Experiment-4
Trapezoidal Rule

(Duration: 120 mins)

Author: Şamil Şirin
samilsirin@iyte.edu.tr

08.06.2020

Purpose: Integrating a bell curve using Trapezoidal Rule with adaptive method.

Introduction

Gaussian is a mathematical function that is generally known among students as 'bell curve' since it is frequently used for grading. It has also very significant role in Probability Theory as a probability density function. Its mathematical expression given in Equation (1) involves 2 parameters, mean μ and standard deviation σ (sigma) which respectively determine the position of the *curve center* and *how the curve is spread out* as seen in the Figure (1).

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{1}$$

Integral of this function is also used very often. However, it cannot be integrated analytically (you can try if you want), so numerical methods step in at this point.

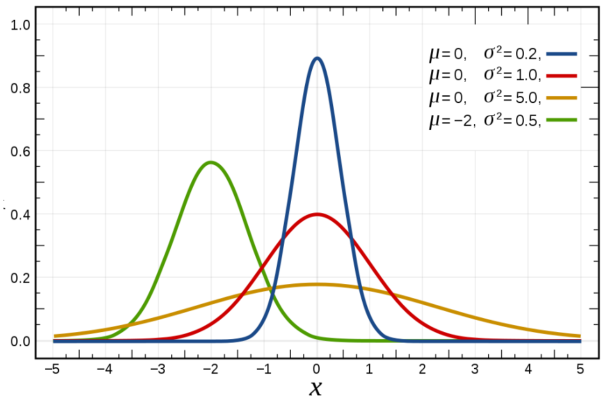

Figure 1: Different Gaussian curves with different parameters*
* Retrieved from: https://en.wikipedia.org/wiki/Normal_distribution

Trapezoidal Rule

Trapezoidal Rule is one of the numerical techniques to approximate a definite integral. In this method, region under the graph is calculated by regarding it as a combination of many trapezoid sections. As number of sections (or panels) increases, approximation error decreases. Region under the function $f(x)$ between a and b (assuming no singularity exists in that range) approximated by only one trapezoid can be found by calculating the area of the trapezoid i.e. $\frac{f(a)+f(b)}{2}(b-a)$. Therefore, we can generalize this to sum of n consecutive trapezoid areas such that

$$\int_a^b f(x) \approx \frac{b-a}{n}\left[\frac{f(a)+f(b)}{2} + \sum_{i=1}^{n-1} f\left(a + i\frac{b-a}{n}\right)\right] \tag{2}$$

In addition to that, error of this approximation would be calculated as in the Equation (3).

$$error = -\frac{(b-a)^3}{12n^2} f''(0) \tag{3}$$

Also note that, error can be reduced below a tolerance value by selecting proper number of panels, n.

Problem Statement

In this lab, you are asked to write a code that approximate the integral of simplified version of bell curve in the Equation (4) for 4 different σ values: $1, 2, 4, 10$ in the range $[0, 30]$. Its graph is also given in Figure (2).

$$f(x) = \frac{1}{\sigma} e^{-\frac{x^2}{\sigma^2}} \tag{4}$$

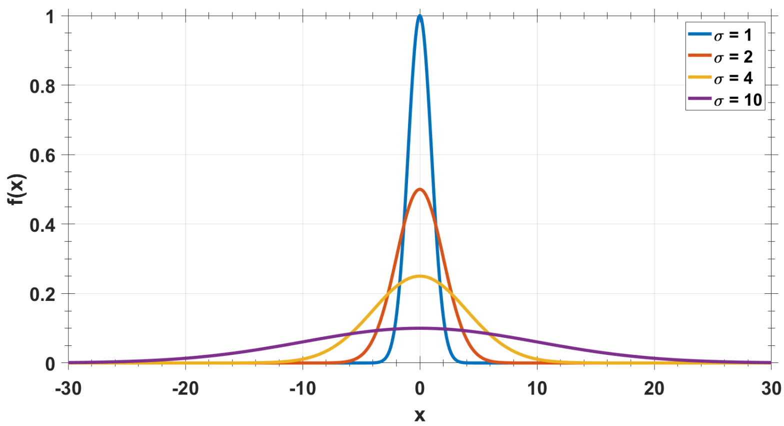

Figure 2: Graph of simplified curve bell in Equation (4) for 4 different σ values

Your approximation should have an error less than the indicated tolerance. Therefore, your code is expected to decide proper number of panels via adaptive method. In adaptive method, you may start applying Trapezoidal Rule with only one panel and then double the number of panels and check if the resulting error is less than the tolerance.

Lab Procedure

1- Implement simplified bell curve given in Equation (4) as a function with 2 arguments, x and σ. Call the function in main with following parameters to test: (*10 pts*)

```
double x_start_test = 0;
double sigma_test = 4;
```

2- Implement Trapezoidal Rule as a function. The function must take starting point (x_{start}), ending point (x_{end}), curve parameter (σ) and number of panels (n) as arguments. Using function pointer is optional (If you will use, include it to arguments). Call the function in main with the following parameters in addition to the ones in the previous step: (*20 pts*)

```
double num_panel_test = 10;
double x_end_test = 20;
```

3- Write a function that calculates error. To do that, first analytically take the second order derivative of the function in Equation (4) with respect to x by hand and substituting $x = 0$, obtain the 2^{nd} order derivative at $x = 0$ in terms of σ. Then, implement Equation (3) as a function with appropriate arguments. (*10 pts*).

4- Write a function in which the necessary number of panels is automatically decided via adaptive method. Therefore, you need to take a tolerance value as an argument in addition to the arguments of the function written in step 2. In this function, you need to calculate the error, using the function written in step 3 and compare it to the tolerance value. As long as the error is greater than the tolerance value, the function should keep calculating a new result with doubled number of panels. At the end, it should return to the most accurate approximation. Besides, **number of panels** with which the returned result is calculated also should be **accessible from outside the function.** (*Static* or *global* declarations are not allowed.) (*30 pts.*)

5- In main function, ask the user for σ and tolerance values. After taking one value per each, find and print the result of approximation in the domain $x \in [0, 30]$ by calling the function you wrote in step 4. Note that, you do not have to specify number of panels as it will be decided automatically. You can initialize the number of panels from 1. Remember also that the domain range is $[0, 30]$. You should print the result by specifying the σ value and number of panels used as seen in the Figure (3). (*20 pts.*)

6- Take the procedure of asking for values and printing the result in step 5 into a loop so that the result could be computed multiple times for different parameters in one execution. (*5 pts.*)

7- Write a break condition to the loop written in the step 6 so that program is ended by entering 0 for σ value. (*5 pts.*)

- You can see some sample outputs of the program in the Figure 3.

```
Test for bell function: 0.250000
Test for trapezoidal function: 0.886227
Enter sigma value:1
Enter the tolerance value: 0.1
For sigma value: 1.000000, required number of panel is 256 and integral result is 0.886227
Enter sigma value:2
Enter the tolerance value: 0.1
For sigma value: 2.000000, required number of panel is 128 and integral result is 0.886227
Enter sigma value:4
Enter the tolerance value: 0.1
For sigma value: 4.000000, required number of panel is 32 and integral result is 0.886227
Enter sigma value:10
Enter the tolerance value: 0.1
For sigma value: 10.000000, required number of panel is 8 and integral result is 0.886199
Enter sigma value:0

--------------------------------
Process exited after 13.45 seconds with return value 0
Press any key to continue . . .
```

Figure 3: Expected output of the program

- Submission address: eelab204@gmail.com