Condition

1. If Statement

What are you doing if you want certain parts of your program to be executed **only** in limited circumstances? The way to do that is by putting the code within an "if" statement. The most basic structure for "if" statement is as following:

```
if  (relation)
   (matlab commands)
end
```

More complicated structures are also possible including combinations like the following:

```
if  (relation)
   (matlab commands)
elseif  (relation)
   (matlab commands)
elseif (relation)
   (matlab commands)
.
.
else
   (matlab commands)
end
```

Standard comparisons can be made by using relations. The relational operators in MATLAB are;

<   less than

>   greater than

<= less than or equal

>= greater than or equal

== equal

~= not equal.

For example, the following code will set the variable j to be –1 if a less than b:

```
a = 5;
b = 3;
if a<b
   j = -1
end
```

Additional statements can be added for more decision. The following code sets the variable j to be 2 when a greater than b.

```
a = 4;b = 3;
if a < b
   j = -1
elseif a > b
   j = 2
end
```

The **else** statement provides all that will be executed if no other condition is met. The following code sets the variable j to be 3 when a is not greater and less than b.

```
a = 4;b = 4;
if a<b
   j = -1
elseif a>b
   j = 2
else
   j = 3
end
```

Matlab allows you to write together multiple condition expressions using the standard logic operators, "&" *(and)*, "¦" *(or)*, and "˜" *(not)*.

For example to check if *a* is less than *b* and at the same time *b* is greater than or equal to *c* you would use the following commands:

```
if a < b & b >= c
  Matlab commands
end
```

For example

```
x=1; y=0;
if x < 0 & y < 0
z = -x * y
elseif x == 0:y == 0
z = 0
else
z = x^2
end
```

The output of the above code is

```
z=
    0
```

2.Loop

Many programs require iteration, or repetitive execution of a block of statements. MATLAB has two loop statements: **for** and **while** statements. All of the loop structures in matlab are started with a keyword such as "for" or "while" and all of them end with the word "end".

2.1 For loop

If you want to repeat a certain commands in a predetermined way, you can use the "for" loop. The "for" loop will loop around some statement, and you must tell Matlab where to start and where to end. Basically, you give a vector in the "for" statement, and Matlab will loop through for each value in the vector:

The for statements have the following structure:

```
For variable = expression
      statement
end
```

For example, a simple loop will go around four times:

```
for j=1:4
j
end
```

The output of the above code is

```
j =
    1
j =
    2
j =
    3
j =
    4
```

For another example, if we define a vector and later want to change it elements, we can step though and change each individual entry:

```
v = [1:3:10];
for j=1:4
v(j) = j;
end
>>v
v =
    1     2     3     4
```

Matrices may also be constructed by loop. Here is an example using two "for" loops.

```
for i=1:10
for j=1:10
t(i,j) = i*j;
end
end
```

Notice that there isn't any output, since the only line that would produce any (**t(i,j) = i/j;**) ends in a semi-colon. Without the semi-colon, Matlab would print the matrix t 100 times!

Example 1

Find summations of even numbers between 0 and 100

```
sum = 0;
for i = 0 : 2 : 100
      sum = sum + i;
end
sum
sum =
      2550
```

Note: if the increment is one you may write the first and the last limit without writing a step. In this case, the for-line above would become **for i = 0:100**.

warning: If you are using complex numbers, you cant use i as the loop index.

2.2 While Loop

If you don't like the "for" loop, you can also use a "while" loop. It is sometimes necessary to repeat statements based on a condition rather than a fixed number of times. The "while" loop repeats a sequence of commands as long as some condition is met. The general form of a while loop is

```
while relation
      statements
end
```

Example 2

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

The statements will be repeatedly executed as long as the relation remains true.

Example 3

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

The condition is evaluated before the body is executed, so it is possible to get zero iteration when x equal to 1 or less.

Example 4

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

2.3 Break statement

It's often a good idea to limit the number of repetitions, to avoid infinite loops (as could happen in above example if x=Inf). This can be done using **break**. A break can be used whenever you need to stop the loop (break statement can be used to stop both for and while loops in same way), for example.

```
n = 0;
x=100
while x > 1
x = x/2;
n = n+1;
if n > 50
break
end
end
>> x
x =
   0.7813
```

A break immediately jumps execution to the first statement after the loop.

Note: When using programs with loops, you should also know how to stop a program in the middle of its execution, such as in the case of your programs contains an infinite loop. During a program is running, the **Ctrl+C** command will stop it.

Exercise 1:

Use loop in 20 iteration to calculate x in equation x=2sin(x)?

Solution:

```
format long
x=1
for i=1:20
   x=2*sin(x)
end
```

The results will be:

x = 1
x = 1.682941969615793
x = 1.987436530272152
x = 1.828907552623580
x = 1.933747642340163
x = 1.869706153630775
x = 1.911316179125262
x = 1.885162348212226
x = 1.901985209663949
x = 1.891312851651419
x = 1.898145620030117
x = 1.893795924017278
x = 1.896575152213355
x = 1.894803506466833
x = 1.895934551140671
x = 1.895213162228076
x = 1.895673549670181
x = 1.895379846173585
x = 1.895567260289186
x = 1.895447688999369
x = 1.895523983862163

You can see the convergence through the digits after the dot. The best guess for x is
x = 1.895523983862163

Exercise 2:

Use loop to find the bubble point of ternary system (Ethanol 40 mol%, Water 20 mol% and Benzene 40 mol%). Knowing that the vapor pressure for three components are calculated by:

Ethanol $P^o_e=\exp(18.5242-3578.91/(T-50.5))$
Water $P^o_w=\exp(18.3036-3816.44/(T-46.13))$
Benzene $P^o_b=\exp(15.9008-2788.51/(T-52.36))$

Where

$K_i= P^o_i/P_t$, $P_t=760$, $y_i=K_i\times x_i$

At Bubble point $\sum yi=\sum Ki\times xi=1$

Solution:

```
Xe=.4;Xw=.2;Xb=.4;
for T=273.15:.01:450;
Pe=exp(18.5242-3578.91/(T-50.5));  % Vapor pressure of Ethanol
Pw=exp(18.3036-3816.44/(T-46.13)); % Vapor pressure of Water
Pb=exp(15.9008-2788.51/(T-52.36)); % Vapor pressure of Benzene
% evaluation of equilibrium constants
Ke=Pe/760; % Ethanol
Kw=Pw/760; % Water
Kb=Pb/760; % Benzene
sum=Ke*Xe+Kw*Xw+Kb*Xb; % The condition to find the boiling point
if sum>1
break
end
end
T
```

Gives the result

```
T =
  355.5300
```

Exercise 3:

Given that the vapor pressure of methyl chloride at 333.15 K is 13.76 bar, write a code to calculate the molar volume of saturated vapor at these conditions using Redlich/Kwong equation. Knowing;

$a=0.42748*R^2T_c^{2.5}/P_c$

$b=0.08664*RT_C/P_C$

$V_{i+1=}(RT/P)+b-(a*(V_i-b))/(T^{1/2}PV_i(V_i+b))$

R=83.14 , Tc= 416.3 k , Pc= 66.8 bar

Solution:

```
T=333.15;
Tc=416.3;
P=13.76;
Pc=66.8;
R=83.14;
a=0.42748*R^2*Tc^2.5/Pc;
b=0.08664*R*Tc/Pc;
```

```
V(1)=R*T/P;
for i=1:1:10
V(i+1)=(R*T/P)+b-(a*(V(i)-b))/(T^.5*P*V(i)*(V(i)+b));
end
    V'
```

The program results as follows

```
ans =
1.0e+003 *
   2.0129
   1.7619
   1.7219
   1.7145
   1.7131
   1.7129
   1.7128
   1.7128
   1.7128
   1.7128
```

Note that after the 7 th trail the value of V_i is constant and further trails does not have any effect on its value.

