LECTURE 8

NUMERICAL DIFFERENTIATION FORMULAE BY INTERPOLATING POLYNOMIALS

Relationship Between Polynomials and Finite Difference Derivative Approximations

- We noted that N^{th} degree accurate Finite Difference (FD) expressions for first derivatives have an associated error

$$E \cong h^N \frac{d^{N+1} f}{dx^{N+1}}$$

- If $f(x)$ is an N^{th} degree polynomial then,

$$\frac{d^{N+1} f}{dx^{N+1}} = 0$$

and the FD approximation to the first derivative is exact!

- Thus if we know that a FD approximation to a polynomial function is exact, we can derive the form of that polynomial by integrating the previous equation.

$$f(x) \cong a_1 x^N + a_2 x^{N-1} + \ldots + a_{N+1}$$

- This implies that a distinct relationship exists between polynomials and FD expressions for derivatives (different relationships for higher order derivatives).
- We can in fact develop FD approximations from interpolating polynomials

Developing Finite Difference Formulae by Differentiating Interpolating Polynomials

Concept

- **The approximation for the p^{th} derivative of some function $f(x)$ can be found by taking the p^{th} derivative of a polynomial approximation, $g(x)$, of the function $f(x)$.**

Procedure

- Establish a polynomial approximation $g(x)$ of degree N such that $N \geq p$
- $g(x)$ is forced to be exactly equal to the functional value at $N+1$ data points or nodes
- The p^{th} derivative of the polynomial $g(x)$ is an approximation to the p^{th} derivative of $f(x)$

Approximations to First and Second Derivatives Using Quadratic Interpolation

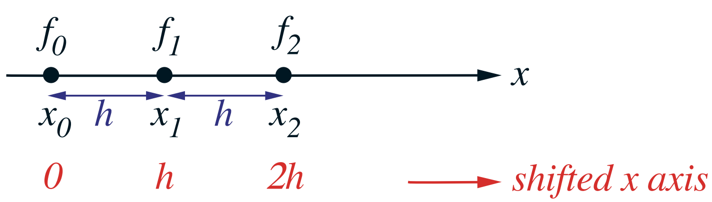
- We will illustrate the use of interpolation to derive FD approximations to first and second derivatives using a 3 node quadratic interpolation function

  - For first derivatives $p=1$ and we must establish at least an interpolating polynomial of degree $N=1$ with $N+1=2$ nodes

  - For second derivatives $p=2$ and we must establish at least an interpolating polynomial of degree $N=2$ with $N+1=3$ nodes

  - Thus a quadratic interpolating function will allow us to establish both first and second derivative approximations

- Apply a shifted coordinate system to simplify the derivation without affecting the generality of the derivation



<u>Develop a quadratic interpolating polynomial</u>

- We apply the ***Power Series*** method to derive the appropriate interpolating polynomial
  - Alternatively we could use either ***Lagrange basis functions*** or ***Newton forward or backward interpolation*** approaches in order to establish the interpolating polynomial
- The 3 node quadratic interpolating polynomial has the form

$$g(x) = a_o x^2 + a_1 x + a_2$$

- The approximating Lagrange polynomial ***must*** match the functional values at all $N+1$ data points or nodes ($x_o = 0$, $x_1 = h$, $x_2 = 2h$)

$$g(x_o) = f_o \quad \Rightarrow \quad a_o 0^2 + a_1 0 + a_2 = f_o$$

$$g(x_1) = f_1 \quad \Rightarrow \quad a_o h^2 + a_1 h + a_2 = f_1$$

$$g(x_2) = f_2 \quad \Rightarrow \quad a_o (2h)^2 + a_1 (2h) + a_2 = f_2$$

- Setting up the constraints as a system of simultaneous equations

$$\begin{bmatrix} 0 & 0 & 1 \\ h^2 & h & 1 \\ 4h^2 & 2h & 1 \end{bmatrix} \begin{bmatrix} a_o \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} f_o \\ f_1 \\ f_2 \end{bmatrix}$$

- Solve for a_o, a_1, a_2

$$a_o = \frac{f_2 - 2f_1 + f_o}{2h^2} \quad , \quad a_1 = \frac{4f_1 - f_2 - 3f_o}{2h} \quad , \quad a_2 = f_o$$

- The interpolating polynomial and its derivative are equal to:

$$g(x) = \left[\frac{f_2 - 2f_1 + f_o}{2h^2}\right]x^2 + \left[\frac{4f_1 - f_2 - 3f_o}{2h}\right]x + f_o$$

$$g^{(1)}(x) = \left[\frac{f_2 - 2f_1 + f_o}{h^2}\right]x + \left[\frac{4f_1 - f_2 - 3f_o}{2h}\right]$$

Evaluating $g^{(1)}(x_o)$ to obtain a forward difference approximation to the first derivative

- Evaluating the derivative of the interpolating function at $x_o = 0$

$$g^{(1)}(x_o) = g^{(1)}(0) \quad \Rightarrow$$

$$g^{(1)}(x_o) = \frac{-3f_o + 4f_1 - f_2}{2h}$$

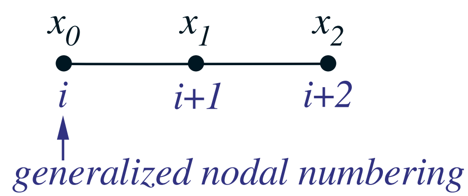
- Since the function $f(x)$ is approximated by the interpolating function $g(x)$

$$f^{(1)}\Big|_{x_o} \cong g^{(1)}(x_o)$$

- Substituting in for the expression for $g^{(1)}(x_o)$

$$f^{(1)}\Big|_{x_o} \cong \frac{-3f_o + 4f_1 - f_2}{2h}$$

- Generalize the node numbering for the approximation



- This results in the generic 3 node forward difference approximation to the first derivative at node i

$$f_i^{(1)} \cong \frac{-3f_i + 4f_{i+1} - f_{i+2}}{2h}$$

Evaluating $g^{(1)}(x_1)$ to obtain a central difference approximation to the first derivative

- Evaluating the derivative of the interpolating function at $x_1 = h$

$$g^{(1)}(x_1) = g^{(1)}(h) \qquad \Rightarrow$$

$$g^{(1)}(x_1) = \frac{(f_2 - 2f_1 + f_o)}{h^2}h + \frac{4f_1 - f_2 - 3f_o}{2h} \qquad \Rightarrow$$

$$g^{(1)}(x_1) = \frac{f_2 - f_o}{2h}$$

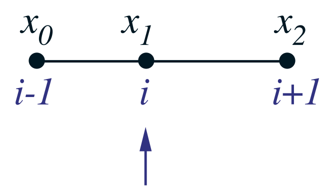
- Again since the function $f(x)$ is approximated by the interpolating function $g(x)$

$$f^{(1)}\Big|_{x_1} \cong g^{(1)}(x_1)$$

- Substituting in for the expression for $g^{(1)}(x_1)$

$$f^{(1)}\Big|_{x_1} \cong \frac{f_2 - f_o}{2h}$$

- Generalize the node numbering



- This results in the generic expression for the three node central difference approximation to the first derivative

$$f_i^{(1)} \cong \frac{f_{i+1} - f_{i-1}}{2h}$$

Evaluating $g^{(1)}(x_2)$ to obtain a backward difference approximation to the first derivative

- Evaluating the derivative of the interpolating function at $x_2 = 2h$

$$g^{(1)}(x_2) = g^{(1)}(2h) \quad \Rightarrow$$

$$g^{(1)}(x_2) = \frac{(f_2 - 2f_1 + f_o)}{h^2}2h + \frac{4f_1 - f_2 - 3f_o}{2h} \quad \Rightarrow$$

$$g^{(1)}(x_2) = \frac{3f_2 - 4f_1 + f_o}{2h}$$

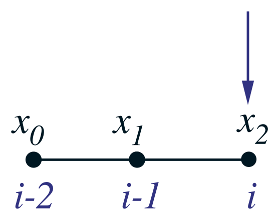
- Again since the function $f(x)$ is approximated by the interpolating function $g(x)$

$$f^{(1)}\Big|_{x_2} \cong g^{(1)}(x_2)$$

- Substituting in for the expression for $g^{(1)}(x_2)$

$$f^{(1)}\Big|_{x_2} \cong \frac{3f_2 - 4f_1 + f_o}{2h}$$

- Generalizing the node numbering



- This results in the generic expression for a three node backward difference approximation to the first derivative

$$f_i^{(1)} \cong \frac{3f_i - 4f_{i-1} + f_{i-2}}{2h}$$

Evaluating $g^{(2)}(x_o)$ to obtain a forward difference approximation to the second derivative

- We note that in general $g^{(2)}(x)$ can be computed as:

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

- Evaluating the second derivative of the interpolating function at $x_o = 0$:

$$g^{(2)}(x_o) = g^{(2)}(0) \qquad \Rightarrow$$

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

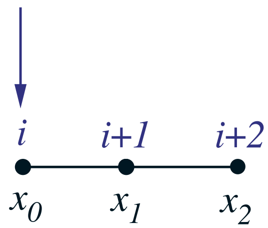
- Again since the function $f(x)$ is approximated by the interpolating function $g(x)$, the second derivative at node x_o is approximated as:

$$f^{(2)}\Big|_{x_o} \cong g^{(2)}(x_o)$$

- Substituting in for the expression for $g^{(2)}(x_o)$

$$f^{(2)}\Big|_{x_o} \cong \frac{f_2 - 2f_1 + f_o}{h^2}$$

- Generalizing the node numbering



- This results in the generic expression for a three node forward difference approximation to the second derivative

$$\boldsymbol{f_i^{(2)} \cong \frac{f_{i+2} - 2f_{i+1} + f_i}{h^2}}$$

Evaluating $g^{(2)}(x_1)$ to obtain a central difference approximation to the second derivative

- Evaluating the second derivative of the interpolating function at $x_1 = h$:

$$g^{(2)}(x_1) = g^{(2)}(h) \qquad \Rightarrow$$

$$g^{(2)}(x_1) = \frac{f_2 - 2f_1 + f_o}{h^2}$$

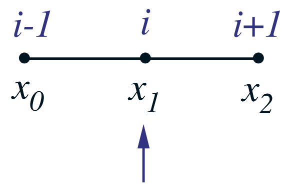
- Again since the function $f(x)$ is approximated by the interpolating function $g(x)$, the second derivative at node x_1 is approximated as:

$$f^{(2)}\Big|_{x_1} \cong g^{(2)}(x_1)$$

- Substituting in for the expression for $g^{(2)}(x_1)$

$$f^{(2)}\Big|_{x_1} \cong \frac{f_2 - 2f_1 + f_o}{h^2}$$

- Generalizing node numbering



- This results in the generic expression for a three node central difference approximation to the second derivative

$$f_i^{(2)} \cong \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2}$$

<u>Notes on developing differentiation formulae by interpolating polynomials</u>

- In general we can ***use*** any of the interpolation techniques to develop an interpolation function of degree $N \geq p$. We can then simply differentiate the interpolating function and evaluate it at any of the nodal points used for interpolation in order to derive an approximation for the p^{th} derivative.
- Orders of accuracy may vary due to the accuracy of the interpolating function varying.
- Exact accuracy can be obtained by substituting in Taylor series expansions ***or*** by considering the accuracy of the approximating polynomial $g(x)$.

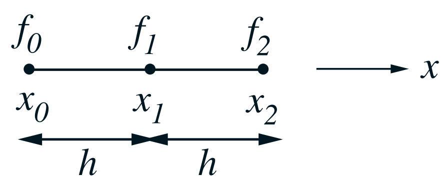
Approximations and Associated Error Estimates to First and Second Derivatives Using Quadratic Interpolation

- ***We can derive an error estimate when using interpolating polynomials to establish finite difference formulae by simply differentiating the error estimate associated with the interpolating function.***

- We will illustrate the use of a 3 node Newton forward interpolation formula to derive:
  - A central approximation to the first derivative with its associated error estimate
  - A forward approximation to the second derivative with its associated error estimate

Developing a 3 node interpolating function using Newton forward interpolation

- A quadratic interpolating polynomial ($N = 2$) has 3 associated nodes ($N + 1 = 3$) or interpolating points. We again assume that the nodes are evenly distributed as:



- With a quadratic interpolating polynomial, we can derive differentiation formulae for both the first and second derivatives but no higher

- The approximating or interpolating function is defined using Newton forward interpolation as:

$$f(x) = g(x) + e(x)$$

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

$$e(x) = \frac{(x - x_o)(x - x_1)(x - x_2)}{3!} f^{(3)}(\xi) \qquad x_o < \xi < x_2$$

