Polynomial Interpolation

Outline

- Background, problem statement and motivation
- Polynomial bases
  - Monomial
  - Lagrange
  - Newton
- Uniqueness of polynomial interpolation
- Error analysis for polynomial interpolation
- Stability of the polynomial interpolation

Problem statement and motivation

We consider a collection of data samples $\{(x_i, y_i)\}_{i=0}^n$.

- The $\{x_i\}_{i=0}^n$ are called the *abscissae* (singular: *abscissa*), and the $\{y_i\}_{i=0}^n$ are called the *data values*.
- We want to find a function $p(x)$ which can be used to estimate $y(x)$ for $x \neq x_i$.
- **Why?** We often get discrete data from sensors or computation, but we want information as if the function were not discretely sampled.
- If possible, $p(x)$ should be inexpensive to evaluate for a given x.

There are lots of ways to define a function $p(x)$ to approximate $\{(x_i, y_i)\}_{i=0}^n$.

- *Interpolation* means $p(x_i) = y_i$ (and we will only evaluate $p(x)$ for $\min_i x_i \leq x \leq \max_i x_i$).
- Most interpolants (and even general data fitting) are done with a linear combination of (usually nonlinear) basis functions $\{\phi_j(x)\}$:

$$p(x) = p_n(x) = \sum_{j=0}^{n} c_j \, \phi_j(x)$$

where c_j are the interpolation coefficients or interpolation weights.

Problem statement and motivation

Our interpolant is $p(x) = \sum_{j=0}^{n} c_j \, \phi_j(x)$.

- From the interpolation condition:

$$p(x_i) = \sum_{j=0}^{n} c_j \, \phi_j(x_i) = y_i \quad \text{for } i = 0, 1, \ldots, n,$$

  which leads to the linear system $Ac = y$ where

$$A = \begin{bmatrix} \phi_0(x_0) & \phi_1(x_0) & \cdots & \phi_n(x_0) \\ \phi_0(x_1) & \phi_1(x_1) & \cdots & \phi_n(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_n) & \phi_1(x_n) & \cdots & \phi_n(x_n) \end{bmatrix}, \quad c = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}, \quad y = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

- When can we accurately solve this linear system?

Conditioning of a linear system

Consider solving the linear system $Ax = b$ given A and b.

- Due to approximation error, we actually use a slight perturbation $\widehat{A} = A + \Delta A$.
- The solution to the perturbed system is $\widehat{x} = x + \Delta x$.
- So $b = \widehat{A}\widehat{x} = A\widehat{x} + \Delta A\widehat{x}$ implies $b - A\widehat{x} = \Delta A\widehat{x}$.
- Then $\Delta x = \widehat{x} - x = A^{-1}A\widehat{x} - A^{-1}b = A^{-1}(A\widehat{x} - b) = -A^{-1}\Delta A\widehat{x}$.
- Now take norms:

$$\|\Delta x\| \leq \|A^{-1}\|\|\Delta A\|\|\widehat{x}\|$$

$$\frac{\|\Delta x\|}{\|\widehat{x}\|} \leq \|A^{-1}\|\|A\|\frac{\|\Delta A\|}{\|A\|} = \mathrm{cond}(A)\frac{\|\Delta A\|}{\|A\|}$$

  where $\mathrm{cond}(A) = \|A^{-1}\|\|A\|$ is the definition of the condition number of matrix A.

- Perturbations on b have a similar effect, so the overall result is roughly (assuming $\|\widehat{x}\| \approx \|x\|$):

$$\frac{\|\Delta x\|}{\|x\|} \lesssim \mathrm{cond}(A)\left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|}\right)$$

Choosing Basis Functions

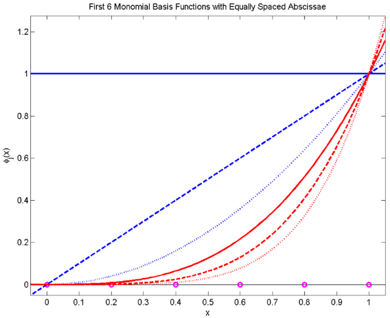
What should we use for the $\{\phi_j\}$?

