

Numerical Differentiation

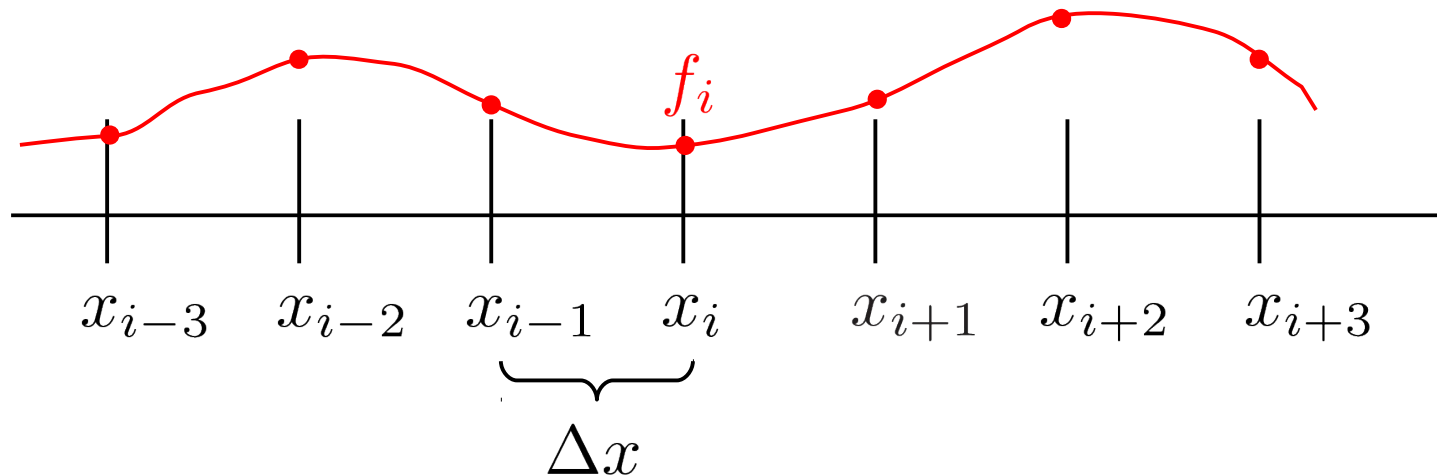
- We'll follow the discussion in Pang (Ch. 3) with some additions along the way
- Numerical differentiation approximations are key for:
 - Solving ODEs
 - PDEs

Numerical Differentiation

- We can imagine 2 situations
 - We have our function $f(x)$ defined only at a set of (possibly regularly spaced) points
 - Generally speaking, asking for greater accuracy involves using more of the discrete points in the approximation for f'
 - We have an analytic expression for $f(x)$ and want to compute the derivative numerically
 - Usually it would be better to take the analytic derivative of $f(x)$, but we can learn something about error estimation in this case.
 - Used, for example, in computing the numerical Jacobian for integrating a system of ODEs (we'll see this later)

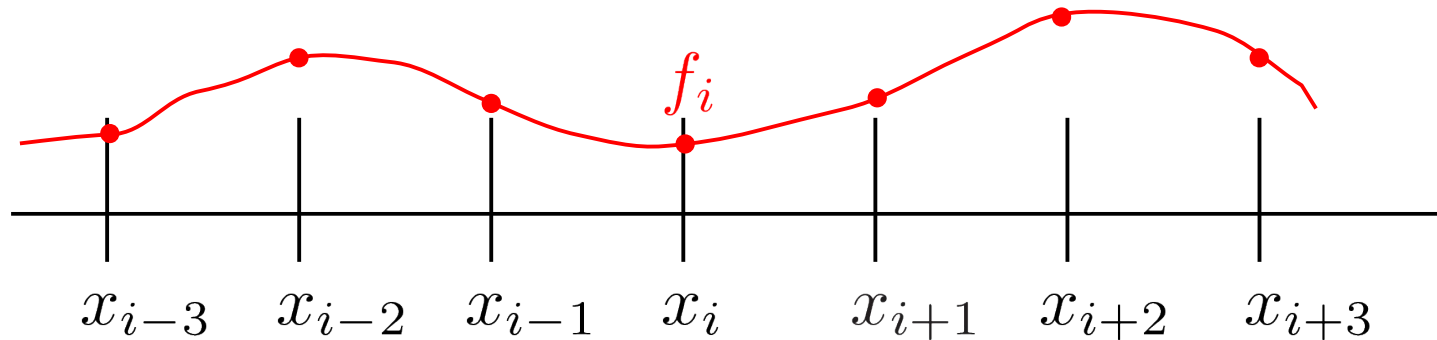
Gridded Data

- Discretized data is represented at a finite number of locations
 - Integer subscripts are used to denote the position (index) on the grid
 - Structured/regular: spacing is constant



- Data is known only at the grid points: $f_i = f(x_i)$

First Derivative / Order of Accuracy



- Taylor expansion:

$$f_{i+1} = f(x_i + \Delta x) = f_i + \left. \frac{\partial f}{\partial x} \right|_{x_i} \Delta x + \frac{1}{2} \left. \frac{\partial^2 f}{\partial x^2} \right|_{x_i} \Delta x^2 + \dots$$

- Solving for the first derivative:

$$\underbrace{\left. \frac{\partial f}{\partial x} \right|_{x_i}}_{\text{Discrete approximation to } f'} = \underbrace{\frac{f_{i+1} - f_i}{\Delta x}}_{\text{Discrete approximation to } f'} - \underbrace{\frac{1}{2} \left. \frac{\partial^2 f}{\partial x^2} \right|_{x_i} \Delta x}_{\text{Leading term in the truncation error}} + \dots$$

First Derivative / Order of Accuracy

- This is a first-order accurate expression for the derivative at point i
 - Alternately, we can use the point to the left (**blackboard**)
 - They are called **difference** or **finite-difference** formulae
- Shorthand: $\mathcal{O}(\Delta x)$
 - “big-O notation”
- How can we get higher order?

