050-test-randomized-object

March 23, 2017

In [1]:
```
from ScatterSim.NanoObjects import RandomizedNanoObject, PolydisperseNanoObject, SphereN
from ScatterSim.CompositeNanoObjects import OctahedronCylindersNanoObject
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline
```

In [2]:
```
radius = 1
sigma_radius = .04
pargs_sphere = {
    'radius' : radius,
}
pargs_polysphere = {
    'radius' : radius,
    'sigma_radius' : sigma_radius

}
argdict_sphere = {
        'radius' : {'distribution_type' : 'gaussian', 'mean' : radius, 'sigma' : sigma_r
}
# try lognormal
argdict_sphere_lognormal = {
        'radius' : {'distribution_type' : 'lognormal', 'mean' : np.log(radius), 'sigma'
}
```

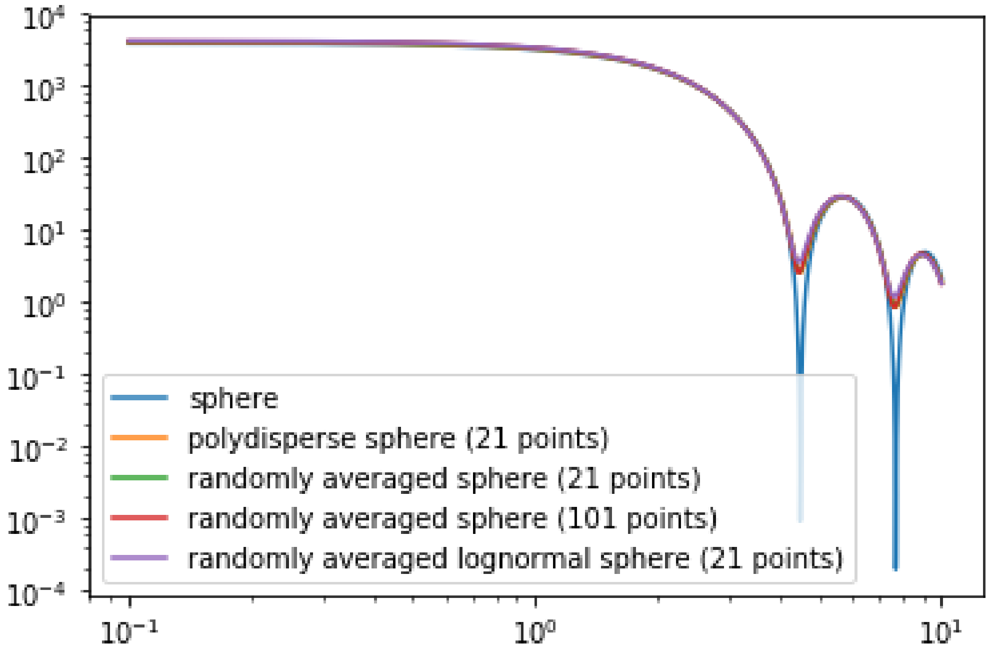
In [3]:
```
sphere = SphereNanoObject(pargs_sphere)
polysphere = PolydisperseNanoObject(SphereNanoObject, pargs_polysphere, argname='radius'
randsphere = RandomizedNanoObject(SphereNanoObject, pargs_sphere, argdict_sphere)
pargs_sphere_101 = pargs_sphere.copy()
# try more points
pargs_sphere_101['distribution_num_points'] = 101
randsphere_101 = RandomizedNanoObject(SphereNanoObject, pargs_sphere_101, argdict_sphere

# lognormal
randsphere_lognormal = RandomizedNanoObject(SphereNanoObject, pargs_sphere, argdict_sphe
```

In [4]:
```
q = np.linspace(.1, 10, 1000)
sqsphere = sphere.form_factor_squared_isotropic(q)
```

```
         sqpolysphere = polysphere.form_factor_squared_isotropic(q)
         sqrandsphere = randsphere.form_factor_squared_isotropic(q)
         sqrandsphere_101 = randsphere_101.form_factor_squared_isotropic(q)
         sqrandsphere_lognormal = randsphere_lognormal.form_factor_squared_isotropic(q)
```

```
In [5]: # they roughly all agree
         plt.figure(0);plt.clf();
         plt.loglog(q,sqsphere,label="sphere")
         plt.loglog(q,sqpolysphere, label="polydisperse sphere (21 points)")
         plt.loglog(q, sqrandsphere, label="randomly averaged sphere (21 points)")
         plt.loglog(q, sqrandsphere_101, label="randomly averaged sphere (101 points)")
         plt.loglog(q, sqrandsphere_lognormal, label="randomly averaged lognormal sphere (21 poin
         plt.legend()
```

```
Out[5]: <matplotlib.legend.Legend at 0x7f331a005f28>
```



```
In [ ]:
```