Interpolation and Polynomial Approximation

Peng Yu

Tel: 0755 8801 8911
Email: yup6@sustech.edu.cn

Introduction

- Why polynomial approximation?

The computational procedures used in computer software for the evaluation of a library function, such as sin*(x)*, cos*(x)*, or e^x , involve polynomial approximation.

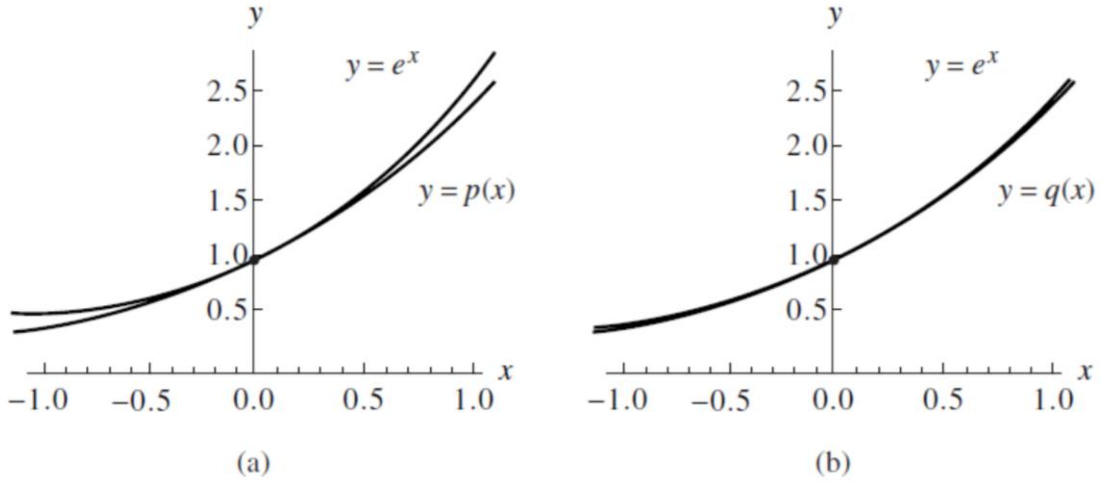


Figure 4.1 (a) The Taylor polynomial $p(x) = 1.000000 + 1.000000x + 0.500000x^2$, which approximates $f(x) = e^x$ over [-1, 1]. (b) The Chebyshev approximation $q(x) = 1.000000 + 1.129772x + 0.532042x^2$ for $f(x) = e^x$ over[-1, 1].

Introduction

- Why interpolation?

Given $n + 1$ points in the plane (no two of which are aligned vertically), the collocation polynomial is the unique polynomial of degree $\leq n$ that passes through the points. In cases where data are known to a high degree of precision, the collocation polynomial is sometimes used to find a polynomial that passes through the given data points.

A variety of methods can be used to construct the collocation polynomial:

solving a linear system for its coefficients, the use of Lagrange coefficient polynomials, and the construction of a divided differences table and the coefficients of the Newton polynomial.

Interpolation and Polynomial Approximation

- Taylor Series and Calculation of Functions
- Introduction to Interpolation
- Lagrange Approximation
- Newton Polynomials
- Chebyshev Polynomials
- Pade Approximations

Taylor Series and Calculation of Functions

Taylor Expansion

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

Taylor Polynomial Approximation

- A finite sum can be used to obtain a good approximation to an infinite sum.
- If enough terms are added, then an accurate approximation will be obtained

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 Polynomial Approximation

Theorem (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

(1) $$f(x) = P_N(x) + E_N(x),$$

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

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

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

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

for some value $c = c(x)$ that lies between x and x_0.

***Error term* can be used to determine a bound for the accuracy of the approximation**

An example

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

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 $f'(x) = f''(x) = \cdots = f^{(16)} = e^x$. The first 15 derivatives are used to calculate the coefficients $a_k = e^0/k!$ and are used to write

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

Setting $x = 1$ in (4) gives the partial sum $S_{15} = P_{15}(1)$. The remainder term is needed to show the accuracy of the approximation:

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

Since we choose $x_0 = 0$ and $x = 1$, the value c lies between them (i.e., $0 < c < 1$), which implies that $e^c < e^1$. Notice that the partial sums are bounded above by 3. So $e^c < 3$,

$$|E_{15}(1)| = \frac{|f^{(16)}(c)|}{16!} \leq \frac{e^c}{16!} < \frac{3}{16!} < 1.433844 \times 10^{-13}.$$

Taylor Polynomial Approximation

- Matches the leading derivatives at x_0,

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

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

- A local approximation, bad if x is away from x_0

Taylor Polynomial Approximation

- 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.

- If we choose the interval width to be $2R$ and x_0 in the center (i.e., $|x - x_0| < R$), the absolute value of the error satisfies the relation.

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

where $M \leq \max\{|f^{(N+1)}(z)| : x_0 - R \leq z \leq x_0 + R\}$.

Methods for Evaluating a Polynomial

- There are several mathematically equivalent ways to evaluate a polynomial, eg. $P(x)=a_0+a_1x+a_2x^2+a_3x^3+\ldots+a_Nx^N$

- To evaluate it directly, it takes $1+2+3+\ldots+N = N(N+1)/2 = O(N^2)$ multiplications.

- Horner's method, which is also called nested multiplication, uses N multiplications, $P(x)= a_0+x(a_1+x(a_2+x(a_3+\ldots+a_Nx)))$.

Taylor Polynomial Approximation

Theorem 4.2 (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

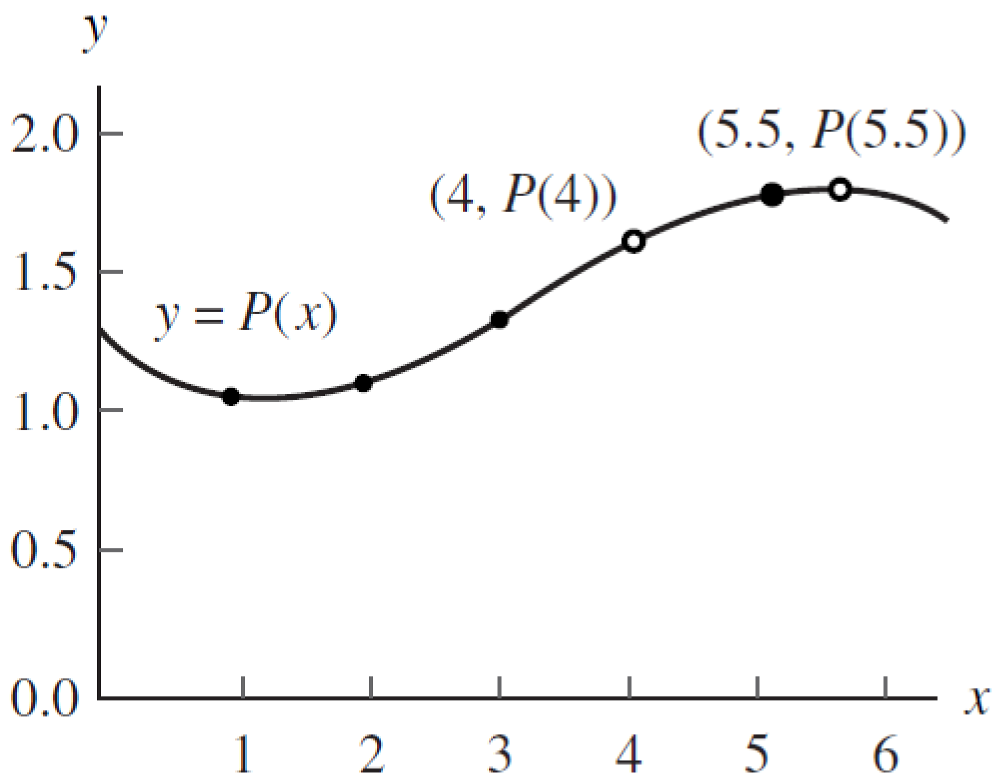
$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k. \tag{13}$$

Introduction to Interpolation

Why we need interpolation

- The information needed to construct the Taylor polynomial is the value of $f(x)$ and its derivatives at x_0.
- A shortcoming is that the higher-order derivatives must be known, and often they are either not available or they are hard to compute.
- For a function $y=f(x)$, suppose $N+1$ points are known, a polynomial $P(x)$ of degree N can be constructed that passes through the $N+1$ points. In the construction, only numerical values of x_k and y_k are needed.
- Situations in statistical and scientific analysis arise where the function $y = f(x)$ is available only at $N + 1$ tabulated points (x_k, y_k), and a method is needed to approximate $= f(x)$ at nontabulated abscissas.

Why we need interpolation



Interpolation

Extrapolation

The approximating polynomial $P(x)$ can be used for interpolation at the point $(4, P(4))$ and extrapolation at the point $(5.5, P(5.5))$.

How to do interpolation

- Solve a linear equation

- For example, for polynomial $P(x) = A + Bx + Cx^2 + Dx^3$, pass through (1, 1.06), (2, 1.12), (3, 1.34), (5, 1.78)

The methods of Chapter 3 can be used to find the coefficients. 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.

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

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

How to do interpolation

- Solve a linear equation

Let $P(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_N x^N$, or more compactly:

$$P(x) = \sum_{k=0}^{N} a_k x^k \,.$$

We need to find the $N + 1$ coefficients, $a_k, k = 0, 1, \ldots, N$.
Using the expression above, we have $N + 1$ linear equations about the $N + 1$ unknows, a_k:

$$\sum_{k=0}^{N} a_k x_i^k = P(x_i) = y_i, \qquad \text{for all points:} \quad i = 0, 1, \ldots, N.$$

Written in matrix form:

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^N \\ 1 & x_1 & x_1^2 & & x_1^N \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^N \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{pmatrix}.$$

The matrix is called **Vandermonde Matrix.**

Lagrange Approximation

Lagrange coefficient polynomials

- 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, (x_0, y_0) and (x_1, y_1)

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

- The French mathematician Joseph Louis Lagrange

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

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

Lagrange coefficient polynomials

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

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

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

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

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

An example

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 $P_3(x)$.

$$P_2(x) = 1.0\frac{(x-0.6)(x-1.2)}{(0.0-0.6)(0.0-1.2)} + 0.825336\frac{(x-0.0)(x-1.2)}{(0.6-0.0)(0.6-1.2)}$$

$$+0.360358\frac{(x-0.0)(x-0.6)}{(1.2-0.0)(1.2-0.6)}$$

$$= 1.388889(x-0.6)(x-1.2) - 2.292599(x-0.0)(x-1.2)$$

$$+0.503275(x-0.0)(x-0.6).$$

An example

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

An example

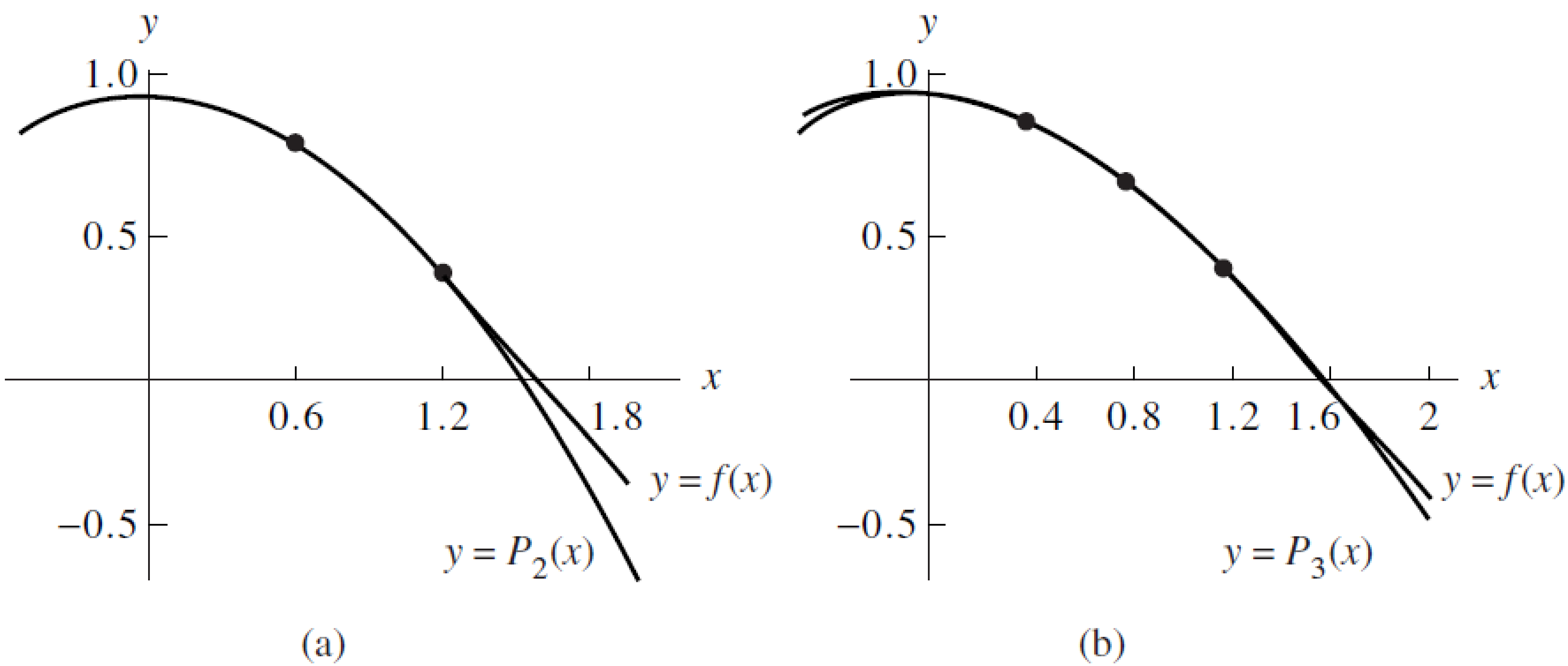


Figure 4.12 (a) The quadratic approximation polynomial $y = P_2(x)$ based on the nodes $x_0 = 0.0$, $x_1 = 0.6$, and $x_2 = 1.2$. (b) The cubic approximation polynomial $y = P_3(x)$ based on the nodes $x_0 = 0.0$, $x_1 = 0.4$, $x_2 = 0.8$, and $x_3 = 1.2$.

Error Terms and Error Bounds

- Similar to the error term for the Taylor polynomial

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

$$f(x) = P_N(x) + E_N(x),$$

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

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

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

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

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

Error Terms and Error Bounds

- For the special case when the nodes for the Lagrange polynomial are equally spaced $x_k = x_0 + hk$, for $k = 0, 1, \ldots, N$

Theorem (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 derivatives 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,

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

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

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

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

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

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

- The significance of previous Theorem 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, $|E_N(x)|$ is proportional to 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 $\boldsymbol{O}(h^{N+1})$ is used when discussing this behavior.
- For example, the error bound can be expressed as

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

- The notation $\boldsymbol{O}(h^2)$ is meant to convey the idea that the bound for the error term is approximately a multiple of h^2

