.01$ | $N = 11$ |
| $\Sigma x_i^2 = 4.6545$ | $\Sigma Y_i = 5.905$ |
| $\Sigma x_i^3 = 4.1150$ | $\Sigma x_i Y_i = 2.1839$ |
| $\Sigma x_i^4 = 3.9161$ | $\Sigma x_i^2 Y_i = 1.3357$ |

Table 1: Data to illustrate curve fitting.

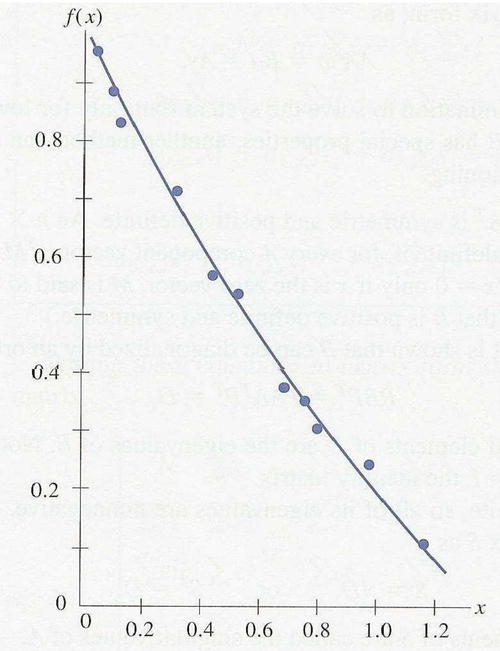


Figure 7: Figure for the data to illustrate curve fitting.

The result is $a_0 = 0.998$, $a_2 = -1.018$,$a_3 = 0.225$, so the least- squares method gives

$$y = 0.998 - 1.018x + 0.225x^2$$

which we compare to $y = 1 - x + 0.2x^2$. Errors in the data cause the equations to differ.

## 2.3 Use of Orthogonal Polynomials

- We have mentioned that the system of normal equations for a polynomial fit is illconditioned when the degree is high. Even for a cubic least-squares polynomial, the condition number of the coefficient matrix can be large.

- In one experiment, a cubic polynomial was fitted to 21 data points. When the data were put into the coefficient matrix of Eq. 7, its condition number (using 2-norms) was found to be 22,000!.

- This means that small differences in the $y$-values will make a large difference in the solution. In fact, if the four right-hand-side values are each changed by only 0.01 (about 0.1%), the solution for the parameters of the cubic were changed significantly, by as much as 44%!

- However, if we fit the data with orthogonal polynomials (A sequence of polynomials is said to be orthogonal with respect to the interval [a,b] if $\int_a^b P_n^*(x)P_m(x)dx = 0 \, when \, n \neq m$) such as the *Chebyshev* polynomials. The condition number of the coefficient matrix is reduced to about 5 and the solution is not much affected by the perturbations.

**Example:** The following data:
R/C: 0.73, 0.78, 0.81, 0.86, 0.875, 0.89, 0.95, 1.02, 1.03, 1.055, 1.135, 1.14, 1.245, 1.32, 1.385, 1.43, 1.445, 1.535, 1.57, 1.63, 1.755;
$V_\theta/V_\infty$: 0.0788, 0.0788, 0.064, 0.0788, 0.0681, 0.0703, 0.0703, 0.0681, 0.0681, 0.079, 0.0575, 0.0681, 0.0575, 0.0511, 0.0575, 0.049, 0.0532, 0.0511, 0.049, 0.0532,0.0426:
Let $x = R/C$ and $y = V_\theta/V_\infty$, We would like our curve to be of the form

$$g(x) = \frac{A}{x}(1 - e^{-\lambda x^2})$$

and our least-squares equation becomes

$$S = \sum_{i=1}^{21}(Y_i - \frac{A}{x_i}(1 - e^{-\lambda x_i^2}))^2$$

Setting $S_\lambda = S_A = 0$ gives the following equations:

$$\sum_{i=1}^{21}(\tfrac{1}{x_i})(1 - e^{-\lambda x_i^2})(Y_i - \tfrac{A}{x_i}(1 - e^{-\lambda x_i^2})) = 0$$
$$\sum_{i=1}^{21} x_i(e^{-\lambda x_i^2})(Y_i - \tfrac{A}{x_i}(1 - e^{-\lambda x_i^2})) = 0$$

When this system of nonlinear equations is solved, we get

$$g(x) = \frac{0.07618}{x}(1 - e^{-2.30574x^2})$$

For these values of $A$ and $\lambda, S = 0.00016$. The graph of this function is presented in Figure 8.

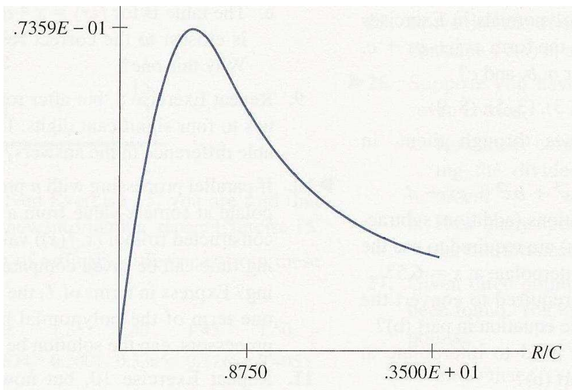


Figure 8: The graph of $V_\theta/V_\infty$ vs $R/C$.

**An algorithm for obtaining a least-squares polynomial:**

Given $N$ data pairs, $(x_i, Y_i), i = 1, \ldots, N$, obtain an nth-degree least-squares polynomial by the following:
Form the coefficient matrix, $M$, with $n+1$ rows (r) and $n+1$ columns (c), by
Set $M_{rc} = \sum_i^N x_i^{r+c-2}$
Form the right-hand-side vector $b$, with $n+1$ rows (r), by:
Set $b_r = \sum_i^N x_i^{r-1} Y_i$
Solve the linear system $Ma = b$ to get the coefficients in
$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$
which is the desired polynomial that fits the data.