- The error can be approximately expressed in either of the following forms:

$$e(x) \cong \frac{(x - x_o)(x - x_1)(x - x_2)}{3!} f_o^{(3)}$$

$$e(x) \cong \frac{(x - x_o)(x - x_1)(x - x_2)}{3!} \frac{\Delta^3 f_o}{h^3}$$

- These latter two forms which do not involve ξ are more suitable for the necessary differentiation w.r.t. x since ξ is functionally dependent on x, i.e. $\xi = \xi(x)$

- The forward difference operators are defined as:

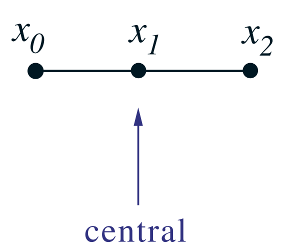
$$\Delta f_o \equiv f_1 - f_o$$

$$\Delta^2 f_o \equiv f_2 - 2f_1 + f_o$$

<u>Deriving a central approximation to the first derivative and the associated error estimate</u>

- Evaluating the first derivative of the function at x_1 :

$$f^{(1)}\Big|_{x = x_1} = g^{(1)}(x_1) + e^{(1)}(x_1)$$



- Evaluating $g^{(1)}(x)$ in the previous expression

$$f^{(1)}\Big|_{x=x_1} = \left[\frac{\Delta f_o}{h} + \frac{1}{2!}(x-x_o)\frac{\Delta^2 f_o}{h^2} + \frac{1}{2!}(x-x_1)\frac{\Delta^2 f_o}{h^2} + e^{(1)}(x)\right]_{x=x_1}$$

$$f^{(1)}\Big|_{x=x_1} = \frac{\Delta f_o}{h} + \frac{1}{2!}(x_1-x_o)\frac{\Delta^2 f_o}{h^2} + e^{(1)}(x_1)$$

$$f^{(1)}\Big|_{x=x_1} = \frac{f_1-f_o}{h} + \frac{1}{2}\frac{h}{h^2}(f_2-2f_1+f_o) + e^{(1)}(x_1)$$

$$f^{(1)}\Big|_{x=x_1} = \frac{f_2-f_o}{2h} + e^{(1)}(x_1)$$

- Now evaluate $e^{(1)}(x)$ using a non ξ dependent expression for the error term and evaluating this expression at x_1

$$e^{(1)}(x_1) \cong \frac{1}{3!} f^{(3)}(x_o)[(x-x_1)(x-x_2)+(x-x_o)(x-x_2)+(x-x_o)(x-x_1)]_{x\,=\,x_1}$$

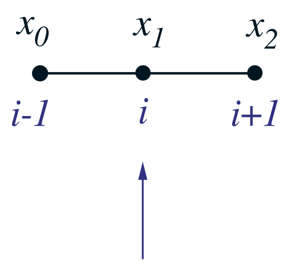
$$e^{(1)}(x_1) \cong \frac{1}{3!} f^{(3)}(x_o)(x_1-x_o)(x_1-x_2)$$

$$e^{(1)}(x_1) \cong -\frac{h^2}{3!} f^{(3)}(x_o)$$

- Substituting for $e^{(1)}(x_1)$ results in:

$$f^{(1)}\Big|_{x\,=\,x_1} \cong \frac{f_2-f_o}{2h} - \frac{h^2}{3!} f^{(3)}(x_o)$$

- Generalizing node numbering as:



- This results in the generic expression for a three node central difference approximation to the first derivative with an appropriate error estimate

$$f_i^{(1)} = \frac{f_{i+1} - f_{i-1}}{2h} + E$$

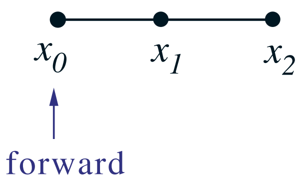
$$E \cong -\frac{h^2}{6} f^{(3)}(x_{i-1})$$

- Notes
  - The same discrete differentiation formulae can be derived using the Taylor series approach.
  - The results for the error estimates are the same regardless of whether you:
    - Apply differentiation to $e(x)$, which represents the error estimate for $g(x)$.
    - Apply a Taylor series analysis to the differentiation formula you derived.
  - The error estimate for the interpolating function $g(x)$ with ξ is ***most*** precise in a formal sense (i.e. it includes all H.O.T. as well!). However there is a weak dependence of ξ on x and therefore some inaccuracies may be incurred when differentiating $e(x)$ to obtain an error estimate for the corresponding finite difference approximation.
  - Practically, for estimating the error of the differentiating formula derived by estimating $e^{(1)}(x)$, we can apply the procedure used and examine an estimate of the error which does not depend on ξ. This is equivalent to examining the leading order term in the truncated series.
  - Note that the derivative in the error formula on the previous page may also be estimated at x_i

