Scilab Textbook Companion for Engineering Electromagnetics by W. H. Hayt And J. A. Buck[1]

Created by
Prof. R. Senthilkumar
B.Tech. + M.Tech
Electronics Engineering
Institute of Road and Transport Technology
College Teacher
Na
Cross-Checked by

May 24, 2016

---

[1] Funded 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: Engineering Electromagnetics

Author: W. H. Hayt And J. A. Buck

Publisher: Tata McGraw - Hill Education, New Delhi

Edition: 7

Year: 2006

ISBN: 0070612234

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

Chapter 1

Vector Analysis

Scilab code Exa 1.1 Program to find the unit vector

```
//clear//
//Caption:Program to find the unit vector
//Example1.1
//page 8
G = [2,-2,-1]; //position of point G in cartesian coordinate system
aG = UnitVector(G);
disp(aG,'Unit Vector aG =')
//Result
//Unit Vector aG =
//      0.6666667  - 0.6666667  - 0.3333333
```

Scilab code Exa 1.2 find the phase angle between two vectors

```
//clear//
//Caption: Program to find the phase angle between two vectors
//Example1.2
```

```
//page 11
clc;
Q = [4,5,2]; //point Q
x = Q(1);
y = Q(2);
z = Q(3);
G = [y,-2.5*x,3]; //vector field
disp(G,'G(rQ) =')
aN = [2/3,1/3,-2/3]; //unit vector- direction of Q
G_dot_aN = dot(G,aN); //dot product of G and aN
disp(G_dot_aN,'G.aN =')
G_dot_aN_aN = G_dot_aN*aN;
disp(G_dot_aN_aN,'(G.aN)aN=')
teta_Ga = Phase_Angle(G,aN) //phase angle between G and unit vector aN
disp(teta_Ga,'phase angle between G and unit vector aN in degrees =')
//Result
// G(rQ) =        5.   - 10.     3.
// G.aN =       - 2.
// (G.aN)aN =     - 1.3333333  - 0.6666667  1.3333333
// phase angle between G and unit vector aN in degrees =   99.956489
```

Scilab code Exa 1.3 Rectangular coordinates into cylindrical

```
//clear//
//Caption:Transform the vector of Rectangular coordinates into cylindrical coordinates
//Example1.3
//page 18
clc;
y = sym('y');
x = sym('x');
```

```
z = sym('z');
ax = sym('ax');
ay = sym('ay');
az = sym('az');
ar = sym('ar');
aphi = sym('aphi');
phi = sym('phi');
B = y*ax-x*ay+z*az;
disp(B,'Given vector in cartesian co-ordiante system
    B=')
Br = B*ar;
Bphi = B*aphi;
Bz = B*az;
disp('Components of cylindrical vector B')
disp(Br,'Br=')
disp(Bphi,'Bphi=')
disp(Bz,'Bz=')
//Result
//Given vector in cartesian co-ordiante system B=
//  az*z+ax*y-ay*x
// Components of cylindrical vector B
// Br=
//  ar*(az*z+ax*y-ay*x)
// Bphi=
//  aphi*(az*z+ax*y-ay*x)
// Bz=
//  az*(az*z+ax*y-ay*x)
//
```

Scilab code Exa 1.4 Rectangular coordinates into spherical

```
//clear//
//Caption:Transform the vector of Rectangular
   coordinates into spherical coordinates
//Example1.4
```

```
//page 22
clc;
y = sym('y');
x = sym('x');
z = sym('z');
ax = sym('ax');
ay = sym('ay');
az = sym('az');
ar = sym('ar');
aTh = sym('aTh');
aphi = sym('aphi');
G = (x*z/y)*ax;
disp(G,'Given vector in cartesian co-ordiante system B=')
r = sym('r');
teta = sym('teta')
phi = sym('phi')
x1 = r*sin(teta)*cos(phi);
y1 = r*sin(teta)*sin(phi);
z1 = r*cos(teta);
G1 = (x1*z1/y1)*ax;
Gr = G1*ar;
GTh = G1*aTh;
Gphi = G1*aphi;
Gsph = [Gr,GTh,Gphi];
disp(Gr,'Gr=')
disp(GTh,'GTh=')
disp(Gphi,'Gphi=')
//Result
//Given vector in cartesian co-ordiante system B = ax*x*z/y
//Gr =    ar*ax*cos(phi)*r*cos(teta)/sin(phi)
//GTh =    ax*cos(phi)*r*cos(teta)*aTh/sin(phi)
//Gphi =    aphi*ax*cos(phi)*r*cos(teta)/sin(phi)
//
```

Chapter 2

Columbs Law and Electric Field Intensity

Scilab code Exa 2.1 Caculate force exerted on Q2 by Q1

```
//clear//
//Caption:Program to Caculate force exerted on Q2 by
    Q1
//Example2.1
//page 29
clc;
r2 = [2,0,5];
r1 = [1,2,3];
R12 = norm(r2-r1);
aR12 = UnitVector(r2-r1);
disp(R12,'R12=')
disp(aR12,'aR12=')
Q1 = 3e-04; //charge 1 in Coulombs
Q2 = -1e-04; //charge 2 in Coulombs
Eps = 8.854e-12; //free space permittivity
F2 = ((Q1*Q2)/(4*%pi*Eps*R12^2))*aR12;
F1 = -F2;
disp(F2,'Force exerted on Q2 by Q1 in N/m F2 =')
disp(F1,'Force exerted on Q1 by Q2 in N/m F1 =')
```

```
//Result
//R12=
//    3.
//aR12=
//     0.3333333  - 0.6666667     0.6666667
//Force exerted on Q2 by Q1 in N/m F2 =
//  - 9.9863805     19.972761  - 19.972761
//Force exerted on Q1 by Q2 in N/m F1 =
//      9.9863805  - 19.972761     19.972761
```

Scilab code Exa 2.2 Caculate Electric Field

```
//clear//
//Caption:Program to Caculate Electric Field E at P due to 4 identical charges
//Example2.2
//page 33
clc;
P = [1,1,1];
P1 = [1,1,0];
P2 = [-1,1,0];
P3 = [-1,-1,0];
P4 = [1,-1,0];
R1 = norm(P-P1);
aR1 = UnitVector(P-P1);
R2  = norm(P-P2);
aR2 = UnitVector(P-P2);
R3  = norm(P-P3);
aR3 = UnitVector(P-P3);
R4  = norm(P-P4);
aR4 = UnitVector(P-P4);
disp(R1,'R1=')
disp(aR1,'aR1=')
disp(R2,'R2=')
disp(aR2,'aR2=')
```

```
disp(R3,'R3=')
disp(aR3,'aR3=')
disp(R4,'R4=')
disp(aR4,'aR4=')
Q = 3e-09; //charge  in Coulombs
Eps = 8.854e-12; //free space permittivity
E1 = (Q/(4*%pi*Eps*R1^2))*aR1;
E2 = (Q/(4*%pi*Eps*R2^2))*aR2;
E3 = (Q/(4*%pi*Eps*R3^2))*aR3;
E4 = (Q/(4*%pi*Eps*R4^2))*aR4;
E = E1+E2+E3+E4;
disp(E,'Electric Field Intesnity at any point P due to four identical Charges in V/m=')
//Result
//R1=       1.
//aR1=      0.     0.     1.
//R2=       2.236068
//aR2=      0.8944272    0.     0.4472136
//R3=       3.
//aR3=      0.6666667    0.6666667    0.3333333
//R4=       2.236068
//aR4=      0.     0.8944272    0.4472136
//Electric Field Intesnity at any point P due to four identical Charges in V/m=
//   6.8206048    6.8206048    32.785194
//
```

Scilab code Exa 2.3 Total Charge Enclosed

```
//clear//
//Example2.3
//page 35
clc;
r = sym('r');
z = sym('z');
```

```
phi = sym('phi');
rv = -5e-06*exp(-1e05*r*z);
disp(rv,'Volume Charge density in C/cubic.metre rv=')
Q1 = integ(rv*r,phi);
Q1 = limit(Q1,phi,2*%pi);
Q2 = integ(Q1,z);
Q2 = limit(Q2,z,0.04)-limit(Q2,z,0.02);
Q3 = integ(Q2,r);
Q3 = limit(Q3,r,0.01)-limit(Q3,r,0);
disp(Q1,'Q1=')
disp(Q2,'Q2=')
disp(Q3,'Total Charge Enclosed in a 2cm length  of electron beam in coulombs Q=')
//Result
//Volume Charge density in C/cubic.metre rv =  -%e^-(100000*r*z)/200000
//Q1=     -103993*r*%e^-(100000*r*z)/3310200000
//Q2=     -103993*%e^-(2000*r)/331020000000000
//Total Charge Enclosed in a 2cm length  of electron beam in coulombs Q=
// 103993/13240800000000000000-103993*%e^-40/13240800000000000000
//Q approximately equal to 103993/13240800000000000000 = 7.854D-14 coulombs
```

Chapter 3

Electric Flux Density Gausss Law and Divergence

Scilab code Exa 3.1 find Electric Flux density 'D

```
//clear//
//Caption: Program to find Electric Flux density 'D' of a uniform line charge
//Example3.1
//page 54
clc;
e0 = 8.854e-12; //free space permittivity in F/m
rL = 8e-09; //line charge density c/m
r = 3; // distance in metre
E = Electric_Field_Line_Charge(rL,e0,r); //electric field intensity of line charge
D = e0*E;
disp(D,'Electric Flux Density in Coulombs per square metre D =')
//Result
// Electric Flux Density in Coulombs per square metre D =
//      4.244D-10
```

Scilab code Exa 3.2 calculate surface charge density

```
//clear//
//Caption: Program to calculate surface charge density,Flux density, Field Intensity of coaxial cable
//Example3.2
//page 64
clc;
Q_innercyl = 30e-09; //total charge on the inner conductor in coulombs
a = 1e-03; // inner radius of coaxial cable in metre
b = 4e-03; // outer radius of coaxial cable in metre
L = 50e-02; //length of coaxial cable
rs_innercyl = Q_innercyl/(2*%pi*a*L);
rs_outercyl = Q_innercyl/(2*%pi*b*L);
e0 = 8.854e-12; //free space relative permittivity F/m
r = sym('r');
Dr = a*rs_innercyl/r;
Er = Dr/e0;
disp(rs_innercyl,'Surface charge density of inner cylinder of coaxial cable in C/square.metre, rs_innercyl=')
disp(rs_outercyl,'Surface charge density of outer cylinder of coaxial cable in C/square.metre, rs_outercyl=')
disp(Dr,'Electric Flux Density in C/square.metre Dr=')
disp(Er,'Electric Field Intensity in V/m Er=')
//Result
//Surface charge density of inner cylinder of coaxial cable in C/square.metre, rs_innercyl=
//     0.0000095
```

```
//Surface charge density of outer cylinder of coaxial cable in C/square.metre, rs_outercyl=
//     0.0000024
//Electric Flux Density in C/square.metre Dr=
// 9.5488183337312011E-9/r
//Electric Field Intensity in V/m Er=
// 1078.47507722286/r
```

