ChE curriculum

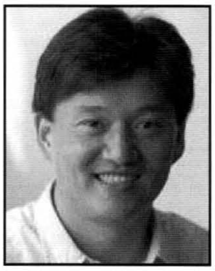
USING VISUALIZATION AND COMPUTATION
in the Analysis of Separation Processes

Yong Lak Joo and Devashish Choudhary
Cornell University • Ithaca, NY 14853

MATLAB[1] is best described as easy-to-use mathematical software that allows powerful graphical presentation and numerical analysis. At Cornell University, MATLAB has been used intensively as a teaching aid in undergraduate courses. For example, every engineering freshman is required to take a computer programming course (COMS100) that includes basic programming concepts and problem analysis using MATLAB. Students in chemical engineering take an engineering distribution course on computers and programming (ENGRD211), which deals extensively with MS Excel and MATLAB. They also develop user-friendly computer programs using MATLAB to solve homework in many chemical engineering core courses, including heat and mass transfer. This early integration of MATLAB provides an excellent background for use in the second semester of the junior year, allowing these students to be comfortable with MATLAB in the separations course. In addition, MATLAB can be a very useful teaching aid in a separations course, as its powerful graphical presentation and numerical analysis tools can be utilized both in an interactive, step-by-step, graphical display of conventional methods, and also in solving systems of equations for complex separation processes. The ability to integrate powerful computer software into the course rests on the availability of appropriate computing equipment. Our department's undergraduate computing laboratory is an excellent facility for such activities, and is equipped with 42 Windows-based PCs with a site license for MATLAB.

THE COURSE

Although typical chemical engineering curricula recognize the importance of recent trends in emerging technologies, it is always a challenge to convey them without sacrificing fundamentals.[2] ChemE332 at Cornell is a three-credit course for chemical engineering juniors covering separation methods. The emphasis of the course had formerly been placed on traditional, equilibrium-based methods that involve using manual graphical techniques, including McCabe-Thiele, Ponchon-Savarit, and Hunter-Nash.[3-7] As computers became readily available, however, the graphical approaches were supplemented with assignments to write Fortran code and/or use spreadsheets for distillation columns.[8-11] Modern tools,



__Yong Lak Joo__ has been an assistant professor of chemical and biomolecular engineering at Cornell University since 2001. He received his B.S. in chemical engineering from Seoul National University in Korea, and his M.S. and Ph.D. in chemical engineering from Stanford University. His research interests are in the area of non-Newtonian fluid mechanics and advanced materials processing, with particular emphasis on molecular modeling and complex flow simulation of polymeric liquids.

__Devashish Choudhary__ was born in New Delhi, India. He majored in chemical engineering at the Indian Institute of Technology, Bombay. In 2004, he received his Ph.D. from the School of Chemical and Biomolecular Engineering at Cornell University. During his Ph.D., he worked on order-property relationships in semiconducting materials. Currently, he works at Intel Corp.

such as the easy-to-use mathematical software MATLAB[1] and Mathematica,[12] can be used to write simple codes that allow undergraduates to calculate and display accurate graphical solutions interactively, and thus make learning graphical methods more enjoyable and effective. We introduced in-class visualization of conventional graphical methods using a simple MATLAB code. The interactive nature of MATLAB allowed "what if" analyses[11, 13] in which the effect of changing parameter values such as relative volatility, reflux ratio, feed condition, and stage efficiency are graphically displayed. By spending less time on the details of solving problems graphically or by trial and error, we can spend more time discussing the conceptual and quantitative descriptions of processes, recent trends, and design aspects. With condensed lectures on equilibrium-based processes, ChemE332 in spring 2001 was reconstructed to reinforce rate-based processes such as membrane and sorption separations. Furthermore, emerging processes in bioseparations, such as electrophoresis and issues in choosing and designing separation processes, were integrated in the course without sacrificing conventional separations. More than half of the total lectures in ChemE332 are currently spent on rate-based methods, bioseparations, and the design of separation processes.

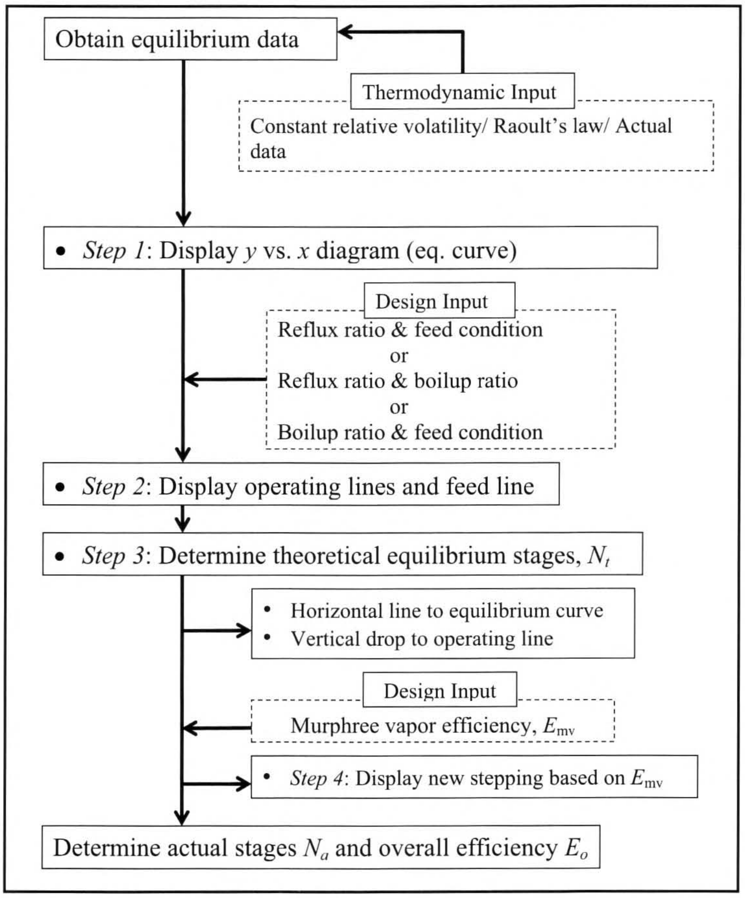

Obtain equilibrium data
Thermodynamic Input
Constant relative volatility/ Raoult's law/ Actual data
• Step 1: Display y vs. x diagram (eq. curve)
Design Input
Reflux ratio & feed condition
or
Reflux ratio & boilup ratio
or
Boilup ratio & feed condition
• Step 2: Display operating lines and feed line
• Step 3: Determine theoretical equilibrium stages, N_t
• Horizontal line to equilibrium curve
• Vertical drop to operating line
Design Input
Murphree vapor efficiency, E_{mv}
• Step 4: Display new stepping based on E_{mv}
Determine actual stages N_a and overall efficiency E_o

Figure 1. *Flowchart of Example 1: McCabe-Thiele method for binary distillation.*

Despite the advantage of helping students visualize the separation, graphical methods no longer represent the modern practice of chemical engineering.[7] Modern practice for designing and simulating separations involves commercial process simulators such as AspenPlus, ChemCad, Hysys, and Prosim.[14] To be prepared for commercial practice, students need experience simulating and designing separation processes using these methods. Unfortunately, students often treat these commercial simulators as black boxes, and tend to believe the results they obtain without further checking.[7, 14] The exact methods used in these simulators involve solving systems of nonlinear equations and large matrices. Although there is a limit for complicated systems, these exact methods are now tractable due to user-friendly routines and software for numerical analysis. To avoid the potential creation of yet another "black box" using MATLAB, students can be asked to implement specific parts of the code such as a thermodynamic model, matrix solving, and time integration scheme.

