Charles Lin
October 29, 2018

ME 146 Project 2: Solar Hybrid Power

Table of Contents

Introduction

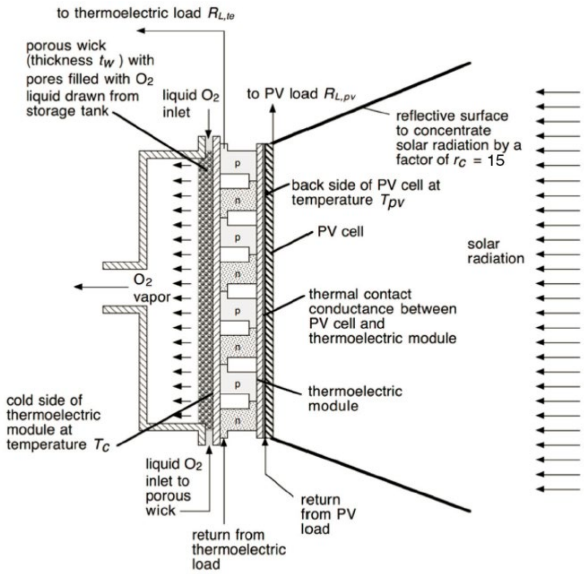


Figure 1: Hybrid Power System Schematic

The purpose of this project is to analyze the performance of a hybrid power system as shown in Figure 1. This power system consists of a concentrating photovoltaic (PV) system that also captures waste heat from the PV cell to produce power. The back of the PV cell is bonded to the hot side of the thermoelectric module.

PV Cell Performance

The PV cell is governed by the following equations:

Equation 1

$$\frac{P"}{\phi} = \frac{\pi^4 k_B T_{source}}{2.404(15)}; \text{ where } P" = r_c I_D$$

Equation 2

$$\frac{\phi_g}{\phi} = 0.416 \int_X^\infty \frac{x^2}{e^x - 1} dx; \text{where } X = \frac{q_e V_g}{k_B T_{source}}$$

Equation 3

$$V_L = \frac{k_B T_{pv}}{q_e} \ln\left[\frac{I_v - I_L}{I_0} + 1\right] - I_L R_s; \text{where } I_v = \phi_g q_e A_{pv} \text{ and } A_{pv} = L_y L_z$$

The relevant variables, constants, and assumptions are set:

- Boltzmann constant: $k_B = 1.3807 * 10^{-23}$ J/K
- Fundamental charge of an electron: $q_e = 1.602 * \ 10^{-19}$ C
- Concentration ratio: $r_c = 15$
- Unconcentrated direct solar intensity: $I_D = 1080$ W/m^2
- Temperature of source of radiation (the sun): $T_{source} = 6000$ K
- $V_g = 1.1$ V
- Y-length of PV cell: $L_y = 10 \text{ cm} = 0.1\text{m}$
- Z-length of PV cell: $L_z = 10 \text{ cm} = 0.1\text{m}$
- Area of PV cell: A_{pv}

These equations are implemented in Tasks 1 and 2.

Thermoelectric Module Performance

The thermoelectric module is governed by the following equations:

Equation 4: Relationship between TH and Tpv

$$T_H = T_{pv} - \dot{Q}_{pv}/\Lambda_{cont}$$

Equation 5: Heat Transfer to Liquid Interface

$$\frac{\dot{Q}_C}{A_{pv}} = \left[\frac{k_{w,eff}}{t_w} + U_{ev}\right](T_C - T_{sat})$$

Equation 6: Single Pair Electrical Resistance & Heat Conductance

$$R = \frac{\iota_A \rho_A}{A_A} + \frac{\iota_B \rho_B}{A_B}, \qquad \Lambda = \frac{A_A \lambda_A}{\iota_A} + \frac{A_B \lambda_B}{\iota_B}$$

Equation 7: Total Battery Electrical Resistance & Thermal Conductance

$$R_{batt} = nR, \qquad \Lambda_{batt} = n\Lambda$$

Equation 8: Voltage across the Load

$$V_L = n\alpha(T_H - T_C) - R_{batt} I_L$$

Equation 9: Heat Rate from TH Source

$$P_H = \dot{Q}_{pv} = \dot{Q}_H = n\alpha I T_H + \Lambda_{batt}(T_H - T_C) - \frac{1}{2} I^2 R_{batt}$$

Equation 10: Power Output

$$\dot{W} = \frac{n^2 \alpha^2 (T_H - T_C)^2 R_{L,te}}{\left(R_{L,te} + R_{batt}\right)^2}$$

Equation 11: Efficiency

$$\eta = \frac{\dot{W}}{\dot{Q}_H} = \left[\frac{T_H - T_C}{T_H}\right]\left[\frac{(1+m)^2}{m}(ZT_H)^{-1} + 1 + \frac{1}{2m}\left(1 + \frac{T_C}{T_H}\right)\right]^{-1};\ \text{where } m = \frac{R_{L,te}}{nR}$$

Equation 12: Figure of Merit

$$Z = \frac{\alpha^2}{R\Lambda}$$

Equation 13: Minimum RΛ Value

$$(R\Lambda)_{\min} = \left[(\lambda_A\rho_A)^{1/2} + (\lambda_B\rho_B)^{1/2}\right]^2$$

Equation 14: "m" Value that Maximizes Efficiency

$$m_{\max\_eff} = \sqrt{1 + 0.5(T_H + T_C)Z}$$

The relevant variables, constants, and assumptions are set as necessary:

- Thermal conductance between PV cell and thermoelectric module: $\Lambda_{cont} = 300\ \mathrm{K}$
- Average Seebeck coefficient over temperature range of interest: $\alpha = 0.0017\ \mathrm{V/K}$
- Electric resistivity of arm A: $\rho_A = 0.0020\ \Omega\ \mathrm{cm}$
- Electric resistivity of arm B: $\rho_B = 0.0030\ \Omega\ \mathrm{cm}$
- Thermal conductivity of arm A: $\lambda_A = 0.032\ \mathrm{W/(cm\ K)}$
- Thermal conductivity of arm B: $\lambda_B = 0.021\ \mathrm{W/(cm\ K)}$
- Thermal conductivity of the nickel wick: $k_{w,eff}$
- Wick thickness: t_w
- Conductance: U_{ev}
- Saturation temperature of liquid oxygen at atmospheric pressure: T_{sat}
- Temperature of wall in contact with wick: T_C
- Temperature of heat input: T_H
- Number of thermoelectric material pairs in battery: n

Design Task Documentation

Task 1

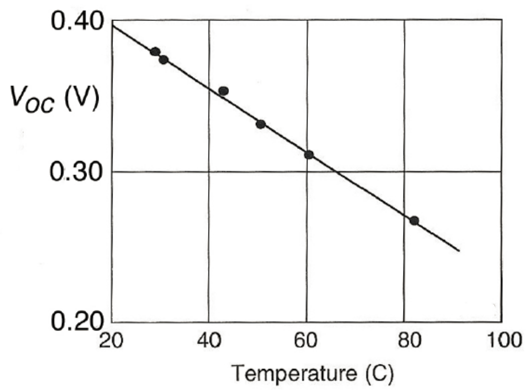


Figure 2: Observed Open-circuit Voltage over Temperature

Task Description

In Task 1, the aim is to use data points from Figure 2 to determine a linear curve-fit to open circuit voltage V_{OC} such that $V_{OC} = \hat{A}T_{pv} + \hat{B}, T_{pv}$ in K.

Assumptions/Idealizations

- Hand measurements of data points is sufficient as the points are not given to us.

Algorithm Summary

1. Manually interpret the figure for data. Perform linear regression to get $\hat{A} = -0.0021, \hat{B} = 1.0183$ such that (1): $V_{OC} = -0.0021T_{pv} + 1.0183$.
2. Using Eq. [3], set $I_L = 0$ to get (2): $V_{OC} = \frac{k_B T_{pv}}{q_e} \ln\left[\frac{I_v}{I_0} + 1\right]$.
3. Equate (1) and (2) and solve for the ratio of $\frac{I_0}{I_v}$ as a function of temperature.

Results

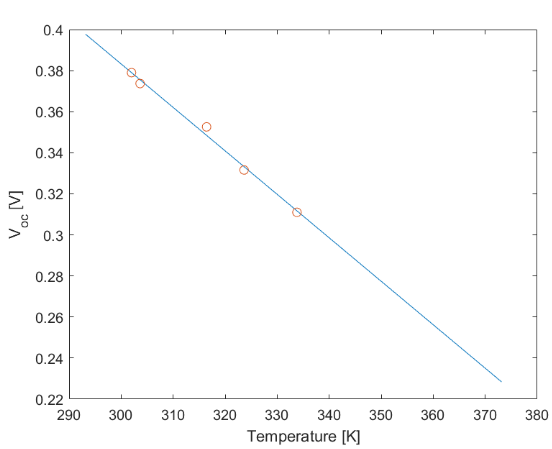


Figure 3: Linear Fit

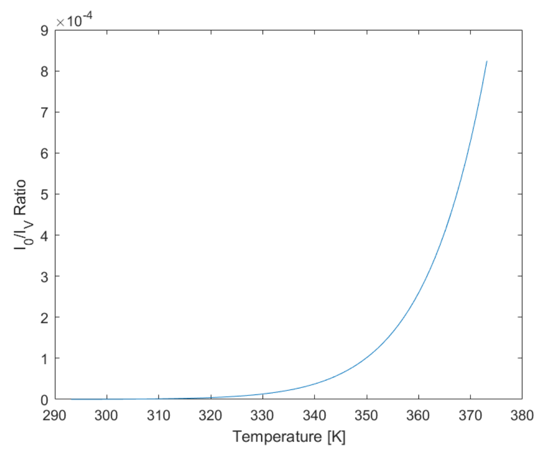


Figure 4: I0/Iv Ratio over Temperature

The resulting curve of linear fit for the Figure 2 data points is: $V_{OC} = -0.0021T_{pv} + 1.0183$. The curve is plotted along with the data points in Figure 3. Meanwhile, Figure 4 displays the I0/IV ratio along a temperature range of 20^{o}C to 80^{o}C.

Task 2

Task Description

Compute PV cell power (as a function of $R_{L,pv}$ and parameters in Equations [1] to [3]) and efficiency.

Assumptions/Idealization

- Efficiency is given by: $\eta = V_L I_L / (P" A_{pv})$
- Operating point is V_L and I_L that satisfies both equation [3] and Ohm's Law $V_L = I_L R_{L,pv}$
- Y-length of PV cell: $L_y = 10 \text{ cm} = 0.1\text{m}$
- Z-length of PV cell: $L_z = 10 \text{ cm} = 0.1\text{m}$
- Area of PV cell: $A_{pv} = 0.01 \text{ m}^2$
- $R_{L,pv} = 0.0070\ \Omega$
- $R_s = 0$
- $V_g = 1.1 \text{ V}$
- $T_{pv} = 20, 80\ ^{\text{o}}\text{C}$
- Power is given by: $P = V_L I_L$

Algorithm Summary

1. Implement Task 1 code to get ratio of $\frac{I_0}{I_v}$
2. Initialize constants
3. Solve [1] for ϕ
4. Solve [2] for ϕ_g and find I_v $(I_v = \phi_g q_e A_{pv})$
5. Set V_L of [3] equal to $I_L R_{L,pv}$ such that $\frac{k_B T_{pv}}{q_e} \ln\left[\frac{I_v - I_L}{I_0} + 1\right] - I_L R_s = I_L R_{L,pv}$. Solve for I_L.
6. Using I_L, solve for V_L. Ohm's Law or Equation [3] can be used; I used Equation [3]. Having both V_L and I_L means that the operating point at both T_{pv} values can be found.
7. Calculate power for both operating points using $P = V_L I_L$.
8. Calculate efficiency for both operating points using $\eta = V_L I_L / (P" A_{pv})$.

Results

At 20°C, the power output of the PV cell is 18.3461 W, while the efficiency is 0.11325. The operating point is at a current of 51.1945 A and voltage of 0.35836 V. Meanwhile, at 80°C, the power output is 8.7168 W and the efficiency is 0.053807. The operating point is at a current of 35.2881 A and a voltage of 0.24702 V. The operating points are reflected in Figure 5 below.

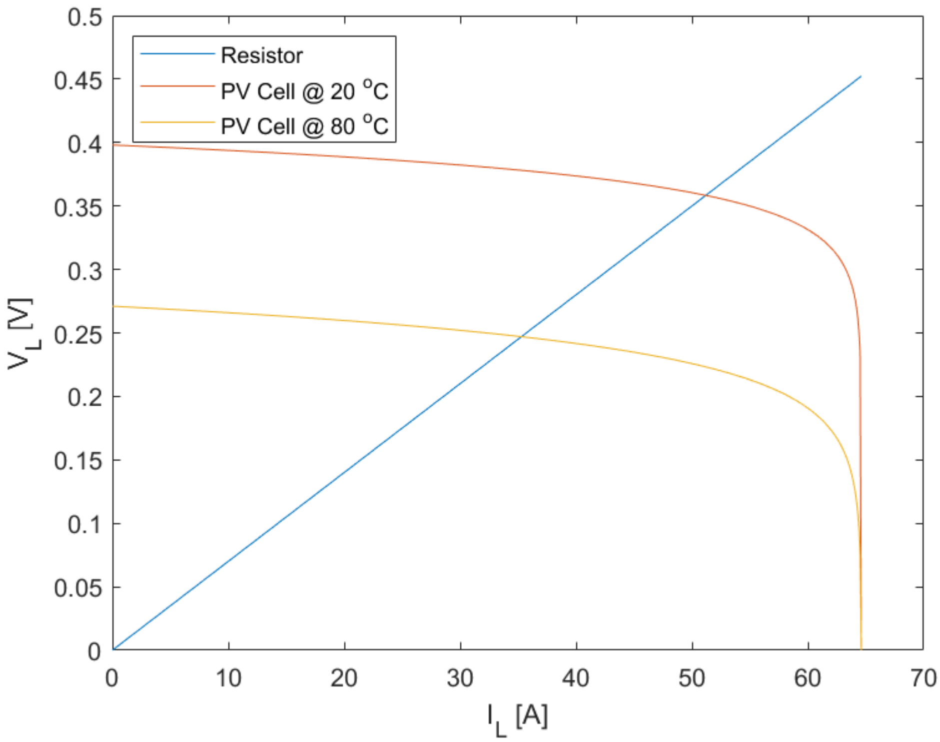


Figure 5: Load Voltage over Load Current

Task 3

Task Description

In Task 3, the objective is to iteratively implement the thermoelectric module governing equations to determine:

- Temperature of wall in contact with wick (cold side) T_C
- Power output of the module $\dot{W}$
- Heat input $\dot{Q}_H$
- Heat transfer out at cold side $\dot{Q}_C$
- Current to the load I_L
- Voltage across the load V_L

Assumptions/Idealizations

- Thermal conductance between PV cell and thermoelectric module: $\Lambda_{cont} = 300\ \mathrm{K}$
- Average Seebeck coefficient over temperature range of interest: $\alpha = 0.0017\ \mathrm{V/K}$
- Electric resistivity of arm A: $\rho_A = 0.0020\ \Omega\ \mathrm{cm}$
- Electric resistivity of arm B: $\rho_B = 0.0030\ \Omega\ \mathrm{cm}$
- Thermal conductivity of arm A: $\lambda_A = 0.032\ \mathrm{W/(cm\ K)}$

- Thermal conductivity of arm B: $\lambda_B = 0.021 \text{ W/(cm K)}$
- Thermal conductivity of the nickel wick: $k_{w,eff} = 6.5 \text{ W/mK}$
- Wick thickness: $t_w = 4.0 \text{ mm} = 0.004 \text{ m}$
- Conductance: $U_{ev} = 25 \text{ W/m}^2\text{K}$
- Saturation temperature of liquid oxygen at atmospheric pressure: $T_{sat} = 95.0 \text{ K}$
- Temperature of heat input: $T_H = 50^{\text{o}}\text{C} = 323 \text{ K}$
- Number of thermoelectric material pairs in battery: $n = 12$
- Resistance across the load: $R_{L,te} = 0.10\ \Omega$
- Area of PV cell: $A_{pv} = 0.01 \text{ m}^2$
- T_C test range from 90 to 120 K
- Error tolerance is 1 e^{-5}

Algorithm Summary

1. Set up the program by initialize constants and compute $(R\Lambda)_{\min}$ and Z using Equations [13] and [12], respectively.
2. Begin iteration to compute the values of T_C and $\dot{Q}_H$. This is accomplished by implementing the bisection root-finding method with a while loop.
    a. T_C is initialized as an array of 3 elements: [90   0   120]. This is the initial guess of the range. Elements 1 and 3 will serve as the upper and lower bound values while element 2 is calculated as the midpoint of 1 and 3. All equations that involve T_C make calculations for all 3 elements using the "." operator. As a result, calculated parameters will have arrays in the form of [X   Y   Z], which correspond to the bound and midpoint values of the initial T_C.
    b. Calculate $m_{\max\_eff}$, R, and Λ. $m_{\max\_eff}$ can be found using Equation [14], while R and Λ can be found by:
        i. $R = \frac{R_{L,te}}{n * m_{\max\_eff}}$
        ii. $\Lambda = (\Lambda R)_{\min}$
    c. Use Equation [5] to solve for $\dot{Q}_C$. This value is noted as $\left(\dot{Q}_C\right)_{boil}$.
    d. Use Equation [11] to solve for efficiency η.
    e. Use Equation [10] to solve for $\dot{W}$. R_{batt} can be found for this equation using Equation [7].
    f. Compute heat input $\dot{Q}_H$ and heat rejection $\dot{Q}_C$, noted as $\left(\dot{Q}_C\right)_{TE}$, using:
        i. $\dot{Q}_H = \dot{W}/\eta$
        ii. $\left(\dot{Q}_C\right)_{TE} = \dot{Q}_H - \dot{W}$
    g. Compute the error $E_{\dot{Q}_C} = \left|\left(\dot{Q}_C\right)_{TE} - \left(\dot{Q}_C\right)_{boil}\right|$.
    h. After performing calculations with initial lower bound, midpoint, and upper bound values of T_C, the errors are calculated for temperatures at those three points $E_{\dot{Q}_C}$ = $[E_{\dot{Q}_C1} \quad E_{\dot{Q}_C2} \quad E_{\dot{Q}_C3}]$. The T_C range is then tightened. If $\left(E_{\dot{Q}_C1} * E_{\dot{Q}_C2}\right) < 0$, the current midpoint becomes the new upper bound as element 3 in the T_C array. Otherwise, the current midpoint becomes the new lower bound as element 1.

i. For all calculated parameters in the loop, the midpoint value (2nd in the array) is always selected as our result value. Therefore, if the 2nd value in the $E_{\dot{Q}_C}$ array is less than the tolerance value of 1 e-5, this means that the T_C range has converged to a point that satisfies Equations [5], [10], and [11]. The resulting 2nd values for T_C, $\dot{W}$, $\dot{Q}_H$, $\dot{Q}_C$, η are saved. Λ_{batt} is also found using Equation [7].

3. The $\dot{Q}_H$ value is used to solve for I (which is I_L) using Equation [9]. The resulting I_L value is used in Equation [8] to compute V_L.

Results

The results of the program are as follows:

- $T_C = 108.9727$
- Power output of the module $\dot{W} = 80.7323\ \mathrm{K}$
- $\dot{Q}_H = 311.2822\ \mathrm{W}$
- $\dot{Q}_C = 230.55\ \mathrm{W}$
- Current to the load $I_L = 28.4134\ \mathrm{A}$
- Voltage across the load $V_L = 2.8413\ \mathrm{V}$

Task 4

Task Description

Task 2 finds power and performance of PV cell, while Task 3 finds power and performance for the thermoelectric (TE) module. Combine the programs from Tasks 2 and 3 in a new program to determine the following:

- PV Cell and TE Module
  - Load voltage V_L
  - Load current I_L
  - Power output P_L, $\dot{W}$ respectively
  - Efficiency η_{PV}, η_{TE}
- Hybrid System (seen in Figure 1)
  - Operating point T_{pv} value
  - Total power output
  - Overall efficiency

Assumptions/Idealizations

- Apply previous conditions from Tasks 2 and 3 with some changes:
  - T_{pv} is now initially guessed as a range from [170 0 230].
  - $T_{sat} = 90.2\ \mathrm{K}$
- $\Lambda_{cont} = 300\ \mathrm{W/K}$
- Waste heat transfer rate: $\dot{Q}_{pv} = P"A_{pv} - V_L I_L$
- T_H is not constant. Instead: $T_H = T_{pv} - \dot{Q}_{pv}/\Lambda_{cont}$.
- Error tolerance is 1 e-5

Algorithm Summary

1. To find the T_{pv} value, the bisection method must be implemented (as seen in Task 3). As a result, T_{pv} will be initialized as an array of 3 values $[T_{pv,1} \quad T_{pv,2} \quad T_{pv,3}]$. Because the Task 3 program is also implemented here, Task 4 performs a nested bisection method (using nested while loops). Other assumed constants also initialized.
2. The Task 2 program is implemented. The while loop for finding T_{pv} begins before the power calculations so that PV cell operating points V_L and I_L are found for all T_{pv} values. To perform element-by-element operations, the "." operator is used extensively.
3. Taking the second value of the T_{pv} array, T_H is computed by $T_H = T_{pv,2} - \dot{Q}_{pv}/\Lambda_{cont}$.
4. The Task 3 program is implemented. Before starting the nested while loop for T_C, T_C is initialized as [90   0   120] such that T_C is reset with every iteration of T_{pv}. The T_C array converges due to the bisection method and the resulting 2nd values for T_C, $\dot{W}$, $\dot{Q}_H$, $\dot{Q}_C$, η_{TE}, and Λ_{batt} are saved.
5. Error $E_{\dot{Q}}$ is computed as $E_{\dot{Q}} = \left|\dot{Q}_{pv} - \dot{Q}_H\right|$. Recall that $E_{\dot{Q}}$ is an array of 3 values such that $E_{\dot{Q}}$ = $[E_{\dot{Q}1} \quad E_{\dot{Q}2} \quad E_{\dot{Q}3}]$. The T_{pv} range is tightened: If $\left(E_{\dot{Q}1} * E_{\dot{Q}2}\right) < 0$, or $\left(E_{\dot{Q}1} * E_{\dot{Q}2}\right) > 0$ while $E_{\dot{Q}1} < 0$, then the midpoint is set as the new upper bound. However, if $\left(E_{\dot{Q}1} * E_{\dot{Q}2}\right) > 0$ while $E_{\dot{Q}1} > 0$, then the midpoint becomes the new lower bound. The midpoint values for $\dot{Q}_{pv}$, η_{PV}, V_L, I_L, T_{pv}, and PV cell power output P_L are saved.
6. After the error $E_{\dot{Q}}$ converges, equations [8] and [9] are solved for V_L and I_L of the TE module to verify the operating point found for the PV cell.
7. Total power P_{tot} and efficiency η_{tot} for the hybrid power system are found:
    a. $P_{tot} = P_L + \dot{W}$
    b. $\eta_{tot} = P_{tot}/(P''A_{pv})$

Results

This program gave the following results:

PV Cell

- Load voltage: $V_L = 0.45092\ \mathrm{V}$
- Load current: $I_L = 64.4175\ \mathrm{A}$
- Power output: $P_L = 29.0473\ \mathrm{W}$
- Efficiency: $\eta_{PV} = 0.1793$

The load voltage and load current for the operating point was validated by the TE module calculations.

TE Module

- Power output: $\dot{W} = 23.1581\ \mathrm{W}$
- Efficiency: $\eta_{TE} = 0.17418$

Hybrid System

- Operating point T_{pv} value: $T_{pv} = 216.6275\ \mathrm{K}$
- Total power output: $P_{tot} = 52.2054\ \mathrm{W}$
- Overall efficiency: $\eta_{tot} = 0.32226$

Figure 6 below demonstrates the convergence of $\dot{Q}_{pv}$ and $\dot{Q}_H$ (error $E_{\dot{Q}}$ goes to 0 as $\dot{Q}$ curves intersect) at $T_{pv} = 216.6275$ K.

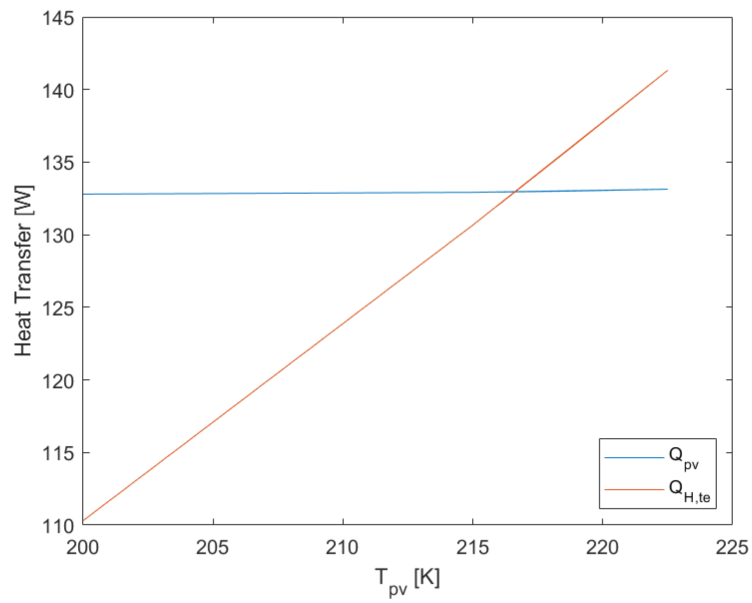


Figure 6: Convergence of Heat Transfer Rates

Task 5(a) & (b)

Task Description

In Task 5, the hybrid power system program built in Task 4 is put to the test.

a. Iterate over a constant $r_c = 15$ and I_D ranging from 720 to $1080\ \mathrm{W/m^2}$.
b. Iterate over a constant $I_D = 1080\ \mathrm{W/m^2}$ and r_c ranging from 10 to 18.

For tests (a) and (b), determine and plot the variation of PV efficiency, TE module efficiency, total system efficiency, and power output.

Assumptions/Idealizations

- Carry over all assumptions from Task 4. However, for parts (a) and (b), adjust r_c and I_D values as described in the Task Description.

Power System Analysis

The program for Task 4 was kept mostly intact, with a few changes:

- The entirety of Task 4, other than initialized r_c and I_D values and linearly-spaced arrays, was put into a for-loop that iterated along the length of whichever variable array was being tested. More plotting sections were added to adequately capture the change in performance due to changes in r_c and I_D.

Results

Part 5(a)

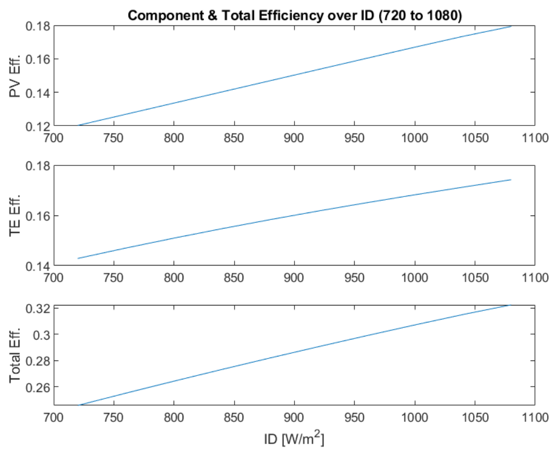


Figure 7: Efficiency Variation over ID

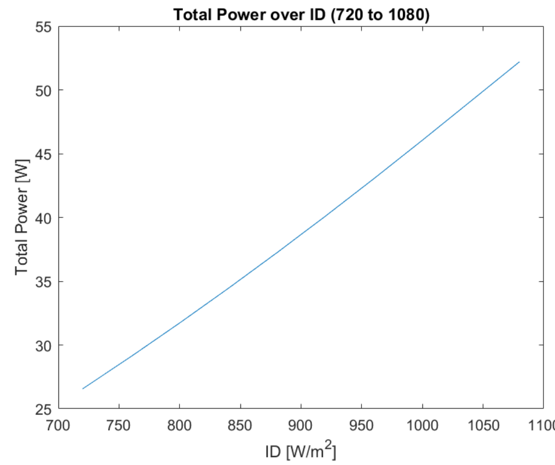


Figure 8: Power Variation over ID

For a variation of I_D from $720 \text{ to } 1080 \text{ W/m}^2$, PV cell efficiency ranges from 0.120 to 0.179 and TE module efficiency ranges from 0.143 to 0.174. Total system efficiency η_{tot} ranges from 0.246 to 0.322, while total power P_{tot} ranges from 26.560 to 52.205 W. All data points are tabulated below in Table 1.

In Figure 7, it is seen that as I_D increases, efficiency of the PV cell, TE module, and overall system increases almost linearly with very slight concavity. As a result, maximum efficiencies for the modules and for the system is found at the maximum I_D value.

This continual upward trend in efficiencies is paralleled by a direct relationship between power output and I_D as shown in Figure 8. I_D must be maximized in order to maximize the power output.

Table 1: Results of ID Variation

ID	720	760	800	840	880	920	960	1000	1040	1080
PV Eff.	0.120243	0.126923	0.133604	0.140284	0.146963	0.15364	0.160307	0.166932	0.173396	0.179304
TE Eff.	0.142863	0.147009	0.150958	0.154724	0.158319	0.161753	0.165036	0.168182	0.171206	0.174183
Total Eff.	0.245928	0.255273	0.264393	0.273303	0.282015	0.290541	0.298887	0.307039	0.314905	0.322256
Total Power	26.56022	29.10118	31.7272	34.43613	37.22597	40.09468	43.03971	46.05588	49.12514	52.20541

Part 5(b)

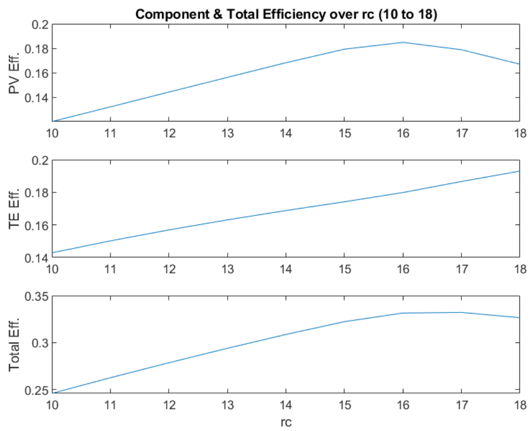


Figure 9: Efficiency Variation over rc

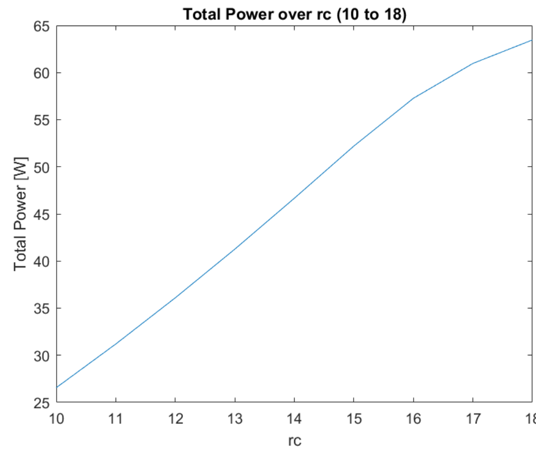


Figure 10: Power Variation over rc

For a variation of r_c from 10 to 18, PV cell efficiency ranges from 0.120 to 0.185 and TE module efficiency ranges from 0.143 to 0.194. Total system efficiency η_{tot} ranges from 0.246 to 0.332, while total power P_{tot} ranges from 26.560 W to 63.680 W. All data points are tabulated in Table 2.

As r_c increases, PV cell efficiency hits a peak at $r_c = 16$. Total system efficiency peaks at $r_c = 17$, only slightly higher than 16. In contrast, TE module efficiency increases nearly linearly, as does the total power output (despite the drop-off towards the higher r_c values).

It is seen that despite the decrease in total system efficiency after peaking, the power output continues to increase. Therefore, if the goal of a hybrid power system is to purely maximize power output and r_c is of no concern, r_c maximized at $r_c = 18$.

Table 2: Results of rc Variation

rc	10	11	12	13	14	15	16	17	18
PV Eff.	0.120243	0.132268	0.144291	0.156309	0.168244	0.179304	0.184837	0.178849	0.165746
TE Eff.	0.142863	0.150184	0.156901	0.163083	0.168797	0.174183	0.179857	0.186625	0.193995
Total Eff.	0.245928	0.262587	0.278553	0.293901	0.308642	0.322256	0.331449	0.332096	0.327574
Total Power	26.56022	31.19529	36.10045	41.26372	46.66666	52.20541	57.27445	60.97287	63.68043

Task 6(a)

Task Description

Using the program, values of r_c and n must be recommended to provide the best design of a system according to these design priorities:

- Maximize power output P_{tot}; must be between 20 to 50 W
- Maximize system efficiency η_{tot}
- Minimize concentration ratio r_c
- Minimize PV cell operating temperature T_{pv}

Power System Analysis

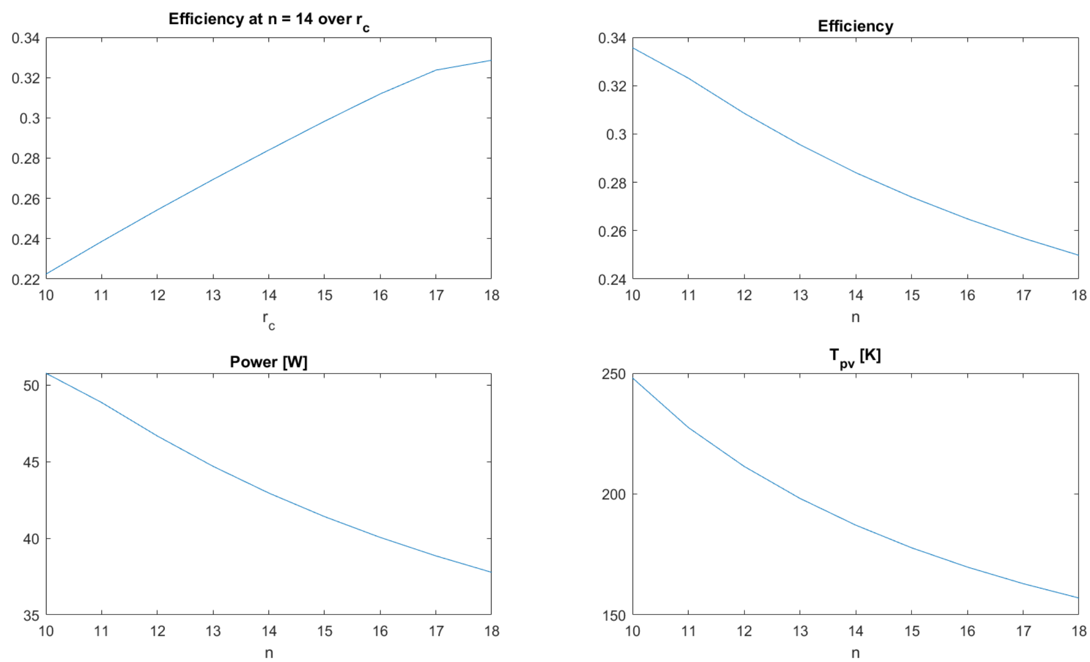


Figure 11: Performance Plots at rc = 14

Figure 9 above charts the variation in power P_{tot}, efficiency η_{tot}, and temperature T_{pv} over a range of n from 10 to 18. This range was loosely based around the previously given $n = 12$. 10 was selected as a lower bound because as n kept decreasing, T_{pv} was increasing too much. 18 was selected as an upper bound because P_{tot} kept decreasing.

At the middle of this range, for $n = 14$, efficiency was plotted in the top left over the r_c range from 10 to 18. Given that given priorities were to minimize r_c while also maximizing η_{tot}, $r_c = 14$ was a balanced choice. It is lower than the r_c value of 15 used in previous tasks, and it also boasted a healthy system efficiency of $\eta_{tot} = 0.284$.

With the r_c value selected, the remaining plots must be analyzed to select the best n value. Criteria for T_{pv} and P_{tot} must be defined. Because P_{tot} should be maximized as a key performance parameter of a power system, a $P_{tot} > 40$ W is desirable. Additionally, a minimal $T_{pv} < 200$ K is also desirable.

Looking at $n = 13$:

- $P_{tot} = 44.68913$ W

- $T_{pv} = 198.2065 \text{ K}$

However, T_{pv} is a little too close to 200 K. Looking at $n = 14$:

- $P_{tot} = 42.9461 \text{ W}$
- $T_{pv} = 187.1328 \text{ K}$

This value of T_{pv} is better, and P_{tot} is a comfortable ~43 W. Therefore, it is recommended to design the hybrid power system with $r_c = 14, n = 14.$ Further data is available in Appendix B.

Task 6(b)

Task Description

Modify the program from Task 4 to eliminate the PV cell and deliver the solar energy directly to the TE module. Determine the following:

- Operating point hot side temperature T_H
- Operating point cold side temperature T_C
- System efficiency η_{tot}
- Power output P_{tot}

Assumptions/Idealizations

- Because the PV cell is eliminated $\dot{Q}_{pv}$ is instead modeled as $\dot{Q}_{sun}$
- $\dot{Q}_{sun} = 0.95 * P" * A_{pv} = 0.95 * r_c * I_D * A_{pv}$, a constant value
- Because the power system only consists of the TE module, $\eta_{tot} = \eta_{TE}$ and $P_{tot} = \dot{W}$.

Algorithm Summary

The algorithm is the same as Task 4 except for a few changes due to modified assumptions:

- $T_H = T_{pv,2} - \dot{Q}_{sun}/\Lambda_{cont}$
- Error $E_{\dot{Q}}$ is computed as $E_{\dot{Q}} = \left|\dot{Q}_{sun} - \dot{Q}_H\right|$
- $P_{tot} = \dot{W}$
- $\eta_{tot} = \eta_{TE}$

Results

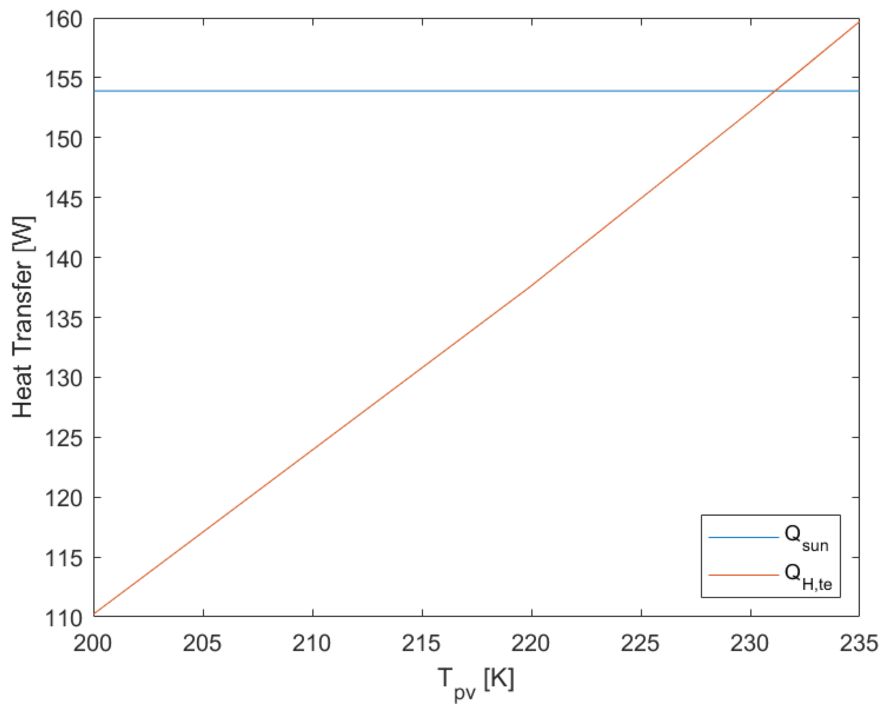


Figure 12: Heat Transfer Rate Convergence for Modified System

Figure 12 shows the convergence of the heat transfer rates of $\dot{Q}_{sun}$ and $\dot{Q}_{H,te}$. T_{pv} converges to 231.1398 K. This result is not relevant due to the removal of the PV cell; however, it can be used to find the temperature T_H of the hot side of the TE module.

This value is utilized in the bisection method to converge upon a cold side T_C value. The T_C bisection method provides the following results:

- $T_H = 230.6268$ K
- $T_C = 97.7673$ K
- $P_{tot} = 29.0402$ W
- $\eta_{tot} = 0.1887$

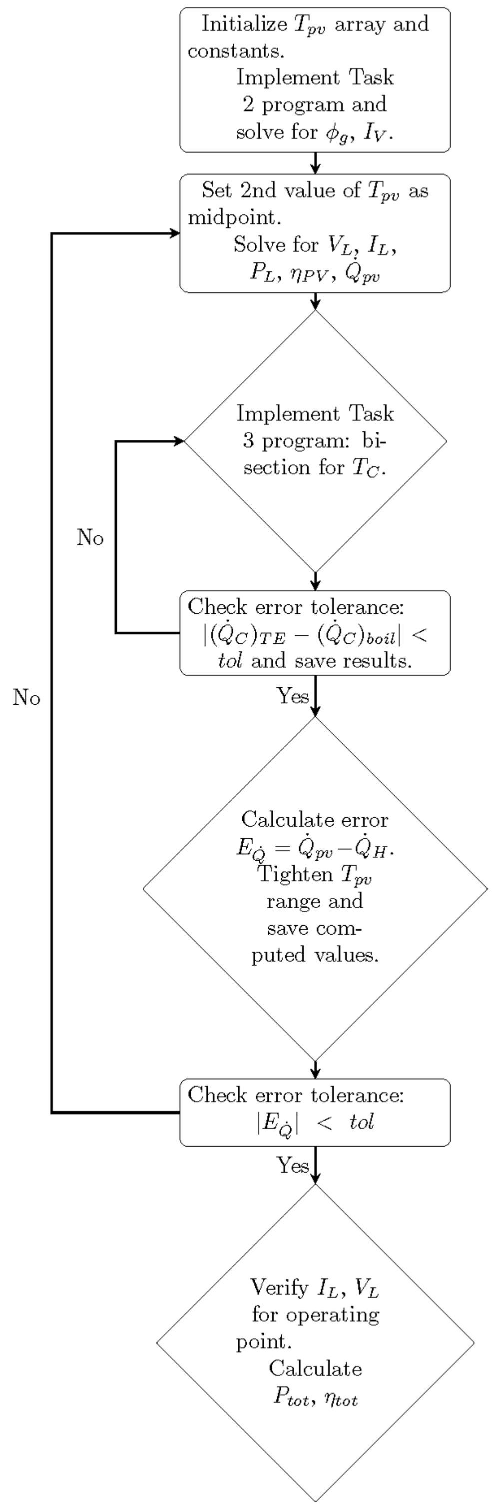
There is a significant drop in system power and efficiency. Recall the full hybrid power system results:

  - $T_H = 216.1843$ K
  - $T_C = 96.8542$ K
  - Total power output: $P_{tot} = 52.2054$ W
  - Overall efficiency: $\eta_{tot} = 0.32226$

<u>Removing the PV cell decreases efficiency by 41.44% (loss of 0.1336 efficiency value) and decreases the power output by 44.37% (loss of 23.1652 W).</u> Meanwhile, T_H also drops by 6.26% (14.4425 K) and T_C drops by 0.93% (0.9131 K).

Appendix A: Program Flowchart

Flow chart for Task 4. The programs from Tasks 2 and 3 are implemented (with Task 1 code built into Task 2).



Appendix B: Program 6(a) Table Results

$n = 10$

rc	10	11	12	13	14	15	16	17	18
PV Eff.	0.1202	0.1323	0.1442	0.1553	0.1607	0.1545	0.1386	0.1246	0.1064
TE Eff.	0.1787	0.1869	0.1944	0.2014	0.2086	0.2161	0.2257	0.2326	0.2427
Total Eff.	0.2774	0.2945	0.3106	0.3254	0.3357	0.3361	0.3331	0.3253	0.3233
Total Power	29.9642	34.9824	40.2548	45.6916	50.7637	54.4531	57.5559	59.7338	62.8491
T_pv	217.8808	225.8618	233.3652	240.5334	248.0469	256.25	267.0532	275	287.1887

$n = 11$

rc	10	11	12	13	14	15	16	17	18
PV Eff.	0.1202	0.1323	0.1443	0.1562	0.1674	0.1735	0.1694	0.1543	0.1383
TE Eff.	0.1594	0.1672	0.1743	0.1809	0.187	0.1932	0.1991	0.2085	0.2167
Total Eff.	0.2605	0.2774	0.2935	0.3088	0.3231	0.3332	0.3327	0.3303	0.3251
Total Power	28.1343	32.9501	38.0318	43.3607	48.8504	53.9765	57.4861	60.6372	63.191
T_pv	201.3451	208.4714	215.1734	221.492	227.5416	233.9063	240	250	259.1296

$n = 12$

rc	10	11	12	13	14	15	16	17	18
PV Eff.	0.1202	0.1323	0.1443	0.1563	0.1682	0.1793	0.1848	0.1788	0.1693
TE Eff.	0.1429	0.1502	0.1569	0.1631	0.1688	0.1742	0.1799	0.1866	0.1912
Total Eff.	0.2459	0.2626	0.2786	0.2939	0.3086	0.3223	0.3314	0.3321	0.3245
Total Power	26.5602	31.1953	36.1005	41.2637	46.6667	52.2054	57.2727	60.9728	63.083
T_pv	187.9248	194.3355	200.3729	206.0656	211.446	216.6275	222.1973	229.018	233.75

$n = 13$

rc	10	11	12	13	14	15	16	17	18
PV Eff.	0.120243	0.132268	0.144292	0.156315	0.16833	0.180221	0.190872	0.194766	0.187532
TE Eff.	0.128544	0.135419	0.141747	0.147585	0.152985	0.158002	0.162801	0.168079	0.174428
Total Eff.	0.233331	0.249775	0.265585	0.280831	0.295563	0.309748	0.322599	0.330109	0.329249
Total Power	25.19971	29.67327	34.41988	39.42865	44.68913	50.17912	55.74504	60.60801	64.00609
T_pv	176.8802	182.684	188.1577	193.3248	198.2065	202.8323	207.3425	212.4011	218.6262

$n = 14$

rc	10	11	12	13	14	15	16	17	18
PV Eff.	0.1202	0.1323	0.1443	0.1563	0.1683	0.1803	0.1921	0.2018	0.2029
TE Eff.	0.1161	0.1226	0.1285	0.134	0.1391	0.1439	0.1483	0.1526	0.1577

Total Eff.	0.2224	0.2386	0.2543	0.2694	0.284	0.2983	0.3119	0.3237	0.3286
Total Power	24.0184	28.3475	32.9518	37.821	42.9461	48.3169	53.899	59.4229	63.8815
T_pv	167.6819	172.9659	177.9562	182.6727	187.1328	191.3529	195.3672	199.3864	204.1656

$n = 15$

rc	10	11	12	13	14	15	16	17	18
PV Eff.	0.1202	0.1323	0.1443	0.1563	0.1683	0.1804	0.1923	0.2038	0.2115
TE Eff.	0.1053	0.1113	0.1169	0.1221	0.1269	0.1314	0.1355	0.1395	0.1436
Total Eff.	0.2129	0.2289	0.2443	0.2593	0.2739	0.288	0.3018	0.3149	0.3247
Total Power	22.988	27.1878	31.6642	36.4077	41.4097	46.6622	52.1534	57.8127	63.1251
T_pv	159.9415	164.7767	169.3491	173.6756	177.771	181.6484	185.3226	188.8478	192.5763

$n = 16$

rc	10	11	12	13	14	15	16	17	18
PV Eff.	0.1202	0.1323	0.1443	0.1563	0.1683	0.1804	0.1924	0.2043	0.215
TE Eff.	0.0958	0.1014	0.1067	0.1116	0.1161	0.1203	0.1243	0.128	0.1315
Total Eff.	0.2045	0.2203	0.2356	0.2504	0.2649	0.279	0.2927	0.3061	0.3182
Total Power	22.0862	26.1694	30.5307	35.1608	40.0513	45.1952	50.5849	56.1994	61.8623
T_pv	153.3684	157.8134	162.0218	166.0083	169.7856	173.3648	176.7564	179.9784	183.1436

$n = 17$

rc	10	11	12	13	14	15	16	17	18
PV Eff.	0.1202	0.1323	0.1443	0.1563	0.1683	0.1804	0.1924	0.2044	0.216
TE Eff.	0.0904	0.0927	0.0976	0.1022	0.1065	0.1105	0.1142	0.1177	0.121
Total Eff.	0.2035	0.2127	0.2278	0.2426	0.2569	0.2709	0.2846	0.298	0.3109
Total Power	21.974	25.2717	29.5293	34.0568	38.8465	43.8915	49.1855	54.7192	60.437
T_pv	150	151.8448	155.7343	159.4226	162.9206	166.2379	169.3832	172.3662	175.2221

$n = 18$

rc	10	11	12	13	14	15	16	17	18
PV Eff.	0.1205	0.1323	0.1443	0.1563	0.1683	0.1804	0.1924	0.2044	0.2163
TE Eff.	0.0801	0.085	0.0896	0.094	0.098	0.1018	0.1053	0.1086	0.1117
Total Eff.	0.1909	0.206	0.221	0.2356	0.2498	0.2638	0.2774	0.2908	0.3038
Total Power	20.6186	24.4775	28.6415	33.0762	37.7744	42.7296	47.9358	53.387	59.0631
T_pv	142.8893	146.6921	150.3006	153.7258	156.9772	160.063	162.991	165.7681	168.4083

Appendix C: MATLAB Scripts

Task1.m

```
%% Data Input & Linear Curve Fit
data = [28.85, 0.379; 30.492, 0.3737; 43.279, 0.3526; 50.49, 0.3316;...
    60.65, 0.311];
x = data(:,1)+273.15; %convert oC to K
y = data(:,2);

X = [ones(length(x),1), x];

b = X\y;

T_pv = linspace(20+273.15,100+273.15,81);

%% Plot Fit
Voc = @(t) b(2).*t + b(1);
plot(T_pv, Voc(T_pv));

hold on;
plot(x,y,'o');
xlabel("Temperature [K]");
ylabel("V_o_c [V]");
title("Linear Fit");

%% Plot I0_IV
kb = 1.3807*10^-23; %[J/K]
qe = 1.602*10^-19; %[C]

%For Kelvin
I0_IV = @(t) (exp(Voc(t).*qe./(kb.*t))-1).^-1;
figure;
plot(T_pv, I0_IV(T_pv));
xlabel("Temperature [K]");
ylabel("I_0/I_V Ratio");
title("I0\_IV Ratio over Temp.");
```

Task2.m

```matlab
%% Task 2 Problem Statement
% Compute power delivered by the PV cell. Compute PV cell conversion
% efficiency. Finally, determine power output. Use the curve fit for I0_IV
% as found in Task 1 to accomplish this.

%% I0_IV Curve (Task 1)
% Data Input & Linear Curve Fit
data = [28.85, 0.379; 30.492, 0.3737; 43.279, 0.3526; 50.49, 0.3316;...
    60.65, 0.311];
x = data(:,1)+273.15; %convert oC to K
y = data(:,2);

X = [ones(length(x),1), x];

b = X\y;

T_pv = linspace(20+273.15,100+273.15,81);

% Plot Fit
Voc = @(t) b(2).*t + b(1);

%For Kelvin
kb = 1.3807*10^-23; %[J/K]
qe = 1.602*10^-19; %[C]
I0_IV = @(t) (exp(Voc(t).*qe./(kb.*t))-1).^-1;

%% Initializing Constants
pressure = 163; %[kPa]
rc = 15; %concentration ratio, unitless
I_d = 1080; %[W/m^2]
P = rc*I_d; %[W/m^2]
L_x = 0.1; %[m]
L_y = 0.1; %[m]
A = L_x*L_y;

Vg = 1.1; %[V]

T_source=6000; %Temperature of the sun (source)

%% Eq. 14.16: Solving for phi
phi = @(T_source) (pi^4*kb*T_source./(2.404*15*P)).^-1;
% Accepts constant or array of source temperature.
% Returns constant phi value or array of phi vals.

%% Eq. 14.25: Solving for phi_g
sigma = @(x) x.^2./(exp(x)-1);

X = qe*Vg./(kb.*T_source);
integ = integral(sigma, X, Inf);
phi_i = phi(T_source);
phi_g = phi_i*0.416*integ;

IV = phi_g.*qe*A;

%% P = IV
RL = 0.0070; %[Ohm]
fg = 2.65*10^14; %265 [THz]
RS = 0;
TK1 = 20+273; %K
TK2 = 80+273; %K
```

```
syms IL
eqn1 = kb*TK1/qe*log(I0_IV(TK1).^-1-I0_IV(TK1).^-1*IL/IV+1) == IL*RL;
IL_op1 = double(solve(eqn1,IL));

eqn2 = kb*TK2/qe*log(I0_IV(TK2).^-1-I0_IV(TK2).^-1*IL/IV+1) == IL*RL;
IL_op2 = double(solve(eqn2,IL));

VL = @(IL, t) kb*t/qe*log(I0_IV(t).^-1-I0_IV(t).^-1.*IL./IV+1);
VL_op1 = VL(IL_op1,TK1);
VL_op2 = VL(IL_op2,TK2);

%% Plot Resistor and PV - Operating Point
IL_arr = linspace(0,IV, ceil(IV)*10);
plot(IL_arr, IL_arr*RL); %Resistor
hold on;
plot(IL_arr, VL(IL_arr,TK1)); %At operating temp. 1
hold on;
plot(IL_arr, VL(IL_arr,TK2)); %At operating temp. 2
xlabel("I_L [A]");
ylabel("V_L [V]");
%title("VL over IL Performance");
legend("Resistor","PV Cell @ 20 ^oC", "PV Cell @ 80 ^oC","Location","northwest");

%% Power and Efficiency Results

P1 = IL_op1*VL_op1;
eff1 = P1/(P*A);
P2 = IL_op2*VL_op2;
eff2 = P2/(P*A);
fprintf("At 20 oC...\n" ...
    + "    Power = " + P1 + " W \n" ...
    + "    Operating point: IL = " + IL_op1 + " A , VL = " + VL_op1 + " V \n"...
    + "    Efficiency = " + eff1 + "\n\n");
fprintf("At 80 oC...\n" ...
    + "    Power = " + P2 + " W \n" ...
    + "    Operating point: IL = " + IL_op2 + " A , VL = " + VL_op2 + " V \n"...
    + "    Efficiency = " + eff2 + "\n");
```

Task3Bisection.m

```
%% Task 3 Problem Statement
% Assume: n = 12, R_L,te = 0.10 ohms, TH = 50 oC. All necessary constants
%           given.
% Find: TC, power output of the module Wdot, QHdot, QCdot, current to the
%           load IL, voltage across the load VL
tic
%% Assumed Constants and Initial Calcs
tol = 1e-5;
a = 0.0017; %[V/K], mean Seebeck coefficient
pA = 0.0020; %[ohm*cm]
pB = 0.0030; %[ohm*cm]
lA = 0.032; %[W/(cm*K)]
lB = 0.021; %[W/(cm*K)]
A_pv = 0.01; %[m^2]
k_wEff = 6.5; %[W/(mK)]
Tsat = 95.0; %[K]
Uev = 25; %[W/(m^2*K)]
tw = 0.004; %[m], 4 mm

RA_min = (sqrt(lA*pA)+sqrt(lB*pB))^2;
Z = a^2/RA_min;

n = 12; %12 thermoelectric pairs
R_lte = 0.10; %[ohms]
TH = 50 + 273.15; %[K]

%% TC and QHdot Computation
error = [1000 1000 1000]; %initialize error
TC = [90 0 120]; %LB Midpoint UB

while abs(error(2))>tol || (error(1)*error(3)<0)
    midpoint = (TC(1)+TC(3))/2;
    TC(2) = midpoint;

    m_maxEff = sqrt(1+0.5.*(TH+TC).*Z);
    R = R_lte./(n.*m_maxEff);
    A = RA_min./R;

    % (ii) Computing QC_boil
    QC_boil = (k_wEff/tw + Uev)*(TC-Tsat)*A_pv; %[W]

    % (iii) Compute efficiency using Eq. 11
    eff = (TH-TC)./TH.*((1+m_maxEff).^2./(m_maxEff.*Z*TH) + 1 +
1./(2*m_maxEff).*(1+TC./TH)).^-1;

    % (iv)  Computer power output Wdot using Eq. 10
    R_batt = n*R;
    Wdot = (n^2*a^2.*(TH-TC).^2.*R_lte)./(R_lte+R_batt).^2; %[W]

    % (v) Compute heat input and rejection
    QH = Wdot./eff; %[W]
    QC_TE = QH - Wdot; %[W]

    % (vi) Computer error - run if statement to find optimal TC and QH for
    %        minimal error
    error = QC_TE-QC_boil;

    if error(1)*error(2)<0
        TC(3) = TC(2);
    else
        TC(1) = TC(2);
```

```
    end
end

error_result = error(2);
TC_result = TC(2);
QH_result = QH(2);
QC_result = QC_boil(2);
A_batt = n*A(2);
Wdot_result = Wdot(2);
eff_result = eff(2);

%% IL and VL Computation
% Use Eqs. (8) and (9)

syms IL
heatRateEqn = QH_result == n*a*IL*TH + A_batt*(TH-TC_result)-0.5*IL^2*R_batt(2);
%We get 2 roots for IL because heatRateEqn is 2nd order eqn
IL_result = double(solve(heatRateEqn, IL));

%Solve for VL using both IL roots
%Select true VL and IL based on most reasonable VL_result
VL_result = n*a*(TH-TC_result)-R_batt(2)*IL_result;
VL_true = VL_result(1); %[V]
IL_true = IL_result(1); %[A]

%% Results
fprintf("TC = " + TC_result + " K \n" ...
    + "Power output of module Wdot = " + Wdot_result + " W \n" ...
    + "Heat rate QH = " + QH_result + " W \n" ...
    + "Contact surface heat rate QC = " + QC_result + " W \n" ...
    + "Current to the load IL = " + IL_true + " A \n" ...
    + "Voltage across the load VL = " + VL_true + " V \n \n" ...
    + "Error in QC = " + error_result + "\n");
  toc
```

Task4Bisection.m

```
%% Task 4 Problem Statement
% Task 2 finds power and performance of PV cell
% Task 3 finds power and params for TE module
% Combine programs from Task 2 and 3 in order to determine operating point
% and performance for HYBRID system (PV + TE).
tic;
tol = 1e-5;
%% Initialize Constants

T_pv = [170, 0, 230];

ID = 1080; %[W/m^2]
a = 0.0017; %[V/K], from Task 3
lA = 0.032; %[W/(cm K)], from Task 3
lB = 0.021; %[W/(cm K)], from Task 3
pA = 0.0020; %[ohm cm], from Task 3
pB = 0.0030; %[ohm cm], from Task 3
A_cont = 300; %[W/K]
Rs = 0; %[ohm]
Vg = 1.1; %[V]
rc = 15; %unitless
Ly = 0.1; %[m] from 10 cm
Lz = 0.1; %[m] from 10 cm
A_pv = Ly*Lz; %[m^2]
RL_pv = 0.0070; %[ohm]
RL_te = 0.10; %[ohm]

T_source=6000; %Temperature of the sun (source)
P = rc*ID; %[W/m^2]

Tsat = 90.2; %[K]
n = 12; %unitless
Uev = 25; %[W/(m^2*K)]
tw = 0.004; %[m], 4 mm
k_wEff = 6.5; %[W/(mK)]

%% I0_IV Curve (Task 1 for PV Cell Task 2)
% Data Input & Linear Curve Fit
data = [28.85, 0.379; 30.492, 0.3737; 43.279, 0.3526; 50.49, 0.3316;...
    60.65, 0.311];
x = data(:,1)+273.15; %convert oC to K
y = data(:,2);
X = [ones(length(x),1), x];
b = X\y;
% Plot Fit
Voc = @(t) b(2).*t + b(1);
%For Kelvin
kb = 1.3807*10^-23; %[J/K]
qe = 1.602*10^-19; %[C]
I0_IV = @(t) (exp(Voc(t).*qe./(kb.*t))-1).^-1;

%% Step 2: Task 2 (PV Cell Analysis)
phi = @(T_source) (pi^4*kb*T_source./(2.404*15*P)).^-1;
sigma = @(x) x.^2./(exp(x)-1);
X = qe*Vg./(kb.*T_source);
integ = integral(sigma, X, Inf);
phi_i = phi(T_source);
phi_g = phi_i*0.416*integ;
IV = phi_g.*qe*A_pv;
i = 1;
errorEQ = [1000 1000 1000];
```

```
while abs(errorEQ(2))>tol || (errorEQ(1)*errorEQ(3)<0)
    midT_pv = (T_pv(1)+T_pv(3))/2;
    T_pv(2) = midT_pv;

    syms IL
    eqn1 = kb.*T_pv(1)./qe.*log(I0_IV(T_pv(1)).^-1-I0_IV(T_pv(1)).^-1.*IL./IV+1) ==
IL.*RL_pv;
    eqn2 = kb.*T_pv(2)./qe.*log(I0_IV(T_pv(2)).^-1-I0_IV(T_pv(2)).^-1.*IL./IV+1) ==
IL.*RL_pv;
    eqn3 = kb.*T_pv(3)./qe.*log(I0_IV(T_pv(3)).^-1-I0_IV(T_pv(3)).^-1.*IL./IV+1) ==
IL.*RL_pv;
    IL_op(1) = double(solve(eqn1,IL));
    IL_op(2) = double(solve(eqn2,IL));
    IL_op(3) = double(solve(eqn3,IL));

    VL = @(IL, t) kb*t/qe.*log(I0_IV(t).^-1-I0_IV(t).^-1.*IL./IV+1);
    VL_op = VL(IL_op,T_pv);

    PL = IL_op.*VL_op; %DELIVERABLE: POWER OUTPUT
    eff_pv = PL/(P*A_pv);

    Q_pv = P*A_pv-PL; %DELIVERABLE: WASTE HEAT TRANSFER RATE
    Qplot(i) = Q_pv(2);
%% Step 3: Task 3 (TE Module Analysis)

    TH = T_pv(2) - Q_pv/A_cont;
    RA_min = (sqrt(lA*pA)+sqrt(lB*pB))^2;
    Z = a^2/RA_min;

    error = [1000 1000 1000]; %initialize error
    TC = [90 0 120]; %LB Midpoint UB

    while abs(error(2))>tol || (error(1)*error(2)<0)
        midTC = (TC(1)+TC(3))/2;
        TC(2) = midTC;

        m_maxEff = sqrt(1+0.5.*(TH+TC).*Z);
        R = RL_te./(n.*m_maxEff);
        A = RA_min./R;

        % (ii) Computing QC_boil
        QC_boil = (k_wEff/tw + Uev)*(TC-Tsat)*A_pv; %[W]

        % (iii) Compute efficiency using Eq. 11
        eff = (TH-TC)./TH.*((1+m_maxEff).^2./(m_maxEff.*Z.*TH) + 1 +
1./(2*m_maxEff).*(1+TC./TH)).^-1;

        % (iv)  Computer power output Wdot using Eq. 10
        R_batt = n*R;
        Wdot = (n^2*a^2.*(TH-TC).^2.*RL_te)./(RL_te+R_batt).^2; %[W]

        % (v) Compute heat input and rejection
        QH = Wdot./eff; %[W]
        QC_TE = QH - Wdot; %[W]

        % (vi) Computer error - run if statement to find optimal TC and QH for
        %       minimal error
        error = QC_TE-QC_boil;
        if error(1)*error(2)<0
            TC(3) = TC(2);
        elseif error(1)*error(2)>0 && error(1)>0
            TC(1) = TC(2);
```

```
        elseif error(1)*error(2)>0 && error(1)<0
            TC(3) = TC(2);
        end
    end

    error_result = error(2);
    TC_result = TC(2);
    QH_result = QH(2);
    QC_result = QC_boil(2);
    A_batt = n*A(2);
    Wdot_result = Wdot(2);
    eff_result = eff(2);
    QHplot(i) = QH_result;
%% Step 4: Compute Error EQ
% Power, Operating Point, PV cell efficiency

    errorEQ = Q_pv-QH;%_result;
    if errorEQ(1)*errorEQ(2)<0
        T_pv(3) = T_pv(2);
    elseif errorEQ(1)*errorEQ(2)>0 && errorEQ(1)>0
        T_pv(1) = T_pv(2);
    elseif errorEQ(1)*errorEQ(2)>0 && errorEQ(1)<0
        T_pv(3) = T_pv(2);
    end

    PL_result = PL(2);
    Q_pv_result = Q_pv(2);
    eff_pv_result = eff_pv(2);
    errorEQ_result = errorEQ(2);
    IL_pv_result = IL_op(2);
    VL_pv_result = VL_op(2);
    T_pv_final = T_pv(2);

    T_pvplot(i) = T_pv_final;
    i= i +1;
end

syms IL_te
heatRateEqn = QH_result == n*a*IL_te*TH(2) + A_batt*(TH(2)-TC_result)-
0.5*IL_te^2*R_batt(2);

IL_result = double(solve(heatRateEqn, IL_te));
VL_result = n*a*(TH(2)-TC_result)-R_batt(2)*IL_result;
VL_true = VL_result(1); %[V]
IL_true = IL_result(1); %[A]

%% Plots
%Error plot
plot(T_pvplot, Qplot)
hold on;
plot(T_pvplot, QHplot);
legend("Q_p_v","Q_H_,_t_e",'Location','southeast');
xlabel("T_p_v [K]");
ylabel("Heat Transfer [W]");

%% Results
P_tot = PL_result + Wdot_result;
eff_tot = P_tot/(P*A_pv);

fprintf("PV Cell Results: \n" ...
    + "    Operating Point: VL = " + VL_pv_result + " V, IL = " + IL_pv_result  + " A
\n" ...
    + "    Operating Point Temperature: T_pv = " + T_pv(2) + " K \n" ...
```

```
    + "    Power = " + PL_result + " W \n" ...
    + "    Efficiency = " + eff_pv_result + "\n \n");

fprintf("Thermoelectric Module Results: \n" ...
    + "    Operating Point: VL = " + VL_op(2) + " V, IL = " + IL_op(2)  + " A \n" ...
    + "    Power Output: Wdot = " + Wdot_result + " W \n" ...
    + "    Efficiency = " + eff_result + "\n \n");

fprintf("Combined power output of hybrid device: \n" ...
    + "    Total Power Output = " + (P_tot) + " W \n" ...
    + "    Efficiency = " + eff_tot + "\n \n");

toc; %Find program runtime
```

Task5.m

```
%% Task 5 Problem Statement
% a) Iterate over a range of ID from [720, 1080]
% b) Iterate over a range of rc from [10, 18]

tic;
tol = 1e-5;
%% Initialize Constants
results = [];
ID_range = linspace(720, 1080, 10);
rc_range = linspace(10, 18, 9);
% Iterate over 10 points for 5(a), 9 points for 5(b)
for h=1:10
T_pv = [170, 0, 240];
% 5(a)
ID = ID_range(h);%[W/m^2]
rc = 15; %unitless

% 5(b)
% ID = 1080;
% rc = rc_range(h);

a = 0.0017; %[V/K], from Task 3
lA = 0.032; %[W/(cm K)], from Task 3
lB = 0.021; %[W/(cm K)], from Task 3
pA = 0.0020; %[ohm cm], from Task 3
pB = 0.0030; %[ohm cm], from Task 3
A_cont = 300; %[W/K]
Rs = 0; %[ohm]
Vg = 1.1; %[V]
Ly = 0.1; %[m] from 10 cm
Lz = 0.1; %[m] from 10 cm
A_pv = Ly*Lz; %[m^2]
%Tw_in = 0.004; %[m] from 4 mm, Task 3
RL_pv = 0.0070; %[ohm]
RL_te = 0.10; %[ohm]

T_source=6000; %Temperature of the sun (source)
P = rc*ID; %[W/m^2]

Tsat = 90.2; %[K]
n = 12; %unitless
Uev = 25; %[W/(m^2*K)]
tw = 0.004; %[m], 4 mm
k_wEff = 6.5; %[W/(mK)]

%% I0_IV Curve (Task 1 for PV Cell Task 2)
% Data Input & Linear Curve Fit
data = [28.85, 0.379; 30.492, 0.3737; 43.279, 0.3526; 50.49, 0.3316;...
    60.65, 0.311];
x = data(:,1)+273.15; %convert oC to K
y = data(:,2);
X = [ones(length(x),1), x];
b = X\y;
% Plot Fit
Voc = @(t) b(2).*t + b(1);
%For Kelvin
kb = 1.3807*10^-23; %[J/K]
qe = 1.602*10^-19; %[C]
I0_IV = @(t) (exp(Voc(t).*qe./(kb.*t))-1).^-1;

%% Step 2: Task 2 (PV Cell Analysis)
```

```
phi = @(T_source) (pi^4*kb*T_source./(2.404*15*P)).^-1;
sigma = @(x) x.^2./(exp(x)-1);
X = qe*Vg./(kb.*T_source);
integ = integral(sigma, X, Inf);
phi_i = phi(T_source);
phi_g = phi_i*0.416*integ;
IV = phi_g.*qe*A_pv;
i = 1;
errorEQ = [1000 1000 1000];

while abs(errorEQ(2))>tol || (errorEQ(1)*errorEQ(3)<0)
    midT_pv = (T_pv(1)+T_pv(3))/2;
    T_pv(2) = midT_pv;

    syms IL
    eqn1 = kb.*T_pv(1)./qe.*log(I0_IV(T_pv(1)).^-1-I0_IV(T_pv(1)).^-1.*IL./IV+1) ==
IL.*RL_pv;
    eqn2 = kb.*T_pv(2)./qe.*log(I0_IV(T_pv(2)).^-1-I0_IV(T_pv(2)).^-1.*IL./IV+1) ==
IL.*RL_pv;
    eqn3 = kb.*T_pv(3)./qe.*log(I0_IV(T_pv(3)).^-1-I0_IV(T_pv(3)).^-1.*IL./IV+1) ==
IL.*RL_pv;
    IL_op(1) = double(solve(eqn1,IL));
    IL_op(2) = double(solve(eqn2,IL));
    IL_op(3) = double(solve(eqn3,IL));

    VL = @(IL, t) kb*t/qe.*log(I0_IV(t).^-1-I0_IV(t).^-1.*IL./IV+1);
    VL_op = VL(IL_op,T_pv);

    PL = IL_op.*VL_op; %DELIVERABLE: POWER OUTPUT
    eff_pv = PL/(P*A_pv);

    Q_pv = P*A_pv-PL; %DELIVERABLE: WASTE HEAT TRANSFER RATE
    Qplot(i) = Q_pv(2);
%% Step 3: Task 3 (TE Module Analysis)

    TH = T_pv(2) - Q_pv/A_cont;
    RA_min = (sqrt(lA*pA)+sqrt(lB*pB))^2;
    Z = a^2/RA_min;

    error = [1000 1000 1000]; %initialize error
    TC = [90 0 120]; %LB Midpoint UB

    while abs(error(2))>tol || (error(1)*error(2)<0)
        midTC = (TC(1)+TC(3))/2;
        TC(2) = midTC;

        m_maxEff = sqrt(1+0.5.*(TH+TC).*Z);
        R = RL_te./(n.*m_maxEff);
        A = RA_min./R;

        % (ii) Computing QC_boil
        QC_boil = (k_wEff/tw + Uev)*(TC-Tsat)*A_pv; %[W]

        % (iii) Compute efficiency using Eq. 11
        eff = (TH-TC)./TH.*((1+m_maxEff).^2./(m_maxEff.*Z.*TH) + 1 +
1./(2*m_maxEff).*(1+TC./TH)).^-1;

        % (iv)  Computer power output Wdot using Eq. 10
        R_batt = n*R;
        Wdot = (n^2*a^2.*(TH-TC).^2.*RL_te)./(RL_te+R_batt).^2; %[W]

        % (v) Compute heat input and rejection
        QH = Wdot./eff; %[W]
```

```matlab
        QC_TE = QH - Wdot; %[W]

        % (vi) Computer error - run if statement to find optimal TC and QH for
        %       minimal error
        error = QC_TE-QC_boil;
        if error(1)*error(2)<0
            TC(3) = TC(2);
        else
            TC(1) = TC(2);
        end
    end

    error_result = error(2);
    TC_result = TC(2);
    QH_result = QH(2);
    QC_result = QC_boil(2);
    A_batt = n*A(2);
    Wdot_result = Wdot(2);
    eff_result = eff(2);
    QHplot(i) = QH_result;
%% Step 4: Compute Error EQ
% Power, Operating Point, PV cell efficiency

    errorEQ = Q_pv-QH;%_result;
    if errorEQ(1)*errorEQ(2)<0
        T_pv(3) = T_pv(2);
    elseif errorEQ(1)*errorEQ(2)>0 && errorEQ(1)>0
        T_pv(1) = T_pv(2);
    elseif errorEQ(1)*errorEQ(2)>0 && errorEQ(1)<0
        T_pv(3) = T_pv(2);
    end

    PL_result = PL(2);
    Q_pv_result = Q_pv(2);
    eff_pv_result = eff_pv(2);
    errorEQ_result = errorEQ(2);
    IL_pv_result = IL_op(2);
    VL_pv_result = VL_op(2);
    T_pv_final = T_pv(2);

    T_pvplot(i) = T_pv_final;
    i= i +1;
    if i>=90
        break
    end
end

syms IL_te
heatRateEqn = QH_result == n*a*IL_te*TH(2) + A_batt*(TH(2)-TC_result)-
0.5*IL_te^2*R_batt(2);

IL_result = double(solve(heatRateEqn, IL_te));
VL_result = n*a*(TH(2)-TC_result)-R_batt(2)*IL_result;
VL_true = VL_result(1); %[V]
IL_true = IL_result(1); %[A]

%% Results
P_tot = PL_result + Wdot_result;
eff_tot = P_tot/(P*A_pv);

% 5(a)
col = [ID; eff_pv_result; eff_result; eff_tot; P_tot];
disp(ID);
```

```
% 5(b)
% col = [rc; eff_pv_result; eff_result; eff_tot; P_tot];
% disp(rc);

results = [results col];

end
%% Plotting 5(a): Variation in ID
results_ID = [720   760 800 840 880 920 960 1000    1040    1080;...
0.120243207 0.126923381 0.133603536 0.140283589 0.146963128 0.153640291 0.160307132
0.166931865 0.173396299 0.179304313;...
0.142863119 0.147008975 0.150958348 0.154724322 0.15831879  0.161752576 0.165036272
0.168182356 0.171205564 0.174183114;...
0.245927994 0.255273475 0.264393325 0.273302617 0.282014897 0.290541145 0.298886905
0.307039218 0.314904727 0.322255628;...
26.5602234  29.10117612 31.72719905 34.43612979 37.22596642 40.09467807 43.03971426
46.05588263 49.12513744 52.20541181];

subplot(3,1,1);
plot(results(1,:),results(2,:));
ylabel("PV Eff.");
title("Component & Total Efficiency over ID (720 to 1080)");
subplot(3,1,2);
plot(results(1,:),results(3,:));
ylabel("TE Eff.");
subplot(3,1,3);
plot(results(1,:),results(4,:));
xlabel("ID [W/m^2]");
ylabel("Total Eff.");

figure;
plot(results(1,:),results(5,:));
xlabel("ID [W/m^2]");
ylabel("Total Power [W]");
title("Total Power over ID (720 to 1080)");

%% Plotting 5(b): Variation in rc
%
% results_rc = [10  11  12  13  14  15  16  17  18;...
% 0.120243207   0.132267509 0.144291435 0.156309166 0.16824432  0.179304313
0.184836665 0.178849182 0.165745873;...
% 0.142863119   0.150183531 0.156900869 0.16308345  0.168796666 0.174183114
0.179856849 0.186624746 0.193995284;...
% 0.245927994   0.262586628 0.278552841 0.293901167 0.308641912 0.322255628 0.33144936
0.332096236 0.327574222;...
% 26.5602234    31.19529139 36.10044822 41.26372385 46.66665708 52.20541181
57.27444937 60.97286891 63.68042874];
%
%
% subplot(3,1,1);
% plot(results(1,:),results(2,:));
% ylabel("PV Eff.");
% title("Component & Total Efficiency over rc (10 to 18)");
% subplot(3,1,2);
% plot(results(1,:),results(3,:));
% ylabel("TE Eff.");
% subplot(3,1,3);
% plot(results(1,:),results(4,:));
% xlabel("rc");
% ylabel("Total Eff.");
%
% figure;
```

```
% plot(results(1,:),results(5,:));
% xlabel("rc");
% ylabel("Total Power [W]");
% title("Total Power over rc (10 to 18)");

toc; %Find program runtime
```

Task6a.m

```
%% Task 6(a) Problem Statement
% a) Iterate over a range of rc and n to recommend system parameters.

tic;
tol = 1e-5;
%% Initialize Constants
results = [];
ID_range = linspace(720, 1080, 10);
rc_range = linspace(10, 18, 9);
% Iterate over 10 points for both 5(a) and 5(b)

for h=1:length(rc_range)
T_pv = [200, 0, 300];
% 5(a)
% ID = ID_range(h);%[W/m^2]
% rc = 15; %unitless

% 5(b)
ID = 1080;
rc = rc_range(h);
n = 10; %unitless

a = 0.0017; %[V/K], from Task 3
lA = 0.032; %[W/(cm K)], from Task 3
lB = 0.021; %[W/(cm K)], from Task 3
pA = 0.0020; %[ohm cm], from Task 3
pB = 0.0030; %[ohm cm], from Task 3
A_cont = 300; %[W/K]
Rs = 0; %[ohm]
Vg = 1.1; %[V]
Ly = 0.1; %[m] from 10 cm
Lz = 0.1; %[m] from 10 cm
A_pv = Ly*Lz; %[m^2]
%Tw_in = 0.004; %[m] from 4 mm, Task 3
RL_pv = 0.0070; %[ohm]
RL_te = 0.10; %[ohm]

T_source=6000; %Temperature of the sun (source)
P = rc*ID; %[W/m^2]

Tsat = 90.2; %[K]
Uev = 25; %[W/(m^2*K)]
tw = 0.004; %[m], 4 mm
k_wEff = 6.5; %[W/(mK)]

%% I0_IV Curve (Task 1 for PV Cell Task 2)
% Data Input & Linear Curve Fit
data = [28.85, 0.379; 30.492, 0.3737; 43.279, 0.3526; 50.49, 0.3316;...
    60.65, 0.311];
x = data(:,1)+273.15; %convert oC to K
y = data(:,2);
X = [ones(length(x),1), x];
b = X\y;
% Plot Fit
Voc = @(t) b(2).*t + b(1);
%For Kelvin
kb = 1.3807*10^-23; %[J/K]
qe = 1.602*10^-19; %[C]
I0_IV = @(t) (exp(Voc(t).*qe./(kb.*t))-1).^-1;

%% Step 2: Task 2 (PV Cell Analysis)
```

```
phi = @(T_source) (pi^4*kb*T_source./(2.404*15*P)).^-1;
sigma = @(x) x.^2./(exp(x)-1);
X = qe*Vg./(kb.*T_source);
integ = integral(sigma, X, Inf);
phi_i = phi(T_source);
phi_g = phi_i*0.416*integ;
IV = phi_g.*qe*A_pv;
i = 1;
errorEQ = [1000 1000 1000];

while abs(errorEQ(2))>tol || (errorEQ(1)*errorEQ(3)<0)
    midT_pv = (T_pv(1)+T_pv(3))/2;
    T_pv(2) = midT_pv;

    syms IL
    eqn1 = kb.*T_pv(1)./qe.*log(I0_IV(T_pv(1)).^-1-I0_IV(T_pv(1)).^-1.*IL./IV+1) ==
IL.*RL_pv;
    eqn2 = kb.*T_pv(2)./qe.*log(I0_IV(T_pv(2)).^-1-I0_IV(T_pv(2)).^-1.*IL./IV+1) ==
IL.*RL_pv;
    eqn3 = kb.*T_pv(3)./qe.*log(I0_IV(T_pv(3)).^-1-I0_IV(T_pv(3)).^-1.*IL./IV+1) ==
IL.*RL_pv;
    IL_op(1) = double(solve(eqn1,IL));
    IL_op(2) = double(solve(eqn2,IL));
    IL_op(3) = double(solve(eqn3,IL));

    VL = @(IL, t) kb*t/qe.*log(I0_IV(t).^-1-I0_IV(t).^-1.*IL./IV+1);
    VL_op = VL(IL_op,T_pv);

    PL = IL_op.*VL_op; %DELIVERABLE: POWER OUTPUT
    %powerPlot(i) = PL;
    eff_pv = PL/(P*A_pv);

    Q_pv = P*A_pv-PL; %DELIVERABLE: WASTE HEAT TRANSFER RATE
    Qplot(i) = Q_pv(2);
%% Step 3: Task 3 (TE Module Analysis)

    TH = T_pv(2) - Q_pv/A_cont;
    RA_min = (sqrt(lA*pA)+sqrt(lB*pB))^2;
    Z = a^2/RA_min;

    error = [1000 1000 1000]; %initialize error
    TC = [90 0 120]; %LB Midpoint UB
    j=1;
    while abs(error(2))>tol || (error(1)*error(2)<0)
        midTC = (TC(1)+TC(3))/2;
        TC(2) = midTC;

        m_maxEff = sqrt(1+0.5.*(TH+TC).*Z);
        R = RL_te./(n.*m_maxEff);
        A = RA_min./R;

        % (ii) Computing QC_boil
        QC_boil = (k_wEff/tw + Uev)*(TC-Tsat)*A_pv; %[W]

        % (iii) Compute efficiency using Eq. 11
        eff = (TH-TC)./TH.*((1+m_maxEff).^2./(m_maxEff.*Z.*TH) + 1 +
1./(2*m_maxEff).*(1+TC./TH)).^-1;

        % (iv)  Computer power output Wdot using Eq. 10
        R_batt = n*R;
        Wdot = (n^2*a^2.*(TH-TC).^2.*RL_te)./(RL_te+R_batt).^2; %[W]

        % (v) Compute heat input and rejection
```

```
        QH = Wdot./eff; %[W]
        QC_TE = QH - Wdot; %[W]

        % (vi) Computer error - run if statement to find optimal TC and QH for
        %       minimal error
        error = QC_TE-QC_boil;
        if error(1)*error(2)<0
            TC(3) = TC(2);
        else
            TC(1) = TC(2);
        end
        j=j+1;
        if j>80
            break;
        end
    end

    error_result = error(2);
    TC_result = TC(2);
    QH_result = QH(2);
    QC_result = QC_boil(2);
    A_batt = n*A(2);
    Wdot_result = Wdot(2);
    eff_result = eff(2);
    QHplot(i) = QH_result;
%% Step 4: Compute Error EQ
% Power, Operating Point, PV cell efficiency

    errorEQ = Q_pv-QH;%_result;
    if errorEQ(1)*errorEQ(2)<0
        T_pv(3) = T_pv(2);
    elseif errorEQ(1)*errorEQ(2)>0 && errorEQ(1)>0
        T_pv(1) = T_pv(2);
    elseif errorEQ(1)*errorEQ(2)>0 && errorEQ(1)<0
        T_pv(3) = T_pv(2);
    end


    PL_result = PL(2);
    Q_pv_result = Q_pv(2);
    eff_pv_result = eff_pv(2);
    errorEQ_result = errorEQ(2);
    IL_pv_result = IL_op(2);
    VL_pv_result = VL_op(2);
    T_pv_final = T_pv(2);

    T_pvplot(i) = T_pv_final;
    i= i +1;
    if i>90
        break;
    end
end
fprintf("Error and ErrorEQ has converged to 0 \n" +...
    "Error = " + error(2) + "\n" + ...
    "ErrorEQ = " + errorEQ(2) + "\n");
syms IL_te
heatRateEqn = QH_result == n*a*IL_te*TH(2) + A_batt*(TH(2)-TC_result)-
0.5*IL_te^2*R_batt(2);
%We get 2 roots for IL because heatRateEqn is 2nd order eqn
IL_result = double(solve(heatRateEqn, IL_te));

%Solve for VL using both IL roots
%Select true VL and IL based on most reasonable VL_result
```

```
VL_result = n*a*(TH(2)-TC_result)-R_batt(2)*IL_result;
VL_true = VL_result(1); %[V]
IL_true = IL_result(1); %[A]

%% Results
P_tot = PL_result + Wdot_result;
eff_tot = P_tot/(P*A_pv);

col = [rc; eff_pv_result; eff_result; eff_tot; P_tot; T_pv_final];
disp(rc);
results = [results col];

end
toc; %Find program runtime
```

Task6a_Results.m

```
%% Results for Task6a
% ErrorEQ > 1e-5 for rc = 14, 15, 16, 17, 18
n_10 =    [10.0000   11.0000   12.0000   13.0000   14.0000   15.0000   16.0000
17.0000   18.0000; ...
    0.1202    0.1323    0.1442    0.1553    0.1607    0.1545    0.1386    0.1246
0.1064; ...
    0.1787    0.1869    0.1944    0.2014    0.2086    0.2161    0.2257    0.2326
0.2427; ...
    0.2774    0.2945    0.3106    0.3254    0.3357    0.3361    0.3331    0.3253
0.3233; ...
   29.9642   34.9824   40.2548   45.6916   50.7637   54.4531   57.5559   59.7338
62.8491; ...
  217.8808  225.8618  233.3652  240.5334  248.0469  256.2500  267.0532  275.0000
287.1887];

% ErrorEQ > 1e-5 for rc = 15, 16, 17, 18
n_11 =    [10.0000   11.0000   12.0000   13.0000   14.0000   15.0000   16.0000
17.0000   18.0000; ...
    0.1202    0.1323    0.1443    0.1562    0.1674    0.1735    0.1694    0.1543
0.1383; ...
    0.1594    0.1672    0.1743    0.1809    0.1870    0.1932    0.1991    0.2085
0.2167; ...
    0.2605    0.2774    0.2935    0.3088    0.3231    0.3332    0.3327    0.3303
0.3251; ...
   28.1343   32.9501   38.0318   43.3607   48.8504   53.9765   57.4861   60.6372
63.1910; ...
  201.3451  208.4714  215.1734  221.4920  227.5416  233.9063  240.0000  250.0000
259.1296];

% ErrorEQ > 1e-5 for rc = 16, 17, 18
n_12 =    [10.0000   11.0000   12.0000   13.0000   14.0000   15.0000   16.0000
17.0000   18.0000; ...
    0.1202    0.1323    0.1443    0.1563    0.1682    0.1793    0.1848    0.1788
0.1693; ...
    0.1429    0.1502    0.1569    0.1631    0.1688    0.1742    0.1799    0.1866
0.1912; ...
    0.2459    0.2626    0.2786    0.2939    0.3086    0.3223    0.3314    0.3321
0.3245; ...
   26.5602   31.1953   36.1005   41.2637   46.6667   52.2054   57.2727   60.9728
63.0830; ...
  187.9248  194.3355  200.3729  206.0656  211.4460  216.6275  222.1973  229.0180
233.7500];

n_13 = [10.0000   11.0000   12.0000   13.0000   14.0000   15.0000   16.0000   17.0000
18.0000; ...
    0.1202    0.1323    0.1443    0.1563    0.1683    0.1802    0.1909    0.1948
0.18755; ...
    0.1285    0.1354    0.1417    0.1476    0.1530    0.1580    0.1628    0.1681
0.1744; ...
    0.2333    0.2498    0.2656    0.2808    0.2956    0.3097    0.3226    0.3301
0.3292; ...
   25.1997   29.6733   34.4199   39.4287   44.6891   50.1791   55.7450   60.6080
64.0061; ...
  176.8802  182.6840  188.1577  193.3248  198.2065  202.8323  207.3425  212.4011
218.6262];

n_14 =    [10.0000   11.0000   12.0000   13.0000   14.0000   15.0000   16.0000
17.0000   18.0000;...
    0.1202    0.1323    0.1443    0.1563    0.1683    0.1803    0.1921    0.2018
0.2029;...
```

```
    0.1161    0.1226    0.1285    0.1340    0.1391    0.1439    0.1483    0.1526
0.1577;...
    0.2224    0.2386    0.2543    0.2694    0.2840    0.2983    0.3119    0.3237
0.3286;...
   24.0184   28.3475   32.9518   37.8210   42.9461   48.3169   53.8990   59.4229
63.8815;...
  167.6819  172.9659  177.9562  182.6727  187.1328  191.3529  195.3672  199.3864
204.1656];

%ErrorEQ is above 1e-5 for rc = 10
n_15 =    [10.0000   11.0000   12.0000   13.0000   14.0000   15.0000   16.0000
17.0000   18.0000;...
    0.1202    0.1323    0.1443    0.1563    0.1683    0.1804    0.1923    0.2038
0.2115;...
    0.1053    0.1113    0.1169    0.1221    0.1269    0.1314    0.1355    0.1395
0.1436;...
    0.2129    0.2289    0.2443    0.2593    0.2739    0.2880    0.3018    0.3149
0.3247;...
   22.9880   27.1878   31.6642   36.4077   41.4097   46.6622   52.1534   57.8127
63.1251;...
  159.9415  164.7767  169.3491  173.6756  177.7710  181.6484  185.3226  188.8478
192.5763];

%ErrorEQ is above 1e-5 for rc = 10
n_16 =    [10.0000   11.0000   12.0000   13.0000   14.0000   15.0000   16.0000
17.0000   18.0000;...
    0.1202    0.1323    0.1443    0.1563    0.1683    0.1804    0.1924    0.2043
0.2150;...
    0.0958    0.1014    0.1067    0.1116    0.1161    0.1203    0.1243    0.1280
0.1315;...
    0.2045    0.2203    0.2356    0.2504    0.2649    0.2790    0.2927    0.3061
0.3182;...
   22.0862   26.1694   30.5307   35.1608   40.0513   45.1952   50.5849   56.1994
61.8623;...
  153.3684  157.8134  162.0218  166.0083  169.7856  173.3648  176.7564  179.9784
183.1436];

%ErrorEQ is above 1e-5 for rc = 10, 17
n_17 =    [10.0000   11.0000   12.0000   13.0000   14.0000   15.0000   16.0000
17.0000   18.0000;...
    0.1202    0.1323    0.1443    0.1563    0.1683    0.1804    0.1924    0.2044
0.2160;...
    0.0904    0.0927    0.0976    0.1022    0.1065    0.1105    0.1142    0.1177
0.1210;...
    0.2035    0.2127    0.2278    0.2426    0.2569    0.2709    0.2846    0.2980
0.3109;...
   21.9740   25.2717   29.5293   34.0568   38.8465   43.8915   49.1855   54.7192
60.4370;...
  150.0000  151.8448  155.7343  159.4226  162.9206  166.2379  169.3832  172.3662
175.2221];

%ErrorEQ is above 1e-5 for rc = 10, 15
n_18 =   [10.0000   11.0000   12.0000   13.0000   14.0000   15.0000   16.0000
17.0000   18.0000;...
    0.1205    0.1323    0.1443    0.1563    0.1683    0.1804    0.1924    0.2044
0.2163;...
    0.0801    0.0850    0.0896    0.0940    0.0980    0.1018    0.1053    0.1086
0.1117;...
    0.1909    0.2060    0.2210    0.2356    0.2498    0.2638    0.2774    0.2908
0.3038;...
   20.6186   24.4775   28.6415   33.0762   37.7744   42.7296   47.9358   53.3870
59.0631;...
```

```
  142.8893  146.6921  150.3006  153.7258  156.9772  160.0630  162.9910  165.7681
168.4083];

nVec = linspace(10,18,9);
rcSelect14 = [n_14(4,1);...
    n_14(4,2);...
    n_14(4,3);...
    n_14(4,4);...
    n_14(4,5);...
    n_14(4,6);...
    n_14(4,7);...
    n_14(4,8);...
    n_14(4,9)];

effVec = [n_10(4,5);...
    n_11(4,5);...
    n_12(4,5);...
    n_13(4,5);...
    n_14(4,5);...
    n_15(4,5);...
    n_16(4,5);...
    n_17(4,5);...
    n_18(4,5)];

powerVec = [n_10(5,5);...
    n_11(5,5);...
    n_12(5,5);...
    n_13(5,5);...
    n_14(5,5);...
    n_15(5,5);...
    n_16(5,5);...
    n_17(5,5);...
    n_18(5,5)];

T_pvVec = [n_10(6,5);...
    n_11(6,5);...
    n_12(6,5);...
    n_13(6,5);...
    n_14(6,5);...
    n_15(6,5);...
    n_16(6,5);...
    n_17(6,5);...
    n_18(6,5)];


%% Plot Results

subplot(2,2,1);
plot(nVec,rcSelect14);
title("Efficiency at n = 14 over r_c");
xlabel("r_c");
subplot(2,2,2);
plot(nVec,effVec);
title("Efficiency");
xlabel("n");
subplot(2,2,3);
plot(nVec,powerVec);
title("Power [W]");
xlabel("n");
subplot(2,2,4);
plot(nVec,T_pvVec);
title("T_p_v [K]");
xlabel("n");
```

Task6b.m

```
%% Task 6(b) Problem Statement
% Modify the program for Task 4 to eliminate the PV cell and deliver the
% solar energy directly to the TE module. This means that in calculating
% errorEQ, Q_pv is instead Q_sun! Q_sun = 0.95*P"*A_pv = 0.95*rc*ID*A_pv
tic;
tol = 1e-5;
%% Initialize Constants

T_pv = [160, 0, 240];

ID = 1080; %[W/m^2]
a = 0.0017; %[V/K], from Task 3
lA = 0.032; %[W/(cm K)], from Task 3
lB = 0.021; %[W/(cm K)], from Task 3
pA = 0.0020; %[ohm cm], from Task 3
pB = 0.0030; %[ohm cm], from Task 3
A_cont = 300; %[W/K]
Rs = 0; %[ohm]
Vg = 1.1; %[V]
rc = 15; %unitless
Ly = 0.1; %[m] from 10 cm
Lz = 0.1; %[m] from 10 cm
A_pv = Ly*Lz; %[m^2]
RL_pv = 0.0070; %[ohm]
RL_te = 0.10; %[ohm]

T_source=6000; %Temperature of the sun (source)
P = rc*ID; %[W/m^2]

Tsat = 90.2; %[K]
n = 12; %unitless
Uev = 25; %[W/(m^2*K)]
tw = 0.004; %[m], 4 mm
k_wEff = 6.5; %[W/(mK)]

%% I0_IV Curve (Task 1 for PV Cell Task 2)
% Data Input & Linear Curve Fit
data = [28.85, 0.379; 30.492, 0.3737; 43.279, 0.3526; 50.49, 0.3316;...
    60.65, 0.311];
x = data(:,1)+273.15; %convert oC to K
y = data(:,2);
X = [ones(length(x),1), x];
b = X\y;
% Plot Fit
Voc = @(t) b(2).*t + b(1);
%For Kelvin
kb = 1.3807*10^-23; %[J/K]
qe = 1.602*10^-19; %[C]
I0_IV = @(t) (exp(Voc(t).*qe./(kb.*t))-1).^-1;

%% Step 2: Task 2 (PV Cell Analysis)
phi = @(T_source) (pi^4*kb*T_source./(2.404*15*P)).^-1;
sigma = @(x) x.^2./(exp(x)-1);
X = qe*Vg./(kb.*T_source);
integ = integral(sigma, X, Inf);
phi_i = phi(T_source);
phi_g = phi_i*0.416*integ;
IV = phi_g.*qe*A_pv;
i = 1;

errorEQ = [1000 1000 1000];
```

```
while abs(errorEQ(2))>tol || (errorEQ(1)*errorEQ(3)<0)
    midT_pv = (T_pv(1)+T_pv(3))/2;
    T_pv(2) = midT_pv;

    syms IL
    eqn1 = kb.*T_pv(1)./qe.*log(I0_IV(T_pv(1)).^-1-I0_IV(T_pv(1)).^-1.*IL./IV+1) ==
IL.*RL_pv;
    eqn2 = kb.*T_pv(2)./qe.*log(I0_IV(T_pv(2)).^-1-I0_IV(T_pv(2)).^-1.*IL./IV+1) ==
IL.*RL_pv;
    eqn3 = kb.*T_pv(3)./qe.*log(I0_IV(T_pv(3)).^-1-I0_IV(T_pv(3)).^-1.*IL./IV+1) ==
IL.*RL_pv;
    IL_op(1) = double(solve(eqn1,IL));
    IL_op(2) = double(solve(eqn2,IL));
    IL_op(3) = double(solve(eqn3,IL));

    VL = @(IL, t) kb*t/qe.*log(I0_IV(t).^-1-I0_IV(t).^-1.*IL./IV+1);
    VL_op = VL(IL_op,T_pv);

    PL = IL_op.*VL_op; %DELIVERABLE: POWER OUTPUT
    eff_pv = PL/(P*A_pv);

    Q_sun = 0.95*rc*ID*A_pv;%P*A_pv-PL; %Q_sun now acts as Q_pv
    Qplot(i) = Q_sun;
%% Step 3: Task 3 (TE Module Analysis)

    TH = T_pv(2) - Q_sun/A_cont;
    RA_min = (sqrt(lA*pA)+sqrt(lB*pB))^2;
    Z = a^2/RA_min;

    error = [1000 1000 1000]; %initialize error
    TC = [90 0 120]; %LB Midpoint UB

    while abs(error(2))>tol || (error(1)*error(2)<0)
        midTC = (TC(1)+TC(3))/2;
        TC(2) = midTC;

        m_maxEff = sqrt(1+0.5.*(TH+TC).*Z);
        R = RL_te./(n.*m_maxEff);
        A = RA_min./R;

        % (ii) Computing QC_boil
        QC_boil = (k_wEff/tw + Uev)*(TC-Tsat)*A_pv; %[W]

        % (iii) Compute efficiency using Eq. 11
        eff = (TH-TC)./TH.*((1+m_maxEff).^2./(m_maxEff.*Z.*TH) + 1 +
1./(2*m_maxEff).*(1+TC./TH)).^-1;

        % (iv)  Computer power output Wdot using Eq. 10
        R_batt = n*R;
        Wdot = (n^2*a^2.*(TH-TC).^2.*RL_te)./(RL_te+R_batt).^2; %[W]

        % (v) Compute heat input and rejection
        QH = Wdot./eff; %[W]
        QC_TE = QH - Wdot; %[W]

        % (vi) Computer error - run if statement to find optimal TC and QH for
        %       minimal error
        error = QC_TE-QC_boil;
        if error(1)*error(2)<0
            TC(3) = TC(2);
        elseif error(1)*error(2)>0 && error(1)>0
            TC(1) = TC(2);
```

```
        elseif error(1)*error(2)>0 && error(1)<0
            TC(3) = TC(2);
        end
    end

    error_result = error(2);
    TC_result = TC(2);
    QH_result = QH(2);
    QC_result = QC_boil(2);
    A_batt = n*A(2);
    Wdot_result = Wdot(2);
    eff_result = eff(2);
    QHplot(i) = QH_result;
%% Step 4: Compute Error EQ
% Power, Operating Point, PV cell efficiency

    errorEQ = Q_sun-QH;%_result;
    if errorEQ(1)*errorEQ(2)<0
        T_pv(3) = T_pv(2);
    elseif errorEQ(1)*errorEQ(2)>0 && errorEQ(1)>0
        T_pv(1) = T_pv(2);
    elseif errorEQ(1)*errorEQ(2)>0 && errorEQ(1)<0
        T_pv(3) = T_pv(2);
    end

    PL_result = PL(2);
    Q_pv_result = Q_sun;
    eff_pv_result = eff_pv(2);
    errorEQ_result = errorEQ(2);
    IL_pv_result = IL_op(2);
    VL_pv_result = VL_op(2);
    T_pv_final = T_pv(2);
    TH_result = TH;
    T_pvplot(i) = T_pv_final;
    i= i +1;
end

syms IL_te
heatRateEqn = QH_result == n*a*IL_te*TH + A_batt*(TH-TC_result)-0.5*IL_te^2*R_batt(2);

IL_result = double(solve(heatRateEqn, IL_te));
VL_result = n*a*(TH-TC_result)-R_batt(2)*IL_result;
VL_true = VL_result(1); %[V]
IL_true = IL_result(1); %[A]

%% Plots
%Error plot
plot(T_pvplot, Qplot)
hold on;
plot(T_pvplot, QHplot);
legend("Q_{sun}","Q_{H,te}",'Location','southeast');
xlabel("T_{pv} [K]");
ylabel("Heat Transfer [W]");

%% Results
P_tot = PL_result + Wdot_result;
eff_tot = P_tot/(P*A_pv);

fprintf("PV Cell Results: \n" ...
    + "    Operating Point: VL = " + VL_pv_result + " V, IL = " + IL_pv_result  + " A
\n" ...
    + "    Operating Point Temperature: T_pv = " + T_pv(2) + " K \n" ...
    + "    Power = " + PL_result + " W \n" ...
```

```
    + "    Efficiency = " + eff_pv_result + "\n \n");

fprintf("Thermoelectric Module Results: \n" ...
    + "    Operating Point: VL = " + VL_op(2) + " V, IL = " + IL_op(2)  + " A \n" ...
    + "    Power Output: Wdot = " + Wdot_result + " W \n" ...
    + "    Efficiency = " + eff_result + "\n \n");

toc; %Find program runtime
```