Scilab Textbook Companion for Optical Communiation by A. Kalavar[1]

Created by
Gaurav Mittal
B.Tech
Electrical Engineering
Uttarakhand Technical University
College Teacher
NA
Cross-Checked by
Chaya

June 2, 2016

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

Book Description

Title: Optical Communiation

Author: A. Kalavar

Publisher: Techmax Publications, Pune

Edition: 1

Year: 2011

ISBN: 978-81-8492-845-7

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

Overview of Optical Fiber Communciations

Scilab code Exa 10.q duration

```
// Question 10
clc;
clear;
Bit_rate = 2d12;        // bit rate of channel
// Given sequence is 010111101110
Shortest_duration = 1 * (1/Bit_rate);      // shortest duration is '1'
Widest_duration = 4 * (1/Bit_rate);      //widest duration is '1111'
Shortest_duration=Shortest_duration*10^12;        // Converting into nano seconds
Widest_duration=Widest_duration*10^12;        // Converting into nano seconds
printf("\nShortest duration is %.1f nano second.", Shortest_duration);
printf("\nWidest duration is %d nano second.", Widest_duration);
```

Scilab code Exa 1.6.1 duration

```
// Example 1.6.1
clc;
clear;
Bit_rate = 2d9;       // bit rate of channel
// Given sequence is 010111101110
Shortest_duration = 1 * (1/Bit_rate);      // shortest duration is '1'
Widest_duration = 4 * (1/Bit_rate);      //widest duration is '1111'
Shortest_duration=Shortest_duration*10^9;        // Converting into nano seconds
Widest_duration=Widest_duration*10^9;        // Converting into nano seconds
printf("\nShortest duration is %.1f nano second.", Shortest_duration);
printf("\nWidest duration is %d nano second.", Widest_duration);
```

Scilab code Exa 1.12.1 capacity

```
// Example 1.12.1
clc;
clear;
Bandwidth = 2d6;      //Bandwidth of channel
Signal_to_Noise_ratio = 1;                //Signal to Noise ratio of channel
Capacity = Bandwidth * log2(1 + Signal_to_Noise_ratio); //computing capacity
Capacity=Capacity/10^6;
```

```
printf("Maximum capacity of channel is %d Mb/sec.",
   Capacity);
```

Chapter 2

optical fibers

Scilab code Exa 4.q birefringence

```
// Question 4
clc;
clear;
L_BL=8d-2;    //beat length
Br=2*3.14/L_BL;      //computing modal briefringence
printf("\nModal briefringence is %.1f per meter.",Br
   );
```

Scilab code Exa 5.q modal birefringence

```
// Question 5
clc;
clear;
L_BL=0.6d-3;    //beat length
lamda=1.4d-6;    //wavelength
L_BL1=70;
Bh=lamda/L_BL;      //computing high briefringence
Bl=lamda/L_BL1;      //computing low briefringence
```

```
printf("\nHigh briefringence is %.2e.\nLow
   briefringence is %.1e.",Bh,Bl);
```

Scilab code Exa 2.4.1 NA and critical angle

```
// Example 2.4.1:Numerical Aperture and critical
   angle
clc;
clear;
close;
n1=1.46;//refractive index
d=0.01;//difference
na=n1*sqrt(2*d);//numerical aperture
x=1-d;//
oc=asind(x);//in degree
disp(na,"numerical aperture is")
disp(oc,"critical angle at core cladding interface
   is ,(degree)=")
```

Scilab code Exa 2.5.1 criticle anlge NA nad accerptance angle

```
// Example 2.5.1:Numerical Aperture ,critical angle
   and acceptance angle
clc;
clear;
close;
n2=1.45;//core refrative index
n1=1.49;//cladding refrative index
oc=asind(n2/n1);//in degree
na=sqrt(n1^2-n2^2);//numerical aperture
pc=asind(na);//degree
disp(oc,"critical angle is,(degree)=")
disp(na,"numerical aperture is,=")
```

```
disp(pc,"acceptance angle is,(degree)=")
```

Scilab code Exa 2.5.2 refractive index

```
// Example 2.5.2
clc;
clear;
delta = 1.2/100;        // Relative refractive difference index
n1=1.45;          // Core refractive index
NA= n1*sqrt(2*delta);        //computing numerical aperture
Acceptance_angle = asind(NA);    //computing acceptance angle
si = %pi * NA^2;          //computing solid acceptance angle
printf("\nNumerical aperture is %.3f.\nAcceptance angle is %.2f degree.\nSolid acceptance angle is %.3f radians.",NA,Acceptance_angle,si);
//answer in the book for Numerical aperture is 0.224, deviation of 0.001
//answer in the book for solid acceptance angle is 0.157, deviation of 0.002
```

Scilab code Exa 2.5.3 acceptance angle

```
// Example 2.5.3
clc;
clear;
NA = 0.45;        // Numerical Aperture
Acceptance_angle = asind(NA);    //computing acceptance angle.
```

```
printf("\nAcceptance angle is %.1f degree.",
   Acceptance_angle);
```

Scilab code Exa 2.5.4 full cone angle

```
// Example 2.5.4
clc;
clear;
diameter = 1;         //Diameter in centimeter
Focal_length = 10;        //Focal length in centimeter
radius=diameter/2;   //computing radius
Acceptance_angle = atand(radius/Focal_length);   //
   computing acceptance angle
Conical_full_angle = 2*Acceptance_angle;         //
   computing conical angle
Solid_acceptance_angle = %pi*Acceptance_angle^2;
   //computing solid acceptance angle
NA = sqrt(Solid_acceptance_angle/%pi);       //
   computing Numerical aperture
printf("\nNumerical aperture is %.2f.\nConical full
   angle is %.2f degree.",NA,Conical_full_angle);
```

Scilab code Exa 2.6.1 acceptance angle

```
// Example 2.6.1
clc;
clear;
NA = 0.45        //Numerical aperture
betaB = 45        // Skew ray change direction by 90
   degree at each reflection
Meridional_theta = asind(NA);    //computing
   acceptacne angle for meridoinal ray
```

```
Skew_theta = asind(NA/cosd(betaB));  //computing
    acceptacne angle for skew ray
printf("\nAcceptacne angle for Meridoinal ray is %.2
    f degree.\nAcceptance angle for Skew ray %.1f
    degree.",Meridional_theta,Skew_theta);
```

Scilab code Exa 2.8.1 normalized frequency and guided modes

```
// Example 2.8.1
clc;
clear;
core_diameter=78d-6;        //core diameter
delta=1.4/100;      //relative index difference
lamda=0.8d-6;       //operating wavelength
n1=1.47;     //core refractive index
a=core_diameter/2;      //computing core radius
v= 2*3.14*a*n1*sqrt(2*delta)/lamda;      //computing
     normalized frequency
M=(v)^2/2;      //computing guided modes
printf("\nNormalized Frequency is %.3f.\nTotal
    number of guided modes are %.1f",v,M);
//answer in the book for normalized frequency is
    given as 75.156(incorrect) and for Guided modes
    is 5648.5(incorrect)
```

Scilab code Exa 2.8.2 cutoff wavelength

```
// Example 2.8.2
clc;
clear;
n1=1.47     //refractive index of core
a=4.3d-6;       //radius of core
delta=0.2/100       //relative index difference
```

```
lamda= 2*3.14*a*n1*sqrt(2*delta)/2.405;      //
   computing wavelength
lamda=lamda*10^9;
printf("Wavelength of fiber is %d nm.",lamda);
//answer in the book is given as 1230nm which is
   incorrect.
```

Scilab code Exa 2.8.3 solid angle

```
// Example 2.8.3
clc;
clear;
n1=1.482;        //refractive index of core
n2=1.474;        //refractive index of cladding
lamda=820d-9;        //Wavelength
NA=sqrt(n1^2 - n2^2);        //computing Numerical
   aperture
theta= asind(NA);        //computing acceptance
   angle
solid_angle=%pi*(NA)^2;      //computing solid angle
a=2.405*lamda/(2*3.14*NA);       //computing core
   radius
a=a*10^6;
printf("\nNumerical aperture is %.3f.\nAcceptance
   angle is %.1f degrees.\nSolid angle is %.3f
   radians.\nCore radius is %.2f micrometer.",NA,
   theta,solid_angle,a);
//answer in the book for Numerical aperture is
   0.155, deviation of 0.001.
//answer in the book for acceptance angle is 8.9,
   deviation of 0.1.
//answer in the book for solid acceptance angle is
   0.075, deviation of 0.001.
//answer in the book for core radius is 2.02
   micrometer, deviation of 0.02 micrometer.
```

Scilab code Exa 2.8.4 normalized frequency

```
// Example 2.8.4
clc;
clear;
NA=0.16     //Numerical aperture
n1=1.45     //core refractive index
d=60d-6     //core diameter
lamda=0.82d-6       //wavelength
a=d/2;      //core radius
v=2*3.14*a*NA/lamda;        //computing normalized
   frequency
v=round(v);
M=v^2/2;        //computing guided modes
M=floor(M);
printf("if normalized frequency is taken as %d, then
    %d guided modes.",v,M);
```

Scilab code Exa 2.8.5 guided modes

```
// Example 2.8.5
clc;
clear;
NA=0.2;     //Numericla aperture
d=50d-6;        //Diameter of core
lamda=1d-6;       //Wavelength
a=d/2;      //computing radius
v=2*3.14*a*NA/lamda;     //computing normalized
   frequency
Mg=v^2/4;       //computing mode volume for
   parabollic profile
```

```
Mg=round(Mg);
printf("\nNormalized Frequency is %.1f.\nTotal
    number of guided modes are %.d.",v,Mg);
//answer in the book for guided modes is 247,
    deviation of 1.
```

Scilab code Exa 2.8.6 core diameter

```
// Example 2.8.6
clc;
clear;
delta=0.015;          //relative refractive index
n1=1.48;      //core refractive index
lamda=0.85d-6;      //wavelength
a=(2.4*lamda)/(2*3.14*n1*sqrt(2*delta));          //
    computing radius of core
d=2*a;       //computing diameter of core
a=a*10^7;
a=round(a);
a=a/10
d=d*10^6;
printf("\nCore radius is %.1f micrometer.\nCore
    diameter is %.1f micrometer.",a,2*a);
printf("\n\nWhen delta is reduced by 10 percent-");
delta=0.0015;
a=(2.4*lamda)/(2*3.14*n1*sqrt(2*delta));          //
    computing radius of core
d=2*a;       //computing diameter of core
a=a*10^7;
a=round(a);
a=a/10
d=d*10^6;
printf("\nCore radius is %.1f micrometer.\nCore
    diameter is %.1f micrometer.",a,2*a);
```

Scilab code Exa 2.8.7 guided modes

```
// Example 2.8.7
clc;
clear;
NA=0.25;        //Numericla aperture
d=45d-6;            //Diameter of core
lamda=1.5d-6;          //Wavelength
a=d/2;          //computing radius
v=2*3.14*a*NA/lamda;       //computing normalized
    frequency
Mg=v^2/4;           //computing mode volume for
    parabollic profile
Mg=round(Mg);
printf("\nNormalized Frequency is %.1f.\nTotal
    number of guided modes are %.d.",v,Mg);
//answer in the book for normalized frequency is
    23.55, deviation 0.05
```

Scilab code Exa 2.8.8 guided modes

```
// Example 2.8.8
clc;
clear;
NA=0.25;        //Numericla aperture
d=45d-6;            //Diameter of core
lamda=1.2d-6;          //Wavelength
a=d/2;          //computing radius
v=2*3.14*a*NA/lamda;       //computing normalized
    frequency
Mg=v^2/4;           //computing mode volume for
    parabollic profile
```

```
Mg=round(Mg);
printf("\nNormalized Frequency is %.1f.\nTotal
   number of guided modes are %.d.",v,Mg);
printf("\n\nNOTE - In the question NA is given 0.22.
    However while solving it is taken as 0.25");
// answer in the book for number of guided modes is
   given as 216, deviation of 1.

printf("\nHence solving for NA = 0.22 also,");
printf("\n\nWhen NA=0.22");
NA=0.22;      //Numericla aperture
d=45d-6;          //Diameter of core
lamda=1.2d-6;        //Wavelength
a=d/2;       //computing radius
v=2*3.14*a*NA/lamda;     //computing normalized
   frequency
Mg=v^2/4;         //computing mode volume for
   parabollic profile
Mg=round(Mg);
printf("\nNormalized Frequency is %.1f.\nTotal
   number of guided modes are %.d.",v,Mg);
```

Scilab code Exa 2.8.9 cut off parameter

```
// Example 2.8.9
clc;
clear;
n1=1.54;          //refractive index of core
n2=1.5;       //refractive index of cladding
a=25d-6;          //Radius of core
lamda=1.3d-6;        //Wavelength
NA=sqrt(n1^2-n2^2);
v=2*3.14*a*NA/lamda;     //computing normalized
   frequency
v=round(v);
```

```
Mg=v^2/4;          //computing mode volume for parabollic profile
Mg=round(Mg);
lamda_cut_off=v*lamda/2.405;     //computing cut off wavelength
lamda_cut_off=lamda_cut_off*10^6;
printf("\nNormalized Frequency is %.d.\nTotal number of guided modes are %.d.\nCut off wavelength is %.1f micrometer.",v,Mg,lamda_cut_off);
```

Scilab code Exa 2.9.1 modal birefringence

```
// Example 2.9.1
clc;
clear;
L_BL=8d-2;    //beat length
Br=2*3.14/L_BL;      //computing modal briefringence
printf("\nModal briefringence is %.1f per meter.",Br);
```

Scilab code Exa 2.9.2 output power

```
// Example 2.9.2
clc;
clear;
Pin=500d-6;      //input power
L=200;        //length of fiber
loss=2;      //loss associated with fiber
Pin_dbm=10 * log10 (Pin/(10^-3));    //computing input power in dBm
Pin_dbm=round(Pin_dbm);
Pout_dbm=Pin_dbm-L*loss;         //computing output power level
```

```
Pout= 10^(Pout_dbm/10);
printf("Output power is %.2e mW.",Pout);
```

Scilab code Exa 2.12.1 cut off wavelength

```
// Example 2.12.1
clc;
clear;
a=4.5d-6;          //core diameter
delta=0.25/100;       //relative index difference
lamda=0.85d-6;        //operating wavelength
n1=1.46;      //core refractive index
v= 2*%pi*a*n1*sqrt(2*delta)/lamda;       //computing normalized frequency
lamda_cut_off=v*lamda/2.405;         //computing cut off wavelength
lamda_cut_off=lamda_cut_off*10^9;
printf("\nCut off wavelength is %.d nanometer.",lamda_cut_off);
printf("\n\nWhen delta is 1.25 percent-");
delta=1.25/100;
v= 2*%pi*a*n1*sqrt(2*delta)/lamda;       //computing normalized frequency
lamda_cut_off=v*lamda/2.405;         //computing cut off wavelength
lamda_cut_off=lamda_cut_off*10^7;
lamda_cut_off=round(lamda_cut_off);
lamda_cut_off=lamda_cut_off*100;
printf("\nCut off wavelength is %.d nanometer.",lamda_cut_off);

//answer in the book for cut off wavelength in the book is given as 1214nm, deviation of 1nm.
```

Scilab code Exa 2.12.2 cut off number

```
// Example 2.12.2
clc;
clear;
a=50d-6;        //core radius
lamda=1500d-9;        //operating wavelength
n1=2.53;      //core refractive index
n2=1.5;      //cladding refractive index
delta=(n1-n2)/n1;        //computing delta
v= 2*3.14*a*n1*sqrt(2*delta)/lamda;       //computing
    normalized frequency
M=(v)^2/2;        //computing guided modes
printf("\nNormalized Frequency is %.1f\nTotal number
    of guided modes are %.d",v,M);
printf("\nNOTE − Calculation error in book. \n
   Normalized frequency is 477, it is calculated as
   47.66");

//Calculation error in book. Normalized frequency is
    477, it is calculated as 47.66, hence answers
   after that are erroneous.
//answers  in the book
//normalized frequency = 48.(incorrect)
//guided modes = 1152.(incorrect)
```

Scilab code Exa 2.12.3 guided modes

```
// Example 2.12.3
clc;
clear;
core_diameter=8d-6;          //core diameter
```

```
delta=0.92/100;        //relative index difference
lamda=1550d-9;         //operating wavelength
n1=1.45;       //core refractive index
a=core_diameter/2;        //computing core radius
v= 2*%pi*a*n1*sqrt(2*delta)/lamda;        //computing
   normalized frequency
M=(v)^2/2;        //computing guided modes
printf("\nNormalized Frequency is %.1f.\nTotal
   number of guided modes are %.d.",v,M);
```

Scilab code Exa 2.12.4 delay diffrence

```
// Example 2.12.4
clc;
clear;
delta=1/100;        //relative index difference
n1=1.5;       //core refractive index
c=3d8;
L=6;
n2=sqrt(n1^2-2*delta*n1^2);        //computing
   refractive index of cladding
delta_T=L*n1^2*delta/(c*n2);     //computing pulse
   broadning
delta_T=delta_T*10^11;
delta_T=round(delta_T);
printf("\nDelay difference between slowest and
   fastest mode is %d ns/km.",delta_T);
printf("\nThis means that a pulse broadnes by %d ns
   after travel time a distance of %d km.",delta_T,L
   );
```

Scilab code Exa 2.17.1 entrance angle

```
// Example 2.17.1
clc;
clear;
n1=1.48;          //core refractive index
n2=1.46;          //cladding refractive index
phi = asind(n2/n1);      //computing critical angle
NA = sqrt(n1^2 - n2^2); //computing numericla aperture
theta= asind(NA);        //computing acceptance angle
printf("\nCritical angle is %.2f degrees.\nNumerical aperture is %.3f.\nAcceptance angle is %.2f degree.",phi,NA,theta);
//answers in the book
//Critical angle is 80.56 degrees, deviation of 0.01.
//Numerical aperture is 0.244, deviation of 0.002.
//Acceptance angle is 14.17 degree, deviation of 0.14.
```

Chapter 4

Signal degradation in fibers

Scilab code Exa 4.3.1 input output ratio

```
// Example 4.3.1
clc;
clear;
L=10;          //fiber length in km
Pin=150d-6;        //input power
Pout=5d-6;         //output power
len=20;          //length of optical link
interval=1;        //splices after interval of 1 km
l=1.2;         //loss due to 1 splice
attenuation=10*log10(Pin/Pout);
alpha=attenuation/L;
attenuation_loss=alpha*20;
splices_loss=(len-interval)*l;
total_loss=attenuation_loss+splices_loss;
power_ratio=10^(total_loss/10);
printf("\nSignal attenuation is %.2f dBs.\nSignal attenuation is %.3f dB/Km.\nTotal loss in 20 Km fiber is %.2f dbs.\nTotal attenuation is %.2f dBs.\ninput/output ratio is %e.",attenuation,alpha,attenuation_loss,total_loss,power_ratio);
printf("\nAs signal attenuation is approximately
```

```
equal to 10^5, we can say that line is very lossy.");
```

Scilab code Exa 4.6.1 attenuation

```
// Example 4.6.1
clc;
clear;
beta_c=8d-11;          //isothermal compressibility
n=1.46;      //refractive index
P=0.286;     //photoelastic constat
k=1.38d-23; //Boltzmnn constant
T=1500;      //temperature
L=1000;      //length
lamda=1000d-9;  //wavelength
gamma_r = 8*(3.14^3)*(P^2)*(n^8)*beta_c*k*T/(3*(lamda^4));      //computing coefficient
attenuation=%e^(-gamma_r*L);        //computing attenuation
printf("\nAttenuation due to Rayleigh scattering is %.3f.",attenuation);
```

Scilab code Exa 4.6.2 attenuation

```
// Example 4.6.2
clc;
clear;
beta_c=7d-11;          //isothermal compressibility
n=1.46;      //refractive index
P=0.29;     //photoelastic constat
k=1.38d-23; //Boltzmnn constant
T=1400;      //temperature
L=1000;      //length
```

```
lamda=0.7d-6;   //wavelength
gamma_r = 8*(3.14^3)*(P^2)*(n^8)*beta_c*k*T/(3*(
    lamda^4));       //computing coefficient
attenuation=%e^(-gamma_r*L);         //computing
    attenuation
gamma_r=gamma_r*1000;
printf("\nRaleigh Scattering corfficient is %.3f *
    10^-3 per meter\n",gamma_r);
printf("\nNOTE - in quetion they have asked for
    attenuation but in solution they have not
    calcualted\n");
printf("\nAttenuation due to Rayleigh scattering is
    %.3f",attenuation);
//answer for Raleigh Scattering corfficient in the
    book is given as 0.804d-3, deviation of 0.003d-3
```