First Derivative / Order of Accuracy

- First derivative approximations:

- First-order (one-sided):

$$f' = \frac{f_i - f_{i-1}}{\Delta x} \quad f' = \frac{f_{i+1} - f_i}{\Delta x}$$

2-point stencil

- Second-order (centered):

$$f' = \frac{f_{i+1} - f_{i-1}}{2\Delta x}$$

3-point stencil

- Fourth-order:

$$f' = \frac{-f_{i+2} + 8f_{i+1} - 8f_{i-1} + f_{i-2}}{12\Delta x}$$

5-point stencil

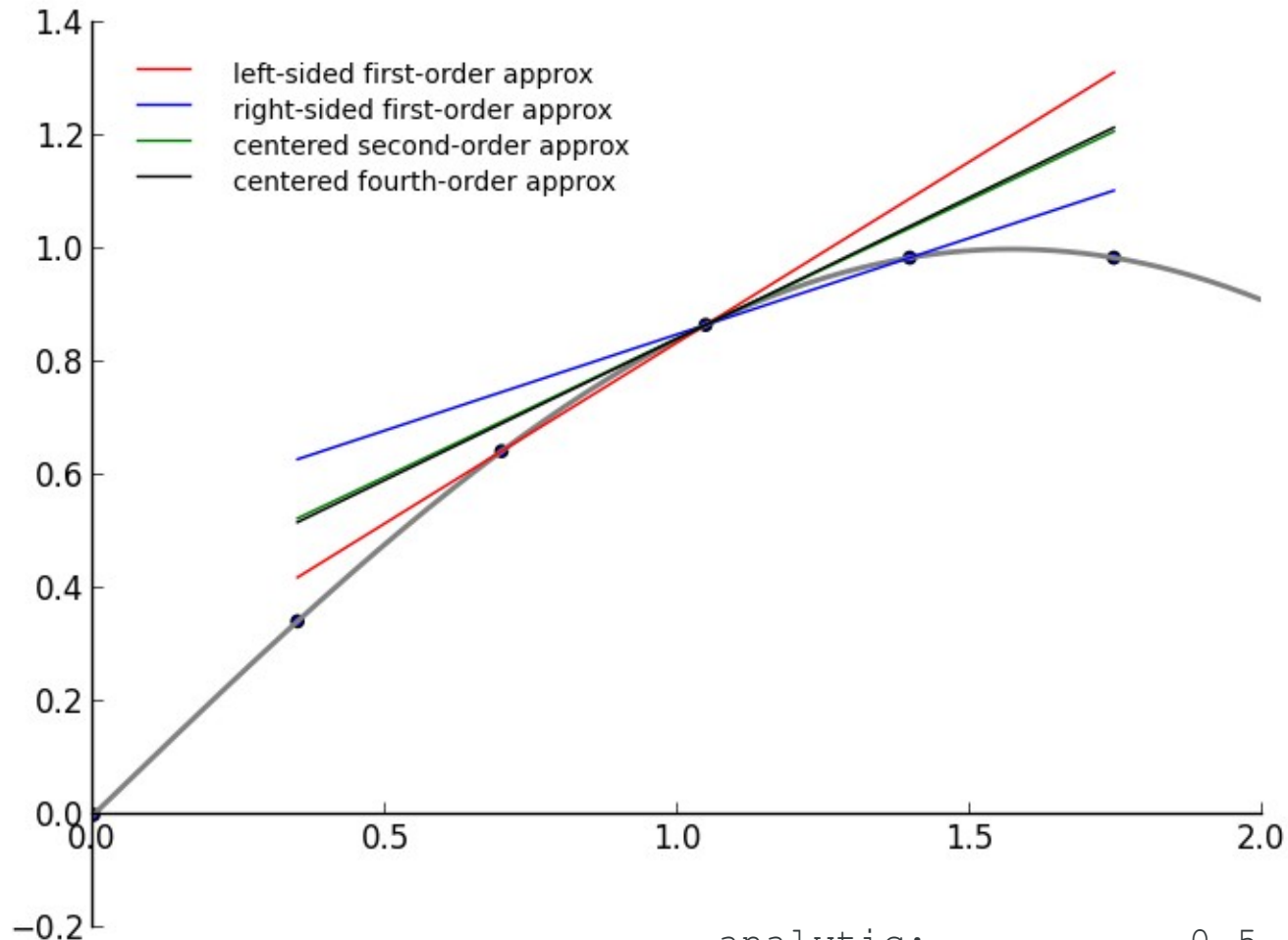
- Range of points involved is called the stencil

- Some points may have a '0' coefficient

First Derivative / Order of Accuracy

- General trend: more points = higher accuracy
 - Found via Taylor expanding from greater distances and algebra
- What happens at the boundaries of our finite-gridded data?
 - Can interpolate past the last point to use the same stencil
 - Can switch to one-sided stencils
- **Practically speaking: the first and second order approximations are the ones that are used the most often.**

First Derivative Comparison



analytic:	0.5
left-sided $O(dx)$:	0.639529171481
right-sided $O(dx)$:	0.34028636457
centered $O(dx^2)$:	0.489907768026
centered $O(dx^4)$:	0.499756119208

Roundoff vs. Truncation Error

(Yakowitz & Szidarovszky)

- Just evaluating f at our gridded points introduces round-off error:

- \bar{f}_{i+1} is an approximation to $f_{i+1} = f(x_{i+1})$
- Assume some bound: $|f_{i+1} - f_i| \leq \delta$
- Error is (blackboard):

$$\left| f' - \frac{\bar{f}_{i+1} - \bar{f}_i}{\Delta x} \right| \leq \underbrace{\frac{|f''|\Delta x}{2}}_{\text{truncation}} + \underbrace{\frac{2\delta}{\Delta x}}_{\text{roundoff}}$$

