/usr/bin/python

```
import matplotlib
#import pyreport
import numpy as np
from pylab import *
#from pylab import show
from matplotlib import pyplot as pl

x_t,y_t_unsc = np.loadtxt('data/CG10e2t.csv',delimiter=',',unpack=True, usecols = [0,1])
x_x,y_x_unsc = np.loadtxt('data/CG10e2x.csv',delimiter=',',unpack=True, usecols = [0,1])
x_y,y_y_unsc = np.loadtxt('data/CG10e2y.csv',delimiter=',',unpack=True, usecols = [0,1])
x_z,y_z_unsc = np.loadtxt('data/CG10e2z.csv',delimiter=',',unpack=True, usecols = [0,1])
x_w,y_w      = np.loadtxt('data/LO_water.csv',delimiter=',',unpack=True, usecols = [0,1])

#x_t_eV ,y_t_unsc = np.loadtxt('data/CG10e2t.csv',delimiter=',',unpack=True, usecols = [0,1])
#x_x_eV ,y_x_unsc = np.loadtxt('data/CG10e2x.csv',delimiter=',',unpack=True, usecols = [0,1])
#x_y_eV ,y_y_unsc = np.loadtxt('data/CG10e2y.csv',delimiter=',',unpack=True, usecols = [0,1])
#x_z_eV ,y_z_unsc = np.loadtxt('data/CG10e2z.csv',delimiter=',',unpack=True, usecols = [0,1])
#x_w_eV ,y_w      = np.loadtxt('data/LO_water.csv',delimiter=',',unpack=True, usecols = [0,1])

#x_t =x_t_eV/6.582e-16 # convert eV to inverse sec by eV/hbar = ev/(eV*s)=eV/6.6e-16
#x_x =x_x_eV/6.582e-16 # convert eV to inverse sec by eV/hbar = ev/(eV*s)=eV/6.6e-16
#x_y =x_y_eV/6.582e-16 # convert eV to inverse sec by eV/hbar = ev/(eV*s)=eV/6.6e-16
#x_z =x_z_eV/6.582e-16 # convert eV to inverse sec by eV/hbar = ev/(eV*s)=eV/6.6e-16
#x_w =x_w_eV/6.582e-16 # convert eV to inverse sec by eV/hbar = ev/(eV*s)=eV/6.6e-16

y_t = y_t_unsc*4.949
y_x = y_x_unsc*4.949
y_y = y_y_unsc*4.949
y_z = y_z_unsc*4.949
## DEFINE FUNCTIONS FOR CALCULATING e(iz)
#-------------------------------------------------------------
# Matsubara frequencies: z_n at room temp is (2pikbT/hbar)*n (ie coeff*n)
coeff = 0.159 # in eV #(2.41*1e14) # in rad/s

#coeff = 2.41e14 # in (1 rad)*(1/s)=inverse seconds
T = 300.0
#kb_J = 1.3806488e-23 # in J/K
#hbar = 6.625e-34 # in J/s
#coeff_J = 2.0*np.pi*kb_J*T/hbar#1.602e-19*0.159e15 # in eV #(2.41*1e14) # in rad/s
n = arange(0,500)
z = n * coeff
#coeff_J = 1.602e-19*0.159e15 # in eV #(2.41*1e14) # in rad/s
```

```
#z = n * coeff
#z = n * coeff_J

eiz_x = empty(len(z))
eiz_y = empty(len(z))
eiz_z = empty(len(z))
eiz_w = empty(len(z))

eiz_x_arg=empty(len(x_x))
eiz_y_arg=empty(len(x_y))
eiz_z_arg=empty(len(x_z))
eiz_w_arg=empty(len(x_w))

for j in range(len(z)):
    for i in range(len(x_x)):
        eiz_x_arg[i]=x_x[i]*y_x[i] / (x_x[i]**2 + z[j]**2)
    eiz_x[j] = 1 + (2./pi) * trapz(eiz_x_arg,x_x)

    for k in range(len(x_y)):
        eiz_y_arg[k]=x_y[k]*y_y[k] / (x_y[k]**2 + z[j]**2)
    eiz_y[j] = 1 + (2./pi) * trapz(eiz_y_arg,x_y)

    for m in range(len(x_z)):
        eiz_z_arg[m]=x_z[m]*y_z[m] / (x_z[m]**2 + z[j]**2)
    eiz_z[j] = 1 + (2./pi) * trapz(eiz_z_arg,x_z)

    for p in range(len(x_w)):
