Interpolation and Polynomial Approximation

1. Weierstrass Approximation Theorem
Suppose $f$ is continuous $[a, b]$. For each $\epsilon > 0$, there exists a polynomial $P(x)$ defined on $[a, b]$, with the property that

$$|f(x) - P(x)| < \epsilon$$

for all $x \in [a, b]$.

2. Polynomial Interpolation
- Problem: Let $f(x_i) = f_i$ for $i = 0, 1, \ldots, n$. Find a polynomial $p(x)$ passing the points $(x_i, f(x_i))$ for $i = 0, 1, \ldots, n$, i.e., $p(x_i) = f_i$ for $i = 0, 1, \ldots, n$.
- Interpolation Theorem:
  There is a unique polynomial of degree $n$ which interpolates $f(x)$ at distinct points $x_0, x_1, \ldots, x_n \in [a, b]$.

3. Method of undetermined coefficients:
- Given $(x_i, f_i)$ for $i = 0, 1, \ldots, n$, choose $1, x, x^2, \ldots, x^n$ as basis functions (monomial basis).
  Then the interpolating polynomial has the form

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n.$$

We want to find coefficients $a_1, a_2, \ldots, a_n$ such that $p(x_i) = f_i$ for $i = 0, 1, \ldots, n$.
- Vandermonde matrix system

$$\begin{bmatrix}
1 & x_1 & \cdots & x_1^n \\
1 & x_2 & \cdots & x_2^n \\
\vdots & \vdots & \ddots & \vdots \\
1 & x_n & \cdots & x_n^n
\end{bmatrix}
\begin{bmatrix}
a_0 \\
a_1 \\
\vdots \\
a_n
\end{bmatrix}
= 
\begin{bmatrix}
f_0 \\
f_1 \\
\vdots \\
f_n
\end{bmatrix}$$

4. Lagrange interpolation
- Lagrange formula

$$p(x) = a_0 l_0(x) + a_1 l_1(x) + \cdots + a_n l_n(x)$$

Let $l_i(x)$ be a polynomial of degree $n$ for $i = 0, 1, \ldots, n$ and

$$l_i(x) = \delta_{ij} = \begin{cases} 
1 & \text{if } i = j \\
0 & \text{if } i \neq j
\end{cases}$$

Then

$$p(x_j) = \sum_{i=0}^{n} a_i l_i(x_j) = a_j l_j(x_j) = a_j = f_j$$

therefore

$$p(x) = \sum_{i=0}^{n} f_i l_i(x).$$
6. Iterative interpolation (Neville’s algorithm)

- Idea: Given \((x_i, f_i)\) for \(i = 0, 1, \ldots, n\), let \(p_{m_1m_2\cdots m_k}(x)\) \((1 \leq m_j \leq n)\) be a polynomial of degree less than \(k\) and

\[
p_{m_1m_2\cdots m_k}(x_{m_j}) = f_{m_j} \quad \text{for } j = 1, \ldots, k
\]

i.e., \(p_{m_1m_2\cdots m_k}(x)\) interpolates \((x_{m_1}, f_{m_1}), (x_{m_2}, f_{m_2}), \ldots, (x_{m_k}, f_{m_k})\).

Interpolating polynomials are linked by the following recursion:

\[
p_{m_1m_2\cdots m_k}(x) = \frac{(x - x_{m_1})p_{m_2m_3\cdots m_k}(x) - (x - x_{m_k})p_{m_1m_2\cdots m_{k-1}}(x)}{x_{m_k} - x_{m_1}}
\leq (1)

\]

for \(k = 2, \ldots, n\).

- \(p_{m_1m_2\cdots m_k}(x)\) defined in (1) interpolates \((x_{m_1}, f_{m_1}), (x_{m_2}, f_{m_2}), \ldots, (x_{m_k}, f_{m_k})\).

