Modeling the parameters for Flexure Correction

1) you start out with a bunch of model parameters, p0

2) the model, flexure_correction3(), returns a set of x,y offsets

3) you compare those offsets with measured offsets, using res()

4) you modify the model parameters so that the model-data is minimized

p1 = leastsq(res,p0,args=(data,angs,err**(-2)),full_output=1)

5) the new parameters, p1, become the new model

In [1]:
```
import numpy as np
```

In [7]:
```
from scipy.optimize import leastsq
```

In [19]:
```
dg=np.pi/180.0
```

In [20]:
```
p0 = [0.161, 9.60, 66.5, 5.33,6.89,0.626] # HK CSU zen fit (3/23/12)
```

In [10]:
```
def res(p,data,t,wt=None):
        '''Returns residuals to flexure model for fitting.'''
        mod = []
        for ii in range(len(t)):
                mod.append(flexure_correction3(t[ii],p,filter=filter_set))
        mod = -np.array(mod) # - because Fx is correction (x + Fx = 0)
        err = (data - mod).flatten()
        wterr = err*(wt.flatten())
        return wterr
```

In [11]:
```
def flexure_correction3(angs,params, filter=""):
        '''Returns flexure correction in pixels for a given Elevation, PA, and
        vector of flexure ellipse parameters.'''
        El=angs[1]*dg
        PA=angs[0]*dg

        k=params[0] # ratio of minor to major axes of flexure ellipse
        ph=params[1]*dg # phase of ellipse
        th=params[2]*dg # rotation of major axis w.r.t. y axis
        amp=params[3]*np.sin(El-90*dg) # semi-major axis length
        Yc=params[4]*(1-np.cos(El-90*dg)) # center of ellipse in Y
        Xc=params[5]*(1-np.cos(El-90*dg)) # center of ellipse in Y
        af = 1

        Fy0=Yc+amp*np.cos(PA+th)*np.cos(ph)+ \
            amp*k*np.sin(PA+th)*np.sin(ph)
        Fx0=Xc+amp*np.cos(PA+th)*np.sin(ph)- \
            af*amp*k*np.sin(PA+th)*np.cos(ph)

        return [Fx0,Fy0]
```

In [ ]:

```
# here is the code to get the parameters
# p0 is the stating parameters
# data is the measured x,y offsets
# angs is the PA and elevation
# err is some weights
p1 = leastsq(res,p0,args=(data,angs,err**(-2)),full_output=1)
```