Report for Assignment 2

Siddharth Nayak-EE16B073

23-01-2018

1 Abstract

This assignment focuses on the concepts of integrating functions using python functions and by using the trapezoidal method. Also the error estimation method for the trapezoidal method of finding area is given emphasis.

2 Definition of Function

The first question is to define the function f(x)=$\frac{1}{1+x^2}$. It can be achieved by the following function:

```
import numpy as np

def function(t):
    return(1./(1+np.multiply(t,t)))
```

3 Definition of Vector

The second question asks us to get a vector x in the region 0≤x≤5 in steps of 0.1. This can be achieved by the following snippet of code:

```
vector_x = np.linspace(0,5,51)
```

4 Plot the Function

In the next part the function
f(x)=$\frac{1}{1+x^2}$ is plotted vs x.For this the
`matplotlib.pyplot` library is used.

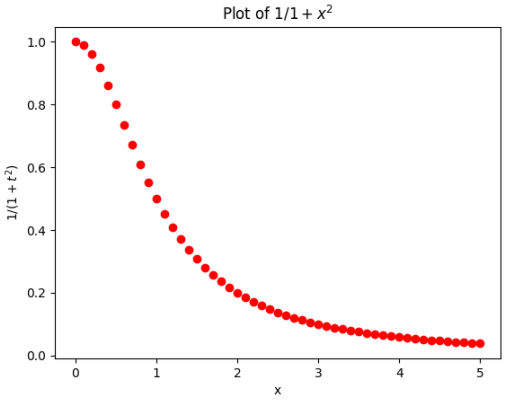


Figure 1: Plot of f(x)=$\frac{1}{1+x^2}$

5 Integrating the function and plotting error

This question has three subparts in it:
1:Integrating the function from x=0 to x=5
2:Plotting the integral along with arctan(x)
3:Plotting the error in the integral on a semilog axis.

5.1 Part 1:

For integrating the function we use
`from scipy.integrate import quad`.The `quad` function takes in two arrays:
1:The function
2:The lower limit
3: The upper limit
The quad function integrates the values and returns the integral and the error in calculation. Here we take lower limit=0 and upper limit=5.

```
# integration code
lower_limit=0
upper=0.001
upper_limit=5
dx=0.01
integrated=[]
```

```
x_quad=[]#contains x for Integration
error=[]
while upper<upper_limit:
  x_quad.append(upper)
  y,err=quad(function,lower_limit,upper)
  integrated.append(y)
  error.append(err)
  upper+=dx
```

5.2 Part 2:

The integrated function or the integral is plotted on red dots and the original function arctan(x) is plotted in a solid black line.

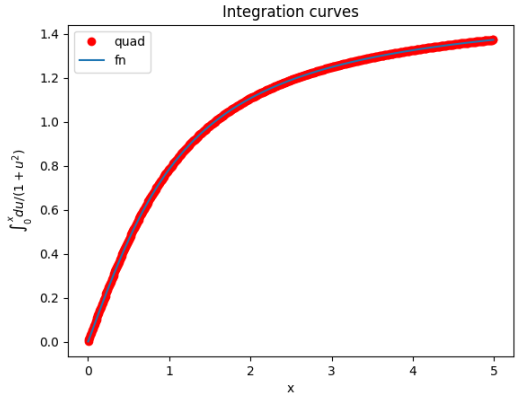


Figure 2: Plot of integrated function and arctan(x)

integrated	arctan(x)
0.000999999666667	0.000999999666667
0.0109995563655	0.0109995563655
0.0209969138166	0.0209969138166
0.0309900753886	0.0309900753886
0.0409770494768	0.0409770494768
0.0509558518771	0.0509558518771
0.0609245081383	0.0609245081383
0.070881055885	0.070881055885
0.0808235471054	0.0808235471054
0.0907500503959	0.0907500503959

Figure 3: Table of integrated function and arctan(x) for same x

5.3 Part 3:

The plot of the error in the integral is plotted on a semilog axis.It can be done by taking the absolute value of the difference between the calculated integral and the original function $\arctan(x)$. This can be dome by the following snippet of code:

```
import numpy as np
import matplotlib.pyplot as plt
error=np.abs(integrated-np.arctan(x_quad))
plt.semilogy(x_quad,error,'ro')
```

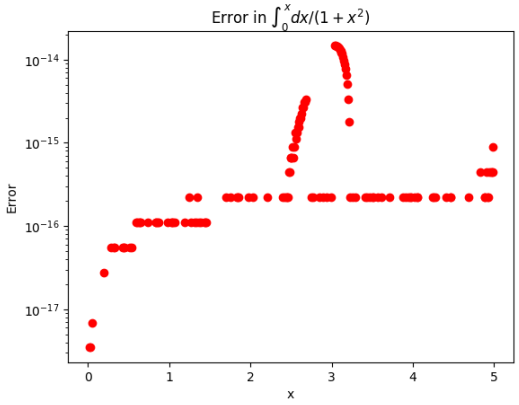
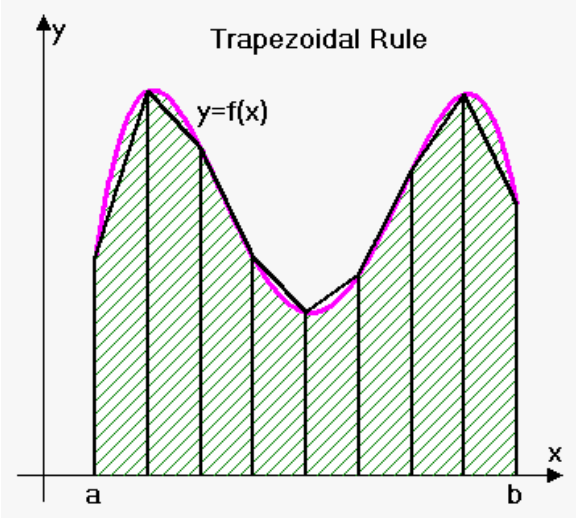


Figure 4: Plot or Error in the integration

6 Integrating the function using Trapezoidal Method

Trapezoidal Method is used for approximating a definite integral $\mathrm{I} = \int_a^b f(x)dx$ using linear approximations of the function f. The trapezoids are drawn as shown below.

The procedure for this method is as follows:
1: Partition [a,b] into the set $(X_0, X_1, X_2, ...X_n)$so that there are n sub-intervals of equal width.
2: The integral $\int_a^b f(x)dx$ is estimated by adding the areas of all the trapezoids as illustrated above.The width of Δx of each sub-interval given by



$\Delta x = \frac{b-a}{n}$.
$\int_a^b f(x)dx = \frac{\Delta x}{n}[f(x_0) + 2f(x_1) + 2f(x_2) + ... + 2f(x_n - 1) + f(x_n)]$

The integration can be done by two methods:
1: Using a for loop
This is done by summing up the array of $f(x_i)$ for each i and storing in another array.After this $\frac{1}{2}(f(x_0)+f(x_n))$ is subtracted from that array.Then it is divided by h.
2: Vectorizing using `np.cumsum`
This is done by applying the `np.cumsum` function to the array containing $f(x_i)$ which would result in an array containing the summation part of the formula.

```
#vector has f(xi)
# generating vector
for i in range(1,int(upper_limit/h)):
  x.append(upper)
  vector.append(function(upper))
  upper+=h
#vectorized integration
summation=np.cumsum(vector)
f_lower=np.ones(int(upper_limit/h)-1)*func_lower
sub=0.5*(f_lower+vector)
I=h*(summation-sub)
```

The integrated function is then plotted.

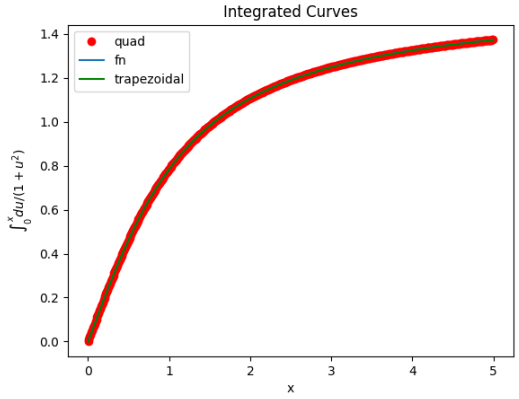


Figure 5: Plot of integrated function with `quad`,trapezoidal method and $\arctan(x)$

7 Error Estimation in Trapezoidal Method

7.1 Estimated Error

According to [1] and [2] the estimated error in trapezoidal method of integration for a continuous function $f(x)$ in the interval $a \leq x \leq b$ is:
$E_h^T(f) = -\frac{h^2(b-a)}{12} f"(c_h)$ for some c_h in the closed interval [a,b].
For our problem we have $f"(x)$ maximum at x=1 and it's value is $\frac{1}{2}$. Thus by plugging in the values we get $|E_h^T(f)| = h^2 \frac{5}{24}$

```
est_error=[]
#using the formula
def error_est_h(h):
 return (h**2*(upper_limit-lower_limit))/48

h=[]
h.append(0.01)
for i in range(1,10):
  h.append(h[i-1]/2)

for i in range(10):
  est_error.append(error_est_h(h[i]))
```

7.2 Exact Error

First of all an array containing the values of h is made by halving it ten times.Then for each value of h the integration value is calculated using the trapezoidal method.Then it is subtracted from $\arctan(x)$.Then the maximum absolute value of the error array is stored in `max_error_arr`.

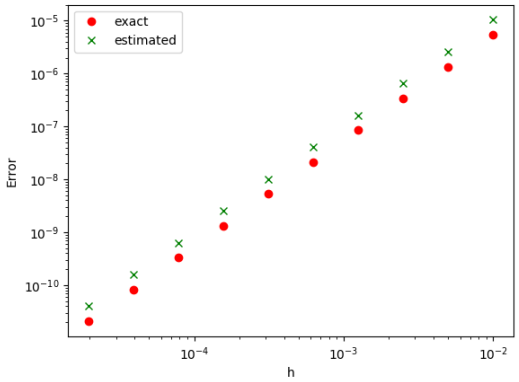


Figure 6: Estimated and Exact Errors

8 References

[1]:`http://homepage.divms.uiowa.edu/~atkinson/ftp/ENA_Materials/Overheads/sec_5-2.pdf`
[2]:`http://math.cmu.edu/~mittal/Recitation_notes.pdf`