Interpolation

8.0 Introduction

This chapter concerns polynomial and trigonometric interpolation. Polynomial interpolation leads to piecewise-polynomial reconstruction in Chapter 9 and to numerical differentiation and integration formulae in Chapter 10. As seen in Section 6.3, computers represent functions by sequences of real numbers. In the last chapter, the numbers were coefficients in functional forms such as Legendre polynomial series. In this chapter, the numbers are samples. In particular, consider any set of *sample points* $(x_0, x_1, \ldots, x_N)$ in the domain of a function $f(x)$. Then the *samples* $f_i = f(x_i)$ represent the function $f(x)$. Notice that the total number of samples is $N+1$ since the sample index starts from zero rather than one. The spacing between samples is $\Delta x_i = x_i - x_{i-1}$. If the sample spacing is constant then

$$\Delta x = \frac{x_N - x_0}{N}. \tag{8.1}$$

For example, six samples with $x_5 - x_0 = 5$ have the sample spacing $\Delta x = 1$. Samples are especially popular for solving differential equations, since differential equations are "pointwise" equations (i.e., they describe solutions in terms of rates of change at individual points). *Finite-difference* methods use samples as their primary representation.

Finite-difference methods often switch from samples to functional representations in order to differentiate or to perform other functional operations. Any function created from samples is called a *reconstruction*. Any reconstruction passing through the sample points is called an *interpolation* on the domain $[x_0, x_N]$ and an *extrapolation* on the domains $(-\infty, x_0)$ and (x_N, ∞). This chapter concerns interpolations that yield a single polynomial. The next chapter describes interpolations and other sorts of reconstructions that yield piecewise-polynomials.

The quality of a sampling is gauged by how well the original function can be reconstructed from the samples. Of course, like all finite-length sequences, samples cannot adequately represent just any arbitrary function.

Example 8.1 Suppose that a function has three extrema but only two samples. Then the two samples must inevitably miss important information about the function, as illustrated in Figure 8.1. Specifically, in every case, the two samples lose critical information about the size or location of an extremum. Thus it is impossible to accurately reconstruct the original function from only two samples. This example shows that, at the very least, there should be one sample near every extremum. Not counting the endpoints of the function, the function in this example requires at least three samples.

Example 8.2 Given only samples, and no other information about the function, it may be impossible to accurately reconstruct the original function. For example, any number

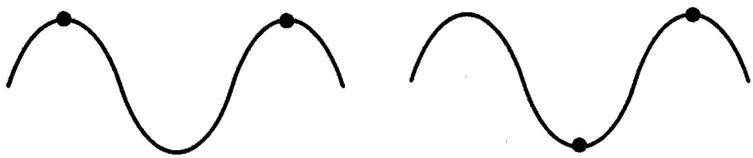

Figure 8.1 Two samples cannot represent three extrema.

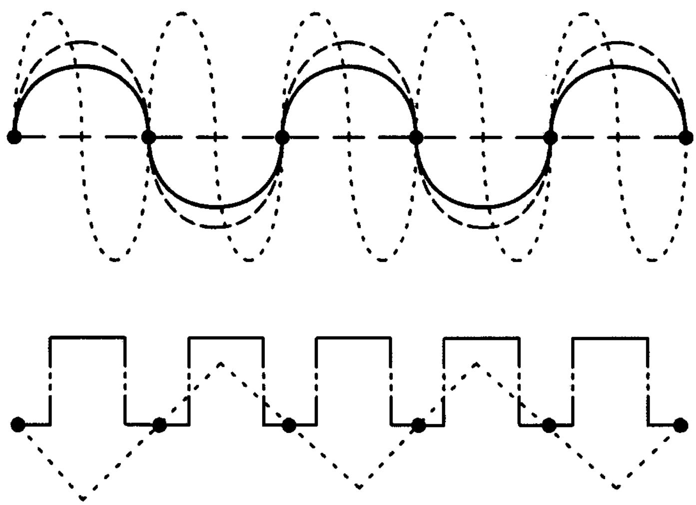

Figure 8.2 Different functions have the same samples.

of functions have uniformly zero samples $f(x_i) = 0$ as illustrated in Figure 8.2. Of course, referring to the previous example, there is absolutely no chance of reconstructing the true function if it is anything other than $f(x) = 0$ – for all of the other possible functions shown, the samples clearly miss critical information about the function.

As seen in these examples, one must know something more about a function other than its samples. In keeping with the spirit of the last chapter, one can restrict the possible functions to a limited finite-dimensional subspace, such that the samples uniquely specify a function from the subspace. In this case, the known subspace constitutes the "additional information" about the function required to make sampling work. As one possibility, suppose we know that the true function is a finite-length polynomial. Then the Nth-order polynomial interpolation equals the true function for large enough N. As another possibility, suppose we know that the true function is bandlimited or, in other words, that the true function equals a finite-length Fourier series. Then an Nth-order trigonometric interpolation equals the true function for large enough N.

8.1 Polynomial Interpolation

There is a unique Nth-order polynomial passing through any set of $N+1$ samples. For example, there is a unique line passing through any two points; there is a unique quadratic passing through any three points; there is a unique cubic passing through any four points; and so on. Depending on where it is evaluated, this polynomial is either an interpolation polynomial or an extrapolation polynomial. The last chapter introduced a number of different polynomial forms, including Taylor series form, Legendre polynomial series form, and Chebyshev polynomial series form. However, these are not necessarily the best forms for expressing interpolation polynomials; this section introduces two new polynomial forms that are well suited for interpolation.

