Scilab Textbook Companion for Non-conventional Energy Sources by G. D. Rai[1]

Created by
Mohammed Touqeer Mohammed Toufique Khan
B.E
Electronics Engineering
Anjuman i islam Kalsekar Technical Campus
College Teacher
None
Cross-Checked by
Chaya Ravindra

July 17, 2017

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

Book Description

Title: Non-conventional Energy Sources

Author: G. D. Rai

Publisher: Khanna Publishers

Edition: 1

Year: 2010

ISBN: 8174090738

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 2

Solar radiation and its Measurement

Scilab code Exa 2.4.1 Local solar time and declination

```
//Ex2.4.1.;Detremine local solar time and
    declination

//The local solar time=IST-4(standard time longitude
    -longitude of location)+Equation of time
    correstion
//IST=12h 30min;for the purpose of calculation we
    are writing it as a=12h,b=29 min 60sec;
a=12;
b=29.60;
//(standard time longitude-longitude of location)=82
     degree 30min - 77 degree 30min;
//for the purpose of calculation we are writing it
    as
STL3=82.5-72.5;
//Equation of time correstion: 1 min 01 sec
//for the purpose of calculation we are writing it
    as
c=1.01;
```

```
//The local solar time=IST-4(standard time longitude -longitude of location)+Equation of time correstion
LST=b-STL3-c;
printf(" The local solar time=%f.%f in hr.min.sec",a,LST);
//Declination delta can be obtain by cooper's eqn : delta=23.45*sin((360/365)*(284+n))
n=170;//(on June 19)
//let
a=(360/365)*(284+n);aa=(a*%pi)/180;
//therefore
delta=23.45*sin(aa);
printf("\n delta=%f degree",delta);
```

Scilab code Exa 2.4.2 Angle made by beam radiation

```
//Ex2.4.2.;Calculate anglr made by beam radiation with the normal to a flat collector.
gama=0;//since collector is pointing due south.
//For this case we have equation : cos_(theta_t)=cos(fie-s)*cos(delta)*cos(w)+sin(fie-s)*sin(delta)
//with the help of cooper eqn on december 1,
n=335;
//let
a=(360/365)*(284+n);aa=(a*%pi)/180;
//therefore
delta=23.45*sin(aa);
printf(" delta=%f degree",delta);
//Hour angle w corresponding to 9.00 hour=45 Degreew
w=45;//degree
//let
a=cos(((28.58*%pi)/180)-((38.58*%pi)/180))*cos(delta*%pi*180^-1)*cos(w*%pi*180^-1);
b=sin(delta*%pi*180^-1)*sin(((28.58*%pi)/180)
```

```
    -((38.58*%pi)/180));
//therefore
cos_of_theta_t=a+b;
theta_t=acosd(cos_of_theta_t);
printf("\n theta_t=%f Degree",theta_t);
```

Scilab code Exa 2.7.1 Average value of Solar radiation on a horizontal surface

```
//Ex2.7.1.;Determine the average values of radiation
    on a horizontal surface

//Declination delta for June 22=23.5 degree, sunrice
    hour angle ws
delta=(23.5*%pi)/180;//unit=radians
fie=(10*%pi)/180;;//unit=radians
//Sunrice hour angle ws=acosd(-tan(fie)*tan(delta))
ws=acosd(-tan(fie)*tan(delta));
printf(" Sunrice hour angle ws=%f Degree",ws);
n=172;//=days of the year (for June 22)
//We have the relation for Average insolation at the
    top of the atmosphere
//Ho=(24/%pi)*Isc*[{1+0.033*(360*n/365)}*((cos (fie)
   *cos(delta)*sin(ws))+(2*%pi*ws/360)*sin(fie)*sin(
   delta))]
Isc=1353;//SI unit=W/m^2
ISC=1165;//MKS unit=kcal/hr m^2
//let
a=24/%pi;
aa=(360*172)/365;aaa=(aa*%pi)/180;
b=cos(aaa);bb=0.033*b;bbb=1+bb;
c=(10*%pi)/180;c1=cos(c);
cc=(23.5*%pi)/180;cc1=cos(cc);
ccc=(94.39*%pi)/180;ccc1=sin(ccc);
c=c1*cc1*ccc1;
d=(2*%pi*ws)/360;
```

```
e=(10*%pi)/180;e1=sin(e);
ee=(23.5*%pi)/180;ee1=sin(ee);
e=e1*ee1;
//therefoe Ho in SI unit
Ho=a*Isc*(bbb*(c+(d*e)));
printf("\n SI UNIT->Ho=%f W/m^2",Ho);
Hac=Ho*(0.3+(0.51*0.55))
printf("\n SI UNIT->Hac=%f W/m^2 day",Hac);
ho=a*ISC*(bbb*(c+(d*e)));
printf("\n MKS UNIT->Ho=%f kcal/m^2",ho);
hac=ho*0.58;
printf("\n MKS UNIT->Hac=%f kcal/m^2 day",hac);

//The values are approximately same as in textbook
```

Chapter 3

Solar Energy Collectors

Scilab code Exa 3.6.1 `Solar altitude angle and Incident angle and Collector effici`

```
//Ex3.6.1.;calculate: solar altitude anglr,Incident angle,Collector efficiency

//Solar declination :delta
n=1
delta=23.45*sin((360/365)*(284+n));
printf(" Solar declination delta=%f degree",delta);
fie=22;//degree
//solar hour angle ws=0,(at mean of 11:30 and 12:30)
ws=0;
//Solar altitude anglr alpha is given by

//alpha=asind(((cos(fie)*cos(delta)*cos(ws))+(sin(fie)*sin(delta)))
//let
a=cos((22*%pi)/180)*cos((-23*%pi)/180)*cos(0);
b=sin((22*%pi)/180)*sin((-23*%pi)/180);
//therefore
sin_alpha=a+b;
printf("\n sin_aplha=%f",sin_alpha);
alpha=asind(sin_alpha);
```

```
printf("\n aplha=%f Degree",alpha);
//Incident angle
theta=(180/2)-alpha;
printf("\n Incident angle=%f Degree",theta);
//Rb is given by
Rb=((cos(((22*%pi)/180)-(37*%pi)/180)*cos((-23*%pi)/180)*cos(0))+(sin(((22*%pi)/180)-(37*%pi)/180)*sin((-23*%pi)/180)))/sin_alpha;
printf("\n Rb=%f",Rb);
//Effective absorptance product is <t.alpha>=t.alpha/ 1-(1-alpha)*pd
pd=0.24;//Diffuse reflectance for two glass covers
//let TA=<t.alpha>
TA=(0.88*0.90)/(1-(1-0.90)*pd);
printf("\n Effective absorptance product is <t.alpha>=%f",TA);
//Solar radiation intensity(consider beam radiation only)
//Hb=0.5 ly/mm = 0.5 cal/cm^2 * min
Hb=((0.5*10^4)/10^3)*60;//unit=kcal/m^2 hr
printf("\n Hb=%f kcal/m^2 hr",Hb);
Hb=Hb*1.163;//unit=W/m^2 hr;   [since  1 kcal = 1.163 watt]
printf("\n Hb=%f W/m^2 hr",Hb);
//S=Hb*Rb*<t.alpha>
S=Hb*Rb*TA;
printf("\n S=%f W/m^2 hr",S);
s=S/1.163;
printf("\n S=%f kcal/m^2 hr",s);
//Useful gain
//qu=FR(S-UL*(Tfi-Ta))
qu=0.810*(s-(6.80*(60-15)))
printf("\n qu=%f kcal/m^2 hr",qu);
//Qu=FR(S-UL*(Tfi-Ta))
Qu=0.810*(S-(7.88*(60-15)))
printf("\n qu=%f W/m^2 hr",Qu);
//Collection Efficiency  : nc=(qu/(Hb*Rb))*100;
nc=(28.07/(300*Rb))*100;
```

```
printf("\n Collection Efficiency=%f persent",nc);


//values of "sine alpha" in the textbook is taken
   approximate to the real values
```

Scilab code Exa 3.9.1 Useful gain and exit fluid temperature and collection effici

```
//calculate the useful gain,exit fluid temperature
   and collection efficiency
//Optical properties are estimated as
p=0.85;
//(T. alpha)=0.77;let A=(T. alpha)
A=0.77
gama=0.94;
Do=0.06;
L=8;//unit=meter,//L=length of concentrator
W=2;//W=width of concentrator in meter
dco=0.09;//dco=diameter of transpaarent cover
Ar= %pi*Do*L;//Ar=area of the receiver pipe
A_alpha=(W-dco)*L;//aperture area of the
   concentration
Cp=0.30;//unit=kcal/kg degree calcius
m=400;//unit=kg/hr,m=flow rate
HbRb=600;//unit=kcal/hr m^2
Tfi=150;//degree calcius
T_alpha=25;//degree calcius
//Heat transfer coefficient from fluid inside to
   surroundings,
Uo=5.2;//unit=kcal/hr-m^2
//Heat transfer coefficient from absorber cover
   surface to surroundings,
UL=6;//unit=kcal/hr-m^2
F=(Uo/UL);
//Heat removed factor FR is
```

```
//FR=((m*Cp)/(Ar*UL))*(1-(%e^-((Ar*UL*F)/(m*Cp))))
//let X=(m*Cp)/(Ar*UL);Y=(%e^-((Ar*UL*F)/(m*Cp)))
X=(m*Cp)/(1.51*UL*0.86);
Y=%e^(-1/X);
FR=X*0.86*(1-Y);
//Absorbed solar energy is
S=HbRb*p*gama*A;
printf(" Area of the receiver pipe Ar= %f=1.51 m^2 \n A_aplha= %f m^2=collection efficiency factor ",Ar,A_alpha);
printf("\n value of F= %f",F);
printf("\n Heat removed factor FR=%f \n Absorbed solar energy is \n S=%f kcal/Hr m^2 .....(MKS) ",FR,S);
//for unit in S.I.   , 1 kcal/Hr m^2 = 1.16298 W/m^2
s= S*1.16298; //in W/m^2
printf("\n S=%f W/m^2.....(SI)",s);
//the values of F,FR will be same in any unit,since they are factors(dimensionless)
//Useful Gain=Qu=A_alpha*FR*(S-((Ar*UL)/A_alpha)*(Tfi-T_alpha))
//In MKS unit
Qu=A_alpha*FR*(S-((1.51*UL)/A_alpha)*(Tfi-T_alpha))
printf("\n useful gain in (MKS) Qu=%f kcal/hr",Qu);
//IN SI unit
qu=A_alpha*FR*(s-((1.51*6.98)/A_alpha)*(Tfi-T_alpha))//UL=6.98 W/m^2 degree celcius
printf("\n useful gain in (SI) Qu=%f Watt",qu);
//the exit fluid temperature can be obtained from
tci=150;//degree celcius
tco=tci+(Qu/(m*Cp));//from Qu=mCp(tco-tc);  where, tco=collector fluid temp. at outlet,tci=Fluid inlet temp.
n=(Qu/(16*HbRb))*100;//ncollector=Qu/(A_alpha*HbRb)*100;
printf("\n collector fluid temp. at outlet  tco=%f degree celcius \n ncollector = %f percent ",tco,n);
```

```
//The values/results/answers is approximate in the
    text book to the real calculated value
```

Chapter 6

Wind Energy

Scilab code Exa 6.2.1 Torque and axial thrust

```
//Ex.6.2.1.
//For air,the value of gas constant
R=0.287 //unit=kj/kg K
//T=15 in degreecalcius
T=15+273;//in kalvin
RT=0.287*10^3*288;
P=1.01325*10^5;//unit=Pa;at 1 atm
Vi=15;//unit=m/s
gc=1;
D=120;//turbine diameter;unit=m
N=40/60;
//Air density
p=(P/RT);
printf("Air density p=%f kg/M^3",p);
//1] Total_power= Ptotal=p*A*Vi^3/2*gc
//power density =Ptotal/A=p*Vi^3/2*gc
power_density=(1/(2*gc))*(p*Vi^3);
//2] Maximum_power_density=Pmax/A=8*p*Vi^3/27*gc
Maximum_power_density=(8/(27*gc))*(p*Vi^3);
printf("\n power density =Ptotal/A= %f W/m^2 \n
    Maximum power density=Pmax/A= %f W/m^2",
```

```
    power_density,Maximum_power_density);
//3]Assuming n=35%
n=0.35;
//let P/A=x
x=n*(power_density);
printf("\n P/A=%f W/m^2",x);
//4]Total power P= power density * Area
Total_power_P=724*(%pi/4)*(D^2) //Total power P=
    power density*(%pi/4)*D^2
printf("\n Total_power_P=%f watt=%f*10^-3 kW",
    Total_power_P,Total_power_P);
//5]Torgue at maximum efficiency
Tmax=(2/(27*gc))*((1.226*D*Vi*Vi*Vi)/N);//Tmax
    =(2/(27*gc))*((p*D*Vi*Vi*Vi)/N);
printf("\n Torgue at maximum efficiency=%f Newton",
    Tmax)
//and maximum axial thurst
Fxmax=(3.14/(9*gc))*1.226*D^2*Vi^2;//Fxmax=(%pi/(9*
    gc))*p*D^2*Vi^2;
printf("\n maximum axial thurst=%f Newton",Fxmax);
```

Chapter 7

Energy from Biomass

Scilab code Exa 7.15.1 `Volume of biogas digester and power available from the dige`

```
//Ex7.15.1; calculate volume of biogas digester and
   power available from the digester
//Mass of the dry input
M0=2*5;//M0=2.5 kg/day * 5
pm=50;//unit=kg/m^3
tr=20;//retention time in days
C=0.24;//unit=m^3 per kg;Biogas yeild.
n=0.6;//efficiency of burner
Hm=28;//unit=MJ/m^3//combustion of methane
Fm=0.8;//methane proportional
//Fluid volume Vf is =M0/pm
Vf=M0/pm;
printf(" Mass of the dry input M0=%f kg/day \n Fluid
    volume Vf=%f m^3 /day",M0,Vf);
//for expression Vd=Vftr,the digester volume is
Vd=Vf*tr;
printf("\n Vd=%f m^3",Vd);
//volume of biogas is Vb=C*M0= biogas yield input *
   mass of dry input
Vb=C*M0;
printf("\n volume of biogas is Vb=%f m^3 /day",Vb);
```

```
//The Power available from the digester is
E=n*Hm*Fm*Vb;
printf("\n The Power available from the digester=%f
    Mj/day",E);
E=E*0.2728;//unit=kWh/day
printf("=%f kWh/day",E);
E=E*41.8//unit=W(continuous thermal)
printf("=%f W(continuous thermal)",E);
```

Chapter 8

Geothermal Energy

Scilab code Exa 8.5.1 `Plant efficiency and Heat rate`

```
//Ex8.5.1.;calculate: steam flow rate,cooling water flow,plant efficiency,Heat rate

//Enthalpy at point 1 at (31 kg/cm^2)=669.6 kcal/kg
//H1=H2=H3,enthalpy remain constant during throttling
H1=669.7;//unit= kcal/kg
H2=669.7;//unit= kcal/kg
H3=669.7;//unit= kcal/kg
//At point 3,
P3=9.55;//unit= kg/cm^2
//specific volume
vs3=0.22;//unit=m^3/kg
//Entropy
S3=1.580
T3=190;//unit=degree C,(degree of superheat=13 degree C)
//S4_s at 0.34 kg/cm^2=S3
//x4_s=0.838
//and H4_s=hs+xL
H4_s=72+(0.838*556)
```

```
printf(" H4_s=%f kcal/kg",H4_s)
//Isentropic turbine work=H3-H4_s
ITW=H3-H4_s;
printf("\n Isentropic turbine work=%f kcal/kg",ITW);
//Actual turbine work
ATW=0.80*ITW;
printf("\n Actual turbine work=%f kcal/kg",ATW);
H4=669.7-ATW;
printf("\n H4=%f kcal/kg",H4)
h5_6=72;//unit= kcal/kg; (Ignoring pump work)
//sensible heat  h7=h5=25 kcal/kg
h5=25;//unit=kcal/kg
h7=25;//unit=kcal/kg
//Turbine steam flow
TSF=(250*0.860*10^6)/(ATW*0.9);
printf("\n Turbine steam flow=%f kg/hr",TSF);
//let
m4=TSF;
//Turbine volume flow
TVF=(TSF/60)*vs3;
printf("\n Turbine volume flow=%f m^3/min",TVF);
//cooling water flow m7:m7(h5_6-h7)=m4(H4-h5_6)
m7=((H4-h5_6)/(h5_6-h7))*m4;
printf("\n cooling water flow m7=%f kg/hr",m7);
Heat_added=H1-h5_6;
printf("\n Heat_added=%f kcal/kg",Heat_added);
//plant efficiency=(Actual Turbine work*nmg)/Heat added
//nmg=combined mechanical and electrical efficiency of turbine-generator
nmg=0.90;
Plant_efficiency=(ATW*nmg)/Heat_added;
plant_efficiency=Plant_efficiency*100;
printf("\n Plant Efficiency nplant=%f persent",plant_efficiency);
//Plant heat rate=(860*Heat_added)/net_work
//net_work=105.36*0.90
Plant_heat_rate=(860/Plant_efficiency);
```

```
printf("\n Plant heat rate=%f kcal/kWH",
    Plant_heat_rate);


//The value of "turbine steam flow" is wrong due to
    calculating mistak in textbook,due to which the
    further value related with it is given wrong
//The values are corrected in this program
```

Scilab code Exa 8.5.2 Cycle efficiency and Plant Heat Rate

```
//Ex8.5.2.;calculate: hot water flow,condenser
    cooling water flow,cycle efficiency,plant heat
    rate.
H1=669.6;//unit=kcal/kg
H2=669.6;//unit=kcal/kg
//pressure at point 2,is 10.5 kg/cm^2;thus,
T2=195;//unit=degree celcius; (14 degree celcius of
    superheat)
s2=1.567;
vsup=0.27;
x3s=0.832;
H3s=535;//unit=kcal/kg
//Isentropic turbine work
ITW=H2-H3s;
printf(" Isentropic turbine work=%f kcal/kg",ITW);
//Actual turbine work
ATW=0.65*ITW;
printf("\n Actual turbine work=%f kcal/kg",ATW);
H3=669.6-ATW;
printf("\n H3=%f kcal/kg",H3)
//h_4-5(ignore bpump work)
h4=72.4;//unit=kcal/kg
//h6 at 27 degree c
h6=27;//unit=kcal/kg
```

```
//Turbine steam flow or hot water flow=power output/
   actual turbine work
TSF=(10*10^6*0.86)/ATW;
printf("\n Turbine steam flow or hot water flow=%f
   kg/hr",TSF);
//consider cooling water flow  m4:m3*(H3-h4)=m4(h4-
   h6)
//or
m4=((582.11-72.4)*0.983*10^5)/(72.4-27);
printf("\n cooling water flow=%f kg/hr",m4);
Heat_added=H1-h4
printf("\n Heat_added=%f kcal/kg",Heat_added);
//plant efficiency=Turbine work/Heat added
Plant_efficiency=(ATW/Heat_added);
plant_efficiency=Plant_efficiency*100;
printf("\n Plant Efficiency=%f persent",
   plant_efficiency);
//Plant heat rate=860/Plant Efficiency
Plant_heat_rate=860/Plant_efficiency;
printf("\n Plant heat rate=%f kcal/kWh",
   Plant_heat_rate);


//The value of m3=14.03*10^5 is given wrong in the
   text book;the actual value is m3=11.03*10^5
```

Chapter 9

Energy from the Oceans

Scilab code Exa 9.3.5.1 Energy Generated

```
//Ex9.3.5.1.;Calculate Energy generated
R=12;//unit=m; R is the range
r=3;//unit=m; the head below turbine stops operating
time=(44700/2);
A=30*10^6;
g=9.80;
p=1025;
//The total theoretical work W=integrate('1','w',R,r);
W=(g*p*A*((R^2)-(r^2)))/2;
printf(" W=%f ",W);
//The average power generated
Pav=W/time;//unit=watts
printf("\n The average power generated=%f watts",Pav);
pav=(Pav/1000)*3600;//unit=kWh
printf("\n The average power generated=%f kWh",pav)
//the energy generated
Energy_generated=pav*0.73
printf("\n Energy generated=%f kWh",Energy_generated);
```

Scilab code Exa 9.3.6.1 The Yearly Power Output

```
//Ex9.3.6.1; calculate power in h.p. at any instant
   and the yearly power output
A=0.5*10^6; //unit=m
h0=8.5; //unit=m
t=3*3600//unit=s; since t=3 hr
p=1025; //unit=kg/m^3
h=8; //unit=m
n0=0.70; //efficiency of the generator;70%
//volume of the basin=Ah0
volume_of_the_basin=A*h0;
//Average discharge Q=volume/time period
Q=(A*h0)/t;
printf(" volume of the basin=%f m^3 \n Average
   discharge Q=%f m^3 /s",volume_of_the_basin,Q);
//power at any instant
P=((Q*p*h)/75)*n0;
printf("\n power at any instant P=%f h.p.",P);
//The total energy in kWh/tidal cycle
E=P*0.736*3;
printf("\n The total energy in kWh/tidal cycle E=%f"
   ,E);
//Total number of tidal cycle in a year=705
printf("\n Total number of tidal cycle in a year=705
   ");
//Therefore Total output per annum
Total_output_per_annum=E*705;
printf("\n Total output per annum=%f kWh/year",
   Total_output_per_annum);

//The value of "power of instant" in a text book is
   misprinted.
```

Chapter 10

Chemical Energy Sources

Scilab code Exa 10.2.8.1 Reversible Voltage for Hydrogen oxygen fuel cell

```
//Ex10.2.8.1;Find Reversible voltage for hydrogen
    oxygen fuel cell
del_G=-237.3*10^3;//Joules/gm-mole of H2
//Reversible voltafe E of a cell is given by =
    del_Wrev/nF=-del_G/nF
//since 2 electrons are transferred per molecule of
    H2.thus
n=2;
F=96500;//Faraday's constant
E=-del_G/(n*F);
printf("Reversible voltage=%f volts",E);
```

Scilab code Exa 10.2.8.2 Voltage output and efficiency and heat transfer

```
//Ex10.2.8.2;calculate voltage output of cell,
    efficiency,electric work output,heat transfer to
    the surroundings

```

```
//1] voltage output of cell
del_G=-237.3*10^3;//Joules/gm-mole of H2
n=2;
F=96500;//Faraday's constant
E=-del_G/(n*F);
printf(" E=%f volts",E);
//2] Efficiency
//nmax=del_Wmax/-(del_H)25 degree celcuis = -(del_G)T/(-del_H)25
del_G_at298k=-56690;//unit=kcal/kg mole
del_H_at298k=-68317;//unit=kcal/kg mole
nmax=del_G_at298k/del_H_at298k,
printf("\n nmax=%f",nmax);
//3]Electric work output per mole
F=(96500/4.184);
del_Wrever=(n*F*E);
printf("\n Electric work output per mole=%f kcal/kg mole",del_Wrever);
//4] Heat transfer to the surroundings
//the heat transfer is Q=T*del-s=del_H_at298k-del_G_at298k
Q=del_H_at298k-del_G_at298k;
printf("\n The heat transfer is Q=%f kcal/kg mole",Q);
//The negative sign indicates that the heat is removed from the cell and transferred to the surrounding

//value of "Electric work output per mole" is approximate in the text book to the real calculated value
```

Scilab code Exa 10.2.8.3 Voltage output and efficiency and heat transferred

```
//Ex10.2.8.3;The heat transferred to the surrounding
```

```
del_G_at298k=-237191;//unit=kJ/kg mole
del_H_at298k=-285838;//unit=kJ/kg mole
ne=2;
F=96500;//Faraday's constant
E=-del_G_at298k/(ne*F);
printf(" E=%f volts",E);
nmax=del_G_at298k/del_H_at298k,
printf("\n nmax=%f",nmax);
nmax=nmax*100;
printf("=%f persent",nmax);
//Electric work output per mole of the fule is We=-
    del_G kJ/kg mole
We=del_G_at298k;//kJ/kg mole
printf("\n Electric work output per mole of the fule
     is We=%f kJ/kg mole",We);
//since there is 1 mol os H2O for each mole of fule,
    there is also a work output of 237191 kJ/kg mole
//Heat transferred is Q=T*del-s=del_H_at298k-
    del_G_at298k
Q=del_H_at298k-del_G_at298k;
printf("\n The heat transfer is Q=%f kJ/kg mole",Q);
//The negative sign indicates that the heat is
    removed from the cell and transferred to the
    surrounding

//value of "Electric work output per mole" is
    misprinted in the text book.
```

Scilab code Exa 10.2.8.4 Gibbs free energy and Entropy change

```
//Ex10.2.8.4;calculate del_G,del_S,del_H;

//We have the relation del_G=-n*F*E
//where,del_G=gibbs free energy of the system at 1
    atm and temperature(T)
```

```
n=1;//numbers of electons transferred per molecule
    of reactant
E=0.0455;//volts ;e.m.f. of the cell
F=96500;//Faraday's constant
//let X=dE/dT
X=0.000338;
del_G=-n*F*E;
printf(" del_G=%f joules",del_G);
//del_S = Entropy change of the system at
    temperature T and press p=1 atm in the case
del_S=n*F*(X);//del_S=n*F*(dE/dT)
printf("\n del_S=%f joules/deg.",del_S);
//And entropy change is given by the relation del_H=
    nF[T(dE/dT)-E]
T=298;
del_H=n*F*((T*X)-E);
printf("\n del_H=%f joule",del_H);


//value are taken approximate in the text book to
    the real calculated value
```

Scilab code Exa 10.2.8.5 Heat transfer rates

```
//Ex10.2.8.5;heat transfer rate would be involved
    under these circumstances

del_G_at25degree_celcius=-195500;//unit=cal/gm mole
del_H_at25degree_celcius=-212800;//unit=cal/gm mole
F=(96500/4.184);//since F=96500 coulombs/gm-mole
n=8
E_at25degree_celcius=-del_G_at25degree_celcius/(n*F)
    ;//Joules/coulomb
printf(" E_at25degree_celcius=%f volts=1.060 volts",
    E_at25degree_celcius);
```

```
//Max. efficiency  nmax=del_Wmax/-(del_H)at25 degree celcuis = -(del_G)T/(-del_H)25
nmax=del_G_at25degree_celcius/del_H_at25degree_celcius;
printf("\n nmax=%f",nmax);
//voltage efficiency nv=on load voltage/open circuit voltage=Operating voltage/Theoretical voltage
Theoretical_voltage=1.060/0.92;
printf("\n Theoretical_voltage=%f volts",Theoretical_voltage);
//power developed=100 kW=100*10^3 W
power_developed=(100*10^3)*0.86;//unit=kcal/hr; since 1 watt=1 joule/sec=0.86 kcal/hr
printf("\n power_developed=%f kcal/hr",power_developed);
del_G=-195500;
//Required flow rate of Methane
R_F_R_O_M=(power_developed*16)/del_G;//kg/hr;
//(methane moles)=16
printf("\n flow rate of Methane=%f kg/hr",R_F_R_O_M);
//Heat transfer Q=T8del_s=del_H+del_w=del_H-del_G
Q=del_H_at25degree_celcius-del_G_at25degree_celcius;
printf("\n The heat transfer is Q=%f kcal/kg mole",Q);

//The value are approximate in the text book to the real calculated value
//value of "Required flow rate of methane" is wrong in the text book.
//value of "Heat transfer" is wrong in the text book.
```

Chapter 12

Magneto Hydro Dynamic Power Generation

Scilab code Exa 12.6.1 Open circuit voltage and maximum power output

```
//Ex12.6.1.;calculate open circuit voltage and
   maximum power output
B=2;//flux density;unit=Wb/m^2
u=10^3;//average gas velocity;unit=m/second
d=0.50;//distance between plates;unit=m
E0=B*u*d;//Open ccircuit voltage
printf(" Open ccircuit voltage E0=%f Volts",E0);
//Generator resistance; Rg=d/sigma*A
sigma=10;//Gaseous conductivity;unit=Mho/m
A=0.25;//Plate Area;unit=m^2
Rg=d/(sigma*A);
printf("\n Generator resistance Rg=%f Ohm",Rg);
//Maximum power
Maximum_power=(E0^2)/(4*Rg);
printf("\n Maximum_power=%f watts",Maximum_power);
```

Chapter 13

Thermo Electric Power

Scilab code Exa 13.2.1 Peltier heats absorbed and rejected

```
//Ex13.2.1.;Peltier heats absorbed and rejected
//peltier coefficients at these junctions are aplha_p_1-2=alpha_s_1-2*T
//Let A=alpha_s_1-2 at 373 k=55*10^-6 v/degree_k and B=alpha_s_1-2 at 273 k=50*10^-6 v/degree_k
A=(55*10^-6);
B=(50*10^-6);
T1=373;//k
T2=273;//k
I=10*10^-3;//current;unit=Ampere
alpha_p_1_2_at_373k=A*T1;
alpha_p_1_2_at_273k=B*T2;
printf(" alpha_p_1_2_at_373k=%f W/amp \n alpha_p_1_2_at_273k=%f W/amp",alpha_p_1_2_at_373k=A*T1,alpha_p_1_2_at_273k=B*T2);
//Peltier heats absorned and rejected to be
q2_peltier=alpha_p_1_2_at_373k*I;
q1_peltier=alpha_p_1_2_at_273k*I;
printf("\n q2_peltier=%f w \n q1_peltier=%f W",q2_peltier,q1_peltier);
c=q2_peltier-q1_peltier;
```

```
printf("\n If no other heat transfer were involved,
    the difference between these vaues,");
printf("\n %f-%f=%f W,would be supplied as electric
    power",q2_peltier,q1_peltier,c);
```

Scilab code Exa 13.3.2 Thomson heat transferred

```
//Ex.13.3.2.;Find the thomson heat transferred


//Let D=dalpha_s1/dT;
D=5.4*10^-3;//unit=micro V/degree k^2
T1=273;//unit=k
T2=373;//unit=k
I=10*10^-3;//unit=A
//Thomson coefficient sigma,varies with temp.
//sigma_1_of_T=-T*D;unit=V/degree k
//The thomson heat is given by equation
//qth=I*Integration of sigma_1_of_T w.r.t. T
Integration=integrate('T','T',T1,T2);
qth=I*D*Integration;
printf("The THOMSON HEAT=%f micro W",qth);
```

Scilab code Exa 13.4.1 Carnot Efficiency

```
//Ex13.4.1.;Determine the efficiency of the
    thermoelectric generator.what will be its carnot
    efficiency

TH=600;//degree k;//temperature of the hot reservior
     of source
TC=300;//degree k;//temperature of the sink
```

```
Z=2*(10^-3);//1/degree k;//Figure of merit for the material
M_optimum=(1+((Z/2)*(TH+TC)))^0.5;
printf(" M_optimum=%f",M_optimum);
//Efficiency of the thermoelectric generator is n=(((TH-TC)/TH)*((M_optimum-1)/(M_optimum+(TC/TH)))*100;
a=((TH-TC)/TH);
b=(M_optimum-1)/(M_optimum+(TC/TH));
n=a*b*100;
printf("\n Efficiency of the thermoelectric generator is n=%f persent",n);
//where as efficiency of the carnot cycle (reversible) nc=((TH-TC)/TH)*100
nc=a*100;
printf("\n Efficiency of the carnot cycle (reversible) nc=%f persent",nc);
```

Scilab code Exa 13.4.2 Maximum generator efficiency and Power Output

```
//Ex13.4.12.;Calculare maximum generator efficiency and the efficiency for maximum power,power output

//seedbeck coefficient(alpha_s);unit=volts/degree celcius
alpha_s1=-190*10^-6;//n-type
alpha_s2=190*10^-6;//p-type
//Specific resistivity(p);unit=Ohm-cm
p1=1.45*10^-3;//n-type
p2=1.8*10^-3;//p-type
//Figure of merit(Z);unit=degree k^-1
Z1=2*10^-3;//n-type
Z2=1.7*10^-3;//p-type
```

```
//conductivity (n-type),
k1=(alpha_s1^2)/(p1*Z1);
//similarly
k2=(alpha_s2^2)/(p2*Z2);
printf(" Conductivity k1=%f W/cm degree celcius \n Conductivity k2=%f W/cm degree celcius",k1,k2);
//Z_opt=((alpha_s1-alpha_s2)^2)/[(p1*k1)^2+(p2*k2)^2];
//let
a=(alpha_s1-alpha_s2)
b=(p1*k1)
c=(p2*k2)
A=sqrt(b)
B=sqrt(c)
C=(A+B);
///therefore
Z_opt=(a/C)^2;
printf("\n Z_opt=%f degree k",Z_opt);
//Thermal conductance
A1=2.3;//cm^2
A2=1.303;//cm^2
l1=1.5;//cm
l2=0.653;//cm
K=((k1*A1)/l1)+((k2*A2)/l2)
printf("\n Thermal conductance K=%f W/degree celcius ",K);
//R=Resistance of the generator=R1+R2
R=((p1*l1)/A1)+((p2*l2)/A2);
printf("\n Resistance of the generator R=%f ohm",R);
TH=923;//unit=k
TC=323;//unit=k
M_opt=(1+((Z_opt/2)*(TH+TC)))^0.5;
printf("\n M_opt=%f ohm",M_opt);
RL=M_opt*R;
printf("\n RL=%f ohms",RL);
//Optimum efficiency n_opt=(((TH-TC)/TH)*((M_opt-1)/(M_opt+(TC/TH)))*100;
aa=((TH-TC)/TH);
```

```
//taking M_opt=1.43
b=(1.43-1)/(1.43+(TC/TH));
n_opt=aa*b*100;
printf("\n Optimum efficiency n_opt=%f persent",n_opt);
//efficiency for max. power output n= (TH-TC)/TH)*m/[((1+m)^2/TH)*(KR/alpha_s_12^2)+(1+m)-(TH-TC)/2TH)]
//Efficiency power output
//RL=R i.e. m=1
// let ab=(1+m)^2/TH;ac=(KR/alpha_s_12^2);ad=(TH-TC)/2TH
m=1;
ab=4/TH;
ac=1/Z_opt;
ad=aa/2;
n_max=[aa/(ab*ac+2-ad)]*100;
printf("\n max. power output n_max %f persent",n_max)
//Power output P_opt=I^2*RL=alpha_s12^2(TH-TC)*RL/(R+RL)^2=alpha_s12^2(TH-TC)/(1+M_opt)^2*RL
//let at=alpha_s12^2(TH-TC);mi=(1+M_opt)^2*RL
at=a*a*(TH-TC)*(TH-TC);
ml=(1+1.43)*(1+1.43)*2.63*10^-3
P_opt=at/ml;
printf("\n Power output P_opt=%f watts",P_opt);
//for max. power P_max (RL=R)
//P_max=alpha_s12^2(TH-TC)*RL/(r+RL)^2=alpha_s12^2(TH-TC)RL*4RL
P_max=at/(4*1.84*10^-3);
printf("\n max. power P_max=%f watts",P_max);


//Many calcuating mistak are there in a following example,which is corrected in program.
```

Scilab code Exa 13.4.3 Maximum efficiency and Thermocouples in series and also Hea

```
//Ex.13.4.3;maximum efficiency,no. of thermocouple in series,open ckt voltage,heat i/p and reject at full load.

kA=0.02;//unit=watt/cm degree kelvin
kB=0.03;//unit=watt/cm degree kelvin
pA=0.01;//unit=ohm cm
pB=0.012;//unit=ohm cm
TH=1500;//unit=degree kelvin
TC=1000;//unit=degree kelvin
AA=43.5;//unit=cm^2
AB=48.6;//unit=cm^2
LA=0.49;//unit=cm
LB=0.49;//unit=cm
I=20*48.6;//Current density in the element limited to,I=20 amp/cm^2
output=100;//unit=kW
//alpha_SAB at 1250 degree kelvin=0.0012 volt/degree kelvin=alpha_SA-alpha_SB
alpha_SAB=0.0012;//unit=volt/degree kelvin
//let
b=(pA*kA);
c=(pB*kB);
A=sqrt(b);
B=sqrt(c);
C=(A+B);
//figure of merit
Z=(alpha_SAB/C)^2;
printf(" Z=%f degree k^-1",Z);
M=(1+((Z/2)*(TH+TC)))^0.5;
printf("\n M=%f",M);
//let
```

```
aa=((TH-TC)/TH);
bb=(M-1)/(M+(TC/TH));
//1] MAx. efficiency of a thermoelectric converter
   is given by n_max=((TH-TC)/TH)*[(M-1)/(M+(TC/TH))
   ]*100;
n_max=aa*bb*100;
printf("\n Maximum efficiency n_max=%f persent",
   n_max);
//2] No. of thermocouple in series
V=alpha_SAB*(TH-TC);
printf("\n V=%f volt",V);
R=((pA*LA)/AA)+((pB*LB)/AB);//since R=RA+RB=((pA*LA
   )/AA)+((pB*LB)/AB);
printf("\n R=%f ohm",R);
VL=V-(R*I);
printf("\n VL=%f volt",VL);
//NTCS=total voltage required/voltage required by
   one couple
NTCS=115/VL;
printf("\n No. of thermocouple in series=%f",NTCS);
//3] Open circuit voltage
OCV=V*309;
printf("\n Open circuit voltage=%f volt",OCV)
//4] Heat input and reject at full load.
//Heat input at full load.=output/efficency
   =100/0.091
HIFL=output/(n_max/100);
printf("\n Heat input at full load=%f kW",HIFL)
// Heat reject at full load. =Heat input-Work output
HRFL=HIFL-output;
printf("\n Heat reject at full load=%f kW",HRFL)



//The value of "pB" is misprinted
//The values are taken in the text book is
   approximately equal to calculated values
```

Chapter 14

Thermionic Generation

Scilab code Exa 14.4.1 Efficiency of the generator and Carnot efficiency

```
//Ex.14.4.1.;Calculate the efficiency of the
    generator and also compare with the carnot
    efficiency

//cathode work funtion
flux_c=2.5;//unit=volts
//anode work funtion
flux_a=2;//unit=volts
//Temp. of cathode
Tc=2000;//unit=degree k
//Temp. of surrounding
Ts=1000;//unit=degree k
//plasma potentail drop
flux_p=0.1;//unit=volts
//Net output voltage
V=flux_c-flux_a-flux_p
printf(" V=%f volt",V);
//charge of an electron
e=1.6*10^-19;//unit=coulomb
//boltzmann constant
k=1.38*10^-23;//unit=joule/degree kelvin
```

```
A=1.20*10^6;
//one electron volt=1.6*10^-19 joule
//The net current in the generator J=J_cathode-J_anode
//let EC=e^(-flux_c/k*Tc)
EC=%e^[-(1.6*10^-19*flux_c)/(k*Tc)];
J_cathode=A*(Tc*Tc)*EC;//J_cathode=A*Tc^2*e^(-flux_c/k*Tc)
printf("\n J_cathode=%f amp/m^2",J_cathode);
//let EA=e^(-flux_c/k*Ts)
EA=%e^[-(1.6*10^-19*flux_a)/(k*Ts)];
J_anode=A*(Ts^2)*EA;//J_cathode=A*Ts^2*e^(-flux_c/k*Ts)
printf("\n J_anode=%f amp/m^2",J_anode);
//The net current can be taken =Jc,as Ja can be neglected in comparison with Jc
J=J_cathode;
printf("\n J=%f amp/m^2",J);
//The heat supplied to the cathode Qc/Ac=J(flux_c+((2*k*Tc)/e))+samestion of sigma*(Tc^4-Ts^4)
//let QA=Qc/Ac; and
a=2.5+((2*1.38*10^-23*2000)/(1.6*10^-19));
b=J*a;
c=(0.2*5.67*(10^-12)*(10^-4)*((2000^4)-(1000^4)));
//therefore
QA=b+c; //since: QA=(J*(2.5+((2*(1.38*10^-23)*2000*)/(1.6*10^-19))))+(0.2*5.67*(10^-12)*(10^-4)*((2000^4)-(1000^4)))
printf("\n The heat supplied to the cathode Qc/Ac=%f watt/m^2",QA);
//efficiency of the generator
ng=((J*V)/(7.026*10^6))*100;
printf("\n ng=%f persent",ng);
//carnot efficiency this device
T1=2000;
T2=1000;
T=2000;
nc=((T1-T2)/T)*100;
```

```
printf("\n nc=%f persent",nc);


//Value of "The heat supplied to the cathode Qc/Ac"
    is given wrong
//value of charge e is taken wrong;corrected by
    giving value 1.6*10^-19
//value of J anode is differ from calculated value.
```