Interpolation, Extrapolation & Polynomial Approximation

November 20, 2016
Introduction

In many cases we know the values of a function \( f(x) \) at a set of points \( x_1, x_2, ..., x_N \), but we don’t have the analytic expression of the function that lets us calculate its value at an arbitrary point. We will try to estimate \( f(x) \) for arbitrary \( x \) by “drawing” a curve through the \( x_i \) and sometimes beyond them.

The procedure of estimating the value of \( f(x) \) for \( x \in [x_1, x_N] \) is called **interpolation** while if the value is for points \( x \notin [x_1, x_N] \) **extrapolation**.

The form of the function that approximates the set of points should be a convenient one and should be applicable to a general class of problems.
Polynomial functions are the most common ones while rational and trigonometric functions are used quite frequently.

We will study the following methods for polynomial approximations:

- Lagrange’s Polynomial
- Hermite Polynomial
- Taylor Polynomial
- Cubic Splines
Lagrange Polynomial

Let’s assume the following set of data:

<table>
<thead>
<tr>
<th>x</th>
<th>x₀</th>
<th>x₁</th>
<th>x₂</th>
<th>x₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>3.2</td>
<td>2.7</td>
<td>1.0</td>
<td>4.8</td>
</tr>
<tr>
<td>f(x)</td>
<td>22.0</td>
<td>17.8</td>
<td>14.2</td>
<td>38.3</td>
</tr>
<tr>
<td>f₀</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f₁</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f₂</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f₃</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Then the interpolating polynomial will be of 4th order i.e. \( ax^3 + bx^2 + cx + d = P(x) \). This leads to 4 equations for the 4 unknown coefficients and by solving this system we get \( a = -0.5275 \), \( b = 6.4952 \), \( c = -16.117 \) and \( d = 24.3499 \) and the polynomial is:

\[
P(x) = -0.5275x^3 + 6.4952x^2 - 16.117x + 24.3499
\]

It is obvious that this procedure is quite labourous and Lagrange developed a direct way to find the polynomial

\[
P_n(x) = f_0L_0(x) + f_1L_1(x) + \ldots + f_nL_n(x) = \sum_{i=0}^{n} f_iL_i(x)
\]

where \( L_i(x) \) are the Lagrange coefficient polynomials.
\[ L_j(x) = \frac{(x - x_0)(x - x_1)\ldots(x - x_{j-1})(x - x_{j+1})\ldots(x - x_n)}{(x_j - x_0)(x_j - x_1)\ldots(x_j - x_{j-1})(x_j - x_{j+1})\ldots(x_j - x_n)} \]  

and obviously:
\[ L_j(x_k) = \delta_{jk} = \begin{cases} 
0 & \text{if } j \neq k \\
1 & \text{if } j = k 
\end{cases} \]

where \( \delta_{jk} \) is Kronecker’s symbol.

**ERRORS**

The error when the Lagrange polynomial is used to approximate a continuous function \( f(x) \) is:
\[ E(x) = (x - x_0)(x - x_1)\ldots(x - x_n) \frac{f^{(n+1)}(\xi)}{(n+1)!} \text{ where } \xi \in [x_0, x_N] \]  

**NOTE**

- Lagrange polynomial applies for **evenly** and **unevenly** spaced points.
- If the points are evenly spaced then it reduces to a much simpler form.
EXAMPLE: Find the Lagrange polynomial that approximates the function $y = \cos(\pi x)$. We create the table

<table>
<thead>
<tr>
<th>$x_i$</th>
<th>0</th>
<th>0.5</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_i$</td>
<td>1</td>
<td>0.0</td>
<td>-1</td>
</tr>
</tbody>
</table>

The Lagrange coefficient polynomials are:

- $L_1(x) = \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} = \frac{(x - 0.5)(x - 1)}{(0 - 0.5)(0 - 1)} = 2x^2 - 3x + 1$,
- $L_2(x) = \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} = \frac{(x - 0)(x - 1)}{(0.5 - 0)(0.5 - 1)} = -4x(x - 1)$,
- $L_3(x) = \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} = \frac{(x - 0)(x - 0.5)}{(1 - 0)(1 - 0.5)} = 2x^2 - x$.

