In [ ]:

```python
# Always run this block first whenever you change the bvd_library.py file, otherwise the change won't be upda
import importlib
import bvd_library
importlib.reload(bvd_library)
from bvd_library import *

# add %matplotlib widget to make plots interactive
%matplotlib widget
mpl.style.use('seaborn')
mpl.style.use('seaborn-muted')
```

First, we create a BVD_Model instance and load file. Then we plot |S11| and identify resonance.

In [ ]:

```python
bvd_model = BVD_Model()
bvd_model.load_file('B0.s2p')
bvd_model.plot_s11(figscale=0.8)
```

Figure

From the |S11| plot, we can identify the background frequency range with [fbs, fbe], and the resonance region with [frs, fre]. We first do background fitting to obtain Rs, Ls, Rp, Cp; then we do resonance fitting to obtain Rm, Cm, Lm. Procedure for fitting S11 with BVD model is:

From S11 data, we compute raw impedance using $Z_0 = Z_{char} \times \frac{1+S11}{1-S11}$, where characteristic impedance Z_{char} is taken to be 50 ohm. Raw admittance is $Y_0 = \frac{1}{Z_0}$.

We slice two flat regions out of raw impedance and concatenate them to obtain background impledance data Z_{bg}, which can be fitted using equation:

$$Z_{bg,fit} = R_s + jwL_s + \frac{1}{\frac{1}{R_p} + jwC_p}$$

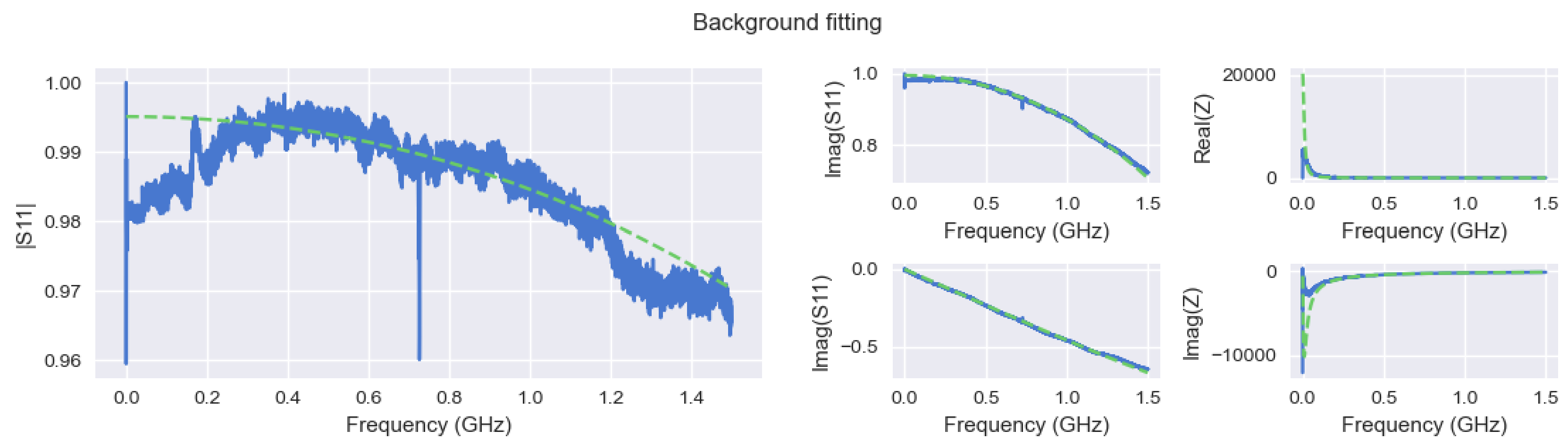
We fit Z_{bg} data with $Z_{bg,fit}$ equation using bvd_model.fit_BVD_model_background method and obtain [Rs, Ls, Rp, Cp];

In [ ]:

```python
fbs, frs, fre, fbe = 0.3e9, 0.71e9, 0.74e9, 14e9
_, _, BG_fit_params = bvd_model.fit_BVD_model_background(start1=fbs, end1=frs, start2=fre, end2=fbe,
                                                        plot_fit=True, figscale=0.8)
[Ls, Rs, Rp, Cp] = BG_fit_params
print('Ls =', Ls, 'nH;', 'Rs =', Rs, 'Ohm;', 'Rp =', Rp, 'Ohm;', 'Cp =', Cp, 'nF')
```

```
Ls = 1.83377519411899 nH; Rs = 4.542031472646649 Ohm; Rp = 20274.471182348385 Ohm; Cp = 0.0007394531193844545
nF
```

Figure



Using the fitted [Ls, Rs, Rp, Cp], we de-background raw admittance data Y_0 to obtain:

$$Y_a = \frac{1}{Z_0 - R_s - jwL_s} - (\frac{1}{R_p} + jwC_p)$$

The admittance data here can be modeled with the RLC circuit, which can be fitted with equation:

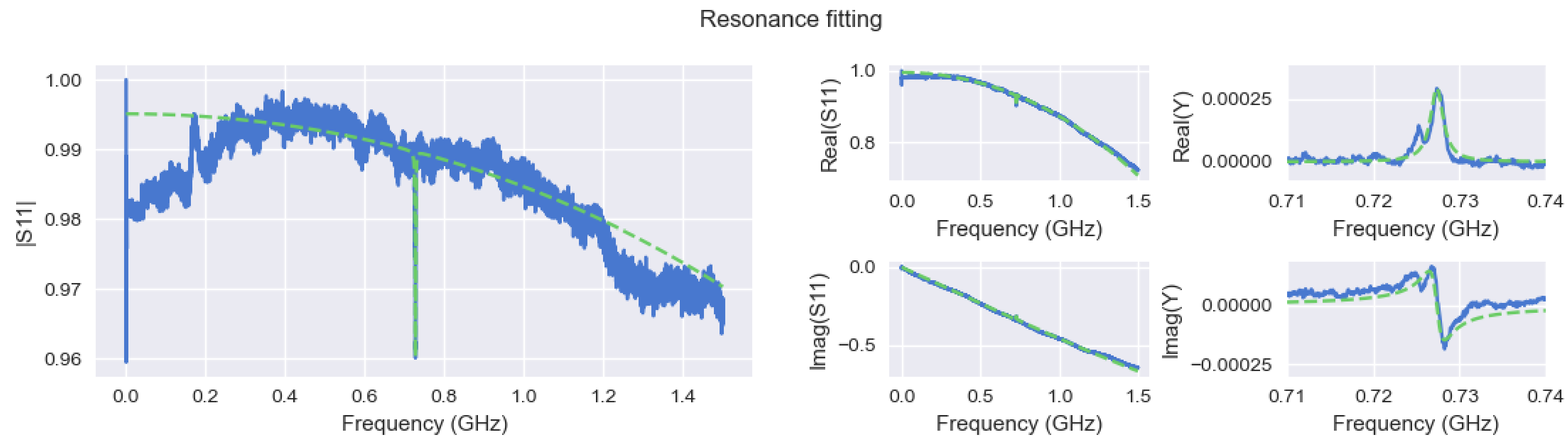
$$Y_{a,fit} = \frac{1}{R_m + jwL_m + \frac{1}{jwC_m}}$$

We fit Y_a data with $Y_{a,fit}$ using bvd_model.fit_BVD_model_resonances method, and obtain [Rm, Lm, Cm].

In [ ]:
```
s11_fit, Z_total, popt, Y_list = bvd_model.fit_BVD_model_resonances(start1=fbs, end1=frs, start2=fre, end2=fb
                                                                  bg_params=BG_fit_params, RS_fit_params_manual=Non
                                                                  plot_fit=True, figscale=0.8)
[Lm, Cm, Rm] = popt
print('Lm =', Lm, 'nH;', 'Cm =', Cm, 'nF;', 'Rm =', Rm, 'Ohm')
```

```
Lm = 301155.6550772744 nH; Cm = 1.58985277318971e-07 nF; Rm = 3403.2052449051102 Ohm
```

Figure



More details about fitting:

When fitting complex numbers, we need to seperate real & and imaginary parts and stack them so that the optimizer will optimize the least square of $|Re(F_{fit}) - Re(F_{data})| + |Im(F_{fit}) - Im(F_{data})|$.

Thus, for bvd_model.fit_BVD_model_background method, we can write:

$$Re(Z_{bg,fit}) = Rs + \frac{R_p}{1 + (wC_pR_p)^2}$$

$$Im(Z_{bg,fit}) = wL_s - \frac{wR_p^2C_p}{1 + (wR_pC_p)^2}$$

which is implemented in Z_background function.

For bvd_model.fit_BVD_model_resonances method: we can write:

$$Re(Y_{a,fit}) = \frac{R_m}{R_m^2 + (wL_m - \frac{1}{wC_m})}$$

$$Im(Y_{a,fit}) = -\frac{wL_m - \frac{1}{wC_m}}{R_m^2 + (wL_m - \frac{1}{wC_m})}$$

which is implemented in Y_resonance function.

When fitting the resonance, a key step is to find good initial guesses. For RLC circuit, at resonance frequencies w_r (peak of real admittance) and w_l (dip of imaginary admittance), we can compute:

$$R_{m0} = \frac{1}{Y_a[w_r]}$$

$$L_{m0} = \frac{R_{m0} * w_l}{w_l^2 - w_r^2}$$

$$C_{m0} = \frac{1}{L_{m0} * w_r^2}$$

These initial guesses [Lm0, Cm0, Rm0] will be fed into the curve_fit optimizer to assist convergence.

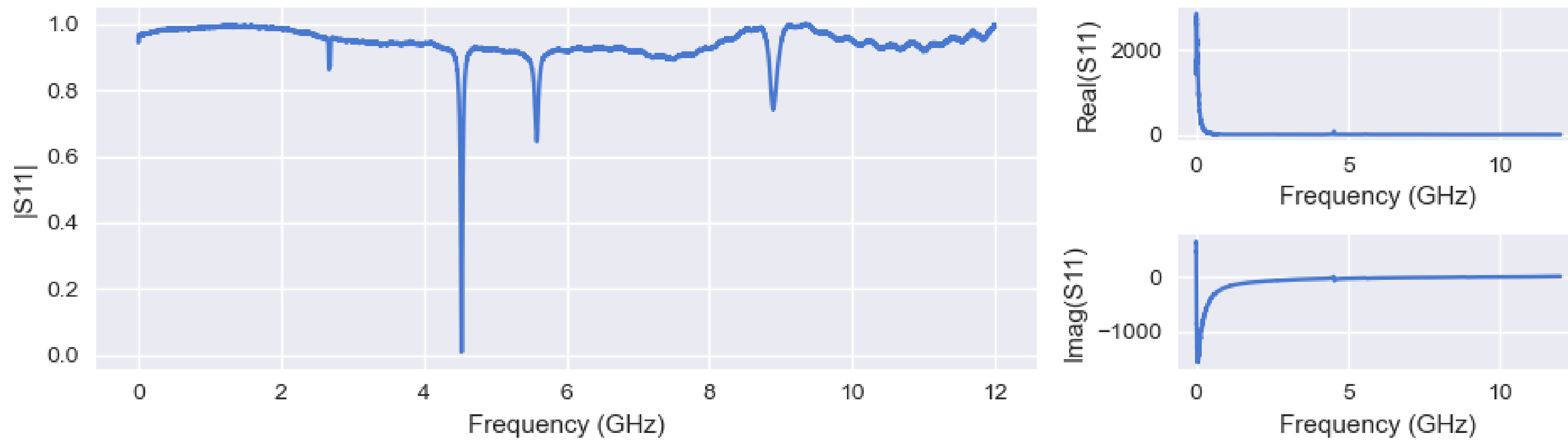
Loading .prn file:

We can also load .prn file. However, this will likely prompt you to install a 'matlabengin' libray, which can be done with "python pip install matlabengine" cammand. It also requires Matlab is already installed on the computer.

In [ ]:
```
bvd_model.load_file('4K-2-S11-pol.prn')
bvd_model.plot_s11(figscale=0.8)
```

Figure

Raw S11 data



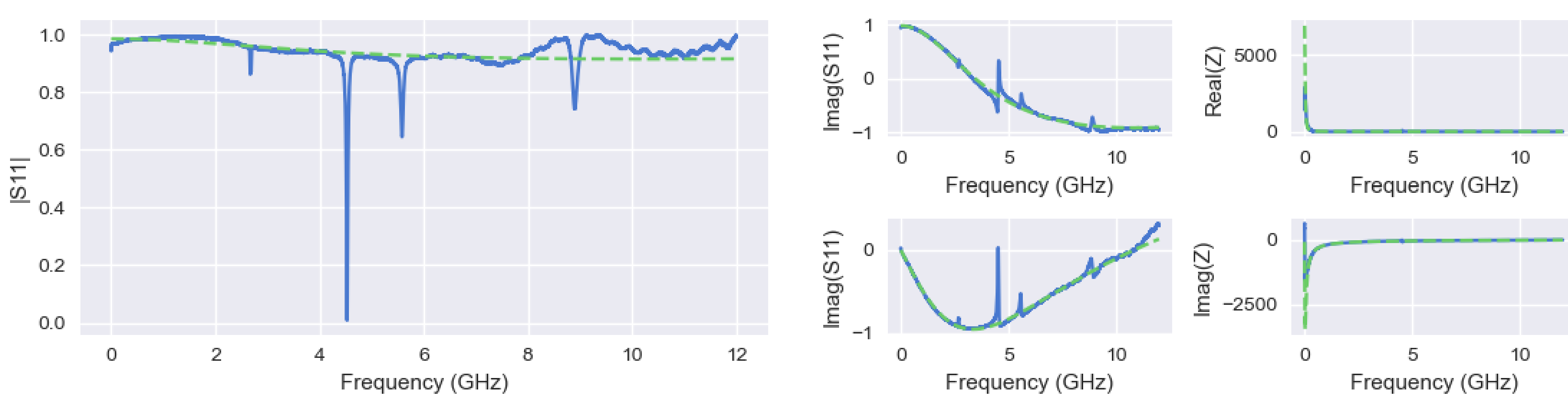
From the |S11| plot, we set [fbs, frs, fre, fbe] as before.

```
In [ ]: fbs, frs, fre, fbe = 2e9, 4.3e9, 4.7e9, 10e9
_, _, BG_fit_params = bvd_model.fit_BVD_model_background(start1=fbs, end1=frs, start2=fre, end2=fbe,
                                                          plot_fit=True, figscale=0.8)
[Ls, Rs, Rp, Cp] = BG_fit_params
print('Ls =', Ls, 'nH;', 'Rs =', Rs, 'Ohm;', 'Rp =', Rp, 'Ohm;', 'Cp =', Cp, 'nF')
s11_fit, Z_total, popt, Y_list = bvd_model.fit_BVD_model_resonances(start1=fbs, end1=frs, start2=fre, end2=fb
                                                            bg_params=BG_fit_params, RS_fit_params_manual=Non
                                                            plot_fit=True, figscale=0.8)
[Lm, Cm, Rm] = popt
print('Lm =', Lm, 'nH;', 'Cm =', Cm, 'nF;', 'Rm =', Rm, 'Ohm')
```

```
Ls = 0.2609268708011194 nH; Rs = 2.164788916496168 Ohm; Rp = 6893.990305717491 Ohm; Cp = 0.00082667040865783
nF
Lm = 145.48679520207793 nH; Cm = 8.535864579693588e-06 nF; Rm = 24.32589033819095 Ohm
```

Figure

Background fitting



Figure

Resonance fitting