

# Mie Theory Task : 3

Name: Jayendra Praveen Kumar Chorapalli

Matriculation Number: 03736602

Importing necessary libraries

```
In [256]: import math
import numpy as np
import matplotlib.pyplot as plt
from scipy.special import eval_legendre
```

Definition and assignment of variables g and u

```
In [268]: g=0.8
u = np.arange(-1,1+0.01,0.01)
```

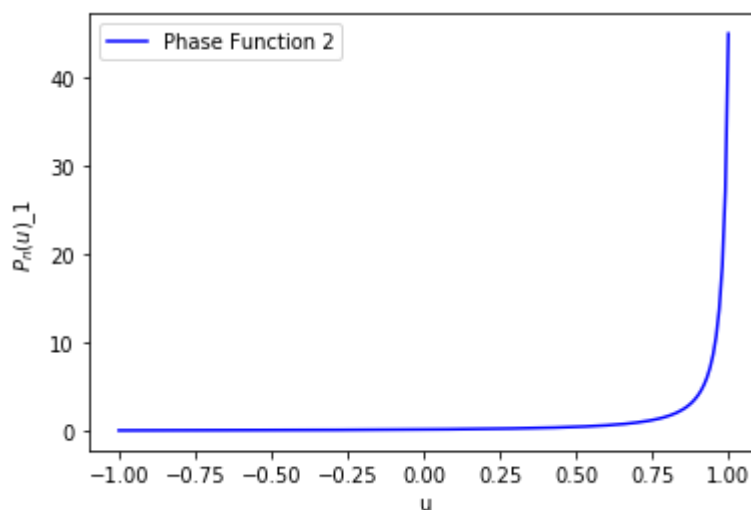
Phase Function 1

```
In [270]: p = []
for i in u:
    x = ((1-g*g))/(1-(2*g*i)+(g*g)**1.5)
    # print(x)
    p.append(x)
```

Plot of Phase Function 1

```
In [271]: plt.xlabel("u")
plt.ylabel(r'$P_n(u)_1$')
plt.plot(u,p, color = 'Blue', label = 'Phase Function 2')
plt.legend()
```

```
Out[271]: <matplotlib.legend.Legend at 0x7fa2b1b7f7d0>
```



Phase Function 2

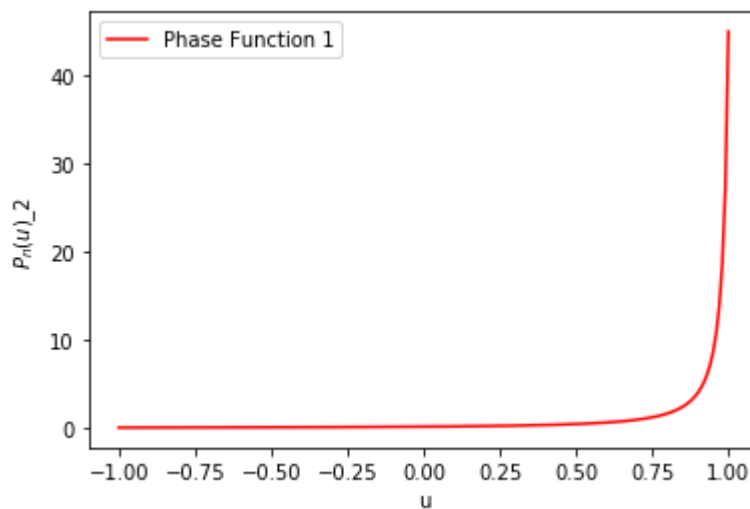
```
In [272]: q=[]
for i in u:
    y=0
    for n in range(0,500):
        y = y + (((2*n)+1)*(g**n)*eval_legendre(n,i))

#     print(y)
q.append(y)
```

Plot of Phase Function 2

```
In [273]: plt.xlabel("u")
plt.ylabel(r'$P_n(u)_2$')
plt.plot(u,q, color = 'Red', label = 'Phase Function 1')
plt.legend()
```

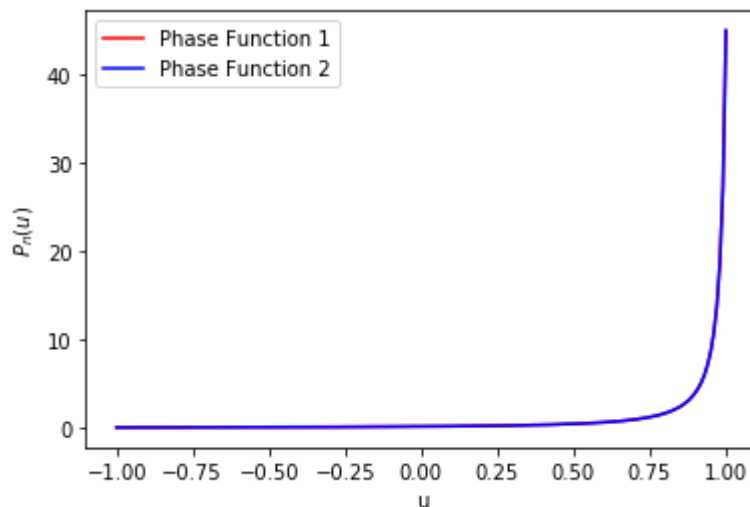
Out[273]: <matplotlib.legend.Legend at 0x7fa2b1c3e990>



Plotting both Phase Function 1&2 in the same graph

```
In [274]: plt.xlabel("u")
plt.ylabel(r'$P_n(u)$')
plt.plot(u,q, color = 'Red', label = 'Phase Function 1')
plt.plot(u,p, color = 'Blue', label = 'Phase Function 2')
plt.legend()
```

Out[274]: <matplotlib.legend.Legend at 0x7fa2b1d58f50>



Conclusion: The results produced by both the phase functions are same as we can see that the graphs are overlay on each other perfectly.

In [ ]: