

## Computational Project 4

Due: Friday March 18<sup>th</sup> at noon (MT)

### 1. Integration

The overall goal of this problem is to find the area of the bat sign using numerical integration. The approach for finding the area of the bat sign will be to integrate two functions,  $b_1(x)$  and  $b_2(x)$ , which evaluate the outline of the top and bottom halves of the sign. The total area is thus

$$I = \int_{-7}^7 b_1 dx - \int_{-7}^7 b_2 dx$$

Code for calculating  $b_1$  and  $b_2$  are provided on D2L (bat\_func\_1 and bat\_func\_2, respectively) where the input for each function is  $x$  and the output is  $f(x)$ . Note that the output will be symbolic. Try the following commands in the command window:

```
>> bat_func_1(1)
```

```
>> double(bat_func_1(1)) %casts output as double—use to show numeric values
```

Treat these codes like “black boxes”, i.e., you can evaluate them for any  $x$ , but pretend you do not know the analytical expressions contained in  $b_1$  and  $b_2$ . However, you may note the following: the symbol is symmetric about  $x = 0$ , the  $x$  coordinate of the discontinuities in  $b_1(x)$  are located at  $x = \pm 0.5, \pm 0.75, \pm 1.0$  and  $\pm 3.0$ , and the  $x$  coordinate of the discontinuities in  $b_2(x)$  are located at  $x = 0, \pm 2$  and  $\pm 4$ .

Find the area of the bat sign accurate to within an absolute error of 0.005 for each integral. Report the number of times you needed to evaluate  $b_1(x)$  and  $b_2(x)$  to find your final integral.

Your code should try more and more points (smaller spacing) until the desired error tolerance is reached. At this error, it should stop automatically and output your answer. For each attempt, display the number of points, the integral estimate, and the estimate of the error.

Part of your grade for this project will be for efficiency. It is easy to write a very simple integration scheme that uses many points with the rectangle method, but it will be quite slow. Part of your grade will also be for program style, so think about how you can write a program that uses separate functions in a clean and clear manner.

## 2. Derivative of right upper edge

Write a code that makes a plot of the derivative of  $b_1(x)$  over the range  $3 \leq x \leq 7$ , calculated such that the average error in the derivative across all points of the plot is less than 0.01. Your code should use finite difference methods to find the derivative, adapting the step size such that this error tolerance is met. Display the step size and estimated error each time the derivative is estimated. Similar to problem 1, the efficiency and style of your program will factor into the overall grade.

Hint: to determine the error in an estimate of the derivative, you may need to compare to a previous estimate with a smaller interval size,  $h$ . This can be a bit tricky, since changing  $h$  changes the number of points for which you calculate the derivative. So be careful with the alignment when making this comparison. You could use the following to compare the previous derivative,  $D_{old}$ , calculated using  $h = h_1$  to  $D_{new}$ , calculated using  $h = h_1/2$  with:

`difference = Dnew(1:2:end) - Dold`

Remember that you should submit only one m file and your code should be commented.