Scilab Textbook Companion for Propagation Engineering in Wireless Communications by Abdollah Ghasemi[1]

Created by
Sufiyan Siddique
BE
Electronics Engineering
A. I. Kalsekar Technical Campus, Mumbai University
College Teacher
None
Cross-Checked by
None

July 31, 2019

[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: Propagation Engineering in Wireless Communications

Author: Abdollah Ghasemi

Publisher: Springer, U. K

Edition: 2

Year: 2012

ISBN: 9781461410768

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

Introduction to Radiowaves

Scilab code Exa 1.1 Example 1

```
//Chapter 1, Example 1.1
clc
//Initialisation
fo=8387.5;        //in MHz

//Calculation
//defining a function for 6 MHZ channels with 14 MHz
    separation
deff('[fn]=F(n,fo)','fn=fo-108.5+(14*n)');
deff('[fn]=F1(n,fo)','fn=fo+10.5+(14*n)');

//defining a function for 12 MHZ channels with 7 MHz
    separation
deff('[fn]=F2(n,fo)','fn=fo-108.5+(7*n)');
deff('[fn]=F3(n,fo)','fn=fo+17.5+(7*n)');

//Result
printf("(1) 6-RF channels with 14 MHz separation")
printf("\n     f1 = %d",F(1,fo))
printf("\n     f11 = %d",F1(1,fo))
printf("\n     f2 = %d",F(2,fo))
```

```
printf("\n     f21 = %d",F1(2,fo))
printf("\n     f3 = %d",F(3,fo))
printf("\n     f31 = %d",F1(3,fo))
printf("\n\n(2) 12-RF channels with 7 MHz separation
   ")
printf("\n     f1 = %d",F2(1,fo))
printf("\n     f11 = %d",F3(1,fo))
printf("\n     f2 = %d",F2(2,fo))
printf("\n     f21 = %d",F3(2,fo))
printf("\n     f3 = %d",F2(3,fo))            //The
   answer provided in the textbook is wrong
printf("\n     f31 = %d",F3(3,fo))           //The
   answer provided in the textbook is wrong
```

Chapter 2

Basic Principles in Radiowave Propagation

Scilab code Exa 2.1 Example 1

```
//Chapter 2, Example 2.1, page 25
clc
//Initialisation
sig=0.005                                    //sigma
ur=1                                         //relative
   permeability
er=12                                        //relative
   permittivity
eo=8.85*10**-12                              //
   permittivity of a free space
f1=10*10**3                                  //frequency
    of radio wave 1
f2=10*10**9                                  //frequency
    of radio wave 2
pi=3.14

//Calculation
c1=sig/(2*pi*f1*eo*er)                  //conductivity
   at f1
```

```
c2=sig/(2*pi*f2*eo*er)                    //conductivity at f2


//Result
printf("conductivity at f1 = %.1f >> 1\n",c1)
printf("conductivity at f2 = %.1f x10^-4 >> 1",(c2*10**4))
```

Scilab code Exa 2.2 Example 2

```
//Chapter 2, Example 2.2, page 26
clc
//Initialisation
c1=3*10**8                        //speed of light in m/s
f1=100*10**6                      //frequency in hertz
f2=1*10**9                        //requency in hertz

//Calculation
v1=c1/(9)                         //velocity in m/s
v2=c1                             //velocity in m/s
h1=v1*f1**-1                      //wavelength at f1, v1
h2=v2*f1**-1                      //wavelength at f1, v2
h3=v1*f2**-1                      //wavelength at f2, v1
h4=v2*f2**-1                      //wavelength at f2, v2

```

```
//Result
printf("Velocity,")
printf("\nV1 = %.2f x10^7 m/s",(v1*10**-7))
printf("\nV2 = %.2f x10^8 m/s",(v2*10**-8))
printf("\n\nfor f1 = 100 MHz,")
printf("\nlambda1 = %f m",h1)
printf("\nlambda2 = %d m",h2)
printf("\n\nfor f2 = 1 GHz,")
printf("\nlambda1 = %.2f cm",(h3*10))
printf("\nlambda2 = %d cm",(h4*10**2))
```

Scilab code Exa 2.3 Example 3

```
//Chapter 2, Example 2_3, page 37
clc

//Initialisation
s=0.08                                   //medium conductivit
w=10**7                                  //angular velocity
e=8.85*10**-7                            //permitivity if free space
u=14                                     //medium permeability
uo=4*3.14*10**-7                         //permeability of free space
pi=3.14

//Calculation
f=w*(2*pi)**-1                           //frequency
a1=sqrt(f*pi*s*uo)                       //attenuation
b=a1                                     //phase
d=complex(a1,b)
```

```
y=d                                         //propagation constants
z=log10(0.5)/(-log10(exp(1))*2*a1)          //Depth of the land

//Result
printf("(1) Attenuation = %.1f Np/m",a1)
printf("\n    Phase = %.1f Rad/m",b)
printf("\n    Propagation constant = %.1f",real(y))
printf("\n                              + %.1f j rad/m",imag(y))
printf("\n(2) Depth of land = %.2f m",z)
```

Scilab code Exa 2.6 Example 6

```
//Chapter 2, Example 2.6, page 38
clc

//Initialisation
W=100*10**-12                               //power in watt
pi=3.14                                     //pi
no=120*pi

//Calculation
Em=sqrt(2*no*W)                             //effective value of E
Ee=Em/sqrt(2)                               //effective value of E
Hm=sqrt((2*10**-10)/(no))                   //effective value of H
He=Hm/sqrt(2)                               //effective value of H

//Result
```

```
printf("Em = %.1f  uV/m",(Em*10**6))
printf("\nEe = %.1f  uV/m",(Ee*10**6))
printf("\nHm = %.3f  uA/m",(Hm*10**6))
printf("\nHe = %.2f  uA/m",(He*10**6))
```

Scilab code Exa 2.7 Example 7

```
//Chapter 2, Example 2.7, page 39
clc
//Initialisation
f=7.5                                          //frequency in GHz
d=40                                           //link distance in Km
Pt=30                                          //transmitter power in dBm
La=15                                          //additional loss
Pth=-78                                        //RX threshold

//Calculation
FSL=92.4+(20*log10(f*d))                   //FSL
RSL=Pt-(0.4*FSL)-La                            //RSL
FM=RSL-Pth                                     //fade margin

//Result
printf("(1) Received signal level (RSL) = %.1f dBm",RSL)
printf("\n(2) Fade margin = %.1f dB",FM)
```

Scilab code Exa 2.8 Example 8

```
//Chapter 2, Example 2.8, page 45
clc
//Initialisation
Pt=10                                              //transmitter power in watt
Gt=5                                               //antenna power in dBm
Lt=2                                               //feeder loss in dB
d=8000                                             //distance in meter
pi=3.14                                            //pi
no=120*pi

//Calculation
EIRP=Pt+Gt-Lt
x=EIRP*10**-1
EIRP2=10**x                                        //Equivalent isotropic radiated power
Ed=sqrt(30*EIRP2)/d                           //Electric Field Intensity
W=(Ed**2)/(2*no)                                   //power in watt

//Result
printf("EIRP = %.1f W",EIRP2)
printf("\n|Ed| = %.2f mV/m",(Ed*10**3))
printf("\n W = %.1f nW/m^2",(W*10**9))
```

Scilab code Exa 2.9 Example 9

```
//Chapter 2, Example 2.9, page 47
clc
//Initialisation
FSL=128                                    //FSL in dB
```

```
Lb=135                                    //Sum of FSL and medium loss Lm
Lc=5
Gt=30                                     //transmitter gain in dB
Gr=30                                     //reciever gain in dB
Pr=-60                                    //received signal level

//Calculation
Lm=Lb-FSL                                 //medium loss in dB
Lm1=10**(Lm*10**-1)                       //medium loss
Pt=Lc+Lb-Gt-Gr+Pr                         //power in dBm
Pt1=10**(Pt*10**-1)                       //power in watt

//Result
printf("Medium Loss = %d",Lm1)
printf("\nPt = %.1f mW",(Pt1))
```

Scilab code Exa 2.10 `Example 10`

```
//Chapter 2, Example 2.10, page 50
clc
//Initialisation
ri1=1.00025                              //refractive index
ri2=1.00023                             //refractive index
h1=1                                     //height in Km
h2=1.5                                   //height in Km
n=1.00026                                //variation


```

```
//Calculation
deln=ri1-ri2
delh=h2-h1
d=deln/delh
R=n/d                              //radius of
   curvature


//Result
printf("Radiowave curvature radius , R = %.d Km",R)
```

Scilab code Exa 2.11 Example 11

```
//Chapter 2, Example 2.11, page 51
clc
//Initialisation
R=25000                                         //path
    curvature radius in Km
Re=6370                                         //
   Earth radius in Km


//Calculation
K=R*(R-Re)**-1                                  //K
   factor
Re1=K*Re                                        //
   equivalent radii of the Earth
R1=(1*Re1**-1)-(1*Re**-1)+(1*R**-1)
d=1*R1**-1                                      //
   equivalent radii of the path


//Result
printf("K = %.3f",K)
printf("\nRe1 = %d Km",Re1)
```

```
printf("\nR1 = %.1f Km\n",d)
printf("Therefore, R1 ~ infinity")
```

Chapter 3

Radiowave Propagation in Troposphere

Scilab code Exa 3.1 Example 1

```
//Chapter 3, Example 3.1, page 61
clc
//Initialisation
h=2                              //height in Km
h1=5                             //height in Km


//Calculation
t2=290-(6.5*h)                          //Proposed
   formula for height h=2Km
p2=950-117*h
e2=8-3*h
t21=294.98-5.22*h-0.007*h**2
p21=1012.82-111.56*h+3.86*h**2
p=14.35*2.72**(-0.42*h-0.02*h*h+0.001*h**3)

t5=290-6.5*h1                          //proposed
   formula for height h=5Km
p5=950-117*h1
```

```
e5=8-3*h1
t51=294.98-5.22*h1-0.007*h1**2
p51=1012.82-111.56*h1+3.86*h1**2
p1=14.35*2.72**(-0.42*h1-0.02*h1**2+0.001*h1**3)


//Results
printf("T(2) = %.1f K",t21)
printf("\nP(2) = %.2f hpa",p21)
printf("\np(2) = %.2f hpa",p)
printf("\nT(5) = %.1f K",t51)
printf("\nP(5) = %.2f hpa",p51)
printf("\np(5) = %.2f hpa",p1)
```

Scilab code Exa 3.2 Example 2

```
//Chapter 3, Example 3.2, page 63
clc

//Initialisation
h=2                                    //Height in Km
T=277                                  //Tempreture in Kelvin
p=716
e=2


//Calculation
er=1+(151.1/T)*(p+(4810*h/T))*10**-6
n=er**(0.515)                           //refractive index of the air
N=(n-1)*10**6                          //refractivity number


```

```
//Results
printf("n = %.5f",n)
printf("\nN = %d",N)
```

Scilab code Exa 3.3 Example 3

```
//Chapter 3, Example 3.3, page 67
clc
//Initialisation
er=1.001                                    //
    relative permittivity of a medium
dn=35*10**-6                                //
    vertical gradient of refractive index
Re=6370                                     //actual
    redius of earth
d=20                                        //
    transmitter and receiver distance in Km
d1=5
d2=15
K1=1.3333                                   //
    standard atmosphere condition

//Calculation
R=(er**0.5)/dn
K=R/(R-Re)
hm=(125*d**2)/(K*Re)                         //Earth
    bulge value in the middle of the path
h1=(500*d1*d2)/(K*Re)                        //h1
h2=(500*d1*d2)/(K1*Re)                       //h2


//Results
printf("Bulge value  = %.1f m",hm)
printf("\nBulge value, h1  = %.2f m",h1)
printf("\nBulge value, h2  = %.2f m",h2)
```

Scilab code Exa 3.4 Example 4

```
//Chapter 3, Example 3.4, page 68
clc
//Initialisation
K=1.33
d1=24                                                    //heigth
    in Km
d2=15                                                     //heigth
    in Km
K1=1
Re=6370                                                   //actual
    redius of earth

//Calculation
R=4.12*(d1**0.5+d2**0.5)
R1=K1*Re
Rrh=(2*R1*d1)**0.5+(2*R1*d2)**0.5

//Results
printf("K=1.33")
printf("\nRrh = %.1f km\n",R)
printf("K=1")
printf("\nRrh = %.1f km",Rrh)                                  //
    The answer provided in the textbook is wrong
```

Scilab code Exa 3.5 Example 5

```
//Chapter 3, Example 3.5, page 74
clc
//Initialisation
```

```
No=1                                            //
   index of refraction
N1=1.3*10**-7
h=20                                            //
   height

//Calculation
wc=asin(((4*No)/((4*No)+((h**2)*N1))))           //
   critical angle


//Results
printf("Critical angle = %f",wc)
   //answer is not written in textbook
```

Scilab code Exa 3.6 Example 6

```
//Chapter 3, Example 3.6, page 76
clc
//Initialisation
dn=-0.2                                   //air
   refractivity gradient
d=20                                      //height
b=0.074                                   //elevation
   angle from graph 3.10
f=7                                       //frequency in
    Ghz from graph 3.11
c=2*10**-6

///Calculation
t=0.156                                    //1000/6370
dm=dn+t
a=(-c*dm*d)**0.5                          //elevation
   angle of waves

```

```
//Results
printf("Elevation angle of waves = %.2f mrad",(a
    *10**3))
printf("\nElevation angle = %.3f",b)
                                              //from graph
    3.10
printf("\nMinimum frequency of coupling waves into
    the duct = %d Ghz",f)       //from graph 3.11
```

Scilab code Exa 3.7 Example 7

```
//Chapter 3, Example 3.7, page 80
clc

//Initialisation
f=18                                        //frequency in
    GHz
d=30                                        //in km
R=25                                        //rainfall
    intensity in mm

//Using Table 3.3
av15=1.128
av20=1.065
av18=1.09
kv15=0.0335
kv20=0.0691
kv18=0.0587

//Calculation
yr=kv18*R**av18                                //rain
    specific attenuation
de=(90*(90+d)**-1)*d
A=de*yr                                        //Maximum
    rain attenuation
```

```
//Results
printf("(1) Rain specific attenuation = %.2f dB/km",
    yr)
printf("\n(2) Maximum rain attenuation = %.1f dB",A)
```

Scilab code Exa 3.9 Example 9

```
//Chapter 3, Example 3.9, page 89
clc
//Initialisation
rh=-1
s=4                                               //sigma in S/
    m
f=5*10**9                                         //frequency
    in Hz
eo=8.85*10**-12                                   //permitivity
    of free space
er=75                                             //permitivity
    of medium
w1=30*3.14*180**-1                                //in radians
pi=3.14


//Calculation
w=2*pi*f
x=s*(w*eo)**-1
a=sin(w1)-sqrt((er-x)-cos(w1)**2)
a1=sin(w1)+sqrt((er-x)-cos(w1)**2)
rh1=a/a1
b1=(er-x)*sin(w1)-sqrt((er-x)-cos(w1)**2)
b2=(er-x)*sin(w1)+sqrt((er-x)-cos(w1)**2)
rv=-b1/b2
```

```
//Results
printf("(2) X = %.1f",x)
printf("\n(3) Rh = %.3f",rh1)
printf("\n     Rv = %.1f",rv)
```

Scilab code Exa 3.10 Example 10

```
//Chapter 3, Example 3.10, page 92
clc
//Initialisation
f=5*10**9          //frequency in Hz
c=3*10**8          //speed of light
h1=6               //in metre
h2=2               //in metre
pi=3.14

//Calculation
h=c*f**-1          //wavelength
w=atan(h1*2250**-1)          //grazing angle in radian
w1=w*180*pi**-1          //grazing angle in degree
a=((2*pi*h1*h2)*(h*300)**-1)*3.14*180**-1
```

```
e=sin(a)
F=e*2*180*pi**-1                                            //
    PGF value (wrong value calculated in textbook)
LR=20*log10(F)                                              //
    Decrease in received signal level


//Results
printf("(1) Grazing angle = %.2f degree",w1)
printf("\n(2) PGF value = %f",F)                            //
    The answer provided in the textbook is wrong
printf("\n(3) Decrease in received signal level = %
    .2f dB",LR)                      //The answer provided
    in the textbook is wrong
```

Scilab code Exa 3.11 Example 11

```
//Chapter 3, Example 3.11, page 98
clc

//Initialisation
h=12.5*10**-2               //in meter
d1=10*10**3                 //in meter
d2=15*10**3                  //in meter
d3=12.5*10**3                //in meter
d4=12.5*10**3                   //in meter
h=1.25                        //in Kilometer

//Calculation
r1=(((d1*d2)/(d1+d2))*h)**0.5          //radius of
    first and fourth Fresnel zones
r4=r1*(4)**0.5
```

```
R1=(((d3*d4)/(d3+d4))*h)**0.5          //radius of
    first and fourth ellipse zones
R4=R1*(4)**0.5

//Results
printf("Radius of first fresnel zones, r1 = %.2f m",
   r1)
printf("\nRadius of Second fresnel zones, r4= %.2f m
   ",r4)
printf("\nh = %.2f x 10^-4 Km",h)
printf("\nRadius of first ellipse , R1 = %.2f m",R1)
printf("\nRadius of second ellipse , R4 = %.1f m",R4)
```

Scilab code Exa 3.12 Example 12

```
//Chapter 3, Example 3.12, page 105
clc


//Initialisation
L=13200                                                //L
    parameter in m
H=10240                                                //H
    parameter
Re=6370000                                             //
    actual redius of earth
ht=30                                                  //
    height in m
hr=20                                                  // in
   m
re1=8453000                                            // in
    metre
h1=30000                                              // in
```

```
    metre
h2=20000                                          //in metre
dt1=22.5
f=10*10**9                                        //frequency in Hz
c=3*10**8                                         //speed of light
d=30000                                           //distance in m
pt=30                                             //transmitter antenna power
gt=40                                             //transmitter antenna gain
gr=40                                             //receiver antenna gain
pi=3.14
F3=-3
H=-34
D=0.75

//Calculation
dt=sqrt(2*re1*ht)
X=3*dt*L**-1
Z1=h1*H**-1
Z2=h2*H**-1
vx=10**-3.5                                      //from fig 3.26
z1=10**0.95                                      //from fig 3.27
z2=10**0.65                                      //from fig 3.27

//for d=3dt
lv=20*log10(vx)
lz1=20*log10(z1)
lz2=20*log10(z2)
F=(lv+lz1+lz2)*20**-1
```

```
F1=10**(F)
F11=20*log10(F1)
X1=2*dt*L**-1
d3=3
f3=-F11

vx1=10**-2.35                                    //from fig
    3.26
lv1=20*log10(vx1)

//for d=2dt
F4=1+D
F5=20*log10(F4)
d2=2
f2=-F5


//for d=1.1dt
F6=sqrt(1+D**2)
F7=20*log10(F6)
d11=1.1
f11=-F7

//for d=dt
d1=1
f1=0.2

//for plotting graph in terms of points



for N=0:1:5
    a=plot(1,0.2,'-o')
    a1=plot(1.1,-1.9,'-o')
    a2=plot(2,-4.8,'-o')
    a3=plot(3,-38,'-o')
end

```

```
title('Path gain F','fontsize',5);
xlabel("d/dt", "fontsize", 3);
ylabel("20log(F)(dB)", "fontsize", 3, "color", "blue
    ");
xstring(1,2,"d/dt",0,0);
xstring(1.2,0.7,"1.1d/dt",0,0);
xstring(2,-0.7,"2d/dt",0,0);
xstring(2.86,-35,"3d/dt",0,0);



h=c*f**-1

    //wavelength
Pr=pt+gt+gr+H+F3-10*log10(4*pi*d**2)
                                    //Received
    signal power


//Results
printf("(1) Effective receiver path gain F = %.4f",
    F11)
printf("\n(2) Path gain F plot is shown")
printf("\n(3) Received signal power Pr = %.1f dBm",
    Pr)
```

Scilab code Exa 3.13 Example 13

```
//Chapter 3, Example 3.13, page 109
clc

//Initialisation
eirp=800                                    //in KW
d=24                                        //in Km
a=0.03                                      //in
```

```
radian
d1=22                                          //in Km
d2=2                                           //in Km
h=0.4*10**-3                                   //
wavelength in m
Er=45                                          //in
microvolt

//Calculation
E=104.8+10*log10(eirp)-20*log10(d)                  //
field intensity
V=a*sqrt((2*d2*d1)/((d1+d2)*h))
                        //knife edge obstacle
attenuation
Lke=23

//from table 3.4
er=10**(Er*20**-1)

//Results
printf("(1) Electric field intensity = %.3f microV/m
",er)
```

Scilab code Exa 3.14 Example 14

```
//Chapter 3, Example 3.14, page 115
clc

//Initialisation
f1=430                                       //upper
frequency band
f2=410                                       //lower
frequency band
d=80                                         //distance
in meter
```

```

//Calculation
Yv=0.1                                          //
   Specific attenuation obtained from graph fig 3.34
Lv=Yv*d                                         //loss of
   forest trees
Am=((f1+f2)/2)**0.5                             //maximum
    value for trees excess loss.

//Results
printf("Specific attenuation index, Yv = %.1f dB/m",
   Yv)
printf("\nLoss of forest trees, Lv = %.1f dB",Lv)
printf("\nMaximum value for trees excess loss = %.1f
    dB",Am)
```

Scilab code Exa 3.15 Example 15

```
//Chapter 3, Example 3.15, page 118
clc

//Initialisation
d=40                                                //
   length in meter
Am=2                                                //area
    in square meter
f=10*10**9                                          //
   frequency in hertz

//Calculation
As=40                                           //using
   graph fig 3.36, As can be obtained
As1=30                                          //using
   graph fig 3.37, As can be obtained

```

```
//Results
printf("Loss in the summer for trees with leaves, As
    = %d dB",As)
printf("\nLoss in winter for trees without leaves,
   As = %d dB",As1)
```

Chapter 4

Radiowave Propagation in Ionosphere

Scilab code Exa 4.1 Example 1

```
//Chapter 4, Example 4.1, page 130
clc

//Initialisation
h=400                                        // height in Km
pd=1*10**8                                   // plasma density at height D
pe=1*10**10                                  // plasma density at height E
pf=3*10**11                                  // plasma density at height F
Wd=20*10**3                                  // thickness of D
We=40*10**3                                  // thickness of E
Wf=190*10**3                                 // thickness of F

```

```
//Calculation
tecd=Wd*pd                                          //total electron content at D
tece=We*pe                                          //total electron content at E
tecf=Wf*pf                                          //total electron content at F
tec=tecd+tece+tecf
tec1=tec*sqrt(2)                                    //total electron content

//Results
printf("(2) TEC (D) = %.1f x 10^12 el/m^2",(tecd/10**12))
printf("\n    TEC (E) = %.1f x 10^14 el/m^2",(tece/10**14))
printf("\n    TEC (F) = %.2f x 10^16 el/m^2",(tecf*10**-16))
printf("\n(3) TEC = %.1f x 10^16 el/m^2",(tec1/10**16))
```

Scilab code Exa 4.2 Example 2

```
//Chapter 4, Example 4.2, page 134
clc

//Initialisation
N=5*10**11                                          //Electron density in F layer

//Calculation
F=9*sqrt(N)                                         //f0F frequency

//Results
```

```
printf("(1) hmin = 200Km  hmax = 400Km")            //
   from graph
printf("\n(2) F = %.1f Mhz",(F*10**-6))
```

Scilab code Exa 4.3 Example 3

```
//Chapter 4, Example 4.3, page 136
clc
//Initialisation
fc=6.3*10**6                              //frequency
   in hertz

//Calculation
f=fc*sqrt(2)                          //maximum usable
   frequency

//Results
printf("Maximum usable frequency = %.3f MHz",(f
   *10**-6))
```

Scilab code Exa 4.4 Example 4

```
//Chapter 4, Example 4.4, page 137
clc
//Initialisation
tec=10**17                                   //total
    electron content
H=200*10**3                                  //
   thickness of F layer

//Calculation
pd=tec/H                                      //
   plasma density at F
```

```
fc=9*sqrt(pd)
ouf=3.6*fc*0.8                                //optimum usable frequency
muf=fc*1.788                                  //maximum usable frequency

//Results
printf("Maximum usable frequency = %.1f MHz",(fc*10**-6))
printf("\nOptimum usable frequency < %.3f MHz",(ouf*10**-6))
printf("\nMaximum usable frequency (30) = %.1f MHz",(muf*10**-6))
```

Scilab code Exa 4.5 Example 5

```
//Chapter 4, Example 4.5, page 138
clc
//Initialisation
d=1000                                        //distance in km
re=6370                                       //earth radius in km
dt=30                                         //in degree
pi=3.14

//Calculation
teta=d*(2*re)**-1                             //theta in radians
tet=teta*180*pi**-1                           //theta in degree
w1=90-dt-tet
a=sin(w1*3.14/180)/re
a1=sin((90+dt)*3.14/180)
```

```
h=(a1*a**-1)-re                                          //
    height

//Results
printf("(1) virtual height of the reflection point =
    %d km",h)            //The answer provided in the
    textbook is wrong
```

Scilab code Exa 4.6 Example 6

```
//Chapter 4, Example 4.6, page 142
clc

//Initialisation
d=200                                    //height in Km
f=700                                    //frequency in Khz

//Calculation
T1e=0.4                                  //from graph 4.10
T2e=0.9
T3e=1.7
T1f=1.3
T2f=2.8
T3f=4.3
Tef=0.3
Tef1=2.7
Tef2=0.5

//Results
printf("(1) Time delay of E layer, Td(1E) = %.1f ms"
    ,T1e)
printf("\n                              Td(2E) = %.1f
    ms",T2e)
printf("\n                              Td(2E) = %.1f
    ms",T3e)
```

```
printf("\n     Time delay of F layer , Td(1F) = %.1f
    ms",T1f)
printf("\n                              Td(2F) = %.1f
    ms",T2f)
printf("\n                              Td(2F) = %.1f
    ms",T3f)
printf("\n(2) Time delay of E and F for a distance
    of 500 km, Td(E,F) = %.1f ms",Tef)
printf("\n(3) Td(1F,3F) = %.1f ms",Tef1)
printf("\n     Td(1E,3E) = %.1f ms",Tef2)
```

Scilab code Exa 4.7 Example 7

```
//Chapter 4, Example 4.7, page 147
clc
//Initialisation
f=1.5*10**9
    //frequency in Hz
tec=10**18
    //total electron content
g=5*10**-3
    //geomagnetic field intensity
a=3.36*10**2

//Calculation
teta= a*g*tec*(f**-2)                              //
    Faraday rotation in Radian
c=0.8422
x=20*log10(c)                                  //loss
    value in dB

//Results
printf("(1) Faraday rotation = %.1f Rad",teta)
printf("\n(2) Loss = %f dB",x)
```

Scilab code Exa 4.8 Example 8

```
//Chapter 4, Example 4.8, page 149
clc
//Initialisation
tec1=10**18                                    //total
    electron content
f=1.5                                          //frequency in
     Hertz
tec2= 10**17                                   //total
    electron content

//Calculation
teta = 600                                     //Faraday
    rotation in mRadian
T=5                                            //time delay
    in ns
gd=0.5                                         //time delay
    difference in ns

//Results
printf("(1) Faraday rotation = %d mRad",teta)
printf("\n(2) Time delay = %d ns",T)
printf("\n(3) G/D = %.1f ns",gd)
```

Scilab code Exa 4.9 Example 9

```
//Chapter 4, Example 4.9, page 158
clc

//Initialisation
```

```
phi=166        //in radian
pi=3.14
t=35*pi/180                    //geographic latitude in radian
t1=60*pi/180                   //zenith angle in radian
N=80*pi/180                    //in radian
x=92
y=35
h=35
p=1.2



//Calculation
m=0.11-0.49*cos(t)
fe=0.004*(1+0.021*166)**2          //minimum value of f0E
A=1+0.0094*(phi-66)                //A value
B=(cos(N))
B1=B**m                            //B value
C=x+y*cos(t)                       //C value
D=cos(t1)**p                       //D value
F=(A*B*C*D)**(0.25)                //exact value of f0E

//Results
printf("(1) Minimum value of f0E = %.2f x 10^-2 MHz",(fe*100))
printf("\n(2) f0E = %.2f MHz",F)
```

```
                                                //The answer
    provided in the textbook is wrong
```

Scilab code Exa 4.10 `Example 10`

```
//Chapter 4, Example 4.10, page 159
clc

//Initialisation
g=50                                                          //
    geomagnetic latitude in degree
R12=100                                                       //
    solar activity number
pi=3.14
x=60*pi/180                                              //zenith
    angle in radians

//Calculation
f0=4.35+0.0058*g-0.00012*g**2
f100=5.35+0.011*g-0.00023*g**2
fs=f0+0.01*(f100-f0)*R12
n=0.093+(0.00461*g)-(0.000054*(g**2))+(0.0031*R12)
                    //The answer provided in the textbook
    is wrong
F1=fs*(cos(x))**n                                            //
    critical frequency

//Results
printf("Critical Frequency = %.2f MHz",F1)
    //The answer provided in the textbook is wrong
```

Scilab code Exa 4.11 `Example 11`

```
//Chapter 4, Example 4.11, page 164
clc
//Initialisation
R12=150                                              //
    12 month average value
fs0=4.416
fs100=5.473
n=0.23
pi=3.14
x=45*pi/180                                     //zenith
     angle in radians

//Calculation
f1=63.7+0.728*R12+0.00089*R12**2
fs=fs0+0.01*(fs100-fs0)*R12
F1=fs*(cos(x))**n                                   //
    critical frequency

//Results
printf("(1) R12 = %d",R12)
printf("\n(2) F12 = %d",f1)
printf("\n(3) f0F1 = %.2f MHz",F1)
```

Chapter 5

Propagation in 3 KHz to 30 MHz Band

Scilab code Exa 5.2 Example 2

```
//Chapter 5, Example 5.2, page 186
clc

//Initialisation
f=5                                          //frequency in Hz
er=15                                        //ground characteristics
s=0.01                                       //for vertically polarized waves
c=3*10**8                                    //speed of light
e0=8.85*10**-12                              //permitivity of free space
d=80000                                      //distance in m
pi=3.14

//Calculation
```

```
a=5**0.333
df=50/a                                        //distance
   in metre
h=c*(f*10**6)**-1                              //wavelength
b=s/(2*pi*f*e0*10**6)
b1=sqrt(er**2+b**2)
p=(pi*d)/(h*b1)

//from fig 5.8
As = 0.05                                     //attenuation
   factor

//Results
printf("p = %d",p)
printf("\n|As| = %.2f",As)
```

Scilab code Exa 5.3 Example 3

```

//Chapter 5, Example 5.3, page 191
clc
//Initialisation
c=3*10**8                                      //speed of
   light
f=10*10**6                                     //frequency
   in Hz
e0=8.85*10**-12                                //permitivity
    of free space
er=10                                          //ground
   characteristics
s=0.005
d=30000
pt=200                                         //
   transmitter power in watt
gt=1                                            //gain of
```

```
transmitter antenna
gr=1                                          //gain of receiver antenna
pi=3.14                                       //pi

//Calculation
h=c*f**-1                                     //wavelength
e=er*e0                                       //epsilon
b=s/(2*pi*f*e)
b1=sqrt(er**2+b**2)
p=(pi*d)/(h*b1)                               //The answer provided in the textbook is wrong
i=((er*e0*2*3.14*f)/s)
b2=atan(i)
b3=b2*180/pi
a1=((2+0.3*p)/(2+p+0.6*p**2))
a2=sqrt(p/2)*(5*10**-82)*sin(-b3)
As=a1-a2                                      //attenuation function
pr=pt*gt*gr*h**2/(4*pi*d)**2
pr1=pr*(2*As)**2                              //The answer provided in the textbook is wrong

//Results
printf("Received signal power Pr = %.2f pW",(pr1*10**12))          //The answer provided in the textbook is wrong
```

Scilab code Exa 5.4 Example 4

```
//Chapter 5, Example 5.4, page 192
clc
//Initialisation
f=0.5                                         //frequency in MHz
```

```
Pa=100                              //transmitter power
Po=1000
e120=68                             //from figure 5.10
e220=85                            //from figure 5.9
e230=80
e330=60                             //from figure 5.10
e380=48
e350=50                            //from figure 5.10
e250=75                            //from figure 5.9
e260=73
e160=60                            //from figure 5.10
e180=48

//Calculation
ETR=e120-e220+e230-e330+e380
ERT=e350-e250+e260-e160+e180  //The answer provided in the textbook is wrong
ER=(ETR+ERT)/2  //The answer provided in the textbook is wrong
Ea=ER+(10*log10(Pa*Po**-1))
lb=137.2+(20*log10(f))-ER

//Results
printf("(1) Electric field = %.1f dB",Ea)              //The answer provided in the textbook is wrong
printf("\n(2) Basic loss path = %.1f dB",lb)             //The answer provided in the textbook is wrong
```

Scilab code Exa 5.5 Example 5

```
//Chapter 5, Example 5.5, page 196
clc
//Initialisation
f1=2.5                                          //frequency
    in MHz
f2=6.3                                          //frequency
    in MHz
K=1.1                                           // K factor

//Calculation
fse=1.05*f1*2                                            //
    frequency in MHz
fsf=K*f2*2                                             //
    frequency in MHz

//Results
printf("Frequency for E layer = %.2f MHz",fse)
printf("\nFrequency for F layer = %.2f MHz",fsf)
```

Scilab code Exa 5.7 Example 7

```
//Chapter 5, Example 5.7, page 201
clc

//Initialisation
f=10                                            //frequency
    in MHz
delta=14.5                                      //in degree
d=1750                                          //distance
    in Km
re=6370                                         //radius of
    earth in Km
pt=100                                          //transmitter
```

```
power in watt
lm=30                                      //in dB
P11=3775                                 //in Km

//Calculation
a=(delta+(d/(2*re)))*(180*3.14**-1)
j=cos(a)
a1=(d*(2*re)**-1)*(180*3.14**-1)
j1=sin(a1)
P=4*re*(j1*j**-1)                          //path length
pt1=10*log10(pt*10**-3)
FSL=32.4+20*log10(f)+20*log10(3775)        //free space loss
Et=136.6+pt1+20*log10(f)-FSL-lm            //median value

//Results
printf("(1) Path length = %d km",P11)
printf("\n(2) Median value = %.2f dB",Et)
```

Scilab code Exa 5.8 Example 8

```
//Chapter 5, Example 5.8, page 202
clc

//Initialisation
et=20                                      //in dB
gr=2                                       //antenna gain in dB
f=15                                       //frequency in MHz


//Calculation
```

```
pr=et+gr-(20*log10(f))-107.2           //received signal power in dB
pr1=10**(pr/10)                                //received signal power in W

//Results
printf("Power Recieved signal = %.2f pW",(pr1*10**12))
```

Scilab code Exa 5.9 Example 9

```
//Chapter 5, Example 5.9, page 202
clc
//Initialisation
pr=-108.7                                      //received signal power in dB
fa=50                                          //noise tempreture
b=2700                                         //frequency in Hz
N=5                                            //noise figure in dB

//Calculation
snr=pr-fa-(10*log10(b))+204                   //signal to noise ratio
snr1=snr-N

//Results
printf("Received signal to noise ratio = %.1f dB",snr)
printf("\nOutput signal to noise ratio = %.1f dB",snr1)
```

Scilab code Exa 5.10 Example 10

```
//Chapter 5, Example 5.10, page 205
clc
//Initialisation
d=3000                                  //distance in Km
re=6370                                 //radius of earth in Km
phi=72                                  //angle in degree
N=5*10**11                              //electron density
pi=3.14

//Calculation
teta=3000*(2*6370)**-1                  //in radian
teta1=teta*180/pi                   //degree
dt=90-teta1-phi                         //Elevation angle
a=re/(sin(phi*pi/180))
b=sin((teta1+phi)*pi/180)
h=(a*b)-re                               //Height in Km
fc=9*sqrt(N)                        //frequency in MHz
MUF=fc*(cos(phi*pi/180))**-1        //Maximum working frequency

//Results
printf("(1) Elevation angle = %.1f degree",dt)
printf("\n(2) Height h = %.1f km",h)
printf("\n(3) MUF = %.1f MHz",(MUF*10**-6))
```

Scilab code Exa 5.11 Example 11

```
//Chapter 5, Example 5.11, page 208
clc

//Initialisation
d=2500                                          //distance in Km
re=6370                                         //radius of earth in Km
dt=6                                            //elevation angle in degree
f1=15                                           //frequency in MHz
los1=42                                         //loss
pi=3.14

//Calculation
teta=d*(2*re)**-1                        //in radian
teta1=teta*180*pi**-1                   //in degree
phi=90-dt-teta1
l=(2*re*sin(teta))/sin(phi*pi/180)
fsl=32.4+(20*log10(f1))+(20*log10(l))         //Free space loss
pr=57+6-fsl-los1                                //receving power in dB
pr1=10**(pr/10)                                 //receving power in Watt

//Results
printf("Power = %.2f pW",(pr1*10**12))
```

Chapter 6

Terrestrial Mobile Radio Propagation

Scilab code Exa 6.1 `Example 1`

```
//Chapter 6, Example 6.1, page 186
clc

//Initialisation
c=3*10**8                    //speed of light
f=400*10**6                  //frequency in Hz
l1=15*10**3                  //distance in m
l2=15*10**3                  //distance in m
l=30*10**3                   //distance in m
k=1.33                       //k factor
d1=15                        //distance in Km
d2=15                        //
```

```
distance in Km
re=6370                                     //distance in Km

//Calculation
h=c*f**-1                                   //wavelength in m
r1=sqrt(l1*l2*h/l)                      //Fresnel radius
ho=(500*d1*d2)/(k*re)                       //Earth bulge

//Results
printf("(1) Fresnel radius, r1 = %d m",r1)
printf("\n(2) h0 = %.2f m",ho)
```

Scilab code Exa 6.2 Example 2

```
//Chapter 6, Example 6.2, page 223
clc
//Initialisation
f=400                                        //frequency in MHz
k=1.33                                        //k factor
er=3                                          //dielectric conductivity
sg=10**-4                                     //Earth effective conductivity
eo=8.85*10**-12                              //permittivity of free space
re1=8500                                      //Effective Earth radius in Km
c=3*10**8                                     //speed of light
```

```
B=1
d=50


//Calculation
kh=1.6*10**-3                                          // horizontal polarization using Fig. 6.2
kv=5*10**-3                                            // vertical polarization using Fig. 6.2
X=2.2*B*f**(1*3**-1)*re1**(-2*3**-1)*d                 // normalized length of the path
FX=11+10*log10(X)-17.6*X                                // distance attenuation value

//Results
printf("(1) Kh = %.1f x 10**-3",(kh*10**3))
printf("\n    Kv = %.1f x 10**-3",(kv*10**3))
printf("\n(2) F(X) = %.2f dB",FX)
```

Scilab code Exa 6.3 Example 3

```
//Chapter 6, Example 6.3, page 228
clc

//Initialisation
f=300*10**6                                 //frequency in Hz
l1=4*10**3                                  //distance in m
l2=6*10**3                                  //distance in m
h1=20                                       //height in m
c=3*10**8                                   //speed of light
```

```
d1=4                                            //distance in km
d2=6                                            //distance in km
R=10                                            //radius in km
m=0.13
n=1.99


//Calculation
h=c*f**-1                                       //wavelength
l=l1+l2
r1=sqrt(l1*l2*h/l)
rat1=h1/r1                                      //ratio
a=sqrt((2*(d1+d2))/(h*d1*d2))
v=0.0316*h1*a
jv=6.9+20*log10(1.585)                          //knife-edge obstacle loss
k=8.2+12*n
Tmn=k*m
A=jv+Tmn                                        //rounded obstacle loss

//Results
printf("(1) Ratio = %f",rat1)
printf("\n(2) Loss J(v) = %.1f dB",jv)
printf("\n(3) Loss A = %.2f dB",A)
```

Scilab code Exa 6.4 Example 4

```
//Chapter 6, Example 6.4, page 233
clc

```

```
//Initialisation
f=150*10**6                                  //frequency in Hz
c1=3*10**8                                   //speed of light
h11=60                                       //in metre
d11=2000                                       //in metre
d1=259.6                                      //in metre
b=2000                                       //in metre
a=250                                       //in metre
h21=80                                       //in metre
d21=7259                                      //in metre
c=7250                                       //in metre

//Calculation
h=c1*f**-1                                                 //wavelength
v1=h11*sqrt((2*(h*d1)**-1)+(1*d11**-1))
L1=6.9+20*log10(sqrt((v1-0.1)**2+1)+v1-0.1)       //path diffraction loss
v2=h21*sqrt((2*(h*d11)**-1)+(1*d21**-1))
L2=6.9+20*log10(sqrt((v2-0.1)**2+1)+v2-0.1)       //path diffraction loss

//Results
printf("Diffraction loss L1 = %.2f dB",L1)
printf("\n                 L2 = %.2f dB",L2)
```

Scilab code Exa 6.5 `Example 5`

```
//Chapter 6, Example 6.5, page 239
clc

//Initialisation
```

```
f=450*10**6                                          //frequency in Hz
q1=1.282                                             //cumulative distribution value
q2=1.645                                             //cumulative distribution value

//Calculation
sg=3.8+1.6*log10(450)                           //standard deviation
fm1=q1*sg                                            //fade margin
fm2=q2*sg                                           //fade margin
fm=fm2-fm1                                          //gain


//Results
printf("Antenna gain = %.2f dB",fm)
```

Scilab code Exa 6.6 Example 6

```
//Chapter 6, Example 6.6, page 240
clc

//Initialisation
q90=1.282                                   //cumulative distribution value of 90%
sl=8                                        //standard deviation
q97=1.881                                   //cumulative distribution value of 97%
pt=5                                        //transmitter power

```

```
//Calculation
fm=q90*sl                                    //fade margin
fm1=q97*sl                                   //fade margin
p=fm1-fm                                     //power in dB
p1=pt*10**(p/10)                             //power in watt

//Results
printf("(1) Fade margin for received signal = %.3f dB",fm)
printf("\n(2) New transmitter power = %d W",p1)
```

Scilab code Exa 6.7 Example 7

```
//Chapter 6, Example 6.7, page 241
clc
//Initialisation
d= 50*10**3                                  //distance in m

//Calculation
sl1=5.3                                      //location standard deviation
st1=3                                        //time standard deviation
sl2=6.2                                      //location standard deviation
st2=2                                        //time standard deviation
sv=sqrt(sl1**2+st1**2)                       //total standard deviation of VHF
su=sqrt(sl2**2+st2**2)                       //total standard deviation of UHF


```

```
//Results
printf("(1)for VHF, sigmaL = %.1f dB ",sl1)
printf("\n            sigmaT = %.1f dB ",st1)
printf("\n   for UHF, sigmaL = %.1f dB ",sl2)
printf("\n            sigmaT = %.1f dB ",st2)
printf("\n(3) Standard deviation values, sigmaVHF =
    %.1f dB",sv)
printf("\n                                sigmaUHF =
    %.1f dB",su)
```

Scilab code Exa 6.9 `Example 9`

```
//Chapter 6, Example 6.9, page 245
clc

//Initialisation
d=5                                        //in dB
h=20                                       //
   Transmitter initial height

//Calculation
ht=h*10**(0.25)                            //
   Transmitter ultimate antenna height

//Results
printf("(1) Antenna Height = %.2f m",round(ht))
```

Scilab code Exa 6.10 `Example 10`

```
//Chapter 6, Example 6.10, page 246
clc
//Initialisation
```

```
f=1800*10**6                                    //frequency in Hz
c=3*10**8                                       //speed of light

//Calculation
h=c*f**-1                                       //wavelength
hv=20*h                                         //in metre
dh=10*h                                         //in metre

//Results
printf("hv = %.2f m ",hv)
printf("\ndh = %.2f m ",dh)
```

Scilab code Exa 6.11 Example 11

```
//Chapter 6, Example 6.11, page 262
clc
//Initialisation
p1=20                                           //transmitter power
g=6                                             //gain
h1=20                                           //height in metre

//Calculation
ct=p1/10                                        //Power gain
ch=(h1*30**-1)**2                               //height gain
cg=g*4**-1                                      //antenna gain
co=10*log10(ct*ch*cg)                           //Total effects
```

```
//Results
printf("(1) Power gain, Ct = %.f",ct)
printf("\n    Height gain = %.2f",ch)
printf("\n    Antenna gain = %.1f",cg)
printf("\n(2) Total effects = %.2f dB",co)
```

Scilab code Exa 6.12 Example 12

```
//Chapter 6, Example 6.12, page 262
clc
//Initialisation
g1=10                                        //transmitter gain
ct=15                                        //power in watt


//Calculation
g2=g1-2.2                                    //gain in dBd
cg=g2-6                                      //Antenna gain
ct1=ct*10**-1
ct2=10*log10(ct1)                        //Power gain
ch=(ct*30**-1)**2
ch1=10*log10(ch)                          //Height gain
ct3=ct1*0.5
ct4=10*log10(ct3)
co=ct4+cg+ch1                                  //Total effects

//Results
printf("(1) Power gain, Ct = %.2f",ct2)
printf("\n    Height gain = %.2f",ch1)
```

```
printf("\n     Antenna gain = %.1f",cg)
printf("\n(2) Total effects = %.2f dB",co)
```

Scilab code Exa 6.13 Example 13

```
//Chapter 6, Example 6.13, page 265
clc
//Initialisation
sr=-106                        //Receiver sensitivity
f=8                            //Fade margin
cl=6                           //Coupler loss
dl=1                           //Duplexer Loss
bf=6.5                         //BTS feeder loss
ba=12                          //BTS antenna gain
pl=138                         //Path loss
pg=15                          //Pathlength in km
ta=2                           //Terminal antenna gain
tf=0.5                         //Terminal feeder loss

//Calculation
prm=sr+f                       //minimum received power
ptb=prm+cl+dl+bf-ba+pl-ta+tf   //BTS transmitter power in dbBm
pw=10**((ptb-30)/10)
```

```
//Results
printf("BTS transmitter power = %.2f dBm",ptb)
printf("\n                        = %d W",pw)
```

Scilab code Exa 6.14 Example 14

```
//Chapter 6, Example 6.14, page 265
clc
//Initialisation
pm=2                                   //transmitter power
ld=1                                   //Duplexer losses
lp=138                                 //Path loss
lfm=0.5                                //terminal feeder losses
lfb=6.5                                //transmitter feeder losses
gt=12                                  //BTS transmitter antenna gain
gr=2                                   //BTS receiver antenna gain
i=3
bs=-110                                //BTS receiver sensitivity

//Calculation
ptm=10*log10(pm*10**3)
prb=ptm-ld-lp-lfm-lfb+gt+gr
pr=prb+i                               //BTS received power
fm=pr-bs                               //fade margin

```

```
//Results
printf("BTS received power = %.1f dBm",pr)
printf("\nFade margin = %.1f dB",fm)
```

Scilab code Exa 6.15 Example 15

```
//Chapter 6, Example 6.15, page 265
clc
//Initialisation
t1=25                                        //terminal transmitter power
t2=2                                         //terminal transmitter power
gd=3                                         //correction factor of receiver antennas
lc=5                                         //coupler loss
prm=-105                                     //receiver sensitivity
prb=-110                                     //receiver sensitivity

//Calculation
ptb=10*log10(t1*10**3)
ptm=10*log10(t2*10**3)
p=ptb-ptm                                     //Transmitting gain in downlink
ga=prm-prb                                   //Receiving gain in uplink
tg=gd+ga+lc                                   //total gain on the uplink


//Results
printf("Transmitting gain in downlink = %.1f dBm",p)
printf("\nReceiving gain in uplink = %.1f dBm",ga)
```

```
printf("\ntotal gain on the uplink = %.1f dBm",tg)
```

Scilab code Exa 6.16 Example 16

```
//Chapter 6, Example 6.16, page 269
clc

//Initialisation
f=450
    //frequency in MHz
d=25
    //distance in m
hb=30
hm=5

//Calculation
fsl=32.4+(20*log10(f))+(20*log10(d))            //
    free space loss
lp=120+(40*log10(d))-(20*log10(hb))-(20*log10(hm))
      //path loss
lm=76.3-10*log10(hm)
l=(40*log10(25))+(20*log10(f))-(20*log10(hb))+lm
    //path loss based on the clutter factor model


//Results
printf("(1) Free space loss = %.1f dB",fsl)
printf("\n(2) Loss = %.1f dB",lp)
printf("\n(3) Loss based on clutter factor = %.1f dB
    ",l)
```

Scilab code Exa 6.17 Example 17

```
//Chapter 6, Example 6.17, page 271
clc


//Initialisation
pt=30                                               //transmitter power in watt
d=15                                                //distance in km
gt=3                                                //transmitter gain
ht=30                                               //transmitter height in m
hr=4                                                //receiver height in m
no=3.77*10**14

//Calculation
gt1=10**(gt*10**-1)
pt1=gt1*pt
e=88*sqrt(pt1)*pt*hr/(2*d**2)                       //Field strength
pr1=(e**2)/(2*no)                                   //Recieved power

//Results
printf("Field strength = %f V/m",e)
printf("\nRecieved power = %.2f pW",(pr1*10**12))
```

Scilab code Exa 6.18 Example 18

```
//Chapter 6, Example 6.18, page 274
clc
```

```

//Initialisation
f=420                                        //frequency in Hz
h1=40                                        //height in m
h2=5                                         //height in m
d=15                                         //distance in km

//Calculation
A=69.55+26.16*log10(f)-13.82*log10(h1)       //Hata parameters
B=44.9-6.55*log10(h1)
C=2*(log10(f*28**-1))**2+5.4
D=4.78*(log10(420))**2-18.33*log10(f)+40.94
E1=3.2*(log10(11.75*h2))**2-4.97
E2=(((1.1*log10(f))-0.7)*h2)-((1.56*log10(f))-0.8)
L3=A+B*log10(d)-D                            //in open area;
L2=A+B*log10(d)-C                            //in suburban area;
L1=A+B*log10(d)-E1                           //in large cities;
L11=A+B*log10(d)-E2                          //in small cities;

//Results
printf("In large cities L1 = %.2f dB",L1)
printf("\nIn small cities L1 = %.2f dB",L11)
printf("\nIn suburban area L2 = %.2f dB",L2)
printf("\nIn open area L2 = %.2f dB",L3)
```

Scilab code Exa 6.19 Example 19

```
//Chapter 6, Example 6.19, page 275
clc

//Initialisation
f=1800
    //frequency in MHz
d=10
    //distance in m
hb=40
hm=3
A=132.57
    //Hata model data
B=34.4
    //Hata model data

//Calculation
E2=(((1.1*log10(f))-0.7)*hm)-((1.56*log10(f))-0.8)
lp=46.3+33.9*log10(f)-13.82*log10(hb)+(44.9-6.55*
    log10(hb)-E2+hm)
L=A+B+-E2


//Results
printf("Path loss based on COST Hata model,")
printf("\n          Lp = %.2f dB",lp)
printf("\nPath loss based on Hata model,")
printf("\n          Lp = %.2f dB",L)
```

Scilab code Exa 6.20 `Example 20`

```
//Chapter 6, Example 6.20, page 277
```

```
clc

//Initialisation
pt=20                                  //transmitter power
    in watt
Hb=30                                  //in metre
Hm=3                                  //in metre
gt=14.2                                 //trasmitter gain
    in dB
gr=0.2                                  //receiver gain in
     dB
f=450                                   //frequency in MHz
gm=-2                                  //in dBd
gr2=-2.2                               //in dBi
r1=10
n=20
hb=10
hm=10


//Calculation
gt1=gt+gr2
pr1=-62-38*log10(r1)-20*log10(f*900**-1)+7
    //received signal level in suburban
pr2=-64-43*log10(r1)-20*log10(f*900**-1)+7
    //received signal level in urban
ao=10*log10(2)+(gr2-6)
                                               //in dB (The
    answer provided in the textbook is wrong)
pr11=-62-38*log10(r1)-20*log10(f*900**-1)+ao
    //received signal level in rural
pr22=-64-43*log10(r1)-20*log10(f*900**-1)+ao
    //received signal level in cities
ptd=10*log10(pt*10**3)
                                               //in dBm
lp1=ptd-pr11

    //Path loss in rural area
```

```
lp2=ptd-pr22                                    //Path loss in cities area

//Results
printf("(2) In the suburban area, Pr = %.1f dBm",pr1)
printf("\n    In the urban area, Pr = %.1f dBm",pr2)
printf("\n(3) Path loss in rural area Lp = %.1f dB",lp1)          //The answer provided in the textbook is wrong
printf("\n    Path loss in cities area Lp = %.1f dB",lp2)          //The answer provided in the textbook is wrong
```

Chapter 7

Line of Sight Propagation

Scilab code Exa 7.1 Example 1

```
//Chapter 7, Example 7.1, page 293
clc

//Initialisation
h=200                                   //height in m
d=30*10**3                              //distance in km
R=40*10**-6                             //height in km
pi=3.14

//Calculation
phi=atan(h*d**-1)                  //in radian
phi1=phi*180/pi                    //in degree
n=cos(phi1)
r=round(n)/R                            //radius

//Results
printf("Radius = %.1f km",r)
```

Scilab code Exa 7.2 Example 2

```
//Chapter 7, Example 7.2, page 294
clc

//Initialisation
h=500                                   //height in m
a=0.000315
b=0.0001361
Re=6370000                              //radius of
    earth in m


//Calculation
n=1+(a*exp(-b*h))
n1=(n-1)*10**6                            //Refraction
    index
c=(a*b*exp(-b*h))
R=1/c                                     //Radius of
    path curvature in km
d=1-(Re/R)
K=1/d                                    //K-factor

//Results
printf("(1) Refraction index = %d ",n1)
printf("\n(2) Radius of path curvature = %d kM",(R
    /10**3))
printf("\n(3) K-factor = %.3f",K)
```

Scilab code Exa 7.3 Example 3

```
//Chapter 7, Example 7.3, page 299
clc
//Initialisation
k1=1.3                                    //K-factor
```

```
k2=0.7                                   //K-factor
H1=1200                                  //sea level in m
H2=1400                                  //sea level in m
re=6370*10**3                            //radius of earth in m
f=15*10**9                               //frequency in Hz
a1=0.6
d1=15*10**3                              //distance in m
d2=20*10**3                              //distance in m
c=3*10**8                                //speed of light
d=30*10**3                               //distance in m
h2=25.24
r2=11.55
a2=0.3


//Calculation
h1=500*d1*d2/(k1*re)
h=c*f**-1
r1=sqrt((d1*d2*h)/d)
H11=H1+h1+a1*r1                        //in meter
H22=H2+h2+a2*r2                        //in meter

//Results
printf("(1) H1 = %d m",H11)                //The answer provided in the textbook is wrong
printf("\n(2) H2 = %d m",H22)
```

Scilab code Exa 7.4 Example 4

```
//Chapter 7, Example 7.4, page 302
clc

//Initialisation
hr=-1                          //obstacle height and
   fresnel radius ratio

//Calculation
Ad=-20*hr+10                   //Diffraction loss in
   dB
Ad2=16                         //from fig 7.8
Ad3=43

//Results
printf("(1) Obstacle loss = %.1f dB",Ad)
printf("\n(2) Knife edge obstacle = %.1f dB",Ad2)
printf("\n(3) Rounded obstacle = %.1f dB",Ad3)
```

Scilab code Exa 7.5 Example 5

```
//Chapter 7, Example 7.5, page 306
clc
//Initialisation
d1=10
   //distance in km
d2=25
   //distance in km
re=6370                                                   //
   earth radius in km
k=0.5
   //climatic factor
f=4*10**9
   //frequency in Hz
```

```
c=3*10**8
    //speed of light
d=35
    //distance in km
h3=400
    //height in m

//Calculation
ho=(500*d1*d2)/(k*re)
                                        //Earth buldge in
    m
h=c*f**-1
                                                        //
    wavelength in m
r1=sqrt(d1*10**3*d2*10**3*h*(d*10**3)**-1)          //
    fresnel radius
amsl=h3+ho+13.9
                                                //AMSL

//Results
printf("(1) Earth buldge = %.2f m",ho)
printf("\n(2) Fresnel radius = %.1f m",r1)
printf("\n(3) AMSL = %.1f m",amsl)
```

Scilab code Exa 7.6 Example 6

```
//Chapter 7, Example 7.6, page 309
clc
//Initialisation
pt=500                                          //
    Transmitter power in mW
gt=42                                           //
    transmitter antenna gain in dB
gr=44                                           //
    receiver antenna gain in dB
```

```
lbt=2.6                                        //
   transmitter branching loss in dB
lbr=3                                          //
   receiver branching loss in dB
flt=45                                         //
   transmitter feeder length
flr=35                                        //
   receiver feeder length
fls=6.5                                        //
   feeder loss
prx=-72                                        //
   Receiver sensitivity
d=30                                           //path
    distance in km
f=8.4                                          //
   frequency in ghz

//Calculation
ptx=10*log10(pt)                           //
   transmitter threshold level
lft1=flt*fls/100
lfr=flr*fls/100
eirp=ptx+gt-lbt-lft1                             //
   Transmitter effective power
sg=ptx-prx
fsl=92.4+20*log10(f)+20*log10(d)
rsl=ptx+gt+gr-fsl-lft1-lfr-lbt-lbr
fm=rsl-prx

//Results
printf("EIRP = %.2f dBm",eirp)
printf("\nSG = %.2f dBm",sg)
printf("\nFSL = %.2f dB",fsl)
printf("\nRSL = %.2f dBm",rsl)
printf("\nFM = %.2f dB",fm)
```

Scilab code Exa 7.7 Example 7

```
//Chapter 7, Example 7.7, page 315
clc

//Initialisation
f=15                                          //
   frequency in GHz
f1=18                                         //
   frequency in GHz
R=50                                          //rain
    intensity
ah=1.154
kh=0.0367
d=20                                          //
   distance in kM

//Calculation
yr=kh*R**ah
do=35*exp(-0.015*R)                       //distance
    in kM
de=d/(1+(d/do))                               //
   distance in kM
Ao=yr*de                                      //
   Rain Loss in dB
phi=(f**2)/(1+10**-4*f**2)
phi1=(f1**2)/(1+10**-4*f1**2)
H=1.12*10**-3*((phi1/phi)**0.5)*(phi*Ao)**0.55
Ah=Ao*(phi1/phi)**(1-H)
Av=(300*Ah)/(335+Ah)                          //
   Rain Loss in dB


//Results
```

```
printf("(1) Rain Loss, A = %.2f dB",Ao)
printf("\n(2) rain loss for vertical polarization,
    Av = %.2f dB",Av)
```

Scilab code Exa 7.8 Example 8

```
//Chapter 7, Example 7.8, page 322
clc

//Initialisation
A=99.8                                          //in
    percent
l=1250                                          //radio
    link in km
C=155*10**6                                     //in bps
T=24*60*60                                      //Total
    measurement time

//Calculation
U=100-A
u=U/100
ue=u*l/2500
uep=ue*0.3                                  //
    propagation unavailability value
M=C*uep*T*10**-3                            //number of
    errored bits due to propagation

//Results
printf("Maximum delay bit error per day = %d bits
    per day",M)
```

Scilab code Exa 7.9 Example 9

```
//Chapter 7, Example 7.9, page 324
clc

//Initialisation
h=24                                    //hours
m=60                                    //minutes
s=60                                    //seconds


//Calculation
dm=0.004*h*m*(1250*2500**-1)            //Maximum
    degraded minutes
ses=0.00054*h*m*s*(1250*2500**-1)       //Severely
    errored seconds

//Results
printf("(1) Maximum degraded minutes per day = %.2f
    min",dm)
printf("\n(2) Severely errored seconds per day = %.2
    f s",ses)
```

Scilab code Exa 7.10 Example 10

```
//Chapter 7, Example 7.10, page 331
clc
//Initialisation
Aa=15                                         //
    Transmitter antenna discrimination
Ab=25                                         //Receiver
    antenna discrimination
AD=110                                       //path in km
s=35                                         //fading in
    dB

//Calculation
```

```
CD=30                                              //path
    in km
Ad=20*log10(AD*CD**-1)                       //Distance
    discrimination

si=Aa+Ab+Ad                                      //in dB
si2=si-s                                          //in dB

//Results
printf("(1) S/I = %.1f dB",si)
printf("\n(2) S/I = %.1f dB",si2)
```

Scilab code Exa 7.11 Example 11

```
//Chapter 7, Example 7.11, page 333
clc

//Initialisation
kq=2.6*10**-6                                         //
    geoclimatic coefficient
f=6                                                   //
    frequency in GHz
d=45                                                  //
    distance in Km
gc=0.098                                              //GC
    factor
ab=0.25                                               //
    geoclimatic factor

//Calculation
po=kq*f*gc*d**3                                      //In
    country
po1=0.3*ab*(f*4**-1)*(d*50**-1)**3                   //In
    mountainous area

```

```
//Results
printf("Fading occurrence probability")
printf("\n(1) In country = %.2f",po)
printf("\n(2) In mountainous area = %.3f",po1)
```

Scilab code Exa 7.12 `Example 12`

```
//Chapter 7, Example 7.12, page 340
clc

//Initialisation
dn=70                                          //dN = 70
d1=1000                                        //height from sea level in m
d2=1400                                        //height from sea level in m
d=45                                           //radio link distance in km

//Calculation
K=10**(-4.2-0.0029*-dn)                         //Climate factor
ep=(d2-d1)/d                                    //magnitude of the path inclination
po=K*d**3*(1+ep)**(-1.2)*10**(0.033*6-1)         //Fading occurrence probability


//Results
printf("(1) Climate factor K = %.4f = 10^-4",K)
printf("\n(2) Fading occurrence probability Po = %.2f percent",po)
```

Scilab code Exa 7.13 Example 13

```
//Chapter 7, Example 7.13, page 342
clc
//Initialisation
fm=35                                   //fade margin
po=0.092                                //fading occurrence probability

//Calculation
pw=po*10**(-fm*10**-1)                  //deep fading occurrence


//Results
printf("Deep fading occurrence probability, Pw = %.1f x 10^-5",(pw*10**5))
```

Scilab code Exa 7.14 Example 14

```
//Chapter 7, Example 7.14, page 343
clc

//Initialisation
d=20                                         //distance in kM
po=0.02                                      //fading occurrence probability at 20 Km
d1=25                                        //distance in kM
d2=40                                        //distance in kM
```

```
fm1=30                                          //link in kM
fm2=35                                          //link in kM
fm3=40                                          //link in kM
tr=30*24*60


//Calculation
po1=po*(d1*d**-1)**3                 //fading occurrence probability at 25 Km
po2=po*(d2/d)**3                     //fading occurrence probability at 40 Km
pw=po*10**(-fm1*10**-1)              //fade margin at 30
pw1=po1*10**(-fm2*10**-1)            //fade margin at 35
pw2=po2*10**(-fm3/10)                //fade margin at 40
u=pw+pw1+pw2                         //total fade margin
to=u*tr                              //network outage time

//Results
printf("(1) Fading occurrence probability at 20 = %.2f",po)
printf("\n    Fading occurrence probability at 25 = %.3f",po1)
printf("\n    Fading occurrence probability at 40 = %.2f",po2)
printf("\n(2) Fade margin at 30 = %.1f x 10^-5",(pw*10**5))
printf("\n    Fade margin at 35 = %.2f x 10^-5",(pw1*10**5))
printf("\n    Fade margin at 40 = %.2f x 10^-5",(pw2*10**5))
```

```
printf("\n      Total fade margin = %.2f x 10**-5",(u
    *10**5))
printf("\n(3) Network outage time , To = %.3f min
    per month",(to))
```

Scilab code Exa 7.15 Example 15

```
//Chapter 7, Example 7.14, page 344
clc
//Initialisation
ur=0.001                    //unavailability budget
    for hypothetical circuit
d=50                        //path-length in km
A=0.25                      //area conditions
B=1                         //area conditions
f=8                         //frequency in GHz


//Calculation
pw=ur*(d*(d*d)**-1)               //deep fading
    occurrence probability
po=6*10**-7*A*B*f*(d**3)          //fading occurrence
    probability of desirable link
FM=-log10(pw/po)*10           //fade margin

//Results
printf("Fade margin = %.1f dB",FM)
```