Basic Concepts in Numerical Analysis

Larry Caretto
Mechanical Engineering 501AB

Seminar in Engineering Analysis

November 6, 2017

California State University Northridge

Outline

- Review last class
- Midterm Exam November 15 covers material on differential equations and Laplace transforms (no phase plots)
- Overview of numerical solutions
  - Initial value problems in first-order equations
  - Systems of first order equations and initial value problems in higher order equations
  - Boundary value problems
  - Stiff systems and eigenvalues

California State University Northridge 2

Review Last Class

- Phase plots, critical points, and stability
- Look at system of two linear homogenous, autonomous equations
  - d**y**/dt = **Ay** (no function of time)
- Critical points and stability depend on matrix eigenvalues which depend on determinant properties
- Described various critical points: node, center, saddle point and spiral

California State University Northridge 3

Numerical Analysis Problems

- Numerical solution of algebraic equations and eigenvalue problems
- Solution of one or more nonlinear algebraic equations **f**(**x**) = **0**
- Linear and nonlinear optimization
- Constructing interpolating polynomials
- Numerical quadrature
- Numerical differentiation
- Numerical differential equations

California State University Northridge 4

Interpolation

- Start with N data pairs x_i, y_i
- Find a function (polynomial) that can be used for interpolation
- Basic rule: the interpolation polynomial must fit all points exactly
- Denote the polynomial as p(x)
- The basic rule is that $p(x_i) = y_i$
- Many different forms

California State University Northridge 5

Newton Polynomials

- $p(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2) + \dots + a_{n-1}(x - x_0)(x - x_1)(x - x_2) \dots (x - x_{n-2})$
  - n – 1 data points numbered 0 to n – 2
- Terms with factors of $x - x_i$ are zero when $x = x_i$
  - Have $p(x_i) = y_i$ to find a_i, i = 0 to n – 1
- $a_0 = y_0$, $a_1 = (y_1 - y_0) / (x_1 - x_0)$
- $y_2 = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1)$
  - Solve for a_2 using results for a_0 and a_1

California State University Northridge 6

Newton Polynomials II

- $y_2 = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1)$

$$a_2 = \frac{y_2 - a_0 - a_1(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)} = \frac{y_2 - y_0 - \frac{y_1 - y_0}{x_1 - x_0}(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)}$$

- Could continue in this fashion to determine coefficients from data
- Use alternative scheme – not derived here – known as divided difference table to compute a_k from same data

Divided Difference Table

- Enter data on x_i and y_i in rows of table skipping one row between entries
- Start with y_i data as zeroth divided difference
- First divided difference, $F_i = (y_{i+1} - y_i) / (x_{i+1} - x_i)$
  - Second (or later) divided difference is difference of first (or later) differences
  - a_i coefficients are initial divided differences

Divided Difference Table

x_0	y_0	$\leftarrow a_0$		
		$F_0 = \frac{y_1 - y_0}{x_1 - x_0}$	$\leftarrow a_1$	
x_1	y_1		$S_0 = \frac{F_1 - F_0}{x_2 - x_0}$	$\leftarrow a_2$
		$F_1 = \frac{y_2 - y_1}{x_2 - x_1}$		$T_0 = \frac{S_1 - S_0}{x_3 - x_0}$
x_2	y_2		$S_1 = \frac{F_2 - F_1}{x_3 - x_1}$	$\uparrow a_3$
		$F_2 = \frac{y_3 - y_2}{x_3 - x_2}$		$T_1 = \frac{S_2 - S_1}{x_4 - x_1}$
x_3	y_3		$S_2 = \frac{F_3 - F_2}{x_4 - x_2}$	

$F_3 = \frac{y_4 - y_3}{x_4 - x_3}$

Divided Difference Example

0	0	$\leftarrow a_0$		
		$F_0 = \frac{10-0}{10-0} = 1$	$\leftarrow a_1$	
10	10		$S_0 = \frac{3-1}{20-0} = .1$	$\leftarrow a_2$
		$F_1 = \frac{40-10}{20-10} = 3$		$T_0 = \frac{.15-.1}{30-0} = \frac{1}{600}$
20	40		$S_1 = \frac{6-3}{30-10} = .15$	$\uparrow a_3$
		$F_2 = \frac{100-40}{30-20} = 6$		
30	100			

Divided Difference Example II

- Divided difference table gives $a_0 = 0$, $a_1 = 1$, $a_2 = .1$, and $a_3 = 1/600$
- Polynomial $p(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2)$ $= 0 + 1(x - 0) + 0.1(x - 0)(x - 10) + (1/600)(x - 0)(x - 10)(x - 20) = x + 0.1x(x - 10) + (1/600)x(x - 10)(x - 20)$
- Check $p(30) = 30 + .1(30)(20) + (1/600)(30)(20)(10) = 30 + 60 + 10 = 100$ (correct)

Start at Any Point in Data Table

x_{-1}	y_{-1}			
x_0	y_0	$\leftarrow a_0$		
		$F_0 = \frac{y_1 - y_0}{x_1 - x_0}$	$\leftarrow a_1$	
x_1	y_1		$S_0 = \frac{F_1 - F_0}{x_2 - x_0}$	$\leftarrow a_2$
		$F_1 = \frac{y_2 - y_1}{x_2 - x_1}$		$T_0 = \frac{S_1 - S_0}{x_3 - x_0}$
x_2	y_2		$S_2 = \frac{F_2 - F_1}{x_3 - x_1}$	$\uparrow a_3$

$F_2 = \frac{y_3 - y_2}{x_3 - x_2}$

Difference Example from x = 10

0	0			
10	10	← a_0		
		$F_0 = \frac{40-10}{20-10} = 3$	← a_1	
20	40		$S_0 = \frac{6-3}{30-10} = .15$ ← a_2	
		$F_1 = \frac{100-40}{30-20} = 6$		$T_0 = \frac{1/5-.15}{40-10} = \frac{1}{600}$
30	100		$S_1 = \frac{10-6}{40-20} = \frac{1}{5}$	↑ a_3
		$F_2 = \frac{200-100}{40-30} = 10$		

Final data point not shown is x = 40, y = 200

Divided Difference Calculation II

- Divided difference table gives $a_0 = 10$, $a_1 = 3$, $a_2 = .15$, and $a_3 = 1/600$
- Polynomial $p(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + a_3(x - x_0)(x - x_1)(x - x_2)$ $= 10 + 3(x - 10) + 0.15(x - 10)(x - 20) + (1/600)(x - 10)(x - 20)(x - 30) = x + 0.1x(x - 10) + (1/600)x(x - 10)(x - 20)$
- Check $p(40) = 10+3(30)+.15(30)(20) + (1/600)(40)(30)(20)=10+90+90+10=200$ *(correct)*

Divided Difference Code

```
for ( i = 0; i < n; i++ )
  D[0][i] = y[i];

for ( k = 1; k < n; k++ )
  for ( i = 0; i < n – k; i++ )
    D[k][i] = ( D[k-1][i+1] –
 D[k-1][i] ) / ( x[i+k] – x[i] );
```

- D[k][i] is i^{th} value of k^{th} divided difference
- Code for n data points (0 to n–1)

Constant Step Size

- Divided differences work for equal or unequal step size in x
- If $\Delta x = h$ is a constant we have simpler results
  - $F_k = \Delta y_k/h = (y_{k+1} - y_k)/h$
  - $S_k = \Delta^2 y_k/2h^2 = (y_{k+2} - 2y_{k-1} + y_k)/2h^2$
  - $T_k = \Delta^3 y_k/6h^3 = (y_{k+3} - 3y_{k+2} + 3y_{k+1} - y_k)/6h^3$
  - $\Delta^n y_k$ is called the n^{th} forward difference
  - Can also define backwards and central differences

Interpolation Approaches

- When we have N data points how do we interpolate among them?
  - Order N-1 polynomial not good choice
  - Use piecewise polynomials of lower order (linear or quadratic)
  - Can match first and or higher derivatives where piecewise polynomials join
  - Cubic splines are piecewise cubic polynomials that match first and second derivatives [as well as (x_k, y_k) values]

