Octagon

November 10, 2024

[1]:
```
import skrf
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline

import warnings
import glob

warnings.filterwarnings("ignore", category=RuntimeWarning)

plt.rcParams["figure.figsize"] = [12,10]
```

[2]:
```
# calculate L series from ABCD

#  P1p ---- inductor ---- P2p
#
#  P1n ------------------ P2n

# ABCD = (A_00 A_01)
#        (A_10 A_11)

def calculate_series_inductance(n):
    L = np.imag( net.a[:,0,1] ) / (2*np.pi*net.f)
    return net.f, L


# calculate L differentially

def calculate_diff_inductance(n):
    dnet = n.copy()
    dnet.se2gmm(1) # convert net to mixed mode (0=D, 1=C)
    L = np.imag( 1 / dnet.y[:,0,0] ) / (2*np.pi*dnet.f)
    return dnet.f, L
```

1 Octagon C4

- loops = 1

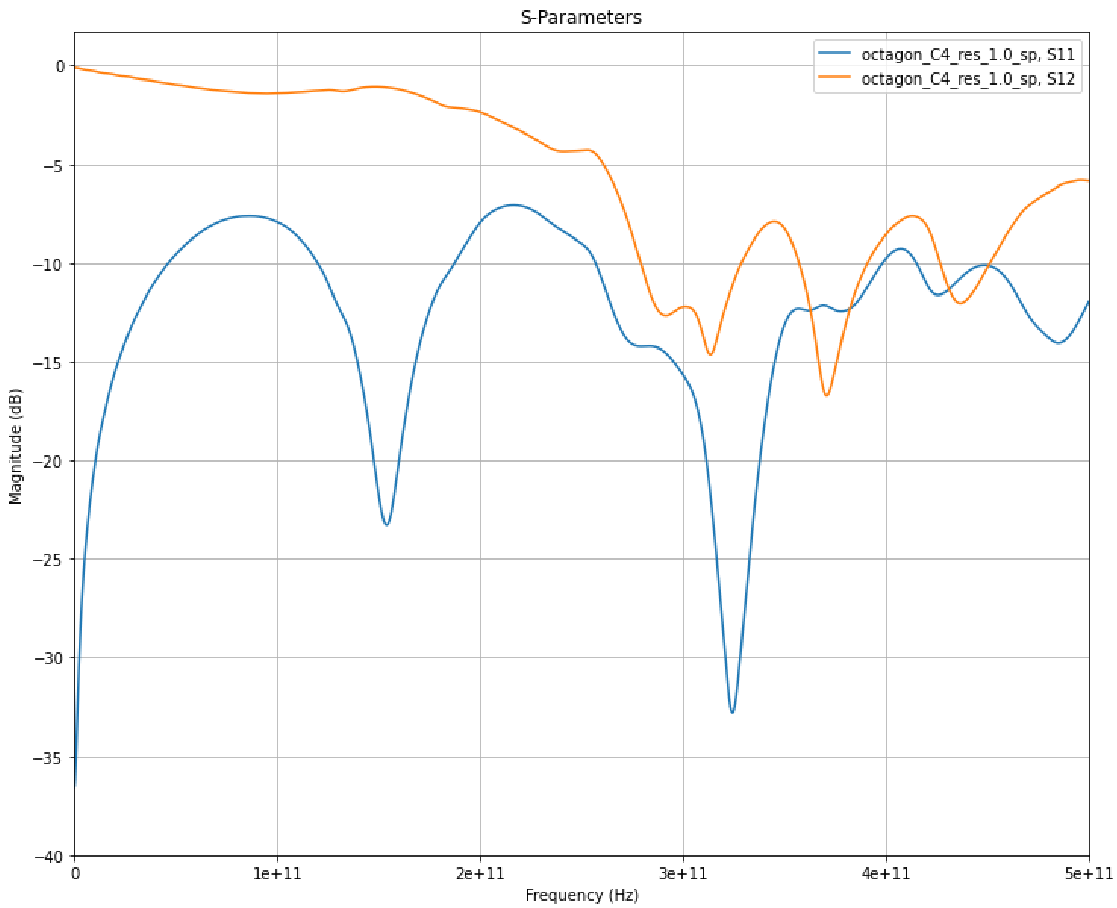
- symmetrical simulation

[3]:
```
net = skrf.Network("octagon_C4_res_1.0_sp.s2p")
```

[4]:
```
print(net)
```

```
2-Port Network: 'octagon_C4_res_1.0_sp',  0.0-500000000000.0 Hz, 20000 pts,
z0=[50.+0.j 50.+0.j]
```

