

pole_correction_plots

May 18, 2020

0.1 # Plots Together

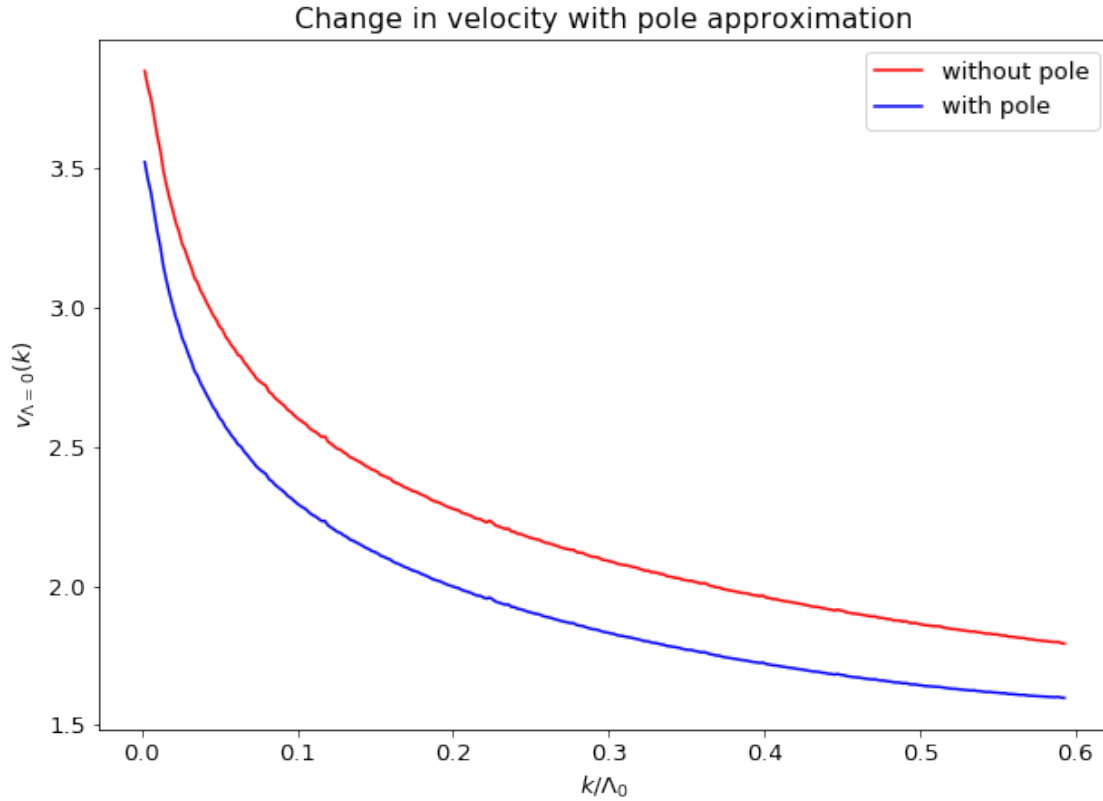
```
[1]: import matplotlib as mpl
import matplotlib.pyplot as plt
import pandas as pd
%matplotlib inline
```

```
[2]: plt.rcParams['figure.figsize'] = 10,7
mpl.rcParams['font.size'] = 13
```

0.2 Velocity

```
[3]: vel = pd.read_csv("/home/biplab/Desktop/Code/fortran/pole/velon0.
↳dat",names=('momentum','velocity'),delim_whitespace=True,dtype='float64')
velw_poles = pd.read_csv("/home/biplab/Desktop/Code/fortran/pole/velcon0.
↳dat",names=('momentum','velocity'),delim_whitespace=True,dtype='float64')
```

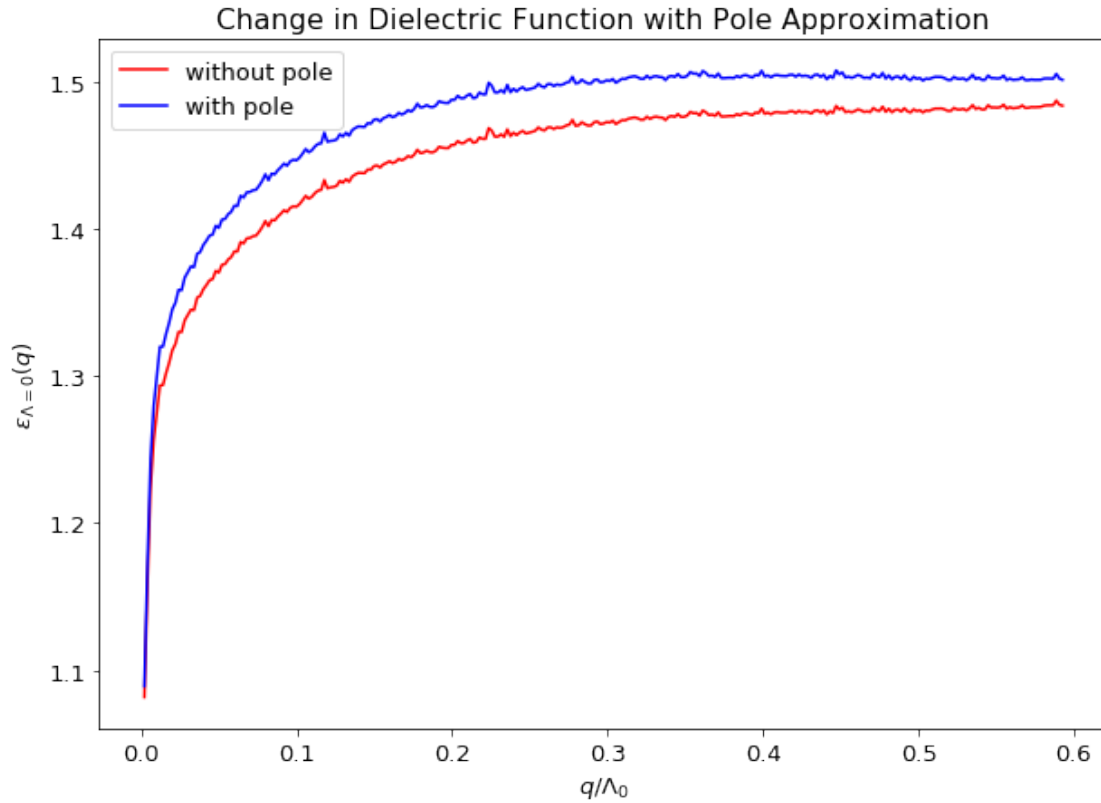
```
[4]: plt.plot(vel['momentum'],vel['velocity'],'r',label='without pole')
plt.plot(velw_poles['momentum'],velw_poles['velocity'],'b',label='with pole')
plt.xlabel(r'$k/\Lambda_0$')
plt.ylabel(r'$v_{\Lambda = 0}(k)$')
plt.title('Change in velocity with pole approximation')
plt.legend()
plt.savefig('image/pole_approximation_velocity_trans.png',transparent=True)
```



0.3 Dielectric Function

```
[5]: eps = pd.read_csv("/home/biplab/Desktop/Code/fortran/pole/epson0.
↳dat",names=('momentum','dielectric'),delim_whitespace=True,dtype='float64')
epsw_poles = pd.read_csv("/home/biplab/Desktop/Code/fortran/pole/epscon0.
↳dat",names=('momentum','dielectric'),delim_whitespace=True,dtype='float64')
```

```
[6]: plt.plot(eps['momentum'],eps['dielectric'],'r',label='without pole')
plt.plot(epsw_poles['momentum'],epsw_poles['dielectric'],'b',label='with pole')
plt.xlabel(r'$q/\Lambda_0$')
plt.ylabel(r'$\epsilon_{\Lambda = 0}(q)$')
plt.title('Change in Dielectric Function with Pole Approximation')
plt.legend()
plt.savefig('image/pole_approximation_dielectric.png',transparent=False)
```



0.4 Correction Term

```
[7]: eps2 = pd.read_csv("/home/biplab/Desktop/Code/fortran/pole/epsc2on0.
↳ dat", names=('momentum', 'dielectric'), delim_whitespace=True, dtype='float64')
```

```
[8]: plt.plot(eps2['momentum'], eps2['dielectric'])
plt.xlabel(r'$q/\Lambda_0$')
plt.ylabel(r'$\epsilon_{\Lambda = 0}(q)$')
plt.title('First Order correction due to frequency')
plt.savefig('image/pole_approximation_correction.png', transparent=False)
```

