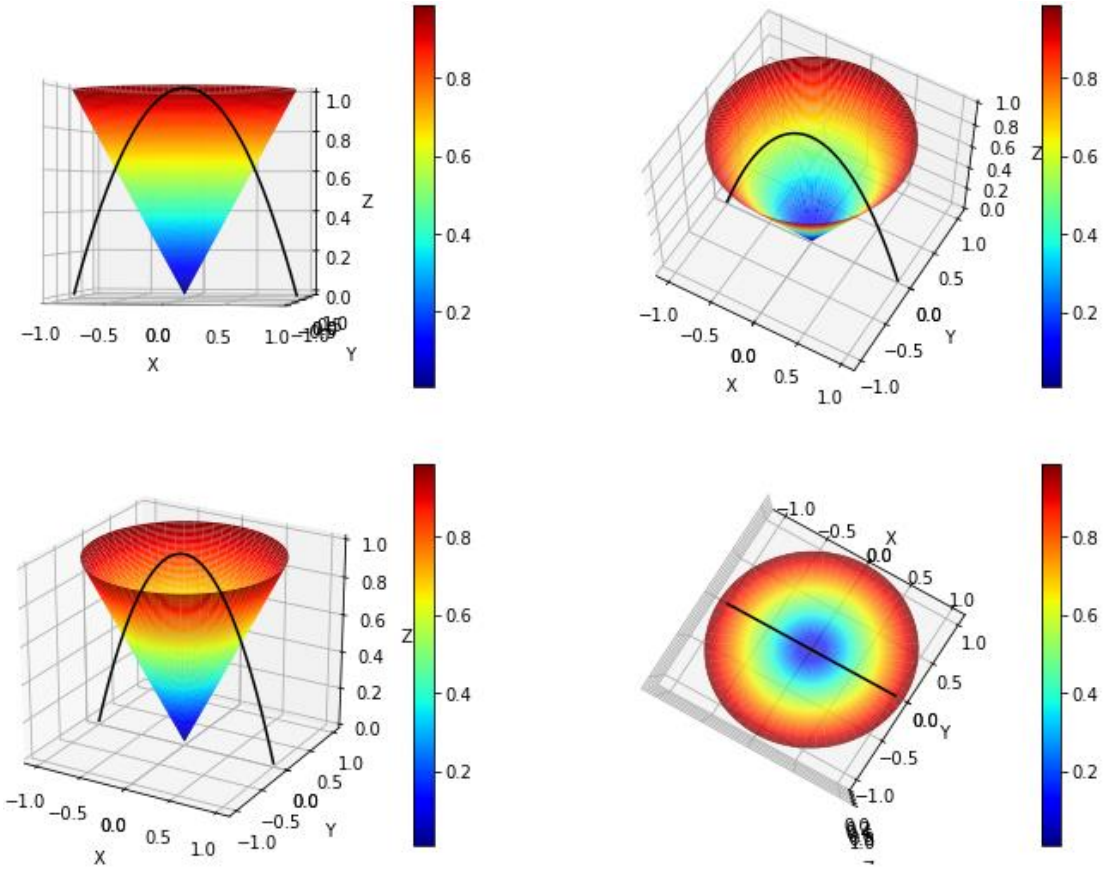
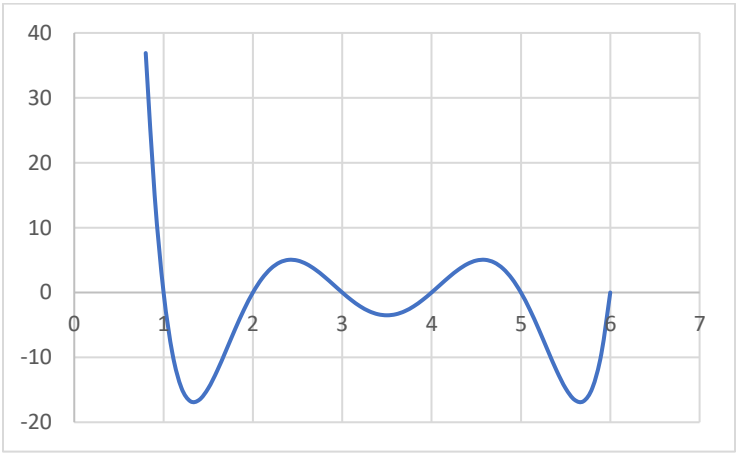
Q3.1)

The Results are generated identical to what was asked:



Q3.2)

The following equation was chosen which crosses x axis 6 times:



Scipy library was called with method of scipy.optimize.fsolve and all root were founded. In addition sympy was used to give the roots and analytical solutions for the roots.

In [1338]:
```
import math
def func(x):
    return x**6-21*(x**5)+175*(x**4)-735*(x**3)+1624*(x*x)-1764*x+720
func(2)
```

Out[1338]:
```
0
```

In [1339]:
```
import numpy as np
import scipy
ans_scipy = scipy.optimize.fsolve(func, np.arange(0,10,1))
```

In [1340]:
```
ans_scipy
```

Out[1340]:
```
array([1., 1., 2., 3., 4., 5., 6., 6., 6.])
```

In [1341]:
```
import sympy as sm
x, y = sm.symbols('x y')
sm.solve(x**6-21*(x**5)+175*(x**4)-735*(x**3)+1624*(x*x)-1764*x+720)
```

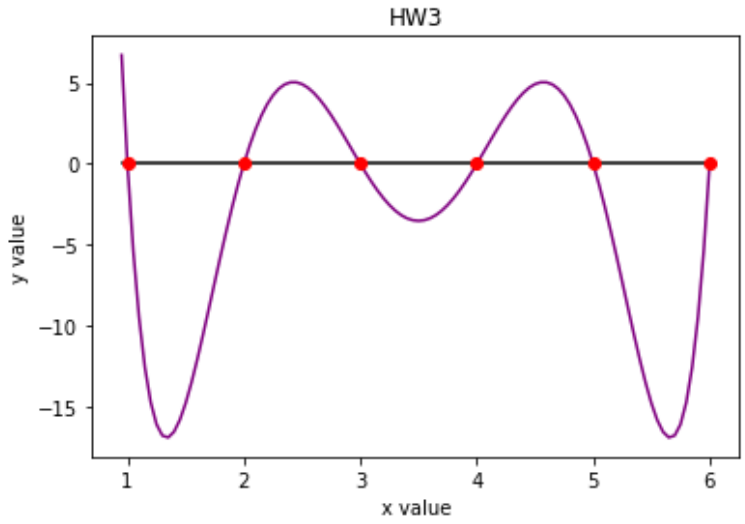
Out[1341]:
```
[1, 2, 3, 4, 5, 6]
```

The analytical solutions are rather length and are not included here. Please go to Python file.

For the last sub section, two separate classes were defined which could generate 2D plot (for line, dot, or scatter of dots) and several 3D plots at various angle. Please go to the HW3.ipynb for the code and results.
At the end, I used my created class to plot my previously generated function

In [1349]:
```
x=np.arange(0.95,6.05,0.05)
y=func(x)
yl=0*x
FirstClu=graph(x,y)
FirstClu.line(color='Purple',title='HW3')
FirstClu=graph(x,yl)
FirstClu.line(color='Black',title='HW3')
a=[]
for i in range(len(x)):
    if -0.01<func(x[i]) and func(x[i])<0.01:
        a.append(x[i])

FirstClu=graph(a,func(np.asarray(a)))
FirstClu.scatter(color='Red',title='HW3')
```



Q3.3)

In this question, I found out that Sympy was not able to analytically solve the equation for x to give the cross section points. However, Scipy was successfully used to obtain the roots.

In [1350]:

```
import matplotlib.pyplot as plt
import matplotlib as mpl
import numpy as np
from scipy.optimize import fsolve
from scipy import optimize
x = np.linspace(0,10,300)
y = (3 + x)+2*np.sin(4*x)
y2 = (20 - 1.5*x)+2*np.cos(5*x)
plt.figure()
plt.plot(x,y,'b-')
plt.plot(x,y2,'r-')
plt.xlabel('x')
plt.ylabel('y')
f = lambda x : abs((20 - 1.5*x)+2*np.cos(5*x)-((3 + x)+2*np.sin(4*x)))
res = np.array([0.0, 20])
#scipy.optimize.brute(f, ranges=((-2,3),) )
ans=scipy.optimize.root(f,[5,7,8])
ans.x
FirstClu=graph(ans.x,(20 - 1.5*ans.x)+2*np.cos(5*ans.x))
FirstClu.scatter(color='Green',title='HW3')
plt.show()
```