Scilab Textbook Companion for Power System Analysis And Design by B. R. Gupta[1]

Created by
Ankur Gupta
B.E.
Electrical Engineering
Thapar University
College Teacher
Dr Sunil K. Singla
Cross-Checked by
Lavitha Pereira

September 17, 2013

[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: Power System Analysis And Design

Author: B. R. Gupta

Publisher: S. Chand & Co. Ltd., New Delhi

Edition: 6

Year: 2011

ISBN: 81-219-2238-0

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes 5

2 Line Parameters 15

3 Performance of Transmission lines 29

4 Overhead Line Insulators 69

5 Mechanical Design of Overhead Lines 75

6 Corona 84

7 Interference Between Power and Communication Lines 88

8 Underground Cables 93

9 Load Flow Studies 101

10 Balanced and Unbalanced Faults 133

11 Digital Techniques in Fault Calculations 176

12 Power System Transients 188

13 Power System Stability 197

16 Distribution 222

17 Design Of Transmission Lines 252

18 Power System Earthing 257

19 Voltage Stability 260

20 Reliability of Transmission and Distribution Systems 271

List of Scilab Codes

Exa 2.1 find loop resistance and inductance per km of line . . 15
Exa 2.3 inductance per km of a 3 phase line in equilateral triangle 15
Exa 2.4 inductance per km of a 3 phase line in parallel formation 16
Exa 2.5 find GMR of 4 bundled conductors 16
Exa 2.6 find inductance of bundled conductors 17
Exa 2.7 find inductance of conductors in vertical configuration 17
Exa 2.8 find inductance of conductors in given configuration . 18
Exa 2.9 capacitance between single phase conductors 19
Exa 2.10 capacitance between three phase conductors 20
Exa 2.11 capacitance of bundled conductors 20
Exa 2.12 capacitance of double circuit three phase lines in hexagon 21
Exa 2.13 capacitance of double circuit three phase lines 22
Exa 2.14 capacitance of conductor taking neutral into account . 22
Exa 2.15 resistance at 20 and 50 deg C 23
Exa 2.16 finding line parameters charging current and charging MVA 24
Exa 2.17 inductance of conductors in horizontal plane 24
Exa 2.18 inductance of conductors in horizontal plane 25
Exa 2.19 inductance of 3 wire 3 phase line in horizontal configuration 26
Exa 2.20 capacitance of conductors in horizontal plane 27
Exa 2.21 inductance per km per phase of bundled conductor . 27
Exa 3.1 convert to per unit system at common base 29
Exa 3.2 convert to per unit system at common base with neutral resistance present 30
Exa 3.3 find X of windings of 3 winding transformer 31
Exa 3.4 find voltage regulation and capacitor required to make voltage regulation 0 32

Exa 3.5 receiving end voltage and current 34
Exa 3.6 receiving end voltage and current 34
Exa 3.7 determine per phase R and X for given efficiency . . . 35
Exa 3.8 receiving end voltage and current power factor and Voltage regulation using nominal T circuit 36
Exa 3.9 receiving end voltage and current power factor and Voltage regulation using nominal pi circuit 37
Exa 3.10 receiving end voltage and current power factor and Voltage regulation using nominal pi circuit 38
Exa 3.11 find receiving end parameters 39
Exa 3.12 find OC receiving end parameters 41
Exa 3.13 find characteristic impedance propagation constant and ABCD for line 42
Exa 3.14 find receiving end voltage and current 43
Exa 3.15 finding and comparing pi and T network parameters . 44
Exa 3.16 sending end parameters using nominal pi circuit and long line equations 45
Exa 3.17 ABCD parameters of pi network 47
Exa 3.18 ABCD parameters of composite system 47
Exa 3.19 ferrenti effect 49
Exa 3.20 P and Q consumed by generator and motor in circuit and line losses 49
Exa 3.21 compensation paramenters 50
Exa 3.22 find tapsetting of transformer 52
Exa 3.23 find tap setting under given conditions 53
Exa 3.26 find capacity of phase modifier at different loads . . . 54
Exa 3.28 power transfer and SPM rating to improve pf 55
Exa 3.29 overall ABCD parameters 56
Exa 3.30 find wavelength and velocity of propagation 57
Exa 3.31 sending end parameters using pu 57
Exa 3.32 find voltage at sending end 59
Exa 3.33 find pu values of system 60
Exa 3.34 find pu values of system 61
Exa 3.35 find pu values of system 63
Exa 3.36 calculate actual values of generator current line current load current load voltage and load power from pu . . . 65
Exa 3.38 sending and receiving end voltage and current in parallel OH lines 66

Exa 3.39	find receiving end voltage and efficiency of transmission	67
Exa 4.1	find voltage across string and string efficiency	69
Exa 4.2	calculate string efficiency with presesnce of guard ring	70
Exa 4.3	find voltage across string and string efficiency	70
Exa 4.4	find capacitance ratio system voltage and string efficiency	71
Exa 4.5	guard ring find string efficiency	72
Exa 4.6	voltage across various discs in insulator	72
Exa 4.7	line to oin capacitances so that voltage distribution is uniform	73
Exa 4.8	find mutual capacitances of insulator discs	73
Exa 4.9	find ratio of capacitances of insulator to earth capacitance of insulator	74
Exa 5.1	finding sag in different weather conditions	75
Exa 5.2	clearance of line	76
Exa 5.3	height of mid point from ground	77
Exa 5.4	finding sag	77
Exa 5.5	finding minimum clearance and position of clearance point	78
Exa 5.6	find sag and tension under erection conditions	79
Exa 5.7	representing line as parabola and catenary	80
Exa 5.8	galloping and dancing conductors find clearance under ice and air conditions	81
Exa 5.9	galloping and dancing conductors find clearance under no ice and air conditions	82
Exa 5.10	find maximum sag under given condition	82
Exa 6.1	Finding local and general visual and disruptive corona voltage	84
Exa 6.2	Finding total loss in fair weather and bad weather using peeks formula	85
Exa 6.3	finding visual corona voltage	86
Exa 6.4	finding minimum distance between conductors to limit disruptive corona	86
Exa 7.1	finding magnitude of voltage induced in telephone line due to EMI of power line	88
Exa 7.2	finding magnitude of voltage induced in telephone line due to EMI of power line under fault	89
Exa 7.3	potential of conductor due electrostatic effect	89

Exa 7.4 Voltage induced in telephone conductor due electrostatic effect 90
Exa 7.5 Voltage induced in conductor due electrostatic effect . 91
Exa 8.1 inductance of a 3 core belted cable 93
Exa 8.2 find most economical diameter of cable so that it not exceed max stress 93
Exa 8.3 find most economical diameter of cable so that it not exceed max stress 94
Exa 8.4 find postitions of intersheaths max min stress and voltage on intersheaths 95
Exa 8.5 radius voltage of intersheath and ratio of maximum stress with and wothout intersheath 96
Exa 8.6 find maximum voltage in a cable having 2 insulation materials 96
Exa 8.7 parameters of underground feeder 97
Exa 8.8 effective capacitance in cables 98
Exa 8.9 find current rating of cable 99
Exa 9.1 form Y bus 101
Exa 9.3 form Y bus and effect of adding a line 102
Exa 9.4 find y bus 103
Exa 9.5 find missing elements of y bus 104
Exa 9.7 find y bus with mutual coupling of lines present 105
Exa 9.8 find reactive power generations losses and powers transferred 106
Exa 9.9 solve using gauss seidel for 1 variable 109
Exa 9.10 solve using gauss seidel for 2 variables 109
Exa 9.11 find bus voltage and load angle using GS 110
Exa 9.12 find bus voltage and load angle using GS minimum value of Q2 given 113
Exa 9.13 solve using newton raphson 1 variable 115
Exa 9.14 solve using newton raphson 2 variables 116
Exa 9.15 solve using newton raphson 1 variable 117
Exa 9.17 solve system using newton raphson method 117
Exa 9.18 solve system using fast decoupled method 123
Exa 9.19 solve system using gauss seidel method with acceleration constant 130
Exa 10.1 find fault current and fault level 133

Exa 10.2 find fault level and X to limit current during 3 phase fault 134
Exa 10.3 find fault current and fault level during 3 phase fault . 135
Exa 10.4 find subtransient currents in system 136
Exa 10.5 calculate total generator and motor current in 3phase fault 137
Exa 10.6 find symmetrical components 138
Exa 10.8 find zero sequence components 139
Exa 10.10 find fault MVA and current and line to line voltages during fault 140
Exa 10.11 thevinin equlvalent impedances of sequence networks as seen from fault point 142
Exa 10.12 find fault current voltage at fault point and current and voltage at generator terminal during LG fault 143
Exa 10.13 calculate fault current during LG fault 148
Exa 10.14 find line currents and voltages under LL fault conditions 149
Exa 10.15 find line currents and voltages under LLG fault conditions 150
Exa 10.16 find line currents under LG fault conditions 152
Exa 10.17 find pu values of sequence networks 154
Exa 10.18 calculate current in generator and motor during fault . 155
Exa 10.19 find short circuit MVA of parallel connection of 2 stations 156
Exa 10.20 find X to prevent overloading of circuit breakers 157
Exa 10.21 determine fault current and voltages during LG fault when different alternator neutrals are grounded or isolated 158
Exa 10.22 determine fault current and voltages during LG fault when alternator neutral is grounded and isolated . . . 159
Exa 10.24 find reactance added to limit fault current in LG fault 161
Exa 10.27 find SC MVA for 3 phase fault 162
Exa 10.28 find fault current and fault level LG 3 phase LL and LLG faults 163
Exa 10.29 find fault current and fault level LG fault in middle of line 165
Exa 10.30 find fault current and fault level LG fault 169
Exa 10.31 find line voltages and currents for OC fault 171
Exa 10.32 fault MVA with and without reactors 172
Exa 10.33 find subtransient current in system 173

Exa 10.34 reactance needed to restrict 6 times fault current . . . 174
Exa 10.35 symmetrical components of line and delta currents . . 174
Exa 11.1 z bus formulation 176
Exa 11.2 formulate positive and negative sequence impedance matrices for the network 177
Exa 11.3 formulate zero sequence impedance matrices for the network 178
Exa 11.4 finding fault current and fault voltage at bus 179
Exa 11.5 finding fault current and fault voltage at bus 181
Exa 11.6 find z bus 184
Exa 11.7 find z bus of an augmented network 185
Exa 12.1 find L C surge impedance and velocity of propagation 188
Exa 12.2 find surge transmitted 188
Exa 12.3 find surge Vand I transmitted 189
Exa 12.4 find voltage across the inductance and the reflected voltage wave 189
Exa 12.7 find surge arrester voltage and current 190
Exa 12.8 find surge arrester voltage and current 191
Exa 12.9 find reflection and refraction coefficients 191
Exa 12.10 reflection and transmission of voltage and current wave 192
Exa 12.11 find V and I transmitted 193
Exa 12.12 reflection transmission and absorption of wave 194
Exa 12.14 find voltage and current surges 195
Exa 12.16 find restriking voltage due to current chopping 196
Exa 13.1 find P Q E and load angle for changes to P and E . . 197
Exa 13.2 find inertia constants retardation 198
Exa 13.3 find steady state stability parameters 199
Exa 13.4 derive expressions for oscillations of delta and freq as functions of time 200
Exa 13.5 finding steady state reactance and transfer limit for different shunt branches 202
Exa 13.6 frequency of oscillation of generator due to loading . . 203
Exa 13.7 system stability and finding critical load angle 203
Exa 13.8 system stability and finding critical load angle in 3 phase fault in line 205
Exa 13.9 system stability and finding critical load angle in 3 phase fault at bus 207

Exa 13.10 system stability and finding critical load angle due to sudden loading 207
Exa 13.11 inertia constant of 2 generators in parallel 209
Exa 13.12 find frequency deviation in case of delay in opening steam valve 209
Exa 13.13 find critical clearing angle during prefault fault and post fault conditions 210
Exa 13.14 point by point solution of swing equation 210
Exa 13.15 find maximum load that can be supplied by generator 216
Exa 13.16 finding steady state reactance and transfer limit for different shunt branches 216
Exa 13.17 find inertia constant and momentum of generator . . . 217
Exa 13.18 find inertia constant of each machine and parallel ombination 218
Exa 13.19 find critical clearing angle and critical clearing time in 3 phase fault conditions 219
Exa 19.20 finding acceleration torque and change in torque angle due to losses 220
Exa 16.1 find voltage at load points in single feeded dc feeder . 222
Exa 16.2 find voltage at load points in addition to distributed load in single feeded dc feeder 223
Exa 16.3 find voltage at load points in doubly feeded dc feeder . 224
Exa 16.4 find voltage at load points in addition to distributed load in doubly feeded dc feeder 225
Exa 16.5 voltage drop in singly feeded ac feeder with concentrated load 226
Exa 16.6 voltage drop in singly feeded ac feeder with distributed and concentrated load 226
Exa 16.7 currents in a 3 phase ac circuit 227
Exa 16.8 voltage drop at the end of one phase in unbalanced 3 phase network 228
Exa 16.9 find supply voltage and phase angle between sending end and receiving end 229
Exa 16.10 find currents in a hexagon shaped concentrated loads . 230
Exa 16.11 find point of minimum in a line 232
Exa 16.12 voltage at far end in a double ac conductor with concentrated load 232

Exa 16.13 voltage at far end in a double ac conductor doubly fed with concentrated load 233
Exa 16.14 find currents in a pentagon shaped concentrated loads with an interconnector 234
Exa 16.15 find currents in a triangle shaped loads 235
Exa 16.16 find optimum cross section of cables for comsumers at different distances 236
Exa 16.17 voltage at far end in a double ac conductor singly fed with uniform and concentrated load 237
Exa 16.18 voltage at far end in a double ac conductor with concentrated load 238
Exa 16.19 find most economical center of distribution 239
Exa 16.20 find optimum cross section for double dc line doubly fed concentrated load 240
Exa 16.21 find currents in a hexagon shaped concentrated loads in 2 line dc ring main 241
Exa 16.22 kelvins law 243
Exa 16.23 find cross section of cable for given losses in a singly fed ac conductor 244
Exa 16.24 find loss factor load factor annual load loss and annual cost of lost energy 246
Exa 16.25 effect of starting of induction motor on domestic load . 247
Exa 16.26 effect of adding capacitor bank on current and voltage 248
Exa 16.27 percentage change in losses by adding capacitor bank . 249
Exa 16.28 rating of switched bank and fixed bank capacitors . . . 250
Exa 17.1 Design Of Transmission Lines 252
Exa 18.1 resistance of grounding electrode 257
Exa 18.2 resistance of different arrangements of grounding electrode 257
Exa 18.3 earthing resistance of wire buried to different depths . 258
Exa 19.1 finding sending and receiving end reactive power . . . 260
Exa 19.2 compensating value of capacitor 261
Exa 19.3 find receiving end voltage if breaker opens suddenly . 262
Exa 19.5 capacity of SVS 263
Exa 19.6 voltage and pf of bus before compensation 263
Exa 19.7 voltage and pf of bus after compensation 264
Exa 19.8 T parameters of compensated and uncompensated system 265

Exa 19.9 pi parameters of compensated and uncompensated system . . . 266
Exa 19.10 voltage regulation of compensated line . . . 267
Exa 19.11 find var injection to bring voltage to original value . . 269
Exa 20.1 find failure rate of system down time per outage annual outage . . . 271
Exa 20.2 find reliability of series connected elements . . . 272
Exa 20.3 find reliability of parallel connected elements . . . 272
Exa 20.4 find reliability of series connected elements and MTTF 273
Exa 20.5 find reliability of parallel connected elements and MTTF 273
Exa 20.6 find improved reliability . . . 274
Exa 20.7 find improved reliability . . . 274
Exa 20.8 find number of components to achieve desired reliability 275
Exa 20.9 find failure rate of system down time per outage annual outage . . . 275
Exa 20.10 find failure rate of system down time per outage annual outage . . . 277

List of Figures

13.1 point by point solution of swing equation 211

Chapter 2

Line Parameters

Scilab code Exa 2.1 find loop resistance and inductance per km of line

```
clear
clc;
dia=1.213;
dist=1.25*100;
f=50;
rad=dia/2;
effrad=.7788*rad;
L=4d-7 * log (dist/effrad)*1d3;
X=2*%pi*f*L;
mprintf("L=%.2f *1e-4 H/km, X=%.1f ohm/km",L*1e4, X)
    ;
```

Scilab code Exa 2.3 inductance per km of a 3 phase line in equilateral triangle

```
clear
clc;
Dia=1.63;
```

```
Dist=3*100;
Reff=.7788*.5*Dia;
Reff=round(Reff*1e3)*1e-3
L = .4605 * log10(Dist/Reff);

mprintf("L= %.4f mH/km",L)
```

Scilab code Exa 2.4 inductance per km of a 3 phase line in parallel formation

```
clear
clc;
rad=1.81;
Dist1=6*100;
Dist2=Dist1;
Dist3=12*100;
Reff=.7788*rad;
Dist=(Dist1*Dist2*Dist3)^(1/3);
L = .4605 * log10(Dist/Reff);
mprintf("L=%.4f mH/km",L);
```

Scilab code Exa 2.5 find GMR of 4 bundled conductors

```
clear
clc;
d=25;
rad=1.573;
Ds=.7788*rad;
Dsb=((Ds*d*d*sqrt(2)*d)^4)^(1/16);
mprintf("GMR= %.3f cm",fix(Dsb*1000)/1000);
```

Scilab code Exa 2.6 find inductance of bundled conductors

```
clear
clc;
d=45;
D=12e2;
rad=1.6;

//(a)
reff=.7788*rad;
GMR=(reff*d*reff*d)^(1/4);
Dab=(D*(D+d)*D*(D-d))^(1/4);
Dbc=Dab;
Dca=((D*2)*((2*D)+d)*(D*2)*((2*D)-d))^(1/4);
Deq=(Dab*Dbc*Dca)^(1/3);
L=.4605 *log10 (Deq/GMR);
mprintf("\n(a)\tL=%.2f mH/km",L);

//(b)
R=sqrt(2)*rad;
reff=R*.7788;
Deq=(D*D*(2*D))^(1/3);
L=.4605 *log10 (Deq/reff);

mprintf("\n(b)\tL=%.3f mH/km",L);
```

Scilab code Exa 2.7 find inductance of conductors in vertical configuration

```
clear
clc;
D=3e2;
h=5e2;
rad=1.1;

```

```
m=(h^2 + D^2)^(1/2);
n=(h^2 + (D*2)^2)^(1/2);
reff=.7788* rad;
L=2e-7 * log ((2)^(1/6)* (D/reff)^(1/2) * (m/n)
    ^(1/3)) *1e3;
mprintf("L=%.2f * 1e-4 H/phase/km",L*1e4);
```

Scilab code Exa 2.8 find inductance of conductors in given configuration

```
clear;
clc;
D=3e2;
d1=6e2;
d2=7e2;
rad=.9;

reff=.7788* rad;
Daa=(d1^2 + d1^2)^(1/2);
Dcc=Daa;
Dbb=d2;
GMRa=sqrt(reff*Daa);
GMRb=sqrt(reff*Dbb);
GMRc=sqrt(reff*Dcc);
Ds=(GMRa*GMRb*GMRc)^(1/3);
Ds=round(Ds*10)/10

Dab=(D^2 + ((d2-d1)/2)^2)^(1/2);
Dcb=Dab;
Dc1b1=Dab;
Da1b1=Dab;

Dab1=(D^2 + (((d2-d1)/2)+d1)^2)^(1/2);
Da1b=Dab1;
Dc1b=Dab1;
Dcb1=Dab1;
```

```
Dac=2*D;
Da1c1=Dac;
Da1c=(d1);
Dac1=Da1c;

GMRab=(Dab*Da1b1*Da1b*Dab1)^(1/4);
GMRbc=(Dcb*Dc1b1*Dc1b*Dcb1)^(1/4);
GMRac=(Dac*Da1c1*Da1c*Dac1)^(1/4);

Deq=(GMRab*GMRbc*GMRac)^(1/3);
Deq=round(Deq*10)/10

L=2e-7 * log (Deq/Ds) * 1e3;
mprintf("L=%.3f *1e-4 H/phase/km",L*1e4);
```

Scilab code Exa 2.9 capacitance between single phase conductors

```
clear;
clc;

l=10;
D=1.25;
dia=1.213e-2;

Cab=.01206/log10(D/(dia/2))
C=l*Cab;
Cn=2*C;

mprintf("Capacitance between 2 conductors= %.4f e-6F
   /km\n", C);
mprintf("Capacitance between conductor and neutral=
   %.4f e-6F/km\n", Cn);
```

Scilab code Exa 2.10 capacitance between three phase conductors

```
clearglobal;
clc;

V=220e3;
f=50;
l=200;
d1=6;
d2=12;
rad=1.81e-2;

Deq=(d1*d1*d2)^(1/3);
Cn=.02412/log10(Deq/rad);
mprintf("Capacitance per phase per km = %.1f *1e-3 e-6F/phase/km \n",Cn *1000);
C=l*Cn;
C=round(C*100)/100
mprintf("Capacitance per phase = %.2f e-6F/phase\n",C);
Xc=1/(2 * %pi * f * C *1e-6);
mprintf("Capacitive reactance per phase = %.0f ohms/phase\n",Xc);
I=2 * %pi * f * C *1e-6 * V / sqrt(3);
I=round(I*100)/100
mprintf("Charging current = %.2f A/phase\n",I);
MVA=sqrt(3)*V *I *1e-6;
mprintf("Charging MVA = %.2f MVA\n",fix(MVA*100)/100);
```

Scilab code Exa 2.11 capacitance of bundled conductors

```
clear;
clc;

r= 1.6e-2;
d= 45e-2;
D=12;
Dscb=sqrt(r*d);
Deq=(D*D*(2*D))^(1/3);
Cn= .02412/(log10 (Deq/Dscb));
mprintf("Capacitance per phase per km= %.4f e-6 F/km
   \n",Cn);
```

Scilab code Exa 2.12 capacitance of double circuit three phase lines in hexagon

```
clear
clc;

D=350;
r=1.09;
f=50;
V=132e3;
d=100;

//(b)
Cn=.04824 / log10((sqrt(3)*D)/(2*r));
Cn=round(Cn*10000)/1e4
C=Cn/2;
mprintf("Capacitance per conductor per km = %.5f e-6
    F/conductor/km\n",C);

//(c)
w=2 * %pi * f;
Vn=V/sqrt(3);
Ic= w * Cn * Vn * 1e-6;
```

```
Ic=round(Ic *1e3)/1e3
I=Ic * d;
MVA= sqrt(3)*V * I / 1e6;
mprintf("Charging MVA = %.2f MVA\n",MVA);
```

Scilab code Exa 2.13 capacitance of double circuit three phase lines

```
clear

clc;

r=.9e-2;
d=6;
D1=6;
D2=7;

Daa=sqrt ((d*d)+(D1*D1));
Daa=round(Daa*1e3)/1e3
Dbb=D2;
Dcc=Daa;
Deq=5.074;

Dsc_a = round(sqrt(r*Daa)*1e4)/1e4;
Dsc_b = round(sqrt(r*Dbb)*1e4)/1e4;
Dsc_c = round(sqrt(r*Dcc)*1e4)/1e4;

Dsc=(Dsc_a*Dsc_b*Dsc_c)^(1/3);
Cn=.02412/log10(Deq/Dsc);

mprintf("Capacitance per conductor per km = %f e-6 F
    /conductor/km\n",Cn);
```

Scilab code Exa 2.14 capacitance of conductor taking neutral into account

```
clear;
clc;

h=5.5;
Dia=1.213e-2;
d=1.25;
l=10;

r=Dia/2;
Cn=(2*.01206)/(log10 (d/(Dia* 0.5 * sqrt(1 + (d*d
    *.25/h^2)))));
C=Cn * l;

mprintf("Capacitance per conductor= %.5f e-6 F/
    conductor\n",C);
```

Scilab code Exa 2.15 resistance at 20 and 50 deg C

```
clear
clc;

dia=4.22e-3;
n=6;
T1=20;
T2=50;
rho1 = 2.826e-8;
a=.004;
dL=1.5e-2;
L=1000;

A=%pi * n *dia *dia / 4;
R1= rho1 * L * (1+dL) / A;
R2= R1 * (1+ (a * (T2-T1)));

mprintf("Temperature at %d = %.4f ohm/km\n", T1, R1)
```

```
;
mprintf("Temperature at %d = %.4f ohm/km", T2, R2);
```

Scilab code Exa 2.16 finding line parameters charging current and charging MVA

```
clear;
clc;

A= 1.5e-4;
Deff=39.8e-3;
D= 8;
rho1=1.73e-6 / 100;
l=1e3;
f=50;
V=132e3;

//(a)
R= rho1 * l / A;
r=.5 * Deff;
L= .4605 * log10 (D/(.7788 *r));
mprintf("L = %.2f mH/km\n",L);

C= .02412/(log10 (D/r));
mprintf("C = %.5f e-6 F/km\n",C);

//(b)
Ic = 2 * %pi * f * C *1e-6 * V / sqrt(3);
mprintf("Charging current = %.4f A/km/phase\n",Ic);
MVA=sqrt(3)*V *Ic *1e-6;
mprintf("Charging MVA = %.4f MVA/km\n",MVA);
```

Scilab code Exa 2.17 inductance of conductors in horizontal plane

```
clear;
clc;

r=.9e-2;
d=3.5;


//(b)

L= .4605 * log10(1.375 * sqrt(d/r));
mprintf("(b) L= %.2f mH/km", L);
```

Scilab code Exa 2.18 inductance of conductors in horizontal plane

```
clear;
clc;
clear all;


dia=26.88;
d=450
d1=15.25*1e3;

R=dia/2;
R1= .7788 * R;
R1=round(R1*100)/100
GMR= 1.09 * (R1 * d*d*d)^(.25);
GMR=round(GMR*10)/10
GMD=(d1* d1 * (2*d1))^(1/3);
GMD=round(GMD/10)*10
L=.4605 * log10 (GMD/GMR);

mprintf(" L= %.4f mH/km", L);
```

Scilab code Exa 2.19 inductance of 3 wire 3 phase line in horizontal configuration

```
clear
clc

dia=2.5e-2
d=3

r=dia/2
r1=.7788*r

c=exp(%i *2*%pi/3)
b=exp(%i *-2*%pi/3)
k=2

Dab=d
Dac=2*d
Dbc=d
Dca=2*d

La=round(k*(log(1/r1) + log(1/Dab)*b + log(1/Dac)*c)
    *1e3)/1e4
Lb=round(k*(log(1/Dab) + log(1/r1)*b + log(1/Dbc)*c)
    /b*1e3)/(1e4)
Lc=round(k*(log(1/Dac) + log(1/Dbc)*b + log(1/r1)*c)
    /c*1e3)/(1e4)

disp(La, "La= (mH/km)")
disp(Lb, "Lb= (mH/km)")
disp(Lc, "Lc= (mH/km)")
```

Scilab code Exa 2.20 capacitance of conductors in horizontal plane

```
clear;
clc;

dia=26.88e-3;
d=45e-2;
d1=15.25;

r=dia/2;

GMR= 1.09 * (r * d*d*d)^(.25);
GMD=(d1* d1 * (2*d1))^(1/3);

C= 0.02412 / log10 (GMD/GMR);
mprintf("C= %.4f e-6 F/km", C);
```

Scilab code Exa 2.21 inductance per km per phase of bundled conductor

```
clear;
clc;
f=50;
dia=25e-3;
sp=.3;
D=6;

rad=dia/2;
r=.7788*rad;
GMR= ((r*r*sp*sp)^.25);
Dab= (D* (D+sp)* D *(D-sp))^.25;
Dbc=Dab;
Dac= ((2*D)* ((2*D) +sp)* (2*D) *((2*D)-sp))^.25;
Deq=(Dab * Dbc *Dac)^(1/3);
L=.4605 * log10 (Deq/GMR);
mprintf("\nL= %.3f mH/phase/km", L);
```

```
XL=2 * %pi * f * L *1e-3;
mprintf("\nXL= %.2f ohm/phase/km", XL);
```

Chapter 3

Performance of Transmission lines

Scilab code Exa 3.1 convert to per unit system at common base

```
clear;
clc;


sg1=10e6;
vg1=13.2e3;
sg2=15e6;
vg2=13.2e3;
sm1=8e6;
vm1=12.5e3;
sm2=12e6;
vm2=12.5e3;
Xg=15;
Xm=20;

sb=50e6;
vb=13.8e3;

xg1=Xg * (vg1/vb)^2 * (sb/sg1);
```

```
xg2=Xg * (vg2/vb)^2 * (sb/sg2);
xm1=Xm * (vm1/vb)^2 * (sb/sm1);
xm2=Xm * (vm2/vb)^2 * (sb/sm2);
 mprintf ("\nReactance of Generator 1= %.2f percent"
    ,xg1);
 mprintf ("\nReactance of Generator 2= %.2f percent"
    ,xg2);
 mprintf ("\nReactance of Motor 1= %.2f percent",xm1
    );
 mprintf ("\nReactance of Motor 2= %.2f percent",xm2
    );
```

Scilab code Exa 3.2 convert to per unit system at common base with neutral resistance present

```
clear;
clc;

vg=11e3;
sg=90e6;
xg=.25;

st1=100e6;
vt1a=10e3;
vt1b=132e3;
nt1=vt1a/vt1b;
xt1=.06;

st2=30e6 *3;
vt2a=66e3 * sqrt(3);
vt2b=10e3;
nt2=vt2a/vt2b;
xt2=.05;

sm1=50e6;
```

```
vm1=10e3;
xm1=.2;

sm2=40e6;
vm2=10e3;
xm2=.2;

xl=100;

Sb=sg;
Vbg=vg;

Xg=xg * (vg/Vbg)^2 * (Sb/sg);
Xt1=xt1 * (vt1a/Vbg)^2 * (Sb/st1);

Vbl=Vbg/nt1;
Xl=xl * (Sb) / (Vbl)^2 ;

Vbm=Vbl/nt2;

Xt2=xt2 * (vt2b/Vbm)^2 * (Sb/st2);
Xm1=xm1 * (vm1/Vbm)^2 * (Sb/sm1);
Xm2=xm2 * (vm2/Vbm)^2 * (Sb/sm2);

 mprintf ("\nReactance of Generator = %.2f ",Xg);
 mprintf ("\nReactance of Transformer 1= %.4f ",Xt1);
 mprintf ("\nReactance of Line = %.3f ",Xl);
 mprintf ("\nReactance of Transformer 2= %.3f ",Xt2);
 mprintf ("\nReactance of Motor 1= %.3f ",Xm1);
 mprintf ("\nReactance of Motor 2= %.3f ",Xm2);
```

Scilab code Exa 3.3 find X of windings of 3 winding transformer

```
clear;
clc;

s1=30e6;
v1=132e3;
s2=20e6;
v2=11e3;
v3=6.6e3;
s3=10e3;

xa=.07;
xb=.09;
xc=.04;
va=v1;
vb=v1;
vc=v2;
sa=s1;
sb=s1;
sc=s2;

Sb=s1;
Vb1=v1;
Vb2=v2;
Vb3=v3;

Xa=xa * (va/Vb1)^2 * (Sb/sa);
Xb=xb * (vb/Vb1)^2 * (Sb/sb);
Xc=xc * (vc/Vb2)^2 * (Sb/sc);

X = [1 1 0; 1 0 1; 0 1 1 ]^(-1) * [ Xa; Xb; Xc];

mprintf("\nX1= %.2f pu",X(1,1));
mprintf("\nX2= %.2f pu",X(2,1));
mprintf("\nX3= %.2f pu",X(3,1));
```

Scilab code Exa 3.4 find voltage regulation and capacitor required to make voltage regulation 0

```
clear;
clc;

d=15;
Vr=11e3/sqrt(3);
pfr=.8;
Pd=5e6;
Pl=.12*Pd;
l=1.1e-3;
L=l*d;

I= Pd/(3*pfr*Vr);
R=Pl/(3*I*I);
X=2 * %pi* 50 *L;
pfa=acos(pfr);
Vs=Vr + (I * R * pfr) + (I * X * sin(pfa));
vs=sqrt(3)*Vs;

VR=(Vs-Vr)/Vr;

mprintf("\n(a) Voltage Regulation = %.2f percent ", VR*100);

pfa0=atan(R/X);
pf0=cos(pfa0);

mprintf("\n(b) pf at VR=0 = %.3f ", pf0);

I0= (I* pfr)/pf0;
Ic= (I * sin(pfa))+(I0*sin(pfa0));
Xc=Vr/Ic
C=1/(100*%pi*Xc);

mprintf("\n(c) C = %.1f e-6 F", C*1e6);
```

Scilab code Exa 3.5 receiving end voltage and current

```
clear;
clc;

Vs=11e3/sqrt(3);
Pd=1200e3;
R=5.31;
X=5.54;
pfr=.8;


VIr= Pd/(3*pfr);
a=1;
b=-1*Vs;
pfa=acos(pfr)
c=(VIr * R * pfr) + (VIr * X * sin(pfa))

Vr=(-b+sqrt(b^2 - (4*a*c)))/(2*a);

I=VIr/Vr;
vr=sqrt(3)*Vr;

mprintf("Recieving end Voltage = %.3f KV and Current
    = %.2f A ",vr/1000,I);
```

Scilab code Exa 3.6 receiving end voltage and current

```
clear;
clc;

Pd=1e6;
```

```
pf=.8;
v1=30e3;
v2=10e3;
Rl=25;
Xl=12;
rt=.8;
xt=2.5;
n=v1/v2;

Rt=rt*(n^2);
Xt=xt*(n^2);

R=Rt+Rl;
X=Xt+Xl;

Vr=v1/sqrt(3);
I=Pd/(3*Vr*.8);

pfa=acos(pf)
Vs=Vr + (I * R * pf) + (I * X * sin(pfa));
vs=sqrt(3)*Vs*1e-3;

VR=(Vs-Vr)/Vr;

mprintf("\nSending End Voltage = %.2f KV",vs);
mprintf("\nVoltage Regulation= %.2f",VR*100);
```

Scilab code Exa 3.7 determine per phase R and X for given efficiency

```
clear;
clc;

Vs=33e3/sqrt(3);
Vr=30e3/sqrt(3);
Pr=10e6;
```

```
pf=.8;
eff=.96;

I=Pr/(3*Vr*pf);

Ps=Pr/eff;
Pl=Ps-Pr;

R=Pl/(3*I*I);
pfa=acos(pf);
X=((Vs-Vr)-(I*R*pf))/(I*sin(pfa));

mprintf("R= %.1f ohm per phase, X= %.1f ohm per
    phase",R,X);
```

