LECTURE 4

NEWTON FORWARD INTERPOLATION ON EQUISPACED POINTS

• Lagrange Interpolation has a number of disadvantages
  
  • The amount of computation required is large
  
  • Interpolation for additional values of \( x \) requires the same amount of effort as the first value (i.e. no part of the previous calculation can be used)
  
  • When the number of interpolation points are changed (increased/decreased), the results of the previous computations can not be used
  
  • Error estimation is difficult (at least may not be convenient)

• Use Newton Interpolation which is based on developing difference tables for a given set of data points

• The \( N^{th} \) degree interpolating polynomial obtained by fitting \( N + 1 \) data points will be identical to that obtained using Lagrange formulae!

  • Newton interpolation is simply another technique for obtaining the same interpolating polynomial as was obtained using the Lagrange formulae
Forward Difference Tables

• We assume equi-spaced points (not necessary)

\[ \begin{align*}
\Delta^0 f_i &\equiv f_i & \text{(Zero\textsuperscript{th} order forward difference)} \\
\Delta f_i &\equiv f_{i+1} - f_i & \text{(First order forward difference)}
\end{align*} \]
\[ \Delta^2 f_i \equiv \Delta f_{i+1} - \Delta f_i \quad \text{(Second order forward difference)} \]
\[ \Delta^2 f_i = (f_{i+2} - f_{i+1}) - (f_{i+1} - f_i) \]
\[ \Delta^2 f_i = f_{i+2} - 2f_{i+1} + f_i \]

\[ \Delta^3 f_i \equiv \Delta^2 f_{i+1} - \Delta^2 f_i \quad \text{(Third order forward difference)} \]
\[ \Delta^3 f_i = (f_{i+3} - 2f_{i+2} + f_{i+1}) - (f_{i+2} - 2f_{i+1} + f_i) \]
\[ \Delta^3 f_i = f_{i+3} - 3f_{i+2} + 3f_{i+1} - f_i \]

\[ \Delta^k f_i = \Delta^{k-1} f_{i+1} - \Delta^{k-1} f_i \quad \text{\((k^{th}\text{ order forward difference})\)} \]

- Typically we set up a difference table
Note that to compute higher order differences in the tables, we take forward differences of previous order differences instead of using expanded formulae.

The order of the differences that can be computed depends on how many total data points, $x_0, \ldots, x_N$, are available.

- **$N + 1$ data points can develop up to $N^{th}$ order forward differences**
Example 1

- Develop a forward difference table for the data given

<table>
<thead>
<tr>
<th>$i$</th>
<th>$x_i$</th>
<th>$f_i$</th>
<th>$\Delta f_i$</th>
<th>$\Delta^2 f_i$</th>
<th>$\Delta^3 f_i$</th>
<th>$\Delta^4 f_i$</th>
<th>$\Delta^5 f_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>-7</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>-3</td>
<td>9</td>
<td>10</td>
<td>8</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>6</td>
<td>19</td>
<td>18</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>25</td>
<td>37</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>62</td>
<td>67</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>129</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Deriving Newton Forward Interpolation on Equi-spaced Points

• Summary of Steps
  - Step 1: Develop a general Taylor series expansion for $f(x)$ about $x_0$.
  - Step 2: Express the various order forward differences at $x_0$ in terms of $f(x)$ and its derivatives evaluated at $x_0$. This will allow us to express the actual derivatives evaluated at $x_0$ in terms of forward differences.
  - Step 3: Using the general Taylor series expansion developed in Step 1, sequentially substitute in for the derivatives evaluated at $x_0$ in terms of forward differences (i.e. substitute in the expressions developed in Step 2).
**Step 1**

