MTF_Interp_Polar_Plots

September 23, 2015

In [1]:
```
import pandas as pd
import pickle
import seaborn as snb
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
```

In [2]:
```
polar_halfs_dat = pd.read_pickle('MTFHalfPolar.p')
interp_halfs_dat = pd.read_pickle('MTFHalfInterp.p')
```

In [3]:
```
grouped_polar = polar_halfs_dat.groupby(['Soffset','Doffset'])

polar_halfs = {}
for a,b in grouped_polar:
    group = b[['xslice','yslice','zslice','xslice_radius','yslice_radius','zslice_radius'\
                ,'xslice_slope','yslice_slope','zslice_slope','mag','Soffset','Doffset']]
    polar_halfs[a] = {'means':group.groupby("mag").mean(),
                 'vars':group.groupby("mag").var(ddof=1),
                'stds':group.groupby("mag").std(ddof=1)}

grouped_interp = interp_halfs_dat.groupby(['S','D'])

interp_halfs = {}
for a,b in grouped_interp:
    group = b[['X1','X2','X3','X4','Y1','Y2','Y3','Y4','Z1','Z2','Z3','Z4','mag','D','S']]
    interp_halfs[a] = {'means':group.groupby("mag").mean(),
                 'vars':group.groupby("mag").var(ddof=1),
                'stds':group.groupby("mag").std(ddof=1)}
```

In [4]:
```
S1D1 = interp_halfs[(1,1)]['means']
avs_S1D1 = pd.DataFrame(columns=['X','Y','Z'])
avs_S1D1.X = S1D1[['X1','X2','X3','X4']].transpose().mean()
avs_S1D1.Y = S1D1[['Y1','Y2','Y3','Y4']].transpose().mean()
avs_S1D1.Z = S1D1[['Z1','Z2','Z3','Z4']].transpose().mean()

S1D1 = interp_halfs[(1,1)]['stds']
stds_S1D1 = pd.DataFrame(columns=['X','Y','Z'])
stds_S1D1.X = S1D1[['X1','X2','X3','X4']].transpose().mean()
stds_S1D1.Y = S1D1[['Y1','Y2','Y3','Y4']].transpose().mean()
stds_S1D1.Z = S1D1[['Z1','Z2','Z3','Z4']].transpose().mean()


S0D1 = interp_halfs[(0,1)]['means']
avs_S0D1 = pd.DataFrame(columns=['X','Y','Z'])
```

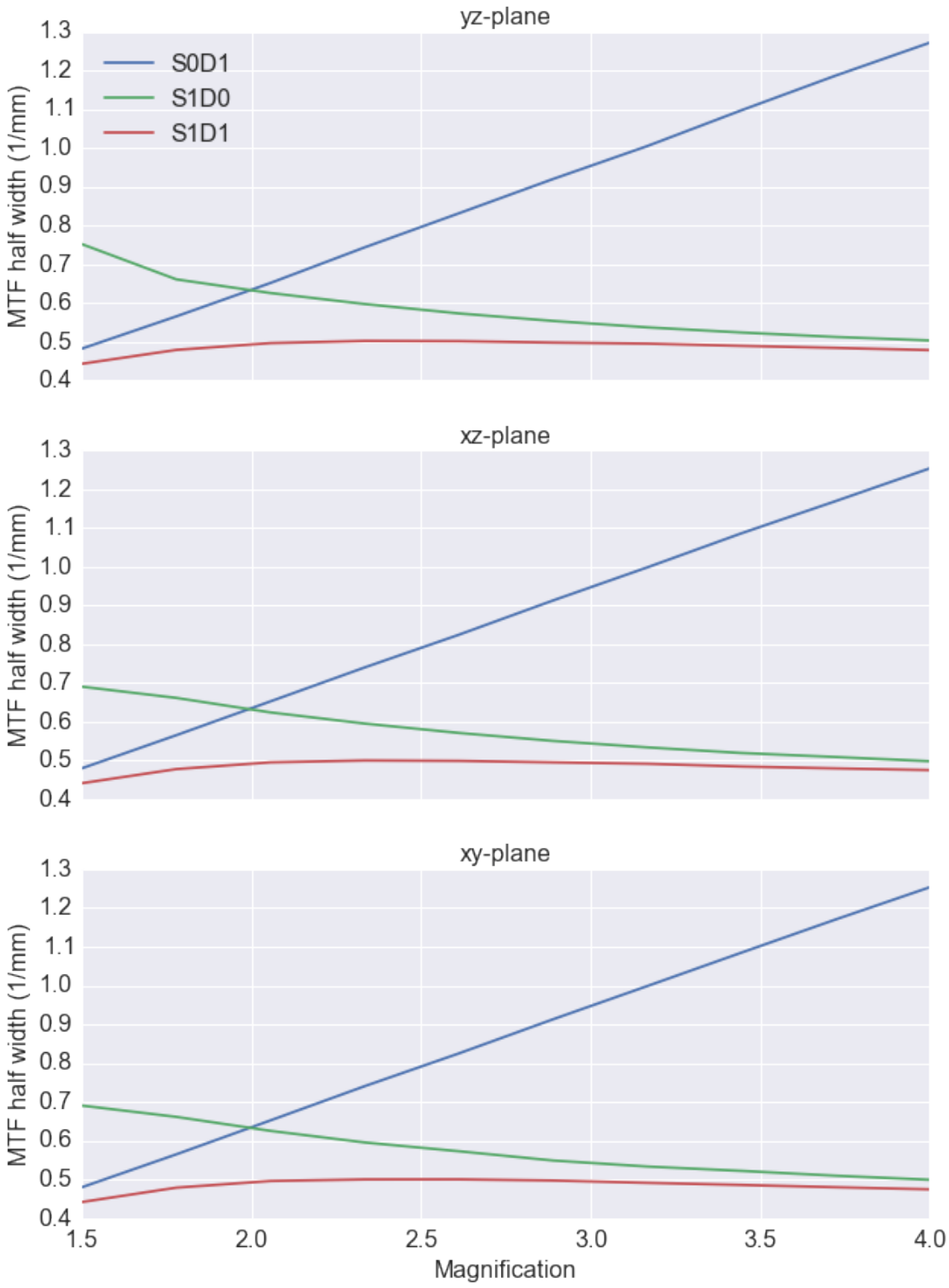
```
avs_S0D1.X = S0D1[['X1','X2','X3','X4']].transpose().mean()
avs_S0D1.Y = S0D1[['Y1','Y2','Y3','Y4']].transpose().mean()
avs_S0D1.Z = S0D1[['Z1','Z2','Z3','Z4']].transpose().mean()

S0D1 = interp_halfs[(0,1)]['stds']
stds_S0D1 = pd.DataFrame(columns=['X','Y','Z'])
stds_S0D1.X = S0D1[['X1','X2','X3','X4']].transpose().mean()
stds_S0D1.Y = S0D1[['Y1','Y2','Y3','Y4']].transpose().mean()
stds_S0D1.Z = S0D1[['Z1','Z2','Z3','Z4']].transpose().mean()

S1D0 = interp_halfs[(1,0)]['means']
avs_S1D0 = pd.DataFrame(columns=['X','Y','Z'])
avs_S1D0.X = S1D0[['X1','X2','X3','X4']].transpose().mean()
avs_S1D0.Y = S1D0[['Y1','Y2','Y3','Y4']].transpose().mean()
avs_S1D0.Z = S1D0[['Z1','Z2','Z3','Z4']].transpose().mean()

S1D0 = interp_halfs[(1,0)]['stds']
stds_S1D0 = pd.DataFrame(columns=['X','Y','Z'])
stds_S1D0.X = S1D0[['X1','X2','X3','X4']].transpose().mean()
stds_S1D0.Y = S1D0[['Y1','Y2','Y3','Y4']].transpose().mean()
stds_S1D0.Z = S1D0[['Z1','Z2','Z3','Z4']].transpose().mean()

S0D0 = interp_halfs[(0,0)]['means']
avs_S0D0 = pd.DataFrame(columns=['X','Y','Z'])
avs_S0D0.X = S0D0[['X1','X2','X3','X4']].transpose().mean()
avs_S0D0.Y = S0D0[['Y1','Y2','Y3','Y4']].transpose().mean()
avs_S0D0.Z = S0D0[['Z1','Z2','Z3','Z4']].transpose().mean()

S0D0 = interp_halfs[(0,0)]['stds']
stds_S0D0 = pd.DataFrame(columns=['X','Y','Z'])
stds_S0D0.X = S0D0[['X1','X2','X3','X4']].transpose().mean()
stds_S0D0.Y = S0D0[['Y1','Y2','Y3','Y4']].transpose().mean()
stds_S0D0.Z = S0D0[['Z1','Z2','Z3','Z4']].transpose().mean()
```

In [23]:
```
from scipy.optimize import curve_fit
mags = polar_halfs[(0,1)]['stds'].index

def recip(x,A,B):
    return A/x + B

def lin(x,A,B):
    return A*x + B

def reciplin(x,A,B,C):
    return A/x + B*x + C

popt1_e, pcov = curve_fit(recip, mags.data.tolist(), polar_halfs[(1,0)]['means'].\
                          xslice.as_matrix())
popt2_e, pcov = curve_fit(recip, mags.data.tolist(), polar_halfs[(1,0)]['means'].\
                          yslice.as_matrix())
popt3_e, pcov = curve_fit(recip, mags.data.tolist(), polar_halfs[(1,0)]['means'].\
                          zslice.as_matrix())

popt4_e, pcov = curve_fit(recip, mags.data.tolist(), avs_S1D0.X.as_matrix())
```

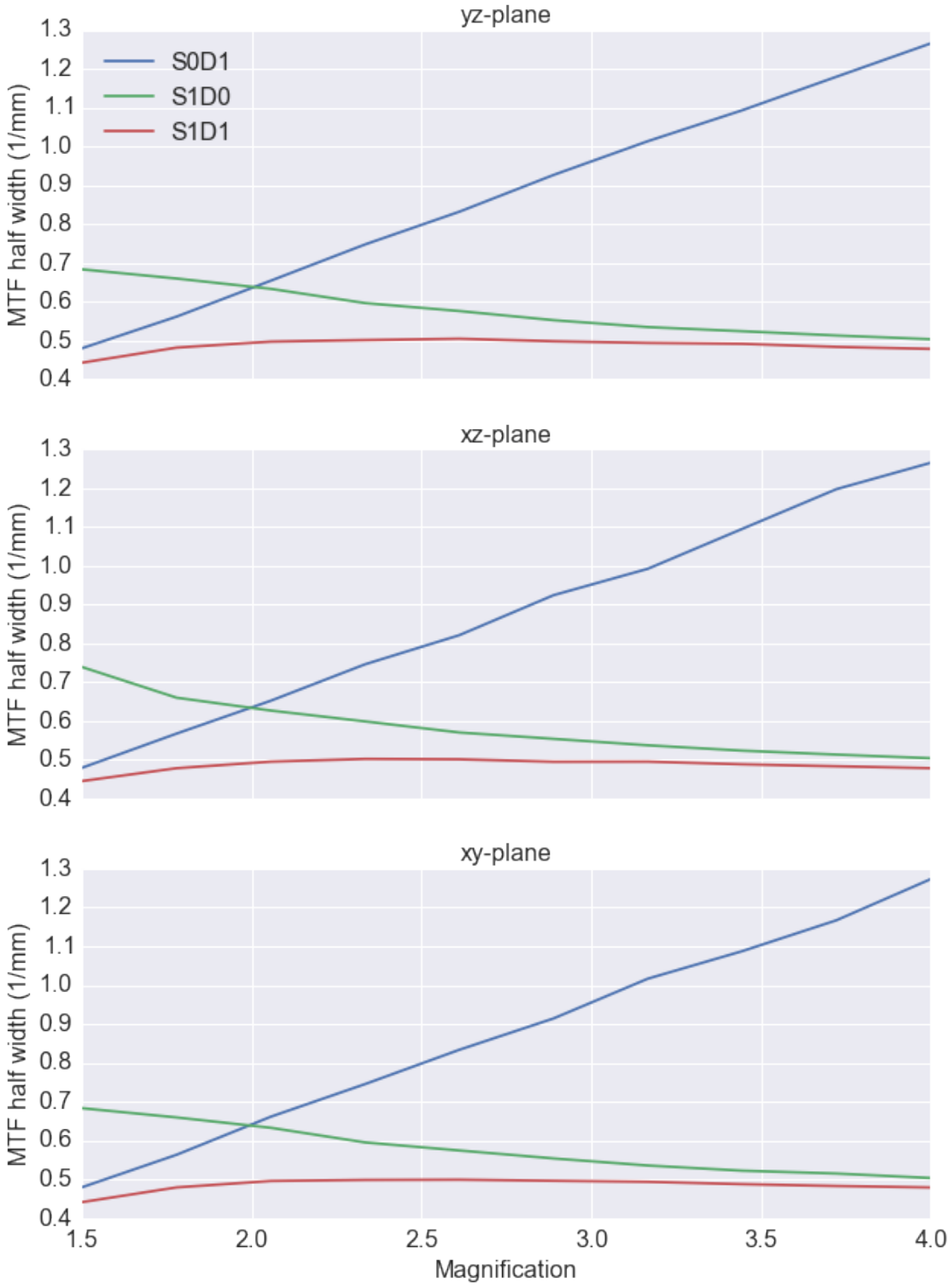
```
popt5_e, pcov = curve_fit(recip, mags.data.tolist(), avs_S1D0.Y.as_matrix())
popt6_e, pcov = curve_fit(recip, mags.data.tolist(), avs_S1D0.Z.as_matrix())

rmse_px10 = np.sqrt(np.mean((recip(np.array(mags.data.tolist()),*popt1_e)- \
                            polar_halfs[(1,0)]['means'].xslice.as_matrix())**2))
rmse_py10 = np.sqrt(np.mean((recip(np.array(mags.data.tolist()),*popt2_e)- \
                            polar_halfs[(1,0)]['means'].yslice.as_matrix())**2))
rmse_pz10 = np.sqrt(np.mean((recip(np.array(mags.data.tolist()),*popt3_e)- \
                            polar_halfs[(1,0)]['means'].zslice.as_matrix())**2))

rmse_ix10 = np.sqrt(np.mean((recip(np.array(mags.data.tolist()),*popt4_e)-avs_S1D0.\
                            X.as_matrix())**2))
rmse_iy10 = np.sqrt(np.mean((recip(np.array(mags.data.tolist()),*popt5_e)-avs_S1D0.\
                            Y.as_matrix())**2))
rmse_iz10 = np.sqrt(np.mean((recip(np.array(mags.data.tolist()),*popt6_e)-avs_S1D0.\
                            Z.as_matrix())**2))

popt1_l, pcov = curve_fit(lin, mags.data.tolist(), polar_halfs[(0,1)]['means'].\
                          xslice.as_matrix())
popt2_l, pcov = curve_fit(lin, mags.data.tolist(), polar_halfs[(0,1)]['means'].\
                          yslice.as_matrix())
popt3_l, pcov = curve_fit(lin, mags.data.tolist(), polar_halfs[(0,1)]['means'].\
                          zslice.as_matrix())

popt4_l, pcov = curve_fit(lin, mags.data.tolist(), avs_S0D1.X.as_matrix())
popt5_l, pcov = curve_fit(lin, mags.data.tolist(), avs_S0D1.Y.as_matrix())
popt6_l, pcov = curve_fit(lin, mags.data.tolist(), avs_S0D1.Z.as_matrix())

rmse_px01 = np.sqrt(np.mean((lin(np.array(mags.data.tolist()),*popt1_l)- \
                            polar_halfs[(0,1)]['means'].xslice.as_matrix())**2))
rmse_py01 = np.sqrt(np.mean((lin(np.array(mags.data.tolist()),*popt2_l)- \
                            polar_halfs[(0,1)]['means'].yslice.as_matrix())**2))
rmse_pz01 = np.sqrt(np.mean((lin(np.array(mags.data.tolist()),*popt3_l)- \
                            polar_halfs[(0,1)]['means'].zslice.as_matrix())**2))

rmse_ix01 = np.sqrt(np.mean((lin(np.array(mags.data.tolist()),*popt4_l)-avs_S0D1.\
                            X.as_matrix())**2))
rmse_iy01 = np.sqrt(np.mean((lin(np.array(mags.data.tolist()),*popt5_l)-avs_S0D1.\
                            Y.as_matrix())**2))
rmse_iz01 = np.sqrt(np.mean((lin(np.array(mags.data.tolist()),*popt6_l)-avs_S0D1.\
                            Z.as_matrix())**2))

popt1, pcov = curve_fit(reciplin, mags.data.tolist(), polar_halfs[(1,1)]['means'].\
                        xslice.as_matrix())
popt2, pcov = curve_fit(reciplin, mags.data.tolist(), polar_halfs[(1,1)]['means'].\
                        yslice.as_matrix())
popt3, pcov = curve_fit(reciplin, mags.data.tolist(), polar_halfs[(1,1)]['means'].\
                        zslice.as_matrix())

popt4, pcov = curve_fit(reciplin, mags.data.tolist(), avs_S1D1.X.as_matrix())
popt5, pcov = curve_fit(reciplin, mags.data.tolist(), avs_S1D1.Y.as_matrix())
popt6, pcov = curve_fit(reciplin, mags.data.tolist(), avs_S1D1.Z.as_matrix())

rmse_px11 = np.sqrt(np.mean((reciplin(np.array(mags.data.tolist()),*popt1)- \
```

```
                                polar_halfs[(1,1)]['means'].xslice.as_matrix())**2))
        rmse_py11 = np.sqrt(np.mean((reciplin(np.array(mags.data.tolist()),*popt2)- \
                                polar_halfs[(1,1)]['means'].yslice.as_matrix())**2))
        rmse_pz11 = np.sqrt(np.mean((reciplin(np.array(mags.data.tolist()),*popt3)- \
                                polar_halfs[(1,1)]['means'].zslice.as_matrix())**2))

        rmse_ix11 = np.sqrt(np.mean((reciplin(np.array(mags.data.tolist()),*popt4)-avs_S1D1.\
                                X.as_matrix())**2))
        rmse_iy11 = np.sqrt(np.mean((reciplin(np.array(mags.data.tolist()),*popt5)-avs_S1D1.\
                                Y.as_matrix())**2))
        rmse_iz11 = np.sqrt(np.mean((reciplin(np.array(mags.data.tolist()),*popt6)-avs_S1D1.\
                                Z.as_matrix())**2))
```

In [40]:
```
fig,(ax1,ax2,ax3) = plt.subplots(ncols=1,nrows=3,figsize=(10,14),sharex=True)

fig.suptitle('Polar method - MTF halfwidth as magnification increases',fontsize=16)

polar_halfs[(0,1)]['means'].xslice.plot(ax=ax1,label='S0D1',fontsize=16)
polar_halfs[(1,0)]['means'].xslice.plot(ax=ax1,label='S1D0',fontsize=16)
polar_halfs[(1,1)]['means'].xslice.plot(ax=ax1,label='S1D1',fontsize=16)
ax1.legend(loc=0,fontsize=16)
ax1.set_title("yz-plane",fontsize = 16)
ax1.set_ylabel('MTF half width (1/mm)',fontsize = 16)
ax1.set_xlabel('',fontsize = 16)
ax1.tick_params(axis='both', which='major', labelsize=16)


polar_halfs[(0,1)]['means'].yslice.plot(ax=ax2,label='S0D1',fontsize=16)
polar_halfs[(1,0)]['means'].yslice.plot(ax=ax2,label='S1D0',fontsize=16)
polar_halfs[(1,1)]['means'].yslice.plot(ax=ax2,label='S1D1',fontsize=16)
ax2.set_title("xz-plane",fontsize = 16)
ax2.set_ylabel('MTF half width (1/mm)',fontsize = 16)
ax2.set_xlabel('',fontsize = 16)
ax2.tick_params(axis='both', which='major', labelsize=16)

polar_halfs[(0,1)]['means'].zslice.plot(ax=ax3,label='S0D1',fontsize=16)
polar_halfs[(1,0)]['means'].zslice.plot(ax=ax3,label='S1D0',fontsize=16)
polar_halfs[(1,1)]['means'].zslice.plot(ax=ax3,label='S1D1',fontsize=16)
ax3.set_title("xy-plane",fontsize = 16)
ax3.set_xlabel('Magnification',fontsize = 16)
ax3.set_ylabel('MTF half width (1/mm)',fontsize = 16)
ax3.tick_params(axis='both', which='major', labelsize=16)
```

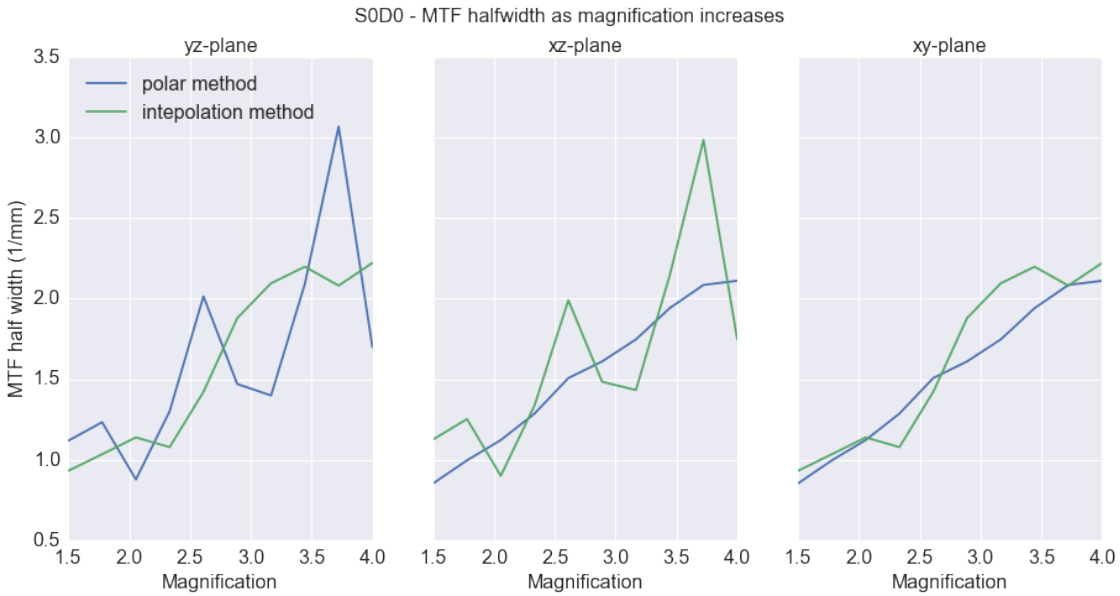

Polar method - MTF halfwidth as magnification increases
yz-plane
xz-plane
xy-plane
S0D1
S1D0
S1D1
MTF half width (1/mm)
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
1.5
2.0
2.5
3.0
3.5
4.0
Magnification

In [41]:
```
fig,(ax1,ax2,ax3) = plt.subplots(ncols=1,nrows=3,figsize=(10,14),sharex=True)

fig.suptitle('Interpolation method - MTF halfwidth as magnification increases',fontsize=16)

avs_S0D1.X.plot(ax=ax1,label='S0D1',fontsize=16)
avs_S1D0.X.plot(ax=ax1,label='S1D0',fontsize=16)
avs_S1D1.X.plot(ax=ax1,label='S1D1',fontsize=16)
ax1.legend(loc=0,fontsize=16)
ax1.set_title("yz-plane",fontsize = 16)
ax1.set_ylabel('MTF half width (1/mm)',fontsize = 16)
ax1.set_xlabel('',fontsize = 16)
ax1.tick_params(axis='both', which='major', labelsize=16)


avs_S0D1.Y.plot(ax=ax2,label='S0D1',fontsize=16)
avs_S1D0.Y.plot(ax=ax2,label='S1D0',fontsize=16)
avs_S1D1.Y.plot(ax=ax2,label='S1D1',fontsize=16)
ax2.set_title("xz-plane",fontsize = 16)
ax2.set_ylabel('MTF half width (1/mm)',fontsize = 16)
ax2.set_xlabel('',fontsize = 16)
ax2.tick_params(axis='both', which='major', labelsize=16)

avs_S0D1.Z.plot(ax=ax3,label='S0D1',fontsize=16)
avs_S1D0.Z.plot(ax=ax3,label='S1D0',fontsize=16)
avs_S1D1.Z.plot(ax=ax3,label='S1D1',fontsize=16)
ax3.set_title("xy-plane",fontsize = 16)
ax3.set_ylabel('MTF half width (1/mm)',fontsize = 16)
ax3.set_xlabel('Magnification',fontsize = 16)
ax3.tick_params(axis='both', which='major', labelsize=16)
```


Interpolation method - MTF halfwidth as magnification increases
yz-plane
xz-plane
xy-plane
S0D1
S1D0
S1D1
MTF half width (1/mm)
Magnification
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
1.5
2.0
2.5
3.0
3.5
4.0

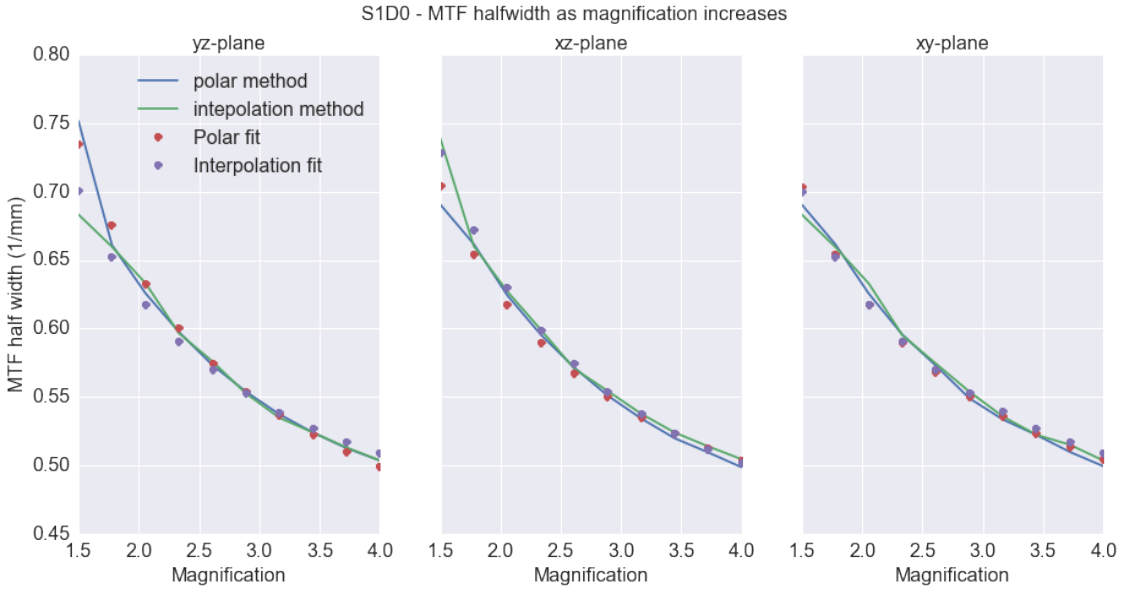
In [24]:
```
fig,(ax1,ax2,ax3) = plt.subplots(ncols=3,nrows=1,figsize = (15,7),sharey=True)

fig.suptitle('S0D0 - MTF halfwidth as magnification increases',fontsize=16)

polar_halfs[(0,0)]['means'].xslice.plot(ax=ax1,label='polar method',fontsize=16)
avs_S0D0.X.plot(ax=ax1,label = 'intepolation method',fontsize=16)
ax1.legend(loc=0,fontsize=16)
ax1.set_ylabel('MTF half width (1/mm)',fontsize = 16)
ax1.set_xlabel('Magnification',fontsize = 16)
ax1.set_title("yz-plane",fontsize = 16)
ax1.tick_params(axis='both', which='major', labelsize=16)

polar_halfs[(0,0)]['means'].yslice.plot(ax=ax2,label='polar method',fontsize=16)
avs_S0D0.Y.plot(ax=ax2,label = 'intepolation method',fontsize=16)
ax2.set_title("xz-plane",fontsize = 16)
ax2.set_xlabel('Magnification',fontsize = 16)
ax2.tick_params(axis='both', which='major', labelsize=16)

polar_halfs[(0,0)]['means'].zslice.plot(ax=ax3,label='polar method',fontsize=16)
avs_S0D0.Z.plot(ax=ax3,label = 'intepolation method',fontsize=16)
ax3.set_title("xy-plane",fontsize = 16)
ax3.tick_params(axis='both', which='major', labelsize=16)
ax3.set_xlabel('Magnification',fontsize = 16)
```

