Numerical Analysis

Polynomial Interpolation

Polynomial

Definition (a polynomial (from $\mathbb{R}$ to $\mathbb{R}$))

A polynomial, from $\mathbb{R} \to \mathbb{R}$, is a function of the form

$$P_n(x) = \sum_{j=0}^{n} a_j x^j = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

where n is a nonnegative integer and $(a_j)_{j=0}^{n}$ are real numbers.

Using polynomials for approximating functions

- The importance of polynomials, for representing/approximating functions stems from the ease of computations of

$$x \mapsto P_n(x)\,, \quad \frac{d}{dx}P_n(x) \quad \text{and} \quad \int P_n(x)\,dx$$

and from the fact they (uniformly) approximate continuous functions (see Theorem below)

Weierstrass theorem

Theorem (Weierstrass)

Every continuous function defined on a closed interval can be uniformly approximated as closely as desired by a polynomial function. More formally, let $f \in C([a, b])$*. Then for every* $\varepsilon > 0$ *there exists a polynomial,* P_n*, such that, for every* $x \in [a, b]$*,*

$$|f(x) - P_n(x)| < \varepsilon \,.$$

Polynomial interpolation problem definition

The fundamental problem:

Definition (the polynomial interpolation problem)

Given $(n+1)$ points,

$$((x_i, y_i))_{i=0}^{n} \qquad x_i \in \mathbb{R}\,,\ y_i \in \mathbb{R} \quad \forall\, i \in \{0, 1, \ldots, n\}$$

such that the x_i's are **distinct**, namely,

$$i \neq j \Rightarrow x_i \neq x_j\,,$$

find a polynomial, $P(x)$, of **minimal degree**, such that "P **interpolates the data**"; *i.e.*

$$\boxed{P(x_i) = y_i \quad \forall\, i \in \{0, 1, \ldots, n\}}\,.$$

Example

- Assume the setting from the previous slide with the additional assumption that $y_i = c$ for every i, where $c \in \mathbb{R}$ is some constant.
- The zeroth-order polynomial,

$$P(x) \equiv c\,,$$

  interpolates the data.
- Since its degree is zero, it is also minimal.
- Clearly, if $c' \in \mathbb{R}$ is some other constant, with $c \neq c'$, then the zeroth-order polynomial $Q(x) \equiv c'$ does not interpolate the data, since, for every i,

$$Q(x_i) = c' \neq c = y_i\,.$$

- It follows that $P(x) \equiv c$ is the the unique polynomial that satisfies the requirements.

Some observations from the example

- In the definition of polynomial interpolation, only the x_i's are required to be distinct, not the y_i's.
- We just saw, in that particular example, existence and uniqueness of the interpolation polynomial. As will see, the interpolation polynomial always exists and is always unique, not just in that example.

Existence and uniqueness of polynomial

Theorem

Given $((x_i, y_i))_{i=0}^{n}$, *with* $x_i \neq x_j$ *whenever* $i \neq j$, *there exists a unique polynomial* P_n, *of degree no greater than* n, *such that*

$$P(x_i) = y_i \quad \forall i \in \{0, 1, \ldots, n\}.$$

Proof – uniqueness.

A proof by contradiction. Suppose there are two different polynomials, P_n and Q_n, such that

$$P_n(x_i) = y_i = Q_n(x_i) \quad \forall\, i \in \{0, 1, \ldots, n\}\,.$$

Thus, defining a third polynomial by $S(x) = P_n(x) - Q_n(x)$ we get

$$S(x_i) = P_n(x_i) - Q_n(x_i) = 0 \quad \forall\, i \in \{0, 1, \ldots, n\}\,.$$

This implies that S has at least $n+1$ distinct roots. By construction, the degree of S is no higher than n. By the fundamental theorem of algebra, a polynomial of degree n has exactly n roots (including complex roots, and including multiplicities), **unless** it is the zero polynomial. It follows that $S(x) \equiv 0$, and thus $P_n \equiv Q_n$, a contradiction. $\square$

[Proof – existence]

A proof by induction.

