Scilab Textbook Companion for Electrical Power Systems by A. Husain[1]

Created by
Shikhar Saxena
B.E. 3rd yr
Electrical Engineering
Thapar University
College Teacher
Dr Sunil Kumar Singla
Cross-Checked by
Lavitha Pereira

July 31, 2019

[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: Electrical Power Systems

Author: A. Husain

Publisher: CBS Publishers

Edition: 5

Year: 2009

ISBN: 8123914482

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes

Chapter 1

Load Characteristics

Scilab code Exa 1.1 `to calculate the average load monthly energy consumption and l`

```
clear;
clc;
nos_lmps=8;
pow_of_lmp=60;                              //power in watts
dur_per_day=5;                             //duration in hours
t_dur=24;                                 //duration in hours
max_dem=1500;                            //demand in watts

nos_heaters=2;
pow_of_heaters=1000;                     //power in watts
dur_per_day_heater=3;                   //duration in hours

enrgy_lmp=(nos_lmps*pow_of_lmp*dur_per_day);      //energy in watt hour
energy_heater=(nos_heaters*pow_of_heaters*dur_per_day_heater);  //energy in watt hour

act_ene_con=(enrgy_lmp + energy_heater);          //energy in watt hour
```

```
avg_load=(act_ene_con/t_dur);                          //avverage load in  watt
printf("\n The average load of the consumer is:  %.2f W",avg_load);
mon_ene_con=(act_ene_con*30);                         //monthly energy consumption in wtthour
printf("\n The monthly energy consumption of the consumer is:  %.0f kWh",mon_ene_con/1000);

load_fac=(avg_load/max_dem);                          //load factor is unitless
printf("\n The load factor of the consumer is:  %.4f",load_fac);
```

Scilab code Exa 1.2 to calculate the diversity factor avg load and laod factor of

```
//to calculate the a).diversity factor  b).avg load and laod factor of each consumer c).avg load and load factor of combined load.
//postfix 1,2...4 shows the consumers.

clc;
//for consumer1
avg_load1=1;                    //load in kwatt
max_dem1=5;                    //demand in kwatt
dem1_at8=5;                      //demand in kwatt
l_f1=(avg_load1/max_dem1);    //load factor is unitless
printf("\n the load factor for consumer 1 is:  %.2f\n",l_f1);

//for consumer2
max_dem2=2;                    //demand in kwatt
dem2_at8=1.6;                        //demand in kwatt
```

```
l_f2=0.15;                    //load factor is unitless
avg_load2=(l_f2*max_dem2); //load in kwatt
printf("\n the avg load for consumer 2 is: %.2f \n",avg_load2);

//for consumer3
avg_load3=0.5;                    //load in kwatt
max_dem3=2;                      //demand in kwatt
dem3_at8=1;                      //demand in kwatt
l_f3=(avg_load3/max_dem3);    //load factor is unitless
printf("\n the load factor for consumer 3 is: %.2f \n",l_f3);

//for consumer4
max_dem4=10;                    //demand in kwatt
dem4_at8=5;                      //demand in kwatt
l_f4=0.25;                    //load factor is unitless
avg_load4=(l_f4*max_dem4); //load in kwatt
printf("\n the avg load for consumer 4 is: %.2f \n",avg_load4);

tot_avg_load=(avg_load1 + avg_load2 + avg_load3 + avg_load4); //load in kwatt
printf("\n the total avg load is: %.2f \n",tot_avg_load);

tot_max_dem=(max_dem1 + max_dem2 + max_dem3 + max_dem4);    //demand in kwatt
printf("\n the total maximum demand is: %.2f\n ",tot_max_dem);

tot_dem_at8=(dem1_at8 + dem2_at8 + dem3_at8 + dem4_at8);               //demand in kwatt
printf("\n the total demand at 8pm is: %.2f\n ",tot_dem_at8);

div_fact=(tot_max_dem/tot_dem_at8);
```

```
                    //diversity factor is unitless
printf("\n the diversity factor is:  %.2f\n ",
    div_fact);

load_factor=(tot_avg_load/tot_dem_at8);
    //load factor is unitless
printf("\n the load factor is:  %.2f\n ",load_factor
    );
```

Scilab code Exa 1.3 to find the maximum value and the consumption of energy in kWh

```
clear;
clc;

ene_con=600;
lf=.45;
t=24;
max_dem=ene_con/(t*lf);
printf("The maximum demand is:%.2f kW\n",max_dem);

lf=.65;
ene_con=lf*max_dem*t;
printf("The energy consumed in 24h is:%.2f kWh",
    ene_con);
```

Scilab code Exa 1.4 to plot the load duration curve from the chronological load cu

```
clear;
clc;

x=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
    20 21 22 23 24 ];//dividing the x axis in 24
    hours
```

```

y=[30 30 30 30 20 20 20 20 20 8 8 8 8 8 8 8 8 8 8 8 8 8 8 5];//load in MW values

bar(x,y,1,'blue');                    //plotting the bargraph with a width of 1
xlabel('time in hours');
ylabel('load in MW');
title('LOAD DURATION CURVE');
```

Scilab code Exa 1.5 to determine the load factor from the load duration curve

```
clear;
clc;
l1=15;                   //load in MW
l2=10;                  //load in MW
l3=5;                  //load in MW

t1=8;                 //time in hours
t2=8;                //time in hours
t3=8;               //time in hours
t=24;              //time in hours

max_dem=max(l1,l2,l3); //load in MW
printf("\n the maximum demand is: %.2f MW\n ",max_dem);

act_ener_consum=(l1*t1 + l2*t2 + l3*t3);  //energy consumed in MWH
printf("\n the actual energy consumed is: %.2f MWH\n ",act_ener_consum);

avg_load=(act_ener_consum/t);    //load in MW
printf("\n the average demand is: %.2f MW\n ",avg_load);
```

```
l_f=(avg_load/max_dem);          //load factor is unitless
printf("\n the load factor is: %.2f\n ",l_f);
```

Scilab code Exa 1.6 to calculate plant capacity factor load factor utilization fac

```
clear;
clc;
max_dem=500;                              //demand is in MW
min_dem=400;                             //demand is in MW
cap_plnt=750;                           //capacity of the plant is MW

avg_ann_load=(max_dem + min_dem)/2;    //load is in MW
printf("\n the average annual load is: %f MW\n ",avg_ann_load);

cap_f=(avg_ann_load/cap_plnt);        //capacity factoe is dimentionless
printf("\n the capacity factor is: %f\n ",cap_f);

l_f=(avg_ann_load/max_dem);         //load factor is dimentionless
printf("\n the load factor is: %f\n ",l_f);

u_f=(max_dem/cap_plnt);            //utilization factor is dimetionless
printf("\n the utilization factor is: %f\n ",u_f);
```

Scilab code Exa 1.7 to plot the chronological load duration load energy curve and

```
// to plot the chronological ,load duration , load
   energy curve and then
//calculating the load factor and the utilization
   factor

clc;
clf();
inst_cap=100;           //installed capacity in MW
max_dem=70;             //demand in MW
t1=6;                   //time in hours
t2=9.5;
t3=0.5;
t4=2;
t5=6;
d1=70;                  //demand in MW
d2=50;
d3=40;
d4=200;
d5=10;

scf(0);
x=[0 1 2 3 4 5 6 7 8 9 10 11 12 12.5 13 14 15 16 17
   18 19 20 21 22 23 24];
//shows the time from 12pm to 12 am
//0 shows 12am and 24 shows 12 pm

y=[20 20 10 10 10 10 10 10 50 50 50 50 40 40 50 50
   50 50 50 50 70 70 70 70 70 70];
//loads in MW

bar(x,y,0.5,'blue');
xlabel('TIME');         //this is time
ylabel('LOAD IN MW');
title('CHRONOLOGICAL LOAD CURVE');


scf(1);
x=[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
```

```
20 21 22 23 24];
//time in hours

y=[70 70 70 70 70 70 70 50 50 50 50 50 50 50 50 50 40 20 20 10 10 10 10 10 10];
//loads in MW

bar(x,y,1,'red');
xlabel('TIME IN HOURS');
ylabel('LOAD IN MW');
title('LOAD DURATION CURVE');

ene=((d1*t1) + (d2*t2) + (d3*t3) + (d4*t4) + (5*t5));      //energy in MWH

d_lf=(ene/(max_dem*24));                              //demand factor is dimentionless
printf("\n the demand factor is:  %f\n ",d_lf);

u_f=(max_dem/inst_cap);                              //utilization factor is dimentionless
printf("\n the utilization factor is:  %f\n ",u_f);

scf(2);
x=[0,10,20,40,50,70];
y=[0,240,420,740,895,1015];
plot(x,y);
xlabel('LOAD IN MW');
ylabel('ENERGY IN MWH');
title('LOAD ENERGY CURVE');
```

Scilab code Exa 1.8 to find the diversity factor of a power station

```
//to find the diversity factor of a power station

```

```
clc;
maxa=150;
maxb=50;
maxc=55;
c_max=205;

tot_max=(maxa + maxb + maxc);
printf("\n the maximum demand is:  %.2f kW\n ",tot_max);

d_f=(tot_max/c_max);
printf("\n the diversity factor is:  %.2f\n ",d_f);
```

Scilab code Exa 1.9 to calculate max demand on the station installed capacity ener

```
//to calculate a).max demand on the station  b).installed capacity  c).energy supplied in a year

clc;
peak_l1=25;            //load is in MW
peak_l2=20;           //load is in MW
peak_l3=30;          //load is in MW
ann_lf=0.6;         //load factor is dimentionless
d_f=1.65;          //diversity factor is dimentionless

max_dem=(peak_l1 + peak_l2 + peak_l3)/d_f;
printf("\n the maximum demand is:  %.2f MW\n ",max_dem);   //demand in    MW

i_cap=(peak_l1 + peak_l2 + peak_l3);      //capacity in MW
printf("\n the installed capacity is:  %.2f MW\n ",i_cap);

```

```
avg_l=(ann_lf*max_dem);
esp_year=(avg_l)*8760;                    //energy
    supplies per year in MWH
printf("\n the energy supplied in a year is: %.2f
    MWH\n ",esp_year);
```

Chapter 2

Supply Systems

Scilab code Exa 2.1 To calculate the percentage saving in conductor material

```
clear;
clc;
//i=p/v;
//a=i/alpha;
//vol=2*l*a;

//calculating itf or the 2 supplies 250V and 400V
//a=v2/v1;
a=.625;
sav=(1-a)*100;
printf("the percenatge saving in the conductor
    material is:%.2f per cent",sav);
```

Scilab code Exa 2.2 To compare the amount of conductor material required

```
clear;
clc;

```

```
//i1=p/v
//line loss=2i^2*r
//vol of conductor required v1=2*l*a1

a=1/6;  //a=r1/r2
b=1/a  //b=a2/a2

//v2=3*l*a2+ 0.5*l*a2
//v2/v1=3.5*l*a2/2*l*a1

c=3.5/(2*b);  //c=v2/v1
printf("the ratio of the volumes of the conductors
    is:%.2f",c);
```

Scilab code Exa 2.3 To calculate the volume of conductor

```
clear;
clc;

s=5*10^6;
l=50*10^3;
pf=0.8;
eff=.9;
v=33*10^3;
rho=2.85*10^(-8);
pl=0.1*s*pf;
i=s/v;
a1=2*i*i*rho*l/pl;
vol=2*l*a1;
printf("the volume of theconductor required is:%.2f
    cubic meter",vol);

//b)

il=s/(sqrt(3)*v);
```

```
a2=3*il*il*rho*l/pl;

vol=3*l*a2
printf("\n the volume of theconductor required is:%
   .2f cubic meter",vol);
```

Scilab code Exa 2.4 To calculate the percentage of additional load

```
clear;
clc;

//a). single phase supply
//p1=v1*i1*pf
//line loss= 2*i1*i1*r
//percentage line loss=line loss*100/(v8i*pf)

//b).three phase supply
//p3=sqrt(3)*v*i3*pf
//percentage line loss=3*i3*i3*r*100/p3

//ratio of load transmitted

a=2;  //a=p3/p1
add_load=(a-1)*100;  //(p3-p1)/p1

printf("the percentage of additional load is:%.2f",
   add_load);
```

Scilab code Exa 2.5 To compare the diameter and weight

```
clear;
clc;

```

```
rhoa=2.85
rhoc=1.70;
sig1=2.71;
sig2=8.89;

a=sqrt(rhoa/rhoc);  //a=diameter of aluminium/
    diameter of copper
printf("the ratio of diameters is:%.2f\n",a);
b=sig1*rhoa/(sig2*rhoc);  //weight of aluminium/
    weight of copper

printf("the ratio of weights is:%.2f",b);
```

Chapter 3

Conductors

Scilab code Exa 3.1 To determine the most economical size

```
clear;
clc;
ld=100;            //load in KW


v=500;............// voltage in volts
res=1.75*(.000001);...........//restivity in milli
   ohm per cm sq
nos_cores=2;.......//number of feeder core
l=0.8;.............//length of tx line in km
area=1;...........//area in cm sq
cost_of_energy=0.12;..//cost in Rs per unit
dep=0.1..............//depreciation percentage is 10
   %

flc=(ld*1000)/500;..//full load current

ra=(res*l*1000*ld);..//resistance* area in ohm-m

pow_loss=2*(flc*flc*ra*.001);...//(power loss*area)
   for the two cores in the cable
```

```
ann_en_loss=pow_loss*365*24;..//annual energy loss
    in KWH
cost=ann_en_loss*cost_of_energy;..//(cost*area) of
    annual energy loss in Rs

ann_dep=6*l*1000*dep;........//(ann_dep*area) in Rs

c=(cost/ann_dep);

area=sqrt(c);.......//area in cm sq

d=(area*4/(%pi));
dia=sqrt(d);..........//diameter in cm
printf("\n the most economical size is:  %.2f cm\n "
    ,dia);
```

Scilab code Exa 3.2 to calculate the most economical cross sectional area

```
clear;
//clc();
l=10;.....//length of tx line in km

s1=2500;....// power in KVA
s2=2000;....// power in KVA
s3=1500;....// power in KVA

t1=8;.......//time in hrs
t2=9;.......//time in hrs
t3=7;.......//time in hrs

dep_rate=0.08;..//depreciation rate
cost=0.15;......//cost of energy in Rs
res=0.173.......//resistance per sq cm

```

```
nos_wd=250;.....//number of working days

ann_int=6000*l*dep_rate;..//annual interst and
    depriciation on capitol cost in Rs

r=res*l;..//(resistance*area) of each conductor in
    ohms

c=sqrt(3);
d=c*33;....//kv of tx line
i1=(s1/d);....//load current at 2500KVA in A

i2=(s2/d);....//load current at 2000KVA in A

i3=(s3/d);....//load current at 1500KVA in A

d_en_loss=(3*r/1000)*(t1*i1*i1 + t2*i2*i2 + t3*i3*i3
    );..//daily energy loss in KWH

ann_cost=d_en_loss*nos_wd;..//annual enrgy loss in
    KWH

cost_el=ann_cost*cost;..//(cost of energy loss per
    annum)*area

e=cost_el/ann_int;..//by kelvin's law area
f=sqrt(e);
printf("\n the most economical area is: %.2f sq cm\
    n ",f);..//area in cm sq
```

Scilab code Exa 3.3 to calculate the best current density of a three phase oh line

```
clear;
//clc();
a=2500;..//usage of the over head line in a year in
```

```
hrs
cost=20;..//cost of copper per kgf in Rs
dep_rate=0.125;..//depriciation rate
den=8.89;..//density of copper in gf per cm sq
r=0.173;...//resistance per conductor per km length and per sq cm in ohms
cost_en=0.16;..//cost of energy perm unit in Rs

cap_cost=cost*den/100;..//(capitol cost of conductor)*length/area in Rs
dep=cap_cost*dep_rate;

cl=r/10000000;..//copper loss per conductor/current sq*length/area

ce=a*cost_en*cl;..//cost of energy loss per year)*area/(length*current sq)

d=dep/ce;

j=sqrt(d);..//best current density in A/cm sq
printf("\n the current density is: %.2f A/sq meter\n",j);
```

Scilab code Exa 3.4 To calculate rms value of curent for a 3 phase

```
clear;
//clc();
s=7500;..//maximum load in KVA
v=33;..//voltage of tx line in volts

a=sqrt(3);
im=s/(v*a);..//maximum current in A

kf=1.3;..//for a 0.4 load factor
```

```
irms=kf*0.4*im;
printf("The rms value of current is:%.2f A",irms);
```

Scilab code Exa 3.5 To determine the most economical size of copper conductor

```
clear;
clc;
funcprot(0);
s=7500;
v=33;
lf=0.6;
kf=1.13;
cost_en=0.15;
dep_rate=0.1;
res=17.6;
im=s/(v*sqrt(3));
irms=0.6*kf*im;
rho=(17.6*10^4)/(10^5);
el=3*irms*irms*.1*rho*8760/1000;
ann_w=el*cost_en;
ann_dep=dep_rate*20000;
c=ann_w/ann_dep;
area=sqrt(c);
printf("The area is:%.2f sq cm",area);
```

Scilab code Exa 3.6 To calculate the most economical current density

```
clear;
//clc();
lc=0.5;
dep=0.09;
cost_en=0.12;
res=1.76;
```

```
l=1;

ann_cost=dep*25000;
lf=0.2*lc + 0.8*(lc^2);
al=3*.1*res*8760*lf/1000;
cost=al*cost_en;
d=ann_cost/cost;
j=sqrt(d);
printf("The current density is:%.2f A/sq cm",j);
```

Chapter 4

Power Cables

Scilab code Exa 4.1 To calculate the maximum stress on the insulation

```
//to calculate the maximum electric stress on the
   insulation

//clc();
v=6.5;..//working voltage in volts
d=10;..//diameter of conductor in mm
t=7.5;..//thickness of insulation in mm
r=d/2;..//radius of conductor in mm
R=(r+t);
a=R/r;
b=log([a]);

gmax=v/(r*b);
printf("\n the maximum electric stress on the
   insulation is:  %.2f kV/mm\n",gmax);
```

Scilab code Exa 4.2 To calculate the overall diameter and its most economical diam

```
clear;
//clc();
v=275;..//working voltage in volts
vrms=v/sqrt(3);..//effective value of phase voltage
pv=vrms*sqrt(2);..//peak value of phase voltage
gmax=15;..//maximum permissible stress in dielectric

r=pv/gmax;

d=2*r;..//economical core diameter in mm
printf("\n the economical core diameter is:  %.2fmm\
   n",d);

R=exp(1)*r;

D=2*R;..//inner diameter of the sheath
printf("\n the inner diameter of the sheath is:  %.2
   fmm\n",D);
```

Scilab code Exa 4.3 To calculate potential gradient at the surface of teh conducto

```
clear;
//clc();

d=10;..//diameter of the conductor in mm
r=d/2;..//radius in mm
t=10;..//thickness in mm
r1=(r+t);
R=(r+ 2*t);
e1=3;
e2=2.5;
v=60;..//voltage in kv

a=r1/r;
b=R/r1;
```

```
c=e1/e2;

d=(r)*(log([a]) + c*log([b]));

gmax1=v/d;

printf("\n the potenial gradient at the surface of
   the conductor is:%.2f kV/mm\n",gmax1);
```

Scilab code Exa 4.4 To calculate the minimum diameter of the lead sheath

```
clear;
//clc();

v=66;
r=10;
e1=5;
e2=4;
e3=3;
gmax1=3.8;

gmax2=2.6;

gmax3=2;

r1=e1*r*gmax1/(e2*gmax2);

r2=e1*r*gmax1/(e3*gmax3);

a=r1/r;
b=r2/r1;

v1=gmax1*r*log([a]);

v2=gmax2*r1*log([b]);
```

```
c=(v-v1-v2)/(gmax3*r2);
e=exp(c);

R=e*r2;

dia=2*R;
printf("\n the minimum diameter is:  %.2f mm\n ",dia
    );
```

Scilab code Exa 4.5 To calculate the diameter of the inter sheath and the voltage

```
clear;
//clc();

v=60;
gmax=4;
v1=v/exp(1);
r=v1/gmax;
d=2*r;
r1=v/gmax;
d1=2*r1;
v2=v-v1;
R=1.88*r1;
D=2*R;
printf("\n the diameter of cable is:  %f cm\n ",D);

//cable without sheath

x=exp(1);
a=log([x]);
r1=v/gmax;
d1=2*r1;

R1=exp(1)*r1;
```

```
D1=2*R1;
printf("\n the diameter of cable without sheath is:
    %.2f mm\n ",D1);
```

Scilab code Exa 4.6 capacitance between any pair of conductors and the charging cu

```
clear;
//clc();
v=11;
ct=0.7;
cs=0.4;
cc=(ct-cs)/2;
printf("\n the capacitance between conductors is: %
    .2f uF\n ",cc);

cl=0.5*(3*cc + cs);

ic=(v*2*3.14*50*2*cl*.001)/sqrt(3);..//charging
    current in ka/phase
printf("\n the charging current is: %.3f A\n ",ic);
```

Scilab code Exa 4.7 To calculate charging kvar

```
clear;
//clc();

v=11;
cl=0.3;
l=5;
c0=2*cl*l;

tot_var=(v*v*2*3.14*50*cl*.01);
```

```
printf("\n the charging kvar is:%.2f KVAR\n ",
   tot_var);
```

Scilab code Exa 4.8 To calculate restitivity of the insulating material

```
clear;
//clc();

r=0.4;
l=5;
d=20;
D=50;
a=D/d;
b=log([a]);

rho=2*3.14*l*r/b;..//resistivity in mega-ohm-m
printf("\n the resistivity is:%.2f M-Ohm-M\n ",rho);
```

Scilab code Exa 4.9 To calculate capacitance charging current ic reactive var diel

```
clear;
//clc();

v=11;
f=50;
d=20;
r=15;
er=2.4;
la=0.031;
l=2.5;

a=r*2/d;
b=log([a]);
```

```

cap=er/(18*b);
tot_cap=cap*l;
printf("\n the total capacitance is: %.2f F\n ",
   tot_cap);

//ic

ic=2*3.14*f*tot_cap*v;
printf("\n the charging current is: %.2f A\n ",ic
   /1000);

//reactive var
var=v*ic;
printf("\n the reactive var is: %.2f kVAR\n ",var
   /1000);

//dielectric loss

pd=v*ic*la;
printf("\n the dielectric loss is: %.2f W\n ",pd);

ri=v*v/pd;..//resistance in mega-ohm
printf("\n the resistance is: %.2f M-OHM\n ",ri);
```

Chapter 5

Line insulators and supports

Scilab code Exa 5.1 To calculate the voltage distribution across each unit and the

```
clear;
//clc();
V=100;..//voltage betwen line conductor and earth
k1=0.1;
n=4;
k=sqrt(k1);

V1=(V*sinh(k)/sinh(n*k));
printf("\n the voltage1 is: %.2f V\n ",V1);

V2=(V*sinh(2*k)/sinh(n*k));
printf("\n the volatge2 is: %.2f V\n ",V2);

V3=(V*sinh(3*k)/sinh(n*k));
printf("\n the voltage3 is: %.2f V\n ",V3);

V4=V;
printf("\n the voltage4 is: %.2f V\n ",V4);

v1=V1;

```

```
v2=V2-V1;

v3=V3-V2;

v4=V4-V3;

eff=(sinh(n*k)/(n*(sinh(n*k)-sinh((n-1)*k))));
printf("\n the string efficiency is:  %.2f percent\n
    ",eff*100);
```

Scilab code Exa 5.2 To calculate voltage across the lowest unt and the string effi

```
clear;
//clc();
n=6;
k1=0.1;
k=sqrt(k1);
a=sinh((n-1)*k)/sinh(n*k);
b=1-a;
printf("\n the voltage across lowest unit is:  %.2f
   V\n ",b*100);
eff=1/(n*b);
printf("\n the string efficiency is:  %.2f percent",
   eff*100);
```

Scilab code Exa 5.3 to find the voltage distribution and the string efficiency

```
clear;
clc;

n=3;
c=[1.6 0 -1;1 0 0 ;1 1 1];//coeffient matrix
b=[0.109;.31;1];
```

```
v=inv(c)*b;
for i=1:1:3
    printf("The voltage across unit%d is:%.2f pu\n",i,v(i))
end

eff=1/(n*v(n));
printf("\n the string efficiency is: %.2f percent\n",eff*100);

//with the string grading
a=[3.6037 0 0;1.3037 0 -1 ;1 1 1];//coeffient matrix
d=[1.1889;0.0389;1];

v=inv(a)*d;
for i=1:1:3
    printf("The voltage across unit%d is:%.2f pu\n",i,v(i))
end

eff=1/(n*v(n));
printf("\n the string efficiency is: %.2f percent",eff*100);
//difference in answers is due to miscalculations
```

Scilab code Exa 5.6 to find the voltage between the conductors and the string effi

```
clear;
clc;

n=4;
v2=14.2;
v3=20;
a=14.2;..//coeffiecients of the quadratic equation
```

```
    14.2*k*k+22.6*k-5.8
b=22.6;
c=-5.8;

k=(-b+sqrt(b*b-4*a*c))/(2*a);
v1=v2/(1+k);
v4=(1+6*k+5*k*k+k*k*k)*v1;
V=v1+v2+v3+v4;
vl=sqrt(3)*V;
eff=V/(n*v4);

printf("\n the string efficiency is:  %.2f percent",
    eff*100);
```

Scilab code Exa 5.7 to calculate the voltage across each unit as a percentage of t

```
clear;
clc;

//by using kcl at node a
//20*v1-18*v2-3*v3=0
//by kcl at node b
//5*v1+20*v2-18*v3=0
//total voltage across the string
//v1+v2+v3=V
c=[20 -18 -3;5 20 -18 ;1 1 1];//coeffient matrix
b=[0;0;1];
v=inv(c)*b;
for i=1:1:3
    printf("The voltage across unit%d is:%.2f
        percent\n",i,v(i)*100)
end
```

Chapter 6

Sag and tension

Scilab code Exa 6.1 To calculate the sag

```
clear;
clc;

uts=5758;   //ultimate tensile strength
l=200;
wt=.604;

h=uts/2;
sag=wt*l*l/(8*h);

printf("The sag is:%.3f m",sag);
```

Scilab code Exa 6.2 to calculate the deflected sag and the vertical component of s

```
clear;
clc;

wc=.844;
```

```
t=9.53;
d=19.53;
bl=7950;
fs=2;
p=40;
wi=(%pi)*t*(t+d)*913.5*10^(-6);
wh=(d+2*t)*p*10^(-3);
wr=sqrt((wc+wi)^2 + wh^2);

H=(bl/fs);  //bl=breaking load     fs=factor of
    safety
l=275;
sag=wr*l*l/(8*H);
printf("The deflected sag is:%.3f m",sag);

vsag=(wc+wi)*l*l/(8*H);
printf("\n The vertical component of sag is:%.2f m",
    vsag);
```

Scilab code Exa 6.3 determine tension and sag

```
clear;
clc;

uts=14740;
a=538.4;
E=7000;
wc=1.805;
p=100;
l=335;
alpha=19.3*10*(-6);
fs=2;
n=4;
ds=3.35;
d=(1+2*n)*ds;
```

```
wh=(2/3)*d*p;

//a).
wr=sqrt((wc+wh)^2 + wh^2);
T=(uts/fs);  //bl=breaking load=uts     fs=factor of safety
printf("The maximum working stress is:%.2f kgf",T);
sag=wr*l*l/(8*T);
printf("\n The deflected sag is:%.3f m",sag/1000);

//b).

t1=60;
T1=T;
w1=wr;

K=T1-w1*w1*l*l*E*a/(24*T1*T1);
b=alpha*(t1-0)*E*a;
//by using the formula t2^2(t2-K+b)=w2^2*l*l*e*a/24 and getting a solution by hit trial

T2=4083;
sag_60=wc*l*l/(8*T2);
printf("\n The sag is:%.3f m",sag_60);

//c).
fs=4;
t2=30;
T=(uts/fs);  //bl=breaking load=uts     fs=factor of safety
K=T1-w1*w1*l*l*E*a/(24*T1*T1);
b=alpha*(t1-t2)*E*a;

//by using the formula t2^2(t2-K+b)=w2^2*l*l*e*a/24 and getting a solution by hit trial

T2=3132;
printf("\n The maximum working stress is:%.2f",T2);
```

```
sag_60=wc*l*l/(8*T2);
printf("\n The sag is:%.2f m",sag_60);
//similar calculations can be made for other temperatures and loading conditions
```

Scilab code Exa 6.4 to calculate the temperature at which sag will remain same und

```
clear;
clc;

wc=.594;
a=64.5;
t=9.53;
d=3*3.45;
sag=3.96;
p=40;
E=12700;
l=160;
alpha=1.7*10^(-5);
wi=(%pi)*t*(t+d)*913.5*10^(-6);
wh=(d+2*t)*p*10^(-3);
wr=sqrt((wc+wi)^2 + wh^2);
w1=wr;
T1=w1*l*l/(sag*8);

w2=wc;
t1=-5.5;
T2=wc*T1/w1;
//by using the formula t2^2(t2-K+b)=w2^2*l*l*e*a/24

t2=t1+(T1-T2)/(alpha*E*a);
printf("The temperature at which the sag will remain the same:%.2f degC",t2);
```

Scilab code Exa 6.5 to find the clearance between the conductor and the water at a

```
clear;
clc;

l=270;
T=1800;
w=1;
h=90-30;
ap=30;
x=(l/2)-T*h/(w*l);
x1=-x+l/2;
sag1=w*x1*x1/(2*T);
sag2=w*x*x/(2*T);
hob=w*(l-x)^2/(2*T);

clearance=ap+sag1-sag2;
printf("The clearance between the conductor and
    water at point m is:%.2f m",clearance);
```

Chapter 7

Line Parameters

Scilab code Exa 7.1 To calculate the inductance of each inductor inductive reactan

```
clear;
//clc();
D=3;..//spacing between the conductors in m
r=0.01;..//radius of each conductor in m
r1=0.7788*r;
mu=50;..//relative permeability of steel
a=log([D/r1]);
b=log([D/r]);

l=2*10^(-7)*a*(10^6);..//inductance of each conductor in henry per km
printf("\n the inductive reactance is: %.2f mH/km\n ",l);

loop_l=2*l;
printf("\n the loop inductance is: %.2f mH/km\n ",loop_l);



react=2*(%pi)*50*loop_l*.001;
```

```
printf("\n the inductive reactance  is:  %.2f Ohm/km
   \n ",react);

//loop inductance with steel conductors

l_in=0.5*mu;
l_ex=2*b;

l_inductance=2*(l_in+l_ex)/10;
printf("\n the loop inductance  is:  %.2f mH/km\n ",
   l_inductance);

2
```

Scilab code Exa 7.2 To calculate inductance and inductive reactance per phase per

```
clear;
//clc();
r=10.5;..//radius of the conductor
d_ab=3;
d_bc=5;
d_ca=3.6;

Deq=(d_ab*d_bc*d_ca)^(1/3);

r1=0.7788*r/1000;

a=log([Deq/r1]);

l=2*10^(-4)*a;
printf("\n the inductance is:  %.5f H/km\n ",l);

xl=2*(%pi)*50*l/10000;
printf("\n the inductive reactance is:  %f Ohm/km\n
   ",xl);
```

Scilab code Exa 7.3 To calculate the inductive reactance per phase per km of the l

```
clear;
//clc();

d=6;
r=12.5;...//radius of each conductor
dm=(d*d*2*d)^(1/3);

dsl=0.7788*r/1000;

a=log([dm/dsl]);
l=2*a*10^(-7);

xl=2*(%pi)*l*50*1000;
printf("\n the inducatnce is:  %.2f Ohm/km\n",xl);
```

Scilab code Exa 7.4 To calculate the effective inductance of the line

```
clear;
//clc();

d_ab=3;
d_bc=4;
d_ca=5;
r=0.015;

d_aa=r*exp(-0.25);
d_bb=r*exp(-0.25);
d_cc=r*exp(-0.25);

```

```
la=.2*(log([sqrt(d_ab*d_ca)/d_aa]) + %i*0.866*log([d_ab/d_ca]));

lb=2*(log([sqrt(d_ab*d_bc)/d_bb])/10 + %i*0.866*log([d_bc/d_ab])/10);

lc=2*(log([sqrt(d_bc*d_ca)/d_cc])/10 + %i*0.866*log([d_ca/d_bc])/10);

lav=(la +lb + lc)/3;
printf("\n the average inductance is:  %f mH/km\n",lav);
```

Scilab code Exa 7.5 To calculate the inductance of the line per km

```
clear;
//clc();
D1=1;
D2=2;
r=0.5*25/1000;
a=sqrt(D1*D1+D2*D2)*D2;
b=0.7788*r*D1;
l=2*log([a/b])/10;
printf("\n the inductance is:  %.3f mH/km\n",l);
```

Scilab code Exa 7.6 To calculate the inductive reactance per phase per km of the s

```
clear;
//clc();
r=0.5*3*4.75/1000;
d1=3;
d2=6;
dac1=6;
```

```
dbb1=9;
dca1=6;
dac=6;
dc1a1=6;

dab=sqrt(d1*d1 + (d1/2)*(d1/2));
dbc=dab;
da1b1=dab;
db1c1=dab;

dab1=sqrt(d1*d1 + (dac+d1/2)*(dac+d1/2));

dbc1=dab1;
dba1=dab1;
db1c=dab1;
da1b=dab1;

daa1=sqrt(d2*d2 + d2*d2);

dcc1=sqrt(d2*d2 + d2*d2);

mgmd=(dab*dbc*dac*dab1*dbc1*dca1*da1b*db1c*dac1*
    da1b1*db1c1*dc1a1)^(1/12);
sgmd=(((0.7788*r)^3)*(daa1*dbb1*dcc1))^(1/6);
l=2*log([mgmd/sgmd]);
xl=2*(%pi)*50*l*10^(-5);
printf("\n the inductance  is:  %.4f Ohm/km\n ",xl);
```

Scilab code Exa 7.7 determine the line inductance and the inductive reactance

```
clear;
//clc();
d=6;
s=0.3;
r=12.5/1000;
```

```
dab=6;
dbc=6;
dca=12;

dsl=sqrt(0.7788*r*s);

dm=(dab*dbc*dca)^(1/3);

lb=2*log([dm/dsl])/10;

xl=2*(%pi)*50*lb;

ds1=0.7788*r*sqrt(2);

l1=2*log([dm/ds1])/10;

xl1=2*(%pi)*50*l1;
printf("\n the line inductance is:  %.3f Ohm/km\n",
   xl1/1000);

pu_red=(xl1-xl)/xl1;
printf("\n the pu reduction is:  %.3f\n",pu_red);
```

Scilab code Exa 7.8 To calculate the capacitance of each conductor to neutral per

```
clear;
//clc();
d=3;
r=0.01;
cn=2*(%pi)*8.85*10^(-12)/log([d/r])*1000000000000;
printf("\n the capacitance is:  %.2f F/km\n ",cn);

cl=0.5*cn;
printf("\n the line to line capacitance is:  %.2f
   *10^(-9)F/km\n ",cl);
```

```
bc=2*(%pi)*50*cn;
printf("\n the capacitance susceptance is:  %.2f *10^(-6) S/km\n ",bc/1000);
```

Scilab code Exa 7.9 To calculate the charging current per km and the reactive volt

```
clear;
//clc();
//taking the values calculated in eg7.2.sce
dm=(3*5*3.6)^(1/3);
dsc=10.5/1000;
vn=132*1000;

cn=2*(%pi)*8.85*10^(-12)/log([dm/dsc]);

xc=1/(2*(%pi)*50*cn);

ic=vn/(xc*sqrt(3));
printf("\n the charging current is:%.3f A/km\n",ic *1000);

va=(vn*vn)/xc;
printf("\n the volt-ampere is:%.2f kVAr\n",va);
```

Scilab code Exa 7.10 to find the capacitance per km to neutral and the capacitive

```
//taking calculations in eg7.3
clear;
//clc();
dm=7.56;
dsc=12.5/1000;
r=dsc;
```

```
cn=1/(18*10^(9)*log([dm/dsc]))*10^(12);
printf("the capacitance is: %.2f*10^(-9) F/km\n ",cn);

xc=1/(2*(%pi)*50*cn);
printf("the reactance is: %.2f*10^(5) Ohm\n ",xc*10000);
```

Scilab code Exa 7.11 To calculate the capacitive reactance per phase per km of the

```
clear;
//clc();
dm=5.4576;
r=7.125/1000;
daa1=6*sqrt(2);
dbb1=9;
dcc1=daa1;

dsc=((r^3)*(daa1*dbb1*dcc1))^(1/6);

cn=1/(18*10^(9)*log([dm/dsc]));
printf("\n the capacitance is: %.2f uF/km\n ",cn*10^(12));

xc=1/(2*(%pi)*50*cn);
printf("\n the reactance is: %.2f Ohm/km\n ",xc*.001);
```

Scilab code Exa 7.12 To calculate the capicitance of the line

```
//values taken from eg7.1
clear;
```

```
//clc();
r=10/1000;
d=3;
h=6;
h1=2*h;
h2=2*h;
h12=sqrt(d^2 + 4*h^2);

h21=h12;

hm=sqrt(h12*h21);
hs=sqrt(h1*h2);

cn=1/(18*(10^9)*(log([d/r])-log([hm/hs])));
printf("\n the capacitance  is:  %f*10^(-12) F/m\n "
    ,cn*10^(12));

cl=0.5*cn;
printf("\n the capacitance  is:  %f*10^(-9) F/km\n "
    ,cl*10^(12));
```

Scilab code Exa 7.13 To calculate the capacitance per km to neutral and the capaci

```
clear;
//clc();
//values as calculated in eg7.7.sce
d=6;
s=0.3;
r=12.5/1000;

dsc=sqrt(r*s);

dm=7.56;
cn=1/(18*(10^(9))*(log([dm/dsc])));
printf("\n the capacitance per km is: %.2f F/km\n",
```

```
    cn*10^(12));

xcb=1/(2*(%pi)*50*cn*1000);
printf("\n the reactance is: %.2f Ohm\n",xcb*.00001)
    ;

ds1=sqrt(2)*r;
dm1=7.56;

c1=1000/(18*(10^9)*(log([dm1/ds1])));
printf("\n the capacitance is: %.2f F/km\n",c1
    *10^(9));

xc1=1/(2*(%pi)*50*c1);
printf("\n the reactance is: %.2f Ohm\n",xc1);

cb=13.721*(10^(-9));
pu_cap=(cb-c1)/c1;
printf("\n the per unit capacitance is: %.3f\n",
    pu_cap);
```

Scilab code Exa 7.14 To calculate the inductive recatance and the capacitive react

```
clear;
//clc();
d=20;
s=0.5;
r=20/1000;
dab=20;
dbc=20;
dca=40;

dsl=((sqrt(2)*0.7788*r*(s*s*s))^(1/4));

dm=(dab*dbc*dca)^(1/3);
```

```

lb=2*log([dm/dsl]);

xlb=2*(%pi)*lb*50;

dsc=(sqrt(2)*r*(s^3))^(1/4);

cn=1/(18*(10^(9))*(log([dm/dsc])));
printf("the capacitance is: %.2f*10^(-9) F/km\n",cn
    *10^(12));

xcb=1/(2*(%pi)*50*cn*1000);
printf(" the reactance is: %.2f*10^(5) Ohm/km\n",xcb
    *.00001);
```

Scilab code Exa 7.15 To calculate the voltage induced per km

```
clear;
//clc();
dab=1.2;
d12=0.4;
db2=0.85;

db1=sqrt(0.8*0.8 + db2*db2);

da2=db1;

da1=sqrt(d12*d12 + db2*db2);

v=2*(%pi)*50*2*60*log([db1*db1/da1*da1]);
printf("\n the voltage induced per unit length is: %
    .3f V/km\n",v/10000);
```

Scilab code Exa 7.16 to calculate the voltage induced in telephone conductor due t

```
clear;
clc;

Db0=6*cosd(30);
Da0=3;
D01=5;
D12=.6;
Db1=Db0+D01
Db2=Db1+D12;
Da1=sqrt(Da0*Da0+D01*D01);
Dc1=Da1;
Dc2=sqrt(Da0*Da0+5.6*5.6);
Da2=Dc2;
ia=200;
ib=200*(cosd(-120)+(%i)*sind(-120));
ic=200*(cosd(120)+(%i)*sind(120));
lam=2*10^(-7)*(ia*log([Da2/Da1])+ib*log([Db2/Db1])+
   ic*log([Dc2/Dc1]));
lamda=sqrt(real(lam)^2+imag(lam)^2);
v=2*(%pi)*50*lamda;
bc=6;
oc=3;
b0=sqrt(bc*bc-oc*oc);
hc=15;
hb=b0+hc;
ha=hc;
r=19.53/2000;
va=132000/sqrt(3);
v1a=(log([(2*ha-Da1)/Da1])/log([(2*ha-r)/r]));
v1b=(log([(2*hb-Db1)/Db1])/log([(2*hb-r)/r]));
v1c=(log([(2*hc-Dc1)/Dc1])/log([(2*hc-r)/r]));
vb=v1b*(cosd(-120)+(%i)*sind(-120));
vc=v1c*(cosd(120)+(%i)*sind(120));
v1=va*(v1a+vb+vc);
V1=sqrt(real(v1)^2+imag(v1)^2);
printf("The voltage induced is :%.2f kV",V1/1000);
```

Scilab code Exa 7.17 To calculate the inductance per unit length

```
clear;
clc();
dad=6;
dbe=dad;
dae1=sqrt(4*4 + 6*6);
dbd=dae1;
dce=dae1;

ded=sqrt(8*8 + 6*6);
dcd=ded;
dm=(dad*dae1*dbd*dbe*dcd*dce)^(1/6);

dbc=4;
dcb=dbc;
daa=0.7788*20/1000;
dbb=daa;
dcc=daa;

dab=4;
dbc=dab;
dba=dab;

dca=8;
dac=dca;

dsa=(daa*dab*dac*dba*dbb*dbc*dca*dcb*dcc)^(1/9);

la=2*10^(-7)*log([dm/dsa]);

ddd=0.7788*40/1000;
dee=ddd;
dde=4;
```

```
ded=dde;

dsb=(ddd*dde*dee*ded)^(1/4);

lb=2*10^(-7)*log([dm/dsb]);

l=100*(la+lb)/1000000;
printf("the inductance per unit length is: %.8f uH/m
    \n",l*10^(5));
```

Scilab code Exa 7.18 To calculate the loop inductance of the line

```
clear;
//clc();
dad=4;
dae1=4.3;
dbd=3.5;
dbe=3.8;
dcd=2;
dce=2.3;

dm=(dad*dae1*dbd*dbe*dcd*dce)^(1/6);

daa=0.7788*3/100;
dbb=daa;
dcc=daa;

dab=0.5;
dbc=dab;
dca=dab;
dba=dab;

dbc=1.5;
dcb=dbc;
dca=2;
```

```
dac=dca;

dsa=(daa*dab*dac*dba*dbb*dbc*dca*dcb*dcc)^(1/9);

la=2*log([dm/dsa]);

ddd=0.7788*4/100;
dee=ddd;
dde=0.3;
ded=dde;

dsb=(ddd*dde*dee*ded)^(1/4);

lb=2*log([dm/dsb]);

l=100*(la+lb);
printf("the loop inductance is: %.3f mH\n",l/10000);
```

Chapter 8

per unit representation

Scilab code Exa 8.1 `To calculate the per unit impedance and admittance`

```
clear;
//clc();
z=complex(4,60);
mva_base=100;
kv_base=230;
zpu=z*mva_base/(kv_base^2);
printf('the per unit impedance is:');
disp(zpu)

ys=(%i)*(2*10^(-3));
ypu=ys*(kv_base^2)/mva_base;
printf("the per unit admittance is:");
disp(ypu);
```

Scilab code Exa 8.2 `Three phase apparent power in pu`

```
clear;
//clc();
```

```
sb=50000;
s=40000;
pu=s/sb;
printf("\n the pu value for the three phase is: %.2f \n",pu);

sb1=sb/(3);
v=110;
vb=v/sqrt(3);
pukva=s/(3*sb1);
printf("\n the pu kva is: %.2f\n",pukva);
```

Scilab code Exa 8.3 determine the total reactance in per unit

```
clear;
//clc();
sb=5000
x1=2.5;
x2=2.5;
n1=400;
n2=200;
x1e=x1+x2*(n1/n2)^2;

vb1=400;
x1epu=x1e*sb/vb1^2;
printf("\n Total per unit resistnce refered to primary is: %.2f",x1epu);

x2e=x1+x2*(n2/n1)^2;
vb2=200;
x2epu=x2e*sb/vb2^2;
printf("\n Total per unit resistnce refered to secondary is: %.2f",x2epu);

//c)if the primary voltage is used as base
```

```
xpu1=x1*sb/vb1^2;
printf(" \n Primary per unit reactance is: %.2f ",
    xpu1);

xpu2=x1*sb/vb2^2;
printf("\n Seconadary per unit reactnace is: %.2f
    Ohm",xpu2);
```

Scilab code Exa 8.4 to calculate per unit impedances at 15 kva base

```
clear;
clc;

x1_pu=.12;
sb2=15;
sb1=10;
xg2=8;
xg1=.12;
sbg2=5;
sbg1=100;
xtf=6;
sbtf=15;
xt=complex(4,60);
kvl=230;
x2_pu=x1_pu*(sb2/sb1);
xg_pu1=x2_pu*sbg1;
printf("\n The per unit reactance of generator 1 is:
    %.2f",xg_pu1);
xg_pu2=xg2*(sb2/sbg2);
printf("\n The per unit reactance of generator 2 is:
    %.2f",xg_pu2);
xtf_pu=xtf*(sb2/sbtf);
printf("\n The per unit reactance of transformer is:
    %.2f",xtf_pu);
```

```
xt_pu=xt*sb2/((kvl)^2);
printf("\n The per unit impedance of transmission
    line is:");
disp(xt_pu)
```

Scilab code Exa 8.5 To calculate the voltage drop in line per unit

```
clear;
//clc();
s3q=10*10^6;
vl=10.5*1000;
il=s3q/(sqrt(3)*vl);
iph=il;
zph=5;
printf("\n Voltage drop in the line is: %.2f V",zph*
    iph);

//b).using per unit method to soolve

sb=10;
vlb=11;
vpu=vl/vlb;
sb3q=12;
spu=s3q/sb3q;
ipu=spu/vpu;

zpu=zph*sb3q/vlb^2;
v=zpu*ipu*vlb/sqrt(3);
printf("\n voltage drop in the line per phase is: %
    .2f V",v);
```

Scilab code Exa 8.6 To calculte the through impedance

```
clear;
//clc();
function [zpu]=puz(z1pu,sb2,sb1,vb1,vb2)
    zpu=z1pu*(sb2/sb1)*(vb1/vb2)^2;
endfunction

previousprot = funcprot(0)
funcprot(0)


function [parallel]=para(z1,z2)
    parallel=z1*z2/(z1+z2);
endfunction

previousprot = funcprot(0)
funcprot(0)


z1pu=0.20;
sb2=25;
sb1=30;
vb1=11;
vb2=11;
zpu=puz(z1pu,sb2,sb1,vb1,vb2);
printf("\n the per unit reactance of the generator
   is:  %f",zpu);
xg1=zpu;

z1pu=0.25;
sb2=25;
sb1=25;
vb1=11;
vb2=11;
zpu=puz(z1pu,sb2,sb1,vb1,vb2);
xg2=zpu;
printf("\n the per unit reactance of generator 2 is:
      %f",zpu);

```

```
z1pu=0.20;
sb2=60;
sb1=30;
vb1=11;
vb2=11;
zpu=puz(z1pu,sb2,sb1,vb1,vb2);
xg11=zpu;
printf("\n the per unit reactnace of generator 1 is
    on 60MVA base:  %f",zpu);

z1pu=0.25;
sb2=60;
sb1=25;
vb1=11;
vb2=11;
zpu=puz(z1pu,sb2,sb1,vb1,vb2);
xg22=zpu;
printf("\n the per unit reactnace of generator 2 is
    on 60MVA base: %f",zpu);

//calcultaion of per unit impedance of transformer

z1pu=0.10;
sb2=25;
sb1=60;
vb1=11;
vb2=11;
zpu=puz(z1pu,sb2,sb1,vb1,vb2);
xt1=zpu;
printf("\n the per unit reactnace of generator 1 is
    on 30MVA base:  %.3f",zpu);

z1pu=0.10;
sb2=60;
sb1=60;
vb1=11;
vb2=11;
zpu=puz(z1pu,sb2,sb1,vb1,vb2);
```

```
xt2=zpu;
printf("\n the per unit reactnace of generator 1 is
   on 60MVA base:  %.3f",zpu);

//calculation of through impedance

zt=para(xg1,xg2) + xt1;
printf("\n the through impedance at 25MVA base is:
   j%.3f",zt);

zt1=para(xg11,xg22)+xt2;
printf("\n the through impedance at 60MVA base is:
   j%.3f",zt1);

//calcultaion in ohms
sb1=25
zb=vb1^2/sb1;
printf("\n actual impedance in ohms on 25MVA base is
   : j%.3f",zb*zt);

sb2=60;
zb=vb1^2/sb2;
printf("\n actual impedance in ohms on 60MVA base is
   : j%.3f",zb*zt1);
```

Scilab code Exa 8.7 `to determine the per phase generator voltage`

```
clear;
clc;

sb=100;
vb=15;
xg=.75;
sbg=75;
xtf=.1
```

```
sbtf=50;
xt=100;
kvl=220;
rl=500;
vl=210;
xg_pu=xg*(sb/sbg);
xtf_pu=xtf*(sb/sbtf);
xt_pu=xt*sb/((kvl)^2);
rl_pu=rl*sb/((kvl)^2);
vpu=vl/kvl
i_pu=vpu/rl_pu;
v=i_pu*(rl_pu+(%i)*(xg_pu+xt_pu+xt_pu));
vg=round(sqrt(real(v)^2+imag(v)^2)*vb);
printf("The terminal voltage per phase is: %.2f kV",
   vg/sqrt(3));

//difference in answer is due to rounding off
```

Scilab code Exa 8.8 to calculate the per unit impedance of all units

```
clear;
clc;

sb=50;
kvl_hv=132;
kvl_lv=11;
blv=15;
btf2=180;
bkvl=180;
n=33/76;
z1pu=(%i)*.1;
z2pu=(%i)*.12;
kvlb1=11;
kvlb2=15;
kvl2b1=33;
```

```
kvl2b2=45.1;
zt=complex(25,75);
zm1=(%i)*.15;
zm2=(%i)*.15;
sm1=30;
sm2=20;
kvlm=30
bt=blv*kvl_hv/kvl_lv;
vm=btf2*n/sqrt(3);
putf1=z1pu*(kvlb1/kvlb2)^2;
printf("\n The per unit reacance of transformer 1 is
    : ");
disp(putf1);
sbtf2=3*20;
putf2=z2pu*(sb/sbtf2)*(kvl2b1/kvl2b2)^2;
printf("\n The per unit reacance of transformer 2 is
    : ");
disp(putf2);
pum1=zm1*(sb/sm1)*(kvlm/kvl2b2)^2;
printf("\n The per unit reacance of motor 1 is: ");
disp(pum1);
pum2=zm2*(sb/sm2)*(kvlm/kvl2b2)^2;
printf("\n The per unit reacance of motor 2 is: ");
disp(pum2);
put=zt*sb/(bkvl)^2;
printf("\n The per unit impedance of transmission
    line is: ");
disp(put);
```

Scilab code Exa 8.9 To calculte the volatge at the terminals of the motor

```
clear;
//clc();
ip1=24;
ip2=16;
```

```
sb=50;
tot_ip=ip1+ip2;
p=tot_ip/sb;
v=30;
vb=45.1;
vpu=v/vb;
ipu=p/vpu;

zt1pu=0.0537;
zlpu=0.0385+(%i)*0.1157;
zt2pu=0.0535;
vg=vpu+((%i)*zt1pu +zlpu+(%i)*zt2pu)*ipu;

function [mag,theta]=c(r,i)
    mag=sqrt(r*r + i*i)
    theta=atand(i/r)
endfunction


previousprot = funcprot(0)
funcprot(0)

r=real(vg);
i=imag(vg);
[mag,theta]=c(r,i);
vt=mag*15;
printf("\n the terminal voltage at the generator is:
    %.2f kV",vt);
```

Scilab code Exa 8.10 To find the generator bus terminal voltage

```
clear;
//clc();
function [zpu]=puz(z1pu,sb2,sb1,vb1,vb2)
    zpu=z1pu*(sb2/sb1)*(vb1/vb2)^2;
```

```
endfunction

previousprot = funcprot(0)
funcprot(0)

z1pu=0.15;
sb2=20;
sb1=20;
vb1=11;
vb2=12.5;
zpu=puz(z1pu,sb2,sb1,vb1,vb2);
printf("\n the per unit reactance of the generator
    is: %.2f",zpu);
xg=zpu;

zpu=puz(0.15,20,15,11,11);
printf("\n the per unit reactance of the generator
    is: %.2f",zpu);
xm=zpu;

zpu=puz(0.10,20,25,132,132);
printf("\n the per unit reactance of the generator
    is: %.2f",zpu);
xt1=zpu;

zpu=puz(0.10,20,20,132,132);
printf("\n the per unit reactance of the generator
    is: %.2f",zpu);
xt2=zpu;

xtl=((200+(%i)*500))*20/(132)^2;

pl=5;
pf=0.8;
p=pl*pf;
q=pl*0.6;

vpu=1;
```

```
sb=20;
rpu=vpu^2*sb*p/(p*p+q*q);
xpu=vpu^2*sb*q/(p*p+q*q);

//when pf is changed to 0.9

vpu=1.1;
im=15*(0.9+(%i)*sqrt(1-0.9^2))/(sb*vpu);

il=(p-(%i)*q)/(sb*vpu);
i=im+il;
zt=(%i)*(xt1+xt2)+xtl;

vg=vpu+zt*i;

function [mag,theta]=c(r,i)
    mag=sqrt(r*r + i*i)
    theta=atand(i/r)
endfunction


previousprot = funcprot(0)
funcprot(0)

[mag,theta]=c(real(vg),imag(vg));
printf("\nthe terminal generator voltage is: %.2f kV
   ",mag*11);
```

Chapter 9

Short and medium lines

Scilab code Exa 9.1 calculate the current voltage and power factor of the load

```
clear;
//clc();
//a).unity power factor

s=200;
vr=2500;
r=1.4;
x=0.8;
i=s*1000/vr;
z=r+(%i)*x;
vs=vr+z*i;
qs=atand(imag(vs)/real(vs));
pf=cosd(qs);
printf("the power factor of the sending end is:%.4f\
   n",pf);

//b).load power factor =0.8

pfl=acosd(0.8);
vs=vr+z*i*(cosd(-pfl)+(%i)*sind(-pfl));
qs=atand(imag(vs)/real(vs));
```

```
pf1=qs-(-pfl);       //negative sign is due to the loadis lagging
pf=cosd(pf1);
printf(" the power factor of the sending end is:%.3f\n",pf);

//c).load factor is 0.8 leading

pfl=acosd(0.8);
vs=vr+z*i*(cosd(pfl)+(%i)*sind(pfl));
qs=atand(imag(vs)/real(vs));
pf1=qs-(pfl);       //negative sign is due to the loadis lagging
pf=cosd(pf1);
printf(" the power factor of the sending end is:%.3f\n",pf);
```

Scilab code Exa 9.2 calculate power factor voltage regulation and efficiency

```
clear;
//clc();
s=15000/3;
v=33000/sqrt(3);
pf=0.85;
l=8;
r=.29*l;
x=0.65*l;
i=s*1000/v;
qs=acosd(pf);
op=3*s*1000*pf;
ploss=3*i*i*r;

z=r+(%i)*x;
vs=v+z*i*(cosd(-qs)+(%i)*sind(-qs));
vsp=sqrt(real(vs)^2+imag(vs)^2);
```

```
vsl=sqrt(3)*vsp;
printf("\n the line voltage at the sending end is: %
    .2f kv",vsl/1000);

//b)phase difference
qs1=atand(imag(vs)/real(vs));
pf1=qs1-(-qs);      //negative sign is due to the
    load is lagging
pf=cosd(pf1);
printf("\n the power factor of the sending end is:%
    .4f",pf);

//c).line regulation

lr=(vsp-v)/v;
printf("\n the line regulation of is:%.2f",lr);

//d).efficiency

n=op/(op+ploss);
printf("\n the transission efficiency is: %.2f
    percent",n*100);
```

Scilab code Exa 9.3 calculate voltage and power factor

```
clear;
//clc();
l=1000000;       //length in cm
s=5000000;
v=11000/sqrt(3);
f=50;
pf=0.8;
rho=.000001774;
i=s/(3*v*pf);
rp=0.1*s/(3*i*i);
```

```
a=rho*l/rp;
r=sqrt(a/%pi);
r1=0.7788*r;
d=200;
L=2*.001*log([d/r1]);

xlp=2*(%pi)*f*L;
qs=acosd(pf);
z=rp+(%i)*xlp;
vs=v+z*i*(cosd(-qs)+(%i)*sind(-qs));
vsp=sqrt(real(vs)^2+imag(vs)^2);
vsl=sqrt(3)*vsp;
printf("\n the line voltage at the sending end is: %.2f kv",vsl/1000);

qs1=atand(imag(vs)/real(vs));
pf1=qs1-(-qs);       //negative sign is due to the load is lagging
pf=cosd(pf1);
printf("\n the power factor of the sending end is:%.2f (lagging)",pf);
```

Scilab code Exa 9.4 `determine the voltage and power factor`

```
clear;
//clc();
a=sqrt(3);
r1=5;
r2=1.5;
pf=0.8;
rehv=r1+(a)*(a)*r2;
x1=10;
x2=3;
xehv=x1+(a)*(a)*x2;
rl=2;        //resistance of the line
```

```
xl=3;        //recatance of the line
r=rl+rehv;
x=xl+xehv;
s=3000;
v=33000;
i=s*1000/(v*sqrt(3));
vr=v/sqrt(3);
qs=acosd(pf);
z=r+(%i)*x;
vs=vr+z*i*(cosd(-qs)+(%i)*sind(-qs));
vsp=sqrt(real(vs)^2+imag(vs)^2);
vsl=sqrt(3)*vsp;
printf("\n the line voltage at the sending end is: %
    .2f kv",vsl/1000);

qs1=atand(imag(vs)/real(vs));
pf1=qs1-(-qs);        //negative sign is due to the
    load is lagging
pf=cosd(pf1);
printf("\n the power factor of the sending end is:%
    .3f",pf);
```

Scilab code Exa 9.5 the voltage at the generator busbars

```
clear;
//clc();

s=6000000;
v=6000;
pf=.8;
a=66/330;
r=7;
x=2;
rt=2;
xt=18;
```

```
totr=r+rt;
totx=x+xt;
vhv=v/a;
il=s/(sqrt(3)*vhv);
vph=vhv/sqrt(3);
vs=sqrt((il*totr+vph*pf)^2+(il*totx+vph*.6)^2);
vl=vs*sqrt(3);
vlv=vl*a;
printf("the volatge at the genertor busbars is:%0.3f
    V",vlv);
```

Scilab code Exa 9.6 determine voltage current power factor apparent power efficien

```
clear;
clc();

function [r,i]=d(mag,theta)
    r=mag*cosd(theta);
    i=mag*sind(theta);

endfunction



previousprot = funcprot(0)
funcprot(0)

vr=60000;
pr=36000000;
p=(12*10^6);
pf=.8;
mag=p/(vr*pf);
theta=acosd(pf);

[r,i]=d(mag,theta);
```

```
ir=conj(complex(r,i));

f=50;
l=.1;
c=.25*10^(-6);
r=2.5;
xl=2*(%pi)*f*l;
z=r+(%i)*xl;
y=(%i)*2*(%pi)*f*c;
//calculations by nominal T model

vab=vr+.5*z*ir;
iab=y*vab;
is=ir+iab;
printf("\n The current in A at the sending end is:");
disp(is);
qi=atand(imag(is)/real(is));

//voltage drop in the left hand half of the line
vd=is*z/2;
vs=vab+is*z/2;
printf("\n The volatge  in V at the sending end is:");
disp(vs);
vl=sqrt(3)*vs;
qs=atand(imag(vs)/real(vs))-qi;
pfs=cosd(qs);
printf("\n The lagging power factor at the sending end is:");
disp(pfs);
s3=3*vs*conj(is);
printf("\n The apparent power  in VA at the sending end is:");
disp(s3);
eff=pr/real(s3);
printf("\n The transmission efficiency is: %.4f per cent",eff*100);
```

```

A=1+.5*z*y;
r=real(A);
i=imag(A);
function [mag,theta]=c(r,i)
    mag=sqrt(r*r + i*i)
    theta=atand(i/r)
endfunction
previousprot = funcprot(0)
funcprot(0)

[mag,theta]=c(r,i);
magA=mag;
r=real(vs);
i=imag(vs);
[mag,theta]=c(r,i);

magV=mag;

vrnl=magV/magA;
reg=(vrnl-vr)/vr;
printf("\n The regulation is:%.4f per cent",reg*100)
    ;

//calculations based on pi model
printf("\n \n Calculations based on pi model:");
iab=y*vr/2;
i=ir+iab;
vd=i*z;
vs=vr+vd;
printf("\n \nThe volatge  in V at the sending end is
    :");
disp(vs);
icd=y*vs/2
is=i+icd;
printf("\n The current in A at the sending end is:")
    ;
disp(is);
```

```
qis=atand(imag(vs)/real(vs))-atand(imag(is)/real(is));
pfs=cosd(qis);
printf("\n The lagging power factor at the sending end is:");
disp(pfs);
s3=3*vs*conj(is);
printf("\n The apparent power  in VA at the sending end is:");
disp(s3);
eff=pr/real(s3);
printf("\n The transmission efficiency is:%.2f per cent",eff*100);

A=1+.5*z*y;
r=real(A);
i=imag(A);
function [mag,theta]=c(r,i)
    mag=sqrt(r*r + i*i)
    theta=atand(i/r)
endfunction
previousprot = funcprot(0)
funcprot(0)

[mag,theta]=c(r,i);
magA=mag;
r=real(vs);
i=imag(vs);
[mag,theta]=c(r,i);

magV=mag;

vrnl=magV/magA;
reg=(vrnl-vr)/vr;
printf("\n The regulation is:%.4f per cent",reg*100);
```

Scilab code Exa 9.7 calculate receiving end load

```
clear;
clc;

f=50;
l=150;
L=.2;
C=1.5*10^(-6);
vr=110000/sqrt(3);
vs=vr;
xl=2*(%pi)*f*L;
z=(%i)*xl;
y=(%i)*2*(%pi)*f*C;
iab=y*vr/2;
//i=ir+iab
//using the forumla vs=vr+i*z

ir=sqrt(vs*vs-(vr+iab*z)^2)/imag(z);
printf("The load current is:%.3f A",ir);
```

Scilab code Exa 9.8 to calculate A B C D constants sending end voltage current pow

```
clear;
clc;

vl=132000;
s=50000000;
pf=.85;
l=80;

function [r,i]=d(mag,theta)
```

```
    r=mag*cosd(theta);
    i=mag*sind(theta);

endfunction



previousprot = funcprot(0)
funcprot(0)

mag=96;
theta=78;
[r,i]=d(mag,theta);
z=complex(r,i);

mag=.001;
theta=90;
[r,i]=d(mag,theta);
y=complex(r,i);

vrp=vl/sqrt(3);
Irp=s/(sqrt(3)*vl*pf);

mag=Irp;
theta=-acosd(pf);
[r,i]=d(mag,theta);
irp=complex(r,i);

//a).for the nominal T network parameters are
A=1+.5*z*y;
B=z*(1+.25*z*y);
C=y;
D=A;

disp(A);
disp(B);
disp(C);
disp(D);
```

```

//phase voltage at the sending end is
vsp=A*vrp+B*irp;
vsl=sqrt(3)*vsp;
disp(vsp);

//c).
is=C*vrp+D*irp;
disp(is);

//d).
qs=atand(imag(vsp)/real(vsp))-atand(imag(is)/real(is));
printf("\n The power factor at the sending end is:%.3f (lagging)",cosd(qs));

//e).
r=real(vsl);
i=imag(vsl);
function [mag,theta]=c(r,i)
    mag=sqrt(r*r + i*i)
    theta=atand(i/r)
endfunction
previousprot = funcprot(0)
funcprot(0)

[mag,theta]=c(r,i);
Vsl=mag;

r=real(is);
i=imag(is);
[mag,theta]=c(r,i);

Is=mag;
eff=s/(sqrt(3)*Vsl*Is*cosd(qs));
printf("\n The efficiency of transmission is:%.2f per cent",eff*100);
```

Chapter 10

Long transmission lines

Scilab code Exa 10.1 To calculate the A B C D constants

```
clear;
//clc();
z=12.5 + (%i)*66;
y=(%i)*4.4*(10^(-4));
yS=sqrt(z*y);

aS=real(yS);

bS=imag(yS);

A=cosh(yS);

printf("\n the constant A is:  %f\n ",A);

D=A;

printf("\n the constant D is:  %f\n ",D);

zo=sqrt(z/y);
B=zo*sinh(yS);

```

```
printf("\n the constant B is:  %f\n ",B);

C=sinh(yS)/(zo);

printf("\n the constant C is:  %f\n ",C);
```

Scilab code Exa 10.2 To calculate the sending end voltage sending end current line

```
clear;
//clc();
v=275*1000;
vrp=v/sqrt(3);
p=250*10^(6);
cosq=0.9;
ir1=p/(sqrt(3)*v*cosq);
ir=ir1*(cosd(25.84) + (%i)*sind(-25.84));

A=0.9855138 + 0.0027367*(%i);
D=0.9855138 + 0.0027367*(%i);
C=- 0.0000004 + 0.0004379*(%i);
B=12.37926 + 65.692432*(%i);

is=C*vrp + D*ir;
printf('the sending end current is:');
disp(is);
vsp=A*vrp + B*ir;
printf('the sending end voltage is:');
disp(vsp);
ps=real(3*vsp*conj(is));

tx_eff=p/ps;
printf("\n the transmission efficiency is:  %.3f pu\n ",tx_eff);

vnl=polar(vsp)/A;
```

```
vfl=158.77*1000;
vr=(vnl-vfl)/vfl;
printf("\n the voltage regulation is:  %.3f pu\n ",
   vr);
```

Chapter 11

General networks constants

Scilab code Exa 11.2 To find the volatge current and pf at the sending end

```
clear;
clc;
function [r,i]=d(mag,theta)
    r=mag*cosd(theta);
    i=mag*sind(theta);

endfunction

previousprot = funcprot(0)
funcprot(0)

//clc();
mag=0.98;
theta=3;
[r,i]=d(mag,theta);
A=complex(r,i);

D=A;

mag=110;
theta=75;
```

```
[r,i]=d(mag,theta);
B=complex(r,i);

mag=0.0005;
theta=88;
[r,i]=d(mag,theta);
C=complex(r,i);

v=132*10^(3);
s=50*10^(6);
pf=0.8;

vrp=v/sqrt(3);
mag=vrp;
theta=0;
[r,i]=d(mag,theta);
vr=complex(r,i);

i=s/(sqrt(3)*v);
mag=i;
theta=-acosd(pf);
[r,i]=d(mag,theta);
ir=complex(r,i);

vsp=A*vr + B*ir;
printf('the voltage at the sending end is:');
disp(vsp);
is=C*vr + D*ir;
printf('the current at the sending end is:');
disp(is);
qs=atand(imag(vsp)/real(vsp))-atand(imag(is)/real(is
   ));
printf("The power factor is:%.4f",cosd(qs));
```

Scilab code Exa 11.3 To calculate sending end voltage

```
clear;
//clc();

function [r,i]=d(mag,theta)
    r=mag*cosd(theta);
    i=mag*sind(theta);

endfunction

previousprot = funcprot(0)
funcprot(0)

mag=0.97;
theta=0.6;
[r,i]=d(mag,theta);
a1=complex(r,i);


mag=60;
theta=70;
[r,i]=d(mag,theta);
b1=complex(r,i);


mag=0.97;
theta=0.4;
[r,i]=d(mag,theta);
a2=complex(r,i);

mag=50;
theta=76;
[r,i]=d(mag,theta);
b2=complex(r,i);

s=50*10^(6);
v=132*10^(3);
pf=0.8;

```

```

vrp=v/sqrt(3);
mag=vrp;
theta=0;
[r,i]=d(mag,theta);
vr=complex(r,i);

i=s/(sqrt(3)*v*pf);
mag=i;
theta=-acosd(pf);
[r,i]=d(mag,theta);
ir=complex(r,i);

A=(a1*b2 + a2*b1)/(b1 + b2);

B=(b1*b2)/(b1+b2);

vsp=A*vr + B*ir;
printf('the sending end phase voltage is:');
disp(vsp);
```

Scilab code Exa 11.4 To calculate the value of ABCD parameters and the characteris

```
clear;
//clc();
function [r,i]=d(mag,theta)
    r=mag*cosd(theta);
    i=mag*sind(theta);

endfunction

previousprot = funcprot(0)
funcprot(0)

mag=120;
```

```
theta=60;
[r,i]=d(mag,theta);
z=complex(r,i);

mag=5*10^(-3);
theta=90;
[r,i]=d(mag,theta);
y=complex(r,i);
A=1 + 0.5*z*y;
printf('the value of A is:');
disp(A)
B=z;
printf('the value of B is:');
disp(B)
C=y*(1+ 0.25*(z)*(y));
printf('the value of C is:');
disp(C)
D=A;
printf('the value of D is:');
disp(D)
```

Scilab code Exa 11.5 To determine the equivalent T network

```
//to determine the equivalent t-network

//clc();
rab=720;
rac_oc=1240;
rcd=910;
r3=sqrt((rac_oc-rab)*(rcd));
printf("\n the R3 resistance is:  %f ohm\n",r3);

r2=rcd-r3;
printf("\n the R2 resistance is:  %f ohm\n",r2);

```

```
r1=rac_oc-r3;
printf("\n the R1 resistance is:  %f ohm\n",r1);
```

Chapter 13

Control of voltage and reactive power

Scilab code Exa 13.1 To calculate the maximum power phase differenec for maximum p

```
clear;
clear;
 //clc();


 function [mag,theta]=c(r,i)
    mag=sqrt(r*r + i*i)
    theta=atand(i/r)
endfunction


previousprot = funcprot(0)
funcprot(0)

 r=10;
 x=30;
 i=x;
 vs=132;
 vr=vs;
```

```
 z=r + (%i)*x;
 B=z;
 A=1;
 b=atand(x/r);
 a=0;

 [mag,theta]=c(r,i);
 pm=(vs*vr/mag) - ((A*vr^(2)/mag)*cosd(theta));
 printf("\n the maximum power is:  %.2f MW\n",pm);

//maximum power is at an angle
del=theta;
printf("\n the maximum power angle is:  %.2f degrees
   \n",del);

//determining the rating of the syncronous phase
   modifier

pl=100;
pr0=-(vr^2)*r/mag^(2);

qr0=-(vr^(2)*x)/mag^(2);

pr=vs*vr/mag;

q=sqrt(pr^(2) - (pr0 - pl)^(2)) +  qr0;

qpm=pl*tand(acosd(0.9)) - q;
printf("\n the rating of the phase modifier is:  %.2
   f (leading)",qpm);
```

Scilab code Exa 13.2 To find the rating of the modifier and the power factor

```
clear;
//clc();
```

```
function [mag,theta]=c(r,i)
    mag=sqrt(r*r + i*i)
    theta=atand(i/r)

endfunction

previousprot = funcprot(0)
funcprot(0)

r=25;
x=90;
i=x;
[mag,theta]=c(r,i);
vs=145;
 vr=132;

pl=0;
p=50;
pr0=-(vr^2)*r/mag^(2);

qr0=-(vr^(2)*x)/mag^(2);

pr=vs*vr/mag;

q=sqrt(pr^(2) - (pr0 - pl)^(2)) +  qr0;

qpm=q;
printf("\n the rating of the synchronous phase
   modifier is: %.2f MVAr\n",qpm);
phi=atand(qpm/p);
printf("\n the pf is: %.2f (lagging)\n",phi);
```

Scilab code Exa 13.3 To calculate the sending voltage maximum power and additional

```
clear;
```

```
 //clc();


 function [mag,theta]=c(r,i)
    mag=sqrt(r*r + i*i)
    theta=atand(i/r)
endfunction


previousprot = funcprot(0)
funcprot(0)


function [r,i]=d(mag,theta)
    r=mag*cosd(theta);
    i=mag*sind(theta);

endfunction



previousprot = funcprot(0)
funcprot(0)

vr=275;
 vrl=vr*1000;
 pf=.9;
 vrp=vrl/sqrt(3);
 vs=290;

magA=.94;
mag=magA;
thetaA=1;
theta=thetaA;
[r,i]=d(mag,theta);
A=complex(r,i);

magB=107;
```

```
mag=magB;
thetaB=78;
theta=thetaB;
[r,i]=d(mag,theta);
B=complex(r,i);
pr=300*10^6;
ir=pr/(sqrt(3)*vrl*pf);
mag=ir;
theta=-acosd(0.9);
 [r,i]=d(mag,theta);
ir=complex(r,i);
vsp=A*vrp+B*ir;
vsl=sqrt(3)*vsp;
printf("The line voltage at the sending end:");
disp(vsl)
pm=(vs*vr/magB) - ((magA*vr^(2)/magB)*cosd(thetaB-thetaA));
 printf("\n the maximum power is: %.2f MW\n",pm);

//determining the rating of the syncronous phase modifier

pl=450;
pr=pl*pf;
pr0=-(vr^2)*magA*cosd(thetaB-thetaA)/magB;
qr0=-(vr^2)*magA*sind(thetaB-thetaA)/magB;
rhor=vs*vr/magB;
q=sqrt(rhor^(2) - (pr - pr0)^(2)) +  qr0;

qpm=pr*tand(acosd(0.9)) - q;
printf("\n the rating of the phase modifier is: %.2f MVAr",qpm);
```

Scilab code Exa 13.4 To find the mva rating of synchronous phase modifier

```
clear;
//clc();

r=14;
x=48;
i=x;
vs=70;
vr=66;

 function [mag,theta]=c(r,i)
    mag=sqrt(r*r + i*i)
    theta=atand(i/r)
endfunction


previousprot = funcprot(0)
funcprot(0)

[mag,theta]=c(r,i);

z=r + (%i)*x;
y=(%i)*4*10^(-4);

A=1 + 0.5*z*y;

[mag,theta]=c(real(A),imag(A));
a=theta;
A1=mag;

B=z;

function [mag,theta]=c(r,i)
    mag=sqrt(r*r + i*i)
    theta=atand(i/r)
endfunction

[mag,theta]= c(r,i);
b=theta;
```

```
B1=mag;

pr0=-(A1*vr^2)*cosd(b-a)/(B1);

qr0=-(A1*vr^2)*sind(b-a)/(B1);

pr=vs*vr/B1;

pl=0;

q=sqrt(pr^(2) - (pr0 - pl)^(2)) +  qr0;

qpm=q;

s=24;
prat=s*0.8;

q=sqrt(pr^(2) - (pr0 - prat)^(2)) +  qr0;

qpmrat=prat*tand(acosd(0.8))-q;
printf("\n the rating of the synchronous phase modifier is:  %.2f MVAr\n",qpmrat);
```

Chapter 14

Load flow analysis

Scilab code Exa 14.1 determine Ybus

```
clear;
clc;

n=5;
m=5;
y=zeros(n,m);
z12=(%i)*.2;
z23=(%i)*.25;
z34=(%i)*.3
z45=(%i)*.25;
z14=(%i)*.5;
z15=(%i)*.2;
y(1,2)=1/z12;
y(2,3)=1/z23;
y(3,4)=1/z34;
y(4,5)=1/z45;
y(1,4)=1/z14;
y(1,5)=1/z15;


for i=1:1:n
```

```
  for j=1:1:m
      y(j,i)=(y(i,j)+y(i,j))/2;

      end
end
Y=y;
for i=1:1:n
   for j=1:1:m
       if (i==j) then
           for k=1:1:n
           y(i,i)=y(i,i)+Y(j,k);
           end
       else
           y(i,j)=y(i,j)*(-1);
       end
   end
end
disp(y)
```

Scilab code Exa 14.2 determine modified Ybus

```
clear;
clc;

y= [- 12*(%i) (%i)*5 0 (%i)*2 (%i)*5;5*(%i) -(%i)*9
   (%i)*4 0 0;0 4*(%i) -(%i)*7.33 (%i)*3.33 0;(%i)*2
    0 3.33*(%i) -(%i)*9.33 (%i)*4;5*(%i) 0 0 (%i)*4
   -(%i)*9]
Y=y;
z25=(%i)*.4;
i=2;
j=5;
y(i,j)=1/z25;
y(i,i)=Y(i,i)+y(i,j);
y(j,j)=Y(j,j)+y(i,j);
```

```
y(i,j)=-y(i,j);
y(j,i)=y(i,j);
disp(y)
```

Scilab code Exa 14.3 `determine Ybus`

```
clear;
clc;

y= [- 12*(%i) (%i)*5 0 (%i)*2 (%i)*5;5*(%i) -(%i)*9 (%i)*4 0 0;0 4*(%i) -(%i)*7.33 (%i)*3.33 0;(%i)*2 0 3.33*(%i) -(%i)*9.33 (%i)*4;5*(%i) 0 0 (%i)*4 -(%i)*9];
Y=y;
i=3;
j=5;
k=1;
l=3;
z35=(%i)*.05;
z13=(%i)*.01;
y(k,l)=1/z13;
y(i,j)=1/z35;
y(k,k)=Y(k,k)+y(k,i);
y(i,i)=Y(i,i)+y(i,j)+y(k,i);
y(j,j)=Y(j,j)+y(l,j);
y(i,j)=-y(i,j);
y(j,i)=y(i,j);
y(k,l)=-y(k,l);
y(l,k)=y(k,l);
disp(y)
```

Scilab code Exa 14.4 `determine Ybus`

```
clear;
clc;

z12=.06+(%i)*.18;
z23=.08+(%i)*.24;
z13=.03+(%i)*.09;

n=3;
m=3;
y=zeros(n,m);
y(1,2)=1/z12;
y(1,3)=1/z13;
y(2,3)=1/z23;

for i=1:1:n
  for j=1:1:m
      y(j,i)=(y(i,j)+y(i,j))/2;

      end
end
Y=y;
for i=1:1:n
   for j=1:1:m
       if (i==j) then
           for k=1:1:n
           y(i,i)=y(i,i)+Y(j,k);
           end
       else
           y(i,j)=y(i,j)*(-1);
       end
   end
end

disp(y)
```

Scilab code Exa 14.5 determine modified Ybus

```
clear;
clc;


y=[(5-15*(%i)) (-1.67+5*(%i)) (-3.33+10*(%i))
    ;(-1.67+5*(%i)) (2.91-8.75*(%i)) (-1.25+3.75*(%i)
    );(- 3.33+10*(%i)) (- 1.25 + 3.75*(%i))
    (4.58-13.75*(%i))];
Y=y;
ya=-2.5*(%i);
yb=ya;
yc=ya;
yd=ya;
ye=ya;
yf=ya;
i=1;
j=2;
k=3;
y(k,k)=Y(k,k)+yb+yf;
y(i,i)=Y(i,i)+ya+yc;
y(j,j)=Y(j,j)+ye+yd;
disp(y)
```

Scilab code Exa 14.6 determine Ybus

```
clear;
clc;

n=4;
m=4;
y=zeros(n,m);
z12=.25+(%i)*1.0;
z13=.20+(%i)*.8;
```

```
z14=.30+(%i)*1.2;
z23=.20+(%i)*.8;
z34=.15+(%i)*.6;
y(1,2)=1/z12;
y(1,3)=1/z13;
y(1,4)=1/z14;
y(2,3)=1/z23;
y(3,4)=1/z34;



for i=1:1:n
  for j=1:1:m
      y(j,i)=(y(i,j)+y(i,j))/2;

      end
end
Y=y;
for i=1:1:n
   for j=1:1:m
       if (i==j) then
           for k=1:1:n
           y(i,i)=y(i,i)+Y(j,k);
           end
       else
           y(i,j)=y(i,j)*(-1);
       end
   end
end
disp(y)
```

Scilab code Exa 14.7 determine modified Ybus

```
clear;
clc;
```

```
n=4;
m=4;
y=zeros(n,m);
z12=.25+(%i)*1.0;
z13=.20+(%i)*.8;
z14=.30+(%i)*1.2;
z23=.20+(%i)*.8;
z34=.15+(%i)*.6;
y(1,2)=1/z12;
y(1,3)=1/z13;
y(1,4)=1/z14;
y(2,3)=1/z23;
y(3,4)=1/z34;



for i=1:1:n
  for j=1:1:m
      y(j,i)=(y(i,j)+y(i,j))/2;

      end
end
Y=y;
for i=1:1:n
   for j=1:1:m
       if (i==j) then
           for k=1:1:n
           y(i,i)=y(i,i)+Y(j,k);
           end
       else
           y(i,j)=y(i,j)*(-1);
       end
   end
end

ya=-.08*(%i);
yb=ya;
```

```
yc=ya;
yd=ya;
yk=ya;
yf=ya;
i=1;
j=2;
k=3;
l=4;
y(k,k)=y(k,k)+yf;
y(i,i)=y(i,i)+ya+yc+yb;
y(j,j)=y(j,j)+yd;
y(l,l)=y(l,l)+yk;
disp(y)
```

Scilab code Exa 14.8 determine the voltages at buses

```
clear;
clc;


n=3;
m=3;
y=zeros(n,m);
y(1,2)=(-(%i)*3);
y(1,3)=(-(%i)*4);
y(2,3)=(-(%i)*5);

for i=1:1:n
  for j=1:1:m
      y(j,i)=(y(i,j)+y(i,j))/2;

      end
end
Y=y;
for i=1:1:n
```

```
    for j=1:1:m
        if (i==j) then
            for k=1:1:n
            y(i,i)=y(i,i)+Y(j,k);
            end
        else
            y(i,j)=y(i,j)*(-1);
        end
    end
end

v=[1.02 1 1];

alpha=1.6;

del=[0 0 0];
pl=[0 50 60];
ql=[0 25 30];
pg=[0 25 0];
qg=[0 15 0];
bmva=100;
p2=(pg(2)-pl(2))/bmva;
q2=(qg(2)-ql(2))/bmva;

p3=(pg(3)-pl(3))/bmva;
q3=(qg(3)-ql(3))/bmva;

p=[0 p2 p3];
q=[0 q2 q3];
for i=1:1:2
    v1=zeros(1,3);
    v2=v;
    for j=2:1:3
        for k=1:1:3
            if (k~=j) then
                v1(j)=v1(j)-y(j,k)*v(k);

            end
```

```
            end
            v(j)=(1/y(j,j))*((p(j)-(%i)*q(j))/conj(v(j))
                +v1(j));
            delv(j)=v(j)-v2(j);
            v(j)=v2(j)+alpha*delv(j);

      end
printf("The voltages at the end of:%d iteration are"
    ,i);
            disp(v)
end
```

Chapter 15

Economic operations of power systems

Scilab code Exa 15.1 to find the incremental cost of two units

```
clear;
clc;

b=[1 -2 ;1 1];
c=[0;450];
a=inv(b)*c;
p1=a(1);
p2=a(2);
ic1=2+.01*p1;
ic2=2+.02*p2;
printf("the incremental fuel cost of first generator
    is:%.2f Rs/MWh",ic1);
printf("\n the incremental fuel cost of second
   generator is:%.2f Rs/MWh",ic2);
```

Scilab code Exa 15.2 to find the load division between the two units

```
clear;
clc;

p1max=200;
p2max=p1max;
p1min=50;
p2min=p1min;

p1=p1min;
p2=p1;
ic1=.16*p1+30;
ic2=.20*p2+25;
printf("For the plant output %.2f MW p1=%.2fMW    p2=
   %.2fMW\n",(p1+p2),p1,p2);

for ic=40:5:60
    ic1=ic;
    ic2=ic;

        p1=(ic1-30)/.16;
    p2=(ic2-25)/.20;

    printf("For the plant output %.2f MW p1=%.2fMW
         p2=%.2fMW\n",(p1+p2),p1,p2);
    end

p1=200;
p2=200;
printf("For the plant output %.2f MW p1=%.2fMW    p2=
   %.2fMW\n",(p1+p2),p1,p2);
```

Scilab code Exa 15.3 determine the saving in fuel cost

```
clear;
clc;
```

```

function ic1=f(p1),ic1=.16*p1+30;
endfunction

function ic2=f1(p2),ic2=.20*p2+25;
endfunction
c1=intg(50,57.5,f);
c2=intg(65,57.5,f1);
printf("the increase in cost of unit 1 is:%.3f Rs/h\n",c1);
printf("the increase in cost of unit 2 is:%.3f Rs/h\n",c2);

net=c1+c2;
sav=net*365*24;
printf("The total yearly saving is:%.2f Rs",sav);
```

Scilab code Exa 15.4 find the loss coefficients and the transmission loss

```
clear;
clc;

zac=complex(.05,.2);
rac=real(zac);
vc=1;
i1=1.05;
i2=.9;
zbc=complex(.04,.16);
rbc=real(zbc);
zcd=complex(.03,.12);
rcd=real(zcd);
va=vc+zac*i1;
vb=vc+zbc*i2;
p1=real(va*conj(i1));
p2=real(vb*conj(i2));
```

```
b11=(rac+rcd)/(real(va)^2);
b12=rcd/(real(va)*real(vb));
b22=(rbc+rcd)/(real(vb)^2);
pl=p1*p1*b11+p2*p2*b22+2*p1*p2*b12;
printf("The transmission loss is:%.4f pu",pl);
```

Scilab code Exa 15.5 find the penalty factor

```
clear;
clc;

p1=200;
p2=p1;
a=.2;..// a=del(pl)/del(p2)
l2=1/(1-a);
ic1=.15*p1+150;
ic2=.25*p2+175;
l1=l2*ic2/ic1;
printf("The penalty factor of plant 1 is: %.4f",l1);
```

Scilab code Exa 15.6 find the penalty factor and the additional cost per hour to i

```
clear;
clc;

b11=.001;
b12=-.0001;
b22=.0013;

p1=150;
p2=275;
a=2*p1*b11+2*p2*b12;
l1=1/(1-a);..// a=del(pl)/del(p2)
```

```
printf("The penalty factor of plant 1 is: %.4f\n",l1);
lamda=200;
ic1=lamda/l1;
printf("The incremental cost is:%.0f Rs/MWh",ic1);
```

Scilab code Exa 15.7 calculate the penalty factors for the two plants

```
clear;
clc;

p1=100;
pl=5;
b11=pl/(p1*p1);
a=2*p1*b11;..// a=del(pl)/del(p1)
l1=1/(1-a);
a1=0;..// a1=del(pl)/del(p1)
l2=1/(1-a1);

printf("The penalty factor of plant 1 is: %.3f\n",l1);
printf("The penalty factor of plant 2 is: %.3f\n",l2);
```

Scilab code Exa 15.8 determine the generation schedule and the load demand

```
clear;
clc;

p1=125;
pl=12.5
b12=0;
b22=0;
```

```
b11=pl/(p1*p1);
lamda=70;
a=.000016;
p1=82.8729;..//ic1=.25*p1+40;
p2=100;
pl=.0008*p1*p1;
pr=p1+p2-pl;
printf("The total load is:%.4f MW\n",pr);
l1=1/(1-a);
a1=0;
l2=1/(1-a1);
p2=(lamda-50)/.2;
printf("for the optimal dispatch P1=%.2f MW\n",p1);
printf("for the optimal dispatch P2=%.2f MW",p2);..
    //ic2=.20*p2+50
```

Chapter 16

Symmetrical faults

Scilab code Exa 16.1 To find the high voltage terminals of a transformer

```
clear;
//clc();
s=15;
s1=10;
xg1=(%i)*0.1;
xg1_pu=xg1*s/s1;

xg2=(%i)*0.075;
s2=5;
xg2_pu=xg2*s/s2;

xt1_pu=(%i)*0.06;
z=5+(%i)*20;
vl=33;
z_pu=z*s/vl^2;
//printf("%z",z_pu);

//three phase fault Fa

x1=(%i)*0.15;
x2=(%i)*0.225;
```

```
ze_pu=xt1_pu+(x1*x2/(x1+x2));
Ssc=s/(imag(ze_pu));
ifault=Ssc*1000/(sqrt(3)*vl);
printf("The fault current is:%.2f A\n",ifault);

//three phase fault at phase b

xt=0.06887 + (%i)*0.27548;
ze_pu2=xt1_pu+(x1*x2/(x1+x2)) +xt;
z_pu=sqrt(real(ze_pu2)^2 + imag(ze_pu2)^2);
fb=s/z_pu;

ifault=fb*1000/(sqrt(3)*vl);
printf("The fault current is:%.2f A",ifault);
```

Scilab code Exa 16.2 to calculate the subtransient transient and synchronous short

```
clear;
//clc();
x2d=(%i)*0.3;
xl1=(%i)*0.08;
xline=(%i)*0.55;
xl2=(%i)*0.08;

ig=0.75;
z2t=x2d +xl1 +xline + xl2;
er=1;
eint=er+ig*z2t;
e2int=sqrt(real(eint)^2 + imag(eint)^2);

x2gf=imag(x2d + xl1);
i2d=e2int/x2gf;

x2bf=imag(xl1 +xline);
```

```
i2df=er/x2bf;
tot_i2d=i2d +i2df;
printf("The total subtransient short circuit current
    is:%.3f pu\n",tot_i2d);

//calculation of effect of maximum dc component
   offset

i2g=sqrt(2)*i2d;
i2f=sqrt(2)*i2df;
tot_i=i2g+i2f;
max_sc=sqrt(tot_i2d^2+tot_i^2);

sb=50;
vlb=138*10^(3);
ilb=sb/(sqrt(3)*vlb*(10^(-6)));
isc=ilb*max_sc;

//caculation of prefault voltage behind transient
   reactance

x1d=(%i)*0.35;
z1t=x1d+xl1+xline+xl2;
eint1=er+ig*z1t;
e1int=sqrt(real(eint1)^2 + imag(eint1)^2);

x1gf=imag(x1d + xl1);
i1d=e1int/x1gf;

x1bf=imag(xl1 +xline);
i1df=er/x1bf;
tot_i1d=i1d +i1df;
printf("The total transient short circuit current is
   :%.3f pu\n",tot_i1d);

isc1=ilb*tot_i1d;

//calculation of prefault voltage behind synchronous
```

```
reactance

xd=(%i)*1.25;
zt=xd+xl1+xline+xl2;

eint3=er+ig*zt;
e3int=sqrt(real(eint3)^2 + imag(eint3)^2);

x3gf=imag(xd + xl1);
i3d=e3int/x3gf;

x3bf=imag(xl1 +xline);
i3df=er/x3bf;
tot_i3d=i3d +i3df;
printf("The total synchronous short circuit current is:%.3f pu\n",tot_i3d);
```

Scilab code Exa 16.3 To calculte the reactance of the reactor to prevent the circu

```
clear;
//clc();
sb=50*10^6;
xg=0.2;
sg=10*10^6;

xgpu=sb*xg/sg;
xpg=1/4;
ssc=500;
xu=1/15;
vl=33*10^3;
ifl=sb/(sqrt(3)*vl);
vn=vl/sqrt(3);
x=vn*xu/ifl;
printf("the reactance of the reactor is:%.2f Ohm",x);
```

Scilab code Exa 16.4 To find the reactor neccessary to prevent the switchgear

```
clear;
//clc();

sb=7.5*10^6;
vl=3.3*10^3;
sga=3;
sgb=4.5;
xgb=0.08;
xga=0.07;
xga_pu=sb*xga/sga;
xgb_pu=sb*xgb/sgb;

xgp=(xga_pu*xgb_pu)/(xga_pu+xgb_pu);
xu=0.0724;

ifl=sb/(sqrt(3)*vl);
vn=vl/sqrt(3);
x=xu*vn/ifl;
printf("the reactance of the reactor is:%.3f A",x);
```

Scilab code Exa 16.5 to find the reactor reactances

```
clear;
clc;
//xt=(.075+1.5x)*(.15)/(.075+1.5x+.15);
ssc=200;
xpu=.15;
ifl=20*10^(6)/(sqrt(3)*11000);
vp=11000/sqrt(3);
```

```
x=xpu*vp/ifl;
printf("The ohmic reactance of each reactor is: %.2f ohm",x);

//x1=.0225/(.3+x);
//x2=x3=.15x/(.3+x);
//xt=.15(.5x+.075)/(.15+.5x+.075);
xpu=.45;
x=xpu*vp/ifl;
printf("\nThe ohmic reactance of each reactor is: %.2f ohm",x);
```

Scilab code Exa 16.6 to determine short circuit MVA and fault current distribution

```
clear;
clc;

xt=(.15+.1)*(.2)/(.15+.1+.2);
sb=25;
ssc=sb/xt;
printf("\n The short circuit MVA is: %.2f \n",ssc);
if=1/((%i)*xt);
ib=sb/(sqrt(3)*11000);
if=if*ib;
printf("\n The fault current in A is:");
disp(round(if*1000000));
if1=(.15+.1)*if/(.15+.1+.2);
printf("\n The fault current in A supplied by generator 1 is:");
disp(round(if1*1000000));
if2=if-if1;
printf("\n The fault current in A in reactor is:");
disp(round(if2*1000000));
xpu=.45;
vfl=6350;
```

```
ifl=1050;
x=xpu*vfl/ifl;
printf("\n The reactance of each reactor is: %.2f
   ohm",x);
```

Chapter 17

Symmetrical components

Scilab code Exa 17.1 To calculate the positive negative zero sequence component of

```
clear;
//clc();

function [r,i]=d(mag,theta)
    r=mag*cosd(theta);
    i=mag*sind(theta);

endfunction

previousprot = funcprot(0)
funcprot(0)

mag=100;
theta=30;

[r,i]=d(mag,theta);
ia=complex(r,i);
mag=50;
theta=300;

[r,i]=d(mag,theta);
```

```
ib=complex(r,i);

mag=30;
theta=180;
[r,i]=d(mag,theta);
ic=complex(r,i);

ia0=(1/3)*(ia+ib+ic);
printf("The zero sequence component of current in amperes is:");
disp(ia0);

mag=1;
theta=120;
[r,i]=d(mag,theta);
alpha=complex(r,i);

ia1=(1/3)*(ia+alpha*ib+alpha^2*ic);
printf("The positive sequence component of current in amperes is:");
disp(ia1);

ia2=(1/3)*(ia+alpha^2*ib+alpha*ic);
printf("The negative sequence component of current in amperes is:");
disp(ia2);

in=ia+ib+ic;
printf("The return current to the neutral conductor in amperes is:");
disp(in);
```

Scilab code Exa 17.2 To detremine the symmetrical components of the 3 phase system

```
clear;
```

```
//clc();

function [r,i]=d(mag,theta)
    r=mag*cosd(theta);
    i=mag*sind(theta);

endfunction

previousprot = funcprot(0)
funcprot(0)

ia=complex(12,6);
ib=complex(12,-12);
ic=complex(-15,10);
mag=1;
theta=120;
[r,i]=d(mag,theta);
alpha=complex(r,i);

ia0=(1/3)*(ia+ib+ic);
printf("The zero sequence component of current in
   amperes is:");
disp(ia0);
ia1=(1/3)*(ia+alpha*ib+alpha^2*ic);
printf("The positive sequence component of current
   in amperes ia1=");
disp(ia1)

ib1=alpha^2*ia1;
printf("The positive sequence component of current
   in amperes ib1=");
disp(ib1);

ic1=alpha*ia1;
printf("The positive sequence component of current
   in amperes ic1=");
disp(ic1);

```

```
ia2=(1/3)*(ia+alpha^2*ib+alpha*ic);
printf("The negative sequence component of current
   in amperes ia2=");
ib2=alpha*ia2;
printf("The negative sequence component of current
   in amperes ib2=");
disp(ib2);
ic2=alpha^2*ia2;
printf("The negative sequence component of current
   in amperes ic2=");
disp(ic2);
```

Scilab code Exa 17.3 To find the symmetrical components of the line current

```
clear;
//clc();

function [r,i]=d(mag,theta)
    r=mag*cosd(theta);
    i=mag*sind(theta);

endfunction

previousprot = funcprot(0)
funcprot(0)

mag=150;
theta=0;
[r,i]=d(mag,theta);
ia=complex(r,i);
mag=150;
theta=-120;
[r,i]=d(mag,theta);
ib=complex(r,i);
mag=150;
```

```
theta=120;
[r,i]=d(mag,theta);
ic=complex(r,i);
mag=1;
theta=120;
[r,i]=d(mag,theta);
alpha=complex(r,i);

ia0=(ia+ib+ic)/3;
printf("The zero sequence component of current in amperes is:");
disp(round(ia0));
ia1=(ia+alpha*ib+ (alpha^2)*ic)/3;
printf("The positive sequence component of current in amperes ia1=");
disp(round(ia1));
ia2=(ia+alpha^2*ib+alpha*ic)/3;
printf("The negative sequence component of current in amperes ia2=");
disp(round(ia2));
//b).after fuse removal between two lines

i_b=0;
i_c=0;
i_a0=(ia+i_b+i_c)/3;
printf("The zero sequence component of current in amperes is:");
disp(i_a0);

ia1=(ia+alpha*i_b+alpha^2*i_c)/3;
printf("The positive sequence component of current in amperes ia1=");
disp(round(ia1));

ia2=(ia+alpha^2*i_b+alpha*i_c)/3;
printf("The negative sequence component of current in amperes ia2=");
disp(ia2);
```

Scilab code Exa 17.7 To determine the complex power represented by three phase vol

```
clear;
//clc();

function [r,i]=d(mag,theta)
    r=mag*cosd(theta);
    i=mag*sind(theta);

endfunction

previousprot = funcprot(0)
funcprot(0)

mag=10;
theta=190;

[r,i]=d(mag,theta);
ia0=complex(r,i);

mag=6;
theta=20;

[r,i]=d(mag,theta);
ia1=complex(r,i);

mag=5;
theta=50;

[r,i]=d(mag,theta);
ia2=complex(r,i);

ia=(ia0+ia1+ia2);

```

```
mag=1;
theta=120;

[r,i]=d(mag,theta);

alpha=complex(r,i);

ib=(ia0 +alpha^2*ia1 +alpha*ia2);

ic=ia0+alpha*ia1+alpha^2*ia2;

mag=30;
theta=-30;

[r,i]=d(mag,theta);
va0=complex(r,i);

mag=450;
theta=0;

[r,i]=d(mag,theta);
va1=complex(r,i);

mag=225;
theta=40;

[r,i]=d(mag,theta);
va2=complex(r,i);

va=(va0+va1+va2);

mag=1;
theta=120;

[r,i]=d(mag,theta);

alpha=complex(r,i);

```

```
vb=(va0 +alpha^2*va1 +alpha*va2);

vc=va0+alpha*va1+alpha^2*va2;

s=3*va0*conj(ia0)+ 3*va1*conj(ia1)+ 3*va2*conj(ia2);
printf("The complex power with symmetrical
   components in VA is:");
disp(s);

s1=va*conj(ia)+vb*conj(ib)+ vc*conj(ic);
printf("The complex power unbalanced three phase
   components in VA is:");
disp(s1);
```

Chapter 18

Unsymmetrical faults

Scilab code Exa 18.1 To determine the fault current and the line to line voltages

```
clear;
//clc();

function [r,i]=d(mag,theta)
    r=mag*cosd(theta);
    i=mag*sind(theta);

endfunction



previousprot = funcprot(0)
funcprot(0)

ea=1;
za0=(%i)*0.08;
za1=(%i)*0.12;
za2=(%i)*0.12;
sb=25*10^6;
vb=11000;

```

```
zf=(%i)*0.03;
ia1=ea/(za0+za1+za2+3*zf);

ia0=ia1;
ia2=ia1;
mag=1;
theta=120;

[r,i]=d(mag,theta);

alpha=complex(r,i);

ia=ia0+ia1+ia2;

ibas=sb/(sqrt(3)*vb);

ia=ia1*ibas;
ib=0;
ic=0;

va1=ea-za1*ia1;

va2=-za1*ia1;

va0=-(za0+3*zf)*ia1;


va=va0+va1+va2;

vb=va0+alpha^2*va1+alpha*va2;

v_c=va0+alpha*va1+alpha^2*va2;

vab1=va-vb;

vbc1=vb-v_c;

vca1=v_c-va;
```

```
vbas=11/sqrt(3);

vab=vab1*vbas;
printf("The voltage of line ab in kV is vab=");
disp(vab);

vbc=vbc1*vbas;
printf("The voltage of line bc in kV is vbc=");
disp(round(vbc));

vca=vca1*vbas;
printf("The voltage of line ca in kV is vca=");
disp(vca);
```

Scilab code Exa 18.2 to find the sequence LLG and LL

```
clear;
//clc();


function [r,i]=d(mag,theta)
    r=mag*cosd(theta);
    i=mag*sind(theta);

endfunction



previousprot = funcprot(0)
funcprot(0)


x0g1=0.05;
x1g1=0.3;
```

```
x2g1=0.2;
x0g2=0.03;
x1g2=0.25;
x2g2=0.15;
x0l1=0.70;
x1l1=0.3;
x2l1=0.3;
x0l2=0.7;
x1l2=0.3;
x2l2=0.3;
x0t1=0.12;
x1t1=0.12;
x2t1=0.12;
x0t2=0.1;
x1t2=0.1;
x2t2=0.1;
vf=1;
ia01=0;

z1=(%i)*((x1g1+x1t1)*(x1g2 +x1t1 + x1l1*0.5))/(x1g1
    +x1t1 + x1g2+ x1t1+x1l1*0.5);
z2=(%i)*((x2g1+x2t1)*(x2g2+ x2t2+ x2l2*0.5))/(x2g1 +
    x2t1 +x2g2 +x2t2 +x2l2*0.5);
z0=(%i)*(x0g1+x0t1);
//a).LLG fault
za0=z0;
za2=z2;
ia1=vf/(z1+(z0*z2/(za0+za2)));
printf("The positive sequence current in amperes is:
    ");
disp(ia1);

ia0=-(3.1729*z2/(z0+z2));
printf("The zero sequence current in amperes is:");
disp(ia0);

ia2=-ia1*z0/(z0+z2);
printf("The negative sequence current in amperes is:
```

```
")
disp(ia2);

//b).LL fault

ia11=vf/(z1+z2);
printf("The positive sequence current in amperes is:
    ");
disp(ia11);

ia21=-ia11;
printf("The negative sequence current in amperes is:
    ");
disp(ia21);

ia=ia01+ia11+ia21;
printf("phase a fault current  is:")
disp(ia);

mag=1;
theta=120;
[r,i]=d(mag,theta);
alpha=complex(r,i);

ib=ia01+alpha^2*ia11+alpha*ia21;
printf("phase b fault current in pu is:")
disp(ib);

ic=ia01+alpha*ia11+alpha^2*ia21;
printf("phase c fault current in pu is:")
disp(ic);
```

Scilab code Exa 18.3 To calculate voltage to neutral of the faulty phase

```
clear;
```

```
//clc();
xz1g=(%i)*1.2;
xz1f=(%i)*1.0;

xz2g=(%i)*0.9;
xz2f=(%i)*1.0;

xz0g=(%i)*0.4;
xz0f=(%i)*3.0;

vf=11000/sqrt(3);
za0=xz0g +xz0f;
za1=xz1g +xz1f;
za2=xz2g +xz2f;

ia0=vf/(za0+za1+za2);
ia1=ia0;
ia2=ia0;

ifault=3*ia0;
va=vf-ia0*(xz0g+xz1g+xz2g);
printf("The voltage to neutral of the faulty phase
    is:%.2f V",va);
```

Chapter 19

Power system stability

Scilab code Exa 19.1 To find the steady state stability limit

```
clear;
//clc();
vs=132;
vr=vs;
A=0.97;
alpha=0.6;
B=60;
z=B;
bet=70;
C=0.001;
del=91;

prm=vs*vr/B -A*vr^2*cosd(bet-alpha)/B;
printf("the steady state stability limit is:%.2f MW\
   n",prm);

//shunt admittance neglected

y=0;
A1=1 +z*y/2;
alpha1=0;
```

```
D=A;
C=y*(1 +z*y/2);

prm=vs*vr/B -A1*vr^2*cosd(bet-alpha1)/B;
printf("the steady state stability limit is:%.2f MW\n",prm);


//both shunt admittance and series resistance is neglected

B1=B*sind(bet);
bet1=90;

prm=vs*vr/B1 -A1*vr^2*cosd(bet1-alpha1)/B1;
printf("the steady state stability limit is:%.2f MW\n",prm);
```

Scilab code Exa 19.2 To determine teh steady state stability limit

```
clear;
//clc();


function [mag,theta]=c(r,i)
    mag=sqrt(r*r + i*i)
    theta=atand(i/r)
endfunction


previousprot = funcprot(0)
funcprot(0)


//a).inductor switch open
```

```
xdg=0.8;
xt=0.1;
xl=0.6;
xr=0.6;

e=1.2;
v=1;

x=xdg+xt+.5*xl+0.5*xl;
pm=e*v/x;
printf("the steady state stability limit with
   inductor switch S open is:%.2f pu\n",pm);


//b).inductor switch closed

z1=(xdg + xt +0.5*xl);
z2=0.5*xl;
z3=xr;
b=(%i)*z1 + (%i)*z2 +(((%i)*z1*(%i)*z2)/((%i)*z3));

B=imag(b);
x1=B;
pm=e*v/x1;
printf("the steady state stability limit with
   inductor switch S closed is:%.2f pu\n",pm);

//c).with the inductor replaced with a shunt
   capaicitor of same per unit

z31=-0.6;
b1=(%i)*z1 + (%i)*z2 +(((%i)*z1*(%i)*z2)/((%i)*z31))
   ;
B1=imag(b1);
x2=B1;
pm2=e*v/x2;
printf("the steady state stability limit with the
   inductor replaced with a shunt capaicitor is:%.2f
```

```
    pu\n",pm);

//d).when the shunt capaicitor is replaced with a
    series capacitor

xc=-0.6;
x3=xdg+xt+xl+xc;
pm=e*v/x3;
printf("the steady state stability limit when the
    shunt capaicitor is replaced with a series
    capacitor is:%.2f pu\n",pm);

//e).when the shunt inductor is replaced with a
    resistor

z11=xdg + xt +0.5*xl;
z3=1.5;

r=1;
i=z11/z3;
[mag,theta]=c(r,i);
A1=mag;
alpha1=theta;

b=((%i)*z11 + (%i)*z2 +((%i)*z11*(%i)*z2)/z3);

r=real(b);
i=imag(b);


[mag,theta]=c(r,i);
B1=mag;
bet2=theta;
pm=e*v/B1 - A1*v^2*cosd(bet2-alpha1)/B1;
printf("the steady state stability limit when the
    shunt inductor is replaced with a resistor is:%.2
    f pu\n",pm);
```

Scilab code Exa 19.4 To calculate the kinetic energy

```
clear;
//clc();
f=50;
s=60;
j=9000;
p=2;
bs=50;
n=120*f/p;
ke=0.5*j*(2*(%pi)*n/60)^2;
printf("The kinetic energy at rated speed is:%.0f MJ
   \n",ke/1000000);

//to calculate inertia constants M and H

H=ke/s;
printf("the inertia constants H is:%.3f MJ/MVA\n",H
   /1000000);

M=ke/(180*f*1000000);
printf("the inertia constants M is:%.3f MJs/elec.deg
   \n",M);

//the inertia constants

iner_const=M/bs;
printf("The inertia constant is:%.6f pu",iner_const)
   ;
```

Scilab code Exa 19.5 To calculate equivalent h constant at a base of 100MVA

```
clear;
//clc();
s1=500;
h1=4.6;
s2=1500;
s=100;
h2=3;
ke=s1*h1 + s2*h2;
H=ke/s;
printf("the inertia constants H is:%.0f MJ/MVA\n",H)
    ;
```

Scilab code Exa 19.6 To calculte whether the generator will remain in synchronism

```
clear;
//clc();

function y=z(x)
y=sin(x)
endfunction

exact=-2.5432;

pe=50;
pm=100;
p_inc=30;

del1=asin(pe/pm);

//at point b
pb=80;
del2=asin(pb/pm);

a1=pb*(0.927-0.523)-100*intg(0.523,0.927,z);

```

```
a2=100*intg(0.927,(3.14-0.927),z)-80*(3.14-2*del2);

if a1<a2 then
    disp("the generator will remain stable");
else
    disp("the system is unstable");
end
```

Scilab code Exa 19.7 To estimate the sudden increase in generator output

```
clear;
//clc();

//at point a

pe=30;
pmax=60;
del1=asin(pe/pmax);

//by hit and trial methid

del2=60.4;
p1=pmax*sind(del2);

per_load=p1-pe;
printf("The maximum permissible sudden increase of
    load is:%.2f MW",per_load);
```

Scilab code Exa 19.8 To detremine the stabilty of the system

```
clear;
//clc();

```

```
function y=z(x)
y=sin(x)
endfunction

exact=-2.5432;


pe=80;
pmax=200;
del1=asin(pe/pmax);

//at curve b

pmax2=100;
del2=asin(pe/pmax2);
delm=3.14-del2;

a1=pe*(del2-del1)- pmax2*intg(del1,del2,z);

a2=pmax2*intg(del2,delm,z)-pe*(delm-del2);

if a1<a2 then
    disp("the generator will remain stable");

   else
        disp("the system is unstable");
end
```

Scilab code Exa 19.9 To determine the critical clearing angle

```
clear;
//clc();
ps=0.6;
pmax3=0.8;
pmax2=0.3333;
```

```
pmax1=1;

del0=asin(ps);
del2=asin(ps/pmax3);
delm=3.14-del2;
delc=acosd(((ps*(delm-del0)-pmax2*cos(del0)+ pmax3*
   cos(delm))/(pmax3-pmax2)));
printf("The critical clearing angle is:%.2f deg",
   delc);
```

Scilab code Exa 19.10 To determine the critical clearing angle

```
clear;
//clc();

//pre fault condtion

xg=0.3;
xl=0.5;
x=xg +xl*0.5;..// since the line reactances are in
   parallel
pe1=1;
e=1.4;
vc=1;
ps=1;
pmax1=e*vc/x;

del0=asin(ps/pmax1);

//after star delta conversion

xb=0.075;
xc=0.175;
xe=0.0525;
xa=0.375;
```

```
xac=xa+xc+(xa*xc/xe);

pmax2=e*vc/xac;

x1=xg+xl;
pmax3=e*vc/x1;

pe3=1;
del2=asin(pe3/pmax3);
delm=3.14-del2;

delc=acosd((ps*(delm-del0)-pmax2*cos(del0)+ pmax3*
    cos(delm))/(pmax3-pmax2));
printf("the critical clearing angle is:%.2f deg",
    delc);
```

Scilab code Exa 19.11 To plot the sqing curve

```
clear;
//clc();

H=2.7;
s=1;
fr=50;

del0=23.13;
pmax1=2.545;
pmax2=0.778;
pmax3=1.75;
ps=1;
del_del0=0;

M=H*s/(180*fr);
 del_del=zeros(1,9);
```

```
 pa=zeros(1,9);
 del=zeros(1,9);
//for t=0-
pmax=pmax1;
//for t=0+;
pmax=pmax2;
p=ps-pmax2*sind(del0);
pa_avg=0.5*p;
pa(1)=pa_avg;
t=[.1:.9:9];
del(1)=del0;
del(2)=del_del0+8.33*pa_avg;

del(2)=del0+del(2);
for i=2:1:9
    pa(i)=ps-.778*sind(del(i));
    del_del(i+1)=del_del(i)+8.33*pa(i);
    del(i+1)=del(i)+ del_del(i+1);
end

plot(t,del)
xlabel("time in secs");
ylabel('torque angle in degrees');
```

Chapter 20

Travelling waves

Scilab code Exa 20.1 to calculate the surge impedances and velocities of the line

```
clear;
//clc();
l1=2*10^(-3);
c1=0.01*10^(-6);

z1=sqrt(l1/c1);
printf("The surge impedance of the overhead line is:
    %.0f Ohm\n",z1);
v1=1/sqrt(l1*c1);
printf("The velocity of the wave in overhead line is
    :%.2f km/s\n",v1);

l2=0.25*10^(-3);
c2=0.102*10^(-6);

z2=sqrt(l2/c2);
printf("The surge impedance of the cable line is:%.2
    f Ohm\n",z2);
v2=1/sqrt(l2*c2);
printf("The velocity of the wave in cable is:%.2f km
    /s\n",v2);
```

```
ef=100*1000;
inc=ef/z1;

tau1=(2*z2)/(z2+z1);
et=tau1*ef;
printf("The transmitted voltage in the cable is:%.2f
    kV\n",et/1000);

taui1=(2*z1)/(z2+z1);
it=taui1*inc;
printf("The transmitted current in the cable is:%.2f
    A\n",it);

rhov1=(z2-z1)/(z1+z2);
er=rhov1*ef;
printf("The reflected voltage in the cable is:%.2f
   kV\n",er/1000);


rhoi1=(z1-z2)/(z1+z2);
ir=rhoi1*inc;
printf("The reflected current in the cable is:%.2f A
   \n",ir);

disp('the surge voltage has been changed to 100kV');


ef=100;
inc=ef*1000/z2;

tau2=(2*z1)/(z2+z1);
et=tau2*ef;
printf("The transmitted voltage in the cable is:%.2f
    kV\n",et);

taui2=(2*z2)/(z2+z1);
```

```
it=taui2*inc;
printf("The transmitted current in the cable is:%.2f
    A\n",it);

rhov2=(z1-z2)/(z1+z2);
er=rhov2*ef;
printf("The reflected voltage in the cable is:%.2f
   kV\n",er);

rhoi2=(z2-z1)/(z1+z2);
ir=rhoi2*inc;
printf("The reflected current in the cable is:%.2f A
   ",ir);
```

Scilab code Exa 20.2 To find the voltage distribution

```
clear;
//clc();
z1=400;
z2=40;
ef=100;
voh=300000;..//velocity of overhead line
vc=140000;..//velocity of cable
rho1=(z2-z1)/(z2 + z1);
er=rho1*ef;
printf("The voltage transmitted into the cable is:%
   .2f kV\n",er);
tauv=(2*z2)/(z2+z1);
et=tauv*ef;
printf("The voltage transmitted into the cable is:%
   .2f kV",et);
```

Scilab code Exa 20.3 to find the reflected voltage and current in the cable

```
clear;
//clc();
z1=100;
z2=600;
z3=1000;
ef=1000;
et=2*ef/((1/z2 + 1/z3)*100 +1);
er=et -ef;
printf("The reflected voltage in the line is:%.0f V\
   n",er);

i1=et/z2;
i2=et/z3;
i=i1 + i2;
printf("The current in the cable is:%.2f A\n",i);
```

Scilab code Exa 20.4 To determine from first principles the magnitudes of first an

```
clear;
//clc();

z2=50;
z1=400;
z3=400;
ef=10;

tau1=(2*z2)/(z1 + z2);
rho1=(z1-z2)/(z2+z1);
tau2=(2*z3)/(z3 + z2);
rho2=(z3-z2)/(z3+z2);
etb=tau1*tau2*ef;
printf("The transmitted part entering the line BC is
   :%.2f kV\n",etb);

eta=rho1*rho2*tau1*ef;
```

```
etb1=rho1*rho2*tau1*tau2*ef;
printf("The second pulse entering the line BC is:%.2
    f kV",etb1);
```

Scilab code Exa 20.5 To find the surge voltage distribution

```
clear;
//clc();

l=265*10^(-6);
c=0.165*10^(-6);

z2=sqrt(l/c);
z1=400;
ef=500;
z3=1000;
et1= (2*z2)*ef/(z1 + z2);
printf("The incident voltage in the cable is:%.0f kV
    \n",et1);

erb=(z3 - z2)*et1/(z3 + z2);
printf("The reflected voltage at the transformer end
     is:%.0f kV\n",erb);

vcd=et1+erb;
printf("The voltage at the junction is:%.0f kV",vcd)
    ;
```

Scilab code Exa 20.6 To calculate the current voltage waves and the resistance ref

```
clear;
//clc();
```

```
z1=400;
z2=600;
zp=2000;
ef1=100;
ef2=(2/z1)*ef1/(1/z1 + 1/z2 + 1/zp);
er1=ef2-ef1;
printf("The voltage reflected in line1 is:%d kV\n",er1);

ir1=round(er1)/z1;
printf("The current reflected in line1 is:%.2f kV\n",ir1*1000);

zp=1/(-(1/z1)-(1/z2)+(2/z1));
printf("The resistance at the junction is:%.0f Ohm",zp);
```

Scilab code Exa 20.7 To calculate the voltage across load

```
clear;
//clc();

z=500;
r=10*10^(3);
c=0.005*10^(-6);
tau=c*r*z/(z + r);

ef=10;
t=5*10^(-6);
e=(2*r*ef/(z + r))*(1- exp(-t/tau));
printf("The voltage across load after 5us is:%.2f kV",e);
```

Scilab code Exa 20.8 To find the maximum voltage

```
clear;
//clc();

z1=400;
z2=300;
z3=300;
z2=300;
zd=50;

et=2*(1/z1)/(1/z1  + 1/z2 + 1/z3);
et1=et*(2*zd/(z2 + zd));
t=3.1534;
v=250*(exp(-.05*t) - exp(-t));
e=et1*v;
printf("The maximum voltage is:%.2f kV",e);
```

Chapter 22

Corona

Scilab code Exa 22.1 to determine the disruptive critical voltage visual critical

```
clear;
//clc();
v=132;
f=50;
l=150;
d=19.53*10^(-3);
t=30;
p=750;
v_grad=21.1;
sur_fact=0.85;
l_sur_fact=0.72;
g_sur_fact=0.82;
D=3.8;

r=d/2;
del=(0.392*p)/(273 + t);
e=v_grad*100*sur_fact*r*del*log([D/r]);
printf("The disruptive critical voltage is:%.2f kV\n
    ",e);

ev=v_grad*100*l_sur_fact*r*del*(1 +(0.0301/sqrt(r*
```

```
   del)))*log([D/r]);
printf("The visual critical voltage for local corona
    is:%.2f kV\n",ev);

evg=v_grad*100*g_sur_fact*r*del*(1 +(0.0301/sqrt(r*
   del)))*log([D/r]);
printf("The visual critical voltage for general
   corona is:%.2f kV",evg);
```

Scilab code Exa 22.2 to estimate the corona loss

```
clear;
//clc();
v=110;
f=50;
l=150;
d=10*10^(-3);
irr_fac=0.85;
v_grad=30;
p=750;
t=30;
D=2.5;

r=d/2;
del=(0.392*p)/(273 + t);
e=v_grad*100*irr_fac*r*del*log([D/r])/sqrt(2);

en=v/sqrt(3);
pc=(244/del)*(f+25)*(en-e)^(2)*sqrt(r/D)*l/100000;

tot_loss=3*pc;
printf("The total corona loss is: %.2f kW",tot_loss)
   ;
//difference in answer is due to rounding off of pc
```

Chapter 24

System neutral grounding

Scilab code Exa 24.1 to calculate reactance to neutralize the capacitance to earth

```
clear;
//clc();
f=50;
cap=1.2*10^(-6);
xl=1/(3*2*(%pi)*cap*f);
printf("The inductive reactance to neutralize 100
   percent of the capacitance is:%.2f Ohm\n",xl);

xl1=xl/0.9;
printf("The inductive reactance to neutralize 90
   percent of the capacitance is:%.2f Ohm\n",xl1);

xl2=xl/0.8;
printf("The inductive reactance to neutralize 80
   percent of the capacitance is:%.2f Ohm",xl2)
```

Scilab code Exa 24.2 To determine the inductance and kva rating

```
clear;
//clc();

l=80
cap=0.00914*10^(-6)*l;
 vl=132*1000;
 vph=vl/sqrt(3);
 f=50;
 l=1/(3*(2*(%pi)*f)^(2)*cap);
printf("The inductance is:%.2f H\n",l)

il=vph/((2*(%pi)*f)*l);
kvar=vph*il/1000;
printf("The rating of the arc suppression coil is:%
   .2f kVA",kvar)
```

Chapter 25

Tarrifs

Scilab code Exa 25.1 To determine the overall cost per kwh

```
clear;
//clc();
max_dem=80;
pf=0.45;
tar_md=750;
tar_en=1.1;
ann_ener_cons=max_dem*pf*8760;
ce=1.1*ann_ener_cons;

cf=tar_md*max_dem;
tot=ce+cf;
cost_per_kwh=tot/ann_ener_cons;
printf("the overall cost is:%.2f Rs",cost_per_kwh);
```

Scilab code Exa 25.2 To calculate the annual bill

```
clear;
//clc();
```

```

con_load=50;
max_dem=40;
t=8;
days=300;
tar_f=5000;
tar_md=800;
tar_kwh=1.15;

ene_con=con_load*t*days;
ann_bill=tar_f + tar_md*max_dem + tar_kwh*ene_con;
printf("the annual bill is:%.0f Rs",ann_bill);
```

Scilab code Exa 25.3 To determine the cost of energy per kwh at the busbars

```
clear;
//clc();
op=960*10^(6);..//kWh generated per year
lf=0.55;
tar_fix=1000;
tar_kwh=0.4;

max_dem=op/(lf*8760);
ann_dem_cost=tar_fix*max_dem;
ann_ene_cost=op*0.4;
tot= ann_dem_cost+ ann_ene_cost;
cost=tot/op;
printf("the cost of energy per kwh at the busbars is
    :%.3f Rs",cost);
//difference in answer is due to the misprint in the
    calculation of total annual charges
```

Scilab code Exa 25.4 to determine the energy consumption per year and the yearly b

```
clear;
//clc();
l1=100;
t1=9;
l2=125;
t2=6;
l3=50;
t3=7;
l4=5;
t4=2;
tar_md=800;
tar_kwh=1.3;

ene_per_day=l1*t1 + l2*t2+ l3*t3+ l4*t4;
ann_ene=ene_per_day*365;
printf("The annual energy consumption is:%.0f kWh\n"
    ,ann_ene);
ann_cost=ann_ene*tar_kwh;
max_dem=l2;
ann_md=max_dem*tar_md;
tot=ann_cost + ann_md;

printf("the energy consumption per year and the
    yearly bill is:%.0f Rs",tot);
```

Scilab code Exa 25.5 To detremine the saving in energy cost

```
clear;
//clc();
ann_con=200000;
lf=0.4;
tar_md=750;
tar_kwh=1.15;
lf1=0.6;

```

```
avg_l=ann_con/8760;
max_dem=avg_l/lf;

max_dem1=avg_l/lf1;
red=max_dem - max_dem1;
sav=tar_md*red;
printf("The saving in energy cost is:%.2f Rs",sav);
```

Scilab code Exa 25.6 To determine the total energy consummed and the total bill

```
clear;
//clc();
max_dem=150;
lf=0.65;
tar_md=900;
tar_kwh=1.3;
pf=0.82;

max_kva_dem=max_dem/pf;
ann_con=(max_dem)*lf*8760;
printf("the total energy consumed is:%.0f kWh\n",
   ann_con);
ann_cost=ann_con*tar_kwh;
ann_dem_cost=tar_md*max_kva_dem;
ann_bill=ann_cost + ann_dem_cost;
printf("The annual electricity bill is:%.0f Rs",
   ann_bill);
```

Scilab code Exa 25.7 To determine which tariff is economical

```
clear;
//clc();
max_dem=100;
```

```
tar_md=800;
tar_kwh=1.3;
tar_kwh1=1.83;
t=3600;
lf=0.8;
avg_dem=max_dem*lf;
ann_ene=avg_dem*t;
ann_bill=(tar_md*max_dem + tar_kwh*ann_ene);

ann_bill1=tar_kwh1*ann_ene;

if (ann_bill>ann_bill1) then
    disp("flat rate tarrif is better");
else
    disp("two part tariff is better");

end
```

Scilab code Exa 25.8 to calculate the annual bill

```
clear;
//clc();
w_days=300;
t=8;
tar_mdh=75;
tar_kwh=1.15;

tar_mdl=80;
tar_kwhl=1.43;

avg_load=400;
pf=0.8;
max_dem=500;
loss=0.04;
dep=0.15;
```

```
cost=900;

max_kva=max_dem/pf;
cap=max_kva/(1-loss);
cost_hv=cost*cap;
ann_int=dep*cost;
ann_max_dem=tar_mdh*12*max_kva;

ene_con=avg_load*w_days*t/(1-loss);

cost_ene_con=ene_con*tar_kwh;
tot_charges=ann_max_dem + cost_ene_con;
printf("The annual bill is:%.0f Rs\n",tot_charges);

//low voltage supply

ann_max_deml=tar_mdl*12*max_kva;
ann_ene_conl=avg_load*w_days*t;
cost_ene_con=ann_ene_conl*tar_kwhl;
tot_costl=ann_max_deml + cost_ene_con;
printf("The annual bill is:%.0f Rs",tot_costl);
```

Scilab code Exa 25.9 To find the number of units for which the tariff is econimica

```
clear;
//clc();

tar_kwh=1.43;
flat_tar=1.83;
fixed=400;
units=fixed/(flat_tar-tar_kwh);
printf("the number of units for which the tariff is
   econimical is:%.0f",units);
```

Scilab code Exa 25.10 to find the energy consumption per year

```
clear;
clc;

//c1=500+.9x;
//c2=1.83x-60;
//c1=c2;
x=(500+60)/(1.83-.90);
printf("The energy consumed per year is: %.2f kWh",
   round(x));
```

Scilab code Exa 25.11 to find the generation cost per kwh

```
clear;
//clc();
ins_cap=500*10^(6);
cap_cost=35000*10^(3);
dep=0.12;
fuel_con=0.85;
fuel_cost=0.8;
oper_cost=0.25;
p_load=475*10^(6);
lf=0.82;
avg_load=p_load*lf;
ene_gen=avg_load*8760/1000;
tot_inv=ins_cap*cap_cost/1000000;

ann_dep=dep*tot_inv;

fuel_con_yr=fuel_con*avg_load*8760/1000;

```

```
ann_fuel_cost=fuel_cost*fuel_con_yr;
other_cost=oper_cost*ann_fuel_cost;
gen_cost=(ann_dep + ann_fuel_cost + other_cost)/
   ene_gen;
printf("the generation cost per kwh is:%.2f Rs",
   gen_cost);
```

Chapter 26

Power factor improvement

Scilab code Exa 26.1 To claculate the value of capicitance

```
clear;
//clc();
i1=0.75;
v=240;
p=80;
q1=acosd(1/(3*i1));
q2=0;
ic=i1*cosd(q1)*(tand(q1)-tand(q2));
cap=ic/(v*2*(%pi)*50);
printf("The capaciatnce is:%.2f uF\n",cap*1000000);

t1=acosd(0.95);

ic1=i1*cosd(q1)*(tand(q1)-tand(t1));
cap1=ic1/(v*2*(%pi)*50);
printf("The capaciatnce is:%.2f uF\n",cap1*1000000);
```

Scilab code Exa 26.2 To calculate the capacitiance kVAr and the new supply of curr

```
clear;
//clc();
i1=20;
f=50;
v=230;
q1=acosd(0.75);
q2=acosd(0.9);

ic=i1*cosd(q1)*(tand(q1)-tand(q2));

cap=ic/(v*2*(%pi)*f);
printf("The capaciatnce is:%.2f uF\n",cap*1000000);

qc=v*ic;
printf("The kVAr is:%.3f kVAr\n",qc/1000);
i2=i1*cosd(q1)/cosd(q2);
printf("the new supply current is:%.2f A",i2);
```

Scilab code Exa 26.3 to calculate the kVAr rating and capacitance per phase

```
clear;
//clc();

p=750/3;
v=400;
q1=acosd(0.7);
q2=acosd(0.95);
qcp=p*(tand(q1)-tand(q2));
printf("The kVAr is:%.3f kVAr\n",qcp);
cap=qcp*1000/(v*v*2*(%pi)*50);
printf("The capacitance is:%.0f uF\n",cap*1000000);
```

Scilab code Exa 26.4 To calculate the capacitance per phase

```
clear;
//clc();
op=40;
eff=0.85;
ip=op/eff;
p=ip/3;
v=400;

q1=acosd(0.72);
q2=acosd(0.98);

qcp=p*(tand(q1)-tand(q2));
cap=qcp*1000/(v*v*2*(%pi)*50);// capacitance in
    milli farad
printf("The capacitance is:%.2f uF\n",cap*1000000);
```

Scilab code Exa 26.5 To calculate current to the motor at full load current throug

```
clear;
//clc();

op=7500;
eff=0.87;
ip=op/eff;
v=230;

p=ip;
pf=.75;
q1=acosd(pf);
q2=acosd(1);

i=p/(v*pf);
printf("the current is:%.2f A\n",i);
qcp=p*(tand(q1)-tand(q2));

```

```
ic=qcp/v;
cap=qcp*1000/(v*v*2*(%pi)*50);// capacitance in
    milli farad
printf("The capacitance is:%.2f uF\n",cap*1000);
```

Scilab code Exa 26.6 To calculate the capacitance per phase

```
clear;
//clc();
p=50/3;
q1=acosd(0.72);
q2=acosd(0.9);

qcp=p*(tand(q1)-tand(q2));
v=415;

ic=qcp/v;
cap=ic*1000/(v*2*(%pi)*50);..// capacitance in milli
    farad
printf("The capacitance is:%.2f uF\n",cap*1000000);

//in delta connection

vp=v/sqrt(3);

icd=qcp/vp;
cap=icd*1000/(vp*2*(%pi)*50);// capacitance in milli
    farad
printf("The capacitance is:%.2f uF\n",cap*1000000);
```

Scilab code Exa 26.7 To calculate the leading kvar kva rating and power factor

```
clear;
```

```
//clc();
p1=500;
p2=100;
q1=acosd(.707);
q2=acosd(0.95);
lkvar=(p1*tand(q1) - (p1+p2)*tand(q2));
printf("The kVAr is:%.2f kVAr\n",lkvar);

kva=sqrt(p2*p2 + lkvar*lkvar);
printf("The kVA is:%.2f kVA\n",kva);

pf=(p2/kva);
printf("The power factor is:%.3f leading\n",pf);
```

Scilab code Exa 26.8 To calculate power factor leading kvar and kva

```
clear;
//clc();
vl=400;
il=36;
pf=0.8;

p1=sqrt(3)*vl*il*pf;
op=15*746;
eff=0.85;
p2=op/eff;

//as per ques 26.7
p1=19.953;
p2=12.979;
q1=acosd(0.8);
q2=acosd(0.92);
qm=((p1*tand(q1)) - (p1+p2)*tand(q2))/p2;
pf=cosd(atand(qm));
printf("The power factor is:%.4f leading\n",pf);
```

```
lkvar=p2*(qm);
printf("The kVAr is:%.3f kVAr\n",lkvar);


kva=sqrt(p2*p2 + lkvar*lkvar);
printf("The kVA is:%.0f kVA\n",kva);
```

Scilab code Exa 26.9 To calculate the total load kVA and power factor

```
clear;
//clc();
s=15;
sa=complex(s,0);

s1=40;
sb1=complex(s1*0.6,s1*0.8);
sb=conj(sb1);

s2=25;
sc1=complex(s2*0.8,s2*0.6);
sc=conj(sc1);

st=sa + sb + sc;

t_kvar=sqrt((real(st)^2) + (imag(st)^2));
printf("The kVA is:%.2f kVA\n",t_kvar);

pf=real(st)/t_kvar;
printf("The power factor is:%.4f lagging\n",pf);

cap=-imag(st);
printf("The capacitance is:%.0f kVAr leading\n",cap)
    ;
```

Scilab code Exa 26.10 To calculate the annual saving

```
clear;
//clc();

x=900;
p=2000;
q1=acosd(0.8);
q2=acosd(0.96);
smd=x*p*((1/cosd(q1))- (1/cosd(q2)));

y=0.15*1950;
ann_cost=y*p*(tand(q1)-tand(q2));
tot_ann=smd-ann_cost;
printf("The annual saving is:%.0f Rs",tot_ann);
```

Scilab code Exa 26.11 To detremine the most economical pf and the kva rating

```
clear;
//clc();
p=900;
q1=acosd(0.65);
x=1000;
y=0.15*2000;
pf=sqrt(1- (y/x)^2);
printf("The power factor is:%.4f lagging\n",pf);

q2=acosd(pf);
qc=p*(tand(q1)-tand(q2));
printf("The capacitance is:%.0f uF\n",qc);
```

Scilab code Exa 26.12 To detremine the annual savings

```
clear;
//clc();
p=800;
pf1=.72;
q1=acosd(pf1);
x=500;
y=160;
pf=sqrt(1- (y/x)^2);

q2=acosd(pf);
qc=p*(tand(q1)-tand(q2));

smd=x*p*((1/pf1)-(1/pf));

ann_cost=y*qc;
sav=smd-ann_cost;
printf("The annual saving is:%.0f Rs",sav);
```

Scilab code Exa 26.13 To estimate the limiting cost per kvar

```
clear;
//clc();
q1=round(acosd(0.707));

q2=round(acosd(0.866));

x=8000;

y=x*((cosd(q2)- cosd(q1))/sind(q1-q2));
printf("the limiting cost per kvar is:%.2f Rs",y);
```

```
//difference in answer is due to the approximation
   of decimals
```