ECE458_HW4

February 17, 2015

```
In [2]: %pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

```
In [3]: print 20*log10(1.2/10.8)
```

```
-19.0848501888
```

```
In [4]: print 20*log10(3.55/10.8)
```

```
-9.66390804864
```

```
In [5]: print 20*log10(12.1/10.8)
```

```
0.98723229659
```

```
In [35]: x = 1.5
         z = 6.
         r = sqrt((x-2)**2 + z**2)
         #r = 190384
         lam = 1.0/3
         gain = sqrt(lam)/sqrt(r)/(2.*pi*(2.-x)/r)
         #gain = sqrt(lam)/(2.*pi*(2.-x)/r)

         print 20*log10(gain)
```

```
-6.91766994654
```

0.1 Problem 3

Part a

Using two Cornu spiral terms:

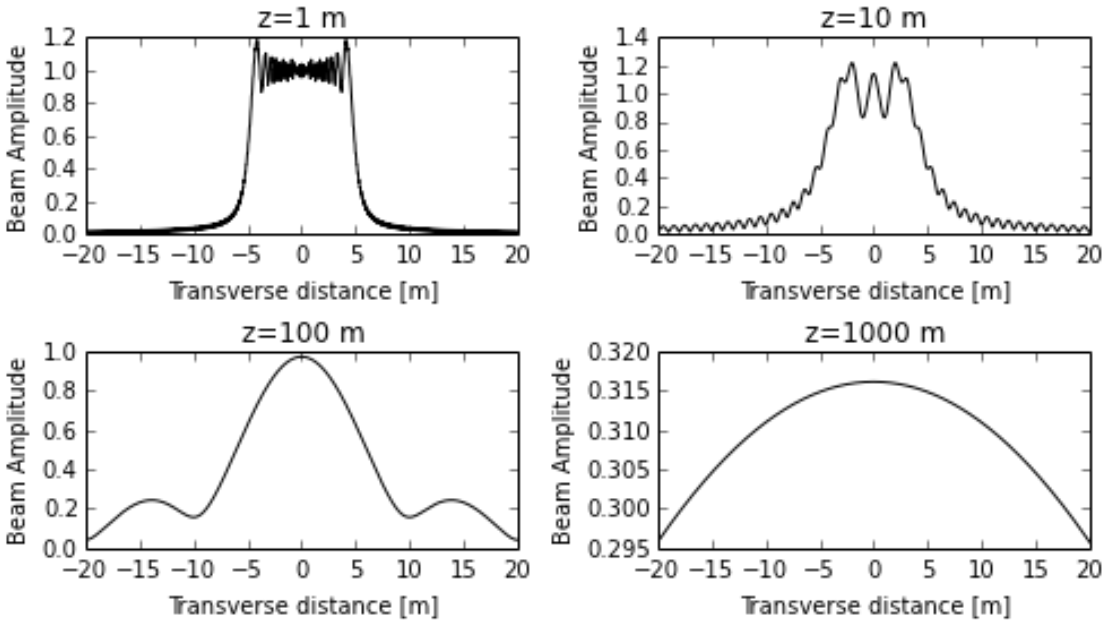
$$|E(x,z)| = E_0 \frac{1}{\sqrt{2}} \left| Fr\left(\sqrt{\frac{2}{\lambda z}}(h_2 - x)\right) - Fr\left(\sqrt{\frac{2}{\lambda z}}(h_1 - x)\right) \right|$$

Take $h_1 = 5\lambda, \quad h_2 = -5\lambda$.

```
In [63]: from scipy.special import fresnel

         def fr(x):
             '''
             Return complex Fresnel function evaluated at x (needs to be array)
             '''
             fsc = fresnel(x) # tuples (real, imag)
             f = zeros(len(fsc[0]), dtype=complex)
```

```
            for i in range(len(f)):
                re = fsc[0][i]
                im = fsc[1][i]
                f[i] = re + 1j*im
            return f

