Cell 1: Bill Knowlton

Extra Credit Assignment #4

Your Assignment: (NOTE: *USE YOUR OWN EQUATION*)

1. Using *Mathematica*, use the Style Sheet called ArticleModern (under Format –> Style Sheet) for this assignment.
Use various Styles (under Format –> Style) to describe your assignment and program. This Cell is using the Subsection Style.

2. Plot an equation (USE YOUR OWN EQUATION) in 2 dimensions that is a function of multiple variables. An example is shown below. You will need to use several commands including:

a. Plot[ ]

b. YourEquation[var1_,var2_,...,var#_]:=equation

c. Print[ ] to output several values of interest

3. Plot your function over 2 variables but in 2 dimensions on many plots.
4. Plot your function over 2 variables but in 2 dimensions all on one plot.
5. Plot your function over 2 variables in 3 dimensions.
6. Plot your function over 2 variables in 2 dimensions suppressing multiple plots and graphing the multiple plots in one plot.

- **Examples:**

Problem 2.

In the Input Cell below, I use the ideal gas law showing a direct relation with temperture, T, and an

inverse relation with pressure and is given by:

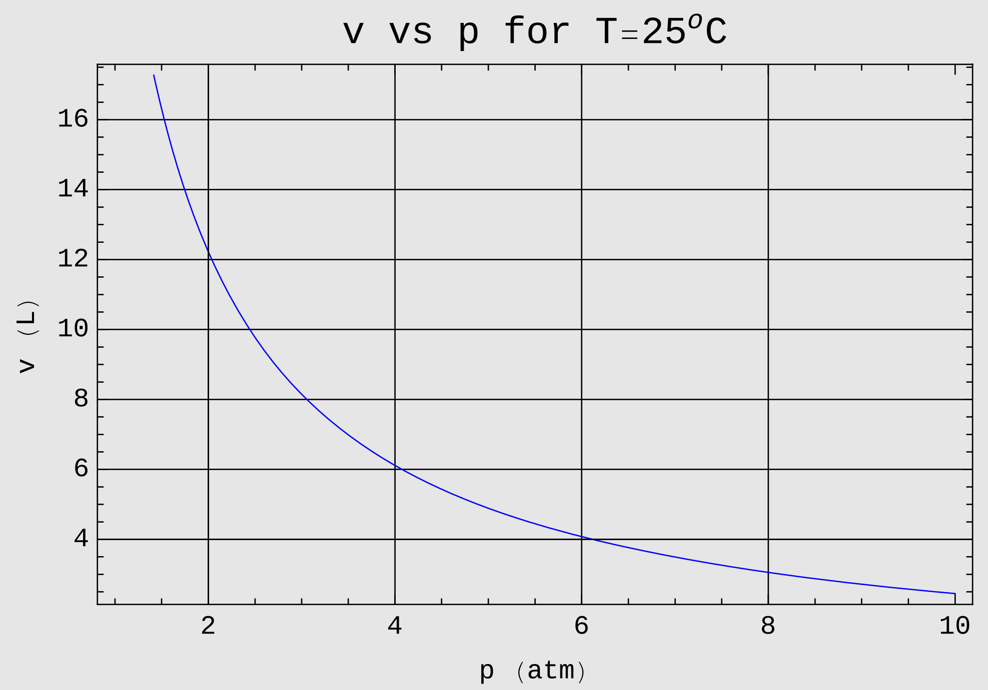
```
v[p_, n_, T_] := n R T / p
```

Note that there are other variables, but they are not included in the bracket [ ]. The variables included in the [ ] are call Arguments allows us to vary or give a value directly for each variable in the [ ]. For more information, see Help –> Master Index –> [ ].

Lastly, to add a comment in the Input Cell that *Mathematica* will ignore, use (* comment *). I use this to document what I'm up to in the program.

In[31]:=
```
Clear[v, R, T, p]
v[p_, n_, T_] := n R T / p
(*Constants*)
R = 0.08206;(*atm L /(mol K)*)

(*Plots*)
Plot[v[p, 1, 298], {p, 1, 10}, Frame → True, GridLines → Automatic,
 PlotStyle → {RGBColor[0, 0, 1]}, FrameLabel → {"p (atm)", "v (L)"},
 PlotLabel -> "v vs p for T=25oC"]
Print["v: ", v[1, 1, 298], "L at T=25oC"]
```

Out[34]=



v: 24.4539L at T=25°C

- **Problem 3.**

Plot your function over 2 variables. If I want to graphically examine a function in 2D but with 2 variables, one way I do so is shown below.

The easiest way to show all of the graphs together would be to use the Table[ ] command instead of using a "For loop" or "Do loop" approach. The Table is called gasplots, and contains the same Print and Plot commands as in the For loop. The second argument {T,300,750,50}, iterates T from 300K to 750K in steps of 50, which returns the same result as the For loop.

Note that this is essentially a program because it includes a For loop type functionality and a Print command and a Plot command and a Table command. Note that there are multiple [ ]s.

In[36]:=

```
gasplots = Table[
  Print["T= ", T, " K"];
  Plot[v[p, 1, T], {p, 1, 10}, Frame → True, GridLines → Automatic,
   PlotStyle → {RGBColor[1, 0, 0]}, FrameLabel → {"p (atm)", "v (L)"},
   PlotLabel -> "v for varies Temps"], {T, 300, 750, 50}]
```

T= 300 K

T= 350 K

T= 400 K

T= 450 K

T= 500 K

T= 550 K

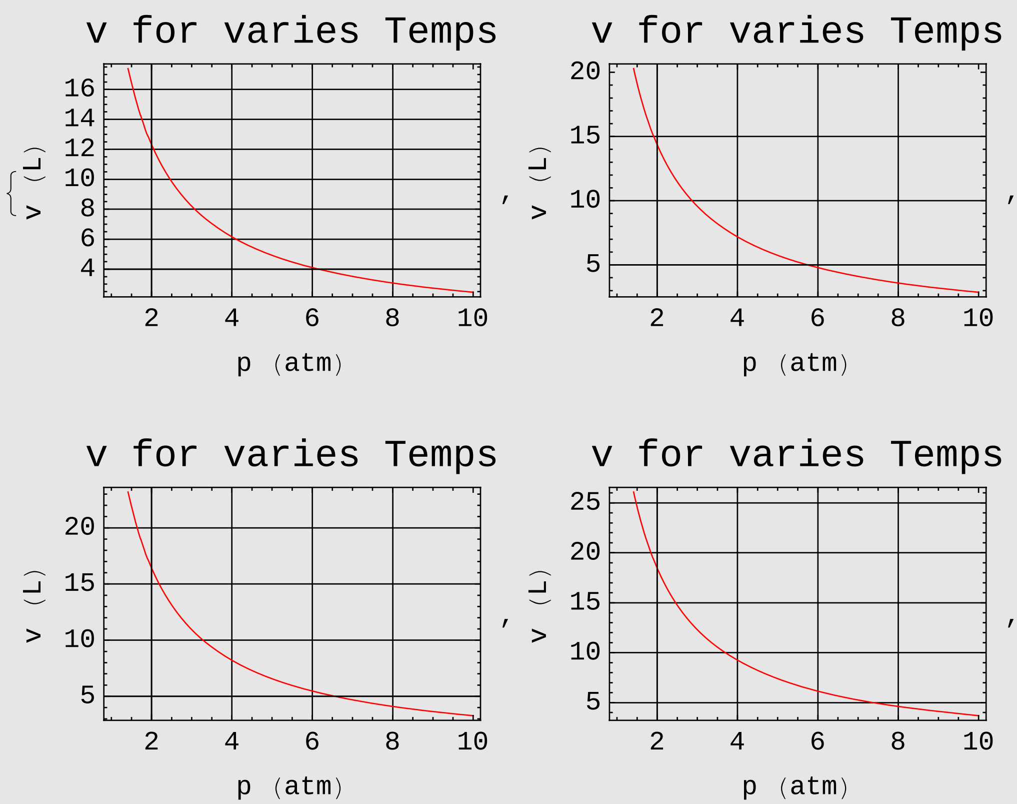
T= 600 K

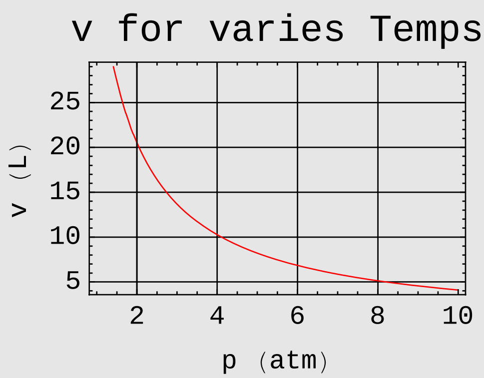
T= 650 K

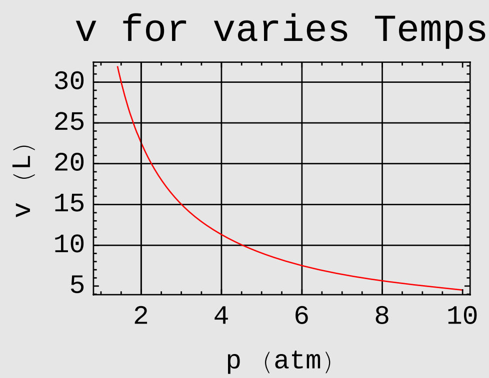
T= 700 K

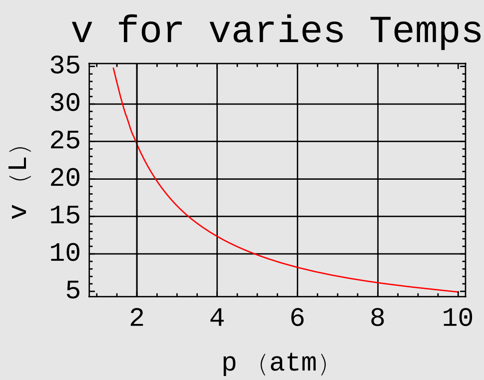
T= 750 K

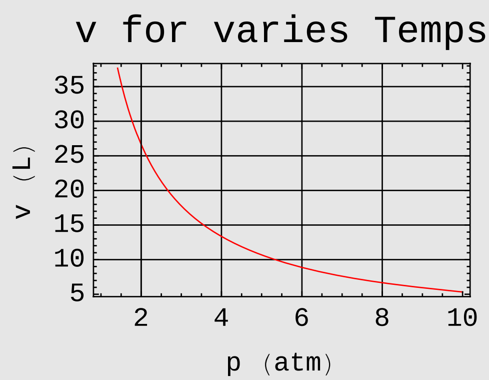
Out[36]=

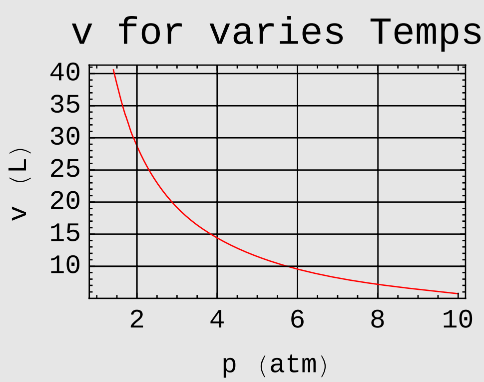


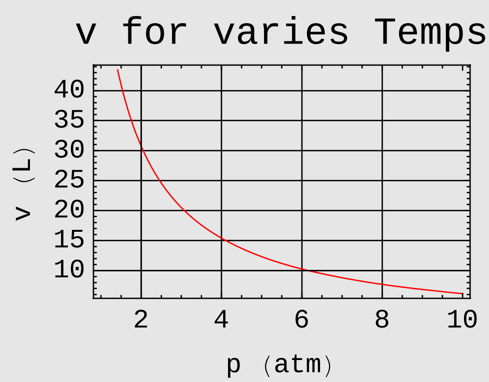


,


,


,


,


,


}

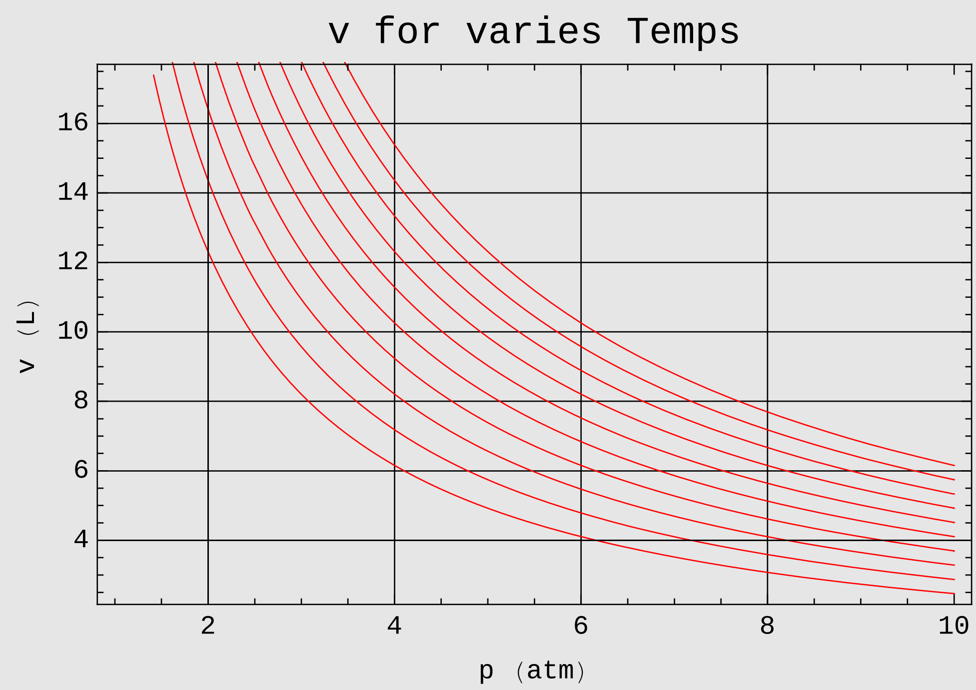
■ Problem 4.

Plot the above plots in one graph.

One way to plot all the data in one graph is to use the Show[ ] command.

In[37]:=
```
Show[gasplots]
```

Out[37]=



■ Problem 5.

Suppressing multiple plots and graphing the multiple plots in one plot.

1st: Setting up the problem by defining a function. We will defing the function ΔGmix as a function of 3 variables.

In[38]:=

```
Clear[ai, aj, R, x1, x2, ΔGmix, T, b]
ΔGmix[x1_, x2_, T_] :=
  x1 x2 (ai + (aj - ai) x2) + R T (x1 Log[x1] + x2 Log[x2]);
```

2nd: Defining constants:

In[40]:=

```
(*Constants*)
ai = 12 500; aj = 5500; R = 8.314; (*J/molK*)
```

3rd: Defining a table "Giplots" to hold an array of ΔGmix outputted by the Plot[ ] command.

Note:

In the Plot[ ]command at the end, I use the command: "DisplayFunction→Identity"

This tells the Plot[ ] not to graph any plots.

In[41]:=
```
Giplots = Table[
   Plot[ΔGmix[1 - x2, x2, T], {x2, 0, 1}, Frame → True, GridLines → Automatic,
    PlotStyle → {RGBColor[0, 0, 1]}, FrameLabel → {"X2", "ΔGmix (J/mol)"},
    PlotLabel -> "ai=12500; aj=5500;", DisplayFunction → Identity],
   {T, 200, 1200, 100}];
Print["Temp. Range: 200K to 1200K @ 100K Increments"]
Print["Curve at top: 200K; Curve at bottom: 1200K;"]
```

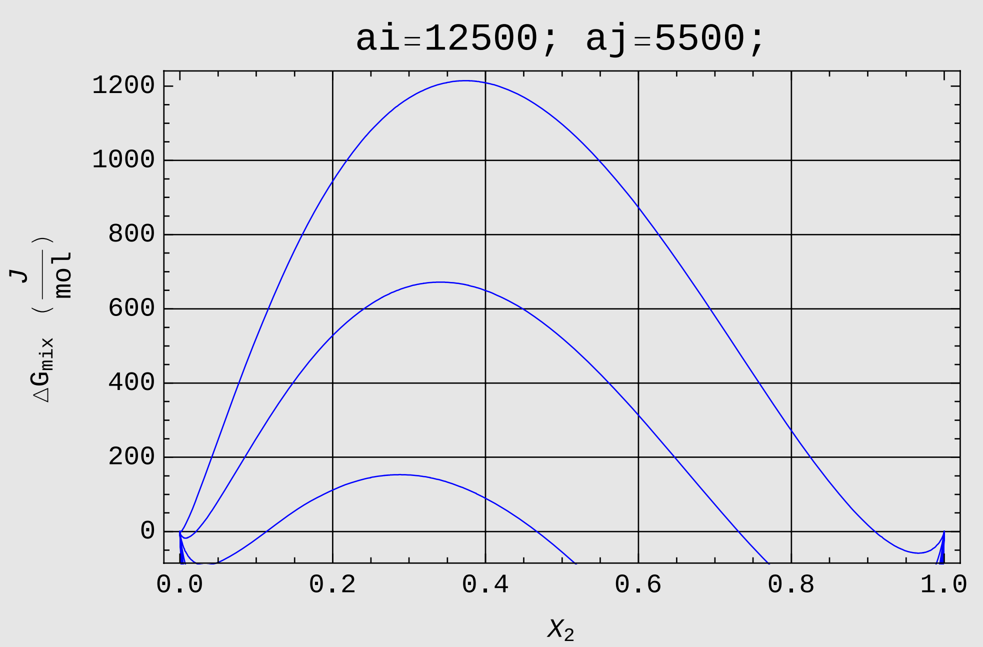
```
Temp. Range: 200K to 1200K @ 100K Increments
Curve at top: 200K; Curve at bottom: 1200K;
```

4th: I use the Show[ ] command to graph all the plots together. Since I told the Plot[ ] command not to plot, then I NEED to tell the Show[ ] command to tell the Plot [ ] command to graph the data. This is done by adding into the Show[ ] the following: "DisplayFunction->$DisplayFunction" .

So I use Show[Giplots, DisplayFunction -> $DisplayFunction] to plot all the plots

In[44]:=
```
Show[Giplots, DisplayFunction → $DisplayFunction]
```

Out[44]=



- **Problem 6.**

Plot the above plots in one graph WITHOUT using the Table[ ] command. Rather, use the Range[ ]

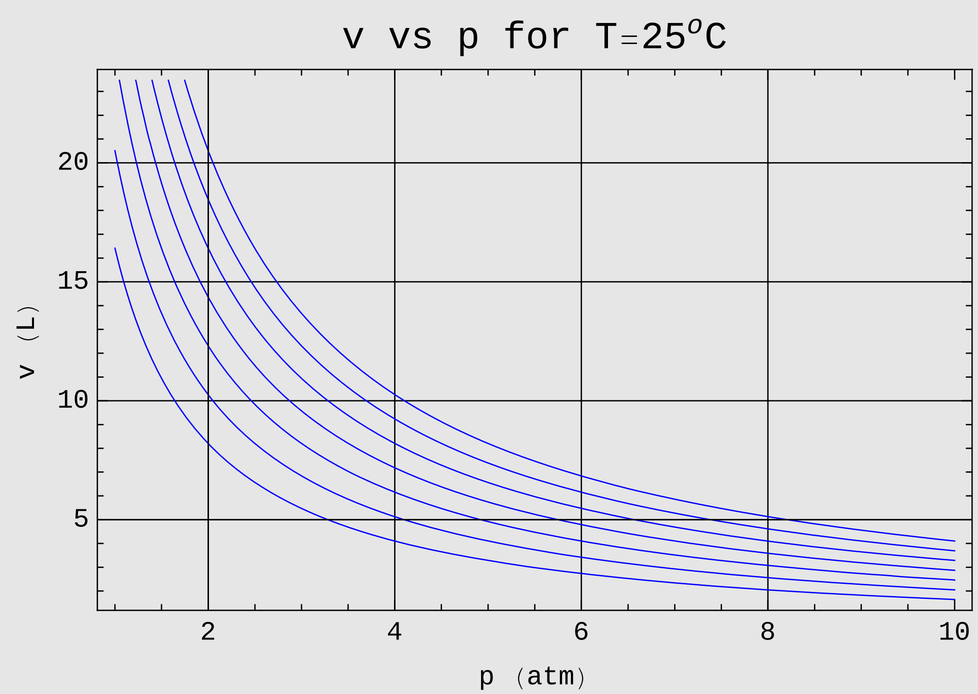
and Evaluate[ ] commands to plot over two variables.

See p. 30-31 of Hollis' A *Mathematica* Companion for Differential Equations

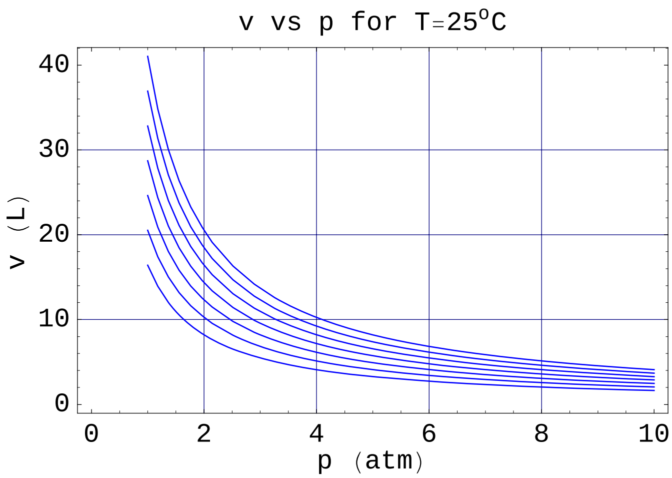
In[45]:=

```
Clear[v, R, T, p]
v[p_, n_, T_] := n R T / p
R = 0.08206`;
imin = 200; imax = 500; step = 50;
T = Range[imin, imax, step];
Plot[Evaluate[v[p, 1, T]], {p, 1, 10}, Frame → True, GridLines → Automatic,
 PlotStyle → {RGBColor[0, 0, 1]}, FrameLabel → {"p (atm)", "v (L)"},
 PlotLabel → "v vs p for T=\!\(\*SuperscriptBox[\(25\), \(o\)]\)C"]
Print["v: ", v[1, 1, 298],
 "L at T=\!\(\*SuperscriptBox[\(25\), \(o\)]\)C"]
```

Out[50]=



v: 24.4539L at T=25^{o}C



v: 24.4539L at T=25^{o}C

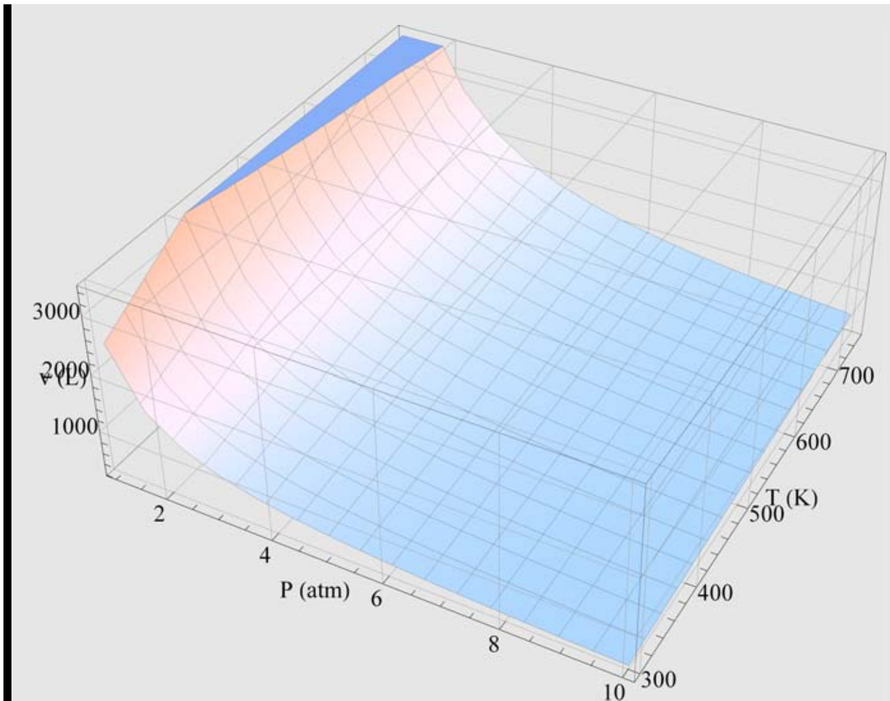
■ Problem 7.

Plot the function in 3 dimensions.

Use the Plot3D[ ] command.

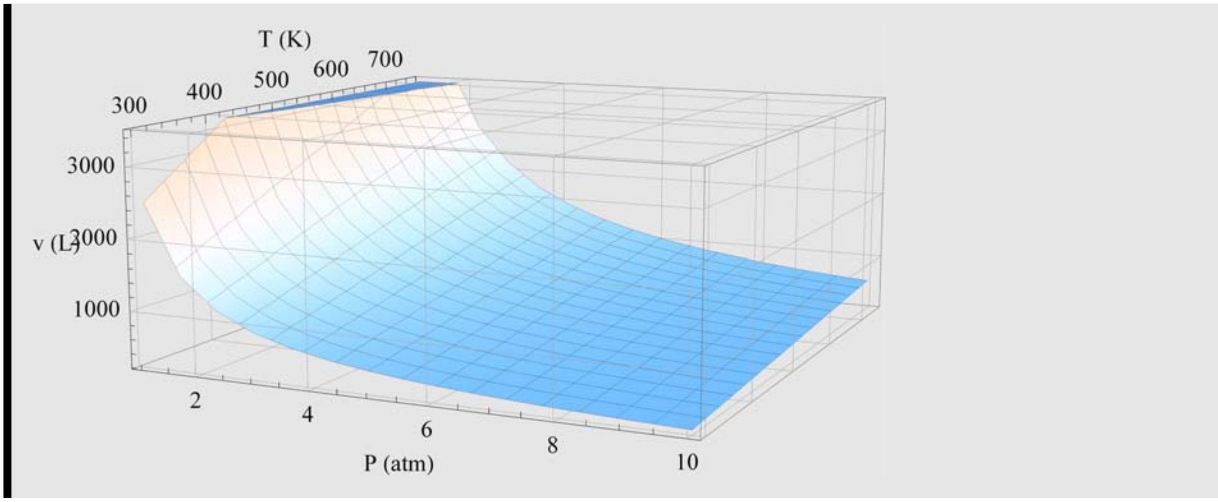
In[65]:=
```
Plot3D[v[p, 1, T], {p, 1, 10}, {T, 300, 750}, Mesh → True, FaceGrids → All,
 AxesLabel → {"P (atm)", "T (K)", "v (L)"}]

(*Used 3D Viewpoint Selecter in the Input pulldown menu for
 viewing plot from a different viewpoint*)
Print["You can click on the 3D plot and move it with your mouse."]
Plot3D[v[p, 1, T], {p, 1, 10}, {T, 300, 750}, Mesh → True, FaceGrids → All,
 AxesLabel → {"P (atm)", "T (K)", "v (L)"},
 ViewPoint -> {1.402, -3.025, 0.580}]
```

Out[65]=




```
You can click on the 3D plot and move it with your mouse.
```

Out[67]=



Problem 8.

Using Manipulate[ ]

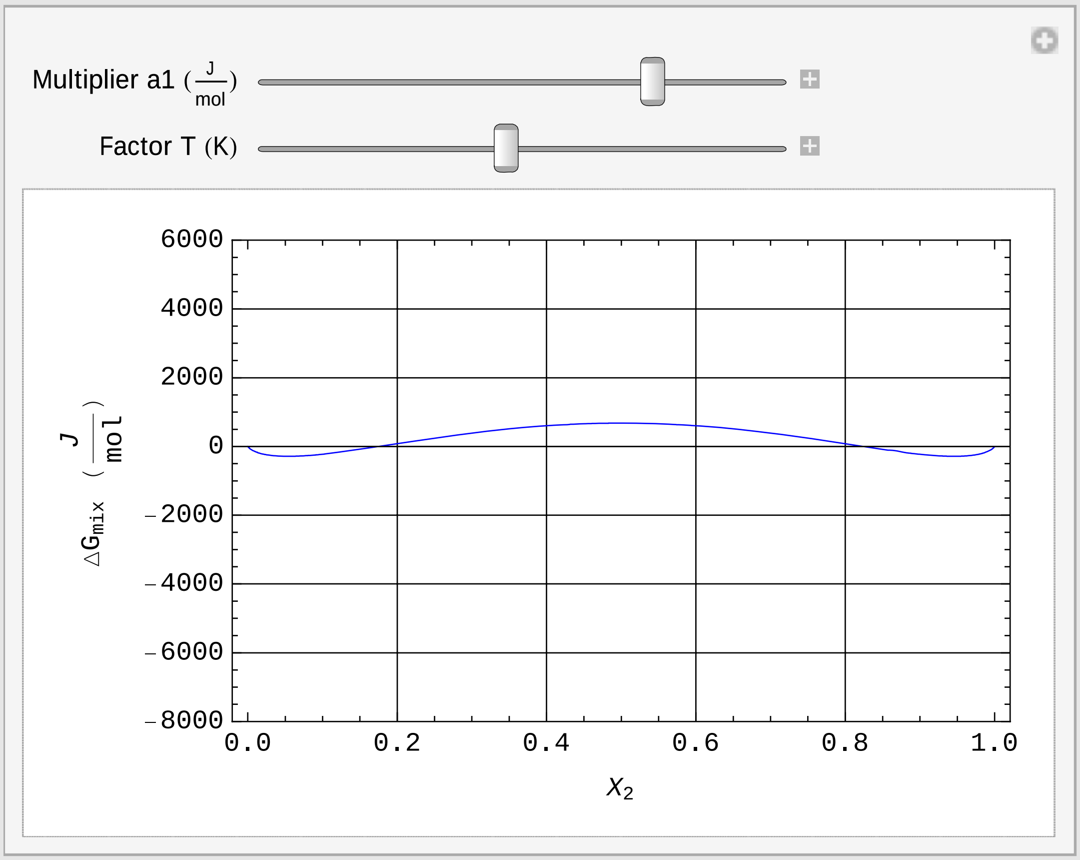
The Manipulate[ ] command provides interactive plots. Variables other than the independent and dependent variables are displayed as sliders and are the interactive part of the plot. The sliders can be moved with the mouse and the plot will change as the slider is moved.

Here is an example plot with 2 sliders.

In[55]:=

```
(*Constants*)
Clear[R, x1, x2, ΔGmix, T, a1]
 R = 8.314; (*J/molK*)

ΔGmix[x2_, T_] := a1 (1 - x2) x2  + R T ((1 - x2) Log[1 - x2] + x2 Log[x2]);
Manipulate[Plot[a1 (1 - x2) x2  + R T ((1 - x2) Log[1 - x2] + x2 Log[x2]),
  {x2, 0, 1}, PlotRange → {-8000, 6000}, Frame → True,
  GridLines → Automatic, PlotStyle → {RGBColor[0, 0, 1]},
  FrameLabel → {"X2", "ΔGmix (J/mol)"}],
 {{a1, 20 000, "Multiplier a1 (J/mol)"}, 3000, 25 000},
 {{T, 750, "Factor T (K)"}, 100, 1500}]
```

Out[58]=



Here is an example plot with 3 sliders.

In[59]:=

```
(*Constants*)
(*ai=12500; aj=5500;*) R = 8.314; (*J/molK*)
Manipulate[
 Plot[(1 - x2) x2 (ai + (aj - ai) x2) + R T ((1 - x2) Log[(1 - x2)] + x2 Log[x2]),
  {x2, 0, 1}, PlotRange → {-8000, 6000}, Frame → True,
  GridLines → Automatic, PlotStyle → {RGBColor[0, 0, 1]},
  FrameLabel → {"X2", "ΔGmix (J/mol)"}],
 {{ai, 12500, "Multiplier ai (J/mol)"}, 3000, 20000},
 {{aj, 5500, "Multiplier aj (J/mol)"}, 3000, 10000},
 {{T, 300, "Factor T (K)"}, 100, 1500}]
```

Out[60]=