Scilab Textbook Companion for Optical Fiber Communication by V. S. Bagad[1]

Created by
Anurag Muglikar
B.E
Electronics Engineering
University of Pune
College Teacher
R.a. Chavan
Cross-Checked by
Lavitha Pereira and Mukul Kulkarni

May 24, 2016

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

Book Description

Title: Optical Fiber Communication

Author: V. S. Bagad

Publisher: Technical Publications, Pune

Edition: 2

Year: 2013

ISBN: 9789350385203

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

Fiber Optics Communications System

Scilab code Exa 1.7.1 To find angle of reflection

```
// Example 1.7.1  page 1.14
//To calculate the angel of refraction if the angle
    of incidence is 30

clc;
clear;
n1= 1.5; // for glass
n2= 1.33; // for water
phi1= (%pi/6);          // phi1 is the angel of
    incidence
// According to Snell's law...
// n1*sin(phi1)= n2*sin(phi2);
sinphi2= (n1/n2)*sin(phi1); // phi2 is the angle of
    refraction..
phi2 = asind(sinphi2);
printf(' The angel of refraction is %.2f degrees',
    phi2);
```

Scilab code Exa 1.7.2 To calculate Critical Angle

```
// Example 1.7.2  page 1.14
// To calculate the critical angel

clc;
clear;

n1= 1.50;    // RI of glass..
n2 = 1;      // RI of air...
// According to Snell's law...
// n1*sin(phi1)= n2*sin(phi2);

// From definition of critical angel phi2 = 90
   degrees and phi1 will be critical angel
phiC=asind((n2/n1)*sin(%pi/2));
printf('The Critical angel is  %.2f degrees',phiC);
```

Scilab code Exa 1.7.3 To find RI and Critical Angle of glass

```
// Example 1.7.3 page 1.15
//  To find RI of glass
// To find the critical angle for glass...

clc;
clear;
phi1 = 33   // Angle of incidence..
phi2 = 90   //Angle of refraction..
// According to Snell's law...
// n1*sin(phi1)= n2*sin(phi2);
//a = sin(phi1*%pi/(180));
// Assume n1 is the RI of glass and n2 is RI of air
```

```
n2= 1;
n1 = sind(90)/sind(33);
printf('The Refractive Index is  %.2f',n1);

// To calculate thre critical angle...
n1 =  1.836;     // From above rounded off to 3 decimal points...
phiC = asind((n2/n1)*sind(90));
phiC=asind(0.54);
printf('\n\nThe Critical angel is  %.2f degrees',phiC);
```

Scilab code Exa 1.7.4 To find the angle of refraction

```
// Example 1.7.4 page 1.15
// To find the angle of refraction..

clc;
clear;
n1= 1.5      // TheRi of medium 1
n2= 1.36     // the RI of medium 2
phi1= 30;    // The angle of incidence
// According to Snell's law...
// n1*sin(phi1)= n2*sin(phi2);
phi2 = asind((n1/n2)*sind(phi1));
printf('The angel of refraction is  %.2f degrees from normal',phi2);
```

Scilab code Exa 1.7.5 To estimate if TIR is possible or not

```
// Example 1.7.5 page 1.16
// Will total internal reflection take place?

```

```
clc;
clear;

n1 = 3.6;        // RI of GaAs..
n2 = 3.4;        // RI of AlGaAs..
phi1 = 80;       // Angle of Incidence..
// According to Snell's law...
// n1*sin(phi1)= n2*sin(phi2);
//At critical angle phi2 = 90...
phiC = asind((n2/n1)*sind(90));
printf('The Critical angel is  %.2f degrees',phiC);
printf('\n\nFor total internal reflection to take
   place angle\n of incidence should be greater than
    the critical angle. \nFrom the calculations, we
   can thus conclude that Total internal reflection
   will take place');
```

Scilab code Exa 1.9.1 To find NA Acceptance and Critical Angle

```
// Example 1.91 page 1.22
// To calculate Numerical Aperture (NA), Acceptance
   angle (phiA), critical Angle (phiC)...

clc;
clear;

n1= 1.5;      // RI of medium 1
n2 =1.45;     // RI of medium 2

del= (n1-n2)/n1;
NA = n1*(sqrt(2*del));
printf('The Numerical aperture is %.2f ',NA);
phiA = asind(NA);
printf('\n\nThe Acceptance angel is  %.2f degrees',
   phiA);
```

```

phiC = asind(n2/n1);
printf('\n\nThe Critical angel is %.2f degrees',
   phiC);
```

Scilab code Exa 1.9.2 To find NA and Acceptance angle

```
// Example 1.9.2 page 1.23
// To calculate Numerical aperture and Acceptance
   angle...

clc;
clear;

n1= 1.5 // RI of core
n2 = 1.48   // RI of cladding..

NA = sqrt((n1^2)-(n2^2));
printf('The Numerical Aperture is %.2f',NA);

phiA = asind(NA);
printf('\n\nThe Critical angel is %.2f degrees',
   phiA);
```

Scilab code Exa 1.9.3 To find RI of core and cladding

```
//Example 1.9.3 page 1.23
// To calculate RI of core and cladding..

clc;
clear;

NA = 0.35;          //Numerical Aperture
```

```
del = 0.01;
//NA= n1*(sqrt(2*del)       n1 is RI of core
n1 = 0.35/(sqrt(2*del));
printf('The RI of core is %.4f',n1);

// Numerical Aperture is also given by
// NA = sqrt(n1^2 - n2^2)    // n2 is RI of cladding
n2 = sqrt((n1^2-NA^2));
printf('\n\nThe RI of Cladding %.3f',n2);
```

Scilab code Exa 1.9.4 To find NA and acceptance angle

```
//Example 1.9.4    page 1.24


clc;
clear;
Vc = 2.01*10^8;       // velocity of light in core in
    m/sec...
phiC= 80;             // Critical angle in degrees...

// RI of Core (n1) is given by (Velocity of light in
    air/ velocity of light in air)...
n1= 3*10^8/Vc;
// From critical angle and the value of n1 we
    calculate n2...
n2 = sind(phiC)*n1; // RI of cladding...
NA = sqrt(n1^2-n2^2);
printf('The Numerical Aperture is %.2f',NA);
phiA = asind(NA);           // Acceptance angle...
printf('\n\nThe Acceptance angel is  %.2f degrees',
    phiA);
```

Scilab code Exa 1.9.5 To find NA acceptance and critical angle

```
// Example 1.9.5 page 1.25
// To calculate critical angle accepatance angle and numerical aperture..

clc;
clear;

n1 = 1.4;        //RI of Core..
n2 = 1.35;       //RI of Cladding

phiC = asind(n2/n1);             //Critical angle..
printf('The Critical angel is  %.2f degrees',phiC);

NA = sqrt(n1^2-n2^2);         // numerical Aperture...
printf('\n\nThe Numerical Aperture is %.2f',NA);

phiA = asind(NA);        // Acceptance angle...
printf('\n\nThe Acceptance angel is  %.2f degrees',phiA);
```

Scilab code Exa 1.9.6 To find NA and entrance angle

```
//Example 1.9.6 page 1.25
//To calculate The Numerical Aperture and maximum angle of entrance of light into air...

clc;
clear;
n1 = 1.48;       // RI of core..
n2 = 1.46;       // RI of Cladding..

NA = sqrt(n1^2-n2^2);         //Numerical Aperture..
printf('The Numerical Aperture is %.3f',NA);
```

```

theta = %pi*NA^2;          // The entrance angle theta
    ..
printf('\n\nThe Entrance angel is  %.3f degrees',
   theta);
```

Scilab code Exa 1.9.7 To find NA

```
//Example 1.9.7 page 1.26
//To find the Numerical Aperture...

clc;
clear;

del = 0.007;          // relative refractive index
    difference
n1 = 1.45;       // RI of core...
NA = n1* sqrt((2*del));
printf('The Numerical Aperture is %.4f',NA);
```

Scilab code Exa 1.9.8 To estimate relative RI

```
//Example 1.9.8 page 1.26
//To finf relative RI difference..

clc;
clear;

phiA = 8         // accepatance angle in degrees...
n1 =1.52;        //RI of core...

NA = sind(phiA);         //Numerical Aperture...

```

```
del = NA^2/(2*(n1^2));       //Relative RI difference ...
printf("The relative refractive index difference is %.5f",del);
```

Scilab code Exa 1.9.9 To find NA and solid acceptance angle

```
//Example 1.9.9 page 1.27
// Calculate NA and solid acceptance angle. Also find critical angle...

clc;
clear;

del = 0.01;      // relative RI difference..
n1 = 1.48;       // RI of core...

NA = n1*(sqrt(2*del));       //Numerical Aperture..
printf('The Numerical Aperture is %.3f',NA);

theta = %pi*NA^2;        //Solid Acceptance angle...
printf('\n\nThe Solid Acceptance angel is  %.4f degrees',theta);

n2 = (1-del)*n1;
phiC = asind(n2/n1);         //Critical Angle...
printf('\n\nThe Critical angel is  %.2f degrees',phiC);
printf("\n\nCritical angle wrong due to rounding off errors in trignometric functions..\n Actual value is 90.98 in book.");
```

Scilab code Exa 1.14.1 To calculate number of modes

```
//Example 1.14.1 page 1.41
// To calculate the number of modes...

clc;
clear;
d = 50*10^-6;          // diameter of fibre...
n1 = 1.48;        //RI of core..
n2 = 1.46;        //RI of cladding..
lamda = 0.82*10^-6;      //wavelength of light..

NA = sqrt(n1^2-n2^2);        // Numerical Aperture..
Vn= %pi*d*NA/lamda;      //normalised frequency...
M = Vn^2/2;      // number of modes...
printf(" The number of modes in the fibre are %d",M)
   ;
```

Scilab code Exa 1.14.2 To find the NA

```
//Example 1.14.2 page 1.42
//to find the Numerical aperture..

clc;
clear;
V = 26.6;       //Normalised frequency..
lamda = 1300*10^-9;      //wavelenght of operation
a = 25*10^-6;         // radius of fibre.
NA = V*lamda/(2*%pi*a);      //Numerical Aperture..
printf("The Numerical Aperture is %.3f",NA);
```

Scilab code Exa 1.14.3 To find the normalised frequence

```
//Example 1.14.3
// to calculate the normalise frequency..
```

```
clc;
clear;

a = 40*10^-6; //radius of core...
del = 0.015;    //relative RI difference..
lamda= 0.85*10^-6;      //wavelength of operation..
n1=1.48;          //RI of core..

NA = n1*sqrt(2*del);    //Numerical Aperture..
printf(" The Numerical Aperture is %.4f",NA);
V = 2*%pi*a*NA/lamda;   //normalised frequency
printf(" \n\nThe Normalised frequency is %.2f",V);

M = V^2/2;      //number of modes..
printf("\n\nThe number of modes in the fibre are %d",M);
```

Scilab code Exa 1.14.4 To find diameter of core and the number of modes

```
//Example 1.14.4 page 1.43
// to find diameter of core, number of modes at 1320, number of modes at 1550 um

clc;
clear;

NA = 0.20;      //Numerical Aperture..
M = 1000;    //number of modes..
lamda = 850*10^-9; // wavelength of operation..

a = sqrt(M*2*lamda^2/(%pi^2*NA^2)); // radius of core..
a=a*10^6; //converting in um for displaying...
printf("The radius of the core is %.2f um",a);
```

```
a=a*10^-6;
M1= ((%pi*a*NA/(1320*10^-9))^2)/2
printf("\n\nThe number of modes in the fibre at 1320
   um are %d",M1);
printf("\n\n***The number of modes in the fibre at
   1320um is calculated wrongly in book");
M2= ((%pi*a*NA/(1550*10^-9))^2)/2
printf("\n\nThe number of modes in the fibre at 1550
   um are %d",M2);
```

Scilab code Exa 1.14.5 To find acceptance and critical angle and the number of modes

```
//Example 1.14.5 page 1.44
//To find acceptance angle ; critical angle;number
   of modes..

clc;
clear;

NA = 0.2;    //Numerical Aperture..
n2= 1.59;   // RI of cladding..
n0= 1.33; //   RI of water..
lamda =  1300*10^-9;       // wavelength..
a = 25*10^-6;    // radius of core..
n1 = sqrt(NA^2+n2^2);    //RI of core..
phiA= asind(sqrt(n1^2-n2^2)/n0);      //Acceptance
   angle..
printf("The Acceptance angle is %.2f",phiA);

phiC= asind(n2/n1); // Critical angle..
printf("\n\nThe critical angle is %.2f",phiC);
V = 2*%pi*a*NA/lamda;    // normalisd frequency
M= V^2/2;         //number of modes
printf("\n\nThe number of modes in the fibre are %d"
```

```
    ,M);

printf("\n\n***The value of the angle differ from
    the book because of round off errors.");
```

Scilab code Exa 1.14.6 To find NA solid acceptance and number of modes

```
//Example 1.14.6 page 1.46
// To find Numerical Aperture, solid acceptance
    angle, and number of modes.

clc;
clear;

V= 26.6;    // Normalised frequency..
lamda= 1300*10^-9;       //wavelength of operation..
a= 25*10^-6;    // radius of core..

NA = V*lamda/(2*%pi*a);     //Numerical Aperture..
printf('The Numerical Aperture is %.2f',NA);
theta = %pi*NA^2;       //solid Acceptance Angle..
printf('\n\nThe solid acceptance angle is %.3f
    radians',theta);

M= V^2/2;       //number of modes..
printf("\n\nThe number of modes in the fibre are %.2
    f",M);
```

Scilab code Exa 1.14.7 To find cutoff wavelength and core diameter

```
//Example 1.14.7 page 1.47
// Cutoff wavelength, MAx core diameter for single
    mode operation..
```

```
clc;
clear;

n1= 1.49; //     RI of core.
n2=1.47;        //RI of cladding..
a= 2;      //radius of core in um..
NA= sqrt(n1^2-n2^2);     // Numerical Aperture..
// The maximum V number for single mode operation is
    2.4...
V= 2.4;      //Normalised frequency..

lamda = 2*%pi*a*NA/V;        // Cutoff wavelength...
printf('The cutoff wavelength is %.2f um',lamda);


lamda1 = 1.310; // Givenn cutoff wavelength in um..
d= V*lamda1/(%pi*NA);       // core diameter..
printf('\n\nThe core diameter is %.2f um',d);
```

Scilab code Exa 1.14.8 To find the cutoff wavelength

```
//Example 1.14.8 page 1.47
//To find cutoff wavelength..

clc;
clear;
 n1= 1.48;  //RI of core..
 a= 4.5;    //core radius in um..
del= 0.0025;     //Relative RI difference..
V= 2.405;        //For step index fibre..
lamda= (2*%pi*a*n1*sqrt(2*del))/V;       //cutoff
   wavelength..
printf('The cutoff wavelength is %.2f um ',lamda);
```

Scilab code Exa 1.14.9 To find normalised frequency and number of modes

```
//Example 1.14.9
//To find normalised frequency and the number of modes for the fibre..

clc;
clear;

lamda= 0.82*10^-6;        //wavelength ofoperation.
a= 2.5*10^-6;        //Radius of core..
n1= 1.48;        //RI of core..
n2= 1.46;    //RI of cladding
NA= sqrt(n1^2-n2^2);        //Numerical Aperture..
V= 2*%pi*a*NA/lamda;        //Normalisd frequency..
printf('The normalised frequency is %.3f',V);
M= V^2/2;    //The number of modes..
printf("\n\nThe number of modes in the fibre are %.2f",M);
```

Scilab code Exa 1.14.10 To find diameter of core

```
//Example 1.14.10 page 1.49
//To find the diameter of the core..

clc;
clear;

del= 0.01;        //Relative RI difference..
n1= 1.5;
M= 1100;        //Number of modes...
lamda= 1.3;     //wavelength of operation in um..
```

```
V= sqrt(2*M);          //Normalised frequency...
d= V*lamda/(%pi*n1*sqrt(2*del));     //diameter of core..
printf('The diameter of the core is %.2f um',d);
```

Scilab code Exa 1.14.11 To find NA solid acceptance angle and number of modes

```
//Example 1.14.11
//To find Numerical Aperture,Solid Acceptance angle,Normalised frequency,Number of modes..

clc;
clear;

n1= 1.5;     // RI of core..
n2= 1.38;    //RI of cladding..
a= 25*10^-6;     //radius of core..
lamda= 1300*10^-9;      // wavelength of operation...
NA= sqrt(n1^2-n2^2);     //Numerical Aperture..
printf('The Numerical Aperture of the given fibre is %.4f',NA);
V= 2*%pi*a*NA/lamda;     //Normalised frequency..
printf('\n\nThe normalised frequency is %.2f',V);

theta= asind(NA);         //Solid acceptance anglr..
printf('\n\nThe Solid acceptance angle is %d degrees',theta);
M= V^2/2;         //Number of modes..
printf("\n\nThe number of modes in the fibre are %d",M);
printf("\n\n***Number of modes wrongly calculated in the book..");
```

Scilab code Exa 1.14.12 To find normalised frequency and number of modes

```
//Example 1.14.12
//To find noramlised frequency and number of modes

clc;
clear;

lamda= 850*10^-9;        //wavelength of operation.
a= 25*10^-6;         //Radius of core
n1= 1.48;        //RI of Core...
n2= 1.46;        //RI of cladding..

NA= sqrt(n1^2-n2^2);     //Numerical Aperture

V= 2*%pi*a*NA/lamda;     //Normalised frequency..
printf('The normalised frequency is %.2f',V);

lamda1= 1320*10^-9;      // wavelength changed...
V1= 2*%pi*a*NA/lamda1;   //Normalised frequency at
    new wavelength..

M= V1^2/2;       //Number of modes at new wavelength
    ..
printf("\n\nThe number of modes in the fibre at 1320
    um are %d",M);
lamda2= 1550*10^-9;      //wavelength 2...
V2= 2*%pi*a*NA/lamda2;   //New normalised frequency..
M1= V2^2/2; // number of modes..
printf("\n\nThe number of modes in the fibre at 1550
    um are %d",M1);
```

Scilab code Exa 1.15.1 To calculate maximum core diameter

```
//Example 1.15.1 page 1.56..
// Maximum core diameter..

clc;
clear;
n1= 1.48;          //RI of core..
del= 0.015;        //relative RI differencr..
lamda= 0.85;    //wavelength of operation..
V= 2.4; // for single mode of operation..

a= V*lamda/(2*%pi*n1*sqrt(2*del));  //radius of core
    ..
printf('The raduis of core is %.2f um',a);
printf('\n\nThe maximum possible core diameter is %
    .2f um',2*a);
```

Scilab code Exa 1.15.2 To find core diameter

```
//Example 1.15.2
// to find maximum core diameter for single mode..

clc;
clear;

n1= 1.5;     //RI of core..
del= 0.01;    //Relative RI difference...
lamda= 1.3;    //Wavelength of operation...
V= 2.4*sqrt(2); // Maximum value of V for GRIN...
a= V*lamda/(2*%pi*n1*sqrt(2*del));  //radius of core
    ..
printf('The radius of core is %.2f um',a);
printf('\n\nThe maximum possible core diameter is %
    .2f um',2*a);
```

Scilab code Exa 1.15.3 To find the cutoff wavelength

```
//Example 1.15.3
//To find the cutoff wavelength..

clc;
clear;

n1= 1.46;         //RI of core..
a = 4.5;     //radius of core in um..
del= 0.0025;     //relative RI difference..
V= 2.405;    // Normalisd frequency for single mode..
lamda= 2*%pi*a*n1*sqrt(2*del)/V;     //cutoff
   wavelength...
printf('The cut off wavelength for the given fibre
   is %.3f um',lamda);
```

Chapter 2

Optical Fiber for Telecommunication

Scilab code Exa 2.2.1 Computation for length for different conditions

```
// Example 2.2.1 page 2.4

clc;
clear;

alpha= 3;   // average loss     Power decreases by
   50% so P(0)/P(z)= 0.5
lamda= 900*10^-9;   //wavelength
z= 10*log10(0.5)/alpha;   //z is the length
z= z*-1;
printf("The length over which power decreases by 50
   %% is =%.2f Kms",z);

z1= 10*log10(0.25)/alpha;     //Power decreases by
   75% so P(0)/P(z)= 0.25
z1=z1*-1;   //as distance cannot be negative...
printf("\n\nThe length over which power decreases by
    75%% is =%.2f Kms",z1);
```

Scilab code Exa 2.2.2 To find the output power

```
//Example 2.2.2 page 2.5

clc;
clear;

z=30;    //Length of the fibre in kms
alpha= 0.8;   //in dB
P0= 200;     //Power launched in uW
pz= P0/10^(alpha*z/10);
printf("The output power is:%.4f uW",pz);
```

Scilab code Exa 2.2.3 To find overall signal attenuation

```
//Example 2.2.3 page 2.6

clc;
clear;

z=8;     //fibre length
p0= 120*10^-6;   //power launched
pz= 3*10^-6;
alpha= 10*log10(p0/pz); // overall attenuation
printf("The overall attenuation is %.2fdB",alpha);
alpha = alpha/z;     // attenuation per km
alpha_new= alpha *10;    // attenuation for 10kms
total_attenuation = alpha_new + 9;   //9dB because of
     splices
printf("\n\nThe total attenuation is : %d dB",
   total_attenuation);
```

Scilab code Exa 2.2.4 To find minimum optical input power

```
//Example 2.2.4 page 2.6

clc;
clear;
z=12;    //fibre length
alpha = 1.5;
p0= 0.3;
pz= p0/10^(alpha*z/10);
pz=pz*1000;   //formatting pz in nano watts...
printf("The power at the output of the cable is:%.2fx10^-9 W",pz);
alpha_new= 2.5;
pz=pz/1000; //pz in uWatts...
p0_new= 10^(alpha_new*z/10)*pz;
printf("\n\nThe Input power is %.2f uW",p0_new);
```

Scilab code Exa 2.2.5 To calculate fiber attenuation in different cases

```
//Example 2.2.5 page 2.7

clc;
clear;
p0=150*10^-6;     //power input
z= 10;  //fibre length in km
pz= -38.2; // in dBm...
pz= 10^(pz/10)*1*10^-3;
alpha_1= 10/z *log10(p0/pz);        //attenuation in 1st window
printf("Attenuation is 1st window is %.2f dB/Km",alpha_1);
```

```
alpha_2= 10/z *log10(p0/(47.5*10^-6));         //attenuation in 2nd window
printf("\n\nAttenuation is 2nd window is %.2f dB/Km",alpha_2);
alpha_3= 10/z *log10(p0/(75*10^-6));         //attenuation in 3rd window
printf("\n\nAttenuation is 3rd window is %.2f dB/Km",alpha_3);
```

Scilab code Exa 2.2.6 To calculate length of the fiber

```
//Example 2.2.6 page 2.8

clc;
clear;

p0=3*10^-3;
pz=3*10^-6;
alpha= 0.5;
z= log10(p0/pz)/(alpha/10);
printf("The Length of the fibre is %.f Km",z);
```

Scilab code Exa 2.2.7 To find overall signal attenuation

```
//Example 2.2.7 page 2.9

clc;
clear;
z= 10;
p0= 100*10^-6;   // input power
pz=5*10^-6;      //output power
alpha = 10*log10(p0/pz);     //total attenuation
```

```
printf("The overall signal attenuation is %.2f dB",alpha);
alpha = alpha/z;    // attenuation per km
printf("\n\nThe attenuation per Km is %.2f dB/Km",alpha);
z_new = 12;
splice_attenuation = 11*0.5;
cable_attenuation = alpha*z_new;
total_attenuation = splice_attenuation+cable_attenuation;
printf("\n\nThe overall signal attenuation for 12Kms is %.1f dB",total_attenuation);
```

Scilab code Exa 2.2.8 To find attenuation per km

```
//Example 2.2.8 page 2.15

clc;
clear;

Tf = 1400;    //fictive temperature
BETA = 7*10^-11;
n= 1.46;        //RI
p= 0.286;    //photo elastic constant
Kb = 1.381*10^-23;     //Boltzmann's constant
lamda = 850*10^-9;  //wavelength
alpha_scat = 8*%pi^3*n^8*p^2*Kb*Tf*BETA/(3*lamda^4);
l= 1000;    //fibre length
TL = exp(-alpha_scat*l);  //transmission loss
attenuation = 10*log10(1/TL);
printf("The attenuation is %.3f dB/Km",attenuation);
```

Scilab code Exa 2.3.1 To find radius of curvature

```
//Example 2.3.1 page 2.20

clc;
clear;

alpha = 2;
n1= 1.5;
del= 0.01;
a= 25*10^-6;
lamda= 1.3*10^-6;
M= 0.5;
NA= sqrt(0.5*2*1.3^2/(%pi^2*25^2));
Rc= 3*n1^2*lamda/(4*%pi*NA^3);
Rc=Rc*1000; // converting into um.....
printf("The radius of curvature is %.2f um",Rc);
```

Scilab code Exa 2.5.1 To find the pulse spreading

```
//Example 2.5.1 page 2.25

clc;
clear;

lamda = 850 *10^-9;
sigma= 45*10^-9;
L= 1;
M= 0.025/(3*10^5*lamda);
sigma_m= sigma*L*M;
sigma_m= sigma_m*10^9; // formatting in ns/km....
printf("The Pulse spreading is %.2f ns/Km",sigma_m);
printf("\n\nNOTE*** - The answer in text book is
   wrongly calculated..");
```

Scilab code Exa 2.5.2 To find the material dispersion induced pulse spreading

```
//Example 2.5.2 pagw 2.26

clc;
clear;
lamda= 2*10^-9;
sigma = 75;
D_mat= 0.03/(3*10^5*2);
sigma_m= 2*1*D_mat;
sigma_m=sigma_m*10^9; //Fornamtting in ns/Km
printf("The Pulse spreading is %d ns/Km",sigma_m);
D_mat_led= 0.025/(3*10^5*1550);
sigma_m_led = 75*1*D_mat_led*10^9; //in ns/Km
printf("\n\nThe Pulse spreading foe LED is  %.2f ns/
    Km",sigma_m_led);
```

Scilab code Exa 2.5.3 To find the material dispersion

```
//Example 2.5.3 page 2.26

clc;
clear;
 lamda = 850;
 sigma= 20;
 D_mat = 0.055/(3*10^5*lamda);
sigma_m= sigma*1*D_mat;
D_mat=D_mat*10^12; // in Ps...
sigma_m=sigma_m*10^9; //in ns////
 printf("The material Dispersion is %.2f Ps/nm-Km",
    D_mat);
 printf("\n\nThe Pulse spreading is %.4f ns/Km",
    sigma_m);
```

Scilab code Exa 2.5.4 To compute the wave guide dispersion

```
//Example 2.5.4 page 2.30

clc;
clear;

n2= 1.48;
del = 0.2;
lamda = 1320;
Dw = -n2*del*0.26/(3*10^5*lamda);
Dw=Dw*10^10; //converting in picosecs....
printf("The waveguide dispersion is %.3f picosec/nm.
   Km",Dw);
```

Scilab code Exa 2.6.1 To find the bandwidth pulse broadening and bandwidth length product

```
//Example 2.6.1 page 2.34

clc;
clear;

t= 0.1*10^-6;
L= 12;
B_opt= 1/(2*t);
B_opt=B_opt/1000000; //converting from Hz to MHz
printf("The maximum optical bandwidth is %d MHz.",
   B_opt);
del= t/L;   //Pulse broadening
del=del*10^9; // converting in ns...
```

```
printf("\n\nThe pulse broadening per unit length is
   %.2f ns/Km",del);
BLP= B_opt*L; //BW length product
printf("\n\nThe Bandwidth-Length Product is %d MHz.
   Km",BLP);
```

Scilab code Exa 2.6.2 To find the bandwidth pulse dispersion and bandwidth length product

```
//Example 2.6.2 page 2.34

clc;
clear;

t= 0.1*10^-6;
L= 10;
B_opt= 1/(2*t);
B_opt=B_opt/1000000; //converting from Hz to MHz
printf("The maximum optical bandwidth is %d MHz.",
   B_opt);
del= t/L;
del=del/10^-6; //converting in us...
printf("\n\nThe dispersion per unit length is %.2f
   us/Km",del);
BLP= B_opt*L;
printf("\n\nThe Bandwidth-Length product is %d MHz.
   Km",BLP);
```

Scilab code Exa 2.6.3 To find the bandwidth and the puse dispersion

```
//Example 2.6.3 page 2.25

clc;
```

```
clear;

t= 0.1*10^-6;
L=15;
B_opt= 1/(2*t);
B_opt=B_opt/1000000; //converting from Hz to MHz
printf("The maximum optical bandwidth is %d MHz.",
    B_opt);
del= t/L*10^9; //in ns...
printf("\n\nThe dispersion per unit length is %.2f
    ns/Km",del);
```

Scilab code Exa 2.6.4 To estimate rms pulse broadening

```
//Example 2.6.4 page 2.35

clc;
clear;

lamda = 0.85*10^-6;
rms_spect_width = 0.0012*lamda;
sigma_m= rms_spect_width*1*98.1*10^-3;
sigma_m=sigma_m*10^9; // converting in ns...
printf("The Pulse Broadening due to material
    dispersion is %.2f ns/Km",sigma_m);
```

Scilab code Exa 2.6.5 To find the delay difference rms pulse broadening and maximum bit rate

```
//Example 2.6.5 page 2.35

clc;
clear;
```

```

L= 5; //in KM
n1= 1.5;
del= 0.01;
c= 3*10^8; // in m/s
delta_t = (L*n1*del)/c;
delta_t=delta_t*10^12; //convertin to nano secs...
printf("The delay difference is %.1f ns",delta_t);
sigma= L*n1*del/(2*sqrt(3)*c);
sigma=sigma*10^12; //convertin to nano secs...
printf("\n\nThe r.m.s pulse broadening is %.2f ns",
   sigma);
B= 0.2/sigma*1000; //in Mz
printf("\n\nThe maximum bit rate is %.2f MBits/sec",
   B);
BLP = B*5;
printf("\n\nThe Bandwidth-Length is %.2f MHz.Km",BLP
   );
```

Scilab code Exa 2.6.6 To compute intermodal intramodal and total dispersion

```
//Example 2.6.6 page 2.36

clc;
clear;

del_t_inter = 5*1;
del_t_intra = 50*80*1;
total_dispersion = sqrt(5^2 + 0.4^2);
printf("Total dispersion is %.3f ns",
   total_dispersion);
```

Scilab code Exa 2.7.1 To find the bandwidth pulse dispersion and bandwidth length product

```
//Example 2.7.1 page 2.37

clc;
clear;

t= 0.1*10^-6;
L=15;
del= t/L*10^9;  //convertin to nano secs...
printf("The Pulse Dispersion is %.2f ns",del);
B_opt= 1/(2*t)/10^6; //convertin to nano secs...
printf("\n\n The maximum possible Bandwidth is %d
   MHz",B_opt);
BLP = B_opt*L;
printf("\n\nThe BandwidthLength product is %d MHz.Km
   ",BLP);
```

Scilab code Exa 2.7.2 To find the delay difference and the rms pulse broadening

```
//Example 2.7.2 page 2.38

clc;
clear;
L= 6;
n1= 1.5;
del= 0.01;
delta_t = L*n1*del/(3*10^8)*10^12; //convertin to
   nano secs...
printf("The delay difference is %d ns",delta_t);
```

Scilab code Exa 2.7.3 To determine modal bifringence

```
//Example 2.7.3 page 2.39

clc;
clear;

Lb= 0.09;
lamda= 1.55*10^-6;
delta_lamda = 1*10^-9;
Bf= lamda/Lb;
Lbc= lamda^2/(Bf*delta_lamda);
printf("The modal Bifriengence is %.2f meters ",Lbc);
beta_xy= 2*%pi/Lb;
printf("\n\nThe difference between propogation constants is %.2f", beta_xy);
```

Scilab code Exa 2.7.4 To estimate maximum possible bandwidth

```
//Example 2.7.4 page 2.37

clc;
clear;

t= 0.1*10^-6;
B_opt= 1/(2*t)/1000000;
printf("The maximum possible Bandwidth is %d MHz", B_opt);
```

Scilab code Exa 2.7.5 To estimate maximum possible bandwidth

```
//Example 2.7.5 page 2.40
```

```
clc;
clear;

t= 0.1*10^-6;

B_opt= 1/(2*t)/1000000;
printf("The maximum possible Bandwidth is %d MHz",
   B_opt);
```

Chapter 3

Optical Sources and Transmitters

Scilab code Exa 3.2.1 To find the emitted wavelength

```
//Example 3.2.1 page 3.10

clc;
clear;

x= 0.07;
Eg= 1.424+1.266*x+0.266*x^2;
lamda= 1.24/Eg;
printf("The emitted wavelength is %.2f um",lamda);
```

Scilab code Exa 3.2.2 To find the emitted wavelength

```
//Example 3.2.2 page 3.10

clc;
clear;
```

```
x= 0.26;
y=0.57;
Eg= 1.35-0.72*y+0.12*y^2;
lamda = 1.24/Eg;
printf("The wavelength emitted is %.2f um",lamda);
```

Scilab code Exa 3.2.3 To find total carrier recombination life time and optical power generated

```
// Example 3.2.3 page 3.12

clc;
clear;
Tr = 60*10^-9;   //radiative recombination time
Tnr= 90*10^-9;   //non radiative recomb time
I= 40*10^-3;     //current
t = Tr*Tnr/(Tr+Tnr);     //total recomb time
t=t*10^9;    //Converting in nano secs...
printf("The total carrier recombination life time is %d ns",t);
t=t/10^9;
h= 6.625*10^-34;     //plancks const
c= 3*10^8;
q=1.602*10^-19;
lamda= 0.87*10^-6;
Pint=(t/Tr)*((h*c*I)/(q*lamda));
Pint=Pint*1000; //converting inmW...
printf("\n\nThe Internal optical power is %.2f mW",Pint);
```

Scilab code Exa 3.2.4 To find bulk recombination life time quantum efficiency and internal power

```
//Example 3.2.4 page 3.13
clc;
clear;
lamda = 1310*10^-9;
Tr= 30*10^-9;
Tnr= 100*10^-9;
I= 40*10^-3;
t= Tr*Tnr/(Tr+Tnr);
t=t*10^9;  //converting in nano secs...
printf("Bulk recombination life time %.2f ns",t);
t=t/10^9;
n= t/Tr;
printf("\n\nInternal quantum efficiency is %.3f",n);
h= 6.625*10^-34;    //plancks const
c= 3*10^8;
q=1.602*10^-19;
Pint=(0.769*h*c*I)/(q*lamda)*1000;
printf("\n\nThe internal power level is %.3f mW",
   Pint);
printf("\n\n***NOTE: Internal Power wrong in text
   book.. Calculation Error..");
```

Scilab code Exa 3.2.5 To estimate external power efficiency

```
//Example 3.2.5 page 3.14

clc;
clear;
nx= 3.6;
TF= 0.68;
n= 0.3;
//Pe=Pint*TF*1/(4*nx^2);
//ne= Pe/Px*100      ..eq0
//Pe = 0.013*Pint      //Eq 1
//Pint = n*P;    //Eq 2
```

```
//substitute eq2 and eq1 in eq0
ne = 0.013*0.3*100;
printf("The external Power efficiency is  %.2f %%",
   ne);
// Wrongly printed in textbook. it should be P
   instead of Pint in last step
```

Scilab code Exa 3.2.6 To find optical power emitted by the device and the external power efficiency

```
//Example 3.2.6 page 3.15
clc;
clear;

lamda= 0.85*10^-6;
Nint = 0.60;
I= 20*10^-3;
h= 6.625*10^-34;     //plancks const
c= 3*10^8;
e=1.602*10^-19;
Pint = Nint*h*c*I/(e*lamda);
printf("The optical power emitted is %.4f W",Pint);

TF= 0.68;
nx= 3.6;
Pe= Pint*TF/(4*nx^2)*1000000;
printf("\n\nPower emitted in the air %.1f uW",Pe);
Pe=Pe/1000000;
Nep=Pe/Pint*100;
printf("\n\nExternal power efficiency is %.1f %%",
   Nep);
```

Scilab code Exa 3.2.7 To find total carrier recombination lifetime and power internally generated

```
//Example 3.2.7 page 3.16

clc;
clear;
lamda = 0.87*10^-6;
Tr= 50*10^-9;
I= 0.04;
Tnr= 110*10^-9;
t= Tr*Tnr/(Tr+Tnr);
t=t*10^9;  //converting in ns...
printf("Total carrier recombination life time is %.2
   f ns",t);
t=t/10^9;
h= 6.625*10^-34;     //plancks const
c= 3*10^8;
q=1.602*10^-19;
n= t/Tr;
printf("\n\nThe efficiency is %.3f %%",n);
Pint=(n*h*c*I)/(q*lamda)*1000;
printf("\n\nInternal power generated is %.2f mW",
   Pint);
printf("\n\n***NOTE- Internal Power wrong in book...
    ");
```

Scilab code Exa 3.2.8 To find the conversion efficiency

```
//Examplr 3.2.8 page 3.16

clc;
clear;

V= 2;
```

```
I= 100*10^-3;
Pc= 2*10^-3;
P= V*I;
Npc= Pc/P*100;
printf("The overall power conversion efficiency is
    %d %%",Npc);
```

Scilab code Exa 3.3.1 To find the optical gain

```
//Example  3.3.1 page 3.25

clc;
clear;
r1= 0.32;
r2= 0.32;
alpha= 10;
L= 500*10^-4;
temp=log(1/(r1*r2));
Tgth = alpha + (temp/(2*L));
printf("The optical gain at threshold is %.2f /cm",
    Tgth);
```

Scilab code Exa 3.3.2 To calculate the frequency and wavelength spacing

```
//Example 3.3.2 page 3.27
clc;
clear;
n= 3.7;
lamda = 950*10^-9;
L= 500*10^-6;
c= 3*10^8;
DELv = c/(2*L*n)*10*10^-10; //converting in GHz...
printf("The frequency spacing is %d GHz",DELv);
```

```
DEL_lamda= lamda^2/(2*L*n)*10^9; //converting to nm..
printf("\n\nThe wavelength spacing is %.2f nm",DEL_lamda);

printf("\n\n***NOTE- The value of wavelength taken wrongly in book");
// value of lamda taken wrongly while soving for DEL_LAMDA inthe book..
```

Scilab code Exa 3.3.3 To find the number of longitudinal modes and their frequency seperation

```
//Exapmle 3.3.3 page 3.30

clc;
clear;

L= 0.04;
n= 1.78;
lamda= 0.55*10^-6;
c= 3*10^8;
q= 2*n*L/lamda;
q=q/10^5;
printf("Number of longitudinal modes is %.2fx10^5",q);
del_f= c/(2*n*L);
del_f=del_f*10^-9;
printf("\n\nThe frequency seperation is %.1f GHz",del_f);
```

Scilab code Exa 3.3.4 To calculate the external power efficiency

```
//Example 3.3.4 page 3.33

clc;
clear;

Nt= 0.18;
V= 2.5;
Eg= 1.43;
Nep= Nt*Eg*100/V;
printf("The total efficiency is %.3f %%",Nep);
```

