Polynomial Interpolation

Polynomials $P_n(x) = a_n x^n + \cdots a_1 x + a_0$ are commonly used for interpolation or approximation of functions.

Benefits include efficient methods, simple differentiation, and simple integration.

Also, Weierstrass Approximation Theorem says that for each $\varepsilon > 0$, there is a $P(x)$ such that

$$|f(x) - p(x)| < \varepsilon \quad \text{for all } x \text{ in } [a, b]$$

for $f \in C[a, b]$. In other words, polynomials are good at approximating general functions.
The Lagrange Polynomial

**Theorem**

If $x_0, \ldots, x_n$ distinct and $f$ given at these numbers, a unique polynomial $P(x)$ of degree $\leq n$ exists with

$$f(x_k) = P(x_k), \quad \text{for each } k = 0, 1, \ldots, n$$

The polynomial is

$$P(x) = f(x_0)L_{n,0}(x) + \ldots + f(x_n)L_{n,n}(x) = \sum_{k=0}^{n} f(x_k)L_{n,k}(x)$$

where

$$L_{n,k}(x) = \frac{(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0)(x_k - x_1) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)}$$

$$= \prod_{i \neq k} \frac{x - x_i}{x_k - x_i}$$
Theorem

$x_0, \ldots, x_n$ distinct in $[a, b]$, $f \in C^{n+1}[a, b]$, then for $x \in [a, b]$ there exists $\xi(x)$ in $(a, b)$ with

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!} (x - x_0)(x - x_1) \cdots (x - x_n)$$

where $P(x)$ is the interpolating polynomial.
Definition

Let $f$ be defined at $x_0, \ldots, x_n$, suppose $m_1, \ldots, m_k$ distinct integers with $0 \leq m_i \leq n$. The Lagrange polynomial that agrees with $f(x)$ at $x_{m_1}, x_{m_2}, \ldots, x_{m_k}$ is denoted $P_{m_1,m_2,\ldots,m_k}(x)$.

Theorem

If $f$ defined at $x_0, \ldots, x_k$, and $x_j, x_i$ two distinct numbers among these. Then

$$P(x) = \frac{(x - x_j)P_{0,1,\ldots,j-1,j+1,\ldots,k}(x) - (x - x_i)P_{0,1,\ldots,i-1,i+1,\ldots,k}(x)}{(x_i - x_j)}$$

is the $k$th Lagrange polynomial that interpolates $f$. 
Neville’s Method

Set \( Q_{i,j} = P_{i-j,i-j+1,...,i-1,i} \) and use the recursive formula to obtain the table:

| \( x \) | \( x_0 \) | \( P_0 = Q_{0,0} \) | \( x_1 \) | \( P_1 = Q_{1,0} \) | \( P_{0,1} = Q_{1,1} \) | \( x_2 \) | \( P_2 = Q_{2,0} \) | \( P_{1,2} = Q_{2,1} \) | \( P_{0,1,2} = Q_{2,2} \) | \( x_3 \) | \( P_3 = Q_{3,0} \) | \( P_{2,3} = Q_{3,1} \) | \( P_{1,2,3} = Q_{3,2} \) | \( P_{0,1,2,3} = Q_{3,3} \) | \( x_4 \) | \( P_4 = Q_{4,0} \) | \( P_{3,4} = Q_{4,1} \) | \( P_{2,3,4} = Q_{4,2} \) | \( P_{1,2,3,4} = Q_{4,3} \) | \( P_{0,1,2,3,4} = Q_{4,4} \) |
function Q = neville(x, xi, fi)
% Compute interpolating polynomial using Neville's method.

