Math 242 Lab 4
Numerical Integration

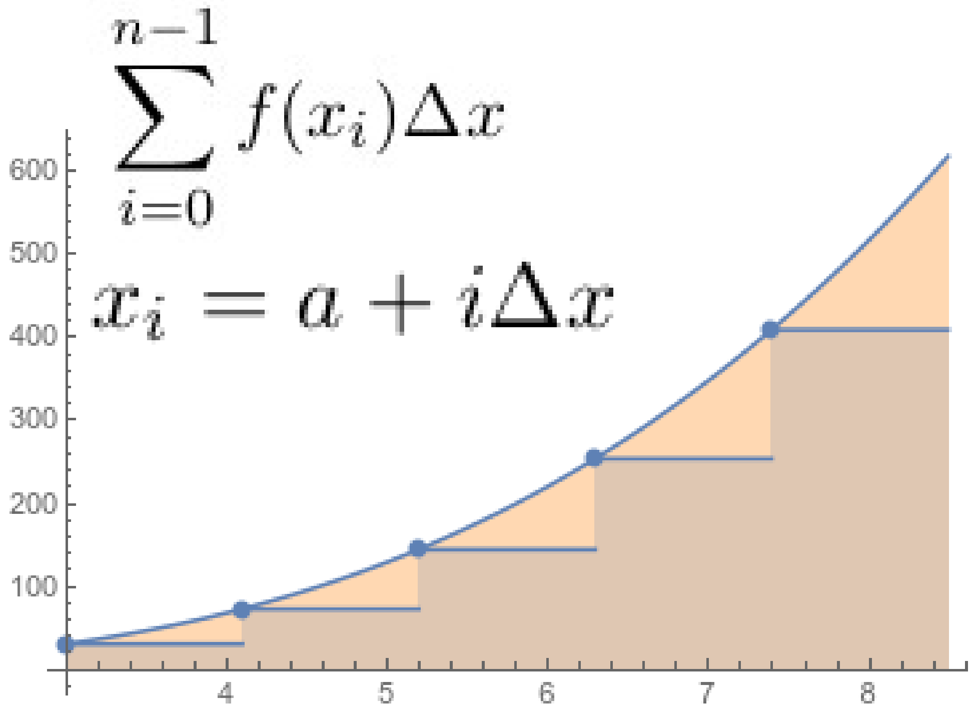
Li-An Chen
Department of Mathematical Sciences, University of Delaware
September 29, 2020

Lab Assignment

- Complete ALL Lab Assignment Questions (with codes, computation results, and essay questions in page 4~6)
- Submit “lastnameLab04.nb” and “lastnameLab04.pdf” **(File->Save As→pdf)** on Canvas
- Deadline: **Tomorrow 11:59pm**
- Correct computation results (without codes) are available on Canvas→Files→Lab→Lab_04_Numerical Integration→lab04_examples_hints

Set up (page 2 Example 1)

- Clear[a, b, n, width]
- a = 3;
- b = 8.5;
- n = 5;
- width = (b – a)/n;
- Clear[f]
- f[x_] := x ^3 + 2



Sum (page 2 Example 1)

- **Sum[(summand) , {i,start,end}]**
- Sum[f[a + i*width]*width, {i, 0, n – 1}]
- This is the same as

  width*Sum[f[a + i*width], {i, 0, n – 1}]

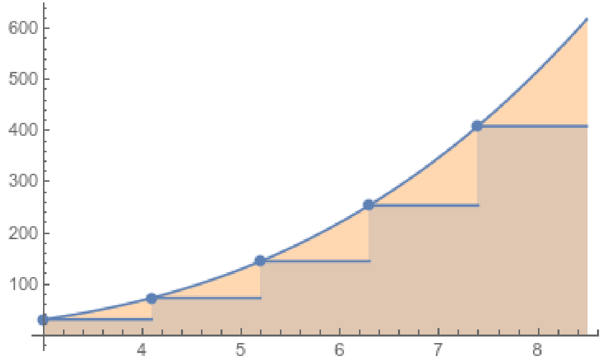
  because “width” doesn’t depend on “i”.