Scilab code Exa 3.3.5 To find the threshold current density and the threshold current

```
//Example 3.3.5 page 3.33

clc;
clear;
n= 3.6;
BETA= 21*10^-3;
alpha= 10;
L= 250*10^-4;

r= (n-1)^2/(n+1)^2;
Jth= 1/BETA *( alpha + (log(1/r)/L));
Jth=Jth/1000; //converting for displaying...
printf("The threshold current density is %.2fx10^3",
   Jth);
Jth=Jth*1000;
Ith =Jth*250*100*10^-8;
Ith=Ith*1000; //converting into mA...
printf("\n\nThe threshold current is %.1f mA",Ith);
```

Scilab code Exa 3.3.6 To compare the threshold current densities

```
//Exapmle 3.3.6 page 3.34
clc;
clear;

T= 305;
T0 = 160;
T1= 373;

Jth_32 = exp(T/T0);
Jth_100 = exp(T1/T0);
R_j = Jth_100/Jth_32;
printf('Ratio of current densities at 160K is %.2f",
    R_j);
printf("\n\n***NOTE- Wrong in book...\nJth(100)
    calculated wrongly...");
To = 55;
Jth_32_new = exp(T/To);
Jth_100_new = exp(T1/To);
R_j_new = Jth_100_new/Jth_32_new;
printf("\n\nRatio of current densities at 55K is %.2
    f",R_j_new);
//wrong in book...
```

Scilab code Exa 3.4.1 To calculate the optical power coupled

```
//Example 3.4.1 page .342

clc;
clear;

Bo= 150;
rs= 35*10^-4;
a1= 25*10^-6;
```

```
NA= 0.20;
a2= 50*10^-6;

Pled = (a1/rs)^2 * (%pi^2*rs^2*Bo*NA^2);
Pled=Pled*10^10; //converting in uW...
printf("The power coupled inthe fibre is %d uW",Pled);
Pled_new = (%pi^2*rs^2*Bo*NA^2);
Pled_new=Pled_new*10^6; //converting in uW...
printf("\n\nThe Power coupled for case 2 is %.2f uW",Pled_new);
```

Scilab code Exa 3.4.2 To find the Fresnel reflection and loss of power

```
//Example 3.4.2 page 3.43

clc;
clear;

n= 1.48;
n1= 3.6;
R= (n1-n)^2/(n1+n)^2;
printf("The Fresnel Reflection is %.4f",R);
L= -10*log10(1-R);
printf("\n\nPower loss is %.2f dB",L);
```

Scilab code Exa 3.4.3 To find the optical power coupled

```
//Example 3.4.3 page 3.44

clc;
clear;

```

```
NA= 0.20;
Bo= 150;
rs= 35*10^-6;
Pled = %pi^2*rs^2*Bo*NA^2;
Pled=Pled*10^10; //convertin in uW for displaying...
printf("The optical power coupled is %.2f uW",Pled);
```

Scilab code Exa 3.4.4 To calculate the optical loss

```
//Example 3.4.4 page 3.44

clc;
clear;

n1= 1.5;
n=1;
R= (n1-n)^2/(n1+n)^2;
L= -10*log10(1-R);
//Total loss is twice due to reflection
L= L+L;
printf("Total loss due to Fresnel Reflection is %.2f
    dB",L);
```

Scilab code Exa 3.4.5 To estimate insertion loss in different cases

```
//Example 3.4.5 page 3.51

clc;
clear;
n1= 1.5;
n=1;
y=5;
a= 25;
```

```
temp1=(1-(y/(2*a)^2))^0.5;
temp1=temp1*(y/a);
temp=2*acosd(0.9996708);// it should be acos(0.1) actually... due to approximations
    // answer varies a lot...
temp=temp-temp1;
//temp=temp;
tem= 16*(1.5^2)/(2.5^4);
tem=tem/%pi;
temp=temp*tem;
Nlat= temp;
printf("The Coupling efficiency is %.3f ",Nlat);
L= -10*log10(Nlat);
printf("\n\nThe insertion loss is %.2f dB",L);
temp1=(1-(y/(2*a)^2))^0.5;
temp1=temp1*(y/a);
temp=2*acosd(0.9996708);// it should be acos(0.1) actually... due to approximations
    // answer varies a lot...
temp=temp-temp1;
temp=temp/%pi;
N_new =temp ;
printf("\n\nEfficiency when joint index is matched is %.3f",N_new);
L_new= -10*log10(N_new);
printf("\n\nThe new insertion loss is %.2f dB",L_new);
```

Chapter 4

Optical Detectors and Receivers

Scilab code Exa 4.1.1 To find the cutoff wavelength

```
//Example 4.1.1 page 4.5

clc;
clear;

Eg= 1.1;
lamda_c = 1.24/Eg;
printf("The cut off wavelength is %.2f um",lamda_c);

Eg_ger =0.67;
lamda_ger= 1.24/Eg_ger;
printf("\n\nThe cut off wavelength for Germanium is
   %.2f um",lamda_ger);
```

Scilab code Exa 4.1.2 To find the upper cutoff wavelength

```
//Example 4.1.2 page 4.5

clc;
clear;
Eg = 1.43;
lamda = 1.24/Eg;
lamda=lamda*1000; //converting in nm
printf("The cut off wavelength is %.2f nm",lamda);
```

Scilab code Exa 4.1.3 To find the quantum efficiency of the detector

```
//Example 4.1.3 page 4.3

clc;
clear;

P = 6*10^6;
Eh_pair= 5.4*10^6;
n= Eh_pair/P*100;
printf("The quantum efficiency is %d %%",n);
```

Scilab code Exa 4.1.4 To find generated photocurrent

```
//Example 4.1.4 page 4.6

clc;
clear;

R= 0.65;
P0= 10*10^-6;
Ip= R*P0;
Ip=Ip*10^6; //convertinf in uA...
printf("The generated photocurrent is %.1f uA",Ip);
```

Scilab code Exa 4.1.5 To find quantum efficiency and responsivity

```
//Example 4.1.5 page 4.6

clc;
clear;

Ec= 1.2*10^11;
P= 3*10^11;
lamda = 0.85*10^-6;
n= Ec/P*100;
printf("The efficiency is %d %%",n);

q= 1.602*10^-19;
h= 6.625*10^-34;
c= 3*10^8;
n= n/100;
R= n*q*lamda/(h*c);
printf("\n\nThe Responsivity of the photodiode is %
    .4f A/W",R);
```

Scilab code Exa 4.1.6 Operational wavelength and the incident optical power required

```
//Example 4.1.6   page 4.7

clc;
clear;

n= 0.65;
E= 1.5*10^-19;
```

```
Ip= 2.5*10^-6;
h= 6.625*10^-34;
c= 3*10^8;
lamda= h*c/E;
lamda=lamda*10^6; //converting in um for displaying
    ...
printf("The wavelength is %.3f um",lamda);
lamda=lamda*10^-6;
q= 1.602*10^-19;
R= n*q*lamda/(h*c);
printf("\n\nThe Responsivity is %.4f A/W",R);
Pin= Ip/R;
Pin=Pin*10^6;// converting in uW for displaying/..
printf("\n\nThe incidnt power is %.1f uW",Pin);
```

Scilab code Exa 4.1.7 To find the average photon current

```
//Example 4.1.7 page 4.8

clc;
clear;
Iin= 1;
lamda= 1550*10^-9;
q= 1.602*10^-19;
h= 6.625*10^-34;
c= 3*10^8;
n=0.65;
Ip=n*q*lamda*Iin/(h*c);
Ip=Ip*1000; //converting in mA for displaying...
printf("The average photon current is %d mA",Ip);
```

Scilab code Exa 4.1.8 To find the wavelength of operation incident power and the responsivity

```
//Example 4.1.8 page 4.9

clc;
clear;
n= 0.70;
Ip= 4*10^-6;
e= 1.602*10^-19;
h= 6.625*10^-34;
c= 3*10^8;
E= 1.5*10^-19
lamda = h*c/E;
lamda=lamda*10^6; //converting um for displaying...
printf("The wavelength is %.2f um",lamda);
R= n*e/E;
Po= Ip/R;
Po=Po*10^6; //converting um for displaying...
printf("\n\nIncident optical Power is %.2f uW",Po);
```

Scilab code Exa 4.2.1 To find the bandwidth

```
//Example 4.2.1 page 4.14

clc;
clear;
Ct= 7*10^-12;
Rt= 50*1*10^6/(50+(1*10^6));
B= 1/(2*%pi*Rt*Ct);
B=B*10^-6; //converting in mHz for displaying...
printf("The bandwidth of photodetector is %.2f MHz",
   B);
```

Scilab code Exa 4.2.2 To find the maximum response time

```
//Example 4.2.2 page 4.15

clc;
clear;

W= 25*10^-6;
Vd= 3*10^4;
Bm= Vd/(2*%pi*W);
RT= 1/Bm;
RT=RT*10^9;  //converting ns for displaying...
printf("The maximum response time is %.2f ns",RT);
```

Scilab code Exa 4.2.3 To find the multiplication factor

```
//Example 4.2.3. pahe 4.15

clc;
clear;
e= 1.602*10^-19;
h= 6.625*10^-34;
v= 3*10^8;
n=0.65;
I= 10*10^-6;
lamda= 900*10^-9;
R= n*e*lamda/(h*v);
Po= 0.5*10^-6;
Ip= Po*R;
M= I/Ip;
printf("The multiplication factor is %.2f",M);
```

Scilab code Exa 4.3.1 To find the multiplication factor

```
//Example 4.3.1 page 4.18
```

```
clc;
clear;

n=0.65;
lamda = 900*10^-9;
Pin= 0.5*10^-6;
Im= 10*10^-6;
q= 1.602*10^-19;
h= 6.625*10^-34;
c= 3*10^8;
R= n*q*lamda/(h*c);
Ip= R*Pin;
M= Im/Ip;
printf("The multiplication factor is %.2f",M);
printf("\n\n***NOTE-Answer wrong in textbook...");
```

Scilab code Exa 4.6.1 To find the various noise terms

```
//Example 4.6.1 page 4.34

clc;
clear;
lamda = 1300*10^-9;
Id= 4*10^-9;
n=0.9;
Rl= 1000;
Pincident= 300*10^-9;
BW= 20*10^6;
q= 1.602*10^-19;
h= 6.625*10^-34;
v= 3*10^8;
Iq= sqrt((q*Pincident*n*lamda)/(h*v));
Iq= sqrt(Iq);
Iq=Iq*100; //converting in proper format for
```

```
displaying...
printf("Mean square quantum noise current is %.2fx10^11 Amp",Iq);
I_dark= 2*q*BW*Id;
I_dark=I_dark*10^19;//converting in proper format for displaying...
printf("\n\nMean square dark current is %.3fx10^-19 Amp",I_dark);
k= 1.38*10^-23;
T= 25+273;
It= 4*k*T*BW/Rl;
It=It*10^16;//converting in proper format for displaying...
printf("\n\nMean square thermal nise current is %.2fx10^-16 Amp",It)
```

Scilab code Exa 4.8.1 To find the quantum limit

```
//Example 4.8.1 page 4.39

clc;
clear;
lamda = 850*10^-9;          //meters
BER= 1*10^-9;
N_bar = 9*log(10);
h= 6.625*10^-34;            //joules-sec
v= 3*10^8;                  //meters/sec
n= 0.65;                    // assumption
E=N_bar*h*v/(n*lamda);
E=E*10^18;            ///converting in proper format for displaying...
printf("The Energy received is %.2fx10^-18 Joules",E);
```

Scilab code Exa 4.8.2 To find minimum optical power

```
//Example 4.8.2 page 4.39

clc;
clear;

lamda = 850*10^-9;
BER = 1*10^-9;
BT=10*10^6;
h= 6.625*10^-34;
c= 3*10^8;
Ps= 36*h*c*BT/lamda;
Ps=Ps*10^12;///converting in proper format for
    displaying...
printf("The minimum incidental optical power
    required id %.2f pW",Ps);
```

Scilab code Exa 4.8.3 To find Signal Noise ratio

```
//Example 4.8.3 page 4.40

clc;
clear;

C= 5*10^-12;
B =50*10^6;
Ip= 1*10^-7;
e= 1.602*10^-19;
k= 1.38*10^-23;
T= 18+273;
M= 1;
```

```
Rl= 1/(2*%pi*C*B);
S_N= Ip^2/((2*e*B*Ip)+(4*k*T*B/Rl));
S_N = 10*log10(S_N); //in db
printf("The S/N ratio is %.2f dB",S_N);
M=41.54;
S_N_new= (M^2*Ip^2)/((2*e*B*Ip*M^2.3)+(4*k*T*B/Rl));
S_N_new = 10*log10(S_N_new); //in db
printf("\n\nThe new S/N ratio is %.2f dB",S_N_new);
printf("\n\nImprovement over M=1 is %.1f dB",S_N_new
    -S_N);
```

Chapter 5

Design Considerations in Optical Links

Scilab code Exa 5.3.1 To design optical fiber link

```
//Example 5.3.1 page 5.7

clc;
clear;

B= 15*10^-6;
L= 4;
BER= 1*10^-9;
Ls= 0.5;
Lc= 1.5;
alpha= 6;
Pm= 8;
Pt= 2*Lc +(alpha*L)+(Pm);
printf("The actual loss in fibre is %d dB",Pt);
Pmax = -10-(-50);
printf("\n\nThe maximum allowable system loss is %d
    dBm",Pmax);
```

Scilab code Exa 5.3.2 Calculate the link power budjet

```
//Example 5.3.2 page 5.8;
 clc;
 clear;

 Ps= 0.1;
 alpha = 6;
 L= 0.5;
 Ps = 10*log10(Ps);
 NA= 0.25;
 Lcoupling= -10*log10(NA^2);
 Lf= alpha*L;
lc= 2*2;
Pm= 4;
Pout = Ps-(Lcoupling+Lf+lc+Pm);
printf("The actual power output is %d dBm",Pout);
Pmin = -35;
printf("\n\nMinimum input power required is %d dBm",
   Pmin);
printf("\n\nAs Pmin > Pout, system will perform
   adequately over the system operating life.");
```

Scilab code Exa 5.3.3 Calculate the loss margin

```
//Example 5.3.3 page 5.8;

clc;
clear;

Ps= 5;
Lcoupling = 3;
```

```
Lc= 2;
L_splicing = 50*0.1;
F_atten  = 25;
L_total = Lcoupling+Lc+L_splicing+F_atten;
P_avail = Ps-L_total;
sensitivity = -40;
loss_margin = -sensitivity-(-P_avail);
printf("The loss margin of the system is -%d dBm",
   loss_margin);
sensitivity_fet = -32;
loss_margin_fet=-sensitivity_fet-(-P_avail);
printf("\n\nThe loss marging for the FET receiver is
    -%d dBm",loss_margin_fet);
```

Scilab code Exa 5.3.4 To perform optical power budjet

```
//Example 5.3.4 page 5.9

clc;
clear ;

LED_output = 3;
PIN_sensitivity = -54;
allowed_loss= LED_output -(-PIN_sensitivity);
Lcoupling = 17.5;
cable_atten = 30;
power_margin_coupling= 39.5;
power_margin_splice=6.2;
power_margin_cable=9.5;
final_margin= power_margin_coupling+
   power_margin_splice+power_margin_cable;
printf("The safety margin is %.2f dB",final_margin)
//Answer in book is wrong...
printf("\n\n***NOTE- Answer wrong in book...");
```

Scilab code Exa 5.3.5 Perform optical power budget

```
//Example 5.3.5 page 5.10

clc;
clear;

optical_power=-10;
receiver_sensitivity=-41;
total_margin= optical_power-receiver_sensitivity;
cable_loss= 7*2.6;
splice_loss= 6*0.5;
connector_loss= 1*1.5;
safety_margin= 6;
total_loss= cable_loss+splice_loss+connector_loss+
    safety_margin;
excess_power_margin= total_margin-total_loss;
printf("The system is viable and provides %.1f dB
    excess power margin.",excess_power_margin);
```

Scilab code Exa 5.4.1 To find the system rise time

```
//Example 5.4.1 page 5.13

clc;
clear;

Ttx= 15;
Tmat=21;
Tmod= 3.9;
BW= 25;
Trx= 350/BW;
```

```
Tsys = sqrt(Ttx^2+Tmat^2+Tmod^2+Trx^2);
printf("The system rise time is %.2f ns.",Tsys);
```

Scilab code Exa 5.4.2 To find system rise time and bandwidth

```
//Example 5.4.2 page 5.14

clc;
clear;
Ttrans = 1.75*10^-9;
Tled = 3.50*10^-9;
Tcable=3.89*10^-9;
Tpin= 1*10^-9;
Trec= 1.94*10^-9;
Tsys= sqrt(Ttrans^2+Tled^2+Tcable^2+Tpin^2+Trec^2);
Tsys=Tsys*10^9;//converting in ns for dislaying...
printf("The system rise time is %.2f ns",Tsys)
Tsys=Tsys*10^-9;
BW= 0.35/Tsys;
BW=BW/1000000;//converting in MHz for dislaying...
printf("\n\nThe system bandwidth is %.2f MHz",BW);
```

Scilab code Exa 5.4.3 To find maximum bit rate for link when using NRZ and RZ

```
//Example 5.4.3 page 5.14

clc;
clear;

Ttx= 8*10^-9;
Tintra= 1*10^-9;
```

```
Tmodal=5*10^-9;
Trr= 6*10^-9;
Tsys= sqrt(Ttx^2+(8*Tintra)^2+(8*Tmodal)^2+Trr^2);

BWnrz= 0.7/Tsys;
BWnrz=BWnrz/1000000;//converting in ns for dislaying
    ...
BWrz=0.35/Tsys;
BWrz=BWrz/1000000;//converting in ns for dislaying
    ...
printf("Maximum bit rate for NRZ format is %.2f Mb/
    sec",BWnrz);
printf("\n\nMaximum bit rate for RZ format is %.2f
    Mb/sec",BWrz);
```

Scilab code Exa 5.4.4 To find if the components give adequate response

```
//Example 5.4.4 page 5.15

clc;
clear;
Ts= 10*10^-9;
Tn=9*10^-9;
Tc=2*10^-9;
Td=3*10^-9;
BW= 6*10^6;
Tsyst= 1.1*sqrt(Ts^2+(5*Tn)^2+(5*Tc)^2+Td^2);
Tsyst=Tsyst*10^9;//converting in ns for displying...
Tsyst_max = 0.35/BW;
Tsyst_max=Tsyst_max*10^9;//converting in ns for
    displying...
printf("Rise system of the system is %.2f ns",Tsyst)
printf("\n\nMaximum Rise system of the system is %.2
    f ns",Tsyst_max)
printf("\n\nSpecified components give a system rise
```

```
time which is\n adequate for the bandwidth and distance requirements of the optical fibre link.");
```

Scilab code Exa 5.5.1 To find the maximum bit rates for different cases

```
//Example 5.5.1 page 5.18

clc;
clear;
del_t_1 = 10*100*10^-9;
Bt_nrz_1 = 0.7/(del_t_1*1000000);
Bt_rz_1 = 0.35/(del_t_1*1000000);
printf("First case. \n");
printf("Bit rate for nrz is:%.1f Mb/sec",Bt_nrz_1);
printf("\nBit rate for rz is:%.2f Mb/sec",Bt_rz_1);
del_t_2 = 20*1000*10^-9;
Bt_nrz_2 = 0.7/(del_t_2*1000000);
Bt_rz_2 = 0.35/(del_t_2*1000000);
printf("\n\nSecond case");
printf("\nBit rate for nrz is:%.3f Mb/sec",Bt_nrz_2);
printf("\nBit rate for rz  is:%.4f Mb/sec",Bt_rz_2);
del_t_3 = 2*2000*10^-9;
Bt_nrz_3 = 0.7/(del_t_3*1000);
Bt_rz_3 = 0.35/(del_t_3*1000);
printf("\n\nThird case");
printf("\nBit rate for nrz is:%d BITS/sec",Bt_nrz_3);
printf("\nBit rate for rz  is:%.1f BITS/sec",Bt_rz_3);
```

Chapter 6

Advanced Optical Systems

Scilab code Exa 6.5.1 To find the maximum input and output power

```
//Example 6.5.1 page 6.11

clc;
clear;

lamda_p= 980*10^-9;
lamda_s=1550*10^-9;
P_in=30;     // in mW....
G=100;

Ps_max= ((lamda_p*P_in)/lamda_s)/(G-1);
printf("\nMaximum input power is:%.5f mW",Ps_max);

Ps_out= Ps_max + (lamda_p*P_in/lamda_s);
Ps_out= 10*log10(Ps_out);
printf("\n\nOutput power is:%.2f dBm",Ps_out);
```

Scilab code Exa 6.5.2 To find the gain of EDFA

```
//Example 6.5.2 page 6.12

clc;
clear;

Ps_out= 30;          //in uW...
Ps_in=1;
Noise_power = 0.5;

G= Ps_out/Ps_in;

G= 10*log10(G);
printf("\nThe Gain EDFA is %.2f dB",G);
```

Scilab code Exa 6.10.1 To compute the performance parameters

```
// Example 6.10.1 page 6.22

clc;
clear;

P0= 200;
P1=90;
P2=85;
P3=6.3;
//All powers in uW...
coupling_ratio= P2/(P1+P2)*100;
printf("\n\n Coupling Ratio is %.2f %%",
   coupling_ratio);
excess_ratio= 10*log10(P0/(P1+P2))
printf("\n\n The Excess Ratio is %.4f dB",
   excess_ratio);
insertion_loss=10*log10(P0/P1);
printf("\n\n The Insertion Loss (from Port 0 to Port
    1) is %.2f dB",insertion_loss);
```

```
insertion_loss1=10*log10(P0/P2);
printf("\n\n The Insertion Loss (from Port 0 to Port
    2) is %.2f dB",insertion_loss1);
cross_talk=10*log10(P3/P0);
printf("\n\n The Cross Talk is %.d dB",cross_talk);
printf("\n\n***NOTE: Cross Talk calculated wrognly
   in book... Value of P3 wrognly taken");
```

Scilab code Exa 6.10.2 To calculate performance parameters

```
// Example 6.10.2 page 6.23

clc;
clear;

P0= 300;
P1=150;
P2=65;
P3=8.3*10^-3;
//All powers in uW...
splitting_ratio= P2/(P1+P2)*100;
printf("\n\n Splitting Ratio is %.2f %%",
   splitting_ratio);
excess_ratio= 10*log10(P0/(P1+P2))
printf("\n\n The Excess Ratio is %.4f dB",
   excess_ratio);
insertion_loss=10*log10(P0/P1);
printf("\n\n The Insertion Loss (from Port 0 to Port
    1) is %.2f dB",insertion_loss);
cross_talk=10*log10(P3/P0);
printf("\n\n The Cross Talk is %.2f dB",cross_talk);
```

Scilab code Exa 6.10.3 To find total loss in the coupler

```
//Example 6.10.3   page 6.25

clc;
clear;

N=32;
Ft=(100-5)/100;
Total_loss= 10*(1-3.322*log10(Ft))*log10(N);
printf("The total loss in the coupler is :%.2f dB",
   Total_loss);
```

Scilab code Exa 6.10.4 To compute total loss

```
//Example 6.10.4   page 6.28

clc;
clear;

N=10;
L=0.5;
alpha=0.4;
Lthru=0.9;
Lc=1;
Ltap=10;
Li=0.5;
Total_loss= N*(alpha*L +2*Lc +Lthru+Li)-(alpha*L)
   -(2*Lthru)+(2*Ltap);
printf("The total loss in the coupler is :%d dB",
   Total_loss);
```

Scilab code Exa 6.11.1 To compute the wave guide length differencr

```
//Example 6.11.1 page 6.33
```

```

clc;
clear;

del_v=10*10^9;
N_eff= 1.5;
c=3*10^11; // speed of light in mm/sec

del_L= c/(2*N_eff*del_v);
printf("The wave guide length differenc is %d mm",
   del_L);
```