If \(x = x_{m_1}\),

\[
p_{m_1m_2\cdots m_k}(x_{m_1}) = 0 - \frac{(x_{m_1} - x_{m_k})p_{m_1m_2\cdots m_{k-1}}(x_{m_1})}{x_{m_k} - x_{m_1}} = p_{m_1m_2\cdots m_{k-1}}(x_{m_1}) = f_{m_1}
\]

If \(x = x_{m_j}\) for \(1 < j < k\),

\[
p_{m_1m_2\cdots m_k}(x_{m_j}) = \frac{(x_{m_j} - x_{m_1})p_{m_2m_3\cdots m_k}(x_{m_j}) - (x_{m_j} - x_{m_k})p_{m_1m_2\cdots m_{k-1}}(x_{m_j})}{x_{m_k} - x_{m_1}} = f_{m_j}
\]

If \(x = x_{m_k}\),

\[
p_{m_1m_2\cdots m_k}(x_{m_k}) = \frac{(x_{m_k} - x_{m_1})p_{m_2m_3\cdots m_k}(x_{m_k}) - 0}{x_{m_k} - x_{m_1}} = p_{m_2m_3\cdots m_k}(x_{m_k}) = f_{m_k}
\]
The result in (1) is used in the Neville’s algorithm as follows:

\[
\begin{align*}
(x_0, f_0) & \quad f_0 = p_0(x) \\
(x_1, f_1) & \quad f_1 = p_1(x) \\
(x_2, f_2) & \quad f_2 = p_2(x) \\
(x_3, f_3) & \quad f_3 = p_3(x) \\
(x_4, f_4) & \quad f_4 = p_4(x) \\
& \vdots \quad \vdots \quad \vdots \quad \vdots 
\end{align*}
\]

Remark: Neville’s algorithm aims at evaluating the interpolating polynomial \( p \) at a single value of \( x \). It is less suitable for determining the interpolating polynomial itself.

7. Newton interpolation

- Given \((x_i, f_i)\) for \(i = 0, 1, \ldots, n\), choose 1, \(x - x_0\), \((x - x_0)(x - x_1)\), \ldots, 
  \((x - x_0)(x - x_1)\ldots(x - x_{n-1})\) as basis functions. Then the interpolating polynomial has the form
  \[
p(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots + a_n(x - x_0)(x - x_1)\ldots(x - x_{n-1}).
\]

We want to find coefficients \(a_0, a_1, \ldots, a_n\) such that \(p(x_i) = f_i\) for \(i = 0, 1, \ldots, n\).

- Undetermined coefficients method

\[
\begin{align*}
p(x_0) &= f_0 & \rightarrow a_0 &= f_0 \\
p(x_1) &= f_1 & \rightarrow a_0 + a_1(x_1 - x_0) &= f_1 \\
& \vdots & \vdots & \vdots \\
p(x_n) &= f_n & \rightarrow a_0 + a_1(x_n - x_0) + a_2(x_n - x_0)(x_n - x_1) + \cdots + a_n(x_n - x_0)\cdots(x_n - x_{n-1}) &= f_n
\end{align*}
\]

The above lower triangular system can be solved by the forward substitution.

- Once coefficients are determined, \( p \) is evaluated by the Horner’s method:

\[
p(x) = a_0 + (x - x_0)(a_1 + (x - x_1)(a_2 + \cdots(a_{n-1} + a_n(x - x_{n-1})\cdots)))
\]

- Divided difference

Define divided differences as follows:

\[
\begin{align*}
f[x_i] &= f_i \quad \text{for } i = 0, 1, \ldots, n \quad \text{(zeroth divided difference)} \\
f[x_i, x_{i+1}] &= \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i} \quad \text{(first divided difference)} \\
f[x_i, x_{i+1}, \ldots, x_{i+k}] &= \frac{f[x_{i+1}, \ldots, x_{i+k}] - f[x_i, \ldots, x_{i+k-1}]}{x_{i+k} - x_i} \quad \text{(kth divided difference)}
\end{align*}
\]

