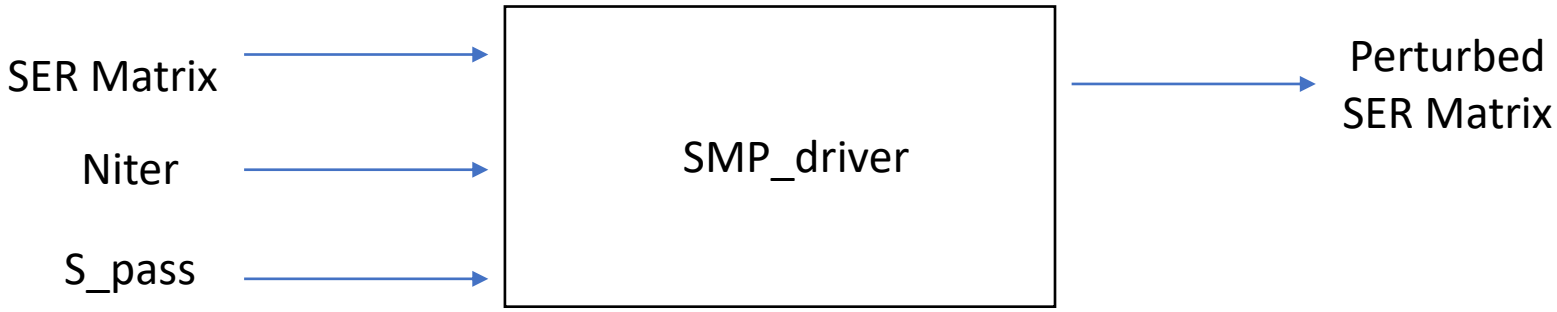
# Initialization and I/O

1. Initialize SMP object
2. Call SMP_driver method to perform passivity assessment and enforcement
   I. SER is in dictionary structure containing A,B,C,D matrices
   II. Niter is the max number of perturbation rounds attempted
   III. S_pass is the frequency range for passivity assessment and enforcement

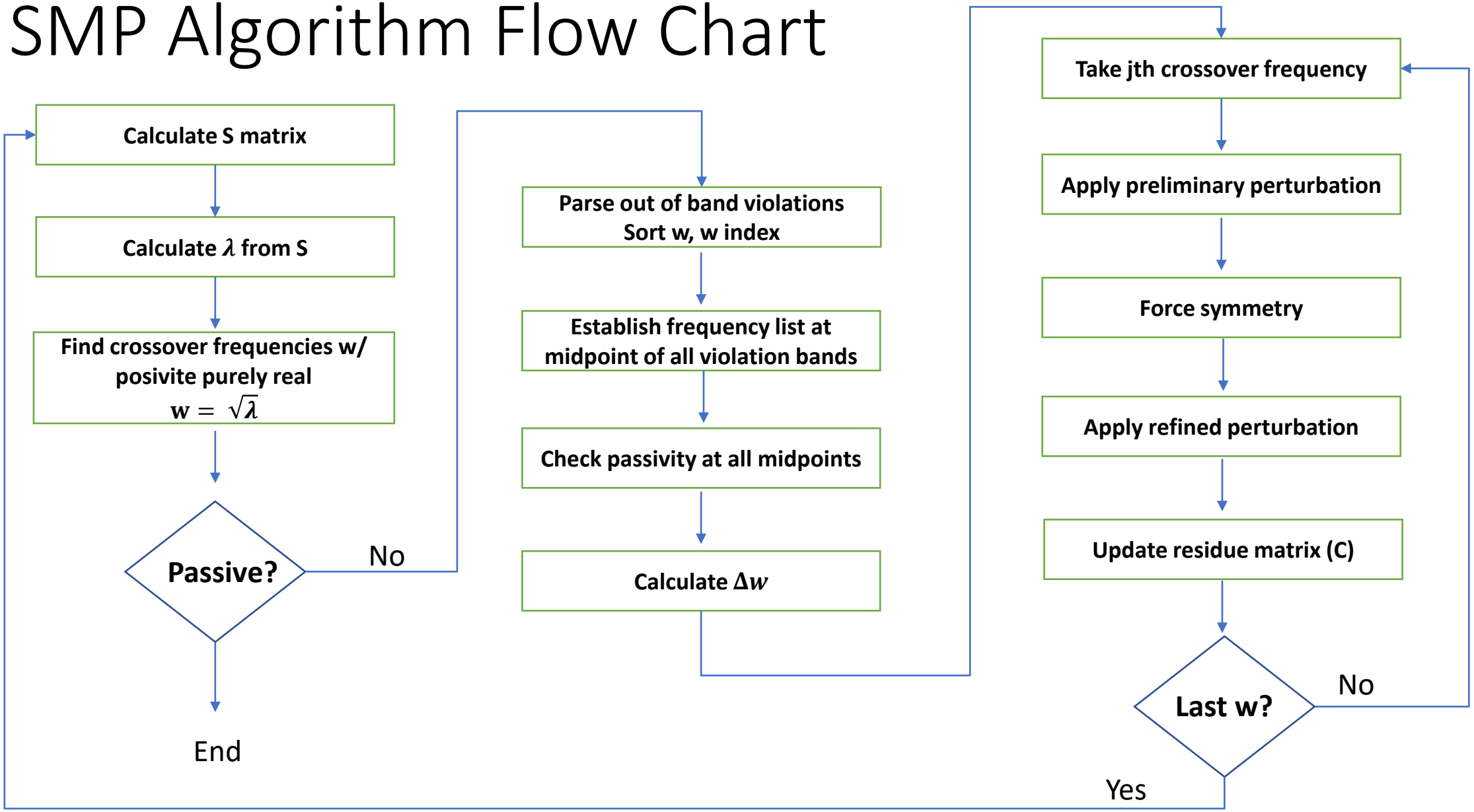
```
smp = SMP()
new_SER = smp.SMP_driver(SER, Niter=5,
s_pass=2*np.pi*1j*np.linspace(0, 2e5, 1001).T)
```

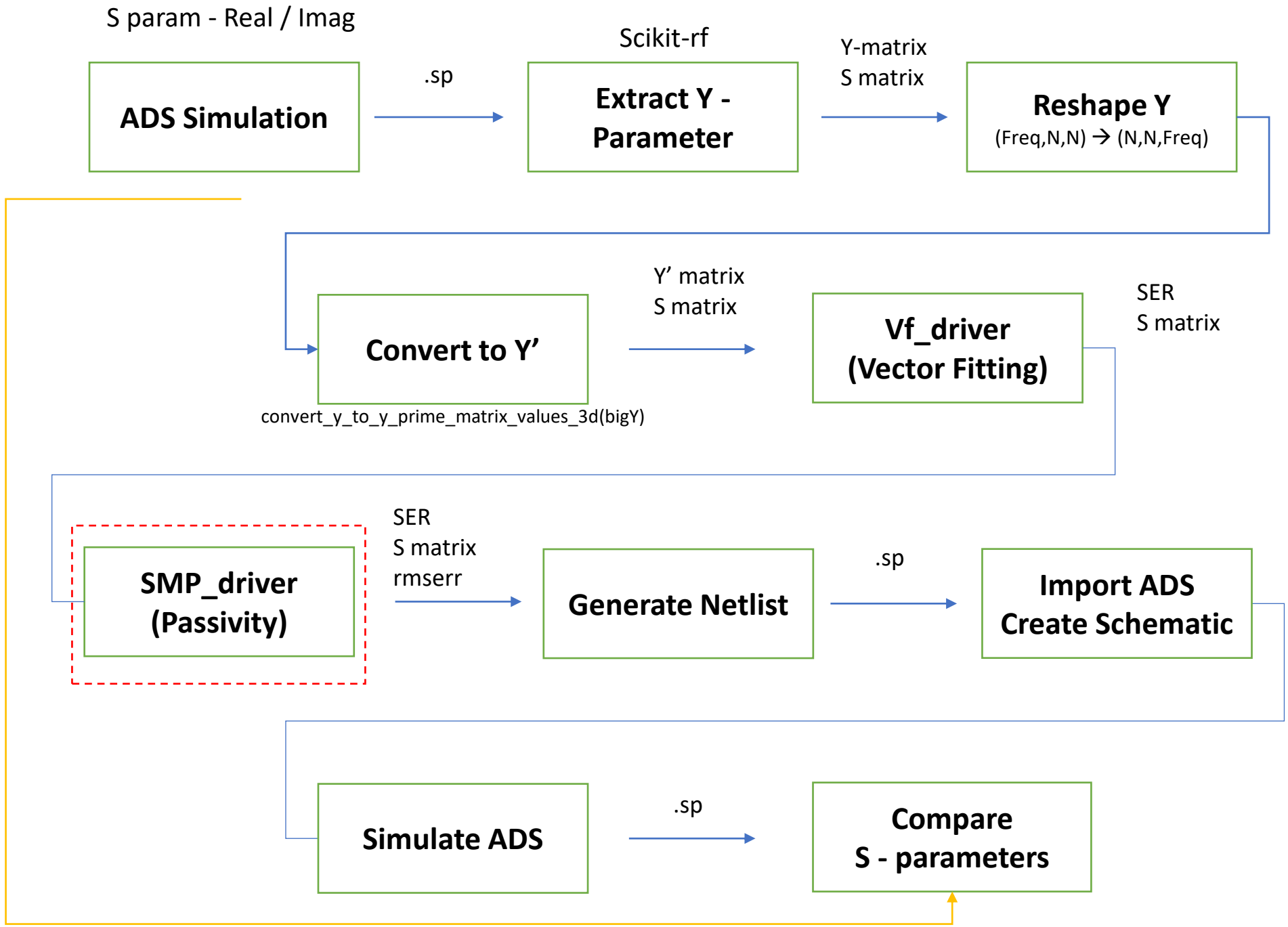



| Element | Dimension |
|---|---|
| A | (Nc*N) x (Nc*N) |
| B | (Nc*N) x Nc |
| C | Nc x (Nc*N) |
| D | Nc x Nc |

| | |
|---|---|
| Nc | Number of ports |
| N | Number of poles |

# SMP Algorithm Flow Chart




S param - Real / Imag
ADS Simulation
.sp
Scikit-rf
Extract Y - Parameter
Y-matrix
S matrix
Reshape Y
(Freq,N,N) → (N,N,Freq)
Convert to Y'
convert_y_to_y_prime_matrix_values_3d(bigY)
Y' matrix
S matrix
Vf_driver (Vector Fitting)
SER
S matrix
SMP_driver (Passivity)
SER
S matrix
rmserr
Generate Netlist
.sp
Import ADS Create Schematic
Simulate ADS
.sp
Compare S - parameters