Scilab Textbook Companion for Fiber Optic Communications: Fundamentals and Applications by S. Kumar and M. J. Deen[1]

Created by
Gajanan K Birajdar
Optical Fiber Communication
Electronics Engineering
Mumbai University
College Teacher
None
Cross-Checked by
None

May 30, 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: Fiber Optic Communications: Fundamentals and Applications

Author: S. Kumar and M. J. Deen

Publisher: Wiley, UK

Edition: 1

Year: 2014

ISBN: 978-0-470-51867-0

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

List of Figures

Chapter 1

Electromagnetics and Optics

Scilab code Exa 1.6 To find refractive index of of the glass

```
// Example 1.6
// To find refractive index of of the glass
// Page no.25

clc;
clear;

// Given data
phi=0.7297;                          // Critical angle for glass-air interface
n2=1;                                // Refractive index of air
n1=n2/sin(phi);                      // Refractive index of glass

// Displaying the result in command window
printf('\n Refractive index of the glass = %0.1f',n1);
```

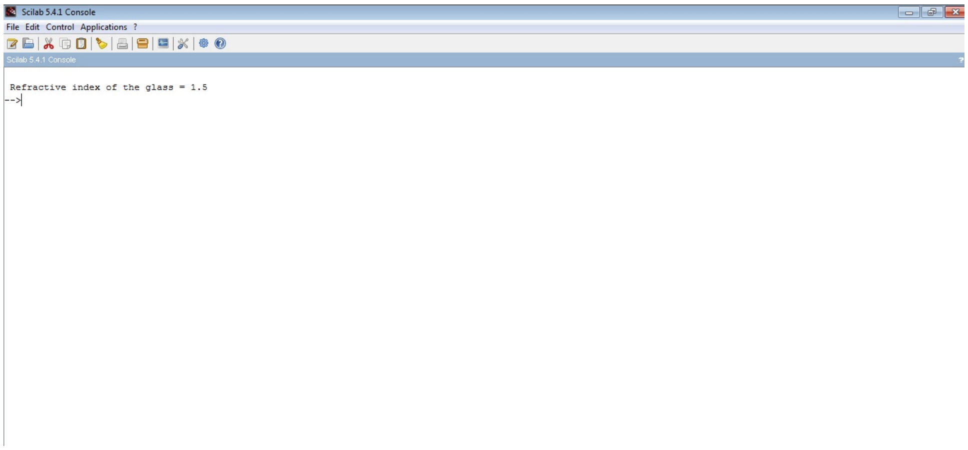

Figure 1.1: To find refractive index of of the glass

Scilab code Exa 1.7 To calculate the speed of light The wavelenght in medium The wavenumber in medium

```
//Example no 1.7
//To calculate a)the speed of light b) The wavelenght in medium c) The wavenumber in medium
//Page no. 25

clc;
clear all;

//a)The speed of light
c=3*10^8;                                    //Speed of light in free space (m/s)
n=1.45;                                      //Given refractive index of dielectric medium
v=(c/n);                                     //Speed of light in medium (in m/s)
```

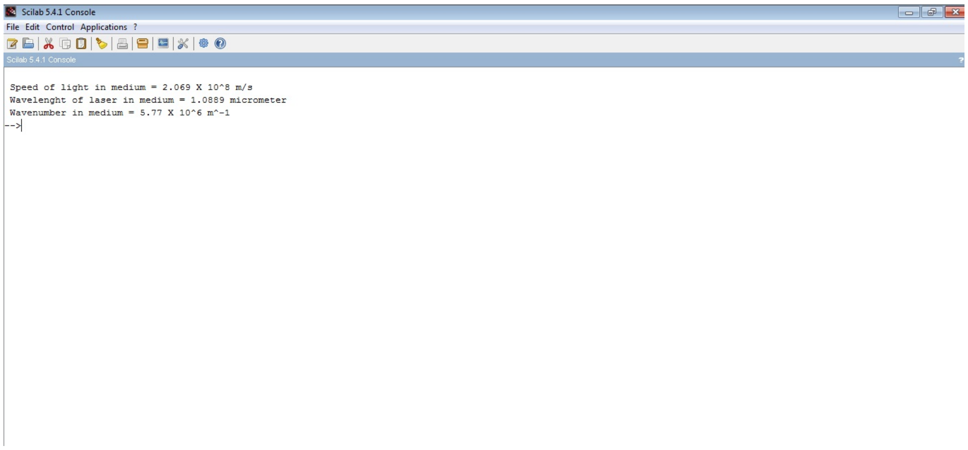

Figure 1.2: To calculate the speed of light The wavelenght in medium The wavenumber in medium

```

//Displaying the result in command window
printf('\n Speed of light in medium = %0.3f X 10^8 m/s',v*10^-8);

//b) The wavelenght in medium
f=190*10^12;                                    //Given operating frequency of laser
lambdam=(v/f);                                  //Wavelenght in medium

//Displaying the result in command window
printf('\n Wavelenght of laser in medium = %0.4f micrometer',lambdam*10^(6));

//c) The wavenumber in medium
k=(2*%pi)/lambdam;                              //Wavenumber in medium

//Displaying the result in command window
printf('\n Wavenumber in medium = %0.2f X 10^6 m^-1',k*10^-6)
```

Figure 1.3: To calculate magnitude of the wave vector of the refracted wave x component and z component of the wave vector

---

Scilab code Exa 1.8 To calculate magnitude of the wave vector of the refracted wave x component and z component of the wave vector

```
// Example no. 1.8
// To calculate a)magnitude of the wave vector of
   the refracted wave b)x-component and z-component
   of the wave vector
// Page no.26

clc;
clear;

//Given data
n1=1;
   // Refractive index of air
n2=1.45;
```

```
                                                              // Refractive index of slap
theta1=%pi/3;                                                 // Angle of incidence
lambdam=1.0889*10^(-6);                                       // Wavelength in medium
theta2=asin(sin(theta1)/n2);                                  // Angle of refraction

// a)To calculate magnitude of the wave vector of the refracted wave
k=((2*%pi)/lambdam);                                          // Wavenumber

// Displaying the result in command window
printf('\n Magnitude of the wave vector of the refracted wave is same as wave number = %0.2f X 10^6 m^-1',k*10^(-6));

// b)To calculate x-component and z-component of the wave vector
kx=k*sin(theta2);                                             // x-component of the wave vector
kz=k*cos(theta2);                                             // z-component of the wave vector

// Displaying the result in command window
printf('\n z-component of the wave vector = %0.2f X 10^6 m^-1',kz*10^(-6));
printf('\n x-component of the wave vector = %0.2f X 10^6 m^-1',kx*10^(-6));
// The answer is varrying due to round-off error
```

Scilab code Exa 1.9 To find length of the medium

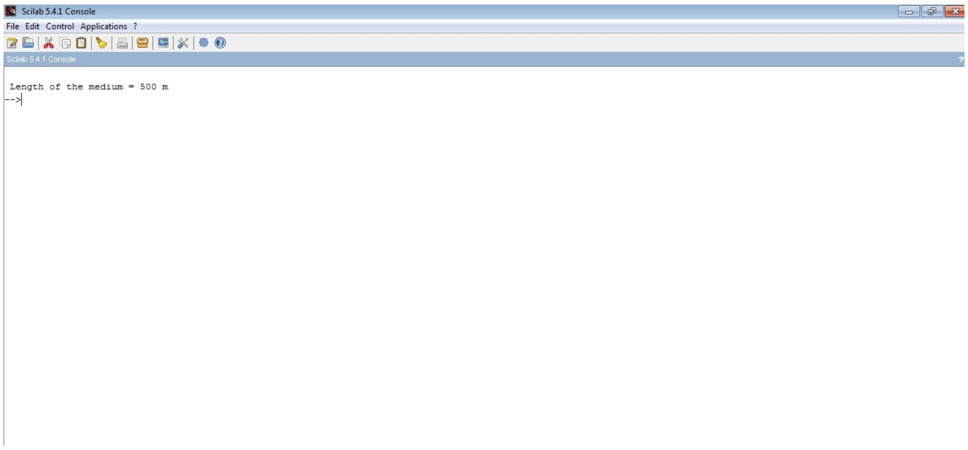

Figure 1.4: To find length of the medium

```
//To find length of the medium
//Example no 1.9
//Page no. 30

clc;
clear all;
bandwidth=100*10^9;                    //Bandwidth of optical signal
w=2*%pi*bandwidth;                     //Bandwidth of optical signal in rad/s
T=3.14*10^(-12);                       //Delay between minimum and maximum frequency component
beta2=10*(10^(-12))^2/10^3;            //Group velocity dispersion parameter in s^2/km
L=T/(beta2*w);                         //Length of the medium

// Displaying the result in command window
printf('\n Length of the medium = %0.0f m',L);
```

Chapter 2

Optical Fiber Transmission

Scilab code Exa 2.1 To find The numerical aperture The acceptanca angle The relative index defference

```
// Example no. 2.1
// To find a)The numerical aperture b)The acceptanca
    angle c)The relative index defference
// Page no. 38

clc;
clear;

// Given data
n1=1.47;                                        //
   Refractive index of core
n2=1.45;                                        //
   Refractive index of cladding

// a)The numerical aperture
NA=(n1^2-n2^2)^(1/2);                          //
   Numerical aperture

```

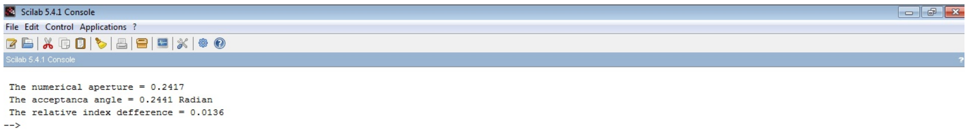

Figure 2.1: To find The numerical aperture The acceptanca angle The relative index defference

```
// Displaying the result in command window
printf('\n The numerical aperture = %0.4f',NA);

// b)The acceptanca angle
imax=asin(NA);                              // The acceptanca angle

// Displaying the result in command window
printf('\n The acceptanca angle = %0.4f Radian',imax);

// c)The relative index defference
delta=(n1-n2)/n1;                           // Relative index defference

// Displaying the result in command window
printf('\n The relative index defference = %0.4f',delta);
```

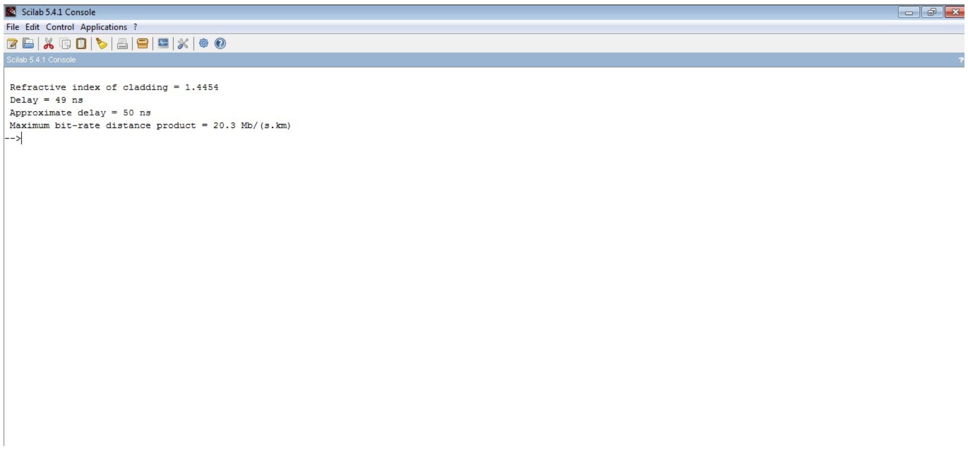

Figure 2.2: To find maximum bit rate distance product

Scilab code Exa 2.2 To find maximum bit rate distance product

```
// Example no. 2.2
// To find maximum bit-rate distance product
// Page no. 41

clc;
clear;

// Given data
n1=1.46;                                        // Refractive index of core
delta=0.01;                                     // Relative difference of refractive index
L=1*10^3;                                       // Fiber length
c=3*10^(8);
```

```
                                                        //
    Speed of ligth in km/sec

n2=n1*(1-delta);
                                                    //
    Refractive index of cladding
deltaT=(n1^2*L*delta)/(c*n2);
                                       // Delay in sec
BL=(((c*n2)/(n1^2*delta))/10^3)*10^-6;
                               // maximum bit-rate distance
    product in Mb/s.km
deltaT=((n1^2*L*delta)/(c*n2))*10^9;
                                 // Delay in ns

// Displaying the result in command window
printf('\n Refractive index of cladding = %0.4f',n2)
    ;
printf('\n Delay = %0.0f ns',deltaT);
printf('\n Approximate delay = %0.0f ns',deltaT+1);
printf('\n Maximum bit-rate distance product = %0.1f
     Mb/(s.km)',BL);
```

Scilab code Exa 2.3 To compare deltaT for step index fiber with parabolic index fiber

```
// Example no.2.3
// To compare deltaT for step index fiber with
    parabolic-index fiber
// Page no. 43

clc;
clear;

```

Figure 2.3: To compare deltaT for step index fiber with parabolic index fiber

```
// Given data
n1=1.47;    // Refractive index of core
n2=1.45;    // Refractive index of cladding
L=1*10^3;    // Length of medium in meter
c=3*10^8;    // speedof ligth in (m/s)
delta=(n1-n2)/n1;

// The deltaT for step index fiber
deltaTSIF=((n1^2*L*delta)/(c*n2))*10^9;    //Pulse width for step index fiber

// deltaT for parabolic-index fiber
deltaTPIF=((n1^2*delta^2*L)/(8*c))*10^9;    // Pulse width for parabolic-index fiber

// Displaying the result in command window
printf('\n Pulse width for step index fiber = %0.2f ns',deltaTSIF);
printf('\n Pulse width for parabolic index fiber =
```

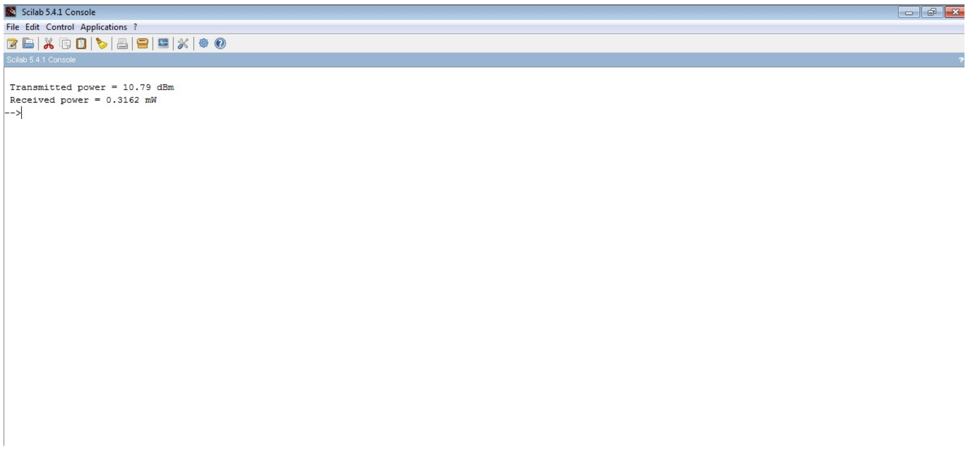

Figure 2.4: To convert transmitted power into dBm To convert received power into mW

```
%0.4f ns',deltaTPIF);

// The answer of pulse width for parabolic index
   fiber is wrong in book

disp('Thus, the intermodal dispersion can be
   significantly reduced by using parabolic-index
   fiber');
```

Scilab code Exa 2.4 To convert transmitted power into dBm To convert received power into mW

```
// Example 2.4
// a)To convert transmitted power into dBm b)To
   convert received power into mW
// Page no. 61

clc;
```

```
clear;

// Given data
Ptr=0.012;                                  // Transmitted power in watt
PrdBm=-5;                                   // Received power in dBm

// a)To convert transmitted power into dBm
PtrdBm=10*log10(Ptr/(10^-3));               // Transmitted power in dBm

// Displaying the result in command window
printf('\n Transmitted power = %0.2f dBm',PtrdBm);

// b)To convert received power into mW
PrmW=10^(-5/10);                           // Received power in mW

// Displaying the result in command window
printf('\n Received power = %0.4f mW',PrmW);
```

Scilab code Exa 2.7 To find the core radius of step index fiber

```
// Example no.2.7
// To find the core radius of step-index fiber
// Page no.69

clc;
clear;

// Given data
n1=1.45;
```

Figure 2.5: To find the core radius of step index fiber

```
    // Refractive index of core
delta=0.005;
n2=n1*(1-delta);
    // Refractive index of cladding
lambdac=1.1;
    // Cutoff wavelength in meter
lambda=1.55;
    // Operating wavelength in micrometer
a=((2.4048*lambdac*10^-6)/(2*%pi*(n1^2-n2^2)^(1/2)))/10^-6;        // Core radius

//Displaying the result in command window
printf('\n The core radius of step-index fiber = %0.3f micrometer',a);
printf('\n Operating wavelength = %0.2f micrometer',lambda);
printf('\n Cutoff wavelength = %0.1f micrometer',lambdac);

```

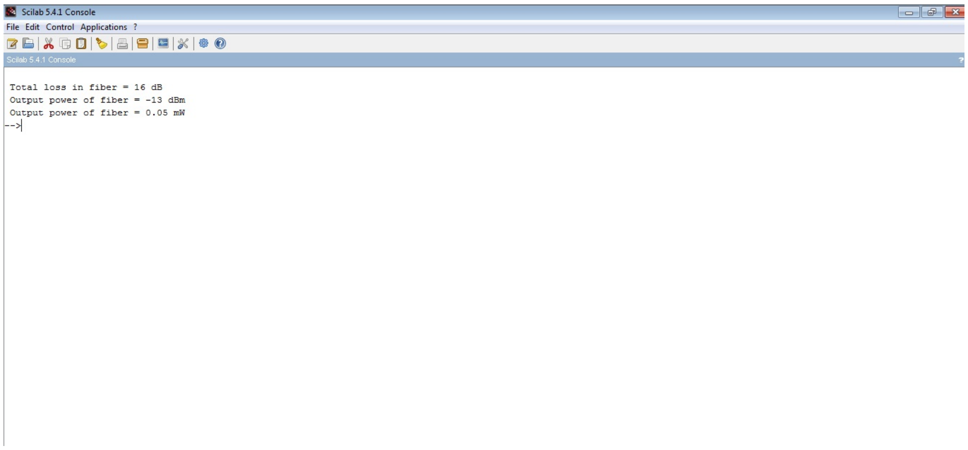

Figure 2.6: To find the total loss and output power in mW and dBm in fiber

```
disp('Since operating wavelength is greater than cutoff wavelength, it is single moded at this wavelength.')
```

Scilab code Exa 2.8 To find the total loss and output power in mW and dBm in fiber

```
// Example no 2.8
// To find the total loss and output power in mW and dBm in fiber
// Page no. 72

clc;
clear;

// Given data
losscoe=0.046;                                    // Loss coefficient in km^-1
```

```
L=80;                                         // Length of fiber in km
PindBm=3;                                     // Input power in dBm

// To find total loss of fiber
loss=round(4.343*losscoe*L);                  // Total loss in fiber

// Displaying the result in command window
printf('\n Total loss in fiber = %0.0f dB',loss);

// To find output power
PoutdBm=PindBm-loss;                          // Output power in dBm

PoutmW=10^(PoutdBm/10);                       // Output power in mW

//Displaying the result in command window
printf('\n Output power of fiber = %0.0f dBm',PoutdBm);
printf('\n Output power of fiber = %0.2f mW',PoutmW);
```

Scilab code Exa 2.10 To design single mode fiber such that absolute accumulated dispersion

```
// Example no. 2.10
// To design single mode fiber such that absolute accumulated dispersion should not exceed 1100ps/nm
// Page no. 77
```

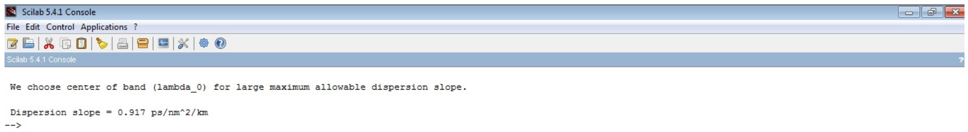

Figure 2.7: To design single mode fiber such that absolute accumulated dispersion

```

clc;
clear;

// Given data
lambda1=1530;                                   // Left edge of wavelength range in nm
lambda2=1560;                                   // Rigth edge of wavelength range in nm
lambda0=1545;                                   // Center of the band in nm
L=80;                                          // Fiber length in km

disp('We choose center of band (lambda_0) for large maximum allowable dispersion slope.');

Dlambda2=1100/L;                               // Dispersion at rigth edge of band in ps/nm/km
S=Dlambda2/(lambda2-lambda0);                  // Dispersion slope in ps/nm^2/km

```

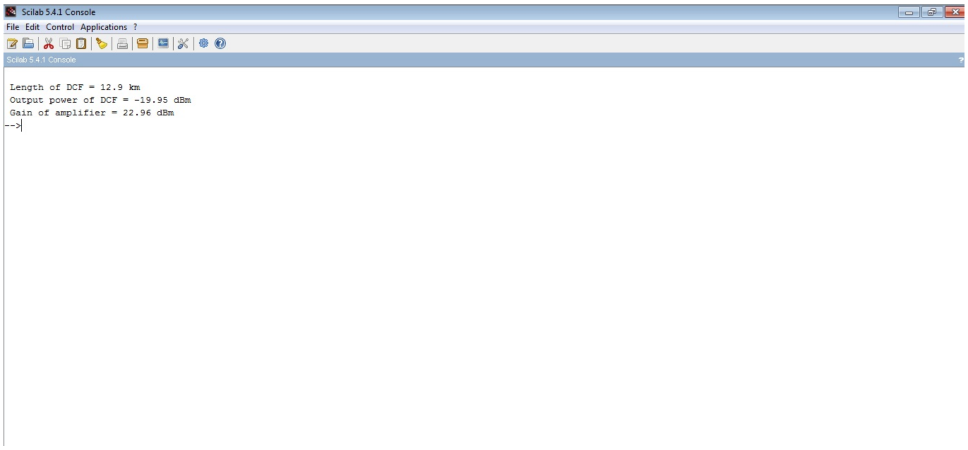

Figure 2.8: To find length of DCF power at the output of DCF gain of amplifier

```
// Displaying the result in command window
printf('\n Dispersion slope = %0.3f ps/nm^2/km',S);
```

Scilab code Exa 2.11 To find length of DCF power at the output of DCF gain of amplifier

```
// Example no.2.11
// To find a)length of DCF b)power at the output of
   DCF c)gain of amplifier
// Page no.80

clc;
clear;

// Given data
LTF=80;

   // Length of transmission fiber
```

```
beta2TF=-21;                                    // Dispersion of transmission fiber in ps^2/km
beta2DCF=130;                                   // Dispersion of DCF in ps^2/km
Pin=2*10^(-3);                                  // Input power of transmission fiber in W
DCFloss=0.5;                                    // Losses of DCF in dB/km
TFloss=0.2;                                     // Losses of TF in dB/km
spliceloss=0.5;                                 // Splice loss in dB

// a)To find length of DCF
LDCF=(-beta2TF*LTF)/beta2DCF;                   // Length of DCF in km

// Displaying the result in command window
printf('\n Length of DCF = %0.1f km',LDCF);

// b)To find power at the output of DCF
PindBm=10*log10(Pin/10^(-3));                   // Input power of transmission fiber in dBm
Totalloss=TFloss*LTF+DCFloss*LDCF+spliceloss;   // Total loss in fiber in dB
PoutdBm=PindBm-Totalloss;                       // Output power of DCF in dBm

// Displaying the result in command window
```

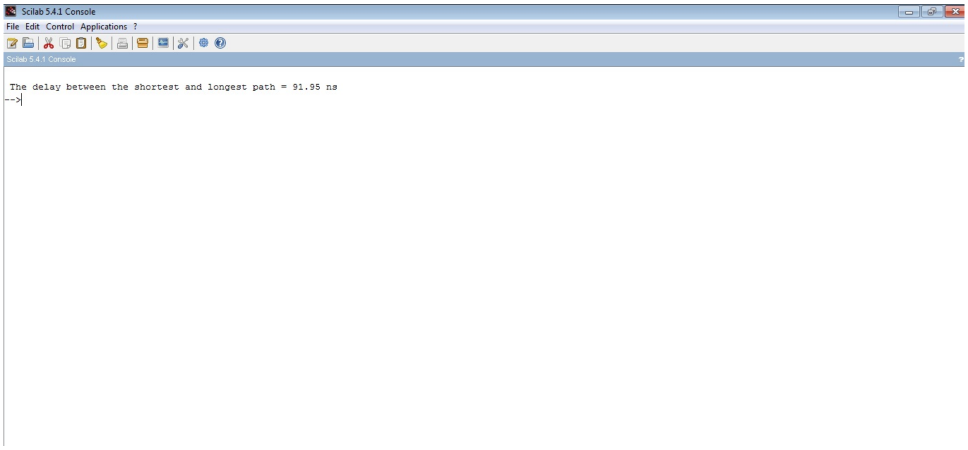

Figure 2.9: To find the delay between the shortest and longest path

```
printf('\n Output power of DCF = %0.2f dBm',PoutdBm)
    ;

// c)To find gain of amplifier
gain=Totalloss;                                            // gain of
    amplifier

// Displaying the result in command window
printf('\n Gain of amplifier = %0.2f dBm',gain);
```

Scilab code Exa 2.12 To find the delay between the shortest and longest path

```
// Example no. 2.12
// To find the delay between the shortest and
    longest path.
// Page no. 81

```

```
clc;
clear;

// Given data
NA=0.2;                                         // Numerical aperture
L=2*10^3;                                       // Fiber length in meters
n1=1.45;                                        // Core refractive index
delta=(NA)^2/(2*n1^2);                          // Relative index difference
n2=n1;                                          // since difference between core index and cladding index is smaller
c=3*10^8;                                       // Speed of ligth in m/s

// The delay between the shortest and longest path.
deltaT=((n1^2*L*delta)/(c*n2));                // the delay between the shortest and longest path.

// Displaying the result in command window
printf('\n The delay between the shortest and longest path = %0.2f ns',deltaT*10^9);
```

Scilab code Exa 2.13 To calculate the propagation constant

```
// Example no. 2.13
// To calculate the propagation constant
// Page no. 82

clc;
```

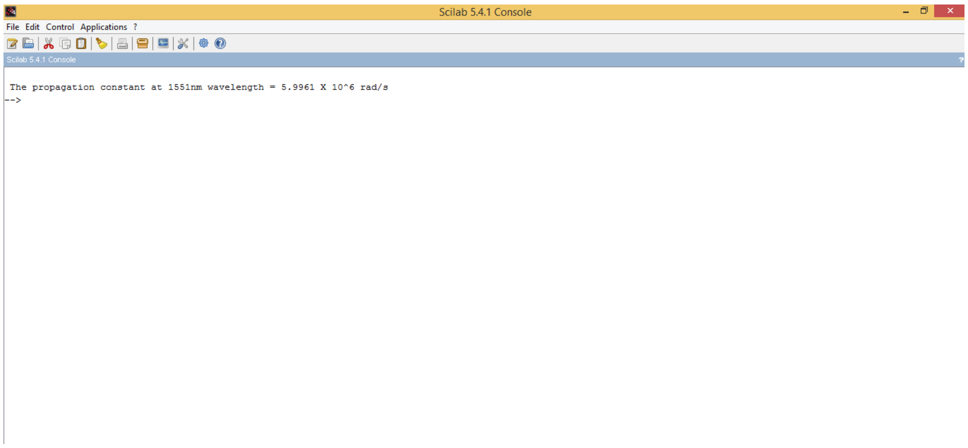

Figure 2.10: To calculate the propagation constant

```
clear;

// Given data
lambda0=1550*10^-9;                                    // wavelength in meter
beta0=6*10^6;                                          // propagation constant in rad/m
lambda1=1551*10^-9;                                    // wavelength in meter
beta1=0.5*10^-8;                                       // inverse group velocity in sec/meter
beta2=-10*10^-24;                                      // second-order dispersion coefficient in sec^2/km
c=3*10^8;                                              // Speed of ligth in m/s
omega0=(2*%pi*c)/lambda0;                              // Radial
```

