Scilab Textbook Companion for Introductory Circuit Analysis by Robert L. Boylestad[1]

Created by
Zaid Siddique
BE
Electronics Engineering
AIKTC, Mumbai University
College Teacher
None
Cross-Checked by
None

October 4, 2018

[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: Introductory Circuit Analysis

Author: Robert L. Boylestad

Publisher: Prentice Hall, Upper Saddle River, N.J

Edition: 10

Year: 2003

ISBN: 9780132359049

Scilab numbering policy used in this document and the relation to the above book.

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

AP Appendix to Example(Scilab Code that is an Appednix to a particular Example of the above book)

For example, Exa 3.51 means solved example 3.51 of this book. Sec 2.3 means a scilab code whose theory is explained in Section 2.3 of the book.

Contents

List of Scilab Codes

Chapter 1

Introduction

Scilab code Exa 1.1 Example 1

```
//Chapter 1, Example 1.1
clc
//Variable Declaration
a1 = 532.6                              //(a) unit less
    variable
a2 = 4.02                               //(a) unit less
    variable
a3 = 0.036                              //(a) unit less
    variable
b1 = 0.04                               //(b) unit less
    variable
b2 = 0.003                              //(b) unit less
    variable
b3 = 0.0064                             //(b) unit less
    variable
c1 = 4.632                              //(c) unit less
    variable
c2 = 2.4                                //(c) unit less
    variable
d1 = 3.051                              //(d) unit less
    variable
```

```
d2 = 802                                  //(d) unit less variable
e1 = 1402                                 //(e) unit less variable
e2 = 6.4                                  //(e) unit less variable
f1 = 0.0046                               //(f) unit less variable
f2 = 0.05                                 //(f) unit less variable

//Calculation
//(a)
a = a1+a2+a3                              //addition function
//(b)
b = b1+b2+b3                              //addition function
//(c)
c = c1*c2                                 //Multiplication function
//(d)
d = d1*d2                                 //Multiplication function
//(e)
e = e1/e2                                 //Division function
//(f)
f = f1/f2                                 //Division function

//Results
printf("(a) %.1f + %.2f + %.3f = %.3f = %.1f \n",a1,a2,a3,a,a)
printf("(b) %.2f + %.3f + %.4f = %.4f = %.2f \n",b1,b2,b3,b,b)
printf("(c) %.3f * %.1f = %.4f = %d \n",c1,c2,c,c)
printf("(d) %.3f * %.0f = %.3f = %d \n",d1,d2,d,d)
printf("(e) %.0f / %.1f = %.4f = %d \n",e1,e2,e,e)
printf("(f) %.4f / %.2f = %.4f = %.2f",f1,f2,f,f)
```

Scilab code Exa 1.2 Example 2

```
//Chapter 1, Example 1.2
clc
//Variable Declaration
a = 1000                    //(a) unit less variable
b = 0.00001                 //(b) unit less variable

//Calculation
//(a)
a1 = 1/a                    //inverse function
//(b)
b1 = 1/b                    //inverse function

//Results
printf("(a) %d^-3 \n",a1*10000)
printf("(b) %d^+5",round(b1/10000))
```

Scilab code Exa 1.3 Example 3

```
//Chapter 1, Example 1.3
clc
//Variable Declaration
a1 = 1000                        //(a) unit less variable
a2 = 10000                       //(a) unit less variable
b1 = 0.00001                     //(b) unit less variable
b2 = 100                         //(b) unit less variable

```

```
//Calculation
//(a)
a = a1*a2                          //multiplication function
//(b)
b = b1*b2                          //multiplication function
//Results
printf("(a) %d^7 \n",a/1000000)
printf("(b) %d^-3",b*10000)
```

Scilab code Exa 1.4 `Example 4`

```
//Chapter 1, Example 1.4
clc
//Variable Declaration
a1 = 100000                        //(a) unit less variable
a2 = 100                           //(a) unit less variable
b1 = 1000                          //(b) unit less variable
b2 = 0.0001                        //(b) unit less variable

//Calculation
//(a)
a = a1/a2                          //Division function
//(b)
b = b1/b2                          //Division function
//Results
printf("(a) %d^3 \n",a/100)
printf("(b) %d^7",b/1000000)
```

Scilab code Exa 1.5 Example 5

```
//Chapter 1, Example 1.5
clc
//Variable Declaration
a1 = 100                              //(a) unit less variable
b1 = 1000                             //(b) unit less variable
c1 = 0.01                             //(c) unit less variable

//Calculation
//(a)
a = a1^4                              //multiplication function
//(b)
b = b1^-2                              //multiplication function
//(c)
c = c1^-3                              //multiplication function
//Results
printf("(a) %d^8 \n",a/10000000)
printf("(b) %d^-6 \n",b*10000000)
printf("(c) %d^6",round(c/100000))
```

Scilab code Exa 1.6 Example 6

```
//Chapter 1, Example 1.6
clc
//Variable Declaration
```

```
a1 = 6300                         //(a) unit less variable
a2 = 75000                        //(a) unit less variable
b1 = 0.00096                      //(b) unit less variable
b2 = 0.000086                     //(b) unit less variable

//Calculation
//(a)
a = a1 + a2                      //addition function
//(b)
b = b1 - b2                      //subtraction function
//Results
printf("(a) %.1f x 10^3 \n",a/1000)
printf("(b) %.1f x 10^-5",b*100000)
```

Scilab code Exa 1.7 Example 7

```
//Chapter 1, Example 1.7
clc
//Variable Declaration
a1 = 0.0002                         //(a) unit less variable
a2 = 0.000007                       //(a) unit less variable
b1 = 340000                         //(b) unit less variable
b2 = 0.00061                        //(b) unit less variable

//Calculation
//(a)
a = a1*a2                          //multiplication
```

```
    function
//(b)
b = b1*b2                                    //multiplication
    function
//Results
printf("(a) %d x 10^-10 \n",a*10000000000)
printf("(b) %.1f",b)
```

Scilab code Exa 1.8 Example 8

```
//Chapter 1, Example 1.8
clc
//Variable Declaration
a1 = 0.00047                                   //(a) unit less
    variable
a2 = 0.002                                     //(a) unit less
    variable
b1 = 690000                                    //(b) unit less
    variable
b2 = 0.00000013                                //(b) unit less
    variable

//Calculation
//(a)
a = a1/a2                                      //division function
//(b)
b = b1/b2                                      //division function
//Results
printf("(a) %.1f x 10^-2 \n",a*100)
printf("(b) %.2f x 10^12",b/1000000000000)
```

Scilab code Exa 1.9 Example 9

```
//Chapter 1, Example 1.9
clc
//Variable Declaration
a1 = 0.00003                    //(a) unit less variable
b1 = 90800000                   //(b) unit less variable

//Calculation
//(a)
a = a1**3                       //multiplication function
//(b)
b = b1**2                       //multiplication function
//Results
printf("(a) %d x 10^-15\n",a*10**15)
printf("(b) %.4f x 10^14",b/10**14)
```

Scilab code Exa 1.10 `Example 10`

```
//Chapter 1, Example 1.10
clc
//Variable Declaration
a = 1000000      //in ohms
b = 1000000      //in meters
c = 0.0001       //in second
d = 0.000001     //in farad

//Results
printf("(a) %d x 10^6 ohms = %d megaohm \n",a/10**6,a/10**6)
printf("(b) %d x 10^3 meters = %d kilometers \n",b/10**4,b/10**4)
printf("(c) %.1f x 10-3 second = %.1f millisecond \n
```

```
",c*1000,c*10**3)
printf("(d) %d x 10-6 farad = %d microfarad",d*10**6,d*10**6)
```

Scilab code Exa 1.11 Example 11

```
//Chapter 1, Example 1.11
clc
//Variable Declaration
a = 41200           //in meter
b = 0.00956         //in Joule
c = 0.000768        //in sec
d1 = 8400           //in meter
d2 = 0.06           //in meter
e1 = 0.0003         //in sec

//Calculation
d = d1/d2           //in meter
e = e1^4        //in sec

//Results
printf("(a) %.1f km \n",a/1000)
printf("(b) %.2f mJ \n",b*1000)
printf("(c) %d us \n",c*10**6)
printf("(d) %d km \n",d/1000)
printf("(e) %.4f ps",e*10**12)
```

Scilab code Exa 1.12 Example 12

```
//Chapter 1, Example 1.12
clc
//Variable Declaration
a = 20*10^3      //frequency in hertz
```

```
b = 0.01          //timeperiod in millisec
c = 0.002         //in Km

//Results
printf("(a) %.2f MHz \n",a/10**6)
printf("(b) %d us \n",b*10**3)
printf("(c) %d mm",c*10**6)
```

Scilab code Exa 1.13 Example 13

```
//Chapter 1, Example 1.13
clc
//Variable Declaration
a1 = 6.8                    //time in minutes
a2 = 60                     //60second in 1 minute
b1 = 0.24                   //measurement in meter
b2 = 100                    //100centimeter in 1 meter

//Calculation
a = a1*a2
b = b1*b2

//Results
printf("(a) %d s  \n",a)
printf("(b) %d cm",b)
```

Scilab code Exa 1.14 Example 14

```
//Chapter 1, Example 1.14
clc
//Variable Declaration
a1 = 0.5                  //half day
a2 = 24                   //24hours in 1day
```

```
a3 = 60                       //60minutes in 1hour
b1 =2.2                       //yards
b2 = 3                        //3ft in 1 yard
b3 = 12                       //12inches in 1ft
b4 = 39.37                    //in inches

//Calculation
a = a1*a2*a3
b = (b1*b2*b3)/b4

//Results
printf("(a) %d min \n",a)
printf("(b) %.3f m",b)
```

Scilab code Exa 1.15 Example 15

```
//Chapter 1, Example 1.15
clc
//Variable Declaration
a1 =100                        //in kilometers
a2 = 1000                      //in meters
a3 = 39.37                     //in inches
a4 = 12                        //in inches
a5 = 5280                      //in ft
b1 = 60                        //in minutes
b2 = 4                         //in hours

//Calculation
a = (a1*a2*a3)/(a4*a5)
b = b1/b2

//Results
printf("(a) %.2f mi/h \n",a)
printf("(b) %d mi/h",b)
```

Scilab code Exa 1.16 Example 16

```
//Chapter 1, Example 1.16
clc
//Variable Declaration
a1 = 9
a2 = 3
c1 = 3
c2 = 9
c3 = 4
d1 = 1/4
d2 = 1/6
d3 = 2/3

//Calculation
a = sqrt(a1/a2)
b = (c1+c2)/c3
c = d1+d2+d3

//Results
printf("(a) %.2f \n",a)
printf("(b) %.d \n",b)
printf("(c) %.2f ",c)
```

Chapter 2

Current and Voltage

Scilab code Exa 2.1 Example 1

```
//Chapter 2, Example 2.1
clc
//Variable Declaration
q = 0.16                    //charge in coulomb (C)
t = 64*10^-3                //time in second (sec)

//Calculation
I = q/t                     //current in ampere (A)

//Results
printf("I = %.2f A \n",I)
```

Scilab code Exa 2.2 Example 2

```
//Chapter 2, Example 2.2
clc
//Variable Declaration
Q = 641*10^-3              //charge in coulomb
```

```
I = 5*10^-3                    //current in ampere
//Calculation
t = Q/I                        //time in seconds
//Results
printf("t = %.3f s",t/100)
```

Scilab code Exa 2.3 Example 3

```
//Chapter 2, Example 2.3
clc
//Variable Declaration
W = 60                          //in joule
Q = 20                          //in coulomb
//Calculation
V = W/Q                         //ans in volt
//Results
printf("V = %d V",V)
```

Scilab code Exa 2.4 Example 4

```
//Chapter 2, Example 2.4
clc
//Variable Declaration
Q = 50*10^-6                   //charge in coulomb
V = 6                        //in volt

//Calculation
W = Q*V                      //in microsec joule

//Results
printf("W = %d uJ",W*10^6)
```

Scilab code Exa 2.5 Example 5

```
//Chapter 2, Example 2.5
clc
//Variable Declaration
a = 450*10^-3            //current in millampere
b = 600*10^-3            //current in millampere
d = 45                   //temperature in degree from
    graph 2.18.b
m = 60                   //unit time in sec

//Calculation
life = (a/b)*m      //life in minutes

//Results
printf("a. Life = %.d min \n",life)
printf("b. High temperature = %d degreeC",d)
```

Chapter 3

Resistance

Scilab code Exa 3.1 Example 1

```
//Chapter 3, Example 3.1
clc
//Variable Declaration
p = 10.37                //in CM.ohm/ft
l = 100                  //length in ft
a = 400                  //area in CM

//Calculation
R = p*l/a                //resistance in ohm

//Results
printf("R = %.2f ohm",R)
```

Scilab code Exa 3.2 Example 3

```
//Chapter 3, Example 3.3
clc
//Variable Declaration
```

```
p = 10.37                    //in CM.ohm/ft
l = 3                        //length in ft
a = 3.185*10^6               //area in cm

//Calculation
R = p*l/a                    //resistance in ohm

//Results
printf("R = %.3f x 10^-6 ohm",R*10^6)
```

Scilab code Exa 3.3 `Example 3`

```
//Chapter 3, Example 3.3
clc
//Variable Declaration
p = 10.37                    //in CM.ohm/ft
l = 3                        //length in ft
a = 3.185*10^6               //area in cm

//Calculation
R = p*l/a                    //resistance in ohm

//Results
printf("R = %.3f x 10^-6 ohm",R*10^6)
```

Scilab code Exa 3.4 `Example 4`

```
//Chapter 3, Example 3.4
clc
//Variable Declaration
l = 650                       //in ft
o = 0.6282                    // in ohm
L = 1000                      //in ft
```

```
//Calculation
R = l*o/L                        //resistance in ohm

//Results
printf("R = %.3f ohm",R)
```

Scilab code Exa 3.5 Example 5

```
//Chapter 3, Example 3.5
clc
//Variable Declaration
A = 6529.9                              //area in cm

//Calculation
d = sqrt(6529.9)/1000                   //answer in inch

//Results
printf("d = %.4f in",d)
```

Scilab code Exa 3.6 Example 6

```
//Chapter 3, Example 3.6
clc
//Variable Declaration
p = 10.37                   //in CM.ohm/ft
l = 100                     //length in ft
r = 0.025                   //in ohm

//Calculation
A = p*l/r                   //in CM

//Results
printf("A = %d cm",A)
```

Scilab code Exa 3.7 Example 7

```
//Chapter 3, Example 3.7
clc
//Variable Declaration
l = 3048                    // in cm
d = 0.032                   //in cm
r  = 1.723*10^-6            //in ohm.cm
p = 3.14
r1 = 10.37                  //in cm.ohm/ft
l1 = 100                    //in ft
a = 159.79                  //in cm

//Calculation
A = (p*d^2)/4
R = (r*l)/A                 //resistance in ohm
R1 = (r1*l1)/a              //resistance in ohm
//Results
printf("R = %.1f ohm \n",R)
printf("R = %.1f ohm",R1)
```

Scilab code Exa 3.8 Example 8

```
//Chapter 3, Example 3.8
clc
//Variable Declaration
r = 100                   //sheet resistance in ohm
l = 0.6                   //length in cm
w = 0.3                   //in cm

//Calculation
```

```
R = (r*l)/w                    //resistance in ohm

//Results
printf("R = %d ohm",R)
```

Scilab code Exa 3.9 Example 9

```
//Chapter 3, Example 3.9
clc
//Variable Declaration
r = 50                          //copper wire in ohm
t1 = 334.5                      //in degree celsius
t2 = 254.5                      //in degree celsius


//Calculation
R = (r*t1)/t2                   //in ohm

//Results
printf("R = %.2f ohm",R)
```

Scilab code Exa 3.10 Example 10

```
//Chapter 3, Example 3.10
clc
//Variable Declaration
r = 30                         //copper wire in ohm
t1 = 194.5                     //in degree celsius
t2 = 234.5                     //in degree celsius

//Calculation
R = (r*t1)/t2                  //in ohm

```

```
//Results
 printf("R2 = %.2f ohm",R)
```

Scilab code Exa 3.11 Example 11

```
//Chapter 3, Example 3.11
clc
//Variable Declaration
r1 = 120*10^-3                //in milliohm
t1 = 256                      //in degree celsius
r2 = 100*10^-3                //in milliohm
t2 = 236                      //in degree celsius
//Calculation
T = r1*(t1/r2)-t2             //in degree celsius

//Results
printf("T2 = %.1f C",T)
```

Scilab code Exa 3.12 Example 12

```
//Chapter 3, Example 3.12
clc
//Variable Declaration
r =1000                     //in ohm
p = 2500
t1 = 60                     //in degree celsius
t2 = 20                     //in degree celsius
a = 10^6

//Calculation
D = (r/a)*p*(t1-t2)
R = r+D                     //resistance in ohm
//Results
```

```
printf("R = %d ohm",R)
```

Scilab code Exa 3.14 Example 14

```
//Chapter 3, Example 3.14
clc
//Variable Declaration
a = 0.7
b = 1.4

//Calculation
Gn = a/b

//Results
printf("Gn = %.1fGi",Gn)
```

Chapter 4

Ohms Law Power and Energy

Scilab code Exa 4.1 Example 4

```
//Chapter 4, Example 4.1
clc
//Variable Declaration
E = 9                       //battery in volt
R = 2.2                     //resistor in ohm


//Calculation
I = E/R                    //current in ampere

//Results
printf("I =  %.2f A",I)
```

Scilab code Exa 4.2 Example 2

```
//Chapter 4, Example 4.2
clc
//Variable Declaration
```

```
E = 120                    //applied voltage
I = 500*10^-3              //current in ampere

//Calculation
R = E/I                    //in ohm

//Results
printf("R = %d ohm",R)
```

Scilab code Exa 4.3 Example 3

```
//Chapter 4, Example 4.3
clc
//Variable Declaration
V = 16                  //voltage
R = 2*10^3              //in ohm

//Calculation
I = V/R                 //in miliampere

//Results
printf("I = %d mA",I*1000)
```

Scilab code Exa 4.4 Example 4

```
//Chapter 4, Example 4.4
clc
//Variable Declaration
I = 1.5                        //current in ampere
R = 80                         //in ohm

//Calculation
E = I*R                        //voltage
```

```
//Results
printf("E = %d V",E)
```

Scilab code Exa 4.5 Example 5

```
//Chapter 4, Example 4.5
clc
//Variable Declaration
V1 = 6                      //in volts
I1 = 3*10^-3                //in miliampere
V2 = 2                      //in volts
I2 = 1*10^-3                //in miliampere
//Calculation
Rdc = V1/I1                 //in kiloohm
R = V2/I2                   //in kiloohm

//Results
printf("Rdc = %d kohm  \n",Rdc/10^3)
printf("R = %d kohm",R/10^3)
```

Scilab code Exa 4.6 Example 6

```
//Chapter 4, Example 4.6
clc
//Variable Declaration
V = 120                      //in volts
I = 5                        //in ampere

//Calculation
P = V*I                     //power in kwatt

//Results
```

```
printf("P = %.1f kW",P/10^3)
```

Scilab code Exa 4.7 Example 7

```
//Chapter 4, Example 4.7
clc
//Variable Declaration
I = 4                        //in ampere
R = 5                        //in ohm

//Calculation
P = (I*I)*R                 //power in watt

//Results
printf("P = %d W",P)
```

Scilab code Exa 4.8 Example 8

```
//Chapter 4, Example 4.8
clc
//Variable Declaration
I = 0.625                    //in ampere
V = 120                      //in volts


//Calculation
P = V*I                   //power in watt
R = V/I                   //in ohm

//Results
printf("P = %d W \n",P)
printf("R = %d ohm",R)
```

Scilab code Exa 4.9 Example 9

```
//Chapter 4, Example 4.9
clc
//Variable Declaration
P = 20*10^-3                    //power in watt
R = 5*10^3                      //in ohm

//Calculation
I = sqrt(P/R)                  //current in miliampere

//Results
printf("I = %d mA",I*1000)
```

Scilab code Exa 4.10 Example 10

```
//Chapter 4, Example 4.10
clc
//Variable Declaration
p1 = 1492                        //output power
n = 0.75                         //efficiency
E = 220                          //applied voltage

//Calculation
Pi = p1/n                        //inputpower in watt
I = Pi/E                         //current in ampere

//Results
printf("Pi = %.2f W \n",Pi)
printf("I = %.2f A",I)
```

Scilab code Exa 4.11 Example 11

```
//Chapter 4, Example 4.11
clc
//Variable Declaration
n = 0.80                    //efficiency
a2 = 120                    //in volts
a3 = 8                      //in ampere
b1 = 1                      //motor in horsepower
b2 = 746                    //power in watt
//Calculation
Po = n*a2*a3                //output power
I = Po*(b1/b2)               //in horsepower
//Results
printf("H = %.3f hp",I)
```

Scilab code Exa 4.12 Example 12

```
//Chapter 4, Example 4.12
clc
//Variable Declaration
n = 0.85                 //efficiency
Wi = 50                //applied energy

//Calculation
Wo = n*Wi              //output energy

//Results
printf("Wo = %.1f J",Wo)
```

Scilab code Exa 4.13 Example 13

```
//Chapter 4, Example 4.13
clc
//Variable Declaration
n1 = 0.90                    //efficiency1
n2 = 0.85                    //efficiency2
n3 = 0.95                    //efficiency3

//Calculation
nT = n1*n2*n3              //efficiency in percentage

//Results
printf("nT = %.1f percentage",nT*100)
```

Scilab code Exa 4.14 Example 14

```
//Chapter 4, Example 4.14
clc
//Variable Declaration
n1 = 0.40                      //efficiency1
n2 = 0.85                      //efficiency2
n3 = 0.95                      //efficiency3

//Calculation
nT = n1*n2*n3             //efficiency in percentage

//Results
printf("nT = %.1f percentage",nT*100)
```

Scilab code Exa 4.15 Example 15

```
//Chapter 4, Example 4.15
```

```
clc
//Variable Declaration
cost = 9
r1 = 5360*10^3                //in kWh
r2 = 4650*10^3                //pervious reading

//Calculation
R = r1-r2                      //used
M = R*(cost/10^3)              //in dollar

//Results
printf("M = %.2f dollar",M/100)
```

Scilab code Exa 4.16 Example 16

```
//Chapter 4, Example 4.16
clc
//Variable Declaration
a1 = 60                       //bulb in watt
a2 = 24                       //hours per day
a3 = 365                      //days

//Calculation
Pt = a1*a2*a3                 //power in Wh
W = Pt/1000                   //energy in kWh

//Results
printf("W = %.2f kWh",W)
```

Scilab code Exa 4.17 Example 17

```
//Chapter 4, Example 4.17
clc
```

```
//Variable Declaration
p1 = 4*10^3                //energy in kWh
p2 = 205                   //power in watt

//Calculation
Pt = p1*1000               //power in watt
W = Pt/p2                  // in h
//Results
printf("W = %.2f h",W/1000)
```

Scilab code Exa 4.18 Example 18

```
//Chapter 4, Example 4.18
clc
//Variable Declaration
p1 = 5                     //motor in horsepower
p2 = 746                   //in W/hp
p3 = 2                     //in h
c = 9                      //in euro/kWh

//Calculation
Pt = p1*p2*p3               //power
W = Pt/1000                 //in kWh
cost = W*c                  //unit

//Results
printf("Cost = %.2f unit",cost)
```

Scilab code Exa 4.19 Example 19

```
//Chapter 4, Example 4.19
clc
//Variable Declaration
```

```
w1 = 1200                    //toaster in watt
w2 = 50                      //bulbs in watt
w3 = 400                     //washing machine in watt
w4 = 4800                    //electric clothes in watt
c = 9                        //cost in unit/kWh
//Calculation
W = ((w1*0.5)+(6*w2*4)+(w3*0.75)+(w4*0.33))/1000
        //energy in kWh
cost = W*c                        //in euro

//Results
printf("Cost = %.1f unit",cost)
```

Chapter 5

Series Circuits

Scilab code Exa 5.1 Example 1

```
//Chapter 5, Example 5.1
clc
//Variable Declaration
r1 = 2                          //resistance1
r2 = 1                          //resistance2
r3 = 5                          //resistance3
E = 20                          //supply voltage



//Calculation
rt = r1+r2+r3                   //total resistance
I = E/rt                        //current
V1 = I*r1                       //voltage1
V2 = I*r2                       //voltage2
V3 = I*r3                       //voltage3
P1 = V1*I                       //power1
P2 = I*I*r2                     //power2
P3 = V3*V3/r3                   //power3
Pdel1 = E*I               //power delivered by source
Pdel = P1+P2+P3           //Total power
```

```
//Results
printf("a. RT = %d ohm \n",rt)
printf("b. Is = %.1f A \n",I)
printf("c. V1 = %d V\n",V1)
printf("   V2 = %.1f V \n",V2)
printf("   V3 = %.1f V \n",V3)
printf("d. P1 = %.1f W \n",P1)
printf("   P2 = %.2f W \n",P2)
printf("   P3 = %.2f W \n",P3)
printf("e. Pdel = %d W \n",Pdel1)
printf("   %d W = %d W  (checks) \n",Pdel1,Pdel)
```

Scilab code Exa 5.2 Example 2

```
//Chapter 5, Example 5.2
clc
//Variable Declaration
N = 3                          //number of resistance
R1 = 7                         //resistance1
R2 = 4                         //resistance2
E = 50                         //supply voltage
//Calculation
Rt = N*R1+R2                   //total resistance
I = E/Rt                       //current in ampere
V2 = I*R2                      //voltage2
//Results
printf("RT = %d ohm \n",Rt)
printf("I = %d A \n",I)
printf("V2 = %d V",V2)
```

Scilab code Exa 5.3 Example 3

```
//Chapter 5, Example 5.3
```

```
clc
//Variable Declaration
Rt = 12*10^3                    //total resistance
R2 = 4*10^3                     //resistance2
R3 = 6*10^3                     //resistance3
I = 6*10^-3                     //current in ampere
//Calculation
R1 = Rt-(R2+R3)                 //resistance1
E = I*Rt                        //voltage
//Results
printf("R1 = %d kohm \n",R1/10^3)
printf("E = %d V",E)
```

Scilab code Exa 5.4 Example 4

```
//Chapter 5, Example 5.4
clc
//Variable Declaration
e1 = 16                          //supply voltage1
v2 = 4.2                         //voltage2
e2 = 9                           //supply voltage2
e = 32                           //supply voltage
v = 12                           //voltage
v3 = 6                           //voltage of R2
v4 =14                           //voltage of R3
//Calculation
v1 = e1-v2-e2                    //voltage1
vx = e-v                         //unknown voltage
vx = v3+v4                       //unknown voltage
//Results
printf("V1 = %.1f V \n",v1)
printf("Vx = %d V \n",vx)
printf("Vx = %d V",vx)
```

Scilab code Exa 5.5 Example 5

```
//Chapter 5, Example 5.5
clc
//Variable Declaration
e1 = 25                                    //supply voltage1
e2 = 15                                    //supply voltage2
e3 = 20                                     //supply voltage3
//Calculation
v1 = e1+e2                                  //voltage1
v2 = -e3                                    //voltage2
//Results
printf("V1 = %d V \n",v1)
printf("V2 = %d V",v2)
```

Scilab code Exa 5.6 Example 6

```
//Chapter 5, Example 5.6
clc
//Variable Declaration
v1 = 60                                //supply voltage
v2 = 40                                //voltage1
v3 = 30                                //voltage2
v4 = -6                                //voltage1
v5 = 14                                //supply voltage
v6 = 2                                 //voltage2
//Calculation
va = v1+v3-v2                          //unknown voltage
vb = v4-v5+v6                          //unknown voltage
//Results
printf("Vx = %d V \n",va)
printf("Vx = %d V \n",vb)
```

Scilab code Exa 5.7 Example 7

```
//Chapter 5, Example 5.7
clc
//Variable Declaration
r1 = 4                          //resistor1
r2 = 6                          //resistor2
e = 20                          //supply voltage

//Calculation
rt = r1+r2                      //total resistance
i = e/rt                        //current
v1 = i*r1                       //voltage1
v2 = i*r2                       //voltage2
p4 = v1*v1/r1                   //power of R1
p6 = (i*i)*r2                   //power of R2
pe = e*i                        //power of source
pe1 = p4+p6                     //Total power
sv = v1+v2                      //Total voltage
//Results
printf("a.  Rt = %d ohm \n",rt)
printf("b.  I = %d A \n",i)
printf("c.  V1 = %d V \n",v1)
printf("    V2 = %d V \n",v2)
printf("d.  P4 = %d W \n",p4)
printf("    P6 = %d W \n",p6)
printf("e.  Pe = %d W \n",pe)
printf("    %d W = %d W (checks)\n",pe,pe1)
printf("f.  %d V = %d V (checks)",e,sv)
```

Scilab code Exa 5.8 Example 8

```
//Chapter 5, Example 5.8
clc
//Variable Declaration
e = 54                          //supply voltage
v1 = 18                         //voltage1
v3 = 15                         //voltage3
r2 = 7                          //resistance2
//Calculation
v2 = e-v1-v3                    //voltage2
i = v2/r2                       //current in ampere
r1 = v1/i                       //resistance1
r3 = v3/i                       //resistance3

//Results
printf("a. V2 = %d V \n",v2)
printf("b.  I = %d A \n",i)
printf("c.  R1 = %d ohm \n",r1)
printf("    R3 = %d ohm \n",r3)
```

Scilab code Exa 5.9 Exampl 9

```
//Chapter 5, Example 5.9
clc
//Variable Declaration
r1 = 4                             //resistance1
r2 = 7                             //resistance2
e = 37.5                           //supply voltage
i = 2.5                            //current in ampere

//Calculation
rt = 2*r1+r2                       //total resistance
i = e/rt                           //current in ampere
v = i*r2                           //voltage of R2

//Results
```

```
printf("I = %.1f A \n",i)
printf("V7ohm = %.1f V \n",v)
```

Scilab code Exa 5.10 Example 10

```
//Chapter 5, Example 5.10
clc
//Variable Declaration
r1 = 20                         //resistor1
e = 64                          //supply voltage
r2 = 60                         //resistor2

//Calculation
rt = r1+r2                      //total resistance
v1 = r1*e/rt                    //voltage1

//Results
printf("V1 = %d V",v1)
```

Scilab code Exa 5.11 Example 11

```
//Chapter 5, Example 5.11
clc
//Variable Declaration
r1 = 2000                                //resistance1
r2 = 5000                                //resistance2
r3 = 8000                                //resistance3
e = 45                                   //supply voltage

//Calculation
rt = r1+r2+r3                           //total resistance
v1 = r1*e/rt                            //voltage1
v3 = r3*e/rt                            //voltage3
```

```
//Results
printf("V1 = %d V \n",v1)
printf("V2 = %d V",v3)
```

Scilab code Exa 5.12 Example 12

```
//Chapter 5, Example 5.12
clc
//Variable Declaration
r1 = 2000                               //resistance1
r2 = 5000                               //resistance2
r3 = 8000                               //resistance3
e = 45                                  //supply voltage


//Calculation
rt = r1+r2+r3                     //Total resistance
r = r1+r2                         //resistance of R1 or R2
v = r*e/rt                               //volage
//Results
printf("V = %d V",v)
```

Scilab code Exa 5.13 Example 13

```
//Chapter 5, Example 5.13
clc
//Variable Declaration
e = 20                                   //supply voltage
i = 4*10^-3                              //current
r2 = 1000                                //resistance2

//Calculation
r1 = 4*r2                                //resistance1
```

```
//Results
printf("R2 = %d Kohm \n",r2/1000)
printf("R1 = %d kohm",r1/10^3)
```

Scilab code Exa 5.14 Example 14

```
//Chapter 5, Example 5.14
clc
//Variable Declaration
va = 16                              //voltage Va
vb = 20                              //voltage Vb

//Calculation
vab = va-vb                         //voltage Vab

//Results
printf("Vab = %d V",vab)
```

Scilab code Exa 5.15 Example 15

```
//Chapter 5, Example 5.15
clc
//Variable Declaration
vab = 5                    //voltage Vab
vb = 4                     //voltage Vb

//Calculation
va = vab + vb             //voltage Va

//Results
printf("Va = %d V",va)
```

Scilab code Exa 5.16 Example 16

```
//Chapter 5, Example 5.16
clc
//Variable Declaration
va = 20                          //voltage Va
vb = -15                         //voltage Vb

//Calculation
vab = va-vb                      //voltage Vab

//Results
printf("Vab = %d V",vab)
```

Scilab code Exa 5.17 Example 17

```
//Chapter 5, Example 5.17
clc
//Variable Declaration
e1 = 10                              //supply voltage1
v1 = 4                                //voltage1
e2 = 20                              //supply voltage2
va = 10                              //voltage Va

//Calculation
vb = e1-v1                           //voltage Vb
vc = vb-e2                           //voltage Vc
vac = va-vc                          //voltage Vac

//Results
printf("Vb = %d V \n",vb)
printf("Vc = %d V \n",vc)
```

```
printf("Vac = %d V \n",vac)
```

Scilab code Exa 5.18 Example 18

```
//Chapter 5, Example 5.18
clc
//Variable Declaration
r2 = 25                         //resistance
r1 = 20                         //resistance in ohm
e1 = -19                        //supply voltage in V
e2 = 35                         //supply voltage in V
//Calculation
r = r1+r2                        //total resistance in
   ohm
e = e2-e1                        //total supply voltage
   in V
i = e/r                          //current in ampere A
vab = i*r2                       //in voltage Vab
vcb = -i*r1                      //in voltage Vcd
vc = e1                          //in voltage Vc

//Results
printf("Vab = %d V \n",vab)
printf("Vcb = %d V \n",vcb)
printf("Vc = %d V \n",vc)
```

Scilab code Exa 5.19 Example 19

```
//Chapter 5, Example 5.19
clc
//Variable Declaration
r1 = 4                                  //resistance1
e = 24                                  //supply voltage
```

```
r2 = 2                                   //resistance2

//Calculation
v1 = (r1*e)/(r1+r2)                      //voltage1
v2 = (r2*e)/(r1+r2)                      //voltage2

//Results
printf("V1 = %d V \n",v1)
printf("V2 = %d V \n",v2)
```

Scilab code Exa 5.20 `Example 20`

```
//Chapter 5, Example 5.20
clc
//Variable Declaration
r1 = 2                            //resistor1
e = 10                            //supply voltage
r2 = 3                            //resistor2
r3 = 5                            //resistor3

//Calculation
rt = r1+r2+r3                     //total resistance
vab = r1*e/rt                     //voltage Vab
vb = (r2+r3)*e/rt                 //voltage Vb
vc = 0                            //coltage Vc
vb1 = e-vab                       //voltage Vb
//Results
printf("a. Vab = +%d V \n",vab)
printf("b. Vb = %d V \n",vb)
printf(" or Vb = %d V \n",vb1)
printf("c. Vc = ground potential = %d V",vc)
```

Scilab code Exa 5.21 `Example 21`

```
//Chapter 5, Example 5.21
clc
//Variable Declaration
vnl = 40                        //voltage with no load
il = 77*10^-3                   //load current in ampere
rl = 0.5*10^3                   //load resistor in ohm

//Calculation
rint = (vnl/il)-rl         //internal resistance in ohm

//Results
printf("Rint = %.2f ohm",rint)
```

Scilab code Exa 5.22 Example 22

```
//Chapter 5, Example 5.22
clc
//Variable Declaration
e = 30                           //supply voltage
rint = 2                         //internal resistance
rl = 13                          //load resistance
vnl = 30                         //voltage with no load


//Calculation
il = e/(rint+rl)               //load current in ampere
vl = vnl-il*rint               //load voltage in volts
plost = il*il*rint             //power lost in Watt

//Results
printf("Vl = %d V \n",vl)
printf("Plost = %d W",plost)
```

Scilab code Exa 5.23 Example 23

```
//Chapter 5, Example 5.23
clc
//Variable Declaration
vnl   = 120                        //voltage with no load
vfl = 100                          //voltage with load

//Calculation
vr = ((vnl-vfl)/vfl)*100           //voltage regulation in
    percent
//Results
printf("VR = %d percent",vr)
```

Scilab code Exa 5.24 Example 24

```
//Chapter 5, Example 5.24
clc
//Variable Declaration
rint = 19.48                       //internal resistance
   in ohm
rl = 500                           //load resistor in ohm

//Calculation
vr = (rint/rl)*100                  //voltage regulation
   in percent

//Results
printf("VR = %.1f percent",vr)
```

Chapter 6

Parallel Circuits

Scilab code Exa 6.1 `Example 1`

```
//Chapter 6, Example 6.1
clc
//Variable Declaration
g1 = 1/3                        //conductance in seimens
g2 = 1/6                        //conductance in seimens

//Calculation
gt = g1+g2                       //total conductance in
   seimens
rt = 1/gt                        //resistance in ohm

//Results
printf("Gt = %.1f S \n",gt)
printf("Rt = %d ohm",rt)
```

Scilab code Exa 6.2 `Example 2`

```
//Chapter 6, Example 6.2
```

```
clc
//Variable Declaration
g1 = 0.5                      //conductance in seimens
g2 = 1/10                      //conductance in seimens

//Calculation
gt = g1 + g2                      //total conductance in
    seimens
rt = 1/gt                      //resistance in ohm

//Results
printf("Gt = %.1f S \n",gt)
printf("Rt = %.3f ohm",rt)
```

Scilab code Exa 6.3 Example 3

```
//Chapter 6, Example 6.3
clc
//Variable Declaration
r1 = 1/2                      //resistance in ohm
r2 = 1/4                      //resistance in ohm
r3 = 1/5                      //resistance in ohm

//Calculation
rt = r1+r2+r3
r = 1/rt                      //total resistance in
   ohm


//Results
printf("Rt = %.3f ohm",r)
```

Scilab code Exa 6.4 Example 4

```
//Chapter 6, Example 6.4
clc
//Variable Declaration
r1 = 12                         //resistor in ohm fig
    6.9
n1 = 3                          //number of component
r2 = 2                          //resistor in ohm fig
    6.10
n2 = 4                          //number of component
//Calculation
rt1 = r1/n1                     //total resistance in
    ohm fig 6.9
rt2 = r2/n2                     //total resistance in
    ohm fig 6.10

//Results
printf("a. Rt = %d ohm \n",rt1)
printf("b. Rt = %.1f ohm",rt2)
```

Scilab code Exa 6.5 Example 5

```
//Chapter 6, Example 6.5
clc
//Variable Declaration
r1 = 3                            //resistor in ohm
r2 = 6                            //resistor in ohm
//Calculation
rt = r1*r2/(r1+r2)                //resistance in ohm

//Results
printf("Rt = %d ohm",rt)
```

Scilab code Exa 6.6 Example 6

```
//Chapter 6, Example 6.6
clc
//Variable Declaration
r1 = 2                              //resistor in ohm
r2 = 4                              //resistor in ohm
r3 = 5                              //resistor in ohm

//Calculation
rt = 1/((1/r1)+(1/r2)+(1/r3))      //total resistance
    in ohm
rt1 = (r1*r2)/(r1+r2)              //R1 and R2
   parallel
rt2 = (rt1*r3)/(rt1+r3)            //Rt1 and R3
   parallel

//Results
printf("Rt = %.3f ohm \n",rt)
printf("Rt = %.3f ohm \n",rt2)
```

Scilab code Exa 6.7 Example 7

```
//Chapter 6, Example 6.7
clc
//Variable Declaration
r = 6                           //resistor in ohm
n = 3                           //number of component
r2 = 9                          //resistor in ohm
r4 = 72                         //resistor in ohm

//Calculation
rt1 = r/n                       //resistance in ohm
rt2 = (r2*r4)/(r2+r4)           //R2 and R4 parallel
rt3 = (rt1*rt2)/(rt1+rt2)       //total resistance in
   ohm

```

```

//Results
printf("Rt = %.1f ohm",rt3)
```

Scilab code Exa 6.8 Example 8

```
//Chapter 6, Example 6.8
clc
//Variable Declaration
rt = 9000                          //total resistance in ohm
r1 = 12000                         //resistor in ohm

//Calculation
r2 = (rt*r1)/(r1-rt)               //required resistor in ohm

//Results
printf("R2 = %d kohm",r2/10^3)
```

Scilab code Exa 6.9 Example 9

```
//Chapter 6, Example 6.9
clc
//Variable Declaration
rt = 16000                          //total resistance in ohm
r = 1.75


//Calculation
r1 = r*rt                          //required resistance in ohm
```

```

//Results
printf("R1 = %d kohm",r1/10^3)
```

Scilab code Exa 6.10 Example 10

```
//Chapter 6, Example 6.10
clc
//Variable Declaration
r1 = 30                                 //resistance in ohm fig 6.18
r2 = 30                                 //resistance in ohm fig 6.18
r3 = 30                                 //resistance in ohm fig 6.19
r4 = 1000                              //resistance in ohm fig 6.19
r5 = 0.1                               //resistance in ohm fig 6.20

//Calculation
rt = (r1*r2)/(r1+r2)                      //total resistance in ohm
rt1 = (rt*r3)/(rt+r3)                     //total resistance in ohm
rt2 = (rt*r4)/(rt+r4)                     //total resistance in ohm
rt3 = (rt*r5)/(rt+r5)                     //total resistance in ohm
//Results
printf("a. Rt = %d ohm \n",rt)
printf("b. Rt = %d ohm \n",rt1)
printf("c. Rt = %.3f ohm \n",rt2)
printf("d. Rt = %.3f ohm",rt3)
```

Scilab code Exa 6.11 Example 11

```
//Chapter 6, Example 6.11
clc
//Variable Declaration
r1 = 9                                  //resistance in ohm
r2 = 18                                 //resistance in ohm
e = 27                                  //voltage supply

//Calculation
rt = (r1*r2)/(r1+r2)                    //total resistance
    in ohm
is = e/rt                               //source current in
     ampere
i1 = e/r1                              //current in ampere
i2 = e/r2                              //current in ampere
is1 = i1+i2                             //applying KCL
p1 = e*i1                              //power of R1 in
    watt
p2 = e*i2                              //power of R2 in
    watt
ps = e*is                              //power supply in
    watt
ps1 = p1+p2                            //power supply in
    watt

//Results
printf("a. Rt = %d ohm \n",rt)
printf("b. Is = %.1f A \n",is)
printf("c. I1 = %d A \n",i1)
printf("   I2 = %.1f A \n",i2)
printf("   %.1f A = %.1f A \n",is1,is)
printf("d. P1 = %d W \n",p1)
printf("   I2 = %.1f W \n",p2)
```

```
printf("e.  Ps = %.1f W \n",ps)
printf("    Ps = %.1f W \n",ps1)
```

Scilab code Exa 6.12 Example 12

```
//Chapter 6, Example 6.12
clc
//Variable Declaration
r1 = 10                          //resistance in ohm
r2 = 20                          //resistance in ohm
rt = 4                           //total resistance in ohm
i1 = 4                           //current of R1 in ampere


//Calculation
r = (1/rt)-(1/r1)-(1/r2)
r3 = (1/r)                               //resistance in ohm
e = i1*r1                                //supply voltage
is = e/rt                                //source current in ampere
i2 = e/r2                                //current of R2 in ampere
p2 = i2*i2*r2                            //power of R2 in watt

//Results
printf("a. R3 = %d ohm \n",r3)
printf("b. E = %d V \n",e)
printf("c. Is = %d A \n",is)
printf("d. I2 = %d A \n",i2)
printf("e. P2 = %d W",p2)
```

Scilab code Exa 6.13 Example 13

```
//Chapter 6, Example 6.13
clc
//Variable Declaration
i1 = 2                          //current in ampere
i2 = 3                          //current in ampere
i5 = 1                          //current in ampere

//Calculation
i3 = i1+i2                      //current in ampere
i4 = i3+i5                      //current in ampere

//Results
printf("a: I3 = %d A \n",i3)
printf("b: I4 = %d A",i4)
```

Scilab code Exa 6.14 Example 14

```
//Chapter 6, Example 6.14
clc
//Variable Declaration
i = 5                               //entering current in
    ampere
i2 = 4                              //leaving current in
    ampere

//Calculation
i1 = i-i2                           //applying KCL, in
    ampere
i3 = i1                               // in ampere
i4 = i2                               // in ampere
```

```
i5 = i3+i4                        //applying KCL, in
   ampere
//Results
printf("I1 = %d A \n",i1)
printf("I3 = %d A \n",i3)
printf("I4 = %d A \n",i4)
printf("I5 = %d A",i5)
```

Scilab code Exa 6.15 `Example 15`

```
//Chapter 6, Example 6.15
clc
//Variable Declaration
i1 = 4                            //current in ampere
i2 = 3                            //current in ampere
i4 = 1                            //current in ampere

//Calculation
i3 = i1+i2                        //curremt in ampere
i5 = i3-i4                        //current in ampere
//Results
printf("I3 = %d A \n",i3)
printf("I5 = %d A",i5)
```

Scilab code Exa 6.16 `Example 16`

```
//Chapter 6, Example 6.16
clc
//Variable Declaration
i1 = 10                       //current in ampere
i2 = 12                       //current in ampere
i5 = 8                        //current in ampere

```

```
//Calculation
i7 = i1
i3 = i2-i1                    //current in ampere
i4 = i2-i5                    //current in ampere
i6 = i4-i3                    //current in ampere
i71 = i5+i6                    //current in ampere
//Results
printf("I7 = %d A \n",i7)
printf("I3 = %d A \n",i3)
printf("I4 = %d A \n",i4)
printf("I6 = %d A \n",i6)
printf(" %d A = %d A \n",i7,i71)
```

Scilab code Exa 6.17 Example 17

```
//Chapter 6, Example 6.17
clc
//Variable Declaration
r1 = 4000                          //resistance
is = 6                             //source current in
   ampere
r2 = 8000                          //resistance in ohm

//Calculation
i2  = (r1*is)/(r1+r2)              //current in ampere

//Results
printf("I2 = %d A",i2)
```

Scilab code Exa 6.18 Example 18

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

```
//Variable Declaration
r1 = 6                              //resistance in ohm
r2 = 24                             //resistance in ohm
r3 = 48                             //resistance  in ohm
i = 42*10^-3                        //current in ampere

//Calculation
r = (1/r1)+(1/r2)+(1/r3)
rt = 1/r                           //total resistance in
   ohm
i1 = (rt*i)/r1                     //current in ampere
r4 = (r2*r3)/(r2+r3)               //resistance in ohm
i11 = (r4*i)/(r4+r1)               //current in ampere

//Results
printf("I1 = %.2f mA \n",i1*1000)
printf("I1 = %.2f mA",i11*1000)
```

Scilab code Exa 6.19 `Example 19`

```
//Chapter 6, Example 6.19
clc
//Variable Declaration
r1 = 2                   //resistance in ohm
r2 = 4                   //resistance in ohm
i = 12                   //current in ampere

//Calculation
i1 = (r2*i)/(r1+r2)      //current in ampere
i2 = i-i1                //current in ampere
i21 = (r1*i)/(r1+r2)       //current in ampere
i3 = i                       //current in ampere
i31 = i1+i2                    //current in ampere

//Results
```

```
printf("I1 = %d A \n",i1)
printf("I2 = %d A \n",i2)
printf("I2 = %d A \n",i21)
printf("I3 = %d A \n",i3)
printf("I3 = %d A \n",i31)
```

Scilab code Exa 6.20 Example 20

```
//Chapter 6, Example 6.20
clc
//Variable Declaration
r2 = 7                         //resistance in ohm
i = 27*10^-3                   //current in ampere
i1 = 21*10^-3                  //current in ampere


//Calculation
r1 = (r2*(i-i1))/i1              //resistance in ohm
i2 = i-i1                       //current in ampere
v2 = i2*r2                      //voltage in volt
v1 = v2                         //voltage in volt
r11 = v1/i1                     //resistance in ohm

//Results
printf("R1 = %.f ohm \n",r1)
printf("R1 = %.f ohm",r11)
```

Scilab code Exa 6.21 Example 21

```
//Chapter 6, Example 6.21
clc
//Variable Declaration
e = 20                           //supply voltage
```

```

//Calculation
vab = e                              //output Vab

//Results
printf("Vab = %d V",vab)
```

Scilab code Exa 6.22 Example 22

```
//Chapter 6, Example 6.22
clc
//Variable Declaration
e1 = 10                            //supply voltage
e2 = 30                            //supply voltage

//Calculation
vab = e1                           //voltage Vab
vcd = e1-e2                        //voltage Vcd

//Results
printf("Vab = %d V \n",vab)
printf("Vcd = %d V",vcd)
```

Scilab code Exa 6.23 Example 23

```
//Chapter 6, Example 6.23
clc
//Variable Declaration
i = 0                                //current in ampere
r = 15000                            //resistance in ohm

//Calculation
v = i*r                               //voltage
```

```

//Results
printf("I = %d A \n",i)
printf("V = %d V",v)
```

Scilab code Exa 6.24 Example 24

```
//Chapter 6, Example 6.24
clc
//Variable Declaration
e = 18                          //supply voltage
r1 = 5000                       //resistance in ohm

//Calculation
i = e/r1                        //current in ampere
v = e                           //voltage
//Results
printf("I = %.1f mA \n",i*1000)
printf("V = %d V",v)
```

Scilab code Exa 6.25 Example 25

```
//Chapter 6, Example 6.25
clc
//Variable Declaration
v = 0                            //voltage
e = 6                            //supply voltage
r1 = 2                           //resistance in ohm

//Calculation
i = e/r1                         //current in ampere

//Results
```

```
printf("V = %d V \n",v)
printf("I = %d A",i)
```

Scilab code Exa 6.26 Example 26

```
//Chapter 6, Example 6.26
clc
//Variable Declaration
e = 20                                  //supply voltage
r = 1*10^6                              //resistance in ohm
rab1 = 11*10^6                          //load resistance in ohm
rm = 2*10^6
//Calculation
vab = e                                //voltage Vab
vab2 = (rab1*e)/(rab1+r)               //Vab by voltage divider rule
vab3 = (rm*e)/(rm+r)                   //Vab for the internal resistance

//Results
printf("a. Vab = %d V \n",vab)
printf("b. Vab = %.2f V \n",vab2)
printf("c. Vab = %.2f V \n",vab3)
```

Chapter 7

Series Parallel Networks

Scilab code Exa 7.1 Example 1

```
//Chapter 7, Example 7.1
clc
//Variable Declaration
rb = 12000                          //resistance in ohm
rc = 6000                           //resistance in ohm
e = 54                              //voltage
ra = 2000                           //resistance in ohm
//Calculation
rbc = (rb*rc)/(rb+rc)               //RB and RC are
    parallel
rt = ra+rbc                         //the equivalent
    resistance in ohm
is = e/rt                           //source current Is in
    ampere
ia = is
ib = (rc*is)/(rb+rc)                //applying current
    divider rule
ic = (rb*is)/(rb+rc)                //applying current
    divider rule
ic1 = is-ib                         //applying KCL

```

```

//Results
printf("Rbc = %d kohm \n",rbc/10^3)
printf("Rt = %d kohm \n",rt/10^3)
printf("Ia = %d mA \n",ia*1000)
printf("Ib = %.f mA \n",ib*1000)
printf("Ic = %.f mA \n",ic*1000)
printf("Ic = %.f mA \n",ic1*1000)
```

Scilab code Exa 7.2 Example 2

```
//Chapter 7, Example 7.2
clc
//Variable Declaration
r1 = 4                    //resistance in ohm
r2 = 4                    //resistance in ohm
r3 = 4                    //resistance in ohm
r4 = 0.5                  //resistance in ohm
r5 = 1.5                  //resistance in ohm
e = 10                    //voltage
is = 2                    //current through R1 in ampere

//Calculation
ra = r1                   //RA is R1 in ohms
rb = (r2*r3)/(r2+r3)      //RB is parallel combination of R2 and R3 in ohms
rc = r4+r5                //RC is equivalent resistor of R4 and R5 in ohms
rbc = (rb*rc)/(rb+rc)     //parallel combination of RB and RC in ohms
rt = ra+rbc                   //resistance in ohm
ia = is                   //same current in ampere
ib = ia/2                //current in ampere
ic = ib                 //current in ampere
```

```
ir2 = ib/2                        //current in ampere
ir3 = ir2                        //current in ampere
va = ia*ra                        //voltage
vb = ib*rb                       //voltage
vc = vb                      //voltage
v = e-va-vb                      //applying KVL

//Results
printf("Rt = %d ohm \n",rt)
printf("Is = %d A \n",is)
printf("Ia = %d A \n",ia)
printf("Ib = %d A \n",ib)
printf("Ir2 = Ir3 = %.1f A \n",ir2)
printf("Va = %d V \n",va)
printf("Vb = %d V \n",vb)
printf("Vc = %d V \n",vc)
```

Scilab code Exa 7.3 Example 3

```
//Chapter 7, Example 7.3
clc
//Variable Declaration
r1 = 9                         //resistance in ohms
r2 = 6                         //resistance in ohms
r3 = 6                         //resistance in ohms
r4 = r3                        //resistance in ohms
r5 = 3                       //resistance in ohms
r6 = 3                         //resistance in ohms
e = 16.8                       //source voltage

//Calculation
ra = (r1*r2)/(r1+r2)             //parallel R1 and R2
   in ohms
rb = r3 + ((r4*r5)/(r4+r5))     //resistance in ohms
rc = r6                          //resistance in ohms
```

```
rt = ra+((rb*rc)/(rb+rc))      //total resistance in
    ohms
is = e/rt                       //current in ampere
ia = is                        //current in ampere
ib = (rc*ia)/(rc+rb)           //applying the current
     divider rule
ic = ia-ib                     //by KCL, current in
    ampere
va = ia*ra                     //by ohm's law,
    voltage
vb = ib*rb                     //by ohm's law,
    voltage
i1 = (r2*ia)/(r2+r1)           //applying the current
     divider rule
i2 = ia-i1                        //current in ampere

//Results
printf("Rt = %.1f ohm \n",rt-0.1)
printf("Is = %d A \n",is+0.1)
printf("Ia = Is = %d A \n",ia+0.1)
printf("Ib = %d A \n",ib+0.3)
printf("Ic = %d A \n",ic)
printf("Va = %.1f A \n",va)
printf("Vb = %d A \n",vb)
printf("I1 = %.1f A \n",i1)
printf("I2 = %.1f A",i2)
```

Scilab code Exa 7.4 Example 4

```
//Chapter 7, Example 7.4
clc
//Variable Declaration
e = 12                     //voltage
r4 = 8                     //resistance in ohms
r2 = 3                     //resistance in ohms
```

```
r3 = 6                        //resistance in ohms
r1 = 4                        //resistance in ohms
rc = r1                       //resistance in ohms

//Calculation
i4 = e/r4                     //current through R4 in
   ampere
rd = (r2*r3)/(r2+r3)          //resistance in ohms
v2 = (rd*e)/(rd+rc)           //applying voltage
   divider rule

//Results
printf("I4 = %.1f A \n",i4)
printf("V2 = %d V",v2)
```

Scilab code Exa 7.5 Example 5

```
//Chapter 7, Example 7.5
clc
//Variable Declaration
r1 = 6                       //resistance in ohms
r2 = 6                       //resistance in ohms
r3 = 2                       //resistance in ohms
r4 = 8                       //resistance in ohms
r5 = 12                      //resistance in ohms
e = 24                       //voltage

//Calculation
r = (r1*r2)/(r1+r2)             //parallel combination
   of R1 and R2
ra = (r*r3)/(r+r3)              //resistance A
   combination in ohm
rb = (r4*r5)/(r4+r5)            //resistance B
   combination in ohm
rt = ra+rb                      //total resistance in
```

```
ohms
is = e/rt                          //current
v1 = is*ra                         //voltage across R'A'
v5 = is*rb                         //voltage across R'B'
i4 = v5/r4                          //current in ampere
   through R'B'
i2 = v1/r2                         //current in ampere
   through R2

//Results
printf("Rt = %d ohm \n",rt)
printf("Is = %d A \n",is)
printf("V1 = %.1f V \n",v1)
printf("V5 = %.1f V \n",v5)
printf("I4 = %.1f A \n",i4)
printf("I2 = %.1f A \n",i2)
```

Scilab code Exa 7.6 `Example 6`

```
//Chapter 7, Example 7.6
clc
//Variable Declaration
r1 = 5                              //resistance in ohms
r2 = 3                              //resistance in ohms
r3 = 6                              //resistance in ohms
r4 = 2                              //resistance in ohms
e = 12                              //voltage

//Calculation
v1 = (r1*e)/(r1+r2)                 //voltage divider rule
v3 = (r3*e)/(r3+r4)                 //voltage divider rule
vab = v3-v1                         //applying KVL, in
   voltage
i1 = v1/r1                          //by ohm's law, in
   amperes
```

```
i3 = v3/r3                        //by ohm's law, in
    amperes
is = i1+i3                        //applying KCL, in
    amperes

//Results
printf("a. V1 = %.1f V \n",v1)
printf("    V3 = %d V \n",v3)
printf("    Vab = %.1f V \n",vab)
printf("b. Is = %d A \n",is)
```

Scilab code Exa 7.7 Example 7

```
//Chapter 7, Example 7.7
clc
//Variable Declaration
e1 = 6                          //voltage
e2 = 18                         //voltage
r1 = 6                          //resistance in ohms
r2 = 5                          //resistance in ohms
r3 = 7                          //resistance in ohms
r4 = 6                          //resistance in ohms

//Calculation
v2 = -e1                            //in voltage
v1 = e2+e1                      //total voltage
i1 = v1/r1                      //by ohm's law, in amperes
i2 = e1/r4                      //by ohm's law, in amperes
i3 = e1/(r2+r3)                 //by ohm's law, in amperes
i = i1+i2+i3                    //applying KCL, in amperes

//Results
printf("V2 = %d V \n",v2)
printf("V1 = %d V \n",v1)
printf("I = %.1f A",i)
```

Scilab code Exa 7.8 Example 8

```
//Chapter 7, Example 7.8
clc
//Variable Declaration
r1 = 40*10^3                    //resistance in ohms
r2 = 4*10^3                     //resistance in ohms
rc = 10*10^3                    //resistance in ohms 'C'
re = 1*10^3                     //resistance in ohms 'E'
vcc = 22                        //power supply Vcc
vb = 2                          //base voltage
vbe = 0.7                       //base emitter voltage

//Calculation
v2 = vb
ve = v2-vbe                    //applying KVL to lower loop
ie = ve/re                     //by ohm's law, in ampere
v1 = vcc-v2                    //applying KVL to upper loop
ic = ie                         // in ampere
vc = vcc-(ie*rc)               //applying KVL , in voltaage
vbc = vb-vc                     //in voltaage
vce = vc-ve                     //in voltaage

//Results
printf("Ve = %.1f V \n",ve)
printf("Ie = %.1f mA \n",ie*1000)
printf("V1 = %d V \n",v1)
printf("Vbc = %d V \n",vbc)
printf("Vce = %.1f V",vce)
```

Scilab code Exa 7.9 Example 9

```
//Chapter 7, Example 7.9
clc
//Variable Declaration
r123 = 24*10^3                //combining resistance
    1,2,3 in ohms
r4 = 24*10^3                  //resistance in ohms
r5 = 12*10^3                  //resistance in ohms
r6 = 12*10^3                  //resistance in ohms
r7 = 9*10^3                   //resistance in ohms
r89 = 9*10^3                  //combining resistance
    8,9 in ohms
e = 72                        //voltage

//Calculation
r = (r123*r4)/(r123+r4)       //parallel resistance
    R123 and R4 in ohm
i5 = e/(r+r5)                 //by ohm's law, in
    ampere
rr = (r7*r89)/(r7+r89)        //parallel resistor R7
    and R89, in ohms
v7 = (rr*e)/(rr+r6)           //voltage divider rule,
     in volts
i6 = v7/rr                    //current through rr,
    in ampere
is = i5+i6                    //soucre current by KCL
    , in ampere

//Results
printf("I5 = %d mA \n",i5*1000)
printf("V7 = %.1f V \n",v7)
printf("I6 = %.2f mA \n",i6*1000)
printf("Is = %.2f mA \n",is*1000)
```

Scilab code Exa 7.10 Example 10

```
//Chapter 7, Example 7.10
clc
//Variable Declaration
e1 = 20                         //emf in voltage
e2 = 5                          //emf in voltage
e3 = 8                          //emf in voltage


//Calculation
v1 = e1-e3                      //voltage
v2 = e2-v1                      //voltage
v3 = e3-v2                      //voltage

//Results
printf("V1 = %d V \n",v1)
printf("V2 = %d V \n",v2)
printf("V3 = %d V \n",v3)
```

Scilab code Exa 7.11 Example 11

```
//Chapter 7, Example 7.11
clc
//Variable Declaration
e = 72                           //emf in voltage
is = 50*10^-3                    //current source in ampere
vr3 = 12                         //R3 voltage
vl1 = 60                         //RL1 voltage
vl2 = 20                         //RL2 voltage
ir1= 30*10^-3                    //R1 current in ampere
ir2 = 20*10^-3                   //R2 current in ampere
```

```
vr2 = 20                    //R2 voltage

//Calculation
r3 = vr3/is                    //resistance in ohms
vr1 = vl1-vl2                  //voltage
r1 = vr1/ir1                   //resistance in ohms
r2 = vr2/ir2                   //resistance in ohms

//Results
printf("R3 = %d ohm \n",r3)
printf("R1  = %.2f Kohm \n",r1/1000)
printf("R2 =  %d kohm",r2/1000)
```

Scilab code Exa 7.12 Example 12

```
//Chapter 7, Example 7.12
clc
//Variable Declaration
e = 120                     //voltage supply
r1 = 4000                   //resistance in ohms
r2 = 6000                   //resistance in ohms
r3 = 12000                  //resistance in ohms
r4 = 30000                  //resistance in ohms
r5 = 10000                  //resistance in ohms

//Calculation
r = (r1*r3)/(r1+r3)         //parallel combination R1
   and R3
rr = (r2*r4)/(r2+r4)        //parallel combination R2
   and R4
v1 = (r*e)/8000             //voltage
v2 = (rr*e)/8000            //voltage

//Results
printf("V1 = %d V \n",v1)
```

```
printf("V2 = %d V \n",v2)
```

Chapter 8

Methods of Analysis and Selected Topics dc

Scilab code Exa 8.1 Example 1

```
//Chapter 8, Example 8.1
clc
//Variable Declaration
i = 10*10^-3                    //source current
r = 20*10^3                     //resistance in ohms

//Calculation
i1 = i                          //current in ampere
v1 = i1*r                      //by ohm's law, in volts
vs = v1                        //in volts

//Results
printf("I1 = %d mA \n",i1*1000)
printf("Vs = %d V",vs)
```

Scilab code Exa 8.2 Example 2

```
//Chapter 8, Example 8.2
clc
//Variable Declaration
e = 12                              //voltage
r = 4                               //resistance in ohm
i = 7                               //current source in
    ampere

//Calculation
vs = e                               //voltage = Vs
i2 = e/r                            //by ohm's law, in
    amperes
i1 = i-i2                           //applying kirchoff's
    current law

//Results
printf("Vs = %d V \n",vs)
printf("I2 = %d A \n",i2)
printf("I1 = %d A",i1)
```

Scilab code Exa 8.3 Example 3

```
//Chapter 8, Example 8.3
clc
//Variable Declaration
r2 = 1                          //resistance in ohms
i = 6                           //current in amperes
r1 = 2                          //resistance in ohms

//Calculation
i1 = (r2*i)/(r2+r1)              //applying the current
    divider rule
v1 = i1*r1                       //by ohm's law, in volts
vs = v1+20                       //applying kirchoff's
    current law
```

```
//Results
printf("I1 = %d A \n",i1)
printf("Vs = %d V",vs)
```

Scilab code Exa 8.4 Example 4

```
//Chapter 8, Example 8.4
clc
//Variable Declaration
e = 6                              //voltage
rs = 2                             //resistance in ohm
rl = 4                             //load resistance in
    ohm
i = 3                              //source current in
    ampere
rl2 = 1000                         //replace load
    resistance in ohm

//Calculation
il = e/(rs+rl)                            //current in
    ampere
il1 = (rs*i)/(rs+rl)               //applying the current
     divider rule
il2 = e/(rs+rl2)                    //current in ampere
il3 = e/rl2                          //current in ampere

//Results
printf("a. IL = %d A \n",il)
printf("   IL = %d A \n",il1)
printf("b. IL = %.2f mA \n",il2*1000)
printf("c. IL = %d mA",il3*1000)
```

Scilab code Exa 8.5 Example 5

```
//Chapter 8, Example 8.5
clc
//Variable Declaration
rs = 3000                    //resistance in ohm
i = 9*10^-3                  //source current in ampere
rl  = 6000                   //load resistance in ohm
e = 27                       //voltage
rl2 = 10                     //replace load resistance in ohm

//Calculation
il = (rs*i)/(rs+rl)         //applying the current divider rule
il1 = e/(rs+rl)              //current in ampere
il2 = (rs*i)/(rs+rl2)       //applying the current divider rule
il3 = i                      // current in ampere

//Results
printf("a. IL = %d mA \n",il*1000)
printf("   IL = %d mA \n",il1*1000)
printf("b. IL = %.2f mA \n",il2*1000)
printf("c. IL = %d mA",il3*1000)
```

Scilab code Exa 8.6 Example 6

```
//Chapter 8, Example 8.6
clc
//Variable Declaration
i1 = 6                      //current in ampere figs. 8.11
```

```
i2 = 10                    //current in ampere figs.
    8.11
r1 = 3                     //resistance in ohms figs.
    8.11
r2 = 6                     //resistance in ohms figs.
    8.11
i3 = 7                     //current in ampere figs.
    8.12
i4 = 3                     //current in ampere figs.
    8.12
i5 = 4                     //current in ampere figs.
    8.12
r3 = 4                     //resistance in ohms figs.
    8.12


//Calculation
is = i2-i1                      //current in ampere
rs = (r1*r2)/(r1+r2)         //R1 and R2 are parallel
is1 = i3+i5-i4                    //current in ampere
rs1 = r3                     //in ohms

//Results
printf("Is = %d A \n",is)
printf("Rs = %d ohm \n",rs)
printf("Is = %d A \n",is1)
printf("Rs = %d ohm \n",rs1)
```

Scilab code Exa 8.7 Example 7

```
//Chapter 8, Example 8.7
clc
//Variable Declaration
i1 = 4                    //current in ampere
i2 = 6                    //current in ampere
```

```
r1 = 8                      //resistance in ohm
r2 = 24                     //resistance in ohm
rl = 14                     //load resistance in ohm

//Calculation
is = i1+i2                 //combining current in ampere
rs = (r1*r2)/(r1+r2)       //R1 and R2 are parallel
il = (rs*is)/(rs+rl)       //applying current divider
   rule

//Results
printf("Is = %d A \n",is)
printf("Rs = %d ohm \n",rs)
printf("IL = %d A",il)
```

Scilab code Exa 8.8 Example 8

```
//Chapter 8, Example 8.8
clc
//Variable Declaration
i1 = 4                        //current in ampere
r1 = 3                        //resistance in ohm
e2 = 5                        //voltage
r2 = 2                        //resistance in ohm

//Calculation
es = i1*r1                    //by ohm's law
rs = r1                       //equal
i2 = (es+e2)/(rs+r2)

//Results
printf("I2 = %.1f A ",i2)
```

Scilab code Exa 8.9 Example 9

```
//Chapter 8, Example 8.9
clc
//Variable Declaration

//three equations substituting the coefficients to
   variables
i11=2
i12=0
i13=4
i1=2

i21=0
i22=1
i23=4
i2=6

i31=1
i32=1
i33=-1
i3=0

//Calculation

//for I1
d=[i11 i12 i13;i21 i22 i23;i31 i32 i33]
di1=[i1 i12 i13;i2 i22 i23;i3 i32 i33]
ia1=det(di1)/det(d)

//for I2
di2=[i11 i1 i13 ; i21 i2 i23 ; i31 i3 i33]
ia2=det(di2)/det(d)

//for I3
di3=[i11 i12 i1 ; i21 i22 i2 ; i31 i32 i3]
ia3=det(di3)/det(d)

```

```
//Results
printf('I1 = %d A \n',ia1)
printf('I2 = %d A \n',ia2)
printf('I3 = %d A \n',ia3)
```

Chapter 9

Network Theorems

Scilab code Exa 9.1 Example 1

```
//Chapter 9, Example 9.1
clc
//Variable Declaration
rsc = 0
r1 = 6                    //resistance in ohms
e = 30                    //voltage
r1 = 6                    //resistance in ohms


//Calculation
i1 = (rsc)/(rsc+r1)              //applied the current
    divider rule, in amperes
i11 = e/r1                      //applying ohm's law
i = i1+i11                      //current in amperes

//Results
printf("I = %d A \n",i)
```

Scilab code Exa 9.2 Example 2

```
//Chapter 9, Example 9.2
clc
//Variable Declaration
r1 = 24                          //resistance in ohms
r2 = 12                          //resistance in ohms
r3 = 4                           //resistance in ohms
e1 = 54                          //voltage
e2 = 48                          //voltage

//Calculation
rt = r1+(r2*r3)/(r2+r3)           //total resistance in
    ohms
i = e1/rt                         //current in amperes in
     loop
i31 = (r2*i)/(r2+r3)              //using current divider
     rule, in amperes
rt2 = r3+(r1*r2)/(r1+r2)          //total resistance in
    ohms
i33 = e2/rt2                      //current in amperes in
     loop
i3 = i33-i31                      //total current in
    amperes through R3

//Results
printf("I3 = %.1f A",i3)
```

Scilab code Exa 9.3 Example 3

```
//Chapter 9, Example 9.3
clc
//Variable Declaration
r1 = 12                     //resistance in ohms
e = 36                      //voltage
r2 = 6                      //resistance in ohms
i = 9                       //source current in amperes
```

```
//Calculation
i21 = e/(r1+r2)             //current in amperes
i22 = (r1*i)/(r1+r2)        //applying current in
    amperes divider rule
i2 = i21+i22                //total current in amperes
     through the R2
p = i2*i2*r2                //power in watt to R2

//Results
printf("I2 = %d A \n",i2)
printf("P = %d W \n",p)
```

Scilab code Exa 9.4 Example 4

```
//Chapter 9, Example 9.4
clc
//Variable Declaration
r1 = 6000                        //resistance in ohms
r2 = 12000                       //resistance in ohms
i = 6*10^-3                      //current in amperes
e = 9                            //voltage

//Calculation
i21 = (r1*i)/(r1+r2)             //current in amperes
    divider rule
i22 = e/(r1+r2)                  //effect of 9-V
    voltage source
i2 = i21+i22                     //sum of the current
    in amperes

//Results
printf("I2 = %.1f mA",i2*1000)
```

Scilab code Exa 9.5 Example 5

```
//Chapter 9, Example 9.5
clc
//Variable Declaration
r1 = 2                              //resistance in ohms
r2 = 4                              //resistance in ohms
e1 = 12                             //voltage
e2 = 6                              //voltage
i = 3                               //current in amperes


//Calculation
i11 = e1/(r1+r2)                   //effect of the 12-V source
i21 = e2/(r1+r2)                   //effect of the 6-V source
i23 = (r2*i)/(r1+r2)               //applying the current divider rule, in amperes
i1 = i21+i23-i11                           //current in amperes

//Results
printf("I1 = %d A \n",i1)
```

Scilab code Exa 9.6 Example 6

```
//Chapter 9, Example 9.6
clc
//Variable Declaration
e1= 9                      //voltage
r1 = 3                     //resistance in ohms
```

```
r2 = 6                      //resistance in ohms
rl1 =2                      //resistance in ohms
rl2 = 10                     //resistance in ohms
rl3 = 100                     //resistance in ohms

//Calculation
rth = (r1*r2)/(r1+r2)              //R1 and R2 are parallel
eth = (r2*e1)/(r2+r1)              //applying voltage divider rule
il1 = eth/(rth+rl1)                //RL = 20 ohm
il2 = eth/(rth+rl2)                //RL = 10 ohm
il3 = eth/(rth+rl3)                //RL = 100 ohm

//Results
printf("Rth = %d ohm \n",rth)
printf("Eth = %d V \n",eth)
printf("IL = %.1f A \n",il1)
printf("IL = %.1f A \n",il2)
printf("IL = %.3f A \n",il3)
```

Scilab code Exa 9.7 `Example 7`

```
//Chapter 9, Example 9.7
clc
//Variable Declaration
r1 = 4                      //resistance in ohms
r2 = 2                      //resistance in ohms
i1 = 12                     //current in amperes source
i2 = 0

//Calculation
rth = r1+r2                 //total resistance in ohms
eth = i1*r1                 //voltage

```

```
//Results
printf("Rth = %d ohm \n",rth)
printf("Eth = %d V",eth)
```

Scilab code Exa 9.8 Example 8

```
//Chapter 9, Example 9.8
clc
//Variable Declaration
r1 = 6                    //resistance in ohms
r2 = 4                    //resistance in ohms
e1 = 8                    //voltage

//Calculation
rth = (r1*r2)/(r1+r2)     //R1 and R2 are parallel
eth = (r1*e1)/(r1+r2)     //applying voltage divider
    rule

//Results
printf("Rth = %.1f ohm \n",rth)
printf("Eth = %.1f V",eth)
```

Scilab code Exa 9.9 Example 9

```
//Chapter 9, Example 9.9
clc
//Variable Declaration
r1 = 6                   //resistance in ohms
r2 = 4                   //resistance in ohms
r3 = 3                   //resistance in ohms
r4 = 12                  //resistance in ohms
e = 72                   //voltage
r22 = 12                  //resistance in ohms
```

```


//Calculation
rth = ((r1*r3)/(r1+r3))+((r2*r4)/(r2+r4))        //total resistance in ohms
v1 = (r1*e)/(r1+r3)                              //voltage divider rule R1
v2 = (r22*e)/(r2+r4)                             //voltage divider rule R2
eth = v2-v1                                      //applying kirchoff's voltage law

//Results
printf("Rth = %d ohm \n",rth)
printf("Eth = %d V",eth)
```

Scilab code Exa 9.10 Example 10

```
//Chapter 9, Example 9.10
clc
//Variable Declaration
r1 = 0.8*10^3                       //resistance in ohms
r2 = 4000                           //resistance in ohms
r3 = 6000                           //resistance in ohms
r4 = 1.4*10^3                       //resistance in ohms
e1= 6                               //voltage
e2 = 10                             //voltage

//Calculation
r12 = (r1*r2)/(r1+r2)               //R1 and R2 are parallel
rth = r4+((r12*r3)/(r12+r3))        //total resistance in ohm
r23 = (r2*r3)/(r2+r3)               //R2 and R3 are parallel
```

```
v31 = (r23*e1)/(r23+r1)                //applying
    voltage divider rule R23
eth1 = v31                           //voltage
v32 = (r12*e2)/(r12+r2)                //applying
    voltage divider rule R12
eth2 = v32                           //voltage
eth = eth1-eth2                          //voltage

//Results
printf("Rth = %d kohm \n",rth/10^3)
printf("Eth = %d V",eth)
```

Scilab code Exa 9.11 Example 11

```
//Chapter 9, Example 9.11
clc
//Variable Declaration
r1 = 3                         //resistance in ohms
r2 = 6                         //resistance in ohms
e = 9                          //voltage

//Calculation
rn = (r1*r2)/(r1+r2)          //R1 and R2 are
    parallel
in = e/r1                     //current in amperes

//Results
printf("Rn = %d ohm \n",rn)
printf("In = %d A",in)
```

Scilab code Exa 9.12 Example 12

```
//Chapter 9, Example 9.12
```

```
clc
//Variable Declaration
r1 = 5                          //resistance in ohms
r2 = 4                          //resistance in ohms
i = 10                          //current in amperes

//Calculation
rn = r1+r2                     //total resistance in ohms
in = (r1*i)/(r1+r2)            //applying the current in amperes divider rule

//Results
printf("Rn = %d ohm \n",rn)
printf("In = %.3f A",in)
```

Scilab code Exa 9.13 Example 13

```
//Chapter 9, Example 9.13
clc
//Variable Declaration
r1 = 4                            //resistance in ohms
r2 = 6                            //resistance in ohms
r3 = 9                            //resistance in ohms
r4 = 10                           //resistance in ohms
e1 = 7                            //voltage
e2 = 12                           //voltage
i = 8                             //current source in amperes

//Calculation
rn = (r1*r2)/(r1+r2)              //R1 and R2 are parallel
in1= e1/r1                        //current in amperes
in2 = i                          //short circuited
in = in2-in1                     //in amperes
```

```

//Results
printf("Rn = %.1f ohm \n",rn)
printf("In = %.2f A \n",in)
```

Scilab code Exa 9.14 Example 14

```
//Chapter 9, Example 9.14
clc
//Variable Declaration
rint1 = 2.5                    //resistance in ohms of dc generator
rint2 = 0.5                    //resistance in ohms of battery
rint3 = 40                     //resistance in ohms of laboratory
n = 0.75                       //efficiency in decimal form

//Calculation
rl1 = rint1                    //for the dc generator, in ohms
rl2 = rint2                    //for the battery, in ohms
rl3 = rint3                    //for the laboratory supply, in ohms
Rl1 = (n*rl1)/(1-n)            //RL dc gen. for efficiency, in ohms
Rl2 = (n*rl2)/(1-n)            //RL battery for efficiency, in ohms
Rl3 = (n*rl3)/(1-n)            //RL laboratory for efficiency, in ohms

//Results
printf("a. RL = %.1f ohm \n",rl1)
```

```
printf("    RL = %.1f ohm \n",rl2)
printf("    RL = %d ohm \n",rl3)
printf("b. RL = %.1f ohm \n",Rl1)
printf("    RL = %.1f ohm \n",Rl2)
printf("    RL = %d ohm \n",Rl3)
```

Scilab code Exa 9.15 Example 15

```
//Chapter 9, Example 9.15
clc
//Variable Declaration
rs = 40*10^3                    //resistance in ohms
in  =10*10^-3                   //current in amperes
   source

//Calculation
rl = rs                        //load resistance in ohms
pl = (in*in*rs)/4              //maximum power in watt

//Results
printf("RL = %d kohm \n",rl/10^3)
printf("PLmax = %d W \n",pl)
```

Scilab code Exa 9.16 Example 16

```
//Chapter 9, Example 9.16
clc
//Variable Declaration
r1 = 6                            //resistance in ohms
r2 = 3                            //resistance in ohms
r3 = 8                            //resistance in ohms
e = 12                            //voltage

```

```
//Calculation
rth = r3+(r1*r2)/(r1+r2)          //total resistance in ohms
eth = (r2*e)/(r2+r1)              //voltage divider
PLmax = (eth*eth)/(4*rth)         //maximum power in watt

//Results
printf("Rth = %d ohm \n",rth)
printf("Eth = %d V \n",eth)
printf("PLmax = %.1f W",PLmax)
```

Scilab code Exa 9.17 Example 17

```
//Chapter 9, Example 9.17
clc
//Variable Declaration
r1 = 3                        //resistance in ohms
r2 = 10                       //resistance in ohms
r3 = 2                        //resistance in ohms
e = 68                        //voltage
i = 6                         //current in amperes

//Calculation
rth = r1+r2+r3                 //total resistance in ohms
rl = rth                       //load resistance in ohms
v2 = i*r2                      //voltage
eth = v2 + e                   //applying kirchoff's voltage law
plmax = (eth*eth)/(4*rth)    //maximum power in watt

//Results
printf("Rth =  %d ohm \n",rth)
```

```
printf("PLmax = %.2f W",plmax)
```

Scilab code Exa 9.18 Example 18

```
//Chapter 9, Example 9.18
clc
//Variable Declaration
r1 = 5                  //resistance in ohms
r2 = 4                  //resistance in ohms
r3 = 2                  //resistance in ohms
e1 = 10                 //voltage
e2= 16                  //voltage
e3 = 8                  //voltage
rl = 3                  //load resistance in ohms

//Calculation
Eeq = ((e1/r1)-(e2/r2)+(e3/r3))/((1/r1)+(1/r2)+(1/r3
   ))
Req = 1/((1/r1)+(1/r2)+(1/r3))                        //
   resistance in ohms
il = Eeq/(Req+rl)            //load current in
   amperes
vl = il*rl                   //load voltage

//Results
printf("Eeq = %.3f V \n",Eeq)
printf("Req = %.3f ohm \n",Req)
printf("IL = %.3f V \n",il)
printf("VL = %.3f V ",vl)
```

Scilab code Exa 9.19 Example 19

```
//Chapter 9, Example 9.19
```

```
clc
//Variable Declaration
i1 = 5                          //source current in
    amperes
i2 = 5/3                        //source current in
    amperes
g1 = 1                          //conductance
g2 = 1/6                        //conductance
r3 = 2                          //resistance in ohms
//Calculation
it = i1+i2                     //total current in
    amperes
gt = g1+g2                     //total conductance in
    mho
Eeq = it/gt                    //voltage
Req = 1/gt                     //equivalent resistance
i2ohm = Eeq/(Req+r3)           //current in amperes of
    2-ohm
Eeq1 = Eeq
Req1 = Req

//Results
printf("a. Eeq = %f V \n",Eeq)        //converted to
    decimals
printf("    Req = %f ohm \n",Req)        //converted to
     decimals
printf("    I2ohm = %d A \n",i2ohm)
printf("b. Eeq = %f V \n",Eeq1)       //converted to
    decimals
printf("    Req = %f ohm ",Req1)       //converted to
    decimals
```

Chapter 10

Capacitors

Scilab code Exa 10.1 `Example 1`

```
//Chapter 10, Example 10.1
clc
//Variable Declaration
c11=5*10^-6                       //capacitance in farad
c12=0.1*10^-6                      //capacitance in farad
c13=20*10^-6                      //capacitance in farad
c14=1000*10^-12                      //capacitance in farad

//Calculation
c1 = 3*(c11)                      //capacitance in farad
c2 = 0.5*(c12)                       //capacitance in farad
c3 = 2.5*(c13)                      //capacitance in farad
c4 = 5*4*(c14)/(1/8)                 //capacitance in farad
```

```
//Results
printf("a. C = %.f uF \n",c1*10^6)
printf("b. C = %.2f uF \n",c2*10^6)
printf("c. C = %.f uF \n",c3*10^6)
printf("d. C = %.2f uF \n",c4*10^6)
```

Scilab code Exa 10.2 Example 2

```
//Chapter 10, Example 10.2
clc
//Variable Declaration
a = 0.01                    //area of plate in meter square
d = 1.5*10^-3               //distance in meter
eo = 8.85*10^-12            //permittivity
v = 450                     //voltage

//Calculation
c = eo*a/d                 //capacitance in farad
E = v/d                    //electric field in V/m
q = c*v                    //charge on each plate in Coulomb

//Results
printf("a. C = %.f pF \n",c*10^12)
printf("b. E = %d x 10^3 V/m \n",E/1000)
printf("c. Q = %.2f nC \n",q*10^9)
```

Scilab code Exa 10.3 Example 3

```
//Chapter 10, Example 10.3
clc
```

```
//Variable Declaration
v = 450                        //voltage
d = 1.5*10^3                   //distance in meter
a = 0.01                       //area of plate in meter
    square
er = 5                         //permittivity
eo = 8.855*10^-12              //permittivity
C = 59*10^-12                  //capacitance in farad

//Calculation
E = v/d                       //electric field in V/m
q = er*eo*E*a                 //charge on each plate in
    coulomb
c = er*C                      //capacitance in farad

//Results
printf("E = %.1f V/m \n",E)      //wrong answer in
    textbook
printf("Q = %f nC \n",q*10^9)    //wrong answer in
    textbook
printf("C = %d pF \n",c*10^12)
```

Scilab code Exa 10.4 Example 10

```
//Chapter 10, Example 10.4
clc
//Variable Declaration
c = 0.2*10^-6                  //capacitance in farad
a = 0.3                        //area of plate in meter
    square
er = 6                         //permittivity
di = 200                       //dielectric strength

//Calculation
d = (8.85*er*a)/(10^12*c)           //find distance
```

```
   in micrometer
d1 = d*(39.371)*1000                 //converting
   micrometer to mils
d2 = di*d1

//Results
printf("d = %.2f um \n",d*10^6)
printf("V = %.2f V \n",d2)
```

Scilab code Exa 10.5 Example 5

```
 //Chapter 10, Example 10.5
clc
//Variable Declaration
r = 8000                    //resistance in ohms
e = 40                      //voltage
c = 4*10^-6                 //capacitor in farad
m = -1/(32*10^-3)

//Calculation
t = r*c                    //time constant
vc = e                     //in volts
ic = e/r                     //in ampere
vr =  e                    //in volts
b = 5*t                   //time constant

//for plotting graph
x1 = 0 : t : 1
y1=vc*(1- exp(-x1/t))
i1=ic*(1- exp(-x1/t))
vr1=vc*(exp(-x1/t))

subplot(221)
plot(x1, y1)
xlabel("t");
```

```
ylabel("vc (V)");
subplot(222)
plot(x1, i1)
xlabel("t");
ylabel("ic (mA)");
subplot(223)
plot(x1, vr1)
xlabel("t");
ylabel("vR (V)");

//Results
printf("a. t = %d ms \n",t*10^3)
printf("    Vc = %d*(1-e^-t/(32*10^-3)) \n",vc)
printf("    ic = (%d x 10^-3)*(e^-t/(32*10^-3)) \n",
   ic*1000)
printf("    vr = %d*(e^-t/(32*10^-3)) \n",vr)
printf("b = %d ms",b*10^3)
```

Scilab code Exa 10.6 Example 6

```
//Chapter 10, Example 10.6
clc
//Variable Declaration
t = 32*10^-3                    //time in seconds
e = 40                          //voltage
r = 8000                        //resistance in ohms

//Calculation
vc = e                            //voltage
ic = -e/r                         //in ampere
vr = -e                         //voltage

//for plotting graph
x1 = 0 : t : 1
y1=vc*(exp(-x1/t))
```

```
i1=ic*(exp(-x1/t))
vr1=-vc*(exp(-x1/t))

subplot(221)
plot(x1, y1)
xlabel("t");
ylabel("vc (V)");
subplot(222)
plot(x1, i1)
xlabel("t");
ylabel("ic (mA)");
subplot(223)
plot(x1, vr1)
xlabel("t");
ylabel("vR (V)");

//Results
printf("a. t = %d ms \n",t*10^3)
printf("    Vc = %d*(e^-t/(32*10^-3)) \n",vc)
printf("    ic = (%d x 10^-3)*(e^-t/(32*10^-3)) \n",
   ic*1000)
printf("    vr = %d*(e^-t/(32*10^-3)) \n",vr)
```

Scilab code Exa 10.7 Example 7

```
//Chapter 10, Example 10.7
clc
//Variable Declaration
r1 = 100*10^3                //resistance in ohms
r2 = 200*10^3                //resistance in ohms
e = 10                       //voltage
c = 0.05*10^-6               //capacitance in farad
t3=0.045

//Calculation
```

```
vc = e                          //voltage across the
   capacitor
t = r1*c                        //time constant
ic = e/r1                       //current across the
   capacitor
t1 = r2*c                       //time constant
ic2 = -e/r2                      //in ampere


//for plotting graph
x1 = 0 : 0.01 : t3
x2 =  t3 : 0.01 : 1
y1=vc*(1-exp(-x1/t))
y2=vc*(exp(-x2/t1))
i1=ic*(exp(-x1/t))
i2=ic2*(exp(-x2/t1))

subplot(221)
plot(x1, y1)
xlabel("t")
ylabel("vc (V)")
xtitle('Vc Charging')
subplot(222)
plot(x2, y2)
xlabel("t")
ylabel("vc (V)")
xtitle('Vc Discharging')
subplot(223)
plot(x1, i1)
xlabel("t")
ylabel("ic (mA)")
xtitle('ic Charging')
subplot(224)
plot(x2, i2)
xlabel("t")
ylabel("ic (mA)")
xtitle('ic Discharging')

```

```
//Results
printf("a. vc = %d(1-e^-t/(5*10^-3)) \n",vc)
printf("b. ic = (%.1f x 10^-3)(e^-t/(5*10^-3)) \n",ic*10^3)
printf("c. vc = %d V \n",vc)
printf("   ic = %d A \n",ic)
printf("d. vc = %d(e^-t/(10*10^-3)) \n",vc)
printf("   ic = (%.2f x 10^-3)(e^-t/(10*10^-3)) \n",ic2*10^3)
```

Scilab code Exa 10.8 Example 8

```
//Chapter 10, Example 10.8
clc
//Variable Declaration
r1 = 5000                  //resistance in ohms
e = 20                     //voltage
r2 = 1000                  //resistance in ohms
r3 = 3000                  //resistance in ohms
c = 10*10^-6               //capacitance in farad


//Calculation
vc = e                     //voltage across capacitor
t1 = (r1+r3)*c             //time constant
t2 = (r2+r3)*c             //time constant
ic = e/(r1+r3)             //current across capacitor
vc1 = 0.632*e              //Vc when switch at position2
ic1 = -vc1/(r2+r3)         //Ic when switch at position2


//for plotting graph
x1 = 0 : 0.001 : t1
```

```
x2 = t1 : 0.001 : 0.4
y1=vc*(1- exp(-x1/t1))
y2=vc1*(exp(-x2/t2))
i1=ic*(exp(-x1/t1))
i2=ic1*(exp(-x2/t2))

subplot(221)
plot(x1, y1)
plot(x2, y2)
xlabel("t")
ylabel("vc (V)")

subplot(222)
plot(x1, i1)
plot(x2, i2)
xlabel("t")
ylabel("ic (mA)")

//Results
printf("a. vc = %d(1-e^-t/(80*10^-3)) \n",vc)
printf("b. ic = (%.1f x 10^-3)(e^-t/(80*10^-3)) \n",
    ic*10^3)
printf("c. vc = %.2f(e^-t/(40*10^-3)) \n",vc1)
printf("ic = %.2f x 10^-3(e^-t/(40*10^-3)) \n",ic1
    *10^3)
```

Scilab code Exa 10.9 Example 9

```
//Chapter 10, Example 10.9
clc
//Variable Declaration
r1 = 2.2*10^3                //resistance in ohms
r2 = 1.2*10^3                //resistance in ohms
c = 3.3*10^-6                //capacitance in farad
e = 24                       //voltage
```

```
vf = 24                          //voltage
vi = 4                           //voltage across
    capapcitor

//Calculation
t = (r1+r2)*c                    //time constant
vc = (vi-vf)                     //Vc when switch close
im = (e-(vf+vc))/(r1+r2)          //current in amperes
ic = im                          //current during the
    transient peroid


//for plotting graph
x1 = 0 : 0.001 : (5*t)
y1=vf-(-vc*exp(-x1/t))
i1=(im*exp(-x1/t))
subplot(221)
plot(x1, y1)
xlabel("t");
ylabel("vc (V)");
subplot(222)
plot(x1, i1)
xlabel("t");
ylabel("ic (mA)");


//Results
printf("a. vc = %dV %dV(e^-t/(11.22*10^-3ms)) \n",vf
    ,vc)
printf("b. ic = %.2fmA(e^-t/(11.22*10^-3ms)) \n",ic
    *10^3)
```

Scilab code Exa 10.10 Example 10

```
//Chapter 10, Example 10.10
```

```
clc
//Variable Declaration
r1 = 60*10^3                //resistance in ohms
r2 = 30*10^3                //resistance in ohms
r3 = 10*10^3                //resistance in ohms
vi = 5.44                   //voltage
vf = 0                   //voltage
c=0.2*10**-6                   //capacitance in farad
e=21                        //emf in voltage

//Calculation
rth = ((r1*r2)/(r1+r2))+r3   //applying tevenin's theorem
eth = r2*e/(r2+r1)              //in volts
t=rth*c                         //time constant
vc1 = eth                      //vc at switch position1
ic = eth/r2                    //ic at switch position1
vc = vf +(vi-vf)              //vc at switch position2
ii = vc/r3                      //current in amperes
t1=r3*c                             //time constant
ic1 = -ii                     //ic at switch position2

//for plotting graph
x1 = 0 : 0.001 : 0.035
x2 =  0 : 0.0005 : 0.05
y1=eth*(1-exp(-x1/t))
y2=vc*(exp(-x2/t1))
i1=ic*(exp(-x1/t))
i2=ic1*(exp(-x2/t1))

subplot(221)
plot2d(x1, y1)
xlabel("t")
ylabel("vc (V)")
xtitle('Vc Charging')
subplot(222)
```

```
plot2d(x2, y2)
xlabel("t")
ylabel("vc (V)")
xtitle('Vc Discharging')
subplot(223)
plot2d(x1, i1)
xlabel("t")
ylabel("ic (mA)")
xtitle('ic Charging')
subplot(224)
plot2d(x2, i2)
xlabel("t")
ylabel("ic (mA)")
xtitle('ic Discharging')

//Results
printf("a. vc = %d(1-e^-t/6ms) \n",vc1)
printf("    ic = (%.3f x 10^-3)(e^-t/6ms) \n",ic
   *10^3)
printf("b. vc = %.2f(e^-t/2ms) \n",vc)
printf("    ic = (%.2f x 10^-3)(e^-t/2ms) \n",ic1
   *10^3)
```

Scilab code Exa 10.11 `Example 11`

```
//Chapter 10, Example 10.11
clc
//Variable Declaration
r2 = 5*10^3                    //resistance in ohms
r1 = 7*10^3                    //resistance in ohms
r3 = 18*10^3                   //resistance in ohms
r4 = 2*10^3                    //resistance in ohms
e = 120                        //voltage
c = 40*10^-6                   //capacitance in farad
vi = 40                        //voltage across
```

```
capacitor

//Calculation
eth = (r3*e)/(r3+r1+r4)                    //in volts
rth = ((r2+r3)*(r2+r3))/((r2+r3)+(r2+r3))    //resistance in ohms
t = rth*c                                  //time constant
vc = vi                          //Vc after closing switch

//Results
printf("vc = %dV - %dV(e^-t/0.44s) \n",eth,vc)
```

Scilab code Exa 10.12 Example 12

```
//Chapter 10, Example 10.12
clc
//Variable Declaration
r1 = 6                    //resistance in ohms
r2 = 10                   //resistance in ohms
c = 500*10^-6             //capacitance in farad
i = 20*10^-3              //source current in ampere

//Calculation
rth = r1+r2                  //resistance in ohms
eth = i*r1                  //voltage
t = rth*c                   //time constant
vc = eth                  //voltage

//Results
printf("vc = %.2f(1-e^-t/8ms) \n",vc)
```

Scilab code Exa 10.13 Example 13

```
//Chapter 10, Example 10.13
clc
//Variable Declaration
delv = 4                          //change in voltage
delt = 2*10^-3                    //change in time (sec)
c = 2*10^-6                       //capacitor in farad
delv1 = 0                         //change in voltage
delt1 = 6*10^-3                   //change in time (sec)

//Calculation
ic = c*delv/(delt)                  //current in amperes
ic1 = c*delv1/delt                  //current in amperes
ic2 = -c*delv/delt1                  //current in amperes

x1 = 0: 0.0009 : 0.002
x3 = 0.005 : 0.0009: 0.011

ic11=[ic ic ic]
ic12=[ic2 ic2 ic2 ic2 ic2 ic2 ic2]
plot(x1, ic11,'LineWidth', 2)
plot(x3, ic12)
a=gca(); // Handle on axes entity
a.x_location = "origin";
xlabel("t")
ylabel("ic (mA)")
xtitle('Average Current')



//Results
printf("a. ic = %d mA \n",ic*10^3)
printf("b. ic = %d \n",ic1)
```

```
printf("c. ic = %.2f mA \n",ic2*10^3)
```

Scilab code Exa 10.14 Example 14

```
//Chapter 10, Example 10.14
clc
//Variable Declaration
c1 = 200*10^-6                      //capacitance in farad
c2 = 50*10^-6                       //capacitance in farad
c3 = 10*10^-6                       //capacitance in farad
e = 60                              //voltage

//Calculation
ctt = (1/c1)+(1/c2)+(1/c3)
ct = 1/ctt                          //total capacitance in farad
qt = ct*e                           //charge on each plate
v1 = qt/c1                          //voltage across each capacitor
v2 = qt/c2                          //voltage across each capacitor
v3 = qt/c3                          //voltage across each capacitor
e = v1+v2+v3                          //voltage

//Results
printf("a. Ct = %d uF \n",ct*10^6)
printf("b. Qt = %d uF \n",qt*10^6)
printf("c. V1 = %.1f V \n",v1)
printf("   V2 = %.1f V \n",v2)
printf("   V3 = %.1f V \n",v3)
printf("   E = %d V \n",e)
```

Scilab code Exa 10.15 Example 15

```
//Chapter 10, Example 10.15
clc
//Variable Declaration
c1 = 800*10^-6              //capacitance in farad
c2 = 60*10^-6               //capacitance in farad
c3 = 1200*10^-6             //capacitance in farad
e = 48                      //voltage

//Calculation
ct = c1+c2+c3               //total capacitance in
    farad
q1 = c1*e                   //charge on each plate in
    coulombs
q2 = c2*e                   //charge on each plate in
    coulombs
q3 = c3*e                   //charge on each plate in
    coulombs
qt = q1+q2+q3               //total charge in coulombs

//Results
printf("a. Ct = %d uF \n",ct*10^6)
printf("b. Q1 = %.1f mC \n",q1*10^3)
printf("   Q2 = %.2f mC \n",q2*10^3)
printf("   Q3 = %.1f mC \n",q3*10^3)
printf("c. Qt = %.2f mC \n",qt*10^3)
```

Scilab code Exa 10.16 Example 16

```
//Chapter 10, Example 10.16
clc
```

```
//Variable Declaration
c1 = 3*10^-6                    //capacitance in farad
c2 = 4*10^-6                    //capacitance in farad
c3 = 2*10^-6                    //capacitance in farad
e = 120                         //voltage

//Calculation
ct1 = c2+c3                   //capacitance in farad
ct = (c1*ct1)/(c1+ct1)                  //capacitance in farad
qt = ct*e                      //charge in coulombs
q1 = qt                        //charge of C1 in coulombs
v1 = q1/c1                      //voltage1 of C1 in coulombs
qt1 = q1                    //charge in coulombs
vt1 = qt1/ct1                   //voltage total
q2 = c2*vt1                    //charge of C2
q3 = c3*vt1                    //charge of C3

//Results
printf("Qt = %d uC \n",qt*10^6)
printf("Q1 = %d uC \n",q1*10^6)
printf("V1 = %d V \n",v1)
printf("Vt = %d uC \n",vt1)
printf("Q2 = %d uC \n",q2*10^6)
printf("Q3 = %d uC \n",q3*10^6)
```

Scilab code Exa 10.17 Example 17

```
//Chapter 10, Example 10.17
clc
//Variable Declaration
r1 = 4                     //resistance in ohms
r2 = 8                     //resistance in ohms
e = 24                     //voltage
```

```
c1 = 20*10^-6          //capacitance in farad


//Calculation
vc = (r2*e)/(r1+r2)    //voltage across capacitor
q1 = c1*vc             //charge in coulombs

//Results
printf("Vc = %d V \n",vc)
printf("Q1 = %d uC \n",q1*10^6)
```

Scilab code Exa 10.18 `Example 18`

```
//Chapter 10, Example 10.18
clc
//Variable Declaration
r1=  2                       //resistance in ohms
r2 = 7                       //resistance in ohms
r3 =8                        //resistance in ohms
e = 72                       //voltage
c1 = 2*10^-6                 //capacitance in farad
c2 = 3*10^-6                 //capacitance in farad

//Calculation
vc2 = (r2*e)/(r1+r2)         //voltage across C2
vc1 = (r1*e)/(r1+r2)         //voltage across C1
q1 = c1*vc1                  //charge of C1
q2 = c2*vc2                  //charge of C2

//Results
printf("Vc2 = %d V \n",vc2)
printf("Vc1 = %d V \n",vc1)
printf("Q1 = %d uC \n",q1*10^6)
printf("Q2 = %d uC \n",q2*10^6)
```

Scilab code Exa 10.19 `Example 19`

```
//Chapter 10, Example 10.19
clc
//Variable Declaration
c1 = 2*10^-6                    //capacitance in farad
v1 = 16                         //voltage
c2 = 3*10^-6                    //capacitance in farad
v2 = 56                         //voltage

//Calculation
wc = c1*v1*v1/2                //energy of charge in joule
wc1 = c2*v2*v2/2               //energy of charge in joule

//Results
printf("Wc = %d uJ \n",wc*10^6)
printf("Wc = %d uJ \n",wc1*10^6)
```

Chapter 11

Magnetic Circuits

Scilab code Exa 11.1 Example 1

```
//Chapter 11, Example 11.1
clc
//Variable Declaration
o = 6*10^-5                        //flux in webers
a = 1.2*10^-3                      //area in square meters

//Calculation
b = o/a                           //Tesla in T


//Results
printf("B = %d x 10^-2 T \n",b*10^2)
```

Scilab code Exa 11.2 Example 2

```
//Chapter 11, Example 11.2
clc
//Variable Declaration
```

```
a = 0.25                              //area
b = 1.2                              //tesla in T
//Calculation
A = a*(1/39.37)*(1/39.37)            //area in square
    meter
o = b*A                              //flux in webers

//Results
printf("O = %.3f x 10^-4 Wb ",o*10^4)
```

Scilab code Exa 11.3 Example 3

```
//Chapter 11, Example 11.3
clc
//Variable Declaration
o = 4*10^-4                      //magnetic flux in
    webers
a = 2*10^-3                      //area in square meter
h = 170                          //magnetizing force in
    At/m
l = 0.16                         //mean length in meter
n = 400                          //number of turns
uo = 12.56*10^-7                 //magnetic permeability
    (uo)

//Calculation
b = o/a                           //the flux density
I = h*l/n                        //applying ampere's
    circuital law in mA
u = b/h                          //the permeability of
    the material in Wb/A.m
ur = u/uo                        //relative permeability
    (ur)
//Results
printf("a. I = %d mA \n",I*10^3)
```

```
printf("b   U = %f x 10^-3 Wb/A.m \n",u*10^3)
printf("    Ur = %.2f \n",ur)
```

Scilab code Exa 11.4 Example 4

```
//Chapter 11, Example 11.4
clc
//Variable Declaration
o = 3.5*10-4                        //magnetic flux in Wb
a = 6.452*10^-4                     //area in square meter
hefab = 70                          //magnetizing force sheet steel in At/m
hbcde = 1600                        //magnetizing force cast iron in At/m
n = 50                              //number of turns

//Calculation
lefab = 4+4+4                       //length of the material in inch
lbcde = 0.5+4+0.5                   //length of the material in inch
lbcde1 = lbcde*(1/39.37)             //in meters
lefab1 = lefab*(1/39.37)              //in meters
b = o/a                              //flux density in T
q = (hefab*lefab1)+(hbcde*lbcde1)       //in At
i = q/n                              //applying ampere's circuital law

//Results
printf("I = %.2f A \n",i)
```

Scilab code Exa 11.5 Example 5

```
//Chapter 11, Example 11.5
clc
//Variable Declaration
o = 1.5*10^-5                    //magnetic flux in  Wb
a = 0.15*10^-3                   //area in square meter
habcda = 20                       //in At/m
i1 = 2                           //primary current in
   ampere
n1 = 60                          //number of turn in 1
   st winding
n2 = 30                          //number of turn in 2
   nd winding
labcda  = 0.16                   //in meter

//Calculation
b = o/a                                   //flux
   density
i2 = ((n1*i1)-(habcda*labcda))/n2         //secondary
   current

//Results
printf("I2 = %.2f A \n",i2)
```

Scilab code Exa 11.6 Example 6

```
//Chapter 11, Example 11.6
clc
//Variable Declaration
o = 0.75*10^-4                     //magnetic flux in
   Wb
a = 1.5*10^-4                      //area in square
   meter
hcore = 280                        //H(cast steel) in
   At/m
hg = 3.98*10^5                     //in At/m
```

```
icore = 100*10^-3                //in meter
ig = 2*10^-3                     //in meter
n = 200                          //number of turns

//Calculation
b = o/a                           //flux density
i = ((hcore*icore)+(hg*ig))/n     //in ampere

//Results
printf("I = %.2f A \n",i)
```

Scilab code Exa 11.7 `Example 7`

```
//Chapter 11, Example 11.7
clc
//Variable Declaration
o2 = 1.5*10^-4                 //magnetic flux in Wb
a = 6*10^-4                   //area in square meter
hbcde = 40                       //in At/m
ibcde = 0.2                      //in meter
ibe = 0.05                       //in meter
b1 = 0.97                     //tesla in T
hefab = 400                      //in At/m
iefab = 0.2                      //in meter
n = 50                        //number of turns
h1 = 40                        //in At/m
uo = 12.57*10^-7               //megnetic permeability
h2 = 160                         // in At/m
b3  = 0.25                    //tesla in T

//Calculation
b2 = o2/a                        //flux density in T
hbe = (hbcde*ibcde)/ibe          //in At/m
o1 = b1*a                        //flux in Wb
ot = o1+o2                       //flux in Wb
```

```
b = ot/a                          //flux density in T
i = ((hefab*iefab)+(hbe*ibe))/n      //in ampere
u = b3/h1                            //permeability
ur = u/uo                            //permeability
u1  = b1/h2                            //permeability
ur1 = u1/uo                            //permeability
u4 = b/hefab                            //
   permeability
ur2 = u4/uo                            //permeability

//Results
printf("I = %.2f A \n",i)            //required
   curent
printf("Ur = %.1f \n",ur)
printf("For section be, Ur = %d \n",ur1)
printf("For section efab, Ur = %.2f \n",ur2)
```

Scilab code Exa 11.8 Example 8

```
//Chapter 11, Example 11.8
clc
//Variable Declaration
n = 60                           //numbers of turns
i = 5                            //in ampere
labcda = 0.3                     //in meter
b = 0.39                         //in T
a = 2*10^-4

//Calculation
habcda = (n*i)/labcda            //by Ampere's
   circuital law
o = b*a                          //magnetic flux in
  Wb

//Results
```

```
printf("O = %.2f*10^-4 Wb",o*10^4)
```

Scilab code Exa 11.9 Example 9

```
//Chapter 11, Example 11.9
clc
//Variable Declaration
n = 100                              //number of turns
i = 4                                //in ampere
ig = 0.001                           //in meter
uo = 12.56*10^-7                     //permeability
hcore = 850                          //in At/m
lcore = 0.16                         //in meter
a = 0.003                            //area in meter
    square
hcore1 = 1500                        //in At/m
lcore1 = 0.16                        //in meter

//Calculation
hg = (n*i)/ig                        //in At/m
bg = uo*hg                           //in T
o = bg*a                             //flux in Wb
ni1 = (hcore*lcore)+(hg*ig)               //applying
    Ampere's circuital law
o1 = (1-0.3)*o                            //magnetic flux
b = o1/a                                 //in T
hgig = (7.96*10^5)*b*ig                  //in At/m
ni = (hcore*lcore)+(hgig)                 //in At
hga = n*i                                //in At

//Results
printf("NI = %.2f At > %d At \n",ni,hga)
printf("O = %.3f x 10^-3 Wb",o1*10^3)
```

Chapter 12

Inductors

Scilab code Exa 12.1 `Example 1`

```
//Chapter 12, Example 12.1
clc
//Variable Declaration
d = 4*10^-3                //diameter in meter square
n = 100                    //number of turns
uo = 12.56*10^-7           //air in Wb/A.m
l = 0.1                    //length of the core in meter

//Calculation
a = (%pi*d*d)/4             //area of the core in square
    meter
lo = (n*n*uo*a)/l          //inductance of the air-coil
   in H

//Results
printf("Lo = %.2f uH",lo*10^6)
```

Scilab code Exa 12.2 `Example 2`

```
//Chapter 12, Example 12.2
clc
//Variable Declaration
ur = 2000              //permeability of the core
lo = 1.58*10^-6        //inductance of the coil

//Calculation
l = ur*lo              //inductance in H

//Results
printf("L = %.2f mH",l*10^3)
```

Scilab code Exa 12.3 Example 3

```
//Chapter 12, Example 12.3
clc
//Variable Declaration
i = 10*10^-3                        //current in milli
    ampere
l = 4*10^-3                         //inductor in henry
t = 2*10^-3                         //time in millisecond
t1 = 5*10^-3                        //time in millisecond
i1 = 0                              //current

//Calculation
vl2 = (l*i1)/t                     //voltage in
    millivolts
vl = (l*i)/t                       //voltage in
    millivolts
vl1 = (-l*i)/t1                    //voltage

//Results
printf("a. VL = %d \n",vl2)
printf("b. VL = %d mV \n",vl*10^3)
printf("c. VL = %d mV \n",vl1*10^3)
```

```
printf("d. VL = %d",vl2)
```

Scilab code Exa 12.4 Example 4

```
//Chapter 12, Example 12.4
clc
//Variable Declaration
l = 4                      //inductance in H
r1 = 2000                  //resistor1 in ohm
e = 50                     //voltage

//Calculation
to = l/r1                 //in millisecond
im = e/r1                 //in milliAmperea
il = im                   //load current
vl = e                    //load voltage

//Results
printf("iL = (%d x 10^-3)(1-e^-t/(%dx10^-3)) \n",il
   *10^3,to*10^3)
printf("VL = %de^-t/(%dx10^-3) \n",vl,to*10^3)
```

Scilab code Exa 12.5 Example 5

```
//Chapter 12, Example 12.5
clc
//Variable Declaration
e = 16                       //battery volts
r1 = 2.2*10^3                //resistance1 in kohm
r2 = 6.8*10^3                //resistance2 in kohm
l = 100*10^-3                //inductance in mH
ii = 4*10^-3                 //loade current in mA

```

```
//Calculation
rt = r1+r2                  //total resistance
if = e/(r1+r2)              //in mA
to = l/rt                   //time constant in microsecond
il = if                     //current through the coil−
il2 = ii-if                 //once the switch is closed
vr1 = ii*r1                  //voltage of resistance1
vr2 = ii*r2                  //voltage of resistance2
vm = e-vr1-vr2                 //voltage across the coil
vl = vm

//Results
printf("a. iL = %.2f mA",il*10^3)
printf(" + %.2f mAe^−t/%.2fus \n",il2*10^3,to*10^6)
printf("b. VL = %d*e^−t/%.2fus \n",vl,to*10^6)
printf("c. iL = %.2f mA ",il*10^3)
```

Scilab code Exa 12.6 Example 6

```
//Chapter 12, Example 12.6
clc
//Variable Declaration
r1 = 2000                  //resistor1 in kohm
r2 = 3000                  //resistor2 in kohm
l = 4                      //inductor in henry
e = 50                     //battery voltage


//Calculation
to = l/r1                  //time constant in ms
vl = e                     //load voltage
im = e/r1                  //current in milliampere
il = im                    //load current
vr1 = e                    //voltage across r1
```

```
vr2 = vr1                  //voltage across r2
to1 = l/(r1+r2)
vi = (1+(r2/r1))*e         //voltage in volts
vl1 = -vi                  //voltage at switch open
im1 = e/r1                 //current in milliampere
il1 = im1                  //current at switch open
vr11 = e                   //voltage across at r1 switch open
vr22 = (-r2/r1)*e          //voltage across at r2 switch open

//Results
printf("a. VL = %d*e^-t/(%d*10^-3) \n",vl,to*10^3)
printf("    iL = %d*10^-3(1-e^-t/%d*10^-3) \n",il*10^3,to*10^3)
printf("    VR1 = %d(1-e^-t/%d*10^-3) \n",vr1,to*10^3)
printf("    VR2 = %d V\n",vr2)
printf("b. iL = %.3fe^-t/(0.8*10^-3) \n",il)
printf("    VL = -%de^-t/(%.1f*10^-3) \n",vi,to1*10^3)
printf("    VR1 = %de-t/(%.1f*10^-3) \n",vr11,to1*10^3)
printf("    VR2 = %de-t/(%.1f*10^-3)",vr22,to1*10^3)
```

Scilab code Exa 12.7 Example 7

```
//Chapter 12, Example 12.7
clc
//Variable Declaration
r1 = 20000                 //resistor1 in kohm
r2 = 4000                  //resistor2 in kohm
r3 = 16000                 //resistor3 in kohm
l = 80                     //inductor in mH
e = 12                     //supply voltage
```

```
n = 2                    //number of turns

//Calculation
rth = r1/n                              //by thevenin theorem
to = l/rth                              //time in microsecond
eth = (r2+r3)*e/(r1+r2+r3)              //applying voltage divider
im = eth/rth                            //current in milliampere
il = im                                 //load current
vl = eth                                //load voltage

//Results
printf("a. iL = (%.1f*10^-3)(1-e^-t/(%d*10^-6)) \n",il*10^3,to*10^3)
printf("    VL = %de^-t/(%d*10^-6) \n",vl,to*10^3)



//Graph
x1=0:(1*10**-3):(50*10**-3)
vc=vl*(exp(-x1/to))                          //for VL
ic=il*(1-exp(-x1/to))                        //for iL

subplot(221)
plot(x1,vc)
xlabel("t(s)")
ylabel("VL (V)")
xtitle("VL for the R-L network")
subplot(222)
plot(x1,ic)
xlabel("t(s)")
ylabel("iL (mA)")
xtitle("iL for the R-L network\")
```

Scilab code Exa 12.8 Example 8

```
//Chapter 12, Example 12.8
clc
//Variable Declaration
r1 = 2.2*10^3                //resistor1 in kohm
r2 = 8.2*10^3                //resistor2 in kohm
r3 = 1000                    //resistor3 in kohm
i = 12*10^-3                 //sources current
e = 6                        //voltage
l = 680*10^-3                //inductor in mH
i = 12*10^-3                 //current source in mA
E = 26.4                     //voltage

//Calculation
ii = -e/r3                  //using ohm's law
rth = r1+r2                 //applying thevenin's theorem
eth = i*r1                  //in volts
if = E/rth                  //in milliampere
to = l/rth                  //time constant
il = if                     //in milliampere
il1 = ii-if                    //in milliampere

//Results
printf("a. ii = %d mA \n",ii*10^3)
printf("b. iL = %.2f mA %.2fmAe^-t/(%.2fus) ",il*10^3,il1*10^3,to*10^6)

//Graph
x1=0:(0.1*10**-3):(0.5*10**-3)
ic=il+(il1*exp(-x1/to))                    //for iL
plot(x1,ic)
xlabel("t(s)")
```

```
ylabel("iL (mA)")
xtitle("The current iL for the network")
```

Scilab code Exa 12.9 Example 9

```
//Chapter 12, Example 12.9
clc
//Variable Declaration
l = 1.2                      //L2 and L3 are equal
n = 2                        //number of turns
l4 = 1.8                     //inductance L4 in henry
l1 = 0.56                    //inductance L1 in henry

//Calculation
lt = l/n                     //total inductance in henry
lt11 = (lt*l4)/(lt+l4)       //total inductance in henry
lt2 = l1+lt11                //total inductance in henry
//Results
printf("LT = %.2f H",lt2)
```

Scilab code Exa 12.10 Example 9

```
//Chapter 12, Example 12.10
clc
//Variable Declaration
r1 = 2                     //resistance1 in ohms
r2 = 3                     //resistance2 in ohms
e  = 10                    //voltage in volts

//Calculation
```

```
il = e/(r1+r2)          //load current in ampere
vc = (r2*e)/(r1+r2)     //load voltage in volts

//Results
printf("IL = %d A \n",il)
printf("Vc = %d V ",vc)
```

Scilab code Exa 12.11 `Example 11`

```
//Chapter 12, Example 12.11
clc
//Variable Declaration
r1 = 2                          //resistance1 in ohms
r2  = 5                         //resistance2 in ohms
r3 = 1                          //resistance3 in ohms
r4 = 4                          //resistance4 in ohms
r5 = 7                          //resistance5 in ohms
e  = 50                         //voltage in volts

//Calculation

i1 = e/(r1+r3+r5)                //current1 in ampere
i2= i1                           //equal
v2 = i2*r5                       //voltage2 by ohm's
    law
v1 = (r3+r5)*e/(r1+r3+r5)        //voltage1 by voltage
     divider

//Results
printf("I1 = %d A \n",i1)
printf("V2 = %d V \n",v2)
printf("V1 = %d V \n",v1)
```

Scilab code Exa 12.12 `Example 12`

```
//Chapter 12, Example 12.12
clc
//Variable Declaration
r1 = 3                    //resistance1
l = 6*10^-3               //inductance in mH
e = 15                    //voltage in volts
r2 = 2                    //resistance2
//Calculation
im = e/(r1+r2)            //inductance current
ws = (l*im*im)/2          //energy stored

//Results
printf("Ws = %d mJ",ws*10^3)
```

Chapter 13

Sinusoidal Alternating Waveforms

Scilab code Exa 13.1 Example 1

```

//Chapter 13, Example 13.1
clc
//Variable Declaration
f1 = 60                       //frequency in hertz
f2 = 1000                     //frequency in hertz

//Calculation
t1 = 1/f1                    //time in milli second
t2 = 1/f2                    //time in milli second

//Results
printf("T = %.2f ms \n",t1*10^3)
printf("T = %d ms ",t2*10^3)
```

Scilab code Exa 13.2 Example 2

```
//Chapter 13, Example 13.2
clc
//Variable Declaration
t = 20*10^-3                    //time in milli second


//Calculation
f = 1/t                         //frequency in hertz

//Results
printf("f = %d Hz",f)
```

Scilab code Exa 13.3 Example 3

```
//Chapter 13, Example 13.3
clc
//Variable Declaration
d = 4                           //division
t= 50*10^-6                     //time in microsecond
d1 = 2                           //division
v = 0.1                          //in volts

//Calculation
t1 = d*t                        //time in microsecond
f = (1/t1)                      //frequency in hertz
vm = d1*v                       //in volts

//Results
printf("T = %d us \n",t1*10^6)
printf("f = %d kHz \n",f/10^3)
printf("Vm = %.1f V \n",vm)
```

Scilab code Exa 13.4 Example 4

```

//Chapter 13, Example 13.4
clc
//Variable Declaration
f = 60                          //frequency in hertz


//Calculation
w = 2*3.14*f                    //in rad/s

//Results
printf("w = %d rad/s",w)
```

Scilab code Exa 13.5 Example 5

```
//Chapter 13, Example 13.5
clc
//Variable Declaration
w = 500                             //in rad/s


//Calculation
t = (2*3.14)/w                  //time in milliseconds
f = 1/t                         //frequency in hertz

//Results
printf("T = %.2f ms \n",t*10^3)
printf("f = %.2f Hz ",f)
```

Scilab code Exa 13.6 Example 6

```
//Chapter 13, Example 13.6
clc
//Variable Declaration
l = (3.14/2)                //in rad/s
w = 200                     //in rad/s

//Calculation
t = l/w                     //time in millisecond

//Results
printf("t = %.2f ms \n",t*10^3)
```

Scilab code Exa 13.7 Example 7

```
//Chapter 13, Example 13.7
clc
//Variable Declaration
f = 60                        //frequency in hertz
t= 5*10^-3                    //time in milli second

//Calculation
l = 2*3.14*f*t                //in rad/s
lo = (180*l)/3.14             //in degree


//Results
printf("alpha = %.3f rad \n",l)
printf("alpha(degree) = %d degree",lo)
```

Scilab code Exa 13.8 Example 8

```
//Chapter 13, Example 13.8
clc
//Variable Declaration
e1 = 5                          //in volts
l = 0.8*3.14                    //in degree

//Calculation
e = e1*sind(40)                 //in volts
lo = 180*l/3.14                 //in degree
e2 = 5*sind(lo)                 //in volts

//Results
printf("e = %.3f V \n",e)
printf("e = %.3f V \n",e2)
```

Scilab code Exa 13.9 Example 9

```
//Chapter 13, Example 13.9
clc
//Variable Declaration
v = 4                              //in volts
em = 10                            //in volts
w = 377                            //in rad/s
//Calculation
l1 = asind(v/em)                   //alpha1 in degree
l2 = 180-(l1)                      //alpha2 in degree
l3 = (%pi/180)*(l1)                //in rad
t1 = l3/w                          //in millisecond
l4 = (%pi/180)*l2                  //in rad
t2 = l4/w                          //in millisecond
```

```
//Results
printf("a. alpha1 = %.3f \n",l1)
printf("   alpha2 = %.3f \n",l2)
printf("b. t1 = %.2f ms \n",t1*10^3)
printf("   t2 = %.2f ms \n",t2*10^3)
```

Scilab code Exa 13.10 Example 10

```
//Chapter 13, Example 13.10
clc
//Variable Declaration
w = 314                    //in rad/s

//Calculation
t = (2*3.14)/w             //in millisecond
t1 = t/2                   //in millisecond
t2 = t/4                   //in millisecond
t3 = t/12                  //in millisecond

//Results
printf("360: T = %d ms \n",t*10^3)
printf("180: T/2 = %d ms \n",t1*10^3)
printf("90:  T/4 = %d ms \n",t2*10^3)
printf("30:  T/12 = %.2f ms \n",t3*10^3)
```

Scilab code Exa 13.11 Example 11

```
//Chapter 13, Example 13.11
clc
//Variable Declaration
w = 1000                  //in rad/s
t = 2*10^-3               //in millisecond
i1 = 6*10^-3              //in milliampere
```

```
//calculation

apl = (180*2)/3.14              //alphain degree
sin114 = 0.9093                 //in degree
i = i1*sin114                   //in milliampere

//Results
printf("i = %.2f mA ",i*10^3)
```

Scilab code Exa 13.13 Example 13

```
//Chapter 13, Example 13.13
clc
//Variable Declaration
v1 = 10                     //in volts
t1 = 1*10^-3                //in millisecond
v2 = 14                     //in volts
v3 = 6                      //in volts

//Calculation
g = ((v1*t1)-(v1*t1))/(2*10^-3)
g1 = ((v2*t1)-(v3*t1))/(2*10^-3)


//Results
printf("a. G = %d V \n",g)
printf("b. G = %d V \n",g1)
```

Scilab code Exa 13.14 Example 14

```
//Chapter 13, Example 13.14
clc
```

```
//Variable Declaration
v1  = 3                    //in volts
v2 = 1                    //in volts
t1 = 4*10^-3              //in millisecond
v3 = 10                   //in volts
t2 = 2*10^-3              //in millisecond
v4 = 4                    //in volts
v5 = 2                    //in volts

//Calculation
g = ((v1*t1)-(v2*t1))/(8*10^-3)
g1 = (-(v3*t2)+(v4*t2)-(v5*t2))/(10*10^-3)


//Results
printf("a. G = %d V \n",g)
printf("b. G = %.1f V \n",g1)
```

Scilab code Exa 13.15 Example 15

```

//Chapter 13, Example 13.15
clc
//Variable Declaration
a = 2                     //in volts
b = 2                     //in volts
am = 1                    //in volts
//Calculation
g = (a*am-b*am)/(2*3.14)    //in volts

//Results
printf("G = %d V ",g)
```

Scilab code Exa 13.16 Example 16

```
//Chapter 13, Example 13.16
clc
//Variable Declaration
v1 = 16                         //in volts
v2 = 2                        //in volts


//Calculation
g1 = (v1+v2)/2                          //in volts
g2 = g1-(16)                          //in volts


//Results
printf("G = %d mv ",g2)
```

Scilab code Exa 13.17 Example 17

```

//Chapter 13, Example 13.17
clc
//Variable Declaration
a = 2                       //in volts
b = 0                       //in volts
am = 10                       //in volts
//Calculation
g = (a*am+b)/(2*3.14)                          //in volts

//Results
printf("G = %.2f V ",g)
```

Scilab code Exa 13.18 Example 18

```
//Chapter 13, Example 13.18
clc
//Variable Declaration
g = 2                          //in volts

//Results
printf("G = %d mV ",g)
```

Scilab code Exa 13.19 `Example 19`

```
//Chapter 13, Example 13.19
clc
//Variable Declaration
a = 0.707                      //in volts
b = 12*10^-3                    //in milliampere
c = 169.73                      //in volts

//Calculation
i = a*b                           //in milliampere
v = a*c                           //in volts
//Results
printf("a. Irms = %.3f mA \n",i*10^3)
printf("b. Irms = %.3f mA \n",i*10^3)
printf("c. Vrms = %d mA ",v)
```

Scilab code Exa 13.20 `Example 20`

```
//Chapter 13, Example 13.20
clc
//Variable Declaration
pdc = 3.6                      //power in watt
vdc= 120                       //in volts

```

```

//Calculation
idc = pdc/vdc                 //in milliampere
im = sqrt(2)*idc              //in milliampere
em = sqrt(2)*vdc               //in millivolts

//Results
printf("Idc = %d  mA \n",idc*10^3)
printf("Im = %.2f mA \n",im*10^3)
printf("Em = %.2f mA \n",em)
```

Scilab code Exa 13.21 Example 21

```
//Chapter 13, Example 13.21
clc
//Variable Declaration
x1 = 4                    //in second
y1 = 9                    //in volts
x2 = 4                    //in second
y2 = 1                     //in volts
x3 = 8                    //in volts

//Calculation
v = sqrt(((y1*x1)+(y2*x2))/x3)          //in volts

//Results
printf("Vrms = %.3f  V ",v)
```

Scilab code Exa 13.22 Example 22

```
//Chapter 13, Example 13.22
clc
//Variable Declaration
```

```
y1 = 100                    //in volts
x1 = 2                      //in second
y2 = 16                     //in volts
y3 = 4                      //in volts

//Calculation
v = sqrt(((y1*x1)+(y2*x1)+(y3*x1))/10)        //in
    volts

//Results
printf("Vrms = %.3f V ",v)
```

Scilab code Exa 13.23 `Example 23`

```
//Chapter 13, Example 13.23
clc
//Variable Declaration
y1 = 1600                        //in volts
x = 10*10^-3                     //time in millisecond

//Calculation
v = sqrt(((y1*x)+(y1*x))/(20*10^-3))     //in volts

//Results
printf("Vrms = %d V ",v)
```

Scilab code Exa 13.24 `Example 24`

```
//Chapter 13, Example 13.24
clc
//Variable Declaration
v1 = 20                         //in volts
v2 = 25                         //in volts
```

```
vm1 = v1                              //in volts
vm2 = 15                              //in volts

//Calculation
m = 1.11*v1                              //in volts
vr1 = 0.707*vm1                              //in volts
vdc = vr1                             //in volts
vr2 = 0.707*vm2                              //in volts
//Results
printf("Meter indication = %.1f V \n",m)
printf("Vrms = %.2f V \n",vr1)
printf("Vrms = Vdc = %d V \n",vdc)
printf("Vrms = %.1f V \n",vr2)
```

Chapter 14

The Basic Elements and Phasors

Scilab code Exa 14.1 Example 1

```
//Chapter 14, Example 14.1
clc
//Variable Declaration
vm = 100                    //voltage magnitude of (a)
r = 10                      //resistance in ohms
vm1 = 25                    //voltage magnitude of (b)


//Calculation
i = vm/r                    //current of (a)
i1 = vm1/r                  //current of (b)

//Results
printf("a. i = %dsin377t \n",i)
printf("b. i = %.1fsin(377t+60dgree) \n",i1)
```

Scilab code Exa 14.2 Example 2

```
//Chapter 14, Example 14.2
clc
//Variable Declaration
im = 40                     //current magnitude in
    ampere
r = 5                       //resistance in ohms

//Calculation
v = im*r                    //voltage in volt

//Results
printf("v = %dsin(377t+30degree) ",v)
```

Scilab code Exa 14.3 Example 3

```
//Chapter 14, Example 14.3
clc
//Variable Declaration
w = 377                       //in rad/s
l = 0.1                       //inductance in ohms
im = 10                       //current of (a)
im1= 7                        //current of (b)

//Calculation
xl = w*l                     //inductance reactance
vm = im*xl                   //voltage of (a)
vm1 = im1*xl                 //voltage of (b)

//Results
printf("a. v = %dsin(377t+90degree) \n",vm)
printf("b. v = %.1fsin(377t+20degree)",vm1)
```

Scilab code Exa 14.4 Example 4

```
//Chapter 14, Example 14.4
clc
//Variable Declaration
w = 20                          //in rad/s
l = 0.5                         //inductance in henry
vm = 100                        //voltage in volts


//Calculation
xl = w*l                        //inductance reactance
im = vm/xl                      //current in ampere

//Results
printf("i = %d sin(20t - 90degree)",im)
```

Scilab code Exa 14.5 Example 5

```
//Chapter 14, Example 14.5
clc
//Variable Declaration
w = 400                                 //in rad/s
c = 1*10^-6                             //capacitance in
    microfarad
vm = 30                                 //voltage in volts


//Calculation
xc = 1/(w*c)                            //capacitance
    reactance
im = vm/xc                              //current in ampere
```

```

//Results
printf("i = %d x 10^-3 sin(400t + 90degree)",im
    *10^3)
```

Scilab code Exa 14.6 Example 6

```
//Chapter 14, Example 14.6
clc
//Variable Declaration
w = 500                             //in rad/s
c = 100*10^-6                       //capacitance in
    micro farad
im = 40                             //current in ampere


//Calculation
xc = 1/(w*c)                        //capacitance
    reactance
vm  = im*xc                         //voltage in volts

//Results
printf("v = %d sin(500t - 30degree)",vm)
```

Scilab code Exa 14.7 Example 7

```
//Chapter 14, Example 14.7
clc
//Variable Declaration
vm1 = 100                         //Vm of (a)
im1 = 20                          //Im of (a)
vm2 = 1000                        //Vm of (b)
im2 = 5                           //Im of (b)
```

```
vm3 = 500                          //Vm of (c)
im3 = 1                            //Im of (c)
vm4 = 50                           //Vm of (d)
im4 = 5                            //Im of (d)
w = 377                            //in rad/s
w1 = 157                           //in rad/s
//Calculation
r = vm1/im1                         //resistance of (a)
xl = vm2/im2                        //inductance reactance
l = xl/w                            //inductance in henry
xc = vm3/im3                        //capacitance reactance
c = 1/(w1*xc)                       //capacitance in farad
r1 = vm4/im4                        //resistance of (d)

//Results
printf("a. R = %d ohm \n",r)
printf("b. L = %.3f H \n",l)
printf("c. C = %.2f uF \n",c*10^6)
printf("d. R = %d ohm \n",r1)
```

Scilab code Exa 14.8 `Example 8`

```
//Chapter 14, Example 14.8
clc
//Variable Declaration
l = 200*10^-3                       //inductance in henry
r = 5*10^3                          //resistance in ohms
xl = 2*3.14*l                       //reactance inductor

//Calculation
f= r/xl                              //frequency in hertz

```

```
//Results
printf("f = %.2f kHz ",f*10^-3)
```

Scilab code Exa 14.9 Example 9

```
//Chapter 14, Example 14.9
clc
//Variable Declaration
l = 5*10^-3                    //inductance in henry
c = 0.1*10^-6                  //capacitance in
   farad
//calculation
xl = 2*3.14*l                  //inductance
   reactance
xc = 1/(2*3.14*c)              //capacitance
   reactance
f= 1/(2*3.14*sqrt(l*c))        //frequency in hertz

//Results
printf("f = %.2f kHz",f*10^-3)
```

Scilab code Exa 14.10 Example 10

```
//Chapter 14, Example 14.10
clc
//Variable Declaration
vm = 10                    //voltage in volts
im = 5                     //current in ampere


//Calculation
p = (vm*im)/2                          //power
   dissipation in watt
```

```
r= (vm/im)                              //resistance in ohm
p1 = ((0.707*vm)*(0.707*vm))/r          //power in watt
p2 = ((0.707*im)*(0.707*im))*r          //power in watt

//Results
printf("P = %d W \n",p)
printf("P = %.2f W \n",p1)
printf("P = %.2f W \n",p2)
```

Scilab code Exa 14.11 Example 11

```
//Chapter 14, Example 14.11
clc
//Variable Declaration
vm = 100                     //peak voltage (a)
im = 20                      //peak current
ov = 40                      //theta voltage in degree (a)
oi = 70                      //theta current in degree (a)
vm1 = 150                    //peak voltage (b)
im1 = 3                      //peak current (b)
ov1 = -70                    //theta voltage in degree (b)
oi1 = -50                    //theta current in degree (b)

//Calculation
p = ((vm*im)/2)*0.866         //power (a) in watt
p1 = ((vm1*im1)/2)*0.9397     //power (b) in watt

//Results
printf("P = %d W \n",p)
printf("P = %.2f W \n",p1)
```

Scilab code Exa 14.12 Example 12

```
//Chapter 14, Example 14.12
clc
//Variable Declaration
a = cosd(40-(-20))                          //leading
b = cosd(80-30)                             //lagging
p = 100                                     //in watt
v = 20                                      //in voltage
i = 5                                       //in ampere

//Calculation
fp = p/(v*i)                          //power factor

//Results
printf("a. Fp = %.1f leading \n",a)
printf("b. Fp = %.4f leading \n",b)
printf("c. Fp = %d \n",fp)
```

Scilab code Exa 14.13 Example 13

```
//Chapter 14, Example 14.13
clc
//Variable Declaration
c1=complex(3,4)
c2=complex(0,-6)
c3=complex(-10,-20)

//Calculation
//Function to create plot on complex plane
function complexPlot(x,y,z)
```

```
    xpts = [0 x];
    ypts = [0 y];
    plot(xpts, ypts);
    xtitle(z);
    xlabel("Real");
    ylabel("Imaginary");
endfunction

//Plot
scf(1)
complexPlot(real(c1),imag(c1),"3 + j4")
scf(2)
complexPlot(real(c2),imag(c2),"0 - j6")
scf(3)
complexPlot(real(c3),imag(c3),"-10 - j20")
```

Scilab code Exa 14.14 Example 14

```
//Chapter 14, Example 14.14
clc

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction


[cr1,ci1]=polar2rect(5,30)
[cr2,ci2]=polar2rect(7,-120)
[cr3,ci3]=polar2rect(-4.2,60)

//Calculation
//Function to create plot on complex plane
function complexPlot(x,y,z)
```

```
    xpts = [0 x];
    ypts = [0 y];
    plot(xpts, ypts);
    xtitle(z);
    xlabel("Real");
    ylabel("Imaginary");
endfunction

//Plot
scf(1)
complexPlot(cr1,ci1,"5 < 30degree")
scf(2)
complexPlot(cr2,ci2,"7 < 120 degree")
scf(3)
complexPlot(cr3,ci3,"4.2 < 60 degree")
```

Scilab code Exa 14.15 Example 15

```
//Chapter 14, Example 14.15
clc
funcprot(0)

//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction


//Variable Declaration
x = 3                           //real part
y = 4                           //imaginary part

//Calculation
[a,b]=rect2polar(x,y)        //function of conversion
```

```

//Results
printf("C = %d < %.2f degree", a,b)
```

Scilab code Exa 14.16 Example 16

```
//Chapter 14, Example 14.16
clc

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction


//Variable Declaration
a=10                        //radius
b=45                        //angular in degree

//Calculation
[r,i]=polar2rect(a,b)       //function of conversion

//Results
printf("C = %.2f + j %.2f",r,i)
```

Scilab code Exa 14.17 Example 17

```
//Chapter 14, Example 14.17
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
```

```
    th= 180-(atand(y/x))
endfunction

//Variable Decleration
a = 6                          //real part
b = 3                          //imaginary

//Calculation
[r,th] = rect2polar(a,b)        //function of
   conversion

//Results
printf("C = %.2f < %.2f degree",r,th)
```

Scilab code Exa 14.18 Example 18

```
//Chapter 14, Example 14.18
clc

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

//Variable Declaration
a = 10                          //real value
b = 230                         //angle in degree
//Calculation
[r,i]=polar2rect(a,b)       //function of conversion

//Results
printf("C = %.3f  j%.3f",r,i)
```

Scilab code Exa 14.19 Example 19

```
//Chapter 14, Example 14.19
clc
//addition
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
endfunction

//Variable Declaration
x1=complex(2,4)                   //complex form c1
y1 = complex(3,1)                 //complex form c2
x2 = complex(3,6)                 //complex form c1
y2 = complex(-6,3)                //complex form c2

//Calculation
[a,b]=addition(real(x1),real(y1),imag(x1),imag(y1))          //function of conversion
[c,d]=addition(real(x2),real(y2),imag(x2),imag(y2))          //function of conversion

//Results
printf("a. C = %d + j%d \n", a,b)
printf("b. C = %d + j%d", c,d)
```

Scilab code Exa 14.20 Example 20

```
//Chapter 14, Example 14.20
clc
//subtraction
function[c1,c2]=subtraction(x1,x2,y1,y2)
```

```
    c1 = x1-x2
    c2 = y1-y2
    endfunction
//Variable Declaration
x1 = complex(1,4)                //complex form c2
y1 = complex(4,6)                //complex form c1
x2 = complex(-2,5)                //complex form c2
y2 = complex(3,3)                //complex form c1
//Calculation
[a,b]=subtraction(real(y1),real(x1),imag(y1),imag(x1
   ))        //function of conversion
[c,d]=subtraction(real(y2),real(x2),imag(y2),imag(x2
   ))        //function of conversion


//Results
printf("a. C = %d + j%d \n", a,b)
printf("b. C = %d %dj", c,d)
```

Scilab code Exa 14.21 Example 21

```
//Chapter 14, Example 14.12
clc
//Variable Declaration
a1 = 2                          //real value
a2 = 3                          //real value
a3= 2                          //real value
a4 = 4                          //real value

//Calculation
x1 = a1+a2                          //real value
x2 = a3+a4                          //real value

//Results
printf("a. %d<45 \n",x1)
```

```
printf("b. %d<0",x2)
```

Scilab code Exa 14.22 Example 22

```
//Chapter 14, Example 14.22
clc
function[c1,c2]=equ(x1,x2,y1,y2)
    c1 = (x1*x2)-(y1*y2)
    c2 = (y1*x2)+(x1*y2)
    endfunction
//Variable Declaration
x1 = complex(2,3)           //complex form of c1
y1 = complex(5,10)           //complex form of c2
x2 = complex(-2,-3)           //complex form of c1
y2 = complex(4,-6)           //complex form of c2
//Calculation
[a,b]=equ(real(x1),real(y1),imag(x1),imag(y1))
[c,d]=equ(real(x2),real(y2),imag(x2),imag(y2))

//Results
printf("a. C = %d + j%d \n", a,b)
printf("b. C = %d + j%d", c,d)
```

Scilab code Exa 14.23 Example 23

```
//Chapter 14, Example 14.23
clc
//multiplication
function[c1,c2]=equ(x1,x2,y1,y2)
    c1 = (x1*x2)
    c2 = (y1+y2)
    endfunction
//Variable Declaration
```

```
x1 = 5                        //real value
x2 = 10                        //real value
y1 = 20                         //angle in degree
y2 = 30                         //angle in degree
x3 = 2                        //real value
x4 = 7                        //real value
y3 = -40                         //angle in degree
y4 = 120                         //angle in degree

//Calculation
[a,b]=equ(x1,x2,y1,y2)          //function of multiplication
[c,d]=equ(x3,x4,y3,y4)          //function of multiplication
//Results
printf("a. C = %d < %d degree \n", a,b)
printf("b. C = %d < %d degree", c,d)
```

Scilab code Exa 14.24 Example 24

```
//Chapter 14, Example 14.24
clc
//muliplication
function[C1,C2]=equ2(x1,x2,y1,y2)
    C1 = ((x1*x2)+(y1*y2))/((y1*y1)+(y2*y2))
    C2 = ((x2*y1)-(x1*y2))/((y1*y1)+(y2*y2))
    endfunction

//Variable Declaration
x1 = complex(1,4)          //complex form of c1
y1 = complex(4,5)          //complex form of c2
x2 = complex(-4,-8)          //complex form of c1
y2 = complex(6,-1)          //complex form of c2
//Calculation
[a,b]=equ2(real(x1),real(y1),imag(x1),imag(y1))    //
```

```
function
[c,d]=equ2(real(x2),real(y2),imag(x2),imag(y2))    //
    function
//Results
printf("C1/C2 = %.3f + j%.3f \n",a,b)
printf("C1/C2 = %.3f + j%.3f \n",c,d)
```

Scilab code Exa 14.25 `Example 25`

```
//Chapter 14, Example 14.25
clc
//division
function[c1,c2]=equ(x1,x2,y1,y2)
    c1 = (x1/x2)
    c2 = (y1-y2)
    endfunction
//Variable Declaration
x1 = 15                         //real value
x2 = 2                         //real value
y1 = 10                         //angle in degree
y2 = 7                         //angle in degree
x3 = 8                         //real value
x4 = 16                         //real value
y3 = 120                         //angle in degree
y4 = -50                         //angle in degree
//Calculation
[a,b]=equ(x1,x2,y1,y2)              //function of
    division
[c,d]=equ(x3,x4,y3,y4)              //function of
    division
//Results
printf("C1/C2 = %.1f < %d degree\n",a,b)
printf("C1/C2 = %.1f < %d degree\n",c,d)
```

Scilab code Exa 14.26 Example 26

```
//Chapter 14, Example 14.26
clc
//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
   //subtraction
function[c1,c2]=subtraction(x1,x2,y1,y2)
    c1 = x1-x2
    c2 = y1-y2
    endfunction
//addition
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction
```

```
  //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
//Variable Declaration
x1 = complex(2,3)          //complex form of x1
x2 = complex(4,6)          //complex form of x2
y1 = complex(7,7)          //complex form of x3
y2 = complex(3,-3)          //complex form of x4
x3 = 50                     //real value
y3 = 30                    //angle in degree
x4 = complex(5,5)          //complex form of x5
x5 = 10                     //real value
y5= -20                    //angle in degree
x6 = 2                     //real value
y6 = 20                    //angle in degree
x7 = complex(3,4)          //complex form of x6
x8 = complex(8,-6)          //complex form of x7
x9 = 3                     //real value
y9 = 27                    //angle in degree
x10 = 6                     //real value
y10 = -40                   //angle in degree
//Calculation
[a,b]=addition(real(x1),real(x2),imag(x1),imag(x2))
      //function of addition
[c,d]=subtraction(real(y1),real(y2),imag(y1),imag(y2
   ))   //function of subtraction
[aa,bb]=rect2polar(a,b)     //function of conversion
[cc,dd]=rect2polar(c,d)     //function of conversion
[e,f]=voltdivider(aa,cc,bb,-dd)    //function of
   multiplication
[g,h]=voltdivider(cc,cc,dd,-dd)    //function of
   multiplication
[m,n]=division(e,g,f,h)    //function of division
[mm,nn]=polar2rect(m,n)     //function of conversion
[o,p]=rect2polar(real(x4),imag(x4))     //function of
    conversion
```

```
[oo,pp]=voltdivider(x3,o,y3,p)    //function of
    multiplication
[ee,ff]=division(oo,x5,pp,y5)    //function of
    division
[gg,hh]=voltdivider(x6,x6,y6,y6)    //function of
    multiplication
[i,j]=rect2polar(real(x7),imag(x7))     //function of
     conversion
[ii,jj]=voltdivider(gg,i,hh,j)    //function of
    multiplication
[s,t]=rect2polar(real(x8),imag(x8))     //function of
     conversion
[ss,tt]=division(ii,s,jj,t)    //function of division
[v,w]=polar2rect(x9,y9)     //function of conversion
[vv,ww]=polar2rect(x10,y10)     //function of
    conversion
[ma,na]=subtraction(v,vv,w,ww)    //function of
    subtraction

//Results
printf("a. = %.3f + j%.3f \n",mm,nn)
printf("b. = %.2f < %d degree \n",ee,ff)
printf("c. = %.1f < %d degree \n",ss,tt)
printf("d. = %.3f + j %.3f \n",ma,na)
```

Scilab code Exa 14.29 Example 29

```
//Chapter 14, Example 14.29
clc
function[t,p]=time2phasor(x,y)
    t=x*0.707
    p=y
endfunction

//Variable Declaration
```

```
x1 = 50                           //real value of time
y1 = 0                            //angle in degree
x2 = 69.6                         //peak voltage
y2 = 72                           //angle in degree
x3 = 45                           //peak voltage
y3 = 90                           //angle in degree
//Calculation
[a,b]=time2phasor(x2,y2)     //function of conversion
[c,d]=time2phasor(x3,y3)     //function of conversion
//Results
printf("a. %d < %d degree \n",x1,y1)
printf("b. %.2f < %d degree \n",a,b)
printf("c. %.2f < %d degree \n",c,d)
```

Scilab code Exa 14.30 Example 30

```
//Chapter 14, Example 14.30
clc
function[m,n]=phasor2time(x,y)
    m=sqrt(2)*x
    n=y
endfunction

//Variable Declaration
x1= 10                        //real value
y1 = 30                       //angle in degree
x2 = 115                      //real value
y2 = -70                      //angle in degree

//Calculation
[a,b]=phasor2time(x1,y1)      //function of conversion
[c,d]=phasor2time(x2,y2)      //function of conversion

//Results
printf("a. I = %.2f sin(377t + %d degree)\n",a,b)
```

```
printf("b. V = %.1f sin(377t %d degree)\n",c,d)
```

Scilab code Exa 14.31 Example 31

```
//Chapter 14, Example 14.31
clc
 funcprot(0)
//time to phasor
function[m,n]=time2phasor(x,y)
    m=0.707*x
    n=y
endfunction

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
//addition
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+y1
    c2 = y2+x2
endfunction

//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction
//phasor to time domain
function[m,n]=phasor2time(x,y)
    m=sqrt(2)*x
    n=y
endfunction
//Variable Declaration
```

```
x1=50                          //maximum voltage of Va
y1=30                          //angle in degree of Va
x2=30                          //maximum voltage of Vb
y2=60                          //angle in degree of Vb
//Calculation
[a,b]=time2phasor(x1,y1)    //function of conversion
[c,d]=time2phasor(x2,y2)    //function of conversion
[e,f]=polar2rect(a,b)    //function of conversion
[g,h]=polar2rect(c,d)    //function of conversion
[i,j]=addition(e,f,g,h)      //function of addition
[k,l]=rect2polar(i,j)    //function of conversion
[o,p]=phasor2time(k,l)    //function of conversion
//Results
printf("ein = %.2f sin(377t + %.2f degree)",o,p)
```

Scilab code Exa 14.32 Example 32

```
//Chapter 14, Example 14.32
clc
 funcprot(0)
function[m,n]=time2phasor(x,y)
    m=0.707*x
    n=y
endfunction

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
//subtraction
function[c1,c2]=subtraction(x1,x2,y1,y2)
    c1 = x1-y1
    c2 = x2-y2
    endfunction
```

```
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=-90
    elseif x<0 then
        th=atand(y/x)+180
    else
        th=atand(y/x)
      if (x<0) & (y<0) then
            th=th-180
        end
    end
endfunction
//phasor to time domain
function[m,n]=phasor2time(x,y)
    m=sqrt(2)*x
    n=y
endfunction
//Variable Declaration
x1= 120*10^-3              //maximum voltage of iT
y1=60                      //angle in degree
x2 = 80*10^-3              //maximum voltage of i1
y2 = 0                     //angle in degree
//Calculation
[a,b]=time2phasor(x1,y1)   //function of conversion
[c,d]=time2phasor(x2,y2)   //function of conversion
[e,f]=polar2rect(a,b)    //function of conversion
[g,h]=polar2rect(c,d)    //function of conversion
[i,j]=subtraction(e,f,g,h)   //subtraction
[k,l]=rect2polar(i,j)    //function of conversion
[o,p]=phasor2time(k,l)    //function of conversion
//Results
printf("i2 = %.1f x 10^-3 sin(wt + %.2f degree) ",o
   *10^3,p)
```

Chapter 15

Series and Parallel ac Circuits

Scilab code Exa 15.1 Example 1

```
//Chapter 15, Example 15.1
clc
funcprot(0)
//time to phasor
function[m,n]=time2phasor(x,y)
    m=0.707*x
    n=y
endfunction

//phasor to time domain
function[m,n]=phasor2time(x,y)
    m=sqrt(2)*x
    n=y
endfunction
//Variable Declaration
x=100                                     //amplitude
y=0
r = 5                                     //resistor
y1=0
//Calculation
[a,b]=time2phasor(x,y)
```

```
i= a/r
[c]=phasor2time(i,y1)

//graph
x1 = 0 : 1 : 1000                    //time in seconds
v = x*sin(377*x1)                    //in voltage
ip = i*sin(377*x1)                   //current in amperes
plot(x1, v)
plot(x1, ip)
xlabel("t")
ylabel("i V")

//Results
printf("i = %.1f sin wt", c)
```

Scilab code Exa 15.2 Example 2

```
//Chapter 15, Example 15.2
clc
//time to phasor
function[m,n]=time2phasor(x,y)
    m=0.707*x
    n=y
endfunction


//phasor to time domain
function[m,n]=phasor2time(x,y)
    m=sqrt(2)*x
    n=y
endfunction
//Variable Declaration
x=4                                  //amplitude
```

```
y=30                                //in degrees
r=2                                 //in ohms

//Calculation
[a,b]=time2phasor(x,y)
c= a*r
[d,e]=phasor2time(c,y)

//graph
x1 = -5 : 1 : 20                    //time in seocnds
v = d*sind((377*x1)+e)              //voltage
ip = x*sind((377*x1)+e)             //current in ampere
plot(x1, v)
plot(x1, ip)
xlabel("t")
ylabel("i V")
//Results
printf("v = %.1f sin(wt + %d degree)", d,e)
```

Scilab code Exa 15.3 Example 3

```
//Chapter 15, Example 15.3
clc
//time to phasor
function[m,n]=time2phasor(x,y)
    m=0.707*x
    n=y
endfunction


//phasor to time domain
function[m,n]=phasor2time(x,y)
    m=sqrt(2)*x
```

```
    n=y
endfunction
//Variable Declaration
x=24                              //amplitude
y=0                             //in degrees
r=3                             //in ohms
d=90                             //in degrees
//Calculation
[a,b]=time2phasor(x,y)
c= a/r
i=d
[f,e]=phasor2time(c,i)

//graph
x1 = 0 : 10 : 500                        //time in
   seocnds
v = x*sind(x1)                  //voltage
ip = f*sind((x1)-d)                 //current in ampere
plot(x1, v)
plot(x1, ip)
xlabel("t")
ylabel("i V")
xtitle("Waveforms for Example 15.3")


//Results
printf("i = %.1f sin(wt - %d degree)", f,e)
```

Scilab code Exa 15.4 Example 4

```
//Chapter 15, Example 15.4
clc
//time to phasor
function[m,n]=time2phasor(x,y)
    m=0.707*x
```

```
    n=y
endfunction


//phasor to time domain
function[m,n]=phasor2time(x,y)
    m=sqrt(2)*x
    n=y
endfunction
//Variable Declaration
x=5                                //amplitude
y=30                               //in degrees
r=4                               //in ohms
d=90                               //in degrees
//Calculation
[a,b]=time2phasor(x,y)
c= a*r
i=b+d
[f,e]=phasor2time(c,i)

//graph
x1 = 0 : 1 : 500                        //time in
   seocnds
v = f*sind((x1)+(d+y))                  //voltage
ip = x*sind((x1)+y)                 //current in ampere
plot(x1, v)
plot(x1, ip)
xlabel("t")
ylabel("i V")

//Results
printf("i = %d sin(wt + %d degree)", f,e)
```

Scilab code Exa 15.5 Example 5

```
//Chapter 15, Example 15.5
clc
//time to phasor
function[m,n]=time2phasor(x,y)
    m=0.707*x
    n=y
endfunction


//phasor to time domain
function[m,n]=phasor2time(x,y)
    m=sqrt(2)*x
    n=y
endfunction
//Variable Declaration
x=15                                //amplitude
y=0                               //in degrees
r=2                               //in ohms
d=-90                               //in degrees
//Calculation
[a,b]=time2phasor(x,y)
c= a/r
i=b-d
[f,e]=phasor2time(c,i)

//graph
x1 = 0 : 1 : 500                          //time in seocnds
v = x*sind(x1)                  //voltage
ip = f*sind(x1-d)                 //current in ampere
plot(x1, v)
plot(x1, ip)
xlabel("t")
ylabel("i V")

//Results
printf("i = %.1f sin(wt + %d degree)", f,e)
```

Scilab code Exa 15.6 Example 6

```
//Chapter 15, Example 15.6
clc
//time to phasor
function[m,n]=time2phasor(x,y)
    m=0.707*x
    n=y
endfunction


//phasor to time domain
function[m,n]=phasor2time(x,y)
    m=sqrt(2)*x
    n=y
endfunction
//Variable Declaration
x=6                                 //amplitude
y=-60                               //in degrees
r=0.5                               //in ohms
d=-90                               //in degrees

//Calculation
[a,b]=time2phasor(x,y)
c= a*r
i=b+d
[f,e]=phasor2time(c,i)


//graph
x1 = 0 : 1 : 500                        //time in
   seocnds
v = f*sind((x1)+(d+y))                  //voltage
ip = x*sind((x1)+ y)                 //current in
```

```
    ampere
plot(x1, v)
plot(x1, ip)
xlabel("t")
ylabel("i V")


//Results
printf("v = %.1f sin(wt%d degree)", f,e)
```

Scilab code Exa 15.7 Example 7

```
//Chapter 15, Example 15.7
clc
//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

//addition
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
    endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction
//Variable Declaration
r = 4                                   //resistance in
    ohms
ro = 0                                  //angle in degree
xl = 8                                  //inductive
```

```
    reactance in ohms
xlo = 90                                //angle in degree
//Calculation
[a,b]=polar2rect(r,ro)                  //Z1
[c,d]=polar2rect(xl,xlo)                //Z2
[e,f]=addition(a,c,b,d)                 //Z1+Z2
[g,h]=rect2polar(e,f)                   //ZT

//Results
printf("ZT = %.3f ohm < %.2f degree",g,h)
```

Scilab code Exa 15.8 Example 8

```
//Chapter 15, Example 15.8
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction


//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

//addition
function[c1,c2]=addition(x1,x2,x3,y1,y2,y3)
    c1 = x1+x2+x3
    c2 = y1+y2+y3
    endfunction

//Variable Declaration
```

```
r=6                                  //resistance in ohms
ro = 0                               //angle in degree
xl = 10                              //in ohms
xlo = 90                             //angle in degree
xc = 12                              //in ohms
xco = -90                            //angle in degree
//Calculation
[a,b]=polar2rect(r,ro)               //Z1
[c,d]=polar2rect(xl,xlo)             //Z2
[e,f]=polar2rect(xc,xco)             //Z3
[g,h]=addition(a,c,e,b,d,f)          //Z1+Z2+Z3
[i,j]=rect2polar(g,h)                //ZT
//Results
printf("ZT = %.3f ohm < %.2f degree",i,j)
```

Scilab code Exa 15.9 Example 9

```
//Chapter 15, Example 15.9
clc
funcprot(0)

function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1-y2
endfunction

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
```

```
    c2 = y1+y2
endfunction

//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction
   //division
   function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction

//Variable Declaration
x1=4                        //real value
x2=100                      //real value
y1=-90                      //angle in degree
y2=0                        //angle in degree
r=3                         //real value
x3=3                        //real value
x4=100                      //real value
y3=0                        //angle in degree
y4=0                        //angle in degree
//Calculation
[a,b]=voltdivider(x1,x2,y1,y2)    //function of multipliation
[c,d]=polar2rect(x1,y1)       //function of conversion
[e,f]=polar2rect(r,y2)       //function of conversion
[i,j]=addition(e,f,c,d)      //function of addition
[o,p]=rect2polar(i,j)       //function of conversion
[m,n]=division(a,o,b,p)     //function of division
[aa,ba]=voltdivider(x3,x4,y3,y4)  //function of multipliation
[ma,na]=division(aa,o,ba,p)     //function of division

```

```
//Results
printf("Vc = %d V < %.2f degree \n",m,n)
printf("Vr = %d V < %.2f degree",ma,na)
```

Scilab code Exa 15.10 `Example 10`

```
//Chapter 15, Example 15.10
clc
funcprot(0)
//addition
function[c1,c2]=addition1(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
    endfunction
//multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
//addition
function[c1,c2,c3]=addition(x1,x2,x3,y1,y2,y3)
    c1 = x1+x2+x3
    c2 = y1+y2+y3
endfunction

//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
```

```
        th=-90
    else
        th=atand(y/x)
    end
endfunction


   function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction

//Variable Declaration
x1=6                           //real value of Zr
x2=50                           //real value of E
y1=0                            //angle in degree of Zr
y2=30                            //angle in degree of E
r2=9                           //real value of Zl
s2=90                            //angle in degree of Zl
r3=17                           //real value of Zc
s3=-90                            //angle in degree of
   Zc
//Calculation
[a,b]=voltdivider(x1,x2,y1,y2)   //function of
   multiplication
[c,d]=polar2rect(x1,y1)        //function of
   conversion
[e,f]=polar2rect(r2,s2)        //function of
   conversion
[ee,ff]=polar2rect(r3,s3)        //function of
   conversion
[i,j]=addition(c,e,ee,d,f,ff)      //function of
   addition
[o,p]=rect2polar(i,j)         //function of
   conversion
[m,n]=division(a,o,b,p)       //function of division
[aa,ba]=voltdivider(r2,x2,s2,y2)         //function
   of conversion
```

```
[ma,na]=division(aa,o,ba,p)        //function of division
[ab,bb]=voltdivider(r3,x2,s3,y2)          //function of conversion
[mb,nb]=division(ab,o,bb,p)        //function of division
[za,ya]=addition1(e,ee,f,ff)       //function of addition
[zb,yb]=rect2polar(za,ya)          //function of conversion
[zc,yc]=voltdivider(zb,x2,yb,y2)      //function of multiplication
[zd,yd]=division(zc,o,yc,p)         //function of division

//Results
printf("Vr = %d V < %.2f degree \n",m,n)
printf("Vl = %d V < %.2f degree \n",ma,na)
printf("Vc = %d V < %.2f degree \n",mb,nb)
printf("V1 = %d V < %.2f degree",zd,yd)
```

Scilab code Exa 15.11 Example 11

```
//Chapter 15, Example 15.11
clc
funcprot(0)
//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

//addition
function[c1,c2]=addition(x1,x2,x3,y1,y2,y3)
    c1 = x1+x2+x3
```

```
    c2 = y1+y2+y3
endfunction

//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction
//division
function[q,x]=division(x1,x2,y1,y2)
        q=x1/x2
        x=y1-y2
   endfunction

   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
//Variable Declaration
r1=6                          //resistance in ohms
r2=4                          //resistance in ohms
l1 = 0.05                     //inductance in henry
l2 = 0.05                     //inductance in henry
c1 = 200*10^-6                //capacitance in farad
c2 = 200*10^-6                //capacitance in farad
w = 377                       //in rad/s
z = 20                        //source in volts
//Calculation
rt = r1+r2              //total resistance in ohms
yt = 0                  //angle in degree
lt = l1+l2              //in henry
ct = c1/2               //in farad
xl = w*lt               //in ohms
xy = 90                 //angle in degree
xc = 1/(w*ct)           //in ohms
cy = -90                //angle in degree
[a,b]=polar2rect(rt,yt)        //function of
```

```
    conversion
[c,d]=polar2rect(xl,xy)        //function of
    conversion
[e,f]=polar2rect(xc,cy)        //function of
    conversion
[i,j]=addition(a,c,e,b,d,f)    //function of addition
[m,n]=rect2polar(i,j)        //function of conversion
[o,p]=division(z,m,yt,n)     //function of division
[w,ww]=voltdivider(o,rt,p,yt)     //function of
    multiplication
[x,y]=voltdivider(o,xl,p,xy)      //function of
    multiplication
[v,k]=voltdivider(o,xc,p,cy)      //function of
    multiplication
fp1 = cosd(48.16)           //in degree
fp2 = (rt/m)                //lagging
pt= z*o*fp1                  //power in watt
q=x-v                        //in volts
e= sqrt((w*w)+(q*q))         //in volts
eo = 0                       //in volts
[aa,bb]=voltdivider(rt,e,yt,eo)     //function of
    multiplication
[cc,dd]=division(aa,m,bb,n)        //function of
    division
[ee,ff]=voltdivider(xc,e,cy,eo)     //function of
    multiplication
[ii,jj]=division(ee,m,ff,n)       //function of
    division
//Results
printf("a. Zt = %d ohm < %.2f degree \n",m,n)
printf("    I = %.3f A < %.2f degree \n",o,p)
printf("    Vr = %.2f V < %.2f degree \n",w,ww)
printf("    Vl = %.2f V < %.2f degree \n",x,y)
printf("    Vc = %.2f V < %.2f degree \n",v,k)
printf("b. Fp = %.3f lagging \n",fp1)
printf("    Fp = %.3f lagging \n",fp2)
printf("c. Pt = %.2f W \n",pt)
printf("e. E = %d V \n",e)
```

```
printf("f. Vr = %.1f V < %.2f degree \n",cc,dd)
printf("   Vc = %.2f V < %.2f degree",ii,jj)
```

Scilab code Exa 15.12 Example 12

```
//Chapter 15, Example 15.12
clc
funcprot(0)
//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
//adition
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
endfunction

//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction
//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction

      //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
```

```
//Function to create plot on complex plane
function complexPlot(x1,y1,x2,y2,x3,y3,z)
    xpts1 = [0 x1];
    ypts1 = [0 y1];

    xpts2 = [0 x2];
    ypts2 = [0 y2];

    xpts3 = [0 x3];
    ypts3 = [0 y3];
    scf(1);
    clf(1);
    set(gca(),"auto_clear","off")
    plot(xpts1,ypts1,'ro-')
    plot(xpts2,ypts2,'ro-')
    plot(xpts3,ypts3,'ro-')
    set(gca(),"auto_clear","on")
    a = gca();
    a.x_location="origin";
    a.y_location="origin";
    set(gca(),"data_bounds",matrix
       ([-0.1,0.1,-0.2,0.2],2,-1))
    xtitle(z);
    xlabel("Real");
    ylabel("Imaginary");
endfunction

//Variable Declaration
g = 0.05                    //in admittance
d = 0                        //in degrees
bl = 0.1                    //in admittance
dl = -90                        //in degrees
zr= 20                          //in impedance
zro = 0                          //in angle
zl = 10                          //in impedance
zlo = 90                          //in degrees

```

```
//Calculation
[a,b]=polar2rect(g,d)
[c,d]=polar2rect(bl,dl)
[e,f]=addition(a,c,b,d)
[i,j]=rect2polar(e,f)
m= (1/i)
n= (-j)
[o,p]=voltdivider(zr,zl,zro,zlo)
[q,r]=polar2rect(zr,zro)
[s,t]=polar2rect(zl,zlo)
[oo,pp]=addition(q,s,r,t)
[qq,rr]=rect2polar(oo,pp)
[ss,tt]=division(o,qq,p,rr)

//Plot
complexPlot(a,b,c,d,e,f,"Admittance Diagram")

//Results
printf("Yr = %.2f S + j%d \n",a,b)
printf("Yl = %d %.1fj s \n",c,d)
printf("Yt = %.2f S %.1fj S \n",e,f)
printf("Zt = %.2f ohm < %.2f degree \n",m,n)
printf("Zt = %.2f ohm < %.2f degree \n",ss,tt)
```

Scilab code Exa 15.13 Example 13

```
//Chapter 15, Example 15.13
clc
funcprot(0)
//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
//adition
```

```
function[c1,c2]=addition(x1,x2,x3,y1,y2,y3)
    c1 = x1+x2+x3
    c2 = y1+y2+y3
endfunction

//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction
//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction

      //multiplication
function [r,s] = voltdivider(x1,x2,x3,y1,y2,y3)
    r=x1*x2*x3
    s=y1+y2+y3
endfunction

  //multiplication
function [r,s] = voltdivider1(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction


//Function to create plot on complex plane
function complexPlot(x1,y1,x2,y2,x3,y3,x4,y4,z)
    xpts1 = [0 x1];
    ypts1 = [0 y1];

    xpts2 = [0 x2];
    ypts2 = [0 y2];

    xpts3 = [0 x3];
```

```
    ypts3 = [0 y3];

    xpts4 = [0 x4];
    ypts4 = [0 y4];
    scf(1);
    clf(1);
    set(gca(),"auto_clear","off")
    plot(xpts1,ypts1,'ro-')
    plot(xpts2,ypts2,'ro-')
    plot(xpts3,ypts3,'ro-')
    plot(xpts4,ypts4,'ro-')
    set(gca(),"auto_clear","on")
    a = gca();
    a.x_location="origin";
    a.y_location="origin";
    set(gca(),"data_bounds",matrix
       ([-0.1,0.25,-0.2,0.2],2,-1))
    xtitle(z);
    xlabel("Real");
    ylabel("Imaginary");
endfunction

//Variable Declaration
r = 5
g = 1/r
xl = 8
d = 0
bl = 1/xl
dl = -90
xc = 20
bc = 1/xc
dc = 90
zr= 5
zro = 0
zl = 8
zlo = 90
zc = 20
zco = -90
```

```
//Calculation
[a,b]=polar2rect(g,d)
[c,d]=polar2rect(bl,dl)
[cc,dd]=polar2rect(bc,dc)
[e,f]=addition(a,c,cc,b,d,dd)
[i,j]=rect2polar(e,f)
m= (1/i)
n= (-j)
[o,p]=voltdivider(zr,zl,zc,zro,zlo,zco)
[mm,nn]=voltdivider1(zr,zl,zro,zlo)
[ma,na]=voltdivider1(zl,zc,zlo,zco)
[mb,nb]=voltdivider1(zr,zc,zro,zco)
[q,r]=polar2rect(mm,nn)
[s,t]=polar2rect(ma,na)
[sa,ta]=polar2rect(mb,nb)
[oo,pp]=addition(q,s,sa,r,t,ta)
[qq,rr]=rect2polar(oo,pp)
[ss,tt]=division(o,qq,p,rr)

//Plot
complexPlot(a,b,c,d,cc,dd,e,f,"Admittance Diagram")


//Results
printf("a. Yr = %.1f S + j%d \n",a,b)
printf("    Yl = %d %.3fj  s \n",c,d)
printf("    Yc = %d + j%.3f S \n",cc,dd)
printf("b. Yt = %.2f S < %.3f degree \n",i,j)
printf("c. Zt = %.2f ohm < %.2f degree \n",m,n)
printf("    Zt = %.2f ohm < %.2f degree \n",ss,tt)
```

Scilab code Exa 15.14 Example 14

```
//Chapter 15, Example 15.14
clc
```

```
//real and imaginary part
function[r,i]=realnimaginary(x,y)
    r= x/((x*x)+(y*y))
    i= y/((x*x)+(y*y))
endfunction


//Variable Declaration
r = 6                  //real part of Z
xc = 8                 //imaginary part of Z
r1 = 10                //real part of Z
xl = 4                 //imaginary part of Z
xc1 = -0.1             //imaginary part of Z

//Calculation
[a,b]=realnimaginary(r,xc)        //function
xo = xl+xc1                     //imaginary part of Z
[c,d]=realnimaginary(r1,xo)      //function
//Results
printf("a. Y = %.2f S + j%.2f S \n",a,b)
printf("b. Y = %.3f S - j%.3f S ",c,d)
```

Scilab code Exa 15.15 Example 15

```
//Chapter 15, Example 15.15
clc
  //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
```

```
    th=atand(y/x)
endfunction

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
//Variable Declaration
zl = 4                        //real value of Zl
zlo = 90                      //angle in degree
it = 20                        //real value of It
ito = 0                       //angle in degree
zr = 3                        //real value of Zr
zro = 0                      //angle in degree

//Calculation
[a,b]=voltdivider(zl,it,zlo,ito)  //function of
    multiplication
[c,d]=polar2rect(zr,zro)         //function of
    conversion
[e,f]=polar2rect(zl,zlo)         //function of
    conversion
[i,j]=addition(c,d,e,f)        //function of addition
[o,p]=rect2polar(i,j)          //function of
    conversion
[m,n]=division(a,o,b,p)       //function of division
```

```
[aa,bb]=voltdivider(zr,it,zro,ito)  //function of multiplication
[mm,nn]=division(aa,o,bb,p)      //function of division
//Results
printf("Ir = %d A < %.2f degree \n",m,n)
printf("It = %d A < %.2f degree \n",mm,nn)
```

Scilab code Exa 15.16 Example 16

```
//Chapter 15, Example 15.16
clc
//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction

   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
//Variable Declaration
zc= 2                   //real value of Zc
zco = -90               //angle in degree
```

```
it = 5                   //real value of It
itt = 30               //angle in degree
zr = 1               //real part of Zrl
zl = 8               //imaginary part of Zl
xx = zl-zc           //imaginary part of Zcrl
//Calculation
[a,b]=voltdivider(zc,it,zco,itt)  //function of multiplication
[c,d]=rect2polar(zr,xx)          //function of conversion
[e,f]=division(a,c,b,d)        //function of division
[m,n]=rect2polar(zr,zl)          //function of conversion
[i,j]=voltdivider(m,it,n,itt)  //function of multiplication
[s,t]=division(i,c,j,d)        //function of division
//Results
printf("Irl = %.3f A < %.2f degree \n",e,f)
printf("Ic = %.3f A < %.2f degree",s,t)
```

Scilab code Exa 15.17 Example 17

```
//Chapter 15, Example 15.17
clc
//Variable Declaration
rp = 8*10^3                      //R in kiloohms
xl = 9*10^3                      //Xl in kiloohms
xc = 4*10^3                      //Xc in kiloohms


//Calculation
xp = xl-xc                       //in kiloohms
rs = (rp*(xp*xp))/((xp*xp)+(rp*rp))     //in kiloohms
xs = ((rp*rp)*xp)/((xp*xp)+(rp*rp))     //in kiloohms

```

```
//Results
printf("Rs = %.3f kohm \n",rs/10^3)
printf("Xs = %.3f kohm ",xs/10^3)
```

Scilab code Exa 15.18 Example 18

```
//Chapter 15, Example 15.18
clc
funcprot(0)
//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
function[c1,c2]=addition(x1,x2,x3,y1,y2,y3)
    c1 = x1+x2+x3
    c2 = y1+y2+y3
    endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction

      //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
```

```
function[c1,c2]=addition1(x1,x2,y1,y2)
     c1 = x1+x2
     c2 = y1+y2
     endfunction
//Variable Declaration
r1 = 10                         //resistance1 in ohms
r2 = 40                         //resistance2 in ohms
l1 = 6*10^-3                    //inductance1 in millihenry
l2 = 12*10^-3                    //inductance2 in millihenry
c1 = 80*10^-6                    //capacitance1 in microfarad
c2 = 20*10^-6                    //capacitance1 in microfarad
w = 1000                         //in rad/s
ia= 12                           //current source in ampere
iao = 0                          //angle in degree
//Calculation
rt = (r1*r2)/(r1+r2)             //total resistance in ohms
lt = (l1*l2)/(l1+l2)             //total inductance in millihenry
ct = c1+c2                       //total capacitance in microfarad
xl = w*lt                       //reactance in ohms
xlo = 90                        //angle in degree
xc = 1/(w*ct)                   //reactance in ohms
g = 1/rt                        //real value of Yr
go = 0                          //angle in degree
bl = 1/xl                       //real value of Bl
blo = -90                        //angle in degree
bc = 1/xc                       //real value of Bc
bco = 90                        //angle in degree
[a,b]=polar2rect(g,go)           //function of conversion
[c,d]=polar2rect(bl,blo)           //function of conversion
```

```
[e,f]=polar2rect(bc,bco)     //function of conversion
[i,j]=addition(a,c,e,b,d,f)  //function of addition
[m,n]=rect2polar(i,j)     //function of conversion
[mm,nn]=division(ia,m,iao,n)  //function of division
[o,p]=division(mm,xl,nn,xlo)  //function of division
fp = (g/m)                    //lagging
po = mm*ia*cosd(nn)           //power in watt
zt1 = 1/m                     //real vlue in ohms of
    Zt
zt2 = -n                      //angle in degree
[vv,ww]=polar2rect(zt1,zt2)    //function of
   conversion
l=vv/w                         //inductance in
   millihenry
[ss,tt]=voltdivider(ia,zt1,iao,zt2)    //function of
    multiplication
ppo= (ia*ia)*vv               //power in watt
rp = ((vv*vv)+(ww*ww))/vv     //resistance in ohms
xp = ((vv*vv)+(ww*ww))/ww     //resistance in ohms
g1=1/rp                      //real value in ohms of
    Yt
g1o=0                        //angle in degree
bl1=1/xp                     //real value
bl1o=-90                        //angle in degree
[za,zb]=polar2rect(g1,go)    //function of
   conversion
[zc,zd]=polar2rect(bl1,bl1o)    //function of
   conversion
[ze,zf]=addition1(za,zc,zb,zd)  //function of
   addition
[zx,zy]=rect2polar(ze,zf)    //function of
   conversion
//Results
printf("Yt = %.3f S < %.3f degree \n",m,n)
printf("E  = %.3f V < %.3f degree \n",mm,nn)
printf("IL = %.3f A < %.3f degree \n",o,p)
printf("Fp = %.3f lagging \n",fp)
printf("P = %.2f W \n",po)
```

```
printf("Zt = %.3f ohm + j%.3f ohm \n",vv,ww)
printf("L = %.3f mH \n",l*10^3)
printf("E = %.3f V < %.3f degree \n",ss,tt)
printf("P = %.2f W \n",ppo)
printf("Rp = %d ohm \n",rp)
printf("Xp = %.3f ohm \n",xp)
printf("Yt = %.3f S < %.3f degree \n",zx,zy)
```

Chapter 16

Series Parallel ac Networks

Scilab code Exa 16.1 Example 1

```
//Chapter 16, Example 16.1
clc

   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction
//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
```

```
        x=y1-y2
    endfunction
    //addition
   function[c1,c2]=addition(x1,x2,y1,y2)
     c1 = x1+x2
     c2 = y1+y2
     endfunction
//Variable Declaration
z1 = 1          //real value of impedance1
z1o = 0         //angle in degree
xc = 2          //real value of Xc
xco = -90         //angle in degree
xl = 3          //real value of Xl
xlo = 90         //angle in degree
ev = 120          //real value of E
eo = 0         //angle in degree


//Calculation
[a,b]=voltdivider(xc,xl,xco,xlo)  //function of
    multiplication
x= -xc+xl                            //real value
xo= 90                              //angle in degree
[c,d]=division(a,x,b,xo)    //function of division
[aa,bb]=polar2rect(z1,z1o)  //function of conversion
[cc,dd]=polar2rect(c,d)  //function of conversion
[e,f]=addition(aa,cc,bb,dd)   //function of addition
[ee,ff]=rect2polar(e,f)  //function of conversion
[m,n]=division(ev,ee,eo,ff)    //function of
    division
[v,w]=voltdivider(m,z1,n,z1o)  //function of
    multiplication
[vv,ww]=voltdivider(m,c,n,d)  //function of
    multiplication
[q,r]=division(vv,xc,ww,xco)    //function of
    division
pdel = (m*m)*z1              //power in watt
fp = cosd(n)                //lagging
```

```
//Results
printf("a. Zt = %.2f ohm < %.2f degree \n",ee,ff)
printf("b. Is = %.2f A < %.2f degree \n",m,n)
printf("c. Vr = %.2f V < %.2f degree \n",v,w)
printf("   Vc = %.2f V < %.2f degree \n",vv,ww)
printf("d. Ic = %.2f A < %.2f degree \n",q,r)
printf("e. Pdel = %.2f W \n",pdel)
printf("f. Fp = %.3f leading \n",fp)
```

Scilab code Exa 16.2 Example 2

```
//Chapter 16, Example 16.2
clc
//multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=-90
    else
        th=atand(y/x)
    end
endfunction
//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
//addition
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
```

```
    c2 = y1+y2
endfunction
//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction

//Variable Declaration
x=complex(3,4)       //complex form of Z1
y=complex(0,-8)       //complex form of Z2
i = 50                //real value of current in ampere
io = 30               //angle in degree
//Calculation

[a,b]=rect2polar(real(x),imag(x))     //function of conversion
[c,d]=rect2polar(real(y),imag(y))     //function of conversion
[aa,bb]=voltdivider(c,i,d,io)  //function og multiplication
[cc,dd]=addition(real(y),real(x),imag(y),imag(x))
[e,f]=rect2polar(cc,dd)     //function of conversion
[ee,ff]=division(aa,e,bb,f)   //function of division
[ac,bc]=voltdivider(a,i,b,io)  //function og multiplication
[ea,fa]=division(ac,e,bc,f)   //function of division
[m,n]=polar2rect(ee,ff)     //function of conversion
[mm,nn]=polar2rect(ea,fa)     //function of conversion
[ma,na]=addition(m,mm,n,nn)    //function of addition
[mb,nb]=rect2polar(ma,na)     //function of conversion
//Results
printf("a. I1 = %d A < %.2f degree \n",ee,ff)
printf("b. I2 = %d A < %.2f degree \n",ea,fa)
printf("c. %d A < %d degree = %d A < %d degree (
```

```
checks) ",mb,nb,mb,nb)
```

Scilab code Exa 16.3 Example 3

```
//Chapter 16, Example 16.3
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=-90
    else
        th=atand(y/x)
    end
endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar1(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

//addition
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
```

```
    endfunction

  //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
//Variable Declaration
z1=complex(5,0)          //complex form of Z1
z2= complex(0,-12)          //complex form of Z2
z3 = complex(0,8)          //complex form of Z3
e1 = 20                       //real part of voltage
e1o = 20                      //angle in degree

//Calculation
[a,b]=rect2polar(real(z2),imag(z2))     //function of
    conversion
[aa,bb]=voltdivider(a,e1,b,e1o)      //function of
   multiplication
[c,d]=addition(real(z1),real(z2),imag(z1),imag(z2))
      //function of addition
[cc,dd]=rect2polar(c,d)     //function of conversion
[e,f]=division(aa,cc,bb,dd)     //function of
   division
[ee,ff]=rect2polar1(real(z3),imag(z3))     //function
    of conversion
[g,h]=division(e1,ee,e1o,ff)     //function of
   division
[gg,hh]=division(e1,cc,e1o,dd)     //function of
   division
[o,p]=polar2rect(g,h)     //function of conversion
[oo,pp]=polar2rect(gg,hh)     //function of
```

```
    conversion
[m,n]=addition(o,oo,p,pp)    //function of addition
[mm,nn]=rect2polar(m,n)    //function of conversion
//Results
printf("a. Vc = %.2f V < %.2f degree \n",e,f)
printf("b. I1 = %.1f A < %.2f degree \n",g,h)
printf("   I2 = %.2f A < %.2f degree \n",gg,hh)
printf("   Is = %.2f A < %.2f degree \n",mm,nn)
```

Scilab code Exa 16.4 Example 4

```
//Chapter 16, Example 16.4
clc

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction
//subtraction
function[c1,c2]=subtraction(x1,x2,y1,y2)
    c1 = x1-x2
    c2 = y1-y2
    endfunction
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
function[c1,c2]=addition(x1,x2,y1,y2)
```

```
    c1 = x1+x2
    c2 = y1+y2
    endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
//Variable Declaration
x=complex(3,4)      //complex form of Z1
y=complex(8,-6)     //complex form of Z2
ea = 100            //source in volts
eo = 0              //angle in degree
xo = 90
//Calculation
[a,b]=rect2polar(real(x),imag(x))   //function of conversion
[c,d]=rect2polar(real(y),imag(y))   //function of conversion
[e,f]=voltdivider(a,c,b,d)   //function of conversion
[g,h]=addition(real(x),real(y),imag(x),imag(y))
[gg,hh]=rect2polar(g,h)   //function of conversion
[i,j]=division(e,gg,f,hh)  //function of division
[k,l]=division(ea,i,eo,j)  //function of division
[m,n]=division(ea,a,eo,b)  //function of division
[mm,nn]=division(ea,c,eo,d)  //function of division
[o,p]=voltdivider(m,real(x),n,eo)  //function of multiplication
[oo,pp]=voltdivider(mm,real(y),nn,eo)  //function of multiplication
[q,r]=voltdivider(real(x),ea,eo,eo)  //function of multiplication
[s,t]=polar2rect(real(x),eo)   //function of conversion
[ss,tt]=polar2rect(imag(x),xo)   //function of conversion
```

```
[sa,ta]=addition(s,ss,t,tt)      //function of
    addition
[sb,tb]=rect2polar(sa,ta)
[qa,ra]=division(q,sb,r,tb)      //function of
    division
[v,w]=polar2rect(oo,pp)    //function of conversion
[vv,ww]=polar2rect(o,p)    //function of conversion
[va,wa]=subtraction(v,vv,w,ww)   //function of
    subtraction
[vb,wb]=rect2polar(va,wa)    //function of conversion
//Results
printf("Zt = %.3f ohm < %.2f degree \n",i,j)
printf("Is = %.2f A < %.2f degree \n",k,l)
printf("I1 = %d A < %.2f degree \n",m,n)
printf("I2 = %d A < %.2f degree \n",mm,nn)
printf("Vr1 = %d V < %.2f degree \n",o,p)
printf("Vr2 = %d V < %.2f degree \n",oo,pp)
printf("Vr1 = %d V < %.2f degree \n",qa,ra)
printf("Vab = %d V < %.2f degree \n",vb,wb)
```

Scilab code Exa 16.5 Example 5

```
//Chapter 16, Example 16.5
clc

   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
endfunction
//Rectangle to Polar Conversion
```

```
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
//Variable Declaration
ra = 50*10^3          //real value of Ra
rao = 0               //angle in degree
rb = 3.3*10^3         //real value of Rb
rbo = 0               //angle in degree
x=complex(1*10^3,-159.16)    //complex form of Z2
ia = 4*10^-3                 //real value of current in milliampere
iao = 0               //angle in degree
y=complex(1*10^3,-0.796)     //complex form of Z2

//Calculation
z1 = (ra*rb)/(ra+rb)           //Z1 in kilo ohms
[a,b]=voltdivider(z1,ia,rao,iao)  //function of mltiplicaion
[c,d]=addition(z1,real(x),rao,imag(x))
[g,h]=rect2polar(c,d)        //function of conversion
[j,k]=division(a,g,b,h)    //function of division
[m,n]=voltdivider(j,real(x),k,rao)  //function of mltiplicaion
[cc,dd]=addition(z1,real(y),rao,imag(y))
[gg,hh]=rect2polar(cc,dd)
[jj,kk]=division(a,gg,b,hh)    //function of division
[mm,nn]=voltdivider(jj,real(y),kk,rao)  //function of mltiplicaion


//Results
```

```
printf("a. VL = %.3f V < %.3f degree \n",m,n)
printf("b. VL = %.3f V < %.3f degree \n",mm,nn)
```

Scilab code Exa 16.6 Example 6

```
//Chapter 16, Example 16.6
clc
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
    endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
   //Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction

//Variable Declaration
it = 2.626*10^-3     //real value of current in
   milliampere
ito = 51.02          //angle in degree
z1 = 1.545*10^3      //real value of Z1
z1o = 0              //angle in degree
x=complex(10*10^3,-20*10^3)    //complex form of Z2
```

```
//Calculation
[a,b]=voltdivider(z1,it,z1o,ito)   //function of
   multiplication
[c,d]=addition(z1,real(x),z1o,imag(x))  //function
   of addition
[e,f]=rect2polar(c,d)   //function of conversion
[m,n]=division(a,e,b,f)  //function of division
[o,p]=rect2polar(real(x),imag(x))   //function of
   conversion
[mm,nn]=voltdivider(m,o,n,p)   //function of
   multiplication



//Results
printf("a. I = %.3f mA < %.3f degree \n",m*10^3,n)
printf("b. V = %.3f V < %.3f degree \n",mm,nn)
```

Scilab code Exa 16.7 Example 7

```
//Chapter 16, Example 16.7
clc
//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction
function[c1,c2]=addition(x1,x2,x3,y1,y2,y3)
    c1 = x1+x2+x3
```

```
    c2 = y1+y2+y3
endfunction
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction


//Variable Declaration
z1 = 10                     //real value of Z1
z1o = 0                     //angle in degree
z2 = complex(3,4)           //complex form of Z2
z3 = complex(8,-6)          //complex form of Z3
ea = 200                    //source voltage in volts
eo = 0                      //angle in degree
//Calculation
[a,b]=rect2polar(real(z2),imag(z2))   //function of conversion
[c,d]=rect2polar(real(z3),imag(z3))   //function of conversion
y1=1/z1                        //admittance1
y1o = 0                        //angle in degree
y2 = 1/a                        //admittance2
ya2 = -b                        //angle in degree
y3 = 1/c                        //admittance3
ya3 = -d                        //angle in degree
[e,f]=polar2rect(y2,ya2)   //function of conversion
[g,h]=polar2rect(y3,ya3)   //function of conversion
[m,n]=addition(y1,e,g,y1o,f,h)  //function of addition
[o,p]=rect2polar(m,n)   //function of conversion
```

```
[s,t]=voltdivider(ea,o,eo,p)      //function of multiplication
[sa,ta]=division(ea,z1,eo,z1o)    //function of division
[sb,tb]=division(ea,a,eo,b)    //function of division
[sc,tc]=division(ea,c,eo,d)    //function of division
zt1 = 1/o                   //real value of Zt
zt2 = -p                     //angle in degree
[v,w]=polar2rect(s,t)   //function of conversion
[va,wa]=polar2rect(sa,ta)   //function of conversion
[vb,wb]=polar2rect(sb,tb)   //function of conversion
[vc,wc]=polar2rect(sc,tc)   //function of conversion
[vv,ww]=addition(va,vb,vc,wa,wb,wc)   //function of addition
//Results
printf("a. I = %.1f A < %.3f degree \n",s,t)
printf("b. I1 = %d A < %d degree \n",sa,ta)
printf("   I2 = %d A < %.2f degree \n",sb,tb)
printf("   I3 = %d A < %.2f degree \n",sc,tc)
printf("c. %d %dj = %d %dj (checks)\n",v,w,vv,ww)
printf("d. Zt = %.3f ohm < %.3f degree \n",zt1,zt2)
```

Scilab code Exa 16.8 Example 8

```
//Chapter 16, Example 16.8
clc
funcprot(0)
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
```

```
    s=y1+y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
endfunction

function[c1,c2]=subtraction(x1,x2,y1,y2)
    c1 = x1-x2
    c2 = y1-y2
    endfunction
//Variable Declaration
z1 = 4                         //real value of Z1
z1o = 0                        //angle in degree
z2=complex(9,-7)               //complex form of Z2
z3=complex(8,6)               //complex form of Z3
ev = 100                     //source voltage in volts
eo = 0                       //angle in degree

//Calculation
[a,b]=rect2polar(real(z2),imag(z2))       //function
   of conversion
[c,d]=rect2polar(real(z3),imag(z3))       //function
   of conversion
[e,f]=voltdivider(a,c,b,d)   //function of
```

```
multiplication
[ee,ff]=addition(real(z2),real(z3),imag(z2),imag(z3)
)
[m,n]=rect2polar(ee,ff)       //function of
conversion
[ea,fa]=division(e,m,f,n)     //function of division
[eb,fb]=polar2rect(ea,fa)       //function of
conversion
[g,h]=addition(z1,eb,z1o,fb)    //function of
addition
[gg,hh]=rect2polar(g,h)       //function of
conversion
[o,p]=division(ev,gg,eo,hh)     //function of
division
fp=gg/gg
[i,j]=voltdivider(a,o,b,p)    //function of
multiplication
[q,r]=division(i,m,j,n)     //function of division
[qa,ra]=polar2rect(q,r)       //function of
conversion
[qb,rb]=polar2rect(o,p)       //function of
conversion
[s,t]=subtraction(qb,qa,rb,ra)   //function of
subtraction
[ss,tt]=rect2polar(s,t)       //function of
conversion
pt = ev*o*cosd(1.5)          //power in watt
//Results
printf("a. Zt = %.3f ohm < %.1f degree \n",gg,hh)
printf("b. I = %.2f A < %.1f degree \n",o,p)
printf("c. Fp = %d \n",fp)
printf("I2 = %.2f A < %d degree \n",q,r)
printf("I1 = %.1f A < %.2f degree \n",ss,tt)
printf("Pt = %.2f W \n",pt)
```

Chapter 17

Methods of Analysis and Selected Topics ac

Scilab code Exa 17.1 Example 1

```
//Chapter 17, Example 17.1
clc
//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
//Variable Declaration
ea = 100          //source voltage in volts
eo = 0            //angle in degree
za = 5            //real value of Z
zo = 53.13        //angle in degree
//Calculation
[a,b]=division(ea,za,eo,zo)   //function of division



//Results
printf("I = %d A < %.2f degree \n",a,b)
```

Scilab code Exa 17.2 Example 2

```
//Chapter 17, Example 17.2
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
```

```
    endfunction
//Variable Declaration
zc = 4                    //real value of Zc
zco = -90                   //angle in degree
zl = 6                    //real value of Zl
zlo = 90                   //angle in degree
ia = 10                    //real value of I
iao = 60                   //angle in degree

//Calculation
[aa,bb]=voltdivider(zc,zl,zco,zlo)  //function of multiplication
[a,b]=polar2rect(zc,zco)         //function of conversion
[c,d]=polar2rect(zl,zlo)         //function of conversion
[e,f]=addition(a,c,b,d)      //function of addition
[ee,ff]=rect2polar(e,f)         //function of conversion
[g,h]=division(aa,ee,bb,ff)     //function of division
[i,j]=voltdivider(ia,g,iao,h)   //function of multiplication

//Results
printf("Z = %d V < %d degree \n",g,h)
printf("E = %d V < %d degree \n",i,j)
```

Scilab code Exa 17.3 Example 3

```
//Chapter 17, Example 17.3
clc

//division
function[q,x]=division(x1,x2,y1,y2)
```

```
        q=x1/x2
        x=y1-y2
    endfunction
//Variable Declaration
ea = 20                          //source voltage in volts
eo = 0                           //angle in degree
z1 = 5*10^3                      //real value of Z
z1o = 0                          //angle in degree


//Calculation
[a,b]=division(ea,z1,eo,z1o)     //function of
    division



//Results
printf("I =(%d X 10^-3 V)A < %d degree",a*10^3,b)
```

Scilab code Exa 17.4 Example 4

```
//Chapter 17, Example 17.4
clc
    //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
     r=x1*x2
     s=y1+y2
endfunction
//Variable Declaration
ia= 100                     //real value of I
iao = 0                     //angle in degree
z1 = 40*10^3          //real value of Z
z1o = 0                     //angle in degree


```

```
//Calculation
[a,b]=voltdivider(ia,z1,iao,z1o)


//Results
printf("E = (%d x 10^6I)V < %d degree",a/10^6,b)
```

Scilab code Exa 17.5 Example 5

```
//Chapter 17, Example 17.5
clc
funcprot(0)
//addition
function[c1,c2]=addition(x1,x2,x3,y1,y2,y3)
    c1 = x1+x2+x3
    c2 = y1+y2+y3
    endfunction
//subtaction
function[c1,c2]=subtraction(x1,x2,y1,y2)
    c1 = x1-x2
    c2 = y1-y2
    endfunction
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=-90
    else
        th=atand(y/x)
      if (x<0) & (y<0) then
```

```
            th=th-180
        end
    end
endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar1(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
      if (x<0) & (y<0) then
            th=th-180
        end
    end
endfunction
//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
//Variable Declaration
z1 = complex(0,2)     //complex form of Z1
z2 = complex(4,0)     //complex form of Z2
z3 = complex(0,-1)     //complex form of Z3
e1 = 2                  //source voltage in volts
e1o = 0                 //angle in degree
e2 = 6                  //source voltage in volts
e2o = 0                 //angle in degree

//Calculation
a= ((e1-e2)*real(z2))     //real part of numerator
```

```
b=e1*imag(z3)                //imaginary part of numerator
[c,d]=rect2polar(a,b)          //function of conversion

[e,f]=rect2polar1(real(z1),imag(z1))     //z1 function of conversion
[g,h]=rect2polar(real(z2),imag(z2))     //z2 function of conversion
[i,j]=rect2polar(real(z3),imag(z3))     //z3 function of conversion
[ca,da]=voltdivider(e,g,f,h)           //z1*z2 function of multiplication
[cb,db]=voltdivider(e,i,f,j)           //z1*z3 function of multiplication
[cc,dd]=voltdivider(g,i,h,j)           //z2*z3 function of multiplication
[s,t]=polar2rect(ca,da)               //function of conversion
[sa,ta]=polar2rect(cb,db)             //function of conversion
[sb,tb]=polar2rect(cc,dd)             //function of conversion

[o,p]=addition(s,sa,sb,t,ta,tb)       //function of addition
[m,n]=rect2polar(o,p)                //function of conversion
[mm,nn]=division(c,m,d,n)          //function of division
//Results
printf("I1 = %.2f A < %.2f degree \n",mm,nn)
```

Scilab code Exa 17.9 Example 9

```
//Chapter 17, Example 17.9
clc


//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

//addition
function[c1,c2]=addition(x1,x2,x3,y1,y2,y3)
    c1 = x1+x2+x3
    c2 = y1+y2+y3
    endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction

function[c1,c2]=subtraction(x1,x2,y1,y2)
    c1 = x1-x2
    c2 = y1-y2
    endfunction
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
```

```
        x=y1-y2
    endfunction
//Variable Declaration
z1 = complex(1,2)      //complex form of Z1
z2 = complex(4,-8)     //complex form of Z2
z3 = complex(0,6)      //complex form of Z3
e1 = 8                 //source of voltage in volts
e1o = 20               //angle in degree
e2 = 10                 //source of voltage in volts
e2o = 0               //angle in degree

//Calculation
[a,b]=rect2polar(real(z2),imag(z2))      //z2
    function of conversion
[aa,bb]=voltdivider(a,e1,b,e1o)       //function of
    multiplication
[ac,bc]=polar2rect(aa,bb)     //function of
    conversion
[c,d]=rect2polar(real(z1),imag(z1))      //z1
    function of conversion
[ca,da]=voltdivider(c,e2,d,e2o)       //function of
    multiplication
[cb,db]=polar2rect(ca,da)     //function of
    conversion
[cc,dd]=subtraction(ac,cb,bc,db)    //function of
    subtraction
[e,f]=rect2polar(cc,dd)     //function of conversion
[ea,fa]=voltdivider(c,a,d,b)          //Z1*z2
    function of multiplication
[ee,ff]=polar2rect(ea,fa)     //function of
    conversion
[g,h]=rect2polar(real(z3),imag(z3))      //z3
    function of conversion
[ga,ha]=voltdivider(c,g,d,h)          //z1*z3
    function of multiplication
[gg,hh]=polar2rect(ga,ha)     //function of
    conversion
[gb,hb]=voltdivider(a,g,b,h)         //z2*z3 function
```

```
    of multiplication
[m,n]=polar2rect(gb,hb)    //function of conversion
[mm,nn]=addition(ee,gg,m,ff,hh,n)    //function of
    addition
[ma,na]=rect2polar(mm,nn)    //function of
    conversion
[mb,nb]=division(e,ma,f,na)    //function of
    division


//Results
printf("I2 = %.2f A < %.2f degree",mb,nb)
```

Scilab code Exa 17.12 Example 12

```
//Chapter 17, Example 17.12
clc
funcprot(0)
//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

//addition
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
    endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
```

```
        th=atand(y/x)
      if (x<0) & (y<0) then
            th=th-180
        end
    end
endfunction

function[c1,c2]=subtraction(x1,x2,y1,y2)
    c1 = x1-x2
    c2 = y1-y2
    endfunction
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
//Variable Declaration
z1 = 24                          //real part in
   milliampere
z1o = 0                       //angle in degree
z2 = -0.539                            //real part in
   milliampere
z2o = 21.80                        //angle in degree
z3 = 0.5                           //real part in
   milliampere
z3o = 0                       //angle in degree
z4 = 4                          //real part in milliampere
z4o = 0                       //angle in degree
z5 = 2.5                           //real part in
   milliampere
z5o= -2.29                        //angle in degree

```

```
//Calculation
[a,b]=voltdivider(z1,z2,z1o,z2o)    //z1*z2 function of multiplication
[aa,bb]=voltdivider(z3,z4,z3o,z4o)   //z3*z4 function of multiplication
[ac,bc]=polar2rect(a,b)                        //function of conversion
[ad,bd]=polar2rect(aa,bb)                        //function of conversion
[c,d]=addition(ac,ad,bc,bd)     //function of addition
[ca,da]=rect2polar(c,d)                        //function of conversion
[m,n]=voltdivider(z5,z2,z5o,z2o)       //function of multiplication
[ma,na]=voltdivider(z3,z3,z3o,z3o)     //function of multiplication
[mb,nb]=polar2rect(m,n)                        //function of conversion
[mc,nc]=polar2rect(ma,na)                        //function of conversion
[md,nd]=addition(mb,mc,nb,nc)     //function of addition
[me,ne]=rect2polar(md,nd)                        //function of conversion
[o,p]=division(ca,me,da,ne)              //function of division
//Results
printf("V1 = %.2f V < %.2f degree ",o,p)
```

Scilab code Exa 17.16 `Example 16`

```
//Chapter 17, Example 17.16
clc
//Rectangle to Polar Conversion
```

```
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=-90
    else
        th=atand(y/x)
    end
endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar1(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction
  //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
//addition
function[c1,c2]=addition1(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
    endfunction
```

```
//addition
function[c1,c2]=addition(x1,x2,x3,y1,y2,y3)
    c1 = x1+x2+x3
    c2 = y1+y2+y3
    endfunction
//Variable Declaration
z1 = complex(1/4,0)
z2 = complex(0,1/5)
z3 = complex(0,1/-2)
i1 = 6
i1o = 0
i2 = 4
i2o = 0

//Calculation
[a,b]=addition1(real(z3),real(z2),imag(z3),imag(z2))    //y3+y2
[aa,bb]=rect2polar(a,b)
[ac,bc]=voltdivider(-aa,i1,-bb,i1o)      //(y3+y2)/i1
[c,d]=rect2polar(real(z2),imag(z2))
[cc,dd]=voltdivider(i2,c,i2o,d)          //i2y2
[w,v]=polar2rect(ac,bc)
[ww,vv]=polar2rect(cc,dd)
[wa,va]=addition1(w,ww,v,vv)
[wb,vb]=rect2polar(wa,va)
[ca,da]=rect2polar(real(z1),imag(z1))
[cb,db]=rect2polar1(real(z3),imag(z3))
[e,f]=voltdivider(ca,cb,da,db)
[ee,ff]=polar2rect(e,f)
[g,h]=voltdivider(c,cb,d,db)
[gg,hh]=polar2rect(g,h)
[m,n]=voltdivider(ca,c,da,d)
[mm,nn]=polar2rect(m,n)
[o,p]=addition(ee,gg,mm,ff,hh,nn)
[oo,pp]=rect2polar(o,p)
[s,t]=division(wb,oo,vb,pp)
//Results
```

```
printf("V1 = %.2f V < %.2f degree ",s,t)
```

Scilab code Exa 17.17 Example 17

```
//Chapter 17, Example 17.17
clc
//Variable Declaration
z1= complex(7,8)
z2 = complex(4,5)
z3 = complex(0,-10)
z4 = complex(8,0)
e1 = 20
e1o = 0
i1 = 10
i1o = 20

//Calculation



//Results
```

Scilab code Exa 17.19 Example 19

```
//Chapter 17, Example 17.19
clc
//Variable Declaration


//Calculation



```

```
//Results
```

Scilab code Exa 17.20 `Example 20`

```
//Chapter 17, Example 17.20
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=-90
    else
        th=atand(y/x)
    end
endfunction

   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
//addition
function[c1,c2]=addition1(x1,x2,y1,y2)
    c1 = x1+x2
```

```
    c2 = y1+y2
    endfunction
function[c1,c2]=addition(x1,x2,x3,y1,y2,y3)
    c1 = x1+x2+x3
    c2 = y1+y2+y3
    endfunction
//Variable Declaration
za=complex(0,-4)              //complex form of Za
zb=complex(0,-4)              //complex form of Zb
zc=complex(3,4)               //complex form of Zc
z4 = complex(2,0)             //complex form of Z4
z5 = complex(3,0)             //complex form of Z5

//Calculation
[a,b]=rect2polar(real(zb),imag(zb))    //function of
    conversion
[c,d]=rect2polar(real(zc),imag(zc))    //function of
    conversion
[e,f]=voltdivider(a,c,b,d)         //function of
   multiplication
[g,h]=addition(real(za),real(zb),real(zc),imag(za),
   imag(zb),imag(zc))     //function of addition
[i,j]=rect2polar(g,h)     //function of conversion
[m,n]=division(e,i,f,j)          //function of
   division
[o,p]=polar2rect(m,n)             //Z1 function of
   conversion
[aa,bb]=rect2polar(real(za),imag(za))     //function
   of conversion
[ee,ff]=voltdivider(aa,c,bb,d)         //function of
   multiplication
[mm,nn]=division(ee,i,ff,j)         //function of
   division
[oo,pp]=polar2rect(mm,nn)       //Z2 function of
   conversion
[ac,bc]=voltdivider(aa,a,bb,b)         //function of
   multiplication
[ad,bd]=division(ac,i,bc,j)         //function of
```

```
division
[ae,be]=polar2rect(ad,bd)       //Z3 function of conversion
[ma,na]=addition1(o,real(z4),p,imag(z4))     //function of addition
[mb,nb]=rect2polar(ma,na)                    //ZT1 function of conversion
[s,t]=addition1(oo,real(z5),pp,imag(z5))     //function of addition
[sa,ta]=rect2polar(s,t)                       //ZT2 function of conversion
[sb,tb]=voltdivider(mb,sa,nb,ta)            //function of multiplication
[sc,tc]=addition1(ma,s,na,t)      //function of addition
[se,te]=rect2polar(sc,tc)                   //function of conversion
[sd,td]=division(sb,se,tb,te)           //function of division
[sf,tf]=polar2rect(sd,td)               //ZT3 function of conversion
[si,ti]=addition1(ae,sf,be,tf)      //function of addition
[sv,tv]=rect2polar(si,ti)                   //function of conversion


//Results
printf("ZT = %.2f ohm < %.2f degree \n",sv,tv)
```

Scilab code Exa 17.21 Example 21

```
//Chapter 17, Example 17.21
clc
//Variable Declaration
```

```
zt = complex(1,2)
zd = complex(3,6)

//Calculation
a= real(zd)/3
b= imag(zd)/3
c = a*3
d = b*3
e = c/2
f = d/2
g = (e*2)/3
h = (f*2)/3
//Results
printf("Zy = %d ohm + j %d ohm \n",a,b)
printf("Zt = %d ohm + j %d ohm \n",real(zt),imag(zt)
  )
printf("Zdel = %d ohm + j %d ohm \n",c,d)
printf("Z1 = %.1f ohm + j %d ohm \n",e,f)
printf("Zt = %d ohm + j %d ohm \n",g,h)
```

Chapter 18

Network Theorems ac

Scilab code Exa 18.1 `Example 1`

```
//Chapter 18, Example 18.1
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=-90
    else
        th=atand(y/x)
    end
endfunction
  //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
```

```
   endfunction
//addition
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
    endfunction

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
//Variable Declaration
z1 = complex(0,4)          //complex form of Z1
z2 = complex(0,4)          //complex form of Z2
z3 = complex(0,-3)         //complex form of Z3
e1 = 10                    //real value of voltage
e1o = 0                    //angle in degree
e2 = 5                     //real value of voltage
e2o = 0                    //angle in degree
//Calculation
[a,b]=rect2polar(real(z2),imag(z2)) //function of conversion
[c,d]=rect2polar(real(z3),imag(z3)) //function of conversion
[e,f]=voltdivider(a,c,b,d)   //function of multiplication
[g,h]=addition(real(z2),real(z3),imag(z2),imag(z3)) //function of addition
[gg,hh]=rect2polar(g,h) //function of conversion
[m,n]=division(e,gg,f,hh)              //z23 in ohms
[mo,no]=polar2rect(m,n)
[o,p]=addition(mo,real(z1),no,imag(z1))  //function of addition
[oo,pp]=rect2polar(o,p) //function of conversion
[s,t]=division(e1,oo,e1o,pp)     //Is1 in ampere
[ss,tt]=polar2rect(s,t) //function of conversion
[aa,ba]=voltdivider(c,s,d,t)  //function of
```

```
multiplication
[ab,bb]=rect2polar(aa,ba) //function of conversion
[ac,bc]=division(ab,gg,bb,hh)   //I  in ampere
z12 = imag(z1)/2                //imaginary part of Z12
[cc,dd]= addition(real(z1),real(z3),z12,imag(z3))
[ca,da]=rect2polar(cc,dd) //function of conversion
[cb,db]=division(e2,ca,e2o,da)     //Is2
i2 = cb/2                        //in ampere
[ma,na]=polar2rect(cb,db) //function of conversion
[mm,nn]=addition(ss,ma,tt,na)    //function of addition
[mb,nb]=rect2polar(mm,nn)        //final I function of conversion


//Results
printf("I = %.2f A < %.2f degree \n",mb,nb)
```

Scilab code Exa 18.2 Example 2

```
//Chapter 18, Example 18.2
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
```

```
endfunction
//addition
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
endfunction
//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
//Variable Declaration
z1 = complex(0,6)        //complex form of Z1
z2 = complex(6,-8)       //complex form of Z2
i1 = 2                   //current in ampere
e1= 20                   //real value of voltage
e1o = 30                 //angle in degree
//Calculation
[a,b]=addition(real(z1),real(z2),imag(z1),imag(z2))    //function of addition
[aa,bb]=rect2polar(a,b)    //function of conversion
c=imag(z1)*i1              //imaginary part
[i,j]=rect2polar(real(z1),c)    //function of conversion
[e,f]=division(i,aa,j,bb)          //I1 function of division
[m,n]=division(e1,aa,e1o,bb)      //I2 function of division
[o,p]=polar2rect(e,f)    //function of conversion
[oo,pp]=polar2rect(m,n)    //function of conversion
[s,t]=addition(o,oo,p,pp)   //function of addition
[ss,tt]=rect2polar(s,t)    //function of conversion
```

```
//Results
printf("I = %.2f A < %.2f degree \n",ss,tt)
```

Scilab code Exa 18.3 Example 3

```
//Chapter 18, Example 18.3
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction


//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
    endfunction
//Variable Declaration
i1= 1.9                    //real value of current in
    ampere
i1o = 108.43               //angle in degree
r1  = 6                    //resistance in ohms
i2 = 3.16                  //real value of current in
    ampere
i2o = 48.43               //angle in degree
it = i1+i2                  //real value of current in
     ampere
ito = i1o-i2o               //angle in degree
```

```
ia = 4.42                      //total current in ampere
iao = 70.2                     //angle in degree
//Calculation
v1 = i1*r1                   //in volts
v2 = i2*r1                   //in volts
[a,b]=polar2rect(v1,i1o)     //function of conversion
[c,d]=polar2rect(v2,i2o)     //function of conversion
[e,f]=addition(a,c,b,d)      //function of addition
[i,j]=rect2polar(e,f)      //function of conversion
vt = ia*r1                   //in volts

//Results
printf("V6ohm = %.1f V < %.1f degree \n",i,j)
printf("V6ohm = %.1f V < %.1f degree (checks)\n",vt,
   iao)
```

Scilab code Exa 18.4 Example 4

```
//Chapter 18, Example 18.4
clc
//multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
```

```
endfunction


//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
    endfunction
//Variable Declaration
r1 = 0.5*10^3              //resistance1 in kiloohms
r1o = 0                    //angle in degree
r2 = 1*10^3                //resistance2 in kiloohms
r2o = 0                    //angle in degree
r3 = 3*10^3                 //resistance3 in kiloohms
r3o = 0                    //angle in degree
e1 = 12                    //source voltage in volts
xc = 10*10^3               //reactance in kiloohms
xco = -90                    //angle in degree
xl = 2*10^3               //reactance in kiloohms
xlo = 90                    //angle in degree
e2 = 4                     //source voltage in volts
e2o = 0                    //angle in degree
//Calculation
rp1 = (r1*r3)/(r1+r3)              //in kiloohms
v3 = (rp1*e1)/(rp1+r2)                   //V3(dc) in volts
[a,b]=voltdivider(r2,xc,r2o,xco)  //function of multiplication
[c,d]=polar2rect(r2,r2o)     //function of conversion
[e,f]=polar2rect(xc,xco)     //function of conversion
[g,h]=addition(c,e,d,f)      //function of addition
[i,j]=rect2polar(g,h)    //function of conversion
[m,n]=division(a,i,b,j)                //z2 function
```

```
   of division
[o,p]=polar2rect(r3,r3o)     //function of conversion
[q,r]=polar2rect(xl,xlo)     //function of conversion
[s,t]=addition(o,q,p,r)      //function of addition
[v,w]=rect2polar(s,t)                    //z3 function of conversion
[ma,na]=voltdivider(m,v,n,w)  //function of multiplication
[mb,nb]=polar2rect(m,n)     //function of conversion
[mc,nc]=addition(mb,s,nb,t)      //function of addition
[md,nd]=rect2polar(mc,nc)                //Z2 + Z31 function of conversion
[oo,pp]=division(ma,md,na,nd)      //function of division
[qa,ra]=polar2rect(r1,r1o)     //function of conversion
[qb,rb]=polar2rect(oo,pp)     //function of conversion
[qc,rc]=addition(qa,qb,ra,rb)      //function of addition
[qd,rd]=rect2polar(qc,rc)                 //Zt function of conversion
[va,wa]=division(e2,qd,e2o,rd)         //Is function of division
[sa,ta]=voltdivider(m,va,n,wa)         //Z2*Is function of multiplication
[sb,tb]=division(sa,md,ta,nd)          //I3 function of division
[oa,pa]=voltdivider(sb,r3,tb,r3o)      //V3(ac) function of multiplication
pb=oa*sqrt(2)          //power in watt
//Results
printf("V3 = %.1f V \n",v3)
printf("V3 = %.1f + %.2f sin(wt %.2f degree)\n",v3,pb,pa)
```

Scilab code Exa 18.5 Example 5

```
//Chapter 18, Example 18.5
clc
funcprot(0)
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
endfunction
//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

  //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
//Variable Declaration
z1 = 4          //real value of Z1
z1o = 0         //angle in degree
z2 = complex(6,8)       //complex form of Z2
u=20                   //unit
v=10                   //real value of voltage
vo = 0                 //angle in degree
h=100                  //unit
iu = 20*10^-3            //current in milliampere
```

```
//Calculation
[a,b]=addition(z1,real(z2),z1o,imag(z2))   //function of addition
[c,d]=rect2polar(a,b)    //function of conversion
e=1/c                                              //I1
f=-d
[g,ha]=voltdivider(z1,e,z1o,f)       //I2 function of multiplication
i=e*u*v
[m,n]=polar2rect(i,f)       //I1 function of conversion
j=g*h*iu
[ma,na]=polar2rect(j,ha)    //function of conversion)
[q,r]=addition(m,ma,n,na)  //function of addition
[qa,ra]=rect2polar(q,r)    //function of conversion
//Results
printf("I2 = %.2f A < %.2f degree \n",qa,ra)
```

Scilab code Exa 18.7 Example 7

```
//Chapter 18, Example 18.7
clc
//multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction

//division
```

```
function[q,x]=division1(x1,x2,y1,y2)
       q=x1/x2
       x=y1+y2
   endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=-90
    else
        th=atand(y/x)
    end
endfunction

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
    endfunction
//Variable Declaration
z1=complex(0,8)       //complex form of Z1
z2=complex(0,-2)      //complex form of Z2
ea= 10                //source voltage in volts
eo = 0                //angle in degree
//Calculation
[a,b]=rect2polar(real(z1),imag(z1))   //function of conversion
[c,d]=rect2polar(real(z2),imag(z2))   //function of conversion
[e,f]=voltdivider(a,c,b,d)  //function of multiplication
[g,h]=addition(real(z1),real(z2),imag(z1),imag(z2))
[gg,hh]=rect2polar(g,h)   //function of conversion
```

```
[o,p]=division(e,gg,f,hh)     //function of division
[s,t]=voltdivider(c,ea,d,eo)  //function of multiplication
[ss,tt]=rect2polar(g,h)   //function of conversion
[oo,pp]=division1(s,ss,t,tt)   //function of conversion


//Results
printf("Zth = %.2f ohm < %d degree \n",o,p)
printf("Eth = %.2f V < %d degree \n",oo,pp)
```

Scilab code Exa 18.8 Example 8

```
//Chapter 18, Example 18.8
clc
//multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction


//Polar to Rectangle conversion
```

```
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
    endfunction
//Variable Declaration
z1=complex(6,8)         //complex form of Z1
z2=complex(3,-4)        //complex form of Z2
z3=complex(0,5)         //complex form of Z3
ea= 10                  //real value of voltage
eo = 0                  //angle in degree
//Calculation
[aa,bb]=rect2polar(real(z1),imag(z1))  //function of conversion
[ac,bc]=rect2polar(real(z2),imag(z2))  //function of conversion
[a,b]=voltdivider(aa,ac,bb,bc)    //function of multiplication
[g,h]=addition(real(z1),real(z2),imag(z1),imag(z2))  //function of addition
[i,j]=rect2polar(g,h)  //function of conversion
[ii,jj]=division(a,i,b,j)    //function of division
[ia,ja]=polar2rect(ii,jj)  //function of conversion
[gg,hh]=addition(real(z3),ia,imag(z3),ja)  //function of addition
[ga,ha]=rect2polar(gg,hh)  //function of conversion
[c,d]=voltdivider(ac,ea,bc,eo)    //function of multiplication
[ca,da]=division(c,i,d,j)    //function of division

//Results
printf("Zth = %.2f ohm < %.2f degree \n",ga,ha)
printf("Eth = %.2f V < %.2f degree \n",ca,da)
```

Scilab code Exa 18.9 Example 9

```
//Chapter 18, Example 18.8
clc

//Variable Declaration
zth = 2*10^3                    //resistance in kiloohms
rs = 0.5*10^3                   //resistance1 in kiloohms
rs1 = 2.3*10^3                   //resistance2 in kiloohms
rl = 1*10^3                   //resistance3 in kiloohms

//Calculation
rsa = rs+rs1                  //total resistance in kiloohms
i1 = 1/rsa                     //wrong answer in textbook
eth = -i1*zth                 //in ohmEi
vl= (rl*eth)/((1*10^3)+(2*10^3))   //in Ei


//Results
printf("Zth = %d kohm \n",zth/10^3)
printf("Eth = %.2fEi \n",eth)
printf("VL = %.2fEi \n",vl)
```

Scilab code Exa 18.14 Example 14

```
//Chapter 18, Example 18.14
clc;

//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
```

```
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction


//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
    endfunction
//multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1-y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
      q=x1/x2
      x=y1-y2
   endfunction
//Variable Declaration
z1=complex(3,4)     //complex form of Z1
z2=complex(0,-5)    //complex form of Z2
e1 = 20             //real value of voltage
e1o = 0             //angle in degree

//Calculation
[a,b]=rect2polar(real(z1),imag(z1))   //function of
```

```
    conversion
[c,d]=rect2polar(real(z2),imag(z2))    //function of
    conversion
[ca,da]=voltdivider(a,c,b,d)    //function of
    multiplication
[cb,db]=addition(real(z1),real(z2),imag(z1),imag(z2)
    )
[cc,dd]=rect2polar(cb,db)    //function of conversion
[e,f]=division(ca,cc,da,dd)    //function of division
[o,p]=polar2rect(e,f)    //function of conversion
[m,n]=division(e1,a,e1o,b)    //function of division
//Resultss
printf("Zn = %.2f ohm %.2fj ohm \n",o,p)
printf("In = %d A < %.2f degree \n",m,n)
```

Scilab code Exa 18.15 Example 15

```
//Chapter 18, Example 18.15
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction


//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
```

```
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
    endfunction
//multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
//Variable Declaration
z1=complex(2,-4)        //complex form of z1
z2=complex(1,0)         //complex form of z2
z3=complex(0,5)         //complex form of z3
i = 3                   //current in ampere
//Calculation
[a,b]=addition(real(z1),real(z2),imag(z1),imag(z2))      //z1+z2 in comp
[aa,bb]=rect2polar(a,b)                    //z1+z2 in polar form
[ac,bc]=rect2polar(real(z3),imag(z3))     //z3 in polar form
[ad,bd]=voltdivider(ac,aa,bc,bb)          //z3(z1+z2)
[ae,be]=addition(real(z3),a,imag(z3),b)    //z3+(z1+z2) in complex
[af,bf]=rect2polar(ae,be)         //function of conversion
[ag,bg]=division(ad,af,bd,bf)     //function of division
[ah,bh]=polar2rect(ag,bg)         //function of conversion
```

```
za = real(z1)*i              //real part
zb = imag(z1)*i             //imaginary pary
[c,d]=rect2polar(za,zb)          //function of conversion
[e,f]=division(c,aa,d,bb)       //function of division
//Results
printf("Zn = %.2f ohm + j%.2f ohm \n",ah,bh)
printf("In = %.2f A < %.1f degree \n",e,f)
```

Scilab code Exa 18.16 Example 16

```
//Chapter 18, Example 18.16
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction
 //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
//Variable Declaration
zn = complex(7.50,2.50)          //complex form of Zn
in = 2.68                       //real value of current
ino = -10.3                     //angle in degree

//Calculation
[a,b]=rect2polar(real(zn),imag(zn))     //function of conversion
[c,d]=voltdivider(in,a,ino,b)    //function of multiplication
```

```


//Results
printf("Eth = %.1f V < %.2f degree \n",c,d)
```

Scilab code Exa 18.19 `Example 19`

```
//Chapter 18, Example 18.19
clc
funcprot(0)
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction


//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

//addition
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
    endfunction
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
```

```
    s=y1+y2
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
        q=x1/x2
        x=y1-y2
   endfunction
//Variable Declaration
z1 = complex(6,-8)
z2 = complex(0,8)
e1 = 9
e1o = 0
//Calculation
[a,b]=rect2polar(real(z1),imag(z1))
[c,d]=rect2polar(real(z2),imag(z2))
[e,f]=voltdivider(a,c,b,d)
[g,h]=addition(real(z1),real(z2),imag(z1),imag(z2))
[gg,hh]=rect2polar(g,h)
[o,p]=division(e,gg,f,hh)
[oo,pp]=polar2rect(o,p)
[aa,bb]=voltdivider(c,e1,d,e1o)
[cc,dd]=division(aa,gg,bb,hh)
pmax = (cc*cc)/(4*a)
//Results
printf("ZL =  %.2f ohm + j%d ohm \n",oo,pp)
printf("Pmax = %.2f W ",pmax)
```

Scilab code Exa 18.20 Example 20

```
//Chapter 18, Example 18.20
clc
//Variable Declaration


```

```
//Calculation



//Results
```

Scilab code Exa 18.21 Example 21

```
//Chapter 18, Example 18.21
clc
//Variable Declaration
rth = 4                               //reistance in ohms
xth = 7                               //in ohms
Xload = 4                             //load resistance in
    ohms
eth = 20                              //source voltage in
    volts
//Calculation
rl = sqrt((rth*rth)+ ((xth-Xload)*(xth-Xload)))  //
    load resistance in ohms
rav = (rth+rl)/2                    //resistance in ohms
p = (eth*eth)/(4*rav)               //power in watt
pmax = (eth*eth)/(4*rth)            //power in watt
//Results
printf("RL = %d ohm \n",rl)
printf("Rav = %.1f ohm \n",rav)
printf("P = %.2f W \n",p)
printf("Pmax = %d W \n",pmax)
```

Chapter 19

Power ac

Scilab code Exa 19.1 `Example 1`

```
//Chapter 19, Example 19.1, page 860
clc
//Initialisation
p1=100                              //power in watt
p2=200                              //power in watt
p3=300                              //power in watt
var1=0                              //volt-amperes reactive
var2=700                              //volt-amperes reactive
var3=1500                              //volt-amperes reactive

//Calculation
va1=sqrt((p1**2)+(var1**2))           //volt-amperes
va2=sqrt((p2**2)+(var2**2))           //volt-amperes
va3=sqrt((p3**2)+(var3**2))           //volt-amperes
pt=p1+p2+p3                              //total power in
```

```
    watt
qt=var3-var2                                      //total volt−
    amperes
st=sqrt((pt**2)+(qt**2))                                      //
    total volt−amperes reactive
fp=pt/st                                    //power factor
i=st/p1                                    //current in ampere
teta=acosd(fp)                                   //angle in
    degree

//Result
printf("VA of load 1 = %d \n",va1)
printf("VA of load 2 = %.1f \n",va2)
printf("VA of load 3 = %.2f \n",va3)
printf("Total Watt, Pt = %d \n",pt)
printf("Total volt−amperes reactive, Qt = %d \n",qt)
printf("Total volt−amperes, St = %d \n",st)
printf("Power factor, Fp = %.1f leading (C) \n",fp)
printf(" I = %d < + %.2f degree \n",i,teta)
```

Scilab code Exa 19.2 Example 2

```
//Chapter 19, Example 19.2, page 861
clc
funcprot()
function [r,th]=rect2pol(x,y)
//rectangle to polar coordinate conversion
  r=sqrt(x^2+y^2);
  th = atan(y,x)*180/%pi;
endfunction

//Initialisation
E=100                         //emf in volt
r=6                           //resistance in ohm
xl=7                            //inductive reactance in ohm
```

```
xlang=90                          //angle in degree
xc=15                          //capacitive reactance in ohm
xcang=-90                          //angle in degree
f1=60                          //frequency in hertz


//Calculation
I=E/(r+complex(0,xl)+complex(0,-xc))                    //current in ampere
[ir,iang]=rect2pol(real(I),imag(I))                     //conversion to polar form
vr=ir*r                                  //in volt
vl=ir*xl                                  //in volt
vlang=iang*xlang                                  //in degree
vc=ir*xc                                  //in volt
vcang=xcang*iang                                  //in degree
pt=E*ir*cosd(iang)                                  //power in watt
st=E*ir                                  //VAR
qt=E*ir*sind(iang)                                  //VA
fp=pt/st                                  //power factor
wr=(vr*ir)/f1                                  //energy in joule
wl=(vl*ir)*(2*3.14*f1)**-1                         //energy in joule
wc=(vc*ir)/(2*3.14*f1)                            //energy in joule

//Result
printf("(a) I = %d A < %.2f degree \n",ir,iang)
printf("    Pt = %d W \n",pt)
printf("    Qt = %d VAR \n",qt)
printf("    St = %d VA \n",st)
```

```
printf("     Fp = %.1f leading (C) \n",fp)
printf("(b) Wr = %d J \n",wr)
printf("(c) Wl = %.2f J \n",wl)
printf("(d) Wc = %.2f J \n",wc)
```

Scilab code Exa 19.3 Example 3

```
//Chapter 19, Example 19.3, page 861
clc
funcprot()
function [r,th]=rect2pol(x,y)
//rectangle to polar coordinate conversion
  r=sqrt(x^2+y^2);
  th = atan(y,x)*180/%pi;
endfunction

//Initialisation
n=12                       //no of device
p1=60                       //power in watt
q1=0                        //in VAR
fp1=1                       //power factor
p2=6400                       //power in watt
q2=0                       //in VAR
fp2=1                       //power factor
n2=5                       //no of device
p0=746                       //power in watt
eta=0.82                            //eta contat
fp3=0.72                        //power factor
e=208                                  //emf in volt
r=9                                 //resistance in ohm
xc=12                               //capacitive reactance
    in ohm
z=complex(9,-12)

//Calculation
```

```
tp1=n*p1                         //power in watt
s1=tp1                            //in VA
s2=p2                            //in VA
p3=(p0*n2)/eta                         //power in watt
s3=p3/fp3                            //in VA
q3=s3*sin(asin(fp3))                          //in VAR
[zr,zang]=rect2pol(real(z),imag(z))    //function of conversion
i=e/zr                               //in ampere
iang=0-zang
p4=(i**2)*r                        //power in watt
q4=(i**2)*xc                        //in VAR
s4=sqrt((p4**2)+(q4**2))                               //in VA
fp4=p4/s4                        //power factor
pt=tp1+p2+p3+p4                        //total power in watt
qt=q1+q2+q3-q4                        //in  VAR(L)
st=sqrt((pt**2)+(qt**2))        //in VA
fpt=pt/st                        //power factor
tetat=acosd(fpt)                 //angle in degree
it=st/e                          //in ampere

//Result
printf("(a) Bulb \n")
printf("     P1 = %d W \n",tp1)
printf("     Q1 = %d VAR \n",q1)
printf("     S1 = %d VA\n",s1)
printf("     Fp1 = %d \n",fp1)
printf("    Heating elements \n")
printf("     P2 = %.1f kW\n",p2/1000)
printf("     Q2 = %d VAR\n",q2)
printf("     S2 = %.1f KVA\n",s2/1000)
printf("     Fp2 = %d \n",fp2)
printf("    Motor \n")
printf("     P3 = %.2f W\n",p3)
printf("     Q3 = %.2f VAR\n",q3)
printf("     S3 = %.2f VA\n",s3)
```

```
printf("       Fp = %.2f lagging \n",fp3)
printf("      Capacitive load \n")
printf("       P4 = %.2f W\n",p4)
printf("       Q4 = %.2f VAR\n",q4)
printf("       S4 = %.2f VA\n",s4)
printf("       Fp = %.1f lagging \n",fp4)
printf("(b) \n")
printf("       Pt = %.2f W \n",pt)
printf("       Qt = %.2f VAR \n",qt)
printf("       St = %.2f VA\n",st)
printf("       Fp = %.3f lagging \n",fpt)
printf("(c) \n")
printf("       I = %.2f A < %.2f degree \n",it,-tetat)
```

Scilab code Exa 19.4 Example 4

```
//Chapter 19, Example 19.4, page 864
clc
//Initialisation

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

ei=5000                          //in VAR
v=100                             //emf in volt
fp=0.6                          //power factor

//Calculation
i=ei/v                              //current in ampere
teta=-acosd(fp)                     //phase angle in degree
zt=v/i                              //impedance of the
   circuit in ohm
```

```
[a,b]=polar2rect(zt,teta)     //function of conversion
//Result
printf("I = %d A < %.2f degree \n",i,teta)
printf("Zt =%.1f ohm + j %.1f ohm",a,-b)
```

Scilab code Exa 19.5 Example 5

```
//Chapter 19, Example 19.5, page 865
clc
//Initialisation
hp1=746                          //1 hp
eta=0.92                          //eta
fp=0.6                             //power factor
v=208                              //voltage
qc=5405.8                             //in VAR
f=60                               //frequency in hertz


//Calculation
p0=5*hp1                        //output power in watt
pi=p0/eta                       //input power in watt
teta=acosd(fp)                  //angle in degree
ql=pi*tand(teta)                //in VAR(L)
s=sqrt((pi**2)+(ql**2))          //in VA
xc=(v**2)/(qc)                   //in ohms
c=1/(2*3.14*f*xc)                //in microfarad
i1=s/v                           //in ampere
i2=pi/v                          //in ampere
im=pi/(v*fp)                     //in ampere


//Result
printf("(a) Power traingle, \n")
printf("    Pi = %.2f W \n",pi)
```

```
printf("     QL = %.2f VAR \n",ql)
printf("     S = %.2f VA \n",s)
printf("(b) C = %.1f uF \n",c*10**6)
printf("(c) At 0.6 Fp, I = %.2f A \n",i1)
printf("     At unity Fp, I = %.2f A \n",i2)
printf("(d) Im = %.2f A < %.2f degree \n",im,teta)
```

Scilab code Exa 19.6 Example 6

```
//Chapter 19, Example 19.6, page 867
clc
//Initialisation
s=20000                        //in VA
fp=0.7                         //power factor
p1=10000                        //power in watt
p2=14000                        //power in watt from fig. 19.28
e=1000                          //voltage
fp2=0.95                      //power factor
f=60                             //frequency in hertz

//Calculation
p=s*fp                           //power factor
teta = acosd(fp)                         //phase angle in degree
ql=s*sind(teta)                        //in VAR
st=sqrt(((p1+p2)**2)+(ql**2))                        //in VA
it=st/e                                  //current in ampere
teta2=acosd(fp2)                       //phase angle in degree
ql1=(p1+p2)*tand(teta2)                     //in VAR
qc=ql-ql1                      //in VAR
xc=(e**2)/qc                               //capacitive impedance in ohm
```

```
c=1/(2*3.14*f*xc)                                       //capacitance ih farad
st1=sqrt(((p1+p2)**2)+(ql1**2))                             //in VA
it1=st1/e                                  //current in ampere

//Result
printf("(a) It = %.2f A \n",it)
printf("      C = %.2f uF \n",c*10**6)
printf("(b)  I = %.2f A \n",it1)
```

Scilab code Exa 19.7 `Example 7`

```
//Chapter 19, Example 19.7, page 872
clc
//Initialisation
p=75                     //power in watt
i=5                     //current in watt
e=120                        //voltage
rad1=377                      //angular frequency in rad/s
p2=80                     //power in watt
i2=4                     //current in watt
p3=52                     //power in watt
i3=2                     //current in watt

//Calculation
r=p/i**2                                       //resistance in ohm
zt=e/i                                       //resistance in ohm
xl=sqrt((zt**2)-(r**2))                        //resistance in ohm
l=xl/(rad1)                                       //inductance in henry
```

```
r2=p2/i2**2                                    //resistance in ohm
r3=p3/i3**2                                    //resistance in ohm
zt3=e/i3                                       //resistance in ohm
xl3=sqrt((zt3**2)-(r3**2))                     //resistance in ohm
l3=xl3/(rad1)                                  //inductance in henry

//Result
printf("(a) R = %d ohm \n      L = %.2f mH \n",r,l*10**3)
printf("(b) R = %d ohm \n",r2)
printf("(c) R = %d ohm \n      L = %.2f mH",r3,l3*10**3)
```

Chapter 20

Resonance

Scilab code Exa 20.1 Example 1

```
//Chapter 20, Example 20.1
clc
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//Variable Declaration
r = 2                      //resistance in ohms
e1 = 10                    //real value in volts
e1o = 0                    //angle in degree
zt = 2                     //real value in ohms
zto = 0                    //angle in degree
x1 = 10                    //real value in ohms
x1o = 90                    //angle in degree
fs = 5000                   //frequency in hertz

//Calculation
i = e1/r                                //in ampere
[a,b]=voltdivider(i,x1,e1o,x1o)    //function of
```

```
multiplication
[c,d]=voltdivider(i,x1,e1o,-x1o)    //function of multiplication
qs = x1/r                             //unit
bw = fs/qs                            //bandwidth in hertz
phpf=0.5*i*i*r                        //power in watt
//Results
printf("a. I = %d A < %d degree \n",i,e1o)
printf("    VL = %d V < %d degree \n",a,b)
printf("    Vc = %d V < %d degree \n",c,d)
printf("b. Qs = %d \n",qs)
printf("c. BW = %d Hz\n",bw)
printf("d. Phpf = %d W \n",phpf)
```

Scilab code Exa 20.2 Example 2

```
//Chapter 20, Example 20.2
clc

//Variable Declaration
fs = 4000                     //frequency in hertz
bw = 400                      //bandwidth in hertz
r = 10                        //resistance in ohm
xc = 100                      //in ohms

//Calculation
qs = fs/bw                     //unit
xl = qs*r                      //in ohms
l = xl/(2*%pi*fs)              //inductance in millihenry
c = 1/(2*%pi*fs*xc)            //capacitance in microfarad
//Results
printf("a. Qs = %d \n",qs)
```

```
printf("b. XL = %d ohms \n",xl)
printf("c. L = %.2f mH \n",l*10^3)
printf("   C = %.3f uF \n",c*10^6)
```

Scilab code Exa 20.3 Example 3

```
//Chapter 20, Example 20.3
clc

//Variable Declaration
xl = 300          //in ohms
r = 5             //in ohms
fs = 12000         //in hertz


//Calculation
qs = xl / r                  //unit
bw = fs/qs                   //bandwidth in hertz
f2 = fs + (bw/2)             //frequency2 in hertz
f1  = fs-100                 //frequency1 in hertz
//Results
printf("a. BW = %d Hz \n",bw)
printf("b. f2 = %d Hz \n",f2)
printf("   f1 = %d Hz \n",f1)
```

Scilab code Exa 20.4 Example 4

```
//Chapter 20, Example 20.4
clc

//Variable Declaration
bw = 200                        //bandwidth in hertz
fs = 2800                       //frequency in hertz
```

```
c = 101.5*10^-9                    //capacitance in
    nanofarad
imax = 200*10^-3                   //max current in
    milliampere


//Calculation
qs = fs/bw                          //unit
l = 1/(4*%pi*%pi*fs*fs*c)           //inductance in mH
xl = 2*%pi*fs*l                      //in ohm
r = xl/qs                            //resistance in ohm
e = imax*r                           //in volts
//Results
printf("a. BW = %d Hz \n",bw)
printf("    Qs = %d \n",qs)
printf("b. L = %.3f mH \n",l*10^3)
printf("    R = %d ohm \n",r)
printf("c. E = %d V \n",e)
```

Scilab code Exa 20.5 Example 5

```
//Chapter 20, Example 20.5
clc

//Variable Declaration
e = 120                             //in volts
p = 16                              //power in watt
ws = 10^5                           //in rad/s

//Calculation
r = (e*e)/p                         //resistance in ohm
fs = ws/(2*%pi)                     //frequency in hertz
bw = 0.15*fs                        //bandwidth in hertz
l = r/(2*%pi*bw)                    //inductance in mH
c = 1/ (4*%pi*%pi*fs*fs*l)          //capacitance in nF
```

```
xl = 2*%pi*fs*l                //in ohm
qs = xl/r                      // unit
a = 1/qs                       //unit

//Results
printf("a. P = %d ohm \n",r)
printf("b. BW = %.2f Hz \n",bw)
printf("c. L = %d mH \n",l*10^3)
printf("    C = %.2f nF \n",c*10^9)
printf("d. Qs = %.2f \n",qs)
printf("e. BW/fs = %.2f \n",a)
```

Scilab code Exa 20.6 Example 6

```
//Chapter 20, Example 20.6
clc

//Variable Declaration
l = 1*10^-3                  //inductance in mH
c = 1*10^-6                  //capacitance in uF
rs = 10*10^3                 //in ohm
i = 10*10^-3                 //current in milliampere


//Calculation
fp = 1/(2*%pi*sqrt(l*c))        //frequency in kHz
xl = 2*%pi*fp*l                 //in ohm
qp = rs/xl                      //unit
bw = fp/qp                      //bandwidth in Hz
f1 = (1/(4*%pi*c))*((1/rs)-sqrt((1/(rs*rs))+((4*c)/l
   )))        //frequency in kHz
f2 = (1/(4*%pi*c))*((1/rs)+sqrt((1/(rs*rs))+((4*c)/l
   )))        //frequency in kHz
vc = i*rs                      //in volts
il = vc/xl                     //load current in ampere
```

```
ic = vc/xl                    //in ampere

//Results
printf("a. fp = %.2f kHz \n",fp/10^3)
printf("b. Rs = %d kohm \n",rs/10^3)
printf("c. Qp = %.2f \n",qp)
printf("   BW = %.2f kHz \n",bw)
printf("   f1 = %.3f kHz \n",-f1/10^3)
printf("   f2 = %.3f kHz \n",f2/10^3)
printf("d. Vc = %d V \n",vc)
printf("e. IL = %.2f A \n",il)
printf("   Ic = %.2f A \n",ic)
```

Scilab code Exa 20.7 Example 7

```
//Chapter 20, Example 20.7
clc
funcprot(0)
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
function[c1,c2]=subtraction(x1,x2,y1,y2)
    c1 = x1-x2
    c2 = y1-y2
```

```
    endfunction
//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
//Variable Declaration
l = 0.3*10^-3                       //inductance in mH
c = 100*10^-9                       //capacitance in nF
rl = 20                             //in ohm
xco = -90                           //angle in degree
i = 2*10^-3                         //current in mA

//Calculation
fs = 1/(2*%pi*sqrt(l*c))                   //frequency in kHz
fm = fs*(sqrt(1-(1/4)*((rl*rl*c/l))))              //frequency in kHz
fp = fs*(sqrt(1-((rl*rl*c/l))))              //frequency in kHz
xl = 2*%pi*fm*l         //in ohm
xc = 1/(2*%pi*fm*c)                 //in ohm
[a,b]=rect2polar(rl,xl)                  //function of conversion
[c,d]=voltdivider(a,xc,b,xco)            //function of multiplication
[e,f]=subtraction(rl,0,xl,xc)          //function of subtraction
[ee,ff]=rect2polar(e,f)                  //function of conversion
[m,n]=division(c,ee,d,ff)          //function of division
vc = i*m                           //in mV
qp = xl/rl                          //unit
bw = fp/qp                        //bandwidth in kHz
ql = (2*%pi*fs*l)/rl               //unit
ztp = ql*ql*rl                     //in ohm
vc1 = i*ztp                         //in mV
bw1 = fp/qp                      //bandwidth in kHz
```

```
//Results
printf("a. fs = %.2f kHz \n",fs/10^3)
printf("   fm = %.2f kHz \n",fm/10^3)
printf("   fp = %.2f kHz \n",fp/10^3)
printf("b. ZTm = %.2f ohm < %.2f degree \n",m,n)
printf("   VCmax = %.2f mV \n",vc*10^3)
printf("c. Qp = %.2f \n",qp)
printf("d. BW = %.2f kHz \n",bw/10^3)
printf("e. fs = %.2f kHz \n",fs/10^3)
printf("   Ql = %.2f \n",ql)
printf("   ZTp = %.2f ohm < 0 degree \n",ztp)
printf("   VCmax = %.1f mV \n",vc1*10^3)
printf("   BW = %.2f kHz \n",bw1/10^3)
```

Scilab code Exa 20.8 Example 8

```
//Chapter 20, Example 20.8
clc

//Variable Declaration
l = 1*10^-3                        //inductance in mH
fp = 0.04*10^6                     //frequency in
    microfarad
rl = 10                            //in ohm
rs = 40*10^3                       //in kohm

//Calculation
xl = 2*%pi*fp*l                  //in ohm
ql = xl/rl                       //unit
rp = ql*ql*rl                    //in kiloohm
ztp = (rs*rp)/(rs+rp)            //in kiloohm
c = 1/(4*%pi*%pi*fp*fp*l)          //capacitance in nF
qp = ztp/xl                        //unit
bw = fp/qp                       //bandwidth in kHz
```

```
f1 = (1/(4*%pi*c))*((1/ztp)-sqrt((1/(ztp*ztp))+((4*c)/l)))   //frequency in kHz
f2 = (1/(4*%pi*c))*((1/ztp)+sqrt((1/(ztp*ztp))+((4*c)/l)))   //frequency in kHz
//Results
printf("a. Ql = %.2f \n",ql)
printf("b. Rp = %.2f kohm \n",rp/10^3)
printf("c. ZTp = %.2f kohm \n",ztp/10^3)
printf("d. C = %.2f nF \n",c*10^9)
printf("e. Qp = %.2f \n",qp)
printf("f. BW = %.2f kHz \n",bw/10^3)
printf("   f1 = %.3f kHz \n",-f1/10^3)
printf("   f2 = %.3f kHz \n",f2/10^3)
```

Scilab code Exa 20.9 `Example 9`

```
//Chapter 20, Example 20.9
clc
stacksize(10000000);
//Variable Declaration
l = 5*10^-3                         //inductance in mH
c = 50*10^-12                       //capacitance in pF
rl = 100                            //in ohm
rs = 50*10^3                        //in kohm
i = 2*10^-3                         //current in mA

//Calculation
fs = 1/(2*%pi*sqrt(l*c))           //frequency in kHz
xl = 2*%pi*fs*l                          //in kohm
ql = xl/rl                          //unit
fp = fs                            //frequency in kHz
rp = ql*ql*rl                         //in Megaohm
ztp=((rs*rp)/(rs+rp))                 //in kohm
qp = ztp/xl                        //unit
bw = fp/qp                        //frequency in kHz
```

```
bw1 = (1/(2*%pi))*((rl/l)+(1/(rs*c)))         //frequency in kHz
vp = i*ztp                            //in volts

//graph
x= 100 : 10000 : 500000
//x=318.31*10**3
rp1=(4*(%pi**2)*(l**2)*(x.^2))/rl
//rp1=rl**2/(4*(%pi**2)*(l**2)*(x.^2))
vp1=i*((rs*rp1)./(rs+rp1))

plot(x,vp1);
xtitle('Passband response')
xlabel('f')
ylabel('Vo')
//insufficient data from textbook for plotting a graph


//Results
printf("a. fp = fs = %.2f kHz \n",fs/10^3)
printf("b. Qp = %.2f \n",qp)
printf("c. BW = %.2f kHz \n",bw/10^3)
printf("   BW = %.2f kHz \n",bw1/10^3)
printf("d. Vp = %.2f V \n",vp)
```

Scilab code Exa 20.10 Example 10

```
//Chapter 20, Example 20.10
clc
//Variable Declaration
fp = 318.31*10^3              //frequency in kHz
ql = 100                      //unit
rp = 1*10^6                   //in Mega ohm
i = 2*10^-3                   //current in mA
```

```
//Calculation
qp = ql                          //unit
bw = fp/qp                       //frequency in kHz
ztp = rp                         //in mega ohm
vp = i*ztp                       //in volts



//Results
printf("a. fp = %.2f kHz \n",fp/10^3)
printf("b. Qp = %d \n",qp)
printf("c. BW = %.3f kHz \n",bw/10^3)
printf("d. Vp = %d V ",vp)
```

Scilab code Exa 20.11 Example 11

```
//Chapter 20, Example 20.11
clc
//Variable Declaration
fp = 50000                          //frequency in Hz
bw = 2500                           //bandwidth in Hz
l = 1*10^-3                         //inductance in mH
rl = 10                             //resistance in ohm
vp = 10                              //in volts
rs = 17.298*10^3                     //in kohms
//Calculation
qp = fp/bw                            //unit
xl = 2*%pi*fp*l                      //in ohm
ql = xl/rl                           //unit
rp = ql*ql*rl                         //in ohm
xc = xl                             //in ohm
c = 1/(2*%pi*fp*xc)                     //in uF
ztp = (rs*rp)/(rs+rp)              //in kohm
i = vp/ztp                                  //current in mA
```

```
//Results
printf("Qp = %d \n",qp)
printf("Ql = %.1f \n",ql)
printf("Rp = %.1f ohm \n",rp)
printf("C = %.2f uF \n",c*10^6)
printf("I = %.1f mA \n",i*10^3)
```

Chapter 21

Transformers

Scilab code Exa 21.1 Example 1

```
//Chapter 21, Example 21.1
clc
//Initialisation
k=0.6                               //coefficient of coupling
Lp=200*10**-3                       //in Henry
Ls=800*10**-3                       //in Henry
Np=50                                //no of turns in primary
phip=450*10**-3                       //in Wp/s
Ns=100                               //no of turns in primary
ip=0.2/10**-3                        //in Wp/s

//Calculation
M=k*sqrt(Lp*Ls)                      //Mutual Inductance
ep=Np*phip                          //induced voltage
es=k*Ns*phip                         //induced voltage
epi=Lp*ip                          //induced voltage
esi=M*ip                          //induced voltage

```

```
//Result
printf("(a) Mutual Inductance, M = %.3f mH \n",M
   *1000)
printf("(b) Induced Voltage, ep = %.1f V \n",ep)
printf("(c) Induced Voltage, es = %d V \n",es)
printf("(d) Induced Voltage, ep = %d V \n",epi)
printf("(e) Induced Voltage, es = %d V",esi)
```

Scilab code Exa 21.2 Example 2

```
//Chapter 21, Example 21.2
clc
//Initialisation
ep=200                          //induced voltage
np=50                           //no of turns
f=60                            //frequency in hertz
es=2400                          //induced voltage

//Calculation
phim=ep/(4.44*np*f)                  //in Wb/m
ns=(np*es)/ep                        //no of turns


//Result
printf("(a) Maximum Flux = %.2f mWb \n",phim*10**3)
printf("(b) Secondary turns Ns = %d turns",ns)
```

Scilab code Exa 21.3 Example 3

```
//Chapter 21, Example 21.3
clc
//Initialisation
np=40                           //no of turns
```

```
ns=5                          //no of turns
vl=200                         //voltage
zl=2*10**3                     //resistance in ohm
is=0.1                         //current in ampere

//Calculation
Ip=(ns/np)*is                  //current in amp
vg=(np/ns)*vl                  //voltage in volt
a=np/ns                       //turn ratio
zp=(a**2)*(zl)                   //in ohm


//Result
printf("(a) Ip = %.1f mA \n",Ip*10**3)
printf("     Vg = %d V \n",vg)
printf("(b) Zp = %d Kohm",zp/10**3)
```

Scilab code Exa 21.4 Example 4

```
//Chapter 21, Example 21.4
clc
//Initialisation
p1=600                 //power in watt
p2=400                 //power in watt
p3=2000                 //power in watt
vp=2400                 //voltage in volt
v1=120                 //voltage in volt
v2=240                 //voltage in volt

//Calculation
pt=p1+p2+p3                   //power in watt
Ip=pt/vp                 //current in ampere
R=vp/Ip                 //resistance in ohm
i1=p1/v1                 //current in ampere
i2=p3/v2                 //current in ampere
```

```
vl=1.73*vp                   //voltage in volt
pt1=3*pt                   //power in watt
a=vp/v2                  //ratio

//Result
printf("(a) R = %d ohm \n",R)
printf("(b) I1 = %d A \n",i1)
printf("     I2 = %.2f A \n",i2)
printf("(c) Vl = %d V \n",vl)
printf("(d) Pt = %d kW \n",pt1/1000)
printf("(e) a = %d \n",a)
```

Scilab code Exa 21.5 Example 5

```
//Chapter 21, Example 21.5
clc
//Initialisation
E=120                  //voltage in volt
rt1=512                  //resistance in ohm
rt2=8                    //resistance in ohm
R=8                    //resistance in ohm
np=8                   //no of turns
ns=1                   //no of turns

//Calculation
is=E/(rt1+rt2)                //current in ampere
p=(is**2)*R                //power in watt
zp=((np/ns)**2)*R                //resistance in ohm
is2=E/(rt1+zp)                //current in ampere
p2=(is2**2)*zp               //power in watt


//Result
printf("(a) P = %.2f W \n",p)
printf("(b) Zp = %d ohm \n",zp)
```

```
printf("      P = %.3 f W \n",p2)
```

Scilab code Exa 21.6 Example 6

```
//Chapter 21, Example 21.6
clc
//Initialisation
vp=70.7                    //voltage in volt
pp=10                      //power in watt
zl=8                       //resistance in ohm
rt=500

//Calculation
pt=pp*4                      //resulting power in watt
ip=pp/vp                  //current in ampere
zp=vp/ip                          //resistance in ohm
zp1=sqrt(zp/zl)                          //resistance in ohm
vl=vp/zp1                          //voltage in volt
rt2=rt/2                          //resistance in ohm
rt3=rt/3                         //resistance in ohm
rt4=rt/4                         //resistance in ohm

//Result
printf("(a) Resulting power = %d W \n",pt)
printf("(b) Zp = %d ohm \n",zp)
printf("(c) Zp = %d : 1 \n",round(zp1))
printf("(d) VL = %d V \n",round(vl))
printf("(e) One Speaker = %d ohm \n",rt)
printf("     Two Speaker = %d ohm \n",rt2)
printf("     Three Speaker = %d ohm \n",rt3)
printf("     Four Speaker = %d ohm \n",rt4)
```

Scilab code Exa 21.7 Example 7

```
//Chapter 21, Example 21.7
clc
funcprot()
function [r,th]=rect2pol(x,y)
//rectangle to polar coordinate conversion
  r=sqrt(x^2+y^2);
  th = atan(y,x)*180/%pi;
endfunction

//Initialisation
rp=1             //in ohm
a=2               //turn ratio
rs=1             //in ohm
xp=2             //in ohm
xs=2             //in ohm
ip=10             //in ampere
rl=60             //in ohm
vl=1200             //in volt

//Calculation
Re=rp+((a**2)*rs)           //in ohm
Xe=xp+((a**2)*xs)           //in ohm
c=Re+((a**2)*rl)
vg=ip*complex(c,Xe)          //in volt
[vgr,vgi]=rect2pol(real(vg),imag(vg))        //
   rectangle to polar conversion
vg2=a*vl        //in volt

//Result
printf("(a) Re = %d ohm \n",Re)
printf("    Xe = %d ohm \n",Xe)
printf("(b) Vg = %.2f V < %.2f degree \n",vgr,vgi)
printf("(c) Vg = %.2f V \n",vg2)
```

Scilab code Exa 21.8 Example 8

```
//Chapter 21, Example 21.8
clc
//Initialisation
L1=5             //in henry
L2=10             //in henry
L3=15             //in henry
M12=2             //in henry
M23=3             //in henry
M13=1             //in henry


//Calculation
Lt=L1+L2+L3+(2*M12)-(2*M23)-(2*M13)           //in
   henry


//Result
printf("Lt = %d H \n",Lt)
```

Scilab code Exa 21.10 Example 10

```
//Chapter 21, Example 21.10
clc
funcprot(0)
function [r,th]=rect2pol(x,y)
//rectangle to polar coordinate conversion
  r=sqrt(x^2+y^2);
  th = atan(y,x)*180/%pi;
endfunction

//Initialisation
Rp=3              //in ohm
Xlp=2400              //in ohm
W=400                 //angular frequency in rad/s
M=0.9                 //in henry
```

```
Rs=0.5                    //in ohm
Rl=40                    //in ohm
Xls=400                    //in ohm

//Calculation
Zp = complex(Rp,Xlp)                                  //in ohm
Zi=Zp+(((W*M)**2)/(complex(Rs+Rl,Xls)))          //Input
   impedance in ohm
[Zir,Zid]=rect2pol(real(Zi),imag(Zi))

//Result
printf("Zi = %.1f ohm + j %d ohm\n",real(Zi),imag(Zi
   ))
printf("    = %.2f ohm < %.2f degree",Zir,Zid)
```

Chapter 22

Polyphase Systems

Scilab code Exa 22.1 `Example 1`

```
//Chapter 22, Example 22.1
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
   //Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
```

```
endfunction
    //addition
function[c1,c2]=addition(x1,x2,x3,y1,y2,y3)
    c1 = x1+x2+x3
    c2 = y1+y2+y3
    endfunction
//Variable Declaration
teta2 = -120        //angle in degree
teta3 = 120        //angle in degree
eo = 120            //in volt
van = 120           //in volt
vano = 0        //angle in degree
vbn = 120           //in volt
vbno = -120        //angle in degree
vcn = 120           //in volt
vcno = 120        //angle in degree
zan = complex(3,4)        //complex form of Zan
zbn = 5             //in ohm
zbno = 53.13        //angle in degree
zcn = 5             //in ohm
zcno = 53.13        //angle in degree

//Calculation
el = sqrt(3)*eo                         //in volt
[a,b]=rect2polar(real(zan),imag(zan))    //function
    of conversion
[b,c]=division(van,a,vano,b)   //function of
    division
[e,f]=division(vbn,zbn,vbno,zbno)   //function of
    division
[g,h]=division(vcn,zcn,vcno,zcno)   //function of
    division
[bb,cc]=polar2rect(b,c)    //function of conversion
[ee,ff]=polar2rect(e,f)    //function of conversion
[gg,hh]=polar2rect(g,h)    //function of conversion
[m,n]=addition(bb,ee,gg,cc,ff,hh)     //function of
    addition
//Results
```

```
printf("a. theta2 = %d degree and theta3 = +%d
   degree \n",teta2,teta3)
printf("b. EL = %d V \n",el)
printf("c. Ian = %d A < %.2f degree \n",b,c)
printf("    Ibn = %d A < %.2f degree \n",e,f)
printf("    Icn = %d A < %.2f degree \n",g,h)
printf("d. IN = %d + j%d \n",m,n)
```

Scilab code Exa 22.2 Example 2

```
//Chapter 22, Example 22.2
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
   //Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
   //addition
function[c1,c2]=addition(x1,x2,x3,y1,y2,y3)
    c1 = x1+x2+x3
```

```
    c2 = y1+y2+y3
    endfunction
//Variable Declaration
teta2 = -120        //angle in degree
teta3 = 120        //angle in degree
vab = 150          //in volts
vabo = 0        //angle in degree
vbc = 150          //in volts
vbco = -120        //angle in degree
vca = 150          //in volts
vcao = 120        //angle in degree
zab = complex(6,8)          //complex form of Zab
zbc = 10                //in ohm
zbco = 53.13         //angle in degree
zca = 10                //in ohm
zcao = 53.13         //angle in degree
io = 15                 //in ampere

//Calculation
[a,b]=rect2polar(real(zab),imag(zab))           //function of conversion
[c,d]=division(vab,a,vabo,b)       //function of division
[e,f]=division(vbc,zbc,vbco,zbco)       //function of division
[g,h]=division(vca,zca,vcao,zcao)       //function of division
il = sqrt(3)*io                          //in ampere
//Results
printf("a. theta2 = %d degree and theta3 = +%d degree \n",teta2,teta3)
printf("b. Iab = %d A < %.2f degree \n",b,c)
printf("    Ibc = %d A < %.2f degree \n",e,f)
printf("    Ica = %d A < %.2f degree \n",g,h)
printf("c. IL = %.2f A \n",il)
```

Scilab code Exa 22.3 Example 3

```
//Chapter 22, Example 22.3
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction

//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
   //Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction
   //addition
function[c1,c2]=addition(x1,x2,x3,y1,y2,y3)
    c1 = x1+x2+x3
    c2 = y1+y2+y3
endfunction
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
```

```
//Variable Declaration
teta2 = -120          //angle in degree
teta3 = 120          //angle in degree
vab = 120             //in volts
vabo = 0        //angle in degree
vbc = 120             //in volts
vbco = 120         //angle in degree
vca = 120             //in volts
vcao = -120          //angle in degree
zab1 = 5                //in volt
zab1o = 0               //angle in degree
zab2 = 5                //in volt
zab2o = -90             //angle in degree
zab = complex(5,-5)        //complex form Zab
zbc = 3.54               //in ohm
zbco = -45         //angle in degree
zca = 3.54               //in ohm
zcao = -45         //angle in degree
io = 34                //in ampere

//Calculation
[aa,bb]=voltdivider(zab1,zab2,zab1o,zab2o)       //
   function of multiplication
[a,b]=rect2polar(real(zab),imag(zab))          //
   function of conversion
[s,t]=division(aa,a,bb,b)                //function of
   division
[c,d]=division(vab,s,vabo,t)       //function of
   division
[e,f]=division(vbc,zbc,vbco,zbco)       //function of
    division
[g,h]=division(vca,zca,vcao,zcao)       //function of
    division
il = sqrt(3)*io                         //in ampere
//Results
printf("a. theta2 = %d degree and theta3 = +%d
   degree \n",teta2,teta3)
printf("b. Iab = %.1f A < %d degree \n",c,d)
```

```
printf("    Ibc = %.1f A < %d degree \n",e,f)
printf("    Ica = %.1f A < %d degree \n",g,h)
printf("c. IL = %.2f A \n",il)
```

Scilab code Exa 22.4 Example 4

```
//Chapter 22, Example 22.4
clc
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
//Variable Declaration
ian = 2                    //in ampere
iano = 0                 //angle in degree
ibn = 2                    //in ampere
ibno = -120                 //angle in degree
icn = 2                    //in ampere
icno = 120                 //angle in degree
zan = 10                      //in ohm
zano = -53.13                 //angle in degree
//Calculation
[a,b]=voltdivider(ian,zan,iano,zano)    //function
   of multiplication
[c,d]=voltdivider(ibn,zan,ibno,zano)    //function
   of multiplication
[e,f]=voltdivider(icn,zan,icno,zano)    //function
   of multiplication
el = sqrt(3)*a                       //in volt
//Results
printf("a. Van = %d V < %.2f degree \n",a,b)
printf("    Vbn = %d V < %.2f degree \n",c,d)
printf("    Vcn = %d V < %.2f degree \n",e,f)
printf("b. El = %.1f V ",el)
```

Scilab code Exa 22.5 Example 5

```
//Chapter 22, Example 22.5
clc
//Variable Declaration
vo = 100                              //in volt
io = 20                         //in ampere
teta = 53.13                    //angle in degree
ro = 3                          //in ohm
vr = 60                         //in volt
el = 173.2                            //in volt
il = 20                                //in ampere
xo = 4                                //in ohm
//Calculation
po = vo*io*cosd(teta)                         //in watt
po1 = io*io*ro                              //in watt
po2 = (vr*vr)/ro                             //in watt
pt = 3*po                                     //in watt
pt1 = sqrt(3)*el*il*cosd(teta)                 //in watt
qo = vo*io*sind(teta)                          //in VAR
qo1 = io*io*xo                                     //in
   VAR
qt = 3*qo                                      //in VAR
qt1 = sqrt(3)*el*il*sind(teta)                     //in
   VAR
so = vo*io                                     //in VA
st = 3*so                                      //in VA
st1 = sqrt(3)*el*il                            //in VA
fp = pt1/st1                                 //lagging
//Results
printf("a. Po = %d W \n",po)
printf("    Po = %d W \n",po1)
printf("    Po = %d W \n",po2)
printf("    Pt = %d W \n",pt)
```

```
printf("    Pt = %d W \n",pt1)
printf("b. Qo = %d VAR \n",qo)
printf("    Qo = %d VAR \n",qo1)
printf("    Qt = %d VAR \n",qt)
printf("    Qt = %d VAR \n",qt1)
printf("c. So = %d VA \n",so)
printf("    St = %d VA \n",st)
printf("    St = %d VA \n",st1)
printf("d. Fp = %.1f lagging \n",fp)
```

Scilab code Exa 22.6 Example 6

```
//Chapter 22, Example 22.6
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction
//Variable Declaration
zd = complex(6,-8)                     //complex form
   of Zd
el = 200                                //in volt
io = 20                                //in ampere
ro = 6                                //in ohm
xo = 8                                 //in ohm
vo = 200                          //in volt
zy = complex(4,3)                   //complex form of Zy
ro1 = 4                                //in ohm
xo1 = 3                                //in ohm
vo1 = 116                              //in volt
```

```

//Calculation
[a,b]=rect2polar(real(zd),imag(zd))      //function of conversion
io = el/a                                //in ampere
pt = 3*io*io*ro                          //power in watt
qt = 3*io*io*xo                          //in VAR
st = 3*vo*io                             //in VA
[c,d]=rect2polar(real(zy),imag(zy))      //function of conversion
io1 = (el/sqrt(3))/(c)                   //in ampere
pty = 3*(io1*io1)*ro1                    //power in watt
qty = 3*(io1*io1)*xo1                    //in VAR
sty = 3*vo1*io1                          //in VA
ptt = pt+pty                             //power in watt
qtt = qt-qty                             //in VAR
stt = sqrt((ptt*ptt)+(qtt*qtt))          //in VA
fp = ptt/stt                             //leading
//Results
printf("For the delta: \n")
printf("Pt = %d W \n",pt)
printf("Qt = %d VAR (C) \n",qt)
printf("St = %d VA \n",st)
printf("For he Y: \n")
printf("Pty = %d W \n",pty)
printf("Qty = %d VAR (L) \n",qty)
printf("Sty = %.2f VA \n",sty)
printf("For the total load:\n")
printf("Pt = %d W \n",ptt)
printf("Qt = %.d VAR (C) \n",qtt)
printf("St = %.1f VA \n",stt)
printf("Fp = %.3f leading \n",fp)
```

Scilab code Exa 22.7 `Example 7`

```
//Chapter 22, Example 22.7
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=atand(y/x)
    end
endfunction
//addition
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
    endfunction
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

//Variable Declaration
vl = 12000                               //in volt
pt = 160000                                //power in watt
vo = 6936.42                               //in volt
voo = 0                                   //angle in degree
co = 0.86                                      //cos
ioo= -30.68                                  //angle in
   degree
zl = 25                                  //in ohm
zlo = 53.13                                  //angle in
```

```
degree
rl = 15                              //in ohm
//Calculation
vo = vl/sqrt(3)                    //in volt
io = pt/(3*vo*co)                  //in ampere
[a,b]=voltdivider(io,zl,ioo,zlo)      //function of multiplication
[c,d]=polar2rect(a,b)                 //function of conversion
[e,f]=polar2rect(vo,voo)                 //function of conversion
[g,h]=addition(c,e,d,f)                   //function of addition
[gg,hh]=rect2polar(g,h)                  //function of conversion
eab = sqrt(3)*gg                         //in volt
pline = 3*io*io*rl                        //power line in watt
ptt= pt+pline                            //total power
co1= pt/(sqrt(3)*eab*io)                 //cos
n = (pt/ptt)*100                      //efficiency in percentage
//Results
printf("a. Io = %.2f A \n",io)
printf("   Eab = %.2f V \n",eab)
printf("b. Fp = %.2f < 0.86 of load \n",co1)
printf("c. n = %.1f percent \n",n)
```

Scilab code Exa 22.8 Example 8

```
//Chapter 22, Example 22.8
clc
funcprot(0)
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
```

```
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=-90
    else
        th=atand(y/x)
      if (x<0) & (y<0) then
            th=th-180
        end
    end
endfunction
//Rectangle to Polar Conversion
function [r,th] = rect2polar1(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=90
    else
        th=-atand(y/x)
    end
endfunction

//subtraction
function[c1,c2]=subtraction(x1,x2,y1,y2)
    c1 = x1-x2
    c2 = y1-y2
    endfunction
//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction
   //Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

//Variable Declaration
eab = 208                     //real value
```

```
eabo = 0                               //angle in degree
zab = 10                              //real value
zabo = 0                               //angle in degree
ebco = -120                               //angle in degree
ecao = 120                               //angle in degree
zbc = complex(15,20)                     //complex form of Zbc
zca = complex(12,12)                     //complex form of Zca

//Calculation
[a,b]=division(eab,zab,eabo,zabo)           //Iab function of division
[c,d]=rect2polar(real(zbc),imag(zbc))                  // function of conversion
[e,f]=division(eab,c,ebco,d)                //Ibc function of division
[g,h]=rect2polar1(real(zca),imag(zca)) //function of conversion
[m,n]=division(eab,g,ecao,h)             //Ica function of division
[aa,bb]=polar2rect(a,b)                //function of conversion
[mm,nn]=polar2rect(m,n)                //function of conversion
[o,p]=subtraction(aa,mm,bb,nn)          //function of subtraction
[s,t]=rect2polar(o,p)                  //Iaa function of conversion
[ee,ff]=polar2rect(e,f)                //function of conversion
[oo,pp]=subtraction(ee,aa,ff,bb)    //function of subtraction
[oa,pa]=rect2polar(oo,pp)                //function of conversion
[ob,pb]=subtraction(mm,ee,nn,ff)    //function of subtraction
[oc,pc]=rect2polar(ob,pb)                //function of
```

```
    conversion
od=(eab*s)*cosd(5.55)                          //in
    watt
pd = (eab*oc)*cosd(70.65)                     //in
    watt
pt = od+pd                                 //in watt
ptt = (a*a*zab)+(e*e*real(zbc))+(m*m*real(zca))    //
    in watt
//Results
printf("a. Iab = %.1f A < %d degree \n",a,b)
printf("   Ibc = %.2f A < %.2f degree \n",e,f)
printf("   Ica = %.2f A < %d degree \n",m,n)
printf("b. IAa = %.2f A < %.2f degree \n",s,t)
printf("   IBb = %.2f A < %.2f degree \n",oa,pa)
printf("   ICc = %.1f A < %.2f degree \n",oc,pc)
printf("c. P1 = %.2f W \n",od)
printf("   P2 = %.1f W \n",pd)
printf("d. PT = %.2f W \n",pt)
printf("e. PT = %.2f W \n",ptt)
```

Scilab code Exa 22.9 Example 9

```
//Chapter 22, Example 22.9
clc
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    if x==0 then
        th=-90
    elseif x<0 then
        th=atand(y/x)+180
    else
        th=atand(y/x)
      if (x<0) & (y<0) then
            th=th-180
```

```
        end
    end
endfunction

//Polar to Rectangle conversion
function [r,i]= polar2rect(x,y)
    r=(x*cosd(y))/1000
    i=(x*sind(y))/1000
endfunction

//addition
function[c1,c2]=addition(x1,x2,y1,y2)
    c1 = x1+x2
    c2 = y1+y2
    endfunction
//addition
function[c1,c2]=addition1(x1,x2,x3,y1,y2,y3)
    c1 = x1+x2+x3
    c2 = y1+y2+y3
    endfunction

function[c1,c2]=subtraction(x1,x2,x3,y1,y2,y3)
    c1 = x1-x2-x3
    c2 = y1-y2-y3
    endfunction

function[c1,c2]=subtraction1(x1,x2,y1,y2)
    c1 = x1-x2
        c2 = y1-y2
    endfunction
   //multiplication
function [r,s] = voltdivider(x1,x2,y1,y2)
    r=x1*x2
    s=y1+y2
endfunction
   //multiplication
function [r,s] = voltdivider1(x1,x2,y1,y2)
    r=x1*x2
```

```
    s=y1-y2
endfunction
//division
function[q,x]=division(x1,x2,y1,y2)
       q=x1/x2
       x=y1-y2
   endfunction

function[c1,c2]=subtraction1(x1,x2,y1,y2)
    c1 = x1-x2
    c2 = y1-y2
    endfunction
//Variable Declaration
eca = 200                          //real value
ecao = 120                             //angle in degree
z2 = 200                          //real value
z2o = 0                             //angle in degree
ebc = 200                          //real value
ebco = -120                            //angle in degree
z1 = 166                          //real value
z1o = -90                             //angle in degree
z3 = 200                          //real value
z3o = 0                             //angle in degree
eab = 200                          //real value
eabo = 0                             //angle in degree

//Calculation
[a,b]=voltdivider(eca,z2,ecao,z2o)          //EcaxZ2
[aa,bb]=voltdivider1(ebc,z1,ebco,z1o)         //
   Ebcxz1
[e,f]=polar2rect(a,b)                         //
   function of conversion
[ee,ff]=polar2rect(aa,bb)                         //
   function of conversion
[ac,bc]=addition(e,ee,f,ff)                //function
   of addiion
[ad,bd]=rect2polar(ac,bc)                  //num
[c,d]=voltdivider(z1,z2,z1o,z2o)             //z1xz2
```

```
[ca,da]=voltdivider(z1,z3,z1o,z3o)          //z1xz3
[cb,db]=voltdivider(z2,z3,z2o,z3o)          //z2xz3
[cc,dd]=polar2rect(c,d)                     //function of conversion
[ce,de]=polar2rect(ca,da)                   //function of conversion
[cf,df]=polar2rect(cb,db)                   //function of conversion
[cg,dg]=addition1(cc,ce,cf,dd,de,df)        //function of addition
[ci,di]=rect2polar(cg,dg)                   //function of conversion
[ch,dh]=division(ad,ci,bd,di)               //function of division
[i,j]=voltdivider(eab,z3,eabo,z3o)          //function of multiplication
[ia,ja]=polar2rect(i,j)                     //function of conversion
[ic,jc]=voltdivider(ebc,z1,ebco,z1o)        //function of multiplication
[id,jd]=polar2rect(ic,jc)                   //function of conversion
[ib,jb]=subtraction1(id,ia,jd,ja)           //function of subtraction
[ie,je]=rect2polar(ib,jb)                   //num
[if,jf]=division(ie,ci,je,di)               //function of division

//Results
printf("Icn = %.2f A < %.2f degree \n",ch,dh)
printf("Ibn = %.2f A < %.2f degree \n",if,jf)
```

Chapter 23

Decibels Filters and Bode Plots

Scilab code Exa 23.1 Example 1

```
//Chapter 23, Example 23.1
clc
//Variable Declaration
d1=7/16         //from plot
d2=3/4          //from plot


//Calculation
d=d1/d2
c=10**d
value=(10**2)*c

//Results
printf("Value = %.1f \n",value)
```

Scilab code Exa 23.2 Example 2

```
//Chapter 23, Example 23.2
```

```
clc
//Variable Declaration
a1=0.004
a2=250000
a31=0.08
a32=240
a41=10**4
a42=10**-4
a5=10**4


//Calculation
a=log10(a1)
b=log10(a2)
c=log10(a31)+log10(a32)
d=log10(a41)-log10(a42)
e=log10(a5)

//Results
printf("(a) %.3f \n",a)
printf("(b) %.3f \n",b)
printf("(c) %.3f \n",c)
printf("(d) %.1f \n",d)
printf("(e) %d \n",e)
```

Scilab code Exa 23.3 Example 3

```
//Chapter 23, Example 23.3
clc
//Variable Declaration
vo=1.2                              //output voltage in V
vi=2*10**-3                              //input voltage in mV


```

```
//Calculation
dbv=20*log10(vo/vi)                    //dB gain

//Results
printf("dBv = %.2f dB\n",dbv)
```

Scilab code Exa 23.4 Example 4

```
//Chapter 23, Example 23.4
clc
//Variable Declaration
vo=6.8                         //output voltage in V
db=36                          //dB gain


//Calculation
vi=vo/(10**(db/20))             //input voltage

//Results
printf("Vi = %.2f mV\n",vi*10**3)
```

Scilab code Exa 23.5 Example 5

```
//Chapter 23, Example 23.5
clc
//Variable Declaration
r1=1000                       //resistance in ohm
c1=500*10**-12                //capacitance in farad
f1=100000                     //frequency in hertz
f2=1*10**6                    //frequency in hertz
vi=20                         //input voltage

//Calculation
```

```
fc=1/(2*3.14*r1*c1)                    //cutoff frequency in hertz
xc=1/(2*3.14*c1*fc)                  //impedance in ohm
xc1=1/(2*3.14*c1*f1)                  //impedance in ohm
xc2=1/(2*3.14*c1*f2)                  //impedance in ohm
vo1=vi/sqrt(((r1/xc1)**2)+1)          //output voltage
vo2=vi/sqrt(((r1/xc2)**2)+1)          //output voltage
voc=vi/sqrt(((r1/xc)**2)+1)         //output voltage

//graph
//for frequency response
x= 0.1*fc : 10000 : fc*20
xc11=(2*3.14*c1*x).^-1
vo=vi./sqrt(((r1./xc11)**2)+1)
plot2d('ln',x,vo)
plot(f1,vo1,'*')
plot(f2,vo2,'*')
plot(fc,voc,'*')
xtitle('Frequency response for the low-pass R-C network')
xlabel('f');
ylabel('Vo');

//for normalized plot
scf(2)
plot2d('ln',x,vo/vi)
plot(f1,vo1/vi,'*')
plot(f2,vo2/vi,'*')
plot(fc,voc/vi,'*')
xtitle('Normalized plot for the low-pass R-C network')
xlabel('f');
ylabel('Av');

//Results
printf('(a) fc = %.2f kHz \n',fc)
```

```
printf('(b) At f = 100 kHz, Vo = %.2f V \n',vo1)
printf('     At f = 1 MHz, Vo = %.1f V \n',vo2)
printf("Vi = %.2f mV\n",vi*10**3)
```

Scilab code Exa 23.6 Example 6

```
//Chapter 23, Example 23.6
clc
//Variable Declaration
r1=20000                    //resistance in ohm
c1=1200*10**-12              //capacitance in farad


//Calculation
fc=1/(2*3.14*r1*c1)             //cutoff frequency in
    hertz
f=fc/2                          //frequency in hertz
xc=1/(2*3.14*c1*f)              //capacitive reactancce
    in ohms
av=1/sqrt(1+((xc/r1)**2))     //voltage gain
teta=atand(xc/r1)               //phase in degree

//graph
x= 0 : 500 : fc*3
xc1=(2*3.14*c1*x).^-1
av1=r1./sqrt((r1**2)+(xc1**2))                 //
    normalised gain
av2=xc1./sqrt((r1**2)+(xc1**2))                 //
    normalised gain

teta1=atand(xc1/r1)                 //phase plot
teta2=atand(r1*xc1**-1)                 //phase plot
scf(1)
plot2d(x,av1)
xtitle('Normalised Plot - High Pass')
```

```
xlabel('f (log scale)')
ylabel('Av')

scf(2)
plot2d(x,av2)
xtitle('Normalised Plot - Low Pass')
xlabel('f (log scale)')
ylabel('Av')


scf(3)
plot2d(x,teta1)
xtitle('Phase Plot - High Pass')
xlabel('f (log scale)')
ylabel('teta')

scf(4)
plot2d(x,teta2)
xtitle('Phase Plot - Low Pass')
xlabel('f (log scale)')
ylabel('teta')

clc
//Results
printf('(a) fc = %.2f Hz \n',fc)
printf("Av = Vo / Vi = %.4f < %.2f degree",av,teta)
```

Scilab code Exa 23.7 Example 7

```
//Chapter 23, Example 23.7
clc
//Variable Declaration
r1=1000                    //resistance in ohm
c1=1.5*10**-9              //capacitance in farad
r2=40000                    //resistance in ohm
```

```
c2=4*10**-12           //capacitance in farad


//Calculation
fc1=1/(2*3.14*r1*c1)          //cutoff frequency in
    hertz
fc2=1/(2*3.14*r2*c2)          //cutoff frequency in
    hertz



//graph
x1= 0 : 1000 : fc1*200
x2= 0 : 1000 : fc2*1000
xc1=(2*3.14*c1*x1).^-1                          //
    impedance in ohm
xc2=(2*3.14*c2*x2).^-1                         //
    impedance in ohm
av1=r1./sqrt((r1**2)+(xc1**2))                  //
    normalised gain
av2=xc2./sqrt((r2**2)+(xc2**2))                  //
    normalised gain


scf(1)
//plot2d('ln',[x1 x2],[av1 av2])
plot2d('ln',x1,av1)
xtitle('Normalised Plot - High Pass')
xlabel('f')
ylabel('Vo')

scf(2)
plot2d('ln',x2,av2)
xtitle('Normalised Plot - Low Pass')
xlabel('f')
ylabel('Vo')
```

```
clc
//Results
printf('(a) High Pass filter , fc = %.1f kHz \n',fc1/1000)
printf('    Low Pass filter , fc = %.1f kHz \n',fc2/1000)
printf("(c) Av = Vo / Vi = %.4f < %.2f degree",av,teta)
```

Scilab code Exa 23.8 `Example 8`

```
//Chapter 23, Example 23.8
clc
//Variable Declaration
l1=1*10**-3                          //inductance in henry
c1=0.01*10**-6                  //capacitance in farad
r1=33                   //resistance in ohm
r2=2                   //resistance in ohm
vi=20*10**-3                   //input voltage

//Calculation
fs=1/(2*3.14*sqrt(l1*c1))           //frequency in hertz
xl=2*3.14*fs*l1                     //inductive reactance
qs=xl/(r1+r2)                        //Q factor
bw=fs/qs                              //bandwidth in hertz
vomax=(r1*vi)/(r1+r2)                 //max voltage
vo=0.707*vi*(r1/(r1+r2))                //voltage


//Insufficient data on textbook problem to plot graph
//graph
```

```
x=0:1000:fs*2
//ztp1=(((2*3.14*x*l1)/(r1+r2))**2)*r2
xl=2*3.14*x*l1
xc=1/(2*3.14*x*c1)
zs=sqrt((r2**2)+(xl-xc')**2)
vo1=(zs*vi)./(zs+r1)
plot2d(x,vo1)
xtitle('Passband response')
xlabel('f')
ylabel('Vo')


//Results
clc
printf('(a) fs = %.1f Hz \n',fs)
printf('Qs = %.2f \n',qs)
printf('BW = %.2f kHz \n',bw/1000)
printf('Vomax = %.2f mV \n', vomax*1000)
```

Scilab code Exa 23.9 Example 9

```
//Chapter 23, Example 23.9
clc
//Variable Declaration
c=500*10**-12                                    //capacitance in farad
f1=200*10**3                                     //frequency in hertz
f2=600*10**3                                     //frequency in hertz


//Calculation
l1=1/(4*(%pi**2)*(f2**2)*c)                      //inductance in henry
```

```
xls=2*%pi*f1*l1                          //inductive
   reactance in ohms
xc=1/(2*%pi*f1*c)                          //
   capacitive reactance in ohms
j=xc-xls                                 //for
   series elements
lp=j/(2*%pi*f1)                //inductance in henry

//Results
printf("Ls = %.1f uH\n",l1*10**6)
printf("Lp = %.2f mH",lp*10**3)
```

Scilab code Exa 23.10 Example 10

```
//Chapter 23, Example 23.10
clc
//Variable Declaration
r1=1000                //resistance in ohm
c1=0.1*10**-6          //capacitance in farad



//Calculation
f1=1/(2*3.14*r1*c1)          //cutoff frequency in
   hertz

//for magnitude plot
f = 100:10:10*f1;
av = (1+(f1./f).^2).^(-1/2);          //-10*log10
av1 = -20*log10(f1/f1);
f2 = f1/10;
av2 = -20*log10(f1/f2);
f3 = f1/2;
av3 = -20*log10(f1/f3);
f4 = f1;
```

```
av4 = -20*log10(f1/f4);
x = [f2 f3 f4 f1];
y = [av2 av3 av4 av1];
scf(1)
gainplot(f,av);
a = gca();
a.y_location = 'left';
a.x_location = 'top';
a.x_label.text = 'frequency';
a.y_label.text = 'Av';
a.title.text = 'Magnitude Plot';
plot2d(x,y);

//for phase plot
f01 = 10:10:10*f1;
teta = atand(f1./f)
f11=f1
teta1 = atand(f1./f11)
f12=f1/10
teta2 = atand(f1./f12)
f13=f1*10
teta3 = atand(f1./f13)
x1 = [f11 f12 f13 ];
y1 = [teta1 teta2 teta3];
scf(2)
phaseplot(f,av,teta)
b = gca();
b.x_label.text = 'frequency';
b.y_label.text = 'Av';
b.title.text = 'Phase Plot';
plot2d(x1,y1)

f5=1000                                        //frequency in hertz
av3 = 20*log10(1/sqrt((((f1/f5)**2)+1)));      //gain


//Results
```

```
printf('(a) fc = %.2f Hz \n',f1)
printf('      Magnitude plot shown in window 1 \n')
printf('(b) |AvdB| = %.2f dB \n',av3)
printf('(c) Phase plot shown in window 2')
```

Scilab code Exa 23.11 Example 11

```
//Chapter 23, Example 23.11
clc
//Variable Declaration
r1=9100                   //resistance in ohm
c1=0.47*10**-6             //capacitance in farad
r2=1000                   //resistance in ohm


//Calculation
r12=(r1*r2)/(r1+r2)          //parallel resistance in ohm
f1=1/(2*3.14*r1*c1)          // frequency in hertz
fc=1/(2*3.14*r12*c1)          // frequency in hertz
al=-20*log10((r1+r2)/r2)       //maximum low-level attentuation

teta1=-(atand(f1/f1))+(atand(fc/f1))     //in degrees
teta2=-(atand(f1/fc))+(atand(fc/fc))     //in degrees
teta3=-(atand(f1/120))+(atand(fc/120))     //in degrees

//for magnitude plot
f = 1:10:8*fc;
av=((1+((f1./f).^2))./(1+((fc./f).^2))).^(1/2)           //-10*log10
f2 = f1
av2=10*log10((1+((f1./f2).^2))./(1+((fc./f2).^2)))
f3 = fc
```

```
av3=10*log10((1+((f1./f3).^2))./(1+((fc./f3).^2)))
x = [f2 f3];
y = [av2 av3];
scf(1)
gainplot(f,av);
a = gca();
a.y_location = 'left';
a.x_location = 'top';
a.x_label.text = 'frequency';
a.y_label.text = 'Av';
a.title.text = 'AcdB versus frequency for the filter';
plot2d(x,y);
plot(f2,av2,'*');
plot(f3,av3,'*');

//for phase plot
teta=-(atand(f1./f))+(atand(fc./f))
scf(2)
plot2d('ln',f,teta)
plot(f1,teta1,'*')
plot(fc,teta2,'*')
plot(120,teta3,'*')
xtitle('phase vs frequency for the filter')
xlabel('frequency')
ylabel('phase in degree')


//Results
printf('(a) f1 = %.1f Hz \n',f1)
printf('    fc = %.2f Hz \n',fc)
printf('(b) At f1 = 32.2 Hz, teta = %.2f degree \n',teta1)
printf('    At fc = 376.26 Hz, teta = %.2f degree \n',teta2)
printf('    At frequency midway between fc and f1, like 120 Hz, teta = %.2f degree \n',teta3)
```

Scilab code Exa 23.12 Example 12

```
//Chapter 23, Example 23.12
clc
f1=50                          //frequency in hertz
f2=200                         //frequency in hertz
f3=10000                       //frequency in hertz
f4=20000                       //frequency in hertz
f8=8500                       //frequency in hertz
f9=1000                      //frequency in hertz

//Calculations
AvdB1=-20*log10(sqrt((1+(f1/f3)^2)*(1+((f2/f3)^2))
   *(1+(f3/f3)^2)*(1+((f3/f4)^2))))
                                          //in dB
AvdB2=-20*log10(sqrt((1+(f8/f3)^2)*(1+((f8/f4)^2))))
                 //in dB
BW=f8-f2                                 //in hertz
AvdB3=-20*log10(sqrt((1+(f1/f9)^2)*(1+((f2/f9)^2))))
                    //in dB

//phase plot
f = 10:100:200000;
teta=(atand(f1./f))+(atand(f2./f))-(atand(f./f3))-(
   atand(f./f4))
scf(1)
plot2d('ln',f,teta)
xgrid(1)
a = gca();
a.y_location = 'left';
a.x_location = 'top';
a.x_label.text = 'frequency';
a.y_label.text = 'Phase';
a.title.text = 'Phase response';
```

```
//for magnitude plot
av = -((1+(f1./f)^2).*(1+((f2./f).^2)))^(-1/2);
av1 = -20*log10(f1/f1);
f12 = f1/10;
av2 = -20*log10(f1/f12);
f13 = f1/4;
av3 = -20*log10(f1/f13);
f14 = f1/2;
av4 = -20*log10(f1/f14);

x = [f12 f13 f14 f1];
y = [av2 av3 av4 av1];
scf(2)
gainplot(f,av);
a = gca();
a.y_location = 'left';
a.x_location = 'top';
a.x_label.text = 'frequency';
a.y_label.text = 'Av';
a.title.text = 'Bode';
plot2d(x,y);

f1=f2
av1 = -20*log10(f1/f1);
f12 = f1/10;
av2 = -20*log10(f1/f12);
f13 = f1/4;
av3 = -20*log10(f1/f13);
f14 = f1/2;
av4 = -20*log10(f1/f14);

x = [f12 f13 f14 f1];
y = [av2 av3 av4 av1];

gainplot(f,av);
a = gca();
a.y_location = 'left';
```

```
a.x_location = 'top';
a.x_label.text = 'frequency';
a.y_label.text = 'Av';
a.title.text = 'Bode';
plot2d(x,y);




av = -((1+(f./f3)^2).*(1+((f./f4).^2)))^(-1/2);

gainplot(f,av);
a = gca();
a.y_location = 'left';
a.x_location = 'top';
a.x_label.text = 'frequency';
a.y_label.text = 'Av';
a.title.text = 'Bode';
plot2d(x,y);


f1 = f3;
f = 0.1*f1:100:10*f1;
av = (1+(f/f1)^2)^(-1/2);
av1 = -20*log10(f1/f1);
f2 = f1*10;
av2 = -20*log10(f2/f1);
f3 = f1*4;
av3 = -20*log10(f3/f1);
f4 = f1*2;
av4 = -20*log10(f4/f1);

x = [f1 f4 f3 f2];
y = [av1 av4 av3 av2];
plot2d(x,y);

f1 = f4;
f = 0.1*f1:100:10*f1;
```

```
av = (1+(f/f1)^2)^(-1/2);
av1 = -20*log10(f1/f1);
f2 = f1*10;
av2 = -20*log10(f2/f1);
f3 = f1*4;
av3 = -20*log10(f3/f1);
f4 = f1*2;
av4 = -20*log10(f4/f1);

x = [f1 f4 f3 f2];
y = [av1 av4 av3 av2];
plot2d(x,y);


printf('for 10kHz,\n')
printf('AvdB = %.1f dB \n',AvdB1)
printf('for 8.5kHz, \n')
printf('AvdB = %.1f dB \n',AvdB2)
printf('BW = %.1f kHz \n',BW/1000)
printf('for midrange of bandwidth 1kHz, \n')
printf('AvdB = %.1f dB \n',AvdB3)
```

Chapter 24

Pulse Waveforms and the RC Response

Scilab code Exa 24.1 Example 1

```
//Chapter 24, Example 24.1
clc
//Initialisation
t1=12           //time period in ms
t2=7            //time period in ms
v1=8           //voltage
v2=4           //voltage
v3=12           //voltage
v4=11           //voltage
vb=-4           //voltage from graph

//Calculation
tp=t1-t2            //time period in ms
vmax=v1+v2           //maximum amplitude voltage
v=(v3+v4)/2          //voltage
vp=((v3-v4)/v)*100          //tilt in percent

//Result
printf("(a) Positive-going \n")       //from graph
```

```
printf("(b) Vb = %d V\n",vb)
printf("(c) tp = %d ms \n",tp)
printf("(d) Vmax = %d V \n",vmax)
printf("(e) V (tilt in percentage) = %.3f percent",
   vp)
```

Scilab code Exa 24.2 Example 2

```
//Chapter 24, Example 24.2
clc
//Initialisation
vb=0               //voltage from graph
tilt=0              //from graph
div=4                 //amplitude in div
perdiv=10             //mV per div
div1=3.2                //division
perdiv1=5               //microsec per div
div2=0.4                //division
div3=0.8                //division



//Calculation
amp=div*perdiv                  //amplitude
tp=div1*perdiv1                   //from graph 24.9
tr=div2*perdiv1                   //from graph 24.9
tf=div3*perdiv1                   //from graph 24.9

//Result
printf("(a) Positive-going \n")        //from graph
printf("(b) Vb = %d V\n",vb)
printf("(c) tilt = %d percent \n",tilt)
printf("(d) amplitude = %d mV \n",amp)
printf("(e) tp = %d us \n",tp)
printf("(f) tr = %d us \n",tr)
```

```
printf("      tf = %d us \n",tf)
```

Scilab code Exa 24.3 Example 3

```
//Chapter 24, Example 24.3
clc
//Initialisation
t1=15          //timeperiod in microsec
t2=6           //timeperiod in microsec
t3=8           //timeperiod in microsec


//Calculation
T=t1-t2             //timeperiod in microsec
prf=1/T             //frequency
tp=t3-t2            //timeperiod in microsec
duty=(tp/T)*100            //duty cycle

//Result
printf("(a) Pulse repition Frequency, prf = %.2f kHz
    \n",prf*1000)
printf("(b) Duty Cycle = %.2f percent \n",duty)
```

Scilab code Exa 24.4 Example 4

```
//Chapter 24, Example 24.4
clc
//Initialisation
div1=3.2            //no of division
ms=1                //ms per divisions
div2=0.8            //no of division


```

```
//Calculation
T=div1*ms              //timeperiod in microsec
tp=div2*ms               //frequency
prf=1/T                //timeperiod in microsec
duty=(tp/T)*100             //duty cycle

//Result
printf("(a) Pulse repition Frequency, prf = %.2f Hz
   \n",prf*1000)
printf("(b) Duty Cycle = %.2f percent \n",duty)
```

Scilab code Exa 24.5 `Example 5`

```
//Chapter 24, Example 24.5
clc
//Initialisation
div1=2.6             //no of division
ms=10                 //microsec per divisions
div2=0.2             //no of division



//Calculation
T=div1*ms              //timeperiod in microsec
tp=div2*ms               //frequency
prf=1/T                //timeperiod in microsec
duty=(tp/T)*100             //duty cycle

//Result
printf("(a) Pulse repition Frequency, prf = %d Hz \n
   ",prf*10**6)
printf("(b) Duty Cycle = %.2f percent \n",duty)
```

Scilab code Exa 24.6 Example 6

```
//Chapter 24, Example 24.6
clc
//Initialisation
t1=12*10**-6            //time period in ms
t2=2*10**-6             //time period in ms
v1=8*10**-3            //voltage
v2=4*10**-6            //voltage
v3=2*10**-3            //voltage
v4=6*10**-6            //voltage




//Calculation
T=t1-t2                      //timeperiod in microsec
g=((v1*v2)+(v3*v4))/T     //Average Value in Volts

//Result
printf("Average Value, G = %.1f mV \n",g*10**3)
```

Scilab code Exa 24.7 Example 7

```
//Chapter 24, Example 24.7
clc
//Initialisation
duty = 0.28               //duty cycle
Vp = 7                    //peak value
Vb = -3                    //base-line voltage
t1=0.28          //time period
t2=0.72           //time period



```

```
//Calculation
Vav = (duty*Vp)+((1-duty)*Vb)        //Average Value in Volts
g=(Vp*t1)+(Vb*t2)                    //Average Value in Volts

//Result
printf("(a) Average Value, Vav = %.1f V \n",Vav)
printf("(c) G = %.1f V \n",g)
```

Scilab code Exa 24.8 Example 8

```
//Chapter 24, Example 24.8
clc
//Initialisation
vi=2                  //voltage in mV
e=8                  //voltage in mV
r=100*10**3               //resistance in ohm
c=10**-6                //capacitance in farad


//Calculation
tau=r*c                    //time constant
vr=e-vi                 //voltage
irm=vr/r                  //current in amperes

//Result
printf("(a) Vc = %d V %d V e^(-t/%.f ms) \n",e,(vi-e),tau*1000)
printf("(b) Irmax = %.2f mA e^(-t/%.f ms)",irm*1000,tau*1000)


//Graph
x1=0:0.1:0.8
```

```
vc=e+(vi-e)*exp(-x1/tau)
ic=irm*exp(-x1/tau)
scf(221)
plot(x1,vc)
xlabel("t(s)")
ylabel("vc(V)")
xtitle("Vc for the network of Fig. 24.20.")
scf(222)
plot(x1,ic)
xlabel("t(s)")
ylabel("ic (mA)")
xtitle("ic for the network of Fig. 24.20.")
```

Scilab code Exa 24.9 Example 9

```
//Chapter 24, Example 24.9
clc
//Initialisation
vi=-4                          //voltage in mV
vf=10                          //voltage in mV
r=1*10**3                   //resistance in ohm
c=0.01*10**-6                       //capacitance in farad


//Calculation
tau=r*c                         //time constant
t=-tau*log(vf/(vf-vi))       //time in seconds

//Result
printf("Vc = %d mV %d mV e^(-t/%.f ms) \n",vf,(vi-vf
    ),tau*10**6)
printf("t = %.2f us",(t*10**6))

//Graph
x1=0:(10**-6):(80*10**-6)
```

```
vc=vf+(vi-vf)*exp(-x1/tau)
plot(x1,vc)
xlabel("t(s)")
ylabel("vc(V)")
xtitle("Vc for the network of Fig. 24.22.")
```

Scilab code Exa 24.10 Example 10

```
//Chapter 24, Example 24.10
clc
//Initialisation
f=1000                    //frequency in hertz
r=5*10**3               //resistance in ohm
c=0.01*10**-6                   //capacitance in farad
vi=0                      //for charging phasse,
   voltage
vf=10                      //voltage in mV
vi2=10                      //for discharging phasse,
   voltage in mV
vf2=0                      //voltage
t1=0                        //for charging phase for
   ic
vr=vf


//Calculation
t=1/f                     //time period in sec
tp=t/2                     //phase time in sec
tau=r*c                     //time constant in sec
irmax=vr/r                  //current in milliampere

//Result
printf("(a) tp = %.1f ms \n",tp*10**3)
printf("tau = %.2f ms \n",tau*10**3)
printf("Therefore, tp = 10*tau = T/2 \n")
```

```
printf("(b)for charging phase,  Vc = %d mV(1 - e^(-t/%.2f ms)) \n",vf,tau*1000)
printf("  for discharging phase, Vc = %d mV(e^(-t/%.2f ms)) \n",vf,tau*1000)
printf("(c) Irmax = %.2f uA e^(-t/%.f ms)",irmax*10**3,tau*1000)


//Graph
x1=0:(0.1*10**-3):(t)
x2=tp:(0.1*10**-3):(t)
x3=0:(0.1*10**-3):(tp)
x4=tp:(0.1*10**-3):(t)
vc=vf+(vi-vf)*(exp(-x1/tau))                //for Vc - Charging phase
vc2=vf2+(vi2-vf2)*exp(-x2/tau)              //for Vc - Discharging phase
ic=irmax*exp(-x3/tau)                       //for Ic - Charging phase
ic2=-irmax*(exp(-x3/tau))                   //for Ic - Discharging phase
subplot(221)
plot(x1,vc)
xlabel("t(s)")
ylabel("Vc(V)")
xtitle("Vc for the R-C network - Charging Phase")
subplot(222)
plot(x2,vc2)
xlabel("t(s)")
ylabel("Vc (V)")
xtitle("Vc for the R-C network - Discharging Phase")
subplot(223)
plot(x3,ic)
xlabel("t(s)")
ylabel("ic (mA)")
xtitle("Ic for the R-C network - Charging Phase")
subplot(224)
plot(x4,ic2)
```

```
xlabel("t(s)")
ylabel("ic (mA)")
xtitle("Ic for the R-C network - Discharging Phase")
```

Scilab code Exa 24.11 Example 11

```
//Chapter 24, Example 24.11
clc
//Initialisation
f=10*10**3                          //frequency in hertz
r=5*10**3                     //resistance in ohm
c=0.01*10**-6                       //capacitance in farad
vi=0                            //for charging phasse, voltage
vf=10                            //voltage in mV
vf2=0                            //voltage
t1=0                              //for charging phase for ic
vr=vf


//Calculation
t=1/f                          //time period in sec
tp=t/2                          //phase time in sec
tau=r*c                          //time constant in sec
irmax=vr/r                        //current in milliampere

//first interval
    vc1=vf+(vi-vf)*(exp(-tau/tau))        //in voltage
    vi1=vc1                                //in voltage
    vc2=vc1*exp(-tau/tau)                  //in voltage
    irmax2=(vi-vc1)/r                      //in milliamp

```

```
//for graph
x1=0:(0.01*10**-3):(tp)
x2=0:(0.01*10**-3):(tp)
vcc=vf+((vi-vf)*(exp(-x1/tau)))                //
   for vc - Charging phase
vcd=vc1*(exp(-x2/tau))                          //
   for vc - Discharging phase

icc=(irmax*10**-3)*exp(-x1/tau)
   //for Ic - Charging phase
icd=(irmax2*10**-3)*(exp(-x2/tau))
   //for Ic - Discharging phase
scf(1)
subplot(221)
plot(x1,vcc)
xlabel("t(s)")
ylabel("Vc(V)")
xtitle("Vc for the R-C network - Charging Phase"
   )
subplot(222)
plot(x2,vcd)
xlabel("t(s)")
ylabel("Vc (V)")
xtitle("Vc for the R-C network - Discharging
   Phase")
subplot(223)
plot(x1,icc)
xlabel("t(s)")
ylabel("ic (mA)")
xtitle("Ic for the R-C network - Charging Phase"
   )
subplot(224)
plot(x2,icd)
xlabel("t(s)")
ylabel("ic (mA)")
xtitle("Ic for the R-C network - Discharging
   Phase")
```

```
//second interval
    vc1=vf+(vc2-vf)*(exp(-tau/tau))   //in voltage
    vi1=vc1   //in voltage
    vc2=vc1*exp(-tau/tau)   //in voltage
    irmax2=(vi-vc1)/r   //in milliamp
    x1=0:(0.01*10**-3):(tp)
    x2=0:(0.01*10**-3):(tp)
    vcc2=vf+((vc2-vf)*(exp(-x1/tau)))   //for vc - Charging phase
    vcd2=vc1*(exp(-x2/tau))   //for vc - Discharging phase
    icc=(irmax*10**-3)*exp(-x1/tau)   //for Ic - Charging phase
    icd=(irmax2*10**-3)*(exp(-x2/tau))   //for Ic - Discharging phase
    scf(2)
    subplot(221)
    plot(x1,vcc2)
    xlabel("t(s)")
    ylabel("Vc(V)")
    xtitle("Vc for the R-C network - Charging Phase")
    subplot(222)
    plot(x2,vcd2)
    xlabel("t(s)")
    ylabel("Vc (V)")
    xtitle("Vc for the R-C network - Discharging Phase")
    subplot(223)
    plot(x1,icc)
    xlabel("t(s)")
    ylabel("ic (mA)")
    xtitle("Ic for the R-C network - Charging Phase")
```

```
subplot(224)
plot(x2,icd)
xlabel("t(s)")
ylabel("ic (mA)")
xtitle("Ic for the R–C network − Discharging
   Phase")
```

Chapter 25

Nonsinusoidal Circuits

Scilab code Exa 25.2 Example 2

```
//Chapter 25, Example 25.2
clc
//Initialisation
A0=20
A10=0
A11=5*10**-3
A12=0
A13=0
A20=8
A21=0
B1=12
B2=0

//Calculation
v=A0                        //from graph 25.12
i=A10+A11+A12+A13             //fourier series
v2=A20+A21          //fourier series
B=B1+B2         //fourier series

//Result
printf("(a) v = %d \n",v)
```

```
printf("(b) i = %d X 10^-3 sin wt \n",i*10**3)
printf("(c) v = %d + %d cos wt",v2,B)
```

Scilab code Exa 25.3 Example 3

```
//Chapter 25, Example 25.3
clc
//Initialisation
funcprot();
//Polar to Rectangle Conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction

//Accordng to phasor algebra
a1=2
[r1,i1]=polar2rect(1,90)
[r2,i2]=polar2rect(2,0)

//Calculation
ar1=r1+r2
ai1=i1+i2
[ar2,ath2] = rect2polar(ar1,ai1)
x = 0 : 0.1 : 10
y = a1 + (ar2*sin(x+ath2*3.14/180))


//Result
```

```
plot(x, y)
xlabel("a = wt");
ylabel("V");
title('v = 2 + 1 cos a + 2 sin a');
```

Scilab code Exa 25.4 Example 4

```
//Chapter 25, Example 25.4
clc
//Initialisation
funcprot();
//Polar to Rectangle Conversion
function [r,i]= polar2rect(x,y)
    r=x*cosd(y)
    i=x*sind(y)
endfunction

//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction

//Accordng to phasor algebra
[r1,i1]=polar2rect(1,0)
[r2,i2]=polar2rect(1,0)

//Calculation
ar1=r1+r2
ai1=i1+i2
[ar2,ath2] = rect2polar(ar1,ai1)
x = 0 : 0.1 : 10
y = ar2*sin(x+ath2*3.14/180)


```

```
//Result
plot(x, y)
xlabel("a = wt");
ylabel("V");
title('v = 2 + 1 cos a + 2 sin a');
```

Scilab code Exa 25.5 Example 5

```
//Chapter 25, Example 25.5
clc
//Initialisation
v=4                    //in voltage
v2=6                    //in voltage

//Calculation
x = 0 : 0.1 : 10
y = v + v2*sin(x)
vrmax=sqrt((v**2)+(v2**2)/2)

//Result
plot(x, y)
xlabel("wt");
ylabel("V");
title('v = 4 + 6 sin wt');
printf("(b) Vrmax = %.3f V",vrmax)
```

Scilab code Exa 25.6 Example 6

```
//Chapter 25, Example 25.6
clc
//Initialisation
v=20                    //in voltage

```

```
//Calculation
v0=(4/%pi)*v
v1=v0
v2=(v0/3)
v3=(v0/5)
v4=(v0/7)
v5=(v0/9)
v6=(v0/11)
vrms=sqrt(((v1**2)+(v2**2)+(v3**2)+(v4**2)+(v5**2)+(
   v6**2))/2)

//Result
printf("(b) Vrmax = %.3f V",vrms)
```

Scilab code Exa 25.7 Example 7

```
//Chapter 25, Example 25.7
clc
//Initialisation
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction

i1=0
R=3
vc=12
C=(1/8)
E=10/sqrt(2)

//Calculation
vr=i1*R
xc=1/(2*C)

```

```
z=complex(R,xc)
[r1,th1]=rect2polar(real(z),-imag(z))
i2=E/r1
vr2=i2*R
vc1=i2*xc
vca=-th1-90

irms = sqrt((i2**2))
vrms = sqrt((vr2**2))
vcrms = sqrt((vc**2)+(vc1**2))

P=(irms**2)*R
//Result
printf("(a)  1. VR = %d V \t\n \tI = %d \t\n\tVc = %d V\t\n",vr,i1,vc)
printf("     2. for AC supply \n")
printf("        i = %.2f A < %.2f \n",i2,-th1)
printf("        Vr = %.2f A < %.2f \n",vr2,-th1)
printf("        Vc = %.2f A < %.2f \n",vc1,vca)
printf("(b)  Irms = %.3f A \n\tVrms = %.3f V \n\tVcrms = %.3f V\t\n",irms,vrms,vcrms)
printf("(c)  P = %d W \n",P)
```

Scilab code Exa 25.8 Example 8

```
//Chapter 25, Example 25.8
clc
//Initialisation
//Rectangle to Polar Conversion
function [r,th] = rect2polar(x,y)
    r=sqrt((x**2)+(y**2))
    th=atand(y/x)
endfunction

xl=0
```

```
R=6
e0=63.6
w=377
L=0.1
e1=70.71
e2=29.98
e2a=-90
w2=754

//Calculation
zt=R
i0=e0/R
vr0=i0*R
vl0=0
p0=(i0**2)*R

xl1=w*L
zt1=complex(R,xl1)
[zt1r,zt1a]=rect2polar(real(zt1),imag(zt1))
i1=e1/zt1r
vr1=i1*R
vl1=i1*xl1
vl1a=90-zt1a
p1=(i1**2)*R


xl2=w2*L
zt2=complex(R,xl2)
[zt2r,zt2a]=rect2polar(real(zt2),imag(zt2))
i2=e2/zt2r
i2a=e2a-zt2a
vr2=i2*R
vl2=i2*xl2
vl2a=90+i2a
p2=(i2**2)*R
Irms=sqrt((i0**2)+(i1**2)+(i2**2))
VRms=sqrt((e0**2)+(vr1**2)+(vr2**2))
Vlrms=sqrt((vl1**2)+(vl2**2))
```

```
//Result
printf("The Fourier series expansion for i is \n")
printf("i = %.1f + sqrt(2)(%.2f)sin(%d*t - %.2f) +
    sqrt(2)(%.3f)sin(%d*t - %.2f) \n",i0,i1,w,zt1a,i2
    ,w2,-i2a)
printf("Irms = %.2f A \n\n",Irms)
printf("The Fourier series expansion for VR is \n")
printf("Vr = %.1f + sqrt(2)(%.2f)sin(%d*t - %.2f) +
    sqrt(2)(%.3f)sin(%d*t - %.2f) \n",e0,vr1,w,zt1a,
    vr2,w2,-i2a)
printf("Vrms = %.2f V \n\n",VRms)
printf("The Fourier series expansion for VL is \n")
printf("VL = sqrt(2)(%.2f)sin(%d*t - %.2f) + sqrt(2)
    (%.3f)sin(%d*t - %.2f) \n",vl1,w,vl1a,vl2,w2,-
    vl2a)
printf("Vlrms = %.2f V \n\n",Vlrms)
```

Chapter 26

System Analysis An Introduction

Scilab code Exa 26.1 Example 1

```
//Chapter 26, Example 26.1, page 1153
clc
//Initialisation
eg=100*10**-3                    //in Volt
ei=96*10**-3                     //in Volt
rz=100                           //in Ohms

//Calculation
vr=eg-ei                   //in Volt
i=vr/rz                   //in Ampere
zi=ei/i                          //in Ohms

//Result
printf("Input Impedance, Zi = %.1f kOhm",zi/1000)
```

Scilab code Exa 26.2 Example 2

```
//Chapter 26, Example 26.2, page 1154
clc
//Initialisation
eg=2                            //in Volt
eo=1.92                    //in Volt
rz=2*10**3                              //in Ohms

//Calculation
vr=eg-eo                    //in Volt
i=vr/rz                    //in Ampere
zo=eo/i                         //in Ohms

//Result
printf("Output Impedance, Zi = %.1f kOhm",zo/1000)
```

Scilab code Exa 26.3 Example 3

```
//Chapter 26, Example 26.3, page 1154
clc
//Initialisation
Vr=2*10**-3                                  //in Volt
Rz=10                                           //in Ohms
Ei=50*10**-3                                      //in Volt
phi1=150                                       //in degree

//Calculation

i=Vr/Rz                          //in Ampere
zi=Ei/i                         //Zi in Ohms
phi2=180-phi1                   //phase angle between Eg
    and Vrz
re=zi*cos(phi2*3.14/180)                 //real number,
   R
im=zi*sin(phi2*3.14/180)                 //Imaginary
   number, Xl
```

```
//Result
printf("Zi = %.2f ohm + j %.1f ohm ",re,im)
```

Scilab code Exa 26.4 Example 4

```
//Chapter 26, Example 26.4, page 1157
clc
//Initialisation
Eo=-20                  //output voltage
Ei=4*10**-3             //input voltage
Rl=2.2*10**3             //load Resistance in Ohm
Ro=50*10**3            //output Resistance in Ohm
Zi=1*1*10**3            //input impedance in Ohm
Rg=1*10**3              //Resistance in Ohm


//Calculation
Avn=Eo/Ei                          //no-load voltage gain
Av=Avn*Rl/(Rl+Ro)                  //loaded voltage gain
Avr=Av*Zi/(Zi+Rg)                  //loaded voltage gain
Ro=Rl*((Avn/Av)-1)                  //Resistance in Ohm

//Result
printf("(a) Avnl = %d \n",Avn)
printf("(b) Av = %.2f \n",Av)
printf("(c) Avr = %.2f \n",Avr)
printf("(d) Ro = %d kOhm",Ro/1000)
```

Scilab code Exa 26.5 Example 5

```
//Chapter 26, Example 26.5, page 1161
clc
```

```
//Initialisation
Rl=4.7*10**3          //load Resistance in Ohm
Ro=40*10**3           //output Resistance in Ohm
Avn=-960              //no-load voltage gain
Ri=2.7*10**3         //input Resistance in Ohm



//Calculation
Av=Avn*Rl/(Rl+Ro)                 //loaded voltage gain
Ai=(-Avn)*Ri/(Rl+Ro)                 //voltage gain, Ai
Rl2=2*Rl                              //Resistance in Ohm
Av2=Avn*Rl2/(Rl2+Ro)                 //loaded voltage gain
Ai2=(-Avn)*Ri/(Rl2+Ro)                 //voltage gain, Ai
Air=(-Av)*Ri/Rl                          //voltage gain, Air
Ag=(Av**2)*Ri/Rl                       //voltage gain, Ag
Ai2=-Ag/Av                             //voltage gain, Ai

//Result
printf("(a) Av = %.2f \n",Av)
printf("(b) Avi = %.2f \n",Ai)
printf("(c) Av = %.2f, Ai = %.2f\n",Av2,Ai2)
printf("(d) Air = %.2f \n",Air)
printf("(e) Ag = %.2f \n",Ag)
printf("(f) Ai = %.2f",Ai2)
```

Scilab code Exa 26.6 Example 6

```
//Chapter 26, Example 26.6, page 1163
clc
//Initialisation
Avn1=1                //no-load voltage gain
Avn2=-600                //no-load voltage gain
Avn3=-1200                //no-load voltage gain
Zi1=50*10**3         //input impedance in Ohm
```

```
Zi2=1.8*10**3         //input impedance in Ohm
Zi3=1.2*10**3         //input impedance in Ohm
Rl=3.3*10**3        //input impedance in Ohm
Ro=25                 //output Resistance in Ohm
Ro2=40*10**3          //output Resistance in Ohm
Ro3=50*10**3          //output Resistance in Ohm


//Calculation
Av1=Avn1*Zi2/(Zi2+Ro)           //loaded voltage gain
Av2=Avn2*Zi3/(Zi3+Ro2)          //loaded voltage gain
Av3=Avn3*Rl/(Rl+Ro3)           //loaded voltage gain
Ai=(-Avn1)*Zi1/(Zi2+Ro)            //voltage gain, Ai
Ai2=(-Avn2)*Zi2/(Zi3+Ro2)            //voltage gain, Ai
Ai3=(-Avn3)*Zi3/(Rl+Ro3)            //voltage gain, Ai
Rl2=2*Rl                         //Resistance in Ohm
Av21=Avn1*Rl2/(Rl2+Ro)            //loaded voltage gain
Ai20=(-Avn1)*Zi1/(Zi2+Ro)             //voltage gain, Ai
Air=(-Avn1)*Zi1/Rl                      //voltage gain, Air
Ag=(Avn1**2)*Zi1/Rl                    //voltage gain, Ag
Ai21=-Ag/Avn1                        //voltage gain, Ai
Avt=Av1*Av2*Av3                       //voltage gain
Ait=Ai*Ai2*Ai3                       //voltage gain
Agt=-Avt*Ait                       //voltage gain
Av2=Av1*Av2                       //voltage gain
Ai22=Ai*Ai2                       //voltage gain
Ai23=Av2*Zi1/Zi3                        //voltage gain
Ag2=Av2*Ai23                       //voltage gain
Ag22=(Av2**2)*(Zi1/Zi3)                          //voltage gain
Avt2=Avn1*Avn2*Avn3                       //voltage gain


```

```
//Result
printf("(a) Av1 = %.3f, Av2 = %.3f, Av3 = %.3f \n Ai1 = %.3f, Ai2 = %.3f, Ai3 = %.3f \n",Av1,Av2,Av3,Ai,Ai2,Ai3)
printf("(b) Avt = %.2f, Ait = %.2f \n",Avt,Ait)
printf("(c) Agt = %.2f x 10^6 \n",Agt/10**6)
printf("(d) Av2 = %.2f, Ai2 = %.3f \n",Av2,Ai22)
printf("(e) Ai2 = %.2f \n",Ai23)
printf("(f) Ag2 = %.2f \n",Ag2)
printf("(g) Ag2 = %.2f \n",Ag22)
printf("(h) Avt = %.1f x 10^5",Avt2/(10**5))
```

Scilab code Exa 26.7 Example 7

```
//Chapter 26, Example 26.7, page 1166
clc
funcprot(0);
// A = p2z(R,Theta) - Convert from polar to rectangular form.
//    R is a matrix containing the magnitudes
//    Theta is a matrix containing the phase angles (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

//Initialisation
z1 = p2z(3,0)
z2 = p2z(5,90)
z3 = p2z(4,-90)


//Calculation
z11 = z1+z3
z12 = z3
```

```
z22 = z2+z3

//Result
printf("Z11 = %d ohm - j %d ohm \n",real(z11),-imag(z11))
printf("Z12 = - j %d ohm \n",-imag(z12))
printf("Z22 = j %d ohm \n",imag(z22))
```

Scilab code Exa 26.9 Example 9

```
//Chapter 26, Example 26.9, page 1171
clc
funcprot(0);
// A = p2z(R,Theta) - Convert from polar to rectangular form.
//    R is a matrix containing the magnitudes
//    Theta is a matrix containing the phase angles (in degrees).
function [A] = p2z(R,Theta)
 A = R*exp(%i*%pi*Theta/180);
endfunction

//Initialisation
Y1 = p2z(0.2,0)                    //admitance
Y2 = p2z(0.02,-90)                   //admitance
Y3 = p2z(0.25,90)                   //admitance


//Calculation
y11 = Y1+Y2               //admitance
y12 = -Y2               //admitance
y22 = Y2+Y3               //admitance

//Result
printf("Y11 = %.1f mS - j %.2f mS \n",real(y11),-
```

```
imag(y11))
printf("Y12 = j %.2f mS \n",imag(y12))
printf("Y22 = j %.2f mS \n",imag(y22))
```

Scilab code Exa 26.11 Example 11

```
//Chapter 26, Example 26.11, page 1177
clc
//Initialisation
hf=50                   //short circuit current gain
ho=25*10**-6            //Output Admittance with input open circuited
zl=2*10**3              //load impedance
hi=1*10**3               //input impedance with output shorted
hr=4*10**-4               //reverse voltage amplification with input open circuited

//Calculation
Ai=hf/(1+(ho*zl))                           //current gain
Av=-hf*zl/((hi*(1+(ho*zl)))-(hr*hf*zl))     //voltage gain

//Result
printf("Current gain, Ai = %.2f \n",Ai)
printf("Voltage gain, Av = %ds \n",Av)
```

Scilab code Exa 26.12 Example 12

```
//Chapter 12, Example 12.12
clc
//Variable Declaration
```

```
r1 = 3                    //resistance1
l = 6*10^-3               //inductance in mH
e = 15                    //voltage in volts
r2 = 2                    //resistance2
//Calculation
im = e/(r1+r2)            //inductance current
ws = (l*im*im)/2          //energy stored

//Results
printf("Ws = %d mJ",ws*10^3)
```