Computation of Residue Curves using Mathematica® and MATLAB®

Housam Binous*
King Fahd University Petroleum and Minerals,
Dhahran, Saudi Arabia
* binoushousam@yahoo.com

Abstract

The Author presents a teaching technique used successfully in an undergraduate course to seniors at The National Institute of Applied Sciences and Technology (INSAT) in Tunis, Tunisia. The present paper details the computational aspects needed to obtain residue curves (RCs) for a binary and various ternary systems using Mathematica® and MATLAB®, which are taught in Tunisia. The first mixture studied is composed of ethanol – isopropanol. A ternary ideal mixture composed of benzene – toluene – p-xylene is treated next. A non-ideal ternary mixture (ethanol – water – ethylene glycol), using the Wilson model to predict liquid-phase activity coefficients, is also presented. Finally, a case, where both high pressure effect and deviation from ideal behavior in the liquid phase must be considered, using the Soave - Redlich - Kwong (SRK) equation of state (EOS) and the Wilson model, respectively, is studied. All problems are solved using the built-in Mathematica® and MATLAB® commands NDSolve and ode15s. The author concludes with insight to using residue curves (RCs) in order to teach distillation with the help of computer software such as Mathematica® and MATLAB®.

Key words: Residue Curve, distillation, Mathematica®, MATLAB®.

Introduction

Distillation is the most ubiquitous unit operation in the chemical industry. Thus, a good grasp of this separation method is essential both to chemical engineering students and professionals. By looking at the composition of the still during a simple batch distillation, one can plot a residue curve and gain insight to the product composition (distillate and bottom) of a particular distillation column. Indeed, for a ternary system for example, the location of the bottom and distillate are near both the residue curve and the distillation curve. One can plot what is called a residue curve map (RCM) by selecting different initial compositions in the still and plotting the corresponding residue curves. Another useful feature of residue curve maps is that the azeotropes appear as extremes (saddle points, stable nodes and unstable nodes; see Figure 1).

Figure 1 – Extreme points for the RCM

Thus, by computing residue curve maps, one finds the location of all azeotropes naturally. Sometimes RCMs show peculiar features such as distillation boundaries. This phenomenon leads to interesting conclusions about the feasibility of any distillation scheme. Indeed, a distillation boundary cannot usually be crossed by a single column unless it is curved. In the present paper, the author attempts to
provide students and professional with details about the programming techniques used to compute RCs with the help of modern scientific computing tools such as Mathematica® and MATLAB®. The first part of the paper gives the governing equations that one has to solve in order to compute RCs. A short description of the rules that allow vapor-liquid prediction, for ideal and non-ideal liquid and gas-phase behaviors, is given next. The first problem studied is a binary mixture composed of ethanol – isopropanol. The entire commented code, given in Appendixes 1 and 2, is described for an ideal ternary mixture composed of benzene – toluene – p-xylene both for Mathematica® and MATLAB®. A ternary system is then studied. The corresponding mixture is composed ethanol – water – ethylene glycol. This system is useful for the production of anhydrous ethanol using extractive distillation. In order to show how one performs calculation of RCs when there is deviation from ideal behavior in both the liquid and gas-phases, results concerning a mixture of chloroform – acetone – methanol are provided. In the last section, the author concludes by sharing his experience teaching distillation to undergraduate students at the National Institute of Applied Sciences and Technology (INSAT) in Tunis, Tunisia.

**Governing equations for Residue Curves**

A residue curve is obtained by solving the following equations:

\[
\frac{dx_i}{d\xi} = x_i - y_i \quad \text{for } i = 1 \text{ to } N_c \quad (1)
\]

where \(x_i\) and \(y_i\) are the liquid mole fraction of component \(i\) in the still and the equilibrium vapor mole fraction. \(\xi\) is the warped time defined by: \(d\xi = \frac{V}{H} \, dt\) where \(t\) is the clock time, \(H\) is the molar hold-up in the still and \(V\) is the vapor boil up rate. These equations are obtained from the overall material balance and the \((N_c - 1)\) independent component balances written for the simple batch distillation scheme shown in Figure 2. A full derivation of these equations is given by Doherty and Malone (2001).

