Interpolation and Polynomial Approximation

Professor Henry Arguello

Universidad Industrial de Santander
Colombia

March 12, 2018

High Dimensional Signal Processing Group
www.hdspgroup.com
henarfu@uis.edu.co

Universidad Industrial de Santander

Outline: Chapter 4

Interpolation is used to approximate different functions as sin(x), cos(x), etc.. using polynomials.

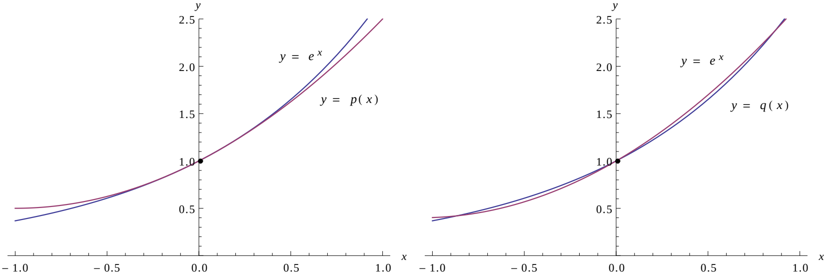


Figure: **Comparison between 2 approximations** (a) The Taylor polynomial $p(x) = 1.000000 + 1.000000x + 0.500000x^2$, which approximates $f(x) = e^x$ over [-1,1]. (b) The Chebyshev approximation $q(x) = 1.000000 + 1.129772x + 0.532042x^2$ for $f(x) = e^x$ over [-1,1].

Taylor Series and Calculation of Functions

Represent the elementary functions: $sin(x)$, $cos(x)$, e^x, $ln(x)$, etc. The next table gives several of the common Taylor series expansions. The partial sums can be accumulated until the accuracy specified.

Table 4.1 Taylor Series Expansions for Some Common Functions

$sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$	for all x
$cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots$	for all x
$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots$	for all x
$ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots$	$-1 \leq x \leq 1$
$arctan(x) = x - \frac{x^3}{3} - \frac{x^5}{5} - \frac{x^7}{7} + \cdots$	$-1 \leq x \leq 1$
$(1+x)^p = 1 + px + \frac{p(p-1)}{2!}x^2 + \frac{p(p-1)(p-2)}{3!}x^3 + \cdots$	for $\|x\| < 1$

Example: Compute the number e using the Taylor series in Table 4.1 with $x = 1$,

$$e^1 = 1 + \frac{1}{1!} + \frac{1^2}{2!} + \frac{1^3}{3!} + \frac{1^4}{4!} + \cdots + \frac{1^k}{k!} + \cdots$$

The sum of an infinite series requires that the partial sums S_N tend to a limit. The values of these sums are given in Table 4.2.

Table 4.2 Partial Sums S_n Used to Determine e

n	$S_n = 1 + \frac{1}{1!} + \frac{1}{2!} + \cdots + \frac{1}{n!}$
0	1.0
1	2.0
2	2.5
3	2.666666666666...
4	2.708333333333...
5	2.716666666666...
6	2.718055555555...
7	2.718253968254...
8	2.718278769841...
9	2.718281525573...
10	2.718281801146...
11	2.718281826199...
12	2.718281828286...
13	2.718281828447...
14	2.718281828458...
15	2.718281828459...

Taylor Series and Calculation of Functions

Theorem 9: Taylor Polynomial Approximation

Assume that $f \in C^{N+1}[a,b]$ and $x_0 \in [a,b]$ is a fixed value. If $x \in [a,b]$, then

$$f(x) = P_N(x) + E_N(x), \tag{1}$$

where $P_N(x)$ is a polynomial that can be used to approximate $f(x)$:

$$f(x) \approx P_N(x) = \sum_{k=0}^{N} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k. \tag{2}$$

The error term $E_N(x)$ has the form

$$E_n(x) = \frac{f^{N+1}(c)}{(N+1)!}(x - x_0)^{N+1} \tag{3}$$

for some value $c = c(x)$ that lies between x and x_0.
$E_N(x)$ is used to determine a bound for the accuracy of the approximation.

Example: Show why 15 terms are all that are needed to obtain the 13-digit approximation $e = 2.718281828459$ in Table 4.2.

Taylor Series and Calculation of Functions

Example: Show why 15 terms are all that are needed to obtain the 13-digit approximation $e = 2.718281828459$ in Table 4.2.

Expand $f(x) = e^x$ in a Taylor polynomial of degree 15 using the fixed value $x_0 = 0$ and involving the powers $(x-0)^k = x^k$. The derivatives required are $f'(x) = f''(x) = ... = f^{(16)} = e^x$. The first 15 derivatives are used to calculate the coefficients $a_k = e^0/k!$ and are used to write the polynomial

$$P_{15}(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^{15}}{15!} \tag{4}$$

Taylor Series and Calculation of Functions

Setting $x = 1$ in (4) gives the partial sum $S_15 = P_15$.Then the accuracy of the approximation is:

$$E_{15}(x) = \frac{f^{(16)}(c)x^{16}}{16!} \tag{5}$$

Setting $x = 1$ in (4) gives the partial sum $S_15 = P_15$.Then the accuracy of the approximation is:

$$E_{15}(x) = \frac{f^{(16)}(c)x^{16}}{16!} \tag{5}$$

Since we chose $x_0 = 0$ and $x = 1$, the value c lies between them, which implies that $e^c < e^1$. Notice that the partial sums in Table 4.2 are bounded above by 3. Combining these two inequalities yields $e^c < 3$, which is used in the following calculation

$$|E_{15}(1)| = \frac{|f^{(16)}(c)|}{16!} \leq \frac{e^c}{16!} < \frac{3}{16!} < 1.433844x10^{-13}$$

Taylor Series and Calculation of Functions

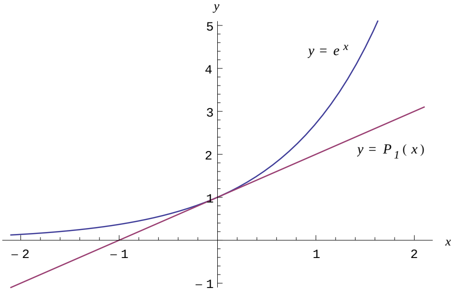
Setting $x = 1$ in (4) gives the partial sum $S_1 5 = P_1 5$.Then the accuracy of the approximation is:

$$E_{15}(x) = \frac{f^{(16)}(c)x^{16}}{16!} \qquad (5)$$

Since we chose $x_0 = 0$ and $x = 1$, the value c lies between them, which implies that $e^c < e^1$. Notice that the partial sums in Table 4.2 are bounded above by 3. Combining these two inequalities yields $e^c < 3$, which is used in the following calculation

$$|E_{15}(1)| = \frac{|f^{(16)}(c)|}{16!} \leq \frac{e^c}{16!} < \frac{3}{16!} < 1.433844x10^{-13}$$

Therefore, all the digits in the approximation $e \approx 2.718281828459$ are correct, because the actual error must be less than 2 in the thirteenth decimal place.



Observe that the approximation $e^x \approx 1 + x$ is good near the center $x_0 = 0$ and that the distance between the curves grows as x moves away from 0. The slopes of the curves agree at (0,1). The study of curvature shows that if two curves $y = f(x)$ and $y = g(x)$ have the property that $f(x_0) = g(x_0), f'(x_0) = g'(x_0)$ and $f''(x_0) = g''(x_0)$ then they have the same curvature at x_0. This property would be desirable for a polynomial function that approximates $f(x)$. Corollary 4.1 shows that the Taylor polynomial has this property for $N \geq 2$.

Corollary

If $P_N(x)$ is the Taylor polynomial of degree N given in Theorem 9, then

$$P_N^{(k)}(x_0) = f^{(k)}(x_0) \;\; for \; k = 0, 1, ..., N. \tag{6}$$

Proof. Set $x = x_0$ in equations (2) and (3), and the result is $P_N(x_0) = f(x_0)$. thus statement (6) is true for $k = 0$. Now differentiate the right-hand side of (2) and get

$$P_N'(x) = \sum_{k=1}^{N} \frac{f^{(k)}(x_0)}{(k-1)!}(x-x_0)^{k-1} = \sum_{k=0}^{N-1} \frac{f^{(k+1)}(x_0)}{k!}(x-x_0)^{k}. \tag{7}$$

Set $x = x_0$ in (7) to obtain $P_N'(x_0) = f'(x_0)$. Thus statement (6) is true for $k = 1$. Successive differentiations of (7) will establish the other identities in (6). The details are left as an exercise.

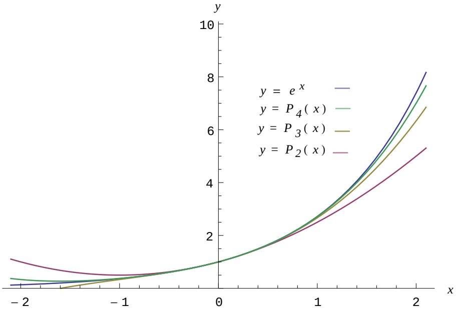
The accuracy of a Taylor polynomial is increased when we choose N large. The accuracy of any given polynomial will generally decrease as the value of x moves away from the center x_0. Hence we must choose N large enough and restrict the maximum value of $|x - x_0|$ so that the error does not exceed a specified bound.

$$|error| = |E_N(x)| \leq \frac{MR^{N+1}}{(N+1)!} \tag{8}$$

Taylor Series and Calculation of Functions

Table 4.3 Values for the Error Bound $|error| < e^R R^{N+1}/(N+1)!$

	$R = 2.0$, $\|x\| \le 2.0$	$R = 1.5$, $\|x\| \le 1.5$	$R = 1.0$, $\|x\| \le 1.0$	$R = 0.5$, $\|x\| \le 0.5$
$e^x \approx P_5(x)$	0.65680499	0.07090172	0.00377539	0.00003578
$e^x \approx P_6(x)$	0.18765857	0.01519323	0.00053934	0.00000256
$e^x \approx P_7(x)$	0.04691464	0.00284873	0.00006742	0.00000016
$e^x \approx P_8(x)$	0.01042548	0.00047479	0.00000749	0.00000001



Example: Establish the error bounds for the approximation $e^x \approx P_8(x)$ on each of the intervals $|x| \leq 1.0$ and $|x| \leq 0.5$.