Then it can be shown that

\[
p_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \cdots + f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1)\cdots(x - x_{n-1}).
\]
Generation of divided differences:

\[ f[x_0] = f_0 \]
\[ f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} \]
\[ f[x_1] = f_1 \]
\[ f[x_1, x_2] = \frac{f[x_2] - f[x_1]}{x_2 - x_1} \]
\[ f[x_2] = f_2 \]
\[ f[x_2, x_3] = \frac{f[x_3] - f[x_2]}{x_3 - x_2} \]
\[ f[x_3] = f_3 \]

8. Error Analysis

- **Theorem:**
  Let \( f \in C^{n+1}[a,b] \) and \( p(x) \) be the polynomial which interpolates \( f(x) \) at the points \( x_0, x_1, \ldots, x_n \), where \( x_0 = a, x_n = b \) and \( x_i \in (a, b) \) for \( i = 1, \ldots, n-1 \). Then

  \[ f(x) - p(x) = \frac{f^{(n+1)}(\theta)}{(n+1)!} w(x), \]

  where \( \theta \in [a, b] \) and \( w(x) = (x - x_0)(x - x_1) \cdots (x - x_n) \).

- **Using the theorem, we can get a rough bound of the approximation error:**

  \[ |f(x) - p(x)| \leq \frac{|f^{(n+1)}(\theta)|}{(n+1)!} |w(x)| \]

  \[ \leq \max_{x \in [a,b]} \frac{|f^{(n+1)}(\theta)|}{(n+1)!} (b - a)^{n+1} \quad \text{if } x \in [a, b] \]

- **Remarks**
  a. \( |w(x)| \) grows very fast for \( \hat{x} \) outside \([a, b]\). Therefore, \( p(\hat{x}) \) is usually not a good approximation to \( f(\hat{x}) \).
  b. An interpolating polynomial with higher degree (using more data points) does not always result in a better approximation. Hence, if \( n \) is too large, sometimes it is better to look for a different approximation such as spline interpolation.
9. Hermite cubic interpolation

- Given \((x_i, f_i)\) and \((x_i, f_i')\) for \(i = 0, 1, \ldots, n\), we seek for a piecewise cubic interpolating polynomial \(p(x)\) satisfying the following.
  a. \(p(x)\) is cubic in each interval \([x_i, x_{i+1}]\).
  b. \(p(x)\) and \(p'(x)\) are continuous such that \(p(x_i) = f_i\) and \(p'(x_i) = f_i'\) for \(i = 0, 1, \ldots, n\).

- Undetermined coefficients method
  Let \(p_i(x) = a_i + b_i x + c_i x^2 + d_i x^3\) on \([x_i, x_{i+1}]\). Then from the conditions \(p(x_i) = f_i\), \(p(x_{i+1}) = f_{i+1}\), \(p'(x_i) = f_i'\) and \(p'(x_{i+1}) = f_{i+1}'\), \(4n\) equations with \(4n\) unknowns are derived.

- Basis function method
  Let
  \[
  \phi_{i0}(x) = \left( \frac{x_{i+1} - x}{x_{i+1} - x_i} \right)^2 \left[ 1 + 2 \left( \frac{x - x_i}{x_{i+1} - x_i} \right) \right]
  \]
  \[
  \phi_{i+1,0}(x) = \left( \frac{x - x_i}{x_{i+1} - x_i} \right)^2 \left[ 1 + 2 \left( \frac{x_{i+1} - x}{x_{i+1} - x_i} \right) \right]
  \]
  \[
  \phi_{i1}(x) = \left( \frac{x_{i+1} - x}{x_{i+1} - x_i} \right)^2 (x - x_i)
  \]
  \[
  \phi_{i+1,1}(x) = -\left( \frac{x - x_i}{x_{i+1} - x_i} \right)^2 (x_{i+1} - x)
  \]
  Then, on each \([x_i, x_{i+1}]\),
  \[
  p(x) = f_i \phi_{i0}(x) + f_i' \phi_{i1}(x) + f_{i+1} \phi_{i+1,0}(x) + f_i' \phi_{i+1,1}(x).
  \]
10. Cubic spline interpolation

- Given \((x_i, f_i)\) for \(i = 0, 1, \ldots, n\), we seek for a twice continuously differentiable piecewise polynomial \(p(x)\) which is cubic in each subinterval \([x_i, x_{i+1}]\) and interpolates \((x_i, f_i)\) for \(i = 0, 1, \ldots, n\). Let \(p(x) = p_i(x)\) on \([x_i, x_{i+1}]\) for \(i = 0, 1, \ldots, n - 1\). Then,

  a. \(p_i(x)\) is cubic in \([x_i, x_{i+1}]\) for \(i = 0, 1, \ldots, n - 1\)
  b. \(p_i(x_i) = f_i\) for \(i = 0, 1, \ldots, n - 1\)
  c. \(p_i(x_{i+1}) = f_{i+1}\) for \(i = 0, 1, \ldots, n - 1\)
  d. \(p_i'(x_{i+1}) = p_{i+1}'(x_{i+1})\) for \(i = 0, 1, \ldots, n - 2\)
  e. \(p_i''(x_{i+1}) = p_{i+1}''(x_{i+1})\) for \(i = 0, 1, \ldots, n - 2\)

- There are total 4\(n\) unknowns (coefficients) to be determined. Total 4\(n - 2\) equations (2\(n\) from b, c and 2\(n - 2\) from d, e) are derived from the conditions b–e.

- In order to determine the coefficients uniquely, two more conditions are needed. The following additional conditions are used in practice.
  a. \(p''(x_0) = p''(x_n) = 0\) (natural cubic spline or free boundary spline)
  b. \(p'(x_0) = f'_0\) and \(p'(x_n) = f'_n\) (complete cubic spline or clamped boundary spline)
  c. \(p'(x_0) = p'(x_n)\) and \(p''(x_0) = p''(x_n)\) (periodic cubic spline)

- Calculation of cubic spline
  Since \(p(x)\) is piecewise cubic, \(p'(x)\) and \(p''(x)\) are piecewise quadratic and linear, respectively, and they are continuous. Let \(M_i = p''(x_i)\) for \(i = 0, 1, \ldots, n\) and \(p(x) = p_i(x)\) on \([x_i, x_{i+1}]\) for \(i = 0, 1, \ldots, n - 1\). Then, \(p''_i(x_i) = M_i\) and \(p''_{i+1}(x_{i+1}) = M_{i+1}\). Using the interpolation conditions \(p_i(x_i) = f_i, p_i(x_{i+1}) = f_{i+1}\) and the continuity of \(p'(x)\) at \(x_i\), we can have the following 3-moment equations

\[
\frac{h_i-1}{6} M_{i-1} + \frac{h_i-1 + h_i}{3} M_i + \frac{h_i}{6} M_{i+1} = \frac{f_{i+1} - f_i}{h_i} - \frac{f_i - f_{i-1}}{h_{i-1}}
\]

for \(i = 1, 2, \ldots, n - 1\), where \(h_i = x_{i+1} - x_i\) for \(i = 0, 1, \ldots, n - 1\)
a. natural cubic spline

\[ M_0 = p''(x_0) = 0, \quad M_n = p''(x_n) = 0 \]

The moments \( M_1, \ldots, M_{n-1} \) are found by solving the system

\[
\begin{bmatrix}
\frac{h_0}{3} & \frac{h_0}{6} & 0 \\
\frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & 0 \\
0 & \frac{h_2}{6} & \frac{h_2+h_3}{3} & \frac{h_3}{6} & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots \\
0 & \frac{h_{n-2}}{6} & \frac{h_{n-2}+h_{n-1}}{3} & \frac{h_{n-1}}{6} & \frac{h_{n-1}}{3} \\
0 & \frac{h_{n-1}}{6} & \frac{h_{n-1}}{3} & \frac{h_{n-1}}{6} & 0
\end{bmatrix}
\begin{bmatrix}
M_1 \\
M_2 \\
M_3 \\
\vdots \\
M_{n-1}
\end{bmatrix}
= \begin{bmatrix}
\frac{f_2-f_1}{h_1} - \frac{f_1-f_0}{h_0} \\
\frac{f_3-f_2}{h_2} - \frac{f_2-f_1}{h_1} \\
\frac{f_4-f_3}{h_3} - \frac{f_3-f_2}{h_2} \\
\vdots \\
\frac{f_n-f_{n-1}}{h_{n-1}} - \frac{f_{n-1}-f_{n-2}}{h_{n-2}}
\end{bmatrix}
\]

b. complete cubic spline

From the conditions \( p'(x_0) = f_0' \) and \( p'(x_n) = f_n' \), we get two extra equations

\[
\frac{h_0}{6} M_0 + \frac{h_0}{6} M_1 = \frac{f_1-f_0}{h_0} - f_0' \\
\frac{h_{n-1}}{6} M_{n-1} + \frac{h_{n-1}}{6} M_n = \frac{f_n-f_{n-1}}{h_{n-1}} + f_n'
\]

The moments \( M_0, \ldots, M_n \) are found by solving the system

\[
\begin{bmatrix}
\frac{h_0}{3} & \frac{h_0}{6} & 0 \\
\frac{h_0}{6} & \frac{h_0+h_1}{3} & \frac{h_1}{6} & 0 \\
0 & \frac{h_1}{6} & \frac{h_1+h_2}{3} & \frac{h_2}{6} & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots \\
0 & \frac{h_{n-2}}{6} & \frac{h_{n-2}+h_{n-1}}{3} & \frac{h_{n-1}}{6} & \frac{h_{n-1}}{3} \\
0 & \frac{h_{n-1}}{6} & \frac{h_{n-1}}{3} & \frac{h_{n-1}}{6} & 0
\end{bmatrix}
\begin{bmatrix}
M_0 \\
M_1 \\
M_2 \\
\vdots \\
M_{n-1} \\
M_n
\end{bmatrix}
= \begin{bmatrix}
\frac{f_1-f_0}{h_0} - f_0' \\
\frac{f_2-f_1}{h_1} - \frac{f_1-f_0}{h_0} \\
\frac{f_3-f_2}{h_2} - \frac{f_2-f_1}{h_1} \\
\vdots \\
\frac{f_n-f_{n-1}}{h_{n-1}} - \frac{f_{n-1}-f_{n-2}}{h_{n-2}} \\
\frac{f_n'-f_{n-1}}{h_{n-1}} + f_n'
\end{bmatrix}
\]

c. In both cases, coefficients matrices are tridiagonal, symmetric, positive definite and diagonally dominant.

d. On \([x_i, x_{i+1}]\), \( p(x) \) can be written as follows:

\[
p(x) = f_i + \left( \frac{f_{i+1} - f_i}{h_i} - \frac{2M_i + M_{i+1}}{6h_i}h_i \right) (x - x_i) + \frac{M_i}{2}(x - x_i)^2 + \frac{M_{i+1} - M_i}{6h_i}(x - x_i)^3
\]
Properties of cubic spline interpolating polynomials

a. Out of all functions in $C^2[x_0, x_n]$ which interpolate $(x_i, f_i)$ for $i = 0, 1, \ldots, n$, the natural cubic spline $p(x)$ has the smallest 2nd derivative measured in $L^2$-norm, i.e.,

$$\int_{x_0}^{x_n} (p''(x))^2 \, dx \leq \int_{x_0}^{x_n} (h''(x))^2 \, dx$$

for any $h \in C^2[x_0, x_n]$ interpolates $(x_i, f_i)$ for $i = 0, 1, \ldots, n$.

b. If $f \in C^4[x_0, x_n]$ and $p(x)$ is a cubic spline which approximates $f$,

$$\max_{x_i \leq x \leq x_{i+1}} |f(x) - p(x)| \leq \frac{\max_{x_i \leq x \leq x_{i+1}} |f^{(4)}(x)|}{4!} (x_{i+1} - x_i)^4.$$