$$\sum_{i=0}^{n-1} f(x_i)\Delta x$$

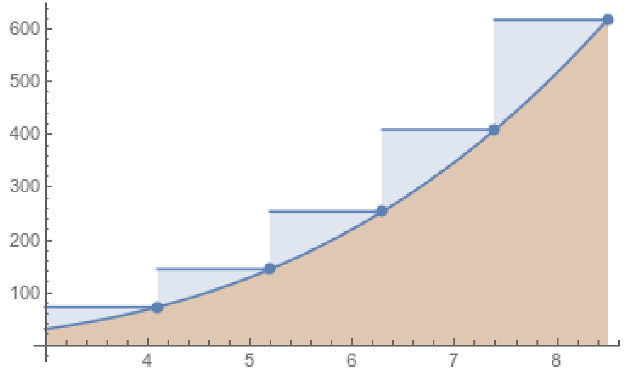
$$x_i = a + i\Delta x$$

Numerical Integration

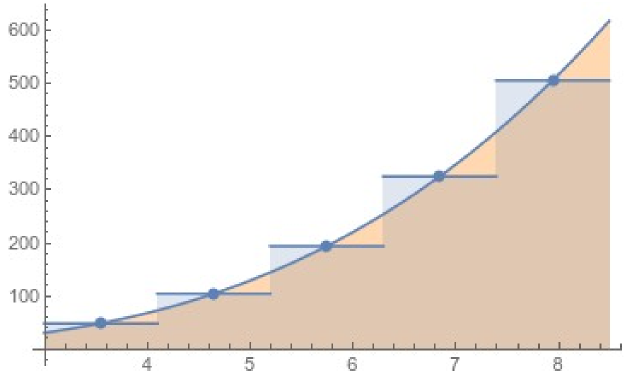
- Example 1: Left endpoint (rectangle)



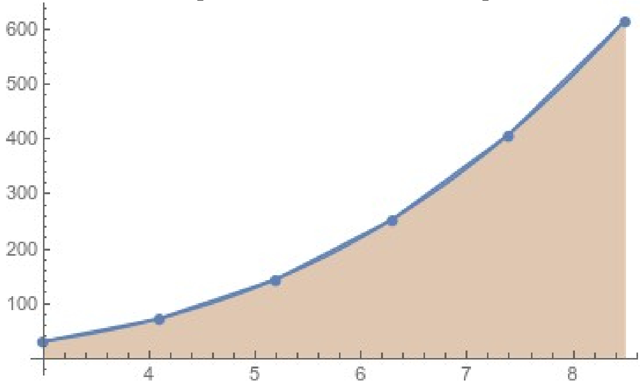
- Example 2: Right endpoint (rectangle)



- Example 3: Midpoint (rectangle)

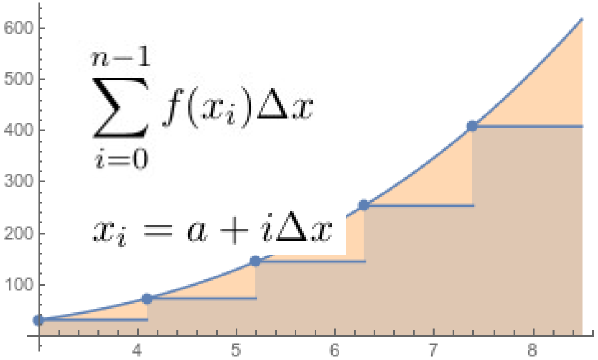


- Example 4: Trapezoid

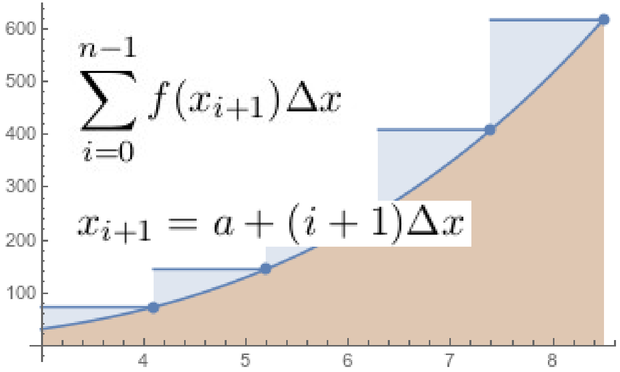


Numerical Integration

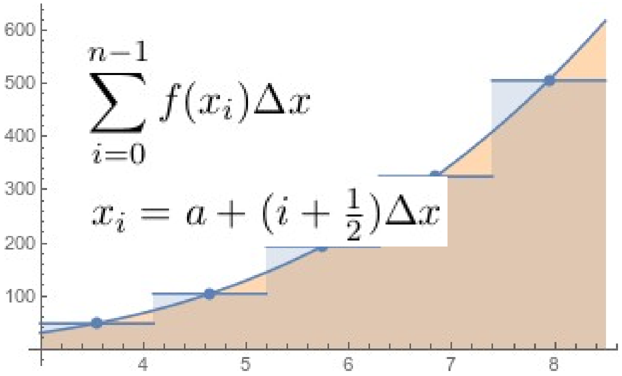
- Example 1: Left endpoint (rectangle)



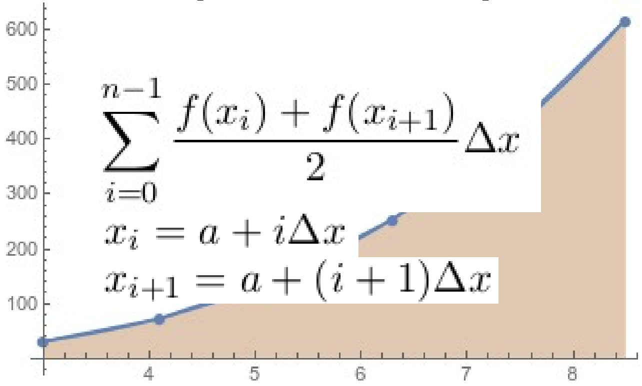
- Example 2: Right endpoint (rectangle)



- Example 3: Midpoint (rectangle)

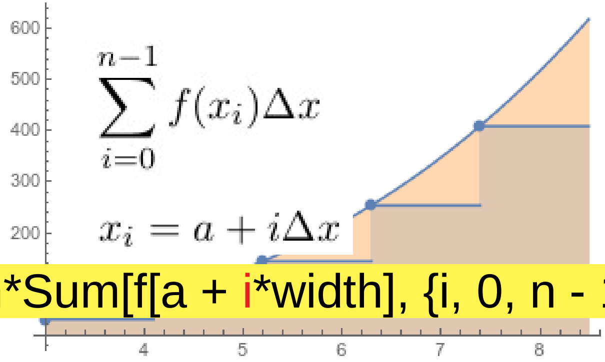


- Example 4: Trapezoid



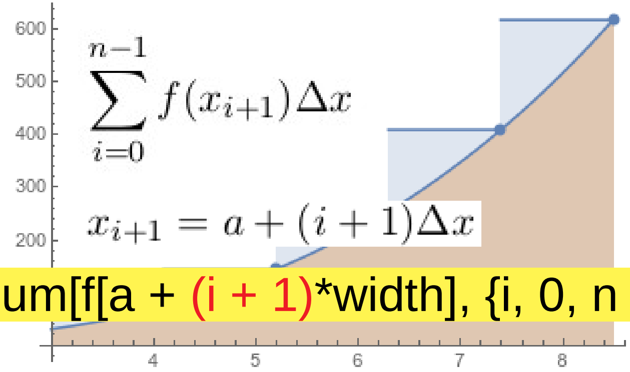
Numerical Integration

- Example 1: Left endpoint (rectangle)



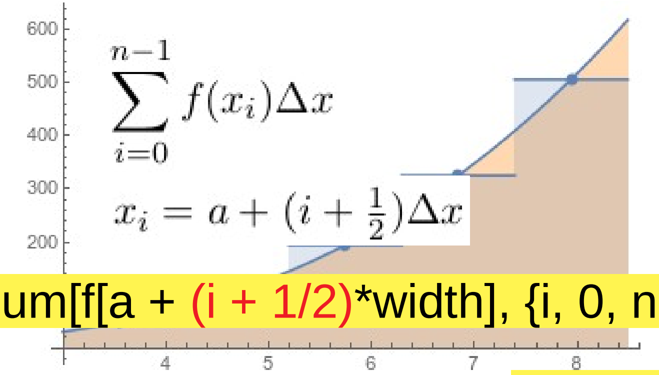
width*Sum[f[a + i*width], {i, 0, n - 1}]//N

- Example 2: Right endpoint (rectangle)



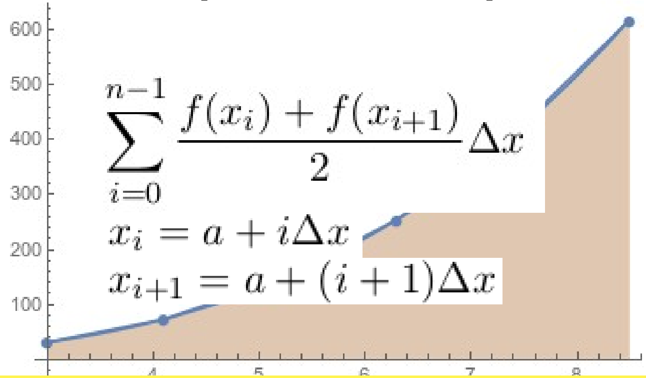
width*Sum[f[a + (i + 1)*width], {i, 0, n - 1}]//N

- Example 3: Midpoint (rectangle)



width*Sum[f[a + (i + 1/2)*width], {i, 0, n - 1}] // N

- Example 4: Trapezoid



width*Sum[(f[a + i*width] + f[a + (i + 1)*width])/2, {i, 0, n - 1}] // N

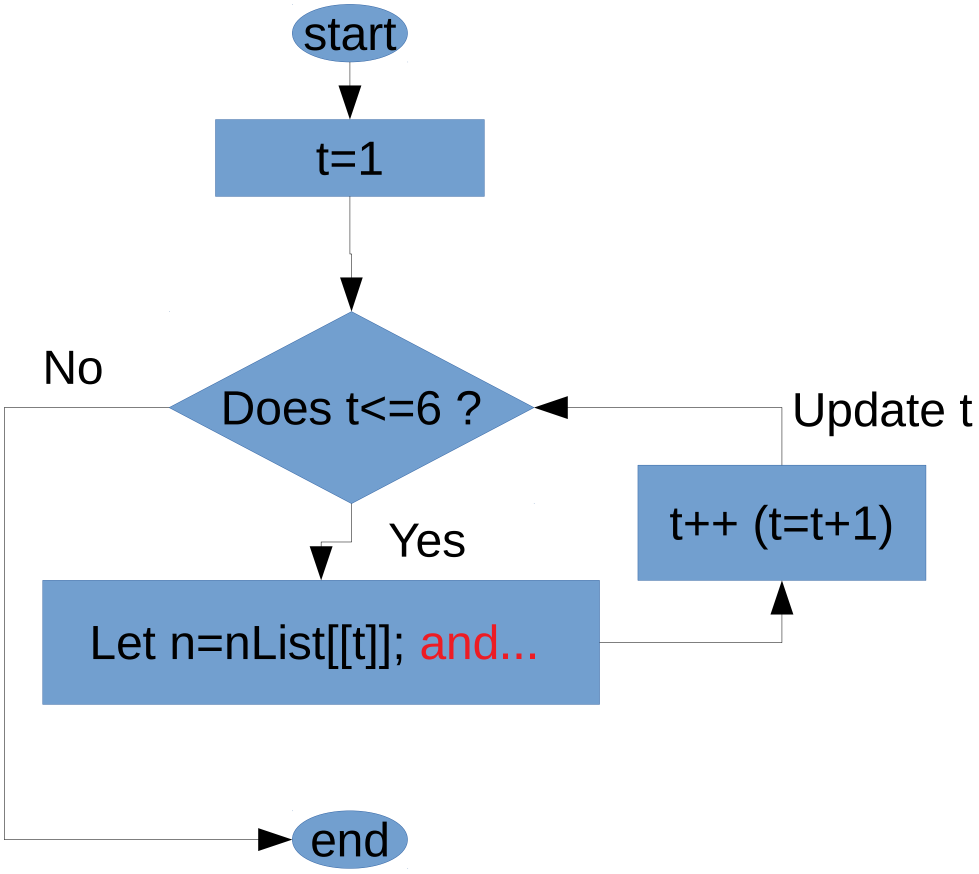
Question 2(a)---Set up

- Clear[a, b, width, f, nList];
- f[x_] := Exp[2 x]*Cos[2 Pi x];
- a = 0;
- b = 5;

Question 2(a)---For Loop

- **nList := {2, 5, 6, 9, 11, 15};**
- **For[**
- **t = 1,**
- **t <= 6,**
- **t++,**
- **{**
- **n = nList[[t]];**
- **(* Insert appropriate code for...*)**
- **}]**

t-th element of nList



Question 2(a)---”appropriate code”---What should we do in each step?

- Get a new “n” from “nList”

  n = nList[[t]];

- Compute a new “width”---because width is depends on “n”

  width=(b-a)/n;

- Using this “n” and “width”, compute a value with midpoint method

  approx = width*Sum[f[a + (i + 1/2)*width], {i, 0, n - 1}] // N;

- Output this estimation

  Print[approx];

Question 2(c)---“within 10 percent of the exact answer”

- **exact = Integrate[f[x], {x, 0, 5}] // N**
- “within 10 percent of the exact answer” means that the approximation is in the interval:

  [exact-exact*0.1, exact+exact*0.1]

  which is the same as saying

  exact*0.9 <= approx <= exact*1.1
- So start with n=1, we keep computing an approximation by midpoints (increase n in each step), and we stop whenever
- exact*0.9 <= approx <= exact*1.1.

  In other words, we keep computing as long as
- **approx<= exact*0.9 OR approx >= exact*1.1**

While Loop (example)

- **n = 1;**
- **While[n < 4,**

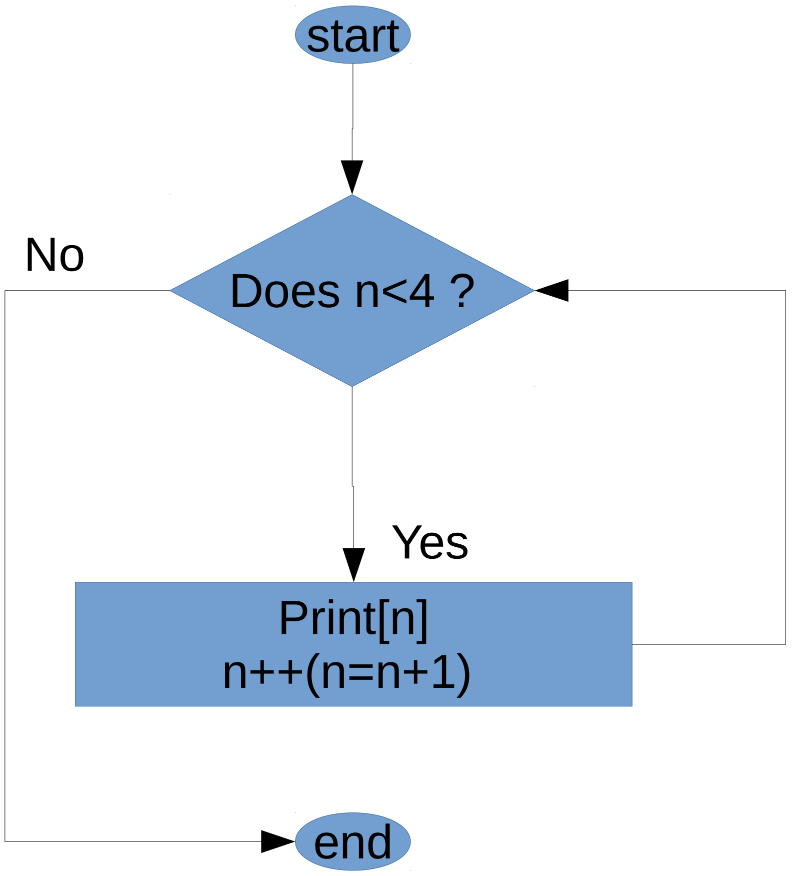
  Print[n]; n++]
- **Caution: If it takes more than**

  5~10 seconds to evaluate a while loop,

  use “Alt”+ “.” (windows) or

  “commend”+ “.” (Mac) to stop it.

