

**ME 370B**  
**Energy Systems II:**  
**Modeling and Advanced Concepts**

Chris Edwards  
Winter 2023

Project #8: Air Distillation and Cryogenic Liquefaction

**Group assignment, due as a PDF file via Canvas, Thursday, March 9, 11 PM**

This week we complete our exploration of multicomponent vapor-liquid equilibrium (VLE) by taking on the problem of air distillation. Distillation processes are ubiquitous in energy systems and can account for significant losses of exergy. Refining of Fischer-Tropsch liquids to fuels, removal of water from ethanol, separation of oxygen from air, and stripping of CO<sub>2</sub> from an amine solution are all places where distillation is not only important, but a determining factor for how efficiently we can deliver energy services in a carbon-constrained world.

Distillation is also a difficult problem to teach. While it is relatively easy to explain what happens in the equilibration process between two streams, the cumulative effect of multiple stages of equilibration—the essence of distillation—can become obscured as you track what is going in/out, up/down along a distillation column. Add to this the fact that the properties are usually not known well, and even when known are not in a form that is easy to use, then distillation can be not only obscure, but downright confusing.

Fortunately we have Matlab to keep track of the details, and we have multicomponent fundamental relations to give us exactly what we need in terms of both properties and procedures. This makes all the difference in the world. Attempting to do distillation without routines for *bubble*, *dew*, *flash* calculations would be like trying to do coal gasification by hand using the properties and equilibrium constants out of the back of Cengel and Boles. It just wouldn't work. And if you can't do it systematically and easily, you can't explore. And if you can't explore, how can you develop significantly better energy options for the future?

The Objectives

- (1) to see how the simple, fundamental phase-equilibrium functions like *bubble*, *dew*, and the various kinds of *flash*, are assembled to solve complex problems,
- (2) to experience how the way a problem is structured—forward vs. backward for example—can affect both the solvability of the problem and the computational time required for a solution,
- (3) to understand how to solve a complex, counter-flowing equilibration problem (I bet you would have no problem doing a counter-flow heat exchanger after this...),
- (4) to experience how the combined effects of thermal and material equilibration can be used to accomplish the almost-magical separations that result from distillation operations, and
- (5) to have a problem that provides an opportunity to wrangle with Newton-Raphson where it is both manageable and essential to solving the energy problem of interest.

## The Assignment

In each of Parts 1-3 there are two logic paths. One involves modeling of a reboiler, trays with conditions specified from below, and assembly of these components into a stripper column. The other involves modeling of a condenser, trays with conditions specified from above, and assembly of an enricher column. Your group should do one of each. By this I mean that two members of the group should follow the stripper path, and two should follow the enricher path. As you will see, they are logically equivalent, and at the end you will have the two halves you need to assemble a complete distillation column for air.

- (1) 20 Points Write a script that gives the output state, inlet temperature, and feed-specific heat transfer required (kJ/kg-feed) for either a reboiler or a condenser with a specified inlet composition and outlet quality. (This is what I would call the forward problem for a reboiler or condenser). Since these devices operate at the ends of distillation columns, their inlet streams are always saturated (liquid for the reboiler, vapor for the condenser) so specification of composition and either temperature or pressure is sufficient to define the inlet state. Assume that the pressure is specified (1 bar), as it would be for use in a distillation column. Test your script by supplying it with the product composition (either saturated liquid or saturated vapor) from a flash of ternary air with 50% mass quality (see *Flash\_zqP*).

Now make a function that solves the reboiler or condenser problem backwards. By this I mean find the inlet stream (temperature, composition, density) required to generate a specified outlet stream (composition, pressure) of any specified outlet quality. If your device is a reboiler, use the saturated liquid output as the specified stream. If your device is a condenser, use the saturated vapor output. In addition to the required inlet stream, have your function pass back the complementary outlet phase (either vapor or liquid) for the specified outlet stream. Note that in doing this, your function will tell you both what your device must be given to achieve a certain outlet product and what it will return for further processing (the reflux). Having the ability to solve the reboiler/condenser problem in this way will be critical to assembling a distillation column in the third part of this project.

Verify that your function is working correctly by passing it the results from the forward script you wrote previously. In addition to the output from a ternary air flash, verify that your function works correctly with a flash from binary (engineering) air, and with other arbitrary binary or ternary (saturated) streams. Just for interest sake, how long does it take the backwards function to run relative to the forward script? (No response required.)

For a reboiler, plot the input feed composition required to get an outlet composition of 95/2.5/2.5 (mole %) oxygen/nitrogen/argon as a function of outlet quality. For a condenser, plot the input feed composition required to get an outlet composition of 98/1/1 (mole %) nitrogen/oxygen/argon as a function of outlet quality. Also plot the inlet and outlet temperatures, and the feed-specific heat transfer required (kJ/kg-feed) as a function of outlet quality for either device. Use ternary air, an operating pressure of 1 bar, and mass quality increments of ~10% (no need for finer resolution, the run time will be too long).

- (2) 30 Points Write a script that finds the output state for an equilibrium distillation stage (a.k.a., an *ideal tray* or a *theoretical plate*) operating in the forward direction. By this I

mean that both inlet streams (saturated vapor from below, saturated liquid from above) are known extensively, and the purpose of the script is to find the (saturated) vapor and liquid streams (again extensive) exiting the stage. Assume that the pressure is specified (1 bar), as it would be for use in a distillation column. Test your script by supplying it with the products (both liquid and vapor) from a flash of ternary air with 50% mass quality.

Now make a function that solves the tray problem in either the upward or downward direction. By upward I mean that *all* of the information supplied to the function comes from below the tray. Specifically, both the vapor entering the tray from below and the liquid emerging from the tray (and passed to the stage below) must be specified (extensively). By downward I mean that *all* the information supplied comes from above—the liquid entering the tray and the vapor emerging from the tray must be specified. (Can you see how this will fit together with you backward reboiler or condenser from Part 1?)

Note that there is an important difference between these “backward” versions of the tray problem and the forward problem: The forward version always has a physically valid solution. Any combination of saturated streams can always be brought together and equilibrated adiabatically to find the outlet streams—and these are almost always saturated. The backward version does not share this characteristic. While the two “above” or “below” streams are physically valid in themselves, arbitrary combinations cannot be guaranteed to satisfy both species continuity and the energy equation adiabatically. Your function should have the capability to see if the solution is valid and return an error signal if it is not.

Verify that your function is working correctly by passing it the results from the forward script you wrote previously. Again for interest sake only, how long does it take the backwards function to run relative to the forward script?

For an upward tray, plot the output vapor and input liquid compositions that are required for an outlet liquid composition and inlet vapor composition that correspond to your backward reboiler plots from Part 1. (In other words, use the output results from those calculations as the input to your stage calculation here, and sweep the same range of quality.) For a downward tray, plot the output liquid and input vapor compositions that are required for an outlet vapor composition and inlet liquid composition that correspond to your backward condenser plots from Part 1. Also plot the inlet (both liquid and vapor) and outlet (saturation) temperatures and tray outlet quality as a function of reboiler/condenser quality. Use ternary air, a pressure of 1 bar, and the same mass quality intervals as before.

- (3) 50 Points Assemble either a stripper column or an enricher column. A stripper column consists of a reboiler above which are positioned trays upon which vapor and liquid equilibrate. The mixture to be separated is injected as a flash at the top of the column with the liquid proceeding downward from tray to tray while the vapor moves upward toward the outlet. The role of the reboiler is to provide an upward flow of warm vapor (the *re-flux*) to equilibrate with the downward flow of cool liquid (the *flux*) on each tray. Similarly, an enricher column consists of a condenser below which are positioned trays for vapor-liquid equilibration. The mixture to be separated is flashed into the enricher at the bottom with the vapor traveling upward through the trays and the flash liquid moving directly to the outlet. The role of the condenser in an enricher is the complement of the

reboiler in a stripper: It provides the downward flow of liquid necessary for equilibration on the trays.

In many ways calculating the performance of a column is like solving for the internal states in a counter-flow heat exchanger. In that case you know the temperature of the two fluids at the opposite ends of the exchanger, but the solution process is tricky because you cannot simply march the temperatures along (solving for Nusselt number and thereby heat transfer) as you can in the co-flow case. Instead you must adopt an implicit/iterative approach where you assign a value to the unknown temperature at one end and then walk the solution through to the other end. If the temperature at the far end matches its known value, you have assigned (guessed) the temperature correctly. If not, you must make an estimate of which direction (up or down) and by how much (step size) you should adjust the inlet temperature to drive the outlet temperature to its known value. As you have probably already sensed, this is the kind of thing that Newton-Raphson does well (if you keep a close eye on it) and so it is the usual method of choice for solving such problems.

In the case of a distillation column you have an additional twist: While the state at one end of the column (the inlet) is usually known, nothing is known about the two outlets from the column except that they come from equilibrium states. The reason is that since both composition and thermal state (temperature here, since pressure is specified) are unknown and interdependent, the actual value of each outlet state is fixed by species continuity (since there are no reactions) and conservation of energy. Note that there is something that you do know at the opposite end of the column (from the inlet that is): You know how much heat you choose to add (for a reboiler) or remove (for a condenser). Stated another way, the quality of the equilibrium stage at the other end of the column is known.

It turns out that the combination of knowing that the outlet streams result from phase equilibrium, combined with control of the quality at the opposite end of the column provides just what we need to march a solution through the column in the same iterative way as for a heat exchanger. We assume a value for the outlet composition at the opposite end of the column, decide how much heat transfer (actually quality) we want, and then march that solution back toward the (known) inlet of the column. If we have guessed correctly, then the conditions required at the tray closest to the inlet will match those that are actually provided to the column. If they do not match, the difference provides a measure of error from which to decide both the direction and magnitude of corrective action to be taken in the outlet composition at the opposite end of the column. Once convergence has been achieved, the composition and state of the outlet stream adjacent to the inlet can be computed by species continuity and conservation of energy.

Use your “backward” functions (reboiler/upward-trays or condenser/downward-trays) to make either a stripper or an enricher. In order to make this a little simpler the first time through (and to save on CPU time), do this for binary (engineering) air. This will make the outer N-R loop one dimensional and allow us to visualize the results on the  $T$ - $s$ - $x$  plot from last week. Note that by taking the backward approach to solving for each of the equilibration stages we have not only formed a natural way of constructing the implicit solution, but we also have functions that run an order of magnitude faster than those for the forward direction (by virtue of using *bubble* and *dew* instead of *flash* calculations).

Develop your solution script by considering the input to be the products from a flash of 50% quality (by mass) binary air. Assume that the flow passages in the column are large enough that the pressure may be taken to be uniform at 1 bar. Again for development, you might want to consider only one or two trays (to save time), but you should write your function so that it can accept any number since we will up the number later. (Hint: Use the *Fast\_Bubble\_cP* and *Fast\_Dew\_cP* functions provided. They can cut your run time by more than a factor of ten.)

After your code is constructed and debugged, consider the following situation:

Binary air is compressed isothermally from ambient conditions to 200 bar. It is then cooled isobarically to 130 K before being flashed (adiabatically) to 1 bar at the inlet to the column.

From here forward, explore the effect of the number of trays and the quality of the reboiler/condenser outlet by completing the problem for these cases: (S = Stripper, E = Enricher)

- (1S) for a stripper column with 2 trays and 60% quality (mass) for the reboiler,
- (2S) for the same conditions as (1S) but with 5 trays,
- (3S) for the same conditions as (1S) but with 10 trays, and
- (4S) for the same conditions as (3S) but with 70% quality for the reboiler.
- (1E) for an enricher column with 2 trays and 70% quality (mass) for the condenser,
- (2E) for the same conditions as (1E) but with 5 trays,
- (3E) for the same conditions as (1E) but with 10 trays, and
- (4E) for the same conditions as (3E) but with 60% quality for the condenser.

For all cases, show a zoomed-in view of the  $T$ - $s$ - $x$  process plot in the  $T$ - $x$  direction (see the *view* function) so that the individual stage states are visible. Connect the equilibrium stage outlet states (e.g., flash, tray closest to flash, next closest tray, ..., reboiler/condenser) by a dashed line. In addition, show the tie lines that connect to each of these states. Besides representing the equilibrium pairings, these tie lines show the compositions flowing upward (vapor) and downward (liquid) in the column between stages.

For each Case 4 (S or E) show the full range of the process on the  $T$ - $s$ - $x$  plot from a direction of your choosing. Make a separate plot showing the flow rates up and down in the column as a function of position (e.g., using tray number). Since these are extensive flow rates, normalize them by the rate of output product (bottom liquid for the stripper, top vapor for the enricher) to make the plot semi-extensive (a.k.a. product-specific).

- (4) Future Modify your stripper and enricher codes to accept ternary mixtures. This will require that you wrap a two-dimensional N-R iteration (using two of the three mole fractions) around the reboiler/condenser and trays of your column. Since your basic functions were already designed for use with ternary mixtures, no modifications to these is required. Use your stripper function to make a simple Linde-Hampson based air separation system (see Barron Section 4.12). Since this system does not include an enricher, it will not be possible to make very pure nitrogen, only oxygen. Make a histogram of exergy loss in the system, including the distillation process... (You get the idea.)

On the use of structures for passing phases: In assembling a column you will have a lot of information being passed around. Each equilibrium stage will have a temperature, multiple compositions, multiple densities, multiple flow rates, etc. Organizing this in terms of variable names can become overwhelming. This is where the use of structures can make your life better. For problems like this I usually define a “phase structure” consisting of  $T$ ,  $x$ ,  $y$ ,  $\rho_f$ ,  $\rho_g$ ,  $V$ ,  $L$  (the last two being the molar flow rates of Vapor and Liquid, respectively) in order to keep things organized. The results for the internal states of a column are then an array of such phases—one for each stage or tray—which I unpack to use as needed in the parent script. For example, if I need the saturated-liquid composition emerging from Tray 3 in the column, even I can remember that it is `tray(3).x` with corresponding density `tray(3).rf`, and temperature `tray(3).T`, etc. You might want to talk this through with your group and settle on a convention before you get too far into the project (particularly Part 3).

On keeping N-R in bounds: Although we mentioned this in the comments on N-R last week, it is worth repeating that when N-R tries to go out of bounds it is usually better to bisect the interval between the last trial value and the bound, rather than to relocate the iterate at the inner edge of the bound. What this does is to avoid having to deal with how well boundary values can be resolved in the calculation, e.g., whether a value within *toler* of the bound is actually resolvable in the calculation (or not). In cases where the bounds are not known, backtracking—cutting the step size in half until an improved solution results—can also be an effective method.

### The Write-Up

Still just the plots. Don’t forget your time estimates and more/less useful feedback.

### The Deliverables

- (1) A plot of the input mole fractions vs. quality for the reboiler/condenser backward problem. Plots of the corresponding temperatures (K) and feed-specific heat rate (kJ/kg).
- (2) Plot the unknown tray output and input mole fractions that correspond to your backward reboiler/condenser plots from Part 1. Show the known values too (for reference). Plot the inlet (both liquid and vapor) and outlet (saturation) temperatures and tray outlet quality too (separate plot).
- (3) Plots showing the view of  $T$ - $s$ - $x$  space in the  $T$ - $x$  direction for Cases 1-4. Be sure to label these with the number of trays and the quality at the outlet of the reboiler/condenser. A plot showing a 3-D view of your choosing of the  $T$ - $s$ - $x$  space for Case 4. A plot showing the product-specific flow rates (mol/mol-product) moving up and down the column for Case 4. Include the flow rates at the reboiler/condenser and flash locations.
- (4) Maybe next year... (Probably too much to ask.)