Computer Applications in Chemical Engineering

Solution of Equation of state

Solving equation of state allows us to find the specific volume (molar volume) of gaseous mixture of chemicals at a specified temperature and pressure. Without using equation of state, it would be virtually impossible to design a chemical plant.
The ideal gas equation of state relates the pressure, temperature, and the molar volume:

$$P V_m = R T \quad \text{where} \quad V_m = \frac{V}{n}$$

This equation is quite adequate when the pressure is low (e.g. 1 atm). To overcome this, many equation of state have been developed such as these listed below:

van der Waals (vdw), 1873 $$P = \frac{RT}{V_m - b} - \frac{a}{V_m^2}$$

Redlich-Kwong (RK), 1949 $$P = \frac{RT}{V_m - b} - \frac{a}{V_m (V_m + b)\sqrt{T}}$$

Soave (RKS), 1972 $$P = \frac{RT}{V_m - b} - \frac{a\,\alpha(T)}{V_m (V_m + b)}$$

Peng-Robinson (PR), 1976 $$P = \frac{RT}{V_m - b} - \frac{a\,\alpha(T)}{V_m (V_m + b) + b(V_m - b)}$$

where a and b are constants and α is a parameter depends on temperature.

Note that all the previous equations are non-linear equations in one unknown (Vm) if the temperature, pressure, and the equation parameters are given. The solution of such equations will be illustrated below.

Finding the zero of a function of one unknown

MATLAB uses the function *fzero* to solve the following problem $f(x) = 0$ for x starting from an initial guess. The function syntax is:

```
fzero('function_nam', guess)
      |              |
      |              └ an initial guess for the solution
      └ The name of the function file which contains the equation to be solved.
```

The solution steps are:

1. make the function equal to zero.
2. Define the function in a function file using the m-file editor (e.g. test.m).
3. Suggest an initial guess for the solution (e.g. 0).
4. Issue the following command in the command window or your script file:

```
>> fzero('test', 0)
ans =
    □  ← A value returned for x that makes the function f(x) equal to zero.
```

Note that you can check the result by saying:

```
>> test(ans)
     |   |
     |   └ function name saved in step 2.
     └ The result of step 4.
```

Ex Find the roots of the equation:

$$x + 2e^{-x} = 3$$

near $x = -0.5$ and $x = 3$.

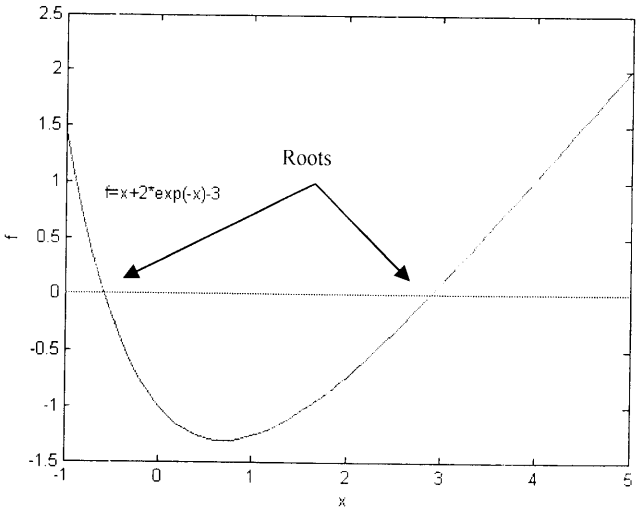
Solution

let $f(x) = x + 2e^{-x} - 3 = 0$

```
function [f]=test(x)
f=x+2*exp(-x)-3;
```

Session:

```
>> x=fzero('test',-0.5)
x =
   -0.5831
>> x=fzero('test',3)
x =
    2.8887
```



Note

If the function have more than one solution as the above function, fzero returns the nearest solution to the guess.

Van der Waals equation of state

Pressure

$$P = \frac{RT}{V_m - b} - \frac{a}{V_m^2}$$

Compressibility factor

$$Z = \frac{V_m}{V_m - b} - \frac{a}{RTV_m}$$

where

n	total number of components
x_i	the mole fraction of component i

$$a = \left(\sum_{i=1}^{n} x_i \sqrt{a_i} \right)^2 \qquad a_i = \frac{27(RT_{ci})^2}{64P_{ci}}$$

$$b = \sum_{i=1}^{n} x_i b_i \qquad b_i = \frac{RT_{ci}}{8P_{ci}}$$

Residual Enthalpy, Residual Entropy, and Fugacity Coefficient:

$$\frac{H_m^{res}}{RT} = \frac{b}{V_m - b} - \frac{2a}{RTV_m}$$

$$\frac{S_m^{res}}{R} = \ln\left[Z\left(1 - \frac{b}{V_m}\right)\right]$$

$$\ln\phi_i = \frac{b_i}{V_m - b} - \frac{2\sqrt{aa_i}}{RTV_m} - \ln\left[Z\left(1 - \frac{b}{V_m}\right)\right]$$

Ex Find the molar volume for ammonia gas at 56atm and 450K using the vander Waals equation of state, given: $T_c = 405.5\,K$, $P_c = 111.3\,atm$

$R = 0.08206\ l\,atm/mol\,K$

(Hint: obtain an initial estimate from the ideal gas law)

MATLAB Session:

```
>> clear
>> global R T P Tc Pc
>> R=0.08206;T=450;P=56;
>> Tc=405.5;Pc=111.3;
>> v_guess=R*T/P
v_guess =
    0.6594
>> Vm=fzero('vdw1',v_guess)
Vm =
    0.5749
```

```
function [f]=vdw1(v)
global R T P Tc Pc
a=27*(R*Tc)^2/(64*Pc);
b=R*Tc/(8*Pc);
f=P-R*T/(v-b)+a/v^2;
```

Ex Continue from the previous session and calculate the residual entropy for ammania gas at the same conditions.

Continue from the previous session:

```
>> a=27*(R*Tc)^2/(64*Pc);
>> b=R*Tc/(8*Pc);
>> Z=Vm/(Vm-b)-a/(R*T*Vm);
>> ResS=R*log(Z*(1-b/Vm))

ResS =

   -0.0168
```

Q Why you should determine the constants a and b again in this session in order to calculate the S_m^{res} ??

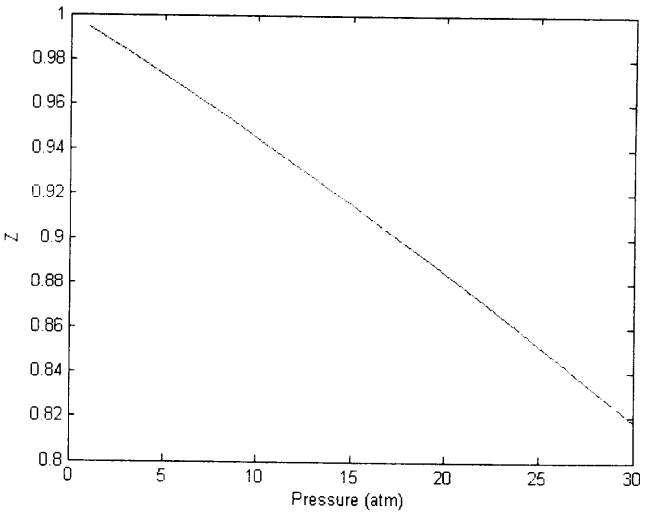
EX Write MATLAB code to compute the compressibility factor for a number of pressure values (1–30 atm) at 500K for n-butane given: $T_c = 425.2K$

$P_c = 37.5$ atm

$R = 0.08206$

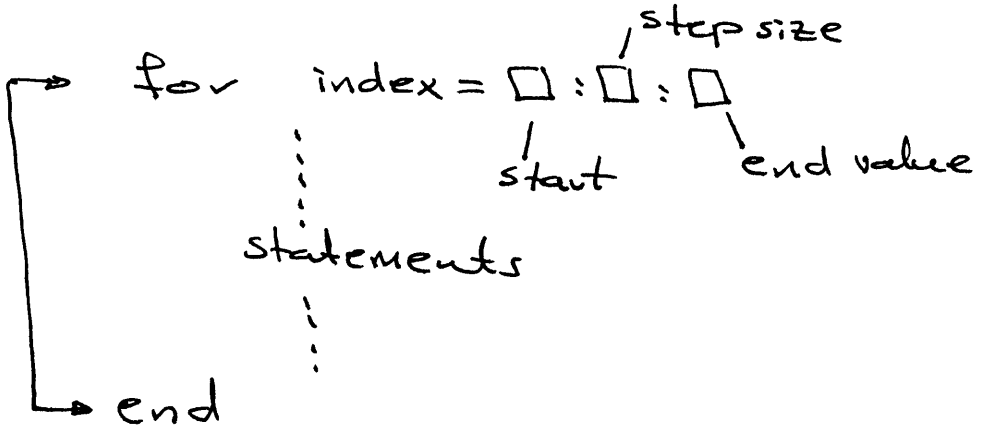
m-file (myfile.m)

```
clear
global R T P Tc Pc
T=500;R=0.08206;
Tc=425.2;Pc=37.5;
for i=1:30
    P=i;
    press(i)=i;
    v_guess=R*T/P;
    Vm(i)=fzero('vdw1',v_guess),
    Z(i)=P*Vm(i)/(R*T);
end
plot(press,Z)
xlabel('Pressure (atm)')
ylabel('Z')
```



The for loop

The for loop is a loop that executes a block of statements a specified number of times.



Redlich-Kwong equation of state:

$$P = \frac{RT}{V_m - b} - \frac{a}{V_m(V_m + b)\sqrt{T}}$$

$$a = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \left(1 - k_{ij}\right)\sqrt{a_i a_j} \qquad a_i = 0.42748\frac{R^2 T_{ci}^{2.5}}{P_{ci}}$$

$$b = \sum_{i=1}^{n} x_i b_i \qquad b_i = 0.08664\frac{RT_{ci}}{P_{ci}}$$

k_{ij} Binary interaction parameter (Note: K_{ii} =0)

Soave equation of state:

$$P = \frac{RT}{V_m - b} - \frac{a\alpha}{V_m(V_m + b)}$$

$$a\alpha = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \left(1 - k_{ij}\right)\sqrt{(a_i\alpha_i)(a_j\alpha_j)} \qquad b = \sum_{i=1}^{n} x_i b_i$$

$$a_i = 0.42747\frac{(RT_{ci})^2}{P_{ci}} \qquad b_i = 0.08664\frac{RT_{ci}}{P_{ci}}$$

$$\alpha_i = \left[1 + n_i\left(1 - \sqrt{T_{ri}}\right)\right]^2 \qquad n_i = 0.48508 + 1.55171\omega_i - 0.15613\omega_i^2$$

ω_i Acentric factor

Peng-Robenson equation of state:

$$P = \frac{RT}{V_m - b} - \frac{a\alpha}{V_m(V_m + b) + b(V_m - b)}$$

$$a\alpha = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \left(1 - k_{ij}\right)\sqrt{(a_i\alpha_i)(a_j\alpha_j)} \qquad b = \sum_{i=1}^{n} x_i b_i$$

$$a_i = 0.45724\frac{(RT_{ci})^2}{P_{ci}} \qquad b_i = 0.07780\frac{RT_{ci}}{P_{ci}}$$

$$\alpha_i = \left[1 + n_i\left(1 - \sqrt{T_{ri}}\right)\right]^2 \qquad n_i = 0.37464 + 1.5422\omega_i - 0.26992\omega_i^2$$

Local/Global Variables

Local Variables

A **local** variable is a variable which is defined in a function only. Indeed, this variable is not in the (base) workspace but in a specific workspace dedicated to the function. As a consequence, these variables cannot be used in other functions.

Global Variables

A **global** variable is a variable that is visible in some specified parts of a program and not restricted to a function only. It is important to note that a **global** variable has to be declared as **global** in the functions where it is used.

Local/Global Variables

Example: Local Variable

```
function [ TheMax ] = MyMaxFunction2(Number1,Number2)

      tmp = -inf;
      if( Number1 > Number2 )
           tmp = Number1;
      else
           tmp = Number2;
      end

      TheMax = tmp;
end
```


Local/Global Variables

Function with Global Variable

```
function [ TheMax ] = MyMaxFunction3(Number1)

     global Number2;
     if( Number1 > Number2 )
         TheMax = Number1;
     else
         TheMax = Number2;
     end
end
```


Local/Global Variables

Example: Global Variable

First, declare the variable Number2 as global:

```
» global Number2;
```

Next, give it a value:

```
» Number2 = 0;
```

Then, call the previous function:

```
» MyMaxFunction3(3)
ans =
     3
» MyMaxFunction3(-1)
ans =
     0
```

Inline Functions

Definition

Inline functions are short functions which require simple implementation. Their implementation is done using the *inline* function whose syntax is as follows:

FunctionName = inline(expr, input)

where *FunctionName* is the name of the inline function, *expr* is a string corresponding the implementation and *input* is the input variable.

Inline Functions: Example

Example

In the following, we create the function $3sin(2x^2)$ called MyInlineFunction.

```
» MyInlineFunction = inline('3*sin(2*x^2)', 'x');

» MyInlineFunction(90)
ans =
     2.7888
```


Exercise 3

Exercise 3

We know that the roots of $ax^2 + bx + c = 0$ are

$$x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$$
$$x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$$

Write a function providing these roots.
What are the solutions if $a = 1, b = 2, c = -3$?
if $a = 1, b = 4, c = -21$?
if $a = 9, b = 6, c = 1$?

Solution

Solution to Exercise 3

```
function [ Roots ] = RootsOfSecondOrderEquation(a, b, c)
      Roots(1) = ( -b + sqrt( b^2 - 4*a*c) )/( 2*a );
      Roots(2) = ( -b - sqrt( b^2 - 4*a*c) )/( 2*a );
end
```

The solutions are:

- $a = 1, b = 2, c = -3 \Longrightarrow x_1 = 1, x_2 = -3$
- $a = 1, b = 4, c = -21 \Longrightarrow x_1 = 3, x_2 = -7$
- $a = 9, b = 6, c = 1 \Longrightarrow x_1 = x_2 = -1/3$

2. **M-files:**

An m-file is a text file where the MATLAB commands are written. When the file is run, MATLAB executes these commands sequentially exactly as they are typed at the command window. M-files are ended with the extension '.m' (e.g. chemical.m).

You create and edit the m-files or script files in one of the following ways:

- Click the **New Script , Creat new document and Open** button on the Home tab.
- Use the command:
  >> edit file_name

  If file_name is unspecified, MATLAB opens a new file called Untitled.
  It is possible to run the commands in m-files in one of the following ways:
- Type the file name in command window and press Enter.
- Click the run button on the editor window.

3. **Input and Output of data:**

- You can use **input** command to enter single element, vector matrix or text as:

```
>> x= input('x= ')
x= 5
x =
    5

>> x= input('x= ')
x= [5 1 7 5]
x =
    5    1    7    5

>> x= input('x= ')
x= [2 7 8 10; 12 4 30 7];
```

```
x =
    2    7    8   10
   12    4   30    7
```

```
>> x= input('x= ')
x= 'test'
x =
test

>> x= input('x= ', 's')
x= test
x =
test
```

- You can use **disp** to display the data stored in x as:
  >> disp(x)

4. **Functions files:**

New functions can be added to MATLAB by using the function files as:

- **function** [out*var1*, out*var2*,...] = *filename* (*invar1*, *invar2*,...)

  where both input and output arguments can be vectors.

Ex: create a new m-file is called : **fun_test** to calculate the $y=x^2+3x-5$ for any given value of x

Solution:

Create a new m-file with the name fun_test and type the following code:

```
function y=fun_test(x) ;
y=x.^2+3*x-5
```

in command window:

```
>> a=1:5;

>> fun_test(a);
```

Ex: Define a function in a file named rect to calculate the area and circumference of the rectangle.

Solution:

Create a new m-file with the name rect and type the following code:

```
function [area, circumference]= rect(a, b)
area=a*b
circumference=(a+b)*2
```

in command window:

```
>> rect(10, 8);

area =

   80

circumference =

   36
```

MATLAB exercises

M-file:

Example:

a) Write a function that performs an ideal gas calculation, where the function is called as follows:

IdealGas(P,V,T,R), and returns the value of n, the number of moles.

```
function n = IdealGas(P,V,T,R)
n = P*V/R/T;
```

b) Apply the function in the command window to calculate the number of moles that exist in a volume of 22.4 L, at a pressure of 1 atm, and at a temperature of 273 K.

```
>> IdealGas(1,22.4,273,0.082) ;
```

c) Write a script file IG_main.m that performs the ideal gas calculation to determine number of moles. The script should ask the user for each of the other four quantities.

```
R = input('Enter the universal gas constant: ');
P = input('Enter the gas pressure: ');
T = input('Enter the gas temperature :');
V = input('Enter the gas volume: ');
n = IdealGas(P,V,T,R);
disp('Moles of ideal gas =')
disp(n)
```

Example:

The quadratic equation provides the roots to the polynomial $ax^2 + bx – c = 0$ according to :

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Write a function called qe(a,b,c), that solves the quadratic equation and returns solutions. Use your function to solve for the roots of the equation $x^2 + 6x – 91 = 0$.

```
function [x1, x2] = qe(a,b,c)
x1 = (-b + sqrt(b.^2 -4.*a.*c))/(2.*a);
x2 = (-b - sqrt(b.^2 -4.*a.*c))/(2.*a);
```

```
>> [x1,x2]=qe(1,6,-91)
```

Example:

Function for calculating enthalpy depending on entering temperature, reference temperature and vector of specific heat equation constants.

```
function dH = deltaH_IG(Ti,Tf,Cp)
dH = Cp(1)*(Tf-Ti)+Cp(2)*(Tf^2-Ti^2)/2+Cp(3)*(Tf^3-Ti^3)/3+Cp(4)*(Tf^4-Ti^4)/4;
```

Cp must be a vector of four values

Example:

Vapor Pressure for a given Temperature ln P = A - B/(T+C)

Input: Antoine Constants (A,B,C), T

Output: Vapor Pressure

A, B, C constants of Antoine equation.

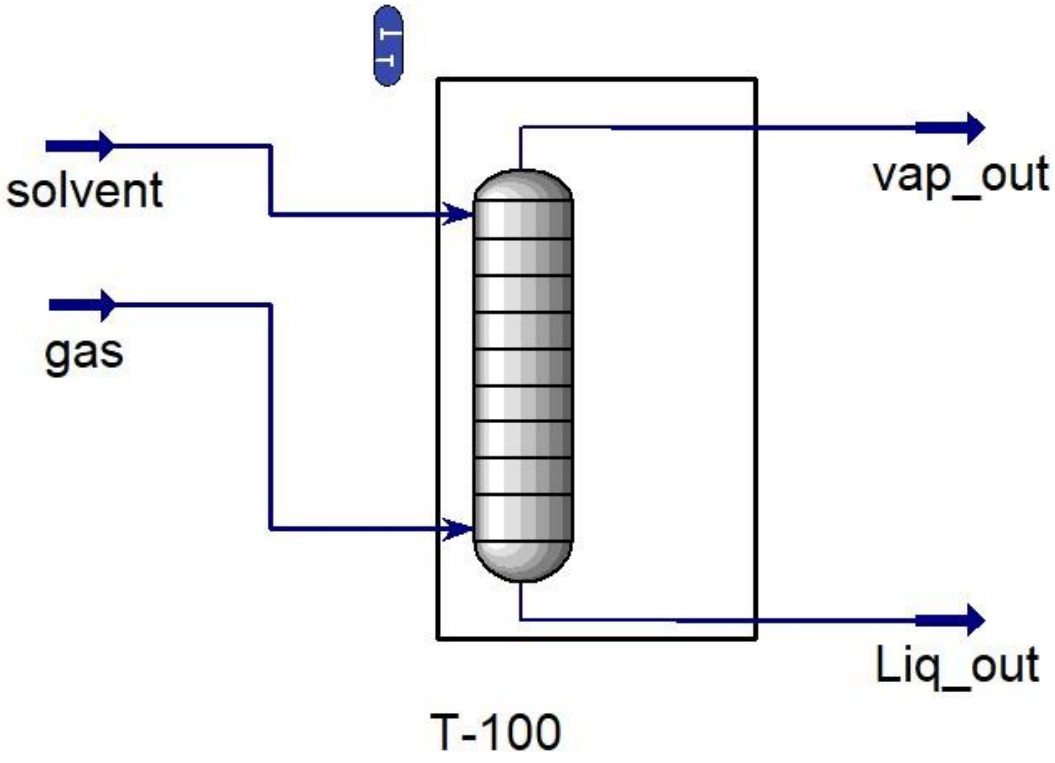
```
function vp = sat_pr_antoine(A, B, C, T)

vp = exp(A - B./(T+C));
```

Example:

Heat capacity of air in J/(mol K), where T is in K. The equation is valid only in the range 273 K < T < 1800 K the file is called as Cp = CpAir(T)

```
function Cp = CpAir(T)
if T < 273
disp('Temperature too low')
elseif T > 1800
disp('Temperature too high')
else
Cp = 28.09 + 0.1965e-2*T + 0.4799e-5*T^2 - 1.965e-9*T^3;

end
```

Modeling of Absorber Using HYSYS V9



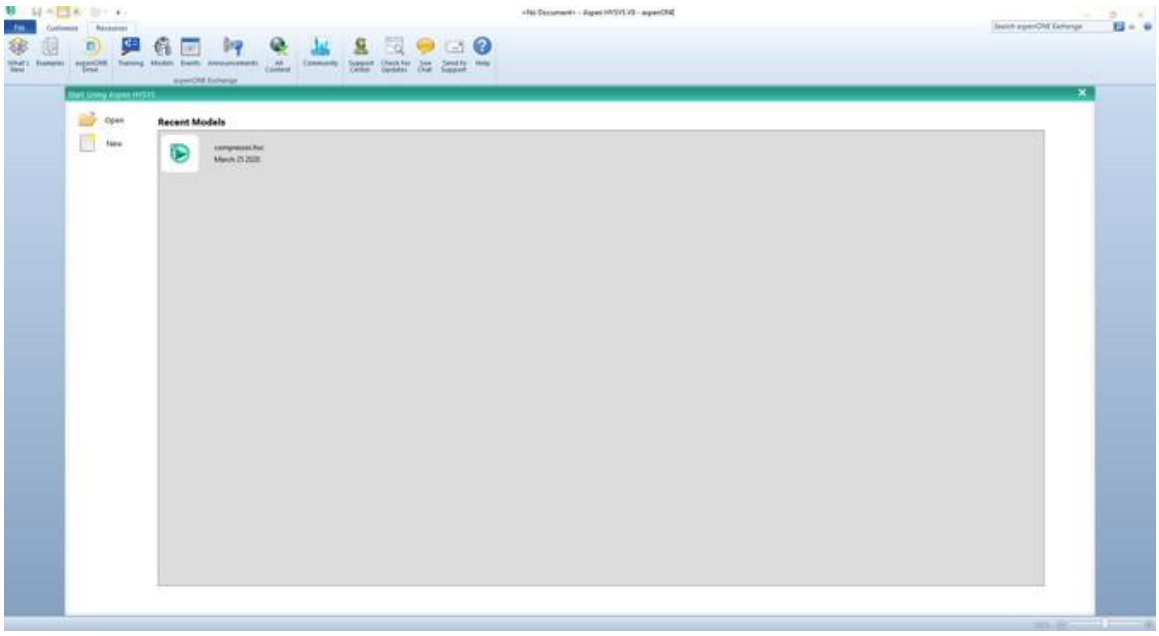
The figure above shows a simple model containing the Absorber with inlet, and outlet streams using Aspen HYSYS V.9.

Example:

CO_2 os absorbed into propylene carbonate in apcked column. The inlet gas stream is 20 mol% Co_2 and 80 mol% methane. The gas stream flows at a rate of 2 m^3/s and the column operates at 60oC and 60.1 atm. The inlet solvent flow is 2000 kmol/h. Use Aspen HYSYS to determine the concentration of CO_2 in the exit gas stream, the column height (m) and the column diameter [1].

Solution:

- Starting HYSYS:
  1. Start menu.
  2. Type in the search field: Aspen HYSYS V9.
  3. Select open.



The above window will pop up once you start HYSYS. It is possible by the Aspen HYSYS start window to open previous cases or create a new window.

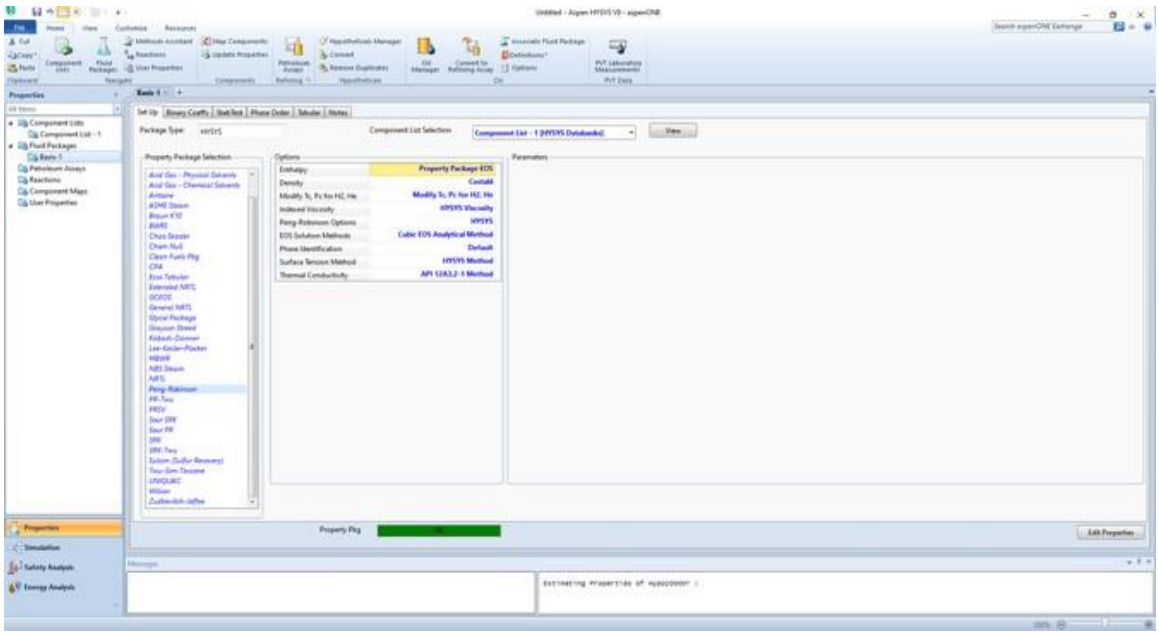
- Create a new case:

1. File menu
2. New \ case

- There are two main necessary information have to be given in the properties tab before starting the simulation:
    1. Property package : Sour PR
    2. Components: C1, CO_2, Propylene Carbonate

1. Properties
2. Component lists \ Add
3. Select: Pure components
4. Type in the serach field C1
5. Select C1 \ add
6. Repeat the steps (4,5) for CO_2 and Propylene Carbonate

- Add fluid package:
    1. Select Fluid package
    2. Click Add
    3. Select (Sour PR)



- Create new material stream:
    1. Select simulation.

2. Object palette.
3. Select the material stream icon.

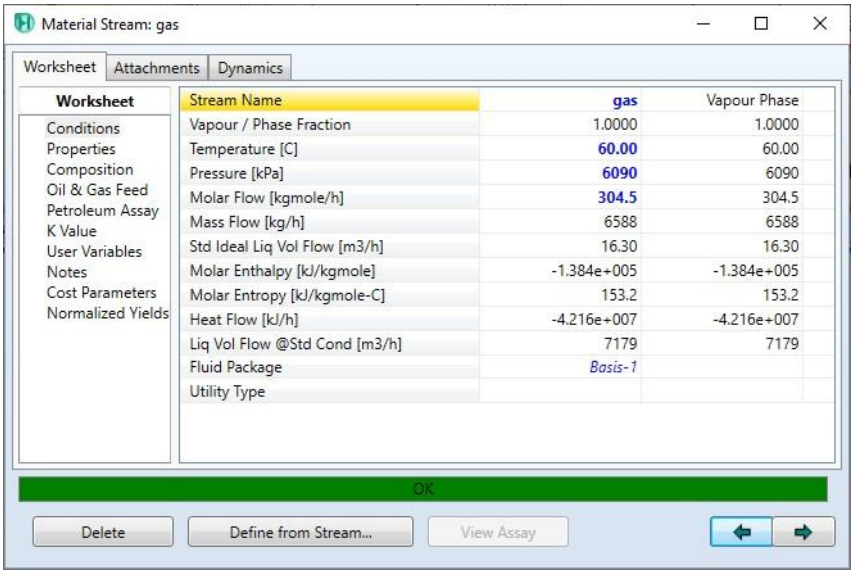
Alternatively, you can add new material stream by pressing F11.

4. Double click on the added material stream
5. Give the below information:

Name	Solvent
Conditions	
Temperature	60^{O} c
PRESSURE	60.1 atm
Molar flow	2000 kmol/h
Composition	
Propylene Carbonate	1

6. Repeat step 5 to add another stream and give the following information:

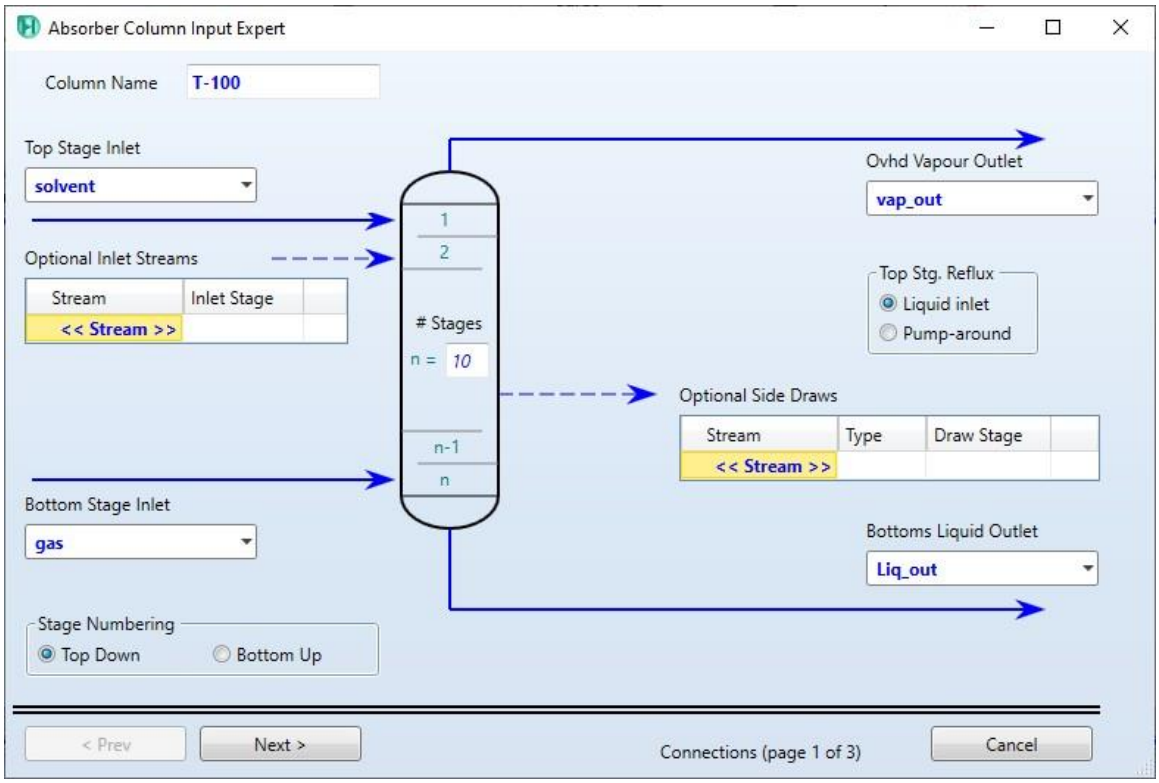
Name	Gas
Conditions	
Temperature	60^{O} c
PRESSURE	60.1 atm
Molar flow	7200 m^3/h
Composition	
C1	0.8
CO_2	0.2



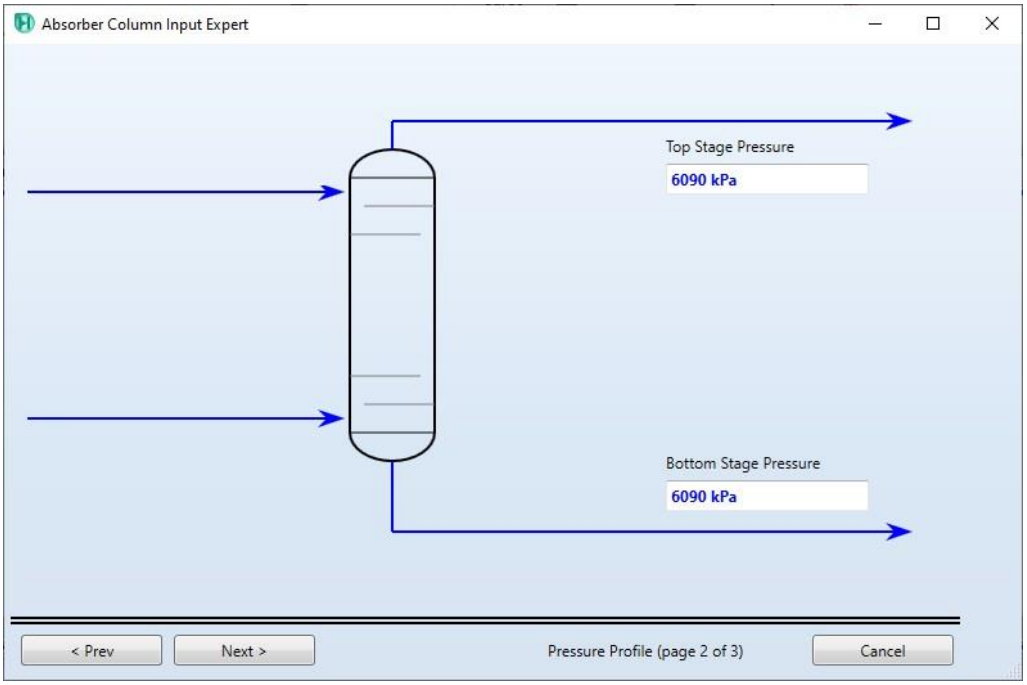
- Adding Absorber:
    1. Object palette.
    2. Select column\absorber
    3. Place the selected unit in the desired position.

You can add a new unit by pressing F12 and select the desired unit.

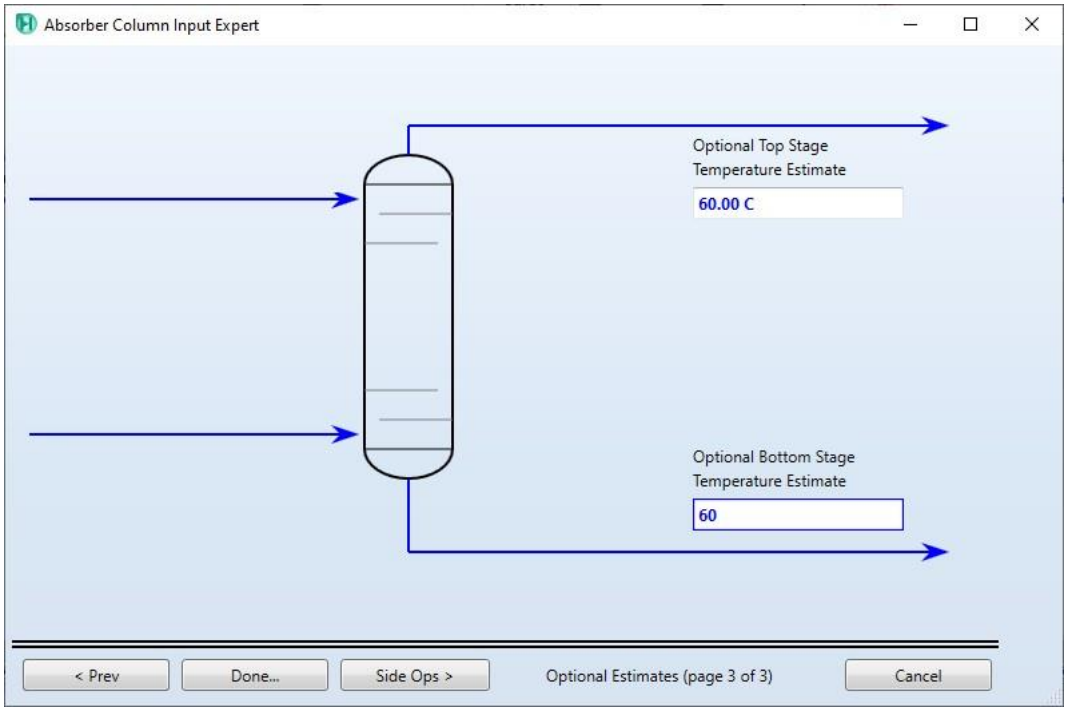
4. Double click on the absorber
5. In the Top Stage inlet, select Solvent
6. In the Bottom Stage inlet, select Gas
7. In the Ovhd Vapour outlet field, type (vap_out). [a new material stream will be created and connected to the compressor]
8. In Bottom Liquid Outlet, type (Liq_out) [a new energy stream will be created and connected to the compressor].



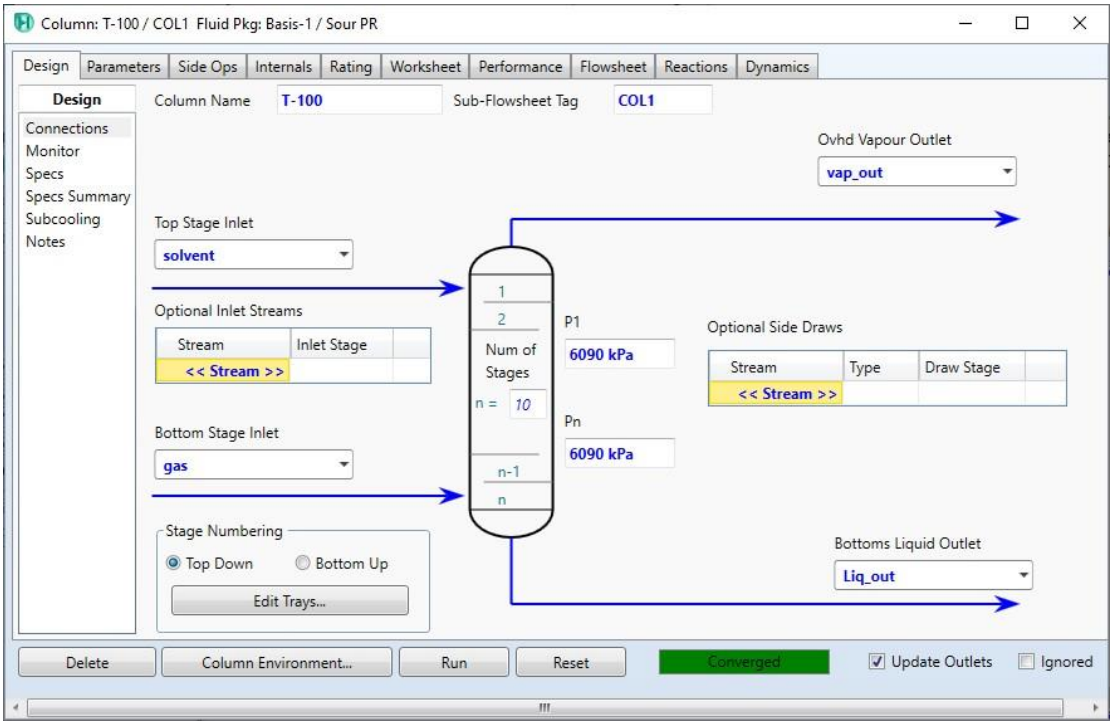
- Click (next)
    9. Set the top stage pressure to 60.1 atm.
    10. Set the Bottom stage pressure to 60.1 atm.



- Click (next)
    11. Set the temperature in the top stage pressure to 60.1 atm.
    12. Set the temperature in the top stage pressure to 60.1 atm.
    13. Click (Done).

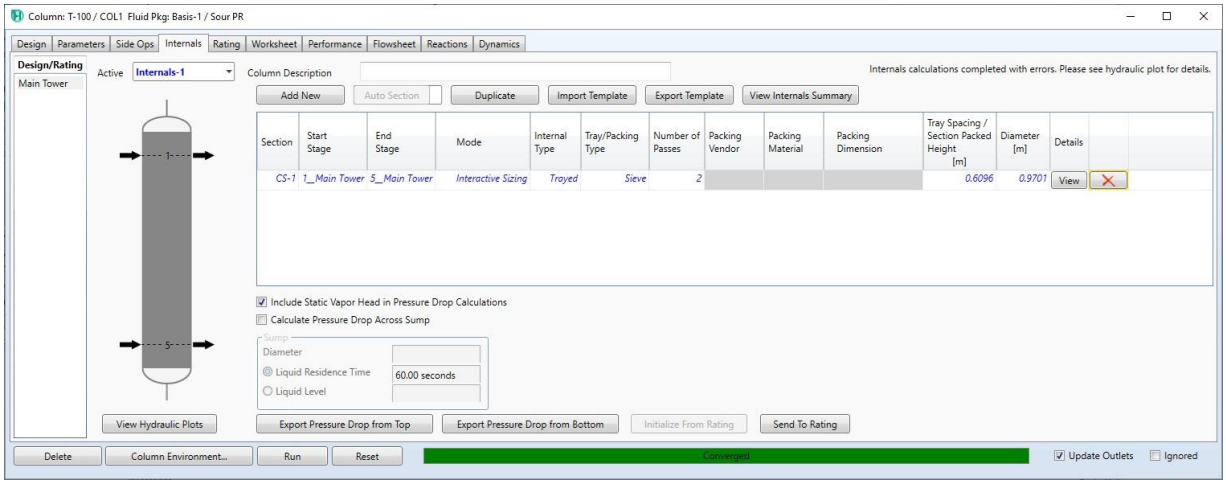


- Convergence:
    1. Design \ connections.
    2. Click (Run).

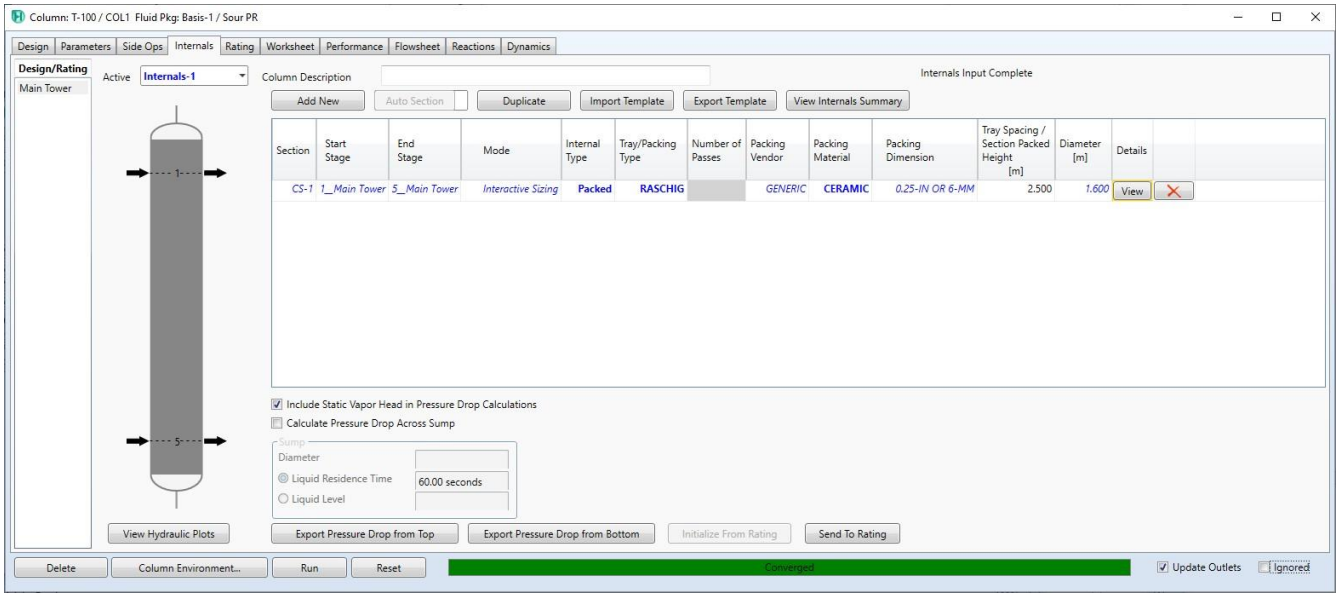


- Sections:

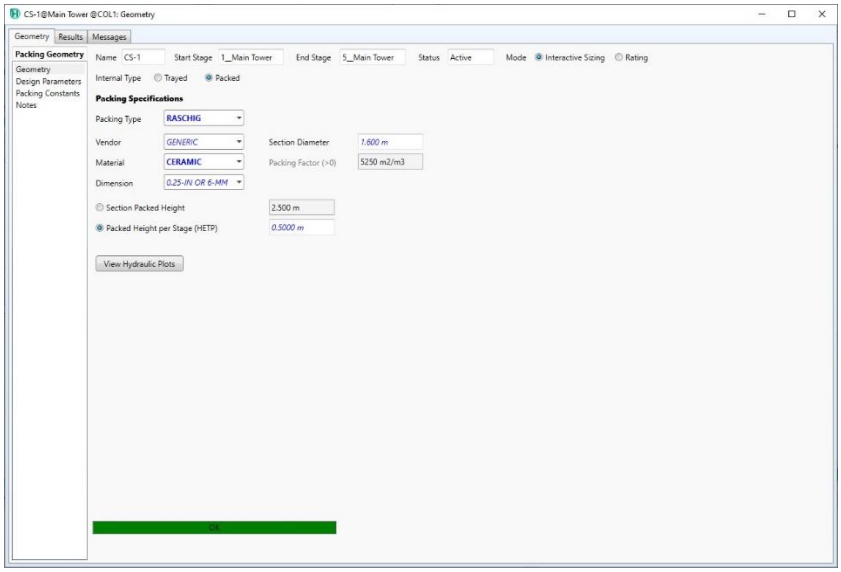
1. Internals
2. Click Auto Sections



3. In Internal type field, select packed
4. In the Tray Packing type field, select Raschig
5. In the packing vendor type, select Generic
6. In the packing material type, select Ceramic
7. In the packing dimension, select (0.25-IN OR 6-MM)



8. Click View
9. Select Packed height per stage
10. Set the height to (0.5 m)
11. Click (Reset)
12. Click (Run)



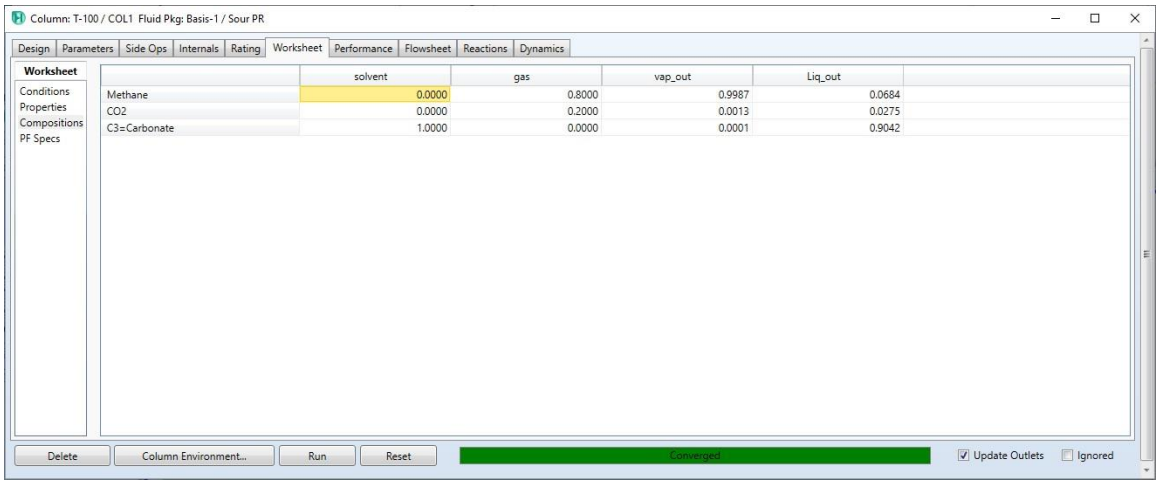
- Convergence:

1. Click Reset
2. Click Run

- Read the column height and diameter.

- Compositions:

1. Worksheet
2. Compositions
3. Read Co_2 compositions



- Save the case:

1. File menu
2. Select save as
3. Give the name (compressor)

References:

1. Hamid, M. K. A. "Hysys: An introduction to chemical engineering simulation." *Apostila de Hamid* (2007).

Separation Columns

Recovery of natural-gas liquids (NGL) from natural gas is quite common in natural gas processing. Recovery is usually done to:

- Produce transportable gas (free from heavier hydrocarbons which may condense in the pipeline).
- Meet a sales gas specification.
- Maximize liquid recovery (when liquid products are more valuable than gas).

HYSYS can model a wide range of different column configurations. In this simulation, an NGL Plant will be constructed, consisting of three columns:

- De-Methanizer (operated and modelled as a Reboiled Absorber column)
- De-Ethanizer (Distillation column)
- De-Propanizer (Distillation column)

Learning Outcomes: At the end of this chapter, the user will be able to:

- Add columns using the Input Experts.
- Add extra specifications to columns.

Prerequisites: Before beginning this chapter, the users need to know how to:

- Navigate the PFD
- Add Streams in the PFD or the Workbook
- Add and connect Unit Operations

12.10 Process Overview

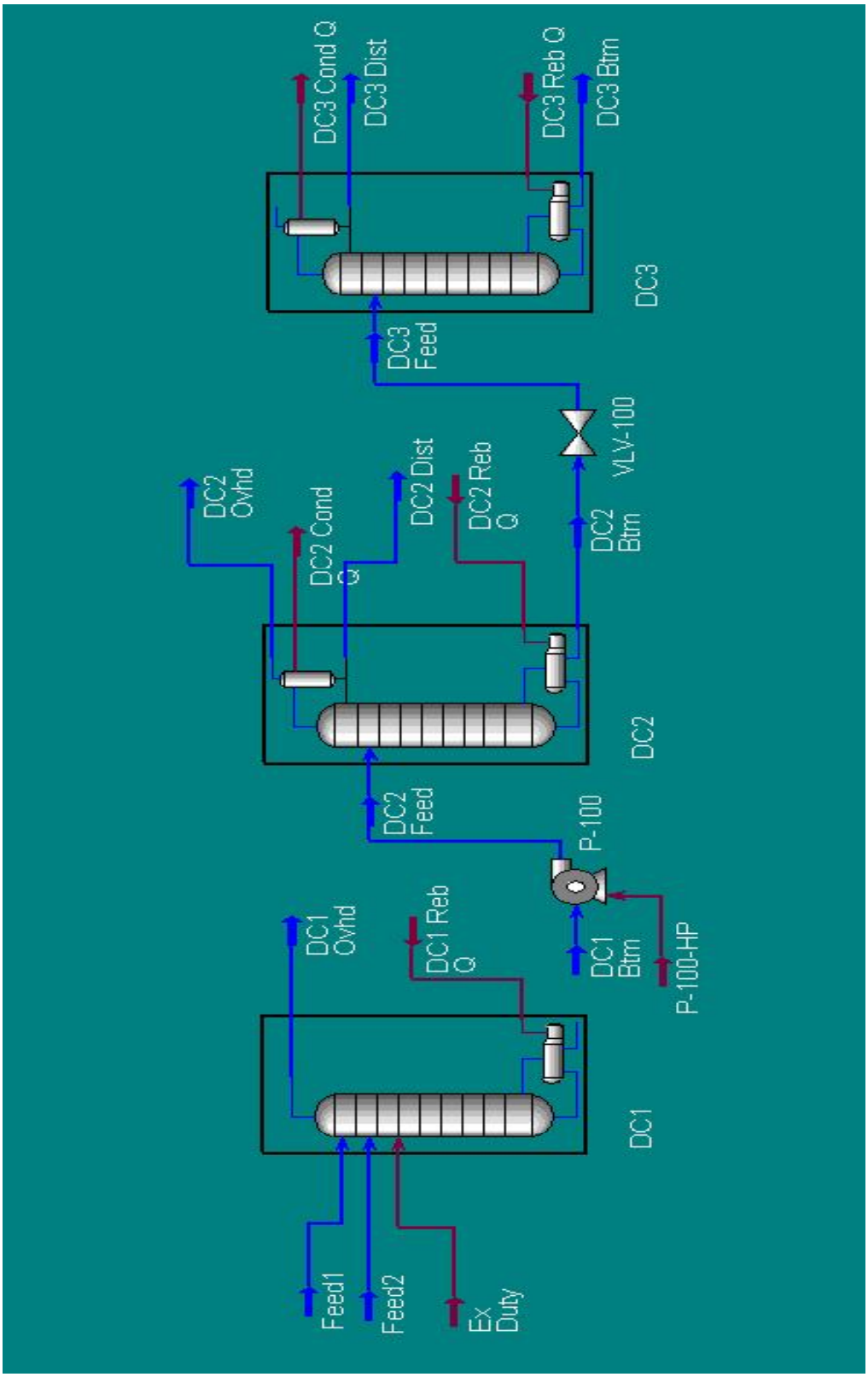


Figure 12-1

12.10 Column Overviews

DC1: De-Methanizer

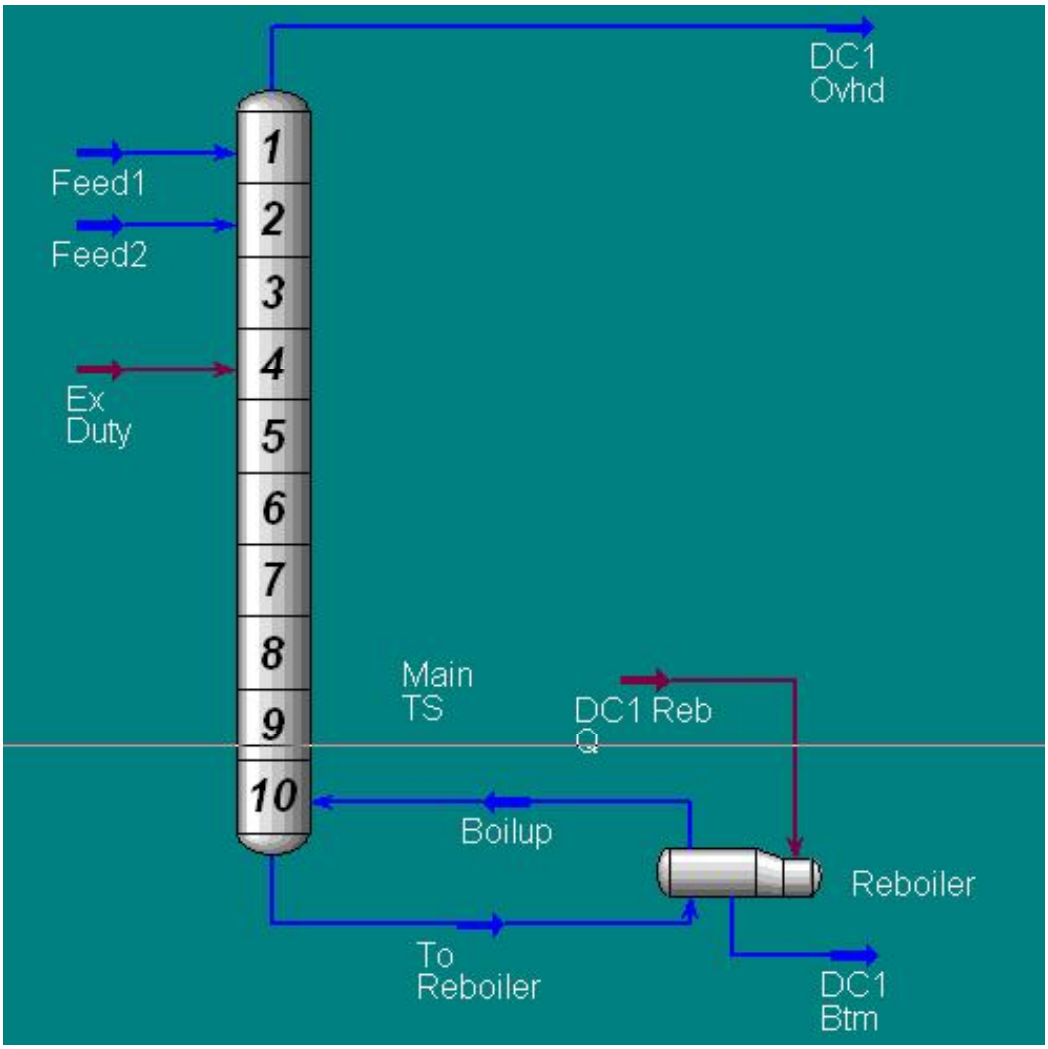


Figure 12-2

DC2: De-Ethanizer

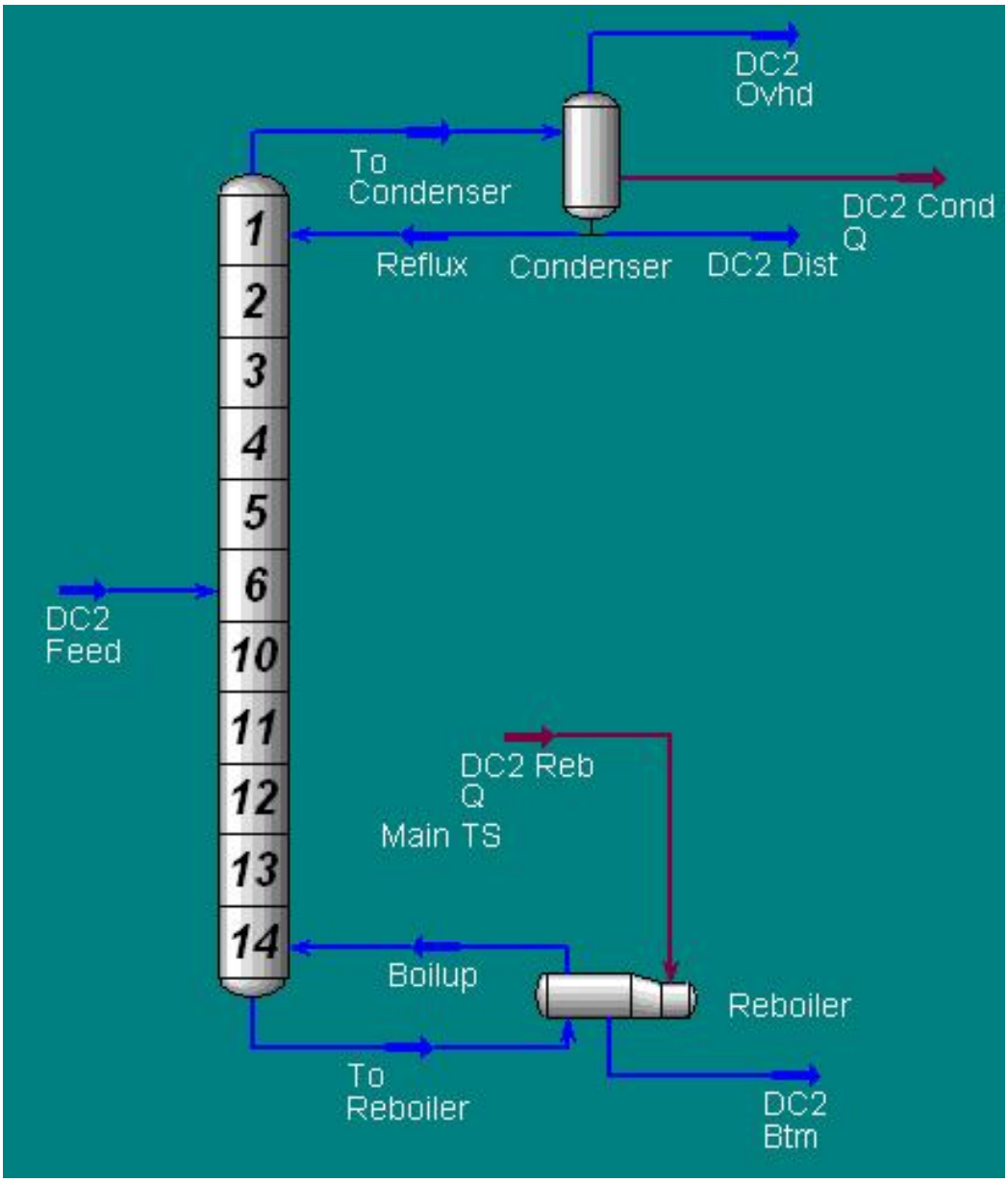


Figure 12-3

Hamid, M. K. A. "Hysys: An introduction to chemical engineering simulation." Apostila de Hamid (2007).

DC3: De-Propanizer

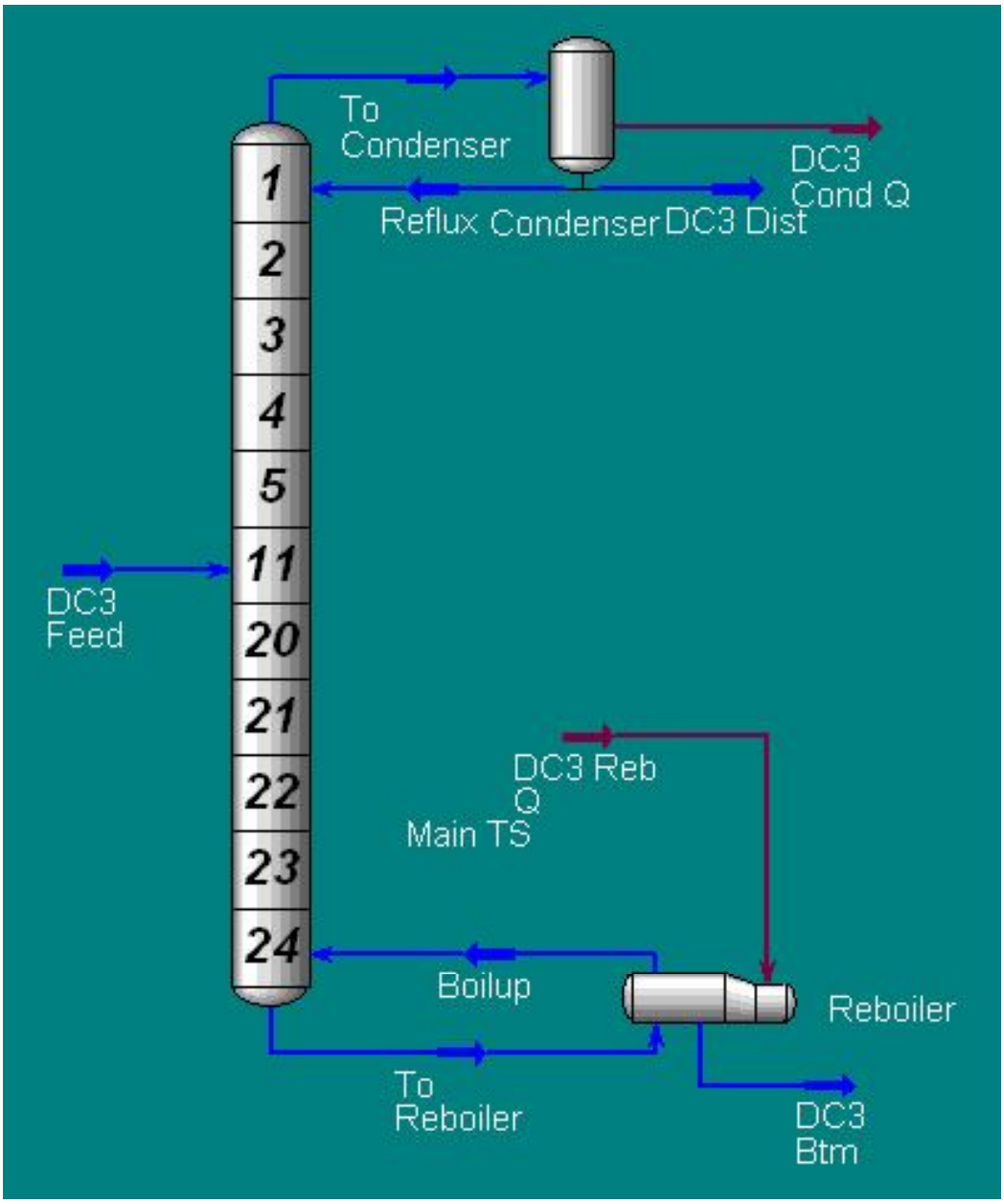


Figure 12-4

12.10 Defining the Simulation Basis

1. Start a new case.
2. Select the **Peng Robinson** EOS.
3. Add the components: N_2, CO_2, C_1 - C_8.
4. Enter the **Simulation Environment**.

12.4 Adding the Feed Streams

1. Add a **Material Stream** with the following data:

In this cell…	Enter…
Name	Feed1
Temperature	-95°C (-140°F)
Pressure	2275 kPa (330 psia)
Flowrate	1620 kgmole/h (3575 lbmole/hr)
Component	Mole Fraction
N_2	0.0025
CO_2	0.0048
C_1	0.7041
C_2	0.1921
C_3	0.0706
i-C_4	0.0112
n-C_4	0.0085
i-C_5	0.0036
n-C_5	0.0020
C_6	0.0003
C_7	0.0002
C_8	0.0001

2. Add another **Material Stream** with the following data:

In this cell…	Enter…
Name	Feed2
Temperature	-85°C (-120°F)
Pressure	2290 kPa (332 psia)
Flowrate	215 kgmole/h (475 lbmole/hr)
Component	Mole Fraction
N_2	0.0057
CO_2	0.0029
C_1	0.7227
C_2	0.1176
C_3	0.0750
i-C_4	0.0204
n-C_4	0.0197
i-C_5	0.0147
n-C_5	0.0102
C_6	0.0037
C_7	0.0047
C_8	0.0027

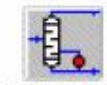
12.5 Adding De-Methanizer

The De-Methanizer is modelled as a reboiled absorber operation, with two feed streams and an energy stream feed, which represents a side heater on the column.

1. Add an **Energy** stream with the following values:

In this cell…	Enter…
Name	Ex Duty
Energy Flow	2.1e+06 kJ/h (2.0e+06 Btu/hr)

2. Double-click on the **Reboiled Absorber** icon on the Object Palette. The first Input Expert view appears.



Reboiled Absorber Column icon

3. Complete the view as shown below:

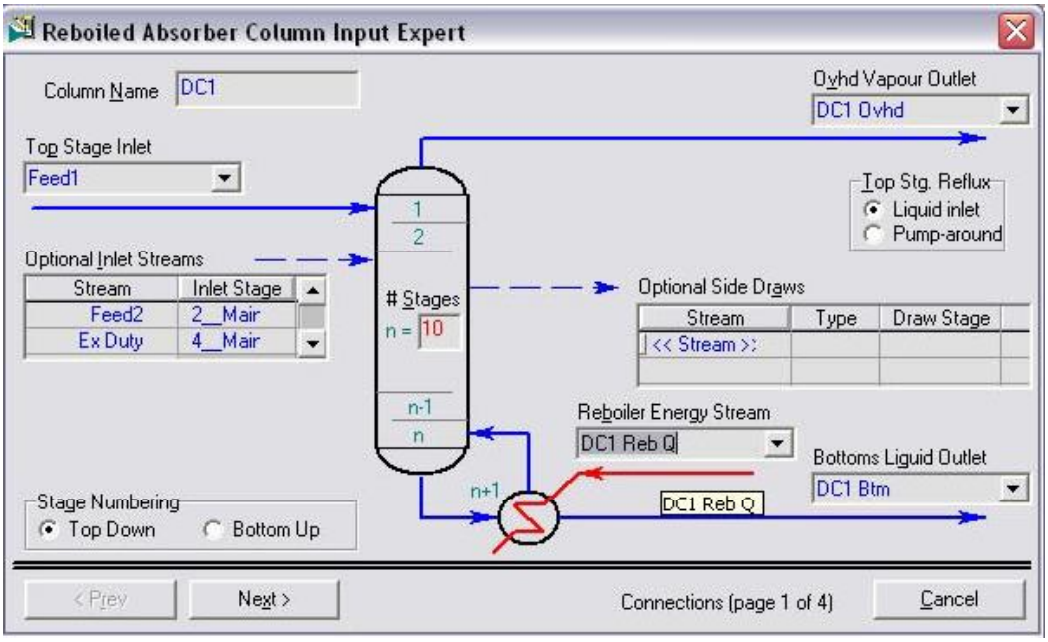


Figure 12-5

4. Click the **Next** button to proceed to the next page.
5. Supply the following information to the Pressure Estimates page. If you are using field units, the values will be **330 psia** and **335 psia**, for the Top Stage Pressure and Reboiler Pressure, respectively.

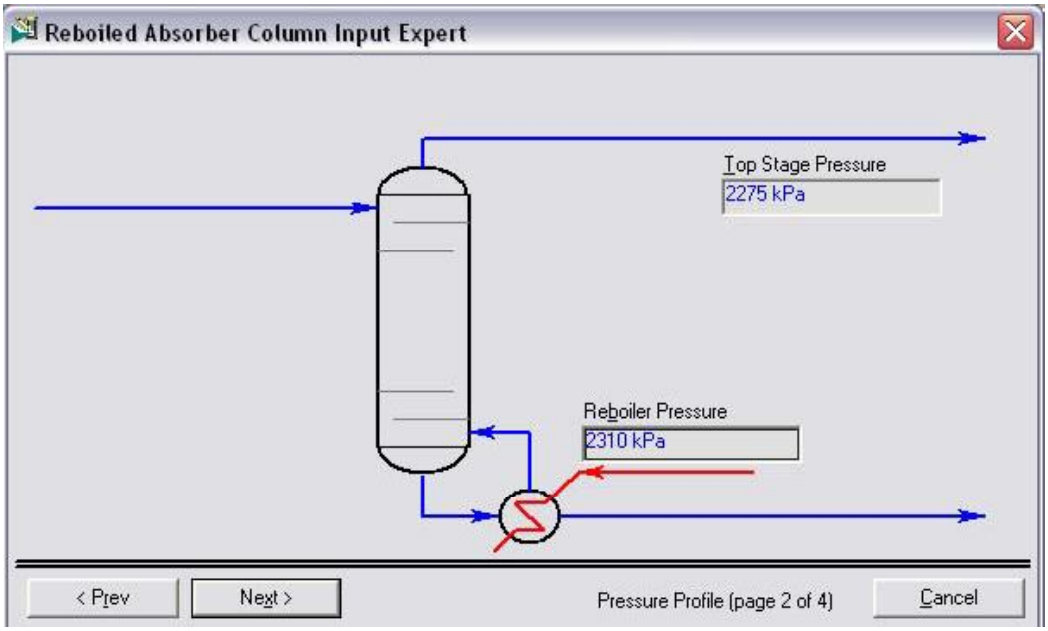


Figure 12-6

6. Click the **Next** button to proceed to the next page.
7. Enter the temperature estimates shown below. In field units, the top stage temperature estimate will be **-125°F**, and the reboiler temperature estimate will be **80°F**.

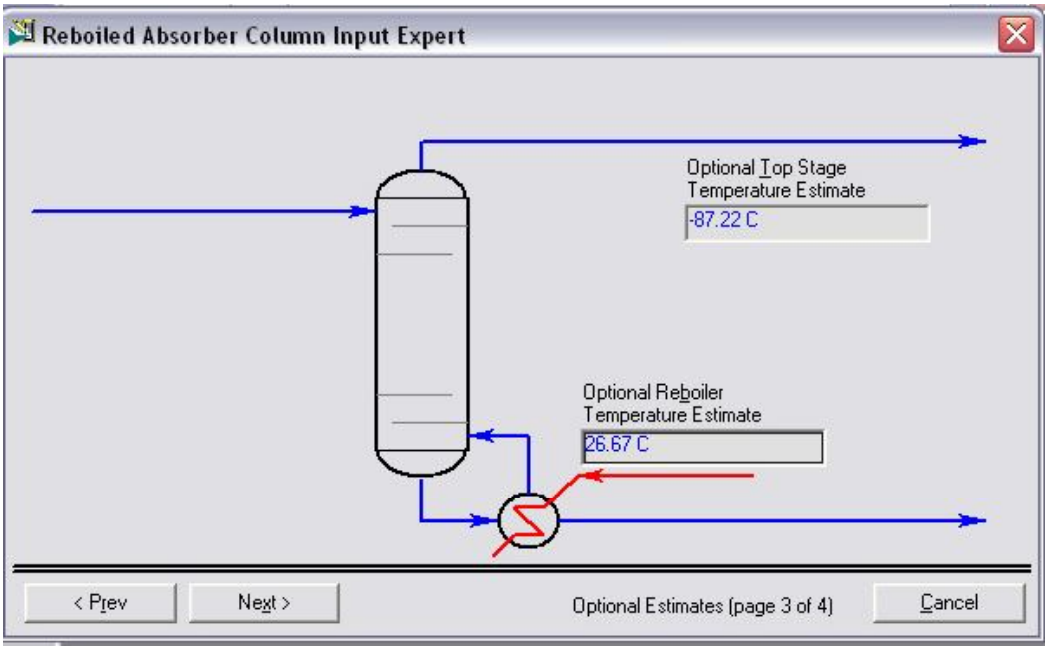


Figure 12-7

8. Click the **Next** button to continue.
9. For this case, no information is supplied on the last page of the Input Expert, so click the **Done** button.

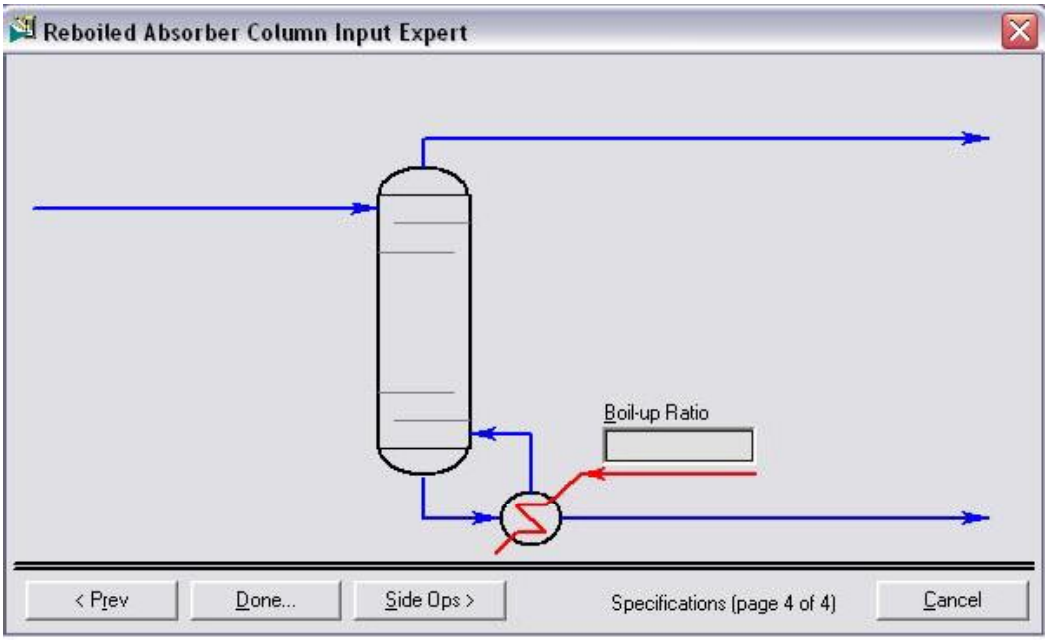


Figure 12-8

When you click the **Done** button, HYSYS will open the Column property view. Access the **Monitor** page on the **Design** tab.

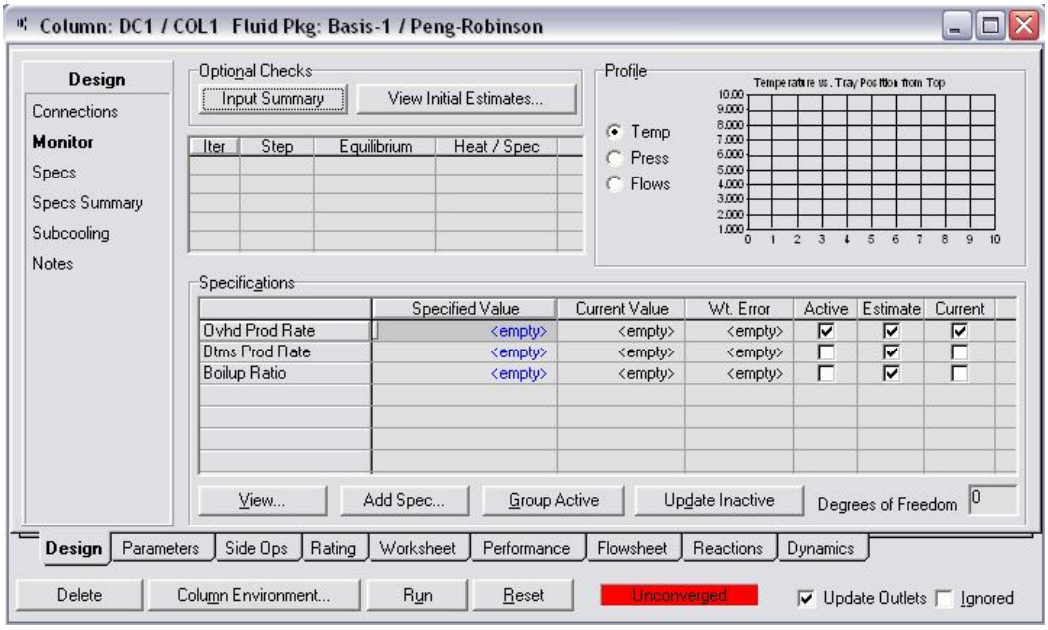


Figure 12-9

Before you converge the column, make sure that the specifications are as shown above. You will have to enter the value for the Ovhd Prod Rate specification. The specified value is **1338 kgmole/h** (2950 lbmole/hr). Once this value is entered, the column will start running and should converge.

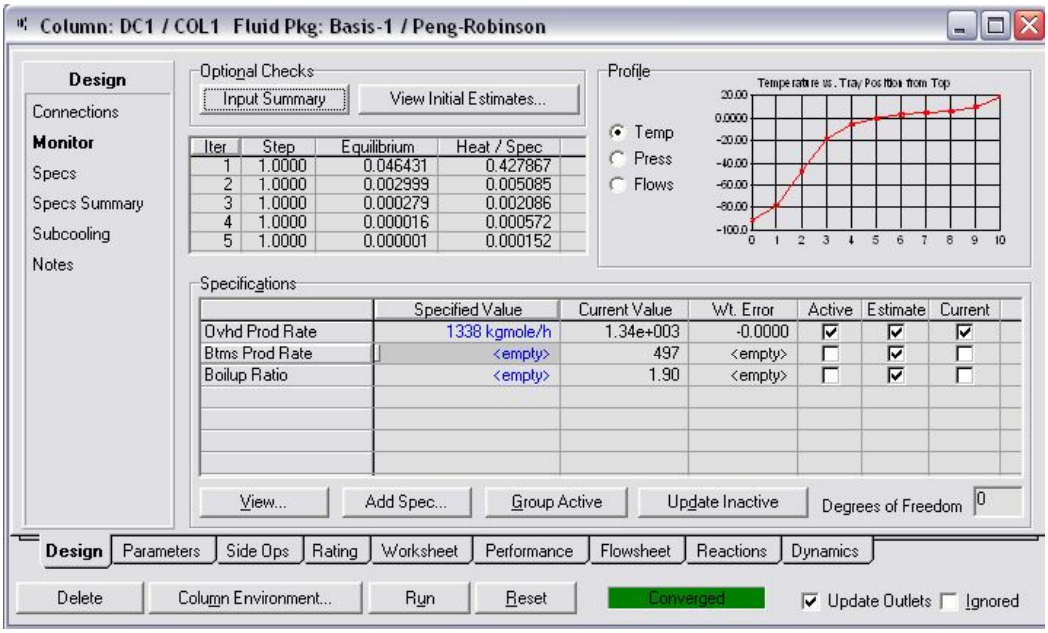


Figure 12-10

What is the mole fraction of Methane in DC1 Ovhd? _____

Although the column is converged, it is not always practical to have flow rate specifications. These specifications can result in columns which cannot be converged or that produce product streams with undesirable properties if the column feed conditions change.

An alternative approach is to specify either component fractions or component recoveries for the column product streams.

1. Go to the **Specs** page on the **Design** tab of the Column property view.

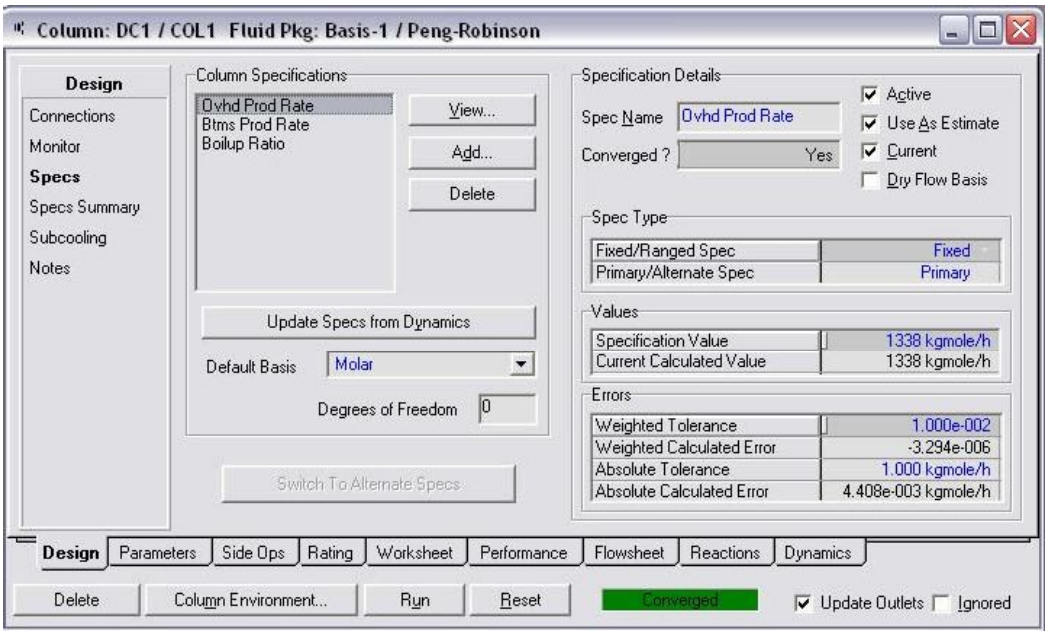


Figure 12-11

2. Click the **Add** button in the Column Specifications group to create a new specification.
3. Select **Column Component Fraction** from the list that appears.

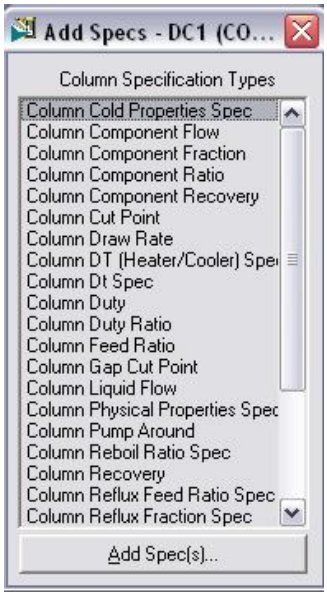


Figure 12-12

4. Click the **Add Spec(s)** button.
5. Complete the spec as shown in the following figure.

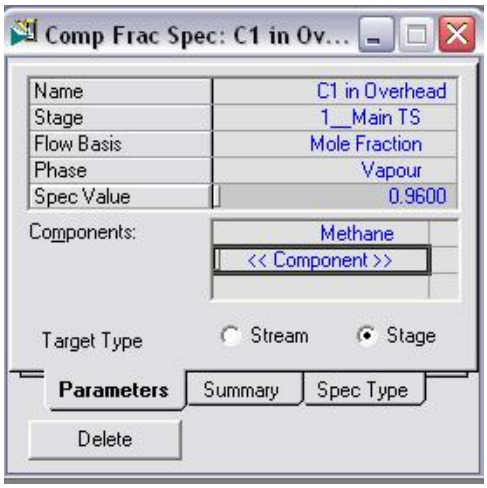


Figure 12-13

6. When you are done, close the view.

The Monitor page of the Column property view shows 0 Degrees of Freedom even though you have just added another specification. This is due to the fact that the specification was added as an estimate, not as an active specification.

7. Go to the **Monitor** page. Deactivate the **Ovhd Prod Rate** as an active specification and activate the **Comp Fraction** specification which you created.

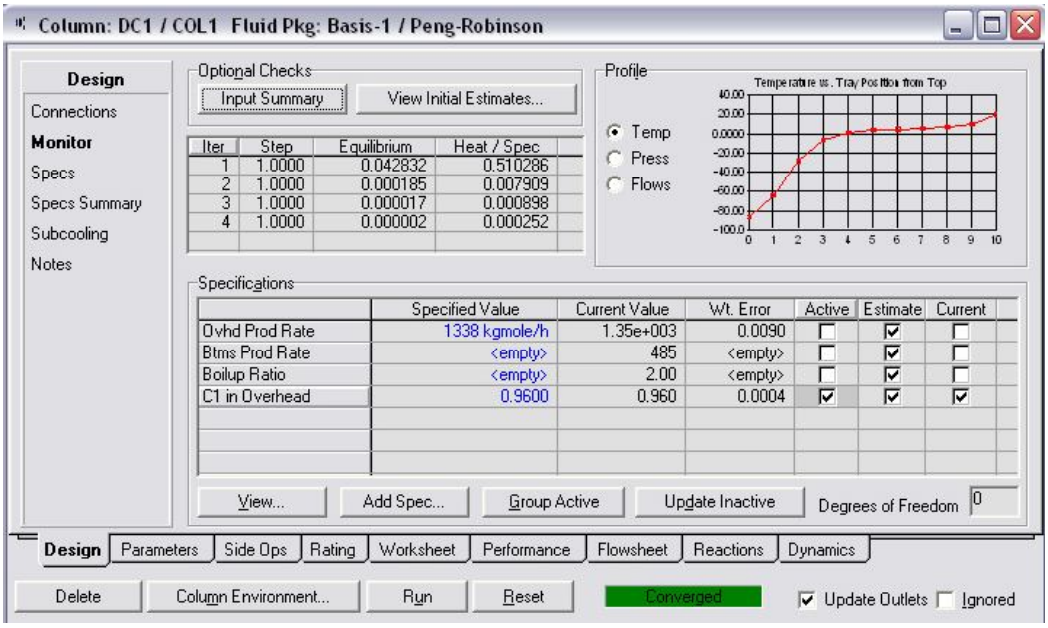


Figure 12-14

What is the flowrate of the overhead product, DC1 Ovhd? _____

Once the column has converged, you can view the results on the **Performance** tab.

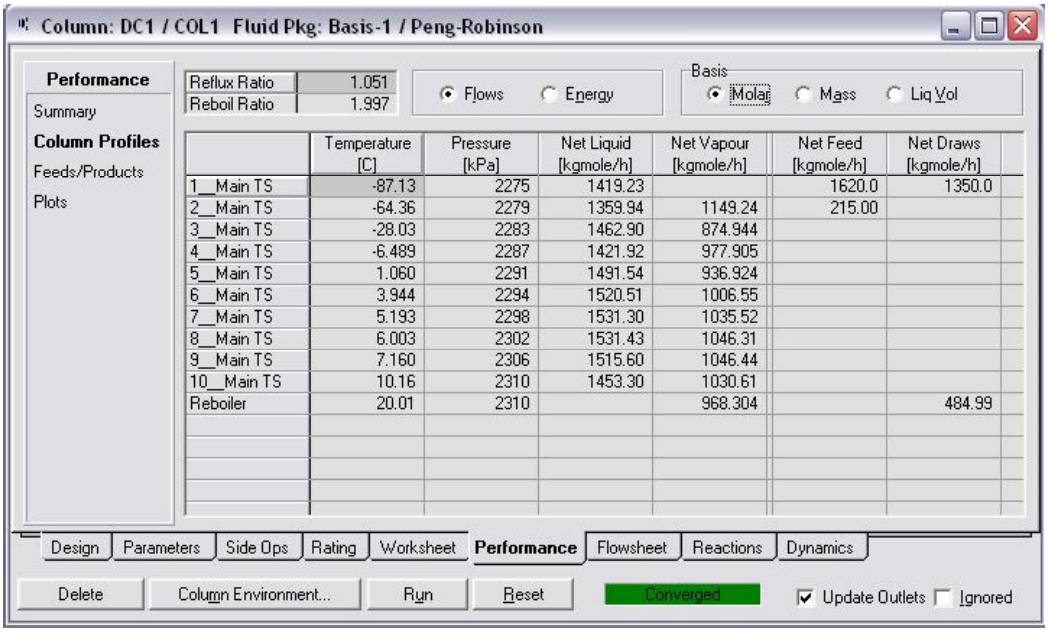


Figure 12-15

12.6 Adding a Pump

The pump is used to move the De-Methanizer bottom product to the De-Ethanizer.

Install a pump and enter the following information:

In this cell…	Enter…
Connections	
Inlet	DC1 Btm
Outlet	DC2 Feed
Energy	P-100-HP
Worksheet	
DC2 Feed Pressure	2790 kPa (405 psia)

12.7 De-Ethanizer

The De-Ethanizer column is modeled as a distillation column, with 16 stages, 14 trays in the column, plus the reboiler and condenser. It operates at a pressure of 2760 kPa (400 psia). The objective of this column is to produce a bottom product that has a ratio of ethane to propane of 0.01.

1. Double-click on the **Distillation Column** button on the Object Palette and enter the following information.

Distillation Column icon

In this cell…	Enter…
Connections	
Name	DC2
No. of Stages	14
Feed Stream/Stage	DC2 Feed/6
Condenser Type	Partial
Overhead Vapour Product	DC2 Ovhd
Overhead Liquid Product	DC2 Dist
Bottom Product	DC2 Btm
Reboiler Duty	DC2 Reb Q
Condenser Duty	DC2 Cond Q
Pressures	
Condenser	2725 kPa (395 psia)
Condenser Delta P	35 kPa (5 psi)
Reboiler	2792 kPa (405 psia)
Temperature Estimates	
Condenser	-4°C (25°F)
Reboiler	95°C (200°F)
Specifications	
Overhead Vapour Rate	320 kgmole/h (700 lbmole/hr)
Distillate Rate	0 kgmole/h
Reflux Ratio	2.5 (Molar)

2. Click the **Run** button to run the column.

What is the flowrate of C_2 and C_3 in DC2 Btms?

*C_2*_____, *C_3*_____, *Ratio of C_2/C_3*_____

3. On the **Specs** page, click the **Add** button to create a new specification.
4. Select **Column Component Ratio** as the specification type and provide the following information:

In this cell…	Enter…
Name	C2/C3
Stage	Reboiler
Flow Basis	Mole Fraction
Phase	Liquid
Spec Value	0.01
Numerator	Ethane
Denominator	Propane

5. On the **Monitor** tab, deactivate the **Ovhd Vap Rate** specification and activate the **C_2/C_3** specification which you created.

What is the flowrate of DC2 Ovhd? _____

12.8 Adding a Valve

A valve is required to reduce the pressure of the stream DC2 Btm before it enters the final column, the De-Propanizer.

Add a Valve operation and provide the following information:

In this cell…	Enter…
Connections	
Feed Stream	DC2 Btm
Product Stream	DC3 Feed
Worksheet	
DC3 Feed Pressure	1690 kPa (245 psia)

12.9 De-Propanizer

The De-Propanizer column is represented by a distillation column consisting of 25 stages, 24 trays in the column plus the reboiler. (Note that a total condenser does not count as a stage). It operates at 1620 kPa (235 psia). There are two process objectives for this column. One is to produce an overhead product that contains no more than 1.50 mole percent of i-C_4 and n-C_4 and the second is that the concentration of propane in the bottom product should be less than 2.0 mole percent.

1. Add a distillation column and provide the following information:

In this cell…	Enter…
Connections	
Name	DC3
No. of Stages	24
Feed Stream/Stage	DC3 Feed/11
Condenser Type	Total
Overhead Liquid Product	DC3 Dist
Bottom Product	DC3 Btm
Reboiler Duty	DC3 Reb Q
Condenser Duty	DC3 Cond Q
Pressures	
Condenser	1585 kPa (230 psia)
Condenser Delta P	35 kPa (5 psi)
Reboiler	1655 kPa 240 psia)
Temperature Estimates	
Condenser	38°C (100°F)
Reboiler	120°C (250°F)
Specifications	
Distillate Rate	100 kgmole/h (240 lbmole/hr)
Reflux Ratio	1.0 (Molar)

2. Run the column.

What is the mole fraction of C_3 in the overhead and bottoms products?

_____ *and* _____

3. Create two new Component Fraction specifications for the column.

In this cell…	Enter…
i-C4 and n-C4 in Distillate	
Name	iC4 and nC4
Stage	Condenser
Flow Basis	Mole Fraction
Phase	Liquid
Spec Value	0.015
Components	i-C4 and n-C4
C3 in Reboiler Liquid	
Name	C3
Stage	Reboiler
Flow Basis	Mole Fraction
Phase	Liquid
Spec Value	0.02
Component	C3

4. Deactivate the **Distillate Rate** and **Reflux Ratio** specifications.
5. Activate the **iC_4, and nC_4**, and **C_3** specifications which you created.

12.10 Save Your Case

1. Go to the **File** menu.
2. Select **Save As**.
3. Give the HYSYS file the name **Separation Columns** then press the OK button.

MATLAB basics

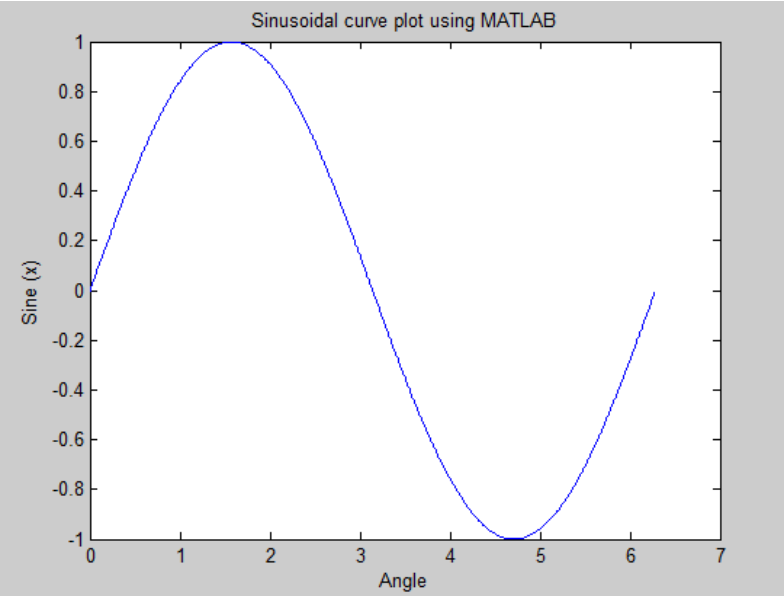
1.3 Plotting Functions

- *Plot(x,y):* draws a graph of y versus x.
- The labels can be added to the x-axes and y-axes:
  >> xlabel ('vertical')
  >> ylabel ('horizontal')
- Titles can be added to the plot as:
  >> title ('2D Plotting')

Ex: Draw a sinusoidal curve for the angle values ranging between 0 and 2π and add the title of "Sinusoidal curve plot using MATLAB" with a label of (sin(x)) on the y-axis and (Angle) on the x-axis.

Sol:

```
>> x=0:0.01:2*pi;
>> y=sin(x);
>> plot (x,y);
>> title('Sinusoidal curve plot using MATLAB');
>> xlabel ('Angle');
>>  ylabel ('Sine (x)');
```



The text and notes for certain points on the curve or inside the figure are added at the given point as:

```
>> text(1, -0.5, '{Sinewave Example}')
```

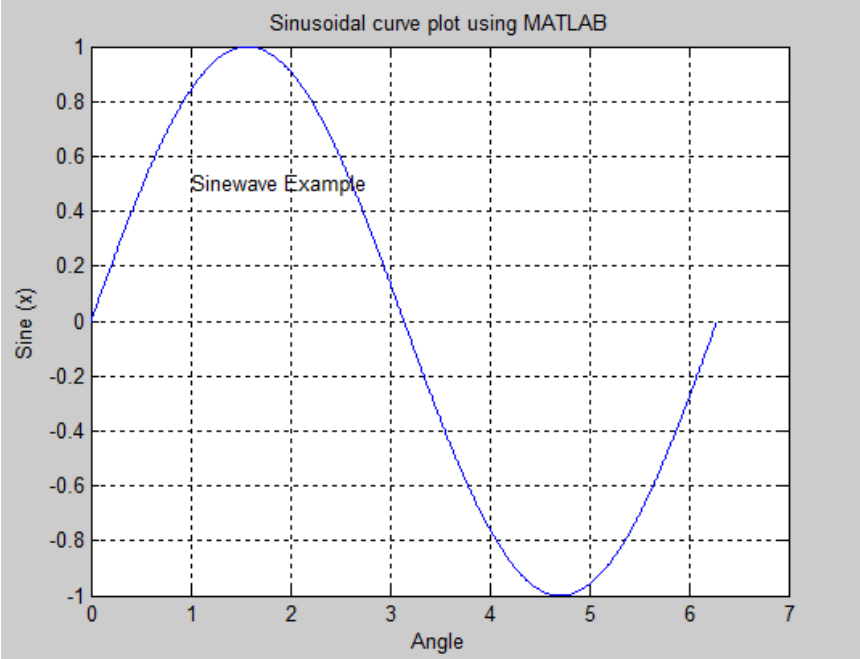
The sentence of “Sinewave example” will be located at (1,-0.5).

The axis limits of the graph can be optimized according to the expected maximum and minimum limits as:

```
>> axis([xmin xmax ymin ymax])
>> axis([0 7 -1 1])
```

The grid of the figure is turned on and off as:

```
>> grid off
>> grid on
```



More than one x-y vectors pairs giving multiple graphs can be combined in a single graph as:

```
>> plot(x,y1,x,y2,x,y3)
```

The legends are placed in order to identify each graphs as:

```
>> legend('y1', 'y2',' y3')
```

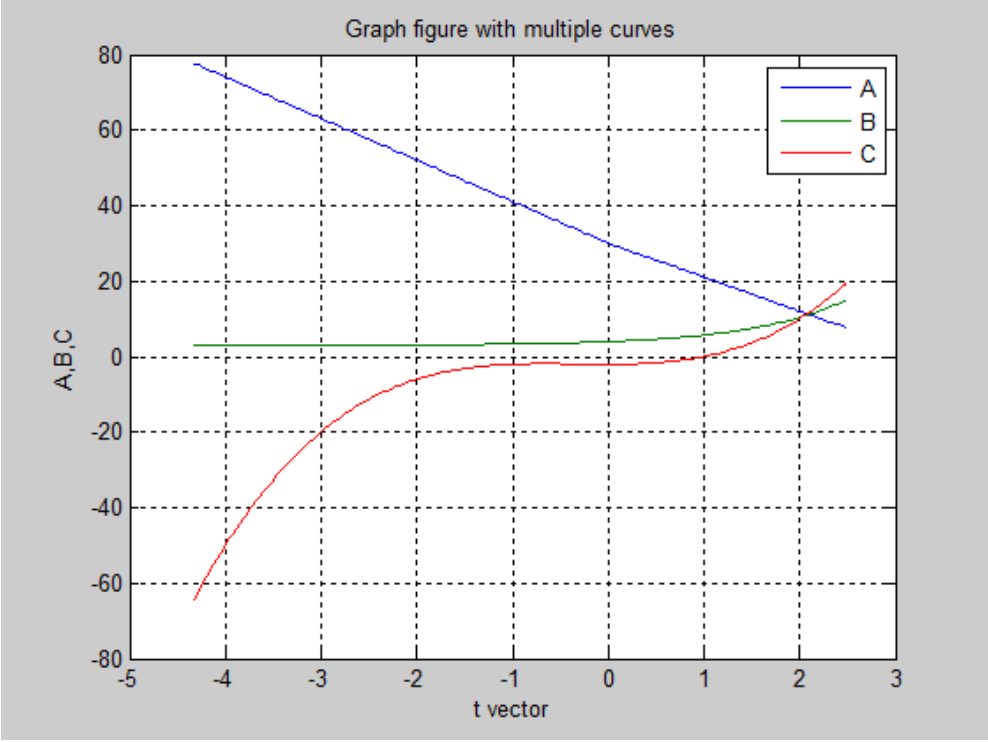
Ex: Draw the following functions for 200 points of t ranging between -5cos(30°) and 5cos(60°) :

1. $A = |t| - 10t + 30$
2. $B = e^t + \ln(20)$
3. $C = t^3 + t^2 - 2$

Put the label (A,B,C) on the y-axis and (t vector) on the x-axis and a title of (figure with multiple curves). Add a legend to the figure to recognize them.

Sol:

```
>> t=linspace (-5*cos(30/180*pi), 5*cos(60/180*pi), 200);
>> A=abs(t)-10*t+30;
>> B=exp(t)+log(20);
>> C=t.^3+t.^2-2;
>> plot(t,A,t,B,t,C);
>> grid on;
>> xlabel('t vector');
>> ylabel('A,B,C');
>> title ('Graph figure with multiple curves');
>> legend ('A', 'B', 'C');
```



The line color, line style and the marker type are customized by:
>> plot(x,y,'color style marker')

Giving theses details are according to the following table

Colors		Style		Marker type	
C	cyan	-	Solid	+	
M	magenta	--	Dashed	o	
Y	yellow	:	Dotted	*	
R	red	-.	Dash-dot	x	
G	green			s	square
B	blue			d	Diamond
W	white			^	Up triangle
K	black			v	Down triangle
				>	Right triangle
				<	Left triangle

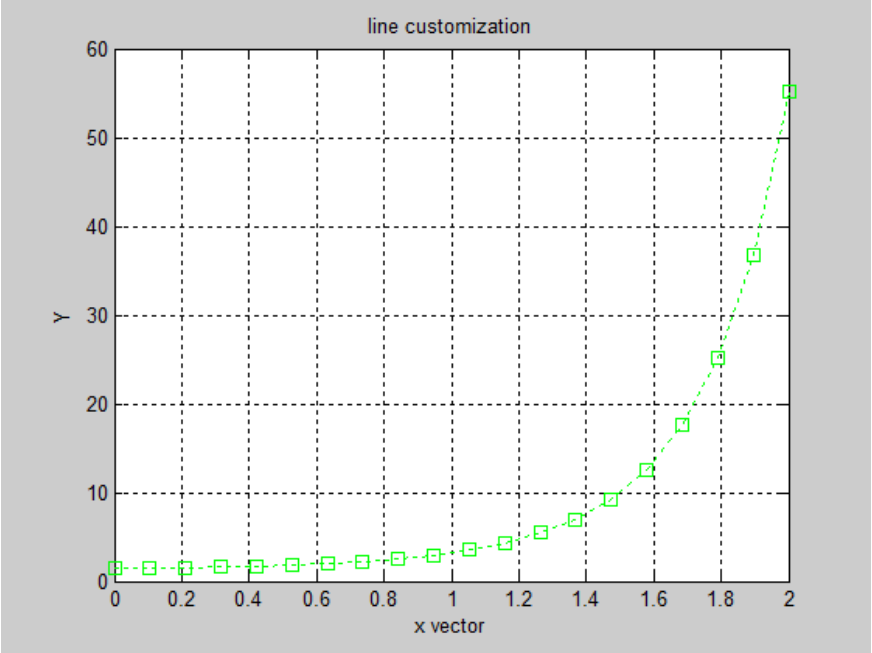
Ex: draw the following function:

$$y = e^{x^2} + \tan(0.15\pi)$$

For 20 values of x between 0 and 2 . Put the label (Y) on the y-axis and (x vector) on the x-axis and a title of (line customization). Make the curve line green and dotted with square marker type.

Sol:

```
>> x=linspace(0,2,20);
>> y=exp(x.^2)+tan(0.15*pi);
>> plot(x,y,'g:s')
>> grid on
>> title('line customization')
>> ylabel('Y')
>> xlabel('x vector')
```

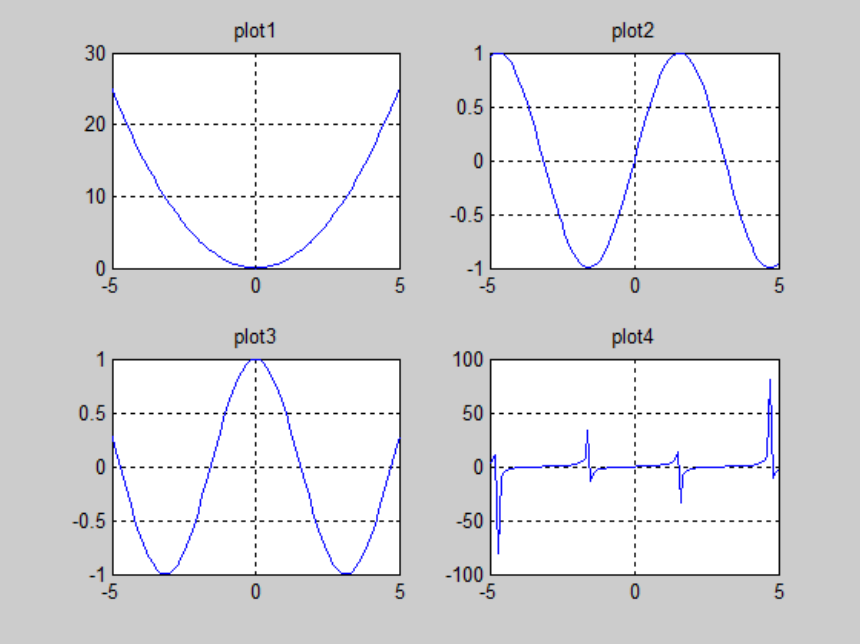


It is possible to add more plots to an existing graph using (hold on) so that the new plot will be added to the current one. In order to recognize the new one, plot with different color any avoid using the default colors.

```
>> plot(x,y,'g:s')
>> y=exp(x);
>> hold on
>> plot (x,y,'r+');
```

The figure window can be divided into an array of (m x n) of smaller size windows while every small window has one or more graph. Theses windows are addressed from 1 to the total number of windows in the array starting from the top left one and moving from left to right.

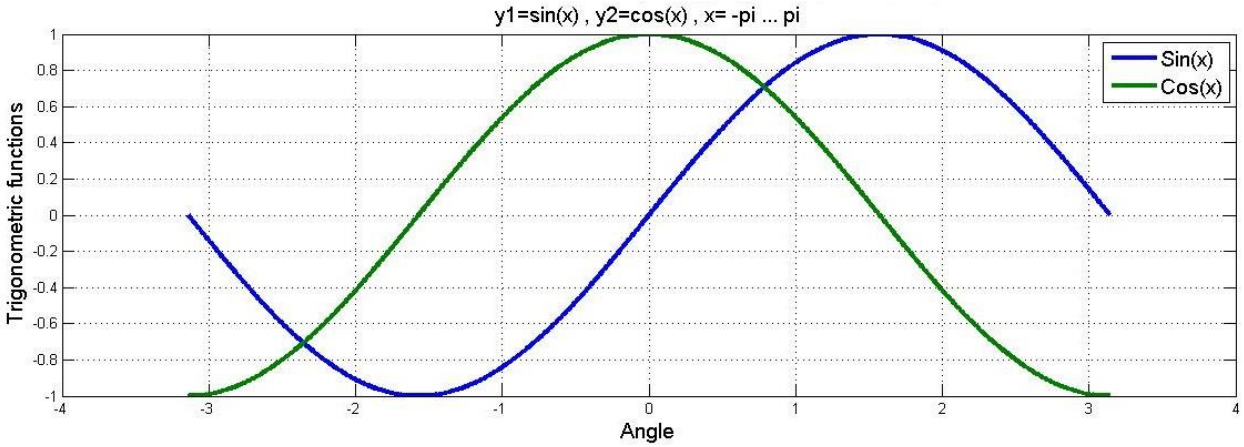
Ex: write the required MATLAB expressions to plot the figures shown below:



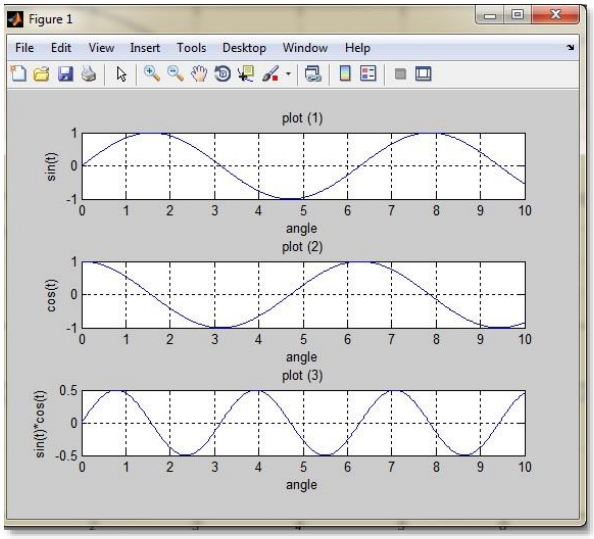
Sol:

```
>> x=-5:0.1:5;
>> y1=x.^2;
>> y2=sin(x);
>> y3=cos(x);
>> y4=tan(x);
>> subplot(2,2,1); plot(x,y1); grid on; title('plot1')
>> subplot(2,2,2); plot(x,y2); grid on; title('plot2')
>> subplot(2,2,3); plot(x,y3); grid on; title('plot3')
>> subplot(2,2,4); plot(x,y4); grid on; title('plot4')
```

H.W: Plot the following figure:



H.W: Write a MATLAB program to plot x1, x2, x3 functions shown in the figure (1) knowing that (t) is regularly spaced vector between (0 to 10 with 512 elements). When: x1=sin(t), x2=cos(t), x3=sin(t)*cos(t)



3D Plotting:

Figures with three dimensions are plotted in MATLAB using **plot3** function

Plot3 (x,y,z, LineSpec) Produce plots according to the given coordinates in 3-D space using the specified line style, marker and color. It is possible to plot multiple plots on the same figure using **Plot3 (x1, y1, z1, x2, y2, z2)**.

Ex: Draw a 3-D plot for 100 points between -5 and 5 for :

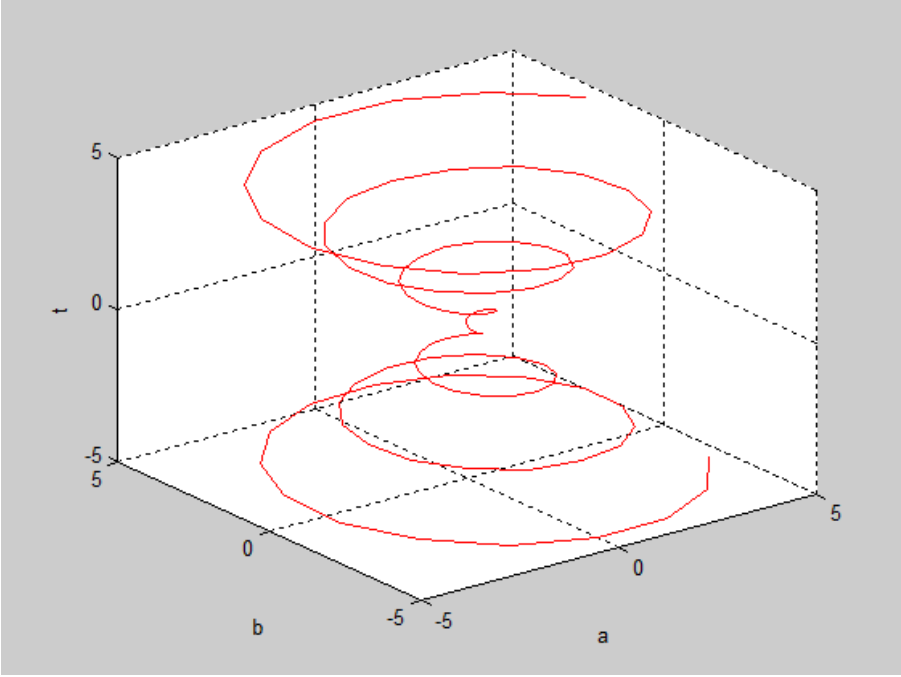
$$a = tsin(4t)$$

$$b = tcos(4t)$$

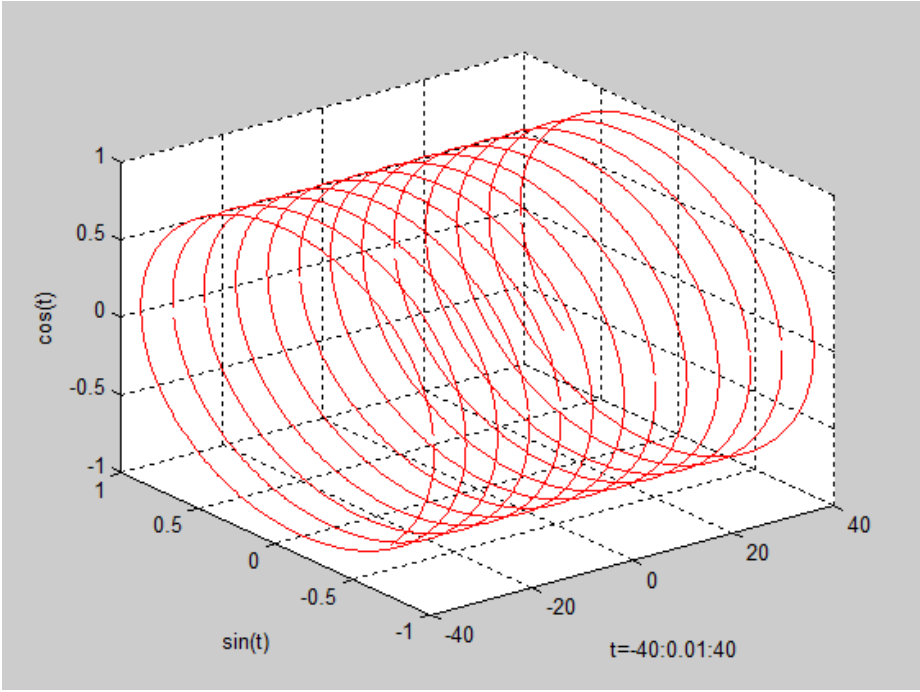
Use red line

Sol:

```
>> t=linspace(-5,5,100);
>> a=t.*sin(4*t);
>> b=t.*cos(4*t);
>> plot3 (a,b,t, 'r'); grid on
```



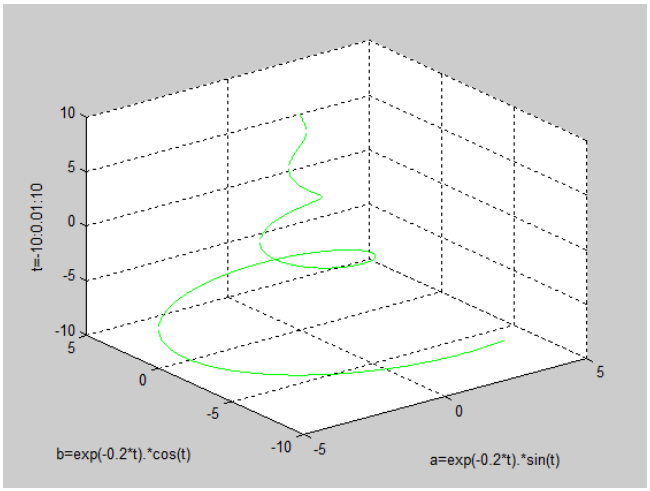
Ex: plot the following figure:



Sol:

```
>> t=-40:0.01:40;
>> a=sin(t);
>> b=cos(t);
>> plot3(t,a,b,'r--'); xlabel('t=-40:0.01:40'), ylabel('sin(t)'), zlabel('cos(t)'); grid on
```

H.w.: Plot the following figure



mesh(X,Y,Z) : produce mesh plot as a 3-D surface. The plot is given by matrix z as heights of a grid in the x-y plane. The plot edges are colored according to the heights given by z.

Ex: Use meshgrid function to generate three matrices with the same size using the following limits and equations:

$$z = \frac{\cos(r)}{r}$$

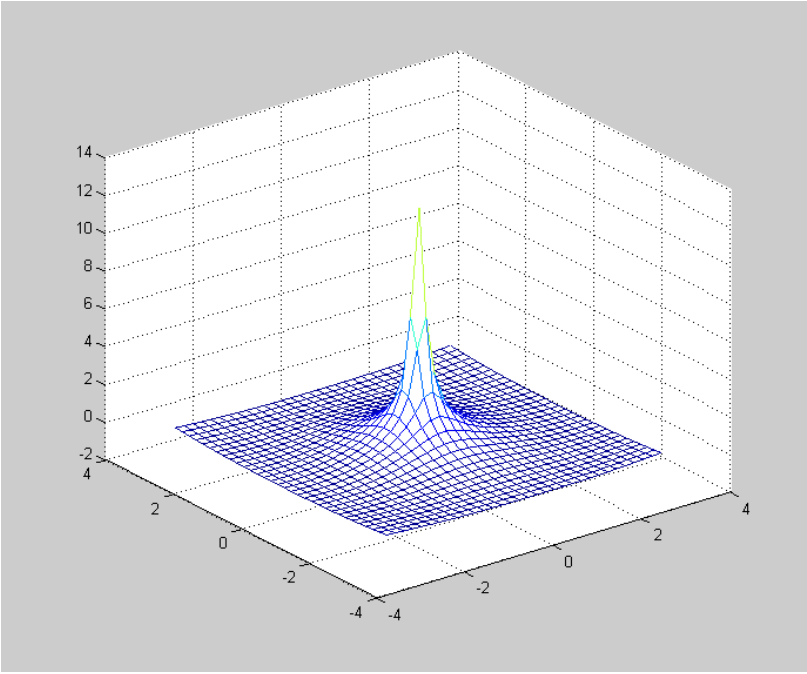
$$r = \sqrt{(x^2 + y^2)}$$

$$[x, y] = -\pi{:}\ 0.2{:}\ \pi$$

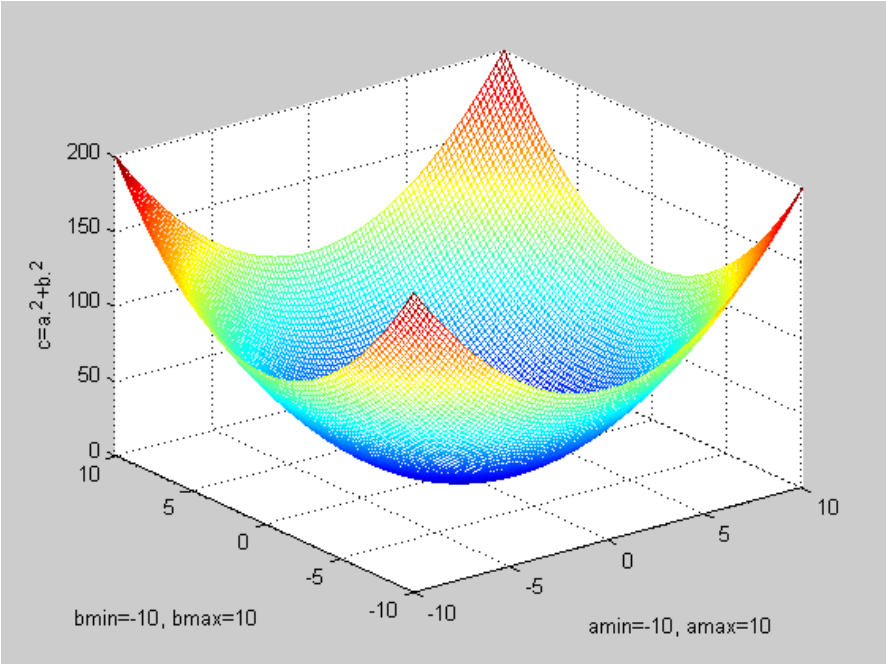
Then plot them using mesh function.

Sol:

```
>> [x,y]=meshgrid (-pi:0.2:pi);

>> r=sqrt(x.^2+y.^2);

>> z=cos(r)./r;

>> mesh(x,y,z)
```



Ex: Plot the figure below:



Sol:

```
>> [a,b]=meshgrid(-10:0.2:10);

>> c=a.^2+b.^2;

>> mesh(a,b,c); xlabel('amin=-10, amax=10'); ylabel('bmin=-10, bmax=10');
   zlabel('c=a.^2+b.^2')
```

surf (x,y,z) : produce a 3-D plot as three dimensional surface. The plot is given by matrix z as heights of a grid in the x-y plane. The plot edges are colored according to the heights given by z.

Ex: Generate three matrices with the same size using the following limits and equations:

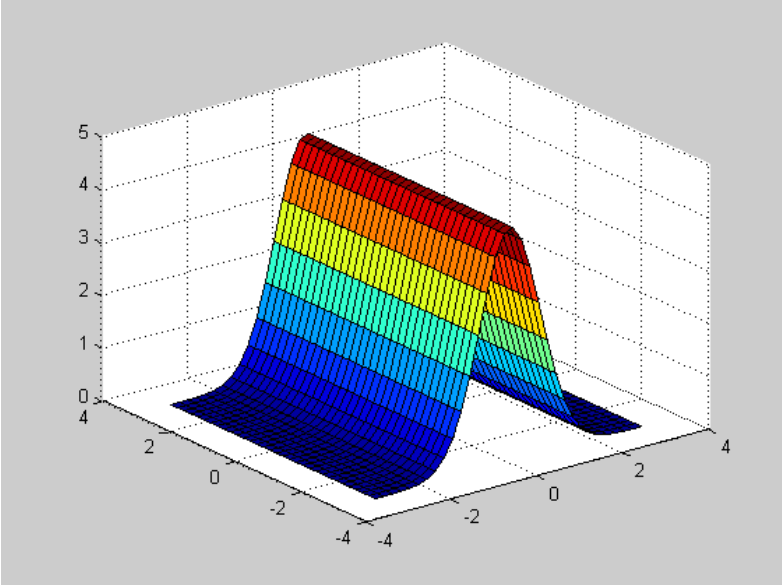
$$z = x^2 e^{(-x^2-y^2)}$$

$$[x, y] = -\pi: 0.2: \pi$$

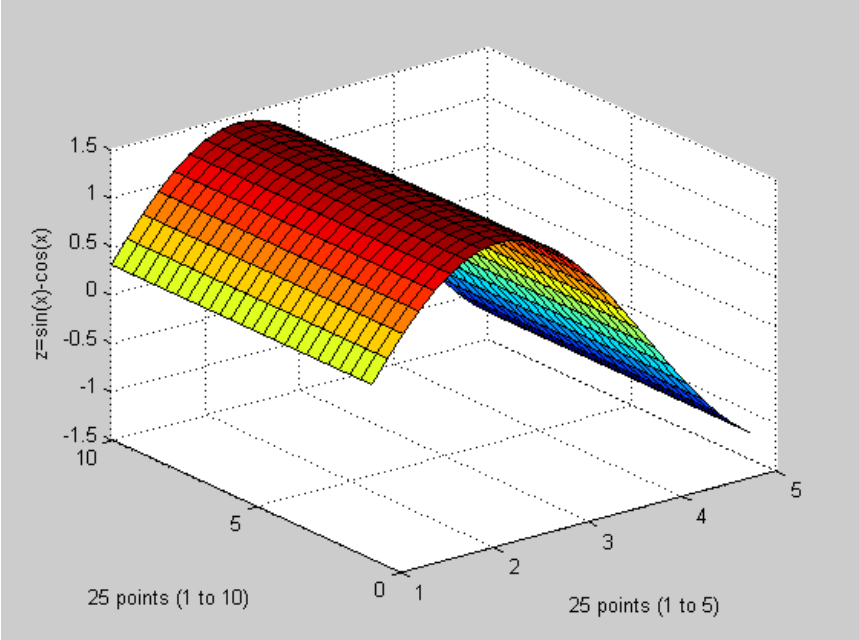
Then plot them using surf function.

Sol:

```
>> [x,y]=meshgrid(-pi:0.2:pi);
>> z=x.^2*exp((-x.^2-y.^2));
>> surf(x,y,z)
```



H.W: Plot the figure below:



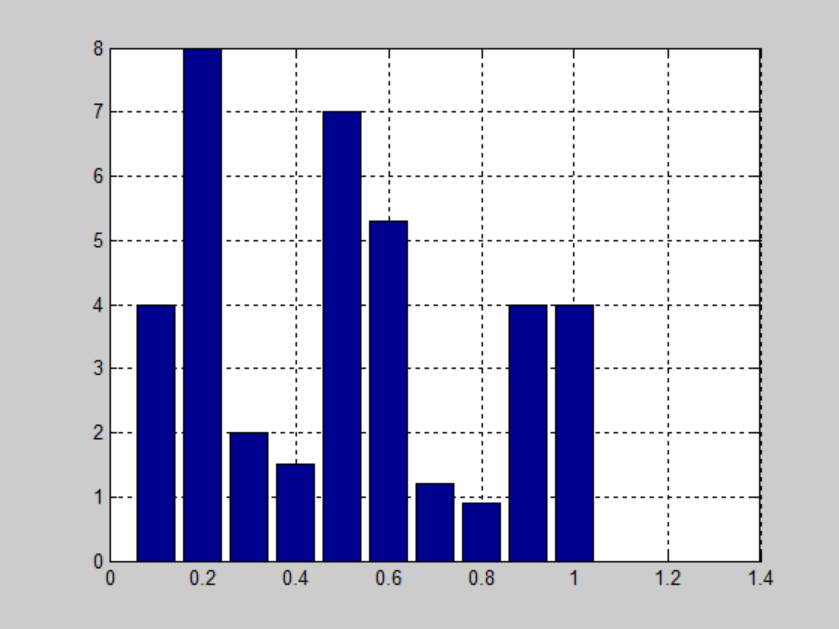
Bar charts

Bar (x,y)

Ex: for x=0.1 0.2 0.3 0.4 … 1 and y=4 8 2 1.5 7 5.3 1.2 0.9 4 4, draw bar chart.

Sol:

```
>> x=0.1:0.1:1;
>> y=[4 8 2 1.5 7 5.3 1.2 0.9 4 4];
>> bar (x,y); grid on
```



Contour line

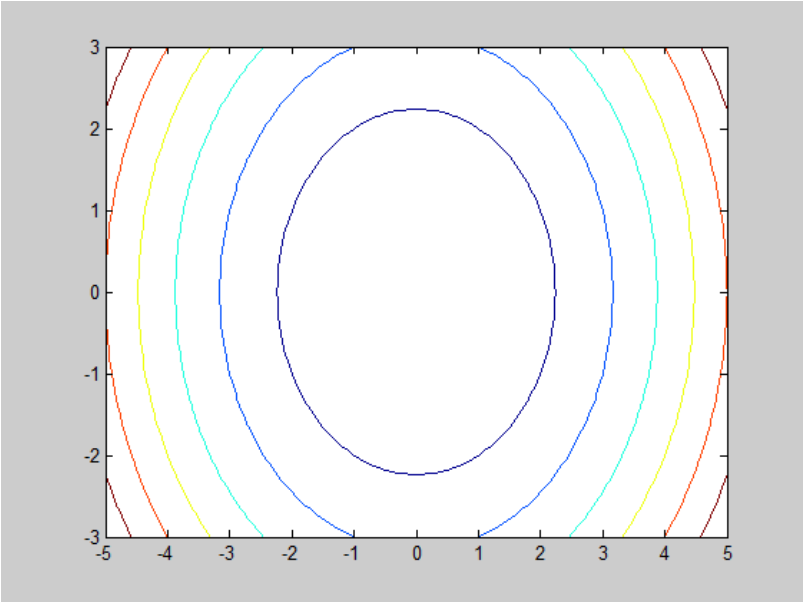
A contour line of a function of two variables is a curve along which the function has a constant value. It is a cross-section of the three-dimensional graph. Two data sets x and y are needed. This is done by calling the meshgrid command.

Ex: For −5 ≤ x ≤ 5 and −3 ≤ y ≤ 3 with increment of 0.1 for both the values and

g=f(x,y)= x2 + y2 . Draw a contour line.

Sol:

```
>> [x,y] = meshgrid(-5:0.1:5,-3:0.1:3);
>> g = x.^2 + y.^2;
>> contour(x,y,g);
```



Loglog(x,y): produces a log-log of x vs y.

Semilogx(x,y): produces a semi-log of x vs y with logarithmic scale on x axis.

Semilogy(x,y): produces a semi-log of x vs y with logarithmic scale on y axis.

Conditional statements

If Statement

The conditional statements are necessary to execute a series of MATLAB expressions with certain conditions.

if (condition)
(MATLAB expression)
End

The condition can be written using the MATLAB logic operations as follows:

Function	Description
==	Equal
<	Less than
>	Greater than
<=	Less than or equal
>=	Greater than or equal
~=	Not equal
&	And
|	Or
~	not

The expression is executed only if the result of the logic operation is true.

This can be extended to have multiple conditions with multiple expressions as follows:

if (condition)
(MATLAB expression)
elseif (condition)
(MATLAB expression)
elseif (condition)
(MATLAB expression)
.
.
else
(MATLAB expression)
End

The above conditions will be checked in sequence with their related expressions.

Ex:

```
>> a = 3; b = 5;

>> if a<b

j = -1
```

```
end
j =   -1
```

Ex:

```
>> a =10;b = 10;
>> if a < b
j = -1
elseif a > b
j = 2
else
j = 3
end
j =   3
```

Ex:

```
>> x=10; y=7;
>> if x < 7 & y < 7
z = -x * y
elseif x == 7 | y == 7
z = 0
else
z=x^2
end
z =   0
```

Repetitive Statement

For loop statement

It is used to repeat a series of MATLAB expressions in a predetermined way. **For** statement generate a vector while MATLAB loop with each element in the vector.

For variable = expression
statement
end

Ex:

```
>> for j=1:4
v(j) = j;
end
>> v
v =    1     2     3     4
```

Ex:

```
>> for i=1:4
for j=1:4
t(i,j) = i*j;
end
end
>> t
t =
     1     2     3     4
     2     4     6     8
     3     6     9    12
     4     8    12    16
```

Ex: Find summation of even numbers between 0 and 1000

```
>> sum = 0;
>> for i = 0 : 2 : 1000
sum = sum + i;
```

```
end
>> sum
sum =    250500
```

While loop statement

It is used to repeat a series of MATLAB expressions in a conditional way. With while-statement, MATLAB will repeat the statement as long as the given condition is met.

while condition
statements
end

Ex:

```
>> x=1;
>> while x <10
x=x+1;
end
>> x
x = 10
```

Ex:

```
>> x=10;
>> while x > 1
x = x/2;
end
>> x
x = 0.6250
```

Ex:

```
>> sum = 0;
>> x = 1;
>> while x < 4
sum = sum + 1/x;
x = x + 1;
end
>> sum
sum = 1.8333
```

Break statement

It is needed to limit the number of iterations or needed to avoid infinite loops. It is useful also to break the iterations with a certain condition.

Ex:

```
>> n = 0;
>> x=100 ;
>> while x > 1
x = x/2;
n=n+1;
if n > 50
break
end
end
>> x

x =    0.7813
```

Ex: Use loop in 20 iterations to calculate y in equation y=xsin(0.3x)?

```
>> for x=1:20
y=x*sin(0.3*x)
end
```

H.W: Write a MATLAB expressions to compute the sum of integers ranging from 1 to 5000.

Ex: Give an approximation value for π according to:

$$\pi \approx 4 \sum_{k=0}^{n} \frac{(-1)^k}{2k+1}$$

What is the error using 100/1000/10000 terms? Give a comparison with the value of **pi** which is given in MATLAB.

Sol:

```
>> summation=0;
>> for k=0:100
summation= summation+ 4* ((-1)^k)/(2*k+1);
end
>> error=summation-pi

error = 0.0099
```

H.W.: Try again with 1000/10000 terms.

Ex: What is the greatest result (Sum) less than 1000 that can be obtained from the following equation:

$$Sum = 1^2 + 2^2 + 3^2 + \cdots + n^2$$

Sol:

```
>> n=1;
>> Sum(n)=0;
>> while Sum(n)<1000
Sum(n+1)=Sum(n)+n^1;
n=n+1;
end
>>  Sum(n-1)

ans =  990
```

Ex: Build the following matrix using for loops and if statements in relationship between the positions of rows and columns:

$$A = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Sol:

```
>> for i=1:4
for j=1:4
if i==j
A(i,j)=1;
else
A(i,j)=0;
end
end
end
>> A
A =
    1    0    0    0
    0    1    0    0
    0    0    1    0
    0    0    0    1
```

H.W: Build the following matrices using for loops and if statements in relationship between the positions of rows and columns:

$$A = \begin{matrix} 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 7 \\ 5 & 6 & 7 & 8 \end{matrix} \qquad B = \begin{matrix} e^{1\pi} & e^{2\pi} & e^{3\pi} & e^{4\pi} \\ e^{2\pi} & e^{4\pi} & e^{6\pi} & e^{8\pi} \\ e^{3\pi} & e^{6\pi} & e^{9\pi} & e^{12\pi} \\ e^{4\pi} & e^{8\pi} & e^{12\pi} & e^{16\pi} \end{matrix}$$

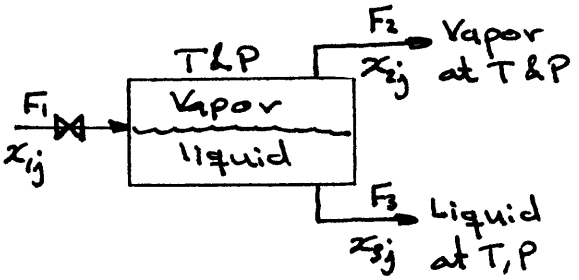
Computer Applications in Chemical Engineering

Flash Vaporization

A flash vaporization unit is probably one of the most important application in chemical engineering thermodynamics. A flash process is one in which a fluid stream of known overall composition and flow rate passes through a throttle, turbine or compressor and into a vessel (flash drum) where liquid and vapor phase are separated before passing through the outlet streams. such process may be operated under many different sets of conditions including the following:

1. Constant temperature and pressure (isothermal flash).
2. Constant enthalpy and pressure (isenthalpic flash).
3. Constant entropy and pressure (isentropic flash).

If the vaporizer is designed so that there is enough contact time for transfer of mass and heat between liquid and vapor, then a condition of equilibrium is reached, and the distribution of each component is given by its equilibrium constant (K). The equilibrium value of K for the jth component is defined by:

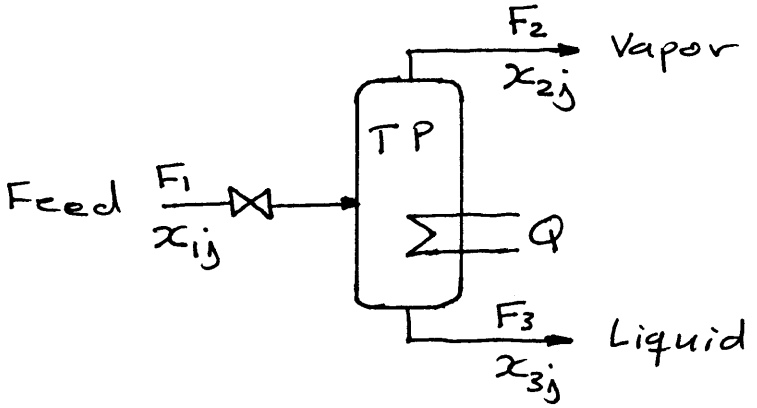


$$K_j = \frac{x_{2j}}{x_{3j}}$$

Thus, K is the ratio of the mole fraction in the vapor to the mole fraction in the liquid. In general, the value of K depends on the temperature, pressure, and product composition.

Isothermal flash (constant T and P)

In this process, the feed, at temperature T and Pressure P, passes through a throttle and enters the flash vessel, where liquid and vapor phases may separate. The operating pressure P of the unit is controlled some way and heat is supplied or removed at rate Q through a heat exchanger so as to maintain isothermal conditions at temperature T. The molar flow rate F_1 of the feed to the unit is specified, together with the overall composition (mole fractions x_{ij}) and the temperature and pressure at which the unit operates. The objectives of the calculation is to determine the compositions (x_{2j} and x_{3j}) and the molar flow rates (F_2 and F_3) of the vapor and liquid streams leaving the unit.



Applying material balance over the unit gives:

$$x_{ij}F_1 = x_{2j}F_2 + x_{3j}F_3 \qquad j=1,2,\cdots,N_c \quad (1)$$

Equation (1) may be rearranged:

$$x_{2j} = \frac{x_{ij}F_1 - x_{3j}F_3}{F_2} \qquad \text{----(2)}$$

or

$$x_{3j} = \frac{x_{ij}F_1 - x_{2j}F_2}{F_3} \qquad \text{----(3)}$$

Recall that:

$$x_{2j} = K_j x_{3j} \qquad \text{----(4)}$$

or

$$x_{3j} = \frac{x_{2j}}{K_j} \qquad \text{----(5)}$$

Substituting eq(4) into eq(3) and solving for x_{3j},

$$x_{3j} = \frac{F_1}{F_2} \frac{x_{1j}}{[(F_3/F_2) + K_j]} \qquad \text{----(6)}$$

Writing eq(6) in summation form:

$$1 = \sum_{j=1}^{N_c} x_{3j} = \sum_{j=1}^{N_c} \frac{F_1}{F_2} \frac{x_{1j}}{[(F_3/F_2)+K_j]} \qquad \text{----(7)}$$

Similarly, eq (5) is substituted into eq(2) to give:

$$x_{2j} = \frac{F_1}{F_2} \left[\frac{K_j x_{1j}}{K_j + (F_3/F_2)} \right] \qquad \text{---- (8)}$$

writing eq(8) in summation form:

$$1 = \sum_{j=1}^{N_c} x_{2j} = \sum_{j=1}^{N_c} \frac{F_1}{F_2} \left[\frac{K_j x_{1j}}{K_j + (F_3/F_2)} \right] \qquad \text{---- (9)}$$

subtracting eq(9) from eq(7) term by term gives:

$$\sum_{j=1}^{N_c} \frac{x_{1j}(1-K_j)}{(\alpha + K_j)} = 0 \qquad \text{--- (10)}$$

where α is defined by: $\alpha = \frac{F_3}{F_2}$ ----(11)

Equation (10) is called the flash equation and it contains one unknown (α). Once α is known, F_2 and F_3 are calculated by combining eq. (11) with the overall material balance equation ($F_1 = F_2 + F_3$), thus:

$$F_2 = \frac{F_1}{1+\alpha} \qquad \text{----(12)}$$

$$F_3 = F_1 - F_2 \qquad \text{----(13)}$$

Then, equations (6) and (8) yield all the unknown compositions of the liquid and vapor streams.

Ex Calculate the vapor and liquid flow rates and stream compositions that exit from a flash vessel at equilibrium at 150°F and 50 psia. Given:

Component	*No.*	*Mole fraction*	*k value (150°F and 50 psia)*
C_2	1	0.0079	16.2
C_3	2	0.1321	5.2
i-C_4	3	0.0849	2.6
n-C_4	4	0.269	1.98
i-C_5	5	0.0589	0.91
n-C_5	6	0.1321	0.72
C_6	7	0.3151	0.28

Script file (myfile.m)

```
% Isothermal Flash Calculations
% Nc      Number of components
% k(j)    Vap-liq equilibrium constant
% x(i,j) Mole fraction of component j in stream i
clear
global x k
Nc=7;
f(1)=100;
x=zeros(3,Nc);
x(1,:)=[0.0079 0.1321 0.0849 0.269 0.0589 0.1321 0.3151];
k=[16.2 5.2 2.6 1.98 0.91 0.72 0.28];
alpha_guess=0.5;
alpha=fzero('flashtp',alpha_guess);
f(2)=f(1)/(1+alpha);
f(3)=f(1)-f(2);
x(2,:)=(f(1)/f(2))*(k.*x(1,:)./(k+f(3)/f(2)));
x(3,:)=(f(1)/f(2))*(x(1,:)./(f(3)/f(2)+k));
% Displaying results
fprintf('Value of alpha is equal to : %f \n\n',alpha)
output=[[1:3]' f' x];
disp(output)
```

```
function f=flashtp(alpha)
global x k
f=sum(x(1,:).*(1-k)./(alpha+k));
```

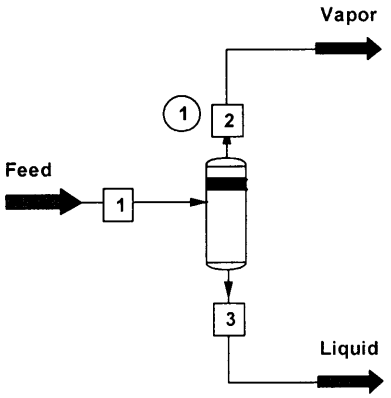
MATLAB Session:

```
>> myfile
Value of alpha is equal to : 0.821687

    1.0000  100.0000    0.0079    0.1321    0.0849    0.2690    0.0589    0.1321    0.3151
    2.0000   54.8942    0.0137    0.2078    0.1175    0.3463    0.0564    0.1124    0.1459
    3.0000   45.1058    0.0008    0.0400    0.0452    0.1749    0.0620    0.1561    0.5210
```

Note

The disp function dispaly the value of a variable without displaying its name.

Results using CHEMCAD simulation software



Flash Vaporization Unit

CHEMCAD 5.2.0 Page 1

Job Name: flashtp Date: 01/26/2008 Time: 19:53:54

Stream No.	1	2	3
Stream Name			
Temp F	150.0000*	150.0000	150.0000
Pres psia	50.0000*	50.0000	50.0000
Enth MMBtu/h	-6.3303	-3.1628	-3.1674
Vapor mole fraction	0.57260	1.0000	0.00000
Total lbmol/h	100.0000	57.2604	42.7396
Total lb/h	6756.7393	3534.7007	3222.0381
Total std L ft3/hr	177.0828	95.9329	81.1499
Total std V scfh	37947.86	21729.11	16218.75
Component mole fractions			
Ethane	0.007900	0.013118	0.000909
Propane	0.132100	0.202302	0.038047
I-Butane	0.084900	0.115052	0.044504
N-Butane	0.269000	0.340223	0.173578
I-Pentane	0.058900	0.057046	0.061384
N-Pentane	0.132100	0.115863	0.153854
N-Hexane	0.315100	0.156396	0.527724

Home work

For the above example, perform a degree of freedom analysis to find the number of design variables.

Answer

$N_v = 31$

$N_e = 17$

$N_d = 14$

Bubble and dew point calculations:

Mixtures do not have boiling points because the temperature of liquid and vapor rises during vaporization at constant pressure. Instead of boiling point, each mixture is charcterized by its bubble point and dew point. Bubble and dew points, like boiling point, depend on pressure. Liquid mixtures form their first bubble of vapor when the temperature is raised to the bubble point. Vapor mixtures form their first drop of liquid when temperature is lowered to the dew point. At intermediate temperatures above the bubble point and below the dew point, liquid and vapor phases coexist in equilibrium.

At equilibrium, the temperature and pressure are equal in both phases, liquid and vapor. Mole fraction constraints are:

$$\sum_{j=1}^{N_c} y_j = 1 \qquad (j = 1, 2, \ldots, N_c) \qquad \text{--- (1)}$$

$$\sum_{j=1}^{N_c} x_j = 1 \qquad (j = 1, 2, \ldots, N_c) \qquad \text{--- (2)}$$

where x refers to liquid and y refers to vapor. For non-ideal liquid mixture under atmospheric or lower pressure, the following equation applies:

$$P y_j = P_j^* \gamma_j x_j \qquad \text{--- (3)}$$

where P is the total pressure, P_j^* is the vapor pressure of each component and it is a function of temperature only,

γ_j is the activity coefficient in the liquid phase mixture, which is a function of temperature and liquid composition. For ideal solutions, γ is equal to 1.

Introducing K-values,

$$K_j = \frac{y_j}{x_j} \qquad \text{--- (4)}$$

where

$$K_j = \frac{P_j^* \gamma_j}{P} \qquad \text{---- (5)}$$

For bubble point calculations, liquid composition is given and the total pressure is known. Combining equations (2), (4), and (5) gives a single equation in one unknown T.

$$f(T) = \sum_1^{N_c} K_j x_j - 1 = 0 \qquad \text{--- (6)}$$

For dew point calculation, vapor composition is given and the total pressure is known. Combining equations (2), (4) and (5) gives a single equation in one unknown T.

$$f(T) = \sum_1^{N_c} \frac{y_j}{K_j} - 1 = 0 \qquad \text{--- (7)}$$

The first guess for temperature, is the molal average of the boiling points of the pure liquids.

<u>Ex</u> Calculate the bubble and dew point for 50% Benzene and 50% Toluene mixture under atmospheric pressure knowing that the solution is ideal. (Use the given composition as liquid composition for bubble point calculation and as vapor composition for dew point calculations).

Vapor pressure is calculated from the Antoine equation:

$$\log_{10} P^* = A - \frac{B}{C+T} \qquad \begin{matrix} P^* \text{ in mmHg} \\ T \text{ in } ^\circ C \end{matrix}$$

	A	B	C	Tb (°C)
Bz :	6.90565	1211.033	220.79	80.1
Tol :	6.95334	1343.943	219.377	110.62

MATLAB script file (myfile.m)

```
% Bubble and dew point calculations for ideal solution
% under atmospheric pressure.
% x      liquid phase mole fractions
% y      vapor phase mole fractions
% A,B,C Antoine equation parameters
% Tb     pure component boiling point (deg C)
% P      Total pressure (mmHg)
clear
global x y A B C P
A=[6.90565 6.95334];
B=[1211.033 1343.943];
C=[220.79 219.377];
Tb=[80.1 110.62];
P=760;
% Bubble point calculations starts here
x=[0.5 0.5];
T_guess=sum(x.*Tb);
T_bubble=fzero('bubble1',T_guess);
fprintf('Mixture bubble point is: %f deg C \n',T_bubble)
% Dew point calculations starts here
y=[0.5 0.5];
T_guess=sum(y.*Tb);
T_dew=fzero('dew1',T_guess);
fprintf('Mixture dew point is: %f deg C \n',T_dew)
```

```
function f=bubble1(T)
global A B C x P
pstar=10.^(A-B./(C+T));
K=pstar/P;
f=sum(K.*x)-1;
```

```
function f=dew1(T)
global y A B C P
pstar=10.^(A-B./(C+T));
K=pstar/P;
f=sum(y./K)-1;
```

MATLAB Session:

```
>> myfile
Mixture bubble point is: 92.111799 deg C
Mixture dew point is: 98.772842 deg C
```

Ex Calculate the bubble and dew point temperature for the following substances under atmospheric pressure. Note that z_j represent liquid composition for bubble point calculations and vapor composition for dew point calculations.

Component properties are:

Component	*No.*	*Mole fraction, z_j*	*Tb (°K)*	*T_c (°K)*	*P_c (atm)*
Neopentane	1	0.35	282.65	433.8	31.6
Carbon tetrachloride	2	0.1	349.85	556.4	45.0
Cyclohexane	3	0.1	353.85	553	40.0
Benzene	4	0.1	353.25	562	48.6
Ethylbenzene	5	0.35	409.35	619.6	38.0

Vapor pressure should be calculated from the Riedel correlation:

$$\log_{10} P_j^* = \log_{10} P_{cj} - 0.1183\,\phi_j + 7\log_{10} T_{rj} - (\alpha_j - 7)\,\psi_j$$

where: $$\phi_j = \frac{36}{Tr_j} + 42 \ln Tr_j - 35 - Tr_j^6$$

$$\Psi_j = 0.0364 \phi_j - \log_{10} Tr_j$$

The coefficient α_j is the Riedel factor, values for these chemicals are:

Component	α_j
Neopentane	6.771
Carbon tetrachloride	6.742
Cyclohexane	6.844
Benzene	6.848
Ethylbenzene	7.330

liquid phase activity coefficients should be calculated from:

$$\ln \gamma_j = \sum_{\substack{K=1 \\ j \neq K}}^{N_c} C_{jK} x_K (1 - x_j) - \sum_{\substack{M=1 \\ j \neq M \\ j \neq n \\ n > m}}^{N_c} \sum_{n=1}^{N_c} C_{mn} x_m x_n$$

The following experimental data have been measured for the various pairs in the five component mixture:

Liquid pair	A_{jk}	B_{jk} (°K)
1-2	0.018	149
1-3	0	88
1-4	0	274
1-5	0	274
2-3	-0.125	71
2-4	-0.045	53
2-5	-0.045	53
3-4	-0.797	394
3-5	-0.797	394
4-5	0	0

A_{jk} and B_{jk} are constants different for each binary liquid pair. C_{jk} which is the binary coefficient depends on temperature should be calculated from:

$$C_{jk} = A_{jk} + \frac{B_{jk}}{T}$$

MATLAB script file:

```
% Bubble and dew point calculations for non-ideal solution
% Tb    boiling point temp (deg K)
% Tc    critical temp (deg K)
% Pc    critical pressure (atm)
% alpha Riedel correlation factor
% x     mole fractions in liquid phase
% y     mole fractions in vapor phase
% Nc    Number of components
% A,B   binary interaction constants
% C     binary coefficient
% P     total pressure
clear
global z A B Nc Pc Tc P alpha
Nc=5;
z=[0.35 0.1 0.1 0.1 0.35];
Tb=[282.65 349.85 353.85 353.25 409.35];
Tc=[433.8 556.4 553 562 619.6];
Pc=[31.6 45 40 48.6 38];
alpha=[6.771 6.742 6.844 6.848 7.33];
A=zeros(Nc,Nc);
B=zeros(Nc,Nc);
A(1,2)=0.018;A(2,3)=-0.125;A(2,4)=-0.045;A(2,5)=-0.045;A(3,4)=-0.797;A(3,5)=-
0.797;
B(1,2:5)=[149 88 274 274];B(2,3:5)=[71 53 53];B(3,4:5)=[394 394];
P=1;
T_guess=sum(z.*Tb);
x=z;
T_bubble=fzero('bubble2',T_guess);
fprintf('Bubble point is: %f deg K \n',T_bubble)
y=z;
T_dew=fzero('dew2',T_guess);
fprintf('Dew point is: %f deg K \n',T_dew)
```

```
function f=bubble2(T)
global x P
k=pstar(T).*gamma(T)/P;
f=sum(k.*x)-1;
```

```
function f=dew2(T)
global y P
k=pstar(T).*gamma(T)/P;
f=sum(y./k)-1;
```

MATLAB Session:

```
>> myfile
Bubble point is: 303.895993 deg K
Dew point is: 381.877751 deg K
```

```
function f=gamma(T)
global x A B Nc
C=A+B/T;v=0;
for j=1:Nc
    sum1=0;
    sum2=0;
    for k=1:Nc
        if j~=k
            sum1=sum1+C(j,k)*x(k)*(1-x(j));
        end
    end
    for m=1:Nc
        for n=1:Nc
            if (j~=m & j~=n & n>m)
                sum2=sum2+C(m,n)*x(m)*x(n);
            end
        end
    end
    g(j)=exp(sum1-sum2);
end
f=g;
```

```
function f=pstar(T)
global Tc Pc alpha
Tr=T./Tc;
phi=36./Tr+42*log(Tr)-35-Tr.^6;
psi=0.0364*phi-log10(Tr);
f=10.^(log10(Pc)-0.1183*phi+7*log10(Tr)...
-(alpha-7).*psi);
```

MATLAB relational operators

There are operators with two numerical or string operands that yield either a true (1) or false (0) result depending on the relationship between the two operand. The following symbols are used for MATLAB relational operators:

operator	operation (logical test)
==	is equal to
~=	is not equal to
>	is greater than
>=	is greater or equal to
<	is less than
<=	is less or equal to

MATLAB logical operators:

There are operators with one or two logical operands that yield a logical result. The following symbols are used for MATLAB logical operators.

operator	operation
&	logical AND
|	logical OR
XOR	logical Exclusive OR
~	logical NOT

Computer Applications in Chemical Engineering

Adiabatic flame temperature

The temperature that results from a combustion process that occurs adiabatically with no work or change in kinetic or potential energy involved is commonly referred to as the adiabatic flame temperature. Because of the assumptions of no work and no changes in kinetic or potential energy, this is the maximum temperature that can be achieved for the given reactants.

EX Prepare a computer program that computes the flame temperature for complete combustion in air of the hydrocarbon $C_K H_\ell$ according to the reaction:

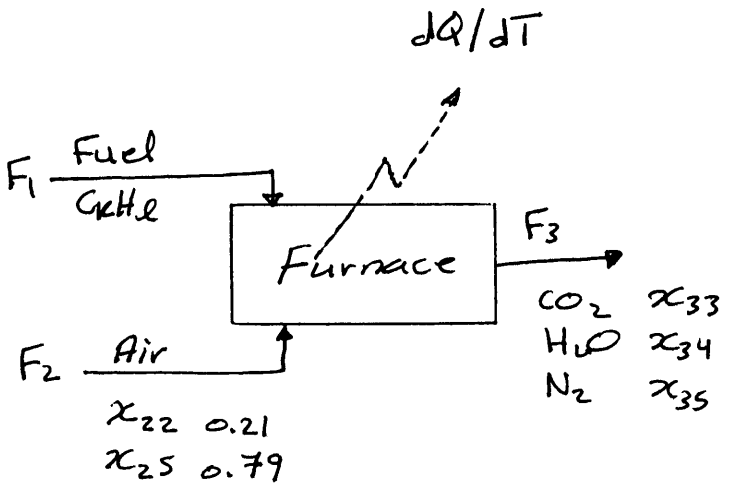
$$C_K H_\ell + \left[K + \frac{\ell}{4}\right] O_2 \rightarrow K CO_2 + \frac{\ell}{2} H_2O$$

Assume that the fuel and O_2 are fed to the flame chamber in stoichiometric ratio at room temperature and atmospheric pressure.

Use the program to compute the adiabatic flame temperature of H_2, CH_4, C_2H_6, C_2H_4, and C_2H_2.

Solution

component	NO.
$C_K H_\ell$	1
O_2	2
CO_2	3
H_2O	4
N_2	5



1. Material balance equations:

(a) For reacting species,

$$C: \quad K F_1 = F_3 x_{33} \qquad \text{---(1)}$$

$$H: \quad l F_1 = 2 F_3 x_{34} \qquad \text{---(2)}$$

$$O: \quad 2F_2 x_{22} = F_3 (2x_{33} + x_{34}) \qquad \text{---(3)}$$

(b) Inerts,

$$N_2: \quad F_2 x_{25} = F_3 x_{35} \qquad \text{---(4)}$$

2. Mole fraction constraints:

$$x_{33} + x_{34} + x_{35} = 1 \qquad \text{---(5)}$$

3. Energy balance equation:

$$F_1 h_1 + F_2 h_2 - F_3 h_3 - \frac{dQ}{dT} = 0 \qquad \text{---(6)}$$

4. Enthalpy equations:

$$h_1 = h_1(T_1, P_1) \qquad \text{---(7)}$$

$$h_2 = h_2(T_2, P_2, x_{22}) \qquad \text{---(8)}$$

$$h_3 = h_3(T_3, P_3, x_{33}, x_{34}) \qquad \text{---(9)}$$

The material balance portion of the problem can be solved separately, solving eq(1) to eq(5) gives:

$$x_{33} = \frac{K}{\beta} \qquad x_{35} = \frac{[(1-x_{22})/x_{22}][K + l/4]}{\beta}$$

$$x_{34} = \frac{l}{2\beta} \qquad F_3 = \beta F_1$$

$$F_2 = \frac{F_1}{x_{22}} \left(K + \frac{l}{4}\right)$$

where $$\beta = K + \frac{l}{2} + \left(\frac{1-x_{22}}{x_{22}}\right)\left(K + \frac{l}{4}\right)$$

$$= K + \frac{l}{2} + 3.76\left(K + \frac{l}{4}\right)$$

Notice that β have three terms in the expression for number of moles of CO_2, H_2O, and N_2 respectively.

For a reference temperature $T_o = 25°C = 298.15°K$ at 1 atm,

$$h_1 = \Delta h_1^f + \int_{T_o}^{T_1} C_{p_1}\, dT = \Delta h_1^f$$

(Δh_1^f: heat of formation)

$$h_2 = \sum x_{2j}\,\Delta h_j^f + \int_{T_0}^{T_2} \overline{C_{p_2}}\,dT$$

$$= 0$$

$$h_3 = \sum x_{3j}\,\Delta h_j^f + \int_{T_0}^{T_3} \overline{C_{p_3}}\,dT$$

Note that h_3 can be calculated from eq (6).

$$\therefore \sum x_{3j}\,\Delta h_j^f + \int_{T_0}^{T_3} \overline{C_{p_3}}\,dT - h_3 = 0 \qquad \text{---(10)}$$

---

where

$$\overline{C_{p_3}} = \sum_{j=3}^{N_c} x_{3j}\left(a_j + b_j T + c_j T^2 + d_j T^{-1/2}\right)$$

$$= \bar{a}_3 + \bar{b}_3 T + \bar{c}_3 T^2 + \bar{d}_3 T^{-1/2} \qquad \text{---(11)}$$

Solution of eq (10) for T_3 gives the adiabatic flame temperature.

Molal average coefficients ($\bar{a}_3$, $\bar{b}_3$, $\bar{c}_3$, and $\bar{d}_3$) are calculated from coefficients for the pure gases given below:

Component	a_j	$b_j \times 10^2$	$c_j \times 10^5$	d_j
O_2	6.732	0.1505	-0.01791	0
CO_2	18.036	-0.004474	0	-158.08
H_2O	6.970	0.3464	-0.04833	0
N_2	6.529	0.1488	-0.02271	0

The heat of formation for CO_2 and H_2O and some other hydrocarbons are listed below:

Component	Δh_f *(cal/mol)*
CO_2	-94050
H_2O	-57800
C_2H_2	54194
C_3H_8	-24820

Script file (myfile.m)

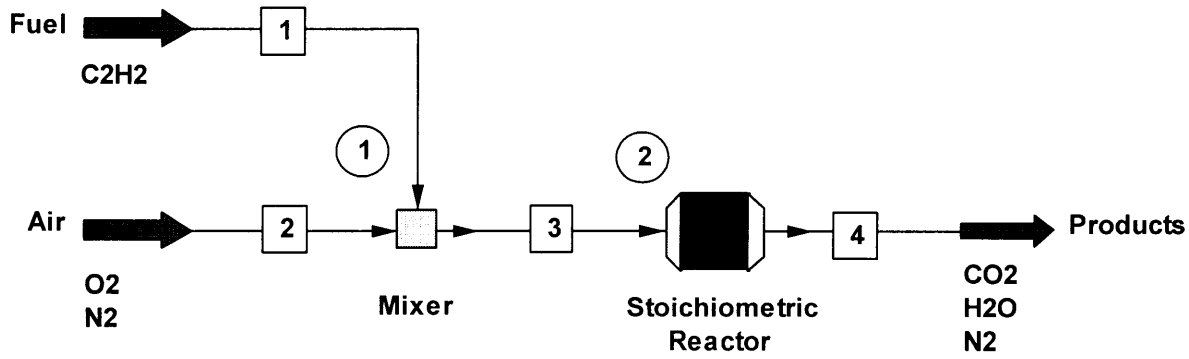
```
% Calculation of the adiabatic flame temperature
% for C(k)H(l) hydrocarbon fuel assuming complete combusion
%
% Nc     Number of components
% x      component mole fractions
% F      Flow rates
% T      Temperature
% P      Pressure
% h      Enthalpy
% hf     Heat of formation
% k      No. of  carbon element in hydrocarbon
% l      No. of  hydrogen element in hydrocarbon
% dq     Heat lost during the combustion process
clear
global a3 b3 c3 d3 T0 h hf x Nc
x(1,1)=1;x(2,2)=0.21;x(2,5)=1-x(2,2);
T0=298.15;T(1)=T0;T(2)=T0;
P(1:3)=1;Nc=5;
F(1)=100;dq=0;
k=input('No. of  carbon element in hydrocarbon (k): ');
l=input('No. of  hydrogen element in hydrocarbon (l):');
beta=k+l/2+((1-x(2,2))/x(2,2))*(k+l/4);
x(3,3)=k/beta;
x(3,4)=l/(2*beta);
x(3,5)=(((1-x(2,2))/x(2,2))*(k+l/4))/beta;
F(3)=beta*F(1);
F(2)=F(1)*(k+l/4)/x(2,2);
hf(1)=input('Enter heat of formation for fuel: ');
hf(2:Nc)=[0 -94050 -57800 0];
a=[18.036 6.97 6.529];
b=[-0.004474 0.3464 0.1488]*1e-2;
c=[0 -0.04833 -0.02271]*1e-5;
d=[-158.08 0 0];
a3=sum(x(3,3:Nc).*a);
b3=sum(x(3,3:Nc).*b);
c3=sum(x(3,3:Nc).*c);
d3=sum(x(3,3:Nc).*d);
h(1)=hf(1);
h(2)=0;
h(3)=(h(1)*F(1)+h(2)*F(2)-dq)/F(3);
T_guess=(h(3)-sum(x(3,3:Nc).*hf(3:Nc)))/a3+T0;
T(3)=fzero('flame1',T_guess);
% Displaying results
disp('      St. No.      F        T         P        h          x        ')
disp('      -------    ------   ------   ------    ------   -------------')
output=[ [1:3]' F' T' P' h' x];
disp(output)
```

```
function f=flame1(T)
global a3 b3 c3 d3 T0 h hf x Nc
f=a3*(T-T0)+b3*(T^2-T0^2)/2+c3*(T^3-T0^3)/3...
    +2*d3*(T^0.5-T0^0.5)...
    +sum(x(3,3:Nc).*hf(3:Nc))-h(3);
```

MATLAB Session:

```
>> myfile
No. of  carbon element in hydrocarbon (k): 0
No. of  hydrogen element in hydrocarbon (l):2
Enter heat of formation for fuel: 0
         St. No.     F          T             P       h                            x
         -------   ------     ------        -----   --------    ------------------------------------------------------------
           1         100      298.15          1          0       1         0           0           0           0
           2       238.1      298.15          1          0       0      0.21           0           0        0.79
           3       288.1      2529.2          1          0       0         0           0     0.34711     0.65289

>> myfile
No. of  carbon element in hydrocarbon (k): 2
No. of  hydrogen element in hydrocarbon (l):2
Enter heat of formation for fuel: 54194
         St. No.     F          T             P       h                            x
         -------   ------     ------        -----   --------    ------------------------------------------------------------
           1         100      298.15          1      54194       1         0           0           0           0
           2      1190.5      298.15          1          0       0      0.21           0           0        0.79
           3      1240.5      2903.8          1     4368.8       0         0     0.16123    0.080614     0.75816
```

Calculation of adiabatic flame temperature of C_2H_2 using CHEMCAD simulation software



Stream No.	1	2	3	4
Stream Name				
Temp K	298.1500*	298.1500*	298.1080	2895.5973
Pres atm	1.0000*	1.0000*	1.0000	1.0000
Enth cal/hr	2.4560E+009	-8.6415E+005	2.4552E+009	2.4552E+009
Vapor mole fraction	1.0000	1.0000	1.0000	1.0000
Total lbmol/hr	100.0000	1190.4760	1290.4760	1240.4760
Total lb/hr	2603.8000	34346.2422	36950.0430	36950.0000
Total std L ft3/hr	67.8302	636.0056	703.8358	721.7128
Total std V scfh	35903.22	427419.19	463322.41	445370.78
Component mole fractions				
Acetylene	1.000000	0.000000	0.077491	0.000000
Oxygen	0.000000	0.210000	0.193727	0.000000
Carbon Dioxide	0.000000	0.000000	0.000000	0.161228
Water	0.000000	0.000000	0.000000	0.080614
Nitrogen	0.000000	0.790000	0.728782	0.758157

1. MATLAB Basics:

1.1 Introduction: MATLAB is a High performance language used mainly for engineering and science applications. It can achieve efficiently many technical computations like:

- Integration
- Solving the differential equations
- Solving the algebric equations
- Data representing and visualization
- Statistics
- Curve fitting

Moreover, MATLAB is widely used as a computational tool in a range of applications including:

- Signal processing and Communications
- Image and video processing
- Control systems
- Test and measurements
- Computational finance

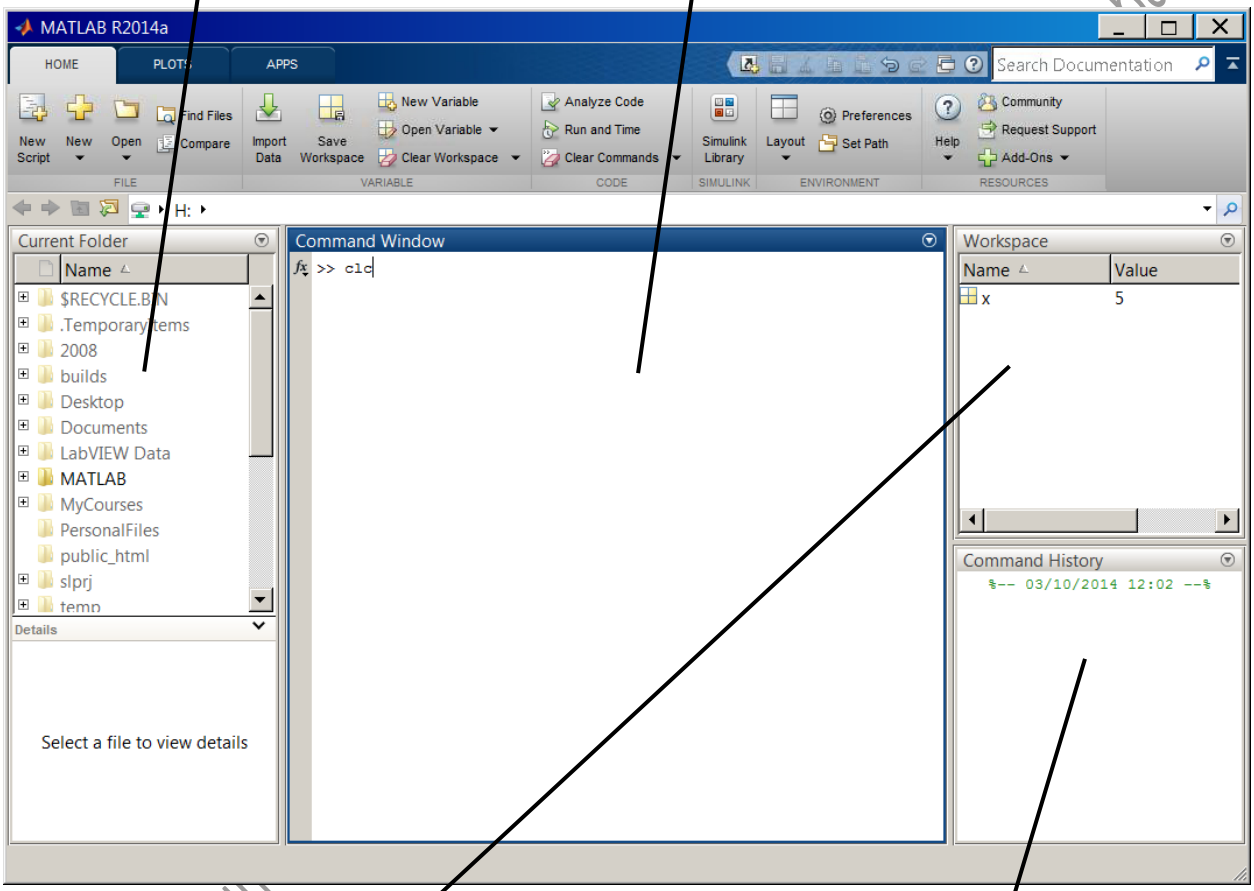
1.2 Starting MATLAB: you can start MATLAB by double-clicking on the MATLAB shortcut icon (MATLAB 7.0.4) on your Windows desktop. Once you start MATLAB, the MATLAB desktop window appears. This window that contains other windows can be listed as (see Fig1.1):

- Command window
- Command History
- Workspace
- Help
- Current directory

1.3 Quitting MATLAB: Type “quit” in the command window or select File: Exit MATLAB

Current folder: allows you to access the project folders and files

Command window: commands are entered directly at the command line.



Workspace: shows all the variables which are entered or imported from other files.

Command history: shows commands and codes which had been entered previously in command window

1.4 Commonly used Operators:

Operator	Description
+	Addition
-	Subtraction
*****	Scalar and matrix multiplication
.*	Array multiplication
^	Scalar and matrix exponentiation
.^	Array exponentiation
/	division
./	Array division
:	generates uniformly spaced elements
( )	Encloses functions arguments and array indices
[ ]	enclosures array elements
.	Decimal point
…	Line-continuation
,	Separates elements in a row
;	Separates columns

1.4 Special variables and constant:

Name	Description
ans	Last answer
i, j	Imaginary unit
Inf	Infinity
Nan	Not a number
Pi	π

1.5 Functions:

Trigonometric

Function	Description
sin (x)	Sine of (x) in radian
cos (x)	Cosine of (x) in radian
tan (x)	Tangent of (x) in radian
asin (x)	Inverse sine , result is an angle in radian
acos (x)	Inverse cosine , result is an angle in radian
atan (x)	Inverse tangent , result is an angle in radian
exp (x)	e^x
log (x)	ln (x)
log10 (x)	Log_{10} (x)
sqrt (x)	Square root
abs (x)	Absolute value
sign (x)	Signal of x
max (x)	Maximum value
min (x)	Minimum value
ceil (x)	Round towards +∞
floor (x)	Round towards -∞
round (x)	Round to nearest integer
rem (x)	Reminder after division
angle (x)	Phase angle
conj (x)	Complex conjugate

Logic operator

Function	Description
==	Equal
<	Less than
>	Greater than
<=	Less than or equal
>=	Greater than or equal
~=	Not equal
&	And
|	Or
~	not

1.6 Vectors

Vectors is a one-dimensional array of elements so that they can be row vectors or column vectors:

V=[3 12 8 -5] (row vector), or V=[3; 12; 8; -5] (column vector)

The elements of any vector can be addressed as:

- V(i) where i is the position of the targeted element.
- V(2:4) giving the elements of positions: second to fourth.

1.6.1 Vectors operations

Given that:

A=[7, 11, 5, 8,12] , B=[4, 10, 3, 6, 13] note: r

Run: A+B , A-B , A*5 , B' , [A,B] , |B| , $[A(1)^2\ A(2)^2\ A(2)^3]$

A+B , A-B Both vectors must have same Number of elements	>> A+B >> A-B	5*A Scalar multiplication: each element multiplied by the element	>> A*5	A' Transpose operation: Changes a column vector into a row vector and vise versa	>> A'		

|A|

Magnitude of a vector:

$$|A| = \sqrt{A_1 + A_2 + A_3 + \cdots}$$

>> Am=A.*A

>> Asum=sum(A)

>> sqrt(A)

1.7 Matrix:

Matrix is a two-dimensional array of elements.

$$m = \begin{matrix} 1 & 2 & 5 & 12 \\ 8 & 7 & 3 & 16 \\ 11 & 4 & 6 & 20 \\ 15 & 0 & 9 & 21 \end{matrix}$$

The elements of the matrix is addressed as:

- m (i,k) where i and k are the positions of the row and column respectively.
- m (:, 2) giving the second column:
  2
  7
  4
  0

- m (:,2:4) giving the columns second to fourth:
  2 5 12
  7 3 16
  4 6 20
  0 9 21

- m (2:3, 3:4) giving a part of the matrix containing the rows: second to third, and columns: third to fourth:
  3 16
  6 20

1.7.1 Matrix processing:

It is possible to delete a row or column of a matrix as:

- >> m(4,:)= []
  1 2 5 12
  8 7 3 16
  11 4 6 20

Delete the fourth row

- >> m(4,:)= []

1	2	5
8	7	3
11	4	6
15	0	9

Delete the fourth column

1.7 Matrix operation:

A=[2 4 6; 1 3 5; 7 9 11] , B=[0 1 2; 3 4 5; 6 7 8]

Operation	Description	Command
A+B , A-B	Both Matrices must have same dimensions	>> A+B >> A-B
A/B	Both Matrices must have same dimensions	>> A/B
A*2, A+2, A/2	Scalar operation: each element is added, multiplied or divided by the number	>> A*2 >> A+2 >> A/2
A’	Transpose operation: switches the rows and columns	>> A’
A*B	Multiplication is possible only when the number of columns in A is equal to the number of rows in B	>> A*B

System of linear equations

For the following systems of linear equations:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

Can be solved in MATLAB by expressing them in matrices forms as a first step as:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$a * x = b$$

where :

$$a = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad . \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_2 \end{bmatrix} \quad . \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Once these matrices are given in workspace, x_1 , x_2 and x_3 can be calculated as:

$$x = inv(a) * b$$

Ex: Solve the following linear equations by using MATLAB expression:

$$x_1 + x_2 + x_3 = 3$$
$$2x_1 - x_2 - 2x_3 = 6$$
$$4x_1 + 2x_2 + 3x_3 = 7$$

Solution:

```
>> a=[1 1 1; 2 -1 -2; 4 2 3]; b= [3; 6; 7];
>> x=inv(a)*b
x =
   2.0000
   4.0000
  -3.0000
```

H.W-1: Solve the following system of linear equations:

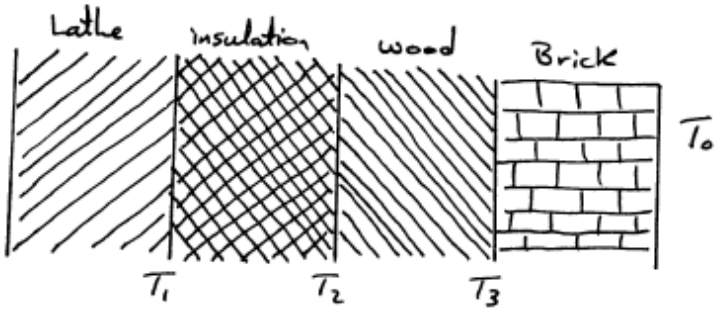
$$x_1 + x_2 + 2x_3 = 8$$
$$-1x_1 - 2x_2 + 3x_3 = 1$$
$$3x_1 - 7x_2 + 4x_3 = 10$$

H.W-2: Solve the following system of linear equations:

$$6x_1 - 2x_2 + x_3 = 11$$
$$x_1 + 2x_2 - 5x_3 = -1$$
$$-2x_1 + 7x_2 + 2x_3 = 5$$

EX. The thermal resistance for a wall area of $1\,m^2$ are $R_1 = 0.036$, $R_2 = 4.01$, $R_3 = 0.408$, and $R_4 = 0.038$ °K/W. Suppose that $T_i = 20$ and $T_o = -10$°C. Find the other three temperatures and the heatloss rate q. Given:

$$q = \frac{1}{R_1}(T_i - T_1) = \frac{1}{R_2}(T_1 - T_2) = \frac{1}{R_3}(T_2 - T_3) = \frac{1}{R_4}(T_3 - T_c)$$



Solution:

The solution gives a set of linear equations:

$$qR_1 + T_1 = T_i$$
$$qR_2 - T_1 + T_2 = 0$$
$$qR_3 - T_2 + T_3 = 0$$
$$qR_4 - T_3 = -T_o$$

$$\Rightarrow A = \begin{bmatrix} R_1 & 1 & 0 & 0 \\ R_2 & -1 & 1 & 0 \\ R_3 & 0 & -1 & 1 \\ R_4 & 0 & 0 & -1 \end{bmatrix} \quad b = \begin{bmatrix} T_i \\ 0 \\ 0 \\ -T_o \end{bmatrix}$$

Solution:

Substituting , R_1 , R_2, R_3 and R_4 as well as T_i and T_o in a and b respectively as:

```
>> a=[0.036 1 0 0; 4.01 -1 1 0; 0.408 0 -1 1; 0.038 0 0 -1]; b= [20; 0; 0; 10];
>> x=inv(a)*b
x =
   6.6785
  19.7596
  -7.0214
  -9.7462
```