ส่วนของโปรแกรมย่อยต่างๆ ที่ใช้ในการวิเคราะห์ข้อมูลที่ได้จากการทดสอบแผงเซลล์แสงอาทิตย์ พัฒนาโดยใช้โปรแกรม MATLAB version 5.3.1

ส่วนของโปรแกรมย่อยในการหาค่ากำลังสูงสุด

```
function  maxpoint
clear all;
close all;
clc;
%**************  CURRENT  DATA ************************
a_data=[1.2  1.2  1.2  1.2  1.1  1.1  1.1  1.1  1.1  0.8  0.6;
1.5  1.5  1.5  1.5  1.5  1.5  1.5  1.5  1.4  0.8  0.5;
2.0 1.98  1.95  1.9  1.9  1.9  1.8  1.8  1.6  0.8  0.6;
2.1  2.1  2.1  2.1  2.1  2.1  2.1  1.98  1.6  0.8  0.6;
2.2  2.2  2.2  2.1  2.1  2.1  2.1  1.98  1.6  0.8  0.6;
2.8  2.8  2.8  1.6  1.6  1.6  1.6  1.6  1.6  0.8  0.5;
2.4  2.4  2.4  2.4  2.4  2.1  2.1  2.1  1.7  0.8  0.6;
2.8  2.7  2.7  2.7  2.7  2.6  2.4  2.0  1.6  0.8  0.5;
2.75  2.7  2.6  2.6  2.6  2.6  2.4  2.0  1.6  0.8  0.6;
2.8  2.8  2.8  2.8  2.8  2.8  2.4  1.8  1.3  0.8  0.5;
2.6  2.6  2.6  2.6  2.6  2.6  2.4  2.0  1.6  0.8  0.6;
1.6  1.0  1.0  1.0  0.9  0.9  0.9  0.9  0.9  0.7  0.5;
0.3  0.3  0.3  0.3  0.3  0.3  0.3  0.3  0.3  0.3  0.3;
2.4  2.4  2.4  2.4  2.4  2.4  2.2  2.0  1.6  0.8  0.5;
2.45  2.4  2.4  2.4  2.4  2.4  2.3  2.0  1.6  0.8  0.5;
0.8  0.8  0.8  0.8  0.8  0.8  0.8  0.8  0.75  0.7  0.5];

%********  Volt  ***************
v_data=[0.0412  0.74  1.71  4  4.67  4.87  5.78  7.8  10.41  16.63  16.8;
0.0573  0.93  2.25  5.12  6.16  6.44  7.75  10.28  13.52  16.67  17.25;
0.0764 1.183  2.75  6.40  7.53  8.01  9.39  12.34  16.49  16.96  17.2;
0.0842 1.33  3.02  7.18  8.14  8.5  10.1  13.39  14.98  16.83  17.24;
0.0715  1.42  3.16  7.46  8.36  8.85  10.69  13.44  14.95  16.7  17.1;
0.0869 1.75  3.45  4.7  5.6  6.6  7.27  8.4  15.21  15.87  16.37;
0.0543  0.92  2.2  7.84  9.27  9.5  10.8  14.4  15.6  17.31  17.41;
```

ตัวอย่างข้อมูลทดสอบ (ข้อมูลกระแส)

ตัวอย่างข้อมูลทดสอบ (ข้อมูลแรงดัน)

```
0.1068  0.925  3.53  9.4  10.8  10.86  12.59  14.10  15.02  16.49  16.86;
0.0997  1.42  3.77  8.94  10.27  10.84  12.45  13.85  14.9  16.35  16.7;
0  2.0  3.0  8.0  9.0  10.0  11.0  11.5  14.0  15.0  16.0;
0.12  1.41  3.53  8.50  10.14  10.74  12.22  14.12  15.21  16.91  17.28;
0.0702  0.546  1.45  3.46  3.97  3.99  4.88  6.54  8.70  14.49  15.36;
0.0127  0.0153  0.42  1.01  1.21  1.28  1.57  2.114  2.85  6.73  9.61;
0.1207  1.43  3.09  7.56  8.96  9.46  11.49  13.60  14.83  16.56  16.99;
0.1301  1.27  3.45  8.15  9.41  9.96  11.56  13.95  15.24  16.94  17.34;
0.0411  0.424  1.09  2.588  2.90  3.40  4.08  5.56  6.99  13.90  15.65];

n_data=1; %number of data

%************************************
%( for loop(1) )
for   nn=1:1:n_data
    vdata=v_data;
    adata=a_data;
  %( for loop(2) )
  vdata_transpost=vdata';%
  rownumber1=size(vdata,1);
    for  n=1:1:rownumber1
        Voltdata=vdata(n,:);
        Ampdata=adata(n,:);
    vn=Voltdata;
    an=Ampdata;
    va=(vn).*(an);
    [yn,indexn1]=max(va);
    max_point_order(1,n)=vn(:,indexn1); %  1 x n matrix(or ROW VECTOR)
    va_max(1,n)=yn; % ROW VECTOR
    van(n,:)=va; % n x m matrix
  end %(END for loop(2) )

  % show data (COLUME VECTOR)
  va_max_show=va_max';%
  vmax_point_order_show=vmax_point_order';%
```

```
  va_max_final=va_max;%%
  vmax_point_order_final=vmax_point_order;%%
  POWER_VA=van;

  VA_max_final(nn,:)=va_max_final;
  Vmax_point(nn,:)=vmax_point_order;

end %(END for loop(1) )

%show results
number_row=size(vdata,1);
number_time1=1:1:(number_row);

number_time=number_time1';%%
VA_max_show=VA_max_final';%%
Vmax_point_show=Vmax_point';%%

z(:,1)=number_time;
z(:,2)=VA_max_show;
z(:,3)=Vmax_point_show;
Number_time=z;

clc;
disp('    ----------------------------------------------------');
disp('                                                  Maximum power point    ');
disp('    ----------------------------------------------------');
POWER_VA
disp('    ----------------------------------------------------');
disp('    ');
disp('    ');
disp('    ');
disp('     Please Enter to continue  !!!      ' );

pause;

clc;
```

```
disp('              ');
disp('    ----------------------------------------------------');
fprintf('      Maximum power point : %g     set\n',n_data);
disp('    ----------------------------------------------------');
disp('             VA_max   Voltage point at Maximum power)   ');
Number_time
disp('    ----------------------------------------------------');
disp('    ');
disp('    ');
disp('    ');
disp('     Please Enter to continue  !!!      ' );
pause;
clear all;
close all;
clc;
```

ส่วนของโปรแกรมย่อยในการหาฟังก์ชันพหุนาม (Polynomial function) อันดับใดๆ

```
%% Polynomial Order n
%% Discharge curve
function   intn=icn(x)
load inn -ascii
load ordern -ascii
%^^^^^^^^^^^^^^^ SHOW  polynomial n-order ^^^^^^^^^^^^
coeff=inn;
ordern1=ordern;
for c=1:1:length(x)
  p=0;
  for ii=1:1:length(coeff)
    w=coeff(:,ii).*(x(:,c).^(ii-1));
    p=p+w;
  end
  pp(c)=p;
end
intn=pp;
```

ส่วนของโปรแกรมย่อยในการหาฟังก์ชันที่เหมาะสมในการประมาณค่า

```
%APPROXIMATION  FUNCTION
function  n_order_fitting
clear all;
close all;
clc;

Y=1;
y=1;
N=0;
n=0;

clc;
disp('              ');
disp('              ');
disp('              ');
disp('              ');
disp('                          ');
tes=input('        Do you want to input data(y or n):');

%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
%                                    (NO)
%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
% USE  Standard Data for test program procedure
% if not input data  or tes=0(NO,no)
%if tes==0
if tes==0

clc;
  disp('              ');
  norder=input('        Enter the order of polynomial:');
  disp('                          ');

Y=1;
```

```
y=1;
N=0;
n=0;
disp('               ');
disp('                        ');
tesno=input('      Do you want to Edit data(y or n):'); %Delete
if tesno==1    %Delete
 while  tesno==1

  clc;
  disp('               ');
  disp('               ');
   norder=input('         Enter the order of polynomial:');
  disp('               ');
  tesno=input('      Do you want to Edit data(y or n):');
 end
end

% Standard Data for test program procedure
% Time interval
y1 = [0 0.2310 1.0710 1.3860 1.9320 2.0580 2.1000 1.9740 1.6590 1.3020 0.6300 0.2940 0];
x1=1:1:length(y1);

clc;
num0=1:1:length(x1);
z(:,1)=num0';
z(:,2)=x1';
z(:,3)=y1';
ni=z;

disp('    ----------------------------------------------------');
fprintf('      order fo polynomial : %g     \n',norder);
disp('    ----------------------------------------------------');
disp('          x      f(x)   ');
ni
disp('    ----------------------------------------------------');
```

```
end

%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
%                                   (YES)
%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
%INPUT DATA (NEW)
% if tes=1(YES,yes)
%tes=1;
if  tes==1

while  tes==1
clc;
    disp('              ');
    disp('                              ');
      norder=input('   Enter the n-order of polynomial:');
      disp('                              ');
      y1=input('  Enter the value of y:');
      disp('                              ');
      x1=input('  Enter the value of x:');
      disp('                              ');
      tes=input('         Do you want to Edit data(y or n):');
   clc;
   end
clc;

num1=1:1:length(x1);
z(:,1)=num1';
z(:,2)=x1';
z(:,3)=y1';
ni=z;
disp('              ');
disp('    ---------------------------------------------------');
fprintf('      order fo polynomial : %g     \n',norder);
disp('    ---------------------------------------------------');
disp('         x      f(x)   ');
```

```
ni
%fprintf('     %g     \n',z);
%fprintf('  ni %g         data of x: %g         data of y: %g   \n',num1,x1,y1);
disp('    ---------------------------------------------------');
end
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
disp('                    ')
disp('                                        Please Enter to continue  !!!       ');
pause;

% APPROXIMATION  FUNCTION
%% Calculate Area of curve
%% Order n approximate

% INPUT DATA  FOR approximation
nn=length(x1);
clc;
[coef_n,ss]=polyfit(x1,y1,norder);
coef_n;
ss;

xn=x1;
[ap1,delta1]=polyval(coef_n,xn,ss);
ap1;
delta1;

q1=y1-ap1;
q2=(q1.^2);
q3=sum(q2);               % sum of residual

%deno  term
y21=(y1.^2);
y22=sum(y21);

y11=sum(y1);
y12=(y11.^2);
```

```
y13=(y12/nn);

yy=y22-y13;
%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
cd1=sqrt(1-(q3/yy));            %  coefficient of determination

sd1=sqrt(q3/(nn-(norder+1)));       %  standard deviation

%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
clc;
  disp('                          ');
   disp('                          ');
   disp('                          ');
   disp(' ____________________________________________________________________');
   disp('                          ');
   fprintf('                      ORDER  %g      \n',norder);
   disp('                                        ');
   fprintf('      sum of  residual  squares   :   %f       \n',q3);
   fprintf('   coefficient of determination   :   %f       \n',cd1);
   fprintf('            standard deviation   :   %f       \n',sd1);
    disp('                          ');
   disp(' ____________________________________________________________________');
   disp('                          ');

pause;

%END

% FINE OPTIMAL ORDER AND LEAST SQUAR
% STEP1 order(1)
nn=length(x1);
coef_n =polyfit(x1,y1,1);
xn=x1;
ap1 =polyval(coef_n,xn);

q1=y1-ap1;
```

```
q2=(q1.^2);
q31=sum(q2);                    % sum of residual

%deno  term
y21=(y1.^2);
y22=sum(y21);

y11=sum(y1);
y12=(y11.^2);
y13=(y12/nn);

yy=y22-y13;
qq3(:,1)=q31;
%!!!!!!!!!!!!!!!!!!!!!!!!!! END  of 1-step !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!1

%     STEP order(2)
nn=length(x1);
coef_n=polyfit(x1,y1,2);
xn=x1;
ap1=polyval(coef_n,xn);

q1=y1-ap1;
q2=(q1.^2);
q32=sum(q2);                    % sum of residual

%deno  term
y21=(y1.^2);
y22=sum(y21);

y11=sum(y1);
y12=(y11.^2);
y13=(y12/nn);

yy=y22-y13;
qq3(:,2)=q32;
```

```
%!!!!!!!!!!!!!!!!!!!!!!!!!! END  of 2-step !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!1

%  SUB program  for  FINE  optimal order
qq3;
o=2;

while   qq3(o)< qq3(o-1)  %| sd1(o)> sd1(o-1)

        o=o+1;   %increase  step
nn=length(x1);
coef_n=polyfit(x1,y1,o);
xn=x1;
ap1=polyval(coef_n,xn);

q1=y1-ap1;
q2=(q1.^2);
qq3(o)=sum(q2);                  % sum of residual

%deno  term
y21=(y1.^2);
y22=sum(y21);

y11=sum(y1);
y12=(y11.^2);
y13=(y12/nn);

yy=y22-y13;
cd1(o)=sqrt(1-(qq3(o)/yy));              %  coefficient of determination
sd1(o)=sqrt(qq3(o)/(nn-(o+1)));         %  standard deviation

end
o=o-1;
nn=length(x1);
coef_n=polyfit(x1,y1,o);
xn=x1;
ap1=polyval(coef_n,xn);
```

```
q1=y1-ap1;
q2=(q1.^2);
qq3=sum(q2);                 % sum of residual

%deno  term
y21=(y1.^2);
y22=sum(y21);

y11=sum(y1);
y12=(y11.^2);
y13=(y12/nn);

yy=y22-y13;
cd1=sqrt(1-(qq3/yy));             %  coefficient of determination
sd1=sqrt(qq3/(nn-(o+1)));         %  standard deviation

orderapp=o;
coef_n=coef_n;
Q3=qq3;
Cd1=cd1;
Sd1=sd1;

pause;
%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
clc;
 disp('                          ');
  disp('                          ');
  disp('                          ');
  disp(' ________________________________________________________________________');
  disp('                          ');
  fprintf('                  OPIMAL ORDER  %g      \n',o);
  disp('                                              ');
  fprintf('     sum of  residual  squares   :   %f       \n',Q3);
  fprintf('   coefficient of determination   :   %f       \n',Cd1);
  fprintf('           standard deviation   :   %f       \n',Sd1);
```

```
 disp('                          ');
disp(' ________________________________________________________________________');
disp('                          ');
pause;


%!!!!!!!!!!!!!!!!!!!!!!!!!! polynomial order n !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% COEFFICIENT of polynomial function
order1=o;
coef_n=polyfit(x1,y1,order1);
coefficient_ordern=coef_n;

an=polyval(coef_n,x1);
x1=x1;
coeffn=coef_n;
posi=o+2;
order2=o+1;
for n=1:order2
  nn=posi-n;
  co(n)=coeffn(:,nn); %co(1) = coefficient order (0)
end
cof=co;

%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
%                  Estimation  of (x)
%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Y=1;
y=1;
N=0;
n=0;
clc;
disp('            ');
disp('            ');
disp('            ');
disp('            ');
disp('                          ');
```

```
tes=input('        Do you want to estimation (y or n):');

%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%INPUT DATA (NEW)
% if tes=1(YES,yes)
%tes=1;
if  tes==1

while  tes==1
clc;
     disp('              ');
     disp('                          ');
     xapp=input('        Enter the value of (x) :');
     disp('              ');
     disp('                          ');
     tes=input('         Do you want to Edit data(y or n):');
   clc;
   end
clc;
ap2=polyval(coef_n,xapp);      % approimation !!!!!
disp('                          ');
disp('                          ');
disp('                          estimated    value                ');
fprintf('                           ORDER  %g     \n',o);
disp(' ____________________________________________________________________');
  disp('                          ');
  fprintf('             x =  %g                    f(x) = %g       \n',xapp,ap2);
  disp('                           ');
disp(' ____________________________________________________________________');
  disp('                          ');
pause;

end
% save coefficient for Integrate  function

x=x1;
```

```
save inn  cof -ascii
save ordern  o -ascii
%^^^^^^^^^^^^^^^ SHOW  polynomial n-order ^^^^^^^^^^^^^^^^^^^^
x=x1;
coeff=cof;  % from less to most
ordern=o;

%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
%                     Integration of function  f(x)
%^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Y=1;
y=1;
N=0;
n=0;

clc;
disp('              ');
disp('              ');
disp('              ');
disp('              ');
disp('                              ');
tes=input('   Do you want to input data(lower,upper); (y or n):');

%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%INPUT DATA (NEW)
% if tes=1(YES,yes)
% tes=1;
if  tes==1

while  tes==1
clc;
     disp('              ');
     disp('                              ');
     disp('              ');
     low1=input('        Enter the value of lower limit :');
     disp('                              ');
```

```
    upp1=input('        Enter the value of upper limit :');
    disp('              ');
    disp('              ');
    disp('                          ');
    tes=input('         Do you want to Edit data(y or n):');
  clc;
  end
clc;
% integration
arean=quad('icn',low1,upp1);  %%%%^^^^^^^^^^^^(quad8 or quad)
AREA_n=arean;
clc;
 disp('                         ');
  disp('                         ');
  disp('                         ');
  disp(' ________________________________________________________________');
  disp('                         ');
  fprintf('                  OPIMAL ORDER  %g      \n',o);
  disp('                                        ');
  fprintf('     lower limit of integration  :   %f       \n',low1);
  fprintf('     upper limit of integration  :   %f       \n',upp1);
  fprintf('           Area of function     :   %f       \n',arean);
   disp('                         ');
  disp(' ________________________________________________________________');
  disp('                         ');


pause;
end
clc;
clear;
```

ส่วนของโปรแกรมหลักที่ใช้ในการหาปริมาณการอัดประจุ และคายประจุ เพื่อประกอบการพิจารณาขนาดของแบตเตอรี่

```
% sizing of Battery Version 1.4
function   sizer v4

% Start SUB (plot)
r=rl;   % initial condition of load
load pvdata -ascii
pvdata;
pn=pvdata;

load  von  -ascii
von;
voltage_v1=von;

Y=1;
y=1;
N=0;
n=0;
clc;
disp('              ');
disp('                                ');
disp('                                ');
tes=input('      Do you want to plot 3-D (y or n):');
disp('                                ');
disp('                                ');

%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%  PLOT  3-D
% if not input data  or tes=1(YES,yes)
if tes==1
 %plot
 for i=0:1:23
```

```
            for j=0:1:18
                z=pn;
    end
  end
% voltage data of 3-D plot
  for i=7:1:17
  h=7:17;
  vii=0:1:18;
  ill=((vii)./(r)) ;
    for  j=0:1:18

      v1=voltage_v1;

    end
   end
il1=((v1)./(r));

mesh(z,[-87,-23]),...
% [Azimuth,Elevation]
xlabel('volt(V)');
ylabel('Time');
zlabel('current(A)');
title(' I-V Characteristic  ')
grid;
disp('              ');
disp('              ');
disp('               Please  Enter to contour plot !!!');
pause;

%  END  PLOT stage 1
% Contour plot
clc;
disp('       ');
disp('       ');
disp('       ');
disp('       ');
```

```
licon=input('        Enter the number of contour line :');
disp('       ');
disp('       ');

% start plot
zz=z;
contour(zz,licon),...
grid;
title(' 3-D of contour plot  '),...
xlabel('volt(V)');
ylabel('Time');
%grid;
clc;
disp('              ');
disp('              ');
disp('               Please  Enter to continue !!!');
pause;
%  END  PLOT  of contour
Y=1;
y=1;
N=0;
n=0;
clc;
disp('              ')
disp('              ')
tesno=input('         Do you want to Edit data (y or n):');
disp('              ')
disp('              ')
 if tesno==1
   while  tesno==1
   disp('              ');
   disp('              ');
    az=input(' Enter the value of Azimuth [(-x)=clockwise,(+x)=counter clockwise]:');
   disp('       ');
    el=input('                Enter the value of Elevation [(+x)=UP,(-x)=DOWN ]:');
          %data for 3-D plot
```

```
 for i=0:1:23
          for j=0:1:18
                        z=z;  %z to z
            end
  end
% voltage data of 3-D plot
      for i=7:1:17
         h=7:17;
         vii=0:1:18;
         ill=(vii)./r ;
        for  j=0:1:18
          v1=v1;   %v1 to v1
        end

      end
      il1=(v1)./r;

     mesh(z,[az,el]),...
     % [Azimuth,Elevation]
     xlabel('volt(V)');
     ylabel('Time');
     zlabel('current(I)');
     title(' I-V Characteristic  ');
     grid;

     Y=1;
     y=1;
     N=0;
     n=0;
     clc;
     disp('              ');
     disp('              ');
     disp('              ');
     tesno=input('       Do you want to plot(3-D) AGAIN ! (y or n):');
     disp('              ');
     disp('              ');
```

```
    end
  end



end  %%%END of sub (plot)
clc;
disp('       ');
disp('       ');
disp('                !!!!!    data plot complete    !!!!!');
disp('          ');
disp('          ');
disp('                    Please  Enter  to continue  !!!        ');
disp('       ');
pause;
% END SUB(plot)
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% load data of PV at 1.00 - 24.00(or 0.00)
load pvdata -ascii
pvdata;
pn=pvdata;

pnt=(pn)';  %show data
%procedure for interpolation function
pv=interp1(pnt,vb,'nearest');
ipv_nearest=pv;
pv1=interp1(pnt,vb,'linear');
ipv_linear=pv1;
%USE spine
pv2=interp1(pnt,vb,'spline');
ipv_spline=pv2;
%
pv3=interp1(pnt,12,'cubic');
ipv_cubic=pv3;
%plot curve of PV system
ipvn=ipv_spline;              % show
m=1:length(ipvn);
```

```
subplot(311);
plot(m,ipvn);
grid;
title('Ipv plot');
xlabel('time(H)');
ylabel('Currrnt  of  PV(A)');
axis([0 24 0 5]);
% Voc approximate
vocapp=((length(pnt))-1); %number of data voltage
volapp=((length(pvdata))-1);%number of data value of pv
v=0:((length(pn))-1);  %0 - Voc
v1=0:((length(pnt))-1); %0 - 23 (24 value)

vn=0:17;  % 0-Voc(0-17) or (18 value)  !!! vnn=vn+1 !!!
il1=(vn)./rl;
il=il1; % show data
%load calculation
tll=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
ill=il';
iload=(ill)*(tll);
vbb=vb+1; %for effect of load-line
il_inter=interp1(iload,vbb,'spline');
il_spline=il_inter;
%plot curve of LOAD
iln=il_spline;                % show
mm=1:length(iln);
subplot(312);
plot(mm,iln);
title('Iload plot');
xlabel('time(H)');
ylabel('Currrnt  of  Load(A)');
grid;
% Battery Current
ib=ipvn-iln;
ibn=ib;         % show
mmm=1:length(ibn);
```

```
subplot(313);
plot(mmm,ibn);
title('Ib plot');
xlabel('time(H)');
ylabel('Currrnt  of  Battery(A)');
grid;
clc;
disp('              ');
disp('              ');
disp('              ');
disp('              ');
disp('               Please   Enter   to   continue !!!          ');
disp('              ');
disp('              ');
pause;

% Charge BATTERY
ibc1=ibn;
for k=1:length(ibc1)
  if ibc1(k)>=0
     ibcn(k)=ibc1(k);
  else
     ibcn(k)=0;
  end
end
% plot current (charge battery)
ib_ch=ibcn;
kk=1:length(ib_ch);
plot(kk,ib_ch);
title('Ibattery (charge)');
grid;

%% Calculate area (charge battery)
%% Order 4 approximate
% use (h only),(h1)
h=1:length(ib_ch);
```

```
ibch1=ib_ch;        %DATA BASE
coef_4=polyfit(h,ibch1,4);   % SHOW

% battery current plot
subplot(311);
plot(mmm,ibn);
title('BATTERY CURRENT');
xlabel('time');
ylabel('Currrnt (A)');
grid;
h1=h;
a1=polyval(coef_4,h1);

%//////////
subplot(312);
plot(h,ibch1,'x');
grid;
title('Data Points');
% //////////
%plot function of optimal
subplot(313);
plot(h,ibch1,'x',h1,a1);
grid;
title('line of function polynomial');
% COEFFICIENT of polynomial function
x=h1;
coeff4=coef_4;
co4=coeff4(:,1);
co3=coeff4(:,2);
co2=coeff4(:,3);
co1=coeff4(:,4);
co0=coeff4(:,5);
po=(((co4).*(x.^4))+((co3).*(x.^3))+((co2).*(x.^2))+((co1).*(x.^1))+((co0).*(x.^0)));
% save coefficient for Integrate  function
save in4  co4 co3 co2 co1 co0 -ascii
% Calculate
```

```
x=h1;
int=po;  %show data
pause;

% Battery charge
arc=quad('ic4',1,24);  % (quad8 or quad)
A_hch=arc;

%  PV
subplot(211);
plot(m,ipvn);
grid;
xlabel('time(h)');
ylabel('current(A)');
axis([0 24 0 5]);

% BATTERY
subplot(212);
plot(mmm,ibn);
xlabel('time(h)');
ylabel('current(A)');
axis([0 24 -5  5]);
grid;
pause;

%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
% discharge BATTERY
% discharge current
ibb=-ibn;
di=1:length(ibb);
%discharge
for k=1:length(ibb)
  if ibb(k)>=0
     ibbn(k)=ibb(k);
  else
     ibbn(k)=0;
```

```
  end
end
ibdis1=ibbn;
nnn=1:length(ibdis1);
%!!!!!!!!!!!!!!!!!!!!!!!!
% discharge 0.00-12.00
% RANGE 1
ibbn1=ibbn;
rang1=((length(ibbn1))./2) ;    % 0.00-12.00

for  p=1:12
   ibbn11(p)=ibbn1(p);

end
%!!!!!!!!!!!!
ibbb1=ibbn11;
p1=1:12;
% I_battery  discharge 1-2
ibdisc=ibb;
% discharge rang 1 (0.00-12.00)
for h=1:((length(ibdisc))./2)
   ibdisc1(h)=ibdisc(h);
   if ibdisc(h)>=0
     ib_dis1(h)=ibdisc(h);
   end
end

i12=ib_dis1;
h3=1:length(i12);

% full scale
i11=ibdisc1;
h2=1:length(i11);
%% Calculate Area of curve (discharge battery)(0.00-12.00)
%% Order 3 approximate
%%use (hd),(hd1)
```

```
hd=1:length(ibbb1);
ibd1=ibbb1; % DATA BASE
coef1_3=polyfit(hd,ibd1,3);
hd1=hd;
ad1=polyval(coef1_3,hd1);

subplot(311);
plot(hd,ibd1,'x');
grid;
title('Data Points');
%plot function of optimal
subplot(312);
plot(hd,ibd1,'x',hd1,ad1);
grid;
title('line of function polynomial');

% COEFFICIENT of polynomial function(discharge0.00-12.00)
x=hd1;
coeffd3=coef1_3;
cod3=coeffd3(:,1);
cod2=coeffd3(:,2);
cod1=coeffd3(:,3);
cod0=coeffd3(:,4);
pod=(((cod3).*(x.^3))+((cod2).*(x.^2))+((cod1).*(x.^1))+((cod0).*(x.^0)));
% save coefficient for Integrate  function
save id31  cod3 cod2 cod1 cod0 -ascii
% Calculate
x=hd1;
intd=pod;
subplot(313);
plot(hd,ibd1,'x',x,intd);
title('line function of Order 3');
grid;
pause;
clc;
% Battery discharge(0.00-12.00)
```

```
ard=quad('id3',1,12);  %%%%^^^^^^^^^^(quad8 or quad)
A_hdis1=ard;
%!!!!!!!!!!!!!!!!!!!!!!!!
% discharge 12.00-24.00
% RANGE 2
ibbn2=ibbn;
rang2=((length(ibbn2))./2) ;    % 12.00-24.00
for  p=12:24
  ibbn12(p)=ibbn2(p);

end

ibbb2=ibbn12;
p1=1:24;

% I_battery  discharge 1-2
ibdisc=ibb;
% discharge rang 1 (12.00-24.00)
for h=1:((length(ibdisc))./2)
  ibdisc1(h)=ibdisc(h);
  if ibdisc(h)>=0
    ib_dis1(h)=ibdisc(h);
  end
end

i22=ib_dis1;
h3=1:length(i22);

i21=ibdisc1;
h2=1:length(i21);

%% Calculate Area of curve (discharge battery)(12.00-24.00)
%% Order 3 approximate
%% (hdd), (hdd1) change 1-2
hdd=1:length(ibbb2);
ibd2=ibbb2; %DATA BASE
```

```
coef2_3=polyfit(hdd,ibd2,3);
hdd1=hdd;
ad2=polyval(coef2_3,hdd1);

% COEFFICIENT of polynomial function(discharge12.00-24.00)
x=hdd1;
coeffdd3=coef2_3;
codd3=coeffdd3(:,1);
codd2=coeffdd3(:,2);
codd1=coeffdd3(:,3);
codd0=coeffdd3(:,4);
podd=(((codd3).*(x.^3))+((codd2).*(x.^2))+((codd1).*(x.^1))+((codd0).*(x.^0)));
% save coefficient for Integrate  function
save id32  codd3 codd2 codd1 codd0 -ascii
% Calculate
x=hdd1;
intdd=podd;
% Battery discharge(12.00-24.00)
ardd=quad('idd3',1,24);  % (quad8 or quad)
A_hdis2=ardd;

% NET value
A_hdis=((A_hdis1)+(A_hdis2));
% ANALYSIS
A_hdiff=((A_hdis)-(A_hch));

%%%% SHOW DATA
clc;
disp('              ');
disp('              ');
disp('              ');
disp('              ');
disp('              ');
disp('              ');
disp('              ');
disp('              ');
```

```
disp('            Please Enter to Result show  !!!  ');
disp('             ');
disp('             ');
disp('             ');
pause;
if   A_hch>=A_hdis


  clc;
  A_hbatt=((A_hch)-(A_hdis)); %over
  Aoverper=(((A_hbatt)./(A_hdis)).*(100)); %over percent
  disp('                          ');
  disp('                          ');
  disp('                          ');
  disp(' ____________________________________________________________________');
  disp('                           ');
  disp('                Sizing   Match !      ');
  disp('                          ');
  disp('                A_hr  match       ');
  fprintf('    discharge current of BATTERY :     %f      A.hr \n',A_hdis);
  fprintf('    charge current of BATTERY    :     %f      A.hr \n',A_hch);
  fprintf('    charge current spare         :     %f      A.hr \n',A_hbatt);
  fprintf('    current spare(in percent)    :     %f      percent \n',Aoverper);
  disp('                          ');
  disp(' ____________________________________________________________________');
  disp('                          ');


else
  clc;
  A_hdiff=((A_hdis)-(A_hch)); % must increase
  disp('                          ');
  disp('                          ');
  disp('                          ');
  disp(' ____________________________________________________________________________');
  disp('                           ');
  disp('              !!!  DATA  mismatch  !!!     ');
  disp('                          ');
```

```
  disp('                Current(A_hr)  mismatch        ');
  disp('                               ');
  disp('    !!! discharge current more than charge current of BATTERY  !!!');
  fprintf('    discharge current of BATTERY          :    %f      A.hr \n',A_hdis);
  fprintf('    charge current of BATTERY             :    %f      A.hr \n',A_hch);
  fprintf('    charge current required more than     :    %f      A.hr \n',A_hdis);
  fprintf('    Additional charge current of BATTERY  :    %f      A.hr \n',A_hdiff);
  disp('                               ');
  disp(' ____________________________________________________________________________');
  disp('                               ');

end
clear;
```

จบ