Example: Establish the error bounds for the approximation $e^x \approx P_8(x)$ on each of the intervals $|x| \leq 1.0$ and $|x| \leq 0.5$.

if $|x| \leq 1.0$, then letting $R = 1.0$ and $|f^{(9)}(c)| = |e^c| \leq e^{1.0} = M$ in (8) implies that

$$|error| = |E_8(x)| \leq \frac{e^{1.0}(1.0)^9}{9!} \approx 0.00000749.$$

If $|x| \leq 0.5$, then letting $R = 0.5$ and $|f^{(9)}(c)| = |e^c| \leq e^{0.5} = M$ in (8) implies that

$$|error| = |E_8(x)| \leq \frac{e^{0.5}(0.5)^9}{9!} \approx 0.00000001.$$

Methods for Evaluating a Polynomial

Horner's Method (Nested Multiplication)

Consider, for example, the function

$$f(x) = (x-1)^8 \tag{9}$$

The binomial formula can be used to expand $f(x)$ in powers of x :

$$f(x) = \sum_{k=0}^{8} \binom{8}{k} x^{8-k}(-1)^k = x^8 - 8x^7 + 28x^6 - 56x^5 + 70x^4 - 56x^3 + 28x^2 - 8x + 1 \tag{10}$$

Now, Horner's Method can be used to evaluate the polynomial in (10). when applied to formula (10), nested multiplication permits us to write

$$f(x) = (((((((x-8)x+28)x-56)x+70)x-56)x+28)x-8)x+1 \tag{11}$$

to evaluate $f(x)$ now requires seven multiplications and eight additions or subtractions. The necessity of using an exponential function to evaluate the polynomial has now been eliminated.

Methods for Evaluating a Polynomial

Theorem 10: Taylor Series

Assume that $f(x)$ is analytic on an interval (a,b) containing x_0. Suppose that the Taylor polynomials (2) tend to a limit

$$S(x) = \lim_{N\to\infty} P_N(x) = \lim_{N\to\infty} \sum_{k=0}^{N} \frac{f^{(k)}(x_0)}{k!}(x-x_0)^k \tag{12}$$

then $f(x)$ has the Taylor series expansion

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x-x_0)^k \tag{13}$$

Proof. This follows directly form the definition of convergence of series. The limit condition is often stated by saying that the error term must go to zero as N goes to infinity. Therefore, a necessary and sufficient condition for (18) to hold is that

$$\lim_{N\to\infty} E_N(x) = \lim_{N\to\infty} \frac{f^{(N+1)}(c)(x-x_0)^{N+1}}{(N+1)!} = 0. \tag{14}$$

where c depends on N and x.

Suppose that the function $y = f(x)$ is known at the $N+1$ points $(x_0, y_0), ..., (x_N, y_N)$,where the values x_k are spread out over the interval $[a, b]$ and satisfy

$$a \leq x_0 < x_1 < \cdots < x_N \leq b \quad and \quad y_k = f(x_k).$$

A polynomial $P(x)$ of degree N will be constructed that passes through these $N+1$ points. In the construction, only the numerical values x_k and y_k are needed. Hence the higher-order derivatives are not necessary.The polynomial $P(k)$ can be used to approximate $f(x)$ over the entire interval $[a, b]$. However, if the error function $E(x) = f(x) - P(x)$ is required, then we will need to know $f^{(N+1)}(x)$ and a bound for its magnitude, that is,

$$M = max\{|f^{(N+1)}(x)| : a \leq x \leq b\}.$$

Let us briefly mention how to evaluate the polynomial $P(x)$:

$$P(x) = a_N x^N + a_{N-1}x^{N-1} + \cdots + a_2 x^2 + a_1 x + a_0 \tag{1}$$

Horner's Method of synthetic division is an efficient way to evaluate $P(x)$. The derivative $P'(x)$ is

$$P'(x) = Na_N x^{N-1} + (N-1)a_{N-1}x^{N-2} + \cdots + 2a_2 x + a1 \tag{2}$$

and the indefinite integral $I(x) = \int P(x)\,dx$, which satisfies $I'(x) = P(x)$, is

$$I(x) = \frac{a_N x^{N+1}}{N+1} + \frac{a_{N-1}x^N}{N} + \cdots + \frac{a_2 x^3}{3} + \frac{a_1 x^2}{2} + a_0 x + C \tag{3}$$

Where C is the constant of integration. Algorithm 4.1 shows how to adapt Horner's method to $P'(x)$ and $I(x)$.

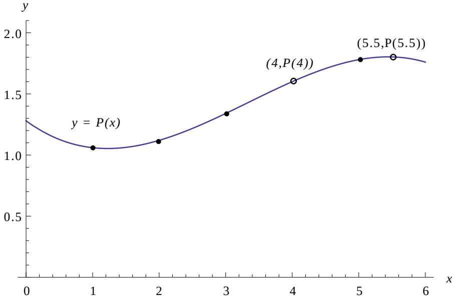
Example: The polynomial $P(x) = -0.02x^3 + 0.2x^2 - 0.4x + 1.28$ passes through the four points $(1, 1.06), (2, 2.12), (3, 1.34)$, and $(5, 1.78)$. Find (**a**) $P(4)$, (**b**) P'(4), (**c**) $\int_1^4 P(x)\,dx$, and (**d**) $P(5.5)$. Finally, (**e**) show how to find the coefficients of $P(x)$.

Example: The polynomial $P(x) = -0.02x^3 + 0.2x^2 - 0.4x + 1.28$ passes through the four points $(1, 1.06)$, $(2, 2.12)$, $(3, 1.34)$, and $(5, 1.78)$. Find (**a**) $P(4)$, (**b**) P'(4), (**c**) $\int_1^4 P(x)\,dx$, and (**d**) $P(5.5)$. Finally, (**e**) show how to find the coefficients of $P(x)$.

(**a**)

$$\begin{aligned} b_3 &= a_3 = -0.02 \\ b_2 &= a_2 + b_3 x = 0.2 + (-0.02)(4) = 0.12 \\ b_1 &= a_1 + b_2 x = -0.4 + (0.12)(4) = 0.08 \\ b_0 &= a_0 + b_1 x = 1.28 + (0.08)(4) = 1.60 \end{aligned}$$

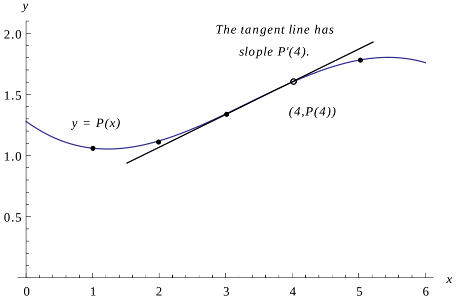
The interpolated value is $P(4) = 1.60$



(b)

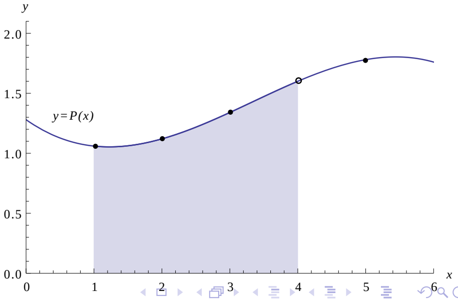
$$d_2 = 3a_3 = -0.06$$
$$d_1 = 2a_2 + d_2 x = 0.4 + (-0.06)(4) = 0.16$$
$$d_0 = a_1 + d_1 x = -0.4 + (0.16)(4) = 0.24$$

The numerical derivative is
$P'(4) = 0.24$



(c)

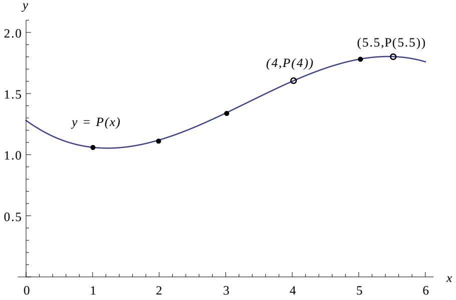
$$i_4 = \frac{a_3}{4} = -0.005$$
$$i_3 = \frac{a_2}{3} + i_4 x = 0.06666667 + (-0.005)(4) = 0.04666667$$
$$i_2 = \frac{a_1}{2} + i_3 x = -0.2 + (0.04666667)(4) = -0.01333333$$
$$i_1 = a_0 + i_2 x = 1.28 + (-0.01333333)(4) = 1.22666667$$
$$i_0 = 0 + i_1 x = 0 + (1.22666667)(4) = 4.90666667.$$



(d) Use Algorithm 4.1(i) with $x = 5.5$

$$b_3 = a_3 = -0.02$$
$$b_2 = a_2 + b_3 x = 0.2 + (-0.02)(5.5) = 0.09$$
$$b_1 = a_1 + b_2 x = -0.4 + (0.09)(5.5) = 0.095$$
$$b_0 = a_0 + b_1 x = 1.28 + (0.095)(5.5) = 1.8025$$

The extrapolated value is $P(5.5) = 1.8025$.



(e) Assume that $P(x) = A + Bx + Cx^2 + Dx^3$; then at each value $x = 1, 2, 3$, and 5 we get a linear equation involving A, B, C and D.

$$\begin{aligned}
Atx &= 1 : A + 1B + 1C + 1D = 1.06 \\
Atx &= 2 : A + 2B + 4C + 8D = 1.12 \\
Atx &= 3 : A + 3B + 9C + 27D = 1.34 \\
Atx &= 5 : A + 5B + 25C + 125D = 1.78
\end{aligned} \tag{4}$$

The solution to (4) is $A = 1.28, B = -0.4, C = 0.2$ and $D = -0.2$.

Algorithm 4.1 (Polynomial Calculus). To evaluate the polynomial $P(x)$, its derivative $P'(x)$, and its integral $\int P(x)\,dx$ by performing synthetic division.

INPUT N	Degree of $P(x)$
INPUT $A(0), A(1), \ldots, A(N)$	Coefficients of $P(x)$
INPUT C	Constant of integration
INPUT X	Independent variable

(i) Algorithm to Evaluate $P(x)$ $B(N) := A(N)$ FOR $K = N - 1$ DOWNTO 0 DO $B(K) := A(K) + B(K+1) * X$ PRINT "The Value $P(x)$ is", $B(0)$	Space-saving version: Poly $:= A(N)$ FOR $K = N - 1$ DOWNTO 0 DO Poly $:= A(K) +$ Poly$*X$ PRINT "The Value $P(x)$ is", Poly
(ii) Algorithm to Evaluate $P'(x)$ $D(N-1) := N * A(N)$ FOR $K = N - 1$ DOWNTO 1 DO $D(K-1) := K * A(K) + D(K) * X$ PRINT "The Value $P'(x)$ is", $D(0)$	Space-saving version: Deriv $:= N * A(N)$ FOR $K = N - 1$ DOWNTO 1 DO Deriv $:= K * A(K)$+Deriv$*X$ PRINT "The Value $P'(x)$ is", Deriv
(iii) Algorithm to Evaluate $I(x)$ $I(N+1) := A(N)/(N+1)$ FOR $K = N$ DOWNTO 1 DO $I(K) := A(K-1)/K + I(K+1) * X$ $I(0) := C + I(1) * X$ PRINT "The Value $I(x)$ is", $I(0)$	Space-saving version: Integ $:= A(N)/(N+1)$ FOR $K = N$ DOWNTO 1 DO Integ $:= A(K-1)/K$ + Integ$*X$ Integ$:= C + Integ * X$ PRINT "The Value $I(x)$ is", Integ

Interpolation means to estimate a missing function value by taking a weighted average of known function values at neighboring points.

Linear interpolation uses a line segment that passes through two points. The slope between (x_0, y_0) and (x_1, y_1) is $m = (y_1 - y_0)/(x_1 - x_0)$, and the point-slope formula for the line $y = m(x - x_0) + y_0$ can be rearranged as

$$y = P(x) = y_0 + (y_1 - y_0)\frac{x - x_0}{x_1 - x_0} \tag{5}$$

When formula (1) is expanded, the result is a polynomial of degree ≤ 1. Evaluation of $P(x)$ at x_0 and x_1 produces y_0 and y_1, respectively:

$$\begin{aligned} P(x_0) &= x_0 + (y_1 - y_0)(0) = y_0, \\ P(x_1) &= y_0 + (y_1 - y_0)(1) = y_1. \end{aligned} \tag{6}$$

Lagrange used a slightly different method to find this polynomial. He noticed that it could be written as

$$y = P_1(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}. \tag{7}$$

Each term on the right side of (3) involves a linear factor; hence the sum is a polynomial of degree ≤ 1. The quotients in (3) are denoted by

$$L_{1,0}(0) = \frac{x - x_1}{x_0 - x_1} \quad and \quad L_{1,1}(x) = \frac{x - x_0}{x_1 - x_0}. \tag{8}$$

Computation reveals that $L_{1,0}(x_0) = 1$, $L_{1,0}(x_1) = 0$, $L_{1,1}(x_0) = 0$, and $L_{1,1}(x_1) = 1$ so that the polynomial $P_1(x)$ in (3) also passes through the two given points:

$$P_1(x_0) = y_0 + y_1(0) = y_0 \quad and \quad P_1(x_1) = y_0(0) + y_1 = y_1 \tag{9}$$

Lagrange Approximation

The terms $L_{1,0}(x)$ and $L_{1,1}(x)$ in (4) are called **Lagrange coefficient polynomials** based on the nodes x_0 and x_1. Using this notation, (3) can be written in summation form

$$P_1(x) = \sum_{k=0}^{1} y_k L_{1,k}(x). \tag{10}$$

Suppose that the ordinates y_k are computed with the formula $y_k = f(x_k)$. If $P_1(x)$ is used to approximate $f(x)$ over the interval $[x_0, x_1]$, we call the process **interpolation**. If $x < x_0$ (or $x_1 < x$), then using $P_1(x)$ is called **extrapolation**.

Lagrange Approximation

Example 4.6: Consider the graph $y = f(x) = cos(x)$ over $[0.0, 1.2]$.
(**a**) Use the nodes $x_0 = 0.0$ and $x_1 = 1.2$ to construct a linear interpolation polynomial $P_1(x)$.
(**b**) Use the nodes $x_0 = 0.2$ and $x_1 = 1.0$ to construct a linear approximating polynomial $Q_1(x)$.

Lagrange Approximation

Example 4.6: Consider the graph $y = f(x) = cos(x)$ over $[0.0, 1.2]$.
(**a**) Use the nodes $x_0 = 0.0$ and $x_1 = 1.2$ to construct a linear interpolation polynomial $P_1(x)$.
(**b**) Use the nodes $x_0 = 0.2$ and $x_1 = 1.0$ to construct a linear approximating polynomial $Q_1(x)$.

(**a**) Using (3) with the abscissas $x_0 = 0.0$ and $x_1 = 1.2$ and the ordinates $y_0 = cos(0.0) = 1.000000$ and $y_1 = cos(1.2) = 0.362358$ produces

$$\begin{aligned} P_1(x) &= 1.000000\frac{x-1.2}{0.0-1.2} + 0.362358\frac{x-0.0}{1.2-0.0} \\ &= -0.833333(x-1.2) + 0.301965(x-0.0). \end{aligned}$$

Example 4.6: Consider the graph $y = f(x) = cos(x)$ over $[0.0, 1.2]$.
(**a**) Use the nodes $x_0 = 0.0$ and $x_1 = 1.2$ to construct a linear interpolation polynomial $P_1(x)$.
(**b**) Use the nodes $x_0 = 0.2$ and $x_1 = 1.0$ to construct a linear approximating polynomial $Q_1(x)$.

(**a**) Using (3) with the abscissas $x_0 = 0.0$ and $x_1 = 1.2$ and the ordinates $y_0 = cos(0.0) = 1.000000$ and $y_1 = cos(1.2) = 0.362358$ produces

$$\begin{aligned} P_1(x) &= 1.000000\frac{x-1.2}{0.0-1.2} + 0.362358\frac{x-0.0}{1.2-0.0} \\ &= -0.833333(x-1.2) + 0.301965(x-0.0). \end{aligned}$$

(**b**) When the nodes $x_0 = 0.2$ and $x_1 = 1.0$ with $y_0 = cos(0.2) = 0.980067$ and $y_1 = cos(1.0) = 0.540302$ are used, the result is

$$\begin{aligned} Q_1(x) &= 0.980067\frac{x-1.0}{0.2-1.0} + 0.540302\frac{x-0.2}{1.0-0.2} \\ &= -1.225083(x-1.0) + 0.675378(x-0.2). \end{aligned}$$

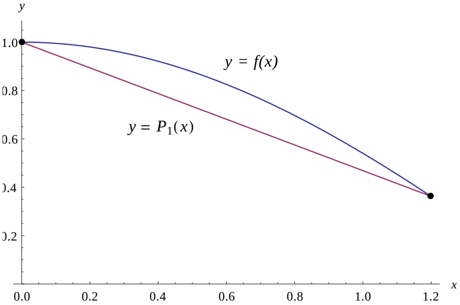
Lagrange Approximation

Table 4.6 Comparison of $f(x) = cos(x)$ and the Linear Approximations $P_1(x)$ and $Q_1(x)$

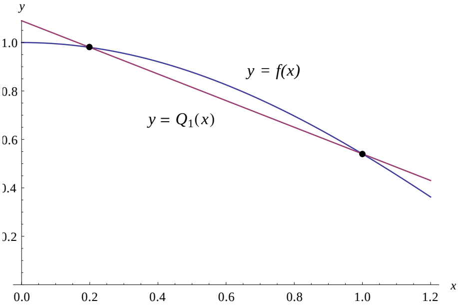
x_k	$f(x_k) = cos(x_k)$	$P_1(x_k)$	$f(x_k) - P_1(x_k)$	$Q_1(x_k)$	$f(x_k) - Q_1(x_k)$
0.0	1.000000	1.000000	0.000000	1.090008	−0.090008
0.1	0.995004	0.946863	0.048141	1.035037	−0.040033
0.2	0.980067	0.893726	0.086340	0.980067	0.000000
0.3	0.955336	0.840589	0.114747	0.925096	0.030240
0.4	0.921061	0.787453	0.133608	0.870126	0.050935
0.5	0.877583	0.734316	0.143267	0.815155	0.062428
0.6	0.825336	0.681179	0.144157	0.760184	0.065151
0.7	0.764842	0.628042	0.136800	0.705214	0.059628
0.8	0.696707	0.574905	0.121802	0.650243	0.046463
0.9	0.621610	0.521768	0.099842	0.595273	0.026337
1.0	0.540302	0.468631	0.071671	0.540302	0.000000
1.1	0.453596	0.415495	0.038102	0.485332	−0.031736
1.2	0.362358	0.362358	0.000000	0.430361	−0.068003

Lagrange Approximation

Numerical Computations are given in Table 4.6 and reveal that $Q_1(x)$ has less error at the points xk that satisfy $0.1 \leq x_k \leq 1.1$. The largest tabulated error, $f(0.6) - P_1(0.6) = 0.144157$, is reduced to $f(0.6) - Q_1(0.6) = 0.065151$ by using $Q_1(x)$.



The linear approximation $y = P_1(x)$ where the nodes $x_0 = 0.0$ and $x_1 = 1.2$ are the endpoints of the interval $[a, b]$.



The linear approximation $y = Q_1(x)$ where the nodes $x_0 = 0.2$ and $x_1 = 1.0$ lie inside the interval $[a, b]$.

Lagrange Approximation

The generalization of (6) is the construction of a polynomial $P_N(x)$ of degree at most N that passes through the $N+1$ points $(x_0, y_0), (x_1, y_1), ..., (x_N, y_N)$ and has the form

$$P_N(x) = \sum_{k=0}^{N} y_k L_{N,k}(x), \tag{11}$$

where $L_{N,k}$ is the Lagrange coefficient polynomial based on these nodes:

$$L_{N,k}(x) = \frac{(x-x_0)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_N)}{(x_k-x_0)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_N)} \tag{12}$$

The product notation for (8), is written as

$$L_{N,k}(x) = \frac{\prod_{\substack{j=0 \\ j\neq k}}^{N}(x-x_j)}{\prod_{\substack{j=0 \\ j\neq k}}^{N}(x_k-x_j)} \tag{13}$$

Lagrange Approximation

For each fixed k, the Lagrange coefficient polynomial $L_{N,k}(x)$ has the property

$$L_{N,k}(x_j) = 1 \ \ when \ \ j = k \ \ \ and \ \ \ L_{N,k}(x_j) = 0 \ \ when \ \ j \neq k. \tag{14}$$

Then direct substitution of these values into (7) is used to show that the polynomial curve $y = P_N(x)$ goes through (x_j, y_j):

$$\begin{aligned} P_N(x_j) &= y_0 L_{N,0}(x_j) + \cdots + y_j L_{N,j}(x_j) + \cdots + y_N L_{N,N}(x_j) \\ &= y_0(0) + \cdots + y_j(1) + \cdots + y_N(0) = y_j \end{aligned} \tag{15}$$

To show that $P_N(x)$ is unique, we invoke the fundamental theorem of algebra, which states that a nonzero polynomial $T(x)$ of degree $\leq N$ has at most N roots. In other words, if $T(x)$ is zero at $N + 1$ distinct abscissas, it is identically zero.

Lagrange Approximation

Suppose that $P_N(x)$ is not unique and that there exist another polynomial $Q_N(x)$ of degree $\leq N$ that also passes through the $N+1$ points.

Form the difference polynomial $T(x) = P_N(x) - Q_N(x)$.

Observe that the polynomial $T(x)$ has degree $\leq N$ and that $T(x_j) = P_N(x_j) - Q_N(x_j) = y_j - y_j = 0$, for $j = 0, 1, ..., N$. Therefore, $T(x) \equiv 0$ and it follows that $Q_N(x) = P_N(x)$.