- Many possibilities: polynomials, trigonometric, exponential, rational (fractions), wavelets / curvelets / ridgelets, radial basis functions, ...
- We will focus for now on polynomial basis functions:
  - Most commonly used
  - Easy to evaluate, integrate, differentiate
  - Illustrates the basic interpolation ideas and techniques
- Virtually all other interpolation problems will follow the same procedure: form A and solve for c.
- Now we are ready to examine particular polynomial bases which are often chosen due to their simplicity, efficiency, numerical robustness, extensibility, etc.

Monomial Basis

The monomial basis functions are typically defined as $\phi_j(x) = x^j$.

- Entries of A are $a_{ij} = \phi_j(x_i) = (x_i)^j$.
- This particular matrix appears so often, it has a name: the *Vandermonde* matrix (in Matlab: `vander`).
- The monomial basis (and the corresponding Vandermonde matrix) are known to be very poorly conditioned as n gets large and/or $\{x_i\}$ cover a large range.
- Notice that $\phi_j(x)$ starts to look very similar to $\phi_{j-1}(x)$ and $\phi_{j+1}(x)$ as j gets larger.



Monomial basis: an example

Construct an interpolant for $\{(x_i, y_i)\} = \{(2, 14), (6, 24), (4, 25), (7, 15)\}$ using the monomial basis.

- Requires four basis functions: $\{\phi_j(x)\} = \{1, x, x^2, x^3\}$.
- The interpolant will be $p(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3$.
- Construct the linear system:

$$A = \begin{bmatrix} 1 & 2 & 4 & 8 \\ 1 & 6 & 36 & 216 \\ 1 & 4 & 16 & 64 \\ 1 & 7 & 49 & 343 \end{bmatrix}, \quad c = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}, \quad y = \begin{bmatrix} 14 \\ 24 \\ 25 \\ 15 \end{bmatrix}$$

  and solve $Ac = y$ to find $c \approx \begin{bmatrix} -0.267 & 1.700 & 2.767 & 3.800 \end{bmatrix}^T$.
- Check $\text{cond}(A) \approx 6.1 \times 10^3$.

Lagrange Basis

The best-conditioned matrix is the identity matrix.

- In order for $A = I$, we need

$$a_{ij} = \phi_j(x_i) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

- We can achieve this by choosing the Lagrange basis functions

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

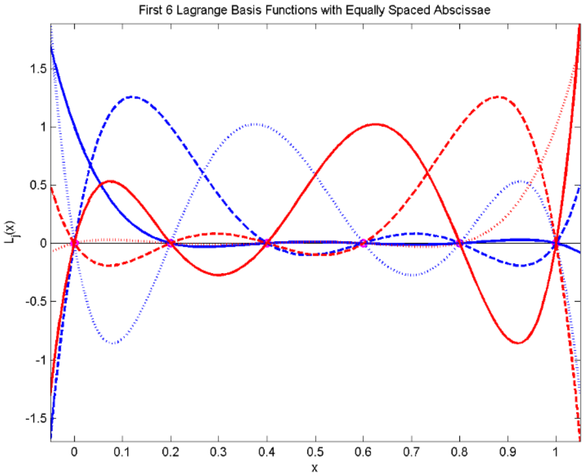
  - The numerator ensures that $a_{ij} = L_j(x_i) = 0$ for $i \neq j$.
  - The denominator normalizes to get $a_{jj} = L_j(x_j) = 1$.

- With $A = I$, there is no need to solve a linear system: $c_j = y_j$ and

$$p(x) = \sum_{j=0}^{n} y_j L_j(x)$$

Lagrange Basis Conditioning

The Lagrange basis functions $L_j(x)$ are clearly distinct. With abscissae $x_i = \frac{i}{5}$ for $i = 0, 1, \ldots, 5$, the Lagrange basis functions $L_j(x)$ are shown below.



Lagrange Basis: an example

Construct an interpolant for $\{(x_i, y_i)\} = \{(2,14),(6,24),(4,25),(7,15)\}$ using the Lagrange basis.

- Four basis functions:

$$L_0(x) = \frac{(x-6)(x-4)(x-7)}{(2-6)(2-4)(2-7)}$$

$$L_1(x) = \frac{(x-2)(x-4)(x-7)}{(6-2)(6-4)(6-7)}$$

$$L_2(x) = \frac{(x-2)(x-6)(x-7)}{(4-2)(4-6)(4-7)}$$

$$L_3(x) = \frac{(x-2)(x-6)(x-4)}{(7-2)(7-6)(7-4)}$$

- The interpolant will be

$$p(x) = 14\frac{(x-6)(x-4)(x-7)}{(-4)(-2)(-5)} + 24\frac{(x-2)(x-4)(x-7)}{(+4)(+2)(-1)}$$

$$+25\frac{(x-2)(x-6)(x-7)}{(+2)(-2)(-3)} + 15\frac{(x-2)(x-6)(x-4)}{(+5)(+1)(+3)}$$

Newton Basis

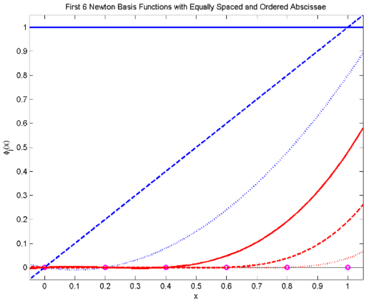
- Can we add a new data point without changing the entire interpolant?
  - Need $n \to n+1$, would prefer well-conditioned and easy to construct and evaluate
- In order to easily add points, we need $\phi_j(x)$ to have certain properties:
  - New basis function cannot disturb prior interpolation: $\phi_j(x_i) = 0$ for $i < j$
  - Old basis function does not need information about new data values: $\phi_j(x)$ is independent of (x_i, y_i) for $i > j$
- Newton basis function

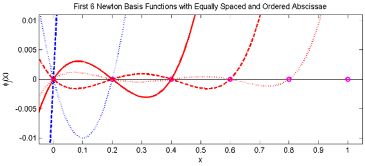
$$\phi_j(x) = \prod_{i=0}^{j-1}(x - x_i)$$

- Leads to special form for matrix A
  - A is not necessarily well-conditioned, but will not be worse than the monomial basis and can be quite good if the order of data values is chosen wisely

Newton Basis Conditioning

- We know that the Newton basis functions are linearly independent because $\phi_j(x)$ has exactly degree $j-1$.
- With abscissae $x_i = \frac{i}{5}$ for $i = 0, 1, \ldots, 5$, the Newton basis functions $\phi_j(x)$ are shown below.
- Visually, they are not as distinct as the Lagrange basis functions but they are better than the monomials.





Newton Basis: an example

- Construct an interpolant for data points

$$\{(x_i, y_i)\} = \{(2, 14), (6, 24), (4, 25), (7, 15)\}$$

  using the Newton basis.
- Four basis functions

$$\phi_0(x) = 1 \quad \phi_1(x) = (x-2)$$

$$\phi_2(x) = (x-6)(x-2) \quad \phi_3(x) = (x-4)(x-6)(x-2)$$

- Construct linear system

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 0 & 0 \\ 1 & 2 & -4 & 0 \\ 1 & 5 & 5 & 15 \end{bmatrix} \quad c = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad y = \begin{bmatrix} 14 \\ 24 \\ 25 \\ 15 \end{bmatrix}$$

  and solve $Ac = y$ to find $c \approx [14 \quad 2.5 \quad -1.5 \quad -0.2667]^T$.
- Check $\text{cond}(A) \approx 17$

Uniqueness of the Polynomial Interpolant

A polynomial interpolant of degree n is unique: any basis will produce the same interpolant (assuming no computational errors).

- Let $q(x)$ and $p(x)$ be two polynomial interpolants of degree n such that the interpolation condition holds:

$$p(x_i) = y_i = q(x_i) \text{ for all } i = 0, 1, \dots, n.$$

- Define $u(x) = q(x) - p(x)$. This is also a polynomial of degree n.
- Note that

$$u(x_i) = q(x_i) - p(x_i) = y_i - y_i = 0 \text{ for all } i = 0, 1, \dots, n,$$

  so $u(x)$ has $n + 1$ zeros.
- The only polynomial of degree n with $n + 1$ zeros is $u(x) \equiv 0$, which implies

$$q(x) = p(x).$$

Interpolation Error Theorem

Theorem (Interpolation Error, Cauchy (1840))

Let $f \in C^{n+1}([a,b])$ and suppose that $x_0, \ldots, x_n \in [a,b]$ are distinct, and that the polynomial p_n satisfies

$$p_n(x_k) = f(x_k), \quad k = 0, \ldots, n.$$

Then

$$E_n[f](x) := f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i),$$

where $\xi \in I$ with I being the smallest open interval containing $x_0, \ldots, x_n$.

Summary of the proof

- Let us introduce $\forall x \in [a,b]$ the following auxiliary function

  $$G(t) = (f(t) - p_n(t))\omega(x) - (f(x) - p_n(x))\omega(t)$$

  where $\omega(x) = \prod_{i=0}^{n}(x - x_i)$.
- $G(t)$ has $n+2$ roots: the $n+1$ interpolation points and the point $t = x$.
- So $G^{(n+1)}(t) = f^{(n+1)}(t)\omega(x) - (f(x) - p_n(x))(n+1)!$ has 2 roots.
- It does exist a point ξ for which $G^{(n+1)}(\xi) = 0$.

Interpolation error: an example

It is observed that while ξ exists, it is not generally possible to determine its exact value. Therefore, it is not possible to specify the exact interpolation error. However, if $M = \max_{s\in[a,b]} |f^{(n+1)}(s)|$ is known, it is still possible to estimate the error as follows:

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

Example. Let $f(x) = \exp(x)$ and let p_4 be the polynomial that interpolates $f(x)$ at the equally spaced points $x_i = -1 + \frac{2i}{4}, \; i = 0, \dots, 4$.

Since $f^{(k)}(x) = \exp(x)$, for $k = 0, 1, \dots,$ and $\exp(x)$ is an increasing function, we have

$$M = \|f^{(5)}\|_\infty = \max_{x\in[-1,1]} |\exp(x)| = \exp(1) \approx 2.7183.$$

Moreover, as can be seen from the graph in the figure, $\prod_{i=0}^{4} |x - x_i| \leq 0.12$. Since $5! = 120$, we have

$$|E_4[f](x)| = \left| \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n}(x - x_i) \right| \leq \frac{\exp(1)\cdot 0.12}{120} \approx 0.0027.$$

Polynomial Interpolation

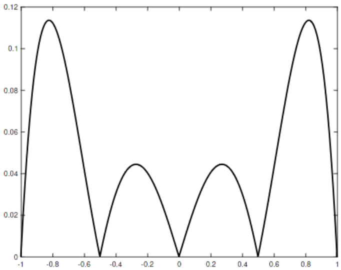


Figure: On the left, the graph of $\prod_{i=0}^{4} |x - x_i|$ where $x_i = -1 + \frac{2i}{4}$, $i = 0, \ldots, 4$.

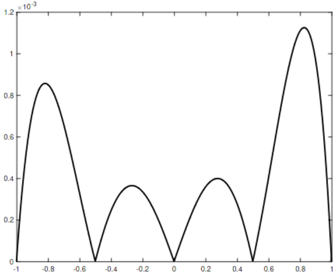


Figure: On the right, the graph of the error $E_n[f]$ in $[-1, 1]$ which is evidently always less than $2.7 \cdot 10^{-3}$.

Polynomial Interpolation Convergence

One relevant question is whether increasing the number of nodes causes the sequence of interpolating polynomials p_n to converge uniformly to the function f being approximated, that is,

$$\lim_{n\to\infty} \max_{x\in[a,b]} |f(x) - p_n(x)| = 0$$

where $-\infty < a < b < +\infty$.
We observe that the interpolation error estimate yields, since $\xi \in (a, b)$,

$$\max_{x\in[a,b]} |f(x) - p_n(x)| \leq \max_{s\in[a,b]} \left|f^{(n+1)}(s)\right| \max_{x\in[a,b]} \frac{\left|\prod_{i=0}^{n}(x - x_i)\right|}{(n+1)!}.$$

if $\max_{s\in[a,b]} \left|f^{(n+1)}(s)\right|$ is sufficiently large for each n, then it could happen that

$$\max_{s\in[a,b]} \left|f^{(n+1)}(s)\right| \max_{x\in[a,b]} \frac{\left|\prod_{i=0}^{n}(x - x_i)\right|}{(n+1)!}$$

is not infinitesimal as $n \to +\infty$, and thus it may happen that

$$\|E_n[f]\|_\infty := \max_{x\in[a,b]} |f(x) - p_n(x)|$$

does not converge to 0.

Polynomial Interpolation Convergence

In general,

Theorem (Faber's Theorem (1914))

For any distribution of nodes, there exists at least one function $f \in C([a, b])$, $-\infty < a < b < +\infty$, such that the interpolation error $\|E_n[f]\|_\infty$ does not converge to 0 as $n \to +\infty$.

On the other hand,

Theorem

For every function $f \in C([a, b])$, $-\infty < a < b < +\infty$, there exists at least one distribution of nodes such that $\|E_n[f]\|_\infty \to 0$ as $n \to +\infty$.

Runge function

Example

Let f be the Runge function (discovered in 1901)

$$f(x) = \frac{1}{1+x^2}, \quad x \in [-5, 5].$$

It is demonstrated that the polynomial p_n which interpolates f at $n+1$ equally spaced nodes does not converge uniformly to f

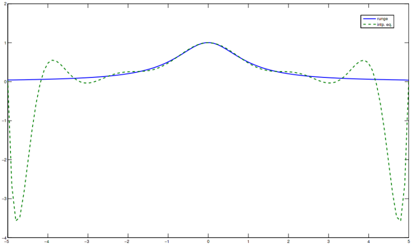


Figure: Graph illustrating the interpolating polynomial of degree 12 over 13 equally spaced nodes of Runge's function. The function is shown with a solid line, and the interpolating polynomial is shown with a dashed line. Notice the oscillations at the extremes.

Runge function

What's happening?

Runge Theorem

If f is analytically extensible to an oval $\mathcal{O}(a, b, R)$ with $R > 0$ we have that

$$||f^{n+1}||_\infty \leq \frac{(n+1)!}{R^{n+1}} ||\widetilde{f}||_\infty$$

where $\widetilde{f}$ denotes the analytical extension of f.

Then the interpolation error gives

$$||E_n[f]||_\infty \leq \left(\frac{b-a}{R}\right)^{n+1} ||\widetilde{f}||_\infty$$

being $||\omega(x)||_\infty \leq |b-a|^{n+1}$. The extension of the Runge function at the complex plane has two roots, $+i$ and $-i$. so $R < 1$. So the polynomial interpolation uniformly converges for $|b-a| < 2b < 1 \implies b < 1/2$.
If we consider a function which is analytically extensible everywhere, for instance $f = sin(x)$, we have that $R \to \infty$, so $||E_n[f]|| \to 0$ everywhere.

Chebyshev-Lobatto points

Theorem (Bernstein's Theorem)

For every function $f \in C^1([a,b])$, $-\infty < a < b < +\infty$, if p_n is the interpolant of f at $n+1$ Chebyshev nodes, then $\|E_n[f]\|_\infty \to 0$ as $n \to +\infty$.

The set of $n+1$ Chebyshev-Lobatto points in $[-1,1]$, useful for polynomial interpolation of degree n, as seen in the figure for $n=6$, is obtained by projecting the equispaced points on the semicircle $\gamma = \{(\cos(\theta), \sin(\theta)) : \theta \in [0,\pi]\}$ onto the x-axis.

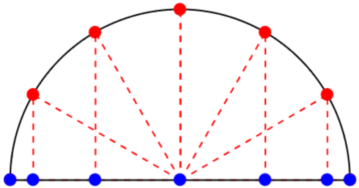


Figure: Geometric interpretation of the Chebyshev-Lobatto points used for determining a polynomial interpolant of degree $n = 6$

Runge function

n	E_n^e	E_n^c
1	9.615e-01	9.615e-01
2	6.462e-01	6.462e-01
3	7.070e-01	8.289e-01
4	4.384e-01	4.600e-01
5	4.327e-01	6.386e-01
6	6.169e-01	3.112e-01
7	2.474e-01	4.596e-01
8	1.045e+00	2.047e-01
9	3.003e-01	3.191e-01
10	1.916e+00	1.322e-01
20	5.982e-01	1.774e-02
30	2.388e+03	2.426e-03
40	1.047e+05	3.399e-04
50	4.822e+06	4.622e-05
60	1.796e+11	6.381e-06
70	4.171e+07	8.755e-07
80	6.774e+05	1.196e-07
90	1.498e+05	1.648e-08
100	5.177e+03	2.256e-09

Table: Degree n and error $|E_n[f]|_\infty$ with the interpolant for $n+1$ equally spaced nodes and Chebyshev nodes (denoted respectively by E_n^E and E_n^C).

Runge function

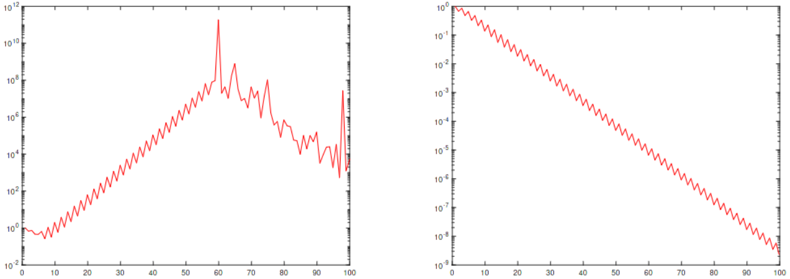


Figure: Degree n and error $E_n[f] = \max_{x\in[-5,5]} |f(x) - p_n(x)|$ with p_n, interpolating f respectively at $n+1$ equally spaced nodes (left figure) and Chebyshev nodes (right figure), in a semilogarithmic scale.

Stability of Polynomial Interpolation

Let $f \in C([a, b])$, with $[a, b]$ a closed and bounded interval, and consider the polynomial $p_n \in \mathbb{P}_n$ that interpolates the pairs $(x_k, f(x_k))$ (for $k = 0, \ldots, n$, with x_k distinct). For simplicity, let $f_k = f(x_k)$. As is known, denoting by L_k the k-th Lagrange polynomial, we have

$$p_n(x) = \sum_{k=0}^{n} f_k L_k(x)$$

with

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

Suppose the values of f_k are perturbed (for example, due to rounding errors) and are replaced with $\tilde{f}_k$.

