Numerical Integration

$$\int_a^b$$

"Applied" Math ≈ Numerical Recipe (power)
Helps understand <u>integration concept</u>
analytic/numeric = clever/easy

Rubin H Landau
With
Sally Haerer

Computational Physics for Undergraduates
BS Degree Program: Oregon State University

"Engaging People in Cyber Infrastructure"
Support by EPICS/NSF & OSU

Quadrature = Box Counting

- ***Definite*** **Integral = area**
- **Standard numerical form (count boxes)**
- **Riemann def: sum over boxes width *h***

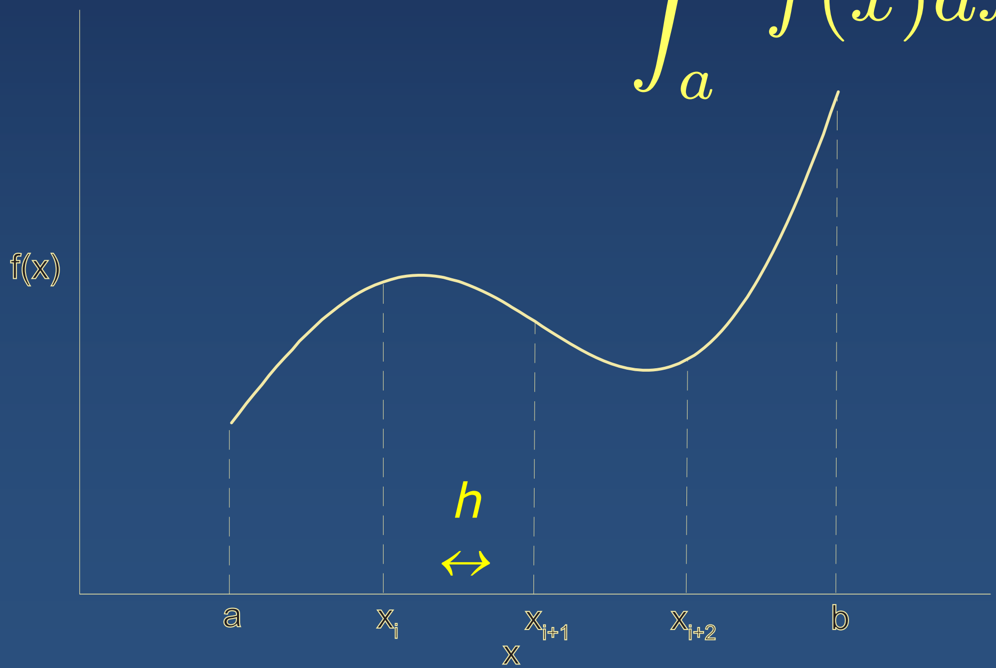
$$\int_a^b f(x)dx = \sum_{i=1}^{N} h f(x_i) \qquad (1)$$

(h → 0 N → ∞) *(exact)*

$$\approx \sum_{i=1}^{N} f(x_i) w_i \qquad (2)$$

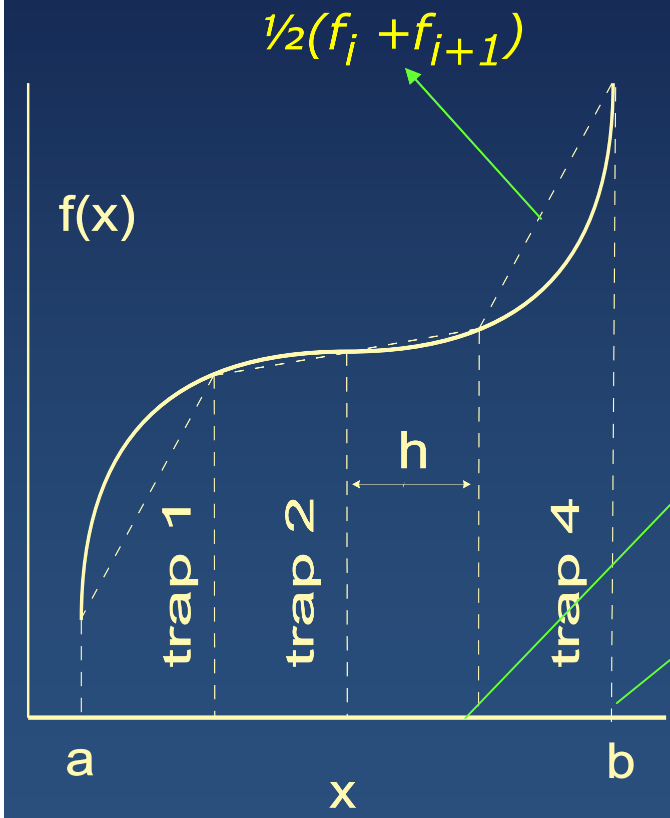
Points, Weights

(algorithm)



Quadrature Rules (4)

- 3 now, Monte Carlo later (Δ, best for ≥ 3D)
- All: integrand *f (x)* ≈ polynomial *O* = 1, 2, N
- **Trapezoid Rule** [f(x) ≈ straight line]



1 trapezoid area: *(1)*

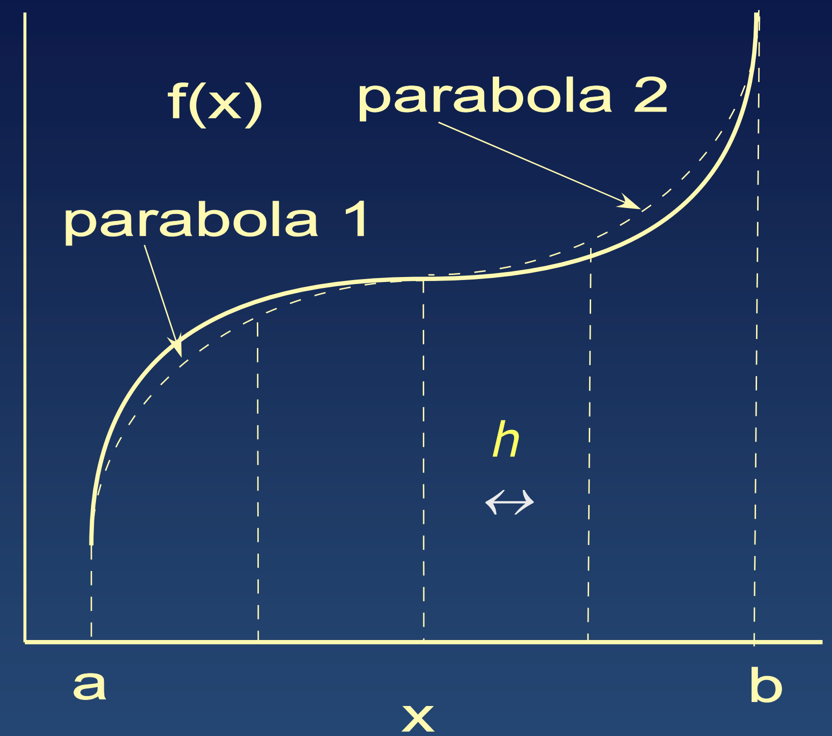
$$\int_{x_i}^{x_{i+1}} f(x)dx \simeq \frac{1}{2}hf_i + \frac{1}{2}hf_{i+1}$$

[a,b], add trapezoids:

$$\int_a^b f(x)dx \approx \qquad (2)$$

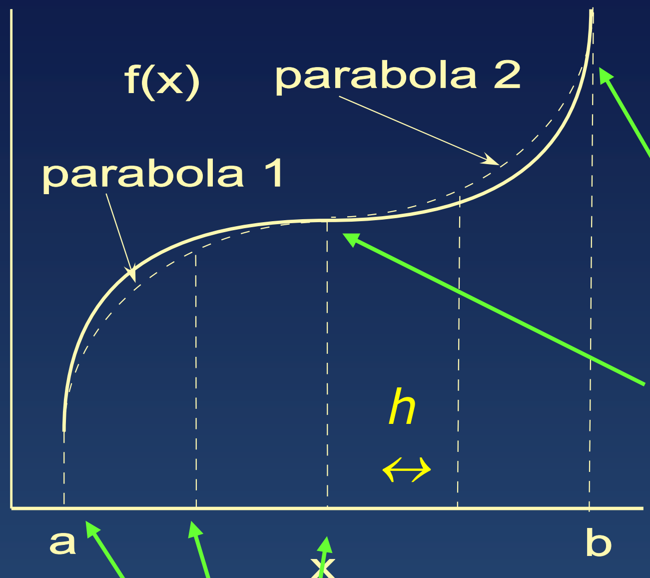
$$\frac{h}{2}f_1 + hf_2 + \cdots hf_{N-1} + \frac{h}{2}f_N$$

Simpson's Rule (# 2)



- **Idea:** ***f(x) ≈ ∑* parabolas *≈ ∑ (ax² + bx + c)***
- ***2* intervals for each parabola**
- ***N* evenly spaced <u>points</u>*:* *x [a, b]***
- **# <u>intervals</u> = *N-1* = even ⇒ *N* odd**

Simpson-Rule Weights



- Parabola = $ax^2 + bx + c$
- Single parabola (2 interval) area:

$$\int_{x_i-h}^{x_i+h} f(x)dx \simeq \frac{h}{3}f_{i-1} + \frac{4h}{3}f_i + \frac{h}{3}f_{i+1} \quad (1)$$

- Add parabolas entire region *[a,b], :*

$$\int_a^b f(x)dx \approx \frac{h}{3}f_1 + \frac{4h}{3}f_2 + \frac{2h}{3}f_3 + \cdots + \frac{h}{3}f_N \quad (2)$$

(N = odd!)

$$= \sum_i^N w_i x_i \qquad \text{(standard form)} \quad (3)$$

$$\Rightarrow \quad w_i = \frac{h}{3}\{1,\ 4,\ 2,\ 4,\ \ldots\ 4,\ 1\} \quad (4)$$

Review: Trapezoid & Simpson's Rules

Trapezoid Rule

$$\int_a^b f(x)dx \;\approx\; \frac{h}{2}f_1 + hf_2 + hf_3 + \cdots hf_{N-1} + \frac{h}{2}f_N \qquad (1)$$

Simpson's Rule (parabolas)

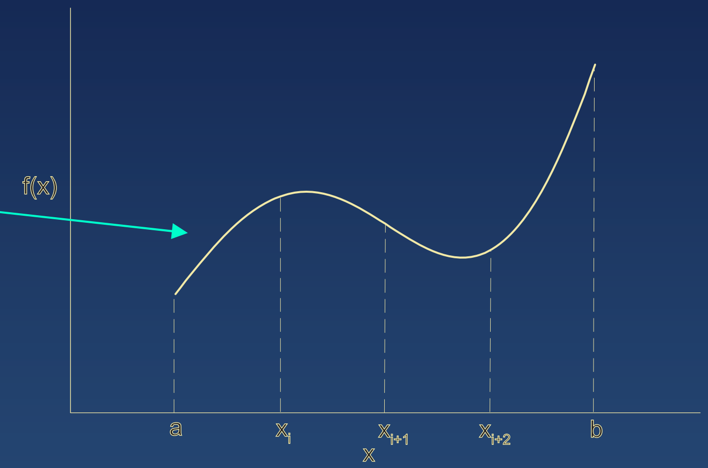
$$\int_a^b f(x)dx \;\approx\; \frac{h}{3}f_1 + \frac{4h}{3}f_2 + \frac{2h}{3}f_3 + \frac{4h}{3}f_4 + \cdots + \frac{4h}{3}f_{N-1} + \frac{h}{3}f_N \qquad (2)$$

Integration Rule 3*: Gaussian Quadrature

- Still follow basic formula

$$\int_a^b f(x)dx \approx \sum_{i=1}^{N} w_i f(x_i) \qquad (1)$$

- Gauss pick $\{x_i, w_i\}$ for N points
- Exact if $f = 2N-1$ degree polynomial
  text: *f (x)* → *W (x) f (x)*



- Not evenly spaced, still *N* points (smart)!

$$P_N(y_i) = 0, \qquad w_i' = \frac{2}{(1-y_i^2)[P_N'(y_i)]^2}, \qquad (-1 < y < 1) \qquad (2)$$

- Just use table/subroutine, and scale to ($a < x < b$)

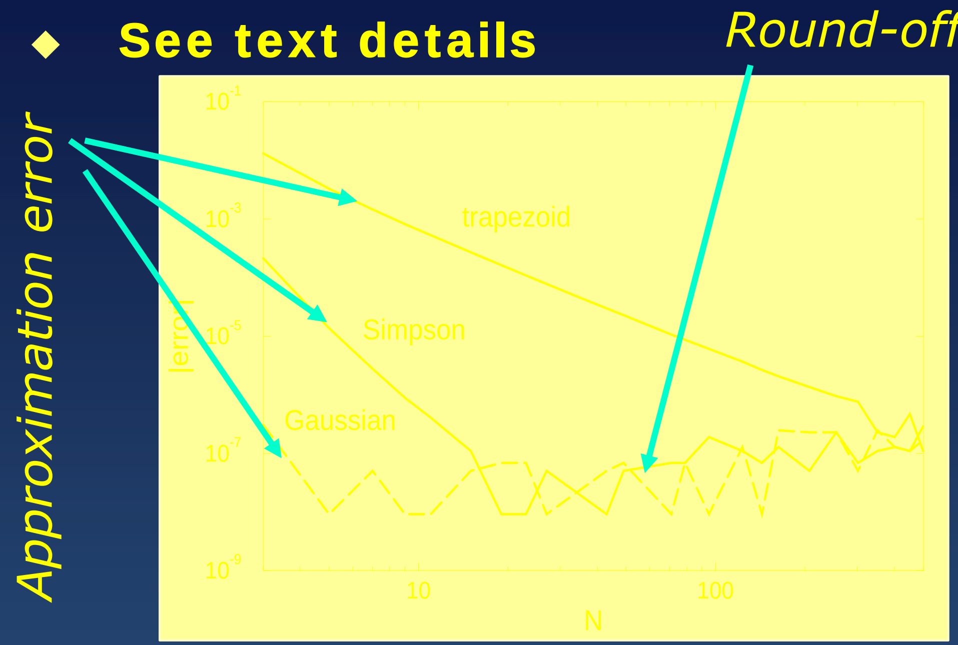
$$x_i = \frac{b+a}{2} + \frac{b-a}{2} y_i, \qquad w_i = \frac{b-a}{2} w_i' \qquad (3)$$

IntegGauss.java

```
public class IntegGauss  {

 public static void main(String[] argv)    {
   int i;   double result;
   for ( i=3;  i <= 11;  i  += 2)   result = gaussint(i, 0., 1.);    }

  public static double f (double x) {return (Math.exp(-x));}            // f(x)

  public static double gaussint (int no,  double min,  double max)  {
    int n;       double quadra = 0.;
    double w[]  =  new double[21],           x[] = new double[21];
    gauss (no, 0, min, max, x, w);                                // Returns pts & wts
    for ( n=0;  n <  no;  n++ )   quadra += f(x[n]) * w[n]; }       // Rule
    return (quadra);
  }
```

Assessment: Integration Errors

- **See text details**



$$err = \frac{\text{numeric - exact}}{\text{exact}}$$

$$E_{trap} = O\left(\frac{[b-a]^3}{N^2}\right) f^{(2)}$$
$$E_{Simp} = O\left(\frac{[b-a]^5}{N^4}\right) f^{(4)} \qquad (1)$$

- **Best *N* (# int points) ? $\Leftarrow$ Min $\varepsilon_{TOT} = \varepsilon_{RO}$ ($\propto \sqrt{N}$) + ε_{APPROX}**

$$N_{trap} \approx \frac{1}{(\epsilon_m)^{2/5}} \approx 631$$
$$N_{Simp} \approx \frac{1}{(\epsilon_m)^{2/9}} \approx 36$$

$$\epsilon_{ro} \approx \begin{cases} 3\times 10^{-6} & \text{(trapezoid)} \\ 6\times 10^{-7} & \text{(Simpson)} \end{cases} \qquad (2)$$

- **Faster $\Rightarrow$ smaller *N* $\Rightarrow$ smaller ε_{RO} $\Rightarrow$ smaller ε_{TOT}**

Lab: Empirical Error Estimates

You deserve a break!

This stuff really works!

