Report - Assignment 2

- Akshay Anand (EE16B046)

February 6, 2018

Abstract

In this week's assignment, $\int_0^x \frac{1}{1+t^2} dt$ was calculated at different step-size h for the variable t using trapezoidal integration method and the result was compared with the actual value of the integral ($\tan^{-1}$x) and the result obtained using the quad function in scipy.integrate module. The actual value of integral ($\tan^{-1}$x) was plotted for $0 \leq x \leq 5$ and compared with the value obtained by the other 2 methods. Then, the maximum actual and estimated error of values obtained using trapezoidal integration method was plotted against decreasing h value (step-size). The error at different values of x when using the quad function was also plotted against x.

1 Function Definition

The function to integrate ($\frac{1}{1+t^2}$) is initially defined.

```
def function(t):
        return 1/(1+t*t)
```

2 Vector Initialisation

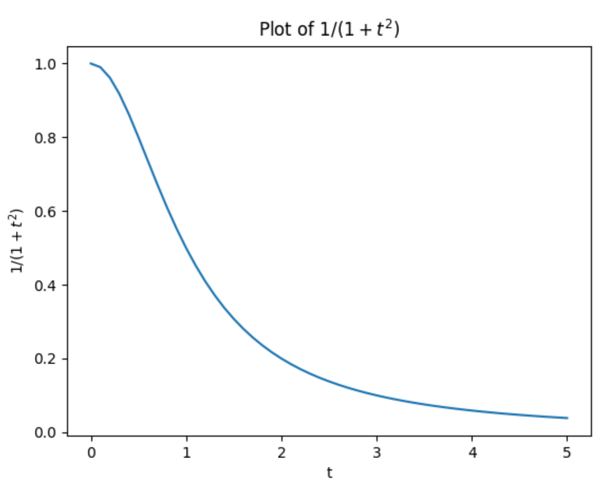
A vector that covers the region: $0 \leq x \leq 5$ in steps of $h = 0.1$ is initialised.

```
h = 0.1
x = np.linspace(0.0, 5.0, num=(5.0/h) + 1)
```

3 Plot of $1/(1+t^2)$ vs t

The function $1/(1+t^2)$ is now plotted against t.

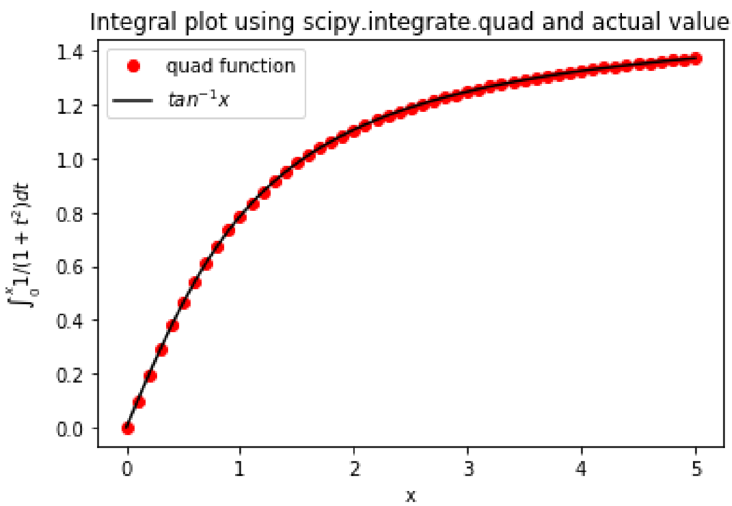
```
plt.plot(x, function(x))
plt.title('Plot of $1/(1+t^{2})$')
plt.xlabel('t')
plt.ylabel('$1/(1+t^{2})$')
plt.show()
```



4 Finding integral using scipy.integrate.quad

The integral of the function $1/(1+t^2)$ is now calculated from 0 to all values in the vector x using scipy.integrate.quad and its value is compared to the actual value of the integral i.e $tan^{-1}x$ and plotted in the same graph. Both the values for a partcular x are tabulated in the following table and saved as a .csv file. The error between the two values is also plotted against x.

```
y = []
for i in x:
        y.append(quad(function, 0, i)[0])
df = pd.DataFrame()                    #df is a pandas DataFrame to tabulate the values
df['tan^(-1) x'] = np.arctan(x)
df['Integral with quad'] = y
df.to_csv('table.csv')

plt.plot(x, y, 'ro')                          #Plot the result of the integral
plt.plot(x, np.arctan(x), '#000000')     #Plot the actual value of tan^(-1) x
plt.legend(('quad function','$tan^{-1} x$'))
plt.title('Integral plot using scipy.integrate.quad and actual value')
plt.xlabel('x')
plt.ylabel('$\int_0^x 1/(1+t^2) dt$')
plt.show()
```



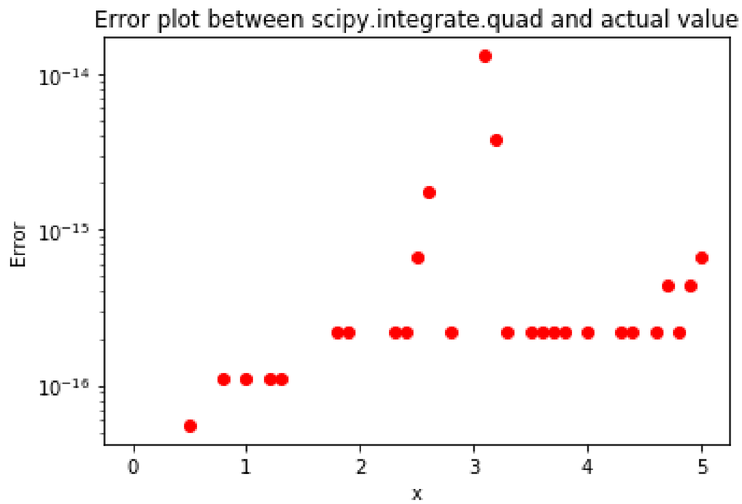
The tabulated value obtained using pandas DataFrame in the above code segment:

	tan^(-1) x	Integral with quad
0	0.0	0.0
1	0.09966865249116204	0.09966865249116204
2	0.19739555984988078	0.19739555984988078
3	0.29145679447786715	0.29145679447786715
4	0.3805063771123649	0.3805063771123649
5	0.4636476090008061	0.46364760900080615
6	0.5404195002705843	0.5404195002705843
7	0.6107259643892087	0.6107259643892087
8	0.6747409422235527	0.6747409422235526
9	0.7328151017865066	0.7328151017865066
10	0.7853981633974483	0.7853981633974484
11	0.8329812666744317	0.8329812666744317
12	0.8760580505981935	0.8760580505981934
13	0.9151007005533605	0.9151007005533603
14	0.9505468408120752	0.9505468408120752
15	0.982793723247329	0.982793723247329
16	1.0121970114513341	1.0121970114513341
17	1.039072259536091	1.039072259536091
18	1.0636978224025597	1.0636978224025595
19	1.0863183977578734	1.0863183977578736
20	1.1071487177940904	1.1071487177940904
21	1.1263771168937977	1.1263771168937977
22	1.1441688336680205	1.1441688336680205
23	1.1606689862534056	1.1606689862534054
24	1.1760052070951352	1.176005207095135
25	1.1902899496825317	1.1902899496825323
26	1.2036224929766774	1.2036224929766792
27	1.2160906747839564	1.2160906747839564
28	1.2277723863741932	1.2277723863741934
29	1.2387368592520112	1.2387368592520112
30	1.2490457723982544	1.2490457723982544
31	1.2587542052323633	1.2587542052323764
32	1.2679114584199251	1.267911458419929
33	1.2765617616837088	1.2765617616837086
34	1.2847448850775784	1.2847448850775784
35	1.2924966677897853	1.2924966677897851
36	1.299849476456476	1.2998494764564759
37	1.3068326031691921	1.306832603169192
38	1.313472611823808	1.3134726118238083
39	1.319793640151862	1.319793640151862
40	1.3258176636680326	1.3258176636680323
41	1.331564726831236	1.331564726831236
42	1.3370531459259951	1.3370531459259951
43	1.3422996875030344	1.3422996875030342
44	1.3473197256542637	1.3473197256542635
45	1.3521273809209546	1.3521273809209546
46	1.356735643231075	1.3567356432310749
47	1.3611564809206842	1.3611564809206838
48	1.3654009376051293	1.365400937605129
49	1.3694792184202558	1.3694792184202562
50	1.373400766945016	1.3734007669450166

The error is now plotted in a semi-log plot where error is in a logarithmic scale and x, in linear scale.

```
plt.semilogy(x, abs(y - np.arctan(x)), 'ro')
plt.title('Error plot between scipy.integrate.quad and actual value')
plt.xlabel('x')
plt.ylabel('Error')
```

```
plt.show()
```



5 Trapezoidal Integration

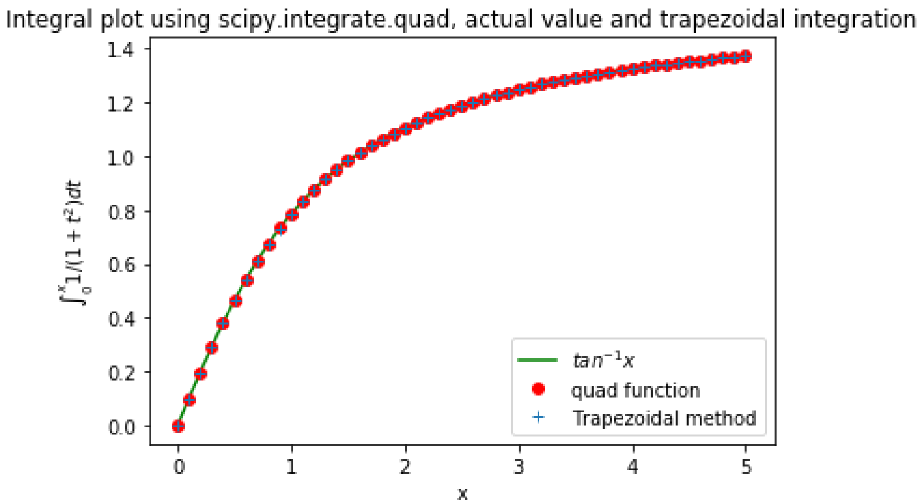
Now, the value of the integral is found using the trapezoidal integration method where the graph of the function is divided into thin trapezoids of width h (step-size) and the area of each is calculated and added together to get the integral.

The formula used is:

$$I_i = h\left(\sum_{j=1}^{i} f(x_j) - \frac{1}{2}\left(f(x_1) + f(x_i)\right)\right)$$

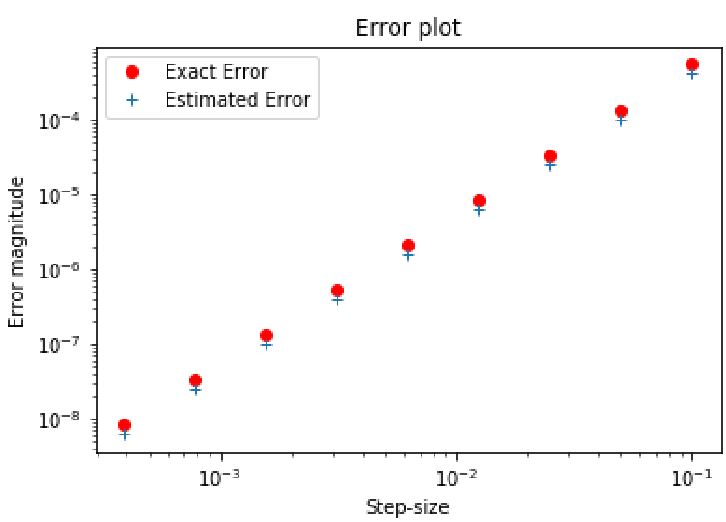
The value so obtained is now plotted in the same plot as the other 2 methods (actual value of $tan^{-1}x$ and quad method).

```
def trapezoidalIntegral(h,a,b):
        x = np.linspace(0.0, 5.0, num = (5.0/h + 1))
        y = h*(np.cumsum(function(x)) - 0.5*(function(0) + function(x)))
        return y

trapezoid = trapezoidalIntegral(0.1,0,5)
plt.title('Integral plot using scipy.integrate.quad, actual value and trapezoidal integration
plt.xlabel('x')
plt.ylabel('$\int_0^x 1/(1+t^2) dt$')
plt.plot(x, np.arctan(x), 'g')
plt.plot(x, y, 'ro')
plt.plot(x, trapezoid, '+')
plt.legend(('$tan^{-1} x$', 'quad function', 'Trapezoidal method'))
plt.show()
```



Now, the maximum actual and estimated error of the values obtained using trapezoidal integration are plotted against step-size h. The estimated error is calculated by subtracting the values obtained by trapezoidal method at the same point at current step-size and half its step-size while actual error is calculated as difference with $tan^{-1}x$ at each point.

```
estError = []
actError = []
h = 0.1
hList = []
maxError = 1
while maxError > 10**-8:
        trapezoid = trapezoidalIntegral(h,0,5)
        actError.append(max(abs(trapezoid - np.arctan(np.linspace(0, 5, num = (int)(5/h + 1)
        hList.append(h)
        h = h/2
        nextTrapezoid = trapezoidalIntegral(h,0,5)
        maxError = max(abs(trapezoid - nextTrapezoid[::2]))
        estError.append(maxError)

plt.loglog(hList, actError, 'ro')
plt.loglog(hList, estError, '+')
plt.title('Error plot')
plt.xlabel('Step-size')
plt.ylabel('Error magnitude')
plt.legend(('Exact Error','Estimated Error'))
plt.show()
```



6 Conclusion

Thus, comparing values of $\int_0^x 1/(1+t^2)dt$ obtained by scipy.integrate.quad and trapezoidal integration to the actual value of the integral at any point x (i.e $tan^{-1}x$), we see that scipy.integrate.quad gives more accurate values as the maximum error is about 10^{-14} while in trapezoidal integration, at an h value of almost 10^{-4}, the maximum error is about 10^{-8}. Also, if h is halved further, the time to compute the integral will increase, as the number of points in the vector x increases exponentially (no. of elements in x $= \frac{5}{h} + 1$). Hence, the quad function can be considered as an efficient and accurate estimate of the actual value of the integral and the trapezoidal method can be used to provide a reasonable estimate of the integral.

Also, it is observed that the difference between the estimated error and the actual error is really minimal at any step-size. This means that, in a real-world scenario, where the actual value of the integral is not known, the estimated error (which does not depend on the actual value of the integral) can be considered as the actual error itself and can be used to accurately find the integral.