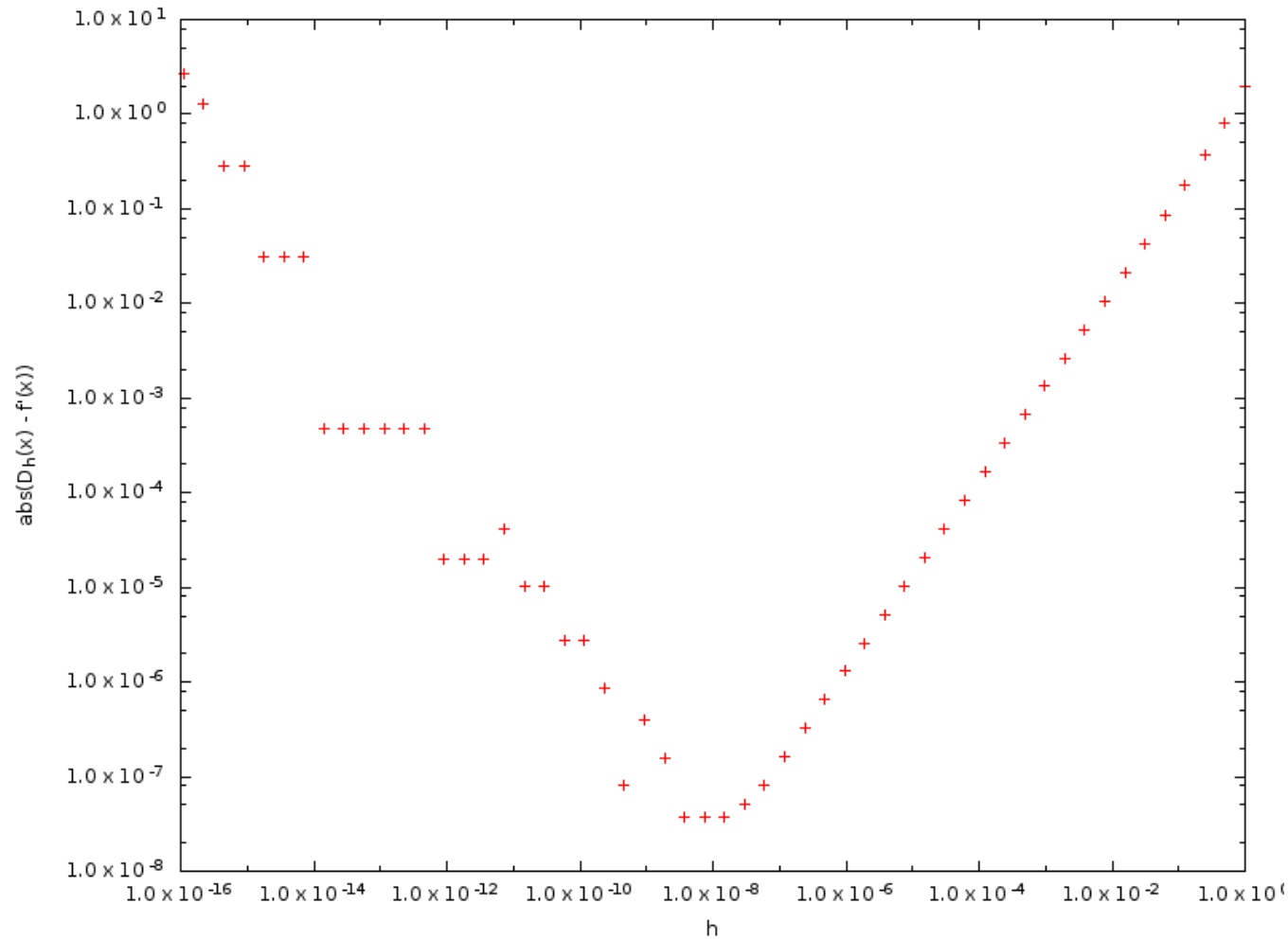
This should be near machine ϵ

As $\Delta x \rightarrow \epsilon$, the roundoff term becomes $O(1)$

Another thing to consider: with roundoff, is $(x + \Delta x) - x = \Delta x$?

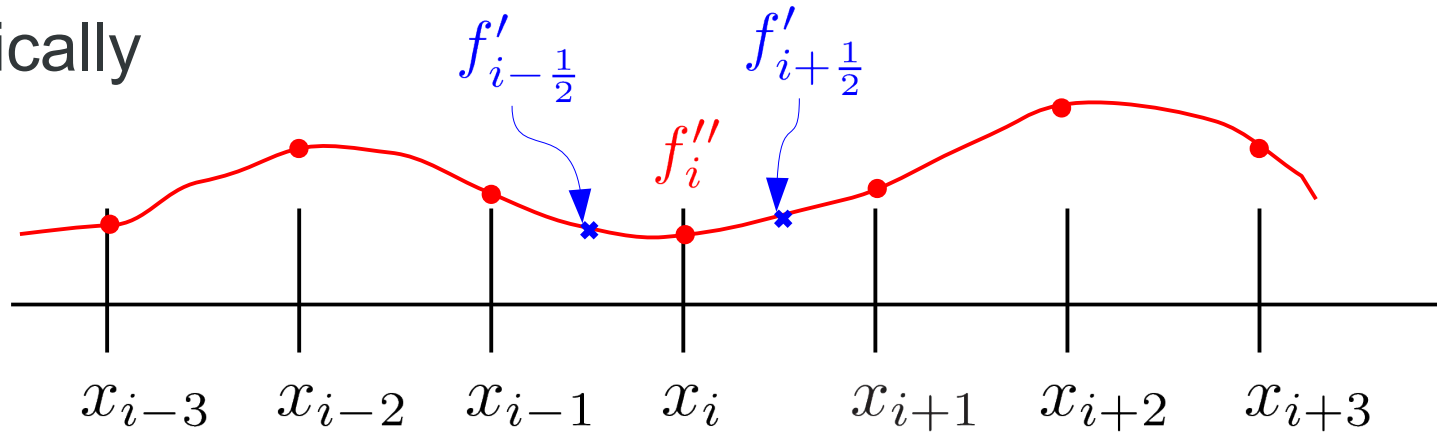
Round-off vs. Truncation Error

- $\exp(x)$



Higher-Derivatives

- Graphically



$$f'_{i+1/2} = \frac{f_{i+1} - f_i}{\Delta x}$$

This is 2nd order at the midpoint between the two points

$$f''_i = \frac{f'_{i+1/2} - f'_{i-1/2}}{\Delta x}$$

This is a centered difference (derivative) of the derivatives = second derivative

