

Project 1.

Due September 28, 2023

Part 1.

You may team up with a partner for this project. Do not share information or results with other groups.

Background Information

Boiling heat transfer can be a useful, efficient heat transfer process in two-phase loops used for thermal management systems in terrestrial and aerospace applications. It is centrally important in, for example, Rankine cycle boiling and air-conditioning evaporators. In terrestrial applications, gravity buoyancy is an important mechanism that moves vapor away from the surface, allowing liquid to replace it to sustain the process. In microgravity or zero gravity, this mechanism is eliminated, and vapor tends to accumulate near the surface. If the surface is partially or fully blanketed with a vapor layer (transition or film boiling) the heat transfer rate can become so low that the system performance may be unacceptably low and/or the accumulation of heat may raise the wall temperature to a level that damages equipment. Boiling vaporization of cryogenic fuels and liquid oxygen is also a concern during transfer into and out of storage tanks during space missions. Note that high g levels can be encountered during acceleration of a vehicle, and very low g levels result during travel in deep space or in orbit.

Boiling in high or low gravity conditions is important to space power, propulsion and thermal management systems. NASA has test facilities in the International Space Station to study boiling processes. They have also studied boiling in parabolic flight aircraft, rockets, and drop towers to achieve reduced gravity. The nucleate boiling data used here was obtained aboard the “vomit comet” NASA aircraft (see photos in Fig. 1.1) for boiling of a water/2-propanol mixture.



Figure 1.1. NASA parabolic aircraft microgravity boiling tests.

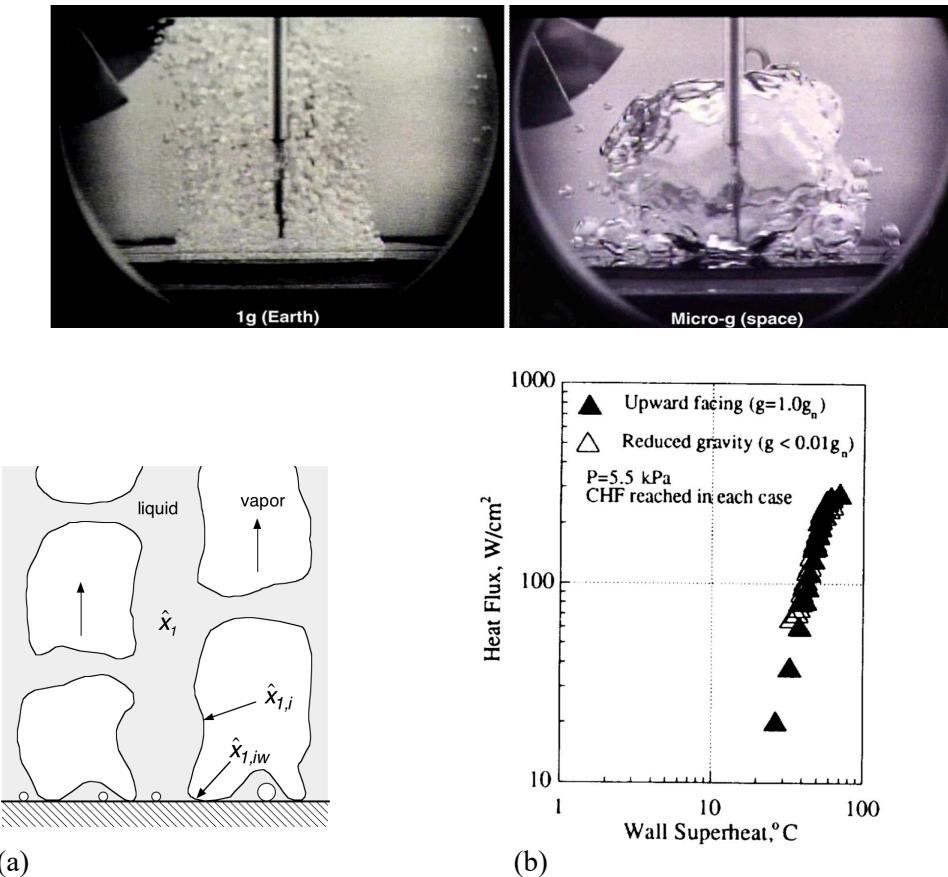


Figure 1.2. (a) Concentration differences due to preferential vaporization of 2-propanol at the interface during nucleate boiling of water/2-propanol mixtures. (b) Nucleate boiling data for a 2-propanol/water mixture at 5.5 kPa pressure at earth normal gravity and reduced gravity (1.2 cm diameter, upward facing copper heated surface, 0.015 mole fraction of 2-propanol in water).

The results of earlier studies indicated that using water/2-propanol mixtures can substantially enhance nucleate boiling heat transfer and the maximum (critical) heat flux. For certain binary fluid mixtures, there is clear evidence that surface tension gradients resulting from concentration differences act to enhance the fluid motion towards the heated surface. As depicted in Fig. 2.a, because the lower vapor pressure alcohol preferentially evaporates at the interface, the lower 2-propanol concentration at the interface near the wall reduces the interfacial concentration of the alcohol there ($\hat{x}_{1,iw} < \hat{x}_{1,i}$) virtually to zero. This, in turn, increases the surface tension close to the contact line at the wall to essentially that for pure water. The resulting gradient in surface tension along the interface produces a Marangoni driven flow of liquid towards the contact line at the wall. In experiments at $1g_{en}$ (earth-normal gravity), the Marangoni effect supplements the effects of buoyancy. However, for binary mixture nucleate boiling experiments under $0.01g_{en}$ reduced gravity conditions, the buoyancy effect is essentially removed, and the ability of Marangoni forces to induce liquid motion towards the surface becomes the dominant liquid delivery mechanism at the surface.