- Base. $n = 0$, given (x_0, y_0). Choose

$$P_0(x) \equiv C_0 \triangleq y_0 \,.$$

- Assumption. Suppose there exists a polynomial, P_{k-1}, of degree no higher than $k-1$, such that it interpolates the first k points:

$$P_{k-1}(x_i) = y_i \quad \forall\, i \in \{0, 1, \dots, k-1\} \,.$$

- Step. We will now build a polynomial, P_k, of degree no higher than k, that interpolates the first $k+1$ points:

$$P_k(x_i) = y_i \quad \forall\, i \in \{0, 1, \dots, k\} \,.$$

(continue to next slide)

Proof – existence.

We do this by setting

$$\boxed{P_k(x) = P_{k-1}(x) + C_k \prod_{j=0}^{k-1}(x - x_j)}.$$

Now, $\prod_{j=0}^{k-1}(x - x_j)$ is a polynomial, of degree k, that vanishes on $(x_j)_{j=0}^{k-1}$. So P_k interpolates the first k points, and its degree is no higher than k. We still need it to interpolate (x_k, y_k), the $(k+1)$-th point. As for C_k:

$$y_k \overset{\text{required}}{=} P_k(x_k) = P_{k-1}(x_k) + C_k \underbrace{\prod_{j=0}^{k-1}(x_k - x_j)}_{\neq 0}$$

$$\Rightarrow \boxed{C_k = \frac{y_k - P_{k-1}(x_k)}{\prod_{j=0}^{k-1}(x_k - x_j)}}$$

$\square$

Newton form

Definition (Newton's form of the interpolation polynomial)

The form we saw in the proof,

$$\begin{aligned} P_k(x) = & C_0 + C_1(x - x_0) + C_2(x - x_0)(x - x_1) + \dots \\ & + C_k(x - x_0) \cdots (x - x_{k-1}), \end{aligned}$$

is called Newton's form of the interpolation polynomial.

Example 1

$$\boxed{P_k(x) = \begin{cases} C_0 & k = 0 \\ P_{k-1}(x) + C_k \prod_{j=0}^{k-1}(x - x_j) & k \in \{1, \ldots, n\} \end{cases}}$$

where $\boxed{C_k = \begin{cases} y_0 & k = 0 \\ \frac{y_k - P_{k-1}(x_k)}{\prod_{j=0}^{k-1}(x_k - x_j)} & k \in \{1, \ldots, n\} \end{cases}}$,

Example ($n = 0$ (a single point))

Given a single point, (x_0, y_0), the interpolation polynomial is

$$P_0(x) = C_0 = y_0$$

Example 2

$$\boxed{P_k(x) = \begin{cases} C_0 & k = 0 \\ P_{k-1}(x) + C_k \prod_{j=0}^{k-1}(x - x_j) & k \in \{1, \ldots, n\} \end{cases}}$$

where $\boxed{C_k = \begin{cases} y_0 & k = 0 \\ \frac{y_k - P_{k-1}(x_k)}{\prod_{j=0}^{k-1}(x_k - x_j)} & k \in \{1, \ldots, n\} \end{cases}}$,

Example ($n = 1$ (two points))

Given two points, $((x_i, y_i))_{i=0}^{1}$, the interpolation polynomial is (the line)

$$P_1(x) = C_0 + C_1(x - x_0) = \overbrace{y_0}^{C_0} + \overbrace{\frac{y_1 - y_0}{x_1 - x_0}}^{C_1} \cdot (x - x_0)$$

$$P_1(x_0) = y_0 + \frac{y_1 - y_0}{x_1 - x_0} \cdot (x_0 - x_0) = y_0$$

$$P_1(x_1) = y_0 + \frac{y_1 - y_0}{x_1 - x_0} \cdot (x_1 - x_0) = y_1$$

Polynomial interpolation by solving linear system

- Here is another way to derive the interpolation polynomial.
- The $n+1$ equations,

$$P_n(x_i) = \sum_{j=0}^{n} a_j x_i^j = \begin{bmatrix} 1 & x_i & x_i^2 & \dots & x_i^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = y_i \, , \; i = 0, 1, \dots, n$$

are linear w.r.t. the a_i's:

$$\begin{bmatrix} \sum_{j=0}^n a_j x_0^j \\ \sum_{j=0}^n a_j x_1^j \\ \vdots \\ \sum_{j=0}^n a_j x_n^j \end{bmatrix} = \begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & & & & \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Polynomial interpolation by solving linear system

$$\underbrace{\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & & & & \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix}}_{\boldsymbol{V}:(n+1)\times(n+1)} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}}_{\boldsymbol{a}:(n+1)\times 1} = \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}}_{\boldsymbol{y}:(n+1)\times 1}$$

