ME 704
Computational Methods in Thermal and Fluids Engineering
(KNI-1 Numerical Differentiation)

Kannan Iyer
Kiyer@iitb.ac.in

Department of Mechanical Engineering
Indian Institute of Technology, Bombay

Numerical Differentiation

Motivation for study

- Obtaining derivative at a point from a table of functional data
  - ❑ Obtaining 'h' from the measurement of T
  - ❑ Obtaining 'f' from a tabulated 'v' data
  - ❑ Generating methods for solving ODE/PDE

Derivatives from Polynomials

- Numerical derivatives can be obtained from polynomials and their function values

❑ First Order

$$P_1(x) = f(0) + s\Delta f(0) \qquad \text{where} \quad s = \frac{x - x_0}{h}$$

$$\text{Therefore} \quad P_1'(x) = \frac{df}{ds}\frac{ds}{dx} = \Delta f(0)\frac{1}{h} \qquad \therefore \frac{ds}{dx} = \frac{1}{h}$$

Derivatives from Polynomials (Cont'd

❑ Second Order

$$P_2(x) = f(0) + s\Delta f(0) + \frac{s(s-1)}{2!}\Delta^2 f(0)$$

$$\therefore P_2'(x) = \left[\Delta f(0) + \frac{2s-1}{2!}\Delta^2 f(0)\right]\frac{1}{h}$$

❑ Higher order approximations can similarly be obtained.

❑ Since each term is divided by h the accuracy of the derivative would be order h^n and not h^{n+1}

Derivatives from Polynomials (Cont'd)

Example: f=1/x

s = (3.44-3.40)/0.1 = 0.4

f ′(3.44) = -0.008404/0.1 | -0.08404

+[{2(0.4)-1}/2] (0.000468)/0.1 | -0.084508

+ [{3(0.4)^2-6(0.4)+2)}/6] (0.000040)/0.1 | -0.084503

Exact Value -0.084505

Derivatives from Polynomials (Cont'd)

- Similarly we can obtain derivatives using backward interpolating polynomial.
- We have shown that they would be equivalent by choosing proper value of 's'.
- We shall make use of these to derive finite difference relations later used in ODEs
- We can similarly obtain higher derivatives.
- As pointed earlier, the accuracies will reduce further due to divisions by higher orders of 'h'.

Numerical Integration

- The function f(x) may be a set of discrete values as in the case of properties
- It can be a complex function, in which case the function can be evaluated at some discrete values and integrated suitably
- We shall derive the procedures using Newton's forward interpolating polynomial
- Unlike differentiation, integration is an accurate process and the order of accuracy increases.

TRAPEZIODAL RULE (First Order)

$$P_1(x) = f(0) + s\Delta f(0)$$

$$\text{where} \quad s = \frac{x - x_0}{h} \qquad \text{or} \quad dx = h\,ds$$

$$\int_{low}^{high} f(x)\,dx = \int_0^1 f(s)\,h\,ds$$

$$= \int_0^1 \left(f(0) + s\Delta f(0)\right) h\,ds = h\left[f(0)s + \frac{s^2}{2}\Delta f(0)\right]_0^1$$

$$= \left[f(0) + \frac{\Delta f(0)}{2}\right]h = \left[f(0) + \frac{1}{2}\left(f(1) - f(0)\right)\right]h$$

$$= \frac{h}{2}\left(f(0) + f(1)\right)$$

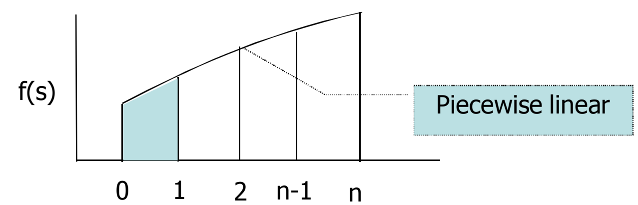
Trapezoidal Rule (Cont'd)

- As we have used first order polynomial the error term for polynomial is o ($h^{2)}$
- Since the integral involves a multiplication with h, the order increases to h^3 locally.

$$Error \quad Term = \int_0^1 \frac{s(s-1)}{2} h^2 f''(\xi) h ds$$

$$= -\frac{h^3}{12} f''(\xi)$$

Trapezoidal Rule (Cont'd)



$$I = \sum_{i=1}^{n-1} I_i = \sum_{i=1}^{n-1} \frac{h}{2}\left(f_i + f_{i+1}\right) = \frac{h}{2}\left(f_0 + 2f_1 + 2f_1 + \dots 2f_{n-1} + f_n\right)$$

$$Error = \sum_{i=0}^{n-1} -\frac{h^3}{12} f_i''(\xi) = -n\left(\frac{h^3}{12} f''(\xi)\right) = -\left(\frac{x_n - x_0}{h}\right)\left(\frac{h^3}{12} f''(\xi)\right)$$