The data from these studies definitively indicate that variation of gravitational acceleration affects the boiling process in water/2-propanol mixtures much differently than in pure water boiling. The objective here

is to apply machine-learning tools to the data to explore the interplay of gravity and Marangoni effects during boiling of a water/2-propanol mixture for a range of gravity acceleration conditions.

The experiments corresponded to the following parameter values:

Circular copper heated surface diameter = 12.5 mm
 Moderately rough and wetted heater surface
 Pressure at heated surface = 5.5 kPa to 9.5 kPa
 Gravitational acceleration = $0.01g_{en}$ to $2g_{en}$
 Wall superheat = 20 to 80 °C
 2-propanol bulk liquid mole fraction ($\hat{x}_{1,b}$) = 0.0 to 0.025
 Fractional change in surface tension due to interface evaporation ($\gamma = (\sigma_w - \sigma_{mix}) / \sigma_{mix}$) = 0.0 to 1.74

The resulting data can be envisioned as an array in which the rows are associated data for one test data point: $[q'', T_w - T_{sat}, g, \gamma, P]$. Consistent with this, in the example code provided, the raw data is organized in an array of the form (see units in nomenclature):

```
[[44.1, 32.5, 0.098, 1.79, 5.5], ]
[47.4, 33.2, 0.098, 1.79, 5.5]
[49.4, 34.2, 0.098, 1.79, 5.5]
:
[59.2, 34.8, 0.098, 1.79, 5.5]
[67.8, 36.3, 0.098, 1.79, 5.5]
```

Nomenclature

$c_{p,l}$	liquid specific heat
g	gravitational acceleration, m/s ²
g_{en}	earth-normal gravitational acceleration, 9.8 m/s ²
h_v	latent heat of vaporization, kJ/kg
k_l	liquid thermal conductivity, W/mK
L_h	heated surface characteristic size, m
q''	heat flux, W/cm ²
q''_{en}	heat flux under earth-normal gravity
Pr_l	liquid Prandtl number
T_{sat}	saturation temperature, °C
$T_w - T_{sat}$	wall superheat, °C
$\hat{x}_{1,b}$	bulk mole fraction of 2-propanol in water/2-propanol solution
$\hat{x}_{1,i}$	interface mole fraction of 2-propanol in water/2-propanol solution
α_l	liquid thermal diffusivity
γ	surface tension parameter $= (\sigma_w - \sigma_{mix}) / \sigma_{mix}$
μ_l	liquid viscosity
ρ_l	liquid density, kg/m ³
ρ_v	vapor density, kg/m ³
σ	surface tension, N/m

Note that skeleton code for this project is provided in files **CodeP1.1F23, **CodeP1.2F23** and **CodeP1.3F23**.**

Task 1.1.

The purposes of Task 1.1 are to install the first portion of the genetic algorithm program **CodeP1.1F23** and to become familiar with package installation and data prep for the program.

Note that in the program, the data is stored explicitly in an array by using commands that first create an array with just one row of data elements

```
ydata = [[44.1, 32.5, 0.098, 1.79, 5.5]]
```

And then uses the append function to add more rows:

```
ydata.append([47.4, 33.2, 0.098, 1.79, 5.5])  
ydata.append([49.4, 34.2, 0.098, 1.79, 5.5])
```

This creates an array of the form

```
[[44.1, 32.5, 0.098, 1.79, 5.5],  
 [47.4, 33.2, 0.098, 1.79, 5.5],  
 [49.4, 34.2, 0.098, 1.79, 5.5],  
 :  
 [207.5, 50.9, 0.098, 1.71, 5.5]]
```

Note that for this format, for `ydata[i][j]`: j is column, i is row downward – both start at 0. So, you access an element as `ydata[row][column]`. This is an array that is essentially a list of lists.

(a) Install this code into a first cell of a new Anaconda notebook. Run the code, which will simply print the array and a single array element in the output region below the cell. Inspect the output to confirm it matches the array in the Appendix on the last page of this write-up. Note that in this code I have commented-out portions of the data with pressures other than 5.5 kPa. More on that in the next Task.

(b) Using the data in the first two groups in the Appendix listing (for $g = 0.098$ and 9.8 m/s^2) make a log-log plot of heat flux versus wall superheat for these two gravity levels to get a sense of how strongly heat flux varies with gravity and superheat. You can do this using a separate Python program (recommended) or you can use another platform such as Excel or Matlab if you prefer. A log-log plot is done in the second python code file so you can see an example there.

Task 1.2.

Install **CodeP1.2 F23** - either append it to the end of **CodeP1.1 F23** to make one program, or copy it in the next cell in the Anaconda notebook so that running both cells in sequence will execute the genetic algorithm for simple equation models used in this Task.

The objective here is to use machine-learning tools to determine how the heat flux varies with the other parameters in the multivariate data. The data reflect the combined variation with these parameters, and the objective of the machine learning analysis is to explore what the individual dependencies are. Although this is not a simple task, genetic algorithms thrive when epistasis gene interaction (suppression of one gene's effect by another) is medium to high. Note, for example, that for pure fluid, in most nucleate boiling processes, heat

flux is well approximated as a power law function of superheat $q'' \sim (T_w - T_{sat})^n$ where n is about 3. Boiling heat transfer data also suggest that for pure fluids, heat flux is roughly a power law function of gravitational acceleration $q'' \sim g^n$, where n is in the range $0.01 < n < 0.1$ for $g < 0.1 g_{en}$, and n is on the order of 0.25 for $g/g_{en} > 0.1$. These trends indicate that heat flux approaches zero as gravitational acceleration approaches zero. For the water solution with small amounts of 2-propanol mixture considered here, Marangoni (variable surface tension) effects result in a finite heat flux even when $g \rightarrow 0$. Hence, the simple power law variation of q'' with g is not expected to apply for water/2-propanol mixtures. In addition, published correlations fit to boiling data imply that the nucleate boiling heat flux exhibits a dependence on fluid properties. Since, for a given fluid, these are taken to be saturated liquid and/or vapor properties, which are a function of saturation pressure for a given fluid, the dependence on these properties could be modeled as a power law dependence on pressure.

The combined **CodeP1.1F23** and **CodeP1.2F23** program is set up to only access the black-text data in the Appendix which has a fixed surface tension parameter and pressure. The heat flux for these data are only a function of superheat and gravitational acceleration. Given the observations above about the dependence of heat flux on superheat and gravity, we adopt the following postulated relation for the dependence of heat flux on superheat and gravitational acceleration:

$$q'' = n_1 (T_w - T_{sat})^{n_2} g^{n_3} \quad (1.1)$$

To apply the genetic algorithm methodology described above, a suitable error function must be defined for a given experimental data point and set of constants n_i . This could be defined to be just the difference between the experimentally measured heat flux and the right side of Eq. (1.1) evaluated at the measured $T_w - T_{sat}$, g and P . However, here we take the natural log of both sides to convert Eq. (1) to:

$$\ln q'' = \ln(n_1) + n_2 \ln(T_w - T_{sat}) + n_3 \ln(g) \quad (1.2)$$

Working with Eq. (1.2) provides an equation that is more linear in the coefficients, and therefore may result in a more well-behaved solution process. Based on Eq. (1.2), an error function $f_{err,i}$ for a given data point i can be defined as

$$f_{err,i} = -\ln q''_{data,i} + \ln(n_1) + n_2 \ln(T_w - T_{sat})_{data,i} + n_3 \ln(g_{data,i}) \quad (1.3)$$

In the program provided, the number of organisms (solutions) N_S is taken to be equal to the number of data points N_D , so for each generation, each solution can be compared to a different data point and all the data is compared in each generation. The order of the solutions in the array that holds the solution constants is constantly changing due to mating and selection, so the pairing is random.

If there are N_D data points in the data set to be analyzed, the total error function F_{err} is the sum of the fractional absolute value of the error for each data point in the population, N_D :

$$\begin{aligned}
 F_{err} &= \sum_{i=1}^{N_D} \left| f_{err,i} \right| / \left| \ln q''_{data,i} \right| \\
 &= \sum_{i=1}^{N_D} \left| -\ln q''_{data,i} + \ln(n_1) + n_2 \ln(T_w - T_{sat})_{data,i} \right. \\
 &\quad \left. + n_3 \ln(g_{data,i}) \right| / \left| \ln q''_{data,i} \right|
 \end{aligned} \tag{1.4}$$

Note that RMS error could be used instead of absolute error. The program is set up to use absolute error.

The assembled code will have a different value of $[F_{err}]_{mean} = F_{err}/N_D$ for each generation. The code seeks the minimum value of $[F_{err}]_{mean}$ achieved during the genetic algorithm simulation. The population average constants n_1, n_2, n_3 for that value can be interpreted as providing a best fit. Once the code is fully assembled, run it to see if it successfully achieves a minimum error $[F_{err}]_{mean} = F_{err}/N_D$ less than 0.04. Try running the code for different initial guesses to see how sensitive the output is to initial guesses. Note that the farther the initial guesses are from values that minimize the error, the more generations may be required to achieve low error. In that case, you may want to increase the number of generation NGEN as well. In your report summarize your results for 4 different sets of initial guesses for n_1, n_2, n_3 . If you choose initial guesses that do not result in convergence, try changing the maximum magnitude of the random perturbations for each constant. Note that the skeleton code provided sets these random perturbations (mutations) in the range $\pm 9\%$. If you increase the maximum percentage, larger random changes in the constants n_1, n_2, n_3 will be introduced, which can allow the algorithm to more quickly explore a broader range of possible solutions. This is better when guesses close to the right values are not known. So try varying the maximum level of random perturbations as you try different initial guesses, especially if the guesses seem to have difficulty converging to a good solution with low error.

Using constants n_1, n_2, n_3 for your best fit to the data, construct a log-log plot with the measured heat flux values as the vertical axis and predicted heat flux values for Eq. (1.1) with your best fit constants for the horizontal axis. Be sure to plot the combinations of (q''_{pred}, q''_{exp}) as discrete points. Note that if the fit were perfect, the points would all lie along a line with a slope of 1 on the log-log plot. Also, determine the rms deviation of the q''_{exp} data from the q''_{pred} predictions. If the uncertainty in the measured heat flux data is estimated to be $\pm 10\%$. Is your fit tighter than the uncertainty (is the average deviation less than the uncertainty)? Assess your fit in the context of this question. Specifically, would you say your fit rms deviation is comparable to the noise in the data, or above or below it? Discuss how this assessment affects the accuracy of predicted trends and accuracy of predictions for this fitted relation.