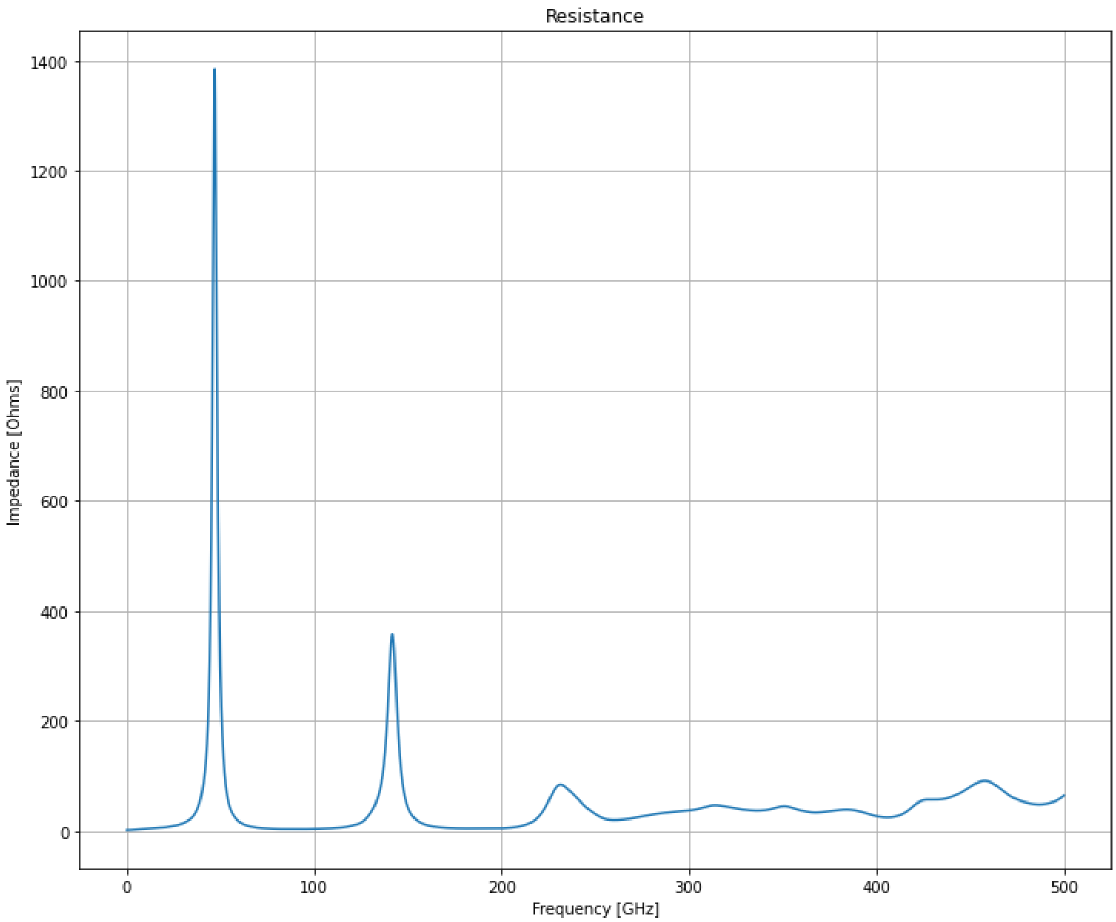
[5]:
```
net.plot_s_db(m=0)
plt.grid()
plt.ylim(bottom=-40)
plt.title("S-Parameters")
plt.show()
```



[6]:
```
f = net.f
R = np.real( 1 / net.y[:,0,0] )

plt.title("Resistance")
```

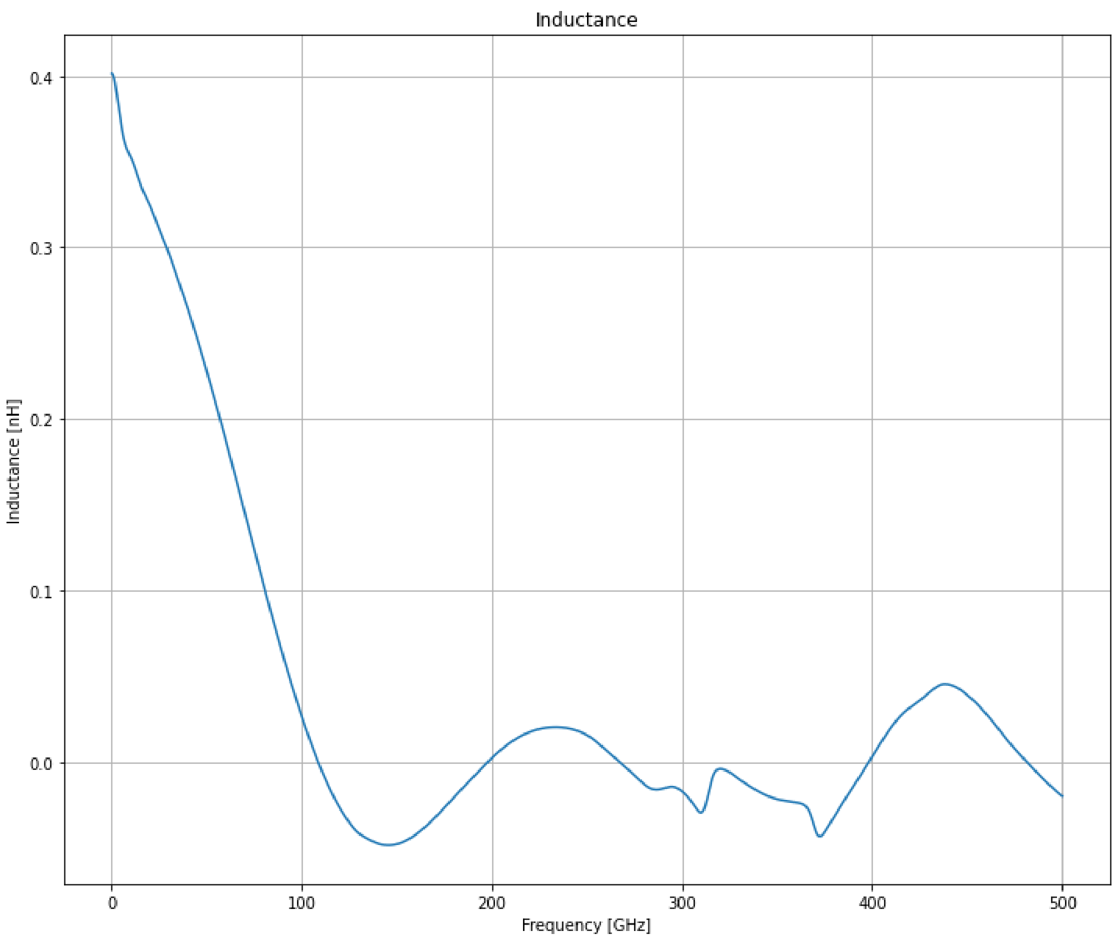
```python
plt.xlabel("Frequency [GHz]")
plt.ylabel("Impedance [Ohms]")
plt.grid()
_ = plt.plot(f/1e9, R)
```



[7]:

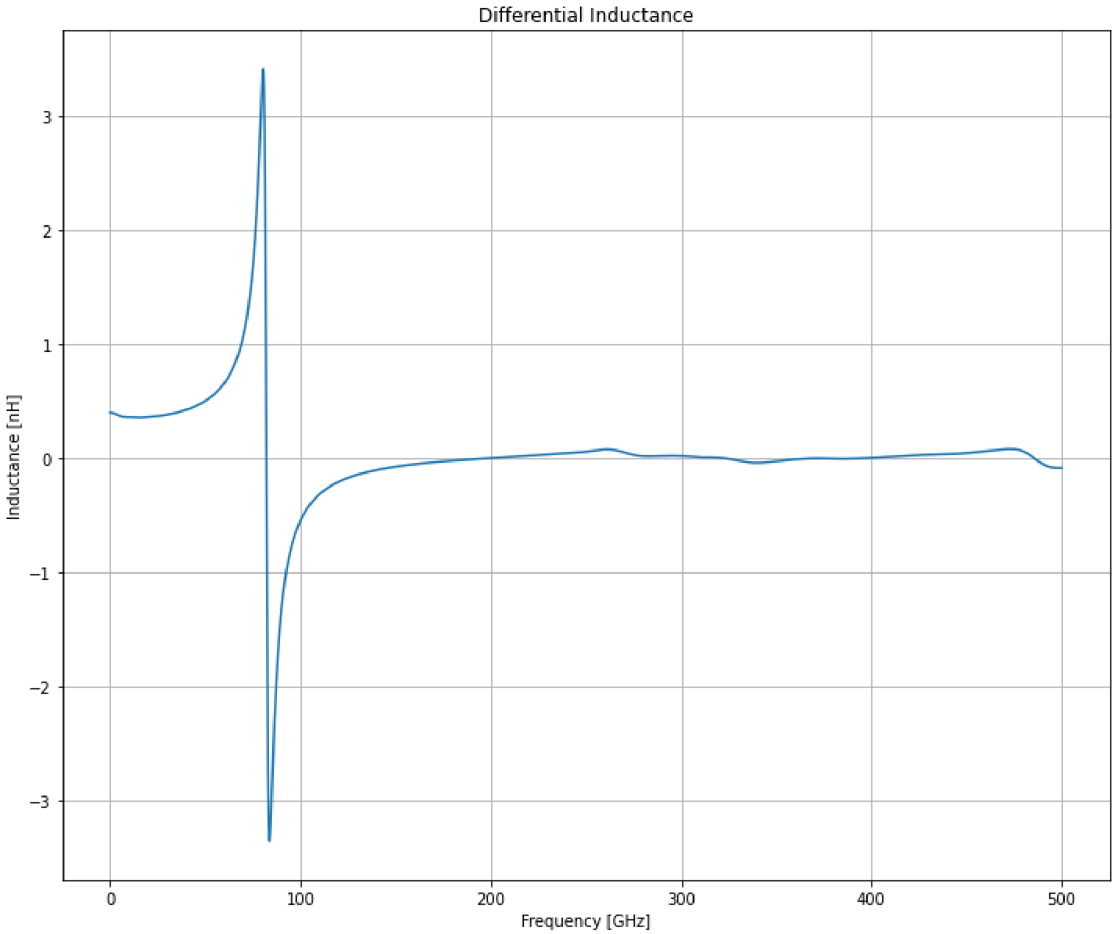
```python
f, L = calculate_series_inductance(net)

plt.title("Inductance")
plt.xlabel("Frequency [GHz]")
plt.ylabel("Inductance [nH]")
plt.grid()
_ = plt.plot(f/1e9, L*1e9)
```



[8]:
```
f, L = calculate_diff_inductance(net)

plt.title("Differential Inductance")
plt.xlabel("Frequency [GHz]")
plt.ylabel("Inductance [nH]")
plt.grid()
_ = plt.plot(f/1e9, L*1e9)
```



2 Octagon C6

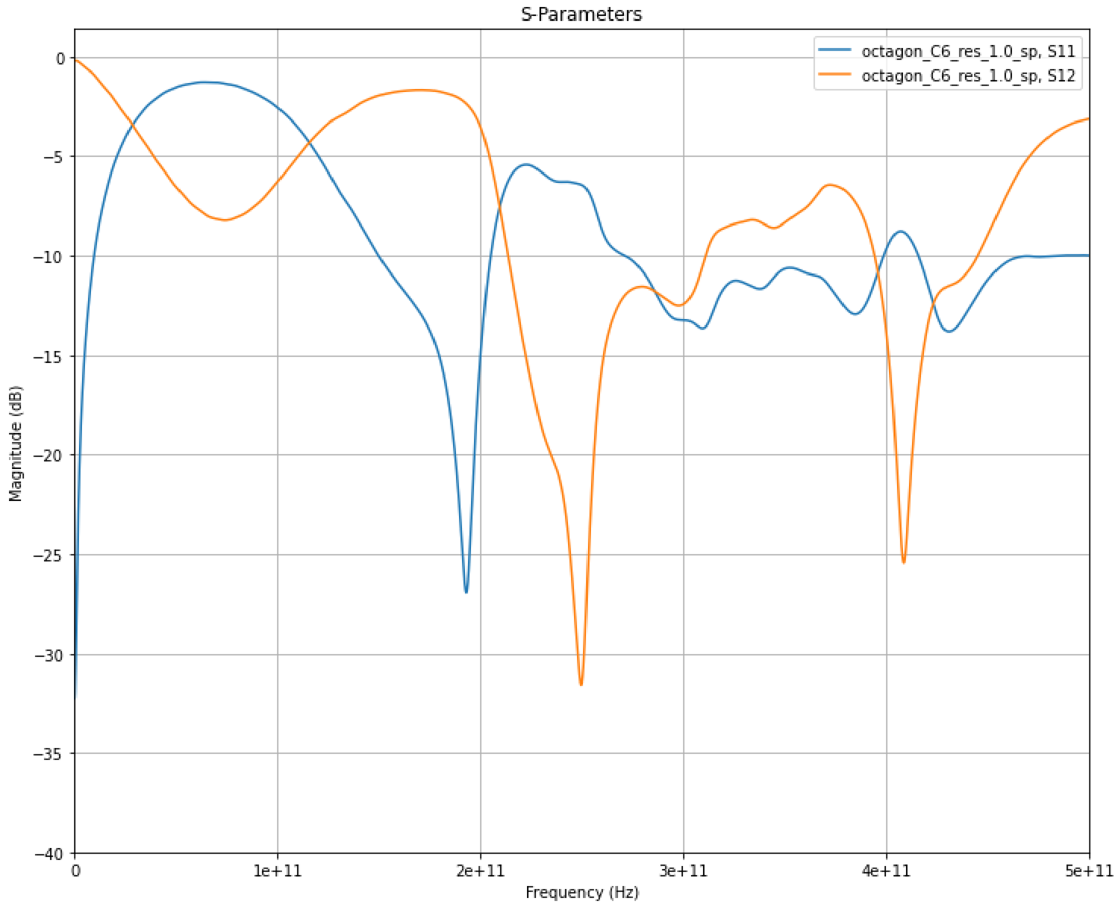
- loops = 2
- symmetrical simulation

[9]:
```
net = skrf.Network("octagon_C6_res_1.0_sp.s2p")
```

[10]:
```
print(net)
```

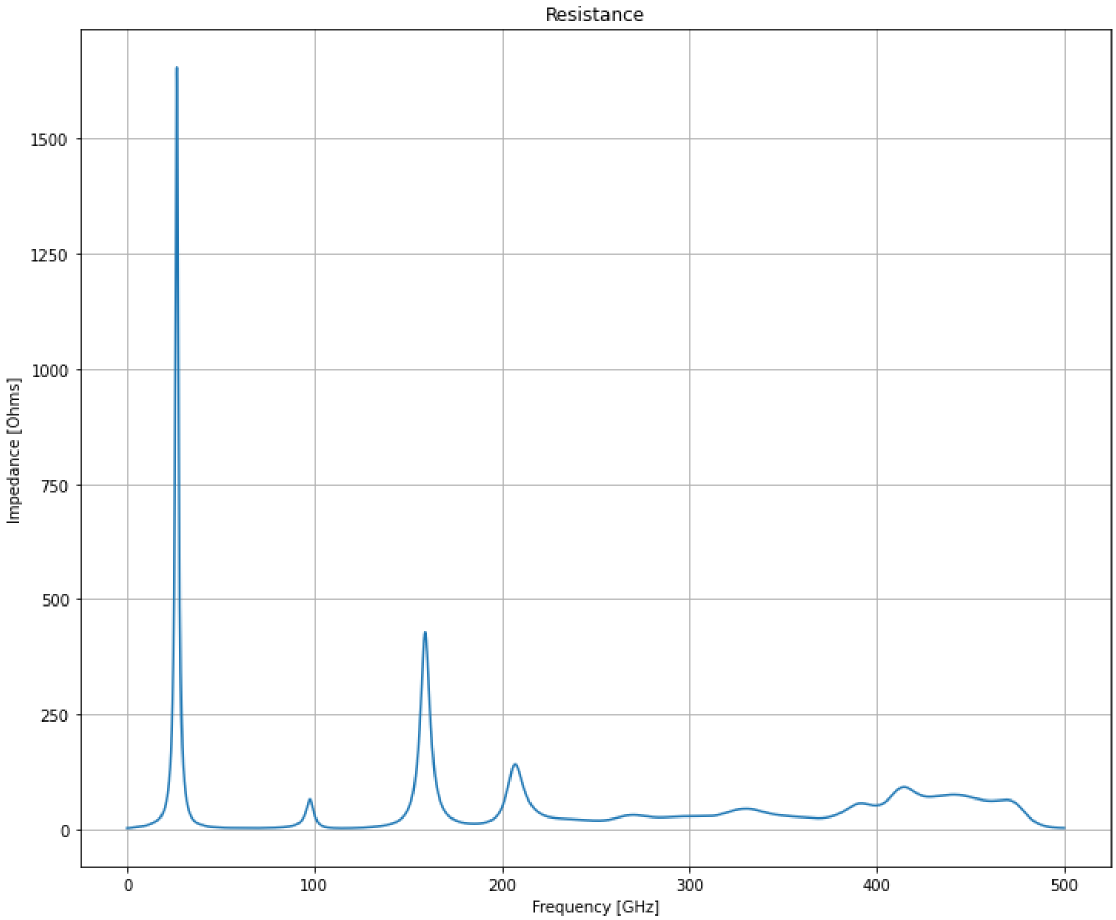
```
2-Port Network: 'octagon_C6_res_1.0_sp',  0.0-500000000000.0 Hz, 20000 pts,
z0=[50.+0.j 50.+0.j]
```

[11]:
```
net.plot_s_db(m=0)
plt.grid()
plt.ylim(bottom=-40)
plt.title("S-Parameters")
plt.show()
```



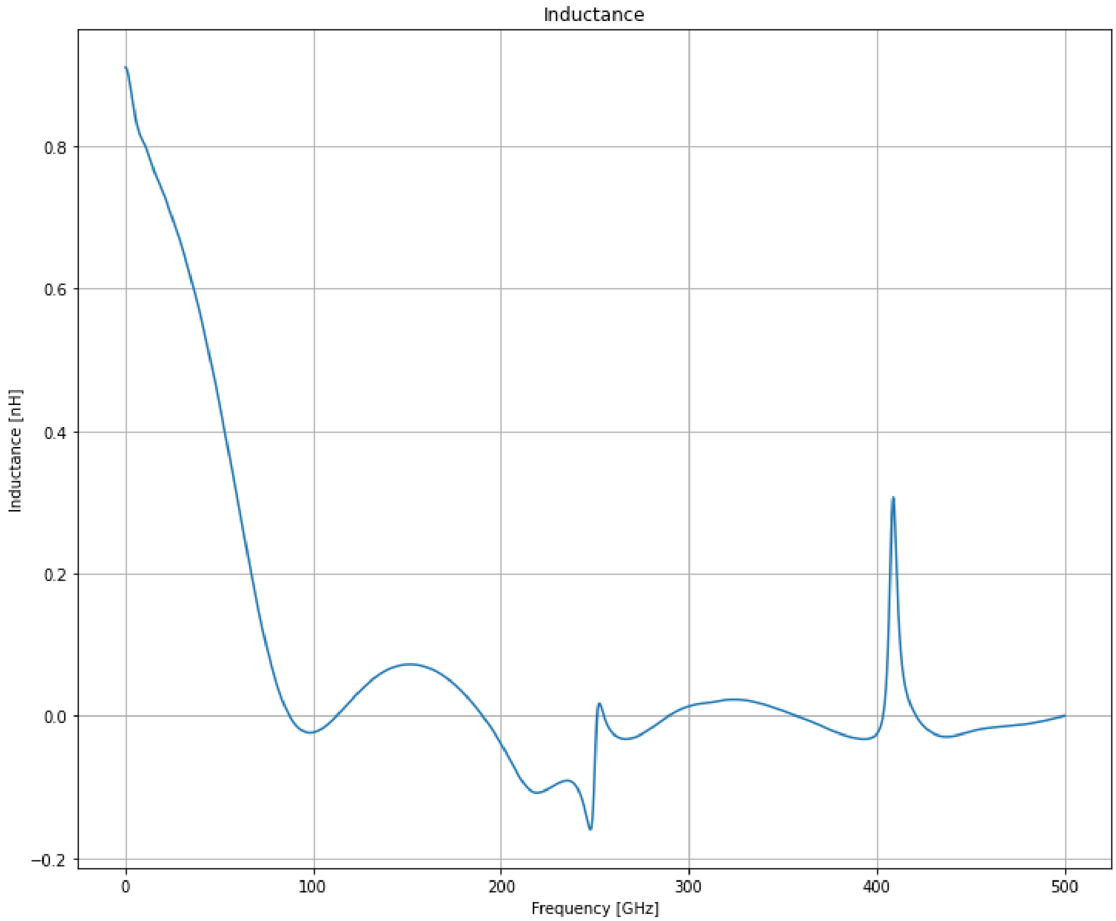
[12]:
```
f = net.f
R = np.real( 1 / net.y[:,0,0] )

plt.title("Resistance")
plt.xlabel("Frequency [GHz]")
plt.ylabel("Impedance [Ohms]")
plt.grid()
_ = plt.plot(f/1e9, R)
```



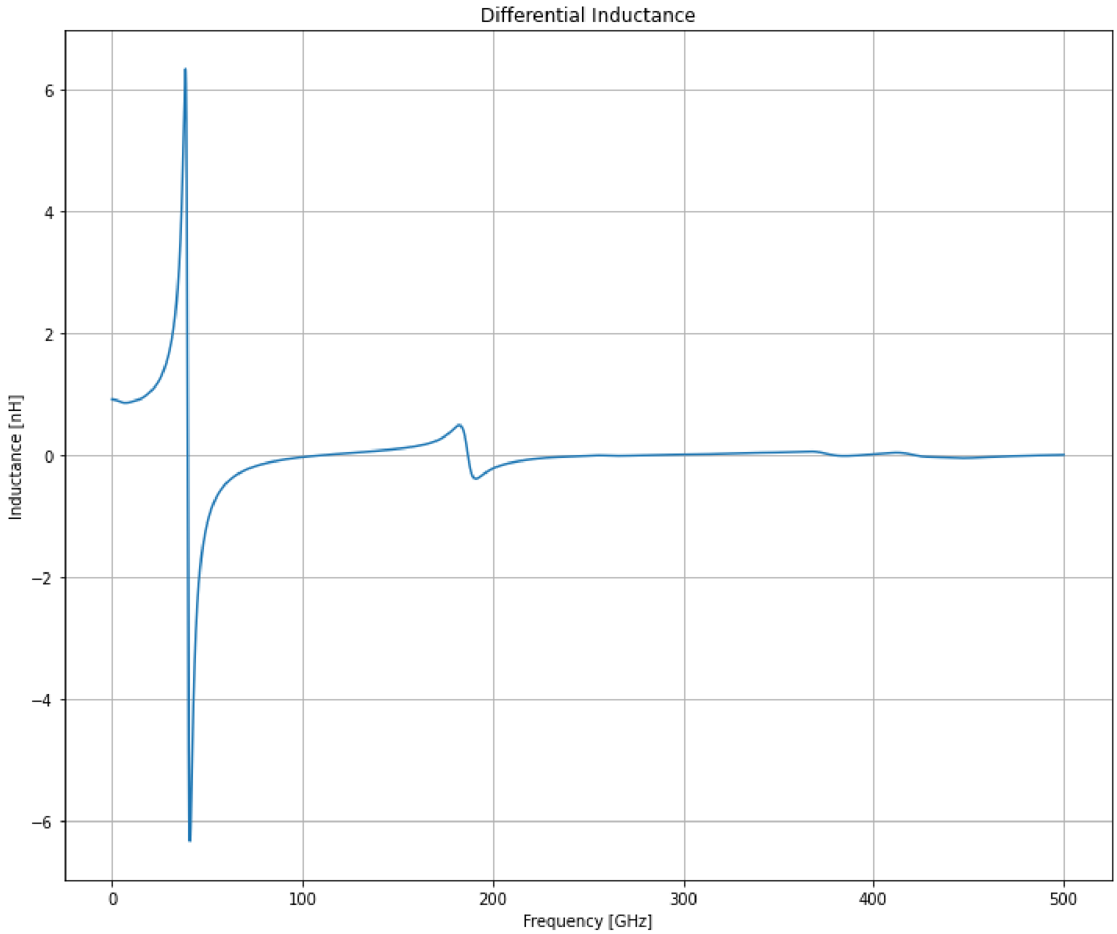
[13]:
```
f, L = calculate_series_inductance(net)

plt.title("Inductance")
plt.xlabel("Frequency [GHz]")
plt.ylabel("Inductance [nH]")
plt.grid()
_ = plt.plot(f/1e9, L*1e9)
```



[14]:
```
f, L = calculate_diff_inductance(net)

plt.title("Differential Inductance")
plt.xlabel("Frequency [GHz]")
plt.ylabel("Inductance [nH]")
plt.grid()
_ = plt.plot(f/1e9, L*1e9)
```



3 Octagon C8

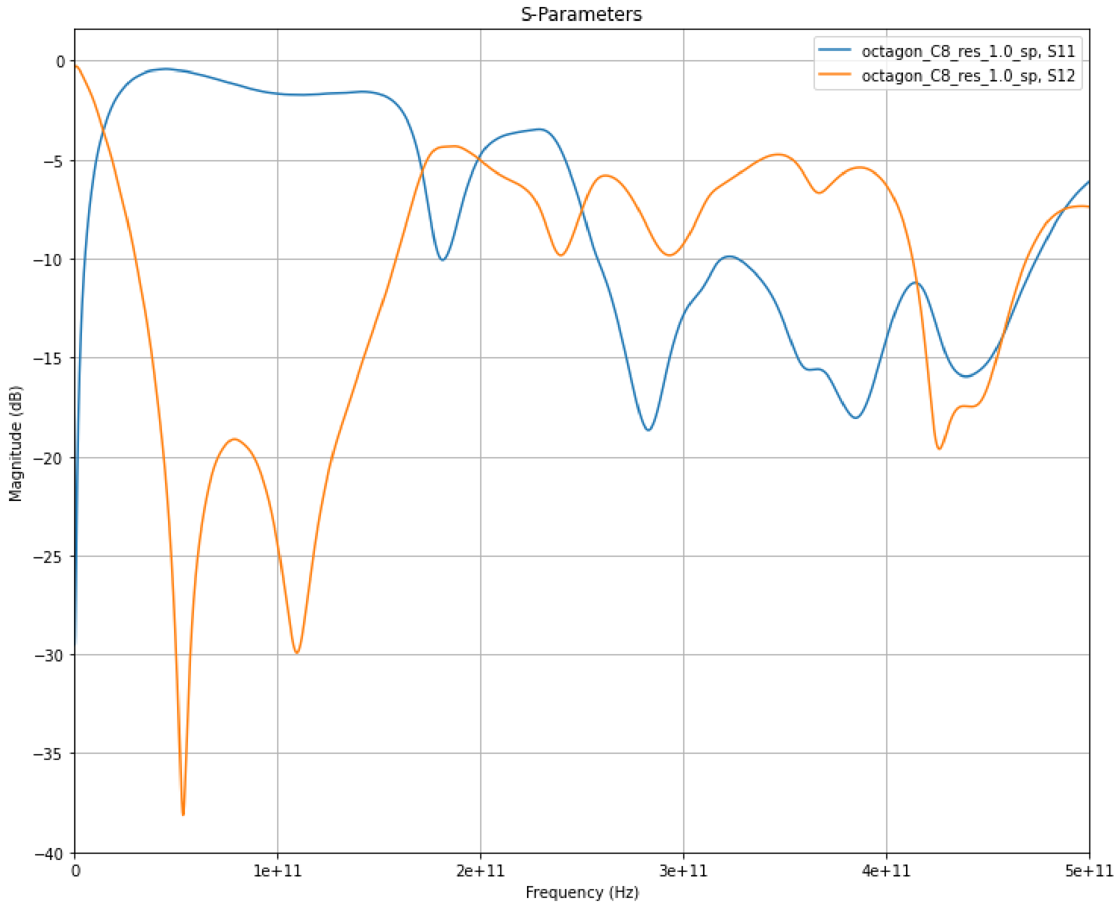
- loops = 3
- symmetrical simulation

[15]:
```
net = skrf.Network("octagon_C8_res_1.0_sp.s2p")
```

[16]:
```
print(net)
```

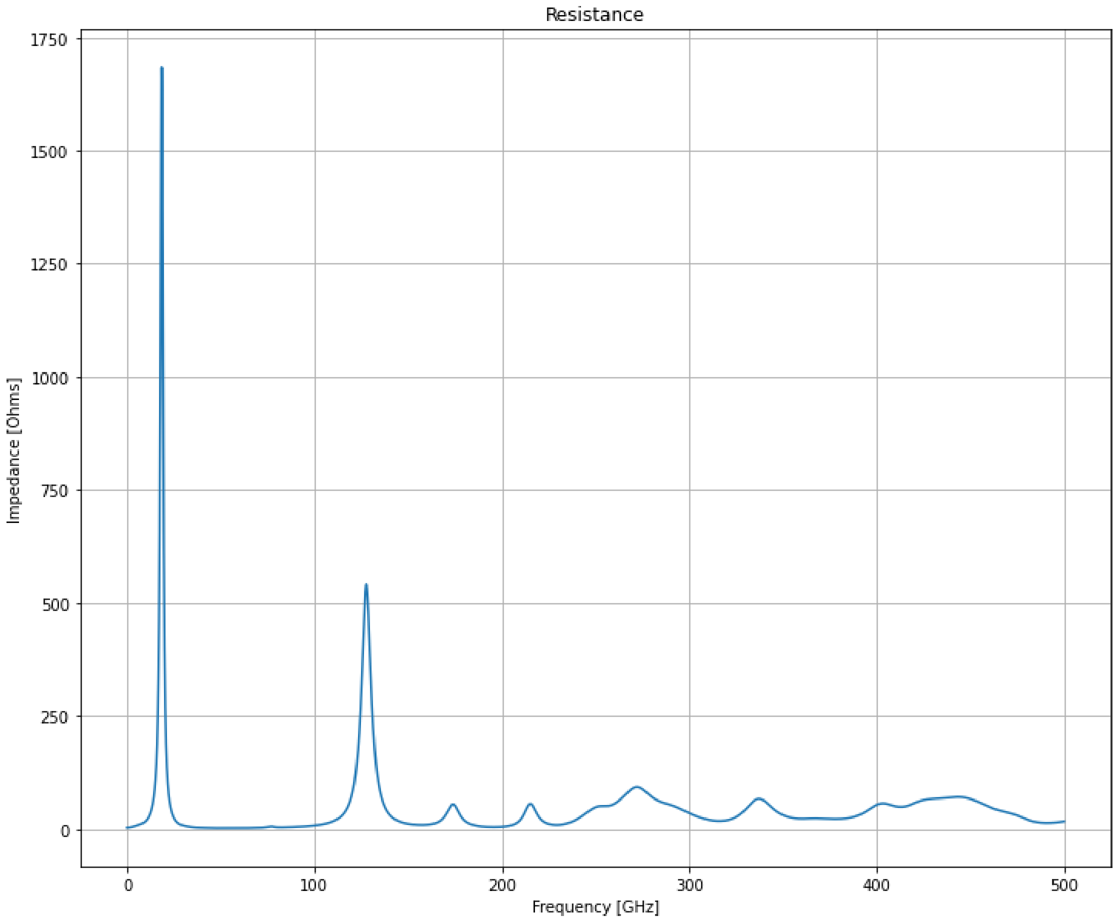
```
2-Port Network: 'octagon_C8_res_1.0_sp',  0.0-500000000000.0 Hz, 20000 pts,
z0=[50.+0.j 50.+0.j]
```

[17]:
```
net.plot_s_db(m=0)
plt.grid()
plt.ylim(bottom=-40)
plt.title("S-Parameters")
plt.show()
```



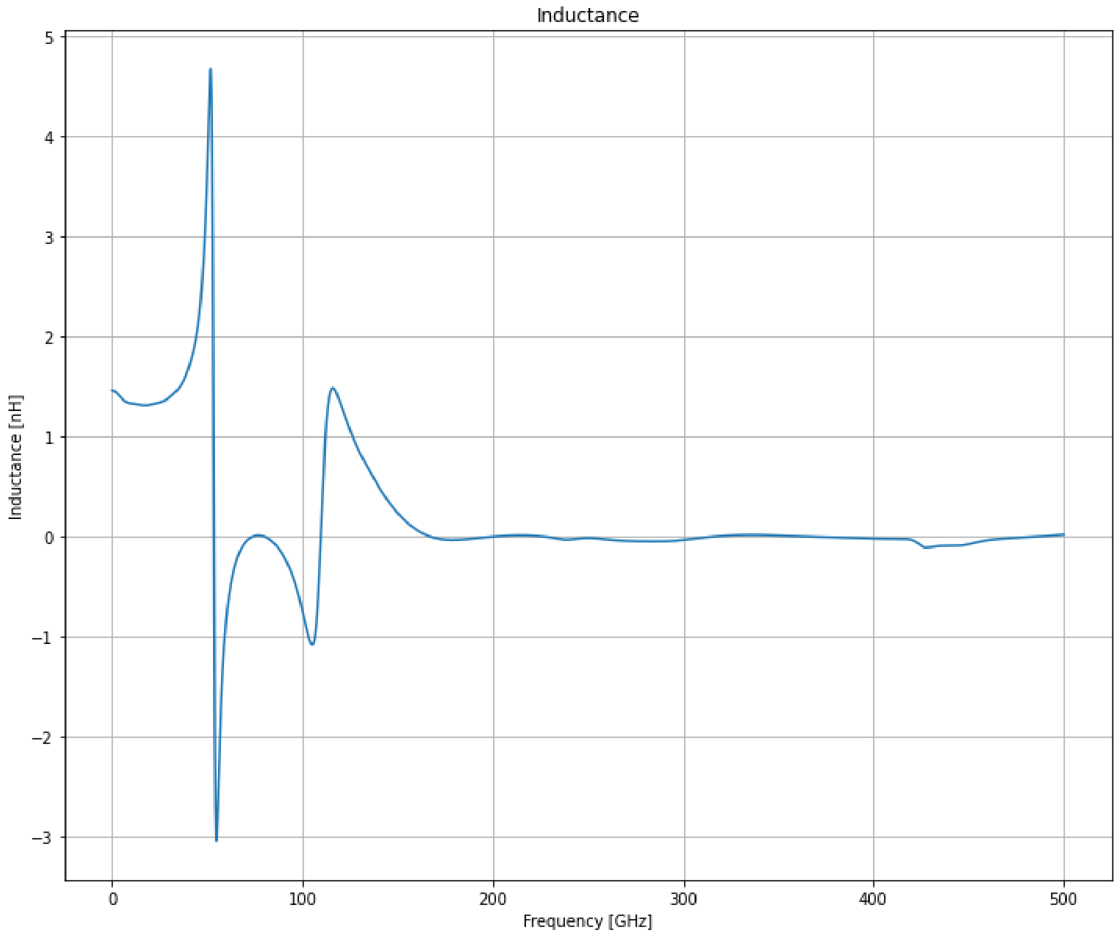
[18]:
```
f = net.f
R = np.real( 1 / net.y[:,0,0] )

plt.title("Resistance")
plt.xlabel("Frequency [GHz]")
plt.ylabel("Impedance [Ohms]")
plt.grid()
_ = plt.plot(f/1e9, R)
```



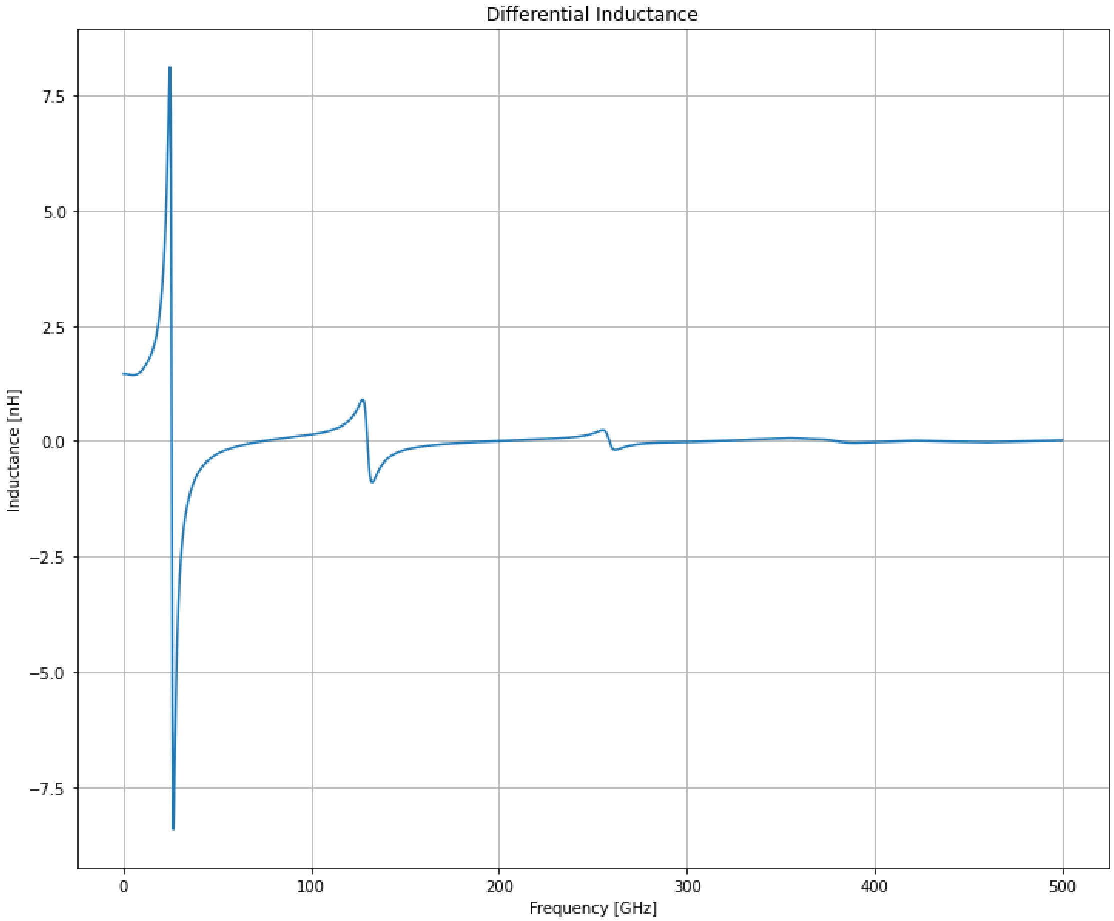
[19]:
```
f, L = calculate_series_inductance(net)

plt.title("Inductance")
plt.xlabel("Frequency [GHz]")
plt.ylabel("Inductance [nH]")
plt.grid()
_ = plt.plot(f/1e9, L*1e9)
```



[20]:
```
f, L = calculate_diff_inductance(net)

plt.title("Differential Inductance")
plt.xlabel("Frequency [GHz]")
plt.ylabel("Inductance [nH]")
plt.grid()
_ = plt.plot(f/1e9, L*1e9)
```


Differential Inductance
Inductance [nH]
Frequency [GHz]
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
0
100
200
300
400
500