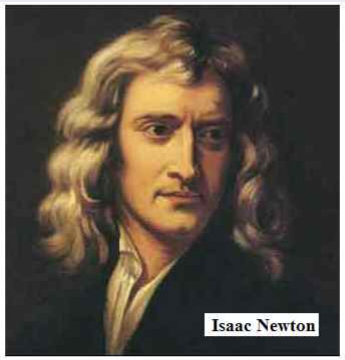
Part 5
Chapter 19

Numerical Integration Formulas

$\int f(x)\, dx$


Isaac Newton


Gottfried Wilhelm Leibniz (1646-1716)

전통적으로 영국인과 독일인들은 서로 라이벌 의식을 가지고 있다。
이와 같은 라이벌 의식은 **17**세기 로 거슬러 올라가도 여전했음을 알 수 있는데, 그것은 영국의 **Newton** 과 독일의 **Leibniz** 의 미적분 발견에 관한 대결이다。

Newton과 **Leibniz**는 각각 독창적으로 미적분을 발견했다。

Leibniz는 **1673**년부터 **1676**년 사이에 미적분을 고안하여 발표했다。

Newton 은 **1660**년 대 후반에 이미 미적분을 발견했으나, 발표를 **1704**년에 광학**(Optics)**이라는 책의 부록에 발표했다。

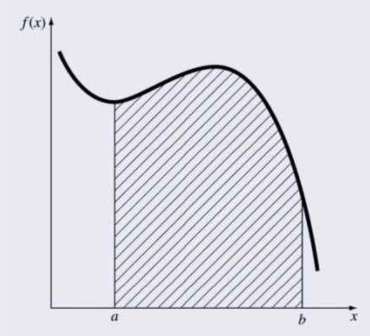
영국인들은 **Leibniz** 의 논문을 보고 **Newton** 의 것을 모방한 것이라고 주장했다。
한편 **Leibniz**의 지지자들은 **Newton** 이 늦게 발표했다는 것을 들어서 오히려 **Newton** 이 표절했다고 주장했다。

상황이 감정 싸움으로 비화해 이후에 수학의 학문적 교류가 끊겼고, 이로 인해 영국 수학의 발전이 **100**년 정도 지연되었다고 한다。

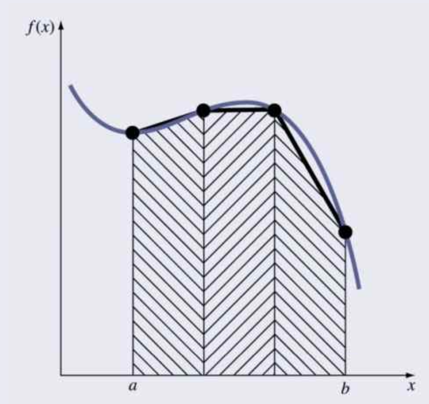
Integration

- Integration is the total value (or summation) of *f(x)dx* over the range from a to b:

$$I = \int_a^b f(x)\,dx$$



$$\Delta x = \frac{b-a}{n}$$



$$\int_a^b f(x)dx = \lim_{n\to\infty} \sum_{m=1}^{n} f(x_i)\Delta x$$

$$\therefore \int_a^b f(x)dx \approx \sum_{m=1}^{N} f(x_i)\Delta x,$$

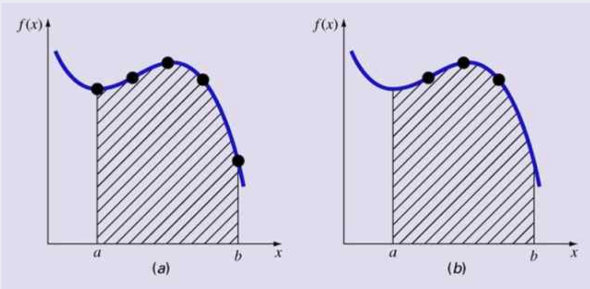
  - To reduce error, *n* must be large (inefficient) → need formula
  - The degree of precision *r* of an integration formula
    - The highest degree of polynomial used for the formula

Newton-Cotes Formulas

- The most common numerical integration schemes.
  - based on replacing a complicated function or tabulated data with
  - a polynomial that is easy to integrate where $f_n(x)$ is an n^{th} order interpolating polynomial.

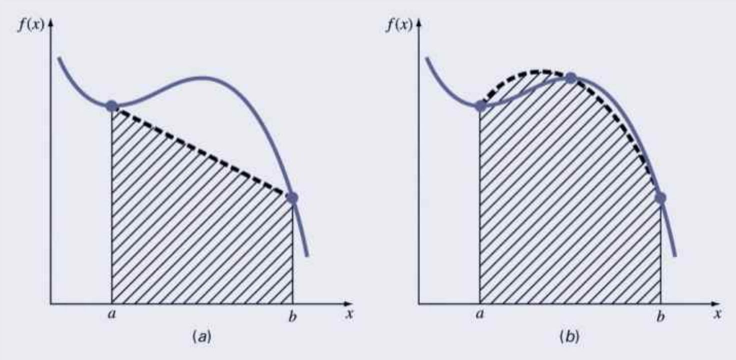
$$I = \int_a^b f(x)\,dx \cong \int_a^b f_n(x)\,dx$$

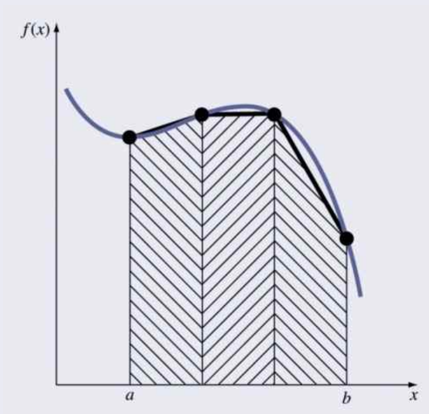
- Newton-Cotes formulas
  - Closed formula : uses the function value at all points
    - Trapezoid(사다리꼴) rule, Simpson rule
  - Open formula : does not use the function values at the endpoints
    - Midpoint rule

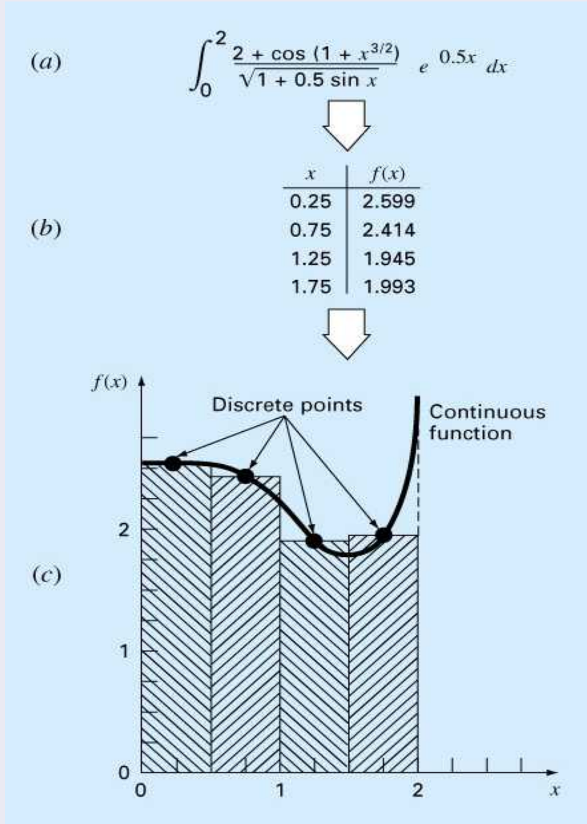


Newton-Cotes Examples

- The integrating function can be polynomials for any order
  - (a) straight line or
  - (b) parabola (포물선, graph of quadratic function)
- The integral can be approximated
  - in one step or
  - in a series of steps to improve accuracy.







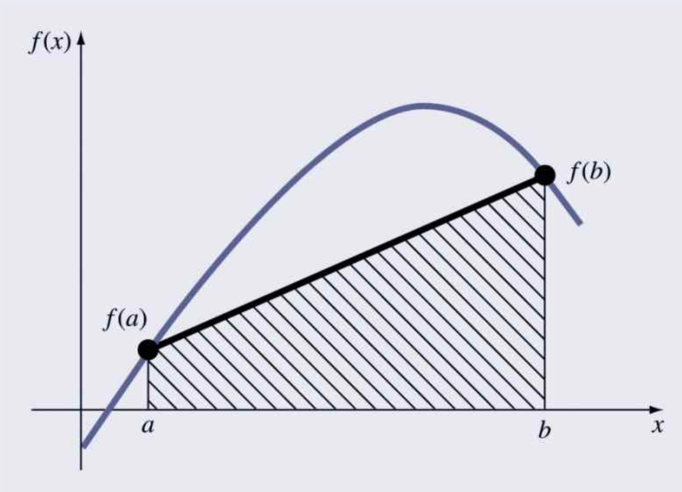
The Trapezoidal Rule

- Uses a straight-line approximation for the function

$$I = \int_a^b f_n(x)\,dx$$

$$I = \int_a^b \left[f(a) + \frac{f(b) - f(a)}{b-a}(x-a) \right] dx$$

$$I = (b-a)\frac{f(a)+f(b)}{2}$$



Error of the Trapezoidal Rule

- Local truncation error of a single application of the trapezoidal rule is:

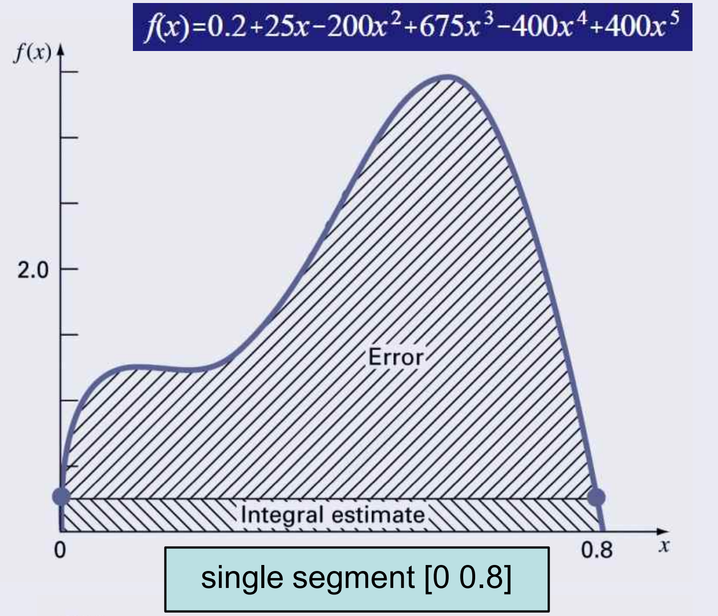
$$E_t = -\frac{1}{12} f''(\xi)(b-a)^3$$

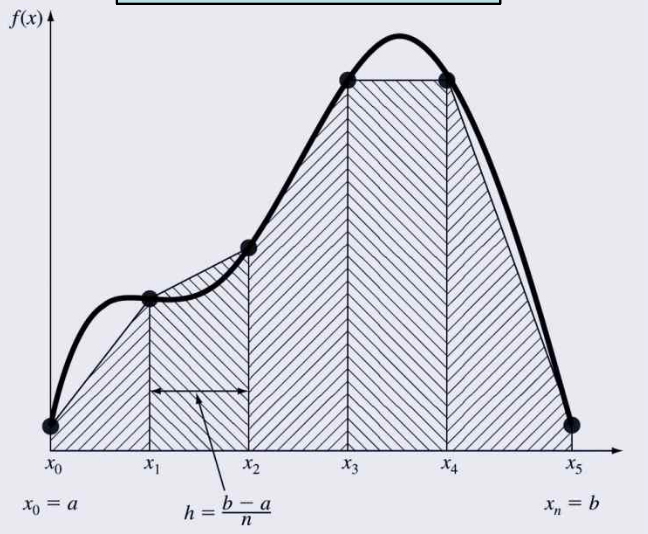
where ξ is somewhere between *a* and *b*.

- Error is dependent upon
  - the curvature of the actual function as well as
  - the distance between the points.

- Error can thus be reduced by breaking the curve into parts.

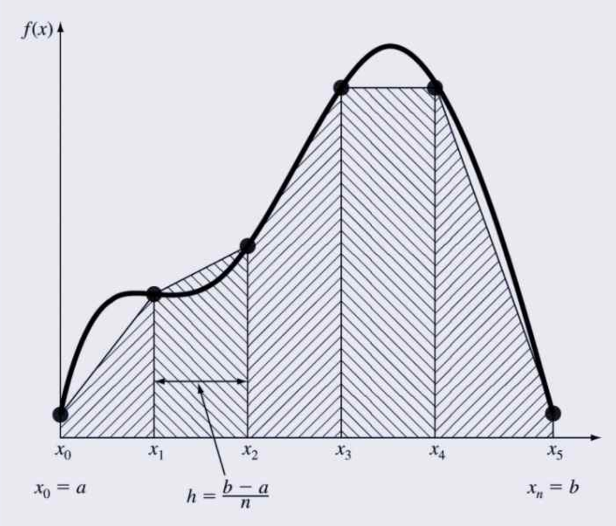
- The degree of precision = 1





Composite Trapezoidal Rule

- Assuming *(n+1)* data points are evenly spaced,
  - *n* intervals over which to integrate.
- The total integral can be calculated by
  - integrating each subinterval and then
  - adding them together:



$$I = \int_{x_0}^{x_n} f_n(x)\,dx = \int_{x_0}^{x_1} f_n(x)\,dx + \int_{x_1}^{x_2} f_n(x)\,dx + \cdots + \int_{x_{n-1}}^{x_n} f_n(x)\,dx$$

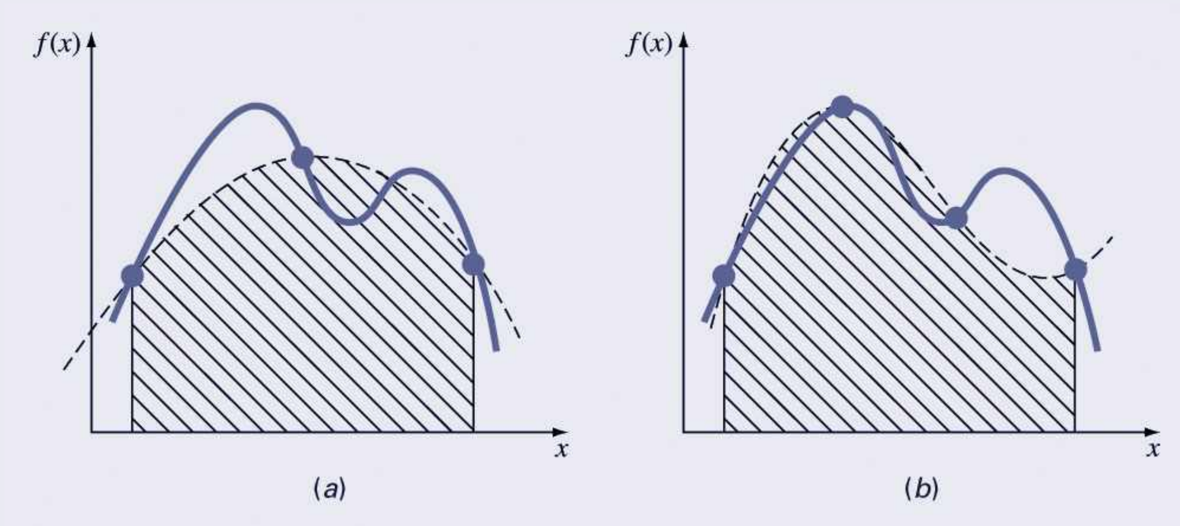
$$I = (x_1 - x_0)\frac{f(x_0)+f(x_1)}{2} + (x_2 - x_1)\frac{f(x_1)+f(x_2)}{2} + \cdots + (x_n - x_{n-1})\frac{f(x_{n-1})+f(x_n)}{2}$$

$$I = \frac{h}{2}\left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n) \right]$$

Simpson's Rules

- Approximation formulas can improve the accuracy using
  - *(a) 2nd* order polynomials
  - *(b) 3rd* order polynomials

- The formulas that result from taking the integrals under these polynomials are called *Simpson's rules*.



Simpson's 1/3 Rule

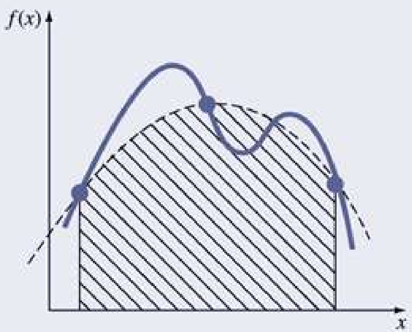
- Simpson's 1/3 rule corresponds to using second-order polynomials.
- Using the Lagrange form for a quadratic fit of three points
  - degree of precision : 2

$$f_n(x)=\frac{(x-x_1)}{(x_0-x_1)}\frac{(x-x_2)}{(x_0-x_2)}f(x_0)+\frac{(x-x_0)}{(x_1-x_0)}\frac{(x-x_2)}{(x_1-x_2)}f(x_1)+\frac{(x-x_0)}{(x_2-x_0)}\frac{(x-x_1)}{(x_2-x_1)}f(x_2)$$

  - Integration over the three points simplifies to:

$$I=\int_{x_0}^{x_2} f_n(x)\,dx$$

$$I=\frac{h}{3}\left[f(x_0)+4f(x_1)+f(x_2)\right]$$



(a)

$$L(x)=\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0)+\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1)+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2)$$

$$\int_a^b f(x)dx \approx \int_a^b \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}f(x_0)+\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}f(x_1)+\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}f(x_2)dx$$

$$=\int_a^b \frac{(x-x_1)(x-x_2)}{2h^2}f(x_0)dx-\int_a^b \frac{(x-x_0)(x-x_2)}{h^2}f(x_1)dx+\int_a^b \frac{(x-x_0)(x-x_1)}{2h^2}f(x_2)dx$$

$$=....$$

$$g(x) = f_o V_o(x) + f_1 V_1(x) + f_2 V_2(x)$$

$$V_o(x) = \frac{(x-x_1)(x-x_2)}{(x_o-x_1)(x_o-x_2)} \Rightarrow V_o(x) = \frac{x^2-3hx+2h^2}{2h^2}$$

$$V_1(x) = \frac{(x-x_o)(x-x_2)}{(x_1-x_o)(x_1-x_2)} \Rightarrow V_1(x) = \frac{4hx-2x^2}{2h^2}$$

$$V_2(x) = \frac{(x-x_o)(x-x_1)}{(x_2-x_o)(x_2-x_1)} \Rightarrow V_2(x) = \frac{x^2-hx}{2h^2}$$

$$I = \int_{x_o}^{x_2} f(x) \quad \Rightarrow \quad I = \int_{x_o}^{x_2} g(x)dx + E \quad \Rightarrow$$

$$I = \int_{x_o=0}^{x_2=2h} \left\{ f_o\left[\frac{x^2-3hx+2h^2}{2h^2}\right] + f_1\left[\frac{4hx-2x^2}{2h^2}\right] + f_2\left[\frac{x^2-hx}{2h^2}\right]\right\} dx + E \quad \Rightarrow$$

$$I = \frac{1}{2h^2}\left[f_o\left(\frac{x^3}{3}-\frac{3hx^2}{2}+2h^2x\right)+f_1\left(\frac{4hx^2}{2}-\frac{2x^3}{3}\right)+f_2\left(\frac{x^3}{3}-h\frac{x^2}{2}\right)\right]_0^{2h} + E \quad \Rightarrow$$

$$I = \frac{1}{2h^2}\left[f_o\left(8\frac{h^3}{3}-\frac{12h^3}{2}+4h^3-0\right)+f_1\left(8h^3-\frac{16}{3}h^3\right)+f_2\left(\frac{8h^3}{3}-h\frac{4h^2}{2}\right)\right] + E \quad \Rightarrow$$

$$I = \frac{h}{3}[f_o + 4f_1 + f_2] + E$$

Error of Simpson's 1/3 Rule

- An estimate for the local truncation error of a single application of Simpson's 1/3 rule is:

$$E_t = -\frac{1}{2880} f^{(4)}(\xi)(b-a)^5$$

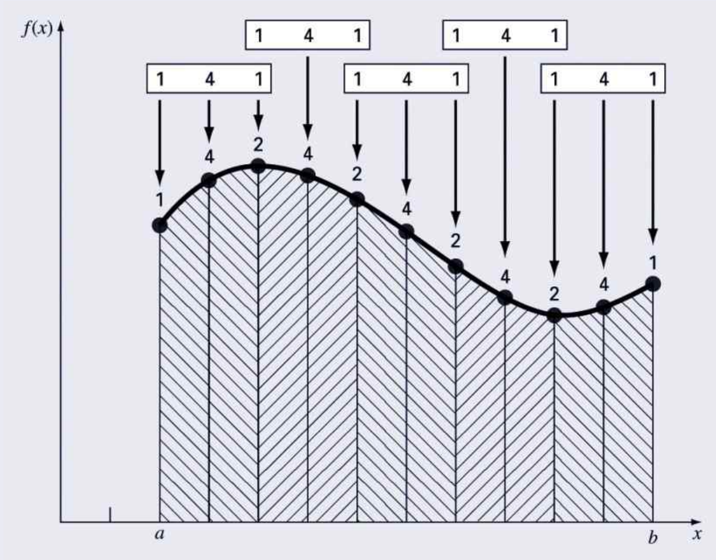
where again ξ is somewhere between *a* and *b*.

- Error is dependent upon
  - the fourth-derivative of the actual function as well as
  - the distance between the points.
    - Error is dependent on the fifth power of the step size
    - (cf.) the third for the trapezoidal rule

- Error can thus be reduced by breaking the curve into parts.

Composite Simpson's 1/3 Rule

- Simpson's 1/3 rule can be used on a set of subintervals.
  - *must* be an odd number of points (even number of intervals)



$$I = \int_{x_0}^{x_n} f_n(x)\,dx = \int_{x_0}^{x_2} f_n(x)\,dx + \int_{x_2}^{x_4} f_n(x)\,dx + \cdots + \int_{x_{n-2}}^{x_n} f_n(x)\,dx$$

$$I = \frac{h}{3}\left[f(x_0)+4f(x_1)+f(x_2)\right]+\frac{h}{3}\left[f(x_2)+4f(x_3)+f(x_4)\right]+\cdots+\frac{h}{3}\left[f(x_{n-2})+4f(x_{n-1})+f(x_n)\right]$$

$$I = \frac{h}{3}\left[f(x_0)+4\sum_{\substack{i=1\\ i,\ \text{odd}}}^{n-1} f(x_i)+2\sum_{\substack{j=2\\ j,\ \text{even}}}^{n-2} f(x_i)+f(x_n)\right]$$

Simpson's 3/8 Rule

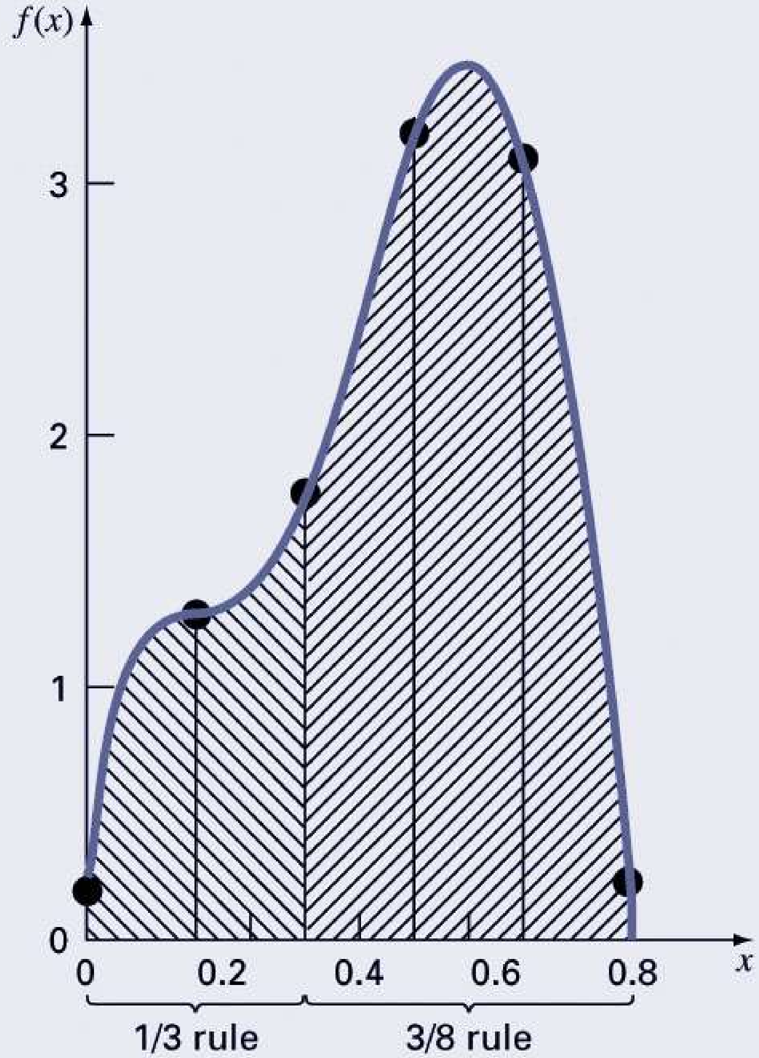
- Simpson's 3/8 rule corresponds to using
  – third-order polynomials to fit four points (degree of precision = 3)

- Integration over the four points simplifies to:

$$I = \int_{x_0}^{x_3} f_n(x)\,dx$$

$$I = \frac{3h}{8}\left[f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)\right]$$

- Simpson's 3/8 rule is generally used
  – in concert with Simpson's 1/3 rule
  – when the number of segments is odd.



$$L(x)=\frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}f(x_0)+\frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)}f(x_1)$$

$$+\frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)}f(x_2)+\frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}f(x_3)$$

$$\int_a^b f(x)dx \approx \int_a^b L(x)dx$$

$$=-\int_a^b \frac{(x-x_1)(x-x_2)(x-x_3)}{6h^3}f(x_0)dx+\int_a^b \frac{(x-x_0)(x-x_2)(x-x_3)}{2h^3}f(x_1)dx$$

$$-\int_a^b \frac{(x-x_0)(x-x_1)(x-x_3)}{2h^3}f(x_2)dx+\int_a^b \frac{(x-x_0)(x-x_1)(x-x_2)}{6h^3}f(x_3)dx$$

$$=....$$

Example

$$f(x) = 0.2 + 25x - 200x^2 + 675x^3 - 400x^4 + 400x^5$$

- Exact solution : *1.640533*
  - Trapezoidal : *1.570265.* $\varepsilon_t = 4.283241\%$
  - 1/3 Simpson : *1.637162.* $\varepsilon_t = 0.205473\%$
  - 3/8 Simpson : *1.632948.* $\varepsilon_t = 0.462341\%$

Higher-Order Formulas

- Higher-order Newton-Cotes formulas may also be used
  - The higher the order of the polynomial used, smaller error.
    - the higher the derivative of the function in the error estimate
      - the more complicated
    - the higher the power of the step size
      - the more restricted by the number of points
  - Most used: 1/3 or 3/8 rules.

Newton-Cotes closed integration formulas.

Points	Name	Formula	Truncation Error
2	Trapezoidal rule	$(b-a)\dfrac{f(x_0)+f(x_1)}{2}$	$-(1/12)h^3 f''(\xi)$
3	Simpson's 1/3 rule	$(b-a)\dfrac{f(x_0)+4f(x_1)+f(x_2)}{6}$	$-(1/90)h^5 f^{(4)}(\xi)$
4	Simpson's 3/8 rule	$(b-a)\dfrac{f(x_0)+3f(x_1)+3f(x_2)+f(x_3)}{8}$	$-(3/80)h^5 f^{(4)}(\xi)$
5	Boole's rule	$(b-a)\dfrac{7f(x_0)+32f(x_1)+12f(x_2)+32f(x_3)+7f(x_4)}{90}$	$-(8/945)h^7 f^{(6)}(\xi)$
6		$(b-a)\dfrac{19f(x_0)+75f(x_1)+50f(x_2)+50f(x_3)+75f(x_4)+19f(x_5)}{288}$	$-(275/12{,}096)h^7 f^{(6)}(\xi)$

Open Method (Midpoint Rule)

- Newton-Cotes closed formula (use all points)

$$\int_a^b f(x)dx \approx (b-a)f(\frac{a+b}{2}) \qquad \int_a^b f(x)dx = (b-a)f(\frac{a+b}{2}) - \frac{(b-a)^3}{12}f''(\eta)$$

- Newton-Cotes open formula using two or three function evaluations (does not use end points)

$$\int_a^b f(x)dx = \frac{b-a}{2}\left[f(x_1)+f(x_2)\right]+\frac{3h^3}{4}f''(\eta) \qquad where\ \ x_1 = \frac{2a+b}{3}, x_2 = \frac{a+2b}{3}$$

$$\int_a^b f(x)dx = \frac{4h}{3}\left[2f(x_1)-f(x_2)+2f(x_3)\right]+\frac{14h^5}{45}f^{(4)}(\eta)$$

$$where\ \ x_1 = \frac{3a+b}{4}, x_2 = \frac{2a+2b}{4}, x_3 = \frac{a+3b}{4}, h = \frac{b-a}{4}$$

Open Method (Midpoint Rule)

Newton-Cotes open integration formulas $h = (b-a)/n$

Segments (n)	Points	Name	Formula	Truncation Error
2	1	Midpoint method	$(b-a)f(x_1)$	$(1/3)h^3 f''(\xi)$
3	2		$(b-a)\dfrac{f(x_1)+f(x_2)}{2}$	$(3/4)h^3 f''(\xi)$
4	3		$(b-a)\dfrac{2f(x_1)-f(x_2)+2f(x_3)}{3}$	$(14/45)h^5 f^{(4)}(\xi)$
5	4		$(b-a)\dfrac{11f(x_1)+f(x_2)+f(x_3)+11f(x_4)}{24}$	$(95/144)h^5 f^{(4)}(\xi)$
6	5		$(b-a)\dfrac{11f(x_1)-14f(x_2)+26f(x_3)-14f(x_4)+11f(x_5)}{20}$	$(41/140)h^7 f^{(6)}(\xi)$

Integration with Unequal Segments

- Previous formulas were simplified based on equispaced data points, though
  - this is not always the case.

- The trapezoidal rule may be used with data containing unequal segments:

$$I = \int_{x_0}^{x_n} f_n(x)\,dx = \int_{x_0}^{x_1} f_n(x)\,dx + \int_{x_1}^{x_2} f_n(x)\,dx + \cdots + \int_{x_{n-1}}^{x_n} f_n(x)\,dx$$

$$I = (x_1 - x_0)\frac{f(x_0)+f(x_1)}{2} + (x_2 - x_1)\frac{f(x_1)+f(x_2)}{2} + \cdots + (x_n - x_{n-1})\frac{f(x_{n-1})+f(x_n)}{2}$$

MATLAB Functions

- MATLAB functions to evaluate integrals based on the trapezoidal rule

```
z = trapz(y)
z = trapz(x, y)
```

  – produces the integral of y with respect to x
    - if x is omitted, the program assumes *h*=1
    - if y is a vector, trapz(y) is the integral of y.
    - if y is a matrix, trapz(y) is a row vector with the integral over each column.

The exact value of $\int_0^{\pi} \sin(x)dx$ is 2.

To approximate this numerically on a uniformly spaced grid, use

```
X = 0:pi/100:pi;
Y = sin(X);
```

Then both

`Z = trapz(X,Y)` and `Z = pi/100*trapz(Y)` h=1

produce

```
Z =
    1.9998
```

MATLAB Cumulative Integral

- **Cumulative sum :** B = cumsum(A) or B = cumsum(A,dim)
  - B = cumsum(A) returns the cumulative sum along different dimensions of an array.
    - If A is a vector, cumsum(A) returns a vector containing the cumulative sum of the elements of A.
    - If A is a matrix, cumsum(A) returns a matrix the same size as A containing the cumulative sums for each column of A.
  - B = cumsum(A,dim) returns the cumulative sum of the elements along the dimension of A specified by scalar dim.
    - cumsum(A,1) works along the first dimension (row cumsum)
    - cumsum(A,2) works along the second dimension (column cumsum)

```
cumsum(1:5)   ans =
                     [1   3   6   10   15]
```

[1 2 3 4 5]

```
A = [1 2 3; 4 5 6];

cumsum(A,1)
ans =
        1     2     3
        5     7     9

cumsum(A,2)
ans =
        1     3     6
        4     9    15
```

MATLAB Cumulative Integral

- **Z = cumtrapz(Y) or Z = cumtrapz(X,Y)**
  - Computes an approximation of the cumulative integral of Y via the trapezoidal method with unit spacing
  - To compute the integral with other than unit spacing, multiply Z by the spacing increment.

```
Y = [0 1 2; 3 4 5];

cumtrapz(Y,1)
ans =
0         0         0
     1.5000    2.5000    3.5000

cumtrapz(Y,2)
ans =
0    0.5000    2.0000
          0    3.5000    8.0000
```

$$I = (b-a)\frac{f(a)+f(b)}{2}$$

0 1 2	**0**	**0**	**0**
3 4 5	(0+3)/2	(1+4)/2	(2+5)/2

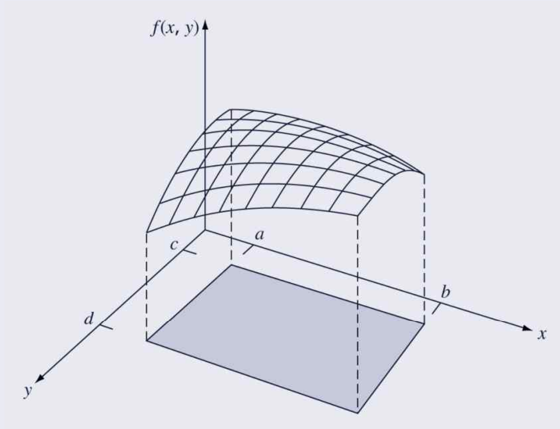
0 1 2	**0**	(0+1)/2	(1+2)/2
3 4 5	**0**	(3+4)/2	(4+5)/2

0	0.5	1.5		**0**	0.5	(0.5+1.5)
0	3.5	4.5		**0**	3.5	(3.5+4.5)

$$I = (x_1 - x_0)\frac{f(x_0)+f(x_1)}{2} + (x_2 - x_1)\frac{f(x_1)+f(x_2)}{2} + \cdots + (x_n - x_{n-1})\frac{f(x_{n-1})+f(x_n)}{2}$$

Multiple Integrals

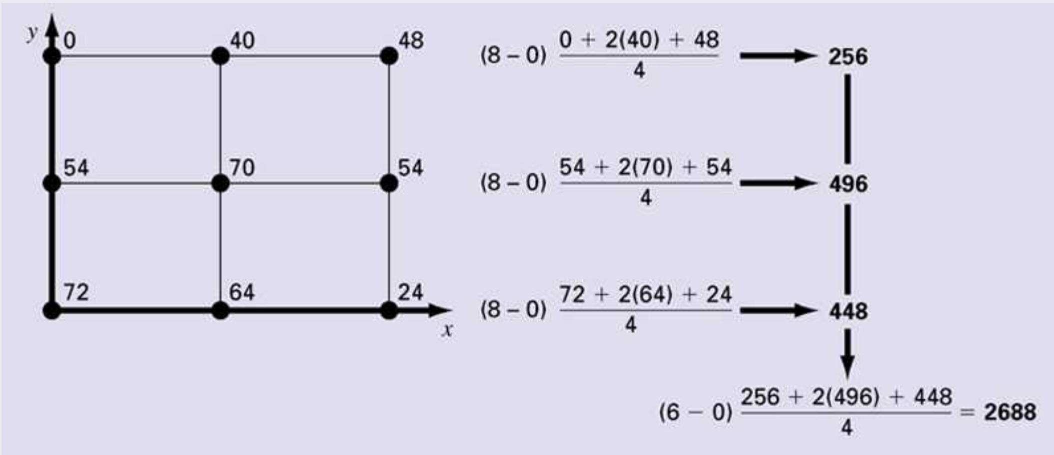
- Multiple integrals can be determined numerically by
  - first integrating in one dimension,
  - then a second, and so on
  - for all dimensions of the problem.



$$\int_c^d \left(\int_a^b f(x,y)dx\right)dy$$

$$f(x,y) = 2xy + 2x - x^2 - 2y^2 + 40$$

$$\int_0^6 \left(\int_0^8 f(x,y)dx\right)dy$$



MATLAB Functions

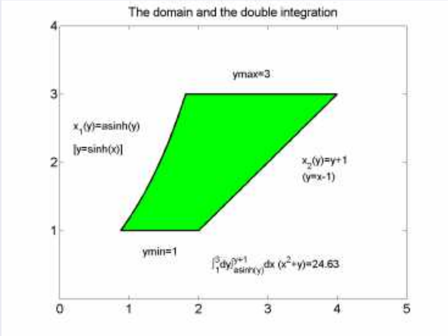
- dblquad and triplequad
  - integration for 2- and 3-dimension

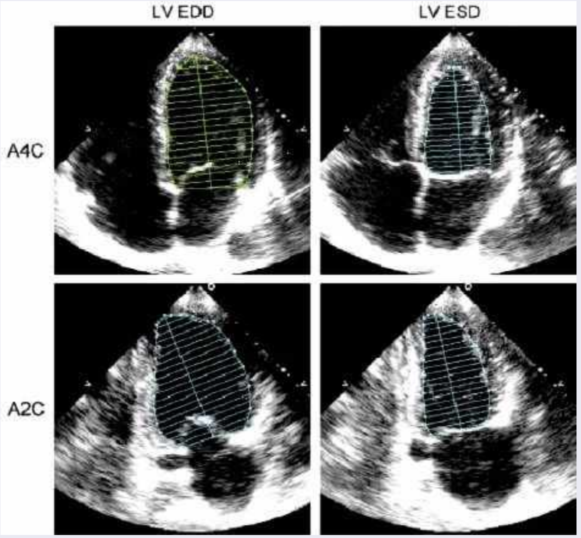
$$Z = \int\int e^{-V(x,y)/kT}\,dx\,dy$$

  - q = dblequad(fun, xmin, xmax, ymin, ymax, tol)
    - xmin, xmax : range of x
    - ymin, xmax : range of y
    - tol : tolerance. (relative error, default : *1e-6)*

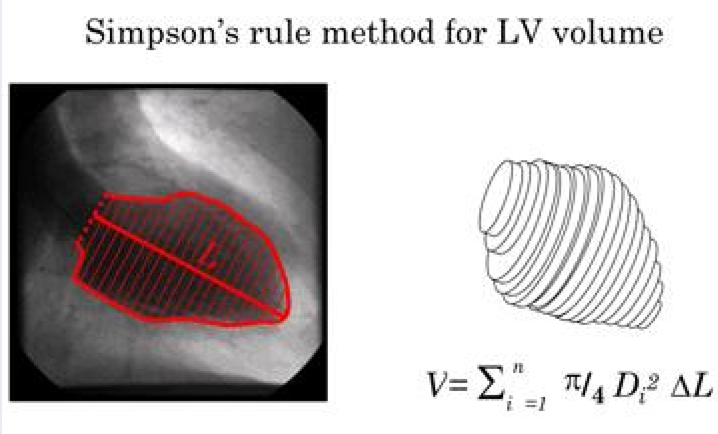
Integrate $y\sin(x)+x\cos(y)$ over $\pi \le x \le 2\pi$, $0 \le y \le \pi$. The true value of the integral is $-\pi^2$

```
Q = quad2d(@(x,y) y.*sin(x)+x.*cos(y),pi,2*pi,0,pi)
```

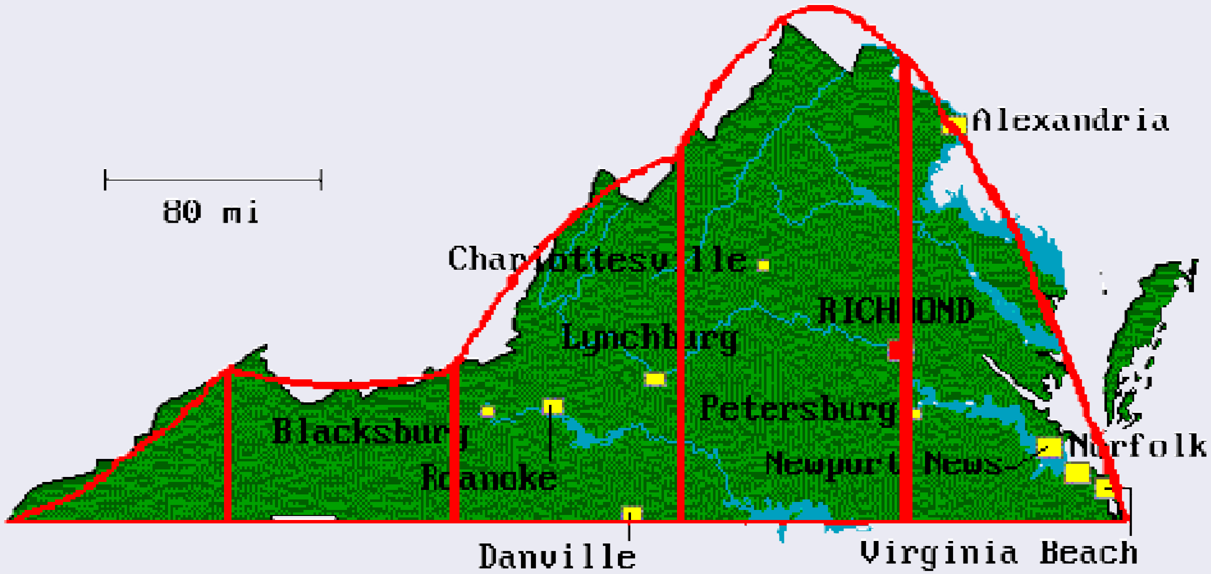




LV EDD
LV ESD
A4C
A2C



Simpson's rule method for LV volume
$V = \sum_{i=1}^{n} \pi/4\, D_i^2\, \Delta L$



80 mi
Alexandria
Charlottesville
Lynchburg
RICHMOND
Blacksburg
Roanoke
Petersburg
Newport News
Norfolk
Danville
Virginia Beach

THE END

Report : 19-4, 19-6, 19-8

Part 5
Chapter 18

Numerical Integration of Functions

Integration of Function

- A function is given either of the two forms
  - Table of values : *(1, 3),(2, 7),(3.5, 1), (5.1, -3)*
  - Function *: $f(x)=x^2$*
    - we can generate as many values of f(x) as are required to attain acceptable accuracy
    - The integration techniques based on this ability
      - Romberg integration
      - Guass quadrature
      - Adaptive quadrature

Error in Multiple Segment Trapezoidal Rule

- The true error gets approximately quartered(1/4) as the number of segments is doubled(2).

$$E(h) \approx -\frac{1}{12}(b-a)h^2 \bar{f}'' = c \cdot h^2$$

  - This information is used to get a better approximation of the integral, and is the basis of Richardson's extrapolation.

- Richardson and Romberg Integration is an extrapolation formula of the Trapezoidal Rule for integration.
  - It provides a better approximation of the integral by reducing the True Error.

Richardson Extrapolation

- Start with

$$I = I(h_1) + E(h_1) = I(h_2) + E(h_2)$$

I : true value

  – where $E(h_1)$ and $E(h_2)$ are the true errors using a different multiple-segment implementation of the trapezoidal rule.
  – As we know, this error is given approximately by the relation

$$E(h) \approx -\frac{1}{12}(b-a)h^2 \bar{f}'' = c \cdot h^2$$

- we can solve for it

$$I = I(h_1) + ch_1^2 = I(h_2) + ch_2^2 \quad \Rightarrow \quad c = \frac{I(h_2) - I(h_1)}{h_1^2 - h_2^2} = \frac{1}{h_2^2}\left[\frac{I(h_2) - I(h_1)}{(h_1/h_2)^2 - 1}\right]$$

- Using c, we get an improved estimate for *I*

$$\boxed{I = I(h_2) + \left[\frac{I(h_2) - I(h_1)}{(h_1/h_2)^2 - 1}\right]}$$

Richardson Extrapolation

- For the special case where *h1 = 2*h2*

$$\boxed{I = I(h_2) + \left[\frac{I(h_2) - I(h_1)}{(h_1 / h_2)^2 - 1} \right]}$$

h1/h2 = 2

$$\boxed{I \approx \frac{4}{3} I(h_2) - \frac{1}{3} I(h_1)}$$

  - The error of the estimate given by this expression is $O(h_2^4)$

- The expression above yields a formula that is identical to that obtained using Simpson's 1/3 rule

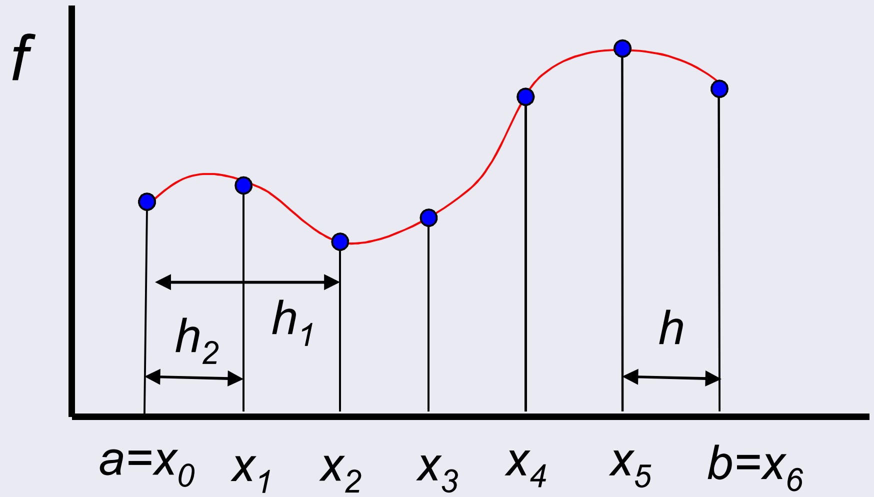
Trapezoidal Rule

2h

$$I(h_1) = \frac{h_1}{2}\left(f(x_0) + 2f(x_2) + 2f(x_4) + f(x_6)\right)$$

$$I(h_2) = \frac{h_2}{2}\begin{pmatrix} f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) \\ + 2f(x_4) + 2f(x_5) + f(x_6) \end{pmatrix}$$

h



Richardson extrapolation: $$I = \frac{4}{3}I(h_2) - \frac{1}{3}I(h_1)$$

$$I = \frac{2h}{3}\left(f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + 2f(x_4) + 2f(x_5) + f(x_6)\right)$$

$$-\frac{h}{3}\left(f(x_0) + 2f(x_2) + 2f(x_4) + f(x_6)\right)$$

$$\boxed{I = \frac{h}{3}\left(f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + 4f(x_5) + f(x_6)\right)}$$

Simpson's 1/3 Rule !!

Richardson Extrapolation (cont)

- For the cases where there are two $O(h^4)$ estimates and the interval is halved ($h_m = h_l/2$),
  - an improved $O(h^6)$ estimate may be formed using:

$$I = \frac{16}{15} I_m - \frac{1}{15} I_l$$

- For the cases where there are two $O(h^6)$ estimates and the interval is halved ($h_m = h_l/2$),
  - an improved $O(h^8)$ estimate may be formed using:

$$I = \frac{64}{63} I_m - \frac{1}{63} I_l$$

Rinchardson's extrapolation

- A method to combine integrals to obtain improved estimates

$$I = \frac{4}{3} I_m - \frac{1}{3} I_l$$

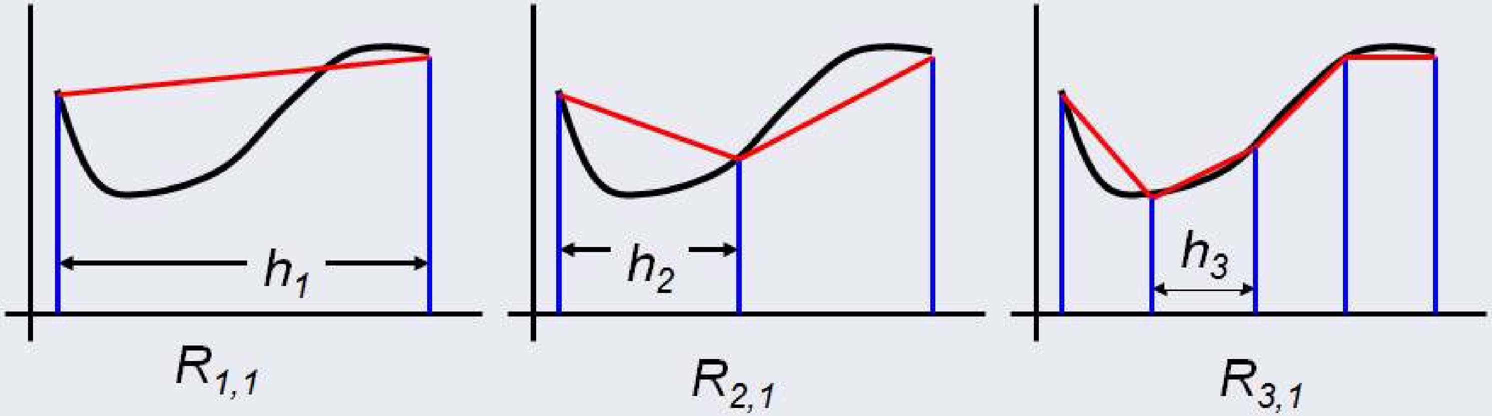
$$I = \frac{16}{15} I_m - \frac{1}{15} I_l$$

$$I = \frac{64}{63} I_m - \frac{1}{63} I_l$$

- Example : estimates of integral using composite trapezoidal rule

Segments	*h*	*Integral*	*error*
1	*.8*	*0.1728*	*89.5%*
2	*.4*	*1.0688*	*34.9%*
4	*.2*	*1.4848*	*9.5%*

Romberg integration



$$R_{1,1} = \frac{h_1}{2}[f(a)+f(b)] = \frac{b-a}{2}[f(a)+f(b)]$$

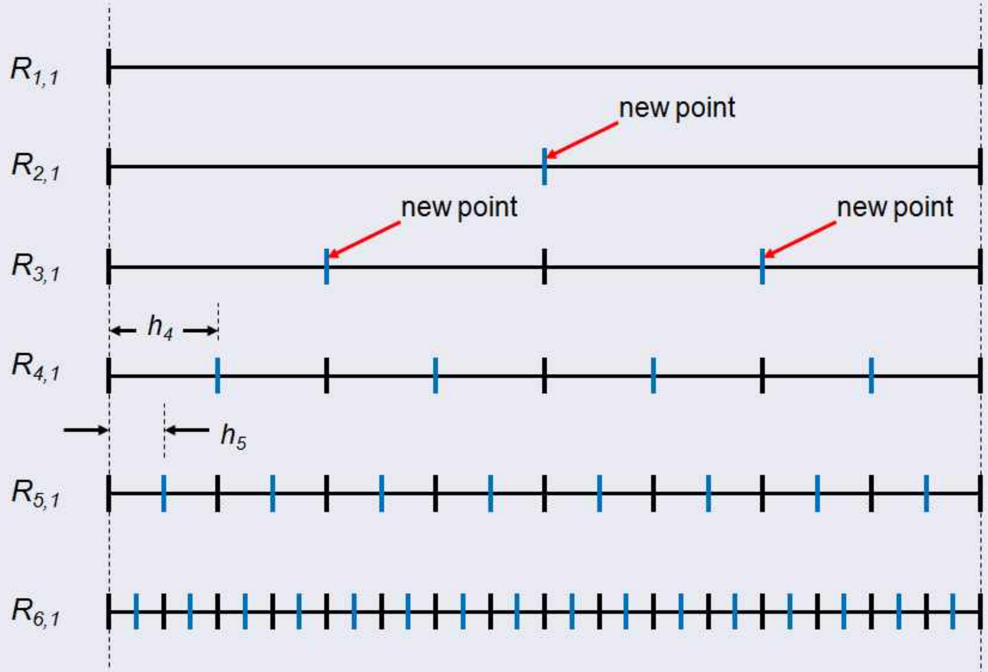
$$R_{2,1} = \frac{h_2}{2}[f(a)+f(b)+2*f(a+h_2)]$$

$$R_{2,1} = \underbrace{\frac{h_2}{2}[f(a)+f(b)]}_{R_{1,1}/2}+[\underbrace{h_2}_{h_1/2} f(a+h_2)] = \frac{1}{2}[R_{1,1}+h_1 f(a+h_2)]$$

$$R_{3,1} = \underbrace{\frac{h_3}{2}[f(a)+f(b)+2f(a+2h_3)]}_{R_{2,1}/2} + \underbrace{h_3}_{h_2/2}[f(a+h_3)+f(a+3h_3)]$$

$$R_{3,1} = \frac{1}{2}\left\{R_{2,1} + h_2[f(a+h_3)+f(a+3h_3)]\right\}$$

$$\boxed{R_{k,1} = \frac{1}{2}\left\{R_{k-1,1} + h_{k-1}\sum_{i=1}^{2^{k-2}} f(a+\overbrace{(2i-1)h_k}^{\text{new (unused) points, odd multiples of } h_k})\right\}}$$



Define:

$$\boxed{R_{k,j} = R_{k,j-1} + \frac{R_{k,j-1} - R_{k-1,j-1}}{4^{j-1} - 1}}$$

Construct Romberg Table

Obtain from a single composite trapezoidal integration

$$\begin{array}{cccccc}
R_{1,1} & & & & & \\
R_{2,1} & R_{2,2} & & & & \\
R_{3,1} & R_{3,2} & R_{3,3} & & & \\
R_{4,1} & R_{4,2} & R_{4,3} & R_{4,4} & & \\
\vdots & \vdots & \vdots & \vdots & & \\
R_{n,1} & R_{n,2} & R_{n,3} & R_{n,4} & \cdots & R_{n,n}
\end{array}$$

$\boxed{I = R_{n,n} + O(h_n^{2n})}$

Example $\int_0^{\pi} \sin x \, dx = 2$

0.00000000					
1.57079633	2.09439511				
1.89611890	2.00455976	1.99857073			
1.97423160	2.00026917	1.99998313	2.00000555		
1.99357034	2.00001659	1.99999975	2.00000001	1.99999999	
1.99839336	2.00000103	2.00000000	2.00000000	2.00000000	2.00000000

accurate to $O(h_6^{12})$

Error in $R_{6,6}$ is only *6.61026789e-011* % !!!!

$h_6 = \pi/32 = 9.817e\text{-}002$, $h_6^{12} = 8.017e\text{-}013$

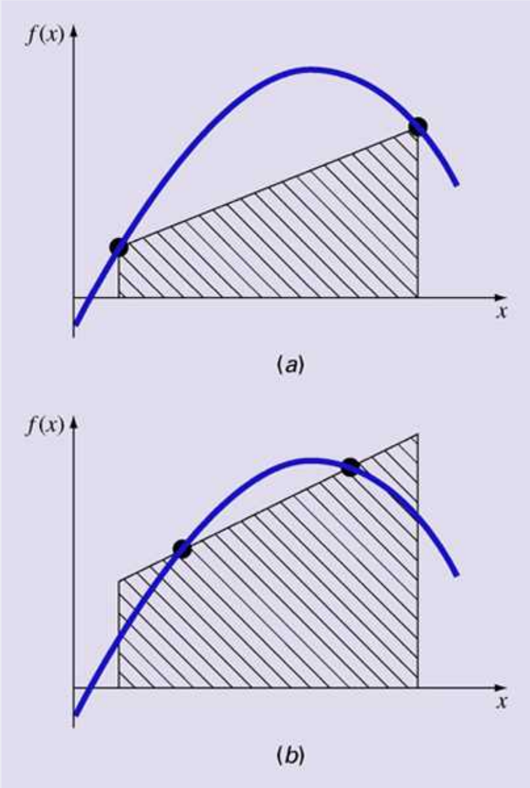
Gauss Quadrature (구적법 *求積法*)

- Gauss quadrature describes a class of techniques for
  - evaluating the area under a straight line
  - by joining any two points
  - rather than simply choosing the endpoints.

  - Variable sampling position instead of fixed or equal spaced.

- To be determined
  - Sampling points
  - Weighting at each sampling points.
  - The key is to choose the line that balances the positive and negative errors.

- Better accuracy than Simpson's rules under the same number of sampling points.



Gauss-Legendre Formulas

- Fix integral intervals from -1 to 1.
  - Integrals over other intervals require a change in variables to set the limits from -1 to 1.

- The integral estimates are of the form:

$$I \cong c_0 f(x_0) + c_1 f(x_1) + \cdots + c_{n-1} f(x_{n-1})$$

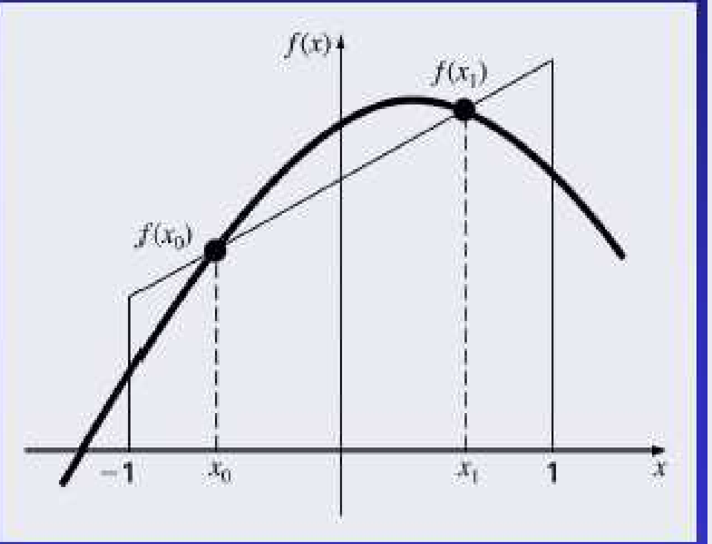
where the c_i and x_i are to be determined to ensure exact solution up to $(2n\text{-}1)^{\text{th}}$ order polynomials over the interval from -1 to 1.

(ex) Two-Point Gauss-Legendre Formula

- Let $I \approx c_0 f(x_0) + c_1 f(x_1)$
- Goal: determine the unknowns c_0, c_1, x_0, x_1.
- Method: Use 4 conditions to solve 4 unknowns, that is I gives the correct result of

$$I = \int_{-1}^{1} f(x)dx$$

when $f(x)=1$, $f(x)=x$, $f(x)=x^2$ and $f(x)=x^3$



(ex) Two-Point Gauss-Legendre Formula (cont.)

$$I \approx c_0 f(x_0) + c_1 f(x_1)$$

$$I = \int_{-1}^{1} f(x)\,dx$$

$$f(x)=1,\ f(x)=x,\ f(x)=x^2 \text{ and } f(x)=x^3$$

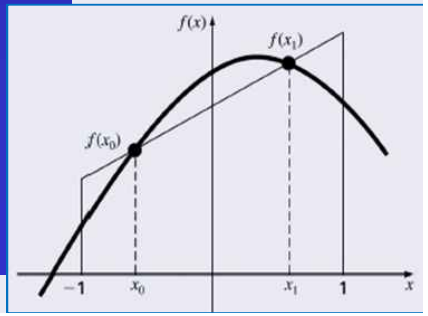
$$c_0 + c_1 = \int_{-1}^{1} 1\,dx = 2$$

$$c_0 x_0 + c_1 x_1 = \int_{-1}^{1} x\,dx = 0$$

$$c_0 x_0^2 + c_1 x_1^2 = \int_{-1}^{1} x^2\,dx = \frac{2}{3}$$

$$c_0 x_0^3 + c_1 x_1^3 = \int_{-1}^{1} x^3\,dx = 0$$

$$\Rightarrow c_0 = c_1 = 1,\ x_0 = -\frac{1}{\sqrt{3}},\ x_1 = \frac{1}{\sqrt{3}}$$



Change of Variables

- For an arbitrary integral interval [*a, b*], let

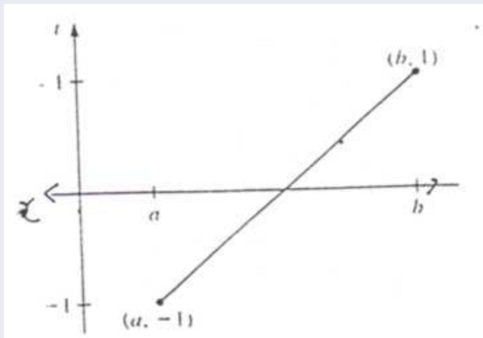
$$x = \frac{(b+a)+(b-a)x_d}{2} \Rightarrow dx = \frac{b-a}{2}\, dx_d$$

$$\therefore \int_a^b f(x)\, dx = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{(b+a)+(b-a)x_d}{2}\right) dx_d$$

$$\int_a^b f(t)dt = \int_{-1}^{1} f\left(\frac{(b-a)x+b+a}{2}\right)\frac{b-a}{2}dx$$

- To sum up:
  - Can only be used to integrate a known function
  - Find the new sampling points by the change of variable formula.
  - Compute the integral by timing the weighting coefficients.
  - Multiply the result by *(b-a)/2*

임의의 구간 [a, b]에서의 적분 ⇒ [-1,1]구간에서의 적분



Gauss Quadrature

Gaussian quadrature $\int_{-1}^{1} f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i)$

Number of points, n	Points, x_i	Weights, w_i
1	0	2
2	$\pm\sqrt{1/3}$	1
3	0	$8/9$
	$\pm\sqrt{3/5}$	$5/9$
4	$\pm\sqrt{\left(3-2\sqrt{6/5}\right)/7}$	$\frac{18+\sqrt{30}}{36}$
	$\pm\sqrt{\left(3+2\sqrt{6/5}\right)/7}$	$\frac{18-\sqrt{30}}{36}$
5	0	$128/225$
	$\pm\frac{1}{3}\sqrt{5-2\sqrt{10/7}}$	$\frac{322+13\sqrt{70}}{900}$
	$\pm\frac{1}{3}\sqrt{5+2\sqrt{10/7}}$	$\frac{322-13\sqrt{70}}{900}$

n	근 $r_{n,i}$	계수 $c_{n,i}$
2	0.5773502692	1.0000000000
	−0.5773502692	1.0000000000
3	0.7745966692	0.5555555556
	0.0000000000	0.8888888889
	−0.7745966692	0.5555555556
4	0.8611363116	0.3478548451
	0.3399810436	0.6521451549
	−0.3399810436	0.6521451549
	−0.8611363116	0.3478548451
5	0.9061798459	0.2369268850
	0.5384693101	0.4786286705
	0.0000000000	0.5688888889
	−0.5384693101	0.4786286705
	−0.9061798459	0.2369268850

(ex) $\int_{1}^{1.5} e^{-x^2} dx$ (소수 7자리 정확한 값 : 0.1093643)

$$\int_a^b f(t)dt = \int_{-1}^{1} f\left(\frac{(b-a)x+b+a}{2}\right)\frac{b-a}{2}dx$$

n	근 $r_{n,i}$	계수 $c_{n,i}$
2	0.5773502692	1.0000000000
	−0.5773502692	1.0000000000
3	0.7745966692	0.5555555556
	0.0000000000	0.8888888889
	−0.7745966692	0.5555555556
4	0.8611363116	0.3478548451
	0.3399810436	0.6521451549
	−0.3399810436	0.6521451549
	−0.8611363116	0.3478548451
5	0.9061798459	0.2369268850
	0.5384693101	0.4786286705
	0.0000000000	0.5688888889
	−0.5384693101	0.4786286705
	−0.9061798459	0.2369268850

$$\int_{1}^{1.5} e^{-x^2} dx = \frac{1}{4}\int_{-1}^{1} e^{(-(t+5)^2/16)} dt$$

$n = 2$:

$$\int_{1}^{1.5} e^{-x^2} dx \approx \frac{1}{4}[e^{-(5+0.5773502692)^2/16} + e^{-(5-0.5773502692)^2/16}] = 0.1094003$$

$n = 3$:

$$\int_{1}^{1.5} e^{-x^2} dx \approx \frac{1}{4}[(0.5555555556)e^{-(5+0.7745966692)^2/16} + (0.8888888889)e^{-(5)^2/16}$$
$$+ (0.5555555556)e^{-(5-0.7745966692)^2/16}]$$
$$= 0.1093642$$

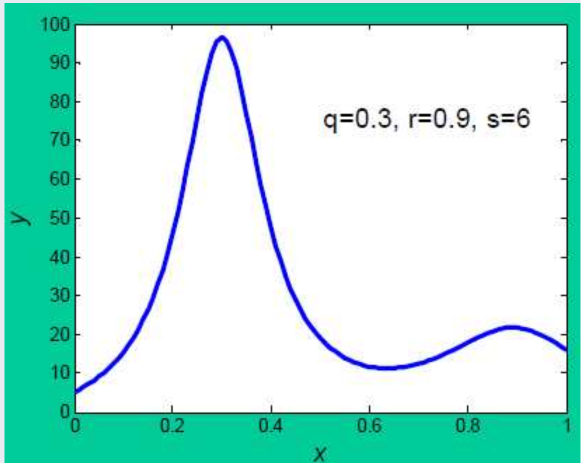
$$\int_{-1}^{1} f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

Adaptive Quadrature

- Methods such as Simpson's 1/3 rule has a disadvantage in that
  - it uses equally spaced points
  - If a function has regions of abrupt changes,
    - small steps must be used over the *entire domain* to achieve a certain accuracy.

- *Adaptive quadrature* methods
  - adjust the step size so that
    - small steps are taken in regions of sharp variations and
    - larger steps are taken where the function changes gradually.
    - use two different rules of quadrature, and use their difference as an estimate of the error from quadrature
    - adjust the step size by error

$$f(x)=\frac{1}{(x-q)^2+0.01}+\frac{1}{(x-r)^2+0.04}-s$$



Adaptive Quadrature in MATLAB

- MATLAB has two built-in functions for implementing adaptive quadrature
  - quad
    - uses adaptive Simpson quadrature
    - possibly more efficient for low accuracies or nonsmooth functions
  - quadl
    - uses Lobatto quadrature
    - possibly more efficient for high accuracies and smooth functions

- ***q* = quad(*fun*, *a*, *b*, *tol*, *trace*)**

  - *fun* : function to be integrates
  - *a*, *b*: integration bounds
  - *tol*: desired absolute tolerance (default: 10^{-6})
  - *trace*: flag to display details or not

  - quadl has the same arguments

Lobatto Quadrature $\int_{-1}^{1} f(x)\,dx = w_1 f(-1) + w_n f(1) + \sum_{i=2}^{n-1} w_i f(x_i).$

n	x_i	x_i	w_i	w_i
3	0	0.00000	$\frac{4}{3}$	1.333333
	± 1	± 1.00000	$\frac{1}{3}$	0.333333
4	$\pm \frac{1}{5}\sqrt{5}$	± 0.447214	$\frac{5}{6}$	0.833333
	± 1	± 1.000000	$\frac{1}{6}$	0.166667
5	0	0.000000	$\frac{32}{45}$	0.711111
	$\pm \frac{1}{7}\sqrt{21}$	± 0.654654	$\frac{49}{90}$	0.544444
	± 1	± 1.000000	$\frac{1}{10}$	0.100000
6	$\sqrt{\frac{1}{21}\left(7 - 2\sqrt{7}\right)}$	± 0.285232	$\frac{1}{30}\left(14 + \sqrt{7}\right)$	0.554858
	$\sqrt{\frac{1}{21}\left(7 + 2\sqrt{7}\right)}$	± 0.765055	$\frac{1}{30}\left(14 - \sqrt{7}\right)$	0.378475
	± 1	± 1.000000	$\frac{1}{15}$	0.066667

THE END

Report : 20-2

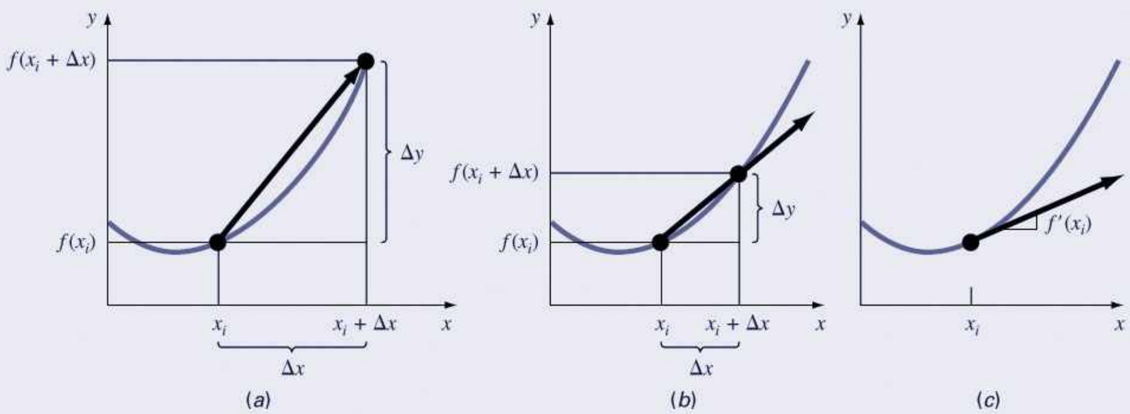
Part 5
Chapter 19

Numerical Differentiation

Differentiation

- The mathematical definition of a derivative
  - begins with a difference approximation: $$\frac{\Delta y}{\Delta x} = \frac{f(x_i + \Delta x) - f(x_i)}{\Delta x}$$
  - as Δx is allowed to approach zero, the difference becomes a derivative:

$$\frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{f(x_i + \Delta x) - f(x_i)}{\Delta x}$$



High-Accuracy Differentiation Formulas

- Taylor series expansion can be used to generate high-accuracy formulas for derivatives by
  - using linear algebra to combine the expansion around several points.

$$f(x) = f(x_0) + f(x_0)'(x-x_0) + \frac{1}{2!}f(x_0)''(x-x_0)^2 + \frac{1}{3!}f(x_0)'''(x-x_0)^3 + \cdots$$

- Three categories for the formula include *forward finite-difference*, *backward finite-difference*, and *centered finite-difference*.

- Forward:

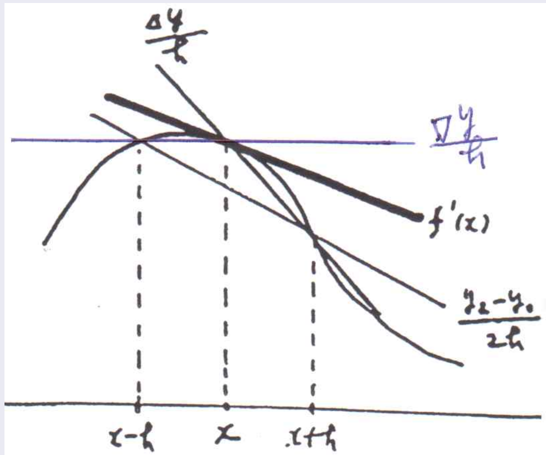
$$f(x_i)' = \frac{f(x_{i+1}) - f(x_i)}{h} + O(h)$$

- Backward:

$$f(x_i)' = \frac{f(x_i) - f(x_{i-1})}{h} + O(h)$$

- Mid value:

$$f(x_i)' = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} + O(h^2)$$



Forward Finite-Difference

First Derivative — Error

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} \qquad O(h)$$

$$f'(x_i) = \frac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(x_i)}{2h} \qquad O(h^2)$$

Second Derivative

$$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2} \qquad O(h)$$

$$f''(x_i) = \frac{-f(x_{i+3}) + 4f(x_{i+2}) - 5f(x_{i+1}) + 2f(x_i)}{h^2} \qquad O(h^2)$$

Third Derivative

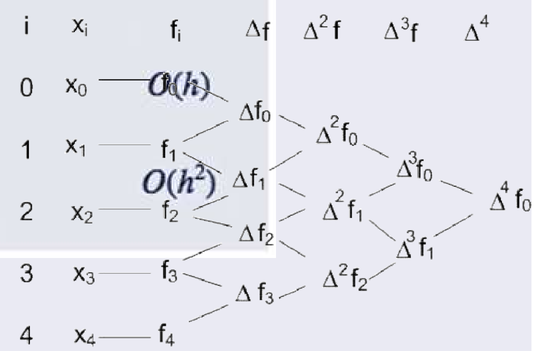
$$f'''(x_i) = \frac{f(x_{i+3}) - 3f(x_{i+2}) + 3f(x_{i+1}) - f(x_i)}{h^3} \qquad O(h)$$

$$f'''(x_i) = \frac{-3f(x_{i+4}) + 14f(x_{i+3}) - 24f(x_{i+2}) + 18f(x_{i+1}) - 5f(x_i)}{2h^3} \qquad O(h^2)$$

Fourth Derivative

$$f''''(x_i) = \frac{f(x_{i+4}) - 4f(x_{i+3}) + 6f(x_{i+2}) - 4f(x_{i+1}) + f(x_i)}{h^4} \qquad O(h)$$

$$f''''(x_i) = \frac{-2f(x_{i+5}) + 11f(x_{i+4}) - 24f(x_{i+3}) + 26f(x_{i+2}) - 14f(x_{i+1}) + 3f(x_i)}{h^4} \qquad O(h^2)$$



Backward Finite-Difference

First Derivative	Error
$f'(x_i) = \dfrac{f(x_i) - f(x_{i-1})}{h}$	$O(h)$
$f'(x_i) = \dfrac{3f(x_i) - 4f(x_{i-1}) + f(x_{i-2})}{2h}$	$O(h^2)$
Second Derivative	
$f''(x_i) = \dfrac{f(x_i) - 2f(x_{i-1}) + f(x_{i-2})}{h^2}$	$O(h)$
$f''(x_i) = \dfrac{2f(x_i) - 5f(x_{i-1}) + 4f(x_{i-2}) - f(x_{i-3})}{h^2}$	$O(h^2)$
Third Derivative	
$f'''(x_i) = \dfrac{f(x_i) - 3f(x_{i-1}) + 3f(x_{i-2}) - f(x_{i-3})}{h^3}$	$O(h)$
$f'''(x_i) = \dfrac{5f(x_i) - 18f(x_{i-1}) + 24f(x_{i-2}) - 14f(x_{i-3}) + 3f(x_{i-4})}{2h^3}$	$O(h^2)$
Fourth Derivative	
$f''''(x_i) = \dfrac{f(x_i) - 4f(x_{i-1}) + 6f(x_{i-2}) - 4f(x_{i-3}) + f(x_{i-4})}{h^4}$	$O(h)$
$f''''(x_i) = \dfrac{3f(x_i) - 14f(x_{i-1}) + 26f(x_{i-2}) - 24f(x_{i-3}) + 11f(x_{i-4}) - 2f(x_{i-5})}{h^4}$	$O(h^2)$

Centered Finite-Difference

First Derivative	Error
$f'(x_i) = \dfrac{f(x_{i+1}) - f(x_{i-1})}{2h}$	$O(h^2)$
$f'(x_i) = \dfrac{-f(x_{i+2}) + 8f(x_{i+1}) - 8f(x_{i-1}) + f(x_{i-2})}{12h}$	$O(h^4)$
Second Derivative	
$f''(x_i) = \dfrac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2}$	$O(h^2)$
$f''(x_i) = \dfrac{-f(x_{i+2}) + 16f(x_{i+1}) - 30f(x_i) + 16f(x_{i-1}) - f(x_{i-2})}{12h^2}$	$O(h^4)$
Third Derivative	
$f'''(x_i) = \dfrac{f(x_{i+2}) - 2f(x_{i+1}) + 2f(x_{i-1}) - f(x_{i-2})}{2h^3}$	$O(h^2)$
$f'''(x_i) = \dfrac{-f(x_{i+3}) + 8f(x_{i+2}) - 13f(x_{i+1}) + 13f(x_{i-1}) - 8f(x_{i-2}) + f(x_{i-3})}{8h^3}$	$O(h^4)$
Fourth Derivative	
$f''''(x_i) = \dfrac{f(x_{i+2}) - 4f(x_{i+1}) + 6f(x_i) - 4f(x_{i-1}) + f(x_{i-2})}{h^4}$	$O(h^2)$
$f''''(x_i) = \dfrac{-f(x_{i+3}) + 12f(x_{i+2}) + 39f(x_{i+1}) + 56f(x_i) - 39f(x_{i-1}) + 12f(x_{i-2}) + f(x_{i-3})}{6h^4}$	$O(h^4)$

Example

$f(x)=1.2-0.25x-0.5x^2-0.15x^3-0.1x^4$

Find $f'(0.5)$ with $h=0.25$

Exact: =0.9125

	Forward $O(h)$	Back $O(h)$	Forward $O(h^2)$	Back $O(h^2)$	Center $O(h^2)$	Center $O(h^4)$
$f'(0.5)$	-1.155	-0.714	-0.859375	-0.87812[illegible]	-0.934	-0.9125
ε_t	-26.5	21.7%	5.82%	3.77%	-2.4%	0

Richardson Extrapolation

- As with integration, the Richardson extrapolation can be used
  - to combine two lower-accuracy estimates of the derivative to produce a higher-accuracy estimate.

