1.1 Viscosity of Liquids

The physical property that characterizes the flow resistance of simple fluids is called the \textit{viscosity}. In this lesson, the plot of the viscosity is plotted versus temperature. Recall that the viscosity of a liquid is given by:

\[
\mu = (\frac{\delta}{a})^2 \frac{Nh}{V} e^{\frac{\Delta G_o}{RT}}
\]

where \((\frac{\delta}{a})\) is approximately unity.

- \(N\) is the Avogadro \# \((6.023 \times 10^{23} \text{ g moles}^{-1})\).
- \(h\) is Planck’s constant \((6.624 \times 10^{-27} \text{ erg sec})\).
- \(V\) is the molar volume of a liquid.
- \(R\) is the molar gas constant.
- \(T\) is the input temperature measure in Kelvin.
- \(\Delta G_o = 0.408 \Delta U_{vap}\).
- \(\Delta U_{vap}\) is the internal energy of vaporization at normal boiling point.
- Trouton’s rule for boiling point: \(\Delta U_{vap} = 9.4RT_b\).
- \(T_b\) Boiling Point Temperature.

The equation for \(\mu\) can be simplified to depend only on: \(T, V\) and \(T_b\). Thus, this describes a function:

\[
\mu = \mu(T, T_b, V)
\]
where the constants $V$ and $T_b$ for different liquids are given below. Also included are the melting points $T_m$.

<table>
<thead>
<tr>
<th>Liquid</th>
<th>$V$ ($cm^2 g^{-1} mole^{-1}$)</th>
<th>$T_b$ (Kelvin)</th>
<th>$T_m$ (Cent)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benzene</td>
<td>89.0</td>
<td>353.3</td>
<td>5.5</td>
</tr>
<tr>
<td>Carbon Tetrachloride</td>
<td>327.0</td>
<td>349.8</td>
<td>-22.6</td>
</tr>
<tr>
<td>Cyclohexane</td>
<td>324.</td>
<td>356.13</td>
<td>6.5</td>
</tr>
</tbody>
</table>

### 1.2 Method

The objective of this lesson is to construct a plot of the viscosity of three liquids versus temperature. In order to obtain a plot it is necessary to first obtain the values of the viscosity of Benzene, $\mu_{Bz}$, Carbon Tetrachloride, $\mu_{C4Ch}$, and Cyclohexane, $\mu_{CHx}$, with respect to temperature.

Consider calculation the values of the viscosity of Benzene for $n$ equally spaced values of the temperature between Benzene melting temperature $T_m$ and its boiling temperature $T_b$. In order to obtain $n$ equally spaced values the spacing between them will be $\text{step} = (T_b - T_m)/(n - 1)$.

**Algorithm** Calculate $\mu_{Bz}$ versus $T$ at equally spaced points:

- $\text{delTemp} = (T_b - T_m)/(n - 1)$
- $T = T_m$
- loop on index from 0 to $n-1$
  - $T = T + \text{index} \times \text{delTemp}$
  - $\mu_{Bz} = \mu(T, T_b, V_{Bz})$
- End loop on index

The calculations of the viscosities of Carbon Tetrachloride, $\mu_{C4Ch}$, and Cyclohexane, $\mu_{CHx}$, would follow similar algorithm where $T_m$, $T_b$, and $V$ are changed to the corresponding values.

Before proceeding to discuss implementation we want to discuss certain aspects of the algorithm.

- There is a “loop” construct that repeats every step in the algorithm between step [4.] and step [5.] as the value of index changes from 1 up to $n$. Please note that, we have assumed that the increments on the variable index is 1 each time through the loop.
- The temperature, $T$ is updated each time through the loop. The values of $T$ progress from $T_m$ up to $T_b$ on equal increments delTemp.
1.3 Implementation

There is a construct that would simplify the implementation of the Algorithm; the counting loop construct. C++ provides such construct.

Most loops are actually quite simple. The counting loop construct needed here is the most common kind of loop. One variable counts up or down through the integers for a fixed numbers of steps.

for( initialization; conditional; increment) // begin for-loop construct
{
    // ....
    // .  // for-loop-block of statements
    // ....
} // end of for-loop.

- The initialization statement can be a declaration statement such as int indx = 0. It must give the starting value for the index.
- The conditional is actually a conditional expression representing a boolean value: True or False by integers (zero for false and non-zero for true).
- The increment statement is used to step through a set of indices. For instance, indx++ would be appropriate. The programming language C++ has a number of built-in incremental operations.

<table>
<thead>
<tr>
<th>C++ Operation</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>x++</td>
<td>post increment</td>
</tr>
<tr>
<td>++x</td>
<td>pre increment</td>
</tr>
<tr>
<td>x–</td>
<td>post decrement</td>
</tr>
<tr>
<td>–x</td>
<td>pre decrement</td>
</tr>
</tbody>
</table>

Consider the following segment of a C++ program:

```cpp
int min, max, sum, fstevn;
...
sum = 0;
fstevn = min;
if ( 2*(min/2) < min ) fstevn++;
for( int countr = fstevn; countr < max; countr= countr+2)
{
    sum = sum + countr;
}
```

The program statement `for( countr = fstevn; countr < max; countr= countr+2)` tells C++ that the initial value of `countr` should be the value of `fstevn`, that the loop should continue as long as the value of `countr` is less than the value of `max`, and that the value of `countr` should be increased by 2 each time.

1.3.1 Implementation of Algorithm

The first thing is to describe the function that computes the value of both viscosity of a liquid. The function `liquid_viscosity( .. )` achieves that and its declaration is given below.

```cpp
float liquid_viscosity( float temp, float boilingP, float molarVal);
```

The calling program is as follows:
// beginning of program liquidViscosityPlots.cpp
#include <fstream>
#include <iostream>

using std::ofstream;
using std::cout;
using std::cin;
using std::endl;

ofstream Benzout("muBenzene.txt");
ofstream C4Chout("muCCl4.txt");
ofstream CHxout("muCycloHex.txt");

float liquid_viscosity( float temp, float boilingP, float molarVal);

void main()
{
    /*
    Written by JCDiaz, Dec 29 2000
    Changed by JCDiaz, Feb 2004
    It computes the values of the viscosity of a liquid
    at a number of equally spaced points between the
    melting point and the boiling point for each liquid.
    The liquid properties are given by the table:

    Liquid    Tmelt   Tboil   Mol-V
    Benzene   89.0    353.3   5.5
    Carbon Tetrachloride 327.0  349.8  -22.6
    Cyclohexane 324.0  356.13  6.5

    It invokes the function liquid_viscosity()
    variables used:
    */
    // int index; // Loop index counter
    float T_melting; // Melting Temperature of Liquid in Kelvin
    float T_boiling; // Boiling Temperature of Liquid in Kelvin
    float V_molar;  // Molar Volume of Liquid
    float T;        // Temperature
    float mu;      // viscosity
    float deltaT;  //increment value for water temperature
    int num_pts;   // number of points in the graph.

    cout << " Please enter the number of points desired in the plot"<<endl;
cin >> num_pts;

    /* Case of Benzene */
    V_molar = 89.0;
    T_boiling = 353.3;
    T_melting = 273.16 + 5.5 ;
    deltaT = (T_boiling - T_melting) / (num_pts -1);
for ( int index = 0; index < num_pts; index++ )
{
    T = T_melting + index * deltaT;
    mu = liquid_viscosity( T, T_boiling, V_molar);
    Benzout << T <<", " << mu <<endl;
} /* End case of Benzene */

/* Case of Carbon Tetrachloride */
V_molar = 327.0;
T_boiling = 349.3;
T_melting = 273.16 -22.6;
deltaT = (T_boiling - T_melting ) / (num_pts -1);
for ( int index = 0; index < num_pts; index++ )
{
    T = T_melting + index * deltaT;
    mu = liquid_viscosity( T, T_boiling, V_molar);
    C4Chout << T <<", " << mu <<endl;
} /* End case of Carbon Tetrachloride */

/* Case of Cyclohexane */
V_molar = 324.0;
T_boiling = 356.13;
T_melting = 273.16 +6.5;
deltaT = (T_boiling - T_melting ) / (num_pts -1);
for ( int index = 0; index < num_pts; index++ )
{
    T = T_melting + index * deltaT;
    mu = liquid_viscosity( T, T_boiling, V_molar);
    CHxout << T <<", " << mu <<endl;
} /* End case of Cyclohexane */

} // end of program liquidViscosityPlots.cpp

The program and function are saved in the file liquidViscosityPlots.cpp.

1.4 Assessment

The assessment consist on the visualization of the results of the program. The desired final result consist of a computer picture that has all three viscosity plots on it.
1. Viscosity of Liquids: Loops and Scientific Visualization

1.4.1 Matlab Plot

The interactive environment matlab is used to plot the three viscosities as functions of temperature. The matlab command plot is used to obtain the desirable figure.

Various line types, plot symbols and colors may be obtained with PLOT(X,Y,S) where X and Y are vectors, and S is a character string made from one element from any or all the following 3 columns:

<table>
<thead>
<tr>
<th>Character</th>
<th>Color</th>
<th>Symbol</th>
<th>Line Style</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>yellow</td>
<td>.</td>
<td>point</td>
</tr>
<tr>
<td>m</td>
<td>magenta</td>
<td>o</td>
<td>circle</td>
</tr>
<tr>
<td>c</td>
<td>cyan</td>
<td>x</td>
<td>x-mark</td>
</tr>
<tr>
<td>r</td>
<td>red</td>
<td>+</td>
<td>plus</td>
</tr>
<tr>
<td>g</td>
<td>green</td>
<td>*</td>
<td>star</td>
</tr>
<tr>
<td>b</td>
<td>blue</td>
<td>s</td>
<td>square</td>
</tr>
<tr>
<td>w</td>
<td>white</td>
<td>d</td>
<td>diamond</td>
</tr>
<tr>
<td>k</td>
<td>black</td>
<td>v</td>
<td>triangle (down)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>triangle (up)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;</td>
<td>triangle (left)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt;</td>
<td>triangle (right)</td>
</tr>
<tr>
<td>p</td>
<td></td>
<td>p</td>
<td>pentagram</td>
</tr>
<tr>
<td>h</td>
<td></td>
<td>h</td>
<td>hexagram</td>
</tr>
</tbody>
</table>

Examples:

1. PLOT(X,Y,'gp:') plots the vector Y versus vector X using a green dotted line with a pentagram at each data point.
2. PLOT(X,Y,'ko') plots the vector Y versus vector X using a black circle at each data point but does not draw any line.
3. PLOT(X,Y,'r-',X,Y,'ys') plots the data twice, with a solid red line interpolating yellow squares at the data points.
4. PLOT(X1,Y1,S1,X2,Y2,S2,X3,Y3,S3) combines the plots defined by the (X,Y,S) triples, where the X's and Y's are vectors and the S's are strings.

For more information on plot, please issue the matlab command help plot.

1.4.2 Plot of Liquid Viscosities

The following sequence of matlab command yields a plot of the three viscosities.

1. load muBenzene.txt.
2. set Sb='yp-', Xb=muBenzene(:,1) and Yb=muBenzene(:,2),
3. load muCCl4.txt.
4. set Scc='cx-.', Xcc=muCCl4(:,1) and Ycc=muCCl4(:,2), and
5. load muCycloHex.txt.
6. set Scy='rv--', Xcy=muCycloHex(:,1) and Ycy=muCycloHex(:,2).
7. the matlab PLOT command: \texttt{plot(Xb,Yb,Sb,Xcc,Ycc,Scc,Xcy,Ycy,Scy)} produces the desired graph.

8. the matlab LEGEND command: \texttt{legend('Benzene','CCl4','Cyclohexane')} is used to label each of the viscosity graphs with the liquids name.

The resulting graph can be captured with \texttt{xv} and saved as \texttt{muGraph.jpg}.

---

1.5 Relative Permeability

Relative permeability is a dynamic quantity in that it changes from rock to rock and from fluid to fluid. This includes the rock and fluid geometry. Other arguments include the pore-size distribution of a rock, the heterogeneity of the rock, and wettability characteristics of the rock in relation to the fluids present. Several methods exist for calculating the relative permeability. These involve lab measurements and empirical correlations which give acceptable accuracy for engineering purposes.

The relative permeability curves are also affected by the physical process under consideration. It can be
1. Viscosity of Liquids: Loops and Scientific Visualization

a drainage process where the water saturation (wetting phase) is being decreased. Or, it can be an imbibition process where the water saturation (the wetting phase) is being increased.

The amount of fluid present of each phase is represented by two variables $S_w$, the water saturation, and $S_o$, the oil saturation. The saturations are measured as a fraction of the pore volume of the rock. The assumption of immiscible fluids such as oil and water implies that $S_w + S_o = 1$. Hence, $S_o = 1.0 - S_w$.

The phase being increased is the “displacing” phase. The displacing phase saturation is given by $S_d$. The relative permeability curves are given in terms of the normalized saturation function

\[ S^* = \frac{S_d}{1 - S_w} \]

where $S_wi$ is the irreducible water saturation. Let $S_oi$ be the irreducible oil saturation.

The type of curve depends on the system being modeled. General equations for the relative permeability curves are:

- **Imbibition Process**

  \[
  S_d = S_w \\
  S^* = \frac{S_d}{1 - S_wi} \\
  k_{rw} = \frac{(1 - 2S^*)^p}{(1 + 2S^*)^q} \\
  k_{ro} = (S^*)^r.
  \]

  Where $p = 3/2$, $q = 1/2$, and $r = 4$ for Naar-Henderson’s approximation. Notice that $p$, $q$ are fractional exponents. Hence, the quantities $(1 - 2S^*)$ and $(1 + 2S^*)$ must be positive. Let $S_wo$ be the largest value of $S_w$ for which these quantities are both positive. Let $S_oi = 1 - S_wo$. The function $k_{rw}$ is extended as having the value of zero (0.0) beyond $S_wo$.

- **Drainage Process**

  \[
  S_d = S_o \\
  S^* = \frac{S_d}{1 - S_wi} \\
  k_{rw} = (1 - S^*)^j \\
  k_{ro} = (S^*)^k(2 - S^*).
  \]

  Where $j = 4$ and $k = 3$ for Corey’s approximation. In this case $S_wo = 1$. However, since $S_oi$ is not negligible, it may be approximated with $S_oi = S_wi$. Hence $S_wo = 1.0 - S_oi$. The function $k_{rw}$ is extended as having the value of zero (0.0) beyond $S_wo$.

The engineer/scientist would modify the equation for relative permeability by making the required changes in the exponents which would make the relative permeability curve more closely duplicate the data.

1.6 Rel Perm Lab

The objective of this laboratory is to construct a plot of $k_{rw}$ and $k_{ro}$ versus $S_w$ where $S_wi \leq S_w \leq 1.0 - S_oi$ for Corey’s approximation.
1. Change directory into your es2503Programs directory.

2. Write a function that calculates Corey’s relative permeabilities for given $S_w$ and $S_{wi}$ and store it in a file named relPerm.cpp under your es2503Programs directory.

3. Modify relPerm.cpp to include a main() program that reads irreducible water saturation $S_{wi}$ from the user input. The program should call the relative permeability function for 20 equally spaced values of $S_w$ where $S_{wi} \leq S_w \leq 1.0 - S_{oi}$. The output should go to the files kroDrain.dat, krwDrain.dat.

4. Once your program works correctly, run it for the case of $S_{wi} = .10$.

5. Use matlab commands to display the corresponding pair $kro$ and $krw$ on one plot. Capture with xv and save as relpermDrain.jpg

6. Hand in relPerm.cpp, kroDrain.dat, krwDrain.dat, and relpermDrain.jpg.

7. Copy the kro, krw, and relPerm files into your public_html/es2503Reports directory.

8. Change directory to your public_html/es2503Reports directory, and create the report file relpermReport.html which explains the lab and includes relative links to the kro, and krw files, and includes as in-lined image relpermDrain.jpg.


11. Clean up your home directory and any of your public areas to make sure that no files containing your program or portion thereof are accessible by any user.

12. Submit a Journal relating your experiences with this assignment.