

Scilab Textbook Companion for
Microwave Engineering
by D. M. Pozar¹

Created by
Amardeep Kumar
B.TECH (pursuing)
Electronics Engineering
M.N.N.I.T, Allahabad
College Teacher
Rajeev Gupta
Cross-Checked by
Sonanya Tatikola, IIT Bombay

July 31, 2019

¹Funded by a grant from the National Mission on Education through ICT, <http://spoken-tutorial.org/NMEICT-Intro>. This Textbook Companion and Scilab codes written in it can be downloaded from the "Textbook Companion Project" section at the website <http://scilab.in>

Book Description

Title: Microwave Engineering

Author: D. M. Pozar

Publisher: Addison - Wesley Longman, Incorporated

Edition: 1

Year: 1993

ISBN: 0-201-50418-9

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes	4
2 ELECTROMAGNETIC THEORY	6
3 TRANSMISSION LINE THEORY	13
4 TRANSMISSION LINE AND WAVEGUIDES	27
5 MICROWAVE NETWORK ANALYSIS	33
6 IMPEDENCE MATCHING AND TUNNING	41
7 MICROWAVE RESONATORS	45
8 POWER DIVIDERS DIRECTIONAL COUPLERS AND HYBRIDS	51
9 MICROWAVE FILTERS	58
10 THEORY AND DESIGN OF FERRIMAGNETIC COM- PONENTS	65
11 ACTIVE MICROWAVE CIRCUITS	69

List of Scilab Codes

Exa 2.1	program to calculate wavelength phase velocity and	6
Exa 2.2	program to find out skin depth	7
Exa 2.3	program to find the resulting fields	7
Exa 2.4	program to show decomposition in to RHCP and LHCP	8
Exa 2.5	program to compute the poynting vector	9
Exa 2.6	program to compute propagation constant and other	10
Exa 2.7	program to plot the reflection coefficients	10
Exa 3.1	program to determine transmission line parameters	13
Exa 3.2	program to find out load impedance	14
Exa 3.3	program to find out return loss in dB and others	16
Exa 3.4	program to find input impedance and SWR of line	20
Exa 3.5	program to find out load admittance and other	20
Exa 3.6	program to find out characteristic impedance	21
Exa 3.7	program to determine unknown load impedance	22
Exa 3.8	program to calculate attenuation constant	23
Exa 3.9	program to find the attenuation constant	25
Exa 3.10	program to calculate attenuaton	26
Exa 4.1	program to find the cut off frequency	27
Exa 4.2	program to find the cut off frequency	28
Exa 4.3	program to find out the highest usable frequency	29

Exa 4.4	program to calculate and plot propagation constant	29
Exa 4.5	program to find width of a copper strip line	30
Exa 4.7	program to calculate width and length . . .	31
Exa 4.9	program to calculate the group velocity . . .	32
Exa 5.1	program to find equivalent voltages and current	33
Exa 5.2	program to compute reflection coefficient . .	33
Exa 5.3	program to find z parameter of two port network	34
Exa 5.4	program to find the s parameter of 3 dB attenuator	34
Exa 5.5	program to determine reciprocity and losslessness	35
Exa 5.6	program to find ABCD parameter of two port network	36
Exa 5.7	program to find admittance matrix for bridge T	36
Exa 5.8	program to compute power gains	37
Exa 5.9	program to derive the expression for τ_{in} .	38
Exa 5.10	program to find out expression for τ_{in} . .	38
Exa 5.11	determine amplitude of forward and backward wave	39
Exa 5.12	find excitation coefficient of forward wave TE ₁₀	39
Exa 6.1	program to design an L section matching network	41
Exa 6.5	design quarter wave matching transformer .	42
Exa 6.6	program to evaluate the worst case percent error	42
Exa 6.7	design three section binomial transformer . .	43
Exa 6.8	design three section chebysev transformer . .	43
Exa 6.9	design triangular taper and a klopfenstein taper	44
Exa 7.1	program to compare the Q factor	45
Exa 7.2	program to compute length and Q of the resonator	46
Exa 7.3	program to find required length and other .	46
Exa 7.4	program to find dimension and Q	47

Exa 7.5	program to find resonant frequency and Q .	48
Exa 7.6	program to find the mode number and Q . .	48
Exa 7.7	program to find value of the coupling capacitor	49
Exa 7.8	derive expression for change in resonant frequency	49
Exa 7.9	derive expression for change in resonant frequency	50
Exa 8.1	program to compute the reflection coefficients	51
Exa 8.2	design equi split wilkinson power divider . .	52
Exa 8.3	design bethe hole coupler for x band waveguide	52
Exa 8.4	program to design a four hole chebysev coupler	53
Exa 8.5	design 50 ohm branchline quadrature hybrid junc	54
Exa 8.6	determine even and odd mode characteristic impeden	54
Exa 8.7	design a 20 db single section coupled line coupler	55
Exa 8.8	design a three section 20 db coupler	55
Exa 8.9	design a 3 dB 50 ohm langer coupler	56
Exa 8.10	design 180 deg ring hybrid for 50 ohm system imped	56
Exa 8.11	calculate even and odd mode characteristic impeden	57
Exa 9.1	program to compute the propagation constant	58
Exa 9.2	program to design a low pass composite filter	60
Exa 9.3	program to find out number of filter elements	60
Exa 9.4	program to design a maximum flat low pass filter	61
Exa 9.5	program to design a band pass filter	61
Exa 9.6	design a low pass filter using micrstrip lines	62
Exa 9.7	design a stepped impedance low pass filter .	62
Exa 9.8	design a coupled line band pass filter	63
Exa 9.9	design a bandpass filter	63
Exa 9.10	design a bandpass filter using capacitive coupled	64
Exa 10.1	calculate and plot phase and attenuation constants	65

Exa 10.2	program to design an e plane resonance isolator	66
Exa 10.3	program to design a resonance isolator	67
Exa 10.5	design a two slab remanent phase shifter	67
Exa 11.1	determine equivalent noise temperature of amplife	69
Exa 11.2	find the dynamic range of the amplifier	69
Exa 11.3	program to calculate the noise figure	70
Exa 11.4	calculate the impedence of the diode	70
Exa 11.5	determine the stability of the transistor	71
Exa 11.6	design an amplifier for maximum gain	72
Exa 11.7	design an amplifier to have a gain of 11 dB	73
Exa 11.8	calculate maximum error in Gt and design amplifier	73
Exa 11.9	design a load matching network	74
Exa 11.10	program to design a transistor oscillator	75
Exa 11.11	obtain the greatest ratio of off to on attenuation	76
Exa 12.1	compute directivity radiation intensity and others	77
Exa 12.2	program to find the reactive power in dbm	78
Exa 12.3	calculate the input and output SNR	78
Exa 12.4	program to find the maximum range of radar	79
Exa 12.5	program to find the J by S ratio	79
Exa 12.6	calculate power density of 20 m from antena	80

List of Figures

2.1	program to plot the reflection coefficients	12
3.1	program to find out load impedance	16
3.2	program to find out return loss in dB and others	19
3.3	program to find out characteristic impedance	22
3.4	program to determine unknown load impedance	24
4.1	program to calculate and plot propagation constant	30
9.1	program to compute the propagation constant	59
10.1	calculate and plot phase and attenuation constants	66

Chapter 2

ELECTROMAGNETIC THEORY

Scilab code Exa 2.1 program to calculate wavelength phase velocity and

```
1 // example 2.1,page no.-24.
2 // program to calculate wavelength,phase velocity
  and wave impedance.
3 f=3*10^9;
4 mur=3;
5 muo=4*%pi*10^-7;
6 eipsilao=8.854*10^-12;
7 eipsilar=7;
8 mue=muo*mur;
9 eipsila=eipsilao*eipsilar;
10 Vp=sqrt(1/(mue*eipsila));
11 lamda=Vp/f;
12 eta=sqrt(mue/eipsila);
13 //Result
14 disp(Vp,'phase velocity in meter per second=')
    // phase velocity.
15 disp(lamda,'wavelength in meter=') // wavelength.
16 disp(eta,'wave impedance in ohm=') // wave
    impedance.
```

Scilab code Exa 2.2 program to find out skin depth

```
1 // example: -2.2.page no.-26.
2 // program to find out skin depth of aluminium,
   copper,gold and silver at frequency 10GHZ.
3 f=10*10^9;
4 muo=4*%pi*10^-7; // permeability in free space.
5 omega=2*%pi*f;
6 sigma_aluminium=3.816*10^7;
7 sigma_copper=5.813*10^7;
8 sigma_gold=4.098*10^7;
9 sigma_silver=6.173*10^7;
10 delta1=sqrt(2/(omega*muo*sigma_aluminium));
11 delta2=sqrt(2/(omega*muo*sigma_copper));
12 delta3=sqrt(2/(omega*muo*sigma_gold));
13 delta4=sqrt(2/(omega*muo*sigma_silver));
14 //result
15 disp(delta1,'skin depth of aluminium in meter=') //
   skin depth of aluminium.
16 disp(delta2,'skin depth of copper in meter=') //
   skin depth of copper.
17 disp(delta3,'skin depth of gold in meter=') //skin
   depth of gold.
18 disp(delta4,'skin depth of silver in meter=') //
   skin depth of silver.
```

Scilab code Exa 2.3 program to find the resulting fields

```
1 // example: -2.3,page no.-31.
2 //program to find the resulting fields by assumibg
   plane waves on either side of the current sheet
   and enforcing the boundary conditions.
```

```

3 syms E x E1 E2 H1 H2 z Jo A B c N n d ko y;
4 sym('n*(E2-E1)=0'); //boundary condition to be
    satisfied at z=0
5 sym('z*(E2-E1)=0'); // " " "
    "
6 sym('n*(H2-H1)=Jo'); // " " "
    "
7 sym('z*(H2-H1)=Jo'); // " " "
    "
8 E1=A*N*exp(%i*ko*z)*x; // x component of electric
    field (region z<0).
9 H1=A*N*exp(%i*ko*z)*(-y); // -y component of
    magnetic field (region z<0).
10 E2=B*N*exp(-%i*ko*z)*x; // x component of electric
    field (region z>0).
11 H2=B*N*exp(-%i*ko*z)*y; // y component of electric
    field (region z>0).
12 disp(E1,'for z<0, E1=')
13 disp(H1,'for z<0, H1=')
14 disp(E2,'for z>0, E2=')
15 disp(H2,'for z>0, H2=')
16 //from boundary conditions imposed.we get:-
17 c=[-1 -1;1 -1];
18 d=[A;B];
19 c*d==[Jo;0];
20 d=inv(c)*[Jo;0];
21 //result
22 // A=-Jo/2; B=-Jo/2.
23 disp(d)

```

Scilab code Exa 2.4 program to show decomposition in to RHCP and LHCP

```

1 // example:-2.4,page no.-34.
2 // program to show that a circularly polarized plane
    wave can be decomposed in to RHCP and LHCP.

```

```

3 A=sym('A');
4 B=sym('B');
5 Eo=sym('Eo');
6 x=sym('x');
7 y=sym('y');
8 Ko=sym('Ko');
9 z=sym('z');
10 E=Eo*(x+2*y)*exp(-%i*Ko*z); // given
11 // can be written as:=>E=A*(x-y)*exp(-%i*Ko*z)+B*(x+
    y)*exp(-%i*Ko*z), so
12 p=[1 1;-%i/2 %i/2];
13 q=[A;B];
14 r=[1;1];
15 p*q==Eo*r;
16 q=inv(p)*Eo*r;
17 //result
18 disp('value of A and B will be=')
19 disp(q)
20 disp(q(1,1)*(x-y)*exp(-%i*Ko*z)+q(2,1)*(x+y)*exp(-%i
    *Ko*z), 'E=')
21 //conclusion:-any linearly polarized wave can be
    decomposed in to two circularly polarized waves.

```

Scilab code Exa 2.5 program to compute the poynting vector

```

1 // example:-2.5,page no.-36.
2 // program to compute the poynting vector for the
    plane wave field.
3 syms E Eo H k s n N x r;
4 E=Eo*exp(-%i*k*r); // electric field.
5 H=(E/N)*n; //N is intrinsic impedance,n is unit
    vector.
6 H1=conj(H) // conjugate of magnetic field.
7 s=E*H1;
8 //result

```

```
9 disp(s, 'poynting vector is (meter square)=')
10 disp('which shows that power density is flowing in
    the direction of propagation.')
```

Scilab code Exa 2.6 program to compute propagation constant and other

```
1 // example: -2.6, page no. -46.
2 // program to compute propagation constant, impedance,
    skin depth, reflection and transmission
    coefficient.
3 f=1*10^9;
4 omega=2*%pi*f;
5 sigma=5.813*10^7; // for copper.
6 mue=4*%pi*10^-7; // permeability in free space.
7 delta=sqrt(2/(mue*sigma*omega)); // skin depth.
8 gama=((1+%i)/delta); //propagation constant.
9 eta=gama/sigma; // impedance
10 etao=377; //intrinsic impedance in free space.
11 tao=((eta-etao)/(eta+etao)); // reflection
    coefficient.
12 t=(2*eta)/(eta+etao); //transmission coefficient.
13 // result
14 disp(delta, 'skin depth in meter=')
15 disp(gama, 'propagation constant=')
16 disp(eta, 'intrinsic impedance in ohm=')
17 disp(tao, 'reflection coefficient=')
18 disp(t, 'transmission coefficient=')
```

Scilab code Exa 2.7 program to plot the reflection coefficients

```
1 // example: -2.7. page no. -50.
2 // program to plot the reflection coefficients for
    parallel and perpendicular polarized plane waves
```

```

    incident from free space on to a dielectric
    region with Er=2.55, versus incidence angle.
3 Er=2.55; // relative permittivity of dielectric
    medium.
4 N1=377; // intrinsic impedance
5 N2=N1/sqrt(Er); // intrinsic impedance of
    dielectric medium.
6 xb=asin(sqrt(1/(1+1/2.55))); // brewster angle
    valid only in case of parallel polarization.
7 xt=acos(sqrt(1-(1/Er)^2*sin(xb))); // angle of
    transmission.
8 xi=[0:0.01:%pi/2]; // incidence angle.
9 // for parallel polarization
10 N2=N2*cos(xt);
11 N1=N1*cos(xi);
12 Tpar=(N2-N1)/(N2+N1);
13 w=abs(Tpar);
14 // result
15 subplot(1,2,1)
16 xtitle("parallel polarization", "xi(incidence angle)"
    ,"Tpar(reflection coefficient)")
17 plot2d(xi,w,style=3,rect=[0,0,%pi/2,1])
18 // for perpendicular polarization. //NOTE:- in
    case of this polarization there is no brewster
    angle.
19 xt=acos(sqrt(1-(1/Er)^2*sin(xi)));
20 n1=377.*cos(xt);
21 n2=(377/sqrt(Er)).*cos(xi);
22 Tper=(n2-n1)/(n1+n2);
23 z=abs(Tper);
24 //result
25 subplot(1,2,2)
26 xtitle("perpendicular polarization", "xi(incidence
    angle)", "Tper(reflection coefficient)")
27 plot2d(xi,z,style=2,rect=[0,0,%pi/2,1])

```

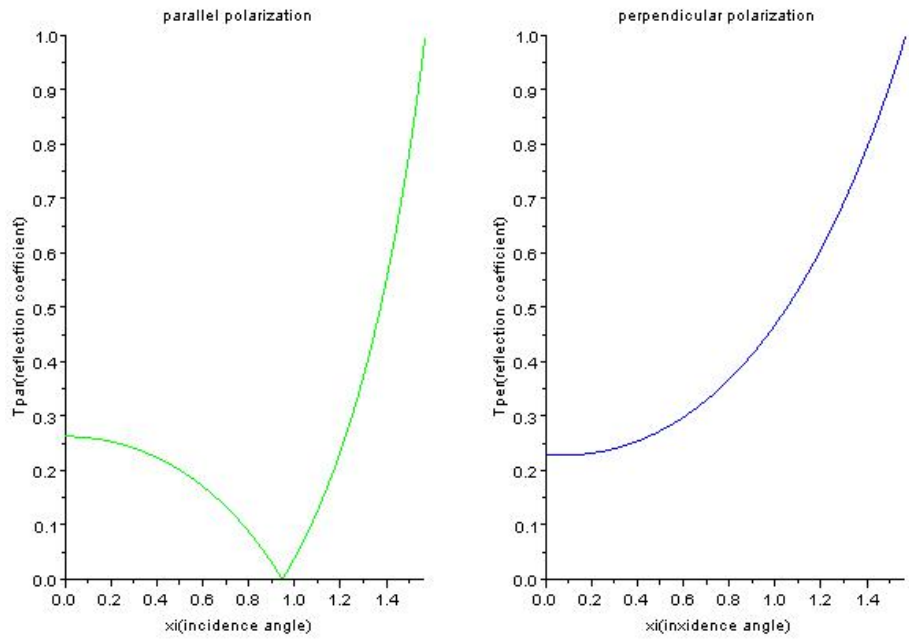


Figure 2.1: program to plot the reflection coefficients

Chapter 3

TRANSMISSION LINE THEORY

Scilab code Exa 3.1 program to determine transmission line parameters

```
1 //example: -3.1, page no. -72.
2 // program to determine transmission line parameters
3
4 syms E H Vo P a b Io mue y z Q pi L eipsila G C R Rs
5 w;
6 E=(Vo/(P*log(b/a)))*exp(-%i*y*z); // in radial
7 direction.
8 H=(Io/(2*pi*P))*exp(-%i*y*z); // in phi direction.
9 H=H*conj(H)*P;
10 E=E*conj(E)*P;
11 L=(mue/((Io)^2))*integ(integ(H,P),Q); // surface
12 integral in culindrical coordinate system
13 L=limit(L,P,b)-limit(L,P,a); // limits when
14 integrated w.r.t rho.
15 L=limit(L,Q,2*pi)-limit(L,Q,0); // limits when
16 integrated w.r.t phi.
17 C=(eipsila/(Vo^2))*integ(integ(E,P),Q); // surface
18 integral in culindrical coordinate system
19 C=limit(C,P,b)-limit(C,P,a); // limits when
```

```

    integrated w.r.t rho.
13 C=limit(C,Q,2*pi)-limit(C,Q,0); // limits when
    integrated w.r.t phi.
14 R=(Rs/(Io^2))*integ(H,Q);
15 R=limit(R,P,b)+limit(R,P,a);
16 R=limit(R,Q,2*pi)-limit(R,Q,0); // limits when
    integrated w.r.t phi.
17 G=((w*eipsila)/(Vo^2))*integ(integ(E,P),Q); //
    surface integral in culindrical coordinate system
18 G=limit(G,P,b)-limit(G,P,a); // limits when
    integrated w.r.t rho.
19 G=limit(G,Q,2*pi)-limit(G,Q,0); // limits when
    integrated w.r.t phi.
20 // result
21 disp(L, 'self-inductance in H/m =')
22 disp(C, 'capacitance in F/m =')
23 disp(R, 'resistance in Ohm/m =')
24 disp(G, 'shunt conductance in S/m =')

```

check Appendix [AP 2](#) for dependency:

smith_chart_tao.sci

Scilab code Exa 3.2 program to find out load impedance

```

1 // example: -3.2, page no. -87.
2 // program to find out load impedance.
3 clc
4 clear
5 // function for smith chart for finding load
    impedance when reflection coefficient is given.
6 function []=smith_chart(tao)
7 theta=0:0.1:2*pi;
8 for r=0:0.1:10
9 x=(1/(1+r))*cos(theta)+(r/(1+r));
10 y=(1/(1+r))*sin(theta);

```

```

11 plot2d(x,y,style=2,rect=[-2,-2,2,2])
12 end
13 for X=-2:0.1:2
14     if X==0
15         X=0.01;
16     end
17 x=1+(1/X)*cos(theta);
18 y=(1/X)*sin(theta)+(1/X);
19 plot2d(x,y,style=3,rect=[-2,-2,2,2])
20 xgrid(2)
21 xtitle("smith chart","Tao_Real","Tao_Imaginary")
22 end
23 x=abs(tao)*cos(theta);
24 y=abs(tao)*sin(theta);
25 plot2d(x,y,style=5,rect=[-2,-2,2,2])
26 theta=-%pi/2:0.1:%pi/2;
27 x=abs(tao)*cos(theta);
28 [r angle]=polar(tao);
29 tao=[r angle]
30 y=x*tan(tao(1,2));
31 plot2d(x,y,style=5,rect=[-2,-2,2,2])
32 endfunction
33 Zo=100; // characteristic impedance.
34 tao=0.560+0.215*i; // reflection coefficient.
35 z=(1+tao)/(1-tao); // normalized impedance(
    normalized w.r.t Zo)
36 Zl=z*Zo;
37 // result
38 disp(Zl,'load impedance = ')
39 // by smith chart.
40 smith_chart(tao)
41 // when analyse with the help of smith chart.see the
    angle from x=0 axis i.e Tao_real axis.if it is
    above this axis take angle anticlockwise and if
    it is below this axis.take angle clockwise from
    Tao_real axis below.

```

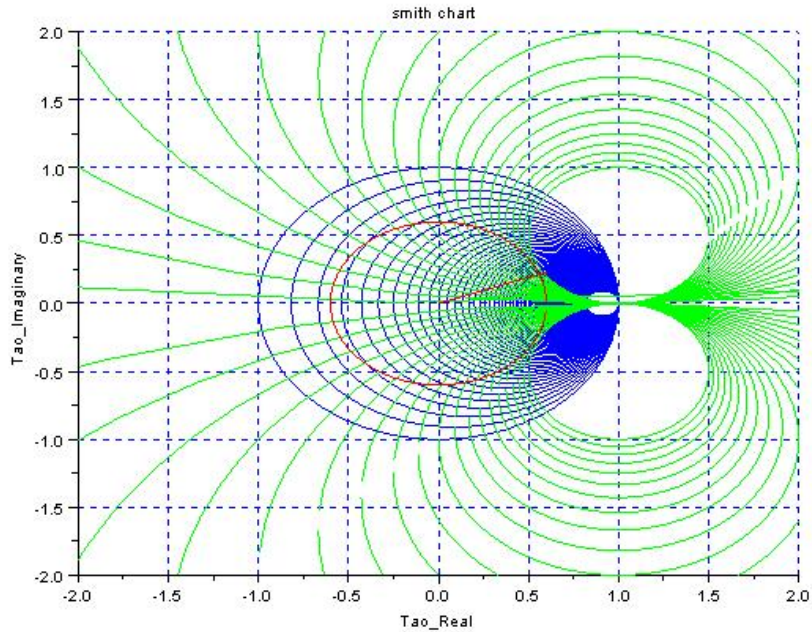


Figure 3.1: program to find out load impedance

check Appendix [AP 4](#) for dependency:

`reflection_coefficient.sci`

check Appendix [AP 2](#) for dependency:

`smith_chart_tao.sci`

check Appendix [AP 5](#) for dependency:

`swr.sci`

Scilab code Exa 3.3 program to find out return loss in dB and others

```

1 // example:-3.3,page no.-87.
2 // program to find out return loss in dB,SWR and
  reflection coefficient.
3 // example:-3.2,page no.-87.
4 // program to find out load impedance.
5 clc
6 clear
7 // function for smith chart for finding load
  impedance when reflection coefficient is given.
8 function []=smith_chart(tao)
9 theta=0:0.1:2*%pi;
10 for r=0:0.1:10
11 x=(1/(1+r))*cos(theta)+(r/(1+r));
12 y=(1/(1+r))*sin(theta);
13 plot2d(x,y,style=2,rect=[-2,-2,2,2])
14 end
15 for X=-2:0.1:2
16     if X==0
17         X=0.01;
18     end
19 x=1+(1/X)*cos(theta);
20 y=(1/X)*sin(theta)+(1/X);
21 plot2d(x,y,style=3,rect=[-2,-2,2,2])
22 xgrid(2)
23 xtitle("smith chart","Tao_Real","Tao_Imaginary")
24 end
25 x=abs(tao)*cos(theta);
26 y=abs(tao)*sin(theta);
27 plot2d(x,y,style=5,rect=[-2,-2,2,2])
28 theta=-%pi/2:0.1:%pi/2;
29 x=abs(tao)*cos(theta);
30 [r angle]=polar(tao);
31 tao=[r angle]
32 y=x*tan(tao(1,2));
33 plot2d(x,y,style=5,rect=[-2,-2,2,2])
34 endfunction
35 Zo=100; // characteristic impedance.
36 tao=0.560+0.215*%i; // reflection coefficient.

```

```

37 z=(1+tao)/(1-tao); // normalized impedance(
    normalized w.r.t Zo)
38 Zl=z*Zo;
39 // result
40 disp(Zl,'load impedance = ')
41 // by smith chart.
42 smith_chart(tao)
43 // when analyse with the help of smith chart.see the
    angle from x=0 axis i.e Tao_real axis.if it is
    above this axis take angle anticlockwise and if
    it is below this axis.take angle clockwise from
    Tao_real axis below.
44
45 Zl=80-40*i; // load impedance.
46 Zo=50; // characteristic impedance.
47 z=Zl/Zo; // normalized impedance.
48 tao=reflection_coefficient(Zl,Zo);
49 SWR=VSWR(abs(tao));
50 Rl=-20*log10(abs(tao));
51 disp(abs(tao),'reflection coefficient = ')
52 disp(SWR,'standing wave ratio = ')
53 disp(Rl,'return loss in dB = ')
54 smith_chart(tao)
55 // when analyse with the help of smith chart.see the
    angle from x=0 axis i.e Tao_real axis.if it is
    above this axis take angle anticlockwise and if
    it is below this axis.take angle clockwise from
    Tao_real axis below.

```

check Appendix [AP 3](#) for dependency:

input_impedence.sci

check Appendix [AP 4](#) for dependency:

reflection_coefficient.sci

check Appendix [AP 5](#) for dependency:

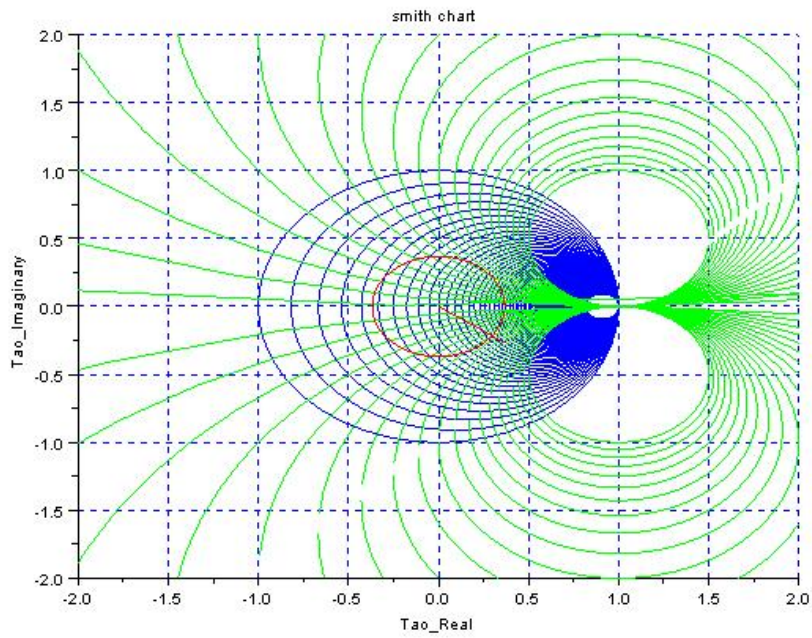


Figure 3.2: program to find out return loss in dB and others

swr.sci

Scilab code Exa 3.4 program to find input impedance and SWR of line

```
1 // example no.-3.4, page no.-88.
2 // program to find input impedance and SWR of line.
3 clc
4 clear
5 exec("DEPENDENCIES/reflection_coefficient.sci");
6 exec("DEPENDENCIES/swr.sci");
7 exec("DEPENDENCIES/input_impedence.sci");
8 Zo=75; Zl=37.5+75*i; l=0.02; eipsilar=2.56; f=3*10^9; c
   =3*10^8;
9 b=(2*pi*f*sqrt(eipsilar))/c; // beta
10 tao=reflection_coefficient(Zl,Zo);
11 Zin=input_impedence(tao,b,l,Zo);
12 // result
13 disp(Zin,'input impedance = ')
14 tao=abs(tao);
15 s=VSWR(tao);
16 // result
17 disp(s,'SWR of the line = ')
```

Scilab code Exa 3.5 program to find out load admittance and other

```
1 // example:-3.5, page no.-91.
2 // program to find out load admittance and input
   admittance of the line
3 syms lamda;
4 Zl=100+50*i;
5 Zo=50;
6 le=0.15; //electrical length(l/lamda).
7 b=(2*pi);
```

```

8 tao=reflection_coefficient(Z1,Zo);
9 Zin=input_impedence(tao,b,le,Zo);
10 Yin=1/Zin;
11 Yl=1/Zl;
12 // result
13 disp(Yin,'input admittance = ')
14 disp(Yl,'load admittance = ')

```

Scilab code Exa 3.6 program to find out characteristic impedance

```

1 //example:-3.6,page no.-93.
2 // program to find out characteristic impedance and
  plot the magnitude of reflection coefficient
  versus normalized frequency.
3 Zl=100;// load impedance
4 Zi=50;//impedence of line which is to be matched
5 //as it is a quarter wave transformer so,Zi=(Zo)^2/
  zl;
6 Zo=sqrt(Zi*Zl);
7 disp(Zo,'characteristic impedance of tne matching
  section=')
8 syms f fo x;
9 x=f/fo;
10 x=0:0.001:4;
11 y=(%pi/2)*(x);
12 Zin=Zo*(((Zl*cos(y))+(Zo*i*sin(y)))/((Zo*cos(y))+(
  Zl*i.*sin(y))))
13 tao=((Zin-Zo)/(Zin+Zo));
14 tao=abs(real(tao)+imag(tao));
15 plot2d(x,tao,style=6,rect=[0,0,4,1])
16 xtitle("reflection coefficient versus normalized
  frequency for quarter wave transformer","f/fo","
  tao(reflection coefficient)")

```

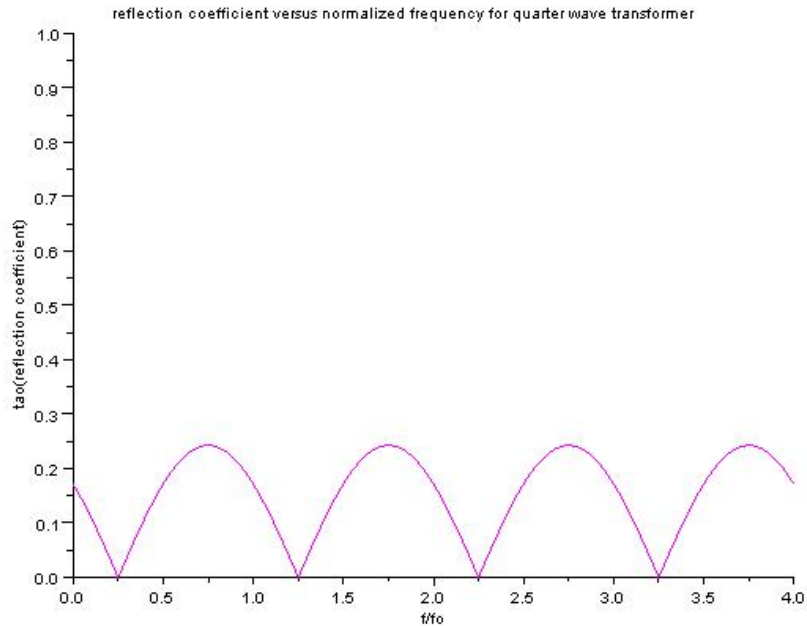


Figure 3.3: program to find out characteristic impedance

check Appendix [AP 2](#) for dependency:

smith_chart_tao.sci

Scilab code Exa 3.7 program to determine unknown load impedance

```

1 // example: -3.7, page no. -101.
2 // NOTE: -this example is a method for calculating
  unknown load impedance of slotted line section.
  all data are given and preassumed.
3 // program to determine unknown load impedance.

```

```

4  clc
5  clear
6  exec("DEPENDENCIES/smith_chart_tao.sci")
7  Zl=0;Zo=50; // for short circuiting the load.
8  SWR=%inf;
9  // short circuit is removed and replace with unknown
   load.
10 SWR=1.5;lamda=0.04;
11 lmin=4.2-2.72;
12 tao=(1.5-1)/(1.5+1);
13 theta=(%pi+((4*%pi)/4)*1.48);
14 tao=abs(tao)*exp(%i*theta);
15 Zl=50*((1+tao)/(1-tao));
16 // result
17 disp(Zl,'load impedance = ')
18 smith_chart(tao)
19 // when analyse with the help of smith chart.see the
   angle from x=0 axis i.e Tao_real axis.if it is
   above this axis take angle anticlockwise and if
   it is below this axis.take angle clockwise from
   Tao_real axis below.

```

Scilab code Exa 3.8 program to calculate attenuation constant

```

1 // example:-3.8,page no.-108.
2 // program to calculate attenuation constant.
3 syms alpha R Rs L G C eta a b w pi eipsila eipsilac
   mue eta;
4 //from example 3.1:- alpha=(R*(sqrt(C/L)+G*sqrt(L/C))
   ;
5 eta=sqrt(mue/eipsila);
6 L=(mue/(2*pi))*(log(b/a));
7 C=(2*pi*eipsila)/log(b/a);

```

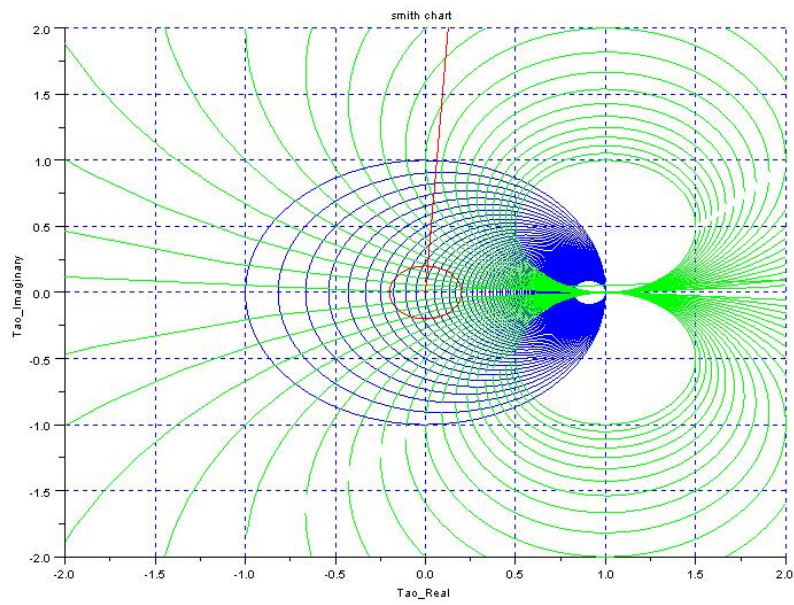


Figure 3.4: program to determine unknown load impedance

```

8 R=(Rs/(2*pi))*((1/a)+(1/b));
9 G=(2*pi*w*eipsilac)/log(b/a);
10 alpha=(1/2)*(R*sqrt(C/L)+G*sqrt(L/C));
11 disp(alpha,'attenuation constant = ')

```

Scilab code Exa 3.9 program to find the attenuation constant

```

1 // example:-3.9,page no.-111.
2 // program to find ht eattenuation constant of
  coaxial line.
3 syms E H Vo Zo P a b B z pi Po Q Rs Plc alpha Pld
  Plc w eipsila;
4 //Zo=(eta/(2*pi))*log(b/a);
5 E=(Vo/(P*(log(b)-log(a))))*exp(-%i*B*z); //B=beta.
6 H=(Vo/(2*pi*P*Zo))*exp(-%i*B*z);
7 H=conj(H)*P; // for defining E cross H*.
8 Po=(1/2)*integ(integ((E*H),P),Q);
9 Po=limit(Po,P,b)-limit(Po,P,a);
10 Po=limit(Po,Q,2*pi)-limit(Po,Q,0);
11 disp(Po,'power flowing on the lossless line = ')
12 H=(H*conj(H))/P; // for defining |H|^2;
13 Plc=(Rs/2)*integ(integ(H,z),Q);
14 Plc=limit(Plc,P,b)+limit(Plc,P,a);
15 Plc=limit(Plc,z,1)-limit(Plc,z,0);
16 Plc=limit(Plc,Q,2*pi)-limit(Plc,Q,0);
17 disp(Plc,'conductor loss = ')
18 E=E*conj(E)*P;
19 Pld=((w*eipsila)/2)*integ(integ(integ(E,P),Q),z);
20 Pld=limit(Pld,P,b)-limit(Pld,P,a);
21 Pld=limit(Pld,z,1)-limit(Pld,z,0);
22 Pld=limit(Pld,Q,2*pi)-limit(Pld,Q,0);
23 disp(Pld,'dielectric loss = ')
24 alpha=(Pld+Plc)/(2*Po); // attenuation constant.
25 disp(alpha,'attenuation constant = ')

```

Scilab code Exa 3.10 program to calculate attenuaton

```
1 // example: -3.10, page no. -114.
2 // program to calculate attenuaton due to conductor
  loss of a coaxial line using incremental
  inductance rule.
3 syms Zo eta pi a b Rs l alpha alpha_c alpha_dash
  delta alpha_c_dash sigma w mue;
4 sd=sqrt(2/(w*mue*sigma))
5 Zo=(eta*log(b/a))/(2*pi);
6 alpha_c=(Rs/(4*Zo*pi^2))*(diff(log(b/a),b)-diff(log(
  b/a),a));
7 disp(alpha_c,'attenuation due to conductor loss = ')
8 alpha_c_dash=alpha_c*(1+((2/pi)*atan((1.4*delta)/sd)
  ));
9 disp(alpha_c_dash,'attenuation corrected for surface
  roughness = ')
```

Chapter 4

TRANSMISSION LINE AND WAVEGUIDES

Scilab code Exa 4.1 program to find the cut off frequency

```
1 // example: -4.1, page no. -148.
2 // program to find the cut off frequency fo the
  first four propagating modes.
3 a=0.02286; b=0.01016; f=10*10^9; k=209.44; sigma
  =5.8*10^7; mue=4*pi*10^-7;
4 c=3*10^8;
5 m=0; n=1;
6 fc=(c/(pi*2))*sqrt(((pi*m)/a)^2+((pi*n)/b)^2);
7 fc=fc/(10^9);
8 disp(fc, 'cut-off frequency for TE01 mode in GHZ=')
9 m=1; n=0;
10 fc=(c/(pi*2))*sqrt(((pi*m)/a)^2+((pi*n)/b)^2);
11 fc=fc/(10^9);
12 disp(fc, 'cut-off frequency for TE10 mode in GHZ=')
13 m=2; n=0;
14 fc=(c/(pi*2))*sqrt(((pi*m)/a)^2+((pi*n)/b)^2);
15 fc=fc/(10^9);
16 disp(fc, 'cut-off frequency for TE20 mode in GHZ=')
17 m=1; n=1;
```



```

18 fc=(c/(%pi*2))*sqrt(((%pi*m)/a)^2+((%pi*n)/b)^2);
19 fc=fc/(10^9);
20 disp(fc,'cut-off frequency for TE11 mode in GHZ=')
21 B=sqrt(k^2-(%pi/a)^2) // for TE10 mode
22 Rs=sqrt(((2*%pi*f)*mue)/(2*sigma)); // surface
    resistance.
23 disp(Rs,'surface resistance in ohm=')
24 ac=(Rs/(a^3*b*B*k*377))*((2*b*%pi^2)+(a^3*k^2)) //
    attenuation constant.
25 ac=-20*(-ac)*log10(%e);
26 disp(ac,'attenuation constant in dB/m=')

```

Scilab code Exa 4.2 program to find the cut off frequency

```

1 //example:-4.2,page no.-160.
2 //program to find the cut off frequency of two
    propagating modes of a circular waveguide.
3 a=0.005;eipsilar=2.25;f=13*10^9;c=3*10^8;d=0.001;
    sigma=6.17*01^7;muo=4*%pi*10^-7;
4 m=1;n=1;
5 p11=1.841;p01=2.405;
6 fc=(p11*c)/(2*%pi*a*sqrt(eipsilar));
7 kc=p11/a;
8 fc=fc/(10^9);
9 disp(fc,'cut-off frequency for TE11 mode in GHZ')
10 m=0;n=1;
11 fc=(p01*c)/(2*%pi*a*sqrt(eipsilar));
12 fc=fc/(10^9);
13 disp(fc,'cut-off frequency for TE01 mode in GHZ')
14 // so,TE01 can't be propagating mode.only TE11 will
    be.
15 k=(2*%pi*f*sqrt(eipsilar))/c;
16 disp(k,'k in m-1=')
17 B=sqrt(k^2-kc^2);
18 disp(B,'propagation constant of TE11 mode')

```

```

19 ac=(k^2*d)/(2*B);
20 Rs=sqrt((2*pi*f*muo)/(2*sigma)); // surface
    resistance.
21 acm=(Rs/(a*k*377*B))*(kc^2+((k^2)/(p11^2-1)));
22 a=ac+acm;
23 a=-20*(-0.547*0.5)*log10(%e);
24 disp(a,'total attenuation factor in dB=')

```

Scilab code Exa 4.3 program to find out the highest usable frequency

```

1 //example:-4.3,page no.-167.
2 //program to find out the highest usable frequency.
3 a=0.000889;b=0.0029464;eipsilar=2.2;c=3*10^8;
4 // here (b/a)=3.3,so for this kc*a=0.47
5 kc=0.47/a;
6 fc=(c*kc)/(2*pi*sqrt(eipsilar))
7 fc=fc/(10^9);
8 fmax=0.95*fc;
9 disp(fmax,'maximum usable frequency in GHZ=')

```

Scilab code Exa 4.4 program to calculate and plot propagation constant

```

1 //example:-4.4,page no.-175.
2 // program to calculate and plot the propagation
    constant of first three propagating surface wave
    mode.
3 eipsilar=2.55;c=3*10^8; // x=d/lamdao;
4 x=0.001:0.01:1.2;
5 for n=0:1:4
6 y=sqrt(eipsilar-((n^2)./(4.*(x^2)*(eipsilar-1))));
    // y=beta/lamdao;
7 plot2d(x,y,style=2,rect=[0,0,1.2,1.6])
8 end

```

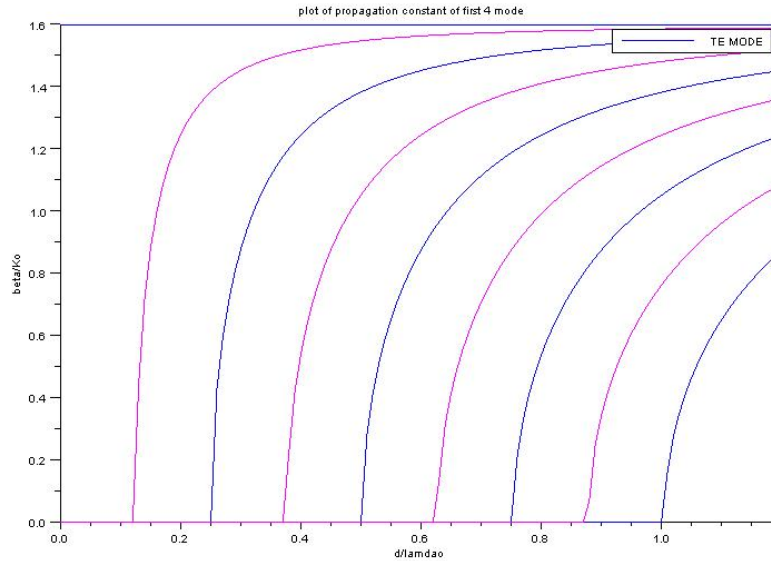


Figure 4.1: program to calculate and plot propagation constant

```

9 x=0.001:0.01:1.2;
10 for n=1:1:4
11     y=sqrt(eipsilar-((((2.*n)-1)^2)./(16.*(x^2)*(
        eipsilar-1))))
12     plot2d(x,y,style=6,rect=[0,0,1.2,1.6])
13 end
14 xtitle("plot of propagation constant of first 4 mode
        ", "d/lamdao", "beta/Ko");
15 legend("TE MODE")

```

Scilab code Exa 4.5 program to find width of a copper strip line

```
1 // example:-4.5,page no.-180.
```

```

2 // program to find width of a copper strip line
  conductor.
3 eipsilar=2.20;Zo=50;b=0.0032;d=0.001,f=10^10;t
  =0.00001;
4 c=3*10^8;Rs=0.026;A=4.74;
5 x=(30*%pi)/(sqrt(eipsilar)*Zo);
6 x=x-0.441;
7 w=b*x;
8 if ((sqrt(eipsilar)*Zo)<120)
9     disp("width of copper strip line conductor is
  0.00266m")
10 end
11 K=(2*%pi*f*sqrt(eipsilar))/c;
12 ad=(K*d)/2;
13 ac=(2.7*(10^-3)*Rs*eipsilar*Zo*A)/(30*%pi*(b-t));
14 a=ac+ad;
15 a=20*a*log10(%e);
16 lamda=c/(sqrt(eipsilar)*f);
17 alambda=lambda*a;
18 disp(K,'wave number=')
19 disp(ad,'dielectric aattenuation=')
20 disp(ac,'conductor attenuation=')
21 disp(a,'total attenuation constant=')
22 disp(alambda,'attenuation in dB/lamda=')

```

Scilab code Exa 4.7 program to calculate width and length

```

1 //example:-4.7,page no.-187.
2 //program to calculate the width and length of
  microstrip line.
3 eipsilae=1.87;//effective dielectric constant.
4 Zo=50;q=%pi/2;c=3*10^8;
5 f=2.5*10^9;
6 ko=(2*%pi*f)/c;
7 d=0.00127;

```

```

8 eipsilar=2.20;
9 // for w/d>2;
10 B=7.985;
11 w=3.081*d*100;
12 disp(w, 'width in centi meter=')
13 l=(q*100)/(sqrt(eipsilae)*ko);
14 disp(l, 'length of microstrip in centi meter=')

```

Scilab code Exa 4.9 program to calculate the group velocity

```

1 //example:-4.9,page no.-197.
2 //program to calculate the group velocity.
3 syms w c v;
4 B=sym('B');
5 ko=sym('ko');
6 kc=sym('kc');
7 ko=w/c;
8 B=sqrt(ko^2-kc^2);
9 v=diff(B,w);
10 vg=v^(-1);
11 vg=(c*B)/ko;
12 vp=w/B;
13 disp(vg, 'group velocity=')
14 disp(vp, 'phase velocity=')
15 disp('conclusion:-since B<ko,we have that vg<c<vp,
      which indicates that the phase velocity of a
      waveguide mode may be greater than the speed of
      light.but the group velocity will be lesser than
      the speed of light.')
```

Chapter 5

MICROWAVE NETWORK ANALYSIS

Scilab code Exa 5.1 program to find equivalent voltages and current

```
1 // example: -5.1, page no. -209.
2 // program to find the equivalent voltages and
  current.
3 syms a b A Zte V I C1 C2 P;
4 P=(a*b*A^2)/(4*Zte);
5 c=(1/2)*V*I;
6 d=(1/2)*(A^2)*C1*C2;
7 C1=sqrt((a*b)/2); // on comparison.
8 C2=sqrt((a*b)/2)*Zte; // on comparison.
9 c=[C1 C2];
10 disp(c)
11 disp("which completes the transmission line
  equivalence for the TE10 mode.")
```

Scilab code Exa 5.2 program to compute reflection coefficient

```

1 //example: -5.2, page no. -212.
2 //program to compute reflection coefficient.
3 a=0.03485; b=0.01580; eipsilao=8.854*10^-12; muo=4*%pi
   *10^-7;
4 f=4.5*10^9;
5 w=2*%pi*f; // angular frequency.
6 // for z<0 region air filled.
7 eipsilar=2.56; //for z>0 region.
8 ko=w*sqrt(muo*eipsilao);
9 k=ko*sqrt(eipsilar);
10 Ba=sqrt(ko^2-(%pi/a)^2); // propagation constant in
   air region z<0.
11 Bd=sqrt(k^2-(%pi/a)^2); // propagation constant in
   dielectric region z>0.
12 Zoa=(ko*377)/Ba;
13 Zod=(ko*377)/Bd;
14 tao=(Zod-Zoa)/(Zod+Zoa);
15 disp(tao, 'reflection coefficient')

```

Scilab code Exa 5.3 program to find z parameter of two port network

```

1 //example: -5.3, page no. -220.
2 // program to find the z parameter of the two port
   network.
3 syms Z11 Z12 Z22 Z21 V1 I1 V2 I2 Za Zb Zc;
4 Z11=Za+Zc; // for I2=0.
5 Z12=(Zc/(Zb+Zc))*(Zb+Zc); //for I1=0.
6 Z21=(Zc/(Za+Zc))*(Za+Zc); // for I2=0.
7 Z22=Zb+Zc; //for I1=0.
8 Z=[Z11 Z12; Z21 Z22]; // z-parameter matrix.
9 disp(Z, 'Z-parameter of two port network = ')

```

Scilab code Exa 5.4 program to find the s parameter of 3 dB attenuator

```

1 // example: -5.4, page no. -221.
2 // program to find the s-parameter of 3-dB
  attenuator circuit.
3 Za=8.56; Zb=8.56, Zc=141.8; Zo=50;
4 S11=((((Zo+Zb)*Zc)/(Zo+Zb+Zc))+Za)-Zo)/((((Zo+Zb)*
  Zc)/(Zo+Zb+Zc))+Za)+Zo); // reflection
  coefficient seen at port 1.
5 S22=((((Zo+Za)*Zc)/(Zo+Za+Zc))+Zb)-Zo)/((((Zo+Za)*
  Zc)/(Zo+Za+Zc))+Zb)+Zo); // reflection
  coefficient seen at port 2.
6 S12=(((1/((((Zo+Za)*Zc)/(Zo+Za+Zc))+Zb))*(((Zo+Za)*
  Zc)/(Zo+Za+Zc)))*(Zo/(Zo+Za))); // transmission
  coefficient from port 2 to 1.
7 S21=(((1/((((Zo+Zb)*Zc)/(Zo+Zb+Zc))+Za))*(((Zo+Zb)*
  Zc)/(Zo+Zb+Zc)))*(Zo/(Zo+Zb))); // transmission
  coefficient from port 1 to 2.
8 S=[S11 S12; S21 S22]; // s-parameter matrix.
9 disp(S, 'S-parameter of 3-dB attenuator circuit is ='
  )

```

Scilab code Exa 5.5 program to determine reciprocity and losslessness

```

1 //example: -5.5, page no. -226.
2 //program to determine the reciprocity and lossless
  of two port network and find return loss.
3 syms S R1 tao;
4 S=[0.1 0.8*i; 0.8*i 0.2]; // s-parameter matrix.
5 if (S(1,2)==S(2,1))
6   disp("the network is reciprocal.")
7 else
8   disp("the network is not reciprocal.")
9 end
10 if (S(1,1)^2+S(1,2)^2==1)
11   disp("the network is lossless.")
12 else

```



```

13   disp("the network is lossy.")
14   end
15   tao=S(1,1)-(S(1,2)*S(2,1))/(1+S(2,2)); //input
      reflection coefficient.
16   Rl=-20*log10(abs(tao)); // return loss in dB.
17   //result
18   disp(Rl,'return loss at port 1 in dB=')

```

Scilab code Exa 5.6 program to find ABCD parameter of two port network

```

1 // example:-5.6,page no.-232.
2 //program to find the ABCD parameter of a two-port
      network.
3 syms A B C D V1 V2 I1 I2 Z;
4 //A=V1/V2; // for i2=0;
5 A=1;
6 B=V1/(V1/Z);
7 C=0;
8 D=I1/I1;
9 ABCD=[A B;C D];
10 // result
11 disp(ABCD,'abcd parameter')

```

Scilab code Exa 5.7 program to find admittance matrix for bridge T

```

1 // example:-5.7,page no.-238.
2 // program to find the admittance matrix for bridge-
      T network.
3 syms Za Z1 Z2 Z3 Y Ya Yb D;
4 Za=[Z1+Z2 Z2;Z2 Z1+Z2];
5 Yb=[1/Z3 -1/Z3;-1/Z3 1/Z3];
6 Y1=1/Z1;Y2=1/Z2;
7 Ya=1/Za;

```

```

8 Y=Ya+Yb;
9 D=((Z2+Z1)^2-Z2^2);
10 // result
11 disp(Y,'admittance matrix for bridge-T network=')

```

Scilab code Exa 5.8 program to compute power gains

```

1 // example: -5.8, page no. -243.
2 // program to compute power gains.
3 f=10^10; Zs=20; Zl=30; Zo=50;
4 S=[-0.39+%i*0.225 0.009848+%i*-0.001736; 2.02+0.356*
    %i -0.3464-%i*0.2];
5 taos=(Zs-Zo)/(Zs+Zo);
6 taol=(Zl-Zo)/(Zl+Zo);
7 taoin=S(1,1)+((S(1,2)*S(2,1)*taol)/(1-S(2,2)*taol));
8 taoout=S(2,2)+((S(1,2)*S(2,1)*taos)/(1-S(1,1)*taos))
    ;
9 Ga=(abs(S(2,1)^2)*(1-abs(taos)^2))/((abs(1-S(1,1)*
    taos)^2)*(1-abs(taoout)^2));
10 Gt=(abs(S(2,1)^2)*(1-abs(taos)^2)*(1-abs(taol)^2))
    /((abs(1-S(2,2)*taol)^2)*abs(1-taos*taoin)^2);
11 G=(abs(S(2,1)^2)*(1-abs(taol)^2))/((abs(1-S(2,2)*
    taol)^2)*(1-abs(taoin)^2));
12 disp(G,'actual power gain=')
13 disp(Ga,'the available power gain=')
14 disp(Gt,'the transducer power gain=')
15 disp(taoin,'reflection coefficient looking at port
    1=')
16 disp(taoout,'reflection coefficient looking at port
    2=')
17 disp(taos,'reflection coefficient at the source=')
18 disp(taol,'reflection coefficient at the load=')

```

Scilab code Exa 5.9 program to derive the expression for tao_{in}

```
1 // example: -5.9, page no. -248.
2 // program to derive the expression for taoin.
3 syms S S11 S22 S12 S21 taol taoin a1 a2 b1 b2 a b;
4 S=[S11 S12;S21 S22];
5 b=[b1;b2];
6 a=[a1;a2];
7 b=S*a;
8 disp(b)
9 //so, S11 will be the reflection coefficient i.e
   taoin.
10 taoin=S11+((S21*S12*taol)/(1-S22*taol));
11 // result
12 disp(taoin, 'the expression for taoin will be=')
```

Scilab code Exa 5.10 program to find out expression for tao_{in}

```
1 // example: -5.10. page no. -250.
2 //program to find out expression for taoin.
3 syms P1 P2 S11 S22 S12 S21 taol taoin L1 l2;
4 P1=S11; // path one.
5 P2=S21*S12*taol; //path second.
6 L1=taol*S22; // loop gain for path 1.
7 L2=L1^2; // loop gain taking two at a time.(but
   only one loop wiil exist.i.e=L1.)
8 L2=0;
9 // from mason's gain formula.
10 taoin=(S11*(1-taol*S22)+(S21*taol*S12))/(1-taol*S22)
   ;
11 // result
12 disp(taoin)
```

Scilab code Exa 5.11 determine amplitude of forward and backward wave

```
1 // example: -5.11, page no. -264.
2 // program to determine the amplitude of the forward
  and backward travelling TE10 modes and the input
  resistance.
3 syms Io a b x y z h1 e1 J P1 A1p A1m pi Z1 delta b1
  j P Rin;
4 e1=sin(pi*x/a); // in y direction.
5 h1=-sin(pi*x/a)/Z1; // in z direction.
6 P1=(2/Z1)*integ(integ(sin(pi*x/a)^2,x),y);
7 P1=limit(P1,x,a)-limit(P1,x,0);
8 P1=limit(P1,y,b)-limit(P1,y,0);
9 // taking sin(2*pi)=0. we get,
10 P1=a*b/Z1;
11 A1p=(-1/P1)*Io*y; // as for x, it will be one at x=a
  /2 and 1 for z at z=0;
12 A1p=limit(A1p,y,b)-limit(A1p,y,0);
13 A1m=(-1/P1)*Io*y; // as for x, it will be one at x=a
  /2 and 1 for z at z=0;
14 A1m=limit(A1m,y,b)-limit(A1m,y,0);
15 P=integ(integ(((A1p^2)/Z1)*sin(pi*x/a)^2,x),y);
16 P=limit(P,x,a)-limit(P,x,0);
17 P=limit(P,y,b)-limit(P,y,0);
18 // taking sin(2*pi)=0. we get,
19 P=(b*(Io^2)*Z1*pi)/(2*a*pi);
20 Rin=2*P/(Io^2);
21 disp(A1p, 'amplitude of the forward travelling wave =
  ')
22 disp(A1m, 'amplitude of the backward travelling wave
  = ')
23 disp(Rin, 'input resistance seen by the probe = ')
```

Scilab code Exa 5.12 find excitation coefficient of forward wave TE10

```

1 // example:-5.12,page no.-265.
2 // program to find the excitation coefficient of the
  forward travelling TE10 mode.
3 syms M Pm uo w j a b Io x y z ro pi Z1 h1 A1p P1 no
  ko uo eo;
4 ko=w*sqrt(uo*eo);
5 no=sqrt(uo/eo);
6 h1=sin(pi*x/a)*(-1/Z1); // in x direction.
7 P1=(2/Z1)*integ(integ(sin(pi*x/a)^2,x),y);
8 P1=limit(P1,x,a)-limit(P1,x,0);
9 P1=limit(P1,y,b)-limit(P1,y,0);
10 // taking sin(2*pi)=0. we get ,
11 P1=a*b/Z1;
12 Pm=Io*pi*(ro^2); // defined at x=a/2,y=b/2 and z=0;
13 M=j*w*uo*Pm;
14 A1p=(1/P1)*(-(1/Z1)*M);
15 disp(A1p,'the forward wave excitation coefficient
  will be = ')
16 disp("!! NOTE:-replace w=sqrt(uo*eo) and no=sqrt(uo/
  eo),the answer will match !! ")
17 disp(" NOTE:- on integrating , x component will
  become one at x=a/2,y component will become one
  at y=b/2 and z component will become one at z=0."
  )

```

Chapter 6

IMPEDENCE MATCHING AND TUNNING

Scilab code Exa 6.1 program to design an L section matching network

```
1 // example: -6.1, page no. -284.
2 // program to design an L section matching network
  to match a series RC load.
3 Zl=200-%i*100; // load impedance.
4 Rl=200; Xl=-100; f=500*10^6; Zo=100;
5 B1=(Xl+sqrt(Rl/Zo)*sqrt(Rl^2+Xl^2-(Rl*Zo)))/(Rl^2+Xl
  ^2);
6 B2=(Xl-sqrt(Rl/Zo)*sqrt(Rl^2+Xl^2-(Rl*Zo)))/(Rl^2+Xl
  ^2);
7 C1=(B1/(2*pi*f))*10^12;
8 L2=(-1/(B2*2*pi*f))*10^9;
9 X1=(1/B1)+((Xl*Zo)/Rl)-(Zo/(B1*R1));
10 X2=(1/B2)+((Xl*Zo)/R1)-(Zo/(B2*R1));
11 L1=(X1/(2*pi*f))*10^9;
12 C2=(-1/(X2*2*pi*f))*10^12;
13 disp(L1,'inductor of first circuit in nH = ')
14 disp(C1,'capacitor of the first circuit in pF = ')
15 disp(L2,'inductor of second circuit in nH = ')
16 disp(C2,'capacitor of the second circuit in pF = ')
```

```

17 disp("NOTE:—for above specific problem  $R_l > Z_o$ ,
        positive X implies inductor, negative X implies
        capacitor, positive B implies capacitor and
        negative B implies inductor.")

```

Scilab code Exa 6.5 design quarter wave matching transformer

```

1 //example:—6.5, page no.—304.
2 //program to design a single section quarter wave
  matching transformer.
3 Zl=10; // load impedance.
4 Zo=50; // characteristic impedance.
5 fo=3*10^9; swr=1.5; // maximum limit of swr.
6 Z1=sqrt(Zo*Zl); // characteristic impedance of the
  matching section.
7 taom=(swr-1)/(swr+1);
8 frac_bw=2-(4/%pi)*acos((taom/sqrt(1-taom^2))*(2*sqrt
  (Zo*Zl)/abs(Zl-Zo))); // fractional bandwidth.
9 disp(Z1, 'characteristic impedance of matching
  section = ')
10 disp(frac_bw, 'fractional bandwidth = ')

```

Scilab code Exa 6.6 program to evaluate the worst case percent error

```

1 // example:—6.6, page no.—307.
2 // program to evaluate the worst case percent error
  in computing magnitude of reflection coefficient.
3 Z1=100; Z2=150; Zl=225;
4 tao_1=(Z2-Z1)/(Z2+Z1);
5 tao_2=(Z1-Z2)/(Z1+Z2);
6 tao_exact=(tao_1+tao_2)/(1+tao_1*tao_2); // this
  results as angle is taken zero.

```

```

7 tao_approx=tao_1+tao_2; // this results as angle is
   taken zero.
8 error=abs(((tao_exact-tao_approx)/tao_exact)*100);
9 disp(tao_approx,'approximate value of reflection
   coefficient is = ')
10 disp(error,'the error in percent is about = ')

```

Scilab code Exa 6.7 design three section binomial transformer

```

1 // example: -6.7, page no. -312.
2 // program to design three section binomial
   transformer.
3 Z1=50; Zo=100; N=3; taom=0.05;
4 A=(2^-N)*abs((Z1-Zo)/(Z1+Zo));
5 frac_bw=2-(4/%pi)*acos(0.5*(taom/A)^2);
6 for c=1
7     Z1=Zo*((Z1/Zo)^((2^-N)*(c^N)));
8     disp(Z1,'Z1 = ')
9 end
10 for c=3^(1/3)
11     Z2=Z1*((Z1/Zo)^((2^-N)*(c^N)));
12     disp(Z2,'Z2 = ')
13 end
14 for c=3^(1/3)
15     Z3=Z2*((Z1/Zo)^((2^-N)*(c^N)));
16     disp(Z3,'Z3 = ')
17 end

```

Scilab code Exa 6.8 design three section chebysev transformer

```

1 // example: -6.8, page no. -316.
2 // program to design a three section chebysev
   transformer.

```



```

3 Z1=100;Zo=50;taom=0.05;N=3;A=0.05;
4 thetam=asec(cosh((1/N)*acosh((1/taom)*abs((Z1-Zo)/(
      Z1+Zo)))))*(180/%pi);
5 x=(cosh((1/N)*acosh((1/taom)*abs((Z1-Zo)/(Z1+Zo))))))
6 tao_o=A*(x^3)/2;
7 tao_1=(3*A*(x^3-x))/2; // from symmetry tao_3=tao_0
      ;
8 Z1=Zo*((1+tao_o)/(1-tao_o));
9 Z2=Z1*((1+tao_1)/(1-tao_1));
10 Z3=Z1*((1-tao_o)/(1+tao_o));
11 disp(Z1,Z2,Z3,'the characteristic impedences are = '
      )

```

Scilab code Exa 6.9 design triangular taper and a klopfenstein taper

```

1 //example:-6.9,page no.-323.
2 //program to designa triangular taper and a
      klopfenstein taper.
3 taom=0.02;Z1=50;Zo=100;
4 tao_o=0.5*log(Z1/Zo);
5 A=acosh(tao_o/taom);
6 A=real(A);
7 disp(tao_o,'tao_o = ')
8 disp(A,'A = ')

```

Chapter 7

MICROWAVE RESONATORS

Scilab code Exa 7.1 program to compare the Q factor

```
1 // example: -7.1, page no. -339.
2 //program to compare the Q of an air filled and
   teflon filled coaxial line resonator.
3 sigma=5.813*10^7; muo=4*pi*10^-7; f=5*10^9; eta=377; a
   =1*10^-3; b=4*10^-3;
4 omega=2*pi*f; ko=104.7; B=104.7; alpha=0.022;
5 Rs=sqrt((omega*muo)/(2*sigma));
6 alphaca=(Rs/(2*eta*log(b/a)))*((1/a)+(1/b)); //
   attenuation due to conductor loss for air filled
   line.
7 eipsilar=2.08; tandelta=0.0004; // for teflon filled
   line.
8 alphact=((Rs*sqrt(2.08)*0.01)/(2*eta*log(b/a)))*((1/
   a)+(1/b)); // attenuation due to conductor loss
   for teflon filled line.
9 alphada=0; // for air filled line.
10 alphadt=ko*(sqrt(eipsilar)/2)*tandelta;
11 Qair=B/(2*alpha);
12 B=B*sqrt(eipsilar);
13 alpha=0.062;
14 Qteflon=B/(2*alpha);
```

```

15 disp(Qair, 'Qair = ')
16 disp(Qteflon, 'Qteflon = ')
17 disp("conclusion:-Qair is almost twice that of
      Qteflon")

```

Scilab code Exa 7.2 program to compute length and Q of the resonator

```

1 //example:-7.2,page no.-342.
2 // program to compute the length of the line for
  resonance at 5 GHZ and the Q of the resonator.
3 W=0.0049;c=3*10^8;f=5*10^9;Zo=50;eipsilar=2.2;ko
  =104.7;tandelta=0.001;
4 Rs=0.0184; // taken from example 7.1.
5 eipsilae=1.87; // effective permittivity.
6 l=c/(2*f*sqrt(eipsilae)); // resonator length.
7 B=(2*pi*f*sqrt(eipsilae))/c;
8 alphac=Rs/(Zo*W);
9 alphad=(ko*eipsilar*(eipsilae-1)*tandelta)/(2*sqrt(
  eipsilae)*(eipsilar-1));
10 alpha=alphac+alphad;
11 Q=B/(2*alpha);
12 disp(l, 'length of the line in meter = ')
13 disp(Q, 'Q of the resonator = ')

```

Scilab code Exa 7.3 program to find required length and other

```

1 // example:-7.3,page no.-347.
2 // program to find required length ,d and Q for l=1
  and l=2 resonator mode.
3 a=0.04755;b=0.02215;eipsilar=2.25;tandelta=0.0004;f
  =5*10^9;c=3*10^8;
4 k=(2*pi*f*sqrt(eipsilar))/c // wave number.
5 for l=1:1:2

```

```

6 d=(1*%pi)/sqrt((k^2)-((%pi/b)^2)); // m=1 & n=0 mode
7 disp(d, 'd in meter = ')
8 end
9 eta=377/sqrt(eipsilar);
10 Qc1=3380; // l=1.
11 Qc2=3864; // l=2.
12 Qd=2500; // Q due to dielectric loss only.
13 Q1=((1/Qc1)+(1/Qd))^-1; // for l=1.
14 Q2=((1/Qc2)+(1/Qd))^-1; // for l=2.
15 disp(Q1, 'Q1 = ');
16 disp(Q2, 'Q2 = ')

```

Scilab code Exa 7.4 program to find dimension and Q

```

1 //example: -7.4, page no. -353.
2 // program to find dimension and Q;
3 f=5*10^9; c=3*10^8; p01=3.832; sigma=5.813*10^7; muo=4*
   %pi*10^-7;
4 eipsilar=2.25;
5 // mode TE011. and d=2a.
6 omega=2*%pi*f;
7 eta=377;
8 lamda=c/f;
9 k=(2*%pi)/lamda;
10 // f=(c/(2*%pi))*sqrt((p01/a)^2+(%pi/(2*a))^2); as d
   =2a given
11 a=sqrt((p01)^2+(%pi/2)^2)/k;
12 Rs=sqrt((omega*muo)/(2*sigma))
13 Qc=(k*a*eta)/(2*Rs); // for m=1, n=0 and d=2a.
14 disp(a, 'a in meter = ')
15 disp(Qc, 'Qc = ')

```

Scilab code Exa 7.5 program to find resonant frequency and Q

```
1 //example:-7.5,page no.-358.
2 // program to find the resonant frequency and Q for
  TE01delta mode.
3 clear
4 clc
5 delta=0.001;eipsilar=95;a=0.413;L=0.008255;c=3*10^8;
  f = 1.133D+09;
6 //tan((B*L)/2)=alpha/beta.
7 ko=(2*pi*f)/c;
8 alpha=sqrt((2.405/a)^2-(ko)^2);
9 B=sqrt((eipsilar*(ko)^2)-(2.405/a)^2); // beta
10 f1=((c*2.405)/(2*pi*sqrt(eipsilar)*a))*10^-7;
11 f2=((c*2.405)/(2*pi*a))*10^-7;
12 disp(f1,'f1 in GHZ = ')
13 disp(f2,'f2 in GHZ = ')
14 Q=1/tan(delta);
15 disp(Q,'approx. value of Q due to dielectric loss =
  ')
```

Scilab code Exa 7.6 program to find the mode number and Q

```
1 // example:-7.6,page no.-361.
2 // program to find the mode number and Q of given
  resonator.
3 fo=94*10^9;d=0.04;c=3*10^8;muo=4*pi*10^-7;sigma
  =5.813*10^7;
4 l=(2*d*fo)/c; // mode number.
5 Rs=sqrt((2*pi*fo*muo)/(2*sigma));
6 Q=(pi*l*377)/(4*Rs);
7 disp(l,'mode number = ')
8 disp(Q,'Q = ')
```

Scilab code Exa 7.7 program to find value of the coupling capacitor

```

1 // example: -7.7, page no. -367.
2 // program to find the value of the coupling
  capacitor required for critical coupling.
3 l=0.02175; Zo=50; eipsilae=1.9; c=3*10^8;
4 fo=c/(2*l*sqrt(eipsilae)); // first resonant
  frequency will occur when the resonator ia about
  l=lamdag/2 in length.
5 lamdag=c/fo;
6 alpha=1/8.7; // in Np/m.
7 Q=%pi/(2*l*alpha);
8 bc=sqrt(%pi/(2*Q));
9 C=bc/(2*%pi*fo*Zo)*10^12;
10 disp(C, 'coupling capacitor in pF = ')
11 C=bc/(2*%pi*fo*Zo);
12 w1=atan(2*%pi*fo*C*Zo)*c/(1*sqrt(eipsilae)); //
  from equation tan(B*l)=-bc;
13 w1=w1*10^-8;
14 disp(w1, 'frequency in GHZ = ')

```

Scilab code Exa 7.8 derive expression for change in resonant frequency

```

1 // example: -7.8, page no. -373.
2 // program to derive an expression for the change in
  resonant frequency.
3 syms Ey Hx Hz A Zte n a pi x z d j k t y er eo c wo
  w b;
4 Ey=A*sin((pi*x)/a)*sin((pi*z)/d);
5 Hx=((-j*A)/Zte)*sin((pi*x)/a)*cos((pi*z)/d);
6 Hz=((j*pi*A)/(k*n*a))*cos((pi*x)/a)*sin((pi*z)/d);
7 Ey=Ey^2;

```

```

8 //c=(er-1)*eo;
9 w=c*integ(integ(integ(Ey,z),y),x);
10 w=limit(w,z,d)-limit(w,z,0);
11 w=limit(w,y,t)-limit(w,y,0);
12 w=limit(w,x,a)-limit(w,x,0);
13 disp(w)
14 // as  $\sin(2\pi)=0$ ; then last term of above result
    will be:-
15 w=(c*A^2*a*t*d)/4;
16 disp(w, 'on taking  $\sin(2\pi)=0$  , w becomes = ')
17 wo=((a*b*d*eo)/2)*A^2;
18 deltaw=(w-wo)/wo;
19 disp(deltaw, 'fractional change in resonant frequency
    = ')

```

Scilab code Exa 7.9 derive expression for change in resonant frequency

```

1 // example:-7.9, page no.-376.
2 // program to derive an expression for the change in
    resonant frequency.
3 syms Ey Hx Hz A Zte n a pi x z d j eo c wo w b l ro;
4 Ey=A*sin((pi*x)/a)*sin((pi*z)/d);
5 Hx=((-j*A)/Zte)*sin((pi*x)/a)*cos((pi*z)/d);
6 Hz=((j*pi*A)/(k*n*a))*cos((pi*x)/a)*sin((pi*z)/d);
7 Ey=A; // at  $x=a/2, y, z=d/2$ ;
8 Hx=0; // at  $x=a/2, y, z=d/2$ ;
9 Hz=0; // at  $x=a/2, y, z=d/2$ ;
10 //where w is perturbed resonant frequency and wo is
    unperturbed resonant frequency.
11 w=-eo*A^2*pi*l*ro^2;
12 wo=(a*b*eo*d*A^2)/2;
13 deltaw=(w-wo)/wo;
14 disp(deltaw, 'the perturbation in resonant frequency
    w.r.t wo = ')

```

Chapter 8

POWER DIVIDERS DIRECTIONAL COUPLERS AND HYBRIDS

check Appendix [AP 1](#) for dependency:

```
parallel_impedance.sce
```

Scilab code Exa 8.1 program to compute the reflection coefficients

```
1 // example: -8.1, page no. -392.
2 // program to compute the reflection coefficients
  seen looking in to the output port.
3 // as the power is divided in to 2:1 ratio. and Pin
  =(1/2)*Vo^2/Zo;
4 // so ,P1=(1/3)*Pin;and P2=(2/3)*Pin .....( i)
5 Zo=50;
6 Z1=3*Zo; // from above condition .....( i)
7 Z2=(3/2)*Zo;
8 Zin=parallel_impedance(Z1,Z2); // input impedance
  to the junction.
9 if Zin==Zo
10   disp("input is matched to the 50 ohm sources")
```



```

11 else
12     disp("not matched")
13 end
14 Zin1=parallel_impedence(Zo,Z2); // looking in to the
    150 ohm source.
15 Zin2=parallel_impedence(Zo,Z1); // looking in to the
    75 ohm source.
16 tao1=(Zin1-Z1)/(Zin1+Z1);
17 tao2=(Zin2-Z2)/(Zin2+Z2);
18 disp(tao1,'reflection coefficient looking at 150 ohm
    line = ')
19 disp(tao2,'reflection coefficient looking at 75 ohm
    line = ')

```

Scilab code Exa 8.2 design equi split wilkinson power divider

```

1 //example:-8.2,page no.-398.
2 // program to design an equi-split wilkinson power
    divider for 50 ohm system impedance.
3 Zo=50;
4 Z=sqrt(2)*Zo; // impedance of quarter wave
    transmission line.
5 R=2*Zo; // shunt resistor.
6 disp(R,'the shunt resistance value should be in ohm
    = ')
7 disp(Z,'the quarter wave transmission line in the
    divide should have a characteristic impedance in
    ohm = ')

```

Scilab code Exa 8.3 design bethe hole coupler for x band waveguide

```

1 // example:-8.3,page no.-404.

```

```

2 // program to design bethe-hole coupler for x-band
  wave guide.
3 f=9*10^9;C=20;a=0.02286;b=0.01016;Ko=188.5;B=129;Z10
  =550.9;P10=4.22*10^-7;lamdao=0.0333;uo=4*%pi
  *10^-7;eo=8.854*10^-12;w=2*%pi*f;
4 s=(a/%pi)*asin(lamdao/sqrt(2*(lamdao^2-a^2)))*10^3;
5 // a=10*b;// as C=20db; // take x=a/b; so x=10;
6 ro=(P10/((10*w)*(((2*eo/3)+(4*uo)/(3*Z10^2))*0.944)
  -((4*%pi^2*uo*0.056)/(3*B^2*a^2*Z10^2))))^(1/3)
  *10^3;
7 disp(s,'the aperture position in mm = ')
8 disp(ro,'the aperture size in mm = ')
9 disp("NOTE:-the above shown results completes the
  design of the betha hole coupler.")

```

Scilab code Exa 8.4 program to design a four hole chebysev coupler

```

1 //example:-8.4,page no.-410.
2 // program to design a four hole chebysev coupler in
  x-band wave guide using round aperture located
  at s=a/4.
3 a=0.02286;b=0.01016;lamdao=0.0333;ko=188.5;bta=129;
  Z10=550.9;P10=4.22*10^-7;f=9*10^9;no=377;N=3;
4 s=a/4;
5 kf=((2*ko)/(3*no*P10))*((sin(%pi*s/a)^2)-(2*(bta^2)
  /(ko^2))*((sin(%pi*s/a)^2)+((%pi^2)/((bta^2)*(a
  ^2))))*(cos(%pi*s/a)^2));
6 kf=abs(kf)
7 kb=((2*ko)/(3*no*P10))*((sin(%pi*s/a)^2)+(2*(bta^2)
  /(ko^2))*((sin(%pi*s/a)^2)-((%pi^2)/((bta^2)*(a
  ^2))))*(cos(%pi*s/a)^2));
8 kb=abs(kb)
9 x=cosh(acosh(100)/3); // x=sec(thetam).
10 thetam=asec(x)*180/%pi; // so ,thetam=70.6 and 109.4
  at the band edge.

```

```

11 k=10^(-171.94/20);
12 ro=((k/2)^(1/3))*x*1000;
13 r1=(1.5*k*((x^3)-x))^(1/3)*1000;
14 disp(kf,'kf = ')
15 disp(kb,'kb = ')
16 disp(thetam,'thetam in degree = ')
17 disp(ro,'ro in mm = ')
18 disp(r1,'r1 in mm = ')

```

Scilab code Exa 8.5 design 50 ohm branchline quadrature hybrid junc

```

1 // example:-8.5,page no.-415.
2 // program to design a 50 ohm branch-line quadrature
  hybrid junction.
3 Zo=50;
4 Z=Zo/sqrt(2);
5 disp(Z,'the branch line impedance in ohm will be = ')
  )

```

Scilab code Exa 8.6 determine even and odd mode characteristic impeden

```

1 // example:-8.6,page no.-419.
2 // program to determine the even and odd mode
  characteristic impedance.
3 syms C A d W C11 C12 Ce Co v eo er s b uo Zoe Zoo
  eipsila;
4 C=A*eipsila/d;
5 C11=(eo*er*W)/((b-s)/2)+(eo*er*W)/((b+s)/2);
6 C12=er*eo*W/s;
7 Ce==C11;
8 Co=C11+2*C12
9 v=1/sqrt(er*eo*uo);
10 Zoe=1/(v*C11); // as Ce=C11;

```

```

11 Zoo=1/(v*Co);
12 disp(Zoe, 'Zoe = ')
13 disp(Zoo, 'zoo = ')

```

Scilab code Exa 8.7 design a 20 db single section coupled line coupler

```

1 // example:-8.7,page no.-425.
2 //design a 20 db single section coupled line coupler
  in stripline.
3 C=10^(-20/20);f=3*10^9;eipsila=2.56;Zo=50;b=0.00158;
4 Zoe=Zo*sqrt((1+C)/(1-C));
5 Zoo=Zo*sqrt((1-C)/(1+C));
6 Zoe=eipsila*Zoe;
7 Zoo=eipsila*Zoo;
8 x=0.72; //x=w/b.
9 y=0.34; // y=s/b.
10 w=0.72*b*100;
11 s=0.34*b*100;
12 disp(w, 'conductor width in cm = ')
13 disp(s, 'conductor seperation in cm = ')

```

Scilab code Exa 8.8 design a three section 20 db coupler

```

1 // example:-8.8,page no.-428.
2 // design a three section 20 db coupler with a
  binomial response.
3 Zo=50;f=3*10^9;N=3;
4 syms C C1 C2 theta;
5 C=10^(-20/20);
6 disp("for a maximally flat response for a three=
  section coupler douple derivative of C will be
  zero.")
7 C1=0.0125;C2=0.125;C3=0.0125;

```

```

8 Zoe1=Zo*sqrt((1+C1)/(1-C1));
9 Zoe3=Zo*sqrt((1+C3)/(1-C3));
10 Zoo1=Zo*sqrt((1-C1)/(1+C1));
11 Zoo3=Zo*sqrt((1-C1)/(1+C1));
12 Zoe2=Zo*sqrt((1+C2)/(1-C2));
13 Zoo2=Zo*sqrt((1+C2)/(1-C2));
14 disp("the even and odd mode characteristic
      impedences for each section are = ")
15 disp(Zoe1, 'Zoe1 = ')
16 disp(Zoo1, 'Zoo1 = ')
17 disp(Zoe2, 'Zoe2 = ')
18 disp(Zoo2, 'Zoo2 = ')
19 disp(Zoe3, 'Zoe3 = ')
20 disp(Zoo3, 'Zoo3 = ')

```

Scilab code Exa 8.9 design a 3 dB 50 ohm langer coupler

```

1 //example:-8.9,page no.-434.
2 // program to design a 3 dB 50 ohm langer coupler
  for operation at 5 GHZ.
3 f=5*10^9;C=10^(-3/20);
4 Zo=50;
5 Zoe=((((4*C)-3+sqrt(9-(8*C^2)))/((2*C)*sqrt((1-C)/(1+
  C))))*Zo;
6 Zoo=((((4*C)+3-sqrt(9-(8*C^2)))/((2*C)*sqrt((1+C)/(1-
  C))))*Zo;
7 disp(Zoe, 'even mode characteristic impedance of a
  pair of adjacent coupled lines is = ')
8 disp(Zoo, 'even mode characteristic impedance of a
  pair of adjacent coupled lines is = ')

```

Scilab code Exa 8.10 design 180 deg ring hybrid for 50 ohm system imped

```

1 // example: -8.10, page no. -440.
2 // design a 180 deg. ring hybrid for a 50 ohm system
  impedance.
3 Zo=50;
4 Z=sqrt(2)*Zo;
5 disp(Z, 'the characteristic impedance of the ring
  transmission line in ohm is = ')

```

Scilab code Exa 8.11 calculate even and odd mode characteristic impeden

```

1 // example: -8.11, page no. -444.
2 // calculate the even and odd-mode characteristic
  impedences for a tapered coupled line 180 deg.
  hybrid for a 3 db coupling ratio and a 50 ohm
  characteristic impedance.
3 alpha=0.707; bta=0.707; Zo=50;
4 k=(1-alpha)/(1+alpha);
5 Zoe=Zo/k;
6 Zoo=k*Zo;
7 disp(Zoo, 'Zoo = ')
8 disp(Zoe, 'at (Z=L) the characteristic impedences of
  the coupled line must be = ')
9 disp('at Z=0, there will be no coupling')

```

Chapter 9

MICROWAVE FILTERS

Scilab code Exa 9.1 program to compute the propagation constant

```
1 // example: -9.1, page no. -462.
2 // program to compute the propagation constant, phase
   velocity and bloch impedance.
3 Co=2.666*10^-12;
4 d=0.01; c=3*10^8;
5 Zo=50; f=3*10^9;
6 p=(Co*Zo*c)/(2*d); // constant of equation given
   below.
7 y=0:0.001:0.96;
8 x=acos(cos(y)-p.*y.*sin(y)); // x=ko*d; and y=beta*d
   ;
9 subplot(2,1,1)
10 plot2d(x,y,style=2,rect=[-%pi,0,%pi,0.96])
11 plot2d(-x,y,style=2,rect=[-%pi,0,%pi,0.96])
12 xtitle("k-beta diagram for first pass band ","beta*d
   ","ko*d")
13 y=3:0.001:4;
14 x=acos(cos(y)-p.*y.*sin(y)); // x=ko*d; and y=beta*d
   ;
15 ko=(2*%pi*f)/c;
16 subplot(2,1,2)
```

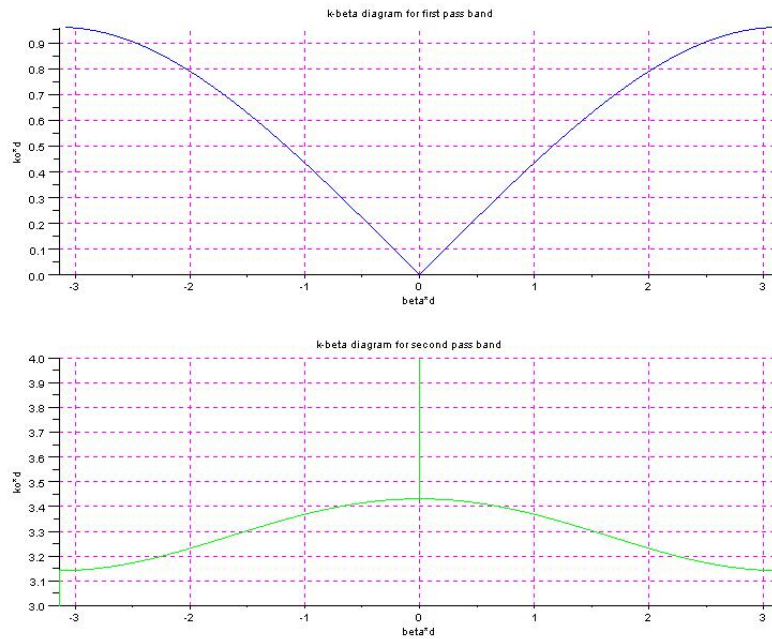


Figure 9.1: program to compute the propagation constant

```

17 plot2d(x,y,style=3,rect=[-%pi,3,%pi,4])
18 plot2d(-x,y,style=3,rect=[-%pi,3,%pi,4])
19 xtitle("k-beta diagram for second pass band ", "beta*
    d", "ko*d")
20 bta=(acos(cos(ko*d)-p*ko*d*sin(ko*d)))/d;
21 vp=(ko*c)/150; // phase velocity.
22 b=2*%pi*f*Co*Zo;
23 A=cos(ko*d)-(b/2)*sin(ko*d);
24 B=%i*(sin(ko*d)+(b/2)*cos(ko*d)-(b/2));
25 Zb=(B*Zo)/sqrt(A^2-1); // bloch impedance.
26 disp(Zb,'Bloch impedance = ')
27 disp(vp,'phase velocity = ')
28 disp(bta,'propagation constant = ')

```

Scilab code Exa 9.2 program to design a low pass composite filter

```
1 // example: -9.2, page no. -473.
2 // program to design a low pass composite filter
  with cutoff frequency of 2 MHZ.
3 fc=2*10^6; f=2.05*10^6; Ro=75;
4 L=(2*Ro)/(2*%pi*fc);
5 C=2/(Ro*2*%pi*fc);
6 for m=sqrt(1-(fc/f)^2)
7 x=m*L/2;
8 y=m*C;
9 z=((1-m^2)/(4*m))*L; // x,y,z are design parameter
  assumed.
10 disp(x,y,z, 'design parameter for m=0.2195 ')
11 end
12 for m=0.6
13 x=m*L/2;
14 y=m*C/2;
15 z=((1-m^2)/(2*m))*L; // x,y,z are design parameter
  assumed.
16 disp(x,y,z, 'design parameter for m=0.6 ')
17 end
```

Scilab code Exa 9.3 program to find out number of filter elements

```
1 // example: -9.3, page no. -482.
2 // program to find out number of filter elements
  required.
3 fc=8*10^9; f=11*10^9;
4 w=2*%pi*f;
5 wc=2*%pi*fc;
6 x=abs(w/wc)-1;
```

```
7 disp(x,"from table we see that an attenuation of 20
   db at this frequency requires that N>=8 for x =
   ")
```

Scilab code Exa 9.4 program to design a maximum flat low pass filter

```
1 // example:-9.4,page no.-488.
2 // program to design a maximum flat low pass filter
   with cut off frequency of 2 GHZ.
3 fc=2*10^9;f=3*10^9;
4 w=2*%pi*f;
5 wc=2*%pi*fc;
6 x=abs(w/wc)-1;
7 // from table we can see that N=5 will be sufficient
   .
8 // then prototype element values are:-
9 g1=0.618;g2=1.618;g3=2.000;g4=1.618;g5=0.618;
10 disp(g1,'g1 = ')
11 disp(g2,'g2 = ')
12 disp(g3,'g3 = ')
13 disp(g4,'g4 = ')
14 disp(g5,'g5 = ')
```

Scilab code Exa 9.5 program to design a band pass filter

```
1 // example:-9.5,page no.-492.
2 // design a band pass filter having a 0.5 db equal
   ripple respnse with N=3.
3 N=3;Zo=50;f=1*10^9;delta=1*10^8;
4 L1=1.596;L3=1.5963;C2=1.0967;R1=1.000;
5 L_1=(L1*Zo)/(2*%pi*f*delta);
6 C_1=delta/(2*%pi*f*L1*Zo);
7 L_2=(delta*Zo)/(2*%pi*f*C2);
```

```

8 C_2=C2/(2*%pi*f*delta*Zo);
9 L_3=(L3*Zo)/(2*%pi*f*delta);
10 C_3=delta/(2*%pi*f*L3*Zo);
11 disp(L_1)
12 disp(L_2)
13 disp(C_1)
14 disp(C_2)
15 disp(L_3)
16 disp(C_3)

```

Scilab code Exa 9.6 design a low pass filter using microstrip lines

```

1 // example: -9.6, page no. -498.
2 // design a low pass filter for fabrication using
  microstrip lines.
3 disp("from table, the normalized low pass prototype
  element values are = ")
4 L1=3.3487; C2=0.7117; L3=3.3487; R1=1.0000;
5 n=1+(1/3.3487);
6 disp(L1)
7 disp(R1)
8 disp(C2)
9 disp(L3)
10 disp(n)

```

Scilab code Exa 9.7 design a stepped impedance low pass filter

```

1 // example: -9.7, page no. -503.
2 // design a stepped-impedance low pass filter having
  a maximally flat response and a cut-off
  frequency of 2.5 GHz.
3 w=4*10^9; wc=2.5*10^9; Zh=150; Ro=50; Z1=10;

```

```

4 C1=0.517;L2=1.414;C3=1.932;L4=1.932;C5=1.414;L6
   =0.517;
5 // above values are taken from table.
6 // for finding electrical lengths.
7 x1=(C1*Zl/Ro)*(180/%pi);
8 x2=(L2*Ro/Zh)*(180/%pi);
9 x3=(C3*Zl/Ro)*(180/%pi);
10 x4=(L4*Ro/Zh)*(180/%pi);
11 x5=(C5*Zl/Ro)*(180/%pi);
12 x6=(L6*Ro/Zh)*(180/%pi);
13 disp(x1)
14 disp(x2)
15 disp(x3)
16 disp(x4)
17 disp(x5)
18 disp(x6)

```

Scilab code Exa 9.8 design a coupled line band pass filter

```

1 // example: -9.8, page no. -516.
2 // design a coupled line band pass filter with N=3.
3 delta=0.1;f=1.8*10^9;fo=2*10^9;Zo=50;fc=1;
4 f=(1/delta)*((f/fo)-(fo/f));
5 x=abs(f/fc)-1; // the value on the horizontal scale.
6 attntn=20; // from above values.
7 disp(attntn,'attenuation in db = ')

```

Scilab code Exa 9.9 design a bandpass filter

```

1 // example: -9.9, page no. -521.
2 // design a bandpass filter using three quarter wave
   open circuit stubs.
3 f=2*10^9;delta=0.15;Zo=50;N=3;gn=1.5963;

```

```

4 Zon=4*Zo/(%pi*gn*delta);
5 Z_on=(%pi*Zo*delta)/(4*gn);
6 disp(Zon,'the cahracteristic impedance of a bandpass
   filter is = ')
7 disp(Z_on,'for a bandpass filter using short
   circuited stub resonators ,the corresponding
   result is = ')

```

Scilab code Exa 9.10 design a bandpass filter using capacitive coupled

```

1 // example: -9.10, page no. -524.
2 // design a bandpass filter using capacitive coupled
   resonators ,with a 0.5 db equal passband
   haracteristic .
3 fo=2*10^9; delta=0.1; Zo=50; f=2.2*10^9; g1=1.5963; g2
   =1.0967; g3=1.5963; g4=1;
4 f=(1/delta)*((f/fo)-(fo/f));
5 x0=sqrt((%pi*delta)/(2*g1))/Zo; // x0=ZoJ1;
6 x1=((%pi*delta)/(2*sqrt(g1*g2)))/Zo; // x0=ZoJn;
7 B0=x0/(1-(Zo*x0)^2)
8 B1=x1/(1-(Zo*x1)^2)
9 theta0=(%pi-0.5*(atan(2*Zo*B0)+atan(2*Zo*B1)))*(180/
   %pi);
10 C0=(B0/(2*%pi*fo))*10^12;
11 disp(theta0,'thetao in degree = ')
12 disp(C0,'the coupling capacitor value in PF = ')

```

Chapter 10

THEORY AND DESIGN OF FERRIMAGNETIC COMPONENTS

Scilab code Exa 10.1 calculate and plot phase and attenuation constants

```
1 // example: -10.1; page no. -547.
2 // problem to calculate and plot the phase and
  attenuation constants for RHCP & LHCP plane wave.
3 M=1800; // M=4*pi*Ms;
4 deltaH=75; eo=8.854*10^-12; muo=4*pi*10^-7; c=3*10^8;
5 Ho=3570; er=14; tandelta=0.001;
6 fo=(2.8*10^9)/Ho; // IN GHZ.
7 wo=2*pi*fo;
8 fm=(2.8*10^9)/M; // IN GHZ.
9 wm=2*pi*fm;
10 mue=muo*(1+(wo*wm)/(wo^2-wm^2));
11 e=eo*er*(1-i*tandelta);
12 f=0:1000000:20*10^9;
13 w=2*pi.*f;
14 k=muo*((w.*wm)/(wo^2-w^2));
15 gama=i*w*sqrt(e.*(mue-k));
16 alpha=abs(real(gama));
```

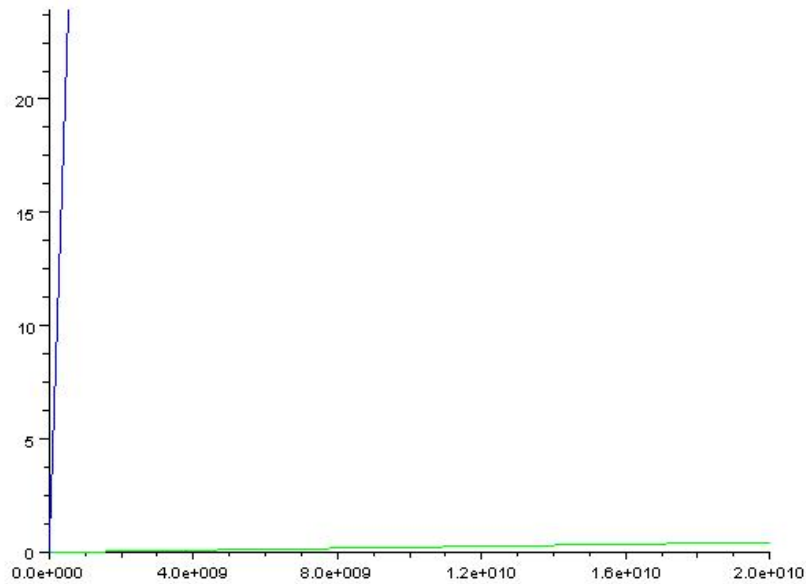


Figure 10.1: calculate and plot phase and attenuation constants

```

17 bta=abs(imag(gama));
18 plot2d(f,gama,style=3,rect=[0,0,20*10^9,24])
19 plot2d(f,bta,style=2,rect=[0,0,20*10^9,24])

```

Scilab code Exa 10.2 program to design an e plane resonance isolator

```

1 //example: -10.2, page no. -559.
2 // program to design an e plane resonance isolator in
  x band waveguide.
3 er=13; revatt=30;
4 deltaH=200; x=1700; // x=4*%pi*Ms.

```

```

5 f=10*10^9;alpha_=12.4; // from graph 10.13.
6 L=revatt/alpha_;
7 alpha_1=27/L;
8 disp(L,'for total reverse attenuation of 20 db,the
length of the slab in cm must be = ')
9 disp(alpha_1,'for total reverse attenuation to be at
least 27 db, alpha_ in db/cm be > ')

```

Scilab code Exa 10.3 program to design a resonance isolator

```

1 //example:-10.3.page no.-560.
2 // program to design a resonance isolator using the
H-plane ferrite slab geometry in x-band.
3 f=10*10^9;delta_sbys=0.01;forpims=1700;deltaH=200;
4 revatt=30;ko=(2*pi*f)/(3*10^8);
5 Ho=f/(2.8*10^9);
6 // for x-band waveguide , a=2.286 cm.
7 a=2.286;
8 kc=(pi*100)/a;
9 betao=sqrt(ko^2-kc^2);
10 x=(1/pi)*atan(kc/betao); // x=c/a.
11 L=revatt/2;
12 disp(L,'the slab length required for 30db total
reverse attenuation in cm = ')
13 disp(kc,'cut-off wave number in m-1 = ')
14 disp(betao,'propagation constant = ')

```

Scilab code Exa 10.5 design a two slab remanent phase shifter

```

1 //example:10.5 ,page no.-567.
2 // program to design a two slab remanent phase
shifter .

```



```

3 forpims=1786;er=13;f=10*10^9;uo=4*pi*10^-7;ko=(2*
  pi*f)/(3*10^10);
4 fm=2.8;s=0.1; // s and a in cm.
5 x=(2*pi*fm*forpims)/(2*pi*f); // x=wm/w = k/uo.
6 a=2.286; // for x-band.
7 t=.274; //from figure 10.19;
8 diffphaseshift=0.4*ko*(180/pi); // differential
  phase shift.
9 L_1=180/diffphaseshift;
10 L_2=90/diffphaseshift;
11 disp(L_1,'the ferrite length required for the 180
  deg. phase shift section in cm = ')
12 disp(L_2,'the ferrite length required for the 90 deg
  . phase shift section in cm = ')

```

Chapter 11

ACTIVE MICROWAVE CIRCUITS

Scilab code Exa 11.1 determine equivalent noise temperature of amplifier

```
1 // example: -11.1, page no. -589.
2 // program to determine the equivalent noise
  temperature of the amplifier.
3 T1=290; P1=-62; G=100; B=10^9; k=1.38*10^-23;
4 T2=77; P2=-64.7; Ts=450;
5 Y=P1-P2; // Y-factor in db.
6 Y=10^0.27;
7 Te=(T1-Y*T2)/(Y-1);
8 Po=G*k*B*(Ts+Te);
9 Po=10*log10(Po/0.001); /// converting in to dBm.
10 disp(Te, 'the equivalent noise temperature in kelwin
  = ')
11 disp(Po, 'the total noise power out of the amplifier
  in dBm will be = ')
```

Scilab code Exa 11.2 find the dynamic range of the amplifier

```

1 //example: -11.2, page no. -591.
2 // program to find the dynamic range of the
  amplifier.
3 G=20;F=3.5; // in db.
4 k=1.38*10^-23;To=290;B=2*10^9;
5 // output noise power => No=G*F*k*To*B.so in dbm it
  will be-
6 No=20+3.5+10*log10((k*To*B)/0.001);
7 DR=10-No;
8 disp(DR,'the dynamic range in dB = ')

```

Scilab code Exa 11.3 program to calculate the noise figure

```

1 // example: -11.3, page no. -593.
2 // program to calculate the noise figure ig antena
  is replaced by amplifier.
3 L=10^0.2;T=300;To=290;Te=150;
4 F1=1+(L-1)*(T/To);
5 Fld=10*log10(F1); // converting in to dBm.
6 Fa=1+(Te/To)
7 Fad=10*log10(Fa); // converting in to dBm.
8 Fcas=F1+L*(Fa-1);
9 Fcasd=10*log10(Fcas); // converting in to dBm.
10 disp(Fcasd,'the noise figure of the cascade in dB =
  ')
11 disp(Fad,'the noise figure of the amplifier in dB =
  ')
12 disp(Fld,'the noise figure of the line in dB = ')

```

Scilab code Exa 11.4 calculate the impedance of the diode

```

1 //example: -11.4, page no. -596.
2 //program to calculate the impedance of the diode.

```

```

3 Cp=0.1*10^-12;Lp=2*10^-9;Cj=0.15*10^-12;Rs=10;Is
  =0.1*10^(-6);
4 Io1=0;Io2=60*10^(-6);alpha=(1/25)*(10^3);
5 R1j=1/(alpha*(Io1+Is)); // for Io=0.
6 R2j=1/(alpha*(Io2+Is)); // for Io=60 mA.
7 disp(R1j,'junction resistance for Io=0, in ohm = ')
8 disp(R2j,'junction resistance for Io=0, in ohm = ')

```

Scilab code Exa 11.5 determine the stability of the transistor

```

1 //example:-11.5,page no.-617.
2 //program to determine the stability of the
  transistor by calculating k and |delta|.
3 s11=0.894*expm(%i*(-60.6)*%pi/180);
4 s21=3.122*expm(%i*(123.6)*%pi/180);
5 s12=0.02*expm(%i*(62.4)*%pi/180);
6 s22=0.781*expm(%i*(-27.6)*%pi/180);
7 delta=(s11*s22)-(s12*s21);
8 [mag_delta,theta_delta]=polar(delta);
9 k=(1+(abs(delta)^2)-(abs(s11)^2)-(abs(s22)^2))/(2*
  abs(s12*s21));
10 C1=conj(s22-delta*conj(s11))/(abs(s22)^2-abs(delta)
  ^2);
11 [mag_C1,theta_C1]=polar(C1);
12 R1=abs(s12*s21)/(abs(s22)^2-abs(delta)^2);
13 Cs=conj(s11-delta*conj(s22))/(abs(s11)^2-abs(delta)
  ^2);
14 [mag-Cs,theta-Cs]=polar(Cs);
15 Rs=abs(s12*s21)/(abs(s11)^2-abs(delta)^2);
16 disp([mag_C1,theta_C1])
17 disp([mag-Cs,theta-Cs])
18 disp(R1)
19 disp(Rs)
20 disp("NOTE:-theta is in radian")

```

Scilab code Exa 11.6 design an amplifier for maximum gain

```

1 // example:11.6 ,page no.-620.
2 // program to design an amplifier for maximum gain
   at 4 GHZ using single stub matching section.
3 s11=0.72*expm(%i*(-116)*%pi/180);
4 s22=0.73*expm(%i*(-54)*%pi/180);
5 s12=0.03*expm(%i*(57)*%pi/180);
6 s21=2.6*expm(%i*(76)*%pi/180);
7 delta=(s11*s22)-(s12*s21)
8 k=(1+(abs(delta)^2)-(abs(s11)^2)-(abs(s22)^2))/(2*
   abs(s12*s21))
9 B1=1-(abs(delta)^2)+(abs(s11)^2)-(abs(s22)^2);
10 B2=1-(abs(delta)^2)-(abs(s11)^2)+(abs(s22)^2);
11 C1=s11-delta*conj(s22);
12 C2=s22-delta*conj(s11);
13 taos=(B1-sqrt(B1^2-4*abs(C1)^2))/(2*C1);
14 [mag_taos,theta_taos]=polar(taos);
15 taol=(B2-sqrt(B2^2-4*abs(C2)^2))/(2*C2);
16 [mag_taal,theta_taal]=polar(taal);
17 Gs=1/(1-abs(taos)^2);
18 Gs=10*log10(Gs);
19 Go=abs(s21)^2;
20 Go=10*log10(Go);
21 G1=(1-abs(taal)^2)/(abs(1-s22*taol)^2);
22 G1=10*log10(G1);
23 Gtmax=Gs+Go+G1;
24 disp(Gs,'Gs = ')
25 disp(Go,'Go = ')
26 disp(G1,'G1 = ')
27 disp(Gtmax,'the over all transducer gain in dB will
   be = ')
28 Gs=1/(1-abs(taos)^2);
29 Gs=10*log10(Gs);

```

Scilab code Exa 11.7 design an amplifier to have a gain of 11 dB

```
1 // example: -11.7, page no. -625.
2 // program to design an amplifier to have a gain of
   11 dB at 4 GHz.
3 s11=0.75*expm(%i*(-120)*%pi/180);
4 s21=2.5*expm(%i*(80)*%pi/180);
5 s12=0;
6 s22=0.6*expm(%i*(-70)*%pi/180);
7 Gsmax=1/(1-abs(s11)^2);
8 Gsmax=10*log10(Gsmax);
9 Glmax=1/(1-abs(s22)^2);
10 Glmax=10*log10(Glmax);
11 Go=abs(s21)^2;
12 Go=10*log10(Go);
13 Gtmax=Gsmax+Glmax+Go;
14 disp(Gsmax, 'the maximum matching section gain in dB
   = ')
15 disp(Glmax, 'the maximum matching section gain in dB
   = ')
16 disp(Go, 'the gain of the mismatched transistor in dB
   = ')
17 disp(Gtmax, 'the maximum unilateral transducer gain
   in dB = ')
```

Scilab code Exa 11.8 calculate maximum error in Gt and design amplifier

```
1 // example: -11.8, page no. -629.
2 // program to maximum error in Gt and design an
   amplifier having a 2 dB noise figure with the
   maximum gain that is compatible with the noise
   figure.
```

```

3 s11=0.6*expm(%i*(-60)*%pi/180);
4 s21=1.9*expm(%i*(81)*%pi/180);
5 s12=0.05*expm(%i*(26)*%pi/180);
6 s22=0.5*expm(%i*(-60)*%pi/180);
7 Fmin=1.6;F=1.58;Zo=50;
8 Fmin1=10^0.16
9 tao_opt=0.62*expm(%i*(100)*%pi/180);
10 atan(imag(tao_opt)/real(tao_opt))
11 Rn=20;
12 U=abs(s12*s21*s11*s22)/((1-abs(s11)^2)*(1-abs(s22)
    ^2));
13 x=1/(1+U)^2;
14 y=1/(1-U)^2;
15 disp("x<(Gt/Gtu)<y")
16 N=((F-Fmin1)*Zo)/(4*Rn)*abs(1+tao_opt)^2
17 Cf=tao_opt/(N+1);
18 [mag_Cf,theta_Cf]=polar(Cf);
19 Rf=sqrt(N*(N+1-abs(tao_opt)^2))/(N+1);
20 disp(N,'N = ')
21 disp([mag_Cf,theta_Cf],'center of the 2 db noise
    figure circle = ')
22 disp(Rf,'the radius of the 2 dB noise figure circle
    = ')
23 G1=1/(1-abs(s22)^2);
24 G1=10*log10(G1);
25 Go=abs(s21)^2;
26 Go=10*log10(Go);
27 Gs=1.7; // all G1,Go,Gtu are in dB.
28 Gtu=Gs+Go+G1;
29 disp(Gtu,'the over all transducer gain in db will be
    = ')

```

Scilab code Exa 11.9 design a load matching network

1 // example: -11.9, page no. -635.

```

2 // program to design a load matching network for a
   50 ohm load impedance.
3 Zo=50;f=6*10^9;taoin=1.25*expm(%i*(40)*%pi/180);
4 Zin=((1+taoin)/(1-taoin))*Zo;
5 Zl=-Zin;
6 disp(Zl,'the load impedance = ')

```

Scilab code Exa 11.10 program to design a transistor oscillator

```

1 //example:11.10 ,page no.-637.
2 // program to design a transistor oscillator at 4
   GHz using a GaAs FET in common gate configuration
   .
3 clear
4 clc
5 Zo = 50.199554 - 0.0089893*%i;
6 s11=2.18*expm(%i*(-35)*%pi/180);
7 s21=2.75*expm(%i*(96)*%pi/180);
8 s12=1.26*expm(%i*(18)*%pi/180);
9 s22=0.52*expm(%i*(155)*%pi/180);// all are s
   parameter that are applicable for transistor in
   common gate configuration with a series inductor.
10 delta=s12*s21-s11*s22;
11 Ct=conj(s22-delta*conj(s11));
12 Rt=abs((s12*s21)/(abs(s22)^2-abs(delta)^2))
13 taot=0.59*expm(%i*(-104)*%pi/180);
14 taoin=s11+(s12*s21*taot)/(1-s22*taot);
15 [mag_taoi,theta_taoi]=polar(taoi)
16 Zin=((1+taoin)/(1-taoin))*Zo;
17 Zl=-((real(Zin)/3)-(%i*imag(Zin)));
18 disp([mag_taoi,theta_taoi])
19 disp(Zl,'the load impedance will be = ')

```

Scilab code Exa 11.11 obtain the greatest ratio of off to on attenuation

```
1 //example:-11.11,page no.-642.
2 // program to obtain the greatest ratio of off to
  on attenuation.
3 Cj=0.1*10^-12;Rr=1;Rf=5;Li=0.4*10^-9;f=5*10^9;Zo=50;
4 w=2*%pi*f;
5 Zr=Rr+%i*((w*Li)-(1/(w*Cj)));
6 Zf=Rf+(%i*w*Li);
7 // for series circuit.
8 ILon=-20*log10(abs((2*Zo)/(2*Zo+Zf)));
9 Iloff=-20*log10(abs((2*Zo)/(2*Zo+Zr)));
10 // for shunt circuit.
11 ILon1=-20*log10(abs((2*Zr)/(2*Zr+Zo)));
12 Iloff1=-20*log10(abs((2*Zf)/(2*Zf+Zo)));
13 disp(ILon,'for series circuit = ')
14 disp(Iloff,'for series circuit = ')
15 disp(ILon1,'for shunt circuit = ')
16 disp(Iloff1,'for shunt circuit = ')
```

Chapter 12

INTRODUCTION TO MICROWAVE SYSTEMS

Scilab code Exa 12.1 compute directivity radiation intensity and others

```
1 // example: -12.1, page no. -668.
2 // program to compute directivity, radiation intensity
  ,F, the effective area.
3 syms Etheta Hphi ko no Io l r pi theta C phi lamda;
4 Etheta=((%i*ko*no*Io*1)/(4*pi*r))*sin(theta)*exp(-%i
  *ko*r);
5 Hphi=((%i*ko*Io*1)/(4*pi*r))*sin(theta)*exp(-%i*ko*r
  );
6 F=(r^2)*(Etheta*conj(Hphi));
7 Prad=C*integ(integ(sin(theta)^3,theta),phi);
8 Prad=limit(Prad,theta,pi)-limit(Prad,theta,0);
9 Prad=limit(Prad,phi,2*pi)-limit(Prad,phi,0); // take
  cos(pi)=-1;
10 Prad=8*pi*C/3;
11 D=4*pi*C/Prad;
12 Ac=((lamda^2)*D)/(4*pi);
13 disp(F,'the radiation intensity is given by = ')
14 disp(D,'directivity is given by = ')
15 disp(Ac,'the effective area of the dipole = ')
```

Scilab code Exa 12.2 program to find the reactive power in dbm

```
1 // example: -12.2, page no. -674.
2 // program to find the reactive power in dbm.
3 Pt=120; f=6*10^9;
4 Gt=10^4.2; Gr=10^3.1;
5 lamda=0.05; R=3.59*10^7;
6 Pr=(Pt*Gt*Gr*(lamda^2))/((4*%pi*R)^2);
7 Pr=10*log10(Pr/0.001);
8 disp(Pr, 'received power in dBm will be = ')
```

Scilab code Exa 12.3 calculate the input and output SNR

```
1 // example: -12.3, page no. -677.
2 // program to calculate the input and output SNR.
3 f=4*10^9; B=1*10^6; Grf=10^2; Gif=10^3; Lt=10^0.15; Lm
   =10^0.6; To=290;
4 Fm=10^0.7; Tm=(Fm-1)*To; Tp=300; Tb=200; eta=0.9;
5 Frf=10^0.3; Fif=10^0.11; k=1.38*10^-23;
6 Trf=(Frf-1)*To;
7 Tif=(Fif-1)*To;
8 Trec=Trf+(Tm/Grf)+((Tif*Lm)/Grf);
9 Ttl=(Lt-1)*Tp;
10 Ta=eta*Tb+(1-eta)*Tp;
11 Ni=k*B*Ta;
12 Ni=10*log10(Ni/0.001); // converting in to dBm.
13 si=-80; // in dBm.
14 SNRi=si-Ni; // input SNR.
15 Tsys=Ta+Ttl+Lt*Trec;
16 SNRo=si-10*log10((k*B*Tsys)/0.001);
17 disp(SNRi, 'input SNR in dB = ')
18 disp(SNRo, 'output SNR in dB = ')
```

Scilab code Exa 12.4 program to find the maximum range of radar

```
1 // example: -12.4, page no. -683.
2 // program to find the maximum range of radar.
3 G=10^2.8; Pt=2000; sigma=12;
4 Pmin=10^-12; lamda=0.03;
5 Rmax=((Pt*(G^2)*sigma*(lamda^2))/(((4*pi)^3)*Pmin))
    ^0.25);
6 disp(Rmax, 'the maximum range of the radar in meter =
    ')
```

Scilab code Exa 12.5 program to find the J by S ratio

```
1 //example: -12.5, page no. -702.
2 // program to find the J/S ratio.
3 Gr=10^3.5; Pj=1000; R=3000; Br=1*10^6; Bj=20*10^6;
4 Gj=10; lamda=0.03; Pt=10^5; sigma=4; Rj=10000;
5 x=(Pj/Pt)*((4*pi*(R^2)*Gj)/(sigma*Gr))*(Br/Bj); //
    x=J/S
6 x=10*log10(x);
7 Grsl=10^(3.5-2); // radar antena gain in its
    sidelobe region.
8 x1=(Pj/Pt)*(((R^4)*Gj*Grsl)/((Gr^2)*(Rj^2)))*(Br/Bj)
    ;
9 x1=10*log10(x1);
10 disp(x, 'THE J/S ration for the SSJ case in dB is = '
    )
11 disp(x1, 'THE J/S ratio for the SOJ case in dB is = '
    )
```

Scilab code Exa 12.6 calculate power density of 20 m from antena

```
1 // example: -12.6, page no. -704.  
2 // program to calculate the power density of 20 m  
  from the antena.  
3 G=10^4; Pin=5; R=20;  
4 S=(Pin*G)/(4*pi*(R^2))*0.1;  
5 disp(S, 'the power density in the main beam of the  
  antena at a distance of 20 m in mw/cm^2 = ')
```

Appendix

Scilab code AP 1 equivalent of two resistances in parallel

```
1 //function example: -5.4, page no. -221.
2 function [Z]=parallel_impedance(Z1,Z2)
3     Z=(Z1*Z2)/(Z1+Z2);
4 endfunction
```

Scilab code AP 2 smith chart for finding load impedance when reflection coefficient

```
1 // function for smith chart for finding load
   impedance when reflection coefficient is given.
2 function []=smith_chart(tao)
3 theta=0:0.1:2*%pi;
4 for r=0:0.1:10
5     x=(1/(1+r))*cos(theta)+(r/(1+r));
6     y=(1/(1+r))*sin(theta);
7     plot2d(x,y,style=2,rect=[-2,-2,2,2])
8 end
9 for X=-2:0.1:2
10     if X==0
11         X=0.01;
12     end
13     x=1+(1/X)*cos(theta);
14     y=(1/X)*sin(theta)+(1/X);
15     plot2d(x,y,style=3,rect=[-2,-2,2,2])
16     xgrid(2)
17     xtitle("smith chart","Tao_Real","Tao_Imaginary")
18 end
```

```

19 x=abs(tao)*cos(theta);
20 y=abs(tao)*sin(theta);
21 plot2d(x,y,style=5,rect=[-2,-2,2,2])
22 theta=-%pi/2:0.1:%pi/2;
23 x=abs(tao)*cos(theta);
24 [r angle]=polar(tao);
25 tao=[r angle]
26 y=x*tan(tao(1,2));
27 plot2d(x,y,style=5,rect=[-2,-2,2,2])
28 endfunction

```

Scilab code AP 3 function for input impedance

```

1 // function for input impedance.
2 function [Zin]=input_impedence(tao,b,l,Zo)
3     Zin=Zo*((1+(tao*exp(-2*%i*b*l)))/(1-(tao*exp(-2*%i
         *b*l))))
4 endfunction

```

Scilab code AP 4 function for reflection coefficient

```

1 " Tao_Real", " Tao_Imaginary" " Tao_Real", " Tao_Imaginary"
   // function for reflection coefficient.
2 function [tao]=reflection_coefficient(Zl,Zo)
3     tao=(Zl-Zo)/(Zl+Zo);
4 endfunction

```

Scilab code AP 5 function to find SWR

```

1 // function to find SWR,
2 function [SWR]=VSWR(tao)
3     SWR=(1+tao)/(1-tao)
4 endfunction

```
