Lecture 9

Interpolation and Curve Fitting III

Nonlinear Data, Curve Fitting

Ceng375 *Numerical Computations* at December 21, 2010

Dr. Cem Özdoğan
Computer Engineering Department
Çankaya University

Contents

Nonlinear Data, Curve Fitting I

- 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.

Nonlinear Data, Curve Fitting II

- Thus, the exponential forms are usually **linearized by taking logarithms** before determining the parameters,

For the case $y = ax^b \Longrightarrow$

$$lny = lna + blnx$$

For the case $y = ae^{bx} \Longrightarrow$

$$lny = lna + bx$$

- We now fit the new variable, $z = lny$, as a linear function of lnx or x as described earlier (normal equations).
- 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.

Nonlinear Data, Curve Fitting III

- 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 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.

Nonlinear Data, Curve Fitting IV

- 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

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

Least-Squares Polynomials I

- Fitting polynomials to data that do not plot linearly is common.
- It will turn out that the normal equations are linear for this situation (an added advantage).
- *n* **as the degree of the polynomial**
- **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 $N > n + 1$.
- We assume the functional relationship

$$y = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n \tag{1}$$

- With errors defined by

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

Least-Squares Polynomials II

- We again use Y_i to represent the observed (experimental) value corresponding to x_i (it is assumed that x_i free of error for the sake of simplicity).
- We minimize the sum of squares;

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

- 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:

$$\frac{\partial S}{\partial a_0} = 0 = \sum_{i=1}^{N} 2(Y_i - a_0 - a_1 x_i - a_2 x_i^2 - \ldots - a_i x_i^n)(-1)$$
$$\frac{\partial S}{\partial a_1} = 0 = \sum_{i=1}^{N} 2(Y_i - a_0 - a_1 x_i - a_2 x_i^2 - \ldots - a_i x_i^n)(-x_i)$$
$$\vdots$$
$$\frac{\partial S}{\partial a_n} = 0 = \sum_{i=1}^{N} 2(Y_i - a_0 - a_1 x_i - a_2 x_i^2 - \ldots - a_i x_i^n)(-x_i^n)$$

Least-Squares Polynomials III

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

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

- Putting these equations in matrix form shows the coefficient matrix (B).

$$\overbrace{\begin{bmatrix}
N & \sum x_i & \sum x_i^2 & \sum x_i^3 & \ldots & \sum x_i^n \\
\sum x_i & \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \ldots & \sum x_i^{n+1} \\
\sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \sum x_i^5 & \ldots & \sum x_i^{n+2} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\sum x_i^n & \sum x_i^{n+1} & \sum x_i^{n+2} & \sum x_i^{n+3} & \ldots & \sum x_i^{2n}
\end{bmatrix}}^{B}
\overbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}}^{a}
=
\begin{bmatrix} \sum Y_i \\ \sum x_i Y_i \\ \sum x_i^2 Y_i \\ \vdots \\ \sum x_i^n Y_i \end{bmatrix} \tag{3}$$

All the summations in Eqs. 2 and 3 run from 1 to N.

Least-Squares Polynomials IV

- Equation 3 represents a linear system.
- However, you need to know that if this system is ill-conditioned and round-off errors can distort the solution: the *a*'s of Eq. 1.
- 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. 3 is called the **normal matrix** for the least-squares problem.

Least-Squares Polynomials V

- There is another matrix that corresponds to this, called the **design matrix**.
- It is of the form;

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

- AA^T is just the coefficient matrix of Eq. 3.
- It is easy to see that Ay, where y is the column vector of y-values, gives the right-hand side of Eq. 3.

$$\overbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & \dots & x_N \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^n & x_2^n & x_3^n & \dots & x_N^n \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}}^{y} = \begin{bmatrix} \sum Y_i \\ \sum x_i Y_i \\ \sum x_i^2 Y_i \\ \vdots \\ \sum x_i^n Y_i \end{bmatrix} \quad (4)$$

