Table of Contents

Parse excel data

```
data = xlsread('readings.xlsx');

shortCircuit = isnan(data);
a_Isc = data(shortCircuit,:);
Isc = mean(a_Isc(:,2))
data(shortCircuit,:) = [];
```

```
Isc =

    0.3600
```

Define constant parameters

```
Is=1*10^(-8);         %%Reverse saturation current [A]
N=16;                 %%Amount of solar cells

k=1.38*10^(-23);      %%Boltzmann constant [J/K]
q=1.6*10^(-19);       %%charge of an electron [As]
T=300;                %%Temperature [K]
Ur=(k*T)/q;           %%Thermal voltage [V]
```

Calculate m using best fit curve

```
fun = @(m)(Isc - Is.*(exp(data(:,1)./(m.*Ur.*N))-1)-data(:,2));
m = lsqnonlin(fun,1)
```

```
Local minimum found.

Optimization completed because the size of the gradient is less than
the default value of the optimality tolerance.




m =
```

```
1.2710
```

Plot!

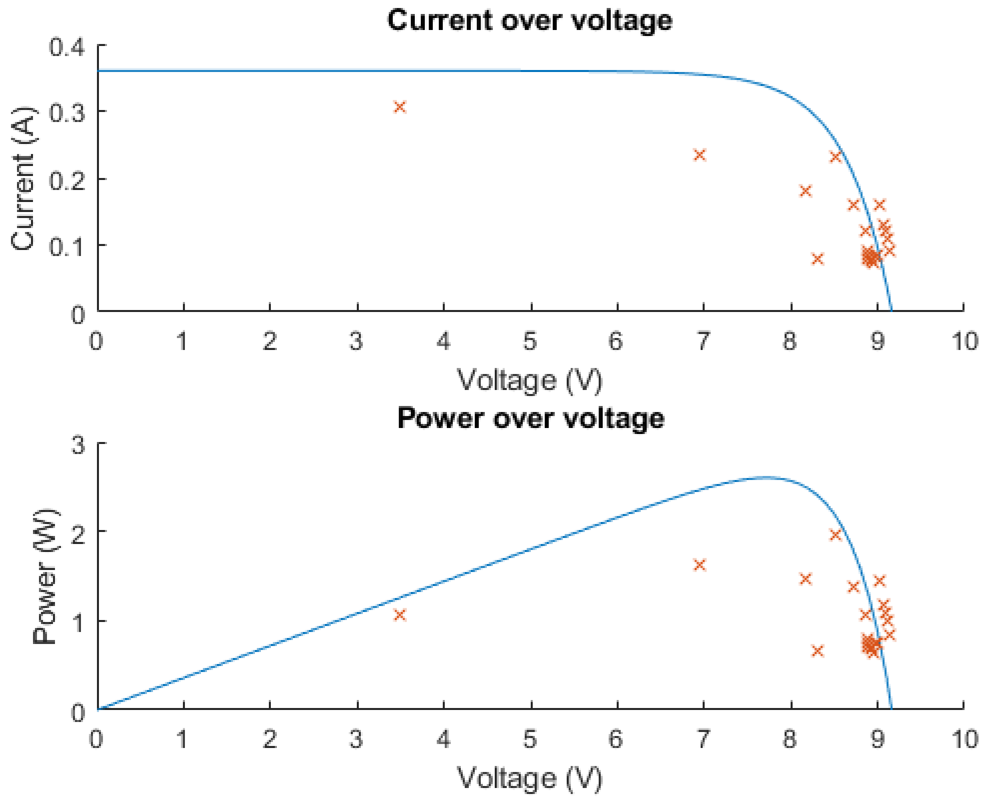
```
sp = SolarPanel(m,Isc);

f1 = figure('Name','Solar panel characteristics','NumberTitle','off');
subplot(2,1,1)
hold on
axis([0 10 0 floor(Isc*10)/10+0.1])
U = 0:0.1:10;
I = sp.current(U);
plot(U,I)
plot(data(:,1),data(:,2),'x')
title('Current over voltage')
xlabel('Voltage (V)');
ylabel('Current (A)');
hold off;

power = data(:,1).*data(:,2);
subplot(2,1,2)

hold on
axis([0 10 0 3])
U = 0:0.1:10;
I = sp.current(U);
plot(U,I.*U)

title('Power over voltage')
xlabel('Voltage (V)')
ylabel('Power (W)')
plot(data(:,1),power, 'x')
hold off;
```



Published with MATLAB® R2017b