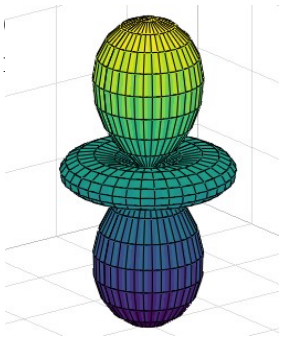
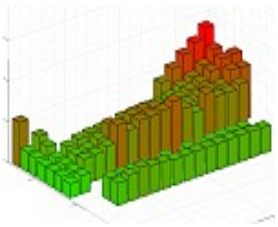
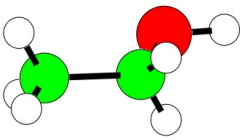
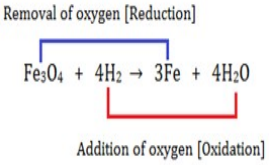
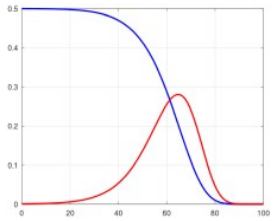
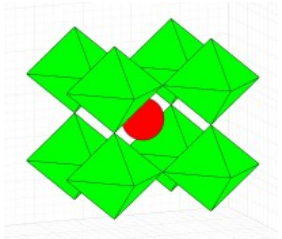
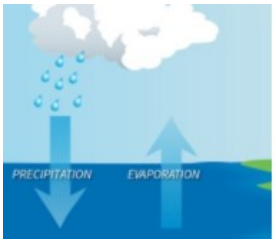
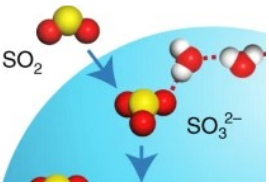
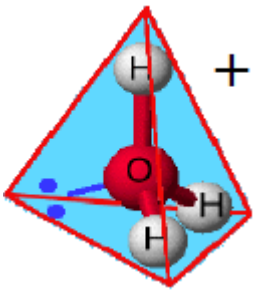
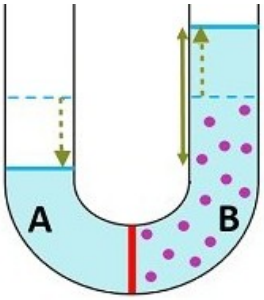
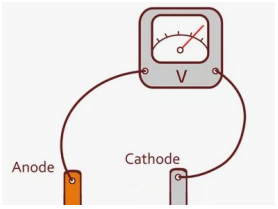
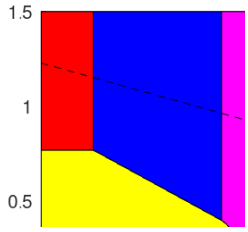
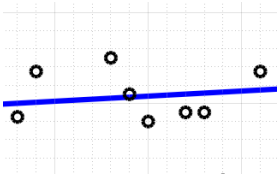
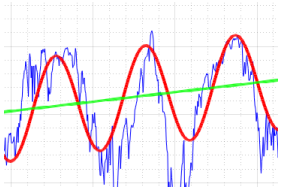
FUNDAMENTAL CHEMISTRY WITH OCTAVE

Daniele Mazza

(The images are links to the single chapters)

 1. Atomic and molecular orbitals	 2. 3D periodic table of the elements	 3. Molecular rendering	 4. Reaction balancing	 5. Chemical kinetics
 6. Crystal structures	 7. Gases and vapours	 8. Chemical gas-phase equilibria	 9. Equilibria in aqueous solution	 10. Colligative properties
 11. Seawater and CO_2 equilibria	 12. Electrochemistry	 13. Prevalence diagrams	 14. Data interpolation	 15. Frequency analysis

About the Author

Introduction

Most developers today use the so called ‘ third-generation languages’ such as C, C++, Python, and Java. A third-generation language, a general purpose language in nature, gives the developer the kind of precise control needed to write exceptionally fast applications that can perform a wide array of tasks.

Fourth-generation languages, like Matlab (short for Matrix Laboratory) or Octave (the free nearly equivalent platform), on the contrary, are designed with a specific purpose in mind, which in the case of Matlab/Octave, is scientific and technical computing. In this sense, it has been designed for empowering the user to work with heterogeneous collections of data, rather than individual variables, making it easier for the user to focus on the task, instead on the language.

In order to obtain the free Octave platform and to gain further information, please visit www.octave.org

For those interested in the Matlab version of this book, of the same Author, please pay a visit here

Chapter 1. Atomic and molecular orbitals

1.1 Atomic orbitals

An atomic orbital is a functions which describes the wave nature of the different electrons (or electron pairs) in an atom. They offer a way for evaluate the probability of finding an electron (interpreted as a charged particle) in any specific spatial region around the atom's nucleus.

There are four different kinds of orbitals appearing in the chemical elements of the Periodic Table, denoted by the letters *s, p, d* and *f*, each one with a different shape. Of the four types, *s* and *p* orbitals are the most common in organic and biological chemistry. An *s*-orbital is spherical with the nucleus at the center, a *p*-orbital is dumbbell-shaped , four of the five *d*-orbitals are cloverleaf shaped . The fifth *d*-orbital is shaped like an elongated dumbbell with a doughnut around its middle . *f*-orbitals have more complex shapes. Orbitals are organized into different layers or electron shells.

A useful aid to describe the three-dimensional (3D) probability density distributions of the various orbitals in the hydrogen atom is the Born interpretation of wave functions (M. Born, 1882-1970). Accordingly, the probability densities of the hydrogen atom orbitals can be represented and plotted as spatial surfaces which encompass most of the electron probability.

The wave function itself has no physical significance. However, the square of the wave function, is a quantity that is related to probabilities. Probability density distributions based on are three-dimensional, and it is these three-dimensional regions that we mean when we refer to the shape of an orbital.

Each point P possess a value of the wave-function, but its coordinates can be expressed in spherical or cartesian (orthogonal) coordinates
In the case of the wave function $\psi(\boldsymbol{P})$, spherical coordinates are preferable since they allow $\psi(\boldsymbol{P})$ to be factorized into the product :

$$\psi(\boldsymbol{P})=\psi(r,\varphi,\theta)=R(r)Y(\varphi,\theta)$$

of the radial function $R(r)$ and of the angular function $Y(\theta,\varphi)$.

Angular wave function (table 1.1) are better suited to describe the 'shape' of the orbital, in a general sense, without a specific scale factor, whilst on the contrary radial part is able to fix the 'scale factor' of the same orbital. In this way one can plot the two different quantities and focus on each of the two concepts.

no.	*l*	*m*		Polar coordinates	*no.*	*l*	*m*		Polar coordinates
1	0	0	s		9	2	-2	d(xy)	
2	1	0	p(z)		10	3	0	$f(z^3)$	
3	1	1	p(x)		11	3	1	$f(xz^2)$	
4	1	-1	p(y)		12	3	-1	$f(yz^2)$	
5	2	0	$d(z^2)$		13	3	2	$f(zx^2 -zy^2)$	
6	2		d(xy)		14	3	-2	f(xyz)	
7	2	-1	d(yz)		15	3	3	$f(x^3 -3xy^2)$	
8	2		$d(x^2-y^2)$		16	3	-3	$f(3yx^2-y^3)$	

Table 1.1 - angular wave functions $Y_{l,m}(\varphi,\theta)$ of hydrogen-like atoms

The first step for the graphics is to define a *meshgrid* of points, a job which is done by these following three lines of code. The first two lines create two row vectors (*theta, psi*) by mean of the function *linspace* . 60 elements are stuffed in *theta* , equally spaced starting from -π (or *pi* in Octave) up to π; 20 are stored in *phi* , from 0 to 2π.

```
theta = linspace(-pi,pi,60);
phi = linspace(0,2*pi,20);
[th1,ph1] = meshgrid(theta,phi);
```

The *meshgrid* function then generates two bidimensional arrays, *th1* and *ph1* , both 20 x 60 (20 rows , 60 columns), which are the basis for the following plotting function.

These arrays are someway redundant, indeed in *th1* all the colums have 20 equal elements and in *ph1* all the rows have 60 equal elements. In this fashion however for each point P (x_i , y_j) with $0 \le i \le 20$ and $0 \le j \le 60$ holds x_i = *th1(i, j) ;* y_i = *ph1(i, j)* . This greatly speeds up following calculations.

Thereafter spherical coordinates are transformed in cartesian, with the function *sph2cart* .The inputs theta, phi, and R must be the same shape, therefore R is defined before, with the command *R=ones(20,60)* . It produces a 20x60 array (R) with all elements equal

to unity. Phi describes the angle relative to the positive x-axis, (or azimuth angle), theta is the angle relative to the xy-plane (therefore the elevation angle is 90°-phi,) and R is the distance to the origin (0, 0, 0), set to unity

```
[x,y,z] = sph2cart(ph1,pi/2 - th1,R);
surf(x,y,z);axis([-3 3 -3 3 -2 2]);
```

In such a way now x,y,and z are three 20x60 matrices with everything is needed to plot in a cartesian reference (defined by *axis* instruction) the orbital surface.

In some instances, the resolution for phi angle has to be increased, so as to obtain a smoother picture (see figure captions below).

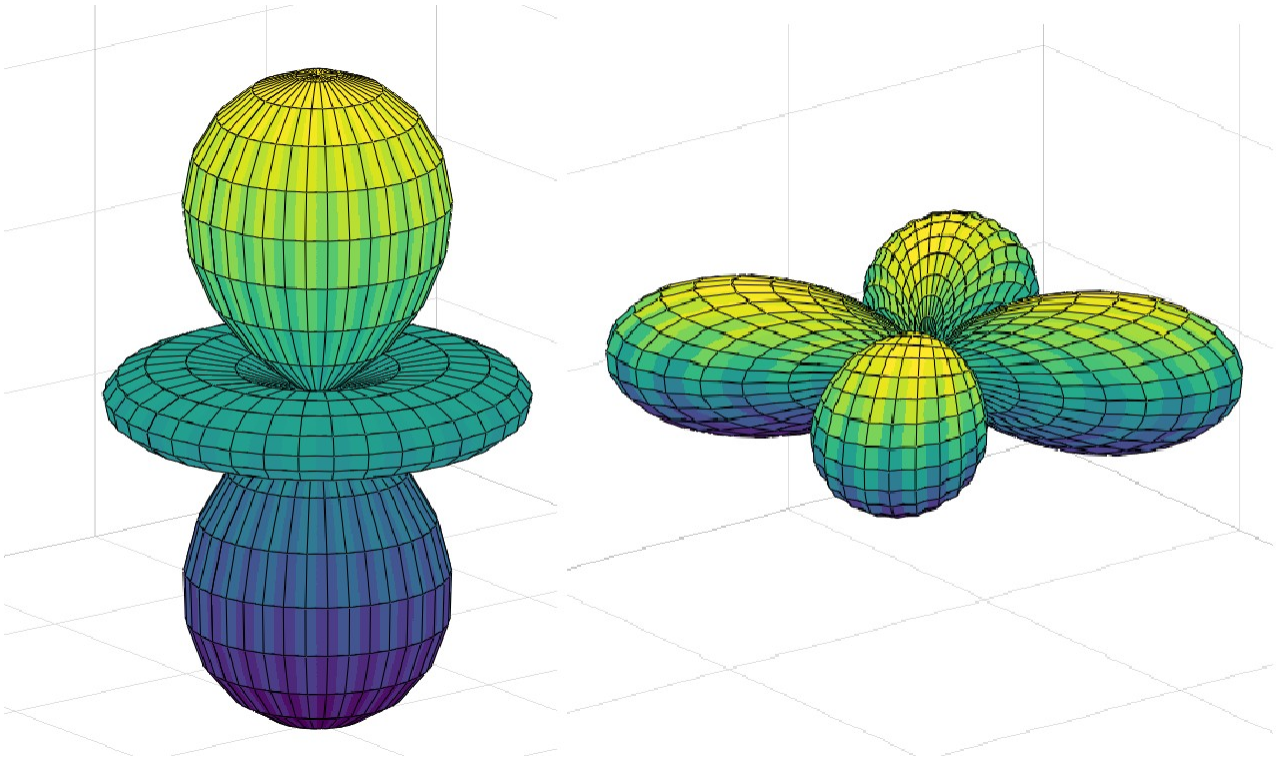


Fig. 1.1 3d(z^2) left and 3d (xy) right. The meshgrid for the left orbital is 20x60 and for the right 40x60.

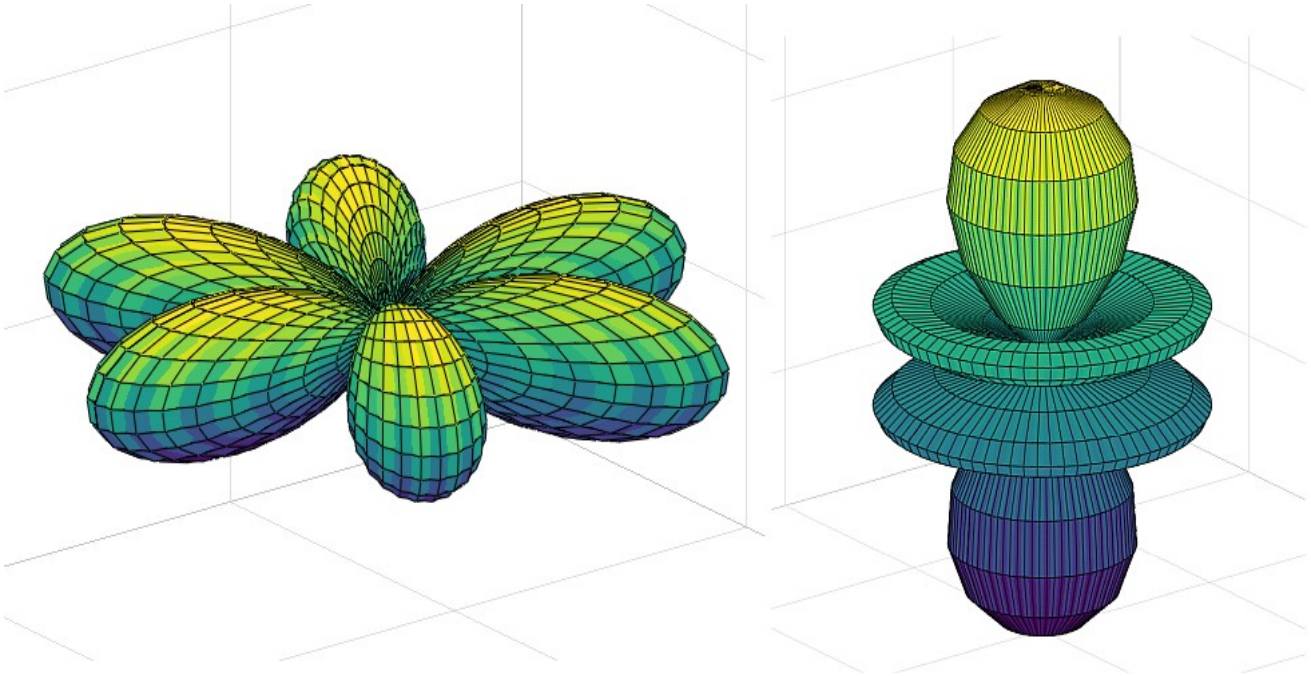


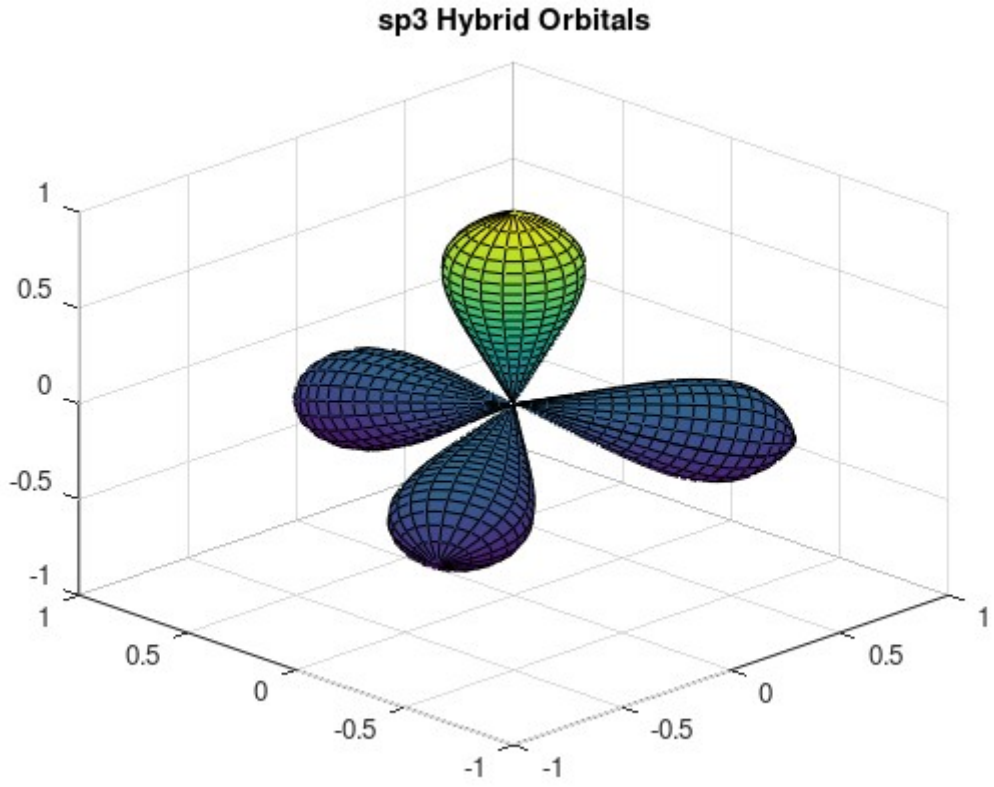
Fig. 1.2 4f(x^3-3xy^2) left and 4f (z^3) right. The meshgrid is 60x60 for the left orbital and 40x60 for the the right.

```
clear;clc;R = ones(60);
orbital = 10;                 % <--- user choice
theta = linspace(-pi,pi,60); % <--- user meshgrid theta
phi = linspace(0,2*pi,40);   % <--- user meshgrid phi
[th1,ph1] = meshgrid(theta,phi);

% Choose orbital type among those available.
switch orbital
  % 1s orbital
  case 1; T1 ='1s orbital'
  % 2p orbitals
  case 2; R = abs(sin(th1).*cos(ph1));T1 ='2p(x) orbital'
  case 3; R = abs(sin(th1).*sin(ph1));T1 ='2p(y) orbital'
  case 4; R = abs(cos(th1));T1 ='2p(z) orbital'
  % 3d orbitals
  case 5; R = abs(3*cos(th1).^2 - 1);T1 ='3d(z^2) orbital'
  case 6; R = abs(sin(th1).^2.*cos(2*ph1));T1 ='3d(x^2-y^2) orbital'
  case 7; R = abs(sin(th1).^2.*sin(2*ph1));T1 ='3d(xy) orbital'
  case 8; R = abs(sin(th1).*cos(th1).*cos(ph1));T1 ='3d(xz) orbital'
  case 9; R = abs(sin(th1).*cos(theta).*cos(ph1));T1='3d(yz) orbital'
  % 4f orbitals in standard set
see--->https://winter.group.shef.ac.uk/orbitron/atomic_orbitals/4f/index.html
  case 10; R = abs(5*cos(th1).^3 - 3*cos(th1));T1 ='4f(z^3) orbital'
  case 11; R = abs((5*cos(th1).^2 - 1).*sin(th1).*cos(ph1));T1 ='4f(xz^2)
orbital'
  case 12; R = abs((5*cos(th1).^2 - 1).*sin(th1).*sin(ph1));T1 ='4f(yz^2)
orbital'
  case 13; R = abs(sin(th1).^2.*cos(th1).*sin(2*ph1));T1 ='4f(xyz) orbital'
  case 14; R = abs(sin(th1).^2.*cos(th1).*cos(2*ph1));T1 ='4f(zx^2 - zy^2)
orbital'
  case 15; R = abs(sin(th1).^3.*cos(3*ph1));T1='4f(x^3 - 3xy^2) orbital'
  case 16; R = abs(sin(th1).^3.*sin(3*ph1));T1='4f(3yx^2 - y^3) orbital'
end

% Transforms spherical to cartesian coordinates.
% elevation angle is 90°-theta, azimuth angle equals phi.
```

```
[x,y,z] = sph2cart(ph1,pi/2 - th1,R);
surf(x,y,z);axis([-2 2 -2 2 -2 2]);
title(T1);
```

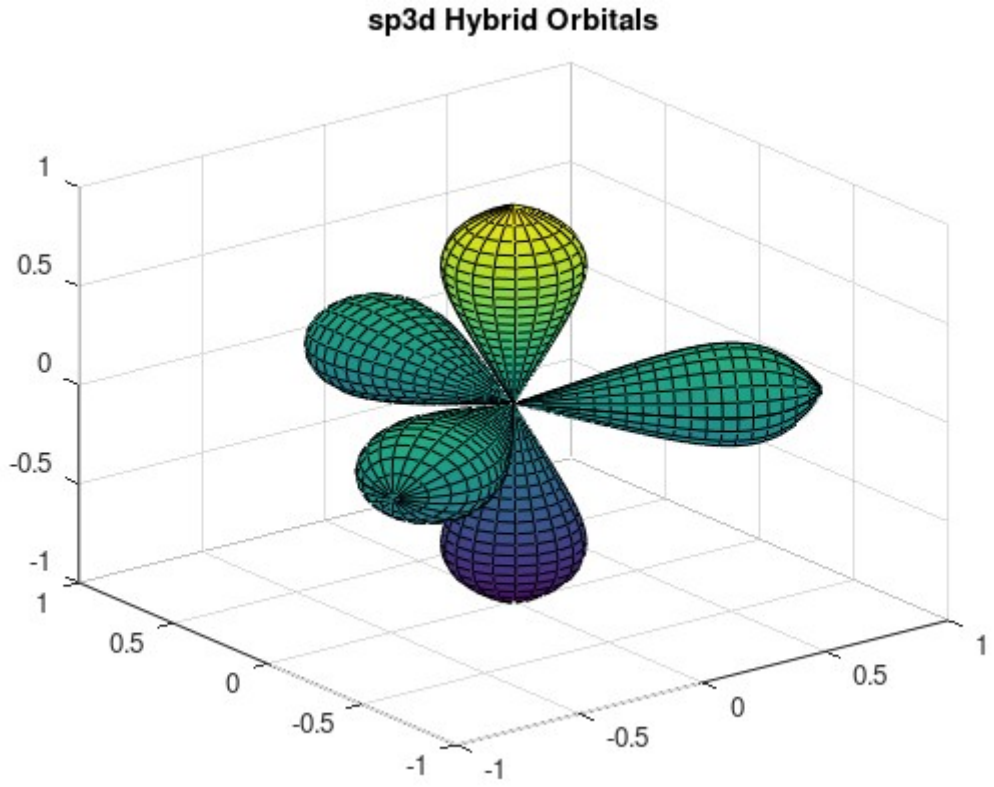
1.2 Hybrid orbitals

In chemistry, orbital hybridization is obtained by mixing atomic orbitals to form new hybrid orbitals (with different energies, shapes, etc., than the component atomic orbitals) suitable for the pairing of electrons to form chemical bonds in valence bond theory. For example, in a carbon atom which forms four single bonds the valence-shell s orbital combines with three valence-shell p orbitals to form four equivalent sp3 mixtures which are arranged in a tetrahedral arrangement around the carbon to bond to four different atoms. Hybrid orbitals are useful in the explanation of molecular geometry and atomic bonding properties and are symmetrically disposed in space. Usually hybrid orbitals are formed by mixing atomic orbitals of comparable energies.



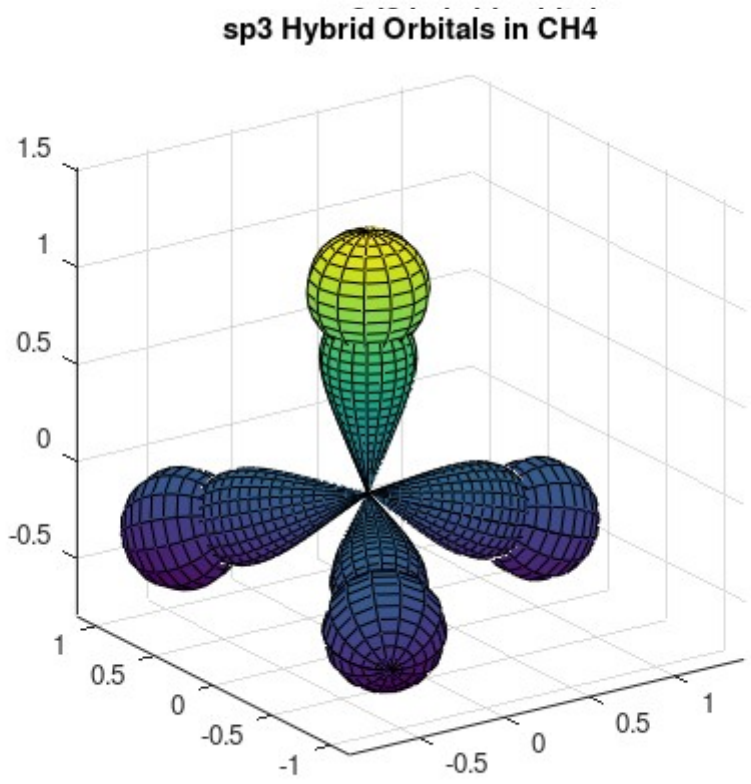

```
clear;clc
t = 0:0.1:2;
r = 0.2*(t - 0.03*t.^6) ;
[X,Y,Z] = cylinder(r);
% Plot the cylinder with the base centered at the origin.
surf(X,Y,Z)
hold on
Y1 = Y.*(-0.334) + Z.*0.943;
Z1 = Z.*(-0.334) - Y.*0.943;
surf(X,Y1,Z1)
X2 = X.*(-0.5) + Y1.*0.866;
Y2 = Y1.*(-0.5) - X.*0.866;
surf(X2,Y2,Z1)
X3 = X.*(-0.5) + Y1.*(-0.866);
Y3 = Y1.*(-0.5) - X.*(-0.866);
surf(X3,Y3,Z1)
title('sp3 Hybrid Orbitals')
```

Fig. 1.3 sp^3 hybrid orbitals formation (with default meshgrid)




```
clear;clc
t = 0:0.1:2;
r = 0.2*(t - 0.03*t.^6) ;
[X,Y,Z] = cylinder(r);
% Plot the cylinder with the base centered at the origin.
surf(X,Y,Z)
hold on
surf(X,Y,-Z)
Y1 = Z;
Z1 = -Y;
surf(X,Y1,Z1)
X2 = X.*(-0.5) + Y1.*0.866;
Y2 = Y1.*(-0.5) - X.*0.866;
surf(X2,Y2,Z1)
X3 = X.*(-0.5) + Y1.*(-0.866);
Y3 = Y1.*(-0.5) - X.*(-0.866);
surf(X3,Y3,Z1)
title('sp3d Hybrid Orbitals')
```

Fig. 1.4 sp^3d hybrid orbitals formation (with default meshgrid)



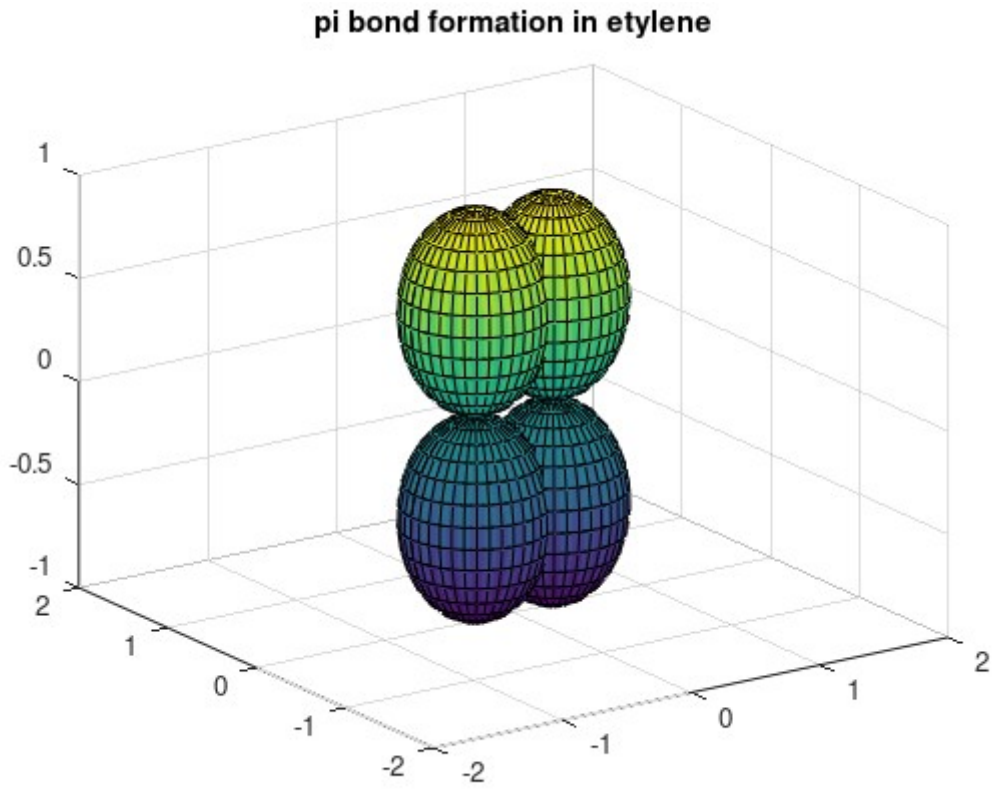

```
clear;clc
t = 0:0.1:2;
r = 0.2*(t - 0.028*t.^6) ;
[X,Y,Z] = cylinder(r);
% Plot the cylinder with the base centered at the origin.
surf(X,Y,Z)
title('sp3d2 hybrid orbitals');hold on
surf(Z,X,Y)
surf(Y,Z,X)
surf(X,Y,-Z)
surf(-Z,X,Y)
surf(Y,-Z,X)
```

Fig. 1.5 sp^3d^2 hybrid orbitals formation (with default meshgrid)

1.3 Molecular orbitals

A molecular orbital diagram, or MO diagram, is a qualitative descriptive tool explaining chemical bonding in molecules in terms of molecular orbital theory in general and the linear combination of atomic orbitals (LCAO) method in particular. A fundamental principle of these theories is that as atoms bond to form molecules, a certain number of atomic orbitals combine to form the same number of molecular orbitals, although the electrons involved may be redistributed among the orbitals.

Linear combination of atomic orbital can be pictured in octave by simple ovelapping of atomic orbitals. As an example molecular π orbitals in ethylene are portrayed in the following figure, whilst in fig. 1.7 the complete overlapping among 1s hydrogen orbitals and hybrid sp^3 of carbon in methane is to be seen.




```
clear;clc;R = ones(20,60);
theta = linspace(-pi,pi,60);
phi = linspace(0,2*pi,20);
[th1,ph1] = meshgrid(theta,phi);
R = abs(cos(th1));T1 ='2p(z) orbital'
% Transforms spherical to cartesian coordinates.
% elevation angle is 90°-theta, azimuth angle equals phi.
[x,y,z] = sph2cart(ph1,pi/2 - th1,R);
surf(x+0.3,y,z);hold on;surf(x-0.3,y,z); % <--- duplicate surf function
axis ([-2 2 -2 2 -1 1]);view(3);
title('pi bond formation in etylene');
```

```
clear;clc
t = 0:0.1:2;
r = 0.2*(t - 0.03*t.^6) ;
[X,Y,Z] = cylinder(r);
[x, y, z] = sphere (14);
x = 0.3*x;y = y*0.3;z = 0.3*z + 1.05;
% Plot the modified cylinder with the base centered at the origin.
surf(X,Y,Z); hold on;
surf(x,y,z);
T = rotz(30)*rotx(-109);
X1 = T(1,1)*X + T(1,2)*Y + T(1,3)*Z;
Y1 = T(2,1)*X + T(2,2)*Y + T(2,3)*Z;
Z1 = T(3,1)*X + T(3,2)*Y + T(3,3)*Z;
x1 = T(1,1)*x + T(1,2)*y + T(1,3)*z;
y1 = T(2,1)*x + T(2,2)*y + T(2,3)*z;
z1 = T(3,1)*x + T(3,2)*y + T(3,3)*z;
surf(X1,Y1,Z1);surf(x1,y1,z1);
T = rotz(120);
X2 = T(1,1)*X1 + T(1,2)*Y1;
Y2 = T(2,1)*X1 + T(2,2)*Y1;
x2 = T(1,1)*x1 + T(1,2)*y1;
y2 = T(2,1)*x1 + T(2,2)*y1;
surf(X2,Y2,Z1);surf(x2,y2,z1);
X3 = T(1,1)*X2 + T(1,2)*Y2;
Y3 = T(2,1)*X2 + T(2,2)*Y2;
x3 = T(1,1)*x2 + T(1,2)*y2;
y3 = T(2,1)*x2 + T(2,2)*y2;
surf(X3,Y3,Z1);surf(x3,y3,z1);
axis equal;
title('sp3 Hybrid Orbitals in CH4')
```

Chapter 2. Periodic table of the elements

2.1 Introduction

The basis of the periodic table is the electron configurations of the elements, which can be used for a discussion of some of properties of elements, including atomic radii, ionization energies, and electron affinities. These atomic properties also form the rationale basis of the chemical bonding.

In 1869, Dmitri Mendeleev observed that when the elements are arranged in order of increasing atomic mass, certain sets of properties recur periodically. Therefore he arranged and grouped together similar elements on the periodic table.

The rationale behind is that the physical and chemical properties of an element are determined largely by its electron configuration, particularly that of the valence (outermost) electronic shell. Adjacent members of a series of main-group elements in the same period (such as P, S, and Cl) have significantly different properties because they differ in their valence-electron configurations.

Mendeleev periodic table consisted of 8 groups, but most modern periodic tables are arranged in 18 groups of elements. The vertical groups bring together elements with similar properties. The horizontal periods of the table are arranged in order of increasing atomic number from left to right. The groups are numbered at the top, and the periods at the extreme left. The first two groups are the s block and the last six groups the p block together constitute the main-group elements. Because they come between the s block and the p block, the d block elements are known as the transition elements. The f block elements, sometimes called the inner transition elements, would extend the table to a width of 32 members if incorporated in the main body of the table.

The table would generally be too wide to fit on a printed page, and so the f block elements are extracted from the table and placed at the bottom. The 15 elements following barium are called the lanthanides, and the 15 following radon are called the actinides.

Through a color scheme of the periodic table, one evidences that the majority of the elements are metals (orange) and that non-metals (blue) are confined to the right side of the table. The noble gases (purple) are treated as a special group. Metals and nonmetals are often separated by a stairstep diagonal line, and several elements near this line are often called metalloids

(green). Metalloids are elements that look like metals and in some ways behave like metals but also have some nonmetallic properties.

2.2 3D-plot of electronegativity

Electronegativity describes an atom's ability to attract the electrons shared with other atoms during the formation of one or more chemical bonds. Many electronegativity scales have been developed in so far, one of the most used in general chemistry textbooks is the Pauling's scale. The values have been transferred in a text file, leving out those for which the value is not available. The picture of the 3d-trend can be obtained in a very simple manner, employing some of the Octave capabilities.

The *importdata* function has the ability to work with a wide variety of data. In our case the text file 'electronegativity.txt' is loaded into a cell. The text file is imported with the \n (new line) option, so as every line becomes now a single cell element. This cell has 94 elements, each consisting in a string of characters, containing the atomic number, atomic symbol, name of the element and finally its electronegativity. (the elements with no value available are omitted). The second step is now to extract the four fields from each element. This is performed by the function *strsplit* ; having no option, the space character (ASCII 32) is used as a delimiter.

```
A = importdata("electronegativity.txt","\n");
ln = length(A)
for i=1:ln
  b = strsplit(A{i});
  nAt = str2num(b{1});eNeg = str2num(b{4});
  .......
endfor
```

As a general rule, electronegativities decrease from top to bottom in a group and increase from left to right in a period of elements, in order to plot this in a 3-D fashion, we must fist introduce the function *patch* . This function creates a patch object in the current axes with vertices at locations (*x,y,z*) and of color defined by *FaceColor* argument. In our case the object is a quadratic prism , whose vertices are numbered according to the figure 2.1.

The function requires two input matrices, one for the vertices (*v2*) and another for the faces (*f2*). The elements in *v2* are the eight coordinated of vertices, whose connectivity to form faces is described in *f2* (see fig. 2.1). So as to simplify the script, only the visible faces are created.

```
v2 = [x1 y1 0;x2 y1 0;x2 y2 0;x1 y2 0;x1 y1 h1;x2 y1 h1;x2 y2 h1;x1 y2 h1];
f2 = [3 4 8 7;1 4 8 5;5 6 7 8];
patch('Faces',f2,'Vertices',v2,'FaceColor',[ (h1*0.25) (1-h1*0.25) 0]);view(3);
```

If needed for having a view from every angle, the complete set of faces can be plotted, and now *f2* becomes :

```
f2 = [3 4 8 7;1 4 8 5;1 2 6 5;2 3 7 6;5 6 7 8];
```

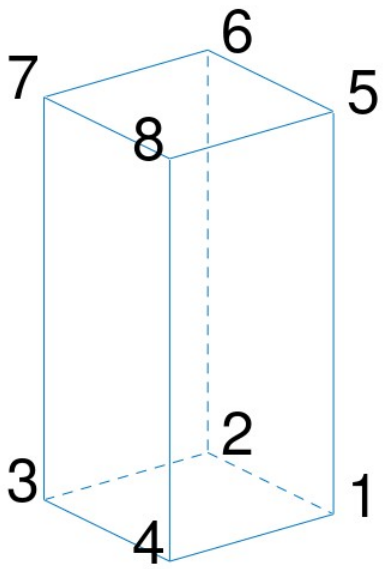
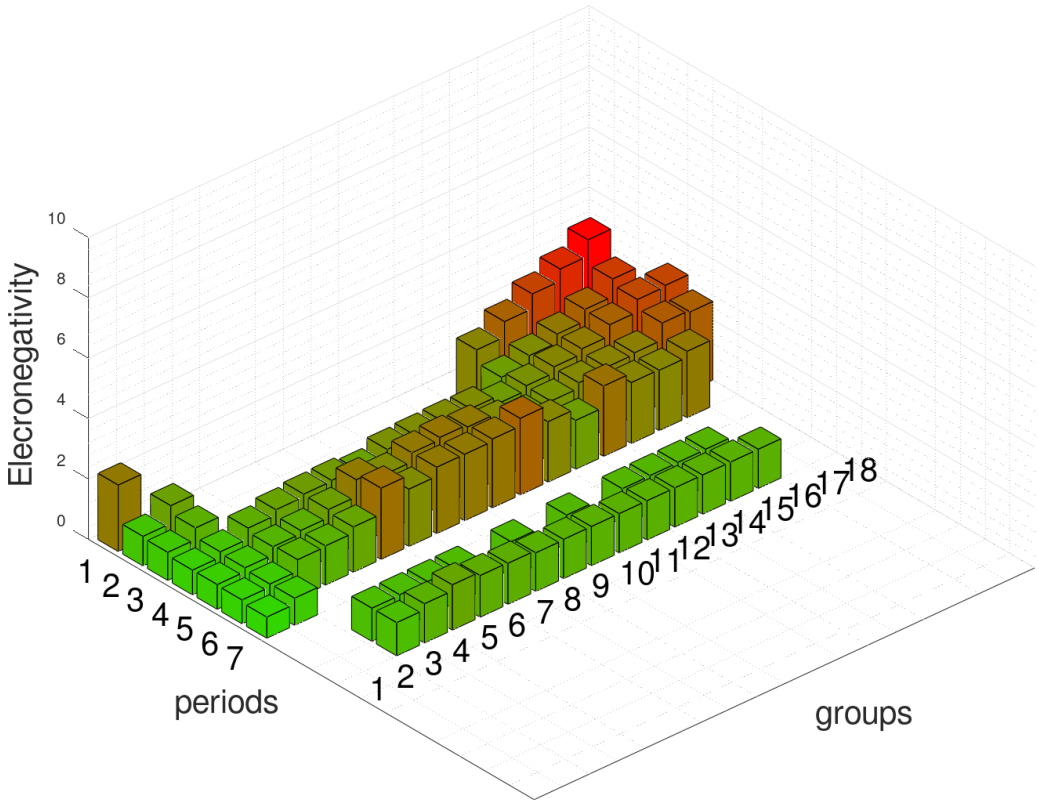


Figure 2.1 Quadratic prims and numbering scheme.

The complete script is listed below, while the auxiliary data (electronegativity.txt) required by the script is to be found in this same web site in text format. It is downloaded by the '*urlwrite*' command into a local file in the octave working directory, to be used immediatly by the following program lines. Once the program is run for the first time, the .txt file remains , so as it is unnecessary to reload it. To do this the line beginning with 'urlwrite' command can be inactivated, transforming it into a remark (adding the octave remark symbol '%' in the script)

It is loaded with '*importdata*' function and the '\n' option, so as each line is translated a single string. Each single string is then split with '*strsplit*' function and further analysed, in order to extract information about atomic number and electronegativity. Array '*AtNo*' contains the positions of the different prism on to the x,y plane for all the elements, from atomic number 1 (hydrogen) to atomic number 118 (oganesson). As not all of the 118 elements have an electronegativity value, the program script only accounts for those listed in the auxiliary data. To the missing elements a blank space is assigned.

The resulting picture is shown in fig 2.2



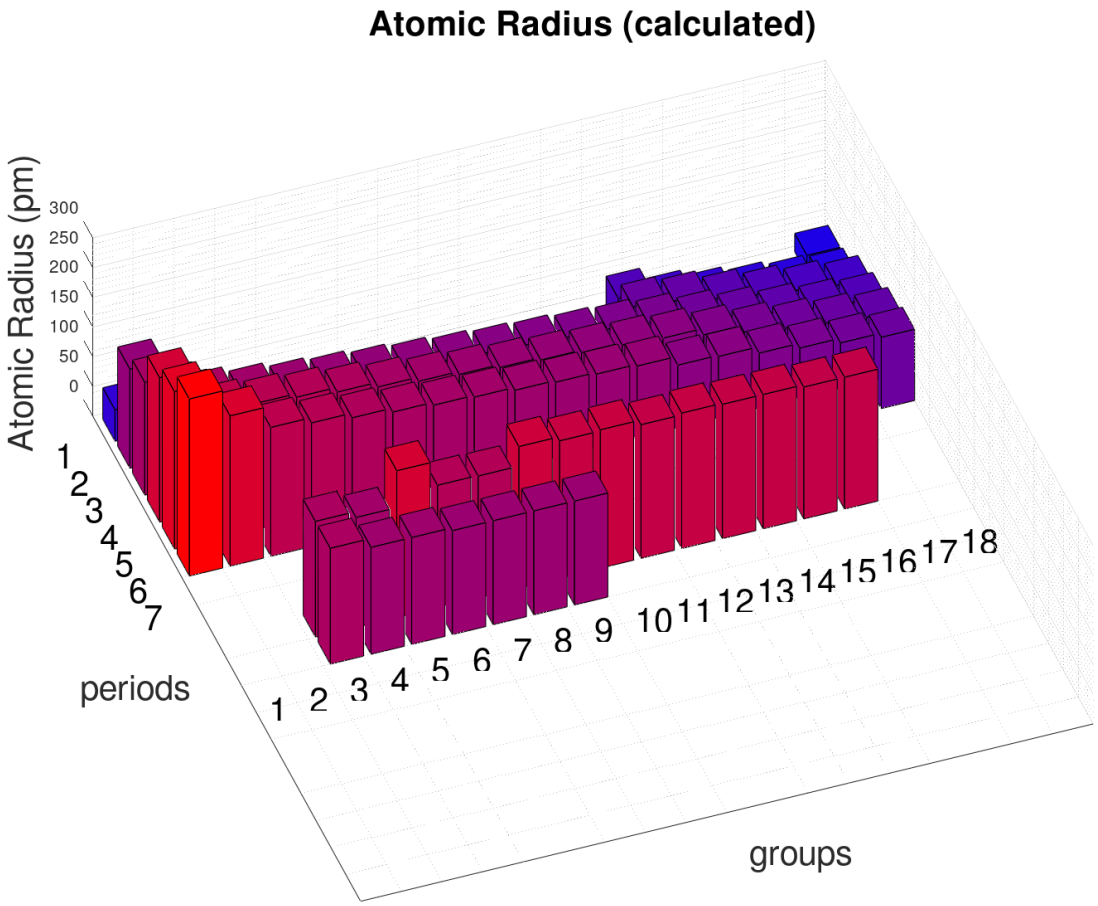

```
Clear all;clc;format short;format compact;
d = 0.18; % clearance

figure(1,'position',[100 100 1000 800]);
axis([0 18 0 18 0 10]);axis on;grid on;grid minor on;xticks(20);yticks(20);

AtNo = [0 18;17 18;0 17;1 17;12 17;13 17;14 17;15 17;16 17;17 17;... % x(i),y(i)
        0 16;1 16;12 16;13 16;14 16;15 16;16 16;17 16;...
        0 15;1 15;2 15;3 15;4 15;5 15;6 15;7 15;8 15;9 15;10 15;11 15;12 15;13 15;14 15;15 15;16 15;17 15;...
        0 14;1 14;2 14;3 14;4 14;5 14;6 14;7 14;8 14;9 14;10 14;11 14;12 14;13 14;14 14;15 14;16 14;17 14;...
        0 13;1 13;2 10;3 10;4 10;5 10;6 10;7 10;8 10;9 10;10 10;11 10;12 10;13 10;14 10;15 10;...
        2 13;3 13;4 13;5 13;6 13;7 13;8 13;9 13;10 13;11 13;12 13;13 13;14 13;15 13;16 13;17 13;...
        0 12;1 12;2 9;3 9;4 9;5 9;6 9;7 9;8 9;9 9;10 9;11 9;12 9;13 9;14 9;15 9;...
        2 12;3 12;4 12;5 12;6 12;7 12;8 12;9 12;10 12;11 12;12 12;13 12;14 12;15 12;16 12;17 12];
% The following line is to be used once. When the file id downloaded, inactivate the line by adding % (remark
symbol)
f = urlwrite('www.molecularmodels.eu/electronegativity.txt','electronegativity.txt');

A = importdata("electronegativity.txt","\n");
ln = length(A)
for i=1:ln
  b = strsplit(A{i});
  nAt = str2num(b{1});eNeg = str2num(b{4});h1 = eNeg;
  x1 = AtNo(nAt,1) + d;y1 = AtNo(nAt,2) - d;x2 = x1 + 1 - d;y2 = y1 -1 + d;
  v2 = [x1 y1 0;x2 y1 0;x2 y2 0;x1 y2 0;x1 y1 h1;x2 y1 h1;x2 y2 h1;x1 y2 h1];
  f2 = [3 4 8 7;1 4 8 5;5 6 7 8];
  patch('Faces',f2,'Vertices',v2,'FaceColor',[ (h1*0.25) (1-h1*0.25) 0]);view(3);
endfor

xlabel("groups",'FontSize',20);ylabel("periods",'FontSize',20);zlabel("Elecronegativity",'FontSize',20);
title("Electronegativity , Pauling scale",'FontSize',20);

for i=0.5:17.5
  text(i,7,num2str(i+0.5),'FontSize',20);
endfor
for i=18:-1:12
  text(-1.3,i-1,num2str(19-i),'FontSize',20);
endfor
```

Fig. 2.2 Tridimensional periodic table of electronegativity

2.3 Tridimensional plot of atomic radius

The atomic radius of a chemical element is a measure of the size of its atoms, usually the mean or typical distance from the center of the nucleus to the boundary of the surrounding shells of electrons. Since the boundary is not a well-defined physical entity, there are various non-equivalent definitions of atomic radius. Four widely used definitions of atomic radius are: Van der Waals radius, ionic radius, metallic radius and covalent radius. Typically, because of the difficulty to isolate atoms in order to measure their radii separately, atomic radius is measured in a chemically bonded state; however theoretical calculations are of course simpler when considering atoms in isolation.

Again, the auxiliary data (atomicRadius.txt) required by the script is to be found in this same web site in text format. It is downloaded by the '*urlwrite*' command into a local file in the octave working directory, to be used immediatly by the following program lines. Once the program is run for the first time, the .txt file remains , so as it is unnecessary to reload it. To do this the line beginning with 'urlwrite' command can be inactivated, transforming it into a remark (adding the octave remark symbol '%' in the script). The listing of the atomic radii is computed from theoretical models, as published by Enrico Clementi and others in 1967. The values are in picometres (pm). The octave script is very similar to the preceding one, so it isn't listed here but in appendix.



Chapter 3. Molecular rendering (or graphics)

Molecular graphics (MG) is the discipline and philosophy of studying molecules and their properties through graphical representation. It might be considered as an alternative to the construction of physical real models.

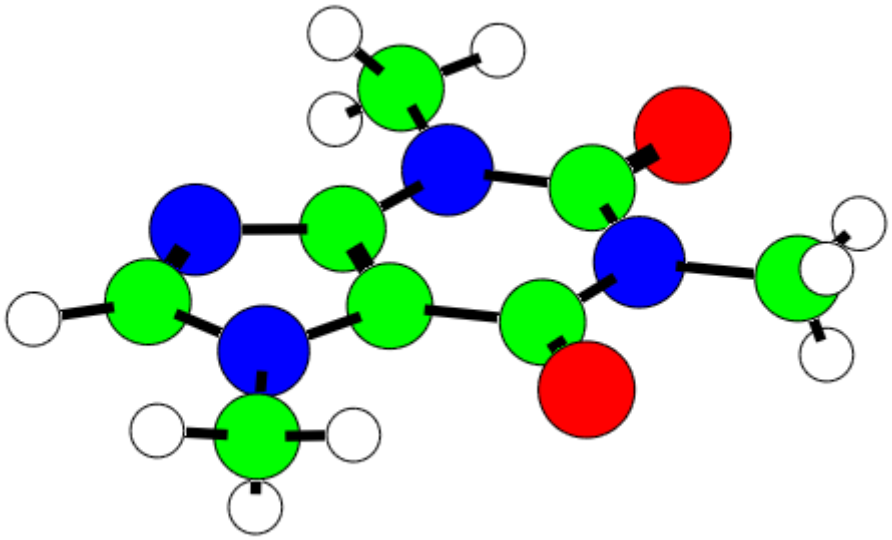


Fig. 3.1 caffeine (or theine) molecule. Red atom is oxygen, green carbon, blue nitrogen, and finally white hydrogen.

```
clear all;clc;format short;format compact;
figure(1,'position',[100 100 1000 800]);

function lineBond (c1,c2,r2,bondOrder)
  r3 = r2./100; % bring r2 in scale with axis
  dist = sqrt((c2(1)-c1(1))^2 + (c2(2)-c1(2))^2 + (c2(3)-c1(3))^2);
  x1 = (c2(1)*r3(1) - c1(1)*r3(1) + c1(1)*dist)/dist;
  y1 = (c2(2)*r3(1) - c1(2)*r3(1) + c1(2)*dist)/dist;
  z1 = (c2(3)*r3(1) - c1(3)*r3(1) + c1(3)*dist)/dist;
  x2 = (c1(1)*r3(2) - c2(1)*r3(2) + c2(1)*dist)/dist;
  y2 = (c1(2)*r3(2) - c2(2)*r3(2) + c2(2)*dist)/dist;
  z2 = (c1(3)*r3(2) - c2(3)*r3(2) + c2(3)*dist)/dist;
  line([x1,x2],[y1,y2],[z1,z2],'Color','black','LineWidth',1 + bondOrder*3);
endfunction

% insert here the file name here below
S = importdata("caffeine.txt","\n"); % line separator is \n (new line, ascii code
10)
% it may be downloaded from   'cactus.nci.nih.gov/chemical/structure/xxxxxxx/file?
format=sdf&get3d=True'
% where xxxxxxx is the name of the molecule

dN = S{4};dN = strtrunc(dN,20);
X = str2num(dN);
nAtom = X(1);
nBond = X(2);j = 0;
for i = 5:(5 + nAtom - 1)
  dM = S{i};dN = strtrunc(dM,30);dA = substr(dM,32,1)
  X = str2num(dN);
  ++j;x(j) = X(1);y(j) = X(2);z(j) = X(3);
  switch dA
    case "H";r(j) = 20;cr(j) = 'w';
```

```
    case "O";r(j) = 36;cr(j) = 'r';
    case "C";r(j) = 32;cr(j) = 'g';
    case "N";r(j) = 34;cr(j) = 'b';
    case "S";r(j) = 40;cr(j) = 'y';
  endswitch
endfor  % coordinates loaded in x() y() z() and radius in r()

j = 0;
for i = (5 + nAtom):(5 + nAtom + nBond -1)
  ++j;dM = S{i};X = str2num(dM);conn(j,1) = X(1);conn(j,2) = X(2);conn(j,3) =
X(3);
endfor  % connectivity loaded in conn()

for i = 1 : nAtom
  plot3(x(i),y(i),z(i),"ok",'markersize',r(i),'markerfacecolor',cr(i)); hold on;
endfor

axis([-5 5 -5 5 -5 5]);axis off
% axis on;grid on;grid minor on;  % choose if display axis
view(3);

for i = 1 : nBond
  h = conn(i,1);k = conn(i,2);
  lineBond([x(h),y(h),z(h)],[x(k),y(k),z(k)],[r(h),r(k)],conn(i,3));
endfor
```

The file containing the atomic coordinates and connectivity is conform to the standard molfile (MDL molfile) format. It can be downloaded from the web from many different repository; one of the most visited is '*cactus.nci.nih.gov/chemical/structure/xxxxxxx/file?format=sdf&get3d=True*' where *xxxxxxx* is the name of the molecule.

An example of such file format follows (caffeine molecule portrayed in fig. 3.1)

```
C8H10N4O2
APtclcactv11192101413D 0   0.00000     0.00000

 24 25  0  0  0  0  0  0  0  0999 V2000
    1.3120   -1.0479    0.0025 N   0  0  0  0  0  0  0  0  0  0  0  0
    2.2465   -2.1762    0.0031 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.7906    0.2081    0.0010 C   0  0  0  0  0  0  0  0  0  0  0  0
    2.9938    0.3838    0.0002 O   0  0  0  0  0  0  0  0  0  0  0  0
    0.9714    1.2767   -0.0001 N   0  0  0  0  0  0  0  0  0  0  0  0
    1.5339    2.6294   -0.0017 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.4026    1.0989   -0.0001 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.4446    1.9342   -0.0010 N   0  0  0  0  0  0  0  0  0  0  0  0
   -2.5608    1.2510   -0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -2.2862   -0.0680    0.0015 N   0  0  0  0  0  0  0  0  0  0  0  0
   -3.2614   -1.1612    0.0029 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.9114   -0.1939    0.0014 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.0163   -1.2853   -0.0022 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.4380   -2.4279   -0.0068 O   0  0  0  0  0  0  0  0  0  0  0  0
    3.2697   -1.8004    0.0022 H   0  0  0  0  0  0  0  0  0  0  0  0
    2.0830   -2.7828    0.8938 H   0  0  0  0  0  0  0  0  0  0  0  0
    2.0821   -2.7846   -0.8862 H   0  0  0  0  0  0  0  0  0  0  0  0
    2.6223    2.5703   -0.0019 H   0  0  0  0  0  0  0  0  0  0  0  0
    1.1987    3.1611   -0.8923 H   0  0  0  0  0  0  0  0  0  0  0  0
    1.1990    3.1632    0.8877 H   0  0  0  0  0  0  0  0  0  0  0  0
   -3.5520    1.6797   -0.0001 H   0  0  0  0  0  0  0  0  0  0  0  0
   -3.5037   -1.4333   -1.0244 H   0  0  0  0  0  0  0  0  0  0  0  0
```

```
   -2.8389   -2.0244    0.5173 H   0  0  0  0  0  0  0  0  0  0  0  0
   -4.1672   -0.8395    0.5168 H   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  1  3  1  0  0  0  0
  3  4  2  0  0  0  0
  3  5  1  0  0  0  0
  5  6  1  0  0  0  0
  5  7  1  0  0  0  0
  7  8  1  0  0  0  0
  8  9  2  0  0  0  0
  9 10  1  0  0  0  0
 10 11  1  0  0  0  0
 10 12  1  0  0  0  0
  7 12  2  0  0  0  0
 12 13  1  0  0  0  0
  1 13  1  0  0  0  0
 13 14  2  0  0  0  0
  2 15  1  0  0  0  0
  2 16  1  0  0  0  0
  2 17  1  0  0  0  0
  6 18  1  0  0  0  0
  6 19  1  0  0  0  0
  6 20  1  0  0  0  0
  9 21  1  0  0  0  0
 11 22  1  0  0  0  0
 11 23  1  0  0  0  0
 11 24  1  0  0  0  0
M  END
$$$$
```

The file was downloaded with .txt extension in the working directory. Double bonds are thicker than single bonds.

Chapter 4. Reaction balancing

4.1 Introduction

The numeric coefficients assigned to a balance equation express the relative amounts of atoms or molecules which take part to the specific reactions. More precisely tells us the proportions of moles involved therein. This fact derives directly from *Proust law* (J. L. Proust, 1750-1826) or *law of definite proportions*. It states that a given chemical compound always contains its component elements in a fixed ratio, irrespectively of the preparation route.
The quantitative aspect, dealing with mass and volume relations among reactants and products, is termed *stoichiometry*

The mole (symbol: mol) is the base unit of amount of substance in the International System of Units (SI). It is defined as exactly 6.02214076×10^{23} elementary entities ("particles"), which may be atoms, molecules, ions, or electrons. This definition was adopted in November 2018, revising the previous definition that specified one mole as the amount of substance in 12 grams of carbon-12 (^{12}C), an isotope of carbon. Thus, for example, one mole of water (H_2O) contains 6.02214076×10^{23} molecules, whose total mass is about 18.015 grams and the mean mass of one molecule of water is about 18.015 amu (atomic mass unit, or dalton), roughly a combined atomic mass number of 18.

Therefore, coefficients in a balanced chemical reaction can be interpreted as the relative number of moles, molecules or volume (if reactants are gases) involved in the reaction. If the chemical equation is properly balanced, the coefficients are expressed by minimal integer numbers, and they determine the quantitative aspects of the same chemical reaction. We seek a procedure that correctly establishes the coefficients of the equation; that is, the arithmetical multipliers of the different chemical units therein expressed by minimal integers, thereby providing the ultimate stoichiometric relationship.

4.2 The nullspace method

The lack of a general method in the reaction balance process causes a widespread use of iterative numerical procedures, their goal being to reach the final set of coefficients by balancing in each step a selected element or functional group.

A general method can be found with the aid of linear algebra , and many reactions, notably *redox reactions*, readily lend themselves to a convenient algebraic algorithm based on the *nullspace (or kernel) concept*. The algorithm bypasses both the oxidation number procedure or other iterative methods in a straighforward manner, and applies to complex reactions as well.

The principle of mass balance is based on the law of conservation of mass, i.e., the number of atoms of an element remains constant in a chemical reaction (we exclude nuclear reactions, where new elements may be originated). Balancing each element in a reaction imposes a set of simple equations in which the unknowns are the stoichiometric coefficients of each substance in reactants and products, provided that the chemical formula of each compound possess well established integer atomic indices, like H_2SO_4.

The whole procedure is better explained by the aid of the classical redox reaction of iron(II) sulfate with potassium permanganate in acidic water solution:

$$FeSO_4 + KMnO_4 + H_2SO_4 \longrightarrow MnSO_4 + Fe_2(SO_4)_3 + K_2SO_4 + H_2O$$

The law of mass conservation requires that the amount of the elements taking part to the reaction must equal in reactants (left side) and products (right side). This requires a set of 7 multiplicative stoichiometric coefficients (*a,b,c,d,e,f,g*) , like :

$a\ FeSO_4 + b\ KMnO_4 + c\ H_2SO_4 \longrightarrow d\ MnSO_4 + e\ Fe_2(SO_4)_3 + f\ K_2SO_4 + g\ H_2O$

These coefficients are tied by 7 linear equations, each of them imposing the mass conservation of a single element. For example, the iron mass conservation requires that $a = 2e$, or $a - 2e = 0$ and so on for sulfur (S), oxygen (O), potassium (K),manganese (Mn) and hydrogen (H).

In the following octave script the 7 linear equation are collected in matrix **A** , which isn't a square matrix (six rows and 7 columns) being 6 the elements to balance and 7 the coefficients. The octave function *null* (**A**) is therefore able to find a vector in the nullspace of the same matrix **A .** Once the vector **Z** is found it might not be expressed by the lowest integers, so other five lines are in charge of finding the lowest integer stoichiometric coefficients.

```
% Oxidation of iron(II)sulphate with potassium permanganate.
% Coefficients finding.
clear;clc;
disp ('a FeSO4 + b KMnO4 + c H2SO4 ==> d MnSO4 + e Fe2(SO4)3 + f K2SO4 + g H2O');
A = [1  0  0  0  -2  0  0;  % Fe balance
     1  0  1  -1 -3 -1  0;  % S balance
     4  4  4 -4 -12 -4 -1;  % O balance
     0  1  0  0  0  -2  0;  % K balance
     0  1  0  -1  0  0  0;  % Mn balance
     0  0  2  0  0   0 -2]; % H balance

Z = null(A);    % Find a vector in the null space of matrix A; System Ax = b is
undetermined
x1 = min(Z);Z1 = Z./x1;
for i = 1:10    % Find the lowest stoichiometric coefficient from 1 to 10
    intZ = round(i*Z1);
    if (sum(abs(i*Z1 - intZ)))<1e-6 ;break;endif  % lowest coefficient ok
endfor

T1 = [num2str(intZ(1)),' FeSO4  + ',num2str(intZ(2)),' KMnO4  +
',num2str(intZ(3)),' H2SO4  ==> ',num2str(intZ(4)),' MnSO4  + '];
T1 = [T1,num2str(intZ(5)),' Fe2(SO4)3  + ',num2str(intZ(6)),' K2SO4  +
',num2str(intZ(7)),' H2O'];
disp('when balanced looks like : ');
disp(T1);
```

The program output pops up in the command window as follows:

```
a FeSO4 + b KMnO4 + c H2SO4 ==> d MnSO4 + e Fe2(SO4)3 + f K2SO4 + g H2O
when balanced looks like :
10 FeSO4  + 2 KMnO4  + 8 H2SO4  ==> 2 MnSO4  + 5 Fe2(SO4)3  + 1 K2SO4  + 8 H2O
```

4.3 Mass conservation between reactants and products

The correctness of the 7 coefficient might be simply tested by the elements equality, indeed 10 iron atoms are counted on the left side as well on the right, as well as 2 Mn, 2 K, 16 H, 18

S, and 80 O. The mass conservation requires some more effort, being necessary to compute the molecular masses of all the compounds. The following table is self-explaining.

Formula	Mass of one mole (g)	Stech. Coeff.	Mass in reactant (g)	Mass in product (g)
$FeSO_4$	151.91	10	1519.10	-
$KMnO_4$	158.03	2	316.06	-
H_2SO_4	98.08	8	784.64	-
$MnSO_4$	151.00	2	-	302.00
$Fe_2(SO_4)_2$	399.88	5	-	1999.4
K_2SO_4	174.26	1	-	174.26
H_2O	18.02	8	-	144.16
Total amount (g)			2619.8	2619.82

The slight difference of 0.02 g between the total mass of reagent and that of product originates from round off errors in molecular masses.

4.4 Other two reaction examples, with scripts and results

Oxidation of silver sulfide by nitric acid in HCl aqueous solution

```
clear;clc;
% Find the coefficient of a redox chemical reaction.
disp ('a Ag2S + b HNO3 + c HCl ==> d S + e AgCl + f NO(gas) + g H2O');
% oxidation of silver sulphide with nitric acid.
A = [2  0  0  0  -1  0  0;  % Ag balance
     1  0  0  -1  0  0  0;  % S balance
     0  3  0  0   0  -1 -1; % O balance
     0  1  0  0  0  -1  0;  % N balance
     0  0  1  0  -1  0  0;  % Cl balance
     0  1  1  0  0   0 -2]; % H balance

Z = null(A);     % Find a vector in the null space of matrix A; System Ax = b is
undetermined
x1 = min(Z);Z1 = Z./x1;
for i = 1:10    % Find the lowest stoichiometric coefficient from 1 to 10
    intZ = round(i*Z1);
    if (sum(abs(i*Z1 - intZ)))<1e-6 ;break;endif  % lowest coefficient ok
endfor

T1 = [num2str(intZ(1)),' Ag2S  + ',num2str(intZ(2)),' HNO3  + ',num2str(intZ(3)),'
HCl  ==> ',num2str(intZ(4)),' S  + '];
T1   =   [T1,num2str(intZ(5)),'   AgCl      +   ',num2str(intZ(6)),'   NO(gas)      +
',num2str(intZ(7)),' H2O'];
disp('when balanced looks like : ');
disp(T1);
```

After execution, the command window is the following :

```
a Ag2S + b HNO3 + c HCl ==> d S + e AgCl + f NO(gas) + g H2O
when balanced looks like :
3 Ag2S  + 2 HNO3  + 6 HCl  ==> 3 S  + 6 AgCl  + 2 NO(gas)  + 4 H2O
```

Oxidation of hydrogen peroxide in acid aqueous solution

```
clear;clc;
% Find the coefficient of a redox chemical reaction.
disp ('a H2O2 + b KMnO4 + c H2SO4 ==> d H2O + e O2(gas) + f MnSO4 + g K2SO4');
% oxidation of hydrogen peroxide with potassium permanganate.
A = [0  1  0  0  0  0 -2;  % K balance
     0  1  0  0  0 -1  0;  % Mn balance
     2  4  4 -1 -2 -4 -4;  % O balance (total)
     0  0  1  0  0 -1 -1;  % S balance
     2  0  2 -2  0  0  0;  % H balance
     2  0  0  0 -2  0  0]; % O balance (redox partecipating)

Z = null(A);     % Find a vector in the null space of matrix A; System Ax = b is
undetermined
x1 = min(Z);Z1 = Z./x1;
for i = 1:10    % Find the lowest stoichiometric coefficient from 1 to 10
    intZ = round(i*Z1);
    if (sum(abs(i*Z1 - intZ)))<1e-6 ;break;endif  % lowest coefficient ok
endfor

T1   =   [num2str(intZ(1)),'   H2O2      +   ',num2str(intZ(2)),'   KMnO4      +
',num2str(intZ(3)),' H2SO4  ==> ',num2str(intZ(4)),' H2O  + '];
T1   =   [T1,num2str(intZ(5)),'   O2(gas)     +   ',num2str(intZ(6)),'   MnSO4     +
',num2str(intZ(7)),' K2SO4'];
disp('when balanced looks like : ');
disp(T1);
```

After execution, the command window is the following :

```
a H2O2 + b KMnO4 + c H2SO4 ==> d H2O + e O2(gas) + f MnSO4 + g K2SO4
when balanced looks like :
5 H2O2  + 2 KMnO4  + 3 H2SO4  ==> 8 H2O  + 5 O2(gas)  + 2 MnSO4  + 1 K2SO4
```

Othe examples can be found in ref. Mazza 2022

Chapter 5. Chemical kinetics

5.1 Introduction

Chemical kinetics , also known as reaction kinetics, is the branch of physical chemistry that is concerned with understanding the rates of chemical reactions. It is to be contrasted with thermodynamics, which deals with the energetics under which a process occurs but in itself tells nothing about its rate. Chemical kinetics includes investigations of how experimental conditions influence the speed of a chemical reaction and yield information about the reaction's mechanism and transition states, as well as the construction of mathematical models that also can describe the characteristics of a chemical reaction.

Time evolution of reactant and product concentration requires an appropriate model of the reactions in terms of Ordinary Differential Equations (ODE). Only simple linear equations (and in few cases also nonlinear equations) can be explicitly integrated so as to arrive at an expression relating well established thermodynamic parameters of the reactions (like activation energy, equilibrium constants) to time - varying concentrations. The alternative and generic approach is the numerical integration by means of appropriate ODE solvers.
More choices are offered by Octave, so that some of them are proposed here.

5.2 First-order reactions

A simple case of reaction is a first order irreversible(5.1), in which products cannot be converted back to the original reactants. In this case, the rate of the forward reaction decreases until all the reactants have been consumed, and the reaction terminates. On the contrary, in a single-step reversible reaction, the apparent rate of the forward reaction decreases together with the accumulation of the reaction products until a condition of dynamic equilibrium is finally established (5.2). At equilibrium, forward and backward reactions proceed at equal rates.

(5.1) $A \rightarrow B$ (5.2) $A \rightarrow B + A \leftarrow B = A \Leftrightarrow B$

The rate of a first-order reaction is proportional to the first power of the concentration of only one reactant. This means that the amount $d[A]$, which undergoes chemical change in the short time interval dt , depends only on the amount of A present at that instant. The rate expression which describes a first-order reaction is:

$\frac{-dA}{dt}=k_1[A]$ irreversible reaction

$\frac{-dA}{dt}=k_1[A]-k_2[B]$ $-\frac{dB}{dt}=k_2[B]-k_1[A]$ reversible reaction

Let us start with a first degree kinetic, irreversible case, using two different ODE solver, namely *lsode* (using Hindmarsh's ODE solver LSODE) and ode45 (using Dormand-Prince method of order 4).

The following script uses *lsode* solver with a rate constant 0.1 , initial concentration 4 (arbitrary unit) and 100 integration steps of 0.5 time unit each.

```
clear;clc;
function xdot = f (c, tspan)
  k1 = 0.1;
  xdot(1) = - k1*c(1);
endfunction
c0 = 4;tspan = linspace (0, 50, 100);
y = lsode (@f, c0, tspan);
plot(tspan,y,'b','LineWidth',2,tspan,c0-y,'r','LineWidth',2);grid on
xlabel('time, arbitrary units');ylabel('concentration');
title('first order irreversible reaction','FontSize',16);
legend('A','B');axis([0 50 0 5]);
```

It produces the following figure:

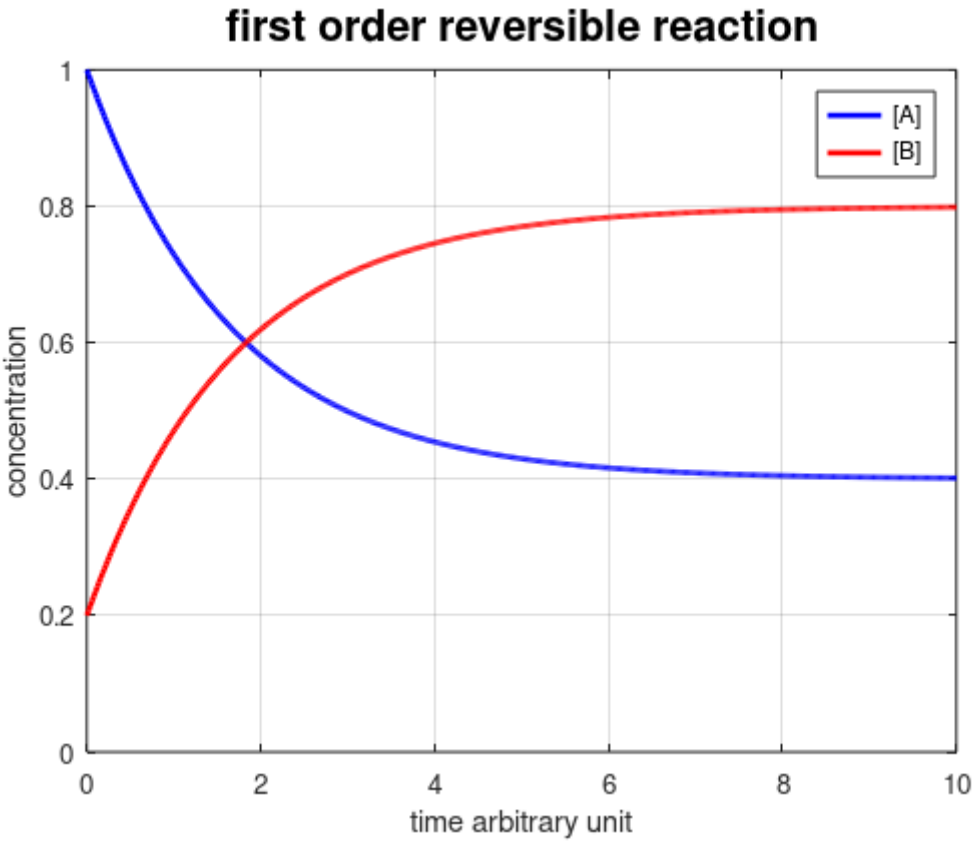
The following script uses *ode45* solver with a rate constant 0.06, initial concentration 0.05 and 30 integration steps of 1 time unit each

```
clear;clc;
k = 0.06;    %rate constant of reaction
timespan = (0:30);
A0 = 0.05;   % initial concentration
first = @(t,A) - k*A;
[t,A_calc] = ode45(first,timespan,A0);
plot(t,A_calc,'LineWidth',2);grid on
```

```
xlabel('time, arbitrary units')
ylabel('[A] concentration')
title('first order irreversible reaction')
```

It also produces a figure, very similat to the above, and therefore it isn't reported in here.

If we consider a reversible reaction, two rate constants are needed, namely k1 = 0.4 and k2 = 0.2. Their ratio defines the equilibrium constant of the reaction Keq = k1/k2 = 2. As seen from the figure below after a certain time the two concentrations converge to asymptotic values 0.8 and 0.4, that satisfy Keq = 0.8/0.4. *ode45* solver is employed, with 500 integration steps, from 0 to 10 time units, as defined by the function `tspan = linspace(0,10,500);`




```
clear all;clc;
function dC = reversible_A_B(t, C)
  % Rate constants.
  k1 = 0.4;
  k2 = 0.2; % therefore Keq = k1/k2 = 2
  % Rate laws.
  r1 = k1*C(1); % C(1) = [A]
  r2 = k2*C(2); % C(2) = [B]
  % Mass balances.
  dCA  = -r1 + r2;
  dCB = - r2 + r1;
  % Assign output variables.
  dC(1,:) = dCA;
  dC(2,:) = dCB;
endfunction
C0 = [1,0.2];                                % Define initial concentrations.
tspan = linspace(0,10,500);                  % Define time span.
[t, y] = ode45(@reversible_A_B, tspan, C0);  % Run ODE solver.
% plot
c1 = y(:,1);c2 = y(:,2);
plot(t,c1,'b','LineWidth',2,t,c2,'r','LineWidth',2);grid on;axis([0 10 0 1]);
xlabel('time arbitrary unit');ylabel('concentration');legend('[A]','[B]');
title('first order reversible reaction','fontsize',16);
```

5.3 Second- and higher order reactions

When two reactants A and B interact in such a way that the rate of reaction is proportional to the first power of the product of their respective concentrations, the compounds are said to undergo a second order reaction. The following (5.3) is the irreversible case

(5.3) $A + B \rightarrow C$

whilst (5.4) is the reversible one

(5.4) $A + B \rightarrow C \;+\; C \rightarrow A + B \;=\; A + B \Leftrightarrow C$

Higher order reactions are also possible, when three reactants produce one or more products, for example. Consecutive reactions like (5.5) can also be encountered

(5.5) $A + B + C \rightarrow D$

In every case, once reaction constants and start concentrations are assigned, is possible to write a function containing reaction rates in differential form, to be integrated by one ODE solver. The following script refers to reaction (5.5) and it is an hybrid one, as user can choose between two ODE solvers (*lsode,* `green lines` *or ode45* `red lines`).

Each solver requires an appropriate syntax, while the remaining part of the script is valid for both. Other reaction are possible, by modifying the lines inside the function xdot= f () (`grey lines`) according to the proposed mechanism.

In the following, r1,r2,r3 are the rate of the three reactions :

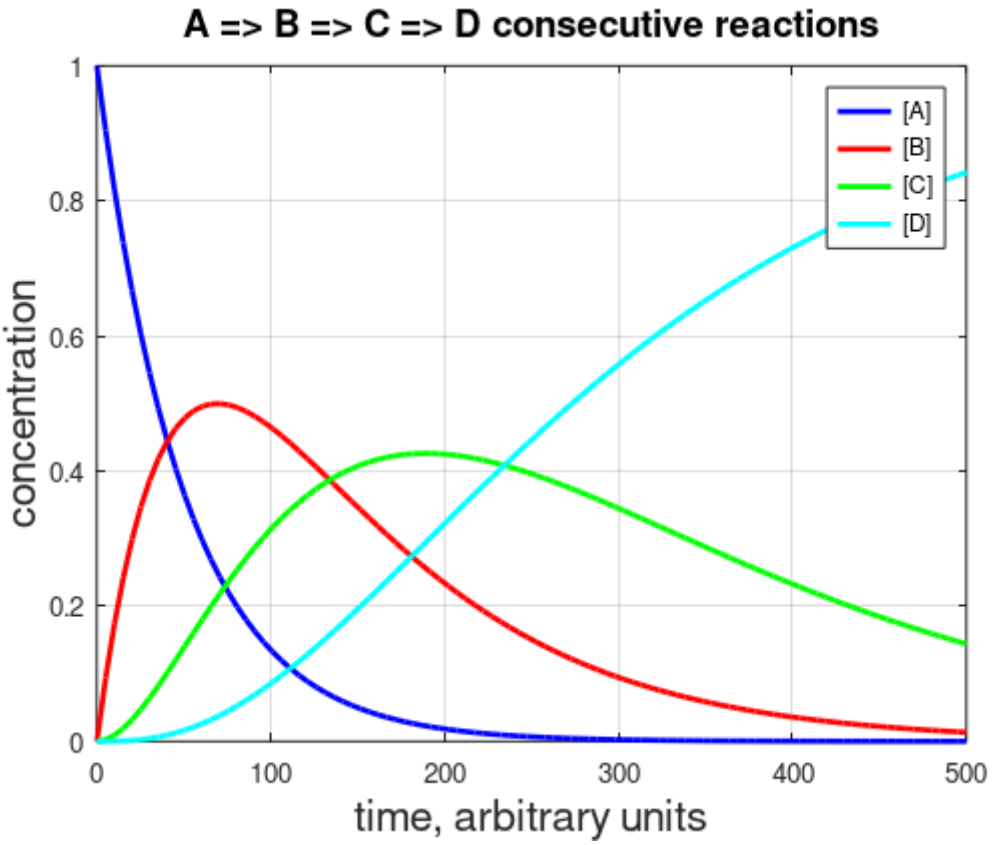
```
A ==> B    ,k1
B ==> C    ,k2
C ==> D    ,k3
```

which in turn are proportional to the respective concentrations c(1),c(2),c(3) multiplied by k1,k2,k3. The change in the A,B,C concentrations is expressed by xdot(1),xdot(2),xdot(3) according to the overall reaction scheme.

```
% Consecutive Irreversible Reactions
% A ==> B    ,k1
% B ==> C    ,k2
% C ==> D    ,k3
clear;clc;
function xdot = f(c,tspan)
function xdot = f(t,c)

  k1 = 0.02;
  k2 = 0.01;
  k3 = 0.006;
  r1 = k1*c(1); % c(1) = [A]
  r2 = k2*c(2); % c(2) = [B]
  r3 = k3*c(3); % c(3) = [C]
  xdot(1) = -r1;
  xdot(2) = r1 - r2;
  xdot(3) = r2 - r3;
endfunction

c0 = [1,0,0];                       % Define initial concentrations.
tspan = linspace (0, 500, 100);     % Define tspan.
y = lsode (@f, c0, tspan);
[t, y] = ode45(@f, tspan, c0);
D = 1 - y(:,1) - y(:,2) - y(:,3); % [D] = 1 - [A] - [B] - [C]
% plot
plot(tspan,y(:,1),'b','LineWidth',2,tspan,y(:,2),'r','LineWidth',2,...
tspan,y(:,3),'g','LineWidth',2,tspan,D,'c','LineWidth',2);grid on;
xlabel('time, arbitrary
units','FontSize',16);ylabel('concentration','FontSize',16);
title('A => B => C => D consecutive reactions','FontSize',14);
```

```
legend('[A]','[B]','[C]','[D]');
```

The output is identical, whichever solver is used, and is shown in figure below:

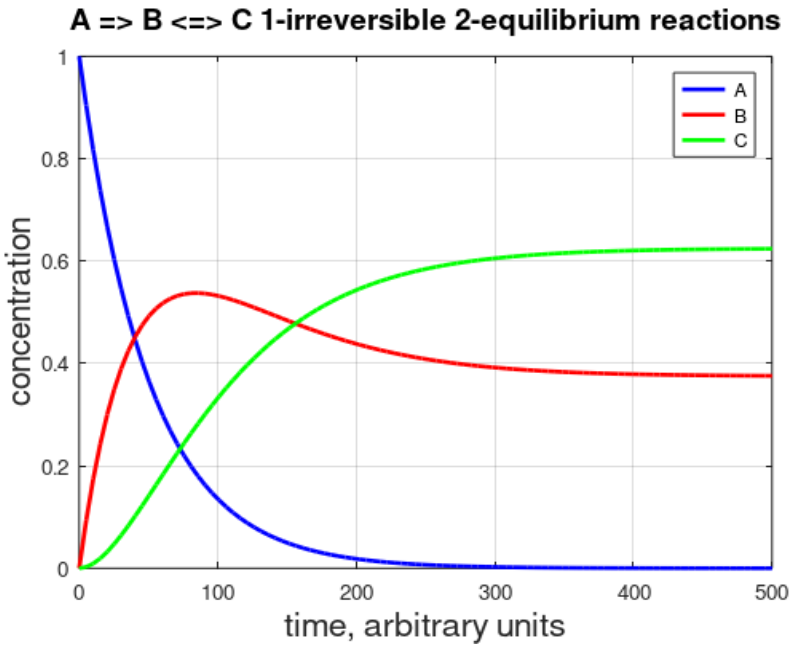


As a second example is reported the nine lines inside the function *xdot* for a coupled reactions set, the first being an irreversible one and the second a reversible (equilibrium) reaction, according to :

(5.6) $A \rightarrow B \;;\; B \Leftrightarrow C$

```
k1 = 0.02;
k2 = 0.01;
k3 = 0.006;
r1 = k1*c(1); % c(1) = [A]
r2 = k2*c(2); % c(2) = [B]
r3 = k3*c(3); % c(3) = [C]
xdot(1) = - r1;
xdot(2) = r1 - r2 + r3;
xdot(3) = r2 - r3;
```

starting from initial concentrations c0 = [1,0,0] the output is in figure below:



5.4 More complex and oscillating reactions

The iodine oscillating reaction, discovered by Briggs-Rauscher in 1973, also known as the oscillating clock, is one of the most common demonstrations of a chemical oscillating reaction.

Reactants are :

1) KIO3 (*potassium iodate*) + H2SO4 (*sulfuric acid*), aqueous solutions.

2) CH2(COOH)2 (*malonic acid*), aqueous solution.

3) H2O2 (*hydrogen peroxide*), aqueous solutions

When the three colorless solutions are mixed together, the color of the resulting mixture oscillates between clear, amber, and deep blue for about three to five minutes. The solution ends up into a blue and black mixture.

Smaller amounts of auxiliary reactants are needed, they are:

1) MnSO4 (*manganese sulfate*), in water solution where Mn^{++} ions act as a catalyst for the reactions.

2) Starch as color enhancer for I2 (*molecular iodine*) in solution.

The reaction encompasses different steps, its overall scheme can be simplified as follows (in ionic form):

(1) $IO3^- + 2\ H2O2 + H^+ + CH2(COOH)2 \rightarrow CHI(COOH)2 + 2\ O2 + 3\ H2O$

This first stage is the reduction of iodine in iodate ions from +5 to +1 oxidation state to HIO (*hypoiodous acid*) with a contemporary oxidation of 4 oxygen atoms in water peroxide from -1 to 0 and evolution of molecular oxygen gas.

Before reaction (1) is complete, newly produced HIO reacts with still present $IO3^-$ to produce molecular iodine and oxygen, the latter evolving again as a gas from solution.

(2) $IO3^- + 2\ H2O + H^+ \rightarrow HIO + 2\ O2 + 2\ H2O$

(3) $2\ IO3^- + 6\ H2O + 2\ HIO + 2\ H^+ \rightarrow 2\ I2 + 2\ O2 + 8\ H2O$

Iodine undergoes in water to a dismutation reaction :

(4) $I2 + H2O \Leftrightarrow HIO + I- + H+$

In acidic medium this equilibrium reaction is shifted to the left, therefore only small traces of I- ions are produce, which are however enough to trigger reaction (5),

(5) $CH2(COOH)2 + I2 + I^- \rightarrow CHI(COOH)2 + 2\ I^-$

Reaction (5) is the so called breeding reaction, indeed two iodide ions are produced from one. At the end hypoiodous acid and iodine are consumed to give as end product iodomalonic acid:

(6) $HIO + I^- + H^+ + 2CH2(COOH)2 \rightarrow 2\ CHI(COOH)2 + 2\ H2O$

Reaction (6) is the so called moderating reaction, as it controls reaction (5) by capturing I^- ions.

The complex reactions can be simplified by assuming that the starting reagent, *potassium iodate, malonic acid* and *sulfuric acid* all keep their concentration constant. This value is indicated as **C**. Water is considered to have constant concentration, as usual in diluted aqueous reactions.

By further indicating HIO = **X** ; I2 = **Y** and I^- = **Z**, the reactions (2),(3),(5),(6) can be resumed as:

C → **X**	k1	Being conc. of C kept constant this is a 'zero' order reaction
2 **X** → 2 **Y**	k2	second order
Y + **Z** → 2 **Z**	k3	second order
X + **Z** → End product	k4	second order

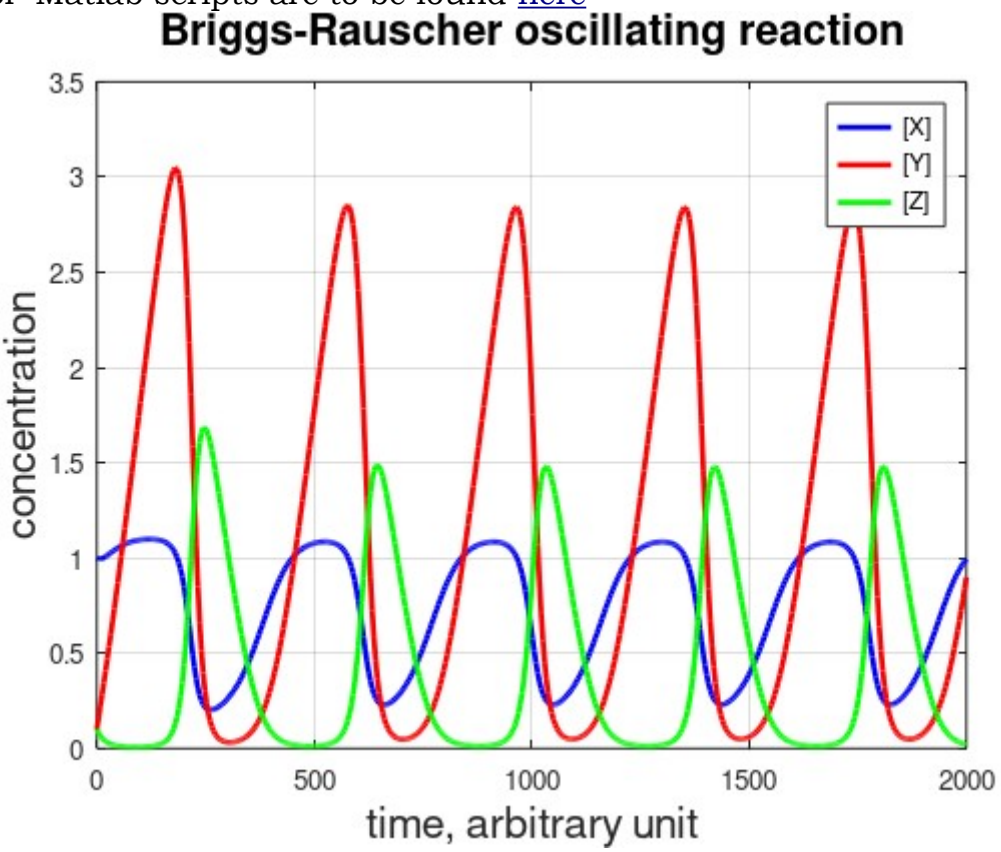
The four kinetic constants and initial concentration C are assigned in the first script lines of the function xdot(), evidenced in gray), to be integrated by the chosen ODE solver (in the example below *lsode*). The reaction rates (r1 to r4) are assigned according to the above mechanism and eventually mass balance of X,Y,Z is imposed.

```
clear;clc;
function xdot = f(c, tspan)
 % Rate constants.
 k1 = 0.02;k2 = 0.008;
 k3 = 0.04;k4 = 0.06;
 c_Init = 1;
 % Rate laws.
 r1 = k1*c_Init;          % c(1) = [X]
 r2 = k2*c(1)^2;          % c(2) = [Y]
 r3 = k3*c(2)*c(3);       % c(3) = [Z]
 r4 = k4*c(1)*c(3);
 % Mass balances and output assignement.
 xdot(1) = r1 - 2*r2 - r4;
 xdot(2) = 2*r2 - r3;
 xdot(3) = r3 - r4;
end
% Define initial concentrations.
c0 = [1,0.1,0.1];
tspan = linspace(0,2000,500);
y = lsode ("f", c0, tspan);
```

```
plot(tspan,y(':',1),'b','LineWidth',2,tspan,y(':',2),'r','LineWidth',2,tspan,y(':'
,3),'g','LineWidth',2);grid on;
xlabel('time,arbitrary  unit','FontSize',16);ylabel('concentration','FontSize',16);
legend('[X]','[Y]','[Z]');
title('Briggs-Rauscher oscillating reaction','FontSize',16);
```

The figure resulting from the orange part of the above script is shown here below.

A deeper discussion about modelling of chemical kinetics is contained in (Mazza 2022) and a wider choice of Matlab scripts are to be found here



5.5 First order reversible reaction with delayed addition of reactant: the CO2 - seawater kinetics

As explained in mode detail in §11.1 and §11.7 atmospheric carbon dioxide (CO2) reacts with seawater, which is, by its own nature, alkaline in character. The equilibrium constant for the equilibrium reaction :

$$CO_2(gas)+H_2O \Leftrightarrow CO_2(aq)+H_2O \Leftrightarrow H_2CO_3$$

is discusse in §11.7 and the scripts are in appendix.

Here we consider the reaction from a point of view of kinetics, therefore we are interested in a situation where the reaction has not yet reached its equilibrium condition. This happens because each year tons of CO2 are emitted in the atmosphere by the combustion of fossil fuels, thereby disturbing the pre-industrial equilibrium. The accepted value for this pre-industrial equilibrium is 280 ppm-CO2. As in any kinetic-controlled reaction we must consider the direct (from left to right) and inverse (from right to left) semi-reactions. The latter has however a nearly constant rate, being the concentration of carbonic acid [H2CO3] in seawater considered constant, at least in the decades time span. By accounting for the direct reaction

$$CO_2(gas)+H_2O \Rightarrow H_2CO_3$$

We can write :

$$direct\ \ reaction\ \ rate\ \ =\ \ \mathrm{v1} = \frac{d[CO_2]}{dt} = -k_1 \cdot [CO_2] \qquad [1]$$

and for inverse reaction

$$reverse\ \ reaction\ \ rate\ \ =\ \ \mathrm{v2} = \frac{d[H_2CO_3]}{dt} = k_2 \cdot [H_2CO_3] = constant\ value = k_3 \qquad [2]$$

The total reaction rate results from the sum (v1 + v2), which holds:

$$\frac{d[CO_2]}{dt} = v1 + v2 = k_3 - k_1 \cdot [CO_2] \qquad [3]$$

This is only a part of the process, because every year a certain known amount of anthropogenic CO_2 in emitted in the atmosphere. It is easy to show that 7.8 Gt-CO2 correspond to 1 ppmCO2 increment: the mass of the earth atmosphere is $5.15 \cdot 10^{18}$ kg (https://en.wikipedia.org/wiki/Atmosphere_of_Earth) while its the average molecular mass (dry air) is 28.946 g (same source). The number of moles of air will be:

$$moles\,of\ air\ =\ \frac{5.15 \cdot 10^{21} g}{28.946} = 1.779 \cdot 10^{20}\ \ mol$$

1 ppm of CO2 (molecular mass 44) will have a mass of:

$$1.779 \cdot 10^{20} \cdot 10^{-6} \cdot 44 = 7.8276 \cdot 10^{15} grams \approx 7.8 Gtonnes\, CO_2 \left(10^6 metric\, tons\, CO_2\right)$$

The CO2 output in the amosphere from the fossil fuels is reported by one of the many plot of OWID for the energy section (https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions) or from the data of the Global Carbon Budget (https://www.globalcarbonproject.org/carbonbudget/). Eq. [3] must be completed by the corresponding term $[CO_2]_{FOSSIL}$

$$\frac{d[CO_2]}{dt} = v1 + v2 + v3 = k_3 - k_1 \cdot [CO_2] + \frac{d[CO_2]_{FOSSIL}}{dt} \qquad [4]$$

In order to integrate this relationship numerically with a step of 1 year we first transform eq. [4] into a discrete, stepwise differential equation. Let us transform differential values in yearly variations first like:

$$\frac{d[CO_2]}{dt} \Rightarrow \Delta\, ppmCO_2 = ppmCO_2(i+1) - ppmCO_2(i) \qquad [2]$$

where $ppmCO_2(i)$ is the atmospheric CO2 concentration in ppmv (parts per million in volume) in the year considered, and $ppmCO_2(i+1)$ the same in the next year (both averaged from the monthly Mauna Loa values).

$$\Delta\, ppmCO_2 = k_3 - k_1 \cdot ppmCO_2(i) + \Delta\, ppmCO_2(FOSSIL) = k_3 - k_1 \cdot ppmCO_2(i) + 7{,}8 \cdot GtCO_2(FOSSIL)$$

By imposing :

$k_3 = k_1 \cdot ppmCO_2(equil.) = k_1 \cdot 280$

one obtains:

$\Delta\, ppmCO_2 = k_1 \cdot (280 - ppmCO_2(i)) + 7{,}8 \cdot GtCO_2(FOSSIL)$

$ppmCO_2(eq) = 280\ ppm$ is considered the standard pre-industrial value, equilibrated with carbonic acid in seawater. It can be adjusted , together with k_1 , by a best-fit procedure, as follows.

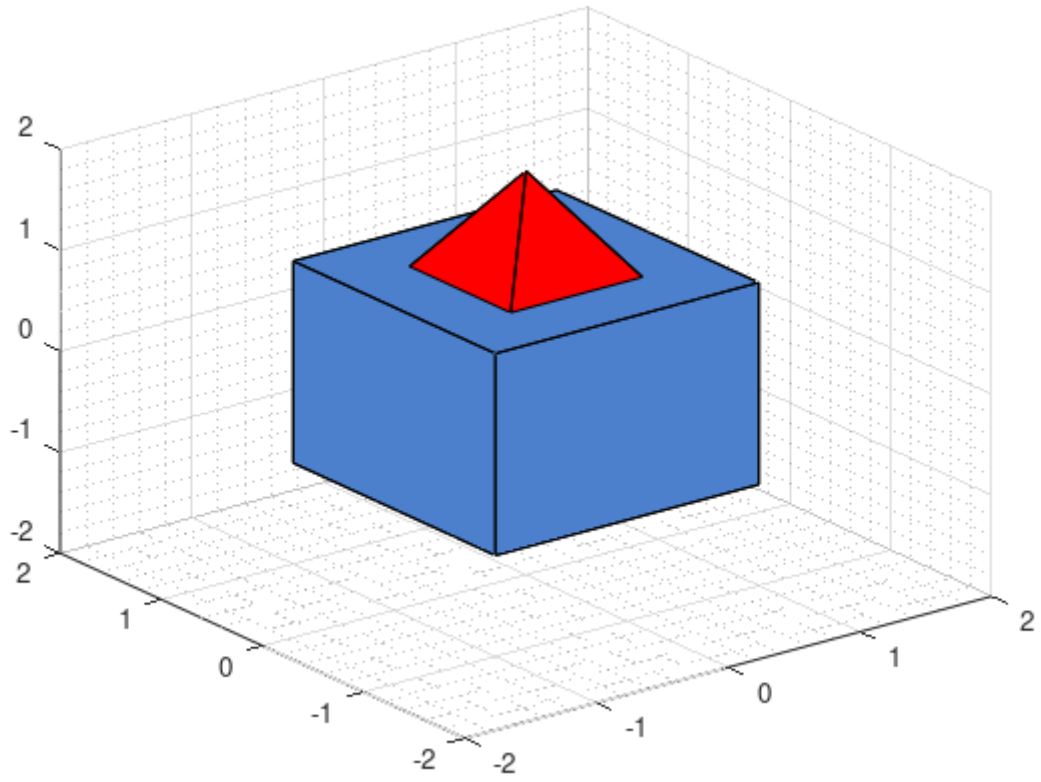
```
% 1 ppm = 7.8 Gt-CO2         1958-->2020
clear all;clc;format short;format compact;

global Ctot Cteor CO2
function z = sigma(p)
  global Ctot Cteor CO2;
  for i = 1:63       % kinetic machine
    Cteor(i+1) = Ctot(i) + Cteor(i) - p(2)*(Cteor(i) - p(1));
  endfor
  z = sumsq(Cteor(1:63) - CO2);
endfunction

% Coal/Oil/Gas emissions in metric tons CO2*10^9 (Gt-CO2)------------------------
C = csvread('co2-emissions-by-fuel-line.txt'); % (1958--->2020)
Ctot = C(:,7) + C(:,4) + C(:,8);
Ctot = Ctot./7.8e9;  % transformed into ppmCO2

% CO2 Mauna Loa  (1958-->2020)---------------------------------------------------
S = fileread('CO2_mm_mlo.txt');
a2 = index(S,' 1958   ');  % from march,1958
a3 = index(S,' 2021   ');  % to december,2020
M = S(a2:(a3-1));X = str2num(M);
CO2 = zeros(63,1);Cteor = zeros(64,1);
CO2(1) = mean(X(1:10,4));year(1) = X(1,1);
for i=2:63
  CO2(i) = mean(X((i*12-13):(i*12-2),4));
endfor
% Call to minimize sigma ---------------------------------------------------------
Cteor(1) = CO2(1);
q = fminunc ("sigma",[295,0.020]);disp(q); % start guess 295,0.020

for i = 1:63       % kinetic machine optimized  --> last check
  Cteor(i+1) = Ctot(i) + Cteor(i) - q(2)*(Cteor(i) - q(1));
endfor
z1 = sumsq(Cteor(1:63) - CO2)
% Graphics -----------------------------------------------------------------------
year = linspace(1958,2020,63)';
figure(1,'position',[200 100 700 500]);clf;
plot(year,Cteor(1:63),'r','linewidth',2);hold on;scatter(year,CO2,6,'g','filled');
axis([1957,2021,300,450]);grid on;grid minor on;
xlabel('year');ylabel('ppmCO2  in  atmosphere');title(['ppmCO2  1958-2020  -  kinetic
model  k1 = ',num2str(q(2)),...
'  ppmCO2(equil.) = ',num2str(q(1))])
```

References

2) La curva sigmoide e il grafico di Hubbert; Daniele Mazza, blogger.com, february 2019

https://www.blogger.com/blog/post/edit/9113085352551052888/995634914753494375

Chapter 6. Crystal structures

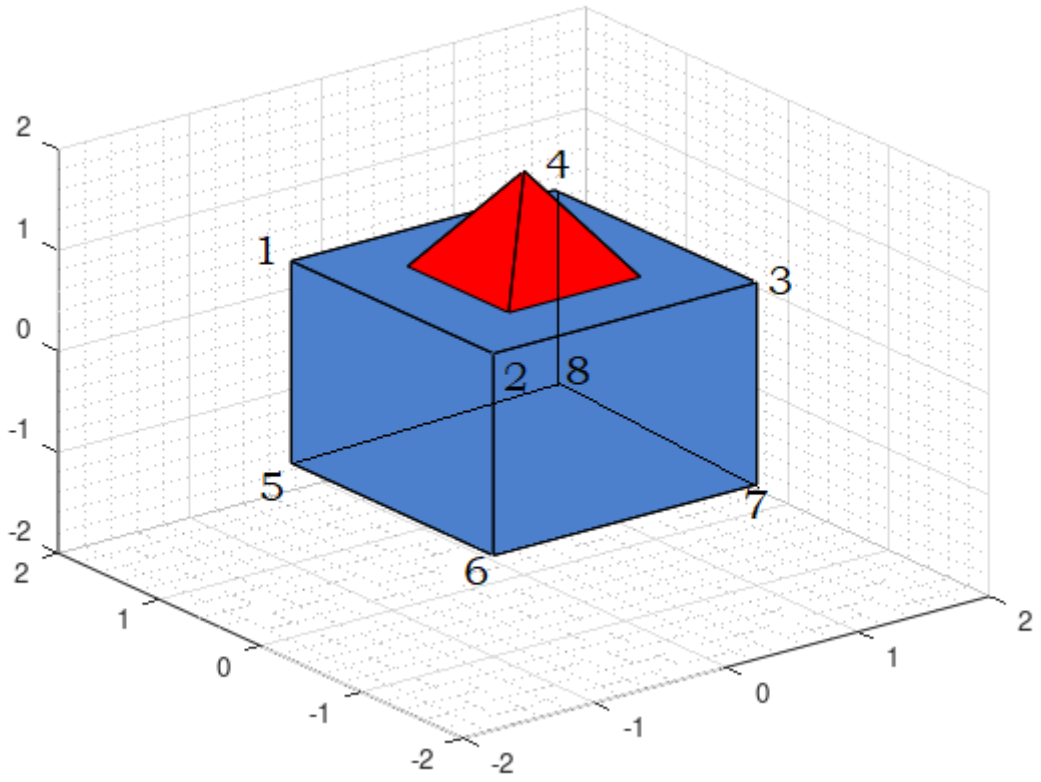


6.1 Introduction

Drawing a 3-D shape requires on Octave a few lines of code. Pictured here below is a blue cube stacked by a red pyramid on its top face, produced by the script that follows. One must supply the cartesian coordinates in a proper reference for the vertices of the cube (f1 array) and then in v1 array their connectivity. Then the powerful *patch* function is called with proper arguments, as shown.

```
clear all;clc;format short;format compact;
figure(1,'position',[100 100 600 450]);clf;
axis([-2 2 -2 2 -2 2]);axis on;grid on;grid minor on;
v1 = [-1 -1 -1;1 -1 -1;1 1 -1;-1 1 -1;-1 -1 1;1 -1 1;1 1 1;-1 1 1];
f1 = [1 2 6 5;2 3 7 6;3 4 8 7;1 4 8 5;5 6 7 8;1 2 3 4];
patch('Faces',f1,'Vertices',v1,'FaceColor',[.3 .5 .8]);
v2 = [-0.5 -0.5 1;0.5 -0.5 1;0.5 0.5 1;-0.5 0.5 1;0 0 2];
f2 = [1 2 5;2 3 5;3 4 5;4 1 5];
patch('Faces',f2,'Vertices',v2,'FaceColor','red');view(3);
```

In the sketch below is shown the numbering sequence for the cube verteces, an analogous scheme is used for the red pyramid.



6.2 How to extract the atomic coordinates for a crystal lattice

Open-access collection of crystal structures of organic, inorganic, metal-organic compounds and minerals, excluding biopolymers can be found at COD (Crystallograpy Open Database) web site. The crystal data are stored in the CIF format.

A CIF or Crystallographic Information File is the standard format for storing crystallographic structural data. CIF information has a specific structure or format that needs to be followed to allow crystallographic programs to read the file.

A short guide to CIF format can be found here.

A CIF file contains information about the crystal structure (such as unit cell values, atom names and their coordinates and any structural model quality indicators, e.g., R Factor) as well as any details of the diffraction experiment (such as temperature, pressure, experimental wavelength and the type and name of equipment used) and any data processing undertaken (such as the programs used to process the data).

The material that is included within a CIF varies and depends on what information the authors of the structure incorporated, as well as the technique and program used. Many crystallographic programs include most information automatically when the CIF is made.

As a matter of fact it is advisable to extract those lines from the CIF file which contains the metric for the structure.

6.3 The BaTiO3 example (cubic perowskite structure)

An example follows here below for the cubic BaTiO3 structure: the following file is a subset of the CIF file. It is written as a plain text file and stored in the same directory of the main program as 'BaTiO3.cif'

```
#------------------------------------------------------------------------------
#$Date: 2016-02-13 21:28:24 +0200 (Sat, 13 Feb 2016) $
#$Revision: 176429 $
#$URL: svn://www.crystallography.net/cod/cif/1/54/21/1542140.cif $
#------------------------------------------------------------------------------
```

```
_cell_angle_alpha                  90
_cell_angle_beta                   90
_cell_angle_gamma                  90
_cell_length_a                     3.996
_cell_length_b                     3.996
_cell_length_c                     3.996
_symmetry_equiv_pos_as_xyz
x,y,z
-y,x,z
-x,-y,z
y,-x,z
x,-y,-z
y,x,-z
-x,y,-z
-y,-x,-z
z,x,y
-x,z,y
-z,-x,y
x,-z,y
z,-x,-y
x,z,-y
-z,x,-y
-x,-z,-y
y,z,x
y,-z,-x
z,y,-x
-y,z,-x
-z,-y,-x
-y,-z,x
z,-y,x
-z,y,x
-x,-y,-z
y,-x,-z
x,y,-z
-y,x,-z
-x,y,z
-y,-x,z
x,-y,z
y,x,z
-z,-x,-y
x,-z,-y
z,x,-y
-x,z,-y
-z,x,y
-x,-z,y
z,-x,y
x,z,y
-y,-z,-x
-y,z,x
-z,-y,x
y,-z,x
z,y,x
y,z,-x
-z,y,-x
z,-y,-x
loop_
_atom_site_label
Ba  0 0 0
Ti  0.5 0.5 0.5
O   0 0.5 0.5
loop_
```

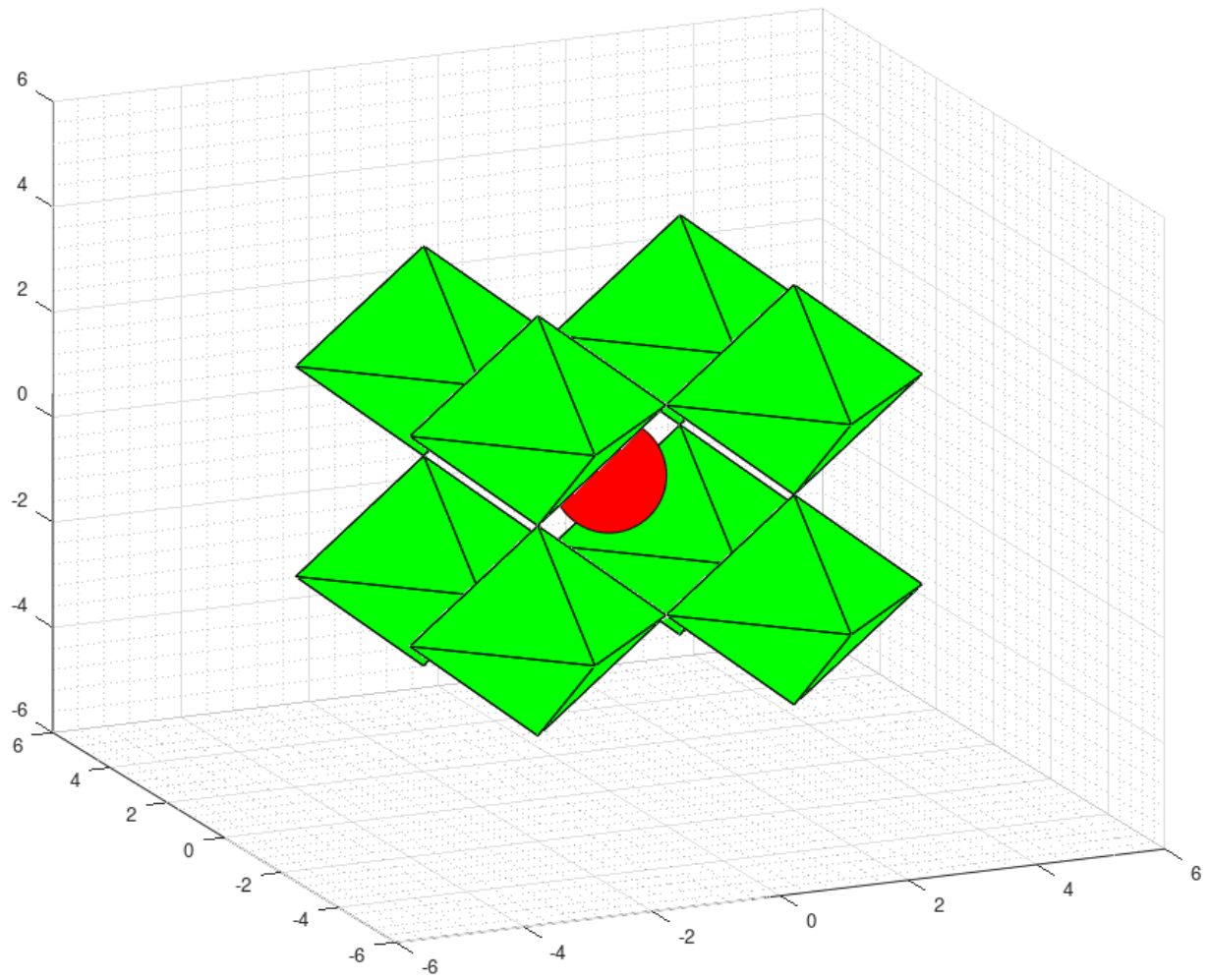
The following Octave script decodes step by step the text file and produces a 3-D image of the repetitive unit of the BaTiO3 lattice (a cubic perovskite structure).

```
clear all;clc;format short;format compact;

% --- section 1 --- the text file is read and decoded, finding the relevant
% informations for unit cell dimensions, its angles and positions of atoms inside

S = fileread('BaTiO3.cif');
a = textscan(S,"%s");
D = a{[1,1]};Lt = length(D);
j = 0;
for  i = 1:Lt
  if strcmp(D{i},"_cell_length_a") == 1;a0 = str2num(D{i+1});endif
  if strcmp(D{i},"_cell_length_b") == 1;b0 = str2num(D{i+1});endif
  if strcmp(D{i},"_cell_length_c") == 1;c0 = str2num(D{i+1});endif
  if strcmp(D{i},"_cell_angle_alpha") == 1;alf = str2num(D{i+1});endif
  if strcmp(D{i},"_cell_angle_beta") == 1; bet = str2num(D{i+1});endif
  if strcmp(D{i},"_cell_angle_gamma") == 1;gam = str2num(D{i+1});endif

  if strcmp(D{i},"_atom_site_label") == 1
    ++i;
    while strcmp(D{i},"loop_") == 0
      ++j;
      ++i;
      xN(j) = str2num(D{i});++i;
      yN(j) = str2num(D{i});++i;
      zN(j) = str2num(D{i});++i;
    endwhile
  endif
endfor
nAtomInd = j;

% --- section 2 ---the symmetry-equivalent positions are transformed in fractional
% coordinates by mean of the 'evalc' function of Octave

for k = 1:nAtomInd; % k=1 for 'Ba'    k=2 for 'Ti'   k=3 for 'O'
  x = xN(k);y = yN(k);z = zN(k);
  j = 0;
  for  i = 1:Lt
    if strcmp(D{i},"_symmetry_equiv_pos_as_xyz") == 1
    i = i+1;
    while strcmp(D{i},"loop_") == 0
      d2 = strsplit(D{i},",");++j;
      xC(j) = str2num(evalc(d2{1}));
      yC(j) = str2num(evalc(d2{2}));
      zC(j) = str2num(evalc(d2{3}));
      i = i+1;
    endwhile
    endif
  endfor

  nTot = j;
  for i = 1:nTot  % bring atoms in unit cell
    if xC(i)<0 ; xC(i) = xC(i) +1;endif
    if yC(i)<0 ; yC(i) = yC(i) +1;endif
    if zC(i)<0 ; zC(i) = zC(i) +1;endif
    if xC(i)>= 1;xC(i) = xC(i) -1;endif
    if yC(i)>= 1;yC(i) = yC(i) -1;endif
    if zC(i)>= 1;zC(i) = zC(i) -1;endif
  endfor
```

```
% --- section 3 --- as usual, many atomic positions are overlapping, so they must
% be purged so as to obtainthe real atoms located in unit cell

  m = 0;
  for i = 1:nTot  % eliminate coincident atoms
    for j = (i+1):nTot
      k2 = abs(xC(i)-xC(j)) + abs(yC(i)-yC(j)) + abs(zC(i)-zC(j));
      if (k2<0.001) && (xC(i)<999);xC(i) = 999;endif
    endfor
  endfor
  for i = 1:nTot
    if xC(i)<999;++m ;xA(m) = xC(i);yA(m) = yC(i);zA(m) = zC(i);endif
  endfor
  nT = m

% --- section 4 --- the fractional positions are transformed in cartesian
% coordinates according to unit cell dimensions a0,b0,c0. This is done for each
% atom type, barium, titanium and oxygen.

  switch (k)
    case 1; % Barium ------------------------------------------------------------
      w = 0;
      for i = 1:nT % transform coordinates in orthogonal, unit = Angstrom
        x = xA(i); % only one unite cell for barium
        y = yA(i);
        z = zA(i);
          ++w;xBa(w) = x*a0;yBa(w) = y*b0;zBa(w) = z*c0;
        endfor

   case 2; % Titanium -----------------------------------------------------------
      w = 0;
      for i = 1:nT % transform coordinates in orthogonal, unit = Angstrom
        for x = xA(i)-1 : xA(i);
        for y = yA(i)-1 : yA(i);
        for z = zA(i)-1 : zA(i);
          ++w;xTi(w) = x*a0;yTi(w) = y*b0;zTi(w) = z*c0;
        endfor
        endfor
        endfor
      endfor

   case 3; % Oxygen -------------------------------------------------------------
      w = 0;
      for i = 1:nT % transform coordinates in orthogonal, unit = Angstrom
        for x = xA(i)-1 : xA(i)+1;
        for y = yA(i)-1 : yA(i)+1;
        for z = zA(i)-1 : zA(i)+1;
          ++w;xO(w) = x*a0;yO(w) = y*b0;zO(w) = z*c0;
        endfor
        endfor
        endfor
      endfor
  endswitch
endfor

% --- section 5 --- Coordination polyhedra are drawn, first finding the six oxygen
% atoms which build the coordination polyhedra of each titanium , TiO6

figure(1,'position',[100 100 1000 800]);
axis([-6 6 -6 6 -6 6]);
axis on;grid on;grid minor on;view(-24,20);
```

```
v2 = zeros(6,3);
for j = 1:length(xTi); % find oxygen octahedra around titanium atoms
  m = 0;disp([num2str(j),'------------']);
  for i = 1:length(xO);
    dist = sqrt((xTi(j)-xO(i))^2 + (yTi(j)-yO(i))^2 + (zTi(j)-zO(i))^2);
    if dist<2.2;
      ++m;
      v2(m,1) = xO(i);v2(m,2) = yO(i);v2(m,3) = zO(i);
    endif
  endfor
% find opposite vertices and then the corners of triangular faces of the oxygen
% octahedron
  v3 = mean(v2);
  w = 0;
  for i = 1:5
    for j = i+1:6
      s1(1) = abs(mean([v2(i,1),v2(j,1)]) - v3(1));
      s1(2) = abs(mean([v2(i,2),v2(j,2)]) - v3(2));
      s1(3) = abs(mean([v2(i,3),v2(j,3)]) - v3(3));
      if sum(s1)<0.05;++w;s2(w,1) = i;s2(w,2) = j;endif
    endfor
  endfor
  f2 =
[s2(1,1),s2(2,1),s2(3,1);s2(1,1),s2(2,1),s2(3,2);s2(1,1),s2(2,2),s2(3,1);s2(1,1),s
2(2,2),s2(3,2);...

s2(1,2),s2(2,1),s2(3,1);s2(1,2),s2(2,1),s2(3,2);s2(1,2),s2(2,2),s2(3,1);s2(1,2),s2
(2,2),s2(3,2)]
  patch('Faces',f2,'Vertices',v2,'FaceColor','green');
endfor
% find barium atom to be pictured by a round marker
for j = 1:length(xBa);
  hold on;plot3(xBa(j),yBa(j),zBa(j),"ok",'markersize',60,'markerfacecolor','r');
endfor
```

The graphic output of the above script looks as follows:


6
4
2
0
-2
-4
-6
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

Chapter 7. Gases and vapors

7.1 Introduction

In an ideal gas, particles, either atoms or molecules, freely move inside a constant volume container without each other interacting, except for very brief collisions in which they exchange energy and momentum. By colliding with the container's walls, they exchange energy with their thermal environment.
As a consequence of the above random motion, the three macroscopic parameters of ideal gases, namely pressure P, volume V, and absolute temperature T, are related by what is known as the ideal gas law. Further assumptions are dimensionless particles and interaction with elastic collisions, not subjected to other attractive/repulsive forces. Under such assumption, the ideal gas law takes the form:

$$PV = nRT \qquad (7.1)$$

where n is the number of moles of the particles, in other terms, it is their real number in the container divided by the Avogadro's number. If the pressure P is measured in atm units, the volume V in dm^3 , and T in Kelvin, the universal gas constant R holds:

$$R = 0.082\, atm\, L\, K^{-1} mol^{-1} \qquad (7.2)$$

In other instances, the SI value for R is employed, namely $8.314\, JK^{-1} mol^{-1}$.
As a consequence, the product PV is a constant value under constant temperature (Boyle's law), implying that, at a very high pressures, the gas volume would shrink to an extremely small size, approaching zero. This fact cannot occur, because the molecules themselves occupy a finite space (called covolume) as they are practically incompressible. Another consideration is that intermolecular attractive forces exist between gas molecules. Because of the intermolecular attractive forces, the momentum exchanged with the container walls during the collisions of the gas molecules must be smaller than that expected by an ideal gas. As a consequence, the measured gas pressure must drop to a smaller value. In summary, gases tend to behave ideally at high temperatures and low pressures, but they tend to depart from the ideal gas behavior (real gases) at low temperatures and high pressures.

A number of equations are available for describing the behavior of real gases in a wide range of temperatures and pressures. Such equations are not as generic as the ideal gas equation, since they must contain terms which are specific of each gas.
Usually, such equations are capable of correcting for the proper volume or covolume of the molecules and intermolecular forces of attraction. Of the equations that chemists employ for modeling the behavior of real gases, the van der Waals (1837-1923) equation (4.3),only requires two specific coefficients, a and b, whose values vary from molecule to molecule .

$$\left(P+\frac{an^2}{V^2}\right)\cdot\left(V-nb\right)=nRT \qquad (7.3)$$

7.2 Distribution of molecular velocities in the case of oxygen

Gas particles move randomly, continuously exchanging their kinetic energy during mutual collisions. The particle absolute velocity v is related to kinetic energy by $E_c=0.5\,m\,v^2$
The probability density function for the particle speed in a gas follows the Maxwell-Boltzmann probability density :

$$f(v)dv = 4\pi m^2\left(\frac{m}{2\pi RT}\right)^{3/2}\cdot\exp\left(\frac{-mv^2}{2RT}\right)dv \qquad (7.4)$$

where v is the absolute velocity in [m/s] , m is the mass in kg of 1 mole of substance
R = 8.314 $JK^{-1}mol^{-1}$ is usually given in SI units, and T [K] is the absolute temperature.
A simple Octave script computes and graphically plots in Fig. 7.1, left, the above probability density for oxygen, with molar mass m = 0.032 kg, at three different temperatures.
Different gases can be simulated by the script, by simply changing the oxygen molar mass, indeed the script refers to the oxygen molecule, but relevant data can be modified by users.
The script sections are initialization, user data, computation, and graphical plot of the results.

```
clear;clc;j = 1;
u = (0:20:2500); % range of velocities (m/s) to be explored.
R = 8.3145;    % R value in SI system.
M = 32;       % mass of 1.00 mol of oxygen (O2) in grams.
M = M/1000;   % mass of one mole must be in SI units (kg).
for Tc = (0:500:1000)      % Temperature in Â°C
  T = Tc + 273.15;         % Temperature in K
  a1 = 4*pi*(M/(2*pi*R*T))^1.5; % constant part of the equation
  u2 = u.^2;
  Y(':',j) = a1*u2.*exp(-M/(2*R*T)*u2); % variable part (from 'u' velocity)
  j = j + 1;
end
plot(u,Y(':',1),'b',u,Y(':',2),'r',u,Y(':',3),'g','LineWidth',2);grid on;
xlabel('velocity m/s');ylabel('probability')
title('Distribution of molecular speeds for O2 at 3 temp.')
legend('0 °C','500 °C','1000 °C')
format compact
disp('Total kinetic energy for 1 mole of oxygen (or whatever gas)')
for T = (0:200:1000)
  kinE = 1.5*R*(T+273.15);
   T1 = ['Total kinetic energy at ',num2str(T),' Â°C = ',num2str(round(kinE)),'
joule'];
  disp(T1);
end
```

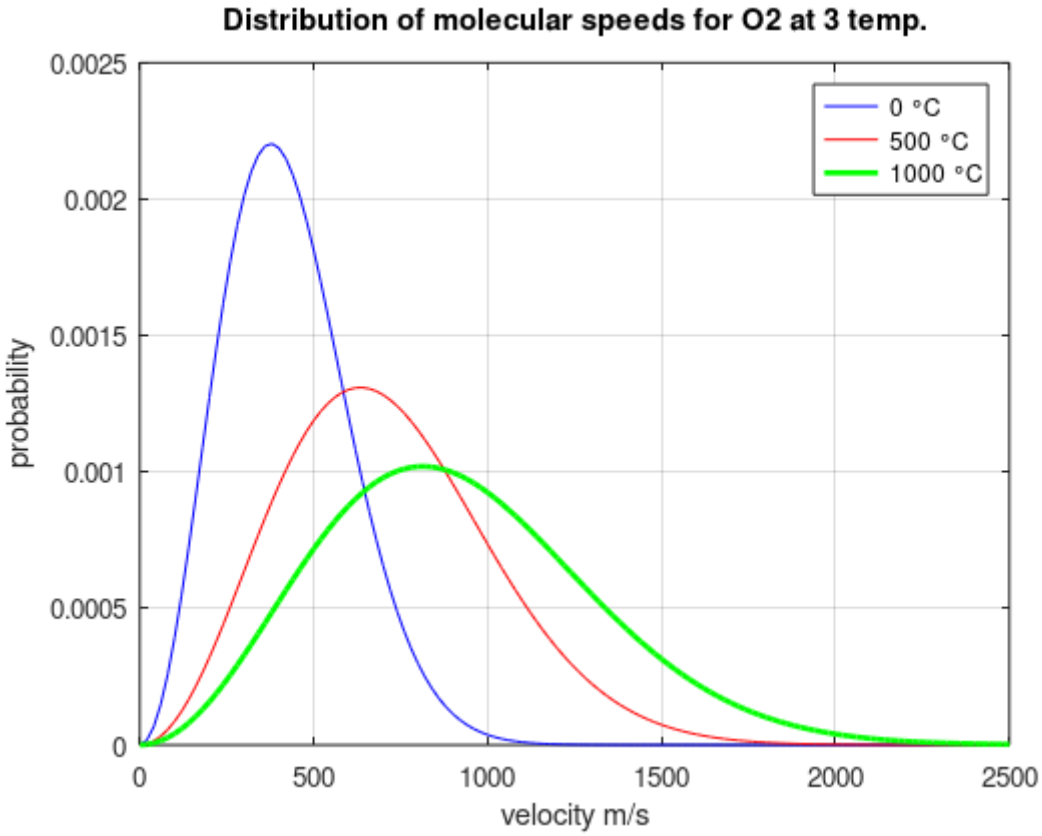


Fig. 7.1 Distribution of molecular speeds for oxygen gas at 3 different temperatures

7.3 Van der Waals isotherm of real gases

Van der Waals equation (7.3) can be rearranged into an expression of the type P = P(V) as follows:

$$P(V) = \frac{nRT}{V-nb} - \frac{an^2}{V^2} \quad \text{where} \quad V>Vmin=nb \qquad (7.5)$$

The graph of P = P(V) described by the above equation, is known as the van der Waals isotherm. The critical point of a gas is defined as the temperature limit above which the examined gas cannot be liquefied for whatever pressure is applied. If the gas temperature is lower than this critical point, the isotherm is nonmonotonic versus the volume V. In other terms, in a certain critical volume interval of the curve, the calculated pressure should decrease with increasing volume, a fact clearly inconsistent. Therefore van der Waals isotherm should be adjusted in this critical region where local minimum and maximum take place so as to become ideally constant. The physical meaning of this region (known as the condensation region), between the extreme volumes V0, V2 to be found, is that both gas and liquid phases coexist, since the gas phase progressively condenses to become liquid as far as the volume decreases at constant pressure.

J.C. Maxwell (1831-79) suggested to better define this region by balancing the areas of negative and positive 'energy', below and above the rectified curve (condensation line), which corresponds to a constant pressure P-condensation. The adjustment is known as the Maxwell construction (see fig. 7.2 and enlarged fig. 7.3).
As a matter of fact, the real behaviour of the gas (in this case carbon dioxide, CO2) by increasing volumed (and decreasing pressure) follows from left to right the red line, then the green one in the condensation region, and finally the dark blue one. In the red, green and dark blue region respectively, liquid, liquid and gas, and gas phase are present.

The problem requires to express the above two negative and positive areas in term of the current pressure P. To this end, the expression (7.5) is numerically evaluated and integrated with the aim of finding the pressure of condensation (`Pc`), satisfying the equal area condition (pink area = orange area) of fig. 7.3

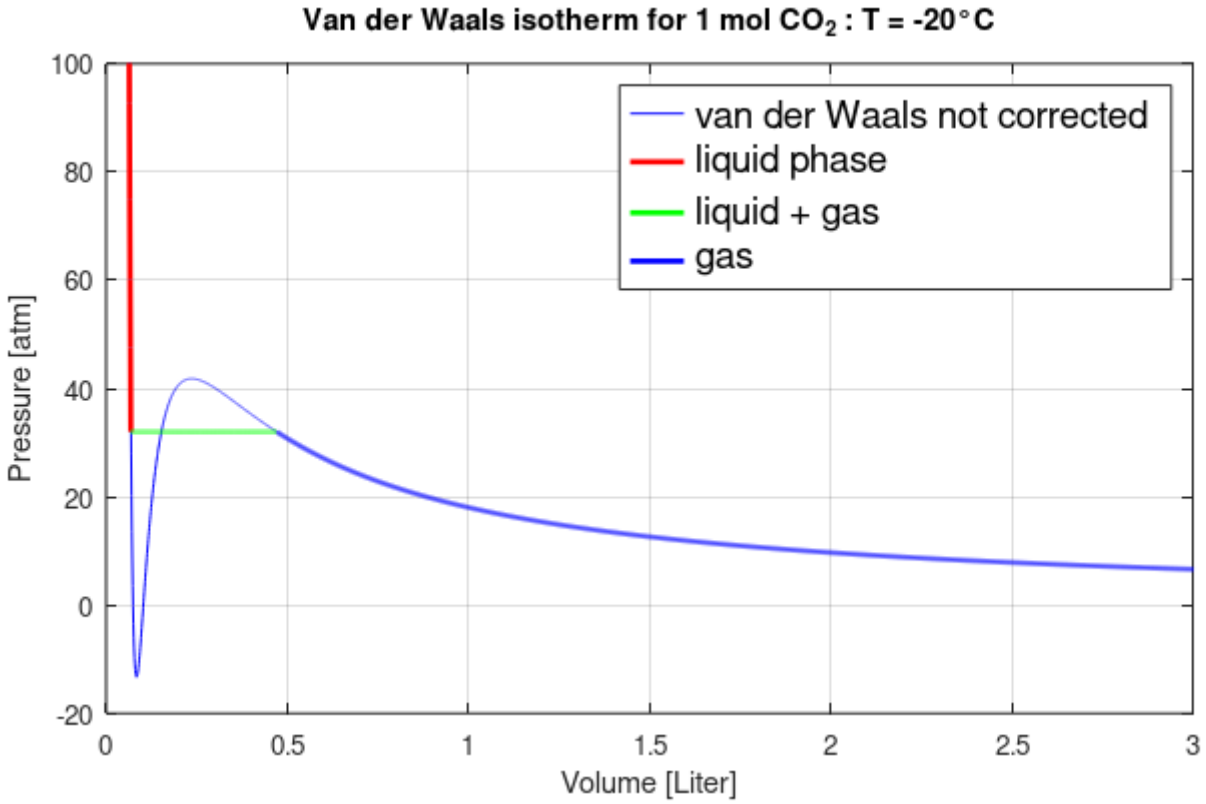


Fig 7.2 Van der Waals isotherm for cabon dioxide, T = -20°C, 1 mol

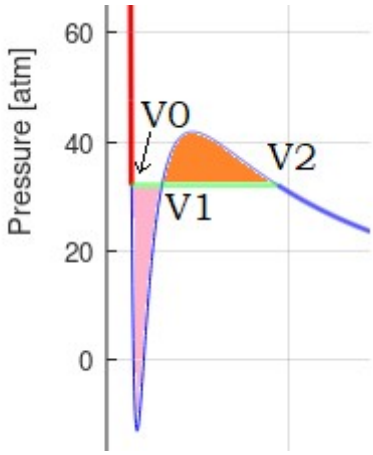


Fig 7.3 Commented enlargement of part of fig. 7.2

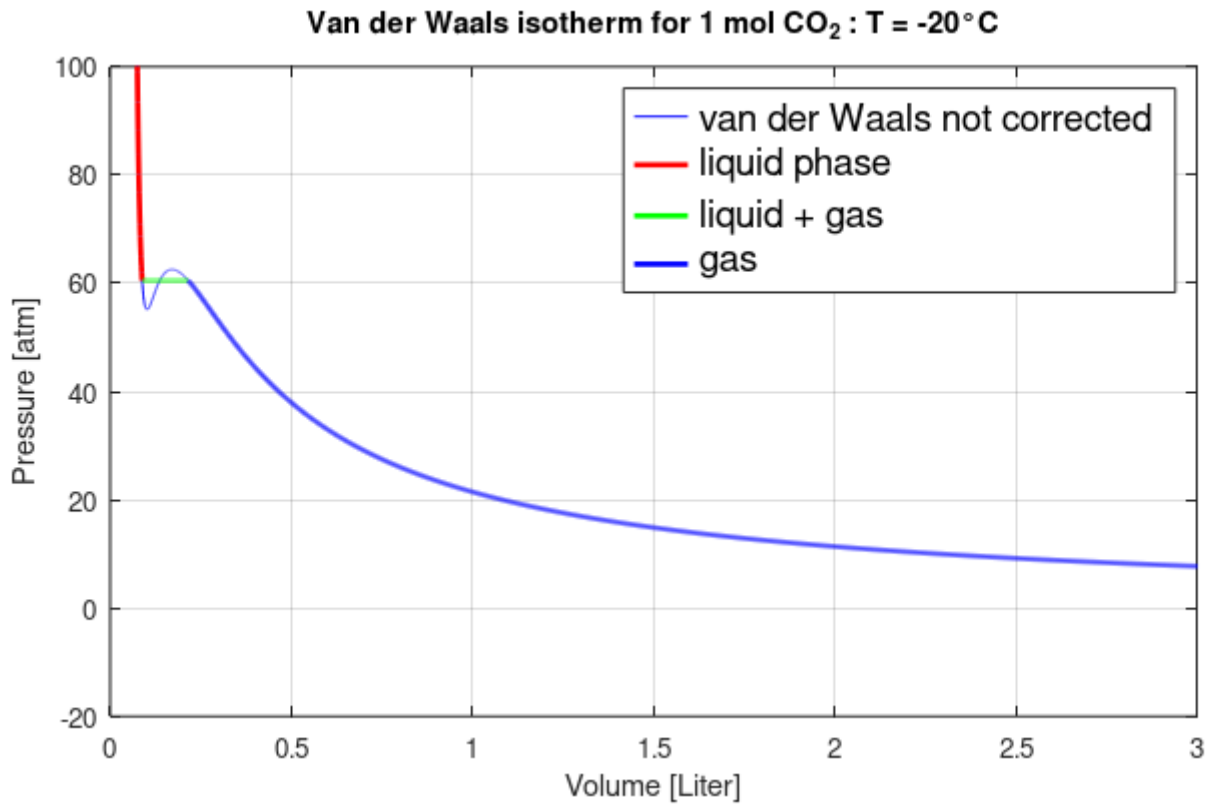


Fig 7.4 Van der Waals isotherm for carbon dioxide, T = 20°C, 1 mol

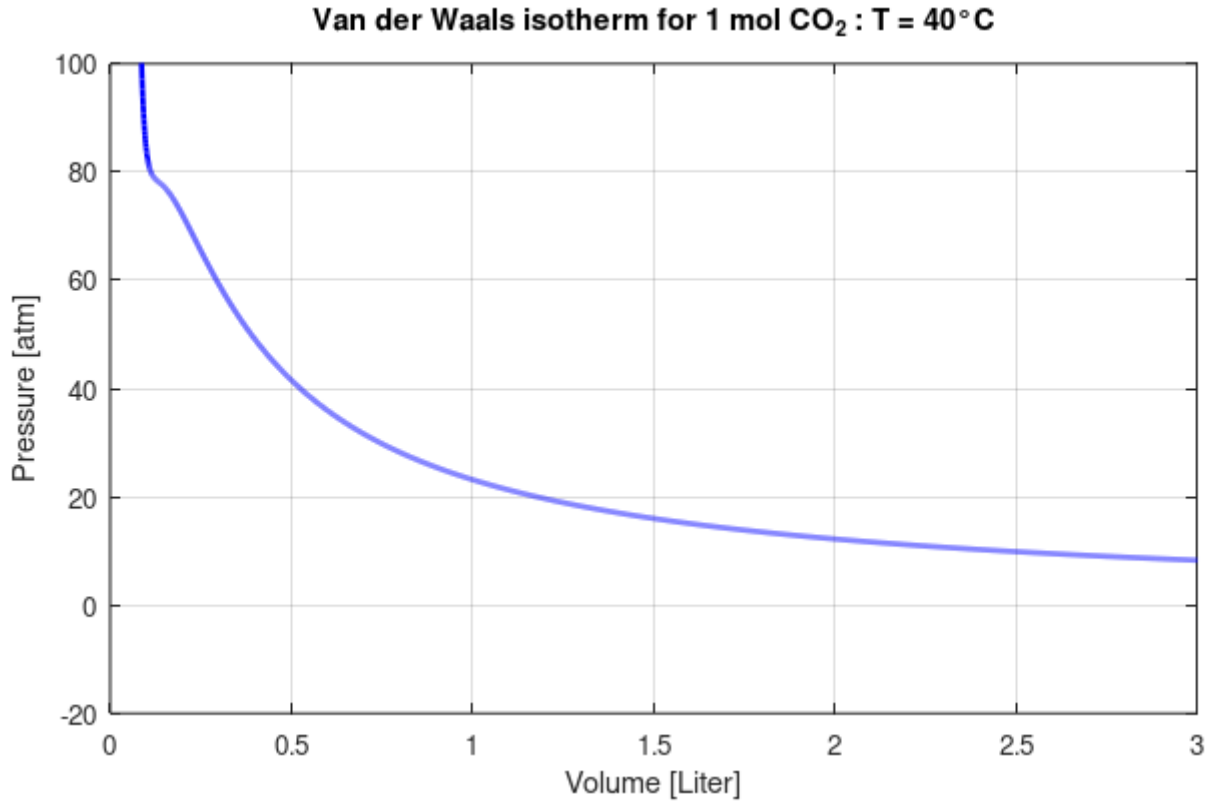


Fig 7.5 Van der Waals isotherm for carbon dioxide, T = 40°C, 1 mol (above critical temperature)

```
clear;clc;format compact %;format short
%------------------------------------ function() ----------
function P = pressure(V,T,a,b)
    R = 0.082;     % R value in liter/atm.
    % Van der Waals equation in the form P = f(V) n(mol) = 1
    P = R*T./(V-b) - a./V.^2;
end
```

```
%------------------------------------ function() ----------

% Some Van der Waals coefficients for common gases (from R.Petrucci et al.)
% a = 4.225; b = 0.0371; % ammonia (NH3)
% a = 1.37; b = 0.0387;  % nitrogen (N2)
 a = 3.658; b =0.0429;  % carbon dioxide (CO2)
Tc = 20; T = Tc + 273.15; % T = temperature in K   ,  Tc = temperature in °C
V = linspace(b+0.001,4+b,4000);
P = pressure(V,T,a,b);m1 = 0;n1 = 0;Plimit = 200;
for i = 2:3999
  if P(i)>Plimit ; a9 = i ;endif
  if P(i-1)>P(i) && P(i+1)>P(i);m1 = i;end           % Find minimum
  if P(i-1)<P(i) && P(i+1)<P(i);n1 = i;break;end     % Find maximum and exit from
for..end cycle
endfor
P = P(a9+1:end);V = V(a9+1:end);  % trimming out from the vector unwanted values
figure(1,'position',[200 100 700 400]); % locate and resize the figure (1)

if (n1 - m1) < 3 % ----------> no condensation occurs
  plot (V,P,'b','LineWidth',2);grid on;axis([0 3 -20 100]);hold on;
  title(['Van der Waals isotherm for 1 mol CO_2 : T = ',num2str(Tc),' °C']);
  xlabel('Volume [Liter]'); ylabel('Pressure [atm]');

else  % ---------------------> yes, condensation occurs
  Pm = round(P(m1));Pn = round(P(n1));
  for Pc = Pm+1:0.5:Pn-1
    a1 = 0;a2 = 0; % cumulative counters for the two areas , pink and orange in fig.
    for i = 1:m1
      if P(i)<Pc
        a1 = a1 + Pc - P(i);
        % summation of Pressure in first half of downward peak (pink)
        else a3 = i;
      endif
    endfor
    for i = m1+1:n1
      if P(i)<Pc ; a1 = a1 + Pc - P(i);endif
      % summation of Pressure second half of downward peak (pink)
      if P(i)>Pc ; a2 = a2 + P(i) - Pc;endif
      % summation of pressure first half of upward peak (orange)
    endfor
    for i = n1+1:3000
      if P(i)>Pc
        a2 = a2 + P(i) - Pc;a4 = i;
        % summation of Pressure second half of upward peak (orange)
        else;break;
        % once the summation is finished, exit from the for..endfor cycle
      endif
    endfor
      if a1>a2 ; break;endif
      % once the two area a1 > a2 exits from for..endfor cycle
  endfor
  % in case for...endfor cycle has not ended in the above line, a new increased value
is assigned to Pc

  plot (V,P,'b');grid on;axis([0 3 -20 100]);hold on;
  P(a3:a4) = Pc;
  plot (V(1:a3),P(1:a3),'r','LineWidth',2)
```

```
  plot (V(a3:a4),P(a3:a4),'g','LineWidth',2)
  plot (V(a4:end),P(a4:end),'b','LineWidth',2);title('Van der Waals isotherm for 1 mol
CO_2 : T = -20°C');
  xlabel('Volume [Liter]'); ylabel('Pressure [atm]');
  legend('van der Waals not corrected','liquid phase','liquid +
gas','gas','fontsize',14)
endif
```

7.4 Water vapor pressure

Water left in an open beaker evaporates completely. A different condition establishes if water is placed in a closed container. If both liquid and vapor are present in the container, vaporization and condensation simultaneously occur. If a sufficient liquid volume is present, the condition eventually is reached in which the amount of vapor remains constant. This is an example of dynamic equilibrium: the two opposing processes, evaporation and condensation, take place simultaneously and at equal rates.

Even in the absence of chemical reactions, we can employ the NASA-CEA coefficients to extrapolate the thermodynamic variables H (enthalpy) and S (entropy) of the solid, liquid, and gaseous water, and then compute ΔG0 (variation of Gibbs energy in standard conditions) of the following transformations:
H2O solid ⇔ H2O liquid ⇔ H2O gas

From the values of ΔG0, we can deduce the water vapor pressure as follows:

$K_{eq} = \frac{p(H_2O)_{gas}}{\{H_2O_{liq}\}}$ where $\{H_2O_{liq}\} = 1$, being the activity of pure liquid water, and ΔG0 refers to a constant pressure equal to 1 atm, but to a variable temperature. Therefore the equilibrium constant Keq can be renamed as K_P since we are treating the equilibrium of a gaseous substance.

$$K_{eq} = K_p = p(H_2O_{gas}) = \exp(\frac{-\Delta G^0}{RT}) \qquad (7.6)$$

The Octave script automatically switches at 0°C from solid water (ice) to liquid water.

```
%       THERMOCHEMICAL DATA FROM https://cearun.grc.nasa.gov/ThermoBuild/
%    H = R*(-a1/T + a2*log(T) + a3*T + a4*T^2/2 + a5*T^3/3 + a6*T^4/4 + a7*T^5/5 + a8)
%    S = R*(-a1/2/T^2 - a2/T + a3*log(T) + a4*T + a5*T^2/2 + a6*T^3/3 + a7*T^4/4 + a9)
clear;clc;
% solid H2O, valid 200 to 273.15 K
H2O_cr = [-4.026777480e+05,2.747887946e+03,5.738336630e+01,-8.267915240e-
01,4.413087980e-03,-1.054251164e-05,9.694495970e-09,-5.530314990e+04,-
1.902572063e+02];
% liquid H2O, valid 273.15 to 373.15 K
H2O_liq = [1.326371304e+09,-2.448295388e+07,1.879428776e+05,-
7.678995050e+02,1.761556813,-2.151167128e-03,1.092570813e-06,1.101760476e+08,-
9.779700970e+05];
% liquid H2O, valid 373.15 to 600
H2O_liq_high = [1.263631001e+09,-1.680380249e+07,9.278234790e+04,-
2.722373950e+02,4.479243760e-01,-3.919397430e-04,1.425743266e-07,8.113176880e+07,-
5.134418080e+05];
% gas H2O, valid 200 to 1000 K
H2O_gas = [-3.947960830e+04,5.755731020e+02,9.317826530e-01,7.222712860e-03,-
7.342557370e-06,4.955043490e-09,-1.336933246e-12,-3.303974310e+04,1.724205775e+01];
```

```
R = 8.314472;
% The NASA coefficients are treated as vector, so they are summed according to
stoichiometry of the reaction.
  Rx1 = H2O_gas - H2O_cr;% H2O(solid) <==> H2O(gas) 200 --> 273.15
  Rx2 = H2O_gas - H2O_liq;% H2O(liquid) <==> H2O(gas) 273.15 --> 373.15
  Rx3 = H2O_gas - H2O_liq_high;% H2O(solid) <==> H2O(gas) 373.15 --> 600
disp('   Temp. (°C)     p(H2O) atm.');
format short;format compact;
j=0;
% start of the heating cycle.
for T1 = (-40:1:200)
  j = j + 1;
  T = T1 + 273.15;a(j) = T1;
  TxH = [-1/T,log(T),T,T^2/2,T^3/3,T^4/4,T^5/5,1,0];
  TxS = [-1/T^2/2,-1/T,log(T),T,T^2/2,T^3/3,T^4/4,0,1];
  if T <= 273.15
     DeltaH = R*sum(TxH.*Rx1);DeltaS = R*sum(TxS.*Rx1);DeltaG = DeltaH - T*DeltaS;
     Kp = exp(-DeltaG/R/T);b(j) = Kp;disp([T1,Kp]);% Kp = p(H2O),atm
  end
  if (273.15 < T &&  T <= 373.15)
     DeltaH = R*sum(TxH.*Rx2);DeltaS = R*sum(TxS.*Rx2);DeltaG = DeltaH - T*DeltaS;
     Kp = exp(-DeltaG/R/T);b(j) = Kp;disp([T1,Kp]);
  end
  if 373.15 < T
     DeltaH = R*sum(TxH.*Rx3);DeltaS = R*sum(TxS.*Rx3);DeltaG = DeltaH - T*DeltaS;
     Kp = exp(-DeltaG/R/T);b(j) = Kp;disp([T1,Kp]);
  end
end
plot(a,b,'b','LineWidth',2) ;grid on;
xlabel('temperature [°C]')
ylabel('water vapour pressure [atm]')
title('water vapour pressure of ice and liquid')
```

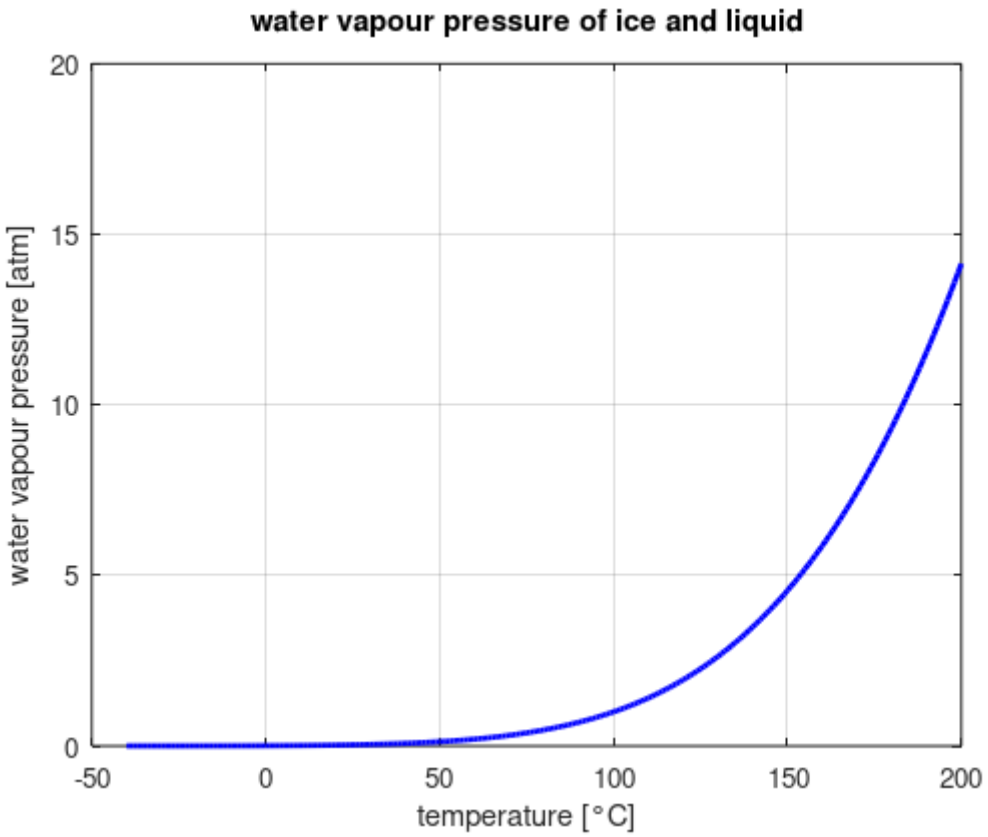


Fig 7.6 Water vapor pressure from -50 to 200 °C

7.5 Water vapor pressure at different altitude and humidity

As a further application of the NASA-CEA coefficients , we compare the water vapor pressure with the CO2 partial pressure at different altitudes in the Earth's atmosphere. Since both gases are infrared active as greenhouse gases, knowledge of the relative concentrations is of utmost interest. The greenhouse effect is particularly active in the upper troposphere, where the concentration of water vapor drops to very low values, due to decreasing temperature and consequent condensation.
The atmosphere's temperature profile, which assumes a constant temperature decrement, is written as

$T = T_0 - Lh = 15 - 0.00649 \cdot h$ where h is the altitude above sea level [m]

By applying the ideal gas law in Eq. (7.1), one obtains $P = \rho RT/M$,where P is in atm units, ρ, the atmosphere density, is given in [kg/m^3], R = 8.314 J K^{-1}mol^{-1} is expressed in SI units, T is the absolute temperature, and M = 0.02896 is the average molecular mass of the air in [kg/mol].
We assume hydrostatic pressure, namely a pressure differential equal to $\delta P = -\rho g \delta h$, where g = 9.806 m/s^2 is the gravity acceleration at sea level. The pressure differential divided by P provides the differential equation:

$$\frac{dP}{P} = \frac{-Mg}{RT} \cdot dh \quad , \quad P_0 = P(h=0) \qquad (7.7)$$

Integration of Eq. (7.7) from sea level to altitude h and the previous assumption of a linear temperature variation $T = T_0 - Lh$ together with a constant air composition, provides the pressure profile

$$P(T) = P_0 \left(\frac{T}{T_0}\right)^{\frac{MG}{RT}} = \left(\frac{T}{288}\right)^{5.256} \quad , \quad T = T[^\circ C] + 273.15 \qquad (7.8)$$

Fig. 7.7(a), plots the water vapor pressure versus the altitude in two colors: the blue color refers to ice and the red color to liquid water. The ice melting point at about 2.2 km altitude, corresponds to the liquid⇔solid equilibrium for a 100% relative humidity (RH).
In Fig. 7.7(b), the concentrations [mol/m^3] partial pressures of water and carbon dioxide (at 410 ppmv) are compared, by assuming a 50% relative humidity. From about 2.2 km altitude, the contribution of CO2 is prevalent, and from 8.0 km altitude, the water contribution becomes very low and nearly negligible. The commented Octave script follows

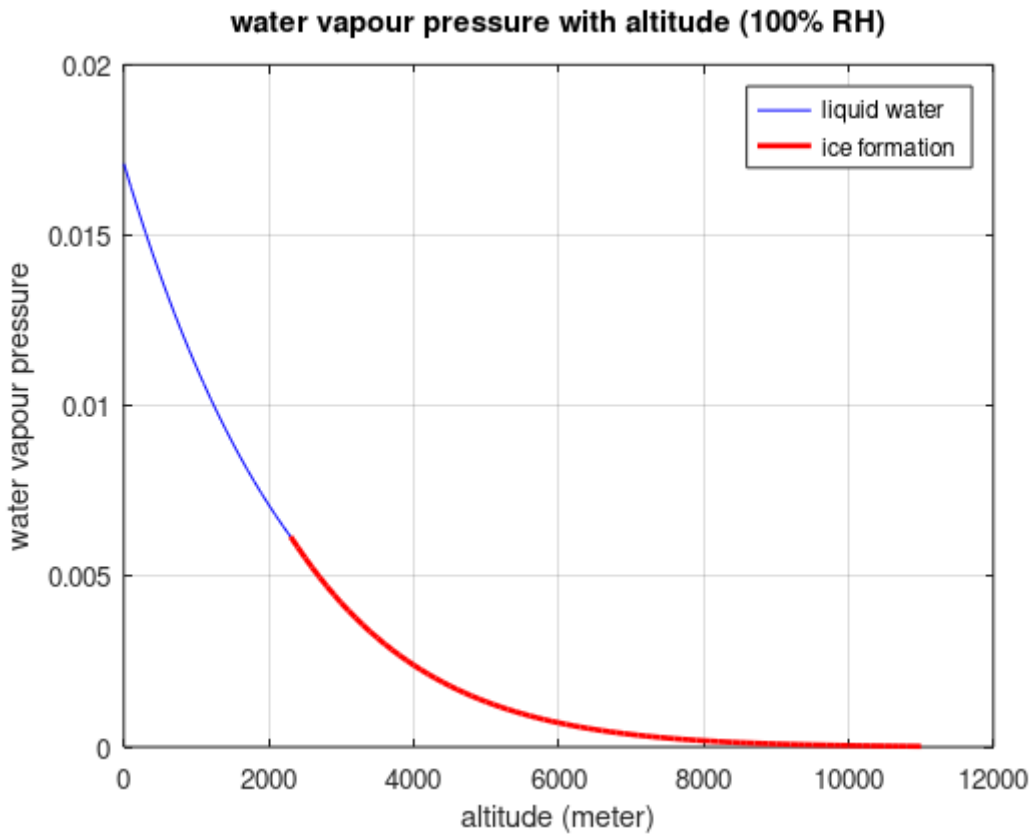


Fig. 7.7(a) Water vapor pressure from sea-level to 12 km (100% relative humidity)

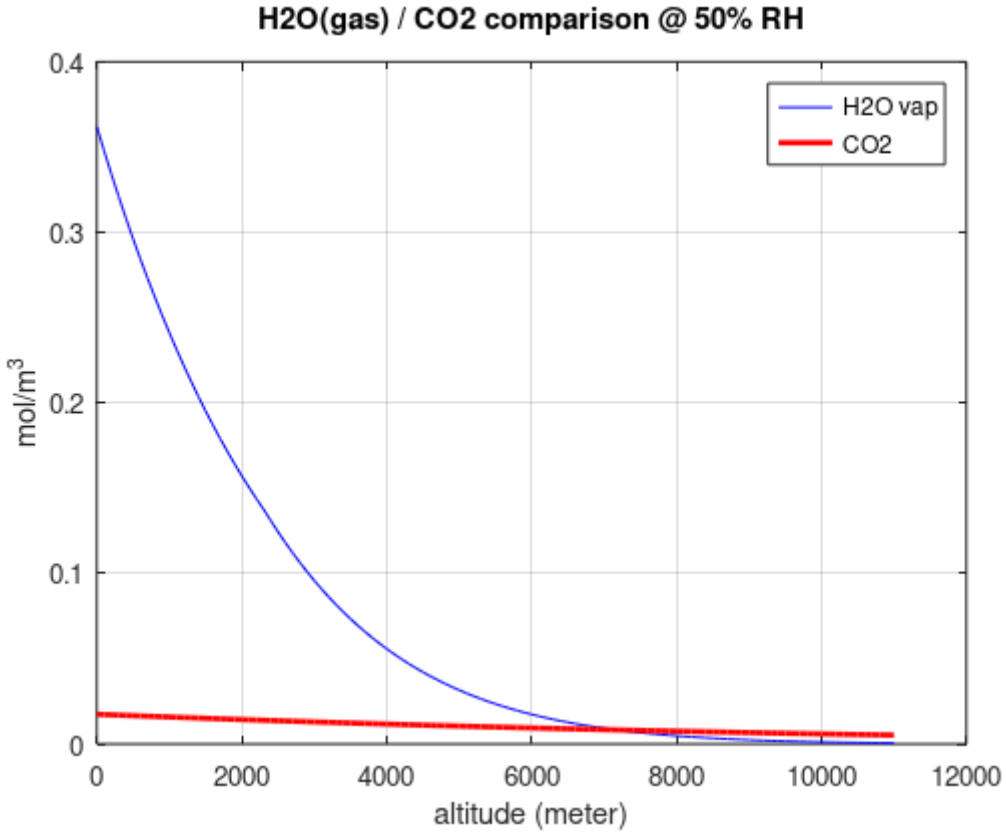


Fig. 7.7(b) Comparison between molar concentrations [mol/m^3] of water gas and carbon dioxide from sea-level to 12 km (50% relative humidity)

```
clear;clc;
% solid H2O, valid 200 to 273.15 K
H2O_cr = [-4.026777480e+05,2.747887946e+03,5.738336630e+01,-8.267915240e-
01,4.413087980e-03,-1.054251164e-05,9.694495970e-09,-5.530314990e+04,-
1.902572063e+02];
```

```
% liquid H2O, valid 273.15 to 373.15 K
H2O_liq = [1.326371304e+09,-2.448295388e+07,1.879428776e+05,-
7.678995050e+02,1.761556813,-2.151167128e-03,1.092570813e-06,1.101760476e+08,-
9.779700970e+05];
% H2O gas, valid 200 to 600 K
H2O_gas = [-3.947960830e+04,5.755731020e+02,9.317826530e-01,7.222712860e-03,-
7.342557370e-06,4.955043490e-09,-1.336933246e-12,-3.303974310e+04,1.724205775e+01];

R = 8.314472;
% The NASA coefficients are treated as vector, so they are summed according to
stoichiometry of the reaction.
  Rx1 = H2O_gas - H2O_cr;% H2O(solid) <==> H2O(gas) 200 --> 273.15
  Rx2 = H2O_gas - H2O_liq;% H2O(liquid) <==> H2O(gas) 273.15 --> 373.15
disp('   Temp. (Â°C)     p(H2O) atm.');
format short;format compact;
j = 0;
% start of the elevation in the atmosphere. h1 = altitude in meter.
for h1 = (0:100:11000)
    j = j + 1;
    T1 = 15.04 - 0.00649*h1;
    T = T1 + 273.15;
    P1 = (101.29*(T/288.08)^5.256)/101.325;
    TxH = [-1/T,log(T),T,T^2/2,T^3/3,T^4/4,T^5/5,1,0];
    TxS = [-1/T^2/2,-1/T,log(T),T,T^2/2,T^3/3,T^4/4,0,1];
  if T <= 273.15
     DeltaH = R*sum(TxH.*Rx1);DeltaS = R*sum(TxS.*Rx1);DeltaG = DeltaH - T*DeltaS;
     Kp = exp(-DeltaG/R/T);a(j) = h1;b(j) = Kp;
  else
     DeltaH = R*sum(TxH.*Rx2);DeltaS = R*sum(TxS.*Rx2);DeltaG = DeltaH - T*DeltaS;
     Kp = exp(-DeltaG/R/T);a(j) = h1;b(j) = Kp;j1 = j;
  end
  mol = Kp*1000/0.082/T*0.5;c(j) = mol;  % mol H2O per cubic meter with 50% relative
humidity.
  mol1 = P1*410*1e-6*1000/0.082/T;c1(j) = mol1; % mol CO2 per cubic meter
  disp([T1,Kp,mol,mol1]);% Kp = p(H2O),atm
end
a1 = a(1:j1);b1 = b(1:j1);
a2 = a(j1:j);b2 = b(j1:j);
plot(a1,b1,'b',a2,b2,'r','LineWidth',2) ;grid on;hold on;
xlabel('altitude (meter)');ylabel('water vapour pressure')
title('water vapour pressure with altitude (100% RH)')
legend('liquid water','ice formation')
figure
plot(a,c,'b',a,c1,'r','LineWidth',2) ;grid on;hold on;
xlabel('altitude (meter)');ylabel('mol/m^3')
title('H2O(gas) / CO2 comparison @ 50% RH')
legend('H2O vap','CO2')
```

7.6 Compressibility of a real gas

The ideal gas equation $PV=nRT$ (7.1) tells us that the ratio $Z = \frac{PV}{nRT}$ is equal to 1 for an ideal gases, whereas, for real gases, the dimensionless compressibility factor Z may have values which are significantly different from 1.

The compressibility factor Z is usually plotted for real gases versus the pressure P under isothermal conditions. Here, the pressure is given in the standard atmosphere unit [atm] and consequently R = 0.082 atm L $K^{-1}mol^{-1}$. At very high pressures, we have Z > 1. In fact, because of the finite size of molecules, the product PV at high pressures is larger than that predicted for ideal gases.
Intermolecular forces of attraction account for Z < 1. These forces become increasingly important at low temperatures, where the molecular translational motion slows down.

The van der Waals (eq. 7.1) for real gases can be rearranged into a third-degree polynomial

$$nRTV^2 = PV^3 - PnbV^2 - an^2V - abn^3 \qquad (7.8)$$

which, solved by the Octave function '*roots*', provides the value of the volume V for a given P and finally the compressibility factor Z.
Fig. 7.8 shows the compressibility profiles (dimensionless) versus pressure in atm for nitrogen(N2) gas. Other gases can be inserted by providing the corresponding {a,b } values.

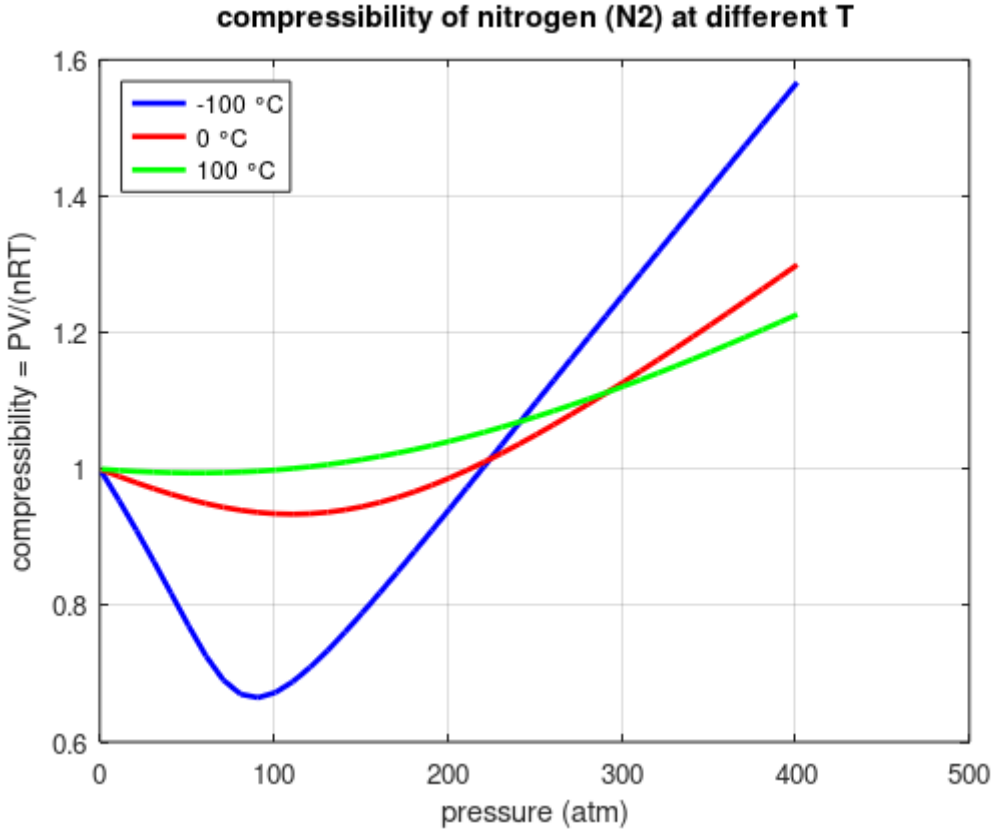


Fig. 7.8 Z (compressibility value) for nitrogen at -100, 0 ,and 100 °C respectively

```
clear all:close;clc;format compact;format short;
% Some Van der Waals coefficients for common gases (from R.Petrucci et al.)
% a = 4.225; b = 0.0371; % ammonia (NH3)
 a = 1.37; b = 0.0387;  % nitrogen (N2)
% a = 3.658; b =0.0429;  % carbon dioxide (CO2)
for Tc = (-100:100:100)        ; % Temperature in °C.
  T = Tc + 273.15; % Temperature in K.
  R = 0.082;j=0;
for P = (1:10:401) % P values are assigned.
    % P*V^3 - (R*T + b*P)*V^2 +a*V -a*b = 0 (1 mol) is solved in 'V'
    c = [P,-(R*T + b*P),a,-a*b];
```

```
    S = roots(c)
    j=j+1;
    Comp(j) = P*S(1)/R/T;Pi(j) = P; % I have established that (2) and s(3) are
complex!
end
switch Tc
    case -100
        plot(Pi,Comp,'b','Linewidth',2);grid on;hold on
    case 0
        plot(Pi,Comp,'r','Linewidth',2);
    case 100
        plot(Pi,Comp,'g','Linewidth',2);
end
end
xlabel('pressure (atm)')
ylabel('compressibility = PV/(nRT)')
title('compressibility of nitrogen (N2) at different T');
legend('-100 °C','0 °C','100 °C','Location','northwest')
```

Chapter 8. Chemical gas-phase equilibria

8.1 Introduction

In dealing with chemical transformations, the first question to be addressed is whether, under the given circumstances, the process is spontaneous. The criterion of the entropy increase does not apply directly, and, as established by J.W. Gibbs (1839-1903), one should account for two entropy changes to be simultaneously considered: the change of the system itself, ΔSsys, and the change of the surroundings ΔSsurr, their sum being the entropy variation of the whole, generally speaking, of the universe:

$$\Delta S_{univ} = \Delta S_{sys} + \Delta S_{surr} \qquad (8.1)$$

Expression (5.1) provides a basic criterion to discriminate spontaneous changes. It is indeed one way of stating the second law of thermodynamics. In other words, all the spontaneous processes produce an increase in the entropy of the universe.
According to expression (5.1), if a process produces positive entropy changes in both the system and the surroundings, the process is surely spontaneous. If both entropy changes are negative, the process occurs to be nonspontaneous. If one of the entropy changes is positive and the other negative, whether the sum is positive or negative depends on the relative magnitude of the pair of changes.

Unfortunately, the evaluation of the entropy change for the universe is difficult, and a criterion to be applied to the system itself is preferable, without having to worry about changes in the surroundings. The same entropy definition in the second principle of thermodynamics states

$$\Delta S = \frac{Q_{rev}}{T} \qquad (8.2)$$

where Qrev is the heat transferred from surrounding to system (positive sign) or from system to surrounding (negative sign) in a reversible manner. In a chemical isothermal transformation when heat is exchanged from the system to the surrounding, Eq. (8.2) becomes $\Delta S_{surr} = \frac{-\Delta H_{sys}}{T}$

By substituting the last definition in Eq. (8.1) and multiplying both terms by -T; we obtain an equation, which is valid for an isothermal transformation:

$$-T\Delta S_{univ} = -\Delta H_{sys} - T\Delta S_{sys} \qquad (8.3)$$

The right side of Eq. (8.3) only involves the system, whereas the left side shows the term corresponding to the criterion of spontaneous process, like chemical reactions. Eq. (8.3) is usually modified by introducing a new thermodynamic function, known as free (or Gibbs) energy, denoted by G. It is defined by the identity $\Delta G = -T\Delta S_{univ}$ (8.4)
At the end, one obtains the identity

$$\Delta G = \Delta H - T\Delta S \qquad (8.5)$$

where, since all the terms refer to system changes, the subscript sys has been omitted.

For a process occurring at constant T and P; the following statements hold TRUE.
1. If ΔG < 0 (negative), the process is spontaneous.
2. If ΔG > 0 (positive), the process is nonspontaneous.
3. If ΔG = 0 (zero), the process is reversible and the system has reached an equilibrium.
At equilibrium, there is no driving force of the reaction, so there is no further reaction and the Gibbs energy must be at a minimum.

8.2 Relationship of ΔG with the equilibrium constant K_P

For a generic reaction at equilibrium, say a gaseous reaction like :

$$aA + bB \Leftrightarrow cC + dD$$

where a,b,c,d are the stoichiometric coefficients of the chemical species A,B,C,D respectively we can express a direct relatioship between ΔG and equilibrium constant K_P :

$$K_P = \frac{P_C^c \cdot P_D^d}{P_A^a \cdot P_B^b}$$

where $P_{\{A,B,C,D\}}$ are the partial pressures of the differen gases. For example, in the case of ammonia synthesis:

$$N_2 + 3H_2 \Leftrightarrow 2NH_3 \qquad K_P = \frac{P_{NH3}^2}{P_{N2} \cdot P_{H2}^3}$$

It follows that, if we know ΔG_0 ,we can use the following equation to compute the equilibrium constant :

$$\Delta G_0(T) = -RT \cdot log(K_P) \qquad K_P = \exp\left(\frac{-\Delta G_0(T)}{RT}\right)$$

This means that standard thermodynamic data can be used as a direct source of equilibrium constants at any temperature. However, since the values of $S_0(T)$ $H_0(T)$ $G_0(T)$ markedly vary with the temperature, particularly the entropy, it is not possible to use their values at the standard temperature of 25°C. The values at the standard temperature can be extrapolated to other temperatures with proper coefficients, like the NASA CEA (Chemical Equilibrium with Applications) interpolation. More informations are here.

$H = R*(-a1/T + a2*\log(T) + a3*T + a4*T^2/2 + a5*T^3/3 + a6*T^4/4 + a7*T^5/5 + a8)$

$S = R*(-a1/2/T^2 - a2/T + a3*\log(T) + a4*T + a5*T^2/2 + a6*T^3/3 + a7*T^4/4 + a9)$

where a1...a9 are the nine coefficients to be found on NASA-CEA web site

8.3 Ammonia synthesis

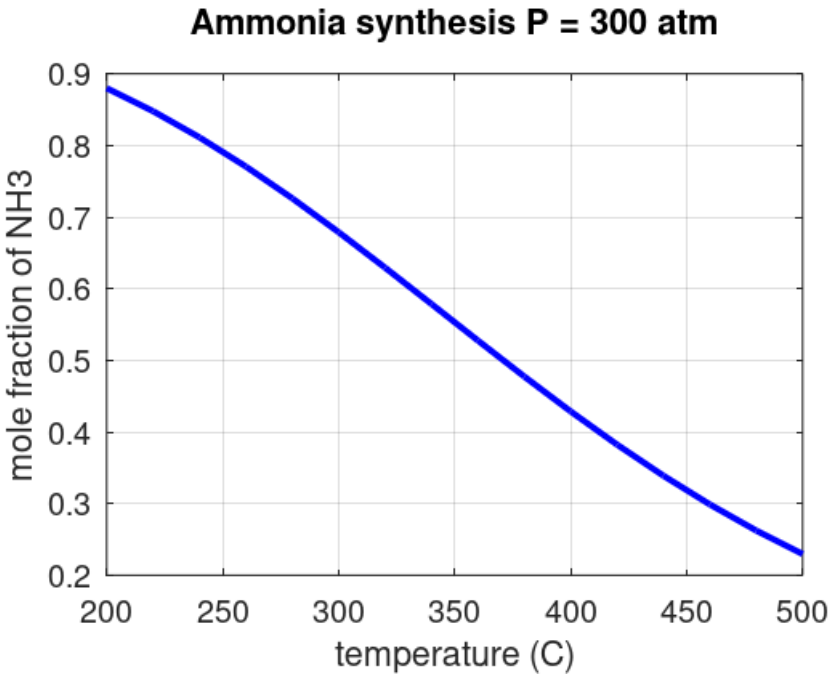
Reaction : $N_2 + 3H_2 \Leftrightarrow 2NH_3$

Scheme, using 'x' as unknown reacted moles of nitrogen

	N2	H2	NH3	Total
Initial moles	1	3	0	4
Equilibrium moles	1-x	3-3x	2x	4-2x
Mole fraction (equil.)	(1-x)/(4-2x)	(3-3x)/(4-2x)	2x/(4-2x)	1
Partial Pressure	$P_{N2} = P_T \cdot (1-x)/(4-2x)$	$P_{H2} = P_T \cdot (3-3x)/(4-2x)$	$P_{NH3} = P_T \cdot (2x)/(4-2x)$	P_T

$$K_P = K_{eq} = \exp(-\Delta G/RT) = \frac{P^2_{(NH3)}}{P_{N2} \cdot P^3_{H2}} = P_T^{-2} \cdot \left(\frac{1-x}{4-2x}\right)^2 \cdot \left(\frac{3-3x}{4-2x}\right)^{-3} \cdot \left(\frac{1-x}{4-2x}\right)^{-1}$$

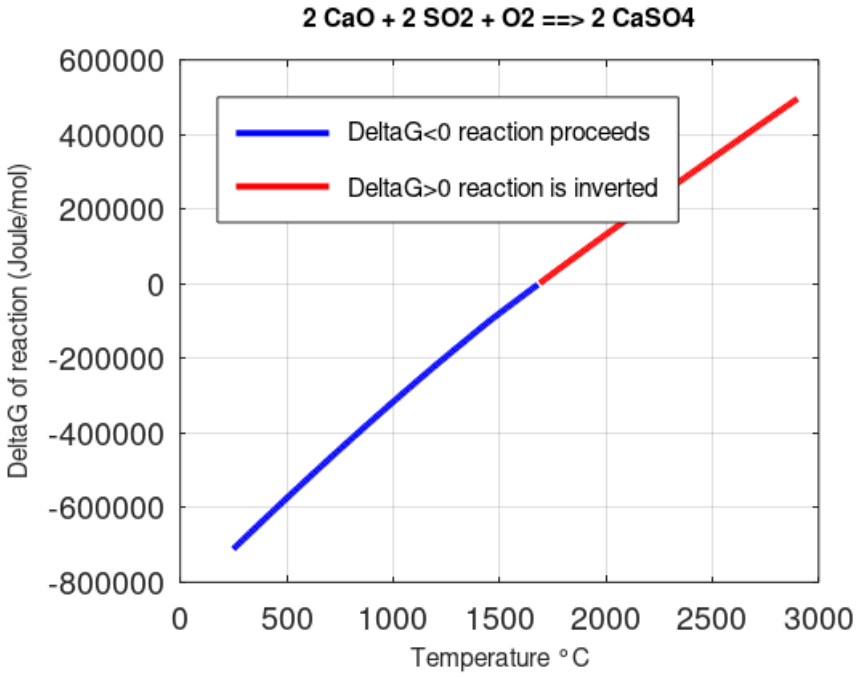
$$K_{eq} = [16x^2 * (2-x)^2] / [27 * (1-x)^4 * P_T^2] \quad ; \quad \Delta G = \Delta H - T \cdot \Delta S$$




```
%    REACTION  :   3 H2 + N2  <==>  2 NH3  (gas state) (Fritz Haber synthesis)
%THERMOCHEMICAL DATA ELABORATED FROM NASA-CEA DATABASE, NINE COEFFICIENT INTERPOLATION
%   H = R*(-a1/T + a2*log(T) + a3*T + a4*T^2/2 + a5*T^3/3 + a6*T^4/4 + a7*T^5/5 + a8)
%   S = R*(-a1/2/T^2 - a2/T + a3*log(T) + a4*T + a5*T^2/2 + a6*T^3/3 + a7*T^4/4 + a9)
%       SOURCE IS  https://www.grc.nasa.gov/www/CEAWeb/ceaThermoBuild.htm
%       TEMPERATURE RANGE 200 - 1000 K (available data cover up to 6000 K)
clear;clc;
H2 = [4.078323210e04,-8.009186040e02,8.214702010,-1.269714457e-02,1.753605076e-05,-
1.202860270e-08,3.368093490e-12,2.682484665e03,-3.043788844e01];
N2 = [2.210371497e04,-3.818461820e02,6.082738360,-8.530914410e-03,1.384646189e-05,-
9.625793620e-09,2.519705809e-12,7.108460860e02,-1.076003744e01];
NH3 = [-7.681226150e04,1.270951578e03,-3.893229130,2.145988418e-02,-2.183766703e-
05,1.317385706e-08,-3.332322060e-12,-1.264886413e04,4.366014588e01];
R = 8.314472;
% The NASA coefficients are treated as vector, so they are summed according to
stoichiometry of the reaction.
```

```
Rx = 2*NH3 - 3*H2 - N2;
disp(' THERMOCHEMICAL DATA ELABORATED FROM
https://www.grc.nasa.gov/www/CEAWeb/ceaThermoBuild.htm');
disp('           REACTION IS :  3 H2 + N2  <==>  2 NH3  (gas state)');
disp('  ');
disp('   Temp. (°C)   Pressure     DeltaH       DeltaS       DeltaG       Kp
Mole fraction NH3');
format shortE;format compact;
Ptot = 300;j=0;
for T1 = 200:20:500
  T = T1 + 273.15;
  TxH = [-1/T,log(T),T,T^2/2,T^3/3,T^4/4,T^5/5,1,0];
  TxS = [-1/T^2/2,-1/T,log(T),T,T^2/2,T^3/3,T^4/4,0,1];
  %Now calculate DeltaH (Enthalpy variation of reaction), DeltaS and therefrom DeltaG
  %Free Energy variation of the reaction.
  DeltaH = R*sum(TxH.*Rx);
  DeltaS = R*sum(TxS.*Rx);
  DeltaG = DeltaH - T*DeltaS;
  % From thermodynamics, Kp (equilibrium constant) derives from DeltaG
  Kp = exp(-DeltaG/R/T);
  % Solve here equilibrium from stoichiometric mixture of N2 and H2
  % Start   H2 = 3 mol  (volume does not matter)
  %         N2 = 1 mol
  %         NH3 = 0 mol
  % Equilibrium H2 = 3-3x; N2 = 1-x; NH3 = 2x;  Total moles = 4 - 2x
  % Keq = [16x^2 * (2-x)^2] / [27*(1-x)^4*Ptot^2]
  % This 4th degree equation is solved by roots(c)
  P = Kp*Ptot^2;
  c = [(27*P - 16),(64 - 108*P),(162*P - 64),(-108*P),(27*P)];
  S = roots(c);
  Xsol=S(4); % S(4) is the only real solution conprised between 0 and 1
  Mole_fract_NH3 = Xsol/(2 - Xsol);
  disp([T1,Ptot,DeltaH,DeltaS,DeltaG,Kp,Mole_fract_NH3]);
  j = j + 1;
  a(j) = T1;b(j) = Mole_fract_NH3;
end
disp('-->Pressure in atm -->DeltaH,DeltaS and DeltaG in J/mol')
plot(a,b,'b','LineWidth',2) ;grid on;xlabel('temperature (C)');
ylabel('mole fraction of NH3');title('Ammonia synthesis P = 300 atm');
```

8.4 Clean coal combustion with CaSO4 formation




```
% Let us examine clean coal combustion with CaSO4 formation from lime and impurity
sulphur.

clear;clc;format short;format compact;

CaO_500_3172 =[-1.45937644e+05,0.0,7.174205094,-1.959947129e-03,1.291116374e-06,-
2.077091735e-10,0.0,-7.89152508e+04,-3.658562837e+01];

SO2_200_1000 =[-5.310842140e+04,9.090311670e+02,-2.356891244,2.204449885e-02,-
2.510781471e-05,1.446300484e-08,-3.36907094e-12,-4.11375208e+04,4.045512519e+01];

SO2_1000_6000 =[-1.127640116e+05,-8.25226138e+02,7.61617863,-1.99932761e-
04,5.65563143e-08,-5.45431661e-12,2.918294102e-16,-3.35130869e+04,-1.655776085e+01];

O2_200_1000 =[-3.42556342e+04,4.84700097e+02,1.119010961,4.29388924e-03,-6.83630052e-
07,-2.0233727e-09,1.039040018e-12,-3.39145487e+03,1.849699470e+01];

O2_1000_6000 = [-1.037939022e+06,2.344830282e+03,1.819732036,1.267847582e-03,-
2.188067988e-07,2.053719572e-11,-8.193467050e-16,-1.689010929e+04,1.738716506e+01];

CaSO4_500_1473 =[-2.983940124e+05,0.0,1.359671225e+01,5.857230312e-03,0.0,0.0,0.0,-
1.777846313e+05,-6.802481458e+01];

CaSO4_1473_1733 =[0.0,0.0,1.984482549e+01,0.0,0.0,0.0,0.0,-1.798298668e+05,-
1.045004358e+02];

CaSO4_1733_6000 =[0.0,0.0,1.984482549e+01,0.0,0.0,0.0,0.0,-1.764622601e+05,-
1.025572120e+02];


R = 8.314472;

% Reaction is 2 CaO + 2 SO2 + O2 <==> 2 CaSO4

% The NASA coefficients are treated as vector, so they are summed according to
stoichiometry of the reaction.

  Rx1_low = 2*CaSO4_500_1473 - 2*CaO_500_3172 - 2*SO2_200_1000 - O2_200_1000; % 500-
1000K

  Rx1_mid = 2*CaSO4_500_1473 - 2*CaO_500_3172 - 2*SO2_1000_6000 - O2_1000_6000; %
1000-1473K
```

```
  Rx1_high = 2*CaSO4_1473_1733 - 2*CaO_500_3172 - 2*SO2_1000_6000 - O2_1000_6000; %
1473-1733K

  Rx1_liq = 2*CaSO4_1733_6000 - 2*CaO_500_3172 - 2*SO2_1000_6000 - O2_1000_6000;  %
1733-3172K

j=0;

% start of the heating cycle.

for Tc = (250:10:2900)

  j = j + 1; T = Tc + 273.15;

  a(j) = Tc;

  TxH = [-1/T,log(T),T,T^2/2,T^3/3,T^4/4,T^5/5,1,0];

  TxS = [-1/T^2/2,-1/T,log(T),T,T^2/2,T^3/3,T^4/4,0,1];

  % Now calculate DeltaH (Enthalpy variation of reaction), DeltaS and therefrom
DeltaG.

  % From thermodynamics, Kp (equilibrium constant) derives from DeltaG.According to
temperature.

  switch true  % reaction n°2  CaO + SO3 <==> CaSO4

      case 500<=T && T<1000

          DeltaH = R*sum(TxH.*Rx1_low);DeltaS = R*sum(TxS.*Rx1_low);DeltaG(j) = DeltaH
- T*DeltaS;

      case 1000<=T && T<1473

          DeltaH = R*sum(TxH.*Rx1_mid);DeltaS = R*sum(TxS.*Rx1_mid);DeltaG(j) = DeltaH
- T*DeltaS;

      case 1473<=T && T<1733

          DeltaH = R*sum(TxH.*Rx1_high);DeltaS = R*sum(TxS.*Rx1_high);DeltaG(j) =
DeltaH - T*DeltaS;

      case 1733<= T

          DeltaH = R*sum(TxH.*Rx1_liq);DeltaS = R*sum(TxS.*Rx1_liq);DeltaG(j) = DeltaH
- T*DeltaS;

  end

  if DeltaG<0;h1 = j;end

end

% transpose the vectors, in order to be nicely printed

DeltaG = DeltaG.';a = a.';

D1 = DeltaG(1:h1);D2 = DeltaG(h1+1:j);

T1 = a(1:h1);T2 = a(h1+1:j);

plot(T1,D1,'b','LineWidth',2);grid on;hold on;

plot(T2,D2,'r','LineWidth',2)

xlabel('Temperature °C','fontsize',8);

ylabel('DeltaG of reaction (Joule/mol)','fontsize',8');

title('2 CaO + 2 SO2 + O2 ==> 2 CaSO4','fontsize',8);
```

```
legend('DeltaG<0 reaction proceeds','DeltaG>0 reaction is
inverted','Location','northwest','fontsize',8);
```

8.5 Syn-gas production

Reaction : CH4 (gas) + H2O (gas) ↔ CO (gas) + 3H2 (gas)

Scheme, using '*x*' as unknown reacted moles of methane reacted

	CH4	H2O	CO	H2	Total
Initial moles	1	1	0	0	2
Equilibrium moles	1-x	1-x	x	3x	2+2x
Mole fraction (equil.)	(1-x)/(2+2x)	(1-x)/(2+2x)	x/(2+2x)	3x/(2+2x)	1

$$K_P = K_{eq} = \exp(-\Delta G/RT) = \frac{P_{CO}P_{H2}^3}{P_{CH4}\cdot P_{H2O}} = P_T^2\cdot\left(\frac{x}{2+2x}\right)\cdot\left(\frac{3x}{2+2x}\right)^3\left(\frac{1-x}{2+2x}\right)^{-1}\cdot\left(\frac{1-x}{2+2x}\right)^{-1}$$

$$(4*Kp-27*P^2)*x^4-8*Kp*x^2+4*Kp=0 \quad ; \quad \Delta G=\Delta H-T\cdot\Delta S$$

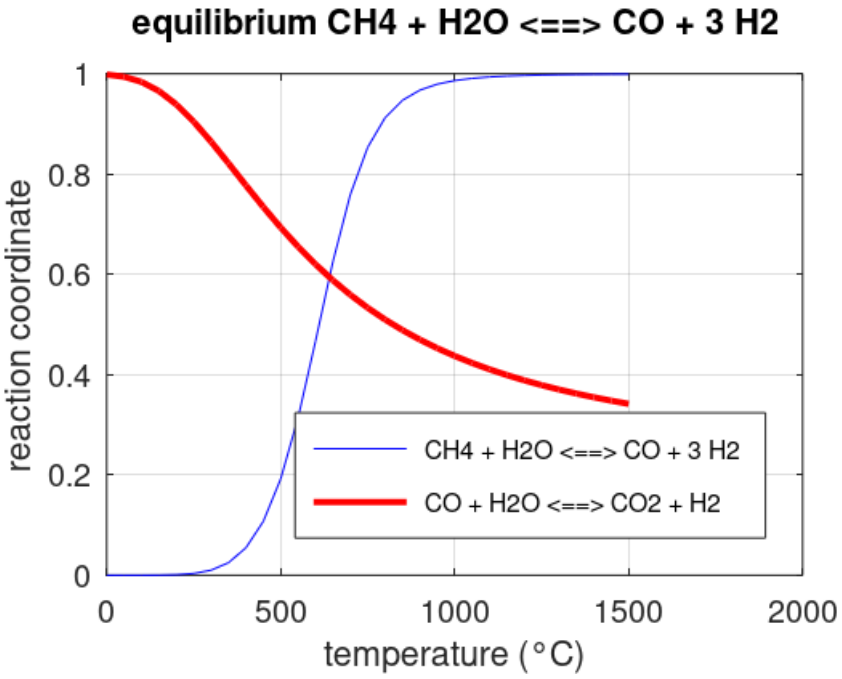
Reaction : CO (gas) + H2O (gas) ↔ CO2 (gas) + H2 (gas

Scheme, using '*x*' as unknown reacted moles of CO

	CO	H2O	CO2	H2	Total
Initial moles	1	1	0	0	2
Equilibrium moles	1-x	1-x	x	x	2
Mole fraction (equil.)	(1-x)/2	(1-x)/2	x/2	x/2	1

$$K_P = K_{eq} = \exp(-\Delta G/RT) = \frac{P_{CO2}\cdot P_{H2}}{P_{CO}\cdot P_{H2O}} = \cdot\left(\frac{1-x}{2}\right)\cdot\left(\frac{1-x}{2}\right)\cdot\left(\frac{x}{2}\right)^{-1}\cdot\left(\frac{x}{2}\right)^{-1}$$

$$(Kp-1)*x^2-2*Kp*x+Kp=0 \quad ; \quad \Delta G=\Delta H-T\cdot\Delta S$$

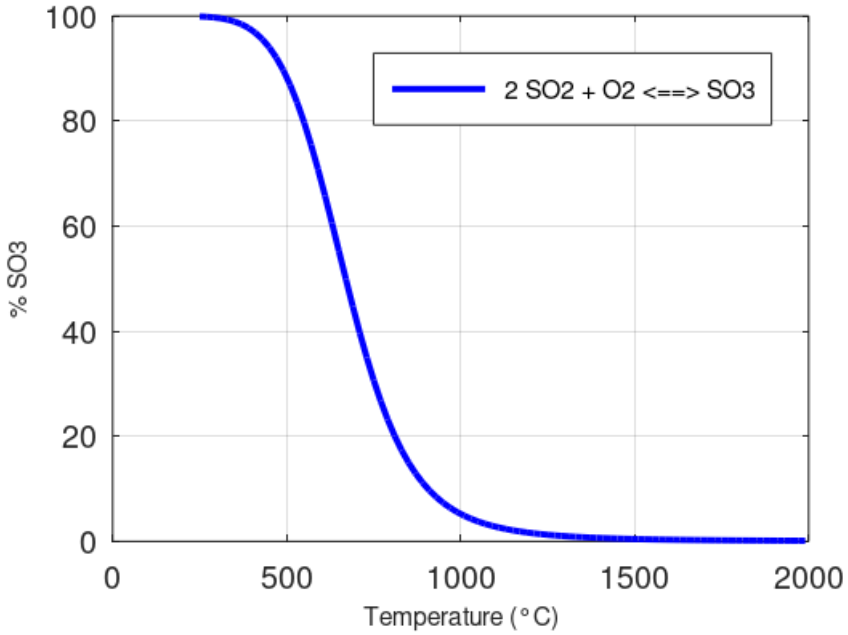



```
%REACTIONS: CH4 + H2O <==> CO + 3 H2 (usually in industrial plants from 700 to 1100°C)
%                     CO + H2O <==> CO2 + H2 (at lower temperatures)
```

```
%THERMOCHEMICAL DATA ELABORATED FROM NASA-CEA DATABASE, NINE COEFFICIENT INTERPOLATION
%   H = R*(-a1/T + a2*log(T) + a3*T + a4*T^2/2 + a5*T^3/3 + a6*T^4/4 + a7*T^5/5 + a8)
%   S = R*(-a1/2/T^2 - a2/T + a3*log(T) + a4*T + a5*T^2/2 + a6*T^3/3 + a7*T^4/4 + a9)
%SOURCE IS  https://cearun.grc.nasa.gov/ThermoBuild/ ; TEMPERATURE RANGE 1000 - 6000 K
% H2
clear;clc;
H2_200_1000 = [4.078323210e+04,-8.009186040e+02,8.214702010,-1.269714457e-
02,1.753605076e-05,-1.202860270e-08,3.368093490e-12,2.682484665e+03,-3.043788844e+01];
H2_1000_6000 = [5.608128010e+05,-8.371504740e+02,2.975364532,1.252249124e-03,-
3.740716190e-07,5.936625200e-11,-3.606994100e-15,5.339824410e+03,-2.202774769];
% H2O
H2O_200_1000 = [-3.947960830e+04,5.755731020e+02,9.317826530e-01,7.222712860e-03,-
7.342557370e-06,4.955043490e-09,-1.336933246e-12,-3.303974310e+04,1.724205775e+01];
H2O_1000_6000 = [1.034972096e+06,-2.412698562e+03,4.646110780,2.291998307e-03,-
6.836830480e-07,9.426468930e-11,-4.822380530e-15,-1.384286509e+04,-7.978148510];
% CO
CO_200_1000 = [1.489045326e+04,-2.922285939e+02,5.724527170,-8.176235030e-
03,1.456903469e-05,-1.087746302e-08,3.027941827e-12,-1.303131878e+04,-7.859241350];
CO_1000_6000 = [4.619197250e+05,-1.944704863e+03,5.916714180,-5.664282830e-
04,1.398814540e-07,-1.787680361e-11,9.620935570e-16,-2.466261084e+03,-
1.387413108e+01];
% CO2
CO2_200_1000 = [4.943650540e+04,-6.264116010e+02,5.301725240,2.503813816e-03,-
2.127308728e-07,-7.689988780e-10,2.849677801e-13,-4.528198460e+04,-7.048279440];
CO2_1000_6000 = [1.176962419e+05,-1.788791477e+03,8.291523190,-9.223156780e-
05,4.863676880e-09,-1.891053312e-12,6.330036590e-16,-3.908350590e+04,-
2.652669281e+01];
% CH4
CH4_200_1000 = [-1.766850998e+05,2.786181020e+03,-1.202577850e+01,3.917619290e-02,-
3.619054430e-05,2.026853043e-08,-4.976705490e-12,-2.331314360e+04,8.904322750e+01];
CH4_1000_6000 = [3.730042760e+06,-1.383501485e+04,2.049107091e+01,-1.961974759e-
03,4.727313040e-07,-3.728814690e-11,1.623737207e-15,7.532066910e+04,-1.219124889e+02];

R = 8.314472;
format shortE;format compact;

%disp('   Temp. (°C)     Kp         xCH4');
% The NASA coefficients are treated as vector, so they are summed according to
stoichiometry of the reaction.
% CH4 + H2O <==> CO + 3 H2
  Rx1_low = CO_200_1000 + 3*H2_200_1000 - CH4_200_1000 - H2O_200_1000;
  Rx1_high = CO_1000_6000 + 3*H2_1000_6000 - CH4_1000_6000 - H2O_1000_6000;
% CO + H2O <==> CO2 + H2
  Rx2_low = CO2_200_1000 + H2_200_1000 - CO_200_1000 - H2O_200_1000;
  Rx2_high = CO2_1000_6000 + H2_1000_6000 - CO_1000_6000 - H2O_1000_6000;
j=0;
P = 1; % pressure in atm.   <it can be varied>
% start of the heating cycle.
for T1 = (0:50:1500)
  j = j + 1;
  T = T1 + 273.15;a(j) = T1;
  TxH = [-1/T,log(T),T,T^2/2,T^3/3,T^4/4,T^5/5,1,0];
  TxS = [-1/T^2/2,-1/T,log(T),T,T^2/2,T^3/3,T^4/4,0,1];
  % Now calculate DeltaH (Enthalpy variation of reaction), DeltaS and therefrom
DeltaG.
  % From thermodynamics, Kp (equilibrium constant) derives from DeltaG.
  % According to temperature (>1000 or <1000) , different values are chosen.
```

```
  switch T<1000
      case 1
          DeltaH1 = R*sum(TxH.*Rx1_low);DeltaS1 = R*sum(TxS.*Rx1_low);DeltaG1 =
DeltaH1 - T*DeltaS1;
          DeltaH2 = R*sum(TxH.*Rx2_low);DeltaS2 = R*sum(TxS.*Rx2_low);DeltaG2 =
DeltaH2 - T*DeltaS2;
      case 0
          DeltaH1 = R*sum(TxH.*Rx1_high);DeltaS1 = R*sum(TxS.*Rx1_high);DeltaG1 =
DeltaH1 - T*DeltaS1;
          DeltaH2 = R*sum(TxH.*Rx2_high);DeltaS2 = R*sum(TxS.*Rx2_high);DeltaG2 =
DeltaH2 - T*DeltaS2;
  end
  % Solve here (with roots) the two equilibria.
  Kp1 = exp(-DeltaG1/R/T); % CH4 + H2O <==> CO + 3 H2
  % (4*Kp1 - 27*P^2)*x^4 - 8*Kp1*x^2 + 4*Kp1 = 0
  S = roots([4*Kp1 - 27*P^2,0,- 8*Kp1,0,4*Kp1]);
  x1 = S(4);b(j) = x1;
  Kp2 = exp(-DeltaG2/R/T); % CO + H2O <==> CO2 + H2
  % (Kp2 - 1)*x^2 - 2*Kp2*x + Kp2 = 0
  S = roots([Kp2-1,-2*Kp2,Kp2])
  x2 = S(2);c(j) = x2;
  disp([T1,x1,x2,DeltaG1,DeltaG2]);
  end
disp('-->Pressure(atm), DeltaH(J/mol), DeltaS(J/mol/K, DeltaG(J/mol)');
plot(a,b,'b',a,c,'r','LineWidth',2) ;grid on;
xlabel('temperature (°C)')
ylabel('reaction coordinate')
title('equilibrium CH4 + H2O <==> CO + 3 H2')
legend('CH4 + H2O <==> CO + 3 H2','CO + H2O <==> CO2 +
H2','fontsize',8,'Location','southeast')
```

8.6 SO2/SO3 equilibrium in gas phase

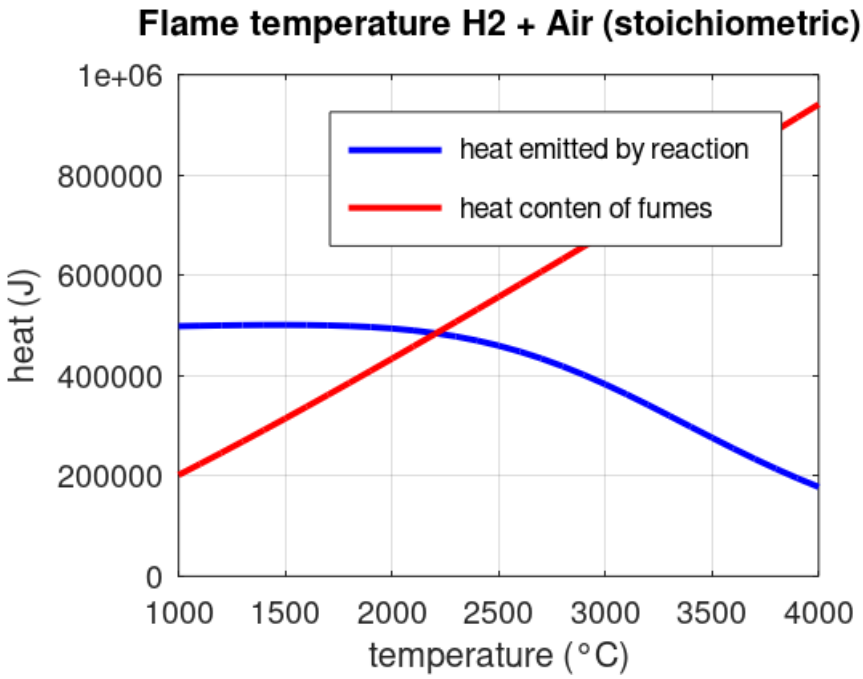



```
clear;clc;format short;format compact;

SO2_200_1000 =[-5.310842140e+04,9.090311670e+02,-2.356891244,2.204449885e-02,-
2.510781471e-05,1.446300484e-08,-3.36907094e-12,-4.11375208e+04,4.045512519e+01];
SO2_1000_6000 =[-1.127640116e+05,-8.25226138e+02,7.61617863,-1.99932761e-
04,5.65563143e-08,-5.45431661e-12,2.918294102e-16,-3.35130869e+04,-1.655776085e+01];
SO3_200_1000 =[-3.952855290e+04,6.20857257e+02,-1.437731716,2.764126467e-02,-
3.144958662e-05,1.792798e-08,-4.12638666e-12,-5.18410617e+04,3.391331216e+01];
SO3_1000_6000 =[-2.166923781e+05,-1.301022399e+03,1.096287985e+01,-3.83710002e-
04,8.46688904e-08,-9.70539929e-12,4.49839754e-16,-4.39828399e+04,-3.655217314e+01];
O2_200_1000 =[-3.42556342e+04,4.84700097e+02,1.119010961,4.29388924e-03,-6.83630052e-
07,-2.0233727e-09,1.039040018e-12,-3.39145487e+03,1.849699470e+01];
O2_1000_6000 = [-1.037939022e+06,2.344830282e+03,1.819732036,1.267847582e-03,-
2.188067988e-07,2.053719572e-11,-8.193467050e-16,-1.689010929e+04,1.738716506e+01];

R = 8.314472;j = 0;
% Reaction is 2 SO2 + O2 <==> 2 SO3
% The NASA coefficients are treated as vector, so they are summed according to
stoichiometry of the reaction.
  Rx1_low = 2*SO3_200_1000 - 2*SO2_200_1000 - O2_200_1000; % 200-1000K
  Rx1_high = 2*SO3_1000_6000 - 2*SO2_1000_6000 - O2_1000_6000; % 1000-6000K
  P = 1; % <total pressure in atm. It can be varied
% start of the heating cycle.
for Tc = (250:20:2000)
  j = j + 1; T = Tc + 273.15;
  a(j) = Tc;
  TxH = [-1/T,log(T),T,T^2/2,T^3/3,T^4/4,T^5/5,1,0];
  TxS = [-1/T^2/2,-1/T,log(T),T,T^2/2,T^3/3,T^4/4,0,1];
  % Now calculate DeltaH (Enthalpy variation of reaction), DeltaS and therefrom
DeltaG.
  % From thermodynamics, Kp (equilibrium constant) derives from DeltaG.According to
temperature.
  switch true
      case T<1000
          DeltaH = R*sum(TxH.*Rx1_low);DeltaS = R*sum(TxS.*Rx1_low);DeltaG = DeltaH -
T*DeltaS;
```

```
      case 1000<=T
          DeltaH = R*sum(TxH.*Rx1_high);DeltaS = R*sum(TxS.*Rx1_high);DeltaG = DeltaH
- T*DeltaS;
  end
  % Solve here (with roots) the equilibrium.
  Kp = exp(-DeltaG/R/T); % Kp = p(SO3)^2 / (p(SO2)^2 * p(O2))
  % (1 - Kp*P)*x^3 + (3*Kp*P - 3)*x^2 - 3*Kp*P*x + Kp*P = 0;
  S = roots([1 - Kp*P,3*Kp*P - 3, -3*Kp*P, Kp*P]);
  x1 = S(3);b(j) = 2*x1/(3-x1)*100;
end
plot(a,b,'b','LineWidth',2);grid on;xlabel('Temperature (°C)','fontsize',8);
ylabel('% SO3','fontsize',8);legend('2 SO2 + O2 <==> SO3','fontsize',8)
```

8.7 Hydrogen combustion and flame temperature

Here the heat generated by the combustion is used to heat up the fumes (gaseous combustion products, including N2 from air). Part of this heat is absorbed by the endothermic H2O(vap) decomposition into H2 + O2.

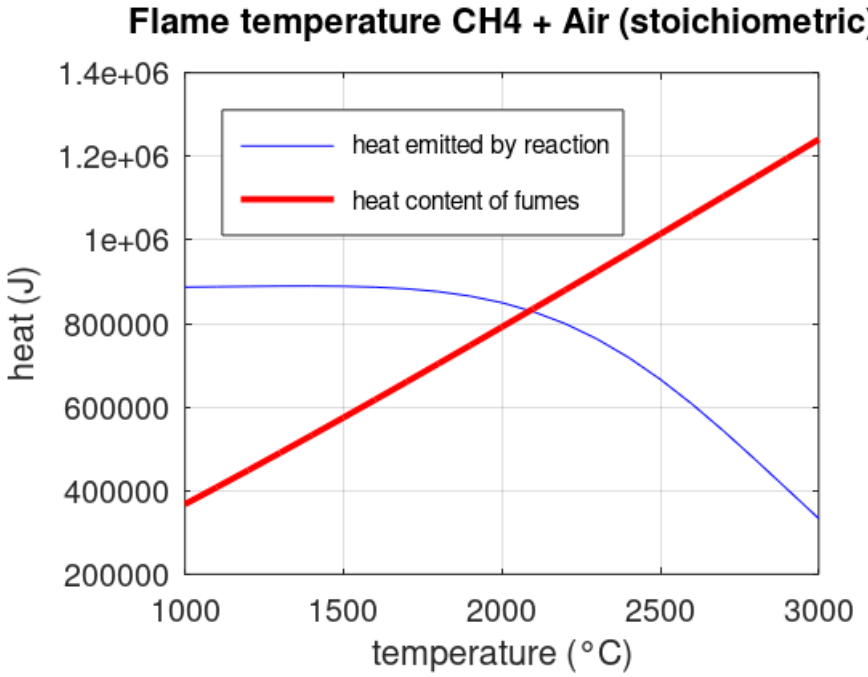



```
%       REACTION  :   2 H2 + O2  <==>  2 H2O  (gas state)
%THERMOCHEMICAL DATA ELABORATED FROM NASA-CEA DATABASE, NINE COEFFICIENT INTERPOLATION
%   H = R*(-a1/T + a2*log(T) + a3*T + a4*T^2/2 + a5*T^3/3 + a6*T^4/4 + a7*T^5/5 + a8)
%   S = R*(-a1/2/T^2 - a2/T + a3*log(T) + a4*T + a5*T^2/2 + a6*T^3/3 + a7*T^4/4 + a9)
%       SOURCE IS  https://www.grc.nasa.gov/www/CEAWeb/ceaThermoBuild.htm
%       TEMPERATURE RANGE 1000 - 6000 K
clear;clc;R = 8.314472;

H2 = [5.608128010e+05,-8.371504740e+02,2.975364532,1.252249124e-03,-3.740716190e-
07,5.936625200e-11,-3.606994100e-15,5.339824410e+03,-2.202774769];
O2 = [-1.037939022e+06,2.344830282e+03,1.819732036,1.267847582e-03,-2.188067988e-
07,2.053719572e-11,-8.193467050e-16,-1.689010929e+04,1.738716506e+01];
H2O = [1.034972096e+06,-2.412698562e+03,4.646110780,2.291998307e-03,-6.836830480e-
07,9.426468930e-11,-4.822380530e-15,-1.384286509e+04,-7.978148510];
N2 = [5.877124060e+05,-2.239249073e+03,6.066949220,-6.139685500e-04,1.491806679e-07,-
1.923105485e-11,1.061954386e-15,1.283210415e+04,-1.586640027e+01];
Ar = [2.010538475e+01,-5.992661070e-02,2.500069401,-3.992141160e-08,1.205272140e-11,-
1.819015576e-15,1.078576636e-19,-7.449939610e+02,4.379180110];

% The NASA coefficients are treated as a vector, so they are summed according to
stoichiometry of the reaction.
% 2H2 + O2 --> 2H2O
Rx1 = 2*H2O - 2*H2 - O2;
% Heat content of gaseous reaction products (fumes). Air is used as oxygen source.
% 3.714 N2 + 0.0476 Ar + O2 + 2 H2 --> 3.714 N2 + 0.0476 Ar + 2H2O(gas)
Rx2 = 3.714*N2 + 0.0476*Ar + 2*H2O;
disp('   Temp. (°C)   Pressure     DeltaH       DeltaS       DeltaG       Kp
Heat Out     2H2 + O2 --> 2H2O reaction yield(%)');
format shortE;format compact;
j=0;T = 298.15;
TxH = [-1/T,log(T),T,T^2/2,T^3/3,T^4/4,T^5/5,1,0];
```

```
H0 = R*sum(TxH.*Rx2); % heat content of fumes @ 25癈
for T1 = (1000:100:4000)
  j = j + 1;a(j) = T1;
  T = T1 + 273.15;
  TxH = [-1/T,log(T),T,T^2/2,T^3/3,T^4/4,T^5/5,1,0];
  TxS = [-1/T^2/2,-1/T,log(T),T,T^2/2,T^3/3,T^4/4,0,1];
  %Now calculate DeltaH (Enthalpy variation of reaction), DeltaS and therefrom DeltaG
  %DeltaG = (Gibbs) free energy variation of the reaction.
  DeltaH = R*sum(TxH.*Rx1);
  DeltaS = R*sum(TxS.*Rx1);
  DeltaG = DeltaH - T*DeltaS;
  % From thermodynamics, Kp (equilibrium constant) derives from DeltaG
  Kp = exp(-DeltaG/R/T);
  % We solve here equilibrium for water synthesis reaction
  %         H2          O2          H2O(gas)    Total
  % Start   2 mol       1 mol       0 mol       3 mol
  % Equil.  2-2x mol    1-x mol     2x mol      3-x mol
  % Kp = (2x/(3-x))^2 / ((2-2x)/(3-x))^2 / (1-x)/(3-x) / Ptot
  % (1-Kp*P)x^3 + (3Kp*P-3)x^2 - 3Kp*P*x + Kp*P = 0
  % This 3th degree equation is solved by roots();supposing Ptot =1 atm.
  P = 1; % total pressure in atm.
  S = roots([1 - Kp*P, 3*Kp*P - 3, - 3*Kp*P, Kp*P]);
  b(j) = -S(3)*DeltaH;  %convention of signs
  % Now we compute heat capacity of the fumes.
  Heat = R*sum(TxH.*Rx2) - H0;c(j) = Heat;
  disp([T1,P,DeltaH,DeltaS,DeltaG,Kp,Heat,S(3)*100]);
end
p1 = polyfit(a,b,3);
p2 = polyfit(a,c,3);
p = p1 - p2;
S = roots([p(1), p(2), p(3), p(4)]);
Tsol = S(1);
disp(['Flame temperature (°C) = ',num2str(Tsol)]);
disp('-->Pressure in atm  -->DeltaH, DeltaG in J/mol  -->DeltaS in J/mol/K')
plot(a,b,'b','LineWidth',2,a,c,'r','LineWidth',2) ;grid on;
xlabel('temperature (°C)');ylabel('heat (J)');title('Flame temperature H2 + Air
(stoichiometric)');
legend('heat emitted by reaction','heat conten of fumes');
```