Deriving a forward difference approximation to the second derivative and the associated error estimate

- Evaluating the second derivative of the function at x_o :

$$f^{(2)}\Big|_{x = x_o} = g^{(2)}(x_o) + e^{(2)}(x_o)$$



- Evaluate $g^{(2)}(x)$ and substitute

$$f^{(2)}\Big|_{x = x_o} = \left[\frac{1}{2!}\frac{\Delta^2 f_o}{h^2} + \frac{1}{2!}\frac{\Delta^2 f_o}{h^2} + e^{(2)}(x)\right]_{x = x_o}$$

$$f^{(2)}\Big|_{x = x_o} = \frac{\Delta^2 f_o}{h^2} + e^{(2)}(x_o)$$

$$f^{(2)}\Big|_{x = x_o} = \frac{f_2 - 2f_1 + f_o}{h^2} + e^{(2)}(x_o)$$

- Now evaluate $e^{(2)}(x)$ using a non ξ dependent expression for the error term and evaluating this expression at x_o

$$e^{(2)}(x_o) \cong \frac{1}{3!} f^{(3)}(x_o)[(x-x_1)+(x-x_2)+(x-x_o)+(x-x_2)+(x-x_o)+(x-x_1)]_{x=x_o}$$

$$e^{(2)}(x_o) \cong \frac{1}{3!} f^{(3)}(x_o)[(x_o-x_1)+(x_o-x_2)+(x_o-x_o)+(x_o-x_2)+(x_o-x_o)+(x_o-x_1)]$$

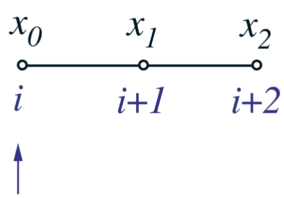
$$e^{(2)}(x_o) \cong \frac{1}{3!} f^{(3)}(x_o)[-h-2h+0-2h+0-h]$$

$$e^{(2)}(x_o) \cong -h\ f^{(3)}(x_o)$$

- Substituting for $e^{(2)}(x_o)$ results in:

$$f^{(2)}\Big|_{x=x_o} \cong \frac{f_2 - 2f_1 + f_o}{h^2} - hf^{(3)}(x_o)$$

- Generalizing the node numbering:



- This results in the generic expression for a three node forward difference approximation to the second derivative with an appropriate error estimate

$$f^{(2)} = \frac{f_{i+2} - 2f_{i+1} - f_i}{h^2} + E$$

$$E \cong -hf^{(3)}(x_i)$$

SUMMARY OF LECTURE 6, 7 AND 8

- Difference formulae can be developed such that linear combinations of functional values at various nodes approximate a derivative at a node.
- In general, to develop a difference formula for $f_i^{(p)}$ you need $p+1$ nodes for $O(h)$ accuracy and $p+N$ nodes for $O(h)^N$ accuracy. Central approximations for even order derivatives require fewer nodes due to a fortunate cancelation of error terms.
- The generic form to evaluate a difference formula

$$f_i^{(p)} - E = \frac{a_\alpha f_\alpha + a_\beta f_\beta + \ldots + a_\lambda f_\lambda}{h^p}$$

  - $E = O(h)^N$ when $p+N$ nodes are used
  - Substitute in Taylor series expansions for f_α, f_β etc. about node i, re-arrange equations such that coefficients multiply equal order derivatives at node i and generate algebraic equations by setting the coefficient of $f_i^{(p)}$ equal to 1 and the $p+N-1$ other coefficients equal to zero.
  - Solve for a_α, a_β, ...

- Forward, central and backward difference operators can be manipulated in ways analogous to differentiation to develop higher order differences

- Formulae to approximate differentiation using the difference operators can be established. However, general formulae for any order accuracy are difficult to establish. Also you still need Taylor series analysis to derive the accuracy of the approximations.

- Numerical differentiation formulae can be established by defining an interpolating polynomial for at least $p+1$ nodes (to evaluate the p^{th} derivative). Any interpolating technique formula can be used. The numerical differencing formula is simply the differentiated interpolating polynomial evaluated at one of the nodes used for interpolation. The node at which the formula is evaluated establishes whether the approximation is forward, backward, central, etc.