For binary mixtures where the equilibrium behavior is modeled by the empirical relation:

\[
y = \frac{ax}{1+(a-1)x} + bx(1-x), \quad (2)
\]

the RCs are given by the following implicit equation:

\[
\xi = \frac{1}{a-1+b} \ln\left(\frac{x_0 (1-x)}{x (1-x_0)}\right) + \frac{(a-1)^2}{(a-1+b)(a-1+a b)} \ln\left(\frac{(1-x_0) (a-1+b [1+(a-1) x_0])}{(1-x) (a-1+b [1+(a-1) x])}\right). \quad (3)
\]

This equation is obtained by using Equations (1) and (2), the method of partial fractions and separation of variables. It is clear from Equation (1) that good vapor-liquid equilibrium prediction will be necessary in order to compute RCs. The next section presents extensive details about this important matter, which is ubiquitous in chemical separation science.
**Vapor-liquid equilibrium prediction**

Vapor pressure, a function of temperature only, is given by empirical equations such as the Antoine equation:

\[ \log(P_i^{sat}) = A_i - \frac{B_i}{C_i + T} \], \hspace{1cm} (4)

and the Modified Antoine equation:

\[ \ln(P_i^{sat}) = A_i + \frac{B_i}{C_i + T} + D_i \ln(T) + E_i T^F_i \], \hspace{1cm} (5)

where \( A_i, B_i, C_i, D_i, E_i \) and \( F_i \) are the constants, which depend on component \( i \).

If the pressure effects cannot be neglected, one has to use Eq. (6) to model vapor-liquid behavior:

\[ \varphi_i y_i P = x_i P_i^{sat} \gamma_i \], \hspace{1cm} (6)

where \( \varphi_i \) is the gas-phase fugacity coefficient always obtained from an EOS such as the Soave-Redlich-Kwong equation of state and \( \gamma_i \) is the activity coefficient given by the widely used Wilson model. This model allows good prediction of vapor-liquid data when one deals with mixtures that do not present demixtion into two liquid phases such as immiscible and partially miscible mixtures. The Wilson model (Wilson, 1964 and Sandler, 1999) is given by the following equations:

\[ \ln(y_k) = -\ln(\sum_{j=1}^{N_c} x_j A_{kj}) + 1 - \sum_{i=1}^{N_c} x_i \frac{A_{ki}}{\sum_{j=1}^{N_c} x_j A_{ij}} \], \hspace{1cm} (7)

In Equation (7), \( A_{ij} \) is the binary interaction parameter, which depends on the molar volumes \( (v_i \) and \( v_j) \) and the energy terms \( \lambda_{ii} \) and \( \lambda_{ij} \),

\[ A_{ij} = \frac{v_j}{v_i} \exp\left(-\frac{\lambda_{ij} - \lambda_{ii}}{R T}\right) \]. \hspace{1cm} (8)

The fugacity coefficient in the gas-phase is given by the following equation (Sandler, 1999 and Soave 1972):

\[ \varphi_i = \exp\left((Z - 1)\frac{p_i}{B} - \ln(Z - B) - \frac{A}{B} \left[ \frac{2A^{0.5}}{A^{0.5} - B_i} \right] \right) \ln\left(\frac{Z + B}{Z}\right) \] \hspace{1cm} (9)

In Equation (9), \( Z \) is the compressibility factor, which is a solution of the cubic equation (Sandler, 1999 and Soave 1972):

\[ Z^3 - Z^2 + Z (A - B - B^2) - AB = 0 \] \hspace{1cm} (10)

Constants \( A_i, B_i, A \) and \( B \) depend on the reduced pressure \( (P_r = \frac{p}{P_c}) \) and temperature \( (T_r = \frac{T}{T_c}) \), where \( P_c \) and \( T_c \) are the critical pressure and temperature, and the acentric factor, \( \omega \).

For pure species, one can use the following definitions (Sandler, 1999 and Soave 1972):

\[ A_i = 0.42747 a_i \frac{P_r}{T_r^{0.5}} \]

with \( a_i = \left[ 1 + m \left( 1 - T_r^{0.5} \right) \right]^2 \) and \( m = 0.480 + 1.574 \omega_i - 0.176 \omega_i^2 \), and \( B_i = 0.08664 \frac{P_r}{T_r^{0.5}} \).