Cubic Spline Overview

- Have N cubic polynomials, $a_i + b_i x + c_i x^2 + d_i x^3$, with end point of 1 polynomial the start of next, requires N + 1 data points
  - Data points numbered 0 to N with polynomials numbered 1 to N
- Need 4N equations to get N values for polynomial coefficients: a_i, b_i, c_i, and d_i
- Each polynomial fits data points at ends: $p_k(x_{k-1}) = y_{k-1}$ and $p_k(x_k) = y_k$, k = 1, N

Cubic Spline Overview II

- Have continuity of first and second derivatives: $p_{k-1}'(x_k) = p_k'(x_k)$ and $p_{k-1}''(x_k) = p_k''(x_k)$
- Matching data points gives 2N equations and derivative continuity gives 2N – 2
- Have 4N – 2 equations for 4N unknown polynomial coefficients
- Different models of end point behavior used to provide additional 2 equations

California State University Northridge 19

Splines in MATLAB

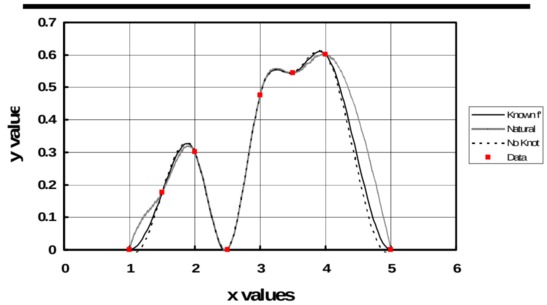
- Use spline function in MATLAB to get one or more interpolated points
  - xIn is array of y data for spline fit
  - yIn is array of x data for spline fit
  - Apply spline to x which can be a single data point or an array using command below

```
>> spline(xIn, yIn, x)
```

  - Generally uses not-a-knot end slopes
- Also has routine `unmkpp` to get details of resulting spline coefficients

California State University Northridge 20

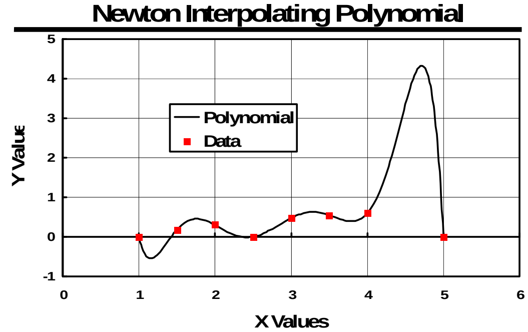
Cubic Spline Interpolation



Results show effect of different methods used to treat end points

California State University Northridge 21

Newton Interpolating Polynomial



High-order polynomials can give unrealistic fits to data

California State University Northridge 22

Polynomial Applications

- Data interpolation
- Approximation functions in numerical quadrature and solution of ODEs
- Basis functions for finite element methods
- Can obtain equations for numerical differentiation
- Statistical curve fitting (not discussed here) usually used in practice

California State University Northridge 23

Derivative Expressions

- Obtain from differentiating interpolation polynomials or from Taylor series
- Series expansion for f(x) about x = a

$$f(x) = f(a) + \left.\frac{df}{dx}\right|_{x=a}(x-a) + \frac{1}{2!}\left.\frac{d^2 f}{dx^2}\right|_{x=a}(x-a)^2 + \frac{1}{3!}\left.\frac{d^3 f}{dx^3}\right|_{x=a}(x-a)^3 + \ldots$$

- Note: $d^0f/dx^0 = f$ and $0! = 1$

$$f(x) = \sum_{n=0}^{\infty} \frac{1}{n!}\left.\frac{d^n f}{dx^n}\right|_{x=a}(x-a)^n$$

- What is error from truncating series?