n = length(xi) - 1;
Q = zeros(n+1, n+1);
Q(:,1) = fi(:);
for i = 1:n
    for j = 1:i
        Q(i+1,j+1) = ((x - xi(i-j+1)) * Q(i+1,j) - ... 
                      (x - xi(i+1)) * Q(i,j)) / ... 
                      (xi(i+1) - xi(i-j+1));
    end
end
Write the $n$th Lagrange polynomial in the form

$$ P_n(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots $$

$$ = a_n(x - x_0)(x - x_1) \cdots (x - x_{n-1}) $$

Introduce the $k$th divided difference

$$ f[x_i, x_{i+1}, \ldots, x_{i+k-1}, x_{i+k}] = $$

$$ \frac{f[x_{i+1}, x_{i+2}, \ldots, x_{i+k}] - f[x_i, x_{i+1}, \ldots, x_{i+k-1}]}{x_{i+k} - x_i} $$

The coefficients are then $a_k = f[x_0, x_1, x_2, \ldots, x_k]$ and

$$ P_n(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, \ldots, x_k](x - x_0) \cdots (x - x_{k-1}) $$
function F = divideddifference(x, f)
% Compute interpolating polynomial using Divided Differences.

n = length(x)−1;
F = zeros(n+1,n+1);
F(:,1) = f(:);
for i = 1:n
    for j = 1:i
        F(i+1,j+1) = (F(i+1,j) − F(i,j)) / (x(i+1) − x(i−j+1));
    end
end
Equal Spacing

Suppose $x_0, \ldots, x_n$ increasing with equal spacing $h = x_{i+1} - x_i$ and $x = x_0 + sh$

The Newton Forward-Difference Formula then gives

$$P_n(x) = f(x_0) + \sum_{k=1}^{n} \binom{s}{k} \Delta^k f(x_0)$$

where

$$\Delta f(x_0) = f(x_1) - f(x_0)$$
$$\Delta^2 f(x_0) = \Delta f(x_1) - \Delta f(x_0) = f(x_2) - 2f(x_1) + f(x_0)$$

\ldots
The Newton Backward-Difference Formula

- Reordering the nodes gives

\[ P_n(x) = f[x_n] + f[x_n, x_{n-1}](x - x_n) + \cdots = f[x_n, \ldots, x_0](x - x_n)(x - x_{n-1}) \cdots (x - x_1) \]

- The Newton Backward-Difference Formula is

\[ P_n(x) = f[x_n] + \sum_{k=1}^{n} (-1)^k \binom{-s}{k} \nabla^k f(x_n) \]

where the \textit{backward difference} \( \nabla p_n \) is defined by

\[ \nabla p_n = p_n - p_{n-1} \]

\[ \nabla^k p_n = \nabla(\nabla^{k-1} p_n) \]
Osculating Polynomials

**Definition**
Let $x_0, \ldots, x_n$ be distinct in $[a, b]$, and $m_i$ nonnegative integers. Suppose $f \in C^m[a, b]$, with $m = \max_{0 \leq i \leq n} m_i$. The osculating polynomial approximating $f$ is the $P(x)$ of least degree such that

$$\frac{d^k P(x_i)}{dx^k} = \frac{d^k f(x_i)}{dx^k}, \quad \text{for } i = 0, \ldots, n \text{ and } k = 0, \ldots, m_i$$

**Special Cases**
- $n = 0$: $m_0$th Taylor polynomial
- $m_i = 0$: $n$th Lagrange polynomial
- $m_i = 1$: Hermite polynomial
Hermite Interpolation

**Theorem**

If $f \in C^1[a, b]$ and $x_0, \ldots, x_n \in [a, b]$ distinct, the Hermite polynomial is

$$H_{2n+1}(x) = \sum_{j=0}^{n} f(x_j) H_{n,j}(x) + \sum_{j=0}^{n} f'(x_j) \hat{H}_{n,j}(x)$$

where

$$H_{n,j}(x) = [1 - 2(x - x_j) L'_{n,j}(x_j)] L^2_{n,j}(x)$$
$$\hat{H}_{n,j}(x) = (x - x_j) L^2_{n,j}(x).$$