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

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

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

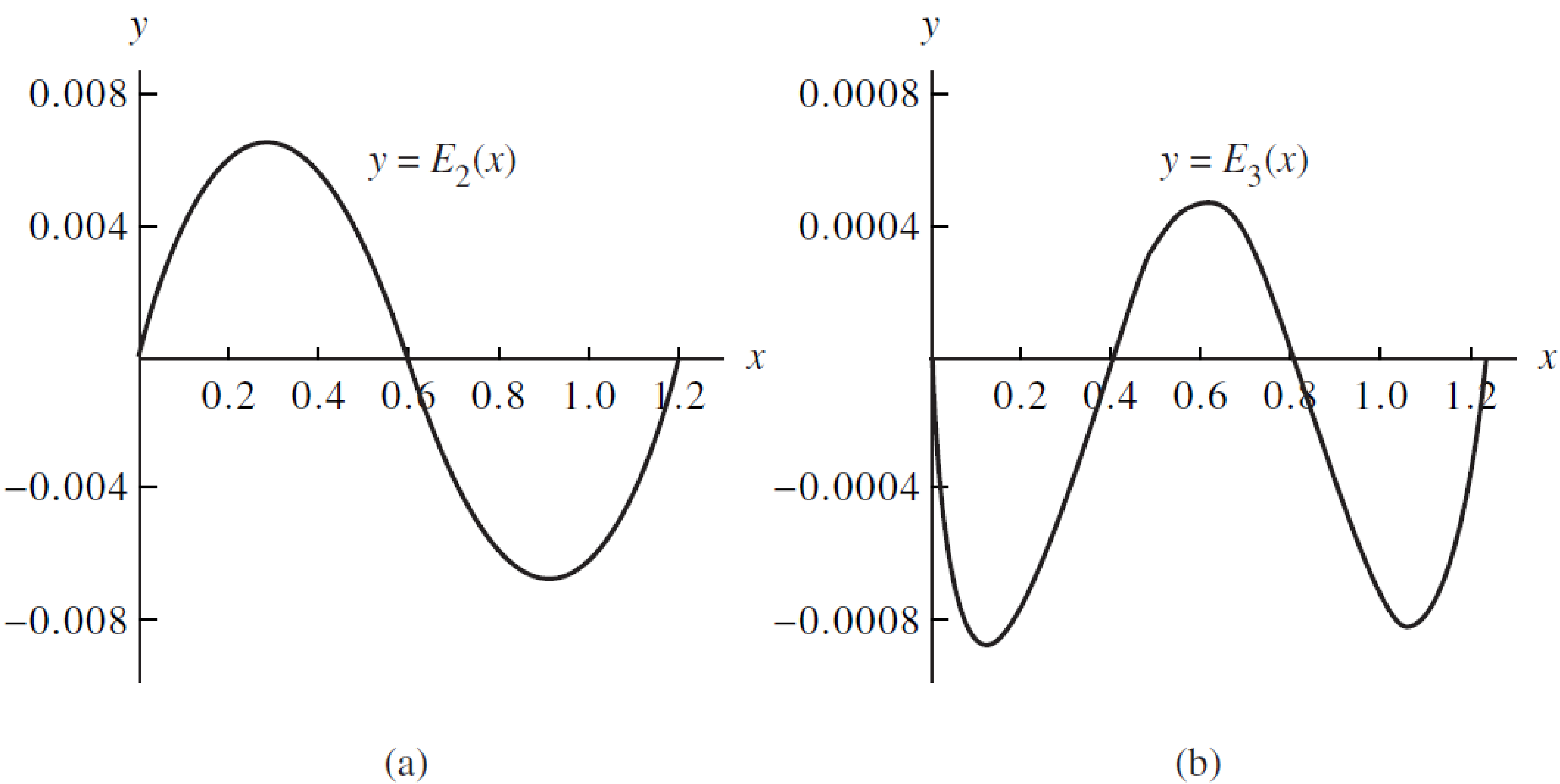


Figure 4.13 (a) The error function $E_2(x) = \cos(x) - P_2(x)$. (b) The error function $E_3(x) = \cos(x) - P_3(x)$.

Matlab Code

Program 4.1 (Lagrange Approximation). To evaluate the Lagrange polynomial $P(x) = \sum_{k=0}^{N} y_k L_{N,k}(x)$ based on $N + 1$ points (x_k, y_k) for $k = 0, 1, \ldots, N$.

```
function [C,L]=lagran(X,Y)

%Input  - X is a vector that contains a list of abscissas
%       - Y is a vector that contains a list of ordinates
%Output - C is a matrix that contains the coefficients of
%         the Lagrange interpolatory polynomial
%       - L is a matrix that contains the Lagrange
%         coefficient polynomials

w=length(X);
n=w-1;
L=zeros(w,w);

%Form the Lagrange coefficient polynomials
for k=1:n+1
    V=1;
    for j=1:n+1
    if k~=j
    V=conv(V,poly(X(j)))/(X(k)-X(j));
    end
end
    L(k,:)=V;
end

%Determine the coefficients of the Lagrange interpolating
%polynomial

C=Y*L;
```

The **conv** commands produces a vector whose entries are the coefficients of a polynomial that is the product of two other polynomials.

Newton Polynomials

Newton Polynomials

- If the Lagrange polynomials are used, there is no constructive relationship between $P_{N\text{-}1}(x)$ and $P_N\,(x)$.

- Each polynomial has to be constructed individually, and the work required to compute the higher-degree polynomials involves many computations.

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

(1) $$P_1(x) = a_0 + a_1(x - x_0),$$

(2) $$P_2(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1),$$

Newton Polynomials

(3) $$P_3(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) \\ + a_3(x - x_0)(x - x_1)(x - x_2),$$

$$\vdots$$

(4) $$P_N(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) \\ + a_3(x - x_0)(x - x_1)(x - x_2) \\ + a_4(x - x_0)(x - x_1)(x - x_2)(x - x_3) + \cdots \\ + a_N(x - x_0)\cdots(x - x_{N-1}).$$

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

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

The polynomial (4) is said to be a Newton polynomial with N ***centers*** $x_0, x_1, \ldots, x_{N-1}$. 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 $P_N(x)$ will simply be an ordinary polynomial of degree $\leq N$.

An example

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 $P_1(x), P_2(x), P_3(x),$ and $P_4(x)$ and evaluate $P_k(2.5)$ for $k = 1, 2, 3, 4$.

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

$$P_1(x) = 5 - 2(x - 1),$$
$$P_2(x) = 5 - 2(x - 1) + 0.5(x - 1)(x - 3),$$
$$P_3(x) = P_2(x) - 0.1(x - 1)(x - 3)(x - 4),$$
$$P_4(x) = P_3(x) + 0.003(x - 1)(x - 3)(x - 4)(x - 4.5).$$

Evaluating the polynomial at $x = 2.5$

$$P_1(2.5) = 5 - 2(1.5) = 2,$$
$$P_2(2.5) = P_1(2.5) + 0.5(1.5)(-0.5) = 1.625,$$
$$P_3(2.5) = P_2(2.5) - 0.1(1.5)(-0.5)(-1.5) = 1.5125,$$
$$P_4(2.5) = P_3(2.5) + 0.003(1.5)(-0.5)(-1.5)(-2.0) = 1.50575.$$

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.

- For example, the nested multiplication form for $P_3(x)$ is

$$P_3(x) = \big((a_3(x - x_2) + a_2)(x - x_1) + a_1\big)(x - x_0) + a_0.$$

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

Polynomial Approximation, Nodes, and Centers

- How 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 $x_0, x_1, \ldots, x_k$ and have the nodes $x_0, x_1, \ldots, x_{k+1}$.

- For the Polynomial $P_1(x) = a_0 + a_1(x - x_0)$

$$P_1(x_0) = f(x_0) \quad \text{and} \quad P_1(x_1) = f(x_1).$$

$$\Downarrow$$

$$f(x_0) = P_1(x_0) = a_0 + a_1(x_0 - x_0) = a_0.$$

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

$$\Downarrow$$

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

Polynomial Approximation, Nodes, and Centers

- For the Polynomial $P_2(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1)$

$$f(x_2) = P_2(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1).$$

$$\Downarrow$$

$$a_2 = \frac{f(x_2) - a_0 - a_1(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)}$$

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

$$\Downarrow$$

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

Polynomial Approximation, Nodes, and Centers

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

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

$$f[x_{k-1}, x_k] = \frac{f[x_k] - f[x_{k-1}]}{x_k - x_{k-1}},$$

$$f[x_{k-2}, x_{k-1}, x_k] = \frac{f[x_{k-1}, x_k] - f[x_{k-2}, x_{k-1}]}{x_k - x_{k-2}},$$

$$f[x_{k-3}, x_{k-2}, x_{k-1}, x_k] = \frac{f[x_{k-2}, x_{k-1}, x_k] - f[x_{k-3}, x_{k-2}, x_{k-1}]}{x_k - x_{k-3}}$$

- The recursive rule for constructing higher-order divided differences is

$$f\left[x_{k-j}, x_{k-j+1}, \ldots, x_k\right] = \frac{f\left[x_{k-j+1}, \ldots, x_k\right] - f\left[x_{k-j}, \ldots, x_{k-1}\right]}{x_k - x_{k-j}}$$

Polynomial Approximation, Nodes, and Centers

- The recursive rule for constructing higher-order divided differences is

$$f\left[x_{k-j}, x_{k-j+1}, \ldots, x_k\right] = \frac{f\left[x_{k-j+1},\ldots,x_k\right] - f\left[x_{k-j},\ldots,x_{k-1}\right]}{x_k - x_{k-j}}$$

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

Newton Polynomial

Theorem 4.5 (Newton Polynomial). Suppose that $x_0, x_1, \ldots, x_N$ 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

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

The Newton form of this polynomial is

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

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

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.

Newton Approximation

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,

(17) $$f(x) = P_N(x) + E_N(x).$$

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

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

Remark. The error term $E_N(x)$ is the same as the one for Lagrange interpolation.

Examples

Let $f(x) = x^3 - 4x$. Construct the divided-difference table based on the nodes $x_0 = 1, x_1 = 2, \ldots, x_5 = 6$, 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

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

Examples

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 interpolation polynomials $P_k(x)$, for $k = 1,\ 2,\ 3,\ 4$.

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

Examples

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

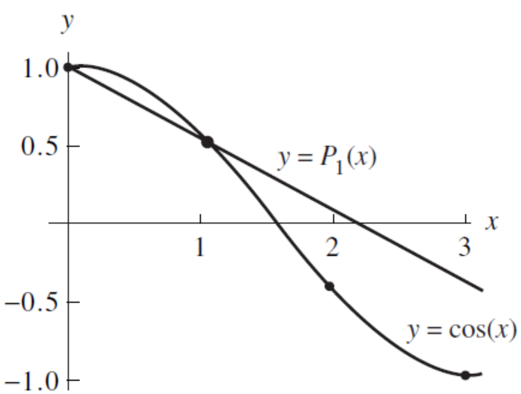


Figure 4.14 (a) The graphs of $y = \cos(x)$ and the linear Newton polynomial $y = P_1(x)$ based on the nodes $x_0 = 0.0$ and $x_1 = 1.0$.

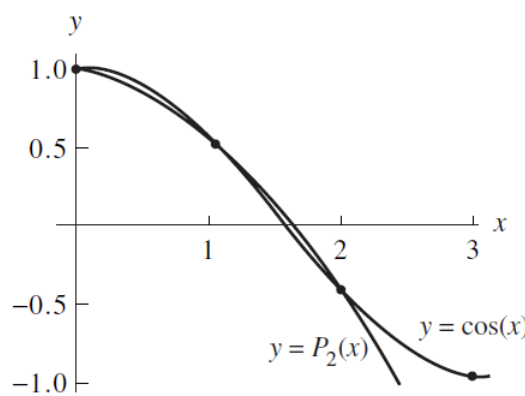


Figure 4.14 (b) The graphs of $y = \cos(x)$ and the quadratic Newton polynomial $y = P_2(x)$ based on the nodes $x_0 = 0.0$, $x_1 = 1.0$, and $x_2 = 2.0$.

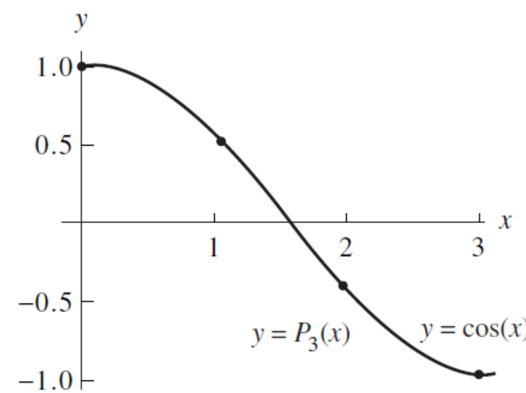


Figure 4.14 (c) The graphs of $y = \cos(x)$ and the cubic Newton polynomial $y = P_2(x)$ based on the nodes $x_0 = 0.0$, $x_1 = 1.0$, $x_2 = 2.0$, and $x_3 = 3.0$.

Matlab Code

Program 4.2 (Newton Interpolation Polynomial). To construct and evaluate the Newton polynomial of degree $\le N$ that passes through $(x_k, y_k) = (x_k, f(x_k))$ for $k = 0, 1, \ldots, N$:

$$P(x) = d_{0,0} + d_{1,1}(x - x_0) + d_{2,2}(x - x_0)(x - x_1) + \cdots + d_{N,N}(x - x_0)(x - x_1)\cdots(x - x_{N-1}), \tag{21}$$

where

$$d_{k,0} = y_k \quad \text{and} \quad d_{k,j} = \frac{d_{k,j-1} - d_{k-1,j-1}}{x_k - x_{k-j}}.$$

```
function [C,D]=newpoly(X,Y)
%Input  - X is a vector that contains a list of abscissas
%       - Y is a vector that contains a list of ordinates
%Output - C is a vector that contains the coefficients
%         of the Newton intepolatory polynomial
%       - D is the divided-difference table

n=length(X);
D=zeros(n,n);
D(:,1)=Y';

% Use formula (20) to form the divided-difference table

for j=2:n
    for k=j:n
      D(k,j)=(D(k,j-1)-D(k-1,j-1))/(X(k)-X(k-j+1));
   end
end

%Determine the coefficients of the Newton interpolating
%polynomial
C=D(n,n);
for k=(n-1):-1:1
   C=conv(C,poly(X(k)));
   m=length(C);
   C(m)=C(m)+D(k,k);
end
```

Chebyshev Polynomials

Why Chebyshev

- Consider polynomial interpolation for $f(x)$ over [-1, 1] based on the nodes $x_0 < x_1 < \ldots < x_N$. Both the Lagrange and Newton polynomials satisfy

$$f(x) = P_N(x) + E_N(x)$$

where

$$E_N(x) = Q(x)\frac{f^{(N+1)}(c)}{(N+1)!}$$

and $Q(x)$ is the polynomial of degree $N+1$:

$$Q(x) = (x - x_0)(x - x_1)\cdots(x - x_N).$$

Using the relationship

$$|E_N(x)| \leq |Q_N(x)|\frac{\max_{-1\leq x\leq 1}\left\{\left|f^{(N+1)}(x)\right|\right\}}{(N+1)!},$$

- How to select the set of nodes $\{x_k\}$ that minimizes

$$\max_{-1\leq x\leq 1}\{|Q_N(x)|\}$$

Classical Orthogonal Polynomials

- In function analysis, an inner product with respect to a weight function, $\omega(x)$, is defined as:

  $$\langle f, g \rangle \equiv \int \omega(x) f(x) \mathrm{g}(x)\, dx.$$

  Think the dot-product of two vectors.

- Two functions, f and g, are said to be orthogonal if $\langle f, \mathrm{g} \rangle = 0$. (two perpendicular, or orthogonal, vectors)

- Orthogonal polynomial sequence, $P_n(x), n = 0, \cdots, \infty$, is a family of polynomial satisfying:

  - P_n is an n-th degree polynomial;

  - Ortho-normal: $\langle P_m, P_n \rangle = \begin{cases} 0 & m \neq n \\ 1 & m = n \end{cases}$. $P_n(x)$ are similar to the base vectors **x, y, z** in three-dimensional Euclidean space.

  - $P_n(x)$ can be constructed explicitly through the **Gram-Schmidt** procedure.

Classical Orthogonal Polynomials

- A family of orthogonal polynomials form a complete basis of the given functional space, i.e. for any function $f(x)$:

$$f(x) = \sum_{n=0}^{\infty} a_n P_n(x)\,,$$

where a_n are the expansion coefficients given by

$$a_n = \int \omega(x) f(x) P_n(x)\, dx.$$

Any vector in Euclidean space can be expanded as $\mathbf{v} = v_x\mathbf{x} + v_y\mathbf{y} + v_z\mathbf{z}$ with $v_x = \mathbf{v}\cdot\mathbf{x}$, $v_y = \mathbf{v}\cdot\mathbf{y}, \cdots$. The expansion is called the $general\ Fourier\ expansion$.

Classical Orthogonal Polynomials

- Orthogonal polynomials arise from the **Sturm-Liuville** problem, they are solutions to the **Sturm-Liuville** equation:

$$\frac{d}{dx}\left[p(x)\frac{dy}{dx}\right] + q(x)y = -\lambda\omega(x)y,$$

and some boundary conditions.

- The most commonly used families are the **Classical Orthogonal Polynomials:**

$$\begin{array}{lll} Jacobi & \omega = (1-x)^{\alpha}(1+x)^{\beta} & [-1,+1] \\ Hermite & \omega = \exp(-x^2) & (-\infty,+\infty) \\ Laguerre & \omega = x^{\alpha}\exp(-x) & [0,\infty) \end{array}$$

Chebyshev Polynomials

- Chebyshev polynomial is a special form of Jacobi polynomial with $\alpha = \beta = 1/2$

Table 4.11 Chebyshev Polynomials $T_0(x)$ through $T_7(x)$

$T_0(x) = 1$
$T_1(x) = x$
$T_2(x) = 2x^2 - 1$
$T_3(x) = 4x^3 - 3x$
$T_4(x) = 8x^4 - 8x^2 + 1$
$T_5(x) = 16x^5 - 20x^3 + 5x$
$T_6(x) = 32x^6 - 48x^4 + 18x^2 - 1$
$T_7(x) = 64x^7 - 112x^5 + 56x^3 - 7x$

Properties of Chebyshev Polynomials

Property 1. Recurrence Relation
Chebyshev polynomials can be generated in the following way. Set $T_0(x) = 1$ and $T_1(x) = x$ and use the recurrence relation

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x) \qquad \text{for} \quad k = 2, 3, \ldots$$

Property 1 is often used as the definition for higher-order Chebyshev polynomials. Let us show that $T_3(x) = 2xT_2(x) - T_1(x)$. Using the expressions for $T_1(x)$ and $T_2(x)$ in Table 4.11, we obtain

$$2xT_2(x) - T_1(x) = 2x(2x^2 - 1) - x = 4x^3 - 3x = T_3(x).$$

Property 2. Leading Coefficient
The coefficient of x^N in $T_N(x)$ is 2^{N-1} when $N \geq 1$.

Property 2 is proved by observing that the recurrence relation doubles the leading coefficient of $T_{N-1}(x)$ to get the leading coefficient of $T_N(x)$.

Properties of Chebyshev Polynomials

Property 3. Symmetry

When $N = 2M$, $T_{2M}(x)$ is an even function, that is,

(4) $$T_{2M}(-x) = T_{2M}(x)$$

When $N = 2M + 1$, $T_{2M+1}(x)$ is an odd function, that is,

(5) $$T_{2M+1}(-x) = -T_{2M+1}(x).$$

Property 3 is established by showing that $T_{2M}(x)$ involves only even powers of x and $T_{2M+1}(x)$ involves only odd powers of x

Properties of Chebyshev Polynomials

Property 4. Trigonometric Representation on [-1,1]

$$T_N(x) = \cos\big(N\, arccos(x)\big) \qquad \text{for} -1 \leq x \leq 1.$$

$$\cos(k\theta) = \cos(2\theta)\cos\big((k-2)\theta\big) - \sin(2\theta)\sin\big((k-2)\theta\big).$$

$$\cos(k\theta) = 2\cos(\theta)\,(\cos(\theta)\cos(k-2)\theta) - \sin(\theta)\sin\big((k-2)\theta\big) - \cos((k-2)\theta).$$

$$\cos(k\theta) = 2\cos(\theta)\cos\big((k-1)\theta\big) - \cos\big((k-2)\theta\big).$$

Finally, substitute $\theta = \arccos(x)$ and obtain

$$2x\cos\big((k-1)\arccos(x)\big) - \cos\big((k-2)\arccos(x)\big)$$
$$= \cos\big(k\arccos(x)\big) \qquad \text{for} -1 \leq \text{x} \leq 1.$$

Properties of Chebyshev Polynomials

- The first two Chebyshev polynomials are

$$T_0(x) = \cos(0 \arccos(x)) = 1$$
$$T_1(x) = \cos(1 \arccos(x)) = x$$

- Now assume

$$T_k(x) = \cos(k \arccos(x)) \quad \text{for } k = 2, 3, \ldots, N\text{-}1$$

$$\begin{aligned} T_N(x) &= 2xT_{N-1}(x) - T_{N-2}(x) \\ &= 2x\cos((N-1)\arccos(x)) - \cos((N-2)\arccos(x)) \\ &= \cos(N\, arccos(x)) \qquad \text{for } -1 \le x \le 1. \end{aligned}$$

Properties of Chebyshev Polynomials

Property 5. Distinct Zeros in [−1, 1]
$T_N(x)$ has N distinct zeros x_k that lie in the interval [−1, 1] (see Figure):

$$x_k = \cos\left(\frac{(2k+1)\pi}{2N}\right) \qquad \text{for } k = 0, 1, \ldots, N-1.$$

These values are called the ***Chebyshev abscissas (nodes).***

Property 6. Extreme Values

$$|T_N(x)| \leq 1 \qquad \text{for } -1 \leq x \leq 1.$$

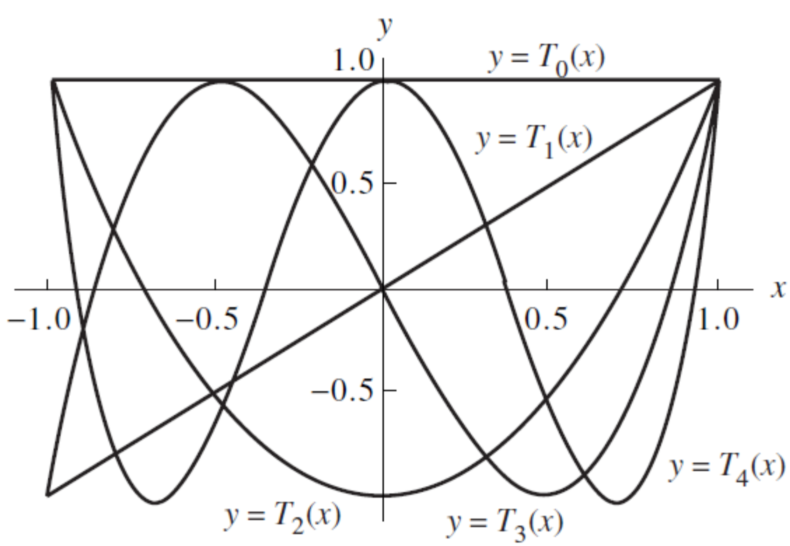


Figure 4.15 The graphs of the Chebyshev polynomials $T_0(x)$, $T_1(x)$, ..., $T_4(x)$ over [−1, 1].

Minimize the upper bound for error

The Russian mathematician Chebyshev studied how to minimize the upper bound for $|E_N(x)|$. One upper bound can be formed by taking the product of the maximum value of $|Q(x)|$ over all x in [−1, 1] and the maximum value $|f^{(N+1)}(x)/(N+1)!|$ over all x in [−1, 1]. To minimize the factor max$\{|Q(x)|\}$, Chebyshev discovered that $x_0, x_1, \ldots, x_N$ should be chosen so that $Q(x) = (1/2^N)T_{N+1}(x)$.

Theorem 4.6. Assume that ***N*** is fixed. Among all possible choice for $Q(x)$ in equation (2), and thus among all possible choices for the distinct nodes $\{x_k\}_{k=0}^{N}$ in [−1, 1], the polynomial $T(x) = T_{N+1}(x)/2^N$ is the unique choice that has the property

$$\max_{-1\le x\le 1}\{|T(x)|\} \le \max_{-1\le x\le 1}\{|Q(x)|\}.$$

Moreover,

$$\max_{-1\le x\le 1}\{|T(x)|\} = \frac{1}{2^N}.$$

Minimize the upper bound for error

- The consequence can be stated by saying that for Lagrange interpolation $f(x)$ on [-1,1], the minimum value of the error bound is achieved when the nodes $\{x_k\}$ are the Chebyshev abscissas of $T_{N+1}(x)$
- See an example, we look at the Lagrange coefficient polynomials that are used in forming $P_3(x)$, and compare using equally spaced nodes and the Chebyshev nodes.
- Recall that

$$P_3(x) = f(x_0)L_{3,0}(x) + f(x_1)L_{3,1}(x) + f(x_2)L_{3,2}(x) + f(x_3)L_{3,3}(x).$$

An Example: Equally spaced nodes

- If $f(x)$ is approximated by a polynomial of degree at most $N = 3$ on $[-1, 1]$, the equally spaced nodes $x_0 = -1$, $x_1 = -1/3$, $x_2 = 1/3$, and $x_3 = 1$ are easy to use for calculations.

Table 4.12 Lagrange Coefficient Polynomials Used to Form $P_3(x)$ Based on Equally Spaced Nodes $x_k = -1 + 2k/3$

$$
\begin{aligned}
L_{3,0}(x) &= -0.06250000 + 0.06250000x + 0.56250000x^2 - 0.56250000x^3 \\
L_{3,1}(x) &= 0.56250000 - 1.68750000x - 0.56250000x^2 + 1.68750000x^3 \\
L_{3,2}(x) &= 0.56250000 + 1.68750000x - 0.56250000x^2 - 1.68750000x^3 \\
L_{3,3}(x) &= -0.06250000 - 0.06250000x + 0.56250000x^2 + 0.56250000x^3
\end{aligned}
$$

An Example: Chebyshev nodes

- When $f(x)$ is to be approximated by a polynomial of degree at most $N = 3$, using the Chebyshev nodes $x_0 = \cos(7\pi/8)$, $x_1 = \cos(5\pi/8)$, $x2 = \cos(3\pi/8)$, and $x3 = \cos(\pi/8)$, the coefficient polynomials are tedious to find (but this can be done by a computer).