Mixing rules for vapor mixtures can be written as follows:

\[ A = \sum_{i=1}^{N_c} \sum_{j=1}^{N_c} y_i y_j A_{ij} \]

where \( A_{ij} = (1 - k_{ij})(A_i A_j)^2 \), and \( B = \sum_{i=1}^{N_c} y_i B_i \). Values of the binary interaction parameters, \( k_{ij} \), are taken equal to zero if not available.
RCs calculation for binary mixtures

Consider a mixture composed of ethanol and isopropanol. Since this is an ideal mixture, we must set $b=0$ in Equations 2 and 3. For this particular mixture $a=1.9$ is the classical relative volatility constant. The total pressure is equal to 750 mmHg. The RC is plotted in Figure 3 and is obtained using Mathematica® 8.0. A neat feature of the new version of this computer algebra is the Manipulate built-in command. Indeed, as can be seen in Figure 3, there is a slider which allows the user to set the initial composition in the still. Since it is an ideal binary mixture, the RCs goes to zero, which means that the mixture gets depleted in ethanol, the light-boiling component of this mixture.

![Figure 3 – RC for the ethanol-isopropanol system at 750 mmHg](image)

The Mathematica® code for this example is given below:

```mathematica
Manipulate[Module[{plt}, a = 1.9; plt = Plot[
    1/(a - 1) 
    Log[x0/(1 - x)] + Log[1/(1 - x0)]},
   {x0, 0.05, 0.95}, Appearance -> "Labeled"],
  ControlPlacement -> Top]
```

Here, the author plots $\xi(x)$ and then uses the built-in command Reverse of Mathematica® to invert the axes in order to get the still’s composition versus warped time or $x(\xi)$.

RCs computation for an ideal ternary mixture

An RC, for an ideal ternary mixture composed of benzene – toluene – p-xylene at 1 atm, is shown in Figures 4. Raoult’s law is used to model the vapor-liquid equilibrium. Antoine’s constants are given in Table 1.

![Figure 4 – RC for the benzene – toluene – p-xylene mixture at 1 atm using Mathematica®](image)
Table 1 – Antoine’s constants for the benzene – toluene – p-xylene mixture

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benzene</td>
<td>6.87987</td>
<td>1196.76</td>
<td>219.161</td>
</tr>
<tr>
<td>Toluene</td>
<td>6.95087</td>
<td>1342.31</td>
<td>219.187</td>
</tr>
<tr>
<td>p-xylene</td>
<td>6.99053</td>
<td>1453.43</td>
<td>215.310</td>
</tr>
</tbody>
</table>

Table 1 – Antoine’s constants for the benzene – toluene – p-xylene mixture

The full computer codes using both Mathematica® and MATLAB® are given in the Appendixes 1 and 2, respectively. Toluene, the intermediate-boiling component is a saddle point. Benzene and p-xylene are the unstable and the stable nodes, respectively. The code uses the built-in commands of MATLAB® and Mathematica®, ode15s and NDSolve, respectively. Both commands can solve a system of differential algebraic equations (DAEs). The differential equations are obtained by writing Eq. (1) for the three compounds and the algebraic equation, which gives the bubble-temperature:

\[ P = x_1 P_1^{\text{sat}} + x_2 P_2^{\text{sat}} + x_3 P_3^{\text{sat}}. \]  

(11)

The code uses the new built-in Mathematica® command, LocatorPane, to get the coordinates of the computer’s mouse when user clicks the left button of the mouse:

\[
\text{LocatorPane}[Dynamic[p, (p = (#[[1]], Min[1 - #[[1]], #[[2]]]) &)],
\]

A small mathematical trick (in the command: \(Min[1 - #[[1]], #[[2]]]\)) avoid having mole fractions that are either negative or above 100%. An initial still’s composition is assigned these particular mouse coordinates. The code then integrates the system of DAEs for this initial composition and draws the RC passing by this user-specified point.

The MATLAB® code operates in a similar manner and uses the built-in MATLAB® command, ginput, in order to determine the initial still’s composition, which corresponds to the position of the computer’s mouse when the user clicks the left button of the mouse.

**RC computation for ethanol – water – ethylene glycol mixture**

The tables 2, 3 and 4 give the modified Antoine’s constants and the data for the Wilson model.

Table 2 – Antoine’s constants for the water – ethanol – ethylene glycol mixture (with \(F=2\) and \(C=0\))

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water</td>
<td>65.9278</td>
<td>-7227.53</td>
<td>-7.17695</td>
<td>4.0313e-6</td>
</tr>
<tr>
<td>Ethanol</td>
<td>86.4860</td>
<td>-7931.10</td>
<td>-10.2498</td>
<td>6.38949e-6</td>
</tr>
<tr>
<td>Ethylene glycol</td>
<td>57.9410</td>
<td>-8860.70</td>
<td>-5.71660</td>
<td>3.10800e-6</td>
</tr>
</tbody>
</table>

Table 3 – binary interaction parameters for the water – ethanol – ethylene glycol mixture

<table>
<thead>
<tr>
<th></th>
<th>Water</th>
<th>Ethanol</th>
<th>Ethylene glycol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water</td>
<td>18.1</td>
<td>58.7</td>
<td>55.9</td>
</tr>
<tr>
<td>Ethanol</td>
<td>1043.84800</td>
<td>-129.2041</td>
<td>0</td>
</tr>
<tr>
<td>Ethylene glycol</td>
<td>276.75570</td>
<td>266.4848</td>
<td>1539.411</td>
</tr>
</tbody>
</table>

Table 4 – molar volumes for the water – ethanol – ethylene glycol mixture

From the RCs (see Figure 5), it is clear that there is no distillation boundary since all RCs start from the binary azeotrope (ethanol-water with 88% ethanol mole fraction) and end-up at the ethylene glycol corner. Thus, the unique azeotrope is an unstable node; ethylene glycol is a stable node and both ethanol and water are saddle point. Again, one can drag and place the Locator at any position inside the triangle.
and the corresponding RC will be plotted instantaneously. Ethylene glycol can serve as an entrainer in order to perform extractive distillation (see Figure 6) and obtain anhydrous ethanol. This can represent a good alternative to the well-known three columns Kubierschky system using heteroazeotropic distillation and cyclohexane as an entrainer.

RC computations for the high pressure case

Figure 7 shows a RC for the ternary system chloroform – acetone – methanol at P=10 atm. The tables 8, 9 and 10 give the modified Antoine’s constants and the data for the Wilson model.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chloroform</td>
<td>73.7</td>
<td>-6055.6</td>
<td>-8.9189</td>
<td>7.74407e-6</td>
</tr>
<tr>
<td>Acetone</td>
<td>71.3</td>
<td>-5952.8</td>
<td>-8.531</td>
<td>7.82393e-6</td>
</tr>
<tr>
<td>Methanol</td>
<td>59.83</td>
<td>-6282.8</td>
<td>-6.378</td>
<td>4.61746e-6</td>
</tr>
</tbody>
</table>

Table 8 – Antoine’s constants for the chloroform – acetone – methanol mixture (with F=2 and C=0)

<table>
<thead>
<tr>
<th></th>
<th>Chloroform</th>
<th>Acetone</th>
<th>Methanol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chloroform</td>
<td>0</td>
<td>116.1711</td>
<td>1694.0240</td>
</tr>
<tr>
<td>Acetone</td>
<td>-506.8518</td>
<td>0</td>
<td>551.4545</td>
</tr>
<tr>
<td>Methanol</td>
<td>-361.7944</td>
<td>-124.932</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 9 – binary interaction parameters for the chloroform – acetone – methanol mixture
Table 10 – molar volumes for the chloroform – acetone – methanol mixture

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Chloroform</td>
<td>Acetone</td>
<td>Methanol</td>
</tr>
<tr>
<td></td>
<td>80.7</td>
<td>74.0</td>
<td>40.7</td>
</tr>
</tbody>
</table>

The critical temperature and pressure and the acentric factor are given in Table 11.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Chloroform</td>
<td>Acetone</td>
</tr>
<tr>
<td></td>
<td>263.2</td>
<td>5370</td>
</tr>
<tr>
<td></td>
<td>235.0</td>
<td>4700</td>
</tr>
<tr>
<td></td>
<td>239.4</td>
<td>7376</td>
</tr>
</tbody>
</table>

The compositions of the azeotropes and their corresponding bubble-temperature, at a pressure equal to 10 atm, are given in Table 12. The three binary azeotropes are obtained as a result of the RCM calculation. For the ternary azeotrope, accurate composition and bubble-temperature computation required a separate calculation using the built-in command of Mathematica®, FindRoot, which can find the solution of systems of nonlinear algebraic equations.

The cubic equation, which solution is the compressibility factor:

\[
\text{NDsolve}\left\{\text{D}[x1[t],\{t,1\}] = y1[t] - x1[t],\text{D}[x2[t],\{t,1\}] = y2[t] - x2[t],\right. \\
x1[t] + x2[t] + x3[t] = 1, y1[t] + y2[t] + y3[t] = 1, \\
y1[t] = \frac{x1[t] PSat1[t] GAM1[t]}{P\phi[1][t]}, \\
y2[t] = \frac{x2[t] PSat2[t] GAM2[t]}{P\phi[2][t]}, \\
y3[t] = \frac{x3[t] PSat3[t] GAM3[t]}{P\phi[3][t]}, \\
T[0] = (T/\text{sol1}), x1[0] = p[1][1], x2[0] = p[2][1], x3[0] = 1 - p[1][1] - p[2][1], \\
Z[0] = (Z/\text{sol1}), y1[0] = (y1/\text{sol1}), y2[0] = (y2/\text{sol1}), y3[0] = (y3/\text{sol1}), \\
\{x1, x2, x3, T, Z, y1, y2, y3, \{t, -100, 100\}\}
\]

Teaching distillation and RCMs to senior students at INSAT

Dr. Housam Binous started teaching RCM calculation at INSAT in 2002. A derivation of Equations 1 and 3 is presented in the first lecture. VLE prediction is taught in a two-hour separate lecture. Next, the author moves on to the application of RCMs in order to understand complex distillation problems such as azeotropic, extractive and heteroazeotropic distillation of ternary systems where one of the component is an entrainer. A brief discussion of entrainer selection criteria is given too. In addition, separation of partially miscible mixtures such as the n-butanol-water mixture using distillation column and decanters is presented. Finally, the author usually ends this part of his course with a brief presentation of pressure swing distillation.
and heat-integrated distillation columns. Since 2006, a four-hour computer laboratory gives students hand-on experience on VLE data computation and RCMs calculations using Mathematica®. Typical course enrollment was less than 50 students, all majoring in chemical engineering. Students where split into four groups for the laboratory sessions where they get hand-on experience using Mathematica® and MATLAB®. Students usually have an important course load with all of them taking as much as six two-hour lectures per week and a four-hour computing laboratory every other week. Feedback from student was both generally very positive and very rewarding to the author; especially that our senior students perceive the fact of obtaining in a couple of hours and without much difficulty a residue curve for an ideal ternary mixture as a small academic achievement and a yet another reason to gain confidence in their potentials.

Conclusion

In the present paper, several example of RC calculation were presented. Binary and ternary mixtures were considered as well as ideal and non-ideal VLE behavior. Sample code, given both in the text and in Appendixes 1-2, should make extension to other mixtures, with or without liquid-phase chemical reaction, quite straightforward. The author hope that this paper will help students and professionals alike to learn how to compute RCs and RCMs using state-of-the-art computer programs such as Mathematica® and MATLAB®. Even though, numerous commercial simulators such as DISTIL® and ASPEN®, allow the determination of RCM, it is useful to know how such computations are performed and how to interpret the results that are obtained. All Mathematica® and MATLAB® code is available upon request from the author or at the Wolfram Demonstration Project (Binous, 2009) and the MATLAB® File Exchange Center (Binous, 2008).

References


Nomenclature

\[ P : \text{total pressure (bar)} \]
\[ P_{c,i} : \text{critical pressure (bar)} \]
\[ P_r : \text{reduced pressure} \]
\[ P_i : \text{partial pressure (bar)} \]
\[ P_{sat} : \text{vapor pressure (bar)} \]
\[ t : \text{time (hr)} \]
\[ T : \text{temperature (K)} \]
\[ T_{c,i} : \text{critical temperature (Kelvin)} \]
\[ T_r : \text{reduced temperature} \]
\[ x : \text{liquid composition (mole fraction)} \]
\[ y : \text{vapor composition (mole fraction)} \]
\[ Z : \text{compressibility factor} \]

Greek Letters

\[ \phi : \text{fugacity coefficient} \]
\[ \gamma : \text{activity coefficient} \]
\[ \xi : \text{warped time} \]
Acknowledgment
The support of King Fahd University of Petroleum and Minerals in duly acknowledged.

Biographical Information
Dr. Housam Binous, a visiting Associate Professor at King Fahd University Petroleum & Minerals, has been a full time faculty member at the National Institute of Applied Sciences and Technology in Tunis for eleven years. He earned a Diplôme d’ingénieur in biotechnology from the Ecole des Mines de Paris and a Ph.D. in chemical engineering from the University of California at Davis. His research interests include the applications of computers in chemical engineering.