Lagrange Approximation

When (7) is expanded, the result is similar to (3). The Lagrange quadratic interpolating polynomial through the three points (x_0, y_0), (x_1, y_1), and (x_2, y_2) is

$$P_2(x) = y_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \quad (16)$$

The Lagrange cubic interpolating polynomial through the four points (x_0, y_0), (x_1, y_1), (x_2, y_2) and (x_3, y_3) is

$$\begin{aligned} P_3(x) &= y_0 \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} + y_1 \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} \quad (17) \\ &\quad + y_2 \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} + y_3 \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} \end{aligned}$$

Example 4.7: Consider $y = f(x) = cos(x)$ over $[0.0, 1.2]$.

(**a**) Use the three nodes $x_0 = 0.0, x_1 = 0.6$ and $x_2 = 1.2$ to construct a quadratic interpolation polynomial $P_2(x)$.
(**b**) Use the four nodes $x_0 = 0.0, x_1 = 0.4, x_2 = 0.8$ and $x_3 = 1.2$ to construct a cubic interpolation polynomial $P_3(x)$.

Example 4.7: Consider $y = f(x) = cos(x)$ over $[0.0, 1.2]$.

(**a**) Use the three nodes $x_0 = 0.0, x_1 = 0.6$ and $x_2 = 1.2$ to construct a quadratic interpolation polynomial $P_2(x)$.
(**b**) Use the four nodes $x_0 = 0.0, x_1 = 0.4, x_2 = 0.8$ and $x_3 = 1.2$ to construct a cubic interpolation polynomial $P_3(x)$.

(**a**) Using $x_0 = 0.0, x_1 = 0.6, x_2 = 1.2$ and
$y_0 = cos(0.0) = 1.0, y_1 = cos(0.6) = 0.825336$, and
$y_2 = cos(1.2) = 0.362358$ in equation (12) produces

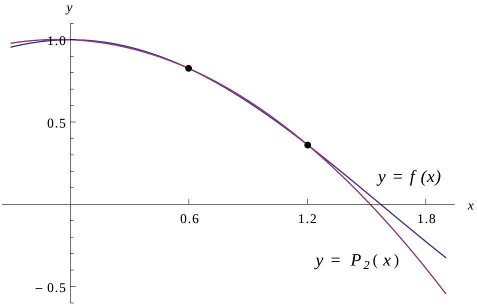
$$\begin{aligned}
P_2(x) &= 1.0\frac{(x-0.6)(x-1.2)}{(0.0-0.6)(0.0-1.2)} + 0.825336\frac{(x-0.0)(x-1.2)}{(0.6-0.0)(0.6-1.2)} \\
&\quad +0.362358\frac{(x-0.0)(x-0.6)}{(1.2-0.0)(1.2-0.6)} \\
&= 1.388889(x-0.6)(x-1.2) - 2.292599(x-0.0)(x-1.2) \\
&\quad +0.503275(x-0.0)(x-0.6).
\end{aligned}$$

Lagrange Approximation

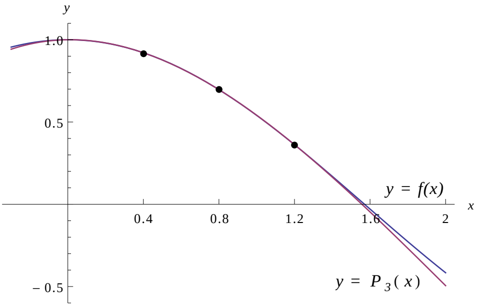
(**b**) Using $x_0 = 0.0, x_1 = 0.4, x_2 = 0.8, x_3 = 1.2$ and $y_0 = cos(0.0) = 1.0, y_1 = cos(0.4) = 0.921061, y_2 = cos(0.8) = 0.696707$, and $y_3 = cos(1.2) = 0.362358$ in equation (13) produces

$$
\begin{aligned}
P_3(x) &= 1.000000\frac{(x-0.4)(x-0.8)(x-1.2)}{(0.0-0.4)(0.0-0.8)(0.0-1.2)} \\
&\quad +0.921061\frac{(x-0.0)(x-0.8)(x-1.2)}{(0.4-0.0)(0.4-0.8)(0.4-1.2)} \\
&\quad +0.696707\frac{(x-0.0)(x-0.4)(x-1.2)}{(0.8-0.0)(0.8-0.4)(0.8-1.2)} \\
&\quad +0.362358\frac{(x-0.0)(x-0.4)(x-0.8)}{(1.2-0.0)(1.2-0.4)(1.2-0.8)} \\
&= -2.604167(x-0.4)(x-0.8)(x-1.2) \\
&\quad +7.195789(x-0.0)(x-0.8)(x-1.2) \\
&\quad -5.443021(x-0.0)(x-0.4)(x-1.2) \\
&\quad +0.943641(x-0.0)(x-0.4)(x-0.8).
\end{aligned}
$$

Lagrange Approximation



The quadratic approximation polynomial $y = P_2(x)$ based on the nodes $x_0 = 0.0, x_1 = 0.6$ and $x_2 = 1.2$.



The cubic approximation polynomial $y = P_3(x)$ based on the nodes $x_0 = 0.0, x_1 = 0.4, x_2 = 0.8$ and $x_3 = 1.2$.

Lagrange Approximation

Theorem 4.3 (Lagrange Polynomial Approximation)

Assume that $f \in C^{N+1}[a,b]$ and that $x_0, x_1, ..., x_N \in [a,b]$ are $N+1$ nodes. If $x \in [a,b]$, then

$$f(x) = P_N(x) + E_N(x), \tag{18}$$

where P_N is a polynomial that can be used to approximate $f(x)$:

$$f(x) \approx P_N(x) = \sum_{k=0}^{N} f(x_k) L_{N,k}(x). \tag{19}$$

The error term $E_N(x)$ has the form

$$E_N(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_N)f^{(N+1)}(c)}{(N+1)!} \tag{20}$$

for some value $c = c(x)$ that lies in the interval $[a,b]$.

Lagrange Approximation

Theorem 4.4 (Error Bounds for Lagrange Interpolation, Equally Spaced Nodes)

Assume that $f(x)$ is defined on $[a, b]$, which contains equally spaced nodes $x_k = x_0 + hk$. Additionally, assume that $f(x)$ and the derivates of $f(x)$, up to the order $N+1$, are continuous and bounded on the special subintervals $[x_0, x_1]$, $[x_0, x_2]$, and $[x_0, x_3]$, respectively; that is,

$$|f^{(N+1)}(x)| \leq M_{N+1} \quad for \ \ x_0 \leq x \leq x_N, \tag{21}$$

for $N = 1, 2, 3$. The error terms (16) corresponding to the cases $N = 1, 2$, and 3 have the following useful bounds on their magnitude:

$$|E_1(x)| \leq \frac{h^2 M_2}{8} \quad valid \ \ for \ \ x \in [x_0, x_1], \tag{22}$$

$$|E_2(x)| \leq \frac{h^3 M_3}{9\sqrt{3}} \quad valid \ \ for \ \ x \in [x_0, x_2], \tag{23}$$

$$|E_3(x)| \leq \frac{h^4 M_4}{24} \quad valid \ \ for \ \ x \in [x_0, x_3]. \tag{24}$$

Comparison of Accuracy and $O(h^{N+1})$

The significance of Theorem 4.4 is to understand a simple relationship between the size of the error terms for linear, quadratic, and cubic interpolation. In each case the error bound $|E_N(x)|$ depends on h in two ways. First h^{N+1} is explicitly present to that $|E_N(x)|$ is proportional h^{N+1}. Second, the values M_{N+1} generally depend on h and tend to $|f^{(N+1)}(x_0)|$ as h goes to zero. Therefore, as h goes to zero, $|E_N(x)|$ converges to zero with the same rapidity that h^{N+1} converges to zero. The notation $O(h^{N+1})$ is used when discussing this behavior. For example, the error bound (18) can expressed as

$$|E_1(x)| = O(h^2) \quad valid\ for\ x \in [x_0, x_1]$$

The notation $O(h^2)$ stands in place of $h^2 M_2/8$ in relation (18) and is meant to convey the idea that the bound for the error term is approximately a multiple of h^2; that is,

$$|E_1(x)| \leq Ch^2 \approx O(h^2).$$

As a consequence, if the derivatives of $f(x)$ are uniformly bounded on the interval $[a, b]$ and $|h| < 1$, the choosing N large will make h^{N+1} small, and the higher-degree approximating polynomial will have less error.

Example 4.8: Consider $y = f(x) = cos(x)$ over $[0.0, 1.2]$. Use formula (18) through (20) and determine the error bounds for the Lagrange polynomials $P_1(x), P_2(x)$, and $P_3(x)$ that were constructed in Examples 4.6 and 4.7.

Example 4.8: Consider $y = f(x) = cos(x)$ over $[0.0, 1.2]$. Use formula (18) through (20) and determine the error bounds for the Lagrange polynomials $P_1(x), P_2(x)$, and $P_3(x)$ that were constructed in Examples 4.6 and 4.7. First, determine the bounds M_2, M_3, and M_4 for the

derivatives $|f^{(2)}(x)|, |f^{(3)}(x)|$ and $|f^{(4)}(x)|$, respectively, taken over the interval $[0.0, 1.2]$:

$$\begin{aligned}
|f^{(2)}(x)| &= |-cos(x)| \leq |-cos(0.0)| = 1.000000 = M_2, \\
|f^{(3)}(x)| &= |sin(x)| \leq |sin(1.2)| = 0.932039 = M_3, \\
|f^{(4)}(x)| &= |cos(x)| \leq |cos(0.0)| = 1.000000 = M_4.
\end{aligned}$$

For $P_1(x)$ the spacing of the nodes is $h = 1.2$, and its error bound is

$$|E_1(x)| \leq \frac{h^2 M_2}{8} \leq \frac{(1.2)^2(1.000000)}{8} = 0.180000 \tag{25}$$

For $P_2(x)$ the spacing of the nodes is $h = 0.6$, and its error bound is

$$|E_2(x)| \leq \frac{h^3 M_3}{9\sqrt{3}} \leq \frac{(0.6)^3(0.932039)}{9\sqrt{3}} = 0.012915 \quad (26)$$

For $P_3(x)$ the spacing of the nodes is $h = 0.4$, and its error bound is

$$|E_3(x)| \leq \frac{h^4 M_4}{24} \leq \frac{(0.4)^4(1.000000)}{24} = 0.001067 \quad (27)$$

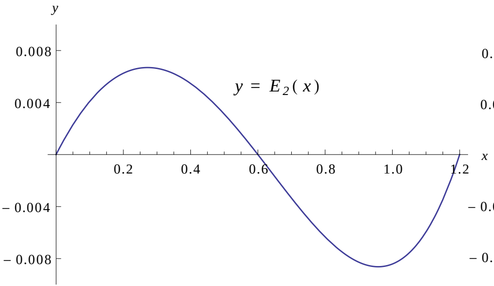
From Example 4.6 we saw that
$|E_1(0.6)| = |cos(0.6) - P_1(0.6)| = 0.144157$, so the bound 0.180000 in (21) is reasonable.

