

# Math 541 - Numerical Analysis

## Lecture Notes – Quadrature – Part A

Joseph M. Mahaffy,  
(jmahaffy@mail.sdsu.edu)

Department of Mathematics and Statistics  
Dynamical Systems Group  
Computational Sciences Research Center  
San Diego State University  
San Diego, CA 92182-7720

<http://jmahaffy.sdsu.edu>

Spring 2018



## Outline

- 1 **Riemann Integral**
  - Fundamental Theorem of Calculus
  - Definition of Riemann Integral and Midpoint Rule
  - Midpoint Rule for Integration
  - Midpoint Example
- 2 **Interpolation and Polynomial Approximation**
  - Fundamentals
  - Moving Beyond Taylor Polynomials
  - Lagrange Interpolating Polynomials
  - MatLab and Lagrange Polynomials
- 3 **Numerical Integration (Quadrature)**
  - Trapezoidal & Simpson's Rules
  - Newton-Cotes Formulas



## Definite Integral

### Theorem (Fundamental Theorem of Calculus)

Let  $f(x)$  be a continuous function on the interval  $[a, b]$  and assume that  $F(x)$  is any **antiderivative** of  $f(x)$ . The **definite integral**, which gives the **area under the curve** of  $f(x)$  between  $a$  and  $b$ , satisfies the following formula:

$$\int_a^b f(x)dx = F(b) - F(a).$$

- Finding integrals was a significant part of Calculus
- Developed many techniques for solving a variety of integrals
- Many integrals are impossible to solve with classic techniques
- We need **numerical methods** to evaluate these **definite integrals**



## Definition of Riemann Integral

**Definition of Riemann Integral:** The standard integral from Calculus is the **Riemann Integral**

- Let  $f(x)$  be a continuous function in the interval  $[a, b]$
- Partition the interval  $[a, b]$  into  $n$  subintervals  $[x_{i-1}, x_i]$  with  $\Delta x_i = x_i - x_{i-1}$  and  $\Delta x_k$  being the largest
- Let  $c_i$  be some point in the subinterval  $[x_{i-1}, x_i]$
- The  $n^{\text{th}}$  **Riemann sum** is given by

$$S_n = \sum_{i=1}^n f(c_i)\Delta x_i$$

- The **Riemann integral** is defined by

$$\int_a^b f(x)dx = \lim_{\Delta x_k \rightarrow 0} \sum_{i=1}^n f(c_i)\Delta x_i$$



# Midpoint Rule

1

The **Midpoint Rule** is based directly on the **definition of the Riemann integral**.

Suppose that we want to approximate the area under some continuous function  $f(x)$  between  $x = a$  and  $x = b$

- Divide the interval  $[a, b]$  into a number of small intervals
- Assume there are  $n$  evenly spaced intervals (which Riemann sums do not require this restriction)
- Evaluate the function,  $f(x)$ , at the midpoint of any subinterval
- Technically, it is important in the definition of the Riemann integral that one chooses arbitrarily any point in the interval, but that is left to other analysis courses



# Midpoint Rule

2

The **Midpoint Rule** is given by the following:

- Let  $x_0 = a$  and  $x_n = b$  and define  $\Delta x = \frac{b-a}{n}$  with  $x_i = a + i\Delta x$  for  $i = 0, \dots, n$
- This **partitions the interval**  $[a, b]$  into  $n$  subintervals  $[x_{i-1}, x_i]$  each with length  $\Delta x$
- The height of the approximating rectangle is found by evaluating the function at the **midpoint**,  $c_i = \frac{x_i+x_{i-1}}{2}$
- The **area of the rectangle**,  $R_i$ , over the interval  $[x_{i-1}, x_i]$  is given by its height times its width or

$$R_i = f(c_i)\Delta x$$

- The area under  $f(x)$  is approximated by adding the areas of the rectangles

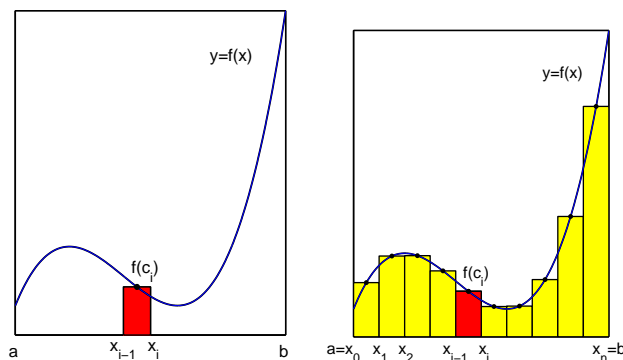
$$S_n = \sum_{i=1}^n f(c_i)\Delta x \approx \int_a^b f(x)dx$$



# Midpoint Rule

3

Figures below show a single rectangle in computing area of the **Riemann Integral** and all of the rectangles using the **Midpoint Rule** for approximating the area under the curve



# Midpoint Rule

4

## Riemann Sums and Riemann Integral

- The **Midpoint Rule** described above is a specialized form of **Riemann sums**
- The more general form of Riemann sums allows the subintervals to have varying lengths,  $\Delta x_i$
- The choice of where the function is evaluated need not be at the midpoint as described above
- The **Riemann integral** is defined using a limiting process, similar to the one described above



# Area under a Curve

1

**Area under a Curve:** Consider the function

$$f(x) = x^3 - 6x^2 + 9x + 2 \quad \text{for } x \in [0, 5]$$

From the **Fundamental Theorem of Calculus** the area under the curve is

$$A_* = \int_0^5 f(x)dx = \left. \frac{x^4}{4} - 2x^3 + \frac{9x^2}{2} + 2x \right|_0^5 = 28.75$$

- Approximate area with rectangles under the curve
- Divide the interval  $x \in [0, 5]$  into even intervals
- Use the midpoint of the interval to get height of the rectangle
- Examine approximation as intervals get smaller