California State University Northridge 24

Truncation Error

- If we truncate series after m terms

$$f(x) = \underbrace{\sum_{n=0}^{m} \frac{1}{n!} \left.\frac{d^n f}{dx^n}\right|_{x=a} (x-a)^n}_{\text{Terms used}} + \underbrace{\sum_{n=m+1}^{\infty} \frac{1}{n!} \left.\frac{d^n f}{dx^n}\right|_{x=a} (x-a)^n}_{\text{Truncation error, } \varepsilon_m}$$

- Use theorem of mean to write truncation error as single term at unknown location, ξ, between x and a

$$\varepsilon_m = \sum_{n=m+1}^{\infty} \frac{1}{n!} \left.\frac{d^n f}{dx^n}\right|_{x=a} (x-a)^n = \frac{1}{(m+1)!} \left.\frac{d^{m+1} f}{dx^{m+1}}\right|_{x=\xi} (x-a)^{m+1}$$

Derivative Expressions

- Look at finite-difference grid with equal spacing: $h = \Delta x$ so $x_i = x_0 + ih$
- Taylor series about $x = x_i$ gives $f(x_i + kh) = f[x_0 + (i+k)h] = f_{i+k}$ in terms of $f(x_i) = f_i$

$$f(x_i + kh) = f(x_i) + \left.\frac{df}{dx}\right|_{x=x_i} kh + \frac{1}{2!}\left.\frac{d^2 f}{dx^2}\right|_{x=x_i} (kh)^2 + \frac{1}{3!}\left.\frac{d^3 f}{dx^3}\right|_{x=x_i} (kh)^3 + \ldots..$$

- Compact derivative notation

$$f_i' = \left.\frac{df}{dx}\right|_{x=x_i} \qquad f_i'' = \left.\frac{d^2 f}{dx^2}\right|_{x=x_i} \quad \cdots \quad f_{i+k}^n = \left.\frac{d^n f}{dx^n}\right|_{x=x_{i+k}}$$

Derivative Expressions II

- Combine all definitions for compact series notation

$$f(x_i + kh) = f(x_i) + \left.\frac{df}{dx}\right|_{x=x_i} kh + \frac{1}{2!}\left.\frac{d^2 f}{dx^2}\right|_{x=x_i} (kh)^2 + \frac{1}{3!}\left.\frac{d^3 f}{dx^3}\right|_{x=x_i} (kh)^3 + \ldots..$$

$$f_{i+k} = f_i + f_i' kh + \frac{f_i''(kh)^2}{2!} + \frac{f_i'''(kh)^3}{3!} + \ldots..$$

- Use this formula to get expansions for various grid locations about $x = x_i$ and use results to get derivative expressions

Derivative Expressions III

- Apply general equation for k = 1 and k = –1

$$f_{i+k} = f_i + f_i' kh + \frac{f_i''(kh)^2}{2!} + \frac{f_i'''(kh)^3}{3!} + \ldots$$

$$f_{i+1} = f_i + f_i' h + \frac{f_i'' h^2}{2!} + \frac{f_i''' h^3}{3!} + \ldots..$$

$$f_{i-1} = f_i - f_i' h + \frac{f_i'' h^2}{2!} - \frac{f_i''' h^3}{3!} + \ldots..$$

Forward

$$f_i' = \frac{f_{i+1} - f_i}{h} - \frac{f_i'' h}{2!} - \frac{f_i''' h}{3!} - \ldots.. = \frac{f_{i+1} - f_i}{h} + Ah$$

Backward

$$f_i' = \frac{f_i - f_{i-1}}{h} + \frac{f_i'' h}{2!} - \frac{f_i''' h}{3!} - \ldots.. = \frac{f_i - f_{i-1}}{h} + Ah$$

Derivative Expressions IV

- Subtract f_{i+1} and f_{i-1} expressions

$$f_{i+1} = f_i + f_i' h + \frac{f_i'' h^2}{2!} + \frac{f_i''' h^3}{3!} + \ldots..$$

$$f_{i-1} = f_i - f_i' h + \frac{f_i'' h^2}{2!} - \frac{f_i''' h^3}{3!} + \ldots..$$

$$f_{i+1} - f_{i-1} = 2 f_i' h + \frac{2 f_i''' h^3}{3!} + \frac{2 f_i''''' h^5}{5!} + \ldots..$$

$$f_i' = \frac{f_{i+1} - f_{i-1}}{2h} + \frac{f_i''' h^2}{3!} + \frac{f_i''''' h^4}{5!} - \ldots.. = \frac{f_{i+1} - f_{i-1}}{2h} + Ah^2$$