- So can solve via $\boldsymbol{a} = \boldsymbol{V}^{-1}\boldsymbol{y}$.
- $\boldsymbol{V}$ is called the Vandermonde Matrix.
- Assuming the x_i's are distinct, we know that $\boldsymbol{V}$ is guaranteed (by the theorem) to be invertible.
- By uniqueness, we know the solution will coincide with Newton's form.

Problems with this method

- **Computationally expensive.** Matrix inversion is expensive; that said, in the reminder of the course we will study practical methods to solve a linear system without inversion.
- **Ill-conditioned.** The values of $\boldsymbol{a} = \begin{bmatrix} a_0 & a_1 & a_2 & \dots & a_n \end{bmatrix}^T$ might be determined inaccurately. We will later see how to characterize the condition number of a linear system.

Remark

- Recall that one advantage of polynomials is the ease of their evaluations.
- By inspection of $P_n(x) = \sum_{j=0}^{n} a_j x_i^j$, it seems that $x \mapsto P_n(x)$ requires n additions and $\sum_{j=1}^{n} j = n\frac{n+1}{2}$ multiplications. This is $O(n^2)$.
- It is possible, however, to reduce the $\#$ of operations to $O(n)$, using Horner's rule (see next slide).

Horner's Rule

- Set $b_n \triangleq a_n$, and then, iteratively, set $b_{k-1} \triangleq a_{k-1} + b_k x$ till obtaining $b_0 = a_0 + b_1 x$ and this is equal to $P_n(x)$.

Example

$$P_4(x) = \sum_{j=0}^{4} a_j x^j = \underbrace{a_0 + x \overbrace{\left(a_1 + x \underbrace{\left(a_2 + x \overbrace{\left(a_3 + \underbrace{a_4}_{b_4} x\right)}^{b_3}\right)}_{b_2}\right)}^{b_1}}_{b_0}$$

- This gives $O(n)$ additions and $O(n)$ multiplications (instead of $O(n)$ and $O(n^2)$).

- Since the interpolation polynomial is unique, if the $n+1$ points were sampled from a polynomial of degree n, then we will recover that polynomial.

Example

Let $P(x) = 2x^3 + 3x^2 - 4x - 5$. Consider the following 4 sampled points.

	$i=0$	$i=1$	$i=2$	$i=3$
x_i	-1	0	2	3
y_i	0	-5	15	64

Reminder: Newton's form

$$P_k(x) = \begin{cases} C_0 & k=0 \\ P_{k-1}(x) + C_k \prod_{j=0}^{k-1}(x - x_j) & k \in \{1,\dots,n\} \end{cases}$$

$$\text{where } C_k = \begin{cases} y_0 & k=0 \\ \frac{y_k - P_{k-1}(x_k)}{\prod_{j=0}^{k-1}(x_k - x_j)} & k \in \{1,\dots,n\} \end{cases}$$

$$P_0(x) \equiv C_0 = y_0 = 0$$

Example (continued)

Let $P(x) = 2x^3 + 3x^2 - 4x - 5$. Consider the following 4 sampled points.

	$i=0$	$i=1$	$i=2$	$i=3$
x_i	-1	0	2	3
y_i	0	-5	15	64

Reminder: Newton's form

$$P_k(x) = \begin{cases} C_0 & k = 0 \\ P_{k-1}(x) + C_k \prod_{j=0}^{k-1}(x - x_j) & k \in \{1, \ldots, n\} \end{cases}$$

$$\text{where } C_k = \begin{cases} y_0 & k = 0 \\ \frac{y_k - P_{k-1}(x_k)}{\prod_{j=0}^{k-1}(x_k - x_j)} & k \in \{1, \ldots, n\} \end{cases}$$

$P_0(x) \equiv 0$ (saw in the previous slide)

$C_1 = \frac{y_1 - P_0(x_1)}{x_1 - x_0} = \frac{-5}{0+1} = -5$

$P_1(x) = P_0(x) + C_1(x - x_0) = 0 - 5(x + 1) = -5x - 5$

Example (continued)

Let $P(x) = 2x^3 + 3x^2 - 4x - 5$. Consider the following 4 sampled points.