8.1.1 *Lagrange Form*

The *Lagrange form* of a polynomial is defined as follows:

$$p_N(x) = \sum_{i=0}^{N} a_i (x - x_0)(x - x_1) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_{N-1})(x - x_N).$$

Notice that the sum skips the factor $x - x_i$. The Lagrange form coefficients a_i of the interpolation polynomial are found by solving the following linear systems of equations:

$$f(x_i) = a_i (x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_{N-1})(x_i - x_N),$$

where $i = 0, \ldots, N$. The solution to this diagonal system of linear equations is trivial:

$$a_i = \frac{f(x_i)}{(x_i - x_0)(x_i - x_1) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_{N-1})(x_i - x_N)}.$$

Consider the following standard notation:

$$\prod_{i=1}^{N} x_i = x_1 \cdots x_N, \tag{8.2}$$

where the capital pi stands for product. Then the Lagrange form of the interpolation polynomial can be written as

$$p_N(x) = \sum_{i=0}^{N} f(x_i) \prod_{\substack{j=0 \\ j \neq i}}^{N} \frac{x - x_j}{x_i - x_j}. \tag{8.3}$$

The Lagrange form is easy to derive and easy to remember but may be difficult to work with (for example, integration and differentiation are difficult in Lagrange form). The Lagrange form is discussed here mainly for completeness.

Example 8.3 Find the Lagrange form of the interpolation polynomial passing through the following points: $(-1, 1)$, $(0, 2)$, $(3,101)$, $(4, 246)$.

Solution The Lagrange form of the interpolation cubic is

$$p_3(x) = -\frac{1}{20} x(x-3)(x-4) + \frac{1}{6}(x+1)(x-3)(x-4)$$
$$- \frac{101}{12} x(x+1)(x-4) + \frac{123}{10} x(x+1)(x-3).$$

8.1.2 *Newton Form*

The *Newton form* of a polynomial is defined as follows:

$$p_N(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) \\ + a_3(x - x_0)(x - x_1)(x - x_2) + \cdots + a_N(x - x_0)\cdots(x - x_{N-1})$$

or

$$p_N(x) = a_0 + \sum_{i=1}^{N} a_i \prod_{j=0}^{i-1}(x - x_j).$$

The Newton form coefficients a_i of the interpolation polynomial are found by solving the following triangular linear systems of equations:

$$\begin{aligned} f(x_0) &= a_0, \\ f(x_1) &= a_0 + a_1(x_1 - x_0), \\ f(x_2) &= a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1), \\ &\vdots \\ f(x_N) &= a_0 + a_1(x_N - x_0) + a_2(x_N - x_0)(x_N - x_1) \\ &\quad + \cdots + a_N(x_N - x_0)(x_N - x_1)\cdots(x_N - x_{N-1}). \end{aligned}$$

Finding the solutions for a_0 and a_1 is relatively easy. In particular $a_0 = f(x_0)$ and

$$a_1 = \frac{f(x_1) - a_0}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}.$$

Determining the solutions for a_2, a_3, and so on becomes progressively more difficult. Luckily, as it turns out, there is a simple recursive solution. To state this solution, we must first define the *Newton divided differences*. In particular, the *zeroth Newton divided differences* are defined as follows:

$$f[x_i] = f(x_i),$$

where $i = 0, \ldots, N$. The *first Newton divided differences* are then defined as

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i} = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i},$$

where $i = 0, \ldots, N - 1$. The *second Newton divided differences* are defined as

$$\begin{aligned} f[x_i, x_{i+1}, x_{i+2}] &= \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i} \\ &= \frac{1}{x_{i+2} - x_i}\left(\frac{f(x_{i+2}) - f(x_{i+1})}{x_{i+2} - x_{i+1}} - \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}\right), \end{aligned}$$

where $i = 0, \ldots, N - 2$. The *third Newton divided differences* are defined as

$$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}] = \frac{f[x_{i+1}, x_{i+2}, x_{i+3}] - f[x_i, x_{i+1}, x_{i+2}]}{x_{i+3} - x_i},$$

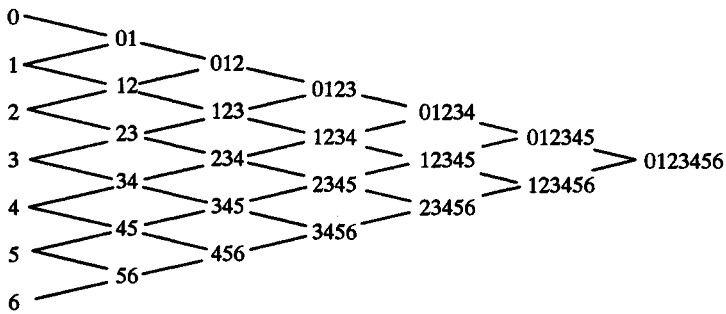

Figure 8.3 Formation of Newton divided differences.

where $i = 0, \ldots, N-3$. In general, the *nth Newton divided differences* are defined as follows:

$$\blacklozenge \qquad f[x_i, \ldots, x_{i+n}] = \frac{f[x_{i+1}, \ldots, x_{i+n}] - f[x_i, \ldots, x_{i+n-1}]}{x_{i+n} - x_i}, \tag{8.4}$$

where $i = 0, \ldots, N-n$. As a way to help remember this definition, notice that the denominator uses the two sample points omitted from one or the other of the divided differences in the numerator. Using this recursive definition, the Newton divided differences form a triangle such as the one for $f[x_0, x_1, x_2, x_3, x_4, x_5, x_6]$ seen in Figure 8.3.

For future reference, it is interesting to note the connection between the Newton divided differences and functional derivatives. If $f(x)$ has n continuous derivatives in $[x_i, x_{i+n}]$ then

$$f[x_i, \ldots, x_{i+n}] = \frac{1}{n!}\frac{d^n f}{dx^n}(\xi) \tag{8.5}$$

for some $x_i < \xi < x_{i+n}$. If the pth derivative of $f(x)$ has a jump discontinuity at z in $[x_i, x_{i+n}]$, but otherwise $f(x)$ is continuously differentiable, then

$$f[x_i, \ldots, x_{i+n}] = O\left(\frac{1}{\Delta x^{n-p}}\left(\frac{d^p f(z_R)}{dx^p} - \frac{d^p f(z_L)}{dx^p}\right)\right) \tag{8.6}$$

for all $n \geq p$, where z_L and z_R refer to the left- and right-hand limits of the pth derivative, respectively. Thus the nth Newton divided difference is proportional to the jump in the pth derivative divided by Δx^{n-p}.

