The Design and Analysis of A Solar Hybrid Power

Deep Dayaramani
Audrey Zhao
Mechanical Engineering 146
Professor Carey
October 25, 2018

Introduction

In today‘s rapidly development of science and technology, tremendous inventions have been combined with others to achieve better performance. A solar hybrid power is one of the most common examples. It’s a system of the concentrating photovoltaic (PV) that recycles and converts waste heat to power. Originally, this solar hybrid system was proposed as a means of completing the travel from the Earth to the moon for a spacecraft. And the power is intended to vaporize stored liquid oxygen and generate electric power to sustain both the crew and spacecraft systems. A detailed design of the system is shown below in Fig. 1.

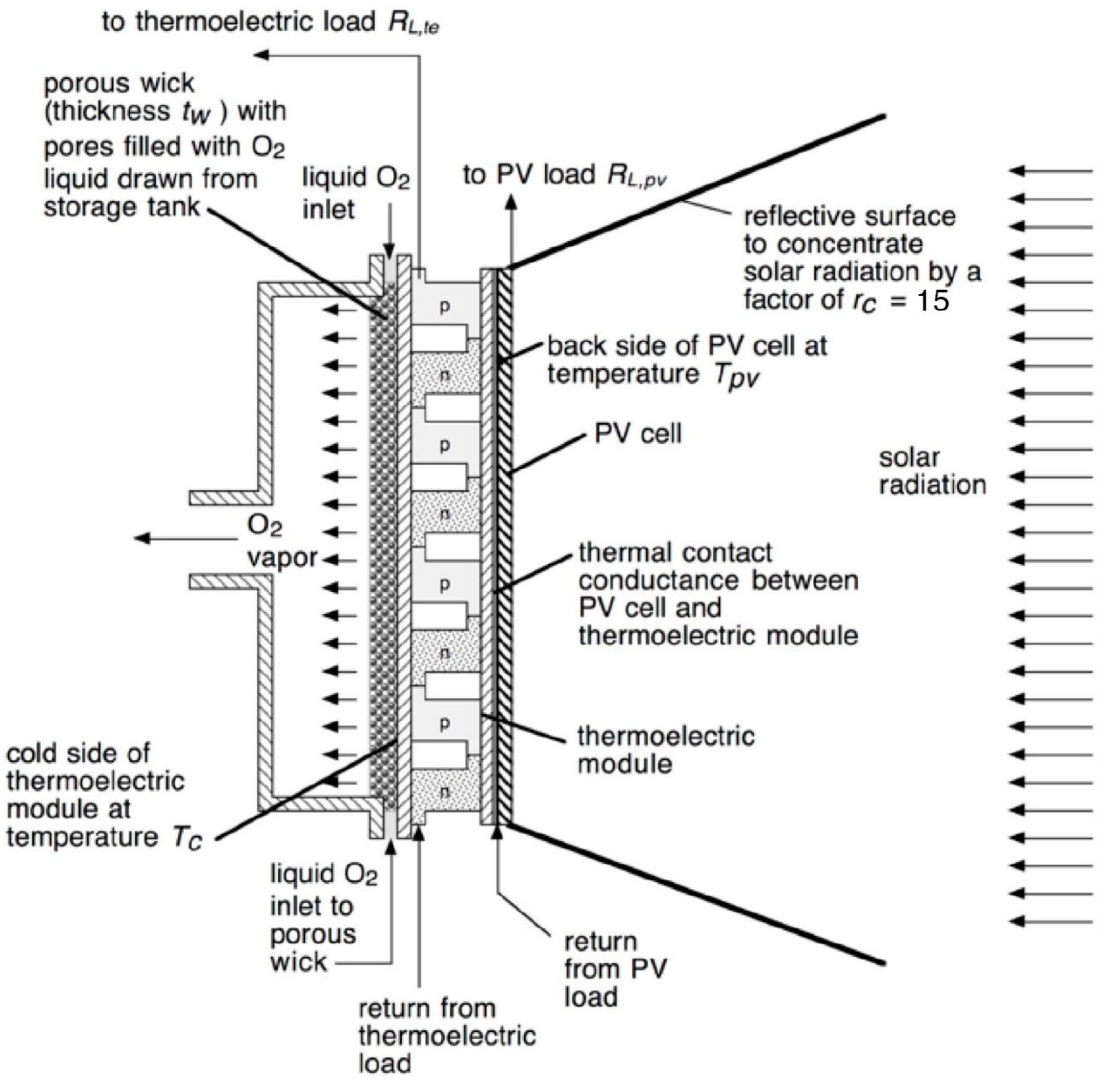


Figure 1. *The illustration of a solar hybrid power generation system.*

The vaporized liquid oxygen is used to cool the silicon back side of the thermoelectric module at a pressure of 163 kPa. On the back side of the PV cell, it’s connected to the hot surface of the thermoelectric model. And the thermal conductance can be evaluated using Eq. (1).

$$\Lambda_{cont}(= A_{cont}\lambda_{eff} / \ell_{eff}) \text{ is } 300 \text{ W/K} \tag{1}$$

Further, the PV cell is designed to be 10 cm by 10 cm and placed on a support frame that as a normal surface always at the sun during the spacecraft travel. Its surface is normal to the sun's rays and regulated by a control system. The ratio of the light power density to the photon flux can be found using Eq. (2). Moreover, P'', the solar intensity incident, is also a function of the unconcentrated direct solar intensity I_D of the surface and the concentration ratio r_c ($P''=r_cI_D$ in W/m^2). Initially, r_c was set to be 15 and P'' was found to be 1080 W/m^2. The total flux of photons is calculated by integrating the photon flux with corresponding limits, shown in Eq.(3). Additionally, for a system that has a real diode, it also has an internal resistant that makes the load voltage decrease and can be measured by Eq.(4). Lastly, the values of the useful constants were also listed below.

$$\frac{P''}{\phi} = \frac{\pi^4 k_B T_{source}}{2.404(15)} \tag{2}$$

$$\frac{\phi_g}{\phi} = 0.416\int_X^{\infty} \frac{x^2}{e^x - 1}dx \ , \quad \text{where } X = \frac{W_g}{k_B T_{source}} = \frac{hf_g}{k_B T_{source}} = \frac{q_e V_g}{k_B T_{source}} \tag{3}$$

$$V_L = \frac{k_B T_{pv}}{q_e} \ln\left[\frac{I_v - I_L}{I_0} + 1\right] - I_L R_s \quad \text{where} \quad I_v = \phi_g q_e A_{pv}, \quad A_{pv} = L_y L_z \tag{4}$$

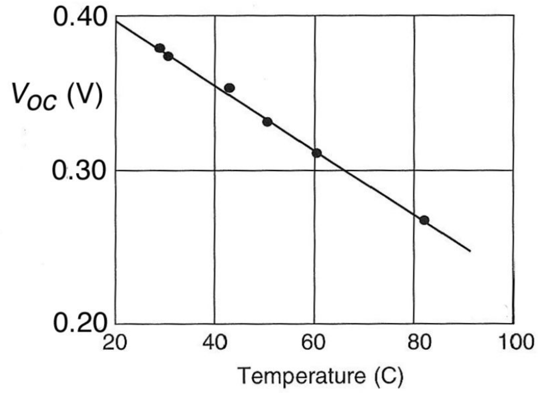
K_b= 1.38e-23 m^2 kg s^{-2} K^{-1}
q_e= 1.6e-19 C

The ultimate purpose of this project is to analyze the performance of the solar hybrid power generation system under different critical conditions.

Task 1

Appendix I

The task was consisted of 2 steps. The first one was to establish the a linear relationship between the PV cell temperature and the the open-circuit voltage. A total of 5 pairs of coordinates for both variables was approximated using the provided graph (Fig.2) and shown in the Table 1, and the coefficients were found by the linear curve-fit. The equation was found to be V_{OC} = $-0.0019T_{PV}+0.9587$. The next step was to calculate the ratio I_0/I_V as a function of temperature between 20°C and 100°C.Plugging into Eq. (4), I_L was given to be 0 and V_{oc} was now only in terms of temperature. Therefore, the ratio could be isolated and measured based on temperature and other constants given. **At 20°C, the value of I_0/I_v was measured to be -1.0614; and at 100°C, it was -1.0051.**



Observed temperature variation of the open-circuit voltage.

Figure 2. *The schematic of the temperature and open-circuit voltage observed*

T_{PV}(K)	V_{OC}(V)
303	0.39
0.37	303
0.35	313
0.32	333
0.27	353

Table 1. *Approximated values of temperature and open-circuit voltage*

Task 2

Appendix II

In this task, it was first suggested that the operating point and the power generated for the load can be found by solving Eq.(4) and Ohms law at the same time:

$$V_L = I_L R_{L,pv} \tag{5}$$

Therefore, the goal of this step was to establish the power of the PV ell as a function of $R_{L,PV}$ and other parameters from Eq.(2)-(4) and to calculate the corresponding conversion efficiency using Eq.(6):

$$\eta_{pv} = V_L I_L / (P'' A_{pv}) \tag{6}$$

It's known that the PV cell was 10 cm by 10 cm; and P" was calculated to be the product of r_c = 15 and I_D = 1080 W/m^2. Other variables given were listed below:

$R_{L,pv} = 0.0070\ \Omega$ $\quad$ $R_s = 0$

$V_g = 1.1$ V, $f_g = 265$ THz (for silicon) $\quad$ T_{pv} = 20°C and 80°C

Firstly, the steps from Task 1 were incorporated to set the ratio of I_0/I_v as a function of T. Based on Eq.(2), the photon flux was found to be 7.5212e+22. Similarly, the total flux was evaluated to be 4.0378e+22 using Eq.(3) using the variables solved and given. I_v and I_0 were further isolated and measured by plugging into Eq.(4) and the ratio of I_0/I_v. Now, I_L remained to be the only unknown variable in Eq.(4) and was solved by adopting the "solve" function in matlab. Once I_L was found, it was then used to calculate the power, which was the product of I_L and V_L, and the efficiency using Eq.(6). **Hence, at T= 20°C, P=17.7292W, η=0.1094, Q_{PV}= 144.2708, V_L=0.3523V, and I_L=50.3263A. When T= 100°C, P=8.9009W, η=0.0549, Q_{PV}= 153.0991, V_L=0.2496V, and I_L=35.6589A.**

Task 3

Appendix III

The heat input temperature was defined to be T_H, and T_{pv} corresponded to the temperature of the back side of the PV cell. They can be linked using the transfer contact conductance equation (Eq.(7)) between the PV cell surface and the thermoelectric module. Moreover, Q_{pv} was the waste heat transfer rate. At steady state, the rate of heat input to the thermoelectric module Q_H had the same value as Q_{PV}.

$$\dot{Q}_{pv} = \Lambda_{cont}(T_{pv} - T_H), \qquad \Rightarrow \quad T_H = T_{pv} - \dot{Q}_{pv} / \Lambda_{cont} \tag{7}$$

For the thermoelectric module, its cold surface converted the absorbed heat to energy to vaporize liquid oxygen, where the liquid was drawn into the nickel wick due to capillary force. The wick was 4.0 mm thick (t_w = 4.0 mm). And the effective thermal conductivity of the wick filled up liquid oxygen was listed in Eq.(8).

$$k_{w,eff} = 6.5 \text{ W/mK} \tag{8}$$

Like the load resistance, there's a contact resistance existing at the intersurface between the thermoelectric module and the evaporator. It was a constant and can be found in Eq.(9).

$$U_{ev} = 25 \text{ W/m}^2\text{K}. \tag{9}$$

Therefore, the heat transfer on the surface can be modeled as:

$$\frac{\dot{Q}_C}{A_{pv}} = \left[\frac{k_{w,eff}}{t_w} + U_{ev}\right](T_C - T_{sat}) \tag{10}$$

The term T_{sat} was the saturation temperature of liquid oxygen at atmospheric pressure. And it's calculated to be 95.0 K at 163 kPa. Also, the thermoelectric unit employed the new quantum dot superlattice material, which the figure of merit Z was reported to be 0.0114 K^{-1} between 20°C and 100°C. The average Seebeck coefficient α was given to be 0.0017 V/K. The other related thermoelectric properties and useful equations mentioned in the lectures were also listed below:

$$\begin{aligned}
&\text{Electric resistivity of arm A: } \rho_A = 0.0020 \ \Omega \text{ cm} \\
&\text{Electric resistivity of arm B: } \rho_B = 0.0030 \ \Omega \text{ cm} \\
&\text{Thermal conductivity of arm A: } \lambda_A = 0.032 \text{ W/(cm K)} \\
&\text{Thermal conductivity of arm B: } \lambda_B = 0.021 \text{ W/(cm K)}
\end{aligned} \tag{11}$$

Single pair electrical resistance and heat conductance:

$$R = \frac{\ell_A \rho_A}{A_A} + \frac{\ell_B \rho_B}{A_B} \quad , \quad \Lambda = \frac{A_A \lambda_A}{\ell_A} + \frac{A_B \lambda_B}{\ell_B} \tag{12}$$

The number of thermoelectric pairs in the battery was designated n. The total battery electrical resistance was shown below, where R was the resistance of each pair:

$$R_{batt} = nR \tag{13}$$

The total thermal conductance in this parallel setup was calculated to be:

$$\Lambda_{batt} = n\Lambda \tag{14}$$

The load voltage with current flow from a battery with n paris was modeled as:

$$V_L = n\alpha(T_H - T_C) - R_{batt} I_L \tag{15}$$

Rate of heat transferred from the high temperature source:

$$P_H = \dot{Q}_{pv} = \dot{Q}_H = n\alpha I T_H + \Lambda_{batt}(T_H - T_C) - \tfrac{1}{2} I^2 R_{batt} \tag{16}$$

Power output:

$$\dot{W} = \frac{n^2 \alpha^2 (T_H - T_C)^2 R_{L,te}}{(R_{L,te} + R_{batt})^2} \tag{17}$$

Efficiency:

$$\eta = \frac{\dot{W}}{\dot{Q}_H} = \left[\frac{T_H - T_C}{T_H}\right]\left[\frac{(1+m)^2}{m}\left(ZT_H\right)^{-1} + 1 + \frac{1}{2m}\left(1 + \frac{T_C}{T_H}\right)\right]^{-1} \tag{18}$$

The variable m was defined as:

$$m = R_{L,te} / nR \tag{19}$$

The definition of the figure of merit was:

$$Z = \frac{\alpha^2}{R\Lambda} \tag{20}$$

The parameters that minimized RΛ:

$$(R\Lambda)_{\min} = \left[\left(\lambda_A \rho_A\right)^{1/2} + \left(\lambda_B \rho_B\right)^{1/2}\right]^2 \tag{21}$$

The value of m that maximized the efficiency:

$$m_{max\,eff} = \sqrt{1 + 0.5(T_H + T_C)Z} \tag{22}$$

The rate of heat input and rejection for the thermoelectric unit:

$$\dot{Q}_H = \dot{W} / \eta \qquad (\dot{Q}_C)_{TE} = \dot{Q}_H - \dot{W} \tag{23}$$

The error $E_{QC:}$

$$E_{\dot{Q}_C} = \dot{Q}_C)_{TE} - (\dot{Q}_C)_{boil} \tag{24}$$

The goal of this task was to find the value of $(R\Lambda)_{min}$, the figure of merit Z based on $(R\Lambda)_{min.}$, the low temperature source T_c, the power output W, rate of heat transfer from high temperature unit $Q_{H,}$ the opposite heat transfer rate Q_C, the current load $I_{L,}$ and the voltage across the load V_L. The given parameters were n =12, $R_{L,\,te}$ = 0.10Ω, and T_H = 50 °C.