Lagrange Approximation

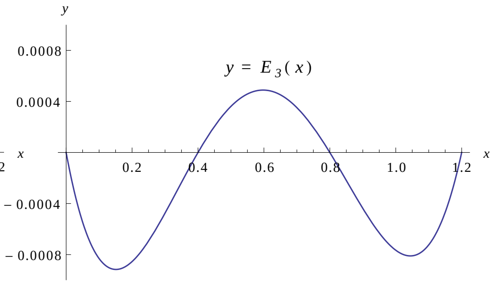
Table 4.7 Comparison of $f(x) = cos(x)$ and the Quadratic and Cubic Polynomial Approximations $P_2(x)$ and $P_3(x)$

x_k	$f(x_k) = cos(x_k)$	$P_2(x_k)$	$E_2(x_k)$	$P_3(x_k)$	$E_3(x_k)$
0.0	1.000000	1.000000	0.000000	1.000000	0.000000
0.1	0.995004	0.990911	0.004093	0.995835	−0.000831
0.2	0.980067	0.973813	0.006253	0.980921	−0.000855
0.3	0.955336	0.948707	0.006629	0.955812	−0.000476
0.4	0.921061	0.915592	0.005469	0.921061	0.000000
0.5	0.877583	0.874468	0.003114	0.877221	0.000361
0.6	0.825336	0.825336	0.000000	0.824847	0.000890
0.7	0.764842	0.768194	−0.003352	0.764491	0.000351
0.8	0.696707	0.703044	−0.006338	0.696707	0.000000
0.9	0.621610	0.629886	−0.008276	0.622048	−0.000438
1.0	0.540302	0.548719	−0.008416	0.541068	−0.000765
1.1	0.453596	0.459542	−0.005946	0.454320	−0.000724
1.2	0.362358	0.362358	0.000000	0.362358	0.000000

Lagrange Approximation



The error function $E_2(x) = cos(x) - P_2(x)$.



The error function $E_3(x) = cos(x) - P_3(x)$.

Lagrange Approximation

Interpolation means to estimate a missing function value by taking a weighted average of known function values at neighboring points. Linear interpolation uses a line segment that passes through two points. The slope between (x_0, y_0) and (x_1, y_1) is $m = (y_1 - y_0)/(x_1 - x_0)$, and the point-slope formula for the line $y = m(x - x_0) + y_0$ can be rearranged as

$$y = P(x) = y_0 + (y_1 - y_0)\frac{x - x_0}{x_1 - x_0} \tag{28}$$

When formula (1) is expanded, the result is a polynomial of degree ≤ 1. Evaluation of $P(x)$ at x_0 and x_1 produces y_0 and y_1, respectively:

$$\begin{aligned} P(x_0) &= x_0 + (y_1 - y_0)(0) = y_0, \\ P(x_1) &= y_0 + (y_1 - y_0)(1) = y_1. \end{aligned} \tag{29}$$

Lagrange Approximation

The French mathematician Joseph Louis Lagrange used a slightly different method to find this polynomial. He noticed that it could be written as

$$y = P_1(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0}. \tag{30}$$

Each term on the right side of (3) involves a linear factor; hence the sum is a polynomial of degree ≤ 1. The quotients in (3) are denoted by

$$L_{1,0}(0) = \frac{x - x_1}{x_0 - x_1} \quad and \quad L_{1,1}(x) = \frac{x - x_0}{x_1 - x_0}. \tag{31}$$

Computation reveals that $L_{1,0}(x_0) = 1$, $L_{1,0}(x_1) = 0$, $L_{1,1}(x_0) = 0$, and $L_{1,1}(x_1) = 1$ so that the polynomial $P_1(x)$ in (3) also passes through the two given points:

$$P_1(x_0) = y_0 + y_1(0) = y_0 \quad and \quad P_1(x_1) = y_0(0) + y_1 = y_1 \tag{32}$$

Lagrange Approximation

The terms $L_{1,0}(x)$ and $L_{1,1}(x)$ in (4) are called **Lagrange coefficient polynomials** based on the nodes x_0 and x_1. Using this notation, (3) can be written in summation form

$$P_1(x) = \sum_{k=0}^{1} y_k L_{1,k}(x). \tag{33}$$

Suppose that the ordinates y_k are computed with the formula $y_k = f(x_k)$. If $P_1(x)$ is used to approximate $f(x)$ over the interval $[x_0, x_1]$, we call the process **interpolation**. If $x < x_0$ (or $x_1 < x$), then using $P_1(x)$ is called **extrapolation**.

Lagrange Approximation

Example 4.6: Consider the graph $y = f(x) = cos(x)$ over $[0.0, 1.2]$.
(**a**) Use the nodes $x_0 = 0.0$ and $x_1 = 1.2$ to construct a linear interpolation polynomial $P_1(x)$.
(**b**) Use the nodes $x_0 = 0.2$ and $x_1 = 1.0$ to construct a linear approximating polynomial $Q_1(x)$.

Lagrange Approximation

Example 4.6: Consider the graph $y = f(x) = cos(x)$ over $[0.0, 1.2]$.
(**a**) Use the nodes $x_0 = 0.0$ and $x_1 = 1.2$ to construct a linear interpolation polynomial $P_1(x)$.
(**b**) Use the nodes $x_0 = 0.2$ and $x_1 = 1.0$ to construct a linear approximating polynomial $Q_1(x)$.

(**a**) Using (3) with the abscissas $x_0 = 0.0$ and $x_1 = 1.2$ and the ordinates $y_0 = cos(0.0) = 1.000000$ and $y_1 = cos(1.2) = 0.362358$ produces

$$\begin{aligned} P_1(x) &= 1.000000\frac{x-1.2}{0.0-1.2} + 0.362358\frac{x-0.0}{1.2-0.0} \\ &= -0.833333(x-1.2) + 0.301965(x-0.0). \end{aligned}$$

Example 4.6: Consider the graph $y = f(x) = cos(x)$ over $[0.0, 1.2]$.
(**a**) Use the nodes $x_0 = 0.0$ and $x_1 = 1.2$ to construct a linear interpolation polynomial $P_1(x)$.
(**b**) Use the nodes $x_0 = 0.2$ and $x_1 = 1.0$ to construct a linear approximating polynomial $Q_1(x)$.

(**a**) Using (3) with the abscissas $x_0 = 0.0$ and $x_1 = 1.2$ and the ordinates $y_0 = cos(0.0) = 1.000000$ and $y_1 = cos(1.2) = 0.362358$ produces

$$\begin{aligned} P_1(x) &= 1.000000\frac{x-1.2}{0.0-1.2} + 0.362358\frac{x-0.0}{1.2-0.0} \\ &= -0.833333(x-1.2) + 0.301965(x-0.0). \end{aligned}$$

(**b**) When the nodes $x_0 = 0.2$ and $x_1 = 1.0$ with $y_0 = cos(0.2) = 0.980067$ and $y_1 = cos(1.0) = 0.540302$ are used, the result is

$$\begin{aligned} Q_1(x) &= 0.980067\frac{x-1.0}{0.2-1.0} + 0.540302\frac{x-0.2}{1.0-0.2} \\ &= -1.225083(x-1.0) + 0.675378(x-0.2). \end{aligned}$$

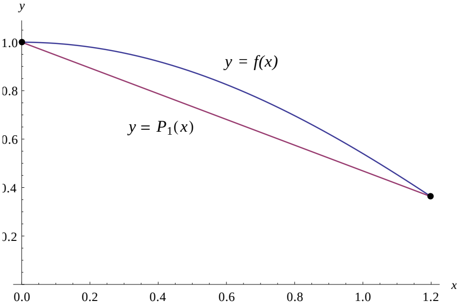
Lagrange Approximation

Table 4.6 Comparison of $f(x) = cos(x)$ and the Linear Approximations $P_1(x)$ and $Q_1(x)$

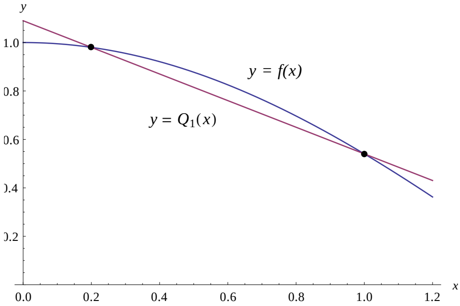
x_k	$f(x_k) = cos(x_k)$	$P_1(x_k)$	$f(x_k) - P_1(x_k)$	$Q_1(x_k)$	$f(x_k) - Q_1(x_k)$
0.0	1.000000	1.000000	0.000000	1.090008	−0.090008
0.1	0.995004	0.946863	0.048141	1.035037	−0.040033
0.2	0.980067	0.893726	0.086340	0.980067	0.000000
0.3	0.955336	0.840589	0.114747	0.925096	0.030240
0.4	0.921061	0.787453	0.133608	0.870126	0.050935
0.5	0.877583	0.734316	0.143267	0.815155	0.062428
0.6	0.825336	0.681179	0.144157	0.760184	0.065151
0.7	0.764842	0.628042	0.136800	0.705214	0.059628
0.8	0.696707	0.574905	0.121802	0.650243	0.046463
0.9	0.621610	0.521768	0.099842	0.595273	0.026337
1.0	0.540302	0.468631	0.071671	0.540302	0.000000
1.1	0.453596	0.415495	0.038102	0.485332	−0.031736
1.2	0.362358	0.362358	0.000000	0.430361	−0.068003

Lagrange Approximation

Numerical Computations are given in Table 4.6 and reveal that $Q_1(x)$ has less error at the points xk that satisfy $0.1 \leq x_k \leq 1.1$. The largest tabulated error, $f(0.6) - P_1(0.6) = 0.144157$, is reduced to $f(0.6) - Q_1(0.6) = 0.065151$ by using $Q_1(x)$.



The linear approximation $y = P_1(x)$ where the nodes $x_0 = 0.0$ and $x_1 = 1.2$ are the endpoints of the interval $[a, b]$.



The linear approximation $y = Q_1(x)$ where the nodes $x_0 = 0.2$ and $x_1 = 1.0$ lie inside the interval $[a, b]$.