Table 4.13 Coefficient Polynomials Used to Form $P_3(x)$ Based on the Chebyshev Nodes $x_k = \cos((7-2k)\pi/8)$

$$
\begin{aligned}
C_0(x) &= -0.10355339 + 0.11208538x + 0.70710678x^2 - 0.76536686x^3 \\
C_1(x) &= 0.60355339 - 1.57716102x - 0.70710678x^2 + 1.84775906x^3 \\
C_2(x) &= 0.60355339 + 1.57716102x - 0.70710678x^2 - 1.84775906x^3 \\
C_3(x) &= -0.10355339 - 0.11208538x + 0.70710678x^2 + 0.76536686x^3
\end{aligned}
$$

An Example

- Compare the Lagrange polynomials of degree $N = 3$ for $f(x) = e^x$ that are obtained by using the coefficient polynomials in Tables 4.12 and 4.13, respectively:
- Using equally spaced nodes, we get

$$f(x_0) = e^{(-1)} = 0.36787944, \qquad f(x_1) = e^{(-1/3)} = 0.71653131,$$
$$f(x_2) = e^{(1/3)} = 1.39561243, \qquad f(x_3) = e^{(1)} = 2.71828183,$$

$$P(x) = 0.36787944L_{3,0}(x) + 0.71653131L_{3,1}(x) + 1.39561243L_{3,2}(x)$$
$$+2.71828183L_{3,0}(x).$$

$$P(x) = 0.99519577 + 0.99904923x + 0.54788486x^2 + 0.17615196x^3.$$

- Using Chebyshev nodes, we obtain

$$V(x) = 0.99461532 + 0.99893323x + 0.54290072x^2 + 0.17517569x^3.$$

An Example

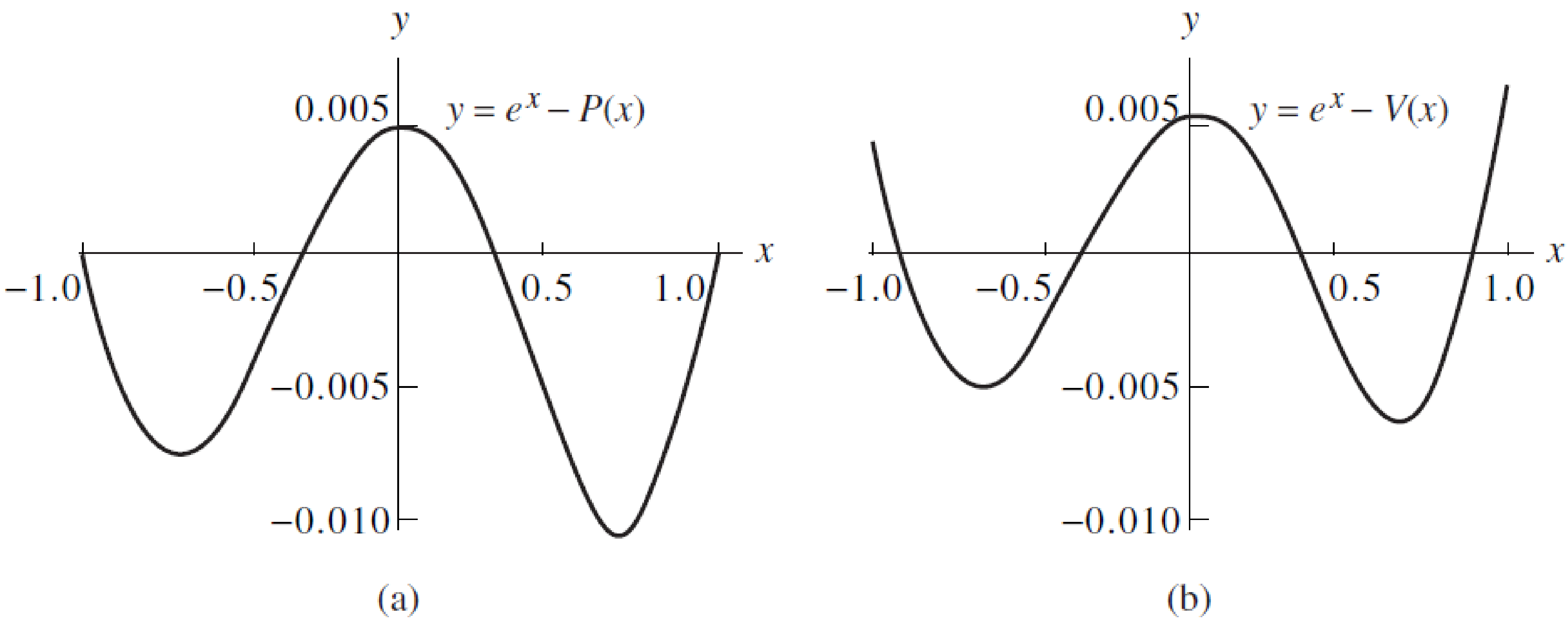


Figure 4.16 (a) The error function $y = e^x - P(x)$ for Lagrange approximation over $[-1, 1]$.
(b) The error function $y = e^x - V(x)$ for Chebyshev approximation over $[-1, 1]$.

Runge Phenomenon

- Consider Lagrange interpolating to $f(x)$ over the interval $[-1, 1]$ based on equally spaced nodes. Does the error $E_N(x) = f(x) - P_N(x)$ tend to zero as N increases?
- For functions link sin(x) or e^x, where all the derivatives are bounded by the same constant M, the answer is yes.
- In general, the answer to this question is no, and it is easy to find functions for which the sequence $\{P_N(\mathrm{x})\}$ does not converge.
- For example, if $f(x) = 1/(1+12x^2)$, the maximum of the error term $E_N(x)$ grows when $N \to \infty$.
- This nonconvergence is called the ***Runge phenomenon***.
- However, the Chebyshev interpolation performs better.
- Under the condition that Chebyshev nodes be used, the error $E_N(x)$ will go to zero as $N \to \infty$.

Runge Phenomenon

- Consider construct an interpolating polynomial of degree 10 for $f(x) = 1/(1+12x^2)$

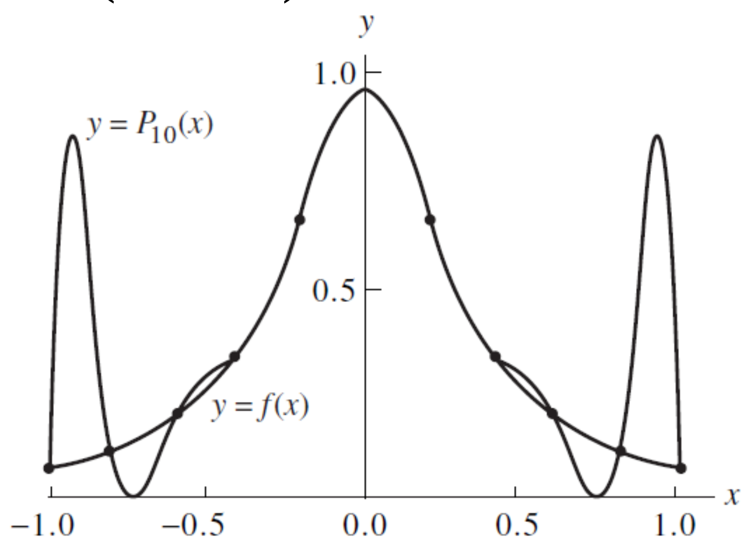


Figure 4.17 (a) The polynomial approximation to $y = 1/(1 + 12x^2)$ based on 11 equally spaced nodes over $[-1, 1]$.

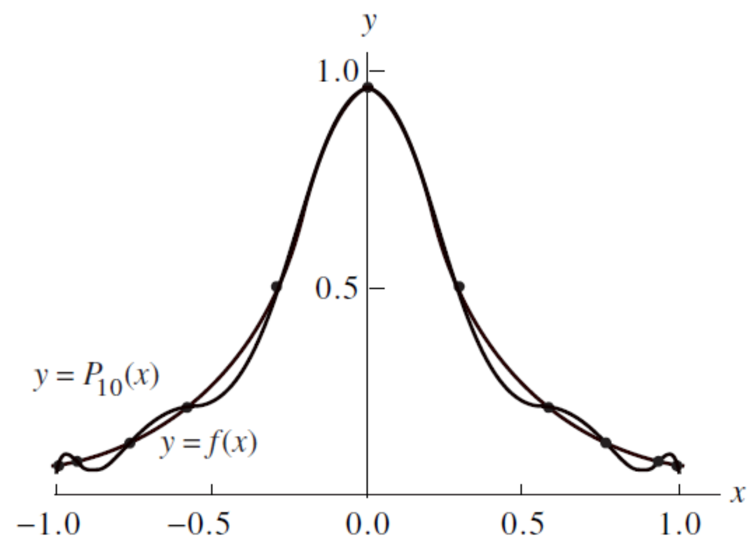


Figure 4.17 (b) The polynomial approximation to $y = 1/(1 + 12x^2)$ based on 11 Chebyshev nodes over $[-1, 1]$.

- In general, if $f(x)$ and $f'(x)$ are continuous on $[-1, 1]$, then it can be proved that Chebyshev interpolation will produce a sequence of polynomials $\{P_N(x)\}$ that converges uniformly to $f(x)$ over $[-1, 1]$.