- The Taylor series expansion for \( f(x) \) about \( x_o \) is:

\[
f(x) = f(x_o) + (x - x_o) \frac{df}{dx} \bigg|_{x=x_o} + \frac{1}{2!} (x - x_o)^2 \frac{d^2f}{dx^2} \bigg|_{x=x_o} + \frac{1}{3!} (x - x_o)^3 \frac{d^3f}{dx^3} \bigg|_{x=x_o} + O(x - x_o)^4
\]

\[\Rightarrow\]

\[
f(x) = f_o + (x - x_o)f_o^{(1)} + \frac{1}{2!} (x - x_o)^2 f_o^{(2)} + \frac{1}{3!} (x - x_o)^3 f_o^{(3)} + O(x - x_o)^4
\]

**Step 2a**

- Express first order forward difference in terms of \( f_o, f_o^{(1)}, \ldots \)

\[
\Delta f_o \equiv f_1 - f_o
\]

- However since \( f_1 = f(x_1) \), we can use the Taylor series given in **Step 1** to express \( f_1 \) in terms of \( f_o \) and its derivatives:

\[
f_1 = f_o + (x_1 - x_o)f_o^{(1)} + \frac{1}{2!} (x_1 - x_o)^2 f_o^{(2)} + \frac{1}{3!} (x_1 - x_o)^3 f_o^{(3)} + O(x_1 - x_o)^4
\]
• We note that the spacing between data points is \( h \equiv x_1 - x_o \):

\[
f_1 = f_o + hf_o^{(1)} + \frac{1}{2!}h^2f_o^{(2)} + \frac{1}{3!}h^3f_o^{(3)} + O(h)^4
\]

• Now, substitute in for \( f_1 \) into the definition of the first order forward differences

\[
\Delta f_o = f_o + hf_o^{(1)} + \frac{1}{2!}h^2f_o^{(2)} + \frac{1}{3!}h^3f_o^{(3)} + O(h)^4 - f_o
\]

\[
\Rightarrow
\]

\[
f_o^{(1)} = \frac{\Delta f_o}{h} - \frac{1}{2!}hf_o^{(2)} - \frac{1}{3!}h^2f_o^{(3)} - O(h)^3
\]

• Note that the first order forward difference divided by \( h \) is in fact an approximation to the first derivative to \( O(h) \). However, we will use all the terms given in this sequence.
Step 2b

• Express second order forward difference in terms of \( f_o, f_o^{(1)}, \ldots \)

\[
\Delta^2 f_o \equiv f_2 - 2f_1 + f_o
\]

• We note that \( f_1 = f(x_1) \) was developed in T.S. form in Step 2a.

• For \( f_2 = f(x_2) \) we use the T.S. given in Step 1 to express \( f_2 \) in terms of \( f_o \) and derivatives of \( f \) evaluated at \( x_o \)

\[
f_2 = f_o + (x_2 - x_o)f_o^{(1)} + \frac{1}{2!}(x_2 - x_o)^2f_o^{(2)} + \frac{1}{3!}(x_2 - x_o)^3f_o^{(3)} + O(x_2 - x_o)^4
\]

• We note that \( x_2 - x_o = 2h \)

\[
f_2 = f_o + 2hf_o^{(1)} + \frac{4}{2!}h^2f_o^{(2)} + \frac{8}{3!}h^3f_o^{(3)} + O(h)^4
\]
• Now substitute in for \( f_2 \) and \( f_1 \) into the definition of the second order forward difference operator

\[
\Delta^2 f_o = f_o + 2h f_o^{(1)} + 2h^2 f_o^{(2)} + \frac{4}{3} h^3 f_o^{(3)} + O(h)^4 - 2f_o - 2h f_o^{(1)} - h^2 f_o^{(2)} - \frac{1}{3} h^3 f_o^{(3)} + O(h)^4 + f_o
\]

\[
\Rightarrow
\]

\[
f_o^{(2)} = \frac{\Delta^2 f_o}{h^2} - h f_o^{(3)} + O(h)^2
\]

• Note that the second order forward difference divided by \( h^2 \) is in fact an approximation to \( f_o^{(2)} \) to \( O(h) \). However, we will use all terms in the expression.
Step 2c

- Express the third order forward difference in terms of $f_o$, $f_o^{(1)}$, ...

$$\Delta^3 f_o \equiv f_3 - 3f_2 + 3f_1 - f_o$$

- We already developed expressions for $f_2$ and $f_1$.

- Develop an expression for $f_3 = f(x_3)$ using the T.S. in Step 1

$$f_3 = f_o + (x_3 - x_o)f_o^{(1)} + \frac{1}{2!}(x_3 - x_o)^2f_o^{(2)} + \frac{1}{3!}(x_3 - x_o)^3f_o^{(3)} + O(x_3 - x_o)^4$$

- Noting that $x_3 - x_o = 3h$

$$f_3 = f_o + 3hf_o^{(1)} + \frac{9}{2}h^2f_o^{(2)} + \frac{9}{2}h^3f_o^{(3)} + O(h)^4$$
Substituting in for $f_3$, $f_2$, and $f_1$ into the definition of the third order forward difference formula.

$$
\Delta^3 f_o = f_o + 3hf_o^{(1)} + \frac{9}{2}h^2f_o^{(2)} + \frac{9}{2}h^3f_o^{(3)} + O(h)^4 - 3f_o - 6hf_o^{(1)} - \frac{12}{2}h^2f_o^{(2)} - \frac{24}{3!}h^3f_o^{(3)} + O(h)^4
$$

$$
+ 3f_o + 3hf_o^{(1)} + \frac{3}{2}h^2f_o^{(2)} + \frac{3}{3!}h^3f_o^{(3)} + O(h)^4 - f_o
$$

$$
\Rightarrow
$$

$$
\Delta^3 f_o = h^3f_o^{(3)} + O(h)^4
$$

$$
\Rightarrow
$$

$$
f_o^{(3)} = \frac{\Delta^3 f_o}{h^3} + O(h)
$$

The third order forward difference divided by $h^3$ is an $O(h)$ approximation to $f_o^{(3)}$.
Step 3a

- Consider the general T.S. expansion presented in Step 1 to define $f(x)$ and substitute in for $f_o^{(1)}$ using the result in Step 2a.

- Note that now we are not evaluating the T.S. at a data point but at any $x$

\[
f(x) = f_o + (x - x_o) \left[ \frac{\Delta f_o}{h} - \frac{1}{2!} h f_o^{(2)} - \frac{1}{3!} h^2 f_o^{(3)} - O(h)^3 \right] + \frac{1}{2!} (x - x_o)^2 f_o^{(2)} + \frac{1}{3!} (x - x_o)^3 f_o^{(3)} + O(x - x_o)^4
\]

\[
\Rightarrow
\]

\[
f(x) = f_o + \frac{x - x_o}{h} \Delta f_o + \frac{1}{2!} \left[ - (x - x_o) h + (x - x_o)^2 \right] f_o^{(2)} + \frac{1}{3!} \left[ - (x - x_o) h^2 + (x - x_o)^3 \right] f_o^{(3)} + O(h)^4
\]
Step 3b

• Substitute in for \( f_o^{(2)} \) using the expression developed in Step 2b.

\[
f(x) = f_o + \frac{x - x_o}{h} \Delta f_o + \frac{1}{2!} \left[ - (x - x_o) h + (x - x_o)^2 \right] \left[ \frac{\Delta^2 f_o}{h^2} - h f_o^{(3)} + O(h)^2 \right]
\]

\[
+ \frac{1}{3!} \left[ -(x - x_o) h^2 + (x - x_o)^3 \right] f_o^{(3)} + O(h)^4
\]

\[
\Rightarrow
\]

\[
f(x) = f_o + \frac{x - x_o}{h} \Delta f_o + \frac{1}{2!} \left[ - \frac{(x - x_o)}{h} + \frac{(x - x_o)^2}{h^2} \right] \Delta^2 f_o
\]

\[
+ \frac{1}{3!} \left[ 2(x - x_o) h^2 + (x - x_o)^3 - 3(x - x_o)^2 h \right] f_o^{(3)} + O(h)^4
\]
**Step 3c**

