Wanjun Gu

A13487962

MAE 119 HW 5,6

HW5

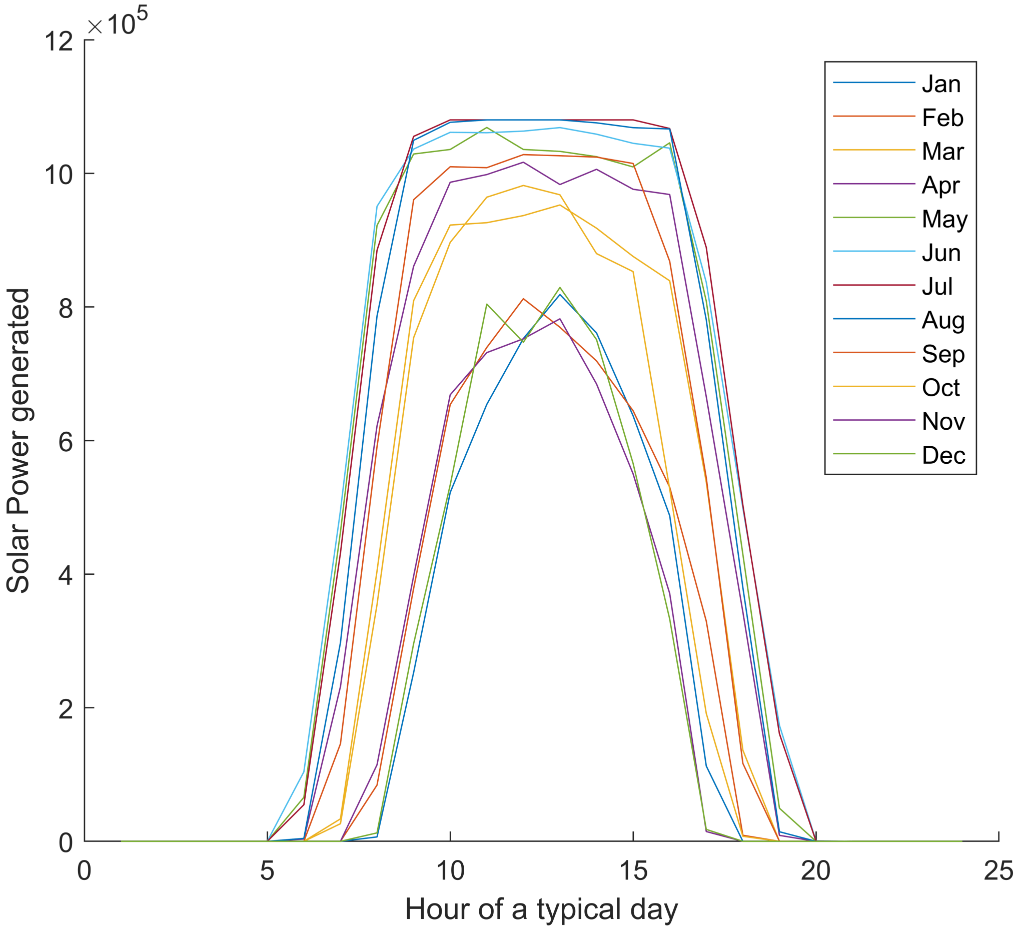
Using the code:

```
clear;
close all;
warning off
addpath(genpath('PVLib'))

power_load = readtable('Konocti_Load.csv');
power_load = table2array(power_load(:,3));

SurfTilt=10;
SurfAz=180;
TMYData=pvl_readtmy3('725905TYA.CSV');
TimeMatlab = TMYData.DateNumber;
Time = pvl_maketimestruct(TimeMatlab,
ones(size(TimeMatlab))*TMYData.SiteTimeZone);
dayofyear = pvl_date2doy(Time.year, Time.month, Time.day);
DNI = TMYData.DNI;
DHI = TMYData.DHI;
GHI = TMYData.GHI;
Location = pvl_makelocationstruct(TMYData.SiteLatitude,TMYData.SiteLongitude,...
    TMYData.SiteElevation);
pressure= TMYData.Pressure*100;
[SunAz, SunEl, ApparentSunEl, SolarTime] = pvl_ephemeris(Time, Location);
SunZen=90-ApparentSunEl;
AM= pvl_relativeairmass(SunZen);
AMa=pvl_absoluteairmass(AM,pressure);

HExtra = pvl_extraradiation(dayofyear);
Ediffsky = pvl_perez(SurfTilt, SurfAz, DHI, DNI, HExtra, SunZen, SunAz, AMa);

ro_g = 0.2;
AOI = pvl_getaoi(SurfTilt, SurfAz, SunZen, SunAz);
Eb=0*AOI;
Eb(AOI<90)=DNI(AOI<90).*cosd(AOI(AOI<90));
GHI(isnan(GHI))=0;
Ediffground=pvl_grounddiffuse(SurfTilt, GHI, ro_g);

POA=Eb + Ediffsky + Ediffground;
Ediff=Ediffsky+Ediffground;

DBfile = 'SandiaModuleDatabase_20120925.xlsx';
Module = pvl_sapmmoduledb(124, DBfile);

Tamb=TMYData.DryBulb(1:8760);
windspeed=TMYData.Wspd(1:8760);
```

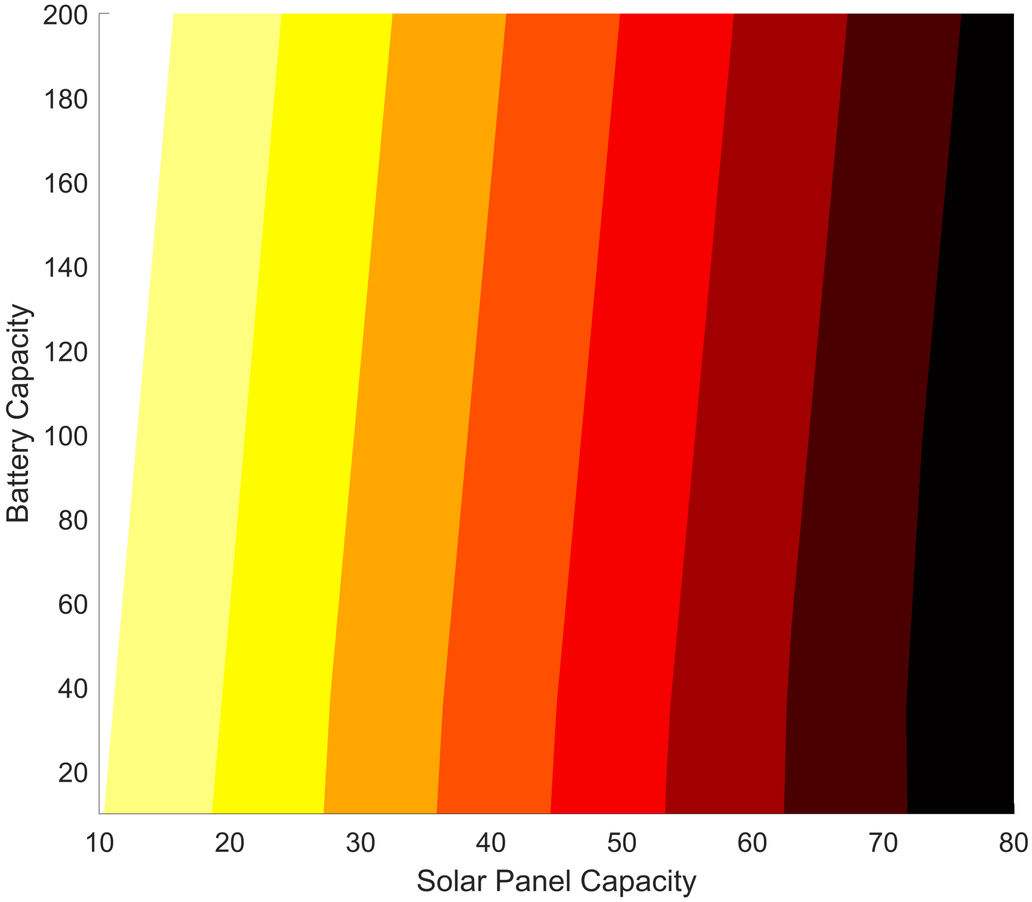
```
a=Module.a_wind;
b=Module.b_wind;
deltaT=Module.delT;

Ee=POA*.98/1000;
Tcell = pvl_sapmcelltemp(Ee, 1000, a, b, windspeed, Tamb, deltaT);
Result = pvl_sapm(Module, Ee, Tcell);

MS=100;
MP=50;

Vdc=MS*Result.Vmp;
Vdc(Vdc<0)=0;
Idc=MP*Result.Imp;
Pdc=(Vdc.*Idc);
load 'SandiaInverterDatabaseSAM2014.1.14.mat';
Inverter = SNLInverterDB(441);

Pac=pvl_snlinverter(Inverter,Vdc,Pdc);
Pac(Pac < 0) = 0;

AvgDayJan = sum(reshape(Pac(1:744),24,31),2)/31;
AvgDayFeb = sum(reshape(Pac(745:1416),24,28),2)/28;
AvgDayMar = sum(reshape(Pac(1417:2160),24,31),2)/31;
AvgDayApr = sum(reshape(Pac(2161:2880),24,30),2)/30;
AvgDayMay = sum(reshape(Pac(2881:3624),24,31),2)/31;
AvgDayJun = sum(reshape(Pac(3625:4344),24,30),2)/30;
AvgDayJul = sum(reshape(Pac(4345:5088),24,31),2)/31;
AvgDayAug = sum(reshape(Pac(5089:5832),24,31),2)/31;
AvgDaySep = sum(reshape(Pac(5833:6552),24,30),2)/30;
AvgDayOct = sum(reshape(Pac(6553:7296),24,31),2)/31;
AvgDayNov = sum(reshape(Pac(7297:8016),24,30),2)/30;
AvgDayDec = sum(reshape(Pac(8017:8760),24,31),2)/31;

fig = figure('units','inch','position',[5,5,6,5]);
hold on
plot(1:24, AvgDayJan)
plot(1:24, AvgDayFeb)
plot(1:24, AvgDayMar)
plot(1:24, AvgDayApr)
plot(1:24, AvgDayMay)
plot(1:24, AvgDayJun)
plot(1:24, AvgDayJul)
plot(1:24, AvgDayAug)
plot(1:24, AvgDaySep)
plot(1:24, AvgDayOct)
plot(1:24, AvgDayNov)
plot(1:24, AvgDayDec)
legend('Jan', 'Feb',    'Mar',  'Apr',  'May',  'Jun',  'Jul',...
    'Aug',  'Sep',  'Oct',  'Nov',  'Dec')
xlabel('Hour of a typical day')
ylabel('Solar Power generated')
hold off
print(fig,'fig1.png','-dpng','-r800');

power_generation =
[AvgDayJan;AvgDayFeb;AvgDayMar;AvgDayApr;AvgDayMay;AvgDayJun;AvgDayJul;AvgDayAug
;...
```

```
    AvgDaySep;AvgDayOct;AvgDayNov;AvgDayDec]*10^-6; %MW

solar_capacity_range = 10:80; %solar capacity, MW
battery_capacity_range = 10:200; %battery capacity, MWh
life_cycle_cost = zeros(200,80);

for sol = solar_capacity_range
    for bat = battery_capacity_range

        total_power_generation = power_generation*sol;

        Battery = zeros(288,1);
        Battery(1) = bat;
        PowerOut = zeros(288,1);

        for d = 1:length(power_load)
               if total_power_generation(d) < power_load(d)
                    Battery(d+1) = Battery(d) - (power_load(d)-
total_power_generation(d)); %charge battery
                    if Battery(d+1) < 0
                        PowerOut(d) = abs(Battery(d+1));
                        Battery(d+1) = 0;
                    end

                elseif total_power_generation(d) >= power_load(d)
                    Battery(d+1) = Battery(d) + (total_power_generation(d) -
power_load(d));
                    if Battery(d+1) > bat
                       Battery(d+1) = bat;
                    end
               end
        end

        power_outage_cost = zeros(21,1);
        investment = zeros(21,1);
        OnM = zeros(21,1);
        incentives = zeros(21,1);

        power_outage_cost(1) = sum(PowerOut)*1000*14;
        investment(1) = 1800*sol*1000 + 600*bat*1000;
        investment(11) = 600*bat*1000;
        OnM(2) = 9*sol*1000;
        incentives(1) = -1800*sol*1000*0.3;

        for m = 3:21
            OnM(m) = OnM(2)*(1.05)^(m-2);
        end

        for p = 2:21
            power_outage_cost(p) = power_outage_cost(1)*(1.05)^(p-2);
        end

        total_cost = investment + power_outage_cost + OnM + incentives;

        discount = zeros(21,1);
        for n = 1:21
            discount(n) = total_cost(n)/(1.06)^(n-1);
        end
        life_cycle_cost(bat,sol) = sum(discount);
    end
end

life_cycle_cost = life_cycle_cost(min(solar_capacity_range):end, ...
min(battery_capacity_range):end);

figure('units','inch','position',[5,5,6,5]);
surf(solar_capacity_range, battery_capacity_range, life_cycle_cost)
% print(fig,'fig2.png','-dpng','-r800');

fig = figure('units','inch','position',[5,5,6,5]);
hold on
contourf(solar_capacity_range,battery_capacity_range,...
life_cycle_cost,'LineStyle','none')
colormap(hot)
xlabel('Solar Panel Capacity')
ylabel('Battery Capacity')
hold off
print(fig,'fig2.png','-dpng','-r800');

minimum = min(min(life_cycle_cost));
[b,s] = find(life_cycle_cost == minimum);

best_battery = battery_capacity_range(b);
best_solar = solar_capacity_range(s);

disp('Best Battery Capacity:  ')
disp(best_battery);
disp('Best Solar Panel Capacity:  ')
disp(best_solar);
```

We are able to determine that the power generation of typital days in different months look like:



And for the life cycle analysis, we are able to know that the cost vary with respect to the size of the solar panel and battery:



In order to reach the maximum efficiency, the ideal battery capacity is 35 MWh and the best solar panel capacity is 80 MW. This way, the 20-year life cycle cost is around $9.579932e+09

In comparison, the ad-hoc sizing suggests a 34 MWh solar panel and a 85 MWh battery, which yields a life cycle cost of $127,508,920. There are differences between the ad-hoc sizing and this calculation because for the ad-hoc analysis, we only considered October, which may on average have an AOI slightly higher than annual average.

HW6 Q1

Shown as written work

HW6 Q2

Using the code:

```
clear all;
clc;

% at z=20m
u=8;
sd=3.7;

syms c k

ksim=vpasolve(sd^2==(u^2)*((gamma(1+2/k)/gamma(1+1/k)^2)-1),k);
ksim=double(ksim);
numc=vpasolve(u==c*gamma(1+1/ksim),c);
numc=double(numc);

k=ksim;
c=numc;
%Check
[m,v]=wblstat(c,k);

disp('At z = 20 m, the values of k and c are: ')
disp(k)
disp(c)

% at z=100m
u=6.357;
sd=2.94;

syms c k

ksim=vpasolve(sd^2==(u^2)*((gamma(1+2/k)/gamma(1+1/k)^2)-1),k);
ksim=double(ksim);
numc=vpasolve(u==c*gamma(1+1/ksim),c);
numc=double(numc);

k=ksim;
c=numc;
```

```
%Check
[m,v]=wblstat(c,k);

disp('At z = 100 m, the values of k and c are: ')
disp(k)
disp(c)
```

Output:

```
At z = 20 m, the values of k and c are:
    2.2921

    9.0306

At z = 100 m, the values of k and c are:
    2.2922

    7.1759
```

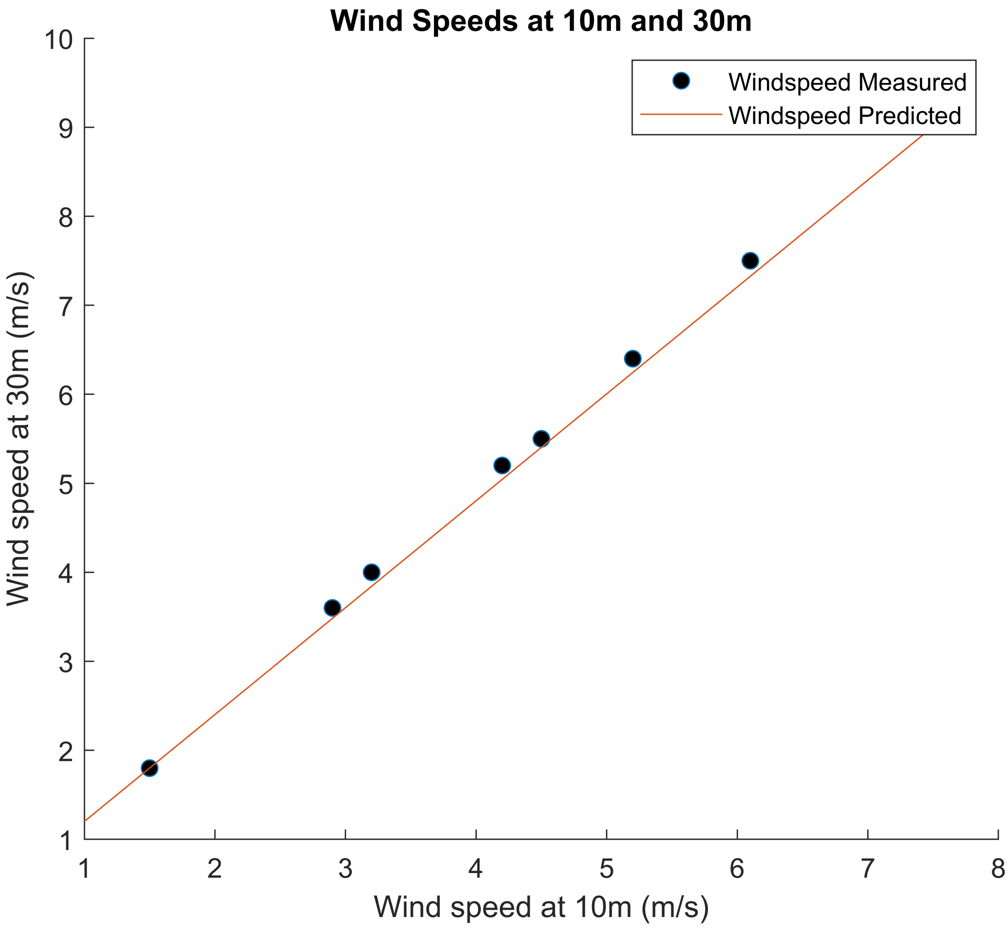
The rest of the calculation is shown on paper.

HW6 Q3

Using the code:

```
clc;
clear;
u2=[3.2 4.5 2.9 5.2 6.1 1.5 4.2]; %z=10m
u1=[4 5.5 3.6 6.4 7.5 1.8 5.2]; %z=30m

%check alpha
a=.16669;
x=1:.5:max(u1);
y = x.*3^a;

%compare results from table and alpha
fig = figure('units','inch','position',[5,5,6,5]);
hold on
scatter(u2,u1, 'MarkerFaceColor', 'k');
plot(x,y);
title('Wind Speeds at 10m and 30m')
xlabel('Wind speed at 10m (m/s)')
ylabel('Wind speed at 30m (m/s)')
legend('Windspeed Measured', 'Windspeed Predicted')
hold off
print(fig,'fig3.png','-dpng','-r800');
```

We are able to plot measured windspeed and predicted windspeed:



The rest of the calculation is shown on paper.