Chapter 6  Numerical Interpolation

• Given some sample measurements, we may need to estimate the values at some other points by fitting a curve to the measured points.

• Interpolation is required in many engineering applications that use tabular data as input.

• The basis of all interpolation algorithms is the fitting of some type of curve or function to a subset of the tabular data.
Method of Undetermined Coefficients

- The $n$th-order polynomial is used as the interpolation function:

$$f(x) = b_0 + b_1x + b_2x^2 + b_3x^3 + \cdots + b_nx^n$$

- Example: To estimate $(2.2)^3$ by using the following data (obtained from the cubic function).

<table>
<thead>
<tr>
<th>$x$</th>
<th>$x^3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>64</td>
</tr>
</tbody>
</table>
• The linear interpolation function:

\[ f(x) = b_0 + b_1 x \]

The linear interpolation form in algebra

\[ f(x) = f(x_i) + \frac{x - x_i}{x_{i+1} - x_i} [f(x_{i+1}) - f(x_i)] \]

Substituting \( x_i = 2 \) and \( x_{i+1} = 3 \), we get

\[ f(x) = 8 + \frac{x - 2}{3 - 2} (27 - 8) = -30 + 19x \]

The prediction of \((2.2)^3\) is

\[ f(2.2) = 11.8 \]

The true value of \((2.2)^3\) is 10.648. The error is 10.8%.
• The second-order polynomial function:

\[ f(x) = b_0 + b_1x + b_2x^2 \]

Set \( x=1, \ x=2, \) and \( x=3. \) We get three simultaneous equations:

1. \( 1 = b_0 + b_1 + b_2 \)
2. \( 8 = b_0 + 2b_1 + 4b_2 \)
3. \( 27 = b_0 + 3b_1 + 9b_2 \)

Solving the equations, we obtain \( b_0=6, \ b_1=-11, \) and \( b_2=6. \) The interpolation function is

\[ f(x) = 6 - 11x + 6x^2 \]

The prediction of \((2.2)^3\) is \( f(2.2)=10.8640. \) The error is 1.8%.

• The \( n \)th-order polynomial interpolation function requires \( n+1 \) distinct data points for determining its coefficients.
Gregory-Newton Interpolation Method

• A higher order interpolation polynomial has more accuracy. However, it becomes more complex to solving the larger set of simultaneous equations.

• The $n$th-order Gregory-Newton interpolation formula:

$$f(x) = a_1 + a_2 (x - x_1) + a_3 (x - x_1)(x - x_2) +$$

$$a_4 (x - x_1)(x - x_2)(x - x_3) + \Lambda$$

$$+ a_n (x - x_1)(x - x_2)\Lambda (x - x_{n-1}) + a_{n+1} (x - x_1)(x - x_2)\Lambda (x - x_n)$$

where $x_i, 1 \leq i \leq n$, are $n$ known values of the independent variable $x$, and $a_i, 1 \leq i \leq n$, are $n$ unknown coefficients.
• We can obtain the values of $a_1, a_2, a_3, \ldots, a_{n+1}$ by the following procedure:

$$a_1 = f(x_1)$$

$$f(x_2) = a_1 + a_2(x_2-x_1)$$

$$a_2 = \frac{f(x_2) - a_1}{x_2 - x_1} = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

$$f(x_3) = a_1 + a_2(x_3-x_1) + a_3(x_3 - x_1)(x_3 - x_2)$$

$$a_3 = \frac{f(x_3) - a_1 - a_2(x_3-x_1)}{(x_3 - x_1)(x_3 - x_2)}$$

• The Gregory-Newton method yields the same polynomial as the solution of the simultaneous equations in the method of undetermined coefficients.
Example: Cubic Function

Using pair (1,1), we get

\[ a_1 = f(x_1) = 1 \]

Using pair (2,8), we get

\[ a_2 = \frac{8 - 1}{2 - 1} = 7 \]

Using pair (3,27), we get

\[ a_3 = \frac{27 - 1 - 7(3 - 1)}{(3 - 1)(3 - 2)} = 6 \]

Thus, we have the following interpolation on polynomial

\[ f(x) = 1 + 7(x-1) + 6(x-1)(x-2) \]

\[ = 6 - 11x + 6x^2 \]

For \( x = 2.2 \), we get \( f(2.2) = 1 + 7(2.2-1) + 6(2.2-1)(2.2-2) = 10.84 \)
Finite-Difference Interpolation

- This scheme can be applied when the known values of the independent variable are equally spaced.

The first finite difference of \( y = f(x) \):
\[
\Delta f = f(x + \Delta x) - f(x)
\]
The second finite difference:
\[
\Delta^2 f = \Delta[\Delta f] = [f(x + 2\Delta x) - f(x + \Delta x)] - [f(x + \Delta x) - f(x)]
\]
The \( n \)th finite difference:
\[
\Delta^n f = \Delta[\Delta^{n-1} f]
\]
• If the original polynomial is of degree $n$, then the $i$th finite difference is a polynomial of degree $n-i$.

Assume $n$th-order polynomial:

$$f(x) = b_0 + b_1 x + b_2 x^2 + \cdots + b_{n-1} x^{n-1} + b_n x^n$$

where $b_n \neq 0$.

$$f(x + \Delta x) = b_0 + b_1 (x + \Delta x) + b_2 (x + \Delta x)^2 + \cdots + b_n (x + \Delta x)^n$$

$$= b_n x^n + (b_{n-1} + n \Delta x b_n) x^{n-1} + \Lambda$$

$$\Delta f = f(x + \Delta x) - f(x)$$

$$= n \Delta x b_n x^{n-1} + \Lambda \quad \quad \text{--- a polynomial of degree } n - 1$$
\[ f(x + 2\Delta x) = b_n(x + 2\Delta x)^n + b_{n-1}(x + 2\Delta x)^{n-1} + \Lambda \]
\[ \Delta^2 f = [f(x + 2\Delta x) - f(x + \Delta x)] - [f(x + \Delta x) - f(x)] \]
\[ = n(n-1)(\Delta x)^2 b_n x^{n-2} + \Lambda \]
(a polynomial of degree \( n-2 \))
the \( i \)th finite difference:
\[ \Delta^i f = n(n-1)(n-2)\Lambda (n-i+1)(\Delta x)^i b_n x^{n-i} + \Lambda \]
the \( n \)th finite difference:
\[ \Delta^n f = n(n-1)(n-2)\Lambda (1)(\Delta x)^n b_n = n!(\Delta x)^n b_n \]
For $n = 2$, we have

\[ f(x) = b_2 x^2 + b_1 x + b_0 \]

\[ f(x + \Delta x) = b_2 (x + \Delta x)^2 + b_1 (x + \Delta x) + b_0 \]

\[ = b_2 (x^2 + 2x\Delta x + (\Delta x)^2) + b_1 (x + \Delta x) + b_0 \]

\[ = b_2 x^2 + (2\Delta x b_2 + b_1) x + (b_2 (\Delta x)^2 + b_1 \Delta x + b_0) \]

\[ f(x + 2\Delta x) = b_2 (x + 2\Delta x)^2 + b_1 (x + 2\Delta x) + b_0 \]

\[ = b_2 (x^2 + 4x\Delta x + (2\Delta x)^2) + b_1 (x + 2\Delta x) + b_0 \]

\[ = b_2 x^2 + (4b_2 \Delta x + b_1) x + (b_2 (2\Delta x)^2 + 2b_1 \Delta x + b_0) \]

\[ \Delta f = f(x + \Delta x) - f(x) \]

\[ = 2b_2 (\Delta x) x + (b_2 (\Delta x)^2 + b_1 \Delta x) \]

\[ \Delta^2 f = [ f(x + 2\Delta x) - f(x + \Delta x) ] - [ f(x + \Delta x) - f(x) ] \]

\[ = 2b_2 (\Delta x)^2 \]
Example: Finite-Difference Interpolation

Table: Data for Finite-difference Interpolation

<table>
<thead>
<tr>
<th>$x$</th>
<th>$f(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
</tr>
</tbody>
</table>

$\Delta f = f(x + \Delta x) - f(x)$

The known values are equally spaced with $\Delta x = 1$.

$5 - 3 = 2 = 2b_2(\Delta x)x + b_2(\Delta x)^2 + b_1\Delta x = 3b_2 + b_1$

$\Delta^2 f = [8 - 5] - [5 - 3] = 1 = 2b_2(\Delta x)^2 = 2b_2$

$f(1) = 3 = b_2 + b_1 + b_0$

We get $b_2 = 0.5$, $b_1 = 0.5$, $b_0 = 2$. $f(x) = 0.5x^2 + 0.5x + 2$

We can compute $f(2.3) = 5.795$
## Finite Difference Table

<table>
<thead>
<tr>
<th>( x )</th>
<th>( f(x) )</th>
<th>( \Delta f )</th>
<th>( \Delta^2 f )</th>
<th>( \Delta^3 f )</th>
<th>( \Lambda )</th>
<th>( \Delta^n f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x )</td>
<td>( f(x) )</td>
<td>( \Delta f(x) )</td>
<td>( \Delta^2 f(x) )</td>
<td>( \Delta^3 f(x) )</td>
<td>( \Lambda )</td>
<td>( \Delta^n f(x) )</td>
</tr>
<tr>
<td>( x + \Delta x )</td>
<td>( f(x + \Delta x) )</td>
<td>( \Delta f(x + \Delta x) )</td>
<td>( \Delta^2 f(x + \Delta x) )</td>
<td>( \Delta^3 f(x + \Delta x) )</td>
<td>( \Lambda )</td>
<td>( \Delta^n f(x + \Delta x) )</td>
</tr>
<tr>
<td>( x + 2\Delta x )</td>
<td>( f(x + 2\Delta x) )</td>
<td>( \Delta f(x + 2\Delta x) )</td>
<td>( \Delta^2 f(x + 2\Delta x) )</td>
<td>( \Delta^3 f(x + 2\Delta x) )</td>
<td>( \Lambda )</td>
<td>( \Delta^n f(x + 2\Delta x) )</td>
</tr>
<tr>
<td>( x + 3\Delta x )</td>
<td>( f(x + 3\Delta x) )</td>
<td>( \Delta f(x + 3\Delta x) )</td>
<td>( \Delta^2 f(x + 3\Delta x) )</td>
<td>( \Delta^3 f(x + 3\Delta x) )</td>
<td>( \Lambda )</td>
<td>( \Delta^n f(x + 3\Delta x) )</td>
</tr>
<tr>
<td>( x + (n - 2)\Delta x )</td>
<td>( f[x + (n - 2)\Delta x] )</td>
<td>( \Delta f[x + (n - 2)\Delta x] )</td>
<td>( \Delta^2 f[x + (n - 3)\Delta x] )</td>
<td>( \Delta^3 f[x + (n - 3)\Delta x] )</td>
<td>( \Lambda )</td>
<td>( \Delta^n f(x) )</td>
</tr>
<tr>
<td>( x + (n - 1)\Delta x )</td>
<td>( f[x + (n - 1)\Delta x] )</td>
<td>( \Delta f[x + (n - 2)\Delta x] )</td>
<td>( \Delta^2 f[x + (n - 2)\Delta x] )</td>
<td>( \Delta^3 f[x + (n - 3)\Delta x] )</td>
<td>( \Lambda )</td>
<td>( \Delta^n f(x) )</td>
</tr>
<tr>
<td>( x + n\Delta x )</td>
<td>( f(x + n\Delta x) )</td>
<td>( \Delta f[x + (n - 1)\Delta x] )</td>
<td>( \Delta^2 f[x + (n - 2)\Delta x] )</td>
<td>( \Delta^3 f[x + (n - 3)\Delta x] )</td>
<td>( \Lambda )</td>
<td>( \Delta^n f(x) )</td>
</tr>
</tbody>
</table>
Example: Finite-difference Table for Cubes

<table>
<thead>
<tr>
<th>$x$</th>
<th>$f(x)$</th>
<th>$\Delta f$</th>
<th>$\Delta^2 f$</th>
<th>$\Delta^3 f$</th>
<th>$\Delta^4 f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>331</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1331</td>
<td>66</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>397</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1728</td>
<td>72</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>469</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>2197</td>
<td>78</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>547</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>2744</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Newton’s Method

\[ f(x + \Delta x) = f(x) + \Delta f(x) \]

\[ \Delta f(x + \Delta x) = \Delta f(x) + \Delta^2 f(x) \]

\[ \Delta^2 f(x + \Delta x) = \Delta^2 f(x) + \Delta^3 f(x) \]

\[ \Delta^{n-1} f(x + \Delta x) = \Delta^{n-1} f(x) + \Delta^n f(x) \]

\[ f(x + 2\Delta x) = f(x) + \Delta f(x) + \Delta f(x + \Delta x) \]

\[ = f(x) + 2\Delta f(x) + \Delta^2 f(x) \]

\[ \Delta f(x + 2\Delta x) = \Delta f(x) + 2\Delta^2 f(x) + \Delta^3 f(x) \]

\[ \Delta^2 f(x + 2\Delta x) = \Delta^2 f(x) + 2\Delta^3 f(x) + \Delta^4 f(x) \]

\[ \Delta^{n-3} f(x + 2\Delta x) = \Delta^{n-3} f(x) + 2\Delta^{n-2} f(x) + \Delta^{n-1} f(x) \]
The general form:

\[ f(x + m\Delta x) = \sum_{i=0}^{m} C_i^m \Delta^i f(x) \]

\[ = f(x) + m\Delta f(x) + \frac{m(m-1)}{2} \Delta^2 f(x) + \Lambda + \Delta^m f(x) \]

where \( C_i^m \) denotes the constant of a binomial expansion.

More general form:

\[ \Delta^r f(x + m\Delta x) = \sum_{i=0}^{m} C_i^m \Delta^{r+i} f(x) \]

\[ = \Delta^r f(x) + m\Delta^{r+1} f(x) + \Lambda + \Delta^{r+m} f(x) \]
• Newton’s interpolation formula:

\[
f(x) = f(x_0) + nf(x_0) + \frac{n(n-1)}{2!} \Delta^2 f(x_0) + \Lambda \\
+ \frac{n(n-1)\Lambda (n-m+1)}{m!} \Delta^m f(x_0) + \Lambda + \Delta^n f(x)
\]

in which

\[x = x_0 + n\Delta x\]

or

\[n = \frac{x - x_0}{\Delta x}\]
• Example: Tangents of Angles
  
  – Assume the tangents of angles (θ) between 30° and 40° with an increment of 2° is available in the next table.
  
  – We want to develop a nonlinear interpolation polynomial to estimate tan(36.5).

\[
\tan(\theta) = 0.5774 + n(0.0475) + \frac{n(n-1)}{2!}(0.0021) + \frac{n(n-1)(n-2)}{3!}(0.0003) + \frac{n(n-1)(n-2)(n-3)}{4!}(0.0001) + \frac{n(n-1)(n-2)(n-3)(n-4)}{5!}(-0.0003)
\]
Table: Finite-difference Table for $\tan \theta$ where $30^\circ \leq \theta \leq 40^\circ$

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>$\tan \theta$</th>
<th>$\Delta \tan \theta$</th>
<th>$\Delta^2 \tan \theta$</th>
<th>$\Delta^3 \tan \theta$</th>
<th>$\Delta^4 \tan \theta$</th>
<th>$\Delta^5 \tan \theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$30^\circ$</td>
<td>0.5774</td>
<td>0.0475</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$32^\circ$</td>
<td>0.6249</td>
<td>0.0021</td>
<td>0.0496</td>
<td>0.0003</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$34^\circ$</td>
<td>0.6745</td>
<td>0.0024</td>
<td>0.0520</td>
<td>0.0004</td>
<td>$-0.0003$</td>
<td></td>
</tr>
<tr>
<td>$36^\circ$</td>
<td>0.7265</td>
<td>0.0028</td>
<td>0.0548</td>
<td>0.0002</td>
<td></td>
<td>$-0.0002$</td>
</tr>
<tr>
<td>$38^\circ$</td>
<td>0.7813</td>
<td>0.0030</td>
<td>0.0578</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$40^\circ$</td>
<td>0.8391</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
For $\theta = 36.5^\circ$,

\[ n = \frac{36.5 - 30}{2} = 3.25 \]

\[
tan(36.5) = 0.5774 + 3.25(0.0475) + \frac{3.25(2.25)}{2}(0.0021) + \frac{3.25(2.25)(1.25)}{6}(0.0003) + \frac{(3.25)(2.25)(1.25)(0.25)}{24}(0.0001) + \frac{3.25(2.25)(1.25)(0.25)(-0.75)}{120}(-0.0003) \]

\[ = 0.73992 \]

The true value of $tan(36.5)$ is 0.73996. Thus, the error is 0.00004.
Lagrange Polynomials

- In Newton’s method, it is assumed that $x$ is measured at a constant interval, $\Delta x$.
- In Lagrange’s method, $x$ is measured on a variable interval $\Delta x$. 
Lagrangian interpolating equation for estimating $f(x_0)$:

$$f(x_0) = \sum_{i=1}^{n} w_i(x_0) f(x_i)$$

$$w_i(x_0) = \frac{\prod_{j=1}^{n} (x_0 - x_j)}{\prod_{j=1}^{n} (x_i - x_j)}$$

where the value of $f(x_i)$ has been measured.
For example, \( n = 3 \)

\[
f(x_0) = w_1(x_0)f(x_1) + w_2(x_0)f(x_2) + w_3(x_0)f(x_3)
\]

\[
w_1(x_0) = \frac{(x_0 - x_2)(x_0 - x_3)}{(x_1 - x_2)(x_1 - x_3)}
\]

\[
w_2(x_0) = \frac{(x_0 - x_1)(x_0 - x_3)}{(x_2 - x_1)(x_2 - x_3)}
\]

\[
w_3(x_0) = \frac{(x_0 - x_1)(x_0 - x_2)}{(x_3 - x_1)(x_3 - x_2)}
\]
• Example: Sine Function

<table>
<thead>
<tr>
<th>Table: Sine Function Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i )</td>
</tr>
<tr>
<td>-----</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

\[
w_1(12) = \frac{(12 - 11)(12 - 13)}{(10 - 11)(10 - 13)} = -\frac{1}{3}
\]

\[
w_2(12) = \frac{(12 - 11)(12 - 13)}{(11 - 10)(11 - 13)} = 1.0
\]

\[
w_3(12) = \frac{(12 - 10)(12 - 11)}{(13 - 10)(13 - 11)} = \frac{1}{3}
\]

\[
f(12) = w_1(12)f(x_1) + w_2(12)f(x_2) + w_3(12)f(x_3)
\]

\[
= -\frac{1}{3}(0.17365) + 1.0(0.19081) + \frac{1}{3}(0.22495)
\]

\[
= 0.20791
\]

The computed value equals the true value to 5 significant digits.
Linear Splines (First-order Spline)

\[ f_1(x) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1) \quad \text{for } x_1 \leq x \leq x_2 \]

\[ f_2(x) = f(x_2) + \frac{f(x_3) - f(x_2)}{x_3 - x_2}(x - x_2) \quad \text{for } x_2 \leq x \leq x_3 \]

\[ \vdots \]

\[ f_i(x) = f(x_i) + \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}(x - x_i) \quad \text{for } x_i \leq x \leq x_{i+1} \]

\[ \vdots \]

\[ f_{n-1}(x) = f(x_{n-1}) + \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}(x - x_{n-1}) \quad \text{for } x_{n-1} \leq x \leq x_n \]
### Example: Linear Spline for Well-water Elevation

<table>
<thead>
<tr>
<th>( i )</th>
<th>( x_i ) (ft)</th>
<th>( f(x_i) ) (ft)</th>
<th>Linear Spline ( f_i(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>14.6</td>
<td>( 14.6 + \frac{10.7 - 14.6}{42 - 15}(x - 15) = 14.6 - 0.1444(x - 15) )</td>
</tr>
<tr>
<td>2</td>
<td>42</td>
<td>10.7</td>
<td>( 10.7 - 0.0686046(x - 42) )</td>
</tr>
<tr>
<td>3</td>
<td>128</td>
<td>4.8</td>
<td>( 4.8 - 0.0164021(x - 128) )</td>
</tr>
<tr>
<td>4</td>
<td>317</td>
<td>1.7</td>
<td>( 1.7 - 0.0120689(x - 317) )</td>
</tr>
<tr>
<td>5</td>
<td>433</td>
<td>0.3</td>
<td></td>
</tr>
</tbody>
</table>

*Mukherjee Intro to Numerical Methods*
• Figure: Linear Spline for Well-water Elevation
Quadratic Splines (Second-order Splines)

- The general form of a quadratic equation between $(x_i, f(x_i))$ and $(x_{i+1}, f(x_{i+1}))$:
  \[ f_i(x) = a_i x^2 + b_i x + c_i \quad \text{for } i = 1, 2, \Lambda, n - 1 \]

- Every two adjacent data points have an interpolation equation as above.

- **Rule 1:** The splines must pass through the data points.

  \[
  f_i(x_i) = a_i x_i^2 + b_i x_i + c_i = f(x_i) \quad \text{for } i = 1, 2, 3, \Lambda, n - 1 \\
  f_i(x_{i+1}) = a_i x_{i+1}^2 + b_i x_{i+1} + c_i = f(x_{i+1}) \quad \text{for } i = 1, 2, 3, \Lambda, n - 1
  \]
• **Rule 2**: At one interior point, the **first derivatives** (slope) of the two interpolation functions representing it must be **equal**.

\[ 2a_i x_{i+1} + b_i = 2a_{i+1} x_{i+1} + b_{i+1} \quad \text{for } i = 1, 2, 3, \ldots, n-2 \]

• **Rule 3**: The second derivative for the spline between the first two data points is zero. That is, the function should be linear.

\[ 2a_1 = 0 \]

• The above equations provide **3(n-1) conditions** to solve the **3(n-1) unknowns** \( a_i, b_i, c_i, \ i=1, 2, \ldots, n-1 \).
• Example: Quadratic Spline for Well-water Elevation

By rule 1 (a):

\[255a_1 + 15b_2 + c_1 = 14.6\]
\[1764a_2 + 42b_2 + c_2 = 10.7\]
\[16384a_3 + 128b_3 + c_3 = 4.8\]
\[100489a_4 + 317b_4 + c_4 = 1.7\]

By rule 1(b):

\[1764a_1 + 42b_1 + c_1 = 10.7\]
\[16384a_2 + 128b_2 + c_2 = 4.8\]
\[100489a_3 + 317b_3 + c_3 = 1.7\]
\[187489a_4 + 433b_4 + c_4 = 0.3\]
By rule 2:
\[
2a_1(42) + b_1 = 2a_2(42) + b_2 \\
2a_2(128) + b_2 = 2a_3(128) + b_3 \\
2a_3(317) + b_3 = 2a_4(317) + b_4
\]

By rule 3:
\[
a_1 = 0
\]
\[
\begin{bmatrix}
15 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1764 & 42 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 16384 & 128 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100489 & 317 & 1 & 0 \\
42 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 16384 & 128 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 100489 & 317 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 187489 & 433 & 1 & 0 \\
1 & 0 & -84 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 256 & 1 & 0 & -256 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 634 & 1 & 0 & -634 & -1 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
b_1 \\
c_1 \\
a_2 \\
b_2 \\
c_2 \\
a_3 \\
b_3 \\
c_3 \\
a_4 \\
b_4 \\
c_4 
\end{bmatrix}
= \begin{bmatrix} 14.6 \\
10.7 \\
4.8 \\
1.7 \\
10.7 \\
4.8 \\
1.7 \\
0.3 \\
0 \\
0 \\
0 \end{bmatrix}
\]
The solution of the system of the equations:

\[
\begin{bmatrix}
    b_1 \\
    c_1 \\
    a_2 \\
    b_2 \\
    c_2 \\
    a_3 \\
    b_3 \\
    c_3 \\
    a_4 \\
    b_4 \\
    c_4
\end{bmatrix} = \begin{bmatrix}
    -0.144444 \\
    16.766667 \\
    0.000818583 \\
    -0.2185206 \\
    18.32227 \\
    -0.00012506 \\
    0.03925179 \\
    1.824836 \\
    0.0002411243 \\
    -0.1929122 \\
    38.62283
\end{bmatrix}
\]
• Figure: Quadratic Spline for Well-water Elevation
Cubic Splines (Third-order Spline)

• The general form of a cubic equation between 
$$(x_i, f(x_i))$$ and $$((x_{i+1}, f(x_{i+1}))$$:

$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \quad \text{for } i = 1, 2, \ldots, n - 1$$

• Every two adjacent data points have an interpolation equation as above.

• Rule 1: The splines must pass through the data points.

$$f_i(x_i) = a_i x_i^3 + b_i x_i^2 + c_i x_i + d_i = f(x_i) \quad \text{for } i = 1, 2, \ldots, n - 1$$

$$f_i(x_{i+1}) = a_i x_{i+1}^3 + b_i x_{i+1}^2 + c_i x_{i+1} + d_i = f(x_{i+1}) \quad \text{for } i = 1, 2, \ldots, n - 1$$
• **Rule 2:** At one point, the first derivatives (slopes) of the two interpolation functions representing it must be equal.

\[ 3a_i x_{i+1}^2 + 2b_i x_{i+1} + c_i = 3a_{i+1} x_{i+1}^2 + 2b_{i+1} x_{i+1} + c_{i+1} \]
for \( i = 1, 2, \Lambda, n - 2 \)

• **Rule 3:** The second derivatives must be equal at the interior points.

\[ 6a_i x_{i+1} + 2b_i = 6a_{i+1} x_{i+1} + 2b_{i+1} \]
for \( i = 1, 2, \Lambda, n - 2 \)
• **Rule 4**: The second derivatives for the spline at the first and last data points are zero.

\[ 6a_1 x_1 + 2b_1 = 0 \]
\[ 6a_{n-1} x_n + 2b_{n-1} = 0 \]

• The above equations provide \(4(n-1)\) conditions to solve the \(4(n-1)\) unknowns \(a_i, b_i, c_i, d_i, i=1,2,\ldots, n-1\).

• Higher-order splines can be developed similarly. However, the computational difficulty greatly increases by needing to solve larger system of simultaneous equations.
Guidelines for Choice of Interpolation Method

• The method of undetermined coefficients is perhaps the easiest method to understand conceptually. However, the method does require the solution of a set of \( n+1 \) simultaneous equations for an \( n \)th-order interpolation polynomial. One advantage of this method is that the interpolation polynomial can be customized for the particular problem at hand; for example, some of the terms of the general \( n \)th-order interpolation polynomial can be dropped, if desired. This is not possible in most of the other interpolation methods.
• Both the **Gregory-Newton** and **Lagrange polynomial** interpolation methods eliminate the need to solve a set of simultaneous equations. Of the two methods, the **Lagrange polynomial scheme** is perhaps the easier to program. Although both of these methods (as well as the method of undetermined coefficients) work with **unequally spaced** data points, the best accuracy is usually achieved when the differences in the spacing of the data points are minimal.

• The **finite-difference interpolation** scheme requires **equally spaced** data points for its application.

• Interpolation using **splines** can be more accurate than other methods, especially in cases where the data show large sudden changes.

• All the interpolation schemes can also be used for **extrapolation**: but the accuracy usually becomes worse.
Linear Interpolation in Two Dimensions

- The form of the data set for 2 independent variables $x_1$ and $x_2$:

<table>
<thead>
<tr>
<th></th>
<th>$x_{11}$</th>
<th>$x_{12}$</th>
<th>$x_{1i}$</th>
<th>$\Lambda$</th>
<th>$x_{1n}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{21}$</td>
<td>$f(x_{11}, x_{21})$</td>
<td>$f(x_{12}, x_{21})$</td>
<td>$f(x_{1i}, x_{21})$</td>
<td>$\Lambda$</td>
<td>$f(x_{1n}, x_{21})$</td>
</tr>
<tr>
<td>$x_{22}$</td>
<td>$f(x_{11}, x_{22})$</td>
<td>$f(x_{12}, x_{22})$</td>
<td>$f(x_{1i}, x_{22})$</td>
<td>$\Lambda$</td>
<td>$f(x_{1n}, x_{22})$</td>
</tr>
<tr>
<td>$x_{2j}$</td>
<td>$f(x_{11}, x_{2j})$</td>
<td>$f(x_{12}, x_{2j})$</td>
<td>$f(x_{1i}, x_{2j})$</td>
<td>$\Lambda$</td>
<td>$f(x_{1n}, x_{2j})$</td>
</tr>
<tr>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>$x_{2m}$</td>
<td>$f(x_{11}, x_{2m})$</td>
<td>$f(x_{12}, x_{2m})$</td>
<td>$f(x_{1i}, x_{2m})$</td>
<td>$\Lambda$</td>
<td>$f(x_{1n}, x_{2m})$</td>
</tr>
</tbody>
</table>
• For given values of $x_1$ and $x_2$, the value of $f(x_1,x_2)$ needs to be determined.

<table>
<thead>
<tr>
<th></th>
<th>$x_{1i}$</th>
<th>$x_1$</th>
<th>$x_{1(i+1)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{2j}$</td>
<td>$f(x_{1i},x_{2j})$</td>
<td>$f(x_1,x_{2j})$</td>
<td>$f(x_{1(i+1)},x_{2j})$</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$f(x_{1i},x_2)$</td>
<td>$f(x_1,x_2)$</td>
<td>$f(x_{1(i+1)},x_2)$</td>
</tr>
<tr>
<td>$x_{2(j+1)}$</td>
<td>$f(x_{1i},x_{2(j+1)})$</td>
<td>$f(x_1,x_{2(j+1)})$</td>
<td>$f(x_{1(i+1)},x_{2(j+1)})$</td>
</tr>
</tbody>
</table>

The values at the four corners are known.

**Step 1:** A one-dimensional interpolation over $x_1$ to obtain $f(x_1,x_{2j})$:

$$f(x_1,x_{2j}) = f(x_{1i},x_{2j}) + \frac{f(x_{1(i+1)},x_{2j}) - f(x_{1i},x_{2j})}{x_{1(i+1)} - x_{1i}}(x_1 - x_{1i})$$
Step 2: A one-dimensional interpolation over $x_1$ to obtain $f(x_1, x_{2(j+1)})$:

$$f(x_1, x_{2(j+1)}) = f(x_{1i}, x_{2(j+1)}) + \frac{f(x_{1(i+1)}, x_{2(j+1)}) - f(x_{1i}, x_{2(j+1)})}{x_{1(i+1)} - x_{1i}} (x_1 - x_{1i})$$

Step 3: A one-dimensional interpolation over $x_2$ to obtain $f(x_1, x_2)$:

$$f(x_1, x_2) = f(x_{1i}, x_{2j}) + \frac{f(x_{1}, x_{2(j+1)}) - f(x_{1}, x_{2j})}{x_{2(j+1)} - x_{2j}} (x_2 - x_{2j})$$