Lecture Notes for Math 5630/6630 **Fall 2024**

Note 6: Polynomial Interpolation

Tags: math.na *Date: 09/17/2024*

Disclaimer: *This lecture note is for math 5630/6630 class only.*

The interpolation (1D) solves problems of the following type:

Given a set of predefined functions $\mathcal{K}$, find an element $f : \mathbb{I} \mapsto \mathbb{R}$ in $\mathcal{K}$ such that $y_j = f(x_j)$ for all $j = 0, \ldots, n$.

Here $\mathbb{I}$ denotes a finite or infinite interval such that $x_1, \ldots x_n \in \mathbb{I}$. One of the important applications for interpolation is Computer-Aided Design (CAD) which is used extensively in the manufacturing industry. Generally speaking, the interpolation provides a closed form of the function to determine the value of y where the parameter x is not accessible.

1 Polynomial Interpolation

The polynomial interpolation considers the set $\mathcal{K} = \Pi_m$, where the set Π_m represents the polynomials of with degree $\leq m$. We will seek for a polynomial $f(x)$ with the constraints that

$$\begin{cases} f \in \mathcal{K} = \Pi_m, \\ f(x_k) = y_k & \text{for } k = 0, 1, \ldots, n. \end{cases}$$

The points x_k are called **interpolation nodes**, if $m > n$ (resp. $m < n$), the problem is underdetermined (resp. overdetermined). For the case that $m = n$, we have

Theorem 1.1. *There exists a unique polynomial function $f \in \Pi_n$ such that $f(x_j) = y_j$ for $j = 0, \ldots, n$.*

Existence: In order to construct the polynomial f, it is straightforward to consider the general form of polynomial $f(x) = \sum_{j=0}^{n} a_j x^j$, then we can formulate a linear system for the coefficients a_j, $j = 0, \dots, n$, which is

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

The matrix

$$V = \begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix}$$

is called **Vandermonde matrix**. To determine the coefficients a_j, one needs the matrix V to be invertible. Its determinant can be computed (as an exercise) as

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

When x_j are distinct, the determinant is nonzero.

Uniqueness: Suppose there are two distinct polynomials $f, g \in \Pi_n$ satisfying the condition that $f(x_j) = g(x_j) = y_j$, then $f - g$ has $(n+1)$ roots x_j, $j = 0, \dots, n$. If $f \neq g$, it is clear that $f - g \in \Pi_n$ has at most n roots. Contradiction.

In the above derivation, the interpolation polynomial can be uniquely determined by solving the linear system

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

However, it is generally easier to compute the polynomial f with the **Lagrange polynomial interpolation** (which is somewhat equivalent to computing the inverse of V).

Example 1.2. *For instance, to find a quadratic polynomial* $p(x) = a + bx + cx^2$ *interpolating points* $(0,1)$, $(1,2)$, $(2,4)$, *we only need to solve the linear system*

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}$$

The solution is $a = 1$, $b = \frac{1}{2}$, $c = \frac{1}{2}$.

1.1 Lagrange Polynomial

Definition 1.3. *For the given distinct* x_j, $j = 0, 1, \ldots, n$, *the* $(n+1)$ *Lagrange polynomials* $L_0, L_1, \ldots, L_n \in \Pi_n$ *are defined by*

$$L_j(x) = \prod_{s=0, s\neq j}^{n} \frac{x - x_s}{x_j - x_s}, \quad j = 0, 1, \ldots, n.$$

It is clear that these polynomials satisfy the conditions that

$$L_j(x_k) = \delta_{jk} := \begin{cases} 1 & \text{for } k = j, \\ 0 & \text{for } k \neq j. \end{cases}$$

Therefore these polynomials are linearly independent, which form a basis of the $(n+1)$-dimensional space Π_n.

The unique interpolating polynomial f satisfying $f(x_j) = y_j$, $j = 0, 1, \ldots, n$ can be represented by

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

It is straightforward to check the interpolation conditions are satisfied. By constructing the Lagrange polynomial, there is another way to derive the existence part of Theorem **??**.