Out[24]:
```
<matplotlib.text.Text at 0x10c190090>
```



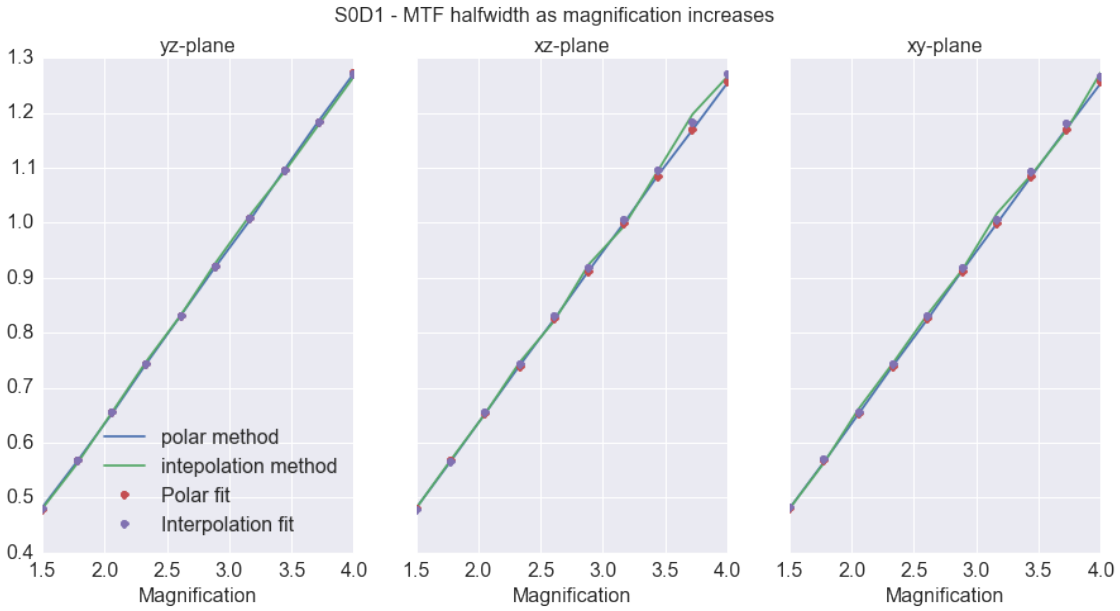
In [25]:
```
fig,(ax1,ax2,ax3) = plt.subplots(ncols=3,nrows=1,figsize = (15,7),sharey=True)

fig.suptitle('S1D0 - MTF halfwidth as magnification increases',fontsize=16)

polar_halfs[(1,0)]['means'].xslice.plot(ax=ax1,label='polar method',fontsize=16)
avs_S1D0.X.plot(ax=ax1,label = 'intepolation method',fontsize=16)
```

```
ax1.plot(mags.data.tolist(),recip(np.array(mags.data.tolist()),*popt1_e),'o' \
         ,label = 'Polar fit')
ax1.plot(mags.data.tolist(),recip(np.array(mags.data.tolist()),*popt4_e),'o' \
         ,label = 'Interpolation fit')
ax1.legend(loc=0,fontsize=16)
ax1.set_xlabel('Magnification',fontsize = 16)
ax1.set_title("yz-plane",fontsize = 16)
ax1.set_ylabel('MTF half width (1/mm)',fontsize = 16)
ax1.tick_params(axis='both', which='major', labelsize=16)

polar_halfs[(1,0)]['means'].yslice.plot(ax=ax2,label='polar method',fontsize=16)
avs_S1D0.Y.plot(ax=ax2,label = 'intepolation method',fontsize=16)
ax2.plot(mags.data.tolist(),recip(np.array(mags.data.tolist()),*popt2_e),'o' \
         ,label = 'Polar fit')
ax2.plot(mags.data.tolist(),recip(np.array(mags.data.tolist()),*popt5_e),'o' \
         ,label = 'Interpolation fit')
ax2.set_title("xz-plane",fontsize = 16)
ax2.set_xlabel('Magnification',fontsize = 16)
ax2.tick_params(axis='both', which='major', labelsize=16)

polar_halfs[(1,0)]['means'].zslice.plot(ax=ax3,label='polar method',fontsize=16)
avs_S1D0.Z.plot(ax=ax3,label = 'intepolation method',fontsize=16)
ax3.plot(mags.data.tolist(),recip(np.array(mags.data.tolist()),*popt3_e),'o' \
         ,label = 'Polar fit')
ax3.plot(mags.data.tolist(),recip(np.array(mags.data.tolist()),*popt6_e),'o' \
         ,label = 'Interpolation fit')
ax3.set_title("xy-plane",fontsize = 16)
ax3.set_xlabel('Magnification',fontsize = 16)
ax3.tick_params(axis='both', which='major', labelsize=16)
```