Lagrange Approximation

The generalization of (6) is the construction of a polynomial $P_N(x)$ of degree at most N that passes through the $N+1$ points $(x_0, y_0), (x_1, y_1), \ldots, (x_N, y_N)$ and has the form

$$P_N(x) = \sum_{k=0}^{N} y_k L_{N,k}(x), \tag{34}$$

where $L_{N,k}$ is the Lagrange coefficient polynomial based on these nodes:

$$L_{N,k}(x) = \frac{(x - x_0)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_N)}{(x_k - x_0)\cdots(x_k - x_{k-1})(x_k - x_{k+1})\cdots(x_k - x_N)} \tag{35}$$

It is understood that the terms $(x - x_k)$ and $x_k - x_k$ do not appear on the right side of equation (8). It is appropriate to introduce the product notation for (8), and we write

$$L_{N,k}(x) = \frac{\prod_{\substack{j=0 \\ j\neq k}}^{N}(x - x_j)}{\prod_{\substack{j=0 \\ j\neq k}}^{N}(x_k - x_j)} \tag{36}$$

A Straightforward calculation shows that for each fixed k, the Lagrange coefficient polynomial $L_{N,k}(x)$ has the property

$$L_{N,k}(x_j) = 1 \quad when \quad j = k \quad and \quad L_{N,k}(x_j) = 0 \quad when \quad j \neq k. \tag{37}$$

Then direct substitution of these values into (7) is used to show that the polynomial curve $y = P_N(x)$ goes through (x_j, y_j):

$$\begin{aligned} P_N(x_j) &= y_0 L_{N,0}(x_j) + \cdots + y_j L_{N,j}(x_j) + \cdots + y_N L_{N,N}(x_j) \\ &= y_0(0) + \cdots + y_j(1) + \cdots + y_N(0) = y_j \end{aligned} \tag{38}$$

To show that $P_N(x)$ is unique, we invoke the fundamental theorem of algebra, which states that a nonzero polynomial $T(x)$ of degree $\leq N$ has at most N roots. In other words, if $T(x)$ is zero at $N+1$ distinct abscissas, it is identically zero. Suppose that $P_N(x)$ is not unique and that there exist another polynomial $Q_N(x)$ of degree $\leq N$ that also passes through the $N+1$ points. Form the difference polynomial $T(x) = P_N(x) - Q_N(x)$. Observe that the polynomial $T(x)$ has degree $\leq N$ and that $T(x_j) = P_N(x_j) - Q_N(x_j) = y_j - y_j = 0$, for $j = 0, 1, ..., N$. Therefore, $T(x) \equiv 0$ and it follows that $Q_N(x) = P_N(x)$.

Lagrange Approximation

When (7) is expanded, the result is similar to (3). The Lagrange quadratic interpolating polynomial through the three points (x_0, y_0), (x_1, y_1), and (x_2, y_2) is

$$P_2(x) = y_0 \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + y_1 \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + y_2 \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \quad (39)$$

The Lagrange cubic interpolating polynomial through the four points (x_0, y_0), (x_1, y_1), (x_2, y_2) and (x_3, y_3) is

$$\begin{aligned} P_3(x) &= y_0 \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} + y_1 \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} \quad (40) \\ &\quad + y_2 \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} + y_3 \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} \end{aligned}$$

Lagrange Approximation

Example 4.7: Consider $y = f(x) = cos(x)$ over $[0.0, 1.2]$.

(**a**) Use the three nodes $x_0 = 0.0, x_1 = 0.6$ and $x_2 = 1.2$ to construct a quadratic interpolation polynomial $P_2(x)$.
(**b**) Use the four nodes $x_0 = 0.0, x_1 = 0.4, x_2 = 0.8$ and $x_3 = 1.2$ to construct a cubic interpolation polynomial $P_3(x)$.

Lagrange Approximation

Example 4.7: Consider $y = f(x) = cos(x)$ over $[0.0, 1.2]$.

(**a**) Use the three nodes $x_0 = 0.0, x_1 = 0.6$ and $x_2 = 1.2$ to construct a quadratic interpolation polynomial $P_2(x)$.
(**b**) Use the four nodes $x_0 = 0.0, x_1 = 0.4, x_2 = 0.8$ and $x_3 = 1.2$ to construct a cubic interpolation polynomial $P_3(x)$.

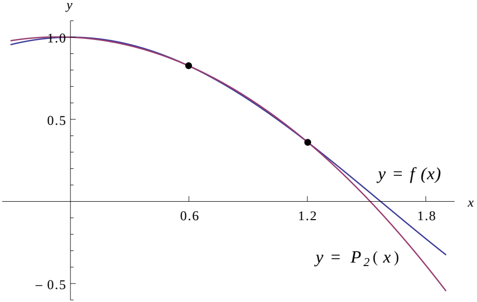
(**a**) Using $x_0 = 0.0, x_1 = 0.6, x_2 = 1.2$ and
$y_0 = cos(0.0) = 1.0, y_1 = cos(0.6) = 0.825336$, and
$y_2 = cos(1.2) = 0.362358$ in equation (12) produces

$$
\begin{aligned}
P_2(x) &= 1.0\frac{(x-0.6)(x-1.2)}{(0.0-0.6)(0.0-1.2)} + 0.825336\frac{(x-0.0)(x-1.2)}{(0.6-0.0)(0.6-1.2)} \\
&\quad +0.362358\frac{(x-0.0)(x-0.6)}{(1.2-0.0)(1.2-0.6)} \\
&= 1.388889(x-0.6)(x-1.2) - 2.292599(x-0.0)(x-1.2) \\
&\quad +0.503275(x-0.0)(x-0.6).
\end{aligned}
$$

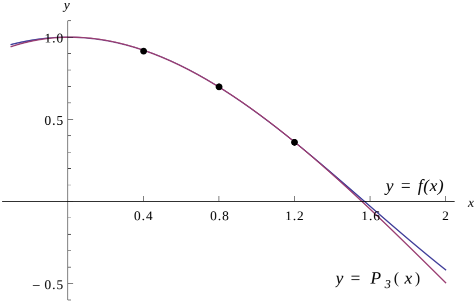
(**b**) Using $x_0 = 0.0, x_1 = 0.4, x_2 = 0.8, x_3 = 1.2$ and $y_0 = cos(0.0) = 1.0, y_1 = cos(0.4) = 0.921061, y_2 = cos(0.8) = 0.696707$, and $y_3 = cos(1.2) = 0.362358$ in equation (13) produces

$$\begin{aligned}
P_3(x) &= 1.000000\frac{(x-0.4)(x-0.8)(x-1.2)}{(0.0-0.4)(0.0-0.8)(0.0-1.2)} \\
&\quad +0.921061\frac{(x-0.0)(x-0.8)(x-1.2)}{(0.4-0.0)(0.4-0.8)(0.4-1.2)} \\
&\quad +0.696707\frac{(x-0.0)(x-0.4)(x-1.2)}{(0.8-0.0)(0.8-0.4)(0.8-1.2)} \\
&\quad +0.362358\frac{(x-0.0)(x-0.4)(x-0.8)}{(1.2-0.0)(1.2-0.4)(1.2-0.8)} \\
&= -2.604167(x-0.4)(x-0.8)(x-1.2) \\
&\quad +7.195789(x-0.0)(x-0.8)(x-1.2) \\
&\quad -5.443021(x-0.0)(x-0.4)(x-1.2) \\
&\quad +0.943641(x-0.0)(x-0.4)(x-0.8).
\end{aligned}$$

Lagrange Approximation



The quadratic approximation polynomial $y = P_2(x)$ based on the nodes $x_0 = 0.0, x_1 = 0.6$ and $x_2 = 1.2$.



The cubic approximation polynomial $y = P_3(x)$ based on the nodes $x_0 = 0.0, x_1 = 0.4, x_2 = 0.8$ and $x_3 = 1.2$.

Theorem 4.3 (Lagrange Polynomial Approximation)

Assume that $f \in C^{N+1}[a,b]$ and that $x_0, x_1, ..., x_N \in [a,b]$ are $N+1$ nodes. If $x \in [a,b]$, then

$$f(x) = P_N(x) + E_N(x), \tag{41}$$

where P_N is a polynomial that can be used to approximate $f(x)$:

$$f(x) \approx P_N(x) = \sum_{k=0}^{N} f(x_k) L_{N,k}(x). \tag{42}$$

The error term $E_N(x)$ has the form

$$E_N(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_N)f^{(N+1)}(c)}{(N+1)!} \tag{43}$$

for some value $c = c(x)$ that lies in the interval $[a,b]$.

Lagrange Approximation

Theorem 4.4 (Error Bounds for Lagrange Interpolation, Equally Spaced Nodes)

Assume that $f(x)$ is defined on $[a, b]$, which contains equally spaced nodes $x_k = x_0 + hk$. Additionally, assume that $f(x)$ and the derivates of $f(x)$, up to the order $N+1$, are continuous and bounded on the special subintervals $[x_0, x_1]$, $[x_0, x_2]$, and $[x_0, x_3]$, respectively; that is,

$$|f^{(N+1)}(x)| \leq M_{N+1} \quad for \;\; x_0 \leq x \leq x_N, \tag{44}$$

for $N = 1, 2, 3$. The error terms (16) corresponding to the cases $N = 1, 2$, and 3 have the following useful bounds on their magnitude:

$$|E_1(x)| \leq \frac{h^2 M_2}{8} \quad valid \;\; for \;\; x \in [x_0, x_1], \tag{45}$$

$$|E_2(x)| \leq \frac{h^3 M_3}{9\sqrt{3}} \quad valid \;\; for \;\; x \in [x_0, x_2], \tag{46}$$

$$|E_3(x)| \leq \frac{h^4 M_4}{24} \quad valid \;\; for \;\; x \in [x_0, x_3]. \tag{47}$$

Comparison of Accuracy and $O(h^{N+1})$

The significance of Theorem 4.4 is to understand a simple relationship between the size of the error terms for linear, quadratic, and cubic interpolation. In each case the error bound $|E_N(x)|$ depends on h in two ways. First h^{N+1} is explicitly present to that $|E_N(x)|$ is proportional h^{N+1}. Second, the values M_{N+1} generally depend on h and tend to $|f^{(N+1)}(x_0)|$ as h goes to zero. Therefore, as h goes to zero, $|E_N(x)|$ converges to zero with the same rapidity that h^{N+1} converges to zero. The notation $O(h^{N+1})$ is used when discussing this behavior. For example, the error bound (18) can expressed as

$$|E_1(x)| = O(h^2) \quad valid\ for\ x \in [x_0, x_1]$$

The notation $O(h^2)$ stands in place of $h^2M_2/8$ in relation (18) and is meant to convey the idea that the bound for the error term is approximately a multiple of h^2; that is,

$$|E_1(x)| \leq Ch^2 \approx O(h^2).$$

As a consequence, if the derivatives of $f(x)$ are uniformly bounded on the interval $[a, b]$ and $|h| < 1$, the choosing N large will make h^{N+1} small, and the higher-degree approximating polynomial will have less error.

Lagrange Approximation

Example 4.8: Consider $y = f(x) = cos(x)$ over $[0.0, 1.2]$. Use formula (18) through (20) and determine the error bounds for the Lagrange polynomials $P_1(x), P_2(x)$, and $P_3(x)$ that were constructed in Examples 4.6 and 4.7.