Transforming the Interval

Sometimes it is necessary to take a problem stated on an interval $[a, b]$ and reformulate the problem on the interval $[c, d]$ where the solution is known. If the approximation $P_N(x)$ to $f(x)$ is to be obtained on the interval $[a, b]$, then we change the variable so that the problem is reformulated on [-1, 1]:

(12) $$x = \left(\frac{b-a}{2}\right)t + \frac{a+b}{2} \quad \text{or} \quad t = 2\frac{x-a}{b-a} - 1,$$

where $a \le x \le b$ and $-1 \le t \le 1$.

The required Chebyshev nodes of $T_{N+1}(t)$ on [-1, 1] are

(13) $$t_k = \cos\left((2N+1-2k)\frac{\pi}{2N+2}\right) \quad \text{for} \quad k = 0, 1, \ldots, N.$$

and the interpolation nodes on $[a, b]$ are obtained by using (12):

(14) $$x_k = t_k\frac{b-a}{2} + \frac{a+b}{2} \quad \text{for} \quad k = 0, 1, \ldots, N.$$

Transforming the Interval

Theorem 4.7 (Lagrange-Chebyshev Approximation Polynomial). Assume that $P_N(x)$ is the Lagrange polynomial that is based on the Chebyshev nodes given in (14). If $f \in C^{N+1}[a, b]$, then

$$|f(x) - P_N(x)| \leq \frac{2(b-a)^{N+1}}{4^{N+1}(N+1)!} \max_{a \leq x \leq b} \{|f^{(N+1)}(x)|\}. \tag{15}$$

Chebyshev Approximation

Theorem 4.8 The Chebyshev approximation polynomial $P_N(x)$ of degree $\leq N \; for \; f(x)$ over $[-1, 1]$ can be written as a sum of $\{T_j(x)\}$:

$$f(x) \approx P_N(x) = \sum_{j=0}^{N} c_j T_j(x)\,. \tag{21}$$

The coefficients $\{c_j\}$ are computed with the formulas

$$c_0 = \frac{1}{N+1} \sum_{k=0}^{N} f(x_k)\, T_0(x_k) = \frac{1}{N+1} \sum_{k=0}^{N} f(x_k) \tag{22}$$

and

$$c_j = \frac{2}{N+1} \sum_{k=0}^{N} f(x_k)\, T_j(x_k)$$

$$= \frac{2}{N+1} \sum_{k=0}^{N} f(x_k) \cos\left(\frac{j\pi(2k+1)}{2N+2}\right) \qquad \text{for } j = 1, 2, \ldots, N. \tag{23}$$

An example

Example 4.16. Find the Chebyshev polynomial $P_3(x)$ that approximates the function $f(x) = e^x$ over $[-1, 1]$.

The coefficients are calculated using formulas (22) and (23), and the nodes $x_k = \cos(\pi(2k+1)/8)$ for $k = 0, 1, 2, 3$.

$$c_0 = \frac{1}{4}\sum_{k=0}^{3} e^{x_k} T_0(x_k) = \frac{1}{4}\sum_{k=0}^{3} e^{x_k} = 1.26606568,$$

$$c_1 = \frac{1}{2}\sum_{k=0}^{3} e^{x_k} T_1(x_k) = \frac{1}{2}\sum_{k=0}^{3} e^{x_k}\, x_k = 1.13031500,$$

$$c_2 = \frac{1}{2}\sum_{k=0}^{3} e^{x_k} T_2(x_k) = \frac{1}{2}\sum_{k=0}^{3} e^{x_k} \cos(2\pi\frac{2k+1}{8}) = 0.27145036,$$

$$c_3 = \frac{1}{2}\sum_{k=0}^{3} e^{x_k} T_3(x_k) = \frac{1}{2}\sum_{k=0}^{3} e^{x_k} \cos(3\pi\frac{2k+1}{8}) = 0.04379392.$$

$$P_3(x) = 1.26606568T_0(x) + 1.13031500T_1(x) \\ +0.27145036T_2(x) + 0.04379392T_3(x).$$

$$P_3(x) = 0.99461532 + 0.99893324x + 0.54290072x^2 + 0.17517568x^3,$$

Matlab Code

Program 4.3 (Chebyshev Approximation). To construct and evaluate the Chebyshev interpolating polynomial of degree N over the interval $[-1, 1]$, where

$$P(x) = \sum_{j=0}^{N} c_j T_j(x)$$

is based on the nodes

$$x_k = \cos\left(\frac{(2k+1)\pi}{2N+2}\right).$$

```
function [C,X,Y]=cheby(fun,n,a,b)
%Input  - fun is the string function to be approximated
%       - N is the degree of the Chebyshev interpolating
%         polynomial
%       - a is the left endpoint
%       - b is the right endpoint
%Output - C is the coefficient list for the polynomial
%       - X contains the abscissas
%       - Y contains the ordinates
if nargin==2, a=-1;b=1;end
d=pi/(2*n+2);
C=zeros(1,n+1);
for k=1:n+1
   X(k)=cos((2*k-1)*d);
end
X=(b-a)*X/2+(a+b)/2;
x=X;
Y=eval(fun);
for k =1:n+1
   z=(2*k-1)*d;
   for j=1:n+1
      C(j)=C(j)+Y(k)*cos((j-1)*z);
   end
end
C=2*C/(n+1);
C(1)=C(1)/2;
```

Padé Approximations

A rational approximation

- A rational approximation to $f(x)$ on $[a, b]$ is the quotient of two polynomials $P_N(x)$ and $Q_M(x)$ of degrees N and M, respectively. We use the notation $R_{N,M}(x)$ to denote this quotient:

$$R_{N,M}(x) = \frac{P_N(x)}{Q_M(x)} \quad \text{for } a \leq x \leq b.$$

- Our goal is to make the maximum error as small as possible. For a given amount of computational effort, one can usually construct a rational approximation that has a smaller overall error on $[a, b]$ than a polynomial approximation.
- Our development is an introduction and will be limited to Padé approximations.

Method of Padé

- Requires that $f(x)$ and its derivative be continuous at $x = 0$.
- Construct

(2) $$P_N(x) = P_0 + P_1 x + P_2 x^2 + \cdots P_N x^N$$

and

(3) $$Q_M(x) = 1 + q_1 x + q_2 x^2 + \cdots q_M x^M.$$

- The polynomials in (2) and (3) are constructed so that $f(x)$ and $R_{N,M}(x)$ agree at $x = 0$ and their derivatives up to $N + M$ agree at $x = 0$.
- For a fixed value of $N + M$ the error is smallest when $P_N(x)$ and $Q_M(x)$ have the same degree or when $P_N(x)$ has degree one higher than $Q_M(x)$.

Method of Padé

- Assume that $f(x)$ is analytic and has the Maclaurin expansion

(4) $$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots a_k x^k + \cdots,$$

and form the difference $f(x)Q_M(x) - P_N(x) = Z(x)$:

(5) $$\left(\sum_{j=0}^{\infty} a_j x^j\right)\left(\sum_{j=0}^{M} q_j x^j\right) - \sum_{j=0}^{N} p_j x^j = \sum_{j=N+M+1}^{\infty} c_j x^j .$$

- When the left side of (5) is multiplied out and the coefficients of the powers of x^j are set equal to zero for $k = 0, 1, \ldots, N + M$, the result is a system of $N + M + 1$ linear equations:

Method of Padé

$$
\begin{aligned}
a_0 - p_0 &= 0 \\
q_1 a_0 + a_1 - p_1 &= 0 \\
q_2 a_0 + q_1 a_1 + a_2 - p_2 &= 0 \\
q_3 a_0 + q_2 a_1 + q_1 a_2 + a_2 - p_3 &= 0 \\
q_M a_{N-M} + q_{M-1} a_{N-M+1} + \cdots + a_N - p_N &= 0
\end{aligned} \tag{6}
$$

and

$$
\begin{aligned}
q_M a_{N-M+1} + q_{M-1} a_{N-M+2} + \cdots + q_1 a_N \qquad + a_{N+1} &= 0 \\
q_M a_{N-M+2} + q_{M-1} a_{N-M+3} + \cdots + q_1 a_{N+1} \qquad + a_{N+2} &= 0 \\
\vdots \qquad\qquad\qquad\qquad\qquad\qquad & \vdots \\
q_M a_N \qquad + q_{M-1} a_{N+1} \qquad + \cdots + q_1 a_{N+M+1} + a_{N+M} &= 0
\end{aligned} \tag{7}
$$