- Result called central difference expression

Order of the Error

- Forward and backward derivative have error term that is proportional to h
- Central difference error is proportional to h^2
- Error proportional to h^n called n^{th} order
- Reducing step size by a factor of α reduces n^{th} order error by α^n

$$\varepsilon_2 \approx \varepsilon_1 \left(\frac{h_2}{h_1}\right)^n$$

Order of the Error Notation

- Write the error term for n^{th} error term as $O(h^n)$
  - Big oh notation, O, denotes order
  - Recognizes that factor multiplying h^n may change slightly with h

First order forward

$$f_i' = \frac{f_{i+1} - f_i}{h} + O(h)$$

First order backward

$$f_i' = \frac{f_i - f_{i-1}}{h} + O(h)$$

Second order central

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

31

Higher Order Derivatives

- Add f_{i+1} and f_{i-1} expressions; solve for f_i''

$$f_{i+1} = f_i + f_i' h + \frac{f_i'' h^2}{2!} + \frac{f_i''' h^3}{3!} + ...$$

$$f_{i-1} = f_i - f_i' h + \frac{f_i'' h^2}{2!} - \frac{f_i''' h^3}{3!} + ...$$

$$f_{i+1} + f_{i-1} = 2 f_i + 2\frac{f_i'' h^2}{2!} + \frac{2 f_i'''' h^4}{4!} + \frac{2 f_i'''''' h^6}{6!} + ...$$

$$f_i'' = \frac{f_{i+1} + f_{i-1} - 2 f_i}{h^2} + \frac{f_i'''' h^2}{3!} + \frac{f_i'''''' h^4}{5!} - ..... = \frac{f_{i+1} + f_{i-1} - 2 f_i}{h^2} + O\left(h^2\right)$$

- f'' is second-order, central difference expression for second derivative

32

Higher Order Directional

- We can get higher truncation error expressions at the expense of more computations
- Get second order forward and backward derivative expressions from previous results and f_{i+2} and f_{i-2}, respectively
- Combine f_{i+2} and f_{i-2} equations with previous expressions for f_{i+1} and f_{i-1} to eliminate first order error term

33

Specific Taylor Series

- General equation $f_{i+k} = f_i + f_i' kh + \frac{f_i''(kh)^2}{2!} + \frac{f_i'''(kh)^3}{3!} + ...$
- k = 2 $f_{i+2} = f_i + 2 f_i' h + 4\frac{f_i'' h^2}{2!} + 8\frac{f_i''' h^3}{3!} + ...$
- k = -2 $f_{i-2} = f_i - 2 f_i' h + 4\frac{f_i'' h^2}{2!} - 8\frac{f_i''' h^3}{3!} + ...$
- k = 3 $f_{i+3} = f_i + 3 f_i' h + 9\frac{f_i'' h^2}{2!} + 27\frac{f_i''' h^3}{3!} + ...$
- k = -3 $f_{i-3} = f_i - 3 f_i' h + 9\frac{f_i'' h^2}{2!} - 27\frac{f_i''' h^3}{3!} + ...$

34

Second Order Forward

- Subtract $4f_{i+1}$ from f_{i+2} to eliminate h^2 term

$$f_{i+2} - 4 f_{i+1} = \left[f_i + 2 f_i' h + 4 f_i'' \frac{h^2}{2} + 8 f_i''' \frac{h^3}{6} + ... \right]$$

$$-4\left[f_i + f_i' h + f_i'' \frac{h^2}{2} + f_i''' \frac{h^3}{6} + ... \right] = -3 f_i - 2 f_i' h + 4 f_i''' \frac{h^3}{6} + ...$$

$$f_{i+2} - 4 f_{i+1} + 3 f_i = -2h f_i' + 4 f_i''' \frac{h^3}{6} + ...$$

$$f_i' = \frac{-f_{i+2} + 4 f_{i+1} - 3 f_i}{2h} + f_i''' \frac{h^2}{3} + ...$$

Second order error

35

Second Order Backwards

- Add $4f_{i-1}$ to $-f_{i-2}$ to eliminate h^2 term

$$-f_{i-2} + 4 f_{i-1} = -\left[f_i - 2 f_i' h + 4 f_i'' \frac{h^2}{2} - 8 f_i''' \frac{h^3}{6} + ... \right]$$