Thus,

$$P(x) = 1 \cdot (2x^2 - 3x + 1) - 0 \cdot 4x(x - 1) + (-1) \cdot (2x^2 - x) = -2x + 1$$
The error will be:

$$E(x) = x \cdot (x - 0.5) \cdot (x - 1) \frac{\pi^3 \sin(\pi \xi)}{3!}$$

e.g. for $x = 0.25$ is $E(0.25) \leq 0.24$. 
Newton Polynomial

Forward Newton-Gregory

\[ P_n(x_s) = f_0 + s \cdot \Delta f_0 + \frac{s(s-1)}{2!} \cdot \Delta^2 f_0 + \frac{s(s-1)(s-2)}{3!} \cdot \Delta^3 f_0 + \ldots \]

\[ = f_0 + \binom{s}{1} \cdot \Delta f_0 + \binom{s}{2} \cdot \Delta^2 f_0 + \binom{s}{3} \cdot \Delta^3 f_0 + \ldots \]

\[ = \sum_{i=0}^{n} \binom{s}{i} \cdot \Delta^i f_0 \] (4)

where \( x_s = x_0 + s \cdot h \) and

\[ \Delta f_i = f_{i+1} - f_i \] (5)

\[ \Delta^2 f_i = f_{i+2} - 2f_{i+1} + f_i \] (6)

\[ \Delta^3 f_i = f_{i+3} - 3f_{i+2} + 3f_{i+1} - f_i \] (7)

\[ \Delta^n f_i = f_{i+n} - nf_{i+n-1} + \frac{n(n-1)}{2!} f_{i+n-2} - \frac{n(n-1)(n-2)}{3!} f_{i+n-3} + \ldots \] (8)
Newton Polynomial

**ERROR** The error is the same with the Lagrange polynomial:

\[ E(x) = (x - x_0)(x - x_1)\ldots(x - x_n) \frac{f^{(n+1)}(\xi)}{(n + 1)!} \text{ where } \xi \in [x_0, x_N] \] (9)

<table>
<thead>
<tr>
<th>x</th>
<th>f(x)</th>
<th>Δf(x)</th>
<th>Δ²f(x)</th>
<th>Δ³f(x)</th>
<th>Δ⁴f(x)</th>
<th>Δ⁵f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.000</td>
<td>0.203</td>
<td>0.017</td>
<td>0.024</td>
<td>0.020</td>
<td>0.032</td>
</tr>
<tr>
<td>0.2</td>
<td>0.203</td>
<td>0.220</td>
<td>0.041</td>
<td>0.044</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>0.423</td>
<td>0.261</td>
<td>0.085</td>
<td>0.096</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>0.684</td>
<td>0.246</td>
<td>0.181</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>1.030</td>
<td>0.527</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1.557</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table:** Example of a difference matrix

**Backward Newton-Gregory**

\[ P_n(x) = f_0 + \binom{s}{1} \cdot \Delta f_{-1} + \binom{s + 1}{2} \cdot \Delta^2 f_{-2} + \ldots + \binom{s + n - 1}{n} \cdot \Delta^n f_{-n} \] (10)
<table>
<thead>
<tr>
<th>x</th>
<th>F(x)</th>
<th>Δf(x)</th>
<th>Δ²f(x)</th>
<th>Δ³f(x)</th>
<th>Δ⁴f(x)</th>
<th>Δ⁵f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>1.06894</td>
<td>0.11242</td>
<td>0.01183</td>
<td>0.00123</td>
<td>0.00015</td>
<td>0.000001</td>
</tr>
<tr>
<td>0.5</td>
<td>1.18136</td>
<td>0.12425</td>
<td>0.01306</td>
<td>0.00138</td>
<td>0.00014</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>1.30561</td>
<td>0.13731</td>
<td>0.01444</td>
<td>0.0152</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>1.44292</td>
<td>0.15175</td>
<td>0.01596</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.4</td>
<td>1.59467</td>
<td>0.16771</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.7</td>
<td>1.76238</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Newton-Gregory forward** with $x_0 = 0.5$:

$$P_3(x) = 1.18136 + 0.12425 s + 0.01306 \left( \frac{s}{2} \right) + 0.00138 \left( \frac{s}{3} \right)$$

$$= 1.18136 + 0.12425 s + 0.01306 (s - 1)/2 + 0.00138 s (s - 1) (s - 2)/6$$

$$= 0.9996 + 0.3354 x + 0.052 x^2 + 0.085 x^3$$

**Newton-Gregory backwards** with $x_0 = 1.1$:

$$P_3(x) = 1.44292 + 0.13731 s + 0.01306 \left( \frac{s + 1}{2} \right) + 0.00123 \left( \frac{s + 2}{3} \right)$$

$$= 1.44292 + 0.13731 s + 0.01306 s (s + 1)/2 + 0.00123 s (s + 1) (s + 2)/6$$

$$= 0.99996 + 0.33374 x + 0.05433 x^2 + 0.007593 x^3$$
This applies when we have information not only for the values of \( f(x) \) but also on its derivative \( f'(x) \)

\[
P_{2n-1}(x) = \sum_{i=1}^{n} A_i(x) f_i + \sum_{i=1}^{n} B_i(x) f'_i
\]

where

\[
A_i(x) = [1 - 2(x - x_i)L'_i(x_i)] \cdot [L_i(x)]^2
\]

\[
B_i(x) = (x - x_i) \cdot [L_i(x)]^2
\]

and \( L_i(x) \) are the Lagrange coefficients.

**ERROR**: the accuracy is similar to that of Lagrange polynomial of order \( 2n! \)

\[
y(x) - p(x) = \frac{y^{(2n+1)}(\xi)}{(2n + 1)!} [(x - x_1)(x - x_2) \ldots (x - x_n)]^2
\]
Hermite Polynomial: Example

Fit a Hermite polynomial to the data of the table:

<table>
<thead>
<tr>
<th>$k$</th>
<th>$x_k$</th>
<th>$y_k$</th>
<th>$y'_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

The Lagrange coefficients are:

\[
L_0(x) = \frac{x - x_1}{x_0 - x_1} = \frac{x - 4}{0 - 4} = -\frac{x - 4}{4} \quad L_1(x) = \frac{x - x_0}{x_1 - x_0} = \frac{x}{4}
\]

\[
L'_0(x) = \frac{1}{x_0 - x_1} = -\frac{1}{4} \quad L'_1(x) = \frac{1}{x_1 - x_0} = \frac{1}{4}
\]

Thus

\[
A_0(x) = \left[1 - 2 \cdot L'_0(x - x_0)\right] \cdot L_0^2 = \left[1 - 2 \cdot \left(-\frac{1}{4}\right) (x - 0)\right] \cdot \left(\frac{x - 4}{4}\right)^2
\]

\[
A_1(x) = \left[1 - 2 \cdot L'_1(x - x_1)\right] \cdot L_1^2 = \left[1 - 2 \cdot \frac{1}{4} (x - 4)\right] \cdot \left(\frac{x}{4}\right)^2 = \left(3 - \frac{x}{2}\right) \cdot \left(\frac{x}{4}\right)^2
\]

\[
B_0(x) = (x - 0) \cdot \left(\frac{x - 4}{4}\right)^2 = x \left(\frac{x - 4}{4}\right)^2 \quad B_1(x) = (x - 4) \cdot \left(\frac{x}{4}\right)^2
\]

And the Hermite polynomial is:

\[
P(x) = (6 - x) \frac{x^2}{16}.
\]
Taylor Polynomial

It is an alternative way of approximating functions with polynomials. In the previous two cases we found the polynomial $P(x)$ that gets the same value with a function $f(x)$ at $N$ points or the polynomial that agrees with a function and its derivative at $N$ points. Taylor polynomial has the same value $x_0$ with the function but agrees also up to the $N$th derivative with the given function. That is:

$$P^{(i)}(x_0) = f^{(i)}(x_0) \quad \text{and} \quad i = 0, 1, \ldots, n$$

and the Taylor polynomial has the well known form from Calculus

$$P(x) = \sum_{i=0}^{N} \frac{f^{(i)}(x)}{i!} (x - x_0)^i \quad (15)$$

**ERROR**: was also estimated in calculus

$$E_N(x) = (x - x_0)^{N+1} \frac{f^{(N+1)}(\xi)}{(N + 1)!} \quad (16)$$
We will show that for the calculation of \( e = 2.718281828459... \) with 13-digit approximation we need 15 terms of the Taylor expansion. All the derivatives at \( x = 1 \) are:

\[
y_0 = y_0^{(1)} = y_0^{(2)} = ... = y_0^{(n)} = 1
\]

thus

\[
p(x) = \sum_{i=1}^{n} \frac{1}{i!} x^i = 1 + x + \frac{x^2}{2} + ... + \frac{1}{n!} x^n
\]

and the error will be

\[
|E_n| = x^{n+1} \frac{e^\xi}{(n+1)!} = \frac{e^\xi}{16!} < \frac{3}{16!} = 1.433 \times 10^{-13}
\]
Interpolation with Cubic Splines

In some cases the typical polynomial approximation cannot smoothly fit certain sets of data. Consider the function

$$f(x) = \begin{cases} 
0 & -1 \leq x \leq -0.2 \\
1 - 5|x| & -0.2 < x < 0.2 \\
0 & 0.2 \leq x \leq 1.0 
\end{cases}$$

We can easily verify that we cannot fit the above data with any polynomial degree!

$$P(x) = 1 - 26x^2 + 25x^4$$

The answer to the problem is given by the spline fitting. That is we pass a set of cubic polynomials (cubic splines) through the points, using a new cubic for each interval. But we require that the slope and the curvature be the same for the pair of cubics that join at each point.
Interpolation with Cubic Splines

(a) Original function

(b) Fitted with quadratic

(c) Fitted with $P_3(x)$

(d) Fitted with $P_6(x)$

(e) Fitted with $P_8(x)$

Interpolation, Extrapolation & Polynomial Approximation
Interpolation with Cubic Splines

Let the cubic for the \( i \)th interval, which lies between the points \((x_i, y_i)\) and \((x_{i+1}, y_{i+1})\) has the form:

\[
y(x) = a_i \cdot (x - x_i)^3 + b_i \cdot (x - x_i)^2 + c_i \cdot (x - x_i) + d_i \quad (17)
\]

Since it fits at the two endpoints of the interval:

\[
y_i = a_i \cdot (x_i - x_i)^3 + b_i \cdot (x_i - x_i)^2 + c_i \cdot (x_i - x_i) + d_i = d_i
\]

\[
y_{i+1} = a_i \cdot (x_{i+1} - x_i)^3 + b_i \cdot (x_{i+1} - x_i)^2 + c_i \cdot (x_{i+1} - x_i) + d_i = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i
\]

where \( h_i = x_{i+1} - x_i \). We need the 1st and 2nd derivatives to relate the slopes and curvatures of the joining polynomials, by differentiation we get

\[
y'(x) = 3a_i \cdot (x - x_i)^2 + 2b_i \cdot (x - x_i) + c_i
\]

\[
y''(x) = 6a_i \cdot (x - x_i) + 2b_i
\]
The procedure is simplified if we write the equations in terms of the 2nd derivatives of the interpolating cubics. Let’s name $S_i$ the 2nd derivative at the point $(x_i, y_i)$ then we can easily get:

$$b_i = \frac{S_i}{2}, \quad a_i = \frac{S_{i+1} - S_i}{6h_i}$$

(18)

which means

$$y_{i+1} = \frac{S_{i+1} - S_i}{6h_i} h_i^3 + \frac{S_i}{2} h_i^2 + c_i h_i + y_i$$

and finally

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i S_i + h_i S_{i+1}}{6}$$

(19)

Now we invoke the condition that the slopes of the two cubics joining at $(x_i, y_i)$ are the same:

$$y'_i = 3a_i \cdot (x_i - x_i)^2 + 2b_i \cdot (x_i - x_i) + c_i = c_i \quad \text{for the } i \text{ interval}$$

$$y'_{i-1} = 3a_{i-1} \cdot (x_i - x_{i-1})^2 + 2b_{i-1} \cdot (x_i - x_{i-1}) + c_{i-1}$$

$$= 3a_{i-1} h_{i-1}^2 + 2b_{i-1} h_{i-1} + c_{i-1} \quad \text{for the } i - 1 \text{ interval}$$
By equating these and substituting $a$, $b$, $c$ and $d$ we get:

$$y_i' = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i S_i + h_i S_{i+1}}{6}$$

$$= 3 \left( \frac{S_i - S_{i-1}}{6h_{i-1}} \right) h_{i-1}^2 + 2 \frac{S_{i-1}}{2} h_{i-1} + \frac{y_i - y_{i-1}}{h_{i-1}} - \frac{2h_{i-1} S_{i-1} + h_{i-1} S_i}{6}$$

and by simplifying we get:

$$h_{i-1} S_{i-1} + 2 (h_{i-1} + h_i) S_i + h_i S_{i+1} = 6 \left( \frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}} \right) \quad (20)$$

If we have $n + 1$ points the above relation can be applied to the $n - 1$ internal points.
Thus we create a system of $n - 1$ equations for the $n + 1$ unknown $S_i$.
This system can be solved if we specify the values of $S_0$ and $S_n$. 

Interpolation, Extrapolation & Polynomial Approximation
The system of \( n - 1 \) equations with \( n + 1 \) unknown will be written as:

\[
\begin{pmatrix}
0 & 2(h_0 + h_1) & h_1 & 2(h_1 + h_2) & \cdots & h_2 & 2(h_2 + h_3) & h_3 & \cdots & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\
h_1 & 2(h_0 + h_1) & h_1 & 2(h_1 + h_2) & \cdots & h_2 & 2(h_2 + h_3) & h_3 & \cdots & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\
h_2 & 2(h_2 + h_3) & h_3 & \cdots & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\
& \vdots & & & & \cdots & & \vdots & & \vdots & \\
\end{pmatrix}
\begin{pmatrix}
S_0 \\
S_1 \\
S_2 \\
\vdots \\
S_{n-2} \\
S_{n-1} \\
S_n
\end{pmatrix}
= 6
\begin{pmatrix}
\frac{y_2 - y_1}{h_1} - \frac{y_1 - y_0}{h_0} \\
\frac{y_3 - y_2}{h_2} - \frac{y_2 - y_1}{h_1} \\
\vdots \\
\frac{y_n - y_{n-1}}{h_{n-1}} - \frac{y_{n-1} - y_{n-2}}{h_{n-2}} \\
\vdots
\end{pmatrix}
\equiv \vec{Y}
\]

From the solution of this linear systems we get the coefficients \( a_i, b_i, c_i \) and \( d_i \) via the relations:

\[
a_i = \frac{S_{i+1} - S_i}{6h_i}, \quad b_i = \frac{S_i}{2} \tag{21}
\]

\[
\frac{c_i}{h_i} = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_iS_i + h_iS_{i+1}}{6} \tag{22}
\]

\[
d_i = y_i \tag{23}
\]
• **Choice I** Take, $S_0 = 0$ and $S_n = 0$ this will lead to the solution of the following $(n - 1) \times (n - 1)$ linear system:

\[
\begin{pmatrix}
2(h_0 + h_1) & h_1 & & & \\
h_1 & 2(h_1 + h_2) & h_2 & & \\
h_2 & h_2 & 2(h_2 + h_3) & h_3 & \\
& & \ddots & \ddots & \ddots \\
& & & h_{n-2} & 2(h_{n-2} + h_{n-1})
\end{pmatrix}
\begin{pmatrix}
S_1 \\
S_2 \\
S_3 \\
\vdots \\
S_{n-1}
\end{pmatrix}
= \vec{Y}
\]

• **Choice II** Take, $S_0 = S_1$ and $S_n = S_{n-1}$ this will lead to the solution of the following $(n - 1) \times (n - 1)$ linear system:

\[
\begin{pmatrix}
3h_0 + 2h_1 & h_1 & & & \\
h_1 & 2(h_1 + h_2) & h_2 & & \\
h_2 & h_2 & 2(h_2 + h_3) & h_3 & \\
& & \ddots & \ddots & \ddots \\
& & & h_{n-2} & 2h_{n-2} + 3h_{n-1}
\end{pmatrix}
\begin{pmatrix}
S_1 \\
S_2 \\
S_3 \\
\vdots \\
S_{n-1}
\end{pmatrix}
= \vec{Y}
\]

Interpolation, Extrapolation & Polynomial Approximation
• **Choice III** Use linear extrapolation

\[
\frac{S_1 - S_0}{h_0} = \frac{S_2 - S_1}{h_1} \Rightarrow S_0 = \frac{(h_0 + h_1)S_1 - h_0 S_2}{h_1}
\]

\[
\frac{S_n - S_{n-1}}{h_{n-1}} = \frac{S_{n-1} - S_{n-2}}{h_{n-2}} \Rightarrow s_n = \frac{(h_{n-2} + h_{n-1})S_{n-1} - h_{n-1} S_{n-2}}{h_{n-2}}
\]

this will lead to the solution of the following \((n - 1) \times (n - 1)\) linear system:

\[
\begin{pmatrix}
\frac{(h_0 + h_1)(h_0 + 2h_1)}{h_1^2} & \frac{h_1^2 - h_0^2}{h_1} & \frac{h_2}{h_1} & \frac{h_2}{2(h_2 + h_3)} & \frac{h_3}{h_1} & \cdots \\
\frac{h_1}{h_1} & \frac{2(h_1 + h_2)}{h_2} & \frac{h_2}{h_2} & \frac{h_3}{2(h_2 + h_3)} & \frac{h_3}{h_2} & \cdots \\
\frac{h_1}{h_2} & \frac{h_2}{h_2} & \frac{h_2}{h_2} & \frac{h_3}{h_3} & \frac{h_3}{h_3} & \cdots \\
\frac{h_1}{h_3} & \frac{h_2}{h_3} & \frac{h_3}{h_3} & \frac{h_3}{h_3} & \frac{h_3}{h_3} & \cdots \\
\frac{h_1}{h_{n-2}} & \frac{h_2}{h_{n-2}} & \frac{h_3}{h_{n-2}} & \cdots & \frac{h_3}{h_{n-2}} & \cdots \\
\frac{h_1}{h_{n-2}} & \frac{h_2}{h_{n-2}} & \frac{h_3}{h_{n-2}} & \cdots & \frac{h_3}{h_{n-2}} & \cdots \\
\end{pmatrix}
\cdot
\begin{pmatrix}
S_1 \\
S_2 \\
S_3 \\
\cdots \\
S_{n-2} \\
S_{n-1}
\end{pmatrix} = \vec{Y}
\]
• **Choice IV** Force the slopes at the end points to assume certain values. If $f'(x_0) = A$ and $f'(x_n) = B$ then