Example 4.8: Consider $y = f(x) = cos(x)$ over $[0.0, 1.2]$. Use formula (18) through (20) and determine the error bounds for the Lagrange polynomials $P_1(x), P_2(x)$, and $P_3(x)$ that were constructed in Examples 4.6 and 4.7. First, determine the bounds M_2, M_3, and M_4 for the

derivatives $|f^{(2)}(x)|, |f^{(3)}(x)|$ and $|f^{(4)}(x)|$, respectively, taken over the interval $[0.0, 1.2]$:

$$\begin{aligned} |f^{(2)}(x)| &= |-cos(x)| \leq |-cos(0.0)| = 1.000000 = M_2, \\ |f^{(3)}(x)| &= |sin(x)| \leq |sin(1.2)| = 0.932039 = M_3, \\ |f^{(4)}(x)| &= |cos(x)| \leq |cos(0.0)| = 1.000000 = M_4. \end{aligned}$$

For $P_1(x)$ the spacing of the nodes is $h = 1.2$, and its error bound is

$$|E_1(x)| \leq \frac{h^2 M_2}{8} \leq \frac{(1.2)^2(1.000000)}{8} = 0.180000 \tag{48}$$

Lagrange Approximation

For $P_2(x)$ the spacing of the nodes is $h = 0.6$, and its error bound is

$$|E_2(x)| \leq \frac{h^3 M_3}{9\sqrt{3}} \leq \frac{(0.6)^3(0.932039)}{9\sqrt{3}} = 0.012915 \tag{49}$$

For $P_3(x)$ the spacing of the nodes is $h = 0.4$, and its error bound is

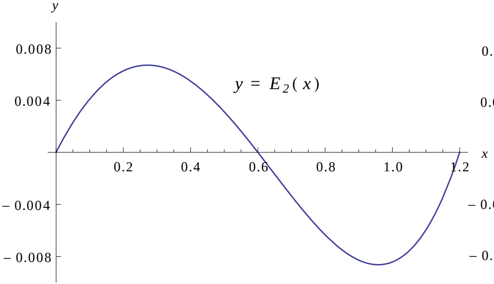
$$|E_3(x)| \leq \frac{h^4 M_4}{24} \leq \frac{(0.4)^4(1.000000)}{24} = 0.001067 \tag{50}$$

From Example 4.6 we saw that
$|E_1(0.6)| = |cos(0.6) - P_1(0.6)| = 0.144157$, so the bound 0.180000 in (21) is reasonable.

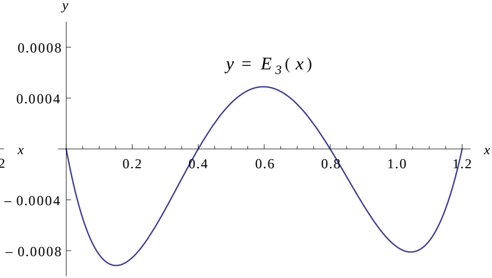
Table 4.7 Comparison of $f(x) = cos(x)$ and the Quadratic and Cubic Polynomial Approximations $P_2(x)$ and $P_3(x)$

x_k	$f(x_k) = cos(x_k)$	$P_2(x_k)$	$E_2(x_k)$	$P_3(x_k)$	$E_3(x_k)$
0.0	1.000000	1.000000	0.000000	1.000000	0.000000
0.1	0.995004	0.990911	0.004093	0.995835	−0.000831
0.2	0.980067	0.973813	0.006253	0.980921	−0.000855
0.3	0.955336	0.948707	0.006629	0.955812	−0.000476
0.4	0.921061	0.915592	0.005469	0.921061	0.000000
0.5	0.877583	0.874468	0.003114	0.877221	0.000361
0.6	0.825336	0.825336	0.000000	0.824847	0.000890
0.7	0.764842	0.768194	−0.003352	0.764491	0.000351
0.8	0.696707	0.703044	−0.006338	0.696707	0.000000
0.9	0.621610	0.629886	−0.008276	0.622048	−0.000438
1.0	0.540302	0.548719	−0.008416	0.541068	−0.000765
1.1	0.453596	0.459542	−0.005946	0.454320	−0.000724
1.2	0.362358	0.362358	0.000000	0.362358	0.000000

Lagrange Approximation



The error function $E_2(x) = cos(x) - P_2(x)$.



The error function $E_3(x) = cos(x) - P_3(x)$.

Newton Polynomials

We take a new approach and construct Newton polynomials that have the recursive pattern

$$
\begin{aligned}
P_1(x) &= a_0 + a_1(x - x_0), & (1)\\
P_2(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1), & (2)\\
P_3(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) & (3)\\
&\quad + a_3(x - x_0)(x - x_1)(x - x_2), &\\
&\ \vdots &\\
P_N(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) & (4)\\
&\quad + a_3(x - x_0)(x - x_1)(x - x_2) &\\
&\quad + a_4(x - x_0)(x - x_1)(x - x_2)(x - x_3) + \cdots &\\
&\quad + a_N(x - x_0)\cdots(x - x_{N-1}). &
\end{aligned}
$$

Here the polynomial $P_N(x)$ is obtained from $P_{N-1}(x)$ using the recursive relationship

$$P_N(x) = P_{N-1}(x) + a_N(x - x_0)(x - x_1)(x - x_2)\cdots(x - x_{N-1}). \tag{5}$$

The polynomial (4) is said to be a Newton polynomial with N **centers** $x_0, x_1, ..., x_{N-1}$. It involves sums of products of linear factors up to

$$a_N(x - x_0)(x - x_1)(x - x_2)\cdots(x - x_{N-1}),$$

so P_N will simply be an ordinary polynomial of degree $\leq N$.

Newton Polynomials

Example 4.10: Given the centers $x_0 = 1, x_1 = 3, x_2 = 4$, and $x_3 = 4.5$ and the coefficients $a_0 = 5, a_1 = -2, a_2 = 0.5, a_3 = -0.1$ and $a_4 = 0.003$, find $P_1(x), P_2(x), P_3(x), P_4(x)$ and evaluate $P_k(2.5)$ for $k = 1, 2, 3, 4$.

Example 4.10: Given the centers $x_0 = 1, x_1 = 3, x_2 = 4$, and $x_3 = 4.5$ and the coefficients $a_0 = 5, a_1 = -2, a_2 = 0.5, a_3 = -0.1$ and $a_4 = 0.003$, find $P_1(x), P_2(x), P_3(x), P_4(x)$ and evaluate $P_k(2.5)$ for $k = 1, 2, 3, 4$.

Using formulas (1) through (4), we have

$$\begin{aligned} P_1(x) &= 5 - 2(x-1), \\ P_2(x) &= 5 - 2(x-1) + 0.5(x-1)(x-3), \\ P_3(x) &= P_2(x) - 0.1(x-1)(x-3)(x-4), \\ P_4(x) &= P_3(x) + 0.003(x-1)(x-3)(x-4)(x-4.5). \end{aligned}$$

Newton Polynomials

Example 4.10: Given the centers $x_0 = 1, x_1 = 3, x_2 = 4$, and $x_3 = 4.5$ and the coefficients $a_0 = 5, a_1 = -2, a_2 = 0.5, a_3 = -0.1$ and $a_4 = 0.003$, find $P_1(x), P_2(x), P_3(x), P_4(x)$ and evaluate $P_k(2.5)$ for $k = 1, 2, 3, 4$.

Using formulas (1) through (4), we have

$$
\begin{aligned}
P_1(x) &= 5 - 2(x-1), \\
P_2(x) &= 5 - 2(x-1) + 0.5(x-1)(x-3), \\
P_3(x) &= P_2(x) - 0.1(x-1)(x-3)(x-4), \\
P_4(x) &= P_3(x) + 0.003(x-1)(x-3)(x-4)(x-4.5).
\end{aligned}
$$

Evaluating the polynomials at $x = 2.5$ results in

Example 4.10: Given the centers $x_0 = 1, x_1 = 3, x_2 = 4$, and $x_3 = 4.5$ and the coefficients $a_0 = 5, a_1 = -2, a_2 = 0.5, a_3 = -0.1$ and $a_4 = 0.003$, find $P_1(x), P_2(x), P_3(x), P_4(x)$ and evaluate $P_k(2.5)$ for $k = 1, 2, 3, 4$.

Using formulas (1) through (4), we have

$$\begin{aligned}
P_1(x) &= 5 - 2(x-1), \\
P_2(x) &= 5 - 2(x-1) + 0.5(x-1)(x-3), \\
P_3(x) &= P_2(x) - 0.1(x-1)(x-3)(x-4), \\
P_4(x) &= P_3(x) + 0.003(x-1)(x-3)(x-4)(x-4.5).
\end{aligned}$$

Evaluating the polynomials at $x = 2.5$ results in

$$\begin{aligned}
P_1(2.5) &= 5 - 2(1.5) = 2, \\
P_2(2.5) &= P_1(2.5) + 0.5(1.5)(-0.5) = 1.625, \\
P_3(2.5) &= P_2(2.5) - 0.1(1.5)(-0.5)(-1.5) = 1.5125, \\
P_4(2.5) &= P_3(2.5) + 0.003(1.5)(-0.5)(-1.5)(-2.0) = 1.50575.
\end{aligned}$$

Nested Multiplication

If N is fixed and the polynomial $P_N(x)$ is evaluated many times, then nested multiplication should be used. The process is similar to nested multiplication for ordinary polynomials, except that the centers x_k must be subtracted from the independent variable x. The nested multiplication form for $P_3(x)$ is

$$P_3(x) = ((a_3(x - x_2) + a_2)(x - x_1) + a_1)(x - x_0) + a_0. \tag{6}$$

To evaluate $P_3(x)$ for a given value of x, start with innermost grouping and form successively the quantities

$$\begin{aligned} S_3 &= a_3, \\ S_2 &= S_3(x - x_2) + a_2, \\ S_1 &= S_2(x - x_1) + a_1, \\ S_0 &= S_1(x - x_0) + a_0. \end{aligned} \tag{7}$$

The quantity S_0 is now $P_3(x)$.

Example 4.11: Compute $P_3(2.5)$ in Example 4.10 using nested multiplication.

Example 4.11: Compute $P_3(2.5)$ in Example 4.10 using nested multiplication.

Using (6), we write

$$P_3(x) = ((-0.1(x-4)+0.5)(x-3)-2)(x-1)+5.$$

The values in (7) are

$$\begin{aligned} S_3 &= -0.1, \\ S_2 &= -0.1(2.5-4)+0.5 = 0.65, \\ S_1 &= 0.65(2.5-3)-2 = -2.325, \\ S_0 &= -2.325(2.5-1)+5 = 1.5125. \end{aligned}$$

Therefore, $P_3(2.5) = 1.5125$.