Stability of Polynomial Interpolation

Thus, the interpolating polynomial is $\tilde{p}_n(x) = \sum_{k=0}^{n} \tilde{f}_k L_k(x)$. Since $p_n(x) = \sum_{k=0}^{n} f_k L_k(x)$, we have

$$p_n(x) - \tilde{p}_n(x) = \sum_{k=0}^{n} (f_k - \tilde{f}_k) L_k(x)$$

from which

$$|p_n(x) - \tilde{p}_n(x)| \leq \sum_{k=0}^{n} |f_k - \tilde{f}_k| |L_k(x)|$$

and

$$\max_{x \in [a,b]} |p_n(x) - \tilde{p}_n(x)| \leq \left(\max_{x \in [a,b]} \sum_{k=0}^{n} |L_k(x)| \right) \max_{k=0,\dots,n} |f_k - \tilde{f}_k|.$$

Stability of Polynomial Interpolation

Thus, let

$$\Lambda_n = \max_{x \in [a,b]} \sum_{k=0}^{n} |L_k(x)|$$

From this, we have

$$|p_n - \tilde{p}_n|_\infty \leq \left(\max_{k=0,\ldots,n} |f_k - \tilde{f}_k| \right) \cdot \Lambda_n$$

We obtain

$$\|p_n - \tilde{p}_n\|_\infty \leq \left(\max_{k=0,\ldots,n} |f_k - \tilde{f}_k| \right) \cdot \Lambda_n.$$

We observe that the number Λ_n depends exclusively on the Lagrange polynomials and thus only on the interpolation points.

Stability of Polynomial Interpolation

Theorem.

Let $f \in C([a,b])$ and let p_n be its interpolating polynomial at the points $x_0, \ldots, x_n$. Then we have

$$\|f - p_n\|_\infty \leq (1 + \Lambda_n) d_n(f)$$

where

$$d_n(f) = \inf_{q_n \in \mathcal{P}_n} \|f - q_n\|_\infty$$

is the error made by the polynomial of the best uniform approximation.

$\Rightarrow$ The smaller Λ_n, the potentially smaller the error made by the interpolating polynomial.

Lebesgue Constants

The value Λ_n is known as the Lebesgue constant (1910) for the set of points $x_0, \dots, x_n$. It is immediately seen that it is an index of stability for Lagrange interpolation: the smaller it is, the more stable the approximation.

Let's see what the sums of Lebesgue constants look like for certain sets of $n+1$ points in the interval $[-1, 1]$:

- Equally spaced points: It is asymptotically demonstrated (Turetskii, 1940) that

$$\Lambda_n \sim \frac{2^{n+1}}{n \log n}$$

- Chebyshev points: Corresponding to $\cos\left(\frac{(2k-1)\pi}{2n+2}\right)$ for $k = 1, \dots, n+1$, it is asymptotically demonstrated that

$$\Lambda_n = \frac{2}{\pi}\left(\log(n+1) + \gamma + \log\left(\frac{8}{\pi}\right)\right) - \frac{1}{2(n+1)^2}$$

where $\gamma \approx 0.5777$ is the Euler-Mascheroni constant.

Lebesgue Constants

n	Λ^n_{eq}	Λ^n_{ch}
5	2.208e+00	1.989e+00
10	1.785e+01	2.429e+00
15	2.832e+02	2.687e+00
20	5.890e+03	2.870e+00
25	1.379e+05	3.012e+00
30	3.448e+06	3.128e+00
35	9.001e+07	3.226e+00
40	2.422e+09	3.311e+00
45	6.665e+10	3.386e+00
50	1.868e+12	3.453e+00
55	5.518e+13	3.514e+00
60	8.354e+16	3.569e+00
65	7.507e+15	3.620e+00
70	3.292e+16	3.667e+00
75	1.339e+17	3.711e+00
80	1.286e+17	3.752e+00
85	3.831e+18	3.791e+00
90	5.980e+18	3.827e+00
95	8.922e+16	3.862e+00
100	3.790e+17	3.894e+00

Table: Value n and Lebesgue constants in n equispaced points belonging to $[-1, 1]$ and in Chebychev points $\cos\left(\frac{(2k-1)\pi}{2n}\right)$, $k = 1, \ldots, n$.