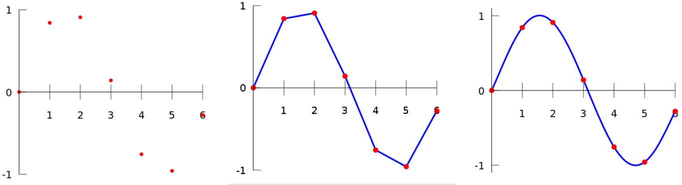
Interpolation



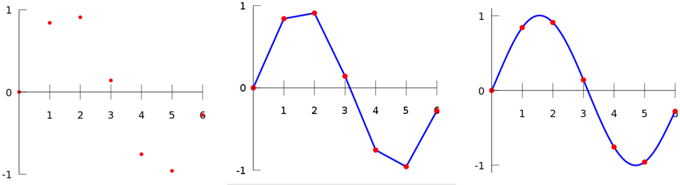
- Basic interpolation problem: for a given data

$$(x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n), \quad x_0 < x_1 < x_2 < \cdots < x_n$$

determine the function $p : \mathbb{R} \to \mathbb{R}$ such that

$$p(x_i) = y_i, \quad i = 0, 1, \ldots, n$$

Interpolation



- Basic interpolation problem: for a given data

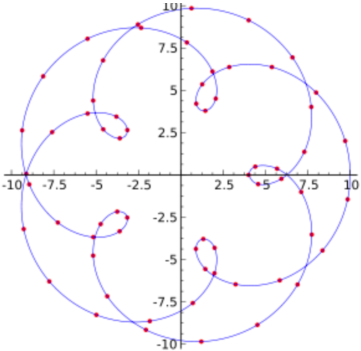
$$(x_0, y_0), (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n), \quad x_0 < x_1 < x_2 < \cdots < x_n$$

determine the function $p : \mathbb{R} \to \mathbb{R}$ such that

$$p(x_i) = y_i, \quad i = 0, 1, \ldots, n$$

- p is interpolating function, or interpolant, for given data
- Additional data might be prescribed, such as slope of interpolant at given points
- Additional constraints might be imposed, such as smoothness, monotonicity, or convexity of interpolant
- p could be function of more than one variable, but we will consider only one-dimensional case

Purposes for Interpolation



- Plotting smooth curve through discrete data points
- Reading between lines of table
- Differentiating or integrating tabular data
- Quick and easy evaluation of mathematical function
- Replacing complicated function by simple one

Interpolation vs Approximation

- By definition, interpolating function fits given data points exactly
- Interpolation is inappropriate if data points subject to significant errors
- It is usually preferable to smooth noisy data, for example by least squares approximation

Issues in Interpolation

Arbitrarily many functions interpolate given set of data points

Choice of function for interpolation based on

- How easy interpolating function is to work with
  - determining its parameters
  - evaluating interpolant
  - differentiating or integrating interpolant
- How well properties of interpolant match properties of data to be fit (smoothness, monotonicity, convexity, periodicity, etc.)

Functions for Interpolation

- Families of functions commonly used for interpolation include
  - Polynomials
  - Piecewise polynomials
  - Trigonometric functions
  - Exponential functions
  - Rational functions
- For now we will focus on interpolation by polynomials

Uniqueness

Theorem

Given (x_0, y_0), $(x_1, y_1), \ldots, (x_n, y_n)$, *where* $x_i \neq x_j$ *if* $i \neq j$, *there exists a uniques interpolating polynomial* $p(x)$ *of degree no greater than* n, *such that* $p(x_i) = y_i, i = 0, \ldots, n$.

Uniqueness

Theorem

Given (x_0, y_0), $(x_1, y_1), \ldots, (x_n, y_n)$, where $x_i \neq x_j$ if $i \neq j$, there exists a uniques interpolating polynomial $p(x)$ of degree no greater than n, such that $p(x_i) = y_i, i = 0, \ldots, n$.