8.8 CH_4 combustion and flame temperature

Here the heat generated by the combustion is used to heat up the fumes (gaseous combustion products, including N2 from air). Part of this heat is absorbed by the endothermic CO2 and H2O decomposition into CO + O2 and H2 + O2 respectively.




```
% REACTION  :   CH4(gas) + 2 O2  -->  CO2 + 2 H2O  (gas state) -- Methane combustion
%       Thermal dissociation at high temperatures is considered:
%       2H2O <==> 2H2 + O2    ;    2CO2 <==> 2CO + O2
%THERMOCHEMICAL DATA ELABORATED FROM NASA-CEA DATABASE, NINE COEFFICIENT INTERPOLATION
% H = R*(-a1/T + a2*log(T) + a3*T + a4*T^2/2 + a5*T^3/3 + a6*T^4/4 + a7*T^5/5 + a8)
% S = R*(-a1/2/T^2 - a2/T + a3*log(T) + a4*T + a5*T^2/2 + a6*T^3/3 + a7*T^4/4 + a9)
%       SOURCE IS  https://www.grc.nasa.gov/www/CEAWeb/ceaThermoBuild.htm ;
TEMPERATURE RANGE 1000 - 6000 K
clear;clc;
% H2                Ref-Elm. Gurvich,1978 pt1 p103 pt2 p31.
H2_1000_6000 = [5.608128010e+05,-8.371504740e+02,2.975364532,1.252249124e-03,-
3.740716190e-07,5.936625200e-11,-3.606994100e-15,5.339824410e+03,-2.202774769];
% O2                Ref-Elm. Gurvich,1989 pt1 p94 pt2 p9.
O2_1000_6000 = [-1.037939022e+06,2.344830282e+03,1.819732036,1.267847582e-03,-
2.188067988e-07,2.053719572e-11,-8.193467050e-16,-1.689010929e+04,1.738716506e+01];
% H2O               Hf:Cox,1989. Woolley,1987. TRC(10/88) tuv25.
H2O_1000_6000 = [1.034972096e+06,-2.412698562e+03,4.646110780,2.291998307e-03,-
6.836830480e-07,9.426468930e-11,-4.822380530e-15,-1.384286509e+04,-7.978148510];
% N2
N2_1000_6000 = [5.877124060e+05,-2.239249073e+03,6.066949220,-6.139685500e-
04,1.491806679e-07,-1.923105485e-11,1.061954386e-15,1.283210415e+04,-1.586640027e+01];
% Ar
Ar_1000_6000 = [2.010538475e+01,-5.992661070e-02,2.500069401,-3.992141160e-
08,1.205272140e-11,-1.819015576e-15,1.078576636e-19,-7.449939610e+02,4.379180110];
%CO
CO_1000_6000 = [4.619197250e+05,-1.944704863e+03,5.916714180,-5.664282830e-
04,1.398814540e-07,-1.787680361e-11,9.620935570e-16,-2.466261084e+03,-
1.387413108e+01];
%CO2
CO2_1000_6000 = [1.176962419e+05,-1.788791477e+03,8.291523190,-9.223156780e-
05,4.863676880e-09,-1.891053312e-12,6.330036590e-16,-3.908350590e+04,-
2.652669281e+01];
%CH4
CH4_1000_6000 = [3.730042760e+06,-1.383501485e+04,2.049107091e+01,-1.961974759e-
03,4.727313040e-07,-3.728814690e-11,1.623737207e-15,7.532066910e+04,-1.219124889e+02];
R = 8.314472;
```

```
% The NASA coefficients are treated as vector, so they are summed according to
stoichiometry of the reaction.
% CH4 + 2 O2 --> CO2 + 2 H2O
  Rx1 = CO2_1000_6000 + 2*H2O_1000_6000 - CH4_1000_6000 - 2*O2_1000_6000;
% 2 CO2 <==> 2 CO + O2
  Rx2 = 2*CO_1000_6000 + O2_1000_6000 - 2*CO2_1000_6000;
% 2 H2O <==> 2 H2 + O2
  Rx3 = 2*H2_1000_6000 + O2_1000_6000 - 2*H2O_1000_6000;
% Heat of fumes
  Rx4 = 7.429*N2_1000_6000 + 0.0952*Ar_1000_6000 + 2*H2O_1000_6000 + CO2_1000_6000;
% heat of intake air
  Rx5 = 7.429*N2_1000_6000 + 0.0952*Ar_1000_6000 + 2*O2_1000_6000;
disp('   Temp. (°C)     heat1      heat 2     dissCO2      dissH2O');
format shortE;format compact;
j=0;
P = 1; % pressure in atm.   <it can be varied>
T = 298.15;
TxH = [-1/T,log(T),T,T^2/2,T^3/3,T^4/4,T^5/5,1,0];
heat0 = R*sum(TxH.*Rx4); % heat content of fumes @ 25°C
heatAir0 = R*sum(TxH.*Rx5); % heat conten of air @25°C
T = 600; % absolute temperature of pre-heated air intake. <it can be varied>
TxH = [-1/T,log(T),T,T^2/2,T^3/3,T^4/4,T^5/5,1,0];
heatAir1 = R*sum(TxH.*Rx5); % heat conten of air @ T=600 K
heatAir = heatAir1 - heatAir0;
% start of the heating cycle.
for T1 = (1000:100:3000)
  j = j + 1;
  T = T1 + 273.15;a(j) = T1;
  TxH = [-1/T,log(T),T,T^2/2,T^3/3,T^4/4,T^5/5,1,0];
  TxS = [-1/T^2/2,-1/T,log(T),T,T^2/2,T^3/3,T^4/4,0,1];
  %Now calculate DeltaH (Enthalpy variation of reaction), DeltaS and therefrom DeltaG
  %Gibbs Energy variation of the reaction.
  DeltaH1 = R*sum(TxH.*Rx1); % methane combustion
  DeltaH2 = R*sum(TxH.*Rx2);DeltaS2 = R*sum(TxS.*Rx2);DeltaG2 = DeltaH2 - T*DeltaS2; %
CO2 diss.
  DeltaH3 = R*sum(TxH.*Rx3);DeltaS3 = R*sum(TxS.*Rx3);DeltaG3 = DeltaH3 - T*DeltaS3; %
H2O diss.
  heat2 = R*sum(TxH.*Rx4) - heat0; % heat of fumes
  c(j) = heat2;
  % From thermodynamics, Kp (equilibrium constant) derives from DeltaG
  KpCO2 = exp(-DeltaG2/R/T);
  KpH2O = exp(-DeltaG3/R/T);
  % Solve here (with roots) the two equilibrium for dissociation of CO2 and H2O
  % 1-dissociation of CO2
  % (KpCO2 - P)*x^3 - 3*KpCO2*x + 2*KpCO2 = 0;
  S = roots([KpCO2 - P, 0 , - 3*KpCO2, 2*KpCO2]);
  xCO2 = S(3);
  % 2-dissociation of H2O
  % (KpH2O - P)*x^3 - 3*KpH2O*x + 2*KpH2O = 0;
  S = roots([KpH2O - P, 0 , - 3*KpH2O, 2*KpH2O]);
  xH2O = S(3);
  % heat absorbed by dissociation of 1 mol CO2 and 2 mol H2O
  heatA_CO2 = xCO2*DeltaH2;
  heatA_H2O = xH2O*2*DeltaH3;
  % heat balance
  heat1 = DeltaH1 + heatA_CO2 + heatA_H2O - heatAir;b(j) = -heat1;
  disp([T1,heat1,heat2,xCO2,xH2O]);
  end
disp('-->Pressure(atm), DeltaH(J/mol), DeltaS(J/mol/K, DeltaG(J/mol)');
p1 = polyfit(a,b,3);
p2 = polyfit(a,c,3);
p = p1 - p2;
S = roots([p(1),p(2),p(3),p(4)])
Tsol = S(3);
disp(['Flame temperature (°C) = ',num2str(Tsol,'%.0f')]);
plot(a,b,'b',a,c,'r','LineWidth',2) ;grid on;
```

```
xlabel('temperature (°C)')
ylabel('heat (J)')
title('Flame temperature CH4 + Air (stoichiometric)')
legend('heat emitted by reaction','heat content of
fumes','fontsize',16,'location','northwest')
```

Chapter 9. Equilibria in aqueous solution

9.1 Introduction

When a strong electrolyte, such as hydrochloric acid, HCl is put into solution, nearly all the molecules dissociate into ions, in this case H^+ and Cl^- . H^+ reacts immediately with a water molecule, therefore H^+ is a shorthand symbol for the hydrated ion $H3O^+$. But when a weak electrolyte is put into solution, such as the acetic acid CH3COOH , only a small fraction of the molecules dissociate. The dissociation reaction is :

$$CH3COOH <===> CH3COO^- + H^+$$

which stands for the complete reaction, including water :

$$CH3COOH + H2O <===> CH3COO^- + H3O^+$$

Because this reaction is at equilibrium, the following expression applies:

$$K_{diss}=\frac{[H^+][CH_3COO^-]}{[CH_3COOH]} \quad . \qquad (1)$$

The equilibrium constant of the ionization of a weak electrolyte is usually denoted by Kdiss and is referred to as the *ionization constant*. Ionization constants are derived by experimental measurements of equilibrium concentrations.

Moreover**,** in practical calculations with weak and very weak acids, self-dissociation of water must be considered, which leads to a system of two equations and four unknowns, $[H^+],[OH^-],[CH_3COO^-]$ and $[CH_3COOH]$. The pair of equations consists of (1) and

$$K_w=[H^+]\cdot[OH^-]=1\times10^{-14} \quad . \qquad (2)$$

Due to four unknowns, we need two equations more, which are derived below, in the case of the hydrofluoric acid *HF* , a simple *monoprotic acid* found for instance in seawater. The Octave script refers to this acid.

9.2 The weak monoprotic acid

First we choose an initial concentration of *HF*, say Atot = 0.18 mol/L, which as soon as the rapid dissociation has reached equilibrium, equals to the sum of [F-] anf [HF] and, which equality provides the first equation

$$Atot = [F-] + [HF]. \qquad (3)$$

The second and third equations derive from the water self-dissociation in and from the hydrofluoric acid dissociation, i.e. from, but rewritten for *HF* as follows

$$K_{diss} = \frac{[H^+][F^-]}{[FH]} \qquad (4)$$

The fourth equation which completes the system, expresses *electrical neutrality*, which should be always accounted for when dealing with electrolyte solutions:

$$\text{neutrality(pH)} = [H+] - [OH-] - [F-] = 0 \qquad (5)$$

where pH = -log10 [H+]

The set of the four equations from 2 to 5 together with the four unknowns [F-],[HF],H+] and [OH-] , could be directly solved in this simple case. To be generic, we search for the value of $0 \le pH \le 14$ which satisfies, in an *iterative way.*

Once the pH value is fixed, indeed, the [H+] ions concentration holds `H = 10^(-pH)` and therefore `[OH-] = Kw/H`. Now we satisfy mass conservation of F element (`1*HF + 1*F = Atot`) and the dissociation equilibrium

$HF \Longleftrightarrow H^+ + F^-$ Kdiss = Ka1 = 3.5e-4 `-Ka1*HF + H*F = 0`

can be solved. A set of two linear equation is then solved by the matrix inversion method.

In order to find the best pH value the function `y = neut(pH)` is defined and the optimized with fzero. Different iterative methods may be employed, each one corresponding to an Octave function. For instance, we started with fminbnd, in order to minimize the magnitude of the *neutrality(pH)* with respect to *pH* within the relevant finite domain. At the end, we preferred to employ fzero, since it searches for the zero crossing of *neutrality*, which looks more appropriate since neutrality separates between between negative and positive charge concentration.

```
%  HF <==> H+  +  F-
%  Ka1 = [H+]*[F-]/[HF]
clear all;clc;
global Atot = 0.18;  % [HF] initial concentration, before dissociation
global Na = 0.0;     % NaOH initial concentration, added to reaction
global Ka1 = 3.5e-4; % dissociation constant of fluoridric acid
global Kw = 1e-14;   % ionic water product
global H X
function y = neut(pH)
    global Atot Ka1 Kw Na X H
    H = 10^(-pH);
    % here we solve the system of 2 equations, 2 unknowns (HF and F)
    A = [-Ka1 H ;     %  -Ka1*HF + H*F = 0
          1   1];     %   1*HF + 1*F   = Atot
    b =[0; Atot];
    X = A\b;         %   X(1)=HF   X(2)=F-
    y = H + Na - Kw/H - X(2);    % electrical neutrality
endfunction

[pH,fval,info] = fzero(@neut,[0,14])
disp (['initial concentration of HF = ',num2str(Atot)]);
disp ([' pH    = ',num2str(pH,6)]);
disp ([' [H+]  = ',num2str(H,6)]);
disp ([' [HF]  = ',num2str(X(1),6)]);
disp ([' [F-]  = ',num2str(X(2),6)]);
disp('---------------------------------');
```

The result printout looks as follow.

```
pH = 2.1099
fval = -5.6379e-17
info = 1
initial concentration of HF = 0.18
 pH    = 2.1099
 [H+]  = 0.00776418
 [HF]  = 0.172236
 [F-]  = 0.00776418
```

The main script searches for the species concentrations which satisfy neutrality, by varying pH within its bounds with the help of the Octave function fzero. To this end, fzero calls the function neutrality, which computes the current neutrality which is assigned to the output.

The function neutrality works in the monoprotic, biprotic and triprotic cases, by building the generic linear system of equations driven by the total concentration Atot and by the proton concentration H. The parameters are the dissociation constants collected in the vector Ka. In the case of a monoprotic weak acid, the most simple case treated in the book ,with the function neutrality, the linear system holds:

$$\begin{bmatrix} A_{tot} \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ K_{a1} & -H & 0 & 0 \\ 0 & K_{a2} & -H & 0 \\ 0 & 0 & K_{a3} & -H \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \ldots \quad . \tag{1}$$

The neutrality is computed as

$$\text{neutrality} = \sum_{k=1}^{\text{dims}} x_k q_k \tag{2}$$

where the pair $\{x_k, q_k\}$ denotes the concentration and the signed electric charge of the species with subscript k .

9.3 The biprotic acid, oxalic acid as an example

One more equation is added, so as now a set of 3 linear equation has to be solved.

Moreover in the example below addition of a strong base like NaOH can be simulated.

```
-Ka1*H2C2O4 + H*HC2O4 + 0*C2O4 = 0       first dissociation
 0*H2C2O4 - Ka2*HC2O4 + H*C2O4 = 0       second dissociation
 1*H2C2O4 + 1*HC2O4 +  1*C2O4  = Atot    mass conservation of oxalate
```

```
%  H2C2O4 <==> H+  +  HC2O4-      Ka1 = [H+]*[HC2O4-]/[H2C2O4]
%  HC2O4- <==> H+  +  C2O4--      Ka2 = [H+]*[HPO4--]/[H2PO4-]
clear all;clc;
global Atot = 0.02;   % total concentration of H2C2O4
global Na = 0.0;    % total concentration of NaOH (i.e. total amount of NaOH if added)
global Ka1 = 5.4e-2;  % 1st dissociation constant
global Ka2 = 5.3e-5;  % 2nd dissociation constant
global Kw = 1e-14;    % ionic water product
global H X
function y = neut(pH)
    global Atot Ka1 Ka2 Kw Na X H
    H = 10^(-pH);
```

```
    % here we solve the system of 3 equations, 3 unknowns
    A = [-Ka1 H 0 ;     %  -Ka1*H2C2O4 + H*HC2O4 + 0*C2O4 = 0
          0 -Ka2 H;     %   0*H2C2O4 - Ka2*HC2O4 + H*C2O4 = 0
          1  1   1];    %   1*H2C2O4 + 1*HC2O4 +  1*C2O4  = Atot
    b =[0; 0; Atot];
    X = A\b;        %   X(1)=H2C2O4   X(2)=HC2O4-   X(3)=HC2O4
    y = H + Na - Kw/H - X(2) - X(3)*2;    % neutrality
endfunction

[pH,fval,info] = fzero(@neut,[0,14])
disp(['initial concentration of H2C2O4 = ',num2str(Atot)]);
disp(['pH        = ',num2str(pH,6)]);
disp(['[H+]      = ',num2str(H,6)]);
disp(['[OH-]     = ',num2str(Kw/H,6)]);
disp(['[H2C2O4]  = ',num2str(X(1),6)]);
disp(['[HC2O4-]  = ',num2str(X(2),6)]);
disp(['[C2O4--]  = ',num2str(X(3),6)]);
disp('---------------------------------')
```

Printout

```
pH = 1.8073
fval = -5.4671e-17
info = 1
initial concentration of H2C2O4 = 0.02
pH        = 1.80729
[H+]      = 0.015585
[OH-]     = 6.41642e-13
[H2C2O4]  = 0.00446763
[HC2O4-]  = 0.0154797
[C2O4--]  = 5.2642e-05
```

9.4 The triprotic acid, phosphoric acid as an example

In the case of a triprotic acid, the most complex case treated in the book with the function neutrality, the linear system holds:

```
-Ka1*H3PO4 + H*H2PO4 + 0*HPO4 + 0*PO4 = 0
 0*H3PO4 - Ka2*H2PO4 + H*HPO4 + 0*PO4 = 0
 0*H3PO4 + 0*H2PO4 - Ka3*HPO4 + H*PO4 = 0
 1*H3PO4 + 1*H2PO4 +  1*HPO4 +  1*PO4 = Ptot
```

The neutrality is computed as

```
y = H + Na - Kw/H - X(2) - X(3)*2 - X(4)*3;
```

```
%  H3PO4 <==> H+  +  H2PO4-      Ka1 = [H+]*[H2PO4-]/[H3PO4]
%  H2PO4- <==> H+  +  HPO4--     Ka2 = [H+]*[HPO4--]/[H2PO4-]
%  HPO4-- <==> H+  +  PO4---     Ka3 = [H+]*[PO4---]/[HPO4--]
clear all;clc;
Atot = 0.4;    % total concentration of H3PO4  (i.e. total amount of H3PO4 added)
Natot = 0.4;   % total concentration of Na3PO4 (i.e. total amount of Na3PO4 if added)
global Ka1 = 7.5e-3;  % 1st dissociation constant
global Ka2 = 6.2e-8;  % 2nd dissociation constant
global Ka3 = 4.4e-13; % 3rd dissociation constant
global Kw = 1e-14;    % ionic water product
global Na = Natot*3;
global Ptot = Natot + Atot;
global H X
function y = neut(pH)
    global Ptot Na Ka1 Ka2 Ka3 Kw H X
    H = 10^(-pH);
    OH = Kw/H;
    % here we solve the system of 4 equations, 4 unknowns
    A = [-Ka1 H 0 0 ;     %  -Ka1*H3PO4 + H*H2PO4 + 0*HPO4 + 0*PO4 = 0
          0 -Ka2 H 0;     %   0*H3PO4 - Ka2*H2PO4 + H*HPO4 + 0*PO4 = 0
          0 0 -Ka3 H;     %   0*H3PO4 + 0*H2PO4 - Ka3*HPO4 + H*PO4 = 0
          1  1  1  1];    %   1*H3PO4 + 1*H2PO4 +  1*HPO4 +  1*PO4 = Ptot
    b =[0; 0; 0; Ptot];
    X = A\b;        %   X(1)=H3PO4   X(2)=H2PO4-   X(3)=HPO4--   X(4)=PO4---
    y = H + Na - Kw/H - X(2) - X(3)*2 - X(4)*3;
    endfunction
[pH,fval,info] = fzero(@neut,[0,14])
disp(['H3PO4 initial concentration   = ',num2str(Atot,5)]);
disp(['Na3PO4 initial concentration  = ',num2str(Natot,5)]);
disp('---------------------------------');
disp (['pH        = ',num2str(pH,6)]);
disp (['[H+]      = ',num2str(H,6)]);
disp (['[OH-]     = ',num2str(Kw/H,6)]);
disp (['[H3PO4]   = ',num2str(X(1),6)]);
disp (['[H2PO4-]  = ',num2str(X(2),6)]);
disp (['[HPO4--]  = ',num2str(X(3),6)]);
disp (['[PO4---]  = ',num2str(X(4),6)]);
disp('---------------------------------')
```

Printout

```
pH = 7.2076
fval = -1.8668e-15
info = 1
H3PO4 initial concentration   = 0.4
Na3PO4 initial concentration  = 0.4
---------------------------------
pH         = 7.20761
[H+]       = 6.19998e-08
[OH-]      = 1.61291e-07
[H3PO4]    = 3.30663e-06
[H2PO4-]   = 0.399996
[HPO4--]   = 0.399998
[PO4---]   = 2.8387e-06
---------------------------------
```

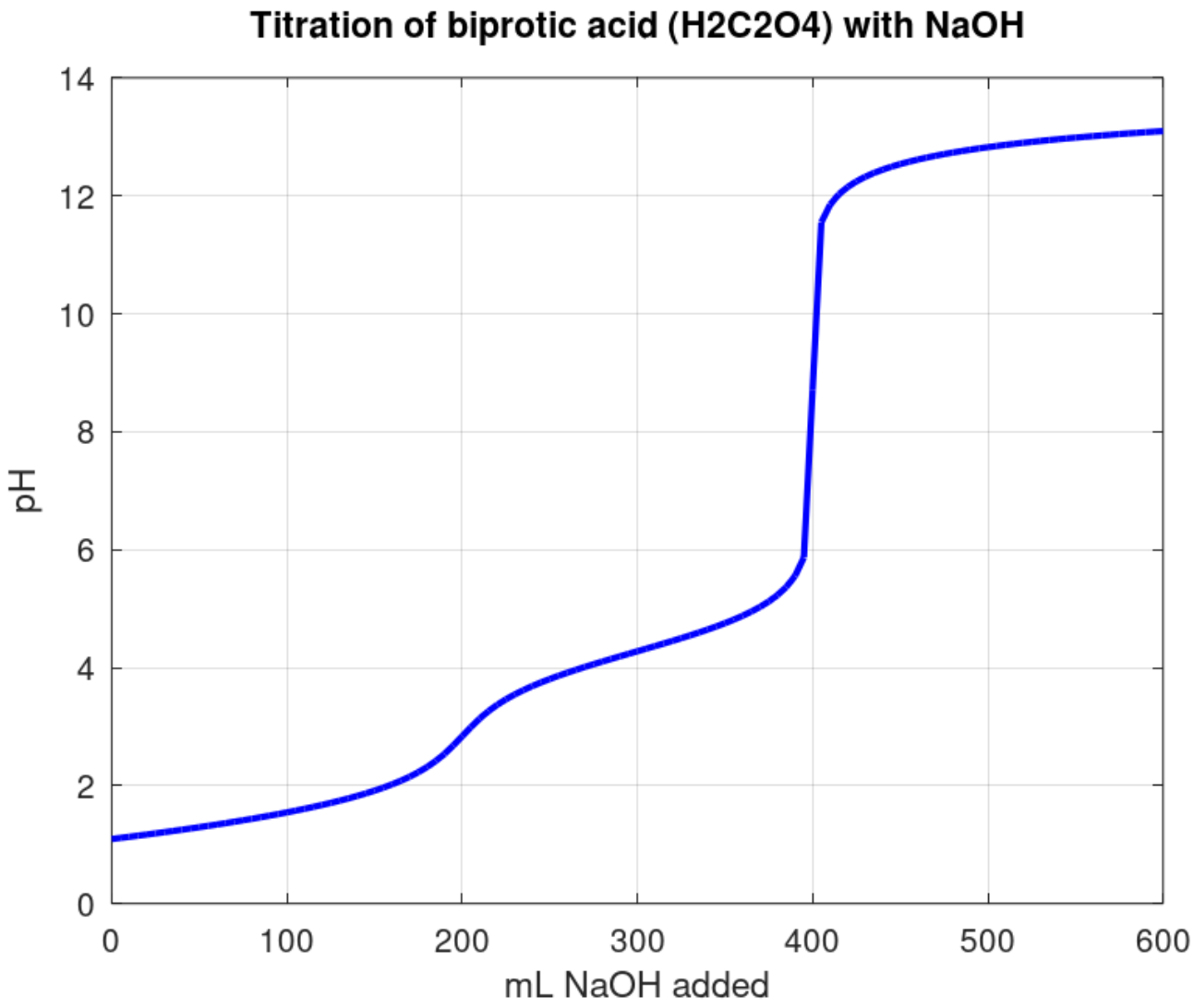
9.5 Titration of a weak biprotic acid with NaOH

A titration curve is the plot of the pH of the analyte solution (a biprotic acid) versus the volume of the titrant added (sodium hydroxide) as the titration progresses.

Titrations are often recorded on graphs called titration curves, which generally contain the volume of the titrant as the independent variable and the pH of the solution as the dependent variable. The equivalence point on the graph is where all of the starting solution has been neutralized by the titrant.

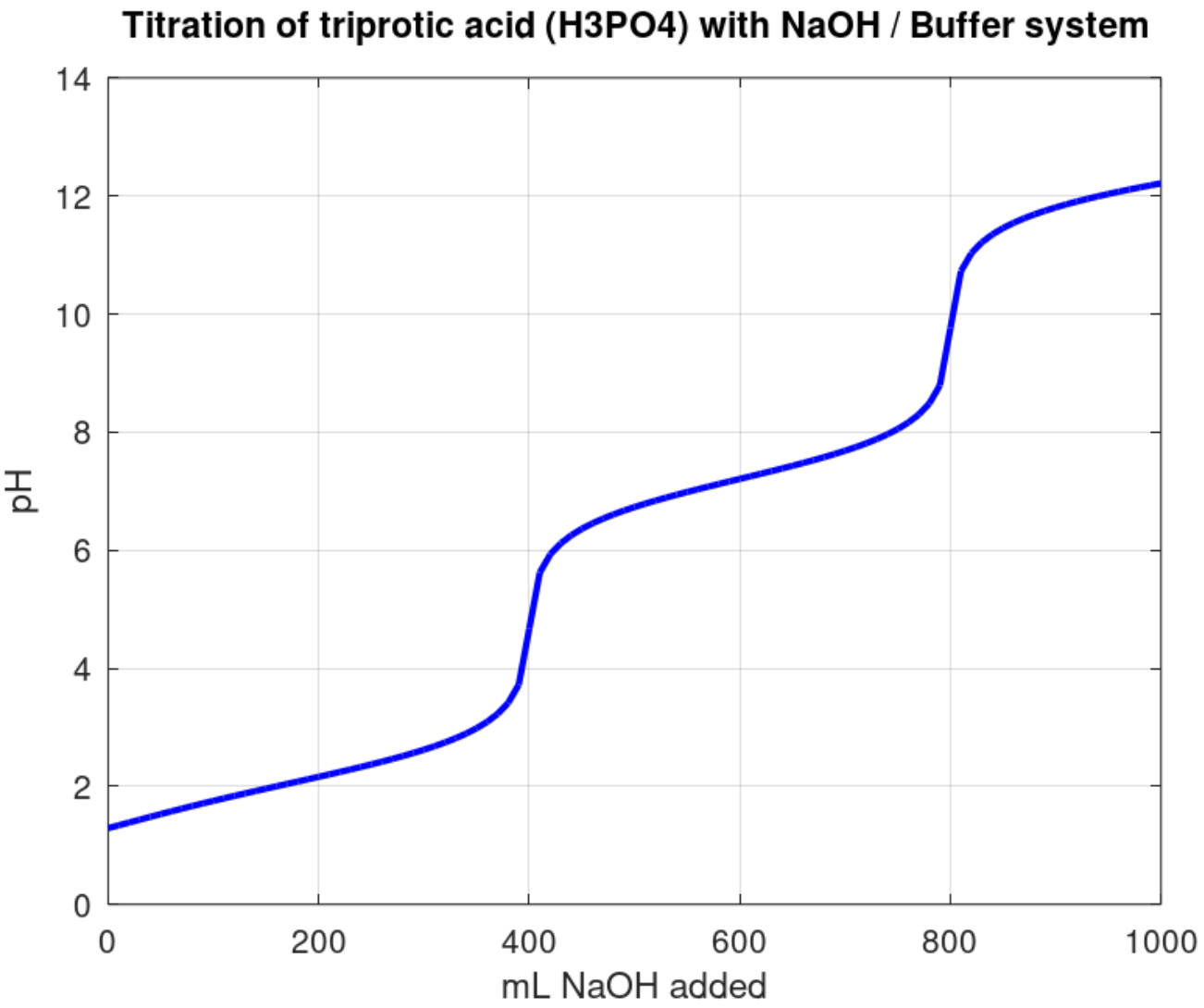
```
% TITRATION OF A WEAK BIPROTIC ACID (oxalic acid) WITH STRONG BASE (NaOH)
% First and secon dissociation of oxalic acid
% HCOO-COOH <==> HCOO-COO(-)  + H(+)
%  Ka1 = [H+]*[H2C2O4]/[HC2O4-] = 5.4e-2
% HCOO-COO(-) <==> (-)COO-COO(-)  + H(+)
%  Ka2 = [H+]*[HC2O4-]/[C2O4--] = 5.3e-5

clc;clear all;format compact
Atot = 0.2;     % total concentration of H2C2O4 before dissociation
Ka1 = 5.4e-2;   % 1st diss H2C2O4 @25Â°C
Ka2 = 5.3e-5;   % 2nd diss HC2O4-
Kw = 1e-14;     % Ionic water product @ 25Â°C
CNaOH = 1.00;   % molar concentration of added sodium hydroxide
T1 = ['[H2C2O4] initial value  = ',num2str(Atot,6),'  mol/L'];
disp(T1)
disp('----------------------------------')
k = 0;
for mL_NaOH = 0:5:600 % mLitre NaOH 1M added to 1 litre solution of H2C2O4
A1 = Atot*1000/(1000 + mL_NaOH);
Na = mL_NaOH*CNaOH/(1000 + mL_NaOH);
pH1 = 0;
pH2 = 14;
pHstep = 1;
for j = 1:8
  for pH = pH1:pHstep:pH2
    H = 10^(-pH);
    % here we solve (3 equations, 3 unknowns)
    % Ka1 = [H+]*[H2C2O4-]/[H2C2O4]
    % Ka2 = [H+]*[C2O4--]/[HC2O4-]
    % Atot = [C2O4--] + [HC2O4-] + [H2C2O4]
    C2O4 = Ka1*Ka2*A1/(H*H + Ka1*H + Ka1*Ka2);
    HC2O4 = H*C2O4/Ka2;
    H2C2O4 = H*HC2O4/Ka1;
    OH = Kw/H;
    Neut = H + Na - OH - HC2O4 - 2*C2O4;
    if Neut<0;break;end
  endfor
  pH2 = pH;
  pH1 = pH2 - pHstep;
  pHstep = pHstep/10;
endfor
T1 = ['NaOH ml added  = ',num2str(mL_NaOH,4),'    pH =  ',num2str(pH,4)];disp(T1);
k = k + 1;
a(k) = mL_NaOH;b(k) = pH;
endfor
plot(a,b,'b','LineWidth',2) ;grid on;
xlabel('mL NaOH added');
```

```
ylabel('pH')
title('Titration of biprotic acid (H2C2O4) with NaOH')
```



9.6 Titration of a triprotic acid with NaOH

```
%  H3PO4 <==> H+  +  H2PO4-
%  Ka1 = [H+]*[H2PO4-]/[H3PO4]
%  H2PO4- <==> H+  +  HPO4--
%  Ka2 = [H+]*[HPO4--]/[H2PO4-]
%  HPO4-- <==> H+  +  PO4---
%  Ka3 = [H+]*[PO4---]/[HPO4--]
clc;clear all;format compact;
Atot = 0.4;    % total concentration of H3PO4  (i.e. total amount of H3PO4 added)
CNaOH = 1.00;  % molar concentration of added sodium hydroxide
Ka1 = 7.5e-3;  % 1st dissociation constant
Ka2 = 6.2e-8;  % 2nd dissociation constant
Ka3 = 4.4e-13; % 3rd dissociation constant
Kw = 1e-14;    % ionic product of water
T1 = ['H3PO4 initial concentration   = ',num2str(Atot,5)];disp(T1);
disp('---------------------------------');
k = 0;
for mL_NaOH = 0:10:1000 % mLitre NaOH 1M added to 1 litre solution of H3PO4
A1 = Atot*1000/(1000 + mL_NaOH);
Na = mL_NaOH*CNaOH/(1000 + mL_NaOH);
pH1 = 0;
pH2 = 14;
pHstep = 1;
for j = (1:8)     % each step of the loop pHstep decreases 10x
  for pH = (pH1:pHstep:pH2)           % loop of pH
    H = 10^(-pH);
    % here we solve the system of 4 equations, 4 unknowns (speed up needed!)
    %  Ka1 = H*H2PO4/H3PO4
    %  Ka2 = H*HPO4/H2PO4
    %  Ka3 = H*PO4/HPO4
    %  A1 = H3PO4 + H2PO4 + HPO4 + PO4
    PO4 = Ka1*Ka2*Ka3*A1/(H*H*H + Ka1*H*H + Ka1*Ka2*H + Ka1*Ka2*Ka3);
    HPO4 = H*PO4/Ka3;
    H2PO4 = H*HPO4/Ka2;
    H3PO4 = H*H2PO4/Ka1;
    Neut = H + Na - Kw/H - H2PO4 - HPO4*2 - PO4*3;
    if Neut<0;break;end
  end
  pH2 = pH;
  pH1 = pH2 - pHstep;
  pHstep = pHstep/10;
end
T1 = ['NaOH ml added  = ',num2str(mL_NaOH,4),'    pH =  ',num2str(pH,4)];disp(T1);
k = k + 1;
a(k) = mL_NaOH;b(k) = pH;
end
disp('---------------------------------')
plot(a,b,'b','LineWidth',2) ;grid on;
xlabel('mL NaOH added');
ylabel('pH')
title('Titration of triprotic acid (H3PO4) with NaOH / Buffer system')
```


Titration of triprotic acid (H3PO4) with NaOH / Buffer system
pH
mL NaOH added
0
2
4
6
8
10
12
14
0
200
400
600
800
1000

9.7 Ca and Mg reactions in carbonatic waters

This is a most complex case, still to be solved with the methos of minimizing the electroneutrality of the solution. Formation of carbonatic equilibria is considered , as well as the CaCO3 and Mg(OH) precipitation. All this equilibria are solved simultaneously.

```
clear all;clc;
global Ctot Catot Na Cl Mgtot Kw Ka1 Ka2 Ksp Ksp2 H OH H2CO3 HCO3 CO3 Ca CaCO3 Mg
MgOH2
Na2CO3 = 0.2;   % added moles. Soluble salt completely dissociated in Na+ and CO3--
ions
NaHCO3 = 0.1;   % added moles. Soluble salt completely dissociated in Na+ and HCO3-
ions
CaCO3  = 0.01;  % added moles. Sparligly soluble salt. Solubility product Ksp needed
NaOH    = 0.20;  % added moles. Strog electrolyte, completely dissociated in Na+  and
OH- ions
HCl     = 0.5;   % added moles. Strog electrolyte, completely dissociated in H+  and
Cl- ions
CaCl2   = 0.25;   % added moles. Soluble salt completely dissociated in Ca++ and Cl-
ions
MgCl2  = 0.4;  % added moles. Soluble salt completely dissociated in Mg++ and Cl- ions
H2CO3  = 0.0;  % added moles of carbonic acid, a weak acid partially dissociated
disp ('---------------------------------');
disp (' added reactants in mol/L');
disp([' Na2CO3 = ',num2str(Na2CO3,4)]);
disp([' NaHCO3 = ',num2str(NaHCO3,4)]);
disp([' CaCO3  = ',num2str(CaCO3,4)]);
disp([' NaOH   = ',num2str(NaOH,4)]);
disp([' HCl    = ',num2str(HCl,4)]);
disp([' CaCl2  = ',num2str(CaCl2,4)]);
disp([' MgCl2  = ',num2str(MgCl2,4)]);
disp([' H2CO3  = ',num2str(H2CO3,4)]);
disp ('---------------------------------');

% Now we defina the global variables :
Ctot = Na2CO3 + NaHCO3 + CaCO3 + H2CO3;   %total concentration of all forms of
carbonate (CO3--) ion
Catot = CaCO3 + CaCl2;                  %total concentration of all forms of carbonate
(Ca++) ion
Na = Na2CO3*2 + NaHCO3 + NaOH;           %total concentration of sodium (Na+) ions
Cl = HCl + CaCl2*2;                      %total concentration of chloride (Cl-) ions
Mgtot = MgCl2;                           %total concentration of magnesium (Mg++) ions
Kw = 1e-14;     % ionic water product,  kw = [H+]*[OH-]
Ka1 = 4.45e-7;  % 1st dissociation constant for H2CO3,  Ka1 =[H+]*[HCO3-]/[H2CO3]
Ka2 = 4.69e-11; % 2nd dissociation constant for H2CO3,  Ka2 =[H+]*[CO3-]/[HCO3-]
Ksp = 3.8e-9;   % Solubility product of calcite at 25°C Ksp = [Ca++]*[CO3--]
Ksp2 = 8.9e-12; % Solubility ptoduct of brucite at 25°C Ksp2 = [Mg++]*[OH-]^2

function y = neut(pH)
    global Ctot Catot Na Cl Mgtot Kw Ka1 Ka2 Ksp Ksp2 H OH H2CO3 HCO3 CO3 Ca CaCO3 Mg
MgOH2
    H = 10^(-pH);OH = Kw/H;
    % Control of precipitation of Mg(OH)2
    Mg = Ksp2/OH/OH;
    if Mg<Mgtot
      MgOH2 = Mgtot - Mg;
      else
```

```
      Mg = Mgtot;MgOH2 = 0;
    end
    % Csol is the total carbon in solution, in the different forms H2CO3,HCO3-,CO3--
    % CaCO3 is the amount of solid precipitate of CaCO3-calcite per liter
    a1 = Ka1*Ka2;
    b1 = Ka1*Ka2*(Catot - Ctot);
    c1 = -Ksp*(H*H + Ka1*H + Ka1*Ka2);
    % a1*x^2 + b1*x + c1 = 0;      % second order polynomial
    Csol = (-b1 + sqrt(b1^2 - 4*a1*c1))/(2*a1);
     % If [Ca]*[CO3] product does not reach the Ksp (solubility product of calcite)
then no CaCO3 is formed.
     % In this case Csol becomes artificially greater than Ctot so as it must be
resized to Ctot.
    if Csol>Ctot ; Csol = Ctot;end
    CO3 = Ka1*Ka2*Csol/(H*H + Ka1*H + Ka1*Ka2);
    HCO3 = H*CO3/Ka2;
    H2CO3 = H*HCO3/Ka1;
    CaCO3 = Ctot - Csol;
    Ca = Catot - CaCO3;
    y = H + Na + Ca*2 + Mg*2 - OH - Cl - HCO3 - CO3*2;
endfunction
[pH,fval,info] = fzero(@neut,[0,14])

disp (' final concentrations in mol/Liter');
disp([' pH        = ',num2str(pH,4)]);
disp([' [H+]      = ',num2str(H,4)]);
disp([' [OH-]     = ',num2str(OH,4)]);
disp([' [H2CO3]   = ',num2str(H2CO3,4)]);
disp([' [HCO3-]   = ',num2str(HCO3,4)]);
disp([' [CO3--]   = ',num2str(CO3,4)]);
disp([' [Ca++]    = ',num2str(Ca,4)]);
disp([' [CaCO3]   = ',num2str(CaCO3,4)]);
disp([' [Mg++]    = ',num2str(Mg,4)]);
disp([' [Mg(OH)2] = ',num2str(MgOH2,4)]);
disp ('-------------some controls----left and right must agree !-----------------');
disp([Ksp,Ca*CO3]);disp([Ksp2,Mg*OH*OH]);
disp([MgCl2,Mg+MgOH2]);
disp([Catot,Ca+CaCO3]);
```

Printout

```
---------------------------------
 added reactants in mol/L
 Na2CO3 = 0.2
 NaHCO3 = 0.1
 CaCO3  = 0.01
 NaOH   = 0.2
 HCl    = 0.5
 CaCl2  = 0.25
 MgCl2  = 0.4
 H2CO3  = 0
---------------------------------
pH = 8.8523e+00
fval = -8.1437e-15
info = 1.0000e+00
 final concentrations in mol/Liter
 pH        = 8.852
```

```
 [H+]      = 1.405e-09
 [OH-]     = 7.117e-06
 [H2CO3]   = 0.0001523
 [HCO3-]   = 0.04824
 [CO3--]   = 0.00161
 [Ca++]    = 2.36e-06
 [CaCO3]   = 0.26
 [Mg++]    = 0.1757
 [Mg(OH)2] = 0.2243
-------------some controls----left and right must agree !-----------------
   3.8000e-09   3.8000e-09
   8.9000e-12   8.9000e-12
   4.0000e-01   4.0000e-01
   2.6000e-01   2.6000e-01
```

9.8 Ionic product of water, pH, conductivity for pure water from 0°C to 80°C

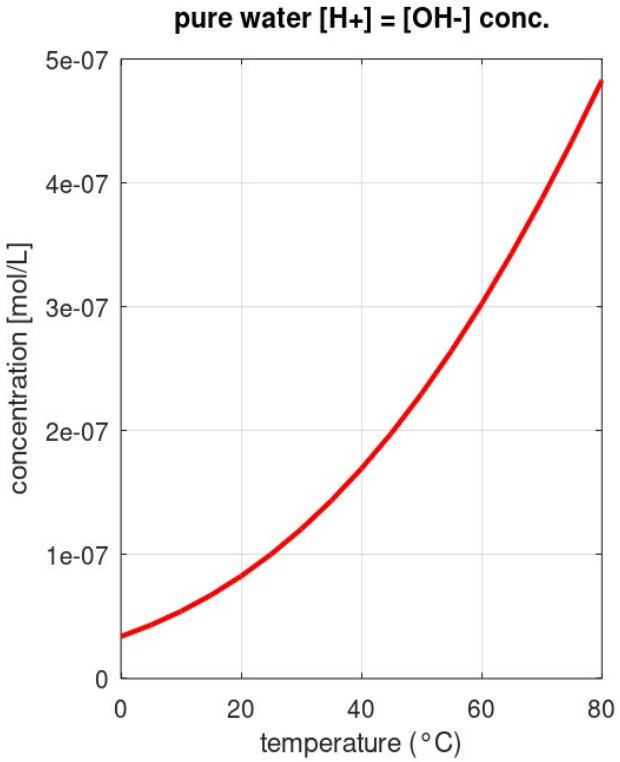
Conductivity (or specific conductance) of an electrolyte solution is a measure of its ability to conduct electricity. The SI unit of conductivity is Siemens per meter (S/m).

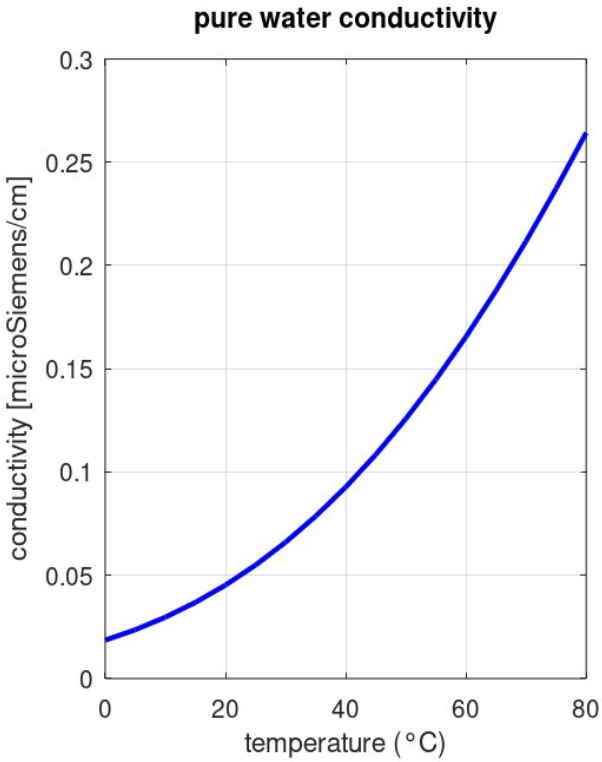
Conductivity measurements are used routinely in many industrial and environmental applications as a fast, inexpensive and reliable way of measuring the ionic content in a solution.[1] For example, the measurement of product conductivity is a typical way to monitor and continuously trend the performance of water purification systems.

The electrolytic conductivity of ultra-high purity water increases as a function of temperature (T) due to the higher dissociation of H2O in H+ and OH− with T.

In many cases, conductivity is linked directly to the total dissolved solids (TDS). High quality deionized water has a conductivity of about 0.05 µS/cm at 25 °C, typical drinking water is in the range of 200–800 µS/cm, while sea water is about 50 mS/cm (or 50,000 µS/cm).

```
% Ionic produc of water,pH, and conductivity at different temperatures.
% H2O(liq) = H+(aq) + OH-(aq)  :  Kw = f(T)  :  pH = -1*log(H)/log(10)
% Thermodynamic data used in the simulation include:
% item          H°        S°      Cp        Lambda0
%   H+          0         0        0         349.8
%  OH-       -229990   -10.75     -148.5     197.6
% H2O (liq)  -285830    69.91     75.29        0

clear;clc;format compact;format shortE;
R = 8.314;j = 0;
disp('   temperature   Kw            [H+]          [OH-]         pH            specific
conductivity uS/cm');
for T1 = (0:5:80)
    j = j + 1;t(j) = T1;
    T = 273.15 + T1;
    DeltaH = (-229990 - (T-298)*148.5) - (-285830 + (T-298)*75.29);
    DeltaS = (-10.75 - (log(T)-log(298))*148.5) - (69.91 + (log(T)-log(298))*75.29);
    DeltaG = DeltaH - T*DeltaS;
    Kw = exp(-DeltaG/(R*T));
    H(j) = sqrt(Kw);OH(j) = H(j);
    pH = -log10(H(j));
    conductivity(j) = (349.8*H(j) + 197.6*OH(j))*1000;
    disp ([t(j),Kw,H(j),OH(j),pH,conductivity(j)]);
end
figure (1,'position',[200 100 800 400]);clf;
subplot(1,2,1)
plot(t,H,'r','LineWidth',2);grid on;
title('pure water [H+] = [OH-] conc.')
xlabel('temperature (°C)')
ylabel('concentration [mol/L]')
subplot(1,2,2)
plot(t,conductivity,'b','LineWidth',2);grid on;
xlabel('temperature (°C)')
ylabel('conductivity [microSiemens/cm]')
title('pure water conductivity')
```


pure water [H+] = [OH-] conc.
concentration [mol/L]
5e-07
4e-07
3e-07
2e-07
1e-07
0
0
20
40
60
80
temperature (°C)


pure water conductivity
conductivity [microSiemens/cm]
0.3
0.25
0.2
0.15
0.1
0.05
0
0
20
40
60
80
temperature (°C)

Chapter 10. Colligative properties of solutions: the case of NaCl dissolved in water.

Here below the octave script

```
% Let us examine colligative properties of NaCl solutions in water, with increasing
concentration.clear;clc;format short;format compact;
R = 8.314;NaCl = 22.9898 + 35.453; % atomic weight of sodium chloride.
Tc = 4;          % Temperature/°C
T = Tc + 273.15; % T temperature/K

% coefficients for density interpolation of NaCl solutions (require conc. and temp.)
D(1) = 919.0202567;D(2) = 8.661163416;D(3) = 0.854264859;D(4) = 0.027175948;D(5) = -
0.00199299;D(6) = -0.00585389;
% end of coefficients


T0 =['|Conc. w/w|Density g/cm3|molarity|molality|freeze pt.|ebullioscopic|osmotic pr.|
rel vap. press.|'];
T2 =['+---------+-------------+--------+--------+----------+-------------+-----------
+---------------+'];
disp(T2);disp(T0);disp(T2);


for cw = (1:20)                          % weight of NaCl/weight of solution * 100
    density = D(1) + D(2)*cw + D(3)*T + D(4)*cw^2 + D(5)*T^2 + D(6)*T*cw; % densiti
g/dm^3
    molality = cw/NaCl/((100-cw)/1000); % molality = mol/(kg-solvent)
    molarity = cw/NaCl*density/100;     % molarity = mol/(L-solution)
    i_vH = 2;                           % van't Hoff coefficient, i
    freeze = 1.853*i_vH*molality;       % freezing point/°C
    ebull = 0.512*i_vH*molality;        % boiling point increase
    osmotic = i_vH*molarity*0.0821*T;   % osmotic pressure/atm.
    vap_press = (100-cw)/18 / ((100-cw)/18 + i_vH*cw/NaCl);   % vapor pressure
relative decrease

    % here output strig is rearranged
    T1 =['|  ',num2str(cw,3)];if cw<10; T1 = [T1,' ']; endif
    T1 = [T1,'     |   ',num2str(density,'%.1f'),'    |  ',num2str(molarity,'%.3f'),'
|  ',num2str(molality,'%.3f'),...
    ' |  -',num2str(freeze,'%.2f')];if freeze<10; T1 = [T1,' ']; endif
    T1 = [T1,'  |    ',num2str(ebull,'%.3f'),'    |   ',num2str(osmotic,'%.1f')];
    if osmotic<10;T1=[T1,' ];endif
    if osmotic<100;T1=[T1,' ];endif
    T1 = [T1,'   |     ',num2str(vap_press,'%.3f'),'     |'];


    % display of the output string 'T1'
    disp(T1);
endfor
disp(T2);
```

Here as follows the output text:

Conc. w/w	Density g/cm3	molarity	molality	freeze pt.	ebullioscopic	osmotic pr.	rel vap. press.
1	1009.8	0.173	0.173	-0.64	0.177	7.9	0.994
2	1016.9	0.348	0.349	-1.29	0.358	15.8	0.988
3	1024.1	0.526	0.529	-1.96	0.542	23.9	0.981
4	1031.3	0.706	0.713	-2.64	0.730	32.1	0.975
5	1038.6	0.889	0.901	-3.34	0.922	40.4	0.969
6	1045.9	1.074	1.092	-4.05	1.118	48.9	0.962
7	1053.3	1.262	1.288	-4.77	1.319	57.4	0.956
8	1060.7	1.452	1.488	-5.51	1.524	66.1	0.949
9	1068.2	1.645	1.692	-6.27	1.733	74.9	0.943
10	1075.8	1.841	1.901	-7.05	1.947	83.8	0.936
11	1083.4	2.039	2.115	-7.84	2.166	92.8	0.929
12	1091.1	2.240	2.333	-8.65	2.389	102.0	0.923
13	1098.8	2.444	2.557	-9.48	2.618	111.2	0.916
14	1106.6	2.651	2.785	-10.32	2.852	120.6	0.909
15	1114.4	2.860	3.020	-11.19	3.092	130.2	0.902
16	1122.3	3.072	3.259	-12.08	3.337	139.8	0.895
17	1130.2	3.288	3.505	-12.99	3.589	149.6	0.888
18	1138.2	3.506	3.756	-13.92	3.846	159.5	0.881
19	1146.2	3.726	4.014	-14.87	4.110	169.6	0.874
20	1154.3	3.950	4.278	-15.85	4.380	179.8	0.867

Chapter 11. Seawater and CO_2 equilibria

11.1 Introduction

Seawater isn't simply a reservoir of different dissolved salts, like sodium chloride, magnesium sulphate and so on; it has the capability to react in different ways, one being with carbon dioxide in the atmosphere, modifying its concentration and buffering its anthropogenic increase.

Oceans cover about 71% of the earth's surface, so even slight variations of critical parameters, like pH, salinity and so on have a huge effect on global climate**,** and therefore on our way of life.

The reactions involved, however, cannot be mathematically treated with the usual chemistry textbook solutions. Seawater has a high salt content (or ionic strength), and the potency of its ionic charge density totally alters the equilibrium constants**,** rendering such solutions (which are based on pure water) of little use. The calculation parameters are further affected by high pressures to be found in the dark abysses.

By employing Octave programming language (very similar to Matlab)**,** the appropriate algorithms for chemical equilibria in seawater are proposed in a plain and simple scripts, ready for use, or even modification, by the reader. There is no need for advanced math or programming expertise. An introduction to basic chemistry calculations in solutions can be found elsewhere in this web-site, user friendly for experts and amateur enthusiasts alike.

Breathe in, breathe out. Let as deal now with the main gas exchange in earth oceans; like a giant lung, oceans absorb vast amounts of carbon dioxide (CO2) from the atmosphere, and release it once again as cold water currents reach warmer areas of the globe. Indeed, CO2 solubility varies with temperature, together with other factors such as salinity and pressure.
Chemically speaking, why does seawater so readily absorb carbon dioxide, thereby buffering the anthropogenic emission of this gas?
The oceans cover about 71% of the earth's surface and gaseous exchange occurs through the ocean's surface. But the answer to this question lies deeper, in what is a widely underestimated fact: the pH (acidity level) of seawater is substantially alkaline, ranging from 8.0 to 8.7. This means that the balance of positive and negative ions is reached through a higher concentration of hydroxide ions (OH^-) compared to hydrogen ions (H^+).

Having a pH value greater than 7 enables seawater to react with and dissolve huge amounts of CO2, absorbing atmospheric excess and thus affecting its concentration. However, there is a reason behind the alkalinity of seawater, its current chemical composition. While different salts are present in seawater, the primary one is sodium chloride. As with any salt, when it dissolves in water, positive charges (cations) and negative charges (anions) are generated.

11.2 Chemical Composition and Imbalance

Let's explore the mean composition of seawater in greater detail: summing up all the positive charges (Na^+, K^+, Mg^{2+}, Ca^{2+}, Sr^{2+}) one obtains 605.85 mmol/Kg of solution. Carrying out the same operation for negative charges (Cl^-, Br^-, F^-, $SO4^{2-}$) the result is slightly less: 603.25; 2.50 millimoles are clearly missing! As with all ionic solutions, seawater must obey the law of electro neutrality, so evidently some negative charges (anions) have been ruled out: they are indeed $HCO3^-$, to a minor extent OH^- and finally, to a far lesser extent, $CO3^{2-}$. The last three ions all react with atmospheric CO2, and are therefore designated as reactive. On the contrary, the former cations and anions are classified as spectator ions (see table 1). Reactive ions have an active role in chemical equilibria, as shown in the same table.

The presence of OH^- (hydroxide ions) is the reason for a pH>7. Their concentration (due to the logarithmic nature of pH scale) is at pH = 8.0 equal to 0.001 mmol/L (in pure water). Under the same conditions, the H^+ ion concentration is 100 times less. OH^- ions alone are insufficient to fill the gap: other negative ions are required; these are mainly $HCO3^-$ ions and also some $CO3^{2-}$.

This has enormous repercussions on the equilibrium of CO2 between the atmosphere and oceans. Compared to the atmosphere, which contains around 850 Gt (gigatons) of carbon (in the form of CO2), the oceans hold 38,000 Gt of carbon. That's nearly 45 times more.
So when we talk about CO2 ppm in the atmosphere that is only the "top of the iceberg"! CO2 dissolves in seawater like O2 and N2. However, being a reactive gas, there is an almost immediate reaction with the water itself (N2 and O2 do not) yielding $HCO3^-$ and $CO3^{2-}$. After completion of these reactions, yet a third slowly takes place (one which is nearly always disregarded): the formation of solid calcium carbonate, CaCO3.

Spectator Ions					
Cation	$mol \cdot kg^{-1}$	$g \cdot kg^{-1}$	**Anion**	$mol \cdot kg^{-1}$	$g \cdot kg^{-1}$
Na^+	0.46906	10.7836	Cl^-	0.54586	19.3524
Mg^{2+}	0.05282	1.2837	SO_4^{2-}	0.02824	2.7123
Ca^{2+}	0.01028	0.4121	Br^-	0.00084	0.0673
K^+	0.01021	0.3991	F-	0.00007	0.0013
Sr^{2+}	0.00009	0.0079			
$\Sigma+$	0.60565	12.8864	$\Sigma-$	0.60325	22.1333
			Δcharge = 0.00240+		
			Σ $g \cdot kg^{-1}$ = 35.0197		
Reactive Ions and Molecules					
H_2CO_3	←→	HCO_3^-	←→	CO_3^{2-}	
$B(OH)_3$	←→	$B(OH)_4^-$			
$2 \cdot H_2O$	←→	H_3O^+	←→	OH^-	

Table 11.1 Seawater composition: spectator ions and reactive species

In chemistry, this is known as precipitation. CaCO3 usually has a calcite structure; aragonite, the other polymorphic structure, is slightly more soluble. Seawater is oversaturated, both in terms of calcite and aragonite, due to its relatively high Ca^{++} ion concentration (10.28 mmol/Kg-solution). However, this reaction requires nucleation and the growth of crystal nuclei, and is usually sluggish (it may speed up in the cells of calcifying organisms like invertebrates). In other words, it is a heterogeneous reaction between a liquid phase and a solid one.
The destiny of this salt is to eventually sedimentate on the ocean floor (if very deep, it may fail to reach the bottom, dissociating again into ions due to the extremely high pressure, and recycle). In any case, CO2 removed from the atmosphere will eventually form limestone.

11.3 Methods and Techniques for Dealing with the Chemistry of Seawater

Every year there are hundreds of publications and articles on this topic: some fearing ocean 'acidification' (a lowering of pH values, remaining in the alkaline range) and the consequence on calcifying organisms, and some stressing a possible increase in the ocean's ability to uptake anthropogenic CO2. Indeed, several groups of scientists have employed computer-

aided modelling and complex models to simulate the chemical/physical behaviour of ocean water and predict the effects of man-made activities such as fossil burning.
These models cover a host of variables, and in the absence of deep insight into the structure of the complex codes used, one has no choice but to take the results at face value. Obviously, the effects of temperature, salinity and pressure on seawater are accounted for, but the codes are far from user-friendly, and even other scientists are unable to draw clear conclusions regarding the behaviour of seawater and related chemical equations.

In this context, simple Octave routines are clearly described for solving the various chemical equilibria in seawater, nothing concealed and everything accurately referenced. Anyone with a little chemical knowledge will be able to follow them. The routines and codes, can be downloaded and modified. The aim is to examine the chemical reactions that occur in seawater, using a simple and intuitive computer approach. Despite quite frequent discussion and examination in scientific papers and the press of the relationship between ocean chemistry and environmental issues (such as CO2 uptake, ocean acidification and carbonate sediment), the basic underlying chemistry is poorly understood.

On the other hand, with Octave codes just a few ten lines long, basic chemistry can offer a variety of simple and extremely interesting results for anybody curious about reactions in seawater. Well, let's not oversimplify! Seawater solution has a high ionic strength (high density of oppositely charged ions), a fact that hinders the direct usage of equilibrium constants taken from standard thermodynamic databases. For the same reason, the temperature, pressure and salinity dependence of the above constants is not at all straightforward and must be carefully modelled. Consequently, simple chemical equilibrium constants are of limited use in the numerical solution of equilibria. On the contrary, employing the parametrisation taken from literature, and using codes for the resolution of simultaneous reactions, results can be obtained in a matter of seconds.

11.4 The fundamental equilibrium relationships

In order to tackle one or more interconnected chemical reactions, the first step is to establish a set of thermodynamic equlibrium constants and mathematically find one or more concentrations that are able to satisfy them. The picture we obtain is an equilibrium one, and at this stage no information is given on how long it takes to the system to reach that chemical equilibrium. In particular heterogeneos reactions (like calcite/aragonite formation from dissolved Ca++ and CO3-- ions) may require long times, like years or decades, to be completed. On the contrary CO2 dissolution in surface seawater requires shorter time intervals to reach equilibrium with all dissolved species (H2CO3,HCO3-,CO3--) deriving from its dissolution. In the following are resumed the available expressions for all the equilibrium constants so far (2021) reported in literature. The reader will see that they all empirical relationships, which however are deducted from experimental accurate measurement. Up to day indeed no reliable theorical treatment for so concentrate salt solutions equilibria is available.

The goal of the procedure is to compute an equilibrium constant by mean of empirical expressions, which account for temperature, hydrostatic pressure, salinity.

The first step is to obtain for a given temperature and salinity, the logarithm (e base) of the equilibrium constant Kc. In a second step this is modified by the hydrostatic pressure in a

new logarithmic value. Finally, in a third step, the from this logarithmic value the true Kc is calculated. In some case the empirical expression, taken from literature, contains base-10 logarithms , which requires a transformation, or temperatures in celsius. Therefor in all the expressions T indicates temperature in K, and Tc in °C.

11.5 CO2 Partial Pressure and Fugacity

Before dealing with chemical equations in seawater, we should first focus on the compound in air that starts a series of reactions when dissolved, namely carbon dioxide. Its partial pressure is continuously monitored, by different stations around the world, the most famous one being the Mauna Loa Observatory (Hawaii).

The partial pressure of a gas in a mixture of gases is simply the total pressure multiplied by its mole fraction. However**,** the activity of CO2 is not exactly equal to its partial pressure. For accurate calculations, the fugacity of CO2, fCO2, may be used instead of its partial pressure. The fugacity of CO2 is numerically very similar to CO2 partial pressure in atm**,** and therefore corresponds to CO2 ppm in dry air by the Dalton law. The fugacity can be calculated from its partial pressure (Koerzinger,1999 Zeebe,2001), requiring two virial coefficients B and δ , which are defined as (Weiss,1974) :

$$B=(-1636.75+12.0408*T-3.27957\cdot10^{-2}\cdot T^2+3.16528\cdot10^{-5}\cdot T^3)\cdot10^{-6}$$
$$\delta=(57.7-0.118\cdot T)\cdot10^{-6} \qquad T[K],\quad B[m^3\,mol^{-1}]$$

$$f(CO_2) = p(CO_2)\cdot\exp(P_{tot}\frac{B+2\delta}{RT}) \qquad P_{tot}[Pa] \qquad f(CO_2), p(CO_2)[\mu atm] \qquad T[K] \qquad R=8.314[JK^{-1}]$$

```
fCO2 = ppmCO2*exp(101325*((-1636.75 + 12.0408*T - 3.27957e-2*T^2 +
3.16528e-5*T^3)*1e-6 + 2*(57.7 - 0.118*T)*1e-6)/R/T);
```

$f(CO_2)$ shall be corrected by accounting for water partial pressure (pH2O) . Indeed it refers to dry air, so it must be slightly adjusted for vapor pressure of water at temperature T. This in turn depends on relative humidity (rH) and saturated water pressure, calculated according to the algorithm proposed by Weiss and Price,1980. This is calculated as a first step :

$$p(H_2O)=\exp\{24.4543-\frac{6745.09}{T}-4.8489*\log(\frac{T}{100})-0.000544*S\}$$

$$T[K], \qquad S[(g-salt)/(kg-soln)] \qquad p(H_2O)[atm], \quad \log=\log_e$$

Then $f(CO_2)$ is corrected introducing rH, althought this value is usually left to 100 in the simulation (air is saturated over ocean surface).

$$f(CO_2)_{corr} = f(CO_2)\cdot[1 - p(H_2O)\cdot\frac{rH}{100}] \qquad f(CO_2)[\mu atm] \qquad 0\le rH\le100$$

```
LnpH2O = 24.4543 -6745.09/T - 4.8489*log(T/100) -0.000544*S;

fCO2 = fCO2*(1 - exp(LnpH2O)*pH2O/100);
```

where pH2O indicates the saturation percent at sea level. If not indicated , this value defaults to 100%, as the air layer above sea surface is likely 100% saturated in water vapor.

11.6 Ionic Strength and Ionic Activity

In seawater single ions are not dissolved in pure water, but seawater itself contains high concentrations of ions. The parameter used to characterize aqueous solutions with different amounts of oppositely charged electric charges (ions) is ionic strength, I . It is defined as half the summation of the concentrations multiplied by the respective squared ionic charge (z).

$$I=\frac{1}{2}\ \Sigma c_i \cdot z_i^2$$

The sum encompasses all ions present in the medium so that, for a NaCl solution, we have

$$I=\frac{1}{2}\ ([Na^+]+[Cl^-])$$

Although NaCl is the salt most responsible for the salinity of water, the properties of seawater and a pure NaCl solution with the same concentration are different. For the standard seawater composition used here the ionic strength is approximately 0.7**,** which corresponds to a salinity of around 35 (grams of salts per kg of water) :

```
Cl  = 0.54586    ' Cl-   Mol/kg(solution)
Na  = 0.46906    ' Na+   Mol/kg(solution)
Mg  = 0.05282    ' Mg++  Mol/kg(solution)
Ca  = 0.01028    ' Ca++  Mol/kg(solution)
SO4 = 0.02824    ' SO4-- Mol/kg(solution)
K   = 0.01021    ' K+    Mol/kg(solution)
Br  = 0.00084    ' Br-   Mol/kg(solution)
Sr  = 0.00009    ' Sr++  Mol/kg(solution)
F   = 0.00007    ' F-    Mol/kg(solution)
B   = 0.00042    ' B(OH)3 + B(OH)4- Mol/kg(solution)
```

The ionic strength of seawater may be calculated from salinity (DOE, 1994)

$$I=\frac{19.924}{1000-1.005\cdot S}$$

The behaviour of an ion dissolved in water depends on the electrical interaction with the other ions present in solution. Therefore the chemical 'activity' of an ion dissolved in fresh water and in seawater is quite different.
The activity of a chemical species, denoted by $\{A\}$, is strictly related to its concentration by the activity coefficient γ(A) :

$$\{A\}=\gamma(A)\cdot[A]$$

For infinite dilution, the activity coefficient is 1**,** but it decreases as the solution becomes more concentrated. If you consider a simple electrolyte, deviation from ideal behaviour can be described through the effect of (relatively) long-range electrostatic interactions. For those interactions**,** approximations can be derived to describe the dependence of activity coefficients $\gamma(i)$ on ionic strength I .
Seawater has however a higher ionic strength which is, in turn, due to the presence of different electrolyte charges: the combination of these two facts leads to the formation of 'ion pairing' and complex formation in the electrolyte mixture.

As an example, bivalent charged carbonate ions may associate to positively charged Mg^{++} or Na^{+} ions forming aliovalent ion pairs such as $NaCO3^{-}$ or $MgCO3^{0}$
According to Skirrow (1975), the most important ion pairing equilibria in seawater are:

$Ca^{++} + CO3^{--} <==> CaCO3^{0}$
$Mg^{++} + CO3^{--} <==> MgCO3^{0}$
$Na^{+} + CO3^{--} <==> NaCO3^{-}$
$Ca^{++} + HCO3^{--} <==> CaHCO3^{+}$
$Mg^{++} + HCO3^{--} <==> MgHCO3^{+}$
$Na^{++} + HCO3^{--} <==> NaHCO3^{+}$

Therefore electrostatic interaction of the $CO3^{--}$ ions with opposite charges in solution isn't the only factor that decreases the activity of the ion. The ion pairing greatly impairs the same activity, as the carbonate ion in seawater is not 'free', being combined in neutral or lower charged (aliovalent) species in solution.
If no ion-pairing would occur, the free activity coefficient γ of an ion in simple electrolyte solutions varies with ionic strength I according to the Debye-Hückel limiting law

$$\log(\gamma) = -Az^2\sqrt{I} \qquad \text{valid for} \quad I<0.005$$

or to the Davies equation

$$\log(\gamma) = -Az^2\left(\frac{\sqrt{I}}{\sqrt{I}+1}-0.2\cdot I\right) \quad \text{valid for} \quad I<0.5$$

with $A=1.82\cdot10^{-6}\cdot(\epsilon T)^{-1.5}$, where ε ≈ 79 is the dielectric constant of water, and T is the absolute temperature in K. At 25°, A is about 0.5 for water; z indicates the charge of the ion and I the ionic strength of the solution.

As the ionic strength of seawater is approximately 0.7, which is only slightly higher than the limit of the Davies equation, it should be used in a reasonable way. However, this equation and the Debye-Hückel limiting law no longer apply since they only hold for dilute solutions and simple electrolytes (as opposed to concentrated solutions and electrolyte mixtures of unlike charges) (Zeebe 2001).

The problem will be tackled by using empirical formulas for the equilibrium constants, which employ concentrations, and without the need to calculate activities. The same approach will be used for the temperature, pressure and salinity dependence of these constants.
The fitting of experimental data has been carried out by DOE 1994, Millero 1995, Weiss 1974 et al. A comprehensive review of such data can be found in Zeebe 2001, Appendix A.

11.7 Hydration of carbon dioxide $CO2 + H2O <==> H2CO3$ (K_0)

$$K_0 = \frac{[H_2CO_3^*]}{f(CO_2)}$$

```
LnK0 = 9345.17/T - 60.2409 + 23.3585*log(T/100) + S*(0.023517 -
0.00023656*T + 0.0047036*(T/100)^2);

K0 = exp(LnK0)
```

$T[K]$ $S[g/kg-soln]$, $\log = \log_e$

As the hydration of carbon dioxide occurs in the surface layer of seawater, to this equilibrium is not assigned a pressure dependent constant.

H2CO3* is the sum of the true acid form H2CO3 and the hydrated CO2, indicated by CO2(aq). From now on asterisk is omitted, so that $[H_2CO_3]$ indicates the sum of acid form and hydrated CO2 . The algorithm is taken from Zeebe,2001

$f(CO_2)$ is the fugacity of CO2, which is numerically very similar to CO2 partial pressure in μatm or ppmv (parts per million in vomume) , if total pressure is 1 atm. by the Dalton law. For sake of simplicity in some of the script proposed here, the CO2 fugacity is approximated by its ppmv concentration, which in turn corresponds to its partial pressure in μatm for an total pressure of 1 atm..

11.8 'Ionic water product' , i.e. the equilibrium constant for the reaction

2 H2O <==> H3O+ + OH- (Kw) $K_w = [H^+] \cdot [OH^-]$

Water itself is a weak electrolyte whose dissociation must be carefully taken into account. In seawater the following expression is used to represent its dependence on temperature and salinity. As is the case for the following expressions, it should not be extrapolated to zero or near zero salinity, as it results from experiments with salinity from 25 to 45 (grams-of-salts/Kg-of-solution) (Zeebe,2001).

The equilibrium constant is first calculated as log(e) in the variable LnKw, as a function of two parameters, temperature T(K) and salinity S(grams of salt/kg of solution).

In a second program line the effect of hydrostatic pressure is accounted for, thus obtaining the LnKwP variable, eventually transformed in Kw, as listed below. The concentration of H+ ions and consequently the pH scale refer to the 'total scale' , as discussed in section 11.18.

```
LnKw = 148.9802 - 13847.26/T - 23.6521*log(T) + (118.67/T - 5.977 +
1.0495*log(T))*S^0.5 - 0.01615*S;

LnKwP = LnKw + (25.6 - 0.2324*Tc + 3.6246e-3*Tc^2)/R1/T*P + 0.5*(-5.13e-3 +
0.0794e-3*Tc)/R1/T*P^2;

Kw = exp(LnKwP);
```

$[H^+] = Total\ scale$, $pH = pH_{SWS}$

11.9 First dissociation of carbonic acid H2CO3 <==> H^+ + $HCO3^-$ (K_1)

$$K_1 = \frac{[HCO_3^-] \cdot [H^+]}{[H_2CO_3]}$$

The computing procedure follows the three steps indicated in the preceding session. The concentration of H+ ions and consequently the pH scale refer to the 'free scale' , as discussed in section 11.18.

```
LnK1 = -log(10)*(6320.813/T + 19.568224*log(T) -126.34048 + 5.592953*S^0.5
+ 0.028845*S - (6.388e-5)*S^2 + (-225.7489*S^0.5 - 4.761*S)/T -
0.8715109*S^0.5*log(T));

LnK1P = LnK1 + (25.5 - 0.1271*Tc)/R1/T*P + 0.5*(-3.08e-3 +
0.0877e-3*Tc)/R1/T*P^2;

K1 = exp(LnK1P);
```

(Waters,2014) $T[K]$ $S[g/kg-soln]$ $\log=\log_e$, $pH=Free\,Scale$

11.10 Second dissociation of carbonic acid $HCO3^-$ <==> $H^+ + CO3^{--}$ (K_2)

$$K_2=\frac{[CO_3^{--}]\cdot[H^+]}{[HCO_3^-]}$$

The computing procedure follows the three steps indicated in the preceding session. The concentration of H+ ions and consequently the pH scale refer to the 'free scale' , as discussed in section 11.18.

```
LnK2 = -log(10)*(5143.692/T + 14.613358*log(T) - 90.18333 + 13.396949*S^0.5
+ 0.12193009*S - (3.8362e-4)*S^2 + (-472.8633*S^0.5 - 19.03634*S)/T -
2.1563270*S^0.5*log(T));

LnK2P = LnK2 + (15.82 + 0.0219*Tc)/R1/T*P + 0.5*(1.13e-3 -
0.1475e-3*Tc)/R1/T*P^2;

K2 = exp(LnK2P);
```

(Waters,2014) $T[K]$, $\log=\log_e$, $pH=Free\,Scale$

11.11 Solubility of calcite (K_{sp}) $CaCO3$ <==> $Ca^+ + CO3^{--}$

$$K_{SP}(cal)=[Ca^{++}]\cdot[CO_3^{--}]$$ [ ] = conc. in mol/(kg-solution)

The solubility product, which equals the equilibrium constant, is first calculated as log(10) in the variable LogKspCal, then transformed into log(e) in the variable LnKspCal. The two parameters are temperature T(K) and salinity S(grams of salt/kg of solution).

In a third program line the effect of hydrostatic pressure is accounted for, thus obtaining the LnKspCalP variable, eventually transformed in Ksp1, as listed below.

```
LogKspCal = -171.9065 - 0.077993*T + 2839.319/T + 71.595*log10(T) +
(-0.77712 + 0.0028426*T + 178.34/T)*S^0.5 - 0.07711*S + 0.0041249*S^1.5;

LnKspCal = LogKspCal*log(10);

LnKspCalP = LnKspCal + (48.76 - 0.5304*Tc)/R1/T*P + 0.5*(-11.76e-3 +
0.3692e-3*Tc )/R1/T*P^2;
```

```
Ksp1 = exp(LnKspCalP)
```

(Zeebe,2001) $T[K]$, $\log = \log_{10}$, $S[g/kg-soln]$

11.12 Solubility of aragonite (K_{sp}) CaCO3 <==> Ca^{+} + $CO3^{--}$

The procedure is similar to that for calcite (11.10). Aragonite is a second crystallographic form of calcium carbonate.

$K_{SP}(ara)=[Ca^{++}]\cdot[CO_3^{--}]$ [ ] = conc. in mol/(kg-solution)

```
LogKspAra = -171.945 - 0.077993*T + 2903.293/T + 71.595*log10(T) + (-
0.068393 + 0.0017276*T + 88.135/T)*S^0.5 - 0.10018*S + 0.0059415*S^1.5;

LnKspAra = LogKspAra*log(10);

LnKspAraP = LnKspAra + (46 - 0.5304*Tc)/R1/T*P + 0.5*(-11.76e-3 +
0.3692e-3*Tc)/R1/T*P^2;

Ksp2 = exp(LnKspAraP);
```

$$K_{SP}(ara)=10\wedge\{-171.945-0.077993\cdot T+\frac{2903.293}{T}+71.595\cdot\log_{10}(T)+(-0.068393+0.0017276\cdot T+\frac{88.135}{T})\cdot S^{0.5}-0.10018\cdot S+0.0059415\cdot S^{1.5}\}$$

(Zeebe,2001) $T[K]$, $\log = \log_{10}$, $S[g/kg-soln]$

11.13 Dissociation of boric acid B(OH)3 + H2O <==> $B(OH)4^{-}$ + H^{+}

$$K_B=\frac{[H^+]\cdot[B(OH)_4^-]}{[B(OH)_3]}$$

```
LnKB = (-8966.9 - 2890.53*S^0.5 - 77.942*S + 1.728*S^1.5 - 0.0996*S^2)/T +
148.0248 + 137.1942*S^0.5 + 1.62142*S - (24.4344 + 25.085*S^0.5 +
0.2474*S)*log(T) + 0.053105*S^0.5*T;
```

Then LnKB is modified by the hydrostatic pressure P

```
LnKBP = LnKB + (29.48 - 0.1622*Tc - 2.608e-3 * Tc^2)/R1/T*P +
0.5*(-2.84e-3)/R1/T*P^2;
```

Finally the value of equilibrium constant is calculated as

```
KB = exp(LnKBP)
```

Ref. Zeebe,2001 T(K), pH = Free Scale , P(atm)

11.14 Second dissociation of sulfuric acid $HSO4^{-}$ <==> $SO4^{--}$ + H^{+}

$$K_S=\frac{[H^+]\cdot[SO_4^{--}]}{[HSO_4^-]}$$

I = Ionic strength, necessary for KS and KF calculations, is calculated from salinity S by means of an empirical expression (Zeebe 2001)

```
I = 19.924*S/(1000 - 1.005*S);

LnKS = -4276.1/T + 141.328 - 23.039*log(T) + (-13856/T + 324.57 -
47.986*log(T))*I^0.5 + (35474/T - 771.54 + 114.723*log(T))*I -2698/T*I^1.5
+ 1766/T*I^2 + log(1-0.001005*S);

LnKSP = LnKS + (18.03 - 0.0466*Tc - 0.3160e-3*Tc^2)/R1/T*P + 0.5*(-4.53e-3
+ 0.09e-3*Tc)/R1/T*P^2;

KS = exp(LnKSP);
```

(Zeebe,2001) $T[K]$, $pH = Free\ Scale$

11.15 Dissociation of hydrofluoric acid HF <==> H^+ + F^-

$$K_F = \frac{[H^+]\cdot[F^-]}{[HF]}$$

```
I = 19.924*S/(1000 - 1.005*S);

LnKF = 1590.2/T - 12.641 + 1.525*I^0.5 + log(1-0.001005*S);

LnKFP = LnKF + (9.78 + 0.009*Tc + 0.942e-3*Tc^2)/R1/T*P + 0.5*(-3.91e-3 +
0.054*Tc)/R1/T*P^2;

KF = exp(LnKFP);
```

(Zeebe,2001) $T[K]$, $pH = Free\ Scale(*)$, $I = ionic\ strenght$, $\log = \log_e$

(*) Differently from Zeebe,2001 K_F value has NOT been converted to pH Total Scale, so pH Free Scale is used. See 11.18 for a description of the different pH scales used in seawater chemistry.

11.16 The three dissociations of phosphoric acid H3PO4

For the three dissociation reactions of phosphoric acid in seawater and the following of silicic acid in seawater, the formulae for empiric constants are reported, but these are not used at the present for the calculation in the 'SeaWaterCalc' code. The concentration of the two acids is very low in seawater and locally variable. They could easily be implemented by inserting them in the code.

$$K_{1P} = \frac{[H^+]\cdot[H_2PO_4^-]}{[H_3PO_4]} \qquad H_3PO_4 \Leftrightarrow H^+ + H_2PO_4^-$$

$$K_{1P} = \frac{[H^+]\cdot[HPO_4^{--}]}{[H_2PO_4^-]} \qquad H_2PO_4^- \Leftrightarrow H^+ + HPO_4^{--}$$

$$K_{1P} = \frac{[H^+]\cdot[PO_4^{---}]}{[HPO_4^{--}]} \qquad HPO_4^{--} \Leftrightarrow H^+ + PO_4^{---}$$

```
LnKa1 = -4576.752/T + 115.525 - 18.453*log(T) + (-106.736/T+0.69171)*S^0.5
+ (-0.65643/T-0.01844)*S;  % 1st dissociation constant

LnKa2 = -8814.715/T + 172.0883 - 27.927*log(T) + (-160.34/T+1.3566)*S^0.5 +
(0.37335/T-0.05778)*S;  % 2nd dissociation constant

LnKa3 = -3070.75/T - 18.141 + (17.27039/T+2.81197)*S^0.5 + (-44.99486/T -
0.09984)*S; % 3rd dissociation constant

Ka1 = exp(LnKa1);Ka2 = exp(LnKa2);Ka3 = exp(LnKa3);
```

(Zeebe,2001) $T[K]$, $pH = Total\,Scale\,(see\,11.18)$, $S[g/kg-soln]$

11.17 Dissociation of orthosilicic acid Si(OH)4 <==> H^+ + H3SiO4$^-$

$$K_{Si} = \frac{[H^+]\cdot[H_3SiO_4^-]}{[H_4SiO_4]}$$

```
I = 19.924*S/(1000 - 1.005*S); % ionic stregth

LnKSi = -8904.2/T + 117.385 - 19.334*log(T) + (-458.79/T + 3.5913)*I^0.5 +
(188.74/T - 1.5998)*I + (0.07871 - 12.1652/T)*I^2 + log(1-0.001005*S);

KSi = exp(LnKSi);
```

(Zeebe,2001) $T[K]$, $[H^+] = Total\,Scale\,(see\,11.18)$,

11.18 Effect of pressure on the equilibrium reactions

The effect of pressure on equilibrium constants is of paramount importance; sinking down into the depths of the oceans, pressure increases by 1 atm with every 10 meters. As the intermolecular distances between water molecules decrease slightly, the density of liquid water increases accordingly. Therefore, interionic interaction and equilibrium constants become progressively altered in relation to the pressure itself. The effects become noticeable when pressure reaches hundreds of bars; pH and solubility of calcium carbonate alter to such an extent that aragonite oversaturation, and calcite at greater depths, disappear and, if formed, these salt readily re-dissolve.

As discussed in Section 4.4 on water density, we recall that water pressure is measured in 'bars' (1 atm = 1.01325 bar; 1 bar = 0.1 Mpa), and that the surface pressure of the sea is assumed to be zero.

The effect of pressure on equilibrium constants can be calculated (Millero 1995) according to a second order polynomial expression of the natural logarithm of the ratio between Ki,P (the value of i-esimal constant at pressure P) and Ki,0 (the value of i-esimal constant at reference zero pressure P)

$$\log\left(\frac{K_{i(P)}}{K_1}\right) = -\frac{\Delta V_i}{R^* T}P + \frac{\Delta K_i}{2\cdot R^* T}P^2 \qquad \log = \log_e \qquad T[K] \qquad P = \text{bar}$$

The constant value R^* is given by $R^*=83.14472\ cm^3 \text{bar}\, mol^{-1} K^{-1}$ (corrected from the original value of 83.131, reported in Zeebe,2001 slightly biased) whereas ΔV_i is the molal volume change, and ΔK_i the compressibility change. They are in turn deconvoluted in terms of a second order polynomial which, strictly speaking, is only valid for Salinity = 35, but can be acceptable in a wider range, say 20-50.

$\Delta V_i = a_0 + a_1 T_c + a_2 T_c^2 \qquad T=[°C]$
$\Delta K_i = b_0 + b_1 T_c \qquad$ T=[°C]

The values for the a_i and b_i parameters are taken from Zeebe,2001 and are reported here for each of the reactions, where Tc indicates the temperature in °C, P the pressure in bar, and T the absolute temperature (T = Tc + 273.15) and finally R* = 83.14472.

	a0	a1	a2	b0	b1
K1(P)	-25.50	0.1271	0.0	-0.00308	0.0000877
K2(P)	-15.82	-0.0219	0.0	0.00113	-0.0001475
KB(P)	-29.48	0.1622	0.002608	-0.00284	0.0
KW(P)	-25.60	0.2324	-0.0036246	-0.00513	0.0000794
KS(P)	-18.03	0.0466	0.0003160	-0.00453	0.00009
KF(P)	-9.780	-0.0090	-0,000942	-0.00391	0.000054
KSP,cal(P)	-48.76	0.5304	0.0	-0.01176	0.0003692
KSP,ara(P)	-46.00	0.5304	0.0	-0.01176	0.0003692

The effect of pressure alters the values for every equilibrium constant, and must therefore be properly accounted for in the program flow. Greater detail on this can be found by inspecting the program script.

11.19 Oversaturation of CaCO3

Generally speaking, in heterogeneous reactions, reactants are in different phases, like solids, liquid solutions or gaseous mixtures. One of the most relevant of such reactions is the formation or dissolution of calcium carbonate (solid/solution), according to its oversaturation value, indicated by **Ω** and discussed elsewhere.

Once formed biologically by calcifying organisms or by inorganic route, and with a density greater than 1, it eventually sinks into the dark abyss. Due to increasingly high pressure, solid CaCO3 begins to dissolve below a certain depth**,** referred to as the saturation horizon where **Ω** is exactly equal to 1**.** Dissolution of the solid is not instantaneous, and the downward flux continues to a depth where the solid particles of calcium carbonate are completely dissolved. This depth is called the carbonate compensation depth. If the sea bottom does not reach such a depth, it becomes undissolved carbonate sediment. The two crystallographic forms of CaCO3, calcite and aragonite, have different solubility products, the former being less soluble. Therefore**,** the saturation horizon and the compensation depth for aragonite are at a higher level compared to calcite. Most calcifying organisms (e.g. Coccolithophores) produce calcite, whilst coral reefs are made of aragonite.

Solving the equilibria involved in CaCO3 formation with the algorithm described the oversaturation profiles at different depths can be calculated. Some of the results can be seen in the graph in Figure . The curious reader, intent on modifying input parameters and looking for new results (in a "see what happens" procedure) is re-directed to the script.
In Fig. 1 oversaturation **Ω**, is plotted against pressure being temperature fixed at 4°C, which is the overall temperature for the ocean's depths below the thermocline (about 300 meters). Below **Ω**=1 (light blue area) carbonates begin their dissolution process.

One topic frequently debated today is the potential hazard for coralline reefs of the rising concentrations of CO2, through the reduction of ocean pH and carbonate ion concentration. The effect of this, is however compensated for by an increase in oversaturation in warmer areas of oceans, where calcifying organisms and coral reefs prosper. Global warming, estimated at about 1°C from the beginning of the twentieth century to the present day, also favours oversaturation and thereby counteracts the effects of increasing CO2 content by anthropogenic emissions. Therefore the two data should be considered together to gain a complete picture.

