Lecture 8

Interpolation and Curve Fitting II

Divided Differences,Least-Squares Approximations

Ceng375 *Numerical Computations* at December 9, 2010

Dr. Cem Özdoğan
Computer Engineering Department
Çankaya University

Contents

Interpolation and Curve Fitting II

- **Divided Differences:** These provide a more efficient way to construct an interpolating polynomial, one that allows one to readily change the degree of the polynomiall. If the data are at evenly spaced x-values, there is some simplification.
- **Spline Curves:** Using special polynomials, *splines*, one can fit polynomials to data more accurately than with an interpolating polynomial. At the expense of added computational effort, some important problems that one has with interpolating polynomials is overcome.
- **Least-Squares Approximations:** are methods by which polynomials and other functions can be fitted to data that are subject to errors likely in experiments. These approximations are widely used **to analyze experimental observations**

Divided Differences I

- There are two disadvantages to using the Lagrangian polynomial or Neville's method for interpolation.
  1. It involves more arithmetic operations than does the divided-difference method.
  2. More importantly, if we desire to add or subtract a point from the set used to construct the polynomial, we essentially have to start over in the computations.
- Both the Lagrangian polynomials and Neville's method also must repeat all of the arithmetic if we must interpolate at a new x-value.
- The divided-difference method avoids all of this computation.
- Actually, we will not get a polynomial different from that obtained by Lagrange's technique.

Divided Differences II

- Every n^{th}-degree polynomial that **passes through the same** $n+1$ **points** is <u>identical</u>.
- Only the way that the polynomial is expressed is different.
- The function, $f(x)$, is known at several values for x:

$$\begin{array}{cc} x_0 & f_0 \\ x_1 & f_1 \\ x_2 & f_2 \\ x_3 & f_3 \end{array}$$

- We do not assume that the x's are evenly spaced or even that the values are arranged in any particular order.
- Consider the n^{th}-degree polynomial written as:

$$P_n(x) = a_0 + (x - x_0)a_1 + (x - x_0)(x - x_1)a_2 + (x - x_0)(x - x_1)\ldots(x - x_{n-1})a_n$$

- If we chose the a_i's so that $P_n(x) = f(x)$ at the $n+1$ known points, then $P_n(x)$ is an interpolating polynomial.

Divided Differences III

- The a_i's are readily determined by using what are called the **divided differences of the tabulated values**.
- A special standard notation for divided differences is

$$f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0}$$

called the *first divided difference* between x_0 and x_1.
- And, $f[x_0] = f_0 = f(x_0)$ (zero-order difference).

$$f[x_s] = f_s$$

- In general,

$$f[x_s, x_t] = \frac{f_t - f_s}{x_t - x_s}$$

- Second- and higher-order differences are defined in terms of lower-order differences.

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

Divided Differences IV

- For n-terms,

$$f[x_0, x_1, \ldots, x_n] = \frac{f[x_1, x_2, \ldots, f_n] - f[x_0, x_1, \ldots, f_{n-1}]}{x_n - x_0}$$

- Using the standard notation, a divided-difference table is shown in symbolic form in Table 1.

x_i	f_i	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}]$
x_0	f_0	$f[x_0, x_1]$	$f[x_0, x_1, x_2]$	$f[x_0, x_1, x_2, x_3]$
x_1	f_1	$f[x_1, x_2]$	$f[x_1, x_2, x_3]$	$f[x_1, x_2, x_3, x_4]$
x_2	f_2	$f[x_2, x_3]$	$f[x_2, x_3, x_4]$	
x_3	f_3	$f[x_3, x_4]$		

Table: Divided-difference table in symbolic form.

Divided Differences V

x_i	f_i	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, \ldots, x_{i+3}]$	$f[x_i, \ldots, x_{i+4}]$
3.2	22.0	8.400	2.856	-0.528	0.256
2.7	17.8	2.118	2.012	0.0865	
1.0	14.2	6.342	2.263		
4.8	38.3	16.750			
5.6	51.7				

Table: Divided-difference table in numerical values.

- Table 2 shows specific numerical values.

$$f[x_0, x_1] = \frac{f_1 - f_0}{x_1 - x_0} = \frac{17.8 - 22.0}{2.7 - 3.2} = 8.4$$

$$f[x_1, x_2] = \frac{f_2 - f_1}{x_2 - x_1} = \frac{14.2 - 17.8}{1.0 - 2.7} = 2.1176$$

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{2.1176 - 8.4}{1.0 - 3.2} = 2.8556$$

and the others..