Proof. Assume that there are two polynomials of degree $\leq n$, $p(x)$ and $q(x)$ such that $p(x_i) = q(x_i) = y_i, i = 0, \ldots, n$. Then

$$r(x) = p(x) - q(x)$$

is a polynomial of degree $\leq n$ which has $n + 1$ distinct roots $x_0, x_1, \ldots, x_n$. This is a contradiction with Fundamental Theorem of Algebra (any polynomial of degree n has at most n roots). $\square$

Forms of Polynomials

Polynomial can be represented in several different forms

Forms of Polynomials

Polynomial can be represented in several different forms

- **Power form**

$$p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$$

If $a_n \neq 0$, then $p(x)$ has degree n.

Forms of Polynomials

Polynomial can be represented in several different forms

- **Power form**

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

  If $a_n \neq 0$, then $p(x)$ has degree n.

- **Shifted power form with center c:**

$$p(x) = b_0 + b_1(x-c) + b_2(x-c)^2 + \cdots + b_n(x-c)^n$$

Forms of Polynomials

Polynomial can be represented in several different forms

- **Power form**

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

If $a_n \neq 0$, then $p(x)$ has degree n.

- **Shifted power form with center c:**

$$p(x) = b_0 + b_1(x-c) + b_2(x-c)^2 + \cdots + b_n(x-c)^n$$

- **Newton form**

$$\begin{aligned} p(x) &= a_0 + a_1(x-c_1) + a_2(x-c_1)(x-c_2) + \cdots + a_n(x-c_1)(x-c_2)\dots(x-c_n) \\ &= a_0 + \sum_{k=1}^{n} a_k \prod_{i=1}^{k}(x-c_i) \end{aligned}$$

Forms of Polynomials

Polynomial can be represented in several different forms

- **Power form**

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

  If $a_n \neq 0$, then $p(x)$ has degree n.

- **Shifted power form with center c:**

$$p(x) = b_0 + b_1(x-c) + b_2(x-c)^2 + \cdots + b_n(x-c)^n$$

- **Newton form**

$$\begin{aligned} p(x) &= a_0 + a_1(x-c_1) + a_2(x-c_1)(x-c_2) + \cdots + a_n(x-c_1)(x-c_2)\ldots(x-c_n) \\ &= a_0 + \sum_{k=1}^{n} a_k \prod_{i=1}^{k}(x-c_i) \end{aligned}$$

  - if $c_1 = c_2 = \cdots = c_n = c$ we obtain the shifted power from
  - if $c_1 = c_2 = \cdots = c_n = 0$: we have the power form.

Lagrange Method

Given $n+1$ pairs

$$(x_0, y_0),\ (x_1, y_1),\ \ldots, (x_n, y_n) \in \mathbb{R}^2$$

where $x_i \neq x_j$ if $i \neq j$, our goal is to find a polynomial $p(x)$ such that

$$p(x_i) = y_i, \quad i = 0, 1, \ldots, n.$$

Lagrange Method

Given $n+1$ pairs

$$(x_0, y_0),\ (x_1, y_1),\ \dots, (x_n, y_n) \in \mathbb{R}^2$$

where $x_i \neq x_j$ if $i \neq j$, our goal is to find a polynomial $p(x)$ such that

$$p(x_i) = y_i, \quad i = 0, 1, \dots, n.$$

We define the nth degree polynomial (called Lagrange polynomials) for $j = 0, 1, \dots, n$:

$$\ell_j(x) = \frac{(x - x_0)(x - x_1)(x - x_2)\dots(x - x_{j-1})(x - x_{j+1})\dots(x - x_n)}{(x_j - x_0)(x_j - x_1)(x_j - x_2)\dots(x_j - x_{j-1})(x_j - x_{j+1})\dots(x_j - x_n)}$$

In short

$$\ell_j(x) = \prod_{i=0, i\neq j}^{n} \frac{(x - x_i)}{(x_j - x_i)}$$

Lagrange Method

Given $n+1$ pairs

$$(x_0, y_0),\ (x_1, y_1),\ \ldots, (x_n, y_n) \in \mathbb{R}^2$$

where $x_i \neq x_j$ if $i \neq j$, our goal is to find a polynomial $p(x)$ such that

$$p(x_i) = y_i, \quad i = 0, 1, \ldots, n.$$

We define the nth degree polynomial (called Lagrange polynomials) for $j = 0, 1, \ldots, n$:

$$\ell_j(x) = \frac{(x - x_0)(x - x_1)(x - x_2)\ldots(x - x_{j-1})(x - x_{j+1})\ldots(x - x_n)}{(x_j - x_0)(x_j - x_1)(x_j - x_2)\ldots(x_j - x_{j-1})(x_j - x_{j+1})\ldots(x_j - x_n)}$$

In short

$$\ell_j(x) = \prod_{i=0, i\neq j}^{n} \frac{(x - x_i)}{(x_j - x_i)}$$

$\ell_j(x_j) =?$

Lagrange Method

Given $n+1$ pairs

$$(x_0, y_0),\ (x_1, y_1),\ \ldots, (x_n, y_n) \in \mathbb{R}^2$$

where $x_i \neq x_j$ if $i \neq j$, our goal is to find a polynomial $p(x)$ such that

$$p(x_i) = y_i, \quad i = 0, 1, \ldots, n.$$

We define the nth degree polynomial (called Lagrange polynomials) for $j = 0, 1, \ldots, n$:

$$\ell_j(x) = \frac{(x - x_0)(x - x_1)(x - x_2)\ldots(x - x_{j-1})(x - x_{j+1})\ldots(x - x_n)}{(x_j - x_0)(x_j - x_1)(x_j - x_2)\ldots(x_j - x_{j-1})(x_j - x_{j+1})\ldots(x_j - x_n)}$$

In short

$$\ell_j(x) = \prod_{i=0, i\neq j}^{n} \frac{(x - x_i)}{(x_j - x_i)}$$

$\ell_j(x_j) =?$
$\ell_j(x_i) =?\ \ i \neq j$

Lagrange Method

Given $n+1$ pairs

$$(x_0, y_0),\ (x_1, y_1),\ \dots, (x_n, y_n) \in \mathbb{R}^2$$

where $x_i \neq x_j$ if $i \neq j$, our goal is to find a polynomial $p(x)$ such that

$$p(x_i) = y_i, \quad i = 0, 1, \dots, n.$$

We define the nth degree polynomial (called Lagrange polynomials) for $j = 0, 1, \dots, n$:

$$\ell_j(x) = \frac{(x - x_0)(x - x_1)(x - x_2)\dots(x - x_{j-1})(x - x_{j+1})\dots(x - x_n)}{(x_j - x_0)(x_j - x_1)(x_j - x_2)\dots(x_j - x_{j-1})(x_j - x_{j+1})\dots(x_j - x_n)}$$

In short

$$\ell_j(x) = \prod_{i=0, i\neq j}^{n} \frac{(x - x_i)}{(x_j - x_i)}$$

$\ell_j(x_j) = ?$
$\ell_j(x_i) = ?\ \ i \neq j$
Can you determine an interpolating polynomial ($p(x_i) = y_i, i = 0, \dots, n$)

Lagrange Interpolating Polynomial

Given $n+1$ pairs

$$(x_0, y_0),\ (x_1, y_1),\ \ldots, (x_n, y_n) \in \mathbb{R}^2$$

where $x_i \neq x_j$ if $i \neq j$

$$p(x) = y_0\ell_0(x) + y_1\ell_1(x) + \cdots + y_n\ell_n(x)$$

$$\ell_j(x) = \prod_{i=0, i\neq j}^{n} \frac{(x - x_i)}{(x_j - x_i)}.$$

Example 1

Use the Lagrange method to find the second degree polynomial $p(x)$ interpolating the data

x	-1	0	1
y	3	5	2

The Method of Undetermined Coefficients

Recall the power form of polynomial

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n.$$

By the fact that $p(x_i) = y_i, i = 0, \ldots, n$, to find the coefficients, we need to solve the following linear system for $a_0, a_1, \ldots, a_n$:

$$\begin{aligned}
a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n &= y_0 \\
a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_n x_1^n &= y_1 \\
&\vdots \\
a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_n^n &= y_n
\end{aligned}$$

In matrix form is given by $Va = y$, where

$$V = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix}, \quad a = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}, \quad y = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

The matrix V is called Vandermonde matrix and it is known to have the determinant

$$\det(V) = \prod_{0 \le i \le j \le n} (x_i - x_j) \neq 0$$

Example 2

Use the method of undetermined coefficients to find the second degree polynomial $p(x)$ interpolating the data

x	-1	0	1
y	3	5	2

Question

Assume that we are given the interpolating polynomial $p_n(x)$ for data points $(x_i, y_i), i = 0, \ldots, n$.
Suppose that a new data point (x_{n+1}, y_{n+1}) is added.

