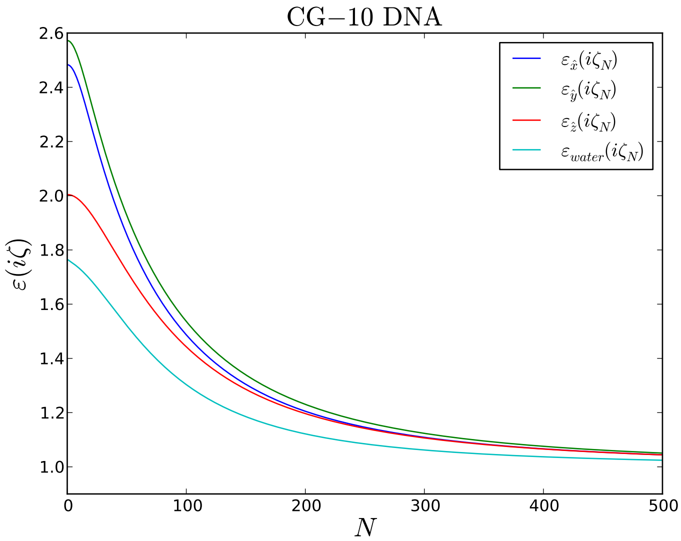
Free energy between two parallel cylinders (CG-10 in water). Nonretarded result, function of separation ↕
Equation 31: $G(\ell,\theta) = -\frac{(\pi R_1^2)(\pi R_2^2)}{2\pi\, \ell^4 \sin\theta}\left(\mathscr{A}^{(0)}(\ell) + \mathscr{A}^{(2)}(\ell)\cos 2\theta\right)$
/usr/bin/python

```
import numpy as np
import scipy.optimize as opt
from scipy.integrate import trapz
import matplotlib.pyplot as pl
import pyreport
from matplotlib import axis as ax
# use pyreport -l file.py
from pylab import show
from matplotlib.ticker import MultipleLocator
from mpl_toolkits.mplot3d import Axes3D
from pylab import pause
from matplotlib.backends.backend_pdf import PdfPages
pp = PdfPages('plots/par_NR_water/par_NR_water.pdf')

eiz_x = np.loadtxt('data/eiz_x_output_eV.txt') #perpendicular, radial

eiz_y = np.loadtxt('data/eiz_y_output_eV.txt')
eiz_z = np.loadtxt('data/eiz_z_output_eV.txt') # parallel,axial

#eiz_w = 1.0 + np.zeros(len(eiz_z))
eiz_w = np.loadtxt('data/eiz_w_output_eV.txt') # water as intervening medium

eiz_w[0] = eiz_w[1] #NOTE: there is a jump from first val down to second val


r_1 = 0.5e-9
r_2 = 0.5e-9
c = 2.99e8 # in m/s

T = 297
kb  = 1.3807e-23 # in J/K

coeff = 2.411e14 # in rad/s

# NOTES:
# h_bar = 1. #1.0546e-34 #in Js
#kb = 8.6173e-5  # in eV/K
# at RT, 1 kT = 4.11e-21 J
# 1 eV = 1.602e-19 J = 0.016 zJ
# h_bar_eV = 6.5821e-16 eVs
# z_n_eV = (2*pi*kT/h_bar)n
#        = (0.159 eV) / (6.5821e-16 eVs)
#        = n*2.411e14 rad/s
# z_n_J = (2*pi*kT/h_bar)n
#        = (1.3807e-23 J/K) / (1.0546e-34 Js))*n
#        = n*2.411e14 rad/s
#coeff = 0.159 # in eV w/o 1/h_bar

ns = np.arange(0.,500.)