        eiz_w_arg[p]=x_w[p]*y_w[p] / (x_w[p]**2 + z[j]**2)
    eiz_w[j] = 1 + (2./pi) * trapz(eiz_w_arg,x_w)
#savetxt("data/eiz_x_output.txt", eiz_x)
#savetxt("data/eiz_y_output.txt", eiz_y)
#savetxt("data/eiz_z_output.txt", eiz_z)
#savetxt("data/eiz_w_output.txt", eiz_w)
#
savetxt("data/eiz_x_output_eV.txt", eiz_x)
savetxt("data/eiz_y_output_eV.txt", eiz_y)
savetxt("data/eiz_z_output_eV.txt", eiz_z)
savetxt("data/eiz_w_output_eV.txt", eiz_w)

#pl.figure()
#pl.plot(x_t,y_t,    color = 'k', label = 'total')
#pl.plot(x_x+10,y_x, color = 'b', label = r'$\hat{x}$')
#pl.plot(x_y+20,y_y, color = 'g', label = r'$\hat{y}$')
#pl.plot(x_z+30,y_z, color = 'r', label = r'$\hat{z}$')
#pl.xlabel(r'$\hbar\omega\,\,\,[eV]\,\,\,!shifted\,10\,eV\,for\,visualization$',
    size = 24)
#pl.ylabel(r'$\varepsilon^{\prime\prime}(\omega)$', size = 24)
#pl.legend()
#pl.title(r'CG-10 DNA eps2... shifted for visualization')
##pl.savefig('plots/131010_Hopkins_CG10_eps2_x_z.png', dpi = 300 )
#pl.savefig('plots/131010_Hopkins_CG10_eps2_all_directions.pdf')
##pl.show()
#
pl.figure()
pl.plot(x_t,y_t, color = 'k', label = r'$\varepsilon^{\prime\prime}_{total}(\
    omega)$')
pl.plot(x_x,y_x, color = 'b', label = r'$\varepsilon^{\prime\prime}_\hat{x}(\
    omega)$')
pl.plot(x_y,y_y, color = 'g', label = r'$\varepsilon^{\prime\prime}_\hat{y}(\
    omega)$')
```

```
pl.plot(x_z,y_z, color = 'r', label = r'$\varepsilon^{\prime\prime}_\hat{z}(\omega)$')
pl.plot(x_w,y_w, color = 'c', label = r'$\varepsilon^{\prime\prime}_{H_{2}O}(\omega)$')
pl.xlabel(r'$\hbar\omega\,\,\,[eV]$', size = 24)
pl.ylabel(r'$\varepsilon^{\prime\prime}(\omega)$', size = 24)
pl.legend()
pl.title(r'CG-10 DNA and water eps2')
pl.show()
#pl.close()
#imshow(1)
#pl.savefig('plots/131010_Hopkins_CG10_eps2_x_z.png', dpi = 300 )
#pl.savefig('plots/131010_Hopkins_CG10_eps2_x_z.pdf')
#
##pl.figure()
###pl.plot(x_t,y_t, label = 'total')
###pl.plot(x_x,y_x, label = r'$\hat{x}$')
##pl.plot(x_y,y_y, label = r'$\hat{y}$')
##pl.plot(x_z,y_z, label = r'$\hat{z}$')
##pl.xlabel(r'$\hbar\omega\,\,\,[eV]$', size = 24)
##pl.ylabel(r'$\varepsilon^{\prime\prime}(\omega)$', size = 24)
##pl.legend()
##pl.show()
###pl.close()
###imshow(1)
##pl.savefig('plots/DNA_spectra_x_y.png', dpi = 300 )
#
pl.figure()
pl.plot(n,eiz_x, color = 'b', label = r'$\varepsilon_{\hat{x}}(i\zeta_{N})$')
pl.plot(n,eiz_y, color = 'g', label = r'$\varepsilon_{\hat{y}}(i\zeta_{N})$')
pl.plot(n,eiz_z, color = 'r', label = r'$\varepsilon_{\hat{z}}(i\zeta_{n})$')
pl.plot(n,eiz_w, color = 'c', label = r'$\varepsilon_{\hat{w}}(i\zeta_{n})$')
pl.xlabel(r'$N$', size = 24)
pl.ylabel(r'$\varepsilon(i\zeta)$', size = 24)
pl.legend()
pl.title(r'CG-10 DNA eiz')
#pl.savefig('plots/DNA_eiz_x_z.png', dpi = 300 )
#pl.savefig('plots/131010_Hopkins_CG10_eiz.pdf')
pl.show()
#pl.close()
```