In this paper we demonstrate that using easy-to-develop mathematical solutions for visualization and numerical computation can make conventional graphical approaches more enjoyable and effective, providing students better understanding of more complex problems. Visualization and interactive display of graphical methods in distillation, solution procedures for complex processes such as multicomponent distillation, and thermal swing adsorption can promote understanding of how these separation processes work. Although we present the examples in distillation and adsorption, this approach can also be extended to many other separation processes such as absorption, stripping, and extraction. We present four examples used in the separations course. In the first two examples, the step-by-step, interactive display of conventional graphical methods for binary distillation were facilitated by MATLAB, while systems of nonlinear equations were rigorously solved using MATLAB in the last two examples on multicomponent distillation and adsorption.

Example 1

Visualization of McCabe-Thiele Method and Stage Efficiency in Binary Distillation

We used MATLAB to visualize the McCabe-Thiele graphical equilibrium-stage method and estimation of stage efficiency in a distillation process for a binary mixture of A and B. As described in Table 1, the code consists of (i) constructing and displaying the equilibrium curve, (ii) drawing operating lines and feed line, (iii) displaying the equilibrium stages, and (iv) illustrating stage and overall efficiency. We use the commands "plot" and "movie" in MATLAB[1] to visualize and animate the diagrams (see Table 1). The code was used for interactive display of the method in lectures and homework assignments.

Interactive Display in Lectures

Before the McCabe-Thiele graphical method was demonstrated by step-by-step display, a lecture was given on the concept and a handout on the detailed description of the options and functions of the MATLAB code for the method was distributed. In-class visualization of the graphical method and stage efficiency consists of four steps, and the overall flowchart of the example is illustrated in Figure 1.

Step 1. We show how the equilibrium curves can be constructed. Three ways of determining the equilibrium relationship between liquid and vapor phases are implemented in the code: using (i) a constant volatility for mixtures with a similar heat of vaporization, (ii) a simple thermodynamic model such as Raoult's law[4] in which the Antoine equation is used to provide the vapor pressure information, and (iii) actual data. For the Antoine equation, the function "fzero" in MATLAB[1] is used to find a temperature at which the sum of partial pressures of two components equals the total pressure (*i.e.*, $P_A^{sat} + P_B^{sat} = P_{total}$) for a liquid composition x_A and x_B (see Table 1).

Step 2. We show how to draw operating lines. Once any two of three parameters (*e.g.*, the reflux ratio, R; boilup ratio, V_B; and feed condition, q) are specified, the operating lines and the feed line are uniquely determined. We also explain the relation between the slope of the q-line and the state of the feed (subcooled, saturated liquid, partially vaporized, saturated vapor, and superheated).

Step 3. We demonstrate how to determine theoretical stages. Once the equilibrium curve, operating lines, and feed line are drawn, the equilibrium composition at each stage is determined by the McCabe-Thiele method. Starting from the distillate x_D (or bottoms product x_B), drawing a horizontal line from (x_D, x_D) on the y = x line to the equilibrium curve, followed by dropping a vertical line to the operating line, is repeated until x reaches x_B. When actual data is used for the equilibrium curve, the MATLAB interpolation function called "interp1" is used to find the intersection points along the equilibrium curve (see Table 1).[1] The transfer in the operating line from the rectifying section to stripping section is typically made when the liquid composition, x, passes the intersection of the two operating lines and feed line. The interactive nature of MATLAB allows "what if" analyses[9, 11] in which parameter values such as relative volatility, reflux ratio, and feed condition may be changed, and their effects on the distillation column are graphically displayed during the presentation.

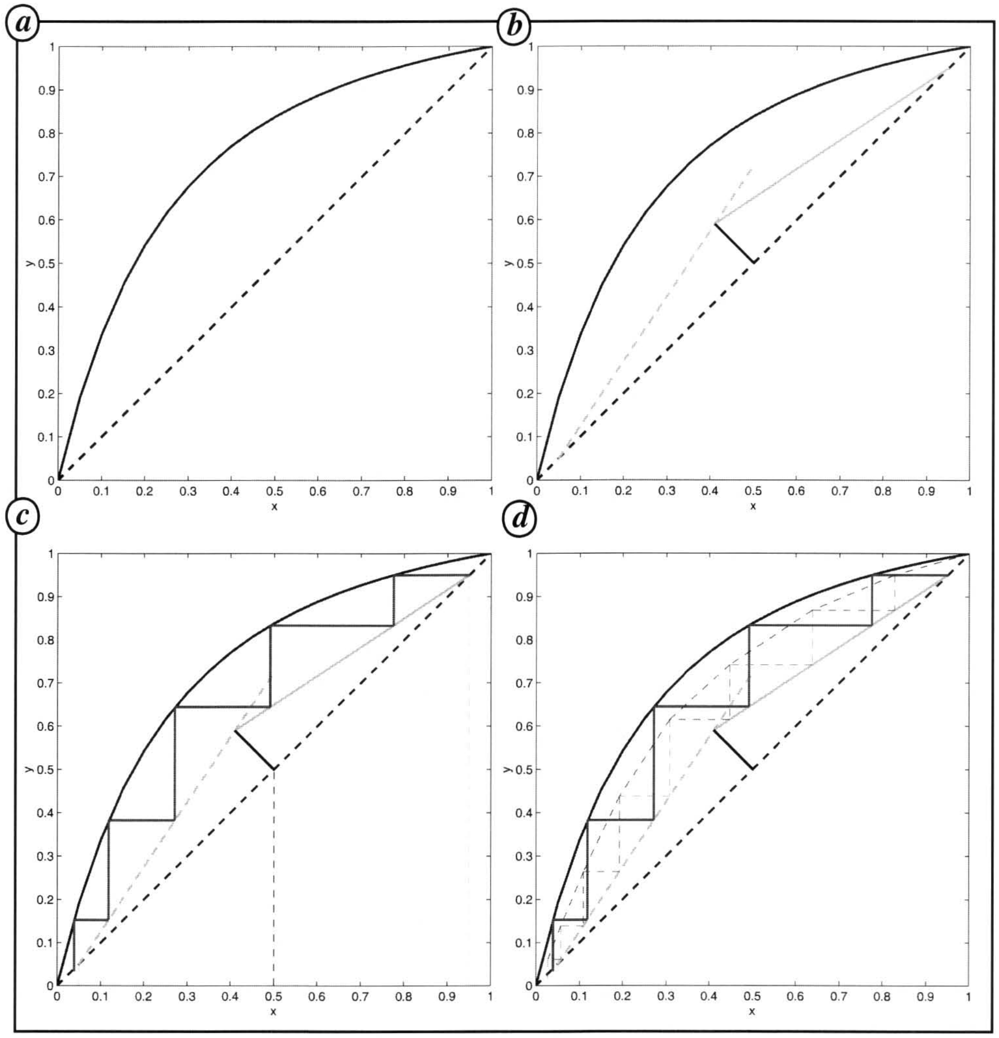
Step 4. The actual stages, based on the Murphree vapor efficiency, E_{MV}, for each stage, are displayed on top of theoretical stages to demonstrate the effect of stage efficiency on the actual number of stages. In the current example, we note that a single Murphree vapor efficiency, E_{MV}, is used throughout the entire distillation column for simplicity and symmetry in the feed stage. The overall efficiency, E_o, is then determined by the ratio of the number of the theoretical equilibrium stages to that of the actual stages, *i.e.*, $E_o = N_t/N_a$. Some snapshots of the McCabe-Thiele method and stage efficiency for distillation of acetone and toluene that are displayed in class are shown in Figure 2 (page 318).

Homework Assignments

After the graphical method by MATLAB code was introduced, a couple of problems associated with using and modifying the MATLAB code were given as homework. For example, students were asked to determine various feed conditions such as subcooled, partially vaporized, and superheated using the thermodynamic properties of benzene and toluene, and then determine the number of equilibrium stages and boilup ratio at a given feed composition and reflux ratio (see Table 2, page 318). The effect of feed conditions on column performance is demonstrated by entering different q values

TABLE 1
Portion of a MATLAB Code for Example 1

```
while x >= x_B                  % loop for stepping
  ynew=y
  if iflag==0                                    % using constant alpha for eq. relation
    xnew=ynew/(a-ynew*(a-1));
  elseif iflag==1                                % using Antoine Eq. (2) for eq. relation
    t=fzero('antoine2',tmid,optimset('disp','iter'),ynew,a1,b1,c1,a2,b2,c2,Ptotal);
    xnew=ynew*Ptotal/pvapor(a1,b1,c1,t);
  else                                           % using actual data for eq. relation
    xnew=interp1(ydata,xdata,ynew);
  end
  plot([x,xnew],[y,ynew],'r','LineWidth',2) % a. Draw a horizontal line to the eq. curve
  hold on
  Frames(:,i)=getframe;
  pause
  i=i+1;
  x=xnew
  if x >= x_c
  %if x >= z
    y=LoverV_D*x+x_D/(R+1)                       % using the op. line for rectifying section
  else
    y=LoverV_B*x-x_B/V_B                         % using the op. line for stripping section
  end
  plot([xnew,x],[ynew,y],'r','LineWidth',2) % b. Draw a vertical  line to the op. line
  hold on
  Frames(:,i)=getframe;
  pause
  if x >= x_B                                    % calculating # of stages
    nstage=nstage+1
  else
    nstage=nstage+x/x_B
  end
end                                              % c. Repeat a and b until x reaches x_B
```



in the MATLAB code and displaying the stage-stepping interactively. In the second problem, students were asked to modify and extend the MATLAB code to determine the actual number of stages based on the stage efficiency. This was demonstrated and displayed in the lecture, but this time the students were asked to reconstruct what they had seen in class and use it to solve a homework problem. About 85% of the students were able to modify the code correctly to determine the actual number of stages.

***Figure 2.** Snapshots of graphical output in Example 1: McCabe-Thiele method for binary distillation of acetone and toluene: **a)** equilibrium curve from Raoult's law; **b)** operating lines and feed line for $z_A = 0.5$, $x_D = 0.95$, $x_B = 0.05$, $q = 0.5$, $R = 2$; **c)** theoretical equilibrium stages; and **d)** actual stages (shown in dashed line) with $E_{mv} = 0.7$ for the entire distillation column.*

TABLE 2
An Example of MATLAB Homework Problem To Link the Effect of Feed Conditions to the Number of Theoretical Stages and Boilup Ratio

4. A mixture of 50 mol% benzene and toluene is to be separated by distillation at atmospheric pressure into products of 95% purity using a reflux ratio L/D=3.0 in the rectifying section. The feed has a boiling point of 92 °C and a dew point of 98 °C at a pressure of 1 atm. Determine the q value if (i) the feed is vapor at 150 °C; (ii) the feed is liquid and at 20 °C; (iii) if the feed is a mixture of two-thirds vapor and one-third liquid.

Component	ΔH^{vap} (cal/g mol)	C_p(cal/g mol °C)	
		Liquid	Vapor
Benzene	7,360	33	23
Toluene	7,960	40	33

Assume a relative volatility of 2.5 and use a simple MATLAB code (feed.m) that is available at the ChemE 332 Web page to determine the number of theoretical stages and the boilup ratio in the stripping section for three different feed conditions. Submit the printouts (graphs). Each graph should have your name and the output (number of stages and boilup ratio) printed on the upper left corner. To do this, the MATLAB code has the following gtext command that writes the specified string at a location clicked with the mouse in the graphics window:

```
gtext({'number of stages:,' num2str(nstage)})
gtext({'boilup ratio:,' num2str(V_B)})
gtext({'run by,' your_name})
```

First, the code asks you your name and input conditions including the q value. After running the code, go to the graph and click a location (total three times) to print out the number of stages, boilup ratio, and your name on the graph.

Example 2

Visualization of Enthalpy Method in Binary Distillation[6]

The McCabe-Thiele method uses an energy balance only at the feed tray, whereas the Ponchon-Savarit graphical method uses a rigorous energy balance throughout the distillation column.[4, 6] Although the Ponchon-Savarit method for distillation has largely been supplemented by rigorous computer-aided methods, the concept of using a diagram for the separating agent (heat in distillation) and difference points is very important and useful in understanding similar graphical approaches in other separation processes, such as the Maloney-Schubert graphical method[4] in extraction that uses the analogous Janecke diagram for the separating agent (the solvent).

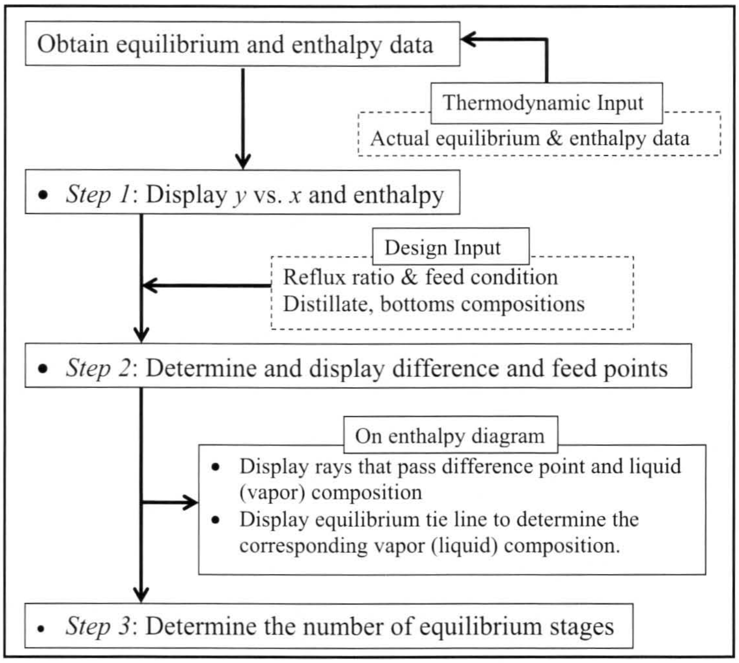

Obtain equilibrium and enthalpy data
Thermodynamic Input
Actual equilibrium & enthalpy data
• Step 1: Display y vs. x and enthalpy
Design Input
Reflux ratio & feed condition
Distillate, bottoms compositions
• Step 2: Determine and display difference and feed points
On enthalpy diagram
• Display rays that pass difference point and liquid (vapor) composition
• Display equilibrium tie line to determine the corresponding vapor (liquid) composition.
• Step 3: Determine the number of equilibrium stages

