Data Analysis of TSiKN65 Material

This file extracts the $\Delta\epsilon$ value from the fitted model I use on my data. The inverse of this value is plotted against the temperature for each spectrum at that given temperature. From this, we fit a quadratic formula to determine the value of the susceptibility critical exponent, γ.

The cell below defines a function used to extract the data. It returns the parameter values determined by our model and scipy's "curve_fit" function, a weighted least-squares fit.

In [167]:
```
%pylab inline
from mpl_toolkits.mplot3d import Axes3D
import scipy.optimize as spo

def data_extract(filenames, guess, bool_to_guess, bool_to_print, bool_to_plot):

    files = [filenames[0],filenames[1]]

    w = np.loadtxt('Data03082013/weights.txt', dtype=float,delimiter='\t')
    w = w.T
  #  w = w[100:]
    test = np.loadtxt('Data03082013/'+files[0], dtype=float,delimiter='\t')
    #magnitude
    m = test.T[0]
  #  m = m[100:]
    #frequency
    f = test.T[1]
  #  f = f[100:]
    #angle projected into imaginery/real plane
    a = test.T[4]
  #  a = a[100:]

    test2 = np.loadtxt('Data03082013/'+files[1], dtype=float,delimiter='\t')
    m2 = test2.T[0]
    f2 = test2.T[1]
    a2 = test2.T[4]

    """ Combine the data into a 3D array """
    magn = np.concatenate((m,m2))
    freq = np.concatenate((f,f2))
    angl = np.concatenate((a,a2))

    """ Put data in terms of the dielectric constant """
    A = .01**2;
    eps0 = 8.854187e-12;
    d = 3e-6;
    omega = 2*np.pi*freq;

    G = cos(angl)/(magn);
```

```
    C = -sin(angl)/(magn);

    C0 = (A*eps0)/(d);

    """ e2 is the epsilon double primed or the imaginery part of the dielectric data
        e1 is the epsilong primed or real part of the dielectric data """

    e2 = G/(omega*C0);
    e1 = C/(omega*C0);

    """ GUESSED VALUES for the fit of the primary peak in the dielectric data. """
    E_infinity = guess[0];
    Delta_E = guess[1];
    Relaxation_frequency = guess[2];
    Beta = guess[3];
    G_low_frequency = guess[4];

    """ GUESSSED VALUES for the fit of the secondary peak in dielectric data. """
    E_infinity2 = guess[5];
    Delta_E2 = guess[6];
    Relaxation_frequency2 = guess[7];
    Beta2 = guess[8];
    G_low_frequency2 = guess[9];

    """ Defining the data as x,y """
    x = freq;
    y = e2;

    """ Model Function to fit our data """
    def func(x, Einf,delE,relaxFreq,beta,g,Einf2,delE2,relaxFreq2,beta2,g2):
        z = (Einf + (delE/(1 + (1j*(x/relaxFreq))**(beta))) + g + Einf2 + (delE2/(1 +
(1j*(x/relaxFreq2))**(beta2))) + g2)
        return -(z.imag)

    dummy = func(x,E_infinity, Delta_E, Relaxation_frequency, Beta, G_low_frequency,
E_infinity2, Delta_E2, Relaxation_frequency2, Beta2, G_low_frequency2)

    if bool_to_guess == True:
        plt.figure(1)
        semilogx(freq,dummy,'-r')
        semilogx(freq,e2,'.')
        xlabel('Frequency')
        ylabel('$\epsilon$')
        title('Dielectric Spectrum')
        grid()

    popt, pcov = spo.curve_fit(func,x,y,p0=[E_infinity, Delta_E, Relaxation_frequency,
Beta, G_low_frequency, E_infinity2, Delta_E2, Relaxation_frequency2, Beta2,
G_low_frequency2], sigma=w, maxfev=10000)

    fittedmodel =
func(x,popt[0],popt[1],popt[2],popt[3],popt[4],popt[5],popt[6],popt[7],popt[8],popt[9])
```

```
    if bool_to_print == True:
        def fit_results(param,values):
            for i in range(0,10,1):
                print param[i] + str(values[i])

        """ Print out the values of the fit """
        params = ["\nRESULTS FOR PRIMARY PEAK: \n\nE_infinity: ", "Delta_E: ",
"Relaxation_frequency: ", "Beta: ", "G_low_frequency: ",
                  "\nRESULTS FOR SECONDARY PEAK: \n\nE_infinity: ", "Delta_E: ",
"Relaxation_frequency: ", "Beta: ", "G_low_frequency: "];
        fit_results(params,popt)

    if bool_to_plot == True:
        """ Plot the fitted model """
        plt.figure(2)
        semilogx(freq,e2,'.')
        semilogx(freq,fittedmodel,'r')
        xlabel('Frequency')
        ylabel('$\epsilon$')
        title('Dielectric Spectrum')
        ylim((0,16))
        grid()

    return popt, fittedmodel
```

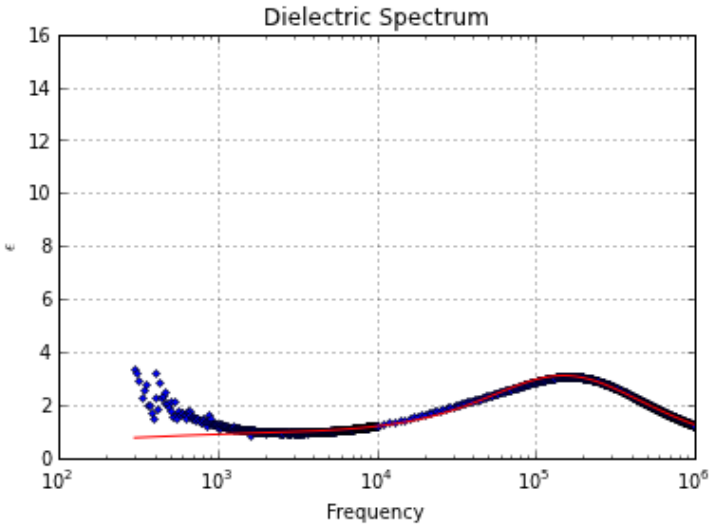
```
Welcome to pylab, a matplotlib-based Python environment [backend:
module://IPython.kernel.zmq.pylab.backend_inline].
For more information, type 'help(pylab)'.
```

Call the function above on each spectrum and return the fit values.

In [168]:
```
""" GUESSED VALUES for the fit of the primary peak in the dielectric data. """
E_infinity = 0;
Delta_E = 15;
Relaxation_frequency = 2500;
Beta = .5;
G_low_frequency = 0;

""" GUESSSED VALUES for the fit of the secondary peak in dielectric data. """
E_infinity2 = 0;
Delta_E2 = 10;
Relaxation_frequency2 = 150000;
Beta2 = .9;
G_low_frequency2 = 0;

guess = [E_infinity, Delta_E, Relaxation_frequency, Beta, G_low_frequency,
E_infinity2, Delta_E2, Relaxation_frequency2, Beta2, G_low_frequency2]

popt1, fittedmodel1 = data_extract(['test01','test02'], guess, False, True, True)
```

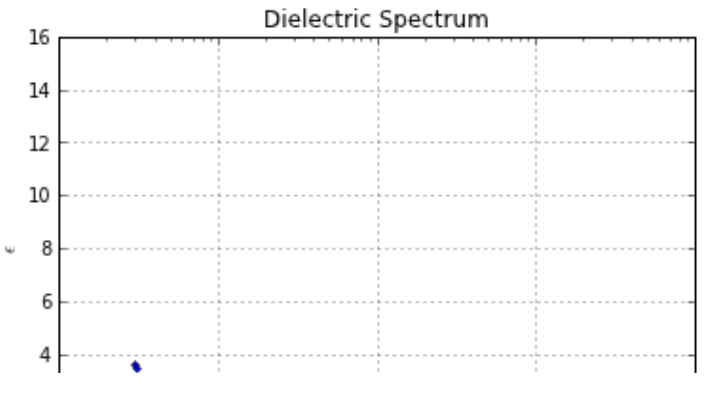
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 4.83484789508
```

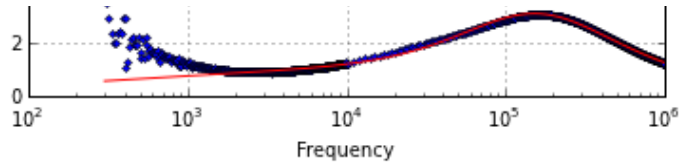
```
Relaxation_frequency: -1451.21798264
Beta: -0.421719282528
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 6.3605015633
Relaxation_frequency: 158203.197642
Beta: 0.911964316778
G_low_frequency: 0.0
```



In [169]:
```
popt2, fittedmodel2 = data_extract(['test03','test04'], guess, False, True, True)
```

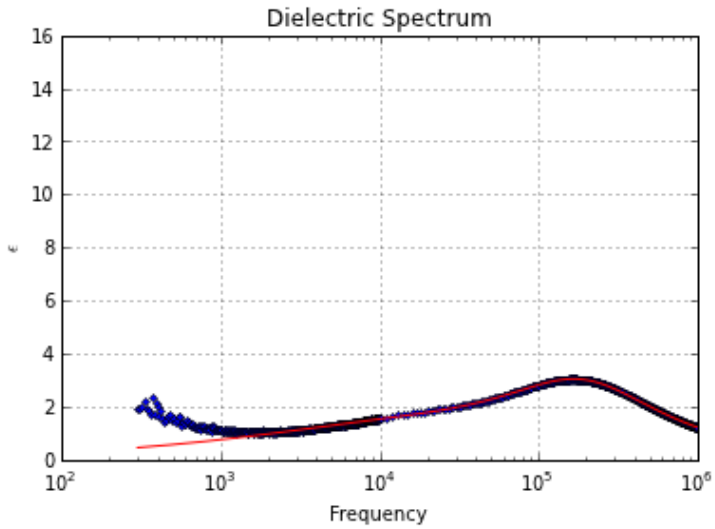
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 4.14723330416
Relaxation_frequency: 4071.36512592
Beta: 0.450793712629
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 6.11989640352
Relaxation_frequency: 161271.846496
Beta: 0.920898335413
G_low_frequency: 0.0
```





In [170]:
```
popt3, fittedmodel3 = data_extract(['test05','test06'], guess, False, True, True)
```

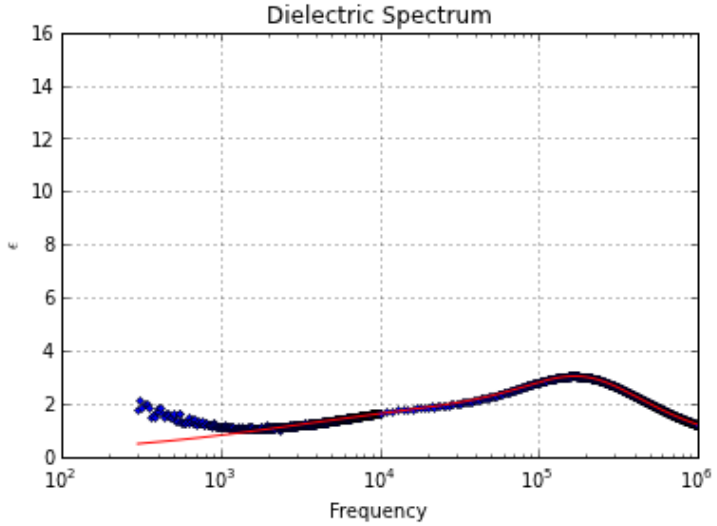
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 6.35539281559
Relaxation_frequency: 19862.0192039
Beta: 0.521108220475
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 4.00329561371
Relaxation_frequency: 188770.167133
Beta: 1.02362490918
G_low_frequency: 0.0
```



In [171]:
```
popt4, fittedmodel4 = data_extract(['test07','test08'], guess, False, True, True)
```

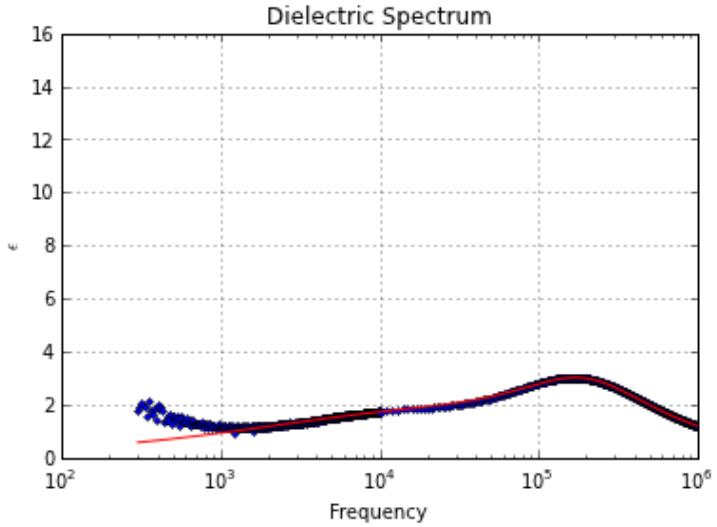
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 6.96647539597
Relaxation_frequency: 20599.3715717
Beta: 0.523842209073
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 3.6519028785
Relaxation_frequency: 193801.66922
Beta: 1.04837737827
```

```
G_low_frequency: 0.0
```



In [172]:
```
popt5, fittedmodel5 = data_extract(['test09','test10'], guess, False, True, True)
```

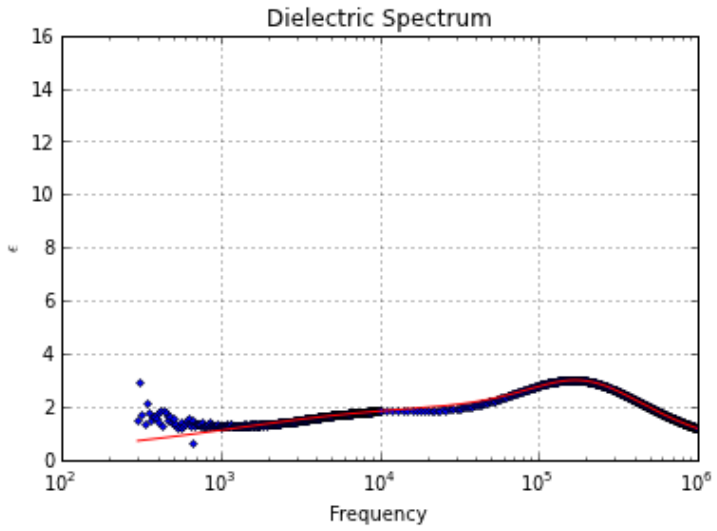
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 7.31835143472
Relaxation_frequency: 15980.8124522
Beta: 0.521151869392
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 3.69971372194
Relaxation_frequency: 195939.540272
Beta: 1.04665821285
G_low_frequency: 0.0
```



In [173]:
```
popt6, fittedmodel6 = data_extract(['test11','test12'], guess, False, True, True)
```

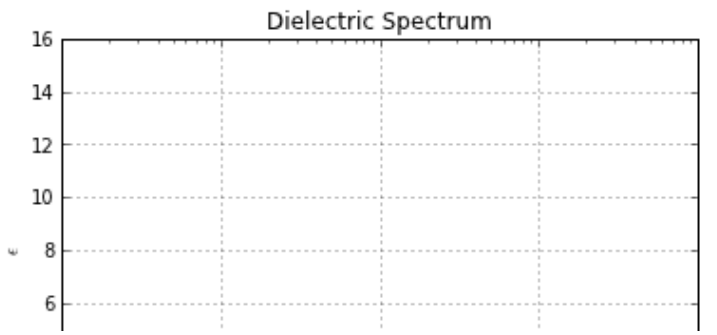
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
```

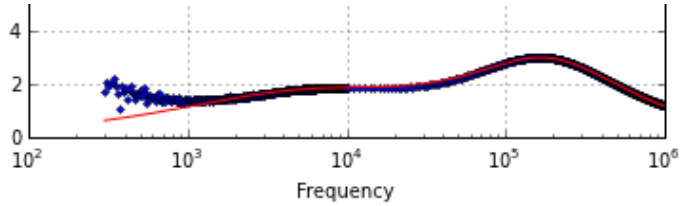
```
Delta_E: 7.85188060788
Relaxation_frequency: 11553.3064374
Beta: 0.51130424142
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 3.7294122309
Relaxation_frequency: 196884.459867
Beta: 1.04819996081
G_low_frequency: 0.0
```



In [174]:
```
popt7, fittedmodel7 = data_extract(['test13','test14'], guess, False, True, True)
```

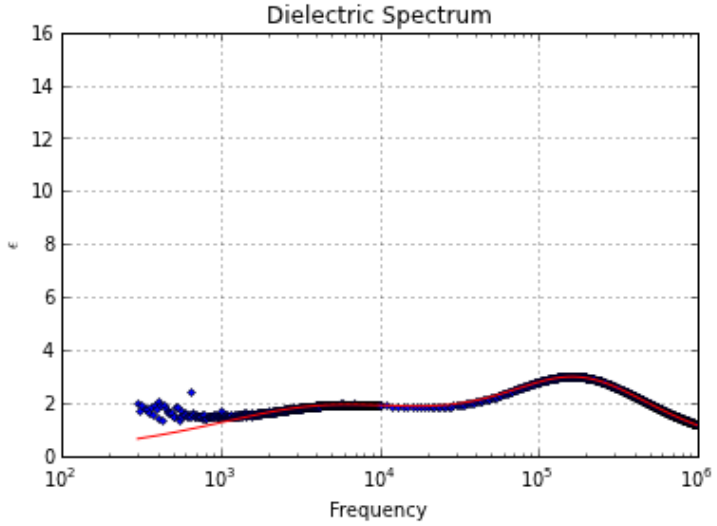
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 5.6516419668
Relaxation_frequency: 5532.25346757
Beta: 0.66331941147
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 5.40776835736
Relaxation_frequency: 185759.82899
Beta: 0.96198871258
G_low_frequency: 0.0
```





In [175]:
```
popt8, fittedmodel8 = data_extract(['test15','test16'], guess, False, True, True)
```

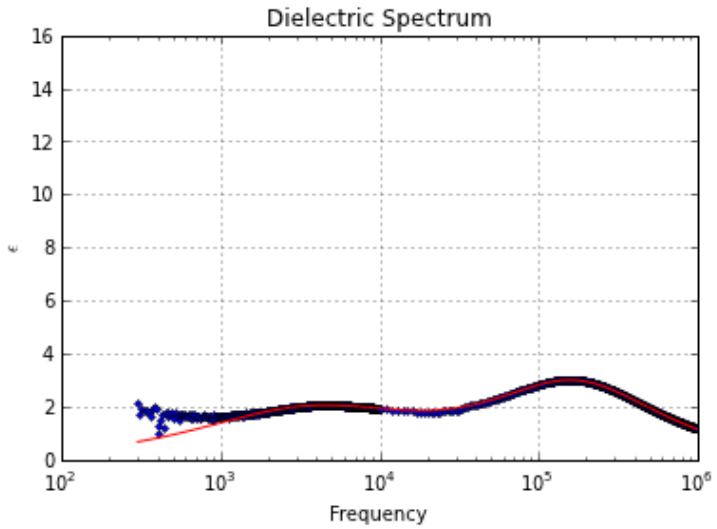
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 5.49236136069
Relaxation_frequency: 4389.63835011
Beta: 0.719688508668
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 5.73904900033
Relaxation_frequency: 180216.427906
Beta: 0.94980055765
G_low_frequency: 0.0
```



In [176]:
```
popt9, fittedmodel9 = data_extract(['test17','test18'], guess, False, True, True)
```

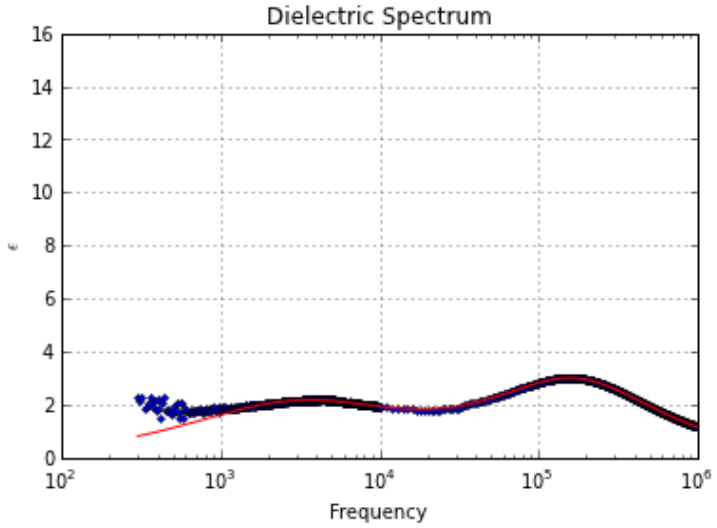
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 5.39744414112
Relaxation_frequency: 3636.26154669
Beta: 0.772699443391
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0

Delta_E: 6.07305303797
```

```
Relaxation_frequency: 171430.642555
Beta: 0.9352542898
G_low_frequency: 0.0
```



In [177]:
```
popt10, fittedmodel10 = data_extract(['test19','test20'], guess, False, True, True)
```

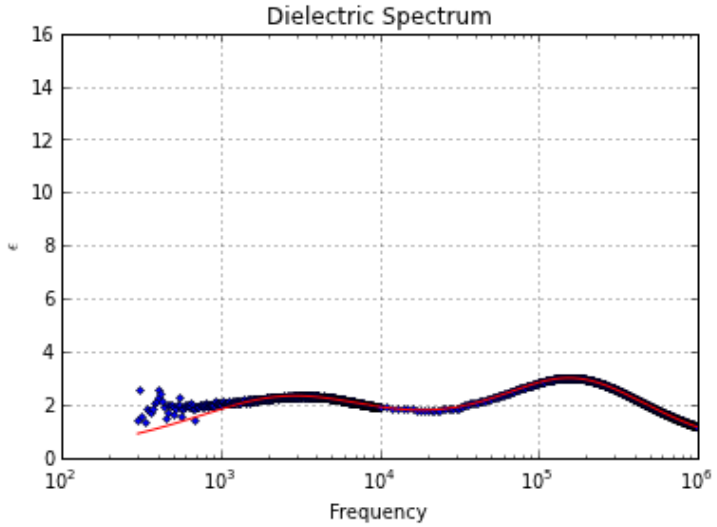
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 5.80146987443
Relaxation_frequency: 3026.42871102
Beta: 0.77620720075
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 6.14303439883
Relaxation_frequency: 169885.392647
Beta: 0.932500124397
G_low_frequency: 0.0
```



In [178]:
```
popt11, fittedmodel11 = data_extract(['test21','test22'], guess, False, True, True)
```

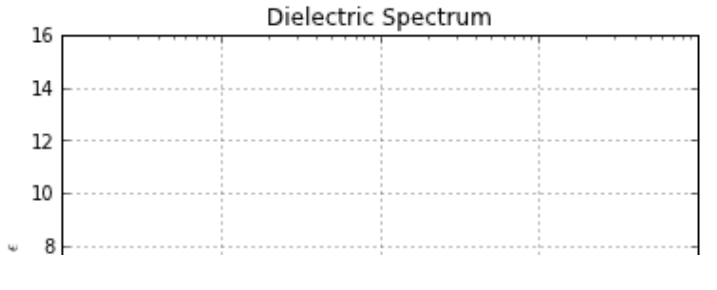
```
RESULTS FOR PRIMARY PEAK:
```

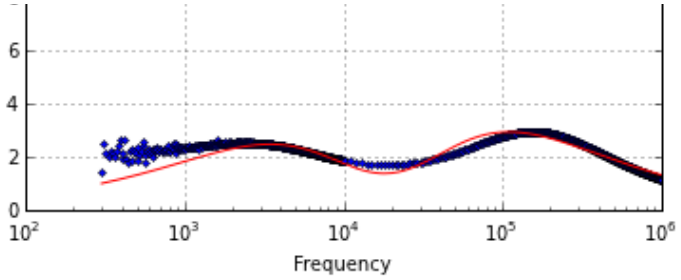
```
E_infinity: 0.0
Delta_E: 6.0155731992
Relaxation_frequency: 2602.20752991
Beta: 0.802478164365
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 6.28064508946
Relaxation_frequency: 167125.122212
Beta: 0.92715827509
G_low_frequency: 0.0
```



In [203]:
```
popt12, fittedmodel12 = data_extract(['test23','test24'], guess, False, True, True)
```

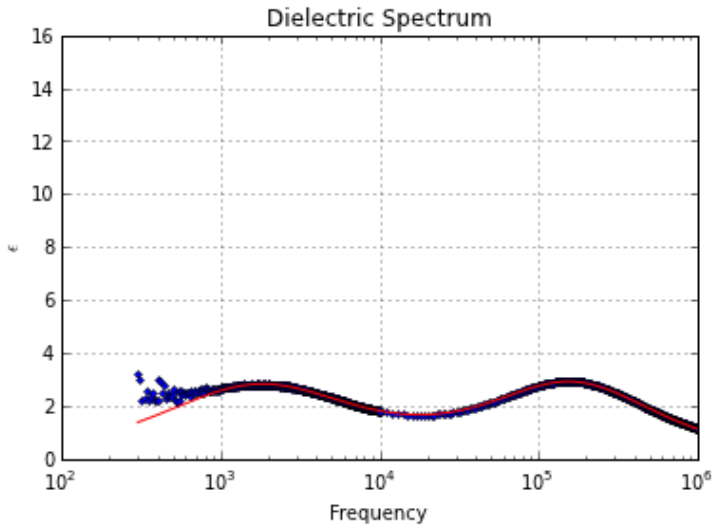
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 1.00001657305
Delta_E: -4973.29944897
Relaxation_frequency: 20390.6290499
Beta: 0.792386872048
G_low_frequency: 1.00001375054

RESULTS FOR SECONDARY PEAK:

E_infinity: 1.00001347511
Delta_E: 4986.13504084
Relaxation_frequency: 20398.6455949
Beta: 0.791300164622
G_low_frequency: 1.00001272438
```





In [180]:
```
popt13, fittedmodel13 = data_extract(['test25','test26'], guess, False, True, True)
```

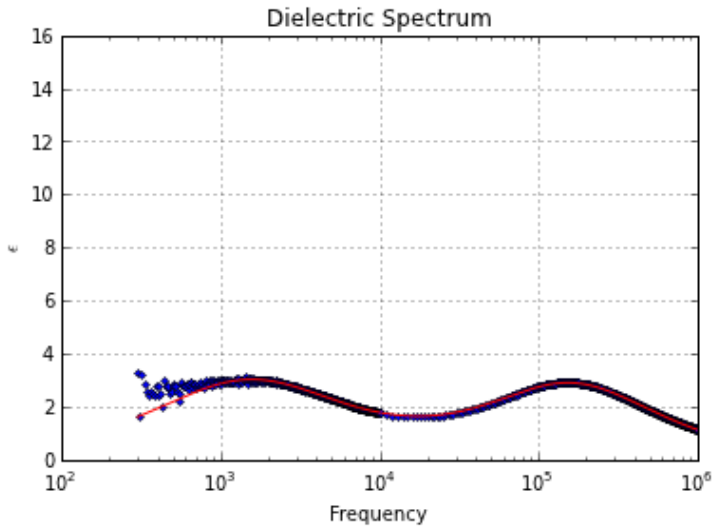
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 7.08990980696
Relaxation_frequency: 1724.32696549
Beta: 0.8357130783
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 6.27489342192
Relaxation_frequency: 163957.678057
Beta: 0.919923648869
G_low_frequency: 0.0
```



In [181]:
```
popt14, fittedmodel14 = data_extract(['test27','test28'], guess, False, True, True)
```

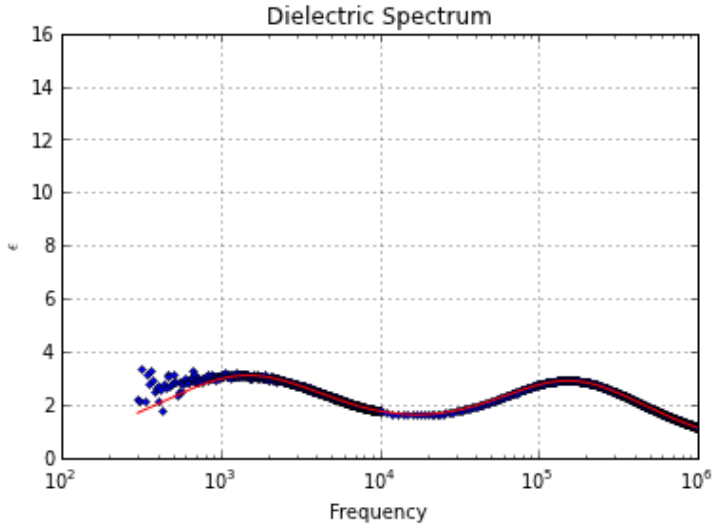
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 7.56690742034
Relaxation_frequency: 1491.87829764
Beta: 0.840289568025
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
```

```
Delta_E: 6.31754123994
Relaxation_frequency: 162386.339084
Beta: 0.913117910437
G_low_frequency: 0.0
```



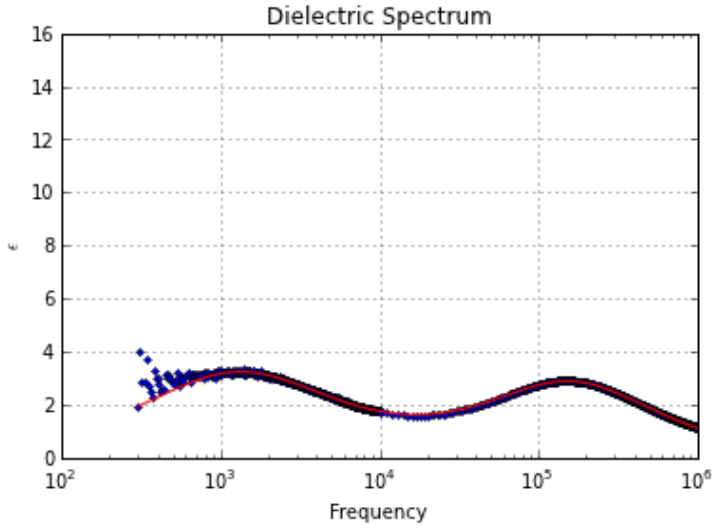
In [182]:
```
popt15, fittedmodel15 = data_extract(['test29','test30'], guess, False, True, True)
```

```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 7.69702571271
Relaxation_frequency: 1426.11365889
Beta: 0.84623533911
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 6.34415766174
Relaxation_frequency: 161775.68699
Beta: 0.912068140767
G_low_frequency: 0.0
```



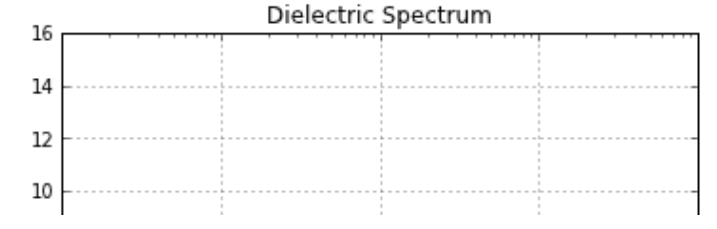
In [183]:
```
popt16, fittedmodel16 = data_extract(['test31','test32'], guess, False, True, True)
```

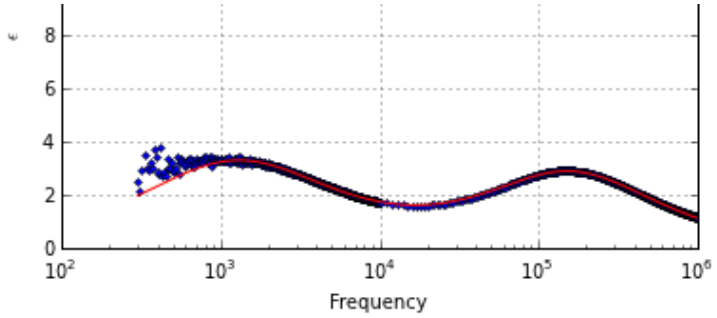
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 8.196919553
Relaxation_frequency: 1279.40845464
Beta: 0.836193795887
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 6.3041159349
Relaxation_frequency: 160432.526906
Beta: 0.911846178456
G_low_frequency: 0.0
```



In [184]:
```
popt17, fittedmodel17 = data_extract(['test33','test34'], guess, False, True, True)
```

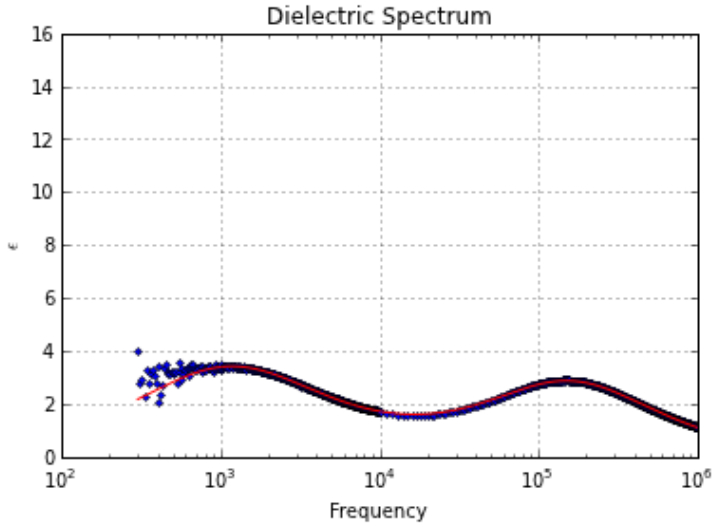
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 8.23158582709
Relaxation_frequency: 1252.61984816
Beta: 0.844877397694
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 6.32962845714
Relaxation_frequency: 159905.222636
Beta: 0.911072474559
G_low_frequency: 0.0
```





In [185]:
```
popt18, fittedmodel18 = data_extract(['test35','test36'], guess, False, True, True)
```

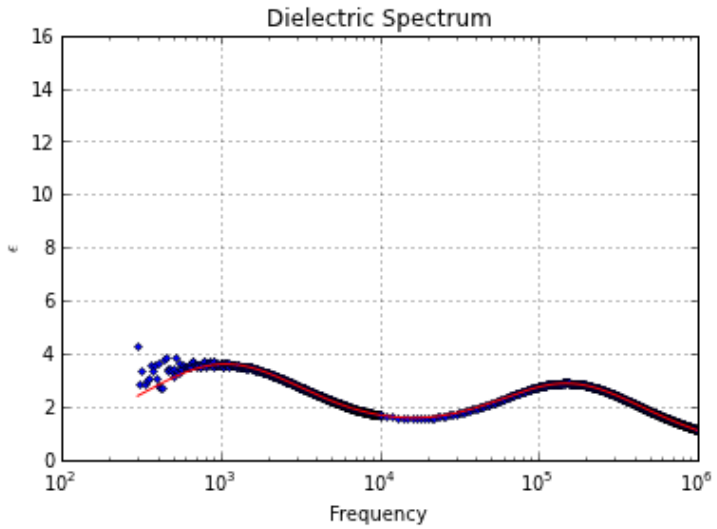
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 8.64142357285
Relaxation_frequency: 1139.74394489
Beta: 0.838668250134
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 6.32403113073
Relaxation_frequency: 158093.878053
Beta: 0.908198135573
G_low_frequency: 0.0
```



In [186]:
```
popt19, fittedmodel19 = data_extract(['test37','test38'], guess, False, True, True)
```

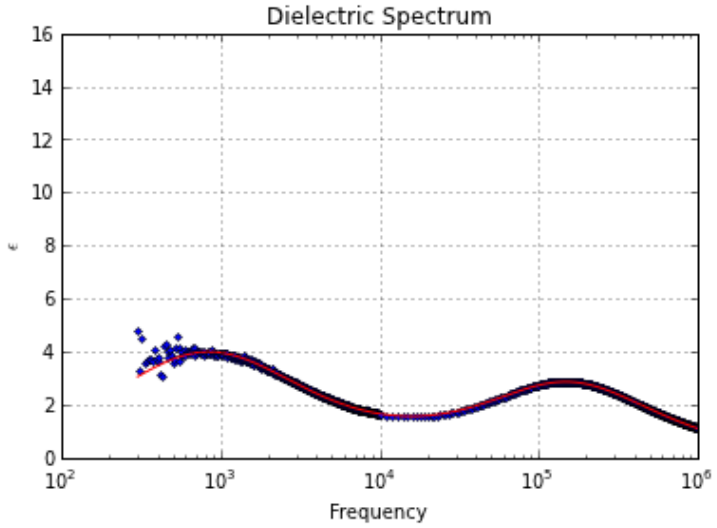
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 9.03181191628
Relaxation_frequency: 1036.30596044
Beta: 0.845716886433
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:
```

```
E_infinity: 0.0
Delta_E: 6.36117099108
Relaxation_frequency: 157230.914157
Beta: 0.906738713181
G_low_frequency: 0.0
```



In [187]:
```
popt20, fittedmodel20 = data_extract(['test39','test40'], guess, False, True, True)
```

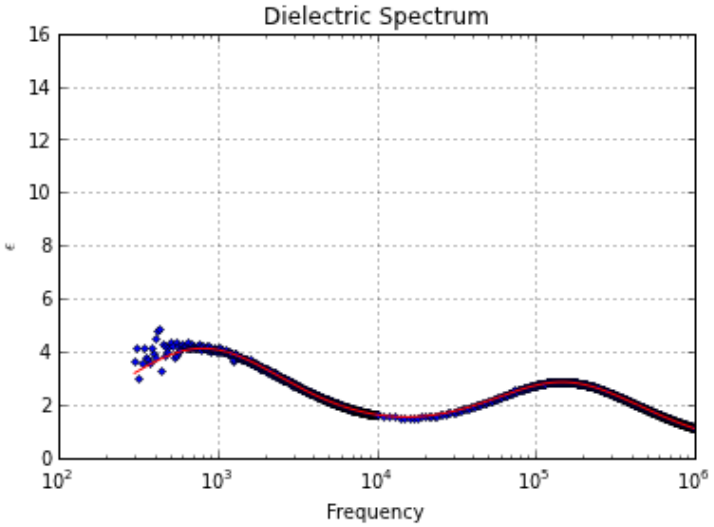
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 10.2574754304
Relaxation_frequency: 822.942258078
Beta: 0.832950325059
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 6.35665452493
Relaxation_frequency: 156192.661725
Beta: 0.904040527225
G_low_frequency: 0.0
```



In [188]:
```
popt21, fittedmodel21 = data_extract(['test41','test42'], guess, False, True, True)
```

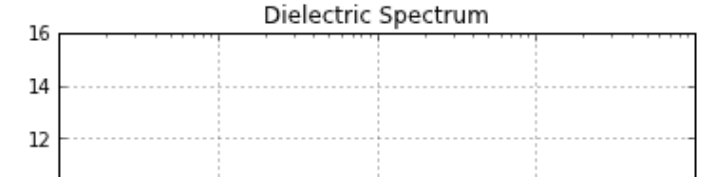
```
popt21, fittedmodel21 = data_extract(['test41','test42'], guess, False, True, True)
```

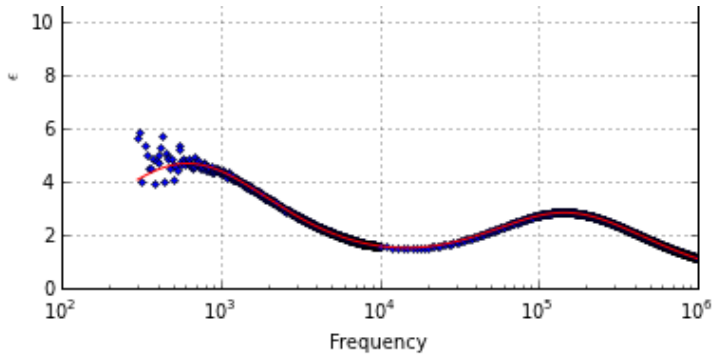
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 10.3881134949
Relaxation_frequency: 788.460717548
Beta: 0.845099869696
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 6.38511369177
Relaxation_frequency: 154616.754588
Beta: 0.900727682047
G_low_frequency: 0.0
```



In [189]:
```
popt22, fittedmodel22 = data_extract(['test43','test44'], guess, False, True, True)
```

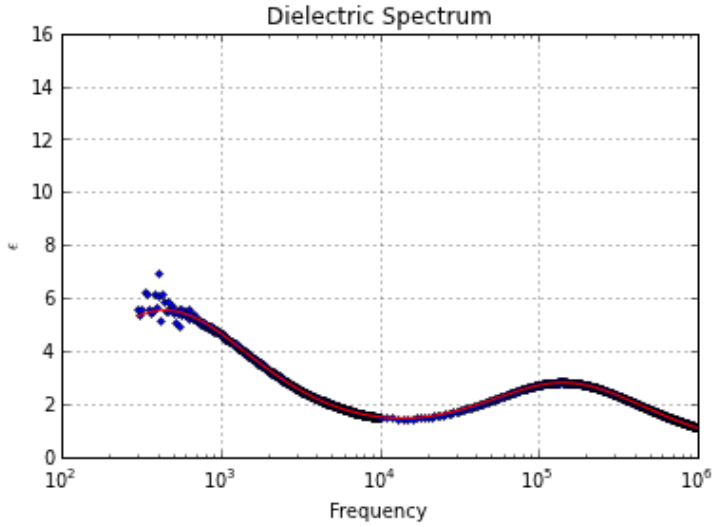
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 11.9141525757
Relaxation_frequency: 608.062391222
Beta: 0.841899302264
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 6.40466699701
Relaxation_frequency: 152504.05802
Beta: 0.895510032412
G_low_frequency: 0.0
```





In [190]:
```
popt23, fittedmodel23 = data_extract(['test45','test46'], guess, False, True, True)
```

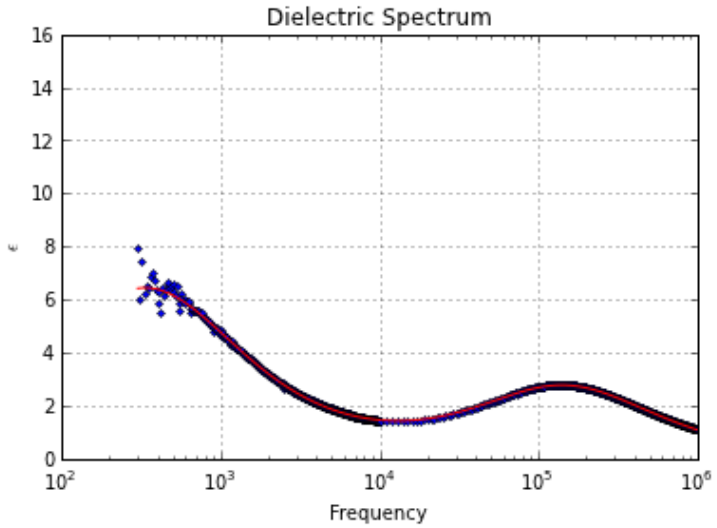
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 14.1787976274
Relaxation_frequency: 438.250740526
Beta: 0.840356693464
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 6.44886575404
Relaxation_frequency: 149436.53564
Beta: 0.886907015655
G_low_frequency: 0.0
```



In [191]:
```
popt24, fittedmodel24 = data_extract(['test47','test48'], guess, False, True, True)
```

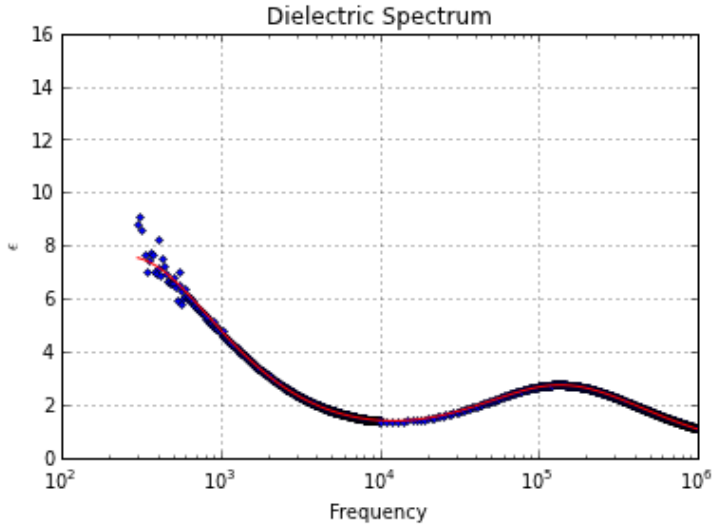
```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 16.4636981851
Relaxation_frequency: 333.175710056
Beta: 0.842160072259
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:
```

```
E_infinity: 0.0
Delta_E: 6.46714937195
Relaxation_frequency: 146644.776785
Beta: 0.880059703238
G_low_frequency: 0.0
```



In [192]:
```
popt25, fittedmodel25 = data_extract(['test49','test50'], guess, False, True, True)
```

```
RESULTS FOR PRIMARY PEAK:

E_infinity: 0.0
Delta_E: 18.994058014
Relaxation_frequency: -261.228077435
Beta: -0.854719377442
G_low_frequency: 0.0

RESULTS FOR SECONDARY PEAK:

E_infinity: 0.0
Delta_E: 6.51625635892
Relaxation_frequency: 143105.438608
Beta: 0.869370021032
G_low_frequency: 0.0
```

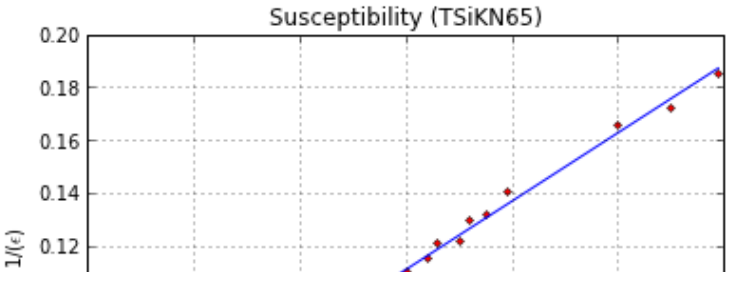


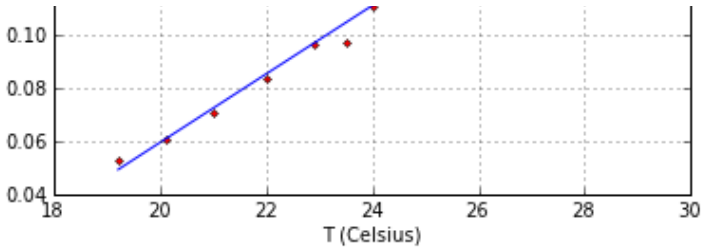
In [200]:
```
#epsilon = [popt1[1],popt2[1],popt3[1],popt4[1],popt5[1],popt6[1],popt7[1],popt8[1],
epsilon = [popt9[1],
popt10[1],popt11[1],popt12[1],popt13[1],popt14[1],popt15[1],popt16[1],popt17[1],popt18[1],

           popt19[1],popt20[1],popt21[1],popt22[1],popt23[1],popt24[1],popt25[1]]

#reson = [popt1[2],popt2[2],popt3[2],popt4[2],popt5[2],popt6[2],popt7[2],popt8[2],
reson = [popt9[2],
popt10[2],popt11[2],popt12[2],popt13[2],popt14[2],popt15[2],popt16[2],popt17[2],popt18[2],

           popt19[2],popt20[2],popt21[2],popt22[2],popt23[2],popt24[2],popt25[2]]

#temperature =  [40.5, 39.5, 36.1, 35.1, 34.1, 33, 31.9, 30.9,
temperature = [ 29.9, 29, 28, 27, 25.9, 25.5, 25.2,
               25, 24.6, 24.4, 24, 23.5]
#, 22.9, 22, 21, 20.1, 19.2]
T = temperature
for i in range(0,len(epsilon),1):
    epsilon[i]=epsilon[i]**(-1)
    T[i] = T[i]
    reson[i] = reson[i]/1000
```

Determining the susceptibility critical exponent of the liquid crystal by fitting to the power law equation:
$(\Delta\epsilon)^{-1} = At^{\gamma} + B$

In [211]:
```
def susc_func(x,A,B,gamma):
    s = A*np.power(x,gamma) + B
    return s

susc, l = spo.curve_fit(susc_func, temperature ,epsilon, p0=[1,0,1], maxfev=10000)
print "The value of Gamma is: " + str(susc[2])
print "The value of m is: " + str(susc[0])
print "The value of B is: " + str(susc[1])
fittedmodel = susc_func(temperature, susc[0],susc[1],susc[2])

plt.figure(2)
plot(T,epsilon,'r.')
plot(T,fittedmodel,'b')
xlabel('T (Celsius)')
ylabel('1/($\epsilon$)')
title('Susceptibility (TSiKN65)')
grid()
```

```
The value of Gamma is: 0.987314830011
The value of m is: 0.0135955511085
The value of B is: -0.202210014365
```





In [195]:
```
susc2, l2 = spo.curve_fit(susc_func, temperature ,reson, p0=[.000007,-.5,3.9])
print "The value of Gamma is: " + str(susc2[2])
print "The value of m is: " + str(susc2[0])
fittedmodel = susc_func(temperature, susc2[0],susc2[1],susc2[2])
```

```
The value of Gamma is: 4.10918331399
The value of m is: 3.69058550296e-06
```

Back to top