"Is there anyway to use $p_n(x)$ in order to construct $p_{n+1}(x)$ for the points $(x_i, y_i), i = 0, \ldots, n+1$?"

Question

Assume that we are given the interpolating polynomial $p_n(x)$ for data points $(x_i, y_i), i = 0, \ldots, n$.
Suppose that a new data point (x_{n+1}, y_{n+1}) is added.

"Is there anyway to use $p_n(x)$ in order to construct $p_{n+1}(x)$ for the points $(x_i, y_i), i = 0, \ldots, n+1$?"

Unfortunately, the Lagrange method and method of undetermined coefficients do not utilize the fact of having $p_n(x)$ when computing $p_{n+1}(x)$. We use Newton method, next.

Newton Form of Polynomial

$$
\begin{aligned}
p(x) \quad &= a_0 + a_1(x-c_1) + a_2(x-c_1)(x-c_2) + \dots \\
&+ a_n(x-c_1)(x-c_2)\dots(x-c_n)
\end{aligned}
$$

or equivalently,

$$
p(x) = a_0 + \sum_{k=1}^{n} a_k \prod_{i=1}^{k}(x-c_i)
$$

Newton Form of Polynomial

$$
\begin{aligned}
p(x) &= a_0 + a_1(x - c_1) + a_2(x - c_1)(x - c_2) + \dots \\
&\quad + a_n(x - c_1)(x - c_2)\dots(x - c_n)
\end{aligned}
$$

or equivalently,

$$
p(x) = a_0 + \sum_{k=1}^{n} a_k \prod_{i=1}^{k}(x - c_i)
$$

We set $c_i = x_{i-1}$, to get interpolating polynomials of degree n passing through the point $\{(x_i, y_i) \in \mathbb{R}^2, i = 0, 1, \dots, n\}$, that is

$$
\begin{aligned}
p_n(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \dots \\
&\quad + a_n(x - x_0)\dots(x - x_{n-1})
\end{aligned}
$$

Newton Method

- Interpolating polynomial, passing through $\{(x_i, y_i) \in \mathbb{R}^2, i = 0, 1, \ldots, n\}$:

$$\begin{aligned} p_n(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots \\ &\qquad + a_n(x - x_0)\ldots(x - x_{n-1}) \end{aligned}$$

Newton Method

- Interpolating polynomial, passing through $\{(x_i, y_i) \in \mathbb{R}^2, i = 0, 1, \ldots, n\}$:

$$\begin{aligned} p_n(x) \quad &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots \\ &+ a_n(x - x_0)\ldots(x - x_{n-1}) \end{aligned}$$

- Interpolating polynomial, passing through $\{(x_i, y_i) \in \mathbb{R}^2, i = 0, 1, \ldots, n+1\}$:

$$\begin{aligned} p_{n+1}(x) \quad &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots \\ &+ a_n(x - x_0)\ldots(x - x_{n-1}) \\ &+ a_{n+1}(x - x_0)\ldots(x - x_n) \end{aligned}$$

Newton Method

- Interpolating polynomial, passing through $\{(x_i, y_i) \in \mathbb{R}^2, i = 0, 1, \ldots, n\}$:

$$\begin{aligned} p_n(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots \\ &\quad + a_n(x - x_0)\ldots(x - x_{n-1}) \end{aligned}$$

- Interpolating polynomial, passing through $\{(x_i, y_i) \in \mathbb{R}^2, i = 0, 1, \ldots, n+1\}$:

$$\begin{aligned} p_{n+1}(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \ldots \\ &\quad + a_n(x - x_0)\ldots(x - x_{n-1}) \\ &\quad + a_{n+1}(x - x_0)\ldots(x - x_n) \end{aligned}$$

- Comparing $p_n(x)$ and $p_{n+1}(x)$, we get

$$p_{n+1}(x) = p_n(x) + r(x)$$

where

$$r(x) = a_{n+1}(x - x_0)\ldots(x - x_n).$$

- If we plug $x = x_j$ for $j = 0, \ldots, n$ in $r(x) = a_{n+1}(x - x_0)\ldots(x - x_n)$ we get
$$r(x_j) = a_{n+1}(x_j - x_0)\ldots(x_j - x_i) = 0.$$

- If we plug $x = x_j$ for $j = 0, \ldots, n$ in $r(x) = a_{n+1}(x - x_0) \ldots (x - x_n)$ we get

$$r(x_j) = a_{n+1}(x_j - x_0) \ldots (x_j - x_i) = 0.$$

Hence

$$p_{n+1}(x_j) = p_n(x_j) = y_j, \qquad j = 0, \ldots, n$$

- If we plug $x = x_j$ for $j = 0, \ldots, n$ in $r(x) = a_{n+1}(x - x_0)\ldots(x - x_n)$ we get

$$r(x_j) = a_{n+1}(x_j - x_0)\ldots(x_j - x_i) = 0.$$

Hence

$$p_{n+1}(x_j) = p_n(x_j) = y_j, \qquad j = 0, \ldots, n$$

- Note that $r(x_{n+1}) \neq 0$.

- If we plug $x = x_j$ for $j = 0, \dots, n$ in $r(x) = a_{n+1}(x - x_0)\dots(x - x_n)$ we get
$$r(x_j) = a_{n+1}(x_j - x_0)\dots(x_j - x_i) = 0.$$
Hence
$$p_{n+1}(x_j) = p_n(x_j) = y_j, \qquad j = 0, \dots, n$$

- Note that $r(x_{n+1}) \neq 0$.
Interpolating polynomial $p_{n+1}(x)$ must satisfy
$$p_{n+1}(x_{n+1}) = y_{n+1}.$$

- If we plug $x = x_j$ for $j = 0, \ldots, n$ in $r(x) = a_{n+1}(x - x_0)\ldots(x - x_n)$ we get
$$r(x_j) = a_{n+1}(x_j - x_0)\ldots(x_j - x_i) = 0.$$
Hence
$$p_{n+1}(x_j) = p_n(x_j) = y_j, \qquad j = 0, \ldots, n$$

- Note that $r(x_{n+1}) \neq 0$.
Interpolating polynomial $p_{n+1}(x)$ must satisfy
$$p_{n+1}(x_{n+1}) = y_{n+1}.$$
This together with $p_{n+1}(x) = p_n(x) + r(x)$ implies that
$$y_{n+1} = p_{n+1}(x_{n+1}) = p_n(x_{n+1}) + a_{n+1}(x_{n+1} - x_0)\ldots(x_{n+1} - x_n)$$

- If we plug $x = x_j$ for $j = 0, \dots, n$ in $r(x) = a_{n+1}(x - x_0)\dots(x - x_n)$ we get
$$r(x_j) = a_{n+1}(x_j - x_0)\dots(x_j - x_i) = 0.$$
Hence
$$p_{n+1}(x_j) = p_n(x_j) = y_j, \qquad j = 0, \dots, n$$

- Note that $r(x_{n+1}) \neq 0$.
Interpolating polynomial $p_{n+1}(x)$ must satisfy
$$p_{n+1}(x_{n+1}) = y_{n+1}.$$
This together with $p_{n+1}(x) = p_n(x) + r(x)$ implies that
$$y_{n+1} = p_{n+1}(x_{n+1}) = p_n(x_{n+1}) + a_{n+1}(x_{n+1} - x_0)\dots(x_{n+1} - x_n)$$

Solving for a_{n+1} gives
$$a_{n+1} = \frac{y_{n+1} - p_n(x_{n+1})}{(x_{n+1} - x_0)\dots(x_{n+1} - x_n)}$$

Example 3

Assume that a new data point $(x_3, y_3) = (2, 4)$ is added to the data set in the previous example:

x	-1	0	1	2
y	3	5	2	4

From the previous example, we know that the quadratic interpolating polynomial for the first three data points is given by

$$p_2(x) = -\frac{5}{2}x^2 - \frac{1}{2}x + 5$$

Use Newton's method to find the 3rd degree interpolating polynomial for the given data.