Firstly, based on the inputs, $(R\Lambda)_{min}$ and Z can be simply calculated using Eq.(20) and (21). In order to find T_c, a bisection method was employed where the initial values and guess were T_{sat}, T_H, and their average, and the error was within the range of e-05 of the efficiency. More specifically, a series of functions were created to be used to figure out corresponding values at different temperatures. Based on Eq.(19) and (22) and Λ = (ΛR)min /R, $m_{max,\,eff}$, R, and Λ were isolated. Using Eq.(10), Q_c was found. Similarly, a function of efficiency can be created in terms of the guessed Tc value from Eq.(18). A power equation was established based on Eq.(17) as well. The heat input and output were measured by Eq.(23), and the error was calculated by Eq.(24). By now, all the functions needed to lead to the measurement of error were created. As for the bisection method, the while loop would keep running and narrowing down the range of temperatures based on the error calculated until the error was less than e-05. Once T_c was found, W, Q_H, Q_C, and η corresponded to same values of them in the bisection method. Similar as Task 2, I_L and V_L were found by adopting the "solve" function in terms of Eq. (8) and (9).

Therefore, according to the equations and parameters given, **$(R\Lambda)_{min}$ was 2.5400e-04, Z was 0.0114K^{-1}, T_C was 108.9611K, W was 80.6204W, Q_H was 310.9793W, Q_C was 230.3588W, I_L was 22.9536A, and V_L was 5.6008V.**

Task 4

Appendix IV

In part (a), after having create functions in Task 2 and 3 to calculate a range of factors necessary for the solar hybrid system, this step challenged students to combine these function to establish a general relationship that reflected the operating conditions for any parameter set. The inputs given included ID, α, λ_A, λ_B, ρ_A, ρ_B, Λ_{cont}, Rs, V_g , r_c, L_y, L_z, $t_{w,in}$, n, R_L, pv , and $R_{L,te}$. Similar as the previous task, the bisection method was adopted to find the desired conditions ny minimizing the error.

The initial range of T_{pv} was taken from 150K to 250K, and the guessed temperature was the average one. Using Task 2, the product of V_L and I_L was found and the waste heat transfer was calculated to be Eq.(25) in terms of T_{PV}:

$$\dot{Q}_{pv} = P''A_{pv} - V_L I_L \tag{25}$$

Then the T_H was created according to Eq.(7) with respect to different T_{PV}. Then from Task 3, the function of T_c and Q_H could be set up for specific T_H parameter. Further, in this case, the error E_Q was defined in Eq.(25) and its magnitude must be less than e-05:

$$E_{\dot{Q}} = \dot{Q}_{pv} - \dot{Q}_H \tag{26}$$

The while loop in the bisection method would then keep generating new values for T_{PV} and other corresponding parameters (introduced in Task 2 and 3) until the error fell in the appropriate range.

Once T_{PV} was found, other critical conditions for the PV cell performance can also be determined: the load voltage, load current, power output, efficiency of the PV cell and thermoelectric unit, the combined power output and efficiency of the hybrid system.

The combined power and efficiency of the hybrid system were determined using the formulas below:

$P_{TOT} = P_{PV} + P_{TE}$ & $\eta_{TOT} = \frac{P_{TOT}}{P''}$

For part (b), all the inputs required to determine T_{pv} and other conditions were provided below:

$I_D = 1080$ W/m^2

$r_c = 15$

$L_y = L_z = 10$ cm $(A_{pv} = L_y L_z)$

$V_g = 1.1$ V, $f_g = 265$ THz (for silicon)

$R_s = 0$

$R_{L,pv} = 0.0070\ \Omega$

values of $\alpha, \lambda_A, \lambda_B, \rho_A, \rho_B$ specified in Task 3

$\Lambda_{cont} = 300$ W/K

$T_{sat} = 90.2$ K

$n, = 12$

$R_{L,te} = 0.10\ \Omega$

After plugging all these values, it was found that **T_{PV}=212.4719K, T_H= 212.0283K, T_C=90.2662K, η_{PV} = 0.1785, η_{te} = 0.1797, P = 28.9193W, W=23.9110W, V_L=0.4499V, I_L= 64.2754A, η_{tot} = 0.3209 and P_{tot} = 51.9779.**

Task 5a

Appendix V

In this task we took the parameters in task 4 and decided to see how the efficiency and the power would vary with the difference in solar input radiation of the pv cell. These changes in turn affect the Temperature of the pv cell, thermoelectric module, the efficiency of both along with the power and voltages of the two. Task_4 provided us with the ability to calculate these factors with a guessed temperature of the photovoltaic cell which can then lead to the temperatures of the hot side and the cold side of the thermoelectric module. Task 4 was altered to generate task_5a, which has a for loop for changing values of I_d or incident radiation. There were 12 values of I_d chosen for the for loop between 720 W/m2 and 1080 W/m2. For these values of I_d, task 4 was run again and all the values were stored in a matrix. We then took the values for total efficiency, PV efficiency, thermoelectric efficiency and total power and plotted them against the changing values of I_d to see how they varied with the increasing values of I_d. The result is shown in the subplots below where we can see that all the values increase with increasing I_d. This shows that as the incident power from the sun becomes larger, the efficiency and power output of the setup increase which makes intuitive sense.

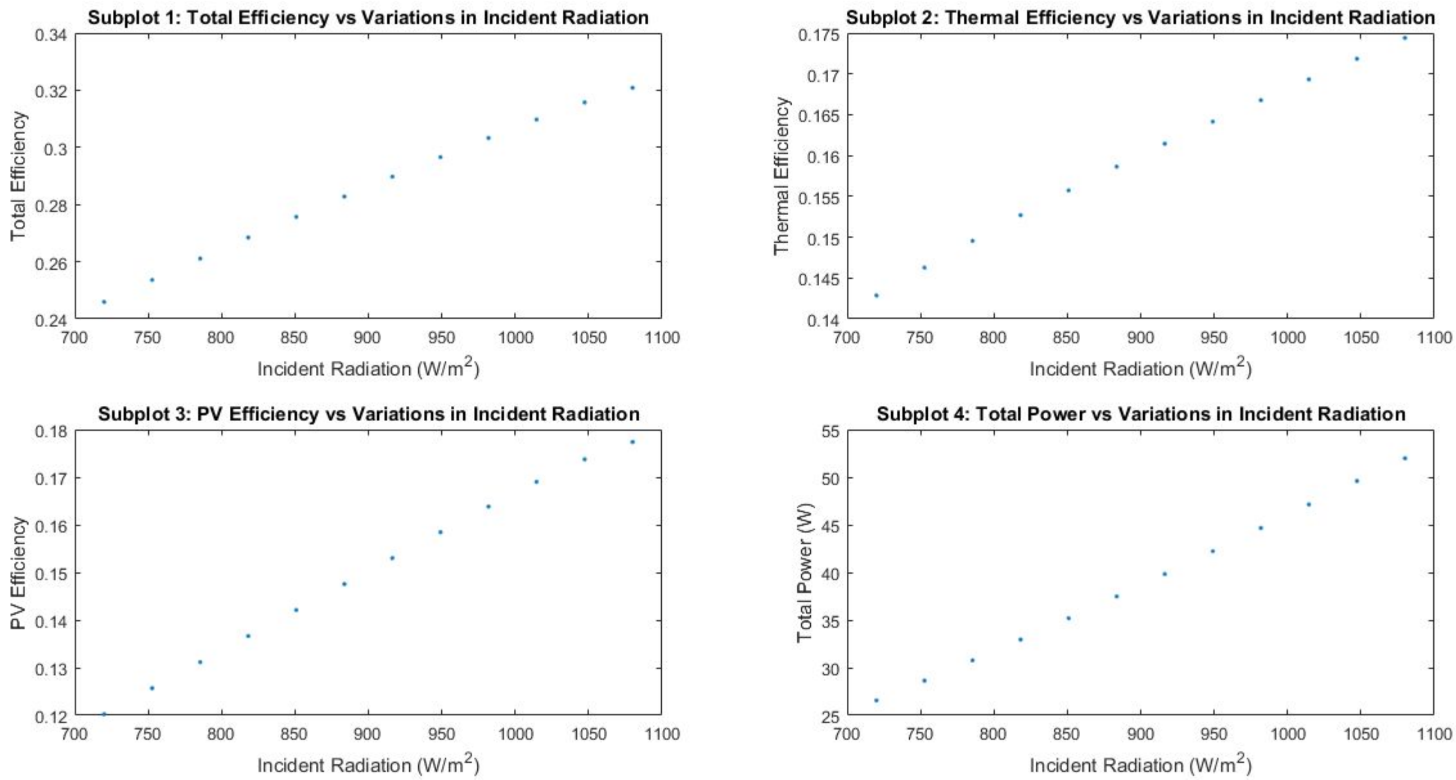


Figure 3*: Various subplots for the variations in efficiencies and power vs incident radiation*

Task 5b

Appendix VI

In this task we took the parameters in task 4 and decided to see how the efficiency and the power would vary with the difference in concentration ratios of the pv cell. These changes in turn affect the Temperature of the pv cell, thermoelectric module, the efficiency of both along with the power and voltages of the two. Task_4 provided us with the ability to calculate these factors with a guessed temperature of the photovoltaic cell which can then lead to the temperatures of the hot side and the cold side of the thermoelectric module. Task 4 was altered to generate task_5a, which has a for loop for changing values of rc or concentration ratios while keeping the incident radiation constant at 1080 W/m2. There were 9 values of rc chosen for the for loop between 10 and 18. For these values of rc, task 4 was run again and all the values were stored in a matrix. We then took the values for total efficiency, PV efficiency, thermoelectric efficiency and total power and plotted them against the changing values of rc to see how they varied with the increasing values of rc. The result is shown in the subplots below. This is to see the variation of all the measured values against changing concentration ratios. As we can see, increasing the concentration ratios, increases the thermal efficiency and total power output. But it results in a decrease in efficiency at high values of PV efficiency which in turn affects total efficiency. This might be because of the increase in temperature of the PV cell which reduces the efficiency. This makes intuitive sense.

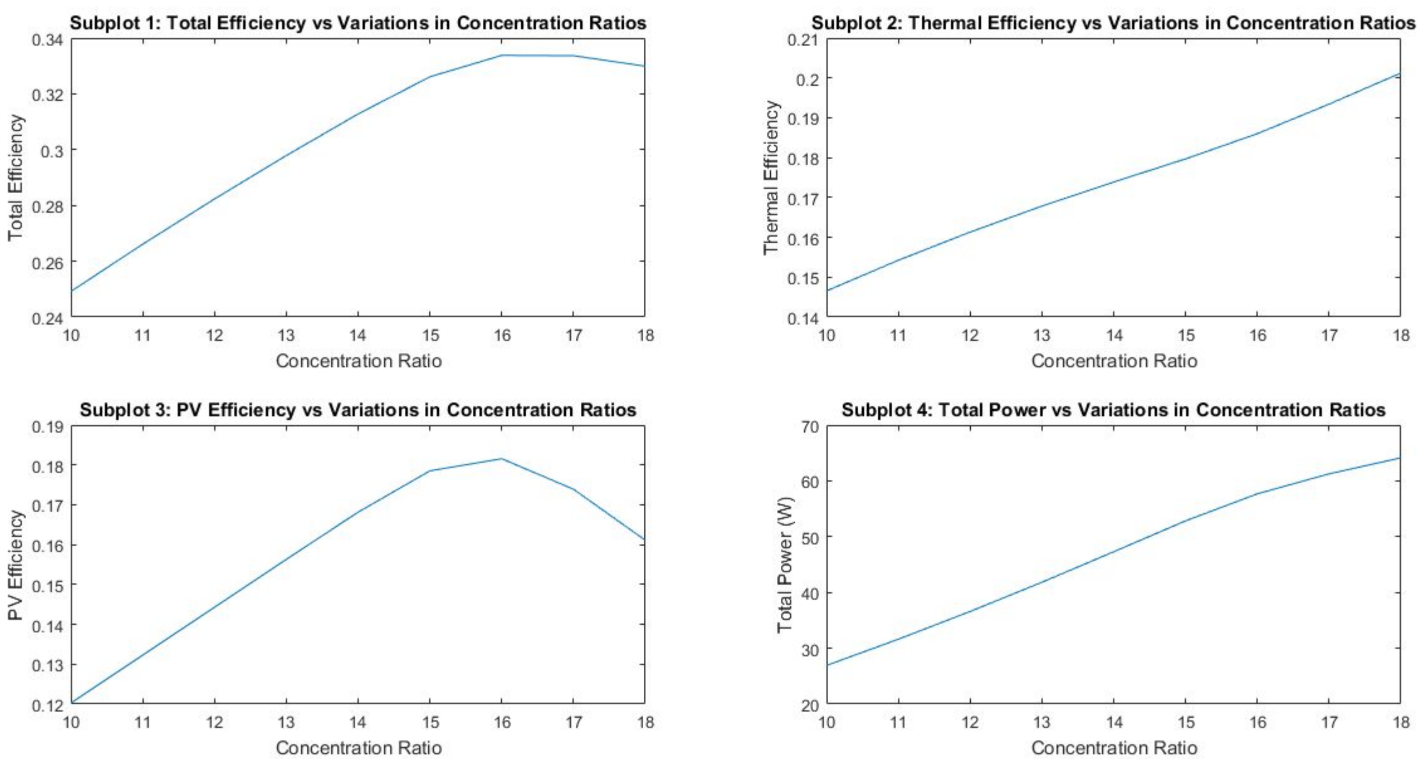


Figure 4*: Various subplots for the variation in efficiencies and power vs concentration ratios*

Task 6a

Appendix VII

In this task, we decided to analyse design choices of the whole setup by choosing two variables , the concentration ratios and the number of thermoelectric pairs. We then used this in our program to evaluate the performance of the system for different combinations of concentration ratio rc and number of thermoelectric pairs n, with α = 0.0017 V/K , the value for the superlattice material. We defined some design specifications that helped us choose the best rc and n values. The specifications are as defined below:

1. The design must have a total power output of 20 to 50 W (more is better).
2. Higher hybrid system conversion efficiency is better.
3. Smaller concentration ratio is better
4. Lower PV operating temperature is better

So we used a script similar to task_5 but now defined another for loop that iterated through the values of rc. For each value of rc, the second for loop iterated through a value for n. After this for loop iterated through the n's, it took a specified rc and n and used a script similar to task_4 to calculate the parameters for each flow. The script can be found below in Appendix **INSERT**. The script stores the values for all the parameters for differing rc's and n's. Looking at the different specifications, we can look at the fact that we need to measure Total power, total Efficiency and the temperature of the PV cell. For this we ran the script and then using the stored values of all the above parameters, we plotted them against rc in 2D plots as well as against rc and n in 3D plots. There are two 2D subplots one with lines and the other with points for purposes of clarity. The 2D plot with points is showing how to identify values with same n's and same rc's. The graphs have same colored points which correspond to same n vs the straight lines perpendicular to the x axis correspond to different values of rc or when u connect points of different colors in one line, those points will have the same rc.

P_tot vs rc vs n:

When we look at the points in the total power vs rc, 2D graph, we can see that for higher rc's we have lower variation in P_tot and it reaches its highest values for higher values of rc. When we look at the 3D plot we can see that it does start to narrow down to a limited set of values for higher values of rc. We can see that for increasing n, the lower limit of P_tot is lower but they reach nearly the same higher limit of P_tot ~ 62 -63W. This shows that high rc and n is good for higher P_tot. But to get between 20W and 50W, we should choose rc's between 12-15, as seen in the point 2D subplots and n between 11 and 14.

Total Efficiency vs rc vs n:

When we look at efficiency vs rc and n, in the 2D plots, we can see that efficiency increase linearly but their growth is suddenly stopped and they all accumulate to a small range of efficiencies ~0.32. Higher rc's, n's result in smaller efficiencies. Looking at the plots we can decide that choosing rc's and n between 11-15 and 9-13 would be the best choice for good total efficiency.

Temperature of the PV cell vs rc vs n:

We see that the temperature is lower for higher values of n and for higher values of n we can see that the temperature increases slowly with increasing rc. Hence higher values of n are mostly preferred and rc will depend on that as slope of increase depends on n, from figure **Insert**.

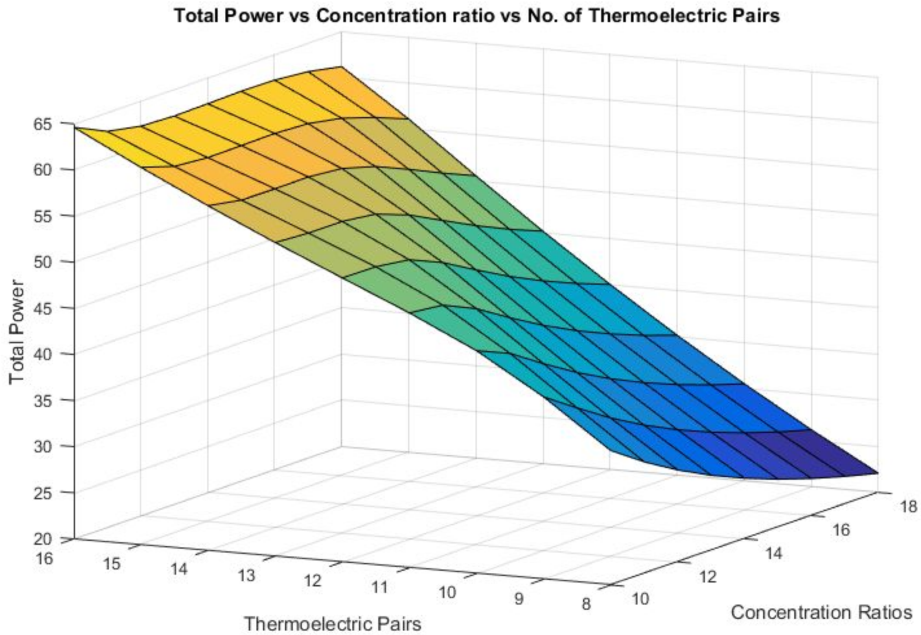


Figure 5*: Total Power vs Concentration ratios vs number of thermoelectric pairs*

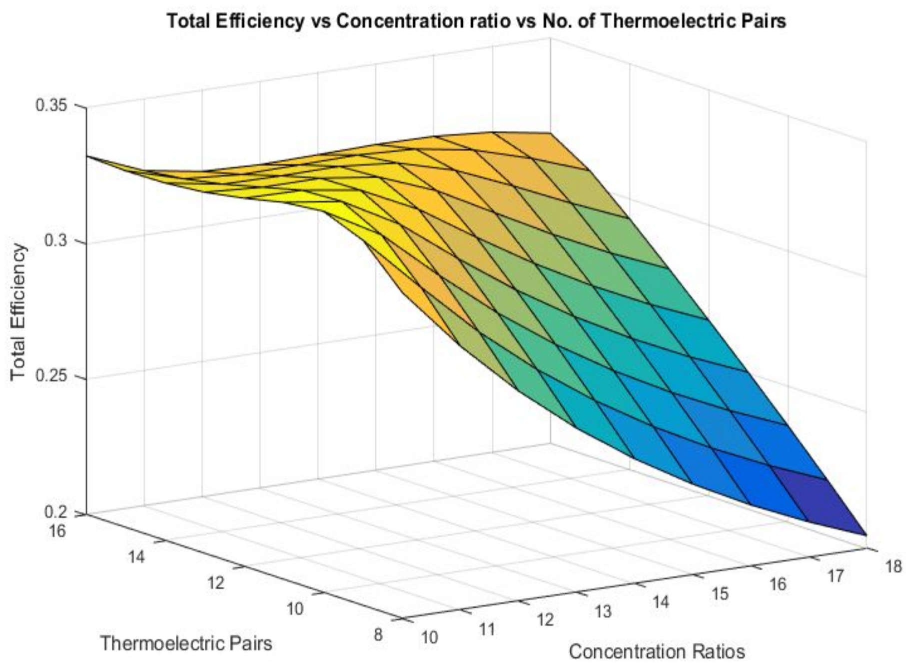


Figure 6*: Total Efficiency vs Concentration ratios vs number of thermoelectric pairs*

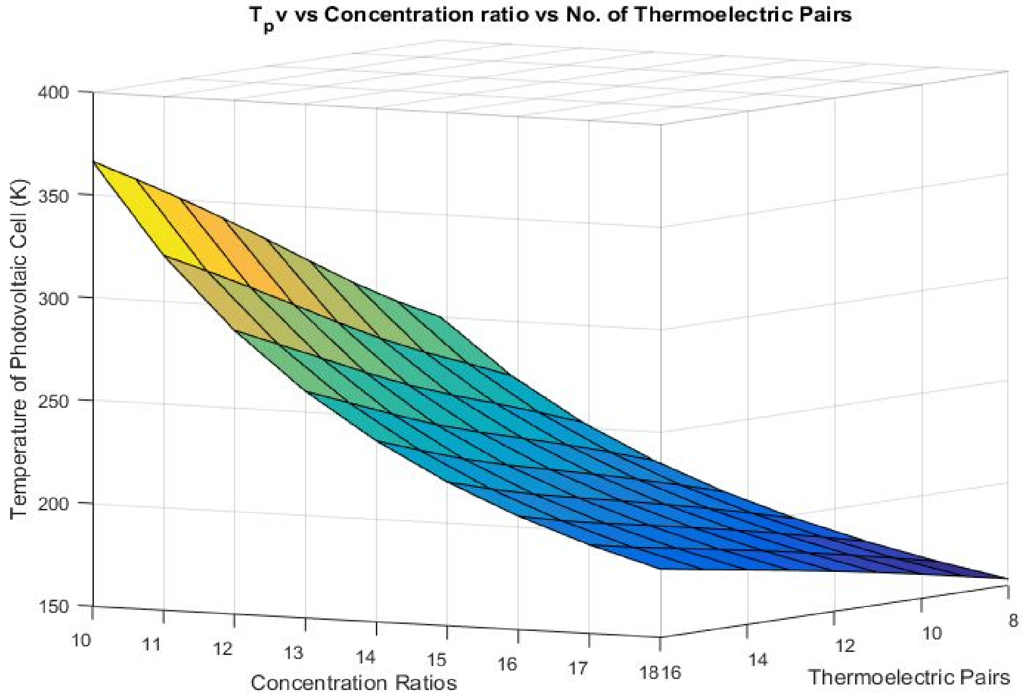


***Figure** 7: Temperature of the Photovoltaic Cell vs Concentration ratios vs number of thermoelectric pairs*

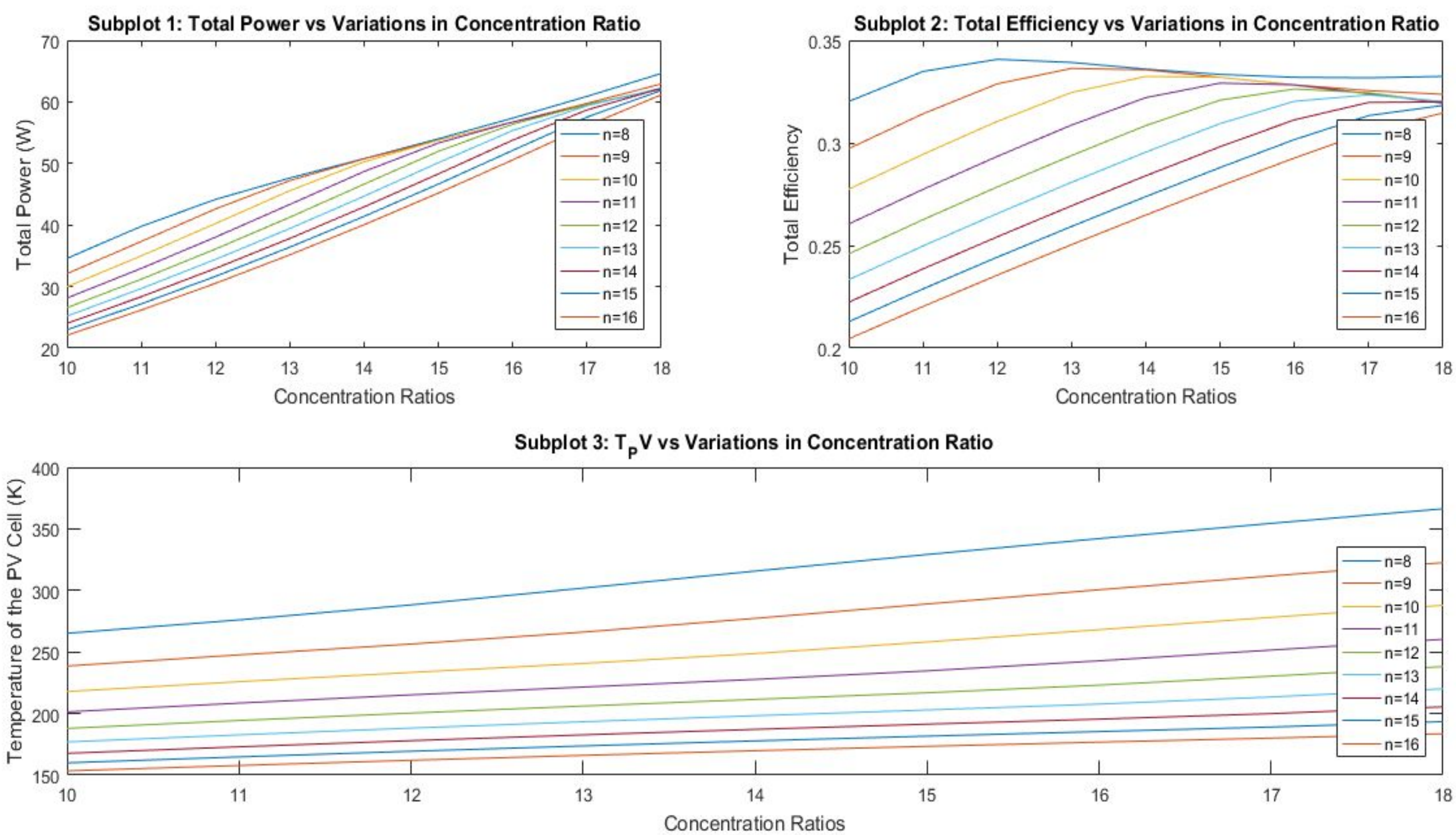


Figure 8(above): *Various subplots of the variation of Efficiency, Temperature and total power output vs concentration* *ratios.*

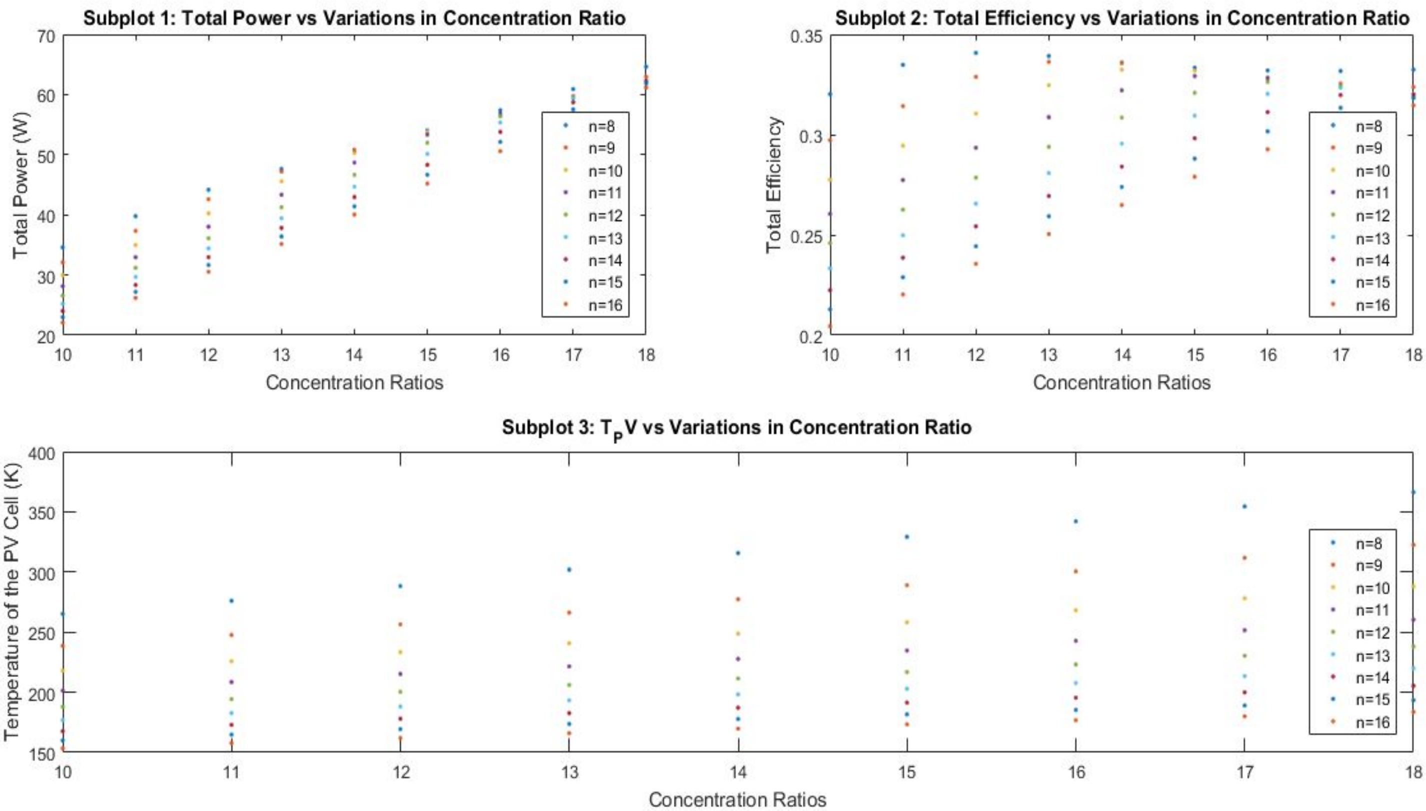


Figure 9: *Various subplots of the variation of Efficiency, Temperature and total power output vs concentration ratios.*

Choosing values:

For P_tot to be between 20W and 50W, we used a find function to record the indices of the positions where P_tot was in this range. Then we used that range and applied it to total efficiency matrix and the temperature of the PV cell matrix. Using these indices we also found their respective rc and n values. We combined all of these values in one matrix T_P_tot which contained information about P_tot, T_pv, rc, n and Eff_tot. Now we could look at the matrix and decide for our perfect design. We first looked for highest power values and their respective T_pv's. We found 3 choices:

1. P_tot = 48.7011W
   T_pv = 227.7106 K,
   n = 11,
   rc = 14,
   eff = 0.3221.
2. P_tot = 48.3057W
   T_pv = 191.3615 K,

n = 14,
rc = 15,
eff = 0.2982.

3. P_tot = 46.6586W
   T_pv = 181.6509 K,
   n = 15,
   rc = 15,
   eff = 0.2880.

Looking at these 3 choices, we can see that for maximizing P_tot and efficiency, we should choose 1, but for nearly equal Power but a much lower temperature, 2 is the best choice. If temperature is our most important choice while power and efficiency don't matter most, we can choose 3.

Our final choice would be between 1 and 2 though. Their power outputs are the closest to 50W and their efficiencies are pretty high. The only differentiating factor is the rc and the temperature of PV operation. The temperature is better for 2 while rc is better for 1. Hence any of these would work and it only depends on the prioritization of these specifications.

Final suggestions:

1. **n = 11, rc = 14 with P_tot = 48.7011W, T_pv = 227.7106 K and eff = 0.3221.**
2. **n = 14, rc = 15 with P_tot = 48.3057W, T_pv = 191.3615 K and eff = 0.2982.**

Task 6b

Appendix VIII

For this task we wanted to compare how the thermoelectric module would do on it's own. Hence we assumed that a proposal was made to eliminate the PV cell and deliver the solar energy directly to just the thermoelectric module. In this modified design, the concentrated solar energy would directly strike a highly absorbing nanocoating on the hot side of the thermoelectric unit, with 95% of the incident radiation absorbed into that surface. We then modified out task_4 to determine the operating point TH and TC values and the thermoelectric efficiency and performance parameters for this modified system design at the conditions we specified in Task 4, part (b). We then determined the percent loss in efficiency and total power.

To make our new function, we now knew that our Q_pv would change to power from the sun, since the pv cell didn't exist. So we had to minimize the error in the difference in the Q_sun and Q_H. Hence we took the code for task_4 and deleted the parts about the pv. We then inputted the value for Q_sun using the same formula for P'' and then multiplied that with 0.95 as the absorptivity. This essentially replaced the conductance between the pv cell and thermoelectric module. We then used a similar format to task_4 where we defined the err as 1 and assumed a value for T_H between 150 and 250 K. We then used this value of T_H and used the function task_3_2 to find the Q_H. We then calculated the error between the difference in Q_sun and Q_H. We then iterated the calculations to go through different temperatures to reduce the error to below 10^-5. We used the bisection method to determine the new temperature limits. You can view task_6b in appendix **insert.** Below are the results from 6_b vs the results of task_4b **.**

	Task 6_b	**Task_4**
Q_H	**153.9000 W**	
I_L	**13.9906 A**	**64.0701 A**
$\mathbf{P_{tot}}$	**29.0402 W**	**51.9779 W**
$\boldsymbol{\eta_{tot}}$	**0.1887**	**0.3209**
T_H	**230.6268 K**	**216.4060 K**
V_L	**3.5364 V**	**0.4485 V**
T_C	**97.7673 K**	**96.8680 K**

Table 2: Values from tasks 6b and 4b.

By observation, we can see that the Power output and the efficiency have decreased considerably.

The percent loss is calculated by: $\frac{(new - old)}{old} * 100$, to give a negative percentage signifying loss**.**

% loss for power = 44% loss.

% loss for efficiency = 41.2% loss.

Hence it isn't advisable to remove the PV cell, as the power output increases.

Conclusion:

The overarching purpose of this project was to analyze the performance of a concentrating photovoltaic (PV) system that also captures waste heat from the PV cell to produce power, a so-called hybrid power system in 6 steps. This system was proposed for a spacecraft designed to transport astronauts from the earth to the moon and back. Both students discussed and contributed to the design and correction of all 6 tasks and the final report together throughout the duration of the project. Deep mainly worked on tasks 4, 5 and 6 and writing down tasks 5, 6 while Audrey worked on tasks 1, 2 design and writing along with Introduction, tasks 3 and 4 written parts. Design of task 3 was shared between both teammates.

Through Task 1 to 2, the general equations were established in order to find the relationship between I_o and I_v and also to find the power delivered and lost by the PV cell. Equations 14.16, 14.17, 14.47 and 2 were mainly used for this purpose.

From task 3 onwards, we started focusing on the thermoelectric module as well. Here we found a way to calculate the cold side temperature for the thermoelectric pairs for a given hot side temperature given the resistivity and conductivity of the arms as well as the resistance of the load and saturation temperature to heat the Oxygen for the astronauts.

Then we looked at combining task 2 and task 3 to analyse the temperatures of the whole system and this involved guessing 1 temperature and guessing another temperature based on that guess. We used this guessing and the bisection method to minimize the error between Q_pv and Q_H, to reduce heat losses. Task 4 gave us a method to find the temperature of the PV cell, the hot side and cold side temperatures of the thermoelectric unit and the different efficiencies and power outputs of the two parts. We then combined this into one given power output and efficiency output. Task_4 was primarily used to determine the characteristics of the whole hybrid system.

Then we looked at how altering different inputs changed the efficiencies of the PV cell, thermoelectric unit and the whole hybrid system along with the power output. Different plots were produced and then we saw the different trends in tasks 5A and 5B.

We then used the same function in task_4 to make a design choice on the rc and n units depending on 4 specifications. We ended up with 2 options which could be prioritized based on different specifications in task_6a.

Final suggestions:

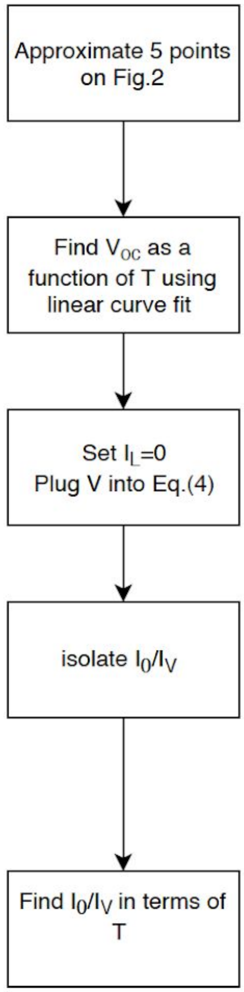
n = 11, rc = 14 with P_tot = 48.7011W, T_pv = 227.7106 K and eff = 0.3221.

n = 14, rc = 15 with P_tot = 48.3057W, T_pv = 191.3615 K and eff = 0.2982. In task_6b, we explored what would happen if we were to only have the thermoelectric unit and discovered that there would be a loss of **44% in Power and 41% in efficiency.**

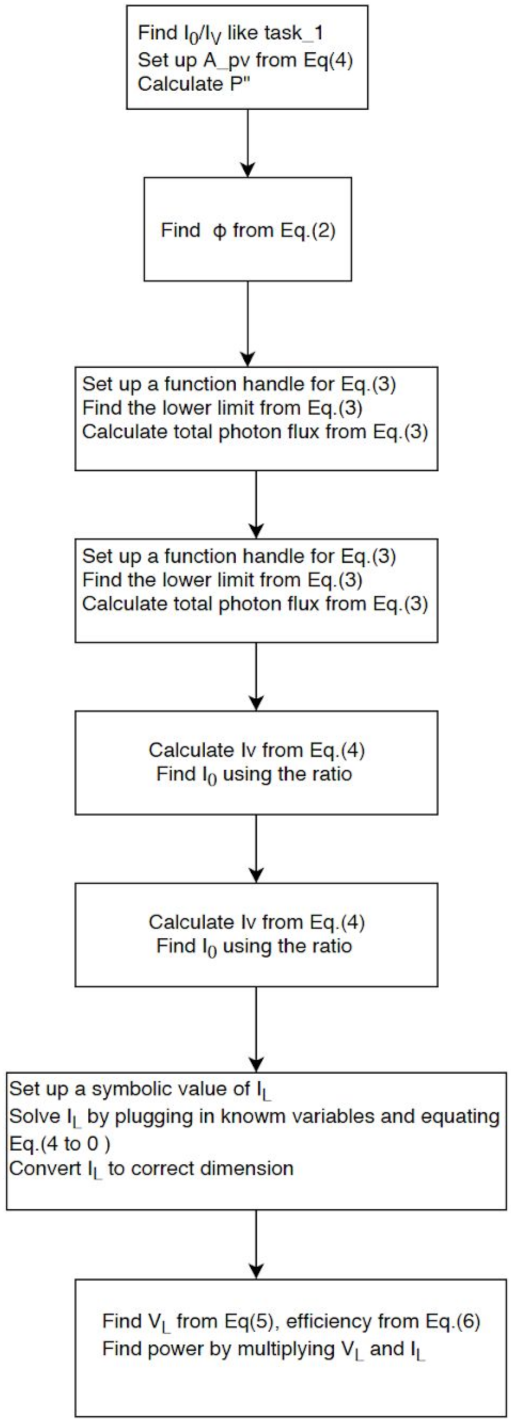
Therefore there were a lot of different things explored and we developed a way of analysing the different parameters of operation of the hybrid system and how they depend on input radiation, concentration ratios and the number of thermoelectric units.

Flowcharts

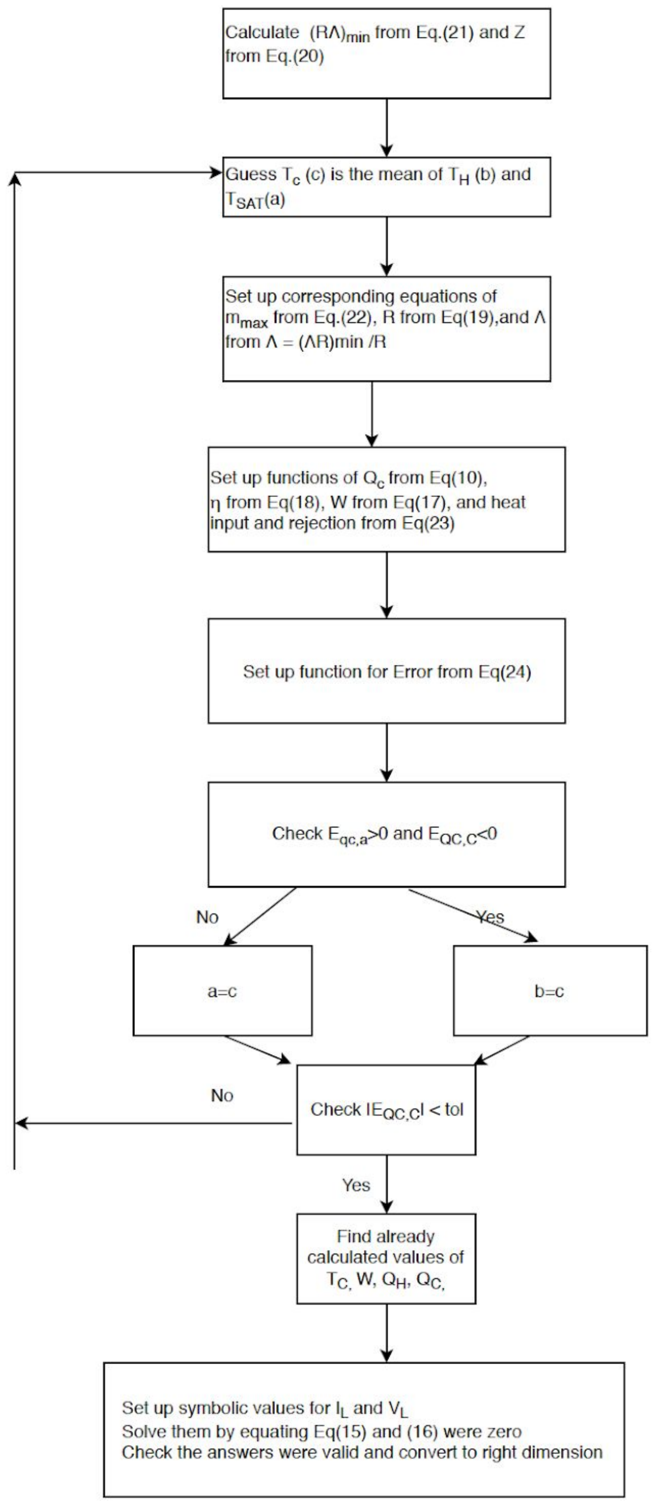
Task 1


Approximate 5 points on Fig.2
Find V_{OC} as a function of T using linear curve fit
Set I_L=0
Plug V into Eq.(4)
isolate I_0/I_V
Find I_0/I_V in terms of T