$$+4\left[f_i - f_i' h + f_i'' \frac{h^2}{2} - f_i''' \frac{h^3}{6} + .. \right] = 3 f_i - 2 f_i' h + 4 f_i''' \frac{h^3}{6} + ...$$

$$-f_{i-2} + 4 f_{i-1} - 3 f_i = -2h f_i' + 4 f_i''' \frac{h^3}{6} + ...$$

$$f_i' = \frac{f_{i-2} - 4 f_{i-1} + 3 f_i}{2h} + f_i''' \frac{h^2}{3} + ...$$

Second order error

36

Other Derivative Expressions

- Can continue in this fashion
  - Write Taylor series for f_{i+1}, f_{i-1}, f_{i+2}, f_{i-2}, f_{i+3}, f_{i-3}, etc.
  - Create linear combinations with factors that eliminate desired terms
  - Eliminate f_i term to obtain central difference
  - Keep only terms in f_k with $k \geq i$ for forward difference expressions
  - Keep only terms in f_k with $k \leq i$ for forward difference expressions
  - Results in numerical analysis texts/online

California State University Northridge 37

Other Derivative Formulas

$$f_i' = \frac{-11f_i + 18f_{i+1} - 9f_{i+2} + 2f_{i+3}}{2h} - f_i'''' \frac{h^3}{3} + \ldots$$

$$f_i' = \frac{11f_i - 18f_{i-1} + 9f_{i-2} - 2f_{i-3}}{2h} + f_i'''' \frac{h^3}{4} + \ldots$$

$$f_i' = \frac{f_{i-2} - 8f_{i-1} + 8f_{i+1} - f_{i+2}}{2h} + f_i''''' \frac{h^4}{30} + \ldots$$

$$f_i'' = \frac{-f_{i-2} + 16f_{i-1} - 30f_i + 16f_{i+1} - f_{i+2}}{12h^2} + f_i'''''' \frac{h^4}{90} + \ldots$$

California State University Northridge 38

Order of Error Examples

- Table 1 in online notes shows error in first derivative for e^x around x = 1
  - Using first- and second-order forward and second-order central differences
  - Step h = 0.4, 0.2, and 0.1
  - Error ratio for doubling step size
    - 4.01 to 4.02 for central differences
    - 2.07 to 2.15 for first-order forward differences
    - 4.32 to 4.69 for second-order forward

$$n \approx \frac{\log\left(\varepsilon_2/\varepsilon_1\right)}{\log\left(h_2/h_1\right)} = \frac{\log(\varepsilon_2) - \log(\varepsilon_1)}{\log(h_2) - \log(h_1)} = \frac{d\log(\varepsilon)}{d\log(h)}$$

California State University Northridge 39

Roundoff Error

- Possible in derivative expressions from subtracting close differences
- Example $f(x) = e^x$: $f'(x) \approx (e^{x+h} - e^{x-h})/(2h)$ and error at x = 1 is $(e^{1+h} - e^{1-h})/(2h) - e$

$$E = \frac{3.004166 - 2.722815}{2(0.1)} - 2.718282 = 4.5x10^{-3}$$

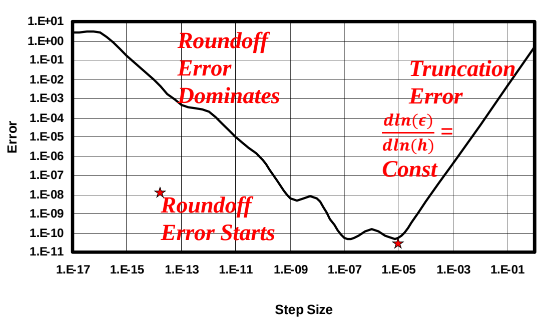
Second order error

$$E = \frac{2.7185536702 - 2.7180100139}{2(0.0001)} - 2.718281828459 = 4.5x10^{-9}$$

$$E = \frac{2.71828210028724 - 2.71828155660388}{2(0.0000001)} - 2.718281828 = 5.9x10^{-9}$$

