IMPEDANCE and NETWORKS

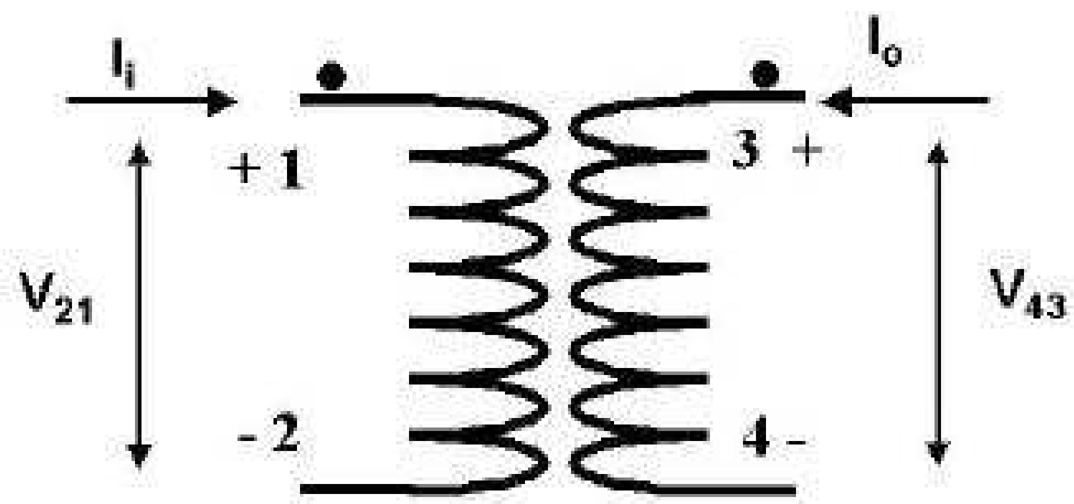
- Transformers
- Networks
- A method of analysing complex networks
- Y-parameters and S-parameters

Transformers

- Combining the effects of self and mutual inductance, the equations of a transformer are given by,

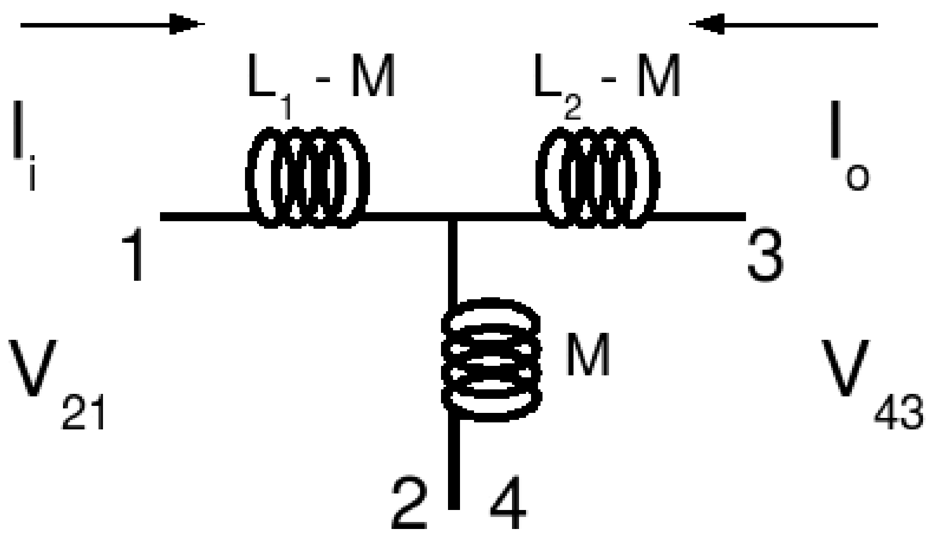
$$V_{21} = j\omega L_1 I_i + j\omega M I_o$$

$$V_{43} = j\omega M I_i + j\omega L_2 I_o$$



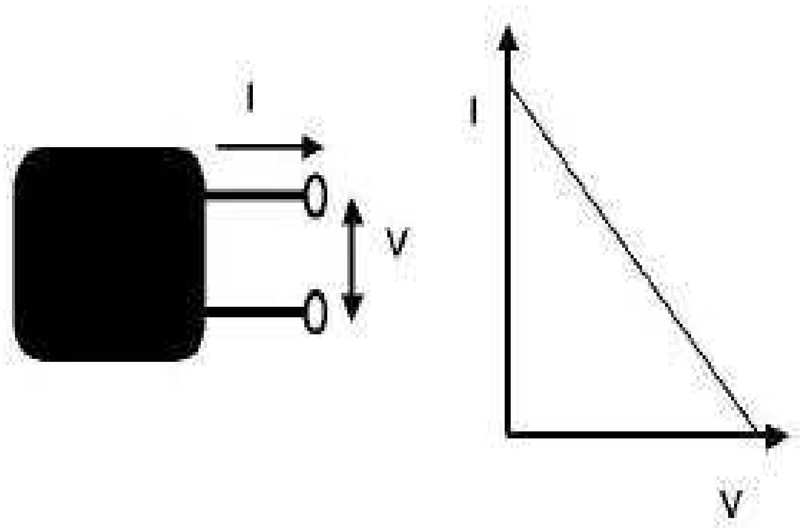
Simplified Transformer Circuit

- The following circuit is equivalent to the above.
- Note the definition of voltage sense.



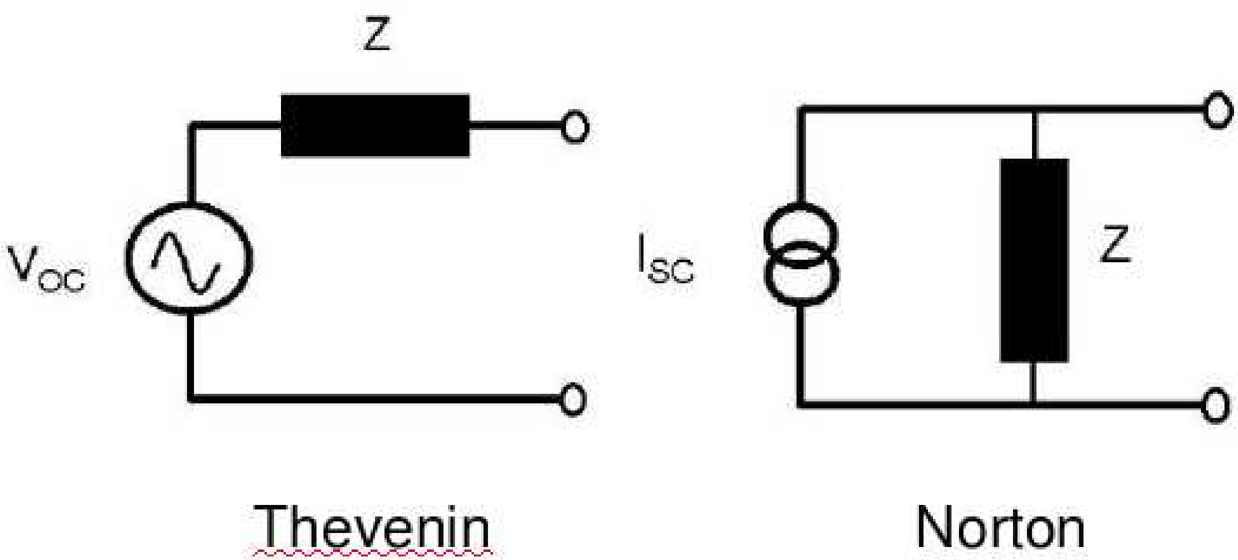
Networks: Linearity

- Maxwell's equations are linear
- Thus if we have an *arbitrary* circuit, then the **E.M.F** seen at any two terminals must be a **linear** function of the current drawn.



Networks: Thevenin and Norton Equivalents

- Every linear circuit can be simplified to the following trivial forms.



Networks: Impedance and Admittance

- Impedance: $\mathbf{V} = \mathbf{Z}\mathbf{I}$
- Impedance: $\mathbf{Z} = \mathbf{R} + \mathbf{jX}$
- R = Resistance and X = Reactance
- Admittance: $\mathbf{I} = \mathbf{Y}\mathbf{V}$
- Admittance: $\mathbf{Y} = \mathbf{G} + \mathbf{jB}$
- G = Conductance and B = Susceptance

Networks: Analysis of Linear Circuits

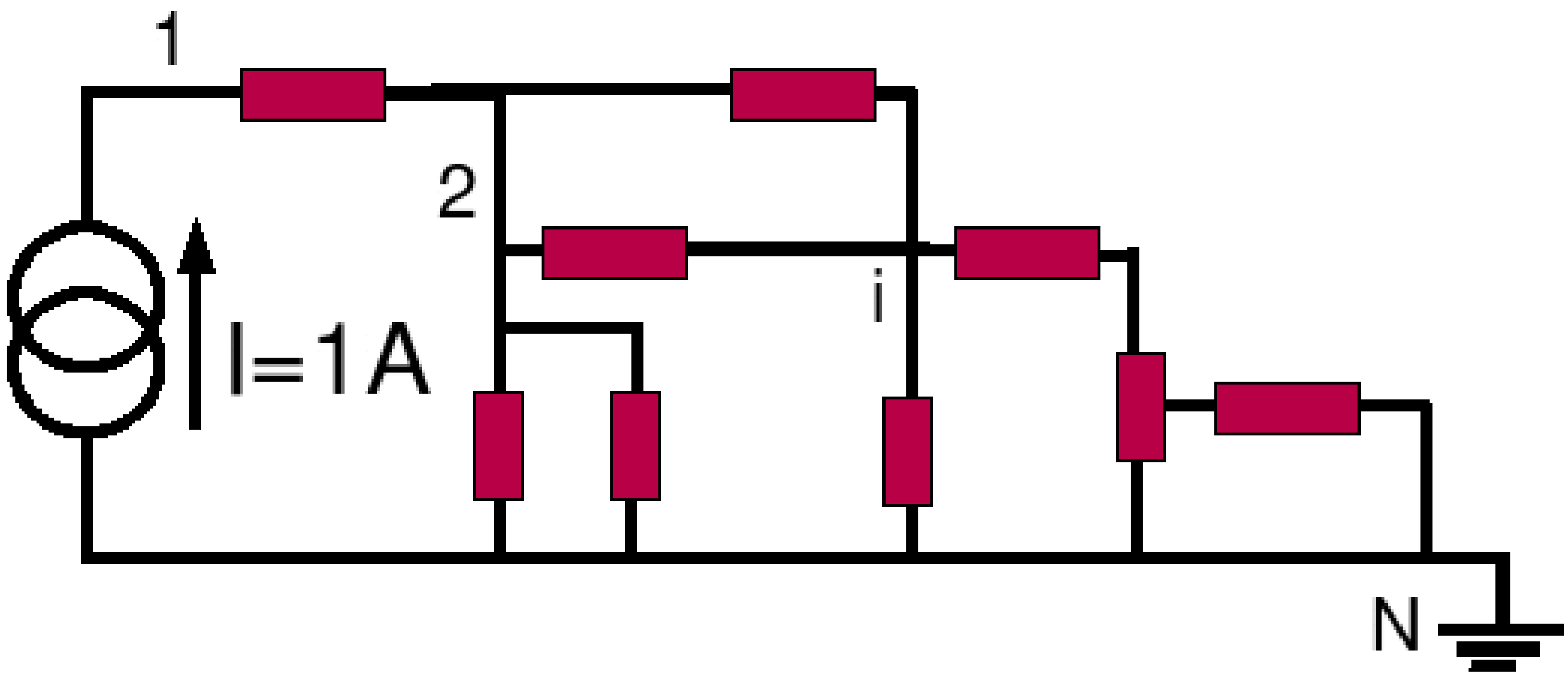
- One of the main differences between radiofrequency compared to low frequency circuits is the effect of parasitics.
- Even the simplest transistor or oscillator models give rise to relatively complex circuits.
- We need a circuit solver.
- I now introduce a simple MATLAB program to solve all linear circuits.

Networks: Analysis of Linear Circuits: SOLVE: The Circuit Solver

- SOLVE is a simple and short MATLAB solver for complex circuits
- Main advantage of SOLVE is that it is so simple that you can easily trace problems.
- You can of course use PSPICE or any other program of your choice.
- I recommend that you get to know PSPICE (or some of the more complex commercial products, ANSOFT, ADA, CADENCE) due to the built in support for RF components and the detailed analysis of non-linear behaviour that these packages can provide.

Networks: Analysis of Linear Circuits

- We consider the following circuit driven by a 1 Ampere current source:

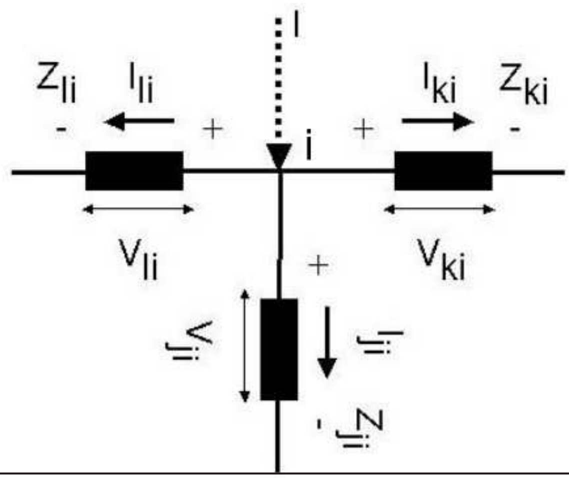


Networks: Analysis of Linear Circuits

- Consider a typical node in the circuit.
- The current I is injected into the node.
- At the first node (on the current source)...

$$1 = \sum_{j=1}^{N} y_{j1}\left(V_1 - V_j\right) = \left(\sum_{j=1}^{N} y_{j1}\right) V_1 - \sum_{j=1}^{N-1} y_{j1} V_j$$

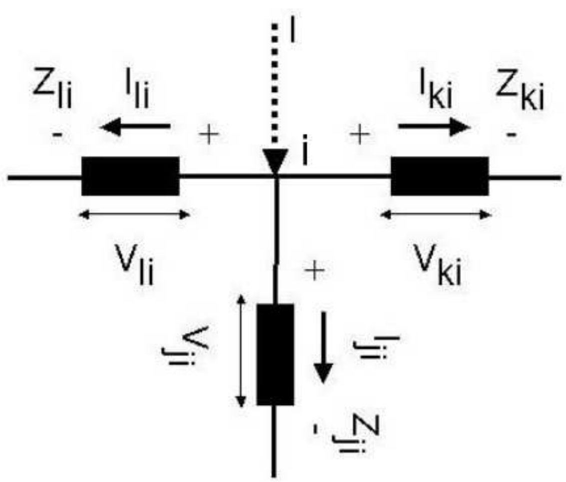
- where each voltage is referenced to ground. $V_N = 0$ and $y_{ii} = 0$.



Networks: Analysis of Linear Circuits

➤ At the i^{th} node where $i \neq 1$, there is no current injected,

$$0 = \sum_{j=1}^{N} y_{ji} \left(V_i - V_j \right) = \left(\sum_{j=1}^{N} y_{ji} \right) V_i - \sum_{j=1}^{N-1} y_{ji} V_j$$



Networks: Analysis of Linear Circuits

- We therefore have an $(N-1) \times (N-1)$ matrix equation.

$$I = \mathbf{M}V$$

where

$$M_{ii} = \sum_{k=1}^{N} y_{ki}$$

where the sum is such that $k \neq i$ and,

$$M_{ij} = -y_{ji}$$

for $i \neq j$ and $i = 1, ..., (N-1)$.

- Notice that only the diagonal terms M_{ii} require knowledge of admittance to ground, y_{kN}.
- Solve for V.

Networks: SOLVE. Matlab Code

```
%%load the relevant circuit parameters
ADMITTANCE_PROGRAM

m = zeros(N-1,N-1,Nvals);
Current_Source = [1,zeros(1,N-2)]'
%'
mm = sum(y,1);

for i = 1:N-1

      m(i,i,:) = mm(1,i,:);

      for j = 1:N-1
            if  i ~= j
                 m(i,j,:) = -y(j,i,:);
            end
      end
end

Volts = zeros(N-1,Nvals);
for i = 1:Nvals
       Volts(:,i) = inv(m(:,:,i))*Current_Source;
end
```

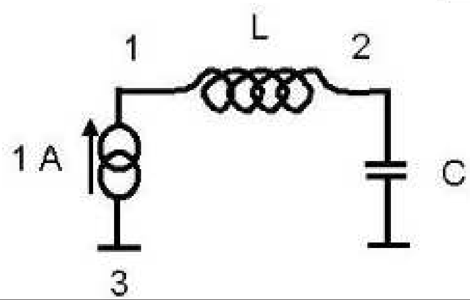
Networks: Analysis of Linear Circuits: Some Simple Rules to Remember

- Node 1 is at the current source. The N^{th} node is ground.
- $V_N = 0$ and is **not** used.
- A "node" might actually be a junction of many joined wires.. so be careful not to leave wires out accidently.
- V_i represents the voltage from ground to the i^{th} node.
- There is always only one admittance between each node. This means that we have to parallel up parallel admittances.
- By the way.. Parallel admittances add. Why?
- The current source injects 1 Ampere. Thus the input impedance is numerically equal to the voltage on node 1.

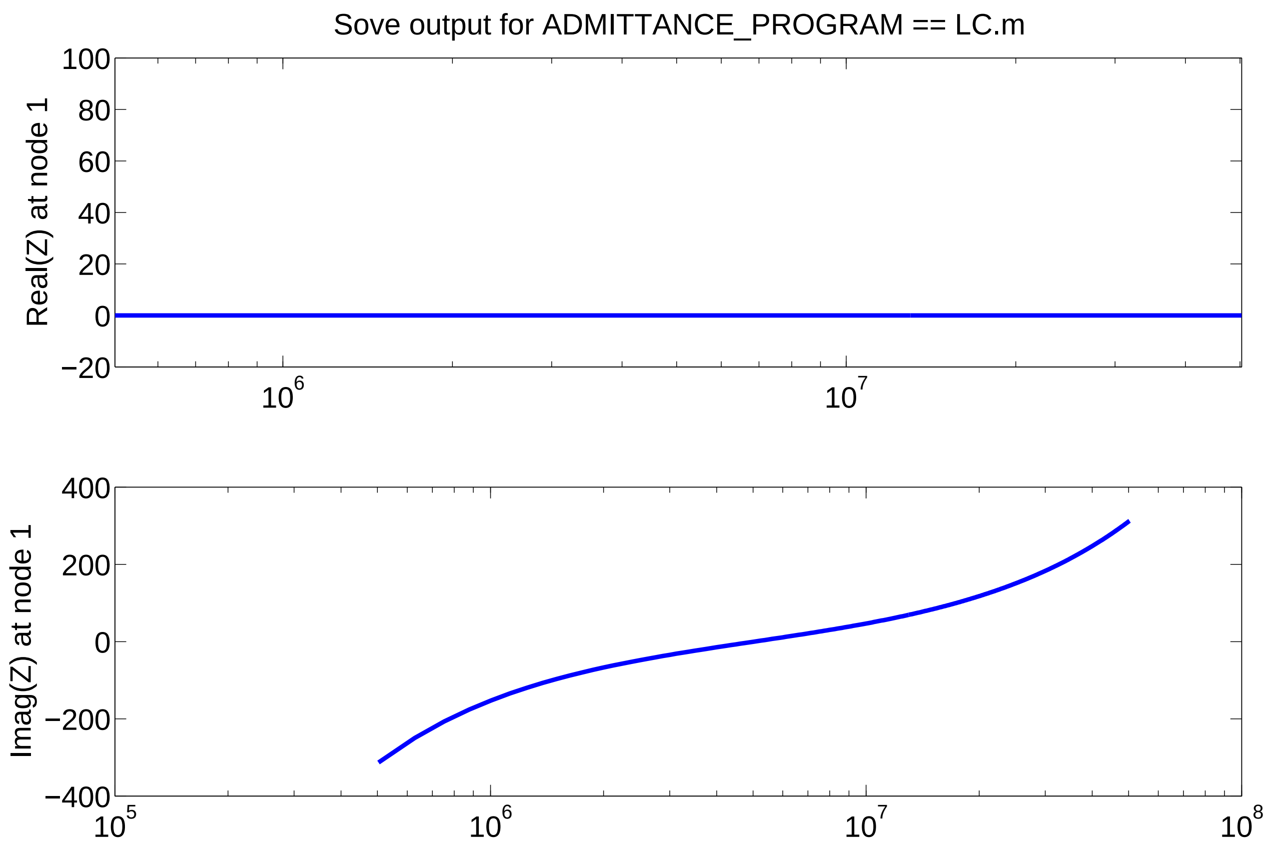
Networks: Analysis of Linear Circuits: Simple Example

- A series LC circuit...

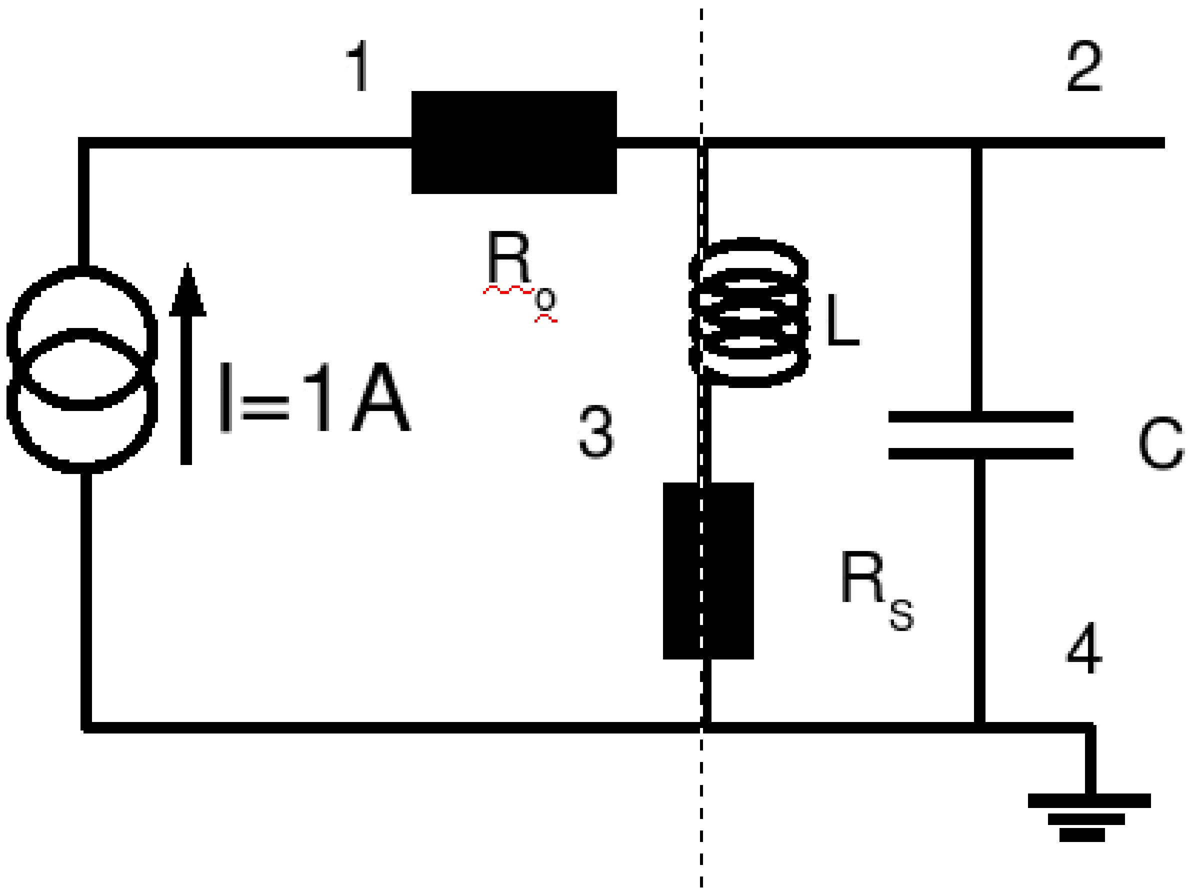
```
clear all
close all
% series LC fed by a current source

%Need to provide Nvals... the number of frequency points
Nvals = 40;

%Specify the number of nodes
N = 3;
y = zeros(N-1,N,Nvals);

%Specify an L and C in SI units
L = 1.e-6; %Henry
C = 1.e-9; %Farads

%Define the frequncies around the resonant frequency
fres = 1/2/pi/sqrt(L*C);

frequency = linspace(.1*fres,10*fres,Nvals);
omega = 2*pi*frequency;

%The admittances between all nodes
y(1,2,:) = 1./(j*omega*L);
y(2,1,:) = 1./(j*omega*L);
y(2,3,:) = j*omega*C;
y(3,2,:) = j*omega*C;
```



Networks: Analysis of Linear Circuits: Simple Example



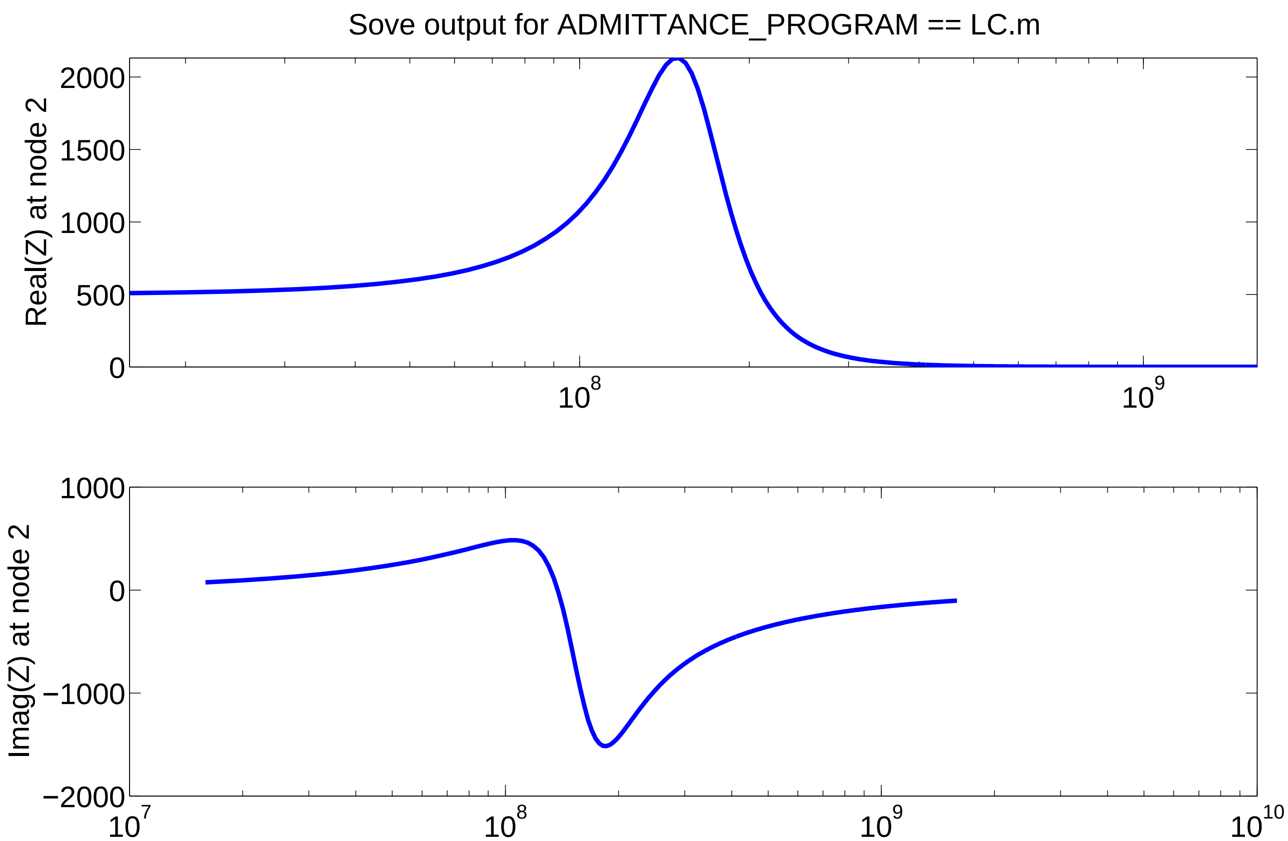
Networks: Analysis of Linear Circuits: More complex Example



Networks: Analysis of Linear Circuits: More complex Example

```
%Specify the number of nodes
N = 4;
y = zeros(N-1,N,Nvals);

%Specify an L and C in SI units
L = 1.e-6; %Henry
C = 1.e-12; %Farads
Ro = 10000;
Rs = 500;
%Define the frequncies around the resonant frequency
fres = 1/2/pi/sqrt(L*C);

frequency = linspace(.1*fres,10*fres,Nvals);
omega = 2*pi*frequency;

%The admittances between all nodes
y(1,2,:) = 1./Ro;
y(2,1,:) = 1./Ro;

y(2,3,:) = 1./(j*omega*L);
y(3,2,:) = 1./(j*omega*L);

y(4,2,:) = j*omega*C;
y(2,4,:) = j*omega*C;

y(4,3,:) = 1./Rs;
y(3,4,:) = 1./Rs;
```

Networks: Analysis of Linear Circuits: More complex Example



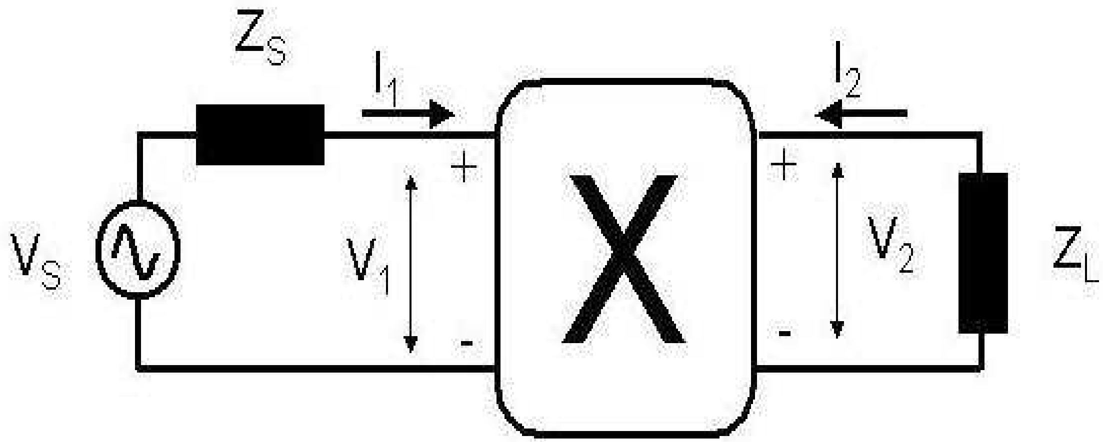
Networks: Y-parameters

- Four port parameters describing a linear passive network

$$I_1 = y_i V_1 + y_r V_2$$

$$I_2 = y_f V_1 + y_o V_2$$

- where y_i is the **input admittance**, y_r the **reverse-transfer admittance**, y_f the **forward-transfer admittance** and y_o the **output admittance**.



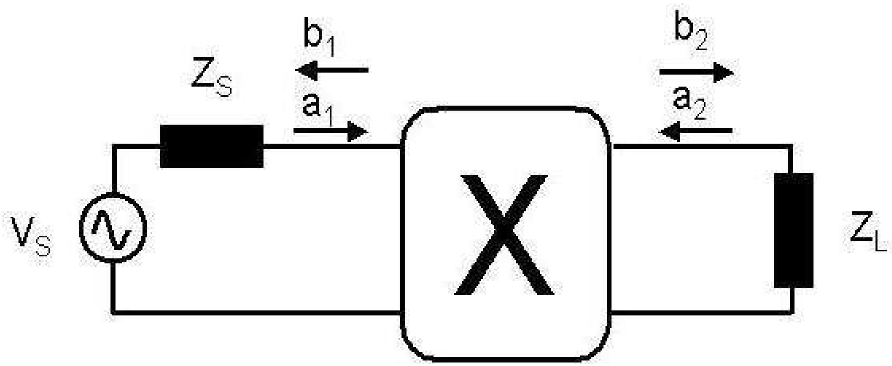
Networks: S-parameters

- Four port parameters describing a linear passive network

$$b_1 = S_{11}a_1 + S_{12}a_2$$

$$b_2 = S_{21}a_1 + S_{22}a_2$$

- where S_{11} is **the input reflection coefficient** or **return loss**, S_{12} is the **reverse transmission coefficient**, S_{21} the **forward transmission coefficient**, and S_{22} is **the output reflection coefficient**.



Networks: S-parameters

- a_1 and b_2 are forward waves and a_2 and b_1 are backward waves. They may be voltages or currents.
- S_{21} is also referred to as **transfer function**, **insertion loss** or **gain**.
- $S_{11} = (b_1/a_1)|a_2 = 0$. To measure S_{11} let $Z_L = Z_0$ (**the characteristic impedance**) and measure the input reflection coefficient.
- $S_{21} = (b_2/a_1)|a_2 = 0$. To measure S_{21} let $Z_L = Z_0$ and measure the ratio of the transmitted to the incoming signal.
- $S_{22} = (b_2/a_2)|a_1 = 0$. To measure S_{22} swap the source and Z_L and let $Z_L = Z_0$

Networks: Y-parameters vs S-parameters

- Y-parameters and S-parameters are equivalent.
- Y-parameters are best for calculations.
- S-parameters are best for measurements. It is easy to measure power in a terminated network.
- S-parameters are usually the parameters given in component datasheets.

Networks: Y-parameters vs S-parameters

- Y-parameters and S-parameters are related:

$$y_i = \frac{(1+S_{22})(1-S_{11})+S_{12}S_{21}}{\Delta Z_0}$$

$$y_r = \frac{-2S_{12}}{\Delta Z0}$$

$$y_f = \frac{-2S_{21}}{\Delta Z0}$$

$$y_o = \frac{(1+S_{11})(1-S_{22})+S_{12}S_{21}}{\Delta Z_0}$$

where $\Delta = (1+S_{11})(1+S_{22}) - S_{21}S_{12}$.