\[
2h_0 S_0 + h_1 S_1 = 6 \left( \frac{y_1 - y_0}{h_0} - A \right)
\]

\[
h_{n-1} S_{n-1} + 2h_n S_n = 6 \left( B - \frac{y_n - y_{n-1}}{h_{n-1}} \right)
\]

\[
\begin{pmatrix}
2h_0 & h_1 \\
h_0 & 2(h_0 + h_1) & h_1 \\
h_1 & 2(h_1 + h_2) & h_2 \\
\vdots & & \ddots \\
h_{n-2} & 2h_{n-1} & h_{n-1}
\end{pmatrix}
\begin{pmatrix}
S_1 \\
S_2 \\
S_3 \\
\vdots \\
S_{n-1}
\end{pmatrix}
= \vec{\bar{Y}}
\]
Interpolation with Cubic Splines: Example

Fit a cubic spline in the data \((y = x^3 - 8)\):

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>-8</td>
<td>-7</td>
<td>0</td>
<td>19</td>
<td>56</td>
</tr>
</tbody>
</table>

Depending on the condition at the end we get the following solutions:

- **Condition I**: \(S_0 = 0, \ S_4 = 0\)

\[
\begin{pmatrix}
4 & 1 & 0 \\
1 & 4 & 1 \\
0 & 1 & 4
\end{pmatrix} \cdot \begin{pmatrix}
S_1 \\
S_2 \\
S_3
\end{pmatrix} = \begin{pmatrix}
36 \\
72 \\
108
\end{pmatrix} \Rightarrow S_1 = 6.4285, \ S_2 = 10.2857, \ S_3 = 24.4285
\]

- **Condition II**: \(S_0 = S_1, \ S_4 = S_3\)

\[
\begin{pmatrix}
5 & 1 & 0 \\
1 & 4 & 1 \\
0 & 1 & 5
\end{pmatrix} \cdot \begin{pmatrix}
S_1 \\
S_2 \\
S_3
\end{pmatrix} = \begin{pmatrix}
36 \\
72 \\
108
\end{pmatrix} \Rightarrow S_1 = S_0 = 4.8, \ S_2 = 12, \ S_3 = 19.2 = S_4
\]
• Condition III:

\[
\begin{pmatrix}
6 & 0 & 0 \\
1 & 4 & 1 \\
0 & 0 & 6
\end{pmatrix} \cdot \begin{pmatrix}
S_1 \\
S_2 \\
S_3
\end{pmatrix} = \begin{pmatrix}
36 \\
72 \\
108
\end{pmatrix} \Rightarrow \begin{align*}
S_0 &= 0 \\
S_1 &= 6 \\
S_2 &= 12 \\
S_3 &= 18 \\
S_4 &= 24
\end{align*}
\]

• Condition IV:

\[
\begin{pmatrix}
2 & 1 & 0 & 0 & 0 \\
1 & 4 & 1 & 0 & 0 \\
0 & 1 & 4 & 1 & 0 \\
0 & 0 & 1 & 4 & 1 \\
0 & 0 & 0 & 1 & 2
\end{pmatrix} \cdot \begin{pmatrix}
S_0 \\
S_1 \\
S_2 \\
S_3 \\
S_4
\end{pmatrix} = \begin{pmatrix}
6 \\
36 \\
72 \\
108 \\
66
\end{pmatrix} \Rightarrow \begin{align*}
S_0 &= 0 \\
S_1 &= 6 \\
S_2 &= 12 \\
S_3 &= 18 \\
S_4 &= 24
\end{align*}
\]
1. The following data are from astronomical observations and represent variations of the apparent magnitude of a type of variable stars called Cepheids.

<table>
<thead>
<tr>
<th>Time</th>
<th>0.0</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.6</th>
<th>0.7</th>
<th>0.8</th>
<th>1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Magnitude</td>
<td>0.302</td>
<td>0.185</td>
<td>0.106</td>
<td>0.093</td>
<td>0.24</td>
<td>0.579</td>
<td>0.561</td>
<td>0.468</td>
<td>0.302</td>
</tr>
</tbody>
</table>

Use splines to create a new table for the apparent magnitude for intervals of time of 0.5.

2. From the following table find the acceleration of gravity at Tübingen (48° 31’) and the distance between two points with angular separation of 1’ of a degree.