Task 1.3

Make a copy of your combined **CodeP1.1F23** and **CodeP1.2F23** program that you can modify to train a **five constant model** that includes variation of pressure P and the surface tension parameter γ . Specifically, this new program will be designed to find the set of constants n_1 through n_5 in the performance equation

$$q'' = n_1 (T_w - T_{sat})^{n_2} (g + n_4 g_{en} \gamma)^{n_3} P^{n_5} \tag{1.5}$$

that best fits an expanded data set that includes variable γ and pressure. Taking the natural log of both sides to convert Eq. (1.5) to:

$$\begin{aligned}\ln q'' &= \ln(n_1) + n_2 \ln(T_w - T_{sat}) \\ &\quad + n_3 \ln(g + n_4 g_{en} \gamma) + n_5 \ln P\end{aligned}\tag{1.6}$$

Based on this equation, an error function $f_{err,i}$ for a given data point i can be defined as

$$\begin{aligned}f_{err,i} &= -\ln q''_{data,i} + \ln(n_1) + n_2 \ln(T_w - T_{sat})_{data,i} \\ &\quad + n_3 \ln(g_{data,i} + n_4 g_{en} \gamma_{data,i}) + n_5 \ln P_{data,i}\end{aligned}\tag{1.7}$$

If there are N_D data points in the data set to be analyzed, the total error function F_{err} is the sum of the fractional absolute value of the error for each data point in the population, N_D :

$$\begin{aligned}F_{err} &= \sum_{i=1}^{N_D} \left| f_{err,i} \right| / \left| \ln q''_{data,i} \right| \\ &= \sum_{i=1}^{N_D} \left| -\ln q''_{data,i} + \ln(n_1) + n_2 \ln(T_w - T_{sat})_{data,i} \right. \\ &\quad \left. + n_3 \ln(g_{data,i} + n_4 g_{en} \gamma_{data,i}) + n_5 \ln P_{data,i} \right| / \left| \ln q''_{data,i} \right|\end{aligned}\tag{1.8}$$

To set up this new analysis, your modifications to the three constant **CodeP1.1F23** and **CodeP1.2F23** program should include the following:

- (i) uncomment the red text lines in the ydata array definition to add that additional data
- (ii) change ND and NS to 77
- (iii) modify the following lines of code to compute the appropriate error and heat flux quantities for the five constant model:

```
Ferr[i] = n[i][0]*lydata[i][0] + math.log(n[i][1]) + n[i][2]*lydata[i][1]
Ferr[i] = Ferr[i] + n[i][3]*math.log( ydata[i][2] )

for i in range(ND):
    Ferravgn[i] = -1.*lydata[i][0] + math.log(n1avg[k]) + n2avg[k]*lydata[i][1]
    Ferravgn[i] = Ferravgn[i] + n3avg[k]*math.log( ydata[i][2] )

for i in range(ND):
    qpppred[i] = n1min*(ydata[i][1]**n2min) * ((ydata[i][2])**n3min)
```

- (iv) To include constants n_4 and n_5 in the mating, uncomment the following lines of code:

```
if (numpy.random.rand() < 0.5):
    ntemp[nkeep+j+1][4] = n[nmate1][4]*(1.+0.09*2.*((0.5-numpy.random.rand())) # property 4,mutation added
else:
    ntemp[nkeep+j+1][4] = n[nmate2][4]*(1.+0.09*2.*((0.5-numpy.random.rand())))
```

```
if (numpy.random.rand() < 0.5):
    ntemp[nkeep+j+1][5] = n[nmate1][5]*(1.+0.092.*((0.5-numpy.random.rand()))) # property 5, mutation added
else:
    ntemp[nkeep+j+1][5] = n[nmate2][5]*(1.+0.092.*((0.5-numpy.random.rand())))
```

(v) For the final output of constants, modify the following lines of code to output all five constants with the corresponding error parameter:

```
print('ENDING: pop.avg n1-n3,aFerrmean:', n1avg[k], n2avg[k], n3avg[k], aFerrmeanavgn[k])
print('MINIMUM: avg n1-n3,aFerrmeanMin:', n1min, n2min, n3min, aFerrmeanavgnMin)
print('TIME AVG: avg n1-n3,aFerrmean:', n1ta, n2ta, n3ta, aFerrta)
```

(vi) To alter the first plot for the five constant model, delete or comment-out the first line of code below and uncomment the three lines following it:

```
plt.legend(['aFerrmeanavgn', 'n1 avg', 'n2 avg', 'n3 avg'], loc='lower left')
#plt.plot(gen, n4avg)
#plt.plot(gen, n5avg)
plt.legend(['aFerrmeanavgn','n1 avg','n2 avg','n3 avg','n4 avg','n5 avg'],loc='upper right')
```

(vii) Change the initial guesses to:

```
n0i = -1.0
n1i = 0.000476
n2i = 3.028
n3i = 0.2249
n4i = 1.054
n5i = 0.217
```

Run this new model to determine n_1 through n_5 to best fit the data. Run for a range of initial guesses to be sure you have a best fit (aFerrmeanavgnMin less than 0.03). Summarize the constants n_1 through n_5 for your best fit to the data in a table. Use the resulting curve-fit equation to create a surface plot of $q''/(T_w - T_{sat})^{n_2}$ versus g and γ for $1.0 < g < 20 \text{ m/s}^2$ and $0.0 < \gamma < 2.0$ at a pressure of 10 kPa.

Project 1, Part 2:

Background

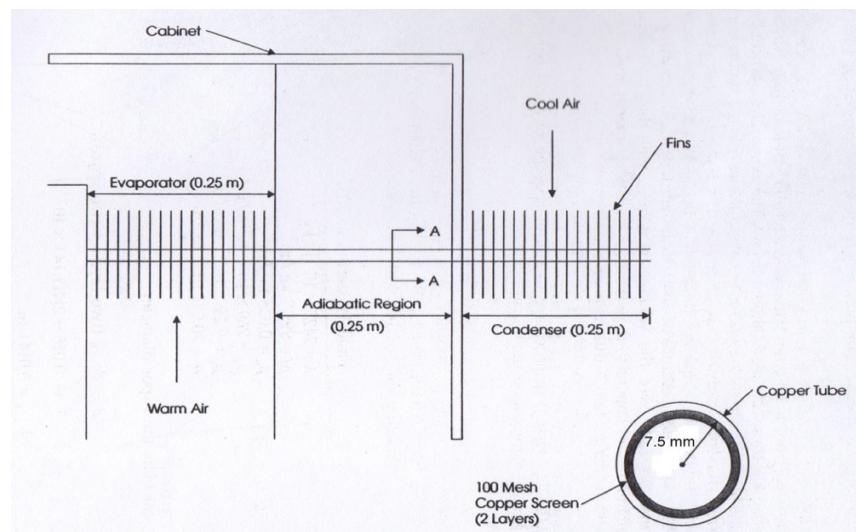


Figure 2.1. Heat pipe for cooling electronics cabinet

Part 2 of this project will focus on the design of the heat pipe heat exchanger depicted in Figure 1, which will provide cooling of a cabinet containing electronic components that dissipate heat. The objective is to transfer

the heat from inside to air outside the cabinet. Fans circulate air over the finned evaporator and condenser portions of the heat pipe to facilitate convection heat transfer from the air inside and to the air outside the cabinet. The schematic in Figure 2.2 depicts the flow circulation within the heat pipe that accomplishes the transport of heat.

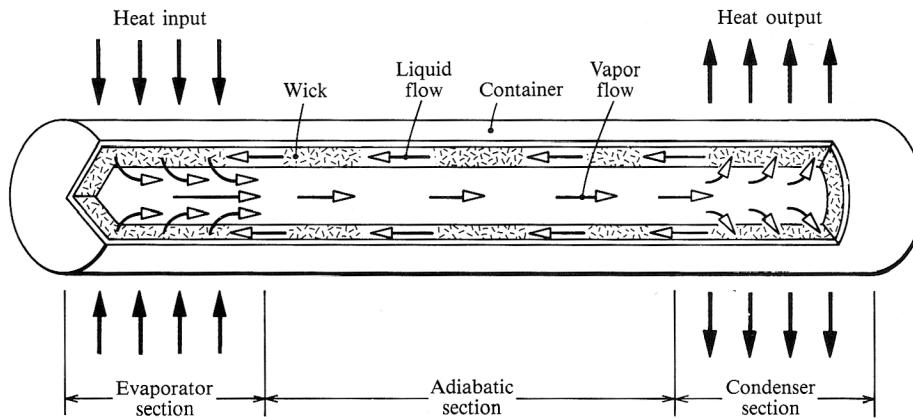


Figure 2.2.

Note that the flow inside the heat pipe is driven by capillary effects alone. Central to the heat pipe's operation is the wick structure, which can take any of the forms shown in Fig. 2.3.

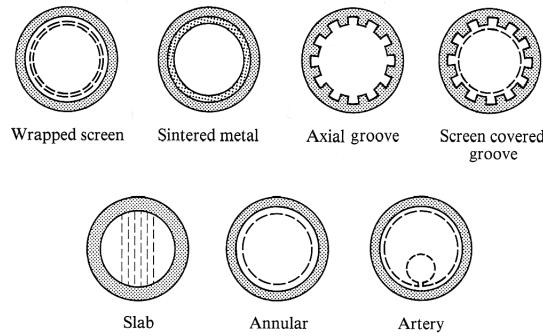


Figure 2.3

Flow in the wick is driven by a pressure difference between the condenser and evaporator portions of the wick as a result of different interface radii of curvature there (see Fig. 2.4).

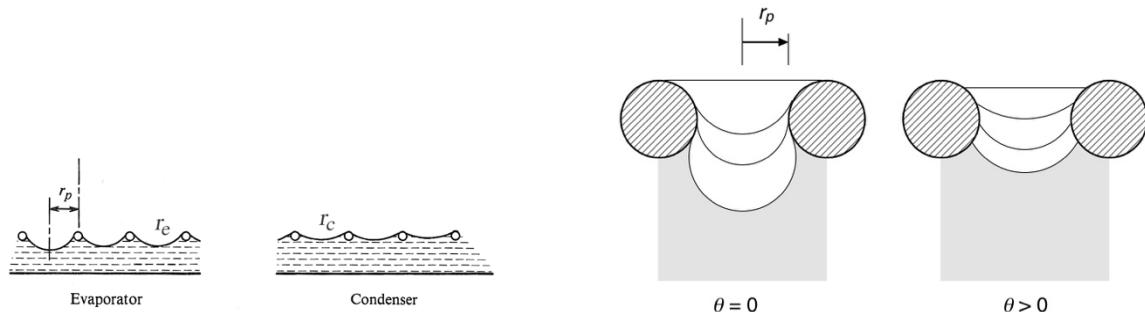


Figure 2.4a.

Figure 2.4b. $r_e \cong r_p / \cos\theta$.

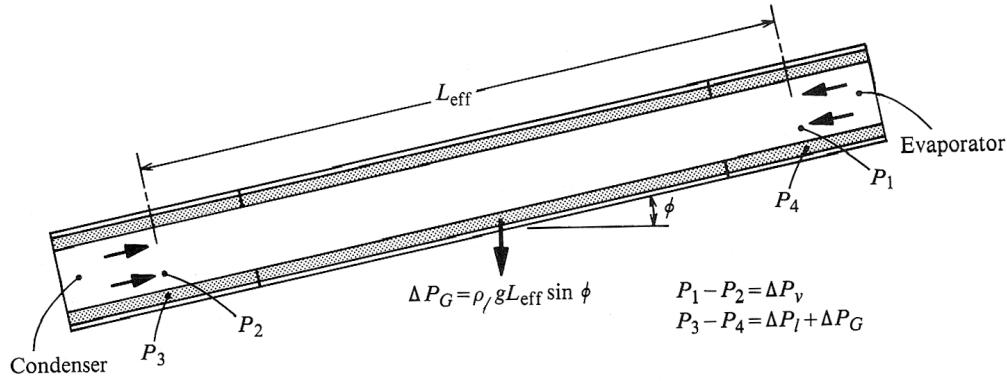


Figure 2.5.

As shown in Fig. 2.6, finned structures on the outside of the heat pipe increase the heat transfer surface area, thereby reducing heat transfer resistance associated with convection heat transfer to the air. Here we neglect the resistance to heat transfer in the copper wall of the heat pipe and assume it is all at one temperature.

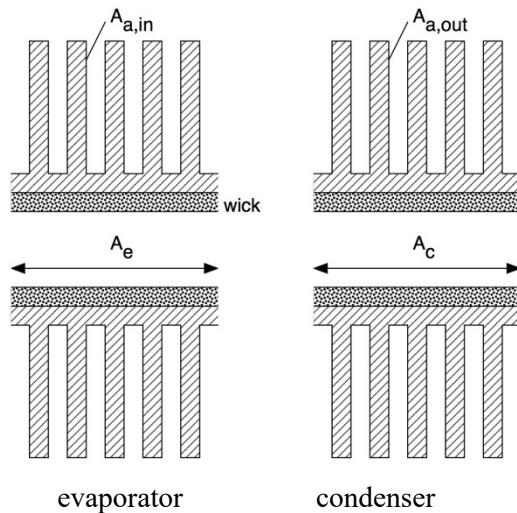


Figure 2.6.

If the heat transfer rate is \dot{q} , it follows that for the evaporator and condenser sections, the thermal resistances of the wick and the fins can be computed using an electrical resistance analogy as

$$\frac{1}{(UA)_e} = \frac{t_w}{k_w A_e} + \frac{1}{h_{a,in} A_{a,in}} \quad (2.1)$$

$$\frac{1}{(UA)_c} = \frac{t_w}{k_w A_c} + \frac{1}{h_{c,ref} A_{a,out}} \quad (2.2)$$

where A_e and A_c are the wick areas of the evaporator and condenser sections, respectively.

A detailed physics-based model of the heat pipe performance can be constructed accounting for the flow of liquid in the wick driven by capillary forces, and heat transport in and out of the heat pipe via fins (as shown in Fig. 2.6) as well as heat transport by condensation, vaporization and flow of vapor inside the heat pipe. Given the heat pipe geometry (D_p , t_w , L_{evap} , L_{adiab} , L_{cond} , $A_{a,in}/A_e$, $A_{a,out}/A_c$), wick characteristics (ω , κ , r_p , θ), air convection coefficients and temperatures ($h_{a,in}$, $h_{a,out}$, $T_{a,in}$, $T_{a,out}$), and working fluid properties, the equations for such a detailed model can be solved simultaneously to determine the operating point for the heat pipe.

However, if the heat pipe is well designed, the operating characteristics of the heat pipe are simplified and can be represented by the following simple model for predicting the heat transfer rate as a function of the source and sink temperatures:

$$\dot{q} = (UA)_c \left(1 + \alpha \frac{h_r}{h_{c,ref}} \right) \left[\frac{(UA)_e(T_{a,in} - T_{a,out})}{(UA)_e + (UA)_c \left(1 + \alpha \frac{h_r}{h_{c,ref}} \right)} \right] \quad (2.3)$$

where

$T_{a,in}$ = temperature of air inside electronics cabinet (°C)

$T_{a,out}$ = temperature of air outside electronics cabinet (°C)

$(UA)_e$ = thermal conductance of the evaporator fin section (W/°C)

$(UA)_c$ = thermal conductance of the condenser fin section with no radiation (W/°C)

$\alpha \frac{h_r}{h_{c,ref}}$ = radiation correction terms for condenser section

$h_r = \sigma(T_{a,in} + T_{a,out})(T_{a,in}^3 + T_{a,out}^3)$ (temperatures here are absolute) (Wm⁻²K⁻¹)

σ = Stephan Boltzmann constant = 5.67×10^{-8} Wm⁻²K⁻⁴

$h_{c,ref}$ = reference convective heat transfer coefficient for condenser fin section = 70 W/m²K

Note that values of the following constants are needed to use this mode for predictive purposes:

$(UA)_e$, $(UA)_c$, and α

The conductances can be computed theoretically by using the equations above that account for the conduction in the finned structures and wick structure in the condenser and evaporator sections. Likewise, the radiation transport from the condenser to the surroundings outside the cabinet can be predicted from an appropriate model if accurate radiation properties of the surfaces are available and idealizations in the model are valid. An Alternative approach is to use a genetic algorithm to find the set of $(UA)_e$, $(UA)_c$, and α values that best fit a collection of experimentally determined performance data for the system.

Task 2.1 Construct a python program like that used in Task 1.2, that uses a genetic algorithm to determine the set of $(UA)_e$, $(UA)_c$, and α values that best fit the data in the accompanying file **CodeP1.3F23.ipynb**. Be sure to divide the data into a proper training set (about 80%) and a randomly selected validation set (about 20%).

Use the genetic algorithm program developed in Task 1.2 as a starting point for your program for this part. Here, replace the data used in Task 1.2 with the data from file **CodeP1.3-F23.ipynb**.

To apply the genetic algorithm, a suitable error function must be defined for a given experimental data point and set of constants n_i . For the form of the equation of interest here, define the error function to be the difference between the experimentally measured heat transfer rate and the right side of the \dot{q} equation above evaluated at the measured $T_{a,in}$ and $T_{a,out}$. Note that here, $n_1 = (UA)_e$, $n_2 = (UA)_c$, and $n_3 = \alpha$, and the error function $f_{err,i}$ for a given data point i can be defined as

$$f_{err,i} = \dot{q}_{data,i} - n_2 \left(1 + n_3 \frac{h_{r,i}}{h_{c,ref}} \right) \left[\frac{n_1(T_{a,in,i} - T_{a,out,i})}{n_1 + n_2 \left(1 + n_3 \frac{h_r}{h_{c,ref}} \right)} \right] \quad (2.4)$$

where $h_{r,i} = \sigma(T_{a,in,i} + T_{a,out,i})(T_{a,in,i}^3 + T_{a,out,i}^3)$, and $h_{c,ref}$ and σ are numerical constants defined above.

In the program used for Task 1.2, the number of organisms (solutions) N_S is taken to be equal to the number of data points N_D , so for each generation, each solution can be compared to a different data point and all the data is compared in each generation. The order of the solutions in the array that holds the solution constants is constantly changing due to mating and selection, so the pairing is random. As before, if there are N_D data points in the data set to be analyzed, the total error function F_{err} is the sum of the fractional absolute value of the error for each data point in the population, N_D :

$$F_{err} = \sum_{i=1}^{N_D} |f_{err,i}| / |\dot{q}_{data,i}|$$

$$F_{err} = \sum_{i=1}^{N_D} \left| \frac{1}{\dot{q}_{data,i}} \left(\dot{q}_{data,i} - n_2 \left(1 + n_3 \frac{h_{r,i}}{h_{c,ref}} \right) \left[\frac{n_1(T_{a,in,i} - T_{a,out,i})}{n_1 + n_2 \left(1 + n_3 \frac{h_r}{h_{c,ref}} \right)} \right] \right) \right| \quad (2.5)$$

The program is to be set up to use mean absolute error.

As in Task 1.2, the assembled code will have a different value of $[F_{err}]_{mean} = F_{err}/N_D$ for each generation. The code seeks the minimum value achieved during the genetic algorithm simulation. The constants n_1, n_2, n_3 for that value can be interpreted as providing a best fit.

Note that in modifying the code from Task 1.2, there are two locations in the code where you will need to change the specified error function, and you will have to modify the section handling the plotting to create a plot of the data value of heat transfer rate \dot{q} versus the value of the heat transfer rate computed using the form of the \dot{q} equation. (2.6) in terms of the n_i values:

$$\dot{q} = n_2 \left(1 + n_3 \frac{h_{r,i}}{h_{c,ref}} \right) \left[\frac{n_1(T_{a,in,i} - T_{a,out,i})}{n_1 + n_2 \left(1 + n_3 \frac{h_r}{h_{c,ref}} \right)} \right] \quad (2.6)$$

Task 2.2 Once the code for Part 2 is fully assembled, run it to see if it successfully achieves a minimum error $[F_{err}]_{mean} = F_{err}/N_D$ less than 0.05. Try running the code for different initial guesses to see how sensitive the output is to initial guesses. Also vary the number of generations NGEN as well to see how it affects the results. In your report summarize your results for 4 different sets of initial guesses for n_1, n_2, n_3 .

Document the mean absolute error obtained in your best fit for the training and validation cases. Try to obtain a fit to the data with a mean absolute error for the validation data that is less than 0.05, and try to get below that. Present a log-log plot of the predicted heat transfer rate versus the data heat transfer rate for both the training data and the validation data. Each combination should be plotted as a discrete circular or square symbol data point in these plots. Also provide a surface plot of \dot{q} as a function of $T_{a,in}$ and $T_{a,out}$ over the range of the data.

Task 2.3.

Assemble a report that summarizes the results of your work on all the Tasks for Part 1 and Part2. Include in your report:

- A statement of how the work in the Tasks was divided between the members of the team.
- In Tasks where you are required to create a new program by modifying an earlier one, summarize the modifications you made to the previous version of the code.
- Be sure to document and changes you made to the basic code structure in the example code provided.
- Be sure to include plots and tables of results specified in the Task descriptions.
- In the report, present a summary of your conclusions regarding how initial guesses of model constants affect the stability of the algorithm and the speed of convergence to a satisfactory fit. Summarize results in tables or plots as appropriate.
- In the report, also provide a discussion of the advantages and disadvantages of raw data analysis versus dimensionless data analysis for this heat transport process.
- Include a copy of each of your programs in an Appendix at the end of the report.

Be sure your report documents the answer to all questions asked in the Task write-up and include all requested explanations, table and plots. Failure to do so will have a negative impact on your grade.

Deliverables:

A pdf of your summary report is due by **9/28/23 @ 11:59 PM.**

Grade will be based on:

- (1) thoroughness of documentation of your analysis and program development
- (2) accuracy of best fit constants
- (3) thoroughness of the assessments of the program performance
- (4) documentation of results, including plots, tables, etc. specified in the Task descriptions
- these should be professional quality

Appendix – data for Part 1

ydata: $[q'', T_w - T_{sat}, g, \gamma, P]$

heat flux $q'' \sim \text{W/cm}^2$, superheat $T_w - T_{sat} \sim ^\circ\text{C}$, gravitational acceleration $g \sim \text{m/s}^2$,
surface tension parameter $\gamma \sim \text{dimensionless}$), pressure $P \sim \text{kPa}$

<pre>[[44.1, 32.5, 0.098, 1.79, 5.5], [47.4, 33.2, 0.098, 1.79, 5.5], [49.4, 34.2, 0.098, 1.79, 5.5], [59.2, 34.8, 0.098, 1.79, 5.5], [67.8, 36.3, 0.098, 1.79, 5.5], [73.6, 37.3, 0.098, 1.79, 5.5], [76.3, 37.8, 0.098, 1.79, 5.5], [85.3, 39.2, 0.098, 1.79, 5.5], [96.5, 39.3, 0.098, 1.79, 5.5], [111.0, 42.3, 0.098, 1.79, 5.5], [124.0, 43.5, 0.098, 1.79, 5.5], [136.2, 45.4, 0.098, 1.79, 5.5], [143.5, 46.7, 0.098, 1.79, 5.5], [154.6, 47.9, 0.098, 1.79, 5.5], [163.1, 48.6, 0.098, 1.79, 5.5], [172.8, 50.9, 0.098, 1.79, 5.5], [184.2, 51.7, 0.098, 1.79, 5.5], [203.7, 56.4, 0.098, 1.79, 5.5], [36.7, 30.2, 9.8, 1.79, 5.5], [55.1, 34.1, 9.8, 1.79, 5.5], [67.5, 35.3, 9.8, 1.79, 5.5], [78.0, 37.8, 9.8, 1.79, 5.5], [92.0, 38.1, 9.8, 1.79, 5.5], [120.0, 44.1, 9.8, 1.79, 5.5], [134.3, 46.9, 9.8, 1.79, 5.5], [150.3, 48.5, 9.8, 1.79, 5.5], [167.0, 49.2, 9.8, 1.79, 5.5], [184.0, 52.7, 9.8, 1.79, 5.5], [196.5, 53.1, 9.8, 1.79, 5.5], [42.4, 28.0, 19.6, 1.79, 9.5], [48.7, 29.3, 19.6, 1.79, 9.5], [54.5, 29.6, 19.6, 1.79, 9.5], [62.1, 28.5, 19.6, 1.79, 9.5], [70.8, 30.5, 19.6, 1.79, 9.5], [73.7, 30.3, 19.6, 1.79, 9.5], [81.8, 30.6, 19.6, 1.79, 9.5], [91.9, 34.5, 19.6, 1.79, 9.5], [103.9, 34.5, 19.6, 1.79, 9.5], [119.1, 35.4, 19.6, 1.79, 9.5], [133.7, 36.8, 19.6, 1.79, 9.5], [139.9, 38.1, 19.6, 1.79, 9.5], [148.3, 39.1, 19.6, 1.79, 9.5], [157.0, 40.0, 19.6, 1.79, 9.5], [169.1, 42.2, 19.6, 1.79, 9.5], [179.2, 43.2, 19.6, 1.79, 9.5], [205.0, 46.0, 19.6, 1.79, 9.5],</pre>	<pre>[42.4, 29.7, 19.6, 1.79, 5.5], [48.7, 31.0, 19.6, 1.79, 5.5], [54.5, 31.2, 19.6, 1.79, 5.5], [70.8, 32.4, 19.6, 1.79, 5.5], [73.7, 31.4, 19.6, 1.79, 5.5], [81.8, 32.5, 19.6, 1.79, 5.5], [91.9, 36.3, 19.6, 1.79, 5.5], [103.9, 36.3, 19.6, 1.79, 5.5], [119.1, 37.2, 19.6, 1.79, 5.5], [133.7, 38.4, 19.6, 1.79, 5.5], [139.9, 39.7, 19.6, 1.79, 5.5], [148.3, 40.9, 19.6, 1.79, 5.5], [157.0, 41.6, 19.6, 1.79, 5.5], [169.1, 43.9, 19.6, 1.79, 5.5], [179.2, 45.0, 19.6, 1.79, 5.5], [205.0, 47.9, 19.6, 1.79, 5.5], [77.0, 41.5, 9.8, 0.0, 7.0], [71.0, 40.5, 9.8, 0.0, 7.0], [66.0, 39.5, 9.8, 0.0, 7.0], [62.0, 38.5, 9.8, 0.0, 7.0], [42.0, 34.0, 9.8, 0.0, 7.0], [60.0, 37.5, 9.8, 0.0, 7.0], [53.0, 37.0, 9.8, 0.0, 7.0], [71.7, 36.4, 0.098, 1.71, 5.5], [81.5, 38.5, 0.098, 1.71, 5.5], [90.7, 39.5, 0.098, 1.71, 5.5], [103.3, 41.6, 0.098, 1.71, 5.5], [117.0, 43.1, 0.098, 1.71, 5.5], [138.6, 45.4, 0.098, 1.71, 5.5], [161.7, 47.9, 0.098, 1.71, 5.5], [207.5, 50.9, 0.098, 1.71, 5.5]]</pre> <p>Validation Data</p> <pre>[[50.9, 37.2, 0.098, 1.79, 5.5], [75.1, 37.7, 0.098, 1.79, 5.5], [99.2, 39.7, 0.098, 1.79, 5.5], [147.7, 47.2, 0.098, 1.79, 5.5], [172.1, 49.7, 9.8, 1.79, 5.5], [49.2, 29.6, 19.6, 1.79, 9.5], [71.6, 30.0, 19.6, 1.79, 9.5], [129.8, 36.4, 19.6, 1.79, 9.5], [173.8, 42.8, 19.6, 1.79, 9.5], [68.7, 32.1, 19.6, 1.79, 5.5], [128.5, 38.0, 19.6, 1.79, 5.5], [69.0, 40.1, 9.8, 0.0, 7.0], [58.2, 37.1, 9.8, 0.0, 7.0], [100.3, 41.2, 0.098, 1.71, 5.5], [156.9, 47.4, 0.098, 1.71, 5.5]]</pre>
---	---