- For the cases where there are two $O(h^2)$ estimates and the interval is halved ($h_2 = h_1/2$),
  - an improved $O(h^4)$ estimate may be formed using:

$$D = \frac{4}{3} D(h_2) - \frac{1}{3} D(h_1)$$

- For the cases where there are two $O(h^4)$ estimates and the interval is halved ($h_2 = h_1/2$),
  - an improved $O(h^6)$ estimate may be formed using:

$$D = \frac{16}{15} D(h_2) - \frac{1}{15} D(h_1)$$

- For the cases where there are two $O(h^6)$ estimates and the interval is halved ($h_2 = h_1/2$),
  - an improved $O(h^8)$ estimate may be formed using:

$$D = \frac{64}{63} D(h_2) - \frac{1}{63} D(h_1)$$

Unequally Spaced Data

- One way to calculated derivatives of unequally spaced data is
  - to determine a polynomial fit and take its derivative at a point.

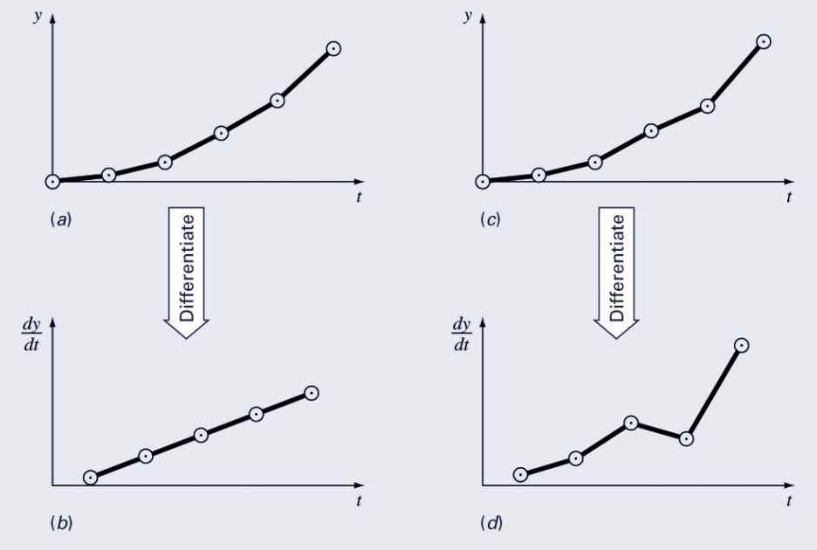
- As an example, using a second-order Lagrange polynomial to fit three points and taking its derivative yields:

$$f'(x) = f(x_0)\frac{2x - x_1 - x_2}{(x_0 - x_1)(x_0 - x_2)} + f(x_1)\frac{2x - x_0 - x_2}{(x_1 - x_0)(x_1 - x_2)} + f(x_2)\frac{2x - x_0 - x_1}{(x_2 - x_0)(x_2 - x_1)}$$

Derivatives and Integrals for Data with Errors

- A shortcoming of numerical differentiation is that
  - Differentiation tends to amplify data errors in data,
  - whereas integration tends to smooth data errors (positive and negative error cancel each other).

- One approach for taking derivatives of data with errors is to
  - fit a smooth (regression) differentiable function to the data and
  - take the derivative of the function.



Numerical Differentiation with MATLAB

- Two built-in functions to help take derivatives, diff and gradient:
- diff(*x*)
  - Returns the difference between adjacent elements in x
  - y = diff(x, n) applies diff recursively n times, resulting in the n^{th} difference.
    - diff(x, 2) is the same as diff(diff(x))
- diff(*y*)./diff(*x*)
  - Returns the difference between adjacent values in y divided by the corresponding difference in adjacent values of x

$$f(x_i)' = \frac{f(x_{i+1}) - f(x_i)}{h}$$

```
x = [1 2 3 4 5];
y = diff(x)
y =
     1     1     1     1
```

```
z = diff(x,2)
z =
     0     0     0
```

Numerical Differentiation with MATLAB

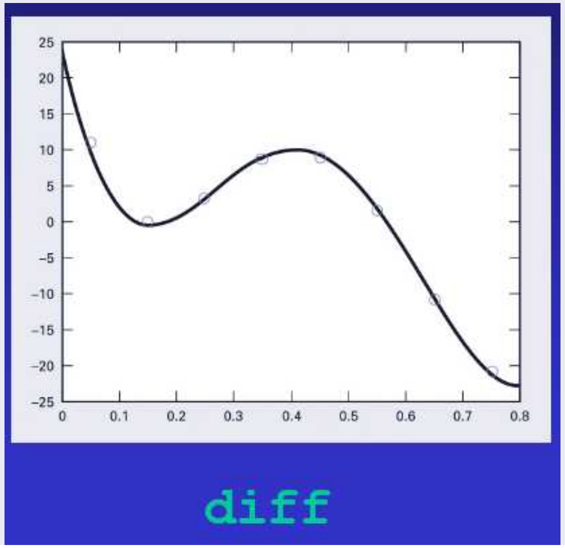
- gradient : *fx = gradient(f, h)* $\nabla F = \frac{\partial F}{\partial x}\hat{i} + \frac{\partial F}{\partial y}\hat{j}$
  - Determines the derivative of the data in f at each of the points.
    - forward difference for the first point,
    - backward difference for the last point,
    - centered difference for the interior points.
  - *h* is the spacing between points (if omitted *h=1)*
  - Partial derivatives for matrices : [*fx*, *fy*] = gradient(*f*, *h*)

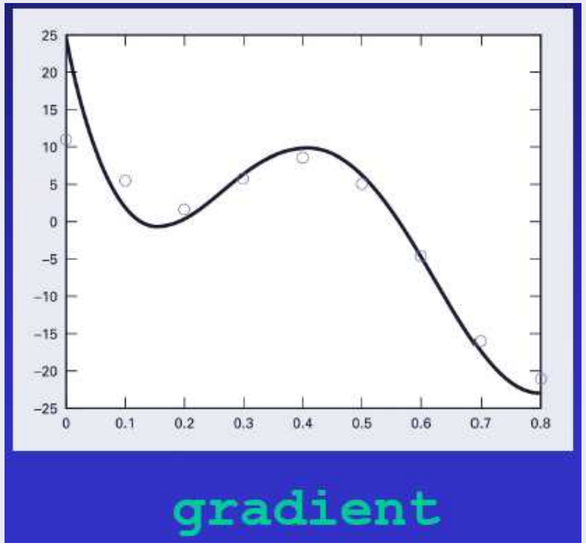
Example

$f(x)=0.2+25x-200x^2+675x^3-900x^4+400x^5$ $f'(x)=25-400x+2025x^2-3600x^3+2000x^4$

\>> x=0:0.1:0.8;
\>> y=f(x);
\>> diff(x) → ans= 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
\>> d=diff(y)./diff(x) → d=10.8900 -0.0100 3.1900 8.4900 8.6900 1.3900 -11.0100 -21.3100
\>> xm=(x(1:n-1)+x(2:n)./2; (∵ vector d 가 인접한 원소의 중점에 해당하는 도함수 값을 가지므로 plot 을 위하여 해당하는 x 축 값 생성) (gradient 의 구간의 ½ 역활 → 정확도)
>> plot(xm, d, 'o', xa, ya)

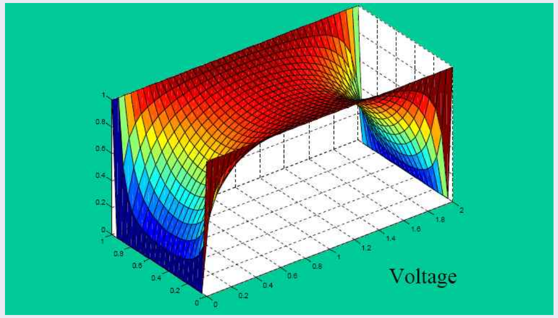
\>> dy=gradient(y, 0.1)
dy = 10.8900 5.4400 1.5900 5.8400 8.5900 5.0400 -4.8100 -16.1600 -31.3100
\>> plot(x, dy, 'o', xa, ya)





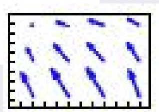
Visualization

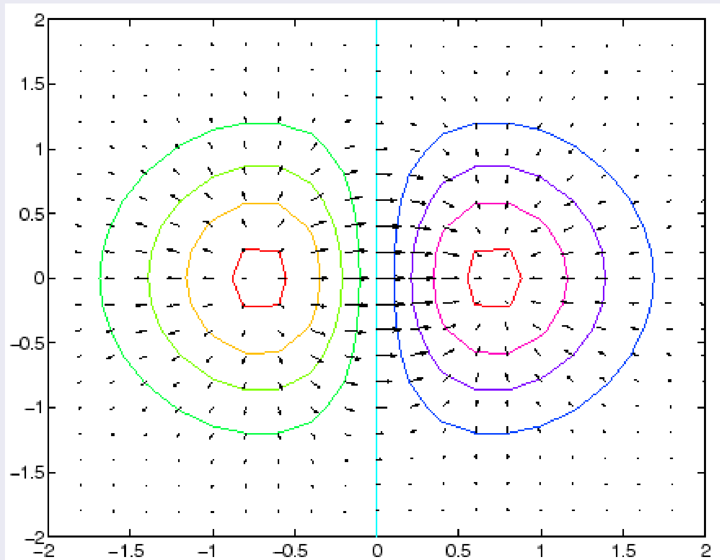
- Assuming x and y represent a meshgrid of x and y values and z represents a function of x and y,
  - contour(*x*, *y*, *z*) : generate a contour plot
  - quiver(x, y, u, v) : displays velocity vectors as
    - Arrows with components (u,v) at the points (x, y)
    - quiver plot or velocity plot



Plot the gradient field of the function $z = xe^{(-x^2 - y^2)}$:

```
[X,Y] = meshgrid(-2:.2:2);
Z = X.*exp(-X.^2 - Y.^2);
[DX,DY] = gradient(Z,.2,.2);
contour(X,Y,Z)
hold on
quiver(X,Y,DX,DY)
```





THE END

Report : 21-1, 21-4, 21-11