Vander Waals Equation

The Vander Waals equation is an equation of state for a fluid composed of particles that have a non-zero volume and a pair wise attractive inter-particle force (such as the Vander Waals force.) It was derived by Johannes Diderik Vander Waals in 1873, who received the Nobel prize in 1910 for "***his work on the equation of state for gases and liquids***". The equation is based on a modification of the ideal gas law and approximates the behavior of real fluids, taking into account the nonzero size of molecules and the attraction between them.

Equation

The first form of this equation is

$$\left(p + \frac{a'}{v^2}\right)(v - b') = kT$$

where

p is the pressure of the fluid, *v* is the volume of the container holding the particles divided by the total number of particles, *k* is Boltzmann's constant, *T* is the absolute temperature, *a'* is a measure for the attraction between the particles, *b'* is the average volume excluded from *v* by a particle.

 # Program to Compute Pressure from Van der Waal's Equation P = $\frac{nRT}{V-nb} - \frac{n^2 a}{V^2}$

```
#include<stdio.h>
# include <conio.h>
Void main()
{
int n;
float a,br,v,p;
clrscr();
printf("\n Enter value of 'a' for the given
gas:");
scanf("%f"; &a);
printf("\n Enter value of 'b' for the given
gas:");
```

```
scanf(%f", & b);
r= 0.0821;
printf("\n Enter no. of moles of
gas:");
scanf("%d", & n));
printf("\ n Enter the volume of
gas");
scanf("\%f"; v);
print*"\Enter the temperature in
Kelvin:");
scanf("%f", & t);
p= ((n*r*t)/(v-n*b)) – ((n*n*a)/(v*v));
Printf("/n The calculated pressure is =% f atm",
p);
getch ();
}
```

Radioactive decay

Radioactive decay is the process by which an atomic nucleus of an unstable atom loses energy by emitting ionizing particles (ionizing radiation). The emission is spontaneous, in that the atom decays without any interaction with another particle from outside the atom (i.e., without a nuclear reaction). Usually, radioactive decay happens due to a process confined to the nucleus of the unstable atom, but, on occasion (as with the different processes of electron capture and internal conversion), an inner electron of the radioactive atom is also necessary to the process.

Radioactivity was first discovered during 1896 by the French scientist Henri Becquerel, while working on phosphorescent materials. These materials glow in the dark after exposure to light, and he suspected that the glow produced in cathode ray tubes by X-rays might be associated with phosphorescence. He wrapped a photographic plate in black paper and placed various phosphorescent salts on it. All results were negative until he used uranium salts. The result with these compounds was a blackening of the plate.

Particular radionuclide's decay at different rates, each having its own decay constant (λ). The negative sign indicates that N decreases with each decay event. The solution to this first-order differential equation is the following function:

$$N(t) = N_0 e^{-\lambda t} = N_0 e^{-t/\tau}.$$

Where N_0 is the value of N at time zero (t = 0). The second equation recognizes that the differential decay constant λ has units of 1/time, and can thus also be represented as $1/\tau$, where τ is a characteristic time for the process. This characteristic time is called the time constant of the process. In radioactive decay, this process time constant is also the mean lifetime for decaying atoms. Each atom "lives" for a finite amount of time before it decays, and it may be shown that this mean lifetime is the arithmetic mean of all the atoms' lifetimes, and that it is τ, which again is related to the decay constant as follows:

$$\tau = \frac{1}{\lambda}.$$

 # Program to Compute the decay constant by K= $\frac{2.303}{t} log \frac{N_0}{N}$

Program

```
# include <stdio.h>
#include<conio.h

>

#include<math.h

>

void main ( );

{

clrscr ( );

float K, t, No, N;

printf("\n Enter the initial amount of radioactive substance

:");

scanf("%f", & No);
```

```
printf("\n Enter the remaining amount of radioactive

substance:");

scanf("%f", &N);

printf("\n Enter the time

interval:");

scanf("%f", &t);

K= (2.303/t)*(log(No/N));

printf("\n Decay constant = %f",

K);

getch () ;

}
```

Half-life

A more commonly used parameter is the half-life. Given a sample of a particular radionuclide, the half-life is the time taken for half the radionuclide's atoms to decay. The half-life is related to the decay constant as follows:

$$t_{1/2} = \frac{\ln 2}{\lambda} = \tau \ln 2.$$

This relationship between the half-life and the decay constant shows that highly radioactive substances are quickly spent, while those that radiate weakly endure longer. Half-lives of known radionuclide's vary widely, from more than 10^{19}years (such as for very nearly stable nuclides, e.g., ^{209}Bi), to 10^{-23} seconds for highly unstable ones.