# Area under a Curve

3

**Area under a Curve:** Height of rectangles from the function

$$f(x) = x^3 - 6x^2 + 9x + 2 \quad \text{for } x \in [0, 5]$$

- Width of the rectangles are  $\Delta x = 1$
- Height of rectangles evaluated at midpoints
- Approximate area satisfies

$$A_1 \approx \left( f\left(\frac{1}{2}\right) + f\left(\frac{3}{2}\right) + f\left(\frac{5}{2}\right) + f\left(\frac{7}{2}\right) + f\left(\frac{9}{2}\right) \right) \Delta x = \sum_{i=0}^4 f\left(i + \frac{1}{2}\right) \cdot 1$$

- This gives

$$A_1 \approx \sum_{i=0}^4 \left( \left(i + \frac{1}{2}\right)^3 - 6\left(i + \frac{1}{2}\right)^2 + 9\left(i + \frac{1}{2}\right) + 2 \right) = 28.125$$

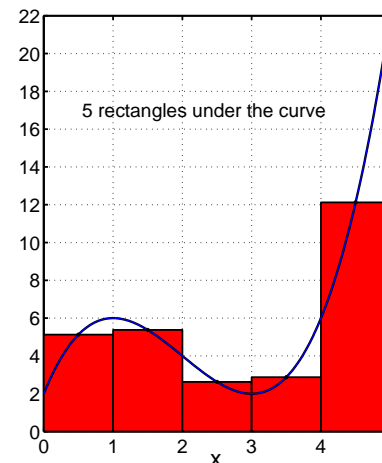
- This is 2.17% less than the actual area



# Area under a Curve

2

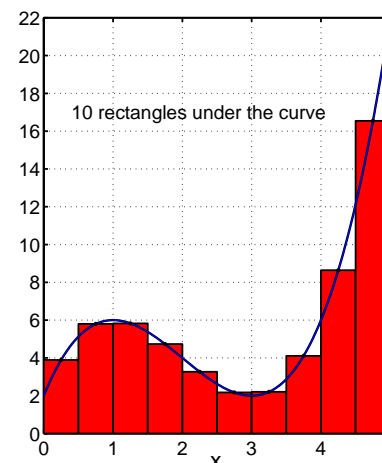
**Area under a Curve** Divide  $x \in [0, 5]$  into 5 intervals



# Area under a Curve

4

**Area under a Curve** Divide  $x \in [0, 5]$  into 10 intervals



## Area under a Curve

5

**Area under a Curve:** Height of rectangles from the function

$$f(x) = x^3 - 6x^2 + 9x + 2 \quad \text{for } x \in [0, 5]$$

- Width of the rectangles are  $\Delta x = \frac{1}{2}$
- Height of rectangles evaluated at midpoints
- Approximate area satisfies

$$A_2 \approx \sum_{i=0}^9 f\left(\frac{i}{2} + \frac{1}{4}\right) \Delta x$$

- This gives

$$A_2 \approx \frac{1}{2} \sum_{i=0}^9 \left( \left(\frac{i}{2} + \frac{1}{4}\right)^3 - 6 \left(\frac{i}{2} + \frac{1}{4}\right)^2 + 9 \left(\frac{i}{2} + \frac{1}{4}\right) + 2 \right) = 28.59375$$

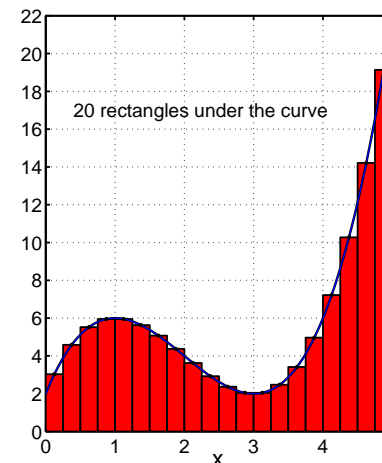
- This is 0.543% less than the actual area

SDSU

## Area under a Curve

6

**Area under a Curve** Divide  $x \in [0, 5]$  into 20 intervals



SDSU

## Area under a Curve

7

**Area under a Curve:** Height of rectangles from the function

$$f(x) = x^3 - 6x^2 + 9x + 2 \quad \text{for } x \in [0, 5]$$

- Width of the rectangles are  $\Delta x = \frac{1}{4}$
- Height of rectangles evaluated at midpoints
- Approximate area satisfies

$$A_3 \approx \sum_{i=0}^{19} f\left(\frac{i}{4} + \frac{1}{8}\right) \Delta x$$

- This gives

$$A_3 \approx \frac{1}{4} \sum_{i=0}^{19} \left( \left(\frac{i}{4} + \frac{1}{8}\right)^3 - 6 \left(\frac{i}{4} + \frac{1}{8}\right)^2 + 9 \left(\frac{i}{4} + \frac{1}{8}\right) + 2 \right) = 28.7109$$

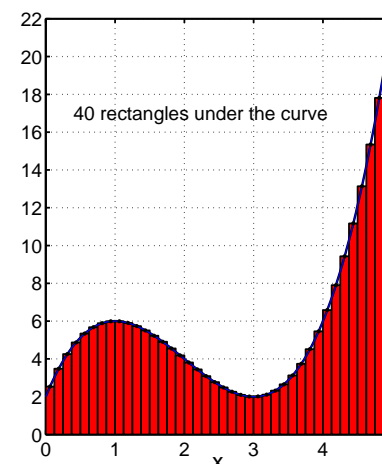
- This is 0.135% less than the actual area

SDSU

## Area under a Curve

8

**Area under a Curve** Divide  $x \in [0, 5]$  into 40 intervals



SDSU

## Area under a Curve

9

**Area under a Curve:** Height of rectangles from the function

$$f(x) = x^3 - 6x^2 + 9x + 2 \quad \text{for } x \in [0, 5]$$

- Width of the rectangles are  $\Delta x = \frac{1}{8}$
- Height of rectangles evaluated at midpoints
- Approximate area satisfies

$$A_4 \approx \sum_{i=0}^{39} f\left(\frac{i}{8} + \frac{1}{16}\right) \Delta x$$