	$i=0$	$i=1$	$i=2$	$i=3$
x_i	-1	0	2	3
y_i	0	-5	15	64

Reminder: Newton's form

$$P_k(x) = \begin{cases} C_0 & k=0 \\ P_{k-1}(x) + C_k \prod_{j=0}^{k-1}(x - x_j) & k \in \{1,\ldots,n\} \end{cases}$$

$$\text{where } C_k = \begin{cases} y_0 & k=0 \\ \frac{y_k - P_{k-1}(x_k)}{\prod_{j=0}^{k-1}(x_k - x_j)} & k \in \{1,\ldots,n\} \end{cases}$$

$P_1(x) = -5x - 5$ (saw in the previous slide)

$$C_2 = \frac{y_2 - P_1(x_2)}{(x_2 - x_0)(x_2 - x_1)} = \frac{15-(-10-5)}{(2+1)(2)} = \frac{30}{6} = 5$$

$$P_2(x) = P_1(x) + C_2(x - x_0)(x - x_1) = 5x^2 - 5$$

Example (continued)

Let $P(x) = 2x^3 + 3x^2 - 4x - 5$. Consider the following 4 sampled points.

	$i=0$	$i=1$	$i=2$	$i=3$
x_i	-1	0	2	3
y_i	0	-5	15	64

Reminder: Newton's form

$$P_k(x) = \begin{cases} C_0 & k=0 \\ P_{k-1}(x) + C_k \prod_{j=0}^{k-1}(x - x_j) & k \in \{1, \dots, n\} \end{cases}$$

$$\text{where } C_k = \begin{cases} y_0 & k=0 \\ \frac{y_k - P_{k-1}(x_k)}{\prod_{j=0}^{k-1}(x_k - x_j)} & k \in \{1, \dots, n\} \end{cases}$$

$P_2(x) = 5x^2 - 5$ (saw in the previous slide)

$C_3 = \frac{y_3 - P_2(x_3)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} = \frac{64 - (5 \cdot 9 - 5)}{(3+1)(3)(3-2)} = \frac{24}{12} = 2$

$P_3(x) = P_2(x) + C_3(x - x_0)(x - x_1)(x - x_2) = 5x^2 - 5 + 2(x+1)(x)(x-2)$

$= 5x^2 - 5 + (2x^2 + 2x)(x - 2) = 5x^2 - 5 + 2x^3 + 2x^2 - 4x^2 - 4x$

$= 2x^3 + 3x^2 - 4x - 5$ as expected.

$P_n(x)$ as a Linear combination of Basis Functions

- An alternative way to represent the interpolation polynomial is as the following linear combination of basis functions.

$$P_n(x) = \sum_{i=0}^{n} y_i L_i(x)$$

where $(L_i(x))_{i=0}^{n}$ are polynomials that depend on the values of x_i's but not on the values of the y_i's.

- Since we can't control the y_i's, the most general constraint on $L_i(x)$ is:

$$L_i(x_j) = \delta_{ij} \triangleq \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad \text{(Kronecker delta)}$$

together with the restriction that $L_i(x)$ is a polynomial of degree no higher than n.

$P_n(x)$ as a Linear combination of Basis Functions

$$P_n(x) = \sum_{i=0}^{n} y_i L_i(x)$$

- The constraint "$i \neq j \Rightarrow L_i(x_j) = 0$" implies the following form:

$$L_i(x) = C_i \prod_{j:j \neq i} (x - x_j)$$

where the value of C_i is found via

$$L_i(x_i) = 1 = C_i \prod_{j:j \neq i} (x_i - x_j) \Rightarrow C_i = \prod_{j:j \neq i} (x_i - x_j)^{-1}$$

$P_n(x)$ as a Linear combination of Basis Functions

- In other words, we have just derived the family of the basis functions, $(L_i(x))_{i=0}^n$:

$$L_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} = \left(\prod_{j=0}^{i-1} \frac{x - x_j}{x_i - x_j} \right) \left(\prod_{j=i+1}^{n} \frac{x - x_j}{x_i - x_j} \right)$$

- And as mentioned earlier, the polynomial

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

satisfies

$$P_n(x_i) = y_i$$

so by uniqueness of the interpolation polynomial, it is the same polynomial we saw in other forms.