- Substitute in for $f_o^{(3)}$ from *Step 2c*

\[
f(x) = f_o + \frac{(x-x_o)}{h} \Delta f_o + \frac{1}{2!} \left[ - \frac{(x-x_o)}{h} + \frac{(x-x_o)^2}{h^2} \right] \Delta^2 f_o
\]

\[
+ \frac{1}{3!} \left[ 2(x-x_o)h^2 + (x-x_o)^3 - 3(x-x_o)^2 h \right] \frac{\Delta^3 f_o}{h^3} + O(h^4)
\]

- Re-arranging the terms in brackets:

\[
f(x) = f_o + \frac{(x-x_o)}{h} \Delta f_o + \frac{1}{2!} (x-x_o) \left[ - h + (x-x_o) \right] \frac{\Delta^2 f_o}{h^2}
\]

\[
+ \frac{1}{3!} (x-x_o) \left[ 2h^2 + (x-x_o)^2 - 3(x-x_o)h \right] \frac{\Delta^3 f_o}{h^3} + O(h^4) + HOT
\]

\[\Rightarrow\]

p. 4.15
\[ f(x) = f_o + (x - x_o) \frac{\Delta f_o}{h} + \frac{1}{2!} (x - x_o)(x - (x_o + h)) \frac{\Delta^2 f_o}{h^2} \]

\[ + \frac{1}{3!} (x - x_o)[(x - (x_o + h))(x - (x_o + 2h)) \frac{\Delta^3 f_o}{h^3} + O(h^4) + HOT \]

- Also considering higher order terms and noting that \( x_o + h = x_1 \), \( x_o + 2h = x_2 \) and \( f(x) = g(x) + e(x) \)

\[ g(x) = f_o + (x - x_o) \frac{\Delta f_o}{h} + \frac{1}{2!} (x - x_o)(x - x_1) \frac{\Delta^2 f_o}{h^2} + \frac{1}{3!} (x - x_o)(x - x_1)(x - x_2) \frac{\Delta^3 f_o}{h^3} \]

\[ + \ldots + \frac{1}{N!} (x - x_o)(x - x_1)(x - x_2) \ldots (x - x_{N-1}) \frac{\Delta^N f_o}{h^N} \]

- This is the \( N^{th} \) degree polynomial approximation to \( N + 1 \) data points and is identical to that derived for Lagrange interpolation or Power series (only the form in which it is presented is different).
Note that the $N + 1$ data point are exactly fit by $g(x)$

$$g(x_o) = f_o$$

$$g(x_1) = f_o + (x_1 - x_o)\frac{f_1 - f_o}{h} = f_o + h\left(\frac{f_1 - f_o}{h}\right) = f_1$$

$$g(x_2) = f_o + (x_2 - x_o)\left(\frac{f_1 - f_o}{h}\right) + \frac{1}{2}(x_2 - x_o)(x_2 - x_1)\frac{1}{h^2}(f_2 - 2f_1 + f_o)$$

$$\Rightarrow$$

$$g(x_2) = f_o + \frac{2h}{h}(f_1 - f_o) + \frac{(2h)h}{2h^2}(f_2 - 2f_1 + f_o)$$

$$\Rightarrow$$

$$g(x_2) = f_o + 2f_1 - 2f_o + f_2 - 2f_1 + f_o = f_2$$

In general

$$g(x_i) = f_i \quad i = 0, N$$
• It can be readily shown that the error at any $x$ is: (by carrying through error terms in the T.S.)

$$e(x) = f(x) - g(x) = \frac{(x-x_0)(x-x_1)\ldots(x-x_N)}{(N+1)!} f^{(N+1)}(\xi)$$

$x_o < \xi < x_N$

• This error function is identical to that for Lagrange Interpolation (since the polynomial approximation is the same).

• However we note that $f^{N+1}(x)$ can be approximated as (can be shown by T.S.)

$$f^{(N+1)}(x) \approx \frac{\Delta^{N+1} f_o}{h^{N+1}}$$

• In fact if $f^{(N+1)}(x)$ does not vary dramatically over the interval

$$f^{(N+1)}(\xi) \approx \frac{\Delta^{N+1} f_o}{h^{N+1}}$$
• Thus the error can be estimated as

\[ e(x) \equiv \frac{(x - x_o)(x - x_1)\ldots(x - x_N)}{(N + 1)!} \frac{\Delta^{N+1} f_o}{h^{N+1}} \]

• Notes

  • Approximation for \( e(x) \) is equal to the term that would follow the last term in the \( N^{th} \) degree polynomial series for \( g(x) \)

  • If we have \( N + 2 \) data points available and develop an \( N^{th} \) degree polynomial approximation with \( N + 1 \) data points, we can then easily estimate \( e(x) \). This was not as simple for Lagrange polynomials since you then needed to compute the finite difference approximation to the derivative in the error function.

  • If the exact function \( f(x) \) is a polynomial of degree \( M \leq N \), then \( g(x) \) will be an (almost) exact representation of \( f(x) \) (with small roundoff errors).

  • Newton Interpolation is much more efficient to implement than Lagrange Interpolation. If you develop a difference table \textit{once}, you can

    • Develop various order interpolation functions very quickly (since each higher order term only involves one more product)

    • Obtain error estimates very quickly
Example 2

• For the data and forward difference table presented in Example 1.
  
  • (a) Develop \( g(x) \) using 3 points \((x_o = 2, x_1 = 4, x_2 = 6)\) and estimate \( e(x) \)
  
  • (b) Develop \( g(x) \) using 4 points \((x_o = 2, x_1 = 4, x_2 = 6, x_3 = 8)\) and estimate \( e(x) \)
  
  • (c) Develop \( g(x) \) using 3 different points \((x_o = 6, x_1 = 8, x_2 = 10)\)

(Part a)

• 3 data points \( \Rightarrow N = 2 \)

\[
g_3(x) = f_o + (x - x_o) \frac{\Delta f_o}{h} + \frac{1}{2!} (x - x_o)(x - x_1) \frac{\Delta^2 f_o}{h^2}
\]

with

\[
x_o = 2 \quad x_1 = 4 \quad x_2 = 6 \quad \text{and} \quad h = 2
\]

• Note that the “3” designation in \( g_3(x) \) indicates \( N+1=3 \) data points
• $f_o$, $\Delta f_o$ and $\Delta^2 f_o$ are obtained by simply picking values off of the difference table (across the row $i = 0$)

\[
f_o = -7 \quad \Delta f_o = 4 \quad \Delta^2 f_o = 5
\]

\[
g_3(x) = -7 + (x - 2)\frac{4}{2} + \frac{1}{2!}(x - 2)(x - 4)\frac{5}{4}
\]

• The error can be estimated as:

\[
e_3(x) = \frac{(x - x_o)(x - x_1)(x - x_2)}{3!} \frac{1}{h^3} \Delta^3 f_o
\]

• Simply substitute in for $x_o$, $x_1$, $x_2$, $h$ and pick off $\Delta^3 f_o = 5$ from the table in Example 1

\[
e_3(x) = \frac{(x - 2)(x - 4)(x - 6)}{6} \cdot \frac{1}{2^3} \cdot 5 = \frac{(x - 2)(x - 4)(x - 6)}{48}
\]
(Part b)

• 4 data points \(x_o = 2, x_1 = 4, x_2 = 6, x_3 = 8 \Rightarrow N = 3\)