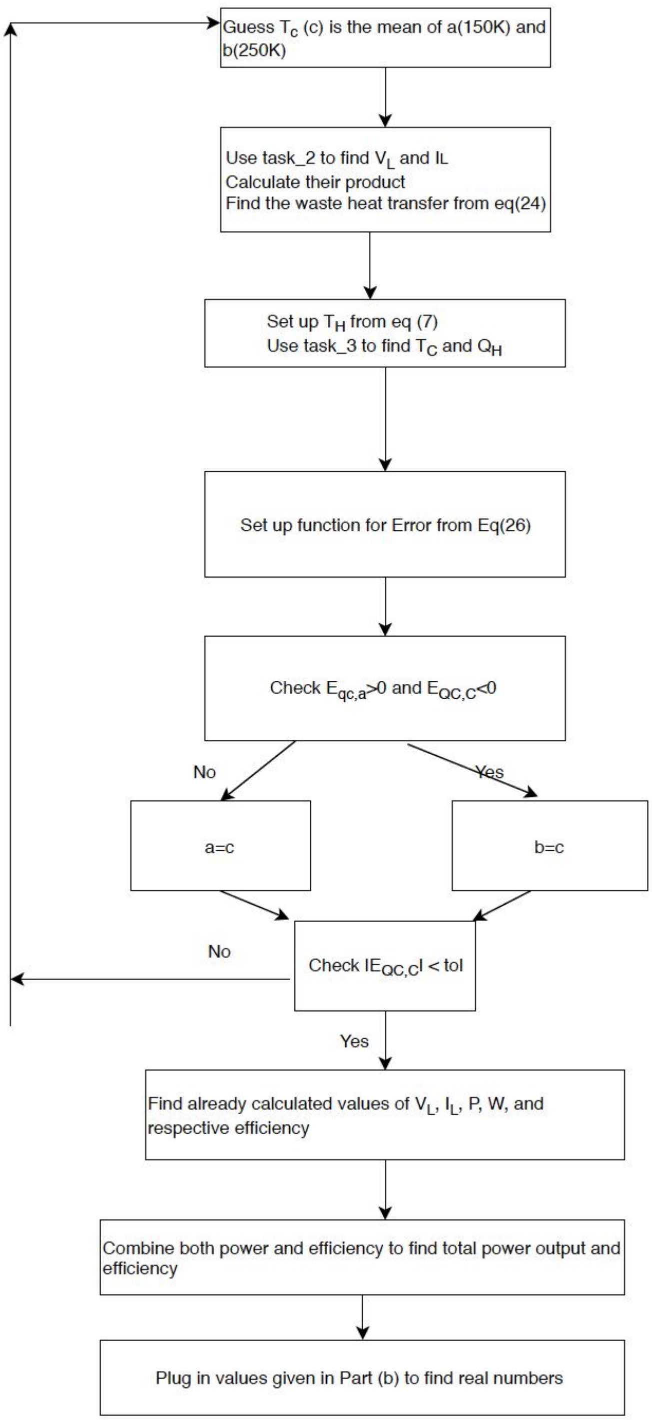
Task 2


Find I0/IV like task_1
Set up A_pv from Eq(4)
Calculate P"
Find ϕ from Eq.(2)
Set up a function handle for Eq.(3)
Find the lower limit from Eq.(3)
Calculate total photon flux from Eq.(3)
Set up a function handle for Eq.(3)
Find the lower limit from Eq.(3)
Calculate total photon flux from Eq.(3)
Calculate Iv from Eq.(4)
Find I0 using the ratio
Calculate Iv from Eq.(4)
Find I0 using the ratio
Set up a symbolic value of IL
Solve IL by plugging in knowm variables and equating
Eq.(4 to 0)
Convert IL to correct dimension
Find VL from Eq(5), efficiency from Eq.(6)
Find power by multiplying VL and IL

Task 3


Calculate $(R\Lambda)_{min}$ from Eq.(21) and Z from Eq.(20)
Guess T_c (c) is the mean of T_H (b) and T_{SAT}(a)
Set up corresponding equations of m_{max} from Eq.(22), R from Eq(19),and Λ from Λ = (ΛR)min /R
Set up functions of Q_c from Eq(10), η from Eq(18), W from Eq(17), and heat input and rejection from Eq(23)
Set up function for Error from Eq(24)
Check $E_{qc,a}$>0 and $E_{QC,C}$<0
No
Yes
a=c
b=c
No
Check $|E_{QC,C}|$ < tol
Yes
Find already calculated values of T_C, W, Q_H, Q_C,
Set up symbolic values for I_L and V_L
Solve them by equating Eq(15) and (16) were zero
Check the answers were valid and convert to right dimension

Task 4


Guess T_c (c) is the mean of a(150K) and b(250K)
Use task_2 to find V_L and IL
Calculate their product
Find the waste heat transfer from eq(24)
Set up T_H from eq (7)
Use task_3 to find T_C and Q_H
Set up function for Error from Eq(26)
Check E_qc,a>0 and E_QC,C<0
No
Yes
a=c
b=c
No
Check IE_QC,CI < tol
Yes
Find already calculated values of V_L, I_L, P, W, and respective efficiency
Combine both power and efficiency to find total power output and efficiency
Plug in values given in Part (b) to find real numbers

