COSC 343: Test 2

Micah Sherry

April 9, 2024

1 Fresnel Integrals

The intesity of difracted light near a straight edge is determied by the values of the Fresnel Integrals:

$$C(x) = \int_0^x cos(x)dx$$

and

$$S(x) = \int_0^x sin(x)dx$$

Use a quadrature routine to evaluate the integrals for enough values of x to draw a smooth plot of C(x) and S(x) over the range $0 \leq x \leq 5$

For my quadrature routine I used a composite four point Guassian routine. For the Fresnel integrals I set the number of subintervals to $2(|x|+1)$. To get a smooth plot on the interval [0,5] I needed to use 120 subintervals.

Code:

```
import numpy as np
import matplotlib.pyplot as plt




def integrate(f,A,B, numInt=10):
    def GaussianQuadrature(f,a,b):
    #this method workes by linearly mapping [a,b] to [-1,1] and multiplying by (b-a)/2
        w = [0.347854845137454,    0.652145154862546,
            0.652145154862546, 0.347854845137454]

        x = [-0.861136311594053, -0.339981043584856,
            0.339981043584856, 0.861136311594053]
        slope = (b-a)/2
        def map(x):
            # derived from the point-slope form of a line
            return slope * (x+1) + a
        area = 0
        for i in range(len(w)):
            area += slope*w[i] * f(map(x[i]))
        return area

    if (numInt<1):
        raise ValueError("Cannot have a number of intervals less than 1")

    x_points= np.linspace(A,B,numInt+1)
    #print(x_points)
```

```
    area = 0
    for i in range(len(x_points)-1):
        area += GaussianQuadrature(f,x_points[i],x_points[i+1])
    return area

if __name__=="__main__":
    def C(x):
        def integrand(t):
            return np.cos((np.pi*t**2)/2)
        return integrate(f = integrand , A=0 ,B=x, numInt=2*(1+int(np.abs(x))))
    def S(x):
        def integrand(t):
            return np.sin((np.pi*t**2)/2)
        return integrate(f = integrand , A=0 ,B=x, numInt=2*(1+int(np.abs(x))))

    xpts = np.linspace(0,5,140)
    Cpts = []
    Spts = []
    for x in xpts:
        Cpts.append(C(x))
        Spts.append(S(x))
    plt.plot(xpts,Cpts,"k")
    plt.xlabel("x")
    plt.ylabel("C(x)")
    plt.show()
    plt.xlabel("x")
    plt.ylabel("S(x)")

    plt.plot(xpts,Spts,"k")

    plt.show()
```

Plots:

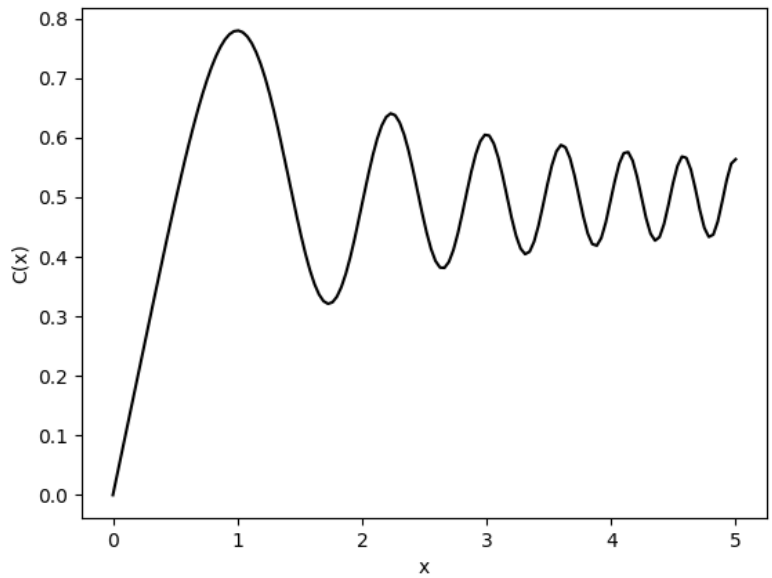

Figure 1: Fresnel integral C(x)

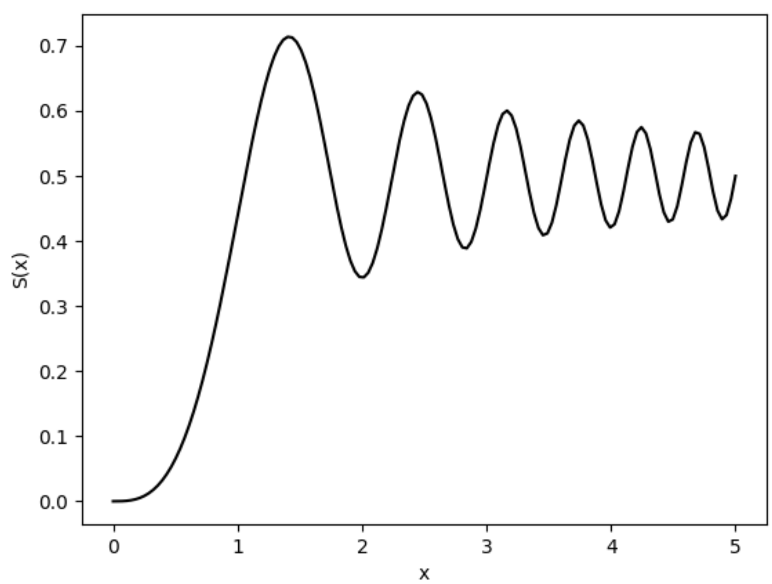

Figure 2: Fresnel integral S(x)