Fourier

Dario Mitnik

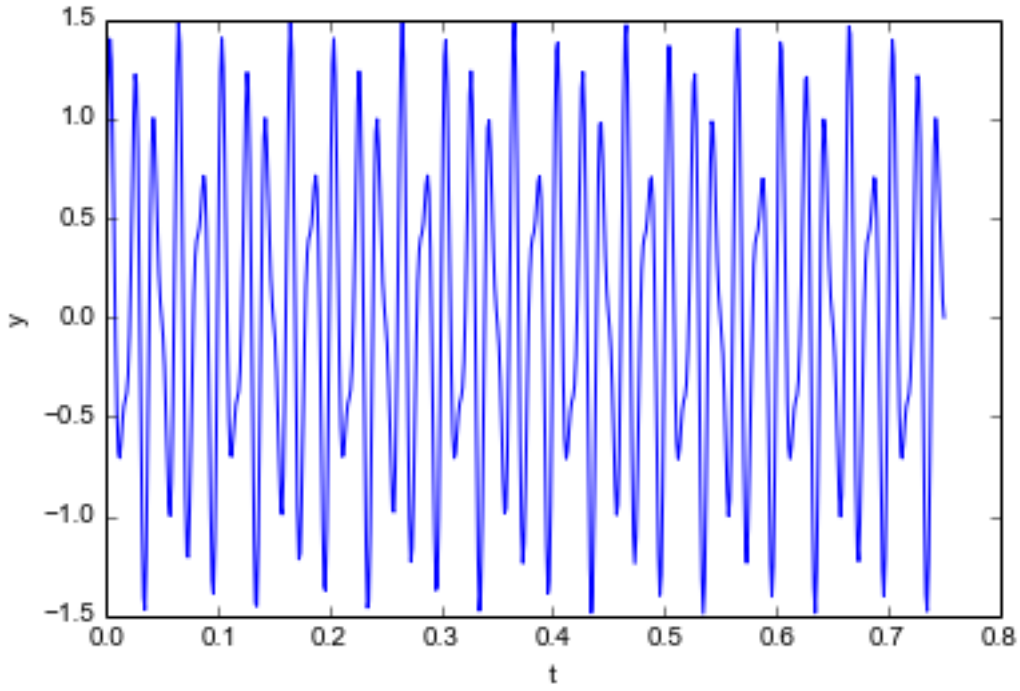
May 20, 2015

Part I

Transformada de Fourier

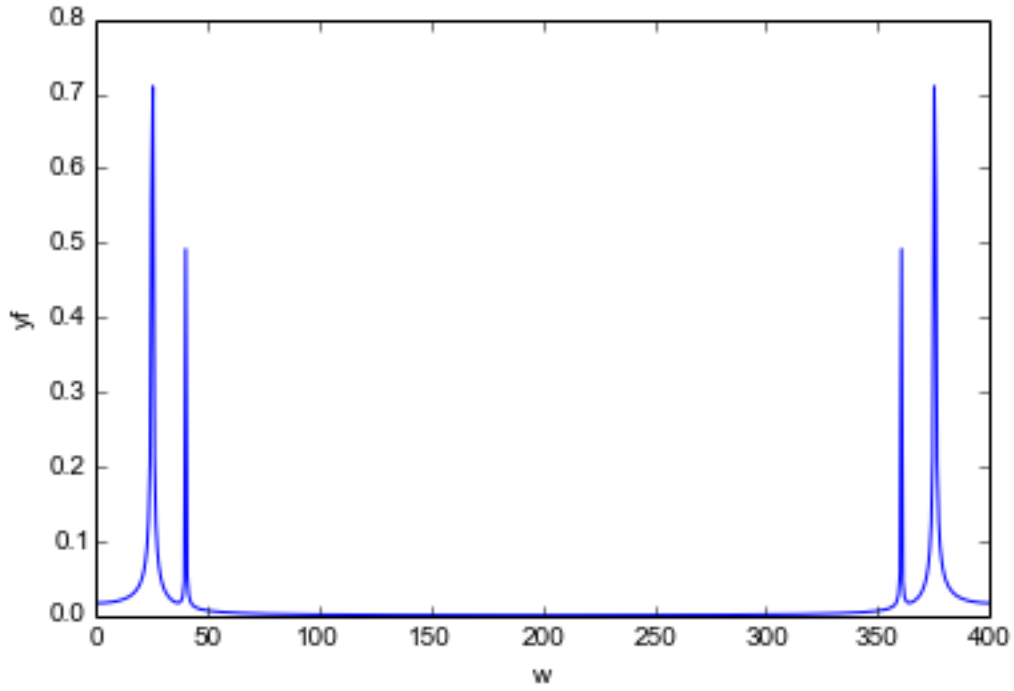
1 Ejemplo Simple

In [47]:
```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import scipy.fftpack

# Number of samplepoints
N = 600
# sample spacing
Dt = 1.0 / 800.0

# Function to transform
t = np.linspace(0.0, N*Dt, N)