Appendix I


```
T = 30;
%linear curve fit to find the relationship b/t T and V_oc, 5 points are
%estimated based on the graph provided

T_p=[293;303;313;333;353];
V_oc=[0.39;0.37;0.35;0.32;0.27];
T_pv = horzcat(T_p,ones(5,1));
co = T_pv\V_oc;
V_L = [(T+273), 1]*co;

%constants
Kb= 1.38*10^-23;
qe= 1.6*10^-19;
x=V_L*qe/(Kb*(T+273));
I0_Iv = 1/(exp(x)-1);
I0_Iv
```

```
function [P, Eta,Q_pv, V_L,I_L] = task_2(R_L, Ly, Lz, Id,rc,T, Vg)
%same as task_1
T_p=[293;303;313;333;353];
V_oc=[0.39;0.37;0.35;0.32;0.27];
T_pv = horzcat(T_p,ones(5,1));
co = T_pv\V_oc;
V_oc = [(T+273), 1]*co;
%constants
Kb= 1.38*10^-23;
qe= 1.6*10^-19;
L=V_oc*qe/(Kb*(T+273));
I0_Iv = 1/(exp(L)-1);
A_pv = (Ly*Lz)/10000;
%I_L = V_L / R_L;
%Eta = V_L * I_L / (p * A_pv);
% finding I_L
P_0= Id*rc*A_pv;

phi = (Id*rc*2.404*15)/(pi^4*Kb*(5778));

syms I_L

fun= @(x) (x.^2)./(exp(x)-1);

xmin = qe*Vg/(Kb* 5778);

inte = integral(fun,xmin,Inf);

phi_g = phi * 0.416 * inte;

Iv = phi_g * qe * A_pv;

I0 =Iv * I0_Iv;
%solving for I_L
eqn= (Kb*(T+273)/qe)*log(((Iv- I_L)/I0)+1)- I_L*R_L==0;
I_L_sol = solve(eqn,I_L);
I_L = double(subs(I_L_sol));
%solving for V_L and then Efficiency
V_L = I_L*R_L;
Eta = V_L*I_L/ (Id*rc*A_pv);
P = I_L*V_L;
Q_pv = -V_L*I_L + P_0;
end
```

```
function [Z, Rlam_min,  T_C,Eta_te, W,Q_H, Q_C, V_L, I_L ] = task_3_2(alpha, lam_A, lam_B, rho_A,↙
rho_B, n, R_L, T_H, T_sat)
Rlam_min = ((lam_A*rho_A)^0.5 + (lam_B*rho_B)^0.5)^2;
Z= alpha^2 / Rlam_min;
A_pv = 0.1^2;
k_w = 6.5;
t_w = 0.4*10^-2;
U_ev = 25;
a = T_sat;
b = T_H+1;
m_max =@(T_C) (1 + 0.5*(T_H+T_C)*Z)^0.5;
R =@(m_max) R_L/(n * m_max);
lam = @(R) Rlam_min / R;
fQ_C =@(T_C)  A_pv*(k_w/t_w + U_ev) * (T_C-T_sat);
Eta =@(T_C, m_max) ((T_H-T_C) / T_H) *(((1+m_max)^2 / m_max) * (Z*T_H)^(-1) + 1 + (1/(2*m_max))*↙
(1+T_C/T_H))^(-1);
R_batt = @(R) n* R;
fW =@(T_C, R_batt) (n^2 * alpha^2 * (T_H-T_C)^2 * R_L )/ ((R_L + R_batt)^2);
fQ_H =@(W,Eta) W/Eta;
Q_CTE = @(Q_H, W)Q_H - W;
E_QC =@(Q_CTE, Q_C) Q_CTE - Q_C;
err = 1;
while err > 10^(-5)
   c=(a+b)/2;
    m_max_a =m_max(a);
    m_max_b =m_max(b);
    m_max_c =m_max(c);
    R_a =R(m_max_a);
    R_b =R(m_max_b);
    R_c =R(m_max_c);
    lam_a = lam(R_a);
    lam_b = lam(R_b);
    lam_c = lam(R_c);
    Q_C_a =fQ_C(a);
    Q_C_b =fQ_C(b);
    Q_C_c =fQ_C(c);
    Eta_a =Eta(a, m_max_a);
    Eta_b =Eta(b, m_max_b);
    Eta_c =Eta(c, m_max_c);
    R_batt_a = R_batt(R_a);
    R_batt_b = R_batt(R_b);
    R_batt_c = R_batt(R_c);
    W_a =fW(a, R_batt_a);
    W_b =fW(b, R_batt_b);
    W_c =fW(c, R_batt_c);
    Q_H_a =fQ_H(W_a,Eta_a);
    Q_H_b =fQ_H(W_b,Eta_b);
    Q_H_c =fQ_H(W_c,Eta_c);
    Q_CTE_a = Q_CTE(Q_H_a, W_a);
    Q_CTE_b = Q_CTE(Q_H_b, W_b);
    Q_CTE_c = Q_CTE(Q_H_c, W_c);
    E_QC_a =E_QC(Q_CTE_a, Q_C_a);
    E_QC_b =E_QC(Q_CTE_b, Q_C_b);
    E_QC_c =E_QC(Q_CTE_c, Q_C_c);
    if E_QC_a>0 &&  E_QC_c < 0
        b=c;
```

```
    else
        a=c;
    end
    err = abs(E_QC_c);
end

T_C = c;
W = W_c;
Q_H = Q_H_c;
Q_C = Q_C_c;
lam_batt = n*lam_c;
R_batt_2 = R_batt_c;
Eta_te = Eta_c;
syms I_L V_L
eqn1=V_L- (n*alpha*(T_H-T_C))-R_batt_2*I_L==0;
eqn2= Q_H- n*alpha*I_L*T_H- lam_batt*(T_H-T_C)-0.5*((I_L)^2)*R_batt_2==0;
[V_L_sol, I_L_sol] = solve([eqn1,eqn2],V_L,I_L,'IgnoreAnalyticConstraints', true);
V_L_2= double(subs(V_L_sol))>0;
V_L = double(V_L_sol(double(subs(V_L_sol))>0));
I_L = double(I_L_sol(double(subs(I_L_sol))>0));

end
```

```
%setting constants;
lamb_cont= 300;
Id = 1080;
rc = 15;
Ly = 10 ;
Lz = 10 ;
Vg = 1.1;
fg = 265 ;
Rs = 0;
R_L_pv = 0.0070;
T_sat = 90.2 ;
errr = 1;
alpha= 0.0017;
lam_A= 0.032;
lam_B= 0.021;
rho_A=0.0020;
rho_B=0.0030;
n=12;
R_L=0.1;
i =1;
a1 = 150;
b1= 250;
%doing error analysis
while errr > 10^(-5)
c1 = (a1+b1)/2;
T_pv(i,:)= [a1,c1,b1];
[P(i,1), Eta(i,1),Q_pv(i,1), V_L(i,1),I_L(i,1)]= task_2(R_L_pv, Ly,Lz, Id, rc,a1,Vg);
[P(i,2), Eta(i,2),Q_pv(i,2), V_L(i,2),I_L(i,2)]= task_2(R_L_pv, Ly,Lz, Id, rc,c1,Vg);
[P(i,3), Eta(i,3),Q_pv(i,3), V_L(i,3),I_L(i,3)]= task_2(R_L_pv, Ly,Lz, Id, rc,b1,Vg);
T_H(i,:) = T_pv(i,:) - Q_pv(i,:)./lamb_cont;
[Z(i,1), Rlam_min(i,1),  T_C(i,1),Eta_te(i,1), W(i,1),Q_H(i,1), Q_C(i,1), V_L_te(i,1), I_L_te(i,↙
1) ] = task_3_2(alpha, lam_A, lam_B, rho_A, rho_B, n, R_L, T_H(i,1), T_sat);
[Z(i,2), Rlam_min(i,2),  T_C(i,2),Eta_te(i,2), W(i,2),Q_H(i,2), Q_C(i,2), V_L_te(i,2), I_L_te(i,↙
2) ] = task_3_2(alpha, lam_A, lam_B, rho_A, rho_B, n, R_L, T_H(i,2), T_sat);
[Z(i,3), Rlam_min(i,3),  T_C(i,3),Eta_te(i,3), W(i,3),Q_H(i,3), Q_C(i,3), V_L_te(i,3), I_L_te(i,↙
3) ] = task_3_2(alpha, lam_A, lam_B, rho_A, rho_B, n, R_L, T_H(i,3), T_sat);
E_Q(i,:) = Q_pv(i,:) - Q_H(i,:);
if E_Q(i,1)>0 &&  E_Q(i,2) < 0
        b1=c1;
    else
        a1=c1;
    end
errr = abs(E_Q(i,2));
i = i+1;
end
ii= i;
T_H = T_H(i-1,2)
T_C= T_C(i-1,2)
T_pv = T_pv(i-1,2)
Eta_te = Eta_te(i-1,2)
Eta= Eta(i-1,2)
P = P(i-1,2)
W= W(i-1,2)
V_L= V_L(i-1,2)
I_L = I_L(i-1,2)
P_tot = P+W;
```

```
Q_H = Q_H(i-1,2)
Eta_tot = P_tot/ (Id*rc*(0.1^2));
function [P, Eta,Q_pv, V_L,I_L] = task_2(R_L, Ly, Lz, Id,rc,T, Vg)
T_p=[293;303;313;333;353];
V_oc=[0.39;0.37;0.35;0.32;0.27];
T_pv = horzcat(T_p,ones(5,1));
co = T_pv\V_oc;
V_oc = [(T), 1]*co;

%constants
Kb= 1.38*10^-23;
qe= 1.6*10^-19;
L=V_oc*qe/(Kb*(T));
I0_Iv = 1/(exp(L)-1);
A_pv = (Ly*Lz)/10000;
%I_L = V_L / R_L;
%Eta = V_L * I_L / (p * A_pv);
P_0= Id*rc*A_pv;

phi = (Id*rc*2.404*15)/(pi^4*Kb*(5778));

syms I_L

fun= @(x) (x.^2)./(exp(x)-1);

xmin = qe*Vg/(Kb* 5778);

inte = integral(fun,xmin,Inf);

phi_g = phi * 0.416 * inte;

Iv = phi_g * qe * A_pv;

I0 =Iv * I0_Iv;

eqn= (Kb*(T)/qe)*log(((Iv- I_L)/I0)+1)- I_L*R_L==0;
I_L_sol = solve(eqn,I_L);
I_L = double(subs(I_L_sol));
V_L = I_L*R_L;
Eta = V_L*I_L/ (Id*rc*A_pv);
P = I_L*V_L;
Q_pv = -V_L*I_L + P_0;
end

function [Z, Rlam_min,  T_C,Eta_te, W,Q_H, Q_C, V_L_te, I_L_te] = task_3_2(alpha, lam_A, lam_B,↙
rho_A, rho_B, n, R_L, T_H, T_sat)
%%%%%add I_l and v_l for this question
Rlam_min = ((lam_A*rho_A)^0.5 + (lam_B*rho_B)^0.5)^2;
Z= alpha^2 / Rlam_min;
A_pv = 0.1^2;
k_w = 6.5;
t_w = 0.4*10^-2;
U_ev = 25;

% guess T_C = (T_H + T_sat)/2 = (323+95)/2 = 209 K
a = T_sat;
```

```
b = T_H+1;
m_max =@(T_C) (1 + 0.5*(T_H+T_C)*Z)^0.5;
R =@(m_max) R_L/(n * m_max);
lam = @(R) Rlam_min / R;
fQ_C =@(T_C)A_pv*  (k_w/t_w + U_ev) * (T_C-T_sat);
Eta =@(T_C, m_max) ((T_H-T_C) / T_H) *(((1+m_max)^2 / m_max) * (Z*T_H)^(-1) + 1 + (1/(2*m_max))*
(1+T_C/T_H))^(-1);
R_batt = @(R) n* R;
fW =@(T_C, R_batt) (n^2 * alpha^2 * (T_H-T_C)^2 * R_L )/ ((R_L + R_batt)^2);
fQ_H =@(W,Eta) W/Eta;
Q_CTE = @(Q_H, W)Q_H - W;
E_QC =@(Q_CTE, Q_C) Q_CTE - Q_C;
err = 1;
while err > 10^(-5)
   c=(a+b)/2;
    m_max_a =m_max(a);
    m_max_b =m_max(b);
    m_max_c =m_max(c);
    R_a =R(m_max_a);
    R_b =R(m_max_b);
    R_c =R(m_max_c);
    lam_a = lam(R_a);
    lam_b = lam(R_b);
    lam_c = lam(R_c);
    Q_C_a =fQ_C(a);
    Q_C_b =fQ_C(b);
    Q_C_c =fQ_C(c);
    Eta_a =Eta(a, m_max_a);
    Eta_b =Eta(b, m_max_b);
    Eta_c =Eta(c, m_max_c);
    R_batt_a = R_batt(R_a);
    R_batt_b = R_batt(R_b);
    R_batt_c = R_batt(R_c);
    W_a =fW(a, R_batt_a);
    W_b =fW(b, R_batt_b);
    W_c =fW(c, R_batt_c);
    Q_H_a =fQ_H(W_a,Eta_a);
    Q_H_b =fQ_H(W_b,Eta_b);
    Q_H_c =fQ_H(W_c,Eta_c);
    Q_CTE_a = Q_CTE(Q_H_a, W_a);
    Q_CTE_b = Q_CTE(Q_H_b, W_b);
    Q_CTE_c = Q_CTE(Q_H_c, W_c);
    E_QC_a =E_QC(Q_CTE_a, Q_C_a);
    E_QC_b =E_QC(Q_CTE_b, Q_C_b);
    E_QC_c =E_QC(Q_CTE_c, Q_C_c);
    if E_QC_a>0 &&  E_QC_c < 0
        b=c;
    else
        a=c;
    end
    err = abs(E_QC_c);
end

T_C = c;
W = W_c;
Q_H = Q_H_c;
```

```
Q_C = Q_C_c;
lam_batt = n*lam_c;
R_batt_2 = R_batt_c;
Eta_te = Eta_c;
syms I_L_te V_L_te
eqn1=V_L_te- (n*alpha*(T_H-T_C))-R_batt_2*I_L_te==0;
eqn2= Q_H- n*alpha*I_L_te*T_H- lam_batt*(T_H-T_C)-0.5*((I_L_te)^2)*R_batt_2==0;
%%%%%need to add task_2
[V_L_sol, I_L_sol] = solve([eqn1,eqn2],V_L_te,I_L_te,'IgnoreAnalyticConstraints', true);
V_L_2= double(subs(V_L_sol))>0;
V_L_te = double(V_L_sol(double(subs(V_L_sol))>0));
I_L_te = double(I_L_sol(double(subs(I_L_sol))>0));

end
```

```
rc = 15;
Ly = 10 ;
Lz = 10 ;
Vg = 1.1;
fg = 265 ;
Rs = 0;
Id_i = linspace(720,1080,12);
R_L_pv = 0.0070;
T_sat = 90.2 ;
errr = 1;
alpha= 0.0017;
lam_A= 0.032;
lam_B= 0.021;
rho_A=0.0020;
rho_B=0.0030;
n=12;
R_L=0.1;
i =1;
a1 = 150;
b1= 250;
lamb_cont= 300;
for l = 1:length(Id_i)
    Id = Id_i(l);
    disp(l);
    i=1;
    errr = 1;
    a1 = T_sat+1;
    b1= 383;
while errr > 10^(-5)
c1 = (a1+b1)/2;
T_pv_i(i,:)= [a1,c1,b1];
[P(i,1), Eta(i,1),Q_pv(i,1), V_L(i,1),I_L(i,1)]= task_2(R_L_pv, Ly,Lz, Id, rc,a1,Vg);
[P(i,2), Eta(i,2),Q_pv(i,2), V_L(i,2),I_L(i,2)]= task_2(R_L_pv, Ly,Lz, Id, rc,c1,Vg);
[P(i,3), Eta(i,3),Q_pv(i,3), V_L(i,3),I_L(i,3)]= task_2(R_L_pv, Ly,Lz, Id, rc,b1,Vg);
T_H_i(i,:) = T_pv_i(i,:) - Q_pv(i,:)./lamb_cont;
[Z(i,1), Rlam_min(i,1),  T_C(i,1),Eta_tee(i,1), W(i,1),Q_H(i,1), Q_C(i,1), V_L_te(i,1), I_L_te(i,↙
1) ] = task_3_2(alpha, lam_A, lam_B, rho_A, rho_B, n, R_L, T_H_i(i,1), T_sat);
[Z(i,2), Rlam_min(i,2),  T_C(i,2),Eta_tee(i,2), W(i,2),Q_H(i,2), Q_C(i,2), V_L_te(i,2), I_L_te(i,↙
2) ] = task_3_2(alpha, lam_A, lam_B, rho_A, rho_B, n, R_L, T_H_i(i,2), T_sat);
[Z(i,3), Rlam_min(i,3),  T_C(i,3),Eta_tee(i,3), W(i,3),Q_H(i,3), Q_C(i,3), V_L_te(i,3), I_L_te(i,↙
3) ] = task_3_2(alpha, lam_A, lam_B, rho_A, rho_B, n, R_L, T_H_i(i,3), T_sat);
E_Q(i,:) = Q_pv(i,:) - Q_H(i,:);
if E_Q(i,1)>0 &&  E_Q(i,2) < 0
        b1=c1;
    else
        a1=c1;
    end
errr = abs(E_Q(i,2));
i = i+1;
end
T_H(1,l) = T_H_i(i-1,2);
T_C(1,l)= T_C(i-1,2);
T_pv(1,l) = T_pv_i(i-1,2);
Eta_te(1,l) = Eta_tee(i-1,2);
Eta_pv(1,l)= Eta(i-1,2);
P_pv(1,l) = P(i-1,2);
```

```
W_te(1,l)= W(i-1,2);
V_L(1,l)= V_L(i-1,2);
I_L(1,l) = I_L(i-1,2);
P_tot(1,l) = P_pv(1,l)+W_te(1,l);
Eta_tot(1,l) = P_tot(1,l)/ (Id*rc*(0.1^2));
V_L_te(1,l) = V_L_te(i-1,2);
I_L_te(1,l) = I_L_te(i-1,2);
end
subplot(2,2,1)
plot(Id_i,transpose(Eta_tot),'.');
title('Subplot 1: Total Efficiency vs Variations in Incident Radiation');
xlabel('Incident Radiation (W/(m^2))')
subplot(2,2,2)
plot(Id_i,transpose(Eta_te),'.');
title('Subplot 2: Thermal Efficiency vs Variations in Incident Radiation');
subplot(2,2,3)
plot(Id_i,transpose(Eta_pv),'.');
title('Subplot 3: PV Efficiency vs Variations in Incident Radiation');
subplot(2,2,4)
plot(Id_i,transpose(P_tot), '.');

title('Subplot 4: Total Power vs Variations in Incident Radiation');
function [P, Eta,Q_pv, V_L,I_L] = task_2(R_L, Ly, Lz, Id,rc,T, Vg)
T_p=[293;303;313;333;353];
V_oc=[0.39;0.37;0.35;0.32;0.27];
T_pv = horzcat(T_p,ones(5,1));
co = T_pv\V_oc;
V_oc = [(T), 1]*co;

%constants
Kb= 1.38*10^-23;
qe= 1.6*10^-19;
L=V_oc*qe/(Kb*(T));
I0_Iv = 1/(exp(L)-1);
A_pv = (Ly*Lz)/10000;
%I_L = V_L / R_L;
%Eta = V_L * I_L / (p * A_pv);
P_0= Id*rc*A_pv;

phi = (Id*rc*2.404*15)/(pi^4*Kb*(5778));

syms I_L

fun= @(x) (x.^2)./(exp(x)-1);

xmin = qe*Vg/(Kb* 5778);

inte = integral(fun,xmin,Inf);

phi_g = phi * 0.416 * inte;

Iv = phi_g * qe * A_pv;

I0 =Iv * I0_Iv;

eqn= (Kb*(T)/qe)*log(((Iv- I_L)/I0)+1)- I_L*R_L==0;
```

```
I_L_sol = solve(eqn,I_L);
I_L = double(subs(I_L_sol));
V_L = I_L*R_L;
Eta = V_L*I_L/ (Id*rc*A_pv);
P = I_L*V_L;
Q_pv = -V_L*I_L + P_0;
end

function [Z, Rlam_min,  T_C,Eta_te, W,Q_H, Q_C, V_L_te, I_L_te] = task_3_2(alpha, lam_A, lam_B,↙
rho_A, rho_B, n, R_L, T_H, T_sat)
%%%%%add I_l and v_l for this question
Rlam_min = ((lam_A*rho_A)^0.5 + (lam_B*rho_B)^0.5)^2;
Z= alpha^2 / Rlam_min;
A_pv = 0.1^2;
k_w = 6.5;
t_w = 0.4*10^-2;
U_ev = 25;

% guess T_C = (T_H + T_sat)/2 = (323+95)/2 = 209 K
a = T_sat;
b = T_H+1;
m_max =@(T_C) (1 + 0.5*(T_H+T_C)*Z)^0.5;
R =@(m_max) R_L/(n * m_max);
lam = @(R) Rlam_min / R;
fQ_C =@(T_C) A_pv* (k_w/t_w + U_ev) * (T_C-T_sat);
Eta =@(T_C, m_max) ((T_H-T_C) / T_H) *(((1+m_max)^2 / m_max) * (Z*T_H)^(-1) + 1 + (1/(2*m_max))*↙
(1+T_C/T_H))^(-1);
R_batt = @(R) n* R;
fW =@(T_C, R_batt) (n^2 * alpha^2 * (T_H-T_C)^2 * R_L )/ ((R_L + R_batt)^2);
fQ_H =@(W,Eta) W/Eta;
Q_CTE = @(Q_H, W)Q_H - W;
E_QC =@(Q_CTE, Q_C) Q_CTE - Q_C;
err = 1;
while err > 10^(-5)
   c=(a+b)/2;
    m_max_a =m_max(a);
    m_max_b =m_max(b);
    m_max_c =m_max(c);
    R_a =R(m_max_a);
    R_b =R(m_max_b);
    R_c =R(m_max_c);
    lam_a = lam(R_a);
    lam_b = lam(R_b);
    lam_c = lam(R_c);
    Q_C_a =fQ_C(a);
    Q_C_b =fQ_C(b);
    Q_C_c =fQ_C(c);
    Eta_a =Eta(a, m_max_a);
    Eta_b =Eta(b, m_max_b);
    Eta_c =Eta(c, m_max_c);
    R_batt_a = R_batt(R_a);
    R_batt_b = R_batt(R_b);
    R_batt_c = R_batt(R_c);
    W_a =fW(a, R_batt_a);
    W_b =fW(b, R_batt_b);
    W_c =fW(c, R_batt_c);
```

```
    Q_H_a =fQ_H(W_a,Eta_a);
    Q_H_b =fQ_H(W_b,Eta_b);
    Q_H_c =fQ_H(W_c,Eta_c);
    Q_CTE_a = Q_CTE(Q_H_a, W_a);
    Q_CTE_b = Q_CTE(Q_H_b, W_b);
    Q_CTE_c = Q_CTE(Q_H_c, W_c);
    E_QC_a =E_QC(Q_CTE_a, Q_C_a);
    E_QC_b =E_QC(Q_CTE_b, Q_C_b);
    E_QC_c =E_QC(Q_CTE_c, Q_C_c);
    if E_QC_a>0 &&  E_QC_c < 0
        b=c;
    else
        a=c;
    end
    err = abs(E_QC_c);
end

T_C = c;
W = W_c;
Q_H = Q_H_c;
Q_C = Q_C_c;
lam_batt = n*lam_c;
R_batt_2 = R_batt_c;
Eta_te = Eta_c;
syms I_L_te V_L_te
eqn1=V_L_te- (n*alpha*(T_H-T_C))-R_batt_2*I_L_te==0;
eqn2= Q_H- n*alpha*I_L_te*T_H- lam_batt*(T_H-T_C)-0.5*((I_L_te)^2)*R_batt_2==0;
%%%%%need to add task_2
[V_L_sol, I_L_sol] = solve([eqn1,eqn2],V_L_te,I_L_te,'IgnoreAnalyticConstraints', true);
V_L_2= double(subs(V_L_sol))>0;
V_L_te = double(V_L_sol(double(subs(V_L_sol))>0));
I_L_te = double(I_L_sol(double(subs(I_L_sol))>0));

end
```

```
Id = 1080;
Ly = 10 ;
Lz = 10 ;
Vg = 1.1;
fg = 265 ;
Rs = 0;
rc_i = linspace(10,18,9);
R_L_pv = 0.0070;
T_sat = 90.2 ;
errr = 1;
alpha= 0.0017;
lam_A= 0.032;
lam_B= 0.021;
rho_A=0.0020;
rho_B=0.0030;
n=12;
R_L=0.1;
i =1;
a1 = 150;
b1= 250;
lamb_cont= 300;
for l = 1:length(rc_i)
    rc = rc_i(l);
    disp(l);
    i=1;
    errr = 1;
    a1 = T_sat+1;
    b1= 383;
while errr > 10^(-5)
c1 = (a1+b1)/2;
T_pv_i(i,:)= [a1,c1,b1];
[P(i,1), Eta(i,1),Q_pv(i,1), V_L(i,1),I_L(i,1)]= task_2(R_L_pv, Ly,Lz, Id, rc,a1,Vg);
[P(i,2), Eta(i,2),Q_pv(i,2), V_L(i,2),I_L(i,2)]= task_2(R_L_pv, Ly,Lz, Id, rc,c1,Vg);
[P(i,3), Eta(i,3),Q_pv(i,3), V_L(i,3),I_L(i,3)]= task_2(R_L_pv, Ly,Lz, Id, rc,b1,Vg);
T_H_i(i,:) = T_pv_i(i,:) - Q_pv(i,:)./lamb_cont;
[Z(i,1), Rlam_min(i,1),  T_C(i,1),Eta_tee(i,1), W(i,1),Q_H(i,1), Q_C(i,1), V_L_te(i,1), I_L_te(i,↙
1) ] = task_3_2(alpha, lam_A, lam_B, rho_A, rho_B, n, R_L, T_H_i(i,1), T_sat);
[Z(i,2), Rlam_min(i,2),  T_C(i,2),Eta_tee(i,2), W(i,2),Q_H(i,2), Q_C(i,2), V_L_te(i,2), I_L_te(i,↙
2) ] = task_3_2(alpha, lam_A, lam_B, rho_A, rho_B, n, R_L, T_H_i(i,2), T_sat);
[Z(i,3), Rlam_min(i,3),  T_C(i,3),Eta_tee(i,3), W(i,3),Q_H(i,3), Q_C(i,3), V_L_te(i,3), I_L_te(i,↙
3) ] = task_3_2(alpha, lam_A, lam_B, rho_A, rho_B, n, R_L, T_H_i(i,3), T_sat);
E_Q(i,:) = Q_pv(i,:) - Q_H(i,:);
if E_Q(i,1)>0 &&  E_Q(i,2) < 0
        b1=c1;
    else
        a1=c1;
    end
errr = abs(E_Q(i,2));
i = i+1;
end
T_H(1,l) = T_H_i(i-1,2);
T_C(1,l)= T_C(i-1,2);
T_pv(1,l) = T_pv_i(i-1,2);
Eta_te(1,l) = Eta_tee(i-1,2);
Eta_pv(1,l)= Eta(i-1,2);
P_pv(1,l) = P(i-1,2);
```

```
W_te(1,l)= W(i-1,2);
V_L(1,l)= V_L(i-1,2);
I_L(1,l) = I_L(i-1,2);
P_tot(1,l) = P_pv(1,l)+W_te(1,l);
Eta_tot(1,l) = P_tot(1,l)/ (Id*rc*(0.1^2));
V_L_te(1,l) = V_L_te(i-1,2);
I_L_te(1,l) = I_L_te(i-1,2);
end
subplot(2,2,1)
plot(rc_i,transpose(Eta_tot),'.');
title('Subplot 1: Total Efficiency vs Variations in Incident Radiation');
subplot(2,2,2)
plot(rc_i,transpose(Eta_te),'.');
title('Subplot 1: Thermal Efficiency vs Variations in Incident Radiation');
subplot(2,2,3)
plot(rc_i,transpose(Eta_pv),'.');
title('Subplot 1: PV Efficiency vs Variations in Incident Radiation');
subplot(2,2,4)
plot(rc_i,transpose(P_tot), '.');
title('Subplot 1: Total Power vs Variations in Incident Radiation');
function [P, Eta,Q_pv, V_L,I_L] = task_2(R_L, Ly, Lz, Id,rc,T, Vg)
T_p=[293;303;313;333;353];
V_oc=[0.39;0.37;0.35;0.32;0.27];
T_pv = horzcat(T_p,ones(5,1));
co = T_pv\V_oc;
V_oc = [(T), 1]*co;

%constants
Kb= 1.38*10^-23;
qe= 1.6*10^-19;
L=V_oc*qe/(Kb*(T));
I0_Iv = 1/(exp(L)-1);
A_pv = (Ly*Lz)/10000;
%I_L = V_L / R_L;
%Eta = V_L * I_L / (p * A_pv);
P_0= Id*rc*A_pv;

phi = (Id*rc*2.404*15)/(pi^4*Kb*(5778));

syms I_L

fun= @(x) (x.^2)./(exp(x)-1);

xmin = qe*Vg/(Kb* 5778);

inte = integral(fun,xmin,Inf);

phi_g = phi * 0.416 * inte;

Iv = phi_g * qe * A_pv;

I0 =Iv * I0_Iv;

eqn= (Kb*(T)/qe)*log(((Iv- I_L)/I0)+1)- I_L*R_L==0;
I_L_sol = solve(eqn,I_L);
I_L = double(subs(I_L_sol));
```

```
V_L = I_L*R_L;
Eta = V_L*I_L/ (Id*rc*A_pv);
P = I_L*V_L;
Q_pv = -V_L*I_L + P_0;
end

function [Z, Rlam_min,  T_C,Eta_te, W,Q_H, Q_C, V_L_te, I_L_te] = task_3_2(alpha, lam_A, lam_B,↙
rho_A, rho_B, n, R_L, T_H, T_sat)
%%%%%add I_l and v_l for this question
Rlam_min = ((lam_A*rho_A)^0.5 + (lam_B*rho_B)^0.5)^2;
Z= alpha^2 / Rlam_min;
A_pv = 0.1^2;
k_w = 6.5;
t_w = 0.4*10^-2;
U_ev = 25;

% guess T_C = (T_H + T_sat)/2 = (323+95)/2 = 209 K
a = T_sat;
b = T_H+1;
m_max =@(T_C) (1 + 0.5*(T_H+T_C)*Z)^0.5;
R =@(m_max) R_L/(n * m_max);
lam = @(R) Rlam_min / R;
fQ_C =@(T_C)  (k_w/t_w + U_ev) * (T_C-T_sat);
Eta =@(T_C, m_max) ((T_H-T_C) / T_H) *(((1+m_max)^2 / m_max) * (Z*T_H)^(-1) + 1 + (1/(2*m_max))*↙
(1+T_C/T_H))^(-1);
R_batt = @(R) n* R;
fW =@(T_C, R_batt) (n^2 * alpha^2 * (T_H-T_C)^2 * R_L )/ ((R_L + R_batt)^2);
fQ_H =@(W,Eta) W/Eta;
Q_CTE = @(Q_H, W)Q_H - W;
E_QC =@(Q_CTE, Q_C) Q_CTE - Q_C;
err = 1;
while err > 10^(-5)
   c=(a+b)/2;
    m_max_a =m_max(a);
    m_max_b =m_max(b);
    m_max_c =m_max(c);
    R_a =R(m_max_a);
    R_b =R(m_max_b);
    R_c =R(m_max_c);
    lam_a = lam(R_a);
    lam_b = lam(R_b);
    lam_c = lam(R_c);
    Q_C_a =fQ_C(a);
    Q_C_b =fQ_C(b);
    Q_C_c =fQ_C(c);
    Eta_a =Eta(a, m_max_a);
    Eta_b =Eta(b, m_max_b);
    Eta_c =Eta(c, m_max_c);
    R_batt_a = R_batt(R_a);
    R_batt_b = R_batt(R_b);
    R_batt_c = R_batt(R_c);
    W_a =fW(a, R_batt_a);
    W_b =fW(b, R_batt_b);
    W_c =fW(c, R_batt_c);
    Q_H_a =fQ_H(W_a,Eta_a);
    Q_H_b =fQ_H(W_b,Eta_b);
```

```
    Q_H_c =fQ_H(W_c,Eta_c);
    Q_CTE_a = Q_CTE(Q_H_a, W_a);
    Q_CTE_b = Q_CTE(Q_H_b, W_b);
    Q_CTE_c = Q_CTE(Q_H_c, W_c);
    E_QC_a =E_QC(Q_CTE_a, Q_C_a);
    E_QC_b =E_QC(Q_CTE_b, Q_C_b);
    E_QC_c =E_QC(Q_CTE_c, Q_C_c);
    if E_QC_a>0 &&  E_QC_c < 0
        b=c;
    else
        a=c;
    end
    err = abs(E_QC_c);
end

T_C = c;
W = W_c;
Q_H = Q_H_c;
Q_C = Q_C_c;
lam_batt = n*lam_c;
R_batt_2 = R_batt_c;
Eta_te = Eta_c;
syms I_L_te V_L_te
eqn1=V_L_te- (n*alpha*(T_H-T_C))-R_batt_2*I_L_te==0;
eqn2= Q_H- n*alpha*I_L_te*T_H- lam_batt*(T_H-T_C)-0.5*((I_L_te)^2)*R_batt_2==0;
%%%%%need to add task_2
[V_L_sol, I_L_sol] = solve([eqn1,eqn2],V_L_te,I_L_te,'IgnoreAnalyticConstraints', true);
V_L_2= double(subs(V_L_sol))>0;
V_L_te = double(V_L_sol(double(subs(V_L_sol))>0));
I_L_te = double(I_L_sol(double(subs(I_L_sol))>0));

end
```

```
Id = 1080;
Ly = 10 ;
Lz = 10 ;
Vg = 1.1;
fg = 265 ;
Rs = 0;
rc_i = linspace(10,18,9);
R_L_pv = 0.0070;
T_sat = 90.2 ;
errr = 1;
alpha= 0.0017;
lam_A= 0.032;
n_k = linspace(8,16,9);
lam_B= 0.021;
rho_A=0.0020;
rho_B=0.0030;
R_L=0.1;
i =1;
a1 = 150;
b1= 250;
lamb_cont= 300;
for k = 1:length(rc_i)
     rc = rc_i(k);
     disp(k);
    for l = 1:length(n_k)
        n = n_k(l);
        i=1;
        disp(l);
        errr = 1;
        a1 = T_sat+1;
        b1= 383;
        while errr > 10^(-5)
            c1 = (a1+b1)/2;
            T_pv_i(i,:)= [a1,c1,b1];
            [P(i,1), Eta(i,1),Q_pv(i,1), V_L(i,1),I_L(i,1)]= task_2(R_L_pv, Ly,Lz, Id, rc,a1,Vg);