Exercise 4:

A simple force balance on a spherical particle reaching terminal velocity in a fluid is given by;

$$V_t = \sqrt{\frac{4g(\rho_P - \rho)Dp}{3C_D\rho}}$$

Where

V_t : Terminal velocity in m/s

g : Acceleration of gravity

p_p: Particle density

D_p: The diameter of the spherical particle in m

C_D: Dimensionless drag coefficient.

The drag coefficient on a spherical particle at terminal velocity varies with Reynolds number (Re) as followings:

C_D=24/Re for Re< 0.1

C_D=24*(1+0.14* Re^0.7)/Re for 0.1=<Re=< 1000

C_D=0.44 for 1000<Re=< 350000

C_D=0.19-8*10^4/Re for 350000 <Re

Where

Re =(D_p v_tp)/μ

g=9.80665 m/s^2

ρ=994.6 kg/m^3

p_p=1800 kg/m^3

μ=8.931×10^{-4} kg/m.s.

D_p=0.000208 m

Calculate the terminal velocity of spherical particle?

Solution:

The above problem cannot be solved without using trial and error method. Therefore you must assume a velocity to calculate the C_D which is a important to calculate new velocity. Write the following code:

```
g=9.80665;
p=994.6; pp=1800;
mu=8.931e-4;
Dp=0.000208;
vt=1; Velocity(1)=vt;
for m=1:1:20
Re=(Dp*vt*p)/mu;
if Re < 0.1
CD=24/Re;
elseif Re >= 0.1 & Re <= 1000
CD=24*(1+0.14*Re^0.7)/Re;
elseif Re >1000  &  Re <= 350000
CD=0.44;
elseif Re>350000
CD=0.19-8*10^4/Re;
end
vt=((4*g*(pp-p)*Dp)/(3*CD*p))^.5;
Velocity(m+1)=vt;
if abs (Velocity(m+1)-Velocity(m))<.0001
   break
end
end
Velocity'
```

This code gives the result:

```
ans =
   1.000000000000000
   0.053845727997707
   0.025084090993404
   0.018981846393486
   0.017008369140870
   0.016271175897107
   0.015980109059108
   0.015862612988699
   0.015814754689716
```

Note: the loop will stop when the desired accuracy is reached

Practice Problems

1) Provide the right answers and use MATLAB to check them.

```
if 0 < x < 10
    y = 2*x
elseif 10 < x < 50
    y = 3*x
else
    y = 200
end
```

a) x = -1 y = ?
b) x = 5 y = ?
c) x=10 y = ?
d) x = 45 y = ?
e) x = 100 y = ?

2) Write a computer program to calculate molar volume and compressibility factor for gaseous ammonia by using of the Redlich-Kwong equation of state. The equations and variables are listed below.

$$P = \frac{RT}{(V-b)} - \frac{a}{V(V+b)\sqrt{T}} \qquad a = 0.42747\left(\frac{R^2 T_C^{5/2}}{P_C}\right) \qquad b = 0.08664\left(\frac{RT_C}{P_C}\right)$$

The variables are defined by

P= 56 atm

V= molar volume in L/g-mol

T=450 K

R= gas constant (*R* = 0.08206 atm·L/g-mol·K)

Tc= the critical temperature (405.5 K for ammonia)

Pc= the critical pressure (111.3 atm for ammonia)

Compressibility factor is given by

$$Z = \frac{PV}{RT}$$

3) If x is a row or column vector, the function **sum(*x*)** returns the sum of the elements of x. Assuming that x is a column vector of length m and k is a row vector containing the integers from 1 to n, for each of the following, write MATLAB code employing **for** loops to evaluate the expression, and then write *single* line MATLAB statements employing the function sum that duplicates the calculation.

1) $\sum_{i=1}^{m} \frac{1}{x_i}$ 2) $\sum_{k=1}^{n} \frac{1}{1+k}$ 3) $\sum_{i=1}^{m} \frac{x_i}{1+\sin x_i}$ 4) $\sum_{i=1}^{m} x_i e^{-x_i^2}$