y = np.sin(50.0 * 2.0*np.pi*t) + 0.5*np.sin(80.0 * 2.0*np.pi*t)
plt.xlabel("t")
plt.ylabel("y")
plt.plot(t,y);
```



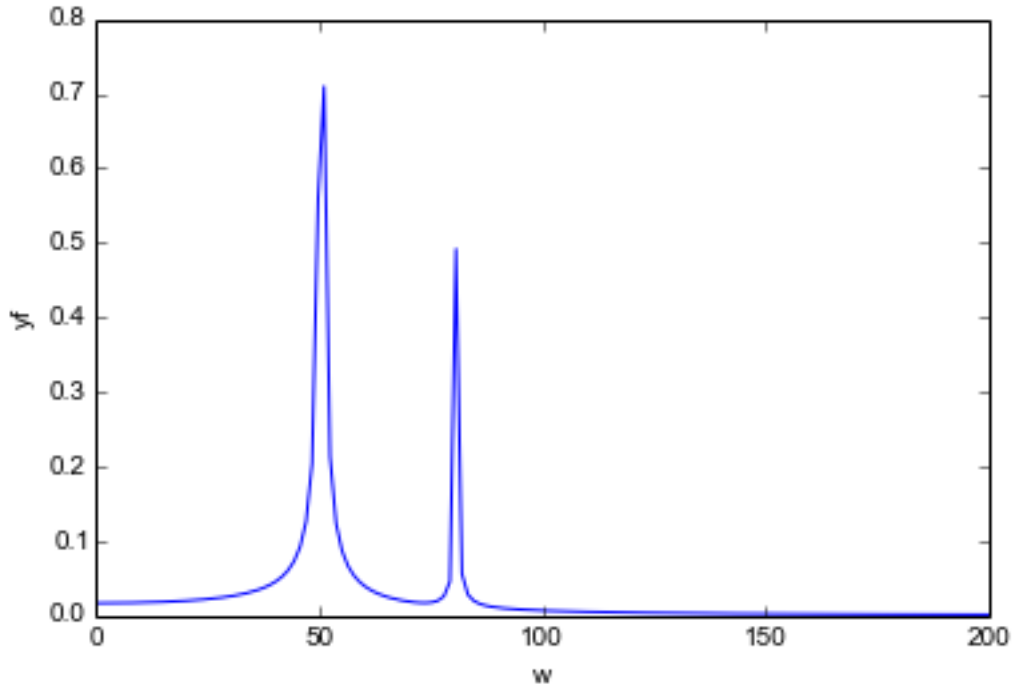
In [48]:

```
# Fourier Transform

yf = scipy.fftpack.fft(y)
wf = np.linspace(0.0, 1.0/(2.0*Dt), N)

# Normalization
wfplot = 2.0/N * abs(yf[0:N])

plt.xlabel("w")
plt.ylabel("yf")
plt.plot(wf,wfplot);
```