$P_n(x)$ as a Linear combination of Basis Functions

- This is Lagrange's form of the interpolation polynomial. It is often called Lagrange Approximation, but of course if the data came from a polynomial (of degree no higher than n) then it is exact.
- Remark: This form is easy to construct by is expensive to evaluate.

Example

Again let $P(x) = 2x^3 + 3x^2 - 4x - 5$ and consider

	$i=0$	$i=1$	$i=2$	$i=3$
x_i	-1	0	2	3
y_i	0	-5	15	64

Lagrange's form of the interpolation polynomial is given by

$$P_n(x) = \sum_{i=0}^{3} y_i L_i(x) = 0 \cdot L_0(x) - 5L_1(x) + 15L_2(x) + 64L_3(x)$$

$$L_0(x) = \text{ don't bother since } y_0 = 0$$

$$L_1(x) = \tfrac{(x+1)(x-2)(x-3)}{(0+1)(0-2)(0-3)} = \tfrac{1}{6}(x+1)(x-2)(x-3)$$

$$L_2(x) = \tfrac{(x+1)x(x-3)}{(2+1)(2-0)(2-3)} = -\tfrac{1}{6}(x+1)x(x-3)$$

$$L_3(x) = \tfrac{(x+1)x(x-2)}{(3+1)(3-0)(3-2)} = \tfrac{1}{12}(x+1)x(x-2)$$

$$P_n(x) = \tfrac{-5}{6}(x+1)(x-2)(x-3) - \tfrac{15}{6}(x+1)x(x-3) + \tfrac{32}{6}(x+1)x(x-2)$$

$$= \ldots = 2x^3 + 3x^2 - 4x - 5 \text{ as expected.}$$

The Interpolation Polynomial's Error

- Our examples so far involved the (exact) recovering of a polynomial.
- Polynomial interpolation, however, can also be used to **approximate** non-polynomials.

Example (approximating $\sin(x)$)

Let $f(x) = \sin(x)$. It is easy to evaluate f in several key points.

	$i=0$	$i=1$	$i=2$	$i=3$	$i=4$
x_i	$-\pi$	$-\pi/2$	0	$\pi/2$	π
y_i	0	-1	0	1	0

Can now apply the tools we learned to find a polynomial, of degree ≤ 4, that approximates $\sin(x)$ on $[-\pi, \pi]$. We will get the following result:

$$P_4(x) = \frac{-8}{3\pi^3}x^3 + \frac{8}{3\pi}x$$

Exercise

Verify this at home.

A natural question: how good is this approximation?

The Interpolation Polynomial's Error

Theorem

Let $f \in C^{n+1}[a,b]$ and let $P_n(x)$ be the interpolation polynomial of f at nodes $(x_i)_{i=0}^{n} \subset [a,b]$. Then, the interpolation error at $x \in [a,b]$ is given by

$$\boxed{E_n(x) \triangleq f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i)}$$

for some $\xi \in [a,b]$.

- This theoretical result is of paramount importance since it lets us known, a-priori, the expected maximal error and enables us to design an interpolation polynomial to meet a given accuracy criterion.

Example

Find the degree of the interpolation polynomial, $P_n(x)$ that will guarantee that approximation error will be $\leq 10^{-5}$ for $f(x) = \sin(x)$ on $[-\pi, \pi]$ (even w/o knowing the nodes, $(x_i)_{i=1}^{n}$).

Solution:

$$|E_n(x)| = \left| \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i) \right|$$

$$\leq \left| \frac{f^{(n+1)}(\xi)}{(n+1)!} \right| (2\pi)^{n+1} \leq \frac{(2\pi)^{n+1}}{(n+1)!} \overset{\text{required}}{\leq} 10^{-5}$$

By search we can find $|E_{25}(x)| \leq M \approx 5.8 \cdot 10^{-6}$. Note, however, that by using a better placement of the nodes, we will be able to get the desired accuracy using a much smaller n.

