Free energy between two skewed cylinders (CG-10 in water). Nonretarded result, function of separation ℓ and angle θ

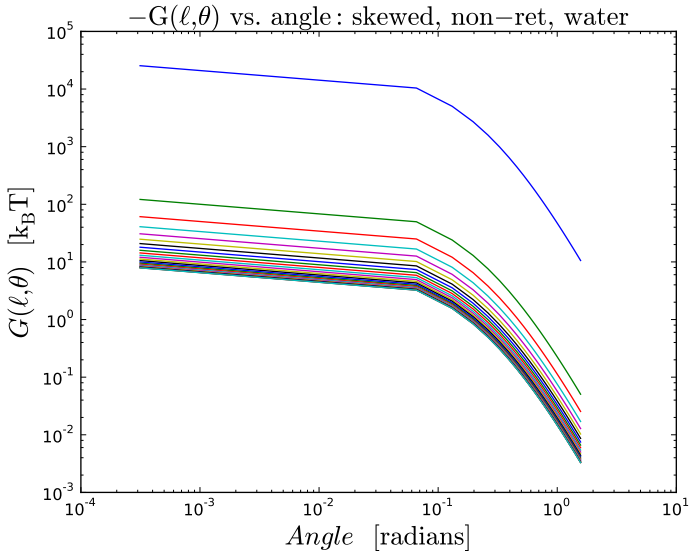
Equation 18: $G(\ell,\theta;c\longrightarrow\infty)=-\frac{k_BT}{64\pi}\frac{\pi^2R_1^2R_2^2}{\ell^4\sin\theta}\sum_{n=0}^{\infty}{}'\Delta_{1,\parallel}\Delta_{2,\parallel}\frac{3}{8}\left[2(1+3a_1)(1+3a_2)+(1-a_1)(1-a_2)\cos 2\theta\right].$

/usr/bin/python

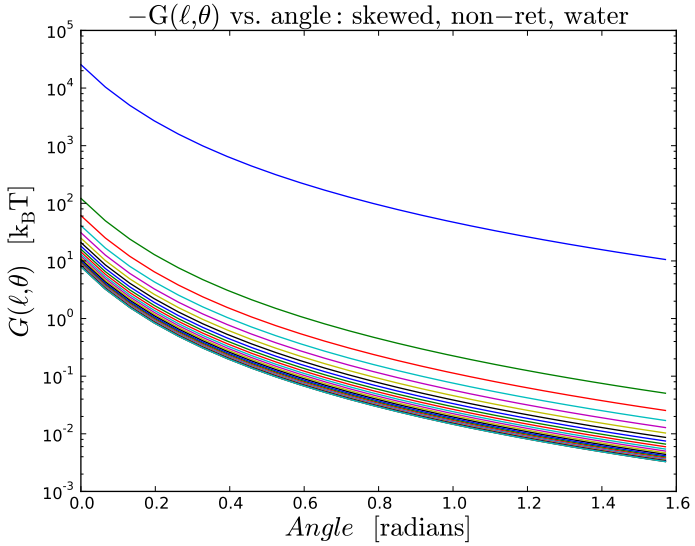
```
import numpy as np
import scipy.optimize as opt
from scipy.integrate import trapz
import matplotlib.pyplot as pl
from matplotlib import axis as ax
# use pyreport -l file.py
from pylab import show
from matplotlib.ticker import MultipleLocator
from mpl_toolkits.mplot3d import Axes3D
from pylab import pause
from matplotlib.backends.backend_pdf import PdfPages
pp = PdfPages('plots/skew_NR_water/skew_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)
thetas = np.linspace((0.0001)*np.pi,(1./2)*np.pi,25)
dt = 1.0
ts = np.arange(1.0,10000.,dt)

def Aiz(perp, par,med):
        return (2.0*(perp-med)*med)/((perp+med)*(par-med))
```

```
62 def ys(a):
63         term1 = (2.0*(1. + 3.*a)*(1.+3.*a))
64         return term1
65 def y_2s(a):
66         term1 = ((1.- a)*(1.- a))
67         return term1
68 def As(eizz,eizw,Y):
69         term1 = 3./8*((eizz-eizw)/eizw)*((eizz-eizw)/eizw)
70         term2 = Y
71         return term1 * term2
72 def A_2s(eizz,eizw,Y):
73         term1 = 3./8*((eizz-eizw)/eizw)*((eizz-eizw)/eizw)
74         term2 = Y
75         return term1 * term2
76
77 y = np.zeros(len(ns))#,len(ls)))
78
79 y_2 = np.zeros(len(ns))#,len(ls)))
80
81 A   = np.zeros(len(ns))#,len(ls)))
82
83 A_2 = np.zeros(len(ns))#,len(ls)))
84
85 EL = np.zeros(len(ls))
86 G_l_t_dt = np.zeros(shape=(len(ls),len(thetas)))
87
88 aiz = []
89 aiz = Aiz(eiz_x,eiz_z, eiz_w) # of length = len(ns)
90
91
92 sum_A = np.empty(len(ls))
93 sum_A_2 = np.empty(len(ls))
94
95 for j,n in enumerate(ns):
96         # Integral:
97         y[j]   = ys(aiz[j])
98         y_2[j] = y_2s(aiz[j])
99         #print 'dt Integral   y = ',i,k,j, y
100        #print 'dt Integral y_2 = ',i,k,j, y_2
101        #print '----'
102        #print 'N terms for A0 = '  , As(eiz_z[j],eiz_w[j],length,n,y)
103        #print 'N terms for A2 = ', A_2s(eiz_z[j],eiz_w[j],length,n,y_2)
104        #print '----'
105        A[j]   = As(eiz_z[j],eiz_w[j],y[j])
106        A_2[j] = A_2s(eiz_z[j],eiz_w[j],y_2[j])# * np.cos(2.0*theta)
107
108 sum_A = np.sum(A,axis=0)
109 #print 'sum of A0 = ', j,sum_A
110 sum_A_2 = np.sum(A_2,axis=0)
111 #print 'sum of A2 = ', j,sum_A_2
112 #print '----'
113 #print 'shape sum_A_2 = ', np.shape(sum_A_2)
114 #sys.exit()
115 for k,length in enumerate(ls):
116         for i, theta in enumerate(thetas):
117                 EL[k] = 1./(length*length*length*length)
118                 G_l_t_dt[k,i] = (1.602e-19 / 4.11e-21) * (1./32) * EL[k]*np.pi*
                        r_1*r_1*r_2*r_2*(sum_A + sum_A_2* np.cos(2.0*theta) )/(2.0*np
                        .sin(theta))# (1e21)*
119
```

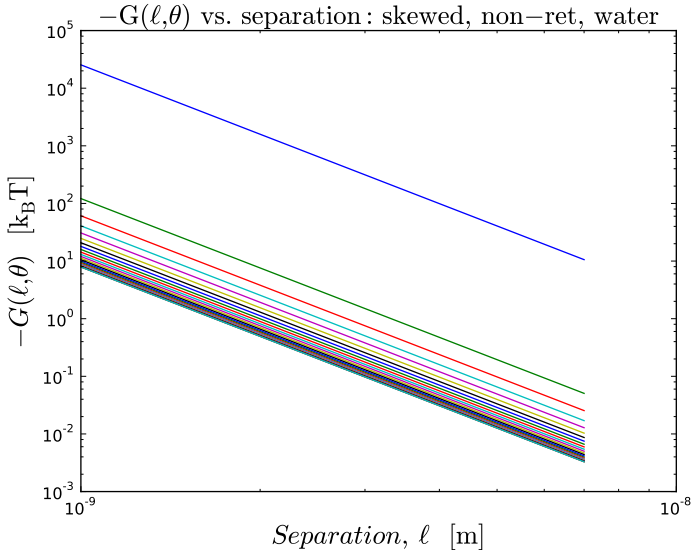
```
120
121 pl.figure()
122 pl.plot(ns,eiz_x, color = 'b', label = r'$\varepsilon_{\hat{x}}(i\zeta_{N})$')
123 pl.plot(ns,eiz_y, color = 'g', label = r'$\varepsilon_{\hat{y}}(i\zeta_{N})$')
124 pl.plot(ns,eiz_z, color = 'r', label = r'$\varepsilon_{\hat{z}}(i\zeta_{N})$')
125 #pl.plot(ns,eiz_w, color = 'c', label = r'$\varepsilon_{vac}(i\zeta_{N})$')
126 pl.plot(ns,eiz_w, color = 'c', label = r'$\varepsilon_{water}(i\zeta_{N})$')
127 pl.xlabel(r'$N$', size = 20)
128 pl.ylabel(r'$\varepsilon(i\zeta)$', size = 20)
129 pl.legend(loc = 'best')
130 pl.title(r'$\mathrm{CG-10\, DNA}$', size = 20)
131 pl.axis([0,500,0.9,2.6])
132 pl.savefig('plots/skew_NR_water/eiz.pdf')
133 pl.show()
134
135 pl.figure()
136 pl.loglog(thetas, G_l_t_dt)#, label = labels_l[k])
137
138 pl.xlabel(r'$Angle\,\,\mathrm{[radians]}$', size = 20)
139 pl.ylabel(r'$G(\ell,\theta)\,\,\mathrm{[k_{B}T]}$', size = 20)
140 #pl.axis([(1./25)*np.pi,(3./4)*np.pi,105,135])
141 pl.title(r'$\mathrm{-G(\ell,\theta)\,vs.\,angle:\,skewed,\,non-ret,\,water}$',
    size = 20)
142 #pl.legend(loc = 'best')
143 pl.savefig('plots/skew_NR_water/G_vs_theta.pdf')
144 show()
```




```
131
132 pl.figure()
133 pl.semilogy(thetas, G_l_t_dt)
134 pl.xlabel(r'$Angle\,\,\mathrm{[radians]}$', size = 20)
135 pl.ylabel(r'$G(\ell,\theta)\,\,\mathrm{[k_{B}T]}$', size = 20)
136 #pl.axis([(1./25)*np.pi,(3./4)*np.pi,105,135])
137 pl.title(r'$\mathrm{-G(\ell,\theta)\,vs.\,angle:\,skewed,\,non-ret,\,water}$',
    size = 20)
138 #pl.legend(loc = 'best')
139 pl.savefig('plots/skew_NR_water/semilog_G_vs_theta.pdf')
140 show()
```

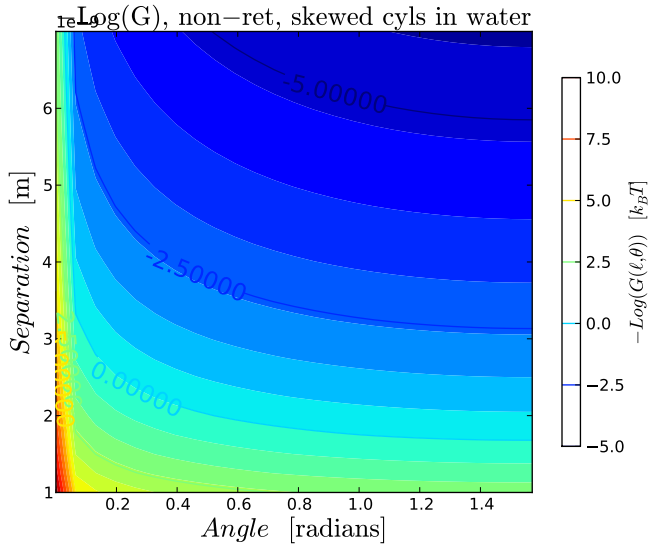



```
141
142 pl.figure()
143 pl.loglog(ls, G_l_t_dt)#, label = labels[i])
144
145 pl.xlabel(r'$Separation ,\ ,\ell\ ,\ ,\mathrm{[m]}$', size = 20)
146 pl.ylabel(r'$-G(\ell ,\theta)\ ,\ ,\mathrm{[k_{B}T]}$', size = 20)
147 #pl.axis([1.5e-9, 6.5e-8,100,145])
148 pl.title(r'$\mathrm{-G(\ell ,\theta)\ ,vs.\ ,separation:\ ,skewed ,\ ,non-ret ,\ ,water}
    $', size = 20)
149 #pl.legend(loc = 'best')
150 pl.savefig('plots/skew_NR_water/G_vs_l.pdf')
151 show()
```



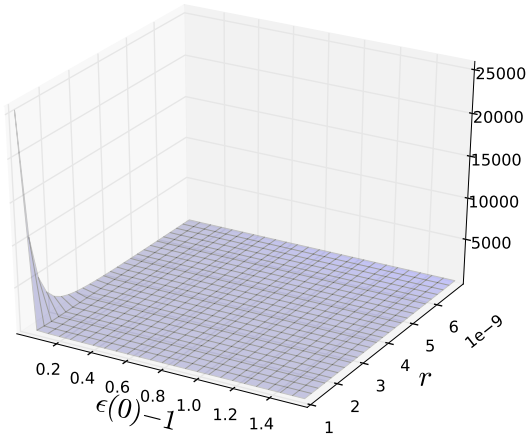

```
150
151 #G_l_t_dt[G_l_t_dt>300]= np.nan #NOTE: remove me later
152 #G_l_t_dt[G_l_t_dt<200e-25]= np.nan #NOTE: remove me later
153
154 # CONTOUR PLOT:
155 X,Y = np.meshgrid(thetas, ls)
156 pl.figure()
157 pl.contourf(X, Y, np.log(G_l_t_dt), 25)#, cmap = cm.hot())
158
159 CS = pl.contour(X,Y,np.log(G_l_t_dt))#, levels = np.linspace(1e-1,1e10,10))
```

```
160
161 pl.clabel(CS, inline =1,fmt = '%1.5f', fontsize = 18,color = 'k')#, manual =
        man_loc)
162
163 pl.xlabel(r'$Angle\,\,\mathrm{[radians]}$', size = 20)
164 pl.ylabel(r'$Separation\,\,\mathrm{[m]}$', size = 20)
165 pl.title(r'$\mathrm{-Log(G)\,,non-ret\,,skewed\,cyls\,in\,water}$', size = 20)#