In [49]:

```
# without aliasing

yf = scipy.fftpack.fft(y)
#wf = np.linspace(0.0, 1.0/(2.0*Dt), N/2)
wfplot = np.linspace(0.0, 1.0/(4.0*Dt), N/4)

# Normalization
yfplot = 2.0/N * abs(yf[0:N/4])

plt.xlabel("w")
plt.ylabel("yf")
plt.plot(wfplot,yfplot);
```

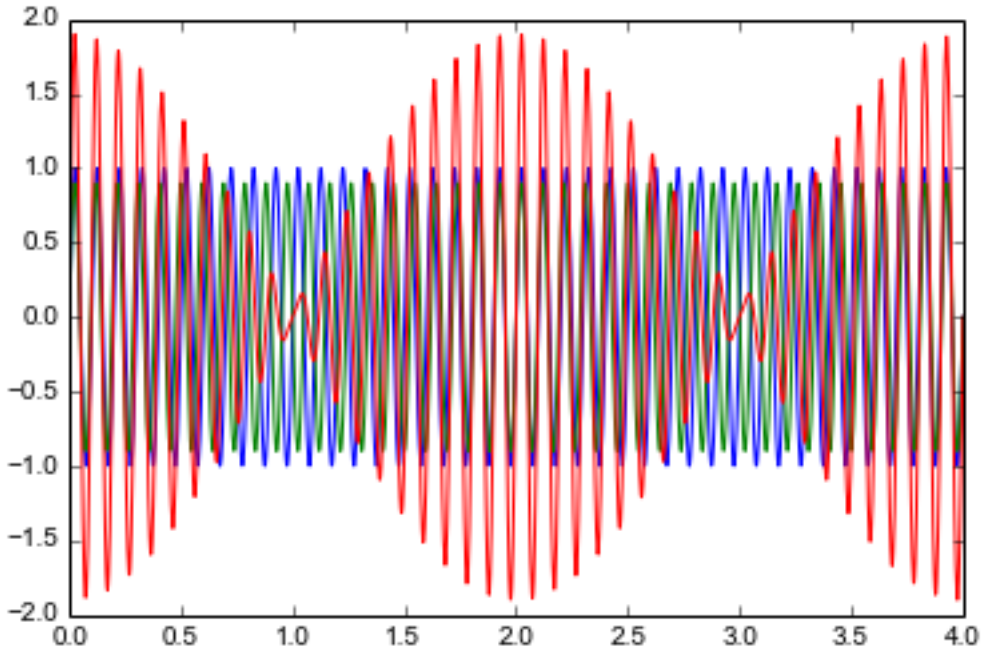


2 Paquete de Ondas

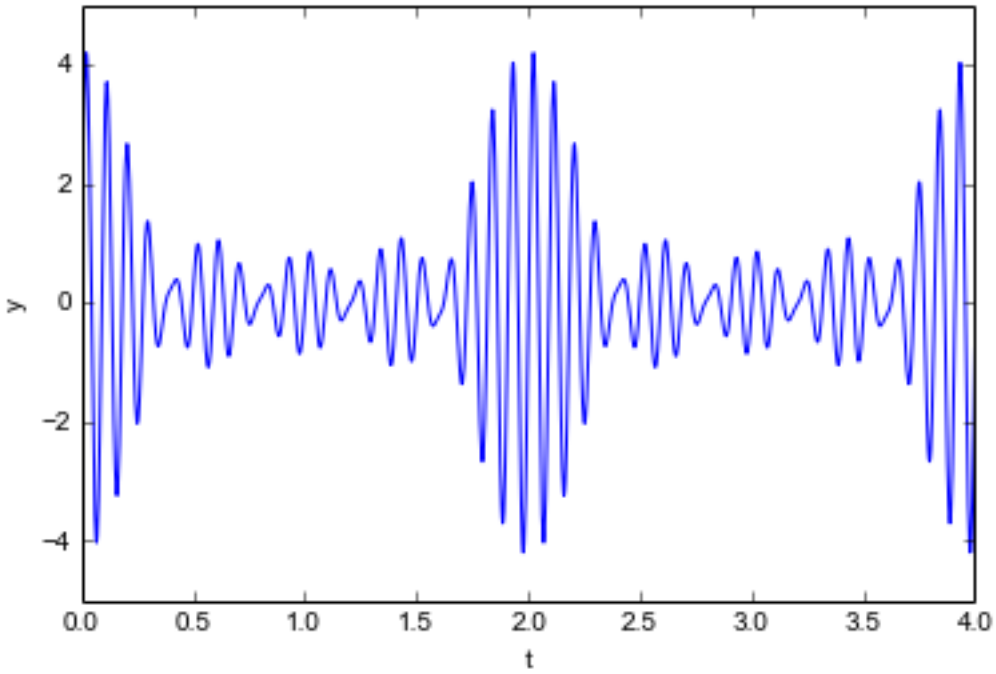
In [258]:
```
# wavepacket
def waven(w0,Dw,An,t):
    arg = (w0+Dw)*t
    return An*np.sin(arg)

def wavepacket(w0,npacket):
     yy = np.zeros(Ntime)
#     for i in range(-npacket,npacket+1):
     for i in range(npacket):
            An = 10.0/(abs(i)+10)
            Dw = i*w0/20.0
            for it in range(Ntime):
                yy[it] =   yy[it] + waven(w0,Dw,An,it*Dt)
     return yy
```

In [297]:
```
# Number of samplepoints
Ntime = 1500

f0 = 10
w0 = 2.0*np.pi*f0
npack = 5
Tau =  1.0 / f0
tfinal = 40*Tau
Dt = tfinal / (Ntime)

# Function to transform
t = np.linspace(0, tfinal, Ntime)

# wavepacket with npack waves, centered at f0
ypack = wavepacket(w0,npack)
```

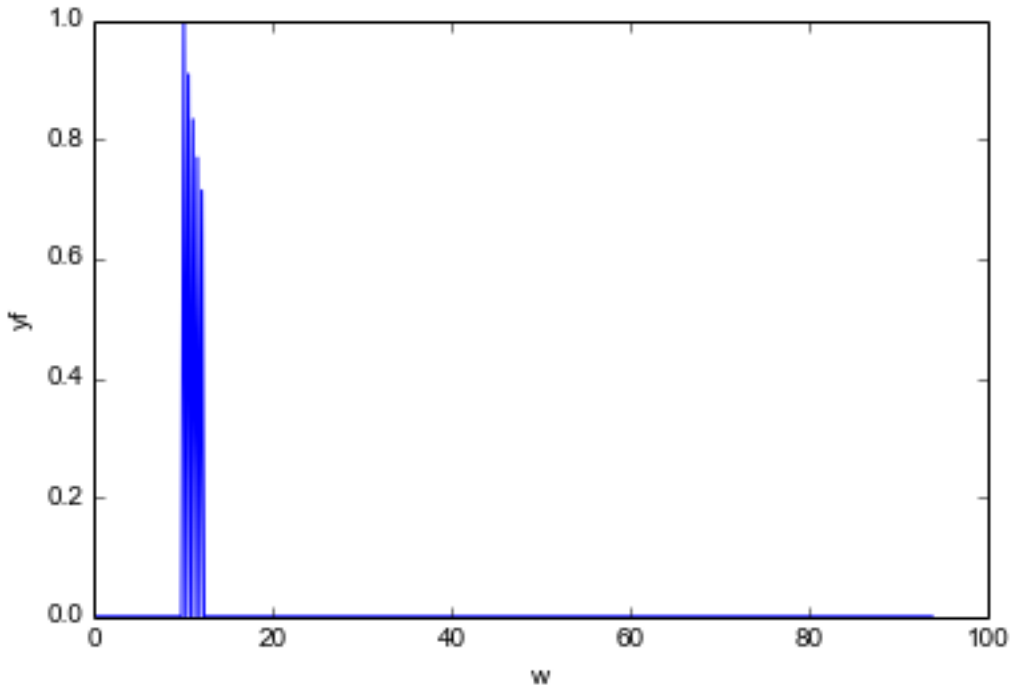
```
plt.plot(t,waven(w0,0,1,t));
plt.plot(t,waven(w0,w0*0.05,0.9,t));
plt.plot(t,waven(w0,0,1,t)+waven(w0,w0*0.05,0.9,t));
```



In [302]:
```
# wavepacket with npacket waves, centered at w0

plt.xlabel("t")
plt.ylabel("y")
plt.plot(t,ypack);
```



In [301]:
```
# Fourier Transform

yfp = scipy.fftpack.fft(ypack)
wfp = np.linspace(0.0, 1/(4.0*Dt), Ntime/4)

# Normalization
yfplot = 2.0/Ntime * abs(yfp[0:Ntime/4])

plt.xlabel("w")
plt.ylabel("yf")
plt.plot(wfp,yfplot);
```

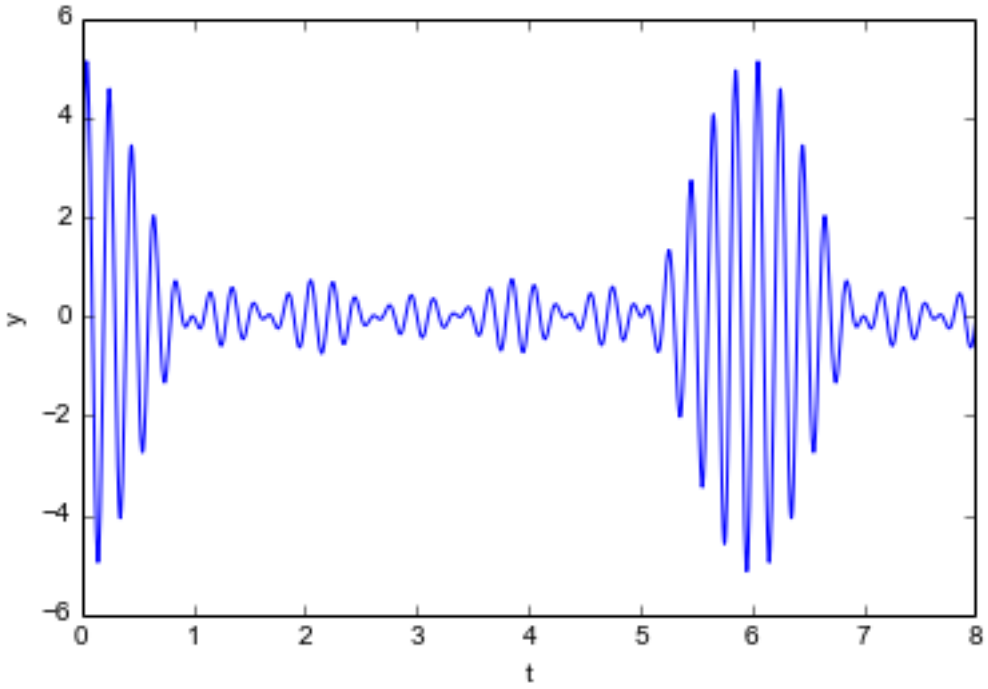


3 symmetric wavepacket

In [364]:
```
def wavepacksim(w0,npacket):
     yy = np.zeros(Ntime)
     for i in range(-npacket,npacket+1):
            An = npacket/(abs(i)*0.7 +  npacket)
            print 'n=',i,'  An=',An
            Dw = i*w0/(npacket*10)
            for it in range(Ntime):
                yy[it] =   yy[it] + waven(w0,Dw,An,it*Dt)
     return yy

# Number of samplepoints
Ntime = 1500

f0 = 5
w0 = 2.0*np.pi*f0
npack = 3
Tau =  1.0 / f0
tfinal = 40*Tau
Dt = tfinal / (Ntime)

# Function to transform
t = np.linspace(0, tfinal, Ntime)
```

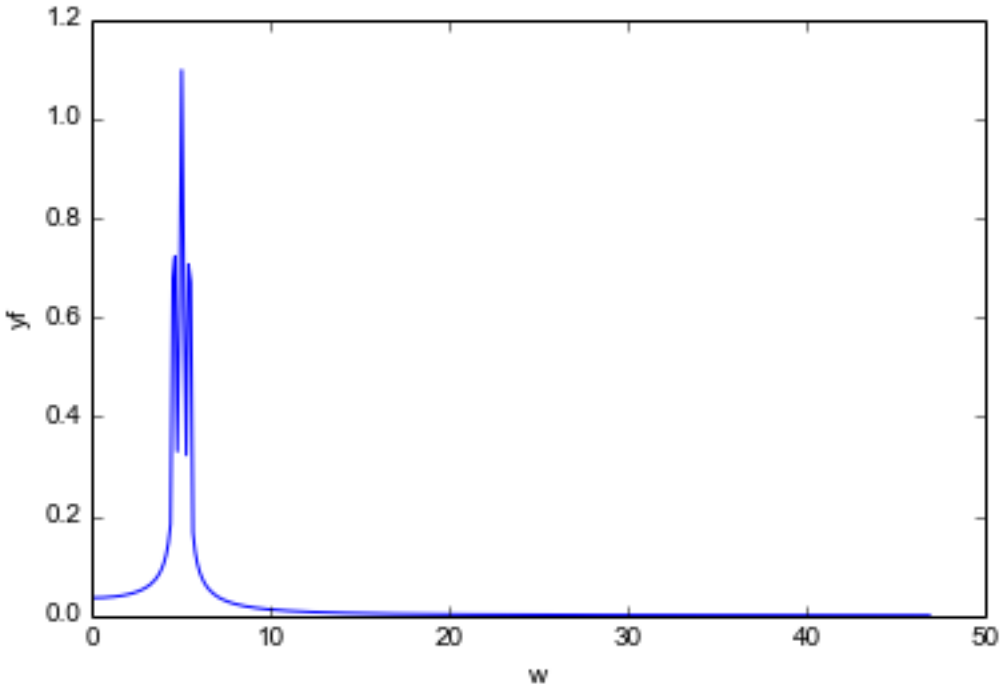
```
# symmetric wavepacket
ypack = wavepacksim(w0,npack)

# wavepacket with npacket waves, centered at w0

plt.xlabel("t")
plt.ylabel("y")
plt.plot(t,ypack);
```

```
n= -3   An= 0.588235294118
n= -2   An= 0.681818181818
n= -1   An= 0.810810810811
n= 0   An= 1.0
n= 1   An= 0.810810810811
n= 2   An= 0.681818181818
n= 3   An= 0.588235294118
```



In [365]:

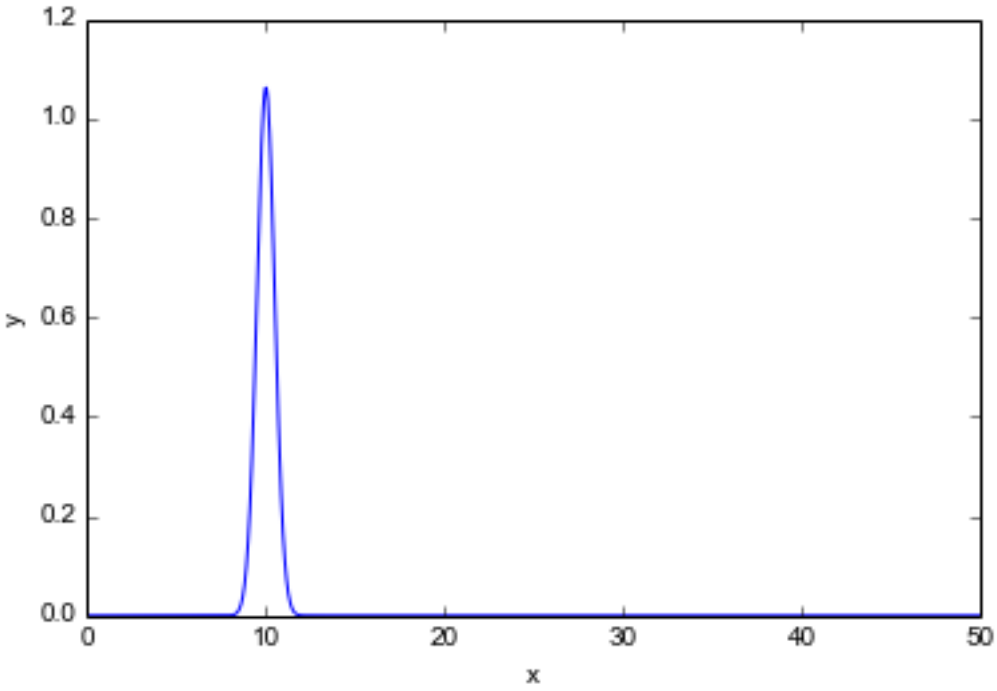
```
# Fourier Transform

yfp = scipy.fftpack.fft(ypack)
wfp = np.linspace(0.0, 1/(4.0*Dt), Ntime/4)

# Normalization
yfplot = 2.0/Ntime * abs(yfp[0:Ntime/4])

plt.xlabel("w")
plt.ylabel("yf")
plt.plot(wfp,yfplot);
```



4 Gaussian Wavepacket

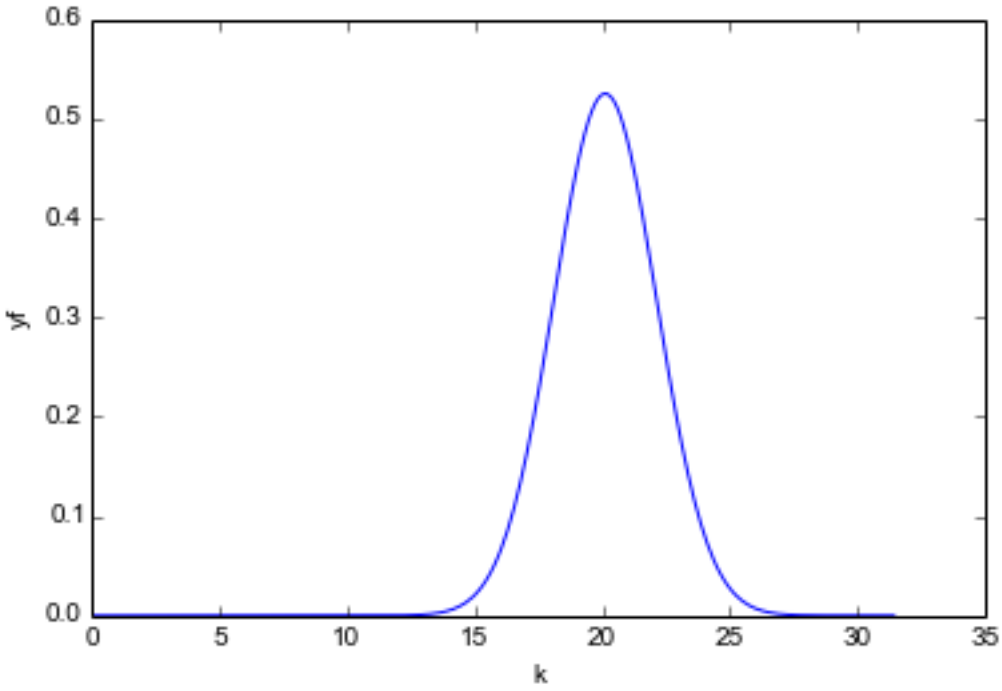
In [366]:
```
# Gaussian Wavepacket centered at x0
def gauss_x(x, a, x0, k0):

# a gaussian wave packet of width a, centered at x0, with momentum k0
    return ((a * np.sqrt(np.pi)) ** (-0.5)
            * np.exp(-0.5 * ((x - x0) * 1. / a) ** 2 + 1j * x * k0))
```

