Development of the CompHX Package for Heat Exchanger Analysis

Department of Nuclear Science and Engineering
Oregon State University
T. M. Moore
Tommy.moore22@gmail.com

1 INTRODUCTION

The computational heat exchanger package or CompHX was written in Python to allow for users to more easily compare the heat removal rates of different heat exchanger configurations. The package allows for finned and tubed heat exchangers to compare the performance of the geometry of the fins and tubes as well as the number of each of them. The results are then plotted for parallel flow and counter flow configurations. With a desired heat removal rate and known temperature profiles, this package can aid users in determining the minimum geometry or number of fins or tubes for their specified design.

This package was initially developed to aid in the first portion of a proposed multi-stage research project. The research project was proposed to support the development of experiments for the up-and-coming Very High Temperature Reactor (VHTR) which is being built at the Idaho National Laboratory. The main goal of this reactor is to assist the nuclear industry in the United States in developing advanced nuclear fuels that are more accident tolerant and produce more energy while maintaining safety margin.

The VHTR is going to be a sodium-cooled fast reactor that has ports located inside of the core where experiments can be placed. These experiments will be placed in the ports so that large neutron fluxes are incident on the nuclear fuel being experimented on. These experiments will be in large capsules that will have either stagnant coolant or flowing coolant to simulate the conditions in a nuclear reactor. This fuel will undergo fissions as it would in a nuclear reactor which produces heat. This heat must be removed from the experiment cartridge to the coolant of the VHTR. However, the coolants cannot mix with one another. The cartridge will be isolated hydraulically from the VHTR coolant. Because of this, a heat exchanger must be designed to remove heat from the experiments and reject it to the VHTR coolant. A cartoon mockup of the cartridge geometry and the reactor coolant is shown in

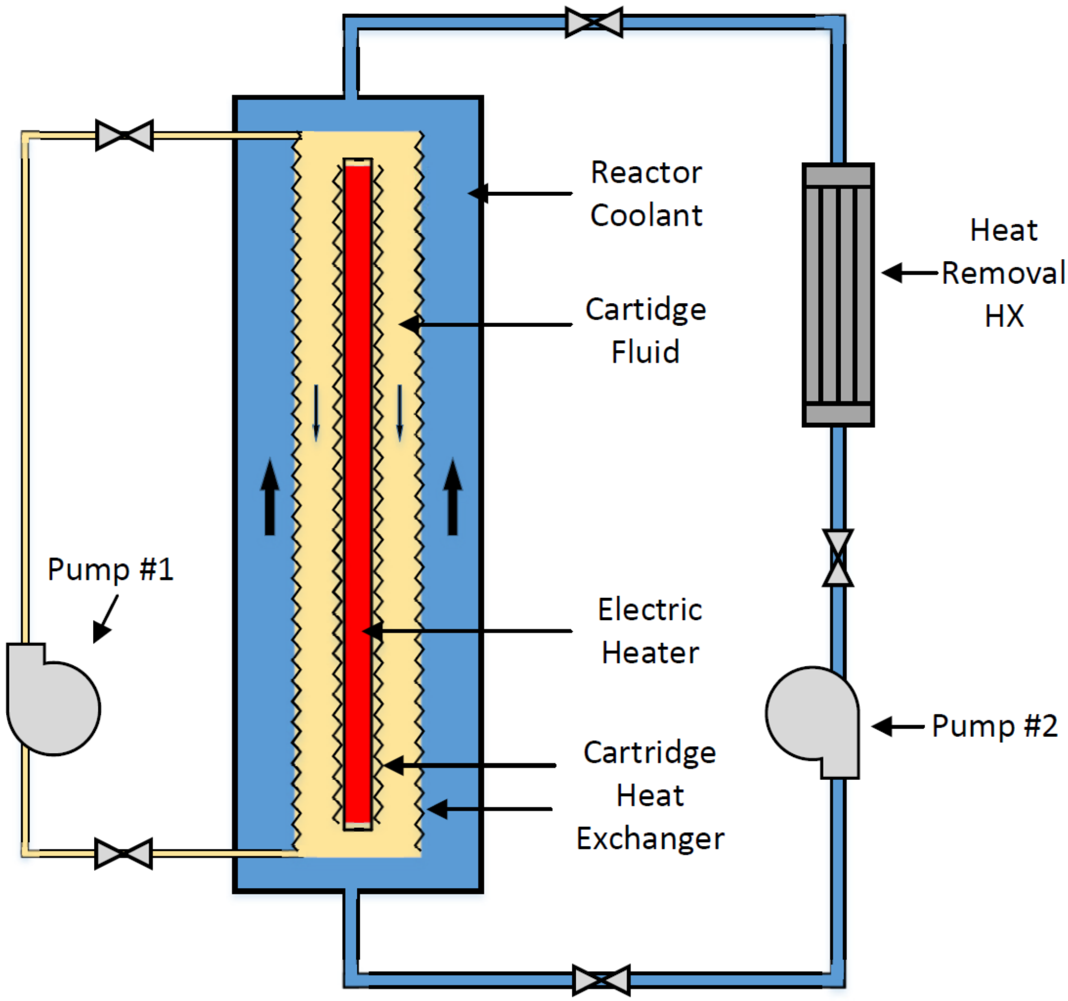

Figure 1 – VHTR experimental loop preliminary design

This package meets the needs of the first portion of the project to analytically determine the heat removal rate of different heat exchanger geometries; specially comparing finned and tubed designs. Following portions of this project would include a Computational Fluid Dynamics analysis of the chosen design and the development of an experimental loop.

Other packages have been previously developed that can be used to analyze heat exchangers. Some a more applicable to the work performed for this project while others are more specialized to other topics. It is possible that these other packages could be used for the type of analysis performed by CompHX, however for learning purposes this package was developed without knowledge of these other packages.

The most closely related package is called ht and can be found on PyPI. It implements the LMTD, ε-NTU method, and the P-NTU method. It also has a library of material properties that can be called to set the material conditions necessary for the analysis. It also has many packages that go into more advanced heat transfer and fluid flow applications that are not currently relevant to this work. This package would be a good substitute for CompHX had it not been developed.

pyHeatTransfer is also located on PyPI and initially sounded like a plausible tool for this package. However, with further investigation, not much documentation was available. What could be discerned from the github repo is that it is more suited for atmospheric science rather than thermal-fluid analysis.

TESPy is another package found that was developed for thermal engineering using Python. This package is well documented and seems like a mature package. This package however deals more with complex systems and cycles rather than simple heat exchanger analyses. This package is located on PyPI and could be useful for future work if a more complicated system of multiple components is developed.

2 METHODOLOGY

Describe equations and physical problem.

The analysis of the heat exchanger designs uses the log-mean temperature difference method as described in [1]. The log mean temperature difference is computed as follows.

$$\Delta T_{lm} = (\Delta T_1 - \Delta T_2)/\ln\left(\frac{\Delta T_1}{\Delta T_2}\right)$$

For parallel flow the following values for the temperature differences are defined.

$$\Delta T_1 = T_{hot,in} - T_{cold,in}$$
$$\Delta T_2 = T_{hot,out} - T_{cold,out}$$

And for counter flow the following values for the temperature differences are defined.

$$\Delta T_1 = T_{hot,in} - T_{cold,out}$$
$$\Delta T_2 = T_{hot,out} - T_{cold,in}$$

The heat removal rate for this method is then defined as follows.

$$q = UA\Delta T_{lm}$$

The heat removal rate is q, U is the overall heat transfer coefficient, and A is the area of the heat exchanger.

The process for computing the overall heat transfer coefficient is different for finned and tubed designs. For a finned heat exchanger, the following equation is used.

$$\frac{1}{UA} = \frac{1}{(\eta_0 hA)_C} + R_w + \frac{1}{(\eta_0 hA)_h}$$

The convective heat transfer coefficient is h, A is the area, R_w is the wall resistance between the two sides of the heat exchanger (hot and cold), and η_0 is the overall fin efficiency term for the fins. η_0 is computed as follows.

$$\eta_0 = 1 - \frac{A_f}{A}(1 - \eta_f)$$

In the above equation, A_f is the area of just the fins while A is the overall area (fins included). η_f is the efficiency of a fin geometry and is computed as follows.

$$\eta_f = \frac{\tanh(mL)}{mL}$$

$$m = \sqrt{\frac{h(2t + 2w)}{ktw}}$$

In these equations, L is the length of the fin, h is the height, t is the thickness, w is the width, and k is the thermal conductivity of the fin.

Computing the overall heat transfer coefficient for tubed designs is much simpler. The following equation yield the overall heat transfer coefficient when inverted.

$$\frac{1}{UA} = \frac{1}{h_i A_i} + \frac{\ln(D_o/D_i)}{2\pi kL} + \frac{1}{h_o A_o}$$

The subscripts i and o denote the inside and outside of the tube, D is the diameter of the tube, L is the length of the tube, and k is the thermal conductivity of the tube wall. Note that the area is for all tubes or all fins (area is computed using the number of each type).

Also implemented in the package is the ε-NTU method. This method is useful when outlet temperatures are not known. To compute the effectiveness, ε, the following equation.

$$\varepsilon = q/q_{max}$$

For this method, the maximum q value is computed as follows.

$$q_{max} = C_{min}(T_{h,i} - T_{c,i})$$

To compute the C value, it is the product of the mass flow rate and the specific heat. The minimum value is the, or C_{min}, is the minimum of the hot and cold side. The maximum value would be the larger of these two values.

The effectiveness is computed using the NTU value or the number of transfer units. Each of these values is dependent on the geometry of the flow. These are provided in chapter 11 of [1].

3 IMPLEMENTATION

The software is made up of three modules: HX_analyze, HX_boundary_cond, and HX_example. The first module, HX_analyze, houses the physics and correlations to solve heat exchanger problems. These include modules to solve for the LMTD, fin efficiency, heat removal rate for un-finned, finned, and tubed heat exchangers. The goal of these modules is to link them together to solve for the unknown/ desired design parameter of the heat exchanger. This can either be one of the temperatures or the heat removal rate. HX_analyze depends on numpy, sympy, and pyyaml. A list of the modules is presented below with a brief description. For more detail, visit the source code provided in [2].

- log_mean_temp_diff_counter: Computes the Log-Mean-Temperature Difference (LMTD) for a counter-current HX.
- log_mean_temp_diff_parallel: Computes the Log-Mean-Temperature Difference (LMTD) for a parallel HX
- q_lmtd_counter: Computes the heat rate for a counter-current Heat Exchanger (HX)
- q_lmtd_parallel: Computes the heat rate LMTD for a parallel Heat Exchanger (HX)
- c_min: Computes the minimum C value for NTU calculations
- c_max: Computes the maximum C value for NTU calculations
- q_max_ntu: Computes the maximum q value for the NTU method
- epsilon_ntu: Computes the effectiveness for different HX types for the NTU method. hx_type are parallel, counter, or shell.
- q_ntu: Computes the q value for the NTU method
- q_fin: Computes the q value for a finned HX using the LMTD method
- q_tube: Computes the q value for a tubed HX using the LMTD method
- temp_ntu_solver: Computes the temp for the NTU method. temp_type options are hot or cold. This are for the inlet to the HX
- lmtd_solver: Computes the lmtd for a specified q value.
- temp_lmtd_solver_parallel: Computes the temperature from a specified q value for a parallel HX using the LMTD method
- temp_lmtd_solver_counter: Computes the temperature from a specified q value for a counter-flow HX using the LMTD method

The boundary condition module reads in a .yaml file of the format shown in Figure 2 and returns the values of this file in dictionary format. Note that the example .yaml file shown in Figure 2 is provided in a two-column format simply for space. The input file should be a single column when used for this module. HX_boundary_cond depends on pyyaml. A list of the modules is presented below with a brief description. For more detail, visit the source code provided in [2].

- read_bc: Reads in Boundary Condition data from a .yaml file.
- set_temp_boundary_conditions: Set the Boundary Condition values to be used in other computations
- set_flow_boundary_conditions: Defines the flow boundary conditions

```yaml
case: example
hot_temp_in: 300
hot_temp_out: 150
cold_temp_in: 20
cold_temp_out: 120
h_cold: 10
area_cold: 2.75
h_hot: 150
area_hot: 2.75
num_fins:
  - 20
  - 40
  - 60
  - 80
  - 100
fin_thickness:
  - .002
  - .004
  - .006
fin_length:
  - .02
  - .04
  - .06
fin_width:
  - .75
  - 1
  - 1.25
wall_k: 200
wall_thickness: .01
num_tubes:
  - 20
  - 40
  - 60
  - 80
  - 100
tube_thickness:
  - .002
  - .004
  - .006
tube_length:
  - .4
  - .6
  - .8
tube_outer_diameter:
  - .02
  - .04
```


Figure 2 – Example .yaml file for inputting geometry and boundary conditions

Each of these modules has been tested using pytest. These modules are also provided in the archive of the code found at [2].

After installting the package the best method to use the software is to fill out a .yaml file in the same format as shown above. Depending on which parameters are of interest vary these parameters in the range desired. Note that only the number of fins/tubes, fin thickness, fin length, fin width, tube thickness, tube length, and tube outer diameter can be varied at this time. If the heat removal rate is desired then simply run python HX_example.py <filename>.yaml and a scatter plot of counter flow and parallel flow for finned and tubed HX will be provided. If the heat removal is known and a temperature is desired then the user will need to edit the example file to utilize the temperature calculating modules to produce the results for this temperature. Once the changes have been implemented, then running the file would be the same format.

4 RESULTS

Using the example simulation file and the input.yaml file results were plotted and tabulated. The plots are shown in Figures 3 - 6. The way to utilize these results would be to determine a desired heat removal rate from your heat exchanger and to determine which designs meet this criterion. The user could then query these results to determine a list of parameter combinations that remove enough heat and can choose a design with this knowledge. The results are also tabulated in Table 1.

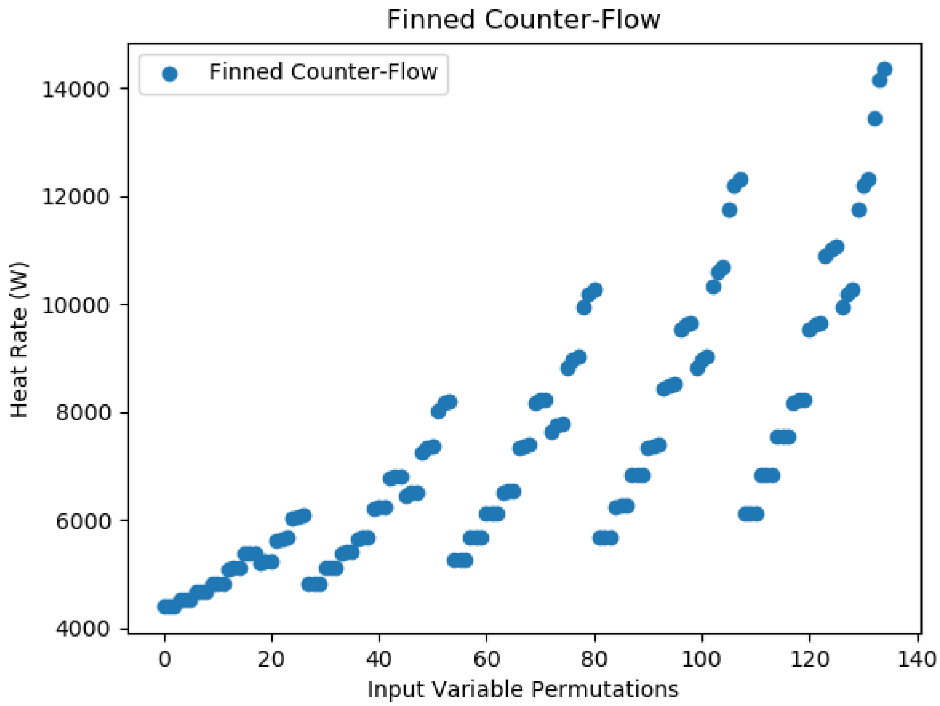

Figure 3 – Heat Removal for Finned HX in Counter Flow

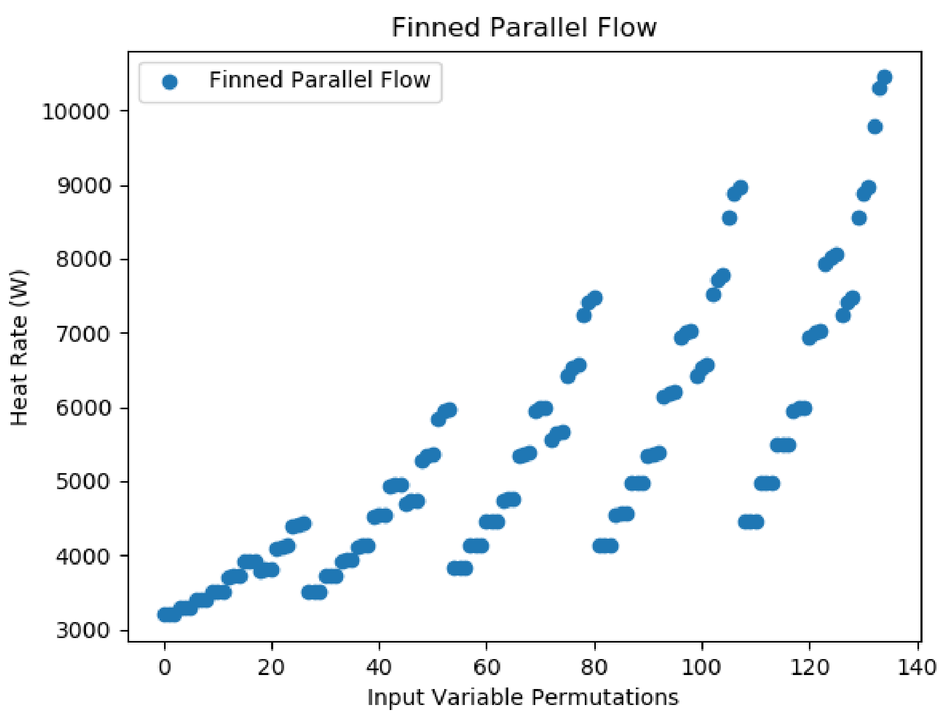

Figure 4 – Heat Removal for Finned HX in Parallel Flow

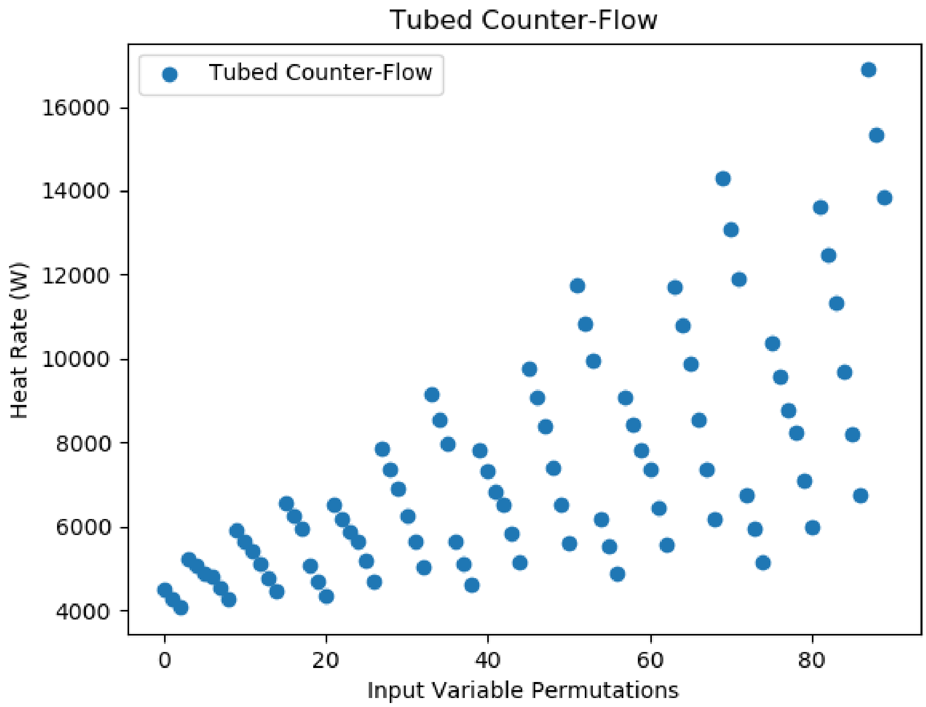

Figure 5 – Heat Removal for Tubed HX in Counter Flow

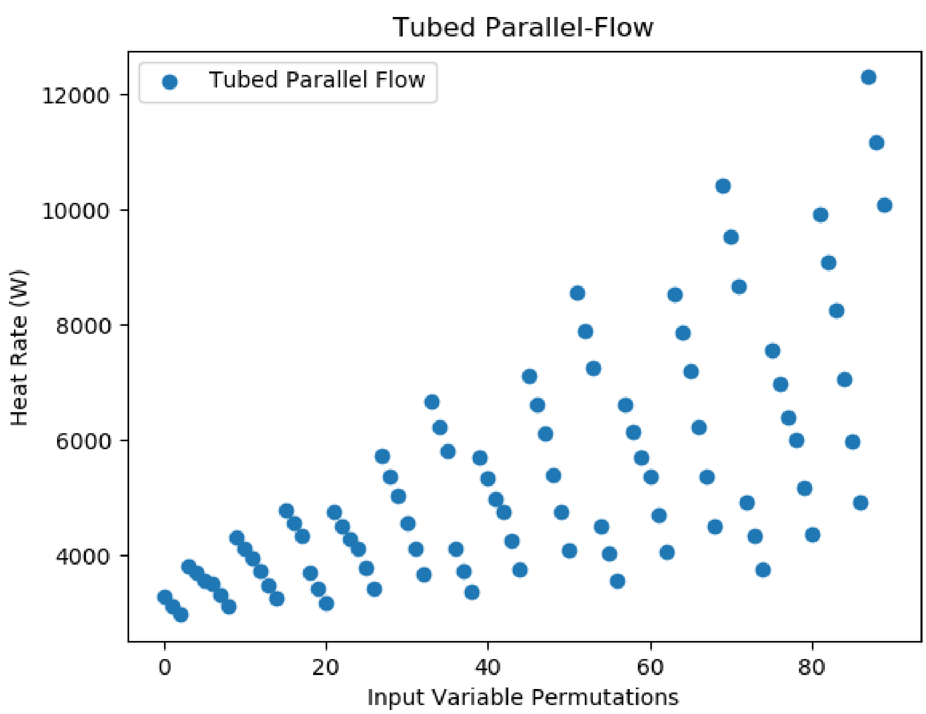

Figure 6 – Heat Removal for Tubed HX in Parallel Flow

Table 1-Maximum Heat Removal Configurations for Finned and Tubed Heat Exchangers

	Max Heat Removed (W)	Number	Length (m)	Width/ Diameter (m)	Thickness (m)
Finned Counter	14345.438	100	0.06	1.25	0.006
Finned Parallel	10450.267	100	0.06	1.25	0.006
Tubed Counter	16881.115	100	0.8	0.04	0.004
Tubed Parallel	12297.439	100	0.8	0.04	0.004

5 CONCLUSIONS AND FUTURE WORK

A tool has been developed that can be used to parametrically compare the design parameters of tubed and finned heat exchangers. An input file with these design parameters as well as material properties and boundary conditions is imported to produce results that show the heat removal for all possible design combinations of the input variables. This tool is useful for the specified comparisons, but could be more broadly applicable with more development. The code could be modified to allow for more of the variables to be analyzed (material properties, temperatures).

Another feature that could improve the capabilities of this tool is to implement an optimization scheme. This could be implemented by specifying the desired outcome (temperature or heat removal) and determining the parameters to vary and the range they can vary within.

6 REFERENCES

1. Incropera, F. P., & DeWitt, D. P. (2002). Fundamentals of heat and mass transfer. New York: J. Wiley.
2. Moore, T. M., (2019). CompHX software package. Github url, https://github.com/tommy-moore22/CompHX/tree/v0.1.0, doi: 10.5281/zenodo.3243668