

## Project 4

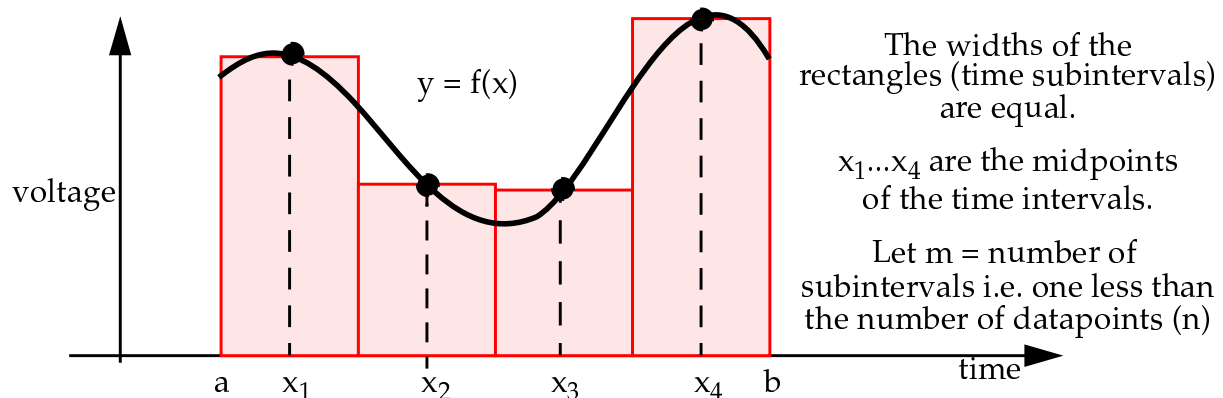
Assigned: Friday, Oct. 13

Due: Monday, Oct. 30

## Project Description:

Write an 80x86 assembly program that performs the following functions:

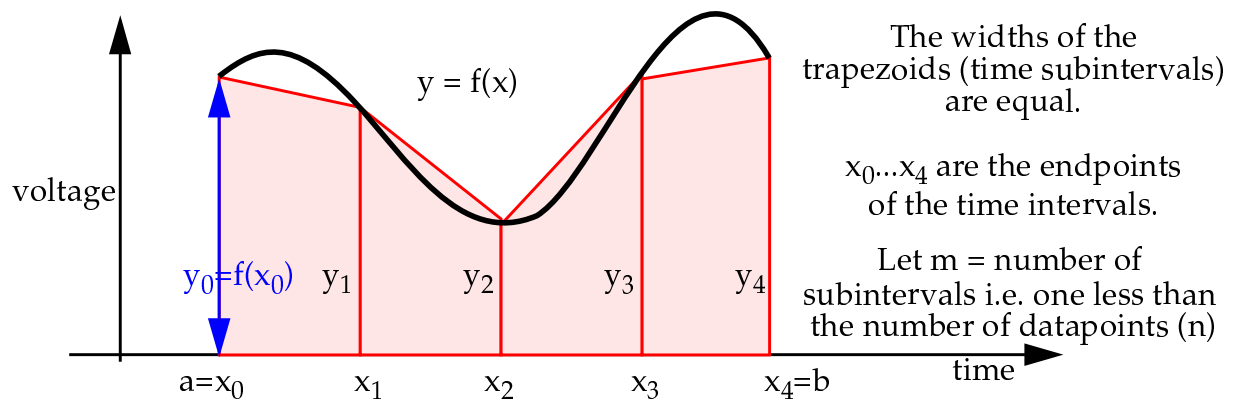
- Reads in a set of waveforms into a set of arrays (or a multidimensional array) from a file. The name of the data file is given as a command line parameter. These waveforms represent variations in voltages as a function of time. You can assume the data file and your executable are in the same directory. Also assume that each waveform can have a maximum of 1000 data points and a file can have maximum 100 waveforms.
- Compute the area under the waveforms using several integration methods outlined below.
  - a) Rectangular Rule:**



$$\int_a^b f(x) dx \approx \left[ \frac{(b-a)}{m} \right] (f(x_1) + f(x_2) + f(x_3) + \dots + f(x_m))$$

For your project,  $a$  = first  $x$  value,  $b$  = last  $x$  value, if  $n$  = # of datapoints,  $m = n - 1$ .

## b) Trapezoidal Rule:



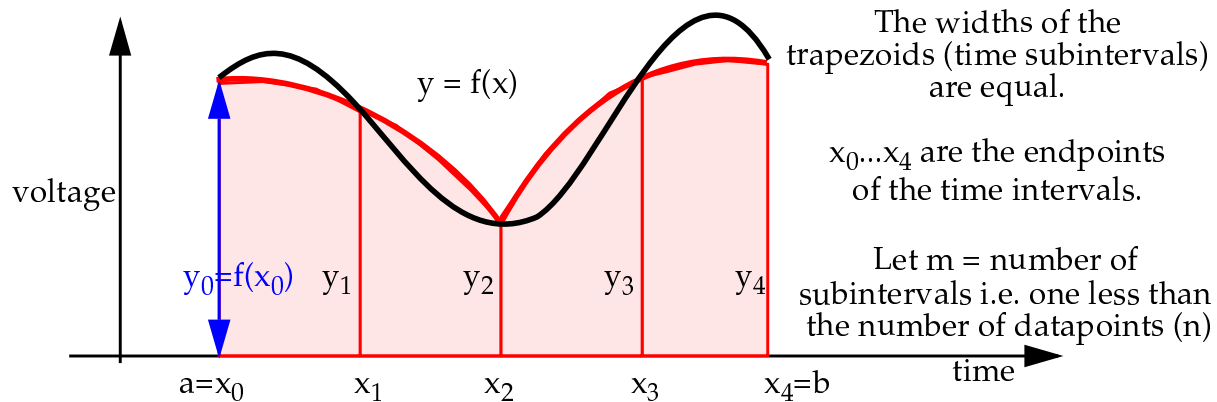
$$\int_a^b f(x) dx \approx \left[ \frac{(b-a)}{2m} \right] (y_0 + 2y_1 + 2y_2 + \dots + 2y_{n-1} + y_n)$$

**c) Simpson's (Parabolic) Rule:**

In the rectangular rule, the approximating rectangles have horizontal lines (constant functions  $y = a$ ) as tops, and the lines meet the graph at one point. In the trapezoidal rule, the top of a trapezoid may be any line (line function  $y = ax + b$ ), and the lines meet the graph at two points. It is also possible to put a quadratic (parabola) function  $y = ax^2 + bx + c$  through three points of the graph.

We approximate the graph of  $f$  over the first two subintervals by a parabola through  $(x_0, y_0)$ ,  $(x_1, y_1)$  and  $(x_2, y_2)$ , over the next two subintervals by a parabola through  $(x_2, y_2)$ ,  $(x_3, y_3)$  and  $(x_4, y_4)$  and so on. The area is then estimated by taking the areas under the parabolas.

When  $x_1 - x_0 = x_2 - x_1$ , (equal spaced  $x$  subintervals) a convenient formula exists for the area under the parabola using the heights  $y_0$ ,  $y_1$ , and  $y_2$ .



$$\int_{x_0}^{x_2} f(ax^2 + bx + c)dx \approx \left[ \frac{(b-a)}{3m} \right] (y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + \dots + 2y_{n-2} + 4y_{n-1} + y_n)$$

**OUTPUT:** It will include the areas under each waveform for each technique. A sample input and sample output file is given on the class webpage. Format your output file as shown in the example file. You can use C functions only to open and print to the output file and to print on the screen.

**The format of the waveform input file is:**

```
w1_n1
w1_x0 w1_y0
w1_x1 w1_y1
w1_x2 w1_y2
...
w1_x_{n-1} w1_y_{n-1}
```

```
w2_n1
w2_x0 w2_y0
w2_x1 w2_y1
w2_x2 w2_y2
...
w2_x_{n-1} w2_y_{n-1}
```

...

The prefix 'w1' indicates the data for waveform 1, 'w2' indicates the data for waveform 2, and so on. The first number in each 'wx' group is the number of (x, y) points following it. The data points are given one point per line with the x value on the left and the y value on the right. You can assume the x data (time subintervals) are equally spaced.

You must use the submit program to submit your project. The class name is cmpe310 and the project name is proj4. Submit the project (project4.asm) file and any code that you use from our examples should be in (common\_code.asm). You are also required to turn in a hardcopy and a write-up as described in project 1. Break-up of points is similar to project 1. Make the code modular i.e. write subroutines to perform common functions. A portion of your grade will depend on the code modularity.

Example input and output files are given on the webpage. You can construct your own data files in the format described (make sure you have odd number of data points in each waveform). We will test your code on our own files. The submitted program is due before you come to class on Monday. You must turn in a hardcopy during class on Monday and it must be identical to the code that you submitted.

**THE LABS ARE INDIVIDUAL EFFORTS: INSTANCES OF CHEATING WILL RESULT  
IN YOU FAILING THE COURSE.**