Lab 1: Using Computers to Approximate Integrals

Online editor and compiler: https://replit.com/languages/python3

Code snippet 1:

sum = 0

n=2

dx=2/n

correct = 2\*\*3 - 0\*\*3

for j in range(n):

 x\_left=0+j\*dx

 ht\_left = 3\*x\_left\*\*2

 area\_LHS = ht\_left\*dx

 sum = sum + area\_LHS

 print("iteration ",j,"sum ",sum)

error = correct - sum

print("sum",sum,"actual value",correct,"error",error)

Optional Advanced Code snippet 2:

correct = 2\*\*3-0\*\*3

for i in range(3):

 sum = 0

 n=10\*\*(i+1)

 dx=2/n

 for j in range(n):

 x\_left=0+j\*dx

 ht = 3\*x\_left\*\*2

 area = ht\*dx

 sum = sum + area

 error = correct - sum

 print("number of divisions",n,"error",error)

Problem 1: A function *f* is represented with the formula *f(x) = 3x2..* We are approximating area under the curve from x = 0 to x = 2 using LHS, RHS and midpoint.

1. Use anti derivatives to find the precise value of the area.
2. Fill in the following table using 2 significant digits:

|  |  |  |  |
| --- | --- | --- | --- |
| n | Error in approximating with the LHS rule | Error in approximating with RHS | Error in approximating with midpoint |
| 10 |  |  |  |
| 100 |  |  |  |
| 1000 |  |  |  |

1. When *n* is multiplied by 10, what happens to the size of the error in
	1. LHS
	2. RHS
	3. Midpoint
2. Which method provides the best approximation and how can we quantify how much better it is?

Problem 2: A function *f* is represented with the formula *f(x) = x3..* We are approximating area under the curve from x = 0 to x = 4 using LHS, RHS and midpoint.

1. Use anti derivatives to find the precise value of the area.
2. Fill in the following table using 2 significant digits:

|  |  |  |  |
| --- | --- | --- | --- |
| n | Error in approximating with the LHS rule | Error in approximating with RHS | Error in approximating with midpoint |
| 10 |  |  |  |
| 100 |  |  |  |
| 1000 |  |  |  |

1. When *n* is multiplied by 10, what happens to the size of the error in
	1. LHS
	2. RHS
	3. Midpoint
2. Which method provides the best approximation and how can we quantify how much better it is?

Optional Problem 3: When the trapezoid rule is used, each division approximates the area under the curve in that division with a rectangle with a base of $∆x$ and height equal to the average of the height on the LHS and the height on the RHS. Using the function$x)=x^{2}$ from x = 0 to x =5:

1. Adapt your python code to fill in the table in the following table

|  |  |
| --- | --- |
| n = the number of divisions | Error in approximating with the Trapezoid rule |
| 10 |  |
| 100 |  |
| 1000 |  |

1. When we multiply the number of divisions by 10, what happens to the error?