            [P(i,2), Eta(i,2),Q_pv(i,2), V_L(i,2),I_L(i,2)]= task_2(R_L_pv, Ly,Lz, Id, rc,c1,Vg);
            [P(i,3), Eta(i,3),Q_pv(i,3), V_L(i,3),I_L(i,3)]= task_2(R_L_pv, Ly,Lz, Id, rc,b1,Vg);
            T_H_i(i,:) = T_pv_i(i,:) - Q_pv(i,:)./lamb_cont;
            [Z(i,1), Rlam_min(i,1),  T_C(i,1),Eta_tee(i,1), W(i,1),Q_H(i,1), Q_C(i,1), V_L_te(i,↙
1), I_L_te(i,1) ] = task_3_2(alpha, lam_A, lam_B, rho_A, rho_B, n, R_L, T_H_i(i,1), T_sat);
            [Z(i,2), Rlam_min(i,2),  T_C(i,2),Eta_tee(i,2), W(i,2),Q_H(i,2), Q_C(i,2), V_L_te(i,↙
2), I_L_te(i,2) ] = task_3_2(alpha, lam_A, lam_B, rho_A, rho_B, n, R_L, T_H_i(i,2), T_sat);
            [Z(i,3), Rlam_min(i,3),  T_C(i,3),Eta_tee(i,3), W(i,3),Q_H(i,3), Q_C(i,3), V_L_te(i,↙
3), I_L_te(i,3) ] = task_3_2(alpha, lam_A, lam_B, rho_A, rho_B, n, R_L, T_H_i(i,3), T_sat);
            E_Q(i,:) = Q_pv(i,:) - Q_H(i,:);
            if E_Q(i,1)>0 &&  E_Q(i,2) < 0
                b1=c1;
            else
                a1=c1;
            end
        errr = abs(E_Q(i,2));
        i = i+1;
        end
T_H(k,l) = T_H_i(i-1,2);
T_C(k,l)= T_C(i-1,2);
T_pv(k,l) = T_pv_i(i-1,2);
```

```
Eta_te(k,l) = Eta_tee(i-1,2);
Eta_pv(k,l)= Eta(i-1,2);
P_pv(k,l) = P(i-1,2);
W_te(k,l)= W(i-1,2);
V_L(k,l)= V_L(i-1,2);
I_L(k,l) = I_L(i-1,2);
P_tot(k,l) = P_pv(1,l)+W_te(1,l);
Eta_tot(k,l) = P_tot(1,l)/ (Id*rc*(0.1^2));
V_L_te(k,l) = V_L_te(i-1,2);
I_L_te(k,l) = I_L_te(i-1,2);
    end
end
[R,C]= find(P_tot<=50 & P_tot>= 20);
for j = 1:length(R);
P_new_tot(j)= P_tot(R(j),C(j));
T_pv_new(j) = T_pv(R(j),C(j));
Eta_new_tot(j) = Eta_tot(R(j),C(j));
end
T_P_tot = vertcat(P_new_tot,T_pv_new, n_k(C),rc_i(R),Eta_new_tot);
[X,Y] = meshgrid(rc_i,n_k);
subplot(2,2,1)
plot(rc_i,P_tot);
title('Subplot 1: Total Power vs Variations in Concentration Ratio');
xlabel('Concentration Ratios')
ylabel('Total Power (W)')
legendCell = cellstr(num2str(n_k', 'n=%-d'));
legend(legendCell, 'Location', 'southeast')
subplot(2,2,2)
plot(rc_i,Eta_tot);
xlabel('Concentration Ratios')
ylabel('Total Efficiency')
legendCell = cellstr(num2str(n_k', 'n=%-d'));
legend(legendCell, 'Location', 'southeast')
title('Subplot 2: Total Efficiency vs Variations in Concentration Ratio');
subplot(2,2,[3,4])
plot(rc_i,T_pv);
xlabel('Concentration Ratios')
ylabel('Temperature of the PV Cell (K)')
legendCell = cellstr(num2str(n_k', 'n=%-d'));
legend(legendCell, 'Location', 'southeast')
title('Subplot 3: T_PV vs Variations in Concentration Ratio');
surf(X,Y,Eta_tot)
rotate3d on
surf(X,Y,Eta_tot)
rotate3d on
xlabel('Concentration Ratios')
ylabel('Thermoelectric Pairs')
zlabel('Total Efficiency')
title('Total Efficiency vs Concentration ratio vs No. of Thermoelectric Pairs')
surf(X,Y,P_tot)
rotate3d on
xlabel('Concentration Ratios')
ylabel('Thermoelectric Pairs')
zlabel('Total Power')
title('Total Power vs Concentration ratio vs No. of Thermoelectric Pairs')
surf(X,Y,T_pv)
```

```
rotate3d on
xlabel('Concentration Ratios')
ylabel('Thermoelectric Pairs')
zlabel('Temperature of Photovoltaic Cell (K)')
title('T_pv vs Concentration ratio vs No. of Thermoelectric Pairs')
function [P, Eta,Q_pv, V_L,I_L] = task_2(R_L, Ly, Lz, Id,rc,T, Vg)
T_p=[293;303;313;333;353];
V_oc=[0.39;0.37;0.35;0.32;0.27];
T_pv = horzcat(T_p,ones(5,1));
co = T_pv\V_oc;
V_oc = [(T), 1]*co;

%constants
Kb= 1.38*10^-23;
qe= 1.6*10^-19;
L=V_oc*qe/(Kb*(T));
I0_Iv = 1/(exp(L)-1);
A_pv = (Ly*Lz)/10000;
%I_L = V_L / R_L;
%Eta = V_L * I_L / (p * A_pv);
P_0= Id*rc*A_pv;

phi = (Id*rc*2.404*15)/(pi^4*Kb*(5778));

syms I_L

fun= @(x) (x.^2)./(exp(x)-1);

xmin = qe*Vg/(Kb* 5778);

inte = integral(fun,xmin,Inf);

phi_g = phi * 0.416 * inte;

Iv = phi_g * qe * A_pv;

I0 =Iv * I0_Iv;

eqn= (Kb*(T)/qe)*log(((Iv- I_L)/I0)+1)- I_L*R_L==0;
I_L_sol = solve(eqn,I_L);
I_L = double(subs(I_L_sol));
V_L = I_L*R_L;
Eta = V_L*I_L/ (Id*rc*A_pv);
P = I_L*V_L;
Q_pv = -V_L*I_L + P_0;
end

function [Z, Rlam_min,  T_C,Eta_te, W,Q_H, Q_C, V_L_te, I_L_te] = task_3_2(alpha, lam_A, lam_B,↙
rho_A, rho_B, n, R_L, T_H, T_sat)
%%%%%add I_l and v_l for this question
Rlam_min = ((lam_A*rho_A)^0.5 + (lam_B*rho_B)^0.5)^2;
Z= alpha^2 / Rlam_min;
A_pv = 0.1^2;
k_w = 6.5;
t_w = 0.4*10^-2;
U_ev = 25;
```

```
% guess T_C = (T_H + T_sat)/2 = (323+95)/2 = 209 K
a = T_sat;
b = T_H+1;
m_max =@(T_C) (1 + 0.5*(T_H+T_C)*Z)^0.5;
R =@(m_max) R_L/(n * m_max);
lam = @(R) Rlam_min / R;
fQ_C =@(T_C) A_pv*(k_w/t_w + U_ev) * (T_C-T_sat);
Eta =@(T_C, m_max) ((T_H-T_C) / T_H) *(((1+m_max)^2 / m_max) * (Z*T_H)^(-1) + 1 + (1/(2*m_max))*
(1+T_C/T_H))^(-1);
R_batt = @(R) n* R;
fW =@(T_C, R_batt) (n^2 * alpha^2 * (T_H-T_C)^2 * R_L )/ ((R_L + R_batt)^2);
fQ_H =@(W,Eta) W/Eta;
Q_CTE = @(Q_H, W)Q_H - W;
E_QC =@(Q_CTE, Q_C) Q_CTE - Q_C;
err = 1;
while err > 10^(-5)
   c=(a+b)/2;
    m_max_a =m_max(a);
    m_max_b =m_max(b);
    m_max_c =m_max(c);
    R_a =R(m_max_a);
    R_b =R(m_max_b);
    R_c =R(m_max_c);
    lam_a = lam(R_a);
    lam_b = lam(R_b);
    lam_c = lam(R_c);
    Q_C_a =fQ_C(a);
    Q_C_b =fQ_C(b);
    Q_C_c =fQ_C(c);
    Eta_a =Eta(a, m_max_a);
    Eta_b =Eta(b, m_max_b);
    Eta_c =Eta(c, m_max_c);
    R_batt_a = R_batt(R_a);
    R_batt_b = R_batt(R_b);
    R_batt_c = R_batt(R_c);
    W_a =fW(a, R_batt_a);
    W_b =fW(b, R_batt_b);
    W_c =fW(c, R_batt_c);
    Q_H_a =fQ_H(W_a,Eta_a);
    Q_H_b =fQ_H(W_b,Eta_b);
    Q_H_c =fQ_H(W_c,Eta_c);
    Q_CTE_a = Q_CTE(Q_H_a, W_a);
    Q_CTE_b = Q_CTE(Q_H_b, W_b);
    Q_CTE_c = Q_CTE(Q_H_c, W_c);
    E_QC_a =E_QC(Q_CTE_a, Q_C_a);
    E_QC_b =E_QC(Q_CTE_b, Q_C_b);
    E_QC_c =E_QC(Q_CTE_c, Q_C_c);
    if E_QC_a>0 &&  E_QC_c < 0
        b=c;
    else
        a=c;
    end
    err = abs(E_QC_c);
end
```

```
T_C = c;
W = W_c;
Q_H = Q_H_c;
Q_C = Q_C_c;
lam_batt = n*lam_c;
R_batt_2 = R_batt_c;
Eta_te = Eta_c;
syms I_L_te V_L_te
eqn1=V_L_te- (n*alpha*(T_H-T_C))-R_batt_2*I_L_te==0;
eqn2= Q_H- n*alpha*I_L_te*T_H- lam_batt*(T_H-T_C)-0.5*((I_L_te)^2)*R_batt_2==0;
%%%%%need to add task_2
[V_L_sol, I_L_sol] = solve([eqn1,eqn2],V_L_te,I_L_te,'IgnoreAnalyticConstraints', true);
V_L_2= double(subs(V_L_sol))>0;
V_L_te = double(V_L_sol(double(subs(V_L_sol))>0));
I_L_te = double(I_L_sol(double(subs(I_L_sol))>0));

end
```

```
lamb_cont= 300;
Id = 1080;
rc = 15;
Ly = 10 ;
Lz = 10 ;
Vg = 1.1;
fg = 265 ;
Rs = 0;
R_L_pv = 0.0070;
T_sat = 90.2 ;
errr = 1;
alpha= 0.0017;
lam_A= 0.032;
lam_B= 0.021;
rho_A=0.0020;
rho_B=0.0030;
n=12;
R_L=0.1;
i =1;
a1 = 95;
b1= 350;
P_sun= (Id*rc*(0.1)^2*0.95);
while errr > 10^(-5)
    c1 = (a1+b1)/2;
    T_H(i,:)= [a1,c1,b1];
[Z(i,1), Rlam_min(i,1),  T_C(i,1),Eta_te(i,1), W(i,1),Q_H(i,1), Q_C(i,1), V_L_te(i,1), I_L_te(i,↙
1) ] = task_3_2(alpha, lam_A, lam_B, rho_A, rho_B, n, R_L, T_H(i,1), T_sat);
[Z(i,2), Rlam_min(i,2),  T_C(i,2),Eta_te(i,2), W(i,2),Q_H(i,2), Q_C(i,2), V_L_te(i,2), I_L_te(i,↙
2) ] = task_3_2(alpha, lam_A, lam_B, rho_A, rho_B, n, R_L, T_H(i,2), T_sat);
[Z(i,3), Rlam_min(i,3),  T_C(i,3),Eta_te(i,3), W(i,3),Q_H(i,3), Q_C(i,3), V_L_te(i,3), I_L_te(i,↙
3) ] = task_3_2(alpha, lam_A, lam_B, rho_A, rho_B, n, R_L, T_H(i,3), T_sat);
E_Q(i,:) = P_sun - Q_H(i,:);
if E_Q(i,1)>0 &&  E_Q(i,2) < 0
        b1=c1;
    else
        a1=c1;
    end
errr = abs(E_Q(i,2));
i = i+1;
end
T_H = T_H(i-1,2);

[Z, Rlam_min,  T_C,Eta_te, W,Q_H, Q_C, V_L_te, I_L_te ] = task_3_2(alpha, lam_A, lam_B, rho_A,↙
rho_B, n, R_L, T_H, T_sat);
T_C = T_C
Q_H = Q_H
V_L = V_L_te
I_L_te = I_L_te
W = W
P_tot = W
Eta = Eta_te
Eta_tot = P_tot/ (Id*rc*(0.1^2)*0.95)


function [Z, Rlam_min,  T_C,Eta_te, W,Q_H, Q_C, V_L_te, I_L_te] = task_3_2(alpha, lam_A, lam_B,↙
rho_A, rho_B, n, R_L, T_H, T_sat)
```

```
%%%%%add I_l and v_l for this question
Rlam_min = ((lam_A*rho_A)^0.5 + (lam_B*rho_B)^0.5)^2;
Z= alpha^2 / Rlam_min;
A_pv = 0.1^2;
k_w = 6.5;
t_w = 0.4*10^-2;
U_ev = 25;

% guess T_C = (T_H + T_sat)/2 = (323+95)/2 = 209 K
a = T_sat;
b = T_H+1;
m_max =@(T_C) (1 + 0.5*(T_H+T_C)*Z)^0.5;
R =@(m_max) R_L/(n * m_max);
lam = @(R) Rlam_min / R;
fQ_C =@(T_C)A_pv*  (k_w/t_w + U_ev) * (T_C-T_sat);
Eta =@(T_C, m_max) ((T_H-T_C) / T_H) *(((1+m_max)^2 / m_max) * (Z*T_H)^(-1) + 1 + (1/(2*m_max))*↙
(1+T_C/T_H))^(-1);
R_batt = @(R) n* R;
fW =@(T_C, R_batt) (n^2 * alpha^2 * (T_H-T_C)^2 * R_L )/ ((R_L + R_batt)^2);
fQ_H =@(W,Eta) W/Eta;
Q_CTE = @(Q_H, W)Q_H - W;
E_QC =@(Q_CTE, Q_C) Q_CTE - Q_C;
err = 1;
while err > 10^(-5)
   c=(a+b)/2;
    m_max_a =m_max(a);
    m_max_b =m_max(b);
    m_max_c =m_max(c);
    R_a =R(m_max_a);
    R_b =R(m_max_b);
    R_c =R(m_max_c);
    lam_a = lam(R_a);
    lam_b = lam(R_b);
    lam_c = lam(R_c);
    Q_C_a =fQ_C(a);
    Q_C_b =fQ_C(b);
    Q_C_c =fQ_C(c);
    Eta_a =Eta(a, m_max_a);
    Eta_b =Eta(b, m_max_b);
    Eta_c =Eta(c, m_max_c);
    R_batt_a = R_batt(R_a);
    R_batt_b = R_batt(R_b);
    R_batt_c = R_batt(R_c);
    W_a =fW(a, R_batt_a);
    W_b =fW(b, R_batt_b);
    W_c =fW(c, R_batt_c);
    Q_H_a =fQ_H(W_a,Eta_a);
    Q_H_b =fQ_H(W_b,Eta_b);
    Q_H_c =fQ_H(W_c,Eta_c);
    Q_CTE_a = Q_CTE(Q_H_a, W_a);
    Q_CTE_b = Q_CTE(Q_H_b, W_b);
    Q_CTE_c = Q_CTE(Q_H_c, W_c);
    E_QC_a =E_QC(Q_CTE_a, Q_C_a);
    E_QC_b =E_QC(Q_CTE_b, Q_C_b);
    E_QC_c =E_QC(Q_CTE_c, Q_C_c);
    if E_QC_a>0 &&  E_QC_c < 0
```

```
        b=c;
    else
        a=c;
    end
    err = abs(E_QC_c);
end

T_C = c;
W = W_c;
Q_H = Q_H_c;
Q_C = Q_C_c;
lam_batt = n*lam_c;
R_batt_2 = R_batt_c;
Eta_te = Eta_c;
syms I_L_te V_L_te
eqn1=V_L_te- (n*alpha*(T_H-T_C))-R_batt_2*I_L_te==0;
eqn2= Q_H- n*alpha*I_L_te*T_H- lam_batt*(T_H-T_C)-0.5*((I_L_te)^2)*R_batt_2==0;
%%%%%need to add task_2
[V_L_sol, I_L_sol] = solve([eqn1,eqn2],V_L_te,I_L_te,'IgnoreAnalyticConstraints', true);
V_L_2= double(subs(V_L_sol))>0;
V_L_te = double(V_L_sol(double(subs(V_L_sol))>0));
I_L_te = double(I_L_sol(double(subs(I_L_sol))>0));

end
```