Section3

Elliott Ashby

November 6, 2022

1 q1

```
def f(x):
    return (x ** 4 * ((1 - x) ** 4) / (1 + (x ** 2)))
```

This is a function that returns the result of the integrand. We can then input it into trap0 like:

```
    n = 3
    a = [trap0(f, 0, 1, 1), trap0(f, 0, 1, 2)]
    while not np.fabs((a[-1] - a[-2])) < 1.0e-6:
        a.append(trap0(f, 0, 1, n))
        n += 1
```

Here, if the change in the returned value is smaller than $1.0e^{-6}$ the list stops being added to as n increases, meaning a[-1] is the final value with an accuracy of $1.0e^{-6}$.

2 q2

Printing a[-1] and n after running the above code results in

$$\int_0^1 \frac{x^4(1-x)^4}{1+x^2} dx = 0.0012649570769764054$$

3 q3

This result in section q2 results from 7 trapeziums.

4 q4

Using:

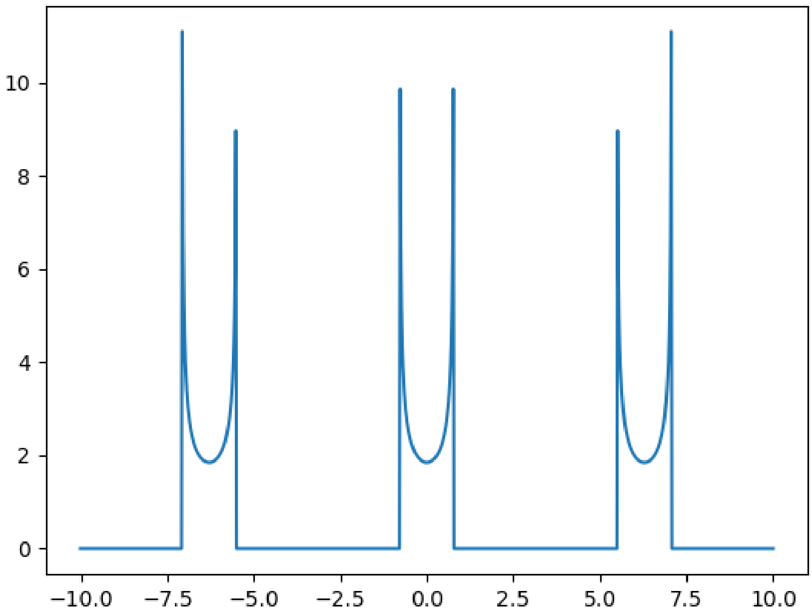
```
from q4_1 import trap1
import numpy as np


def fe(x):
    return np.e ** (-x ** 2)
```

```
if __name__ == '__main__':
    x = 10
    print(trap1(fe, -x, x, 1.0e-3))
```

We print a value for the integral

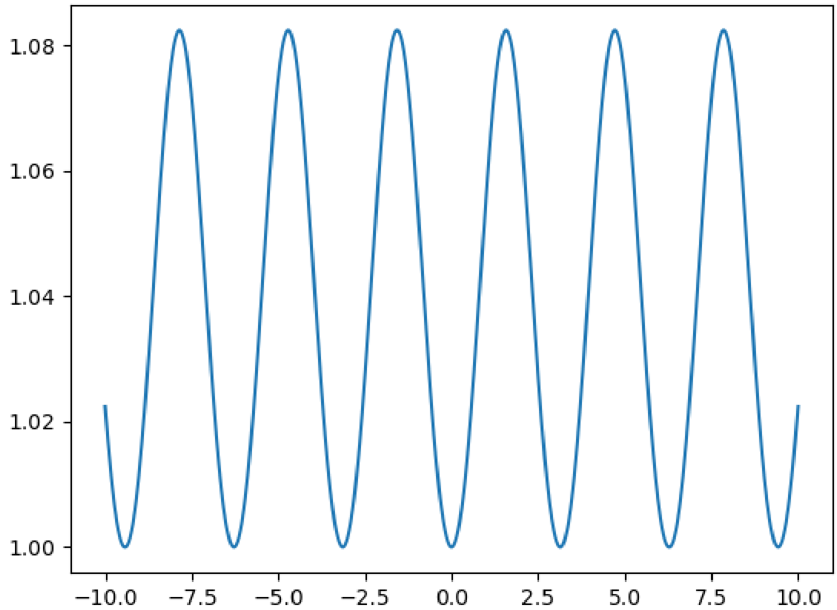
$$\int_{-\infty}^{\infty} e^{-x^2} dx = 1.7724538509055157$$

5 q5



Here, we can see spikes to infinity at values where $\alpha = \theta$. The inconsistent height of the asymptotes are because only 1000 discrete value were calculated to graph, meaning that these asymptotes are only at $\alpha \approx \theta$.

6 q6

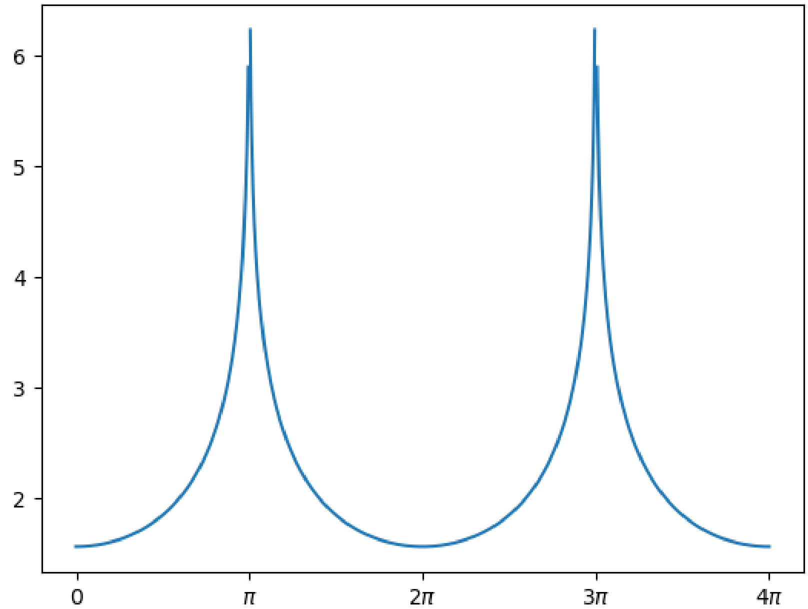


Any numerical integration will fail due to division by zero.
After changing the variable we plot as above.

7 q7

```
from q4_1 import trap1
import numpy as np
from math import pi, sin
from matplotlib import pyplot as plt
from tabulate import tabulate


def period2(phi):
    return 1 / (1 - (sin(a / 2) ** 2) * (sin(phi)) ** 2) ** 0.5


if __name__ == '__main__':
    global a
    alphas = np.linspace(0, 4 * pi, 1000)
    q = []
    for a in alphas:
        q.append([a, trap1(period2, 0, pi / 2, 1.0e-3)])
```

```
#        print(f'for alpha {x[0]}, the ratio is {x[1] * 2 / pi}')
    print(tabulate(q, headers=['alpha', 'ratio']))

    plt.plot(alphas, [i[1] for i in q])
    plt.xticks(ticks=[0, pi, 2 * pi, 3 * pi, 4 * pi], labels=['0',
                        '$\pi$', '2$\pi$', '3$\pi$', '4$\pi$'])
    plt.savefig('q4_7.png')
```



In this script we create a large list of 1000 numbers equally spaced 0 through 4π. Then for each value of alpha we calculate the integral and add it to the end of a list. We can now tabulate the data since the appended list is a list of lists including the alpha value used.
Plotting the alpha vs the integral reveals that when alpha equals π and 3π the integral is equal to infinity since this is also division by zero.

8 q8

```
    for x in q1:
        x[0] = round(x[0], 2)

    q1 = np.array(q1)
    condition1, condition2 = list(np.where(q1 == 1.57))
    condition1 = int(condition1)
    condition2 = int(condition2 + 1)
    print(f'when amplitude is at roughly pi/2, the value of \
T/T_0 is { 2 / pi * q1[condition1][condition2] }')
```

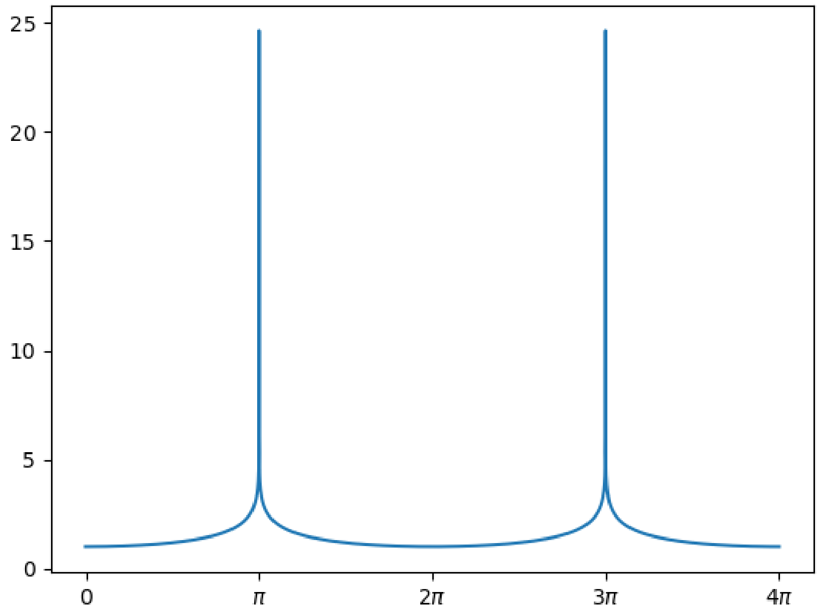
In order to gain a value of T/T_0 at 90° we simply round both $\pi/2$ and our values of alpha to 2 decimal places, find the index of where $\alpha \approx \pi/2$ and print the associated ratio.

at $\alpha = 90°$:

$$\frac{T}{T_0} = \frac{2}{\pi}\int_0^{\pi/2} \frac{d\phi}{(1-\sin^2(\alpha/2)\sin^2\phi)^{1/2}} = 1.1807649945835401 \tag{1}$$

9 bonus

```
import sympy as sym
import numpy as np
from matplotlib import pyplot as plt
import bisect


# having a gander at algebraic integration in python

x = sym.Symbol('x')
theta = sym.Symbol('theta')
a = sym.Symbol('a')
# y = sym.integrate(sym.exp(-x ** 2), (x, -sym.oo, sym.oo))
# print(y)


integ = sym.Integral(1 / ((1 - sym.sin(a / 2) ** 2 * (sym.sin(theta) ** 2)
                         ) ** 0.5), (theta, 0, np.pi/2))
alphas = np.linspace(0, 4 * np.pi, 5000)
alphas = list(alphas)
bisect.insort(alphas, np.pi)
bisect.insort(alphas, 3 * np.pi)

k = integ.doit()
q = []
for alpha in alphas:
    q.append([alpha, 2 / np.pi * k.subs(a, alpha)])
plt.plot(alphas, [i[1] for i in q])
plt.xticks(ticks=[0, np.pi, 2 * np.pi, 3 * np.pi, 4 * np.pi], labels=['0',
                        '$\pi$', '2$\pi$', '3$\pi$', '4$\pi$'])
plt.savefig('q4_bonus.png')

# find exact value of amplitude 90 deg
print(2 / np.pi * k.subs(a, np.pi/2))
```



Using the sympy package we can evaluate the integral in q6 entirely algebraicly and then after doing so, sub in values for iα, including subbing in specific values, either by subbing in separetely or inserting specific values into the list created in order to graph α vs T/T_0.

Subbing $\pi/2$ directly into α of the integrated equation and multiplying by $2/\pi$ gives a result:

$$\frac{T}{T_0} = \frac{2}{\pi}\int_0^{\pi/2} \frac{d\phi}{(1-\sin^2(\alpha/2)\sin^2\phi)^{1/2}} = 1.18034059901610 \tag{2}$$