Scilab Textbook Companion for Satellite Communication by A. K. Maini And V. Agrawal[1]

Created by
Laxman Ghanasham Sole
B.Tech
Electronics Engineering
Vishwakarma Institute of Technology
College Teacher
Prof. Vijay Mane
Cross-Checked by
K. V. P. Pradeep

May 8, 2014

[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: Satellite Communication

Author: A. K. Maini And V. Agrawal

Publisher: Wiley India Pvt. Ltd. New Delhi

Edition: 1

Year: 2010

ISBN: 978-81-265-2071-8

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 2

Satellite orbits and Trajectories

Scilab code Exa 2.1 velocity of satellite

```
// Example 2.1, page no-36
clear
clc

r1=6370        //Earth's Orbit in km
r2=630         //Height of satellite from surface in
   km
G=6.67*10^-11// Gravitational constant inNm^2/kg^2
M=5.98*10^24 //Mass of earth in kg

R=r1+r2

v=sqrt(G*M/(R*10^3))
printf("The velocity of sattelite %.2fkm/s",floor(v
   /10)*10^-2)
```

Scilab code Exa 2.2 Orbit parameters

```
// Example 2.2, page no-37
clear
clc

A=45000     //Apogee in km
P=7000      //Perigee in km
//(a)
a=(A+P)/2
//(b)
e=(A-P)/(2*a)
//(c)
e=(floor(e*100))/100
d=a*e

printf("(a)\nSemi-major axis of elliptical orbit is %d km",a)
printf("\n(b)\nEccentricity = %.2f",e)
printf("\n(c)\nThe distance between centre of earth and centre of ellipse is %d km ",d)
```

Scilab code Exa 2.3 Orbit parameters

```
// Example 2.3, page no-37
clear
clc

ma=42000   //Major axis distance in Km
P=8000     //Perigee distance in Km

A=ma-P
e=(A-P)/ma

printf("Apogee=%dkm\n Eccentricity=%.2f",A,e)
```

Scilab code Exa 2.4 Orbit parameters

```
// Example 2.4, page no-37
clear
clc

e=0.6    //Eccentricity
d=18000 //distance between earth's centre and centre
    of ellipse

a=d/e
A=a*(1+e)
P=a*(1-e)

printf("Semi-major axis of elliptical orbit is %d km
    \n Apogee distance=%dkm\n Perigee distance=%dkm",
    a,A,P)
```

Scilab code Exa 2.5 Orbit eccentricity

```
// Example 2.5, page no-38
clear
clc
AP_diff=30000    //difference between apogee and
    perigee in km
AP_sum=62800     //Apogee+perigee

E=AP_diff/AP_sum
printf("Orbit Eccentricity= %.3f",E)
```

Scilab code Exa 2.6 Satellite velocity at particular point

```
// Example 2.6 , page no-38
clear
clc

R=7000*10^3 //sattelite orbit in m
mu=39.8*10^13//constant G*M in Nm^2/kg
A=47000*10^3      //appogee distance in m
P=7000*10^3  //perigee distance in m
v=sqrt(mu/R)
a=(A+P)/2
v1=sqrt(mu*((2/R)-(1/a)))
printf("Velocity of satellite A at point X is v=%.2
   fkm/s\n Velocity of satellite B at point X is V=%
   .3fkm/s",v/1000,v1/1000)//value in book is
   different at 3rd decimal place.
```

Scilab code Exa 2.7 Satellite velocity at particular point

```
// Example 2.7 , page no-39
clear
clc

R=42000*10^3         //sattelite orbit in m
mu=39.8*10^13        //constant G*M in Nm^2/kg
A=42000*10^3         //appogee distance in m
P=7000*10^3          //perigee distance in m
v=sqrt(mu/R)
a=(A+P)/2
v1=sqrt(mu*((2/R)-(1/a)))
printf("Velocity of satellite A at point X is v=%.3
   fkm/s\n Velocity of satellite B at point X is V=%
   .3fkm/s",v/1000,v1/1000)
```

Scilab code Exa 2.8 Satellite velocity at particular point

```
// Example 2.8, page no-40
clear
clc

R=25000*10^3          //sattelite orbit in m
mu=39.8*10^13         //constant G*M in Nm^2/kg
A=43000*10^3          //appogee distance in m
P=7000*10^3           //perigee distance in m
v=sqrt(mu/R)
a=(A+P)/2
v1=sqrt(mu*((2/R)-(1/a)))
printf("Velocity of satellite A at point X is v=%.3
   fkm/s\n Velocity of satellite B at point X is V=%
   .3fkm/s",v/1000,v1/1000)//value in book is
   different at 3rd decimal place.
```

Scilab code Exa 2.9 Orbital time period

```
// Example 2.9, page no-40
clear
clc

a=(50000/2)*10^3                //Semi-major axis in m
mu=39.8*10^13         //constant G*M in Nm^2/kg

T=2*3.14*sqrt((a^3)/mu)
h=T/(60*60)
x=modulo(T,3600)
m=x/60
s=modulo(x,60)
```

```
printf("Orbital time period is given by, T = %dsec\n\t\t\t\t      = %dh %dm %ds",T,floor(h),floor(m),floor(s))//value in book is different for seconds.
```

Scilab code Exa 2.10 Ratio of orbital time period

```
// Example 2.10, page no-42
clear
clc

a1=18000*10^3              //Semi-major axis for first satellite in m
a2=24000*10^3              //Semi-major axis f0r 2nd satellite in m

T2_by_T1=(a2/a1)^(3/2)
printf("Orbital time period of sattelite 2 is %.2f times that of sattelite 1",T2_by_T1)//value in book is different for seconds.
```

Scilab code Exa 2.11 Orbit parameters

```
// Example 2.11, page no-42
clear
clc

a=25000*10^3         //appogee distance in m
b=18330*10^3         //perigee distance in m
e=(sqrt(a^2-b^2)/a)
printf("Apogee distance  = a(1+e)= %dkm\n Perigee distance = a(1-e)= %dkm\n",a*(1+e)/1000,ceil(a*(1-e)/1000))
```

Scilab code Exa 2.12 Time required to move between two points

```
// Example 2.12, page no-43
clear
clc

e=0.6                          // eccentricity of elliptical
    orbit
a=0.97                         // area of shaded region
b=2.17                         //Area of non-shaded region
t=3                           // time taken by satellite to
   move from pt B to A

x=b/a
y=x*t
printf("Time taken by satellite to move from A to B
   is %.3f hours ",y)
```

Scilab code Exa 2.13 velocity at apogee

```
// Example 2.13, page no-44
clear
clc

A=42000                 //Apogee in km
P=8000                  //Perigee in km
v_p=9.142               // velocity at perigee point
v_a=v_p*P/A
printf("Velocity at apogee = %.3f km/s",v_a)
```

Scilab code Exa 2.14 velocity of satellite at particular point

```
// Example 2.14, page no-44
clear
clc
theta=56.245     //angle made by direction of satellite with local horizontal
d=16000          //distance of particular point
P=8000           //Perigee in m
v_p=9.142        // velocity at perigee point

v=(P*v_p)/(d*floor(cos(theta*%pi/180)*1000)/1000)
printf("The velocity of satellite at that particular point is %.3f km/s",v)
```

Scilab code Exa 2.16 Apogee distance

```
// Example 2.16, page no-49
clear
clc

A1=12000                    //first Apogee distance
P=8000                      // Perigee distance
v1=1                        // assume v1 as 1
v2=1.2*v1                   //20% higher than v1

x=(v2/v1)^2
k=(((1+(P/A1))/x)-1)
k=floor(k*10^4)/10^4
A2=P/k

printf("A2 = %.0fkm",ceil(A2))
```

Scilab code Exa 2.17 velocity of satellite

```
// Example 2.17, page no-50
clear
clc
vp=8          //horizontal velocity of satellite in
   km/s
r=1620        // distance from earth's surface in km
R=6380        // Earth's radius in km
d=10000       // distance of point at which velocity
   to be calculated
theta=30      // angle made by satellite with local
   horzon at that point

P=r+R
v=(vp*P)/(d*cos(theta*%pi/180))
printf("v = %.2f km/s",v)
```

Scilab code Exa 2.18 Apogee Distance

```
// Example 2.18, page no-50
clear
clc

r=620        // distance from earth's surface in km
vp=8          //horizontal velocity of satelliteat
   9000km height in km/s
R=6380        // Earth's radius in km
d=9000       // distance of point at which velocity
   to be calculated
theta=30      // angle made by satellite with local
   horzon at that point
mu=39.8*10^13   // Nm^2/kg

P=r+R
```

```
m=vp*d*cos(theta*%pi/180)/P        //m=sqrt((2mu/P)-[2mu
    /(A+P)])
m=(m*10^3)^2
x=(2*mu/(P*10^3))-m                //x=[2mu/(A+P)]
x=floor(x/10^4)*10^4
k=(2*mu)/x                         //k=A+P
k=ceil(k/10^4)*10^4
A=k-(P*10^3)
printf("A = %.0f km",A/1000)
```

Chapter 3

Satellite Launch and In orbit Operations

Scilab code Exa 3.1 Inclination Angle

```
// Example 3.1, page no-72
clear
clc

Az=85        // Azimuth angle of injection point
l=5.2        // latitude of launch site
cosi=sin(Az*%pi/180)*cos(l*%pi/180)
i=acos(cosi)
i=i*180/%pi
printf("Inclination angle attained, i=%.1f  ",i)
```

Scilab code Exa 3.2 Velocity thrust

```
// Example 3.2, page no-73
clear
clc
```

```
delta_i=7        //orbital plane inclination
V=3000           //velocity of satellite in circularized orbit
vp=2*V*sin(delta_i*%pi/(2*180))
printf("Velocity thrust to make the inclination 0 = %.0f m/s",vp)
```

Scilab code Exa 3.3 Velocity thrust

```
// Example 3.3, page no-73
clear
clc

mu=39.8*10^13            // Nm^2/kg
P=7000*10^3              // Perigee distance in m
e=0.69                   // eccentricity of eliptical orbit
w=60/2                   // angle made by line joing centre of earth and perige e with the line of nodes
k=(e/sqrt(1+e))
k=floor(k*100)/100
v=2*(sqrt(mu/P))*k*sin(w*%pi/180)
printf("The velocity thrust required to rotate the perigee point\n by desired amount is given by, v= %.1f m/s = %.3fkm/s",v,v/1000)
```

Scilab code Exa 3.4 Velocity thrust

```
// Example 3.4, page no-74
clear
clc
```

```
A=15000*10^3                    //Original apogee distance
A1=25000*10^3                   // Raised opogee distance
P=7000*10^3                     // Perigee Distance
mu=39.8*10^13                   //Nm^2/kg

A_d=A1-A
v=sqrt((2*mu/P)-(2*mu/(A+P)))
del_v=A_d*mu/(v*(A+P)^2)
printf("required Thrust velocity Delta_v = %.1f m/s"
    ,del_v)
```

Scilab code Exa 3.5 Velocity thrust

```
// Example 3.5, page no-75
clear
clc

A=15000*10^3                    //Original apogee distance
A1=7000*10^3                   // Raised opogee distance
P=7000*10^3                     // Perigee Distance
mu=39.8*10^13                   //Nm^2/kg

A_d=A-A1
v=sqrt((2*mu/P)-(2*mu/(A+P)))
del_v=A_d*mu/(v*(A+P)^2)
printf("required Thrust velocity Delta_v = %.1f m/s"
    ,del_v)
```

Scilab code Exa 3.6 Velocity thrust

```
// Example 3.6, page no-76
clear
```

```
clc

A=15000*10^3              //Original apogee distance
A1=16000*10^3             // Raised opogee distance
P=7000*10^3               // Perigee Distance
mu=39.8*10^13             //Nm^2/kg

A_d=A1-A
v=sqrt((2*mu/P)-(2*mu/(A+P)))
v=v*P/A
del_v=A_d*mu/(v*(A+P)^2)
printf("required Thrust velocity Delta_v = %.1f m/s"
   ,del_v)
```

Scilab code Exa 3.7 Velocity thrust

```
// Example 3.7, page no-77
clear
clc

R=6378*10^3               //Radius of earth
mu=39.8*10^13             //Nm^2/kg
r1=500*10^3                // original orbit from
   earths surface
r2=800*10^3                // orbit to be raised to
   thisdistance
R1=R+r1
R2=R+r2
delta_v=sqrt(2*mu*R2/(R1*(R1+R2)))-sqrt(mu/R1)
delta_v_dash=sqrt(mu/R2)-sqrt(2*mu*R1/(R2*(R1+R2)))

printf("Two thrusts to be applied are,\n Delta_v = %
   .2f m/s \n Delta_v_dash = %.2f m/s",delta_v,
   delta_v_dash)
```