***Figure 3.** Flowchart of Example 2: Ponchon-Savarit method for binary distillation.*

We used recitation sessions as well as lectures to introduce and demonstrate the Ponchon-Savarit graphical method. A handout on the method using the MATLAB code was distributed first, and the graphical method was demonstrated using step-by-step display. The visualization of the Ponchon-Savarit method consists of determining difference points and displaying rays and equilibrium tie lines on the enthalpy diagram. The flowchart of the Ponchon-Savarit method for binary distillation is shown in Figure 3. We again used the commands "plot" and "movie" in MATLAB to visualize and graphically display the diagrams,[1] and some snapshots of the method for distillation of acetone and water mixtures are shown in Figure 4, right, as well as in Figure 5 (next page). The operating lines obtained under the assumption of constant molal overflow are shown by the dashed lines in the y vs. x diagram for comparison in the figure. The students were asked to run the same code as the lecture to solve similar homework problems by varying design inputs such as feed conditions. In the future, we will ask students to modify the MATLAB code for the Ponchon-Savarit method for

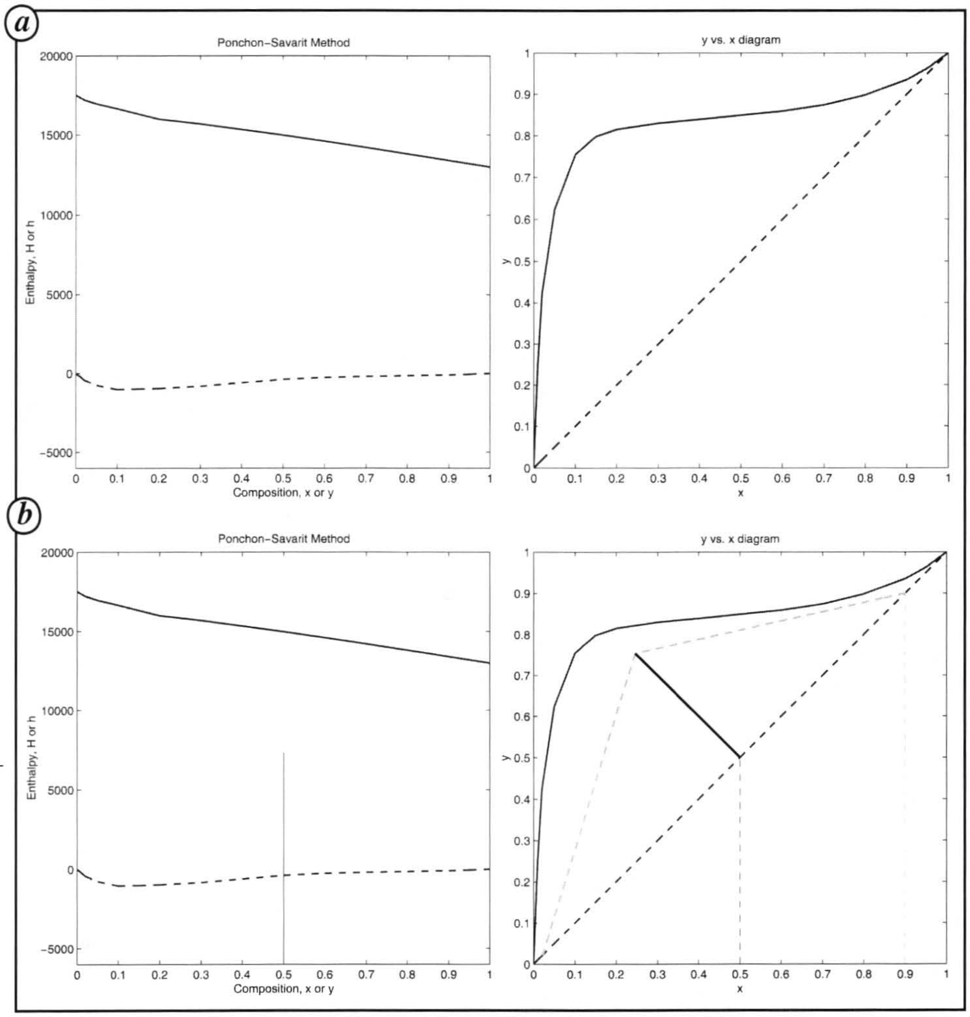


***Figure 4**. Graphical output for Example 2: **a)** enthalpy-composition diagram from enthalpy data; and **b)** difference points (open circles) and feed line for $z_A = 0.5$, $x_D = 0.90$, $x_B = 0.0216$, $q = 0.5$, and $R = 0.288$. The y-x composition diagram is also shown at the*

distillation such that the extraction process can be solved, analyzed, and displayed interactively.

Example 3
Direct Solving Exact Methods for Multicomponent Distillation[4]

Despite its practical importance, multicomponent distillation has not been thoroughly discussed in first courses on separations. This is mainly because analysis of multicomponent separations requires solving material balances, enthalpy balances, and equilibrium relations at each stage, and solution procedures can be difficult and tedious. Hence, only an approximate method commonly referred to as Fenske-Underwood-Gilliland (FUG) has been used to make preliminary designs and optimize simple distillation.[4] Alternatively, commercial simulators have been introduced to solve multicomponent separations in detail, but students often treat these commercial process simulators as black boxes.[7,14] We used MATLAB to solve the nonlinear algebraic equations for multicomponent distillation in this example. More specifically, user-friendly routines in MATLAB were used to employ the equation-tearing, bubble-point method in solving the governing equations. This numerical method consists of calculating equilibrium compositions and enthalpies, solving the modified material balance equations, and updating solutions using Newton's method (see Table 3). As indicated in the flowchart of the procedure in Figure 6 (page 322), the system of equations was solved for compositions at each stage by the matrix solver "sparse" in MATLAB.[1] The Newton's method was used to update the guess of tearing variables, temperature, and vapor rate at each stage. A function "froot.m" was created which solves nonlinear equations using a Newton's method to update the temperature and vapor rate at each stage. Once temperature, enthalpy, and compositions are obtained, the heat duties can be determined.

Homework Assignments

Using the developed MATLAB code, students were asked to solve a multicomponent distillation of hydrocarbons and compare the results with those obtained from the commercial process simulator, AspenPlus (see Table 4, page 322). Again, a handout that describes the method used in the code was distributed and explained in a recitation session before the homework was distributed. As depicted in Figure 7 (page 323), a simple thermodynamic model (Raoult's law in which the Antoine equation has been used to provide the vapor pressure information) overpredicts the volatility of "light non-key (LNK)" component (ethane) and underpredicts that of "heavy non-key (HNK)" components (pentane and hexane) in the multicomponent distillation of hydrocarbons. As a result, the compositions of the "light key (LK)" component (propane) in the distillate and the "heavy key (HK)" component (butane) in the bottoms are slightly lower than the values obtained from Aspen simulation with more accurate thermodynamics models such as Soave-Redlich-Kwong equation.