The coefficients of the Newton form of the interpolation polynomial are easily defined in terms of Newton divided differences as follows:

$$a_i = f[x_0, x_1, \ldots, x_i]. \tag{8.7}$$

In other words, the Newton form of the interpolation polynomial is as follows:

$$\blacklozenge \qquad p_N(x) = f[x_0] + \sum_{i=1}^{N} f[x_0, \ldots, x_i] \prod_{j=0}^{i-1} (x - x_j). \tag{8.8}$$

Sensibly enough, the ordering of the sample points does not affect the Newton divided differences nor the interpolation polynomial.

The Newton form of the interpolation polynomial becomes particularly simple if the spacing between the samples is a constant Δx. Suppose $x_{i+1} - x_i = \Delta x = const.$ Then

$$f[x_i, x_{i+1}] = \frac{f(x_{i+1}) - f(x_i)}{\Delta x},$$
$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f(x_{i+2}) - 2f(x_{i+1}) - f(x_i)}{2\Delta x^2},$$
$$f[x_i, x_{i+1}, x_{i+2}, x_{i+3}] = \frac{f(x_{i+3}) - 3f(x_{i+2}) + 3f(x_{i+1}) - f(x_i)}{6\Delta x^3},$$

and so on. Define *zeroth differences* by

$$\Delta_i^0 f = f(x_i),$$

where $i = 0, \ldots, N$, *first differences* by

$$\Delta_i^1 f = \Delta_{i+1}^0 f - \Delta_i^0 f = f(x_{i+1}) - f(x_i),$$

where $i = 0, \ldots, N-1$, *second differences* by

$$\Delta_i^2 f = \Delta_{i+1}^1 f - \Delta_i^1 f = f(x_{i+2}) - 2f(x_{i+1}) + f(x_i),$$

where $i = 0, \ldots, N-2$, *third differences* by

$$\Delta_i^3 f = \Delta_{i+1}^2 f - \Delta_i^2 f = f(x_{i+3}) - 3f(x_{i+2}) + 3f(x_{i+1}) - f(x_i),$$

where $i = 0, \ldots, N-3$, and *fourth differences* by

$$\Delta_i^4 f = \Delta_{i+1}^3 f - \Delta_i^3 f = f(x_{i+4}) - 4f(x_{i+3}) + 6f(x_{i+2}) - 4f(x_{i+1}) + f(x_i),$$

where $i = 0, \ldots, N-4$. In general, we can define *nth differences* as follows:

$$\Delta_i^n f = \Delta_{i+1}^{n-1} f - \Delta_i^{n-1} f, \tag{8.9}$$

where $i = 0, \ldots, N-n$. Then

$$f[x_i, \ldots, x_{i+n}] = \frac{\Delta_i^n f}{n!\Delta x^n} \tag{8.10}$$

and the Newton form of the interpolation polynomial is

$$\begin{aligned} p_N(x) = f(x_0) &+ \frac{\Delta_0^1 f}{\Delta x}(x - x_0) + \frac{1}{2}\frac{\Delta_0^2 f}{\Delta x^2}(x - x_0)(x - x_1) \\ &+ \frac{1}{6}\frac{\Delta_0^3 f}{\Delta x^3}(x - x_0)(x - x_1)(x - x_2) \\ &+ \cdots + \frac{1}{N!}\frac{\Delta_0^N f}{\Delta x^N}(x - x_0)(x - x_1)\cdots(x - x_{N-1}), \end{aligned}$$

which, not coincidentally, looks quite a bit like a Taylor series.

Example 8.4 Find the Newton form of the interpolation polynomial passing through the following points: $(-1, 1)$, $(0, 2)$, $(3, 101)$, $(4, 246)$.

Solution The first divided differences are

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{2-1}{0-(-1)} = 1,$$

$$f[x_1, x_2] = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{101-2}{3-0} = 33,$$

$$f[x_2, x_3] = \frac{f(x_3) - f(x_2)}{x_3 - x_2} = \frac{246-101}{4-3} = 145.$$

The second divided differences are

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{33-1}{3-(-1)} = 8,$$

$$f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1} = \frac{145-33}{4-0} = 28.$$

The third divided difference is

$$f[x_0, x_1, x_2, x_3] = \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} = \frac{28-8}{4-(-1)} = 4.$$

Then the Newton form of the interpolation polynomial is

$$p_3(x) = 1 + (x+1) + 8x(x+1) + 4x(x+1)(x-3).$$

Although it may look different, this polynomial is the same as the one found in Example 8.3.

8.1.3 *Taylor Series Form*

Although better than the Lagrange form, the Newton form is still awkward for many functional operations including integration and differentiation. This section introduces the Taylor series form of the interpolation polynomial. The Taylor series form is the hardest to create, costing three times as much to create as the Newton form, but it is the easiest to differentiate or integrate. The Taylor series form of a polynomial is

$$p_N(x) = a_0 + a_1(x-b) + \cdots + a_N(x-b)^N.$$

Then the Taylor coefficients a_i of the interpolation polynomial are found by solving the following linear systems of equations:

$$f(x_i) = a_0 + a_1(x_i - b) + a_2(x_i - b)^2 + \cdots + a_N(x_i - b)^N,$$

where $i = 0, \ldots, N$. The solution involves coefficients d_{ij} defined as follows:

$$d_{0j}(b) = 1$$

for $j = 0, \ldots, N$. Also,

$$d_{i0}(b) = (b - x_{i-1})d_{i-1,0}(b)$$

for $i = 0, \ldots, N$. In general,

♦ $$d_{ij}(b) = d_{i,j-1}(b) + (b - x_{i+j-1})d_{i-1,j}(b) \tag{8.11}$$

for $i=1,\ldots,N$ and $j=1,\ldots,N$. The coefficients d_{ij} may be considered as elements of an $(N+1)\times(N+1)$ matrix D. The only caution is that the elements of D are indexed starting from 0 rather than 1. The elements in the zeroth row of D are all 1 whereas the elements in the zeroth column of D are $1, b-x_0, (b-x_0)(b-x_1), (b-x_0)(b-x_1)(b-x_2)$, and so on. In general, d_{ij} is formed by adding the element $d_{i,j-1}$ on the left and $(b-x_{i+j-1})$ times the element $d_{i-1,j}$ above.