Scilab code Exa 3.3 total charge enclosed in a volume

```
//clear//
//Caption: Program to calculate the total charge enclosed in a volume at the origin
//Example3.3
//page 67
clc;
V = 1e-09; //volume in cubic metre
x = sym('x');
y = sym('y');
z = sym('z');
//Components of Electric Flux Density in cartesian coordinate system
Dx = exp(-x)*sin(y);
Dy = -exp(-x)*cos(y);
Dz = 2*z;
//Divergence of electric flux density 'D'
dDx = diff(Dx,x);
dDy = diff(Dy,y);
dDz = diff(Dz,z);
//Total charge enclosed in a given volume
del_Q = (dDx+dDy+dDz)*V;
disp(del_Q,'Total charge enclosed in an incremental volume in coulombs, del_Q =')
//Total Charge enclosed in a given volume at origin (0,0,0)
```

```
del_Q = limit(del_Q,x,0);
del_Q = limit(del_Q,y,0);
del_Q = limit(del_Q,z,0);
disp(del_Q*1e09,'Total charge enclosed in an incremental volume in  nano coulombs at origin, del_Q =')
//Result
//Total charge enclosed in an incremental volume in coulombs, del_Q =     2.0000000000000001E-9
//Total charge enclosed in an incremental volume in  nano coulombs at origin, del_Q =
//   2.0
```

Scilab code Exa 3.4 Find the Divergence

```
//clear//
//Caption: Program to Find the Divergence of 'D' at the origin
//Example3.4
//page 70
clc;
x = sym('x');
y = sym('y');
z = sym('z');
//Components of Electric Flux Density in cartesian coordinate system
Dx = exp(-x)*sin(y);
Dy = -exp(-x)*cos(y);
Dz = 2*z;
//Divergence of electric flux density 'D'
dDx = diff(Dx,x);
dDy = diff(Dy,y);
dDz = diff(Dz,z);
divD = dDx+dDy+dDz
disp(divD,'Divergence of Electric Flux Density D in
```

```
C/cubic.metre, divD =')
divD = limit(divD,x,0);
divD = limit(divD,y,0);
divD = limit(divD,z,0);
disp(divD,'Divergence of Electric Flux Density D in C/cubic.metre at origin, divD =')
//Result
//Divergence of Electric Flux Density D in C/cubic.metre, divD =
// 2
//Divergence of Electric Flux Density D in C/cubic.metre at origin, divD =
// 2
```

Scilab code Exa 3.5 verify the Divergence theorem

```
//clear//
//Caption: Program to verify the Divergence theorem for the field 'D'
//Example3.5
//page 74
clc;
x = sym('x');
y = sym('y');
z = sym('z');
//Components of Electric Flux Density in cartesian coordinate system
Dx = 2*x*y;
Dy = x^2;
Dz = 0;
//Divergence of electric flux density 'D'
dDx = diff(Dx,x);
dDy = diff(Dy,y);
dDz =0;
divD = dDx+dDy+dDz
```

```
disp(divD,'Divergence of Electric Flux Density D in C/cubic.metre, divD =')
//Evaluate volume integral on divergence of 'D'
Vol_int_divD = integ(divD,x);
Vol_int_divD = limit(Vol_int_divD,x,1)-limit(Vol_int_divD,x,0);
Vol_int_divD = integ(Vol_int_divD,y);
Vol_int_divD = limit(Vol_int_divD,y,2)-limit(Vol_int_divD,y,0);
Vol_int_divD = integ(Vol_int_divD,z);
Vol_int_divD = limit(Vol_int_divD,z,3)-limit(Vol_int_divD,z,0);
disp(Vol_int_divD,'Volume Integral of divergence of D, in coulombs vol_int(divD)=')
//Evaluate surface integral on field D
Dx = limit(Dx,x,1);
sur_D = integ(Dx,y);
sur_D = limit(sur_D,y,2) - limit(sur_D,y,0);
sur_D = integ(sur_D,z);
sur_D = limit(sur_D,z,3) - limit(sur_D,z,0);
disp(sur_D,'Surface Integral of field D, in coulombs sur_int(D.ds)=')
if(sur_D==Vol_int_divD)
  disp('Divergence Theorem verified')
end
//Result
// Divergence of Electric Flux Density D in C/cubic.metre, divD =
//  2*y
//Volume Integral of divergence of D, in coulombs vol_int(divD)=
// 12
// Surface Integral of field D, in coulombs sur_int(D.ds)=
// 12
```

Chapter 4

Energy and Potential

Scilab code Exa 4.1 find the work involved

```
//clear//
//Caption: Program to find the work involved 'W' in
   moving a charge 'Q' along shorter arc of a circle
//Example4.1
//page 84
clc;
x = sym('x');
y = sym('y');
z = sym('z');
y1 = sym('y1');
y = sqrt(1-x^2);
Q = 2; //charge in coulombs
Edot_dL1 = integ(y,x);
disp(Edot_dL1,'E.dx*ax =')
Edot_dL1 = limit(Edot_dL1,x,0.8)-limit(Edot_dL1,x,1)
   ;
disp(Edot_dL1,'Value of E.dx*ax =')
Edot_dL2 = 0;
disp(Edot_dL2,'Value of E.dz*az=')
x = sqrt(1-y1^2);
Edot_dL3 = integ(x,y1)
```

```
disp(Edot_dL3,'E.dy*ay=')
Edot_dL3 = limit(Edot_dL3,y1,0.6)-limit(Edot_dL3,y1,0);
disp(Edot_dL3,'Value of E.dy*ay =')
W = -Q*(Edot_dL1+Edot_dL2+Edot_dL3);
disp(W,'Work done in moving a point charge along shorter arc of circle in Joules, W=')
//Result
// E.dx*ax =     asin(x)/2+x*sqrt(1-x^2)/2
// Value of E.dx*ax =     (25*asin(4/5)+12)/50-%pi/4
// Value of E.dz*az =        0.
// E.dy*ay =     asin(y1)/2+y1*sqrt(1-y1^2)/2
// Value of E.dy*ay =     (25*asin(3/5)+12)/50
//Work done in moving a point charge along shorter arc of circle in Joules, W =
// -2*((25*asin(4/5)+12)/50+(25*asin(3/5)+12)/50-%pi/4)
//Which is equivalent to
// -2*((25*0.9272952+12)/50+(25*0.6435011+12)/50-%pi/4) = -0.96 Joules
```

Scilab code Exa 4.2 find the work involved 'W'

```
//clear//
//Caption: Program to find the work involved 'W' in moving a charge 'Q' along straight line
//Example4.2
//page 84
clc;
x = sym('x');
y = sym('y');
z = sym('z');
y1 = sym('y1');
y = -3*(x-1);
Q = 2; //charge in coulombs
```

```
Edot_dL1 = integ(y,x);
disp(Edot_dL1,'E.dx*ax =')
Edot_dL1 = limit(Edot_dL1,x,0.8)-limit(Edot_dL1,x,1);
disp(Edot_dL1,'Value of E.dx*ax =')
Edot_dL2 = 0;
disp(Edot_dL2,'Value of E.dz*az=')
x = (1-y1/3);
Edot_dL3 = integ(x,y1)
disp(Edot_dL3,'E.dy*ay=')
Edot_dL3 = limit(Edot_dL3,y1,0.6)-limit(Edot_dL3,y1,0);
disp(Edot_dL3,'Value of E.dy*ay =')
W = -Q*(Edot_dL1+Edot_dL2+Edot_dL3);
disp(W,'Work done in moving a point charge along shorter arc of circle in Joules, W=')
//Result
//E.dx*ax = -3*(x^2/2-x)
//Value of E.dx*ax = -3/50
//Value of E.dz*az =   0.
//E.dy*ay =  y1-y1^2/6
//Value of E.dy*ay =   27/50
//Work done in moving a point charge along shorter arc of circle in Joules, W = -24/25  = -0.96 Joules
```

Scilab code Exa 4.3 Program to calculate E, D and volume charge

```
//clear//
//Caption: Program to calculate E, D and volume charge density using divergence of D
//Example4.3
//page 100
clc;
x = -4;
```

```
y = 3;
z = 6;
V = 2*(x^2)*y-5*z;
disp(float(V),'Potential V at point P(-4,3,6)in volts is Vp =')
x1 = sym('x1');
y1 = sym('y1');
z1 = sym('z1');
ax = sym('ax');
ay = sym('ay');
az = sym('az');
V1 = 2*(x1^2)*y1-5*z1;
//Electric Field Intensity from gradient of V
Ex = -diff(V1,x1);
Ey = - diff(V1,y1);
Ez = - diff(V1,z1);
Ex1 = limit(Ex,x1,-4);
Ex1 = limit(Ex1,y1,3);
Ex1 = limit(Ex1,z1,6);
Ey1 = limit(Ey,x1,-4);
Ey1 = limit(Ey1,y1,3);
Ey1 = limit(Ey1,z1,6);
Ez1 = limit(Ez,x1,-4);
Ez1 = limit(Ez1,y1,3);
Ez1 = limit(Ez1,z1,6);
E = Ex1*ax+Ey1*ay+Ez1*az;
Ep = sqrt(float(Ex1^2+Ey1^2+Ez1^2));
disp(Ep,'Electric Field Intensity E at point P(-4,3,6) in volts E =')
aEp = float(E/Ep);
disp(aEp,'Direction of Electric Field E at point P(-4,3,6) aEp=')
Dx = float(8.854*Ex);
Dy = float(8.854*Ey);
Dz = float(8.854*Ez);
D = Dx*ax+Dy*ay+Dz*az;
disp(D,'Electric Flux Density in pico.C/square.metre D =')
```

```
dDx = diff(Dx,x1);
dDx = limit(dDx,x1,-4);
dDx = limit(dDx,y1,3);
dDx = limit(dDx,z1,6);
dDy = diff(Dy,y1);
dDy = limit(dDy,x1,-4);
dDy = limit(dDy,y1,3);
dDy = limit(dDy,z1,6);
dDz = diff(Dz,z1);
dDz = limit(dDz,x1,-4);
dDz = limit(dDz,y1,3);
dDz = limit(dDz,z1,6);
rV = dDx+dDy+dDz;
disp(rV,'Volume Charge density from divergence of D in pC/cubic.metre is rV=')
//Result
//Potential V at point P(-4,3,6)in volts is Vp = 66.
//Electric Field Intensity E at point P(-4,3,6) in volts E = 57.9050947672137
//Direction of Electric Field E at point P(-4,3,6) aEp=
//0.01726963756851*(5*az-32*ay+48*ax)
//equivalent to aEp= 0.0863482*az-0.5526284*ay+0.8289426*ax
//Electric Flux Density in pico.C/square.metre D =
//  -35.416*ax*x1*y1-17.708*ay*x1^2+44.27*az
//Volume Charge density from divergence of D in pC/cubic.metre is rV=
//  -106.248
```

Chapter 5

Current and Conductors

Scilab code Exa 5.1 find the resistance, current and current density

```
//clear//
//Caption: Program to find the resistance, current and current density
//Example5.1
//page 123
clc;
clear;
D = 0.0508; //diameter of conductor in inches
D = 0.0508*0.0254; //diameter in metres
r = D/2; //radius in metres
A = %pi*r^2; //area of the conductor in square metre
L = 1609; //length of the copper wire in metre
sigma = 5.80e07; //conductivity in siemens/metre
R = L/(sigma*A); //resistance in ohms
I = 10; //current in amperes
J = I/A; //current density in amps/square.metre
disp(R,'Rresistance in ohms of given copper wire R = ')
disp(J,'Current density in A/square.metre J = ')
//Result
//Rresistance in ohms of given copper wire R =
```

```
//      21.215013
//Current density in A/square.metre J =
//      7647425.6
```

Scilab code Exa 5.2 find potential at point P

```
//clear//
//Caption: Program to find potential at point P, Electricf Field Intensity E, Flux density D
//Example5.2
//page 126
clc;
x = sym('x');
y = sym('y');
z = sym('z');
ax = sym('ax');
ay = sym('ay');
az = sym('az');
V = 100*(x^2-y^2);
disp(V,'Potential in Volts V =')
Ex = diff(V,x);
Ey = diff(V,y);
Ez = diff(V,z);
E = -(Ex*ax+Ey*ay+Ez*az);
disp(E,'Electric Field Intensity in V/m E =')
E = limit(E,x,2);
E = limit(E,y,-1);
V = limit(V,x,2);
V = limit(V,y,-1);
disp(V,'Potential at point P in Volts Vp =')
disp(E,'Electric Field Intensity at point P in V/m Ep =')
D = 8.854e-12*E;
disp(D*1e09,'Electric FLux Density at point P in nC/square.metre Dp =')
```

```
//Result
//Potential in Volts V =  100*(x^2-y^2)
//Electric Field Intensity in V/m E =  200*ay*y-200*ax*x
//Potential at point P in Volts Vp =   300
//Electric Field Intensity at point P in V/m Ep = -200*ay-400*ax
//Electric FLux Density at point P in nC/square.metre Dp = 0.008854*(-200*ay-400*ax)
//which is equivalent to Dp = -3.5416*ax -1.7708*ay
```

Scilab code Exa 5.3 equation of the streamline

```
//clear//
//Caption: Program to determine the equation of the streamline passing through any point P
//Example5.3
//page 128
clc;
x = sym('x');
y = sym('y');
z = sym('z');
C1 = integ(1/y,y)+integ(1/x,x);
disp(C1,'C1 = ')
C2 = exp(C1);
disp(C2,'The Stream line Equation C2 = ')
C2 = limit(C2,x,2);
C2 = limit(C2,y,-1);
disp(C2,'The value of constant in the streamline equation passing through the point P is C2=')
//Result
//C1 = log(y)+log(x)
//The Stream line Equation C2 = x*y
//The value of constant in the streamline equation passing through the point P is C2 = -2
```

Chapter 6

Dielectrics and Capacitance

Scilab code Exa 6.1 calculate D,E and Polarization P

```
//clear//
//Caption: Program to calculate D,E and Polarization
    P for Teflon slab
//Example6.1
//page 142
clc;
ax = sym('ax');
e0 = sym('e0');
E0 = sym('E0');
Ein = sym('Ein');
er = 2.1; //relative permittivity of teflon
chi = er-1; //electric susceptibility
Eout = E0*ax;
Dout = float(e0*Eout);
Din = float(er*e0*Ein);
Pin = float(chi*e0*Ein);
disp(Dout,'Dout in c/square.metre = ')
disp(Din,'Din in c/square.metre = ')
disp(Pin,'Polarization in coulombs per square metre
   Pin =')
//Result
```

```
//Dout in c/square.metre =   ax*e0*E0
//Din in c/square.metre =    2.1*e0*Ein
//Polarization in coulombs per square metre Pin = 1.1*e0*Ein
```

Scilab code Exa 6.2 Program to calculate E and Polarization P

```
//clear//
//Caption: Program to calculate E and Polarization P for Teflon slab
//Example6.2
//page 146
clc;
ax = sym('ax');
e0 = sym('e0');
E0 = sym('E0');
er = 2.1; //relative permittivity of teflon
chi = er-1; //electric susceptibility
Eout = E0*ax;
Ein = float(Eout/er);
Din = float(e0*Eout);
Pin = float(Din - e0*Ein);
disp(Ein,'Ein in V/m = ')
disp(Pin,'Polarization in coulombs per square metre Pin =')
//Result
//Ein in V/m =  0.47619047619048*ax*E0
//Polarization in coulombs per square metre Pin = 0.52380952380952*ax*e0*E0
```

Scilab code Exa 6.3 Program to calculate the capacitance

```
//clear//
```

```
//Caption: Program to calculate the capacitance of a
   parallel plate capacitor
//Example6.3
//page 151
clc;
S = 10;//area in square inch
S = 10*(0.0254)^2; //area in square metre
d = 0.01; //distance between the plates in inch
d = 0.01*0.0254; //distance between the plates in
   metre
e0 = 8.854e-12; //free space permittivity in F/m
er = 6; //relative permittivity of mica
e = e0*er;
C =  parallel_capacitor(e,S,d);
disp(C*1e09,'Capacitance of a parallel plate
   capacitor in pico farads C =')
//Result
//Capacitance of a parallel plate capacitor in pico
   farads C = 1.3493496
```

Chapter 7

Poissons and Laplaces Equation

Scilab code Exa 7.1 Derivation of capacitance

```
//clear//
//Caption: Derivation of capacitance of a parallel
    plate capacitor
//Example7.1
//page 177
clc;
x = sym('x');
d = sym('d');
Vo = sym('Vo');
e = sym('e');
ax = sym('ax');
A = sym('A');
B = sym('B');
S = sym('S');
V = integ(A,x)+B;
V = limit(V,A,Vo/d);
V = limit(V,B,0);
disp(V,'Potential in Volts V =')
E = -diff(V,x)*ax;
disp(E,'Electric Field in V/m E =')
D = e*E;
```

```
DN = D/ax;
disp(D,'Electric Flux Density in C/square metre D ='
    )
Q = -DN*S;
disp(Q,'Charge in Coulombs Q =')
C = Q/Vo;
disp(C,'Capacitance of parallel plate capacitor C ='
    )
//Result
//Potential in Volts V =    Vo*x/d
//Electric Field in V/m E =    -ax*Vo/d
//Electric Flux Density in C/square metre D =  -ax*e
    *Vo/d
//Charge in Coulombs Q =    e*Vo*S/d
//Capacitance of parallel plate capacitor C =  e*S/d
```

Scilab code Exa 7.2 Capacitance of a Cylindrical Capacitor

```
//clear//
//Caption: Capacitance of a Cylindrical Capacitor
//Example7.2
//page 179
clc;
A = sym('A');
B = sym('B');
r = sym('r');
ar = sym('ar');
ruo = sym('ruo');
a = sym('a');
b = sym('b');
L = sym('L');
Vo = sym('Vo');
V = integ(A/r,r)+B;
disp(V,'Potential V = ')
V = limit(V,A,Vo/log(a/b));
```

```
V = limit(V,B,-Vo*log(b)/log(a/b));
disp(V,'Potential V by substitute the values of constant A & B = ')
V = Vo*log(b/r)/log(b/a);
E = -diff(V,r)*ar;
disp(E,'E = ');
E = limit(E,r,a);
disp(E,'E at r =a is =')
D = e*E;
DN = D/ar;
disp(DN,'DN =')
S = float(2*%pi*a*L); //area of cylinder
Q = DN*S
disp(Q,'Q =')
C = Q/Vo;
disp(C,'Capacitance of a cylindrical Capacitor C =')
//Result
// Potential V = B+log(r)*A
// Potential V by substitute the values of constant A & B =(log(r)-log(b))*Vo/log(a/b)
// E = ar*Vo/(log(b/a)*r)
// E at r =a is = ar*Vo/(a*log(b/a))
// DN = e*Vo/(a*log(b/a))
// Q = 6.283185306023805*e*Vo*L/log(b/a)
// Capacitance of a cylindrical Capacitor C = 6.283185306023805*e*L/log(b/a)
```

Scilab code Exa 7.3 Determine the electric field

```
//clear//
//Caption: Program to Determine the electric field of a two infinite radial planes with an interior angle alpha
//Example 7.3
//page 180
```

```
clc;
phi = sym('phi');
A = sym('A');
B = sym('B');
Vo = sym('Vo');
alpha = sym('alpha');
aphi = sym('aphi');
r = sym('r');
V = integ(A,phi)+B;
disp(V,'V =');
V = limit(V,B,0);
V = limit(V,A,Vo/alpha);
disp(V,'Potential V after applying boundary conditions =')
E = -(1/r)*diff(V,phi)*aphi;
disp(E,'E =')
//Result
// V =  B+phi*A
// Potential V after applying boundary conditions = phi*Vo/alpha
// E =    -aphi*Vo/(alpha*r)
```

Scilab code Exa 7.4 capacitance of a spherical capacito

```
//clear//
//Caption: Derivation of capacitance of a spherical capacitor
//Example7.4
//page 181
clc;
a = sym('a');
b = sym('b');
Vo = sym('Vo');
r = sym('r');
e = sym('e');
```

```
V = Vo*((1/r)-(1/b))/((1/a)-(1/b));
disp(V,'V =')
E = -diff(V,r)*ar;
disp(E,'E =')
D = e*E;
DN = D/ar;
disp(DN,'DN =')
S = float(4*%pi*r^2); //area of sphere
Q = DN*S;
disp(Q,'Q =')
C = Q/Vo;
disp(C,'Capacitance of a spherical capacitor =')
//Result
//V =  (1/r-1/b)*Vo/(1/a-1/b)
//E =     ar*Vo/((1/a-1/b)*r^2)
//DN =    e*Vo/((1/a-1/b)*r^2)
//Q =     12.56637060469643*e*Vo/(1/a-1/b)
//Capacitance of a spherical capacitor =
   12.56637060469643*e/(1/a-1/b)
```

Scilab code Exa 7.5 Potential in spherical coordinates

```
//clear//
//Caption: Potential in spherical coordinates as a
   function of teta V(teta)
//Example7.5
//page 182
clc;
teta = sym('teta');
A = sym('A');
B = sym('B');
V = integ(A/float(sin(teta)),teta)+B;
disp(V,'V = ')
//Result
//V =  B+(log(cos(teta)-1)/2-log(cos(teta)+1)/2)*A
```

```
//Equivalent to V = B+log(tan(teta/2))*A
```

Chapter 8

The Steady Magnetic Field

Scilab code Exa 8.1 find the magnetic field intensity

```
//clear//
//Caption: Program to find the magnetic field
    intensity of a current carrying filament
//Example8.1
//page 217
clc;
I = 8; //current in amps
alpha1x = -90/57.3; // phase angle along with x-axis
x = 0.4;
y = 0.3;
z =0;
alpha2x = atan(x/y);
aphi = sym('aphi');
az = sym('az');
rx = y; // distance in metres in cynlindrical
    coordiante system
H2x = float((I/(4*%pi*rx))*(sin(alpha2x)-sin(alpha1x
    )))*-az;
disp(H2x,'H2x = ')
alpha1y = -atan(y/x);
alpha2y = 90/57.3;
```

```
ry = 0.4;
H2y = float((I/(4*%pi*ry))*(sin(alpha2y)-sin(alpha1y)))*-az;
disp(H2y,'H2y =')
H2 = H2x+H2y;
disp(H2,'H2 =')
//Result
//H2x =   -3.819718617079289*az
//H2y =    -2.546479080730701*az
//H2 =      -6.366197697809999*az
```

Scilab code Exa 8.2 to find the curl H

```
//clear//
//Caption: Program to find the curlH of a square path of side 'd'
//Example8.2
//page 230
clc;
ax = sym('ax');
az = sym('az');
ay = sym('ay');
z = sym('z');
y  = sym('y');
d = sym('d');
H = 0.2*z^2*ax;
Hx = float(H/ax);
HdL = float(0.4*z*d^2);
//curlH evaluated from the definition of curl
curlH = (HdL/(d^2))*ay;
//curlH evaluated from the determinant
del_cross_H = -ay*(-diff(Hx,z))+az*(-diff(Hx,y));
disp(curlH,'curlH = ')
disp(del_cross_H,'del_cross_H = ')
//Result
```

```
//curlH =  0.4*ay*z
//del_cross_H = 0.4*ay*z
```

Scilab code Exa 8.3 verify Stokes theorem

```
//clear//
//Caption: Program to verify Stokes theorem
//Example8.3
//page 233
clc;
teta = sym('teta');
phi = sym('phi');
ar = sym('ar');
aphi = sym('aphi');
az = sym('az');
r = sym('r');
curlH = float(36*cos(teta)*cos(phi)*r^2*sin(teta));
curlH_S = integ(curlH,teta);
curlH_S = float(limit(curlH_S,r,4));
curlH_S = float(limit(curlH_S,teta,0.1*%pi))-float(limit(curlH_S,teta,0));
curlH_S = integ(curlH_S,phi);
curlH_S = float(limit(curlH_S,phi,0.3*%pi))-float(limit(curlH_S,phi,0));
disp(curlH_S,'Surface Integral of curlH in Amps =')
Hr = 6*r*sin(phi);
Hphi = 18*r*sin(teta)*cos(phi);
HdL = float(limit(Hphi*r*sin(teta),r,4));
HdL = float(limit(HdL,teta,0.1*%pi));
HdL = float(integ(HdL,phi))
HdL = float(limit(HdL,phi,0.3*%pi));
disp(HdL,'Closed Line Integral of H in Amps =')
//Result
//Surface Integral of curlH in Amps = 22.24922359441324
```

```
// Closed Line Integral of H in Amps =
   22.24922359441324
```

Chapter 9

Magnetic Forces Materials and Inductance

Scilab code Exa 9.1 find magnetic field and force produced

```
//clear//
//Caption: Program to find magnetic field and force
   produced in a square loop
//Example9.1
//page 263
clc;
x = sym('x');
y = sym('y');
z = sym('z');
ax = sym('ax');
ay = sym('ay');
az = sym('az');
I = 15; //filament current in amps
I1 = 2e-03; //current in square loop
u0 = 4*%pi*1e-07; //free space permeability in H/m
H = float(I/(2*%pi*x))*az;
disp(H,'Magnetic Field Intensity in A/m  H =')
B = float(u0*H);
disp(B,'Magnetic Flux Density in Tesla B = ')
```

```
Bz = B/az;
//Bcross_dL  = ay*diff(Bz,x);
F1 = float(-I1*integ(ay*Bz,x));
F1 = float(limit(F1,x,3)-limit(F1,x,1));
F2 = float(-I1*integ(ax*-Bz,y));
F2 = float(limit(F2,x,3));
F2 = float(limit(F2,y,2)-limit(F2,y,0));
F3 = float(-I1*integ(ay*Bz,x));
F3 = float(limit(F3,x,1)-limit(F3,x,3));
F4 = float(-I1*integ(ax*-Bz,y));
F4 = float(limit(F4,x,1));
F4 = float(limit(F4,y,0)-limit(F4,y,2));
F =float((F1+F2+F3+F4)*1e09);
disp(F,'Total Force acting on a square loop in nN F = ')
//Result
//Magnetic Field Intensity in A/m H = 2.387324146817574*az/x
//Magnetic Flux Density in Tesla B = 3.0000000003340771E-6*az/x
//Total Force acting on a square loop in nN F = -8.000000000890873*ax
```

Scilab code Exa 9.2 determine the differential force

```
//clear//
//Caption: Program to determine the differential force between two differential current elements
//Example9.2
//page 265
clc;
ax = sym('ax');
ay = sym('ay');
az = sym('az');
//position of filament in cartesian coordinate
```

```
   system
P1 = [5,2,1];
P2 = [1,8,5];
//distance between filament 1 and filament 2
R12 = norm(P2-P1);
disp(R12,'R12 =')
I1dL1 = [0,-3,0]; //current carrying first filament
   1
I2dL2 = [0,0,-4]; //current carrying second filament
    2
u0 = 4*%pi*1e-07; //free space permeability in H/m
aR12  = UnitVector(P2-P1); //unit vector
disp(aR12,'aR12 =')
C1 = cross_product(I1dL1,aR12);
C2 = cross_product(I2dL2,C1);
dF2 = (u0/(4*%pi*R12^2))*C2;
dF2_y = float(dF2(2)*1e09);
disp(dF2_y*ay,'the differential force between two
   differential current elements in nN =')
//Result
//R12 = 8.2462113
//aR12 =  - 0.4850713    0.7276069    0.4850713
//the differential force between two differential
   current elements in nN = 8.560080878105142*ay
```

Scilab code Exa 9.3 calculate the total torque acting

```
//clear//
//Caption: Program to calculate the total torque
   acting on a planar rectangular current loop
//Example9.3
//page 271
clc;
ax = sym('ax');
ay = sym('ay');
```

```
az = sym('az');
x = 1;//length in metre
y = 2; //wide in metre
S = [0,0,x*y]; //area of current loop in square
    metre
I = 4e-03; //current in Amps
B = [0,-0.6,0.8];
T = I*cross_product(S,B);
Tx = float(T(1));
disp(Tx*ax*1e03,'Total Torque acting on the
    rectangular current loop in milli N/m=')
//Result
//Total Torque acting on the rectangular current
    loop in milli N/m = 4.8*ax
```

Scilab code Exa 9.4 find the torque and force acting

```
//clear//
//Caption: Program to find the torque and force
    acting on each side of planar loop
//Example9.4
//page 271
clc;
ax = sym('ax');
ay = sym('ay');
az = sym('az');
I = 4e-03; //current in Amps
B = [0,-0.6,0.8]; //Magentic Field acting on current
     loop in Tesla
L1 = [1,0,0];  //length along x-axis
L2 = [0,2,0]; //length along y-axis
F1 = I*cross_product(L1,B);
F3 = -F1;
F2 = I*cross_product(L2,B);
F4 = -F2;
```

```
R1 = [0,-1,0];  //distance from center of loop for
   side1
R2 = [0.5,0,0]; //distance from center of loop for
   side2
R3 = [0,1,0]; //distance from center of loop for
   side3
R4 = [-0.5,0,0];//distance from center of loop for
   side4
T1 = cross_product(R1,F1);
T2 = cross_product(R2,F2);
T3 = cross_product(R3,F3);
T4 = cross_product(R4,F4);
T = T1+T2+T3+T4;
Tx = float(T(1)*1e03);
disp(F1,'F1 =')
disp(F2,'F2 =')
disp(F3,'F3 =')
disp(F4,'F4 =')
disp(T1,'T1 =')
disp(T2,'T2 =')
disp(T3,'T3 =')
disp(T4,'T4 =')
disp(Tx*ax,'Total torque acting on the rectangular
   planar loop in milli N/m T =')
//Result
// F1 =
//      0.
//   - 0.0032
//   - 0.0024
// F2 =
//     0.0064
//     0.
//     0.
// F3 =
//     0.
//     0.0032
//     0.0024
// F4 =
```

```
//    − 0.0064
//      0.
//      0.
// T1 =
//      0.0024
//      0.
//      0.
// T2 =
//      0.
//      0.
//      0.
// T3 =
//      0.0024
//      0.
//      0.
// T4 =
//      0.
//      0.
//      0.
// Total torque acting on the rectangular planar
   loop in milli N/m T = 4.8*ax
```

Scilab code Exa 9.5 find Magnetic Susceptibility, H,Magentization M

```
//clear//
//Caption: Program to find Magnetic Susceptibility ,
   H,Magentization M
//Example9.5
//page 279
clc;
ur = 50; //relative permeability of ferrite material
u0 = 4*%pi*1e-07; //free space permeability in H/m
chim = ur-1; //magnetic susceptibility
B = 0.05; //magnetic flux density in tesla
u = u0*ur;
```

```
H = B/u; //magnetic field intensity in A/m
M = chim*ceil(H); //magnetization in A/m
disp(chim,'chim =')
disp(H,'H =')
disp(M,'M = ')
//Reuslt
//chim = 49.
//H =    795.77472
//M =    39004.
```

Scilab code Exa 9.6 find the boundary conditions on magnetic field

```
//clear//
//Caption: Program to find the boundary conditions on magnetic field
//Example9.6
//page 283
clc;
ax = sym('ax');
ay = sym('ay');
az = sym('az');
u1 = 4e-06; // relative permeability in medium1
u2 = 7e-06; //relative permeability in medium2
k = [80,0,0]; //in A/m
B1 = [2e-03,-3e-03,1e-03];//field in region1
aN12 = [0,0,-1];
//To find Normal Components of Magnetic Field
Bz = dot(B1,aN12);
BN1 = [0,0,-Bz];
BN1 = float(BN1);
BN2 = float(BN1);
//To Find the Tangential Components of Magnetic Field
Bt1 = float(B1 - BN1);
Ht1 = float(Bt1/u1);
```

```
v = cross_product(aN12,k);
Ht2 = float(Ht1-v');
Bt2 = float(u2*Ht2);
disp(BN1(1)*ax+BN1(2)*ay+BN1(3)*az,'BN1 =')
disp(BN2(1)*ax+BN2(2)*ay+BN2(3)*az,'BN2 =')
disp(Bt1(1)*ax+Bt1(2)*ay+Bt1(3)*az,'Bt1 =');
disp(Ht1(1)*ax+Ht1(2)*ay+Ht1(3)*az,'Ht1 =');
disp(Ht2(1)*ax+Ht2(2)*ay+Ht2(3)*az,'Ht2 =');
disp(Bt2(1)*ax+Bt2(2)*ay+Bt2(3)*az,'Bt2 =');
//Total Magnetic Field Region2
B2 = (BN2+Bt2)*1e03;
B2 = B2(1)*ax+B2(2)*ay+B2(3)*az;
disp(B2,'Total Magnetic Field Region2 in milli Tesla B2 =')
//Result
// BN1 =
//   0.001*az
//BN2 =
// 0.001*az
//Bt1 =
// 0.002*ax-0.003*ay
//Ht1 =
// 500.0*ax-750.0*ay
//Ht2 =
// 500.0*ax-670.0*ay
//Bt2 =
// 0.0035*ax-0.00469*ay
//Total Magnetic Field Region2 in milli Tesla B2 =
// 1.0*az-4.69*ay+3.5*ax
```

Scilab code Exa 9.7 magnetomotive force 'Vm'

```
//clear//
//Caption: Program to find find magnetomotive force 'Vm' and reluctance 'R'
```

```
//Example9.7
//page 288
clc;
u0 = 4*%pi*1e-07 ;//free space permeability in H/m
ur = 1;//relative permeability
u = u0*ur;
dair = 2e-03; //air gap in toroid
dsteel = 0.3*%pi;
S = 6e-04; //area of cross section in square metre
B = 1; //flux density 1 tesla
N = 500; //number of turns
Rair = dair/(u*S);
disp(Rair,'Reluctance in A.t/Wb Rair =')
phi = B*S;
disp(phi,'Magnetic Flux in weber phi =')
Vm_air = S*Rair;
disp(Vm_air,'mmf required for the air gap in A.t Vm_air =')
Hsteel = 200; //magnetic field intensity of steel in A/m
Vm_steel = Hsteel*dsteel;
disp(Vm_steel,'mmf required for the steel in A.t Vm_steel =')
disp(Vm_steel+Vm_air,'Totla mmf required for toroid in A.t Vm =')
I = (Vm_steel+Vm_air)/N;
disp(I,'Total coil current in Amps I =')
//Result
//Reluctance in A.t/Wb Rair = 2652582.4
//Magnetic Flux in weber phi = 0.0006
//mmf required for the air gap in A.t Vm_air = 1591.5494
//mmf required for the steel in A.t Vm_steel = 188.49556
//Totla mmf required for toroid in A.t Vm = 1780.045
//Total coil current in Amps I =   3.56009
```

Scilab code Exa 9.8 total Magnetic Flux Density

```
//clear//
//Caption: Program to find total Magnetic Flux
   Density in Weber
//Example9.8
//page 289
clc;
I = 4; //current through toroid in Amps
r = 1e-03; //air gap radius in metre
Hphi = I/(2*%pi*r);
u0 = 4*%pi*1e-07 ;//free space permeability in H/m
ur = 1;//relative permeability
u = u0*ur;
N = 500;//number of turns
S = 6e-04; //cross section area in square metre
Rair = 2.65e06; //reluctance in air A.t/Wb
Rsteel = 0.314e06; //reluctance in steel A.t/Wb
R = Rair+Rsteel;//total reluctance in A.t/Wb
Vm = I*500; //total mmf in A.t
phi = Vm/R;//total flux in webers
B = phi/S; //flux density in Wb/Square metre
disp(B,'Magentic Flux Density in tesla B =')
//Result
//Magentic Flux Density in tesla B = 1.1246064
```

Scilab code Exa 9.9 self inductances and Mutual Inductances

```
//clear//
//Caption: Program to calculate self inductances and
    Mutual Inductances between two coaixal solenoids
//Example9.9
```

```
//page 297
clc;
n1 = sym('n1');
n2 = sym('n2');
I1 = sym('I1');
I2 = sym('I2');
az = sym('az');
R1 = sym('R1');
R2 = sym('R2');
u0 = sym('u0');
H1 = n1*I1*az;
disp(H1,'H1 =');
H2 = n2*I2*az;
disp(H2,'H2 =');
S1 = float(%pi*R1^2);
S2 = float(%pi*R2^2);
Hz =  float(H1/az);
phi12 = float(u0*Hz*S1);
disp(phi12,'phi12 = ')
M12 = n2*phi12/I1;
disp(M12,'M12 =')
//R1 = 2e-02;
//R2 = 3e-02;
//n1 = 50*100; //number of turns/m
//n2 = 80*100; //number of turns/m
//u0 = 4*%pi*1e-07;
M12 = float(limit(M12,R1,2e-02));
M12 = float(limit(M12,R2,3e-02));
M12 = float(limit(M12,n1,5000));
M12 = float(limit(M12,n2,8000));
M12 = float(limit(M12,u0,4*%pi*1e-07));
disp(M12*1e03,'Mutual Inductance in mH/m M12=')
L1 = u0*n1^2*S1;
L1 = float(limit(L1,u0,4*%pi*1e-07));
L1 = float(limit(L1,n1,5000));
L1 = float(limit(L1,R1,2e-02));
disp(L1*1e3,'Self Inductance of solenoid 1 in mH/m L1 =')
```

```
L2 = u0*n2^2*S2;
L2 = float(limit(L2,u0,4*%pi*1e-07));
L2 = float(limit(L2,n2,8000));
L2 = float(limit(L2,R2,3e-02));
disp(L2*1e3,'Self Inductance of solenoid 1 in mH/m L2 =')
//Result
// H1 =    az*n1*I1
// H2 =     az*n2*I2
// phi12 =      3.141592653011903*n1*u0*I1*R1^2
// M12 =     3.141592653011903*n1*n2*u0*R1^2
// Mutual Inductance in mH/m M12=    63.16546815077
// Self Inductance of solenoid 1 in mH/m L1 = 39.47841759423
// Self Inductance of solenoid 1 in mH/m L2 = 227.39568534276
```

Chapter 11

Transmission Lines

Scilab code Exa 11.1 determine the total voltage

```
//clear//
//Caption:Program to determine the total voltage as
   a function
//of time and position in a loss less transmisson
   line
//Example11.1
//page342
//syms z,t,B,w,Vo;
VST = sym('2*Vo*cos(B*z)');
V_zt = VST*sym('cos(w*t)');
disp(V_zt,'V(z,t)=')
//Result
//V(z,t)= 2*Vo*cos(t*w)*cos(z*B)
```

Scilab code Exa 11.2 characteristic impedance

```
//clear//
//Caption:Program to find the characteristic
```

```
impedance, the phase constant an the phase velocity
//Example11.2
//page344
clear;
clc;
close;
L = 0.25e-6; //0.25uH/m
C = 100e-12; //100pF/m
f = 600e06; //frequency f = 100MHz
W = 2*%pi*f; //angular frequency
Zo = sqrt(L/C);
B = W*sqrt(L*C);
Vp = W/B;
disp(Zo,'Characteristic Impedance in ohms Zo =')
disp(B,'Phase constant in rad/m B=')
disp(Vp,'Phase velocity in m/s Vp=')
//Result
//Characteristic Impedance in ohms Zo =
//     50.
//Phase constant in rad/m B=
//     18.849556
//Phase velocity in m/s Vp=
//     2.000D+08
```

Scilab code Exa 11.3 magnitude and phase of characteristic

```
//clear//
//Caption:Program tofind the magnitude and phase of characteristic
//impedance Zo
//Example11.3
//page347
Zo = sym('sqrt(L/C)*(1-sqrt(-1)*R/(2*W*L))');
teta = sym('atan(-R/(2*W*L))');
```

```
disp(Zo,'Characteristic impedance  Zo =')
disp(teta,'The phase angle teta=')
//Result
//Characteristic impedance  Zo =
//   sqrt(L/C)*(1-%i*R/(2*L*W))
//The phase angle teta=
// -atan(R/(2*L*W))
```

Scilab code Exa 11.4 output power and attenuation coefficient

```
//clear//
//Caption:Program to find the output power and attenuation coefficient
//Example11.4
//page349
clear;
clc;
close;
z = 20; //distance in meters
Pz_P0_dB = -2; //fraction of power drop in dB
Pz_P0 = 10^(Pz_P0_dB/10);
disp(Pz_P0,'Fraction of input power reaches output P(z)/P(0)=')
P0_mid_dB = -1; //fraction of power drop at midpoint in dB
P0_mid = 10^(P0_mid_dB/10);
disp(P0_mid,'Fraction of the input power reaches the midpoint P(10)/P(0)=')
alpha = -Pz_P0_dB/(8.69*z);
disp(alpha,'attenuation in Np/m alpha=')
//Result
//Fraction of input power reaches output P(z)/P(0)=
//      0.6309573
//Fraction of the input power reaches the midpoint P(10)/P(0)=
```

```
//     0.7943282
//attenuation in Np/m alpha=
//     0.0115075
```

Scilab code Exa 11.5 power dissipated in the lossless

```
//clear//
//Caption:Program to find the power dissipated in
   the lossless
//transmission line
//Example11.5
//page352
clc;
close;
ZL = 50-%i*75; //load impedance in ohms
Zo = 50; //characteristic impedance in ohms
R = reflection_coeff(ZL,Zo);
Pi = 100e-03; //input power in milliwatts
Pt = (1-abs(R)^2)*Pi;//power dissipated by the load
disp(R,'Reflection coefficient R =')
disp(Pt*1000,'power dissipated by the load in milli
   watss Pt=')
//Result
//Reflection coefficient R =    0.36 - 0.48i
//power dissipated by the load in milli watss Pt =
   64.
```

Scilab code Exa 11.6 find the total loss

```
//clear//
//Caption:Program to find the total loss in lossy
   lines
//Example11.6
```

```
//page352-353
clc;
close;
L1 = 0.2*10;//loss(dB) in first line of length =10 m
L2 = 0.1*15;//loss(dB) in second line of length =15m
R = 0.3; //reflection coefficient
Pi = 100e-03;//input power in milli watts
Lj = 10*log10(1/(1-abs(R)^2));
Lt = L1+L2+Lj;
Pout = Pi*(10^(-Lt/10));
disp(Lt,'The total loss of the link in dB is Lt=')
disp(Pout*1000,'The output power will be in milli
   watss Pout =')
//Result
//The total loss of the link in dB is Lt=
//     3.9095861
//The output power will be in milli watss Pout =
//    40.648207
```

Scilab code Exa 11.7 find the load impedance

```
//clear//
//Caption:Program to find the load impedance of a
   slotted line
//Example11.7
//page357
clear;
clc;
close;
S = 5; //standing wave ratio
T = (1-S)/(1+S); //reflection coefficient
Zo = 50; //characteristic impedance
ZL  = Zo*(1+T)/(1-T);
disp(ZL,'Load impedance of a slotted line in ohms ZL
  =')
```

```
//Result
//Load impedance of a slotted line in ohms ZL = 10.
```

Scilab code Exa 11.8 find the input impedance and power delivered

```
//clear//
//Caption:Program to find the input impedance and power delivered to
//the load
//Example11.8
//page363
clc;
close;
ZR1 = 300; //input impedance of first receiver
ZR2 = 300; //input impedance of second receiver
Zo = ZR1; //characteristic impedance = 300 ohm
Zc = -%i*300;//capacitive impedance
L = 80e-02;//length = 80 cm
Lambda = 1; //wavelength = 1m
Vth = 60; // voltage 300 volts
Zth = Zo;
ZL1 = parallel(ZR1,ZR2);
ZL = parallel(ZL1,Zc); //net load impedane
T = reflection_coeff(ZL,ZR2);//reflection coefficient
[R,teta1] = polar(T); //reflection coefficient in polar form
teta1 = real(teta1)*57.3;//teta value in degrees
S = VSWR(R); //voltage standing wave ratio
EL = electrical_length(L,Lambda);
EL = EL/57.3; //electrical length in degrees
Zin = Zo*(ZL*cos(EL)+%i*Zo*sin(EL))/(Zo*cos(EL)+%i*ZL*sin(EL));
disp(Zin,'Input Impedance in ohms Zin =')
Is = Vth/(Zth+Zin);//source current in amps
```

```
[Is,teta2] = polar(Is);//source current in polar form
Pin = (1/2)*(Is^2)*real(Zin);
PL = Pin; //for lossless line
disp(Pin,'Power delivered to a loss less line in watss PL =')
//Result
//Input Impedance in ohms Zin =   755.49551 - 138.46477i
// Power delivered to a loss less line in watss PL = 1.2
```

Scilab code Exa 11.9 find the input impedance

```
//clear//
//Caption:Program to find the input impedance for a line terminated with pure capacitive impedance
//Example11.9
//page363
clc;
close;
ZL = -%i*300; //load impdance is purely capacitive impedance
ZR  = 300;
T = reflection_coeff(ZL,ZR);//reflection coefficient in rectandular form
[R,teta] = polar(T);//reflection coefficient in polar form
S = VSWR(R)
if(S ==%inf)
  Zo = ZR;
end
Zin =Zo*(ZL*cos(EL)+%i*Zo*sin(EL))/(Zo*cos(EL)+%i*ZL*sin(EL));
disp(T,'Reflection coefficient in rectangular form')
```

```
disp(S,'Voltage Standing Wave Ratio S=')
disp(Zin,'Input impedance in ohms Zin =')
//Result
//Reflection coefficient in rectangular form
//    - i
//Voltage Standing Wave Ratio S=
//     Inf
//Input impedance in ohms Zin =
//     588.78315i
```

Scilab code Exa 11.10 find the input impedance

```
//clear//
//Caption:Program to find the input impedance for a line terminated with impedance(with inductive reactance)
//Example11.10
//page369
clc;
close;
ZL = 25+%i*50; //load impdance in ohms
Zo  = 50; //characteristic impedance in ohms
T = reflection_coeff(ZL,Zo);//reflection coefficient in rectandular form
[R,teta] = polar(T);//reflection coefficient in polar form
L = 60e-02;//length 60 cm
Lambda = 2; //wavelength = 2m
EL = electrical_length(L,Lambda);
EL = EL/57.3; //electrical length in radians
Zin =(1+T*exp(-%i*2*EL))/(1-T*exp(-%i*2*EL));
disp(Zin,'Input impedance in ohms Zin =')
//Result
//Input impedance in ohms Zin =
//     0.2756473 - 0.4055013i
```

Scilab code Exa 11.11 Steady state voltage

```
//clear//
//Caption:
//Example11.11
//page381
clc;
close;
Rg = 50; //series resistance with battery in ohms
Zo = Rg; //characteristic impedance
RL = 25; //load resistance
Vo = 10; //battery voltage in volts
V1_S = (Rg/(Zo+Rg))*Vo;
T = reflection_coeff(RL,Zo);
V1_R = T*V1_S;
I1_S = V1_S/Zo;
I1_R = -V1_R/Zo;
IB = Vo/(Zo+RL);
VL = Vo*(RL/(Rg+RL));
disp(V1_S,'Voltage at source in volts V1plus =')
disp(V1_R,'Voltage returns to battery in volts V1minus=')
disp(I1_S,'Current at battery in amps I1plus=')
disp(I1_R,'Current at battery in amps I1minus=')
disp(IB,'Steady state current through battery in amps IB=')
disp(VL,'Steady state load voltage in volts VL=')
//Result
//Voltage at source in volts V1plus =
//     5.
//Voltage returns to battery in volts V1minus=
//  - 1.6666667
//Current at battery in amps I1plus=
//    0.1
```

```
//Current at battery in amps I1minus=
//     0.0333333
//Steady state current through battery in amps IB=
//     0.1333333
//Steady state load voltage in volts VL=
//     3.3333333
```

Scilab code Exa 11.12 voltage and current through a resistor

```
//clear//
//Caption:Program to plot the voltage and current through a resistor
//Example11.12
//page 386
clear;
close;
clc;
t1 = 0:0.1:2;
t2 = 2:0.1:4;
t3 = 4:0.1:6;
t4 = 6:0.1:8;
VR=[40*ones(1,length(t1)),-20*ones(1,length(t2)),10*ones(1,length(t3)),-5*ones(1,length(t4))];
IR =[-1.2*ones(1,length(t1)),0.6*ones(1,length(t2)),-0.3*ones(1,length(t3)),0.15*ones(1,length(t4))];
subplot(2,1,1)
a=gca();
a.x_location = "origin";
a.y_location = "origin";
a.data_bounds = [0,-100;10,100];
plot2d([t1,t2,t3,t4],VR,5)
xlabel('

t')
```

```
ylabel('                              VR')
title('Resistor Voltage as a function of time')
subplot(2,1,2)
a=gca();
a.x_location = "origin";
a.y_location = "origin";
a.data_bounds = [0,-1.4;10,1.4];
plot2d([t1,t2,t3,t4],IR,5)
xlabel('

   t')
ylabel('                              IR')
title('Current through Resistor as a function of
   time')
```

Chapter 12

The Uniform Plane Wave

Scilab code Exa 12.1 phasor of forward propagating field

```
//clear//
//Caption:Program to determine the phasor of forward propagating field
//Example12.1
//page400
clc;
close;
Eyzt = sym('100*exp(%i*10^8*t-%i*0.5*z+30)');
Eysz = sym('100*exp(%i*10^8*t-%i*0.5*z+30)*exp(-%i*10^8*t)');
disp(Eyzt)
disp(Eysz,'Forward Propagating Field in phasor form =')
//Result
//100*exp(-0.5*%i*z+100000000*%i*t+30)
// Forward Propagating Field in phasor form =100*exp(30-0.5*%i*z)
```

Scilab code Exa 12.2 determine the instanteous field

```
//clear//
//Caption:Program to determine the instanteous field of a wave
//Example12.2
//page400-401
clc;
t = sym('t');
z = sym('z');
Ezt1 =sym('100*cos(-0.21*z+2*%pi*1e07*t)');
Ezt2 = sym('20*cos(-0.21*z+30+2*%pi*1e07*t)');
ax = sym('ax');
ay = sym('ay');
Ezt = Ezt1*ax+Ezt2*ay;
disp(Ezt,'The real instantaneous field Ezt =')
//Result
//The real instantaneous field Ezt =
//   100*ax*cos(0.21*z-2.0E+7*%pi*t)+20*ay*cos(0.21*z-2.0E+7*%pi*t-30)
//
```

Scilab code Exa 12.3 find the Phase constant, Phase velocity, Electric Field

```
//clear//
//Caption:Program to find the Phase constant, Phase velocity, Electric Field Intensity and Intrinsci ratio.
//Example12.3
//page408
clc;
syms t;
z = %z;
[uo,eo] = muo_epsilon();
ur = 1;
f = 10^6;
```

```
er1 = 81;
er2 =0;
etta0 = 377;
Ex0 = 0.1;
beta1 = phase_constant_dielectric(uo,eo,f,er1,er2,ur);
disp(beta1,'phase constant in rad/m beta=')
Lambda = 2*%pi/beta1;
Vp = phase_velocity(f,beta1);
disp(Vp,'Phase velocity in m/sec')
etta =  intrinsic_dielectric(etta0,er1,er2)
disp(etta,'Intrinsic impedancein ohms =')
Ex = 0.1*cos(2*%pi*f*t-beta1*z)
disp(Ex,'Electric field in V/m Ex=')
Hy = Ex/etta;
disp(Hy,'Magnetic Field in A/m Hy=')
//Result
// phase constant in rad/m beta=    0.1886241
// Phase velocity in m/sec =      33310626.
// Intrinsic impedancein ohms =       41.888889
// Electric field in V/m Ex=    cos(58342*z/309303-81681409*t/13)/10
//equivalent to Ex = 0.1*cos(0.19*z-6283185.3*t)
// Magnetic Field in A/m Hy = 9*cos(58342*z/309303-81681409*t/13)/3770
//equivalent to Hy = 0.0023873*cos(0.19*z-6283185.3*t)
```

Scilab code Exa 12.4 find the penetration depth and intrinsic impedance

```
//clear//
//Caption:Program to find the penetration depth and intrinsic impedance
//Example12.4
//page409
```

```
clc;
f = 2.5e09;//high microwave frequency = 2.5GHz
er1 = 78;//relative permittivity
er2 = 7;
C = 3e08; //free space velocity in m/sec
[uo,eo] = muo_epsilon(); //free space permittivity and permeability
ur = 1; //relative permeability
etta0 = 377; //free space intrinsic imedance in ohms
alpha = attenuation_constant_dielectric(uo,eo,f,er1,er2,ur);
etta = intrinsic_dielectric(etta0,er1,er2);
disp(alpha,'attenuation constant in Np/m alpha=')
disp(etta,'Intrinsic constant in ohms etta=')
//Result
//attenuation constant in Np/m alpha=     20.727602
// Intrinsic constant in ohms etta=       42.558673 + 1.9058543i
```

Scilab code Exa 12.5 find the attenuation constant,propagation constant

```
//clear//
//Caption:Program to find the attenuation constant,propagation constant and intrinsic impedance
//Example12.5
//page412
clc;
f = 2.5e09;//high microwave frequency = 2.5GHz
er1 = 78;//relative permittivity
er2 = 7;
C = 3e08; //free space velocity in m/sec
[uo,eo] = muo_epsilon(); //free space permittivity and permeability
ur = 1; //relative permeability
etta0 = 377; //free space intrinsic imedance in ohms
```

```
alpha = attenuation_constant_gooddie(uo,eo,f,er1,er2
    ,ur);
etta = intrinsic_good_dielectric(etta0,er1,er2);
beta1 = phase_constant_gooddie(uo,eo,f,er1,er2,ur);
disp(alpha,'attenuation constant per cm alpha=')
disp(beta1,'phase constant in rad/m beta1 =')
disp(etta,'Intrinsic constant in ohms etta=')
//Result
//attenuation constant per cm alpha=
//      20.748417
//phase constant in rad/m beta1 =
//     462.3933
//Intrinsic constant in ohms etta=
//     42.558673 + 1.9058543i
```

Scilab code Exa 12.6 find skin depth, loss tangent and phase velocity

```
//clear//
//Caption:Program to find skin depth, loss tangent
    and phase velocity
//Example12.6
//page419
clc;
f1 = 1e06; //frequency in Hz
//er1 = 81;
ur = 1;
[uo,eo] = muo_epsilon();//free space permittivity
    and permeability
sigma = 4;//conductivity of a conductor in s/m
[del] = SkinDepth(f1,uo,ur,sigma);
pi = 22/7;
Lambda = 2*pi*del;
Vp = 2*pi*f1*del;
disp(del*100,'skin depth in cm delta =')
disp(Lambda,'Wavelength in metre Lambda =')
```

```
disp(Vp,'Phase velocity in m/sec Vp =')
//Result
//skin depth in cm delta =
//      25.17737
//Wavelength in metre Lambda =
//     1.5825775
//Phase velocity in m/sec Vp =
//     1582577.5
```

Scilab code Exa 12.7 Electric field

```
//clear//
//
clc;
s = sym('s');
B = sym('B');
Eo = sym('Eo');
z = sym('z');
ax = sym('ax');
EsL = Eo*(ax+%i*ay)*exp(%i*s)*exp(-%i*B*z);
EsR = Eo*(ax-%i*ay)*exp(-%i*B*z);
Est = Eo*exp(%i*s/2)*(2*cos(s/2)*ax-%i*2*%i*sin(s/2)*ay)*exp(-%i*B*z);
disp(EsL,'Left circularly polarized field EsL=')
disp(EsR,'Right circularly polarized field EsR=')
disp(Est,'Total Elecetric field of a linearly polarized wave EsT =')
//Result
//Left circularly polarized field EsL=
//   (%i*ay+ax)*Eo*exp(%i*s-%i*z*B)
//Right circularly polarized field EsR=
//   (ax-%i*ay)*Eo*%e^-(%i*z*B)
//Total Elecetric field of a linearly polarized wave EsT =
// Eo*(2*ay*sin(s/2)+2*ax*cos(s/2))*exp(%i*s/2-%i*z*
```

B)

Chapter 13

Plane Wave Reflection and Dispersion

Scilab code Exa 13.1 electric field of incident, reflected and transmitted waves

```
//clear//
//Caption:Program to finid the electric field of
   incident , reflected and transmitted waves
//Example13.1
//page439
etta1 = 100;
etta2 = 300; //intrinsic impedance in ohms
T = reflection_coefficient(etta1,etta2);
Ex10_i = 100;//incident electric field in v/m
Ex10_r = T*Ex10_i;//reflected electric field in v/m
Hy10_i = Ex10_i/etta1;//incident magnetic field A/m
Hy10_r = -Ex10_r/etta1; //reflected magnetic field A
   /m
Si = (1/2)*Ex10_i*Hy10_i;//average incident power
   density in W/square metre
Sr = -(1/2)*Ex10_r*Hy10_r;//average reflected power
   denstiy in W/square metre
tuo = 1+T; //transmission coefficient
```

```
Ex20_t = tuo*Ex10_i; //transmitted electric field v/m
Hy20_t = Ex20_t/etta2; //transmitted magnetic field A/m
St = (1/2)*Ex20_t*Hy20_t; //average power density transmitted
disp(T,'reflection coefficient t =');
disp(Ex10_i,'incident electric field in v/m Ex10_i =')
disp(Ex10_r,'reflected electric field in v/m Ex10_r =')
disp(Hy10_i,'incident magnetic field A/m Hy10_i =')
disp(Hy10_r,'reflected magnetic field A/m Hy10_r=')
disp(Si,'average incident power density in W/square metre Si=')
disp(Sr,'average reflected power denstiy in W/square metre Sr=')
disp(St,'average power density transmitted in W/square metre St=')
//Result
//reflection coefficient t =      0.5
//incident electric field in v/m Ex10_i =     100.
//reflected electric field in v/m Ex10_r =     50.
//incident magnetic field A/m Hy10_i =     1.
//reflected magnetic field A/m Hy10_r=    - 0.5
//average incident power density in W/square metre Si=   50.
//average reflected power denstiy in W/square metre Sr=   12.5
//average power density transmitted in W/square metre St=     37.5
```

Scilab code Exa 13.2 maxima and minma electric field

```
//clear//
```

```
//Caption:Program to find the maxima and minma electric field
//Example13.2
//page443
clc;
er1 = 4;
ur1 = 1;
er2 = 9;
ur2 = 1;
[uo,eo] = muo_epsilon();//free space permittivity and permeability
u1 = uo*ur1; //permeability of medium 1
u2 = uo*ur2; //permeability of medium 2
e1 = eo*er1; //permittivity of medium 1
e2 = eo*er2; //permittivity of medium 2
etta1 = sqrt(u1/e1);
etta2 = sqrt(u2/e2);
T = reflection_coefficient(etta1,etta2)
Exs1_i = 100; //incident electric field in v/m
Exs1_r = -20; //reflected electric field in v/m
Ex1T_max = (1+abs(T))*Exs1_i;//maximum transmitted electric field in v/m
Ex1T_min = (1-abs(T))*Exs1_i;//minimum transmitted electric field in v/m
S = VSWR(T); //voltage standing wave ratio
disp(Ex1T_max,'maximum transmitted electric field in v/m =')
disp(Ex1T_min,'minimum transmitted electric field in v/m =')
disp(S,'voltage standing wave ratio S=')
//Result
//maximum transmitted electric field in v/m =
//      120.
//minimum transmitted electric field in v/m =
//     80.
//voltage standing wave ratio S=
//     1.5
```

Scilab code Exa 13.3 determine the intrinsic impedance

```
//clear//
//Caption:Program to determine the intrinsic
   impedance of the unkonwn material
//Eample13.3
//page441
clc;
maxima_spacing = 1.5;//Lambda/2 in metres
Lambda = 2*maxima_spacing; //wavelength in metres
C = 3e08;//free space velocity in m/sec
f = C/Lambda; //frequency in Hz
S = 5; //voltage standing wave ratio
T = (1-S)/(1+S); //reflection coefficient
etta0 = 377;//intrinsic impedance in ohms
ettau = etta0/S;//intrinsic impedance of unkonwn
   material in ohms
disp(T,'reflection coefficient T=')
disp(ettau,'intrinsic impedance in ohms =')
//Result
//reflection coefficient T =   - 0.6666667
// intrinsic impedance in ohms =       75.4
```

Scilab code Exa 13.4 determine the required range of glass thickness

```
//clear//
//Caption:Program to determine the required range of
    glass thickness for Fabry-perot interferometer
//Example13.4
//page450
clear;
clc;
```

```
Lambda0 = 600e-09; //wavelength of red part of visible spectrum 600nm
n = 1.45;//refractive index of glass plate
delta_Lambda = 50e-09; //optical spectrum of full width = 50nm
l = Lambda0^2/(2*n*delta_Lambda);
disp(l*1e06,'required range of glass thickness in micro meter l=')
//Result
//required range of glass thickness in micro meter l = 2.4827586
```

Scilab code Exa 13.5 Index for coating

```
//clear//
//Caption:Program to find the required index for the coating and its thickness
//Example13.5
//page451
clear;
clc;
etta1 = 377;//intrinsic impedance of free space in ohms
n3 = 1.45; //refractive index of glass
etta3 = etta1/n3;//intrinsic impedance in glass
etta2 = sqrt(etta1*etta3);//intrinsic impedance in ohms for coating
n2 = etta1/etta2; //refractive index of region2
Lambda0 = 570e-09;//free space wavelength
Lambda2 = Lambda0/n2; //wavelength in region2
l = Lambda2/4; //minimum thickness of the dielectric layer
disp(l*1e06,'minimum thickness of the dielectric layer in um =')
//Result
```

```
//minimum thickness of the dielectric layer in um =
//      0.1183398
```

Scilab code Exa 13.6 phasor expression

```
//clear//
//Caption:Program to find the phasor expression for the electric field
//Example13.6
//page456
clc;
ax = sym('ax');
ay = sym('ay');
az = sym('az');
x = sym('x');
y = sym('y');
z = sym('z');
teta = 30;  //phase angle in degrees
teta = 30/57.3; //phase angle in radians
Eo = 10; //Electric field in v/m
f = 50e06; //frequency in Hz
er = 9.0; //relative permittivity
ur = 1; //relative permeability
[uo,eo] = muo_epsilon();
k = propagation_constant(f,uo,ur,eo,er);
K = k*(cos(teta)*ax+sin(teta)*ay);
r = x*ax+y*ay;
Es = Eo*exp(-sqrt(-1)*K*r)*az;
disp(K,'propagation constant per metre K=')
disp(r,'distance in metre r=')
disp(Es,'Phasor expression for the electric field of the uniform plane wave Es=')
//Result
//K=5607*(14969*ay/29940+25156*ax/29047)/1784
// r=  ay*y+ax*x
```

```
//Es=10*az*%e^-(5607*%i*(14969*ay/29940+25156*ax/29047)*(ay*y+ax*x)/1784)
```

Scilab code Exa 13.7 find the fraction of incident power

```
//clear//
//Caption:Program to find the fraction of incident power that is reflected and transmitted
//Example13.7
//page460
clc;
teta1 = 30; //incident angle in degrees
n2 = 1.45;//refractive index of glass
teta2 = snells_law(teta1,n2);
etta1 = 377*cos(teta1/57.3); // intrinsic impedance in medium 1 in ohms
etta2 = (377/n2)*cos(teta2); //intrinsic impedance in medium2 in ohms
Tp = reflection_coefficient(etta1,etta2);//reflection coefficient for p-polarization
Reflected_Fraction_p = (abs(Tp))^2;
Transmitted_Fraction_p = 1-(abs(Tp))^2;
etta1s = 377*sec(teta1/57.3); //intrinsic impedance for s-polarization
etta2s = (377/n2)*sec(teta2);
Ts = reflection_coefficient(etta1s,etta2s);//reflection coefficient for s-polarization
Reflected_Fraction_s = (abs(Ts))^2;
Transmitted_Fraction_s = 1-(abs(Ts))^2;
disp(teta2*57.3,'Transmission angle using snells law in degrees teta2 =')
disp(Tp,'Reflection coefficient for p-polarization Tp=')
disp(Reflected_Fraction_P,'Fraction of incident power that is reflected for p-polarization =')
```

```
disp(Transmitted_Fraction_p,'Fraction of power transmitted for p-polarization =')
disp(Ts,'Reflection coefficient for s-polarization Tp=')
disp(Reflected_Fraction_s,'Fraction of incident power that is reflected for s-polarization =')
disp(Transmitted_Fraction_s,'Fraction of power transmitted for s-polarization =')
//Result
//Transmission angle using snells law in degrees teta2 =
//     20.171351
//Reflection coefficient for p-polarization Tp=
//  - 0.1444972
//Fraction of incident power that is reflected for p-polarization =
//    0.0337359
//Fraction of power transmitted for p-polarization =
//   0.9791206
//Reflection coefficient for s-polarization Tp=
//   - 0.2222748
//Fraction of incident power that is reflected for s-polarization =  //    0.0494061
//Fraction of power transmitted for s-polarization =
//   0.9505939
```

Scilab code Exa 13.8 find the refractive index

```
//clear//
//Caption:Program to find the refractive index of the prism material
//Example13.8
//page463
clear;
clc;
```

```
n2 =1.00; //refractive index of air
teta1 = 45; //incident angle in degrees
teta1 = 45/57.3;//incident angle in radians
n1 = n2/sin(teta1);
disp(n1,'refractive index of prism material n1=')
//Result
//refractive index of prism material n1=
//     1.4142954
```

Scilab code Exa 13.9 determine incident and transmitted anlges

```
//clear//
//Caption:Program to determine incident and transmitted anlges
//Example13.9
//page464
clear;
clc;
n1 =1.00; //refractive index of air
n2 =1.45; //refractive index of glass
teta1 = asin(n2/sqrt(n1^2+n2^2));
teta2 = asin(n1/sqrt(n1^2+n2^2));
Brewster_Condition = teta1+teta2;
disp(teta1*57.3,'Incident angle in degrees teta1 =')
disp(teta2*57.3,'transmitted angle in degrees teta2=')
disp(Brewster_Condition*57.3,'sum of the incident angle and transmitted angle, Brewster_Condition=')
//Result
//Incident angle in degrees teta1 =  55.411793
//transmitted angle in degrees teta2 = 34.594837
//sum of the incident angle and transmitted angle, Brewster_Condition=   90.00663
```

Scilab code Exa 13.10 group velocity and phase velocity

```
//clear//
//Caption:Program to determine group velocity and phase velocity of a wave
//Example13.10
//page470
clc;
w = sym('w');
wo = sym('wo');
no = sym('no');
c = sym('c');
beta_w = (no*w^2)/(wo*c);
disp(beta_w,'Phase constant=')
d_beta_w = diff(beta_w,w);
disp(d_beta_w,'Differentiation of phase constant w.r.to w =')
Vg = 1/d_beta_w;
Vg = limit(Vg,w,wo);
Vp = w/beta_w;
Vp = limit(Vp,w,wo);
disp(Vg,'Group velocity =')
disp(Vp,'Phase velocity=')
//Result
//Phase constant=
//   no*w^2/(c*wo)
//Differentiation of phase constant w.r.to w =
//   2*no*w/(c*wo)
//Group velocity =
//   c/(2*no)
//Phase velocity=
//   c/no
```

Scilab code Exa 13.11 pulse width at the optical fiber

```
//clear//
//Caption:Program to determine the pulse width at
   the optical fiber output
//Example13.11
//page474
clear;
clc;
T = 10; //width of light pulse at the optical fiber
   input in pico secs
beta2 = 20; //dispersion in pico seconds square pre
   kilometre
z = 15; // length of optical fiber in kilometre
delta_t = beta2*z/T;
T1 = sqrt(T^2+delta_t^2);
disp(delta_t,'Pulse spread in pico seconds delta_t =
   ')
disp(ceil(T1),'Output pulse width in pico seconds T1
    =')
//Result
//Pulse spread in pico seconds delta_t =
//     30.
//Output pulse width in pico seconds T1 =
//    32.
```

Chapter 14

Guided Wave and Radiation

Scilab code Exa 14.1 determine the cutoff frequency

```
//clear//
//Caption:Program to determine the cutoff frequency for the first waveguide mode(m=1)
//Example14.1
//page 499
clear;
clc;
er1 = 2.1; //dielectric constant of teflon material
er0 = 1; //dielectric constant of air
d = 1e-02; //parallel plate waveguide separation in metre
C = 3e08; //free space velocity in m/sec
n = sqrt(er1/er0); //refractive index
fc1 = C/(2*n*d);
disp(fc1,'cutoff frequency for the first waveguide mode in Hz fc1 =')
//Result
//cutoff frequency for the first waveguide mode in Hz fc1 =
//     1.035D+10
```

Scilab code Exa 14.2 number of modes propagate in waveguide

```
//clear//
//Caption:Program to determine the number of modes
   propagate in waveguide
//Example14.2
//page 499
clear;
clc;
er1 = 2.1; //dielectric constant of teflon material
er0 = 1; //dielectric constant of air
n = sqrt(er1/er0); //refractive index
Lambda_cm = 2e-03; //operating cutoff wavelength in
   metre
d = 1e-02; //parallel-plate waveguide separation
m = (2*n*d)/Lambda_cm;
disp(floor(m),'Number of waveguides modes propagate
   m =')
//Result
//Number of waveguides modes propagate m =
//     14.
```

Scilab code Exa 14.3 determine the group delay and difference

```
//clear//
//Caption:Program to determine the group delay and
   difference in propagation times
//Example14.3
//page 502
clc;
C = 3e08; //free space velocity in m/sec
er = 2.1; //dielectric constant of teflon material
```

```
fc1 = 10.3e09;//cutoff frequency for mode m =1
fc2 = 2*fc1; //cutoff frequency for mode m =2
f = 25e09; //operating frequency in Hz
Vg1 = group_delay(C,er,fc1,f);//group delay for mode
    m = 1
Vg2 = group_delay(C,er,fc2,f);//group delay for mode
    m = 2
del_t = group_delay_difference(Vg1,Vg2);
disp(ceil(del_t*1e10),'group delay difference in ps/
   cm del_t=')
//Result
//group delay difference in ps/cm del_t=
//     33.
```

Scilab code Exa 14.4 determine the operating range

```
//clear//
//Caption:Program to determine the operating range
    of frequency for TE10 mode of air filled
    rectangular waveguide
//Example14.4
//page 509
clear;
clc;
 //dimensions of air filled rectangular waveguide
a = 2e-02;
b = 1e-02;
//Free space velocity in m/sec
C = 3e08;
//the value of m for TE10 mode
m = 1;
n = 1;//refractive index for air filled waveguide
fc = (m*C)/(2*n*a);
disp(fc*1e-09,'Operating range of frequency for TE10
    mode in GHz fc=')
```

```
//Result
//Operating range of frequency for TE10 mode in GHz fc=
//      7.5
```

Scilab code Exa 14.5 maximum allowable refractive index

```
//clear//
//Caption: Program to determine the maximum allowable refractive index of the slab material
//Example14.5
//page 517
clear;
clc;
Lambda = 1.30e-06;//wavelength range over which single-mode operation
d = 5e-06;//slab thickness in metre
n2 = 1.45; //refractive index of the slab material
n1 = sqrt((Lambda/(2*d))^2+n2^2);
disp(n1,'The maximum allowable refractive index of the slab material n1=')
//Result
//The maximum allowable refractive index of the slab material n1=
//      1.4558159
```

Scilab code Exa 14.6 find the V number of a step index fiber

```
//clear//
//Caption:Program to find the V number of a step index fiber
//Example14.6
//page 524
```

```
clear;
clc;
Lambda = 1.55e-06; //operating wavelength in metre
LambdaC = 1.2e-06; //cutoff wavelength in metre
V = (LambdaC/Lambda)*2.405;
disp(V,'the V number of a step index fiber V=')
//Result
//the V number of a step index fiber V=
//      1.8619355
```