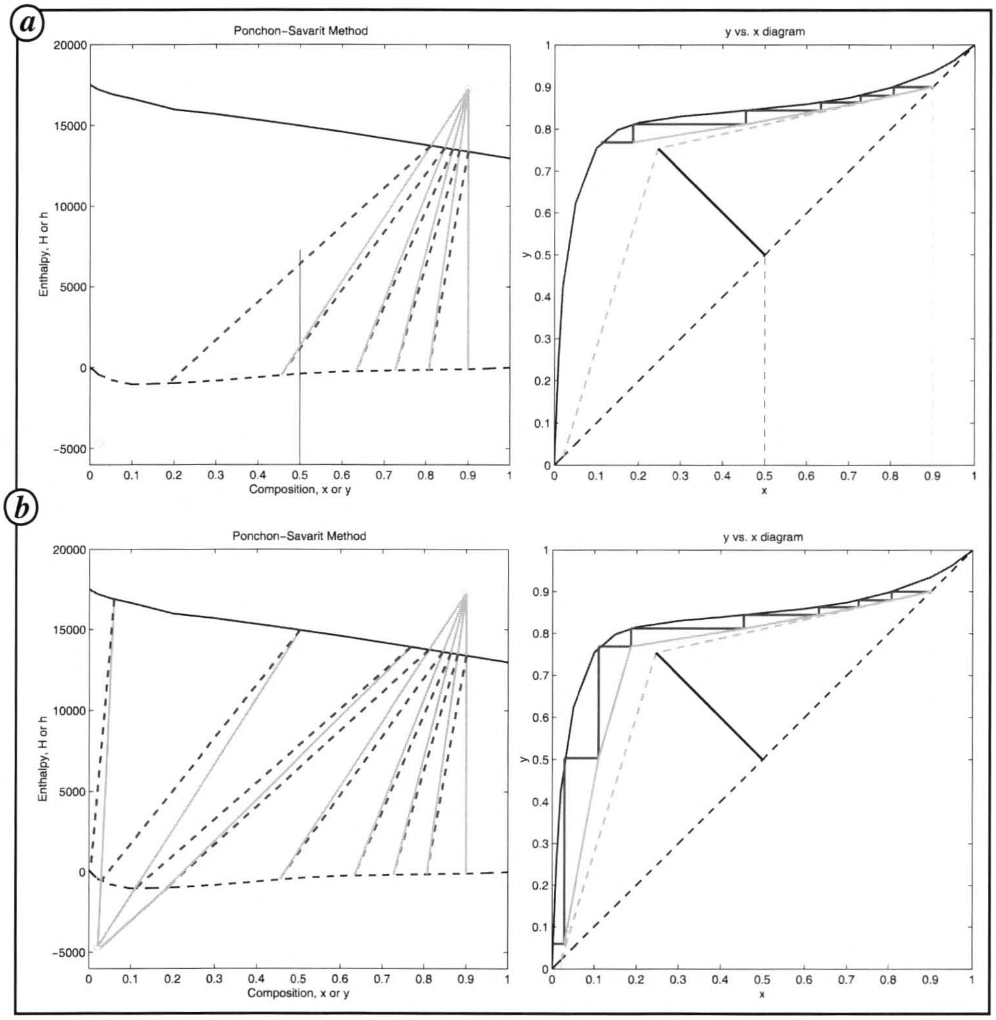


Figure 5. *Snapshots of graphical output of Example 2: Pochon-Savarit method for binary distillation of acetone and water: **a)** rays (solid) and equilibrium lines (dashed) in the rectifying section; and **b)** rays and equilibrium tie lines in the rectifying section. The operating lines obtained under the assumption of constant molal overflow are shown by the dashed lines in the y vs. x diagram at the right for comparison.*

Example 4
Direct Time Integration of Thermal Swing Adsorption[4]

Adsorption is one of the most difficult separation processes to teach since it is rate-based, which requires a mass transfer analysis, and is usually operated as a time dependent process. As a result, adsorption with very simple isotherms, such as an irreversible isotherm, has been analyzed in most separation texts.[3, 4] After we introduced the concept of adsorption isotherms and a breakthrough curve in fixed-bed adsorption, we used MATLAB to develop a numerical model for rate-based, time-dependent adsorption processes such as thermal swing adsorption. In thermal swing adsorption, one bed is adsorbing the solute at ambient temperature, while the other bed is desorbing the adsorbate at a higher temperature. A numerical solution for the regeneration (desorption) step can be obtained using a procedure discussed by Wong and Niedzwiecki[15] (see Figure 8, page 324). Again, a handout that describes the method used in the MATLAB code was distributed in the lecture. In the absence of axial dispersion and a constant fluid velocity, the partial differential equations can be solved using the five-point, biased upwind, finite difference approximation derived from Taylor's series expansion. The time integration of the sets of ordinary differential equations was carried out using a simple Euler method with a small step size. Regeneration-loading profiles in thermal swing adsorption at two different regeneration air interstitial velocities, v = 30 m/min and v = 60 m/min, are shown in Figure 9 (page 324), and the effect of air flow on the heating

TABLE 3
Portion of a MATLAB Tutorial Handout for Example 3

If we assume that phase equilibrium is achieved at each stage, the governing equations for a distillation process for n components consisting of N stages can be written as[4]

$$\mathbf{L}_{j-1}\mathbf{x}_{j-1} + \mathbf{V}_{j+1}\mathbf{y}_{j+1} + \mathbf{F}_j\mathbf{z}_j - \left(\mathbf{L}_j + \mathbf{U}_j\right)\mathbf{x}_j - \left(\mathbf{V}_j + \mathbf{W}_j\right)\mathbf{y}_j = \mathbf{0} \tag{1}$$

$$y_j = K_{i,j}\mathbf{x}_j \tag{2}$$

$$\sum_{i=1}^{n} x_{i,j} = 1; \sum_{i=1}^{n} y_{i,j} = 1 \tag{3}$$

$$\mathbf{L}_{j-1}\mathbf{h}_{j-1} + \mathbf{V}_{j+1}\mathbf{H}_{j+1} + \mathbf{F}_j\mathbf{h}_{F\,j} - (\mathbf{L}_j + \mathbf{U}_j)\mathbf{h}_j - (\mathbf{V}_j + \mathbf{W}_j)\mathbf{H}_j - \mathbf{Q}_j = \mathbf{0} \tag{4}$$

where $\mathbf{L}_j$, $\mathbf{V}_j$, $\mathbf{F}_j$, $\mathbf{U}_j$ and $\mathbf{W}_j$ are liquid, vapor, feed, liquid side stream, and vapor side stream rates at stage j, respectively. $\mathbf{h}_j$, $\mathbf{H}_j$ and $\mathbf{Q}_j$ are liquid and vapor enthalpies, and heat transfer at stage j, respectively. We utilize the equation-tearing, bubble-point method in solving the governing Eq. (1)-(4) which consists of: i) calculating equilibrium compositions and enthalpy; ii) solving the modified material balance equations; and iii) updating solutions using the Newton's method.