In [427]:
```
# Grid parameters
xmax = 50
nxpts = 1000
Dx = xmax*1.0/nxpts

# parameters for Gaussian wavepacket
x0=10 # position
a=0.5   # width
k0=20 # velocity

x = np.linspace(0, xmax, nxpts)
yg = gauss_x(x,a,x0,k0)

plt.xlabel("x")
plt.ylabel("y")
plt.plot(x,abs(yg));
```

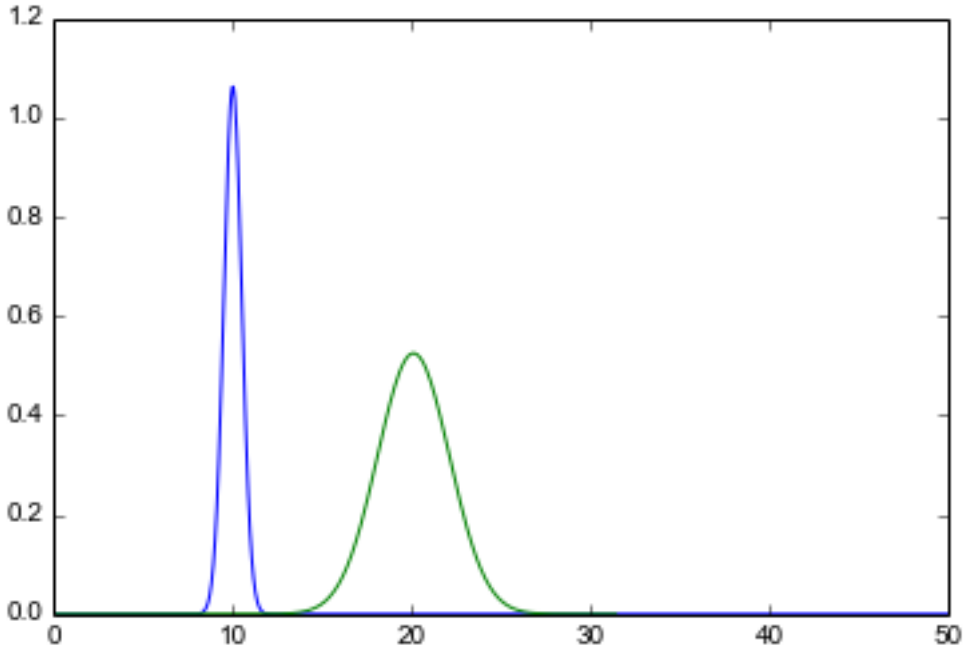


In [428]:

```
# Fourier Transform

yfp = scipy.fftpack.fft(yg)
wfp = np.linspace(0, np.pi/(2*Dx), nxpts/4)

# Normalization
yfplot = 2.0*(np.pi**2)/(nxpts) * abs(yfp[0:nxpts/4])

plt.xlabel("k")
plt.ylabel("yf")
plt.plot(wfp,yfplot);
```



In [429]:
```
plt.plot(x,abs(yg))
plt.plot(wfp,yfplot);
```



In [396]:

In []: