Polynomial Interpolation

Min-Hsiung Lin

Department of Mathematics
National Cheng Kung University
Thank Prof. Moody T. Chu for this lecture note.

September 7, 2016

- Two distinct points can uniquely determine a straight line. What can three points in a plane that are not collinear determine?
  - Given $\{(x_i, f_i)\}_{i=0}^{2}$, determine a quadratic polynomial

    $$p(x) = a_0 + a_1 x + a_2 x^2$$

    such that

    $$p(x_i) = f_i, \; i = 0, 1, 2.$$

  - The coefficients can be determined, in principle, by solving the linear equation

    $$\begin{bmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \end{bmatrix}.$$

  - Concerns in numerical calculation:
    - Is the system solvable?
    - How expensive?
    - How about conditioning of the linear system?

- The general interpolation problem:
    - Given points $\{(x_i, f_i)\}_{i=0}^n$, where x_i are distinct, determine a polynomial $p(x)$ satisfying

$$\begin{aligned} \deg(p) &\leq n, \\ p(x_i) &= f_i, \ i = 0, 1, \ldots, n. \end{aligned}$$

    - If $p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$, then the interpolation problem is equivalent to solving the Vandermonde linear system

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{bmatrix}.$$

- Can we construct n polynomials $\ell_j(x)$ for $j = 0, 1, \ldots, n$, each of which has degree n and does the following interpolation?

$$\ell_j(x_i) = \begin{cases} 0, & \text{if } i \neq j, \\ 1, & \text{if } i = j. \end{cases}$$

Solution.

$$\ell_j(x) := \prod_{i=0, i\neq j}^{n} \frac{x - x_i}{x_j - x_i}, \; j = 0, 1, \ldots, n$$

$\square$

Theorem

Given $n+1$ distinct points $x_0, \cdots, x_n$ and $n+1$ corresponding values $f_0, \cdots, f_n$, there exists a unique polynomial $p(x)$ of degree $\leq n$ such that $p(x_i) = f_i$, for $i = 0, \cdots, n$.

- For the general interpolation problem, the polynomial is given by

$$p(x) = \sum_{j=0}^{n} f_j \ell_j(x). \tag{1}$$

- $p(x_i) = \sum_{j=0}^{n} f_j \ell_j(x_i) = f_i$. Hence, **existence** question raised in the interpolation theory is satisfied.
- The **uniqueness** of the interpolating polynomial follows from the (weak form of) Fundamental Theorem of Algebra, i.e., a polynomial of degree n vanishing at $n+1$ distinct points is identically zero. (Prove it!!) $\square$

Remark

1. *The polynomial $p(x)$ defined by (1) is called the Lagrange interpolation polynomial.*
2. *Note that we can only assure that deg$(p) \leq n$, but not necessarily always deg$(p) = n$. (Why?)*

- The Lagrange polynomials provide useful insights into the approximation theory in general, but is difficult to apply in practice.
  - If more data points are added to the interpolation problem, all the function $\ell_j(x)$ have to be recalculated.
  - We shall now derive the interpolating polynomials in a manner that use the previous calculations to greater advantage.

Example

Consider the case $n = 1$, i.e., two points $(2, 2.5)$ and $(3, 4)$ are to be interpolated.

- The two Lagrange polynomials are easy to construct.

$$\begin{aligned} \ell_0(x) &= \frac{x-3}{2-3} \\ \ell_1(x) &= \frac{x-2}{3-2}. \end{aligned}$$

- Their geometry is sketched below.

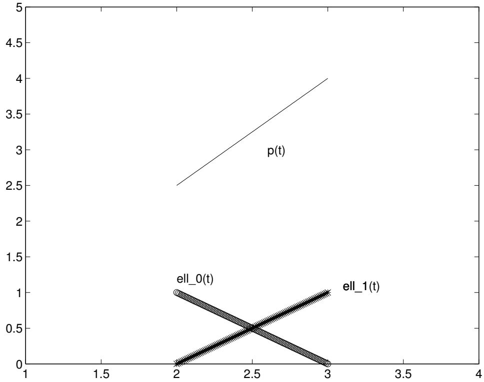

Figure: First degree Lagrange polynomials.

- The straight line that interpolates the two given nodes can be obtained by linearly combined $\ell_0(t)$ and $\ell_1(t)$ together. (Why?)

- Given the following four data points

x_i	0	1	3	5
f_i	1	2	6	7

find a polynomial in Lagrange form to interpolate these data.

  - The interpolating polynomial in the Lagrange form is $p_3(x) = \ell_0(x) + 2\ell_1(x) + 6\ell_2(x) + 7\ell_3(x)$ with

$$\begin{cases} \ell_0(x) = \frac{(x-1)(x-3)(x-5)}{(0-1)(0-3)(0-5)} = -\frac{(x-1)(x-3)(x-5)}{15} \\ \ell_1(x) = \frac{(x-0)(x-3)(x-5)}{(1-0)(1-3)(1-5)} = \frac{x(x-3)(x-5)}{8} \\ \ell_2(x) = \frac{(x-0)(x-1)(x-5)}{(3-0)(3-1)(3-5)} = -\frac{x(x-1)(x-5)}{12} \\ \ell_3(x) = \frac{(x-0)(x-1)(x-3)}{(5-0)(5-1)(5-3)} = \frac{x(x-1)(x-3)}{40} \end{cases}$$

- Given a polynomial in the *natural form*

$$p(t) = a_n t^n + a_{n-1} t^{n-1} + \ldots + a_1 t + a_0,$$

  the evaluation of $p(t)$ can be done stably by an algorithm called *synthetic division*:

```
p = a[n]
for i = n-1:-1:0
    p = p*t + a_[i]
end
```

  - Synthetic division requires only n additions and n multiplications. It is quite efficient.
  - Note that the Lagrange polynomials are not in the natural form and hence is difficult to evaluate.

- We say a polynomial $p(t)$ is the *Netwon form* (Newton interpolation polynomial) if

$$p(t) = c_0 + c_1(t - x_0) + c_2(t - x_0)(t - x_1) + \ldots + c_n(t - x_0)(t - x_1)\ldots(t - x_{n-1}).$$

  - Evaluation of a Newton form is easy :

```
p = c[n]
for i = n-1:-1:0
    p = p*(t-x[i]) + c[i]
end
```

  - It remains to determine the coefficients $c_0, \ldots, c_n$ so that $p(t)$ interpolates the data $\{(x_i, f_i)\}$ for $i = 0, 1, \ldots, n$.

- Newton interpolation polynomial is mathematically equivalent to the Lagrange interpolation polynomial (why?), but is much more efficient.
    - One of the most important features of Newton's formula is that one can gradually increase the support data without recomputing what is already computed.
- The coefficients of the Newton form of an interpolation can be determined through the system

$$
\begin{aligned}
f_0 &= c_0 \\
f_1 &= c_0 + c_1(x_1 - x_0) \\
&\vdots \\
f_n &= c_0 + c_1(x_n - x_0) + \ldots + c_n(x_n - x_0)\ldots(x_n - x_{n-1}).
\end{aligned}
$$

- This is a lower triangular system whose diagonal elements are nonzero, if all given nodes are distinct.

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & (x_1 - x_0) & \dots & 0 \\ 1 & (x_2 - x_0) & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 1 & (x_n - x_0) & \dots & (x_n - x_0)\dots(x_n - x_{n-1}) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}.$$

- If new points are added to be interpolated, the coefficients already determined will *not* be affected. We just need to add a new row to determine c_{n+1}.
- There is a yet better way, called the Newton divided differences, to determine the coefficients.

- Divided Difference:
  - Let $P_{i_0 i_1 \ldots i_n}(x)$ represent the n-th degree polynomial that satisfies

$$P_{i_0 i_1 \ldots i_n}(x_{i_j}) = f_{i_j} \tag{2}$$

    for all $j = 0, \ldots, n$.
  - The recursion formula holds:

$$P_{i_0 i_1 \ldots i_n}(x) = \frac{(x - x_{i_0})P_{i_1 \ldots i_n}(x) - (x - x_{i_n})P_{i_0 \ldots i_{n-1}}(x)}{x_{i_n} - x_{i_0}} \tag{3}$$

    - The right-hand side of (3), denoted by $R(t)$, is a polynomial of degree $\leq n$.
    - $R(x_{i_j}) = f_{i_j}$ for all $j = 0, \ldots, n$. That is, $R(x)$ interpolates the same set of data as does the polynomial $P_{i_0 i_1 \ldots i_n}(x)$.
    - By uniqueness, $R(x) = P_{i_0 i_1 \ldots i_n}(x)$ (i.e. R(x) describes the n-th Lagrange polynomial that interpolates f at the $n + 1$ points $x_0, x_1, \ldots, x_n$).