z = ns * coeff
ls = np.linspace(1.0e-9, 7.0e-9, 25)

def Aiz(perp, par,med):
        return (2.0*(perp-med)*med)/((perp+med)*(par-med))
def ys(a):
        term1 = 3.0 + 5.*(a + a)
```

```
        return term1
def y_2s(a):
        term1 = (19.*a*a)
        return term1
def As(eizz,eizw,Y):
        term1 = ((eizz-eizw)/eizw)*((eizz-eizw)/eizw)
        term2 = Y
        return term1 * term2
def A_2s(eizz,eizw, Y):
        term1 = ((eizz-eizw)/eizw)*((eizz-eizw)/eizw)
        term2 = Y
        return term1 * term2

y = np.zeros(len(ns))#,len(ls)))

y_2 = np.zeros(len(ns))#,len(ls)))

A   = np.zeros(len(ns))
A_2 = np.zeros(len(ns))
sum_A = np.empty(len(ls))
sum_A_2 = np.empty(len(ls))
EL = np.zeros(len(ls))
G_l_t_dt = np.empty(len(ls))

aiz = []
aiz = Aiz(eiz_x,eiz_z, eiz_w) # of length = len(ns)


for j,n in enumerate(ns):
        y[j]   = ys(aiz[j])
        y_2[j] = y_2s(aiz[j])
        #print 'dt Integral   y = ',i,k,j, y
        #print 'dt Integral y_2 = ',i,k,j, y_2
        #print '----'
        #print 'N terms for A0 = '  , As(eiz_z[j],eiz_w[j],length,n,y)
        #print 'N terms for A2 = ', A_2s(eiz_z[j],eiz_w[j],length,n,y_2)
        #print '----'
        A[j]   = As(eiz_z[j],eiz_w[j],y[j])
        A_2[j] = A_2s(eiz_z[j],eiz_w[j],y_2[j])# * np.cos(2.0*theta)

        A[0] = (1./2)*A[0]
        A_2[0] = (1./2)*A_2[0]
sum_A = np.sum(A,axis=0)
#print 'sum of A0 = ', j,sum_A
sum_A_2 = np.sum(A_2,axis=0)
#print 'sum of A2 = ', j,sum_A_2
#print '----'
for k,length in enumerate(ls):
        EL[k] = 1./(length*length*length*length)
        G_l_t_dt[k] = (1.602e-19 / 4.11e-21) * (9./(32.*64.)) * EL[k]*np.pi*r_1*
            r_1*r_2*r_2*(sum_A + sum_A_2)

pl.figure()
pl.plot(ns,eiz_x, color = 'b', label = r'$\varepsilon_{\hat{x}}(i\zeta_{N})$')
pl.plot(ns,eiz_y, color = 'g', label = r'$\varepsilon_{\hat{y}}(i\zeta_{N})$')
pl.plot(ns,eiz_z, color = 'r', label = r'$\varepsilon_{\hat{z}}(i\zeta_{N})$')
#pl.plot(ns,eiz_w, color = 'c', label = r'$\varepsilon_{vac}(i\zeta_{N})$')
pl.plot(ns,eiz_w, color = 'c', label = r'$\varepsilon_{water}(i\zeta_{N})$')
pl.xlabel(r'$N$', size = 20)
pl.ylabel(r'$\varepsilon(i\zeta)$', size = 20)
```

```
pl.legend(loc = 'best')
pl.title(r'$\mathrm{CG-10\, DNA}$', size = 20)
pl.axis([0,500,0.9,2.6])
pl.savefig('plots/par_NR_water/eiz.pdf' )
show()
```




```

pl.figure()
pl.loglog(ls, G_l_t_dt, label = r'$\mathrm{\mathcal{A}\,=\,}%3.2f$'%(sum_A+
    sum_A_2))
pl.xlabel(r'$Separation,\,\ell\,\,\mathrm{[m]}$', size = 20)
pl.ylabel(r'$\mathrm{-g(\ell;c\rightarrow\infty)\,\,[k_{B}T]}$', size = 20)
#pl.axis([1.5e-9, 6.5e-8,100,145])
pl.title(r'$\mathrm{-g(\ell;c\rightarrow\infty)\,vs.\,separation:\,parallel,\,
    non-ret,\,water}$', size = 20)
pl.legend(loc = 'best')
pl.savefig('plots/par_NR_water/G_vs_l.pdf')
show()
```