Scilab code Exa 4.7.1 threshold power

```
// Example 4.7.1
clc;
clear;
d=5;      //core diameter
alpha=0.4;   //attenuation
B=0.5;     //Bandwidth
lamda=1.4;    //wavelength
PB=4.4d-3*d^2*lamda^2*alpha*B;       //computing
    threshold power for SBS
PR=5.9d-2*d^2*lamda*alpha;       //computing
    threshold power for SRS
PB=PB*10^3;
PR=PR*10^3;
printf("\nThreshold power for SBS is %.1f mW.\
    nThreshold power for SRS is %.3f mW.",PB,PR);
printf("\nNOTE - Calculation error in the book while
     calculating threshold for SBS.\nAlso, while
```

```
calculating SRS, formula is taken incorrectly ,
   Bandwidth is multiplied in second step , which is
   not in the formula.");
//Calculation error in the book while calculating
   threshold for SBS. Also, while calculating SRS,
   formula is taken incorrectly ,Bandwidth is
   multiplied in second step , which is not in the
   formula
//answers in the book
//PB=30.8mW
//PR=0.413mW
```

Scilab code Exa 4.8.1 critical radius

```
// Example 4.8.1
clc;
clear;
n1=1.5;      //refractive index of core
delta=0.03/100;      //relative refractive index
lamda=0.82d-6;      //wavelength
n2=sqrt(n1^2-2*delta*n1^2);      //computing
   cladding refractive index
Rc=(3*n1^2*lamda)/(4*3.14*(n1^2-n2^2)^1.5);      //
   computing critical radius
Rc=Rc*10^3;
printf("\nCritical radius is %.1f micrometer.",Rc);
//answer in the book is 9 micrometer, deviation of
   0.1 micrometer.
```

Scilab code Exa 4.8.2 singal mode and multi mode

```
// Example 4.8.2
clc;
```

```
clear;
n1=1.45;      //refractive index of core
delta=3/100;      //relative refractive index
lamda=1.5d-6;      //wavelength
a=5d-6;      //core radius
n2=sqrt(n1^2-2*delta*n1^2);      //computing
    cladding refractive index
Rc=(3*n1^2*lamda)/(4*3.14*(n1^2-n2^2)^0.5);      //
    computing critical radius for single mode
Rc=Rc*10^6;
printf("\nCritical radius is %.2f micrometer",Rc);
lamda_cut_off= 2*3.14*a*n1*sqrt(2*delta)/2.405;
RcSM= (20*lamda/(n1-n2)^1.5)*(2.748-0.996*lamda/
    lamda_cut_off)^-3;      //computing critical
    radius for single mode
RcSM=RcSM*10^6;
printf("\nCritical radius for single mode fiber is %
    .2f micrometer.",RcSM);
printf("\nNOTE - Calculation error in the book.\n
    (2.748-0.996*lamda/lamda_cut_off)^-3; in this
    term raised to -3 is not taken in the book.");
//Calculation error in the book.(2.748-0.996*lamda/
    lamda_cut_off)^-3; in this term raised to -3 is
    not taken in the book.
//answer in the book is 7.23mm.(incorrect)
```

Scilab code Exa 4.13.1 material dispersion

```
// Example 4.13.1
clc;
clear;
lamda=1550d-9;
lamda0=1.3d-6;
s0=0.095;
Dt=lamda*s0/4*(1-(lamda0/lamda)^4);      //computing
```

```
material dispersion
Dt=Dt*10^9;
printf("\nMaterial dispersion at 1550 nm is %.1f ps/
nm/km",Dt);
printf("\n\nNOTE - Slight deviation in the answer
because of printig mistake\nIn problem they have
given lamda0 as 1300 nanometer \nbut while
solving they have taken it as 1330 nanometer");
//answer in the book 15.6 ps/nm/km, deviaton due to
printing mistake.
```

Scilab code Exa 4.14.1 bandwidth

```
// Example 4.14.1
clc;
clear;
tau=0.1d-6;      //pulse broadning
dist=20d3;       //distance
Bopt=1/(2*tau);     //computing optical bandwidth
Bopt=Bopt*10^-6;
dispertion=tau/dist;         //computing dispersion
dispertion=dispertion*10^12;
BLP=Bopt*dist;      //computing Bandwidth length
product
BLP=BLP*10^-3;
printf("\noptical bandwidth is %d MHz.\nDispersion
per unit length is %d ns/km.\nBandwidth length
product is %d MHz.km.",Bopt,dispertion,BLP);
```

Scilab code Exa 4.15.1 rms pulse

```
// Example 4.15.1
clc;
```

```
clear;
RSW=0.0012;      //relative spectral width
lamda=0.90d-6;   //wavelength
L=1;          //distance in km (assumed)
P=0.025;      //material dispersion parameter
c=3d5;        //speed of light in km/s
M=10^3*P/(c*lamda);      //computing material dispersion
sigma_lamda=RSW*lamda;
sigmaM=sigma_lamda*L*M*10^7;         //computing RMS pulse broadning
sigmaB=25*L*M*10^-3;
printf("\nMaterial dispersion parameter is %.2f ps/nm/km.\nRMS pulsr broadning when sigma_lamda is 25 is %.1f ns/km.\nRMS pulse broadning is %.1f ns/km.",M,sigmaB,sigmaM);
//answer in the book for RMS pulse broadning is 0.99 ns/km, deviation of 0.01ns/km.
```

Scilab code Exa 4.17.1 maximum bit rate

```
// Example 4.17.1
clc;
clear;
L=10;          //length of optical link
n1=1.49        //refractive index
c=3d8;       //speed of light
delta=1/100;          //relative refractive index
delTS=L*n1*delta/c;      //computing delay difference
delTS=delTS*10^12;
sigmaS=L*n1*delta/(2*sqrt(3)*c);     //computing rms pulse broadning
sigmaS=sigmaS*10^12;
B=1/(2*delTS);         //computing maximum bit rate
B=B*10^3;
```

```
B_acc=0.2/(sigmaS);      //computing accurate bit
    rate
B_acc=B_acc*10^3;
BLP=B_acc*L;       //computing Bandwidth length
    product
printf("\nDelay difference is %d ns.\nRMS pulse
    broadning is %.1f ns.\nBit rate is %.1f Mbit/s.\
    nAccurate bit rate is %.3f Mbits/s.\nBandwidth
    length product is %.1f MHz.km",delTS,sigmaS,B,
    B_acc,BLP);
//answer for maximum bit rate is given as 1.008 Mb/s
    , deviation of 0.008 Mb/s.
```

Chapter 5

optical fiber connection splicing

Scilab code Exa 5.2.1 loss

```
// Example 5.2.1
clc;
clear;
n1=1.47;          //refractive index of fiber
n=1;          //refractive index of air
r=((n1-n)/(n1+n))^2;     //computing fraction of light reflected
loss=-10*log10(1-r);     //loss
total_loss=2*loss;
printf("r = %.3f, which means %.1f percent of the transimitted light is reflected at one interface",r,r*100);
printf("\nTotal loss is %.3f dB",total_loss);
//answer in the book for total loss of fiber is 0.318 dB, deviation of 0.002
```

Scilab code Exa 5.2.2 loss

```
// Example 5.2.2
clc;
clear;
n1=1.47;          //refractive index of fiber
n=1;         //refractive index of air
d=40d-6;     //core diameter
y=4d-6;      //lateral dispalcement
a=d/2;       //computing core radius
eta_lateral = (16*(n1/n)^2)/(%pi*(1+(n1/n))^4)*(2*acos(y/(2*a))-(y/a)*(1-(y/(2*a))^2)^0.5);     //computing eta_lateral with air gap
loss=-10*log10(eta_lateral);         //computing loss when air gap is present
eta_lateral1=(2*acos(y/(2*a))-(y/a)*(1-(y/(2*a))^2)^0.5)/%pi;        //computing eta_lateral without air gap
loss1=-10*log10(eta_lateral1);       //computing loss when air gap is not present
printf("\nloss with air gap is %.2f dB.\nloss with no air gap is %.2f dB.\n Thus we can say that loss reduces considerably if there is no air gap.",loss,loss1);
//answer in the book for loss with air gap is 0.91dB, deviation of 0.01dB.
```

Scilab code Exa 5.2.3 loss

```
// Example 5.2.3
clc;
clear;
n1=1.48;          //refractive index of fiber
n=1;         //refractive index of air
theta=10;    //angle in degree
NA1=0.3;
NA2=0.6
```

```
eta_angular1= (16*(n1/n)^2)/((1+(n1/n))^4)*(1-((n*
   theta*%pi/180)/(%pi*NA1)));    //computing eta
   angular
eta_angular2= (16*(n1/n)^2)/((1+(n1/n))^4)*(1-((n*
   theta*%pi/180)/(%pi*NA2)));    //computing eta
   angular
loss1=-10*log10(eta_angular1);       //computing loss
loss2=-10*log10(eta_angular2);       //computing loss
printf("\nLoss when NA is %.1f is %.2f dB.\nLoss
   when NA is %.1f is %.2f dB.",NA1,loss1,NA2,loss2)
   ;
printf("\nThus we can say that insertion loss is
   considerably reduced with higher NA.");
```

Scilab code Exa 5.4.1 loss

```
// Example 5.4.1
clc;
clear;
d=1d-6;       //lateral displacement
W=4.95d-6;    //MFD
Lsm_lat= -10*log10(%e^(-(d/W)^2));       //computing
   loss
printf("\nInsertion loss is %.2f dB.",Lsm_lat);
```

Scilab code Exa 5.4.2 angular misalignment loss

```
// Example 5.4.2
clc;
clear;
lamda=1.3d-6;    //wavelength
theta=1;         //angle in degree
n2=1.465;        //cladding refractive index
```

```
W=4.95d-6;        //MFD
Lsm_ang= -10*log10(%e^(-(%pi*n2*W*(theta*%pi/180)/
   lamda)^2));       //computing loss
printf("\nInsertion loss is %.2f dB.",Lsm_ang);
```

Scilab code Exa 5.6.1 split ratio

```
// Example 5.6.1
clc;
clear;
p1=50d-6;
p2=0.003d-6;
p3=25d-6;
p4=26.5d-6
EL=10*log10(p1/(p3+p4));        //computing excess
   loss
IL13=10*log10(p1/p3);      //computing insertion loss
IL14=10*log10(p1/p4);      //computing insertion loss
ct=10*log10(p2/p1);      //computing cross talk
sr=(p3/(p3+p4))*100;     //computing split ratio
printf("\nExcess loss is %.2f dB.\nInsertion loss
   from port 1 to port 3 is %.2f dB.\nInsertion loss
    from port 1 to port 4 is %.2f dB.\ncross talk is
    %.2f dB.\nSplit ratio is %.2f percent",EL,IL13,
   IL14,ct,sr  );
printf("\nNOTE - calculation error in the book.\n
   Minus sign is not printed in the answer of excess
    loss.\nP1 is taken 25 instead of 50 while
   calculating cross talk.");
//calculation error in the book.Minus sign is not
   printed in the answer of excess loss.P1 is taken
   25 instead of 50 while calculating cross talk.
//answers in the book with slight deviations
//Excess loss is 0.12 dB.(printing error)
//Insertion loss from port 1 to port 4 is 2.75 dB.
```

```
//cross talk is −39.2 dB. (calculation error)
```

Scilab code Exa 5.6.2 average insertion loss

```
// Example 5.6.2
clc;
clear;
N=16;        //Number of ports
Pin=1d-3;    //input power
Pout=12d-6; //output power
split_loss=10*log10(N);      //computing split loss
excess_loss=10*log10(Pin/(Pout*N));      //computing
    excess loss
total_loss=split_loss+excess_loss;       //computing
    total loss
insertion_loss= 10*log10(Pin/Pout);      //computing
    insertion loss
printf("\nTotal loss is %.2f dB.\nInsertion loss is
    %.2f dB.",total_loss,insertion_loss);

//answer in the book for Total loss is 19.14,
    deviation of 0.06dB.
//answer in the book for insertion loss is 19.20,
    deviation of 0.01dB.
```

Chapter 6

optical sources

Scilab code Exa 6.3.1 operating wavelength

```
// Example 6.3.1
clc;
clear;
x=0.07;
Eg=1.424+1.266*x+0.266*x^2;
lamda=1.24/Eg;      //computing wavelength
printf("\nWavlength is %.3f micrometer.",lamda);
```

Scilab code Exa 6.3.2 longitudinal modes

```
// Example 6.3.2  page 6.12

clc;
clear;

n=1.7;      //refractive index
L=5d-2;     //distance between mirror
c=3d8;      //speed of light
```

```
lamda=0.45d-6;   //wavelength

k=2*n*L/lamda;       //computing number of modes
delf=c/(2*n*L);      //computing mode separation
delf=delf*10^-9;

printf("\nNumber of modes are %.2e.\nFrequency
   separation is %.2f GHz.",k,delf);
```

Scilab code Exa 6.7.1 power

```
// Example 6.7.1
clc;
clear;
tr=50;        //radiative recombination lifetime
tnr=85;       //non-radiative recombination lifetime
h=6.624d-34;   //plank's constant
c=3d8;        //speed of light
q=1.6d-19;    //charge of electron
i=35d-3;      //current
lamda=0.85d-6;       //wavelength
t=tr*tnr/(tr+tnr);            //computing total
   recombination time
eta=t/tr;                     //computing internal
   quantum efficiency
Pint=eta*h*c*i/(q*lamda);    //computing internally
   generated power
Pint=Pint*10^3
printf("\nTotal recombinaiton time is %.2f ns.\
   nInternal quantum efficiency is %.3f.\nInternally
    generated power is %.1f mW.",t,eta,Pint);
//answer in the book for Internal quantum efficiency
    is 0.629, deviation of 0.001.
//answer in the book for Internally generated power
   is 32.16 mW, deviation of 0.04 mW.
```

Scilab code Exa 6.8.1 bandwidth

```
// Example 6.8.1
clc;
clear;
f1=10d6;      //frequency
f2=100d6
t=4d-9;
Pdc=280d-6;      //optincal output power
w1=2*%pi*f1;     //computing omega
Pout1=Pdc*10^6/(sqrt(1+(w1*t)^2));       //computing output power
w2=2*%pi*f2;     //computing omega
Pout2=Pdc*10^6/(sqrt(1+(w2*t)^2));       //computing output power
printf("Ouput power at 10 MHz is %.2f microwatt.\nOuput power at 100 MHz is %.2f microwatt.\nConclusion when device is drive at higher frequency the optical power reduces.\nNOTE - calculation error. In the book square term in the denominater is not taken.",Pout1,Pout2);
BWopt = sqrt(3)/(2*%pi*t);
BWelec = BWopt/sqrt(2);
BWopt=BWopt*10^-6;
BWelec=BWelec*10^-6;
printf("\n3 dB optical power is %.2f MHz.\n3 dB electrical power is %.2f MHz.",BWopt,BWelec);
//calculation error. In the book square term in the denominater is not taken.
//answers in the book -
//Ouput power at 10 MHz is 228.7 microwatt.(incorrect)
//Ouput power at 100 MHz is 175 microwatt.(incorrect)
```

```
//3 dB optical power is 68.8 MHz, deviation of 0.12
//3 dB electrical power is 48.79 MHz, deviation of 0.06
```

Scilab code Exa 6.8.2 power

```
// Example 6.8.2
clc;
clear;
n1=3.5;  //refractive index
n=1;     //refractive index of air
F=0.69; //transmission factor
eta = 100*(n1*(n1+1)^2)^-1;      //computing eta
printf("\neta external is %.1f percent i.e. small fraction of intrnally generated opticalpower is emitted from the device.",eta);
printf("\n\n OR we can also arrive at solution,\n");
r= 100*F*n^2/(4*n1^2);       //computing ratio of Popt/Pint
printf("\n Popt/Pint is %.1f percent",r);
printf("\nNOTE - printing mistake at final answer.\nThey have printed 40 percent it should be 1.4 percent");
```

Scilab code Exa 6.8.3 operating lifetime

```
// Example 6.8.3

clc;
clear;

beta0=1.85d7;
T=293;       //temperature
```

```
k=1.38d-23; //Boltzman constant
Ea=0.9*1.6d-19;
theta=0.65; //thershold

betar=beta0*%e^(-Ea/(k*T));
t=-log(theta)/betar;

printf("\nDegradation rate is %.2e per hour.\
   nOperating lifetime is %.1e hour.",betar,t);

//answer in the book for Degradation rate is 6.4e-09
    per hour, deviation of 0.08e-9
//answer in the book for Operating lifetime is 6.7e
   +07 hour, deviaiton of 0.1e1
```

Chapter 7

Source to Fiber Power Launching and Photodetectors

Scilab code Exa 7.2.1 Fresenel Reflection and Power loss

```
// Example 7.2.1
clc;
clear;
n1=3.4;     //refractive index of optical source
n=1.46;      //refractive index of silica fiber
r=((n1-n)/(n1+n))^2;     //computing Frensel
    reflection
L=-10*log10(1-r);       //computing loss
printf("\nFrensel reflection is %.3f.\nPower loss is
     %.2f dB.",r,L);
```

Scilab code Exa 7.2.2 optical power

```
// Example 7.2.2
clc;
clear;
```

```
r=35d-6;          //radius
R=150;        //Lambertian emission pattern
NA=0.2;       //Numerical aperture
Pled= %pi^2*r^2*R*NA^2;
Pled=Pled*10^7;
printf("\nOptical power for larger core of 35
   micrometer is %.3f mW.",Pled);
r1=25d-6;
Pled1=(r1/r)^2*Pled;
printf("\nOptical power for smaller core of 25
   micrometer is %.2f mW.",Pled1);
```

Scilab code Exa 7.2.3 optical power

```
// Example 7.2.3
clc;
clear;
r=25d-6;          //radius
R=39;         //Lambertian emission pattern
NA=0.25;      //numerical aperture
a=35d-6;      //area
Pc1= %pi^2*a^2*R*NA^2;   //computing coupled power
   when r<a
Pc1=Pc1*10^7;
Pc= %pi^2*r^2*R*NA^2;    //computing coupled power
   when r>a
Pc=Pc*10^7;
printf("\nOptical power when r>a is %.2f mW.\
   nOptical power when r<a is %.3f mW.",Pc,Pc1);
```

Scilab code Exa 7.5.1 wavelength

```
// Example 7.5.1
```

```
clc;
clear;
h=6.626d-34;     //plank's constant
c=3d8;        //speed of light
e=1.6d-19;   //charge of electron
q=1.43;       //Bandgap energy
lamda=h*c/(q*e)*10^9;     //computing wavelength
printf("\nWavelength is %d nm",lamda);
printf("\nThis proves that photodiode will not
   operate for photon of wavelength greater than %d
   nm.",lamda);

//answer in the book 868nm; deviation of 1nm
```

Scilab code Exa 7.5.2 photocurrent

```
// Example 7.5.2
clc;
clear;
R=0.6;        //responsivity
Pin=15;       //optical power in microwatt
Ip=R*Pin;         //computing photocurrent
printf("\nPhotocurrent generated is %d microAmpere."
   ,Ip);
```

Scilab code Exa 7.5.3 responsivity

```
// Example 7.5.3
clc;
clear;
lamda1=1300d-9;
lamda2=1600d-9;
h=6.625d-34;     //plank's constant
```

```
c=3d8;       //speed of light
q=1.6d-19;   //charge of electron
eta=90/100; //quantum efficiency
E=0.73;       //energy gap in eV
R1=eta*q*lamda1/(h*c);
R2=eta*q*lamda2/(h*c);
lamdac=1.24/E;
printf("\nResponsivity at 1300nm is %.2f A/W.\nResponsivity at 1600nm is %.2f A/W.\nCut-off wavelength is %.1f micrometer.",R1,R2,lamdac);

//R1 is calculated as 0.92 in the book, deviation of 0.02.
```

Scilab code Exa 7.5.4 responsivity

```
// Example 7.5.4
clc;
clear;
lamda=0.8d-6;
h=6.625d-34;    //plank's constant
c=3d8;       //speed of light
q=1.6d-19;   //charge of electron
ne=1.8d11;   //electrons collected
np=4d11;     //photons incident
eta=ne/np;       //computing quantum efficiency
R=eta*q*lamda/(h*c);    //computing responsivity
printf("\nResponsivity of photodiode at 0.8 micrometer is %.3f A/W.",R);

//answer in the book is 0.289. deviation of 0.001 A/W
```

Scilab code Exa 7.5.6 wavelength

```
// Example 7.5.6
clc;
clear;
h=6.626d-34;     //plank's constant
c=3d8;       //speed of light
q=1.6d-19;  //charge of electron
E=1.35;       //energy gap in eV
lamda=h*c/(q*E);    //computing wavelength
lamda=lamda*10^6;
printf("\nThe InP photodetector will stop operation above %.2f micrometer.",lamda);
printf("\nNOTE - calculation error in the book");
//calculation error in the book
//answer in the book 1.47 micrometer.(incorrect)
```

Scilab code Exa 7.5.7 wavelength responsivity and optical power

```
// Example 7.5.7
clc;
clear;
h=6.626d-34;     //plank's constant
c=3d8;       //speed of light
eta=65/100; //quantum efficiency
I=2.5d-6;      //photocurrent
E=1.5d-19;  //energy of photns
q=1.6d-19;  //charge of electron
lamda=h*c/E;    //computing wavelength
R=eta*q*lamda/(h*c);     //computing responsivity
Popt=I/R;        //computing optical power
lamda=lamda*10^6;
Popt=Popt*10^6;
printf("\nWavelength is %.3f micrometer.\
    nResponsivity is %.3f A/W.\nIncident optical
```

```
power required is %.1f microWatt.",lamda,R,Popt);
//answer of R(responsivity) in the book is calculated as 0.694 A/W, deviation of 0.001.
```

Scilab code Exa 7.5.8 quantum efficiency

```
// Example 7.5.8
clc;
clear;
ne=3.9d6;   //electrons collected
np=6d6;     //photons incident
eta=100*ne/np;  //computing efficiency
printf("\nQuantum efficiency is %d percent.",eta);
```

Scilab code Exa 7.8.1 drift time and capacitance

```
// Example 7.8.1
clc;
clear;
w=25d-6;      //width
v=1d5;        //velocity
r=40d-6;      //radius
eps=12.5d-13;
t=w/v;        //computing drift time
c=eps*3.14*(r)^2/w;      //computing junction capacitance
c=c*10^16;
printf("\nDrift time %.1e sec.\nJunction capacitance %.1f pf.",t,c);
printf("\nCalculation error in the book at the answer of drift time.");

```

```
//calculation error in drift time answer in the book
    is 25*10^-10. it should be 2.5*10^-10.
```

Scilab code Exa 7.8.2 response time

```
// Example 7.8.2
clc;
clear;
w=20d-6;     //width
v=4d4;       //velocity
t=w/v;       //computing drift time
BW=(2*%pi*t)^-1;         //computing bandwidth
rt=1/BW;     //computing response time
rt=rt*10^9;
printf("\nMaximum response time is %.1f ns.",rt);
printf("\nNOTE - Calculation error in the book.");
//Calculation error in the book, answer given is 6.2
   ns
```

Scilab code Exa 7.9.1 noise equivalent power and specific directivity

```
// Example 7.9.1
clc;
clear;
lamda=1.4d-6;
h=6.626d-34;     //plank's constant
c=3d8;       //speed of light
q=1.6d-19;   //charge of electron
eta=65/100;  //quantum efficiency
I=10d-9;     //current
NEP= h*c*sqrt(2*q*I)/(eta*q*lamda);
D=NEP^-1;
```

```
printf("\nNoise equivalent power is %.3e W.\nSpecific directivity is %.2e.",NEP,D);

//answers in the book for NEP is 7.683*10^-14, deviation of 0.04*10^-14.
//answers in the book for D is 13.01 *10^12, deviation of 0.11*10^12.
```

Scilab code Exa 7.9.2 shot noise and thermal noise

```
// Example 7.9.2
clc;
clear;
lamda=1300d-9;
h=6.626d-34;     //plank's constant
c=3d8;       //speed of light
q=1.6d-19;   //charge of electron
eta=90/100;  //quantum efficiency
P0=300d-9;   //optical power
Id=4;     //dark current
B=20d6;      //bandwidth
K=1.39d-23;  //Boltzman constant
T=298;       //temperature
R=1000;      //load resister
Ip= 10^9*eta*P0*q*lamda/(h*c);
Its=10^9*(2*q*B*(Ip+Id));
Its=sqrt(Its);
printf("\nrms shot noise current is %.2f nA.",Its);
It= 4*K*T*B/R;
It=sqrt(It);
printf("\nThermal noise is %.2e A.",It);
//answer given in book for shot noise is 1.34nA, deviation of 0.01nA.
//answer given in book for Thermal noise it is 1.81*10^-8 A, deviation of 0.01*10^-8.
```

Scilab code Exa 7.10.1 multiplication factor

```
// Example 7.10.1
clc;
clear;
lamda=0.85d-6;
h=6.626d-34;     //plank's constant
c=3d8;       //speed of light
q=1.6d-19;   //charge of electron
eta=75/100;  //quantum efficiency
P0=0.6d-6;   //incident optical power
Im=15d2;   //avalanche gain
R= eta*q*lamda/(h*c);   //computing responsivity
Ip=10^8*P0*R;        //computing photocurrent
Ip=floor(Ip);
M=Im/Ip;          //computing multiplication factor
printf("\nMultiplication factor is %d.",M);
```

Scilab code Exa 7.10.2 multiplication factor

```
// Example 7.10.3
clc;
clear;
lamda=900d-9;
h=6.626d-34;     //plank's constant
c=3d8;       //speed of light
q=1.6d-19;   //charge of electron
eta=65/100;  //quantum efficiency
P0=0.5d-6;   //incident optical power
Im=10d2;   //avalanche gain
R= eta*q*lamda/(h*c);   //computing responsivity
```

```
Ip=10^8*P0*R;         //computing photocurrent
M=Im/Ip;         //computing multiplication factor
printf("\nMultiplication factor is %d.",M);

//answer in the book is 41.7 deviation 0.3.
```

Scilab code Exa 7.10.3 multiplication factor

```
// Example 7.10.3
clc;
clear;
lamda=900d-9;
h=6.626d-34;     //plank's constant
c=3d8;       //speed of light
q=1.6d-19;   //charge of electron
eta=65/100;  //quantum efficiency
P0=0.5d-6;   //incident optical power
Im=10d2;    //avalanche gain
R= eta*q*lamda/(h*c);    //computong responsivity
Ip=10^8*P0*R;         //computing photocurrent
Ip=floor(Ip);
M=Im/Ip;         //computing multiplication factor
printf("\nMultiplication factor is %d.",M);
```

Chapter 8

optical receiver operation

Scilab code Exa 7.q maximum responce time

```
// Question 7  page 8.44
clc;
clear;
w=25d-6;     //width
v=3d4;       //velocity
t=w/v;       //computing drift time
BW=(2*%pi*t)^-1;        //computing bandwidth
rt=1/BW;     //response time
rt=rt*10^9;

printf("\nMaximum response time is %.2f ns.",rt);

//Answer in the book is given as 5.24ns deviation of
    0.01ns
```

Scilab code Exa 8.2.1 quantum limit

```
// Example 8.2.1
```

```
clc;
clear;
P=10^-9;     //probability of error
eta=1;       //ideal detector
h=6.626d-34 //plank's constant
c=3d8;      //speed of light
lamda=1d-6; //wavelength
B=10^7;      //bit rate

Mn= - log(P);
printf("\n The quantum imit at the receiver to maintain bit error rate 10^-9 is (%.1f*h*f)/eta.",Mn);
f=c/lamda
Popt= 0.5*Mn*h*f*B/eta;      //computing optical power
Popt_dB = 10 * log10(Popt) + 30;     //optical power in dbm
Popt=Popt*10^12;

printf("\nMinimum incident optical power is %.1f W or %.1f dBm.",Popt,Popt_dB);
```

Scilab code Exa 8.2.2 optical power

```
// Example 8.2.2
clc;
clear;
SN_dB=60;     //signal to noise ratio
h=6.626d-34 //plank's constant
c=3d8;      //speed of light
lamda=1.3d-6; //wavelength
eta=1;
B=6.5d6;      //Bandwidth
SN=10^(SN_dB/10);
```

```
f=c/lamda
Popt= 2*SN*h*f*B/eta;      //computing optical power
Popt_dB = 10 * log10(Popt) + 30;     //optical power in dbm
Popt=Popt*10^6;
printf("\nIncident power required to get an SNR of 60 dB at the receiver is %.4f microWatt or %.3f dBm",Popt,Popt_dB);
printf("\nNOTE - Calculation error in the book.\nThey have take SN as 10^5 while calculating, which has lead to an error in final answer");

//Calculation error in the book.They have take SN as 10^5 while calculating, which has lead to an error in final answer
//answer in the book 198.1nW and -37.71 dBm
```

Scilab code Exa 8.3.1 shot noise

```
// Example 8.3.1
clc;
clear;
lamda=0.85d-6;
h=6.626d-34;     //plank's constant
c=3d8;        //speed of light
q=1.6d-19;   //charge of electron
eta=65/100;  //quantum efficiency
P0=300d-9;   //optical power
Id=3.5;     //dark current
B=6.5d6;      //bandwidth
K=1.39d-23;  //Boltzman constant
T=293;       //temperature
R=5d3;      //load resister
Ip= 10^9*eta*P0*q*lamda/(h*c);
Its=10^9*(2*q*B*(Ip+Id));
```

```
Its=sqrt(Its);
printf("\nrms shot noise current is %.2f nA.",Its);
It= 4*K*T*B/R;
It=sqrt(It);
It=It*10^9;
printf("\nThermal noise is %.2f nA.",It);

//answer given in book for Thermal noise it is 4.58
   nA, deviation is 0.02nA.
```

Scilab code Exa 8.3.2 S N ratio

```
// Example 8.3.2
clc;
clear;
lamda=0.85d-6;
h=6.626d-34;     //plank's constant
c=3d8;       //speed of light
q=1.6d-19;   //charge of electron
eta=65/100;  //quantum efficiency
P0=300d-9;   //optical power
Id=3.5;     //dark current
B=6.5d6;      //bandwidth
K=1.39d-23;  //Boltzman constant
T=293;       //temperature
R=5d3;      //load resister
F_dB=3;      //noise figure
F=10^(F_dB/10);
Ip=10^9*eta*P0*q*lamda/(h*c);
Its=10^9*(2*q*B*(Ip+Id));
It1= 4*K*T*B*F/R;
SN= Ip^2/(Its+It1);
SN_dB=10*log10(SN);
SN=SN/10^4;
printf("\nSNR is %.2f*10^4 or %.2f dB.",SN,SN_dB);
```

```
//answer given in the book is 6.16*10^4 (deviation
   of 0.9) and 47.8dB (deviation of 0.16dB)
```

Scilab code Exa 8.4.1 bandwidth

```
// Example 8.4.1
clc;
clear;
Cd=7d-12;
B=9d6;
Ca=7d-12;
R=(2*3.14*Cd*B)^-1;
B1=(2*3.14*R*(Cd+Ca))^-1;
R=R/1000;
B1=B1/10^6;
printf("\nThus for 9MHz bandwidth maximum load
   resistance is %.2f Kohm\nNow if we consider input
    capacitance of following amplifier Ca then
   Bandwidth is %.2fMHz\nMaximum post detection
   bandwidth is half.",R,B1);

//answer for resistance in the book is 4.51Kohm,
   deviation of 0.01Kohm, while for bandwidth it is
   4.51 MHz, deviation of 0.01MHz
```

Chapter 9

link design

Scilab code Exa 9.4.1 power margin

```
// Example 9.4.1
clc;
clear;
output=13;       //laser output
sensitivity=-31;    //APD sensitivity
coupling_loss=0.5;
L=80;   //length in km
sl=0.1; //loss correspond to one splice in dB
fl=0.35;    //fiber loss in dB/km
noise=1.5;
allowed_loss=output-sensitivity;
splices_loss=(L-1)*sl;
fiber_loss=L*fl;
margin=allowed_loss-(splices_loss+fiber_loss+
   coupling_loss+noise);
printf("\nFinal margin is %.1f dB.",margin);
```

Scilab code Exa 9.6.1 maximum bit rate

```
// Example 9.6.1
clc;
clear;
L=10;
ts=10;
tD=8;
tmod=L*6;
tt=L*2;
Tsys=1.1*sqrt(ts^2+tmod^2+tt^2+tD^2);
Bt=0.7/Tsys;
Bt=Bt*10^3;
printf("Maximum bit rate for link using NRZ data format is %.2f Mbits/sec.",Bt);
printf("\nNOTE - calculation error in the book");
//calculation error in the book
//answer given in the book is 10.3mbits/sec.(incorrect)
```

Chapter 10

performance measurement and monitoring

Scilab code Exa 10.5.1 pulse broadning and optical bandwidth

```
// Example 10.5.1
clc;
clear;
To=12.6;      //width of output pulse
Ti=0.3;       //width of input pulse
l=1.2;        //length of measurement
Pulse_dispersion = sqrt(To^2 - Ti^2);    //computing
   pulse dispersion
PDKM=Pulse_dispersion/l;          //computing pulse
   dispersion per Kilometer
BW=0.44/PDKM;        //computing optical bandwidth
BW=BW*1000;
printf("\nPulse broadning is %.1f ns/km.\nOptical
   bandwidth is %.1f MHz.Km.",PDKM,BW);
```

Scilab code Exa 10.6.1 attenuation

```
// Example 10.6.1
clc;
clear;
V2=12;
V1=2.5;
L2=3;
L1=0.004;
alpha_dB = 10* log10(V2/V1)/(L2-L1);
un = 0.2/(L2-L1);
printf("\nAttenuation is %.2f dB/km\nUncertainity
    +/- %.3f dB.",alpha_dB,un);
//answer for attenuation in the book is 2.26
    deviation of 0.01 and for uncertaininty is 0.066
    deviation of 0.001
```