        uas a function of separation and angle')
166
167 cbar = pl.colorbar(CS, shrink = 0.8, extend = 'both')
168 cbar.ax.set_ylabel(r'$-Log(G(\mathcal{\ell},\theta))\,\,[k_{B}T]$', size = 14)
169 cbar.add_lines(CS)
170 ##pl.axis([0,1.0,0,1.0])
171 #pl.grid()
172 pl.savefig('plots/skew_NR_water/logG_contour.pdf')
173 show()
```




```
171
172 fig = pl.figure()
173 ax = fig.gca(projection = '3d')
174 #ax.text(-7, 6, 0.7, r'$\zeta/\omega_{0}$', zdir = (-1,1,-3), size = 21)
175 surf = ax.plot_surface(X,Y, G_l_t_dt, rstride = 1, cstride = 1,alpha = 0.2,
        linewidth = 0.3)#edgecolor = 'none',antialiased = True, shade = False, norm =
         norm, linewidth = 0.3)
176
177 #surf = ax.plot_surface(X,Y, G_l_t_dt, rstride = 20, cstride = 20,alpha = 0.2)#,
         cmap = cm.gnuplot, linewidth = 0.5)#gray)#coolwarm)#bone)#hot, linewidth =
        0.01, antialiased = True, shade = False)# True)#, cmap = hot()
178 #colorbar(surf)
179 #cbar.ax.set_ylabel(r'$\frac{\xi}{\omega_{0}}$', size = 24)
180 #cset = ax.contour(X,Y,h, zdir = 'z', offset = 0, cmap = cm.jet)
181 #cset = ax.contour(X,Y,h, zdir = 'x', offset = 5, cmap = cm.jet)
182 #cset = ax.contourf(X,Y,h, zdir = 'y', offset = 6, cmap = cm.jet)# puts plot of
        max xi vs discrete r values at r=0 plane
183 #ax.view_init(elev = 19, azim = -112)
184 #zlabel(r'$\xi/\omega_{0}$', size = 21)
185 #ylabel(r'$r$', size = 24)
186 #xlabel(r'$(\epsilon(0) -1)$', size = 24)
187 #text = Axes.text(self, x, y, s, **kwargs)
188 #art3d.text_2d_to_3d(text, z, zdir)
189 #return text
190 #pl.text(6,0, 0, r'$\xi/\omega_{0}$',size = 21 ,rotation = 'horizontal')
```

```
191 #ax.text(r'$\xi/\omega_{0}$',6,0, 0, size = 21 ,rotation = 'horizontal')
192 #ax.set_zlabel(r'$\xi/\omega_{0}$',size = 21 ,rotation = 'horizontal' )
193 ax.set_xlabel(r'$\epsilon(0)-1$', size = 21)
194 ax.set_ylabel(r'$r$', size = 22)
195 show()
```




```
193 #pp.savefig()
```