Scilab code Exa 3.8 receiving end voltage and current power factor and Voltage regulation using nominal T circuit

```
clear;
clc;

R=48.7;
X=80.2;
Z=complex(R,X);
c=8.42e-9;
l=200;
C=c*l;
Y=complex(0,(C* 100*%pi));

Vr=88e3/sqrt(3);
Pr=13.5e6;
pf=.9;
pfa=-1* acos(pf);
Irm=Pr/(3*Vr*pf);
Ir=complex(Irm *pf, Irm * sin(pfa));
```

```

Vs=(Vr*(1+((Z*Y)/2)))+ (Ir*Z*(1+(Z*Y/4)));
V=abs(Vs);
vs=sqrt(3)*V*1e-3;
phi=atan(imag(Vs)/real(Vs))*(180/%pi);
mprintf("\nSending End Voltage = %.2f kV",vs);
mprintf("\nSending End Power Angle = %.1f deg ",phi);

Is=(Vr*Y)+(Ir*(1+(Y*Z/2)));
I=abs(Is);
mprintf("\nSending End Current = %.2f A",I);

Vr0=V/(1+(Y*Z/2));
V0=abs(Vr0);
VR=(V0-Vr)/Vr;
mprintf("\nVoltage Regulation = %.1f ",VR*100);
```

Scilab code Exa 3.9 receiving end voltage and current power factor and Voltage regulation using nominal pi circuit

```
clear;
clc;

R=48.7;
X=80.2;
Z=complex(R,X);
c=8.42e-9;
l=200;
C=c*l;
Y=complex(0,(C* 100*%pi));

Vr=88e3/sqrt(3);
Pr=13.5e6;
pf=.9;
```

```
pfa=-1* acos(pf);
Irm=Pr/(3*Vr*pf);
Ir=complex(Irm *pf, Irm * sin(pfa));

Vs=(Vr*(1+((Z*Y)/2)))+ (Ir*Z);
V=abs(Vs);
vs=sqrt(3)*V*1e-3;
phi=atan(imag(Vs)/real(Vs))*(180/%pi);
mprintf("\nSending End Voltage = %.2f kV",vs);
mprintf("\nSending End Power Angle = %.2f deg ",phi);

Is=(Vr*Y*(1+(Z*Y/4)))+(Ir*(1+(Y*Z/2)));
I=abs(Is);
mprintf("\nSending End Current = %.2f A",I);

Vr0=V/(1+(Y*Z/2));
V0=abs(Vr0);
VR=(V0-Vr)/Vr;
mprintf("\nVoltage Regulation = %.2f ",VR*100);
```

Scilab code Exa 3.10 receiving end voltage and current power factor and Voltage regulation using nominal pi circuit

```
clear;
clc;

d=100;
f=50;
r=.153;
l=1.21e-3;
c=.00958e-6;
xl=2*%pi*f*l;
xc=1/(2*%pi*f*c);
Vr=110e3/sqrt(3);
```

```
Pr=20e6;
pf=.9;
pfa=-1*acos(pf);
Irm=Pr/(3*Vr*pf);
Ir=complex(Irm *pf , Irm * sin(pfa));

Z=complex(r,xl) * d;
Y=complex(0,1/xc) *d;

//disp(abs(Z),Y,abs(Ir));

Vs=(Vr*(1+((Z*Y)/2)))+ (Ir*Z);
V=abs(Vs);
vs=sqrt(3)*V*1e-3;
mprintf("\nSending End Voltage = %.2f kV",vs);

Is=(Vr*Y*(1+(Z*Y/4)))+(Ir*(1+(Y*Z/2)));
I=abs(Is);
mprintf("\nSending End Current = %.2f A",I);

phi1=atan(imag(Vs)/real(Vs))*(180/%pi);
phi2=atan(imag(Is)/real(Is))*(180/%pi);
phi=phi1-phi2;
pfs=cosd(phi);
mprintf("\nSending End Power factor = %.3f  ",pfs);


Vr0=V/(1+(Y*Z/2));
V0=abs(Vr0);
VR=(V0-Vr)/Vr;
mprintf("\nVoltage Regulation = %.2f ",VR*100);

eff=Pr*100/(3*pfs*V*I)
mprintf("\nEfficiency = %.0f percent ",eff);
```

Scilab code Exa 3.11 find receiving end parameters

```
clear;
clc;

d=500;
z=complex(.105, .3768);
y=complex(0, 2.822e-6);
Z=z*d;
Y=y*d;
YZ=Y*Z;

A=1+(YZ/2)+((YZ)^2/24);
B=Z * (1+(YZ/6)+((YZ)^2/120));
C=Y * (1+(YZ/6)+((YZ)^2/120));
D=A;

A=round(abs(A)*10000)/10000 * exp(%i * round(atan(imag(A)/real(A))*100)/100)
B=round(abs(B)*1000)/1000   * exp(%i * round(atan(imag(B)/real(B))*100)/100)

//see(B)


Vr=220e3/sqrt(3); //incorrectly taken as 127021 in textbook.
Vr=round(Vr)
Pr=40e6;
pf=.9;
pfa=-1*acos(pf);
Irm=Pr/(3*Vr);
Ir=complex(Irm *pf, Irm * round(sin(pfa)*100)/100);

Vs=(A*Vr)+(B*Ir);
V=abs(Vs);
vs=sqrt(3)*V*1e-3;
phi1=atand(imag(Vs)/real(Vs));
```

```
mprintf("\nSending End Voltage = %.2f kV",vs);

Is=(C*Vr)+(D*Ir);
I=abs(Is);
phi2=atand(imag(Is)/real(Is))
mprintf("\nSending End Current = %.2f A",I);

phi=phi2-phi1;
pfs=cosd(phi);
mprintf("\nSending End Power factor = %.3f  ",pfs);
mprintf("\nSending End Power Angle = %.3f  ",phi1);

MVA=sqrt(3) * vs* I /1000;
mprintf("\nSending End Power = %.3f  ",MVA);

disp("difference in results is due to taking Vr=127021V instead of 127017V")
```

Scilab code Exa 3.12 find OC receiving end parameters

```
clear;
clc;

d=500;
z=complex(.105, .3768);
y=complex(0, 2.822e-6);
Z=z*d;
Y=y*d;
YZ=Y*Z;

A=1+(YZ/2)+((YZ)^2/24);
B=Z * (1+(YZ/6)+((YZ)^2/120));
C=Y * (1+(YZ/6)+((YZ)^2/120));
D=A;

```

```
Vr=220e3/sqrt(3);
Pr=40e6;
pf=.9;
pfa=-1*acos(pf);
Ir=0;

Vs=(A*Vr)+(B*Ir);
V=abs(Vs);
vs=sqrt(3)*V*1e-3;
phi1=atan(imag(Vs)/real(Vs))*(180/%pi);
mprintf("\nSending End Voltage = %.2f kV",vs);

Is=(C*Vr)+(D*Ir);
I=abs(Is);
phi2=atan(imag(Is)/real(Is))*(180/%pi);
mprintf("\nSending End Current = %.1f A",I);

phi2=phi2+180;
phi=phi1-phi2;
pfs=cosd(phi);
mprintf("\nSending End Power factor = %.4f  ",pfs);
```

Scilab code Exa 3.13 find characteristic impedance propagation constant and ABCD for line

```
clear;
clc;

Z=complex(14.1, 51.48);
Y=complex(0, 1.194e-3);
Zc=sqrt(Z/Y);
g=sqrt(Z*Y);
A=cosh(g);
B=Zc* sinh(g);
C=sinh(g)/Zc;
```

```
D=A;

mprintf("\nZc= %s", string(round(abs(Zc)*1000)/1000) +'/_'+ string(round(atand(imag(Zc)/real(Zc))*100)/100) )
mprintf("\npropgation const= %s", string(round(abs(g)*1000)/1000) +'/_'+ string(round(atand(imag(g)/real(g))*100)/100) )
mprintf("\nA= %s", string(round(abs(A)*1000)/1000) + '/_'+ string(round(atand(imag(A)/real(A))*100)/100) )
mprintf("\nB= %s", string(round(abs(B)*1000)/1000) + '/_'+ string(round(atand(imag(B)/real(B))*100)/100) )
mprintf("\nC= %s", string(round(abs(C)*1000)/1000) + '/_'+ string(round(atand(imag(C)/real(C))*100)/100 +180) )
mprintf("\nD= %s", string(round(abs(D)*1000)/1000) + '/_'+ string(round(atand(imag(D)/real(D))*100)/100) )
```

Scilab code Exa 3.14 find receiving end voltage and current

```
clear;
clc;

Z=complex(200*cosd(80), 200*sind(80));
Y=complex(.0013*cosd(90),.0013*sind(90));
YZ=Y*Z;

A=1+(YZ/2)+((YZ)^2/24);
B=Z * (1+(YZ/6)+((YZ)^2/120));
phiA=atan(imag(A)/real(A))*(180/%pi);
phiB=atan(imag(B)/real(B))*(180/%pi);

```

```
P=60e6
pf=.8;
Vs=round(220/sqrt(3))*1e3;
VIrm=P/(3*pf)
pfa=acos(pf);
VIr=complex(VIrm *pf, VIrm * sin(pfa));

pfa=pfa*(180/%pi);
a=(round(abs(A)*1000)/1000)^2;
b=round(((2*(abs(A)* (VIrm)*cosd(phiA)*abs(B)*cosd(phiB-pfa)) )+ (2*(abs(A)* (VIrm)*sind(phiA)*abs(B)*sind(phiB-pfa))) - (Vs^2))/1e7)*1e7;
c=abs(B)^2 * (VIrm)^2;
Vr=sqrt((-b+sqrt((b*b)-(4*a*c)))/(2*a));

vr=sqrt(3) * Vr / 1000;
Ir=VIr/(Vr*pf);
mprintf("Receiveing End Line voltage= %.0f kV", fix(vr));
mprintf("\n Receiveing End Line Current= %.0f A", Ir);
```

Scilab code Exa 3.15 finding and comparing pi and T network parameters

```
clear;
clc;

Z=complex(180*cosd(75) , 180*sind(75));
Y=complex(1e-3*cosd(90) , 1e-3*sind(90));

g=sqrt(Y*Z);
Zc=sqrt(Z/Y);

Z1=Zc * sinh(g);
Y1=(1/Zc) *1e3*( (cosh(g)-1)/sinh(g));
```

```


mprintf("\nZpi= %s", string(round(abs(Z1)*100)/100) +'/_'+ string(round(atand(imag(Z1)/real(Z1))*100)/100) )
mprintf("\nYpi/2= %s *1E-3", string(round(abs(Y1)*10000)/10000) +'/_'+ string(round(atand(imag(Y1)/real(Y1))*10)/10) )
```

Scilab code Exa 3.16 sending end parameters using nominal pi circuit and long line equations

```
clear;
clc;

Vr=132e3/sqrt(3);
P=40e6;
pf=.8;
Irm=P/(3*Vr)
pfa=-1* acos(pf);
Ir=complex(Irm *pf, Irm * sin(pfa));

Z=complex(52, 200)
Y=complex(0, 1.5e-3)
YZ=Y*Z;

A=1+ (YZ/2);
D=A;
B=Z;
C=Y*(1+(YZ/4));

Vs=(A*Vr)+(B*Ir);
V=abs(Vs)
vs=V*sqrt(3)*1e-3;
mprintf("(a)\nSending End Voltage= %.0f kV", vs)
```

```

Is=(C*Vr)+(D*Ir);
I=abs(Is)
mprintf("\nSending End Current= %.1f A", I)

phi1=atan(imag(Vs)/real(Vs))*(180/%pi);
phi2=atan(imag(Is)/real(Is))*(180/%pi);
phi=phi1-phi2;
pfs=cosd(phi);
mprintf("\nSending End pf= %.3f ",pfs)

Ps=sqrt(3)* vs * I * pfs /1000;
mprintf("\nSending End Power= %.1f ",Ps)


//(b)
Zc=sqrt(Z/Y);
g=sqrt(Z*Y);
A=cosh(g);
B=Zc* sinh(g);
C=sinh(g)/Zc;
D=A;

Vs=(A*Vr)+(B*Ir);
V=abs(Vs)
vs=V*sqrt(3)*1e-3;
mprintf("\n\n\n(b)\nSending End Voltage= %.1f kV", vs)

Is=(C*Vr)+(D*Ir);
I=abs(Is)
mprintf("\nSending End Current= %.1f A", fix(I*10)/10)

phi1=atan(imag(Vs)/real(Vs))*(180/%pi);
phi2=atan(imag(Is)/real(Is))*(180/%pi);
phi=phi1-phi2;
pfs=cosd(phi);
```

```
mprintf("\nSending End pf= %.3f ",pfs)

Ps=sqrt(3)* vs * I * pfs /1000;
mprintf("\nSending End Power= %.1f ",Ps)
```

Scilab code Exa 3.17 ABCD parameters of pi network

```
clear;
clc;

Y1=500^-1;
Y2=1000^-1;
Z=100;

A= 1+Y2 * Z;
B=Z;
C=Y1+Y2+(Y1*Y2*Z);
D=1+Y1 * Z

mprintf("A= %.1f  ; B= %.1f ohm  ; C=%.1f  *1e-3seimens
    ; D= %.1f", A, B, C*1e3, D);
```

Scilab code Exa 3.18 ABCD parameters of composite system

```
clear;
clc;

A1=complex(.98 * cosd(2), .98* sind(2));
B1=complex(28 * cosd(69), 28* sind(69));
C1=complex(.0002 * cosd(80), .0002* sind(80));
D1=A1;

A2=complex(.95 * cosd(3), .95* sind(3));
```

```
B2=complex(40 * cosd(85), 40* sind(85));
C2=complex(.0004 * cosd(90), .0004* sind(90));
D2=A2;

//(a)

A= (A1* A2) + (B1* C2);
B= (A1* B2) + (B1* D2);
C= (C1* A2) + (D1* C2);
D= (C1* B2) + (D1* D2);


mprintf("(a)");
mprintf("\nA= %s", string(round(abs(A)*1000)/1000) + '/_'+ string(round(atand(imag(A)/real(A))*10)/10) )
mprintf("\nB= %s", string(round(abs(B)*100)/100) +'/_'+ string(round(atand(imag(B)/real(B))*100)/100) )
mprintf("\nC= %s *1e-4", string(round(abs(C)*100000)/10) +'/_'+ string(round(atand(imag(C)/real(C))*10)/10 ) )
mprintf("\nD= %s", string(round(abs(D)*1000)/1000) + '/_'+ string(round(atand(imag(D)/real(D))*10)/10) )


//(b)

Vr=110e3/sqrt(3);
pf=.95;
Irm=200
pfa=-1* acos(pf);
Ir=complex(Irm *pf, Irm * sin(pfa));

Vs=(A*Vr)+(B*Ir);
V=abs(Vs)
vs=V*sqrt(3)*1e-3;
```

```
mprintf("\n\n\n(b)\nSending End Voltage= %.2f kV",vs)

Is=(C*Vr)+(D*Ir);
I=abs(Is)
mprintf("\nSending End Current= %.1f A", I)

phi1=atan(imag(Vs)/real(Vs))*(180/%pi);
phi2=atan(imag(Is)/real(Is))*(180/%pi);
phi=phi1-phi2;
pfs=cosd(phi);
mprintf("\nSending End pf= %.2f ",pfs)
```

Scilab code Exa 3.19 ferrenti effect

```
clear;
clc;

Vr=220e3/sqrt(3);
d=300;
f=50;

V=-1*((Vr*(2*%pi*f)^2 * d*d *1e-10)/18);
Vs=Vr+V;
vs=sqrt(3) * Vs /1000;
mprintf("Sending end voltage=%.2f kV, and voltage rise =%.0f V/phase", vs, -1*V)
```

Scilab code Exa 3.20 P and Q consumed by generator and motor in circuit and line losses

```
clear;
clc;
```

```
E=11e3/sqrt(3);
E1=complex(E*cosd(0), E*sind(0));
E2=complex(E*cosd(40), E*sind(40));
Z=complex(0,15);
I=(E1-E2)/Z;

if (real(I)<0)     then
      mprintf("E1=generator, E2=motor");
    else
        mprintf("E2=generator, E1=motor");
end

S1=3 * E1 *1e-6* conj(I);
S2=3 * E2 *1e-6* conj(I);

mprintf("\n\n(a)Real Power consumed by E2= %.3f MW,\nPower delivered by E1= %.3f MW", -1*real(S2), -1*real(S1) );
mprintf("\n\n(b) Reactive Power supplied by E1= %.3f MVAr,\nPower supplied by E2= %.3f MVAr", imag(S1),imag(S2)*-1 );
mprintf("\n\n(C) Reactive Power absorbed by line= %.3f MVAr", 2* imag(S1) );
```

Scilab code Exa 3.21 compensation paramenters

```
clear;
clc;

Vr=132e3/sqrt(3);
P=50e6;
pf=.8;
Irm=P/(3*Vr)
pfa=-1* acos(pf);
```

```
Ir=complex(Irm *pf, Irm * sin(pfa));

A=complex(.98*cosd(3), .98*sind(3));
B=complex(110*cosd(75), 110*sind(75));

Vs=(A*Vr)+(B*Ir);
V=abs(Vs)
vs=V*sqrt(3)*1e-3;
mprintf("\n\n\n(a)\nSending End Voltage= %.1f kV", vs)

phi1=atan(imag(Vs)/real(Vs))*(180/%pi);
mprintf("\nPower Angle= %.2f ",phi1)


Ss=(((vs)^2 * conj(A/B))-((sqrt(3)*Vr/1000)*(sqrt(3)*Vs/1000)/conj(B)));
Ps=real(Ss);
Qs=imag(Ss);

mprintf("\n\n\n(b)\nSending End Active Power= %.1f MW", Ps)
mprintf("\nSending End Reactive Power= %.1f MVar lagging", Qs)

Pl=Ps-(P * cos(pfa)*1e-6);
Ql=Qs-(P* -1*sin(pfa)*1e-6);
mprintf("\n\n\n(c)\nLine Loss= %.1f MW", Pl)
mprintf("\nMVar absorbed by line= %.1f MVar", Ql)

Pr=(P * cos(pfa)*1e-6);
Qr=(P * sin(pfa)*1e-6);
Vs1=140;
Vr1=132;

bd=acos(( Pr+real(((Vr1)^2 * conj(A/B))) ) * (abs(B)/(Vs1 *Vr1)));
Qr1= (((Vs1 *Vr1)/abs(B))*sin(bd))- imag(((Vr1)^2 *
```

```
    conj(A/B)));
Q=-Qr-Qr1;
mprintf("\n\n\n(d)\nCapacity of static capacitor= %
    .1f MVar", Q)

Vs2=132;
Vr2=132;
bd2=asin(( imag(((Vr2)^2 * conj(A/B))) ) * (abs(B)/(
    Vs2 *Vr2)));
P2= (((Vs2 *Vr2)/abs(B))*cos(bd2))- real(((Vr2)^2 *
    conj(A/B)));
mprintf("\n\n\n(e)\nPower supplied= %.2f MW",P2)
```

Scilab code Exa 3.22 find tapsetting of transformer

```
clear;
clc;

Pr=90e6
pf=.9;
S=Pr/(3*pf);

P=Pr/3
Q=sqrt(S^2 - P^2);

V1=220e3/sqrt(3);
V2=220e3/sqrt(3);

R=15
X=50;

tr=sqrt(1-(((R*P)+(X*Q))/(V1^2)));
mprintf("tap setting  tr= %.4f, ts=%.3f",fix(1e4/tr)
    /1e4,tr)
```

Scilab code Exa 3.23 find tap setting under given conditions

```
clear;
clc;

Vb=132
Sb=100
X=.15

v1=125
V1=v1/Vb;
Q1=50;
Qpu1=Q1/Sb;

Vn1=(V1 + sqrt(V1^2 - (4*Qpu1*X)))/(2*1);
vn1=Vn1 * Vb;
Vo1=33;
t1=vn1/Vo1;

v2=140
V2=v2/Vb;
Q2=20;
Qpu2=Q2/Sb;

Vn2=(V2 + sqrt(V2^2 - (4*Qpu2*X)))/(2*1);
vn2=Vn2 * Vb;
Vo2=33;
t2=vn2/Vo2;

tm=(t1+t2)/2;
dt=tm-t1;
ts=dt*100/tm

mprintf("tap setting = +- %.0f percent", ts);
```

Scilab code Exa 3.26 find capacity of phase modifier at different loads

```
clear;
clc;

Vr1=132
Vs1=140
VA=40;
pf=.8;
Pr=VA*pf;
pfa=-1* acos(pf);
Qr=(VA * sin(pfa));

A=complex(.98*cosd(3), .98*sind(3));
B=complex(110*cosd(75), 110*sind(75));

bd1=acos(( Pr+real(((Vr1)^2 * conj(A/B))) ) * (abs(B)/(Vs1 *Vr1)));
Qr1= (((Vs1 *Vr1)/abs(B))*sin(bd1))-imag(((Vr1)^2 * conj(A/B)));

Q1=-Qr-Qr1;
mprintf("\n\n\n(a)\nCapacity of static capacitor= %.2f MVar leading", fix(Q1*100)/100)

Prn=0;
Qrn=0;

bd2=acos((Prn+real(((Vr1)^2 * conj(A/B))) ) * (abs(B)/(Vs1 *Vr1)));
Qr2= (((Vs1 *Vr1)/abs(B))*sin(bd2))-imag(((Vr1)^2 * conj(A/B)));

Q2=-Qrn-Qr2;
```

```
mprintf("\n\n\n(b)\nCapacity of static capacitor= %.1f MVar lagging", -Q2)
```

Scilab code Exa 3.28 power transfer and SPM rating to improve pf

```
clear;
clc;

Vr=220
Vs=240

A=complex(.9*cosd(1), .9*sind(1));
B=complex(140*cosd(84), 140*sind(84));

b=atan(imag(B)/real(B));
d=b
vs=complex (Vs *cos(d), Vs *sin(d));
P= (((Vs *Vr)/abs(B))*cos(b-d))- real(((Vr)^2 * conj(A/B)));
mprintf("(a) Max Power Transmitted =  %.2f MW", P);

Pr=80;
Sr=100;
Qr=sqrt(Sr^2 - Pr^2);
bd1=round(acos(( Pr+real(((Vr)^2 * conj(A/B))) ) * (abs(B)/(Vs *Vr)))*1000)/1000;
Qr1= (((Vs *Vr)/abs(B))*sin(bd1))-imag(((Vr)^2 * conj(A/B)));

Q=+Qr-Qr1;
mprintf("\n\n\n(b)\nCapacity of static capacitor= %.2f MVar leading", Q)
disp("There is a calculation error in the textbook. 40-49.37=10.63")

```

```
d1=b-bd1;
d1=d1*180/%pi
mprintf("\n\n\n(c)\nLoad Angle factor= %.2f deg", d1)
```

Scilab code Exa 3.29 overall ABCD parameters

```
clear;
clc;

A=complex(.93*cosd(3), .93*sind(3));
B=complex(150*cosd(70), 150*sind(70));
D=A;

C=((A*D)-1)/B

Z=complex(100*cosd(70), 100*sind(70));
Y=complex(.00025*cosd(-75), .00025*sind(-75));

m1=[A B; C D]
m2=[1 0 ; Y 1 ]
m3=[1 Z ; 0 1 ]

A0=m1*m2*m3;
A1=A0(1,1)
B1=A0(1,2)
C1=A0(2,1)
D1=A0(2,2)

mprintf("\nA0= %s", string(round(abs(A1)*1000)/1000)+'/_'+ string(round(atand(imag(A1)/real(A1))*10)/10) )
mprintf("\nB0= %s", string(round(abs(B1)*1000)/1000)+'/_'+ string(round(atand(imag(B1)/real(B1))*100)/100) )
```

```
mprintf("\nC0= %s", string(round(abs(C1)*100000)
    /100000) +'/_'+ string(round(atand(imag(C1)/real(
    C1))*1)/1 ) )
mprintf("\nD0= %s", string(round(abs(D1)*1000)/1000)
     +'/_'+ string(round(atand(imag(D1)/real(D1))*10)
    /10) )
```

Scilab code Exa 3.30 find wavelength and velocity of propagation

```
clear;
clc;

f=50;
l=200;
Z=complex(14.1, 51.48);
Y=complex(0, 1.194e-3);
g=sqrt(Z*Y);

b=imag(g)/l;
wl=2*%pi/b;
v=f*wl
mprintf("\nwavelength = %.4f*1e3 km",wl*1e-3);
mprintf("\nVelocity of Propagation = %.2f*1e5 km/sec
    ",v*1e-5);
```

Scilab code Exa 3.31 sending end parameters using pu

```
clear;
clc;

clear;
clc;

```

```
vr=220e3/sqrt(3);
Vb=vr
Vr=vr/Vb
Sr=40e6;
Sb=40e6;
Ib=Sb/(3*Vb)
Zb=Vb/Ib;

pf=.9;
pfa=-1*acos(pf);
Irm=(Sr/(3*vr))/Ib;
Ir=complex(Irm *pf, Irm * sin(pfa));


d=500;
z=complex(.105, .3768);
y=complex(0, 2.822e-6);
Z1=z*d;
Y1=y*d;
Z=Z1/Zb;
Y=Y1*Zb;
YZ=Y*Z;

A=1+(YZ/2)+((YZ)^2/24);
B=Z * (1+(YZ/6)+((YZ)^2/120));
C=Y * (1+(YZ/6)+((YZ)^2/120));
D=A;


Vs=(A*Vr)+(B*Ir);
V=abs(Vs);
vs=sqrt(3)*V*1e-3*Vb;
phi1=atan(imag(Vs)/real(Vs))*(180/%pi);
mprintf("\nSending End Voltage = %.2f kV",vs);

Is=(C*Vr)+(D*Ir);
I=abs(Is)*Ib;
```

```
phi2=atan(imag(Is)/real(Is))*(180/%pi);
mprintf("\nSending End Current = %.1f A",I);

phi=phi2-phi1;
pfs=cosd(phi);
mprintf("\nSending End Power factor = %.3f  ",pfs);

MVA=sqrt(3) * vs* I /1000;
mprintf("\nSending End Power = %.2f  ",MVA);
```

Scilab code Exa 3.32 find voltage at sending end

```
clear;
clc;

VAt1=10
VAt2=10
Xt1=.1;
Xt2=.08
Vt1a=13.8
Vt1b=138
Vt2a=138
Vt2b=69
n1=Vt1b/Vt1a
n2=Vt2b/Vt2a


Sb=10
Vbb=138;
Vba=Vbb / n1;
Vbc=Vbb*n2;

Zbc=Vbc^2/Sb;

R=300;
```

```
Rpu=R/Zbc;

v=66
V=v/Vbc;
I=V/Rpu;
Va=V+(I*complex(0,Xt1+Xt2));
va=abs(Va)*Vba;

mprintf("Voltage in ckt A = %.3f kV", va);
```

Scilab code Exa 3.33 find pu values of system

```
clear;
clc;

xg1=.2
xg2=.3
xt1=.2;
xt2=.06
zl=complex(40,150);

Vg1=250
Vg2=250;
Vt1a=250
Vt1b=800;
Vt2a=1000;
Vt2b=500
nt1=Vt1b/Vt1a;
nt2=Vt2b/Vt2a;

sg1=2000;
sg2=2000;
st1=4000;
st2=8000;

```

```
Vb1=250;
Vb2=Vb1*nt1;
Vb3=Vb2*nt2;

Sb=5000;
Zbl=Vb2^2/Sb;

Zl=zl/Zbl;
Xt1=xt1 /( (Vb1/Vt1a)^2 * (st1/Sb));
Xt2=xt2 / ((Vb2/Vt2a)^2 * (st2/Sb));
Xg1=xg1 /  (sg1/Sb);
Xg2=xg2 /  (sg2/Sb);

 mprintf ("\nReactance of Generator 1= %.1f ",Xg1);
 mprintf ("\nReactance of Generator 2= %.2f ",Xg2);
 mprintf ("\nReactance of Transformer 1= %.2f ",Xt1)
    ;
 disp (round(Zl*1e4)/1e4,"Impedance of Line = ");
 mprintf ("\nReactance of Transformer 2= %.4f ",Xt2)
    ;
```

Scilab code Exa 3.34 find pu values of system

```
clear;
clc;

sg1=10
sg2=20
st1=10
st2=10*3;

vg1=6.6;
vg2=11.5;
vt1a=6.6
vt1b=115
```

```
vt2a=75*sqrt(3);
vt2b=7.5*sqrt(3);
nt1=vt1b/vt1a;
nt2=vt2b/vt2a;
xg1=.1
xg2=.1
xt1=.15
xt2=.1


Sb=20;
Vb1=6.6;
Vb2=Vb1*nt1;
Vb3=Vb2*nt2;

Xg1=xg1*Sb/sg1;
Xg2=xg2*Sb/sg2;
Xt1=xt1 * Sb/st1;
Xt2= xt2 * Sb/st2 * (vt2a/Vb2)^2

 mprintf ("\nReactance of Generator 1= %.1f",Xg1);
 mprintf ("\nReactance of Generator 2= %.1f",Xg2);
 mprintf ("\nReactance of Transformer 1= %.1f ",Xt1)
    ;
 mprintf ("\nReactance of Transformer 2= %.5f ",fix(
    Xt2*1e5)/1e5);


Zbl=Vb2^2/Sb;

xab=100;
xad=100
xbc=150;
xbd=200;
xdc=50;

Xab=xab/Zbl;
Xad=xad/Zbl;
```

```
Xbc=xbc/Zbl;
Xbd=xbd/Zbl;
Xdc=xdc/Zbl;

printf("\nXab=%.4f\t", Xab);
printf("Xad=%.4f\t", Xad);
printf("Xbc=%.4f\t", Xbc);
printf("Xbd=%.4f\t", Xbd);
printf("Xdc=%.4f\t", Xdc);
```

Scilab code Exa 3.35 find pu values of system

```
clear;
clc;

sg1=25
sg2=15
sg3=30
st1=30
st2=15
st3=10*3;

vg1=6.6;
vg2=6.6;
vg3=13.2;
vt1a=6.6
vt1b=115
vt2a=6.6
vt2b=115
vt3a=69*sqrt(3);
vt3b=6.9*sqrt(3);
nt1=vt1b/vt1a;
nt2=vt2b/vt2a;
nt3=vt3b/vt3a;

```

```
xg1=.2
xg2=.15
xg3=.15
xt1=.1
xt2=.1
xt3=.1

Sb=30;
Vb1=6.6;
Vb2=Vb1*nt1;
Vb3=Vb2*nt2;
Vb4=Vb2*nt3;

Xg1=xg1*Sb/sg1;
Xg2=xg2*Sb/sg2;
Xg3=xg3*Sb/sg3* (vg3/Vb4)^2;
Xt1=xt1 * Sb/st1;
Xt2=xt2 * Sb/st2;
Xt3= xt2 * Sb/st3 * (vt3b/Vb4)^2

 mprintf ("\nReactance of Generator 1= %.2f pu",Xg1);
 mprintf ("\nReactance of Generator 2= %.1f pu",Xg2);
 mprintf ("\nReactance of Generator 3= %.4f pu",Xg3);
 mprintf ("\nReactance of Transformer 1= %.1f pu",Xt1);
 mprintf ("\nReactance of Transformer 2= %.1f pu",Xt2);
 mprintf ("\nReactance of Transformer 3= %.3f pu",Xt3);


Zbl=Vb2^2/Sb;

xl1=120;
xl2=90
```

```
Xl1=xl1/Zbl;
Xl2=xl2/Zbl;

 mprintf ("\nReactance of Line 1 = %.4f pu",Xl1);
 mprintf ("\nReactance of line 2 = %.3f pu",Xl2);
```

Scilab code Exa 3.36 calculate actual values of generator current line current load current load voltage and load power from pu

```
clear;
clc;

vg=11e3;
sg=80e6;
xg=.25;

st1=100e6;
vt1a=11e3;
vt1b=220e3;
nt1=vt1a/vt1b;
xt1=.05;

st2=150e6;
vt2a=230e3 ;
vt2b=33e3;
nt2=vt2a/vt2b;
xt2=.04;

r=250;

zl=complex(5,100);

Sb=100e6;
Vbg=11e3;
```

```
Xg=xg * (vg/Vbg)^2 * (Sb/sg);
Xt1=xt1 * (vt1a/Vbg)^2 * (Sb/st1);

Vbl=Vbg/nt1;
Zl=zl * (Sb) / (Vbl)^2 ;

Vbm=Vbl/nt2;
Xt2=xt2 * (vt2a/Vbl)^2 * (Sb/st2);
R=r * (Sb) / (Vbm)^2 ;


Vg=vg/Vbg;
i=Vg/(Zl+complex(R,Xt1+Xt2+Xg))
I=round(abs(i)*1e4)/1e4;

Ic=I*Sb/(sqrt(3)*Vbg);
Il=I*Sb/(sqrt(3)*Vbl);
Ir=I*Sb/(sqrt(3)*Vbm);
Vload=Ir*r/1000;
Vloadll=sqrt(3) * Vload;
Pr=3*Ir*Ir*r/1e6;
mprintf("\n Generator Current = %.1f A",Ic);
mprintf("\n Line Current = %.3f A",Il);
mprintf("\n Load Current = %.1f A",Ir);
mprintf("\n Load Voltage = %.2f kV",Vloadll);
mprintf("\n Load Power = %.3f MW",Pr);
```

Scilab code Exa 3.38 sending and receiving end voltage and current in parallel OH lines

```
clear;
clc;

z1=complex(4,6)
```

```
z2=complex(3,2)

Vs=3.3e3/sqrt(3)
Is=250
pf=.8
pfa=acos(pf)
I=Is *(exp(%i * -pfa))

I1=I * z2/(z1+z2)
pfa1=atan(imag(I1)/real(I1))
pf1=cos(pfa1)
mprintf("\n(a) Current in OH line = %.1f A pf= %.3f", abs(I1), pf1)

I2=I * z1/(z1+z2)
pfa2=atan(imag(I2)/real(I2))
pf2=cos(pfa2)
mprintf("\n(b) Current in cable = %.2f A pf= %.2f", abs(I2), pf2)

vr=sqrt((Vs)^2-imag(I1*z1)^2)- real(I1*z1)
Vr=vr*sqrt(3)/1000;
mprintf("\n(c) Receiving end voltage = %.3f KV", Vr)

d=atan(imag(I1*z1)/(Vr+real(I1*z1)))
phi=pfa-d;
pfr=cos(phi)
mprintf("\n(d) Receiving end pf = %.1f lagging", pfr)
```

Scilab code Exa 3.39 find receiving end voltage and efficiency of transmission

```
clear;
clc;
```

```
l=300
R=.4 *3
X=.8*3
Vs=11e3/sqrt(3);
P=3000;
pf=.8
pfa=acos(pf)
VIr=P/(3*pf)

a=1;
b=-Vs
c=VIr * 1e3 * ((R*cos(pfa))+(X*sin(pfa)))
vr=(-b+sqrt((b*b )-  (4*a*c)))/(2*a)
Ir=VIr*1e3/vr;
Vr=vr*sqrt(3)/1000;
mprintf("\nReceiving End Voltage = %.2f kV",Vr)

Pl=3* (Ir)^2 * R/ 1000;
eff=P*100/(P+Pl)
mprintf("\nefficiency = %.2f percent",eff)
```

Chapter 4

Overhead Line Insulators

Scilab code Exa 4.1 find voltage across string and string efficiency

```
clear;
clc;

C=1;
C1=0.1;
V=66;
n=4;

v1= 1;
v2= (C+C1) * v1;
v3= (C * v2) + (C1*(v1+v2));
v4= (C * v3) + (C1*(v1+v2+v3));

V1= V/(sqrt(3)*(v1+v2+v3+v4));
V2= v2* V1;
V3= v3* V1;
V4= v4* V1;
mprintf("\n V1= %.2f kV",V1);
mprintf("\n V2= %.2f kV",V2);
mprintf("\n V3= %.2f kV",V3);
mprintf("\n V4= %.2f kV",V4);
```

```

eff= (V1+V2+V3+V4)/(n*V4) *100;
mprintf("\n string efficiency= %.1f percent",eff);
```

Scilab code Exa 4.2 calculate string efficiency with presesnce of guard ring

```
clear
clc

c1=.15
c2=.05
V=100

A=[
1+c1 -(1+c2) -c2
c1 1+c1 -(1+c2)
1 0 0
]

B=[0 0 1]'
Vm=inv(A) * B
Vm = round(Vm*1e4)/1e4
V1=V/(Vm(1)+Vm(2)+Vm(3))
V2=Vm(2) * V1
V3=Vm(3) * V1

ef=V/(3*V3)

mprintf("Voltage distribution in percentage of total voltage:\nV1=%.2f\tV2=%.2f\tV3=%.2f\nstring efficiency=%.1f percent",V1, V2, V3,ef*100)
```

Scilab code Exa 4.3 find voltage across string and string efficiency

```
clear;
clc;

n=3
V=11;
C1=1;
C2=.2 * C1;

v1=1;
v2=(C1+C2)*v1/C1;
v3=((C1*v2)+(C2*(v1+v2)))/C1;

V3=V
V1=fix((V3/v3)*100)/100;
V2=round((V1*v2)*100)/100;

Vln=V1+V2+V3;
Vll=sqrt(3)*Vln;
eff=Vln*100/(n*V);

mprintf("\n(a) Maximum line to neutral voltage = %.2
   f kV", Vln);
mprintf("\n(b) String Efficiency = %.0f percent",
   eff);
```

Scilab code Exa 4.4 find capacitance ratio system voltage and string efficiency

```
clear;
clc;
v3=20;
v2=15;
//putting v1=15/(1+k)
s=poly([-1 5 3],"x","coeff");
K=roots(s);
```

```
k=K(2)
v1=15/(1+k);
//disp(v1)
x=v1(1);
//disp(x);
vnew=x+v3+v2;
xl=sqrt(3)*vnew;
n=vnew/(3*v3);
mprintf("capacitance ratio= %.2f \nthe line to
    neutral voltage= %.1fkV \n string efficiency=%.1
    fpercent",k,xl,n*100);
```

Scilab code Exa 4.5 guard ring find string efficiency

```
clear;
clc;
a=.26;
b=.15;
c=.35;
y=[(1+b) b;-(1+a) (1+c)];
z=[1+a;a];
v=round(inv(y)*z*1e3)*1e-3;
t=v(1,1);
u=v(2,1);
n=(t+u+1)/(3*u);
mprintf("the string efficiency is =%.2f",n);
```

Scilab code Exa 4.6 voltage across various discs in insulator

```
clear;
clc;
k=.1;
n=4;
```

```
for i=1:4
    z(i)=2*cosh((i-.5)*sqrt(k))*sinh(.5*sqrt(k))/
        sinh(n*sqrt(k));
end
mprintf("v1= %.3fV,  v2= %.3fV,  v3= %.3fV,  v4= %.3fV"
    , z(1), z(2), z(3), z(4))
```

Scilab code Exa 4.7 line to oin capacitances so that voltage distribution is uniform

```
clear;
clc;
c=1;
w=1;
v=1;
c1=c*w*v/(4*w*v);
c2=2*w*v/(3*w*v);
c3=3*w*v/(2*w*v);
c4=4*w*v/(1*w*v);
mprintf("the capacitance are \nc1=%.2fC\nc2=%.2fC\
    nc3=%.1fC\nc4=%.0fC",c1,c2,c3,c4);
```

Scilab code Exa 4.8 find mutual capacitances of insulator discs

```
clear;
clc;
w=1;
c=1;
v=1;
y=5;
c2=w*c*v+(w*y*v);
c3=2*w*c+6*w*c;
c4=3*w*c+8*w*c;
```

```
mprintf("the capacitance is\nc2=%dC\nc3=%dC\nc3=%dC"
    ,c2,c3,c4);
```

Scilab code Exa 4.9 find ratio of capacitances of insulator to earth capacitance of insulator

```
clear;
clc;
v=1;
v1=.4*v;
v2=.6*v;
k=(v2-v1)/v1;
mprintf("the ratio of capacitance to insulator to
    the capacitance to earth=%d",1/k);
```

Chapter 5

Mechanical Design of Overhead Lines

Scilab code Exa 5.1 finding sag in different weather conditions

```
clear;
clc;

m=.847;
g=9.81;
dia=1.95e-2;
l=244;
T=3.56e4;
Th_ice=.96e-2;
F_wind= 382;
W_ice=8920;
h=7.62;
L=1.43;

//(a)
w=m*g;
S= (w*l*l)/(8*T);
mprintf("\n (a)Sag= %.2f m", S);

```

```
//(b)
D=dia+Th_ice+Th_ice;
Fw=F_wind * D;
Wice = W_ice * (%pi/4) * ((D*D)-(dia*dia));
F=((w+Wice)^2 + Fw^2)^.5;
s=(F*l*l)/(8*T);
a=atan(Fw/(w+Wice));
S2=s * cos (a);
mprintf("\n (b)Vertical Sag= %.2f m", S2);

//(c)
H=h+L+S2;
mprintf("\n (c)Height of lowest cross arm= %.2f m",
   H);
```

Scilab code Exa 5.2 clearance of line

```
clear;
clc;

l=336;
h1=33.6;
h2=29;
w=8.33;
T=3.34e4;

//(a)
lc=l+ (2*T * (h1-h2)/(w*l));
S=w*lc*lc/(8*T);
cl= h1-S;
mprintf("\n (a)Clearance= %.3f m", cl);

//(b)
d1=lc/2;
d2=l-d1;
```

```
mprintf("\n (b)Distance of point O from lower support = %.2f m", d2);
```

Scilab code Exa 5.3 height of mid point from ground

```
clear;
clc;
l=300;
h1=80;
h2=50;
w=8.28;
T=19620;

//(a)
lc=l+ (2*T * (h1-h2)/(w*l));
dOC=(lc/2)-l;
hOC=w*dOC*dOC/(2*T);
dOP=dOC + (l/2);
hOP=w*dOP*dOP/(2*T);
hPC=hOP-hOC;
hP=hPC+ h2;
mprintf("\nHeight of mid point P above C =%.3f m", hPC);
mprintf("\nHeight of mid point P above water level = %.3f m",hP);
```

Scilab code Exa 5.4 finding sag

```
clear;
clc;

m=2.292;
g=9.81;
```

```
l=152;
Pw=39.063;
dia=2.068e-2;
ar=3.065;
stress=1054.63;

//(a)
w=m*g;
T=stress * ar *g;
Fw=Pw* g * dia;
Ft= sqrt(w^2 + Fw^2);
S=Ft * l *l /(8* T);
mprintf("\n Sag= %.2f m", S);
Sy= S * w/Ft;
mprintf("\n Vertical Component of Sag= %.3f m", fix(
    Sy*1000)/1000);
```

Scilab code Exa 5.5 finding minimum clearance and position of clearance point

```
clear;
clc;

h1=55;
h2=50;
l=300;
g=9.81;
T=2000*g;
m=.85;
w=m*g;

//(a)
lc=l+ (2*T * (h1-h2)/(w*l));
S= w * lc *lc /(8*T);
cl=h1-S;
```

```
mprintf("\n (a)Minimum Clearance between conductor
    and water= %.2f m", cl);

//(b)
dOB=lc/2;
dOA=l-dOB;
mprintf("\n (b)Distance of point O from lower
    support= %.1f m", dOA);
```

Scilab code Exa 5.6 find sag and tension under erection conditions

```
clear
clc

safety=2
d=1.95e-2
A=2.25e-4
E=91.4 *1e9
alpha=18.44 *1e-6
Temp21=10
Temp22=40
Tmax=77900
w=8.31
span=250


Fw=378 * d
Fw=round(Fw*100)/100
Ft1=sqrt(w^2 + Fw^2)
T1=Tmax/safety
Ft2=w


c_1=1
c_2=T1 -(alpha * A * E * (Temp22-Temp21)) - A*E*Ft1
```

```
^2 * span^2 /(24*T1^2)
c_3=0
c_4=A*E*Ft2^2 * span^2 /24
pol=poly([-c_4 -c_3 -c_2 c_1], "xx", "c")
T2s=roots(pol)

T2=T2s(1)
T2=round(T2)
Sag1= w * span *span / (8 * T2)

//difference in results is seen as the author has used hit and trial aproach to solve T2, while the program uses iterative method to solve equations. The equations have the same coefficients
mprintf("sag at erection= %.2f m", Sag1)

disp("difference in results is seen as the author has used hit and trial aproach to solve T2, while the program usesiterative method to solve equations. The equations have the same coefficients")
```

Scilab code Exa 5.7 representing line as parabola and catenary

```
clear;
clc;

l=600;
wc=12;
wi=14;
T=50000;

//(a)
F=wc+wi;
S= F * l * l/ (8*T);
```

```
mprintf("\n (a)Sag when representing line as a
    parabola= %.1f m", S);

//(b)
S=(F * l * l/ (8*T)) * (1+ ((l*l/48)*(F *F/(T*T))));
mprintf("\n (b)Sag when representing line as a
    catenary= %.3f m", S);
```

Scilab code Exa 5.8 galloping and dancing conductors find clearance under ice and air conditions

```
clear;
clc;

h1=75;
h2=45;
l=300;
g=9.81;
T=2500*g;
m=.9;
w=m*g;

//(a)
lc=l+ (2*T * (h1-h2)/(w*l));
dOC=(lc/2)-l;
hCO=w * dOC *dOC / (2*T);
dOP=dOC+(l/2);
hPO=w * dOP *dOP / (2*T);

hPC=hPO-hCO;
mprintf("\nHeight of mid point P above C =%.2f m",
   hPC);
hP=hPC+h2;
mprintf("\nHeight of mid point P above water level =
   %.2f m",hP);
```

Scilab code Exa 5.9 galloping and dancing conductors find clearance under no ice and air conditions

```
clear;
clc;

l=244;
m=.847;
g=9.81;
w=m*g;
T=3.56e4;

L=l*(1+ ((w*w * l*l)/(24* T *T)));
mprintf("length of conductor between 2 towers = %.3f
    m", L);
```

Scilab code Exa 5.10 find maximum sag under given condition

```
clear;
clc;

ar=.484;
d=.889e-2;
m=428e-3;
g=9.81;
T=1973 *g;
sfac=2;
w=m*g;
l=200;
T=T/sfac;

```

```
//(a)
S=w* l*l/(8*T);
mprintf("\n (a)maximum sag due to copper weight = %.3f m",S);

//(b)
t=1e-2;
D=d+(2*t);
wi=8920 * (D^2 - d^2) * %pi /4;

F=w+wi;
S=F* l*l/(8*T);
mprintf("\n (b)maximum sag due to addition weight of ice = %.1f m",S);
```

Chapter 6

Corona

Scilab code Exa 6.1 Finding local and general visual and disruptive corona voltage

```
clear;
clc;

dia=22.26e-3;
r=dia/2;
V=220e3;
d=6;
mvg=.82;
mvl=.72;
temp=25;
P=73;
m0=.84;


del=3.86*P/(273+temp);
Vd=(3e6/sqrt(2))*r*del*m0* log(d/r) *1e-3;
mprintf("\nDisruptive critical voltage = %.0f KV/
    phase", Vd)

Vvl=(3e6/sqrt(2))*r*del*mvl* log(d/r)* (1+(.03/sqrt(
```

```
    del*r))) *1e-3;
mprintf("\nVisual local voltage = %.1f KV/phase",Vvl)

Vvg=(3e6/sqrt(2))*r*del*mvg* log(d/r)* (1+(.03/sqrt(del*r))) *1e-3;
mprintf("\nVisual general voltage = %.1f KV/phase",Vvg)
```

Scilab code Exa 6.2 Finding total loss in fair weather and bad weather using peeks formula

```
clear;
clc;

dia=22.26e-3;
r=dia/2;
V=220;
d=6;
temp=25;
P=73;
m0=.84;
f=50;
l=250;

V=round(V/sqrt(3));
del=round((3.86*P/(273+temp))*10000)/10000;
Vd=round((3e6/sqrt(2))*r*del*m0* log(d/r) *1e-3);

//(a)Good Weather
Pc=243.5 * ((f+25)/del) * sqrt(r/d)* (V-Vd)^2 * 1e-5;
PC= Pc * l;
Ptot= 3 *PC;
mprintf("\ntotal loss in good weather = %.2f kW",
```

```
Ptot);

//(b)Bad Weather
Vd1=.8*Vd;
Pc1=243.5 * ((f+25)/del) * sqrt(r/d)* (V-Vd1)^2 * 1e
    -5;
PC1= Pc1 * l;
Ptot1= 3 *PC1;
mprintf("\ntotal loss in bad weather = %.0f kW",
    Ptot1);
```

Scilab code Exa 6.3 finding visual corona voltage

```
clear;
clc;

dia=1.04e-2;
r=dia/2;
m=.85;
d=2.44;
P=74;
temp=21;

del=round((3.86*P/(273+temp))*1000)/1000;

Vv=(3e6/sqrt(2))*r*del*m* log(d/r)* (1+(.03/sqrt(del
    *r))) *1e-3;
mprintf("\nVisual local voltage = %.2f KV/phase", Vv
    )
```

Scilab code Exa 6.4 finding minimum distance between conductors to limit disruptive corona

```
clear;
clc;

dia=30e-3;
r=dia/2;
del=.95;
m0=.95;
Vd=230e3;
Vd=Vd/sqrt(3);

x=round(Vd*100/((3e6/sqrt(2))* r * del * m0))/100;
d= exp(x) * r;
mprintf("\n minimum spacing between conductors = %.2
   f m", d);
```

Chapter 7

Interference Between Power and Communication Lines

Scilab code Exa 7.1 finding magnitude of voltage induced in telephone line due to EMI of power line

```
clear;
clc;

D1=1.2
D2=.6;
h=1;
H=10

DaP=sqrt((D1-(D2/2))^2+1)
DcQ=DaP
DbP=sqrt(((D2/2))^2+1)
DbQ=DbP
DcP=sqrt((D1+(D2/2))^2+1)
DaQ=DcP

Ia=300*exp(%i * 0);
Ib=300*exp(%i *-2* %pi/3);
Ic=300*exp(%i *2* %pi/3);
```

```
si=2e-7* ((Ia * log(DaQ/DaP))+(Ib * log(DbQ/DbP))+(
    Ic * log(DcQ/DcP)))

V=2*%pi * 50* abs(si);

mprintf("voltage induced in telephone line = %.2f e
    -3V/m", V*1e3)
```

Scilab code Exa 7.2 finding magnitude of voltage induced in telephone line due to EMI of power line under fault

```
clear;
clc;

D1=1.2
D2=.6;
h=1;
H=10

DaP=sqrt((D1-(D2/2))^2+h^2)
DaQ=sqrt((D1+(D2/2))^2+h^2)

Ia=2000*exp(%i * 0);
si=2e-6* ((Ia * log(DaQ/DaP)))

V=2*%pi * 50* abs(si);

mprintf("voltage induced in telephone line = %.3f V/
    m", V)
```

Scilab code Exa 7.3 potential of conductor due electrostatic effect

```
clear;
clc;

D1=1.2
D2=.6;
h=1;
H=10

DaP=sqrt((D1-(D2/2))^2+1)
DcQ=DaP
DbP=sqrt(((D2/2))^2+1)
DbQ=DbP
DcP=sqrt((D1+(D2/2))^2+1)
DaQ=DcP

dia=14.15e-3;
r=dia/2;

Va=11e3*exp(%i * 0)/sqrt(3);
Vb=11e3*exp(%i *-2* %pi/3)/sqrt(3);
Vc=11e3*exp(%i *2* %pi/3)/sqrt(3);

Vpa=Va * (log(((2*H) - DaP)/DaP)/log(((2*H) - r)/r))
Vpb=Vb * (log(((2*H) - DbP)/DbP)/log(((2*H) - r)/r))
Vpc=Vc * (log(((2*H) - DcP)/DcP)/log(((2*H) - r)/r))

Vp=Vpa+Vpb+Vpc;
mprintf("Potential of P= %.0f V", abs(Vp))
```

Scilab code Exa 7.4 Voltage induced in telephone conductor due electrostatic effect

```
clear;
clc;

```

```
f=50;
d=9e-3;
l=3.5;
h=16;
dc=.5;
hp=4

ha= round((sqrt(3)*l/2)*100)/100;
DaP=ha+hp;
DaQ=DaP+dc;

DbP=round(sqrt((l/2)^2+hp^2) *100)/100
DcP=DbP
DbQ=round(sqrt((l/2)^2+(hp+dc)^2) *100)/100
DcQ=DbQ

Ia=200*exp(%i * 0);
Ib=200*exp(%i *-2* %pi/3);
Ic=200*exp(%i *2* %pi/3);

si=2e-7* ((Ia * log(DaQ/DaP))+(Ib * log(DbQ/DbP))+(Ic * log(DcQ/DcP)))

V=2*%pi * 50* abs(si);

mprintf("voltage induced in telephone line = %.3f e-3V/m", V*1e3)
```

Scilab code Exa 7.5 Voltage induced in conductor due electrostatic effect

```
clear;
clc;

f=50;
dia=9e-3;
```

```
l=3.5;
h=16;
dc=.5;
hp=4

ha= round((sqrt(3)*l/2)*100)/100;
DaP=ha+hp;
DaQ=DaP+dc;
H=ha+h;

DbP=round((sqrt((l/2)^2+hp^2))*100)/100;
DcP=DbP
DbQ=round((sqrt((l/2)^2+(hp+dc)^2))*100)/100;
DcQ=DbQ

r=dia/2;

Va=132e3*exp(%i * 0)/sqrt(3);
Vb=132e3*exp(%i *-2* %pi/3)/sqrt(3);
Vc=132e3*exp(%i *2* %pi/3)/sqrt(3);

Vpa=Va * (log(((2*H) - DaP)/DaP)/log(((2*H) - r)/r))
Vpb=Vb * (log(((2*h) - DbP)/DbP)/log(((2*h) - r)/r))
Vpc=Vc * (log(((2*h) - DcP)/DcP)/log(((2*h) - r)/r))

Vp=Vpa+Vpb+Vpc;
mprintf("Potential of P= %.0f V", fix(abs(Vp)))
```

Chapter 8

Underground Cables

Scilab code Exa 8.1 inductance of a 3 core belted cable

```
clear;
clc;

n=37;
r=.238
t=.5;

r1=r*3.5;
GMR=.7788*r1;
D= 2*(r1+t);
L=.4605 * log10(D/GMR);
mprintf("\n L= %.4f mH/km/conductor", fix(L*1e4)*1e
    -4);
```

Scilab code Exa 8.2 find most economical diameter of cable so that it not exceed max stress

```
clear;
```

```
clc;

V=33;
V=V/sqrt(3);
T=35;

x=1;
r=V/(T * x);
R= %e * r;
t=R-r;

mprintf("\nMost economical conductor radius = %.3f
    cm", fix(R*1000)/1000);
mprintf("\nInsulation Thickness = %.3f cm", t);
```

Scilab code Exa 8.3 find most economical diameter of cable so that it not exceed max stress

```
clear;
clc;

V=132;
V=V/sqrt(3);
T=60;

V=V*sqrt(2);
x=1;
r=V/(T * x);
R= %e * r;
t=R-r;

mprintf("\nMost economical conductor diameter = %.2f
    cm", fix(2*R*100)/100);
mprintf("\nOverall diameter of insulation = %.3f cm"
    , fix(2*r*1000)/1000);
```

Scilab code Exa 8.4 find postitions of intersheaths max min stress and voltage on intersheaths

```
clear;
clc;

V=66;
V=V/sqrt(3);
T=60;
d=2;
r=d/2;
D=5.3;
R=D/2;

V=fix(V*sqrt(2) *10)/10;

a=(R/r)^(1/3);
a=round(a*1e3)/1e3;
r1=a*r;
r2=round(a*r1*1000)/1000;
V1= V *((1/r)-(1/r2)) *r ;
V2= (V-( V1* ((1/r)-(1/r1))) )*r1;
//An error exists in the text book while calculating V1 and V2 and hence Emax and Emin

Emax= (V-V1)/(r * log (a));
Emin= (V-V1)/(r1 * log (a));
mprintf("\n When intersheaths are used:\n Emax= %.2f kV/cm \t Emin= %2f kV/cm",Emax,Emin);
mprintf("\n Peak voltages at intersheaths:\n V1= %.2f kV \t V2= %.2f kV\n",V1,V2);
disp("An error exists in the text book while calculating V1 and V2 and hence Emax and Emin")

```

```
Emax= (V)/(r * 3*log (a));
Emin= (V)/(R * 3*log (a));
mprintf("\n When intersheaths are not used:\n Emax=
    %.1f kV/cm \t Emin= %.2f kV/cm",Emax,Emin);
```

Scilab code Exa 8.5 radius voltage of intersheath and ratio of maximum stress with and wothout intersheath

```
clear;
clc;

//solving for prt (c) only

R=3;
r=1;
V=60;


a=sqrt(R/r);
r1=a*r;

mprintf("\n (c) r1= %.3f cm", r1);

V1=V*((a)/(1+a));
mprintf("\n     V1 voltage at intersheath= %.2f cm",
     V1);

b=2/(1+a);
mprintf("\n     ratio of max electric stress with
    and without intersheath= %.3f cm", b);
```

Scilab code Exa 8.6 find maximum voltage in a cable having 2 insulation materials

```
clear;
clc;

r=0.5;
R=2.5;
Vga=60;
ea=4;
eb=2.5;
Vgb=50;

v=Vga*ea*r;

r1=v/(eb*Vgb);

V=v *((log(r1/r)/ea)+(log(R/r1)/eb));

mprintf("Maximum working voltage = %.2f kV", V);
```

Scilab code Exa 8.7 parameters of underground feeder

```
clear;
clc;

V=33e3;
V=V/sqrt(3);
f=50;
l=3.4e3;
d=2.5
r=d/2;
t=.6;
R=r+t;
e=3.1;

//disp(R);

```

```
c=2* %pi * 8.85 * e/ log(R/r);
C=c*l;
C=C*1e-6;
mprintf("\n(a)C= %.3f e-6 F/phase", C);

I=V * 2 * %pi * f * C*1e-6 ;
mprintf("\n(b)Charging Current I= %.2f A/phase", I);

KVAR=3*V*I *1e-3;
mprintf("\n(c)Charging KVAR= %.1f ", KVAR);

pf=.03;
phi=acos(pf);
lossang=(%pi/2)-phi;
Loss=2*%pi*f*C*1e-6*V*V*sin(lossang);
mprintf("\n(d)Dielectric Loss per phase= %.1f W", Loss);

Emax=V*1e-3/(r*log(R/r));
mprintf("\n(e)Emax= %.2f KV/cm (rms)",Emax)
```

Scilab code Exa 8.8 effective capacitance in cables

```
clear;
clc;

V=11e3;
V=V/sqrt(3);
f=50;
C1=.65;
C2=.75;

Cs=C2/3;
Cc=(C1/2)-(C2/6);
C=Cs+(3*Cc);
```

```
mprintf("\n(a)effective capacitance of each of the core to neutral = %.2f e-6 F",C);

I=V * 2 * %pi * f * C*1e-6 ;
mprintf("\n(b)Charging Current I= %.3f A/phase", I);

Cap=(1.5*Cc) + (.5 * Cs);
mprintf("\n(c)Capacitance between any 2 conductors = %.3f e-6 F",Cap);
```

Scilab code Exa 8.9 find current rating of cable

```
clear;
clc;

T2=65;
T1=20;
n=3;
restivity=.02826;
A=400;


R=restivity/A;

Eff_SnL=2/100;
Rdc20= (1+Eff_SnL)*(1+Eff_SnL)*R;

a=0.004
Rdc65=Rdc20 * (1+(a*(T2-T1)));

Eff_Skin_Prox=3.5/100;
Rac= (1+Eff_Skin_Prox)*(1+Eff_Skin_Prox)*Rdc65;

ti_core=2.7;
ti_belt=1.2;
```

```
tins=ti_belt+ti_core;

r=13;
ratio_tins_dia=tins/(2*r);

Gi3=.65;

Gi=Gi3/3;

R1=35.2e-3;
R2=40e-3;
h=.75
g1=5;
g2=1.5;
Gp=g1*log(R2/R1)/(2*%pi);
Gs=g2*log((2*h)/R2)/(2*%pi);

sheathlosses=.1;

I=((T2-T1)/(n* Rac * (Gi + ((1+sheathlosses)*(Gp+Gs)
    ))))^.5
mprintf("Current Rating =%d Amperes",I);
```

Chapter 9

Load Flow Studies

Scilab code Exa 9.1 form Y bus

```
clear;
clc

y=[0 1 (-.4*%i)
0 2 (-.3*%i)
1 2 (-.5*%i)]
n=2
e=3

Y=zeros(n,n)

for i=1:e
    n1=real(y(i,1))
    n2=real(y(i,2))
    ynew=y(i,3)
    if(n1==0)
        Y(n2,n2)=Y(n2,n2)+ynew
    else
        Y(n1,n1)= Y(n1,n1)+ynew
        Y(n1,n2)= Y(n1,n2)-ynew
        Y(n2,n1)= Y(n2,n1)-ynew
```

```
        Y(n2,n2)= Y(n2,n2)+ynew
    end
end
disp(Y)
```

Scilab code Exa 9.3 form Y bus and effect of adding a line

```
clear
clc

data=[ 1 2 .025 .1
 2 3 .02 .08
  3 4 .05 .2
   1 4 .04 .16
]

n=4
e=4
Y=zeros(n,n)

for i=1:e
    ynew = 1/(data(i,3)+(%i *real(data(i,4))))
    n1=real(data(i,1))
    n2=real(data(i,2))
    if(n1==0)
        Y(n2,n2)=Y(n2,n2)+ynew
    else
        Y(n1,n1)= Y(n1,n1)+ynew
        Y(n1,n2)= Y(n1,n2)-ynew
        Y(n2,n1)= Y(n2,n1)-ynew
        Y(n2,n2)= Y(n2,n2)+ynew
    end
end

disp(round(Y*1e2)/100,"(a)")
```

```
data(e+1,:)=[1 3 .1 .4]
i=e+1
ynew = 1/(data(i,3)+(%i *real(data(i,4))))

n1=real(data(i,1))
n2=real(data(i,2))
if(n1==0)
    Y(n2,n2)=Y(n2,n2)+ynew
else
    Y(n1,n1)= Y(n1,n1)+ynew
    Y(n1,n2)= Y(n1,n2)-ynew
    Y(n2,n1)= Y(n2,n1)-ynew
    Y(n2,n2)= Y(n2,n2)+ynew
end


disp(round(Y*1e2)/100,"(b)")
```

Scilab code Exa 9.4 find y bus

```
clear
clc

data=[ 1 2 .2+%i*.8 %i*.02
2 3 .3+%i*.9 %i*.03
2 4 .25+%i*1 %i*.04
3 4 .2+%i*.8 %i*.02
1 3 .1+%i*.4 %i*.01
]

n=4
e=5
Y=zeros(n,n)

```

```
for i=1:e
    ynew = 1/(data(i,3))
    y0=data(i,4)
    n1=real(data(i,1))
    n2=real(data(i,2))
    if(n1==0)
        Y(n2,n2)=Y(n2,n2)+ynew+y0
    else
        Y(n1,n1)= Y(n1,n1)+ynew+y0
        Y(n1,n2)= Y(n1,n2)-ynew
        Y(n2,n1)= Y(n2,n1)-ynew
        Y(n2,n2)= Y(n2,n2)+ynew+y0
    end
end

disp(fix(Y*1e3)/1e3)
```

Scilab code Exa 9.5 find missing elements of y bus

```
clear
clc

Y=[
.7-%i*3      -.2+%i    -.5+2*%i            %inf
%inf                  %inf     -.3+2*%i        -.5+3*%i
%inf                  %inf         %inf            -1+4*%i
%inf                  %inf         %inf            %inf
]
disp("inf shows that this value is to be found ")
disp(Y,"given")

Y(1,4)=round(Y(1,1)+Y(1,3)+Y(1,2))
Y(4,4)=0-Y(1,4)-Y(2,4)-Y(3,4)
Y(4,1)=Y(1,4)
```

```
Y(2,1)=Y(1,2)
Y(3,2)=Y(2,3)
Y(3,1)=Y(1,3)
Y(4,2)=Y(2,4)
Y(4,3)=Y(3,4)
Y(2,2)=0-Y(2,1)-Y(2,4)-Y(2,3)
Y(3,3)=0-Y(3,1)-Y(3,4)-Y(3,2)



disp(Y,"completed")
```

Scilab code Exa 9.7 find y bus with mutual coupling of lines present

```
clear
clc

z=[
%i*1     0    0    0    0
0        %i*.4    %i*.2  0  0
0        %i*.2    %i*.5  0  0
   0    0    0  %i*.2     0
0    0    0    0  %i*.25
]

y=inv(z)

A=[
1  0  0  0
-1  1  0  0
-1  0  1  0
0  0  1  -1
0  -1  0  -1
]

```

```
Y=A'*y*A

disp(Y)
```

Scilab code Exa 9.8 find reactive power generations losses and powers transferred

```
clear
clc

G=[
%inf %inf 1.5 1
0    %inf 2.0 .5
3.5  %inf 0   0
0    %inf 2   .8
]


ld=[
1 2 %i*.2
2 3 %i*.1
3 4 %i*.1
1 4 %i*.2
1 3 %i*.2
]

n=4
e=5
Y=zeros(n,n)

for i=1:e
    ynew = 1/(ld(i,3))
    y0=0
    n1=real(ld(i,1))
    n2=real(ld(i,2))
```

```
    if(n1==0)
        Y(n2,n2)=Y(n2,n2)+ynew+y0
    else
        Y(n1,n1)= Y(n1,n1)+ynew+y0
        Y(n1,n2)= Y(n1,n2)-ynew
        Y(n2,n1)= Y(n2,n1)-ynew
        Y(n2,n2)= Y(n2,n2)+ynew+y0
    end
end

G(1,1)= G(1,3)+G(2,3)+G(3,3)+G(4,3)-(G(2,1)+G(3,1)+G(4,1))

G(1,5)=G(1,1)-G(1,3)
G(2,5)=G(2,1)-G(2,3)
G(3,5)=G(3,1)-G(3,3)
G(4,5)=G(4,1)-G(4,3)

//d1=0
A=%i * [
Y(2,2) Y(2,3) Y(2,4)
Y(3,2) Y(3,3) Y(3,4)
Y(4,2) Y(4,3) Y(4,4)
]
d(1)=0
d(2:4)=inv(A) * [G(2,5) G(3,5) G(4,5)]'
d=round((d)*1000)/1000
G(1,6)= %i *((Y(1,1) * cos(d(1)-d(1)))+(Y(1,2) * cos(d(1)-d(2)))+(Y(1,3) * cos(d(1)-d(3)) )+(Y(1,4) *  cos(d(1)-d(4))));
G(2,6)= %i *((Y(2,1) * cos(d(2)-d(1)))+(Y(2,2) * cos(d(2)-d(2)))+(Y(2,3) * cos(d(2)-d(3)) )+(Y(2,4) *  cos(d(2)-d(4))));
G(3,6)= %i *((Y(3,1) * cos(d(3)-d(1)))+(Y(3,2) * cos(d(3)-d(2)))+(Y(3,3) * cos(d(3)-d(3)) )+(Y(3,4) *  cos(d(3)-d(4))));
G(4,6)= %i *((Y(4,1) * cos(d(4)-d(1)))+(Y(4,2) * cos(d(4)-d(2)))+(Y(4,3) * cos(d(4)-d(3)) )+(Y
```

```
(4,4) *  cos(d(4)-d(4))));

G(:,6)=round(G(:,6) *10000)/10000
G(:,2)=G(:,6) + G(:,4)
G(:,2)=round(G(:,2)*1e4)/1e4
mprintf("\n(a) Q1= %.4f, Q2= %.4f, Q3= %.3f, Q4= %.4f, ", G(1,2),G(2,2),G(3,2),G(4,2))

Ql=G(1,6)+G(2,6)+G(3,6)+G(4,6)
mprintf("\n(b) reactive line losses=%.4fpu",Ql)

X=ld(:,3)/%i

P(1)=round((sin(d(1)-d(2))/X(1))*1000)/1000
P(2)=round((sin(d(2)-d(3))/X(2))*1000)/1000
P(3)=round((sin(d(3)-d(4))/X(3))*1000)/1000
P(4)=round((sin(d(1)-d(4))/X(4))*1000)/1000
P(5)=round((sin(d(1)-d(3))/X(5))*1000)/1000

Q(1)=round(((1-cos(d(1)-d(2)))/X(1))*10000)/10000
Q(2)=round(((1-cos(d(2)-d(3)))/X(2))*10000)/10000
Q(3)=round(((1-cos(d(3)-d(4)))/X(3))*10000)/10000
Q(4)=round(((1-cos(d(1)-d(4)))/X(4))*10000)/10000
Q(5)=round(((1-cos(d(1)-d(3)))/X(5))*10000)/10000

mprintf("\n(c)")
mprintf("\nP12 = -P21 = %.3f, Q12=Q21=%.4f",P(1),Q(1))
mprintf("\nP23 = -P32 = %.3f, Q23=Q32=%.3f",P(2),Q(2))
mprintf("\nP34 = -P43 = %.3f, Q34=Q43=%.3f",P(3),Q(3))
mprintf("\nP14 = -P41 = %.3f, Q14=Q41=%.4f",P(4),Q(4))
mprintf("\nP13 = -P31 = %.3f, Q13=Q31=%.4f",P(5),Q(5))
```

Scilab code Exa 9.9 solve using gauss seidel for 1 variable

```
clear
clc

e=1e-5
x=1
E=100
while (E>e)
    x1=.5*(7+log10(x))
    E=abs(x-x1)
    x=x1
end

mprintf("x = %f",x)
```

Scilab code Exa 9.10 solve using gauss seidel for 2 variables

```
clear
clc

x=.5
y=.5
for i=1:4
    x=(.7*sin(x))+(.2*cos(y))
    y=(.7*cos(x))-(.2*sin(y))

end

mprintf("x= %f,  y=%f",x,y)
```

Scilab code Exa 9.11 find bus voltage and load angle using GS

```
clear
clc

ey=1/(.05 + %i*.15)
y=[
1 2 ey
1 5 ey
2 5 ey
2 3 ey
3 4 ey
4 5 ey
]
n=5
e=6

Y=zeros(n,n)

for i=1:e
    n1=real(y(i,1))
    n2=real(y(i,2))
    ynew=y(i,3)
    if(n1==0)
        Y(n2,n2)=Y(n2,n2)+ynew
    else
        Y(n1,n1)= Y(n1,n1)+ynew
        Y(n1,n2)= Y(n1,n2)-ynew
        Y(n2,n1)= Y(n2,n1)-ynew
        Y(n2,n2)= Y(n2,n2)+ynew
    end
end
// bus no | PL | QL | PG | QG | V | th | btype |
    Qmin | Qmax
```

```
data = [
1 1 .5 %inf %inf 1.02 0 1 %inf %inf
2 0 0 2 %inf 1.02 0 2 .2 .6
3 .5 .2 0 0 1 0   3 %inf %inf
4 .5 .2 0 0 1 0   3 %inf %inf
5 .5 .2 0 0 1 0   3 %inf %inf

]
disp(Y, "(a)")
j = data(:,1);     // Bus number.
PL = data(:,2);
QL = data(:,3);
PG = data(:,4);
QG = data(:,5);
V = data(:,6);
th = data(:,7);
btype = data(:,8);    // Type of Bus 1-Slack , 2-PV, 3-PQ.
Qmin = data(:,9);
Qmax = data(:,10);
n = max(j);
P = PG - PL;
Q = QG - QL

for i=1:n
    V(i,1)=V(i,1) * exp (%i * th(i,1)* %pi / 180)
end

Vprev = V;
toler = 1000;
iteration = 1;
while (iteration==1)
    for i = 2:n
        summ = 0;
        for k = 1:n
            if i ~= k
                summ = summ + Y(i,k)* V(k);
```

```
            end
        end
        if btype(i) == 2
            Q(i) = -imag(conj(V(i))*(summ + Y(i,i)*V(i)));
            if (Q(i) > Qmax(i)) | (Q(i) < Qmin(i))
                if Q(i) < Qmin(i)
                    Q(i) = Qmin(i);
                else
                    Q(i) = Qmax(i);
                end
                btype(i) = 3;
            end
        end
        V(i) = (1/Y(i,i))*(((P(i)- %i*Q(i))/conj(V(i))) - summ);
        if btype(i) == 2
            V(i) = abs(Vprev(i))*exp(%i * atan(imag(V(i))/real(V(i))));
        end
    end
    iteration = iteration + 1;
    toler = max(abs(abs(V) - abs(Vprev)));
    Vprev = V;
end
disp("(b)")
mprintf("\nV3 = %.2f ang(%.2f) deg", abs(V(3)),atand(imag(V(3))/real(V(3))))
mprintf("\nV4 = %.3f ang(%.2f) deg", abs(V(4)),atand(imag(V(4))/real(V(4))))
mprintf("\nV5 = %.4f ang(%.2f) deg", abs(V(5)),atand(imag(V(5))/real(V(5))))
mprintf("\ndelta 2 = %.2f deg",atand(imag(V(2,1))/real(V(2,1))))
mprintf("\nQ2 = %.4f ",Q(2,1))
```

Scilab code Exa 9.12 find bus voltage and load angle using GS minimum value of Q2 given

```
clear
clc

ey=1/(.05 + %i*.15)
y=[
1 2 ey
1 5 ey
2 5 ey
2 3 ey
3 4 ey
4 5 ey
]
n=5
e=6

Y=zeros(n,n)

for i=1:e
    n1=real(y(i,1))
    n2=real(y(i,2))
    ynew=y(i,3)
    if(n1==0)
        Y(n2,n2)=Y(n2,n2)+ynew
    else
        Y(n1,n1)= Y(n1,n1)+ynew
        Y(n1,n2)= Y(n1,n2)-ynew
        Y(n2,n1)= Y(n2,n1)-ynew
        Y(n2,n2)= Y(n2,n2)+ynew
    end
end
// bus no | PL | QL | PG | QG | V | th | btype |
```

```
    Qmin | Qmax

data = [
1 1 .5 %inf %inf 1.02 0 1 %inf %inf
2 0 0 2 %inf 1.02 0 2 .3 %inf
3 .5 .2 0 0 1 0  3 %inf %inf
4 .5 .2 0 0 1 0  3 %inf %inf
5 .5 .2 0 0 1 0  3 %inf %inf

]

j = data(:,1);     // Bus number.
PL = data(:,2);
QL = data(:,3);
PG = data(:,4);
QG = data(:,5);
V = data(:,6);
th = data(:,7);
btype = data(:,8);    // Type of Bus 1-Slack , 2-PV,
    3-PQ.
Qmin = data(:,9);
Qmax = data(:,10);
n = max(j);
P = PG - PL;
Q = QG - QL

for i=1:n
    V(i,1)=V(i,1) * exp (%i * th(i,1)* %pi / 180)
end

Vprev = V;
toler = 1000;
iteration = 1;
while (iteration==1)
    for i = 2:n
        summ = 0;
        for k = 1:n
            if i ~= k
```

```
                summm = summ + Y(i,k)* V(k);
            end
        end
        if btype(i) == 2
            Q(i) = -imag(conj(V(i))*(summ + Y(i,i)*V(i)));
            if (Q(i) > Qmax(i)) | (Q(i) < Qmin(i))
                if Q(i) < Qmin(i)
                    Q(i) = Qmin(i);
                else
                    Q(i) = Qmax(i);
                end
                btype(i) = 3;
            end
        end
        V(i) = (1/Y(i,i))*(((P(i)- %i*Q(i))/conj(V(i))) - summ);
        if btype(i) == 2
            V(i) = abs(Vprev(i))*exp(%i * atan(imag(V(i))/real(V(i))));
        end
    end
    iteration = iteration + 1;
    toler = max(abs(abs(V) - abs(Vprev)));
    Vprev = V;
end
V=round(V*1e3)/1e3
ansmat(:,1)=[1;2;3;4;5]
ansmat(:,2)=round(abs(V(:,1))*1000)/1000
for i=1:5
    ansmat(i,3)=round(atand(imag(V(i))/real(V(i)))*1e3)/1e3
end
disp(ansmat," bus no   |V|      delta")
```

Scilab code Exa 9.13 solve using newton raphson 1 variable

```
clear
clc
E=10
x=1
e=1e-5

while (E>e)
    f=(2*x) - (.43429 * log(x))-7
    df=2-(.43429 /x)
    x1=x-(f/df)
    E=abs(x1-x)
    x=x1
end
mprintf("x=%f",x)
```

Scilab code Exa 9.14 solve using newton raphson 2 variables

```
clear
clc

E=10
x=3.4
y=2.2
e=1e-5

while (E>e)
    X=[x;y]
    f=(x) + (3* .43429 * log(x))-y^2
    dfx=1 + (3* .43429 /x)
    dfy=-2*y
    g=(2*x*x) - (x*y)-(5*x)+1
    dgx=(4*x) - (y)-(5)
    dgy=-x
```

```
    J=[dfx dfy; dgx dgy]
    F=[f;g]
    X1=X-(inv(J)* F)
    E=max(abs(X1-X))
    x=X1(1,1)
    y=X1(2,1)
end
mprintf("x= %.4f,  y=%.4f", x,y)
```

Scilab code Exa 9.15 solve using newton raphson 1 variable

```
clear
clc
E=10
x=0
e=1e-4

while (E>e)
    f=(3*x) - (cos(x))-1
    df=3+sin(x)
    x1=x-(f/df)
    E=abs(x1-x)
    x=x1
end
mprintf("x=%.4f",x)
```

Scilab code Exa 9.17 solve system using newton raphson method

```
clear
clc

y=[
1 2 .026+%i*.11 %i*.04
```

```
2 3 .026+%i*.11 %i*.04
1 3 .026+%i*.11 %i*.04
]


n=3
e=3
Y=zeros(n,n)

for i=1:e
    ynew = 1/(y(i,3))
    y0=y(i,4)/2
    n1=real(y(i,1))
    n2=real(y(i,2))
    if(n1==0)
        Y(n2,n2)=Y(n2,n2)+ynew+y0
    else
        Y(n1,n1)= Y(n1,n1)+ynew+y0
        Y(n1,n2)= Y(n1,n2)-ynew
        Y(n2,n1)= Y(n2,n1)-ynew
        Y(n2,n2)= Y(n2,n2)+ynew+y0
    end
end
for i=1:n
    for j=1:n
        if i==j then
            Yb(i,j)= string(round(abs(Y(i,j))*1000)/1000) +'/_'+ string(round(atand(imag(Y(i,j))/real(Y(i,j)))*100)/100)
        else
            Yb(i,j)= string(round(abs(Y(i,j))*1000)/1000) +'/_'+ string(round((atand(imag(Y(i,j))/real(Y(i,j)))+180)*100)/100)
        end
    end
end
Y=round(Y*1e3)/1e3
```

```
disp(Yb,"(a)Ybus")
// bus no | PL | QL | PG | QG | V | th | btype

data = [
1 %inf %inf 1 .5  1.03 0 1
2 1.5 %inf  0 0 1.03 0 2
3 0 0  1.2 .5  1 0   3
]

j = data(:,1);     // Bus number.
PG = data(:,2);
QG = data(:,3);
PL = data(:,4);
QL = data(:,5);
V = data(:,6);
th = data(:,7)* %pi / 180;
btype = data(:,8);    // Type of Bus 1-Slack, 2-PV,
   3-PQ.
nbus = max(j);
P = PG - PL;
Q = QG - QL
Psp = P;
Qsp = Q;
G = real(Y);
B = imag(Y);
for i=1:nbus
    V(i,1)=V(i,1) * exp (%i * th(i,1))
end

pv = [2]         //whicih bus is PV
pq = [3]         //whicih bus is PQ
npv = length(pv);
npq = length(pq);

Tol = 1;
Iter = 1;
while (Iter == 1)

```

```
    P = zeros(nbus,1);
    Q = zeros(nbus,1);

    for i = 1:nbus
        for k = 1:nbus
            P(i) = P(i) + V(i)* V(k)*(G(i,k)*cos(th(i)-th(k)) + B(i,k)*sin(th(i)-th(k)));
            Q(i) = Q(i) + V(i)* V(k)*(G(i,k)*sin(th(i)-th(k)) - B(i,k)*cos(th(i)-th(k)));
        end
    end

    dPa = Psp-P;
    dQa = Qsp-Q;
    k = 1;
    dQ = zeros(npq,1);
    for i = 1:nbus
        if btype(i) == 3
            dQ(k,1) = dQa(i);
            k = k+1;
        end
    end
    dP = dPa(2:nbus);
    M = [dP; dQ];

    H = zeros(nbus-1,nbus-1);
    for i = 1:(nbus-1)
        m = i+1;
        for k = 1:(nbus-1)
            n = k+1;
            if n == m
                for n = 1:nbus
                    H(i,k) = H(i,k) + V(m)* V(n)*(-G(m,n)*sin(th(m)-th(n)) + B(m,n)*cos(th(m)-th(n)));
                end
                H(i,k) = H(i,k) - V(m)^2*B(m,m);
            else
```

```
                H(i,k) = V(m)* V(n)*(G(m,n)*sin(th(m)-th(n)) - B(m,n)*cos(th(m)-th(n)));
            end
        end
    end

    N = zeros(nbus-1,npq);
    for i = 1:(nbus-1)
        m = i+1;
        for k = 1:npq
            n = pq(k);
            if n == m
                for n = 1:nbus
                    N(i,k) = N(i,k) + V(n)*(G(m,n)*cos(th(m)-th(n)) + B(m,n)*sin(th(m)-th(n)));
                end
                N(i,k) = N(i,k) + V(m)*G(m,m);
            else
                N(i,k) = V(m)*(G(m,n)*cos(th(m)-th(n)) + B(m,n)*sin(th(m)-th(n)));
            end
        end
    end

    J3 = zeros(npq,nbus-1);
    for i = 1:npq
        m = pq(i);
        for k = 1:(nbus-1)
            n = k+1;
            if n == m
                for n = 1:nbus
                    J3(i,k) = J3(i,k) + V(m)* V(n)*(G(m,n)*cos(th(m)-th(n)) + B(m,n)*sin(th(m)-th(n)));
                end
                J3(i,k) = J3(i,k) - V(m)^2*G(m,m);
```

```
            else
                J3(i,k) = V(m)* V(n)*(-G(m,n)*cos(th(m)-th(n)) - B(m,n)*sin(th(m)-th(n)));
            end
        end
    end

    L = zeros(npq,npq);
    for i = 1:npq
        m = pq(i);
        for k = 1:npq
            n = pq(k);
            if n == m
                for n = 1:nbus
                    L(i,k) = L(i,k) + V(n)*(G(m,n)*sin(th(m)-th(n)) - B(m,n)*cos(th(m)-th(n)));
                end
                L(i,k) = L(i,k) - V(m)*B(m,m);
            else
                L(i,k) = V(m)*(G(m,n)*sin(th(m)-th(n)) - B(m,n)*cos(th(m)-th(n)));
            end
        end
    end

    J = [H N; J3 L];

    X = inv(J)*M;
    dTh = X(1:npq+npv);
    dV = X(nbus:nbus+npq-1);


    th(2:nbus) = dTh + th(2:nbus);
    k = 1;
    for i = 2:nbus
        if btype(i) == 3
```

```
            V(i) = dV(k) + V(i);
            k = k+1;
        end
    end

    Iter = Iter + 1;
    Tol = max(abs(M));

end
mprintf("\n\n(b)P2= %.3f, P2= %.3f, P3= %.3f, Q3= %
   .3f", P(2),Q(2) ,P(3), Q(3))
J=fix(J*1e3)/1e3
disp(J, "(c)J")

mprintf("\n(d)\n")
mprintf("%.3f = %.3f dd2 + %.3f dd3 +%.3f dV3/|V3|\n
   ", dP(1), J(1,1), J(1,2), J(1,3) )
mprintf("%.3f = %.3f dd2 + %.3f dd3 +%.3f dV3/|V3|\n
   ", dP(2), J(2,1), J(2,2), J(2,3) )
mprintf("%.3f = %.3f dd2 + %.3f dd3 +%.3f dV3/|V3|\n
   ", dQ(1), J(3,1), J(3,2), J(3,3) )
```

Scilab code Exa 9.18 solve system using fast decoupled method

```
clear
clc

y=[
1 2 %i*.11
2 3 %i*.11
1 3 %i*.11
]


n=3
```

```
e=3
Y=zeros(n,n)

for i=1:e
    ynew = 1/(y(i,3))
    n1=real(y(i,1))
    n2=real(y(i,2))
    if(n1==0)
        Y(n2,n2)=Y(n2,n2)+ynew
    else
        Y(n1,n1)= Y(n1,n1)+ynew
        Y(n1,n2)= Y(n1,n2)-ynew
        Y(n2,n1)= Y(n2,n1)-ynew
        Y(n2,n2)= Y(n2,n2)+ynew
    end
end
Y=round(Y*1e3)/1e3
disp(Y,"(a)Ybus")
// bus no | PL | QL | PG | QG | V | th | btype

data = [
1 %inf %inf 1 .5  1.03 0 1
2 1.5 %inf  0 0 1.03 0 2
3 0 0  1.2 .5  1 0   3
]

j = data(:,1);     // Bus number.
PG = data(:,2);
QG = data(:,3);
PL = data(:,4);
QL = data(:,5);
V = data(:,6);
th = data(:,7)* %pi / 180;
btype = data(:,8);    // Type of Bus 1-Slack, 2-PV, 3-PQ.
nbus = max(j);
P = PG - PL;
Q = QG - QL
```

```
Psp = P;
Qsp = Q;
G = real(Y);
B = imag(Y);
for i=1:nbus
    V(i,1)=V(i,1) * exp (%i * th(i,1))
end

pv = [2]        //whicih bus is PV
pq = [3]        //whicih bus is PQ
npv = length(pv);
npq = length(pq);

Tol = 1;
Iter = 1;
while (Iter == 1)

    P = zeros(nbus,1);
    Q = zeros(nbus,1);

    for i = 1:nbus
        for k = 1:nbus
            P(i) = P(i) + V(i)* V(k)*(G(i,k)*cos(th(
               i)-th(k)) + B(i,k)*sin(th(i)-th(k)));
            Q(i) = Q(i) + V(i)* V(k)*(G(i,k)*sin(th(
               i)-th(k)) - B(i,k)*cos(th(i)-th(k)));
        end
    end

    dPa = Psp-P;
    dQa = Qsp-Q;
    k = 1;
    dQ = zeros(npq,1);
    for i = 1:nbus
        if btype(i) == 3
            dQ(k,1) = dQa(i);
            k = k+1;
        end
```

```
    end
    dP = dPa(2:nbus);
    M = [dP; dQ];

    H = zeros(nbus-1,nbus-1);
    for i = 1:(nbus-1)
        m = i+1;
        for k = 1:(nbus-1)
            n = k+1;
            if n == m
                for n = 1:nbus
                    H(i,k) = H(i,k) + V(m)* V(n)*(-G(m,n)*sin(th(m)-th(n)) + B(m,n)*cos(th(m)-th(n)));
                end
                H(i,k) = H(i,k) - V(m)^2*B(m,m);
            else
                H(i,k) = V(m)* V(n)*(G(m,n)*sin(th(m)-th(n)) - B(m,n)*cos(th(m)-th(n)));
            end
        end
    end

    N = zeros(nbus-1,npq);
    for i = 1:(nbus-1)
        m = i+1;
        for k = 1:npq
            n = pq(k);
            if n == m
                for n = 1:nbus
                    N(i,k) = N(i,k) + V(n)*(G(m,n)*cos(th(m)-th(n)) + B(m,n)*sin(th(m)-th(n)));
                end
                N(i,k) = N(i,k) + V(m)*G(m,m);
            else
                N(i,k) = V(m)*(G(m,n)*cos(th(m)-th(n
```

```
)) + B(m,n)*sin(th(m)-th(n)));
            end
        end
    end

    J3 = zeros(npq,nbus-1);
    for i = 1:npq
        m = pq(i);
        for k = 1:(nbus-1)
            n = k+1;
            if n == m
                for n = 1:nbus
                    J3(i,k) = J3(i,k) + V(m)* V(n)*(G(m,n)*cos(th(m)-th(n)) + B(m,n)*sin(th(m)-th(n)));
                end
                J3(i,k) = J3(i,k) - V(m)^2*G(m,m);
            else
                J3(i,k) = V(m)* V(n)*(-G(m,n)*cos(th(m)-th(n)) - B(m,n)*sin(th(m)-th(n)));
            end
        end
    end

    L = zeros(npq,npq);
    for i = 1:npq
        m = pq(i);
        for k = 1:npq
            n = pq(k);
            if n == m
                for n = 1:nbus
                    L(i,k) = L(i,k) + V(n)*(G(m,n)*sin(th(m)-th(n)) - B(m,n)*cos(th(m)-th(n)));
                end
                L(i,k) = L(i,k) - V(m)*B(m,m);
            else
```

```
                L(i,k) = V(m)*(G(m,n)*sin(th(m)-th(n)) - B(m,n)*cos(th(m)-th(n)));
            end
        end
    end

    J = [H N; J3 L];

    X = inv(J)*M;
    dTh = X(1:npq+npv);
    dV = X(nbus:nbus+npq-1);


    th(2:nbus) = dTh + th(2:nbus);
    k = 1;
    for i = 2:nbus
        if btype(i) == 3
            V(i) = -dV(k) + V(i);
            k = k+1;
        end
    end

    Iter = Iter + 1;
    Tol = max(abs(M));

end

V=round(V*10000)/10000
th=round(th*1e5)/1e5

mprintf("\n(b)P2=%.0f P3=%.0f Q3=%.3f \n", P(2), P(3), Q(3))
disp(-imag(Y(2:3,2:3)), "(c)B''")
disp(-imag(Y(3,3)), "B''''")

P = zeros(nbus,1);
Q = zeros(nbus,1);

```

```
for i = 1:nbus
    for k = 1:nbus
        P(i) = P(i) + real(V(i)* V(k)*Y(i,k)* exp(%i
            * (th(k)-th(i))))
        Q(i) = Q(i) - imag(V(i)* V(k)*Y(i,k)* exp(%i
            * (th(k)-th(i))))
    end
end
//"please note: there is an error in the book in
   calculation of Q3=-.0497 (part(e)) during value
   substitution in formula vi. The variation in
   results is due to the same error, verified
   seperately on calculator"
mprintf("\n\n(e)P2= %.3f, P3= %.3f, Q3= %.3f", P(2),
    P(3), Q(3))

Q(3)=-.49
Pl=P(1)+P(2)+P(3)
Ql=Q(1)+Q(2)+Q(3)


mprintf("\n\n(f)real line losses= %.1f, reactive
   line losses= %.1f", Pl, Ql)


data(1,2)=PL(1)+P(1);
data(1,3)=QL(1)+Q(1);
data(2,3)=QL(2)+Q(2);


mprintf("\n\n(g)PG1= %.1f,QG1= %.2f, PG2= %.1f,QG2=
   %.2f\n\n", data(1,2), data(1,3),data(2,2), data
   (2,3))

disp("please note: there is an error in the book in
   calculation of Q3=-.0497 (part(e)) during value
   substitution in formula vi. The variation in
   results is due to the same error, verified
```

```
seperately on calculator")
```

Scilab code Exa 9.19 solve system using gauss seidel method with acceleration constant

```
clear
clc

y=[
1 2 2-%i*8
1 3 1-%i*4
2 3 .666-%i*2.664
2 4 1-%i*4
3 4 2-%i*8
]
n=max(real(y(:,1:2)))
e=5

Y=zeros(n,n)

for i=1:e
    n1=real(y(i,1))
    n2=real(y(i,2))
    ynew=y(i,3)
    if(n1==0)
        Y(n2,n2)=Y(n2,n2)+ynew
    else
        Y(n1,n1)= Y(n1,n1)+ynew
        Y(n1,n2)= Y(n1,n2)-ynew
        Y(n2,n1)= Y(n2,n1)-ynew
        Y(n2,n2)= Y(n2,n2)+ynew
    end
end

disp(Y,"Ybus")
```

```
// bus no | P | Q | V | th | btype

data = [
1 %inf %inf 1.06 0 1
2 .5 .2 1 0   3
3 .4 .3 1 0   3
4 .3 .1 1 0   3
]

j = data(:,1);     // Bus number.
P = data(:,2);
Q = data(:,3);
V = data(:,4);
th = data(:,5);
btype = data(:,6);    // Type of Bus 1-Slack, 2-PV, 3-PQ.
n = max(j);

a=1.6

for i=1:n
    V(i,1)=V(i,1) * exp (%i * th(i,1)* %pi / 180)
end

Vprev = V;
toler = 1000;
iteration = 1;
disp("(b)")
while (iteration==1)
    for i = 2:n
        summ = 0;
        for k = 1:n
            if i ~= k
                summ = summ + Y(i,k)* V(k);
            end
        end
        V(i) = (1/Y(i,i))*(((P(i)- %i*Q(i))/conj(V(i))) - summ);
```

```
        dv=V(i)-Vprev(i)
        mprintf("\nV%d = %.3f ang(%.2f) deg", i,abs(V(i)),atand(imag(V(i))/real(V(i))))
        V(i)=Vprev(i)+(a*dv)
        if btype(i) == 2
            V(i) = abs(Vprev(i))*exp(%i * atan(imag(V(i))/real(V(i))));
        end
    end
    iteration = iteration + 1;
    toler = max(abs(abs(V) - abs(Vprev)));
    Vprev = V;
end
```

Chapter 10

Balanced and Unbalanced Faults

Scilab code Exa 10.1 find fault current and fault level

```
clear
clc

Sg1=100
Vg1=11
xg1=.15 *%i

Sg2=50
Vg2=11
xg2=.1 *%i

St1=100
Vt1a=11
Vt1b=132
xt1=.1 *%i
nt1=Vt1b/Vt1a

St2=50
Vt2a=11
```

```
Vt2b=132
xt2=.08 *%i
nt2=Vt2b/Vt2a

Sb=100
Vb1=11
Vb2=nt1*Vb1

xl=.2 * 200 *%i
Xl=xl/(Vb2*Vb2/Sb)

Xg2=xg2*Sb/Sg2
Xt2=xt2 * Sb/St2

X=( ((xg1 +xt1) * (Xg2 +Xt2) )/ ((xg1 +xt1) + (Xg2 +
    Xt2) )) +Xl/2

I=1/X

Ib1=Sb*1e3/(Vb1*sqrt(3))
Ib2=Sb*1e3/(Vb2*sqrt(3))
If=abs(I*Ib2)
Ifg=abs(I*Ib1)
Ifg1=Ifg * (Xg2 +Xt2)/(xg1+xt1+Xt2+Xg2)
Ifg2=Ifg * (xg1 +xt1)/(xg1+xt1+Xt2+Xg2)
MVAf=abs(I*1*Sb)

mprintf("Total fault current = %.2f A, Fault Level=
    %f MVA,\n Fault current supplied by generator 1=
    %f A, generator 2=%f A",If, MVAf, Ifg1, Ifg2)
```

Scilab code Exa 10.2 find fault level and X to limit current during 3 phase fault

```
clear
```

```
clc

s=50
v=11
x=.15
S=50
V=11

Xe=.15/4
FMVA= round(10000/Xe)/10000
Fault=FMVA * S
mprintf("\n(a)fault level = %.3f pu", Fault)

sga=.5*Fault
sgb=800-sga

xb=.15/2
X=(S/sgb)-xb
x=X*(V*V/S)
mprintf("\n(b)X = %.3f ohms", x)
```

Scilab code Exa 10.3 find fault current and fault level during 3 phase fault

```
clear
clc

G=[100 11 .2]
T=[100 11 132 .05]
L=[
1.2e-3 100
1e-3 50
1e-3 50
]
B=[100 11]
T(5)= T(3)/T(2)
```

```
B(3)=B(2)* T(5)
B(4)=B(3)^2/B(1)
for(i=1:3)
    L(i,3)= 2*%pi*50* L(i,1)* L(i,2);
    L(i,4)= L(i,3)/B(4)
end
l=L(:,4)
X1=l(1)*l(2)/(l(1)+l(2)+l(3))
X2=l(3)*l(2)/(l(1)+l(2)+l(3))
X3=l(1)*l(3)/(l(1)+l(2)+l(3))
X=(((G(3)+T(4)+(X1))^-1) +  ((G(3)+T(4)+(X3))^-1))
   ^-1 +X2
SF=round(10/X)/10
S=SF* B(1)
IF=round(10/X)/10
I=IF*B(1)*1e6/(sqrt(3)*B(3)*1e3)
mprintf("Fault level = %.0f MVA, Fault current %.1f
   A", S,I)
```

Scilab code Exa 10.4 find subtransient currents in system

```
clear
clc

T=[10 132 6.6 .15]
M=[5 6.6 .3 .2 ]
B=[10 6.6]
T(5)= T(3)/T(2)
B(3)=B(2)* T(5)
B(4)= B(1)*1e6/(sqrt(3)*B(2)*1e3)
M(5)=M(4) *B(1)/M(1)
M(6)=M(3) *B(1)/M(1)

X1=1/((1/M(5))+(1/M(5))+(1/T(4)))
IF1=round(100/X1)/100
```

```
I1=IF1*B(4)
mprintf("\n(a) sub transient fault current=%.0f A", I1)

It=round(100/T(4))/100
Im=1/M(5)
ID=It+Im
iD=ID*B(4)
mprintf("\n(b) current through D=%.0f A", iD)

RD=iD*1.6
mprintf("\n(c) current rating of  D=%.0f A", RD)

X2=1/((1/M(6))+(1/T(4)))
IF2=round(100/X2)/100
I2=IF2*round(B(4)*10)/10
iCB=1.1 *I2
mprintf("\n(d) current to be interrrupted by D=%.1f A", fix(iCB*10)/10)
```

Scilab code Exa 10.5 calculate total generator and motor current in 3phase fault

```
clear
clc

G=[100 11 .25]
M=[50  11 .2 40 .8]
xl=.05
vt=10.95

B=[100 11]
B(3)= B(1)*1e6/(sqrt(3)*B(2)*1e3)

IL=M(4)*1e6 / (sqrt(3)*vt*1e3*M(5))
```

```
Il=round(((IL/B(3) * exp(%i * acos(M(5)))))*1000)/1000

Vt=round(1e3*vt/B(2))/1e3

V=fix((Vt + Il*xl*%i)*1e4)*1e-4

M(6)=M(3) *B(1)/M(1)

xth= round( G(3) * (M(6) +xl)/(G(3) + (M(6) +xl))*10000)/10000

If=fix(V*1e3/(%i *xth))/1e3
temp= fix(imag(If)*100)/100
If=complex(real(If), temp)
Ifg=fix((If * (M(6) +xl)/(G(3) + (M(6) +xl)))*100)/100
Ifm=round((If * (G(3)    )/(G(3) + (M(6) +xl)))*1000)/1000

Ig=(Ifg + Il) * B(3)
Im=(Ifm - Il) *B(3)

mprintf("\ntotal generator current during fault= %s A", string(round(abs(Ig)*10)/10) +'/_'+ string(round(atand(imag(Ig)/real(Ig))*10)/10))
mprintf("\ntotal motor current during fault= %s A", string(fix(abs(Im)*1)/1) +'/_'+ string(fix(atand(imag(Im)/real(Im))*10)/10 +180))
```

Scilab code Exa 10.6 find symmetrical components

```
clear
clc

```

```
I=[
5*exp(%i * %pi * 60 / 180)
5*exp(%i * %pi * -60 / 180)
0
]
a=exp(%i * 2 * %pi/3)
A=[1 1 1
1 a^2 a
1 a a^2
]

Is=inv(A)*I

mprintf("\nIa0= %s", string(round(abs(Is(1))*1000)/1000) +'/_'+ string(round(atand(imag(Is(1))/real(Is(1)))*100)/100) )
mprintf("\nIa1= %s", string(round(abs(Is(2))*1000)/1000) +'/_'+ string(round(atand(imag(Is(2))/real(Is(2)))*100)/100) )
mprintf("\nIa2= %s", string(round(abs(Is(3))*1000)/1000) +'/_'+ string(round(atand(imag(Is(3))/real(Is(3)))*100)/100 +180) )
```

Scilab code Exa 10.8 find zero sequence components

```
clear
clc

G=[50 11 0 .08
30 11 0 .07]
T=[50 11 220 .1
30 220 11 .09]

B=[50 11]

```

```
T(1,5)= T(1,3)/T(1,2)
T(2,5)= T(2,3)/T(2,2)
B(3)=B(2)* T(1,5)
B(4)=B(3)* T(2,5)

B(5)= B(3)^2/B(1)

Z=555.6
z=Z/B(5)

zt2=T(2,4) * B(1)/T(2,1)
zg2=G(2,4) * B(1)/G(2,1)

Zn=3
zn=Zn *3 / ( B(4)^2/B(1))

mprintf("zero seq netwk: xt1= %.1f, xt2=%.2f, Xg1=%
    .2f Xg2=%.3f, xl=%.3f, Zn=%.2fi",T(1,4), zt2, G
    (1,4), zg2, z,zn)
```

Scilab code Exa 10.10 find fault MVA and current and line to line voltages during fault

```
clear
clc

Z=[.2 .2 .05]' * %i
S=30
V=11
I=S*1e6/(sqrt(3)*V*1e3)
E=1

Ia1=E/(Z(1)+Z(2)+Z(3))
If= 3*abs(Ia1) * S*1e6/(sqrt(3)* V*1e3)
Ia2=Ia1
```

```
Ia0=Ia1

a=exp(%i * 2 * %pi/3)
A=[1 1 1
1 a^2 a
1 a a^2
]

Va1=E-Ia1*Z(1)
Va2=0-Ia2*Z(2)
Va0=0-Ia0*Z(3)

Vp=[ Va0 Va1 Va2]'
v=A*Vp

vab=v(1)-v(2)
vbc=v(2)-v(3)
vca=v(3)-v(1)

Vbll=V/sqrt(3)

Vab=vab * Vbll
Vbc=vbc * Vbll
Vca=vca * Vbll
Vll=[Vab Vbc Vca]
mprintf("\n(a)\n line currents: If= %f A, Line voltages in KV ", If)
mprintf("\nVab= %s", string(round(abs(Vll(1))*10)/10) +'/_'+ string(round(atand(imag(Vll(1))/real(Vll(1)))*10)/10) )
mprintf("\nVbc= %s", string(round(abs(Vll(2))*10)/10) +'/_'+ string(round(atand(imag(Vll(2))/real(Vll(2)))*10)/10 +180) )
mprintf("\nVca= %s", string(round(abs(Vll(3))*10)/10) +'/_'+ string(round(atand(imag(Vll(3))/real(Vll(3)))*10)/10 +180))
```

```

If3=E/Z(1)
if3=abs(If3)*  S*1e6/(sqrt(3)* V*1e3)

mprintf("\n(b)3 phase fault current is -%.0fi A",
   if3)
```

Scilab code Exa 10.11 thevinin equlvalent impedances of sequence networks as seen from fault point

```
clear
clc

X=[
 .25 .25 .05
 .2 .2 .05
 .06 .06 .06
 .07 .07 .07
 .1 .1 .3
 .1 .1 .3
]

B=[
    100 11
    100 11
    100 11
    100 11
    100 220
    100 220
    ]
V1=11
V2=220
S=100
Xe=3*.03
//end 9
```

```
X1=(((X(1,1)*B(1,1) *V1/(S*B(1,2))) +(X(3,1)*B(3,1) *V1/(S*B(3,2))) )^-1+((X(2,1)*B(2,1) *V1/(S*B(2,2)))+(X(4,1)*B(4,1) *V1/(S*B(4,2))) +((X(5,1)*B(5,1) *V2/(S*B(5,2)))^-1 +(X(6,1)*B(6,1) *V2/(S*B(6,2)))^-1)^-1)^-1)^-1

X2=(((X(1,2)*B(1,1) *V1/(S*B(1,2))) +(X(3,2)*B(3,1) *V1/(S*B(3,2))) )^-1+((X(2,2)*B(2,1) *V1/(S*B(2,2)))+(X(4,2)*B(4,1) *V1/(S*B(4,2))) +((X(5,2)*B(5,1) *V2/(S*B(5,2)))^-1 +(X(6,2)*B(6,1) *V2/(S*B(6,2)))^-1)^-1)^-1)^-1

X0=((X(3,3)*B(3,1) *V1/(S*B(3,2)))^-1 + ((Xe    *B(4,1) *V1/(S*B(4,2))) + (X(2,3)*B(2,1) *V1/(S*B(2,2))) + (X(4,3)*B(4,1) *V1/(S*B(4,2))) +(((X(5,3)*B(5,1) *V2/(S*B(5,2))))^-1 +(X(6,3)*B(6,1) *V2/(S*B(6,2)))^-1)^-1 )^-1 )^-1

Z1=%i * round(X1*1e3)/1e3
Z2=%i * round(X2*1e3)/1e3
Z0=%i * round(X0*1e3)/1e3

Z=[Z1 Z2 Z0]'

mprintf("\nZ1= %.3fj\n", imag(Z(1)))
mprintf("Z2= %.3fj\n", imag(Z(2)))
mprintf("Z0= %.3fj\n", imag(Z(3)))
```

Scilab code Exa 10.12 find fault current voltage at fault point and current and voltage at generator terminal during LG fault

```
clear
clc

X=[
```

```
 .25 .25 .05
 .2 .2 .05
 .06 .06 .06
 .07 .07 .07
 .1 .1 .3
 .1 .1 .3
]

B=[
    100 11
    100 11
    100 11
    100 11
    100 220
    100 220
    ]
V1=11
V2=220
S=100
Xe=3*.03
//end 9
X1=(((X(1,1)*B(1,1) *V1/(S*B(1,2))) +(X(3,1)*B(3,1) *V1/(S*B(3,2))) )^-1+((X(2,1)*B(2,1) *V1/(S*B(2,2)))+(X(4,1)*B(4,1) *V1/(S*B(4,2))) +((X(5,1)*B(5,1) *V2/(S*B(5,2)))^-1 +(X(6,1)*B(6,1) *V2/(S*B(6,2)))^-1)^-1)^-1)^-1

X2=(((X(1,2)*B(1,1) *V1/(S*B(1,2))) +(X(3,2)*B(3,1) *V1/(S*B(3,2))) )^-1+((X(2,2)*B(2,1) *V1/(S*B(2,2)))+(X(4,2)*B(4,1) *V1/(S*B(4,2))) +((X(5,2)*B(5,1) *V2/(S*B(5,2)))^-1 +(X(6,2)*B(6,1) *V2/(S*B(6,2)))^-1)^-1)^-1)^-1

X0=((X(3,3)*B(3,1) *V1/(S*B(3,2)))^-1 + ((Xe    *B(4,1) *V1/(S*B(4,2))) + (X(2,3)*B(2,1) *V1/(S*B(2,2))) + (X(4,3)*B(4,1) *V1/(S*B(4,2))) +(((X(5,3)*B(5,1) *V2/(S*B(5,2))))^-1 +(X(6,3)*B(6,1) *V2/(S*B(6,2)))^-1)^-1 )^-1 )^-1
```

```

Z1=%i * round(X1*1e3)/1e3
Z2=%i * round(X2*1e3)/1e3
Z0=%i * round(X0*1e3)/1e3
Z=[Z1 Z2 Z0]'
//end ques 11

a=exp(%i * 2 * %pi /3)
A= [1 1 1 ; 1 a^2 a; 1 a a^2]

//12(a)
Ia1a=1/(Z1+Z2+Z0)
Ia1a=round(Ia1a *1e2)/1e2
mIa1a=[ Ia1a Ia1a Ia1a]'
mIa=A*mIa1a
Ia=round(mIa(1)*100)/100
Iaa = round(abs(mIa(1))*100)/100
Iba=round(S*1e7/(sqrt(3) * V2 * 1e3))/10
IFa=round(Iba * Iaa *100)/100

mprintf("\n(a) Fault current = %.2f A",IFa)

//12(b)

Va1=round((1- (Z1 * Ia1a))*100)/100
Va2=round((0- (Z2 * Ia1a))*100)/100
Va0=round((0- (Z0 * Ia1a))*100)/100
mVa1=[ Va0  Va1 Va2]'
mVa=A*mVa1
v=mVa * V2 / sqrt(3)
v=round(v *10000)/10000

mprintf("\n\n(b)line to neutral voltages in KV  ")
mprintf("\nVa= %s", string(fix(abs(v(1))*100)/100)
   )
mprintf("\nVb= %s", string(fix(abs(v(2))*100)/100) +
   '/_'+ string(round(atand(imag(v(2))/real(v(2)))
   *100)/100 +180 ) )
```

```
mprintf("\nVc= %s", string(fix(abs(v(3))*100)/100) + '/_'+ string(round(atand(imag(v(3))/real(v(3)))*100)/100 +180))

//12(c) --- g2

Ia1g2= Ia1a * (((X(1,1)*B(1,1) *V1/(S*B(1,2))) +(X(3,1)*B(3,1) *V1/(S*B(3,2))) ))/((((X(1,1)*B(1,1) *V1/(S*B(1,2))) +(X(3,1)*B(3,1) *V1/(S*B(3,2))))+((X(2,1)*B(2,1) *V1/(S*B(2,2)))+(X(4,1)*B(4,1) *V1/(S*B(4,2))) +((X(5,1)*B(5,1) *V2/(S*B(5,2)))^-1 +(X(6,1)*B(6,1) *V2/(S*B(6,2)))^-1)^-1)))
Ia1g2=round(Ia1g2 *1e2)/1e2

Ia2g2= Ia1a * (((X(1,2)*B(1,1) *V1/(S*B(1,2))) +(X(3,2)*B(3,1) *V1/(S*B(3,2))) ))/((((X(1,2)*B(1,1) *V1/(S*B(1,2))) +(X(3,2)*B(3,1) *V1/(S*B(3,2))))+((X(2,2)*B(2,1) *V1/(S*B(2,2)))+(X(4,2)*B(4,1) *V1/(S*B(4,2))) +((X(5,2)*B(5,1) *V2/(S*B(5,2)))^-1 +(X(6,2)*B(6,1) *V2/(S*B(6,2)))^-1)^-1)))
Ia2g2=round(Ia2g2 *1e2)/1e2

Ia0g2= Ia1a * (((X(3,3)*B(3,1) *V1/(S*B(3,2))) ))/((((Xe *B(4,1) *V1/(S*B(1,2))) +(X(3,3)*B(3,1) *V1/(S*B(3,2))) )+(X(2,3)*B(2,1) *V1/(S*B(2,2)))+(X(4,3)*B(4,1) *V1/(S*B(4,2))) +((X(5,3)*B(5,1) *V2/(S*B(5,2)))^-1 +(X(6,3)*B(6,1) *V2/(S*B(6,2)))^-1)^-1))
Ia0g2=round(Ia0g2 *1e2)/1e2

mIa1g2=[ Ia0g2 Ia1g2 Ia2g2]'
mIag2=A*mIa1g2
Ibc=round(S*1e7/(sqrt(3) * V1 * 1e3))/10
Iag2=abs(mIag2) * Ibc


mprintf("\n\n(c)line currents in A at generator 2 ")
mprintf("\nIa= %s", string(round(abs(Iag2(1))*10)
```

```
/10)  )
mprintf("\nIb= %s", string(round(abs(Iag2(2))*10)/10) )
mprintf("\nIc= %s", string(round(abs(Iag2(3))*10)/10) )


Va1g2=round((1- (X(2,1) *%i * Ia1g2))*100)/100
Va2g2=round((0- (X(2,2) *%i * Ia2g2))*100)/100
Va0g2=round((0- ((X(2,3) +Xe) *%i * Ia0g2))*10000)/10000
mVa1g2=[ Va0g2  Va1g2 Va2g2]'
mVag2=A*mVa1g2
vg2=mVag2 * V1 / sqrt(3)

mprintf("\n\nline to neutral voltages in KV at generator 2 ")
mprintf("\nVa= %s", string(fix(abs(vg2(1))*100)/100) )
mprintf("\nVb= %s", string(fix(abs(vg2(2))*100)/100) )
mprintf("\nVc= %s", string(fix(abs(vg2(3))*100)/100) )

//12(c) --- g1

Ia1g1= (Ia1a-Ia1g2) * exp(%i * 1*%pi /6) *-1
Ia1g1=round(Ia1g1 *1e2)/1e2

Ia2g1= (Ia1a-Ia2g2)* exp(%i *-1*%pi /6) *-1
Ia2g1=round(Ia2g1 *1e2)/1e2

Ia0g1= 0

mIa1g1=[ Ia0g1 Ia1g1 Ia2g1]'
mIag1=A*mIa1g1
mIag1=round(mIag1*1e1)/1e1
Ibc=round(S*1e7/(sqrt(3) * V1 * 1e3))/10
```

```
Iag1=abs(mIag1) * Ibc


mprintf("\n\nline currents in A at generator 1")
mprintf("\nIa= %s", string(round(abs(Iag1(1))*1)/1))
mprintf("\nIb= %s", string(round(abs(Iag1(2))*1)/1))
mprintf("\nIc= %s", string(round(abs(Iag1(3))*1)/1))


Va1g1=round((1- (X(1,1) *%i * (Ia1a-Ia1g2)))*10000)/10000
Va1g1=Va1g1 *  exp(%i * 1*%pi /6)
Va2g1=round((0- (X(1,2) *%i * (Ia1a-Ia2g2)))*10000)/10000
Va2g1=Va2g1 *  exp(%i *- 1*%pi /6)
Va0g1=0
mVa1g1=[ Va0g1  Va1g1 Va2g1]'
mVag1=A*mVa1g1
vg1=mVag1 * V1 / sqrt(3)

mprintf("\n\nline to neutral voltages in KV at generator 1 ")
mprintf("\nVa= %s", string(fix(abs(vg1(1))*100)/100))
mprintf("\nVb= %s", string(fix(abs(vg1(2))*100)/100))
mprintf("\nVc= %s", string(fix(abs(vg1(3))*100)/100))
```

Scilab code Exa 10.13 calculate fault current during LG fault

```
clear
```

```
clc

Sb=37.5
Vb=33
Zb=Vb*Vb/Sb
Ib= Sb *1e6 / (sqrt(3) *Vb *1e3)
x1=[.18 .12 .1]
x2=[6.3 6.3 12.6]
X2=x2/Zb

X=x1+X2

x=X(1)+X(2)+X(3)

If=3*1/x
IF=If*Ib
mprintf("Fault current is %.1f A", IF)
```

Scilab code Exa 10.14 find line currents and voltages under LL fault conditions

```
clear
clc

Z=[.2 .2 .05]' * %i
S=30
V=11
I=S*1e6/(sqrt(3)*V*1e3)
E=1

Ia1=E/(Z(1)+Z(2))
Ia2=-Ia1
Ia0=0
Ibase=S*1e6/(sqrt(3)* V*1e3)

```

```
Ia=0
Ib=sqrt(3)*Ia1*Ibase
Ic=-Ib

mprintf("\nLine currents: (Ia Ib Ic) in Ampere")
mprintf("\nIa= %d", Ia)
mprintf("\nIb= -%d",abs(Ib))
mprintf("\nIc= %d", abs(Ic))

a=exp(%i * 2 * %pi/3)
A=[1 1 1
1 a^2 a
1 a a^2
]

Va1=E-Ia1*Z(1)
Va2=0-Ia2*Z(2)
Va0=0-Ia0*Z(3)

Vbll=V/sqrt(3)
Vp=[ Va0 Va1 Va2]'
v=A*Vp*Vbll

mprintf("\nline to neutral voltages in KV  ")
mprintf("\nVa= %s", string(round(abs(v(1))*1000)/1000) +'/_'+ string(round(atand(imag(v(1))/real(v(1)))*10)/10) )
mprintf("\nVb= %s", string(round(abs(v(2))*1000)/1000) +'/_'+ string(round(atand(imag(v(2))/real(v(2)))*10)/10 +180) )
mprintf("\nVc= %s", string(round(abs(v(3))*1000)/1000) +'/_'+ string(round(atand(imag(v(3))/real(v(3)))*10)/10 +180))
```

Scilab code Exa 10.15 find line currents and voltages under LLG fault conditions

```
clear
clc

clear
clc

Z=[.2 .2 .05]' * %i
S=30
V=11
I=round(S*1e8/(sqrt(3)*V*1e3) )/1e2
E=1

Ia1=E/(Z(1)+(Z(2)*Z(3)/(Z(2)+Z(3))))
Ia1=round(Ia1*1000)/1000
Ia2=-Ia1 * Z(3)/(Z(2)+Z(3))
Ia0=-Ia1 * Z(2)/(Z(2)+Z(3))

a=exp(%i * 2 * %pi/3)
A=[1 1 1
1 a^2 a
1 a a^2
]

Ia=A*[ Ia0 Ia1 Ia2]' * I

mprintf("Line currents: ")
mprintf("\nIa= %.2f ang(0) A",abs(Ia(1)))
mprintf("\nIb= %.2f ang(%.2f) A",abs(Ia(2)),atand(
    imag(Ia(2))/real(Ia(2)))+180)
mprintf("\nIc= %.2f ang(%.2f) A",abs(Ia(3)),atand(
    imag(Ia(3))/real(Ia(3))))


If=Ia(2)+Ia(3)
mprintf("\nFault current= %.0f ang(%.2f) A",abs(If),
```

```
    atand(imag(If)/real(If))

Va1=1-(Ia1*Z(1))
Va2=Va1
Va0=Va1

Va=A*[Va0 Va1 Va2]' * V/sqrt(3)
mprintf("\nLine to neutral voltages: ")
mprintf("\nVa= %.3f ang(%.2f) kV",abs(Va(1)),atand(
    imag(Va(1))/real(Va(1))))
mprintf("\nVb= %.3f KV",abs(Va(2)))
mprintf("\nVc= %.3f KV",abs(Va(3)))
```

Scilab code Exa 10.16 find line currents under LG fault conditions

```
clear
clc

X=[
 .25 .25 .05
 .2 .2 .05
 .06 .06 .06
 .07 .07 .07
 .1 .1 .3
 .1 .1 .3
]

B=[
    100 11
    100 11
    100 11
    100 11
    100 220
    100 220
    ]
```

```
V1=11
V2=220
S=100
Xe=3*.03
//end 9
X1=(((X(1,1)*B(1,1) *V1/(S*B(1,2))) +(X(3,1)*B(3,1)
   *V1/(S*B(3,2))) )^-1+((X(2,1)*B(2,1) *V1/(S*B
   (2,2)))+(X(4,1)*B(4,1) *V1/(S*B(4,2))) +((X(5,1)*
   B(5,1) *V2/(S*B(5,2)))^-1 +(X(6,1)*B(6,1) *V2/(S*
   B(6,2)))^-1)^-1)^-1)^-1

X2=(((X(1,2)*B(1,1) *V1/(S*B(1,2))) +(X(3,2)*B(3,1)
   *V1/(S*B(3,2))) )^-1+((X(2,2)*B(2,1) *V1/(S*B
   (2,2)))+(X(4,2)*B(4,1) *V1/(S*B(4,2))) +((X(5,2)*
   B(5,1) *V2/(S*B(5,2)))^-1 +(X(6,2)*B(6,1) *V2/(S*
   B(6,2)))^-1)^-1)^-1)^-1

X0=((X(3,3)*B(3,1) *V1/(S*B(3,2)))^-1 + ((Xe    *B
   (4,1) *V1/(S*B(4,2))) + (X(2,3)*B(2,1) *V1/(S*B
   (2,2))) + (X(4,3)*B(4,1) *V1/(S*B(4,2))) +(((X
   (5,3)*B(5,1) *V2/(S*B(5,2))))^-1 +(X(6,3)*B(6,1)
   *V2/(S*B(6,2)))^-1)^-1 )^-1 )^-1

Z1=%i * round(X1*1e3)/1e3
Z2=%i * round(X2*1e3)/1e3
Z0=%i * round(X0*1e3)/1e3

//end q11

a=exp(%i * 2 * %pi /3)
A= [1 1 1 ; 1 a^2 a; 1 a a^2]


Ia1=1/(Z1 + (Z2*Z0/(Z2+Z0)))
Ia1=round(Ia1 *1e3)/1e3
Ia2=(Z0/(Z2+Z0)) * Ia1 *-1
Ia2=round(Ia2 *1e3)/1e3
Ia0=(Z2/(Z2+Z0)) * Ia1 *-1
```

```
Ia0=round(Ia0 *1e3)/1e3

mIa1=[ Ia0 Ia1 Ia2]'
mIa=A*mIa1
Ib=round(S*1e7/(sqrt(3) * V2 * 1e3))/10
Ia=round(abs(mIa)) * Ib


mprintf("\nline currents are: in Amperes ")
mprintf("\nIa= %s", string(round(abs(Ia(1))*10)/10)
    )
mprintf("\nIb= %s", string(round(abs(Ia(2))*10)/10)
    +'/_'+ string(round(atand(imag(mIa(2)*Ib)/real(
    mIa(2)*Ib))*100)/100 +180 ))
mprintf("\nIc= %s", string(round(abs(Ia(3))*10)/10)
    +'/_'+ string(round(atand(imag(mIa(3)*Ib)/real(
    mIa(3)*Ib))*100)/100  ))
```

Scilab code Exa 10.17 find pu values of sequence networks

```
clear
clc

S=50
V=11
Z=V*V/S

If1=1870
If2=2590
If3=4130

X1=V*1e3 /(sqrt(3)*If1)
x1=X1/Z
X2=((V*1e3*sqrt(3)/(sqrt(3)) )/(If2))- X1
x2=X2/Z
```

```
X0=((sqrt(3)* V *1e3)/If3) - X1-X2
x0=X0/Z

mprintf("x1= %.2f pu \n x2= %.2f pu \n x0= %.2f pu", x1, x2,x0)
```

Scilab code Exa 10.18 calculate current in generator and motor during fault

```
clear
clc

Vt=10.95
V=11
S=100
I=round(S*1e7/(sqrt(3)*V*1e3))/10    //Error in evaluation of base current in textbook

vt=Vt/V
pf=.8
P=40

Il=round(P*1e7/(sqrt(3)*Vt*pf*1e3))/10
Il=Il * exp (%i * acos(pf))/I

x1g=.2
x2g=.2
x0g=.05

Sm=50
x1m=.2 * S/Sm
x2m=.2 * S/Sm
x0m=.05

xt1=.05
```

```
xt2=.05
xt0=.15


V=vt+ (Il * %i * x0m)
x0m=.05* S/Sm
Ia1=V/(%i*((x1g*(x1m+xt1)/(x1g+x1m+xt1))+(x2g*(x2m+
   xt2)/(x2g+x2m+xt2))+(x0g*(x0m+xt0)/(x0g+x0m+xt0))
   ))
Ia1=round(Ia1 * 1000)/1000
Ig1=round((Ia1 * (x1m + xt1)/(x1m+x1g+xt1) + Il)
   *1000)/1000
Im1=round((Ia1 * (x1g)/(x1m+x1g+xt1) - Il)*1000)
   /1000
Ig2=round((Ia1 * (x2m + xt2)/(x2m+x2g+xt2))*1000)
   /1000
Im2=round((Ia1 * (x2g)/(x2m+x2g+xt2))*1000)/1000
Ig0=round((Ia1 * (x0m + xt0)/(x0m+x0g+xt0))*100)/100
Im0=round((Ia1 * (x0g)/(x0m+x0g+xt0))*100)/100

Im=round((Im1+Im2+Im0)*1000)  *1e-3
Ig=round((Ig1+Ig2+Ig0)*1000)  *1e-3

mprintf("\nCurrent Through motor = %.2f, ang (%.1f)
   deg", abs(Im)*I, atand(imag(Im)/real(Im))+180)
mprintf("\nCurrent Through generator = %.2f, ang (%
   .1f) deg", abs(Ig)*I, atand(imag(Ig)/real(Ig)))
disp("Error in evaluation of base current in
   textbook")
```

Scilab code Exa 10.19 find short circuit MVA of parallel connection of 2 stations

```
clear
clc
```

```
S=100
V=11
Z=V*V/S

SCA=1000
SCB=650

xa=S/SCA
xb=S/SCB
Xc=.5
xc=Xc/Z

X=round(((xa *(xb+xc))/(xa+xb+xc))*1000)/1000
FS=S/X
mprintf("Fault MVA= %.2f MVA",FS)
```

Scilab code Exa 10.20 find X to prevent overloading of circuit breakers

```
clear
clc

V=33
S=75
Z=V*V/S

sg=15
xg=.15
Xg=xg *S/sg
xt=.08
x=Xg/3

CB=750
XF=S/CB
xi=((x*xt)-(XF*(x+xt)))/(XF-x)
```

```
xi=round(xi*10000)/10000
X=xi*Z
mprintf("reactance of reactor X= %.3f ohm",X)
```

Scilab code Exa 10.21 determine fault current and voltages during LG fault when different alternator neutrals are grounded or isolated

```
clear
clc

V=6.6
S=10
Z=V*V/S
I=S*1e6/(sqrt(3)*V*1e3)

X1=.15
X2=.75*X1
X0=.3*X1

x1=X1/3
x2=X2/3
x0=X0/3

E=1

x1=X1/3
x2=X2/3
x0=X0/3
IFa=abs(3*E/(%i * (x1+x2+x0)))
IFa=round(IFa*1000)/1000
ifa=IFa * I
mprintf("\n(a)fault current when all gen neutrals
    gounded= %.2f A",ifa)

x1=X1/3
```

```
x2=X2/3
x0=X0
IFb=abs(3*E/(%i * (x1+x2+x0)))
IFb=round(IFb*1000)/1000
ifb=IFb * I
mprintf("\n(b)fault current when one gen neutral
   gounded= %.2f A",ifb)

x1=X1/3
x2=X2/3
x0=X0
R0=.3
r0=round(3*R0/Z*1000)/1000
IFc=abs(3*E/(complex(r0,(x1+x2+x0))))
IFc=round(IFc*1000)/1000
ifc=IFc * I
mprintf("\n(c)fault current when one neutral gounded
    thru resistance= %.2f A",ifc)
```

Scilab code Exa 10.22 determine fault current and voltages during LG fault when alternator neutral is grounded and isolated

```
clear
clc

x1g=.1
x2g=.1
x0g=.05

x1t=.05
x2t=.05
x0t=.05

x1l=.4
x2l=.4
```

```
x0l=.8

x1lm=x1l/2
x2lm=x2l/2
x0lm=x0l/2

X1=x1g+x1lm
X2=x2g+x2lm
X0=(x0g+x0lm)*(x0lm+x0t)/(x0g+x0lm+x0lm+x0t)

X=X1+X2+X0
Ia1=round((1/(%i*X))*1000)/1000
Ia2=round((Ia1)*1000)/1000
Ia0=round((Ia1)*1000)/1000
Ia=Ia1+Ia2+Ia0
IFa=abs(Ia)

Va1=1-(Ia1 * X1 *%i)
Va2=0-(Ia2 * X2 *%i)
Va0=0-(Ia0 * X0 *%i)

a=exp(%i * 2 * %pi/3)
A=[1 1 1
1 a^2 a
1 a a^2
]

va1=[Va0 Va1 Va2]'
Va=A*va1
mprintf("\n(a)")
mprintf("\If= %.3f  ang(%.2f)",abs(Ia),270)
mprintf("\nVa= %.3f  ang(%.2f)",abs(Va(1)),atand(imag(Va(1))/real(Va(1))))
mprintf("\nVb= %.3f  ang(%.2f)",abs(Va(2)),atand(imag(Va(2))/real(Va(2)))+180)
mprintf("\nVc= %.3f  ang(%.2f)",abs(Va(3)),atand(imag(Va(3))/real(Va(3)))+180)

```

```
X1=x1g+x1lm
X2=x2g+x2lm
X0=(x0g+x0lm)

X=X1+X2+X0
Ia1=round((1/(%i*X))*1000)/1000
Ia2=round((Ia1)*1000)/1000
Ia0=round((Ia1)*1000)/1000
Ia=Ia1+Ia2+Ia0
IFa=abs(Ia)

Va1=1-(Ia1 * X1 *%i)
Va2=0-(Ia2 * X2 *%i)
Va0=0-(Ia0 * X0 *%i)

a=exp(%i * 2 * %pi/3)
A=[1 1 1
1 a^2 a
1 a a^2
]

va1=[Va0 Va1 Va2]'
Va=A*va1
mprintf("\n(b)")
mprintf("\If= %.3f ang(%.2f)",abs(Ia),270)
mprintf("\nVa= %.3f ang(%.2f)",abs(Va(1)),atand(imag(Va(1))/real(Va(1))))
mprintf("\nVb= %.3f ang(%.2f)",abs(Va(2)),atand(imag(Va(2))/real(Va(2)))+180)
mprintf("\nVc= %.3f ang(%.2f)",abs(Va(3)),atand(imag(Va(3))/real(Va(3)))+180)
```

Scilab code Exa 10.24 find reactance added to limit fault current in LG fault

```
clear
clc

X1=.4
X2=.3
X0=.05

S=15
V=13.2
Z=V*V/S
E=1
If=1

Xn=((3*E/(If))-(X1+X2+X0))/3
xn=Xn*Z
mprintf("\n(a)Xn= %.3f ohm ",xn)

Rn=((3*E/(If))-((X1+X2+X0)*%i))/3
rn=Rn*Z
mprintf("\n(b)Rn= %.2f ohm ",rn)
//the differnece in result is due to error in
    calculation in textbook
disp("the differnece in result is due to error in
    calculation in textbook")
```

Scilab code Exa 10.27 find SC MVA for 3 phase fault

```
clear
clc

S=50
data=[.05 20
.08 50
.06 30
.08 50
```

```
.04 30
.05 40
.05 50
.05 40
]
for(i=1:8)
    X(i)=round(data(i,1) * S/data(i,2)*10000)/10000
end

X1=round((((X(2)*X(8))+(X(2)*X(7))+(X(7)*X(8)))/X(2))*1000)/1000
X2=round((((X(2)*X(8))+(X(2)*X(7))+(X(7)*X(8)))/X(7))*1000)/1000
X3=round((((X(2)*X(8))+(X(2)*X(7))+(X(7)*X(8)))/X(8))*1000)/1000
X4=round((((1/X(1)) + (1/X2))^-1)*1000)/1000
X5=round((((1/X(4)) + (1/X3))^-1)*1000)/1000
X6=round((X4*X5/(X4+X5+X1))*1000)/1000
X7=round((X4*X1/(X4+X5+X1))*1000)/1000
X8=round((X1*X5/(X4+X5+X1))*1000)/1000
X9=round((X7+X(5))*1000)/1000
X10=round((X8+X(6))*1000)/1000
X11=round((((1/X10) + (1/X9))^-1)*1000)/1000
X12=round((X11+X6)*1000)/1000
X13=((1/X12) + (1/X(3)))^-1
MVA=S/X13
//Mismatch is due to error in calculation in the textbook
mprintf("Fault MVA=%.2f MVA",MVA)
disp("Mismatch is due to error in calculation in the textbook")
```

Scilab code Exa 10.28 find fault current and fault level LG 3 phase LL and LLG faults

```
clear
clc

S=30
SF=1000
V1=33
V2=132
Z1=V1*V1/S
I2=S*1e6/(sqrt(3)*V2*1e3)


x1g=%i*S/SF
x2g=x1g*2/3
x0g=x1g/3
r0g=60/Z1
z0g=r0g+x0g
xt=%i *.1

X1=xt+x1g
X2=xt+x2g
X0=xt

FMVAa=round(abs(S/X1)*100)/100
IFa=abs(1/X1)* I2
mprintf("\n(a)3 phase fault Fault MVA=%.2f MVA,
    Fault Current=%.2f A",FMVAa,IFa)

IFb=abs(3/(X1+X2+X0))* I2
FMVAb=IFb * sqrt(3) * V2 * 1e-3
mprintf("\n(b)single line to ground fault Fault MVA=
    %.2f MVA, Fault Current=%.2f A",FMVAb,IFb)

IFc=abs(sqrt(3)/(X1+X2))* I2
FMVAc=IFc * sqrt(3) * V2 * 1e-3
mprintf("\n(c)L-L Fault MVA=%.2f MVA, Fault Current
    =%.1f A",FMVAc,IFc)

IFd=abs(1/(X1+(X2*X0/(X2+X0))))* I2
```

```
FMVAd=IFd * sqrt(3) * V2 * 1e-3
mprintf("\n(d)L-L-G fault Fault MVA=%.2f MVA, Fault
    Current=%.2f A",FMVAd,IFd)
```

Scilab code Exa 10.29 find fault current and fault level LG fault in middle of line

```
clear
clc

a=exp(%i * 2*%pi/3)

V=33
S=45
SF=2000
V2=132
Z=V2*V2/S
I=S*1e6 /(sqrt(3) * V2 * 1e3)

X1=.4* 60
X2=.4* 60
X0=1 * 60

x1=X1/Z
x2=X2/Z
x0=X0/Z

xg=.0225
xt=.1

xf1=x1/2 + xg + xt
xf2=x2/2 + xg + xt
xf0=(x0/2 + xt)/2
xf=xf1+xf2+xf0
ia1=round(1000/xf)/1000
```

```
iF1=3*ia1
IF1=iF1*I
mprintf("\n(a)Fault Current = %.2fA",IF1)

IA1=ia1*-1*%i
IA2=ia1*-1*%i
IA0= ia1 * .5*-1*%i

IA=round((IA1 +IA2 +IA0)*10000)/10000
IB=round(((IA1*a*a) + (a*IA2) +IA0)*10000)/10000
IC=round(((IA1*a) + (a*a*IA2) +IA0)*10000)/10000
mprintf("\n(b)From T1 to P: (in order - IA,IB, IC in
    pu)")
disp(IC, IB, IA)

IA1b=0
IA2b=0
IA0b=ia1 * .5*-1*%i
IAb=round((IA1b +IA2b +IA0b)*10000)/10000
IBb=round(((IA1b*a*a) + (a*IA2b) +IA0b)*10000)/10000
ICb=round(((IA1b*a) + (a*a*IA2b) +IA0b)*10000)/10000
mprintf("\nFrom P to T2: (in order - IA,IB, IC in pu
   )")
disp(ICb, IBb, IAb)



//(c)


Ia0=0
Ia1=IA1 * exp(%i * -1*%pi/6)
Ia2=IA2 * exp(%i * 1*%pi/6)

Ia=round((Ia1 +Ia2 +Ia0)*1000)/1000
Ib=round(((Ia1*a*a) + (a*Ia2) +Ia0)*1000)/1000
Ic=round(((Ia1*a) + (a*a*Ia2) +Ia0)*1000)/1000
mprintf("\n(c) Currents in lines connecting source
```

```
to T1 (in order - Ia,Ib, Ic in pu)")
disp(Ic, Ib, Ia)


mprintf("\n(d) Currents in star wdg of T1")
mprintf("\nIa= %.4fj, Ib= %.4fj, Ic= %.4fj", imag(IA), imag(IB), imag(IC))
mprintf("\nIa1= %.4fj, Ia2= %.4fj, Ia0= %.4fj", imag(IA1), imag(IA2), imag(IA0))
mprintf("\nIb1= %.4f /_%.3f, Ib2=%.4f/_%.3f, Ib0= %.4fj", abs(IA1 *a*a),atand(imag(IA1*a*a)/real(IA1*a*a))+180,abs(IA1 *a),atand(imag(IA2*a)/real(IA2*a)),imag(IA0))
mprintf("\nIc1= %.4f /_%.3f, Ic2=%.4f/_%.3f, Ic0= %.4fj", abs(IA1 *a),atand(imag(IA1*a)/real(IA1*a)),abs(IA1 *a*a),atand(imag(IA2*a*a)/real(IA2*a*a))+180,imag(IA0))


Iab1 =  round(IA1*1e3 /sqrt(3))/1e3
Iab2 =  round(IA2*1e3 /sqrt(3))/1e3
Iab0 =  round(IA0*1e3 /sqrt(3))/1e3

Ibc1=Iab1 * a*a
Ibc2=Iab2 * a
Ibc0=Iab0

Ica1=Iab1 * a
Ica2=Iab2 * a *a
Ica0=Iab0

Iab= Iab1 +Iab2 +Iab0
Ibc= Ibc1 +Ibc2 +Ibc0
Ica= Ica1 +Ica2 +Ica0

mprintf("\n\nCurrents in delta wdg of T1")
mprintf("\nIab= %.4fj, Ibc= %.4fj, Ica= %.4fj", imag(Iab), imag(Ibc), imag(Ica))
```

```
mprintf("\nIab1= %.4fj , Iab2= %.4fj , Iab0= %.4fj",
    imag(Iab1), imag(Iab2), imag(Iab0))
mprintf("\nIbc1= %.4f /_%.3f, Ibc2=%.4f/_%.3f, Ibc0=
     %.4fj", abs(Ibc1),atand(imag(Ibc1)/real(Ibc1))
    +180,abs(Ibc2),atand(imag(Ibc2)/real(Ibc2)),imag(
    Ibc0))
mprintf("\nIca1= %.4f /_%.3f, Ica2=%.4f/_%.3f, Ica0=
     %.4fj", abs(Ica1),atand(imag(Ica1)/real(Ica1)),
    abs(Ica2),atand(imag(Ica2)/real(Ica2))+180,imag(
    Ica0))

mprintf("\n\n Currents in star wdg of T2")
mprintf("\nIa= %.4fj , Ib= %.4fj , Ic= %.4fj", imag(
    IAb), imag(IBb), imag(ICb))
mprintf("\nIa1= %.4f, Ia2= %.4f, Ia0= %.4fj", imag(
    IA1b), imag(IA2b), imag(IA0b))
mprintf("\nIb1= %.4f, Ib2= %.4f, Ib0= %.4fj", imag(
    IA1b *a*a), imag(IA1b *a),imag(IA0b))
mprintf("\nIc1= %.4f, Ic2= %.4f, Ic0= %.4fj", imag(
    IA1b   *a), imag(IA1b*a *a),imag(IA0b))


Iab1b =  round(IA1b*1e3 /sqrt(3))/1e3
Iab2b =  round(IA2b*1e3 /sqrt(3))/1e3
Iab0b =  round(IA0b*1e3 /sqrt(3))/1e3

Ibc1b=Iab1b * a*a
Ibc2b=Iab2b * a
Ibc0b=Iab0b

Ica1b=Iab1b * a
Ica2b=Iab2b * a *a
Ica0b=Iab0b

Iabb= Iab1b +Iab2b +Iab0b
Ibcb= Ibc1b +Ibc2b +Ibc0b
Icab= Ica1b +Ica2b +Ica0b

```

```
mprintf("\n\nCurrents in delta wdg of T2")
mprintf("\nIab= %.3fj , Ibc= %.3fj , Ica= %.3fj", imag(Iabb), imag(Ibcb), imag(Icab))
mprintf("\nIab1= %.3f, Iab2=%.3f, Iab0= %.3fj", imag(Iab1b), imag(Iab2b), imag(Iab0b))
mprintf("\nIbc1= %.3f, Ibc2=%.3f, Ibc0= %.3fj", imag(Ibc1b),imag(Ibc2b),imag(Ibc0b))
mprintf("\nIca1= %.3f, Ica2=%.3f, Ica0= %.3fj", imag(Ica1b),imag(Ica2b),imag(Ica0b))
```

Scilab code Exa 10.30 find fault current and fault level LG fault

```
clear
clc

R1=4
R2=2

S=50
V1=11
V2=132
V3=33
Z1=V1*V1/S
Z2=V2*V2/S
Z3=V3*V3/S

r1=3*R1/Z1
r2=3*R2/Z3


x1g=.4*%i
x2g=.3*%i
x0g=.1*%i

x1t1=.08*%i
```

```
x2t1=.08*%i
x0t1=.08*%i

x1t2=.05*%i
x2t2=.05*%i
x0t2=.05*%i

x1t3=.04*%i
x2t3=.04*%i
x0t3=.04*%i

x1t4=.06*%i
x2t4=.06*%i
x0t4=.06*%i


X1l=20*%i
X2l=20*%i
X0l=50*%i

x1l=X1l/Z2
x2l=X2l/Z2
x0l=X0l/Z2

X1=x1g+x1t2+x1l+x1t1+x1t3
X2=x2g+x2t2+x2l+x2t1+x2t3
X0=r2+(((x0t2+x0l+x0t1)*x0t4/(x0t2+x0l+x0t1+x0t4))+
    x0t3)

IF=abs(3*1/(X1+X2+X0))
IB=S*1e6/(sqrt(3)*V3*1e3)
If=IF*IB
SF=IF*S
mprintf("fault current= %.0fA, fault level=%.2f MVA"
    ,If,SF)
```

Scilab code Exa 10.31 find line voltages and currents for OC fault

```
clear
clc

a=exp(%i *2*%pi/3)

Z1=complex(2.8,1)
Z2=complex(.1,.6)

V=400
E=V/sqrt(3)

Ia1=E/(Z1+Z2)
Ia2=-Ia1

Ia=Ia1+Ia2
Ib= (a^2-a)*Ia1
Ic=-Ib

disp( "Line Currents Ia, Ib, Ic, in amperes")
mprintf("\nIa= %s", string(round(abs(Ia)*10)/10) +'/_'+ string(0) )
mprintf("\nIb= %s", string(round(abs(Ib)*10)/10) +'/_'+ string(round(atand(imag(Ib)/real(Ib))*100)/100 -180) )
mprintf("\nIc= %s", string(round(abs(Ic)*10)/10) +'/_'+ string(round(atand(imag(Ic)/real(Ic))*100)/100) )

Va2=-Z2 * Ia2
Vaa=3*Va2
Van=(Z1*Ia1)+(Z2*Ia2)
Vcn=(a*Z1*Ia1)+(a*a*Z2*Ia2)
```

```
Vbn=(a*a*Z1*Ia1)+(a*Z2*Ia2)
VNn=Va2


mprintf("\n\n\nVaa= %s", string(round(abs(Vaa)*100)/100) +'/_'+ string(round(atand(imag(Vaa)/real(Vaa))*10)/10))
mprintf("\nVan= %s", string(round(abs(Van)*100)/100) +'/_'+ string(round(atand(imag(Van)/real(Van))*10)/10))//error in value substitution in textbook
mprintf("\nVbn= %s", string(round(abs(Vbn)*10)/10) +'/_'+ string(round(atand(imag(Vbn)/real(Vbn))*10)/10 -180))
mprintf("\nVcn= %s", string(round(abs(Vcn)*10)/10) +'/_'+ string(round(atand(imag(Vcn)/real(Vcn))*10)/10 +180)) //error in value substitution in textbook
mprintf("\nVNn= %s", string(round(abs(VNn)*100)/100) +'/_'+ string(round(atand(imag(VNn)/real(VNn))*10)/10))
disp("error is due to mistake in value substitution in textbook")
```

Scilab code Exa 10.32 fault MVA with and without reactors

```
clear
clc

S=10
xg=.1
xe=.08

X1= 1/((1/.1) + 1/(xe + ((xg+xe)/2)))
FMVA1=S* (1/X1)
```

```
mprintf("When reactors are used, fault level=%.2f MVA", FMVA1)

X2= xg/3
FMVA2=S* (1/X2)
mprintf("\nWhen reactors are not used, fault level=%.0f MVA",FMVA2)
```

Scilab code Exa 10.33 find subtransient current in system

```
clear
clc

S=25
pf=.8
P=15
Vt=10.6
V1=11
V2=11 * 66/11
I1=S*1e6/(sqrt(3)*V1*1e3)
I2=S*1e6/(sqrt(3)*V2*1e3)
Il=(P*1e6/(sqrt(3)*Vt*1e3*pf)) * exp (%i * acos(pf))
vt=Vt/V1
Z=V2*V2/S
XL=10
xl=XL/Z
xt=.1
xg=.15
xm=.15

Xth=xm * (xg+xt+xt+xl)/(xm + xg+xt+xt+xl)
IF=vt/Xth
If=IF*I2

Ifg=IF*I1 *%i*-1* xm /(xm + xg+xt+xt+xl)
```

```
ifg=abs(Ifg + Il)
Ifm=IF*I1 *%i *-1* (xg+xt+xt+xl)/(xg+xt+xt+xl+xm)
ifm=abs(Ifm - Il)

mprintf("total fault current = %.0f A, current through generator=%.0f A, current through motor=%.0f A\n",If, ifg, ifm)
//error in calculation of Ifm-I =-.623 - 6.891j instead of -.623-j5.96
disp("error in calculation of Ifm-I =-.623 - 6.891j ->(correct) instead of -.623-j5.96 -> incorrect")
```

Scilab code Exa 10.34 reactance needed to restrict 6 times fault current

```
clear
clc

S=25
V=11
Z=V*V/S
I=S*1e6/(sqrt(3)*V*1e3)
Isc=6*I
Xt=V*1e3/(sqrt(3)*Isc)
Xi=.15*Z
Xo=Xt-Xi
x=Xo*100/Z
mprintf("External reactance required is %.3f pu",x)
```

Scilab code Exa 10.35 symmetrical components of line and delta currents

```
clear
clc

```

```
Ia=10*exp(%i *30 *%pi/180)
Ib=15*exp(%i *-60*%pi/180)
Ic=0-(Ia+Ib)

Iac=(Ia +Ia +Ib)/3
Icb=Ic +Iac
Iba=Iac-Ia
Ia0=(Ia +Ib +Ic)/3

a=exp(%i * 2 * %pi/3)

Ia1=(Ia + a*Ib + a*a*Ic)/3
Ia2=(Ia + a*Ic + a*a*Ib)/3

disp(round(Ia1*100)/100, "Ia1","(a)Symmetrical Line Components")
disp(round(Ia2*100)/100, "Ia2")
disp(round(Ia0*100)/100, "Ia0")

Iac0=(Iac+Icb+Iba)/3
Iac1=(Iac+ a*Icb+ a*a*Iba)/3
Iac2=(Iac+a*a*Icb+a*Iba)/3

disp(round(Iac1*100)/100, "Iac1","(b)Symmetrical Delta Components")
disp(round(Iac2*100)/100, "Iac2")
disp(round(Iac0*100)/100, "Iac0")
```

Chapter 11

Digital Techniques in Fault Calculations

Scilab code Exa 11.1 z bus formulation

```
clear;
clc;

n=5;//no of elements
Z=0;
z=[ 4 1 1 1; 4 2 1 1; 4 3 1 1; 1 2 1 4; 1 3 1 4];

// z = [ from node | to node| z between nodes | type
    modification] type modification should be in
   ascending order

for(i=1:n)
    mcase=z(i,4)
    znew=z(i,3)
    n1=z(i,1)
    n2=z(i,2)
    dim=max(size(Z))
    select mcase
    case 1 then
```

```
            if Z(1,1)==0 then
                dim=dim-1
            end
           Z(dim+1, dim+1)=znew
        case 2 then
            Z(1:dim,dim+1)=Z(1:dim, n1)
            Z(dim+1,1:dim)=Z(n1,1:dim)
            Z(dim+1, dim+1)=znew+Z(n1,n1)
        case 3 then
            Z=Z-((Z(1:dim, n2)*Z(n2,1:dim))/(znew+Z(n2,n2)))
        case 4 then
            Z=Z-(((Z(1:dim, n1)-Z(1:dim, n2))*(Z(n1,1:dim)-Z(n2,1:dim)))/(znew+Z(n2,n2)+Z(n1,n1)-(2*+Z(n1,n2))))
        else
            break
    end
end
disp(Z)
```

Scilab code Exa 11.2 formulate positive and negative sequence impedance matrices for the network

```
clear;
clc;

n=5;
Z=0;
z=[ 0 1 .25*%i 1; 1 2 .06*%i 2; 2 3 .05*%i 2;3 4 .07*%i 2; 0 4 .2*%i 3];

for(i=1:n)
    mcase=z(i,4)
    znew=z(i,3)
```

```
    n1=real(z(i,1))
    n2=real(z(i,2))
    dim=max(size(Z))
    select mcase
        case 1 then
            if Z(1,1)==0 then
                dim=dim-1
            end
           Z(dim+1, dim+1)=znew
        case 2 then
            Z(dim+1,dim+1)=znew+Z(n1,n1)
            Z(1:dim,dim+1)=Z(1:dim, n1)
            Z(dim+1,1:dim)=Z(n1,1:dim)
        case 3 then

            Z=Z-((Z(1:dim, n2)*Z(n2,1:dim))/(znew+Z(n2,n2)))
        case 4 then
            Z=Z-(((Z(1:dim, n1)-Z(1:dim, n2))*(Z(n1,1:dim)-Z(n2,1:dim)))/(znew+Z(n2,n2)+Z(n1,n1)-(2*+Z(n1,n2))))
        else
            break
    end
end
mprintf("Z1bus=Z2bus=");
disp(Z)
```

Scilab code Exa 11.3 formulate zero sequence impedance matrices for the network

```
clear;
clc;

n=5;
```

```
Z=0;
z=[ 0 1 .05*%i 1; 0 2 .06*%i 1; 2 3 .15*%i 2;3 4
    .07*%i 2; 0 4 .14*%i 3];

for(i=1:n)
    mcase=z(i,4)
    znew=z(i,3)
    n1=real(z(i,1))
    n2=real(z(i,2))
    dim=max(size(Z))
    select mcase
        case 1 then
            if Z(1,1)==0 then
                dim=dim-1
            end
           Z(dim+1, dim+1)=znew
        case 2 then
            Z(dim+1,dim+1)=znew+Z(n1,n1)
            Z(1:dim,dim+1)=Z(1:dim, n1)
            Z(dim+1,1:dim)=Z(n1,1:dim)
        case 3 then

            Z=Z-((Z(1:dim, n2)*Z(n2,1:dim))/(znew+Z(
               n2,n2)))
        case 4 then
            Z=Z-(((Z(1:dim, n1)-Z(1:dim, n2))*(Z(n1
               ,1:dim)-Z(n2,1:dim)))/(znew+Z(n2,n2)+
               Z(n1,n1)-(2*+Z(n1,n2))))
        else
            break
    end
end
mprintf("Z0bus=");
disp(Z)
```

Scilab code Exa 11.4 finding fault current and fault voltage at bus

```
clear;
clc;

n=5;
Z=0;
z=[ 0 1 .25*%i 1; 1 2 .06*%i 2; 2 3 .05*%i 2;3 4 .07*%i 2; 0 4 .2*%i 3];

for(i=1:n)
    mcase=z(i,4)
    znew=z(i,3)
    n1=real(z(i,1))
    n2=real(z(i,2))
    dim=max(size(Z))
    select mcase
        case 1 then
            if Z(1,1)==0 then
                dim=dim-1
            end
           Z(dim+1, dim+1)=znew
        case 2 then
            Z(dim+1,dim+1)=znew+Z(n1,n1)
            Z(1:dim,dim+1)=Z(1:dim, n1)
            Z(dim+1,1:dim)=Z(n1,1:dim)
        case 3 then

            Z=Z-((Z(1:dim, n2)*Z(n2,1:dim))/(znew+Z(n2,n2)))
        case 4 then
            Z=Z-(((Z(1:dim, n1)-Z(1:dim, n2))*(Z(n1,1:dim)-Z(n2,1:dim)))/(znew+Z(n2,n2)+Z(n1,n1)-(2*+Z(n1,n2))))
        else
            break
    end
end
```

```
E=1
V=ones(1,4);
Ib=262.4;
i2=V(1,2) / Z(2,2);
I2=Ib*i2;

Ia=I2 * exp(%i * 0);
Ib=I2 * exp(%i *-2* %pi /3);
Ic=I2 * exp(%i *2 * %pi /3);
mprintf("(a)\nLine currents at bus 2\nIa = %.2f ang
    ( %.0f ) deg A,\nIb = %.2f ang ( %.0f ) deg A,\
    nIc = %.2f ang ( %.0f ) deg A",abs(Ia),acotd(real
    (Ia)/imag(Ia))-180,abs(Ib),acotd(real(Ib)/imag(Ib
    )),abs(Ic),acotd(real(Ic)/imag(Ic)))


Vb=220;
v3=E* (1-(Z(3,2)/Z(2,2)))
V3=v3*Vb/sqrt(3);
Va=V3 * exp(%i * 0);
Vb=V3 * exp(%i *-2* %pi /3);
Vc=V3 * exp(%i *2 * %pi /3);
mprintf("\n(b)\nLine voltages at bus 2\nVa = %.3f
    ang ( %.2f ) degKv,\nVb = %.3f ang ( %.2f ) degkV
    ,\nVc = %.3f ang ( %.2f ) degkV",abs(Va),atand(
    imag(Va)/real(Va)),abs(Vb),atand(imag(Vb)/real(Vb
    ))+180,abs(Vc),atand(imag(Vc)/real(Vc))+180)
```

Scilab code Exa 11.5 finding fault current and fault voltage at bus

```
clear;
clc;

n=5;
```

```
Z=0;
z=[ 0 1 .25*%i 1; 1 2 .06*%i 2; 2 3 .05*%i 2;3 4 .07*%i 2; 0 4 .2*%i 3];

for(i=1:n)
    mcase=z(i,4)
    znew=z(i,3)
    n1=real(z(i,1))
    n2=real(z(i,2))
    dim=max(size(Z))
    select mcase
        case 1 then
            if Z(1,1)==0 then
                dim=dim-1
            end
           Z(dim+1, dim+1)=znew
        case 2 then
            Z(dim+1,dim+1)=znew+Z(n1,n1)
            Z(1:dim,dim+1)=Z(1:dim, n1)
            Z(dim+1,1:dim)=Z(n1,1:dim)
        case 3 then

            Z=Z-((Z(1:dim, n2)*Z(n2,1:dim))/(znew+Z(n2,n2)))
        case 4 then
            Z=Z-(((Z(1:dim, n1)-Z(1:dim, n2))*(Z(n1,1:dim)-Z(n2,1:dim)))/(znew+Z(n2,n2)+Z(n1,n1)-(2*+Z(n1,n2))))
        else
            break
    end
end
Z1=Z;
Z2=Z;

n=5;
Z=0;
z=[ 0 1 .05*%i 1; 0 2 .06*%i 1; 2 3 .15*%i 2;3 4
```

```
    .07*%i 2; 0 4 .14*%i 3];

for(i=1:n)
    mcase=z(i,4)
    znew=z(i,3)
    n1=real(z(i,1))
    n2=real(z(i,2))
    dim=max(size(Z))
    select mcase
        case 1 then
            if Z(1,1)==0 then
                dim=dim-1
            end
           Z(dim+1, dim+1)=znew
        case 2 then
            Z(dim+1,dim+1)=znew+Z(n1,n1)
            Z(1:dim,dim+1)=Z(1:dim, n1)
            Z(dim+1,1:dim)=Z(n1,1:dim)
        case 3 then

            Z=Z-((Z(1:dim, n2)*Z(n2,1:dim))/(znew+Z(n2,n2)))
        case 4 then
            Z=Z-(((Z(1:dim, n1)-Z(1:dim, n2))*(Z(n1,1:dim)-Z(n2,1:dim)))/(znew+Z(n2,n2)+Z(n1,n1)-(2*+Z(n1,n2))))
        else
            break
    end
end
Z0=Z;

Ib=262.4;
Vb=220;
E=1
V=ones(1,4);

I3(3,1)=E/(Z1(3,3)+Z2(3,3)+Z0(3,3));
```

```
I3(1,1)=I3(3,1)
I3(2,1)=I3(3,1)
i3=I3*Ib
a=exp(%i * 2*%pi/3)
A=[1 1 1; 1 a^2 a; 1 a a^2]
I=A*i3;
//I=round(I*100)/100;
mprintf("(a)\nLine currents at bus 3\nIa = %.2f ang ( %.0f ) deg A,\nIb = %.2f ang ( %.0f ) deg A,\nIc = %.2f ang ( %.0f ) deg A",abs(I(1)),acotd(real(I(1))/imag(I(1)))-180,abs(I(2)),acotd(real(I(2))/imag(I(2)))-180,abs(I(3)),acotd(real(I(3))/imag(I(3)))-180)


V2(1,1)=-1* Z0(2,3)*I3(3,1);
V2(2,1)=E-( Z1(2,3)*I3(1,1));
V2(3,1)=-1* Z2(2,3)*I3(2,1);
v=A*V2;
V=Vb*v/sqrt(3);
mprintf("\n(b)\nLine voltages at bus 2\nVa = %.2f ang ( %.2f ) degKv,\nVb = %.2f ang ( %.2f ) degkV,\nVc = %.2f ang ( %.2f ) degkV",abs(V(1)),atand(imag(V(1))/real(V(1))),abs(V(2)),atand(imag(V(2))/real(V(2)))+180,abs(V(3)),atand(imag(V(3))/real(V(3)))+180)
```

Scilab code Exa 11.6 find z bus

```
clear;
clc;

n=4;
Z=0;
z=[ 4 1 .5*%i 1; 4 2 .4*%i 1; 1 3 .2*%i 2; 2 3 .1*%i
```

```
    4];

for(i=1:n)
    mcase=z(i,4)
    znew=z(i,3)
    n1=real(z(i,1))
    n2=real(z(i,2))
    dim=max(size(Z))
    select mcase
        case 1 then
            if Z(1,1)==0 then
                dim=dim-1
            end
           Z(dim+1, dim+1)=znew
        case 2 then
            Z(dim+1,dim+1)=znew+Z(n1,n1)
            Z(1:dim,dim+1)=Z(1:dim, n1)
            Z(dim+1,1:dim)=Z(n1,1:dim)
        case 3 then

            Z=Z-((Z(1:dim, n2)*Z(n2,1:dim))/(znew+Z(
               n2,n2)))
        case 4 then
            Z=Z-(((Z(1:dim, n1)-Z(1:dim, n2))*(Z(n1
               ,1:dim)-Z(n2,1:dim)))/(znew+Z(n2,n2)+
               Z(n1,n1)-(2*+Z(n1,n2))))
        else
            break
    end
end
mprintf("Zbus=");
Z=round(Z*1e5)/1e5
disp(Z)
```

Scilab code Exa 11.7 find z bus of an augmented network

```
clear;
clc;

no=4;
Z=0;
z=[ 4 1 .5*%i 1; 4 2 .4*%i 1; 1 3 .2*%i 2; 2 3 .1*%i
    4];

for(i=1:no)
    mcase=z(i,4)
    znew=z(i,3)
    n1=real(z(i,1))
    n2=real(z(i,2))
    dim=max(size(Z))
    select mcase
        case 1 then
            if Z(1,1)==0 then
                dim=dim-1
            end
           Z(dim+1, dim+1)=znew
        case 2 then
            Z(dim+1,dim+1)=znew+Z(n1,n1)
            Z(1:dim,dim+1)=Z(1:dim, n1)
            Z(dim+1,1:dim)=Z(n1,1:dim)
        case 3 then

            Z=Z-((Z(1:dim, n2)*Z(n2,1:dim))/(znew+Z(
               n2,n2)))
        case 4 then
            Z=Z-(((Z(1:dim, n1)-Z(1:dim, n2))*(Z(n1
               ,1:dim)-Z(n2,1:dim)))/(znew+Z(n2,n2)+
               Z(n1,n1)-(2*+Z(n1,n2))))
        else
            break
    end
end


```

```
m=1
n=3
p=1
q=4
no2=4
znew=.5*%i
zm=.1*%i
za=.2*%i

for j=1:no2
    if j==q then
        Z(q,q)=Z(p,q)-((zm/za)*(Z(m,q)-Z(n,q)))-((zm
           *zm/za)-znew);
    else
        Z(q,j)=Z(p,j)-((zm/za)*(Z(m,j)-Z(n,j)))
        Z(j,q)=Z(q,j)
    end
end
Z=round(Z*1e5)/1e5
disp(Z)
```

Chapter 12

Power System Transients

Scilab code Exa 12.1 find L C surge impedance and velocity of propagation

```
clear;
clc;

rc=.5e-2;
rs=1.5e-2;
u=4

L=2e-7 * log(rs/rc);
mprintf("\nL= %.1f e-7H/m", L*1e7)
C=u*1e-9/(18 * log(rs/rc))
mprintf("\nC= %.3f e-9F/m", C*1e9)
v=1/sqrt(L*C);
mprintf("\nv= %.1f e8m/s", v*1e-8)
Zc=sqrt(L/C)
mprintf("\nZc= %.0f ohm", Zc)
```

Scilab code Exa 12.2 find surge transmitted

```
clear;
clc;

ef=100;
Zc=400;
Z=50;
et=2*ef*Z/(Z+Zc)
mprintf("Surge transmitted= %.2f kV",et)
```

Scilab code Exa 12.3 find surge Vand I transmitted

```
clear;
clc;

ef=200;
Zc=400;
Z1=500;
Z2=300;
et=2*ef*(Z1*Z2/(Z1+Z2))/((Z1*Z2/(Z1+Z2))+Zc)
mprintf("\nSurge Voltage transmitted= %.2f kV",et)
it1=et/Z1;
mprintf("\nSurge Current transmitted= %.3f kA",it1)
it2=et/Z2;
mprintf("\nSurge Current transmitted= %.3f kA",fix(
    it2*100)/100)
er=et-ef;
mprintf("\nSurge Voltage Reflected= %.2f kV",er)
ir=it1+it2-(ef/Zc)
mprintf("\nSurge Current Reflected= %.2f kA",ir)
```

Scilab code Exa 12.4 find voltage across the inductance and the reflected voltage wave

```
clear
clc

E=100
Zc=400
L=4000

mprintf("et= %d exp( - %.1f t) KV\n", 2*E, Zc/L)
mprintf("er= %d (2*exp( - %.1f t) -1) KV\n", E, Zc/L
   )
```

Scilab code Exa 12.7 find surge arrester voltage and current

```
clear
clc

V=300e3
R=400
k=1.5e-27

E=10
x=1
e=1e-5
while (E>e)
    f=(k*R*x^6) +x -(2*V)
    df=(6* k*R*x^5) +1
    x1=x-(f/df)
    E=abs(x1-x)
    x=x1
end
eA=round(x)
IA=k*eA^6

mprintf("eA=%d, Ia=%d ",eA, IA)
```

Scilab code Exa 12.8 find surge arrester voltage and current

```
clear
clc

V=300e3
R1=400
R2=50
R=1+(400/50)
k=1.5e-27

E=10
x=1
e=1e-5
while (E>e)
    f=(k*R1*x^6) +(R*x) -(2*V)
    df=(6* k*R1*x^5) +R
    x1=x-(f/df)
    E=abs(x1-x)
    x=x1
end
eA=round(x)
IA=k*eA^6

mprintf("eA=%d,  Ia=%.1f  ",eA, IA)
```

Scilab code Exa 12.9 find reflection and refraction coefficients

```
clear;
clc;

ef=3000;
```

```
Zc=300;
ea=1700;
iF=ef/Zc
mprintf("\nCurrent in line= %d kA",iF)
Ia=((2*ef)-ea)/Zc
mprintf("\nCurrent through Arrester= %.3f kA",Ia)
Ia=round(Ia *1000)/1000
R=ea/Ia
mprintf("\nresistance of arrester= %.2f ohm",R)
er=ea-ef;
mprintf("\nSurge Voltage Reflected= %.0f kV",er)
Cr=er/ef;
CR=ea/ef;
mprintf("\nCoeff of Reflection = %.3f, Coeff of
    Refraction=%.3f",Cr,CR)
Cr=(R-Zc)/(R+Zc);
CR=(R*2)/(R+Zc);
mprintf("\nVerification: Coeff of Reflection = %.3f,
    Coeff of Refraction=%.3f",Cr,CR)
```

Scilab code Exa 12.10 reflection and transmission of voltage and current wave

```
clear;
clc;

ef=10000;
Zc=400;
iF=ef/Zc
mprintf("\n(a)\nIncident Wave magnitude= %d A",iF)

R=1000
et=ef*(R*2)/(R+Zc);
it=et/R;
er=et-ef;
```

```
mprintf("\n(b)\nSurge Voltage Reflected= %.3f KV",er/1000)
ir=-1*er/Zc
mprintf("\nSurge Current Reflected= %.3f A",ir)
edr=et*it;
mprintf("\nRate of dissipation of energy= %.2f KW",edr/1000)
err=er*-ir;
mprintf("\nRate of reflection of energy= %.3f KW",err/1000)

mprintf("\n(c)\nfor complete dissipation , R=Zc= %.0f ohm",Zc);

R=50
et=ef*(R*2)/(R+Zc);
mprintf("\n(d)\nSurge Voltage Transmitted= %.3f KV",et/1000)
it=et/R;
mprintf("\nSurge Current Transmitted= %.2f A",it)
er=et-ef;
mprintf("\nSurge Voltage Reflected= %.3f kV",er/1000)
ir=-1*er/Zc
mprintf("\nSurge Current Reflected= %.3f A",ir)
edr=et*it;
mprintf("\nRate of dissipation of energy= %.2f KW",edr/1000)
err=er*-ir;
mprintf("\nRate of reflection of energy= %.2f KW",err/1000)
```

Scilab code Exa 12.11 find V and I transmitted

```
clear;
```

```
clc;

Zc=400
ef=20
z1=150;
z2=200
z=round((z1*z2/(z1+z2))*100)/100

et=2*ef*z/(Zc+z)
mprintf("\nSurge Voltage Transmitted= %.4f kV",et)

it1=et*1000/z1;
mprintf("\nSurge Current Transmitted in line 1= %.3f
    A",it1)

it2=et*1000/z2;
mprintf("\nSurge Current Transmitted in line 2= %.2f
    A",it2)

er=et-ef
mprintf("\nSurge Voltage Reflected= %.4f kV",er)
ir=-1*er*1000/Zc
mprintf("\nSurge Current Reflected= %.2f A",ir)
```

Scilab code Exa 12.12 reflection transmission and absorption of wave

```
clear;
clc;

ef=100
Zc=400
z=50

R=z+Zc;
E=(2*ef/(Zc+z+R))^2 *R
```

```
E=round(E*100)/100
mprintf("\n(a)Energy transfer max when R= %.0f ohm,
   energy= %.2f KW", R,E);

etB=2*ef*z/(z+Zc+R);
etB=round(etB*100)/100
mprintf("\n(b)Surge Voltage Transmitted= %.3f kV",
   etB)

it=etB*1000/z;
it=round(it*100)/100
mprintf("\nSurge Current Transmitted = %.2f A",it)

etA=2*ef*(z+R)/(z+Zc+R);
etA=round(etA*100)/100

erA=etA-ef
mprintf("\n(c)Surge Voltage Reflected= %.2f kV",erA)
irA=-1*erA*1000/Zc
mprintf("\nSurge Current Reflected= %.3f A",irA)


iF=ef*1000/Zc
Pi=ef*iF
mprintf("\n(d)Power Incident= %.0f kW",Pi)
Pr=erA*-irA
mprintf("\nPower Reflected= %.2f kW",Pr)
Pt=erA*it
mprintf("\nPower Transmitted= %.0f kW",Pt)
```

Scilab code Exa 12.14 find voltage and current surges

```
clear;
clc

```

```
I=5;
z1=400
z2=50
V=I * z1* z2/(z2+z1)
mprintf("\nSurge Voltage Transmitted= %f kV",V)
ic=V/z2
mprintf("\nSurge Current Transmitted in cable= %f kA
  ",ic)

io=-V/z1;
mprintf("\nSurge Current Transmitted in OH line= %f
  kA",io)
```

Scilab code Exa 12.16 find restriking voltage due to current chopping

```
clear;
clc

i=100
L=4e-3
C=300e-12
E=i* sqrt(L/C)
T=1/ sqrt(L*C)
mprintf("e= %.0f *1e3 sin( %.3f *1e6 t) kV",E/1e3,T
  /1e6)
```

Chapter 13

Power System Stability

Scilab code Exa 13.1 find P Q E and load angle for changes to P and E

```
clear;
clc

Xd=.7
pf=.8
pfa=acos(pf)
V=1
I0=1* exp( %i * pfa *-1)
E0=V+ (%i * Xd * I0)
E=round(abs(E0)*100)/100
d0=atand(imag(E0)/real(E0))
E0=E * exp(%i * d0 * %pi/180)
Pe0=E*V*sind(d0)/Xd
Qe0=(E*V*cosd(d0)/Xd)-(V*V/Xd)

mprintf("\n(a)\nPe= %.1f Qe=%.1f E= %.2f load angle=
    %.1f",Pe0, Qe0, E, d0);

e1=E0
E1=abs(e1)
Pe1=1.2* Pe0;
```

```
d1=asind(Pe1* Xd/ (V*E1))
Qe1=(E1*V*cosd(d1)/Xd)-(V*V/Xd)

mprintf("\n(b)\nPe= %.2f Qe=%.2f E= %.2f load angle=
    %.1f",Pe1, Qe1, E1, d1);

e2=1.2 * E0
E2=abs(e2)
Pe2=Pe0;
d2=asind(Pe2* Xd/ (V*E2))
Qe2=(E2*V*cosd(d2)/Xd)-(V*V/Xd)

mprintf("\n(c)\nPe= %.1f Qe=%.2f E= %.2f load angle=
    %.1f",Pe2, Qe2, E2, d2);
```

Scilab code Exa 13.2 find inertia constants retardation

```
clear;
clc

P=4
f=50
G=200
H=6
J=G*H
mprintf("\n(a)stored energy = %.0f MJ",J)

Pl=120
Pl1=160
Pa=Pl-Pl1
M=J/(180*f)
a=Pa/M
mprintf("\n(b)acceleration = %.0f elec deg/ sec sq",
    a)

```

```
c=5
t=c/f
dd=.5*a*t*t
N=120*f/P
a=a*60/(180*P)
Nn=N+(a*t)
mprintf("\n(c)change in power angle = %.1f elec deg, change in speed = %.1f rpm",dd,Nn)

G2=150
H2=4
Gb=100
Heq=((G*H)+(H2*G2))/Gb
mprintf("\n(d)Equivalent inertia constant = %.0f MJ/MVA",Heq)
```

Scilab code Exa 13.3 find steady state stability parameters

```
clear;
clc

f=50;
H=9;
x=.6
P=.7
pf=.8
pfa=acos(pf)
V=1
D=.14
dd=10 * %pi/180

S=P/pf * exp(%i * pfa)
I=conj(S/V)
E=V+ (%i * x *I)
d0=atand(imag(E)/real(E))
```

```
Pr=abs(E) * V * cosd(d0)/x
mprintf("\nSynchronising Power Coefficient = %.3f pu ",Pr)

w1=sqrt(Pr *f *%pi/H)
mprintf("\nUndamped Speed Of oscillations = %.2f rad/sec",w1)
w1=round(w1*100)/100

z=(D/2) * sqrt(%pi * f/(H*Pr))
mprintf("\nDamping Ratio = %.4f ",z)

wd=w1 *sqrt(1-(z*z))
Wd=wd / (2*%pi)
mprintf("\ndamped angular frequency Of oscillations = %.3f rad/sec = %.3f Hz",wd, Wd)

z=round(z*1e4)/1e4
c_1=10/sqrt(1-z^2)
c_2=z*w1

mprintf("\n\nd = %.2f + %.3f exp(-%.3f t) sin(%.3f t + %.1f deg)", d0, c_1, c_2,wd, acosd(z))

c_3=w1 * 10 /(360 * sqrt(1-z^2))
mprintf("\n\nf = %.0f - %.4f exp(-%.3f t) sin(%.3f t)", f, c_3, c_2,wd)
```

Scilab code Exa 13.4 derive expressions for oscillations of delta and freq as functions of time

```
clear;
clc
dP=.1
f=50;
```

```
H=9;
x=.6
P=.7
pf=.8
pfa=acos(pf)
V=1
D=.14
dd=10 * %pi/180

S=P/pf * exp(%i * pfa)
I=conj(S/V)
E=V+ (%i * x *I)
d0=atand(imag(E)/real(E))
Pr=abs(E) * V * cosd(d0)/x

w1=sqrt(Pr *f *%pi/H)
w1=round(w1*100)/100

z=(D/2) * sqrt(%pi * f/(H*Pr))

wd=w1 *sqrt(1-(z*z))
Wd=wd / (2*%pi)

z=round(z*1e4)/1e4
c_1=1/sqrt(1-z^2)
c_2=z*w1
c_3=180 * f * dP/(9 * w1^2)

mprintf("\n\nd = %.2f + %.3f(1- (%.3f exp(-%.3f t) sin(%.3f t + %.1f deg)))", d0, c_3,c_1, c_2,wd, acosd(z))

c_4= dP * f /(w1* 9 * sqrt(1-z^2))
mprintf("\n\nf = %.0f + %.4f exp(-%.3f t) sin(%.3f t)", f, c_4, c_2,wd)
```

Scilab code Exa 13.5 finding steady state reactance and transfer limit for different shunt branches

```
clear;
clc

E=1.1
V=1
xg=.7
xt=.1
x1=.2
X1=x1+xg+xt
X2=.2
xL=.5
xC=-.5


X=X1+X2
P1=E*V/X
mprintf("\n(a) Steady State limit = %.4f", P1)

X3=xL
X=((X1*X2) + (X2*X3) + (X1*X3))/X3
P2=E*V/X
mprintf("\n(b) Steady State limit = %.4f", P2)

X3=xC
X=((X1*X2) + (X2*X3) + (X1*X3))/X3
P3=E*V/X
mprintf("\n(c) Steady State limit = %.3f", P3)

X=X1+xC+X2
P4=E*V/X
mprintf("\n(d) Steady State limit = %.2f", P4)
```

Scilab code Exa 13.6 frequency of oscillation of generator due to loading

```
clear;
clc

f=50
xg=1
xl=.5
E=1.1
V=1
H=5
p=.5

X=xl+xg
d0=asin(p)
Pr=E*V*cos(d0)/X
M=H/(%pi*f)
wn=sqrt(Pr/M)/(2*%pi)

mprintf("Freg of oscillation = %.2f Hz", wn)
```

Scilab code Exa 13.7 system stability and finding critical load angle

```
clear
clc

xd=.2
x1=.4
x2=.4
Pi=1.5
E=1.2
V=1
```

```
X=xd  +((x1*x2)/(x1+x2))
pe=E*V/X
d0 = asin(Pi/pe)

X2=xd+x1
pe2=E*V/X2
d1 = asin(Pi/pe2)
dm=%pi- d1

A1=((Pi * d1)+ ( pe2 * cos(d1)))- ((Pi * d0)+ ( pe2 * cos(d0)))
A2=((Pi * dm)+ ( pe2 * cos(dm)))- ((Pi * d1)+ ( pe2 * cos(d1)))

if abs(A1)<abs(A2) then
    mprintf("STABLE\n\n")
else
    mprintf("UNSTABLE\n\n")
end

E=10
x=2
e=1e-3
a=Pi
b=pe2
c=-A1 +(Pi * d1 ) + (cos (d1) * pe2)

while (E>e)
    f=(a*x) +  (b * cos(x)) - c
    df=1.5 - (2*sin(x))
    x1=x-(f/df)
    E=abs(x1-x)
    x=x1
end
d2=x1 * 180/%pi
mprintf("\ndelta 2 = %.2f deg",d2)
```

Scilab code Exa 13.8 system stability and finding critical load angle in 3 phase fault in line

```
clear
clc

xd=.2
x1=.4
x2=.4
Pi=1.5
E=1.2
V=1

Xs1=xd  +((x1*x2)/(x1+x2))
pe=E*V/Xs1
d0 = asin(Pi/pe)
dc= (%pi/2)-d0
dc=round(dc*1e3)/1e3

X1=x1;
X2=x2/2
X3=x2/2
Xs2=((X1*X2) + (X2*X3) + (X1*X3))/X3
pe2=E*V/Xs2



Xs3=xd+ x1
pe3=E*V/Xs3
d2 = asin(Pi/pe3)
dm=%pi- d2


if pe2<Pi then
```

```
    mprintf("UNSTABLE for sustained fault\n\n")
else
    mprintf("STABLE for sustained fault\n\n")
end


A1=((Pi * dc)+ ( pe2 * cos(dc)))- ((Pi * d0)+ ( pe2
    * cos(d0)))
A2=((Pi * dm)+ ( pe3 * cos(dm)))- ((Pi * dc)+ ( pe3
    * cos(dc)))

if abs(A1)<abs(A2) then
    mprintf("STABLE system\n\n")
else
    mprintf("UNSTABLE  system\n\n")
end


E=10
x=2
e=1e-3
a=Pi
b=pe3
c=-A1 +(Pi * dc ) + (cos (dc) * pe3)

while (E>e)
    f=(a*x) +  (b * cos(x)) - c
    df=1.5 - (2*sin(x))
    x1=x-(f/df)
    E=abs(x1-x)
    x=x1
end
d2=x1 * 180/%pi
mprintf("\ndelta 2 = %.2f deg",d2)

Pmb=pe2
Pmc=pe3
d0=round(d0*1000)/1000
```

```
dcc=acosd(((Pi*(dm-d0))- (Pmb*cos(d0))+ (Pmc*cos(dm)))/(Pmc-Pmb))
mprintf("\nCritical Clearing angle = %.1f deg", dcc)
```

Scilab code Exa 13.9 system stability and finding critical load angle in 3 phase fault at bus

```
clear
clc

xd=.2
x1=.4
x2=.4
Pi=1.5
E=1.2
V=1

Xs1=xd  +((x1*x2)/(x1+x2))
pe=E*V/Xs1
d0 = asin(Pi/pe)

Xs3=xd+ x1
pe3=E*V/Xs3
d2 = asin(Pi/pe3)
dm=%pi- d2

Pmb=0
Pmc=pe3
dcc=acosd(((Pi*(dm-d0))- (Pmb*cos(d0))+ (Pmc*cos(dm)))/(Pmc-Pmb))
mprintf("Critical Clearing angle = %.2f deg", dcc)
```

Scilab code Exa 13.10 system stability and finding critical load angle due to sudden loading

```
clear
clc

Pm=1
Pe1=.25
d1=round(asin(Pe1)*1000)/1000

Pe2=.5
d2=round(asin(Pe2)*1000)/1000

d3=0;

A1=((Pe2 * d2)+ ( 1 * cos(d2)))- ((Pe2 * d1)+ ( 1 *
    cos(d1)))

E=10
x=.811
e=1e-3
a=Pe2
b=1
c=A1 -((Pe2 * d2)+ ( 1 * cos(d2)))

while (E>e)
//for (i=1:4)
    f=(a*x) +  (b * cos(x)) + c
    df=a - (b*sin(x))
    x1=x-(f/df)
    disp(x1,f,df)
    E=abs(x1-x)
    x=x1
end
d3=x1 * 180/%pi

mprintf("\ndelta 3 = %.1f deg",d3)
```

Scilab code Exa 13.11 inertia constant of 2 generators in parallel

```
clear
clc

G1=50
H1=8

G2=100
H2=4
Gb=100

Ha=(H1*G1/Gb) + (H2*G2/Gb)
mprintf("\n(a)Ha= %d MJ/MVA", Ha)

Hb=Ha*2
mprintf("\n(b)Hb= %d MJ/MVA", Hb)

He= (Ha*Hb)/(Ha+Hb)
mprintf("\n(c)He= %.3f MJ/MVA", He)
```

Scilab code Exa 13.12 find frequency deviation in case of delay in opening steam valve

```
clear
clc

G=100
f=50
H=5
dL=50
t=.6
```

```
J = G*H*1e3;
dJ=dL*1e3*t
f2=sqrt((J-dJ)/J)*f
fd=(f-f2)/f;
mprintf("Freq deviation = %.3f percent", fd*1e2)
```

Scilab code Exa 13.13 find critical clearing angle during prefault fault and post fault conditions

```
clear
clc

Pi=1
Pma=1.75
Pmb=.4
Pmc=1.25

d0=asin(Pi/Pma)
dm=%pi - asin(Pi/Pmc)

dcc=acosd(((Pi*(dm-d0))- (Pmb*cos(d0))+ (Pmc*cos(dm)))/(Pmc-Pmb))
mprintf("Critical Clearing angle = %.1f deg", dcc)
```

Scilab code Exa 13.14 point by point solution of swing equation

```
clear
clc
clf
```

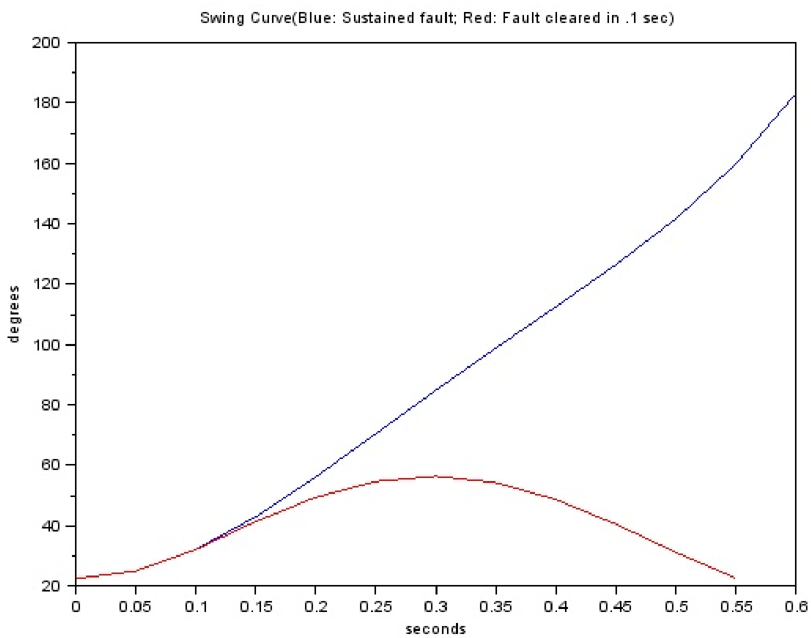

Figure 13.1: point by point solution of swing equation

```
Sb = 50;
S=50;
V =1;
Xd = 0.2;
X1 =0.4;
X2 = 0.4;
H = 2.7;
E=1.05 ;
G=1;

M = G*H/(180*50);

pe0 = (E*V/X1);
d0=asind(S/(Sb*pe0 ));
Pe0=pe0 * sind(d0);

pe1 = (E*V/(X1+X2+Xd));

pe2 = (E*V/(X1+Xd));

dt=.05
c_1=dt*dt/M

for i=1:14
    if i==1 then
        m_t(i)=0;
        m_Pm(i)=Pe0
        m_sind(i)=sind(d0)
        m_Pe(i)=S/Sb
        m_Pa(i)=0
        m_cPe(i)=c_1 * m_Pa(i)
        m_dd(i)=0
        m_d(i)=d0
    else if i==2 then
        m_t(i)=0;
        m_Pm(i)=pe1
        m_d(i)=d0
```

```
        m_sind(i)=sind(m_d(i))
        m_Pe(i)=m_sind(i)*m_Pm(i)
        m_Pa(i)=(1 - m_Pe(i) + m_Pa(i-1))/2
        m_cPe(i)=c_1 * m_Pa(i)
        m_dd(i)=0
    else
        m_t(i)=m_t(i-1) +dt;
        m_Pm(i)=pe1
        m_dd(i)=m_dd(i-1) + m_cPe(i-1)
        m_d(i)=m_d(i-1)+m_dd(i)
        m_sind(i)=sind(m_d(i))
        m_Pe(i)=m_Pm(i) * m_sind(i)
        m_Pa(i)=(1 - m_Pe(i))
        m_cPe(i)=c_1 * m_Pa(i)
    end
    end

end
res1(:,1)=m_t(:)
res1(:,2)=m_Pm(:)
res1(:,3)=m_sind(:)
res1(:,4)=m_Pe(:)
res1(:,5)=m_Pa(:)
res1(:,6)=m_cPe(:)
res1(:,7)=m_dd(:)
res1(:,8)=m_d(:)
res1=round(res1*1000)/1000
i=1
head=['' '' 't' '' '' 'Pm' '' '' 'sin d' '' '' 'Pe' '' '''Pa' '' '''8.33Pa' '' '''d delta' 'delta']
disp(res1, head, "(a)")
plot(m_t, m_d)
title('Swing Curve(Blue: Sustained fault; Red: Fault cleared in .1 sec)');
xlabel('seconds');
ylabel('degrees');
```

```
//(b)
while i<15
    if i==1 then
        m_t2(i)=0;
        m_Pm2(i)=Pe0
        m_sind2(i)=sind(d0)
        m_Pe2(i)=S/Sb
        m_Pa2(i)=0
        m_cPe2(i)=c_1 * m_Pa2(i)
        m_dd2(i)=0
        m_d2(i)=d0
    else if i==2 then
            m_t2(i)=0;
            m_Pm2(i)=pe1
            m_d2(i)=d0
            m_sind2(i)=sind(m_d2(i))
            m_Pe2(i)=m_sind2(i)*m_Pm2(i)
            m_Pa2(i)=(1 - m_Pe2(i) + m_Pa2(i-1))/2
            m_cPe2(i)=c_1 * m_Pa2(i)
            m_dd2(i)=0
        else
            m_t2(i)=m_t2(i-1) +dt;
            if m_t2(i) == .1 then
                m_Pm2(i)=pe1
                m_dd2(i)=m_dd2(i-1) + m_cPe2(i-1)
                m_d2(i)=m_d2(i-1)+m_dd2(i)
                m_sind2(i)=sind(m_d2(i))
                m_Pe2(i)=m_Pm2(i) * m_sind2(i)
                m_Pa2(i)=(1 - m_Pe2(i))
                m_cPe2(i)=c_1 * m_Pa2(i)

                i=i+1
                m_t2(i)=m_t2(i-1)
                m_Pm2(i)=pe2
                m_dd2(i)=m_dd2(i-1)
                m_d2(i)=m_d2(i-1)
                m_sind2(i)=sind(m_d2(i))
```

```
                    m_Pe2(i)=m_Pm2(i) * m_sind2(i)
                    m_Pa2(i)=(1 - m_Pe2(i) + m_Pa2(i-1))
                       /2
                    m_cPe2(i)=c_1 * m_Pa2(i)
                else
                    m_Pm2(i)=m_Pm2(i-1)
                    m_dd2(i)=m_dd2(i-1) + m_cPe2(i-1)
                    m_d2(i)=m_d2(i-1)+m_dd2(i)
                    m_sind2(i)=sind(m_d2(i))
                    m_Pe2(i)=m_Pm2(i) * m_sind2(i)
                    m_Pa2(i)=(1 - m_Pe2(i))
                    m_cPe2(i)=c_1 * m_Pa2(i)
                end
            end
        end
        i=i+1
end
res2(:,1)=m_t2(:)
res2(:,2)=m_Pm2(:)
res2(:,3)=m_sind2(:)
res2(:,4)=m_Pe2(:)
res2(:,5)=m_Pa2(:)
res2(:,6)=m_cPe2(:)
res2(:,7)=m_dd2(:)
res2(:,8)=m_d2(:)
res2=round(res2*1000)/1000
disp(res2,head, "(b)")

plot(m_t2, m_d2,'r')
//(c)
D0=d0 * %pi/180
Pi=1
Dm=%pi -  asin(Pi/pe2)

dcc=acosd(((Pi * (Dm-D0))-(pe1*cos(D0))+(pe2*cos(Dm)
   ))/(pe2 -pe1))
tcc=.395
mprintf("\n\n(c) dcc= %.1f deg; clearing time=%.3f
```

```
sec", dcc,tcc)
```

Scilab code Exa 13.15 find maximum load that can be supplied by generator

```
clear
clc

P=50
pe=100


E=10
x=1
e=1e-3

d0=30/180 * %pi
while (E>e)
    f=cos(d0)+cos(x) - ((%pi - d0-x)*sin(x))
    df= (-%pi+d0+x)*cos(x)
    x1=x-(f/df)

    E=abs(x1-x)
    x=x1
end
d1=x1 * 180/%pi

P1=sind(d1) *pe
Pr=P1-P
mprintf("\nPower Required = %.2f MW",Pr)
```

Scilab code Exa 13.16 finding steady state reactance and transfer limit for different shunt branches

```
clear;
clc

E=1.5
V=1
xg=1
xm=1

xt1=.1
xt2=.1
x1=.4
x2=.4
xl=(x1*x2)/(x1+x2)
X1=xl+xg+xt1
X2=xt2+xm
xL=.8
xC=-.8


X3=xC
X=((X1*X2) + (X2*X3) + (X1*X3))/X3
P1=E*V/X
mprintf("\n(a) Steady State limit = %.3f", P1)

X3=xL
X=((X1*X2) + (X2*X3) + (X1*X3))/X3
P2=E*V/X
mprintf("\n(b) Steady State limit = %.3f", P2)
```

Scilab code Exa 13.17 find inertia constant and momentum of generator

```
clear
clc

f=50
```

```
w=2*%pi*f
I=8800
pf=.85
J=.5*I*w*w*1e-6
P=60
MVA=P/pf
H=J/MVA
M=J/(180*f)

mprintf("Inertia const= %.3f MJ/MVA, Momentum= %.5f
   MJ-s/elec deg",H,M)
```

Scilab code Exa 13.18 find inertia constant of each machine and parallel ombination

```
clear
clc

f=50
w=2*%pi*f
I1=25000
pf1=.8
J1=.5*I1*w*w*1e-6
P1=45
G1=P1/pf1
H1=J1/G1
M1=J1/(180*f)

mprintf("machine 1 \nInertia const= %.2f MJ/MVA,
   Momentum= %.3f MJ-s/elec deg",H1,M1)

I2=9000
pf2=.75
J2=.5*I2*w*w*1e-6
P2=60
```

```
G2=P2/pf2
H2=J2/G2
M2=J2/(180*f)

mprintf("\n\nmachine 2 \nInertia const= %.2f MJ/MVA, Momentum= %.5f MJ-s/elec deg",H2,M2)

M=(M1*M2)/(M1+M2)
Gb=100
H=M*180 * f /Gb
mprintf("\n\n Equivalent constant at 100MVA base \nInertia const= %.4f MJ/MVA, Momentum= %.5f MJ-s/elec deg",H,fix(M*100000)/100000)
```

Scilab code Exa 13.19 find critical clearing angle and critical clearing time in 3 phase fault conditions

```
clear
clc

Pm=2
Pi=1
H=6
G=1
f=50
p=Pi/Pm
M=G*H/(%pi*f)
d0=asin(p)

dcc=acos(((p*(%pi - (2*d0)))- (Pi*cos(d0)))/(Pm-Pi))
mprintf("Critical Clearing angle = %.4f rad\n\n",dcc)

tcc=sqrt(2*M*(dcc-d0)/Pi)
mprintf("Critical Clearing time = %.3f sec = %.2f
```

```
cycles", tcc , tcc*50)
```

Scilab code Exa 19.20 finding acceleration torque and change in torque angle due to losses

```
clear
clc

f=50
G=20
V=13.2
H=9
nP=4

J=G*H
mprintf("\n(a)Stored Energy = %.0f MJ", J)
disp("The unit is incorrectly mentioned as "'mJ'' in
    the textbook.");

Pi= 25*.735
P=15
Pa=Pi-P
M=G*H/(180*f)
a=Pa/M
mprintf("\n(b)Acceleration = %.2f elec deg/sec sq",a
    )


c=15
t=c/f
dd=.5*a*t*t
mprintf("\n(c)change in angle = %.2f deg",dd )

A=a * 60 / (180*nP)
Ns=120*f/nP
```

```
N=Ns+(A*t)
mprintf("\n(d)New speed = %.3f rpm",fix(N*1000)/1000
    )
```

Chapter 16

Distribution

Scilab code Exa 16.1 find voltage at load points in single feeded dc feeder

```
clear
clc

I1=100
I2=150
I3=200

l1=150
l2=100
l3=100

r=.1/1000

Va=200

rac=l1*r;
rcd=l2*r
rbd=l3*r

dvc=(I1+I2+I3)*rac;
dvd=(I1+I2+I3)*rac + (I2+I3)*rcd ;
```

```
dvb=(I1+I2+I3)*rac + (I2+I3)*rcd + (I3*rbd);

Vc=Va-dvc
Vd=Va-dvd
Vb=Va-dvb

mprintf("voltage at B= %.2f V  C= %.2f V  D= %.2f V
    ", Vb, Vc, Vd)
```

Scilab code Exa 16.2 find voltage at load points in addition to distributed load in single feeded dc feeder

```
clear
clc

I1=100
I2=150
I3=200

l1=150
l2=100
l3=100

r=.1/1000
U=1

Va=200

rac=l1*r;
rcd=l2*r
rbd=l3*r

Iac= I1+I2+I3 + U*(l1+l2+l3)
dvc=(Iac - (.5* U * l1))*rac;
Vc=Va-dvc
```

```

Icd= I2+I3 + U*(l2+l3)
dvd=(Icd - (.5* U * l2))*rcd ;
Vd=Vc-dvd

Idb= I3 + U*(l3)
dvb=(Idb - (.5* U * l3))*rbd ;
Vb=Vd-dvb

mprintf("voltage at B= %.3f V C= %.3f V D= %.3f V
    ", Vb, Vc, Vd)
```

Scilab code Exa 16.3 find voltage at load points in doubly feeded dc feeder

```
clear;
clc

r1=.02
r2=.05
r3=.03

Ic1=100
Id1=180

Va=255;
Vb=250

dV=abs(Va-Vb)
Ia=(dV+(r1*0)+(r2*Ic1)+(r3*(Id1+Ic1)))/(r1+r2+r3)

Ib=-(Ia-(Ic1+Id1))

Vc=Va-Ia*r1
Vd=Vc-((Ia-Ic1)*r2)

```

```
mprintf("IA= %.0f A, IB=%.0fA, Vc=%.2f V, Vd=%.2f V", Ia, Ib, Vc,Vd)
```

Scilab code Exa 16.4 find voltage at load points in addition to distributed load in doubly feeded dc feeder

```
clear
clc

l=300
l1=120
l3=120
l2=l-l1-l3
U=.25
Ic1=40
Id1=60

r=.1/100
r1=l1*r;
r2=l2*r
r3=l3*r

Va=300
Vb=300

dV=abs(Va-Vb)
Ia=(dV+(r1*.5*U*l1)+(r2*.5*U*l2)+(r3*.5*U*l3)+(r2*(Ic1+U*l1))+(r3*(Ic1+U*l1 +Id1+U*l2)))/(r1+r2+r3)
I=Ic1+Id1+(U*l)
Ib=I-Ia

Vc=Va-(Ia-.5*U*l1)*r1
Vd=Vb-((Ib-.5*U*l3)*r3)

mprintf("IA= %.1f A, IB=%.1fA, Vc=%.2f V, Vd=%.2f V"
```

```
    , Ia, Ib, Vc,Vd)
```

Scilab code Exa 16.5 voltage drop in singly feeded ac feeder with concentrated load

```
clear
clc

l=400
l1=100
l2=250
l3=400
r=.25/1000

I1=100
I2=120
I3=80

V=240

dv=r*((l1*I1)+(l2*I2)+(l3*I3))
Ve=V-dv

mprintf("\nVoltage at end is %.0f V, drop = %.0f V",
    Ve, dv)
```

Scilab code Exa 16.6 voltage drop in singly feeded ac feeder with distributed and concentrated load

```
clear
clc

l1=100
```

```
l2=250
l3=400
r=.25/1000
l=.125/1000

I1=100
I2=120
I3=80

pf1=.7
pf2=1
pf3=.8

phi1=acos(pf1)
phi2=acos(pf2)
phi3=acos(pf3)

Z1=l1 * ((r*cos(phi1))+(l*sin(phi1)))
Z2=l2 * ((r*cos(phi2))+(l*sin(phi2)))
Z3=l3 * ((r*cos(phi3))+(l*sin(phi3)))

V=240

dv=(Z1*I1)+(Z2*I2)+(Z3*I3)
Ve=V-dv

mprintf("\nVoltage at end is %.2f V, drop = %.2f V",
    Ve, dv)
```

Scilab code Exa 16.7 currents in a 3 phase ac circuit

```
clear
clc

V=240
```

```

P1=50e3
P2=50e3
P3=200e3
Pm=500e3
pfm=.8


I1=P1/V
I2=P2/V
I3=P3/V
Im=Pm/(3*V*pfm)

i1=fix(I1*exp(%i *0*%pi/3)*100)/100
i2=fix((fix(I2*100)/100)*exp(%i *-2*%pi/3)*1000)/1000
i3=fix(I3*exp(%i *2*%pi/3)*100)/100
in=abs(round((i1+i2+i3)*100)/100)

iR=sqrt((I1)^2+(Im)^2+(2*I1*Im*pfm))
iY=sqrt((I2)^2+(Im)^2+(2*I2*Im*pfm))
iB=sqrt((I3)^2+(Im)^2+(2*I3*Im*pfm))

mprintf("\nIR= %.0f A", iR)
mprintf("\nIY= %.0f A", iY)
mprintf("\nIB= %.0f A", iB)
mprintf("\nIn= %.2f A", in)
```

Scilab code Exa 16.8 voltage drop at the end of one phase in unbalanced 3 phase network

```
clear
clc

V=230
```

```

I1=80
I2=70
I3=50

pf1=.8
pf2=.9
pf3=1

phi1=acos(pf1)
phi2=acos(pf2)
phi3=acos(pf3)

i1=I1*exp(%i *0*%pi/3) * exp(%i * -phi1)
i2=I2*exp(%i*-2*%pi/3) * exp(%i * -phi2)
i3=I3*exp(%i *2*%pi/3) * exp(%i * -phi3)

r=.2

in=i1+i2+i3

dvR=r*i1
dvN=r*in

VR=V+dvR+dvN

mprintf("VR= %.1f V ang (%.1f)deg V", abs(VR), atand
   (imag(VR)/real(VR)))
```

Scilab code Exa 16.9 find supply voltage and phase angle between sending end and receiving end

```
clear
clc

```

```
Vb=240

Ib=100* exp(%i *-1* acos (.8))
Ia=100* exp(%i *-1* acos (.6))

z=complex(.2, .3)

Va=round((Vb + (Ib * z/2)))
Isa = (Ia * exp(%i * 1 * atan(imag(Va)/real(Va)))) +
    Ib
//Isa=Isa*%i

Vs=Va + (Isa * z/2)
Vs=round(Vs*100)/100
vs=abs(Vs)
phi=atand(imag(Vs)/real(Vs))

mprintf("|Vs| = %.2f A, phase angle between Vs & Vb
    = %.2f deg", fix(vs*100)/100, phi)
```

Scilab code Exa 16.10 find currents in a hexagon shaped concentrated loads

```
clear
clc

I=200

r1=.05
r2=.06
r3=.02
r4=.04
r5=.03
r6=.01
ra=.02
```

```
rb=.03

I1=100
I3=30
I4=50
I5=20
//(a)
dv=0
A=[
(ra) (-rb) (r6+r5 +r4+r3)
1 1 0
(ra+r1+r2) (-rb) -(r1+r2)
]

B=[
dv+ (r5*(I5) +r4*(I5+I4)+r3*(I5+I4+I3))
I
dv+(r2*I1)
]
i=inv(A)*B

mprintf("\n(a) Ia = %.0f A, Ib= %.0f A", i(1), i(2))

//(b)
dv=-5
A=[
(ra) (-rb) (r6+r5 +r4+r3)
1 1 0
(ra+r1+r2) (-rb) -(r1+r2)
]

B=[
dv+(r5*(I5) +r4*(I5+I4)+r3*(I5+I4+I3))
I
dv+(r2*I1)
]
i=inv(A)*B

```

```
mprintf("\n(b) Ia = %d A, Ib= %d A", i(1), i(2))
```

Scilab code Exa 16.11 find point of minimum in a line

```
clear
clc

L=1200
L1=900
L2=600

r=1.5

x=(L1*L + L*L*r/2)/(L1 + L2 + (r*2*L/2))
y=L-x
mprintf("\nCurrent in CB= %.0f a A", x)
mprintf("\nCurrent in CA= %.0f a A", y)
```

Scilab code Exa 16.12 voltage at far end in a double ac conductor with concentrated load

```
clear
clc

V=250

L=[
150 200 280 320 390 450 500
20 40 35 25 10 20 30
]

r=.1/(2*500)

```

```
D=L(1,:)
I=L(2,:)'

dv=2*r*D*I

Ve=V-dv

mprintf("Voltage at far end is %.2f V", Ve)
```

Scilab code Exa 16.13 voltage at far end in a double ac conductor doubly fed with concentrated load

```
clear
clc

V=250

L=[
150 50 80 40 70 60 50 150
0 20 60 95 120 130 150 180
]

r=.1/(2*500)

D=L(1,:)
I1=L(2,:)'
I2=ones(8,1)

dv1=2*r*D*I1
dv2=2*r*D*I2

Ia=dv1/dv2
Ib=L(length(L))-Ia

Vc=V-(2*r*((Ia*D(1))+((Ia-I1(2))*D(2))+((Ia-I1(3))*D
```

```
    (3))))
mprintf("Ia= %.2f A, Ib= %.2f A, Vmin at C = %.3f V"
    ,Ia, Ib, Vc)
```

Scilab code Exa 16.14 find currents in a pentagon shaped concentrated loads with an interconnector

```
clear
clc


r1=.03
r2=.02
r3=.03
r4=.04
r5=.04
r6=.01
r7=.02

I1=20
I2=30
I3=25
I4=30
I5=125
I6=20

dv=0
A=[
(r5+r1+r4+r3+r2) (r3+r2)
(r3+r2) (r3+r2+r6+r7)
]


B=[
((r4*I5)+(r3*(I5-I4))+ (r2*(I5-I4-I3)) + (r1*I1))
```

```
((r3*(I5-I4))+ (r2*(I5-I4-I3))+(r7*I6))
]
i=inv(A)*B
x=i(1)
y=i(2)

mprintf("\nEA = %.2f A, AB= %.2f A, ED= %.2f A, DF=
   %.2f A, DC= %.2f A, BC= %.2f A, FB= %.2f A", x,x-
   I1, I5-x,y, I5-I4-x-y, I5-I4-I3-x-y, y-I6)
```

Scilab code Exa 16.15 find currents in a triangle shaped loads

```
clear
clc

z1=complex(2,1)
z2=complex(2,3)
z3=complex(1,2)

ib= 40 * exp (%i * -1 * acos (.8))
ic= 60 * exp (%i * -1 * acos (.6))

i1=((ib*z2)+((ib+ic)*z3))/(z1+z2+z3)
i2=i1-ib
i3=i2-ic

mprintf("\n Current A to B in Ampere")
disp(round(i1*100)/100)
mprintf("\n Current B to C in Ampere")
disp(round(i2*100)/100)
mprintf("\n Current A to C in Ampere")
disp(round(-i3*100)/100)
```

Scilab code Exa 16.16 find optimum cross section of cables for comsumers at different distances

```
clear
clc

//(a)
V=230
df=5/100

I1=20
I2=10
L1=300
L2=200
L=500
I=I1+I2

rho=.0286
T=20
dT=30
a=.004
rho1=rho*(1+(dT*a))
rho1=round(rho1*1000)/1000

A=1
A1=I1/I
A2=I2/I

A=rho1*2*((L*round(I/A))+(L1*round(I1/A1)))/(df * V)
A=round(A*100)/100

A1=A*A1
A2=A*A2

mprintf("\n(a)CROSSECTIONS: SC= %.2f mm sq, CA= %.2f
    mm sq, CB= %.2f mm sq",A, A1, A2 )

R=2*rho1 * (L/A)
```

```
R1=2*rho1 * (L1/A1)
R2=2*rho1 * (L2/A2)

P=(I*I*R)+(I1*I1*R1)+(I2*I2*R2)

PL=P*100/((V*I1)+(V*I2))
mprintf("\n(b)Percentage power loss = %.1f percent",
    PL )
```

Scilab code Exa 16.17 voltage at far end in a double ac conductor singly fed with uniform and concentrated load

```
clear
clc

I1=100
I2=50
I3=50
I4=100
I5=0

L=500
l1=50
l2=100
l3=100
l4=150
l5=L-l1-l2-l3-l4

R=.075
r=R/L
U=1

Va=230

r1=l1*r;
```

```
r2=l2*r
r3=l3*r
r4=l4*r
r5=l5*r

Ia= I1+I2+I3+I4+I5 + U*(l1+l2+l3+l4+l5)
dvab=(Ia - (.5* U * l1))*r1;
Vb=Va-dvab

Ib= Ia- (U*(l1)) - I1
dvbc=(Ib - (.5* U * l2))*r2;
Vc=Vb-dvbc

Ic= Ib- (U*(l2)) - I2
dvcd=(Ic - (.5* U * l3))*r3;
Vd=Vc-dvcd

Id= Ic- (U*(+l3)) - I3
dvde=(Id - (.5* U * l4))*r4;
Ve=Vd-dvde

Ie= Id- (U*(l4)) - I4
dvef=(Ie - (.5* U * l5))*r5;
Vf=Ve-dvef

mprintf("voltage at B= %.4f V  C= %.4f V  D= %.4f V
    E= %.2f V  F= %.1f V ", Vb, Vc, Vd,Ve, Vf)
```

Scilab code Exa 16.18 voltage at far end in a double ac conductor with concentrated load

```
clear
clc

Vc=220
```

```
Z=2*complex(.15, .2)
Z1=.5*Z
Z2=.5*Z

Ib= 60 * exp (%i * -acos (.9))
Ic= 100 * exp (%i * -acos (.8))

Vb=Vc + (Ic * Z2)
Iab=Ib + Ic

Va=Vb + (Iab * Z1)
Vs=abs(Va)

Is= abs(Iab)
pf=cosd(abs(atand(imag(Va)/real(Va))+abs(atand(imag(
   Iab)/real(Iab)))))

mprintf("\nsending end voltage  = %.1f V", Vs)
mprintf("\nsending end current  = %.2f A", Is)
mprintf("\nsending end pf  = %.3f", pf)
```

Scilab code Exa 16.19 find most economical center of distribution

```
clear
clc

ab=1000
bc=900
ac=600

A=500
B=750
C=450

```

```
theta=acos((ab^2 + ac^2 - bc^2)/(2*ac*ab))
am=ac * cos(theta)
cm=ac * sin(theta)

x= ((B*ab)+(C*am))/(A+B+C)
y= ((C*cm))/(A+B+C)

mprintf("Coordinates wrt A are (%.2f, %.0f)", x,y)
```

Scilab code Exa 16.20 find optimum cross section for double dc line doubly fed concentrated load

```
clear
clc

rho=1/58

l1=90
l2=90
l3=100
l4=80
l5=90
l6=90
l7=90

I1=50
I2=40
I3=20
I4=30
I5=45
I6=50

I=I1+I2+I3+I4+I5+I6
L=l1+l2+l3+l4+l5+l6+l7
Ia=((I1*l2)+((I1+I2)*l3)+((I1+I2+I3)*l4)+((I1+I2+I3+
```

```
    I4)*l5)+((I1+I2+I3+I4+I5)*l6)+((I1+I2+I3+I4+I5+I6
    )*l7))/L

Ib=I-Ia
V=230*2
V1=440

dv=V-V1

r=dv/((Ia*(l1+l2+l3+l4))-((I1*l2)+((I1+I2)*l3)+((I1+
    I2+I3)*l4)))

a=rho*2/r

mprintf("crossection area (in mm sq)= %.2f", a)
```

Scilab code Exa 16.21 find currents in a hexagon shaped concentrated loads in 2 line dc ring main

```
clear
clc

r1=.08
r2=.1
r3=.12
r4=.14
r5=.09
r6=.16

I1=20
I2=50
I3=25
I4=40
I5=30

```

```
x=((r2*I1)+(r3*(I1+I2))+(r4*(I1+I2+I3))+(r5*(I1+I2+
    I3+I4))+(r6*(I1+I2+I3+I4+I5)))/(r1+r2+r3+r4+r5+r6
    )

iab=x
ibc=iab-I1
icd=ibc-I2
ide=icd-I3
ief=ide-I4
ifa=ief-I5

iab=round(iab*10)/10
ibc=round(ibc*10)/10
icd=round(icd*10)/10
ide=round(ide*10)/10
ief=round(ief*10)/10
ifa=round(ifa*10)/10


mprintf("\n(a) Current A to B in Ampere = %.1f", iab
    )
mprintf("\n Current B to C in Ampere = %.1f", ibc)
mprintf("\n Current C to D in Ampere = %.1f", icd)
mprintf("\n Current D to E in Ampere = %.1f", ide)
mprintf("\n Current E to F in Ampere = %.1f", ief)
mprintf("\n Current F to A in Ampere = %.1f", ifa)



Vb=230+((r6*ifa)+(r5*ief)+(r4*ide))
mprintf("\n Minimum voltage at B, in Volts = %.3f\n\
    n", Vb)

//(b)
r7=.1


B=[
```

```
(r4+r5+r6) -r7
(r1+r2+r3) (r7+r1+r3+r2)
]

C=[
((r5*I5)+(r4*(I5+I4)))
((r2*(I5+I2+I3+I4))+(r1*(I1+I2+I3+I4+I5))+(r3*(I1+I2
    +I3)))
]

A= inv(B)*C
x=A(1)
y=A(2)
iab=I1+I2+I3+I4+I5 - x-y
ibc=iab-I1
icd=ibc-I2
iaf=x
ife=iaf-I5
ied=ife-I4
iad=y

mprintf("\n(b) Current A to B in Ampere = %.1f", iab
    )
mprintf("\n Current B to C in Ampere = %.1f", ibc)
mprintf("\n Current C to D in Ampere = %.1f", icd)
mprintf("\n Current E to D in Ampere = %.1f", ied)
mprintf("\n Current F to E in Ampere = %.1f", ife)
mprintf("\n Current A to F in Ampere = %.1f", iaf)
mprintf("\n Current A to D in Ampere = %.1f", iad)

Ve=230-((r6*iaf)+(r5*ife))
mprintf("\n Minimum voltage at E, in Volts = %.3f",
    Ve)
```

Scilab code Exa 16.22 kelvins law

```
clear
clc

V=33e3
L=10
Ce=.8
data=[
4e6 .8 10
2e6 .8 6
1e6 .8 8
]
rho=.0286*1e3
P=2000
q=0
for i=1:3
    I(i,1)=data (i,1)/(sqrt(3) * V * data(i,2))
    E(i,1)= I(i)^2 * rho*3* data (i,3) * 1e-3
    q=q+E(i)
end
Q=q*365*Ce

a=sqrt(Q/P)

mprintf("a=%.2f sq mm", fix(a*100)/100)
```

Scilab code Exa 16.23 find cross section of cable for given losses in a singly fed ac conductor

```
clear
clc

V=220

rho1=.0286
T=20
```

```
dT=30
a=.004
rho=rho1*(1+(dT*a))
rho=round(rho*1000)/1000

P1=5
ef1=.81
pf1=.77
I1= P1*735.5/(V*pf1*ef1)
I1=I1 * exp(%i *-1 * acos (pf1))
I1=round(I1 *1000)/1000
Ic=I1


P2=26
ef2=.87
pf2=.85
I2= P2*735.5/(V*pf2*ef2)
I2=round(I2*100)/100
I2=I2 * exp(%i *-1 * acos (pf2))
Ib=fix((I1+I2)*100)/100

P3=10
ef3=.83
pf3=.82
I3= P3*735.5/(V*pf3*ef3)
I3=I3 * exp(%i *-1 * acos (pf3))
Ia=I1+I2+I3

P=real (Ia * V )
PLa=round(.05*P)

l1=40
l2=35
l3=30

PL1=2 * (round(abs(Ia)*100)/100)^2 * rho * l1
PL2=2 * (fix(abs(Ib)*100)/100)^2 * rho * l2
```

```
PL3=2 * (round(abs(Ic)*100)/100)^2 * rho * l3

PL1=fix(PL1/100)*100
PL2=round(PL2/100)*100
PL3=round(PL3/10)*10


a=(PLa/(PL1+PL2+PL3))^-1

mprintf("a=%.2f sq mm", a)
disp("the difference in the result is due to error in calculation of abs(I2)=141.12 instead of 144.12")
```

Scilab code Exa 16.24 find loss factor load factor annual load loss and annual cost of lost energy

```
clear
clc

Pp=3e3;
Et=1e7
Plp=220
C=2.1

lf=(Et/8760)/Pp
lf=round(lf*100)/100
mprintf("\nLoad Factor = %.2f ", lf)

L=(.3*lf) + (.7*lf*lf)
L=round(L*1000)/1000
mprintf("\nLoss Factor = %.3f ", L)

Loss=L*Plp
mprintf("\nLoss = %.1f kW", Loss)
```

```
CLY=Loss*8760*C
mprintf("\nCost of Energy Loss = Rs %.1f ",CLY)
```

Scilab code Exa 16.25 effect of starting of induction motor on domestic load

```
clear
clc

xl=.1

st1=10
vt1a=66
vt1b=11
xt1=.1
n1=vt1b/vt1a

st2=5
vt2a=11
vt2b=3.3
xt2=.08
n2=vt2b/vt2a

st3=05
vt3a=11
vt3b=.415
xt3=.06
n3=vt3b/vt3a

Sm=5
pfm=.8

Sl=1
pfl=.8
```

```
Sb=10
Vb1=66
Vb2=Vb1*n1
Vb3=Vb2*n2
Vb4=Vb2*n3

Xt2=xt2* (Sb/st2)
Xt3=xt3* (Sb/st3)

Il=(Sl/Sb) * exp(%i * -acos(pfl))
Im=(Sm/Sb) * exp(%i * -acos(pfm))

It=Im+Il
Vt= 1- (It*(%i*(xt1+xl)))- (Il *%i* Xt3)

mprintf("\n(a) Voltage at domestic load is %.3f pu",
    abs(Vt))

Im=-1 * %i * 5 * Sm / Sb
It=Il +Im
Vt= 1- (It*(%i*(xt1+xl)))- (Il *%i* Xt3)

mprintf("\n(b) Voltage at domestic load is %.3f pu",
    abs(Vt))
```

Scilab code Exa 16.26 effect of adding capacitor bank on current and voltage

```
clear
clc

V=400
Im=40
pfm=.75
```

```
pfd=.95

Pm=sqrt(3)* V * Im * pfm * 1e-3
phi1=acos(pfm)
phi2=acos(pfd)
kvar1=Pm * tan ( phi1)
kvar2=Pm * tan ( phi2)
kvarC=kvar2-kvar1
KVarC=(abs(kvarC)*10)/10
Ic= KVarC* 1e3 /(3 * V)
Ic=round(Ic*1000)/1000
Xc=round(V*100/Ic)/100
C=1/(2*%pi*50 * Xc)

mprintf("\nkVAR of Capacitor Bank = %.1f KVAR, Phase
    Current Of Capacitor Bank=%.2f A, C= %.2f e-6 F/
    phase",KVarC, Ic, C*1e6)
```

Scilab code Exa 16.27 percentage change in losses by adding capacitor bank

```
clear
clc

V=400
Im=40
pfm=.75
pfd=.95

Pm=sqrt(3)* V * Im * pfm * 1e-3
phi1=acos(pfm)
phi2=acos(pfd)
kvar1=Pm * tan ( phi1)
kvar2=Pm * tan ( phi2)
kvarC=kvar2-kvar1
```

```
Ic= abs(kvarC)* 1e3 /(3 * V)
Xc=V/Ic
C=1/(2*%pi*50 * Xc)


Imx=Im * sin (acos(pfm))
Iline= (Im * pfm) - (%i * (Imx - (Ic*sqrt(3))))
dL=(1-(abs(Iline)/Im)^2)*100
mprintf("\npercentage reduction in power loss=%.2f",
   dL)
```

Scilab code Exa 16.28 rating of switched bank and fixed bank capacitors

```
clear
clc

t1=15
P1=1.3
pf1=.75

t2=9
P2=.4
pf2=.8

pfd=.95

kvar1i=P1*1e3*tan (acos(pf1))
kvar1f=P1*1e3*tan (acos(pfd))
kvarr1=kvar1i-kvar1f

kvar2i=P2*1e3*tan (acos(pf2))
kvar2f=P2*1e3*tan (acos(pfd))
kvarr2=kvar2i-kvar2f

SBC=abs(kvarr2-kvarr1)
```

```
FBC=min(kvarr2,kvarr1)

mprintf("Switch Bank Capacity: %.2f KVAR, Fixed Bank Capacity: %.2f KVAR",SBC, FBC)
```

Chapter 17

Design Of Transmission Lines

Scilab code Exa 17.1 Design Of Transmission Lines

```
clear;
clc;

P=100e3;
pf=.9;
Len=200;
eff=.95;
Vreg=.15;

//(a)
V=5.5 * sqrt((Len/1.6)+(P/100));
V=220;
Z0=400;
SIL= V * V/Z0;
n=(P/(SIL*1e3))+1;
mprintf("\n (a) Voltage rating= %.0f kV,\n no of circuits= %d",V, n);

//(b)
Ir= P/(sqrt(3)*V * pf);
r20=.302
```

```
Temp2=75
Temp1=20
r75=round(r20 * ((228+Temp2)/(228+Temp1)) *100)/100
R=r75*Len
eff=P*1e3 /((P*1e3) + (3*Ir*Ir*R))
mprintf("\n(b)\nACSR 6/6/4.50 gives efficiency %.2f.
    so not suitable", eff)

r20=.0898
r75=round(r20 * ((228+Temp2)/(228+Temp1)) *100)/100
R=r75*Len
eff=P*1e3 /((P*1e3) + (3*Ir*Ir*R))
mprintf("\nACSR 30/7/3.71 gives efficiency %.2f.
   Suitable for temp less than 75, span =300m (by
   experience)", eff)
span=300
dia=25.97
dAl=3.71
dSt=3.71

//(c)
mprintf("\n(c)Keep interphase distance to be 6m for
   220KV line. 12 m between 2 outer phases")
D1=6
D2=12

//(d)
Deq=(D1*D1*D2)^(1/3)
r=dia/2;
GMR=.7788 * r
GMR=round(GMR*100)/1e5
L=round(.4605 * log10(Deq/GMR)*100)/100
Z=round(complex(R, (2*%pi*50 * L *1e-3*Len)) *10)/10
C=.02412/ log10(Deq/GMR)
Y=%i * 2*%pi*50 * C *1e-6*Len

E1= round((1+((Z*Y)/2))*1000)/1000
E2=round((Y*(1+((Z*Y)/4)))*1e7)/1e7
```

```
Vr=V*1e3/sqrt(3)
pf=.9
Ir=Ir * exp(%i * -acos(pf))
Vs=(Vr * E1) + (Ir*Z)
Is=(Vr *Y* E2) + (Ir*E1)
//Error in answer (Ps) is due to mutiple rounding off in a step in the textbook (Is)
pfs=cos(atan(imag(Vs)/real(Vs))+atan(imag(Is)/real(Is)))
Ps=round(real(3*Vs*Is))/1000000
Ps=105.07
pfs=round(pfs*100)/100
eff=P*.1/Ps
Vr0=abs(Vs)/abs(E1)
VR=(Vr0-abs(Vr))/abs(Vr)
mprintf("\n(d)\nline efficiency= %.2f percent, Voltage regulation= %.2f percent",eff, VR*100)

//(e)
p=74;
t=50
d=3.86 * p/(273+50)
m0=.84
Vd=(3*1e6/sqrt(2)) * r *1e-3 * d * m0 * log(Deq/(r*1e-3))
ratio=V*1e3/(Vd*sqrt(3))
F=.05

corona=3* 21 * 1e-6 * 50 *(V/(sqrt(3))) *(V/(sqrt(3))) * F/(log10(Deq*1e3/r) *log10(Deq*1e3/r) )
corona=round(corona*100)/100
corona=corona * Len
mprintf("\n(e)Corona loss =%.1f KW",corona)

//(f)
tphi1=tan(acos(pf))
tphi2=tan(acos(pfs))
Q1=P*1e-3 *tphi1
```

```
Q2=P*1e-3 *tphi2
Cap=Q1-Q2
mprintf("\n(f)capacity of capacitor = %.2f MVAR leading",Cap)
//(g)
Vr=V*1e3/sqrt(3)
Vr=round(Vr)
Ir=(P*1e3/(3*Vr*pfs) ) * exp(%i * -acos(pfs))
Vs=(Vr * E1) + (Ir*Z)
Is=(Vr *Y* E2) + (Ir*E1)
//Error in answer (Ps) is due to mutiple rounding off in a step in the textbook (Is)
pfs=cos(atan(imag(Vs)/real(Vs))+atan(imag(Is)/real(Is)))
Ps=round(real(3*Vs*Is))/1000000
Ps=104.74
pfs=round(pfs*100)/100
eff=P*.1/Ps
Vr0=abs(Vs)/abs(E1)
VR=(Vr0-abs(Vr))/abs(Vr)
mprintf("\n(g)\nline efficiency= %.1f percent, Voltage regulation= %.2f percent",eff, VR*100)

//(h)
A=37 * %pi * (dAl/1000)^2 /4
E=91.4 *1e9

alpha=18.44 *1e-6
w=14.64
Fw=378 * dia * 1e-3
Fw=round(Fw*100)/100
Ft1=sqrt(w^2 + Fw^2)
T1=135.5*1e3/2.5
Ft2=w
Temp21=5
Temp22=30

c_1=1
```

```
c_2=T1 -(alpha * A * E * (Temp22-Temp21)) - A*E*Ft1^2 * span^2 /(24*T1^2)
c_3=0
c_4=A*E*Ft2^2 * span^2 /24
pol=poly([-c_4 -c_3 -c_2 c_1], "xx", "c")
T2s=roots(pol)


T2=T2s(1)
Sag1= w * span *span / (8 * T2)
Sag2= round(Ft1*100)*span *span / (800 * T1)
VS=Sag2 * cos (atan(Fw/w))
mprintf("\n(h)Tension = %.0f N, Sag under erection = %.2f m , vertical sag due to bad weather = %.2f m", T2, Sag1, VS)

//(i)
mprintf("\n(i)Using experience, use 2 ground wires of 7/3.66 mm galvanised steel wires")
```

Chapter 18

Power System Earthing

Scilab code Exa 18.1 resistance of grounding electrode

```
clear;
clc;

d=2e-2;
l=3;
rho=60;

//(a)
R=(rho/(2 * %pi * l)) * log (4* l / d);
mprintf("\n(a) Earthing resistance = %.2f ohm", R)

//(b)
r= rho / (2 * %pi * R);
mprintf("\n(b) Radius of hemspherical electrode = %.3f m", r)
```

Scilab code Exa 18.2 resistance of different arrangements of grounding electrode

```
clear;
clc;

d=2e-2;
l=3;
rho=60;
D=3;

R=(rho/(2 * %pi * l)) * log (4* l / d);
r= rho / (2 * %pi * R);

//(a)
a=r/D;
R2p=((1+a)/2)*R;
mprintf("\n(a) Resistance of 2 rods in parallel = %.3f ohm", R2p)

//(b)

R3p=((2+a-(4*a*a))/(6-(7*a)))*R;
mprintf("\n(b) Resistance of 3 rods in parallel = %.2f ohm", R3p)

//(c)
R3t=((1+a+a)/3)*R;
mprintf("\n(c) Resistance of 3 rods in equilateral triangle = %.1f ohm", R3t)

//(d)
R4s=((1+(2.707*a))/4)*R;
mprintf("\n(d) Resistance of 4 rods in square = %.2f ohm", R4s)
```

Scilab code Exa 18.3 earthing resistance of wire buried to different depths

```
clear;
clc;

l=3;
r=.25e-2;
rho=100;
h=.5;

//(a)
Ra=(rho/(%pi * l)) *(log(2*l/r)-1);
mprintf("\n (a) Resistance when buried at surface of
    earth = %.2f ohm", Ra)

//(b)
Rb=(rho/(%pi * l)) *(log(2*l/sqrt(2*r*h))-1);
mprintf("\n (a) Resistance when buried .5m under
   earth earth = %.2f ohm", Rb)

//(c)
Rc=Ra/2
mprintf("\n (c) Resistance when buried infinte deep
   = %.2f ohm", Rc)
```

Chapter 19

Voltage Stability

Scilab code Exa 19.1 finding sending and receiving end reactive power

```
clear;
clc;

V=400e3;
X=96;
Y=.001 * exp(%i * 90/180*%pi);
Sb=500e6;
Pr=2;

Z=complex(0,X);
Zb=V*V/Sb;
A=1+(Z*Y/2);
B=Z/Zb;



Vs=1;
Vr=.95;
d1=asin(Pr*abs(B)/(Vs*Vr));
Qr1=((Vs*Vr)/abs(B)) * cos(d1) - (abs(A)* Vr*Vr/abs(
    B))
Qs1= (abs(A)* Vs*Vs/abs(B)) - ((Vs*Vr)/abs(B)) * cos
```

```
(d1)

mprintf("\n(a) Qr= %.3f MVAR, Qs=%.3f MVAR", Qr1, Qs1);

Vs=1;
Vr=.9;
d2=asin(Pr*abs(B)/(Vs*Vr));
Qr2=((Vs*Vr)/abs(B)) * cos(d2) - (abs(A)* Vr*Vr/abs(B))
Qs2= (abs(A)* Vs*Vs/abs(B)) - ((Vs*Vr)/abs(B)) * cos(d2)

mprintf("\n(b) Qr= %.3f MVAR, Qs=%.3f MVAR", Qr2, Qs2);

Vs=1;
Vr=1;
d3=asin(Pr*abs(B)/(Vs*Vr));
Qr3=((Vs*Vr)/abs(B)) * cos(d3) - (abs(A)* Vr*Vr/abs(B))
Qs3= (abs(A)* Vs*Vs/abs(B)) - ((Vs*Vr)/abs(B)) * cos(d3)

mprintf("\n(c) Qr= %.3f MVAR, Qs=%.3f MVAR", fix(Qr3*1000)/1000, fix(Qs3*1000)/1000);
```

Scilab code Exa 19.2 compensating value of capacitor

```
clear;
clc;

V=10000;
P=12.5e6;
f=50;
```

```
Xl=4;


d=asin(P*Xl/(V*V));

VL=2*V*sin(d/2);
QL=VL^2/4;
Qc=QL/2;

C=Qc/(2*%pi*f*V*V);

mprintf("C=%.1f e-6F",C*1e6);
```

Scilab code Exa 19.3 find receiving end voltage if breaker opens suddenly

```
clear;
clc;

xs=.22;
xl=.15;
Sb=1000;
Vr=1;

X=xl+xs;

Pr=1;

pf=.8;
pfa=acos(pf);
Qr=Pr*tan(pfa);

Vs=complex(Vr + (X * Qr / Vr) , (X * Pr / Vr));
V=abs(Vs);

mprintf("Vr = %.2f ang (%.1f) deg pu",V, fix(atand(
```

```
imag(Vs)/real(Vs))*10)/10)
```

Scilab code Exa 19.5 capacity of SVS

```
clear;
clc;

vf=3;
VA=8000;

dQ=vf/100 * VA

mprintf("capacity of SVS= +- %d MVAR", dQ);
```

Scilab code Exa 19.6 voltage and pf of bus before compensation

```
clear;
clc;

V=220e3;
Zl=complex(.8, .2);

Xline=.2;
Xt=.05;
Sb=100e6;
Vb=220e3;
v=V/Vb;

X=Xline+ Xt;
I=conj(Zl/v)
phi1=atand(imag(I)/real(I))

Vbus=1+ I * X*exp(%i * %pi/2)
```

```
phi2=atand(imag(Vbus)/real(Vbus))

vbus=abs(Vbus)
vbus=round(vbus *1000)/1000
vbus=vbus*Vb*1e-3;
pf=cosd(-phi1+phi2)

mprintf("Voltage at bus = %.2f Kv, pf= %.3f lagging"
    , vbus, pf)
```

Scilab code Exa 19.7 voltage and pf of bus after compensation

```
clear;
clc;
//the data used is from Ex 19.6, not 19.5 as
    incorrectly mentioned in statement
V=220e3;
Zl=complex(.8,0);

Xline=.2;
Xt=.05;
Sb=100e6;
Vb=220e3;
v=V/Vb;

X=Xline+ Xt;
I=conj(Zl/v)
phi1=atand(imag(I)/real(I))

Vbus=1+( I * X*exp(%i * %pi/2))

phi2=atand(imag(Vbus)/real(Vbus))

vbus=abs(Vbus)*Vb*1e-3;
```

```
pf=cosd(-phi1+phi2)


mprintf("Voltage at bus = %.1f Kv, pf= %.2f", vbus, pf)
disp("the data used is from Ex 19.6, not 19.5 as incorrectly mentioned in statement")
```

Scilab code Exa 19.8 T parameters of compensated and uncompensated system

```
clear;
clc;

Z=complex(180*cosd(75) , 180*sind(75));
Y=complex(1e-3*cosd(90) , 1e-3*sind(90));

g=sqrt(Y*Z);
Zc=sqrt(Z/Y);
Zc=round(abs(Zc)) * exp (%i * atan(imag(Zc)/real(Zc)))

Z1=Zc * sinh(g);
Y1=(1/Zc) *( (cosh(g)-1)/sinh(g));
A=cosh(g);
B=round(abs(sinh(g) )*1000) * exp (%i * atan(imag(sinh(g) )/real(sinh(g) ))) * Zc/1000
C=B / (Zc *Zc)
D=A;


mprintf("\n(a)");

mprintf("\nA =D= %.4f ang (%.2f) deg",abs(A), round(atand(imag(A)/real(A))*100)/100)
```

```
mprintf("\nB = %.2f ang (%.2f) deg ohm",abs(B),round(atand(imag(B)/real(B))*100)/100)
mprintf("\nC = %.5f ang (%.2f) deg ohm",abs(C),round(atand(imag(C)/real(C))*100)/100 +180)


B2=B-(0.6 * imag(Z) *%i);
Y2=(cosh(g)-1)/(sinh(g)*Zc);

A2=1+(B2*Y2)
D2=A2;
C2=(2* Y2 )+ (B2*Y2*Y2);
mprintf("\n(b)");

mprintf("\nA =D= %.4f ang (%.2f) deg",abs(A2), round(atand(imag(A2)/real(A2))*100)/100)
mprintf("\nB = %.0f ang (%.2f) deg ohm",abs(B2),round(atand(imag(B2)/real(B2))*100)/100)
mprintf("\nC = %.3f ang (%.1f) deg ohm",abs(C2),round(atand(imag(C2)/real(C2))*100)/100 +180)
```

Scilab code Exa 19.9 pi parameters of compensated and uncompensated system

```
clear;
clc;

Z=complex(180*cosd(75) , 180*sind(75));
Y=complex(1e-3*cosd(90) , 1e-3*sind(90));
YZ=Z*Y;


B=Z;
A=1+(YZ/2)
D=A
```

```
C= Y* (1+(YZ/4))

mprintf("\n(a)");
mprintf("\nA =D= %.3f ang (%.2f) deg",abs(A), round(atand(imag(A)/real(A))*100)/100)
mprintf("\nB = %.0f ang (%.0f) deg ohm",abs(B),round(atand(imag(B)/real(B))*100)/100)
mprintf("\nC = %.6f ang (%.1f) deg ohm",abs(C),round(atand(imag(C)/real(C))*100)/100 +180)


B2=B-(0.6 * imag(Z) *%i);
YZ2=B2*Y;
A2=1+(YZ2/2)
D2=A2
C2= Y* (1+(YZ2/4))


mprintf("\n(b)");

mprintf("\nA =D= %.3f ang (%.2f) deg",fix(abs(A2)*1000)/1000, round(atand(imag(A2)/real(A2))*100)/100)
mprintf("\nB = %.2f ang (%.2f) deg ohm",abs(B2),round(atand(imag(B2)/real(B2))*100)/100)
mprintf("\nC = %.6f ang (%.1f) deg ohm",abs(C2),round(atand(imag(C2)/real(C2))*100)/100 +180)
```

Scilab code Exa 19.10 voltage regulation of compensated line

```
clear;
clc;

d=500;
z=complex(.105, .3768);
```

```
y=complex(0, 2.822e-6);
Z=z*d;
Y=y*d;
YZ=Y*Z;

A=1+(YZ/2)+((YZ)^2/24);
B=Z * (1+(YZ/6)+((YZ)^2/120));
C=Y * (1+(YZ/6)+((YZ)^2/120));
D=A;

Ys=-.6*Y;

As=1;
Ds=1;
Bs=0;
Cs=Ys;

Anew=A + (B*Cs);

Vr=round(220e3/sqrt(3));
Pr=40e6;
pf=.9;
pfa=-1*acos(pf);
Irm=Pr/(3*Vr);
Ir=complex(Irm*pf, Irm * sin(pfa));

Vs=(A*Vr)+(B*Ir);
Vr0=abs(Vs)/(round(abs(Anew)*100)/100);

VRc= (Vr0-Vr)*100/Vr
mprintf("\n(a) Voltage Regulation = %.2f percent",
    VRc);

Vr02=abs(Vs)/abs(A);

VRc2= (Vr02-Vr)*100/Vr
mprintf("\n(b) Voltage Regulation(uncompensated) = %
    .2f percent",VRc2);
```

Scilab code Exa 19.11 find var injection to bring voltage to original value

```
clear;
clc;

v1=220;
v2=132
vb1=220;
n=132/220
vb2=vb1*n

Sb=200;
Zb= vb2 *vb2/Sb;

x1=75;
x2=70;
x3=90

X1=x1/Zb;
X2=x2/Zb;
X3=x3/Zb;
X1=fix(X1*100)/100
X2=fix(X2*100)/100
X3=fix(X3*100)/100



Xt1=.08
Xt2=.08

X=((Xt1+X1)^-1 +(Xt2+X2)^-1 +(X3)^-1 )^-1;
X=fix(X*1000)/1000

dV=4
```

```
dS= Sb/X;
dQ=round(dS*1000/vb2)*1e-3 * dV

mprintf("\n(a)\n X1= %.2f pu; X2= %.2f pu; X3= %.2f
    pu", X1, X2, X3)
mprintf("\n(b)\n VAR injected = %.2f MVAR",dQ)
```

Chapter 20

Reliability of Transmission and Distribution Systems

Scilab code Exa 20.1 find failure rate of system down time per outage annual outage

```
clear;
clc;

l=.4;
Ft=.8;
Fd=8.2;
Tt=6;
Td=5;

Rf=Ft + (Fd*l);
mprintf("\n(a)Failure Rate = %.2f outages/year",Rf);

t=((Ft*Tt)+(Fd*Td*l))/Rf;
mprintf("\n(b)Down Time = %.3f Hours per outage",t);;

T=t*Rf;
mprintf("\n(c)Total Outage Time = %.1f Hours per
```

```
year",T);;
```

Scilab code Exa 20.2 find reliability of series connected elements

```
clear;
clc;

r1=.95;
r2=.92;
r3=.98;
r4=.88

R=r1*r2*r3*r4;
mprintf("\nNet system reliability = %.4f ",R);
```

Scilab code Exa 20.3 find reliability of parallel connected elements

```
clear;
clc;

r1=.95;
r2=.92;
r3=.98;
r4=.88

R=r1*r2*r3*r4;

Q=(1-R)*(1-R);
Rs=1-Q;
mprintf("\nNet system reliability = %.4f ",Rs);
```

Scilab code Exa 20.4 find reliability of series connected elements and MTTF

```
clear;
clc;

na=3;
nb=2;
nc=1;
Fa=3e-3;
Fb=2e-3;
Fc=4e-3;
t=20;


F=(Fa*na)+(Fb*nb)+(Fc*nc);
R=exp(-1*F*t);
MTTF=1/F;
mprintf("\nReliability = %.5f \nMTTF = %.3f HOURS",R
    ,fix(MTTF*1000)/1000);
```

Scilab code Exa 20.5 find reliability of parallel connected elements and MTTF

```
clear;
clc;

r1=.8;
r2=.86;
r3=.92;

R=r1*r2*r3;
mprintf("\nNet system reliability = %.3f ",R);
```

Scilab code Exa 20.6 find improved reliability

```
clear;
clc;

r1=.8;
r2=.86;
r3=.92;

Rs=r1*r2*r3;
Q=(1-Rs)*(1-Rs);
R=1-Q;
mprintf("\nNet system reliability = %.3f ",R);
```

Scilab code Exa 20.7 find improved reliability

```
clear;
clc;

r1=.8;
r2=.86;
r3=.92;

Q1=(1-r1)^2;
R1=1-Q1;

Q2=(1-r2)^2;
R2=1-Q2;

Q3=(1-r3)^2;
R3=1-Q3;

R=R1*R2*R3;


```

```
mprintf("\nNet system reliability = %.3f ",R);
```

Scilab code Exa 20.8 find number of components to achieve desired reliability

```
clear;
clc;

r=.3
R=.85;
n=log(1-R)/log(1-r);
N=round(n);
if(N<n)
    N=N+1;
mprintf("\nComponents required = %d",N)
Rs=1-((1-.3)^N);
mprintf("\nNet System Reliability = %.3f",Rs)
```

Scilab code Exa 20.9 find failure rate of system down time per outage annual outage

```
clear;
clc;

Oat1=0.1;
Oacb=0.15;
Oafd=1.2;
Oat2=0.3;

Obt1=0.1;
Obcb=0.15;
Obfd=5.2;
Obt2=0.3;
```

```
Oct1=0.5;
Occb=0.4;
Ocfd=2;
Oct2=0.6;

Rat1=6;
Racb=4;
Rafd=5;
Rat2=5;

Rct1=8;
Rccb=5;
Rcfd=3;
Rct2=4;

N=120;
S=3;

Of=((N/(N+S))*Oafd)+((S/(N+S))*Obfd);
Oafe=Oat1+Oacb+Of+Oat2;
Ocfe=Oct1+Occb+Ocfd+Oct2;

Rae=((Oat1*Rat1)+(Oacb*Racb)+(Of*Rafd)+(Oat2*Rat2))/Oafe;
Rce=((Oct1*Rct1)+(Occb*Rccb)+(Ocfd*Rcfd)+(Oct2*Rct2))/Ocfe;

R=Oafe+Ocfe;
mprintf("\n(a)Annual Outage Rate = %.3f outage per year",R);

TperO=((Oafe*Rae)+(Ocfe*Rce))/R;
mprintf("\n(b)Downtime per Outage = %.2f hours",TperO);

T=TperO*R;
mprintf("\n(a)Total outage time per year = %.2f
```

```
hours per year",fix(T*100)/100);
```

Scilab code Exa 20.10 find failure rate of system down time per outage annual outage

```
clear;
clc;

On=3;
Os=9
Oa=1
rn=6
ra=8;
n=110;
s=4;

y=24*365;
Rn=rn/y;
Ra=ra/y;
N=n/y;
S=s/y;

Ofe=((N/(N+S))*((On*On*2*Rn)+((S/N)*2*On*Os*Rn)+((S/N)*2*On*Os*Rn)+((2*S*S/N)*(Os*Os))));
Oes=2*(Oa*Ra*On);

Rfe=Rn*Rn*y/(2*Rn);
Res=Rn*Ra*y/(Ra+Rn);

R=Ofe+Oes;
mprintf("\n(a) Annual Outage rate= %.4f outages per year", fix(R*10000)/10000);

TO=(Ofe*Rfe)+(Oes*Res);
mprintf("\n(b) Total outage time per year = %.2f
```

```
    hours per year",TO);

TperO=TO/R;
mprintf("\n(c) Downtime per Outage = %.1f hours",
    TperO);
```