ALChemE – User Manual v0.3.0 <WIP>

github.com/PedroSeber/ALChemE

<Read the Docs website>

Table of Contents

Section 1 – Introduction to ALChemE

ALChemE is a software to assist users in calculations related to the chemical engineering field and process design. As of v0.3.0, the software can assist in the analysis of heat exchange networks (HENs) and water recycle networks (WReNs), with other relevant processes (such as distillation) to be implemented in the future.

Section 1.1 – Installation

Currently, there are two ways of installing ALChemE: downloading Python 3 and obtaining the .py files or obtaining the pre-compiled executable. The former is the easiest if the user already has or desires to install Python, and it maximizes the software customizability. It allows the software to be operated via a GUI or regular Python methods (such as Jupyter Notebooks or IDEs). The latter is better suited to those who do not have or want to install Python and are not interested in customizing the software or operating it without a GUI. Both options are completely free.

As Python is free and open-source, there are multiple ways to install it. The official Python website has a downloads page with multiple versions; installing the newest one should be sufficient for this software. A popular alternative is Anaconda, which includes Python, some commonly used packages, and a package manager. Finally, there is also Miniconda, which includes Python and the same package manager. If the user is new to Python and does not know which option to choose, we recommend Anaconda.

After installing Python, the user should obtain our package. Currently, we have not yet published our package, so please contact us to obtain the software. In the future, this will occur through pip, conda, and similar methods. We will update this user guide once our package is published.

Please contact us to obtain the pre-compiled executable. After downloading and extracting it, one should run the “frontend.exe” to open the GUI.

Section 1.2 – Running ALChemE

After installing our package, the user can run ALChemE by opening a shell program (PowerShell or cmd in Windows, Terminal in Macs and Linux), using the cd command to change the working directory to the installation directory, then running

python ALChemE_frontend.py

If you do not know how to use shell, there are many tutorials on it, one of which was written by Vanderbilt's ACCRE and can be found here. The following paragraphs contain alternative routes for people not versed in shell.

In Windows, open the folder where you installed ALChemE, then hold the Shift key and right-click anywhere in that folder (not on a file, though). A menu should open. Click on "Open PowerShell window here". Then, type the python command above in that PowerShell window. Alternatively, open PowerShell, type cd, then drag-and-drop the installation folder into the PowerShell window, then type the python command above in that same window.

In Macs, open Terminal. Then, drag-and-drop the folder where you installed ALChemE into that Terminal window. Finally, type the python command above in that same window.

If you obtained the pre-compiled executable, then simply open the folder and find "ALChemE_frontend.exe". Double-click that file to open the Home Screen.

Troubleshooting

1) Windows: "python : The term 'python' is not recognized as [...]" or Mac: <error message to be added in the future>

Ensure you have installed Python, and that it is added to the PATH. If you have already installed Python, then it probably has not been added to the PATH.

Section 1.3 – The Home Screen

The Home Screen is the first screen shown by ALChemE to the user. It lets the user access all software modules to assist in optimization. If the user is interested in HENs, please proceed to Section 2 – Heat Exchanger Networks. If the user is interested in WReNs, please proceed to Section 3 – Water Recycle Networks.

Section 2 – Heat Exchanger Networks

ALChemE allows users to analyze HENs to maximize energy recovery and minimize costs. As of v0.3.0, users can input streams and utilities (with units of measurement), plot temperature interval diagrams and composite curves, and manually add heat exchangers. The program automatically handles unit conversions, calculates properties such as MER targets or the pinch point location, and can automatically place exchangers to minimize their number. This function also returns the cost of the HENs generated. Finally, the solution depth setting allows the code to return more solutions at the expense of increasing its runtime. Note that there are no guarantees all solutions will be returned, even if a very large depth is used.

After clicking on the New Project button, the user may input the minimum ΔT and the default units for the HEN. Note that all values input will be converted automatically to these default units.