Divided Differences VI

$$
\begin{array}{ll}
x = x_0: & P_0(x_0) = a_0 \\
x = x_1: & P_1(x_1) = a_0 + (x_1 - x_0)a_1 \\
x = x_2: & P_2(x_2) = a_0 + (x_2 - x_0)a_1 + (x_2 - x_0)(x_2 - x_1)a_2 \\
\vdots & \vdots \\
x = x_n: & P_n(x_n) = a_0 + (x_n - x_0)a_1 + (x_n - x_0)(x_n - x_1)a_2 + \ldots \\
 & +(x_n - x_0)\ldots(x_n - x_{n-1})a_n
\end{array}
$$

- If $P_n(x)$ is to be an interpolating polynomial, it must match the table for all $n+1$ entries:

$$P_n(x_i) = f_i \text{ for } i = 0, 1, 2, \ldots, n.$$

- Each $P_n(x_i)$ will equal f_i, if $a_i = f[x_0, x_1, \ldots, x_i]$. We then can write:

$$
\begin{aligned}
P_n(x) = f[x_0] + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] \\
+(x - x_0)(x - x_1)(x - x_2)f[x_0, \ldots, x_3] \\
+(x - x_0)(x - x_1)\ldots(x - x_{n-1})f[x_0, \ldots, x_n]
\end{aligned}
$$

Divided Differences VII

- Write interpolating polynomial of degree-3 that fits the data of Table 2 at all points $x_0 = 3.2$ to $x_3 = 4.8$.

$$P_3(x) = 22.0 + 8.400(x - 3.2) + 2.856(x - 3.2)(x - 2.7)$$

$$-0.528(x - 3.2)(x - 2.7)(x - 1.0)$$

- What is the fourth-degree polynomial that fits at all five points?
- **We only have to add one more term to $P_3(x)$**

$$P_4(x) = P_3(x) + 0.2568(x - 3.2)(x - 2.7)(x - 1.0)(x - 4.8)$$

- If we compute the interpolated value at $x = 3.0$, we get the same result: $P_3(3.0) = 20.2120$.
- This is not surprising, because all third-degree polynomials that pass through the same four points are identical.
- **They may look different but they can all be reduced to the same form.**

Divided Differences VIII

- **Example m-file:** Constructs a table of divided-difference coefficients. Diagonal entries are coefficients of the polynomial. (divDiffTable.m)

```
>> x=[3.2 2.7 1.0 4.8]; y=[22.0 17.8 14.2 38.3];
>> D=divDiffTable(x,y)
D =
   22.0000         0         0         0
   17.8000    8.4000         0         0
   14.2000    2.1176    2.8556         0
   38.3000    6.3421    2.0116   -0.5275
>> c=diag(D);
>> xx=3;
>>p3=c(1)+c(2)*(xx-x(1))+c(3)*(xx-x(1))*(xx-x(2))+
c(4)*(xx-x(1))*(xx-x(2))*(xx-x(3))
p3 =
   20.2120
```

Divided Differences IX

- **Divided differences for a polynomial**
- It is of interest to look at the divided differences for $f(x) = P_n(x)$.
- Suppose that $f(x)$ is the cubic

$$f(x) = 2x^3 - x^2 + x - 1.$$

- Here is its divided-difference table:

x_i	$f[x_i]$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$f[x_i, \ldots, x_{i+3}]$	$f[x_i, \ldots, x_{i+4}]$	$f[x_i, \ldots, x_{i+5}]$
0.30	-0.736	2.480	3.000	2.000	0.000	0.000
1.00	1.000	3.680	3.600	2.000	0.000	
0.70	-0.104	2.240	5.400	2.000		
0.60	-0.328	8.720	8.200			
1.90	11.008	21.020				
2.10	15.212					

- Observe that the third divided differences are all the same.
- It then follows that all higher divided differences will be zero.

Divided Differences X

$$P_3(x) = f[x_0] + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2]$$
$$+(x - x_0)(x - x_1)(x - x_2)f[x_0, x_1, x_2, x_3]$$

```
>> syms x
>> P3=-0.736+(x-0.3)   *2.48+(x-0.3)  *(x-1) *3+(x-0.3)  *(x-1)
                                              *(x-0.7) *2
P3 = -37/25+62/25  *x+3*(x-3/10) *(x-1)+2 *(x-3/10) *(x-1) *(x-7/10)
>> expand(P3)
ans = -1+x-x^2+2  *x^3
```

which is same with the starting polynomial.