Appendix 1 – Mathematica® code

```mathematica
Manipulate[
DynamicModule[
{sol1, sol2, x1, x2, x3, y1, y2, y3, PS1, PS2, PS3, P = 760,
(* Antoine Constants Antoine Equation and Raoult’s Law *)
A1 = 6.87987, B1 = 1196.76, C1 = 219.161,
A3 = 6.95087, B3 = 1342.31, C3 = 219.187,
A2 = 6.99053, B2 = 1453.43, C2
= 215.31}, PS1[t_]
:= 10^A1 - \frac{B1}{C1 + T[t]};
PS2[t_]:= 10^{A2 - \frac{B2}{C2 + T[t]}};PS3[t_]
:= 10^{A3 - \frac{B3}{C3 + T[t]}};y1[t_]
= \frac{PS1[t]*x1[t]}{P};
y2[t_] = \frac{PS2[t]*x2[t]}{P}; y3[t_] = \frac{PS3[t]*x3[t]}{P};
(* Solving the system of DAEs *)
Dynamic[sol1 = Quiet[NDSolve[{x1'[t] =
= y1[t] - x1[t], x2'[t] =
= y2[t] - x2[t],
\[\Box]3'[t] == y3[t] - x3[t], x1[t] PS1[t]
+ x2[t] PS2[t] + x3[t] PS3[t]
- P == 0, T[0] == 380,
x1[0] == p[[1]], x2[0] == p[[2]], x3[0] =
= 1 - p[[1]] - p[[2]]),
{x1, x2, x3, T}, {t, 0, 10})];
sol2 = Quiet[NDSolve[{ -x1'[t] =
= y1[t] - x1[t], -x2'[t] =
= y2[t] - x2[t],
-x3'[t] == y3[t] - x3[t], x1[t] PS1[t]
+ x2[t] PS2[t] + x3[t] PS3[t]
- P == 0,
T[0] == 380, x1[0] == p[[1]], x2[0] =
= p[[2]],
x3[0] =
= 1 - p[[1]]
- p[[2]]}, {x1, x2, x3, T}, {t, 0, 10})];

LocatorPane[
Dynamic[p, (p
= {#[[1]], Min[1
- #[[1]], #2[[2]]])))&,
(* Plotting the results *)
Dynamic[ParametricPlot[
Evaluate[{x1[t], x2[t]}/. sol1],
Evaluate[{x1[t], x2[t]}/. sol2}, {t, 0, 10},

PlotStyle -> {{Thickness[0.005], Red},
{Thickness[0.005], Red}},
PlotRange -> {{0, 1}, {0, 1}},
AxesLabel -> {benzene,p-xylene},

Epilog

\rightarrow \{Text[toluene,{-0.1,-0.05},{-1.0}],
Line[{{1.0},{0.1}}]}}];
```
Appendix 2 - MATLAB® code

```matlab
h=figure(1);
line([0 1], [1 0])
axis([0 1 0 1])
hold on

button = questdlg('do you want to draw a residue curve','...','yes','no','yes');

while (button(1)=='y')

    % determination of the position of the mouse
    [x y] = ginput(1);
    if x<0
        x=max(x,0);
    end;
    if y<0
        y=max(0,y);
    end;
    if y+x>1
        y=min(1-x,y);
    end;

    % Solving the system of DAEs
    opts = odeset('Mass','M','MassSingular','yes');
    [t X]=ode15s('RCM1',[0 10],[x y 1-x-y 350],opts);
    plot(X(:,1),X(:,2),'r')

    button = questdlg('do you want to draw a residue curve','...','yes','no','yes');
end

close(h)

function xdot=RCM1(t,x,flag)

    if isempty(flag),
        T=x(4);
        % Total Pressure
        P=760;
        % Antoine constants
        A1=6.87987;B1=1196.76;C1=219.161;
        A2=6.99053;B2=1453.43;C2=215.310;
        A3=6.95087;B3=1342.31;C3=219.187;
        % Antoine Equation
        PS1=10^(A1-B1/(C1+T));
        PS2=10^(A2-B2/(C2+T));
        PS3=10^(A3-B3/(C3+T));
        % Raoult's Law
        y1=PS1*x(1)/P;
        y2=PS2*x(2)/P;
        y3=PS3*x(3)/P;
        % System of DAEs
        xdot(1)=x(1)-y1;
        xdot(2)=x(2)-y2;
        xdot(3)=x(3)-y3;
        xdot(4)=y1+y2+y3-1;
        xdot = xdot';  % xdot must be a column vector
    else
        % Return M
        M = zeros(4,4);
        M(1,1) = 1;
        M(2,2) = 1;
        M(3,3) = 1;
        xdot = M;
    end
```