• Simply add the next term to the series for \(g_3(x)\) in Part a:

\[
g_4(x) = g_3(x) + \frac{1}{3!}(x-x_o)(x-x_1)(x-x_2)\frac{\Delta^3 f_o}{h^3}
\]

• We note that the term we are adding to \(g_3(x)\) is actually \(e_3(x)\)

• Pick off \(\Delta^3 f_o = 5\) from the table in Example 1 and substitute in

\[
g_4(x) = g_3(x) + \frac{1}{3!}(x-2)(x-4)(x-6) \cdot \frac{5}{8}
\]

• The error is estimated as

\[
e_4(x) = \frac{(x-x_o)(x-x_1)(x-x_2)(x-x_3)}{4!} \frac{1}{h^4} \Delta^4 f_o \quad \Rightarrow
\]

\[
e_4(x) = (x-2)(x-4)(x-6)(x-8) \frac{3}{384}
\]
(Part c)

• 3 data points \( x_o = 6 \), \( x_1 = 8 \) and \( x_3 = 10 \) \( \Rightarrow \) \( N=2 \)

• We must shift \( i \) in the table such that \( x_o = 6 \) etc.

\[
g_{3/s}(x) = f_o + (x-x_o)\frac{\Delta f_o}{h} + \frac{1}{2!}(x-x_o)(x-x_1)\frac{\Delta^2 f_o}{h^2}
\]

• Pick off \( f_o \), \( \Delta f_o \) and \( \Delta^2 f_o \) from the same difference table with a shifted index

\[
f_o = 6 \, , \quad \Delta f_o = 19 \, , \quad \Delta^2 f_o = 18
\]

• Substituting

\[
g_{3/s}(x) = 6 + (x-6)\frac{19}{2} + \frac{1}{2!}(x-6)(x-8)\frac{18}{2^2}
\]
Newton Backward Interpolation

• Newton backward interpolation is essentially the same as Newton forward interpolation except that backward differences are used.

• Backward differences are defined as:

\[ \nabla^0 f_i \equiv f_i \quad \text{Zero}^{th} \text{ order backward difference} \]

\[ \nabla f_i = f_i - f_{i-1} \quad \text{First order backward difference} \]

\[ \nabla^2 f_i = \nabla f_i - \nabla f_{i-1} \quad \text{Second order backward difference} \]

\[ \nabla^k f_i = \nabla^{k-1} f_i - \nabla^{k-1} f_{i-1} \quad k^{th} \text{ order backward difference} \]
• For \( N + 1 \) data points which are fitted with an \( N^{th} \) degree polynomial

\[
g(x) = f_N + (x - x_N)\frac{\nabla f_N}{h} + \frac{1}{2!}(x - x_N)(x - x_{N-1})\frac{\nabla^2 f_N}{h^2} \]

\[
+ \frac{1}{3!}(x - x_N)(x - x_{N-1})(x - x_{N-2})\frac{\nabla^3 f_N}{h^3} \]

\[
+ \frac{1}{N!}(x - x_N)(x - x_{N-1})... (x - x_1)\frac{\nabla^N f_N}{h^N} \]

• Note that we are really expanding about the right most point to the left. Therefore we must develop \( f_N, \nabla f_N \) etc. in the difference table
Newton Interpolation on Non-uniformly Spaced Data Points

• Newton interpolation can be readily extended to deal with non-uniformly spaced data points

• The difference table for non-uniformly spaced nodes is developed and an appropriate interpolation formula is developed and used
SUMMARY OF LECTURE 4

• Newton formulae can be obtained by manipulating Taylor series

• Newton interpolating function is related to easily computed forward/backward differences

• Error is readily established and estimated from the difference table (as long as you have one more data point than used in interpolation)

• Newton interpolation through \( N + 1 \) data points gives the same \( N^{th} \) degree polynomial as Lagrange interpolation

• Newton interpolation is more efficient than Lagrange interpolation and is easily implemented