

## CSE 232, Lab Exercise 3

### More Command Line

#### Pagers

Often, you want to be able to view files in the terminal (instead of opening them in a text editor like Atom or gedit). There are a few ways view files from the terminal.

##### **cat**

The command `cat` is short for concatenate, which means to link things together.

The `cat` program is often used to link the output from one program to the input from another. But another useful trait of `cat` is to output the contents of a file to the terminal. Invoking the `cat` program is quite simple:

```
cat some_file_name.txt
```

The above command will print the contents of the file to the terminal. However, for large files, `cat` is not very user-friendly. Instead, you should use a pager.

##### **less**

Pagers are programs that show you content one **page** at a time. They are useful for viewing large files (files too big to fit in the terminal window). The most popular pager is called `less` because it was derived from an older program called `more`, computer science history is a little silly. You can invoke the `less` program like so:

```
less some_file_name.txt
```

To go the the next page, push the `f` key (forward). To go back a page, push the `b` key (backwards). To quit the pager and return to the command line, push `q` (use your imagination for why it is the `q` key).

#### Help

There are a few programs that are intended to provide documentation on the programs install on your computer. However, they are often difficult to understand, especially for a beginner. You invoke them by typing the help program and the name of the program you want information about. They may open up a pager if the entry is long.

##### **help**

This command is used to provide information about `tcsh`, which is the language the terminal in x2go is running (Mimir uses another called `bash` )

##### **man**

This command (short for manual) is used to retrieve the documentation about the programs installed on the computer. They can be long (try `man g++` for example. It comes automatically with the pager keys described above).

You can search through large man files looking for a keyword. When the pager stops at the bottom of the page you can type a `/keyword`. For example, if you type `man g++` (a big page), then you type `/memory` while the pager is active (command will show up in the bottom of the window) it will page to the next example of the word `memory` in the file. If you then hit the `n` key, it will go to the next example of `memory` in the file. You can keep hitting `n` looking for the next example of a keyword. The `q` quits as before. If you want to see more commands, hit the `h` key while the pager is active and it will give you help in the pager.

### **info**

This command is an alternate to `man`, often used by unix distributions.

Examples:

```
help cd
info ls
man python3
```

### **-h or --help**

Many programs will give you a bit of documentation about themselves (like what arguments they accept) if you invoke the program with the `-h` or `--help` flags. Example:

```
python -h
g++ --help
```

## **Assignment**

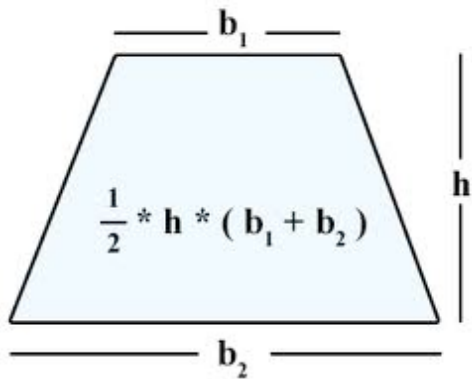
### **Background**

You remember calculus don't you? The basic concept of an integral is the area under the curve, the curve represented by some function. If you can integrate a function, you can calculate that area directly, but for some functions it is easier to approximate that area using discrete, iterative methods. We are going to investigate one of those methods, the Trapezoid Rule ([http://en.wikipedia.org/wiki/Trapezoidal\\_rule](http://en.wikipedia.org/wiki/Trapezoidal_rule)) .

### **Trapezoid Rule, area of a Trapezoid**

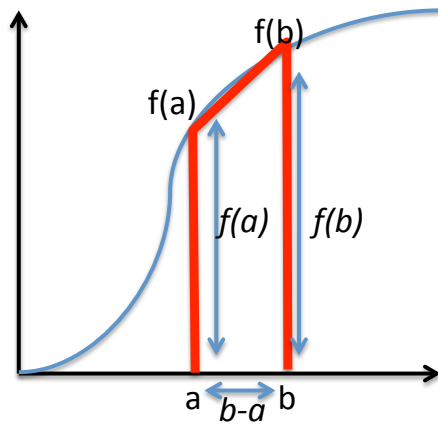
The basic idea is to draw a series of trapezoids that approximate the area under a curve, where the more trapezoids we draw, the better the approximation.

First, remember how to calculate the area of a trapezoid?



One measures the length of the parallel sides ( $b_1$  and  $b_2$ ), then the distance between the parallel sides ( $h$ ). Add the parallel distances, multiply those values by  $h$ , divide by 2.

Let's rotate that trapezoid  $90^\circ$  and model it as below. Now the  $b_1$  length is the curve value  $f(a)$ , the  $b_2$  length  $f(b)$  and the height  $h$  the difference between  $b - a$ .