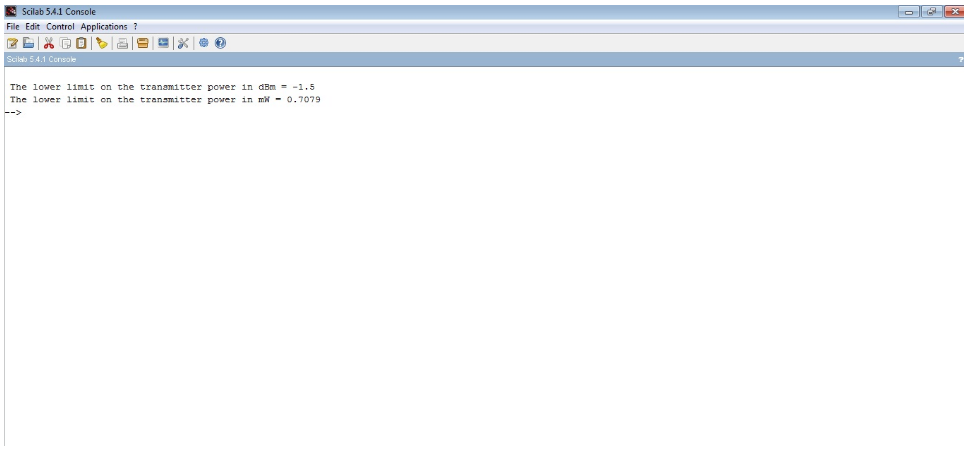

Figure 2.11: To calculate the lower limit on the transmitter power in dBm and mW units

```
   frequency at lambda0
omega1=(2*%pi*c)/lambda1;                                      // Radial
   frequency at lambda1
omega=omega1-omega0;

// The propagation constant at 1551nm wavelength
betaomega1=(beta0+beta1*omega+beta2*omega^2/2);
          // Propagation constant at 1551nm
   wavelength

// Displaying the result in command window
printf('\n The propagation constant at 1551nm
   wavelength = %0.4f X 10^6 rad/s',betaomega1
   *10^-6);
```

Scilab code Exa 2.14 To calculate the lower limit on the transmitter power in dBm and mW units

```
// Example No. 2.14
// To calculate the lower limit on the transmitter power in dBm and mW units.
// Page No. 83

clc;
clear;

// Given data
l=80;                          // Length of fiber in km
F1=-0.2*l;                     // Fiber loss in dB
F2=-0.5;                       // Filter loss in dB
G=15;                          // Amplifier gain in dB
Pout=-3;                       // Minimum power required at the receiver in dBm

// Lower limit on the transmitter power
Pin=Pout-F1-F2-G;              // Lower limit on the transmitter power in dBm
PinmW=10^(0.1*Pin);            // Lower limit on the transmitter power in mW

// Displaying the result in command window
printf('\n The lower limit on the transmitter power in dBm = %0.1f',Pin);
printf('\n The lower limit on the transmitter power in mW = %0.4f',PinmW);
```

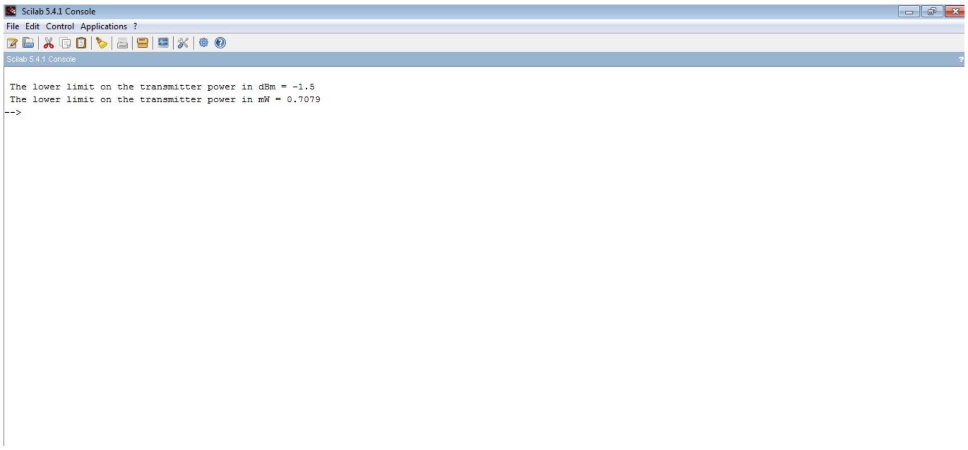

Figure 2.12: To find the length of DCF so that the pulse width at the output of the DCF is twice the pulse width at the input of the TF

Scilab code Exa 2.16 To find the length of DCF so that the pulse width at the output of the DCF is twice the pulse width at the input of the TF

```
// Example No. 2.16
// To find the length of DCF so that the pulse width (FWHM) at the output of the DCF is twice the pulse width at the input of the TF
// Page No. 84

clc;
clear;

// Given data
beta2TF=-21*(10^(-12))^2;                    // Dispersion coefficient of transmission fiber in s^2/km
beta2DCF=130*(10^(-12))^2;                   // Dispersion coefficient of dispersion compensating fiber in s^2/km
LTF=80;
```

```
    // Length of transmission fiber in km
TFWHM=12.5*10^(-12);
                                                  // Full-width
    at half-maximum
T0=TFWHM/1.665;
                                                       // Half-
    width

// The length of required DCF
LDCF1=(sqrt(3)*T0^2-beta2TF*LTF)/beta2DCF;
                  // Length of dispersion
    compensating fiber in km
LDCF2=(-sqrt(3)*T0^2-beta2TF*LTF)/beta2DCF;
                 // Length of dispersion
    compensating fiber in km

// Displaying the result in command window
printf('\n The length of DCF so that the pulse width
     (FWHM) at the output of the DCF is twice the
    pulse width at the input of the TF = %0.2f km',
    LDCF1);
printf(' or = %0.2f km',LDCF2);
```

Scilab code Exa 2.17 To find the accumulated dispersion of the DCF so that the net accumulated dispersion

```
// Example No. 2.17
// To find the accumulated dispersion of the DCF so
    that the net accumulated dispersion does not
    exceed 1100 ps/nm
// Page no. 85

clc;
```

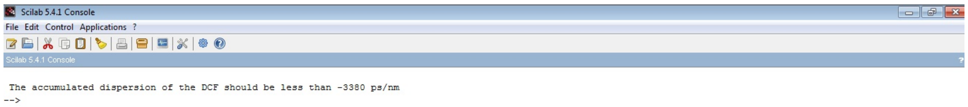

Figure 2.13: To find the accumulated dispersion of the DCF so that the net accumulated dispersion

```
clear;

// Given data
lambda0=1490;        // Zero dispersion wavelength in nm
lambda=1560;         // Upper limit of wavelength range in nm
Sc=0.08;             // Dispersion slope of transmission fiber ps/nm2/km
LTF=800;             // Length of transmission fiber in km
DTF=Sc*(lambda-lambda0);        // Dispersion at 1560 nm in ps/nm/km

// The accumulated dispersion of the DCF
DLDCF=1100-DTF*LTF;
```

```
// The accumulated dispersion of the DCF in ps/nm

// Displaying the result in command window
printf('\n The accumulated dispersion of the DCF should be less than %0.0f ps/nm',DLDCF);
```

Chapter 3

Lasers

Scilab code Exa 3.1 To calculate the Einstein A and B coefficients

```
// Example No. 3.1
// To calculate the Einstein A and B coefficients
// Page no.99

clc;
clear;

// Given data
tsp=2*10^-9;
    // Spontaneous lifetime associated with 2    1
    transition in seconds

deltaE=2.4*10^(-19);
    // The energy difference between the levels
h=1.054*10^(-34);
    // The distance between two levels
```

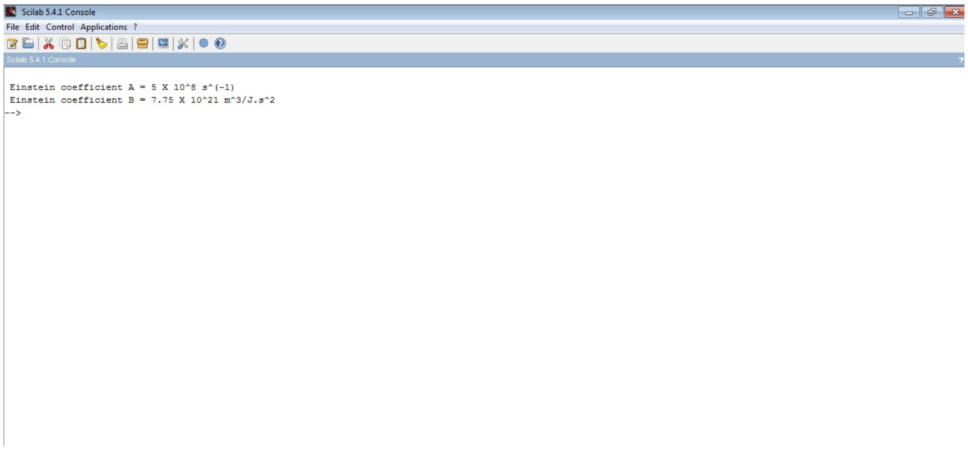

Figure 3.1: To calculate the Einstein A and B coefficients

```
omega=deltaE/h;
    // Frequency in rad/sec
v=1.25*10^8;
    // The velocity of light in the medium in m/s

// The Einstein A and B coefficients
A=(1/tsp)*10^-8;
    // Einstein coefficient A
B=(((1/tsp)*%pi^2*v^3)/(h*omega^3))*10^-21;
                                        // Einstein coefficient
    B

// Displaying the result in command window
printf('\n Einstein coefficient A = %0.0f X 10^8 s
    ^(-1)',A);
printf('\n Einstein coefficient B = %0.2f X 10^21 m
    ^3/J.s^2',B);

// The answers are varrying due to round off error
```

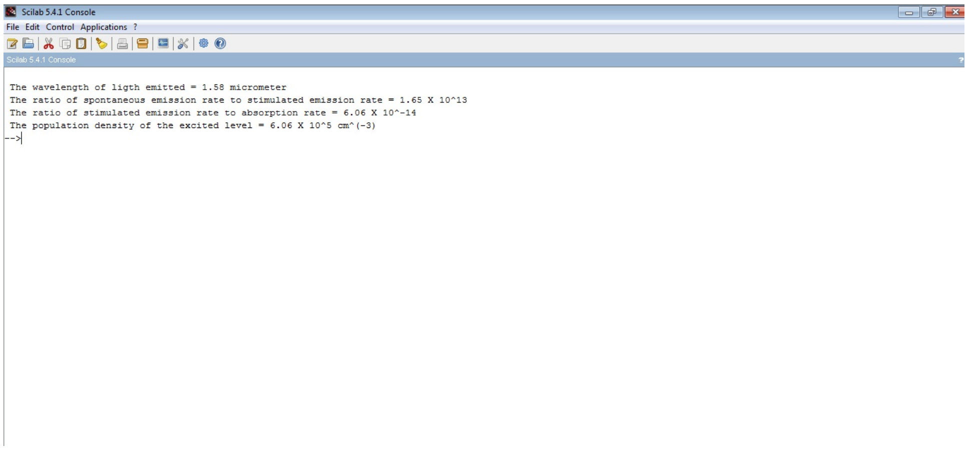

Figure 3.2: To calculate the wavelength of light emitted the ratio of spontaneous emission rate to stimulated emission rate the ratio of stimulated emission rate to absorption rate and the population density of the excited level

Scilab code Exa 3.2 To calculate the wavelength of light emitted the ratio of spontaneous emission rate to stimulated emission rate the ratio of stimulated emission rate to absorption rate and the population density of the excited level

```
// Example no. 3.2
// To calculate (a) the wavelength of light emitted,
    (b) the ratio of spontaneous emission rate to
   stimulated emission rate, (c) the ratio of
   stimulated emission rate to absorption rate, and
   (d) the population density of the excited level.
// Page no. 100

clc;
clear;
```

```
// Given data
deltaE=1.26*10^-19;    // The energy difference between two levels
h=1.054*10^(-34);    // The distance between two levels
c=3*10^8;    // The speed of ligth in m/s
kB=1.38*10^(-23);    // The Boltzmann s constant J/K
T=300;    // The absolute temperature in Kelvin
N1=10^19;    // The population density in the ground state in cm^(-3)

// (a)The wavelength of light emitted
h=2*%pi*h;    // The distance between two levels in J.s
f=deltaE/h;    // The frequency in Hz
lambda=(c/f)*10^6;    // The wavelength of ligth emitted in micrometer

// Displaying the result in command window
printf('\n The wavelength of ligth emitted = %0.2f micrometer',lambda);

// The calculation of this answer is wrong in the
```

```
    book

// (b)The ratio of spontaneous emission rate to stimulated emission rate
RspRst=(exp(deltaE/(kB*T))-1);                                    // The ratio of spontaneous emission rate to stimulated emission rate

// Displaying the result in command window
printf('\n The ratio of spontaneous emission rate to stimulated emission rate = %0.2f X 10^13',RspRst*10^-13);

// The calculation of this answer is wrong in the book

// (c)The ratio of stimulated emission rate to absorption rate
RstRab=(exp(-deltaE/(kB*T)));                                     // The ratio of stimulated emission rate to absorption rate

// Displaying the result in command window
printf('\n The ratio of stimulated emission rate to absorption rate = %0.2f X 10^-14',RstRab*10^14);

// The calculation of this answer is wrong in the book

// (d)The population density of the excited level
N2=(N1*exp(-deltaE/(kB*T)));                                      // The population density of the excited level in cm^(-3)

// Displaying the result in command window
```

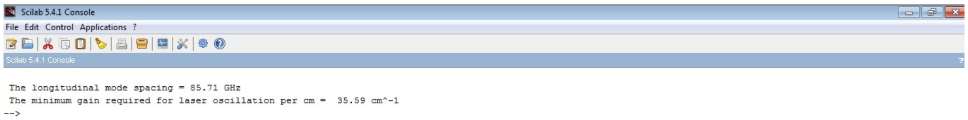

Figure 3.3: To calculate the longitudinal mode spacing and the minimum gain required for laser oscillation

```
printf('\n The population density of the excited
    level = %0.2f X 10^5 cm^(-3)',N2*10^-5);

// The calculation of this answer is wrong in the
    book
```

Scilab code Exa 3.3 To calculate the longitudinal mode spacing and the minimum gain required for laser oscillation

```
// Example No. 3.3
// To calculate the longitudinal mode spacing and
    the minimum gain required for laser oscillation
// Page no. 106

clc;
clear;

// Given data
```

```
c=3*10^8;    // The speed of ligth in air
L=500*10^(-6);    // The distance between mirrors
n=3.5;    // The refractive index
inlossdB=50;    // The internal loss in dB/cm
R1=0.3;    // The reflectivity of ligth wave which is reflected at A
R2=0.3;    // The reflectivity of ligth wave which is reflected at B

// The longitudinal mode spacing
deltaf=(c/(2*n*L))*10^-9;    // The longitudinal mode spacing
L=0.05;    // The distance between mirrors in cm
amir=(1/(2*L))*log(1/(R1*R2));    // The loss due to mirrors per cm
aint=log(10^(inlossdB/10));    // The coefficient of internal loss due to scattering

// The minimum gain required
g=aint+amir;    //
```

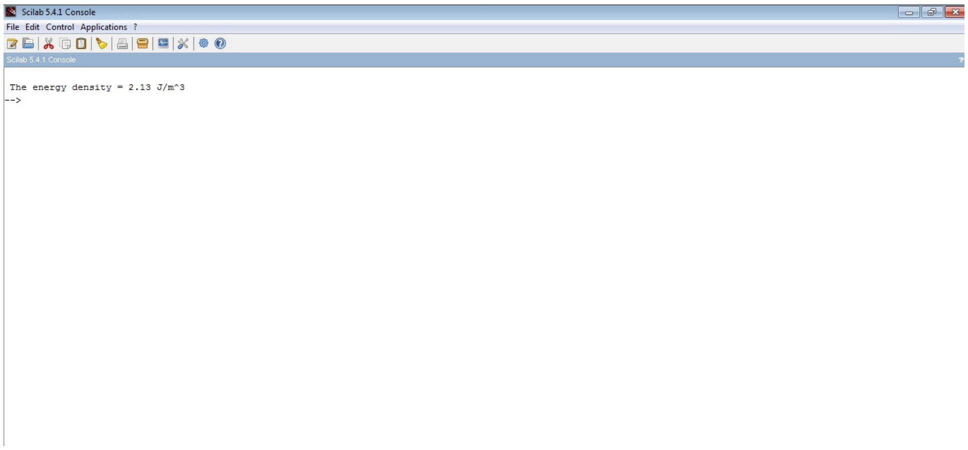

Figure 3.4: To calculate the energy density

```
    The minimum gain required for laser oscillation

// Displaying the result in command window
printf('\n The longitudinal mode spacing = %0.2f GHz
    ',deltaf);
printf('\n The minimum gain required for laser
    oscillation per cm =  %0.2f cm^-1',g);
```

Scilab code Exa 3.4 To calculate the energy density

```
// Example no. 3.4
// To calculate the energy density.
// Page no. 107

clc;
clear;

// Given data
```

Figure 3.5: To calculate the frequency of the electromagnetic wave emitted by stimulated emission

```
P=20*10^(-3);                                              //
    The mean power in W
A=100*10^(-12);                                            //
    The area perpendicular to the direction of light
    propagation in m^2
n=3.2;                                                     //
    Refractive index of gain medium
c=3*10^8;                                                  //
    Speed of ligth in m/s
I=P/A;                                                     //
    The optical intensity in W/m^2

// The energy density
u=(n*I)/c;                                                 //
    The energy density in J/m^3

// Displaying the result in command window
printf('\n The energy density = %0.2f J/m^3',u);
```

Scilab code Exa 3.5 To calculate the frequency of the electromagnetic wave emitted by stimulated emission

```
// Example no.3.5
// To calculate the frequency of the electromagnetic
    wave emitted by stimulated emission.
// Page no.110

clc;
clear;

// Given data
E=10^(-4);                                          // The
   energy difference between two levels in eV
E=10^(-4)*1.602*10^(-19);                           // The
   energy difference between two levels in J
h=1.054*10^(-34);                                   // The
   distance between two levels

// The frequency of the electromagnetic wave emitted
    by stimulated emission.
f=(E/(2*%pi*h))*10^-9;                              // The
   frequency of the electromagnetic wave emitted by
   stimulated emission in GHz

// Displaying the result in command window
printf('\n The frequency of the electromagnetic wave
    emitted by stimulated emission = %0.0f GHz',f);
```

Scilab code Exa 3.6 To calculate the band gap energy

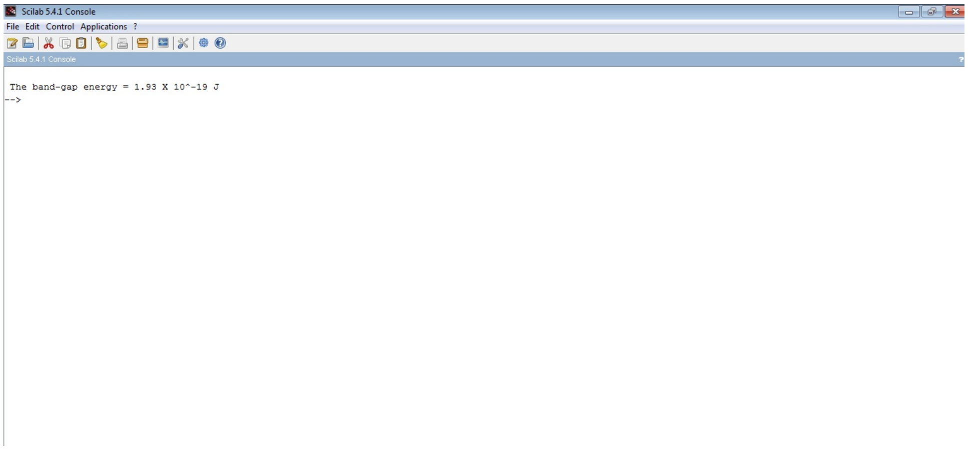

Figure 3.6: To calculate the band gap energy

```
// Example 3.6
// To calculate the band-gap energy.
// Page no.123

clc;
clear;

// Given data
m=9.109*10^(-31);                          // The electron rest mass in kg
meff1=0.07*m;                              // The effective mass of an electron in the conduction band
meff2=0.5*m;                               // The effective mass of an electron in the valence band
mr=(meff1*meff2)/(meff1+meff2);            // The reduced mass
hkl=7.84*10^(-26);                         // The electron momentum in kg.m/s
lambda=0.8*10^(-6);                        // The wavelength of electromagnetic wave in m
h=1.054*10^(-34);                          // The
```

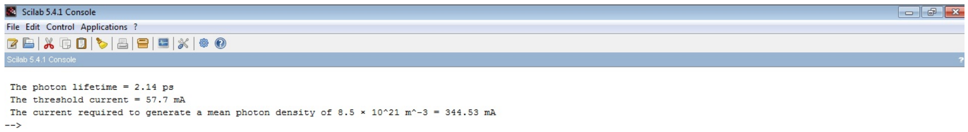

Figure 3.7: To calculate the photon lifetime the threshold current and the current required to generate a mean photon density

```
    distance between two levels
c=3*10^8;                                    //
    Speed of ligth in m/s
hw=(h*2*%pi*c)/lambda;                       // The
    poton energy in J

// The band-gap energy.
Eg=hw-(hkl^2/(2*mr));                        // The
    band-gap energy in J

// Displaying the result in command window
printf('\n The band-gap energy = %0.2f X 10^-19 J',
    Eg*10^19);
```

Scilab code Exa 3.7 To calculate the photon lifetime the threshold current and the current required to generate a mean photon density

```
// Example no.3.7
```

```
// To calculate (a) the photon lifetime, (b) the threshold current, and (c) the current required to generate a mean photon density of 8.5  10^21 m 3
// Page no.130

clc;
clear;

// Given data
w=3*10^(-6);                                    // The active area width in meter
d=0.3*10^(-6);                                  // The active area thickness in meter
L=500*10^(-6);                                  // The length
Te=1*10^(-9);                                   // Electron lifetime
Neth=0.8*10^(24);                               // Threshold electron density
aint=46*10^2;                                   // Internal cavity loss in m^-1
n=3.5;                                          // Refrective index of the medium
R1=0.65;                                        // The reflectivity of ligth wave which is reflected at A
R2=0.65;                                        // The reflectivity of ligth wave which is
```

```
reflected at B

// (a)The photon lifetime
amir=(1/(2*L))*log(1/(R1*R2));                    // The loss due to mirrors per m
c=3*10^8;                                          // Speed of ligth in m/s
v=c/n;                                             // Speed of ligth in medium (m/s)
Tp=1/(v*(aint+amir));                              // The photon lifetime in sec

// Displaying the result in command window
printf('\n The photon lifetime = %0.2f ps',Tp*10^12);

// (b)The threshold current
V=w*d*L;                                           // The active volume in m^3
q=1.602*10^(-19);                                  //The electron charge in C
Te=10^-9;                                          // The electron lifetime in sec
Ith=(Neth*q*V)/Te;                                 //The threshold current in mA

// Displaying the result in command window
printf('\n The threshold current = %0.1f mA',Ith*10^3);

```

```
// The answer calculated in book is wrong

// (c)The current required to generate a mean photon density of 8.5   10^21 m  3
Nph=8.5*10^21;                                                    //Mean photon density
Tph=Tp;                                                           // The photon lifetime in sec
I=(Ith+(Nph*q*V)/Tph);                                            //The current required to generate a mean photon density of 8.5   10^21 m  3

// Displaying the result in command window
printf('\n The current required to generate a mean photon density of 8.5   10^21 m^-3 = %0.2f mA',I*10^3)

// The answer calculated in book is wrong
```

Scilab code Exa 3.8 To find the wavelength of the light emitted

```
// Example no.3.8
// To find the wavelength of the light emitted
// Page no.133

clc;
clear;

// Given data
RspRst=2*10^14;
```

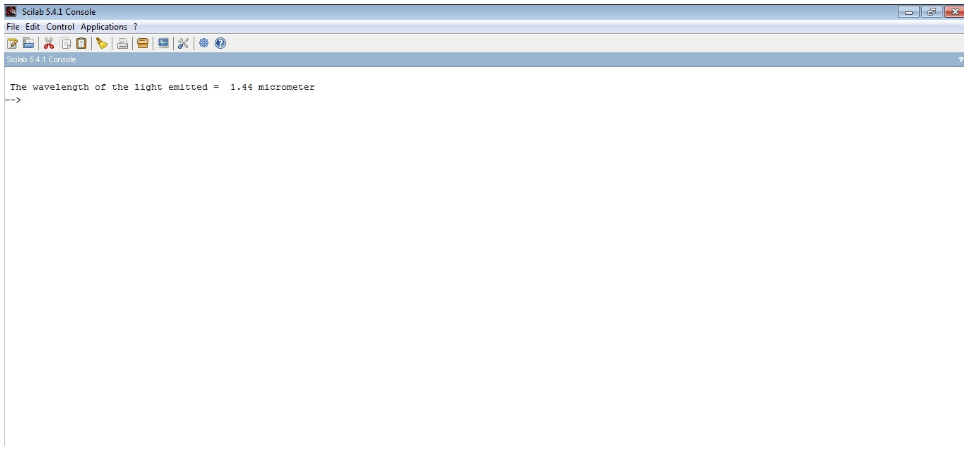

Figure 3.8: To find the wavelength of the light emitted

```
                                                         // The
   ratio of spontaneous emission rate to stimulated
   emission rate
T=30;
   // Temperature in degree celcius
kB=1.38*10^(-23);
                                                         // The
   Boltzmann s constant J/K
h=1.054*10^(-34);
                                                         // The
   distance between two levels
c=3*10^8;
   // Speed of ligth in air

T=T+273;
   // Temperature in Kelvin
w=(log(RspRst)*kB*T)/h;
                                                   // Frequency in
    Rad

```

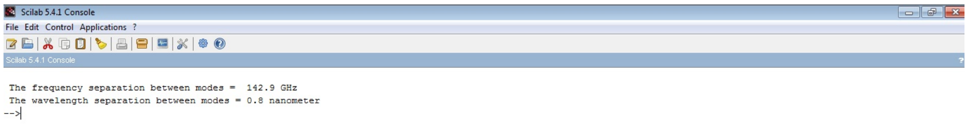

Figure 3.9: To calculate the frequency separation between modes the wavelength separation between modes

```
// The wavelength of the light emitted
lambda=(2*%pi*c)/w;                                    // The
    wavelength of the light emitted

// Displaying the result in command window
printf('\n The wavelength of the light emitted = %0
    .2f micrometer',lambda*10^6);
```

Scilab code Exa 3.9 To calculate the frequency separation between modes the wavelength separation between modes

```
// Example no.3.9
// To calculate a) the frequency separation between
    modes (b) the wavelength separation between modes
// Page no.133

clc;
```

```
clear;

// Given data
lambda=1.3*10^-6;                                         // Laser diode operating wavelength
L=300*10^-6;                                              // Cavity length
n=3.5;                                                    // Refractive index of active region
c=3*10^8;                                                 // Speed of ligth in air (m/s)

// a)The frequency separation between modes
deltaf=c/(2*n*L);                                         // The frequency separation between modes in GHz

// Displaying the result in command window
printf('\n The frequency separation between modes = %0.1f GHz',deltaf*10^-9);

// (b)The wavelength separation between modes
deltalambda=(lambda^2*deltaf)/c;                          // The wavelength separation between modes

// Displaying the result in command window
printf('\n The wavelength separation between modes = %0.1f nanometer',deltalambda*10^9);

// The wrong unit is givan in book
```

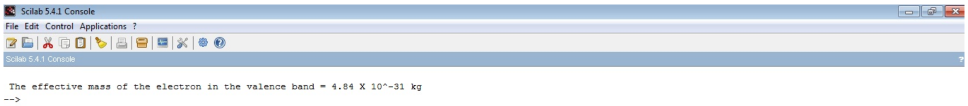

Figure 3.10: To calculate the effective mass of the electron in the valence band

Scilab code Exa 3.10 To calculate the effective mass of the electron in the valence band

```
// Example no.3.10
// To calculate the effective mass of the electron
   in the valence band.
// Page no.134

clc;
clear;

// Given data
Eg=1.18;                                               //
   Band gap in eV
Eg=1.18*1.602*10^-19;
                                          // Band gap in J
```

```
hk1=9*10^-26;                                           // The
    crystal momentum in Kg.m/s
h=1.054*10^(-34);                                       // The
    distance between two levels
f=3.94*10^14;                                           // Light
     wave of frequency
m=9.109*10^(-31);                                       // The
    electron rest mass in kg

mr=(hk1)^2/(2*(h*2*%pi*f-Eg));                  // The reduced mass in kg
meff1=0.07*m;                                           // The
    effective mass of an electron in the conduction
    band

// The effective mass of the electron in the valence
     band.
meff2=(mr*meff1)/(meff1-mr);                            //
    The effective mass of the electron in the valence
     band.

// Displaying the result in command window
printf('\n The effective mass of the electron in the
     valence band = %0.2f X 10^-31 kg',meff2*10^31);
// The answer is varrying due to round-off error
```

Scilab code Exa 3.11 To calculate the optical gain coefficient required to balance the cavity loss and the threshold electron density Ne

Figure 3.11: To calculate the optical gain coefficient required to balance the cavity loss and the threshold electron density Ne

```
// Example no.3.11
// To calculate (a) the optical gain coefficient  g
    required to balance the cavity loss and (b) the
   threshold electron density Ne
// Page no.135

clc;
clear;

// Given data
L=320*10^-6;
                                                    // Cavity length
R1=0.35;
                                                    // The reflectivity of ligth wave which is
   reflected at A
R2=0.35;
                                                    // The reflectivity of ligth wave which is
   reflected at B
aint=10^3;
```

```
    // Internal cavity loss in m^-1
c=3*10^8;    // Speed of ligth in air
Go=1.73*10^-12;    // Gain coefficient in m^3/s
Neo=3.47*10^23;    // The value of the carrier density at which the gain coefficient becomes zero in m^-3
n=3.3;    // Refractive index of medium

// (a) the optical gain coefficient  g required to balance the cavity loss
amir=(1/(2*L))*log(1/(R1*R2));    // The loss due to mirrors per m
acav=amir+aint;    // The total cavity loss coefficient
gammag=acav;    // The optical gain coefficient in m^-1

// Displaying the result in command window
printf('\n The optical gain coefficient = %0.2f X 10^3 m^-1',gammag*10^-3);

//(b) the threshold electron density Ne
v=c/n;    // Velocity of ligth in medium
Tph=1/(v*acav);
```

```
// The photon lifetime in sec
Neth=Neo+1/(Go*Tph);                                   // The threshold electron density Ne

// Displaying the result in command window
printf('\n The threshold electron density = %0.2f X 10^23 m^-3',Neth*10^-23);
```

Chapter 4

Optical Modulators and Modulation Schemes

Scilab code Exa 4.2 To calculate the voltage required to introduce a phase shift

```
// Example no.4.2
// To calculate the voltage required to introduce a
   phase shift of pi/2.
// Page no.152

clc;
clear;

// Given data
lambda0=1530*10^-9;                          // An electro-
   optic modulator operating wavelength
d=10*10^-6;                                          //
   Thickness
L=5*10^-2;
```

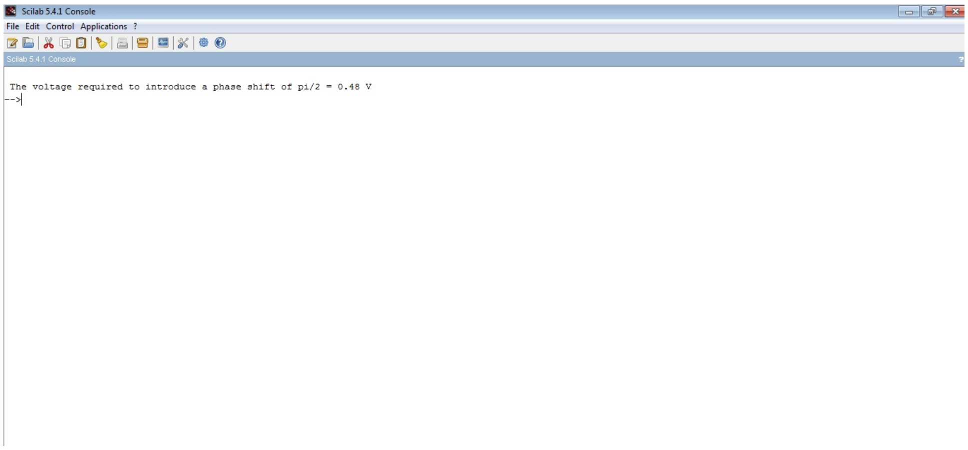

Figure 4.1: To calculate the voltage required to introduce a phase shift

```
                                                           //
   Length
n0=2.2;                                                       //
   Refractive index
r33=30*10^-12;                                            // Pockel
   coefficient in m/V
deltaphi=%pi/2;                                          // Phase
   shift
V=(deltaphi*lambda0*d)/(%pi*L*n0^3*r33);
              // The voltage required to introduce
   a phase shift of pi/2

//Displaying the result in command window
printf('\n The voltage required to introduce a phase
    shift of pi/2 = %0.2f V',V);
```

Chapter 5

Optical Receivers

Scilab code Exa 5.1 To calculate the photon incidence rate the photon absorption rate and the quantum efficiency

```
// Example no.5.1
// To calculate (a) the photon incidence rate, (b) the photon absorption rate, and, (c) the quantum efficiency.
// Page no.196

clc;
clear all;
// Given data
lambda=550*10^(-9);                    // The wavelength of electromagnetic wave in m
c=3*10^8;                              // Speed of ligth in air
h=6.626*10^(-34);                      // Planck's constant
alpha=10^4;                            // absorption coefficient
W=3*10^-4;                             // width
```

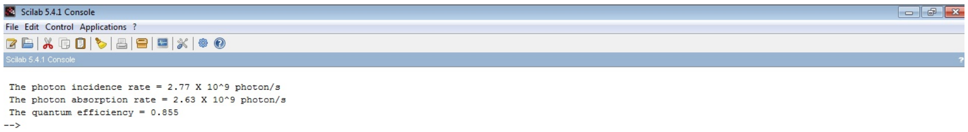

Figure 5.1: To calculate the photon incidence rate the photon absorption rate and the quantum efficiency

```
    of the active region
Pi=1*10^-9;                                          //
   optical power
eta=0.9;                                             // the
   fraction of photocarriers that contribute to the
   photocurrent
Rp=0;                                                // the
   power transmission coefficient at the
   air semiconductor interface

// (a) the photon incidence rate
Eph=(h*c)/lambda;                                    // The
   energy of a photon
Rincident=Pi/Eph;                                    // The
   photon incidence rate

// Display result on command window
printf('\n The photon incidence rate = %0.2f X 10^9
   photon/s',Rincident*10^-9);

// (b) the photon absorption rate
Rabs=(Rincident*(1-exp(-alpha*W)));                  // The
```

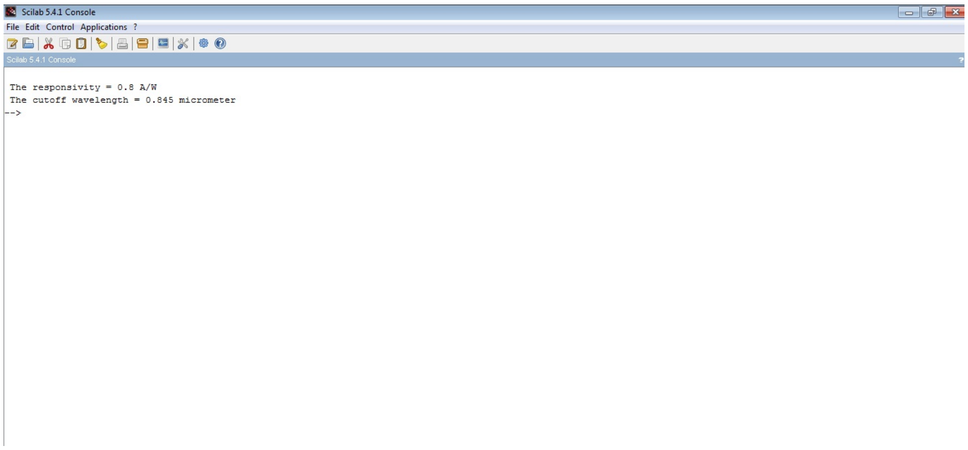

Figure 5.2: To calculate the responsivity R and the cutoff wavelength

```
    photon absorption rate

// Display result on command window
printf('\n The photon absorption rate = %0.2f X 10^9
     photon/s',Rabs*10^-9)

//c) the quantum efficiency
neta=(1-Rp)*eta*(1-exp(-alpha*W));           // The
    quantum efficiency

// Display result on command window
printf('\n The quantum efficiency = %0.3f',neta)
```

Scilab code Exa 5.2 To calculate the responsivity R and the cutoff wavelength

```
// Example no.5.2
// To calculate (a) the responsivity R and (b) the
    cutoff wavelength
```

```
// Page no.198

clc;
clear;

// Given data
neta=0.9;                                       // The quantum efficiency
Eg=1.42;                                        // The band-gap energy in eV
lambda=1.1;                                     // The operating (free-space) wavelength in micrometer

// (a) The responsivity
R=(neta*lambda)/1.24;                           // The responsivity in A/W

// Display result on command window
printf('\n The responsivity = %0.1f A/W',R)     //Wrong answer in book

// (b) The cutoff wavelength
lambdac=1.2/Eg;                                 //The cutoff wavelength in micrometer

// Display result on command window
printf('\n The cutoff wavelength = %0.3f micrometer',lambdac)    //Wrong answer in book
```

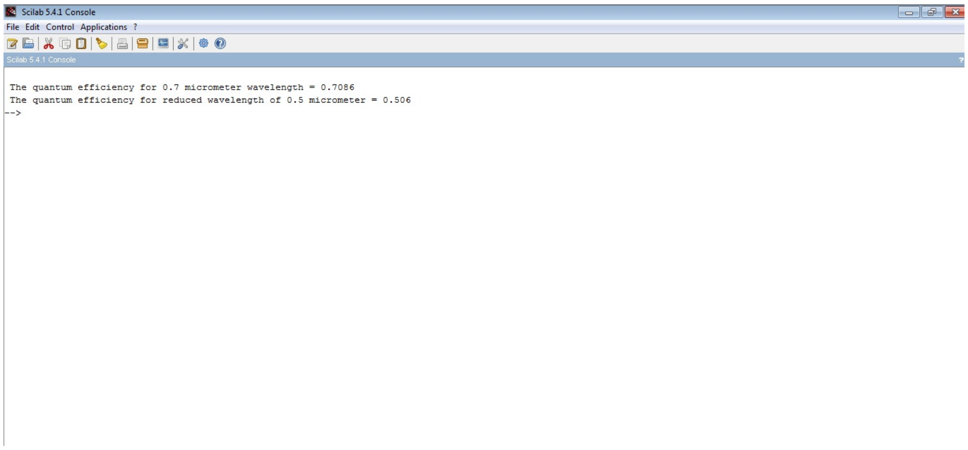

Figure 5.3: To find quantum efficiency at different wavelength and same responsivity

Scilab code Exa 5.3 To find quantum efficiency at different wavelength and same responsivity

```
// Example no.5.3
// To find quantum efficiency at different wavelength and same responsivity
// Page no.199

clc;
clear;

// Given data
lambda1=0.7;                                    // The radiation wavelength in micrometer
R=0.4;                                          // The responsivity in A/W
lambda2=0.5;                                    // The reduced wavelength in micrometer
neta1=(R*1.24)/lambda1;
```

Figure 5.4: To determine the refractive index and thickness of the antireflection coating

```
                                               // The quantum
    efficiency for 0.7 micrometer wavelength
neta2=neta1*(lambda2/lambda1);
                                    // The quantum efficiency
    for reduced wavelength 0.5 micrometer

// Display result on command window
printf('\n The quantum efficiency for 0.7 micrometer
     wavelength = %0.4f',neta1)
printf('\n The quantum efficiency for reduced
    wavelength of 0.5 micrometer = %0.3f',neta2)
```

Scilab code Exa 5.4 To determine the refractive index and thickness of the antireflection coating

```
// Example no.5.4
// To determine the refractive index and thickness
    of the antireflection coating
```

```
// Page no.199

clc;
clear;

// Given data
lambda=680*10^-9;                  // Wavelength of red ligth in meter
nair=1;                            // Refractive index of air
nsilicon=3.6;                      // Refractive index of silicon
nAR=sqrt(nair*nsilicon);           // Refractive index of antireflection coating
tAR=lambda/(4*nAR);                // Thickness of antireflection coating

// Display result on command window
printf('\n Refractive index of antireflection coating = %0.1f ',nAR)
printf('\n Thickness of antireflection coating = %0.0f nm',tAR*10^9)
```

Scilab code Exa 5.5 To calculate the inaccuracy with which resonator should be fabricated

```
// Example 5.5
// To calculate the inaccuracy with which resonator should be fabricated
// Page no.216

clc;
clear;
```

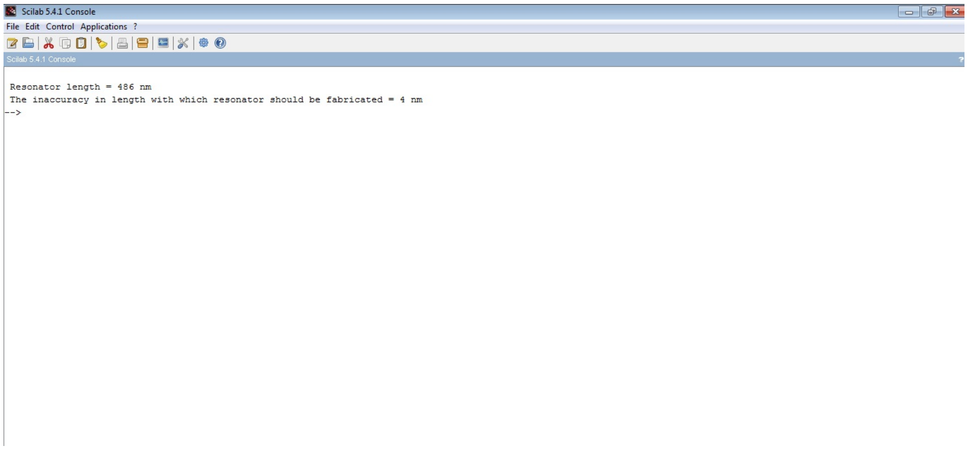

Figure 5.5: To calculate the inaccuracy with which resonator should be fabricated

```

// Given data
R1=0.9;                                          // Reflectivity at point A
integer=4;
n=3.5;                                           // Reflection index of silicon
F=%pi/(1-sqrt(R1));                              // The finesse of the resonator and also called as the ratio of the free spectral range
lambda0=850;                                     // Wavelength in nanometer
L=integer*lambda0/(2*n);                         // Resonator length in nanometer

// The inaccuracy with which resonator should be fabricated
deltaL=L*0.5/F;

// Display result on command window
printf('\n Resonator length = %0.0f nm',L)
printf('\n The inaccuracy in length with which
```

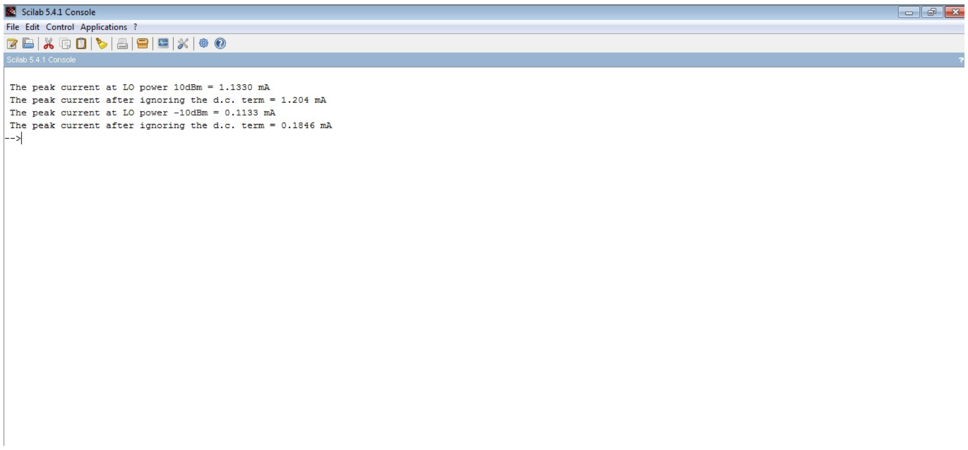

Figure 5.6: To find the peak current

```
    resonator should be fabricated = %0.0f nm',deltaL
    )
```

Scilab code Exa 5.6 To find the peak current

```
// Example no.5.6
// To find the peak current if (a) LO power = 10 dBm
   , (b) LO power =   10 dBm for the single-branch
   receiver
// Page no.229

clc;
clear;

// Given data
L=100;                                              //
   Length of fiber
loss=0.2*L;                                         //
   Total fiber loss
```

```
PtdBm=12;                                    // The peak power of the signal at the transmitter
R=0.9;                                       // Responsivity in A/W
PrdBm=PtdBm-loss;                            // The power at the receiver

// (a) the peak current LO power = 10 dBm
PLO1dBm=10;                                  // Power at local oscillator in dBm
PLO1=10^(0.1*PLO1dBm);                       // Power at local oscillator in mW
Pr=10^(0.1*PrdBm);                           // Power at receiver in mW
Id1=R*sqrt(Pr*PLO1);                         // The peak current at LO power = 10dBm
I1=R*Pr/2+R*sqrt(Pr*PLO1);                   // The peak current after ignoring the d.c. term

// Display result on command window
printf('\n The peak current at LO power 10dBm = %0.4f mA',Id1)
printf('\n The peak current after ignoring the d.c. term = %0.3f mA',I1)

// (b) the peak current LO power = -10 dBm
PLO2dBm=-10;                                 // Power at local oscillator in dBm
PLO2=10^(0.1*PLO2dBm);                       // Power at local oscillator in mW
Id2=R*sqrt(Pr*PLO2);                         // The peak current at LO power = -10dBm
I2=R*Pr/2+R*sqrt(Pr*PLO2);                   // The peak current after ignoring the d.c. term

// Display result on command window
printf('\n The peak current at LO power -10dBm = %0.4f mA',Id2)
```

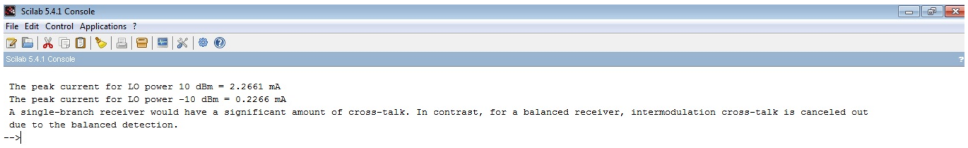

Figure 5.7: To find the peak current

```
printf('\n The peak current after ignoring the d.c.
    term = %0.4f mA',I2)
```

Scilab code Exa 5.7 To find the peak current

```
// Example no.5.7
// To find the peak current if (a) LO power = 10 dBm
    , (b) LO power =   10 dBm for the balanced
    receiver
// Page no.234

clc;
clear;

// Given data
L=100;                                               //
    Length of fiber
loss=0.2*L;                                          //
    Total fiber loss
```

```
PtdBm=12;                              // The peak power of the signal at the transmitter
R=0.9;                                 // Responsivity in A/W
PrdBm=PtdBm-loss;                      // The power at the receiver

// (a) the peak current LO power = 10 dBm
PLO1dBm=10;                            // Power at local oscillator in dBm
PLO1=10^(0.1*PLO1dBm);                 // Power at local oscillator in mW
Pr=10^(0.1*PrdBm);                     // Power at receiver in mW
Id1=2*R*sqrt(Pr*PLO1);                 // The peak current LO power = 10 dBm

// Display result on command window
printf('\n The peak current for LO power 10 dBm = %0.4f mA',Id1)

// (b) the peak current LO power = -10 dBm
PLO2dBm=-10;                           // Power at local oscillator in dBm
PLO2=10^(0.1*PLO2dBm);                 // Power at local oscillator in mW
Id2=2*R*sqrt(Pr*PLO2);                 // The peak current LO power = -10 dBm

// Display result on command window
printf('\n The peak current for LO power -10 dBm = %0.4f mA',Id2)

// comment on the intermodulation cross-talk in a single-branch receiver and the balanced receiver
printf('\n A single-branch receiver would have a significant amount of cross-talk. In contrast, for a balanced receiver, intermodulation cross-
```

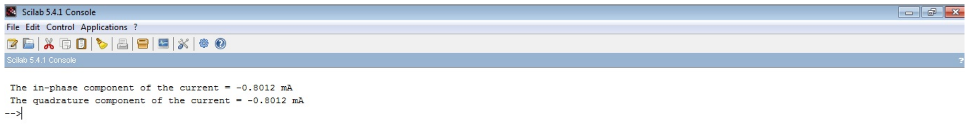

Figure 5.8: To find the in phase and quadrature components of the current of a balanced IQ receiver

```
    talk is canceled out \n due to the balanced
    detection.')
```

Scilab code Exa 5.8 To find the in phase and quadrature components of the current of a balanced IQ receiver

```
// Example no.5.8
// To find the in-phase and quadrature components of
    the current of a balanced IQ receiver.
// Page no.238

clc;
clear;

// Given data
PLO=10;                                          //
    Local oscillator power in mW from Example 5.7a
```

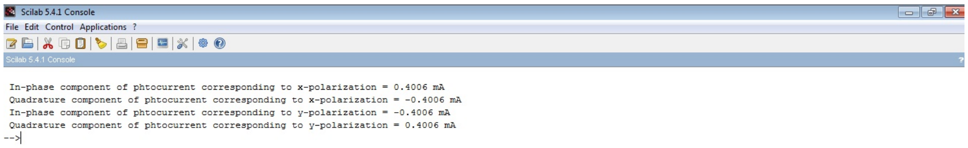

Figure 5.9: To find the in phase and quadrature components of the current of a polarization modulated QPSK signal

```
Pr=0.1585;                                    // Power at receiver in mW
R=0.9;                                        // Responsivity in A/W
st=complex((-1/sqrt(2)),(1/sqrt(2)));         // The QPSK transmitted signal
Ii=R*sqrt(Pr*PLO)*real(st);                   // The in-phase component of the current in mA
Iq=-R*sqrt(Pr*PLO)*imag(st);                  // The quadrature component of the current in mA

// Display result on command window
printf('\n The in-phase component of the current = %0.4f mA',Ii)
printf('\n The quadrature component of the current = %0.4f mA',Iq)
```

Scilab code Exa 5.9 To find the in phase and quadrature components of the current of a polarization modulated QPSK signal

```
// Example 5.9
// To find the in-phase and quadrature components of
    the current of a polarization modulated (PM)
   QPSK signal
// Page no. 241

clc;
clear;

// Given data
theta1=%pi/4;
Sx=expm(%i*theta1);

   // Signal data in x-polarization
theta2=(5*%pi)/4;
Sy=expm(%i*theta2);

   // Signal data in y-polarization
PLO=10;

   // Local oscillator power in mW from Example 5.8
Pr=0.1585;

   // Power at receiver in mW from Example 5.8
R=0.9;

   // Reflectivity

// The complex photocurrent corresponding to x-
   polarization
Ix= (R*sqrt(Pr*PLO))*Sx/2;
                                            // The
   complex photocurrent corresponding to x-
   polarization
Iix=real(Ix);
```

```
    // In-phase component of phtocurrent
    corresponding to x-polarization
Iqx=-imag(Ix);
    // Quadrature component of phtocurrent
    corresponding to x-polarization

// The complex photocurrent corresponding to y-
    polarization
Iy= (R*sqrt(Pr*PLO))*Sy/2;
                                                  // The
    complex photocurrent corresponding to y-
    polarization
Iiy=real(Iy);
    // In-phase component of phtocurrent
    corresponding to y-polarization
Iqy=-imag(Iy);
    // Quadrature component of phtocurrent
    corresponding to y-polarization

// Display result on command window
printf('\n In-phase component of phtocurrent
    corresponding to x-polarization = %0.4f mA',Iix);
printf('\n Quadrature component of phtocurrent
    corresponding to x-polarization = %0.4f mA',Iqx);
printf('\n In-phase component of phtocurrent
    corresponding to y-polarization = %0.4f mA',Iiy);
printf('\n Quadrature component of phtocurrent
    corresponding to y-polarization = %0.4f mA',Iqy);
```

Chapter 6

Optical Amplifiers

Scilab code Exa 6.1 Calculation of the gain

```
// Example 6.1
// Calculation of the gain
// Page no 249

clc;
clear;
close;

//Given data

n=1.5;                          // Refractive ondex of air
lambda=1550*10^-9;              // Wavelength
c=3*10^8;                       // Velocity of light
p=5.73*10^-17;                  // Power spectral density
h=6.63*10^-34                   // Planck constant


// Gain
f=c/lambda;
```

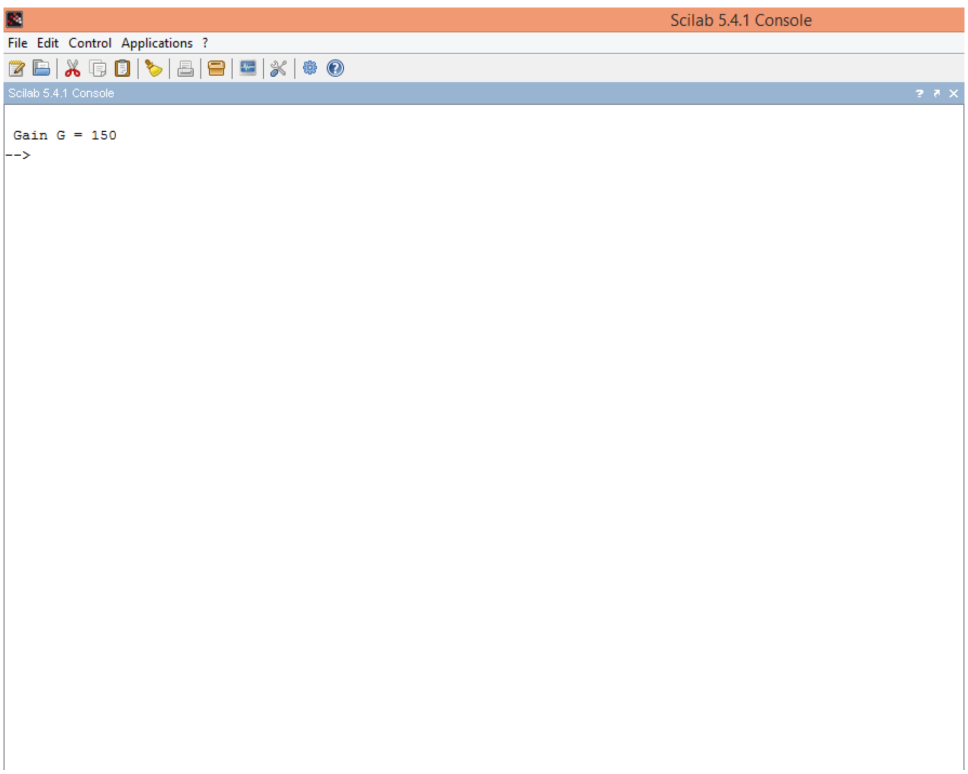

Figure 6.1: Calculation of the gain

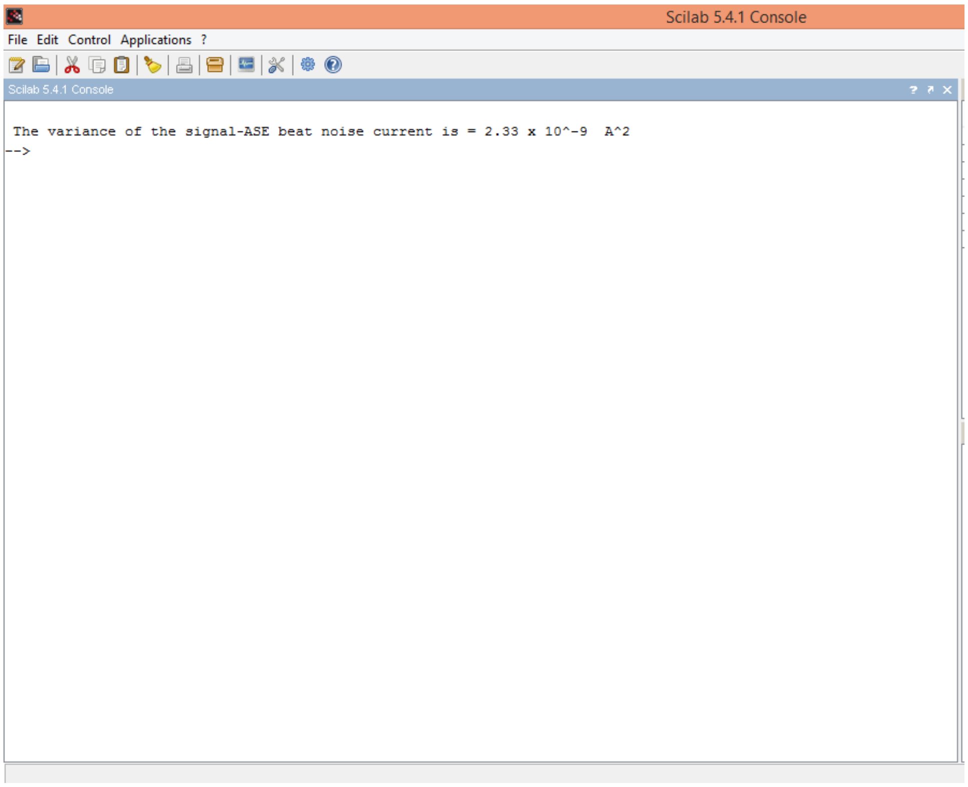

Figure 6.2: Calculation of the variance of the signal ASE beat noise

```

G=(p/(2*n*h*f))+1;

//Displaying results in the command window
printf("\n Gain G = %0.0f ",G);
```

Scilab code Exa 6.2 Calculation of the variance of the signal ASE beat noise

```
// Example 6.2
```

```
// Calculation of the variance of the signal ASE
   beat noise
// Page no 255

clc;
clear;
close;

//Given data

a=1.3*10^-16;                          // PSD of an amplifier
f=7*10^9;                             // Cut off frequency
Pi=10*10^-6;                          // Input power
R=0.8;                               // Responsivity
G=20;                                // Gain of an amplifier

// The variance of the signal ASE beat noise
G=10^(G/10);
P=G*Pi;

r=4*R^2*P*a*f;
r=r*10^9;




//Displaying results in the command window
printf("\n The variance of the signal ASE beat
   noise current is = %0.2f x 10^-9  A^2",r);


// The answers vary due to round off error
```

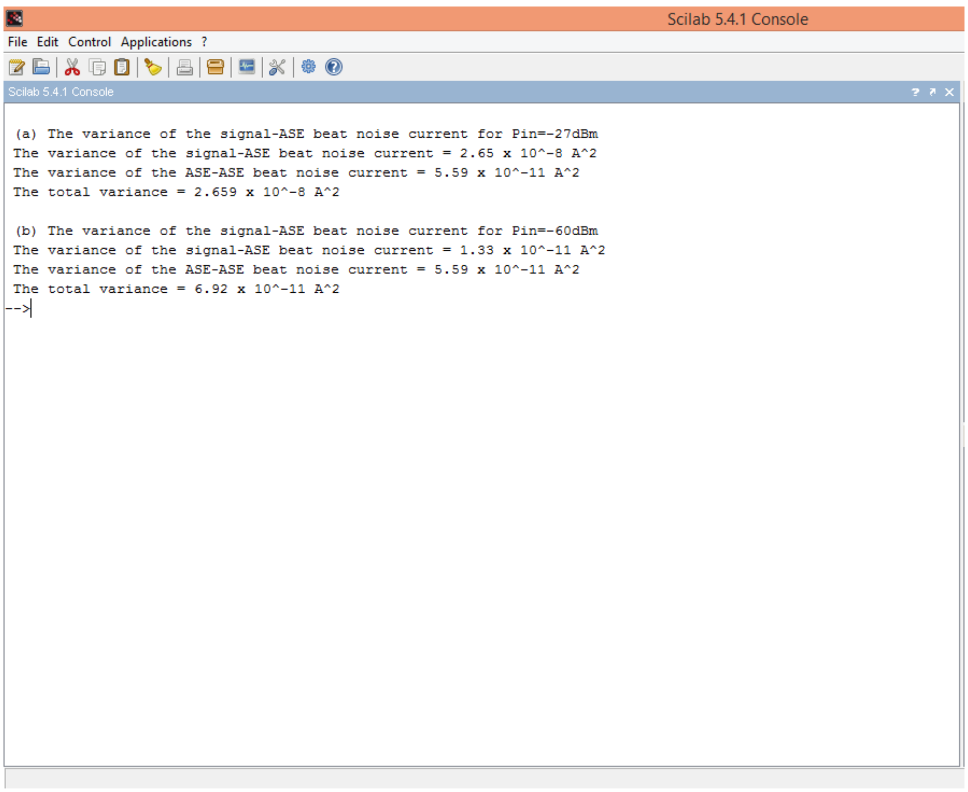

Figure 6.3: Calculation of the the variance of the signal ASE beat noise current the variance of the ASE ASE beat noise current and the total variance

Scilab code Exa 6.3 Calculation of the the variance of the signal ASE beat noise current the variance of the ASE ASE beat noise current and the total variance

```
// Example 6.3
// Calculation of the (a) the variance of the
   signal ASE beat noise current, (b) the variance
    of the ASE ASE beat noise current, and (c) the
    total variance.
// Page no 257

clc;
clear;
close;

//Given data

G=30;          // Gain

nsp=5;
R=0.8;
f1=16*10^9;
fe=9*10^9;
//Pi1=-60;
c=3*10^8;                // Velocity of light
h=6.63*10^-34           // Planck constant
lambda=1530*10^-9;      // Wavelegth
Pi1=-27;               // Input power
Pi2=-60;
f0=16*10^9;


//(a) The variance of the signal ASE beat noise
   current for Pin=-27dBm
Po=G+Pi1;
Po=10^(Po/10);
Po=Po*10^-3;
G=10^(G/10);
```

```
f=c/lambda;
r=nsp*h*f*(G-1);
B=8*10^9;
//B=min(f/2,fe);
r0=4*R^2*Po*r*B;
//r0=r0*10^12;

//(b) The variance of the ASE ASE beat noise current

r1=R^2*r^2*((2*f0)-fe)*fe;

//r1=r1*10^11;
// (c) The total variance.

rt=r0+r1;

// Displaying results in the command window
printf("\n (a) The variance of the signal ASE beat noise current for Pin=-27dBm");

printf("\n The variance of the signal ASE beat noise current = %0.2f x 10^-8 A^2",r0*10^8);
printf("\n The variance of the ASE ASE beat noise current = %0.2f x 10^-11 A^2",r1*10^11);

printf("\n The total variance = %0.3f x 10^-8 A^2",rt*10^8);
// The answers vary due to round off error


//Given data

G=30;        // Gain
nsp=5;
R=0.8;        //
f1=16*10^9;
fe=9*10^9;
```

```
//Pi1=-60;
c=3*10^8;     // Velocity of light
h=6.63*10^-34     // Planck constant
lambda=1530*10^-9;  // Wavelegth
Pi1=-27;                // Input power
Pi2=-60;
f0=16*10^9;

//(b) The variance of the signal ASE beat noise
   current for Pin=-60dBm
Po2=G+Pi2;
Po=10^(Po2/10);
Po=Po*10^-3;
G=10^(G/10);
f=c/lambda;
r=nsp*h*f*(G-1);
B=8*10^9;
//B=min(f/2,fe);
r0=4*R^2*Po*r*B;
//r0=r0*10^12;

//(b) The variance of the ASE ASE beat noise
   current

r1=R^2*r^2*((2*f0)-fe)*fe;

//r1=r1*10^11;
// (c) The total variance.

rt=r0+r1;

// Displaying results in the command window
printf("\n \n (b) The variance of the signal ASE
   beat noise current for Pin=-60dBm");

printf("\n The variance of the signal ASE beat
   noise current = %0.2f x 10^-11 A^2",r0*10^11);
printf("\n The variance of the ASE ASE beat noise
```

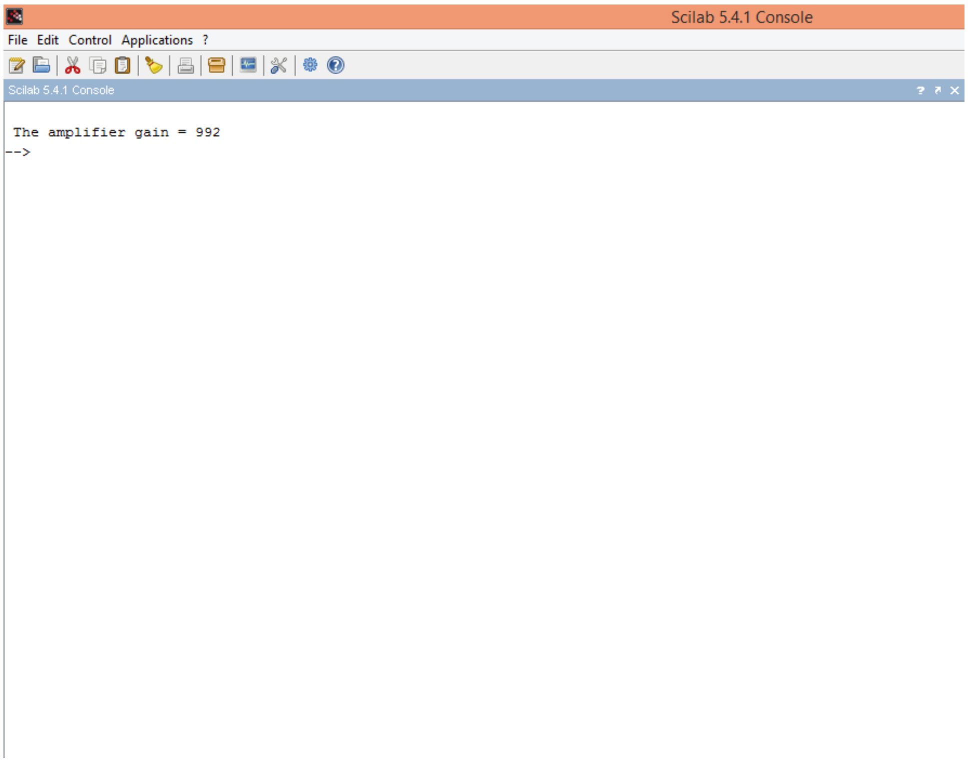

Figure 6.4: Calculation of the amplifier gain

```
    current = %0.2f x 10^-11 A^2",r1*10^11);

printf("\n The total variance = %0.2f x 10^-11 A^2 "
    ,rt*10^11);
```

Scilab code Exa 6.4 Calculation of the amplifier gain

```
// Example 6.4
// Calculation of the amplifier gain
// Page no 262
```

```
clc;
clear;
close;

//Given data

Po=0;                      // Signal output of amplifier
//f=7*10^9;                // Cut off frequency
B=7.5*10^9;                // Bandwidth
R=0.9;                     // Responsivity
c=3*10^8;                  // Velocity of light
lambda=1550*10^-9;         // Operating frequency
fn=4.5;                    // Noise figure
Ro=0.066*10^-3;            // Beat noise current
h=6.626*10^-34;            // Planck constant

// The amplifier gain
P=10^(Po/10)*10^-3;
r=Ro^2/(4*R^2*B*P);
fn=10^(fn/10);
f=c/lambda;
G=(1/fn)*(((2*r)/(h*f))+1);




//Displaying results in the command window
printf("\n The amplifier gain = %0.0f  ",G);


// The answers vary due to round off error
```

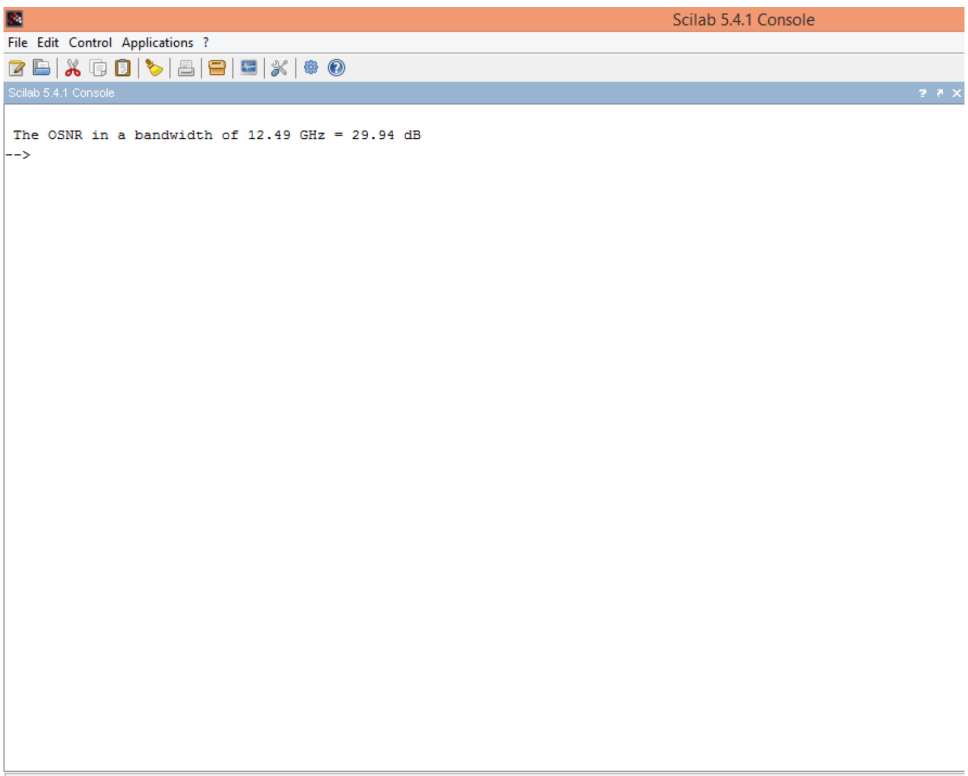

Figure 6.5: Calculation of the OSNR in a given bandwidth

Scilab code Exa 6.5 Calculation of the OSNR in a given bandwidth

```
// Example 6.5
// Calculation of the OSNR in a bandwidth of 12.49
    GHz.
// Page no 263

clc;
clear;
close;

//Given data

G=25;                       // Gain
c=3*10^8;                   // Velocity of light
h=6.63*10^-34               // Planck constant
lambda=1545*10^-9;          // Wavelegth
Pi=-22;                    // Input power
fn=6;
B=12.49*10^9;

// The OSNR in a bandwidth of 12.49 GHz
Po=G+Pi;
Po=10^(Po/10);
Po=Po*10^-3;
fn=10^(fn/10);
G=10^(G/10);
f=c/lambda;
r=(G*fn-1)*(h*f/2);
O=Po/(2*r*B);
O=10*log10(O);

// Displaying results in the command window
printf("\n The OSNR in a bandwidth of 12.49 GHz = %0
    .2f dB",O);
```

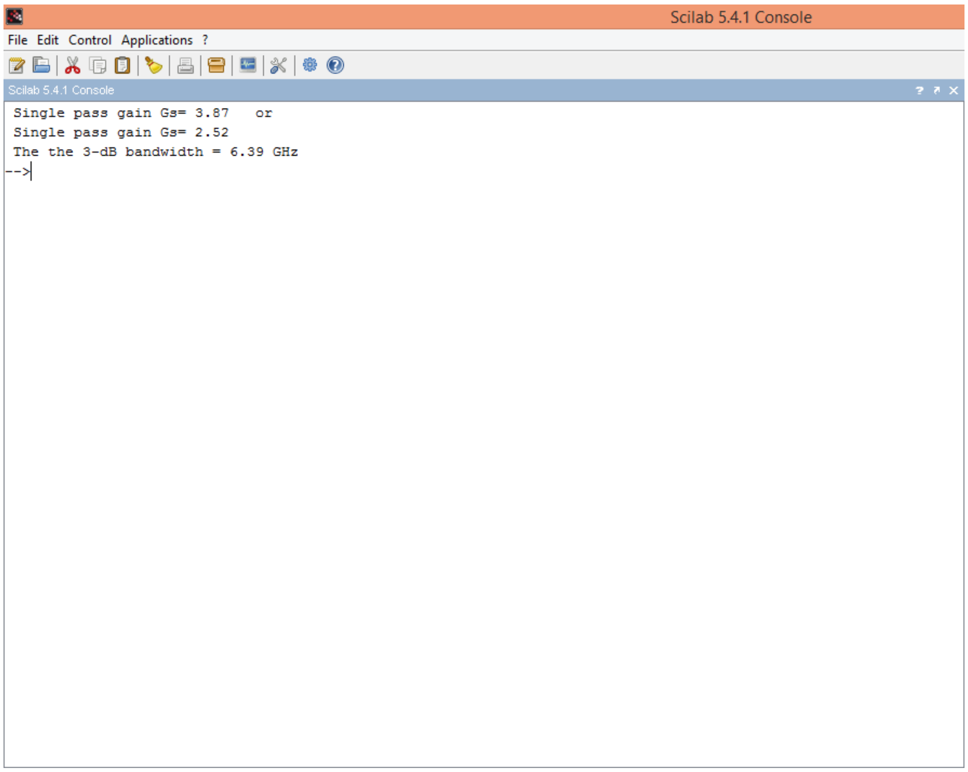

Figure 6.6: Calculation of the single pass gain and 3 db bandwidth

```
// The answers vary due to round off error
```

Scilab code Exa 6.6 Calculation of the single pass gain and 3 db bandwidth

```
// Example 6.6
// Calculation of the single-pass gain and 3-dB
   bandwidth
// Page no 268

```

```
clc;
clear;
close;

//Given data

c1=3*10^8;                    // Velocity of light
f=7*10^9;                     // Cut off frequency
L=500*10^-6;                  // Input power
Gp=15;                        // Peak gain
n=3.2;
Gs=2.52;
R=0.32;
a=0.1024;
b=-0.6546;
c=1;

// The single-pass gain

x1 =( -1*b+ sqrt ((b ^2) -4*a*c)) /(2* a); // 1st root
x2 =( -1*b- sqrt ((b ^2) -4*a*c)) /(2* a); // 2nd root

// The 3-dB bandwidth
G=10^(Gp/10);
x=(1-(R*x2))/(2*sqrt(R*x2));
f=(c1/(%pi*L*n))*asin(x);
// f=f*10^-9;

// Displaying results in the command window

printf ( 'Single pass gain Gs= %0.2f   or' , x1);
printf ( ' \n Single pass gain Gs= %0.2f ' , x2);
printf("\n The 3-dB bandwidth = %0.2f GHz ",f*10^-9);


```

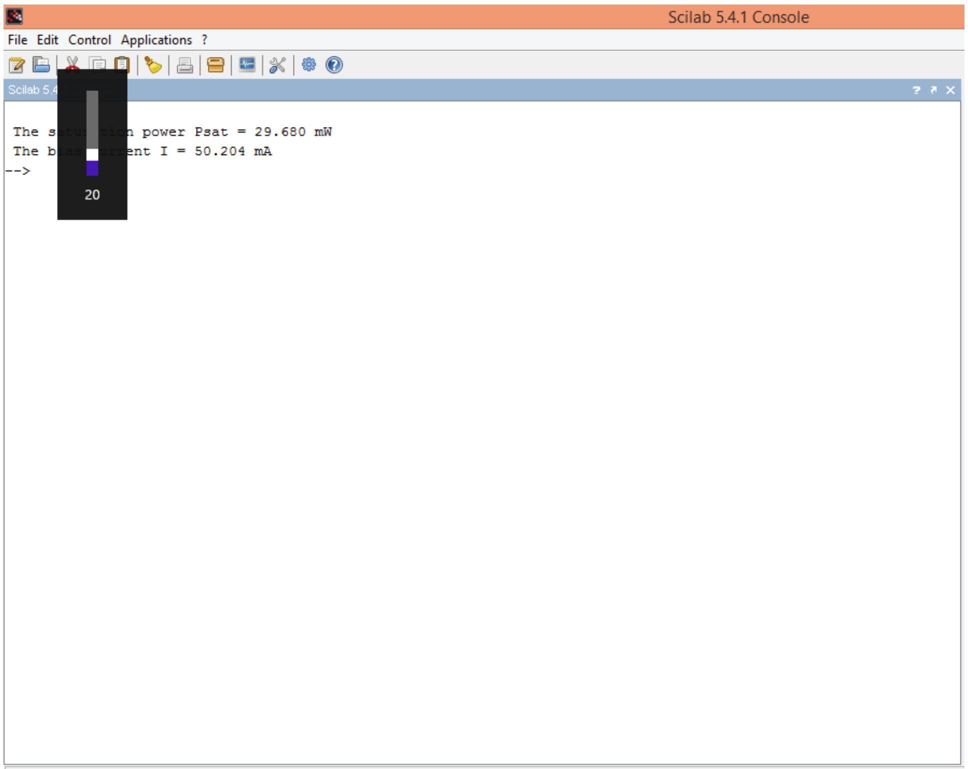

Figure 6.7: Calculation of the saturation power and the bias current I

```
// The answers vary due to round off error
```

Scilab code Exa 6.7 Calculation of the saturation power and the bias current I

```
// Example 6.6
// Calculation of (a) the saturation power and (b)
   the bias current I
// Page no 273

```

```
clc;
clear;
close;

//Given data

c=3*10^8;                     // Velocity of light
lambda=1530*10^-9;           // Wavelength
t=0.3                        // Overlap factor
r=7.3*10^-20;                // Gain cross section
r0=1*10^-9;                 // Carrier lifetime
q=1.609*10^-19;
v=7.5*10^-16;               // Active volume
h=6.63*10^-34               // Planck constant
A=5*10^-6;                  // Effective area
g=4.82*10^3;                // Small signal gain
    coeffifient
N=3.5*10^23;                //

// (a) the saturation power and


f=c/lambda;
Ps=(h*f*A)/(t*r*r0);
Ps=Ps*10^-3;

// (b) the bias current I

I=(g/(r*r0)+N/r0)*q*v;
I=I*10^3;
// Displaying results in the command window
printf("\n The saturation power Psat = %0.3f mW ",Ps
   );

printf("\n The bias current I = %0.3f mA ",I);


// The answers vary due to round off error
```

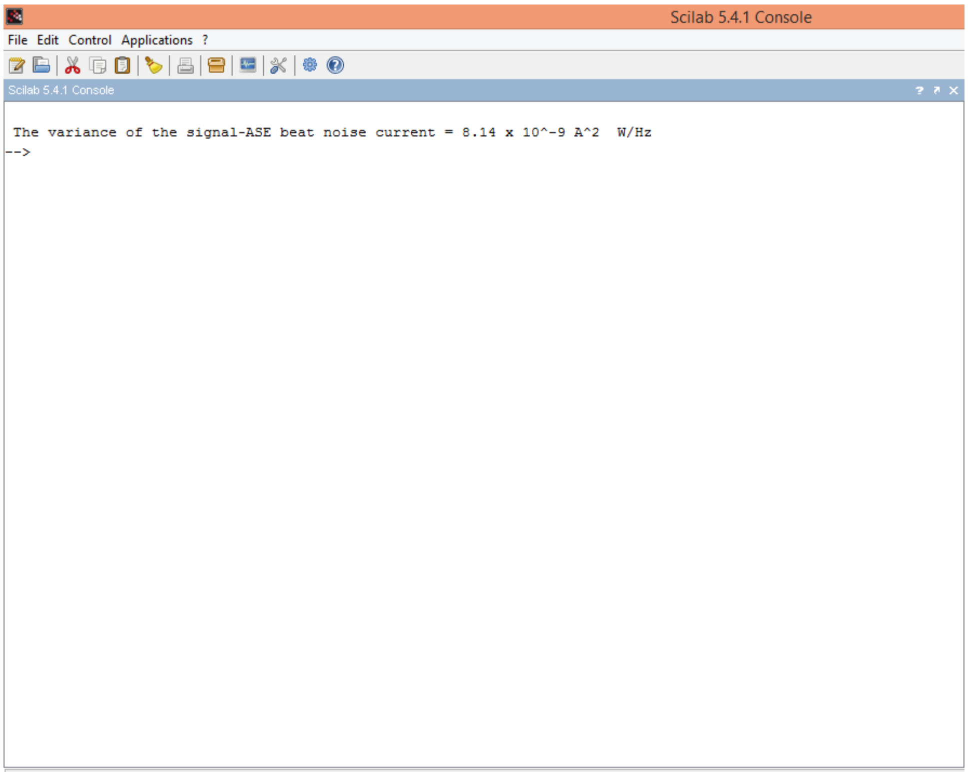

Figure 6.8: Calculation of the variance of the signal ASE beat noise current

---

Scilab code Exa 6.9 Calculation of the variance of the signal ASE beat noise current

```
// Example 6.9
// Calculation of the variance of the signal ASE
   beat noise current.
// Page no 290

clc;
```

```
clear;
close;

//Given data

si=30;                    // Electrical SNRs at the amplifier input
so=25;                   // Electrical SNRs at the amplifier output
p=0;                     // Signal power at output
r=-126;                 // Signal power at input
R=0.9;                  // Planck constant
f=195*10^12;          // Frequency
b=20*10^9;            // Bandwidth

// The variance of the signal ASE beat noise current
p1=10^(p/10)*10^-3;
rn=10^(r/10)*10^-3;
r1=rn*b;
r0=2*R^2*p1*r1;


//Displaying results in the command window
printf("\n The variance of the signal ASE beat noise current = %0.2f x 10^-9 A^2  W/Hz",r0*10^9);


// The value of noise power given in example as -126 but for calculation it is taken as -128 in book. Therefore answer is varying.
```

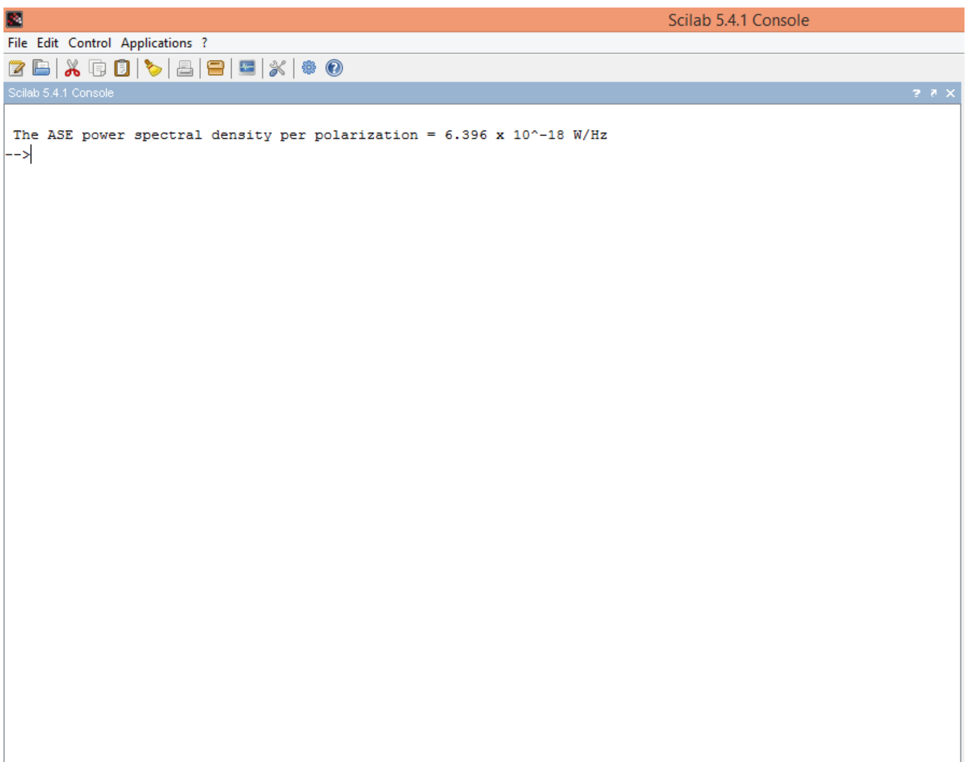

Figure 6.9: Calculation of the ASE power spectral density per polarization

Scilab code Exa 6.12 Calculation of the ASE power spectral density per polarization

```
// Example 6.12
// Calculation of the ASE power spectral density per
    polarization.
// Page no 296

clc;
clear;
close;

//Given data

si=30;                  // Electrical SNRs at the
   amplifier input
so=25;                  // Electrical SNRs at the
   amplifier output
po=2;                   // Signal power at output
pi=-13;                 // Signal power at input
h=6.626*10^-34;         // Planck constant
f=195*10^12;

// The ASE power spectral density per polarization
fn=si-so;
fn=10^(fn/10);
G=po-pi;
G=10^(G/10);
r=(h*f*(G*fn-1))/2;
r=r*10^18;

//Displaying results in the command window
printf("\n The ASE power spectral density per
   polarization = %0.3f x 10^-18 W/Hz ",r);
```

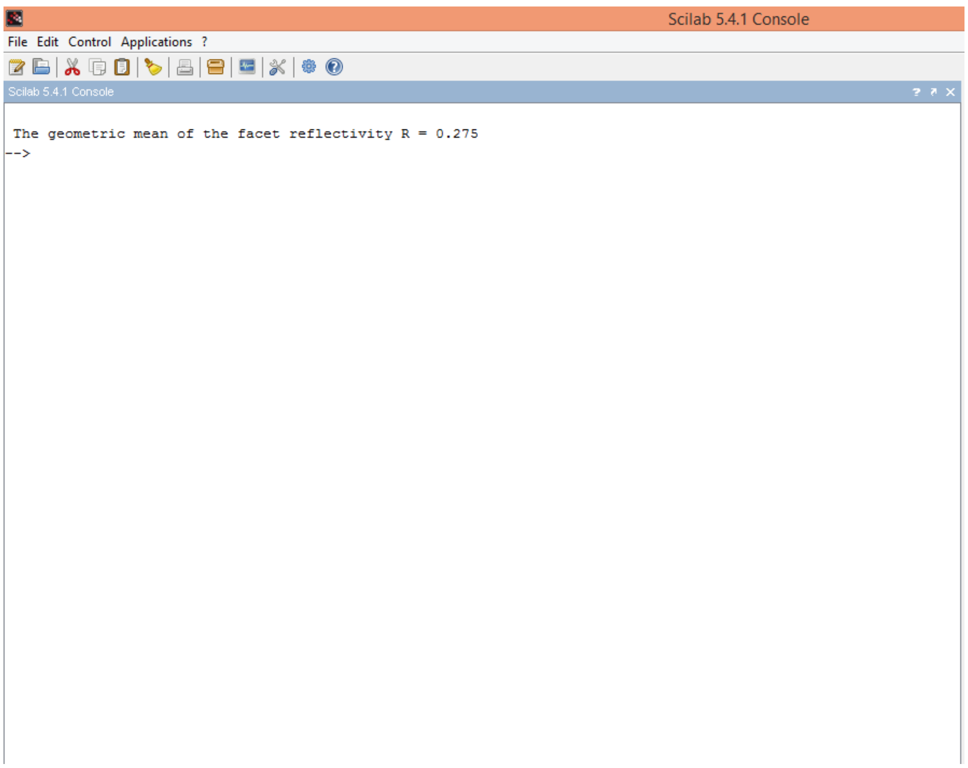

Figure 6.10: Calculation of the geometric mean of the facet reflectivity R

Scilab code Exa 6.13 Calculation of the geometric mean of the facet reflectivity R

```
// Example 6.13
// Calculation of the geometric mean of the facet
   reflectivity R
// Page no 296

clc;
clear;
close;

//Given data
Gm=20;
G1=5;

// The geometric mean of the facet reflectivity R
Gmax=10^(Gm/10);                          // Peak Gain
Gs=10^(G1/10);                            // Single pass
   gain
R=(sqrt(Gs)-10)/(sqrt(Gs)-Gs*10);




//Displaying results in the command window
printf("\n The geometric mean of the facet
   reflectivity R = %0.3f  ",R);
```

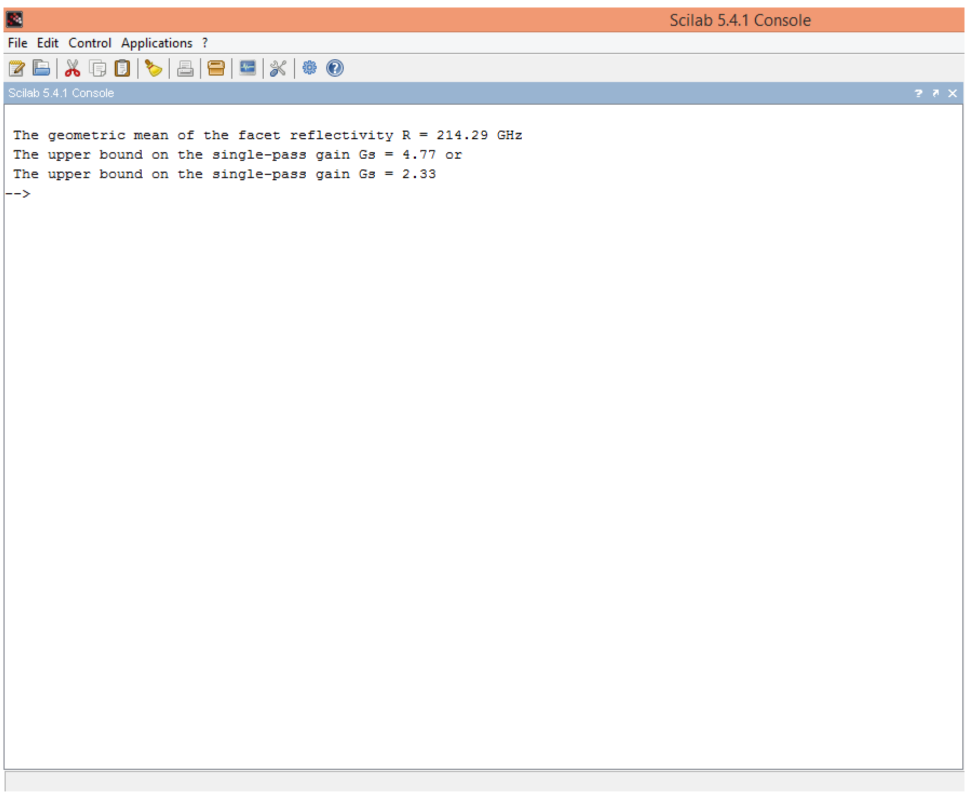

Figure 6.11: Calculation of the upper bound on the single pass gain

Scilab code Exa 6.14 Calculation of the upper bound on the single pass gain

```
// Example 6.13
// Calculation of the upper bound on the single-pass gain
// Page no 297

clc;
clear;
close;

//Given data

n=3.5;                          // Refractive index
c1=3*10^8;                      // Velocity of light
L=200*10^-6;                   // Amplifier length
a=0.09;
b=-(1.2*0.1805^2+0.6);
c=1;

// The geometric mean of the facet reflectivity R
f=c1/(2*n*L);

x1 =( -1*b+ sqrt ((b ^2) -4*a*c)) /(2* a); // 1 s t r o o t
x2 =( -1*b- sqrt ((b ^2) -4*a*c)) /(2* a); // 2nd r o o t



//Displaying results in the command window
printf("\n The geometric mean of the facet reflectivity R = %0.2f GHz ",f*10^-9);
printf("\n The upper bound on the single-pass gain Gs = %0.2f or  ",x1);
printf("\n The upper bound on the single-pass gain Gs = %0.2f  ",x2);
```

```
// The answers vary due to round off error
```

Chapter 7

Transmission System Design

Scilab code Exa 7.1 Compuatation of the lower limit on the transmitter power

```
// Example 7.1
// Compuatation of the lower limit on the
   transmitter power
//
// Page no. 305

clc;
clear;
close;

//Given data
q=1.6*10^-19;
R=1;
B=7*10^9;
c=3*10^8;            // Velocity of light
h=6.62*10^-34;      // Planck constant
Q=6;
k=1.38*10^-23;     // Boltzman constant
```

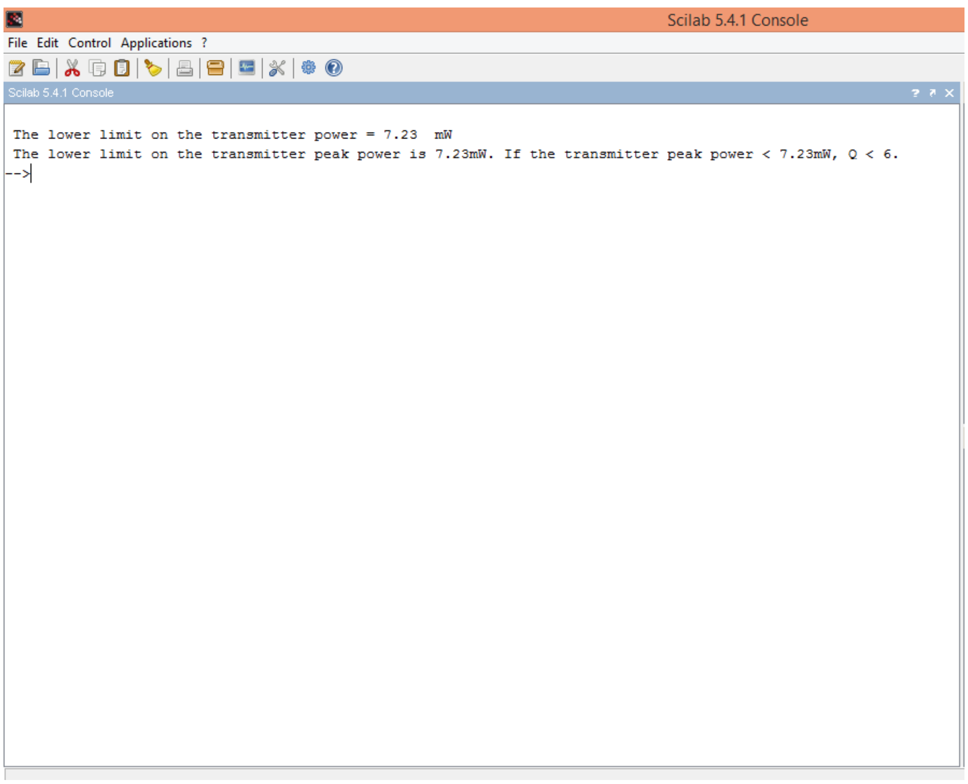

Figure 7.1: Compuatation of the lower limit on the transmitter power

```
T=298;
Rl=50;
alpha=0.046;        // Fiber loss coefficient
L=130;              // Length


// The lower limit on the transmitter power
a=2*q*R*B;
b=(4*k*T*B)/Rl;
p=(2*sqrt(b)/R*Q)+((a*Q^2)/R^2);
Pi=p*exp(alpha*L);

//Displaying the result in command window
printf("\n The lower limit on the transmitter power
    = %0.2f  mW",Pi*10^3);
printf("\n The lower limit on the transmitter peak
    power is 7.23mW. If the transmitter peak power <
    7.23mW, Q < 6.");

// The answer vary due to round off error
```

Scilab code Exa 7.2 To calculate exact and approximate Q factor if the signal is OOK and PSK

```
// Example no. 7.2
// To calculate exact and approximate Q-factor if
    the signal is (a)OOK, (b) PSK
// Page no. 311

clc;
clear;

// Given data
```

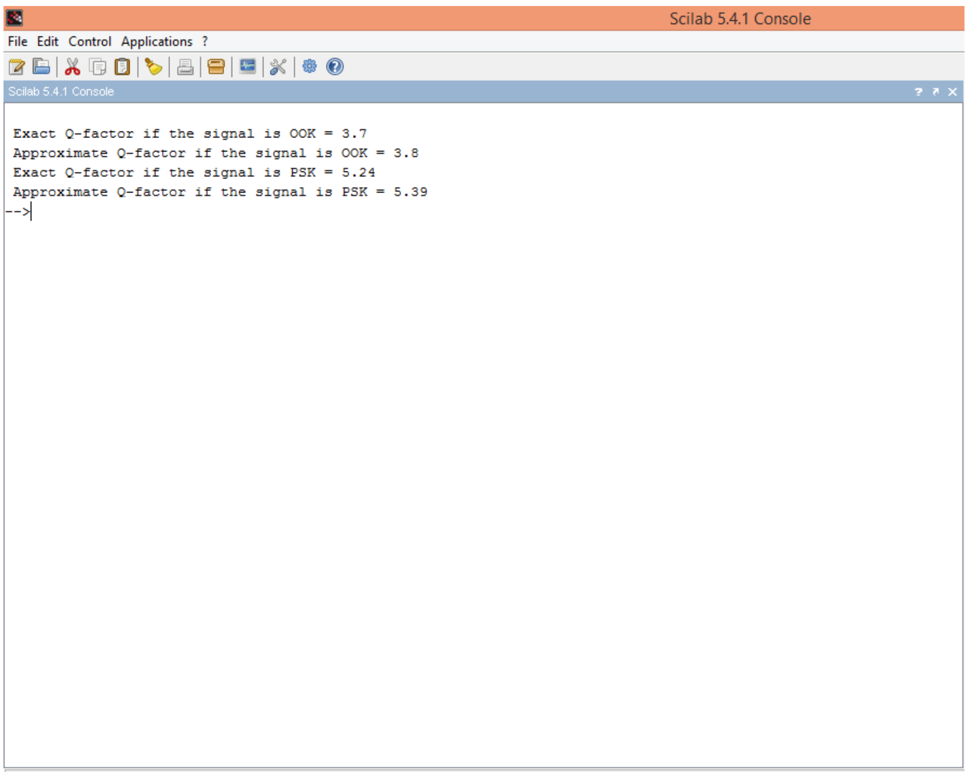

Figure 7.2: To calculate exact and approximate Q factor if the signal is OOK and PSK

```
lambda=1.55*10^(-6);                                   // Wavelength of given signal
meanPin=1;                                             // Mean fiber launch power in dBm
alpha=0.2;                                             // fiber loss in dB/km
l=240;                                                 // fiber length in km
neta=0.7;                                              // quantum efficiency
T = 290;                                               // Tempearture in  K
RL=100;                                                // Length resistance in
PLOdBm = 10;                                           // Power at local oscillator in dBm
Be = 7.5*10^9;                                         // Efficient bandwidth of filter in Hz
c=3*10^8;                                              // Speed of ligth in air in m/s
loss=alpha*l;                                          // Total fiber loss
q=1.602*10^(-19);                                      // Charge of electron
h=6.626*10^(-34);
```

```
// Planck constant
kB=1.38*10^(-23);
// Bolzman constant

f=c/lambda;
// mean frequency
R=(neta*q)/(h*f);
// Responsivity

//For OOK
Pin=10*log10(2)+meanPin;
// peak power in dBm
P1rdBm=Pin-loss;
// received peak power in dBm
P1r=(10^(P1rdBm/10))*10^(-3);
// received peak power in W
PLO=(10^(PLOdBm/10))*10^(-3);
// Power at local oscillator in W
I1=2*R*sqrt(P1r*PLO);
// mean of bit 1
sigma1=2*q*Be*R*(P1r+PLO)+(4*kB*T*Be)/RL;
// Square of variance of bit 1
I0=0;
// mean of bit 0
sigma0=sigma1;
// Square of variance of bit 0
Q1=(I1-I0)/(2*sqrt(sigma1));
```

```
                                                    // Exact Q-factor
Q2=sqrt((neta*P1r)/(2*h*f*Be));                     // Approximate Q-factor

// Displaying the result in command window
printf('\n Exact Q-factor if the signal is OOK = %0.1f',Q1);
printf('\n Approximate Q-factor if the signal is OOK = %0.1f',Q2);

// For PSK
P1rdBm=meanPin-loss;                                // received peak power in dBm
P1r=(10^(P1rdBm/10))*10^(-3);                       // received peak power in W
I1=2*R*sqrt(P1r*PLO);                               // mean of bit 1
sigma1=2*q*Be*R*(P1r+PLO)+(4*kB*T*Be)/RL;           // Square of variance of bit 1
I0=-I1;                                             // mean of bit 0
sigma0=sigma1;                                      // Square of variance of bit 0
Q1=I1/sqrt(sigma1);                                 // Exact Q-factor
Q2=sqrt((2*neta*P1r)/(h*f*Be));                     // Approximate Q-factor

```

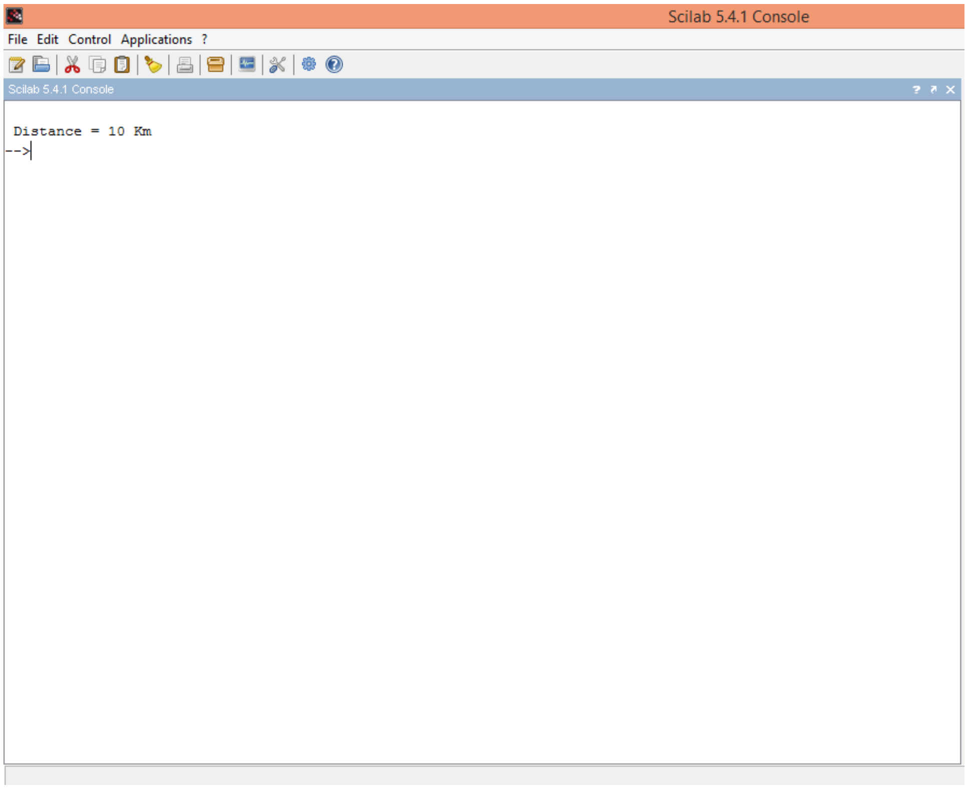

Figure 7.3: Calculation of the distance

```
// Displaying the result in command window
printf('\n Exact Q-factor if the signal is PSK = %0
    .2f',Q1);
printf('\n Approximate Q-factor if the signal is PSK
     = %0.2f',Q2);
```

Scilab code Exa 7.3 Calculation of the distance

```
// Example 7.3
// Calculation of the distance.
```

```
// Page no 315

clc;
clear;
close;

//Given data
B1=2.5*10^9;                                    // Mean optical power
B2=10*10^9;                                     // Split loss
L1=160*10^3;                                     // Total system margin



// Distance
L2=((B1/B2)^2*L1);
L2=L2*10^-3;



//Displaying results in the command window
printf("\n Distance = %0.0f Km ",L2);


// The answers vary due to round off error
```

Scilab code Exa 7.4 Compuatation of OSNR in a reference bandwidth and Q factor

```
// Example 7.4
// Compuatation of (a) OSNR in a reference bandwidth of 0.1 nm, (b) Q-factor.
```

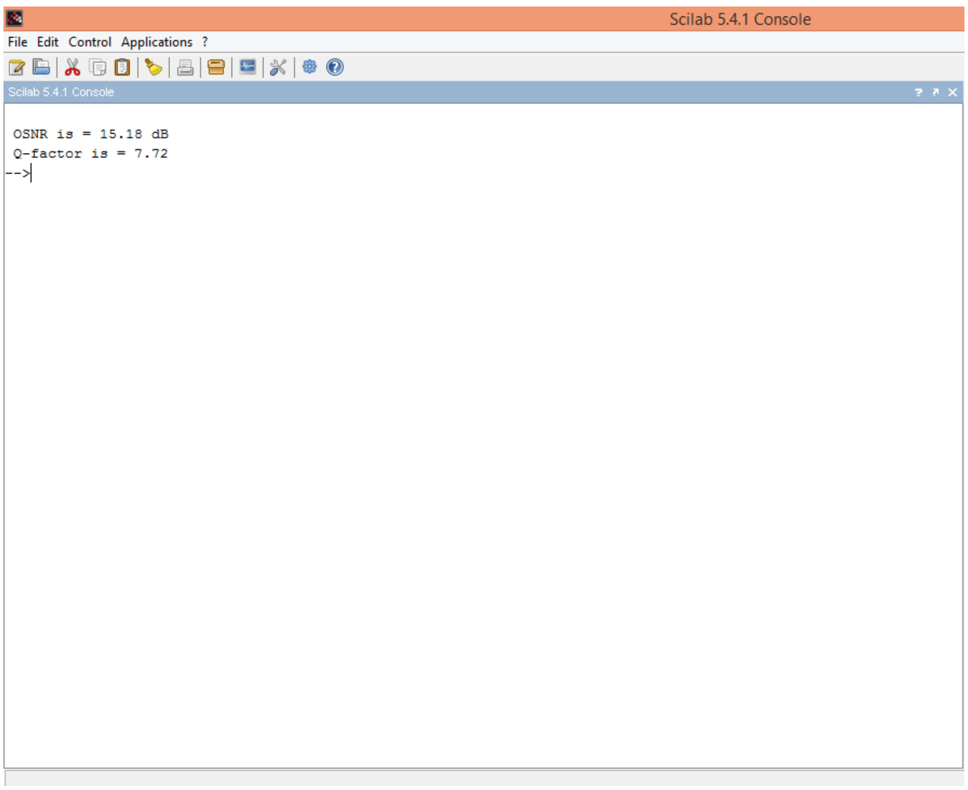

Figure 7.4: Compuatation of OSNR in a reference bandwidth and Q factor

```
// Page no. 321

clc;
clear;
close;

// Given data

f=10*10^9;
n=1.5;                                                  //Refractive index
h=6.63*10^-34;                                          // Planck constant
c=3*10^8;                                               // Velocity of light
lambda=1.55*10^-6;                                      //
q=1.6*10^-19;                                           // Electron charge
d=0.1*10^-9;                                            // Reference bandwidth
alpha=0.0461;                                           // Fiber loss coefficient
L=80;                                                   // Spacing
Pi=-3;                                                  // Mean fiber launch power
N=80;                                                   // Identical amplifers
```

```
fe=7*10^9;                                                      // Electrical filter bandwidth


// Signal calculation
df=-((c*d)/lambda^2);                                           //Reference frequency
G=exp(alpha*L);
G1=10*log10(G);
N1=10*log10(N);
Fn=2*n;                                                         //Noise figure
Fn=10*log10(Fn);

O=Pi-N1-G1-Fn+58;                                               //OSNR
Pi1=2*10^(-(3/10));                                             // Peak power in mW
f=c/lambda;
Q=sqrt((Pi1*10^-3)/(4*N*n*h*f*(G-1)*fe));                 //Q-factor

// Displaying the result in command window
printf("\n OSNR is = %0.2f dB",O);
printf("\n Q-factor is = %0.2f ",Q);

// The answer vary due to round off error
```

Scilab code Exa 7.5 Compuatation of the transmission distance

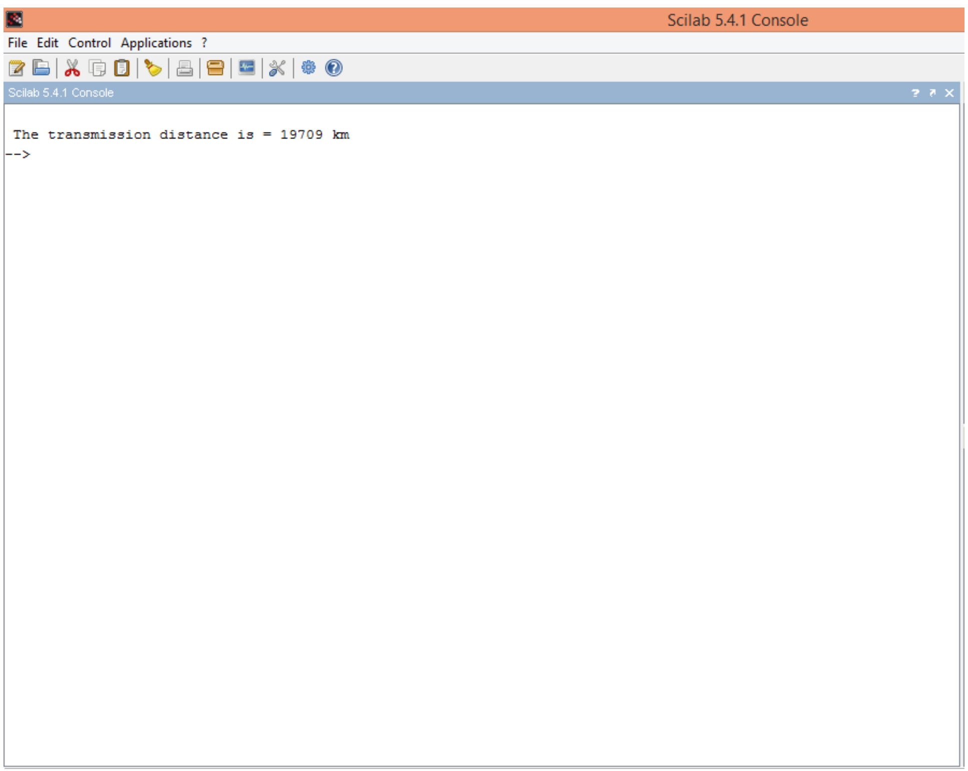

Figure 7.5: Compuatation of the transmission distance

```
// Example 2.1
// Compuatation of the transmission distance
//
// Page no. 325

clc;
clear;
close;

//Given data

fl=0.2                      // Fiber loss
L=100;                      // Amplifeir spacing
n=1.4;
h=6.63*10^-34;              // Planck constant
c=3*10^8;                   // Velocity of light
lambda=1.55*10^-6;

q=1.6*10^-19;              // Electron charge
R=0.9;
d=0.1*10^-9;
alpha=0.0461;
L=100;                    // Spacing
Pi=-3;                    // Mean fiber launch power
//N=80;                   // Identical amplifers
fe=7*10^9;                //  Electrical filter bandwidth
q=6;
B=5*10^9;


// The transmission distance
l=fl*L;
G=10^(l/10);
f=c/lambda;
// r=N*n*h*f*(G-1);
Pi=10^(-(2/10));
N=Pi/(q^2*n*h*f*(G-1)*B);
```

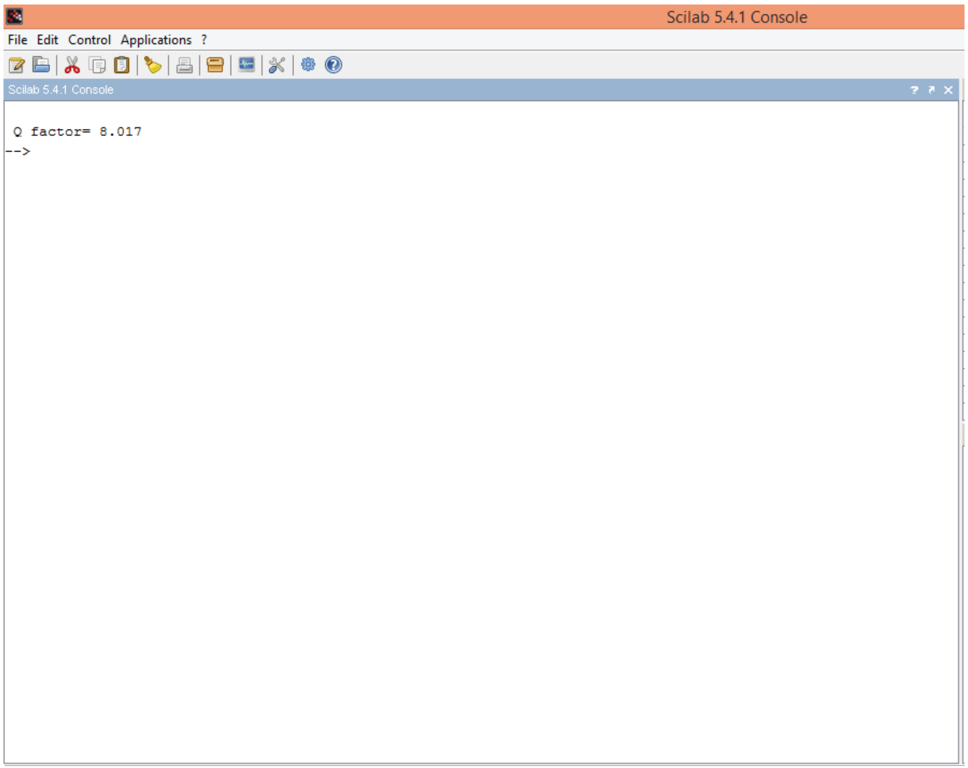

Figure 7.6: Compuatation of the Q factor

```
Td=N*L;
Td=Td*10^-3;

//Displaying the result in command window
printf("\n The transmission distance is = %0.0f km",
   Td);
```

Scilab code Exa 7.6 Compuatation of the Q factor

```
// Example 7.6
```

```
// Compuatation of the Q-factor.
//
// Page no. 327

clc;
clear;
close;

//Given data
alpha=0.18;            // Fiber loss coefficient
L=190;                 // Fiber length
G=20;                  // Gain of preamplifier
lambda=1.55*10^-6;     // Operating wavelength
h=6.63*10^-34;         // Planck constant
n=1.409;
G1=10^(G/10);
f0=20*10^9;
R=1.1;
q=1.6*10^-19;
fe=7.5*10^9;
Pi=1;                  // Input power
c=3*10^8;              // Velocity of light
k=1.38*10^-23;
T=298;
Rl=200;

// The Q factor
l=alpha*L;
Po=Pi-l+G;
Po=10^(Po/10)*10^-3;
f=c/lambda;
r=h*f*(G1-1)*n;
fn=2*n;
fn=10^(fn/10);
I1=R*Po+2*r*f0;
I0=2*R*r*f0;
o1=(2*q*I1*fe)+((4*k*T*fe)/Rl)+(2*R^2*r*(2*Po*fe+r
   *(2*f0-fe)*fe));
```

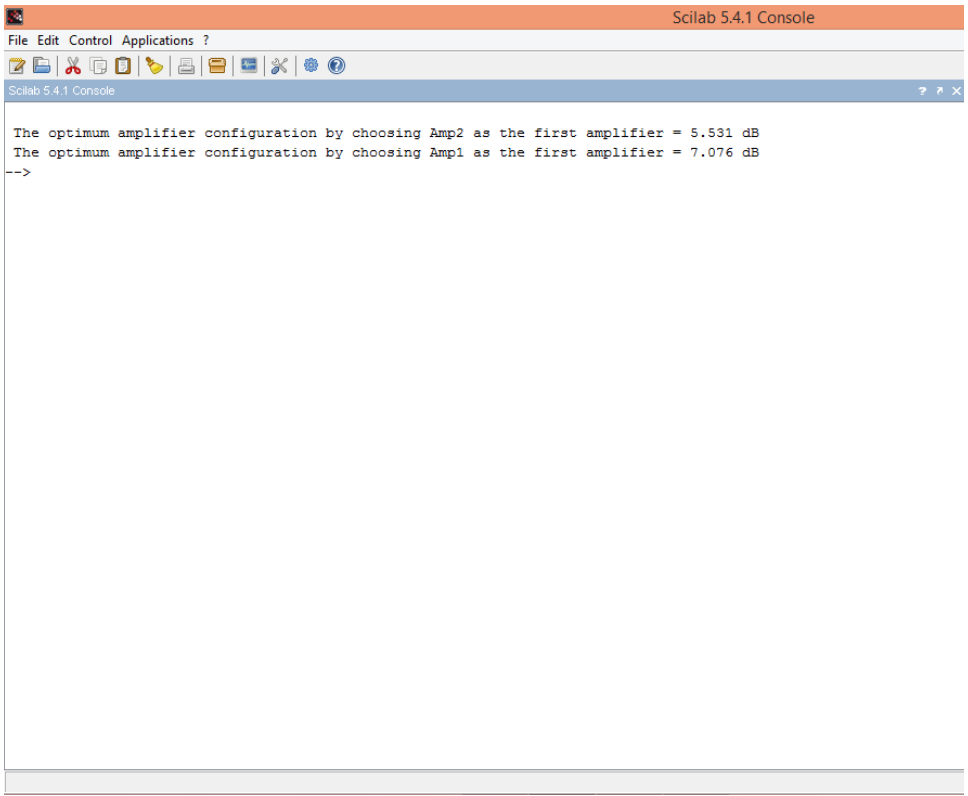

Figure 7.7: Compuatation of the optimum amplifier configuration

```
o2=(2*q*I0*fe)+((4*k*T*fe)/Rl)+(2*R^2*r^2*(2*f0-fe)*
    fe);
Q=(I1-I0)/(sqrt(o1)+sqrt(o2));

//Displaying the result in command window

printf("\n Q factor= %0.3f ",Q);

// The answer vary due to round off error
```

Scilab code Exa 7.7 Compuatation of the optimum amplifier configuration

```
// Example 7.7
// Compuatation of the optimum amplifier
   configuration
//
// Page no. 329

clc;
clear;
close;

//Given data

G1=8;            // Amplifier gain 1
G2=16;           // Amplifier gain 2
fn1=7;           // Noise figure of amplifier 1
fn2=5.5;         // Noise figure of amplifier 2
H=7;             // Insertion loss of the DCF
//N=80;          // Identical amplifers
fe=7*10^9;       //  Electrical filter bandwidth
// q=6;


// The optimum amplifier configuration

fn1=10^(fn1/10);
fn2=10^(fn2/10);
G2=10^(G2/10);
H=10^(H/10);
Fna=fn2+(fn1/(G2*H));
Fna=10*log10(Fna);
G=G2+G1+H;
Fnb=fn1+(fn2/(G1*H));

Fnb=10*log10(Fnb);

```

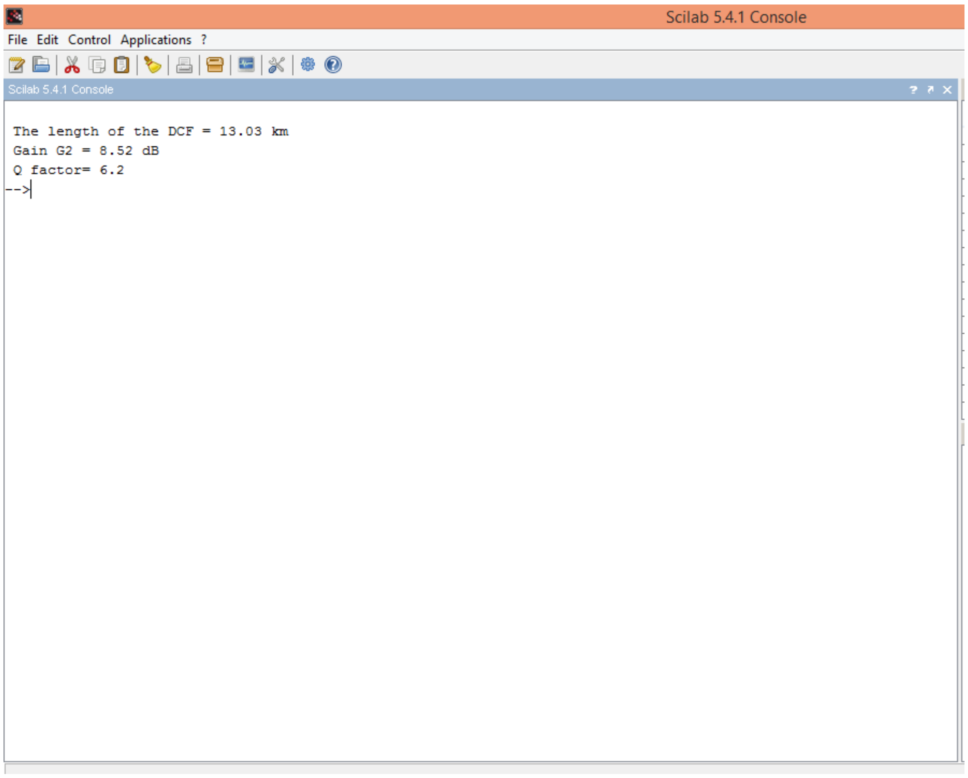

Figure 7.8: Compuatation of the length of the DCF and the gain G2 and the Q factor

```
//Displaying the result in command window
printf("\n The optimum amplifier configuration by choosing Amp2 as the first amplifier = %0.3f dB", Fna);
printf("\n The optimum amplifier configuration by choosing Amp1 as the first amplifier = %0.3f dB", Fnb);

// The answer vary due to round off error
```

Scilab code Exa 7.9 Compuatation of the length of the DCF and the gain G2 and the Q factor

```
// Example 7.9
// Compuatation of the (a) the length of the DCF (b) the gain G2 and (c) the Q-factor.
//
// Page no. 331

clc;
clear;
close;

//Given data
b=-21*10^-27;
L=100*10^3;
Lt=100;
l=0.18;                    // Loss
l1=0.5;                    // Dispersion coefficients of the TF
G1=16;                     // Amplifier gain
p=-2;                      // Mean transmitter output power
fe=7*10^9;
c=3*10^8;                  // Velocity of light
h=6.62*10^-34;            // Planck constant
fn1=5.5;                  // Noise figure of amplifier 1
fn2=7.5;                  // Noise figure of amplifier 2
lambda=1.55*10^-6;
bd=145*10^-27;           // Dispersion coefficients of the DCF

// (a) The length of the DCF
st=b*L;
```

```
sd=-0.9*st;
Ld=sd/bd;
Ld=Ld*10^-3;
// (b) Gain G2
Ht=l*Lt;
Hd=l1*Ld;
G2=Ht+Hd-G1;

// (c) Q factor
Ge=G1+G2+-Hd;
Ge=10^(Ge/10);
fn1=10^(fn1/10);
fn2=10^(fn2/10);
G1=10^(G1/10);
Hd=10^(-Hd/10);
Fe=fn1+(fn2/(G1*Hd))-(1/G1);
f=c/lambda;
r=70*h*f*(((Ge*Fe)-1)/2);
Pi=2*10^(p/10)*10^-3;
Q=sqrt(Pi/(4*r*fe));


//Displaying the result in command window
printf("\n The length of the DCF = %0.2f km",Ld);
printf("\n Gain G2 = %0.2f dB",G2);
printf("\n Q factor= %0.1f ",Q);

// The answer vary due to round off error
```

Chapter 8

Performance Analysis

Scilab code Exa 8.2 Compuatation of error probability if the receiver is a balanced homodyne or a balanced heterodyne

```
// Example 8.1
// Compuatation of error probability if the receiver
    is (a) a balanced homodyne or (b) a balanced
   heterodyne
// Page no. 354

clc;
clear;
close;

// Given data
Po=5;                       // Lunch peak power
fl=50;                      // Fiber loss
G=30;                       // Preamplifier Gain
f=10*10^9;
n=1.5;
h=6.63*10^-34;              // Planck constant
c=3*10^8;                   // Velocity of light
```

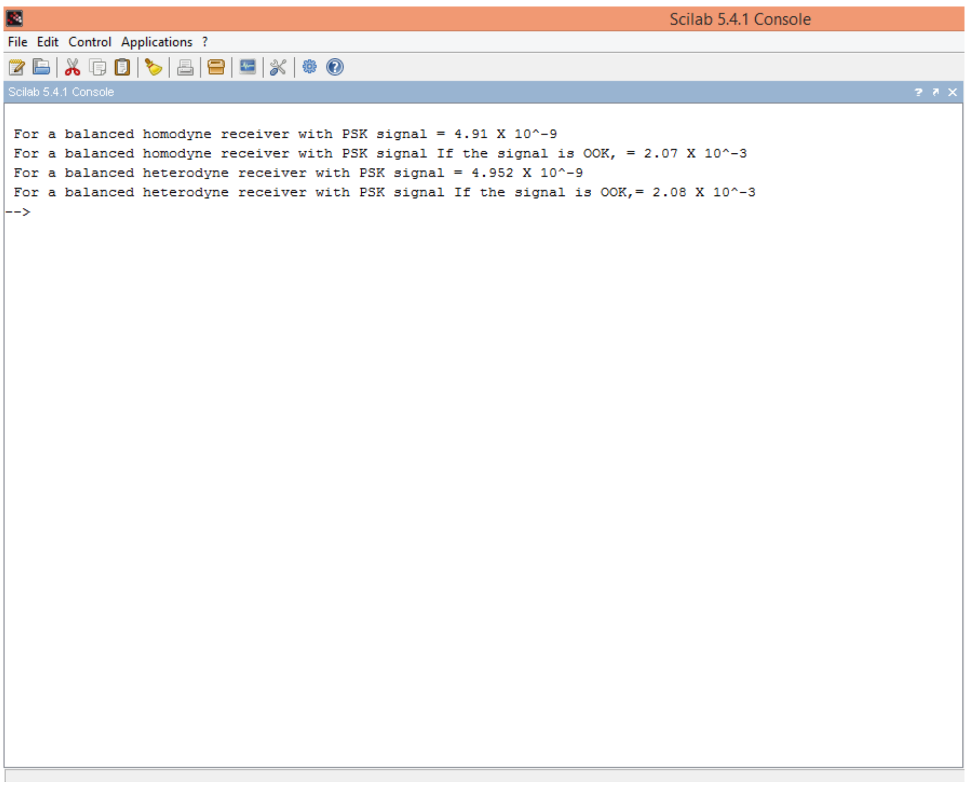

Figure 8.1: Compuatation of error probability if the receiver is a balanced homodyne or a balanced heterodyne

```
lambda=1550*10^-9;
q=1.6*10^-19;              // Electron charge
R=0.9;

// Signal calculation
Pr=Po-fl+G;
Pr=10^(Pr/10)*10^-3;

Tb=1/(f);
E=Pr*Tb;
f1=c/lambda;
G=10^(G/10);
r=n*h*f1*(G-1);
//rs=q*I;
N=r+(q/(2*R));
Nh=r/2+(q/(2*R));

// Error probability
// (a) For a balanced homodyne receiver with PSK signal
Ps=1/2*erfc(sqrt(E/N));
E1=E/2;
// If the signal is OOK
Pso=1/2*erfc(sqrt(E1/(2*N)));

//(b) For a balanced heterodyne receiver with PSK signal
Pb=1/2*erfc(sqrt(E/(2*Nh)));
//E1=E/2;
// If the signal is OOK
Pbo=1/2*erfc(sqrt(E1/(4*Nh)));

//Displaying the result in command window
printf("\n For a balanced homodyne receiver with PSK signal = %0.2f X 10^-9 ",Ps*10^9);

printf("\n For a balanced homodyne receiver with PSK signal If the signal is OOK, = %0.2f X 10^-3",
```

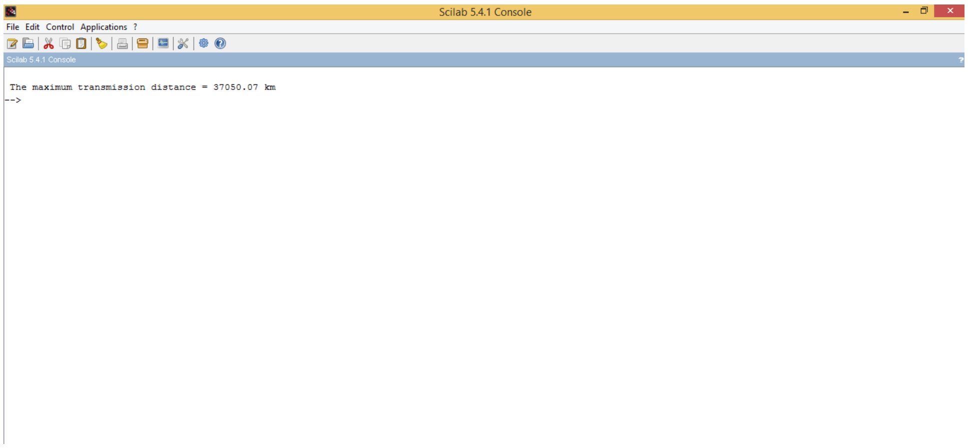

Figure 8.2: Calculation of the maximum transmission distance

```
    Pso*10^3);
printf("\n For a balanced heterodyne receiver with
    PSK signal = %0.3f X 10^-9",Pb*10^9);
printf("\n For a balanced heterodyne receiver with
    PSK signal If the signal is OOK,= %0.2f X 10^-3",
    Pbo*10^3);

// The answer vary due to round off error
```

Scilab code Exa 8.3 Calculation of the maximum transmission distance

```
// Example 8.3
// Calculation of the maximum transmission distance.
// Page no 394

clc;
clear;
close;

```

```
//Given data
p=3;                                    // Peak power
tb=40*10^9;                             // Bit rate
c=3*10^8;                               // Velocity of light
lambda=1550*10^-9;                      // Operating frequency
l=0.2;                                  // Loss
d=80;                                   // Distance
G=16                                    // Gain
h=6.626*10^-34                          // Planck constant
n=1;
pb=10^-5;                               // Error probability
l1=80*10^3;                             // N spans


// The maximum transmission distance

p=p+10*log10(1/2);
p=10^(p/10)*10^-3;
t=1/(tb);
E=p*t;
f=c/lambda;
fl=l*d;
G=10^(G/10);
r=n*h*f*(G-1);   // Calculation is wrong in book.
//pb=1/2*(exp(-(E/r)));
N=-(E/(log(2*pb)*r));

L=N*l1;

// Displaying results in the command window
printf("\n The maximum transmission distance = %0.2f km",L*10^-3);

// In the book PSD per amplifier calcualation is
```

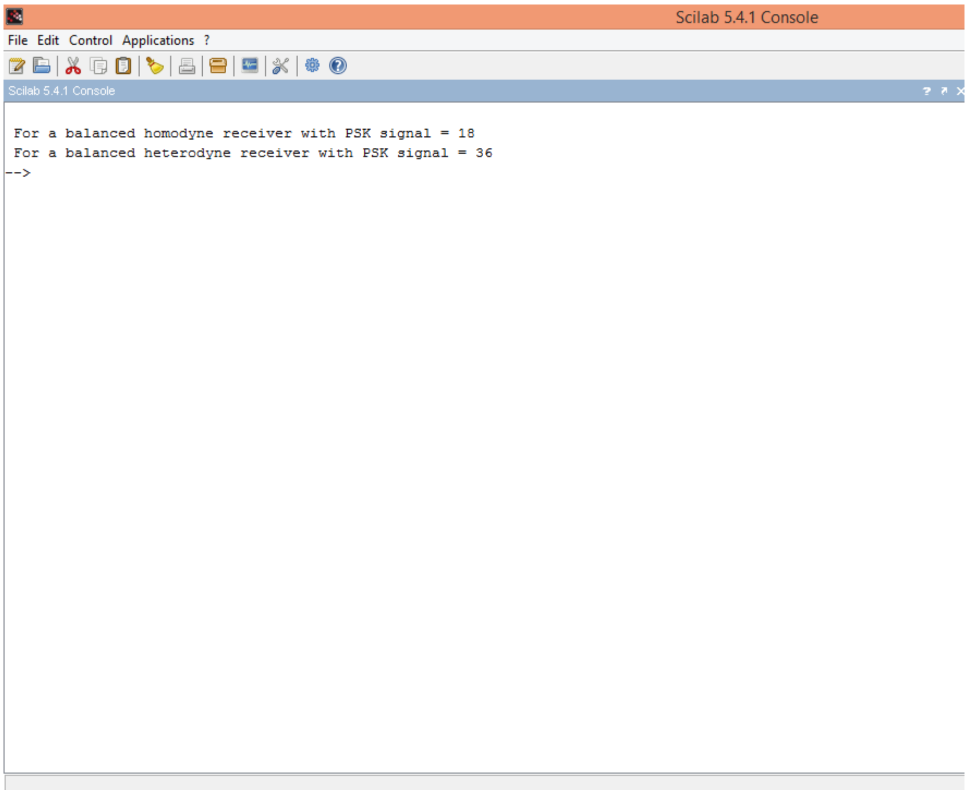

Figure 8.3: o find the mean number of signal photons required in a shot noise limited coherent communication system

```
    wrong, therefore final answer is wrong.
```

Scilab code Exa 8.6 o find the mean number of signal photons required in a shot noise limited coherent communication system

```
// Example 8.6
// To find the mean number of signal photons
   required in a shot noise-limited coherent
   communication system based on OOK for the
```

```
following cases: (i) balanced homodyne receiver; (ii)balanced heterodyne receiver (a) a balanced homodyne or (b) a balanced heterodyne
// Page no. 384

clc;
clear;
close;

// Given data
Pb=1*10^-9;                    //Error probability
neta=1;                        //quantum efficiency

//a)for balanced homodyne receiver
Ns=(erfinv(1-(2*neta*Pb)))^2;

//(b)for balanced heterodyne receiver
Ns1=(erfinv(1-(2*neta*Pb))*sqrt(2))^2;

//Displaying the result in command window
printf("\n For a balanced homodyne receiver with PSK signal = %0.0f  ",Ns);
printf("\n For a balanced heterodyne receiver with PSK signal = %0.0f  ",Ns1);
```

Chapter 9

Channel Multiplexing Techniques

Scilab code Exa 9.1 Calculation of the channel spacing and the signal bandwidth in a channel and cther total bandwidth of the WDM signal and the total data rate

```
// Example 9.1
// Calculation of the (a) the channel spacing, (b)
   the signal bandwidth in a channel and cther total
    bandwidth of the WDM signal, and (c) the total
   data rate.
// Page no 392

clc;
clear;
close;

//Given data
Bs=10*10^12;                  // Symbol rate
n=6;                         // Spectral efficiency
Fs=10*10^12;                 // Symbol rate
```

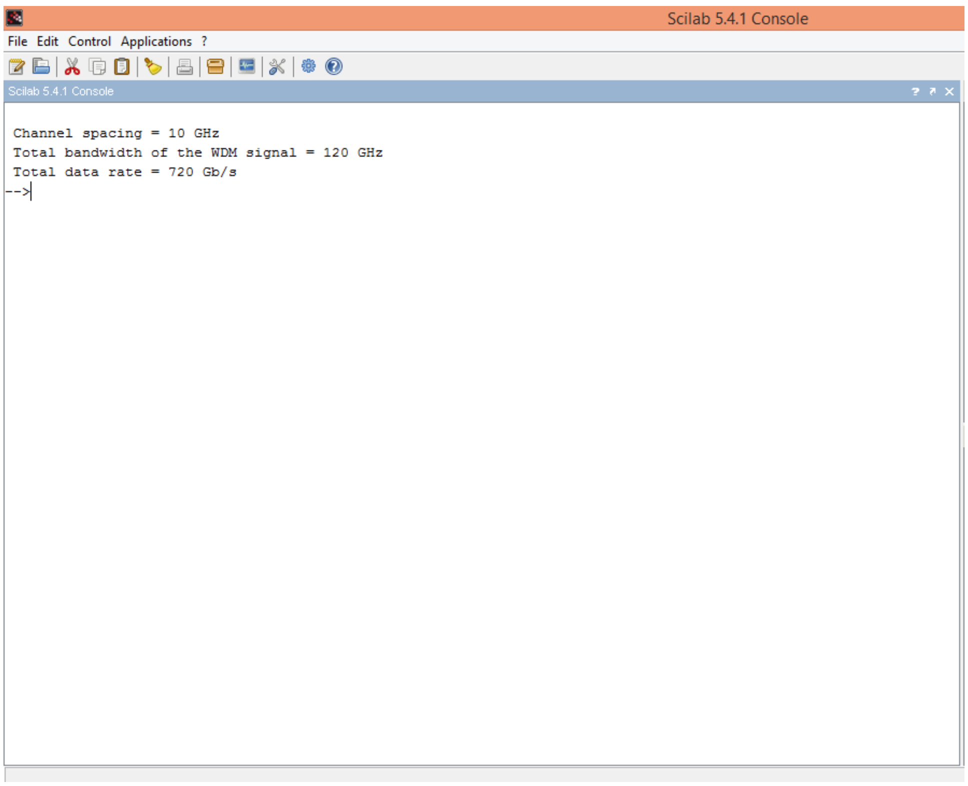

Figure 9.1: Calculation of the channel spacing and the signal bandwidth in a channel and cther total bandwidth of the WDM signal and the total data rate

```
N=12;                          // No of channels



// (a) Channel spacing
B=Bs*log2(64);
f=B/n;

// (b) Total bandwidth of the WDM signal
T1=(N-1)*f+(2*Fs)/2;
T1=T1*10^-12;
// (c) Total data rate
T2=N*B;
T2=T2*10^-12;

// Displaying results in the command window
printf("\n Channel spacing = %0.0f GHz ",f*10^-12);

printf("\n Total bandwidth of the WDM signal = %0.0f GHz ",T1);
printf("\n Total data rate = %0.0f Gb/s ",T2);
```

Scilab code Exa 9.2 Calculation of the total power at the fiber output

```
// Example 9.2
// Calculation of the total power at the fiber output.
// Page no 393

clc;
clear;
close;

```

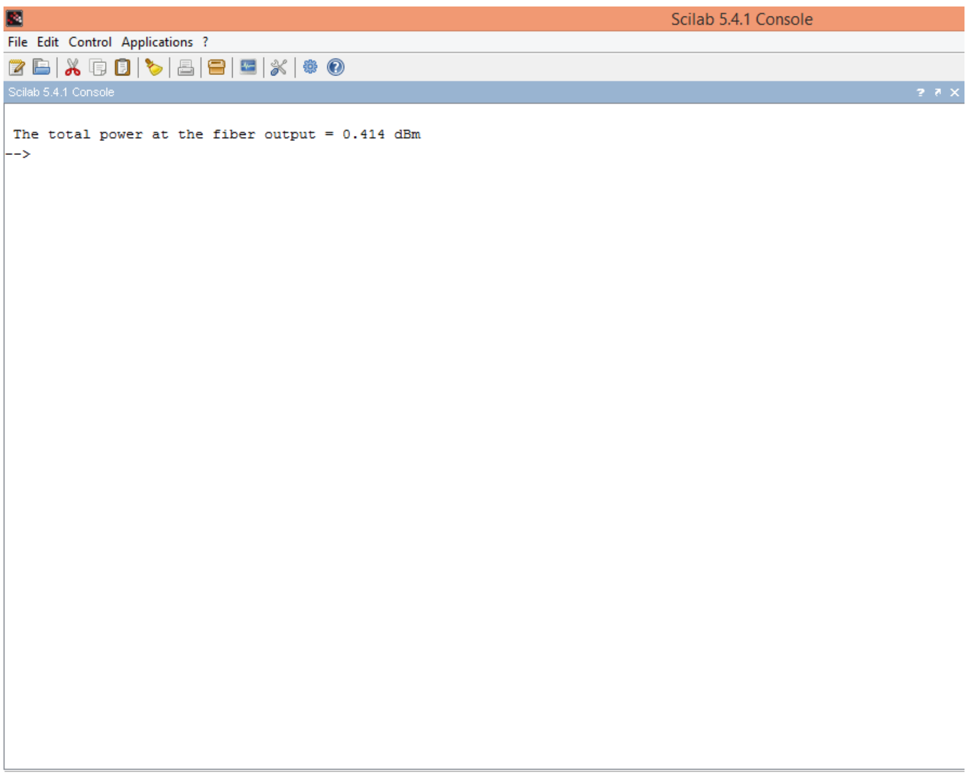

Figure 9.2: Calculation of the total power at the fiber output

```
//Given data

p=0;                    // Power per channel
fl=0.2;                 // Fiber loss
f=50;                   // Wavelength


// The total power at the fiber output.
pc=10^(0.1*p);
tp=pc*11;
tp1=10*log10(tp);
tfl=fl*f;
to=tp1-tfl;




//Displaying results in the command window
printf("\n The total power at the fiber output = %0
    .3f dBm ",to);
```

Scilab code Exa 9.3 Calculation of The lengths of the adjacent waveguides and phase shift phi1 and phi2

```
// Example 9.3
// Calculation of a) The lengths of the adjacent
    waveguides and b) phase shift phi1 and phi2.
// Page no 400

clc;
clear;
close;

```

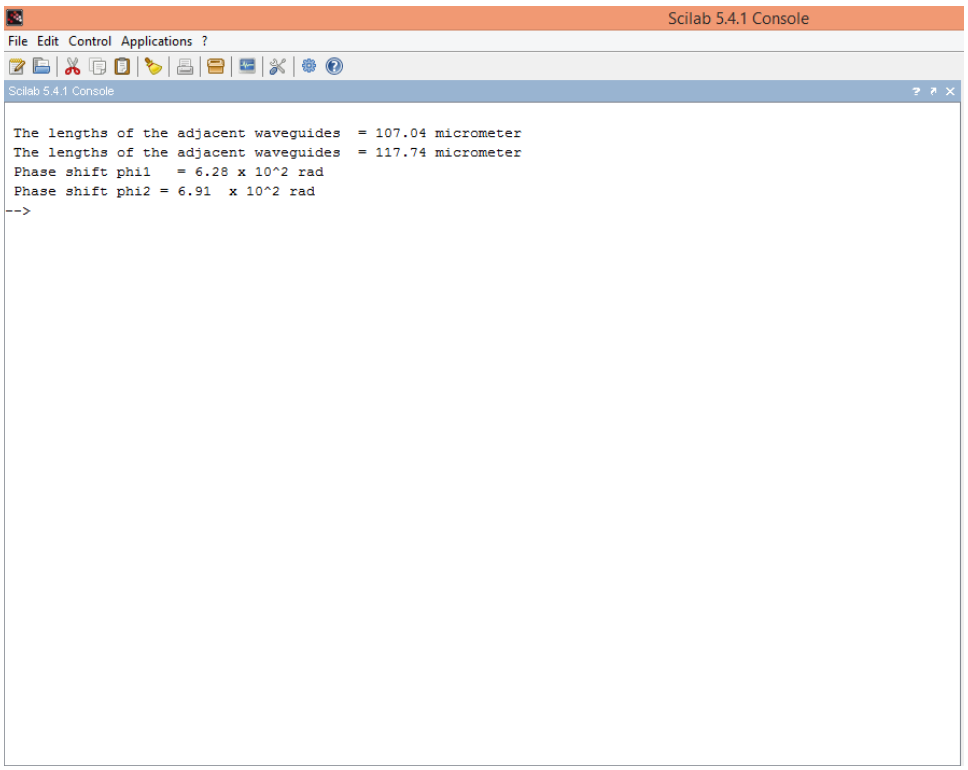

Figure 9.3: Calculation of The lengths of the adjacent waveguides and phase shift phi1 and phi2

```
//Given data

p=0;                        // Power per channel
fl=0.2;                     // Fiber loss
m1=100;                     // Wavelength
m2=110;
lambda1=1550*10^-9;
lambda2=1550.8*10^-9;
c=3*10^8;                   // Velocity of light
b0=5.87*10^6;
b1=4.86*10^-9;

// a) The lengths of the adjacent waveguides
l1=(2*%pi*m1)/b0;
l2=(2*%pi*m2)/b0;


// b) Phase shift phi1 and phi2.
dfdl=-(c/lambda1^2);
dbdl=2*%pi*b1*dfdl;
phi1=2*%pi*m1+(lambda2-lambda1)*l1*dbdl;
phi2=2*%pi*m2+(lambda2-lambda1)*l2*dbdl;

//Displaying results in the command window
printf("\n The lengths of the adjacent waveguides = %0.2f micrometer ",l1*10^6);
printf("\n The lengths of the adjacent waveguides = %0.2f micrometer",l2*10^6);
printf("\n Phase shift phi1  = %0.2f x 10^2 rad ",phi1*10^-2);
printf("\n Phase shift phi2 = %0.2f  x 10^2 rad",phi2*10^-2);

// The answers vary due to round off error
```

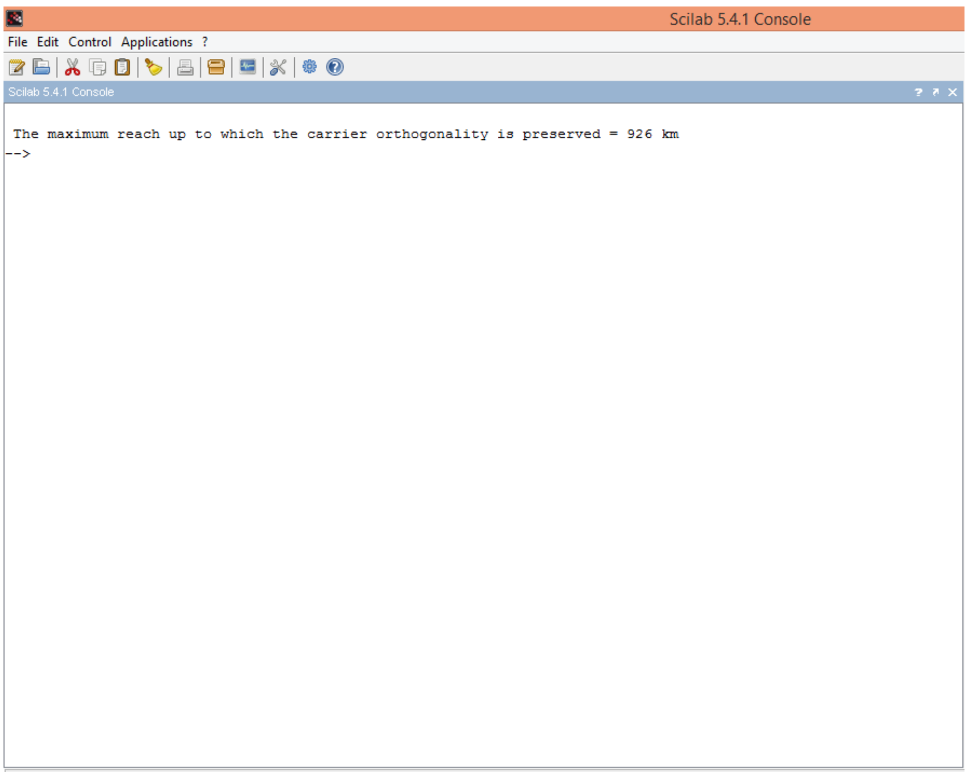

Figure 9.4: Calculation of the maximum reach up to which the carrier orthogonality is preserved

Scilab code Exa 9.4 Calculation of the maximum reach up to which the carrier orthogonality is preserved

```
// Example 9.4
// Calculation of the maximum reach up to which the carrier orthogonality is preserved.
// Page no 408

clc;
clear;
close;

//Given data
b=22*10^-27;                    // Power launched in port 1
T=1.28*10^-9;                   // Guard interval
N=128;                          //  Subcarriers
f=78.125*10^6;                  // Frequency spacing between subcarriers

// Bit rate of communication system
I=T/(b*2*%pi*N*f);
I=I*10^-3;



//Displaying results in the command window
printf("\n The maximum reach up to which the carrier orthogonality is preserved = %0.0f km ",I);

// The answers vary due to round off error
```

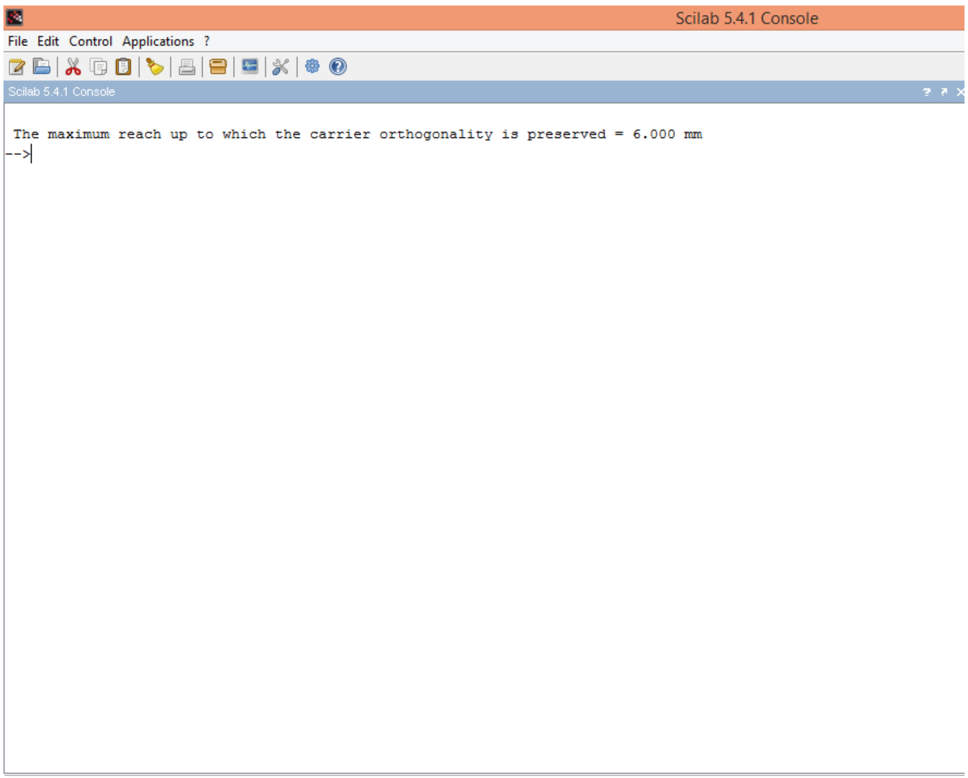

Figure 9.5: Calculation of the maximum reach up to which the carrier orthogonality is preserved

Scilab code Exa 9.5 Calculation of the maximum reach up to which the carrier orthogonality is preserved

```
// Example 9.5
// Calculation of the maximum reach up to which the
   carrier orthogonality is preserved.
// Page no 410

clc;
clear;
close;

//Given data
d=30*10^-12;            // Delay
b=0.5*10^-8;

// The maximum reach up to which the carrier
   orthogonality is preserved
L=d/b;
L=L*10^3;



//Displaying results in the command window
printf("\n The maximum reach up to which the carrier
    orthogonality is preserved = %0.3f mm ",L);

// The answers vary due to round off error
```

Scilab code Exa 9.6 Calculation of the ODTM to multiplex data

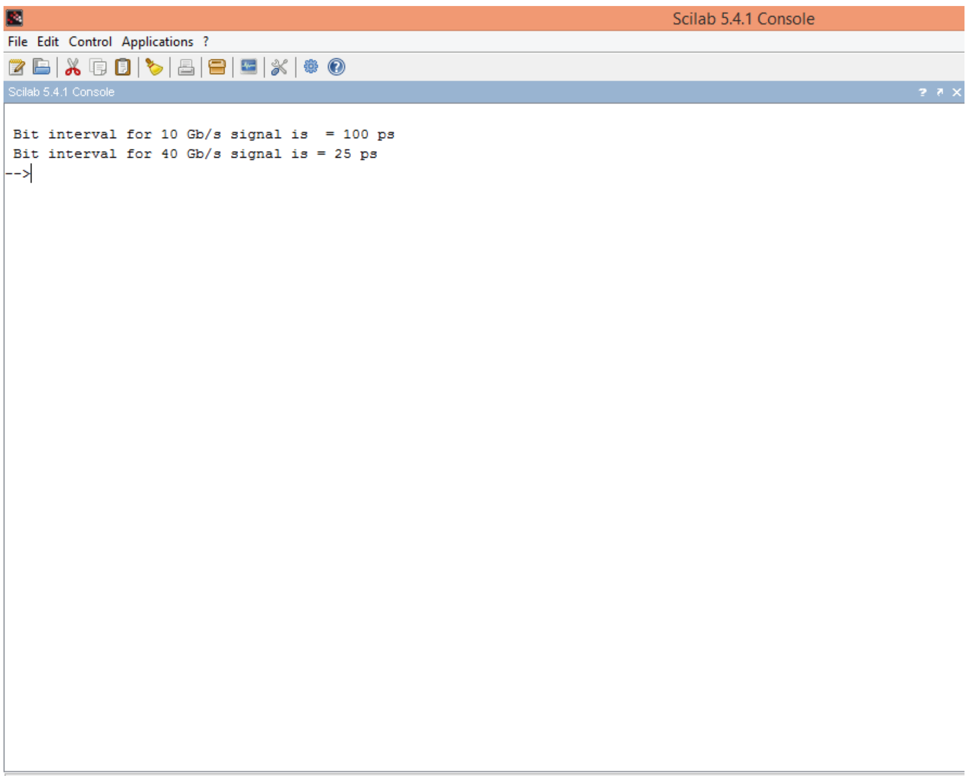

Figure 9.6: Calculation of the ODTM to multiplex data

```
// Example 9.6
// Calculation of the ODTM to multiplex data.
// Page no 411

clc;
clear;
close;

//Given data
f1=10*10^9;
f2=40*10^9;


// The ODTM to multiplex data
b1=1/(f1);
b2=1/(f2);
b1=b1*10^12;
b2=b2*10^12;

//Displaying results in the command window
printf("\n Bit interval for 10 Gb/s signal is  = %0
    .0f ps ",b1);
printf("\n Bit interval for 40 Gb/s signal is = %0.0
    f ps ",b2);

// The answers vary due to round off error
```

Scilab code Exa 9.7 Calculation of the the total data rate and the spectral efficiency

```
// Example 9.7
// Calculation of the (a) the total data rate and (b
    ) the spectral efficiency.
```

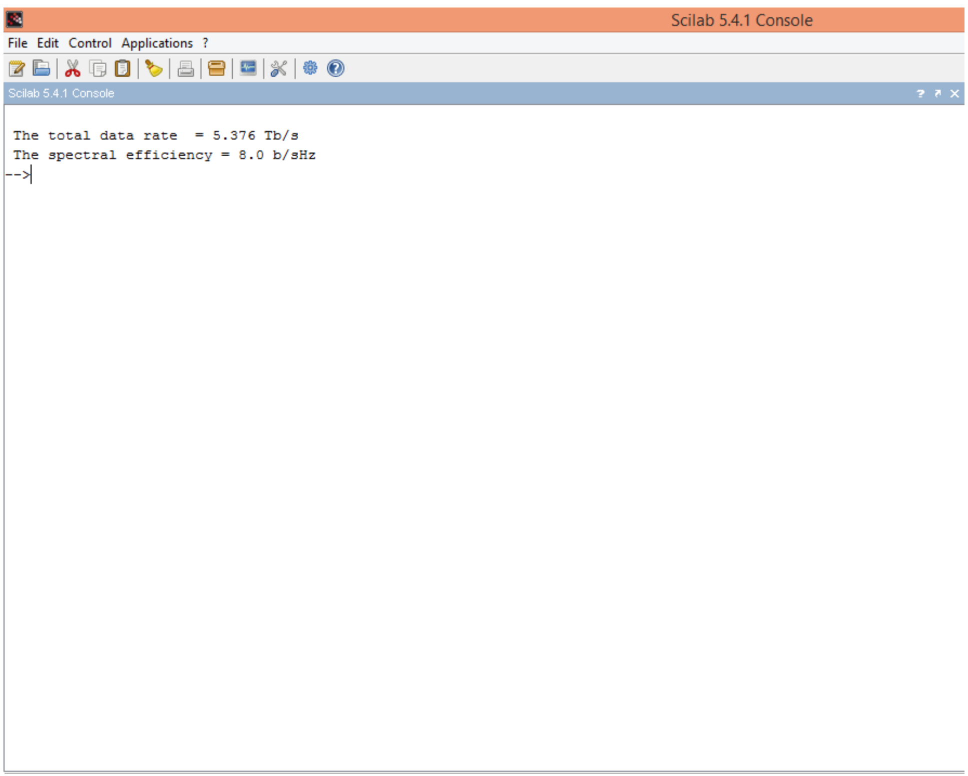

Figure 9.7: Calculation of the the total data rate and the spectral efficiency

```
// Page no 413

clc;
clear;
close;

//Given data
M=16;
np=2;                  // No of polarization
nc=24;                 // No of channels
bs=28*10^9;          // Symbol rate per polarization

// (a) The total data rate
B=bs*log2(M);
T=B*np*nc;


// (b) The spectral efficiency
N=bs*nc;
s=T/N;

//Displaying results in the command window
printf("\n The total data rate  = %0.3f Tb/s ",T*10^-12);

printf("\n The spectral efficiency = %0.1f b/s/Hz ",s);
```

Scilab code Exa 9.8 Calculation of the number of subcarriers required to transmit information

```
// Example 9.8
// Calculation of the number of subcarriers required
```

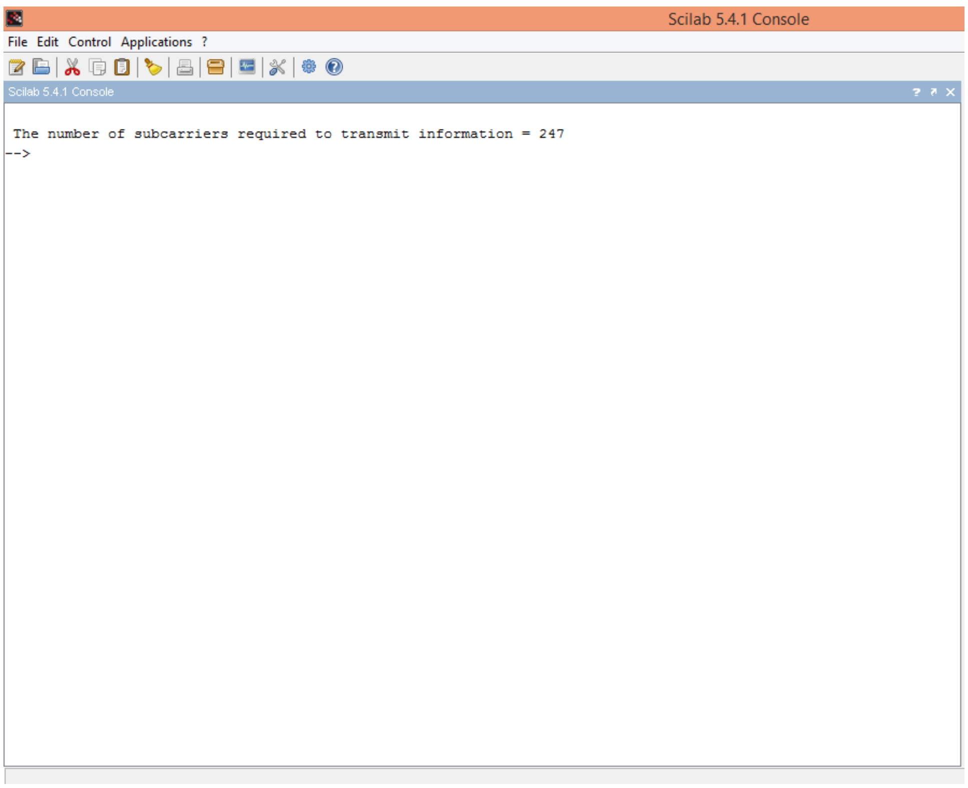

Figure 9.8: Calculation of the number of subcarriers required to transmit information

```
    to transmit information.
// Page no 413

clc;
clear;
close;

//Given data
M=4;
np=2;                   // No of polarization
nc=24;                 // No of channels
bs=10*10^9;          // Symbol rate per polarization
d=5000*10^3;         // Transmission distance
b=22*10^-27;
ts= 49.3*10^-9;

// The total data rate
B=bs*log2(M);
T=d*b*%pi*bs;
//L=T/(b*2*%pi*N*bs);
N=(bs*ts)/2;


//Displaying results in the command window
printf("\n The number of subcarriers required to
   transmit information = %0.0f ",N);

// The answers vary due to round off error
```

Scilab code Exa 9.9 Calculation of the the signal power subcarrier polarization at the fiber output and the data rate and the spectral efficiency

```
// Example 9.9
```

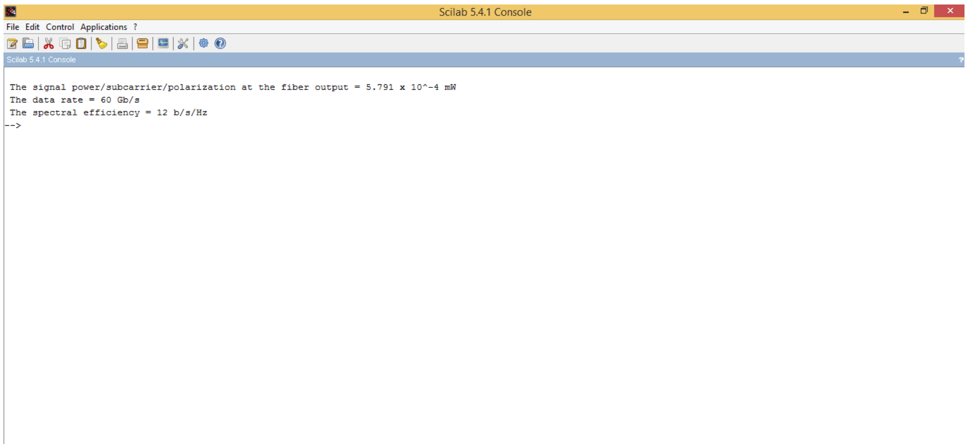

Figure 9.9: Calculation of the the signal power subcarrier polarization at the fiber output and the data rate and the spectral efficiency

```
// Calculation of the (a) the signal power/
   subcarrier/polarization at the fiber output, (b)
   the data rate and (c) the spectral efficiency
// Page no 414

clc;
clear;
close;

//Given data
fl=0.19;               // Fiber loss
fg=70;                 // Fiber length
nc=24;                // No of channels
ip=2;
bs=10*10^9;           // Symbol rate per polarization
ts= 12.8*10^-9;       // Symbol period
n=64;                 // No of subcarriers
np=2;                 // Launch power to the fiber


// (a) The signal power/subcarrier/polarization at
   the fiber output
```

```
T=fl*fg;
p=ip-T;
p1=10^(p/10);
s=p1/(np*n);
//s=s*10^4;

// (b) The data rate
bs=1/ts;
B=log2(n)*bs;
bt=B*2*n;

// (c) the spectral efficiency
Tb=n*bs;
se=bt/Tb;




//Displaying results in the command window
printf("\n The signal power/subcarrier/polarization at the fiber output = %0.3f x 10^-4 mW ",s*10^4);

printf("\n The data rate = %0.0f Gb/s ",bt*10^-9);

printf("\n The spectral efficiency = %0.0f b/s/Hz ",se);

// The answers vary due to round off error
```

Chapter 10

Nonlinear Effects in Fibers

Scilab code Exa 10.1 Calculation of the non linear coeffifient

```
// Example 10.1
// Calculation of the non linear coeffifient.
// Page no 429

clc;
clear;
close;

//Given data
n2=2.5*10^-20;                          // Kerr coefficient
lambda=1550*10^-9;                      // Wavelength
A=80*10^-12;                            // Effective area



// Non linear coeffifient
g=(n2*2*%pi)/(lambda*A);
g=g*10^3;
```

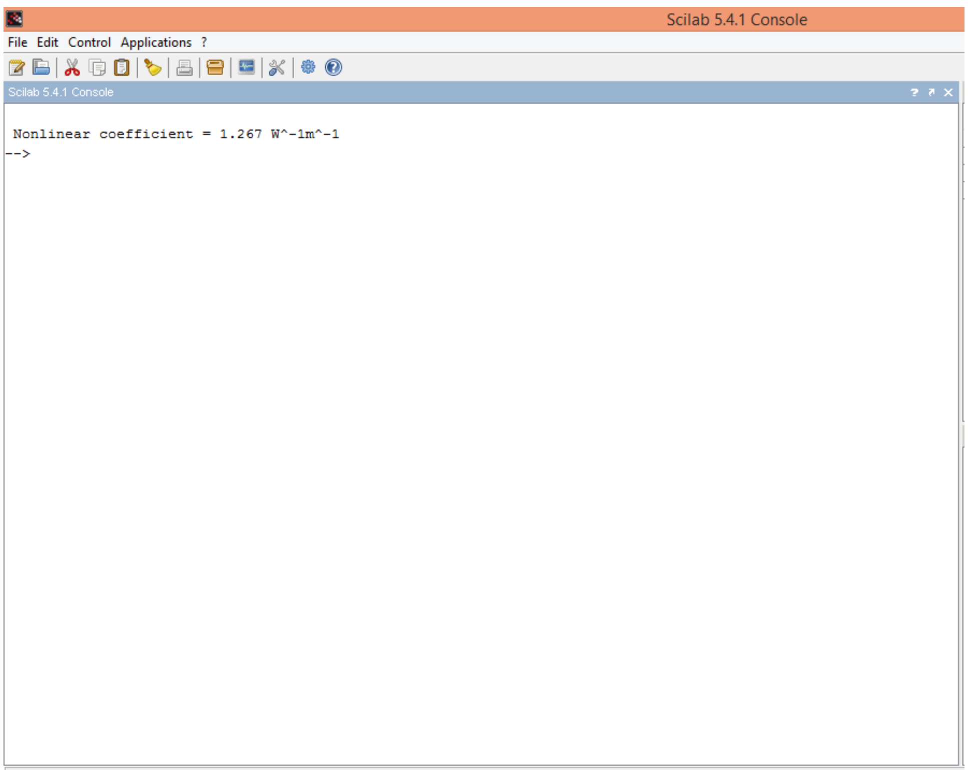

Figure 10.1: Calculation of the non linear coeffifient

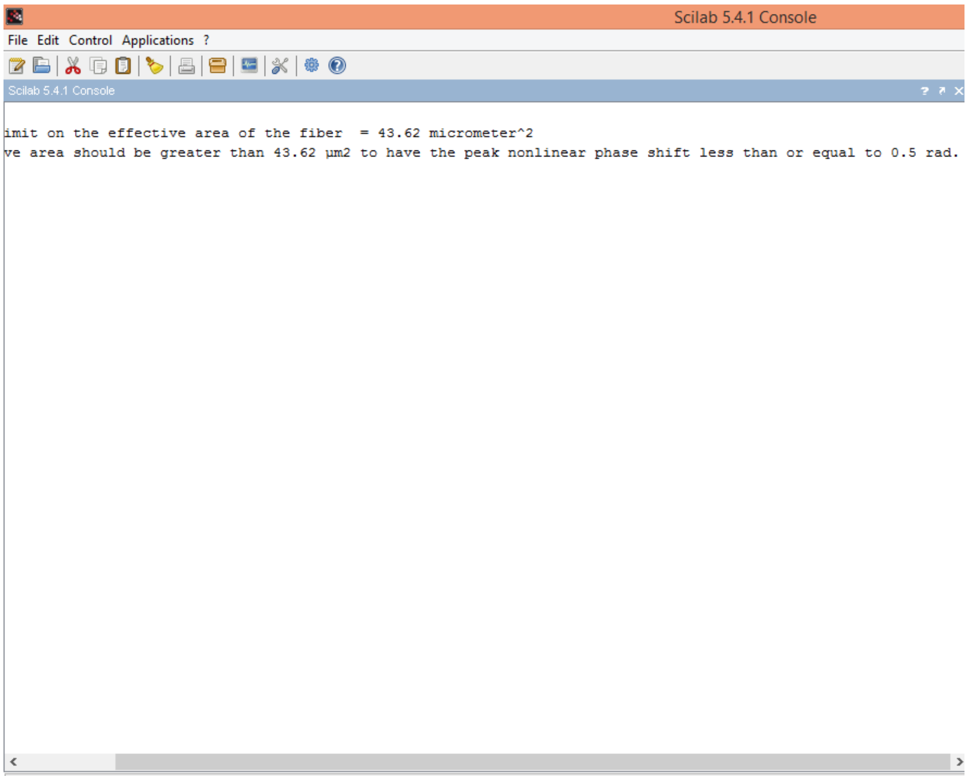

Figure 10.2: Calculation of the lower limit on the effective area of the fiber

```


//Displaying results in the command window
printf("\n Nonlinear coefficient = %0.3f W^-1m^-1 ",
   g);
```

Scilab code Exa 10.2 Calculation of the lower limit on the effective area of the fiber

```
// Example 10.2
```

```
// Calculation of the lower limit on the effective
   area of the fiber.
// Page no 431

clc;
clear;
close;

//Given data

c=3*10^8;                   // Velocity of light
tl=1000*10^3;               // Total length
as=100*10^3;                // Amplifier spacing
alpha=0.046*10^-3;          // Loss coefficient
L=100*10^3;
n2=2.5*10^-20;             // Kerr coefficient
p=0;                       // Peak power at the fiber
   input
lambda=1550*10^-9;         // Operating frequency

// The peak power required to form a soliton
Le=(1-exp(-alpha*L))/alpha;
n=tl/as;
p=10^(p/10);
r=0.5/(Le*p);
A=(2*%pi*n2)/(lambda*r);
A=A*10^12;

// Displaying results in the command window
printf("\n The lower limit on the effective area of
   the fiber  = %0.2f micrometer^2",A*10^-2);
printf("\n The effective area should be greater than
    43.62  m2 to have the peak nonlinear phase
   shift less than or equal to 0.5 rad.");


// The answers vary due to round off error
```

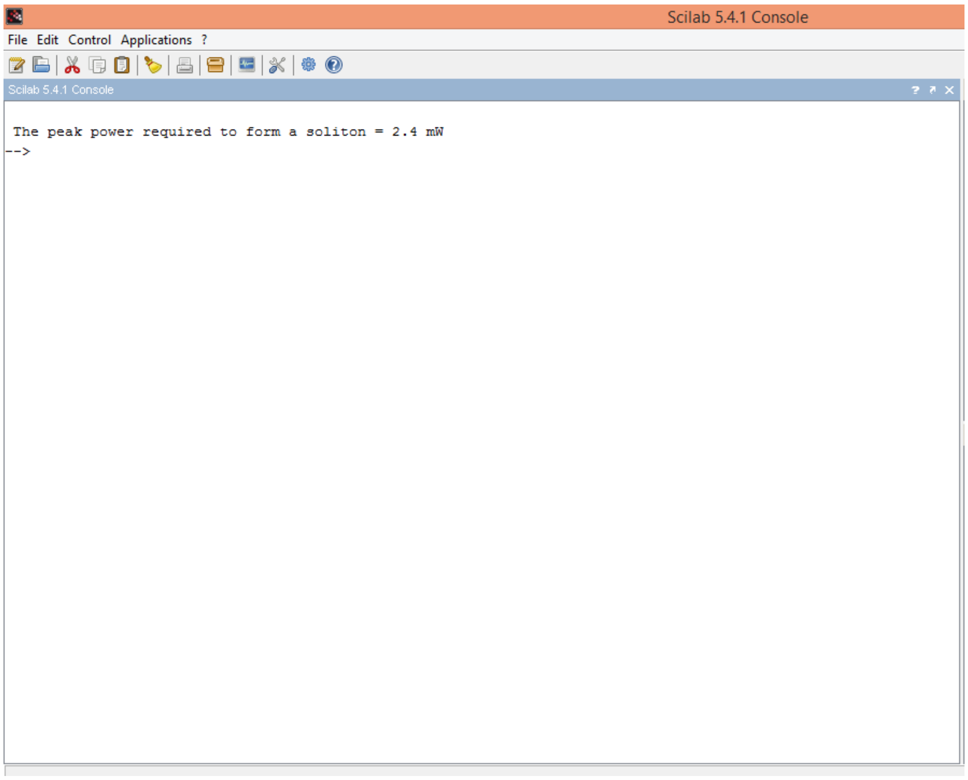

Figure 10.3: Calculation of the peak power required to form a soliton

Scilab code Exa 10.3 Calculation of the peak power required to form a soliton

```
// Example 10.3
// Calculation of the peak power required to form a
    soliton
// Page no 435

clc;
```

```
clear;
close;

//Given data

b=-21*10^-27;          // FWHM of a fundamental soliton
Tf=50*10^-12;          // Fiber dispersion coefficient
r=1.1*10^-3;           // Nonlinear coefficient

// The peak power required to form a soliton
Th=asech(sqrt(0.5));
f=2*Th;
T0=Tf/f;
n=(sqrt(-b))/T0;
P=(n^2)/r;
//P=P*10^2;


// Displaying results in the command window
printf("\n The peak power required to form a soliton
    = %0.1f mW",P*10^2);

// Answer is wrong in book
```

Scilab code Exa 10.4 Calculation of the peak power required to form a soliton

```
// Example 10.4
// Calculation of the peak power required to form a
   soliton
// Page no 444

clc;
```

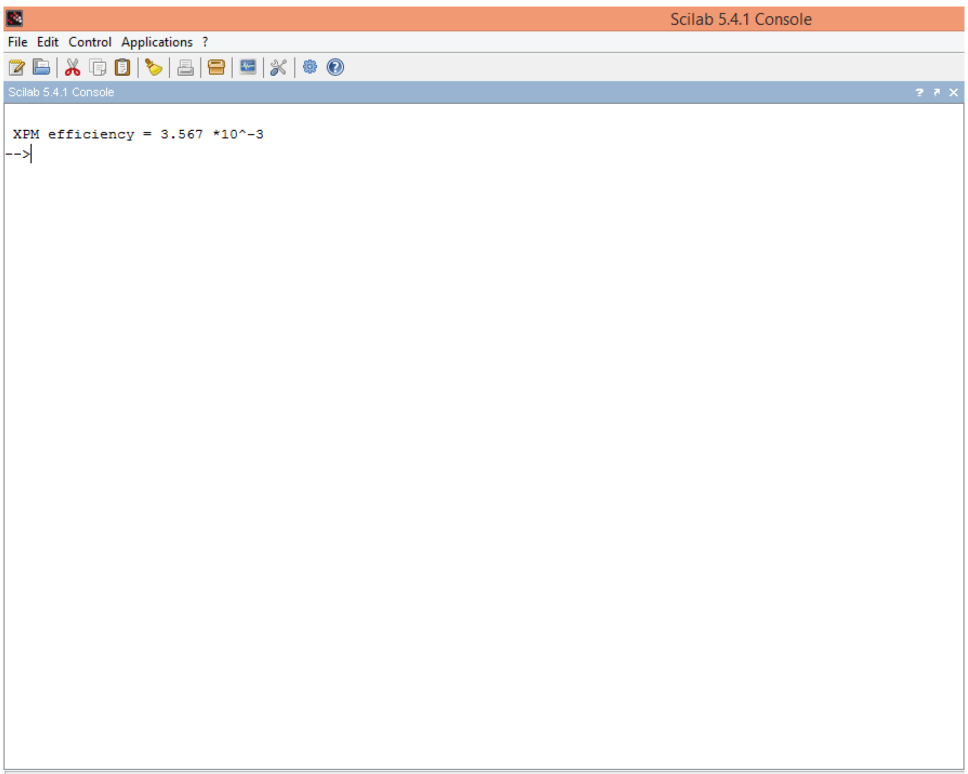

Figure 10.4: Calculation of the peak power required to form a soliton

```
clear;
close;

// Given data

c=3*10^8;                    // Velocity of light
S=0.06*10^3;                // Dispersion slope
D=17*10^-6;                 // Dispersion coefficient
lambda=1550*10^-9;         // Signal Wavelength
lc=1550*10^-9;             // Signal Wavelength
lp=1549.6*10^-9;           // Pump wavelength
l=50*10^3;                 // Length
r=2*%pi*10^10;
alpha=0.046*10^-3;          // Loss coefficient

// The peak power required to form a soliton
b3=S*(lambda^2/(2*%pi*c))+D*(lambda^3/(2*%pi^2*c^2));
b2=-(D*lambda^2)/(2*%pi*c);
o=2*%pi*(c/lp-c/lc);
d=(b2*o)+(b3*o^2)/2;
n=alpha^2/alpha^2*r*4*d^2*(1+(4*(sin(r*d*l))^2*%e^(-alpha*l))/(1-%e^(-alpha*l)^2));
n=n*10^-18;
// Displaying results in the command window
printf("\n XPM efficiency = %0.3f *10^-3",n);


// The answers vary due to round off error
```

Scilab code Exa 10.5 Calculate the efficiency of the non degenerate FWM tone if beta2 equal to negative 4 and beta2 equal to 0

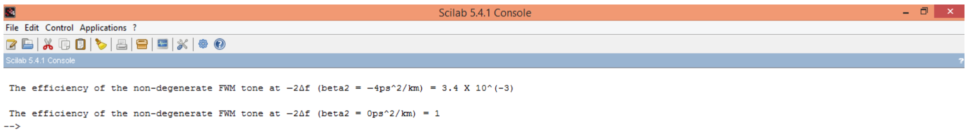

Figure 10.5: Calculate the efficiency of the non degenerate FWM tone if beta2 equal to negative 4 and beta2 equal to 0

```
// Example 10.5
// Calculate the efficiency of the non−degenerate
   FWM tone at   2 f if (a) beta2 =   4ps^2/km, (
   b) beta2 = 0ps^2/km.
// Page no 453

clc;
clear;
close;

//Given data
f=50*10^9;                                        //
   The bandwidth
alpha= 0.046*10^-3;                               //
   The fiber loss coefficient
L=40*10^3;                                        //
   The fiber length

Leff=(1-exp(-(alpha*L)))/alpha;                  //
   Effective fiber length

// (a) Calculate the efficiency of the non−
```

```
degenerate FWM tone at     2   f  beta2 =    4ps^2/
km
bet21=-4*10^(-12);
j=-1;
k=0;
l=1;
n=j+k-l;

bet1=bet21*10^(-12)/10^(3)*(2*%pi*f)^2*n;

//The efficiency of the non-degenerate FWM tone
neta1=(alpha^2+4*exp(-alpha*L*10^3)*(sind(bet1*(L
    *10^3)/2))/Leff^2)/(alpha^2+bet1^2);

//Displaying results in the command window
printf("\n The efficiency of the non-degenerate FWM
    tone at     2   f  (beta2 =    4ps^2/km) = %0.1f X
    10^(-3) ",neta1*10^3);

// (b) Calculate the efficiency of the non-
    degenerate FWM tone at     2   f  beta2 = 0ps^2/km
bet22=0*10^(-12);
j=-1;
k=0;
l=1;
n=j+k-l;

bet2=bet22*10^(-12)/10^(3)*(2*%pi*f)^2*n;

//The efficiency of the non-degenerate FWM tone
neta2=(alpha^2+4*exp(-alpha*L*10^3)*(sind(bet2*(L
    *10^3)/2))/Leff^2)/(alpha^2+bet2^2);

//Displaying results in the command window
printf("\n\n The efficiency of the non-degenerate
    FWM tone at     2   f  (beta2 = 0ps^2/km) = %0.0f ",
    neta2);
```

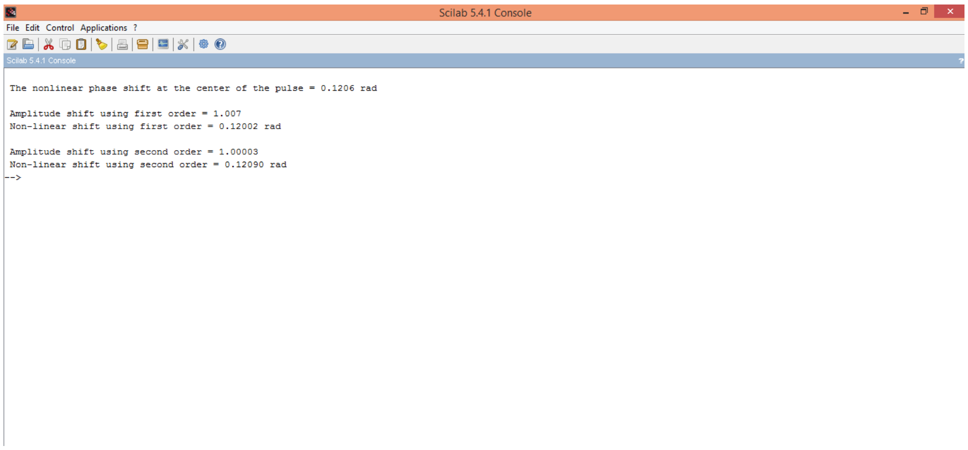

Figure 10.6: To find the nonlinear phase shift at the center of the pulse and Compare the exact results with those obtained using first and second order perturbation theory

Scilab code Exa 10.6 To find the nonlinear phase shift at the center of the pulse and Compare the exact results with those obtained using first and second order perturbation theory

```
// Example 10.6
// to find the nonlinear phase shift at the center
   of the pulse. Compare the exact results with
   those obtained using first and second-order
   perturbation theory
// Page no 469

clc;
clear;
close;

//Given data
```

```
P=6*10^(-3);                    // The peak power of rectangular pulse
L=40*10^3;                      // Fiber of length
Floss=0.2;                      // The fiber loss (dB/Km)
gamm=1.1*10^(-3);

alpha=Floss/4.343;              // Attenuation coefficient
Zeff=(1-exp(-alpha*10^(-3)*L))/alpha*10^3;

// The nonlinear phase shift at the center of the pulse
phi=gamm*P*Zeff;                // Nonlinear phase shift

//Displaying results in the command window
printf("\n The nonlinear phase shift at the center of the pulse = %0.4f rad ",phi);


// Results using first order
B01=sqrt(1+gamm^2*P^2*(Zeff)^2);   // Amplitude shift
thet1=atan(gamm*P*Zeff);           // Non-linear phase shift

//Displaying results in the command window
printf("\n\n Amplitude shift using first order = %0.3f ",B01);
printf("\n Non-linear shift using first order = %0.5f rad",thet1);

// Results using second order
x=1-((gamm)^2/2*P^2*Zeff^2);
y=gamm*P*Zeff;
thet2=atan(y/x);                //
```

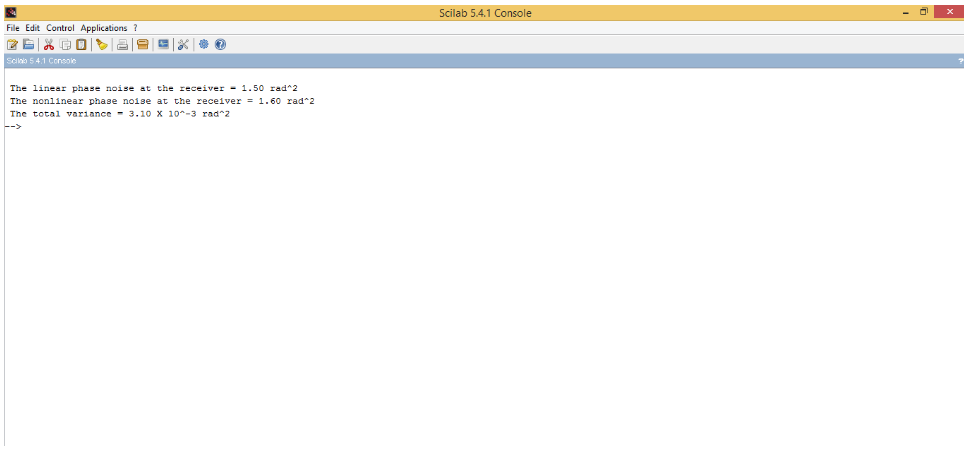

Figure 10.7: Calculation of the variance of linear phase noise and nonlinear phase noise at the receiver

```
   Nonlinear phase shift
B02=x/cos(thet2);                                        //
   Amplitude shift

//Displaying results in the command window
printf("\n\n Amplitude shift using second order = %0
   .5f ",B02);                     // Answer is varying due
   to round-off error
printf("\n Non-linear shift using second order = %0
   .5f rad",thet2);                 // Answer is varying due
    to round-off error
```

Scilab code Exa 10.7 Calculation of the variance of linear phase noise and nonlinear phase noise at the receiver

```
// Example 10.7
// Calculation of the variance of (a) linear phase
   noise, (b) nonlinear phase noise at the receiver
```

```
// Page no 477

clc;
clear;
close;

//Given data

alpha=0.0461;                    // Loss coeffient
na=20;                           // No of amplifiers
L=80;                            // Amplifier spacing
tb=25*10^-12;                    // Pulse width
P=2*10^-3;                       // Peak power
c=3*10^8;                        // Velocity of light
lambda=1550*10^-9;
n=1.5;                           // Spontaneous emission factor
h=6.626*10^-34;                  // Planck constant
r0=1.1*10^-3;                    // Nonlinear coefficient

// a) linear phase noise at the receiver
G=exp(alpha*L);
f=c/lambda;
R=h*f*(G-1)*n;
E=P*tb;
rl=(na*R)/(2*E);
rl=rl*10^3;

// (b) nonlinear phase noise at the receiver
Le=(1-exp(-alpha*L))/alpha;
rnl=((na-1)*na*(2*na-1)*R*E*r0^2*Le^2)/(3*tb^2);
rnl=rnl*10^9;

t=rl+rnl;

//Displaying results in the command window
printf("\n The linear phase noise at the receiver = %0.2f rad^2 ",rl);
```

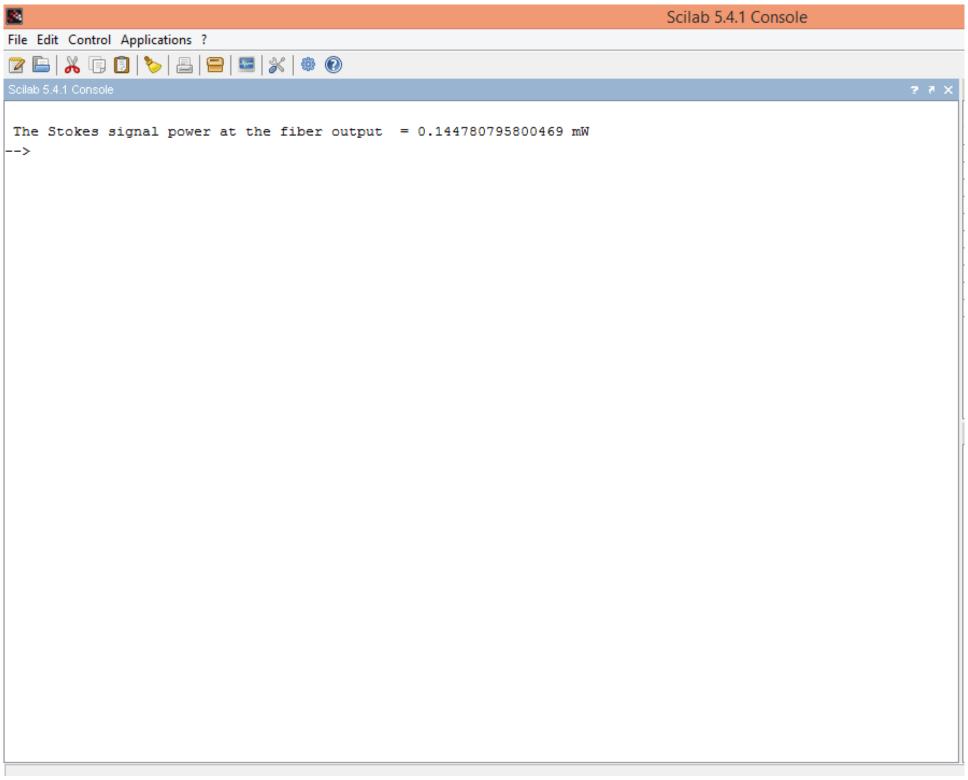

Figure 10.8: Calculation of the Stokes signal power at the fiber output

```
printf("\n The nonlinear phase noise at the receiver
    = %0.2f rad^2 ",rnl);
printf("\n The total variance = %0.2f X 10^-3 rad^2
    ",t);
```

Scilab code Exa 10.8 Calculation of the Stokes signal power at the fiber output

```
// Example 10.8
```

```
// Calculation of the Stokes signal power at the
   fiber output
// Page no 480

clc;
clear;
close;

//Given data
p1=20;                          // Input power pump
ps=-10;                         // Input Stokes s
   signal power
alpha=0.08;
L=2;                           // Length of fiber
alpha1=0.046;
A=40*10^-12;                  // Effective area of
   fiber
g=1*10^-13;                   // Raman coefficient of
   the fiber

// The Stokes signal power at the fiber output
p1=10^(p1/10);
ps=10^(ps/10);
Le=(1-exp(-alpha*L))/alpha;
s=(g*p1*Le)/A;
d=alpha1*L;
pd=ps*%e^(-d+s);



// Displaying results in the command window
printf("\n The Stokes signal power at the fiber
   output = %0.15f mW ",pd);
```

Chapter 11

Digital Signal Processing

Scilab code Exa 11.1 Calculation of the minimum number of taps needed to compensate for the fiber dispersion

```
// Example 11.1
// Calculation of the minimum number of taps needed
   to compensate for the fiber dispersion
// Page no 509

clc;
clear;
close;

// Given data
b=22*10^-27;                    // Power launched in
   port 1
l=800*10^3;                     // Power launched in
   port 2
T=50*10^-12;                   // Power launched in
   port 3


```

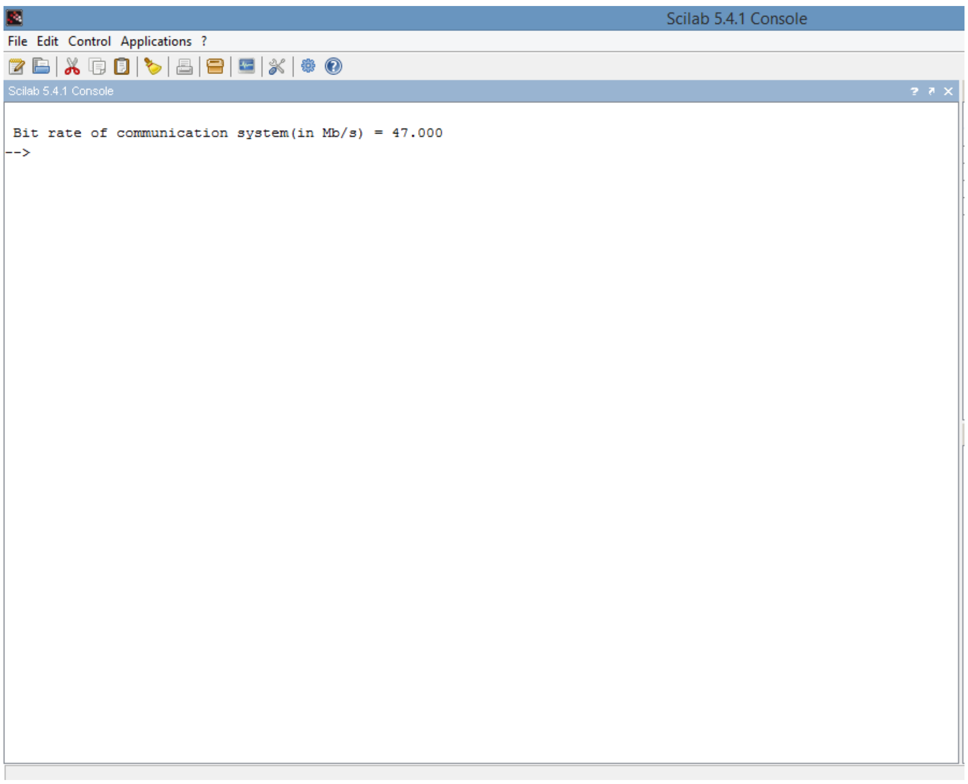

Figure 11.1: Calculation of the minimum number of taps needed to compensate for the fiber dispersion

```
// Bit rate of communication system
k=ceil((%pi*b*l)/T^2);
n=(2*k)+1;


// Displaying results in the command window
printf("\n The number of the taps = %0.3f ",n);

// The answers vary due to round off error
```