To get a further sense of d_{ij}, consider the following: *$d_{ij}(b)$ is the sum of all possible products of i distinct factors chosen from the set of $i+j$ elements* $\{(b-x_0),(b-x_1),\ldots,(b-x_{i+j-1})\}$. Although elegant, this way of thinking about d_{ij} should never be used computationally for large N, since it would be extremely expensive compared to the recursive definition of Equation (8.11).

The coefficients of the Taylor series form of the interpolation polynomial can be written as follows:

$$\blacklozenge \qquad a_j = \sum_{i=0}^{N-j} d_{ij}(b) f[x_0,\ldots,x_{i+j}]. \tag{8.12}$$

Recall that the *diagonal* of a matrix runs from the upper left to the lower right. Similarly, the *antidiagonal* of a matrix runs from the lower right to the upper left. Thinking in terms of matrices, only the elements of D above the antidiagonal are required to form a_j – the elements below the antidiagonal can either be ignored or considered to be zero.

Example 8.5 Find a general expression for the Taylor series form of the interpolation polynomial passing through three samples $(x_0, f(x_0)), (x_1, f(x_1)), (x_2, f(x_2))$.

Solution In this case, $N=2$ and D is the following 3×3 matrix:

$$D = \begin{bmatrix} d_{00}=1 & d_{01}=1 & d_{02}=1 \\ d_{10}=b-x_0 & d_{11}=b-x_0+b-x_1 & 0 \\ d_{20}=(b-x_0)(b-x_1) & 0 & 0 \end{bmatrix}.$$

For example, d_{11} is formed as follows:

$$d_{11} = d_{10} + (b-x_1)d_{01} = b - x_0 + (b-x_1)\times 1.$$

Then a general Taylor series form of the quadratic interpolation polynomial is

$$p_2(x) = a_2(x-b)^2 + a_1(x-b) + a_0,$$

where $a_2 = f[x_0,x_1,x_2]$,

$$a_1 = f[x_0,x_1] + (b-x_0+b-x_1) f[x_0,x_1,x_2],$$

and

$$a_0 = f[x_0] + (b-x_0) f[x_0,x_1] + (b-x_0)(b-x_1) f[x_0,x_1,x_2].$$

Example 8.6 For $N=3$, show that a general expression for D is

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ b-x_0 & b-x_0+b-x_1 & \begin{array}{c} b-x_0+b \\ -x_1+b-x_2 \end{array} & 0 \\ (b-x_0)(b-x_1) & \begin{array}{c} (b-x_0)(b-x_1) \\ +(b-x_0)(b-x_2) \\ +(b-x_1)(b-x_2) \end{array} & 0 & 0 \\ (b-x_0)(b-x_1)(b-x_2) & 0 & 0 & 0 \end{bmatrix}.$$

Solution The element d_{21} is formed as follows:

$$\begin{aligned} d_{21} &= d_{20}+(b-x_2)\,d_{11} \\ &= (b-x_0)(b-x_1)+(b-x_2)(b-x_0+b-x_1). \end{aligned}$$

Notice that d_{21} is the sum of all possible products of two distinct factors chosen from the set $\{(b-x_0),(b-x_1),(b-x_2)\}$. The other elements are formed similarly.

Example 8.7 Find a general expression for a third-order interpolation polynomial in Taylor series form.

Solution Any third-order interpolation polynomial can be written as follows:

$$p_3(x)=a_3(x-b)^3+a_2(x-b)^2+a_1(x-b)+a_0.$$

Using the matrix D from the previous example, the zeroth column of D yields

$$\begin{aligned} a_0 = {} & f[x_0]+(b-x_0)f[x_0,x_1]+(b-x_0)(b-x_1)f[x_0,x_1,x_2] \\ & +(b-x_0)(b-x_1)(b-x_2)f[x_0,x_1,x_2,x_3]. \end{aligned}$$

The first column of D yields

$$\begin{aligned} a_1 = {} & f[x_0,x_1]+(b-x_0+b-x_1)f[x_0,x_1,x_2] \\ & +[(b-x_0)(b-x_1)+(b-x_0)(b-x_2) \\ & +(b-x_1)(b-x_2)]f[x_0,x_1,x_2,x_3]. \end{aligned}$$

The second column of D yields

$$a_2=f[x_0,x_1,x_2]+(b-x_0+b-x_1+b-x_2)f[x_0,x_1,x_2,x_3].$$

Finally, the third column of D yields $a_3=f[x_0,x_1,x_2,x_3]$.

Example 8.8 Find the Taylor series form of the interpolation polynomial with $b=0$ passing through the following points:

$$(-1,1), \quad (0,2), \quad (3,101), \quad (4,246).$$

Solution From Example 8.4 recall that $f[x_0,x_1]=1$, $f[x_0,x_1,x_2]=8$, and $f[x_0,x_1,x_2,x_3]=4$. Consider the expression for D given in Example 8.6. If $b=0$,

$x_0 = -1$, $x_1 = 0$, $x_2 = 3$, and $x_3 = 4$ then

$$D = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -2 & 0 \\ 0 & -3 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The coefficients of the Taylor series form polynomial are

$$\begin{aligned} a_0 &= d_{00} f(x_0) + d_{10} f[x_0, x_1] + d_{20} f[x_0, x_1, x_2] + d_{30} f[x_0, x_1, x_2, x_3] \\ &= (1 \times 1) + (1 \times 1) + (0 \times 8) + (0 \times 4) = 2, \\ a_1 &= d_{01} f[x_0, x_1] + d_{11} f[x_0, x_1, x_2] + d_{21} f[x_0, x_2, x_3] \\ &= (1 \times 1) + (1 \times 8) - (3 \times 4) = -3, \\ a_2 &= d_{02} f[x_0, x_1, x_2] + d_{12} f[x_0, x_1, x_2, x_3] \\ &= (1 \times 8) - (2 \times 4) = 0, \end{aligned}$$

and

$$a_3 = d_{03} f[x_0, x_1, x_2, x_3] = 1 \times 4 = 4.$$

The Taylor series form of the interpolation polynomial is then

$$p_3(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 = 2 - 3x + 4x^3.$$

Having worked this example three times now, in three different forms, we see that this is certainly the simplest form of the three. However, in general, the Taylor series form is *not* always the simplest form.

Example 8.9 Find a general expression for D when $N = 4$.

Solution The zeroth column of D is

$$\begin{bmatrix} 1 \\ b - x_0 \\ (b - x_0)(b - x_1) \\ (b - x_0)(b - x_1)(b - x_2) \\ (b - x_0)(b - x_1)(b - x_2)(b - x_3) \end{bmatrix}.$$

This column leads to the expression for a_0. The first column of D is

$$\begin{bmatrix} 1 \\ b - x_0 + b - x_1 \\ (b - x_0)(b - x_1) + (b - x_0)(b - x_2) + (b - x_1)(b - x_2) \\ (b - x_0)(b - x_1)(b - x_2) \; + \; (b - x_0)(b - x_1)(b - x_3) \\ + (b - x_0)(b - x_2)(b - x_3) + (b - x_1)(b - x_2)(b - x_3) \\ 0 \end{bmatrix}.$$

This column leads to the expression for a_1. The second column of D is

$$\begin{bmatrix} 1 \\ b-x_0+b-x_1+b-x_2 \\ (b-x_0)(b-x_1)+(b-x_0)(b-x_2) \\ +(b-x_1)(b-x_2)+(b-x_0)(b-x_3) \\ +(b-x_1)(b-x_3)+(b-x_2)(b-x_3) \\ 0 \\ 0 \end{bmatrix}.$$

This column leads to the expression for a_2. The third column of D is

$$\begin{bmatrix} 1 \\ b-x_0+b-x_1+b-x_2+b-x_3 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

This column leads to the expression for a_3. The fourth and last column of D has 1 as its first element and 0 everywhere else. This column leads to the expression for a_4, namely $a_4 = f[x_0, x_1, x_2, x_3, x_4]$.

8.1.4 *Accuracy of Polynomial Interpolation*

Most readers will be familiar with Taylor's remainder theorem for Taylor series. There is a similar result for polynomial interpolation. In particular, let the $N+1$ interpolation points be $(x_0, f(x_0)), \ldots, (x_N, f(x_N))$. Suppose $f(x)$ has $N+1$ continuous derivatives. Then the absolute error $e_N(x) = f(x) - p_N(x)$ of the interpolation polynomial $p_N(x)$ is

$$e_N(x) = (x-x_0)(x-x_1)\cdots(x-x_N)\frac{1}{(N+1)!}\frac{d^{N+1}f(\xi)}{dx^{N+1}} \tag{8.13}$$

for all $\min x_i \leq x \leq \max x_i$ and where $\xi = \xi(x)$ is some number $\min x_i \leq \xi \leq \max x_i$. Notice that Equation (8.13) looks like the next term in the Newton form of the interpolation polynomial, except that the Newton divided difference is replaced by a derivative, as in Equation (8.5).

Suppose that $f(x)$ has infinitely many continuous derivatives. According to Equation (8.13), the error will decrease as N increases if the $(N+1)$th derivative of $f(x)$ grows slower than $1/(N+1)!$ times $(x-x_0)\cdots(x-x_N)$. Then the interpolation polynomial $p_N(x)$ converges to the exact function $f(x)$ as $N \to \infty$. However, according to Equation (8.13), the error may increase as N increases if the $(N+1)$th derivative of $f(x)$ grows faster than $1/(N+1)!$ times $(x-x_0)\cdots(x-x_N)$. Then the interpolation polynomial $p_N(x)$ may diverge from $f(x)$ as $N \to \infty$. If $f(x)$ does not have infinitely many continuous derivatives, then Equation (8.13) does not apply. In this case, the error increases rapidly as N increases, at least for large enough N, and the interpolation polynomial $p_N(x)$ diverges from $f(x)$ as $N \to \infty$.

Large polynomial interpolation errors, including those that prevent convergence, tend to take the form of spurious oscillations called the *Runge phenomenon*. The Runge phenomenon is similar to the Gibbs oscillations found in Lagrange, Chebyshev, and Fourier series, as seen in the last chapter. However, unlike Gibbs oscillations, the Runge phenomenon can occur even when the function is completely smooth. The period and amplitude of Gibbs oscillations decrease everywhere as N increases except for the one oscillation next to each jump discontinuity, whose amplitude remains steady for large N. By contrast, the amplitude of Runge oscillations can increase throughout the domain as N increases, even for perfectly smooth functions.

Example 8.10 Consider the function

$$f(x) = -\sin \pi x.$$

Suppose the function is sampled using $N+1$ evenly spaced samples on the domain $[-1, 1]$. Find the interpolation polynomials $p_N(x)$ for $N = 1, \ldots, 6$.

Solution The samples are $x_i = -1 + 2i/N$. Then $p_1(x) = p_2(x) = 0$,

$$p_3(x) = 2.923x(x^2 - 1),$$
$$p_4(x) = 2.667x(x^2 - 1),$$
$$p_5(x) = -3.134x + 4.962x^3 - 1.827x^5,$$

and

$$p_6(x) = -3.118x + 4.871x^3 - 1.754x^5.$$

In this case, the function is extremely smooth and thus the interpolation is extremely successful.

Example 8.11 Consider the following square-wave function:

$$f(x) = \begin{cases} 0 & -1 \leq x < -1/3, \\ 1 & -1/3 \leq x \leq 1/3, \\ 0 & 1/3 < x \leq 1. \end{cases}$$

Suppose the function is sampled using $N+1$ evenly spaced samples on the domain $[-1, 1]$. Find the interpolation polynomials $p_N(x)$ for $N = 1, \ldots, 5$ and $N = 20$.

Solution The first five interpolation polynomials are

$$p_1(x) = 0,$$
$$p_2(x) = 1 - x^2,$$
$$p_3(x) = 1.125(1 - x^2),$$
$$p_4(x) = 4x^4 - 5x^2 + 1,$$

and

$$p_5(x) = 3.255x^4 - 4.427x^2 + 1.172.$$

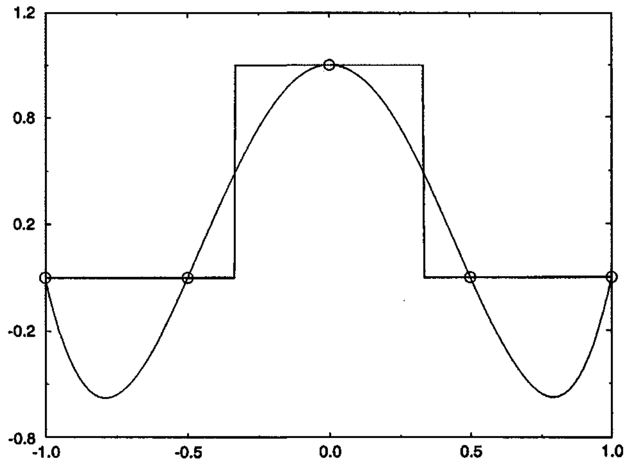

Figure 8.4 Fourth-order polynomial interpolation for a square wave.

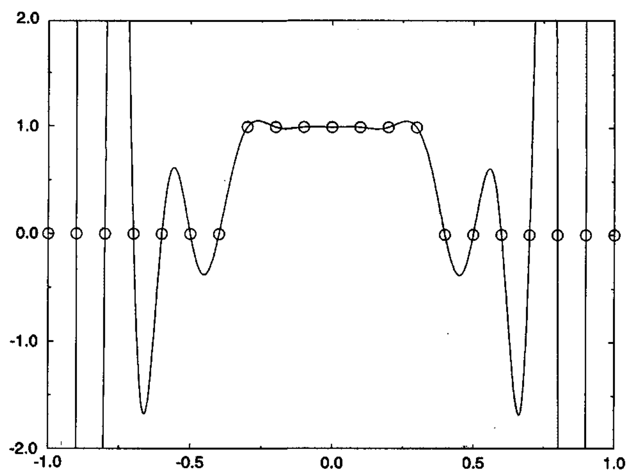

Figure 8.5 Twentieth-order polynomial interpolation for a square wave.

The interpolation polynomials $p_4(x)$ and $p_{20}(x)$ are plotted in Figures 8.4 and 8.5, respectively. The Runge oscillations in this example are incredibly severe. While the very center of the square wave improves as N increases, the edges blow up. This example shows that higher-order interpolation should be used cautiously, if at all, in the presence of jump discontinuities. Comparing Examples 7.3 and 7.5, we see that interpolation polynomials can be far worse than Legendre or Chebyshev polynomials. In other words, the Nth-order interpolation polynomial is much worse than the best Nth-order polynomial as measured in the 2-norm, the ∞-norm, or any other norm. In fact, in this case, even the best Nth-order polynomial is not especially good, and the interpolation polynomial does not even come close to the best. The interpolation polynomial gets worse as N increases whereas Legendre and Chebyshev polynomial series get better.

8.1.5 *Summary of Polynomial Interpolation*

The properties of polynomial interpolation are summarized as follows:

- Error tends to decrease as N increases unless $|d^{N+1}f/dx^{N+1}|$ increases rapidly with N or unless $d^{N+1}f/dx^{N+1}$ is discontinuous. If $|d^{N+1}f/dx^{N+1}|$ is reasonably small for all N, the error decreases as N^{-N}.
- Error tends to take the form of spurious oscillations, called the Runge phenomenon. The error tends to be greatest near the edges of the interpolation domain and least near the center. The Runge phenomenon often grows with N. As a result, most experts advise against using interpolations with orders greater than four. This rule of thumb has proven reliable both in general and in the specific context of computational gasdynamics.
- Large and discontinuous derivatives tend to cause the most error when located near the center of the interpolation domain and the least error when located near the edges of the domain.
- On a domain $[-1, 1]$, the 2-norm interpolation error is nearly minimized by choosing the samples x_i to be the roots of the Legendre polynomial $P_{N+1}(x)$ defined in Section 7.2. Although this sampling nearly minimizes the 2-norm error among interpolation polynomials, the error may still be far worse than in a Legendre polynomial series.
- On a domain $[-1, 1]$, the maximum interpolation error is nearly minimized by choosing the samples x_i to be the roots of a Chebyshev polynomial $T_{N+1}(x)$, as given by Equation (7.20).
- Error generally increases extremely rapidly outside of the interpolation domain $[a, b]$. Hence even when interpolation error is small, extrapolation error may be extremely large.

8.2 Trigonometric Interpolation and the Nyquist Sampling Theorem

Most numerical methods studied in this book use polynomial interpolation rather than trigonometric interpolation. Even still, trigonometric interpolation sheds an interesting light on the limitations of sampling, which justifies a brief consideration here. Assume that a reconstruction has the same form as a Fourier series. Given $2N + 1$ samples, the $2N + 1$ coefficients are determined by the following system of equations:

$$f(x_i) = a_0 + \sum_{n=1}^{N} a_n \cos\left(2\pi n \frac{x_i - a}{b - a}\right) + \sum_{n=1}^{N} b_n \sin\left(2\pi n \frac{x_i - a}{b - a}\right)$$

for $i = 0, \ldots, 2N$. This system appears to have $2N + 1$ equations in the $2N + 1$ unknowns a_n and b_n. Unfortunately, relations such as

$$\sin\theta = \cos(\theta - \pi/2) = -\sin(\theta - \pi) = -\cos(\theta - 3\pi/2)$$

imply that as many as half of the equations may be duplicates. Hence, despite appearances, there are generally more unknowns than equations, and thus the system yields *infinitely* many solutions for the Fourier series form coefficients. There are two ways to deal with this situation. One approach is to roughly double the number of samples, from $2N + 1$ to $4N + 1$. Indeed, $4N + 1$ or more samples always yield a unique solution for the coefficients

a_n and b_n. Alternatively, given any number of samples $M + 1$ less than $4N + 1$, find the coefficients a_n and b_n that minimize the 2-norm error of the $2N$-order trigonometric interpolation. As it turns out, both approaches yield the same expressions for the coefficients a_n and b_n. In particular, if $f(x)$ has domain $[0, 2\pi]$ then both approaches yield

$$f(x) = a_0 + \sum_{n=1}^{N} a_n \cos(nx) + \sum_{n=1}^{N} b_n \sin(nx), \tag{8.14}$$

where

$$a_0 = \frac{1}{M+1} \sum_{i=0}^{M} f(x_i), \tag{8.15}$$

$$a_n = \frac{2}{M+1} \sum_{i=0}^{M} f(x_i) \cos(nx_i), \tag{8.16}$$

and

$$b_n = \frac{2}{M+1} \sum_{i=0}^{M} f(x_i) \sin(nx_i), \tag{8.17}$$

and where $M+1$ is the number of samples and $2N$ is the order of the trigonometric series. Even though the trigonometric polynomial is written in Fourier series form, the trigonometric interpolation equals the true Fourier series only if $M \geq 4N$.

Even when the trigonometric interpolation equals the "best case" Fourier series, it may still exhibit large Gibbs oscillations, as discussed in Section 7.4. But suppose we know that the true function $f(x)$ is bandlimited, as defined in Section 7.4. In other words, suppose the true function $f(x)$ exactly equals its Fourier series:

$$f(x) = a_0 + \sum_{n=1}^{N} a_n \cos\left(2\pi n \frac{x-a}{b-a}\right) + \sum_{n=1}^{N} b_n \sin\left(2\pi n \frac{x-a}{b-a}\right) \tag{8.18}$$

for some N. Then the unique $2N$th-order trigonometric interpolation passing through any $4N + 1$ samples equals the true function. Let us rewrite this result in terms of the sample spacing Δx instead of N. Assume that $x_0 = a$ and $x_N = b$. Then, substituting $4N$ for N in Equation (8.1), we get $\Delta x = (b - a)/4N$. Notice that the shortest wavelength in Equation (8.18) is $(b - a)/N$, that is, the shortest wavelength in Equation (8.18) equals $4\Delta x$. Then *samples spaced apart by* Δx *perfectly represent functions whose shortest wavelengths are* $4\Delta x$. This is called the *Nyquist sampling theorem.*

Example 8.12 If $f(x_i) = 0$ and the function is bandlimited according to the Nyquist sampling theorem, then the function must be $f(x) = 0$. This is the natural choice, as seen in Example 8.2.

Example 8.13 The shortest wavelength that can be captured by sampling with spacing Δx is $2\Delta x$. In particular, for $2\Delta x$-oscillations, also known as *odd–even oscillations*, every sample is alternately a maximum and a minimum. However, the Nyquist sampling theorem says that $4\Delta x$ is the shortest wavelength that can be *accurately* captured. In particular, consider the samplings shown in Figure 8.6; by the Nyquist sampling theorem, only the

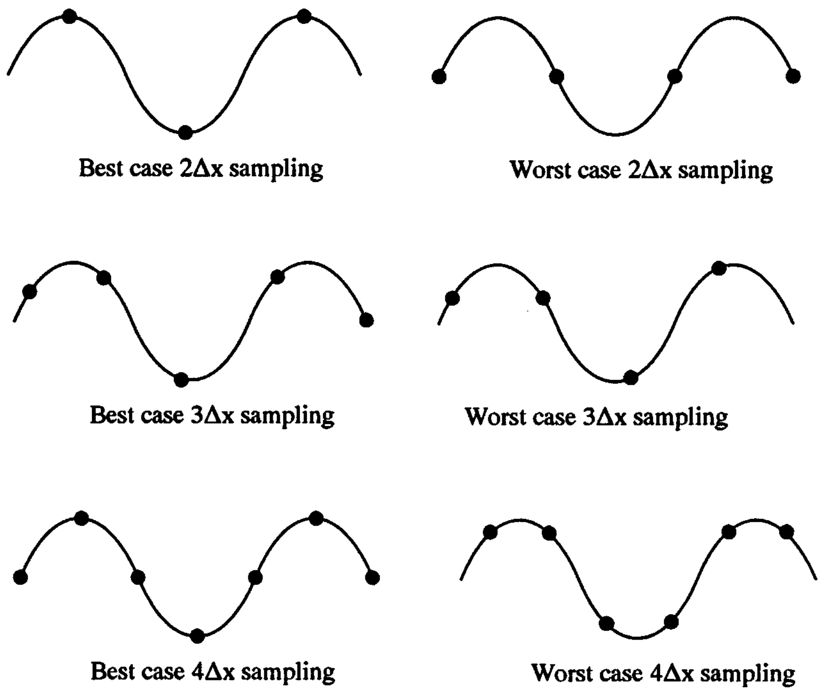

Figure 8.6 An illustration of the Nyquist sampling theorem.

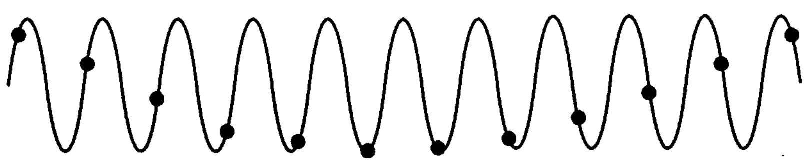

Figure 8.7 An illustration of aliasing.

$4\Delta x$-samplings always adequately represent the sinusoid. Intuitively, $4\Delta x$-samplings always place a sample at or near each extremum, whereas $2\Delta x$- and $3\Delta x$-samplings may or may not. Thus, among other things, the Nyquist sampling theorem requires that there be at least one sample near every extremum, so that the sampled local maximum is only slightly less than the true local maximum and the sampled local minimum is only slightly more than the true local minimum.

Example 8.14 When the Nyquist sampling theorem is violated, short-wavelength components in the original function may appear to be long-wavelength components in the sampling – this is known as *aliasing*. For example, the worst-case $2\Delta x$ sampling seen in Figure 8.6 represents a short-wavelength component by a very long wavelength; in fact, it turns a sinusoid into a constant, and a constant has an *infinite* wavelength! As another example, consider the sampling shown in Figure 8.7. The samples still vary sinusoidally, like the original function, but the samples have a much longer wavelength.

Strictly speaking, the Nyquist sampling theorem and the conclusions drawn therefrom, especially regarding the dangers of $2\Delta x$-oscillations, apply only to trigonometric interpolation. However, without too much effort, you can convince yourself that $2\Delta x$ "odd–even" oscillations pose threats to other sorts of interpolation, including polynomial interpolations. In particular, if a sampling misses the peaks of the $2\Delta x$-oscillations, then no sort of interpolation can hope to accurately recover the lost information about the oscillations. However, if the sampling does capture the peaks of the $2\Delta x$-oscillations, the resulting rapid oscillations in the samples will tend to spur large-amplitude spurious Runge oscillations in interpolation polynomial and other types of interpolations. In short, you should probably avoid oscillations with periods of less than $4\Delta x$, regardless of whether you intend to use trigonometric, polynomial, or some other sort of interpolation. For more on the Nyquist sampling theorem, see Hamming (1973) or any number of electrical engineering books on digital signal processing.

References

Hamming, R. W. 1973. *Numerical Methods for Scientists and Engineers*, 2nd ed., New York: Dover, Chapters 14, 31, and 32.

Mathews, J. H. 1992. *Numerical Methods for Mathematics, Science, and Engineering*, 2nd ed., Englewood Cliffs, NJ: Prentice-Hall, Sections 4.2–4.5, 5.4, 6.2.

Problems

8.1 Suppose $f[-5, -1, 0, 3, 5] = 24$, $f[0, 3, 5, 6] = 12$, and $f[-1, 0, 3, 5] = 15$. Find $f[-5, -1, 0, 3, 5, 6]$.

8.2 Consider the samples (−5, 25), (0, 5), (1, 10), (2, 20). The Taylor series form of the interpolation polynomial requires a matrix D. Write the matrix D for $b = 0$.

8.3 According to Equation (8.13), which of the following functions are subject to substantial Runge oscillations when sampled and interpolated? Explain briefly in each case.
(a) $\sin^2(\pi x)$ (b) $1/(1 + 10x^2)$
(c) e^{-8x^2} (d) $\ln 5x^2$

8.4 Consider $f(x) = 3\sin^2(\pi x/6)$. Sample the function at $x = 0, 1, 2$, and 4. Show all work (this is not a programming exercise).
(a) Find the interpolation polynomial in Lagrange form.
(b) Find the Newton divided differences of the samples.
(c) Find the interpolation polynomial in Newton form.
(d) For $b = 3$, find the coefficient matrix D.
(e) Find the interpolation polynomial in the form of a Taylor series about $b = 3$.

8.5 Consider the following samples: $f(-1) = 0$, $f(-0.2) = 0.1$, $f(0) = 1$, $f(0.2) = 0.1$, and $f(1) = 0$. Show all work (this is not a programming exercise).
(a) Find the interpolation polynomial in Lagrange form.
(b) Find the Newton divided differences of the samples.
(c) Find the interpolation polynomial in Newton form.
(d) For $b = 0$, find the coefficient matrix D.
(e) Find the interpolation polynomial in the form of a Taylor series about $b = 0$.

8.6 Draw at least five different functions through the samples shown below. The functions should represent a variety of behaviors. In particular, at least one of the functions should be

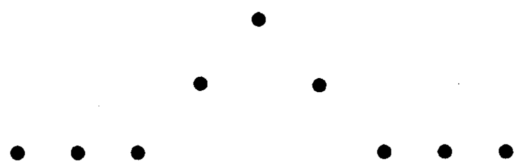

Figure 8.8 Samples for Problem 8.6.

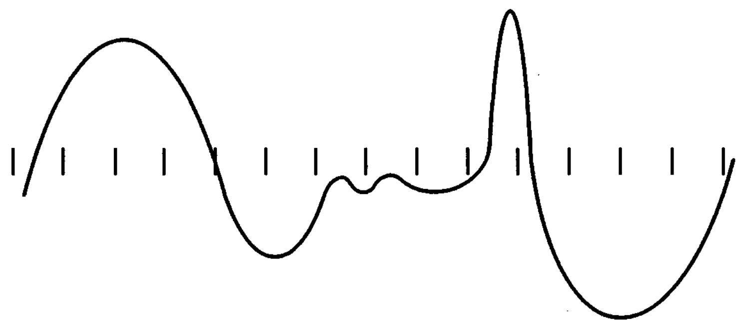

Figure 8.9 Function for Problem 8.7.

nonoscillatory. Will any of the functions passing through the samples satisfy the Nyquist sampling theorem?

8.7 Consider the function shown in the figure.

(a) Using the reasoning of Example 8.1, what is the minimum order of a polynomial that could adequately represent this function?

(b) Would you expect this function to be bandlimited? If so, roughly, what is the minimum wavelength in the trigonometric series that could adequately represent this function? If not, what features of the function create the unlimited short wavelengths?

(c) If the samples must be evenly spaced, what is the minimum number of samples required, according to the Nyquist sampling theorem?

(d) Suppose you were allowed uneven sample spacings, and you wanted to make a rough sketch of the function. Argue that the minimum number of samples required for an adequate rough sketch of the function equals one sample for each maximum and minimum, assuming that the samples are placed exactly on the maxima and minima.

8.8 Consider the following function:

$$f(x) = \sin x(\sin x + \cos x + \sin 3x + \cos 7x).$$

(a) This function is periodic. What is its period?

(b) Using trigonometric identities, write this in the form of a Fourier series over one period.

(c) What is the minimum number of samples required to represent one period of this function, according to the Nyquist sampling theorem?