The factor of ln2 in the above relations results from the fact that concept of "half-life" is merely a way of selecting a different base other than the natural base e for the lifetime expression. The time constant τ is the "1/e" life (time till only 1/e = about 36.8% remains) rather than the "1/2" life ofradionuclide where 50% remains (thus, τ is longer than $t_{½}$). Thus, the following equation can easily be shown to be valid. n=

#Program to Compute the number of half life n by N=N$_0$ $\left(\frac{1}{2}\right)^n$

Program

```
#include<stdio.h

>
```

```
#include<conio.h
>
#include<math.h
>
main()
{
float N, No.;
int n;
clrscr ();
printf("\n Enter the initial amount of radioactive substance:");
scanf("%f", & No);
printf("\n The remaining of radioactive
substance:"); scanf("%f", & N);
n= ((log N/No)) / (log (0.5));
print ("\n Number of half lives =
%d",n); getch ( ) ;
}
```

 # Program to Compute average life period by T= $1.44 \times t$

Program

```
#           include
<stdio.h>
#include
<conio.h>    void
main ( );
{
float t,
T; clrscr
```

```
( );
printf("\n Enter the value of half life
period:");
scanf("%f", & t);
T= (1.44 * t);
printf("\ n Average life period
=%f",T);
 getch ( );
}
```

 # Program to Compute the half life by $t_{½}$= $\frac{0.6931}{k}$

Program

```
#include<stdio.h
>
#include<conio.h
>
 void main ( );
{
float t, k;
clarcr (
);
printf("\n Enter the decay constant of Radioactive Reaction
:");
scanf("%f", & k");
t= (0.6931/k);
printf("\n The half life period of given substance =
%f", t);
```

```
getch ( );

}
```

 # Program to Compute lattice energy by $_0= \left(\frac{NAe^2 Z1\,Z2}{r_0}\right)^n$

Program

```
#   include

<stdio.h>

#include

<conio.h>

main ( );

{

clrsc ( );

long double u, N, A, e,

Z1, Z2, r; int n;

printf("\n Enter Avogadro

Number:");

scanf("%Lf", & N);

printf ("\n Enter Madlung

Constant:");

scanf("% Lf', & A);

printf("\n Enter Electrical

Charge:");

scanf("/%Lf", & e);

printf("\n Enter Charge on

cations:");

scanf("%Lf", &Z1);
```

```
printf("\n Enter charge

Anions:");

scanf("%Lf",& Z2)

printf("/n Enter minimum distance between cation and anion at which potential energy
becomes minimum:");

scanf("%Lf", &r);

printf ("\n Enter Born Exponent:");

scanf("%d", & n);

u=(N*A*e*e*Z1*Z2*(1-

n))/r*n); printf("\n

lattice energy = %Lf", u);

getch ( );

}
```

 # Calculate kinetic energy of a gas by E = 3/2 nRT where R = 8.324 JK^{-1}, mol^{-1} T = temperature in kelvin (^0C + 273) n = given number of moles of gas.

```
# include < stdio . h >

# include < conio . h >

main()

{

 int T, n;float R, E;

clrscr()

;R = 8.324;

printf("\n Enter the number of moles:");
```

```
scanf("% f", &n);

printf("\n Enter the temperature in centigrade:");

scanf("%d", &T);T = T + 273;

E = (3 * n * R * T)/2;

printf("\n The kinetic energy of given no. of moles of gas = %f joule",E);

getch();

}
```

Normality, Molarity and Molality of Solution

Molarity(M) is the number of moles of solute dissolved in one liter of a solution and the unit for molarity is moles/L.

M = moles of soluteliters of solution (Note: If you are given volume in mL or some other volume unit, you need to convert it to liters.)

Lets say you dissolve 1.00mol of a solute into 0.500L of solution. The molarity(M) would be 1.00mol/0.500L = 2.00mol/L = 2.00M

Molality(*m*) is the number of moles per kilogram of solvent. It is determined by dividing the number of moles (n) of the solute by the mass of the solvent in kg.

m = moles of solutemass of solvent in kg

Lets say you dissolve 0.75mol of a solute into 2.50L of water. Since the density of water is 1.00g/mL and one liter of water is 1000mL, the mass of a liter of water is 1.00kg. So 2.50L of water has a mass of 2.50kg.

The molality, *m* , of this solution would by 0.75mol/2.50kg = 0.3mol/kg = 0.3 *m* .

Normality is a measure of concentration equal to the gram equivalent weight per liter of solution. Gram equivalent weight is the measure of the reactive capacity of a molecule. The solute's role in the reaction determines the solution's **normality**. **Normality** is also known as the equivalent concentration of a solution. In chemistry, the **equivalent concentration** or **normality** of a solution is defined as the molar concentration c_i divided by an equivalence factor f_{eq}:

Normality = c_i/f_{eq}