Project 4. Solar Thermal Power and Electronics Cooling

Due date: December 14, 2021

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

Part 1. Introduction

Part 1 of this project will focus on modeling a boiler design like that used in the PS10 solar thermal power plant 15 km west of Seville, Spain. An aerial photo of the plant is shown in Fig.1.

Files to be used:

Part 1
ME249Proj4F21data1a.ipynb
ME249Proj4F21data1b.ipynb
CodeP4.1F21.ipynb

Part 2
ME249Proj4F21data2.ipynb
CodeP4.1F21.ipynb

Fig. 1 (from [1]).

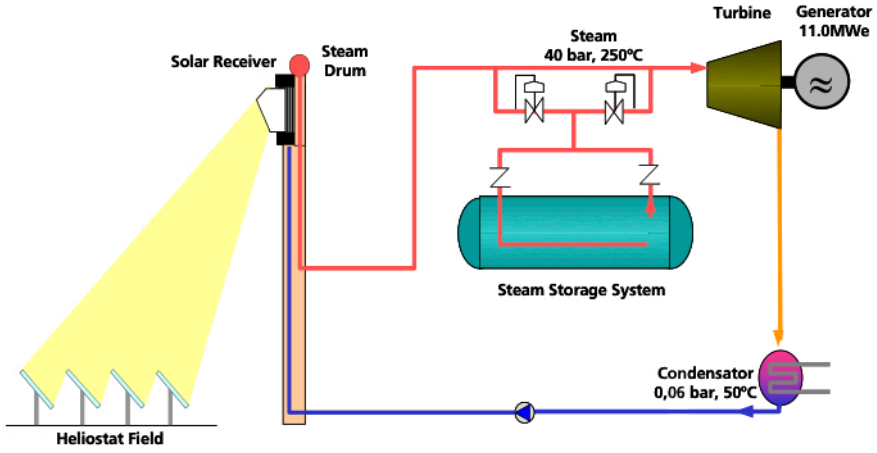


Fig. 2. System schematic (from [1]).

This system uses solar energy to generate saturated steam for power production. In this system, 624 heliostats reflect solar radiation to the absorber atop the solar receiver tower. Each heliostat has an area of 121 m^2. The nominal design intensity of solar radiation delivered to the absorber is 55 MW. Further details of the system are indicated in Table 1 and Fig. 3.

The heliostats focus the solar energy on four riser tube arrays, each of which is 4.8 m wide by 12.0 m tall. These four tube arrays are arranged in a semicircle within the absorber cavity in the receiver tower (see Fig. 4)

Table 1. SP10 system details (from [2]).

General description	
Emplacement	Sanlúcar M. (Sevilla), Lat 37.4°, Lon 6.23°
Nominal power	11.02MWe
Tower height	90m
Receiver technology	Saturated steam
Receiver geometry	Cavity180°, 4 Pannels 5m x 12m
Heliostats	624 @ 121m2
Thermal storage technology	Water/steam
Thermal storage capacity	15MWh, 50min @ 50% Rate
Steam cycle	40 bar 250°C, 2 Pressures
Electric generation	6.3kV, 50Hz -> 66kV, 50Hz
Land	60 Has
Annual electricity production	23.0GWh

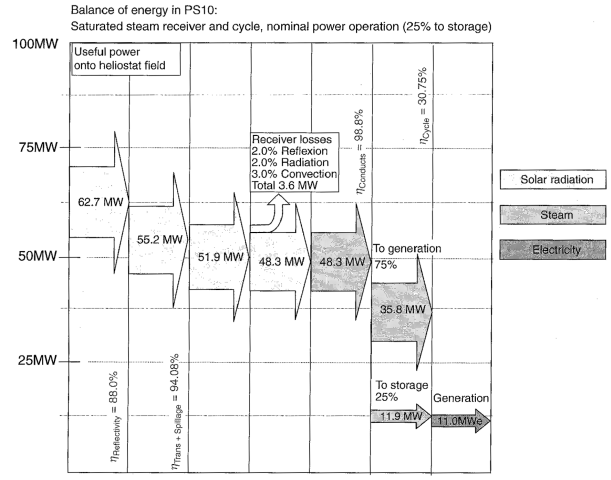


FIGURE 19.61 Energy cascade for PS10 subsystems.

Fig. 3 (from [2]).

Heliostat test onto the cavity of PS10

Figure 4. (from [1])

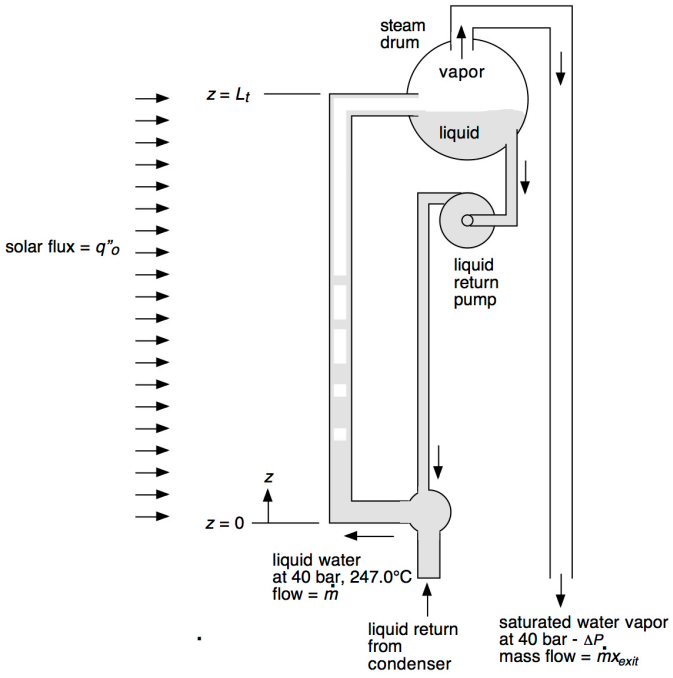
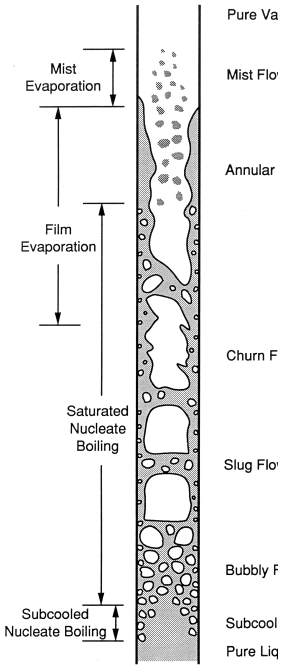


Fig. 5.



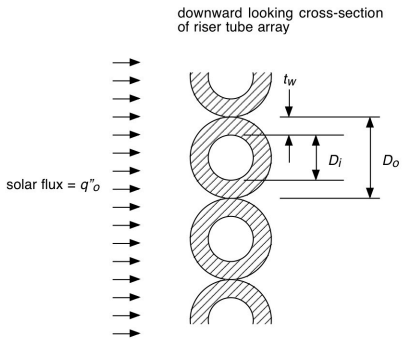


Fig. 6.

Task 1.1	**Task 1.3**
<u>Specified operating (input) parameters:</u> Tube inside diameter D_i (m) Incident solar flux, q''_o (W/m^2) Water mass flow rate $\dot{m}$ (kg/s) <u>Performance (output) parameters:</u> Exit quality, x_e Maximum tube wall temperature $T_{w,\max}$ (deg C)	<u>Specified operating (input) parameters:</u> Tube inside diameter D_i (m) Incident solar flux, q''_o (W/m^2) Exit quality, x_e Maximum tube wall temperature $T_{w,\max}$ (deg C) <u>Performance (output) parameters:</u> Water mass flow rate $\dot{m}$ (kg/s)

Performance test data for this type of system is provided as a dataset that includes the parameters in the table above. Data set ME249Proj4F21data1a.ipyn (with input data [$D_i, q''_o, \dot{m}$] and outputs [x_e, $T_{w,\max}$]) provided for this project is a collection of data for flow boiling of water in a vertical tube at a design saturation pressure of 4.0 MPa. The inlet flow is saturated liquid at 4.0 MPa.

In Task 1.3, you will be asked to consider [$D_i, q''_o, x_e, T_{w,\max}$] as inputs and the corresponding mass flow rate $\dot{m}$ as the predicted output. File ME249Proj4F21data1b.ipyn contains data in a format to facilitate analysis in this framework.

Task 1.1
Consider CodeP4.1F21.ipynb (which is equivalent to CodeP3.1.2 used in Project 3), provided with this project, to be a starting-point skeleton code for the activities in this task.

(a) For the data set ME249Proj4F21data1a.ipyn, determine the median value for each parameter and normalize the data by dividing each parameter value by its median value.

(b) Take the normalized data and separate it randomly into two data sets: a training set with 3/4ths of the data and a second validation set with 1/4th of the data.

(c) Substitute the normalized training data into the skeleton code and convert it to a neural network model that can be trained using the training data set. For this first model, use a `keras.sequential` network with having these specs:

- specify a `RandomUniform` initializer (see skeleton code)
- an inlet layer having 6 neurons with `activation=K.elu, input_shape=[3]`
- 3 hidden layers with 8, 16, and 8 neurons
- an outlet layer with 2 neurons with no activation function

Set `activation=K.elu` for all the neurons except the outlet layer, and use the `RMSprop` optimizer, as configured in the skeleton program. Using the `model.fit` routine as configured in the skeleton program is recommended.

(d) Train the neural network model constructed in part (c) using the training data. Try to get the mean absolute error below 0.025 if possible. You can adjust the initialization and/or the learning parameter a bit to try to improve convergence. Lower is better, but the main objective it that it be small compared to one.

(e) Compare the trained model predictions to the training data set, report the mean absolute error for the fit, and create a log-log plot of predicted exit quality vs. data value exit quality for each set of data point operating conditions. Do the same for the maximum wall temperature. Plot these as a scatter plots of data points and show the slope = 1 line for comparison.

(f) Repeat the steps of part (e), comparing the model predictions this time to the normalized validation data. Report the mean absolute error and include the log-log plot

specified in (e) for these data in the summary report. Assess the results for evidence of overfitting, and include your assessment in your report.

(g) Taking the heat flux q''_o to be fixed at 750 kW/m^2, use the trained model created in this task to generate predictions of the exit quality x_e and maximum wall temperature $T_{w,\max}$ outputs for 7 mm $< D_i <$ 13 mm and 0.05 $< \dot{m} <$ 0.15 kg/s, and create a surface plots of the exit quality x_e and maximum wall temperature $T_{w,\max}$ as a function of D_i and $\dot{m}$. Which combination of these would you recommend if you want an output quality of about 0.75 and a maximum wall temperature no more than about 310 °C?

Task 1.2

Repeat steps (a)-(g) in Task 1.1 to construct, train and assess a neural network model with the same specs as Task 1.1, except for the following change to the network design:

Use 4 hidden layers (instead of 3) having 8, 12, 16, and 8 neurons
Add a dropout layer after each of the hidden layers, like that added to the image processing model in the example file ME249-17b_F21_CNNtrial2.ipynb discussed in lecture (use a value of 0.25 for the Dropout() layer argument for all the layers – see the example file ME249-17b_F21_CNNtrial2.ipynb.)

Note that adding another hidden layer makes the network deeper, with more capability to model complex detail. This tends to make the network more prone to overfitting. Adding the dropout layers can compensate for this. With this new model, repeat steps (a)-(g) in Task 1.1, and do this additional step (h):

(h) Compare the results for this task with those for Task 1.1, and assess whether (1) this model better matches the data, and (2) whether there are any signs of overfitting in the model with dropout layers. Summarize your conclusions in your report.

Task 1.3

Use the CodeP4.1F21.ipynb provided with this project as the starting point skeleton code for the activities in this task.

(a) Data set ME249Proj4F21data1b.ipyn, contains the arrays for the input data [$D_i, q''_o, x_e, T_{w,\max}$] and the output parameter [$\dot{m}$] for the flow boiling depicted in Fig. 5. Determine the median value for each parameter and normalize the data by dividing each parameter value by its median value.

(b) Take the normalized ME249Proj4F21data1b.ipyn data and separate it randomly into two data sets: a training set with 3/4ths of the data and a second validation set with 1/4 of the data.

(c) Substitute the normalized training data into the skeleton code and convert the code to a neural network model that can be trained using the training data set. For this model,

[$D_i, q''_o, x_e, T_{w,\max}$] should be the inputs, and the model should be trained to match corresponding data values of mass flow rate [$\dot{m}$]. Here, use a sequential network and make appropriate choices for the number of inputs, the number of hidden layers, and the number of neurons in each layer (including the output layer). Base your choices on your experience in constructing previous models, and make the network complex enough to accurately fit the data, but avoid making it so complex that convergence takes an extreme number of iterations and/or the model over-fits the data. Consider use of dropout layers. Be sure to clearly document all your network design choices in your final report.

(d) Train the neural network model constructed in part (c) using the training data. Try to get the mean absolute error below 0.025 if possible. You can adjust the initialization and/or the learning parameter a bit to try to improve convergence.

(e) Compare the trained model predictions to the training data set, report the mean absolute error for the fit, and create a log-log scatter plot of predicted mass flow rate output vs. the data value mass flow rate, for each set of data point operating conditions.

(f) Repeat the steps of part (e), comparing the model predictions, this time to the normalized validation data. Report the mean absolute error and include the log-log plots specified in (e) in the summary report.

Note that this model can be used as part of a model-based control scheme. For a given system design, D_i is fixed, and if the desired exit quality and maximum acceptable wall temperature are specified, as the solar heat input varies, the model will predict a mass flow rate that will meet the specified x_e and $T_{w,\max}$ targets. For D_i = 0.010 m, x_e = 0.70, and $T_{w,\max}$ = 300 °C, use your trained model to predict the variation of $\dot{m}$ for 500 < q''_o < 800 kW/m^2. Show your results on a plot of $\dot{m}$ vs. q''_o for these operating conditions.

Part 2.
Background – Numerical Modeling of Natural convection Heat Transfer

A vertical heated surface in still air produces a natural convection boundary layer flow like that depicted in Fig. 7.

Invoking the usual boundary layer approximations for 2D flow, the Boussinesq approximations, neglecting pressure work and viscous dissipation, the governing equations, and the initial and boundary conditions for time-varying laminar flow are:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0 \qquad (1)$$

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=g\beta(T-T_\infty)+\nu\frac{\partial^2 u}{\partial y^2} \qquad (2)$$

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \alpha\frac{\partial^2 T}{\partial y^2} \qquad (3)$$

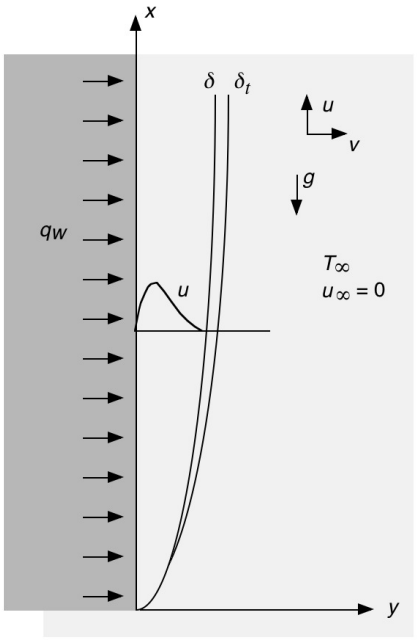


Figure 7.

at $t = 0$: $u = v = 0, T = T_\infty$ for all x, y (4a)

for $t > 0$: at $y = 0: u = v = 0, \partial T/\partial y = -q_w/k$, at $y = \infty: u = 0, T = T_\infty$ (4b)

The initial and boundary conditions model the sudden application of the uniform heat flux at the surface at time $t = 0$, with zero u velocity and $T = T_\infty$ everywhere at $t = 0$.

Numerical Analysis of Boundary Layer Flow

The system of equations, initial conditions and boundary conditions can be solved on the finite-difference mesh shown in the diagram below.

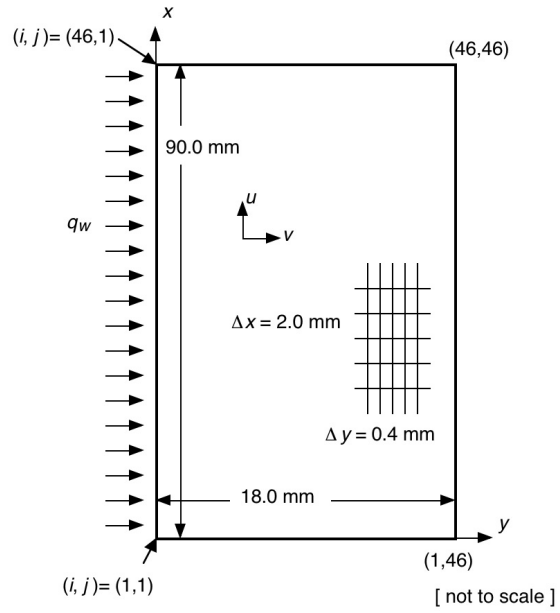


Figure 8.

First-order upwind differences are used for the first derivatives in the convection terms. Otherwise backward differences are used for first derivatives. For second order derivatives, central differences are used. The resulting finite difference equations are

$$\frac{T'_{i,j}-T_{i,j}}{\Delta t}+u_{i,j}\frac{T_{i,j}-T_{i-1,j}}{\Delta x}+v_{i,j}\frac{T_{i,j+1}-T_{i,j}}{\Delta y}=\alpha\frac{T_{i,j+1}-2T_{i,j}+T_{i,j-1}}{(\Delta y)^2} \tag{5}$$

$$\frac{u'_{i,j}-u_{i,j}}{\Delta t}+u_{i,j}\frac{u_{i,j}-u_{i-1,j}}{\Delta x}+v_{i,j}\frac{u_{i,j+1}-u_{i,j}}{\Delta y}=g\beta(T'_{i,j}-T_\infty)+\nu\frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{(\Delta y)^2} \tag{6}$$

$$\frac{u'_{i,j}-u'_{i-1,j}}{\Delta x}+\frac{v'_{i,j}-v'_{i,j-1}}{\Delta y}=0 \tag{7}$$

The above equations can be rearranged to solve for u, v and T at the next time step

$$T'_{i,j}=T_{i,j}+\left[\alpha\frac{T_{i,j+1}-2T_{i,j}+T_{i,j-1}}{(\Delta y)^2}-u_{i,j}\frac{T_{i,j}-T_{i-1,j}}{\Delta x}-v_{i,j}\frac{T_{i,j+1}-T_{i,j}}{\Delta y}\right]\Delta t \tag{8}$$

$$u'_{i,j}=u_{i,j}+\left[g\beta(T'_{i,j}-T_\infty)+\nu\frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{(\Delta y)^2}-u_{i,j}\frac{u_{i,j}-u_{i-1,j}}{\Delta x}-v_{i,j}\frac{u_{i,j+1}-u_{i,j}}{\Delta y}\right]\Delta t$$

$$v'_{i,j}=v'_{i,j-1}-\left[\frac{u'_{i,j}-u'_{i-1,j}}{\Delta x}\right]\Delta y$$

The heat flux boundary condition at $y = 0$ is represented in finite difference form as

$$\frac{-q_w}{k}=\frac{T_{i,2}-T_{i,1}}{\Delta y} \tag{9}$$

Algorithm

The explicit Forward-Time-Central Space (FTCS) algorithm can be used to advance the temperature and velocity fields in time. To implement it, a computer program must be created that executes the following operations:

[1] Create arrays to store the old (unprimed) and new (primed) field variables at each node

$$u_{i,j}, v_{i,j}, T_{i,j}, u'_{i,j}, v'_{i,j}, T'_{i,j}$$

where $1 \le i \le 46$ and $1 \le j \le 46$.

[2] Initialize old and new fields everywhere to the values

$$u_{i,j} = 0, \quad v_{i,j} = 0, \quad T_{i,j} = T_\infty$$

$$u'_{i,j} = 0, \quad v'_{i,j} = 0, \quad T'_{i,j} = T_\infty$$

[3] Compute the fields at the next time step at all non-boundary points

$$T'_{i,j} = T_{i,j} + \left[\alpha \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta y)^2} - u_{i,j} \frac{T_{i,j} - T_{i-1,j}}{\Delta x} - v_{i,j} \frac{T_{i,j+1} - T_{i,j}}{\Delta y} \right] \Delta t$$

$$u'_{i,j} = u_{i,j} + \left[g\beta(T'_{i,j} - T_\infty) + \nu \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} - u_{i,j} \frac{u_{i,j} - u_{i-1,j}}{\Delta x} - v_{i,j} \frac{u_{i,j+1} - u_{i,j}}{\Delta y} \right] \Delta t$$

$$v'_{i,j} = v'_{i,j-1} - \left[\frac{u'_{i,j} - u'_{i-1,j}}{\Delta x} \right] \Delta y$$

[4] The temperatures along the wall at $y = 0$ at the next time step are computed as

$$T'_{i,1} = \frac{q_w}{k} \Delta y + T'_{i,2}$$

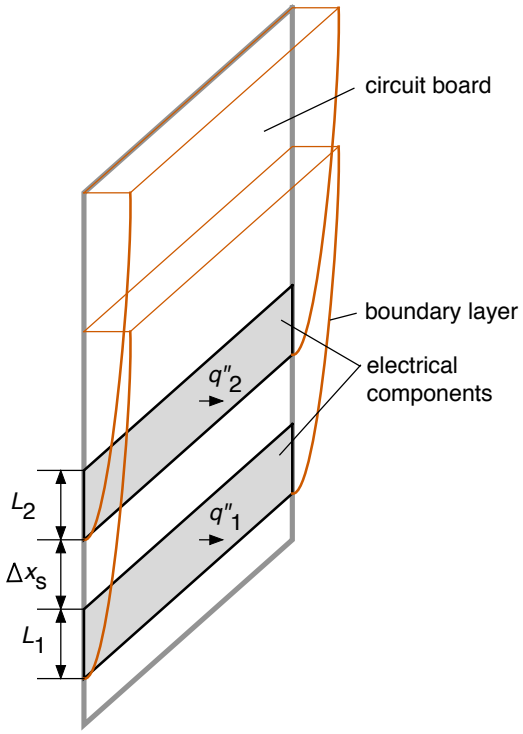
[5] If time has exceeded the specified limit, exit the algorithm. If not, store new field values in old field variables at each point

$$u_{i,j} = u'_{i,j}, \quad v_{i,j} = v'_{i,j}, \quad T_{i,j} = T'_{i,j}$$

and return to step [3].

The above steps advance the solution in time, eventually establishing the steady flow and temperature fields. Here we are interested in the wall temperature variation established for these steady state conditions.

Part 2 of this project considers natural convection cooling of two discrete heated electronic components mounted on a vertical circuit board, as depicted in Fig. 9.



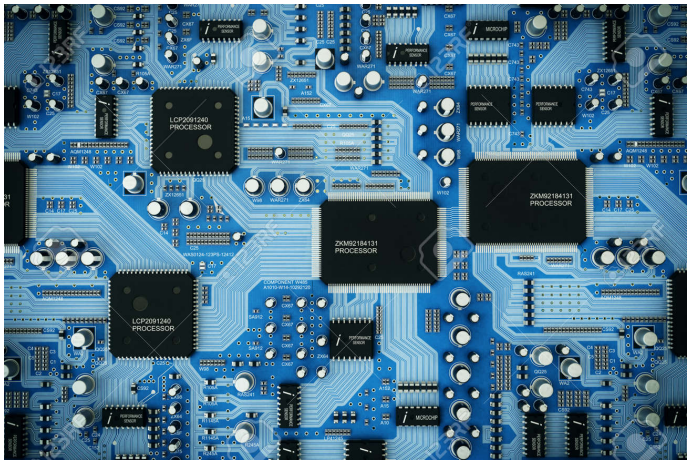


Figure 9.

A program implementing the algorithm described above was used to compute the steady flow, heat transfer and temperature field for a vertical surface with the boundary conditions set to model the two discrete heated components depicted above.

The solution depends on a number of properties and choices of the mesh parameters. For the analysis considered here, the following parameters values were chosen:

g = gravitational acceleration = 9.8 m/s^2
β = air thermal expansion coefficient = 0.0033 1/K
ν = air kinematic viscosity = 1.613 x10^{-5} m^2/s
α = air thermal diffusivity = 2.200 x10^{-5} m^2/s
k = air thermal conductivity = 0.0261 W/mK
T_∞ = ambient air temperature = 30 deg C
Δx = 2 mm
Δy = 0.4 mm
Δt = 0.0005 sec

Computations used the 46 x 46 node grid shown in Fig. 8 with Δx = 2 mm, Δy = 0.4 mm and Δt = 0.0005 seconds. The heated components in this analysis were both taken to be 8 mm wide (in the vertical direction), and the separation distance was variable.

Task 2.1

The model described above was used to predict the system heat transfer performance for a spectrum of values of the three parameters $[q_1'', q_2'', \Delta x_s]$ in the ranges $50 < q_1'' < 600$ W/m^2, $50 < q_2'' < 600$ W/m^2, and $0 < \Delta x_s < 0.010$ m. The maximum surface temperature $T_{s,\max}$ is of primary interest, because keeping the component temperatures within acceptable limits is essential for the reliable performance of the system. For example, it is generally acknowledged that the majority of today's desktop processors should not exceed temperatures of 45-50°C when idle, or 80°C when under full load.

In this task, the goal is to take performance data generated by this CFD-type model and train a neural network to predict the maximum (component) surface temperature when it is provided $[q_1'', q_2'', \Delta x_s]$ as inputs. The neural network model can then be used to explore the parametric trends in performance. The model can also be a design tool that can determine the effect of component spacing on maximum operating temperature at different power consumption levels.

(a) Use the CodeP4.1F21.ipynb provided with this project as the starting point skeleton code for the activities in this task. Data set ME249Proj4F21data2.ipynb contains the arrays for the input data $[q_1'', q_2'', \Delta x_s]$ and the output parameter $[T_{s,\max}]$ for the natural-convection cooled, two-component circuit board system depicted in Fig. 9. Determine the median value for each parameter and normalize the data by dividing each parameter value by its median value.

(b) Take the normalized ME249Proj4F21data2.ipynb data and separate it randomly into two data sets: a training set with 3/4ths of the data and a second validation set with 1/4 of the data.

(c) Substitute the normalized training data into the skeleton code and convert the code to a neural network model that can be trained using the training data set. For this model, $[q_1'', q_2'', \Delta x_s]$ should be the inputs, and the model should be trained to match data values of $[T_{s,\max}]$. Here, use a sequential network and make appropriate choices for the number of inputs, the number of hidden layers, and the number of neurons in each layer (including the output layer). Base your choices on your experience in constructing previous models, and make the network complex enough to accurately fit the data, and avoid making it so complex that convergence takes an extreme number of iterations and/or the model over-fits the data. Consider including dropout layers to help avoid overfitting. Be sure to clearly document all your network design choices in your final report.

(d) Train the neural network model constructed in part (c) using the training data. Try to get the mean absolute error below 0.025 if possible.

(e) Compare the trained model predictions to the training data set, report the mean absolute error for the fit, and create a log-log plot of the predicted $T_{s,\max}$ output versus the $T_{s,\max}$ data value for each set of data point operating conditions.

(f) Repeat the steps of part (e), comparing the model predictions, this time to the normalized validation data. Report the mean absolution error and include the log-log plots specified in (e) and (f) in the summary report.

(h) Taking the heat flux of the two components to be equal $q_1'' = q_2'' = q_{2\&3}''$, use the trained model created in this task to create predictions of the maximum surface temperature $T_{w,\max}$ for $100 < q_{2\&3}'' < 500$ W/m^2, and $0.0 < \Delta x_s < 0.015$ m, and create a surface plot of the maximum wall temperature $T_{s,\max}$ as a function of $q_{2\&3}''$ and Δx_s. Which region of $q_{2\&3}''$ and Δx_s space would you recommend if you want a maximum component temperature $T_{s,\max}$ no more than about 72 °C?

Project 4 Tasks to be divided between coworkers:

(1) Data prep and program modifications for Part 1
(2) Training process and computations for comparisons
(3) Plotting and interpretations of results for Part 1
(4) Data prep for Part 2
(3) Program modifications for neural network modeling in Part 2
(5) Plotting and analysis of the results for Part 2
(6) Write-up of the results and conclusions for the report

Deliverables:

Written final report should include:

(1) Written summary of how the work was divided between coworkers.

(2) Assessment of the results and comparisons for the two different neural network designs considered in Part 1.

(3) Plots requested in Parts 1 and 2

(4) An assessment of viability of the first neural network design considered for system control in Part 1.

(5) Your assessments and conclusions should be clearly written with quantitative information to justify them.

(6) A copy of your programs should be attached to the report as an appendix.

Grade will be based on:

(1) thoroughness of documentation of your analysis , especially the logic behind design choices for neural network
(2) accuracy and clarity of interpretation
(3) thoroughness and the documentation of the reasons for your assessments of results.

Summary report due: Tuesday December14, 2021

References

[1] Inabensa, Flichtner, Ciemat, DLR, *10 MW Solar Thermal Power Plant for Southern Spain*, Final Report, NNE5-1999-356, European Community under the 5th Framework Programme (1998-2002), November 2006, URL source (04/26/11): http://ec.europa.eu/energy/res/sectors/doc/csp/ps10_final_report.pdf

[2] Goswami, D.Y. and Kreith, F., *Energy Conversion*, Chapter 19, CRC Press, Boca Raton, FL, 2008.