16.216: ECE Application Programming

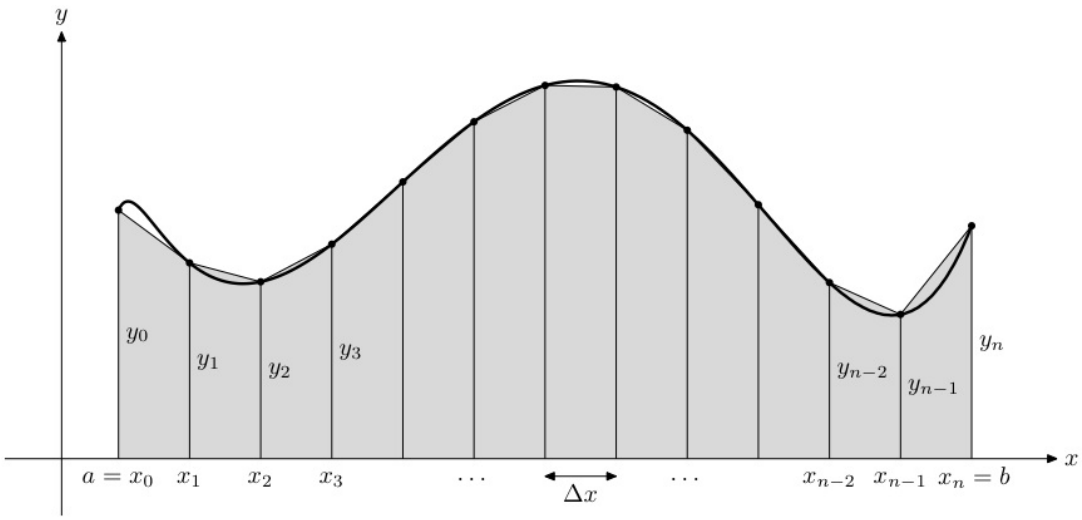
Fall 2015

Programming Assignment #5: Integral Approximation with Functions

Due **Monday, 10/19/15**, 11:59:59 PM

1. Introduction

In this program, you will design functions that allow you to approximate the integral of a function, $\mathrm{f}(x)$, using the trapezoidal rule. To find this integral—the area under the curve—we can approximate the area as a series of trapezoids, as shown below:



In the figure, the range $[\mathrm{a}, \mathrm{b}]$ has been divided into *n* different trapezoids, each of which has the same base, $\Delta x = (b - a)/n$. Recall that a trapezoid with base b and sides $h1$ and $h2$ has area:

$$0.5 \times b \times (h1 + h2)$$

Therefore, the area of trapezoid number *k* (1 ≤ *k* ≤ *n*) from the figure above is:

$$0.5 \times \Delta x \times (y_{k-1} + y_k) = 0.5 \times \Delta x \times (f(x_{k-1}) + f(x_k))$$

To find the total area under the curve—and therefore the approximate integral—sum the areas of all trapezoids:

$$\begin{aligned} Area &= 0.5 \times \Delta x \times (y_0 + y_1) + 0.5 \times \Delta x \times (y_1 + y_2) + \cdots + 0.5 \times \Delta x \times (y_{n-1} + y_n) \\ &= 0.5 \times \Delta x \times (y_0 + y_1 + y_1 + y_2 + \cdots + y_{n-1} + y_n) \\ &= 0.5 \times \Delta x \times (y_0 + 2y_1 + 2y_2 + \cdots + 2y_{n-1} + y_n) \\ &= \mathbf{0.5} \times \Delta \boldsymbol{x} \times \left(\boldsymbol{y_0} + \mathbf{2} \sum_{\boldsymbol{k=1}}^{\boldsymbol{n-1}} \boldsymbol{y_k} + \boldsymbol{y_n} \right) \approx \int_a^b f(x)dx \end{aligned}$$

Your integral function will use the equation shown in bold above to approximate the integral, given the endpoints of the interval $[\mathrm{a}, \mathrm{b}]$ and the number of trapezoids, n.

2. Deliverables

This assignment uses multiple files; starter versions of each file are provided on the course web page:

- ***prog5_integral.c:*** Source file containing your main function.
- ***prog5_functions.h:*** Header file containing function prototypes. **Do not change the contents of this file**.
- ***prog5_functions.c:*** Source file containing other user-defined functions

Submit all three files by uploading these files to your Dropbox folder. Ensure your file names match the names specified above. Failure to meet this specification will reduce your grade, as described in the program grading guidelines.

3. Specifications

Program structure: Your program should follow the general outline below:

1. Prompt for and read the following values:

- The low and high points of the interval $[a, b]$, over which $f(x)$ is to be integrated.
- The number of trapezoids, n, to be used in that integration.

If an input error occurs, print an error message and repeat the prompt for that input. Input errors are as follows:

- `scanf()` cannot read the input values (for example, if the user types "`A 3`" for the interval endpoints).
  - In this case, you must clear the rest of the line before retrying `scanf()`.
  - See Lecture 4: PE2 for a reminder of how to check that the input is properly formatted and how to clear the rest of the line if it is not.
- The "low" interval endpoint is greater than or equal to the "high" endpoint.
- The number of trapezoids is less than 1.

2. Once the user has entered error-free input values, call the `integrate()` function (described below), which will use the trapezoidal method to approximate the integral of $f(x)$ over the interval $[a, b]$ using n trapezoids.

3. After printing the results, ask the user if he or she wants to repeat the program, and then read a single character that serves as the response to that question. If the user enters:

- `'Y'` or `'y'` → Return to Step 1.
- `'N'` or `'n'` → End the program.
- Any other character → Print an error message and repeat the question.

See Section 4 for test cases that demonstrate the proper format for input and output.

Functions: Your program should contain functions with the prototypes shown below:

- `double f(double x);`

  The function being integrated, which should calculate the value:

$$f(x) = \sin(x) + \frac{x^2}{10}$$

  Note that you should include the math library `<math.h>` in order to use the sin() function; this function takes a single argument and returns the sine of that value.

- `double integrate(double a, double b, int n);`

  This function should use the trapezoidal method to approximate the integral of $f(x)$ over the interval $[a, b]$ using n trapezoids, as described above. The return value of the function is the result of the approximation. Note that the function should not print any values to the screen—the output should be handled in the main function.

- `void badInput();`

  Clear the current line of input. This function should be used after input formatting errors occur.

4. Test Cases

Your output should match these test cases exactly for the given input values. I will use these test cases in grading of your lab, but will also generate additional cases that will not be publicly available. Note that these test cases do not cover all possible program outcomes. You should create your own tests to help debug your code and ensure proper operation for all possible inputs.

```
Enter endpoints of interval to be integrated (low hi): -2.1 1.5
Enter number of trapezoids to be used: 15
Using 15 trapezoids, integral between -2.100000 and 1.500000 is -0.148162
Evaluate another interval (Y/N)? Y

Enter endpoints of interval to be integrated (low hi): -2.1 1.5
Enter number of trapezoids to be used: 45
Using 45 trapezoids, integral between -2.100000 and 1.500000 is -0.153692
Evaluate another interval (Y/N)? Y

Enter endpoints of interval to be integrated (low hi): A 3
Error: Improperly formatted input

Enter endpoints of interval to be integrated (low hi): 5 4
Error: low must be < hi

Enter endpoints of interval to be integrated (low hi): 0 0.1
Enter number of trapezoids to be used: Q
Error: Improperly formatted input
Enter number of trapezoids to be used: 0
Error: numT must be >= 1
Enter number of trapezoids to be used: 10
Using 10 trapezoids, integral between 0.000000 and 0.100000 is 0.005029
Evaluate another interval (Y/N)? Q
Error: must enter Y or N
Evaluate another interval (Y/N)? N
```