Globally $O(h^2)$

Simpson's 1/3 Rule

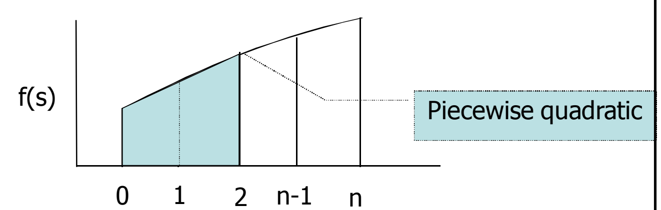
$$= \int_0^2 \left(f(0) + s\Delta f(0) + \frac{(s^2 - s)}{2} \Delta^2 f(0) \right) hds$$

$$= \left[sf(0) + \frac{s^2 \Delta f(0)}{2} + \left(\frac{s^3}{6} - \frac{s^2}{4} \right) \Delta^2 f(0) \right]_0^2 h$$

$$= \left[2f(0) + 2(f(1) - f(0)) + \frac{1}{3}(f(2) - 2f(1) + f(0)) \right] h$$

$$= \frac{h}{3}[f(0) + 4f(1) + f(2)]$$

Simpson's 1/3 Rule



$$I = \sum I_i = \sum \frac{h}{3}\left(f_i + 4f_{i+1} + f_{i+2}\right)$$

$$= \frac{h}{12}\left(f_0 + 4f_1 + 2f_2 + 4f_3 \dots + 2f_{n-2} + 4f_{n-1} + f_n\right)$$

Simpson's 1/3 Rule

$$Error \quad Term = \int_0^2 \frac{s(s-1)(s-2)}{6} h^3 f'''(\xi) h ds = 0$$

$$= \int_0^2 \frac{s(s-1)(s-2)(s-3)}{24} h^4 f''''(\xi) h ds$$

$$= -\frac{1}{90} h^5 f''''(\xi)$$

$$Global \quad Error = \sum -\frac{h^5}{90} f_i^{iv}(\xi) = -\frac{(x_n - x_0)}{2h} \frac{h^5}{90} f_i^{iv}(\xi)$$

Globally $O(h^4)$

Simpson's 3/8 Rule

- Simpson's 1/3 rule can be applied if only odd number of data points are available
  - To integrate when even number of points are available, the first 4 points can be integrated by Simpson's 3/8 rule and the remaining by Simpson's 1/3 rule.

$$I = \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3)$$

$$Local \quad Error = -\frac{3h^5}{80} f_i^{iv}(\xi)$$

Globally $O(h^4)$

Deferred Approach to the Limit

- Richardson's Extrapolation, also called 'Deferred Approach to the Limit' is a method to improve the accuracy of lower order methods.
  - In the domain of integration this method is called Romberg Integration
  - It can be shown that in using trapezoidal integration, the truncation errors can be written as $C_1 h^2 + C_2 h^4 + C_3 h^6 + \ldots$

Romberg Integration-I

- If the integration procedure is carried out for intervals h and 2h, we can write

$$I = I(h) + C_1 h^2 + C_2 h^4 + \ldots, \qquad (1)$$

$$I = I(2h) + C_1 (2h)^2 + C_2 (2h)^4 + \ldots, \qquad (2)$$

Multiplying Eq. (1) by 4 and subtracting Eq.(2) and then dividing by 3, we get

$$I = [4I(h) - I(2h)]/3 + O(h)^4$$

This can be rewritten as

$$I = I(h) + [I(h) - I(2h)]/3 + O(h)^4$$

- Thus we have a higher order solution from lower order solutions. This is called the **Richardson extrapolation**

Romberg Integration-II

- The above can be generalized by assuming a power law form

$$I = I(h) + C_1 h^n + C_2 h^m \ldots, \qquad (1)$$

$$I = I(2h) + C_1 (2h)^n + C_2 (2h)^m \ldots, \qquad (2)$$

Multiplying Eq. (1) by 2^n and subtracting Eq.(2) and then dividing by 2^n-1 we get

$$I = [2^n I(h) - I(2h)]/(2^n-1) + O(h)^m$$

This can be rewritten as

$$I = I(h) + [I(h) - I(2h)]/(2^n-1) + O(h)^m$$

Romberg Integration-III

- Thus, by using trapezoidal rule repeatedly, the accuracy can be further improved by computing with say h/2 and eliminating the next constant in the previous slide.
- In general, the correction formula is

$$\text{Improved Value} = \text{More accurate value} + \frac{[\text{More accurate value} - \text{Less accurate value}]}{2^n-1}$$

- Such a recursive algorithm is called **Romberg Integration** Procedure

Romberg Integration-IV

Level	Steps	Trapez $O(h^2)$	Richard $O(h^4)$	Richard $O(h^6)$	Richard $O(h^8)$
1	1 (2^0)	$I_{1,1}$			
2	2 (2^1)	$I_{2,1}$	$I_{2,2}$		
3	4(2^2)	$I_{3,1}$	$I_{3,2}$	$I_{3,3}$	
4	8(2^3)	$I_{4,1}$	$I_{4,2}$	$I_{4,3}$	$I_{4,4}$

Romberg Integration-V

```
l=1
 aint(l,l)=(ahigh-alow)*(f(alow)+f(ahigh))/2.
 error=2.*errmax | Just to make algorithm proceed
  do while(l.lt.nromax.and.dabs(error).gt.errmax)
    l=l+1
    nsteps=2**(l-1)
    aint(l,1)=trapez(alow,ahigh,nsteps)
     Do j = 2,l
      aint(l,j)=aint(i,j-1)+(aint(i,j-1)-aint(i-1,j-1))
1                 /(2**(2*(j-1))-1)
     enddo
    best=aint(l,l)
    error=abs((best-aint(l,l-1))/best)
  enddo
 stop
 end
```