CA_lab_integration_GL

October 26, 2020

#Romberg's Integration #### Computational Astrophysics * Shivam Kumaran * SC17B122 * 26 Oct 2020

Open Program

0.0.1 Function Transformation

Transforming Function to convert \ integration limit (-1,1)

[1]:
```
import numpy as np
```

[2]:
```
def trans_f(f, a, b):
    '''
    Returns transformed function,
    Such that the limits of integration
    will be -1,1
    '''
    m = (b-a)/2
    c = (b+a)/2
    def f_mod(x):
        val = ((b-a)/2)*f(m*x+c)
        #print(val)
        return val
    return f_mod
```

0.0.2 Calculation of integral ,

For given value of * Number of points N * Given Function - f * Integration Limits (a,b)

[56]:
```
def calc_gauss(f ,a, b , N):
    #obtains weights and roots from Numpy function
    x,w = np.polynomial.legendre.leggauss(N)
    # transforms function to match limits (-1,1)
    val = [trans_f(f,a,b)(x_n) for x_n in x]
    return(np.dot(val ,w))
```

0.0.3 Analysis

Given Function and the integation limit, here we find the number of poiints of evaluation of integral Corresponding to the provided precision

returns N , Integral Value, corresponding error

[57]:
```
def analysis(f,a,b,e):
    err = 1
    prev = calc_gauss(f,a,b,1)
    i = 2
    while(err>e):
        nxt = calc_gauss(f,a,b,i)
        err = abs(prev-nxt)
        prev = nxt
        i+=1
    return(i-1 , nxt , format(err , '1.2e'))
```

1 Problem Function 01

[29]:
```
def f_a(x):
    val = np.exp(-1*x**2)
    return val

a,b = 0,2
e = 1e-8
N , val , err = analysis(f_a,a,b,e)
print("N: {}\nIntegration Value:{:.8f}\nError{}".format(N,val,err))
```

```
N: 8
Integration Value:0.88208139
Error  7.13E-09
```

1.1 Problem function 02

[46]:
```
def f_b(k):
    def f(x):
        val=1/((1-k*(np.sin(x))**2)**0.5)
        return val
    return f

k=0.2
a,b = 0,2
e = 1e-8
N , val , err = analysis(f_b(k),a,b,e)
print("N : {}\nIntegration Value : {:.8f}\nError : {}".format(N,val,err))
```

```
N : 8
```

```
Integration Value : 2.13600612
Error : 1.24e-10
```

1.1.1 Function 02 : K dependecy

[58]:
```
k=0.2
a,b = 0,2
e = 1e-8
k_list = np.linspace(0 ,  0.9 , 10)
print('K \t \t N \t Value \t \t Error')
print('_____________________________________________')
for k in k_list:
    N , integ_val , err = analysis(f_b(k),a,b,e)
    print('{:.6f} \t {} \t  {:.8f} \t {}'.format(k, N ,integ_val, err))
```

```
K               N       Value           Error
-----------------------------------------------
0.000000        2        2.00000000     0.00e+00
0.100000        8        2.06338710     1.11e-09
0.200000        8        2.13600612     1.24e-10
0.300000        10       2.22059055     2.13e-09
0.400000        10       2.32122529     9.21e-09
0.500000        12       2.44438264     4.13e-09
0.600000        13       2.60116462     3.57e-09
0.700000        15       2.81292508     4.76e-10
0.800000        17       3.12896328     2.98e-09
0.900000        24       3.71144323     2.88e-09
```

1.1.2 Conclusion

We see that the nuber of points depends on the value of k, which increases as k approaches 1

[ ]:
```
```