        def plot_beam(z):
            '''
            Plot the beam shape at the given distance z
            '''
            lam = 1.0
            h_1 = -5*lam
            h_2 = 5*lam
            N = 1000
            x = linspace(-20*lam, 20*lam, N)
            Fsc = 1/sqrt(2)*abs(fr(sqrt(2/(lam*z))*(h_1-x)) - fr(sqrt(2/(lam*z))*(h_2-x)))
            plot(x,Fsc,'k-')
            xlabel('Transverse distance [m]')
            ylabel('Beam Amplitude')
```

In [70]:
```
figure(figsize=(7,4))
zvec = [1,10,100,1000]
for i in range(len(zvec)):
    subplot(2,2,i+1)
    plot_beam(zvec[i])
    title('z=%i m' % zvec[i])
tight_layout()
```



In [80]:
```
%matplotlib inline
from IPython.html.widgets import interact, interactive
from IPython.display import clear_output, display, HTML
```

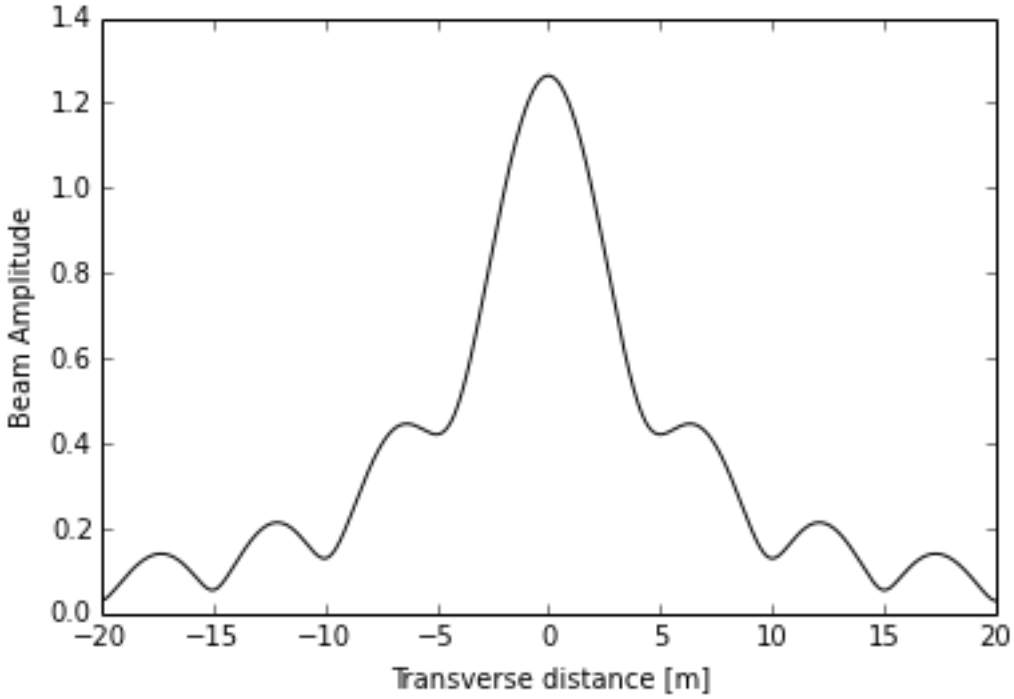
```
import numpy as np
from scipy import integrate

from matplotlib import pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.colors import cnames
from matplotlib import animation
```

Part b

```
In [88]: def interactive_beam(log10z):
             plt.figure()
             plot_beam(10**log10z)
             plt.show()
```

```
In [87]: w = interactive(interactive_beam, log10z=(0.,3.))
         display(w)
```

```
In [94]: interactive_beam(1.7)
```



Based on my qualitative experiments with different values of z, I found that $10^{1.7} = 50$ m seems to be the transition to a far field beam pattern. I based this off of qualitatively deciding when the shape looked more like a gradual falloff than a torch-light.

```
In [89]: rayleigh_distance = 2*(10*lam)**2/lam
         print 'Rayleigh distance is %.1f m' % rayleigh_distance
```

```
Rayleigh distance is 200.0 m
```

My transition distance of 50 m is shorter than the Rayleigh distance of 200 m, but they are on the same order of magnitude. It seems that the concept of Rayleigh distance is applicable in this case.

0.2 Problem 4

In [ ]:

In [ ]:

0.2.1 Convert to pdf or slides

In [95]:
```
import subprocess
import shutil
nb_name = 'ECE458_HW4.ipynb' # There's probably a way to get this programmatically
cmd = ['ipython', 'nbconvert', '--to', 'slides', nb_name, \
       '--reveal-prefix', '"http://cdn.jsdelivr.net/reveal.js/2.6.2"',]
#      '--template','output_toggle']
# The --template command points to a custom template which opens and closes the input code cel
subprocess.call(cmd)
html_name = '%s.slides.html' % (nb_name.split('.')[0])
shutil.copy(html_name, '/home/bhardin2/public_html/slideshows/')
```

In [ ]:
```
nb_name = 'ECE458_HW4.ipynb' # There's probably a way to get this programmatically
cmd = ['ipython', 'nbconvert', '--to', 'pdf', nb_name,]
#      '--template','output_toggle']
# The --template command points to a custom template which opens and closes the input code cell
subprocess.call(cmd)
html_name = '%s.slides.html' % (nb_name.split('.')[0])
shutil.copy(html_name, '/home/bhardin2/public_html/slideshows/')
```

In [ ]:

In [ ]: