```
clear
%close all

g.finger.l = 24;  %length of a capacitor 'finger', um add a bit to represent
fringe effects
g.finger.w = 0.25; %.25;  %width of the microstrip line (and fingers)
g.finger.p = 1;    %period of the fingers
g.ms.w = g.finger.w;
g.ms.h = 0.15; %dielectric thickness
g.ms.eps = 10.3;  %dielectric epsilon
g.ms.eps_upper = 10.4;
g.ms.t1 = 0.030;  %microstrip conductor metal thickness
g.ms.t2 = 0.200;    %ground plane thickness

%modulation parameters
g.finger.modperiod = 50;
g.finger.modamp = 4;

L = g.finger.modperiod*500; % length in microns

filename = '0714TWPaX.mat';

%loop over frequencies
f = linspace( .01, 50.01, 20001)*1e9;

%resistivities to consider
rhon = 480.*1e-8; %film resistivity, ohm.m
sn = 1 / rhon;  %normal state conductivity

%loop over position
ms = g.ms;

ev = 1.6e-19;  %J/ev
mu0 = pi*4e-7; %H/m
omega = 2*pi*f;
hbo = 1e-34 * omega / ev;

T = 0.01;  %operating temperature, K
kbt = 1.38e-23 * T / ev;

Tc = 14.5;    %device Tc
gap = 0.5 * 3.5 * Tc * 1.38e-23 / ev;   %device gap parameter

lambda = zeros( 1, numel(f));
Lperm = lambda;  Cperm = lambda; Z0 = lambda;

z = 1:1:round(L/g.finger.modperiod/2);
zmax = z(end);
```

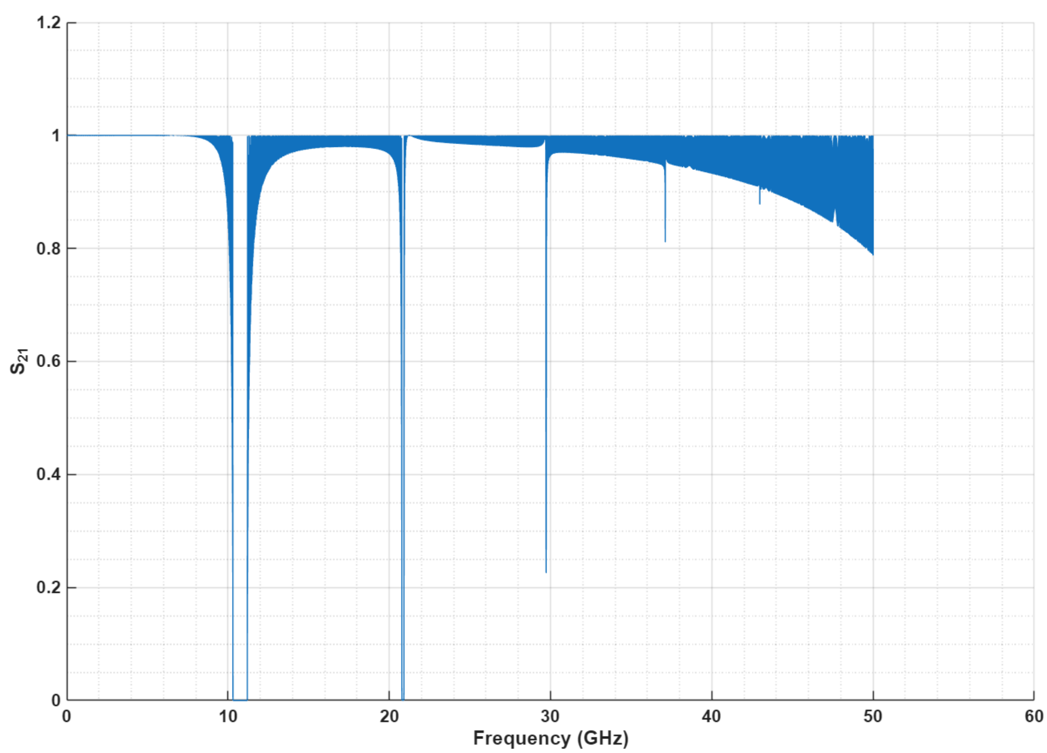
Calculate or Load Resistivity-based Z,vph

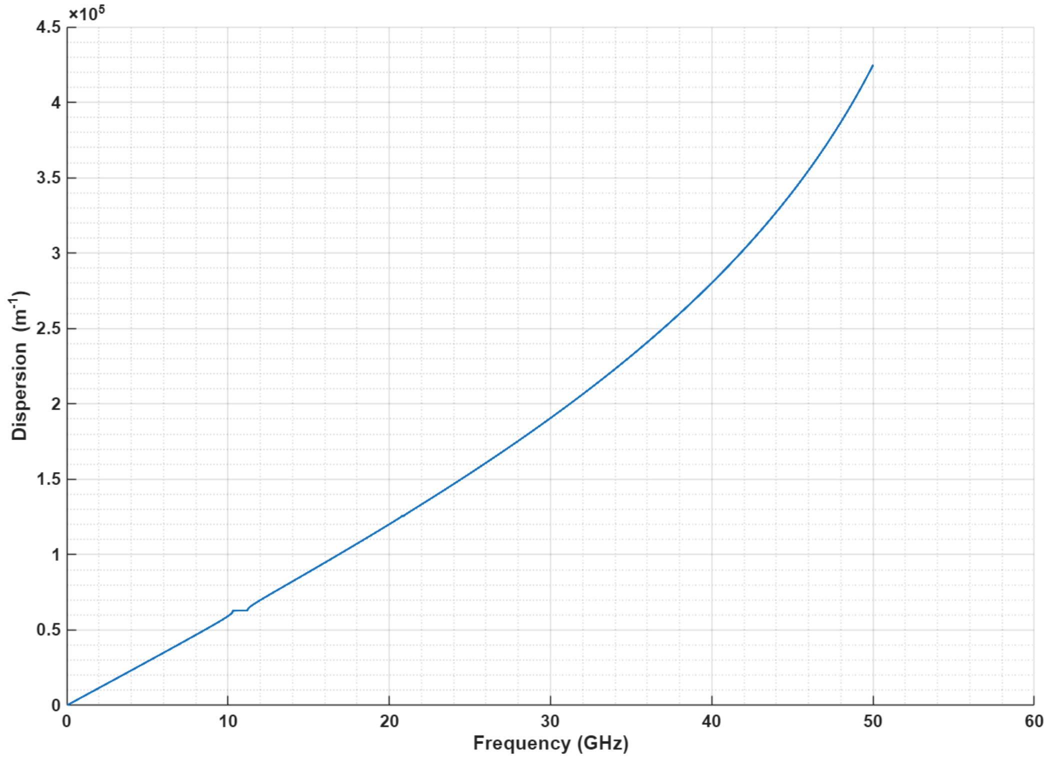
```
%calculate array of lambdas

for ii = 1:numel(f)

    s2 = mb2(gap, kbt, hbo(ii)) * sn;       %Mattis-Bardeen parameter
    lambda(ii) = 1 / sqrt(mu0 * s2 * omega(ii));

    [Lperm(ii), Cperm(ii), Z0(ii), ~, ~, ~] = mstrip_sc_Ls( ms.h, ms.w,
ms.t1, ms.t2, ...
        ms.eps, ms.eps_upper, 10, 1e9, 1e9, lambda(ii)*1e6, lambda(ii)*1e6);
end

vph = 1 ./ sqrt(Lperm .* Cperm);  %m/s
IcOverIstar = 0.15;
vph = vph./sqrt(1+IcOverIstar^2+IcOverIstar^4);

Dfinger = g.finger.p * 1e-6; %m, distance between fingers
Lfinger = g.finger.l * 1e-6; %m, Length of each finger (note:  2 fingers per
section)
Zref = 50;
S21 = zeros(size(f));

ncell = round(L/g.finger.modperiod);

for ii = 1:numel(f)

    abcd_tot = [1 0; 0 1];
    abcd = [1 0; 0 1];

    n_unit_cell = g.finger.modperiod / g.finger.p ;

    %calculate abdc matrix for a unit cell
    for jj = 1:n_unit_cell

        beta = 2*pi*f(ii) / vph(ii);
        betaf = 2*pi*f(ii) / vph(ii);

        Lfinger = (g.finger.w/2 + g.finger.l + g.finger.modamp *
cos( 2*pi*(jj-0.5)*g.finger.p / g.finger.modperiod))* 1e-6;

        Zin_finger = -1i * Z0(ii) * cot( betaf * Lfinger);

        %abcd matrix for shunt admittance - factor of 2 accounts for 2
fingers
        %per section
        abcd_finger = [1 0; 2/Zin_finger 1];

        %half length section -
        %modify the length by a factor to try to represent the widening of
        %the line as it connects to the finger
        xfactor = (g.finger.p - g.finger.w)/g.finger.p;
```

```
        abcd_trl = [cos( beta * Dfinger * xfactor / 2)          1i * Z0(ii) 
* sin( beta * Dfinger * xfactor / 2); ...
            1i * sin( beta * Dfinger * xfactor / 2) / Z0(ii)    cos( beta * 
Dfinger * xfactor / 2)];

        %cascade the finger and TRL section S-matrices
        abcd = abcd * abcd_trl;
        abcd = abcd * abcd_finger;
        abcd = abcd * abcd_trl;
    end

    %Combine into total abcd
    for n = 1:ncell
        abcd_tot = abcd_tot * abcd;
    end

    %Store new Sparamps for the position
    S = abcd2s(abcd_tot, Zref);
    S21(ii) = S(2, 1);
end

%figure
%plot(f, angle(S21))

f_GHz = f / 1e9;

uth = -1*unwrap(angle(S21(:)));
for fj = 2:length(f)
    while uth(fj)<uth(fj-1)
        uth(fj:end) = uth(fj:end) + 2*pi;
    end
end

slope = (uth(2)-uth(1))./(f(2)-f(1));
intercept = uth(1) - slope*f(1);
uth = uth - intercept;

len_meters = L./1e6;
kperm = uth/len_meters;

p = polyfit(f(f<0.1e9), kperm(f<0.1e9), 1);
vph = 2*pi/p(1);

figure(1)
hold on
plot(f./1e9,abs(S21))
grid on
grid minor
xlabel('Frequency (GHz)')
ylabel('S_{21}')
set(gca,'FontSize',16)
set(gca,'FontWeight','bold')
set(gcf,'Position',[1000 100 1500 1000])
```

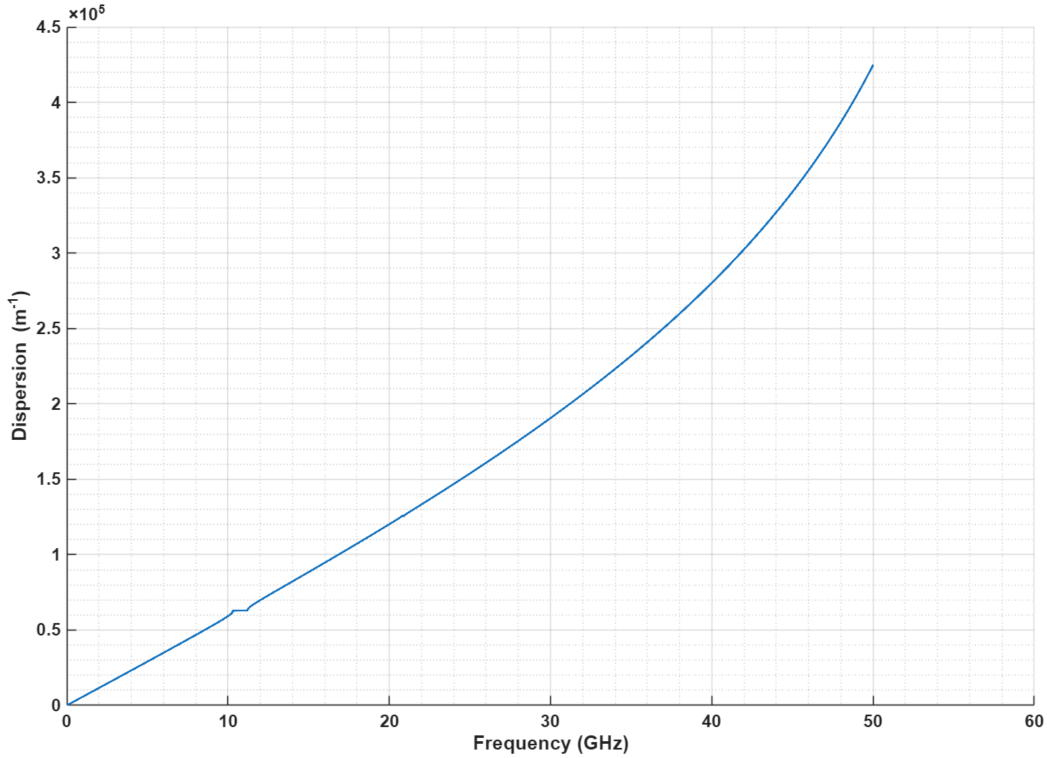
```
figure(2)
hold all
plot(f./1e9,kperm,'Linewidth',2)
% plot(f./1e9,-unwrap(angle(S21z(end,:))))
% plot(f,z2dB((S21z(end,:))),'Linewidth',2)
% xlim([0 30])
grid on
grid minor
xlabel('Frequency (GHz)')
ylabel('Dispersion (m^{-1})')
set(gca,'FontSize',16)
set(gca,'FontWeight','bold')
set(gcf,'Position',[1000 100 1500 1000])

Ctot = Cperm.*(2*g.finger.l + g.finger.p)./g.finger.p;

Zfin = real(1./(vph*Ctot));

disp(['Z = ',num2str(Zfin(1)),' Ohms'])
disp(['vph = ',num2str(vph./3e8),' c'])
```

Z = 49.7853 Ohms
vph = 0.0036148 c






```
save(filename, 'kperm', 'f', 'S21', 'len_meters', 'vph','Zfin')
```



Published with MATLAB® R2025a