*i) Equilibrium Compositions and Enthalpy Calculation*s

For simplicity, the Antoine equation is used to evaluate K-values and enthalpy of each component. One of tear variables, temperature is assumed and the volatility of each component is determined by $K_i = y_i/x_i = P^{sat}_i/P_{total}$. Meanwhile, the enthalpy of each species can be determined from[4] $h_{v,i} = h^0_{v,i} + \Delta H^{vap}_i$ where the ideal gas species molar enthalpy $h^0_{v,i} = \int_{T_0}^{T} C^0_{pV,i}\,dT = \sum_{i=1}^{4} a_{k,i}(T^K - T^K_0)/k$ and $C^0_{pV,i} = a_{1,i} + a_{2,i}T + a_{3,i}T^2 + a_{4,i}T^3$ is the heat capacity at constant pressure. At low pressures, the enthalpy of vaporization is given in terms of vapor pressure by classical thermodynamics[4]

$$\Delta H^{vap}_i = RT^2\left(\frac{d\ln P^{sat}_i}{dT}\right) = RT^2\left(-\frac{\bar{B}_i}{\left(T+\bar{C}_i\right)^2}\right) \tag{5}$$

ii) Modified Material Balance Equations

By rearranging the governing equations Eq. (1)-(3) for each stage, the following systems of equations for component i at stage j are obtained:[4]

$$\begin{bmatrix} \mathbf{B}_1 & \mathbf{C}_1 & 0 & \cdots & 0 \\ \mathbf{A}_2 & \mathbf{B}_2 & \mathbf{C}_2 & \ddots & \vdots \\ 0 & \mathbf{A}_3 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \mathbf{B}_{N-1} & \mathbf{C}_{N-1} \\ 0 & \dots & 0 & \mathbf{A}_N & \mathbf{B}_N \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_{N-1} \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} \mathbf{D}_1 \\ \mathbf{D}_2 \\ \vdots \\ \mathbf{D}_{N-1} \\ \mathbf{D}_N \end{bmatrix} \tag{6}$$

where $x_j=[x_{1,j}, x_{2,j}, x_{3,j}, \cdots x_{n,j}]$ is the liquid composition vector at stage j, and the components of the tridiagonal matrix in each stage are

$$\mathbf{A}_j = \mathbf{V}_j + \sum_{m=1}^{j-1}(\mathbf{F}_m + \mathbf{W}_m + \mathbf{U}_m) - \mathbf{V}_1, \qquad 2 \le j \le N \tag{7}$$

$$\mathbf{B}_j = -\left[\mathbf{V}_{j+1} + \sum_{m=1}^{j}(\mathbf{F}_m - \mathbf{W}_m - \mathbf{U}_m) - \mathbf{V}_1 + \mathbf{U}_j + (\mathbf{V}_j + \mathbf{W}_j)K_{i,j}\right], \qquad 1 \le j \le N \tag{8}$$

$$\mathbf{C}_j = \mathbf{V}_{j+1}K_{i,j+1}, \qquad 1 \le j \le N-1 \tag{9}$$

and the right-hand-side vector at each stage $\mathbf{D}_j = -\mathbf{F}_j\mathbf{z}_j, \qquad 1 \le j \le N \quad (10)$

The system of equations Eq. (6) is solved for $x_{i,j}$ by the sparse matrix solver "sparse" in MATLAB[1] . . . [instructions continue].

TABLE 4
An Example of MATLAB Homework Problem Paired With a Problem Using Aspen Plus to Solve a Multicomponent Distillation of Hydrocarbons.

1. Multicomponent Distillation using Aspen Plus

Distillation column specifications are given as below:

- Feed (saturated liquid at 250 psia and 213 °F)

Component	Lbmol/h
Ethane	3.0
Propane	20.0
n-Butane	37.0
n-Pentane	35.0
n-Hexane	5.0

- Column pressure = 250 psia
- Partial condenser and partial reboiler
- Distillate rate = 23.0 lbmol/h
- Reflux rate = 150.0 lbmol/h
- Number of equilibrium plates (exclusive of condenser and reboiler) = 15
- Feed is sent to middle stage

E-mail the following to the TA:

1) a printout of your Aspen process with your NetID as the column name as well as a stream table showing the results using the conditions described in the exercise including stage temperatures, vapor and liquid flow rates, and reboiler and condenser duties.

2) a graph of liquid composition of each component vs. stage number

3) a graph of vapor composition of each component vs. stage number

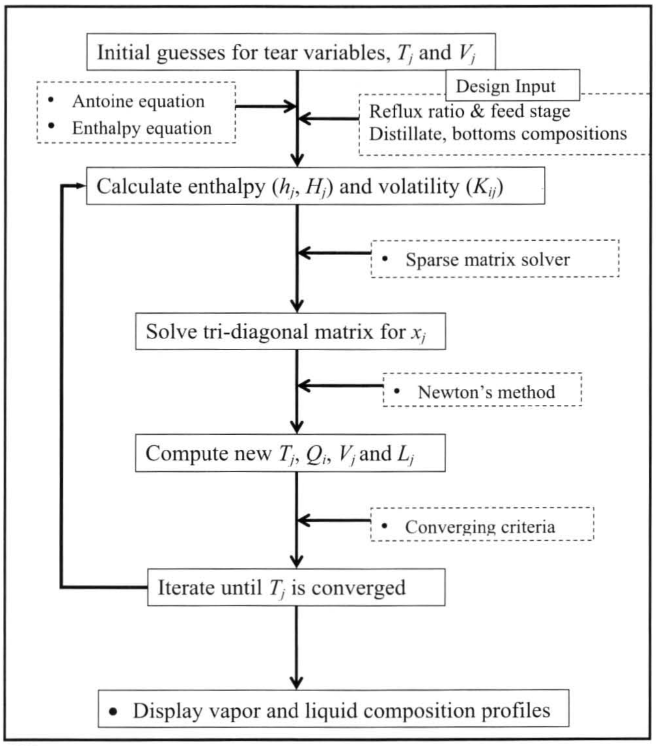
2. Multicomponent Distillation using MATLAB

Repeat Problem 1 using simple MATLAB codes (problem2.m and froot.m) available at the ChemE 332 Web site. The code utilizes the equation-tearing, bubble-point method in solving the MESH equations as described in the handout. For simplicity, the Antoine equation is used to evaluate K-values and enthalpy of each component. The file froot.m is a function routine which solves nonlinear equations using a Newton's method. See the handout for details. When the code is run, you are asked to input the conditions described in the problem. Submit the following printouts

1) a graph of liquid composition of each component vs. stage number

2) a graph of vapor composition of each component vs. stage number

Compare your results with those obtained in Problem 1.


Initial guesses for tear variables, T_j and V_j
• Antoine equation
• Enthalpy equation
Design Input
Reflux ratio & feed stage
Distillate, bottoms compositions
Calculate enthalpy (h_j, H_j) and volatility (K_{ij})
• Sparse matrix solver
Solve tri-diagonal matrix for x_j
• Newton's method
Compute new T_j, Q_i, V_j and L_j
• Converging criteria
Iterate until T_j is converged
• Display vapor and liquid composition profiles

◀ ***Figure 6.*** *Flowchart of Example 3: multicomponent distillation.*

TABLE 5
Responses of the Students

Responses to:
"How valuable were the lectures and homework assignments based on MATLAB?"

% responses	
1 = taught me little	3.0
2 = taught me some	4.2
3 = educational	16.3
4 = very educational	57.7
5 = extremely educational	17.8

Some comments from the students

"The mix of conventional method and animation helps us to understand the concept from the front."

"I like how much of it is graphic. This makes the learning more intuitive."

"I hate graphical methods, but using MATLAB is okay."

"Some MATLAB homework problems were too easy because I just punched in numbers"

"No more MATLAB, please."

and cooling cycle in thermal swing adsorption was discussed in detail. Students were asked to use the MATLAB code to determine the regeneration characteristics in thermal swing adsorption at various operating conditions, such as air flow. In the future, the students will be asked to extend the code to solve similar rate-based sorption processes such as ion exchange and chromatography.

PEDAGOGICAL ASPECTS OF STUDENT ACTIVITIES AND RESPONSES OF STUDENTS

The pedagogical aspects of student activities have evolved over the years. The incorporation of interactive display of graphical methods was done in lectures to effectively demonstrate the effect of design parameters on the distillation column. Then, students were asked to run the same code used in lecture to solve similar problems by varying design inputs such as feed conditions. We started asking the students to modify the MATLAB codes to extend its capabilities and analyze the results. A tutorial on how to develop a MATLAB code was instituted in the recitation sessions to make this transition smoother. We conducted a survey on using MATLAB in lectures and homework assignments as a part of mid-term evaluation, and results are summarized in Table 5. The wording of questions and responses in the table is taken verbatim from the survey. The survey also provided a space for written comments. As indicated in Table 5, the use of MATLAB was generally accepted as a useful aid in teaching separations. In the future, we would like to allow the students to play more active roles in solving various separation problems using MATLAB. In particular, students will be asked to modify the MATLAB codes and extend them to work out many other separation processes such as absorption, stripping, and extraction.

CONCLUSIONS

We have demonstrated that simple mathematical software, MATLAB, can be integrated into a separations course as a useful and effective teaching aid for visualization and numerical computation of many separation processes. The benefits of using MATLAB are the following:

- *Step-by-step and interactive display can make conventional graphical approaches more enjoyable to students and more effective in classroom. Visualization of the graphical methods has been further extended to the study of packed-column analysis.*
- *By spending less time on the details of solving problems graphically and by trial-and-error, we were able to spend more time discussing the conceptual and quantitative description of processes, and incorporate recent trends and design aspects in the separations course.*
- *User-friendly routines of MATLAB can be used to solve systems of nonlinear equations and perform numerical time integration, which, in turn, provides students with a better understanding of complex separation processes such as multicomponent distillation and thermal swing adsorption.*

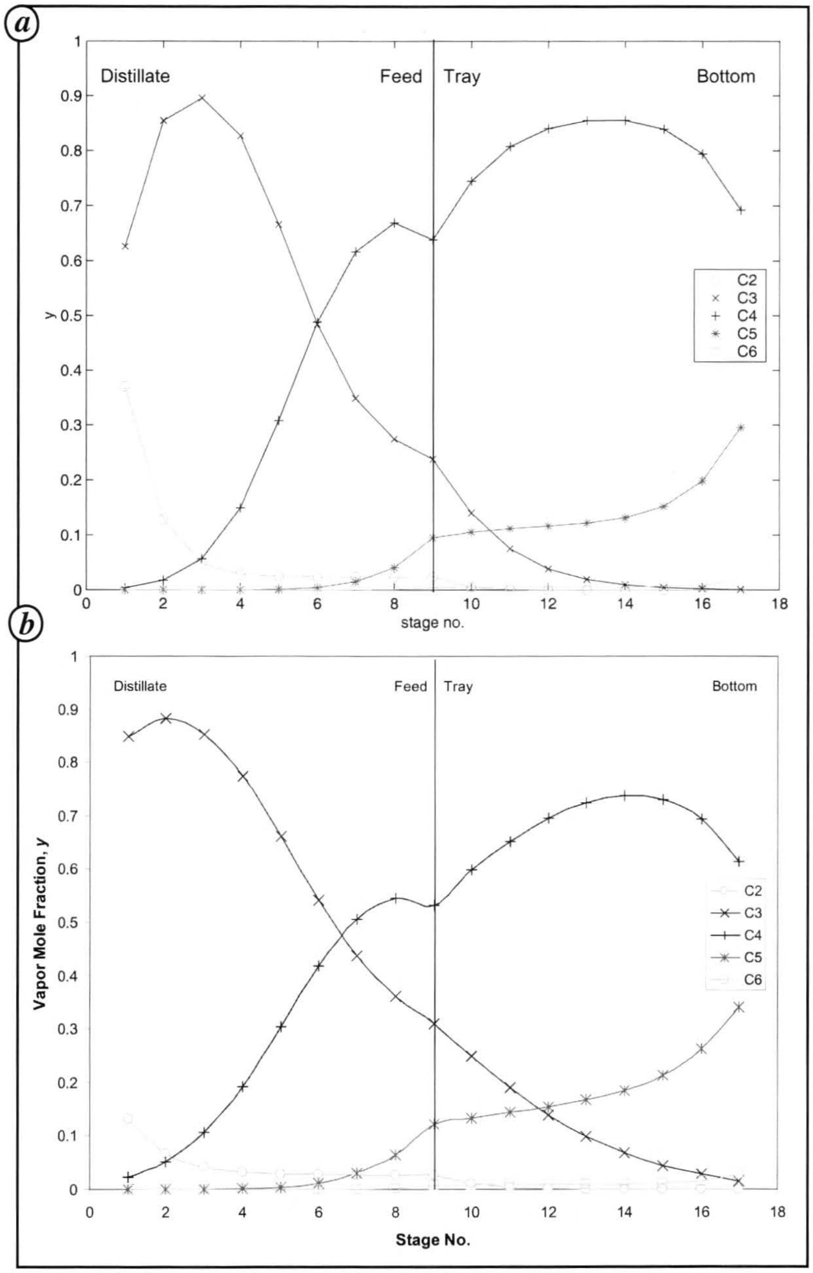


Figure 7. *Vapor composition of each component at each stage in Example 3: multicomponent distillation of hydrocarbons,* ***a)*** *obtained using direct matrix solver in MATLAB with a simple thermodynamic model (Raoult's law), and* ***b)*** *obtained from Aspen Plus with the Soave-Redlich-Kwong model. Operating conditions are listed in Table 4.*

