Scilab Textbook Companion for Elements Of Power System Analysis by W. D. Stevenson[1]

Created by
S.Puneeth Kumar
B.E
Electrical Engineering
Sri Jayachamarajendra College Of Engineering
College Teacher
Prof.R.S.Ananda Murthy
Cross-Checked by
TechPassion

July 13, 2017

[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: Elements Of Power System Analysis

Author: W. D. Stevenson

Publisher: McGraw - Hill Book Comp.

Edition: 4

Year: 1982

ISBN: 0-07-066584-2

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes

List of Figures

Chapter 2

Basic Concepts

Scilab code Exa 1 problem on power

problem on power

```
//Chapter 2
//Example 2.1
//Page 21
clear;clc;
E1=100+0*%i;
E2=86.6+50*%i;
Z=5*%i;
X=5;
printf("\n\n Value of voltage source one designated as a machine 1 =  %.4f V \n\n",E1)
printf("\n\n Value of voltage source two designated as a machine 2 =  %.4f V \n\n",E2)
printf("\n\n Impedance connected  = %.4f ohms \n\n",abs(Z))


//Calculation Of Current
I=(E1-E2)/Z;
```

```
printf("\n\n Current through the impedance = %.4f A \n\n",I)


//Calculation Of Power
M1=E1*conj(I);
disp(M1,'Machine One Power = ')
M2=E2*conj(I);
disp(M2,'Machine Two Power = ')


//Calculation Of Reactive Power Required By Inductive Reactance
RP=(abs(I))^2*X;
printf("\n\n Reactive power required by inductive reactance i.e, impedance = %.4f VAR \n\n",RP)
printf("\n\n Machine 1 consumes energy at the rate of %.4f W \n\n",abs(real(M1)))
printf("\n\n Machine 2 generates energy at the rate of %.4f W \n\n",abs(real(M2)))
printf("\n\n Machine 1 supplies reactive power at the rate of  %.4f VAR \n\n",imag(M1))
printf("\n\n Machine 2 supplies reactive power at the rate of %.4f VAR \n\n",abs(imag(M2)))
printf("\n\n Reactive power required by inductive reactance i.e, impedance = Sum of reactive power supplied by machine 1 + reactive power supplied by machine 2  = %.4f VAR \n\n",RP)
disp('Real Power consumed by impedance is Zero')
disp('The real power generated by machine two is transferred to machine one')
```

Scilab code Exa 2 Phase

Phase

```
//Chapter 2
//Example 2.2
//Page 26
clear;clc;
//Given line-line voltages
Vab = 173.2 * (cos (0)+ sin(0) * %i);
Vbc = 173.2 * (cos (240*%pi/180)+ sin(240*%pi/180) * %i);
Vca = 173.2 * (cos (120*%pi/180)+ sin(120*%pi/180) * %i);
disp('The given line-line voltages are')
//disp(abs(Vab),'Magnitude  of Vab = ')
//disp(atan(imag(Vab),real(Vab))*180/%pi,'Phase Angle of Vab = ')
//'/_' this symbol has been used to show angle
printf("\n\n Vab = %.4f  /_ %.4f V \n\n",abs(Vab),((atan(imag(Vab),real(Vab)))*180/%pi))
printf("\n\n Vbc = %.4f  /_ %.4f V \n\n",abs(Vbc),((atan(imag(Vbc),real(Vbc)))*180/%pi))
printf("\n\n Vca = %.4f  /_ %.4f V \n\n",abs(Vca),((atan(imag(Vca),real(Vca)))*180/%pi))
//Calculation of line-neutral voltage
Van = (Vab / sqrt(3)/(0.866+0.5*%i));
Vbn = (Vbc / sqrt(3)/(0.866+0.5*%i));
Vcn = (Vca / sqrt(3)/(0.866+0.5*%i));
disp('The line-neutral voltages are')
printf("\n\n Van = %.4f  /_ %.4f V \n\n",abs(Van),((atan(imag(Van),real(Van)))*180/%pi))
printf("\n\n Vbn = %.4f  /_ %.4f V \n\n",abs(Vbn),((atan(imag(Vbn),real(Vbn)))*180/%pi))
printf("\n\n Vcn = %.4f  /_ %.4f V \n\n",abs(Vcn),((atan(imag(Vcn),real(Vcn)))*180/%pi))
ZL = 10 * (cos(20*%pi/180) + sin(20*%pi/180) * %i);
printf("\n\n Load Impedance ZL = %.4f  /_ %.4f ohms \n\n",abs(ZL),((atan(imag(ZL),real(ZL)))*180/%pi)
```

```
)
//Calculation of line-neutral current
Ian = Van / ZL;
Ibn = Vbn / ZL;
Icn = Vcn / ZL;
disp('The resulting current in each phase')
printf("\n\n Ian = %.4f  /_ %.4f A \n\n",abs(Ian),((
    atan(imag(Ian),real(Ian)))*180/%pi))
printf("\n\n Ibn = %.4f  /_ %.4f A \n\n",abs(Ibn),((
    atan(imag(Ibn),real(Ibn)))*180/%pi))
printf("\n\n Icn = %.4f  /_ %.4f A \n\n",abs(Icn),((
    atan(imag(Icn),real(Icn)))*180/%pi))
```

Scilab code Exa 3 Substation Bus

`Substation Bus`

```
//Chapter 2
//Example 2.3
//Page 28
clear;clc;
//Given values
Vll = 4.4e3;
Vln = Vll / sqrt(3);
Zline = 1.4 * (cos(75 * %pi / 180) + %i * sin (75 *
    %pi / 180));
Van = Vln * (cos(0) + %i * sin(0));
Zload = 20 * (cos(30 * %pi / 180) + %i * sin(30 *
    %pi / 180));
printf("\n\n Given line-line voltage = %.4f V \n\n",
    Vll)
printf("\n\n Line-neutral voltage = %.4f V \n\n",Vln
    )
```

```
//disp(abs(Zload),'Magnitude  of load impedance Z =
    ')
//disp(atan(imag(Zload),real(Zload))*180/%pi,'Phase
    Angle of load impedance Z = ')
//'/_' this symbol has been used to specify angle
printf("\n\n Impedance of the load = %.4f  /_ %.4f
    ohms \n\n",abs(Zload),((atan(imag(Zload),real(
    Zload)))*180/%pi))
printf("\n\n Impedance of the line = %.4f  /_ %.4f
    ohms \n\n",abs(Zline),((atan(imag(Zline),real(
    Zline)))*180/%pi))
//Calculation of phase current
Ian = Van / Zload;
//'/_' this symbol has been used to specify angle
printf("\n\n Van = %.4f  /_ %.4f V \n\n",abs(Van),((
    atan(imag(Van),real(Van)))*180/%pi))
printf("\n\n Ian = %.4f  /_ %.4f V \n\n",abs(Ian),((
    atan(imag(Ian),real(Ian)))*180/%pi))
//Calculation of line-neutral voltage at the
    substation
Vltn = Van + Ian * Zline;
//Magnitude of the voltage at the substation bus
magVl = sqrt(3) * abs(Vltn);
printf("\n\n The line to neutral voltage at the
    substation = %.4f /_ %.4f V \n\n",abs(Vltn),((
    atan(imag(Vltn),real(Vltn)))*180/%pi))
printf("\n\n The magnitude of the voltage at the
    substation bus = %.4f V",magVl)
```

Scilab code Exa 4 example 3 in per unit

```
example 3 in per unit
```

```
//Chapter 2
//Example 2.4
//Page 32
//Given values
clear;clc;
Vll = 4.4e3;
Zline = 1.4 * (cos(75 * %pi / 180) + %i * sin (75 * %pi / 180));
Zload = 20 * (cos(30 * %pi / 180) + %i * sin(30 * %pi / 180));
Vbase = Vll;
Ibase = 127;
Zbase = (Vbase / sqrt(3)) / Ibase ;
printf("\n\n Given line-line voltage = %.4f V \n\n",Vll)
printf("\n\n Impedance of the load = %.4f  /_ %.4f ohms \n\n",abs(Zload),((atan(imag(Zload),real(Zload)))*180/%pi))
printf("\n\n Impedance of the line = %.4f  /_ %.4f ohms \n\n",abs(Zline),((atan(imag(Zline),real(Zline)))*180/%pi))
printf("\n\n Base Voltage = %.4f V \n\n ",Vbase)
printf("\n\n Base Current = %.4f V \n\n",Ibase)
printf("\n\n Base Impedance = %.4f V \n\n",Zbase)
Van = (Vll / sqrt(3)) * (cos(0) + %i * sin(0));
Ian = Van / Zload;
printf("\n\n Van = %.4f  /_ %.4f V \n\n",abs(Van),((atan(imag(Van),real(Van)))*180/%pi))
printf("\n\n Ian = %.4f  /_ %.4f V \n\n",abs(Ian),((atan(imag(Ian),real(Ian)))*180/%pi))
//Calculation of per-unit quantities
V_pu = Vll / Vbase;
I_pu = Ian / Ibase;
Zline_pu = Zline / Zbase;
Zload_pu = Zload / Zbase;
Van_pu = V_pu + I_pu * Zline_pu;
disp('Per-unit Quantities')
printf("\n\n Per Unit line-line voltage = %.4f  /_ %
```

```
.4f per unit \n\n",abs(V_pu),((atan(imag(V_pu),
real(V_pu)))*180/%pi))
printf("\n\n Per Unit line-neutral current = %.4f /
_ %.4f per unit \n\n",abs(I_pu),((atan(imag(I_pu)
,real(I_pu)))*180/%pi))
printf("\n\n Per Unit line-neutral voltage = %.4f /
_ %.4f per unit \n\n",abs(Van_pu),((atan(imag(
Van_pu),real(Van_pu)))*180/%pi))
printf("\n\n Per Unit line impedance = %.4f /_ %.4
f per unit \n\n",abs(Zline_pu),((atan(imag(
Zline_pu),real(Zline_pu)))*180/%pi))
printf("\n\n Per Unit load impedance = %.4f /_ %.4f
 per unit \n\n",abs(Zload_pu),((atan(imag(
Zload_pu),real(Zload_pu)))*180/%pi))
//Calculation of line-neutral and line-line voltage
Vln = abs(Van_pu) * Vll / sqrt(3);
VLL = abs(Van_pu) * Vll;
printf("\n\n The line to neutral voltage at the
substation , VLN = %.4f /_ %.4f V \n\n",abs(Vln)
,((atan(imag(Vln),real(Vln)))*180/%pi))
printf("\n\n The magnitude of the voltage at the
substation bus , VLL= %.4f V",VLL)
```

Scilab code Exa 5 Per Unit

Per Unit

```
//Chapter 2
//Example 2.5
//Page 34
clear;clc;
pu_Z_given = 0.25;
base_kV_given = 18;
```

```
base_kV_new = 20;
base_kVA_new = 100;
base_kVA_given = 500;
X11=(pu_Z_given * (base_kV_given/base_kV_new)^2 * (
    base_kVA_new/base_kVA_given));
disp('Per-Unit Znew = per-unit Zgiven * (base
    kVgiven/base kVnew)^2 * (base kVAnew/base
    kVAgiven)')
disp(' = 0.25 * (18/20)^2 * (100/500)')
printf("\n Per-Unit Znew = %f per unit",X11)
```

Scilab code Exa 2.1 ProbOnPwr

ProbOnPwr

```
//Chapter 2
//Example 2.1
//ProbOnPwr
//Page 21
clear;clc;
E1=100+0*%i;
E2=86.6+50*%i;
Z=5*%i;
X=5;
printf("\n\n Value of voltage source one designated
    as a machine 1 =  %.4f V \n\n",E1)
printf("\n\n Value of voltage source two designated
    as a machine 2 =  %.4f V \n\n",E2)
printf("\n\n Impedance connected  = %.4f ohms \n\n",
    abs(Z))


//Calculation Of Current
```

```
I=(E1-E2)/Z;
printf("\n\n Current through the impedance = %.4f A
   \n\n",I)


//Calculation Of Power
M1=E1*conj(I);
disp(M1,'Machine One Power = ')
M2=E2*conj(I);
disp(M2,'Machine Two Power = ')


//Calculation Of Reactive Power Required By
   Inductive Reactance
RP=(abs(I))^2*X;
printf("\n\n Reactive power required by inductive
   reactance i.e, impedance = %.4f VAR \n\n",RP)
printf("\n\n Machine 1 consumes energy at the rate
   of %.4f W \n\n",abs(real(M1)))
printf("\n\n Machine 2 generates energy at the rate
   of %.4f W \n\n",abs(real(M2)))
printf("\n\n Machine 1 supplies reactive power at
   the rate of  %.4f VAR \n\n",imag(M1))
printf("\n\n Machine 2 supplies reactive power at
   the rate of %.4f VAR \n\n",abs(imag(M2)))
printf("\n\n Reactive power required by inductive
   reactance i.e, impedance = Sum of reactive power
   supplied by machine 1 + reactive power supplied
   by machine 2  = %.4f VAR \n\n",RP)
disp('Real Power consumed by impedance is Zero')
disp('The real power generated by machine two is
   transferred to machine one')
```

Scilab code Exa 2.2 PhaseProb

PhaseProb

```
//Chapter 2
//Example 2.2
//PhaseProb
//Page 26
clear;clc;


//Given line-line voltages
Vab = 173.2 * (cos (0)+ sin(0) * %i);
Vbc = 173.2 * (cos (240*%pi/180)+ sin(240*%pi/180) *
    %i);
Vca = 173.2 * (cos (120*%pi/180)+ sin(120*%pi/180) *
    %i);
disp('The given line-line voltages are')


//'/_' this symbol has been used to show angle
printf("\n\n Vab = %.4f  /_ %.4f V \n\n",abs(Vab),((
   atan(imag(Vab),real(Vab)))*180/%pi))
printf("\n\n Vbc = %.4f  /_ %.4f V \n\n",abs(Vbc),((
   atan(imag(Vbc),real(Vbc)))*180/%pi))
printf("\n\n Vca = %.4f  /_ %.4f V \n\n",abs(Vca),((
   atan(imag(Vca),real(Vca)))*180/%pi))


//Calculation of line-neutral voltage
Van = (Vab / sqrt(3)/(0.866+0.5*%i));
Vbn = (Vbc / sqrt(3)/(0.866+0.5*%i));
Vcn = (Vca / sqrt(3)/(0.866+0.5*%i));
disp('The line-neutral voltages are')
printf("\n\n Van = %.4f  /_ %.4f V \n\n",abs(Van),((
   atan(imag(Van),real(Van)))*180/%pi))
printf("\n\n Vbn = %.4f  /_ %.4f V \n\n",abs(Vbn),((
   atan(imag(Vbn),real(Vbn)))*180/%pi))
```

```
printf("\n\n Vcn = %.4f  /_ %.4f V \n\n",abs(Vcn),((
    atan(imag(Vcn),real(Vcn)))*180/%pi))
ZL = 10 * (cos(20*%pi/180) + sin(20*%pi/180) * %i);
printf("\n\n Load Impedance ZL = %.4f  /_ %.4f ohms
    \n\n",abs(ZL),((atan(imag(ZL),real(ZL)))*180/%pi)
    )


//Calculation of line-neutral current
Ian = Van / ZL;
Ibn = Vbn / ZL;
Icn = Vcn / ZL;
disp('The resulting current in each phase')
printf("\n\n Ian = %.4f  /_ %.4f A \n\n",abs(Ian),((
    atan(imag(Ian),real(Ian)))*180/%pi))
printf("\n\n Ibn = %.4f  /_ %.4f A \n\n",abs(Ibn),((
    atan(imag(Ibn),real(Ibn)))*180/%pi))
printf("\n\n Icn = %.4f  /_ %.4f A \n\n",abs(Icn),((
    atan(imag(Icn),real(Icn)))*180/%pi))
```

Scilab code Exa 2.3 SubstationBus

`SubstationBus`

```
//Chapter 2
//Example 2.3
//SubstationBus
//Page 28
clear;clc;


//Given values
Vll = 4.4e3;
```

```
Vln = Vll / sqrt(3);
Zline = 1.4 * (cos(75 * %pi / 180) + %i * sin (75 * %pi / 180));
Van = Vln * (cos(0) + %i * sin(0));
Zload = 20 * (cos(30 * %pi / 180) + %i * sin(30 * %pi / 180));
printf("\n\n Given line-line voltage = %.4f V \n\n",Vll)
printf("\n\n Line-neutral voltage = %.4f V \n\n",Vln)


//'/_' this symbol has been used to specify angle
printf("\n\n Impedance of the load = %.4f  /_ %.4f ohms \n\n",abs(Zload),((atan(imag(Zload),real(Zload)))*180/%pi))
printf("\n\n Impedance of the line = %.4f  /_ %.4f ohms \n\n",abs(Zline),((atan(imag(Zline),real(Zline)))*180/%pi))
//Calculation of phase current
Ian = Van / Zload;


printf("\n\n Van = %.4f  /_ %.4f V \n\n",abs(Van),((atan(imag(Van),real(Van)))*180/%pi))
printf("\n\n Ian = %.4f  /_ %.4f V \n\n",abs(Ian),((atan(imag(Ian),real(Ian)))*180/%pi))
//Calculation of line-neutral voltage at the substation
Vltn = Van + Ian * Zline;


//Magnitude of the voltage at the substation bus
magVl = sqrt(3) * abs(Vltn);
printf("\n\n The line to neutral voltage at the substation = %.4f /_ %.4f V \n\n",abs(Vltn),((atan(imag(Vltn),real(Vltn)))*180/%pi))
printf("\n\n The magnitude of the voltage at the
```

```
substation bus = %.4f V",magVl)
```

Scilab code Exa 2.4 Ex3inPerUnit

Ex3inPerUnit

```
//Chapter 2
//Example 2.4
//Ex3inPerUnit
//Page 32
clear;clc;


//Given values
Vll = 4.4e3;
Zline = 1.4 * (cos(75 * %pi / 180) + %i * sin (75 *
    %pi / 180));
Zload = 20 * (cos(30 * %pi / 180) + %i * sin(30 *
    %pi / 180));
Vbase = Vll;
Ibase = 127;
Zbase = (Vbase / sqrt(3)) / Ibase ;

//Displaying the given values and the base values
printf("\n\n Given line-line voltage = %.4f V \n\n",
    Vll)
printf("\n\n Impedance of the load = %.4f  /_ %.4f
    ohms \n\n",abs(Zload),((atan(imag(Zload),real(
    Zload)))*180/%pi))
printf("\n\n Impedance of the line = %.4f  /_ %.4f
    ohms \n\n",abs(Zline),((atan(imag(Zline),real(
    Zline)))*180/%pi))
printf("\n\n Base Voltage = %.4f V \n\n ",Vbase)
```

```
printf("\n\n Base Current = %.4f V \n\n",Ibase)
printf("\n\n Base Impedance = %.4f V \n\n",Zbase)
Van = (Vll / sqrt(3)) * (cos(0) + %i * sin(0));
Ian = Van / Zload;
printf("\n\n Van = %.4f  /_ %.4f V \n\n",abs(Van),((atan(imag(Van),real(Van)))*180/%pi))
printf("\n\n Ian = %.4f  /_ %.4f V \n\n",abs(Ian),((atan(imag(Ian),real(Ian)))*180/%pi))


//Calculation of per-unit quantities
V_pu = Vll / Vbase;
I_pu = Ian / Ibase;
Zline_pu = Zline / Zbase;
Zload_pu = Zload / Zbase;
Van_pu = V_pu + I_pu * Zline_pu;
disp('Per-unit Quantities')
printf("\n\n Per Unit line-line voltage = %.4f  /_ %.4f per unit \n\n",abs(V_pu),((atan(imag(V_pu),real(V_pu)))*180/%pi))
printf("\n\n Per Unit line-neutral current = %.4f  /_ %.4f per unit \n\n",abs(I_pu),((atan(imag(I_pu),real(I_pu)))*180/%pi))
printf("\n\n Per Unit line-neutral voltage = %.4f  /_ %.4f per unit \n\n",abs(Van_pu),((atan(imag(Van_pu),real(Van_pu)))*180/%pi))
printf("\n\n Per Unit line impedance  = %.4f  /_ %.4f per unit \n\n",abs(Zline_pu),((atan(imag(Zline_pu),real(Zline_pu)))*180/%pi))
printf("\n\n Per Unit load impedance = %.4f  /_ %.4f per unit \n\n",abs(Zload_pu),((atan(imag(Zload_pu),real(Zload_pu)))*180/%pi))


//Calculation of line-neutral and line-line voltage
Vln = abs(Van_pu) * Vll / sqrt(3);
VLL = abs(Van_pu) * Vll;
printf("\n\n The line to neutral voltage at the
```

```
   substation , VLN = %.4f /_ %.4f V \n\n",abs(Vln)
   ,((atan(imag(Vln),real(Vln))*180/%pi))
printf("\n\n The magnitude of the voltage at the
   substation bus , VLL= %.4f V",VLL)
```

Scilab code Exa 2.5 PerUnit

PerUnit

```
//Chapter 2
//Example 2.5
//PerUnit
//Page 34
clear;clc;
pu_Z_given = 0.25;
base_kV_given = 18;
base_kV_new = 20;
base_kVA_new = 100;
base_kVA_given = 500;
X11=(pu_Z_given * (base_kV_given/base_kV_new)^2 * (
   base_kVA_new/base_kVA_given));
disp('Per-Unit Znew = per-unit Zgiven * (base
   kVgiven/base kVnew)^2 * (base kVAnew/base
   kVAgiven)')
disp(' = 0.25 * (18/20)^2 * (100/500)')
printf("\n Per-Unit Znew = %f per unit",X11)
```

Chapter 3

Series Impedance Of Transmission Lines

Scilab code Exa 3.1 Resistance

Resistance

```
//Chapter 3
//Example 3.1
//Resistance
//Page 42
clear;clc;

//Given Values
R_dc = 0.01558; //in ohm per 1000 ft at 20 degree
   Celsius
R_ac = 0.0956; //in ohm per mi at 50 degree Celsius
A = 1113000; //in cmil
l = 1000;//in ft
p = 17; // $p$(rho) = 2.83e-8 ohm.m = 17 ohm.cmil
   per ft

//Verification of dc resistance
```

```
R_0_1 = p*l*1.02/A; //1.02 is to account for 2\%
    increase in spiraling
printf("\n\n The dc resistance at 20 degree Celsius
    = %f ohm per 1000 ft\n\n",R_0_1)

T = 228;//in degree Celsius
t1  = 50;//in degree Celsius
t2 = 20;//in degree Celsius

//to obtain dc resistance at 50 degree celsius
R0 = R_0_1 * (T + t1)/(T + t2);

printf("\n\n The dc resistance at 50 degree Celsius
    = %f ohm per 1000 ft \n\n",R0)

R = R_ac / 5.280e3; //to convert ohm per mi to ohm
    per ft,ac resistance
R_0 = R0 / 1000; //to convert ohm per 1000 ft to ohm
     per ft,dc resistance

// to calculate ratio of ac to dc resistance
printf("\n\n Ratio of ac to dc resistance = %.3f  \n
    \n",R / R_0)
printf("\n\n Skin effect causes a %.1f increase in
    resistance.\n\n",((R/R_0)-1)*100)
```

Scilab code Exa 3.2 singlephaseinductance

singlephaseinductance

```
//Chapter 3
//Example 3.2
//singlephaseinductance
```

```
//Page 55
clear;clc;

//Given Values
r_x = 0.25e-2;//radius of circuit in m,composed of 3 wires a,b,c
r_y = 0.50e-2;//radius of return circuit in m, composed of 2 wires d,e
d_c = 9;//distance between the two circuits
d_w = 6;//distance between wires of same circuit

//To find GMD between sides X and Y
D_ad = d_c;D_be = D_ad;
D_ae = sqrt(d_w^2+d_c^2);
D_bd = D_ae;D_ce = D_ae;
D_cd = sqrt(d_c^2+(2*d_w)^2);
//GMD is given by
D_m = (D_ad * D_ae * D_bd * D_be * D_cd * D_ce)^(1/6);
printf("\n\n The GMD between the sides X and Y = %.3f m \n\n",D_m)

//To find GMR for the side X
D_aa = r_x * 0.7788;//multiplication by 0.7788 to adjust the radiuss
                    //in order to account for internal flux
D_ab = d_w;D_ac = 2 * d_w;D_ba = d_w;D_bb = D_aa;D_bc = D_ab;
D_ca = D_ac;D_cb = D_ab;
D_cc = D_aa;

//GMR for side X
D_s_x = (D_aa * D_ab * D_ac * D_ba * D_bb * D_bc * D_ca * D_cb * D_cc)^(1/9);
printf("\n\n The GMR for side X = %.3f m \n\n",D_s_x)

```

```
//To find GMR for the side Y
D_dd = r_y * 0.7788;
D_de = d_w;D_ee = D_dd;D_ed = D_de;

//GMR for side Y
D_s_y = (D_dd * D_de * D_ee * D_ed)^(1/4);
printf("\n\n The GMR for side Y = %.3f m \n\n",D_s_y)

//Inductance
L_x = 2e-7 * log(D_m / D_s_x);
L_y = 2e-7 * log(D_m / D_s_y);
L = L_x + L_y;
printf("\n\n Inductance of side X = %.3fe-7 H/m \n\n",L_x*10^7)
printf("\n\n Inductance of side Y = %.3fe-7 H/m \n\n",L_y*10^7)
printf("\n\n Inductance of Complete line = %.3fe-7 H/m\n\n",L*10^7)
printf("\t\t\t \t= %.2f mH/mi \n\n",L * 1609e3)
```

Scilab code Exa 3.3 Partridge

Partridge

```
//Chapter 3
//Example 3.3
//Partridge
//Page 57
clear;clc;

//Given Values
f = 60 //in Hz
```

```
D_m = 20 //in ft

//Inductive Reactance with D_s known
D_s = 0.0217 //in ft from Table A(appendix).1
X_L = 2.022e-3 * f * log(D_m/D_s)
disp('With GMR known')
printf("\n\n Inductive Reactance for one conductor,
    XL = %.3f ohm/mi \n\n",X_L)

//Inductive Reactance with D_s not known
disp('With GMR not known')
X_a = 0.465 //inductive reactance at 1 ft spacingin
    ohm per mi from Table A.1
X_d = 0.3635 //inductive reactance spacing factor in
     ohm per mi from Table A.1
printf("\n\n Inductive reactance of one conductor =
    %.4f ohm/mi \n\n",X_a + X_d)
disp('Since conductors composing the two lines are
    identical')
XL = 2 * (X_a + X_d)
printf("\n\n Inductive reactance,XL = %.3f ohm/mi \n
    \n",XL)
```

Scilab code Exa 3.4 Inductanceof3phaseline

`Inductanceof3phaseline`

```
//Chapter 3
//Example 3.4
//Inductanceof3phaseline
//Page 60
clear;clc;

```

```
//Given Values
D_12 = 20;D_23 = D_12;D_31 = 38;//in ft
f=60;//frequency in Hz

//From Table A.1
D_s = 0.0373;//in ft
X_a = 0.399;//inductive reactance at 1 ft spacing in ohm/mi
X_d = 0.389;//inductive reactance spacing factor in ohm/mi for 24.8ft

//Calculations
D_eq = (D_12 * D_23 * D_31)^(1/3);
L = 2e-7 * log(D_eq/D_s)*10^7;
X_L = 2*%pi*f*L*10^-7*1609;//multiplication by 1609 to convert to ohm/mi

//Displaying
printf("\n\n Inductance = %.4fe-7 H/m \n\n",L)
printf("\n\n Inductive reactance = %.4f ohm/mi/phase \n\n",X_L)

//inductance from X_a and X_d
X_L1 = X_a + X_d;
printf("\n\n Inductive reactance from Xa and Xd = %.4f ohm/mi/phase \n\n",X_L1)
```

Scilab code Exa 3.5 BundledConductors

BundledConductors

```
//Chapter 3
//Example 3.5
```

```
//BundledConductors
//Page 62
clear;clc;

//Given Values
d = 0.45;//in m
l = 160 ;//in km
d_12 = 8;d_23 = 8;d_31 = 16;//in m
f = 60;//in Hz
P = 100;//in MVA
V = 345;//in kV

//From Table A1
D_s = 0.0466;//in ft

//Calculations
D_b_s = sqrt(D_s * 0.3048 * d);//multiplicqation by
    0.3048 is to convert D_s
                              //from ft to m
D_eq = (d_12 * d_23 * d_31)^(1/3);
X_L = 2 * %pi * f * 2e-7 * 10^3 * log(D_eq / D_b_s);
    //10^3 to get ohm/km

//To find per unit series reactance
Z = V^2 / P;//Base Impedance
X = X_L * l / Z;//per unit series reactance

printf("\n\n Inductive reactance = %.3f ohm/km/phase
    \n\n",X_L)
printf("\n\n Base Impedance = %.0f ohm \n\n",Z)
printf("\n\n Per unit series resistance of the line
    = %.3f per unit \n\n",X)
```

Scilab code Exa 3.6 ParallelCircuit

`ParallelCircuit`

```
//Chapter 3
//Example 3.6
//ParallelCircuit
//Page 63
clear;clc;
//Example 3.6

//Given
f = 60;//in Hz

//From Table A.1
D_s = 0.0229 //in ft

//Distances from given figure 3.15
d_a_c = 18;d_c_a = d_a_c;d_b_b = 21;
d = 10;//distance between conductors
d_a_b = sqrt(d^2 + (d_b_b - 19.5)^2);
d_a_b1 = sqrt(d^2 + (d_b_b - 1.5)^2);
d_aa1_actual = sqrt((d * 2)^2 + d_a_c^2);
d_bb1_actual = d_b_b;
d_cc1_actual = d_aa1_actual;
d_aa1_pos = sqrt(d_aa1_actual * D_s);
d_bb1_pos = sqrt(d_bb1_actual * D_s);
d_cc1_pos = sqrt(d_cc1_actual * D_s)

//GMD's between phases
D_p_ab = ((d_a_b * d_a_b1)^(2*1/4));//in ft
D_p_bc = D_p_ab;
D_p_ca = (((d*2) * d_c_a)^(2*1/4));//in ft
D_eq = (D_p_ab * D_p_bc * D_p_ca)^(1/3);//in ft
printf("\n\n Equivalent GMD = %.1f ft \n\n",D_eq)

//GMR
D_p_s = (d_aa1_pos * d_bb1_pos * d_cc1_pos)^(1/3);
```

```
printf("\n\n GMR = %.3f ft \n\n",D_p_s)

//Inductance
L = 2e-7 * log(D_eq / D_p_s);
X_L = 2 * %pi * f * L * 1609;//multiplication by
    1609 is to convert to ohm/mi

printf("\n\n The Inductive reactance = %.3f ohm/mi/
    phase \n\n",X_L)
```

Chapter 4

Capacitance Of Transmission Lines

Scilab code Exa 4.1 capacitivesusceptance

capacitivesusceptance

```
//Chapter 4
//Example 4.1
//capacitivesusceptance
//Page 75
clear;clc;

//Given Values
D = 20; //in ft
f = 60; //in Hz

//From Table A.1 and A.3
d = 0.642 //in inches
X_a = 0.1074e6; //in ohm-mi
X_d = 0.0889e6; //in ohm-mi

//finding radius
r = d /(2 * 12); //divided by 12 convert in to ft
```

```

//calculations using D and r
disp('Calculations using conductor spacing and radius')
X_c = 1.779 * log(D / r) / f;
B_c = 1 / X_c;
printf("\n\n Capactive reatance = %.4fe6 ohm mi to neutral \n\n",X_c)
printf("\n\n Capactive susceptance = %.4fe-6 mho/mi to neutral \n\n",B_c)

//calculations using capacitive reactance at 1-ft spacing and spacing factor
disp('Calculations using capacitive reactance at 1-ft spacing and spacing factor')
X_c1 = X_a + X_d;
printf("\n\n Capactive reatance = %.4fe6 ohm mi per conductor \n\n",X_c1 / 10^6)
X_c11 = 2 * X_c1;
B_c1 = 1 / X_c11;
printf("\n\n Line-to-line capactive reatance = %.4fe6 ohm mi \n\n",X_c11 / 10^6)
printf("\n\n Line-to-line capactive susceptance = %.4fe-6 mho mi \n\n",B_c1 * 10^6 )
```

Scilab code Exa 4.2 ChargingMVA

ChargingMVA

```
//Chapter 4
//Example 4.2
//ChargingMVA
//Page 80
```

```
clear;clc;

//Given values
D_12 = 20;//in ft
D_23 = D_12;
D_31 = 38;//in ft
f = 60;//in Hz
V = 220e3;//in volts
l = 175;//in mi
k = 8.85e-12;//permittivity in F/m
//From tables A.1 and A.3
d = 1.108;//in inches
X_a1 = 0.0912e6;//in ohm mi
X_d1 = 0.0952e6;//in ohm mi

//Calculations
r = d / ( 2 * 12);//division by 12 to convert in to
    ft
D_eq = (D_12 * D_23 * D_31)^(1/3);
C_n = (2 * %pi * k)/log(D_eq/r);
X_c = 1 / (2 * %pi * f * C_n * 1609);//division by
    1609 to convert to ohm mi

printf("\n\n Capacitance = %.4fe-12 F/m \n\n",C_n *
    1e12)
printf("\n\n Capacitive reactance = %.4fe6 ohm mi \n
    \n",X_c / 1e6)

//Calculations From tables
X_c1 = X_a1 + X_d1;

disp('Using capacitive reactance at 1-ft spacing and
     spacing factor')
printf("\n\n Capacitive reactance = %.4fe6 ohm mi \n
    \n",X_c1 / 1e6)

X_c_l = X_c1 / l;//Capacitive reactance for 175mi
I_chg = 2 * %pi * f * V * C_n * 1609 / sqrt(3);
```

```
I_chg_l = I_chg * l;
Q = sqrt(3) * V * I_chg_l;

disp('For a lenght of 175mi')
printf("\n\n Capacitive reactance = %.4f ohm to
    neutral \n\n",X_c_l)
printf("\n\n Charging current per mile = %.3f A/mi \
    n\n",I_chg)
disp('For a lenght of 175mi')
printf("\n\n Charging current = %.0f A \n\n",I_chg_l
    )
printf("\n\n Total charging megavolt-amperes = %.1f
    Mvar \n\n",Q / 1e6)
```

Scilab code Exa 4.3 chap3ex5

chap3ex5

```
//Chapter 4
//Example 4.3
//chap3ex5
//Page 85
clear;clc;

//Given Values
d = 0.45;//in m
k = 8.85e-12;//in F/m
D_ab = 8;//in m
D_bc = D_ab;
D_ca = 16;//in m
f = 60;//in Hz

//From tables
```

```
D = 1.382;//in inches

//Calculations
r = D * 0.3048 / (2 * 12)//divison by 12 to convert
    in to ft
                          //multiplication by 0.3048
                             to convert ft to m
D_b_sC = sqrt( r * d);
D_eq = (D_ab * D_bc * D_ca)^(1/3);
C_m = 2 * %pi * k / log(D_eq / D_b_sC);
X_c = 1e-3 / (2 * %pi * f *C_m);//1e-3 to convert m
    to km
printf("\n\n Capacitance = %.3fe-12 F/m \n\n",C_m *
    1e12)
printf("\n\n Capacitive reactance = %.4fe6 ohm km
    per phase to neutral",X_c/1e6)
```

Scilab code Exa 4.4 chap3ex6

chap3ex6

```
//Chapter 4
//Example 4.4
//chap3ex6
//Page 85
clear;clc;

//Given
f = 60;//in Hz
k = 8.85e-12;//in F/m

//From example 3.6
D_eq = 16.1;//in ft
```

```
D_a_a1 = 26.9;D_b_b1 = 21;D_c_c1 = D_a_a1 ;//in ft

//From Table A.1
d = 0.680;//in inches

//calculations
r = d /(2*12);
D_p_sC = (sqrt(D_a_a1 * r) * sqrt(D_b_b1 * r) * sqrt
    (D_c_c1 * r))^(1/3);
C_n = 2 * %pi * k / log(D_eq / D_p_sC);
B_c = 2 * %pi * f * C_n * 1609;//1609 to convert
    from m to mi
printf("\n\n Capacitance = %.3fe-12 F/m \n\n",C_n *
    1e12)
printf("\n\n Capacitive susceptance = %.2fe-6 mho
    per mi per phase to neutral",B_c * 1e6)
```

Chapter 5

Current And Voltage Relations On A Transmission Line

Scilab code Exa 5.1 Velocity

Velocity

```
//Chapter 5
//Example 5.1
//Page 101
//Velocity
clear;clc;

//Given
D_12 = 23.8 ; D_23 = 23.8 ; D_31 = 47.6 ;//in ft
l = 230 ; //in mi
f = 60 ; //in Hz
P = 125e6 ; //in W
V = 215e3 ; //in V

D_eq = (D_12 * D_23 * D_31)^(1/3);

//From Table A.1 and A.2 for 30ft Rook
//z = R + i(Xa + Xd)
```

```
z = 0.1603 + %i * (0.415+0.4127);

//From Table A.1 and A.3 for 30ft Rook
y = %i * [1e-6 / ( 0.0950 + 0.1008)]

//Calculations
yl = sqrt(y*z)*l;
Z_c = sqrt(z/y);
V_r = V / sqrt(3);
I_r = P / (sqrt(3)*V);

cosh_yl = cosh(real(yl)) * cos(imag(yl)) + %i * sinh(real(yl)) * sin(imag(yl));
sinh_yl = sinh(real(yl)) * cos(imag(yl)) + %i * cosh(real(yl)) * sin(imag(yl));

V_s = V_r * cosh_yl + I_r * Z_c * sinh_yl;
I_s = I_r * cosh_yl + V_r * sinh_yl / Z_c;
printf("\n\n Sending end voltage = %.0f /_%.2f V \n",abs(V_s),(atan(imag(V_s),real(V_s))*180/%pi))
printf("\n Sending end current = %.2f /_%.2f V \n\n",abs(I_s),(atan(imag(I_s),real(I_s))*180/%pi))

Line_voltage = sqrt(3) * abs(V_s) / 1000 ;
Line_current = abs(I_s);
Power_factor = cos(atan(imag(V_s),real(V_s)) - atan(imag(I_s),real(I_s)));
Power = sqrt(3) * Line_voltage * Line_current * Power_factor;
printf("\n\n Sending end line voltage = %.1f kV \n\n",Line_voltage)
printf("\n\n Sending end line current = %.1f A \n\n",Line_current)
printf("\n\n Sending end power = %.0f kW \n\n",Power)

voltage_regulation = (((abs(V_s)/abs(cosh_yl)) - V_r)/V_r)*100;
```

```
printf("\n\n Voltage Regulation = %.1f percent \n\n"
    ,voltage_regulation)

B = imag(yl)/l;
y1 = 2 * %pi / B;
Velocity = f * y1;
printf("\n\n Wavelength = %.0f mi \n\n",y1)
printf("\n\n Velocity = %.0f mi/s \n\n",Velocity)
```

Scilab code Exa 5.2 example1inpu

example1inpu

```
//Chapter 5
//Example 5.2
//Page 103
//example1inpu
clear;clc;

//Given
l = 230 ; //in mi
f = 60 ; //in Hz
P = 125e6 ; //in W
V = 215e3 ; //in V

//From Table A.1 and A.2 for 30ft Rook
//z = R + i(Xa + Xd)
z = 0.1603 + %i * (0.415+0.4127);

//From Table A.1 and A.3 for 30ft Rook
y = %i * [1e-6 / ( 0.0950 + 0.1008)]

//Calculations
```

```
yl = sqrt(y*z)*l;
Z_c = sqrt(z/y);
V_r = V / sqrt(3);
I_r = P / (sqrt(3)*V);

cosh_yl = cosh(real(yl)) * cos(imag(yl)) + %i * sinh
    (real(yl)) * sin(imag(yl));
sinh_yl = sinh(real(yl)) * cos(imag(yl)) + %i * cosh
    (real(yl)) * sin(imag(yl));

//Per Unit calculations
Base_impedance = V^2 / P;
Base_current = P / (sqrt(3)*V);
Z_c_pu = Z_c / Base_impedance;
V_r_pu = (V / sqrt(3)) / (V / sqrt(3));
I_r_pu = (P / (sqrt(3)*V)) / Base_current;

V_s_pu = V_r_pu * cosh_yl + I_r_pu * Z_c_pu *
    sinh_yl;
I_s_pu = I_r_pu * cosh_yl + V_r_pu * sinh_yl /
    Z_c_pu;

Line_voltage = abs(V_s_pu)*V / 1000;
Line_current = abs(I_s_pu)*Base_current;

printf("\n\n Sending end line voltage = %.1f V \n\n"
    ,Line_voltage)
printf("\n\n Sending end line current = %.1f A \n\n"
    ,Line_current)
```

Scilab code Exa 5.3 equivalentpicircuit

```
equivalentpicircuit
```

```
//Chapter 5
//Example 5.3
//Page 106
//equivalentpicircuit
clear;clc;

//Given
l = 230 ; //in mi
f = 60 ; //in Hz
P = 125e6 ; //in W
V = 215e3 ; //in V

//From Table A.1 and A.2 for 30ft Rook
//z = R + i(Xa + Xd)
z = 0.1603 + %i * (0.415+0.4127);

//From Table A.1 and A.3 for 30ft Rook
y = %i * [1e-6 / ( 0.0950 + 0.1008)]

//Calculations
yl = sqrt(y*z)*l;
Z_c = sqrt(z/y);

cosh_yl = cosh(real(yl)) * cos(imag(yl)) + %i * sinh(real(yl)) * sin(imag(yl));
sinh_yl = sinh(real(yl)) * cos(imag(yl)) + %i * cosh(real(yl)) * sin(imag(yl));

//Equivalent pi circuit
Z1 = Z_c * sinh_yl;
Y1_2 = (cosh_yl - 1)/(Z_c * sinh_yl);

disp('Equivalent PI circuit')
printf("\n\n Total series impedance of the line = %.2f /_%.2f ohm in series arm \n\n",abs(Z1),atan(imag(Z1),real(Z1))*180/%pi)
printf("\n\n Total Shunt admittance of the line = %.6f /_%.2f mho in each shunt arm \n\n",abs(Y1_2),
```

```
    atan(imag(Y1_2),real(Y1_2))*180/%pi)

//Nominal pi Circuit
Z = l * z;
Y_2 = y * l/2;

disp('Nominal PI circuit')
printf("\n\n Total series impedance of the line = %
    .2f /_%.2f ohm in series arm \n\n",abs(Z),atan(
    imag(Z),real(Z))*180/%pi)
printf("\n\n Total Shunt admittance of the line = %
    .6f /_%.2f mho in each shunt arm \n\n",abs(Y_2),
    atan(imag(Y_2),real(Y_2))*180/%pi)

zp = ((abs(Z)-abs(Z1))/abs(Z1))*100;
yp = ((abs(Y_2)-abs(Y1_2))/abs(Y1_2))*100;

printf("\n\n Line impedace of the series arm of the
    nominal pi exceeds that of equivalent pi by %.1f
    percent \n\n",zp)
printf("\n\n Conductance of the shunt arms of the
    nominal pi is %.0f percent less than that of
    equivalent pi \n\n",abs(yp))
```

Scilab code Exa 5.4 compensation

compensation

```
//Chapter 5
//Example 5.4
//Page 111
//compensation
clear;clc;
```

```

//Given
l = 230 ; //in mi
f = 60 ; //in Hz
P = 125e6 ; //in W
V = 215e3 ; //in V

//From Table A.1 and A.2 for 30ft Rook
//z = R + i(Xa + Xd)
z = 0.1603 + %i * (0.415+0.4127);

//From Table A.1 and A.3 for 30ft Rook
y = %i * [1e-6 / ( 0.0950 + 0.1008)]

//Calculations
yl = sqrt(y*z)*l;
Z_c = sqrt(z/y);

cosh_yl = cosh(real(yl)) * cos(imag(yl)) + %i * sinh(real(yl)) * sin(imag(yl));
sinh_yl = sinh(real(yl)) * cos(imag(yl)) + %i * cosh(real(yl)) * sin(imag(yl));

//Equivalent pi circuit
Z1 = Z_c * sinh_yl;
Y1_2 = (cosh_yl - 1)/(Z_c * sinh_yl);

A = cosh_yl;D = cosh_yl;
B = Z1;
C = sinh_yl / Z_c;

disp('For an uncompensated line')
printf("\n\n A = D = %.4f /_%.2f \n\n",abs(A),(atan(imag(A),real(A))*180/%pi))
printf("\n\n B = %.4f /_%.2f ohm \n\n",abs(B),(atan(imag(B),real(B))*180/%pi))
printf("\n\n C = %f /_%.2f mho \n\n",abs(C),(atan(imag(C),real(C))*180/%pi))
```

```

//For a series compensation factor of 70%
cf = 0.7
B1 = Z1 - %i * cf * l * (0.415 + 0.4127) ;//X_a = 0.415 ohm/mi,X_d = 0.4127 in
A1 = B1 * Y1_2 + 1;
C1 = 2 * Y1_2 + B1 * (Y1_2)^2;

disp('For a series compensation factor of 70%')
printf("\n\n B = %.2f /_%.2f ohm \n\n",abs(B1),(atan(imag(B1),real(B1))*180/%pi))
printf("\n\n A = %.3f /_%.2f \n\n",abs(A1),(atan(imag(A1),real(A1))*180/%pi))
printf("\n\n C = %f /_%.2f mho \n\n",abs(C1),(atan(imag(C1),real(C1))*180/%pi))
```

Scilab code Exa 5.5 voltageregulation

`voltageregulation`

```
//Chapter 5
//Example 5.5
//Page 112
//voltageregulation
clear;clc;

//Given
l = 230 ; //in mi
f = 60 ; //in Hz
P = 125e6 ; //in W
V = 215e3 ; //in V

//From Table A.1 and A.2 for 30ft Rook
```

```
//z = R + i(Xa + Xd)
z = 0.1603 + %i * (0.415+0.4127);

//From Table A.1 and A.3 for 30ft Rook
y = %i * [1e-6 / ( 0.0950 + 0.1008)]

//Calculations
yl = sqrt(y*z)*l;
Z_c = sqrt(z/y);
V_r = V / sqrt(3);
I_r = P / (sqrt(3)*V);
yl = sqrt(y*z)*l;
Z_c = sqrt(z/y);

cosh_yl = cosh(real(yl)) * cos(imag(yl)) + %i * sinh
    (real(yl)) * sin(imag(yl));
sinh_yl = sinh(real(yl)) * cos(imag(yl)) + %i * cosh
    (real(yl)) * sin(imag(yl));

V_s = V_r * cosh_yl + I_r * Z_c * sinh_yl;
I_s = I_r * cosh_yl + V_r * sinh_yl / Z_c;

//Equivalent pi circuit
Z1 = Z_c * sinh_yl;
Y1_2 = (cosh_yl - 1)/(Z_c * sinh_yl);

//Total capacitive Susceptance
B_c = %i * y * l;

//For 70% Compensation
cf = 0.7;
B_L = - B_c * 0.7;

//From appendix
A = 1;D = 1;B = 0;
C = -%i*B_L;
```

```
//From Table A.6 for combining two networks in
    series
A_eq = cosh_yl + Z1 * C;
voltage_regulation = ((abs(V_s)/abs(A_eq))-V_r)*100/
    V_r;
printf("\n\n Voltage regulation = %.2f percent \n\n"
    ,voltage_regulation)
```

Scilab code Exa 5.6 reflection

`reflection`

```
//Chapter 5
//Example_5_6
//Page 119
//reflection
clear;clc;
V=120; //Applied DC voltage at the sending end
Zc=30; //Characteristic Impedance of the line
Zs=0; //Source Imedance taken zero since its not
    given
m=2; //Two times we are calculating.i.e Zr=90 and Zr
    =10
for j=1:m

    if j==1 then
        Zr=90; //for the first time  i.e Case(i)
    elseif j==2 then
        Zr=10; //for the second time  i.e Case(ii)
    end
rho_R=(Zr-Zc)/(Zr+Zc); //reflection coefficient for
    voltage at receiving end
```

```
rho_S=(Zs-Zc)/(Zs+Zc); //reflection coefficient for voltage at the sending end
printf('\n\nCase(%d) Reflected and Receiving End voltages When Zr=%dohm',j,Zr);
printf('\nAt time\t\tV+\t\tVr\t\tV-\n')
v_plus=V; //initial value at the instant of switching
Vr=0;Vr_plot=Vr;
n=5; // for 5 time periods
for i=0:5
    if i==0 then
      printf('%d\t\t%d-->\t\t%d-->\t\t%d-->\n',i,0,0,0); //for before the instant of switching
    else
      v_minus=rho_R*v_plus; //reflected wave the receiving end
      Vr=Vr+v_plus+v_minus; //receiving end voltage
      Vr_plot=[Vr_plot,Vr];  //saving Vr for plotting
      printf('%dT\t\t%d-->\t\t%d-->\t\t%d-->\n',i,v_plus,Vr,v_minus);
      v_plus=v_minus*rho_S; //second reflected wave at the sending end
    end
end
t=[0,1,3,5,7,9]; //time periods at which the receiving voltage has to be plotted
if j==1 then          //to discriminate between two cases
   clf();
   subplot(121);
   plot2d2(t,Vr_plot);
   xstring(t,Vr_plot,+string(Vr_plot));
elseif j==2 then
   subplot(122);
   plot2d2(t,Vr_plot);
   xstring(t,Vr_plot,+string(Vr_plot));
```

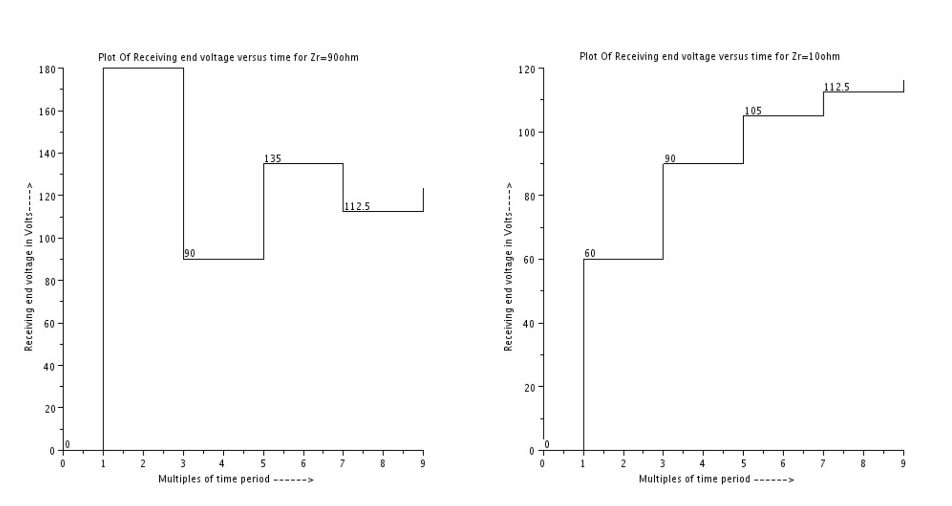

Figure 5.1: reflection

```
end
xlabel('Multiples of time period ------->');
ylabel('Receiving end voltage in Volts----->');
title('Plot Of Receiving end voltage versus time for Zr='+string(Zr)+'ohm');
end
```

Chapter 6

System Modelling

Scilab code Exa 6.1 Secondary

Secondary

```
//Chapter 6
//Example 6.1
//Page 142
//Secondary
clear;clc;
N_1 = 2000;
N_2 = 500;
V_1 = 1200 * (cos(0)+%i*sin(0));
I_1 = 5 * (cos(-30*%pi/180) + %i * sin(-30*%pi/180))
    ;

//Calculations
a = N_1 / N_2;
V_2 = V_1/a;
I_2 = a * I_1;
Z_2 = V_2 / I_2;
Z1_2 = Z_2 * a^2;
printf("\n\n V_2 = %.0f /_%.0f V \n\n",abs(V_2),((
    atan(imag(V_2),real(V_2)))*180/%pi))
```

```
printf("\n\n I_2 = %.0f /_%.0f A \n\n",abs(I_2),((
    atan(imag(I_2),real(I_2)))*180/%pi))
printf("\n\n Z_2 = %.0f /_%.0f ohm \n\n",abs(Z_2),((
    atan(imag(Z_2),real(Z_2)))*180/%pi))
printf("\n\n Z1_2 = %.0f /_%.0f ohm \n\n",abs(Z1_2)
    ,((atan(imag(Z1_2),real(Z1_2)))*180/%pi))
```

Scilab code Exa 6.2 voltageregulation

voltageregulation

```
//Chapter 6
//Example 6.2
//Page 144
//voltageregulation
clear;clc;

N_1 = 2000;
N_2 = 500;
V_1 = 1200 * (cos(0)+%i*sin(0));
r1 = 2;
r2 = 0.125;
x1 = 8;
x2 = 0.5;
Z_2 = 12;

//Calculations
a = N_1 / N_2;
R1 = r1 + a^2 * r2;
X1 = x1 + a^2 * x2;
Z1_2 = Z_2 * a^2;

I_1 = V_1 / (Z1_2 + R1 + %i * X1);
```

```
V_2 = I_1 * Z1_2 / a;
voltage_regulation = ((V_1/4)-abs(V_2))*100/V_2;

printf("\n\n I_1 = %.2f /_%.2f A \n\n",abs(I_1),((atan(imag(I_1),real(I_1)))*180/%pi))
printf("\n\n V_2 = %.2f /_%.2f V \n\n",abs(V_2),((atan(imag(V_2),real(V_2)))*180/%pi))
printf("\n\n Voltage Regulation = %.2f percent \n\n",voltage_regulation)
```

Scilab code Exa 6.3 autotransformer

autotransformer

```
//Chapter 6
//Example 6.3
//Page 145
//autotransformer
clear;clc;

//Given
P = 30e3;
V_lt = 120;
V_ht = 240;

//Calculations
I_1 = P / V_lt;
I_2 = P / V_ht;
V_2 = V_lt + V_ht;
I_in = I_1 + I_2;
input_kva = I_in * V_lt / 1e3;
output_kva = I_2 * V_2 / 1e3;

```

```
printf("\n\n Input kVA = %.0f kVA \n\n",input_kva)
printf("\n\n Output kVA = %.0f kVA \n\n",output_kva)
```

Scilab code Exa 6.4 leakagereactance

`leakagereactance`

```
//Chapter 6
//Example 6.4
//Page 147
//leakagereactance
clear;clc;

//Given
V_lt = 110;
V_ht = 440;
P = 2.5e3;
x_lt = 0.06;

//Calculations
disp('Viewed from low-tension side')
lt_base_impedance = (V_lt)^2 / P;
printf("\n Leakage reactance from low-tension side =
    %.2f ohm",x_lt)
printf("\n Low-tension base impedance = %.2f ohmn",
   lt_base_impedance)
X_lt_pu = x_lt / lt_base_impedance;
printf("\n Leakage reactance in per unit from Low-
   tension side = %.4f per unit \n\n",X_lt_pu)
disp('Viewed from high-tension side')
x_ht = x_lt * (V_ht / V_lt)^2;
ht_base_impedance = (V_ht)^2 / P;
```

```
printf("\n Leakage reactance from high-tension side
    = %.2f ohm",x_ht)
printf("\n High-tension base impedance = %.2f ohmn",
    ht_base_impedance)
X_ht_pu = x_ht / ht_base_impedance;
printf("\n Leakage reactance in per unit from Low-
    tension side = %.4f per unit \n\n",X_ht_pu)
```

Scilab code Exa 6.5 ABCtransformer

ABCtransformer

```
//Chapter 6
//Example 6.5
//Page 147
//ABCtransformer
clear;clc;

//Given
P_AB = 10e6;
V_AB_lt = 13.8e3;
V_AB_ht = 138e3;
x_AB = %i*0.1;
P_BC = 10e6;
V_BC_lt = 69e3;
V_BC_ht = 138e3;
x_BC = %i*0.08;
P_base_B = 10e6;
V_base_B = 138e3;
Z_L = 300;
V_load = 66e3;

//Calculations
```

```
V_base_A = (V_AB_lt/V_AB_ht)*V_base_B;
V_base_C = (V_BC_lt/V_BC_ht)*V_base_B;
base_impedance_C = (V_base_C)^2 / P_BC;
Z_L_pu = Z_L / base_impedance_C;

//impedance diagram is shown in the xcos file
V_load_pu = V_load / V_base_C;
I_L_pu = V_load_pu / Z_L_pu;
voltage_input = (I_L_pu * (x_AB + x_BC)) + V_load_pu
    ;
voltage_regulation = (abs(voltage_input)-abs(
    V_load_pu))*100/abs(V_load_pu);

printf("\n\n Voltage regulation = %.2f percent \n\n"
    ,voltage_regulation)
```

Scilab code Exa 6.6 3phasetransformers

3phasetransformers

```
//Chapter 6
//Example 6.6
//Page 151
//3phasetransformers
clear;clc;

//Given
P = 25e6;
V_ht = 38.1e3;
V_lt = 3.81e3;
R_l = 0.6;
P_ht_base = 75e6;
V_ht_base = 66e3;
```

```
//Low-tension side base ratings
disp('Low-tension side')
P_lt_base = P_ht_base;
V_lt_base = (V_lt/(V_ht*sqrt(3)))*V_ht_base;
printf("\n Base for low tension side is %.0f MVA,%.2
    f kV",P_lt_base/1e6,V_lt_base/1e3)
Z_lt_base = (V_lt_base)^2/P_lt_base;
R_lt_l_base = R_l / Z_lt_base;
printf("\n Base impedance for the low-tension side =
     %.2f ohm",Z_lt_base)
printf("\n Per unit impedance of load on the low-
    tension side = %.2f per unit \n\n",R_lt_l_base)

disp('High-tension side')
R_l_ht = R_l * ((V_ht*sqrt(3))/V_lt)^2;
Z_ht_base = (V_ht_base)^2 / P_ht_base;
R_ht_l_base = R_l_ht / Z_ht_base;
printf("\n Base impedance for the high-tension side
    = %.2f ohm",Z_ht_base)
printf("\n Per unit impedance of load on the high-
    tension side = %.2f per unit",R_ht_l_base)
```

check Appendix AP 1 for dependency:

`pucalc.sci`

Scilab code Exa 6.7 3perunitreactance

`3perunitreactance`

```
//Chapter 6
//Example 6.7
//Page 152
```

```
//3perunitreactance
//run clear command then execute dependancy file and
    then the source file
//dependency file is pucalc.sci
clc;

//Given
z = 0.121;
P = 400e6;
V_ht = 220e3;
V_lt = 22e3;

V_ht_base = 230e3;
P_ht_base = 100e6;

z_pu = z * P / (V_lt)^2;
printf("\n\n On its own base the transformer
   reactance = %.2f per unit",z_pu)
z_new_pu = pucalc(z_pu,V_ht,V_ht_base,P_ht_base,P);
printf("\n\n On the chosen base the reactance = %.4f
    per unit",z_new_pu)
```

Scilab code Exa 6.8 3windingtransformer

3windingtransformer

```
//Chapter 6
//Example 6.8
//Page 154
//3windingtransformer
clear;clc;

//Given
```

```
Z_ps = %i * 0.07;
P_ps_base = 15e6;
V_ps_base = 66e3;

Z_pt = %i * 0.09;
P_pt_base = 15e6;
V_pt_base = 66e3;

Z_st = %i * 0.08;
P_st_base = 10e6;
V_st_base = 13.2e3;

//Calculations
Z_st_new = Z_st * P_ps_base / P_st_base;
Z_p = (Z_ps + Z_pt - Z_st_new)/2;
Z_s = (Z_ps - Z_pt + Z_st_new)/2;
Z_t = (-Z_ps + Z_pt + Z_st_new)/2;
disp(Z_p,'Z_p in per unit = ')
disp(Z_s,'Z_s in per unit = ')
disp(Z_t,'Z_t in per unit = ')
```

Scilab code Exa 6.9 3winding3ex8

3winding3ex8

```
//Chapter 6
//Example 6.9
//Page 155
//3winding3ex8
clear;clc;

//Given
Z_ps = %i * 0.07;
```

```
P_ps_base = 15e6;
V_ps_base = 66e3;
P_r_load = 5e6;
V_r_load = 2.3e3;
P_m = 7.5e6;
V_m = 13.2e3;
R_l = 1;
X_11 = 0.20;

Z_pt = %i * 0.09;
P_pt_base = 15e6;
V_pt_base = 66e3;

Z_st = %i * 0.08;
P_st_base = 10e6;
V_st_base = 13.2e3;

//Calculations
Z_st_new = Z_st * P_ps_base / P_st_base;
Z_p = (Z_ps + Z_pt - Z_st_new)/2;
Z_s = (Z_ps - Z_pt + Z_st_new)/2;
Z_t = (-Z_ps + Z_pt + Z_st_new)/2;

R_pu = R_l * P_ps_base / P_r_load;
X11_pu = X_11 *  P_ps_base / P_m;
printf("\n\n On a base of 15MVA,2.3kV load
   resistance = %.1f per unit",R_pu)
printf("\n\n Reactance of the motor at a base of 15
   MVA,13.2kV = %.2f per unit",X11_pu)

//Reactance diagram is drawn in the xcos file
```

This code can be downloaded from the website wwww.scilab.in check

Appendix AP 1 for dependency:

pucalc.sci

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 6.10 impedance

impedance

```
//Chapter 6
//Example 6.10
//Page 159
//impedance
//run clear command then execute dependancy file and
    then the source file
//dependency file is pucalc.sci
clc;

//Given
P_g = 300e6;
V_g = 20e3;
X11_g = 0.20;
l = 64;
V_m = 13.2e3;
P_m1 = 200e6;
P_m2 = 100e6;
X11_m = 0.20;
T1_P = 350e6;
T1_vht = 230e3;
T1_vlt = 20e3;
x_T1 = 0.10;
T2_1_P = 100e6;
T2_1_vht = 127e3;
T2_1_vlt = 13.2e3;
x_T2 = 0.10;
```

```
x_line = 0.5;//ohm per km
V_base = V_g;
P_base = P_g;

//Calculations
T2_P = 3*T2_1_P;
T2_vht = sqrt(3)*T2_1_vht;
T2_vlt = T2_1_vlt;
V_base_line = (T1_vht/T1_vlt)*V_base;
V_base_m = V_base_line * (T2_vlt/T2_vht);
x_T1_base = x_T1 * (P_base/T1_P);
x_T2_base = x_T2 * (T2_vlt/V_base_m);
z_line_base = (V_base_line)^2/P_base;
x_line_pu = x_line * l / z_line_base;
X11_m1_pu = pucalc(X11_m,V_m,V_base_m,P_base,P_m1);
X11_m2_pu = pucalc(X11_m,V_m,V_base_m,P_base,P_m2);
//Reactance diagram is given in xcos file
disp('Base Voltages in different parts of circuit')
printf("\n Generator voltage = %.0f kV",V_g/1e3)
printf("\n Line voltage = %.0f kV",V_base_line/1e3)
printf("\n Motor voltage = %.1f kV \n\n\n",V_base_m
    /1e3)

disp('Base reactance in different parts of circuit')
printf("\n Transformer 1 reactance = %.4f per unit",
    x_T1_base)
printf("\n Transformer 2 reactance = %.4f per unit",
    x_T2_base)
printf("\n Line reactance = %.4f per unit",x_line_pu
    )
printf("\n Motor 1 reactance = %.4f per unit",
    X11_m1_pu)
printf("\n Motor 2 reactance = %.4f per unit",
    X11_m2_pu)
//impedance diagram is shown in the xcos file
```

Scilab code Exa 6.11 terminalvoltage

terminalvoltage

```
//Chapter 6
//Example 6.11
//Page 160
//terminalvoltage
clear;clc;

//Given
P_g = 300e6;
V_g = 20e3;
X11_g = 0.20;
l = 64;
V_m = 13.2e3;
P_m1 = 120e6;
P_m2 = 60e6;
X11_m = 0.20;
T1_P = 350e6;
T1_vht = 230e3;
T1_vlt = 20e3;
x_T1 = 0.10;
T2_1_P = 100e6;
T2_1_vht = 127e3;
T2_1_vlt = 13.2e3;
x_T2 = 0.10;
x_line = 0.5;//onhm per km
V_base = V_g;
P_base = P_g;

//Calculations
```

```
T2_P = 3*T2_1_P;
T2_vht = sqrt(3)*T2_1_vht;
T2_vlt = T2_1_vlt;
V_base_line = (T1_vht/T1_vlt)*V_base;
V_base_m = V_base_line * (T2_vlt/T2_vht);
x_T1_base = x_T1 * (P_base/T1_P);
x_T2_base = x_T2 * (T2_vlt/V_base_m);
z_line_base = (V_base_line)^2/P_base;
x_line_pu = x_line * l / z_line_base;
P = P_m1 + P_m2;
P_pu = P / P_base;
V = V_m / V_base_m;
I = P_pu / V;
Vg = V + I * (%i * x_T1_base + %i * x_T2_base + %i *
    x_line_pu);
V_terminal = abs(Vg) * V_g;
printf("\n\n The generator terminal voltage = %.2f
    kV",V_terminal / 1e3)
```

Chapter 7

Network Calculations

Scilab code Exa 7.1 busadmittancematrix

`busadmittancematrix`

```
//chapter 7
//Example 7.1
//Page 170
//busadmittancematrix
clear;clc;
//Given
//Voltage Sources
Ea = 1.5;
Eb = 1.5*(cos(-36.87 * %pi / 180) + %i * sin(-36.87
   * %pi / 180))
Ec = 1.5;
//admittances
Ya = -%i*0.8;
Yb = Ya;
Yc= Ya;
Yd = -%i*5;
Ye = -%i*8;
Yf = -%i*4;
Yg = -%i*2.5;
```

```
Yh = Yd;
//current sourcs
I1 = Ea * Ya;
I2 = Eb * Yb;
I3 = I1;
I4 = 0;
disp('Current Sources are')
printf("\n I1 = - j%.2f per unit",-imag(I1))
printf("\n I2 = %.2f - j%.2f per unit",real(I2),-
    imag(I2))
printf("\n I3 = - j%.2f per unit \n\n",-imag(I3))
//Self-admittances
Y11 = Yd + Yf + Ya;
Y22 = Yh + Yg + Yb;
Y33 = Ye + Yc + Yg + Yf;
Y44 = Yd + Ye + Yh;
disp('Self-admittances are')
printf("\n Y11 = - j%.2f per unit",-imag(Y11))
printf("\n Y22 = - j%.2f per unit",-imag(Y22))
printf("\n Y33 = - j%.2f per unit",-imag(Y33))
printf("\n Y44 = - j%.2f per unit\n\n",-imag(Y44))
//Mutual-admittances
Y12 = 0;Y21 = Y12;
Y13 = -Yf;Y31 = Y13;
Y14 = -Yd;Y41 = Y14;
Y23 = -Yg;Y32 = Y23;
Y24 = -Yh;Y42 = Y24;
Y34 = -Ye;Y43 = Y34;
disp('Mutual admittances are')
printf("\n Y12 = Y21 = %.2f per unit",imag(Y12))
printf("\n Y13 = Y31 = j%.2f per unit",imag(Y13))
printf("\n Y14 = Y41 = j%.2f per unit",imag(Y14))
printf("\n Y23 = Y32 = j%.2f per unit",imag(Y23))
printf("\n Y24 = Y42 = j%.2f per unit",imag(Y24))
printf("\n Y34 = Y43 = j%.2f per unit \n\n",imag(Y34
    ))
//Matrix Form
I = [I1 ; I2 ; I3 ; I4];
```

```
Y = [Y11 Y12 Y13 Y14;Y21 Y22 Y23 Y24;Y31 Y32 Y33 Y34
    ;Y41 Y42 Y43 Y44];
disp('Current Vector =')
disp(I)
disp('Bus admittance matrix =')
disp(Y)
```

Scilab code Exa 7.2 solvingBAM

solvingBAM

```
//chapter 7
//Example 7.2
//Page 171
//solvingBAM
clear;clc;
//Voltage Sources
Ea = 1.5;
Eb = 1.5*(cos(-36.87 * %pi / 180) + %i * sin(-36.87
    * %pi / 180))
Ec = 1.5;
//admittances
Ya = -%i*0.8;
Yb = Ya;
Yc= Ya;
Yd = -%i*5;
Ye = -%i*8;
Yf = -%i*4;
Yg = -%i*2.5;
Yh = Yd;
//current sourcs
I1 = Ea * Ya;
I2 = Eb * Yb;
```

```
I3 = I1;
I4 = 0;
//Self-admittances
Y11 = Yd + Yf + Ya;
Y22 = Yh + Yg + Yb;
Y33 = Ye + Yc + Yg + Yf;
Y44 = Yd + Ye + Yh;
//Mutual-admittances
Y12 = 0;Y21 = Y12;
Y13 = -Yf;Y31 = Y13;
Y14 = -Yd;Y41 = Y14;
Y23 = -Yg;Y32 = Y23;
Y24 = -Yh;Y42 = Y24;
Y34 = -Ye;Y43 = Y34;
//Matrix Form
I = [I1 ; I2 ; I3 ; I4];
Y = [Y11 Y12 Y13 Y14;Y21 Y22 Y23 Y24;Y31 Y32 Y33 Y34
    ;Y41 Y42 Y43 Y44];
V = Y\I;
disp('Node Voltages V1,V2,V3 and V4 in per unit is')
disp(V)
disp('In polar form')
printf("\n V1 = %.2f /_%.2f per unit",abs(V(1,1)),
    atan(imag(V(1,1)),real(V(1,1))) * 180 / %pi)
printf("\n V2 = %.2f /_%.2f per unit",abs(V(2,1)),
    atan(imag(V(2,1)),real(V(2,1))) * 180 / %pi)
printf("\n V3 = %.2f /_%.2f per unit ",abs(V(3,1)),
    atan(imag(V(3,1)),real(V(3,1))) * 180 / %pi)
printf("\n V4 = %.2f /_%.2f per unit \n\n",abs(V
    (4,1)),atan(imag(V(4,1)),real(V(4,1))) * 180 /
    %pi)
```

Scilab code Exa 7.3 matrixpartition

matrixpartition

```
//chapter 7
//Example 7.3
//Page 177
//matrixpartition
clear;clc;
//Given
//Voltage Sources
Ea = 1.5;
Eb = 1.5*(cos(-36.87 * %pi / 180) + %i * sin(-36.87
    * %pi / 180))
Ec = 1.5;
//admittances
Ya = -%i*0.8;
Yb = Ya;
Yc= Ya;
Yd = -%i*5;
Ye = -%i*8;
Yf = -%i*4;
Yg = -%i*2.5;
Yh = Yd;
//Self-admittances
Y11 = Yd + Yf + Ya;
Y22 = Yh + Yg + Yb;
Y33 = Ye + Yg + Yf;
Y44 = Yd + Ye + Yh;
//Mutual-admittances
Y12 = 0;Y21 = Y12;
Y13 = -Yf;Y31 = Y13;
Y14 = -Yd;Y41 = Y14;
Y23 = -Yg;Y32 = Y23;
Y24 = -Yh;Y42 = Y24;
Y34 = -Ye;Y43 = Y34;
//Bus Impedance Matrix
Y = [Y11 Y12 Y13 Y14;Y21 Y22 Y23 Y24;Y31 Y32 Y33 Y34
    ;Y41 Y42 Y43 Y44];
```

```
K = Y(1:2,1:2);
L = Y(1:2,3:4);
L_T = Y(3:4,1:2);
M = Y(3:4,3:4);
M_1 = inv(M);
LMT = L * M_1 *  L_T;
Ybus = K - LMT;
Y_12 = - Ybus(1,2);
Y_10 = Ybus(1,1) - Y_12;
Y_20 = Y_10;
printf("\n Admittance between buses 1 and 2 = - j%.4f per unit\n",-imag(Y_12))
printf("\n Admittance between buse 1 and reference bus = - j%.4f per unit\n",-imag(Y_10))
printf("\n Admittance between buse 2 and reference bus = - j%.4f per unit\n",-imag(Y_20))
Z = 1/Y_12 + 1/Y_10 + 1/Y_20;
I = (Ea-Eb) / Z;
printf("\n I = %.2f /_%.2f per unit \n",abs(I),atan(imag(I),real(I)) * 180 / %pi)
Pa = Ea * I';
printf("\n Power out of source ''a'' = %.3f + j%.3f per unit \n",real(Pa),imag(Pa))
Pb = Eb * I';
printf("\n Power out of source ''b'' = %.3f - j%.3f per unit \n",real(Pb),-imag(Pb))
Var = (abs(I))^2 * imag(Z);
printf("\n Reactie voltamperes in circuit equivalent = %.3f per unit \n",Var)
V_1 = Ea - I/Y_10;
printf("\n Voltage at node 1 = %.3f - j%.3f per unit \n",real(V_1),-imag(V_1))
```

Scilab code Exa 7.4 matrixmanipulation

matrixmanipulation

```
//chapter 7
//Example 7.4
//Page 177
//matrixmanipulation
clear;clc;
//admittances
Ya = -%i*0.8;
Yb = Ya;
Yc= Ya;
Yd = -%i*5;
Ye = -%i*8;
Yf = -%i*4;
Yg = -%i*2.5;
Yh = Yd;
//Self-admittances
Y11 = Yd + Yf + Ya;
Y22 = Yh + Yg + Yb;
Y33 = Ye + Yg + Yf;
Y44 = Yd + Ye + Yh;
//Mutual-admittances
Y12 = 0;Y21 = Y12;
Y13 = -Yf;Y31 = Y13;
Y14 = -Yd;Y41 = Y14;
Y23 = -Yg;Y32 = Y23;
Y24 = -Yh;Y42 = Y24;
Y34 = -Ye;Y43 = Y34;
//Bus Impedance Matrix
Y = [Y11 Y12 Y13 Y14;Y21 Y22 Y23 Y24;Y31 Y32 Y33 Y34
    ;Y41 Y42 Y43 Y44];
//Removing node 4
[row_4,column_4] = size(Y)
Y_bus_4 = zeros(row_4-1,column_4-1);
for a = 1:row_4-1
    for b = 1:column_4-1
```

```
    Y_bus_4(a,b) = Y(a,b) - (Y(a,column_4) * Y(row_4
       ,b) / Y(row_4,column_4))
    end
end
disp('Y bus matrix after removing node four')
disp(Y_bus_4)
//Removing node 3
[row_3,column_3] = size(Y_bus_4)
Y_bus_3 = zeros(row_3-1,column_3-1);
for c = 1:row_3-1
    for d = 1:column_3-1
    Y_bus_3(c,d) = Y_bus_4(c,d) - (Y_bus_4(c,
       column_3) * Y_bus_4(row_3,d) / Y_bus_4(row_3,
       column_3))
    end
end
disp('Y bus matrix after removing node three')
disp(Y_bus_3)
```

Scilab code Exa 7.5 introcapacitor

introcapacitor

```
//chapter 7
//Example 7.5
//Page 181
//introcapacitor
clear;clc;
//Voltage Sources
Ea = 1.5;
Eb = 1.5*(cos(-36.87 * %pi / 180) + %i * sin(-36.87
   * %pi / 180))
Ec = 1.5;
```

```
//admittances
Ya = -%i*0.8;
Yb = Ya;
Yc= Ya;
Yd = -%i*5;
Ye = -%i*8;
Yf = -%i*4;
Yg = -%i*2.5;
Yh = Yd;
//Value of capacitor introduced in node 4
C = 5.0 ;//in per unit
Xc = %i*C;
//current sourcs
I1 = Ea * Ya;
I2 = Eb * Yb;
I3 = I1;
I4 = 0;
//Self-admittances
Y11 = Yd + Yf + Ya;
Y22 = Yh + Yg + Yb;
Y33 = Ye + Yc + Yg + Yf;
Y44 = Yd + Ye + Yh;
//Mutual-admittances
Y12 = 0;Y21 = Y12;
Y13 = -Yf;Y31 = Y13;
Y14 = -Yd;Y41 = Y14;
Y23 = -Yg;Y32 = Y23;
Y24 = -Yh;Y42 = Y24;
Y34 = -Ye;Y43 = Y34;
//Matrix Form
I = [I1 ; I2 ; I3 ; I4];
Y = [Y11 Y12 Y13 Y14;Y21 Y22 Y23 Y24;Y31 Y32 Y33 Y34
    ;Y41 Y42 Y43 Y44];
V = Y\I;
E_th = V(4,1);
Z = inv(Y);
Z_th = Z(4,4);
I_c = E_th / (Z_th - Xc);
```

```
disp('Thevenin equivalent of the circuit behind node
    four')
printf("\n Eth = %.2f /_%.2f per unit \n\n",abs(E_th
   ),atan(imag(E_th),real(E_th)) * 180 / %pi)
disp('Thevenin equivalent impedance')
printf("\n Z_th = j%.2f per unit \n\n",imag(Z_th))
disp('Current drawn by the capacitor')
printf("\n Ic = %.2f /_%.2f per unit \n\n",abs(I_c),
   atan(imag(I_c),real(I_c)) * 180 / %pi)
```

Scilab code Exa 7.6 currentinjection

currentinjection

```
//chapter 7
//Example 7.6
//Page 181
//currentinjection
clear;clc;
//Voltage Sources
Ea = 1.5;
Eb = 1.5*(cos(-36.87 * %pi / 180) + %i * sin(-36.87
   * %pi / 180))
Ec = 1.5;
//admittances
Ya = -%i*0.8;
Yb = Ya;
Yc= Ya;
Yd = -%i*5;
Ye = -%i*8;
Yf = -%i*4;
Yg = -%i*2.5;
Yh = Yd;
```

```
//current sourcs
I1 = Ea * Ya;
I2 = Eb * Yb;
I3 = I1;
I4 = 0;
//Current Injected
I4_1 = -0.316 * (cos(78.03 * %pi / 180) + %i * sin(78.03 * %pi / 180));
//Self-admittances
Y11 = Yd + Yf + Ya;
Y22 = Yh + Yg + Yb;
Y33 = Ye + Yc + Yg + Yf;
Y44 = Yd + Ye + Yh;
//Mutual-admittances
Y12 = 0;Y21 = Y12;
Y13 = -Yf;Y31 = Y13;
Y14 = -Yd;Y41 = Y14;
Y23 = -Yg;Y32 = Y23;
Y24 = -Yh;Y42 = Y24;
Y34 = -Ye;Y43 = Y34;
//Matrix Form
I = [I1 ; I2 ; I3 ; I4];
Y = [Y11 Y12 Y13 Y14;Y21 Y22 Y23 Y24;Y31 Y32 Y33 Y34;Y41 Y42 Y43 Y44];
V = Y\I;
Z = inv(Y);
V_ci_1 = I4_1 * Z(1,4);
V_ci_2 = I4_1 * Z(2,4);
V_ci_3 = I4_1 * Z(3,4);
V_ci_4 = I4_1 * Z(4,4);
disp('Voltages with all emfs shorted')
printf("\n V1 = %.2f /_%.2f per unit",abs(V_ci_1),atan(imag(V_ci_1),real(V_ci_1)) * 180 / %pi)
printf("\n V2 = %.2f /_%.2f per unit",abs(V_ci_2),atan(imag(V_ci_2),real(V_ci_2)) * 180 / %pi)
printf("\n V3 = %.2f /_%.2f per unit ",abs(V_ci_3),atan(imag(V_ci_3),real(V_ci_3)) * 180 / %pi)
printf("\n V4 = %.2f /_%.2f per unit \n\n",abs(
```

```
    V_ci_4),atan(imag(V_ci_4),real(V_ci_4)) * 180 /
    %pi)
disp('Resulting voltages are determined by
    superposition of voltages caused by injected
    current and emfs shorted to the node voltage')
V_new_1 = V(1,1) + V_ci_1;
V_new_2 = V(2,1) + V_ci_2;
V_new_3 = V(3,1) + V_ci_3;
V_new_4 = V(4,1) + V_ci_4;
printf("\n V1 = %.2f /_%.2f per unit",abs(V_new_1),
    atan(imag(V_new_1),real(V_new_1)) * 180 / %pi)
printf("\n V2 = %.2f /_%.2f per unit",abs(V_new_2),
    atan(imag(V_new_2),real(V_new_2)) * 180 / %pi)
printf("\n V3 = %.2f /_%.2f per unit ",abs(V_new_3),
    atan(imag(V_new_3),real(V_new_3)) * 180 / %pi)
printf("\n V4 = %.2f /_%.2f per unit \n\n",abs(
    V_new_4),atan(imag(V_new_4),real(V_new_4)) * 180
    / %pi)
```

Scilab code Exa 7.7 matrixmodification

matrixmodification

```
//chapter 7
//Example 7.7
//Page 186
//matrixmodification
clear;clc;
//Voltage Sources
Ea = 1.5;
Eb = 1.5*(cos(-36.87 * %pi / 180) + %i * sin(-36.87
    * %pi / 180))
Ec = 1.5;
```

```
//admittances
Ya = -%i*0.8;
Yb = Ya;
Yc= Ya;
Yd = -%i*5;
Ye = -%i*8;
Yf = -%i*4;
Yg = -%i*2.5;
Yh = Yd;
//Capacitor
Zb = -%i * 5
//current sourcs
I1 = Ea * Ya;
I2 = Eb * Yb;
I3 = I1;
I4 = 0;
//Self-admittances
Y11 = Yd + Yf + Ya;
Y22 = Yh + Yg + Yb;
Y33 = Ye + Yc + Yg + Yf;
Y44 = Yd + Ye + Yh;
//Mutual-admittances
Y12 = 0;Y21 = Y12;
Y13 = -Yf;Y31 = Y13;
Y14 = -Yd;Y41 = Y14;
Y23 = -Yg;Y32 = Y23;
Y24 = -Yh;Y42 = Y24;
Y34 = -Ye;Y43 = Y34;
//Matrix Form
I = [I1 ; I2 ; I3 ; I4];
Y = [Y11 Y12 Y13 Y14;Y21 Y22 Y23 Y24;Y31 Y32 Y33 Y34
    ;Y41 Y42 Y43 Y44];
V = Y\I;
Z = inv(Y);
disp('Original bus impedance matrix')
disp(Z)
[m,n] = size(Z)
for i = 1:m
```

```
    for j = 1:n
    Z(5,i) = Z(i,j);
    Z(i,5) = Z(i,j)
    end
end
Z(5,5) = Z(4,4) + Zb;
disp('Modified bus impedance matrix')
disp(Z)
[m1,n1] = size(Z);
Z_new = zeros(m1-1,n1-1);
for c = 1:m1-1
    for d = 1:n1-1
    Z_new(c,d) = Z(c,d) - ((Z(c,5)*Z(5,d)) / Z(5,5));
    end
end
disp('Modified bus impedance matrix after eliminating fifth row and column')
disp(Z_new)
V_4 = Z_new(4,:) * I;
printf("\n V4 = %.2f /_%.2f per unit \n\n",abs(V_4),atan(imag(V_4),real(V_4)) * 180 / %pi)
disp('V4 same as found in Example 7.6')
```

Scilab code Exa 7.8 directZbus

directZbus

```
//chapter 7
//Example 7.8
//Page 187
//directZbus
clear;clc;
```

```
//Given Impedances
Z10 = %i*1.2;
Z21 = %i*0.2;
Z23 = %i*0.15;
Z13 = %i*0.3;
Z30 = %i*1.5;
//1*1 bus
Zbus = Z10;
disp('1X1 bus impedance matrix with bus 1 and
   reference bus')
disp(Zbus)
//to establish bus 2
[m,n] = size(Zbus)
for i = 1:m
    for j = 1:n
    Zbus(2,i) = Zbus(i,j);
    Zbus(i,2) = Zbus(i,j)
    end
end
Zbus(2,2) = Z10 + Z21;
disp('After establishing bus 2')
disp(Zbus)
//to establish bus 3 with impedance connecting it to
    bus 1
[m,n] = size(Zbus)
for i = 1:m
    for j = 1
    Zbus(3,i) = Zbus(i,j);
    Zbus(i,3) = Zbus(i,j);
    end
end
Zbus(3,3) = Z10 + Z13;
disp('Connecting a impedance between bus 3 and 1')
disp(Zbus)
//to add an impedance from bus 3 to reference
[m,n] = size(Zbus)
for i = 1:m
    for j = 1:n
```

```
    Zbus(4,i) = Zbus(i,j);
    Zbus(i,4) = Zbus(i,j)
    end
end
Zbus(4,4) = Zbus(3,3) + Z30;
disp('After adding impedance from bus 3 to reference')
disp(Zbus)
[m1,n1] = size(Zbus);
Z_new = zeros(m1-1,n1-1);
for c = 1:m1-1
    for d = 1:n1-1
    Z_new(c,d) = Zbus(c,d) - ((Zbus(c,4)*Zbus(4,d)) / Zbus(4,4));
    end
end
disp('After elemination of 4th row and column')
disp(Z_new)
//to add the impedance between buses 2 and 3
Z_new(1,4) = Z_new(1,2) - Z_new(1,3);
Z_new(2,4) = Z_new(2,2) - Z_new(2,3);
Z_new(3,4) = Z_new(3,2) - Z_new(3,3);
Z_new(4,1) = Z_new(1,4);
Z_new(4,2) = Z_new(2,4);
Z_new(4,3) = Z_new(3,4);
Z_new(4,4) = Z23 + Z_new(2,2) + Z_new(3,3) - 2*Z_new(2,3);
disp('After adding impedance between buses 2 and 3')
disp(Z_new)
[m1,n1] = size(Z_new);
Zbus_new = zeros(m1-1,n1-1);
for c = 1:m1-1
    for d = 1:n1-1
    Zbus_new(c,d) = Z_new(c,d) - ((Z_new(c,4)*Z_new(4,d)) / Z_new(4,4));
    end
end
disp('The Bus Impedance Matrix is')
```

```
disp(Zbus_new)
```

Scilab code Exa 7.9 impedacedetermination

`impedacedetermination`

```
//chapter 7
//Example 7.9
//Page 190
//impedacedetermination
clear;clc;
//Given Impedances
Z10 = %i*1.2;
Z21 = %i*0.2;
Z23 = %i*0.15;
Z13 = %i*0.3;
Z30 = %i*1.5;
//Solution
Z_eq = (Z13 * (Z21+Z23) / (Z13+Z21+Z23));
Z11 = Z10 * (Z30 + Z_eq) / (Z10 + Z30 + Z_eq);
disp('Z11 is given by')
disp(Z11)
```

Chapter 8

Load Flow Solutions And Control

Scilab code Exa 8.1 NewtonRaphson

NewtonRaphson

```
//Chapter 8
//Page 200
//Example 8.1
//NewtonRaphson
clear;clc;
//Given
P = 100e6;
V = 138e3;
//From Table 8.1
R_12 = 0.042;R_15 = 0.031;R_23 = 0.031;
R_34 = 0.084;R_25 = 0.053;R_45 = 0.063;
X_12 = 0.168;X_15 = 0.126;X_23 = 0.126;
X_34 = 0.336;X_25 = 0.210;X_45 = 0.252;
//From Table 8.2
V1 = 1.04;V2 = 1;V3 = 1.02;V4 = 1;V5 = 1;
P_2 = 115e6;
//Calculation
```

```
Y_21 = - 1 / (R_12 + %i * X_12);
printf("\n Y21 = %.2f /_%.2f per unit",abs(Y_21),
    atan(imag(Y_21),real(Y_21))*180/%pi)
Y_23 = - 1 / (R_23 + %i * X_23);
printf("\n Y23 = %.2f /_%.2f per unit \n\n",abs(Y_23
    ),atan(imag(Y_23),real(Y_23))*180/%pi)
Y_21mag = abs(Y_21);Y_21ang = atan(imag(Y_21),real(
    Y_21));
Y_23mag = abs(Y_23);Y_23ang = atan(imag(Y_23),real(
    Y_23));
Y_22 = - Y_21 - Y_23;
Y_24 = 0;Y_25 = 0;
P0_2calc = (V2 * V1 * Y_21mag * cos(Y_21ang)) - (V2
    * V2 * Y_21mag * cos(Y_21ang)) - (V2 * V2 *
    Y_23mag * cos(Y_23ang)) + (V2 * V3 * Y_23mag *
    cos(Y_23ang));
P_2scheduled = - P_2 / P;
printf("\n Scheduled power into the network at bus 2
     is %.2f per unit\n",P_2scheduled)
delta_P0_2 = P_2scheduled - P0_2calc;
delP_2_3 = - V2 * V3 * Y_23mag * sin(Y_23ang);
printf("\nDifference between calculated value and
    scheduled value = %.4f per unit\n",delta_P0_2)
printf("\nThe value of the second element in the
    first row of the Jacobian = %.4f per unit \n",
    delP_2_3)
```

Scilab code Exa 8.2 Thevnin

Thevnin

```
//Chapter 8
//Page 210
```

```
//Example 8.2
//Thevnin
clear;clc;
//Given
Zth = %i * 0.2;Xg = %i * 1;
Vt = 0.97;
I = 0.8 - %i* 0.2;
Vt_b =1;
//Calculations
//a
S = Vt * I';
Eg = Vt + Xg * I;
printf("\n P = %.3f per unit \n Q = %.3f per unit\n",real(S),imag(S))
printf("\n Eg = %.2f /_%.2f per unit \n",abs(Eg),atan(imag(Eg),real(Eg))*180/%pi)
//b
Eth = Vt - Zth * I;
delta = asin(real(S) * abs(Zth) / (abs(Eth) * Vt_b));
printf("\n Eth = %.2f /_%.2f per unit \n",abs(Eth),atan(imag(Eth),real(Eth))*180/%pi)
printf("\n Phase angle of Vt = %.2f \n",delta*180/%pi)
ang = (atan(imag(Eth),real(Eth)) + delta)*180/%pi;
Vt_b1 = Vt_b * (cos(ang * %pi / 180)+ %i * sin(ang * %pi / 180));
I_b = (Vt_b1 - Eth) / Zth;
printf("\n I_b = %.2f /_%.2f per unit \n",abs(I_b),atan(imag(I_b),real(I_b))*180/%pi)
Eg_b = Vt_b1 + Xg * I_b;
S_b = Vt_b1 * I_b';
printf("\n P = %.3f per unit \n Q = %.3f per unit\n",real(S_b),imag(S_b))
printf("\n Eg = %.2f /_%.2f per unit \n",abs(Eg_b),atan(imag(Eg_b),real(Eg_b))*180/%pi)
```

Scilab code Exa 8.3 TranformerControl

TranformerControl

```
//Chapter 8
//Page 218
//Example 8.3
//TranformerControl
clear;clc;
//Given
Z = 0.8 + %i * 0.6;
V2 = 1;
Z_Ta = %i * 0.1;Z_Tb = %i * 0.1;
a=1.05;
I2 = - V2 / Z;
Y21_Ta = - 1/Z_Ta;Y22_Ta = 1/Z_Ta;
disp('For transformer Ta Y21 and Y22 in per unit is'
    )
disp(Y21_Ta);disp(Y22_Ta);
Y21_Tb = - (1/Z_Ta) / a;Y22_Tb = (1/Z_Ta) / a^2;
disp('For transformer Tb Y21 and Y22 in per unit is'
    )
disp(Y21_Tb);disp(Y22_Tb);
Y21 = Y21_Ta + Y21_Tb;Y22 = Y22_Ta + Y22_Tb;
disp('For the two transformers in parallel')
disp(Y21,'Y21 in per unit');disp(Y22,'Y22 in per
    unit');
V1 = (I2 - Y22 * V2) / Y21;
disp(V1,'V1 in per unit =')
V_1_2 = V1 - V2;
disp(V_1_2,'Difference between V1 and V2 in per unit
    ')
```

```
I_Ta = V_1_2 * Y22_Ta;
I_Tb_a1 = -I2 - I_Ta;
S_Ta = V2 * I_Ta';
S_Tb = V2 * I_Tb_a1';
disp('Complex power transmitted from the two
    transformers to the load')
disp(S_Ta,'From transformer Ta in per unit')
disp(S_Tb,'From transformer Tb in per unit')
```

Scilab code Exa 8.4 Tapchange

Tapchange

```
//Chapter 8
//Page 221
//Example 8.4
//Tapchange
clear;clc;
//Given
Z = 0.8 + %i * 0.6;
V2 = 1;
Z_Ta = %i * 0.1;Z_Tb = %i * 0.1;
Z1_Tb = %i*0.1;Z2_Tb = %i*0.1;
a=1 * (cos(3*%pi/180) + %i * sin(3*%pi/180));
I2 = - V2 / Z;
Y21_Ta = - 1/Z_Ta;Y22_Ta = 1/Z_Ta;
Y21_Tb = Y21_Ta / a';Y22_Tb = Y22_Ta / (abs(a))^2;
printf("\n Y21 = %.2f /_%.2f per unit \n",abs(Y21_Tb
    ),atan(imag(Y21_Tb),real(Y21_Tb))*180/%pi)
printf("\n Y21 = -%.2fj per unit \n",abs(Y22_Tb))
Y21 = Y21_Ta + Y21_Tb;Y22 = Y22_Ta + Y22_Tb;
disp('For the two transformers in parallel')
```

```
disp(Y21,'Y21 in per unit');disp(Y22,'Y21 in per
   unit');
V1 = (I2 - Y22 * V2) / Y21;
disp(V1,'V1 in per unit =')
V_1_2 = V1 - V2;
disp(V_1_2,'Difference between V1 and V2 in per unit
   ')
I_Ta = V_1_2 * Y22_Ta;
I_Tb_a1 = -I2 - I_Ta;
S_Ta = V2 * I_Ta';
S_Tb = V2 * I_Tb_a1';
disp('Complex power transmitted from the two
   transformers to the load')
disp(S_Ta,'From transformer Ta in per unit')
disp(S_Tb,'From transformer Tb in per unit')
V = a - V2;
I = I2/2;
I_circ = V / (Z1_Tb + Z2_Tb);
I_Ta_1 = - I - I_circ;
I_Tb_1 = -I + I_circ;
S_Ta_1 = V2 * I_Ta_1';
S_Tb_1 = V2 * I_Tb_1';
disp('Complex power transmitted from the two
   transformers to the load')
disp(S_Ta_1,'From transformer Ta in per unit')
disp(S_Tb_1,'From transformer Tb in per unit')
```

Chapter 9

Economic Operation Of Power Systems

Scilab code Exa 9.1 loaddistribution

`loaddistribution`

```
//Chapter 9
//Page 231
//Example 9.1
//loaddistribution
clear;clc;
dF_dP = [0.008 8;0.0096 6.4];
P1_min = 100;
l = [7.84 8.8 9.6 10.4 11.2 12 12.4 13];
P2_p1min = (l(1) - dF_dP(2,2)) / dF_dP(2,1);
disp('Outputs of each unit and total output for various values of incremental fuel cost')
printf("\n Plant \t Unit 1 P1 \t Unit 2 P2 \t P1+P2")
printf("\n %.2f \t %.2f \t %.2f \t %.2f",l(1),P1_min,P2_p1min,P1_min+P2_p1min)
for n = 2:8
    P1 = (l(n) - dF_dP(1,2)) / dF_dP(1,1);
```

```
    P2 = (l(n) - dF_dP(2,2)) / dF_dP(2,1);
    printf("\n %.2f \t %.2f \t %.2f \t %.2f",l(n),P1
       ,P2,P1+P2)
end
deff('[y]=mysol(P)','[y]=[P(1)+P(2)-1000;0.008*P(1)
   -0.0096*P(2)+8-6.4]');
Presult = fsolve([1,1],mysol);
printf("\n\n")
disp(Presult,'The allocation of load between units
   for the minimum cost of various total loads in MW
   (P1 followed by P2)')
l_result = dF_dP(2,1) * Presult(2) + dF_dP(2,2);
disp(l_result,'Incremental fuels cost for the above
   mentioned load is')
```

Scilab code Exa 9.2 integrate

integrate

```
//Chapter 9
//Page 234
//Example 9.2
//integrate
clear;clc;
U1 = integrate('(0.008 * P1 + 8)','P1',400,450);
U2 = integrate('(0.0096 * P2 + 6.4)','P2',450,500);
U = U1 - abs(U2);
disp(U1,'Increase in cost for unit 1 in $ per hour
   is')
disp(abs(U2),'Increase in cost for unit 2 in $ per
   hour is')
disp(U,'Net increase in cost in $ per hour is')
```

Scilab code Exa 9.3 losscoeff

losscoeff

```
//Chapter 9
//Page 236
//Example 9.3
//losscoeff
clear;clc;
I1 = 1;I2 = 0.8;
V3 = 1;pf1 =1;pf2 = pf1;pf3 = pf1;
Za = 0.04 + %i * 0.16;Ra = real(Za);
Zb = 0.03 + %i * 0.12;Rb = real(Zb);
Zc = 0.02 + %i * 0.08;Rc = real(Zc);
V1 = V3 + I1 * Za;disp(V1,'Voltage at bus 1,V1 in
    per unit')
V2 = V3 + I2 * Zb;disp(V2,'Voltage at bus 2,V2 in
    per unit')
disp('Transmission Loss Co-efficients')
B11 = (Ra + Rc) / (abs(V1) * pf1)^2;disp(B11,'B11 in
     per unit')
B12 = Rc / (abs(V1) * abs(V2) * pf1 * pf2);disp(B12,
    'B12 in per unit')
B22 = (Rb + Rc) / (abs(V2) * pf2)^2;disp(B22,'B22 in
     per unit')
```

Scilab code Exa 9.4 loss

loss

```
//Chapter 9
//Page 237
//Example 9.4
//loss
clear;clc;
I1 = 1;I2 = 0.8;
V3 = 1;pf1 =1;pf2 = pf1;pf3 = pf1;
Za = 0.04 + %i * 0.16;Ra = real(Za);
Zb = 0.03 + %i * 0.12;Rb = real(Zb);
Zc = 0.02 + %i * 0.08;Rc = real(Zc);
V1 = V3 + I1 * Za;
V2 = V3 + I2 * Zb;
B11 = (Ra + Rc) / (abs(V1) * pf1)^2;
B12 = Rc / (abs(V1) * abs(V2) * pf1 * pf2);
B22 = (Rb + Rc) / (abs(V2) * pf2)^2;
P1 = real(I1 * V1);disp(P1,'P1 in per unit')
P2 = real(I2 * V2);disp(P2,'P2 in per unit')
PL = (P1)^2 * B11 + 2 * P1 * P2 * B12 + (P2)^2 * B22
    ;disp(PL,'Loss calculated using loss coefficients
     in per unit is')
PL_I2R = I1^2 * Ra + (I1+I2)^2 * Rc + I2^2 * Rb;disp
    (PL_I2R,'Loss calculated using current and
    resistance in per unit is')
```

Scilab code Exa 9.5 generation

generation

```
//Chapter 9
//Page 240
```

```
//Example 9.5
//generation
clear;clc;
l = 12.5;
dF_dP = [ 0.01 8.5;0.015 9.5];
B22 = 0;B12 = 0;//since all the load is at plant 2
P1_trans = 200;
PL_trans = 16;
B11 = PL_trans / P1_trans^2;
printf("\n Penalty factors are \n L1 = 1 / (1 - %fP1
    ) \n L2 = 1",2*B11)
P1 = (l - dF_dP(1,2)) / (2*B11 * l + dF_dP(1,1));
P2 = (l - dF_dP(2,2)) / dF_dP(2,1);
PL = B11 * P1^2;
Pr = P1 + P2 - PL;
printf("\n Required generation from plant \n P1 = %
    .0fMW \n P2 = %.0fMW",P1,P2)
printf("\n Power loss in transmisison is %.0fMW",PL)
printf("\n The delivered load is %.0fMW",Pr)
```

Scilab code Exa 9.6 savings

savings

```
//Chapter 9
//Page 241
//Example 9.6
//savings
clear;clc;
l = 12.5;
dF_dP = [ 0.01 8.5;0.015 9.5];
B22 = 0;B12 = 0;//since all the load is at plant 2
P1_trans = 200;
```

```
PL_trans = 16;
B11 = PL_trans / P1_trans^2;
P1 = (l - dF_dP(1,2)) / (2*B11 * l + dF_dP(1,1));
P2 = (l - dF_dP(2,2)) / dF_dP(2,1);
PL = B11 * P1^2;
Pr = P1 + P2 - PL;
deff('[y]=mysol(P)','[y]=[0.01*P(1)-0.015*P(2)
   +8.5-9.5;P(1)+P(2)-0.0004*(P(1))^2-384]');
Presult=fsolve([1,1],mysol);
disp(Presult,'Values of P1 and P2 in MW')
U1 = integrate('(0.010 * P1 + 8.5)','P1',P1_trans,
   Presult(1,1));
U2 = integrate('- (0.015 * P2 + 9.5)','P2',P1_trans,
   Presult(1,2));
disp(U1,'Increase in fuel cost due to increase in
   load on plant 1 in $ per hour')
disp(U2,'Increase in fuel cost due to increase in
   load on plant 2 in $ per hour')
disp(U1 - U2,'The net savings by accounting for
   transmission loss in scheduling the received load
    of 384MW in $ per hour is')
```

Chapter 10

Symmetrical Three Phase Faults

check Appendix AP 1 for dependency:

pucalc.sci

Scilab code Exa 10.1 unloadedfault

unloadedfault

```
//Chapter 10
//Example 10.1
//Page 253
//unloadedfault
//run clear command then execute dependancy file and
    then the source file
//dependency file is pucalc.sci
clc;
//Given
P_g1 = 50e6;
V_g1 = 13.8e3;
P_g2 = 25e6;
V_g2 = 13.8e3;
```

```
P_t = 75e6;
V_t_lt = 13.8e3;
V_t_ht = 69e3;
X11_g = 0.25;
X11_t = 0.10;
Vbase = 69e3;
Pbase = 75e6;
Vbase_lt = 13.8e3;
V_ht = 66e3;
X11_d_g1 = pucalc(X11_g,V_t_lt,Vbase_lt,Pbase,P_g1);
X11_d_g2 = pucalc(X11_g,V_t_lt,Vbase_lt,Pbase,P_g2);
E_g1 = V_ht / Vbase;
E_g2 = V_ht / Vbase;
disp('For Generator 1')
printf("Xd11 = %.3f per unit \n Eg1 = %.3f per unit \n",X11_d_g1,E_g1)
disp('For Generator 2')
printf("Xd11 = %.3f per unit \n Eg2 = %.3f per unit \n",X11_d_g2,E_g2)
X_g12 = (X11_d_g1 * X11_d_g2) / (X11_d_g1 + X11_d_g2);
I11 = E_g1 / (%i*(X_g12 + X11_t));
disp(I11,'Subtransient current in the short circuit in per unit is')
Vdt = I11 * (%i*X11_t);
disp(Vdt,'Voltage on the delta side of the transformer in per unit is')
I11_g1 = (E_g1 - Vdt) / (%i*X11_d_g1);
I11_g2 = (E_g2 - Vdt) / (%i*X11_d_g2);
disp('Subtransient current in generator 1 and 2 in per unit respectively')
disp(I11_g1)
disp(I11_g2)
Ibase = Pbase / (sqrt(3) * Vbase_lt);
I11_1 = abs(I11_g1) * Ibase;
I11_2 = abs(I11_g2) * Ibase;
disp('Subtransient current in generator 1 and 2 in Amperes respectively')
```

```
disp(I11_1)
disp(I11_2)
```

Scilab code Exa 10.2 loadedfault

loadedfault

```
//Chapter 10
//Example 10.2
//Page 256
//loadedfault
clear;clc;
//Given
Pgm = 30e6;
Vgm = 13.2e3;
Xgm = 0.20;
Xl = 0.10;
Pm = 20e6;pfm = 0.8;Vt_m = 12.8e3;
Pbase = Pgm;
Vbase = Vgm;
Vf = Vt_m / Vbase;
Ibase = Pbase / (sqrt(3) * Vbase);
I_L = (Pm / (pfm * sqrt(3) * Vt_m)) * (cos(36.9 *
    %pi/180) + %i * sin(36.9 * %pi / 180)) / Ibase;
disp(I_L,'Line Current in per unit is')
Vt_g = Vf + (%i * Xl) * I_L;
E11_g = Vt_g + (%i * Xgm) * I_L;
I11_g = E11_g / (%i * (Xgm + Xl));
I11_gA = Ibase * I11_g;
disp(I11_g,'Fault current in the generator side in
    per unit')
disp(I11_gA,'Fault current in the generator side in
    A')
```

```
E11_m = Vf - (%i * Xgm) * I_L;
I11_m = E11_m / (%i * (Xgm));
I11_mA = Ibase * I11_m;
disp(I11_m,'Fault current in the motor side in per
    unit')
disp(I11_mA,'Fault current in the motor side in A')
If = I11_g + I11_m;
disp(If,'Toatl Fault current in per unit')
disp(If * Ibase,'Total Fault current in A')
```

Scilab code Exa 10.3 thevninloadedfault

thevninloadedfault

```
//Chapter 10
//Example 10.3
//Page 259
//thevninloadedfault
clear;clc;
//Given
Pgm = 30e6;
Vgm = 13.2e3;
Xgm = 0.20;
Xl = 0.10;
Pm = 20e6;pfm = 0.8;Vt_m = 12.8e3;
Pbase = Pgm;
Vbase = Vgm;
Vf = Vt_m / Vbase;
Ibase = Pbase / (sqrt(3) * Vbase);
I_L = (Pm / (pfm * sqrt(3) * Vt_m)) * (cos(36.9 *
    %pi/180) + %i * sin(36.9 * %pi / 180)) / Ibase;
Zth = (%i*(Xgm+Xl) * (%i * Xgm)) / (%i*(Xgm+Xl) + (
    %i * Xgm));
```

```
disp(Zth,'Zth in per unit')
I11_f = Vf / Zth;
disp(I11_f,'Subtransient fault current in per unit')
If_g = I11_f * (%i * Xgm) / (%i*(Xgm+Xl) + (%i * Xgm
   ));
If_m = I11_f * (%i * (Xgm + Xl)) / (%i*(Xgm+Xl) + (
   %i * Xgm));
I11_g = I_L + If_g ;
disp(I11_g,'Subtransient fault current in generator
   side in per unit')
I11_m = If_m - I_L ;
disp(I11_m,'Subtransient fault current in motor side
    in per unit')
```

Scilab code Exa 10.4 faultthrubus

faultthrubus

```
//Chapter 10
//Example 10.4
//Page 265
//faultthrubus
clear;clc;
//Given
Pg1 = 270e6;
Pg3 = 225e6;
Pbase = 100e6;
V = 1;
X = 0.3;
Xg_b1 = X * Pbase /Pg1;
Xg_b3 = X * Pbase /Pg3;
y10 = 1 / (%i * Xg_b1);
y30 = 1 / (%i * Xg_b3);
```

```
y12 = 1 / (%i * 0.168);y15 = 1 / (%i * 0.126);
y23 = 1 / (%i * 0.126);y34 = 1 / (%i * 0.336);
y35 = 1 / (%i * 0.210);y45 = 1 / (%i * 0.252);
//Ybus by inspection
Ybus = zeros(5,5);
Ybus(1,1) = y10 + y12 + y15;
Ybus(2,2) = y12 + y23;
Ybus(3,3) = y30 + y23 + y35 + y34;
Ybus(4,4) = y34 + y45;
Ybus(5,5) = y45 + y15 + y35;
Ybus(1,2) = -y12;Ybus(2,1) = Ybus(1,2);Ybus(1,3) =
    0;Ybus(1,4) =0;
Ybus(2,3) = -y23;Ybus(3,2E) = Ybus(2,3);Ybus(2,5) =
    0;Ybus(2,4) =0;
Ybus(3,4) = -y34;Ybus(4,3) = Ybus(3,4);Ybus(3,1) =
    0;
Ybus(4,5) = -y45;Ybus(5,4) = Ybus(4,5);Ybus(4,1) =
    0;Ybus(4,2) = 0;
Ybus(1,5) = -y15;Ybus(5,1) = Ybus(1,5);Ybus(5,2) =
    0;
Ybus(3,5) = -y35;Ybus(5,3) = Ybus(3,5);
disp(Ybus,'Ybus')
Zbus = inv(Ybus);
disp(Zbus,'Zbus')
I11 = V / Zbus(4,4);
disp(I11,'The subtransient current in a three-phase
    fault on bus 4 in per unit is')
V3 = V - I11 * Zbus(3,4);
V5 = V - I11 * Zbus(5,4);
disp(V3,V5,'Voltages at bus 3 and 5 repectively in
    per unit')
I3 = V3 * y34;
I5 = V5 * y45;
disp(I3,'Currents to fault from bus 3 in per unit')
disp(I5,'Currents to fault from bus 5 in per unit')
disp(I3 + I5,'Total current to fault in per unit')
```

Scilab code Exa 10.5 breakerrating

`breakerrating`

```
//Chapter 10
//Example 10.5
//Page 268
//breakerrating
clear;clc;
//Given
Pg = 25e6;Vg = 13.8e3;X11_dg = 0.15;
X11_dm = 0.20;Pmbase = 5e6;Vbasem = 6.9e3;
P_tr = 25e6;V_ht = 13.8e3;V_lt = 6.9e3;X_tr = 0.10;
Vbus_m = 6.9e3;
//the subtransient current in the fault
X11_dm1 = X11_dm * (P_tr / Pmbase);
Vf = 1;Xeqm =  1/(4*X11_dm1);Xeq_trg = X11_dg + X_tr
    ;
Zth = Xeqm * Xeq_trg / (Xeqm + Xeq_trg);
I11_f = Vf / (%i * Zth);
Ibase_vbus = P_tr / (sqrt(3) * Vbus_m);
I11_f_a = abs(I11_f) * Ibase_vbus;
printf("\n Fault current in per unit = -j%.0f \n",
    abs(I11_f))
printf("\n Fault current in amperes = %.0f \n",
    I11_f_a)
//the subtransient current in breaker A
Ig_f = I11_f * Xeq_trg / (Xeqm + Xeq_trg);
Im_f = 0.25 * (I11_f - Ig_f);
I11_pu = Ig_f + 3 * Im_f;
I11_a = I11_pu * Ibase_vbus;
printf("\n Subtransient current through breaker A in
     per unit is -j%.0f \n",abs(I11_pu))
```

```
printf("\n Subtransient current through breaker A in amperes is %.0f \n",abs(I11_a))
//Symmetrical short-circuit interrupting current in the fault and in breaker A
X11_dm2 = 1.5 * X11_dm1;
Xeqm1 = X11_dm2 / 4;
Zth_c = (Xeqm1 * Xeq_trg) / (Xeqm1 + Xeq_trg);
Ig_f1 = Vf * Xeqm1 / (Zth_c * (Xeqm1 + Xeq_trg));
Im_f1 = (Vf * Xeq_trg) / (4 * Zth_c * (Xeqm1 +
    Xeq_trg));
I11_1pu = Ig_f1 + 3 * Im_f1;
I11_1a = I11_1pu * Ibase_vbus;
disp(I11_1a,'Symmetrical Short circuit current to be interrupted in A')
I11_pu_cb = Ig_f1 + 4 * Im_f1;
I11_a_cb = I11_pu_cb * Ibase_vbus;
disp(I11_a_cb,'The short circuit current rating of breakers must be atleast(in amperes)')
Vcb = 15.5e3;I_ic = 8900;k = 2.67;
Iic = I_ic * k;
Icb = Vcb * I_ic / Vbus_m;
printf("\n The required capability of %.0f A is well below 80 percent of %.0f A and the breaker is suitable with respect to the short-circuit current\n",abs(I11_a_cb),abs(Icb))
```

Chapter 11

Symmetrical Components

Scilab code Exa 11.1 linecurrents

`linecurrents`

```
//Chapter 11
//Example 11.1
//Page 280
//linecurrents
clear;clc;

//Given
I_a = 10 * (cos(0) + %i * sin(0));
I_b = 10 * (cos(180 * %pi / 180) + %i * sin(180 *
    %pi / 180));
I_c = 0;
a = 1 * (cos(120 * %pi / 180) + %i * sin(120 * %pi /
    180));
//Phase 'a'
disp('Phase a')
I_a0=(1/3)*(I_a+I_b+I_c);
I_a1=(1/3)*(I_a+a*I_b+a^2*I_c);
I_a2=(1/3)*(I_a+a^2*I_b+a*I_c);
```

```
printf(" I_a0 = %.2f /_%.2f A",abs(I_a0),atan(imag(
    I_a0),real(I_a0)) * 180 / %pi)
printf("\n I_a1 = %.2f /_%.2f A",abs(I_a1),atan(imag
    (I_a1),real(I_a1)) * 180 / %pi)
printf("\n I_a2 = %.2f /_%.2f A \n\n",abs(I_a2),atan
    (imag(I_a2),real(I_a2)) * 180 / %pi)

//Phase 'b'
disp('Phase b')
I_b0 = I_a0;
I_b1=a^2*I_a1;
I_b2=a*I_a2;
printf(" I_b0 = %.2f /_%.2f A",abs(I_b0),atan(imag(
    I_b0),real(I_b0)) * 180 / %pi)
printf("\n I_b1 = %.2f /_%.2f A",abs(I_b1),atan(imag
    (I_b1),real(I_b1)) * 180 / %pi)
printf("\n I_b2 = %.2f /_%.2f A \n\n",abs(I_b2),atan
    (imag(I_b2),real(I_b2)) * 180 / %pi)

//Phase 'c'
disp('Phase c')
I_c0=I_a0;
I_c1=a*I_a1;
I_c2=a^2*I_a2;
printf(" I_c0 = %.2f /_%.2f A",abs(I_c0),atan(imag(
    I_c0),real(I_c0)) * 180 / %pi)
printf("\n I_c1 = %.2f /_%.2f A",abs(I_c1),atan(imag
    (I_c1),real(I_c1)) * 180 / %pi)
printf("\n I_c2 = %.2f /_%.2f A \n\n",abs(I_c2),atan
    (imag(I_c2),real(I_c2)) * 180 / %pi)
```

Scilab code Exa 11.2 sequence

`sequence`

```
//Chapter 11
//Example 11.2
//Page 285
//sequence
clear;clc;

//Give
V_ab = 0.8 * (cos(82.8 * %pi /180) + %i * sin(82.8 *
    %pi / 180));
V_bc = 1.2 * (cos(-41.4 * %pi /180) + %i * sin(-41.4
    * %pi / 180));
V_ca = 1 * (cos(180 * %pi /180) + %i * sin(180 * %pi
    / 180));
a = 1 * (cos(120 * %pi / 180) + %i * sin(120 * %pi /
    180));

//Symmetrical components of line voltage
//Since neutral connection is absent zero sequence
   components are absent
V_ab1=(1/3)*(V_ab+a*V_bc+a^2*V_ca);
V_ab2=(1/3)*(V_ab+a^2*V_bc+a*V_ca);

V_a1 = V_ab1 * (cos(-30 * %pi / 180) + %i * sin(-30
   * %pi / 180));
V_a2 = V_ab2 * (cos(30 * %pi / 180) + %i * sin(30 *
   %pi / 180));

r = 1 * (cos(0) + %i * sin(0));

I_a1 = V_a1 / r;
I_a2 = V_a2 / r;

V_A1 = -1 * %i * V_a1 ;
V_A2 = %i * V_a2 ;
V_A = V_A1 + V_A2;

V_B1 = a^2 * V_A1;
V_B2 = a * V_A2;
```

```
V_B = V_B1 + V_B2;

V_C1 = a * V_A1;
V_C2 = a^2 * V_A2;
V_C = V_C1 + V_C2;

V_AB = V_A - V_B;
V_BC = V_B - V_C;
V_CA = V_C - V_A;

I_A = V_A / r;
I_B = V_B / r;
I_C = V_C / r;

disp('Line-neutral voltages')
printf("\n V_AB = %.2f /_%.2f per unit",abs(V_AB),atan(imag(V_AB),real(V_AB))*180/%pi)
printf("\n V_BC = %.2f /_%.2f per unit",abs(V_BC),atan(imag(V_BC),real(V_BC))*180/%pi)
printf("\n V_CA = %.2f /_%.2f per unit \n\n",abs(V_CA),atan(imag(V_CA),real(V_CA))*180/%pi)

disp('Line-line voltages')
printf("\n V_AB = %.2f /_%.2f per unit",abs(V_AB)/sqrt(3),atan(imag(V_AB),real(V_AB))*180/%pi)
printf("\n V_BC = %.2f /_%.2f per unit",abs(V_BC)/sqrt(3),atan(imag(V_BC),real(V_BC))*180/%pi)
printf("\n V_CA = %.2f /_%.2f per unit \n\n",abs(V_CA)/sqrt(3),atan(imag(V_CA),real(V_CA))*180/%pi)

disp('Line currents')
printf("\n I_A = %.2f /_%.2f per unit",abs(I_A),atan(imag(I_A),real(I_A))*180/%pi)
printf("\n I_B = %.2f /_%.2f per unit",abs(I_B),atan(imag(I_B),real(I_B))*180/%pi)
printf("\n I_C = %.2f /_%.2f per unit \n\n",abs(I_C),atan(imag(I_C),real(I_C))*180/%pi)
```

This code can be downloaded from the website wwww.scilab.in check Appendix AP 1 for dependency:

`pucalc.sci`

Scilab code Exa 11.4 zerosequence

`zerosequence`

```
//Chapter 11
//Example 11.4
//Page 301
//zerosequence
//run clear command then execute dependancy file and
    then the source file
//dependency file is pucalc.sci
clc;

//Given
P_g = 300e6;
V_g = 20e3;
X11_g = 0.20;
l = 64;
V_m = 13.2e3;
P_m1 = 200e6;
P_m2 = 100e6;
X11_m = 0.20;
T1_P = 350e6;
T1_vht = 230e3;
T1_vlt = 20e3;
```

```
x_T1 = 0.10;
T2_1_P = 100e6;
T2_1_vht = 127e3;
T2_1_vlt = 13.2e3;
x_T2 = 0.10;
x_line = 0.5;//ohm per km
V_base = V_g;
P_base = P_g;
x0 = 0.05;
x_cl = 0.4;
x0_line = 1.5;//ohm per km

//Calculations
T2_P = 3*T2_1_P;
T2_vht = sqrt(3)*T2_1_vht;
T2_vlt = T2_1_vlt;
V_base_line = (T1_vht/T1_vlt)*V_base;
V_base_m = V_base_line * (T2_vlt/T2_vht);
z_line_base = (V_base_line)^2/P_base;
x_line_pu = x_line * l / z_line_base;
x0_g = x0;
x0_m1 = pucalc(x0,V_m,V_base_m,P_base,P_m1);
x0_m2 = pucalc(x0,V_m,V_base_m,P_base,P_m2);
Z_g = (V_g^2) / P_base;
Z_m = (V_base_m)^2 / P_base;
Zn_g = 3 * x_cl / Z_g;
Zn_m = 3 * x_cl / Z_m;
X_0 = x0_line * l / z_line_base;
printf("\n\n Generator X0 = %.2f per unit",x0_g)
printf("\n\n Motor 1 X0 = %.4f per unit",x0_m1)
printf("\n\n Motor 2 X0 = %.4f per unit",x0_m2)
printf("\n\n Generator base impedance = %.3f per
   unit",Z_g)
printf("\n\n Motor base impedance = %.3f per unit",
   Z_m)
printf("\n\n In generator 3Zn = %.3f per unit",Zn_g)
printf("\n\n In motor 3Zn = %.3f per unit",Zn_m)
printf("\n\n Transmission line X0 = %.4f per unit",
```

```
    X_0)
//zero-sequence diagram is shown in xcos file
```

This code can be downloaded from the website wwww.scilab.in

Chapter 12

Unsymmetrical Faults

Scilab code Exa 12.1 1phasetogroundfault

1phasetogroundfault

```
//Chapter 12
//Page 308
//Example 12.1
//1phasetogroundfault
clear;clc;
//Given
P = 20e6;
V = 13.8e3;
P_b = 20e6;
V_b = 13.8e3;
Z1 = %i * 0.25;
Z2 = %i * 0.35;
Z0 = %i * 0.10;
a = 1 * (cos(120 * %pi / 180) + %i * sin(120 * %pi /
    180));
//Calculations
Ea = V / V_b;
Ia1 = Ea / (Z0 + Z1 + Z2);
Ia2 = Ia1;Ia0 = Ia1;
```

```
Ia = 3 * Ia1;
I_b = P / (sqrt(3) * V);
Ia_1 = Ia * I_b;
printf("\n Base Current = %f A",I_b)
printf("\n Subtransient current in line a = -j%.0f A \n\n",abs(imag(Ia_1)))
//Symmetrical Components of voltage from point a to ground
Va1 = Ea - Ia1 * Z1;
Va2 = -Ia2 * Z2;
Va0 = -Ia0 * Z0;
disp('Symmetrical Components of voltage from point a to ground')
printf("\n Va1 = %.2f per unit",Va1)
printf("\n Va2 = %.2f per unit",Va2)
printf("\n Va0 = %.2f per unit \n\n",Va0)
//Line to ground voltages
Va = Va0 + Va1 + Va2;
Vb = Va0 + Va1 * a^2 + Va2 * a;
Vc = Va0 + Va2 * a^2 + Va1 * a;
disp('Line to ground voltages')
printf("\n Va = %.2f /_%.2f per unit",abs(Va),atan(imag(Va),real(Va))*180/%pi)
printf("\n Vb = %.2f /_%.2f per unit",abs(Vb),atan(imag(Vb),real(Vb))*180/%pi)
printf("\n Vc = %.2f /_%.2f per unit \n\n",abs(Vc),atan(imag(Vc),real(Vc))*180/%pi)
//Line to line voltages in per-unit are
Vab = Va - Vb;
Vbc = Vb - Vc;
Vca = Vc - Va;
disp('Line to line voltages in per-unit are')
printf("\n Vab = %.2f /_%.2f per unit",abs(Vab),atan(imag(Vab),real(Vab))*180/%pi)
printf("\n Vbc = %.2f /_%.2f per unit",abs(Vbc),atan(imag(Vbc),real(Vbc))*180/%pi)
printf("\n Vca = %.2f /_%.2f per unit \n\n",abs(Vca),atan(imag(Vca),real(Vca))*180/%pi)
```

```
//Line to line voltages in volts
Vab_1 = Vab * V / sqrt(3);
Vbc_1 = Vbc * V / sqrt(3);
Vca_1 = Vca * V / sqrt(3);
disp('Line to line voltages in volts')
printf("\n Vab = %.2f /_%.2f kV",abs(Vab_1)/1e3,atan(imag(Vab_1),real(Vab_1))*180/%pi)
printf("\n Vbc = %.2f /_%.2f kV",abs(Vbc_1)/1e3,atan(imag(Vbc_1),real(Vbc_1))*180/%pi)
printf("\n Vca = %.2f /_%.2f kV \n\n",abs(Vca_1)/1e3,atan(imag(Vca_1),real(Vca_1))*180/%pi)
```

Scilab code Exa 12.2 linetolinefault

`linetolinefault`

```
//Chapter 12
//Page 311
//Example 12.2
//linetolinefault
clear;clc;
//Given
P = 20e6;
V = 13.8e3;
P_b = 20e6;
V_b = 13.8e3;
Z1 = %i * 0.25;
Z2 = %i * 0.35;
Z0 = %i * 0.10;
a = 1 * (cos(120 * %pi / 180) + %i * sin(120 * %pi / 180));
//Calculations
Ea = V / V_b;
```

```
Ia1 = Ea / (Z1 + Z2);
Ia2 = - Ia1;Ia0 = 0;
Ia = Ia1 + Ia2 + Ia0;
Ib = a^2 * Ia1 + a*Ia2 + Ia0;
Ic = -Ib;
I_b = P / (sqrt(3) * V);
printf("\n Base Current = %f A",I_b)
Ia_1 = Ia * I_b;
Ib_1 = Ib * I_b;
Ic_1 = Ic * I_b;
printf("\n Subtransient current in line a = %.0f A",Ia_1)
printf("\n Subtransient current in line b =  %.2f /_%.2f A",abs(Ib_1),atan(imag(Ib_1),real(Ib_1))*180 / %pi)
printf("\n Subtransient current in line c =  %.2f /_%.2f A \n\n",abs(Ic_1),atan(imag(Ic_1),real(Ic_1))*180 / %pi)
////Symmetrical Components of voltage from point a to ground
Va1 = Ea - Ia1 * Z1;
Va2 = Va1;
Va0 = 0;
disp('Symmetrical Components of voltage from point a to ground')
printf("Va1 = %.2f per unit",Va1)
printf("\n Va2 = %.2f per unit",Va2)
printf("\n Va0 = %.2f per unit \n\n",Va0)
//Line to ground voltages
Va = Va0 + Va1 + Va2;
Vb = Va0 + Va1 * a^2 + Va2 * a;
Vc = Vb;
disp('Line to ground voltages')
printf("Va = %.2f /_%.2f per unit",abs(Va),atan(imag(Va),real(Va))*180/%pi)
printf("\n Vb = Vc = %.2f per unit \n\n",Vb)
//Line to line voltages in per-unit are
Vab = Va - Vb;
```

```
Vbc = Vb - Vc;
Vca = Vc - Va;
disp('Line to line voltages in per-unit are')
printf("Vab = %.2f /_%.2f per unit",abs(Vab),atan(
    imag(Vab),real(Vab))*180/%pi)
printf("\n Vbc = %.2f per unit",Vbc)
printf("\n Vca = %.2f /_%.2f per unit \n\n",abs(Vca)
    ,atan(imag(Vca),real(Vca))*180/%pi)
//Line to line voltages in volts
Vab_1 = Vab * V / sqrt(3);
Vbc_1 = Vbc * V / sqrt(3);
Vca_1 = Vca * V / sqrt(3);
disp('Line to line voltages in volts')
printf("Vab = %.2f /_%.2f kV",abs(Vab_1)/1e3,atan(
    imag(Vab_1),real(Vab_1))*180/%pi)
printf("\n Vbc = %.2f  kV",Vbc_1)
printf("\n Vca = %.2f /_%.2f kV \n\n",abs(Vca_1)/1e3
    ,atan(imag(Vca_1),real(Vca_1))*180/%pi)
```

Scilab code Exa 12.3 linetogroundfaultunloadedG

linetogroundfaultunloadedG

```
//Chapter 12
//Page 314
//Example 12.3
//linetogroundfaultunloadedG
clear;clc;
//Given
P = 20e6;
V = 13.8e3;
P_b = 20e6;
V_b = 13.8e3;
```

```
Z1 = %i * 0.25;
Z2 = %i * 0.35;
Z0 = %i * 0.10;
a = 1 * (cos(120 * %pi / 180) + %i * sin(120 * %pi /
    180));
//Calculations
Ea = V / V_b;
Ia1 = Ea / (Z1 + (Z2*Z0)/(Z2+Z0));
I_b = P / (sqrt(3) * V);
Va1 = Ea - Ia1 * Z1;
Va2 = Va1;Va0 = Va1;
Ia2 = - Va2 / Z2;
Ia0 = - Va0 / Z0;
Ia = Ia1 + Ia2 + Ia0;
Ib = a^2*Ia1 + a*Ia2 + Ia0;
Ic = a*Ia1 + a^2*Ia2 + Ia0;
In = 3 * Ia0;
Va = Va1 + Va2 + Va0;
Vb = 0;
Vc = 0;
disp('Line to ground voltages')
printf("Va = %.2f /_%.2f per unit",abs(Va),atan(imag
   (Va),real(Va))*180/%pi)
printf("\n Vb = Vc = %.0f per unit \n\n",Vb)
Vab = Va - Vb;
Vbc = Vb - Vc;
Vca = Vc - Va;
disp('Line to line voltages in per-unit are')
printf("Vab = %.2f per unit",Vab)
printf("\n Vbc = %.2f per unit",Vbc)
printf("\n Vca = %.2f per unit \n\n",Vca)
I_a1 = I_b * Ia;
I_b1 = I_b * Ib;
I_c1 = I_b * Ic;
I_n1 = I_b * In;
printf("\n Base Current = %f A",I_b)
printf("\n Subtransient current in line a = %.0f A",
   I_a1)
```

```
printf("\n Subtransient current in line b = %.0f /_%.2f A",abs(I_b1),atan(imag(I_b1),real(I_b1)) * 180 / %pi)
printf("\n Subtransient current in line c = %.0f /_%.2f A",abs(I_c1),atan(imag(I_c1),real(I_c1)) * 180 / %pi)
printf("\n Subtransient current in neutral = %.0f /_%.2f A \n\n",abs(I_n1),atan(imag(I_n1),real(I_n1)) * 180 / %pi)
Vab_1 = Vab * V / sqrt(3);
Vbc_1 = Vbc * V / sqrt(3);
Vca_1 = Vca * V / sqrt(3);
disp('Line to line voltages in volts')
printf("Vab = %.2f /_%.2f kV",abs(Vab_1)/1e3,atan(imag(Vab_1),real(Vab_1))*180/%pi)
printf("\n Vbc = %.2f  kV",Vbc_1)
printf("\n Vca = %.2f /_%.2f kV \n\n",abs(Vca_1)/1e3,atan(imag(Vca_1),real(Vca_1))*180/%pi)
```

Scilab code Exa 12.4 interconnected

interconnected

```
//Chapter 12
//Page 321
//Example 12.4
//interconnected
clear;clc;
//Given
V_bus1 = 4.16e3;
V_bus_2 = 600;
Vm = 600;
n_m = 0.895;
```

```
Pop_m = 6000;
X11_m = 0.2;X_2_m = 0.20;X_0_m = 0.04;X_n_m= 0.02;
Vtr_ht = sqrt(3) * 2400;Vtr_lt = 600;Ptr =3 * 2500e3;
X11_tr = 0.10;
Pg = 7500e3;Vg = 4.16e3;
X11_g = 0.10;X_2_g = 0.10;X_0_g = 0.05;X_n_g = 0.05;
//At the time of fault
Pload = 5000;pf_load = 0.85;n_load = 0.88;
Vbase_sysbus = Vg;Pbase_sysbus = Pg;
Vbase_m = Vtr_lt;Pbase_m = Ptr;
Pin_m =(Pop_m * 0.746) * 1e3/ n_m;
printf("\n Input Rating of the single equivalent motor = %.0f kVA \n",Pin_m)
X11_m_new = X11_m * Pbase_m / Pin_m;
X_2_m_new = X_2_m * Pbase_m / Pin_m;
X_0_m_new = X_0_m * Pbase_m / Pin_m;
X_n_m_new = 3 * X_n_m * Pbase_m / Pin_m;
disp('For Motor')
printf("\nX11 = %.1f per unit\n X_2 = %.1f per unit\n X_0 = %0.2f per unit\n 3X_n = %.2f per unit\n",X11_m_new,X_2_m_new,X_0_m_new,X_n_m_new)
printf("\n The equivalent generator reactance from neutral to ground in the zero-sequence network = %.2f per unit\n",3*X_0_g)
Vf = 1 * (cos(0) + %i * sin(0));
Ibase_m = Pbase_m / (sqrt(3) * Vbase_m);
printf("\n Base current in motor circuit = %.0f \n\n",Ibase_m)
Iactual_m = 746 * Pload / (n_load * sqrt(3) * Vbase_m * pf_load);
magIa = Iactual_m / Ibase_m;
angleIa = - acos(0.85);
Ia_prefault = magIa * (cos(angleIa) + %i * sin(angleIa));
printf("\n Prefault current through line a = %.3f -j%.3f per unit\n\n",real(Ia_prefault),abs(imag(Ia_prefault)))
```

```
Eg_11 = 1;Em_11 = 1;
Z1 = ((%i * X11_g + %i * X_2_g) * (%i * X11_m_new)) / (%i * (X11_g + X_2_g + X11_m_new));
Z2 = Z1;Z0 = 3 * %i * X_0_g;
printf("\n\n Z1 = j%.2f per unit\n Z2 = j%.2f per unit\n Z0 = j%.2f per unit\n",abs(Z1),abs(Z2),abs(Z0))
Ia1 = Vf / (Z1 + Z2 + Z0);
Ia2 = Ia1;Ia0 = Ia1;
Ia_fault = 3 * Ia0;
printf("\n Current Ia in fault = -j%.3f per unit \n",abs(Ia_fault))
Ia1_tr = Ia1 * (%i * X11_m_new) / (%i * X11_m_new + %i * X11_g + %i * X_2_g);
Ia1_m = Ia1 * (%i * X11_g + %i * X_2_g ) / (%i * X11_m_new + %i * X11_g + %i * X_2_g);
a = 1 * (cos(120 * %pi / 180) + %i * sin(120 * %pi / 180));
A = [ 1 1 1; 1 a^2 a ; 1 a a^2];
Ia_tr = [ 0 ;Ia1_tr ;Ia1_tr];
I_tr = A * Ia_tr;
disp('Currents in the line at the fault from the transformer in the order Ia,Ib,Ic in per unit are')
disp(I_tr)
disp('Currents in the line at the fault from the transformer in the order Ia,Ib,Ic in A are')
disp(abs(I_tr) * Ibase_m)
Ia_m = [Ia1 ; Ia1_m ; Ia1_m];
I_m = A * Ia_m;
disp('Currents in the line at the fault from the motor in the order Ia,Ib,Ic in per unit are')
disp(I_m)
disp('Currents in the line at the fault from the motor in the order Ia,Ib,Ic in A are')
disp(abs(I_m) * Ibase_m)
I_A1 = -%i * Ia1_tr;I_A2 = %i * Ia1_tr;I_a0 = 0;
I_A = I_A1 + I_A2;
```

```
I_B1 = a^2 * I_A1;I_B2 = a * I_A2;
I_B = I_B1 + I_B2;
I_C1 = a * I_A1;I_C2 = a^2 * I_A2;
I_C = I_C1 + I_C2;
disp('Per Units currents in the order I_A,I_B,I_C in
    per unit are')
disp(I_A);disp(I_B);disp(I_C);
Ibase_ht = Ptr / (sqrt(3) * Vtr_ht);
disp('Per Units currents in the order I_A,I_B,I_C in
    A are')
disp(abs(I_A) * Ibase_ht);disp(abs(I_B) * Ibase_ht);
    disp(abs(I_C) * Ibase_ht);
disp('Under loaded conditions')
disp('Current from transformer to the fault phase a'
    )
disp(Ia_prefault + Ia1_tr)
disp('Current from motor to the fault phase a')
disp(- Ia_prefault + Ia1_m)
```

This code can be downloaded from the website wwww.scilab.in

Scilab code Exa 12.5 busimpedancematrix

busimpedancematrix

```
//Chapter 12
//Page 329
//Example 12.5
//busimpedancematrix
clear;clc;
//Given
V_bus1 = 4.16e3;
```

```
V_bus_2 = 600;
Vm = 600;
n_m = 0.895;
Pop_m = 6000;
X11_m = %i * 0.2;X_2_m = %i * 0.20;X_0_m = %i * 0.04;X_n_m= %i * 0.02;
Vtr_ht = sqrt(3) * 2400;Vtr_lt = 600;Ptr =3 * 2500e3;
X11_tr = %i * 0.10;
Pg = 7500e3;Vg = 4.16e3;
X11_g = %i * 0.10;X_2_g = %i * 0.10;X_0_g = %i * 0.05;X_n_g = %i * 0.05;
//At the time of fault
Pload = 5000;pf_load = 0.85;n_load = 0.88;
Vbase_sysbus = Vg;Pbase_sysbus = Pg;
Vbase_m = Vtr_lt;Pbase_m = Ptr;
Pin_m =(Pop_m * 0.746) * 1e3/ n_m;
X11_m_new = X11_m * Pbase_m / Pin_m;
X_2_m_new = X_2_m * Pbase_m / Pin_m;
X_0_m_new = X_0_m * Pbase_m / Pin_m;
X_n_m_new = 3 * X_n_m * Pbase_m / Pin_m;
X_n_g_new = 3 * X_n_g;
Y1 = zeros(2,2);Y2 = zeros(2,2);Y0 = zeros(2,2);
Y1(1,1) = 1/X11_g + 1/X11_m_new;Y2(1,1) = Y1(1,1);
Y1(1,2) = - 1 / X11_g;Y2(1,2) = Y1(1,2);
Y1(2,2) = 1/X11_g + 1/X_2_g;Y2(2,2) = Y1(2,2)
Y1(2,1) = Y1(1,2);Y2(2,1) = Y2(1,2);
Y0(1,1) = 1 / X_n_g_new;
Y0(2,2) = 1/X11_m + 1/X_2_g;
disp('Y_bus1 = Y_bus2 = ')
disp(Y2)
disp('Y_bus0 = ')
disp(Y0)
Z1 = inv(Y1);Z2 = inv(Y2);Z0 = inv(Y0);
disp('Z_bus1 = Z_bus2 = ')
disp(Z1)
disp('Z_bus0 = ')
disp(Z0)
```

```
Vf = 1 * (cos(0) + %i * sin(0));
If_bus1 = 3 * Vf / (Z1(1,1) + Z2(1,1) + Z0(1,1));
disp(If_bus1,'The current in fault on bus 1 in per
    unit is')
If_bus2 = 3 * Vf / (Z1(2,2) + Z2(2,2) + Z0(2,2));
disp(If_bus2,'The current in fault on bus 2 in per
    unit is')
Ia_1 =If_bus1 / 3;Ia_2 = Ia_1;Ia_0 = Ia_1;
Va1 = Vf - (Z1(2,1) * Ia_1);
Va2 = - Ia_2 * Z2(2,1);
Va0 = - Ia_0 * Z0(2,1);
disp('Sequence components of phase A in the order
    Va1,Va2,Va0 in per unit are')
disp(Va1);disp(Va2);disp(Va0)
VA1 = -%i *Va1;
VA2 = %i *Va2;
VA = VA1 + VA2;
a = 1 * (cos(120 * %pi / 180) + %i * sin(120 * %pi /
     180));
VB = a^2 * VA1 + a * VA2;
VC = a * VA1 + a^2 * VA2;
disp('Currents in phases in the order VA,VB,VC in
    per unit are')
disp(VA);disp(VB);disp(VC)
```

Chapter 13

System Protection

Scilab code Exa 13.1 Zones

`Zones`

```
//Chapter 13
//Example 13.1
//Page 341
//Zones
clear;clc;
disp('Solution to this problem can be got by theory
   from Section 13.2 in the textbook')
```

Scilab code Exa 13.2 MaxMinI

`MaxMinI`

```
//Chapter 13
//Page 355
```

```
//Example 13.2
//MaxMinI
clear;clc;
//Given
V = 13.8e3;
Z_tr = %i * 5;
Z_tr_eq = Z_tr / 2;//since two reactances of equal value are in parallel
Z1 = %i*9.6;Z2 = %i*6.4;Z3 = %i*8.0;Z4 = %i*12.8;
m = sqrt(3) / 2;//to obtain line-to-line fault from a three-phase fault current
//At bus 5
//Max. Current
If_b5_max = (V/sqrt(3)) / (Z_tr_eq + Z1 + Z2 + Z3 + Z4);
disp(If_b5_max,'Maximum fault current at bus 5 in A')
//Min. Current
If_b5_min = (V/sqrt(3)) * m / (Z_tr + Z1 + Z2 + Z3 + Z4);
disp(If_b5_min,'Minimum fault current at bus 5 in A')
//At bus 4
//Max. Current
If_b4_max = (V/sqrt(3)) / (Z_tr_eq + Z1 + Z2 + Z3);
disp(If_b4_max,'Maximum fault current at bus 4 in A')
//Min. Current
If_b4_min = (V/sqrt(3)) * m / (Z_tr + Z1 + Z2 + Z3);
disp(If_b4_min,'Minimum fault current at bus 4 in A')
//At bus 3
//Max. Current
If_b3_max = (V/sqrt(3)) / (Z_tr_eq + Z1 + Z2);
disp(If_b3_max,'Maximum fault current at bus 3 in A')
//Min. Current
If_b3_min = (V/sqrt(3)) * m / (Z_tr + Z1 + Z2);
```

```
disp(If_b3_min,'Minimum fault current at bus 3 in A')
//At bus 2
//Max. Current
If_b2_max = (V/sqrt(3)) / (Z_tr_eq + Z1);
disp(If_b2_max,'Maximum fault current at bus 2 in A')
//Min. Current
If_b2_min = (V/sqrt(3)) * m / (Z_tr + Z1);
disp(If_b2_min,'Minimum fault current at bus 2 in A')
//At bus 2
//Max. Current
If_b1_max = (V/sqrt(3)) / (Z_tr_eq);
disp(If_b1_max,'Maximum fault current at bus 1 in A')
//Min. Current
If_b1_min = (V/sqrt(3)) * m / (Z_tr);
disp(If_b1_min,'Minimum fault current at bus 1 in A')
```

Scilab code Exa 13.3 selection

selection

```
//Chapter 13
//Page 357
//Example 13.3
//selection
//This problem contains many assumptions and values are taken from Figure 13.7 in page 348 after intial calculations,it is done in order to select equipment of the available rated value in the
```

```
    market to meet the required conditions.So only
    the required calculations are shown and final
    answer after the required changes are displayed.
clear;clc;
//Given
V = 13.8e3;
Z_tr = %i * 5;
Z_tr_eq = Z_tr / 2;//since two reactances of equal
    value are in parallel
Z1 = %i*9.6;Z2 = %i*6.4;Z3 = %i*8.0;Z4 = %i*12.8;
m = sqrt(3) / 2;//to obtain line-to-line fault from
    a three-phase fault current
If_b5_max = (V/sqrt(3)) / (Z_tr_eq + Z1 + Z2 + Z3 +
    Z4);
If_b5_min = (V/sqrt(3)) * m / (Z_tr + Z1 + Z2 + Z3 +
     Z4);
If_b4_max = (V/sqrt(3)) / (Z_tr_eq + Z1 + Z2 + Z3);
If_b4_min = (V/sqrt(3)) * m / (Z_tr + Z1 + Z2 + Z3);
If_b3_max = (V/sqrt(3)) / (Z_tr_eq + Z1 + Z2);
If_b3_min = (V/sqrt(3)) * m / (Z_tr + Z1 + Z2);
If_b2_max = (V/sqrt(3)) / (Z_tr_eq + Z1);
If_b2_min = (V/sqrt(3)) * m / (Z_tr + Z1);
If_b1_max = (V/sqrt(3)) / (Z_tr_eq);
If_b1_min = (V/sqrt(3)) * m / (Z_tr);
//Settings for relay R4
R4_I_1_p = If_b5_min /3;disp(abs(R4_I_1_p),'One
    third of minimum fault current in A')
R4_I_p = R4_I_1_p * 5 /55;disp(abs(R4_I_p),'For CT
    ratio 50/5 resulting relay current in A will be')
disp('Settings for relay R4')
disp('CT ratio = 50:5')
disp('Pick up setting in A = 5')
disp('Time-dial setting = 1/2')
//Settings for relay R3
R3_I_p = If_b4_max * 5 / 50;disp(abs(R3_I_p),'The
    relay current of both R3 and R4 for higest fault
    current seen by R4')
R4_t_op = 0.135;disp(R4_t_op,'Operating time for R4
```

```
    with time dial setting 1/2 in sec is')
t=0.3;
R3_t_op = R4_t_op + t;disp(R3_t_op,'Required
    operating time of relay R3')
disp('Settings for relay R3')
disp('CT ratio = 50:5')
disp('Pick up setting in A = 5')
disp('Time-dial setting = 2')
//Settings for relay R2
R2_I_p = (1/3) * If_b4_min * (5/100);disp(abs(R2_I_p
    ),'Pickup setting in A')
R3_I_1_p = If_b3_max * (5/50) * (1/5);disp(abs(
    R3_I_1_p),'Reatio of relay current to picup
    setting in A for max fault current through R3')
R3_t_op = 0.31;
R2_t_op = R3_t_op + t;disp(R2_t_op,'Operating time
    of R2 in sec')
R2_I_1_p = If_b3_max * (5/100) * (1/4);disp(abs(
    R2_I_1_p),'For backing up R3 the ratio of relay
    current to pickup setting of R2 in A')
disp('Settings for relay R2')
disp('CT ratio = 100:5')
disp('Pick up setting in A = 4')
disp('Time-dial setting = 2.6')
//Settings for relay R1
R1_I_p = If_b3_min * (1/3) * (5/100);
//taking tap as 5.0
R2_1_I_1_op = If_b2_max * (5/50) * (1/5);
//Operating time will come to 0.33s
R1_t_op = 0.33+t;
R1_1_I_1_op = If_b2_max * (5/100) * (1/5);
disp('Settings for relay R1')
disp('CT ratio = 100:5')
disp('Pick up setting in A = 5')
disp('Time-dial setting = 2.9')
```

Scilab code Exa 13.4 Zone

Zone

```
//Chapter 13
//Page 363
//Example 13.4
//Zone
clear;clc;
//Given
l_12 = 64;l_23 = 64;l_24 = 96;//in km
l_12m = 40;l_23m = 40;l_24m = 60;
z = 0.05 + %i * 0.5;
Pmax = 50e6;
V = 138e3;pf = 0.8;cvt = 67;
Z_12 = z * l_12;Z_23 = z * l_23;Z_24 = z * l_24;
disp('The positive sequnce impedances of the three
   line in ohms in the order line 1-2,line 2-3,line
   2-4 are')
disp(Z_12);disp(Z_23);disp(Z_24);
Il_max = Pmax / (sqrt(3) * V);disp(Il_max,'Maximum
   load current in A')
Vn = V/ sqrt(3);disp(Vn,'System Voltage to neutral')
ratio_cvt = Vn / cvt;disp('cvt ratio = 1089.1/1')
b1_factor = l_12m / ratio_cvt;
Z_r12 = Z_12 * b1_factor;
Z_r23 = Z_23 * b1_factor;
Z_r24 = Z_24 * b1_factor;
disp('The impedance of the lines as seen by R12 in
   ohms in the order line 1-2,line 2-3,line 2-4 are'
   )
disp(Z_r12);disp(Z_r23);disp(Z_r24);
```

```
Zload = (cvt * (pf + %i * sqrt(1-pf^2))) / (Il_max * (5/200));disp(Zload,'Impedance of load current')
zone1 = 0.8 * Z_r12;disp(zone1,'Setting of zone one on secondary in ohms')
zone2 = 1.2 * Z_r23;disp(zone2,'Setting of zone two on secondary in ohms')
zone3 = Z_r23 + 1.2 * Z_r24;disp(zone3,'Setting of zone three on secondary in ohms')
```

Scilab code Exa 13.5 transformer

`transformer`

```
//Chapter 13
//Page 368
//Example 13.5
//transformer
clear;clc;
//Given
V_ht = 345e3;
V_lt = 34.5e3;
P = 50e6;
P_short_term = 60e6;
I_ht = P_short_term / (sqrt(3) * V_ht);
I_lt = P_short_term / (sqrt(3) * V_lt);
disp(I_ht,I_lt,'Under maximum load the currents on 345-kV and 34.5-kV side of the transformer respectively in A')
//CT ratio on the 34.5kV side 1000/5
I_r_lt = I_lt * 5 / 1000;disp(I_r_lt,'Current flowing through the differential relay from 34.5-kV side')
I_balance = 5;
```

```
I_lt_sec_ct = I_balance / sqrt(3);disp(I_lt_sec_ct,'
    To balance the above current each of the
    secondary windings of the delta connected CTs
    should have a current(in A) of')
ct_sec = I_ht / I_lt_sec_ct;disp(ct_sec,'CT ratios
    in secondary for the above currents')
I_ht_sec_ct = I_ht * 5 / 200;disp(I_ht_sec_ct,'CT
    secondary currents for ratio 200/5 on the
    secondary side of 345-kV will be')
I_line_ctr = I_ht_sec_ct * sqrt(3);disp(I_line_ctr,'
    Line currents from CTs to differential relays')
turns_ratio = I_r_lt / I_line_ctr;disp(turns_ratio,'
    Required turns ratio for the three auxillary CTs
    uses is')
```

Chapter 14

Power System Stability

Scilab code Exa 14.1 inertia

inertia

```
//Chapter 14
//Example 14.1
//Page 380
//inertia
clear;clc;
WR2 = 5.82;
Smach = 1333;
n = 1800;
ft_lb = 746 / 550;
w = 2 * %pi * n / 60;
H = ft_lb * WR2 * w^2 / (2 * 32.2 * Smach);
disp(H,'The inertia constant in MJ/MVA is')
disp(H * Smach / 100,'Converting H to a 100-MVA
    system base,units in MJ/MVA')
```

Scilab code Exa 14.2 parallel

parallel

```
//Chapter 14
//Example 14.2
//Page 381
//parallel
clear;clc;
P1 = 500;pf1 = 0.85;V1 = 20;n1 = 3600;
P2 = 1333;pf2 = 0.9;V2 = 22;n2 = 1800;
Pbase = 100;
H1 = 4.8;H2 = 3.27;
KE = H1 * P1 + H2 * P2;
disp(KE,'The total kinetic energy of rotation of the
    two machines in MJ is')
disp(KE/Pbase,'The inertia constant for the
    equivalent machine on 100-MVA base in MJ/MVA is')
```

Scilab code Exa 14.3 infinitebus

infinitebus

```
//Chapter 14
//Example 14.3
//Page 386
//infinitebus
clear;clc;
Pm =1;
Vt = 1;V_ib = 1;
X1_g =0.2;X1_t = 0.1;X1_l1 = 0.4;X1_l2 = 0.4;
X = X1_t + X1_l1 /2;
a = asin(Pm * X / (Vt * V_ib)) * 180 / %pi;
```

```
Vt1 = Vt * (cos(a * %pi / 180) + %i * sin(a * %pi /
    180));
I = (Vt1 - V_ib) / (%i * X);
E1 = Vt1 + (%i * X1_g * I);
X1 = X1_g + X1_t + X1_l1 /2;
Pmax = abs(E1) * V_ib / X1;
disp(Vt1,'The terminal voltage after considering the
    angle is,in per unit')
disp(I,'The output current from the generator in per
    unit is')
disp(E1,'The transient internal voltage in per unit'
    )
disp('Power angle equation is')
printf("\n Pe = %.2f * sin(delta) \n where delta is
    the machine rotor angle wrt to the infinite bus",
    Pmax)
```

Scilab code Exa 14.4 onfault

`onfault`

```
//Chapter 14
//Example 14.4
//Page 388
//onfault
clear;clc;
H = 5;
Pm =1;
Vt = 1;V_ib = 1;
X1_g =0.2;X1_t = 0.1;X1_l1 = 0.4;X1_l2 = 0.4;
X = X1_t + X1_l1 /2;
a = asin(Pm * X / (Vt * V_ib)) * 180 / %pi;
```

```
Vt1 = Vt * (cos(a * %pi / 180) + %i * sin(a * %pi / 180));
I = (Vt1 - V_ib) / (%i * X);
E1 = Vt1 + (%i * X1_g * I);
y10 = %i * 3.33;
y32 = %i * 2.5;
y30 = %i * 5;
y20 = %i * 5;
Ybus = zeros(3,3);
Ybus(1,1) = -y10;Ybus(1,2) =0;Ybus(1,3) = y10;
Ybus(2,1) = Ybus(1,2);Ybus(2,2) = -(y32 + y30);Ybus(2,3) = y32;
Ybus(3,1) = Ybus(1,3);Ybus(3,2) = Ybus(2,3);Ybus(3,3) = -(y10 + y32 + y30);
disp(Ybus,'Ybus formed by inspection is')
[m,n] = size(Ybus);
Ybus_new = zeros(m-1,n-1);
for c = 1:m-1
    for d = 1:n-1
    Ybus_new(c,d) = Ybus(c,d) - ((Ybus(c,3)*Ybus(3,d)) / Ybus(3,3));
    end
end
disp(Ybus_new,'Ybus formed after elimination of Bus 3')
Pmax = abs(E1) * V_ib * abs(Ybus_new(1,2));
delta = 28.44;
Pa = Pm - Pmax * sin(delta * %pi / 180);
b = 180 * Pa / H;
disp('The power abgle equation is')
printf("\n Pe = %.3f * sin(delta) \n where delta is the machine rotor angle wrt to the infinite bus",Pmax)
disp('The swing equation is')
printf("\n (%.2f/180f) * d(delta)^2/dt^2 = %.2f - %.2fsin(delta) \n",H,Pm,Pmax)
printf("\n Intial Accelerating power is %.3f per unit \n",Pa)
```

```
printf("\n Initial acceleration is %.2f*f \n where f
    is the system frequency",b)
```

Scilab code Exa 14.5 postfault

postfault

```
//Chapter 14
//Example 14.5
//Page 389
//postfault
clear;clc
H = 5;
Pm =1;
Vt = 1;V_ib = 1;
X1_g =0.2;X1_t = 0.1;X1_l1 = 0.4;X1_l2 = 0.4;
X = X1_t + X1_l1 /2;
a = asin(Pm * X / (Vt * V_ib)) * 180 / %pi;
Vt1 = Vt * (cos(a * %pi / 180) + %i * sin(a * %pi /
    180));
I = (Vt1 - V_ib) / (%i * X);
E1 = Vt1 + (%i * X1_g * I);
y12 = 1 / (%i*(X1_g + X1_t + X1_l1));
Y12 = -y12;
Pe = abs(E1) * V_ib * abs(Y12);
disp('The power abgle equation is')
printf("\n Pe = %.3f * sin(delta) \n where delta is
    the machine rotor angle wrt to the infinite bus",
    Pe)
disp('The swing equation is')
printf("\n (%.2f/180f) * d(delta)^2/dt^2 = %.2f - %
    .2fsin(delta) \n",H,Pm,Pe)
```

Scilab code Exa 14.6 frequency

frequency

```
//Chapter 14
//Example 14.6
//Page 392
//frequency
clear;clc;
delta = 28.44;
H = 5;
ws = 377;
Pm =1;
Vt = 1;V_ib = 1;
X1_g =0.2;X1_t = 0.1;X1_l1 = 0.4;X1_l2 = 0.4;
X = X1_t + X1_l1 /2;
a = asin(Pm * X / (Vt * V_ib)) * 180 / %pi;
Vt1 = Vt * (cos(a * %pi / 180) + %i * sin(a * %pi /
    180));
I = (Vt1 - V_ib) / (%i * X);
E1 = Vt1 + (%i * X1_g * I);
X1 = X1_g + X1_t + X1_l1 /2;
Pmax = abs(E1) * V_ib / X1;
Sp = Pmax * cos(delta * %pi / 180);
wn = sqrt(ws * Sp / (2 * H));
fn = wn / (2 * %pi);
T = 1 / fn;
printf("\n The angular frequency of oscillation is %
    .3f elec rad/s \n",wn)
printf("\n The corresponding frquency of oscillation
     is %.2f Hz \n",fn)
printf("\n The period of oscillation is %.3f s",T)
```

Scilab code Exa 14.7 ccangle

ccangle

```
//Chapter 14
//Example 14.7
//Page 392
//ccangle
clear;clc;
delta = 28.44;
H = 5;
ws = 377;
Pm =1;
Vt = 1;V_ib = 1;
X1_g =0.2;X1_t = 0.1;X1_l1 = 0.4;X1_l2 = 0.4;
X = X1_t + X1_l1 /2;
a = asin(Pm * X / (Vt * V_ib)) * 180 / %pi;
Vt1 = Vt * (cos(a * %pi / 180) + %i * sin(a * %pi /
    180));
I = (Vt1 - V_ib) / (%i * X);
E1 = Vt1 + (%i * X1_g * I);
X1 = X1_g + X1_t + X1_l1 /2;
Pmax = abs(E1) * V_ib / X1;
delta_rad = delta * %pi / 180;
delta_cr = acos((%pi - 2 * delta_rad) * sin(
    delta_rad) - cos(delta_rad));
t_cr = sqrt(4 * H * (delta_cr - delta_rad) / (ws *
    Pm))
printf("\n Critical clearing angle = %.3f elec rad \
    n",delta_cr)
printf("\n Critical clearing angle for the system =
    %.3f s",t_cr)
```

Scilab code Exa 14.8 deltamax

deltamax

```
//Chapter 14
//Example 401
//Page 401
//deltamax
clear;clc;
Pm = 1;
//from previous examples
Pmax_before = 2.1;
Pmax_during = 0.808;
Pmax_after = 1.5;
delta = 28.44 * %pi / 180;
disp('The power angle equations for different times
    of fault are')
printf("\n Before the fault : \t Pmax * sin(delta) =
     %.3f * sin(delta)\n",Pmax_before)
printf("\n During the fault : \t r1 * Pmax * sin(
    delta) = %.3f * sin(delta)\n",Pmax_during)
printf("\n After the fault : \t r2 * Pmax * sin(
    delta) = %.3f * sin(delta)\n",Pmax_after)
r1 = Pmax_during / Pmax_before;
r2 = Pmax_after / Pmax_before;
delta_max = %pi - asin(Pm / Pmax_after);
cos_delta_cr = (((Pm/Pmax_before) * (delta_max -
    delta) + (r2 * cos(delta_max) - (r1 * cos(delta))
    )) / (r2 - r1));
delta_cr = acos(cos_delta_cr);
printf("\n\n r1 = %.3f \n r2 = %.3f \n",r1,r2)
```

```
printf("\n\n delta_max = %.3f rad \n cos(delta_cr) =
    %.3f \n",delta_max,cos_delta_cr)
printf("\n Critical clearing angle is %.3f degrees",
    delta_cr * 180 / %pi)
```

Scilab code Exa 14.9 multimachine

multimachine

```
//Chapter 14
//Example 14.9
//Page 404
//multimachine
clear;clc;
//Given
P_g1 = 400e6;V_g1 = 20e3;X1_dg1 = 0.067;H_g1 = 11.2;
P_g2 = 250e6;V_g2 = 18e3;X1_dg2 = 0.10;H_g2 = 8;
E_3 = 1;
//From Table 14.2
X_14 = 0.022;
X_25 = 0.040;
R_34 = 0.007;X_34 = 0.04;Y_34 = 0.082;
R_35_1 = 0.008;X_35_1 = 0.047;Y_35_1 = 0.098;
R_35_2 = 0.008;X_35_2 = 0.047;Y_35_2 = 0.098;
R_45 = 0.018;X_45 = 0.11;Y_45 = 0.226;
//From Table 14.3
V1 = 1.03 * (cos(8.88 * %pi / 180) + %i * sin(8.88 *
    %pi / 180));
P1 = 3.5;Q1 = 0.712;
V2 = 1.02 * (cos(6.38 * %pi / 180) + %i * sin(6.38 *
    %pi / 180));
P2 = 1.85;Q2 = 0.298;
V3 = 1;
```

```
V4 = 1.018 * (cos(4.68 * %pi / 180) + %i * sin(4.68
    * %pi / 180));
P4 = 1;Q4 = 0.44;
V5 = 1.011 * (cos(2.27 * %pi / 180) + %i * sin(2.27
    * %pi / 180));
P5 = 0.5;Q5 = 0.16;
//Calculations
I1 = (P1 + %i * Q1)' / V1';
I2 = (P2 + %i * Q2)' / V2';
E1_1 = V1 + %i * X1_dg1 * I1;
E1_2 = V2 + %i * X1_dg2 * I2;
E1_3 = E_3;
Y_L4 = (P4 + %i * Q4)' / (abs(V4))^2;
Y_L5 = (P5 + %i * Q5)' / (abs(V5))^2;
//formation of bus admittance matrix
Ybus = zeros(5,5);
Ybus(1,1) = 1 / (%i * (X1_dg1 + X_14));Ybus(1,4) = -
     Ybus(1,1);
Ybus(2,2) = 1 / (%i * (X1_dg2 + X_25));Ybus(2,5) = -
     Ybus(2,2);
Ybus(3,3) =  1 / (R_34 + %i * X_34) + 1 / (R_35_1 +
    %i * X_35_1) + 1 / (R_35_2 + %i * X_35_2) + %i *
    Y_34 / 2 + %i * Y_35_1;
Ybus(3,4) = - 1 / (R_34 + %i * X_34);
Ybus(3,5) = - 2 / (R_35_1 + %i * X_35_1);
Ybus(4,1) = Ybus(1,4);Ybus(4,3) = Ybus(3,4);Ybus
    (4,3) = Ybus(3,4);
Ybus(4,4) = 1 / (%i * (X1_dg1 + X_14)) + %i * Y_34 /
     2 + %i * Y_45 / 2 + 1 / (R_34 + %i * X_34) + 1 /
     (R_45 + %i * X_45) + Y_L4;
Ybus(4,5) = - 1 / (R_45 + %i * X_45);
Ybus(5,2) = Ybus(2,5);Ybus(5,3) =Ybus(3,5);Ybus(5,4)
     = Ybus(4,5);
Ybus(5,5) = 2 / (R_35_1 + %i * X_35_1) + 1 / (R_45 +
     %i * X_45) + Y_L5 + %i * Y_35_1 + %i * Y_45 / 2
    + Ybus(2,2);
disp(Ybus,'Elements of prefault bus admittance
    matrix')
```

```
printf("\n\n\n")
Ybus_1 = Ybus(1:3,1:3);
Ybus_2 = [Ybus_1 Ybus(1:3,5:5)];
Ybus_new = [Ybus_2 ; Ybus(5:5,1:3) Ybus(5,5)];
disp(Ybus_new,'After bus 4 is shorted the matrix
    becomes')
[m,n] = size(Ybus_new);
Ybus_fault = zeros(m-1,n-1);
for c = 1:m-1
    for d = 1:n-1
    Ybus_fault(c,d) = Ybus_new(c,d) - ((Ybus_new(c
        ,4)*Ybus_new(4,d)) / Ybus_new(4,4));
    end
end
printf("\n\n\n")
disp(Ybus_fault,'Elements of faulted bus admittance
    matrices')
//calculations for swing equation
Pe1 = 0;//Since G11 = real(Ybus_fault(1,1)) = 0;
    Ybus_fault(1,2) = Ybus_fault(1,3) = 0;
Pe2_1 = abs(E1_2)^2 * real(Ybus_fault(2,2));
Pe2_2 = abs(E1_2) * abs(E_3) * abs(Ybus_fault(2,3));
theta_23 = atan(real(Ybus_fault(2,3)),imag(
    Ybus_fault(2,3)));
printf("\n Pe1 = 0 \n Pe2 = %.4f + %.4fsin(delta_2 -
    %.3f) \n",Pe2_1,Pe2_2,- theta_23 * 180 / %pi)
Pa1 = P1 - Pe1;
printf("\n Swing Equation for machine 1  on fault in
     elec deg/square sec \n d^2(delta_1)/dt^2 = (180f
    /%.1f) * (%.1f) \n",H_g1,Pa1)
Pa2 = P2 - Pe2_1;
printf("\n Swing equation for machine 2 on fault in
    elec deg/square sec \n d^2(delta_2)/dt^2 = (180f/
    %.1f) * (%.4f - %.4fsin(delta_2 - %.3f)\n",H_g2,
    Pa2,Pe2_2,- theta_23 * 180 / %pi)
```

Scilab code Exa 14.10 postperiod

postperiod

```
//Chapter 14
//Example 14.10
//Page 408
//postperiod
clear;clc;
//Given
P_g1 = 400e6;V_g1 = 20e3;X1_dg1 = 0.067;H_g1 = 11.2;
P_g2 = 250e6;V_g2 = 18e3;X1_dg2 = 0.10;H_g2 = 8;
E_3 = 1;
//From Table 14.2
X_14 = 0.022;
X_25 = 0.040;
R_34 = 0.007;X_34 = 0.04;Y_34 = 0.082;
R_35_1 = 0.008;X_35_1 = 0.047;Y_35_1 = 0.098;
R_35_2 = 0.008;X_35_2 = 0.047;Y_35_2 = 0.098;
R_45 = 0.018;X_45 = 0.11;Y_45 = 0.226;
//From Table 14.3
V1 = 1.03 * (cos(8.88 * %pi / 180) + %i * sin(8.88 *
    %pi / 180));
P1 = 3.5;Q1 = 0.712;
V2 = 1.02 * (cos(6.38 * %pi / 180) + %i * sin(6.38 *
    %pi / 180));
P2 = 1.85;Q2 = 0.298;
V3 = 1;
V4 = 1.018 * (cos(4.68 * %pi / 180) + %i * sin(4.68
    * %pi / 180));
P4 = 1;Q4 = 0.44;
V5 = 1.011 * (cos(2.27 * %pi / 180) + %i * sin(2.27
    * %pi / 180));
```

```
P5 = 0.5;Q5 = 0.16;
//Calculations
I1 = (P1 + %i * Q1)' / V1';
I2 = (P2 + %i * Q2)' / V2';
E1_1 = V1 + %i * X1_dg1 * I1;
E1_2 = V2 + %i * X1_dg2 * I2;
E1_3 = E_3;
Y_L4 = (P4 + %i * Q4)' / (abs(V4))^2;
Y_L5 = (P5 + %i * Q5)' / (abs(V5))^2;
//formation of bus admittance matrix
Ybus = zeros(5,5);
Ybus(1,1) = 1 / (%i * (X1_dg1 + X_14));Ybus(1,4) = -
    Ybus(1,1);
Ybus(2,2) = 1 / (%i * (X1_dg2 + X_25));Ybus(2,5) = -
    Ybus(2,2);
Ybus(3,3) =  1 / (R_34 + %i * X_34) + 1 / (R_35_1 +
   %i * X_35_1) + 1 / (R_35_2 + %i * X_35_2) + %i *
   Y_34 / 2 + %i * Y_35_1;
Ybus(3,4) = - 1 / (R_34 + %i * X_34);
Ybus(3,5) = - 2 / (R_35_1 + %i * X_35_1);
Ybus(4,1) = Ybus(1,4);Ybus(4,3) = Ybus(3,4);Ybus
   (4,3) = Ybus(3,4);
Ybus(4,4) = 1 / (%i * (X1_dg1 + X_14)) + %i * Y_34 /
    2 + %i * Y_45 / 2 + 1 / (R_34 + %i * X_34) + 1 /
    (R_45 + %i * X_45) + Y_L4;
Ybus(4,5) = - 1 / (R_45 + %i * X_45);
Ybus(5,2) = Ybus(2,5);Ybus(5,3) =Ybus(3,5);Ybus(5,4)
    = Ybus(4,5);
Ybus(5,5) = 2 / (R_35_1 + %i * X_35_1) + 1 / (R_45 +
    %i * X_45) + Y_L5 + %i * Y_35_1 + %i * Y_45 / 2
   + Ybus(2,2);
disp(Ybus,'Elements of prefault bus admittance
   matrix')
Ybus(4,5) = 0;Ybus(5,4) = 0;
Ybus(4,4) = 1 / (%i * (X1_dg1 + X_14)) + %i * Y_34 /
    2 + 1 / (R_34 + %i * X_34) + Y_L4;
Ybus(5,5) = 2 / (R_35_1 + %i * X_35_1) + Y_L5 + %i *
    Y_35_1 + Ybus(2,2);
```

```
disp(Ybus,'After removing line 4-5')
printf("\n\n\n")
[m,n] = size(Ybus);
Ybus_1 = zeros(m-1,n-1);
for c = 1:m-1
    for d = 1:n-1
    Ybus_1(c,d) = Ybus(c,d) - ((Ybus(c,5)*Ybus(5,d)) / Ybus(5,5));
    end
end
[m1,n1] = size(Ybus_1);
Ybus_2 = zeros(m1-1,n1-1);
for c = 1:m1-1
    for d = 1:n1-1
    Ybus_2(c,d) = Ybus_1(c,d) - ((Ybus_1(c,4)*Ybus_1(4,d)) / Ybus_1(4,4));
    end
end
printf("\n\n\n")
disp(Ybus_2,'Elements of post faulted bus admittance matrices')
Pe1_1 = abs(E1_1)^2 * real(Ybus_2(1,1));
Pe1_2 = abs(E1_1) * E_3 * abs(Ybus_2(1,3));
theta_13 = atan(real(Ybus_2(1,3)),imag(Ybus_2(1,3)));
printf("\n Pe1 = %.4f + %.4fsin(delta_1 - %.3f) \n",Pe1_1,Pe1_2,- theta_13 * 180 / %pi)
Pe2_1 = abs(E1_2)^2 * real(Ybus_2(2,2));
Pe2_2 = abs(E1_2) * E_3 * abs(Ybus_2(2,3));
theta_23 = atan(real(Ybus_2(2,3)),imag(Ybus_2(2,3)));
printf("\n Pe2 = %.4f + %.4fsin(delta_2 - %.3f) \n",Pe2_1,Pe2_2,- theta_23 * 180 / %pi)
Pa1 = P1 - Pe1_1;
printf("\n Swing equation for machine 1 on fault in elec deg/square sec\n d^2(delta_2)/dt^2 = (180f/%.1f) * (%.4f - %.4fsin(delta_1 - %.3f)\n",H_g1,Pa1,Pe1_2,- theta_13 * 180 / %pi)
```

```
Pa2 = P2 - Pe2_1;
printf("\n Swing equation for machine 2 on fault in elec deg/square sec \n d^2(delta_2)/dt^2 = (180f/%.1f) * (%.4f - %.4fsin(delta_2 - %.3f)\n",H_g2,Pa2,Pe2_2,- theta_23 * 180 / %pi)
```

Scilab code Exa 14.11 stepbystep

stepbystep

```
//Chapter 14
//Page 411
//Example 14.11
//stepbystep
clear;clc;
f = 60;
t_fault = 0.225;
H = 8;
Pm = 1.85;
n = 1:10;
t = 0:0.05:1;
delta(1) = 16.19;
y = 0.755;
del_t = t(2) - t(1);
k = 180 * f * del_t^2 / H;
Pa(1) = 1.6955 - (5.5023 * sin(delta(1) * %pi / 180 - y * %pi / 180));
kPa(1) = k * Pa(1) / 2;
del_delta(1) = kPa(1);
delta(2) = delta(1) + del_delta(1);
disp('Compuatation of Swing Curve for clearing at 0.225s')
```

```
printf("\nt,s \t delta(n)-y \t Pmaxsin \t Pa \t\t
   kPa \t\t del_delta(n) \t delta(n)")
printf("\n %.2f \t %.4f \t %.4f \t %.4f \t %.4f \t
   %.4f",t(1),delta(1)-y,1.6955 - Pa(1),Pa(1),kPa(1)
   ,del_delta(1))
for m = 2:5
    Pa(m) = 1.6955 - (5.5023 * sin(delta(m) * %pi /
       180 - y * %pi / 180));
    kPa(m) = k * Pa(m);
    del_delta(m) = del_delta(m-1) + kPa(m);
    delta(m+1) = delta(m) + del_delta(m);
    printf("\n %.2f \t %.4f \t %.4f \t %.4f \t %.4f
       \t %.4f \t %.4f",t(m),delta(m)-y,1.6955 - Pa(
       m),Pa(m),kPa(m),del_delta(m),delta(m))
end
y1 = 0.847;
for m = 6:17
    Pa(m) = 1.6696 - (6.4934 * sin(delta(m) * %pi /
       180 - y1 * %pi / 180));
    kPa(m) = k * Pa(m);
    del_delta(m) = del_delta(m-1) + kPa(m);
    delta(m+1) = delta(m) + del_delta(m);
    printf("\n %.2f \t %.4f \t %.4f \t %.4f \t %.4f
       \t %.4f \t %.4f",t(m),delta(m)-y1,1.6696 - Pa
       (m),Pa(m),kPa(m),del_delta(m),delta(m))
end
```

Appendix

Scilab code AP 1 perunit

```
//x2 = pucalc(pu_Z_given,base_kV_given,base_kV_new,
   base_kVA_new,base_kVA_given)
function [X11]=pucalc(pu_Z_given,base_kV_given,
   base_kV_new,base_kVA_new,base_kVA_given)
    X11=(pu_Z_given * (base_kV_given/base_kV_new)^2
       * (base_kVA_new/base_kVA_given));
endfunction
```