Least-Squares Polynomials VI

- We can rewrite Eq. 3 in matrix form, as

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

1 $AA^T = B$. To find the solution (with MATLAB)
$>> a = Ay \backslash A * transpose(A)$

$$\overbrace{\begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ x_1 & x_2 & x_3 & \dots & x_N \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^n & x_2^n & x_3^n & \dots & x_N^n \end{bmatrix}}^{A} * \overbrace{\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ 1 & x_3 & x_3^2 & \dots & x_3^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 & \dots & x_N^n \end{bmatrix}}^{A^T} =$$

$$\underbrace{\begin{bmatrix} N & \sum x_i & \sum x_i^2 & \sum x_i^3 & \dots & \sum x_i^n \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \dots & \sum x_i^{n+1} \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \sum x_i^5 & \dots & \sum x_i^{n+2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum x_i^n & \sum x_i^{n+1} & \sum x_i^{n+2} & \sum x_i^{n+3} & \dots & \sum x_i^{2n} \end{bmatrix}}_{B}$$

Least-Squares Polynomials VII

2 $A^T a = y$

$$\overbrace{\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ 1 & x_3 & x_3^2 & \dots & x_3^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_N & x_N^2 & \dots & x_N^n \end{bmatrix}}^{A^T} * \overbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \dots \\ a_n \end{bmatrix}}^{a} = \overbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \dots \\ y_N \end{bmatrix}}^{y}$$

- That is

$$\begin{aligned} a_0 + a_1 x_1 + a_2 x_1^2 + \dots + a_n x_1^n &= y_1 \\ a_0 + a_1 x_2 + a_2 x_2^2 + \dots + a_n x_2^n &= y_2 \\ a_0 + a_1 x_3 + a_2 x_3^2 + \dots + a_n x_3^n &= y_3 \\ \vdots \quad\quad \vdots \quad\quad \vdots \quad\quad \vdots \quad & \vdots \quad \vdots \\ a_0 + a_1 x_N + a_2 x_N^2 + \dots + a_n x_N^n &= y_N \end{aligned}$$

- Least-squares polynomials with all x-values (from given xy-pair data) are inserted.

Least-Squares Polynomials VIII

- It is illustrated the use of Eqs. 2 to fit a quadratic to the data of Table 1.

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: Data to illustrate curve fitting.

- To set up the normal equations, we need the sums tabulated in Table 1. The equations to be solved are:

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

Least-Squares Polynomials IX

- 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.

- Figure 1 shows a plot of the data.
- The data are actually a perturbation of the relation $y = 1 - x + 0.2x^2$.

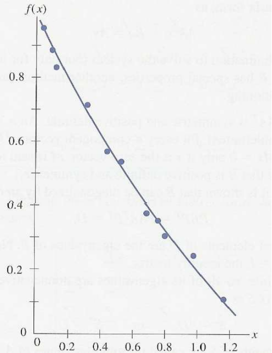

Figure: Figure for the data to illustrate curve fitting.

Least-Squares Polynomials X

- **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_θ/V_∞**:** 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

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

Least-Squares Polynomials XI

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

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

- 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 λ, $S = 0.00016$.
- The graph of this function is presented in Figure 2.

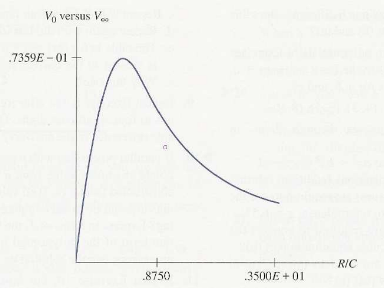

Figure: The graph of V_θ / V_∞ vs R/C.

Use of Orthogonal Polynomials I

- We have mentioned that the system of normal equations for a polynomial fit is ill-conditioned 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. 3, its condition number (using 2-norms) was found to be 22000!.
- This means that small differences in the y-values will make a large difference in the solution.

Use of Orthogonal Polynomials II

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