In [40]:
```
import math
import numpy as np
import matplotlib.pyplot as plt
import random
```

Problem 2:

b) Ironically this didn't work because numpy doesn't store that many digits?

In [53]:
```
A = np.array([[1,1],[1.0000000001,1-1e-10]])
A_t = A.T
Ai = np.array([[1-1e10,1e10],[1+1e10,-1e10]])
Ai_t = Ai.T
```

In [54]:
```
print(A)
print(np.linalg.eig(np.matmul(A_t,A)))
```

```
[[1. 1.]
 [1. 1.]]
EigResult(eigenvalues=array([4.0000000e+00, 4.4408921e-16]), eigenvectors=array([[ 0
.70710678, -0.70710678],
       [ 0.70710678,  0.70710678]]))
```

Problem 3:

c)

In [2]:
```
x = 9.999999995000000e-10
#checking the given algorithm
y=np.exp(x)
f=y-1
```

In [3]:
```
print(f)
```

```
1.000000082740371e-09
```

This gives accuracy up to 10^{-7}

d)

In [10]:
```
#the function to be used by the iterative method below
fxn= lambda n,x: (x**n)/math.factorial(n)
```

In [56]:
```
ftot=0
n=1

while (n<100 and (ftot!=1e-9)):
    #iterating through each term of the taylor series, and
    #stopping at either n=100, or when the calcualted value is
    #the same as the expected.
```

```
    print(ftot)
    ftot= ftot + fxn(n,x)

    print(n)
    n+=1

print(ftot)
```

```
0
1
9.999999995e-10
2
1e-09
```

So taylor series is $T(f(x)) = x + x^2/2, n = 2$

Problem 4:

a)

In [57]:
```
t=np.linspace(0,np.pi,31)

y=np.cos(t)
```

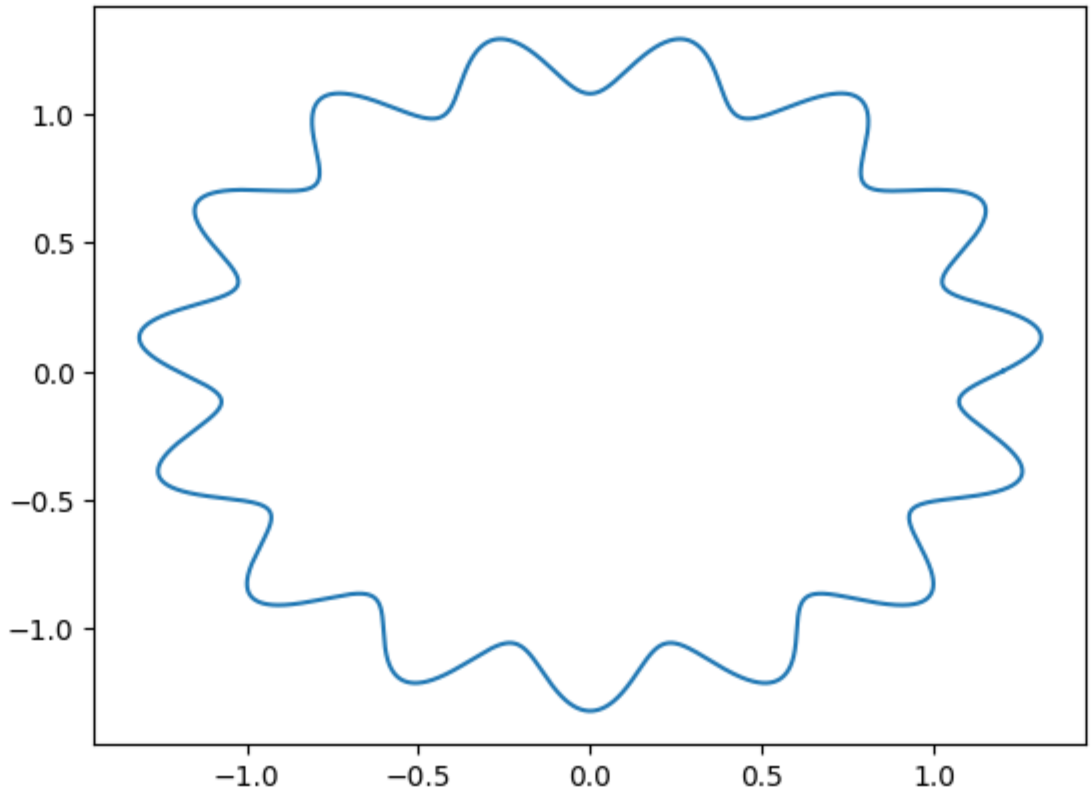
In [31]:
```
k=1
S=0
while k<len(t):
    #summing by adding previous value to current calculated value
    S=S+t[k]*y[k]
    k+=1

print("the sum is:",S)
```

```
the sum is: -20.686852364346837
```

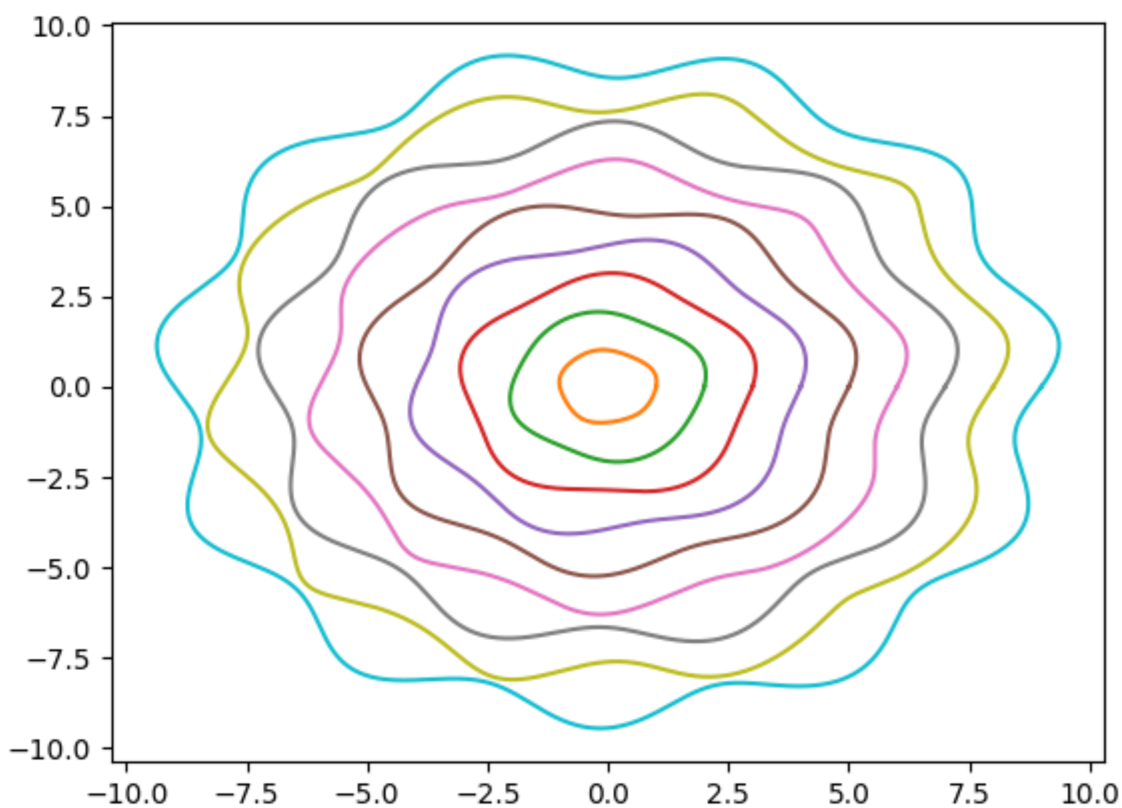
b)

In [ ]:
```
#setup
theta = np.linspace(0,2*np.pi,10000)
R=1.2
deltar=0.1
f1=15
p=0
#Establishing x and y
x4 = lambda R,deltar,f1,p: R*(1+deltar*np.sin(f1*theta+p))*np.cos(theta)
y4 = lambda R,deltar,f1,p: R*(1+deltar*np.sin(f1*theta+p))*np.sin(theta)
#plot
plt.plot(x4(R,deltar,f,p),y4(R,deltar,f,p))
```

Out[ ]:
```
[<matplotlib.lines.Line2D at 0x1c86f15b790>]
```



In [41]:
```
for i in range(10):
    #plot each one with ith term
    plt.plot(x4(i,0.05,2+i,p),y4(i,0.05,2+i,random.uniform(0,2)))
```



In [ ]: