```
function varargout = Programa1(varargin)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%FERRAMENTA COMPUTACIONAL DE MODELAÇÃO DO DESEMPENHO DE PAINEIS
%FOTOVOLTAICOS / TEMPERATURA E IRRADIÂNCIA ESPECÍFICAS
%
%AUTOR: ROMEU SANTOS & PROF. PEDRO DINIS
%CURSO: ENGENHARIA ELECTROMECÂNICA
%
%UNIVERSIDADE DA BEIRA INTERIOR
%DATA: 19/09/2011
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% PROGRAMA1 M-file for Programa1.fig
%      PROGRAMA1, by itself, creates a new PROGRAMA1 or raises the
existing
%      singleton*.
%
%      H = PROGRAMA1 returns the handle to a new PROGRAMA1 or the
handle to
%      the existing singleton*.
%
%      PROGRAMA1('CALLBACK',hObject,eventData,handles,...) calls the
local
%      function named CALLBACK in PROGRAMA1.M with the given input
arguments.
%
%      PROGRAMA1('Property','Value',...) creates a new PROGRAMA1 or
raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before Programa1_OpeningFcn gets called.
An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to Programa1_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Programa1

% Last Modified by GUIDE v2.5 17-Sep-2011 20:16:34

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Programa1_OpeningFcn, ...
                   'gui_OutputFcn',  @Programa1_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
```

```
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Programa1 is made visible.
function Programa1_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Programa1 (see VARARGIN)

% Choose default command line output for Programa1
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Programa1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Programa1_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in Simular_Pushbutton.
function Simular_Pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to Simular_Pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%PARÂMETROS ESPECÍFICOS (TEMPERATURA E
IRRADIANCIA)%%%%%%%%%%%%%%%%%%%%%%%%

%TEMPERATURA DE FUNCIONAMENTO [K]
temp = get(handles.temp,'string');
temp = str2num(temp);

%IRRADIANCIA [W/m^2]
Irra_in = get(handles.Irra_in,'string');
Irra_in = str2num(Irra_in);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
```

```
%INPUTS: PARAMETROS DO FABRICANTE
STC%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%TENSAO DE CIRCUITO ABERTO NAS CONDIÇOES STC
Voc_stc = get(handles.Voc_in,'string');
Voc_stc = str2num(Voc_stc);

%CORRENTE DE CURTO-CIRCUITO NAS CONDIÇOES STC
Isc_stc = get(handles.Isc_in,'string');
Isc_stc = str2num(Isc_stc);

%TENSAO NO PONTO DE POTENCIA MAXIMA (STC)
Vmp_stc = get(handles.Vmp_in,'string');
Vmp_stc = str2num(Vmp_stc);

%CORRENTE NO PONTO DE POTENCIA MAXIMA (STC)
Imp_stc = get(handles.Imp_in,'string');
Imp_stc = str2num(Imp_stc);

%COEFICIENTE DE TEMPERATURA DE Isc
Ki = get(handles.ki_in,'string');
Ki = str2num(Ki);

%COEFICIENTE DE TEMPERATURA DE Voc
Kv = get(handles.kv_in,'string');
Kv = str2num(Kv);

%NUMERO DE CELULAS
Ns = get(handles.Nc_in,'string');
Ns = str2num(Ns);
Nc=Ns;

%LARGURA DA CELULA
l = get(handles.l_in,'string');
l = str2num(l);

%COMPRIMENTO DA CELULA
c = get(handles.c_in,'string');
c = str2num(c);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
if  temp==0 | Irra_in ==0 | Voc_stc ==0 | Isc_stc==0 | Vmp_stc==0 |
Imp_stc==0 | Ns==0;
errordlg('Introduza pelo menos os valores de Temp., Irra., Voc, Isc,
Vmp, Imp, N.Células','Erro');
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CONSTANTES%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
G_stc = 1;            %IRRADIANCIA DE REFERENCIA
T_stc = 298.15;       %TEMPERATURA DE REFERENCIA
q = 1.60217646e-19;   %CARGA DO ELECTRÃO
k = 1.3806503e-23;    %CONSTANTE DE BOLTZMANN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%OUTROS
FACTORES%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
a1 = 1;               %FACTOR DE IDEALIDADE DO DIODO 1
a2 = 1.2;             %FACTOR DE IDEALIDADE DO DIODO 2
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CONDIÇOES DE
TESTE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T = 298.15;           %TEMPERATURA DO MODULO
G = 1;                %IRRADIANCIA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%INICIALIZAÇAO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
P_MAX_M = 0;          %INICIALIZAÇÃO DA POTÊNCIA MÁXIMA NUMÉRICA
RS_0 = 0.2;           %INICIALIZAÇAO DA RESISTENCIA SERIE
RS = RS_0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%OUTRAS
VARIAVEIS%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DT = T - T_stc;         %DIFERENÇA ENTRE A TEMP. DO MÓDULO E A TEMP.
NOMINAL
PMAX_E = Vmp_stc .* Imp_stc;      %POTENCIA MAXIMA EXPERIMENTAL
(DATASHEET)
Eps_POT = abs(P_MAX_M - PMAX_E); %CONDIÇAO PARA CALCULO DO ERRO
TOL = 0.001;                      %TOLERANCIA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CALCULO DAS TENSOES
TERMICAS%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Vt_1 = ((a1 .* Ns .* k .* T) ./ q);
Vt_2 = ((a2 .* Ns .* k .* T) ./ q);
Vt = ((Ns .* k .* T) ./ q);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CALCULO DA CORRENTE
FOTOVOLTAICA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Ipv_stc = Isc_stc;
Ipv = (Ipv_stc + (Ki .* DT)) .* (G ./ G_stc);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%OUTRAS VARIAVEIS PARA O CALCULO DE
Eg%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T_max = 348;
dt = T_max - T_stc;
Isc_Tmax = Isc_stc + Ki .* dt;
Voc_Tmax = Voc_stc + Kv .* dt;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CALCULO DAS CORRENTES DE SATURAÇAO PARA OS DIODOS 1 E
2%%%%%%%%%%%%%%%%%%%%%%%%
%CALCULO DE
EG%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
I_0_n = ((Isc_stc) ./ (exp((Voc_stc) ./ (Vt)) - 1));

Eg = (- log (((Isc_Tmax ./ I_0_n) .* ((T_stc ./ T_max)^3)) ./ ...
    (exp((q .* Voc_Tmax) ./ (a2 .* Ns .* k .*T_max)) - 1))) .* ...
    ((a2 .* k .* T_stc .* T_max) ./ ((q .* (T_stc - T_max))));

I_0 = (I_0_n) .* ((T_stc ./ T) ^ 3) .* (exp (((q .* Eg) ...
    ./ (k)) .* ((1 ./ T_stc) - (1./T))));

I01 = I_0;
```

```
I02 = I_0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%INICIO DO
CICLO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x = 1;
y = 2;
limit = 1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
while (Eps_POT>TOL) & (limit >= 1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CALCULO DA RESISTÊNCIA SHUNT
(RSH)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x1 = Vmp_stc + Imp_stc .* RS;
x2 = Ipv;
x3 = I01 .* (exp((Vmp_stc + Imp_stc .* RS) ./ (a1 .* Vt)) - 1);
x4 = I02 .* (exp((Vmp_stc + Imp_stc .* RS) ./ (a2 .* Vt)) - 1);
x5 = (PMAX_E ./ Vmp_stc);

RSH = x1 ./ (x2 - x3 - x4 - x5);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CALCULO DO MELHOR A2 (IDEALIDADE DO DIODO
2)%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
a2 = ((Vmp_stc + Imp_stc .* RS - Voc_stc) ./ (Vt .* (log (...
     (Isc_stc - (Vmp_stc ./ RSH) - Imp_stc)) - ...
     (log ((Isc_stc - (Voc_stc ./ RSH)))) ...
     + (Imp_stc ./ Isc_stc - (Voc_stc ./ RSH)))));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CALCULO DE I APLICANDO O
METODO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Va = 0;
Ia = 0;
i = 1;
while Ia>=0;
ID1 = (I01 .* (exp((Va + Ia .* RS )./ (a1 .* Vt_1)) - 1));

ID2 = (I02 .* (exp((Va + Ia .* RS )./ (a2 .* Vt_2)) - 1));

fIa = (Ipv - Ia - ID1 - ID2 - ((Va + Ia .* RS) ./ RSH));

fdIa =(-1 - I01 .* (RS ./ (a1 .* Vt_1)) .* exp ((Va + Ia .* RS)...
    ./ (a1 .* Vt_1)) - I02 .* (RS ./ (a2 .* Vt_2)) .* exp ((Va + Ia
.* RS)...
    ./ (a2 .* Vt_2)) - (RS ./ RSH));

Ia = Ia - ( fIa ./ fdIa);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
V(i) = Va;
I(i) = Ia;
i = i + 1;
Va = Va + 0.05;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CAlCULO DA
POTENCIA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
P=V.*I;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CURVA I-V QUE CARACTERIZA O
MODULO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure (1)
grid on
hold on
title('CURVA I-V - AJUSTAMENTO');
xlabel('V [V]');
ylabel('I [A]');
xlim([0 Voc_stc*1.1]);
ylim([0 Isc_stc*1.1]);
plot(V,I,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%POTENCIA VS
TENSAO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(2)
grid on
hold on
title('CURVA P-V - AJUSTAMENTO');
xlabel('V [V]');
ylabel('P [W]');
xlim([0 Voc_stc*1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(V,P,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
Q = 1;
while P(Q) < P(Q+1);
Q = Q + 1;
end
P_MAX_M = P(Q);
V_MP = V(Q);
I_MP = I(Q);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%EPSILON%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
Eps_POT=abs((P_MAX_M - PMAX_E) ./ PMAX_E) .* 100;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%INCREMENTO DA RESISTENCIA
SERIE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RS = RS + 0.01;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
A(x,1) = RSH;
A(x,2) = RS;
A(x,3) = P_MAX_M;
A(x,4) = V_MP;
A(x,5) = I_MP;
A(x,6) = Eps_POT;
A(x,7) = a2;
A;
B(1) = 10;
B(y) = Eps_POT;

if B(y-1) - B(y) >= 0
    limit = 1;
```

```
else
    limit = 0;
end
y = y + 1;
x = x + 1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CONDIÇOES DE
REPRESENTAÇAO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RSH = A(x-2,1);
RS = A(x-2,2);
Eps_POT = A(x-2,6);
a2 = A(x-2,7);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%FACTOR FORMA & CALCULO DE RENDIMENTO
A = l .* c .* Ns;
FF = (I_MP .* V_MP) ./ (Isc_stc .* Voc_stc);
PMAX_real = Voc_stc .* Isc_stc.* FF;
P_in = A .* 1000;
ren = ((I_MP .* V_MP)./ P_in) .* 100;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
I_0 = I_0*1e8;

disp(sprintf('Dados de Entrada [STC]'));
disp(sprintf('Temp. = %f[K]',temp));
disp(sprintf('Irra. = %f[kW/m^2]',Irra_in));
disp(sprintf('Voc = %f[V]',Voc_stc ));
disp(sprintf('Isc_stc = %f[A]',Isc_stc));
disp(sprintf('Vmp_stc  = %f [V]',Vmp_stc ));
disp(sprintf('Imp_stc  = %f [A]',Imp_stc ));
disp(sprintf('Ki  = %f [A/°C]',Ki ));
disp(sprintf('Kv  = %f',Kv ));
disp(sprintf('Ns = %f',Ns));
disp(sprintf('l  = %f [m]',l ));
disp(sprintf('c  = %f[m]',c ));
disp(sprintf('\n\n'));
disp(sprintf('Dados de Saída [STC]'));
disp(sprintf('Rp = %f[Ohm]',RSH));
disp(sprintf('Rs = %f[Ohm]',RS));
disp(sprintf('Eg = %f [eV]',Eg));
disp(sprintf('a2 = %f',a2));
disp(sprintf('Eps_POT = %f [%]',Eps_POT));
disp(sprintf('I01 = I02 = %f[x10^-8 A]',I_0));
disp(sprintf('Ipv = %f[A]',Ipv));
disp(sprintf('FF (STC) = %f',FF));
disp(sprintf('PMAX_real (STC) = %f[W]',PMAX_real));
disp(sprintf('ren (STC)= %f [%]',ren));
disp(sprintf('\n\n'));

RSH = round(RSH.*100)/100;
RS = round(RS.*100)/100;
Eg = round(Eg.*100)/100;
a2 = round(a2.*100)/100;
Eps_POT = round(Eps_POT.*100)/100;
I_0 = round(I_0.*100)/100;
FF = round(FF.*100)/100;
PMAX_real = round(PMAX_real.*100)/100;
ren = round(ren.*100)/100;
```

```
Ipv = round(Ipv.*100)/100;

set(handles.Rsh_res,'String',RSH);
set(handles.Rs_res,'String',RS);
set(handles.Eg_res,'String',Eg);
set(handles.a2_res,'String',a2);
set(handles.Error_res,'String',Eps_POT);
set(handles.I0_res,'String',I_0);
set(handles.Ipv_res,'String',Ipv);
set(handles.FF_res,'String',FF);
set(handles.PMAX_res,'String',PMAX_real);
set(handles.Rend_res,'String',ren);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T = temp;
G = Irra_in; %IRRADIANCIA
V=0;
I=0;
T_stc = 298.15;
DT = T - T_stc;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CALCULO DAS TENSOES
TERMICAS%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Vt_1 = ((a1 .* Ns .* k .* T) ./ q);
Vt_2 = ((a2 .* Ns .* k .* T) ./ q);
Vt = ((Ns .* k .* T) ./ q);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CALCULO DA CORRENTE FOTOVOLTAICA
Ipv_stc = Isc_stc;
Ipv = (Ipv_stc + (Ki .* DT)) .* (G ./ G_stc);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CALCULO DAS CORRENTES DE SATURAÇAO PARA OS
DIODOS%%%%%%%%%%%%%%%%%%%%%%%%%%%
I_0 = ((Isc_stc + Ki .* DT) ./ (exp((Voc_stc + Kv .* DT) ./ Vt)-1));

I01 = I_0;

I02 = I_0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%INICIO DO
CICLO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x = 1;
y = 2;
limit = 1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%CALCULO DE I APLICANDO O
METODO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Va = 0;
Ia = 0;
i = 1;
while Ia >= 0;
ID1 = (I01 .* (exp((Va + Ia .* RS )./ (a1 .* Vt_1)) - 1));

ID2 = (I02 .* (exp((Va + Ia .* RS )./ (a2 .* Vt_2)) - 1));
```

```
fIa = (Ipv - Ia - ID1 - ID2 - ((Va + Ia .* RS) ./ RSH));

fdIa =(-1 - I01 .* (RS ./ (a1 .* Vt)) .* exp ((Va + Ia .* RS)...
    ./ (a1 .* Vt)) - I02 .* (RS ./ (a2 .* Vt)) .* exp ((Va + Ia .*
RS)...
    ./ (a2 .* Vt)) - (RS ./ RSH));

Ia = Ia - ( fIa ./ fdIa);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
V(i) = Va;
I(i) = Ia;
i = i + 1;
Va = Va + 0.05;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CAlCULO DA
POTENCIA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
P = V .* I;
Q = 1;
while P(Q) < P(Q+1);
Q = Q + 1;
end
P_MAX_M = P(Q);
V_MP = V(Q);
I_MP = I(Q);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% CURVA (I-V)/PARA
GUARDAR%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure (3)
grid on
hold on
title('CURVA I-V [Condições Específicas]');
xlabel('V [V]');
ylabel('I [A]');
xlim([0 Voc_stc*1.4]);
ylim([0 Ipv * 1.1]);
plot(V,I,'LineWidth',2,'Color','k')
saveas(figure (3),'CURVA I-V (STC)') ;
close
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% CURVA (I-V)/PARA APARECER NA
FERRAMENTA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.I_V);
grid on
hold on
%title('CURVA I-V [Condições Específicas]');
xlabel('V [V]');
ylabel('I [A]');
xlim([0 Voc_stc * 1.4]);
ylim([0 Ipv * 1.1]);
plot(V,I,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% CURVA P-V/PARA
GUARDAR%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure (4)
```

```
grid on
hold on
title('CURVA P-V [Condições Específicas]');
xlabel('V [V]');
ylabel('P [W]');
xlim([0 Voc_stc*1.1]);
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(V,P,'LineWidth',2,'Color','k')
saveas(figure (4),'CURVA P-V (STC)') ;
close
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
% CURVA P-V/PARA APARECER NA
FERRAMENTA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.P_V)
grid on
hold on
%title('CURVA P-V [Condições Específicas]');
xlabel('V [V]');
ylabel('P [W]');
xlim([0 Voc_stc*1.1]);
ylim([0 Vmp_stc*Imp_stc*1.5]);
plot(V,P,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
contents = cellstr(get(handles.mono_poli,'String'));
temp1 = contents{get(handles.mono_poli,'Value')};

if strcmp(temp1, 'Monocristalino')
%temp1 = 'csc';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CELULAS MOCRISTALINAS (SEM DEPOSIÇAO DE
POEIRAS)%%%%%%%%%%%%%%%%%%%%%%%%%%
for t = 0:0.05:30
P_max = PMAX_E .* ((-0.2826 .* t + 44.176) ./ (44.176));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%PARA
GRAVAR%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
figure (9);
grid on
hold on
title ('Células Monocristalinas Sem Deposição de Poeiras [Condições
STC]');
xlabel('t [Anos]');
ylabel('P [W] (Monocristalino)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%PARA APARECER NA
FERRAMENTA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.axes3)
grid on
hold on
%title('MONO. SEM PÓ [STC]');
xlabel('t [Anos]');
ylabel('P [W] (Monocristalino)');
```

```
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
end
saveas(figure (9),'figure 9.fig') ;
close figure 9
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CÉLULAS MOCRISTALINAS (COM DEPOSIÇÃO DE
POEIRAS)%%%%%%%%%%%%%%%%%%%%%%%%%
for t = 0:0.05:30
P_max = PMAX_E .* ((-0.4999 .* t + 43.808) ./ (43.808));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%PARA
GRAVAR%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
figure (10);
grid on
hold on
title('Células Monocristalinas Com Deposição de Poeiras [Condições
STC]');
xlabel('t [Anos]');
ylabel('P [W] (Monocristalino)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%PARA APARECER NA
FERRAMENTA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.axes4)
grid on
hold on
%title('MONO. COM PÓ [STC]');
xlabel('t [Anos]');
ylabel('P [W] (Monocristalino)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
end
saveas(figure (10),'figure 10.fig') ;
close figure 10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
elseif strcmp(temp1, 'Policristalino')

    if PMAX_E < 80
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CELULAS POLICRISTALINAS (SEM PO - PARA POTENCIAS INFERIORES A
80W)%%%%%%%%
for t = 0:0.05:30
P_max = PMAX_E .* ((-0.1096 .* t + 31.964) ./ (31.964));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%PARA
GRAVAR%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
figure (9);
grid on
```

```
hold on
title('Células Policristalinas / SEM PÓ / PMAX < 80W [Condições
STC]');
xlabel('t [Anos]');
ylabel('P[W] (Poli. Pmáx < 80)');
xlim([0 Voc_stc*1.5])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
%PARA APARECER NA
FERRAMENTA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.axes3)
grid on
hold on
%title('POLI. SEM PÓ PMAX < 80W [STC]');
xlabel('t [Anos]');
ylabel('P[W] (Poli. Pmáx < 80)');
xlim([0 Voc_stc*1.5])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
end
saveas(figure (9),'figure 9.fig');
close figure 9
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CÉLULAS POLICRISTALINAS (COM PO - PARA POTENCIAS INFERIORES A
80W)%%%%%%%%
for t = 0:0.05:30
P_max = PMAX_E .* ((-0.2838 .* t + 31.774) ./ (31.774));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%PARA
GRAVAR%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
figure (10);
grid on
hold on
title('Células Policristalinas / COM PÓ / PMAX < 80W [Condições
STC]');
xlabel('t [Anos]');
ylabel('P [W] Com Pó (Poli. Pmáx < 80) [STC]');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%PARA APARECER NA
FERRAMENTA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.axes4)
grid on
hold on
%title('POLI. COM PÓ PMÁx < 80W [STC]');
xlabel('t [Anos]');
ylabel('P [W] (Poli. Pmáx < 80)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
end
saveas(figure (10),'figure 10.fig');
close figure 10
```

```
    else
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CELULAS POLICRISTALINAS (SEM PO - PARA POTENCIAS SUPERIORES A
80W)%%%%%%%%
for t = 0:0.05:30
P_max = PMAX_E .* ((-2.6173 .* t + 111.03) ./ (111.03));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%PARA
GRAVAR%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
figure (9);
grid on
hold on
title('Células Policristalinas / SEM PÓ / PMAX > 80W [Condições
STC]');
xlabel('t [Anos]');
ylabel('P [W] (Poli. Pmáx > 80)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%PARA APARECER NA
FERRAMENTA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.axes3)
grid on
hold on
%title('POLI. PMÁX > 80W');
xlabel('t [Anos]');
ylabel('P [W] (Poli. Pmáx > 80)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
end
saveas(figure (9),'figure 9.fig');
close figure 9
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%CELULAS POLICRISTALINAS (COM PO - PARA POTENCIAS SUPERIORES A
80W)%%%%%%%%
for t = 0:0.05:30
P_max = PMAX_E .* ((-2.839 .* t + 108.69) ./ (108.69));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
%PARA
GRAVAR%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
figure (10)
grid on
hold on
title('Células Policristalinas / COM PÓ / PMAX > 80W [Condições
STC]');
xlabel('t [Anos]');
ylabel('P [W] (Poli. Pmáx > 80)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%
```

```
%PARA APARECER NA
FERRAMENTA%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
axes(handles.axes4)
grid on
hold on
xlabel('t [Anos]');
ylabel('P [W] (Poli. Pmáx > 80)');
xlim([0 30 .* 1.1])
ylim([0 Vmp_stc*Imp_stc*1.1]);
plot(t,P_max,'LineWidth',2,'Color','k')
end
saveas(figure (10),'figure 10.fig');
close figure 10
    end
end
guidata(hObject, handles);
% --- Executes on button press in Limpar_pushbutton.
function Limpar_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to Limpar_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.temp,'String','0')
set(handles.Irra_in,'String','0')
set(handles.Voc_in,'String','0')
set(handles.Isc_in,'String','0')
set(handles.Vmp_in,'String','0')
set(handles.Imp_in,'String','0')
set(handles.ki_in,'String','0')
set(handles.kv_in,'String','0')
set(handles.Nc_in,'String','0')
set(handles.l_in,'String','0')
set(handles.c_in,'String','0')
set(handles.Rsh_res,'String','0');
set(handles.Rs_res,'String','0');
set(handles.Eg_res,'String','0');
set(handles.a2_res,'String','0');
set(handles.Error_res,'String','0');
set(handles.I0_res,'String','0');
set(handles.Ipv_res,'String','0');
set(handles.FF_res,'String','0');
set(handles.PMAX_res,'String','0');
set(handles.Rend_res,'String','0');
arrayfun(@cla,findall(0,'type','axes'))

% --- Executes on button press in Gerar_pushbutton.
function Gerar_pushbutton_Callback(hObject, eventdata, handles)
% hObject    handle to Gerar_pushbutton (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in Curva_1.
function Curva_1_Callback(hObject, eventdata, handles)
% hObject    handle to Curva_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
CURVA_1 = 'CURVA I-V (STC).fig';
open(CURVA_1);

% --- Executes on button press in Curva_2.
function Curva_2_Callback(hObject, eventdata, handles)
% hObject    handle to Curva_2 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
CURVA_2 = 'CURVA P-V (STC).fig';
open(CURVA_2);

% --- Executes on button press in Curva_3.
function Curva_3_Callback(hObject, eventdata, handles)
% hObject    handle to Curva_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
CURVA_3 = 'figure 9.fig';
open(CURVA_3);

% --- Executes on button press in Curva_4.
function Curva_4_Callback(hObject, eventdata, handles)
% hObject    handle to Curva_4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
CURVA_4 = 'figure 10.fig';
open(CURVA_4);

function Voc_in_Callback(hObject, eventdata, handles)
% hObject    handle to Voc_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Voc_in as text
%        str2double(get(hObject,'String')) returns contents of
Voc_in as a double

% --- Executes during object creation, after setting all properties.
function Voc_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Voc_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Isc_in_Callback(hObject, eventdata, handles)
% hObject    handle to Isc_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Isc_in as text
%        str2double(get(hObject,'String')) returns contents of
Isc_in as a double

% --- Executes during object creation, after setting all properties.
function Isc_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Isc_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
```

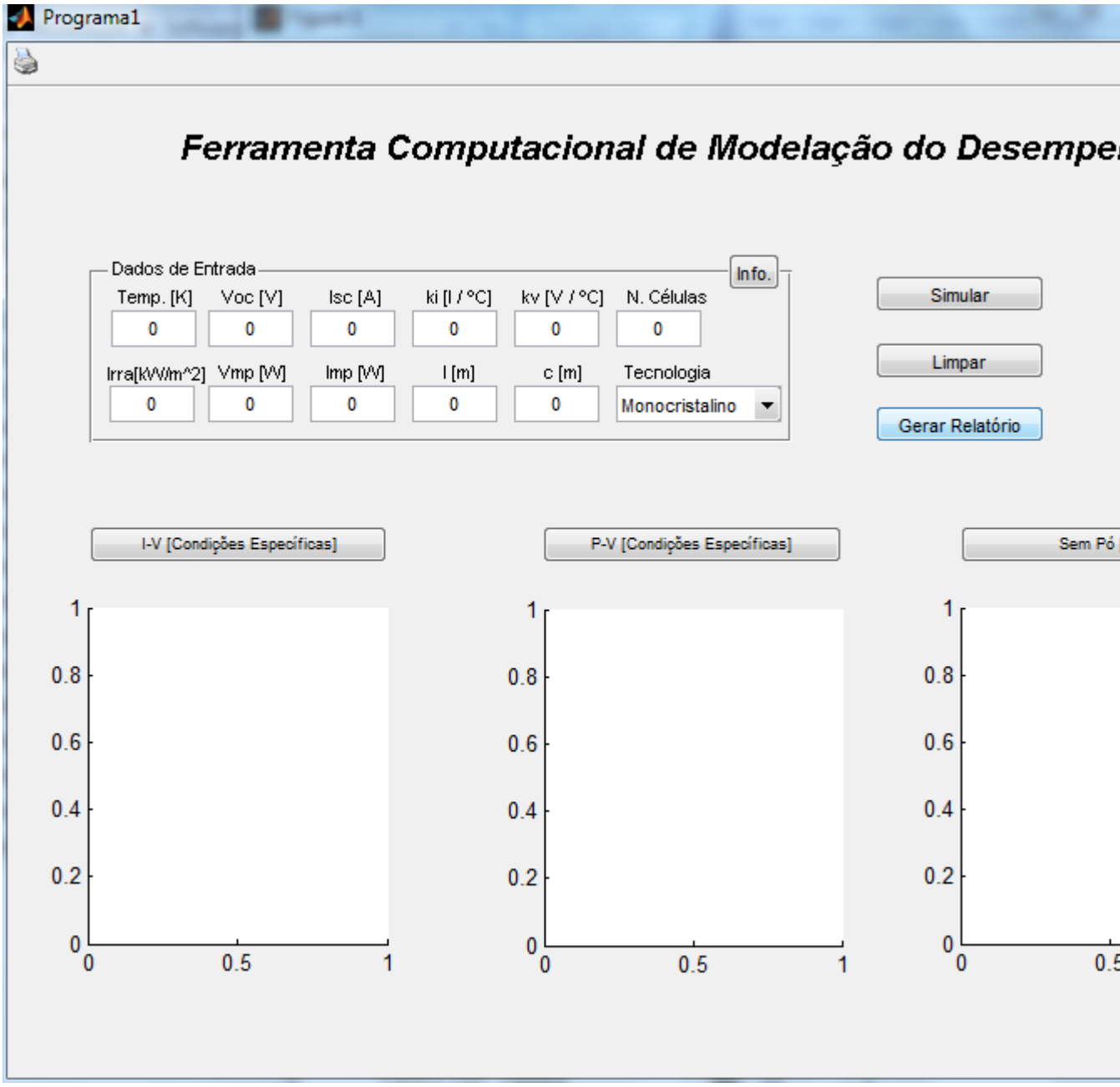
```
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Vmp_in_Callback(hObject, eventdata, handles)
% hObject    handle to Vmp_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Vmp_in as text
%        str2double(get(hObject,'String')) returns contents of
Vmp_in as a double

% --- Executes during object creation, after setting all properties.
function Vmp_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Vmp_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Imp_in_Callback(hObject, eventdata, handles)
% hObject    handle to Imp_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Imp_in as text
%        str2double(get(hObject,'String')) returns contents of
Imp_in as a double

% --- Executes during object creation, after setting all properties.
function Imp_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Imp_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function ki_in_Callback(hObject, eventdata, handles)
% hObject    handle to ki_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ki_in as text
%        str2double(get(hObject,'String')) returns contents of ki_in
as a double
```

```
% --- Executes during object creation, after setting all properties.
function ki_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ki_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function l_in_Callback(hObject, eventdata, handles)
% hObject    handle to l_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of l_in as text
%        str2double(get(hObject,'String')) returns contents of l_in
as a double

% --- Executes during object creation, after setting all properties.
function l_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to l_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function kv_in_Callback(hObject, eventdata, handles)
% hObject    handle to kv_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of kv_in as text
%        str2double(get(hObject,'String')) returns contents of kv_in
as a double

% --- Executes during object creation, after setting all properties.
function kv_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to kv_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function c_in_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to c_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of c_in as text
%        str2double(get(hObject,'String')) returns contents of c_in
as a double

% --- Executes during object creation, after setting all properties.
function c_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to c_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Nc_in_Callback(hObject, eventdata, handles)
% hObject    handle to Nc_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Nc_in as text
%        str2double(get(hObject,'String')) returns contents of Nc_in
as a double

% --- Executes during object creation, after setting all properties.
function Nc_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Nc_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in mono_poli.
function mono_poli_Callback(hObject, eventdata, handles)
% hObject    handle to mono_poli (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns mono_poli
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
mono_poli

% --- Executes during object creation, after setting all properties.
function mono_poli_CreateFcn(hObject, eventdata, handles)
% hObject    handle to mono_poli (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns
called

% Hint: popupmenu controls usually have a white background on
Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function temp_Callback(hObject, eventdata, handles)
% hObject    handle to temp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of temp as text
%        str2double(get(hObject,'String')) returns contents of temp
as a double

% --- Executes during object creation, after setting all properties.
function temp_CreateFcn(hObject, eventdata, handles)
% hObject    handle to temp (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Irra_in_Callback(hObject, eventdata, handles)
% hObject    handle to Irra_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Irra_in as text
%        str2double(get(hObject,'String')) returns contents of
Irra_in as a double

% --- Executes during object creation, after setting all properties.
function Irra_in_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Irra_in (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in Menu.
function Menu_Callback(hObject, eventdata, handles)
% hObject    handle to Menu (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
Close
FCMDPF

% --- Executes on button press in Info2.
function Info2_Callback(hObject, eventdata, handles)
% hObject    handle to Info2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
helpdlg('[STC]: Condições Normais de Teste - Temp.= 25 [graus],
Irrad.= 1 [kW/m^2];
Rsh [Ohm]: Resistência de Shunt (Paralela) [STC];
Rs [Ohm]: Resistência Série [STC];
Eg [eV]: Banda de Valência [STC];
a2: Factor de Idealidade do Díodo 2 [STC];
Erro [%]: Erro Relativo no Cálculo da Potência Máxima [STC];
I0 [A]: Corrente de Saturação [STC];
Ipv [A]: Corrente Fotovoltaica [STC];
FF: Factor de Forma [STC];
PMax [W]: Potência Máxima Real [STC];
Rend.[%]: Rendimento do Módulo [STC].','Descrição dos Dados de
Saída');

% --- Executes on button press in Info1.
function Info1_Callback(hObject, eventdata, handles)
% hObject    handle to Info1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
helpdlg('[STC]: Condições Normais de Teste - Temp.= 25 [graus],
Irrad.= 1 [kW/m^2];
Temp. [K]: Temperatura em Kelvin;
Voc [V]: Tensão em Circuito Aberto [STC];
Isc [A]: Corrente em Curto-Circuito [STC];
ki [I/Grau Celsius]: Coeficiente de Temperatura de Isc [STC];
kv [V/Grau Celsius]: Coeficiente de Temperatura de Voc [STC];
N.Células: Número de Células do Módulo;
Irra. [kW/m^2]: Irradiância;
Vmp [W]: Tensão no Ponto de Potência Máxima [STC];
Imp [W]: Corrente no Ponto de Potência Máxima [STC];
l [m]: Largura da Célula Fotovoltaica;
c [m]: Comprimento da Célula Fotovoltaica;
Tecnologia: Seleccionar Monocristalino ou
Policristalino.','Descrição dos Dados de Entrada');
```



Published with MATLAB® 7.9