Section 2.1 – User Input

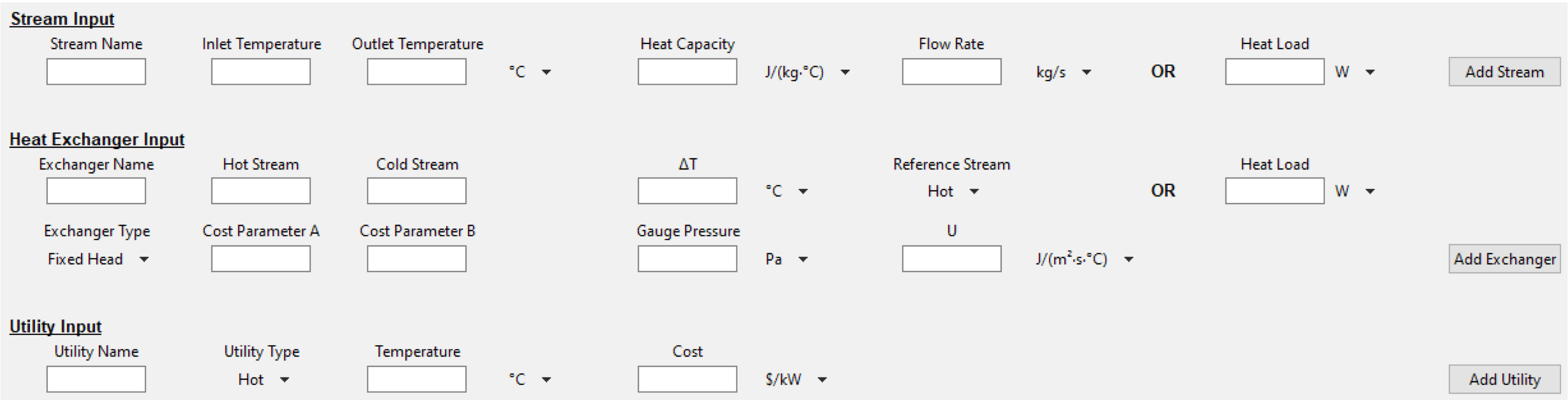

Figure 1 – The user input section

The user input section of our GUI allows users to create streams, heat exchangers, and utilities. As seen in Figure 1, it consists of text boxes for text or numerical input and drop-down menus for the units of measurement.

To add a new stream, the user must supply numerical values to the "Inlet Temperature" and "Outlet Temperature" fields. The user must also supply numerical values to one of the following combinations of text boxes: (1) "Heat Capacity" and "Flow Rate", (2) "Heat Capacity" only, which causes the value to be treated as an FC_P, or (3) "Heat Load". Unused boxes should be left blank. Note that the software automatically generates a name for streams, exchangers, or utilities if the name text box is left blank.

To add a new utility, select whether it is a hot or cold utility using the drop-down menu, then input its temperature and cost per power provided. As of v0.3.0, utility temperatures are used only to calculate exchanger costs. The software assumes that the hot utility is hotter and the cold utility is colder than any other stream. Thus, only one utility of each type should be used.

First and foremost, we highlight that we do not recommend adding heat exchangers manually. To manually add a heat exchanger, the user must supply the stream names to the "Hot Stream" and "Cold Stream". The user can also provide numerical values for either the ΔT and choose a reference stream from which FC_P

values will be obtained or the exchanger's heat load. If both fields are left blank, the software will automatically calculate the maximum heat that can be exchanged between the two streams. The second input row consists of parameters used to calculate the exchanger's cost. "Cost Parameter A" and "Cost Parameter B" are parameters that depend on the exchanger's material of construction, as defined in p. 462 of Seider *et al.*, "Product and Process Design Principles". If these fields are left blank, they default to 0. "U" is the overall heat transfer coefficient. If that field is left blank, it defaults to 100.

Section 2.2 – Object Explorer

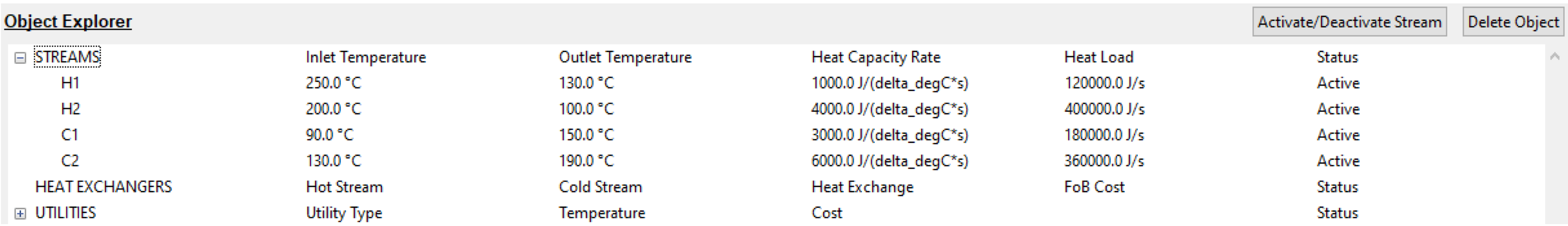

Figure 2 – The Object Explorer

The Object Explorer resides below the input section. It allows the user to browse previously added streams, heat exchangers, and utilities. As seen in Figure 2, the Object Explorer shows some relevant information on each object. Double-clicking any object prints additional information about it on the Terminal. At the top-right corner of the Object Explorer, there are two buttons: “Activate/Deactivate Stream” and “Delete Object”. To use them, the user should click on an object then click on the corresponding button. Note that only streams can be activated/deactivated, but any object can be deleted. Inactive streams will not be plotted in the diagrams or taken into account during automatic stream matching.

Section 2.3 – Diagrams

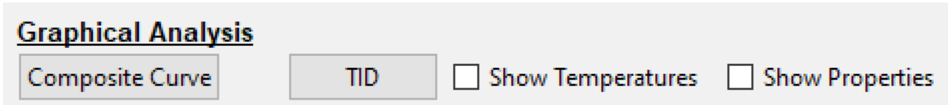

Figure 3 – The buttons used to generate plots

Based on the input provided by the user, ALChemE can automatically generate composite curves and TIDs. Composite curves also show the minimum utilities required and the maximum heat recovery possible. TIDs show the temperature of each interval, the FC_p of each stream, and the heat available for each stream in each subnetwork. The two checkboxes next to the "TID" button allow the user to show or hide that information. A sample TID with all information shown is provided in Figure 4 below.

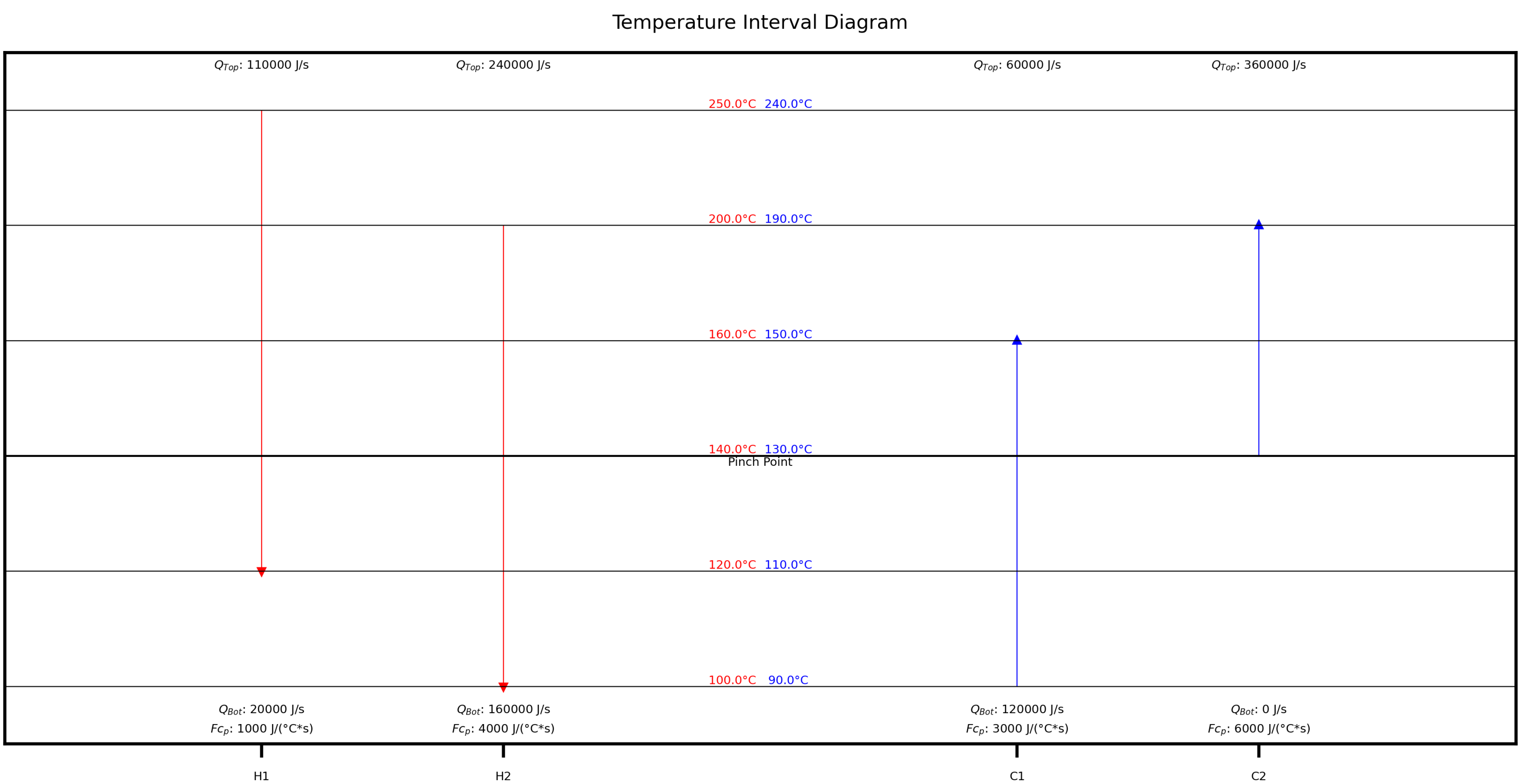

Figure 4 – A TID generated by ALChemE

Section 2.4 – Automatic HEN Optimization

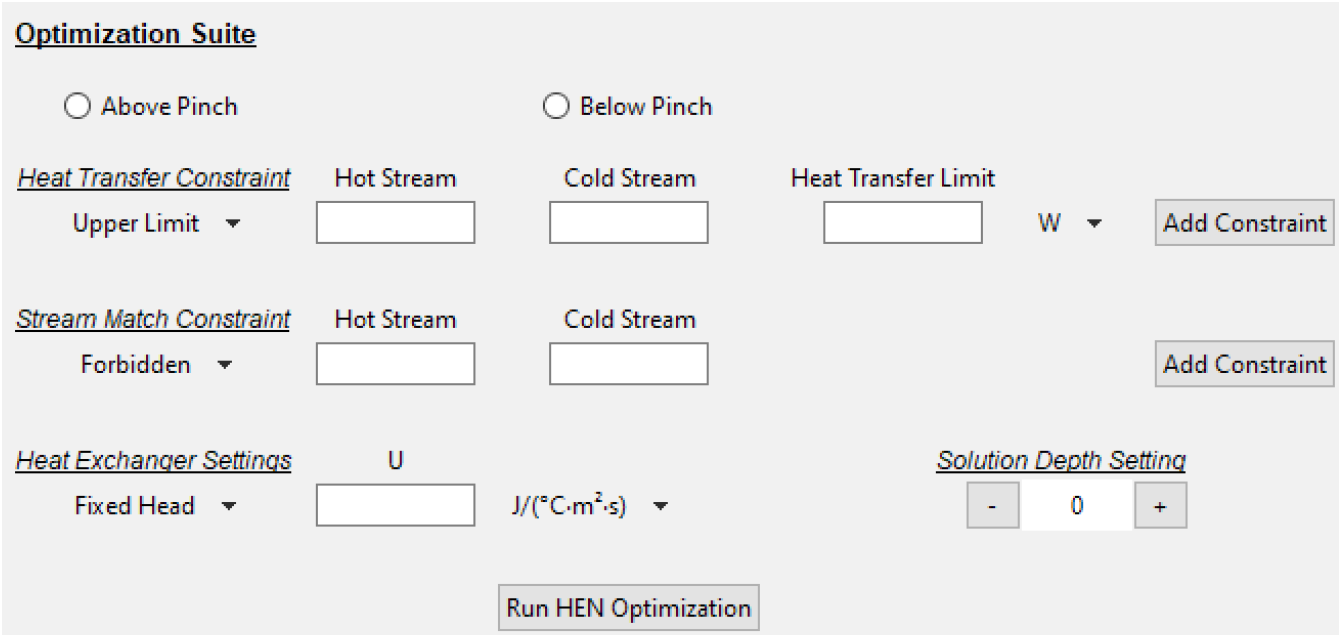

Figure 5 – The automatic HEN optimization section

Using a mixed-integer linear optimization algorithm based on the streams and utilities input by the user, ALChemE can automatically place heat exchangers to minimize their number while following the MER. In its most basic form, the software calculates the maximum heat that can be exchanged between any two streams automatically and returns the location of, the heat exchanged within, and the cost of each exchanger. Costs are calculated based on the input in the "Heat Exchanger Settings" area, where the defaults are Fixed Head exchangers with $U=100\left(\frac{J}{°C*m^2*s}\right)$.

The user can manually set upper or lower limits on each match (for example, to prevent very small heat exchangers from being used). As of v0.3.0, if the user declares an upper limit for any match, he/she must declare upper limits for all other matches. The user can also forbid or require certain matches. No HEN may exist which simultaneously obeys the MER and the forbidden / restricted matches, which will cause the program to return an error. To add any constraint, type the name of the streams or utilities, as seen in the Object Explorer.

Finally, the user can change the solution depth setting to have the program attempt to find novel solutions by forbidding or requiring certain matches. A depth of 0 returns only one solution, while a higher depth will attempt to find more. While there is no guarantee the software will find all possible solutions, this can provide alternative HENs, some of which may be better than the first HEN found.

Section 2.5 – Solved Example

We will solve example 11.1 from Seider *et al.*, "Product and Process Design Principles". It uses a $\Delta T = 10\,°C$ and contains the following streams and utilities:

Streams		
Temperature in (°C)	Temperature Out (°C)	$Fc_p \left(\frac{kJ}{°C * s}\right)$
180	80	1
130	40	2
60	100	4
30	120	1.8
Utilities		
Type	Temperature (°C)	Cost ($ per kW)
Hot	300	0.70
Cold	30	0.05

First, insert the four streams and the two utilities using the user input area. Figure 6 below shows the input for the first stream and the hot utility. When adding the cold utility, remember to change the Utility Type using the drop-down menu.

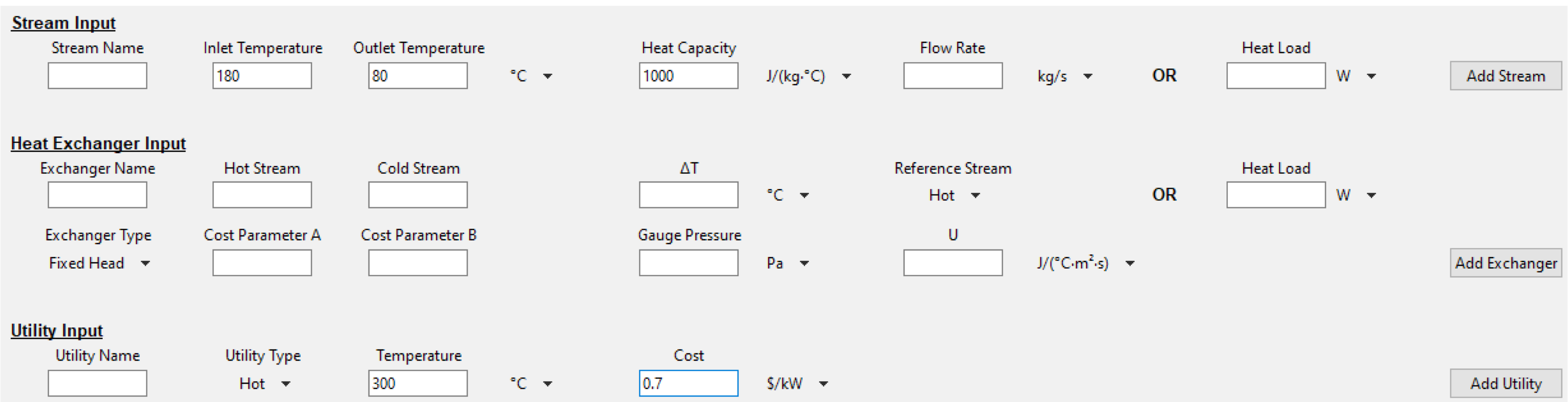

Figure 6 – Sample input of the first stream and hot utility

Use the Object Explorer to ensure all streams and utilities were properly added. You may now generate plots using the Graphical Analysis area or solve the HEN using the Optimization Suite.

To add a constraint, select its type using the appropriate drop-down menu, input the names of the streams involved in the constrained match. For example, if one wanted to require a match between the first and the last stream, one would change the "Stream Match Constraint" drop-down menu to Required, then type H1 in the Hot Stream text box and C2 in the Cold Stream text box, then click on the "Add Constraint" button. Note that adding an upper or lower heat exchanged limit also requires the user to type the limit in the third text box.

If one ran the solver without any constraints and for the subnetwork above the pinch, one would get the following output:

```
>>> Running HEN optimization method...
--------------------
Solution 1

No. Exchangers: 4
Cost: $44603.68

Solution Match Matrix (Q in J/s)
     CU1             C1            C2
HU1  0.0   40000.00002    7999.99998
H1   0.0       0.00000  100000.00000
H2   0.0  119999.99998       0.00000

Solution Match Matrix (Cost in $)
     CU1        C1        C2
HU1  0.0   9463.46   9339.45
H1   0.0      0.00  10884.91
H2   0.0  14915.86      0.00
```

Section 3 – Water Recycle Networks

<Information to be added in the future>

References

[1]: W.D. Seider et al., PRODUCT AND PROCESS DESIGN PRINCIPLES, 4th ed., p. 462