Scilab code Exa 3.8 Maximum line of sight distance

```
// Example 3.8, page no-97
clear
clc

H=36000         //Height of geostationary satellite
   from the surface of earth
R=6370          // Radius of earth in km
k=acos(R/(R+H))
//k=k*180/%pi
k=sin(k)
k=ceil(k*1000)/1000
d=2*(H+R)*k
printf("Maximum line-of-sight distance is %.2f km",d
   )
```

Scilab code Exa 3.9 line of sight distance

```
// Example 3.9, page no-98
clear
clc

H=36000         //Height of geostationary satellite
   from the surface of earth
R=6370          // Radius of earth in km
theta=20        // angular separation between two
   satellites

D=(H+R)
k=ceil(cos(theta*%pi/180)*100)/100
d=sqrt(2*D^2*(1-k))
```

```
printf("The line-of-sight distance is %.4f km",d)
```

Scilab code Exa 3.10 Inter satellite distance

```
// Example 3.9, page no-98
clear
clc

//IntelSat-VI location= 37 W
// IntelSat-VII location=74 E
theta=37+74        // angular separation between two
   satellites
D=42164            //circular equilateral
   geostationary orbit in km
k=cosd(theta)
//printf("%f\n",k)
k=-0.357952
d=sqrt(2*D^2*(1-k))

printf("Inter-satellite distance is %.2f km",d)
```

Scilab code Exa 3.11 Azimuth and elevation angle

```
// Example 3.11, page no-99
clear
clc

theta_l=30            //earth station's location 30 W
    longitude
theta_s=50            //satellite's location 50 W
   longitude
```

```
theta_L=60              //earth station's location 60 N latitude
r=42164                 // orbital radius of the satellite in km
R=6378                  //Earth's radius in km

A_dash=atan((tand(theta_s-theta_l))/sind(60))
A_dash=A_dash*180/%pi
A=180+A_dash             //Azimuth angle

x=(180/%pi)*acos(cosd(theta_s-theta_l)*cosd(theta_L))
y=r-ceil(R*(cosd(theta_s-theta_l)*cosd(theta_L)))
z=R*sind(x)
E=(atan(y/z)*180/%pi)-x
printf("Azimuth  angle =%.1f  \n Elevation angle =%.1f  ",A,E)
```

Scilab code Exa 3.12 Delay calculation

```
// Example 3.12, page no-100
clear
clc

theta_l=60              //earth station's location 60 W longitude
theta_s=105              //satellite's location 105 W longitude
theta_L=30              //earth station's location 30 N latitude

theta_l1=90              //earth station's location 90 W longitude
theta_s1=105              //satellite's location 105 W longitude
```

```
theta_L1=45           //earth station's location 45
    N latitude

c=3*10^8              //speed of light
r=42164              // orbital radius of the
   satellite in km
R=6378               //Earth's radius in km

x=(180/%pi)*acos(cosd(theta_s-theta_l)*cosd(theta_L)
   )
y=r-ceil(R*(cosd(theta_s-theta_l)*cosd(theta_L)))
z=R*sind(x)
E=(atan(y/z)*180/%pi)-x

x1=(180/%pi)*acos(cosd(theta_s1-theta_l1)*cosd(
   theta_L1))
y1=r-ceil(R*(cosd(theta_s1-theta_l1)*cosd(theta_L1))
   )
z1=R*sind(x1)
E1=(atan(y1/z1)*180/%pi)-x1
E1=floor(E1)

//calculation of slant range dx
k=(R/r)*cosd(E)
k=(180/%pi)*asin(k)
k=k+E
k=sind(k)
k=ceil(k*1000)/1000
//k=k+E
//k=sin(k)
dx=(R)^2+(r)^2-(2*r*R*k)
dx=sqrt(dx)


//calculation of slant range dy
k1=(R/r)*cosd(E1)
k1=(180/%pi)*asin(k1)
k1=k1+E1
```

```
k1=floor(k1)
k1=sind(k1)
k1=ceil(k1*1000)/1000
dy=(R)^2+(r)^2-(2*r*R*k1)
dy=sqrt(dy)

tr=dy+dx
delay=tr*10^6/c
x=50
td=delay+x

printf("Elevation angle, Ex =%.1f ",E)
printf("\n Elevation angle, Ey =%.1f ",floor(E1))
printf("\n Slant range dx of the earth station X is dx=%.2fkm",dx)
printf("\n Slant range dy of the earth station Y is dy=%.1fkm",dy)
printf("\n Therefore, total range to be covered is %.2fkm",tr)
printf("\n propagation delay=%.2fms",delay)
printf("\n\n Time required too transmit 500 kbs of information at \n a transmisssion speed of 10Mbps is given by 500000/10^7=%.0fms",500000000/10^7)
printf("\n\n Total Delay= %.2fms",td)
```

Scilab code Exa 3.13 Angular spacing and intersatellite distance

```
// Example 3.13, page no-102
clear
clc


da=38000                    // slant range of satellite A
db=36000                    // slant range of satellite
```

```
B
beeta=60                    // difference between longitudes of two satellites
R=42164                     // radius of the orbit of satellites


theta=(da^2+db^2-2*(R^2)*(1-cosd(beeta)))/(2*da*db)
theta=(180/%pi)*acos(theta)

d=sqrt(2*(R^2)*(1-cosd(beeta)))
printf("Angular spacing between two satellites viewed by earth station is ,\n theta= %.1f  ",theta)
printf("\nInter-satellite distance , d=%.0fkm",d)
```

Scilab code Exa 3.14 covered surface area

```
// Example 3.14, page no-107
clear
clc

r=42164                 // orbital radius of the satellite in km
R=6378                  //Earth's radius in km

//refer to Figure 3.53
//for E=0
alfa=asin(R/r)*(180/%pi)
alfa=floor(alfa*10)/10
theta=90-alfa
//in the right angle triangle OAC,
k=sind(alfa)
k=floor(k*1000)/1000
oc=R*k
```

```
oc=ceil(oc*10)/10
A=2*%pi*R*(R-oc)


//for E=10
E=10
alfa1=asin((R/r)*cosd(E))*(180/%pi)
//alfa1=ceil(alfa1*100)/100
theta1=90-alfa1-E
//in the right angle triangle OAC,
k1=sind(alfa1+E)
k1=floor(k1*1000)/1000
oc1=R*k1
oc1=floor(oc1*10)/10
A1=2*%pi*R*(R-oc1)

printf("for E=0 ,\n covered surface area is %.1f km
    ^2",A)
printf("\n\n for E=10 ,\n covered surface area is %
    .1f km^2",A1)
```

Scilab code Exa 3.15 Area swept by ground track of satellite

```
// Example 3.15, page no-108
clear
clc


theta=30                    //satellite inclination to
    the equitorial plan
//the extreme latitudes covered in northern and
    southern hemisphere are the same as orbit
    inclination

printf("Extreme Northern latitude covered = %.0f   N
```

```
",theta)
printf("\n Extreme Southern latitude covered = %.0f   S",theta)
printf("\n\n In fact, the ground track would sweep\n all latitudes between %d N and %d S",theta,theta)
```

Chapter 4

Satellite Hardware

Scilab code Exa 4.1 Ejection velocity

```
// Example 4.1, page no-122
clear
clc


I=250      //specific impulse of a propellant
g=9.807    // acceleration due to gravity

v=I*g
printf("Ejection velocity of the propellant mass is,
    v= %.2f m/s",v)
```

Scilab code Exa 4.2 Mass of propellant to be burnt

```
// Example 4.2, page no-122
clear
clc

```

```
m=4330        //initial mass of the satellite
i=290         //specific impulse of a propellant
del_v=-100    //velocity increment
g=9.807       // acceleration due to gravity

m1=m*(1-exp(del_v/(g*i)))
printf("Mass of propellant necessary to be burnt is,
    m= %.0fkg",ceil(m1))
```

Scilab code Exa 4.3 Mass of propellant to be burnt

```
// Example 4.3, page no-123
clear
clc

m=2950        //initial mass of the satellite
F=450         //required thrust
T=10          // thrust for time period
i=300         //specific impulse of a propellant
g=9.807       // acceleration due to gravity

mi=F*T/(i*g)
printf("Mass of propellant that would be consumed is
    , m=%.2fkg",mi)
```

Scilab code Exa 4.5 Required no of solar cells

```
// Example 4.5, page no-134
clear
clc

p=2000        //electrical energy to be generated from
    solar panel in Watt
```

```
fi=1250      // solar flux falling normally to the solar cell in worst case
s=4*10^-4    // Area of each solar cell
e=0.15       //conversion efficiency of solar cell includingthe losses
theta=10     // angle made by rays of sun with normal

n=p/(fi*s*e)
n1=ceil(n)*%pi
n2=ceil(n1)/cosd(theta)
printf("Required no of solar cells , n = %.0f cells",ceil(n1))
printf("\n No of cells when sunrays are making an angle of 10  are %.0f",ceil(n2))
```

Scilab code Exa 4.6 Mass of battery system

```
// Example 4.6, page no-134
clear
clc

p=3600       //Power required
t=1.2        //worst case eclipse period
c=90         //capacity of each cell in Ah
v=1.3        //voltage of each cell in V
d=0.8        // Depth of discharge
e=0.95       //Discharge efficiency
E_sp=60      //specific energy specification of the battery

energy=p*t
n=energy/(c*v*d*e)
E_b=energy/(d*e)
m=E_b/E_sp
```

```
printf("No of cells , n= %.0f cells\n Energy required
    to be stored in the battery system is %.1f Wh\n
   Mass of battery system = %.2f kg",n,E_b,m)
```

Scilab code Exa 4.7 Antenna Gain

```
// Example 4.7, page no-153
clear
clc

theta=0.5        //azimuth beam width=Elevation beam
   width
f=6*10^9         // operating frequency 6 Ghz
c=3*10^8       //speed of light in cm/s
theta_r=theta*%pi/180
theta_r=ceil(theta_r*10^5)/10^5
A=4*%pi/(theta_r^2)
A=ceil(A*100)/100
A_dB=10*log10(A)
lambda=c/f
Ag=(A*lambda^2)/(4*%pi)

printf("\nGain in dB = %.2f dB \nAntenna gain
   expressed in terms of\nantenna aperture(A) is
   given by G = %.2f m^2",A_dB,Ag)
```

Scilab code Exa 4.8 Aperture efficiency and effective aperture

```
// Example 4.8, page no-153
clear
clc

la=0.5       //length efficiency in azimuth direction
```

```
le=0.7        //length efficiency in elevation
   direction
A=10          // Actual projected area of an antenna

Ae=la*le
Aee=Ae*A
printf("Aperture efficiency = %.2f \n Effective
   Aperture = %.1f m^2",Ae,Aee)
```

Scilab code Exa 4.9 Directivity

```
// Example 4.9, page no-154
clear
clc

p=100          //Antenna power in W
pd=10          //Power Density in mW/m^2
d=1000         //distance in m
p2=10000       // New antenna power
directivity=10*log10(p2/p)
printf("Directivity (in dB)= %d dB",directivity)
```

Scilab code Exa 4.10 null to null beam width

```
// Example 4.10, page no-154
clear
clc

beam_w=0.4     //antenna's 3dB beam width
Ae=5           //Effective Aperture of Antenna

printf("The null-to-null beam width of a paraboloid
   reflector is twice its 3dB beam width. \n
```

```
    Therefore ,  Null−to−null  beam  width  = %.1 f   ” ,2*
    beam_w)
```

Scilab code Exa 4.11 received signal strength

```
// Example 4.11 , page no−154
clear
clc
d=20          //received signal strenth in dB
loss=3        //incident polarization is circular and
    antenna is circularly polarized
theta=60      //received wave making angle with
    horizontal
total=d+loss
los=d*log10(1/cosd(theta))

printf(”(a)\n When received polarization is same as
    antenna \n polarization ,thepolarization loss is
    zero.\n Therefore , received sinal strenth = %ddB”
    ,total)
printf(”\n\n(b)\n When the incident wave is
    vertically polarized ,\n the angle between
    incident polarization and antenna polarization is
     90  \n Hence , Polarization loss = infinity\n
    received signal strength = 0”)
printf(”\n\n(c)\n When incident wave is left−hand
    circularly polarized\n and antenna polarization
    is linear ,\n then there is polarization loss of
    %ddB and\n received signal strength is %ddB”,loss
    ,d)
printf(”\n\n(d)\n Polarization loss = %ddB \n
    Received signal strength = %ddB”,los,ceil(total-
    los))
```

Scilab code Exa 4.12 gain and beam width

```
// Example 4.12, page no-155
clear
clc
Ea=1          //effective aperture
f=11.95*10^9     //downlink operating frequency
c=3*10^8   //speed of light

Ae=floor((%pi*1000*Ea^2)/4)/1000
lamda=floor(c*1000/f)/1000
ag=floor(100*4*%pi*Ae/lamda^2)/100
adb=floor(100*10*log10(ag))/100
width=70*lamda/Ea
printf("Operating wavelength = %.3fm\n Antenna Gain
   = %.2f\n Antenna Gain in dB = %.2fdB\n 3dB beam
   width = %.2f  ",lamda,ag,adb,width)
```

Scilab code Exa 4.13 beam width

```
// Example 4.13, page no-155
clear
clc
f=2.0            //reflector focal length
d=2.0            // reflector diameter
l=90/100         //90% of the angle
theta=4*(atand(1/(4*f/d)))
theta=4*atand(0.25007)     // this value gives exact
   answer as in book
dbw=l*theta
printf("The angle subtended by the focal point feed\
   n at the edges of the reflector is, theeta = %.2
```

```
f \n\n 3dB beam width = %.2 f \n null-to-null
beam width = % .2 f ",theta,dbw,floor(200*dbw)
/100)
```

Scilab code Exa 4.14 phase angle

```
// Example 4.14, page no-155
clear
clc
c=3*10^8    //speed of light
f=2.5*10^9  //operating frequency
s=0.1       //inter element spacing
theta =10   //10  right towards array axis
l=c/f
fi=(360*s/l)*ceil(10000*sind(theta))/10000
fi=ceil(10*fi)/10
printf("The phase angle for elements 1,2,3,4 and 5 \
   n are respecively 0 ,%.1 f ,%.1 f ,%.1 f  and %
   .1 f ",fi,2*fi,3*fi,4*fi)
```

Scilab code Exa 4.15 Earth station EIRP

```
// Example 4.15, page no-156
clear
clc

p=10000        //power fed to the antenna in W
ag=60           //Antenna gain
loss=2          //Power lossin feed system
adb=10*log10(p)
EIRP=adb+ag-loss
printf("Earth station EIRP = %ddB",EIRP)
```

Chapter 5

Communication Techniques

Scilab code Exa 5.1 Power Saving

```
// Example 5.1, page no-174
clear
clc
//for case (a)
m=0.5            //modulation index
//for AM
pt1=(1+(m^2)/2)

//for SSBSC
pt2=(m^2)/4

//% power saving
p=(pt1-pt2)*100/pt1
p=floor(p*10)/10
printf("Percentage power saving is %.1f%%",p)

//for case (b)
m=1              //modulation index
//for AM
pt1=(1+(m^2)/2)

```

```
//for SSBSC
pt2=(m^2)/4

//% power saving
p=(pt1-pt2)*100/pt1
p=floor(p*10)/10
printf("\n Percentage power saving is %.1f%%",p)
```

Scilab code Exa 5.2 Total Power in the modulated signal

```
// Example 5.2, page no-174
clear
clc

pc=500        //energy of carrier signal
m=0.6        //AM modulation index
//for (a)
pt=pc*(1+(m^2)/2)

//for (b)
pt2=pc*(m^2)/4

printf("(a)\n A3E is the double side band AM with
    full carrier.\n Therefore, Pt= %.0f W\n\n (b)\n
    J3E is an SSBSC system.\n Therefore, Pt= %.0f W",
    pt,pt2)
```

Scilab code Exa 5.3 Percentage power saving

```
// Example 5.3, page no-175
clear
clc

```

```
m=0.6              //60% modulation
//for A3E
pt1=(1+(m^2)/2)

//for J3E
pt2=(m^2)/4

//% power saving
p=(pt1-pt2)*100/pt1
p=ceil(p*10)/10
printf("Percentage power saving is %.2f%%",p)
```

Scilab code Exa 5.4 Carrier Frequency

```
// Example 5.4, page no-175
clear
clc

//multiplication of two signals gives AM with
   frequency component(wc-wm) and (wc+wm) and its BW
    is 2wm
bw=0.5/100         //bw is 0.5% of carrier freq.
wc=2/bw
printf("Wc = %.0fWm",wc)
```

Scilab code Exa 5.5 modulation parameters

```
// Example 5.5, page no-190
clear
clc

//comparing given equation with stanard equation
m=6                  //Modulation Index
```

```
wc=7.8*10^8          //unmodulated carrier frequency
wm=1450              //Modulating frequency

fc=wc/(2*%pi)
fm=wm/(2*%pi)
printf("Unmodulated carrier frequency, fc = %.2f MHz
    \n The modulation index m = %d \n Modulating
   frequency, fm = %.2f Hz",fc/10^6,m,fm)
```

Scilab code Exa 5.6 maximum phase and frequency deviation

```
// Example 5.6, page no-190
clear
clc

printf("comparing given equation with stanard
   equation, we have,\n Maximum phase deviation  = 6
    radian\n Maximum frequency deviation =
   12*3.14*10^3 radian/s = 6 KHz")
```

Scilab code Exa 5.7 frequency deviation and Bandwidth

```
// Example 5.7, page no-191
clear
clc

//comparing given equation with stanard equation
mf=150               //modulation index
fm=1                 // modulating frequency in KHz
fd=mf*fm
bw=2*(mf+1)*fm
printf("frequency deviation = %.0f kHz\n Bandwidth =
    %.0f kHz \n\n Expression for instantaneous
```

```
frequency is given by, \n f = 10^8-150*(10^3)*sin
(2*3.14*10^3*t)",fd,bw)
```

Scilab code Exa 5.8 modulation index and Bandwidth

```
// Example 5.8, page no-191
clear
clc


fd=50              //frequency deviation in kHz
fm=1               //modulating frequency in kHz for
    case 1
fm2=100            //modulating frequency in kHz for
    case 2

//for casse 1
m=fd/fm
bw=2*(m+1)*fm
//for case 2
m2=fd/fm2
bw2=2*(m2+1)*fm2

printf("For first case\n Modulation index = %.0f \n
    Bandwidth = %.0f kHz \n\n For second case\n
    Modulation index = %.1f \n Bandwidth = %.0f kHz",
    m,bw,m2,bw2)
```

Scilab code Exa 5.9 modulation index and Bandwidth

```
// Example 5.9, page no-192
clear
clc
```

```

bw=20*10^3      //bandwidth in Hz
fm=1* 10^3      //modulating frequency in Hz
mf=(bw/(2*fm))-1
new_mf=mf*6
new_fm=0.5  //kHz
new_bw=2*(new_mf+1)*new_fm
printf("mf=%.0f\n New modulation index = %.0f\n New bandwidth = %.0f kHz",mf,new_mf,new_bw)
```

Scilab code Exa 5.10 Daviation ratio and Bandwidth

```
// Example 5.10, page no-192
clear
clc

fd=75       //Maximum allowed frequency deviation in kHz
fm=15       //Highest modulating frequency in kHz

D=fd/fm
bw=2*(D+1)*fm
printf("Deviation Ratio, D = %.0f\n Bandwidth = %.0f kHz",D,bw)
```

Scilab code Exa 5.11 Sampling level quantizing levels and no of bits

```
// Example 5.11, page no-199
clear
clc

fm=3200         //highest frequency component in message signal
```

```
k=48000          //channel capacity in b/s
fs=2*fm
n=k/fs
n=floor(n)
printf("n = %.0f\n L = 2^7 = %.0f\n fs = %.3f kHz",n,2^7,(k/7)/1000)
```

Scilab code Exa 5.12 Nyquist rate

```
// Example 5.12, page no-199
clear
clc

//rearrangin equation and comparing it with standard equation we have,
f=2500            //Highest frequency component in the signal in Hz
printf("Nyquist rate = 2 x f\n\t       = %.0f Hz = %.0f kHz",2*f,2*f/1000)
```

Scilab code Exa 5.13 bits per sample and time duration for one bit

```
// Example 5.13, page no-199
clear
clc

l=128         //no of Quantizing levels
fs=10000      //sampling frequency in Hz
n= log2(l)
t=1/(n*fs)
printf("Number of bits per sample (n) = %.0f\n Time duration of one bit of binary encoded signal is %.3f micro second",n,t*10^6)
```

Scilab code Exa 5.15 Sampling rate and sampling interval

```
//Example 5.15, page no-208
clear
clc

f1=2.4          //first signal frequency
f2=3.2          //2nd signal frequency
f3=3.4          //3rd signal frequency

//minimum sampling rate for each of the signals
   would be twice the highest frequency component


sr=3*(f3*2)
st=10^6/(sr*10^3)
printf("Sampling rate of the composite signal = %.1f
    kHz \n Sampling interval of the composite signal
    = %.0f micro second",sr,st)
```

Scilab code Exa 5.16 Bit duration and Transmission Rate

```
//Example 5.16, page no-209
clear
clc

bw=3.2            // voice channel band limited
   frequency in kHz
r=1.2             //1.2 times the Nyquist rate
n=24              //no of  voice channel
b=8               // 8-bit PCM
```

```
sr=2*bw*r
p=10^6/(sr*10^3)
N=(n*b)+1
bit_d=p/N
bit_d=ceil(bit_d*1000)/1000
tr=1/bit_d

printf("Number of bits in each frame = %.0f \n Bit
   duration = %.3f micro second \n Transmission rate
    = %.3f Mbps",N,bit_d,ceil(tr*1000)/1000)
```

Chapter 6

Multiple Access Techniques

Scilab code Exa 6.1 TDMA frame

```
// Example 6.1, page no-230
clear
clc
t=20        //TDMA frame length in ms
lc=352      //length of carrier and clock recovery
   frequency in bits
lu1=48      //length of unique word in bits
lo=510      //length of order wire channel in bits
lm= 256     //length of management channel in bits
lt=320      // length of transmit timming channel in
   bits
ls1=24      // length of service channel in bits
gt=64       // Guard time in bits
rb=2        // reference burst

lr=lc+lu1+lo+lm+lt
tb=lc+lu1+lo+ls1
tob=(lr*rb)+(tb*t)+((t+rb)*gt)
printf("(a)\nThe length of reference burst(from
   given data) is %d bits\n\n(b)\nThe length of
   traffic burst premable(from given data)is %d bits
```

```
\n\n(c)\nTotal number of overhead bits is %d bits
",lr,tb,tob)
```

Scilab code Exa 6.2 Frame efficiency

```
// Example 6.2, page no-230
clear
clc
t=20        //TDMA frame length in ms
lc=352      //length of carrier and clock recovery
   frequency in bits
lu1=48      //length of unique word in bits
lo=510      //length of order wire channel in bits
lm= 256     //length of management channel in bits
lt=320      // length of transmit timming channel in
   bits
ls1=24      // length of service channel in bits
gt=64       // Guard time in bits
rb=2        // reference burst
br=90*10^6 //burst bit rate 90Mbps

bfr=br*t*10^-3
lr=lc+lu1+lo+lm+lt
tb=lc+lu1+lo+ls1
tob=(lr*rb)+(tb*t)+((t+rb)*gt)
feff=(bfr-tob)*100/bfr
feff=ceil(feff*100)/100
printf("Frame efficiency = %.2f%%",feff)
```

Scilab code Exa 6.3 TDMA Frame

```
// Example 6.2, page no-230
clear
```

```
clc
t=20        //TDMA frame length in ms
lc=352      //length of carrier and clock recovery
   frequency in bits
lu1=48      //length of unique word in bits
lo=510      //length of order wire channel in bits
lm= 256     //length of management channel in bits
lt=320      // length of transmit timming channel in
   bits
ls1=24      // length of service channel in bits
gt=64       // Guard time in bits
rb=2        // reference burst
br=90*10^6 //burst bit rate 90Mbps
dr= 64*10^3  //data rate 64 kbps
bfr=br*t*10^-3
lr=lc+lu1+lo+lm+lt
tb=lc+lu1+lo+ls1
tob=(lr*rb)+(tb*t)+((t+rb)*gt)
feff=(bfr-tob)*100/bfr
feff=ceil(feff*100)/100
vsb=dr*t*10^-3
x=bfr*feff/100
printf("The number of bits in a frame for a voice
   sub-burst is %d\n\n The total no of bits
   available in a frame for carrying traffic is %d\n
   \n Maximum no of PCM voice channels in a frame is
    %d channels",vsb,x,x/vsb)
```

Scilab code Exa 6.4 Doppler Shift

```
// Example 6.4, page no-231
clear
clc

R=42150       //orbital radius of satellite
```

```
oi=0.25/100  // orbit inclination
acc=0.3      //error of 0.3 degree
c=3*10^8     // speed of light
x=oi*R
x=ceil(x*10)/10
y=R*2*%pi*acc/360
y=ceil(y*10)/10
z=sqrt(x^2+y^2)
z=ceil(z*10)/10
delay=z*10^6/c
delay=floor(delay*1000)/1000
pd=2*delay
printf("variation in alitude caused byorbit
    inclination = %.1fkm\n variation due to station-
    keeping error of 0.3  = %.1fkm",x,y)
printf("\n Both these errors will introduce a
    maximum range variation of %.1fkm\n This cause a
    one-way propagation delay of %.3fms\n Round trip
    propagation delay =%.2fms\n Dopler Shift = %.2f
    ms in 8h=56.25 ns/s",z,delay,delay*2,pd)
```

Scilab code Exa 6.5 Chip Duration and chip rate

```
// Example 6.5, page no-238
clear
clc

de=40        //Doppler effect variation due to
    station-keeping errors in ns/s
d=280        //Sttelite round trip delay in ms
c=20/100     // DS-CDMA signals should not exceed 20
    % of the chip duration

te=de*10^-9*d*10^-3
tc=te/c
```

```
printf("Chip Duration, Tc = %.0f ns \n This gives
   maximum chip rate as (1/56)Gbps = 1000/56 Mbps =
   %.3f Mbps",tc*10^9,1000/56)
```

Scilab code Exa 6.6 Maximum Permissible Doppler Effect

```
// Example 6.6, page no-238
clear
clc

cr=25               //Chip rate is 25 Mbps
c=20/100            // DS-CDMA signals should not exceed
    20% of the chip duration
d=1000/cr           //chip duration in ns
tr=c*d
x=tr/(280*10^-3)
printf("The maximum allowable timing error per
   satellite round trip is %.0f ns\n This %.0f ns
   error is to occur in 280 ms.\n Therefore, maximum
    permissible Dopler effect variation is %.2f ns/s
   ",tr,tr,x)
```

Scilab code Exa 6.7 Noise reduction

```
// Example 6.7, page no-238
clear
clc
cr=20*10^6              //chip rate in Mbps
ir= 20*10^3             //information bit rate
g=10*log10((cr)/(ir))
printf("Noise reduction achhievable = Processing
   gain = %.0f dB",g)
```

Chapter 7

Satellite Link Design Fundamentals

Scilab code Exa 7.1 Power received by the receiving antenna

```
// Example 7.1, page no-249
clear
clc

d=36000 *10^3 //distance of geostationary satellite
   from earth's surface
Gt=100  // Antenna gain of 20dB
Pt=10   // Power radiated by earth station

Prd=Pt*Gt/(4*%pi*d^2)
printf("Prd = %.4f * 10 ^-12 W/m^2\n Power received
   by the receiving antenna is given by  Pr = %.3f
   pW",Prd*10^12,Prd*10^13)
```

Scilab code Exa 7.2 free space path loss

```
// Example 7.2, page no-262
clear
clc

c=3*10^8           //speed of light
R=10000            //path length
f=4                // operating frequencyin GHz
EIRP=50            //in dB
gr=20              //antenna gain in dB
rp=-120            // received power in dB
//(a)
lamda=c/(f*10^9)
pl=20*log10(4*%pi*R/lamda)

//(b)
Lp=EIRP+gr-rp
printf("(a)\n Operating wavelength = %.3f m\n Path loss(in dB) = %.2f dB",lamda,pl)
printf("\n\n (b)\n Path loss = %.0fdB",Lp)
```

Scilab code Exa 7.3 Attenuation

```
// Example 7.3, page no-262
clear
clc

p=75          // rotation of plane of polarization

//Polarization rotation is inversaly propotional to square of the operating frequency
f= 5          //frequency increased by factor
x=f^2         //rotation angle will decrease by aa factor of 25
p_ex=p/x
Apr=-20*log10(cosd(p))
```

```
Apr2=-20*log10(cosd(p_ex))
printf("For polarization  mismatch angle = 75  \n
    Attenuation = %.2f dB",Apr)
printf("\n\n For polarization  mismatch angle = 3
    \n Attenuation = %.3f dB",Apr2)
```

Scilab code Exa 7.4 Noise temperature and noise figure

```
// Example 7.4, page no-270
clear
clc

g1=30           //gain of RF stage in dB
t1=20           //Noise temperature in K
g2=10           //down converter gain in dB
t2=360          //noise temperature in K
g3=15           //gain of IF stage in dB
t3=1000         //noise temperature in K
t=290           //reference temperature in K

G1=1000         //30 dB equivalent gain
Te=t1+(t2/G1)+t3/(G1*g2)
F=1+Te/t
printf("Effective noise temperature, Te = %.2fK",Te)
printf("\n\n System Noise Figure, F = %.2f",F)
```

Scilab code Exa 7.5 overall noise figure

```
// Example 7.5, page no-271
clear
clc

g1=30           //gain of RF stage in dB
```

```
t1=20           //Noise temperature in K
g2=10           //down converter gain in dB
t2=360          //noise temperature in K
g3=15           //gain of IF stage in dB
t3=1000         //noise temperature in K
t=290           //reference temperature in K

G1=1000         //30 dB equivalent gain
//Te=t1+(t2/G1)+t3/(G1*g2)
F1=1+t1/t
F2=1+t2/t
F3=1+t3/t

F=F1+((F2-1)/G1)+(F3-1)/(G1*g2)
printf("Noise Figure specificatios of the three
    stages are as follow,\n\n F1 = %.3f\n F2 = %.2f\n
     F3 = %.2f",F1,F2,F3)
printf("\n\n The overall noise figure is, F = %.2f",
    F)
```

Scilab code Exa 7.6 noise figure

```
// Example 7.6, page no-272
clear
clc

L=1.778             //Loss factor of the feeder 2.5dB
    equivalent
ts=30               //Noise temperature of sattelite
    receiver in K
t=50                //Noise temperature in K
ti=290              // reference temperature in K
x=t/L
y=ti*(L-1)/L
Te=x+y+ts
```

```
F1=1+(ts/ti)
F2=1+(Te/ti)
printf("contribution of antenna noise temperature
   when\n referred to the input of the receiver is %
   .1f K",x)
printf("\n\n Contribution of feeder noise when
   referred to the\n input of the receiver is %.1f",
   y)
printf("\n\n1. Noise figure in first case = %.3f = %
   .3f dB",F1,10*log10(F1))//answer in book is
   different 0.426dB
printf("\n\n2. Noise figure in second case = %.3f =
   %.2f dB",F2,10*log10(F2))
```

Scilab code Exa 7.7 Loss factor

```
// Example 7.7, page no-272
clear
clc

Ta= 40          //Antenna Noise temperature
Ti=290          //Reference temperature in K
T=50           // Effecitve input noise temperatuire
Tf=Ti
L=(Ta-Tf)/(T-Tf)
L=ceil(L*10^4)/10^4
printf("Loss factor = %.4f = %.3f dB",L,10*log10(L))
```

Scilab code Exa 7.8 System noise temperature

```
// Example 7.8, page no-273
clear
clc
```

```

Ta=50            //Antenna Noise temperature
Tf=300           //Thermodynamic temperature of the feeder
Te=50            // Effecitve input noise temperatuire

//(a)
Lf=1
T=(Ta/Lf)+(Tf*(Lf-1)/Lf)+Te
printf("(a)\n System noise temperature = %.0fK",T)

//(b)
Lf=1.413
T=(Ta/Lf)+(Tf*(Lf-1)/Lf)+Te
printf("\n\n (b)\n System noise temperature = %.3fK",ceil(T*10^3)/10^3)
```

Scilab code Exa 7.9 carrier to interface ratio

```
// Example 7.9, page no-278
clear
clc
e=35       //EIRP radiated by satellite in dBW
g=50       //receiver antenna gain in dB
e1=30      //EIRP of interfacing satellite in dBW
theeta=4 //line-of-sight between earth station and interfacing sattelite

x=(e-e1)+(g-32+25*log10(theeta))
printf("carrier-to-interface (C/I) = %.2f dB",x)
```

Scilab code Exa 7.10 carrier to interface ratio

```
// Example 7.10, page no-279
clear
clc

ea=80      //EIRP value of earth station A in dBW
eb=75      //EIRP value of earth station B in dBW
g=50      //transmit antenna gain in dB
gra=20      //receiver antenna gain for earth station A in dB
grb=15      //receiver antenna gain for earth station B in dB
theeta=4 //viewing angle of the sattelite from two earth station
eirp_d=eb-g+32-25*log10(theeta)
c_by_i=ea-eirp_d+(gra-grb)
printf("carrier-to-interference ratio at the satellite due to\n inteference caused by Eart station B is, (C/I) = %.0f dB ",c_by_i)
```

Scilab code Exa 7.11 carrier to interface ratio

```
// Example 7.11, page no-279
clear
clc

//carrier sinal strength at sattelite by uplink
u=10000    // equivalent to 40dB

//carrier sinal strength at eart station by downlink
d=3162.28 //equivalent to 35dB

x=1/((1/u)+(1/d))
printf("Total carrier-to-interference ratio is %.2f = %.1f dB",x,10*log10(x))
```

Scilab code Exa 7.12 Longitudinal separation between two satellites

```
//Example 7.12, Page no.280
clear
clc

theeta=5  //Angle form by slant ranges of two
    satellites
dA=42100*10^3  //Slant range of satellite A
dB=42000*10^3  //Slant range of satellite B
r=42164*10^3   //radius of geostationary orbit
beeta=((dA^2+dB^2-cosd(theeta)*2*dA*dB)/(2*r^2))
beeta=ceil(beeta*10^3)/10^3
beeta=(180/%pi)*acos(1-beeta)
printf("Longitudinal separation between two
    satellites is %.3f ",beeta)
```

Scilab code Exa 7.13 Gain per degree Kelvin

```
//Example 7.13, Page no.281
clear
clc
Ga=60      //Antenna Gain in dB
Ta= 60     //Noise teperature of Antenna
L1=1.12    //Feeder Loss equivalent to dB
T1=290     //Noise teperature of stage 1
G2=10^6      //Gain of stage 2 in dB
T2=140     //Noise teperature of stage 2
T3=10000   //Noise teperature of stage 3
G=Ga-0.5   // input of low noise amplifier
Ts=(Ta/L1)+(T1*(L1-1)/L1)+T2+(T3/G2)
Ts=floor(Ts*100)/100
```

```
x=G-10*log10(Ts)
printf("Tsi = %.2fK\n\n G/T(in dB/K)= %.0f dB/K",Ts,
  x)
```

Scilab code Exa 7.14 Gain per degree Kelvin

```
//Example 7.14, Page no.282
clear
clc

Ga=60       //Amplifier Gain in dB
Ta= 60      //Noise teperature of Antenna
L1=1.12     //Feeder Loss equivalent to dB
T1=290      //Noise teperature of stage 1
G2=10^6       //Gain of stage 2 in dB
T2=140      //Noise teperature of stage 2
T3=10000    //Noise teperature of stage 3
G=Ga-0.5    // input of low noise amplifier

T=Ta+T1*(L1-1)+L1*(T2+(T3/G2))
x=G-10*log10(T)
printf("T = %.1fK\n\n G/T = %.0f dB/k",T,ceil(x))
printf("\n\n It is evident from the solutions of the
    problems 13 and 14\n that G/T ratio is invarient
    regardless of the reference point in agreement \
  n with a statement made earlier in the text.")
```

Scilab code Exa 7.15 Link Margin

```
//Example 7.15, Page no.286
clear
clc

```

```
f=6*10^9      //uplink frequency
eirp= 80      //Earth station EIRP in dBW
r=35780       //Earth station satellite distance
l=2           //attenuation due to atomospheric factors in dB
e=0.8         // satellite antenna's aperture efficiency
a=0.5         // satellite antenna's aperture area
T=190        // Satellite receiver's effective noise temperature
bw=20 *10^6 //Satellite receiver's bandwidth
cn=25        // received carrier-to-noise ratioin dB
c=3*10^8     //speed of light

k=1.38*10^-23
lamda=c/f
G=e*4*%pi*a/lamda^2
G=ceil(G*100)/100
Gd=10*log10(G)
p=10*log10(k*T*bw)
pl=20*log10(4*%pi*r*10^3/lamda)
rp=eirp-l-pl+Gd
rp=floor(rp*100)/100
rc=floor((rp-p)*100)/100
lm=rc-cn
printf("Satellite Antenna gain, G = %.2f = %.2f dB \n Receivers Noise Power = %.1f dB\n free-space path loss = %.2f dB \n received power at satellite = %.2f dB \n receiver carrier = %f is stronger than noise.\n It is %.2f dB more than the required threshold value.\n Hence,  link margin = %.2f dB",G,Gd,p,pl,rp,rc,lm,lm)
```