- The difference $P_{i_0 i_1 \dots i_n}(x) - P_{i_0 i_1 \dots i_{n-1}}(x)$ is a n-th degree polynomial that vanishes at x_{i_j} for $j = 0, \dots, n-1$. Thus we may write

$$\begin{aligned} P_{i_0 i_1 \dots i_n}(x) &= P_{i_0 i_1 \dots i_{n-1}}(x) \\ &+ f_{i_0 \dots i_n}(x - x_{i_0})(x - x_{i_1}) \dots (x - x_{i_{n-1}}). \end{aligned} \tag{4}$$

- Following from formula (4), the leading coefficients $f_{i_0 \dots i_n}$ can be determined by considering the coefficient of x^n in formula (3), i.e.

$$\begin{aligned} &[f_{i_0 \dots i_n}(x - x_{i_0}) \dots (x - x_{i_{n-1}}) + P_{i_0 i_1 \dots i_{n-1}}(x)] \\ &= P_{i_0 i_1 \dots i_n}(x) \\ &= \tfrac{(x - x_{i_0}) P_{i_1 \dots i_n}(x) - (x - x_{i_n}) P_{i_0 \dots i_{n-1}}(x)}{x_{i_n} - x_{i_0}} \end{aligned} \tag{5}$$

  That is,

$$f_{i_0 \dots i_n} = \frac{f_{i_1 \dots i_n} - f_{i_0 \dots i_{n-1}}}{x_{i_n} - x_{i_0}} (\text{why?}) \tag{6}$$

  where $f_{i_1 \dots i_n}$ and $f_{i_0 \dots i_{n-1}}$ are the leading coefficients of the polynomials $P_{i_1 \dots i_n}(x)$ and $P_{i_0 \dots i_{n-1}}(x)$, respectively.

- Let $x_0, \ldots, x_n$ be support arguments (but not necessarily in any order) over the interval $[a, b]$. We define the ***Newton divided difference*** as follows:

$$f[x_0] := f(x_0) \tag{7}$$

$$f[x_0, x_1] := \frac{f[x_1] - f[x_0]}{x_1 - x_0} \tag{8}$$

$$f[x_0, \ldots, x_n] := \frac{f[x_1, \ldots, x_n] - f[x_0, \ldots, x_{n-1}]}{x_n - x_0} \tag{9}$$

- The n-th degree polynomial that interpolates the set of support data $\{(x_i, f_i) | i = 0, \ldots, n\}$ is determined recursively form the formula (4), i.e.,

$$\begin{aligned} P_{x_0 \ldots x_n}(x) &= f[x_0] + f[x_0, x_1](x - x_0) \qquad (10) \\ &+ \ldots + f[x_0, \ldots, x_n](x - x_0)(x - x_1) \ldots (x - x_{n-1}). \end{aligned}$$

  This equation is known as ***Newton divided difference formula***.

- Let d_{ij} denote the (i, j)-entry in the following table where the indexing begins with $d_{00} = f[x_0]$.

$$
\begin{array}{llll}
f[x_0] = f_0 & & & \\
f[x_1] = f_1 & f[x_0, x_1] & & \\
f[x_2] = f_2 & f[x_1, x_2] & f[x_0, x_1, x_2] & \\
f[x_3] = f_3 & f[x_2, x_3] & f[x_1, x_2, x_3] & f[x_0, x_1, x_2, x_3] \\
\vdots & \vdots & \vdots & \vdots
\end{array}
$$

  - The array can be built up columnwise.

$$
d_{ij} = \frac{d_{i,j-1} - d_{i-1,j-1}}{x_i - x_{i-j}}.
$$

  - The **diagonal** elements are the coefficients of the Newton interpolant.

- It is not necessary to store the entire 2-dimensional table. Suppose the values of $f_0, \ldots, f_n$ have been stored in the array $c[0], \ldots, c[n]$ (For convenience of indexing, only $n+1$ support data are marked in this example.) Then

```
for j=1:1:n
    for i=n:-1:j
        c[i]=(c[i]-c[i-1])/(x[i]-x[i-j]);
    end
end
```

  - Entries of the resulting array c are the desirable coefficients for the Newton interpolant. This can be explained by writing down above iterative algorithm step by step.
  - The columns are generated from the bottom up to avoid premature overwriting of values of c.
  - The operation counts is n^2+n(i.e. $2\sum_{j=1}^{n}(n-j+1)$) additions and $\frac{n^2+n}{2}$(i.e. $\sum_{j=1}^{n}(n-j+1)$) divisions.