Solution. Note that

$$\begin{aligned} p_3(x) &= p_2(x) + a_3(x - x_0)(x - x_1)(x - x_2) \\ &= -\frac{5}{2}x^2 - \frac{1}{2}x + 5 + a_3(x+1)(x-0)(x-1) \end{aligned}$$

where

$$a_3 = \frac{y_3 - p_2(x_3)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} = \frac{4 - p_2(2)}{(2+1)(2-0)(2-1)} = \frac{5}{3}$$

Thus

$$\begin{aligned} p_3(x) &= -\frac{5}{2}x^2 - \frac{1}{2}x + 5 + \frac{5}{3}(x+1)(x)(x-1) \\ &= -\frac{5}{2}x^2 - \frac{1}{2}x + 5 + \frac{5}{3}(x^3 - x) \\ &= \frac{5}{3}x^3 - \frac{5}{2}x^2 - \frac{7}{6}x + 5 \end{aligned}$$

Error of Interpolation

Assume that we interpolate a given function f using the data points $\{(x_i, f(x_i)), i = 0, \dots, n\}$, where $x_0 < x_1 < \cdots < x_n$.

Theorem

Let $f(x) \in C^{n+1}([x_0, x_n])$. Then there exists $\xi \in [x_0, x_n]$ such that for all $x \in [x_0, x_n]$, the error

$$e_n(x) = f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$$

where

$$\prod_{i=0}^{n} (x - x_i) = (x - x_0)(x - x_1) \dots (x - x_n)$$

Error of Interpolation

Assume that we interpolate a given function f using the data points $\{(x_i, f(x_i)), i = 0, \dots, n\}$, where $x_0 < x_1 < \dots < x_n$.

Theorem

Let $f(x) \in C^{n+1}([x_0, x_n])$. Then there exists $\xi \in [x_0, x_n]$ such that for all $x \in [x_0, x_n]$, the error

$$e_n(x) = f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$$

where

$$\prod_{i=0}^{n} (x - x_i) = (x - x_0)(x - x_1) \dots (x - x_n)$$

Note that ξ depends on x and its value is not available explicitly. However, if we obtain an upper bound c_n such that $|f^{(n+1)}(x)| \le c_n, \ \forall x \in [x_0, x_n]$, then

$$|e_n(x)| \le \frac{c_n}{(n+1)!} \prod_{i=0}^{n} |x - x_i| \quad \forall x \in [x_0, x_n].$$

Example 4

Consider the quadratic interpolation of $f(x) = \cos x$ using the three data points $x_0 = -\pi/2$, $x_1 = 0$, and $x_2 = \pi/2$. Estimate the error of interpolation at $x = \pi/4$.

Note that the upper bound on the error

$$|e_n(x)| \leq \frac{c_n}{(n+1)!} \prod_{i=0}^{n} |x - x_i| \quad \forall x \in [x_0, x_n].$$

does depend on the choice of the interpolation nodes $x_0, \ldots, x_n$.

Is there anyway to select these nodes in such a way that the largest possible value of the upper bound is as small as possible?

Function Transformation

Consider an arbitrary interval $[x_0, x_n]$ within the domain of $f(x)$. Denoting by

$$F(x) = f\left(\frac{x(x_n - x_0) + x_0 + x_n}{2}\right)$$

we transfer the function $f(x)$ defined over $[x_0, x_n]$ into the function $F(x)$ within the domain $[-1, 1]$.

Function Transformation

Consider an arbitrary interval $[x_0, x_n]$ within the domain of $f(x)$. Denoting by

$$F(x) = f\left(\frac{x(x_n - x_0) + x_0 + x_n}{2}\right)$$

we transfer the function $f(x)$ defined over $[x_0, x_n]$ into the function $F(x)$ within the domain $[-1, 1]$.

Example. Consider $f(x) = x^2$ on $[-1, 2]$. Then

$$\begin{aligned} F(x) = f\left(\frac{x(2-(-1)) + (-1) + 2}{2}\right) &= f(\frac{3x+1}{2}) \\ &= (\frac{3x+1}{2})^2 \\ &= \frac{9}{4}x^2 + \frac{3}{2}x + \frac{1}{4} \end{aligned}$$

The function $F(x)$ is now defined on $[-1, 1]$.

Denote by

$$R_{n+1}(x) = \prod_{k=0}^{n} (x - x_k)$$

Then from the error formula,

$$\begin{aligned} |e_n(x)| &\leq \frac{c_n}{(n+1)!} \prod_{i=0}^{n} |x - x_i| \\ &\leq \frac{c_n}{(n+1)!} \max_{-1 \leq x \leq 1} |R_{n+1}(x)| \quad \forall x \in [x_0, x_n]. \end{aligned}$$

Denote by

$$R_{n+1}(x) = \prod_{k=0}^{n}(x - x_k)$$

Then from the error formula,

$$\begin{aligned} |e_n(x)| &\leq \frac{c_n}{(n+1)!}\prod_{i=0}^{n}|x - x_i| \\ &\leq \frac{c_n}{(n+1)!}\max_{-1\leq x\leq 1}|R_{n+1}(x)| \quad \forall x \in [x_0, x_n]. \end{aligned}$$

Goal: control this error band by choosing a set of nodes $x_i, i = 0, \ldots, n$ which would minimize the largest deviation of $R_{n+1}(x)$ from zero over $[-1, 1]$. That is, our objective to minimize

$$\max_{-1\leq x\leq 1}|R_{n+1}(x)|$$

by selecting an appropriate set of interpolating nodes. This leads to a discussion on Chebyshev polynomials.

Chebyshev polynomials

Theorem

For any $n = 0, 1, \ldots,$ *function* $T_n(x)$ *defined by*

$$T_n(x) = \cos(n \arccos x)$$

is a polynomial of degree n *(called the* n*th Chebyshev polynomial).*

Moreover,

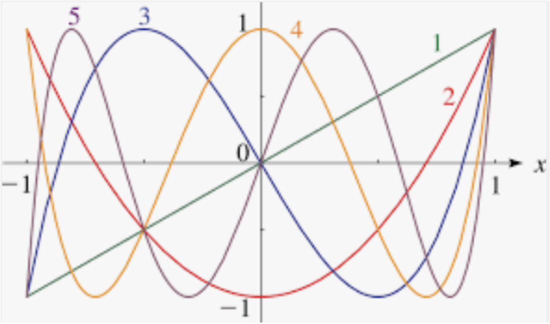
$$T_0(x) = 1, \quad T_1(x) = x$$

and for $n \geq 2$, $T_n(x)$ *can be found recursively by the following relation:*

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x).$$

Chebyshev polynomials $T_0(x)$ to $T_5(x)$

$$
\begin{aligned}
T_0(x) &= 1 \\
T_1(x) &= x \\
T_2(x) &= 2xT_1(x) - T_0(x) = 2x^2 - 1 \\
T_3(x) &= 2xT_2(x) - T_1(x) = 4x^3 - 3x \\
T_4(x) &= 8x^4 - 8x^2 + 1 \\
T_5(x) &= 16x^5 - 20x^3 + 5x
\end{aligned}
$$



Given

$$T_n(x) = \cos(n \arccos x), \quad n = 0, 1, \ldots$$

- $T_0(x) = \cos 0 = 1$

Given

$$T_n(x) = \cos(n \arccos x), \quad n = 0, 1, \ldots$$

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

Given

$$T_n(x) = \cos(n \arccos x), \quad n = 0, 1, \dots$$

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

  $$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

  set

  $$\alpha = \arccos x$$

  and use the identity

  $$\cos((n+1)\alpha) = \cos(n\alpha + \alpha) = 2\cos\alpha\cos(n\alpha) - \cos((n-1)\alpha), \quad n = 1, 2, \dots$$

  leads to

  $$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

Properties of Chebyshev polynomials

Coefficient of dominant terms.
If $n \geq 1$, then the coefficient of x^n in $T_n(x)$ is 2^{n-1}.