  Otherwise it might crash!



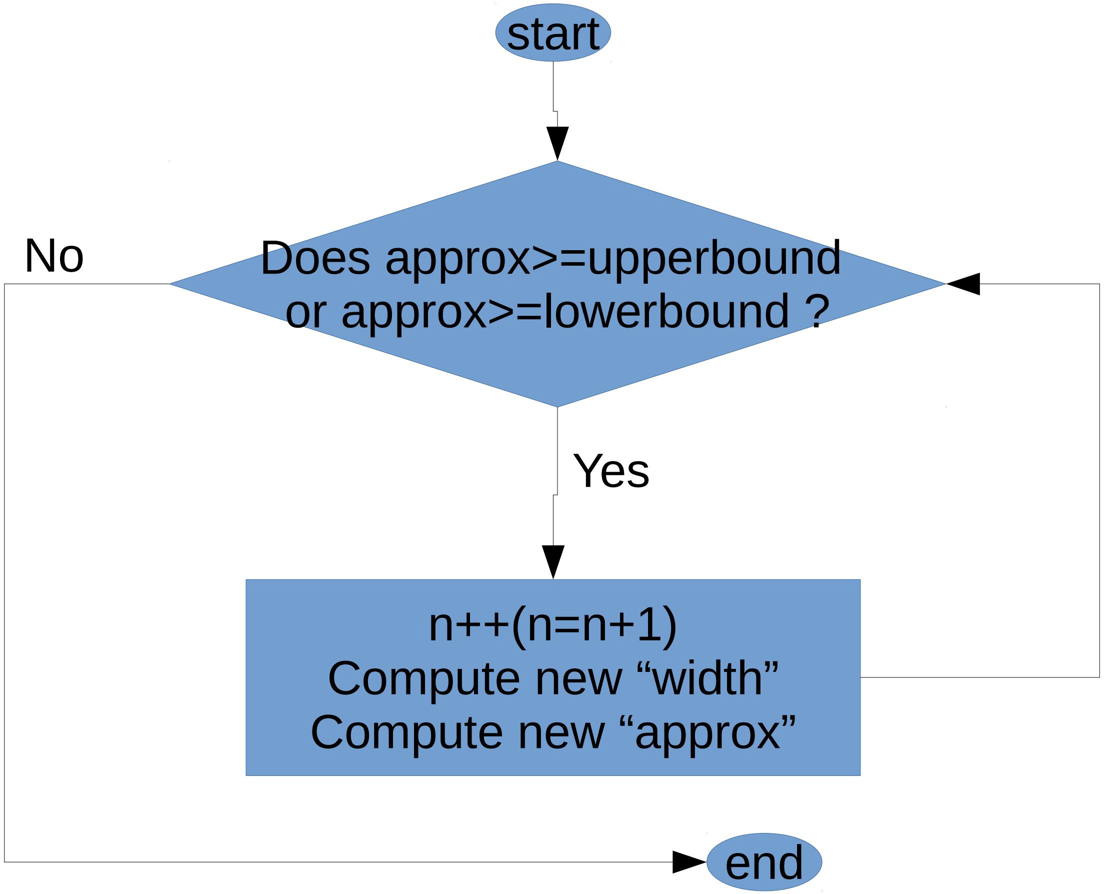
Questions 2(c)---While loop

- n = 1;
- approx=0;
- upperbound=exact*1.1;
- lowerbound=exact*0.9;
- While[approx >= upperbound || approx <= lowerbound,
- n++;
- width = (b - a)/n;
- approx = width*Sum[f[a + (i + 1/2)*width], {i, 0, n - 1}] // N;
-]
- Print[{n, approx}]

|| means OR

After the While loop stops,
print the current “n” and “approx”

Questions 2(c)---while loop



Wrong

- e^2x
- exp^2x
- Exp^[2x]
- e(2x)
- Cos[2Pix]
- Cos[2pi x]
- cos(2pix)

Correct

- E^(2x)
- Exp[2x]
- Note: “E” is the number e=2.71828…, and “Exp[x]” is the function e^x.
- Cos[2Pi x]
- Cos[2Pi*x]