Spline Curves I

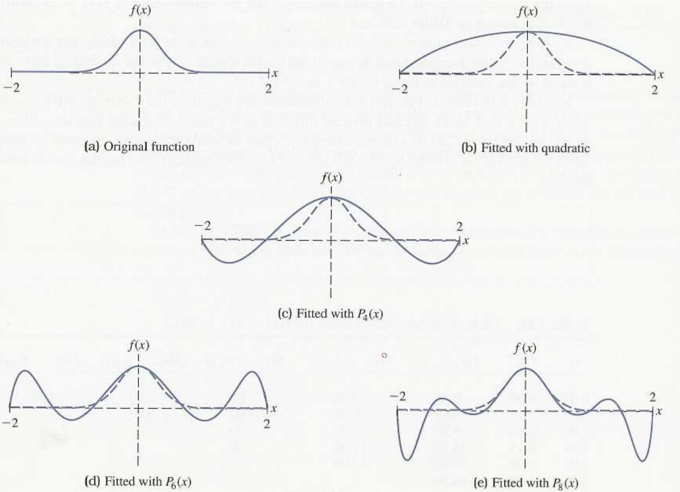

Figure: Fitting with different degrees of the polynomial.

Spline Curves II

- 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.

Spline Curves III

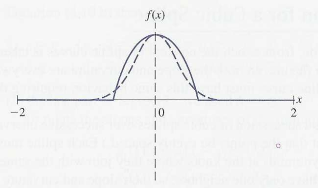

Figure: Fitting with quadratic in subinterval.

- 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.
- This solution is known as **spline curves**.

Spline Curves IV

- 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.

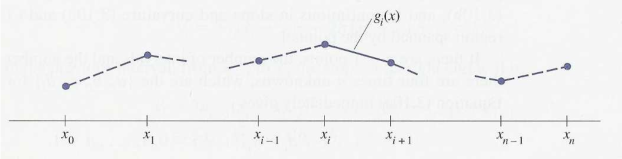

Figure: Linear spline.

- 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.

The Equation for a Cubic Spline I

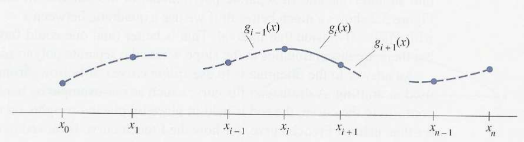

Figure: Cubic spline.

- We will create a succession of cubic splines over successive intervals of the data (See Fig. 4).
- Each spline must join with its neighbouring cubic polynomials at the knots where they join with the **same slope and curvature**.
- We write the equation for a cubic polynomial, $g_i(x)$, in the ith 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:

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

The Equation for a Cubic Spline II

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

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

- and meets these conditions:
  - $$g_i(x_i) = y_i,\ i = 0, 1, \ldots, n-1 \text{ and } g_{n-1}(x_n) = y_n \quad (1)$$
  - $$g_i(x_{i+1}) = g_{i+1}(x_{i+1}),\ i = 0, 1, \ldots, n-2 \quad (2)$$
  - $$g_i'(x_{i+1}) = g_{i+1}'(x_{i+1}),\ i = 0, 1, \ldots, n-2 \quad (3)$$
  - $$g_i''(x_{i+1}) = g_{i+1}''(x_{i+1}),\ i = 0, 1, \ldots, n-2 \quad (4)$$

- 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.

Least-Squares Approximations I

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

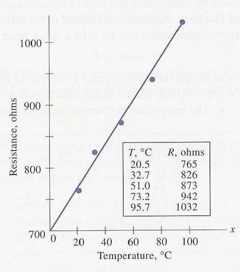

Figure: Resistance vs Temperature graph for the Least-Squares Approximation.

- 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.

Least-Squares Approximations II

- 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.
- 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.

Least-Squares Approximations III

- 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**.

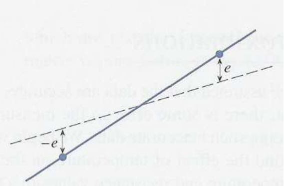

Figure: Minimizing the deviations by making the sum a minimum.

- Consider 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*.

Least-Squares Approximations IV

- We might first suppose we could minimize the deviations by making their sum a minimum, but this is **not an adequate criterion**.
- We might accept the criterion that we make the magnitude of the maximum error a minimum (the so-called *minimax* criterion).
- 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.

Least-Squares Approximations V

- 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 S be a minimum.

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

- N is the number of (x, Y)-pairs.

Least-Squares Approximations VI

- 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 will be zero.

$$\partial S/\partial a \qquad \& \qquad \partial S/\partial b$$

- 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 $i = N$.

Least-Squares Approximations VII

- Solving these equations simultaneously gives the values for slope and intercept a and b.
- For the data in Fig. 5 we find that

$$N = 5, \sum T_i = 273.1, \sum T_i^2 = 18607.27,$$

$$\sum R_i = 4438, \sum T_i R_i = 254932.5$$

- Our *normal equations* are then

$$\begin{aligned} 18607.27a + 273.1b &= 254932.5 \\ 273.1a + 5b &= 4438 \end{aligned}$$

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

$$R = 702.2 + 3.395T$$

Least-Squares Approximations VIII

- 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
```