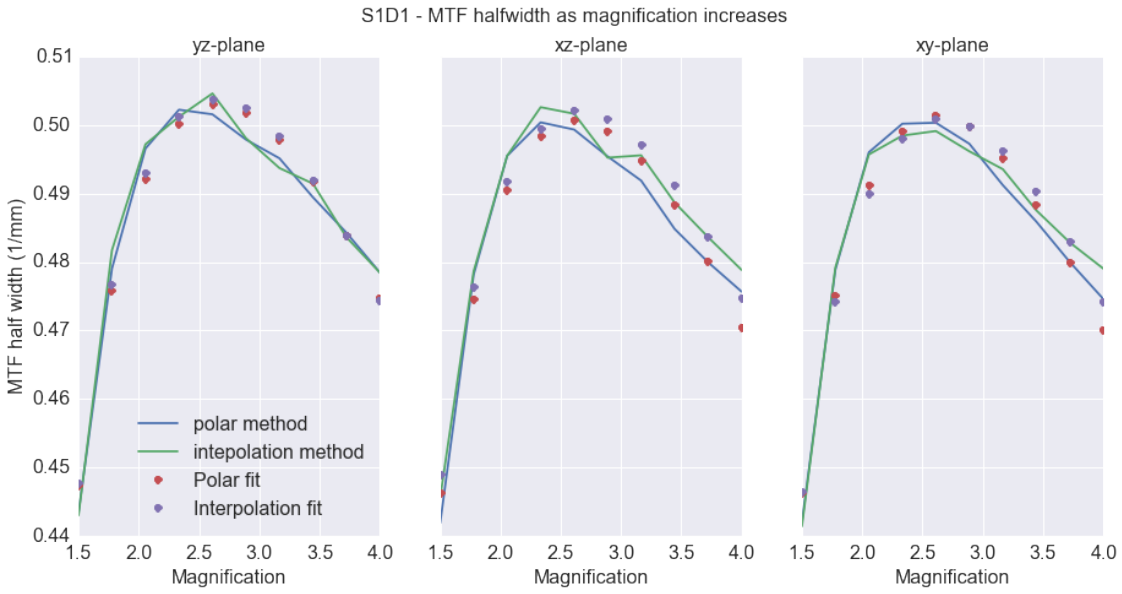
In [26]: `fig,(ax1,ax2,ax3) = plt.subplots(ncols=3,nrows=1,figsize = (15,7),sharey=True)`

```
fig.suptitle('S0D1 - MTF halfwidth as magnification increases',fontsize=16)

polar_halfs[(0,1)]['means'].xslice.plot(ax=ax1,label='polar method',fontsize=16)
avs_S0D1.X.plot(ax=ax1,label = 'intepolation method',fontsize=16)
ax1.plot(mags.data.tolist(),lin(np.array(mags.data.tolist()),*popt1_l),'o' \
         ,label = 'Polar fit')
ax1.plot(mags.data.tolist(),lin(np.array(mags.data.tolist()),*popt4_l),'o' \
         ,label = 'Interpolation fit')
ax1.legend(loc=0,fontsize=16)
ax1.set_title("yz-plane",fontsize = 16)
ax1.set_xlabel('Magnification',fontsize = 16)
ax1.tick_params(axis='both', which='major', labelsize=16)

polar_halfs[(0,1)]['means'].yslice.plot(ax=ax2,label='polar method',fontsize=16)
avs_S0D1.Y.plot(ax=ax2,label = 'intepolation method',fontsize=16)
ax2.plot(mags.data.tolist(),lin(np.array(mags.data.tolist()),*popt2_l),'o' \
         ,label = 'Polar fit')
ax2.plot(mags.data.tolist(),lin(np.array(mags.data.tolist()),*popt5_l),'o' \
         ,label = 'Interpolation fit')
ax2.set_title("xz-plane",fontsize = 16)
ax2.set_xlabel('Magnification',fontsize = 16)
ax2.tick_params(axis='both', which='major', labelsize=16)

polar_halfs[(0,1)]['means'].zslice.plot(ax=ax3,label='polar method',fontsize=16)
avs_S0D1.Z.plot(ax=ax3,label = 'intepolation method',fontsize=16)
ax3.plot(mags.data.tolist(),lin(np.array(mags.data.tolist()),*popt3_l),'o' \
         ,label = 'Polar fit')
ax3.plot(mags.data.tolist(),lin(np.array(mags.data.tolist()),*popt6_l),'o' \
         ,label = 'Interpolation fit')
ax3.set_title("xy-plane",fontsize = 16)
ax3.set_xlabel('Magnification',fontsize = 16)
ax3.tick_params(axis='both', which='major', labelsize=16)
```



In [27]:
```
fig,(ax1,ax2,ax3) = plt.subplots(ncols=3,nrows=1,figsize = (15,7),sharey=True)

fig.suptitle('S1D1 - MTF halfwidth as magnification increases',fontsize=16)

polar_halfs[(1,1)]['means'].xslice.plot(ax=ax1,label='polar method',fontsize=16)
avs_S1D1.X.plot(ax=ax1,label = 'intepolation method',fontsize=16)
ax1.plot(mags.data.tolist(),reciplin(np.array(mags.data.tolist()),*popt1),'o' \
         ,label = 'Polar fit')
ax1.plot(mags.data.tolist(),reciplin(np.array(mags.data.tolist()),*popt4),'o' \
         ,label = 'Interpolation fit')
ax1.legend(loc=0,fontsize=16)
ax1.set_ylabel('MTF half width (1/mm)',fontsize = 16)
ax1.set_xlabel('Magnification',fontsize = 16)
ax1.set_title("yz-plane",fontsize = 16)
ax1.tick_params(axis='both', which='major', labelsize=16)

polar_halfs[(1,1)]['means'].yslice.plot(ax=ax2,label='polar method',fontsize=16)
avs_S1D1.Y.plot(ax=ax2,label = 'intepolation method',fontsize=16)
ax2.plot(mags.data.tolist(),reciplin(np.array(mags.data.tolist()),*popt2),'o' \
         ,label = 'Polar fit')
ax2.plot(mags.data.tolist(),reciplin(np.array(mags.data.tolist()),*popt5),'o' \
         ,label = 'Interpolation fit')
ax2.set_title("xz-plane",fontsize = 16)
ax2.tick_params(axis='both', which='major', labelsize=16)
ax2.set_xlabel('Magnification',fontsize = 16)

polar_halfs[(1,1)]['means'].zslice.plot(ax=ax3,label='polar method',fontsize=16)
avs_S1D1.Z.plot(ax=ax3,label = 'intepolation method',fontsize=16)
ax3.plot(mags.data.tolist(),reciplin(np.array(mags.data.tolist()),*popt3),'o' \
         ,label = 'Polar fit')
ax3.plot(mags.data.tolist(),reciplin(np.array(mags.data.tolist()),*popt6),'o' \
         ,label = 'Interpolation fit')
ax3.set_title("xy-plane",fontsize = 16)
ax3.tick_params(axis='both', which='major', labelsize=16)
ax3.set_xlabel('Magnification',fontsize = 16)
```

Out[27]:
```
<matplotlib.text.Text at 0x10d3fdf90>
```



In [28]:
```
S1D0 = polar_halfs[(1,0)]['means'].xslice_radius.apply(lambda x:np.abs(x-30)/30).as_matrix()
S0D1 = polar_halfs[(0,1)]['means'].xslice_radius.apply(lambda x:np.abs(x-30)/30).as_matrix()
S1D1 = polar_halfs[(1,1)]['means'].xslice_radius.apply(lambda x:np.abs(x-30)/30).as_matrix()
```

In [29]:
```
mags = polar_halfs_dat.mag.unique()
fig,ax = plt.subplots(ncols=1,nrows=1,figsize = (10,7))
ax.plot(mags,S1D0,'g*-',label="S1D0")
ax.plot(mags,S0D1,'r*-',label="S0D1")
ax.plot(mags,S1D1,'m*-',label="S1D1")

ax.legend(loc=0,fontsize=16)
ax.set_title("Relative error in Radius - Magnification",fontsize = 16)
ax.set_ylabel("Relative Error (%)",fontsize = 16)
ax.tick_params(axis='both', which='major', labelsize=16)
ax.set_xlabel("Magnification",fontsize = 16);
```

In [36]:
```
# interp fit rmse and params S0D1
print(rmse_ix01)
print(rmse_iy01)
print(rmse_iz01)

print(popt4_l)
print(popt5_l)
print(popt6_l)
```

```
0.00442263917271
0.00767628461625
0.00663447812617
[ 0.31577521  0.00678286]
[ 0.31692495  0.0025919 ]
[ 0.31401344  0.01062478]
```

In [37]:
```
# polar fit rmse and params S0D1
print(rmse_px01)
print(rmse_py01)
print(rmse_pz01)

print(popt1_l)
print(popt2_l)
print(popt3_l)
```

```
0.00211379325631
0.00136108594292
```

```
0.00134643254512
[ 0.31720477  0.002951  ]
[ 0.31003435  0.01573987]
[ 0.31029769  0.01497927]
```

```
In [38]: # interp fit rmse and params S1D0
         print(rmse_ix10)
         print(rmse_iy10)
         print(rmse_iz10)

         print(popt4_e)
         print(popt5_e)
         print(popt6_e)
```

```
0.00856441139703
0.00525399469885
0.00828031556333
[ 0.46067748  0.39338197]
[ 0.54436188  0.36537491]
[ 0.45868277  0.3940719 ]
```

```
In [39]: # polar fit rmse and params S1D0
         print(rmse_px10)
         print(rmse_py10)
         print(rmse_pz10)

         print(popt1_e)
         print(popt2_e)
         print(popt3_e)
```

```
0.00770432756451
0.00624199963884
0.00630602499898
[ 0.56581034  0.35770887]
[ 0.48093065  0.38332747]
[ 0.4791566   0.38430641]
```

```
In [34]: # interp fit rmse and params S1D1
         print(rmse_ix11)
         print(rmse_iy11)
         print(rmse_iz11)

         print(popt4)
         print(popt5)
         print(popt6)
```

```
0.0035205201825
0.00313518834094
0.0037079031746
[-0.44940586 -0.06424152  0.84365347]
[-0.42255085 -0.06010057  0.82082675]
[-0.42665255 -0.05999987  0.82088555]
```

```
In [35]: # polar fit rmse and params S1D1
         print(rmse_px11)
         print(rmse_py11)
```

```
        print(rmse_pz11)

        print(popt1)
        print(popt2)
        print(popt3)
```

```
0.0030633637539
0.0035346552361
0.0032839156787
[-0.44218504 -0.06273519  0.83612175]
[-0.44324303 -0.06416706  0.83798045]
[-0.45329828 -0.06602064  0.84743266]
```