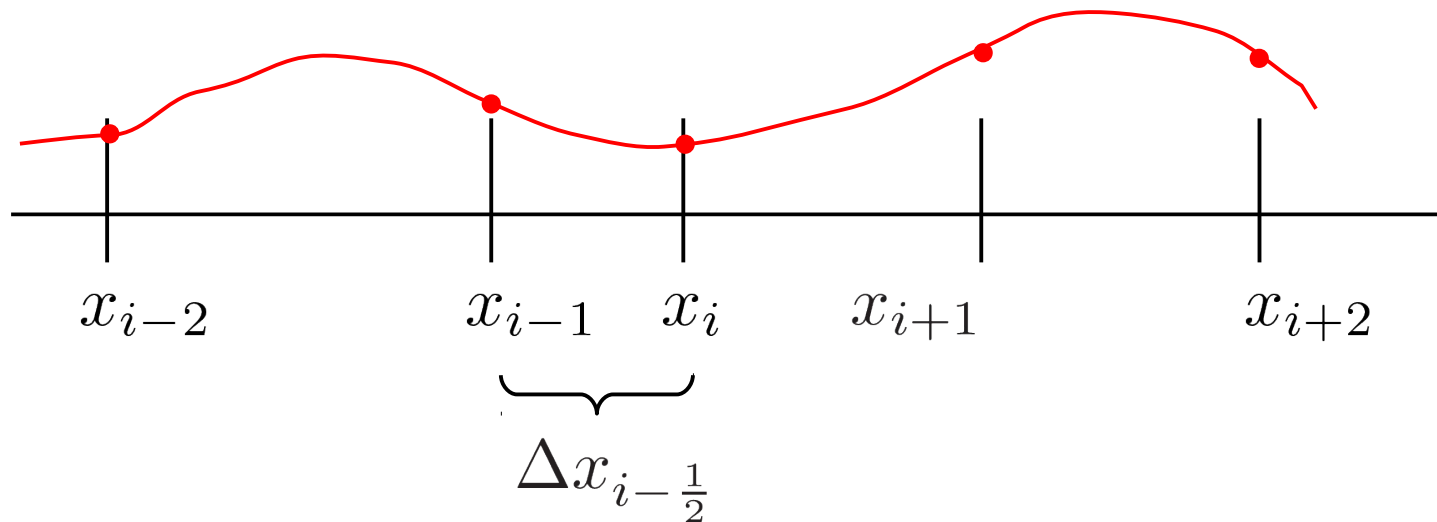
$$= \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2}$$

Second-order accurate

- Also via Taylor expansion (**blackboard**)

Non-Uniform Data

- Two choices:
 - Interpolate to a uniform grid
 - Re-derive our expressions for a non-uniform grid (preferred)



$$f' = \frac{\Delta x_{i-\frac{1}{2}}^2 f_{i+1} + (\Delta x_{i+\frac{1}{2}}^2 - \Delta x_{i-\frac{1}{2}}^2) f_i - \Delta x_{i+\frac{1}{2}}^2 f_{i-1}}{\Delta x_{i-\frac{1}{2}} \Delta x_{i+\frac{1}{2}} (\Delta x_{i+\frac{1}{2}} + \Delta x_{i-\frac{1}{2}})} + \mathcal{O}(\Delta x_{i-\frac{1}{2}}^2) + \mathcal{O}(\Delta x_{i+\frac{1}{2}}^2)$$

(blackboard derivation...)

Analytic f Given

- If we have $f(x)$ available analytically, we can make estimates of the error
 - This will come into play with ODEs, where we have the analytic righthand side
- Controlling accuracy
 - Consider:
$$\Delta_1(h) \equiv \frac{f(x+h) - f(x-h)}{2h}$$
 - We are free to choose h
 - Compare $\Delta_1(h)$ to $\Delta_1(h/2)$ estimate error

Analytic f Given

- Iteratively build more accurate approximations

- $f(x \pm h) = f(x) \pm hf' + \frac{1}{2}h^2f'' \pm \frac{1}{3!}h^3f''' + \frac{1}{4!}h^4f^{(4)} \pm \frac{1}{5!}h^5f^{(5)} + \dots$

- This gives:

$$\Delta_1(h) = f' + \frac{1}{6}h^2f''' + \mathcal{O}(h^4)$$

- Consider: $\Delta_1(h/2) = f' + \frac{1}{6}\frac{1}{4}h^2f''' + \mathcal{O}(h^4)$

- Combine: $f' = -\frac{\Delta_1(h) - 4\Delta_1(h/2)}{3} + \mathcal{O}(h^4)$

- This is an example of Richardson extrapolation—we'll see this more when we go to ODEs

Numerical Integration

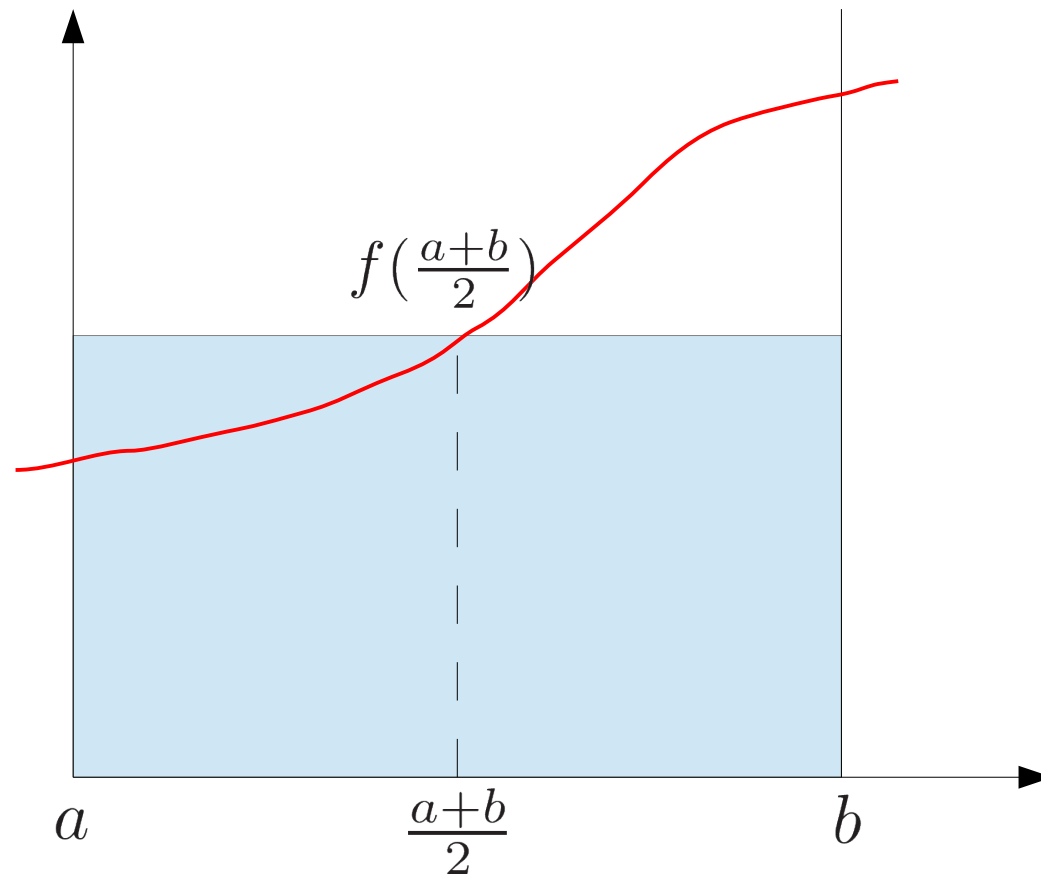
- We want to solve:

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

- Again, we have two distinct cases:
 - $f(x)$ is provided at discrete points on a grid
 - We have an analytic expression for $f(x)$
- We'll follow the discussion in Pang and also that of Garcia

Numerical Integration

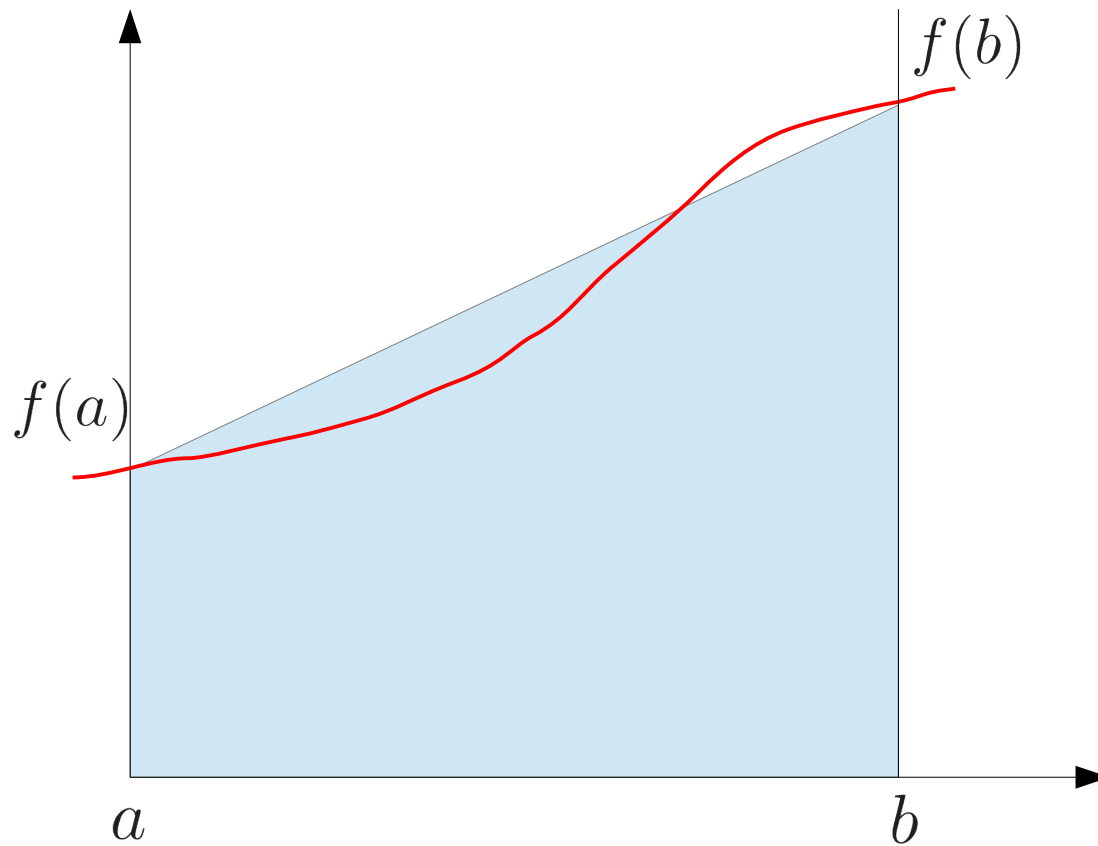
- Simplest case: piecewise constant interpolant (**midpoint rule**)



$$I \approx (b - a) f\left(\frac{a+b}{2}\right)$$

Numerical Integration

- One step up: piecewise linear interpolant (**trapezoid rule**)



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

← This is just the area of a trapezoid

Numerical Integration

- As you might expect, the accuracy gets better the higher-order the interpolating polynomial
 - Trapezoid rule will integrate a linear $f(x)$ perfectly
- What about a parabola?
 - For now, we'll stick with equally spaced locations at which we evaluate $f(x)$

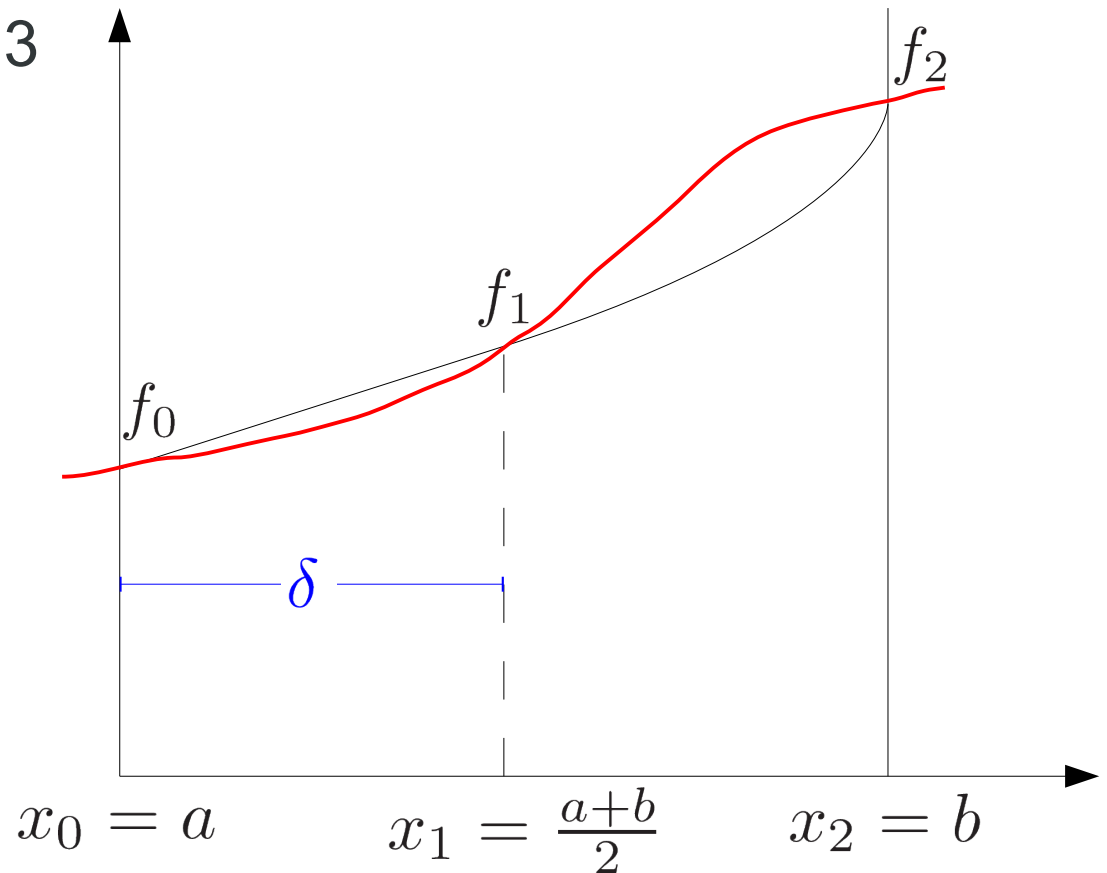
Simpson's Rule

- Piecewise linear interpolant (**Simpson's rule**)
 - 3 unknowns (A, B, C) and 3 points
 - **Blackboard algebra...**

$$A = \frac{f_0 - 2f_1 + f_2}{2\delta^2}$$

$$B = -\frac{f_2 - 4f_1 + 3f_0}{2\delta}$$

$$C = f_0$$



$$f(x) = A(x - x_0)^2 + B(x - x_0) + C$$

Simpson's Rule

- Then integrate under the parabola

$$\begin{aligned} I &= \int_{x_0}^{x_2} [A(x - x_0)^2 + B(x - x_0) + C] dx \\ &= \frac{\delta}{3} (f_0 + 4f_1 + f_2) \end{aligned}$$

Summary of Simple Rules

(Yakowitz & Szidarovszky)

- Error estimates

- Actually rather complicated to derive (see a math text on Numerical Methods)

- Simple trapezoidal:

$$\int_a^b f(x) \approx \frac{\delta}{2}(f(a) + f(b)) - \frac{\delta^3}{12}f''(\zeta) \quad \delta = b - a$$

- Simple Simpson's:

$$\int_a^b f(x)dx \approx \frac{\delta}{3}(f(a) + 4f(\frac{a+b}{2}) + f(b)) - \frac{\delta^5}{90}f^{(4)}(\zeta) \quad \delta = \frac{b-a}{2}$$

- Note the only way to reduce the error here is to make $[a, b]$ smaller
- Here, ζ is some unknown point in $[a, b]$

Summary of Simple Rules

- Any numerical integration method that represents the integral as a (weighted) sum at a discrete number of points is called a **quadrature rule**
- Fixed spacing between points (what we've seen so far): **Newton-Cotes quadrature**

Open Integration Rules

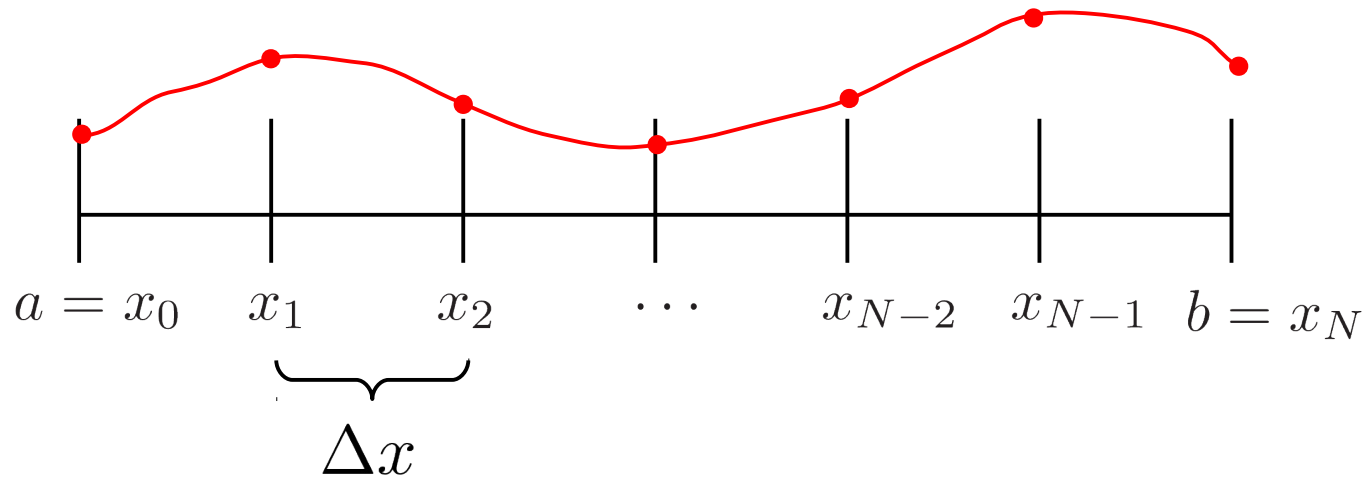
- Forms of these exist where the end-points of the interval are not used—these are **open integration rules**
 - Usually not very desirable
 - See, for example, Numerical Recipes

Compound Integration

- Mid-point, trapezoidal, and Simpson's integration as we wrote them are ok **when $[a,b]$ is small**.
- Integrating over large domain is not very accurate
 - We could keep adding terms to our polynomials (getting higher and higher degree), or we could string together our current expressions
 - More points = more accuracy
 - **Compound integration**—break domain into sub-domains and use these rules in each sub-domain.

Compound Integration

- Break interval into chunks



$$I \equiv \int_a^b f(x) dx = \sum_{i=0}^{N-1} \underbrace{\int_{x_i}^{x_{i+1}} f(x) dx}_{\text{Integral over a single slab}}$$

Compound Integration

- Compound Trapezoidal

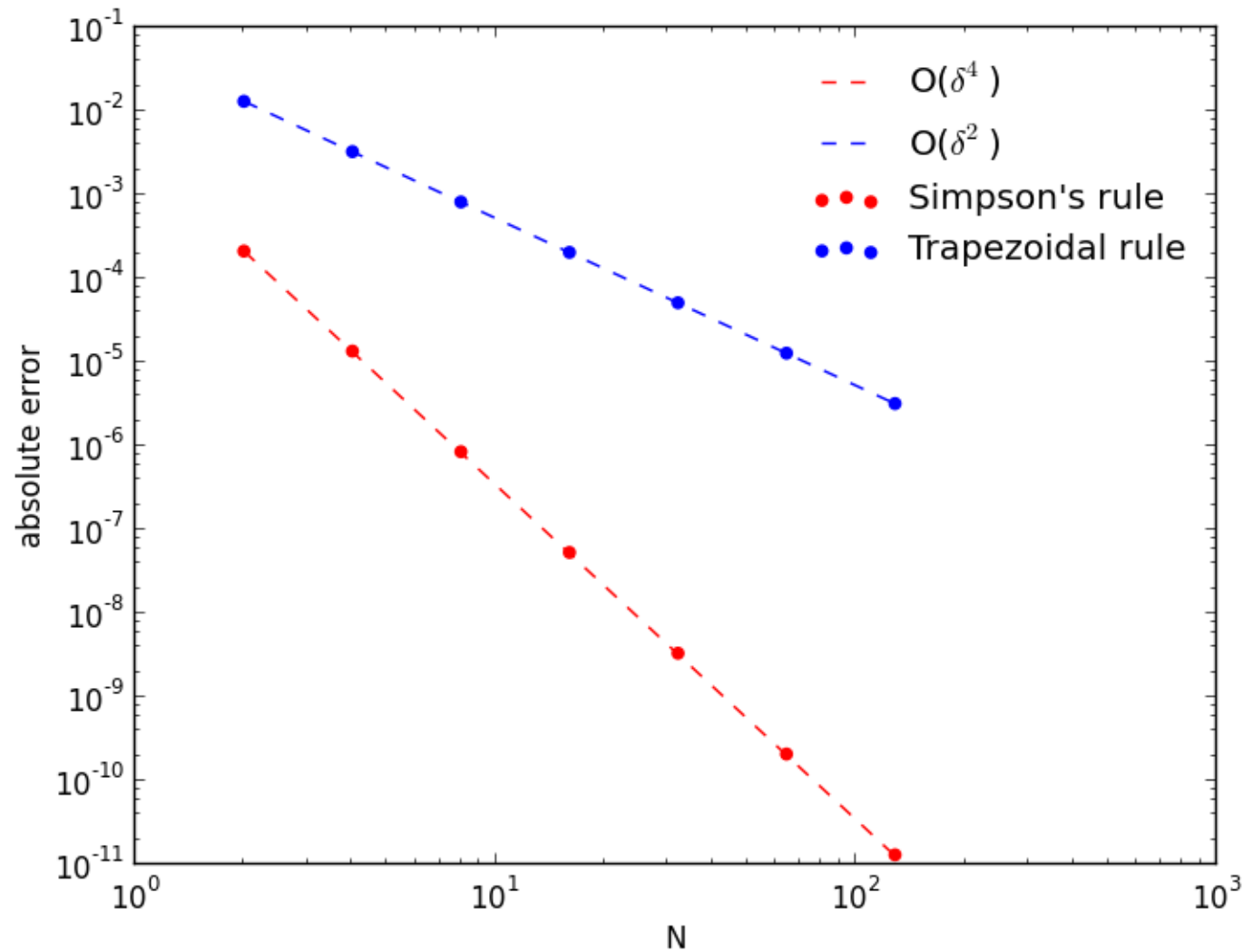
$$\int_a^b f(x)dx = \frac{\Delta x}{2} \sum_{i=0}^{N-1} (f_i + f_{i+1}) + \mathcal{O}(\Delta x^2)$$

- Compound Simpson's

- Integrate pairs of slabs together (requires even number of slabs)

$$\int_a^b f(x)dx = \frac{\Delta x}{3} \sum_{i=0}^{N/2-1} (f_{2i} + 4f_{2i+1} + f_{2i+2}) + \mathcal{O}(\Delta x^4)$$

Compound Integration



$$\int_0^1 e^{-x} dx$$

Always a good idea to check the convergence rate!

Adaptive Integration

- If you know the analytic form of $f(x)$, then you can estimate the error by increasing N
 - Can make use of previous function evaluations (see Garcia)
- Recover Simpson's rule from adaptive trapezoidal (see NR)

Gaussian Quadrature

- Instead of fixed spacing, what if we strategically pick the spacings?

- We want to express

$$\int_a^b f(x)dx \approx w_1 f(x_1) + \dots w_N f(x_N)$$

- w's are weights. We will choose the location of points x_i

Gaussian Quadrature

(Garcia, Ch. 10)

- **Gaussian quadrature**: fundamental theorem

- $q(x)$ is a polynomial of degree N , such that

$$\int_a^b q(x)\rho(x)x^k dx = 0$$

- $k = 0, \dots, N-1$ and $\rho(x)$ is a specified weight function.
- Choose x_1, x_2, \dots, x_N as the roots of the polynomial $q(x)$
- We can write

$$\int_a^b f(x)\rho(x)dx \approx w_1 f(x_1) + \dots w_N f(x_N)$$

and there will be a set of w 's for which the integral is exact if $f(x)$ is a polynomial of degree $< 2N$!

Gaussian Quadrature

(Garcia, Ch. 10)

- The amazing result of this theorem is that by picking the points strategically, we are exact for polynomials up to degree $2N-1$
 - With a fixed grid, of N points, we can fit an $N-1$ degree polynomial
 - Exact integration for $f(x)$ only if it is a polynomial of degree $N-1$ or less
 - If our $f(x)$ is closely approximated by a polynomial of degree $2N-1$, then this will be very accurate.
- Many choices of weighting function, $\rho(x)$, leading to different q 's and x 's and w 's.

Gaussian Quadrature

(Garcia, Ch. 10)

- Example from Garcia:
 - 3-point quadrature
 - This means 3 roots, so $q(x)$ is a cubic
 - Weight function $p(x) = 1$
 - Work in the interval $[-1, 1]$
 - Easy to transform from $[a, b]$ to $[-1, 1]$:

$$x = \frac{1}{2}(b + a) + \frac{1}{2}(b - a)z$$

$$z = \frac{x - \frac{1}{2}(b + a)}{\frac{1}{2}(b - a)}; \quad dx = \frac{1}{2}(b - a)dz$$

$$\int_a^b f(x)dx = \frac{b - a}{2} \int_{-1}^1 f(z)dz$$

Gaussian Quadrature

(Garcia, Ch. 10)

- 3-point quadrature:

- $q(x) = c_0 + c_1x + c_2x^2 + c_3x^3$

- Step 1: Apply the theorem to find the c's:

$$\left. \begin{aligned} \int_{-1}^1 q(x) &= 0 \\ \int_{-1}^1 xq(x) &= 0 \\ \int_{-1}^1 x^2q(x) &= 0 \end{aligned} \right\}$$

This can give 3 equations for the c's, allowing us to find $q(x)$ up to some arbitrary factor.

Alternately, these are the conditions for the Legendre polynomials (the Gram-Schmidt orthogonalization)

$$q(x) = P_3(x) = \frac{5}{2}x^3 - \frac{3}{2}x$$

Gaussian Quadrature

(Garcia, Ch. 10)

- Step 2: find the roots (those are our quadrature points)
 - For

$$q(x) = P_3(x) = \frac{5}{2}x^3 - \frac{3}{2}x$$

we can easily factor this:

$$x = 0, \pm\sqrt{\frac{3}{5}}$$

- This means that our quadrature becomes:

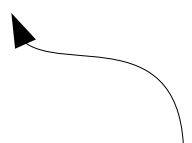
$$\int_{-1}^1 f(x)dx \approx w_1 f(-\sqrt{3/5}) + w_2 f(0) + w_3 f(\sqrt{3/5})$$

Gaussian Quadrature

(Garcia, Ch. 10)

- Step 3: find the weights

- The theorem tells us that with the x 's as the roots, then the proper choice of weights makes the integration exact for polynomials up to degree $2N-1$

$$f(x) = 1 : \int_{-1}^1 dx = w_1 + w_2 + w_3$$


Note the “=”—this is exact

this is: $2 = w_1 + w_2 + w_3$

$$f(x) = x : \int_{-1}^1 x dx = -\sqrt{\frac{3}{5}}w_1 + \sqrt{\frac{3}{5}}w_3 = 0$$

$$f(x) = x^2 : \int_{-1}^1 x^2 dx = \frac{3}{5}w_1 + \frac{3}{5}w_3 = \frac{2}{3}$$

Gaussian Quadrature

(Garcia, Ch. 10)

- These three equations can be solved to find the weights:

$$w_1 = \frac{5}{9}; w_2 = \frac{8}{9}; w_3 = \frac{5}{9}$$

- Therefore, our 3-point quadrature is:

$$\int_{-1}^1 f(x) dx \approx \frac{5}{9} f(-\sqrt{3/5}) + \frac{8}{9} f(0) + \frac{5}{9} f(\sqrt{3/5})$$

Our choice of weight function and integration interval results in the Gauss-Legendre method

Gaussian Quadrature

(Garcia, Ch. 10)

- Example:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2} dy$$

<code>erf(1) (exact):</code>	<code>0.84270079295</code>	
<code>3-point trapezoidal:</code>	<code>0.825262955597</code>	<code>-0.017437837353</code>
<code>3-point Simpson's:</code>	<code>0.843102830043</code>	<code>0.000402037093266</code>
<code>3-point Gauss-Legendre:</code>	<code>0.842690018485</code>	<code>-1.0774465204e-05</code>

Notice how well the Gauss-Legendre does for this integral.

Gaussian Quadrature

- Other quadratures exist:

Interval	$\omega(x)$	Orthogonal polynomials	A & S	For more information, see ...
$[-1, 1]$	1	Legendre polynomials	25.4.29	Section Gauss–Legendre quadrature , above
$(-1, 1)$	$(1-x)^\alpha(1+x)^\beta, \quad \alpha, \beta > -1$	Jacobi polynomials	25.4.33 ($\beta = 0$)	Gauss–Jacobi quadrature
$(-1, 1)$	$\frac{1}{\sqrt{1-x^2}}$	Chebyshev polynomials (first kind)	25.4.38	Chebyshev–Gauss quadrature
$[-1, 1]$	$\sqrt{1-x^2}$	Chebyshev polynomials (second kind)	25.4.40	Chebyshev–Gauss quadrature
$[0, \infty)$	e^{-x}	Laguerre polynomials	25.4.45	Gauss–Laguerre quadrature
$[0, \infty)$	$x^\alpha e^{-x}$	Generalized Laguerre polynomials		Gauss–Laguerre quadrature
$(-\infty, \infty)$	e^{-x^2}	Hermite polynomials	25.4.46	Gauss–Hermite quadrature

(Wikipedia)

- In practice, the roots and weights are tabulated for these out to many numbers of points, so there is no need to compute them.

Multi-dimensional Integration

- For multi-dimensional integration, Monte Carlo methods may be faster (less function evaluations)