$$
\begin{aligned}
T_1(x) &= x \qquad (2^{1-1}) \\
T_2(x) &= 2x^2 - 1 \qquad (2^{2-1}) \\
T_3(x) &= 4x^3 - 3x \qquad (2^{3-1}) \\
T_4(x) &= 8x^4 - 8x^2 + 1 \qquad (2^{4-1}) \\
T_5(x) &= 16x^5 - 20x^3 + 5x \qquad (2^{5-1})
\end{aligned}
$$

Roots of Chebyshev polynomials.

$T_n(x)$ has n distinct roots $x_0, x_1, \ldots, x_{n-1} \in [-1, 1]$ called Chebyshev nodes, which are given by

$$x_k = \cos\Big(\frac{(2k+1)\pi}{2n}\Big), \quad k = 0, 1, \ldots, n-1$$

$$\begin{aligned} x_0 &= \cos(\frac{\pi}{2n}) \\ x_1 &= \cos(\frac{3\pi}{2n}) \\ x_2 &= \cos(\frac{5\pi}{2n}), etc \end{aligned}$$

- The deviation of $T_n(x)$ from zero on $[-1, 1]$ is bounded by

$$\max_{-1 \leq x \leq 1} |T_{n+1}(x)| = 1.$$

- The deviation of $T_n(x)$ from zero on $[-1, 1]$ is bounded by

$$\max_{-1 \le x \le 1} |T_{n+1}(x)| = 1.$$

- $T_n(x)$ is an even function for $n = 2k$, and an odd function for $n = 2k + 1$, that is

$$T_{2k}(-x) = T_{2k}(x), \qquad T_{2k+1}(-x) = -T_{2k+1}(x), \quad k = 0, 1, \ldots$$

- The deviation of $T_n(x)$ from zero on $[-1, 1]$ is bounded by

$$\max_{-1 \leq x \leq 1} |T_{n+1}(x)| = 1.$$

- $T_n(x)$ is an even function for $n = 2k$, and an odd function for $n = 2k + 1$, that is

$$T_{2k}(-x) = T_{2k}(x), \qquad T_{2k+1}(-x) = -T_{2k+1}(x), \quad k = 0, 1, \ldots$$

- $T_{n+1}(x)/2^n$ is polynomial of degree n and leading coefficient 1, has the smallest deviation from zero over $[-1, 1]$ among all normalized polynomials of degree $n + 1$.

Theorem

There is a **unique** *polynomial of degree* $n+1$ *of the form*

$$T_{n+1}(x)/2^n$$

such that for any polynomial $R_{n+1}(x)$ *of degree* $n+1$ *(with the coefficient for* x^{n+1} *equal to* 1*), the following property holds:*

$$\max_{-1\leq x\leq 1}\{|T_{n+1}(x)|\} \leq \max_{-1\leq x\leq 1}\{|R_{n+1}(x)|\}$$

Any polynomial of degree $n+1$ with leading coefficient 1 can be represented as

$$R_{n+1}(x) = (x - x_0)(x - x_1)\ldots(x - x_n).$$

The Theorem implies that among all choices of nodes $x_k \in [-1, 1]$, $k = 0, 1, \ldots, n$ the maximum deviation of $R_{n+1}(x)$ from zero,

$$\max_{-1 \leq x \leq 1} \{|R_{n+1}(x)|\}$$

is minimized if $x_k, k = 0, \ldots, n$ are chosen as the roots of $T_{n+1}(x)$ given by

$$x_k = \cos\Big(\frac{\pi + 2\pi k}{2n}\Big), \quad k = 0, 1, \ldots, n$$

Example

This figure shows interpolating polynomial for $f(x) = \frac{1}{1+10x^2}$ over $[-1, 1]$ based on two distinct sets of nodes. One is based on 11 equally spaced nodes (dotted), whereas the other one is based on the Chebyshev nodes (dashed). As can be seen from the figure, the polynomial based on the Chebyshev nodes provides a much better overall approximation of $f(x)$ over $[-1, 1]$.