Polynomial Approximation, Nodes, and Centers

Suppose that we want to find the coefficients a_k for all the polynomials $P_1(x), ..., P_N(x)$ that approximate a given function $f(x)$. Then $P_k(x)$ will be based on the centers $x_0, x_1, ..., x_k$ and have the nodes $x_0, x_1, ..., x_{k+1}$. For the polynomial $P_1(x)$ the coefficients a_0 and a_1 have a familiar meaning. In this case

$$P_1(x_0) = f(x_0) \quad \textit{and} \quad P_1(x_1) = f(x_1). \tag{8}$$

Using (1) and (8) to solve for a_0, we find that

$$f(x_0) = P_1(x_0) = a_0 + a_1(x_0 - x_0) = a_0. \tag{9}$$

Hence $a_0 = f(x_0)$. Next, using (1), (8), and (9), we have

$$f(x_1) = P_1(x_1) = a_0 + a_1(x_1 - x_0) = f(x_0) + a_1(x_1 - x_0),$$

which can be solved for a_1, and we get

$$a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \tag{10}$$

Hence a_1 is the slope of the secant line passing through the two points $(x_0, f(x_0))$ and $(x_1, f(x_1))$.
The coefficients a_0 and a_1 are the same for both $P_1(x)$ and $P_2(x)$. Evaluating (2) at the node x_2, we find that

$$f(x_2) = P_2(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1). \tag{11}$$

The values for a_0 and a_1 in (9) and (10) can be used in (11) to obtain

$$\begin{aligned} a_2 &= \frac{f(x_2) - a_0 - a_1(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)} \\ &= \left(\frac{f(x_2) - f(x_0)}{x_2 - x_0} - \frac{f(x_1) - f(x_0)}{x_1 - x_0} \right) / (x_2 - x_1). \end{aligned}$$

For computational purposes we prefer to write this last quantity as

$$a_2 = \left(\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0} \right) / (x_2 - x_0). \tag{12}$$

Definition 4.1 (Divided differences)

The **divided differences** for a function $f(x)$ are defined as follows:

$$
\begin{aligned}
f[x_k] &= f(x_k), \\
f[x_{k-1}, x_k] &= \frac{f[x_k] - f[x_{k-1}]}{x_k - x_{k-1}}, \\
f[x_{k-2}, x_{k-1}, x_k] &= \frac{f[x_{k-1}, x_k] - f[x_{k-2}, x_{k-1}]}{x_k - x_{k-2}}, \qquad (13) \\
f[x_{k-3}, x_{k-2}, x_{k-1}, x_k] &= \frac{f[x_{k-2}, x_{k-1}, x_k] - f[x_{k-3}, x_{k-2}, x_{k-1}]}{x_k - x_{k-3}}.
\end{aligned}
$$

(14)

The recursive rule for constructing higher-order divided differences is

$$f[x_{k-j}, x_{k-j+1}, \ldots, x_k] = \frac{f[x_{k-j+1}, \ldots, x_k] - f[x_{k-j}, \ldots, x_{k-1}]}{x_k - x_{k-j}} \qquad (15)$$

Newton Polynomials

Table 4.8 Divided-Difference Table for $y = f(x)$

x_k	$f[x_k]$	$f[\ ,\ ]$	$f[\ ,\ ,\ ]$	$f[\ ,\ ,\ ,\ ]$	$f[\ ,\ ,\ ,\ ,\ ]$
x_0	$f[x_0]$				
x_1	$f[x_1]$	$f[x_0, x_1]$			
x_2	$f[x_2]$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$		
x_3	$f[x_3]$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3]$	
x_4	$f[x_4]$	$f[x_3, x_4]$	$f[x_2, x_3, x_4]$	$f[x_1, x_2, x_3, x_4]$	$f[x_0, x_1, x_2, x_3, x_4]$

The coefficients a_k of $P_n(x)$ depend on the values $f(x_j)$, for $j = 0, 1, ..., k$.

Newton Polynomials

Theorem 4.5 (Newton Polynomial)

Suppose that $x_0, x_1, ..., x_N$ are $N+1$ distinct numbers in $[a, b]$. There exists a unique polynomial $P_N(x)$ of degree at most N with the property that

$$f(x_j) = P_N(x_j) \quad for \ j = 0, 1, ..., N$$

The Newton form of this polynomial is

$$P_N(x) = a_0 + a_1(x - x_0) + \cdots + a_N(x - x_0)(x - x_1)\cdots(x - x_{N-1}), \qquad (16)$$

where $a_k = f[x_0, x_1, ..., x_k]$, for $k = 0, 1, ..., N$.

Remark. If $\{(x_j, y_j)\}_{j=0}^{N}$ is a set of points whose abscissas are distinct, the values $f(x_j) = y_j$ can be used to construct the unique polynomial of degree $\leq N$ that passes through the $N+1$ points.

Corollary 4.2 (Newton Approximation)

Assume that $P_N(x)$ is the Newton polynomial given in Theorem 4.5 and is used to approximate the function $f(x)$, that is,

$$f(x) = P_N(x) + E_N(x). \tag{17}$$

If $f \in C^{N+1}[a, b]$, then for each $x \in [a, b]$ there corresponds a number $c = c(x)$ in (a, b), so that the error term has the form

$$E_N(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_N) f^{(N+1)}(c)}{(N+1)!}. \tag{18}$$

Remark. The error term $E_N(x)$ is the same as the one for Lagrange interpolation, which was introduced in equation (16) of Section 4.3.

Example 4.12. Let $f(x) = x^3 - 4x$. Construct the divided-difference table based on the nodes $x_0 = 1, x_1 = 2, ..., x_5 = 6$, and find the Newton polynomial $P_3(x)$ based on x_0, x_1, x_2, and x_3.

Example 4.12. Let $f(x) = x^3 - 4x$. Construct the divided-difference table based on the nodes $x_0 = 1, x_1 = 2, ..., x_5 = 6$, and find the Newton polynomial $P_3(x)$ based on x_0, x_1, x_2, and x_3.

Table 4.9 Divided-Difference Table Used for Constructing the Newton Polynomial $P_3(x)$

x_k	$f[x_k]$	First divided difference	Second divided difference	Third divided difference	Fourth divided difference	Fifth divided difference
$x_0 = 1$	$\underline{-3}$					
$x_1 = 2$	0	$\underline{3}$				
$x_2 = 3$	15	15	$\underline{6}$			
$x_3 = 4$	48	33	9	$\underline{1}$		
$x_4 = 5$	105	57	12	1	$\underline{0}$	
$x_5 = 6$	192	87	15	1	0	$\underline{0}$

The coefficients $a_0 = -3, a_1 = 3, a_2 = 6$, and $a_3 = 1$ of $P_3(x)$ appear on the diagonal of the divided-difference table. The centers $x_0 = 1, x_1 = 2$, and $x_2 = 3$ are the values in the first column. Using formula (3),we write

$$P_3(x) = -3 + 3(x - 1) + 6(x - 1)(x - 2) + (x - 1)(x - 2)(x - 3).$$

Newton Polynomials

Example 4.13. Construct a divided-difference table for $f(x) = cos(x)$ based on the five points $(k, cos(k))$, for $k = 0, 1, 2, 3, 4$. Use it to find the coefficients a_k and the four Newton interpolating polynomials $P_k(x)$, for $k = 1, 2, 3, 4$.

Example 4.13. Construct a divided-difference table for $f(x) = cos(x)$ based on the five points $(k, cos(k))$, for $k = 0, 1, 2, 3, 4$. Use it to find the coefficients a_k and the four Newton interpolating polynomials $P_k(x)$, for $k = 1, 2, 3, 4$.

For simplicity we round off the values to seven decimal places, which are displayed in Table 4.10. The nodes x_0, x_1, x_2, x_3 and the diagonal elements a_0, a_1, a_2, a_3, a_4 in Table 4.10 are used in formula (16), and we write down the first four Newton polynomials

$$
\begin{aligned}
P_1(x) &= 1.0000000 - 0.4596977(x - 0.0), \\
P_2(x) &= 1.0000000 - 0.4596977(x - 0.0) - 0.2483757(x - 0.0)(x - 1.0), \\
P_3(x) &= 1.0000000 - 0.4596977(x - 0.0) - 0.2483757(x - 0.0)(x - 1.0) \\
&\quad +0.1465592(x - 0.0)(x - 1.0)(x - 2.0), \\
P_4(x) &= 1.0000000 - 0.4596977(x - 0.0) - 0.2483757(x - 0.0)(x - 1.0), \\
&\quad +0.1465592(x - 0.0)(x - 1.0)(x - 2.0) \\
&\quad -0.0146568(x - 0.0)(x - 1.0)(x - 2.0)(x - 3.0)
\end{aligned}
$$

Newton Polynomials

Table 4.10 Divided-Difference Table Used for Constructing the Newton Polynomials $P_k(x)$

x_k	$f[x_k]$	$f[\ ,\ ]$	$f[\ ,\ ,\ ]$	$f[\ ,\ ,\ ,\ ]$	$f[\ ,\ ,\ ,\ ,\ ]$
$x_0 = 0.0$	1.0000000				
$x_1 = 1.0$	0.5403023	−0.4596977			
$x_2 = 2.0$	−0.4161468	−0.9564491	−0.2483757		
$x_3 = 3.0$	−0.9899925	−0.5738457	0.1913017	0.1465592	
$x_4 = 4.0$	−0.6536436	0.3363499	0.4550973	0.0879318	−0.0146568

The following sample calculation shows how to find the coefficient a_2.

$$\begin{aligned}
f[x_0, x_1] &= \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{0.5403023 - 1.0000000}{1.0 - 0.0} = -0.4596977, \\
f[x_1, x_2] &= \frac{f[x_2] - f[x_1]}{x_2 - x_1} = \frac{-0.4161468 - 0.5403023}{2.0 - 1.0} = -0.9564491, \\
a_2 = f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{-0.9564491 + 0.4596977}{2.0 - 0.0} = -0.2483757.
\end{aligned}$$

Newton Polynomials

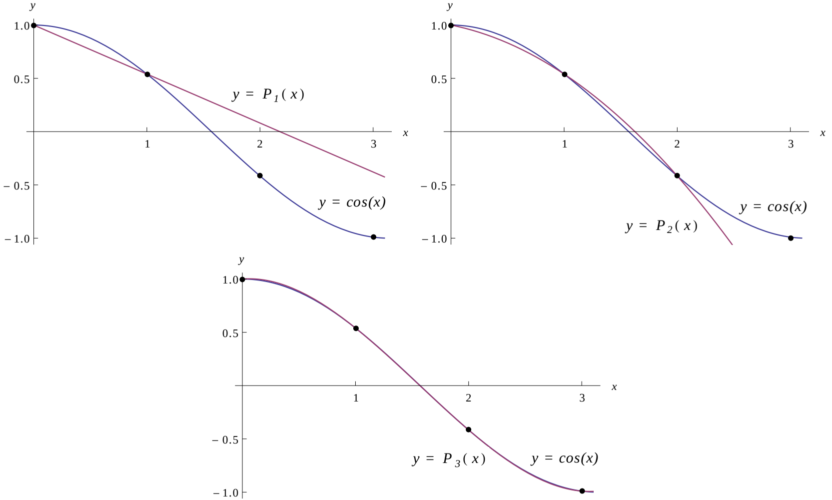


Figure: Newton polynomials $P_k(x)$ for $k = 1, 2$ and 3