- This gives

$$A_4 \approx \frac{1}{8} \sum_{i=0}^{39} \left( \left(\frac{i}{8} + \frac{1}{16}\right)^3 - 6 \left(\frac{i}{8} + \frac{1}{16}\right)^2 + 9 \left(\frac{i}{8} + \frac{1}{16}\right) + 2 \right) = 28.7402$$

- This is 0.034% less than the actual area

SDSU

## Numerical Methods for Integration

### Numerical Methods for Integration

- As noted before, many integrals cannot be solved exactly, so **numerical methods** need to be used to estimate **definite integrals**

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

- The **Midpoint Rule** is an approximation based on the definition of a **Riemann integral**
- The **Midpoint Rule** is **NOT** a very efficient way to estimate the area under the curve
- Once again we turn to **polynomials** to approximate our functions and improve the convergence of the **numerical routine** to the actual value of the **definite integral**

SDSU

## Area under a Curve

10

**Area under a Curve:** The actual area is,

$$A_* = \int_0^5 f(x) dx = 28.75$$

The approximate areas were

$\Delta x_1 = 1$	$\Delta x_2 = \frac{1}{2}$	$\Delta x_3 = \frac{1}{4}$	$\Delta x_4 = \frac{1}{8}$
$A_1 = 28.125$	$A_2 = 28.59375$	$A_3 = 28.7109$	$A_4 = 28.7402$

The error ratio for this example is

$$\frac{|A_{n+1} - A_*|}{|A_n - A_*|} \approx 0.25$$

Thus, as the stepsize decreases by  $\frac{1}{2}$ , the error in the approximate area decreases by a factor of  $\frac{1}{4}$

We will demonstrate that the error of the **Midpoint Rule** is  $\mathcal{O}((\Delta x)^2)$ , depending on the stepsize

SDSU

## Interpolation and Polynomial Approximation

### Interpolation and Polynomial Approximation

- Polynomials provide “nice” smooth functions for approximations
- **Taylor’s series** give excellent estimates near a point
- For integration, we need to extend over an interval
- **Interpolating polynomials** have many applications to fit functions or data at various  $x$  values

SDSU

## Weierstrass Approximation Theorem

The following theorem is the basis for polynomial approximation:

### Theorem (Weierstrass Approximation Theorem)

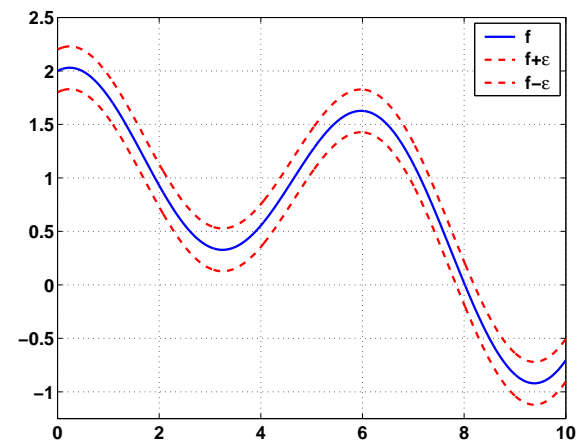
Suppose  $f \in C[a, b]$ . Then for every  $\epsilon > 0$  there exists a polynomial  $P(x) : |f(x) - P(x)| < \epsilon$ , for all  $x \in [a, b]$ .

**Note:** The bound is *uniform*, i.e., valid for all  $x$  in the interval.

**Note:** The theorem says nothing about how to find the polynomial, or about its order.



## Illustrated: Weierstrass Approximation Theorem



**Figure:** Weierstrass approximation Theorem guarantees that we (maybe with substantial work) can find a polynomial which fits into the “tube” around the function  $f$ , no matter how thin we make the tube.



## Candidates: the Taylor Polynomials???

### Natural Question:

Are our old friends, the Taylor Polynomials, good candidates for polynomial interpolation?

### Answer:

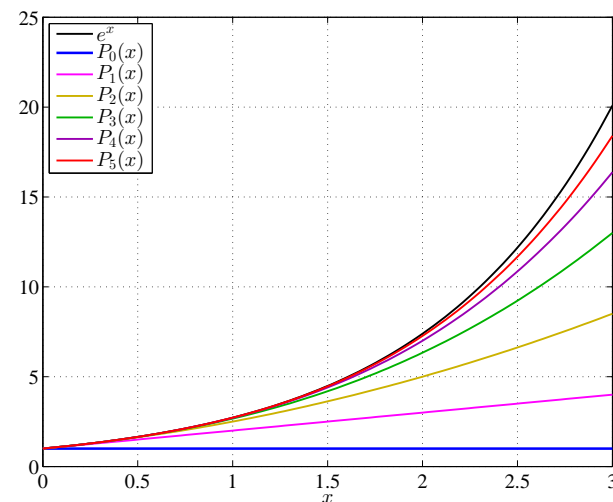
**No.** The Taylor expansion works very hard to be accurate in the neighborhood of *one point*. But we want to fit data at many points (in an extended interval).

[Next slide: The approximation is great near the expansion point  $x_0 = 0$ , but get progressively worse as we get further away from the point, even for the higher degree approximations.]



## Taylor Approximation of $e^x$ on the Interval $[0, 3]$

We learned that  $e^x$  outgrows any polynomial



## Interpolation: Lagrange Polynomials

**Idea:** Instead of working hard at *one point*, we will prescribe a number of points through which the polynomial must pass.

Consider a function that passes through the points  $(x_0, f(x_0))$  and  $(x_1, f(x_1))$ . From techniques of algebra, we have the slope

$$m = \frac{f(x_1) - f(x_0)}{x_1 - x_0},$$

so the point slope form of a line gives

$$y(x) - f(x_0) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0).$$

This is rearranged to give

$$y(x) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) + f(x_0),$$

$$y(x) = f(x_1) \frac{(x - x_0)}{x_1 - x_0} + f(x_0) \frac{(x - x_0)}{x_0 - x_1} + f(x_0) \frac{(x_0 - x_1)}{x_0 - x_1},$$

$$y(x) = f(x_1) \frac{(x - x_0)}{x_1 - x_0} + f(x_0) \frac{(x - x_1)}{x_0 - x_1},$$

SDSU

## Interpolation: Lagrange Polynomials

From the previous slide we have:

$$y(x) = f(x_1) \frac{(x - x_0)}{x_1 - x_0} + f(x_0) \frac{(x - x_1)}{x_0 - x_1}.$$

If we define:

$$L_0(x) = \frac{x - x_1}{x_0 - x_1}, \quad L_1(x) = \frac{x - x_0}{x_1 - x_0},$$

then we obtain the interpolating polynomial

$$P(x) = L_0(x)f(x_0) + L_1(x)f(x_1),$$

with  $P(x_0) = f(x_0)$ , and  $P(x_1) = f(x_1)$ .

–  $P(x)$  is the unique linear polynomial passing through  $(x_0, f(x_0))$  and  $(x_1, f(x_1))$ .

SDSU

## An $n$ -degree polynomial passing through $n + 1$ points

We are going to construct a polynomial passing through the points  $(x_0, f(x_0))$ ,  $(x_1, f(x_1))$ ,  $(x_2, f(x_2))$ ,  $\dots$ ,  $(x_n, f(x_n))$ .

We define  $L_{n,k}(x)$ , the **Lagrange coefficients**:

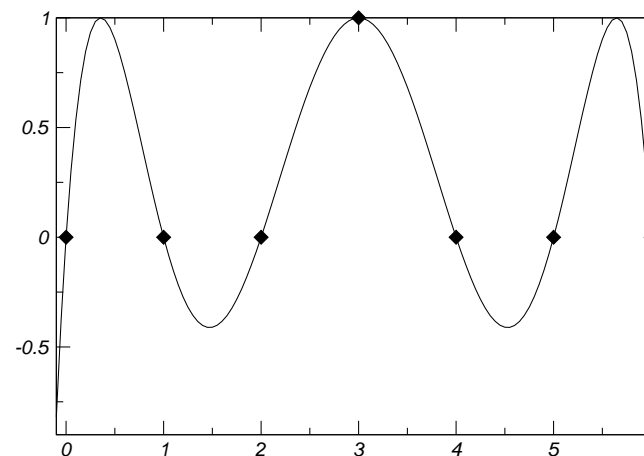
$$L_{n,k}(x) = \prod_{i=0, i \neq k}^n \frac{x - x_i}{x_k - x_i} = \frac{x - x_0}{x_k - x_0} \dots \frac{x - x_{k-1}}{x_k - x_{k-1}} \frac{x - x_{k+1}}{x_k - x_{k+1}} \dots \frac{x - x_n}{x_k - x_n},$$

which have the properties

$$L_{n,k}(x_k) = 1; \quad L_{n,k}(x_i) = 0, \quad \text{for all } i \neq k.$$

SDSU

## Example of $L_{n,k}(x)$



This is  $L_{6,3}(x)$ , for the points  $x_i = i$ ,  $i = 0, \dots, 6$ .

SDSU

## The $n^{\text{th}}$ Lagrange Interpolating Polynomial

We use  $L_{n,k}(x)$ ,  $k = 0, \dots, n$  as building blocks for the Lagrange interpolating polynomial:

$$P(x) = \sum_{k=0}^n f(x_k)L_{n,k}(x),$$

which has the property

$$P(x_i) = f(x_i), \quad \text{for all } i = 0, \dots, n.$$

This is the **unique**  $n^{\text{th}}$  degree polynomial passing through the points  $(x_i, f(x_i))$ ,  $i = 0, \dots, n$ .

SDSU

## Error bound for the Lagrange interpolating polynomial

Suppose  $x_i$ ,  $i = 0, \dots, n$  are distinct numbers in the interval  $[a, b]$ , and  $f \in C^{n+1}[a, b]$ . Then for all  $x \in [a, b]$  there exists  $\xi(x) \in (a, b)$  so that:

$$f(x) = P_{\text{Lagrange}}(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^n (x - x_i),$$

where  $P_{\text{Lagrange}}(x)$  is the  $n^{\text{th}}$  Lagrange interpolating polynomial.

Compare with the error formula for Taylor polynomials

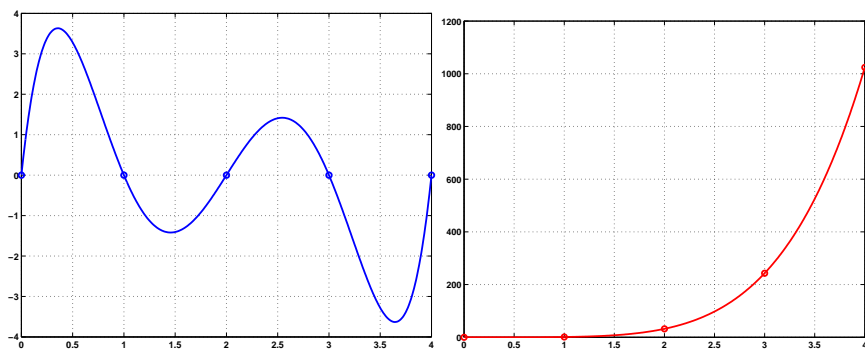
$$f(x) = P_{\text{Taylor}}(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!} (x - x_0)^{n+1},$$

**Problem:** Applying the error term may be difficult...

SDSU

## The Lagrange and Taylor Error Terms

Just to get a feeling for the non-constant part of the error terms in the Lagrange and Taylor approximations, we plot those parts on the interval  $[0, 4]$  with interpolation points  $x_i = i$ ,  $i = 0, 1, \dots, 4$ :



**Figure:** [LEFT] The non-constant error terms for the Lagrange interpolation oscillates in the interval  $[-4, 4]$  (and takes the value zero at the node point  $x_k$ ), and [RIGHT] the non-constant error term for the Taylor extrapolation grows in the interval  $[0, 1024]$ .

SDSU

## MatLab and Lagrange Polynomials

**Example:** Find the **Lagrange polynomial** through the points:

$$(0, -5), \quad (1, -6), \quad (2, -1), \quad \text{and} \quad (3, 16).$$

The **Lagrange polynomial** satisfies

$$\begin{aligned} P(x) &= \frac{(x-1)(x-2)(x-3)}{(0-1)(0-2)(0-3)}(-5) + \frac{x(x-2)(x-3)}{(1-0)(1-2)(1-3)}(-6) \\ &\quad + \frac{x(x-1)(x-3)}{(2-0)(2-1)(2-3)}(-1) + \frac{x(x-1)(x-2)}{(3-0)(3-1)(3-2)}(16) \\ &= x^3 - 2x - 5 \end{aligned}$$

This example was reverse engineered to have clean numbers

SDSU



## MatLab Lagrange Program

Below is a code for accepting vector data  $x$  and  $y$  and generating the **Lagrange polynomial**

It outputs points on this polynomial at  $(u(k), v(k))$

```
1 function v = polyinterp(x,y,u)
2 % Creates Lagrange polynomial
3 n = length(x);
4 v = zeros(size(u));
5 for k = 1:n
6     w = ones(size(u));
7     for j = [1:k-1 k+1:n]
8         w = (u-x(j))./(x(k)-x(j)).*w;
9     end
10    v = v + w*y(k);
11 end
```

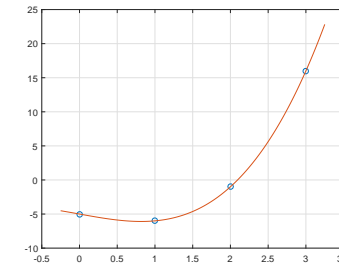


## MatLab and Lagrange Polynomials

**Example (with MatLab):** Our example satisfies  
 $x = 0:3;$   $y = [-5 -6 -1 16];$

We enter closely spaced points,  $u$ , the function `polyinterp`, and plot the results

```
u = -0.25:0.01:3.25;
v = polyinterp(x,y,u);
plot((x,y,'o',u,v,'-');grid;
```



## MatLab and Lagrange Polynomials

**Example (with MatLab):** Continuing our example we can use the **symbolic package** in MatLab to obtain the polynomial expression:

```
symx = sym('x')
```

The polynomial is given by:

```
P = polyinterp(x,y,symx)
P = (x*(x - 1)*(x - 3))/2 + 5*(x/2 - 1)*(x/3 - ...
1)*(x - 1)+ (16*x*(x/2 - 1/2)*(x - 2))/3 - ...
6*x*(x/2 - 3/2)*(x - 2)
```

This is simplified with

```
P = simplify(P)
P = x^3 - 2*x - 5
```



## Vandermonde Matrix and Interpreting Polynomial

**Alternate Scheme:** Suppose we want an interpreting polynomial of the form:

$$P(x) = c_1x^{n-1} + c_2x^{n-2} + \dots + c_{n-1}x + c_n$$

Given data points  $x = [x_1, \dots, x_n]$  and  $y = [y_1, \dots, y_n]$  we can obtain the coefficients  $c_1, \dots, c_n$  by solving the system:

$$\begin{pmatrix} x_1^{n-1} & x_1^{n-2} & \dots & x_1 & 1 \\ x_2^{n-1} & x_2^{n-2} & \dots & x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n^{n-1} & x_n^{n-2} & \dots & x_n & 1 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

This system  $Vc = y$  contains the important **Vandermonde matrix**,  $V$



## Vandermonde Matrix and MatLab

Given the data  $x = [x_1, \dots, x_n]$ , the elements of the **Vandermonde matrix**,  $V$ , satisfy

$$v_{k,j} = x_k^{n-j}$$

**MatLab** has the function `vander`, which generates the **Vandermonde matrix**,  $V$

For our example above,  $\mathbf{x} = 0:3$ ;  $\mathbf{y} = [-5 \ -6 \ -1 \ 16]$ ;  
 $V = \text{vander}(\mathbf{x})$  produces

$$V = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 8 & 4 & 2 & 1 \\ 27 & 9 & 3 & 1 \end{pmatrix}$$

Then  $\mathbf{c} = V \setminus \mathbf{y}'$  produces  $\mathbf{c} = [1, 0, -2, -5]^t$  or

$$P(x) = x^3 - 2x - 5$$

## Numerical Quadrature – Basics

### Numerical Quadrature: Basics

- Integration is valuable in many **applications** – often the anti-derivative is unavailable
- Introduction showed the definition of the **Riemann integral**
  - **Midpoint rule** directly uses the definition with even intervals and function evaluations at the midpoint of the subintervals
  - The **convergence** of this method appeared to be  $\mathcal{O}((\Delta x)^2)$
- Can we do better with interpolating functions on the subintervals  $[x_j, x_{j+1}]$ ?

## Numerical Quadrature – Basics

There are **two** primary means of improving **Numerical integration**

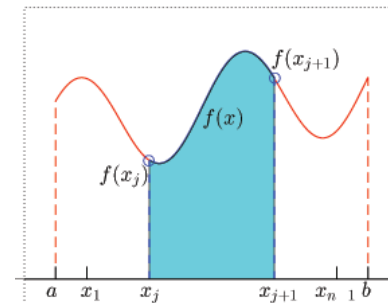
- 1 If  $a = x_0 < x_1 < \dots < x_{n-1} < x_n = b$ , then **properties of the integral** give

$$\int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \dots + \int_{x_{n-1}}^{x_n} f(x)dx.$$

We create **composite integrals** and choose appropriate  $x_i$ 's, which subdivide our function  $f(x)$  into  $n$  subintervals with each subinterval providing a smaller domain and better approximation of  $f$  on that subinterval.

- 2 Take a particular subinterval, then partition that subinterval further to obtain a reasonable approximation of  $f(x)$  on the subinterval by an **interpolating polynomial**, which is precisely integrable and has a known error bound.

## Numerical Quadrature – Basics



Our aim is to obtain the greatest accuracy approximating the integral with the minimum amount of computation

- We can vary the spacing  $x_j$ , not necessarily uniform
- We can alter how  $f(x)$  is approximated – Using polynomials, which are exactly integrable

## Numerical Quadrature – Single Interval

We begin our analysis with the second point above (avoiding the **composite integral** for now)

We focus on a single interval and consider **interpolating polynomials** approximating  $f(x)$  on the single interval

The basic idea is to replace integration by a clever summation:

$$\int_a^b f(x) dx \rightarrow \sum_{i=0}^n a_i f_i,$$

where  $a \leq x_0 < x_1 < \dots < x_n \leq b$ ,  $f_i = f(x_i)$ .

**The coefficients  $a_i$  and the nodes  $x_i$  are to be selected.**

Various means of selecting  $a_i$  and  $x_i$  alter the efficiency and accuracy of our **algorithm**

## Building Integration Schemes with Lagrange Polynomials

Given the nodes  $\{x_0, x_1, \dots, x_n\}$  we can use the **Lagrange interpolating polynomial**

$$P_n(x) = \sum_{i=0}^n f_i L_{n,i}(x), \quad \text{with error} \quad E_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^n (x - x_i)$$

to obtain

$$\int_a^b f(x) dx = \underbrace{\int_a^b P_n(x) dx}_{\text{The Approximation}} + \underbrace{\int_a^b E_n(x) dx}_{\text{The Error Estimate}}$$

## Identifying the Coefficients

The **Lagrange interpolating polynomials** are readily integrated to give the weighting coefficients  $a_i$

$$\int_a^b P_n(x) dx = \int_a^b \sum_{i=0}^n f_i L_{n,i}(x) dx = \sum_{i=0}^n f_i \underbrace{\int_a^b L_{n,i}(x) dx}_{a_i} = \sum_{i=0}^n f_i a_i.$$

Hence we write

$$\int_a^b f(x) dx \approx \sum_{i=0}^n a_i f_i$$

with error given by

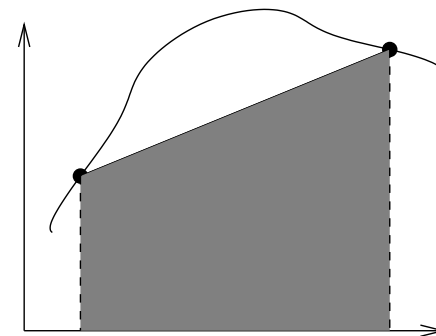
$$E(f) = \int_a^b E_n(x) dx = \int_a^b \frac{f^{(n+1)}(\xi(x))}{(n+1)!} \prod_{i=0}^n (x - x_i) dx.$$

## Example 1: Trapezoidal Rule

1 of 3

Let  $a = x_0 < x_1 = b$ , and use the linear interpolating polynomial

$$P_1(x) = f_0 \left[ \frac{x - x_1}{x_0 - x_1} \right] + f_1 \left[ \frac{x - x_0}{x_1 - x_0} \right].$$



## Example 1: Trapezoidal Rule

2 of 3

Then

$$\int_a^b f(x) dx = \int_{x_0}^{x_1} \left[ f_0 \left[ \frac{x - x_1}{x_0 - x_1} \right] + f_1 \left[ \frac{x - x_0}{x_1 - x_0} \right] \right] dx + \frac{1}{2} \int_{x_0}^{x_1} f''(\xi(x))(x - x_0)(x - x_1) dx.$$

The error term (use the Weighted Mean Value Theorem):

$$\int_{x_0}^{x_1} f''(\xi(x))(x - x_0)(x - x_1) dx = f''(\xi) \int_{x_0}^{x_1} (x - x_0)(x - x_1) dx = f''(\xi) \left[ \frac{x^3}{3} - \frac{x_1 + x_0}{2} x^2 + x_0 x_1 x \right]_{x_0}^{x_1} = -\frac{h^3}{6} f''(\xi).$$

where  $h = x_1 - x_0 = b - a$ .



## Example 1: Trapezoidal Rule

3 of 3

Hence,

$$\int_a^b f(x) dx = \left[ f_0 \left[ \frac{(x - x_1)^2}{2(x_0 - x_1)} \right] + f_1 \left[ \frac{(x - x_0)^2}{2(x_1 - x_0)} \right] \right]_{x_0}^{x_1} - \frac{h^3}{12} f''(\xi) = \frac{(x_1 - x_0)}{2} [f_0 + f_1] - \frac{h^3}{12} f''(\xi)$$

$$\int_a^b f(x) dx = h \left[ \frac{f(x_0) + f(x_1)}{2} \right] - \frac{h^3}{12} f''(\xi), \quad h = b - a.$$



## Example 2a: Simpson's Rule (sub-optimal error bound)

Let  $x_0 = a$ ,  $x_1 = \frac{a+b}{2}$ ,  $x_2 = b$ , let  $h = \frac{b-a}{2}$  and use the *quadratic interpolating polynomial*

$$\int_a^b f(x) dx = \int_{x_0}^{x_2} \left[ f(x_0) \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + f(x_1) \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + f(x_2) \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \right] dx + \int_{x_0}^{x_2} \frac{(x - x_0)(x - x_1)(x - x_2)}{6} f^{(3)}(\xi(x)) dx \dots$$

$$\int_a^b f(x) dx = h \left[ \frac{f(x_0) + 4f(x_1) + f(x_2)}{3} \right] + \mathcal{O}(h^4 f^{(3)}(\xi)).$$



## Example 2b: Simpson's Rule (optimal error bound)

The optimal error bound for Simpson's rule can be obtained by Taylor expanding  $f(x)$  about the mid-point  $x_1$ :

$$f(x) = f(x_1) + f'(x_1)(x - x_1) + \frac{f''(x_1)}{2}(x - x_1)^2 + \frac{f'''(x_1)}{6}(x - x_1)^3 + \frac{f^{(4)}(\xi(x))}{24}(x - x_1)^4,$$

then formally integrating this expression, to get:

$$\int_a^b \left[ f(x_1) + f'(x_1)(x - x_1) + \frac{f''(x_1)}{2}(x - x_1)^2 + \frac{f'''(x_1)}{6}(x - x_1)^3 + \frac{f^{(4)}(\xi(x))}{24}(x - x_1)^4 \right] dx.$$

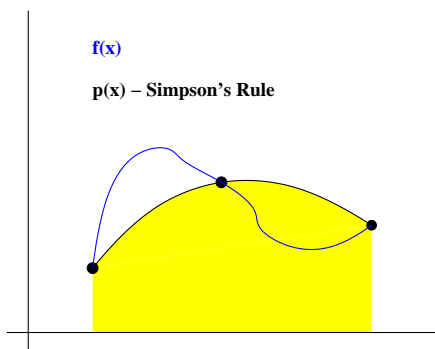
After use of the weighted mean value theorem, and the approximation  $f''(x_1) = \frac{1}{h^2}[f(x_0) - 2f(x_1) + f(x_2)] - \frac{h^2}{12} f^{(4)}(\xi)$ , and a whole lot of algebra (**Yanofsky - UCLA Notes**) we end up with

$$\int_{x_0}^{x_2} f(x) dx = h \left[ \frac{f(x_0) + 4f(x_1) + f(x_2)}{3} \right] - \frac{h^5}{90} f^{(4)}(\xi).$$



## Example 2: Simpson's Rule

$$\int_a^b f(x) dx = h \left[ \frac{f(x_0) + 4f(x_1) + f(x_2)}{3} \right] + \mathcal{O}(h^5 f^{(4)}(\xi)).$$



SDSU

## Integration Examples

$f(x)$	$[a, b]$	$\int_a^b f(x) dx$	Trapezoidal	Error	Simpson	Error
$x$	$[0, 1]$	$1/2$	0.5	0	0.5	0
$x^2$	$[0, 1]$	$1/3$	0.5	0.16667	0.33333	0
$x^3$	$[0, 1]$	$1/4$	0.5	0.25000	0.25000	0
$x^4$	$[0, 1]$	$1/5$	0.5	0.30000	0.20833	0.0083333
$e^x$	$[0, 1]$	$e - 1$	1.8591	0.14086	1.7189	0.0005793

The Trapezoidal rule gives exact solutions for linear functions. — The error term contains a second derivative.

Simpson's rule gives exact solutions for polynomials of degree less than 4. — The error term contains a fourth derivative.

SDSU

## Degree of Accuracy (Precision)

### Definition (Degree of Accuracy)

The **Degree of Accuracy**, or **precision**, of a quadrature formula is the largest positive integer  $n$  such that the formula is exact for  $x^k$  for all  $k = 0, 1, \dots, n$ .

With this definition:

Scheme	Degree of Accuracy
Trapezoidal	1
Simpson's	3

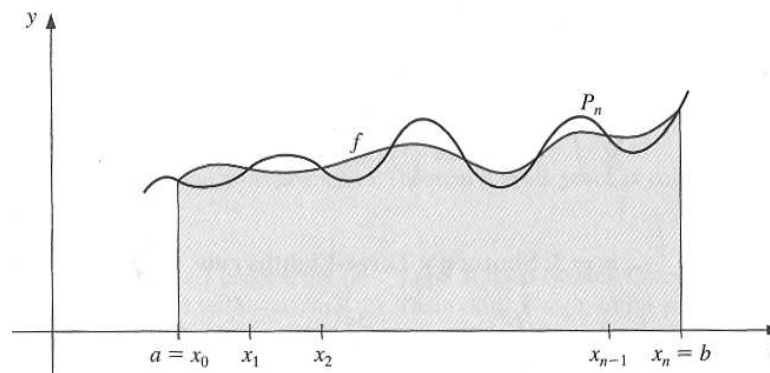
Trapezoidal and Simpson's are examples of a class of methods known as *Newton-Cotes formulas*.

SDSU

## Newton-Cotes Formulas — Two Types

Closed

**Closed** The  $(n + 1)$  point closed NCF uses nodes  $x_i = x_0 + ih$ ,  $i = 0, 1, \dots, n$ , where  $x_0 = a$ ,  $x_n = b$  and  $h = (b - a)/n$ . It is called closed since the endpoints are included as nodes.

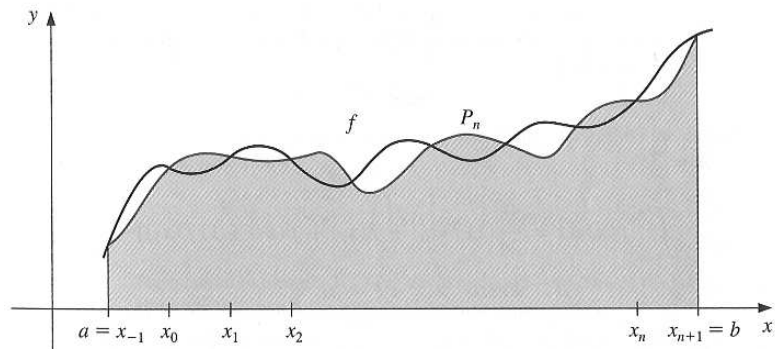


SDSU

Newton-Cotes Formulas — Two Types

Open

**Open** The  $(n + 1)$  point open NCF uses nodes  $x_i = x_0 + ih, i = 0, 1, \dots, n$  where  $h = (b-a)/(n+2)$  and  $x_0 = a+h, x_n = b-h$ . (We label  $x_{-1} = a, x_{n+1} = b$ .)



SDSU

Closed Newton-Cotes Formulas

The approximation is

$$\int_a^b f(x) dx \approx \sum_{i=0}^n a_i f(x_i),$$

where

$$a_i = \int_{x_0}^{x_n} L_{n,i}(x) dx = \int_{x_0}^{x_n} \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)} dx.$$

**Note:** The Lagrange polynomial  $L_{n,i}(x)$  models a function which takes the value 0 at all  $x_j (j \neq i)$ , and 1 at  $x_i$ . Hence, the coefficient  $a_i$  captures the integral of a function, which is 1 at  $x_i$  and zero at the other node points.

SDSU

Closed Newton-Cotes Formulas — Error

Theorem

Suppose that  $\sum_{i=0}^n a_i f(x_i)$  denotes the  $(n + 1)$  point closed Newton-Cotes formula with  $x_0 = a, x_n = b$ , and  $h = (b - a)/n$ . Then there exists  $\xi \in (a, b)$  for which

$$\int_a^b f(x) dx = \sum_{i=0}^n a_i f(x_i) + \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!} \int_0^n t^2(t-1) \cdots (t-n) dt,$$

if  $n$  is even and  $f \in C^{n+2}[a, b]$ , and

$$\int_a^b f(x) dx = \sum_{i=0}^n a_i f(x_i) + \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!} \int_0^n t(t-1) \cdots (t-n) dt,$$

if  $n$  is odd and  $f \in C^{n+1}[a, b]$ .

Note that when  $n$  is an even integer, the degree of precision is  $(n + 1)$ . When  $n$  is odd, the degree of precision is only  $n$ .

SDSU

Closed Newton-Cotes Formulas — Examples

**n = 1: Trapezoid Rule**

$$\frac{h}{2} [f(x_0) + f(x_1)] - \frac{h^3}{12} f''(\xi)$$

**n = 2: Simpson's Rule**

$$\frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{90} f^{(4)}(\xi)$$

**n = 3: Simpson's  $\frac{3}{8}$ -Rule**

$$\frac{3h}{8} [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] - \frac{3h^5}{80} f^{(4)}(\xi)$$

**n = 4: Boole's Rule**

$$\frac{2h}{45} [7f(x_0) + 32f(x_1) + 12f(x_2) + 32f(x_3) + 7f(x_4)] - \frac{8h^7}{945} f^{(6)}(\xi)$$

SDSU

## Open Newton-Cotes Formulas

The approximation is

$$\int_a^b f(x) dx = \int_{x_{-1}}^{x_{n+1}} f(x) dx \approx \sum_{i=0}^n a_i f(x_i),$$

where

$$a_i = \int_{x_{-1}}^{x_{n+1}} L_{n,i}(x) dx = \int_{x_0}^{x_n} \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)} dx.$$



## Open Newton-Cotes Formulas — Examples

**n = 0: Midpoint Rule**

$$2hf(x_0) + \frac{h^3}{3} f''(\xi)$$

**n = 1: Trapezoid Method**

$$\frac{3h}{2} [f(x_0) + f(x_1)] + \frac{3h^3}{4} f''(\xi)$$

**n = 2: Milne's Rule**

$$\frac{4h}{3} [2f(x_0) - f(x_1) + 2f(x_2)] + \frac{14h^5}{45} f^{(4)}(\xi)$$

**n = 3: No Name**

$$\frac{5h}{24} [11f(x_0) + f(x_1) + f(x_2) + 11f(x_3)] + \frac{95h^5}{144} f^{(4)}(\xi)$$



## Open Newton-Cotes Formulas — Error

### Theorem

Suppose that  $\sum_{i=0}^n a_i f(x_i)$  denotes the  $(n+1)$  point open Newton-Cotes formula with  $x_{-1} = a$ ,  $x_{n+1} = b$ , and  $h = (b-a)/(n+2)$ . Then there exists  $\xi \in (a, b)$  for which

$$\int_a^b f(x) dx = \sum_{i=0}^n a_i f(x_i) + \frac{h^{n+3} f^{(n+2)}(\xi)}{(n+2)!} \int_{-1}^{n+1} t^2(t-1)\cdots(t-n) dt,$$

if  $n$  is even and  $f \in C^{n+2}[a, b]$ , and

$$\int_a^b f(x) dx = \sum_{i=0}^n a_i f(x_i) + \frac{h^{n+2} f^{(n+1)}(\xi)}{(n+1)!} \int_{-1}^{n+1} t(t-1)\cdots(t-n) dt,$$

if  $n$  is odd and  $f \in C^{n+1}[a, b]$ .

Note that when  $n$  is an even integer, the degree of precision is  $(n+1)$ . When  $n$  is odd, the degree of precision is only  $n$ .



## Divide and Conquer!

Say you want to compute:

$$\int_0^{100} f(x) dx.$$

Is it a Good Idea™ to directly apply your favorite Newton-Cotes formula to this integral!?!?

**No!**

With the closed 5-point NCF, we have  $h = 25$  and  $h^5/90 \sim 10^5$  so even with a bound on  $f^{(6)}(\xi)$  the error will be large.

**Better:** Apply the closed 5-point NCF to the integrals

$$\int_{4i}^{4(i+1)} f(x) dx, \quad i = 0, 1, \dots, 24$$

then sum. **“Composite Numerical Integration”**