Bounding the error

- We are now in a position to discuss a bound on the error.
- Clearly, the theorem is useful only if we can bound $f^{(n+1)}(x)$.
- **If** we can assume, or prove, a bound on $|f^{(n+1)}(x)|$ (as we could in the example with $f(x) = \sin(x)$),

$$|f^{(n+1)}(x)| < M_{n+1} \quad \forall x \in [a, b]$$

then

$$\boxed{|E_n(x)| = |f(x) - P_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \max_{x \in [a,b]} \prod_{j=0}^{n} |x - x_j|}$$

Evenly-spaced Nodes

- Often, it is convenient to sample the function at evenly-spaced points:

$$x_i = x_0 + i \cdot h \quad \forall i = 1, \ldots, n$$

- In which case, it is possible to derive more convenient error bounds.

Example

Consider a quadratic interpolation on $[a,b]$, where $x_0 = a$, $x_1 = a+h$, and $x_2 = a+2h = b$. By the theorem, if M_3 is an upper bound on $|f^{(3)}(x)|$,

$$E_2(x) \le \frac{M_3}{(3)!} \max_{x\in[a,b]} |(x-x_0)(x-x_1)(x-x_2)| \triangleq \frac{M_3}{3!} \max_{x\in[a,b]} |H(x)|$$

$$H(x) = (x-x_0)(x-x_1)(x-x_2) =$$

$$\ldots = x^3 - x^2(x_0+x_1+x_2) - x(x_0x_1 + x_0x_2 + x_1x_2) - x_0x_1x_2$$

Looking for the extremum, we set $H'(x) = 3x^2 - 2x(x_0+x_1+x_2) - (x_0x_1 + x_0x_2 + x_1x_2) = 0$. Now substitute $x_i = x_0 + i\cdot h$ and obtain

$$0 = 3x^2 - 6x(x_0+h) + 3(x_0^2 + 2x_0h + \tfrac{2}{3}h^2)$$

$$x_{\max} = \frac{6(x_0+h) \pm \sqrt{36(x_0+h)^2 - 36(x_0^2 + 2x_0h + \frac{2}{3}h^2)}}{6} = x_0 + h \pm \sqrt{\frac{1}{3}h^2}$$

$$= x_0 + h(1 \pm \frac{1}{\sqrt{3}})$$

(continue next slide)

Example (continued)

Going back to $\max_{x\in[a,b]}|H(x)|$, we get

$$H_{\max} = (x_{\max} - x_0)(x_{\max} - x_0 - h)(x_{\max} - x_0 - 2h)$$
$$= h(1 \pm \frac{1}{\sqrt{3}})h(\pm\frac{1}{\sqrt{3}})h(\pm\frac{1}{\sqrt{3}} - 1)$$
$$= h^3\frac{\sqrt{3}+1}{\sqrt{3}}\frac{1}{\sqrt{3}}\frac{\sqrt{3}-1}{\sqrt{3}} = \frac{2}{3\sqrt{3}}h^3$$
$$\Rightarrow |E_2(x)| \leq \frac{M_3}{3!}\max_{x\in[a,b]}|H(x)| = \frac{h^3}{9\sqrt{3}}M_3$$

More generally:

Theorem

Let $f \in C^{n+1}([a,b])$ and let $P_n(x)$ be its interpolation polynomial for the evenly-spaced nodes $x_i = X_0 + ih$, $i = 0, 1, \ldots, n$. If

$$|f^{(n+1)}(x)| \leq M_{n+1} \quad \forall x \in [a,b]$$

then

$$\boxed{|E_n(x)| = |f(x) - P_n(x)| \leq O(h^{n+1})M_{n+1}}$$

We omit the proof. Some particular cases:

- $|E_0(x)| \leq hM_1$
- $|E_1(x)| \leq \frac{1}{8}h^2M_2$
- $|E_2(x)| \leq \frac{1}{9\sqrt{3}}h^3M_3$
- $|E_3(x)| \leq \frac{1}{24}h^4M_4$