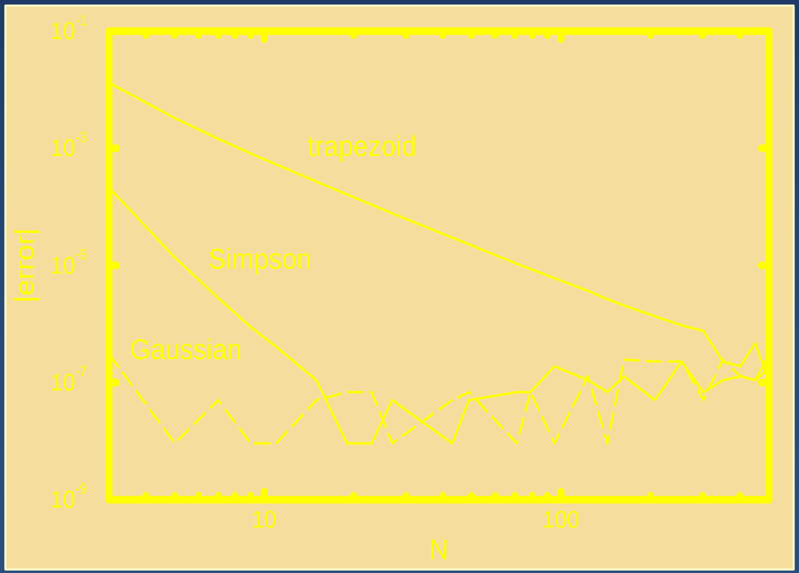
Lab: Empirical Error Estimates

1. **Integrate *f(x)* via trapezoid, Simpson, Gaussian quadrature**
2. **Debug with *f(x)* = e^{-x} :**

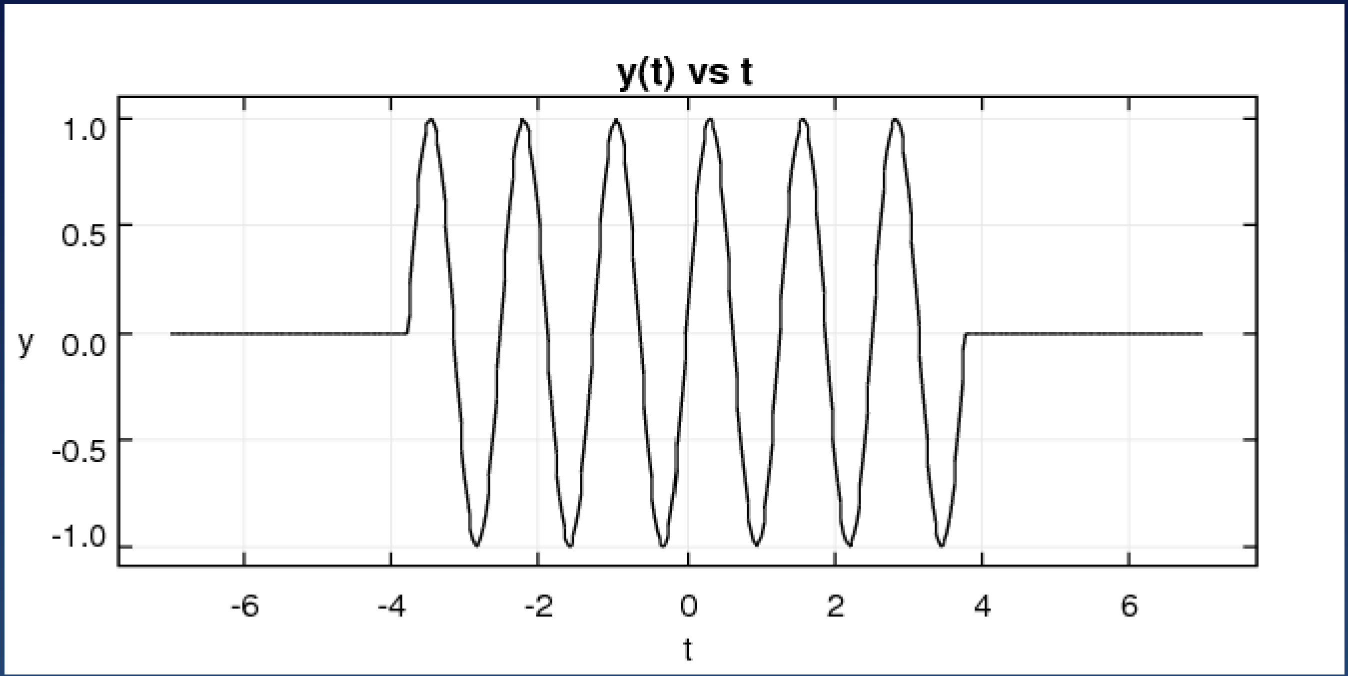
$$\int_0^1 e^{-x} dx \;=\; 1 - e^{-1} \qquad \epsilon = \frac{\text{numeric - exact}}{\text{exact}}$$

3. **Plot $\log_{10}(|\varepsilon|)$ *vs* $\log_{10}(N)$,**

$$\epsilon = CN^{\alpha} \;\Rightarrow\; \log\epsilon = \alpha\log N + \mathbf{C}$$



Trouble



$$F_1 = \int_0^{2\pi} \sin(100x)dx$$

$$F_2 = \int_0^{2\pi} \sin^x(100x)dx$$