<table>
<thead>
<tr>
<th>Latitude</th>
<th>Length of 1’ of arc on the parallel</th>
<th>local acceleration of gravity $g$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0°</td>
<td>1855.4 m</td>
<td>9.7805 m/sec^2</td>
</tr>
<tr>
<td>15°</td>
<td>1792.0 m</td>
<td>9.7839 m/sec^2</td>
</tr>
<tr>
<td>30°</td>
<td>1608.2 m</td>
<td>9.7934 m/sec^2</td>
</tr>
<tr>
<td>45°</td>
<td>1314.2 m</td>
<td>9.8063 m/sec^2</td>
</tr>
<tr>
<td>60°</td>
<td>930.0 m</td>
<td>9.8192 m/sec^2</td>
</tr>
<tr>
<td>75°</td>
<td>481.7 m</td>
<td>9.8287 m/sec^2</td>
</tr>
<tr>
<td>90°</td>
<td>0.0 m</td>
<td>9.8322 m/sec^2</td>
</tr>
</tbody>
</table>
Rational function approximations

Here we introduce the notion of **rational approximations for functions**. We will constrain our discussion to the so called **Padé approximation**. A rational approximation of \( f(x) \) on \([a, b]\), is the quotient of two polynomials \( P_n(x) \) and \( Q_m(x) \) with degrees \( n \) and \( m \)

\[
f(x) = R_N(x) \equiv \frac{P_n(x)}{Q_m(x)} = \frac{a_0 + a_1x + a_2x^2 + \ldots + a_nx^n}{1 + b_1x + b_2x^2 + \ldots + b_mx^m}, \quad N = n + m
\]

i.e. there are \( N + 1 = n + m + 1 \) constants to be determined.

The method of Padé requires that \( f(x) \) and its derivatives are continuous at \( x = 0 \). This choice makes the manipulation simpler and a change of variable can be used to shift, if needed, the calculations over to an interval that contains zero.

We begin with the Maclaurin series for \( f(x) \) (up to the term \( x^N \)), this can be written as :

\[
f(x) \approx c_0 + c_1x + \ldots + c_Nx^N
\]

where \( c_i = f^{(i)}(0)/(i!) \).

\(^1\)Sometimes we write \( R_N(x) \equiv R_{n,m}(x) \).
Then we create the difference

\[
f(x) - R_N(x) \approx \left( c_0 + c_1 x + \ldots + c_N x^N \right) - \frac{a_0 + a_1 x + \ldots + a_n x^n}{1 + b_1 x + \ldots + b_m x^m}
\]

\[
= \left( c_0 + c_1 x + \ldots + c_N x^N \right) \frac{(1 + b_1 x + \ldots + b_m x^m) - (a_0 + a_1 x + \ldots + a_n x^n)}{1 + b_1 x + \ldots + b_m x^m}
\]

If \( f(0) = R_N(0) \) then \( c_0 - a_0 = 0 \).

In the same way, in order for the first \( N \) derivatives of \( f(x) \) and \( R_N(x) \) to be equal at \( x = 0 \) the coefficients of the powers of \( x \) up to \( x^N \) in the numerator must be zero also.
This gives additionally \( N \) equations for the \( a's \) and \( b's \)

\[
\begin{align*}
    b_1 c_0 + c_1 - a_1 &= 0 \\
    b_2 c_0 + b_1 c_1 + c_2 - a_2 &= 0 \\
    b_3 c_0 + b_2 c_1 + b_1 c_2 + c_3 - a_3 &= 0 \\
    \vdots \\
    b_m c_{n-m} + b_{m-1} c_{n-m+1} + \ldots + c_n - a_n &= 0 \\
    b_m c_{n-m+1} + b_{m-1} c_{n-m+2} + \ldots + c_{n+1} &= 0 \\
    b_m c_{n-m+2} + b_{m-1} c_{n-m+3} + \ldots + c_{n+2} &= 0 \\
    \vdots \\
    b_m c_{N-m} + b_{m-1} c_{N-m+1} + \ldots + c_N &= 0
\end{align*}
\]  

(24)

Notice that in each of the above equations, the sum of subscripts on the factors of each product is the same, and is equal to the exponent of the \( x \)-term in the numerator.
Padé approximation: Example

For the $R_9(x)$ or $R_{5,4}(x)$ Padé approximation for the function $\tan^{-1}(x)$ we calculate the Maclaurin series of $\tan^{-1}(x)$:

$$\tan^{-1}(x) = x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \frac{1}{7}x^7 + \frac{1}{9}x^9$$

and then

$$f(x) - R_9(x) = \frac{(x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \frac{1}{7}x^7 + \frac{1}{9}x^9) \left(1 + b_1x + b_2x^2 + b_3x^3 + b_4x^4\right) - \left(a_0 + a_1x + a_2x^2 + \ldots + a_5x^5\right)}{1 + b_1x + b_2x^2 + b_3x^3 + b_4x^4}$$

and the coefficients will be found by the following system of equations:

\begin{align*}
    a_0 &= 0, \quad a_1 = 1, \quad a_2 = b_1, \quad a_3 = -\frac{1}{3} + b_2, \quad a_4 = -\frac{1}{3}b_1 + b_3, \quad a_5 = \frac{1}{5} - \frac{1}{3}b_2 + b_4 \\
    \frac{1}{5}b_1 - \frac{1}{2}b_3 &= 0, \quad -\frac{1}{7} + \frac{1}{5}b_2 - \frac{1}{3}b_4 = 0, \quad -\frac{1}{7}b_1 + \frac{1}{5}b_3 = 0, \quad \frac{1}{9} - \frac{1}{7}b_2 + \frac{1}{5}b_4 = 0 \quad (25)
\end{align*}

from which we get

$$a_0 = 0, \quad a_1 = 1, \quad a_2 = 0, \quad a_3 = \frac{7}{9}, \quad a_4 = 0, \quad a_5 = \frac{64}{945}, \quad b_1 = 0, \quad b_2 = \frac{10}{9}, \quad b_3 = 0, \quad b_4 = \frac{5}{21}.$$ 

$$\tan^{-1}x \approx R_9(x) = \frac{x + \frac{7}{9}x^3 + \frac{64}{945}x^5}{1 + \frac{10}{9}x^2 + \frac{5}{21}x^4}$$

For $x = 1$, exact 0.7854, $R_9(1) = 0.78558$ while from the Maclaurin series we get 0.8349!
Rational approximation for sets of data

If instead of the analytic form of a function \( f(x) \) we have a set of \( k \) points \((x_i, f(x_i))\) in order to find a rational function \( R_N(x) \) such that for every \( x_i \) we will get \( f(x_i) = R_N(x_i) \) i.e.

\[
R_N(x_i) = \frac{a_0 + a_1x_i + a_2x_i^2 + \cdots + a_nx_i^n}{1 + b_1x_i + b_2x_i^2 + \cdots + b_mx_i^m} = f(x_i)
\]

we will follow the approach used in constructing the approximate polynomial. In other words the problem will be solved by finding the solution of the following system of \( k \geq m + n + 1 \) equations:

\[
\begin{align*}
a_0 + a_1x_1 + \cdots + a_nx_1^n &- (f_1x_1)\ b_1 - \cdots - (f_1x_1^m)\ b_m = f_1 \\
\vdots \\
a_0 + a_1x_i + \cdots + a_nx_i^n &- (f_ix_i)\ b_1 - \cdots - (f_ix_i^m)\ b_m = f_i \\
\vdots \\
a_0 + a_1x_k + \cdots + a_nx_k^n &- (f_kx_k)\ b_1 - \cdots - (f_kx_k^m)\ b_m = f_k 
\end{align*}
\]

i.e. we get \( k \) equations for the \( k \) unknowns \( a_0, a_1, \ldots, a_n \) and \( b_1, b_2, \ldots, b_m \).
We will find the rational function approximations for the following set of data \((-1,1), (0,2)\) and \((1,-1)\).

It is obvious that the sum of degrees of the polynomials in the nominator and denominator must be \((n + m + 1 \leq 3)\). Thus we can write:

\[
R_{1,1}(x) = \frac{a_0 + a_1 x}{1 + b_1 x}
\]

which leads to the following system

\[
\begin{align*}
  a_0 + (-1) a_1 - (-1) b_1 &= 1 \\
  a_0 + 0 \cdot a_1 - 0 \cdot b_1 &= 2 \\
  a_0 + 1 \cdot a_1 - (-1) b_1 &= -1
\end{align*}
\]

\[
\implies \begin{cases}
  a_0 = 2 \\
  a_1 = -1 \\
  b_1 = -2
\end{cases}
\]

and the rational function will be:

\[
R_{1,1}(x) = \frac{2 - x}{1 - 2x}.
\]

Alternatively, one may derive the following rational function:

\[
R_{0,1}(x) = \frac{a_0}{1 + b_1 x + b_2 x} \Rightarrow R(x) = \frac{2}{1 - 2x - x^2}
\]
Find the Padé approximation $R_{3,3}(x)$ for the function $y = e^x$. Compare with the Maclaurin series for $x = 1$.

Find the Padé approximation $R_{3,5}(x)$ for the functions $y = \cos(x)$ and $y = \sin(x)$. Compare with the Maclaurin series for $x = 1$.

Find the Padé approximation $R_{4,6}(x)$ for the function $y = 1/x \sin(x)$. Compare with the Maclaurin series for $x = 1$.

Find the rational approximation for the following set of points:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.83</td>
<td>1.06</td>
<td>1.25</td>
<td>4.15</td>
</tr>
</tbody>
</table>