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

July 13, 2017

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

Book Description

Title: 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

List of Figures

Chapter 2

Line Parameters

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

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

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

```
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

```
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

```
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

```
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

```
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

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

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;
```

```
15 C=round(C*100)/100
16 mprintf("Capacitance per phase = %.2f e-6F/phase\n",
     C);
17 Xc=1/(2 * %pi * f * C *1e-6);
18 mprintf("Capacitive reactance per phase = %.0f ohms/
     phase\n",Xc);
19 I=2 * %pi * f * C *1e-6 * V / sqrt(3);
20 I=round(I*100)/100
21 mprintf("Charging current = %.2f A/phase\n",I);
22 MVA=sqrt(3)*V *I *1e-6;
23 mprintf("Charging MVA = %.2f MVA\n",fix(MVA*100)
     /100);
```

Scilab code Exa 2.11 capacitance of bundled conductors

```
capacitance of bundled conductors
```

```
1  clear;
2  clc;
3
4  r= 1.6e-2;
5  d= 45e-2;
6  D=12;
7  Dscb=sqrt(r*d);
8  Deq=(D*D*(2*D))^(1/3);
9  Cn= .02412/(log10 (Deq/Dscb));
10 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

```
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

capacitance of double circuit three phase lines

```
clear

clc;

r=.9e-2;
d=6;
```

```
7  D1=6;
8  D2=7;
9
10 Daa=sqrt ((d*d)+(D1*D1));
11 Daa=round(Daa*1e3)/1e3
12 Dbb=D2;
13 Dcc=Daa;
14 Deq=5.074;
15
16 Dsc_a = round(sqrt(r*Daa)*1e4)/1e4;
17 Dsc_b = round(sqrt(r*Dbb)*1e4)/1e4;
18 Dsc_c = round(sqrt(r*Dcc)*1e4)/1e4;
19
20 Dsc=(Dsc_a*Dsc_b*Dsc_c)^(1/3);
21 Cn=.02412/log10(Deq/Dsc);
22
23 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

```
capacitance of conductor taking neutral into account
```

```
1  clear;
2  clc;
3
4  h=5.5;
5  Dia=1.213e-2;
6  d=1.25;
7  l=10;
8
9  r=Dia/2;
10 Cn=(2*.01206)/(log10 (d/(Dia* 0.5 * sqrt(1 + (d*d
     *.25/h^2)))));
11 C=Cn * l;
```

```
12
13 mprintf("Capacitance per conductor= %.5f e-6 F/
      conductor\n",C);
```

Scilab code Exa 2.15 resistance at 20 and 50 deg C

`resistance at 20 and 50 deg C`

```
1  clear
2  clc;
3
4  dia=4.22e-3;
5  n=6;
6  T1=20;
7  T2=50;
8  rho1 = 2.826e-8;
9  a=.004;
10 dL=1.5e-2;
11 L=1000;
12
13 A=%pi * n *dia *dia / 4;
14 R1= rho1 * L * (1+dL) / A;
15 R2= R1 * (1+ (a * (T2-T1)));
16
17 mprintf("Temperature at %d = %.4f ohm/km\n", T1, R1)
      ;
18 mprintf("Temperature at %d = %.4f ohm/km", T2, R2);
```

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

`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

inductance of conductors in horizontal plane

```
clear;
clc;

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

```
6
7
8 //(b)
9
10 L= .4605 * log10(1.375 * sqrt(d/r));
11 mprintf("(b) L= %.2f mH/km", L);
```

Scilab code Exa 2.18 inductance of conductors in horizontal plane

inductance of conductors in horizontal plane

```
1 clear;
2 clc;
3 clear;
4
5
6 dia=26.88;
7 d=450
8 d1=15.25*1e3;
9
10 R=dia/2;
11 R1= .7788 * R;
12 R1=round(R1*100)/100
13 GMR= 1.09 * (R1 * d*d*d)^(.25);
14 GMR=round(GMR*10)/10
15 GMD=(d1* d1 * (2*d1))^(1/3);
16 GMD=round(GMD/10)*10
17 L=.4605 * log10 (GMD/GMR);
18
19 mprintf(" L= %.4f mH/km", L);
```

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

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

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

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);
```

```
17 XL=2 * %pi * f * L *1e-3;
18 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

convert to per unit system at common base

```
1  clear;
2  clc;
3
4
5  sg1=10e6;
6  vg1=13.2e3;
7  sg2=15e6;
8  vg2=13.2e3;
9  sm1=8e6;
10 vm1=12.5e3;
11 sm2=12e6;
12 vm2=12.5e3;
13 Xg=15;
14 Xm=20;
15
16 sb=50e6;
17 vb=13.8e3;
```

```
18
19 xg1=Xg * (vg1/vb)^2 * (sb/sg1);
20 xg2=Xg * (vg2/vb)^2 * (sb/sg2);
21 xm1=Xm * (vm1/vb)^2 * (sb/sm1);
22 xm2=Xm * (vm2/vb)^2 * (sb/sm2);
23  mprintf ("\nReactance of Generator 1= %.2f percent"
      ,xg1);
24  mprintf ("\nReactance of Generator 2= %.2f percent"
      ,xg2);
25  mprintf ("\nReactance of Motor 1= %.2f percent",xm1
      );
26  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

convert to per unit system at common base with neutral resistance present

```
1  clear;
2  clc;
3
4  vg=11e3;
5  sg=90e6;
6  xg=.25;
7
8  st1=100e6;
9  vt1a=10e3;
10 vt1b=132e3;
11 nt1=vt1a/vt1b;
12 xt1=.06;
13
14 st2=30e6 *3;
15 vt2a=66e3 * sqrt(3);
16 vt2b=10e3;
```

```
17 nt2=vt2a/vt2b;
18 xt2=.05;
19
20 sm1=50e6;
21 vm1=10e3;
22 xm1=.2;
23
24 sm2=40e6;
25 vm2=10e3;
26 xm2=.2;
27
28 xl=100;
29
30 Sb=sg;
31 Vbg=vg;
32
33 Xg=xg * (vg/Vbg)^2 * (Sb/sg);
34 Xt1=xt1 * (vt1a/Vbg)^2 * (Sb/st1);
35
36 Vbl=Vbg/nt1;
37 Xl=xl * (Sb) / (Vbl)^2 ;
38
39 Vbm=Vbl/nt2;
40
41 Xt2=xt2 * (vt2b/Vbm)^2 * (Sb/st2);
42 Xm1=xm1 * (vm1/Vbm)^2 * (Sb/sm1);
43 Xm2=xm2 * (vm2/Vbm)^2 * (Sb/sm2);
44
45  mprintf ("\nReactance of Generator = %.2f ",Xg);
46  mprintf ("\nReactance of Transformer 1= %.4f ",Xt1)
       ;
47  mprintf ("\nReactance of Line = %.3f ",Xl);
48  mprintf ("\nReactance of Transformer 2= %.3f ",Xt2)
       ;
49  mprintf ("\nReactance of Motor 1= %.3f ",Xm1);
50  mprintf ("\nReactance of Motor 2= %.3f ",Xm2);
```

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

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];
```

```
31
32 mprintf("\nX1= %.2f  pu",X(1,1));
33 mprintf("\nX2= %.2f  pu",X(2,1));
34 mprintf("\nX3= %.2f  pu",X(3,1));
```

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

```
find voltage regulation and capacitor required to make voltage regulation 0
```

```
1  clear;
2  clc;
3
4  d=15;
5  Vr=11e3/sqrt(3);
6  pfr=.8;
7  Pd=5e6;
8  Pl=.12*Pd;
9  l=1.1e-3;
10 L=l*d;
11
12 I= Pd/(3*pfr*Vr);
13 R=Pl/(3*I*I);
14 X=2 * %pi* 50 *L;
15 pfa=acos(pfr);
16 Vs=Vr + (I * R * pfr) + (I * X * sin(pfa));
17 vs=sqrt(3)*Vs;
18
19 VR=(Vs-Vr)/Vr;
20
21 mprintf("\n(a) Voltage Regulation = %.2f percent ",
      VR*100);
22
23 pfa0=atan(R/X);
24 pf0=cos(pfa0);
```

```
25
26 mprintf("\n(b) pf at VR=0 = %.3f ", pf0);
27
28 I0= (I* pfr)/pf0;
29 Ic= (I * sin(pfa))+(I0*sin(pfa0));
30 Xc=Vr/Ic
31 C=1/(100*%pi*Xc);
32
33 mprintf("\n(c) C = %.1f e-6 F", C*1e6);
```

Scilab code Exa 3.5 receiving end voltage and current

receiving end voltage and current

```
1 clear;
2 clc;
3
4 Vs=11e3/sqrt(3);
5 Pd=1200e3;
6 R=5.31;
7 X=5.54;
8 pfr=.8;
9
10
11 VIr= Pd/(3*pfr);
12 a=1;
13 b=-1*Vs;
14 pfa=acos(pfr)
15 c=(VIr * R * pfr) + (VIr * X * sin(pfa))
16
17 Vr=(-b+sqrt(b^2 - (4*a*c)))/(2*a);
18
19 I=VIr/Vr;
20 vr=sqrt(3)*Vr;
21
```

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

Scilab code Exa 3.6 receiving end voltage and current

receiving end voltage and current

```
1  clear;
2  clc;
3
4  Pd=1e6;
5  pf=.8;
6  v1=30e3;
7  v2=10e3;
8  Rl=25;
9  Xl=12;
10 rt=.8;
11 xt=2.5;
12 n=v1/v2;
13
14 Rt=rt*(n^2);
15 Xt=xt*(n^2);
16
17 R=Rt+Rl;
18 X=Xt+Xl;
19
20 Vr=v1/sqrt(3);
21 I=Pd/(3*Vr*.8);
22
23 pfa=acos(pf)
24 Vs=Vr + (I * R * pf) + (I * X * sin(pfa));
25 vs=sqrt(3)*Vs*1e-3;
26
27 VR=(Vs-Vr)/Vr;
28
```

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

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

```
determine per phase R and X for given efficiency
```

```
1  clear;
2  clc;
3
4  Vs=33e3/sqrt(3);
5  Vr=30e3/sqrt(3);
6  Pr=10e6;
7  pf=.8;
8  eff=.96;
9
10 I=Pr/(3*Vr*pf);
11
12 Ps=Pr/eff;
13 Pl=Ps-Pr;
14
15 R=Pl/(3*I*I);
16 pfa=acos(pf);
17 X=((Vs-Vr)-(I*R*pf))/(I*sin(pfa));
18
19 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

```
receiving end voltage and current power factor and Voltage regulation using nomina
```

```
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

```
receiving end voltage and current power factor and Voltage regulation using nomina
```

```
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;
```

```
33 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

```
receiving end voltage and current power factor and Voltage regulation using nomina
```

```
1  clear;
2  clc;
3
4  d=100;
5  f=50;
6  r=.153;
7  l=1.21e-3;
8  c=.00958e-6;
9  xl=2*%pi*f*l;
10 xc=1/(2*%pi*f*c);
11 Vr=110e3/sqrt(3);
12 Pr=20e6;
13 pf=.9;
14 pfa=-1*acos(pf);
15 Irm=Pr/(3*Vr*pf);
16 Ir=complex(Irm *pf, Irm * sin(pfa));
17
18 Z=complex(r,xl) * d;
19 Y=complex(0,1/xc) *d;
20
21 //disp(abs(Z),Y,abs(Ir));
22
23 Vs=(Vr*(1+((Z*Y)/2)))+ (Ir*Z);
24 V=abs(Vs);
25 vs=sqrt(3)*V*1e-3;
26 mprintf("\nSending End Voltage = %.2f kV",vs);
27
28 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

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));
```

```
14 D=A;
15
16 A=round(abs(A)*10000)/10000 * exp(%i * round(atan(
      imag(A)/real(A))*100)/100)
17 B=round(abs(B)*1000)/1000   * exp(%i * round(atan(
      imag(B)/real(B))*100)/100)
18
19 //see(B)
20
21
22 Vr=220e3/sqrt(3); //incorrectly taken as 127021 in
      textbook.
23 Vr=round(Vr)
24 Pr=40e6;
25 pf=.9;
26 pfa=-1*acos(pf);
27 Irm=Pr/(3*Vr);
28 Ir=complex(Irm *pf, Irm * round(sin(pfa)*100)/100);
29
30 Vs=(A*Vr)+(B*Ir);
31 V=abs(Vs);
32 vs=sqrt(3)*V*1e-3;
33 phi1=atand(imag(Vs)/real(Vs));
34 mprintf("\nSending End Voltage = %.2f kV",vs);
35
36 Is=(C*Vr)+(D*Ir);
37 I=abs(Is);
38 phi2=atand(imag(Is)/real(Is))
39 mprintf("\nSending End Current = %.2f A",I);
40
41 phi=phi2-phi1;
42 pfs=cosd(phi);
43 mprintf("\nSending End Power factor = %.3f  ",pfs);
44 mprintf("\nSending End Power Angle = %.3f  ",phi1);
45
46 MVA=sqrt(3) * vs* I /1000;
47 mprintf("\nSending End Power = %.3f  ",MVA);
48
```

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

Scilab code Exa 3.12 find OC receiving end parameters

find OC receiving end parameters

```
1  clear;
2  clc;
3
4  d=500;
5  z=complex(.105, .3768);
6  y=complex(0, 2.822e-6);
7  Z=z*d;
8  Y=y*d;
9  YZ=Y*Z;
10
11 A=1+(YZ/2)+((YZ)^2/24);
12 B=Z * (1+(YZ/6)+((YZ)^2/120));
13 C=Y * (1+(YZ/6)+((YZ)^2/120));
14 D=A;
15
16 Vr=220e3/sqrt(3);
17 Pr=40e6;
18 pf=.9;
19 pfa=-1*acos(pf);
20 Ir=0;
21
22 Vs=(A*Vr)+(B*Ir);
23 V=abs(Vs);
24 vs=sqrt(3)*V*1e-3;
25 phi1=atan(imag(Vs)/real(Vs))*(180/%pi);
26 mprintf("\nSending End Voltage = %.2f kV",vs);
27
28 Is=(C*Vr)+(D*Ir);
```

```
29 I=abs(Is);
30 phi2=atan(imag(Is)/real(Is))*(180/%pi);
31 mprintf("\nSending End Current = %.1f A",I);
32
33 phi2=phi2+180;
34 phi=phi1-phi2;
35 pfs=cosd(phi);
36 mprintf("\nSending End Power factor = %.4f  ",pfs);
```

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

find characteristic impedance propagation constant and ABCD for line

```
1  clear;
2  clc;
3
4  Z=complex(14.1, 51.48);
5  Y=complex(0, 1.194e-3);
6  Zc=sqrt(Z/Y);
7  g=sqrt(Z*Y);
8  A=cosh(g);
9  B=Zc* sinh(g);
10 C=sinh(g)/Zc;
11 D=A;
12
13 mprintf("\nZc= %s", string(round(abs(Zc)*1000)/1000)
       +'/_'+ string(round(atand(imag(Zc)/real(Zc))
      *100)/100) )
14 mprintf("\npropgation const= %s", string(round(abs(g
      )*1000)/1000) +'/_'+ string(round(atand(imag(g)/
      real(g))*100)/100) )
15 mprintf("\nA= %s", string(round(abs(A)*1000)/1000) +
      '/_'+ string(round(atand(imag(A)/real(A))*100)
      /100) )
```

```
16 mprintf("\nB= %s", string(round(abs(B)*1000)/1000) +
     '/_'+ string(round(atand(imag(B)/real(B))*100)
     /100) )
17 mprintf("\nC= %s", string(round(abs(C)*1000)/1000) +
     '/_'+ string(round(atand(imag(C)/real(C))*100)
     /100 +180) )
18 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

find receiving end voltage and current

```
1  clear;
2  clc;
3
4  Z=complex(200*cosd(80), 200*sind(80));
5  Y=complex(.0013*cosd(90),.0013*sind(90));
6  YZ=Y*Z;
7
8  A=1+(YZ/2)+((YZ)^2/24);
9  B=Z * (1+(YZ/6)+((YZ)^2/120));
10 phiA=atan(imag(A)/real(A))*(180/%pi);
11 phiB=atan(imag(B)/real(B))*(180/%pi);
12
13 P=60e6
14 pf=.8;
15 Vs=round(220/sqrt(3))*1e3;
16 VIrm=P/(3*pf)
17 pfa=acos(pf);
18 VIr=complex(VIrm *pf, VIrm * sin(pfa));
19
20 pfa=pfa*(180/%pi);
21 a=(round(abs(A)*1000)/1000)^2;
```

```
22 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;
23 c=abs(B)^2 * (VIrm)^2;
24 Vr=sqrt((-b+sqrt((b*b)-(4*a*c)))/(2*a));
25
26 vr=sqrt(3) * Vr / 1000;
27 Ir=VIr/(Vr*pf);
28 mprintf("Receiveing End Line voltage= %.0f kV", fix(
     vr));
29 mprintf("\n Receiveing End Line Current= %.0f A", Ir
     );
```

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

finding and comparing pi and T network parameters

```
1 clear;
2 clc;
3
4 Z=complex(180*cosd(75) , 180*sind(75));
5 Y=complex(1e-3*cosd(90) , 1e-3*sind(90));
6
7 g=sqrt(Y*Z);
8 Zc=sqrt(Z/Y);
9
10 Z1=Zc * sinh(g);
11 Y1=(1/Zc) *1e3*( (cosh(g)-1)/sinh(g));
12
13
14 mprintf("\nZpi= %s", string(round(abs(Z1)*100)/100)
     +'/_'+ string(round(atand(imag(Z1)/real(Z1))*100)
     /100) )
15 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

```
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);
```

```
30 phi2=atan(imag(Is)/real(Is))*(180/%pi);
31 phi=phi1-phi2;
32 pfs=cosd(phi);
33 mprintf("\nSending End pf= %.3f ",pfs)
34
35 Ps=sqrt(3)* vs * I * pfs /1000;
36 mprintf("\nSending End Power= %.1f ",Ps)
37
38
39 //(b)
40 Zc=sqrt(Z/Y);
41 g=sqrt(Z*Y);
42 A=cosh(g);
43 B=Zc* sinh(g);
44 C=sinh(g)/Zc;
45 D=A;
46
47 Vs=(A*Vr)+(B*Ir);
48 V=abs(Vs)
49 vs=V*sqrt(3)*1e-3;
50 mprintf("\n\n\n(b)\nSending End Voltage= %.1f kV",
     vs)
51
52 Is=(C*Vr)+(D*Ir);
53 I=abs(Is)
54 mprintf("\nSending End Current= %.1f A", fix(I*10)
     /10)
55
56 phi1=atan(imag(Vs)/real(Vs))*(180/%pi);
57 phi2=atan(imag(Is)/real(Is))*(180/%pi);
58 phi=phi1-phi2;
59 pfs=cosd(phi);
60 mprintf("\nSending End pf= %.3f ",pfs)
61
62 Ps=sqrt(3)* vs * I * pfs /1000;
63 mprintf("\nSending End Power= %.1f ",Ps)
```

Scilab code Exa 3.17 ABCD parameters of pi network

```
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

```
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)
```

```
39 vs=V*sqrt(3)*1e-3;
40 mprintf("\n\n\n(b)\nSending End Voltage= %.2f kV",
      vs)
41
42 Is=(C*Vr)+(D*Ir);
43 I=abs(Is)
44 mprintf("\nSending End Current= %.1f A", I)
45
46 phi1=atan(imag(Vs)/real(Vs))*(180/%pi);
47 phi2=atan(imag(Is)/real(Is))*(180/%pi);
48 phi=phi1-phi2;
49 pfs=cosd(phi);
50 mprintf("\nSending End pf= %.2f ",pfs)
```

Scilab code Exa 3.19 ferrenti effect

ferrenti effect

```
1 clear;
2 clc;
3
4 Vr=220e3/sqrt(3);
5 d=300;
6 f=50;
7
8 V=-1*((Vr*(2*%pi*f)^2 * d*d *1e-10)/18);
9 Vs=Vr+V;
10 vs=sqrt(3) * Vs /1000;
11 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

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

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

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);
```

```
12 V2=220e3/sqrt(3);
13
14 R=15
15 X=50;
16
17 tr=sqrt(1-(((R*P)+(X*Q))/(V1^2)));
18 mprintf("tap setting  tr= %.4f, ts=%.3f",fix(1e4/tr)
     /1e4,tr)
```

Scilab code Exa 3.23 find tap setting under given conditions

find tap setting under given conditions

```
1  clear;
2  clc;
3
4  Vb=132
5  Sb=100
6  X=.15
7
8  v1=125
9  V1=v1/Vb;
10 Q1=50;
11 Qpu1=Q1/Sb;
12
13 Vn1=(V1 + sqrt(V1^2 - (4*Qpu1*X)))/(2*1);
14 vn1=Vn1 * Vb;
15 Vo1=33;
16 t1=vn1/Vo1;
17
18 v2=140
19 V2=v2/Vb;
20 Q2=20;
21 Qpu2=Q2/Sb;
22
```

```
23 Vn2=(V2 + sqrt(V2^2 - (4*Qpu2*X)))/(2*1);
24 vn2=Vn2 * Vb;
25 Vo2=33;
26 t2=vn2/Vo2;
27
28 tm=(t1+t2)/2;
29 dt=tm-t1;
30 ts=dt*100/tm
31
32 mprintf("tap setting = +- %.0f percent", ts);
```

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

```
find capacity of phase modifier at different loads
```

```
1  clear;
2  clc;
3
4  Vr1=132
5  Vs1=140
6  VA=40;
7  pf=.8;
8  Pr=VA*pf;
9  pfa=-1* acos(pf);
10 Qr=(VA * sin(pfa));
11
12 A=complex(.98*cosd(3), .98*sind(3));
13 B=complex(110*cosd(75), 110*sind(75));
14
15 bd1=acos(( Pr+real(((Vr1)^2 * conj(A/B))) ) * (abs(B
      )/(Vs1 *Vr1)));
16 Qr1= (((Vs1 *Vr1)/abs(B))*sin(bd1))-imag(((Vr1)^2 *
      conj(A/B)));
17
18 Q1=-Qr-Qr1;
```

```
19 mprintf("\n\n\n(a)\nCapacity of static capacitor= %
      .2f MVar leading", fix(Q1*100)/100)
20
21 Prn=0;
22 Qrn=0;
23
24 bd2=acos((Prn+real(((Vr1)^2 * conj(A/B))) ) * (abs(B
      )/(Vs1 *Vr1)));
25 Qr2= (((Vs1 *Vr1)/abs(B))*sin(bd2))-imag(((Vr1)^2 *
      conj(A/B)));
26
27 Q2=-Qrn-Qr2;
28 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

power transfer and SPM rating to improve pf

```
1  clear;
2  clc;
3
4  Vr=220
5  Vs=240
6
7  A=complex(.9*cosd(1), .9*sind(1));
8  B=complex(140*cosd(84), 140*sind(84));
9
10 b=atan(imag(B)/real(B));
11 d=b
12 vs=complex (Vs *cos(d), Vs *sin(d));
13 P= (((Vs *Vr)/abs(B))*cos(b-d))- real(((Vr)^2 * conj
      (A/B)));
14 mprintf("(a) Max Power Transmitted = %.2f MW", P);
15
```

```
16 Pr=80;
17 Sr=100;
18 Qr=sqrt(Sr^2 - Pr^2);
19 bd1=round(acos(( Pr+real(((Vr)^2 * conj(A/B))) ) * (
      abs(B)/(Vs *Vr)))*1000)/1000;
20 Qr1= (((Vs *Vr)/abs(B))*sin(bd1))-imag(((Vr)^2 *
      conj(A/B)));
21
22 Q=+Qr-Qr1;
23 mprintf("\n\n\n(b)\nCapacity of static capacitor= %
      .2f MVar leading", Q)
24 disp("There is a calculation error in the textbook.
      40-49.37=10.63")
25
26 d1=b-bd1;
27 d1=d1*180/%pi
28 mprintf("\n\n\n(c)\nLoad Angle factor= %.2f deg", d1
      )
```

Scilab code Exa 3.29 overall ABCD parameters

```
overall ABCD parameters
```

```
1 clear;
2 clc;
3
4 A=complex(.93*cosd(3), .93*sind(3));
5 B=complex(150*cosd(70), 150*sind(70));
6 D=A;
7
8 C=((A*D)-1)/B
9
10 Z=complex(100*cosd(70), 100*sind(70));
11 Y=complex(.00025*cosd(-75), .00025*sind(-75));
12
```

```
13 m1=[A B; C D]
14 m2=[1 0 ; Y 1 ]
15 m3=[1 Z ; 0 1 ]
16
17 A0=m1*m2*m3;
18 A1=A0(1,1)
19 B1=A0(1,2)
20 C1=A0(2,1)
21 D1=A0(2,2)
22
23 mprintf("\nA0= %s", string(round(abs(A1)*1000)/1000)
       +'/_'+ string(round(atand(imag(A1)/real(A1))*10)
      /10) )
24 mprintf("\nB0= %s", string(round(abs(B1)*1000)/1000)
       +'/_'+ string(round(atand(imag(B1)/real(B1))
      *100)/100) )
25 mprintf("\nC0= %s", string(round(abs(C1)*100000)
      /100000) +'/_'+ string(round(atand(imag(C1)/real(
      C1))*1)/1 ) )
26 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

find wavelength and velocity of propagation

```
1 clear;
2 clc;
3
4 f=50;
5 l=200;
6 Z=complex(14.1, 51.48);
7 Y=complex(0, 1.194e-3);
8 g=sqrt(Z*Y);
```

```
9
10 b=imag(g)/l;
11 wl=2*%pi/b;
12 v=f*wl
13 mprintf("\nwavelength = %.4f*1e3 km",wl*1e-3);
14 mprintf("\nVelocity of Propagation = %.2f*1e5 km/sec
     ",v*1e-5);
```

Scilab code Exa 3.31 sending end parameters using pu

```
sending end parameters using pu
```

```
1  clear;
2  clc;
3
4  clear;
5  clc;
6
7
8  vr=220e3/sqrt(3);
9  Vb=vr
10 Vr=vr/Vb
11 Sr=40e6;
12 Sb=40e6;
13 Ib=Sb/(3*Vb)
14 Zb=Vb/Ib;
15
16 pf=.9;
17 pfa=-1*acos(pf);
18 Irm=(Sr/(3*vr))/Ib;
19 Ir=complex(Irm *pf, Irm * sin(pfa));
20
21
22 d=500;
23 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

```
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

```
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);
```

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

Scilab code Exa 3.34 find pu values of system

find pu values of system

```
1  clear;
2  clc;
3
4  sg1=10
5  sg2=20
6  st1=10
7  st2=10*3;
8
9  vg1=6.6;
10 vg2=11.5;
11 vt1a=6.6
12 vt1b=115
13 vt2a=75*sqrt(3);
14 vt2b=7.5*sqrt(3);
15 nt1=vt1b/vt1a;
16 nt2=vt2b/vt2a;
17 xg1=.1
18 xg2=.1
19 xt1=.15
20 xt2=.1
21
22
23 Sb=20;
24 Vb1=6.6;
25 Vb2=Vb1*nt1;
```

```
26 Vb3=Vb2*nt2;
27
28 Xg1=xg1*Sb/sg1;
29 Xg2=xg2*Sb/sg2;
30 Xt1=xt1 * Sb/st1;
31 Xt2= xt2 * Sb/st2 * (vt2a/Vb2)^2
32
33  mprintf ("\nReactance of Generator 1= %.1f",Xg1);
34  mprintf ("\nReactance of Generator 2= %.1f",Xg2);
35  mprintf ("\nReactance of Transformer 1= %.1f ",Xt1)
       ;
36  mprintf ("\nReactance of Transformer 2= %.5f ",fix(
       Xt2*1e5)/1e5);
37
38
39 Zbl=Vb2^2/Sb;
40
41 xab=100;
42 xad=100
43 xbc=150;
44 xbd=200;
45 xdc=50;
46
47 Xab=xab/Zbl;
48 Xad=xad/Zbl;
49 Xbc=xbc/Zbl;
50 Xbd=xbd/Zbl;
51 Xdc=xdc/Zbl;
52
53 printf("\nXab=%.4f\t", Xab);
54 printf("Xad=%.4f\t", Xad);
55 printf("Xbc=%.4f\t", Xbc);
56 printf("Xbd=%.4f\t", Xbd);
57 printf("Xdc=%.4f\t", Xdc);
```

Scilab code Exa 3.35 find pu values of system

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;
```

```
35 Vb4=Vb2*nt3;
36
37 Xg1=xg1*Sb/sg1;
38 Xg2=xg2*Sb/sg2;
39 Xg3=xg3*Sb/sg3* (vg3/Vb4)^2;
40 Xt1=xt1 * Sb/st1;
41 Xt2=xt2 * Sb/st2;
42 Xt3= xt2 * Sb/st3 * (vt3b/Vb4)^2
43
44  mprintf ("\nReactance of Generator 1= %.2f pu",Xg1)
      ;
45  mprintf ("\nReactance of Generator 2= %.1f pu",Xg2)
      ;
46  mprintf ("\nReactance of Generator 3= %.4f pu",Xg3)
      ;
47  mprintf ("\nReactance of Transformer 1= %.1f pu",
      Xt1);
48  mprintf ("\nReactance of Transformer 2= %.1f pu",
      Xt2);
49  mprintf ("\nReactance of Transformer 3= %.3f pu",
      Xt3);
50
51
52 Zbl=Vb2^2/Sb;
53
54 xl1=120;
55 xl2=90
56
57 Xl1=xl1/Zbl;
58 Xl2=xl2/Zbl;
59
60  mprintf ("\nReactance of Line 1 = %.4f pu",Xl1);
61  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

```
calculate actual values of generator current line current load current load voltag
```

```
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

```
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)
```

```
11 I=Is *(exp(%i * -pfa))
12
13 I1=I * z2/(z1+z2)
14 pfa1=atan(imag(I1)/real(I1))
15 pf1=cos(pfa1)
16 mprintf("\n(a) Current in OH line = %.1f A pf= %.3f"
     , abs(I1), pf1)
17
18 I2=I * z1/(z1+z2)
19 pfa2=atan(imag(I2)/real(I2))
20 pf2=cos(pfa2)
21 mprintf("\n(b) Current in cable = %.2f A pf= %.2f",
     abs(I2), pf2)
22
23 vr=sqrt((Vs)^2-imag(I1*z1)^2)- real(I1*z1)
24 Vr=vr*sqrt(3)/1000;
25 mprintf("\n(c) Receiving end voltage = %.3f KV", Vr)
26
27 d=atan(imag(I1*z1)/(Vr+real(I1*z1)))
28 phi=pfa-d;
29 pfr=cos(phi)
30 mprintf("\n(d) Receiving end pf = %.1f lagging", pfr
     )
```

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

```
find receiving end voltage and efficiency of transmission
```

```
1 clear;
2 clc;
3
4 l=300
5 R=.4 *3
6 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

find voltage across string and string efficiency

```
1  clear;
2  clc;
3
4  C=1;
5  C1=0.1;
6  V=66;
7  n=4;
8
9  v1= 1;
10 v2= (C+C1) * v1;
11 v3= (C * v2) + (C1*(v1+v2));
12 v4= (C * v3) + (C1*(v1+v2+v3));
13
14 V1= V/(sqrt(3)*(v1+v2+v3+v4));
15 V2= v2* V1;
16 V3= v3* V1;
17 V4= v4* V1;
18 mprintf("\n V1= %.2f kV",V1);
19 mprintf("\n V2= %.2f kV",V2);
```

```
20 mprintf("\n V3= %.2f kV",V3);
21 mprintf("\n V4= %.2f kV",V4);
22
23 eff= (V1+V2+V3+V4)/(n*V4) *100;
24 mprintf("\n string efficiency= %.1f percent",eff);
```

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

calculate string efficiency with presesnce of guard ring

```
1  clear
2  clc
3
4  c1=.15
5  c2=.05
6  V=100
7
8  A=[
9  1+c1 -(1+c2) -c2
10 c1 1+c1 -(1+c2)
11 1 0 0
12 ]
13
14 B=[0 0 1]'
15 Vm=inv(A) * B
16 Vm = round(Vm*1e4)/1e4
17 V1=V/(Vm(1)+Vm(2)+Vm(3))
18 V2=Vm(2) * V1
19 V3=Vm(3) * V1
20
21 ef=V/(3*V3)
22
23 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

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

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

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);
```

```
10 u=v(2,1);
11 n=(t+u+1)/(3*u);
12 mprintf("the string efficiency is =%.2f",n);
```

Scilab code Exa 4.6 voltage across various discs in insulator

```
voltage across various discs in insulator
```

```
1 clear;
2 clc;
3 k=.1;
4 n=4;
5 for i=1:4
6     z(i)=2*cosh((i-.5)*sqrt(k))*sinh(.5*sqrt(k))/
        sinh(n*sqrt(k));
7 end
8 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

```
line to oin capacitances so that voltage distribution is uniform
```

```
1 clear;
2 clc;
3 c=1;
4 w=1;
5 v=1;
6 c1=c*w*v/(4*w*v);
7 c2=2*w*v/(3*w*v);
8 c3=3*w*v/(2*w*v);
```

```
9  c4=4*w*v/(1*w*v);
10 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

`find mutual capacitances of insulator discs`

```
1  clear;
2  clc;
3  w=1;
4  c=1;
5  v=1;
6  y=5;
7  c2=w*c*v+(w*y*v);
8  c3=2*w*c+6*w*c;
9  c4=3*w*c+8*w*c;
10 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

`find ratio of capacitances of insulator to earth capacitance of insulator`

```
1  clear;
2  clc;
3  v=1;
4  v1=.4*v;
5  v2=.6*v;
6  k=(v2-v1)/v1;
7  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

finding sag in different weather conditions

```
1  clear;
2  clc;
3
4  m=.847;
5  g=9.81;
6  dia=1.95e-2;
7  l=244;
8  T=3.56e4;
9  Th_ice=.96e-2;
10 F_wind= 382;
11 W_ice=8920;
12 h=7.62;
13 L=1.43;
14
15 //(a)
16 w=m*g;
17 S= (w*l*l)/(8*T);
```

```
18 mprintf("\n (a)Sag= %.2f m", S);
19
20 //(b)
21 D=dia+Th_ice+Th_ice;
22 Fw=F_wind * D;
23 Wice = W_ice * (%pi/4) * ((D*D)-(dia*dia));
24 F=((w+Wice)^2 + Fw^2)^.5;
25 s=(F*l*l)/(8*T);
26 a=atan(Fw/(w+Wice));
27 S2=s * cos (a);
28 mprintf("\n (b)Vertical Sag= %.2f m", S2);
29
30 //(c)
31 H=h+L+S2;
32 mprintf("\n (c)Height of lowest cross arm= %.2f m",
     H);
```

Scilab code Exa 5.2 clearance of line

clearance of line

```
1 clear;
2 clc;
3
4 l=336;
5 h1=33.6;
6 h2=29;
7 w=8.33;
8 T=3.34e4;
9
10 //(a)
11 lc=l+ (2*T * (h1-h2)/(w*l));
12 S=w*lc*lc/(8*T);
13 cl= h1-S;
14 mprintf("\n (a)Clearance= %.3f m", cl);
```

```
15
16 //(b)
17 d1=lc/2;
18 d2=l-d1;
19 mprintf("\n (b)Distance of point O from lower
      support = %.2f m", d2);
```

Scilab code Exa 5.3 height of mid point from ground

height of mid point from ground

```
1  clear;
2  clc;
3  l=300;
4  h1=80;
5  h2=50;
6  w=8.28;
7  T=19620;
8
9  //(a)
10 lc=l+ (2*T * (h1-h2)/(w*l));
11 dOC=(lc/2)-l;
12 hOC=w*dOC*dOC/(2*T);
13 dOP=dOC + (l/2);
14 hOP=w*dOP*dOP/(2*T);
15 hPC=hOP-hOC;
16 hP=hPC+ h2;
17 mprintf("\nHeight of mid point P above C =%.3f m",
      hPC);
18 mprintf("\nHeight of mid point P above water level =
      %.3f m",hP);
```

Scilab code Exa 5.4 finding sag

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

finding minimum clearance and position of clearance point

```
clear;
clc;

h1=55;
h2=50;
```

```
6  l=300;
7  g=9.81;
8  T=2000*g;
9  m=.85;
10 w=m*g;
11
12 //(a)
13 lc=l+ (2*T * (h1-h2)/(w*l));
14 S= w * lc *lc /(8*T);
15 cl=h1-S;
16 mprintf("\n (a)Minimum Clearance between conductor
      and water= %.2f m", cl);
17
18 //(b)
19 dOB=lc/2;
20 dOA=l-dOB;
21 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

find sag and tension under erection conditions

```
1  clear
2  clc
3
4  safety=2
5  d=1.95e-2
6  A=2.25e-4
7  E=91.4 *1e9
8  alpha=18.44 *1e-6
9  Temp21=10
10 Temp22=40
11 Tmax=77900
12 w=8.31
```

```
13 span=250
14
15
16 Fw=378 * d
17 Fw=round(Fw*100)/100
18 Ft1=sqrt(w^2 + Fw^2)
19 T1=Tmax/safety
20 Ft2=w
21
22
23 c_1=1
24 c_2=T1 -(alpha * A * E * (Temp22-Temp21)) - A*E*Ft1
     ^2 * span^2 /(24*T1^2)
25 c_3=0
26 c_4=A*E*Ft2^2 * span^2 /24
27 pol=poly([-c_4 -c_3 -c_2 c_1], "xx", "c")
28 T2s=roots(pol)
29
30 T2=T2s(1)
31 T2=round(T2)
32 Sag1= w * span *span / (8 * T2)
33
34 //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
35 mprintf("sag at erection= %.2f m", Sag1)
36
37 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

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

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

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

```
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

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)
```

```
18
19 Vvl=(3e6/sqrt(2))*r*del*mvl* log(d/r)* (1+(.03/sqrt(
      del*r))) *1e-3;
20 mprintf("\nVisual local voltage = %.1f KV/phase",
      Vvl)
21
22 Vvg=(3e6/sqrt(2))*r*del*mvg* log(d/r)* (1+(.03/sqrt(
      del*r))) *1e-3;
23 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

Finding total loss in fair weather and bad weather using peeks formula

```
1  clear;
2  clc;
3
4  dia=22.26e-3;
5  r=dia/2;
6  V=220;
7  d=6;
8  temp=25;
9  P=73;
10 m0=.84;
11 f=50;
12 l=250;
13
14 V=round(V/sqrt(3));
15 del=round((3.86*P/(273+temp))*10000)/10000;
16 Vd=round((3e6/sqrt(2))*r*del*m0* log(d/r) *1e-3);
17
18 //(a)Good Weather
```

```
19 Pc=243.5 * ((f+25)/del) * sqrt(r/d)* (V-Vd)^2 * 1e
   -5;
20 PC= Pc * l;
21 Ptot= 3 *PC;
22 mprintf("\ntotal loss in good weather = %.2f kW",
   Ptot);
23
24 //(b)Bad Weather
25 Vd1=.8*Vd;
26 Pc1=243.5 * ((f+25)/del) * sqrt(r/d)* (V-Vd1)^2 * 1e
   -5;
27 PC1= Pc1 * l;
28 Ptot1= 3 *PC1;
29 mprintf("\ntotal loss in bad weather = %.0f kW",
   Ptot1);
```

Scilab code Exa 6.3 finding visual corona voltage

finding visual corona voltage

```
1 clear;
2 clc;
3
4 dia=1.04e-2;
5 r=dia/2;
6 m=.85;
7 d=2.44;
8 P=74;
9 temp=21;
10
11 del=round((3.86*P/(273+temp))*1000)/1000;
12
13 Vv=(3e6/sqrt(2))*r*del*m* log(d/r)* (1+(.03/sqrt(del
   *r))) *1e-3;
14 mprintf("\nVisual local voltage = %.2f KV/phase", Vv
   )
```

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

```
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

```
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);
```

```
17 Ib=300*exp(%i *-2* %pi/3);
18 Ic=300*exp(%i *2* %pi/3);
19
20 si=2e-7* ((Ia * log(DaQ/DaP))+(Ib * log(DbQ/DbP))+(
      Ic * log(DcQ/DcP)))
21
22 V=2*%pi * 50* abs(si);
23
24 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

```
finding magnitude of voltage induced in telephone line due to EMI of power line ur
```

```
1 clear;
2 clc;
3
4 D1=1.2
5 D2=.6;
6 h=1;
7 H=10
8
9 DaP=sqrt((D1-(D2/2))^2+h^2)
10 DaQ=sqrt((D1+(D2/2))^2+h^2)
11
12 Ia=2000*exp(%i * 0);
13 si=2e-6* ((Ia * log(DaQ/DaP)))
14
15 V=2*%pi * 50* abs(si);
16
17 mprintf("voltage induced in telephone line =%.3f V/
      m", V)
```

Scilab code Exa 7.3 potential of conductor due electrostatic effect

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

```
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

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

inductance of a 3 core belted cable

```
1  clear;
2  clc;
3
4  n=37;
5  r=.238
6  t=.5;
7
8  r1=r*3.5;
9  GMR=.7788*r1;
10 D= 2*(r1+t);
11 L=.4605 * log10(D/GMR);
12 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

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

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;
```

```
12 t=R-r;
13
14 mprintf("\nMost economical conductor diameter = %.2f
       cm", fix(2*R*100)/100);
15 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

`find postitions of intersheaths max min stress and voltage on intersheaths`

```
1  clear;
2  clc;
3
4  V=66;
5  V=V/sqrt(3);
6  T=60;
7  d=2;
8  r=d/2;
9  D=5.3;
10 R=D/2;
11
12 V=fix(V*sqrt(2) *10)/10;
13
14 a=(R/r)^(1/3);
15 a=round(a*1e3)/1e3;
16 r1=a*r;
17 r2=round(a*r1*1000)/1000;
18 V1= V *((1/r)-(1/r2)) *r ;
19 V2= (V-( V1* ((1/r)-(1/r1))) )*r1;
20 //An error exists in the text book while calculating
       V1 and V2 and hence Emax and Emin
21
22 Emax= (V-V1)/(r * log (a));
```

```
23 Emin= (V-V1)/(r1 * log (a));
24 mprintf("\n When intersheaths are used:\n Emax= %.2f
       kV/cm \t Emin= %2f kV/cm",Emax,Emin);
25 mprintf("\n Peak voltages at intersheaths:\n V1= %.2
      f kV \t V2= %.2f kV\n",V1,V2);
26 disp("An error exists in the text book while
      calculating V1 and V2 and hence Emax and Emin")
27
28 Emax= (V)/(r * 3*log (a));
29 Emin= (V)/(R * 3*log (a));
30 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

radius voltage of intersheath and ratio of maximum stress with and wothout intersh

```
1 clear;
2 clc;
3
4 //solving for prt (c) only
5
6 R=3;
7 r=1;
8 V=60;
9
10
11 a=sqrt(R/r);
12 r1=a*r;
13
14 mprintf("\n (c) r1= %.3f cm", r1);
15
16 V1=V*((a)/(1+a));
```

```
17 mprintf("\n      V1 voltage at intersheath= %.2f cm",
       V1);
18
19 b=2/(1+a);
20 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

`find maximum voltage in a cable having 2 insulation materials`

```
1  clear;
2  clc;
3
4  r=0.5;
5  R=2.5;
6  Vga=60;
7  ea=4;
8  eb=2.5;
9  Vgb=50;
10
11 v=Vga*ea*r;
12
13 r1=v/(eb*Vgb);
14
15 V=v *((log(r1/r)/ea)+(log(R/r1)/eb));
16
17 mprintf("Maximum working voltage = %.2f kV", V);
```

Scilab code Exa 8.7 parameters of underground feeder

`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

```
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

```
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);
```

```
40
41 sheathlosses=.1;
42
43 I=((T2-T1)/(n* Rac * (Gi + ((1+sheathlosses)*(Gp+Gs)
   ))))^.5
44 mprintf("Current Rating =%d Amperes",I);
```

Chapter 9

Load Flow Studies

Scilab code Exa 9.1 form Y bus

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
```

```
20          Y(n1,n2)= Y(n1,n2)-ynew
21          Y(n2,n1)= Y(n2,n1)-ynew
22          Y(n2,n2)= Y(n2,n2)+ynew
23      end
24  end
25  disp(Y)
```

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

form Y bus and effect of adding a line

```
1  clear
2  clc
3
4  data=[ 1 2 .025 .1
5   2 3 .02 .08
6    3 4 .05 .2
7     1 4 .04 .16
8  ]
9
10 n=4
11 e=4
12 Y=zeros(n,n)
13
14 for i=1:e
15      ynew = 1/(data(i,3)+(%i *real(data(i,4))))
16      n1=real(data(i,1))
17      n2=real(data(i,2))
18      if(n1==0)
19          Y(n2,n2)=Y(n2,n2)+ynew
20      else
21          Y(n1,n1)= Y(n1,n1)+ynew
22          Y(n1,n2)= Y(n1,n2)-ynew
23          Y(n2,n1)= Y(n2,n1)-ynew
24          Y(n2,n2)= Y(n2,n2)+ynew
```

```
25      end
26  end
27
28  disp(round(Y*1e2)/100,"(a)")
29
30  data(e+1,:)=[1  3  .1  .4]
31  i=e+1
32  ynew  =  1/(data(i,3)+(%i  *real(data(i,4))))
33
34  n1=real(data(i,1))
35  n2=real(data(i,2))
36  if(n1==0)
37      Y(n2,n2)=Y(n2,n2)+ynew
38  else
39      Y(n1,n1)= Y(n1,n1)+ynew
40      Y(n1,n2)= Y(n1,n2)-ynew
41      Y(n2,n1)= Y(n2,n1)-ynew
42      Y(n2,n2)= Y(n2,n2)+ynew
43  end
44
45
46  disp(round(Y*1e2)/100,"(b)")
```

Scilab code Exa 9.4 find y bus

find y bus

```
1  clear
2  clc
3
4  data=[  1  2  .2+%i*.8  %i*.02
5  2  3  .3+%i*.9  %i*.03
6  2  4  .25+%i*1  %i*.04
7  3  4  .2+%i*.8  %i*.02
8  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

```
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
```

```
8  %inf                  %inf          %inf
       %inf
9  ]
10 disp("inf shows that this value is to be found ")
11 disp(Y,"given")
12
13 Y(1,4)=round(Y(1,1)+Y(1,3)+Y(1,2))
14 Y(4,4)=0-Y(1,4)-Y(2,4)-Y(3,4)
15 Y(4,1)=Y(1,4)
16 Y(2,1)=Y(1,2)
17 Y(3,2)=Y(2,3)
18 Y(3,1)=Y(1,3)
19 Y(4,2)=Y(2,4)
20 Y(4,3)=Y(3,4)
21 Y(2,2)=0-Y(2,1)-Y(2,4)-Y(2,3)
22 Y(3,3)=0-Y(3,1)-Y(3,4)-Y(3,2)
23
24
25
26 disp(Y,"completed")
```

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

find y bus with mutual coupling of lines present

```
1  clear
2  clc
3
4  z=[
5  %i*1     0    0    0    0
6  0        %i*.4    %i*.2  0  0
7  0        %i*.2    %i*.5  0  0
8     0    0    0  %i*.2     0
9  0    0    0    0  %i*.25
10 ]
```

```
11
12 y=inv(z)
13
14 A=[
15 1  0  0  0
16 -1  1  0  0
17 -1  0  1  0
18 0  0  1  -1
19 0  -1  0  -1
20 ]
21
22 Y=A'*y*A
23
24 disp(Y)
```

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

```
find reactive power generations losses and powers transferred
```

```
1  clear
2  clc
3
4  G=[
5  %inf  %inf  1.5  1
6  0     %inf  2.0  .5
7  3.5   %inf  0    0
8  0     %inf  2    .8
9  ]
10
11
12 ld=[
13 1  2  %i*.2
14 2  3  %i*.1
15 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
```

```
53 d(2:4)=inv(A) * [G(2,5) G(3,5) G(4,5)]'
54 d=round((d)*1000)/1000
55 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))));
56 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))));
57 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))));
58 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))));
59
60 G(:,6)=round(G(:,6) *10000)/10000
61 G(:,2)=G(:,6) + G(:,4)
62 G(:,2)=round(G(:,2)*1e4)/1e4
63 mprintf("\n(a) Q1= %.4f, Q2= %.4f, Q3= %.3f, Q4= %.4
      f, ", G(1,2),G(2,2),G(3,2),G(4,2))
64
65 Ql=G(1,6)+G(2,6)+G(3,6)+G(4,6)
66 mprintf("\n(b) reactive line losses=%.4fpu",Ql)
67
68 X=ld(:,3)/%i
69
70 P(1)=round((sin(d(1)-d(2))/X(1))*1000)/1000
71 P(2)=round((sin(d(2)-d(3))/X(2))*1000)/1000
72 P(3)=round((sin(d(3)-d(4))/X(3))*1000)/1000
73 P(4)=round((sin(d(1)-d(4))/X(4))*1000)/1000
74 P(5)=round((sin(d(1)-d(3))/X(5))*1000)/1000
75
76 Q(1)=round(((1-cos(d(1)-d(2)))/X(1))*10000)/10000
77 Q(2)=round(((1-cos(d(2)-d(3)))/X(2))*10000)/10000
78 Q(3)=round(((1-cos(d(3)-d(4)))/X(3))*10000)/10000
79 Q(4)=round(((1-cos(d(1)-d(4)))/X(4))*10000)/10000
80 Q(5)=round(((1-cos(d(1)-d(3)))/X(5))*10000)/10000
81
```

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

Scilab code Exa 9.9 solve using gauss seidel for 1 variable

```
solve using gauss seidel for 1 variable
```

```
1  clear
2  clc
3
4  e=1e-5
5  x=1
6  E=100
7  while (E>e)
8      x1=.5*(7+log10(x))
9      E=abs(x-x1)
10     x=x1
11 end
12
13 mprintf("x = %f",x)
```

Scilab code Exa 9.10 solve using gauss seidel for 2 variables

```
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

```
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
```

```
19      n1=real(y(i,1))
20      n2=real(y(i,2))
21      ynew=y(i,3)
22      if(n1==0)
23          Y(n2,n2)=Y(n2,n2)+ynew
24      else
25          Y(n1,n1)= Y(n1,n1)+ynew
26          Y(n1,n2)= Y(n1,n2)-ynew
27          Y(n2,n1)= Y(n2,n1)-ynew
28          Y(n2,n2)= Y(n2,n2)+ynew
29      end
30  end
31  // bus no | PL | QL | PG | QG | V | th | btype |
       Qmin | Qmax
32
33  data = [
34  1 1 .5 %inf %inf 1.02 0 1 %inf %inf
35  2 0 0 2 %inf 1.02 0 2 .2 .6
36  3 .5 .2 0 0 1 0   3 %inf %inf
37  4 .5 .2 0 0 1 0   3 %inf %inf
38  5 .5 .2 0 0 1 0   3 %inf %inf
39
40  ]
41  disp(Y, "(a)")
42  j = data(:,1);     // Bus number.
43  PL = data(:,2);
44  QL = data(:,3);
45  PG = data(:,4);
46  QG = data(:,5);
47  V = data(:,6);
48  th = data(:,7);
49  btype = data(:,8);    // Type of Bus 1-Slack, 2-PV,
       3-PQ.
50  Qmin = data(:,9);
51  Qmax = data(:,10);
52  n = max(j);
53  P = PG - PL;
54  Q = QG - QL
```

```
55
56 for i=1:n
57     V(i,1)=V(i,1) * exp (%i * th(i,1)* %pi / 180)
58 end
59
60 Vprev = V;
61 toler = 1000;
62 iteration = 1;
63 while (iteration==1)
64     for i = 2:n
65         summ = 0;
66         for k = 1:n
67             if i ~= k
68                 summ = summ + Y(i,k)* V(k);
69             end
70         end
71         if btype(i) == 2
72             Q(i) = -imag(conj(V(i))*(summ + Y(i,i)*V
                   (i)));
73             if (Q(i) > Qmax(i)) | (Q(i) < Qmin(i))
74                 if Q(i) < Qmin(i)
75                     Q(i) = Qmin(i);
76                 else
77                     Q(i) = Qmax(i);
78                 end
79                 btype(i) = 3;
80             end
81         end
82         V(i) = (1/Y(i,i))*(((P(i)- %i*Q(i))/conj(V(i
               ))) - summ);
83         if btype(i) == 2
84             V(i) = abs(Vprev(i))*exp(%i * atan(imag(
                   V(i))/real(V(i))));
85         end
86     end
87     iteration = iteration + 1;
88     toler = max(abs(abs(V) - abs(Vprev)));
89     Vprev = V;
```

```
90 end
91 disp("(b)")
92 mprintf("\nV3 = %.2f ang(%.2f) deg", abs(V(3)),atand
     (imag(V(3))/real(V(3))))
93 mprintf("\nV4 = %.3f ang(%.2f) deg", abs(V(4)),atand
     (imag(V(4))/real(V(4))))
94 mprintf("\nV5 = %.4f ang(%.2f) deg", abs(V(5)),atand
     (imag(V(5))/real(V(5))))
95 mprintf("\ndelta 2 = %.2f deg",atand(imag(V(2,1))/
     real(V(2,1))))
96 mprintf("\nQ2 = %.4f ",Q(2,1))
```

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

```
find bus voltage and load angle using GS minimum value of Q2 given
```

```
1  clear
2  clc
3
4  ey=1/(.05 + %i*.15)
5  y=[
6  1 2 ey
7  1 5 ey
8  2 5 ey
9  2 3 ey
10 3 4 ey
11 4 5 ey
12 ]
13 n=5
14 e=6
15
16 Y=zeros(n,n)
17
18 for i=1:e
```

```
19      n1=real(y(i,1))
20      n2=real(y(i,2))
21      ynew=y(i,3)
22      if(n1==0)
23          Y(n2,n2)=Y(n2,n2)+ynew
24      else
25          Y(n1,n1)= Y(n1,n1)+ynew
26          Y(n1,n2)= Y(n1,n2)-ynew
27          Y(n2,n1)= Y(n2,n1)-ynew
28          Y(n2,n2)= Y(n2,n2)+ynew
29      end
30  end
31  // bus no | PL | QL | PG | QG | V | th | btype |
       Qmin | Qmax
32
33  data = [
34  1 1 .5 %inf %inf 1.02 0 1 %inf %inf
35  2 0 0 2 %inf 1.02 0 2 .3 %inf
36  3 .5 .2 0 0 1 0   3 %inf %inf
37  4 .5 .2 0 0 1 0   3 %inf %inf
38  5 .5 .2 0 0 1 0   3 %inf %inf
39
40  ]
41
42  j = data(:,1);     // Bus number.
43  PL = data(:,2);
44  QL = data(:,3);
45  PG = data(:,4);
46  QG = data(:,5);
47  V = data(:,6);
48  th = data(:,7);
49  btype = data(:,8);    // Type of Bus 1-Slack, 2-PV,
       3-PQ.
50  Qmin = data(:,9);
51  Qmax = data(:,10);
52  n = max(j);
53  P = PG - PL;
54  Q = QG - QL
```

```
55
56 for i=1:n
57     V(i,1)=V(i,1) * exp (%i * th(i,1)* %pi / 180)
58 end
59
60 Vprev = V;
61 toler = 1000;
62 iteration = 1;
63 while (iteration==1)
64     for i = 2:n
65         summ = 0;
66         for k = 1:n
67             if i ~= k
68                 summ = summ + Y(i,k)* V(k);
69             end
70         end
71         if btype(i) == 2
72             Q(i) = -imag(conj(V(i))*(summ + Y(i,i)*V
                   (i)));
73             if (Q(i) > Qmax(i)) | (Q(i) < Qmin(i))
74                 if Q(i) < Qmin(i)
75                     Q(i) = Qmin(i);
76                 else
77                     Q(i) = Qmax(i);
78                 end
79                 btype(i) = 3;
80             end
81         end
82         V(i) = (1/Y(i,i))*(((P(i)- %i*Q(i))/conj(V(i
               ))) - summ);
83         if btype(i) == 2
84             V(i) = abs(Vprev(i))*exp(%i * atan(imag(
                   V(i))/real(V(i))));
85         end
86     end
87     iteration = iteration + 1;
88     toler = max(abs(abs(V) - abs(Vprev)));
89     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)))*1
       e3)/1e3
end
disp(ansmat," bus no   |V|       delta")
```

Scilab code Exa 9.13 solve using newton raphson 1 variable

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

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

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

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
```

```
23          Y(n1,n1)= Y(n1,n1)+ynew+y0
24          Y(n1,n2)= Y(n1,n2)-ynew
25          Y(n2,n1)= Y(n2,n1)-ynew
26          Y(n2,n2)= Y(n2,n2)+ynew+y0
27      end
28  end
29  for i=1:n
30      for j=1:n
31          if i==j then
32              Yb(i,j)= string(round(abs(Y(i,j))*1000)
                   /1000) +'/_'+ string(round(atand(imag
                   (Y(i,j))/real(Y(i,j)))*100)/100)
33          else
34              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)
35          end
36      end
37  end
38  Y=round(Y*1e3)/1e3
39  disp(Yb,"(a)Ybus")
40  // bus no | PL | QL | PG | QG | V | th | btype
41
42  data = [
43  1 %inf %inf 1 .5  1.03 0 1
44  2 1.5 %inf  0 0 1.03 0 2
45  3 0 0  1.2 .5  1 0   3
46  ]
47
48  j = data(:,1);     // Bus number.
49  PG = data(:,2);
50  QG = data(:,3);
51  PL = data(:,4);
52  QL = data(:,5);
53  V = data(:,6);
54  th = data(:,7)* %pi / 180;
55  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
```

```
91          if btype(i) == 3
92              dQ(k,1) = dQa(i);
93              k = k+1;
94          end
95      end
96      dP = dPa(2:nbus);
97      M = [dP; dQ];
98
99      H = zeros(nbus-1,nbus-1);
100     for i = 1:(nbus-1)
101         m = i+1;
102         for k = 1:(nbus-1)
103             n = k+1;
104             if n == m
105                 for n = 1:nbus
106                     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)));
107                 end
108                 H(i,k) = H(i,k) - V(m)^2*B(m,m);
109             else
110                 H(i,k) = V(m)* V(n)*(G(m,n)*sin(th(m
                        )-th(n)) - B(m,n)*cos(th(m)-th(n)
                        ));
111             end
112         end
113     end
114
115     N = zeros(nbus-1,npq);
116     for i = 1:(nbus-1)
117         m = i+1;
118         for k = 1:npq
119             n = pq(k);
120             if n == m
121                 for n = 1:nbus
122                     N(i,k) = N(i,k) + V(n)*(G(m,n)*
                            cos(th(m)-th(n)) + B(m,n)*sin
                            (th(m)-th(n)));
```

```
123                 end
124                 N(i,k) = N(i,k) + V(m)*G(m,m);
125             else
126                 N(i,k) = V(m)*(G(m,n)*cos(th(m)-th(n
                    )) + B(m,n)*sin(th(m)-th(n)));
127             end
128         end
129     end
130
131     J3 = zeros(npq,nbus-1);
132     for i = 1:npq
133         m = pq(i);
134         for k = 1:(nbus-1)
135             n = k+1;
136             if n == m
137                 for n = 1:nbus
138                     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)));
139                 end
140                 J3(i,k) = J3(i,k) - V(m)^2*G(m,m);
141             else
142                 J3(i,k) = V(m)* V(n)*(-G(m,n)*cos(th
                    (m)-th(n)) - B(m,n)*sin(th(m)-th(
                    n)));
143             end
144         end
145     end
146
147     L = zeros(npq,npq);
148     for i = 1:npq
149         m = pq(i);
150         for k = 1:npq
151             n = pq(k);
152             if n == m
153                 for n = 1:nbus
154                     L(i,k) = L(i,k) + V(n)*(G(m,n)*
                        sin(th(m)-th(n)) - B(m,n)*cos
```

```
                            (th(m)-th(n)));
155                     end
156                     L(i,k) = L(i,k) - V(m)*B(m,m);
157                 else
158                     L(i,k) = V(m)*(G(m,n)*sin(th(m)-th(n
                           )) - B(m,n)*cos(th(m)-th(n)));
159                 end
160             end
161         end
162
163         J = [H N; J3 L];
164
165         X = inv(J)*M;
166         dTh = X(1:npq+npv);
167         dV = X(nbus:nbus+npq-1);
168
169
170         th(2:nbus) = dTh + th(2:nbus);
171         k = 1;
172         for i = 2:nbus
173             if btype(i) == 3
174                 V(i) = dV(k) + V(i);
175                 k = k+1;
176             end
177         end
178
179         Iter = Iter + 1;
180         Tol = max(abs(M));
181
182     end
183     mprintf("\n\n(b)P2= %.3f, P2= %.3f, P3= %.3f, Q3= %
          .3f", P(2),Q(2) ,P(3), Q(3))
184     J=fix(J*1e3)/1e3
185     disp(J, "(c)J")
186
187     mprintf("\n(d)\n")
188     mprintf("%.3f = %.3f dd2 + %.3f dd3 +%.3f dV3/|V3|\n
          ", dP(1), J(1,1), J(1,2), J(1,3) )
```

```
189 mprintf("%.3f = %.3f dd2 + %.3f dd3 +%.3f dV3/|V3|\n
    ", dP(2), J(2,1), J(2,2), J(2,3) )
190 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

solve system using fast decoupled method

```
1  clear
2  clc
3
4  y=[
5  1 2 %i*.11
6  2 3 %i*.11
7  1 3 %i*.11
8  ]
9
10
11 n=3
12 e=3
13 Y=zeros(n,n)
14
15 for i=1:e
16     ynew = 1/(y(i,3))
17     n1=real(y(i,1))
18     n2=real(y(i,2))
19     if(n1==0)
20         Y(n2,n2)=Y(n2,n2)+ynew
21     else
22         Y(n1,n1)= Y(n1,n1)+ynew
23         Y(n1,n2)= Y(n1,n2)-ynew
24         Y(n2,n1)= Y(n2,n1)-ynew
25         Y(n2,n2)= Y(n2,n2)+ynew
26     end
```

```
27 end
28 Y=round(Y*1e3)/1e3
29 disp(Y,"(a)Ybus")
30 // bus no | PL | QL | PG | QG | V | th | btype
31
32 data = [
33 1 %inf %inf 1 .5  1.03 0 1
34 2 1.5 %inf  0 0 1.03 0 2
35 3 0 0  1.2 .5  1 0   3
36 ]
37
38 j = data(:,1);     // Bus number.
39 PG = data(:,2);
40 QG = data(:,3);
41 PL = data(:,4);
42 QL = data(:,5);
43 V = data(:,6);
44 th = data(:,7)* %pi / 180;
45 btype = data(:,8);    // Type of Bus 1-Slack, 2-PV,
     3-PQ.
46 nbus = max(j);
47 P = PG - PL;
48 Q = QG - QL
49 Psp = P;
50 Qsp = Q;
51 G = real(Y);
52 B = imag(Y);
53 for i=1:nbus
54     V(i,1)=V(i,1) * exp (%i * th(i,1))
55 end
56
57 pv = [2]          //whicih bus is PV
58 pq = [3]          //whicih bus is PQ
59 npv = length(pv);
60 npq = length(pq);
61
62 Tol = 1;
63 Iter = 1;
```

```
64 while (Iter == 1)
65
66     P = zeros(nbus,1);
67     Q = zeros(nbus,1);
68
69     for i = 1:nbus
70         for k = 1:nbus
71             P(i) = P(i) + V(i)* V(k)*(G(i,k)*cos(th(
                   i)-th(k)) + B(i,k)*sin(th(i)-th(k)));
72             Q(i) = Q(i) + V(i)* V(k)*(G(i,k)*sin(th(
                   i)-th(k)) - B(i,k)*cos(th(i)-th(k)));
73         end
74     end
75
76     dPa = Psp-P;
77     dQa = Qsp-Q;
78     k = 1;
79     dQ = zeros(npq,1);
80     for i = 1:nbus
81         if btype(i) == 3
82             dQ(k,1) = dQa(i);
83             k = k+1;
84         end
85     end
86     dP = dPa(2:nbus);
87     M = [dP; dQ];
88
89     H = zeros(nbus-1,nbus-1);
90     for i = 1:(nbus-1)
91         m = i+1;
92         for k = 1:(nbus-1)
93             n = k+1;
94             if n == m
95                 for n = 1:nbus
96                     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)));
97                 end
```

```
98                  H(i,k) = H(i,k) - V(m)^2*B(m,m);
99              else
100                 H(i,k) = V(m)* V(n)*(G(m,n)*sin(th(m
                        )-th(n)) - B(m,n)*cos(th(m)-th(n)
                        ));
101             end
102         end
103     end
104
105     N = zeros(nbus-1,npq);
106     for i = 1:(nbus-1)
107         m = i+1;
108         for k = 1:npq
109             n = pq(k);
110             if n == m
111                 for n = 1:nbus
112                     N(i,k) = N(i,k) + V(n)*(G(m,n)*
                            cos(th(m)-th(n)) + B(m,n)*sin
                            (th(m)-th(n)));
113                 end
114                 N(i,k) = N(i,k) + V(m)*G(m,m);
115             else
116                 N(i,k) = V(m)*(G(m,n)*cos(th(m)-th(n
                        )) + B(m,n)*sin(th(m)-th(n)));
117             end
118         end
119     end
120
121     J3 = zeros(npq,nbus-1);
122     for i = 1:npq
123         m = pq(i);
124         for k = 1:(nbus-1)
125             n = k+1;
126             if n == m
127                 for n = 1:nbus
128                     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)));
```

```
129                     end
130                     J3(i,k) = J3(i,k) - V(m)^2*G(m,m);
131                 else
132                     J3(i,k) = V(m)* V(n)*(-G(m,n)*cos(th
                        (m)-th(n)) - B(m,n)*sin(th(m)-th(
                        n)));
133                 end
134             end
135         end
136
137         L = zeros(npq,npq);
138         for i = 1:npq
139             m = pq(i);
140             for k = 1:npq
141                 n = pq(k);
142                 if n == m
143                     for n = 1:nbus
144                         L(i,k) = L(i,k) + V(n)*(G(m,n)*
                            sin(th(m)-th(n)) - B(m,n)*cos
                            (th(m)-th(n)));
145                     end
146                     L(i,k) = L(i,k) - V(m)*B(m,m);
147                 else
148                     L(i,k) = V(m)*(G(m,n)*sin(th(m)-th(n
                        )) - B(m,n)*cos(th(m)-th(n)));
149                 end
150             end
151         end
152
153         J = [H N; J3 L];
154
155         X = inv(J)*M;
156         dTh = X(1:npq+npv);
157         dV = X(nbus:nbus+npq-1);
158
159
160         th(2:nbus) = dTh + th(2:nbus);
161         k = 1;
```

```
162     for i = 2:nbus
163         if btype(i) == 3
164             V(i) = -dV(k) + V(i);
165             k = k+1;
166         end
167     end
168
169     Iter = Iter + 1;
170     Tol = max(abs(M));
171
172 end
173
174 V=round(V*10000)/10000
175 th=round(th*1e5)/1e5
176
177 mprintf("\n(b)P2=%.0f P3=%.0f Q3=%.3f \n", P(2), P
    (3), Q(3))
178 disp(-imag(Y(2:3,2:3)), "(c)B''")
179 disp(-imag(Y(3,3)), "B''''")
180
181 P = zeros(nbus,1);
182 Q = zeros(nbus,1);
183
184 for i = 1:nbus
185     for k = 1:nbus
186         P(i) = P(i) + real(V(i)* V(k)*Y(i,k)* exp(%i
              * (th(k)-th(i))))
187         Q(i) = Q(i) - imag(V(i)* V(k)*Y(i,k)* exp(%i
              * (th(k)-th(i))))
188     end
189 end
190 //"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"
191 mprintf("\n\n(e)P2= %.3f, P3= %.3f, Q3= %.3f", P(2),
     P(3), Q(3))
```

```
192
193 Q(3)=-.49
194 Pl=P(1)+P(2)+P(3)
195 Ql=Q(1)+Q(2)+Q(3)
196
197
198 mprintf("\n\n(f)real line losses= %.1f, reactive
      line losses= %.1f", Pl, Ql)
199
200
201 data(1,2)=PL(1)+P(1);
202 data(1,3)=QL(1)+Q(1);
203 data(2,3)=QL(2)+Q(2);
204
205
206 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))
207
208 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

```
solve system using gauss seidel method with acceleration constant
```

```
1 clear
2 clc
3
4 y=[
5 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);
```

```
44 th = data(:,5);
45 btype = data(:,6);    // Type of Bus 1-Slack, 2-PV,
     3-PQ.
46 n = max(j);
47
48 a=1.6
49
50 for i=1:n
51     V(i,1)=V(i,1) * exp (%i * th(i,1)* %pi / 180)
52 end
53
54 Vprev = V;
55 toler = 1000;
56 iteration = 1;
57 disp("(b)")
58 while (iteration==1)
59     for i = 2:n
60         summ = 0;
61         for k = 1:n
62             if i ~= k
63                 summ = summ + Y(i,k)* V(k);
64             end
65         end
66         V(i) = (1/Y(i,i))*(((P(i)- %i*Q(i))/conj(V(i
             ))) - summ);
67         dv=V(i)-Vprev(i)
68         mprintf("\nV%d = %.3f ang(%.2f) deg", i,abs(
             V(i)),atand(imag(V(i))/real(V(i))))
69         V(i)=Vprev(i)+(a*dv)
70         if btype(i) == 2
71             V(i) = abs(Vprev(i))*exp(%i * atan(imag(
                 V(i))/real(V(i))));
72         end
73     end
74     iteration = iteration + 1;
75     toler = max(abs(abs(V) - abs(Vprev)));
76     Vprev = V;
77 end
```

Chapter 10

Balanced and Unbalanced Faults

Scilab code Exa 10.1 find fault current and fault level

find fault current and fault level

```
1  clear
2  clc
3
4  Sg1=100
5  Vg1=11
6  xg1=.15 *%i
7
8  Sg2=50
9  Vg2=11
10 xg2=.1 *%i
11
12 St1=100
13 Vt1a=11
14 Vt1b=132
15 xt1=.1 *%i
16 nt1=Vt1b/Vt1a
17
```

```
18 St2=50
19 Vt2a=11
20 Vt2b=132
21 xt2=.08 *%i
22 nt2=Vt2b/Vt2a
23
24 Sb=100
25 Vb1=11
26 Vb2=nt1*Vb1
27
28 xl=.2 * 200 *%i
29 Xl=xl/(Vb2*Vb2/Sb)
30
31 Xg2=xg2*Sb/Sg2
32 Xt2=xt2 * Sb/St2
33
34 X=( ((xg1 +xt1) * (Xg2 +Xt2) )/ ((xg1 +xt1) + (Xg2 +
      Xt2) )) +Xl/2
35
36 I=1/X
37
38 Ib1=Sb*1e3/(Vb1*sqrt(3))
39 Ib2=Sb*1e3/(Vb2*sqrt(3))
40 If=abs(I*Ib2)
41 Ifg=abs(I*Ib1)
42 Ifg1=Ifg * (Xg2 +Xt2)/(xg1+xt1+Xt2+Xg2)
43 Ifg2=Ifg * (xg1 +xt1)/(xg1+xt1+Xt2+Xg2)
44 MVAf=abs(I*1*Sb)
45
46 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

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

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

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

```
calculate total generator and motor current in 3phase fault
```

```
clear
clc
```

```
3
4  G=[100  11  .25]
5  M=[50   11  .2  40  .8]
6  xl=.05
7  vt=10.95
8
9  B=[100  11]
10 B(3)= B(1)*1e6/(sqrt(3)*B(2)*1e3)
11
12 IL=M(4)*1e6 / (sqrt(3)*vt*1e3*M(5))
13 Il=round(((IL/B(3) * exp(%i * acos(M(5)))))*1000)
      /1000
14
15 Vt=round(1e3*vt/B(2))/1e3
16
17 V=fix((Vt + Il*xl*%i)*1e4)*1e-4
18
19 M(6)=M(3) *B(1)/M(1)
20
21 xth= round( G(3) * (M(6) +xl)/(G(3) + (M(6) +xl))
      *10000)/10000
22
23 If=fix(V*1e3/(%i *xth))/1e3
24 temp= fix(imag(If)*100)/100
25 If=complex(real(If), temp)
26 Ifg=fix((If * (M(6) +xl)/(G(3) + (M(6) +xl)))*100)
      /100
27 Ifm=round((If * (G(3)    )/(G(3) + (M(6) +xl)))
      *1000)/1000
28
29 Ig=(Ifg + Il) * B(3)
30 Im=(Ifm - Il) *B(3)
31
32 mprintf("\ntotal generator current during fault= %s
      A", string(round(abs(Ig)*10)/10) +'/_'+ string(
      round(atand(imag(Ig)/real(Ig))*10)/10))
33 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

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

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

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

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
```

```
35
36 Z=[Z1 Z2 Z0]'
37
38 mprintf("\nZ1= %.3 fj\n", imag(Z(1)))
39 mprintf("Z2= %.3 fj\n", imag(Z(2)))
40 mprintf("Z0= %.3 fj\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

find fault current voltage at fault point and current and voltage at generator ter

```
1  clear
2  clc
3
4  X=[
5   .25 .25 .05
6   .2 .2 .05
7   .06 .06 .06
8   .07 .07 .07
9   .1 .1 .3
10  .1 .1 .3
11 ]
12
13 B=[
14      100 11
15      100 11
16      100 11
17      100 11
18      100 220
19      100 220
20      ]
21 V1=11
22 V2=220
23 S=100
```

```
24 Xe=3*.03
25 //end 9
26 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
27
28 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
29
30 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
31
32 Z1=%i * round(X1*1e3)/1e3
33 Z2=%i * round(X2*1e3)/1e3
34 Z0=%i * round(X0*1e3)/1e3
35 Z=[Z1 Z2 Z0]'
36 //end ques 11
37
38 a=exp(%i * 2 * %pi /3)
39 A= [1 1 1 ; 1 a^2 a; 1 a a^2]
40
41 //12(a)
42 Ia1a=1/(Z1+Z2+Z0)
43 Ia1a=round(Ia1a *1e2)/1e2
44 mIa1a=[ Ia1a Ia1a Ia1a]'
45 mIa=A*mIa1a
46 Ia=round(mIa(1)*100)/100
47 Iaa = round(abs(mIa(1))*100)/100
48 Iba=round(S*1e7/(sqrt(3) * V2 * 1e3))/10
49 IFa=round(Iba * Iaa *100)/100
```

```
50
51 mprintf("\n(a) Fault current = %.2f A",IFa)
52
53 //12(b)
54
55 Va1=round((1- (Z1 * Ia1a))*100)/100
56 Va2=round((0- (Z2 * Ia1a))*100)/100
57 Va0=round((0- (Z0 * Ia1a))*100)/100
58 mVa1=[ Va0  Va1 Va2]'
59 mVa=A*mVa1
60 v=mVa * V2 / sqrt(3)
61 v=round(v *10000)/10000
62
63 mprintf("\n\n(b)line to neutral voltages in KV ")
64 mprintf("\nVa= %s", string(fix(abs(v(1))*100)/100)
   )
65 mprintf("\nVb= %s", string(fix(abs(v(2))*100)/100) +
   '/_'+ string(round(atand(imag(v(2))/real(v(2)))
   *100)/100 +180 ) )
66 mprintf("\nVc= %s", string(fix(abs(v(3))*100)/100) +
   '/_'+ string(round(atand(imag(v(3))/real(v(3)))
   *100)/100 +180))
67
68 //12(c) --- g2
69
70 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)))
71 Ia1g2=round(Ia1g2 *1e2)/1e2
72
73 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)
     )
```

```
100 mprintf("\nVb= %s", string(fix(abs(vg2(2))*100)/100)
    )
101 mprintf("\nVc= %s", string(fix(abs(vg2(3))*100)/100)
    )
102
103 //12(c) --- g1
104
105 Ia1g1= (Ia1a-Ia1g2) * exp(%i * 1*%pi /6) *-1
106 Ia1g1=round(Ia1g1 *1e2)/1e2
107
108 Ia2g1= (Ia1a-Ia2g2)* exp(%i *-1*%pi /6) *-1
109 Ia2g1=round(Ia2g1 *1e2)/1e2
110
111 Ia0g1= 0
112
113 mIa1g1=[ Ia0g1 Ia1g1 Ia2g1]'
114 mIag1=A*mIa1g1
115 mIag1=round(mIag1*1e1)/1e1
116 Ibc=round(S*1e7/(sqrt(3) * V1 * 1e3))/10
117 Iag1=abs(mIag1) * Ibc
118
119
120 mprintf("\n\nline currents in A at generator 1")
121 mprintf("\nIa= %s", string(round(abs(Iag1(1))*1)/1)
    )
122 mprintf("\nIb= %s", string(round(abs(Iag1(2))*1)/1)
   )
123 mprintf("\nIc= %s", string(round(abs(Iag1(3))*1)/1)
   )
124
125
126 Va1g1=round((1- (X(1,1) *%i * (Ia1a-Ia1g2)))*10000)
   /10000
127 Va1g1=Va1g1 *  exp(%i * 1*%pi /6)
128 Va2g1=round((0- (X(1,2) *%i * (Ia1a-Ia2g2)))*10000)
   /10000
129 Va2g1=Va2g1 *  exp(%i *- 1*%pi /6)
130 Va0g1=0
```

```
131 mVa1g1=[ Va0g1  Va1g1 Va2g1]'
132 mVag1=A*mVa1g1
133 vg1=mVag1 * V1 / sqrt(3)
134
135 mprintf("\n\nline to neutral voltages in KV at
      generator 1 ")
136 mprintf("\nVa= %s", string(fix(abs(vg1(1))*100)/100)
     )
137 mprintf("\nVb= %s", string(fix(abs(vg1(2))*100)/100)
    )
138 mprintf("\nVc= %s", string(fix(abs(vg1(3))*100)/100)
    )
```

Scilab code Exa 10.13 calculate fault current during LG fault

calculate fault current during LG fault

```
1  clear
2  clc
3
4  Sb=37.5
5  Vb=33
6  Zb=Vb*Vb/Sb
7  Ib= Sb *1e6 / (sqrt(3) *Vb *1e3)
8  x1=[.18 .12 .1]
9  x2=[6.3 6.3 12.6]
10 X2=x2/Zb
11
12 X=x1+X2
13
14 x=X(1)+X(2)+X(3)
15
16 If=3*1/x
17 IF=If*Ib
18 mprintf("Fault current is %.1f A", IF)
```

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

```
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
]

```

```
30 Va1=E-Ia1*Z(1)
31 Va2=0-Ia2*Z(2)
32 Va0=0-Ia0*Z(3)
33
34 Vbll=V/sqrt(3)
35 Vp=[ Va0 Va1 Va2]'
36 v=A*Vp*Vbll
37
38 mprintf("\nline to neutral voltages in KV ")
39 mprintf("\nVa= %s", string(round(abs(v(1))*1000)
     /1000) +'/_'+ string(round(atand(imag(v(1))/real(
     v(1)))*10)/10) )
40 mprintf("\nVb= %s", string(round(abs(v(2))*1000)
     /1000) +'/_'+ string(round(atand(imag(v(2))/real(
     v(2)))*10)/10 +180) )
41 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

```
find line currents and voltages under LLG fault conditions
```

```
1  clear
2  clc
3
4  clear
5  clc
6
7  Z=[.2 .2 .05]' * %i
8  S=30
9  V=11
10 I=round(S*1e8/(sqrt(3)*V*1e3) )/1e2
11 E=1
```

```
12
13 Ia1=E/(Z(1)+(Z(2)*Z(3)/(Z(2)+Z(3))))
14 Ia1=round(Ia1*1000)/1000
15 Ia2=-Ia1 * Z(3)/(Z(2)+Z(3))
16 Ia0=-Ia1 * Z(2)/(Z(2)+Z(3))
17
18 a=exp(%i * 2 * %pi/3)
19 A=[1 1 1
20 1 a^2 a
21 1 a a^2
22 ]
23
24 Ia=A*[ Ia0 Ia1 Ia2]' * I
25
26 mprintf("Line currents: ")
27 mprintf("\nIa= %.2f ang(0) A",abs(Ia(1)))
28 mprintf("\nIb= %.2f ang(%.2f) A",abs(Ia(2)),atand(
      imag(Ia(2))/real(Ia(2)))+180)
29 mprintf("\nIc= %.2f ang(%.2f) A",abs(Ia(3)),atand(
      imag(Ia(3))/real(Ia(3))))
30
31
32 If=Ia(2)+Ia(3)
33 mprintf("\nFault current= %.0f ang(%.2f) A",abs(If),
      atand(imag(If)/real(If)))
34
35 Va1=1-(Ia1*Z(1))
36 Va2=Va1
37 Va0=Va1
38
39 Va=A*[Va0 Va1 Va2]' * V/sqrt(3)
40 mprintf("\nLine to neutral voltages: ")
41 mprintf("\nVa= %.3f ang(%.2f) kV",abs(Va(1)),atand(
      imag(Va(1))/real(Va(1))))
42 mprintf("\nVb= %.3f KV",abs(Va(2)))
43 mprintf("\nVc= %.3f KV",abs(Va(3)))
```

Scilab code Exa 10.16 find line currents under LG fault conditions

```
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
```

```
27
28 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
29
30 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
31
32 Z1=%i * round(X1*1e3)/1e3
33 Z2=%i * round(X2*1e3)/1e3
34 Z0=%i * round(X0*1e3)/1e3
35
36 //end q11
37
38 a=exp(%i * 2 * %pi /3)
39 A= [1 1 1 ; 1 a^2 a; 1 a a^2]
40
41
42 Ia1=1/(Z1 + (Z2*Z0/(Z2+Z0)))
43 Ia1=round(Ia1 *1e3)/1e3
44 Ia2=(Z0/(Z2+Z0)) * Ia1 *-1
45 Ia2=round(Ia2 *1e3)/1e3
46 Ia0=(Z2/(Z2+Z0)) * Ia1 *-1
47 Ia0=round(Ia0 *1e3)/1e3
48
49 mIa1=[ Ia0 Ia1 Ia2]'
50 mIa=A*mIa1
51 Ib=round(S*1e7/(sqrt(3) * V2 * 1e3))/10
52 Ia=round(abs(mIa)) * Ib
53
54
55 mprintf("\nline currents are: in Amperes ")
56 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

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

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
```

```
32 Ia1=V/(%i*((x1g*(x1m+xt1)/(x1g+x1m+xt1))+(x2g*(x2m+
     xt2)/(x2g+x2m+xt2))+(x0g*(x0m+xt0)/(x0g+x0m+xt0))
     ))
33 Ia1=round(Ia1 * 1000)/1000
34 Ig1=round((Ia1 * (x1m + xt1)/(x1m+x1g+xt1) + Il)
     *1000)/1000
35 Im1=round((Ia1 * (x1g)/(x1m+x1g+xt1) - Il)*1000)
     /1000
36 Ig2=round((Ia1 * (x2m + xt2)/(x2m+x2g+xt2))*1000)
     /1000
37 Im2=round((Ia1 * (x2g)/(x2m+x2g+xt2))*1000)/1000
38 Ig0=round((Ia1 * (x0m + xt0)/(x0m+x0g+xt0))*100)/100
39 Im0=round((Ia1 * (x0g)/(x0m+x0g+xt0))*100)/100
40
41 Im=round((Im1+Im2+Im0)*1000)  *1e-3
42 Ig=round((Ig1+Ig2+Ig0)*1000)  *1e-3
43
44 mprintf("\nCurrent Through motor = %.2f, ang (%.1f)
     deg", abs(Im)*I, atand(imag(Im)/real(Im))+180)
45 mprintf("\nCurrent Through generator = %.2f, ang (%
     .1f) deg", abs(Ig)*I, atand(imag(Ig)/real(Ig)))
46 disp("Error in evaluation of base current in
     textbook")
```

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

```
find short circuit MVA of parallel connection of 2 stations
```

```
1 clear
2 clc
3
4 S=100
5 V=11
6 Z=V*V/S
```

```
7
8  SCA=1000
9  SCB=650
10
11 xa=S/SCA
12 xb=S/SCB
13 Xc=.5
14 xc=Xc/Z
15
16 X=round(((xa *(xb+xc))/(xa+xb+xc))*1000)/1000
17 FS=S/X
18 mprintf("Fault MVA= %.2f MVA",FS)
```

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

```
find X to prevent overloading of circuit breakers
```

```
1  clear
2  clc
3
4  V=33
5  S=75
6  Z=V*V/S
7
8  sg=15
9  xg=.15
10 Xg=xg *S/sg
11 xt=.08
12 x=Xg/3
13
14 CB=750
15 XF=S/CB
16 xi=((x*xt)-(XF*(x+xt)))/(XF-x)
17 xi=round(xi*10000)/10000
18 X=xi*Z
```

```
19 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

`determine fault current and voltages during LG fault when different alternator neu`

```
1  clear
2  clc
3
4  V=6.6
5  S=10
6  Z=V*V/S
7  I=S*1e6/(sqrt(3)*V*1e3)
8
9  X1=.15
10 X2=.75*X1
11 X0=.3*X1
12
13 x1=X1/3
14 x2=X2/3
15 x0=X0/3
16
17 E=1
18
19 x1=X1/3
20 x2=X2/3
21 x0=X0/3
22 IFa=abs(3*E/(%i * (x1+x2+x0)))
23 IFa=round(IFa*1000)/1000
24 ifa=IFa * I
25 mprintf("\n(a)fault current when all gen neutrals
      gounded= %.2f A",ifa)
26
27 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

```
determine fault current and voltages during LG fault when alternator neutral is gr
```

```
clear
clc

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

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

```

```
12 x1l=.4
13 x2l=.4
14 x0l=.8
15
16 x1lm=x1l/2
17 x2lm=x2l/2
18 x0lm=x0l/2
19
20 X1=x1g+x1lm
21 X2=x2g+x2lm
22 X0=(x0g+x0lm)*(x0lm+x0t)/(x0g+x0lm+x0lm+x0t)
23
24 X=X1+X2+X0
25 Ia1=round((1/(%i*X))*1000)/1000
26 Ia2=round((Ia1)*1000)/1000
27 Ia0=round((Ia1)*1000)/1000
28 Ia=Ia1+Ia2+Ia0
29 IFa=abs(Ia)
30
31 Va1=1-(Ia1 * X1 *%i)
32 Va2=0-(Ia2 * X2 *%i)
33 Va0=0-(Ia0 * X0 *%i)
34
35 a=exp(%i * 2 * %pi/3)
36 A=[1 1 1
37 1 a^2 a
38 1 a a^2
39 ]
40
41 va1=[Va0 Va1 Va2]'
42 Va=A*va1
43 mprintf("\n(a)")
44 mprintf("\If= %.3f ang(%.2f)",abs(Ia),270)
45 mprintf("\nVa= %.3f ang(%.2f)",abs(Va(1)),atand(imag
      (Va(1))/real(Va(1))))
46 mprintf("\nVb= %.3f ang(%.2f)",abs(Va(2)),atand(imag
      (Va(2))/real(Va(2)))+180)
47 mprintf("\nVc= %.3f ang(%.2f)",abs(Va(3)),atand(imag
```

```
    (Va(3))/real(Va(3)))+180)
48
49
50 X1=x1g+x1lm
51 X2=x2g+x2lm
52 X0=(x0g+x0lm)
53
54 X=X1+X2+X0
55 Ia1=round((1/(%i*X))*1000)/1000
56 Ia2=round((Ia1)*1000)/1000
57 Ia0=round((Ia1)*1000)/1000
58 Ia=Ia1+Ia2+Ia0
59 IFa=abs(Ia)
60
61 Va1=1-(Ia1 * X1 *%i)
62 Va2=0-(Ia2 * X2 *%i)
63 Va0=0-(Ia0 * X0 *%i)
64
65 a=exp(%i * 2 * %pi/3)
66 A=[1 1 1
67 1 a^2 a
68 1 a a^2
69 ]
70
71 va1=[Va0 Va1 Va2]'
72 Va=A*va1
73 mprintf("\n(b)")
74 mprintf("\If= %.3f ang(%.2f)",abs(Ia),270)
75 mprintf("\nVa= %.3f ang(%.2f)",abs(Va(1)),atand(imag
    (Va(1))/real(Va(1))))
76 mprintf("\nVb= %.3f ang(%.2f)",abs(Va(2)),atand(imag
    (Va(2))/real(Va(2)))+180)
77 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

```
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

```
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

```
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
```

```
28 FMVAb=IFb * sqrt(3) * V2 * 1e-3
29 mprintf("\n(b)single line to ground fault Fault MVA=
     %.2f MVA, Fault Current=%.2f A",FMVAb,IFb)
30
31 IFc=abs(sqrt(3)/(X1+X2))* I2
32 FMVAc=IFc * sqrt(3) * V2 * 1e-3
33 mprintf("\n(c)L-L Fault MVA=%.2f MVA, Fault Current
     =%.1f A",FMVAc,IFc)
34
35 IFd=abs(1/(X1+(X2*X0/(X2+X0))))* I2
36 FMVAd=IFd * sqrt(3) * V2 * 1e-3
37 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

```
find fault current and fault level LG fault in middle of line
```

```
1  clear
2  clc
3
4  a=exp(%i * 2*%pi/3)
5
6  V=33
7  S=45
8  SF=2000
9  V2=132
10 Z=V2*V2/S
11 I=S*1e6 /(sqrt(3) * V2 * 1e3)
12
13 X1=.4* 60
14 X2=.4* 60
15 X0=1 * 60
16
```

```
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)


```

```
53
54 //(c)
55
56
57 Ia0=0
58 Ia1=IA1 * exp(%i * -1*%pi/6)
59 Ia2=IA2 * exp(%i * 1*%pi/6)
60
61 Ia=round((Ia1 +Ia2 +Ia0)*1000)/1000
62 Ib=round(((Ia1*a*a) + (a*Ia2) +Ia0)*1000)/1000
63 Ic=round(((Ia1*a) + (a*a*Ia2) +Ia0)*1000)/1000
64 mprintf("\n(c) Currents in lines connecting source
     to T1 (in order - Ia,Ib, Ic in pu)")
65 disp(Ic, Ib, Ia)
66
67
68 mprintf("\n(d) Currents in star wdg of T1")
69 mprintf("\nIa= %.4fj , Ib= %.4fj , Ic= %.4fj", imag(IA
     ), imag(IB), imag(IC))
70 mprintf("\nIa1= %.4fj , Ia2= %.4fj , Ia0= %.4fj", imag
     (IA1), imag(IA2), imag(IA0))
71 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))
72 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))
73
74
75 Iab1 =  round(IA1*1e3 /sqrt(3))/1e3
76 Iab2 =  round(IA2*1e3 /sqrt(3))/1e3
77 Iab0 =  round(IA0*1e3 /sqrt(3))/1e3
78
79 Ibc1=Iab1 * a*a
80 Ibc2=Iab2 * a
81 Ibc0=Iab0
```

```
82
83 Ica1=Iab1 * a
84 Ica2=Iab2 * a *a
85 Ica0=Iab0
86
87 Iab= Iab1 +Iab2 +Iab0
88 Ibc= Ibc1 +Ibc2 +Ibc0
89 Ica= Ica1 +Ica2 +Ica0
90
91 mprintf("\n\nCurrents in delta wdg of T1")
92 mprintf("\nIab= %.4fj , Ibc= %.4fj , Ica= %.4fj", imag
   (Iab), imag(Ibc), imag(Ica))
93 mprintf("\nIab1= %.4fj , Iab2= %.4fj , Iab0= %.4fj",
   imag(Iab1), imag(Iab2), imag(Iab0))
94 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))
95 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))
96
97 mprintf("\n\n Currents in star wdg of T2")
98 mprintf("\nIa= %.4fj , Ib= %.4fj , Ic= %.4fj", imag(
   IAb), imag(IBb), imag(ICb))
99 mprintf("\nIa1= %.4f, Ia2= %.4f, Ia0= %.4fj", imag(
   IA1b), imag(IA2b), imag(IA0b))
100 mprintf("\nIb1= %.4f, Ib2= %.4f, Ib0= %.4fj", imag(
   IA1b *a*a), imag(IA1b *a),imag(IA0b))
101 mprintf("\nIc1= %.4f, Ic2= %.4f, Ic0= %.4fj", imag(
   IA1b   *a), imag(IA1b*a *a),imag(IA0b))
102
103
104 Iab1b =  round(IA1b*1e3 /sqrt(3))/1e3
105 Iab2b =  round(IA2b*1e3 /sqrt(3))/1e3
106 Iab0b =  round(IA0b*1e3 /sqrt(3))/1e3
107
```

```
108 Ibc1b=Iab1b * a*a
109 Ibc2b=Iab2b * a
110 Ibc0b=Iab0b
111
112 Ica1b=Iab1b * a
113 Ica2b=Iab2b * a *a
114 Ica0b=Iab0b
115
116 Iabb= Iab1b +Iab2b +Iab0b
117 Ibcb= Ibc1b +Ibc2b +Ibc0b
118 Icab= Ica1b +Ica2b +Ica0b
119
120 mprintf("\n\nCurrents in delta wdg of T2")
121 mprintf("\nIab= %.3fj , Ibc= %.3fj , Ica= %.3fj", imag
      (Iabb), imag(Ibcb), imag(Icab))
122 mprintf("\nIab1= %.3f , Iab2=%.3f , Iab0= %.3fj", imag
      (Iab1b), imag(Iab2b), imag(Iab0b))
123 mprintf("\nIbc1= %.3f , Ibc2=%.3f , Ibc0= %.3fj", imag
      (Ibc1b),imag(Ibc2b),imag(Ibc0b))
124 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

```
find fault current and fault level LG fault
```

```
1 clear
2 clc
3
4 R1=4
5 R2=2
6
7 S=50
8 V1=11
9 V2=132
```

```
10 V3=33
11 Z1=V1*V1/S
12 Z2=V2*V2/S
13 Z3=V3*V3/S
14
15 r1=3*R1/Z1
16 r2=3*R2/Z3
17
18
19 x1g=.4*%i
20 x2g=.3*%i
21 x0g=.1*%i
22
23 x1t1=.08*%i
24 x2t1=.08*%i
25 x0t1=.08*%i
26
27 x1t2=.05*%i
28 x2t2=.05*%i
29 x0t2=.05*%i
30
31 x1t3=.04*%i
32 x2t3=.04*%i
33 x0t3=.04*%i
34
35 x1t4=.06*%i
36 x2t4=.06*%i
37 x0t4=.06*%i
38
39
40 X1l=20*%i
41 X2l=20*%i
42 X0l=50*%i
43
44 x1l=X1l/Z2
45 x2l=X2l/Z2
46 x0l=X0l/Z2
47
```

```
48 X1=x1g+x1t2+x1l+x1t1+x1t3
49 X2=x2g+x2t2+x2l+x2t1+x2t3
50 X0=r2+(((x0t2+x0l+x0t1)*x0t4/(x0t2+x0l+x0t1+x0t4))+
      x0t3)
51
52 IF=abs(3*1/(X1+X2+X0))
53 IB=S*1e6/(sqrt(3)*V3*1e3)
54 If=IF*IB
55 SF=IF*S
56 mprintf("fault current= %.0fA, fault level=%.2f MVA"
      ,If,SF)
```

Scilab code Exa 10.31 find line voltages and currents for OC fault

```
find line voltages and currents for OC fault
```

```
1  clear
2  clc
3
4  a=exp(%i *2*%pi/3)
5
6  Z1=complex(2.8,1)
7  Z2=complex(.1,.6)
8
9  V=400
10 E=V/sqrt(3)
11
12 Ia1=E/(Z1+Z2)
13 Ia2=-Ia1
14
15 Ia=Ia1+Ia2
16 Ib= (a^2-a)*Ia1
17 Ic=-Ib
18
19 disp( "Line Currents Ia, Ib, Ic, in amperes")
```

```
20 mprintf("\nIa= %s", string(round(abs(Ia)*10)/10) +'/
   _'+ string(0) )
21 mprintf("\nIb= %s", string(round(abs(Ib)*10)/10) +'/
   _'+ string(round(atand(imag(Ib)/real(Ib))*100)
   /100 -180) )
22 mprintf("\nIc= %s", string(round(abs(Ic)*10)/10) +'/
   _'+ string(round(atand(imag(Ic)/real(Ic))*100)
   /100) )
23
24 Va2=-Z2 * Ia2
25 Vaa=3*Va2
26 Van=(Z1*Ia1)+(Z2*Ia2)
27 Vcn=(a*Z1*Ia1)+(a*a*Z2*Ia2)
28 Vbn=(a*a*Z1*Ia1)+(a*Z2*Ia2)
29 VNn=Va2
30
31
32 mprintf("\n\nVaa= %s", string(round(abs(Vaa)*100)
   /100) +'/_'+ string(round(atand(imag(Vaa)/real(
   Vaa))*10)/10))
33 mprintf("\nVan= %s", string(round(abs(Van)*100)/100)
    +'/_'+ string(round(atand(imag(Van)/real(Van))
   *10)/10))//error in value substitution in
   textbook
34 mprintf("\nVbn= %s", string(round(abs(Vbn)*10)/10) +
   '/_'+ string(round(atand(imag(Vbn)/real(Vbn))*10)
   /10 -180))
35 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
36 mprintf("\nVNn= %s", string(round(abs(VNn)*100)/100)
    +'/_'+ string(round(atand(imag(VNn)/real(VNn))
   *10)/10))
37 disp("error is due to mistake in value substitution
   in textbook")
```

Scilab code Exa 10.32 fault MVA with and without reactors

```
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

```
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

```
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

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

```

```
18 disp(round(Ia1*100)/100, "Ia1","(a)Symmetrical Line
     Components")
19 disp(round(Ia2*100)/100, "Ia2")
20 disp(round(Ia0*100)/100, "Ia0")
21
22 Iac0=(Iac+Icb+Iba)/3
23 Iac1=(Iac+ a*Icb+ a*a*Iba)/3
24 Iac2=(Iac+a*a*Icb+a*Iba)/3
25
26 disp(round(Iac1*100)/100, "Iac1","(b)Symmetrical
     Delta Components")
27 disp(round(Iac2*100)/100, "Iac2")
28 disp(round(Iac0*100)/100, "Iac0")
```

Chapter 11

Digital Techniques in Fault Calculations

Scilab code Exa 11.1 z bus formulation

z bus formulation

```
1  clear;
2  clc;
3
4  n=5;//no of elements
5  Z=0;
6  z=[ 4 1 1 1; 4 2 1 1; 4 3 1 1; 1 2 1 4; 1 3 1 4];
7
8  // z = [ from node | to node| z between nodes | type
       modification] type modification should be in
     ascending order
9
10 for(i=1:n)
11      mcase=z(i,4)
12      znew=z(i,3)
13      n1=z(i,1)
14      n2=z(i,2)
15      dim=max(size(Z))
```

```
16      select mcase
17      case 1 then
18              if Z(1,1)==0 then
19                  dim=dim-1
20              end
21             Z(dim+1, dim+1)=znew
22          case 2 then
23              Z(1:dim,dim+1)=Z(1:dim, n1)
24              Z(dim+1,1:dim)=Z(n1,1:dim)
25              Z(dim+1, dim+1)=znew+Z(n1,n1)
26          case 3 then
27              Z=Z-((Z(1:dim, n2)*Z(n2,1:dim))/(znew+Z(
                   n2,n2)))
28          case 4 then
29              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))))
30          else
31              break
32      end
33  end
34  disp(Z)
```

Scilab code Exa 11.2 formulate positive and negative sequence impedance matrices for the network

formulate positive and negative sequence impedance matrices for the network

```
1  clear;
2  clc;
3
4  n=5;
5  Z=0;
6  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];
```

```
7
8  for(i=1:n)
9      mcase=z(i,4)
10     znew=z(i,3)
11     n1=real(z(i,1))
12     n2=real(z(i,2))
13     dim=max(size(Z))
14     select mcase
15         case 1 then
16             if Z(1,1)==0 then
17                 dim=dim-1
18             end
19            Z(dim+1, dim+1)=znew
20         case 2 then
21             Z(dim+1,dim+1)=znew+Z(n1,n1)
22             Z(1:dim,dim+1)=Z(1:dim, n1)
23             Z(dim+1,1:dim)=Z(n1,1:dim)
24         case 3 then
25
26             Z=Z-((Z(1:dim, n2)*Z(n2,1:dim))/(znew+Z(
                   n2,n2)))
27         case 4 then
28             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))))
29         else
30             break
31     end
32 end
33 mprintf("Z1bus=Z2bus=");
34 disp(Z)
```

Scilab code Exa 11.3 formulate zero sequence impedance matrices for the network

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

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
```

```
28              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))))
29          else
30              break
31      end
32  end
33
34  E=1
35  V=ones(1,4);
36  Ib=262.4;
37  i2=V(1,2) / Z(2,2);
38  I2=Ib*i2;
39
40  Ia=I2 * exp(%i * 0);
41  Ib=I2 * exp(%i *-2* %pi /3);
42  Ic=I2 * exp(%i *2 * %pi /3);
43  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)))
44
45
46  Vb=220;
47  v3=E* (1-(Z(3,2)/Z(2,2)))
48  V3=v3*Vb/sqrt(3);
49  Va=V3 * exp(%i * 0);
50  Vb=V3 * exp(%i *-2* %pi /3);
51  Vc=V3 * exp(%i *2 * %pi /3);
52  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

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
```

```
28              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))))
29          else
30              break
31      end
32  end
33  Z1=Z;
34  Z2=Z;
35
36  n=5;
37  Z=0;
38  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];
39
40  for(i=1:n)
41      mcase=z(i,4)
42      znew=z(i,3)
43      n1=real(z(i,1))
44      n2=real(z(i,2))
45      dim=max(size(Z))
46      select mcase
47          case 1 then
48              if Z(1,1)==0 then
49                  dim=dim-1
50              end
51             Z(dim+1, dim+1)=znew
52          case 2 then
53              Z(dim+1,dim+1)=znew+Z(n1,n1)
54              Z(1:dim,dim+1)=Z(1:dim, n1)
55              Z(dim+1,1:dim)=Z(n1,1:dim)
56          case 3 then
57
58              Z=Z-((Z(1:dim, n2)*Z(n2,1:dim))/(znew+Z(
                   n2,n2)))
59          case 4 then
60              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))))
61          else
62              break
63      end
64  end
65  Z0=Z;
66
67  Ib=262.4;
68  Vb=220;
69  E=1
70  V=ones(1,4);
71
72  I3(3,1)=E/(Z1(3,3)+Z2(3,3)+Z0(3,3));
73  I3(1,1)=I3(3,1)
74  I3(2,1)=I3(3,1)
75  i3=I3*Ib
76  a=exp(%i * 2*%pi/3)
77  A=[1 1 1; 1 a^2 a; 1 a a^2]
78  I=A*i3;
79  //I=round(I*100)/100;
80  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)
81
82
83  V2(1,1)=-1* Z0(2,3)*I3(3,1);
84  V2(2,1)=E-( Z1(2,3)*I3(1,1));
85  V2(3,1)=-1* Z2(2,3)*I3(2,1);
86  v=A*V2;
87  V=Vb*v/sqrt(3);
88  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

```
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
```

```
28              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))))
29          else
30              break
31      end
32  end
33  mprintf("Zbus=");
34  Z=round(Z*1e5)/1e5
35  disp(Z)
```

Scilab code Exa 11.7 find z bus of an augmented network

```
find z bus of an augmented network
```

```
1   clear;
2   clc;
3
4   no=4;
5   Z=0;
6   z=[ 4 1 .5*%i 1; 4 2 .4*%i 1; 1 3 .2*%i 2; 2 3 .1*%i
        4];
7
8   for(i=1:no)
9       mcase=z(i,4)
10      znew=z(i,3)
11      n1=real(z(i,1))
12      n2=real(z(i,2))
13      dim=max(size(Z))
14      select mcase
15          case 1 then
16              if Z(1,1)==0 then
17                  dim=dim-1
18              end
19             Z(dim+1, dim+1)=znew
```

```
20          case 2 then
21              Z(dim+1,dim+1)=znew+Z(n1,n1)
22              Z(1:dim,dim+1)=Z(1:dim, n1)
23              Z(dim+1,1:dim)=Z(n1,1:dim)
24          case 3 then
25
26              Z=Z-((Z(1:dim, n2)*Z(n2,1:dim))/(znew+Z(
                    n2,n2)))
27          case 4 then
28              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))))
29          else
30              break
31      end
32  end
33
34
35  m=1
36  n=3
37  p=1
38  q=4
39  no2=4
40  znew=.5*%i
41  zm=.1*%i
42  za=.2*%i
43
44  for j=1:no2
45      if j==q then
46          Z(q,q)=Z(p,q)-((zm/za)*(Z(m,q)-Z(n,q)))-((zm
                *zm/za)-znew);
47      else
48          Z(q,j)=Z(p,j)-((zm/za)*(Z(m,j)-Z(n,j)))
49          Z(j,q)=Z(q,j)
50      end
51  end
52  Z=round(Z*1e5)/1e5
53  disp(Z)
```

Chapter 12

Power System Transients

Scilab code Exa 12.1 find L C surge impedance and velocity of propagation

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

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

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

```
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

```
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)
```

```
12      f=(k*R*x^6) +x -(2*V)
13      df=(6* k*R*x^5) +1
14      x1=x-(f/df)
15      E=abs(x1-x)
16      x=x1
17  end
18  eA=round(x)
19  IA=k*eA^6
20
21  mprintf("eA=%d, Ia=%d ",eA, IA)
```

Scilab code Exa 12.8 find surge arrester voltage and current

find surge arrester voltage and current

```
1   clear
2   clc
3
4   V=300e3
5   R1=400
6   R2=50
7   R=1+(400/50)
8   k=1.5e-27
9
10  E=10
11  x=1
12  e=1e-5
13  while (E>e)
14      f=(k*R1*x^6) +(R*x) -(2*V)
15      df=(6* k*R1*x^5) +R
16      x1=x-(f/df)
17      E=abs(x1-x)
18      x=x1
19  end
20  eA=round(x)
```

```
21 IA=k*eA^6
22
23 mprintf("eA=%d, Ia=%.1f ",eA, IA)
```

Scilab code Exa 12.9 find reflection and refraction coefficients

find reflection and refraction coefficients

```
1  clear;
2  clc;
3
4  ef=3000;
5  Zc=300;
6  ea=1700;
7  iF=ef/Zc
8  mprintf("\nCurrent in line= %d kA",iF)
9  Ia=((2*ef)-ea)/Zc
10 mprintf("\nCurrent through Arrester= %.3f kA",Ia)
11 Ia=round(Ia *1000)/1000
12 R=ea/Ia
13 mprintf("\nresistance of arrester= %.2f ohm",R)
14 er=ea-ef;
15 mprintf("\nSurge Voltage Reflected= %.0f kV",er)
16 Cr=er/ef;
17 CR=ea/ef;
18 mprintf("\nCoeff of Reflection = %.3f, Coeff of
      Refraction=%.3f",Cr,CR)
19 Cr=(R-Zc)/(R+Zc);
20 CR=(R*2)/(R+Zc);
21 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

```
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;
```

```
29 mprintf("\nSurge Voltage Reflected= %.3f kV",er
   /1000)
30 ir=-1*er/Zc
31 mprintf("\nSurge Current Reflected= %.3f A",ir)
32 edr=et*it;
33 mprintf("\nRate of dissipation of energy= %.2f KW",
   edr/1000)
34 err=er*-ir;
35 mprintf("\nRate of reflection of energy= %.2f KW",
   err/1000)
```

Scilab code Exa 12.11 find V and I transmitted

```
find V and I transmitted
```

```
1  clear;
2  clc;
3
4  Zc=400
5  ef=20
6  z1=150;
7  z2=200
8  z=round((z1*z2/(z1+z2))*100)/100
9
10 et=2*ef*z/(Zc+z)
11 mprintf("\nSurge Voltage Transmitted= %.4f kV",et)
12
13 it1=et*1000/z1;
14 mprintf("\nSurge Current Transmitted in line 1= %.3f
    A",it1)
15
16 it2=et*1000/z2;
17 mprintf("\nSurge Current Transmitted in line 2= %.2f
    A",it2)
18
```

```
19 er=et-ef
20 mprintf("\nSurge Voltage Reflected= %.4f kV",er)
21 ir=-1*er*1000/Zc
22 mprintf("\nSurge Current Reflected= %.2f A",ir)
```

Scilab code Exa 12.12 reflection transmission and absorption of wave

reflection transmission and absorption of wave

```
1  clear;
2  clc;
3
4  ef=100
5  Zc=400
6  z=50
7
8  R=z+Zc;
9  E=(2*ef/(Zc+z+R))^2 *R
10 E=round(E*100)/100
11 mprintf("\n(a)Energy transfer max when R= %.0f ohm,
       energy= %.2f KW", R,E);
12
13 etB=2*ef*z/(z+Zc+R);
14 etB=round(etB*100)/100
15 mprintf("\n(b)Surge Voltage Transmitted= %.3f kV",
       etB)
16
17 it=etB*1000/z;
18 it=round(it*100)/100
19 mprintf("\nSurge Current Transmitted = %.2f A",it)
20
21 etA=2*ef*(z+R)/(z+Zc+R);
22 etA=round(etA*100)/100
23
24 erA=etA-ef
```

```
25 mprintf("\n(c)Surge Voltage Reflected= %.2f kV",erA)
26 irA=-1*erA*1000/Zc
27 mprintf("\nSurge Current Reflected= %.3f A",irA)
28
29
30 iF=ef*1000/Zc
31 Pi=ef*iF
32 mprintf("\n(d)Power Incident= %.0f kW",Pi)
33 Pr=erA*-irA
34 mprintf("\nPower Reflected= %.2f kW",Pr)
35 Pt=erA*it
36 mprintf("\nPower Transmitted= %.0f kW",Pt)
```

Scilab code Exa 12.14 find voltage and current surges

```
find voltage and current surges
```

```
1 clear;
2 clc
3
4 I=5;
5 z1=400
6 z2=50
7 V=I * z1* z2/(z2+z1)
8 mprintf("\nSurge Voltage Transmitted= %f kV",V)
9 ic=V/z2
10 mprintf("\nSurge Current Transmitted in cable= %f kA
    ",ic)
11
12 io=-V/z1;
13 mprintf("\nSurge Current Transmitted in OH line= %f
    kA",io)
```

Scilab code Exa 12.16 find restriking voltage due to current chopping

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

find P Q E and load angle for changes to P and E

```
1  clear;
2  clc
3
4  Xd=.7
5  pf=.8
6  pfa=acos(pf)
7  V=1
8  I0=1* exp( %i * pfa *-1)
9  E0=V+ (%i * Xd * I0)
10 E=round(abs(E0)*100)/100
11 d0=atand(imag(E0)/real(E0))
12 E0=E * exp(%i * d0 * %pi/180)
13 Pe0=E*V*sind(d0)/Xd
14 Qe0=(E*V*cosd(d0)/Xd)-(V*V/Xd)
15
16 mprintf("\n(a)\nPe= %.1f Qe=%.1f E= %.2f load angle=
     %.1f",Pe0, Qe0, E, d0);
17
18 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

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)
```

```
15 a=Pa/M
16 mprintf("\n(b)acceleration = %.0f elec deg/ sec sq",
      a)
17
18 c=5
19 t=c/f
20 dd=.5*a*t*t
21 N=120*f/P
22 a=a*60/(180*P)
23 Nn=N+(a*t)
24 mprintf("\n(c)change in power angle = %.1f elec deg,
      change in speed = %.1f rpm",dd,Nn)
25
26 G2=150
27 H2=4
28 Gb=100
29 Heq=((G*H)+(H2*G2))/Gb
30 mprintf("\n(d)Equivalent inertia constant = %.0f MJ/
      MVA",Heq)
```

Scilab code Exa 13.3 find steady state stability parameters

find steady state stability parameters

```
1 clear;
2 clc
3
4 f=50;
5 H=9;
6 x=.6
7 P=.7
8 pf=.8
9 pfa=acos(pf)
10 V=1
11 D=.14
```

```
12 dd=10 * %pi/180
13
14 S=P/pf * exp(%i * pfa)
15 I=conj(S/V)
16 E=V+ (%i * x *I)
17 d0=atand(imag(E)/real(E))
18 Pr=abs(E) * V * cosd(d0)/x
19 mprintf("\nSynchronising Power Coefficient = %.3f pu
     ",Pr)
20
21 w1=sqrt(Pr *f *%pi/H)
22 mprintf("\nUndamped Speed Of oscillations = %.2f rad
     /sec",w1)
23 w1=round(w1*100)/100
24
25 z=(D/2) * sqrt(%pi * f/(H*Pr))
26 mprintf("\nDamping Ratio = %.4f ",z)
27
28 wd=w1 *sqrt(1-(z*z))
29 Wd=wd / (2*%pi)
30 mprintf("\ndamped angular frequency Of oscillations
     = %.3f rad/sec = %.3f Hz",wd, Wd)
31
32 z=round(z*1e4)/1e4
33 c_1=10/sqrt(1-z^2)
34 c_2=z*w1
35
36 mprintf("\n\nd = %.2f + %.3f exp(-%.3f t) sin(%.3f t
     + %.1f deg)", d0, c_1, c_2,wd, acosd(z))
37
38 c_3=w1 * 10 /(360 * sqrt(1-z^2))
39 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

```
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)

```

```
33 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))
34
35 c_4= dP * f /(w1* 9 * sqrt(1-z^2))
36 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

```
finding steady state reactance and transfer limit for different shunt branches
```

```
1  clear;
2  clc
3
4  E=1.1
5  V=1
6  xg=.7
7  xt=.1
8  x1=.2
9  X1=x1+xg+xt
10 X2=.2
11 xL=.5
12 xC=-.5
13
14
15 X=X1+X2
16 P1=E*V/X
17 mprintf("\n(a) Steady State limit = %.4f", P1)
18
19 X3=xL
20 X=((X1*X2) + (X2*X3) + (X1*X3))/X3
21 P2=E*V/X
22 mprintf("\n(b) Steady State limit = %.4f", P2)
```

```
23
24 X3=xC
25 X=((X1*X2) + (X2*X3) + (X1*X3))/X3
26 P3=E*V/X
27 mprintf("\n(c) Steady State limit = %.3f", P3)
28
29 X=X1+xC+X2
30 P4=E*V/X
31 mprintf("\n(d) Steady State limit = %.2f", P4)
```

Scilab code Exa 13.6 frequency of oscillation of generator due to loading

frequency of oscillation of generator due to loading

```
1  clear;
2  clc
3
4  f=50
5  xg=1
6  xl=.5
7  E=1.1
8  V=1
9  H=5
10 p=.5
11
12 X=xl+xg
13 d0=asin(p)
14 Pr=E*V*cos(d0)/X
15 M=H/(%pi*f)
16 wn=sqrt(Pr/M)/(2*%pi)
17
18 mprintf("Freg of oscillation = %.2f Hz", wn)
```

Scilab code Exa 13.7 system stability and finding critical load angle

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
```

```
33 a=Pi
34 b=pe2
35 c=-A1 +(Pi * d1 ) + (cos (d1) * pe2)
36
37 while (E>e)
38     f=(a*x) +  (b * cos(x)) - c
39     df=1.5 - (2*sin(x))
40     x1=x-(f/df)
41     E=abs(x1-x)
42     x=x1
43 end
44 d2=x1 * 180/%pi
45 mprintf("\ndelta 2 = %.2f deg",d2)
```

Scilab code Exa 13.8 system stability and finding critical load angle in 3 phase fault in line

system stability and finding critical load angle in 3 phase fault in line

```
1  clear
2  clc
3
4  xd=.2
5  x1=.4
6  x2=.4
7  Pi=1.5
8  E=1.2
9  V=1
10
11 Xs1=xd  +((x1*x2)/(x1+x2))
12 pe=E*V/Xs1
13 d0 = asin(Pi/pe)
14 dc= (%pi/2)-d0
15 dc=round(dc*1e3)/1e3
16
```

```
17 X1=x1;
18 X2=x2/2
19 X3=x2/2
20 Xs2=((X1*X2) + (X2*X3) + (X1*X3))/X3
21 pe2=E*V/Xs2
22
23
24
25 Xs3=xd+ x1
26 pe3=E*V/Xs3
27 d2 = asin(Pi/pe3)
28 dm=%pi- d2
29
30
31 if pe2<Pi then
32     mprintf("UNSTABLE for sustained fault\n\n")
33 else
34     mprintf("STABLE for sustained fault\n\n")
35 end
36
37
38 A1=((Pi * dc)+ ( pe2 * cos(dc)))- ((Pi * d0)+ ( pe2
      * cos(d0)))
39 A2=((Pi * dm)+ ( pe3 * cos(dm)))- ((Pi * dc)+ ( pe3
      * cos(dc)))
40
41 if abs(A1)<abs(A2) then
42     mprintf("STABLE system\n\n")
43 else
44     mprintf("UNSTABLE  system\n\n")
45 end
46
47
48 E=10
49 x=2
50 e=1e-3
51 a=Pi
52 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

`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

```
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
```

```
16 x=.811
17 e=1e-3
18 a=Pe2
19 b=1
20 c=A1 -((Pe2 * d2)+ ( 1 * cos(d2)))
21
22 while (E>e)
23 //for (i=1:4)
24     f=(a*x) +  (b * cos(x)) + c
25     df=a - (b*sin(x))
26     x1=x-(f/df)
27     disp(x1,f,df)
28     E=abs(x1-x)
29     x=x1
30 end
31 d3=x1 * 180/%pi
32
33 mprintf("\ndelta 3 = %.1f deg",d3)
```

Scilab code Exa 13.11 inertia constant of 2 generators in parallel

inertia constant of 2 generators in parallel

```
1 clear
2 clc
3
4 G1=50
5 H1=8
6
7 G2=100
8 H2=4
9 Gb=100
10
11 Ha=(H1*G1/Gb) + (H2*G2/Gb)
12 mprintf("\n(a)Ha= %d MJ/MVA", Ha)
```

```
13
14 Hb=Ha*2
15 mprintf("\n(b)Hb= %d MJ/MVA", Hb)
16
17 He= (Ha*Hb)/(Ha+Hb)
18 mprintf("\n(c)He= %.3f MJ/MVA", He)
```

Scilab code Exa 13.12 find frequency deviation in case of delay in opening steam valve

```
find frequency deviation in case of delay in opening steam valve
```

```
1  clear
2  clc
3
4  G=100
5  f=50
6  H=5
7  dL=50
8  t=.6
9
10
11 J = G*H*1e3;
12 dJ=dL*1e3*t
13 f2=sqrt((J-dJ)/J)*f
14 fd=(f-f2)/f;
15 mprintf("Freq deviation = %.3f percent", fd*1e2)
```

Scilab code Exa 13.13 find critical clearing angle during prefault fault and post fault conditions

```
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

point by point solution of swing equation

```
clear
clc
clf

Sb = 50;
S=50;
V =1;
Xd = 0.2;
X1 =0.4;
X2 = 0.4;
H = 2.7;
E=1.05 ;
G=1;

```

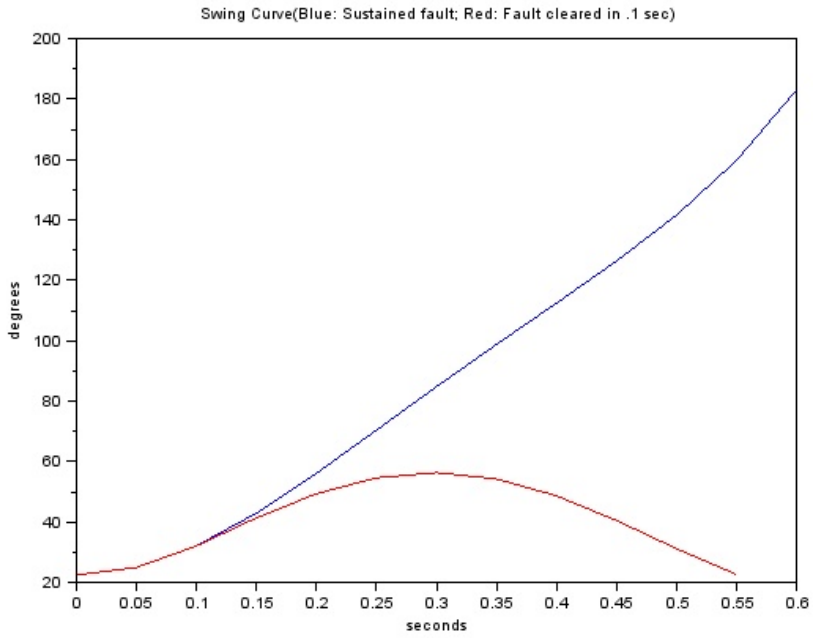

Figure 13.1: point by point solution of swing equation

```
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))
```

```
53          m_Pe(i)=m_Pm(i) * m_sind(i)
54          m_Pa(i)=(1 - m_Pe(i))
55          m_cPe(i)=c_1 * m_Pa(i)
56      end
57      end
58
59  end
60  res1(:,1)=m_t(:)
61  res1(:,2)=m_Pm(:)
62  res1(:,3)=m_sind(:)
63  res1(:,4)=m_Pe(:)
64  res1(:,5)=m_Pa(:)
65  res1(:,6)=m_cPe(:)
66  res1(:,7)=m_dd(:)
67  res1(:,8)=m_d(:)
68  res1=round(res1*1000)/1000
69  i=1
70  head=['' '' 't'  '' '' 'Pm'  '' '' 'sin d'  '' '' '
       Pe'  '' ''Pa'  '' ''8.33Pa'  '' ''d delta' '
       delta']
71  disp(res1, head, "(a)")
72  plot(m_t, m_d)
73  title('Swing Curve(Blue: Sustained fault; Red: Fault
        cleared in .1 sec)');
74  xlabel('seconds');
75  ylabel('degrees');
76
77
78  //(b)
79  while i<15
80      if i==1 then
81          m_t2(i)=0;
82          m_Pm2(i)=Pe0
83          m_sind2(i)=sind(d0)
84          m_Pe2(i)=S/Sb
85          m_Pa2(i)=0
86          m_cPe2(i)=c_1 * m_Pa2(i)
87          m_dd2(i)=0
```

```
88          m_d2(i)=d0
89      else if i==2 then
90              m_t2(i)=0;
91              m_Pm2(i)=pe1
92              m_d2(i)=d0
93              m_sind2(i)=sind(m_d2(i))
94              m_Pe2(i)=m_sind2(i)*m_Pm2(i)
95              m_Pa2(i)=(1 - m_Pe2(i) + m_Pa2(i-1))/2
96              m_cPe2(i)=c_1 * m_Pa2(i)
97              m_dd2(i)=0
98          else
99              m_t2(i)=m_t2(i-1) +dt;
100             if m_t2(i) == .1 then
101                 m_Pm2(i)=pe1
102                 m_dd2(i)=m_dd2(i-1) + m_cPe2(i-1)
103                 m_d2(i)=m_d2(i-1)+m_dd2(i)
104                 m_sind2(i)=sind(m_d2(i))
105                 m_Pe2(i)=m_Pm2(i) * m_sind2(i)
106                 m_Pa2(i)=(1 - m_Pe2(i))
107                 m_cPe2(i)=c_1 * m_Pa2(i)
108
109                 i=i+1
110                 m_t2(i)=m_t2(i-1)
111                 m_Pm2(i)=pe2
112                 m_dd2(i)=m_dd2(i-1)
113                 m_d2(i)=m_d2(i-1)
114                 m_sind2(i)=sind(m_d2(i))
115                 m_Pe2(i)=m_Pm2(i) * m_sind2(i)
116                 m_Pa2(i)=(1 - m_Pe2(i) + m_Pa2(i-1))
                       /2
117                 m_cPe2(i)=c_1 * m_Pa2(i)
118             else
119                 m_Pm2(i)=m_Pm2(i-1)
120                 m_dd2(i)=m_dd2(i-1) + m_cPe2(i-1)
121                 m_d2(i)=m_d2(i-1)+m_dd2(i)
122                 m_sind2(i)=sind(m_d2(i))
123                 m_Pe2(i)=m_Pm2(i) * m_sind2(i)
124                 m_Pa2(i)=(1 - m_Pe2(i))
```

```
125                     m_cPe2(i)=c_1 * m_Pa2(i)
126                 end
127             end
128         end
129         i=i+1
130 end
131 res2(:,1)=m_t2(:)
132 res2(:,2)=m_Pm2(:)
133 res2(:,3)=m_sind2(:)
134 res2(:,4)=m_Pe2(:)
135 res2(:,5)=m_Pa2(:)
136 res2(:,6)=m_cPe2(:)
137 res2(:,7)=m_dd2(:)
138 res2(:,8)=m_d2(:)
139 res2=round(res2*1000)/1000
140 disp(res2,head, "(b)")
141
142 plot(m_t2, m_d2,'r')
143 //(c)
144 D0=d0 * %pi/180
145 Pi=1
146 Dm=%pi -  asin(Pi/pe2)
147
148 dcc=acosd(((Pi * (Dm-D0))-(pe1*cos(D0))+(pe2*cos(Dm)
    ))/(pe2 -pe1))
149 tcc=.395
150 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

find maximum load that can be supplied by generator

```
1 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

```
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

```
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

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
```

```
21 H2=J2/G2
22 M2=J2/(180*f)
23
24 mprintf("\n\nmachine 2 \nInertia const= %.2f MJ/MVA,
       Momentum= %.5f MJ-s/elec deg",H2,M2)
25
26 M=(M1*M2)/(M1+M2)
27 Gb=100
28 H=M*180 * f /Gb
29 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

```
find critical clearing angle and critical clearing time in 3 phase fault condition
```

```
1  clear
2  clc
3
4  Pm=2
5  Pi=1
6  H=6
7  G=1
8  f=50
9  p=Pi/Pm
10 M=G*H/(%pi*f)
11 d0=asin(p)
12
13 dcc=acos(((p*(%pi - (2*d0)))- (Pi*cos(d0)))/(Pm-Pi))
14 mprintf("Critical Clearing angle = %.4f rad\n\n",
      dcc)
15
16 tcc=sqrt(2*M*(dcc-d0)/Pi)
```

```
17 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

```
finding acceleration torque and change in torque angle due to losses
```

```
1  clear
2  clc
3
4  f=50
5  G=20
6  V=13.2
7  H=9
8  nP=4
9
10 J=G*H
11 mprintf("\n(a)Stored Energy = %.0f MJ", J)
12 disp("The unit is incorrectly mentioned as ''mJ'' in
       the textbook.");
13
14 Pi= 25*.735
15 P=15
16 Pa=Pi-P
17 M=G*H/(180*f)
18 a=Pa/M
19 mprintf("\n(b)Acceleration = %.2f elec deg/sec sq",a
       )
20
21
22 c=15
23 t=c/f
24 dd=.5*a*t*t
25 mprintf("\n(c)change in angle = %.2f deg",dd )
```

```
26
27 A=a * 60 / (180*nP)
28 Ns=120*f/nP
29 N=Ns+(A*t)
30 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

find voltage at load points in single feeded dc feeder

```
1  clear
2  clc
3
4  I1=100
5  I2=150
6  I3=200
7
8  l1=150
9  l2=100
10 l3=100
11
12 r=.1/1000
13
14 Va=200
15
16 rac=l1*r;
17 rcd=l2*r
18 rbd=l3*r
19
```

```
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

```
find voltage at load points in addition to distributed load in single feeded dc fe
```

```
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
```

```
20
21 Iac= I1+I2+I3 + U*(l1+l2+l3)
22 dvc=(Iac - (.5* U * l1))*rac;
23 Vc=Va-dvc
24
25 Icd= I2+I3 + U*(l2+l3)
26 dvd=(Icd - (.5* U * l2))*rcd ;
27 Vd=Vc-dvd
28
29 Idb= I3 + U*(l3)
30 dvb=(Idb - (.5* U * l3))*rbd ;
31 Vb=Vd-dvb
32
33 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

```
find voltage at load points in doubly feeded dc feeder
```

```
1  clear;
2  clc
3
4  r1=.02
5  r2=.05
6  r3=.03
7
8  Ic1=100
9  Id1=180
10
11 Va=255;
12 Vb=250
13
14 dV=abs(Va-Vb)
15 Ia=(dV+(r1*0)+(r2*Ic1)+(r3*(Id1+Ic1)))/(r1+r2+r3)
```

```
16
17 Ib=-(Ia-(Ic1+Id1))
18
19 Vc=Va-Ia*r1
20 Vd=Vc-((Ia-Ic1)*r2)
21
22 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

```
find voltage at load points in addition to distributed load in doubly feeded dc fe
```

```
1 clear
2 clc
3
4 l=300
5 l1=120
6 l3=120
7 l2=l-l1-l3
8 U=.25
9 Ic1=40
10 Id1=60
11
12 r=.1/100
13 r1=l1*r;
14 r2=l2*r
15 r3=l3*r
16
17 Va=300
18 Vb=300
19
20 dV=abs(Va-Vb)
```

```
21 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)
22 I=Ic1+Id1+(U*l)
23 Ib=I-Ia
24
25 Vc=Va-(Ia-.5*U*l1)*r1
26 Vd=Vb-((Ib-.5*U*l3)*r3)
27
28 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

`voltage drop in singly feeded ac feeder with concentrated load`

```
1  clear
2  clc
3
4  l=400
5  l1=100
6  l2=250
7  l3=400
8  r=.25/1000
9
10 I1=100
11 I2=120
12 I3=80
13
14 V=240
15
16 dv=r*((l1*I1)+(l2*I2)+(l3*I3))
17 Ve=V-dv
18
19 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

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
```

```
30
31 mprintf("\nVoltage at end is %.2f V, drop = %.2f V",
       Ve, dv)
```

Scilab code Exa 16.7 currents in a 3 phase ac circuit

currents in a 3 phase ac circuit

```
1  clear
2  clc
3
4  V=240
5
6  P1=50e3
7  P2=50e3
8  P3=200e3
9  Pm=500e3
10 pfm=.8
11
12
13 I1=P1/V
14 I2=P2/V
15 I3=P3/V
16 Im=Pm/(3*V*pfm)
17
18 i1=fix(I1*exp(%i *0*%pi/3)*100)/100
19 i2=fix((fix(I2*100)/100)*exp(%i *-2*%pi/3)*1000)
      /1000
20 i3=fix(I3*exp(%i *2*%pi/3)*100)/100
21 in=abs(round((i1+i2+i3)*100)/100)
22
23 iR=sqrt((I1)^2+(Im)^2+(2*I1*Im*pfm))
24 iY=sqrt((I2)^2+(Im)^2+(2*I2*Im*pfm))
25 iB=sqrt((I3)^2+(Im)^2+(2*I3*Im*pfm))
26
```

```
27 mprintf("\nIR= %.0f A", iR)
28 mprintf("\nIY= %.0f A", iY)
29 mprintf("\nIB= %.0f A", iB)
30 mprintf("\nIn= %.2f A", in)
```

Scilab code Exa 16.8 voltage drop at the end of one phase in unbalanced 3 phase network

voltage drop at the end of one phase in unbalanced 3 phase network

```
1  clear
2  clc
3
4  V=230
5
6  I1=80
7  I2=70
8  I3=50
9
10 pf1=.8
11 pf2=.9
12 pf3=1
13
14 phi1=acos(pf1)
15 phi2=acos(pf2)
16 phi3=acos(pf3)
17
18 i1=I1*exp(%i *0*%pi/3) * exp(%i * -phi1)
19 i2=I2*exp(%i*-2*%pi/3) * exp(%i * -phi2)
20 i3=I3*exp(%i *2*%pi/3) * exp(%i * -phi3)
21
22 r=.2
23
24 in=i1+i2+i3
25
```

```
26 dvR=r*i1
27 dvN=r*in
28
29 VR=V+dvR+dvN
30
31 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

find supply voltage and phase angle between sending end and receiving end

```
1  clear
2  clc
3
4  Vb=240
5
6  Ib=100* exp(%i *-1* acos (.8))
7  Ia=100* exp(%i *-1* acos (.6))
8
9  z=complex(.2, .3)
10
11 Va=round((Vb + (Ib * z/2)))
12 Isa = (Ia * exp(%i * 1 * atan(imag(Va)/real(Va)))) +
       Ib
13 //Isa=Isa*%i
14
15 Vs=Va + (Isa * z/2)
16 Vs=round(Vs*100)/100
17 vs=abs(Vs)
18 phi=atand(imag(Vs)/real(Vs))
19
20 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

```
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

find point of minimum in a line

```
clear
clc

L=1200
L1=900
L2=600

r=1.5
```

```
9
10 x=(L1*L + L*L*r/2)/(L1 + L2 + (r*2*L/2))
11 y=L-x
12 mprintf("\nCurrent in CB= %.0f a A", x)
13 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

```
voltage at far end in a double ac conductor with concentrated load
```

```
1  clear
2  clc
3
4  V=250
5
6  L=[
7  150 200 280 320 390 450 500
8  20 40 35 25 10 20 30
9  ]
10
11 r=.1/(2*500)
12
13 D=L(1,:)
14 I=L(2,:)'
15
16 dv=2*r*D*I
17
18 Ve=V-dv
19
20 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

`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

`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

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

find optimum cross section of cables for comsumers at different distances

```
clear
```

```
2  clc
3
4  //(a)
5  V=230
6  df=5/100
7
8  I1=20
9  I2=10
10 L1=300
11 L2=200
12 L=500
13 I=I1+I2
14
15 rho=.0286
16 T=20
17 dT=30
18 a=.004
19 rho1=rho*(1+(dT*a))
20 rho1=round(rho1*1000)/1000
21
22 A=1
23 A1=I1/I
24 A2=I2/I
25
26 A=rho1*2*((L*round(I/A))+(L1*round(I1/A1)))/(df * V)
27 A=round(A*100)/100
28
29 A1=A*A1
30 A2=A*A2
31
32 mprintf("\n(a)CROSSECTIONS: SC= %.2f mm sq, CA= %.2f
       mm sq, CB= %.2f mm sq",A, A1, A2 )
33
34 R=2*rho1 * (L/A)
35 R1=2*rho1 * (L1/A1)
36 R2=2*rho1 * (L2/A2)
37
38 P=(I*I*R)+(I1*I1*R1)+(I2*I2*R2)
```

```
39
40 PL=P*100/((V*I1)+(V*I2))
41 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

```
voltage at far end in a double ac conductor singly fed with uniform and concentrat
```

```
1  clear
2  clc
3
4  I1=100
5  I2=50
6  I3=50
7  I4=100
8  I5=0
9
10 L=500
11 l1=50
12 l2=100
13 l3=100
14 l4=150
15 l5=L-l1-l2-l3-l4
16
17 R=.075
18 r=R/L
19 U=1
20
21 Va=230
22
23 r1=l1*r;
24 r2=l2*r
25 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

voltage at far end in a double ac conductor with concentrated load

```
clear
clc

Vc=220
```

```
5
6 Z=2*complex(.15, .2)
7 Z1=.5*Z
8 Z2=.5*Z
9
10 Ib= 60 * exp (%i * -acos (.9))
11 Ic= 100 * exp (%i * -acos (.8))
12
13 Vb=Vc + (Ic * Z2)
14 Iab=Ib + Ic
15
16 Va=Vb + (Iab * Z1)
17 Vs=abs(Va)
18
19 Is= abs(Iab)
20 pf=cosd(abs(atand(imag(Va)/real(Va))+abs(atand(imag(
      Iab)/real(Iab)))))
21
22 mprintf("\nsending end voltage  = %.1f V", Vs)
23 mprintf("\nsending end current  = %.2f A", Is)
24 mprintf("\nsending end pf  = %.3f", pf)
```

Scilab code Exa 16.19 find most economical center of distribution

find most economical center of distribution

```
1 clear
2 clc
3
4 ab=1000
5 bc=900
6 ac=600
7
8 A=500
9 B=750
```

```
10 C=450
11
12 theta=acos((ab^2 + ac^2 - bc^2)/(2*ac*ab))
13 am=ac * cos(theta)
14 cm=ac * sin(theta)
15
16 x= ((B*ab)+(C*am))/(A+B+C)
17 y= ((C*cm))/(A+B+C)
18
19 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

find optimum cross section for double dc line doubly fed concentrated load

```
1  clear
2  clc
3
4  rho=1/58
5
6  l1=90
7  l2=90
8  l3=100
9  l4=80
10 l5=90
11 l6=90
12 l7=90
13
14 I1=50
15 I2=40
16 I3=20
17 I4=30
18 I5=45
19 I6=50
```

```
20
21 I=I1+I2+I3+I4+I5+I6
22 L=l1+l2+l3+l4+l5+l6+l7
23 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
24
25 Ib=I-Ia
26 V=230*2
27 V1=440
28
29 dv=V-V1
30
31 r=dv/((Ia*(l1+l2+l3+l4))-((I1*l2)+((I1+I2)*l3)+((I1+
      I2+I3)*l4)))
32
33 a=rho*2/r
34
35 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

```
find currents in a hexagon shaped concentrated loads in 2 line dc ring main
```

```
1 clear
2 clc
3
4 r1=.08
5 r2=.1
6 r3=.12
7 r4=.14
8 r5=.09
9 r6=.16
10
```

```
11 I1=20
12 I2=50
13 I3=25
14 I4=40
15 I5=30
16
17 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
      )
18
19 iab=x
20 ibc=iab-I1
21 icd=ibc-I2
22 ide=icd-I3
23 ief=ide-I4
24 ifa=ief-I5
25
26 iab=round(iab*10)/10
27 ibc=round(ibc*10)/10
28 icd=round(icd*10)/10
29 ide=round(ide*10)/10
30 ief=round(ief*10)/10
31 ifa=round(ifa*10)/10
32
33
34 mprintf("\n(a) Current A to B in Ampere = %.1f", iab
      )
35 mprintf("\n Current B to C in Ampere = %.1f", ibc)
36 mprintf("\n Current C to D in Ampere = %.1f", icd)
37 mprintf("\n Current D to E in Ampere = %.1f", ide)
38 mprintf("\n Current E to F in Ampere = %.1f", ief)
39 mprintf("\n Current F to A in Ampere = %.1f", ifa)
40
41
42
43 Vb=230+((r6*ifa)+(r5*ief)+(r4*ide))
44 mprintf("\n Minimum voltage at B, in Volts = %.3f\n\
      n", Vb)
```

```
45
46 //(b)
47 r7=.1
48
49
50 B=[
51 (r4+r5+r6) -r7
52 (r1+r2+r3) (r7+r1+r3+r2)
53 ]
54
55 C=[
56 ((r5*I5)+(r4*(I5+I4)))
57 ((r2*(I5+I2+I3+I4))+(r1*(I1+I2+I3+I4+I5))+(r3*(I1+I2
      +I3)))
58 ]
59
60 A= inv(B)*C
61 x=A(1)
62 y=A(2)
63 iab=I1+I2+I3+I4+I5 - x-y
64 ibc=iab-I1
65 icd=ibc-I2
66 iaf=x
67 ife=iaf-I5
68 ied=ife-I4
69 iad=y
70
71 mprintf("\n(b) Current A to B in Ampere = %.1f", iab
      )
72 mprintf("\n Current B to C in Ampere = %.1f", ibc)
73 mprintf("\n Current C to D in Ampere = %.1f", icd)
74 mprintf("\n Current E to D in Ampere = %.1f", ied)
75 mprintf("\n Current F to E in Ampere = %.1f", ife)
76 mprintf("\n Current A to F in Ampere = %.1f", iaf)
77 mprintf("\n Current A to D in Ampere = %.1f", iad)
78
79 Ve=230-((r6*iaf)+(r5*ife))
80 mprintf("\n Minimum voltage at E, in Volts = %.3f",
```

```
Ve)
```

Scilab code Exa 16.22 kelvins law

```
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

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
```

```
36
37 P=real (Ia * V )
38 PLa=round(.05*P)
39
40 l1=40
41 l2=35
42 l3=30
43
44 PL1=2 * (round(abs(Ia)*100)/100)^2 * rho * l1
45 PL2=2 * (fix(abs(Ib)*100)/100)^2 * rho * l2
46 PL3=2 * (round(abs(Ic)*100)/100)^2 * rho * l3
47
48 PL1=fix(PL1/100)*100
49 PL2=round(PL2/100)*100
50 PL3=round(PL3/10)*10
51
52
53 a=(PLa/(PL1+PL2+PL3))^-1
54
55 mprintf("a=%.2f sq mm", a)
56 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

```
find loss factor load factor annual load loss and annual cost of lost energy
```

```
1 clear
2 clc
3
4 Pp=3e3;
5 Et=1e7
6 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

```
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
```

```
15 xt2=.08
16 n2=vt2b/vt2a
17
18 st3=05
19 vt3a=11
20 vt3b=.415
21 xt3=.06
22 n3=vt3b/vt3a
23
24 Sm=5
25 pfm=.8
26
27 Sl=1
28 pfl=.8
29
30 Sb=10
31 Vb1=66
32 Vb2=Vb1*n1
33 Vb3=Vb2*n2
34 Vb4=Vb2*n3
35
36 Xt2=xt2* (Sb/st2)
37 Xt3=xt3* (Sb/st3)
38
39 Il=(Sl/Sb) * exp(%i * -acos(pfl))
40 Im=(Sm/Sb) * exp(%i * -acos(pfm))
41
42 It=Im+Il
43 Vt= 1- (It*(%i*(xt1+xl)))- (Il *%i* Xt3)
44
45 mprintf("\n(a) Voltage at domestic load is %.3f pu",
       abs(Vt))
46
47 Im=-1 * %i * 5 * Sm / Sb
48 It=Il +Im
49 Vt= 1- (It*(%i*(xt1+xl)))- (Il *%i* Xt3)
50
51 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

```
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

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

rating of switched bank and fixed bank capacitors

```
clear
clc

```

```
4  t1=15
5  P1=1.3
6  pf1=.75
7
8  t2=9
9  P2=.4
10 pf2=.8
11
12 pfd=.95
13
14 kvar1i=P1*1e3*tan (acos(pf1))
15 kvar1f=P1*1e3*tan (acos(pfd))
16 kvarr1=kvar1i-kvar1f
17
18 kvar2i=P2*1e3*tan (acos(pf2))
19 kvar2f=P2*1e3*tan (acos(pfd))
20 kvarr2=kvar2i-kvar2f
21
22 SBC=abs(kvarr2-kvarr1)
23 FBC=min(kvarr2,kvarr1)
24
25 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

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

```

```
53 E1= round((1+((Z*Y)/2))*1000)/1000
54 E2=round((Y*(1+((Z*Y)/4)))*1e7)/1e7
55 Vr=V*1e3/sqrt(3)
56 pf=.9
57 Ir=Ir * exp(%i * -acos(pf))
58 Vs=(Vr * E1) + (Ir*Z)
59 Is=(Vr *Y* E2) + (Ir*E1)
60 //Error in answer (Ps) is due to mutiple rounding
     off in a step in the textbook (Is)
61 pfs=cos(atan(imag(Vs)/real(Vs))+atan(imag(Is)/real(
     Is)))
62 Ps=round(real(3*Vs*Is))/1000000
63 Ps=105.07
64 pfs=round(pfs*100)/100
65 eff=P*.1/Ps
66 Vr0=abs(Vs)/abs(E1)
67 VR=(Vr0-abs(Vr))/abs(Vr)
68 mprintf("\n(d)\nline efficiency= %.2f percent,
     Voltage regulation= %.2f percent",eff, VR*100)
69
70 //(e)
71 p=74;
72 t=50
73 d=3.86 * p/(273+50)
74 m0=.84
75 Vd=(3*1e6/sqrt(2)) * r *1e-3 * d * m0 * log(Deq/(r*1
     e-3))
76 ratio=V*1e3/(Vd*sqrt(3))
77 F=.05
78
79 corona=3* 21 * 1e-6 * 50 *(V/(sqrt(3)))  *(V/(sqrt
     (3))) * F/(log10(Deq*1e3/r) *log10(Deq*1e3/r) )
80 corona=round(corona*100)/100
81 corona=corona * Len
82 mprintf("\n(e)Corona loss =%.1f KW",corona)
83
84 //(f)
85 tphi1=tan(acos(pf))
```

```
86 tphi2=tan(acos(pfs))
87 Q1=P*1e-3 *tphi1
88 Q2=P*1e-3 *tphi2
89 Cap=Q1-Q2
90 mprintf("\n(f)capacity of capacitor = %.2f MVAR
      leading",Cap)
91 //(g)
92 Vr=V*1e3/sqrt(3)
93 Vr=round(Vr)
94 Ir=(P*1e3/(3*Vr*pfs) ) * exp(%i * -acos(pfs))
95 Vs=(Vr * E1) + (Ir*Z)
96 Is=(Vr *Y* E2) + (Ir*E1)
97 //Error in answer (Ps) is due to mutiple rounding
      off in a step in the textbook (Is)
98 pfs=cos(atan(imag(Vs)/real(Vs))+atan(imag(Is)/real(
      Is)))
99 Ps=round(real(3*Vs*Is))/1000000
100 Ps=104.74
101 pfs=round(pfs*100)/100
102 eff=P*.1/Ps
103 Vr0=abs(Vs)/abs(E1)
104 VR=(Vr0-abs(Vr))/abs(Vr)
105 mprintf("\n(g)\nline efficiency= %.1f percent,
      Voltage regulation= %.2f percent",eff, VR*100)
106
107 //(h)
108 A=37 * %pi * (dAl/1000)^2 /4
109 E=91.4 *1e9
110
111 alpha=18.44 *1e-6
112 w=14.64
113 Fw=378 * dia * 1e-3
114 Fw=round(Fw*100)/100
115 Ft1=sqrt(w^2 + Fw^2)
116 T1=135.5*1e3/2.5
117 Ft2=w
118 Temp21=5
119 Temp22=30
```

```
120
121 c_1=1
122 c_2=T1 -(alpha * A * E * (Temp22-Temp21)) - A*E*Ft1
      ^2 * span^2 /(24*T1^2)
123 c_3=0
124 c_4=A*E*Ft2^2 * span^2 /24
125 pol=poly([-c_4 -c_3 -c_2 c_1], "xx", "c")
126 T2s=roots(pol)
127
128
129 T2=T2s(1)
130 Sag1= w * span *span / (8 * T2)
131 Sag2= round(Ft1*100)*span *span / (800 * T1)
132 VS=Sag2 * cos (atan(Fw/w))
133 mprintf("\n(h)Tension = %.0f N, Sag under erection =
      %.2f m , vertical sag due to bad weather = %.2f
      m", T2, Sag1, VS)
134
135 //(i)
136 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

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

`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

```
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

finding sending and receiving end reactive power

```
1  clear;
2  clc;
3
4  V=400e3;
5  X=96;
6  Y=.001 * exp(%i * 90/180*%pi);
7  Sb=500e6;
8  Pr=2;
9
10 Z=complex(0,X);
11 Zb=V*V/Sb;
12 A=1+(Z*Y/2);
13 B=Z/Zb;
14
15
16 Vs=1;
17 Vr=.95;
18 d1=asin(Pr*abs(B)/(Vs*Vr));
```

```
19 Qr1=((Vs*Vr)/abs(B)) * cos(d1) - (abs(A)* Vr*Vr/abs(
   B))
20 Qs1= (abs(A)* Vs*Vs/abs(B)) - ((Vs*Vr)/abs(B)) * cos
   (d1)
21
22 mprintf("\n(a) Qr= %.3f MVAR, Qs=%.3f MVAR", Qr1,
   Qs1);
23
24 Vs=1;
25 Vr=.9;
26 d2=asin(Pr*abs(B)/(Vs*Vr));
27 Qr2=((Vs*Vr)/abs(B)) * cos(d2) - (abs(A)* Vr*Vr/abs(
   B))
28 Qs2= (abs(A)* Vs*Vs/abs(B)) - ((Vs*Vr)/abs(B)) * cos
   (d2)
29
30 mprintf("\n(b) Qr= %.3f MVAR, Qs=%.3f MVAR", Qr2,
   Qs2);
31
32 Vs=1;
33 Vr=1;
34 d3=asin(Pr*abs(B)/(Vs*Vr));
35 Qr3=((Vs*Vr)/abs(B)) * cos(d3) - (abs(A)* Vr*Vr/abs(
   B))
36 Qs3= (abs(A)* Vs*Vs/abs(B)) - ((Vs*Vr)/abs(B)) * cos
   (d3)
37
38 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

compensating value of capacitor

```
1 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

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;
```

```
14 pfa=acos(pf);
15 Qr=Pr*tan(pfa);
16
17 Vs=complex(Vr + (X * Qr / Vr) , (X * Pr / Vr));
18 V=abs(Vs);
19
20 mprintf("Vr = %.2f ang (%.1f) deg pu",V, fix(atand(
     imag(Vs)/real(Vs))*10)/10)
```

Scilab code Exa 19.5 capacity of SVS

capacity of SVS

```
1 clear;
2 clc;
3
4 vf=3;
5 VA=8000;
6
7 dQ=vf/100 * VA
8
9 mprintf("capacity of SVS= +- %d MVAR", dQ);
```

Scilab code Exa 19.6 voltage and pf of bus before compensation

voltage and pf of bus before compensation

```
1 clear;
2 clc;
3
4 V=220e3;
5 Zl=complex(.8, .2);
```

```
6
7  Xline=.2;
8  Xt=.05;
9  Sb=100e6;
10 Vb=220e3;
11 v=V/Vb;
12
13 X=Xline+ Xt;
14 I=conj(Zl/v)
15 phi1=atand(imag(I)/real(I))
16
17 Vbus=1+ I * X*exp(%i * %pi/2)
18
19 phi2=atand(imag(Vbus)/real(Vbus))
20
21 vbus=abs(Vbus)
22 vbus=round(vbus *1000)/1000
23 vbus=vbus*Vb*1e-3;
24 pf=cosd(-phi1+phi2)
25
26 mprintf("Voltage at bus = %.2f Kv, pf= %.3f lagging"
     , vbus, pf)
```

Scilab code Exa 19.7 voltage and pf of bus after compensation

voltage and pf of bus after compensation

```
1  clear;
2  clc;
3  //the data used is from Ex 19.6, not 19.5 as
      incorrectly mentioned in statement
4  V=220e3;
5  Zl=complex(.8,0);
6
7  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

```
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

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))


```

```
27 mprintf("\n(b)");
28
29 mprintf("\nA =D= %.3f ang (%.2f) deg",fix(abs(A2)
   *1000)/1000, round(atand(imag(A2)/real(A2))*100)
   /100)
30 mprintf("\nB = %.2f ang (%.2f) deg ohm",abs(B2),
   round(atand(imag(B2)/real(B2))*100)/100)
31 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

voltage regulation of compensated line

```
1  clear;
2  clc;
3
4  d=500;
5  z=complex(.105, .3768);
6  y=complex(0, 2.822e-6);
7  Z=z*d;
8  Y=y*d;
9  YZ=Y*Z;
10
11 A=1+(YZ/2)+((YZ)^2/24);
12 B=Z * (1+(YZ/6)+((YZ)^2/120));
13 C=Y * (1+(YZ/6)+((YZ)^2/120));
14 D=A;
15
16 Ys=-.6*Y;
17
18 As=1;
19 Ds=1;
20 Bs=0;
21 Cs=Ys;
```

```
22
23 Anew=A + (B*Cs);
24
25 Vr=round(220e3/sqrt(3));
26 Pr=40e6;
27 pf=.9;
28 pfa=-1*acos(pf);
29 Irm=Pr/(3*Vr);
30 Ir=complex(Irm *pf, Irm * sin(pfa));
31
32 Vs=(A*Vr)+(B*Ir);
33 Vr0=abs(Vs)/(round(abs(Anew)*100)/100);
34
35 VRc= (Vr0-Vr)*100/Vr
36 mprintf("\n(a) Voltage Regulation = %.2f percent",
     VRc);
37
38 Vr02=abs(Vs)/abs(A);
39
40 VRc2= (Vr02-Vr)*100/Vr
41 mprintf("\n(b) Voltage Regulation(uncompensated) = %
     .2f percent",VRc2);
```

Scilab code Exa 19.11 find var injection to bring voltage to original value

find var injection to bring voltage to original value

```
1 clear;
2 clc;
3
4 v1=220;
5 v2=132
6 vb1=220;
7 n=132/220
8 vb2=vb1*n
```

```
9
10 Sb=200;
11 Zb= vb2 *vb2/Sb;
12
13 x1=75;
14 x2=70;
15 x3=90
16
17 X1=x1/Zb;
18 X2=x2/Zb;
19 X3=x3/Zb;
20 X1=fix(X1*100)/100
21 X2=fix(X2*100)/100
22 X3=fix(X3*100)/100
23
24
25
26 Xt1=.08
27 Xt2=.08
28
29 X=((Xt1+X1)^-1 +(Xt2+X2)^-1 +(X3)^-1 )^-1;
30 X=fix(X*1000)/1000
31
32 dV=4
33 dS= Sb/X;
34 dQ=round(dS*1000/vb2)*1e-3 * dV
35
36 mprintf("\n(a)\n X1= %.2f pu; X2= %.2f pu; X3= %.2f
     pu", X1, X2, X3)
37 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

find failure rate of system down time per outage annual outage

```
1  clear;
2  clc;
3
4  l=.4;
5  Ft=.8;
6  Fd=8.2;
7  Tt=6;
8  Td=5;
9
10 Rf=Ft + (Fd*l);
11 mprintf("\n(a)Failure Rate = %.2f outages/year",Rf);
12
13 t=((Ft*Tt)+(Fd*Td*l))/Rf;
14 mprintf("\n(b)Down Time = %.3f Hours per outage",t)
     ;;
15
```

```
16 T=t*Rf;
17 mprintf("\n(c)Total Outage Time = %.1f Hours per
      year",T);;
```

Scilab code Exa 20.2 find reliability of series connected elements

find reliability of series connected elements

```
1 clear;
2 clc;
3
4 r1=.95;
5 r2=.92;
6 r3=.98;
7 r4=.88
8
9 R=r1*r2*r3*r4;
10 mprintf("\nNet system reliability = %.4f ",R);
```

Scilab code Exa 20.3 find reliability of parallel connected elements

find reliability of parallel connected elements

```
1 clear;
2 clc;
3
4 r1=.95;
5 r2=.92;
6 r3=.98;
7 r4=.88
8
9 R=r1*r2*r3*r4;
```

```
10
11 Q=(1-R)*(1-R);
12 Rs=1-Q;
13 mprintf("\nNet system reliability = %.4f ",Rs);
```

Scilab code Exa 20.4 find reliability of series connected elements and MTTF

```
find reliability of series connected elements and MTTF
```

```
1  clear;
2  clc;
3
4  na=3;
5  nb=2;
6  nc=1;
7  Fa=3e-3;
8  Fb=2e-3;
9  Fc=4e-3;
10 t=20;
11
12
13 F=(Fa*na)+(Fb*nb)+(Fc*nc);
14 R=exp(-1*F*t);
15 MTTF=1/F;
16 mprintf("\nReliability = %.5f \nMTTF = %.3f HOURS",R
       ,fix(MTTF*1000)/1000);
```

Scilab code Exa 20.5 find reliability of parallel connected elements and MTTF

```
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

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

find improved reliability

```
clear;
clc;
```

```
3
4 r1=.8;
5 r2=.86;
6 r3=.92;
7
8 Q1=(1-r1)^2;
9 R1=1-Q1;
10
11 Q2=(1-r2)^2;
12 R2=1-Q2;
13
14 Q3=(1-r3)^2;
15 R3=1-Q3;
16
17 R=R1*R2*R3;
18
19
20 mprintf("\nNet system reliability = %.3f ",R);
```

Scilab code Exa 20.8 find number of components to achieve desired reliability

find number of components to achieve desired reliability

```
1 clear;
2 clc;
3
4 r=.3
5 R=.85;
6 n=log(1-R)/log(1-r);
7 N=round(n);
8 if(N<n)
9     N=N+1;
10 mprintf("\nComponents required = %d",N)
11 Rs=1-((1-.3)^N);
```

```
12 mprintf("\nNet System Reliability = %.3f",Rs)
```

Scilab code Exa 20.9 find failure rate of system down time per outage annual outage

find failure rate of system down time per outage annual outage

```
1  clear;
2  clc;
3
4  Oat1=0.1;
5  Oacb=0.15;
6  Oafd=1.2;
7  Oat2=0.3;
8
9  Obt1=0.1;
10 Obcb=0.15;
11 Obfd=5.2;
12 Obt2=0.3;
13
14 Oct1=0.5;
15 Occb=0.4;
16 Ocfd=2;
17 Oct2=0.6;
18
19 Rat1=6;
20 Racb=4;
21 Rafd=5;
22 Rat2=5;
23
24 Rct1=8;
25 Rccb=5;
26 Rcfd=3;
27 Rct2=4;
28
```

```
29 N=120;
30 S=3;
31
32 Of=((N/(N+S))*Oafd)+((S/(N+S))*Obfd);
33 Oafe=Oat1+Oacb+Of+Oat2;
34 Ocfe=Oct1+Occb+Ocfd+Oct2;
35
36 Rae=((Oat1*Rat1)+(Oacb*Racb)+(Of*Rafd)+(Oat2*Rat2))/
      Oafe;
37 Rce=((Oct1*Rct1)+(Occb*Rccb)+(Ocfd*Rcfd)+(Oct2*Rct2)
      )/Ocfe;
38
39 R=Oafe+Ocfe;
40 mprintf("\n(a)Annual Outage Rate = %.3f outage per
      year",R);
41
42 TperO=((Oafe*Rae)+(Ocfe*Rce))/R;
43 mprintf("\n(b)Downtime per Outage = %.2f hours",
      TperO);
44
45 T=TperO*R;
46 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

`find failure rate of system down time per outage annual outage`

```
1 clear;
2 clc;
3
4 On=3;
5 Os=9
6 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);
```