Moreover, if $f \in C^{2n+2}[a, b]$, then

$$f(x) = H_{2n+1}(x) + \frac{(x - x_0)^2 \cdots (x - x_n)^2}{(2n + 2)!} f^{(2n+2)}(\xi(x))$$

for some $\xi(x) \in (a, b)$. 
Divided Differences

Suppose \(x_0, \ldots, x_n\) and \(f, f'\) are given at these numbers. Define \(z_0, \ldots, z_{2n+1}\) by

\[ z_{2i} = z_{2i+1} = x_i \]

Construct divided difference table, but use

\[ f'(x_0), f'(x_1), \ldots, f'(x_n) \]

instead of the undefined divided differences

\[ f[z_0, z_1], f[z_2, z_3], \ldots, f[z_{2n}, z_{2n+1}] \]

The Hermite polynomial is

\[ H_{2n+1}(x) = f[z_0] + \sum_{k=1}^{2n+1} f[z_0, \ldots, z_k](x - z_0) \cdots (x - z_{k-1}) \]
Cubic Splines

Definition

Given a function \( f \) on \( [a, b] \) and nodes \( a = x_0 < \cdots < x_n = b \), a cubic spline interpolant \( S \) for \( f \) satisfies:

(a) \( S(x) \) is a cubic polynomial \( S_j(x) \) on \( [x_j, x_{j+1}] \)
(b) \( S_j(x_j) = f(x_j) \) and \( S_j(x_{j+1}) = f(x_{j+1}) \)
(c) \( S_{j+1}(x_{j+1}) = S_j(x_{j+1}) \)
(d) \( S'_{j+1}(x_{j+1}) = S'_j(x_{j+1}) \)
(e) \( S''_{j+1}(x_{j+1}) = S''_j(x_{j+1}) \)
(f) One of the following boundary conditions:
   (i) \( S''(x_0) = S''(x_n) = 0 \) (free or natural boundary)
   (ii) \( S'(x_0) = f'(x_0) \) and \( S'(x_n) = f'(x_n) \) (clamped boundary)
Computing Natural Cubic Splines

Solve for coefficients $a_j, b_j, c_j, d_j$ in

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

by setting $a_j = f(x_j)$, $h_j = x_{j+1} - x_j$, and solving $A\mathbf{x} = \mathbf{b}$:

$$A = \begin{bmatrix}
1 & 0 \\
h_0 & 2(h_0 + h_1) & h_1 \\
& \ddots & \ddots & \ddots \\
& & h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\
& & & 0 & 1
\end{bmatrix}$$

$$\mathbf{b} = (0, 3(a_2 - a_1)/h_1 - 3(a_1 - a_0)/h_0, \ldots, 3(a_n - a_{n-1})/h_{n-1} - 3(a_{n-1} - a_{n-2})/h_{n-2}, 0)^T$$

$$\mathbf{x} = (c_0, \ldots, c_n)^T$$

Finally,

$$b_j = (a_{j+1} - a_j)/h_j - h_j(2c_j + c_{j+1})/3, \quad d_j = (c_{j+1} - c_j)/(3h_j)$$
Computing Clamped Cubic Splines

Solve for coefficients $a_j, b_j, c_j, d_j$ in

$$S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$$

using same procedure as for natural cubic splines, but with

$$A = \begin{bmatrix}
2h_0 & h_0 & & \\
h_0 & 2(h_0 + h_1) & h_1 & \\
& \ddots & \ddots & \ddots \\
& & h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\
& & & h_{n-1} & 2h_{n-1}
\end{bmatrix}$$

$$b = (3(a_1 - a_0)/h_0 - 3f'(a), 3(a_2 - a_1)/h_1 - 3(a_1 - a_0)/h_0, \ldots,$$

$$3(a_n - a_{n-1})/h_{n-1} - 3(a_{n-1} - a_{n-2})/h_{n-2},$$

$$3f'(b) - 3(a_n - a_{n-1})/h_{n-1})^T$$