An example

Example 4.17. Establish the Padé approximation

$$\cos(x) \approx R_{4,4}(x) = \frac{15{,}120 - 6900x^2 + 313x^4}{15{,}120 + 660x^2 + 13x^4}. \tag{8}$$

If the Maclaurin expansion for $\cos(x)$ is used, we will obtain nine equations in nine unknows. Instead, notice that both $\cos(x)$ and $R_{4,4}(x)$ are even functions and involve powers of x^2. We can simplify the computations if we start with $f(x) = \cos(x^{1/2})$:

$$f(x) = 1 - \frac{1}{2}x + \frac{1}{24}x^2 - \frac{1}{720}x^3 + \frac{1}{40{,}320}x^4 - \cdots .$$

In this case, equation (5) becomes

$$\left(1 - \frac{1}{2}x + \frac{1}{24}x^2 - \frac{1}{720}x^3 + \frac{1}{40{,}320}x^4 - \cdots\right)(1 + q_1x + q_2x^2) - p_0 - p_1x - p_2x^2$$

$$= 0 + 0x + 0x^2 + 0x^3 + 0x^4 + c_5x^5 + c_6x^6 + \cdots .$$

An example

$$1 - p_0 = 0$$

$$-\frac{1}{2} + q_1 - p_1 = 0$$

$$\frac{1}{24} - \frac{1}{2}q_1 + q_2 - p_2 = 0$$

$$-\frac{1}{720} + \frac{1}{24}q_1 - \frac{1}{2}q_2 = 0$$

$$\frac{1}{40{,}320} - \frac{1}{720}q_1 + \frac{1}{24}q_2 = 0.$$

$$p_1 = -\frac{1}{2} + \frac{11}{252} = -\frac{115}{252},$$

$$p_2 = \frac{1}{24} - \frac{11}{504} + \frac{13}{15{,}120} = \frac{313}{15{,}120}.$$

$$q_2 = \frac{1}{18}\left(\frac{1}{30} - \frac{1}{56}\right) = \frac{13}{15{,}120},$$

$$q_1 = \frac{1}{30} + \frac{156}{15{,}120} = \frac{11}{252}.$$

$$f(x) \approx \frac{1 - 115\,x/252 + 313\,x^2/15{,}120}{1 + 11\,x/252 + 13\,x^2/15{,}120}.$$

$$R_{4,4}(x) = \frac{15{,}120 - 6900x^2 + 313x^4}{15{,}120 + 660x^2 + 13x^4}.$$

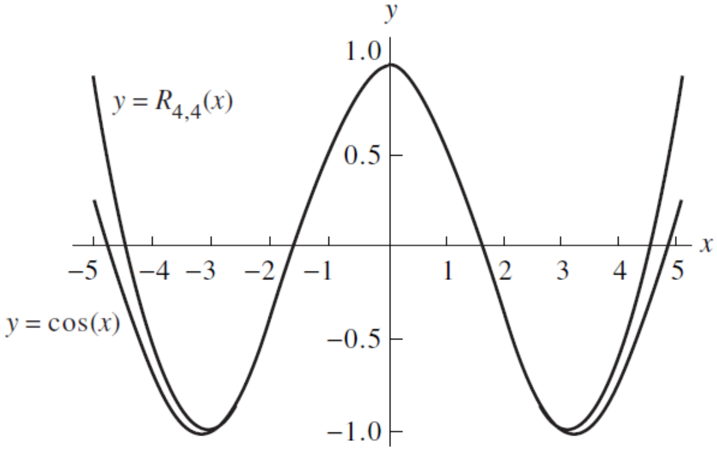


Figure 4.18 The graph of $y = \cos(x)$ and its Padé approximation $R_{4,4}(x)$.

Continued Fraction Form

- The Padé approximation $R_{4,4}(x)$ in Example 4.17 requires a minimum of 12 arithmetic operations to perform an evaluation. It is possible to reduce this number to seven by the use of continued fractions.

$$R_{4,4}(x) = \frac{15{,}120/313 - (6900/313)x^2 + x^4}{\dfrac{15{,}120}{13} + (660/13)x^2 + x^4}$$

$$= \frac{313}{13} - \left(\frac{296{,}280}{169}\right)\left(\frac{12{,}600/823 + x^2}{15{,}120/13 + (600/13)x^2 + x^4}\right).$$

$$R_{4,4}(x) = \frac{313}{13} - \frac{296{,}280/169}{\dfrac{15{,}120/13 + (660/13)x^2 + x^4}{12{,}600/823 + x^2}}$$

$$= \frac{313}{13} - \frac{296{,}280/169}{\dfrac{379{,}380}{10{,}699} + x^2 + \dfrac{420{,}078{,}960/677{,}329}{12{,}600/823 + x^2}}$$

Continued Fraction Form

$$R_{4,4}(x) = 24.07692308 - \frac{1753.13609467}{35.45938873 + x^2 + 620.19928277/(15.30984204 + x^2)}.$$

Compared with the Taylor polynomial $P_6(x)$ of degree N = 6

$$P_6(x) = 1 + x^2\left(-\frac{1}{2} + x^2\left(\frac{1}{24} - \frac{1}{720}x^2\right)\right)$$
$$= 1 + x^2(-0.5 + x^2(0.0416666667 - 0.0013888889x^2)).$$

Continued Fraction Form

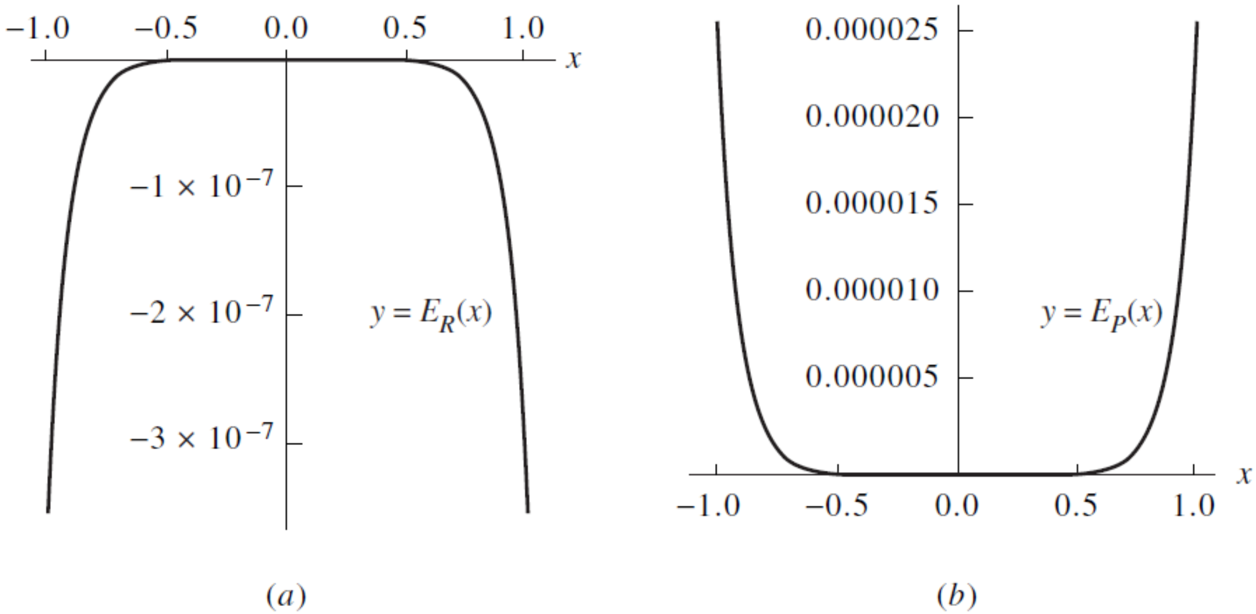


Figure 4.19 (a) The graph of the error $E_R(x) = \cos(x) - R_{4,4}(x)$ for the Padé approximation $R_{4,4}(x)$. (b) The graph of the error $E_P(x) = \cos(x) - P_6(x)$ for the Taylor approximation $P_6(x)$.

- The largest errors occur at the endpoints and are $E_R(1) = -0.0000003599$ and $E_P(1) = 0.0000245281$, respectively. The magnitude of the largest error for $R_{4,4}(x)$ is about 1.467% of the error for $P_6(x)$.
- The Padé approximation outperforms the Taylor approximation better on smaller intervals, and over $[-0.1, 0.1]$ we find that $E_R(0.1) = -0.0000000004$ and $E_P(0.1) = 0.0000000966$, so the magnitude of the error for $R_{4,4}(x)$ is about 0.384% of the magnitude of the error for $P_6(x)$.