Given the above, the area would be:

$$\frac{(b-a) * [f(a) + f(b)]}{2}$$

For two trapezoids next to each other with heights  $f(a)$ ,  $f(b)$  and  $f(c)$ ,  $f(b)$  a common side used by the two trapezoids, we get the formula:

$$\frac{(b-a)[f(a) + f(b)]}{2} + \frac{(c-b)[f(c) + f(b)]}{2}$$

In general, for the definite integral:

1. from  $a$  to  $b$
2. with  $n$  equally space trapezoids ( $n+1$  points on the x axis)
3. grid spacing of  $h=(b-a/n)$

$$\int_a^b f(x) dx \approx \frac{h}{2} \sum_{k=1}^N (f(x_{k+1}) + f(x_k))$$

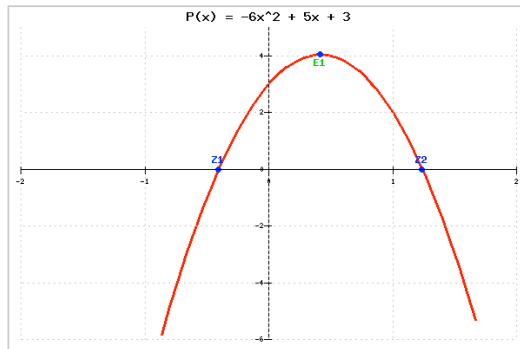
$$= \frac{b-a}{2N} (f(x_1) + 2f(x_2) + 2f(x_3) + \dots + 2f(x_N) + f(x_{N+1})).$$

### Program Specifications

1. Write three functions:

- a. a function `fn` to integrate over. Let's use  $fn \rightarrow -6x^2 + 5x + 3$ 
  - i. `fn` takes a single double parameter, the value `x`
  - ii. `fn` returns a double, the result of the function evaluation
- b. a function `integral`, the actual integral of `fn` which is  $integral \rightarrow -2x^3 + \frac{5}{2}x^2 + 3x$ 
  - i. `integral` takes a single double parameter, `x`
  - ii. `integral` returns a double, the result of the function

let's further fix the interval to calculate the integral over as 0 to 1

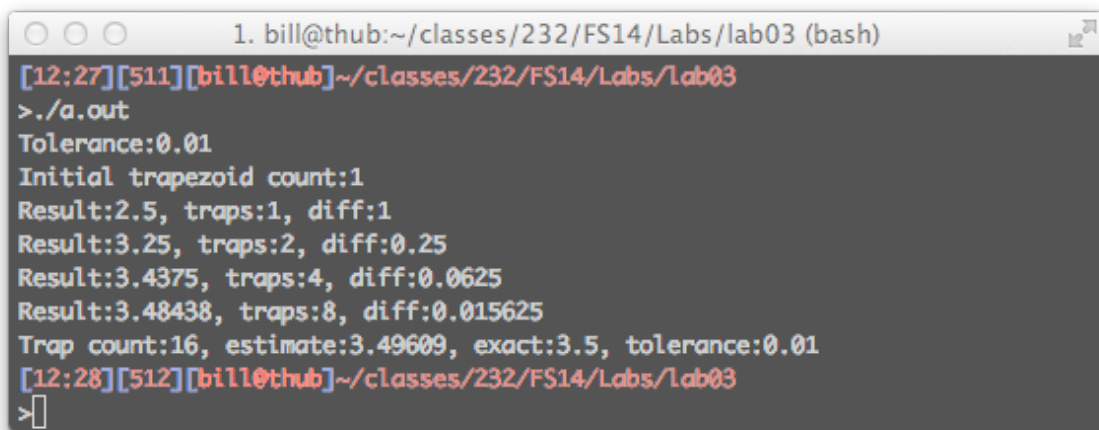


- c. a function `trapezoid`. The specification for `trapezoid` is:
  - i. take in three parameters:
    1. the two definite points of the integral `a` and `b` as doubles
    2. the number `n`, the number of trapezoids, a long
  - ii. calculates the area under the curve represented by the function `fn` given the provided number of trapezoids over the interval 0 to 1.
  - iii. returns a double, the sum of the area of the trapezoids.
2. A main function that does the following:
  - a. takes in two values (in order)
    - i. a float value tolerance
    - ii. an initial guess at the number `n`, the number of trapezoids
  - b. you run a loop that measures the difference between the actual value of the integration (you have the `integral` function so you can calculate the exact value between `a` and `b`) and the estimated value from `trapezoid`
    - i. if the difference is within tolerance, report to the user 4, space separated values on a single line. All output should be fixed, `setprecision(6)`
      1. `n`
      2. estimate value

3. exact value
    4. tolerance
  - ii. if the difference is not within tolerance, **double** the value of  $n$  and rerun. Continue the doubling and re-running until the estimate of the trapezoid function it is within tolerance of the actual value from integral.
3. You can test this on Mimir, Lab03:calculus

## Example Run

As I've said before, you might want to put some extra output in to check yourself before you try the actual Mimir tests. This is what I did.



```
1. bill@thub:~/classes/232/FS14/Labs/lab03 (bash)
[12:27][511][bill@thub]~/classes/232/FS14/Labs/lab03
> ./a.out
Tolerance:0.01
Initial trapezoid count:1
Result:2.5, traps:1, diff:1
Result:3.25, traps:2, diff:0.25
Result:3.4375, traps:4, diff:0.0625
Result:3.48438, traps:8, diff:0.015625
Trap count:16, estimate:3.49609, exact:3.5, tolerance:0.01
[12:28][512][bill@thub]~/classes/232/FS14/Labs/lab03
>
```