California State University Northridge 40

Figure 2-1. Effect of Step Size on Error



California State University Northridge 41

Richardson Extrapolation

- Uses finite-difference method with two step sizes to get improved accuracy
- Start with E = F(h) + TE = F(h) + $O(h^n)$
  - E is exact result
  - F(h) is finite difference approximation with step size h
  - Truncation error, TE, is $O(h^n)$
  - Actually have an infinite series for error

$$TE = \frac{h^n}{n!}\left(\frac{d^n f}{dx^n}\right)_{x=\xi} = \sum_{k=n}^{\infty} \frac{h^n}{n!}\left(\frac{d^n f}{dx^n}\right)_{x=a} = \sum_{k=n}^{\infty} A_k h^k$$

California State University Northridge 42

Richardson Extrapolation II

- Look at evaluating error with two step sizes, h and kh
  - Exact value will not change
  - Create sum to display first error term

$$E = F(h) + TE = F(h) + \sum_{m=n}^{\infty} A_m h^m = F(h) + \boxed{A_n h^n} + \sum_{m=n+1}^{\infty} A_m h^m$$

$$E = F(kh) + TE = \cdots\cdots = F(kh) + \boxed{A_n (kh)^n} + \sum_{m=n+1}^{\infty} A_m (kh)^m$$

  - Multiply first equation by k^n and subtract the second equation to eliminate the A_n term

$$k^n E - E = k^n F(h) - F(kh) + \cancel{k^n A_n h^n - A_n (kh)^n} + O(h^{n+1})$$

May be h^{n+2}

Northridge 43

Richardson Extrapolation III

- Solve equation from previous slide for E

$$k^n E - E = k^n F(h) - F(kh) + O(h^{n+1})$$

$$E = \frac{k^n F(h) - F(kh)}{k^n - 1} + O(h^{n+1}) = RE + O(h^{n+1})$$

- The formula for the Richardson extrapolation, RE, has a higher order of the error
  - Truncation error for RE shown below

$$TE = \frac{k^n \sum_{k=n+1}^{\infty} A_m h^m - \sum_{k=n+1}^{\infty} A_m (kh)^m}{k^n - 1} = \sum_{k=n+1}^{\infty} B_m h^m \qquad B_m = \frac{k^n - k^m}{k^n - 1} A_m$$

Northridge 44

Richardson Extrapolation IV

- What does this mean?

$$E = \frac{k^n F(h) - F(kh)}{k^n - 1} + O(h^{n+1}) = RE + O(h^{n+1})$$

- E is the exact result, F(h) is a finite difference result with step size h
  - If we have two n^{th}-order finite difference results, with two step sizes h and kh, we can use this formula to get an improved result with an error order of n + 1 (or higher if the error term has every other power of h)

Northridge 45

Richardson Extrapolation V

- Richardson extrapolation for forward dcos(x)/dx at x = 1 and h = 0.1 & h = 0.2
  - What are k and n? ***$k = h_2/h_1 = 2$; n = order = 1***

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

$$f_i'(h=0.1) = \frac{\cos(1.1) - \cos(1)}{0.1} = -0.8670618$$

$$f_i'(h=0.2) = \frac{\cos(1.2) - \cos(1)}{0.2} = -0.8897228$$

$$RE = \frac{k^n F(h) - F(kh)}{k^n - 1} = \frac{2^1(-0.86706) - -0.88972}{2^1 - 1} = -0.84440093$$

- Extrapolation closer to correct value of $dcos(x)/dx|_{x=1} = -\sin(1) = -0.84147098$

Northridge 46

Richardson Extrapolation VI

- Richardson extrapolation for central dcos(x)/dx at x = 1 and h = 0.1 & h = 0.2
  - What are k and n? ***$k = h_2/h_1 = 2$; n = order = 2***

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

$$f_i'(h=0.1) = \frac{\cos(1.1) - \cos(0.9)}{2(0.1)} = -0.8400692$$

$$f_i'(h=0.2) = \frac{\cos(1.2) - \cos(0.8)}{2(0.2)} = -0.8358872$$

$$RE = \frac{k^n F(h) - F(kh)}{k^n - 1} = \frac{2^1(-0.84007) - -0.83587}{2^1 - 1} = -0.841468$$

- Extrapolation closer to correct value of $dcos(x)/dx|_{x=1} = -\sin(1) = -0.84147098$

Northridge 47