Applied Computational Science I

Lab: Adaptive Quadrature

Anand Kamble

amk23j@fsu.edu

17th November 2023

In [3]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [4]:
```
from kronrod import kronrod
```

The period of a simple pendulum is given by an integral,

$$K(x) = \int_0^{\frac{\pi}{2}} \frac{d\theta}{\sqrt{1 - x^2 sin^2\theta}}$$

In [24]:
```
def K(x,theta):
    return 1/np.sqrt(1 - x**2*(np.sin(theta)**2))
```

To transform the integral domain from [0,π/2] to [-1,1], we can use

$$\int_a^b f(x)dx = \frac{b-a}{2}\sum_{i=0}^{n} w_i f(\frac{b-a}{2}x_i + \frac{b-a}{2})$$

In [38]:
```
def f(n):
    tol = 1e-5
    Nodes,w1,w2 = kronrod(n,tol)
    y0 = np.zeros([n+1])
    x = np.linspace(0,1,51)
    y = np.pi/4 * K(0.5, (np.pi / 4 * x) + (np.pi/4))
    intgG = np.pi/4 *np.sum(w1 * K(0.5,Nodes))
    intgK = np.pi/4 *np.sum(w2 * K(0.5,Nodes))
    intgG_ = np.pi/4 *np.sum( w1 * K(0.5,-Nodes[::-1]))
    intgK_ = np.pi/4 *np.sum( w2 * K(0.5,-Nodes[::-1]))
    G_Result = intgG_ + intgG
    K_Result = intgK_ + intgK
    error = np.abs(G_Result - K_Result)
    return G_Result, K_Result, error
```

1.

Evaluating the integral for $0 \leq x \leq 1$ and increasing the value of n at every iteration until the error is less than the tolerance.

In [46]:

```
tolerance = 1e-2
maxN = 100

for n in range(1,maxN):
    intgG, intgK, error = f(n)
    if(error <= tolerance):
        print(f"Error is Less than tolerance at n={n}")
        break
    if(n == (maxN-1)):
        print("Reached Max number of N")
```

```
Reached Max number of N
```

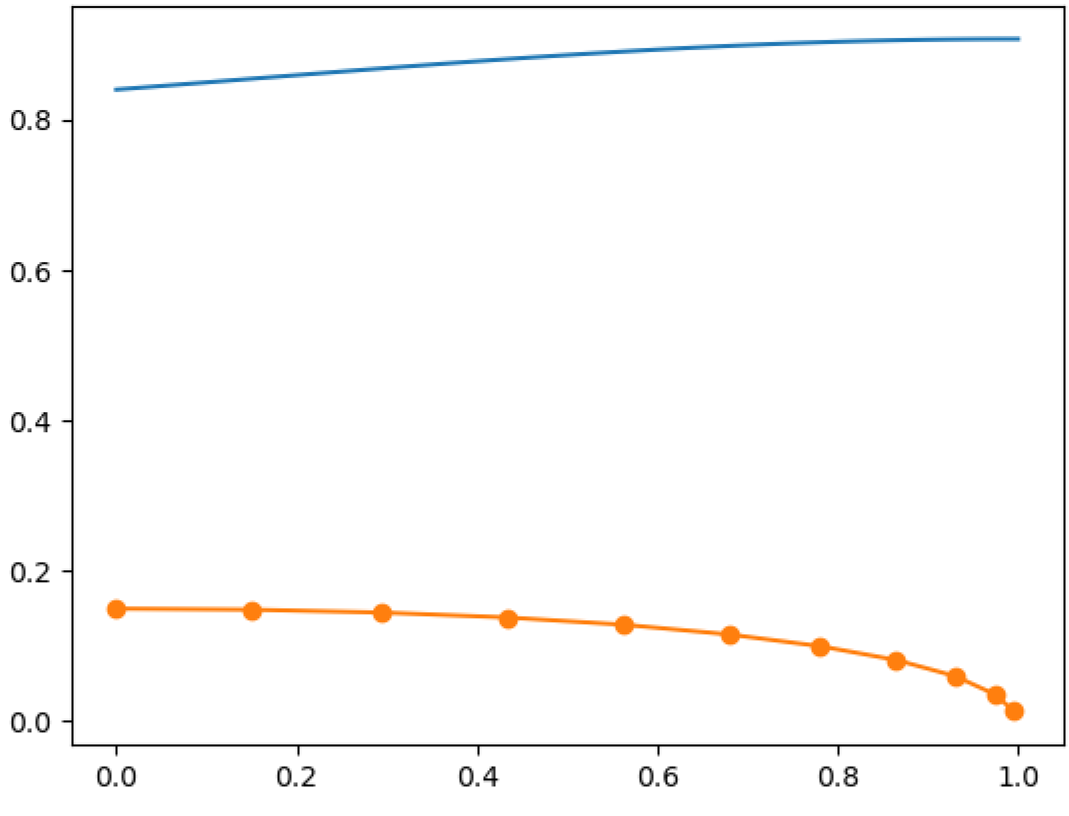
2.

Using $x = 0.5$, we get,

In [67]:

```
n = 10
tol = 1e-5

Nodes,w1,w2 = kronrod(n,tol)

y0 = np.zeros([n+1])

x = np.linspace(0,1,51)
y = np.pi/4 * K(0.5, (np.pi / 4 * x) + (np.pi/4))

plt.plot(x,y,Nodes,w1 * K(0.5,Nodes),'o-')



intgG = np.pi/4 *np.sum(w1 * K(0.5,Nodes))
intgK = np.pi/4 *np.sum(w2 * K(0.5,Nodes))

intgG_ = np.pi/4 *np.sum( w1 * K(0.5,-Nodes[::-1]))
intgK_ = np.pi/4 *np.sum( w2 * K(0.5,-Nodes[::-1]))

G_Result = intgG_ + intgG
K_Result = intgK_ + intgK

error = G_Result - K_Result
print('Integral by Gauss Weights          : ',G_Result)
print('Integral by Gauss-Kronrod Weights : ', K_Result)
print('Benchmark                          : ',np.pi/2)
print("Error                              : ",error)
```

```
Integral by Gauss Weights          :  1.7755579623831594
Integral by Gauss-Kronrod Weights :  1.6521617510399158
Benchmark                          :  1.5707963267948966
Error                              :  0.12339621134324363
```


0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0