We introduce a preliminary procedure to compute the interpolating polynomial f evaluated at a point x. Let constants k_j and $q(x)$ be defined as

$$k_j = \prod_{s=0, s\neq j} \frac{1}{x_j - x_s}, \quad q(x) = \prod_{j=0}^{n} (x - x_j),$$

then

$$f(x) = \sum_{j=0}^{n} y_j L_j(x) = q(x) \sum_{j=0}^{n} k_j y_j \frac{1}{x - x_j}.$$

One can first compute k_j with $\mathcal{O}(n^2)$ flops, then $f(x)$ can be computed with $\mathcal{O}(n)$ flops. The advantage of the above scheme is the constants k_j are independent of y_j, therefore evaluating another instance of the interpolating polynomial will not need to re-compute them. The disadvantage is that if we add a new node, the constants k_j will be updated with an additional cost of $\mathcal{O}(n)$ flops.

This formulation is also called *barycentric*. A more aesthetic form of the interpolation is

$$f(x) = \frac{\sum_{j=0}^{n} k_j y_j \frac{1}{x-x_j}}{\sum_{j=0}^{n} k_j \frac{1}{x-x_j}}.$$

Because of the identity (leave as an exercise)

$$q(x) \sum_{j=0}^{n} k_j \frac{1}{x - x_j} \equiv 1.$$

Later we will see how Newton's form can overcome this issue.

1.2 Interpolation Error

When the data pairs (x_j, y_j), $j = 0, 1, \ldots, n$ are generated by a sufficiently smooth function $h(x)$, it is possible to quantify the error between the interpolating polynomial $f(x)$ and $h(x)$.

Theorem 1.4. *Let $h : [a, b] \mapsto \mathbb{R}$ be a $(n + 1)$-times differentiable function. If $f(x) \in \Pi_n$ is the interpolating polynomial that*

$$f(x_j) = h(x_j),$$

for $j = 0, 1, \ldots, n$. Then for each $\overline{x} \in [a, b]$, the error can be represented by

$$h(\overline{x}) - f(\overline{x}) = \frac{\omega(\overline{x})}{(n + 1)!} h^{(n+1)}(\xi),$$

where $\xi = \xi(\overline{x}) \in [a, b]$ and $\omega(x) = \prod_{j=0}^{n}(x - x_j)$.

Select any $\overline{x} \in [a,b]$ such that $\omega(\overline{x}) \neq 0$, then let

$$\psi(x) = h(x) - f(x) - k\omega(x)$$

the constant k is chosen such that $\psi(\overline{x}) = 0$. Then $\psi(x) = 0$ at $(n+2)$ points

$$x_0, x_1, \dots, x_n, \overline{x} \in [a,b]$$

By Rolle's Theorem, $\psi^{(n+1)}$ has at least one zero ξ in $[a,b]$. Therefore

$$\psi^{(n+1)}(\xi) = h^{(n+1)}(\xi) - 0 - k(n+1)! = 0.$$

We can apply the theorem directly to derive the following result.

Corollary 1.5. *If $h(x) \in C^\infty([a,b])$ satisfies that $\max_{x\in[a,b]}|h^{(n)}(x)| \leq M < \infty$ for all $n \geq 0$, then the interpolating polynomial approximates h uniformly as the number of nodes $n \to \infty$.*

Because the error is bounded by $\frac{|b-a|^{n+1}}{(n+1)!}M \to 0$ as $n \to \infty$. Note that the distribution of nodes is not important here.

Example 1.6. *For instance, let $h(x) = \sin(x)$, its derivatives are all bounded, then the interpolating polynomial approximates $h(x)$ uniformly as the number of nodes grows to infinity.*

It is interesting to think about the converse: under which condition, the interpolation error is not vanishing as the number of nodes tends to infinity? From Theorem **??**, the error $\frac{\omega(\overline{x})}{(n+1)!}h^{(n+1)}(\xi)$ depends on the sizes of three terms.

i. The bound of the $(n+1)$-th derivative, $\max_{x\in[a,b]}|h^{(n+1)}(x)|$. This can be growing fast. For instance, $h(x) = 1/\sqrt{x}$ on $[\frac{1}{2}, \frac{3}{2}]$,

$$h^{(n+1)}(x) = \frac{(-1)^{n+1}}{2^{n+1}}(2n+1)!!\,x^{-(2n+3)/2}.$$

ii. The function $\omega(x) = \prod_{j=0}^{n}(x - x_j)$, such product could be large if x and the nodes x_j are not so close.

iii. The term $\frac{1}{(n+1)!}$, which decays fast.