Before analyzing the error in interpolation, let us discuss two important results first.

Theorem

Let $f \in \mathbb{C}[a,b]$ be n times differentiable in (a,b). If f vanishes at $n+1$ distinct points $x_0, x_1, \ldots, x_n$ in $[a,b]$, then there exist $\xi \in (a,b)$ such that $f^n(\xi) = 0$.

Theorem

Suppose $f \in \mathbb{C}^n[a,b]$ and $x_0, \cdots, x_n$ are distinct numbers in [a,b]. Then there exists $\xi \in (a,b)$ such that

$$f[x_0, \cdots, x_n] = \frac{f^n(\xi)}{n!}.$$

Proof. Let $p(x)$ be the nth divided difference formula such that $p(x_i) = f(x_i)$, for $i = 0, \cdots, n$. Define

$$g(x) = f(x) - p(x).$$

By the generalized Rolle's Theorem, $\exists \xi \in (a, b)$ such that

$$0 = g^n(\xi) = f^n(\xi) - p^n(\xi).$$

That is,

$$p^n(\xi) = n! f[x_0, x_1, \cdots, x_n].$$

Hence,

$$f[x_0, \cdots, x_n] = \frac{f^n(\xi)}{n!}.$$

$\square$

Theorem

Suppose $x_0, x_1, \ldots, x_n$ are distinct numbers in the interval $[a, b]$ and $f \in \mathbb{C}^{n+1}[a, b]$. Then, for each x in [a,b], a number $\xi(x)$ in (a, b) exists with

$$f(x) = p(x) + \frac{\prod_{i=0}^{n}(x - x_i)}{(n+1)!} f^{(n+1)}(\xi(x)) \tag{11}$$

where $p(x)$ is the interpolating polynomial of $f(x)$.

Proof.

- If $x = x_i$ for any $i = 0, 1, \ldots, n$, then $f(x_i) = p(x_i)$, and (11) is satisfied for any $\xi(x_i) \in (a, b)$

- If $x \neq x_i$ for any $i = 0, 1, \ldots, n$. Define

$$F(t) = f(t) - p(t) - (f(x) - p(x)) \frac{\displaystyle\prod_{i=0}^{n}(t - x_i)}{\displaystyle\prod_{i=0}^{n}(x - x_i)}.$$

- Observe $F(x_i) = 0$ for $i = 0, 1, \ldots n$ and $F(x) = 0$, i.e., $F(t)$ has $n + 2$ zeros.
- By Rolle's theorem, there exists ξ betweein $x, x_0, \ldots, x_n$ such that $F^{(n+1)}(\xi) = 0$.

- Note that

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

- Upon solving for $f(x)$, we have

$$f(x) = p(x) + \frac{\displaystyle\prod_{i=0}^{n}(x - x_i)}{(n+1)!} f^{(n+1)}(\xi(x)) \tag{12}$$

for some $\xi(x)$ between $x, x_0, \ldots, x_n$.

- Reasons we want to use polynomials for interpolation is because
  - Polynomials are easy to generate (say, by the Newton's formula),
  - Polynomials are easy to manipulate (say, for differentiation or integration),
  - Polynomials *fill up* the function space:
    - Let $f(x)$ be a piecewise continuous function over the interval $[a, b]$. Then for any $\epsilon > 0$, there exist an integer n and numbers $a_0, \ldots, a_n$ such that

$$\int_a^b \{f(x) - \sum_{i=0}^{n} a_i x^i\}^2 dx < \epsilon.$$

    - (Weierstrass Approximation Theorem) Let $f(x)$ be a continuous function on $[a, b]$. For any $\epsilon > 0$, there exist an integer n and a polynomial $p_n(x)$ of degree n such that $\max_{x \in [a,b]} |f(x) - p_n(x)| < \epsilon$.
      In fact, if $[a, b] = [0, 1]$, then the Bernstein polynomial

$$B_n(x) := \sum_{k=0}^{n} \binom{n}{k} x^k (1-x)^{n-k} f(\frac{k}{n}) \tag{13}$$

      converges to $f(x)$ as $n \to \infty$.

- However, polynomial interpolation *cannot* do all the magic. There are severe limitations:
  - Weierstrass's theoretical result, while valid, may require very high degree polynomials.
  - Polynomials can do a better job in interpolation than extrapolation. That is, a polynomial outside the range of its interpolation may not represent the function well.
  - Even within the range of interpolation, like the Runge's example, equally spaced interpolation can diverge.

- Thus far, the interpolation has been required only to interpolate the functional values. Sometimes it is desirable that the derivatives are also interpolated.
- Given $\{x_i\}, i = 0, \dots n$ and values $a_i^{(0)}, \dots, a_i^{(r_i)}$ where r_i are nonnegative integers. We want to construct a polynomial $p(x)$ such that

$$P^{(j)}(x_i) = a_i^{(j)} \tag{14}$$

for $i = 0, \dots, n$ and $j = 0, \dots, r_i$.

  - Such a polynomial is called an **osculatory (kissing) interpolating polynomial** of a function f if $a_i^{(j)} = f^{(j)}(x_i)$ for all i and j.

- Some examples of osculatory interpolation:
  - Suppose $r_i = 0$ for all i. Then this is simply the ordinary Lagrange or Newton interpolation.
  - Suppose $n = 0, x_0 = a, r_0 = k$, then the osculatory polynomial becomes

$$p(x) = \sum_{j=0}^{k} f^{(j)}(a)\frac{(x-a)^j}{j!}$$

    which is the Taylor's polynomial of f at $x = a$.
- One of the most interesting osculatory interpolations is when $r_i = 1$ for all $i = 0, \ldots, n$. That is, the values of $f(x_i)$ and $f'(x_i)$ are to be interpolated.
  - The resulting $(2n+1)$-degree polynomial is called the **Hermite interpolating polynomial**.
  - Very useful for deriving numerical integration scheme of high precision.

- Recall the n-degree polynomial

$$\ell_i(x) = \prod_{\substack{j=1 \\ j\neq i}}^{k} \frac{x - x_j}{x_i - x_j}$$

has the property $\ell_i(x_j) = \delta_{ij}$.

- Define

$$\begin{aligned} h_i(x) &= [1 - 2(x - x_i)\ell_i'(x_i)]\ell_i^2(x) && (15) \\ g_i(x) &= (x - x_i)\ell_i^2(x). && (16) \end{aligned}$$

  - Note that both $h_i(x)$ and $g_i(x)$ are of degree $2n + 1$.
  - The following properties can be checked out:

$$\begin{aligned} h_i(x_j) &= \delta_{ij}; \\ g_i(x_j) &= 0; \\ h_i'(x_j) &= [1 - 2(x - x_i)\ell_i'(x_i)]2\ell_i(x)\ell_i'(x) - 2\ell_i'(x_i)\ell_i^2(x)|_{x=x_j} = 0; \\ g_i'(x_j) &= (x - x_i)2\ell_i(x)\ell_i'(x) + \ell_i^2(x)|_{x=x_j} = \delta_{ij}. \end{aligned}$$

- The Hermite polynomial can be written down as

$$p(x) = \sum_{i=0}^{n} f(x_i)h_i(x) + f'(x_i)g_i(x)). \tag{17}$$

- Exercise:
  - Show that formula (17) expresses the unique polynomial of least degree agreeing with f and f' at $x_0, x_2, \cdots, x_n$.
    (Hint: Assume that q(x) is another such polynomial and consider $G = p - q$ and G' at $x_0, x_2, \cdots, x_n$.)
  - If $x_0, x_2, \cdots, x_n \in [a, b]$ and $f \in \mathbb{C}^{2n+2}[a, b)$, then

$$f(x) = p(x) + \frac{(x - x_0)^2 \cdots (x - x_n)^2}{(2n+2)!} f^{(2n+2)}(\xi).$$

    for some ξ with $a < \xi < b$.
    (Hint: Use the method as in the Lagrange error derivation, defining

$$h(t) = f(t) - p(t) - \frac{(t - x_0)^2 \cdots (t - x_n)^2}{(x - x_0)^2 \cdots (x - x_n)^2}[f(x) - p(x)]$$

    and showing that $h'(t)$ has $2n + 2$ distinct zeros in $[a, b]$.)