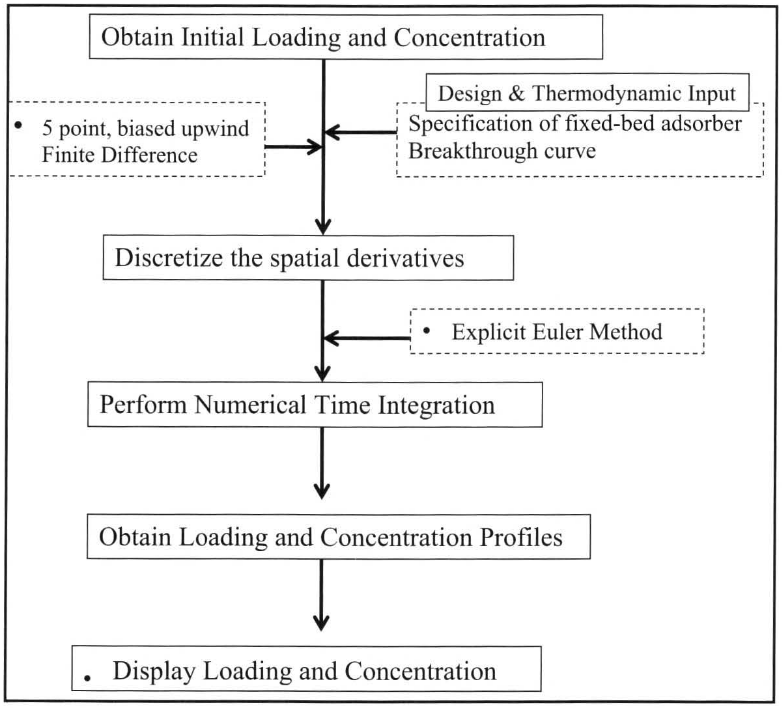

Obtain Initial Loading and Concentration
Design & Thermodynamic Input
• 5 point, biased upwind Finite Difference
Specification of fixed-bed adsorber Breakthrough curve
Discretize the spatial derivatives
• Explicit Euler Method
Perform Numerical Time Integration
Obtain Loading and Concentration Profiles
• Display Loading and Concentration

Figure 8. *Flowchart of Example 4: Thermal Swing Adsorption.*

- *Both display of conventional graphical methods and solving of complex systems of nonlinear equations can be achieved using MATLAB, which eliminates the requirement of multiple numerical tools in the course such as spreadsheet, for graphical methods, and computer languages, for numerical computation.*

The aforementioned integration of graphical display and computational approaches into various separation processes together with the implementation of emerging separation technologies and design aspects can provide students with the ability to choose an appropriate separation technology for a particular application, and to analyze the performance of modern separation processes. The MATLAB source codes and handouts for the examples can be downloaded from the home page of the Analysis of Separation Processes Course, Chemical Engineering 332 at Cornell University (<http://www.cheme.cornell.edu/courses/cheme332>).

ACKNOWLEDGMENTS

The authors thank the students and teaching assistants of ChemE332 for their feedback on the methods described in this paper. We also thank Professor T. Michael Duncan for insightful suggestions.

REFERENCES

1. Pratap, R., *Getting Started with MATLAB, A Quick Introduction for Scientists and Engineers,* Oxford University Press (2002)
2. Chickering, A.W., and Z.F. Gamson, "Appendix A: Seven Principles for Good Practice in Undergraduate Education, New Directions for Teaching and Learning," **47**, 63 (1991)
3. McCabe, W. L., J.C. Smith, and P. Harriott, *Unit Operations of Chemical Engineering,* 6th Ed., McGraw Hill (2001)
4. Seader, J.D., and E.J. Henley, *Separation Process Principles,* John Wiley & Sons (1998)
5. Humphrey, J.L., and G.E. Keller II, *Separation Process Technology,* McGraw-Hill, New York (1997)
6. King, C.J., *Separation Processes,* McGraw Hill (1980)
7. Wankat, P.C., "Teaching Separations: Why, What, When, and How," *Chem. Eng. Ed.*, **35,** 168 (2001)
8. Golnaraghi, M., P. Clancy, and K.E. Gubbins, "Improvements in the Teaching of Staged Operations," *Chem. Eng. Ed.*, **19**, 132 (1985)
9. Jolls, K.R., M. Nelson, and D. Lumba, "Teaching Staged-Process Design Through Interactive Computer Graphics," *Chem. Eng. Ed.*, **28**, 110 (1994)
10. Burns, M.A., and J.C. Sung, "Design of Separation Units Using Spreadsheets," *Chem. Eng. Ed.*, **30**, 62 (1996)
11. Hinestroza, J.P., and K. Papadopoulos, "Using Spreadsheets and Visual Basic Applications," *Chem. Eng. Ed.*, **37**, 316 (2003)
12. Dorgan, J.R., and J.T. McKinon, "Mathematica in the ChE Curriculum," *Chem. Eng. Ed.*, **30**, 136 (1996)
13. Rives, C., and D. Lacks, "Teaching Process Control with a Numerical Approach Based on Spreadsheets," *Chem. Eng. Ed.*, **36**, 242 (2002)
14. Wankat, P.C., "Integrating the Use of Commercial Simulators into Lecture Courses," *J. Eng. Ed.*, **91**, 19 (2002)
15. Wong, Y.W., and J.L. Niedzwiecki, "Simplified Model For Multicomponent Fixed Bed Adsorption," *AIChE Symposium Series*, **78**, 120-127 (1982) ❒

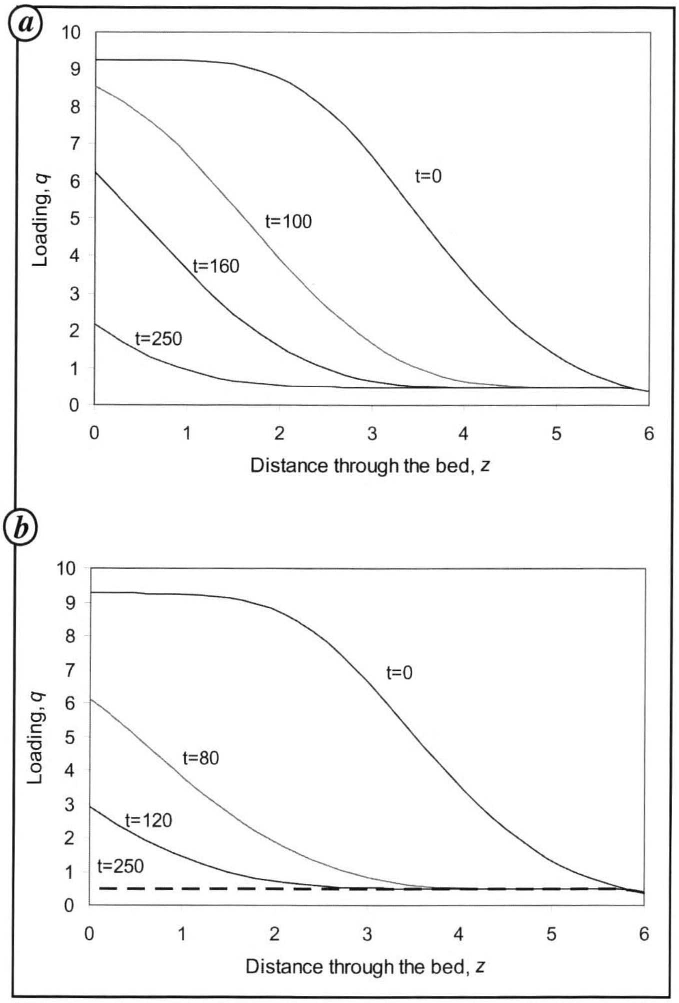


Figure 9. *Regeneration loading profiles in Example 4: Thermal Swing Adsorption with regeneration air interstitial velocity **a)** v = 30 m/min, and **b)** v = 60 m/min.*