Example 1.7. *We can see that for the function $h(x) = 1/\sqrt{x}$ on $[\frac{1}{2}, \frac{3}{2}]$, it is not trivial to show the interpolating polynomial could converge to $h(x)$ anymore (it is still true for certain choices of x_j) as the number of nodes increases.*

Next, we try to provide a better estimate of ω for the special choice: equally spaced nodes. Let the nodes $x_j = a + j\Delta$, where $\Delta = \frac{b-a}{n}$. It is not difficult (prove it) to see $\omega(x)$ will be the worst if x is located on the end sub-intervals, $[x_0, x_1]$ and $[x_{n-1}, x_n]$. Without loss of generality, we assume x is located on $[x_0, x_1]$, then

$$|x - x_j| \le j\Delta$$

for $j = 2, \dots, n$, which implies

$$|\omega(x)| \le \prod_{j=0}^{n} |x - x_j| \le n!\,\Delta^{n-1} \sup_{x\in[x_0,x_1]} |(x - x_0)(x - x_1)| = \frac{n!}{4} \frac{(b-a)^{n+1}}{n^{n+1}}.$$

Thus the interpolation error is bounded by

$$\|h - f\|_\infty \le \frac{\sup_{x\in[a,b]} |h^{(n+1)}(x)|}{4(n+1)} \frac{(b-a)^{n+1}}{n^{n+1}}.$$

Such an estimate is useful in deriving uniform convergence (as $n \to \infty$).

Example 1.8. *Consider $h(x) = 1/x$ on $[\frac{1}{2}, \frac{3}{2}]$. Then $h^{(n+1)}(x) = \frac{(n+1)!(-1)^{n+1}}{x^{n+2}}$, hence*

$$\frac{|h^{(n+1)}(x)|}{4(n+1)} \frac{(b-a)^{n+1}}{n^{n+1}} \le \frac{1}{4n^{n+1}} \max_{x\in[\frac{1}{2},\frac{3}{2}]} \left| \frac{1}{x^{n+2}} \right| = \frac{1}{2} \left(\frac{2}{n} \right)^{n+1},$$

therefore, the interpolation error converges to zero exponentially. It is important to notice that the above method only works for intervals away from the origin.

2 Runge's Phenomenon

From the previous discussion, we can see there is a possibility that $\max_{x\in[a,b]} |h^{n+1}(x)|\omega(x)$ grows faster than $(n+1)!$, which would lead to divergence. Hence increasing the number of interpolation nodes (at least for equally spaced nodes) is not guaranteed for better approximation. The most famous example is the one found by *Carl Runge*.

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

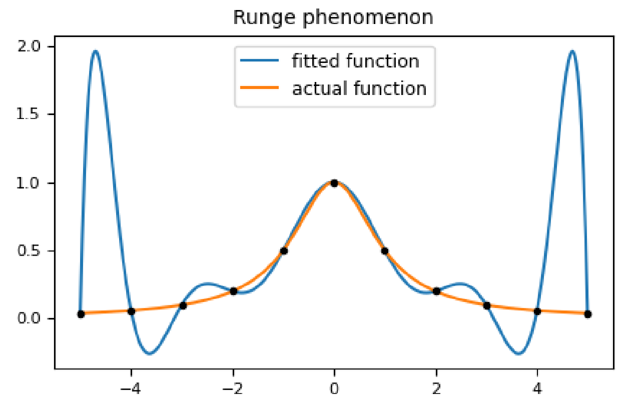

Figure 6.1: Runge's Phenomenon

It can be shown that the interpolation will diverge at around 3.6 as $n \to \infty$ and the maximum error $\max_{x\in[-5,5]}|f_n(x) - h(x)|$ grows exponentially, where f_n is the interpolating polynomial with $(n + 1)$ equally spaced nodes.