Scilab Textbook Companion for Electric Power Generation, Transmission And Distribution

by S. N. Singh[1]

Created by
Ajith Kumar R
B.E.
Electrical Engineering
St.Xavier's Catholic College of Engineering
College Teacher
None
Cross-Checked by
None

August 23, 2018

---

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

Book Description

Title: Electric Power Generation, Transmission And Distribution

Author: S. N. Singh

Publisher: Phi Learning Private Limited

Edition: 2

Year: 2012

ISBN: 978-81-203-3560-8

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

Exa Example (Solved example)

Eqn Equation (Particular equation of the above book)

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

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

Contents

List of Scilab Codes

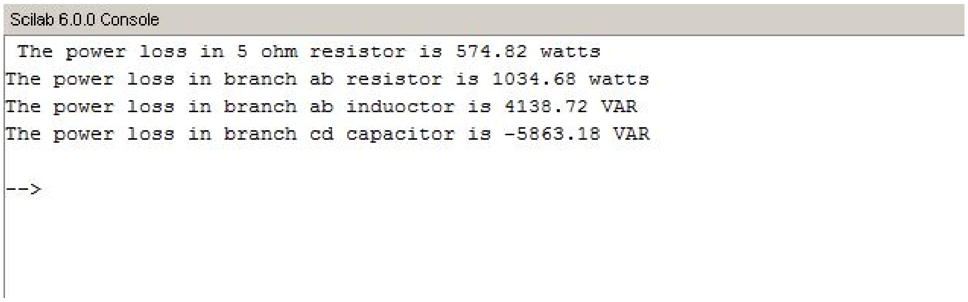
List of Figures

Chapter 3

Basic principles

Scilab code Exa 3.2 `Computation of Power`

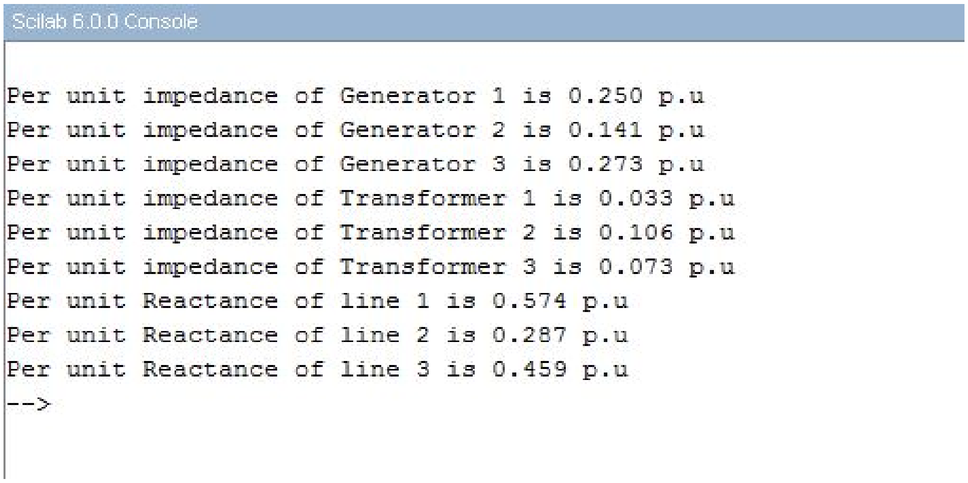
```
//Electric Power Generation, Transmission and
   Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 3.2
//Scilab Version : 6.0.0      ; OS : Windows

clc;
```



```
 The power loss in 5 ohm resistor is 574.82 watts
The power loss in branch ab resistor is 1034.68 watts
The power loss in branch ab induoctor is 4138.72 VAR
The power loss in branch cd capacitor is -5863.18 VAR

-->
```

Figure 3.1: Computation of Power

```
clear;

vs=220;                              // Supply voltage in Volts
rs=5;                                // Series resistance in Ohms
rp=2;                                // Parallel resistance in Ohms
xlp=8*%i;                            // Parallel inductive reactance in Ohms
xcp=-6*%i;                           // Parallel capacitive reactance in Ohms
zeq=((rp+xlp)*xcp)/(rp+xlp+xcp); // Equivalent impedance of parallel branch in Ohms
I=vs/(rs+zeq);                       // Current in the series branch in Ampere
Ps=((I)^2)*rs;                       // Power in 5 ohm resistor Watts
I1=I*xcp/(rp+xlp+xcp);               // Current in branch ab in Ampere
I2=I*(rp+xlp)/(rp+xlp+xcp);          // Current in branch cd in Ampere
Pab=(I1^2)*rp;                       // Power loss in branch ab resistor in Watts
Qab=(I1^2)*xlp;                      // Power loss in branch ab inductor in VAR
Qcd=(I2^2)*(xcp);                    // Power loss in branch cd capacitor in VAR

printf('The power loss in 5 ohm resistor is %.2f watts \n',abs(Ps))
printf('The power loss in branch ab resistor is %.2f watts \n',abs(Pab))
printf('The power loss in branch ab induoctor is %.2f VAR \n',abs(Qab))
printf('The power loss in branch cd capacitor is %.2f VAR \n',-abs(Qcd))         //Negative sign since capacitor supplies reactive power
```



```
Per unit impedance of Generator 1 is 0.250 p.u
Per unit impedance of Generator 2 is 0.141 p.u
Per unit impedance of Generator 3 is 0.273 p.u
Per unit impedance of Transformer 1 is 0.033 p.u
Per unit impedance of Transformer 2 is 0.106 p.u
Per unit impedance of Transformer 3 is 0.073 p.u
Per unit Reactance of line 1 is 0.574 p.u
Per unit Reactance of line 2 is 0.287 p.u
Per unit Reactance of line 3 is 0.459 p.u
-->
```

Figure 3.2: Find the impedance

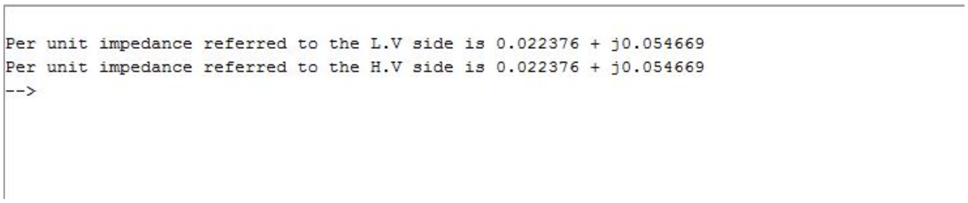
Scilab code Exa 3.4 `Find the impedance`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 3.4
//Scilab Version : 6.0.0    ; OS : Windows

clc;
clear;

MVAnew=100;
    //MVA reference in MVA
KVnew=11;
```

```
    //KV reference before Transformer in kV
KVnew1=132;
    //KV reference after Transformer in kV
MVAg1=100;
    //Apparent power in Generator 1 in MVA
KVg1=11;
    //Voltage at Generator bus 1 in kV
Xg1=0.25;
    //Reactance of Generator 1 at individual p.u. Ohm
MVAg2=150;
    //Apparent power in Generator 2 in MVA
KVg2=16;
    //Voltage at Generator bus 2 in kV
Xg2=0.10;
    //Reactance of Generator 2 at individual p.u Ohm
MVAg3=200;
    //Apparent power in Generator 3 in MVA
KVg3=21;
    //Voltage at Generator bus 3 in kV
Xg3=0.15;
    //Reactance of Generator 3 at individual p.u Ohm
MVAt1=150;
    //Apparent power in Transformer 1 in MVA
t1pry=11;
    //Primary voltage in Transformer 1 in kV
```

```
t1sec=132;
   //Secondary voltage in Transformer 1 in kV
Xt1=0.05;
   //Reactance of Transformer 1 at individual p.u
   Ohm
MVAt2=200;
   //Apparent power in Transformer 2 in MVA
t2pry=16;
   //Primary voltage in Transformer 2 in kV
t2sec=132;
   //Secondary voltage in Transformer 2 in kV
Xt2=0.10;
   //Reactance of Transformer 2 at individual p.u
   Ohm
MVAt3=250;
   //Apparent power in Transformer 3 in MVA
t3pry=21;
   //Primary voltage in Transformer 3 in kV
t3sec=132;
   //Secondary voltage in Transformer 3 in kV
Xt3=0.05;
   //Reactance of Transformer 3 at individual p.u
   Ohm
Xl1=100;
   //Reactance of Transmission line 1 at individual
   p.u Ohm
Xl2=50;
```

```
    //Reactance of Transmission line 2 at individual p.u Ohm
Xl3=80;
    //Reactance of Transmission line 3 at individual p.u Ohm
X1=Xg1*(MVAnew/MVAg1)*(KVg1/KVnew)^2;                //Reactance of Generator 1 at individual p.u Ohm
X2=Xg2*(MVAnew/MVAg2)*(KVg2/KVnew)^2;                //Reactance of Generator 2 at individual p.u Ohm
X3=Xg3*(MVAnew/MVAg3)*(KVg3/KVnew)^2;                //Reactance of Generator 3 at individual p.u Ohm
T1=Xt1*(MVAnew/MVAt1)*(t1pry/KVnew)^2;              //Impedance of Transformer 1 at individual p.u Ohm
T2=Xt2*(MVAnew/MVAt2)*(t2pry/KVnew)^2;              //Impedance of Transformer 2 at individual p.u Ohm
T3=Xt3*(MVAnew/MVAt3)*(t3pry/KVnew)^2;              //Impedance of Transformer 3 at individual p.u Ohm
Zb=((KVnew1)^2)/MVAnew;                                        //Base Reactance of Transmission line at Ohm
Ll=Xl1/Zb;
    //Reactance of Transmission line 1 at individual p.u Ohm
L2=Xl2/Zb;
    //Reactance of Transmission line 2 at individual p.u Ohm
L3=Xl3/Zb;
```



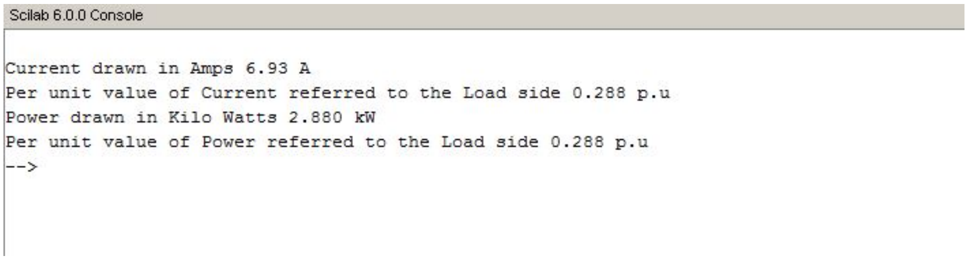
```
Per unit impedance referred to the L.V side is 0.022376 + j0.054669
Per unit impedance referred to the H.V side is 0.022376 + j0.054669
-->
```

Figure 3.3: Per unit calculation

```
    //Reactance of Transmission line 3 at individual p.u Ohm


printf("\nPer unit impedance of Generator 1 is %.3f p.u",X1);
printf("\nPer unit impedance of Generator 2 is %.3f p.u",X2);
printf("\nPer unit impedance of Generator 3 is %.3f p.u",X3);
printf("\nPer unit impedance of Transformer 1 is %.3f p.u",T1);
printf("\nPer unit impedance of Transformer 2 is %.3f p.u",T2);
printf("\nPer unit impedance of Transformer 3 is %.3f p.u",T3);
printf("\nPer unit Reactance of line 1 is %.3f p.u",L1);
printf("\nPer unit Reactance of line 2 is %.3f p.u",L2);
printf("\nPer unit Reactance of line 3 is %.3f p.u",L3);
```

Scilab code Exa 3.5 `Per unit calculation`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 3.5
//Scilab Version : 6.0.0     ; OS : Windows

clc;
clear;

kVpry=220;                                   //Primary voltage of Transformer in kV
kVsec=400;                                   //Secondary voltage of Transformer in kV
MVAb=240;                                    //Apparent Base power in Transformer in MVA
Zpry=3+%i*8;                                 //Primary Impedance of Transformer in Ohm
Zsec=5+%i*10;                                //Secondary Impedance of Transformer in Ohm
Zlv=(Zpry)+(Zsec)*(kVpry/kVsec)^2;           //Impedance referred to LV side in Ohm
Zlvpu=(Zlv)*(MVAb/(kVpry)^2);                //Per unit impedance referred to LV side in p.u. Ohm
Zhv=(Zsec)+(Zpry)*(kVsec/kVpry)^2;           //Impedance referred to HV side in Ohm
Zhvpu=(Zhv)*(MVAb/(kVsec)^2);                //Per unit impedance referred to HV side in p.u. Ohm


printf("\nPer unit impedance referred to the L.V side is %.6f + j%.6f",real (Zlvpu),imag (Zlvpu));
printf("\nPer unit impedance referred to the H.V side is %.6f + j%.6f",real (Zhvpu),imag (Zhvpu));
```



```
Current drawn in Amps 6.93 A
Per unit value of Current referred to the Load side 0.288 p.u
Power drawn in Kilo Watts 2.880 kW
Per unit value of Power referred to the Load side 0.288 p.u
-->
```

Figure 3.4: Per unit calculation

Scilab code Exa 3.6 `Per unit calculation`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 3.6
//Scilab Version : 6.0.0    ; OS : Windows

clc;
clear;

V=240;                                    //Three phase supply voltage in Volts
Rl=20;                                    //Load Resistance in Ohms
Vbase=240;                                //Three phase Base voltage in Volts
VAbase=10*10^3;                           //Base voltage in kVA
Vpu=V/Vbase;                              //Voltage in p.u.
Zbase=(Vbase^2/VAbase);                   //Base Impedance in Ohms
Zpu=Rl/Zbase;                             //
```

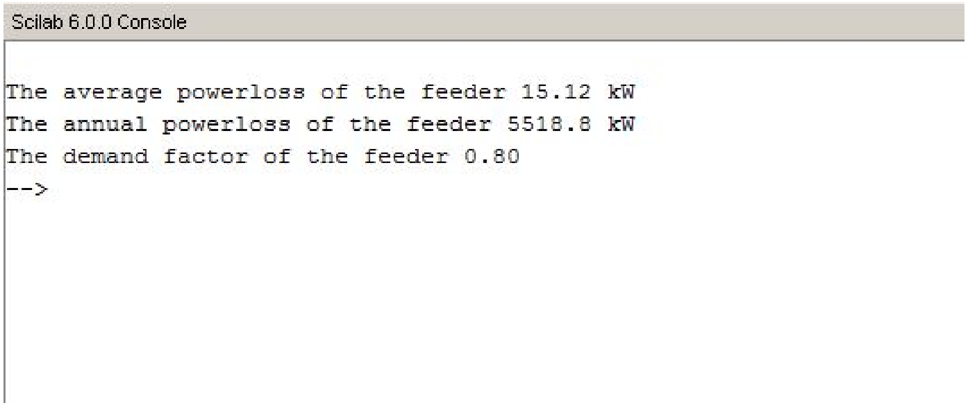
```
   Load Impedance in p.u.
Ibase=VAbase/((nthroot(3,2))*Vbase);                    //
   Base Current in Amps
Ipu=Vpu/Zpu;                                            //
   Current drawn in p.u.
Ia=Ipu*Ibase;                                           //
   Current drawn in Amps
P=Vpu*Ipu;                                              //
   Power drawn in p.u.
Pt=(Ipu*VAbase)/1000;                                   //
   Power drawn in kW


printf("\nCurrent drawn in amps %.2f A",Ia);
printf("\nPer unit value of current referred to the
   load side %.3f p.u",Ipu);
printf("\nPower drawn in kilo watts %.3f kW",Pt);
printf("\nPer unit value of Power referred to the
   load side %.3f p.u",P);
```

Chapter 4

Load Characteristics and Economic Aspects

Scilab code Exa 4.1 `Load characteristics calculation`

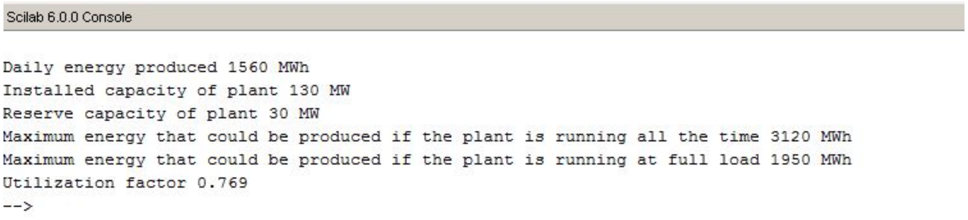
```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 4.1
//Scilab Version : 6.0.0     ; OS : Windows

clc;
clear;


Z =[400,380,350,300,350,500,700,750,900,1200,1350,1200,1000,950,125
Totalenergy = 0;    //Initial Total energy
```



```
The average powerloss of the feeder 15.12 kW
The annual powerloss of the feeder 5518.8 kW
The demand factor of the feeder 0.80
-->
```

Figure 4.1: Load characteristics calculation

```
for i=1:length(Z)
Totalenergy=Z(i)+Totalenergy;
end
Averagedemand=Totalenergy/24;                    //Average demand of the feeder in kW
Maximumdemand=2000;                    //Maximum demand of the feeder in kW
Loadfactor=Averagedemand/Maximumdemand;                    //Load factor of the feeder
Lossfactor=0.14;                    //Loss factor of the feeder
Peakloadpowerloss=108;                    //Peakload power loss of the feeder in kW
Averagepowerloss=Lossfactor*Peakloadpowerloss;                    //Average power loss of the feeder in kW
Annualpowerloss=Averagepowerloss*365;                    //Annual power loss of the feeder in kW
```



```
Daily energy produced 1560 MWh
Installed capacity of plant 130 MW
Reserve capacity of plant 30 MW
Maximum energy that could be produced if the plant is running all the time 3120 MWh
Maximum energy that could be produced if the plant is running at full load 1950 MWh
Utilization factor 0.769
-->
```

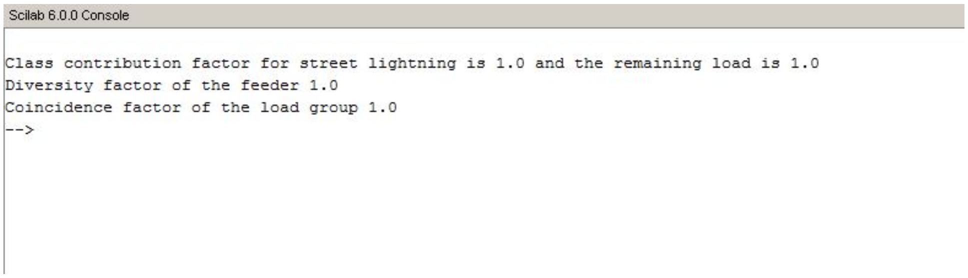
Figure 4.2: Load characteristics calculation

```
Connecteddemand=2500;                    //Connected demand of the feeder in kW
Demandfactor=Maximumdemand/Connecteddemand;                    //Demand factor of the feeder



printf("\nThe average powerloss of the feeder %.2f kW",Averagepowerloss);
printf("\nThe annual powerloss of the feeder %.1f kW",Annualpowerloss);
printf("\nThe demand factor of the feeder %.2f",Demandfactor);
```

Scilab code Exa 4.2 Load characteristics calculation

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
```

```
//Example 4.2
//Scilab Version : 6.0.0     ; OS : Windows

clc;
clear;

max_demand=100;    //Maximum demand of generating station in MW
LF=0.65;    //Load factor of generating station in percentage
PCF=0.50;    //Plant capacity factor of generating station in percentage
PUF=0.80;    //Plant use factor of generating station in percentage
avg_demand=max_demand*LF;    //Average demand of generating station in MW
daily_energy=avg_demand*24;    //Daily energy produced by generating station in MWh
PRC=avg_demand/PCF;    //Plant rated capacity  of generating station in MW
RC=PRC-max_demand;    //Reserve capacity of generating station in MW
max_energy=PRC*24;    //Maximum energy produced if plant is running all the time in MWh
FL_max_energy=daily_energy/PUF;    //Maximum energy produced if plant is running at full load in MWh
UF=max_demand/PRC;    //Utilization factor of generating station


printf(”\nDaily energy produced %.f MWh”, daily_energy);
```



```
Class contribution factor for street lightning is 1.0 and the remaining load is 1.0
Diversity factor of the feeder 1.0
Coincidence factor of the load group 1.0
-->
```

Figure 4.3: Load characteristics calculation

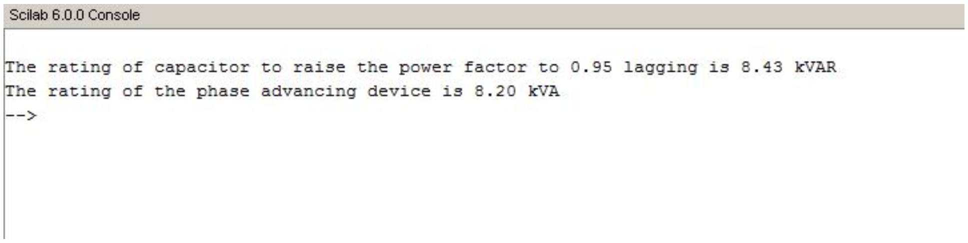
```
printf("\nInstalled capacity of plant %.f MW",PRC);
printf("\nReserve capacity of plant %.f MW",RC);
printf("\nMaximum energy that could be produced if
   the plant is running all the time %.f MWh",
   max_energy);
printf("\nMaximum energy that could be produced if
   the plant is running at full load %.f MWh",
   FL_max_energy);
printf("\nUtilization factor %.3f",UF);
```

Scilab code Exa 4.3 Load characteristics calculation

```
//Electric Power Generation, Transmission and
   Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 4.3
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


```

```
peak_dem_light=200;                                          //Peak demand of the light load in kW
max_dem_light=200;                                           //Maximum demand of the light load in kW
max_dem_rest=1800;                                           //Maximum demand of the rest load in kW
peak_dem_rest=1800;                                          //Peak demand of the rest load in kW
c_light=peak_dem_light/max_dem_light;                        //Contribution factor for street lighting load
c_rest=peak_dem_rest/max_dem_rest;                           //Contribution factor for street rest load
DF=(peak_dem_light+peak_dem_rest)/(c_light*max_dem_light+c_rest*max_dem_rest);    //Diversity factor of the feeder
CF=1/DF;                                                     //Coincidence factor of the load group

printf("\nClass contribution factor for street lightning is %.1f and the remaining load is %.1f",c_light,c_rest);
printf("\nDiversity factor of the feeder %.1f",DF);
printf("\nCoincidence factor of the load group %.1f",CF);
```

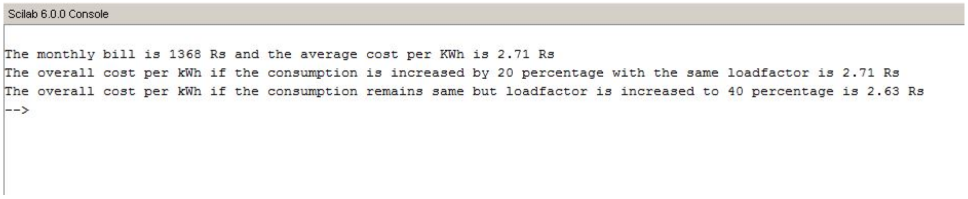
Scilab code Exa 4.4 Economics of power factor correction



```
The rating of capacitor to raise the power factor to 0.95 lagging is 8.43 kVAR
The rating of the phase advancing device is 8.20 kVA
-->
```

Figure 4.4: Economics of power factor correction

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 4.4
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


P=20;                                             //Load in kW
pf1=0.8;                                          //Actual Power factor
pf2=0.95;                                         //Required Power factor
phi1=acos(pf1);                                   //Actual Power factor angle in degree
phi2=acos(pf2);                                   //Required Power factor angle in degree
S1=P/pf1;                                         //Actual Apparent Power in kVA
S2=P/pf2;                                         //Modified Apparent Power in kVA
C_VAR=S1*sin(phi1)-S2*sin(phi2);                  //Required rating of the Capacitor in kVAR
phi3=acos(0.1);                                   //Power
```



```
The monthly bill is 1368 Rs and the average cost per KWh is 2.71 Rs
The overall cost per kWh if the consumption is increased by 20 percentage with the same loadfactor is 2.71 Rs
The overall cost per kWh if the consumption remains same but loadfactor is increased to 40 percentage is 2.63 Rs
-->
```

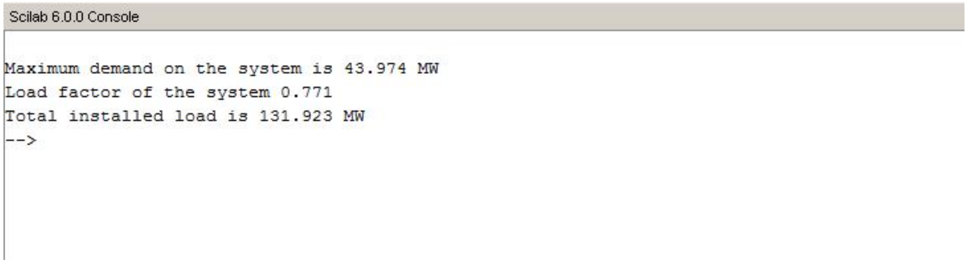
Figure 4.5: Economics of power factor correction

```
factor Angle of Phase Advancing device in degree
alpha=phi1-phi2;                                    //Angle in degree
Beta=%pi/2-acos(0.1)+%pi-(phi1+%pi/2);              //Angle in degree
del=%pi-(Beta+alpha);                               //Angle in degree
ph_adv_KVA=S1*sin(alpha)/sin(del);                  //Apparent Power of the Phase advancing device  in kVA

printf("\nThe rating of capacitor to raise the power factor to 0.95 lagging is %.2f kVAR",C_VAR);
printf("\nThe rating of the phase advancing device is %.2f kVA",ph_adv_KVA);
```

Scilab code Exa 4.5 `Economics of power factor correction`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 4.5
//Scilab Version : 6.0.0      ; OS : Windows

```

```
clc;
clear;


LF=0.35;                                    //Load factor in percentage
mon_consumption=504;                        //Monthly consumption in kWh
max_dem_rate=180;                           //Maximum demand  per kWh in Rs
Unit_rate=2.00;                             //Unit rate of electricity  per kWh in Rs
max_dem1=mon_consumption/(LF*24*30);                          //Maximum demand of consumer in kW
mon_bill1=(max_dem1*max_dem_rate)+(Unit_rate*mon_consumption);           //Monthly bill of consumer in Rs
overall_cost1=mon_bill1/mon_consumption;                      //Overall cost of consumer in Rs
new_consumption=mon_consumption*1.20;                         //New consumption of consumer in kWh
max_dem2=new_consumption/(LF*24*30);                          //Maximum demand of same Load factor in kW
mon_bill2=(max_dem2*max_dem_rate)+(Unit_rate*new_consumption);           //Monthly bill of consumer in Rs
overall_cost2=mon_bill2/new_consumption;                      //Overall cost of consumer in Rs
max_dem3=mon_consumption/(0.40*24*30);
```



```
Maximum demand on the system is 43.974 MW
Load factor of the system 0.771
Total installed load is 131.923 MW
-->
```

Figure 4.6: Maximum and minimum calculation

```
                                                //Maximum demand of increased load factor in kW
mon_bill3=(max_dem3*max_dem_rate)+(Unit_rate*mon_consumption);        //Monthly bill of consumer in Rs
overall_cost3=mon_bill3/mon_consumption;                                    //Overall cost of consumer in Rs

printf("\nThe monthly bill is %.f Rs and the average cost per KWh is %.2f Rs",mon_bill1,overall_cost1);
printf("\nThe overall cost per kWh if the consumption is increased by 20 percentage with the same load factor is %.2f Rs",overall_cost2);
printf("\nThe overall cost per kWh if the consumption remains same but loadfactor is increased to 40 percentage is %.2f Rs",overall_cost3);
```

Scilab code Exa 4.6 `Maximum and minimum calculation`

```
//Electric Power Generation, Transmission and
```

```
   Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 4.6
//Scilab Version : 6.0.0     ; OS : Windows


clc;
clear;


k=0.6;
t=1.503032/0.6;
   //Time in hours
Df=3;
   //Density factor
P=30- 8*sin(k*t)+0.325*t;
   //Load variation at a power supply station in MW
i=1;
n=1;
while (t(i)<24)
t(i+1)=(2*n*%pi-1.503032)/0.6;
t(i+2)=(2*n*%pi+1.503032)/0.6;
if (t(i+1)<24)&(t(i+2)<24) then
    i=i+2;
else
    t(i+1)=25;
    i=i+1;
end
n=n+1;
end
P=30- 8*sin(k*t)+0.325*t;
Max_demand=max(P);
   //Maximum demand on the system in MW
```

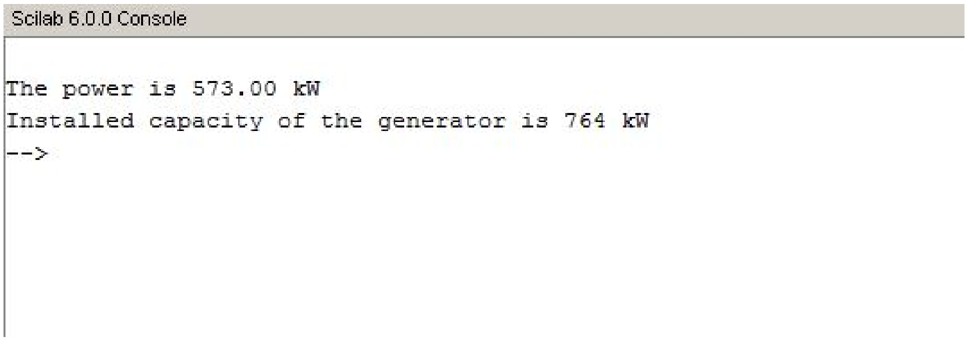
```
Avg_load=(1/24)*(30*24+(8/0.6)*(cosd(0.6*24)-cosd(0.6*0))+0.325*24^(2)/2);     //Applying integration for power equation
Lf=Avg_load/Max_demand;                                        //Load factor of the system
Total_load=Max_demand*Df;                                      //Total installed load of the system in MW


printf("\nMaximum demand on the system is %.3f MW",Max_demand);
printf("\nLoad factor of the system %.3f",Lf);
printf("\nTotal installed load is %.3f MW",Total_load);
```

Chapter 6

Hydroelectric Power Plants

Scilab code Exa 6.1 `Power calculation`

```
//Electric Power Generation, Transmission and
   Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 6.1
//Scilab Version : 6.0.0      ; OS : Windows

```



```
The power is 573.00 kW
Installed capacity of the generator is 764 kW
-->
```

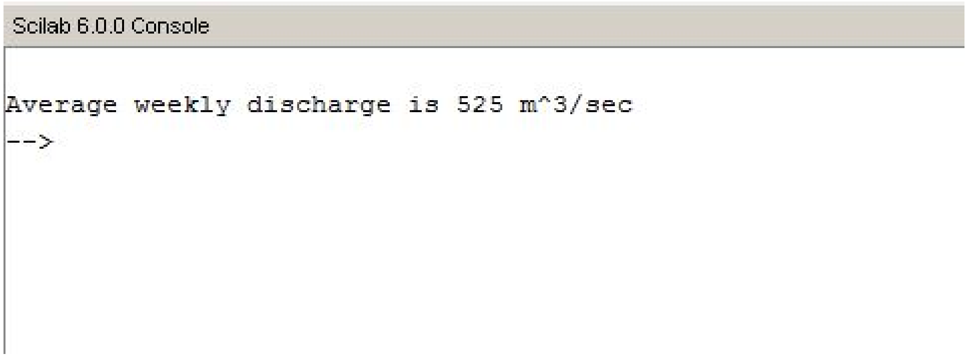
Figure 6.1: Power calculation

```
clc;
clear;


reser_catch_area=50;                //Catchment area of reservoir in km^2
avg_rainfall=150;                   //Average rainfall in cm/year
station_head=40;                    //Mean head of station in m
UF=0.75;                            //Utilization factor
LF=0.75;                            //Load factor
tur_eff=0.88;                       //Efficiency of turbine
gen_eff=0.93;                       //Efficiency of generator
water_volume=reser_catch_area*10^6*1.5*UF;                //Available water for electricity production in m^3
Q=water_volume/(365*24*60*60);                            //Available quantity in m^3/sec
P=(0.736/75)*Q*1000*station_head*tur_eff*gen_eff;         //Power of station in kW
install_cap_gen=P/LF;               //Generator installed capacity in kW

printf("\nThe power is %.2f kW",P);
```

```
printf("\nInstalled capacity of the generator is %.f kW",install_cap_gen);

```

Scilab code Exa 6.2 `Average weekly discharge calculation`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 6.2
```



```
Average weekly discharge is 525 m^3/sec
-->
```

Figure 6.2: Average weekly discharge calculation

```
//Scilab Version : 6.0.0     ; OS : Windows



clc;
clear;
clf;



q=[500 500 350 200 300 800 1100 900 400 200 0];
t=0:1:10;
subplot(3,1,1);
title(”Hydrograph”);
xlabel(”Time (Weeks)”);
ylabel(”Q (m3/Sec)”);
plot2d2(t,q);
Avg=sum(q)/max(t);                    //Average Discharge in a Week in m^3/sec
percent=[0 1100];
j=1;
for temp=1100:-200:100
    count=0;
    for i=1:1:11
        if q(i) >= temp then
```

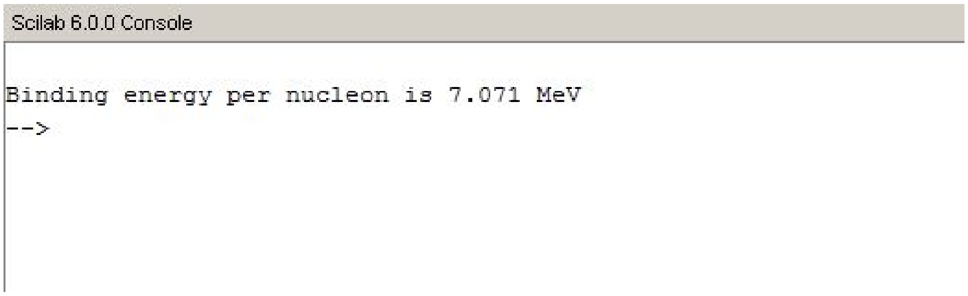
```
           count=count+1;
       else
           count=count+0;
        end
    end
    j=j+1;
    percent(j,:)=[count*10 temp];
end
subplot(3,1,2);
title(”Flow duration curve”);
xlabel(”Percentage of time”);
ylabel(”Q (m3/Sec)”);
plot2d(percent(:,1),percent(:,2));
y=cumsum(7*q);
subplot(3,1,3);
title(”Mass curve”);
xlabel(”Time (Weeks)”);
ylabel(”Cumulative flow(day−sec−metre)”);
plot2d([1:1:10],resize_matrix(y,-1,10),rect=[0 0 11 40000]);


printf(”\nAverage weekly discharge is %.f m^3/sec”,Avg);
```

Chapter 7

Nuclear Power Plants

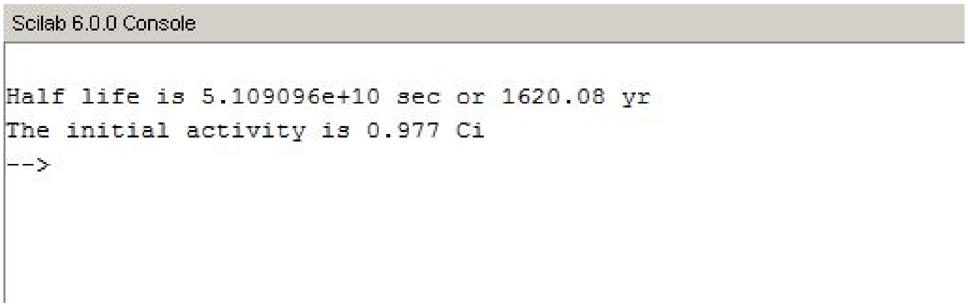
Scilab code Exa 7.1 `Compute binding energy`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 7.1
//Scilab Version : 6.0.0     ; OS : Windows

clc;
```



```
Binding energy per nucleon is 7.071 MeV
-->
```

Figure 7.1: Compute binding energy



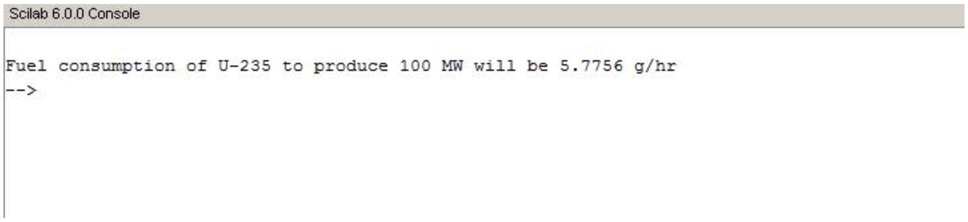
```
Half life is 5.109096e+10 sec or 1620.08 yr
The initial activity is 0.977 Ci
-->
```

Figure 7.2: Half life and Activity Calculation

```
clear;


helium_atomic_mass=4.002603;                    //Atomic
    mass of Helium in amu
mp=1.007277;                                    //Atomic
    mass of Proton in amu
mn=1.008665;                                    //Atomic
    mass of Neutron in amu
me=0.00055;                                     //Atomic
    mass of Electron in amu
del_m=2*mp+2*me+2*mn-helium_atomic_mass;        //Mass
    Defect in amu
Be=del_m*931;                                   //Helium
    Binding Energy in MeV
Be_molecule=Be/4;                               //Helium
    Binding Energy per Nucleon in MeV


printf("\nBinding energy per nucleon is %.3f MeV",
    Be_molecule);
```

Scilab code Exa 7.2 `Half life and Activity Calculation`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 7.2
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


rad_atomic_mass=226.095;                                    //Atomic Mass of Radium in amu
rad_decay_const=1.3566*10^-11;                              //Decay Constant of Radium in 1/s
Half_life=0.6931/rad_decay_const;                           //Radium Half Life in sec
Half_life_yr=Half_life/(365*24*60*60);                      //Radium Half Life in year
N=6.023*10^23/rad_atomic_mass;                              //Number of atoms per gram of Radium
Activity=rad_decay_const*N;                                 //Activity of Radium in disintegration/second
Activity_curi=Activity/(3.7*10^10);                         //Activity of Radium in Ci


printf("\nHalf life is %e sec or %.2f yr",Half_life,Half_life_yr);
printf("\nThe initial activity is %.3f Ci",Activity_curi);
```



```
Fuel consumption of U-235 to produce 100 MW will be 5.7756 g/hr
-->
```

Figure 7.3: Compute Fuel Consumption

Scilab code Exa 7.3 `Compute Fuel Consumption`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 7.3
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


neu_absor=0.80;                                    //Absorbed Neutrons of Uranium_235 in percentage
P=100;                                                  //Power of Uranium_235 in kW
use_energy=190;                                    //Useful Energy of Uranium_235 in MeV
energy=use_energy*10^6*1.60*10^-19;          //Fission Energy of Uranium_235
```

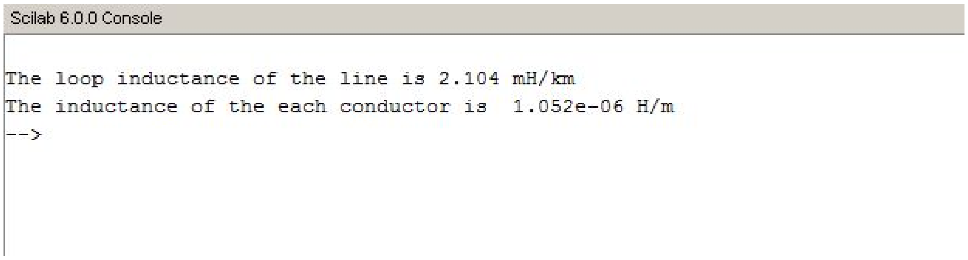
```
 in J
fission_energy=1/energy;                                //Number of Fission to Produced One Joule of Energy
nuclei_power=fission_energy*3600*10^6/neu_absor;        //Number of Nuclei Burnt during 1 hour per MW of Power
Mass=nuclei_power*235/(6.023*10^23);                    //Mass of Uranium_235  to produce required Power in g/hr


printf("\nFuel consumption of U-235 to produce 100 MW will be %.4f g/hr",Mass*100);
```

Chapter 10

Transmission Line Parameters Calculations

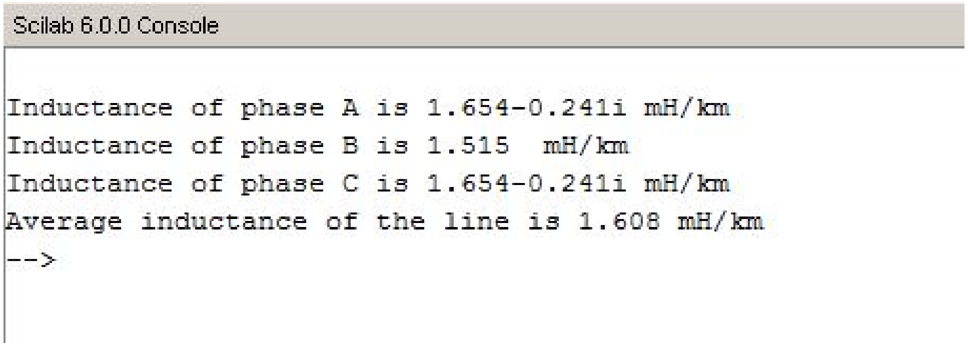
Scilab code Exa 10.3 `Compute Transmission line parameters`

```
//Electric Power Generation, Transmission and
    Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 10.3
//Scilab Version : 6.0.0     ; OS : Windows

```



```
The loop inductance of the line is 2.104 mH/km
The inductance of the each conductor is  1.052e-06 H/m
-->
```

Figure 10.1: Compute Transmission line parameters



```
Inductance of phase A is 1.654-0.241i mH/km
Inductance of phase B is 1.515  mH/km
Inductance of phase C is 1.654-0.241i mH/km
Average inductance of the line is 1.608 mH/km
-->
```

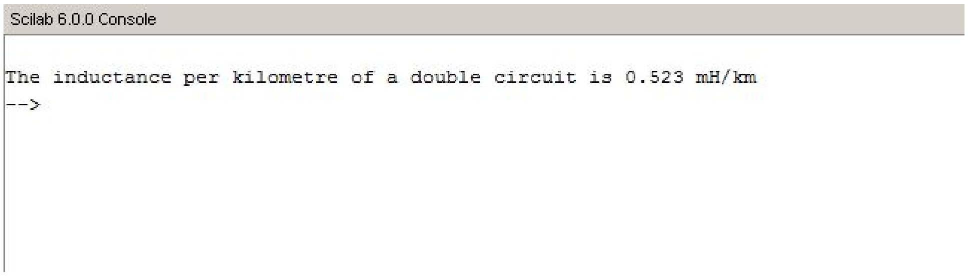
Figure 10.2: Compute Transmission line parameters

```
clc;
clear;


d=3;                                     //Distance of two Wires in m
r=0.02;                                  //Radius of conductor in m
L1=(2*10^(-7))*(log(d/(0.7788*r)));      //Inductance of each conductor in H/m
LI=2*L1/(10^(-6));                       //Loop inductance of the line in mH/km


printf("\nThe loop inductance of the line is %.3f mH/km",LI);
printf("\nThe inductance of the each conductor is %.3e H/m",L1);
```

Scilab code Exa 10.5 `Compute Transmission line parameters`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 10.5
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


r=0.03;                                     //Conductor Radius in m
d=0.35;                                     //Spacing between Phase Conductors in m
D=4;                                        //Distance between Phases in m
LA=((1*10^(-7))*((log((D*(D+d)*2*D*(2*D+d)*D*(D-d)*2*D*(2*D-d))^(1/2)/(0.7788*r*d)^2))+(%i*0.866)*log((D*(D+d)*D*(D-d))/(2*D*(2*D+d)*2*D*(2*D-d)))))*10^(6); //Phase A Inductance in mH/km
LB=(1*10^(-7))*((log(((D*(D-d)*D*(D+d)*D*(D+d)*D*(D-d))^(1/2))/(0.7788*r*d)^2))+(%i*0.866)*log((D*(D+d)*D*(D-d))/(D*(D-d)*D*(D+d))))*10^(6);                //Phase B Inductance in mH/km
LC=LA;                                      //Phase C Inductance in mH/km
L_avg=(LA+LB+LC)/3;                         //Average Inductance in mH/km


printf("\nInductance of phase A is %.3f%.3fi mH/km",real(LA),imag(LA));
```

```
printf("\nInductance of phase B is %.3f  mH/km",LB);
printf("\nInductance of phase C is %.3f%.3fi mH/km",real(LC),imag(LC));
printf("\nAverage inductance of the line is %.3f mH/km",L_avg);

//Varia prese in resul due to wrong calcu of LA and LB value
```



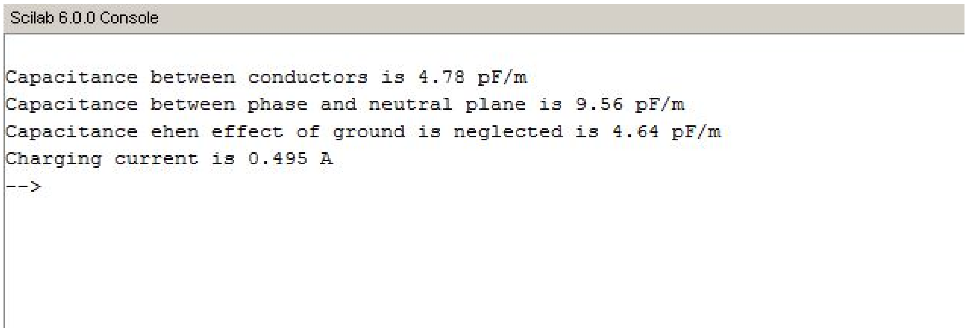
```
The inductance per kilometre of a double circuit is 0.523 mH/km
-->
```

Figure 10.3: Compute Transmission line parameters

Scilab code Exa 10.6 `Compute Transmission line parameters`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 10.6
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


r=0.025;                                    //Conductor radius in m
Dac1=4;                                     //Distance between two conductors a & c1 in m
Dac=6;                                      //Distance between two conductors a & c in m
Dbb1=10;                                    //Distance between two conductors b & b1 in m
Dab=(((Dbb1-Dac1)/2)^2+(Dac/2)^2)^(1/2);    //
```

```
   Distance between two conductors a & b in m
Daa1=(((Dac1)^2)+((Dac)^2))^(1/2);
                                                        //
   Distance between two conductors a & a1 in m
Dbc1=((((Dbb1-Dac1)/2)^2+((Dac1)+((Dbb1-Dac1)/2))^2)
   ^(1/2);                        //Distance between
    two conductors b & c1 in m
GMD1=nthroot((Dab*Dac*Dbc1*Dac1),4);
                                                      //
   Mutual GMD of phase a position 1 in m
GMR1=(0.7788*r*Daa1)^(1/2);

   //Self GMR of phase a position 1 in m
GMD2=nthroot((Dab*Dab*Dbc1*Dbc1),4);
                                                      //
   Mutual GMD of phase a position 2 in m
GMR2=(0.7788*r*Dbb1)^(1/2);

   //Self GMR of phase a position 2 in m
GMD3=GMD1;

   //Mutual GMD of phase a position 3 in m
GMR3=GMR1;

   //Self GMR of phase a position 3 in m
Dm=nthroot((GMD1*GMD2*GMD3),3);

   //Equivalent mutual GMD in m
Ds=nthroot((GMR1*GMR2*GMR3),3);

   //Equivalent self GMR in m
LA=(2/10)*(log(Dm/Ds));

   //Inductance of phase a in mH/km



printf("\nThe inductance per kilometre of a double
   circuit is %.3f mH/km",LA);
```



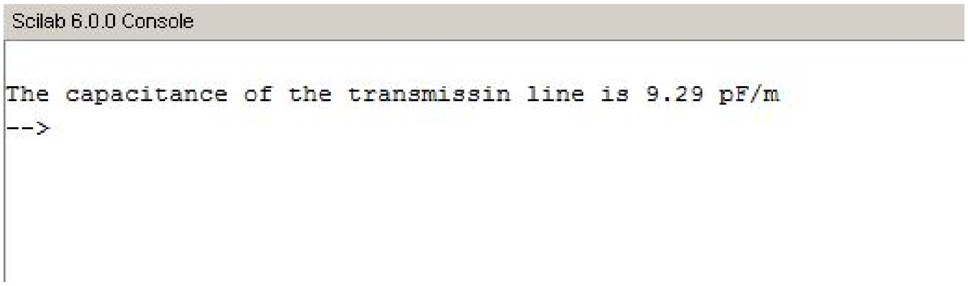
```
Capacitance between conductors is 4.78 pF/m
Capacitance between phase and neutral plane is 9.56 pF/m
Capacitance ehen effect of ground is neglected is 4.64 pF/m
Charging current is 0.495 A
-->
```

Figure 10.4: Compute Transmission line parameters

Scilab code Exa 10.7 `Compute Transmission line parameters`

```
//Electric Power Generation, Transmission and
    Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 10.7
//Scilab Version : 6.0.0     ; OS : Windows

clc;
clear;


h=8;

    //Height of conductor in m
r=0.01;

    //Radius of conductor in m
D=4;
```

```
//Distance of conductor in m
L=10;
//Length of the line in km
V=33;
//Supply voltage in kV
f=50;
//Supply frequency in Hz
Cab=(%pi*(10^(-9))/(36*%pi))/(log(D/r)*(1/(sqrt(1+(D^(2)/(2*h)^(2))))));      //Capacitance between conductors a and b in pF/m
Can=Cab*2;
//Capacitance between phase and neutral plane in pF/m
Cab1=(%pi*(10^(-9))/(36*%pi))/log(D/r);                                       //Capacitance between conductors when effect of earth is ignored in pF/m
Ic=2*%pi*f*Cab*L*V*10^3*10^3;                                                 //Charging Current of the line in A


printf("\nCapacitance between conductors is %.2f pF/m",Cab/(10^-12));
printf("\nCapacitance between phase and neutral plane is %.2f pF/m",Can/(10^-12));
printf("\nCapacitance ehen effect of ground is neglected is %.2f pF/m",Cab1/(10^-12));
printf("\nCharging current is %.3f A",Ic);
```

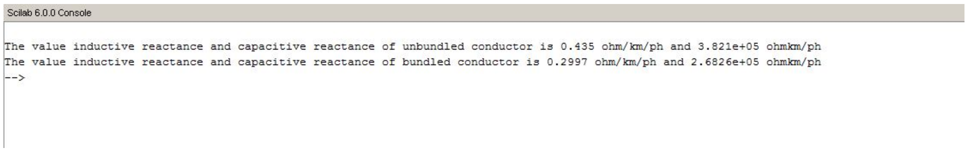


```
The capacitance of the transmissin line is 9.29 pF/m
-->
```

Figure 10.5: Compute Transmission line parameters

Scilab code Exa 10.8 `Compute Transmission line parameters`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 10.8
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


d=0.25;                //Diameter of conductor in m
r=0.0125;              //Radius of conductor in m
Dab=5;                 //Distance between conductors a & b in m
Dbc=4;
```



```
The value inductive reactance and capacitive reactance of unbundled conductor is 0.435 ohm/km/ph and 3.821e+05 ohmkm/ph
The value inductive reactance and capacitive reactance of bundled conductor is 0.2997 ohm/km/ph and 2.6826e+05 ohmkm/ph
-->
```

Figure 10.6: Compute Transmission line parameters

```
    //Distance between conductors b & c in m
Dac=6;    //Distance between conductors a & c in m
Deq=nthroot((Dab*Dbc*Dac),3);                //Diameter equivalent of line in m
Can=(2*%pi*10^(-9)/(36*%pi))/log(Deq/r);          //Capacitance between phase a & neutral in pF/m


printf("\nThe capacitance of the transmissin line is %.2f pF/m",Can/(10^-12));
```

Scilab code Exa 10.9 `Compute Transmission line parameters`

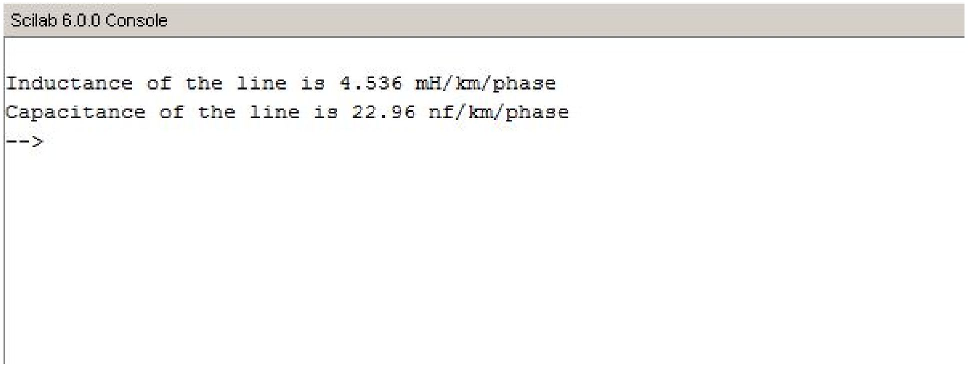
```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 10.9
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;
```

```


f=50;                 //Frequency of the condutor in Hz
D1=31.8;              //Diameter of ACSR Moose conductor in mm(Unbundled conductor)
d1=10;                //Hoizontal spacing between adjacent conductors in m
D2=19.6;              //Diameter of ASCR Lynx conductor in mm(Bundled conductor)
d2=10;                //Spacing measured by centre of the bundle in m
Db=0.4;               //Spacing between the bundled conductors in m
r1=D1*10^-3/2;        //Radius of unbundled conductor in m
Dm=nthroot((d1*d2*(d1+d2),3));    //Mutual GMD of unbundled conductorin m
Dls1=0.7788*r1;       //GMR For Inductance of unbundled conductor in m
Dcs1=r1;              //GMR For Capacitance of unbundled conductor in m
XL1=2*%pi*f*2*10^(-4)*log(Dm/Dls1);    //Inductive Reactance  of unbundled conductor in ohm/km/phase
```

```
XC1=(log(Dm/Dcs1))/(((2*%pi)^2)*f*8.85*10^(-12));        //Capacitive Reactance of unbundled conductor in ohm-km/phase
r2=D2*10^-3/2;                                          //Radius of bundled conductor in m
Dls2=nthroot((0.7788*r2*Db^2),3);                    //Self GMR for Inductance of bundled conductor in m
Dcs2=nthroot((r2*Db^2),3);                              //Self GMR for Capacitance of bundled conductor in m
XL2=2*%pi*f*2*10^(-4)*log(Dm/Dls2);                  //Inductive Reactance of bundled conductor in ohm/km/phase
XC2=(log(Dm/Dcs2))/(((2*%pi)^2)*f*8.85*10^(-12));        //Capacitive Reactance of bundled conductor in ohm-km/phase


printf("\nThe value inductive reactance and capacitive reactance of unbundled conductor is %.3f ohm/km/ph and %.3e ohmkm/ph ",XL1,XC1/10^(3));
printf("\nThe value inductive reactance and capacitive reactance of bundled conductor is %.4f ohm/km/ph and %.4e ohmkm/ph ",XL2,XC2/10^(3));
```

Scilab code Exa 10.10 `Compute Transmission line parameters`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
```



```
Inductance of the line is 4.536 mH/km/phase
Capacitance of the line is 22.96 nf/km/phase
-->
```

Figure 10.7: Compute Transmission line parameters

```
//Year: 2012 ; Edition - 2
//Example 10.10
//Scilab Version : 6.0.0       ; OS : Windows

clc;
clear;


r=0.05;
    //Radius of the conductor in m
bc=5;
    //Distance between b & c in m
bb1=3*bc;
    //Distance between b & b1 in m
aa1=2*bc;
    //Distance between a & a1 in in m
ab=(((bb1-aa1)/2)^(2)+bc^(2))^(1/2);
ab1=((aa1+((bb1-aa1)/2))^(2)+bc^(2))^(1/2);
ac1=((bc*2-((bb1-aa1)/2))^(2)+bc^(2))^(1/2);
Dab=nthroot(ab*ab1*ab1*ab,4);
Dbc=nthroot(bc*aa1*aa1*bc,4);
Dca=nthroot(ab*ac1*ab*ac1,4);
Deq=nthroot(Dab*Dbc*Dca,3);
Dsa=sqrt(aa1*0.7788*r);
Dsb=sqrt(bb1*0.7788*r);
```

```
Dsc=sqrt(bc*0.7788*r);
Ds=nthroot(Dsa*Dsb*Dsc,3);
L=(2*10^(-7)*log(Deq/Ds))*10^(6);
    //Inductance of double circuit in mH/km/phase
Dsa1=sqrt(aa1*r);
    //GMR for capacitance
Dsb1=sqrt(bb1*r);
    //GMR for capacitance
Dsc1=sqrt(bc*r);
    //GMR for capacitance
Ds1=nthroot(Dsa1*Dsb1*Dsc1,3);
    //Equivalent GMR for capacitance
C=(2*%pi*(10^(-9)/(36*%pi)))/log(Deq/Ds1);
    //Capacitance of double circuit in nF/km/phase


printf("\nInductance of the line is %.3f mH/km/phase
    ",L);
printf("\nCapacitance of the line is %.2f nF/km/
    phase",C/10^(-12));
                                                                        //
                                                                          V

                                                                          p

                                                                          i

                                                                          r

                                                                          d

                                                                          t

                                                                          w

                                                                          c

                                                                          o
```

D

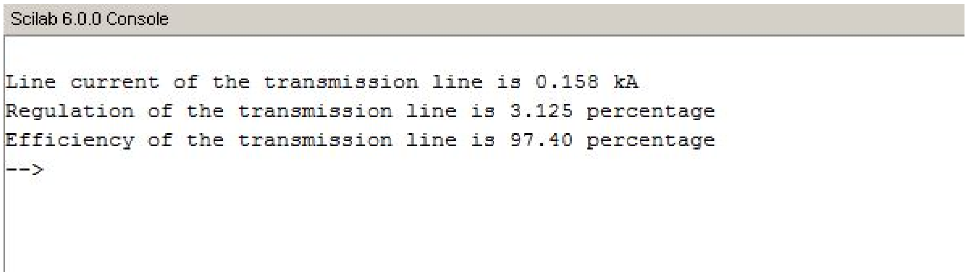
v

Chapter 11

Analysis of Transmission Lines

Scilab code Exa 11.2 `Compute Transmission line parameters`

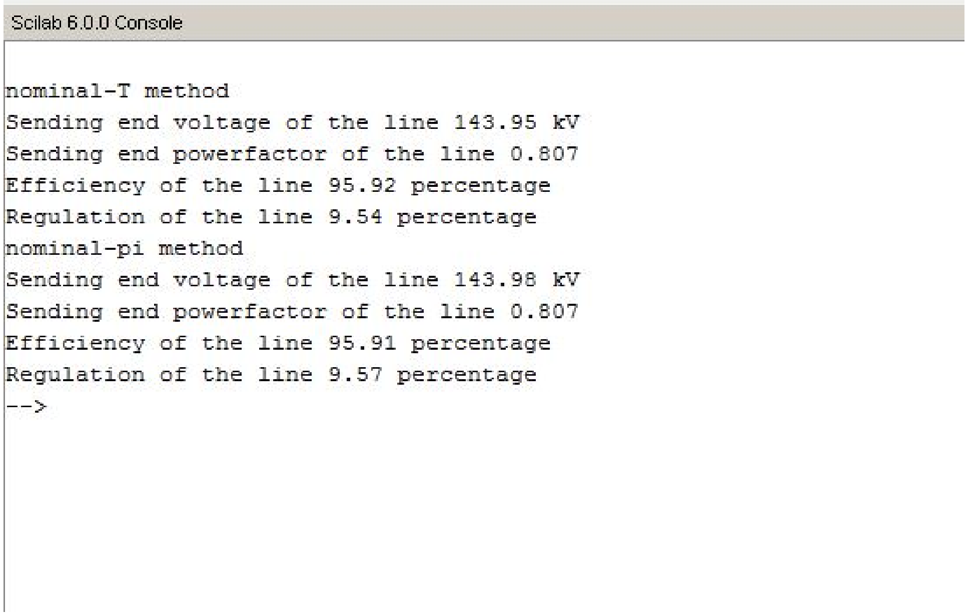
```
//Electric Power Generation, Transmission and
   Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 11.2
//Scilab Version : 6.0.0      ; OS : Windows

clc;
```



```
Line current of the transmission line is 0.158 kA
Regulation of the transmission line is 3.125 percentage
Efficiency of the transmission line is 97.40 percentage
-->
```

Figure 11.1: Compute Transmission line parameters

```
clear;


r=0.5;                                      //Resistance of the line in Ohm/km
l=5;                                        //Length of the line in km
L=1.76;                                     //Inductance of the line in mH/km
f=50;                                       //Supply frequency in Hz
sen_vtg=33;                                 //Sending end voltage in kV
rec_vtg=32;                                 //Receiving end voltage in kV
Vs=sen_vtg/(3)^(1/2);                       //Sending end phase voltage in kV
Vr=rec_vtg/(3)^(1/2);                       //Receiving end phase voltage in kV
R=r*l;                                      //Total resistance of line in Ohm
X=2*(%pi)*f*L*l*10^(-3);                    //Total Inductance of line in Ohm
pf1=0.8;                                    //Power factor
A=X^(2)+R^(2);                              //Coeffcient of Ir^(2) simlified in quadratic eqn
B=2*Vr*(R*pf1+X*sin(acos(pf1)));            //Coeffcient of Ir simlified in quadratic eqn
C=Vr^(2)-Vs^(2);                            //Constant simlified in quadratic eqn
Ir=(-B+sqrt(B^(2)-4*A*C))/(2*A);            //Receiving end current in A
reg=((Vs-Vr)/Vr)*100;                       //Efficiency of the line
P=3*Vr*Ir*pf1;                              //Output power in MW
Loss=3*Ir^(2)*R;                            //Line
```



```
nominal-T method
Sending end voltage of the line 143.95 kV
Sending end powerfactor of the line 0.807
Efficiency of the line 95.92 percentage
Regulation of the line 9.54 percentage
nominal-pi method
Sending end voltage of the line 143.98 kV
Sending end powerfactor of the line 0.807
Efficiency of the line 95.91 percentage
Regulation of the line 9.57 percentage
-->
```

Figure 11.2: Compute Transmission line parameters

```
loss in MW
eff=(P/(P+Loss))*100;                          //Efficiency of the line


printf("\nLine current of the transmission line is %.3f kA",Ir);
printf("\nRegulation of the transmission line is %.3f percentage ",reg);
printf("\nEfficiency of the transmission line is %.2f percentage",eff);
```

Scilab code Exa 11.3 `Compute Transmission line parameters`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 11.3
//Scilab Version : 6.0.0     ; OS : Windows

clc;
clear;


P=50;                                   //Power of the line in MW
l=100;                                  //Length of the line in km
pf=0.8;                                 //Power factor
V=132;                                  //Voltage of the line in kV
R=0.1;                                  //Resistance of the conductor in Ohm/km
X=0.3;                                  //Reactance of the conductor in Ohm/km
y=3*10^(-6);                            //Admittance of the conductor in mho/km
Vr=V/(3)^(1/2);                         //Receiving end voltage in kV
Z=(R+%i*X)*100;                         //Series impedance in Ohm
```

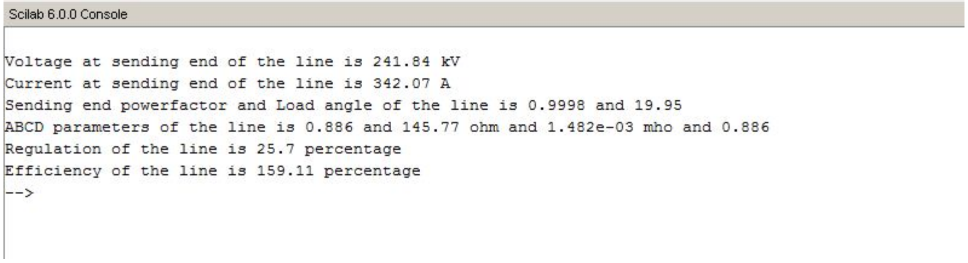
```
Y=(0.0+%i*y)*100;                                  //Shunt admittance on mho
Ir=P*10^(3)/(3*Vr*pf);                             //Receiving end current in A
Vc=Vr*(pf+%i*0.6)+(Ir*Z/2)*10^(-3);                //Capacitance voltage in kV
Ic=Y*Vc*10^(3);                                    //Shunt branch current in A
Is=Ic+Ir;                                          //Sending end current in A
Vs=Vc+(Is*Z/2)*10^(-3);                            //Sending end voltage in kV
Vsl=abs(Vs)*3^(1/2);                               //Line to line sending end voltage in kV
pf1=cos(atan(imag(Vs),real(Vs))-atan(imag(Is),real(Is)));          //Sending end power factor
Vr1=abs(Vs)/(1+(Z*Y/2));                           //Receiving end voltage at no_load in kV
reg=((abs(Vr1)-Vr)/Vr)*100;                        //Regulation of the line
eff=P*10^(6)/(P*10^(6)+3*((abs(Is)^(2)*R*l)/2+(Ir^(2)*R*l)/2))*100;          //Efficiency of the line
Ic1=(Y/2)*Vr*10^(3);                               //Capacitance 1 current in A
Il=Ir*(0.8-%i*0.6)+Ic1;
```

```
    //Line current in A
Vs1=Vr+Il*Z*10^(-3);

    //Sending end voltage in kV
Vsl1=abs(Vs1)*3^(1/2);

    //Line to line sending end voltage in kV
Ic2=((Y/2)*Vs1*10^(3));

    //Capacitance 2 current in A
Is1=Il+Ic2;

    //Sending end current in A
pf2=cos(atan(imag(Vs1),real(Vs1))-atan(imag(Is1),
    real(Is1)));                    //Power factor
V=abs(Vs1)/(1+(Z*Y/2));

    //Receiving end voltage at no_load in kV
reg1=((abs(V)-Vr)/Vr)*100;

    //Regulation of the line
eff1=(P*10^(6)/(P*10^(6)+3*(abs(Il)^(2)*R*l)))*100;
                                    //Efficiency of the line


printf("\nnominal-T method");
printf("\nSending end voltage of the line %.2f kV",
    Vsl);
printf("\nSending end powerfactor of the line %.3f",
    pf1);
printf("\nEfficiency of the line %.2f percentage",
    eff);
printf("\nRegulation of the line %.2f percentage",
    reg);
printf("\nnominal-pi method");
printf("\nSending end voltage of the line %.2f kV",
    Vsl1);
printf("\nSending end powerfactor of the line %.3f",
```

```
pf2);
printf("\nEfficiency of the line %.2f percentage",eff1);
printf("\nRegulation of the line %.2f percentage",reg1);
//Variat presen in result due to wrong calcul of Ic2 value
```

Scilab code Exa 11.4 `Compute Transmission line parameters`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
```



```
Voltage at sending end of the line is 241.84 kV
Current at sending end of the line is 342.07 A
Sending end powerfactor and Load angle of the line is 0.9998 and 19.95
ABCD parameters of the line is 0.886 and 145.77 ohm and 1.482e-03 mho and 0.886
Regulation of the line is 25.7 percentage
Efficiency of the line is 159.11 percentage
-->
```

Figure 11.3: Compute Transmission line parameters

```
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 11.4
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;



f=50;
    //Supply frequency in Hz
l=450;
    //Length of the line in km
V=400;
    //Supply voltage in kV
R=0.033;
    //Resistance of the line in Ohm/km
L=1.067;
    //Inductance of the line in mH/km
C=0.0109;
    //Capacitance of the line in microFarad/km
```

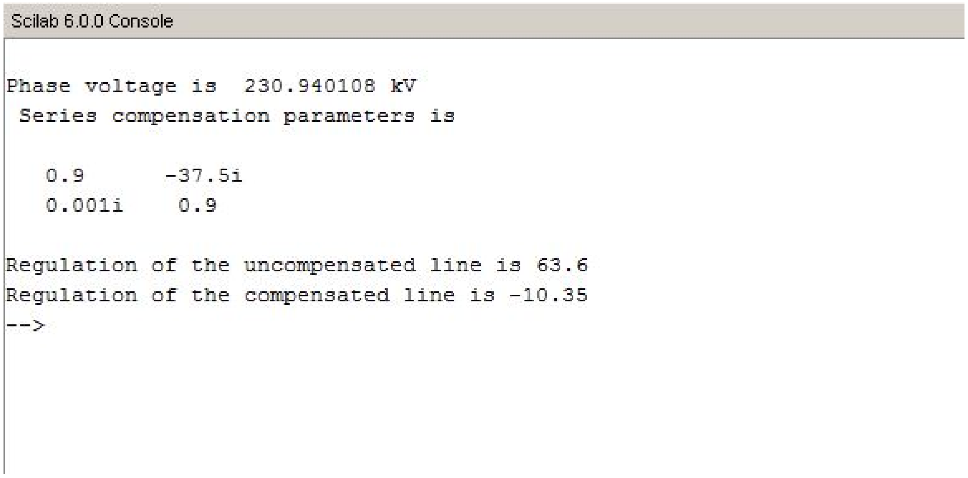
```
P=420;                                    //Power in MW
pf=0.95;                                  //Power factor
Z=R+%i*(2*%pi*f*L*10^(-3));               //Impedance of the line in Ohm/km
Y=%i*(2*%pi*f*C);                         //Admittance of the line in mho/km
Zc=((Z/Y)^(1/2))*10^(3);                  //Characteristic impedance of the line in Ohm/km
pro_const=(Z*Y)^(1/2);                    //Propagation constant  of the line
angle=pro_const*l*10^(-3);
s=sinh(angle);                            //Sinusoidal angle
c=cosh(angle);                            //Cosine angle
Ir=P*10^(6)/((3)^(1/2)*V*10^(3)*pf);      //Magnitude of receiving end current in A
Ir1=(Ir*(cosd(-acosd(pf))+%i*sind(-acosd(pf))))*10^(-3);       //Receiving end current including power factor angle
Vr=V/(3)^(1/2);                           //Receiving end phase voltage in kV
Vs=Vr*c+(Zc*Ir1*s);                       //Sending end voltage in kV
llv=abs(Vs)*sqrt(3);
```

```
                                                    //Line to line voltage in kV
Is=((Vr*10^(3)/Zc)*s)+(Ir1*c);                                       //Sending end current in A
pfs=cosd(atan(imag(Vs),real(Vs))-atan(imag(Is),real(Is)));      //Sending end power factor
delta=atand(imag(Vs),real(Vs));                                      //Load angle in degree
A=cosh(angle);                                                    //Parameter of voltage and current eqn in degree
B=Zc*sinh(angle);                                                    //Parameter of voltage and current eqn in Ohm
C=sinh(angle)/Zc;                                                    //Parameter of voltage and current eqn in mho
D=A;                                                              //Parameter of voltage and current eqn in degree
reg=(((abs(Vs)/abs(A))-Vr)/Vr)*100;                                  //Regulation of the line
inp_pow=(3*abs(Vs)*abs(Is)*pfs)*10^(-3);                             //Input power in MW
eff=(P/inp_pow)*100;                                                 //Efficiency of the line

printf("\nVoltage at sending end of the line is %.2f kV",Vs);
printf("\nCurrent at sending end of the line is %.2f A",abs(Is));
printf("\nSending end powerfactor and Load angle of the line is %.4f and %.2f",pfs,delta);
printf("\nABCD parameters of the line is %.3f and %
```

```
.2f ohm and %.3e mho and %.3f ",A,abs(B),abs(C),D);
printf("\nRegulation of the line is %.1f percentage",reg);
printf("\nEfficiency of the line is %.2f percentage",eff);
                                                                    //
```

Scilab code Exa 11.5 `Compute Transmission line parameters`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
```

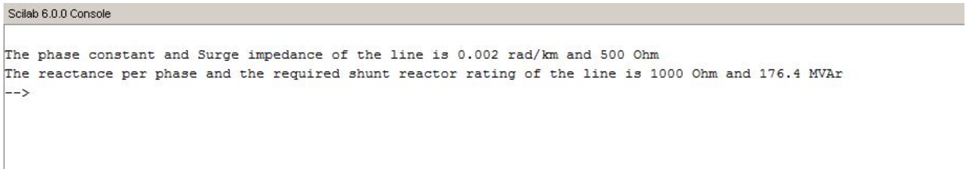


```
Phase voltage is  230.940108 kV
 Series compensation parameters is

   0.9       -37.5i
   0.001i     0.9

Regulation of the uncompensated line is 63.6
Regulation of the compensated line is -10.35
-->
```

Figure 11.4: Compute Transmission line parameters

```
//Year: 2012 ; Edition - 2
//Example 11.5
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


V=400;
    //Supply voltage in kV
Load=750;
    //Load in MVA
A=0.85;
    //Loss less three phase line constant
B=%i*150;
    //Loss less three phase line constant
C=%i*0.001;
```

```
    //Loss less three phase line constant
D=A;                    //Loss less three phase line constan
Vr=V/3^(1/2);           //Receiving end voltage in kV
Ir=Load*10^(3)/(3^(1/2)*V);                             //Receiving end current in A
Vs=(A*Vr*10^(3)+B*Ir*(0.8-%i*0.6))*10^(-3);             //Sending end voltage in kV
Is=C*Vr*10^(3)+A*Ir*(0.8-%i*0.6);                       //Sending end current in A
vtg_reg=(((abs(Vs)/abs(A))-Vr)/Vr)*100;                 //Regulation of the line
ABCD=[1 -50*%i;0 1]*[0.85 50*%i;0.001*%i 0.85]*[1 -50*%i;0 1];    //Matrix of compensated line
Vs1=ABCD(1,1)*Vr+ABCD(1,2)*(abs(Ir)/1000)*(0.8-%i*0.6);           //Sending end voltage of compensated line in kV
Is1=ABCD(2,1)*Vr*10^(3)+ABCD(2,2)*abs(Ir)*(0.8-%i*0.6);           //Sending end current of compensated line in A
vtg_reg2=(((abs(Vs1)/ABCD(1,1))-Vr)/Vr)*100;            //Regulation of the of compensated line


printf("\nPhase voltage is %f kV",Vr);
disp(ABCD,'Series compensation parameters is ');
printf("\nRegulation of the uncompensated line is %.1f",vtg_reg);
printf("\nRegulation of the compensated line is %.2f",vtg_reg2);
                                                                                /
```



```
The phase constant and Surge impedance of the line is 0.002 rad/km and 500 Ohm
The reactance per phase and the required shunt reactor rating of the line is 1000 Ohm and 176.4 MVAr
-->
```

Figure 11.5: Compute Transmission line parameters

Scilab code Exa 11.6 `Compute Transmission line parameters`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
```

```
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 11.6
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


Vs=420;                                                   //Supply voltage in kV
f=60;                                                     //Supply frequency in Hz
l=463;                                                    //Length of the line in km
Vr=700;                                                   //Receiving end voltage in kV
sen_end_crt=646.6;                                        //Sending end current in A
pha_con_len=acos(Vs/Vr);                                  //Phase constant length in rad
pha_const=pha_con_len/l;                                  //Phase constant in rad/km
Z=Vr*10^(3)*sin(pha_con_len)/(3^(1/2)*sen_end_crt);      //Surge impedance in Ohm
X=(sin(pha_con_len)/(1-cos(pha_con_len)))*Z;             //Reactance of the line in Ohm
shu_rat=Vs^(2)/X;                                         //Shunt reactor rating in MVAr

```

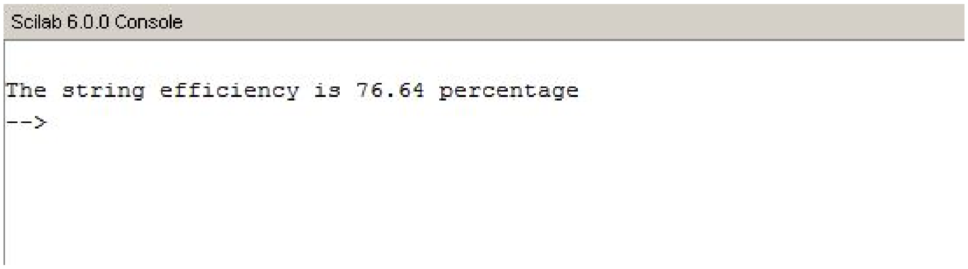
```

printf("\nThe phase constant and Surge impedance of the line is %.3f rad/km and %.f Ohm",pha_const,Z);
printf("\nThe reactance per phase and the required shunt reactor rating of the line is %.f Ohm and %.1f MVAr",X,shu_rat);
```

Chapter 12

Insulators for Overhead Transmission Lines

Scilab code Exa 12.1 `Compute String Efficiency`

```
//Electric Power Generation, Transmission and
   Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 12.1
//Scilab Version : 6.0.0      ; OS : Windows
```



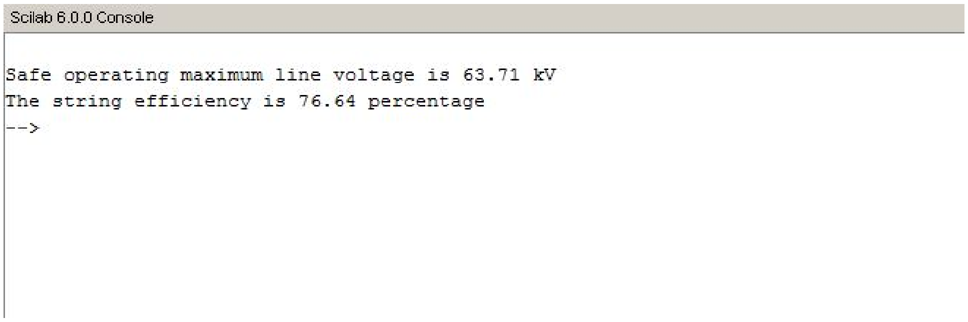
```
The string efficiency is 76.64 percentage
-->
```

Figure 12.1: Compute String Efficiency

```

clc;
clear;


V=33;                                               //Operating
    voltage of line in kV
m=10;                                               //Mutual
    capacitance of unit
n=4;                                                //No of
    string units
V_tot=V/3^(1/2);                                    //Total
    voltage across the string in kV
V1=V_tot/(1+1.1+1.31+1.651);                        //Voltage
    across the topmost unit in kV
V2=V1*(1+(1/m));                                    //Voltage
    across the second unit from the top in kV
V3=V1*(1+(3/m)+(1/m^(2)));                          //Voltage
    across the third unit from the top in kV
V4=V3*(1+(1/m))+(V2/m)+(V1/m);                      //Voltage
    across the fourth unit from the top in kV
str_eff=V_tot/(n*V4);                               //String
    efficiency in percentage


printf("\nThe string efficiency is %.2f percentage",
    str_eff*100);
```

Scilab code Exa 12.4 `Compute String Efficiency`

```
//Electric Power Generation, Transmission and
    Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
```



```
Safe operating maximum line voltage is 63.71 kV
The string efficiency is 76.64 percentage
-->
```

Figure 12.2: Compute String Efficiency

```
//Year: 2012 ; Edition - 2
//Example 12.4
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


V4=12;                                  //Rated voltage of each unit in kV
m=10;                                   //Mutual capacitance of unit
n=4;                                    //No of string units
V1=V4/1.651;                            //Voltage across the topmost unit in kV
V2=1.1*V1;                              //Voltage across the second unit from the top in kV
V3=1.31*V1;                             //Voltage across the third unit from the top in kV
V_tot=V1+V2+V3+V4;                      //Total voltage Voltage across the string in kV
mlv=3^(1/2)*V_tot;                      //Maximum line voltage in kV
str_eff=(V_tot/(n*V4))*100;             //String efficiency in percentage
```

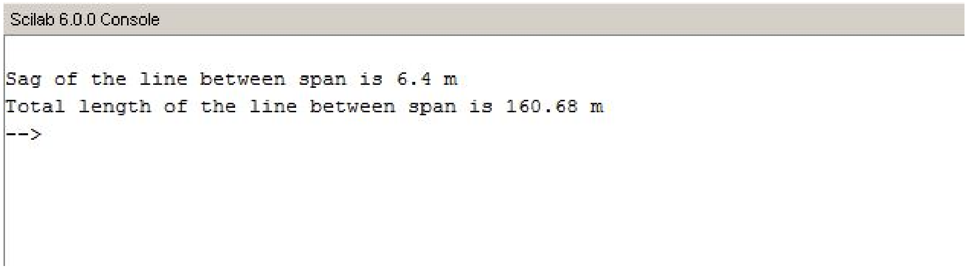
```


printf("\nSafe operating maximum line voltage is %.2f kV",mlv);
printf("\nThe string efficiency is %.2f percentage",str_eff);
```

Chapter 13

Design of Transmission Lines

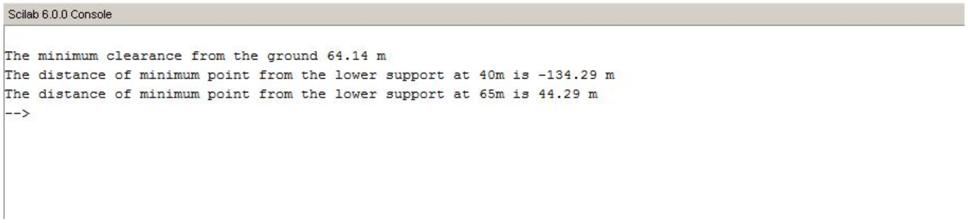
Scilab code Exa 13.1 `Calculation of Sag and Tension`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 13.1
//Scilab Version : 6.0.0     ; OS : Windows

clc;
```



```
Sag of the line between span is 6.4 m
Total length of the line between span is 160.68 m
-->
```

Figure 13.1: Calculation of Sag and Tension



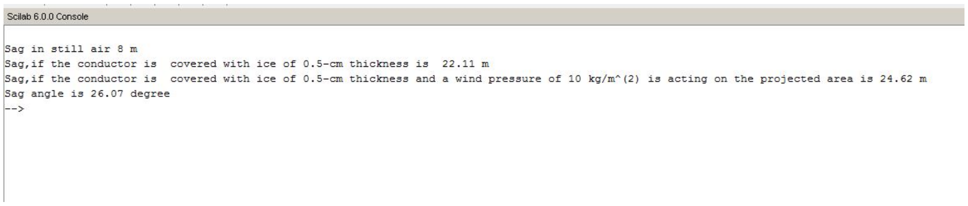
```
The minimum clearance from the ground 64.14 m
The distance of minimum point from the lower support at 40m is -134.29 m
The distance of minimum point from the lower support at 65m is 44.29 m
-->
```

Figure 13.2: Calculation of Sag and Tension

```
clear;


L=160;                                  //Span length in m
w=4;                                    //Weight of the conductor in N/m
Ts=8000;                                //Tensile strength in N
T=Ts/w;                                 //Working stress in N
d=w*L^(2)/(8*T);                        //Sag of the line in m
l=L+(w^(2)*L^(3)/(24*T^(2)));           //Total length of conductor in spans in m


printf(”\nSag of the line between span is %.1f m”,d);
printf(”\nTotal length of the line between span is %.2f m”,l);
```

Scilab code Exa 13.2 `Calculation of Sag and Tension`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 13.2
//Scilab Version : 6.0.0     ; OS : Windows

clc;
clear;


w=0.35;                                             //Weight of the conductor in kg/m
Ts=800;                                             //Tensile strength in kg
Sf=2;                                               //Safety factor
L=160;                                              //Span length in m
h=70;                                               //Height of support from the ground in m
T=Ts/Sf;                                            //Working stress in kg
h1=h-40;;                                           //Difference between supports in m
x1=(L/2)-(T*h1/(w*L));                              //Distance of minimum point from the lower support in m
h2=h-65;                                            //Difference between supports in m
x2=(L/2)-(T*h2/(w*L));                              //Distance of minimum point from the lower support in m
d1=w*x2^(2)/(2*T);                                  //Sag from lower support in m
mgc=65-d1;                                          //Minimum ground clearance in m


printf("\nThe minimum clearance from the ground %.2f
```



```
Sag in still air 8 m
Sag,if the conductor is  covered with ice of 0.5-cm thickness is  22.11 m
Sag,if the conductor is  covered with ice of 0.5-cm thickness and a wind pressure of 10 kg/m^(2) is acting on the projected area is 24.62 m
Sag angle is 26.07 degree
-->
```

Figure 13.3: Calculation of Sag and Tension

```
    m",mgc);
printf("\nThe distance of minimum point from the lower support at 40m is %.2f m",x1);
printf("\nThe distance of minimum point from the lower support at 65m is %.2f m",x2);
```

Scilab code Exa 13.3 `Calculation of Sag and Tension`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 13.3
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


A=120;                                    //Normal copper area in mm^2
con_size=(30+7)/6.30;                     //Conductor size in mm
w=0.4;                                    //Conductor weight in kg/m
```

```
Ts=1250;                                              //Tensile strength in kg
Sf=5;                                                 //Safety factor
L=200;                                                //Span length in m
t=0.5;                                                //Thickness of ice in cm
p=10;                                                 //Wind pressure in kg/m^2
D=(2*4-1)*6.30*10^(-1);                               //Total diameter of conductor in cm
T=Ts/Sf;                                              //Working stress in kg
d=w*L^(2)/(8*T);                                      //Sag in still air in m
wi=%pi*((D+t)*10^(-2)*t*10^(-2))*915;                 //Weight of ice in kg/m
W=w+wi;                                               //Total weight of ice in kg/m
d1=W*L^(2)/(8*T);                                      //Sag in m
Ww=(D+2*t)*10^(-2)*p;                                 //Wind loading in kg/m
We=sqrt(Ww^(2)+(w+wi)^(2))                            //Effective loading in kg/m
d2=We*L^(2)/(8*T);                                    //Total Sag in m
angle=atand(Ww/(w+wi));                               //Sag angle in degree

printf("\nSag in still air %.f m",d);
printf("\nSag,if the conductor is  covered with ice of 0.5-cm thickness is % .2f m",d1);
printf("\nSag,if the conductor is  covered with ice of 0.5-cm thickness and a wind pressure of 10 kg/m^(2) is acting on the projected area is %.2f m",d2);
```

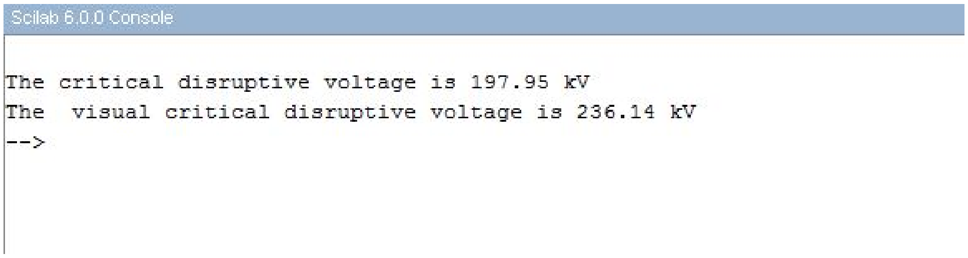
```
printf("\nSag angle is %.2f degree",angle);
```

Chapter 14

Corona and Radio Interference

Scilab code Exa 14.1 `Compute Transmission line parameters`

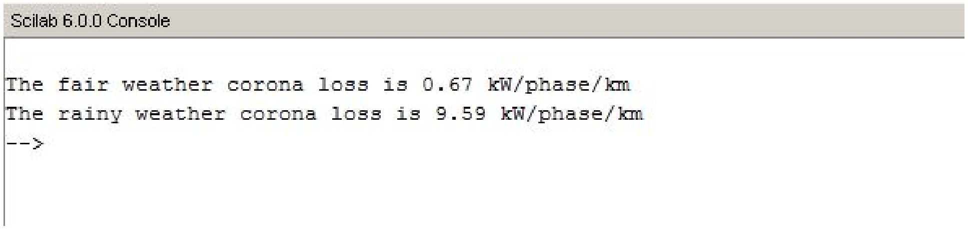
```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 14.1
//Scilab Version : 6.0.0     ; OS : Windows

clc;
clear;
```



```
The critical disruptive voltage is 197.95 kV
The  visual critical disruptive voltage is 236.14 kV
-->
```

Figure 14.1: Compute Transmission line parameters

```



d=600;                    //Distance between three conductors in cm
b=72;                     //Pressure in Hg
r=1;                      //Radius of the conductor in cm
t=27;                     //Air temperature in Celcius
m=0.90;                   //Irregularity factor
mv=0.82;                  //Surface factor
adf=3.92*b/(273+t);                                         //Air density factor
Vc=r*21.1*m*0.9408*log(d/r);                                //Phase to neutral critical disruptive voltage in kV
cdv=Vc*sqrt(3);                                             //Line to line critical disruptive voltage in kV
Vv=21.1*mv*r*0.9408*(1+(0.3/sqrt(r*0.9408)))*log(d/r);       //Critical visual disruptive voltage
cvdv=Vv*sqrt(3);                                            //Line to line critical visual disruptive voltage in kV


printf("\nThe critical disruptive voltage is %.2f kV
```



Scilab 6.0.0 Console

```
The fair weather corona loss is 0.67 kW/phase/km
The rainy weather corona loss is 9.59 kW/phase/km
-->
```

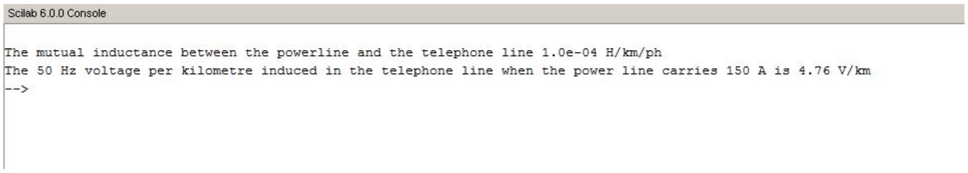
Figure 14.2: Compute Transmission line parameters

```
    ",cdv);
printf("\nThe visual critical disruptive voltage is
    %.2f kV",cvdv);
```

Scilab code Exa 14.2 `Compute Transmission line parameters`

```
//Electric Power Generation, Transmission and
    Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 14.2
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


V=220;

    //Supply voltage in kV
f=50;

    //Supply frequency in Hz
r=1;
```

```
    //Radius of the conductor in cm
d=400;
    //Distance between the conductor in cm
m=0.96;
    //Smooth surface value of conductor
b=73;
    //Barometric pressure in cm of Hg
t=20;
    //Temperature in celcius
adf=3.92*b/(273+t);
    //Air density factor
Vc=r*21.1*m*0.9767*log(d/r);
                                                    //Phase to neutral critical disruptive voltage in kV
Vp=V/sqrt(3);
    //Line phase voltage in kV
Pc=241*10^(-5)*(f+25)/0.9767*sqrt(r/d)*(Vp-Vc)^(2);
                        //Peek's formula for corona loss in a fair weather in kW/phase/km
Pc1=241*10^(-5)*(f+25)/0.9767*sqrt(r/d)*(Vp-0.8*Vc)^(2);                    //Peek's formula for corona loss in a rainy weather in kW/phase/km


printf("\nThe fair weather corona loss is %.2f kW/phase/km",Pc);
printf("\nThe rainy weather corona loss is %.2f kW/phase/km",Pc1);
```



```
The mutual inductance between the powerline and the telephone line 1.0e-04 H/km/ph
The 50 Hz voltage per kilometre induced in the telephone line when the power line carries 150 A is 4.76 V/km
-->
```

Figure 14.3: Compute Power line and Telephone line parameters

Scilab code Exa 14.4 `Compute Power line and Telephone line parameters`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 14.4
//Scilab Version : 6.0.0     ; OS : Windows

clc;
clear;


f=50;                                                    //Supply frequency in Hz
I=150;                                                   //Power line current in Amps
dac=1.8;                                                 //Spacing between conductors a and d in m
dab=2.5;                                                 //Spacing between conductors a and b in m
```

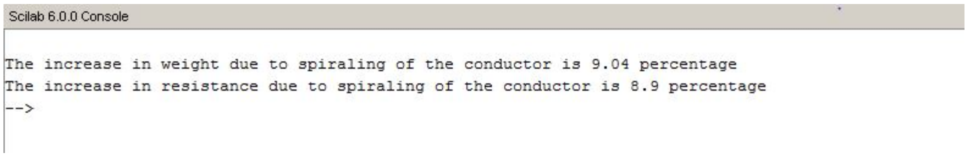
```
dcd=1;                                                  //Spacing between conductors c and d in m
Dad=sqrt((dac)^(2)+((dab/2)+(dcd/2))^(2));              //Distance between conductors a and d in m
Dac=sqrt((dac)^(2)+((dab/2)-(dcd/2))^(2));              //Distance between conductors a and c in m
M=4*10^(-4)*log(Dad/Dac);                               //Mutual inductance in H/km/ph
X=2*%pi*f*M;                                            //Inductive reactance in per km
emf=I*X;                                                //Emf induced in telephone line in V/km

printf("\nThe mutual inductance between the powerline and the telephone line %.1e H/km/ph",M);
printf("\nThe 50 Hz voltage per kilometre induced in the telephone line when the power line carries 150 A is %.2f V/km",emf);
```

Chapter 15

Insulated Cables

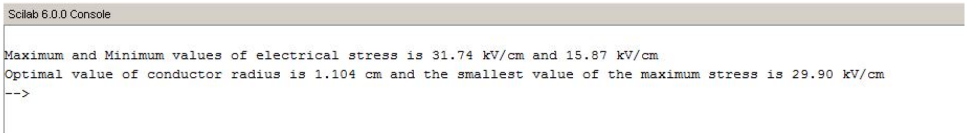
Scilab code Exa 15.1 Compute increased in resistance and weight

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 15.1
//Scilab Version : 6.0.0     ; OS : Windows

clc;
clear;


```



```
The increase in weight due to spiraling of the conductor is 9.04 percentage
The increase in resistance due to spiraling of the conductor is 8.9 percentage
-->
```

Figure 15.1: Compute increased in resistance and weight



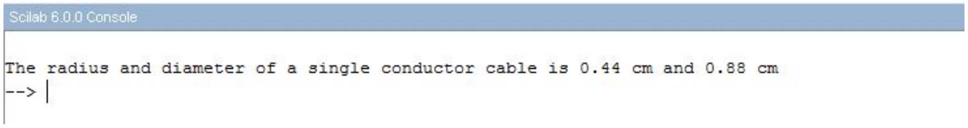
```
Maximum and Minimum values of electrical stress is 31.74 kV/cm and 15.87 kV/cm
Optimal value of conductor radius is 1.104 cm and the smallest value of the maximum stress is 29.90 kV/cm
-->
```

Figure 15.2: Compute Electric stress values

```
D=2;                                          //Conductor diameter in cm
l=40;                                         //Length of lay in cm
n=1;                                          //Strand of layer one
l1=sqrt(l^(2)+(%pi*(2*n+1)*D)^(2));           //Length is a strand of layer one in cm
Tl1=l+6*l1;                                   //Total length of strands in cm
Tl2=7*l;                                      //Total length of strands,Not spiraled in cm
W=((Tl1-Tl2)/Tl2)*100;                        //Weight increased in percentage
R1=1/l+(6/l1);
R2=7/l;
R=(R2/R1)*100;                                //Change in resistance in percentage
R1=R-100;                                     //Increased resistance in percentage


printf("\nThe increase in weight due to spiraling of the conductor is %.2f percentage",W);
printf("\nThe increase in resistance due to spiraling of the conductor is %.1f percentage",R1);
```

Scilab code Exa 15.2 Compute Electric stress values

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 15.2
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


r=1.5;                                          //Conductor radius in cm
R=3;                                            //Lead sheath radius in cm
V=33;                                           //Operating voltage in kV
E_max=V/(r*log(R/r));                           //Maximum value of electric stress in kV/cm
E_min=V/(R*log(R/r));                           //Minimum value of electric stress in kV/cm
r1=R/2.718;                                     //Optimum value of conductor radius in cm
E_max1=V/(r1*log(R/r1));                        //Smallest value of Maximum stress in kV/cm

printf("\nMaximum and Minimum values of electrical stress is %.2f kV/cm and %.2f kV/cm",E_max,E_min);
printf("\nOptimal value of conductor radius is %.3f cm and the smallest value of the maximum stress is %.2f kV/cm",r1,E_max1);
```

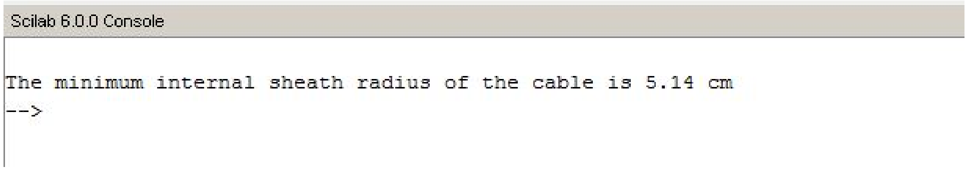


```
The radius and diameter of a single conductor cable is 0.44 cm and 0.88 cm
-->
```

Figure 15.3: Compute the radius and diameter

Scilab code Exa 15.3 `Compute the radius and diameter`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 15.3
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


V=11;                                   //Supply voltage in kV
die_strength=50;                        //Dielectric strength of conductor in kV/cm
Sf=2;                                   //Safety factor
e=2.718;                                //Constant value
E_max=die_strength/Sf;                  //Maximum stress in kV/cm
R=11*e/25;                              //Outer insulation radius in cm
r=R/e;                                  //Radius of the conductor in cm
D=2*r;                                  //Diameter of
```



```
The minimum internal sheath radius of the cable is 5.14 cm
-->
```

Figure 15.4: Compute radius

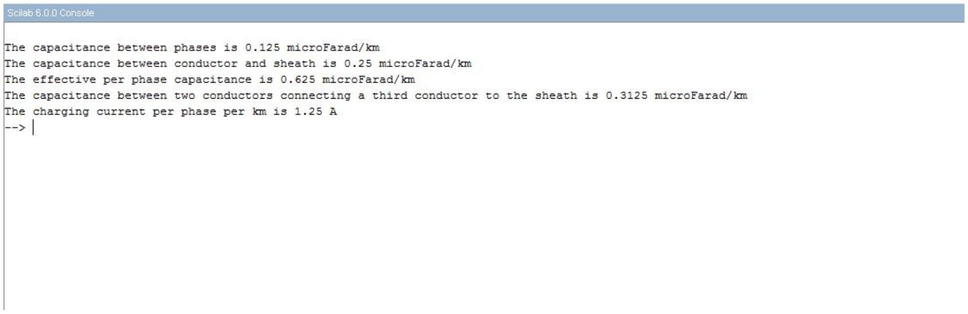
```
    the conductor in cm

printf("\nThe radius and diameter of a single
    conductor cable is %.2f cm and %.2f cm",r,D);
```

Scilab code Exa 15.4 `Compute radius`

```
//Electric Power Generation, Transmission and
    Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 15.4
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


V=110;
    //Line voltage in kV
r=1;
    //Conductor radius in cm
p1=5;
```

```
    //Permittivitie of the material A
p2=4;
    //Permittivitie of the material B
p3=2;
    //Permittivitie of the material C
G1=50;
    //Permissible stress of the material A in kV/cm
G2=40;
    //Permissible stress of the material B in kV/cm
G3=30;
    //Permissible stress of the material C in kV/cm
r1=p1*r*G1/(p2*G2);                                        //
    Outer radius of the material A in cm
r2=p2*r1*G2/(p3*G3);                                       //
    Outer radius of the material B in cm
R=exp(1.638);
    //Outer radius of the material C in cm(solving
    the eqn 15.24 in the book )

printf("\nThe minimum internal sheath radius of the
    cable is %.2f cm",R)
```

Scilab code Exa 15.6 `Compute capacitance`



```
The capacitance between phases is 0.125 microFarad/km
The capacitance between conductor and sheath is 0.25 microFarad/km
The effective per phase capacitance is 0.625 microFarad/km
The capacitance between two conductors connecting a third conductor to the sheath is 0.3125 microFarad/km
The charging current per phase per km is 1.25 A
-->
```

Figure 15.5: Compute capacitance

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 15.6
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


V=11;       //Supply voltage in kV
f=50;       //Supply frequency in Hz
C=0.5;      //Capacitance between two conductors in microFarad/km
Cx=0.75;    //Capacitance between sheath and three conductors in microFarad/km
Cy=0.50;    //Capacitance between sheath and remaining conductor in microFarad/km
C1=Cx/3;    //Capacitance between conductor and sheath in
```

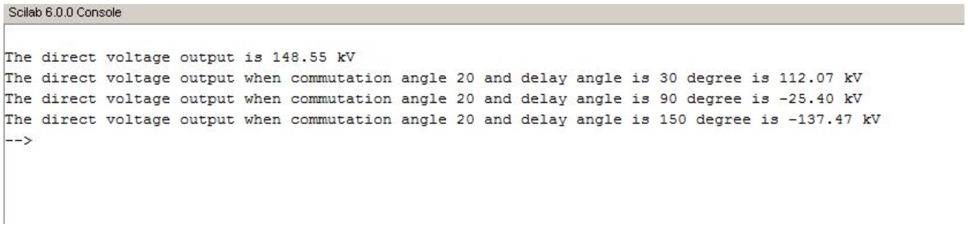
```
microFarad/km
C2=(Cy-C1)/2;    //Capacitance between phases in microFarad/km
C0=C1+3*C2;    //Effective capacitance in microFarad/km
C3=C0/2;    //Capacitance between two conductors connecting a third conductor to the sheath in microFarad/km
I=(V*10^(3)/sqrt(3))*2*%pi*f*C0*10^(-6);    //Charging current in A/ph/km


printf("\nThe capacitance between phases is %.3f microFarad/km",C2);
printf("\nThe capacitance between conductor and sheath is %.2f microFarad/km",C1);
printf("\nThe effective per phase capacitance is %.3f microFarad/km",C0);
printf("\nThe capacitance between two conductors connecting a third conductor to the sheath is %.4f microFarad/km",C3);
printf("\nThe charging current per phase per km is %.2f A",I);
```

Chapter 16

HVDC Transmission and FACTS Technology

Scilab code Exa 16.1 `Compute dc output voltages`

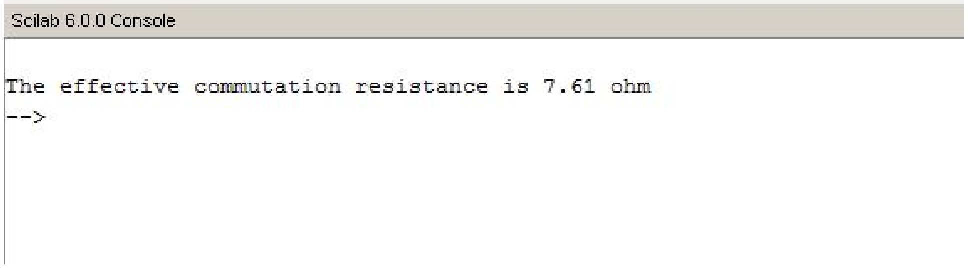
```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 16.1
//Scilab Version : 6.0.0      ; OS : Windows

clc;
```



```
The direct voltage output is 148.55 kV
The direct voltage output when commutation angle 20 and delay angle is 30 degree is 112.07 kV
The direct voltage output when commutation angle 20 and delay angle is 90 degree is -25.40 kV
The direct voltage output when commutation angle 20 and delay angle is 150 degree is -137.47 kV
-->
```

Figure 16.1: Compute dc output voltages

```
clear;


V=238;                  //Transformer primary voltage in kV
Em=110;                 //Transformer secondary voltage in kV
f=50;                   //Supply frequency in Hz
u=20;                   //Commutation angle in degree
alpha1=30;              //Delay angle 1 in degree
alpha2=90;              //Delay angle 2 in degree
alpha3=150;             //Delay angle 3 in degree
Vdo=3*sqrt(3*2)*Em/(%pi*sqrt(3));       //Direct output voltage in kV
Vd1=Vdo/2*(cosd(alpha1)+cosd(alpha1+u));        //Direct output voltage when commutation angle 20 and delay angle is 30 degree in kV
Vd2=Vdo/2*(cosd(alpha2)+cosd(alpha2+u));        //Direct output voltage when commutation angle 20 and delay angle is 90 degree in kV
Vd3=Vdo/2*(cosd(alpha3)+cosd(alpha3+u));        //Direct output voltage when commutation angle 20 and delay angle is 150 degree in kV

printf("\nThe direct voltage output is %.2f kV",Vdo);
printf("\nThe direct voltage output when commutation angle 20 and delay angle is 30 degree is %.2f kV",Vd1);
printf("\nThe direct voltage output when commutation angle 20 and delay angle is 90 degree is %.2f kV",Vd2);
printf("\nThe direct voltage output when commutation
```



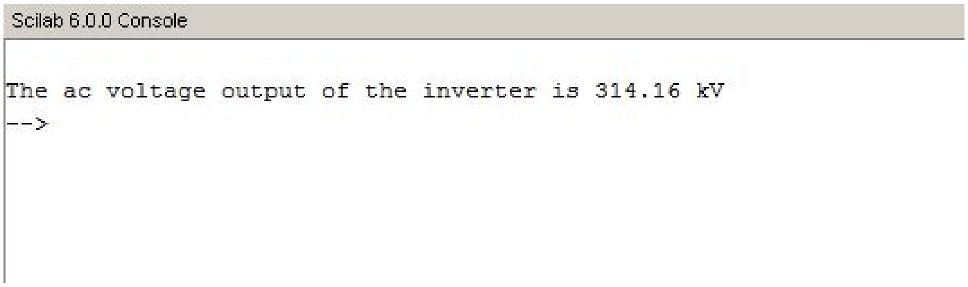
```
The effective commutation resistance is 7.61 ohm
-->
```

Figure 16.2: Compute resistance

```
angle 20 and delay angle is 150 degree is %.2f kV",Vd3);
```

Scilab code Exa 16.2 `Compute resistance`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 16.2
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


Em=400;                                         //Ac supply voltage in kV
Vd=500;                                         //Dc supply voltage in kV
Id=1;                                           //Dc current in A
alpha=20;                                       //Firing
```



```
The ac voltage output of the inverter is 314.16 kV
-->
```

Figure 16.3: Compute Ac output voltages

```
   angle in degree
Vdo=3*sqrt(3*2)*Em/(%pi*sqrt(3));          //Direct output voltage in kV
Rc=-(Vd-(Vdo*cosd(alpha))/Id);             //Effective Commutation resistance in Ohm


printf("\nThe effective commutation resistance is %.2f Ohm",Rc);
```

Scilab code Exa 16.3 Compute Ac output voltages

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 16.3
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


```

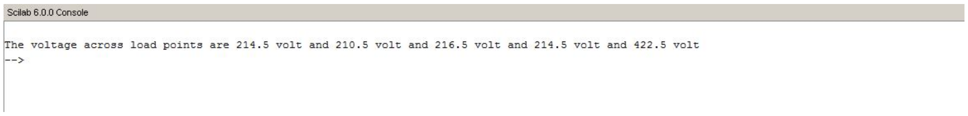
```
V=500;                                 //Dc supply voltage in kV
ang1=20;                               //Advance angle in degree
ang2=10;                               //Extinction angle in degree
Vdi=1/2*(cosd(20)+cosd(10));           //Dc voltage in kV
Em=(V*%pi)/(Vdi*3*sqrt(3));            //Ac output voltage in kV


printf("\nThe ac voltage output of the inverter is %.2f kV",Em);
```

Chapter 17

Distribution Systems

Scilab code Exa 17.2 `Compute current`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 17.2
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


V=110;                                              //Supply voltage in kV
P1=30;                                              //Load for 5 hours in MW
P2=15;                                              //Load for 10 hours in MW
P3=8;                                               //Load for 9 hours in MW
pf1=0.8;                                            //Lagging power factor of 30 MW load
```



```
The voltage across load points are 214.5 volt and 210.5 volt and 216.5 volt and 214.5 volt and 422.5 volt
-->
```

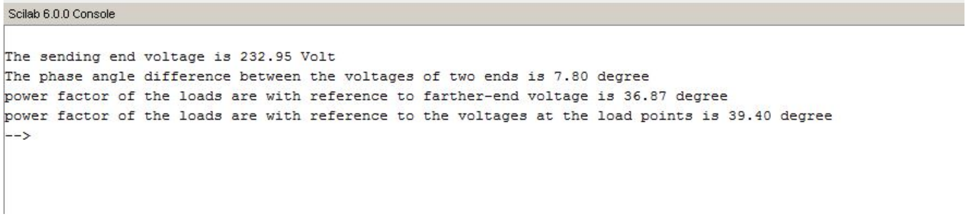
Figure 17.1: Compute voltage across load points

```
pf2=0.9;                                                  //Lagging power factor of 15 MW load
pf3=1;                                                    //Unity power factor of 8 MW load
I1=P1*10^(6)/(sqrt(3)*V*10^(3)*pf1);                      //Current of 30 MW load in Amps
I2=P2*10^(6)/(sqrt(3)*V*10^(3)*pf2);                      //Current of 15 MW load in Amps
I3=P3*10^(6)/(sqrt(3)*V*10^(3)*pf3);                      //Current of 8 MW load in Amps

//The remaining of the problem cannot be solved using SCILAB
```

Scilab code Exa 17.7 `Compute voltage across load points`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 17.7
//Scilab Version : 6.0.0     ; OS : Windows

clc;
clear;


```

```
Vs=220;                                       //Supply voltage in Volt
r=1;                                          //Resistance in Ohm/km
D_aC=5;                                       //Voltage drop in section aC in Volt
D_C1E1=1;                                     //Voltage drop in section C1E1 in Volt
D_PE1=0.5;                                    //Voltage drop in section pE1 in Volt
D_CB=3;                                       //Voltage drop in section CB in Volt
D_B1D1=1;                                     //Voltage drop in section B1D1 in Volt
D_D1C1=0;                                     //Voltage drop in section D1C1 in Volt
D_Eb=3;                                       //Voltage drop in section Eb in Volt
D_BA=1.5;                                     //Voltage drop in section BA in Volt
D_A1D=2;                                      //Voltage drop in section A1D in Volt
D_DE=3;                                       //Voltage drop in section DE in Volt
CC1=Vs-D_aC-D_C1E1+D_PE1;                     //Voltage across section CC1 in Volt
BB1=CC1-D_CB-D_B1D1-D_D1C1;                   //Voltage across section BB1 in Volt
E1E=Vs-D_PE1-D_Eb;                            //Voltage across section E1E in Volt
D1D=E1E+D_C1E1+D_D1C1-D_DE;                   //Voltage across section D1D in Volt
AA1=2*Vs-D_aC-D_CB-D_BA-D_A1D-D_DE-D_Eb;      //Voltage across section AA1 in Volt

printf("\nThe voltage across load points are %.1f volt and %.1f volt and %.1f volt and %.1f volt and %.1f volt",CC1,BB1,E1E,D1D,AA1);
```



```
The sending end voltage is 232.95 Volt
The phase angle difference between the voltages of two ends is 7.80 degree
power factor of the loads are with reference to farther-end voltage is 36.87 degree
power factor of the loads are with reference to the voltages at the load points is 39.40 degree
-->
```

Figure 17.2: Compute power factor with respective load points

---

Scilab code Exa 17.9 `Compute power factor with respective load points`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 17.9
//Scilab Version : 6.0.0      ; OS : Windows

clc;
clear;


V=220;                                                  //Supply voltage in Volt
R=0.06;                                                 //Resistance in Ohm
X=0.1;                                                  //Reactance in Ohm/km
L=1;                                                    //Length of section AB and BC in km
IB=100*(0.8+%i*0.6);                                    //Current at point B in Amps
```

```
IC=100*(1+%i*0);                                              //
   Current at point C in Amps
ZBC=(R+%i*X);                                                 //
   Impedance of section BC in Ohm
ZAB=(R+%i*X);                                                 //
   Impedance of section AB in Ohm
BC=IC*ZBC;                                                    //
   Drop in section BC in Volt
VB=V+BC;                                                      //
   Potential at point B in Volt
I_AB=IB+IC;                                                   //
   Current in section AB in Amps
V_AB=(IB+IC)*ZAB;                                             //
   Voltage drop in section AB Volt
VA=VB+V_AB;                                                   //
   Voltage at point A in Volt
VB1=V+BC;                                                     //
   Potential at point B in Volt
pfa=acosd(0.8);                                               //
   Power factor angle of the load at point B
ref_ang=-pfa-atand(imag(VB1),real(VB1));                      //
   Leading Power factor angle with reference to Vc
IB1=100*(cosd(ref_ang)+%i*sind(ref_ang));                     //
   Current at point B in Amps
I_AB1=IB+IC;                                                  //
   Current in section AB in Amps
V_AB1=(IB+IC)*ZAB;                                            //
   Voltage drop in section AB Volt
VA1=VB1+V_AB1;                                                //
   Voltage at point A in Volt


printf("\nThe sending end voltage is %.2f Volt",abs(
   VA));
printf("\nThe phase angle difference between the
   voltages of two ends is %.2f degree",atand(imag(
   VA),real(VA)));
printf("\npower factor of the loads are with
```

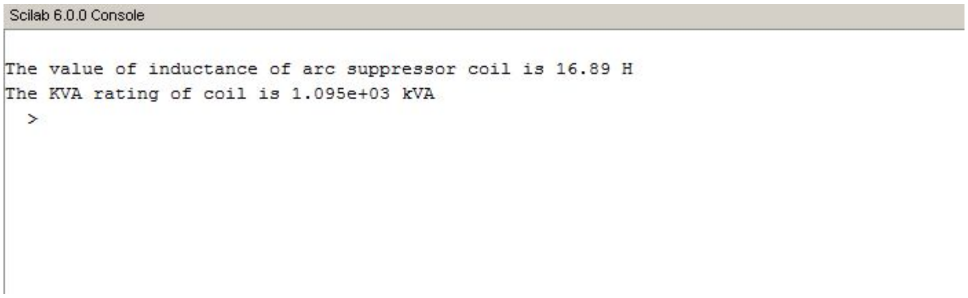
```
reference to farther-end voltage is %.2f degree",pfa);
printf("\npower factor of the loads are with reference to the voltages at the load points is %.2f degree",-(ref_ang));
```

Chapter 19

Grounding Systems

Scilab code Exa 19.1 `Calculate KVA rating`

```
//Electric Power Generation, Transmission and Distribution by S.N.Singh
//Publisher:PHI Learning Private Limited
//Year: 2012 ; Edition - 2
//Example 19.1
//Scilab Version : 6.0.0      ; OS : Windows

clc;
```



```
The value of inductance of arc suppressor coil is 16.89 H
The KVA rating of coil is 1.095e+03 kVA
  >
```

Figure 19.1: Calculate KVA rating

```
clear;


V=132;                                                      //Operating Voltage in kV
C=0.2;                                                      //Line to Ground Capacitance in microFarad
f=50;                                                       //Supply Frequency in Hz
L=1/(3*(2*%pi*f)^(2)*C*10^(-6));                            //Inductance of Coil in H
VA_coil=(132e3/1.732)^(2)/(2*%pi*f*L);                      //Rating of Coil in VA
KVA_coil=VA_coil/1e3;                                       //To convert VA value into kVA value

printf("\nThe value of inductance of arc suppressor coil is %.2f H",L);
printf("\nThe KVA rating of coil is %.3e kVA",KVA_coil);
```