11.20 Dissolved Inorganic Carbon (DIC)

In ocean chemistry it is a coventional term used to indicate the sum of the concentration of seawater-dissolved CO2 , carbonic acid, bicarbonate and carbonate anions. It is measured in millimol/kg-of-solution. This value depends on different factors, the main being temperature, salinity and CO2 concentation in the atmosphere

$$DIC[mmol/kg-soln] = [H_2CO_3]^* + [HCO_3^-] + [CO_3^{--}] \qquad \text{where} \qquad [H_2CO_3]^* = [CO_2]_{aq} + [H_2CO_3]$$

Salinity can be varied by user , the default value being 35 g/kg-soln. The imbalance (difference between cations and anions, not accounting for H+ and OH-) depends linearly on salinity, according to :

Imb = 0.00218*35/S (S = salinity in g/kg-soln)

The program calculates the equilibrium values of DIC as a function of temperature and CO2 concentration for a given salinity. It should be considered that seawater reaches the thermodinamic equilibrium with a delay of some months.

The simple script is listed here below.

```
clear;clc;format shortE;format compact;
global K1 K2 Kw H2CO3 HCO3 CO3 OH H Imb

function y = neut(pH) % ===> pH free scale, i.e. chemical true scale, always used if not
otherwise stated !!
  global K1 K2 Kw H2CO3 HCO3 CO3 OH H Imb
  H = 10^(-pH);
  HCO3 = K1*H2CO3/H;
  CO3 = K2*HCO3/H;
  OH = Kw/H;
  y = Imb + H - OH - HCO3 - 2*CO3;
endfunction

R = 8.314;
i = 0;S = 35; % salinity, in grams of salts in 1 kg of solution , this can be varied
for Tc = 0:2:30 % temperature in Celsius (°C)
  ++i;j = 0;
```

```
  T = Tc + 273.15;Imb = 0.00218;
  Imb = Imb*35/S;
  % H2CO3 <==> H+ + HCO3-  (Waters,2014)
  LnK1 = -log(10)*(6320.813/T + 19.568224*log(T) -126.34048 + 5.592953*S^0.5 + 0.028845*S -
(6.388e-5)*S^2 + (-225.7489*S^0.5 - 4.761*S)/T -0.8715109*S^0.5*log(T));
  % HCO3- <==> H+ + CO3--  (Waters,2014)
  LnK2 = -log(10)*(5143.692/T + 14.613358*log(T) - 90.18333 + 13.396949*S^0.5 +
0.12193009*S - (3.8362e-4)*S^2 + (-472.8633*S^0.5 - 19.03634*S)/T -
2.1563270*S^0.5*log(T));
  % CO2 + H2O <==> H2CO3 (Zeebe,2001)
  LnK0 = 9345.17/T - 60.2409 + 23.3585*log(T/100) + S*(0.023517 - 0.00023656*T +
0.0047036*(T/100)^2);
  % H2O <==> H+ + OH- (DOE 1994)  (Zeebe,2001) pH = pH(SWS) considered = pH free !
  LnKw = 148.9802 - 13847.26/T - 23.6521*log(T) + (118.67/T - 5.977 + 1.0495*log(T))*S^0.5
- 0.01615*S;
  K1 = exp(LnK1);K2 = exp(LnK2);K0 = exp(LnK0);Kw = exp(LnKw);
  for ppmCO2 = 300:10:500  % parts per million (in volume) of CO2 in the atmosphere
    H2CO3 = K0*ppmCO2*1e-6;++j;
    [pH,fval,info] = fzero(@neut,[0,14]); % call to fzero
    DIC(i,j) = H2CO3 + HCO3 + CO3;
    temp(i,j) = Tc;ppm(i,j) = ppmCO2;
  endfor
endfor
plot(temp(:,11),DIC(:,11)*1000,'r','LineWidth',2);grid on;grid minor on;xlabel('temperature
°C');ylabel('DIC [mmol/kg]')
title('DIC versus temperature @ 400 ppm CO2');
figure;
plot(ppm(11,:),DIC(11,:)*1000,'r','LineWidth',2);grid on;grid minor on;xlabel('ppm CO2 in
dry air');ylabel('DIC [mmol/kg]')
title('DIC versus ppmCO2 @ 20 °C');
figure;
surf(temp,ppm,DIC);colorbar;title('DIC vs. temperature and ppmCO2');
xlabel('temperature °C');ylabel('ppm CO2 in dry air');zlabel('DIC');
```

Two nested loops are evident, the innermost being for ppmCO2 and the outermost for temperature. The CO2 concentration (parts per million) varies from 300 to 500 (step 10) and temperature varies from 0 to 30°C (step 2°C). For each pair of ppmCO2 and temperature the statement *fzero* is invoked for the '*neut(pH)*' function. This function finds the electric charge imbalance of the solution. By varying the pH scale between 0 and 14, *fzero* tries to find the y value as close as possible to zero

```
y = Imb + H - OH - HCO3 - 2*CO3;
```

```
[pH,fval,info] = fzero(@neut,[0,14]);
```

Once pH satisfying the above condition is determined, the concentrations of H2CO3, HCO3-,CO3--,OH- follow as a consequence, and hence DIC is calculated

The script produces three figures, for a given salinity value. The first two are bidimensional plots, the third is a tridimensional one

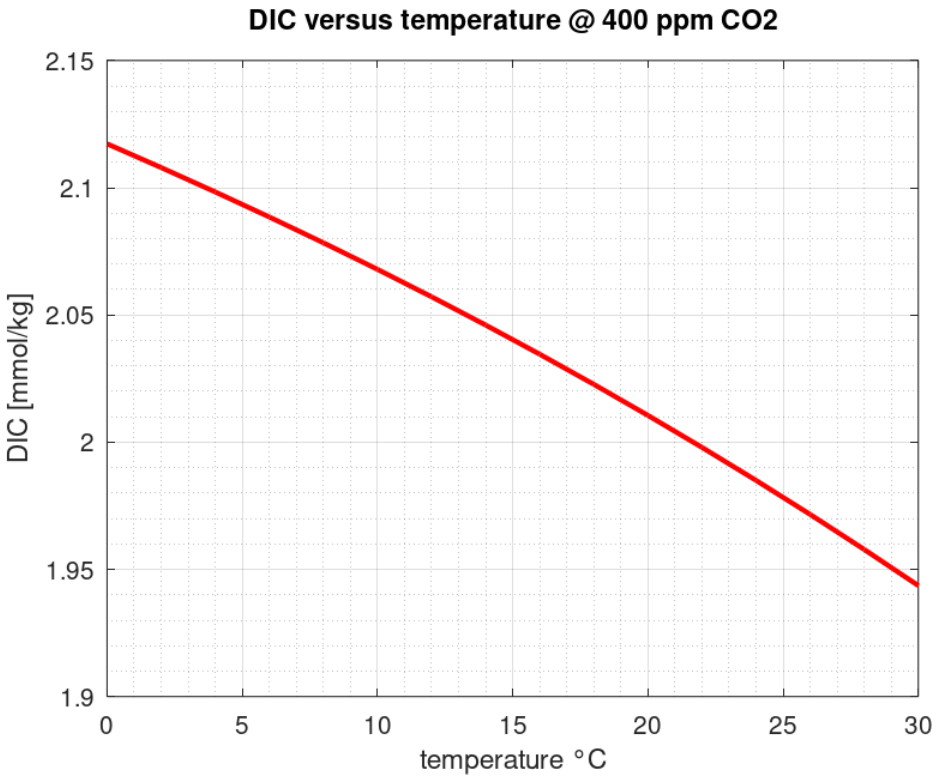


Fig 11.1 DIC as a function of temperature (S = 35; ppmCO2 = 400)

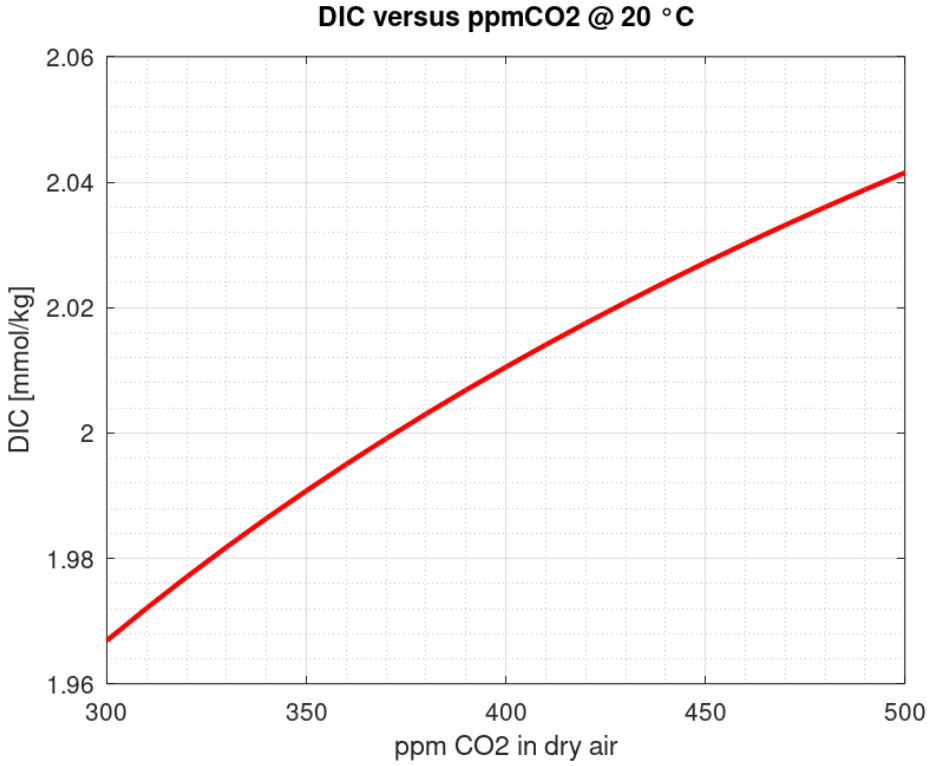


Fig 11.2 DIC as a function of ppmCO2 (S = 35; temperature = 20°C)

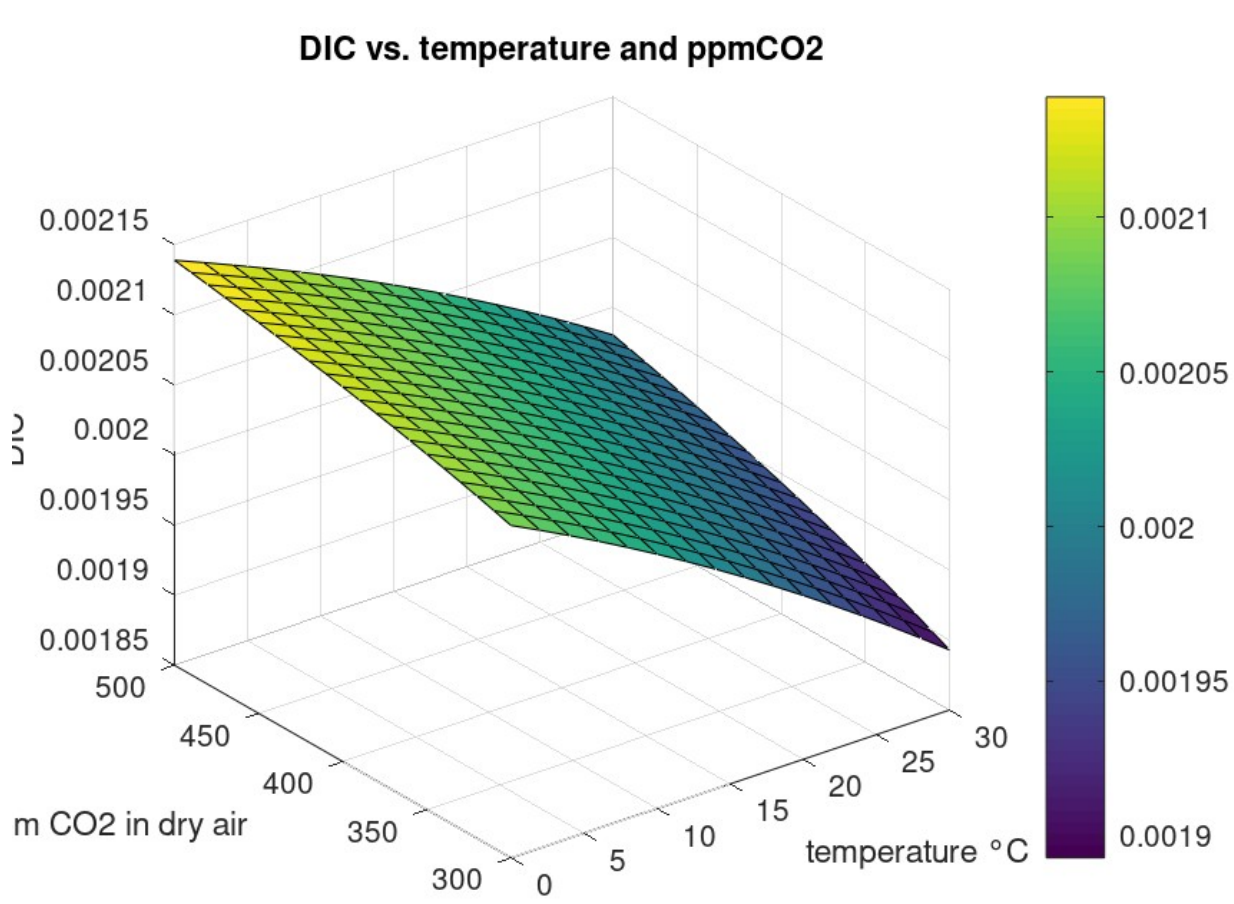


Fig 11.3 DIC as a function of both temperature and ppmCO2 (S = 35)

11.21 Different pH scales in Seawater Chemistry

In chemical oceanography**,** three main different pH scales are currently used; free, total and seawater. This point is not to be neglected; when dealing with acidity constants of hydrogen ion transfer reactions (as is the case of H2CO3) the use of a consistent pH scale is mandatory. The values of different pH scales in seawater differ by up to 0.12 units (Zeebe 2001).

The pH value is in theory defined as the negative logarithm of the activity (in braces) of hydrogen ions

$$pH = \log_{10}\{H^+\}$$

Unfortunately**,** individual ion activities cannot be determined experimentally. Indeed**,** the concentration of one single ion cannot be varied independently, because electroneutrality is required. Therefore the 'free' pH scale for seawater has been proposed:

$$pH_F = \log_{10}[H^+]_F$$

where $[H^+]_F$ stands for the free hydrogen ion concentration, including hydrated forms, like $H3O^+$ and $H9O4^+$ (Dickson 1984)

Indeed, free protons do not exist in any significant amount in aqueous solutions. Rather**,** the proton is bonded to a water molecule thus forming $H3O^+$. This in turn is hydrogen bonded to three other water molecules to form an $H9O4^+$ ion (Dickson 1984, p.2299).

To be noted that, as usual in ocean chemistry, the concentrations in square brackets are expressed in mol/Kg (water) and not in mol/L as is more usual in general chemistry.

In 1973, Hanson defined a total scale for pH so as to include the effect of sulphate ions in its definition.

$pH_T = \log_{10}[H^+]_T$ where $[H^+]_T = [H^+] + [HSO_4^-]$

The bisulphate ion (HSO_4^-) is a rather weak acid ($Ka1 \approx 2.1 \cdot 10^{-2}$) so it is not completely dissociated in H^+ and SO_4^{--} ions. Once the Ka is known, a relationship between the two scales can be inferred. This in turn requires an accurate value of Ka in seawater, which would be difficult to obtain. But by using Hanson's total pH scale in seawater, the calculation of Ka for bisulphate ion can be avoided.

This total scale will be used in the empirical expressions for the various acidity constants, including the ionic water product. This choice seems to conflict with the usual free pH scale as used in general chemistry, but is necessary due the use of the total scale in the experimental determination of the various constants. Therefore, in the code shown below, this total scale will be employed.

The third scale is the so-called seawater scale, which only slightly differs from the preceding one.

The need to introduce this scale is due to the presence of fluoride ions (F^-) in seawater.

Consequently, we have to account for the protonation of F^- ions according to the equilibrium:

HF <==> H^+ + F^- with a $Ka2 \approx 3.5 \cdot 10^{-4}$

Indeed, hydrofluoric acid is a weak acid. In standard seawater however, the concentration of fluoride ions is $7.0 \cdot 10^{-5}$ Mol/Kg of water, about 400 times lower than the concentration of sulphate ions, $2.8 \cdot 10^{-2}$ Mol/Kg, therefore the seawater scale differs by no more than 0.01 pH units from the total scale. In the following table the transformation between the three pH scales are shown, both in terms of concentrations and pH units.

$$[H^+]_T = [H^+]_F \cdot \left(1 + \frac{[SO_4^{--}]}{K_{a1}}\right)$$

$$[H^+]_{SW} = [H^+]_F \cdot \left(1 + \frac{[SO_4^{--}]}{K_{a1}} + \frac{[F^-]}{K_{a2}}\right)$$

$$[H^+]_{SW} = [H^+]_T \cdot \left(1 + \frac{[F^-]}{K_{a2}}\right)$$

$$pH_T = pH_F - \log_{10}\left(1 + \frac{[SO_4^{--}]}{K_{a1}}\right)$$

$$pH_{SW} = pH_F - \log_{10}\left(1 + \frac{[SO_4^{--}]}{K_{a1}} + \frac{[F^-]}{K_{a2}}\right)$$

$$pH_{SW} = pH_T - \log_{10}\left(1 + \frac{[F^-]}{K_{a2}}\right)$$

If $Ka1 \approx 2.1 \cdot 10^{-2}$ and $[SO_4^{--}] = 2.8 \cdot 10^{-2}$ then the difference between pHT and pHF scale would be ≈ 0.37 pH unit, but this would be valid only in pure water.

Under the same conditions (pure water) then the difference between pHSW and pHT scale would be ≈ 0.37 pH unit.

11.22 Complete equilibria solved at hydrostatic pressure greater than zero

The following script employs the equations 11.5 troughout 11.15, therefore dissociation of phosphoric acid and orthosilicic acid are not considered. Indeed their concentrations are locally widely variable in seawater, so as they are considered apart, in a dedicated script. The script looks complex, however only apparently.

The function *y=neut(pH)* computes imbalance of ionic charges as a function of pH. Solubility of CaCO3 (calcite form) is also taken into account inside the function. In the function the various equilibria are solved, namely first and second dissociation of carbonic acid, dissociation of fluoridric acid, second dissociation of sulfuric acid (the first is considered to be complete), reaction of boric acid. The hydrolysis of calcium and magnesium aqua ions (solvated ions) schematically written as:

$Ca^{++} + H_2O <==> Ca(OH)^+ + H^+$; $Mg^{++} + H_2O <==> Mg(OH)^+ + H^+$

althought occurring to some minor extent, is not considered in here. The contribution of such reactions can be neglected, as demonstrated in some previous calculations.

The pH scale used is the (true) free scale, except in some instances were the sws scale is required (second dissociation of sulfuric acid) (see section 11.19).

Six parameters can be varied in the main for...endfor cycle. They are listed here, together with their default values. However, only one parameter at a time can be varied, by inserting it value and range in the proper script line (in red).

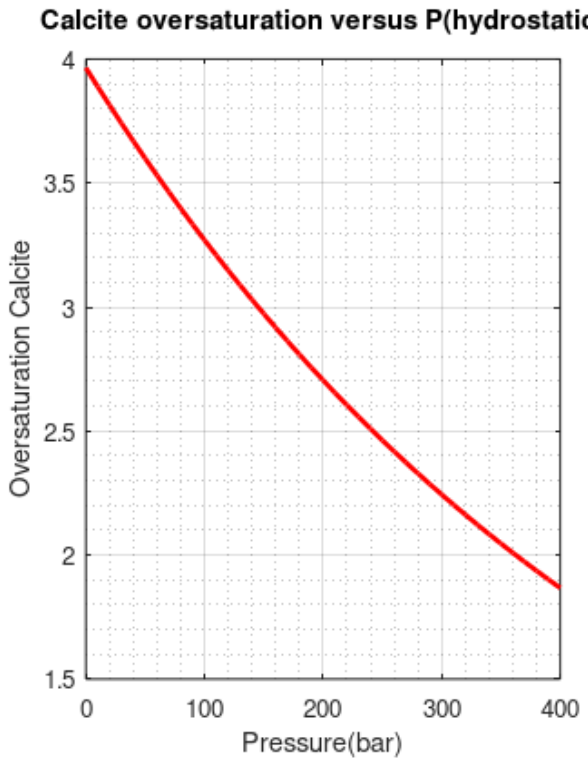
```
Tc = 15;          temperature in Celsius (°C)
ppmCO2 = 410;     parts per million (in volume) of CO2 in the atmosphere
P = 0;            hydrostatic pressure in atm. P=0 means sea surface.
S = 35;           salinity, in grams of salts in 1 kg of solution
pptF = 0.001;     fraction of CaCO3 which actually precipitates
pH2O =  100;      H2O vapour pressure in % of saturation at sea level
```

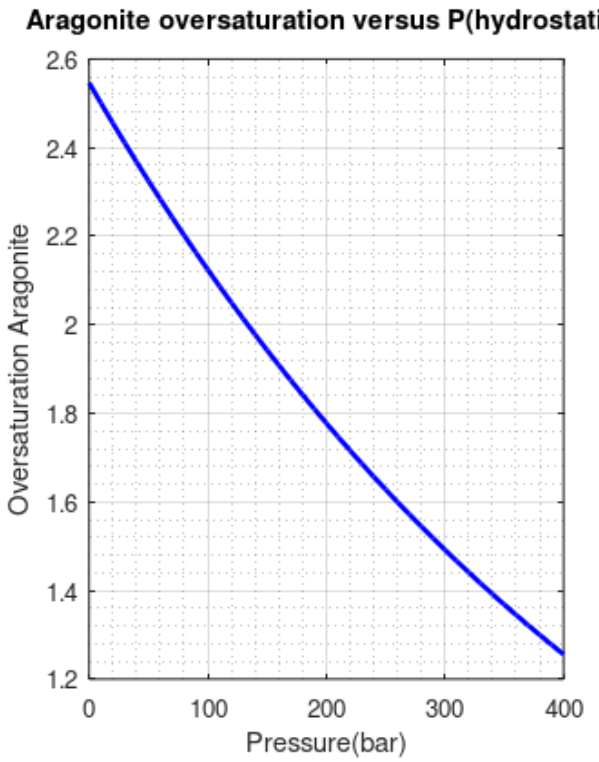
The program output is both numerical and graphic. In the following are reported an output text example, in which the hydrostatic pressure is varied and the corresponding graphics. The complete script is listed in appendix together with some different output text tables as examples.

Output example in command window:

```
Temp = temperature in degree Celsius (°C)
CO2  = ppmv, parts per million in volume of CO2 in dry air
pptF = fraction of CaCO3(calcite) which precipitates, ratio to complete precipitation(when omega=1)
Pres = pressure in atm x 100
Saln = salinity in grams of dissoved salts per kg of solution
pHfr. pHtot pHsws = different pH scales, free, total and seawater
pOH  = -log10[OH-]
H2CO3  HCO3  CO3 = concentrations in millimol/kg-solution
DIC  = Dissolved Inorganic Carbon in mmol/kg-soln
Alk  = alkalinity in mmol/kg-soln  Alk = 2*CO3+HCO3+OH+BOH4-H
Ca++ CaCO3   = concentrations in mmol/kg-soln
sovr1  sovr2 = Calcite and aragonite overasaturation (omega)
pH2O = water vapour pressure above sea level
```

```
Cout = mmol/kg-soln of carbon absorbed(-) or outgassed (+) from the first iteration
+----+---+----+----+----+-----+-----+-----+-----+------+-----+------+-----+------+-----+-----+-----+-----+------+
|Temp|CO2|pptF|Pres|Saln|pHfr.|pHtot|pHsws| pOH |H2CO3 |HCO3-|CO3-- | DIC | Ca++ | Alk |CaCO3|over1|over2| pH2O |
Cout
+----+---+----+----+----+-----+-----+-----+-----+------+-----+------+-----+------+-----+-----+-----+-----+------+
|15.0|410|0.00|0.00|35.0|8.142|8.084|8.075|5.542|0.0151|1.969|0.1665|2.151|10.272|2.385|0.008|3.965|2.546|0.0165|
0.00
|15.0|410|0.00|0.20|35.0|8.134|8.076|8.067|5.541|0.0151|1.971|0.1658|2.152|10.272|2.385|0.008|3.814|2.455|
0.0165|-0.00
|15.0|410|0.00|0.40|35.0|8.126|8.069|8.060|5.540|0.0151|1.972|0.1650|2.152|10.273|2.385|0.007|3.670|2.368|
0.0165|-0.00
|15.0|410|0.00|0.60|35.0|8.118|8.062|8.053|5.539|0.0151|1.974|0.1642|2.153|10.273|2.385|0.007|3.531|2.283|
0.0165|-0.00
|15.0|410|0.00|0.80|35.0|8.110|8.054|8.046|5.538|0.0151|1.975|0.1634|2.154|10.273|2.385|0.007|3.398|2.202|
0.0165|-0.00
|15.0|410|0.00|1.00|35.0|8.101|8.047|8.039|5.537|0.0151|1.977|0.1627|2.154|10.273|2.386|0.007|3.270|2.125|
0.0165|-0.00
|15.0|410|0.00|1.20|35.0|8.093|8.039|8.032|5.535|0.0151|1.978|0.1619|2.155|10.273|2.386|0.007|3.148|2.050|
0.0165|-0.00
|15.0|410|0.00|1.40|35.0|8.085|8.032|8.025|5.534|0.0151|1.979|0.1611|2.156|10.273|2.386|0.007|3.030|1.978|
0.0165|-0.00
|15.0|410|0.00|1.60|35.0|8.077|8.024|8.019|5.533|0.0151|1.981|0.1604|2.156|10.273|2.386|0.007|2.917|1.909|
0.0165|-0.00
|15.0|410|0.00|1.80|35.0|8.069|8.017|8.012|5.531|0.0151|1.982|0.1596|2.157|10.273|2.387|0.007|2.809|1.842|
0.0165|-0.01
|15.0|410|0.00|2.00|35.0|8.061|8.009|8.005|5.530|0.0151|1.984|0.1589|2.158|10.274|2.387|0.006|2.705|1.778|
0.0165|-0.01
|15.0|410|0.00|2.20|35.0|8.053|8.002|7.998|5.529|0.0151|1.985|0.1582|2.159|10.274|2.387|0.006|2.605|1.716|
0.0165|-0.01
|15.0|410|0.00|2.40|35.0|8.044|7.995|7.991|5.528|0.0151|1.987|0.1574|2.159|10.274|2.388|0.006|2.510|1.657|
0.0165|-0.01
|15.0|410|0.00|2.60|35.0|8.036|7.987|7.984|5.527|0.0151|1.988|0.1567|2.160|10.274|2.388|0.006|2.418|1.600|
0.0165|-0.01
|15.0|410|0.00|2.80|35.0|8.028|7.980|7.977|5.526|0.0151|1.990|0.1560|2.161|10.274|2.388|0.006|2.329|1.545|
0.0165|-0.01
|15.0|410|0.00|3.00|35.0|8.020|7.972|7.970|5.525|0.0151|1.991|0.1553|2.161|10.274|2.389|0.006|2.244|1.492|
0.0165|-0.01
|15.0|410|0.00|3.20|35.0|8.012|7.965|7.963|5.524|0.0151|1.993|0.1546|2.162|10.274|2.389|0.006|2.163|1.441|
0.0165|-0.01
|15.0|410|0.00|3.40|35.0|8.004|7.957|7.956|5.523|0.0151|1.994|0.1539|2.163|10.275|2.389|0.005|2.084|1.392|
0.0165|-0.01
|15.0|410|0.00|3.60|35.0|7.996|7.950|7.949|5.523|0.0151|1.996|0.1532|2.164|10.275|2.390|0.005|2.009|1.345|
0.0165|-0.01
|15.0|410|0.00|3.80|35.0|7.988|7.943|7.942|5.522|0.0151|1.997|0.1525|2.165|10.275|2.390|0.005|1.937|1.300|
0.0165|-0.01
|15.0|410|0.00|4.00|35.0|7.980|7.935|7.935|5.522|0.0151|1.999|0.1518|2.165|10.275|2.390|0.005|1.867|1.256|
0.0165|-0.01
+----+---+----+----+----+-----+-----+-----+-----+------+-----+------+-----+------+-----+-----+-----+-----+------+
```


Calcite oversaturation versus P(hydrostatic)
Oversaturation Calcite
Pressure(bar)
4
3.5
3
2.5
2
1.5
0
100
200
300
400


Aragonite oversaturation versus P(hydrostatic)
Oversaturation Aragonite
Pressure(bar)
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
0
100
200
300
400

Chapter 12. Electrochemistry

This chapter is introductive to chapter 13, where the application of the following concepts will be applied to the study of prevalence diagrams. Let us start with the electrode potential, E^0

The standard electrode potential E^0 (also called redox potential)of a semireaction , usually written in the reduction direction like

$$Zn^{2+}_{(aq)} + 2\ e^- = Zn_{(s)}$$

corresponds to the standard electrode potential (in volt) of an electrochemical cell where this semireaction is coupled with a standard hydrogen electrode.
The standard state corresponds to pure solids, unit standard concentrations, 25°C , 1.00 atm. pressure for gases.
On standard hydrogen electrode (or simply SHE) , according to the various semireactions coupled, both oxidation of molecular H_2 to solvated protons or the reduction of solvated protons to molecular hydrogen gas may happen. To this electrode a reference potential of 0.000 is always assigned, irrespectively of the reaction.

If the semireaction electrode has a negative standard potential (e.g. -0.763 volt for zinc electrode) the hydrogen electrode is positively charged and there a reduction reaction occurs :

$$2\ H^+_{(aq)} + 2\ e^- = H_{2(g)}$$

On the zinc electrode necessarily an oxidation reaction occurs,

$$Zn(s) = Zn^{2+}_{(aq)} + 2\ e^-$$

even if for convention the semireaction of zinc is usually written as a reduction.

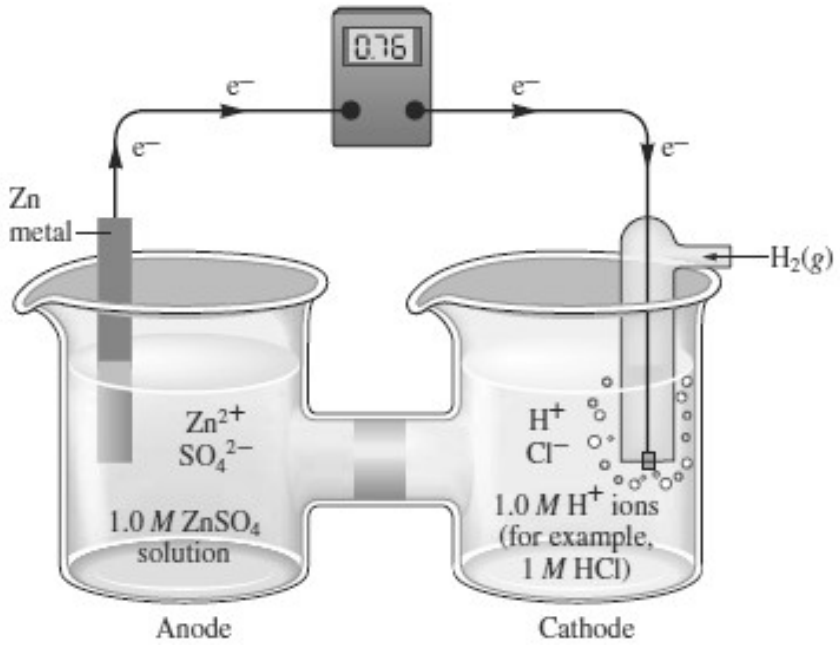


Fig.1

On the contrary, if the semireaction electrode has a positive standard potential, like

$$Cu^{2+}_{(aq)} + 2\ e^- = Cu_{(s)} \qquad E^0 = +0{,}337 \text{ volt}$$

when it is coupled with the hydrogen electrode, this becomes negatively charged and a reduction reaction takes place

$$H_{2(g)} = 2\,H^{+}_{(aq)} + 2\,e^{-}$$

When all species in a cell reaction are in their standard states (pure solids, unit standard concentrations, 25°C , 1,00 atm) the measured electromotive force (emf) of the cell (usually measured in volts) corresponds to the standard potential of the cell, E^0 (this is exactly true if we could measure the potential with zero intensity, but with modern voltmeters this is (nearly) the case).
Electrode potential is sometime called redox potential. It is usually referred to SHE, standard hydrogen electrode. A list of standard redox potential (25°C, p = 1 atm, concentrations = 1) for most common semi-reactions, always written as reductions (electrons on the left) is known as Electrochemical Series. Here is a small subset of the whole list of potentials, a full reference can be found elsewhere.

Reaction	Log *K* at 25°C	Standard Electrode Potential (V) at 25°C
$Na^+ + e^- = Na(s)$	−46	−2.71
$Mg^{2+} + 2e^- = Mg(s)$	−79.7	−2.35
$Zn^{2+} + 2e^- = Zn(s)$	−26	−0.76
$Fe^{2+} + 2e^- = Fe(s)$	−14.9	−0.44
$Co^{2+} + 2e^- = Co(s)$	−9.5	−0.28
$V^{3+} + e^- = V^{2+}$	−4.3	−0.26
$2H^+ + 2e^- = H_2(g)$	0.0	0.00
$S(s) + 2H^+ + 2e^- = H_2S$	+4.8	+0.14
$Cu^{2+} + e^- = Cu^+$	+2.7	+0.16
$AgCl(s) + e^- = Ag(s) + Cl^-$	+3.7	+0.22
$Cu^{2+} + 2e^- = Cu(s)$	+11.4	+0.34
$Cu^+ + e^- = Cu(s)$	+8.8	+0.52
$Fe^{3+} + e^- = Fe^{2+}$	+13.0	+0.77
$Ag^+ + e^- = Ag(s)$	+13.5	+0.80
$Fe(OH)_3(s) + 3H^+ + e^- = Fe^{2+} + 3H_2O$	+17.1	+1.01
$IO_3^- + 6H^+ + 5e^- = \frac{1}{2}I_2(s) + 3H_2O$	+104	+1.23
$MnO_2(s) + 4H^+ + 2e^- = Mn^{2+} + 2H_2O$	+43.6	+1.29
$Cl_2(g) + 2e^- = 2Cl^-$	+46	+1.36
$Co^{3+} + e^- = Co^{2+}$	+31	+1.82

Energetics of a galvanic cell

A galvanic cell is easily constructed by joining two semi-elements by an external electric wiring and a salt bridge.
Electrons flow in the conductor, whilst ionic neutrality is assured by proper ionic movement in the salt bridge. This can be also substituted by another mean that allows ions to flow, like a porous glass membrane, in the figure below.

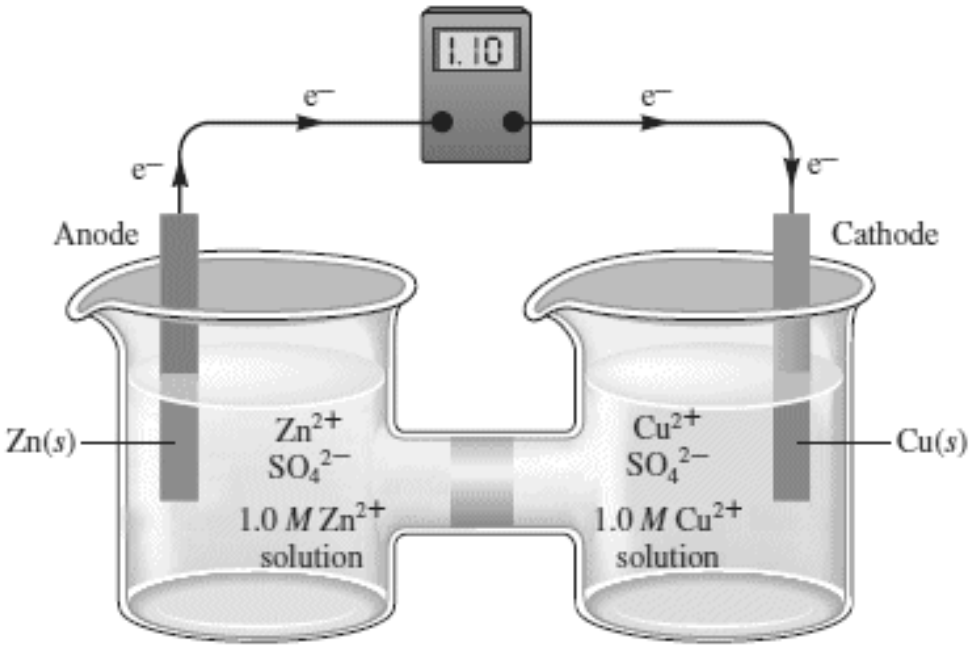


Fig.2

In the example above the two semi-reactions and the corresponding electrodes

(1) $Cu^{2+}_{(aq)} + 2\ e^- = Cu_{(s)}$ $E^0 = +0.337$ volt

(2) $Zn^{2+}_{(aq)} + 2\ e^- = Zn_{(s)}$ $E^0 = -0.763$ volt

are coupled together. Due to the fact that electrons are transferred only (neither created nor destroyed) they are emitted from anode (-) and collected through the wire by the catode (+). It follows that reaction occurs in opposite ways on the two electrodes, so that they can be added together, like the respective E^0 :

(anode) $Cu^{2+}_{(aq)} + 2\ e^- = Cu_{(s)}$ $E^0 = +0.337$ volt

(catode) $Zn_{(s)} = Zn^{2+}_{(aq)} + 2\ e^-$ $E^0 = +0.763$ volt

---

(overall) $Cu^{2+}_{(aq)} + Zn_{(s)} = Cu_{(s)} + Zn^{2+}_{(aq)}$ $E^0 = +1.100$ volt

A galvanic cell produces spontaneously an e.m.f (electromotive force, or difference of potential, in the above example 1.100 volt in standard conditions) . The system is very efficient, so that nearly the chemical energy is transormed in electricity. If efficiency is 100%, we know from themodynamcis that the electrical work produced by the cell ($E^0 \cdot n \cdot F$) equals the free energy variations (for the sign convention the sign has changed):

$$E^0 \cdot n \cdot F = -\Delta G^0 \quad \text{therefore} \quad E^0_H = -\frac{\Delta G^0}{n \cdot F}$$

We know from thermodynamics as well that $\Delta G^0 = -R \cdot T \cdot \ln(K_{eq})$ so that finally we obtain :

(1) $$E^0 = -\frac{\Delta G^0}{n \cdot F} = \frac{R \cdot T}{n \cdot F}\ln(K_{eq}) = \frac{2.303 \cdot R \cdot T}{n \cdot F}\log(K_{eq})$$

where :

n is the number of transferred electron in reaction (as written)

E_H refers to potential as measured relative to SHE (standard hydrogen electrode (taken as zero). In the following this will be written Eh for simplicity

K_{eq} is the product of all oxidized concentrations divided by the product of reduced ones. (at equilibrium).

In the copper/zinc battery above, we can write:

(2) $$K_{eq} = \frac{\{Cu\}\cdot\{Zn^{++}\}}{\{Cu^{++}\}\cdot\{Zn\}} = \frac{\{Zn^{++}\}}{\{Cu^{++}\}}$$ (3) $$K_{eq} = \frac{[Zn^{++}]}{[Cu^{++}]}$$

Equation (2) uses activities. The activities of solid substances are exactly 1 so that they can disappear. Equation (3) uses concentrations.
If we are at 25°C the different constant before log in eq(1) can be merged, thus obtaining:

$$E^0 = \frac{2.303 \cdot R \cdot T}{n \cdot F}\log(K_{eq}) = \frac{2.303 \cdot 8.314 \cdot 298.15}{n \cdot 96485}\log(K_{eq}) = \frac{0.0592}{n}\log(K_{eq})$$ (4

Eq (4) is valid for both semi-reaction and a complete redox system, we shall use mainly for half-reactions (*with free e^- on the left side*).

Example 1 : calculate the K_{eq} for the copper/zinc galvanic cell:
We have from eq.(1) Keq = exp(E^0·n·F/RT) = 10^(E^0·n/0.0592) = 10^(2.200/0.0592) = $1.45\cdot10^{37}$

When $[Zn^{++}]/[Cu^{++}] = 1.45\cdot10^{37}$ (nearly all copper has disappeared from solution) we have reached themodynamic equilibrium, the cell e.m.f is 0.000 volt exactly !

The Nernst equation

If we have reached the thermodynamical equilibrium, the galvanic cell is no more able to give off a potential difference.ΔG = 0 ; E = 0
What happens if we aren't at equilibrium ? Basically a e.m.f. raises at the electrodes. Let's start again from the basic relationship of thermodynamics $\Delta G^0 = - R\cdot T\cdot\ln(K_{eq})$ which becomes at non-equilibrium

$$\Delta G = \Delta G^0 + R\cdot T\cdot\ln(Q)$$

where Q is the product of all oxidized concentrations divided by the product of reduced ones. (at non-equilibrium). Therefore :

$$E_{eff} = -\Delta G /nF = (-\Delta G^0 - R\cdot T\cdot\ln(Q))/nF$$

so that we can write (see eq.(4) for comparison) :

$$E = -\frac{\Delta G^0}{n\cdot F} - \frac{2.303 \cdot R \cdot T}{n \cdot F}\log(Q) = E^0 - \frac{0.0592}{n}\log(Q)$$ (5)

Eq (5), is the famous Nernst equation, where :

$Q = \Pi_i$ {right side}/ Π_i {left side}

The symbol Π_i represent the product of the activities(or concentrations) of all substances, each raised by its stoichiometric coefficient.
Nernst equation is valid for complete redox system as well as for half-reactions (*with free e^- on the left side*). In this case

$Q = \Pi_i$ {reduced side}/ Π_i {oxidized side}

Example 2 (see fig.2, copper/zinc cell)
$Cu^{2+}_{(aq)} + Zn_{(s)} = Cu_{(s)} + Zn^{2+}_{(aq)}$ $E^0 = +1.100$ volt

(6) $E_{eff} = 1.100 - 0.0592/2 \cdot \log(\{Zn++\}/\{Cu++\})$ (Nernst equation for the whole cell)
(7) $E_{cathode} = +0.337 - 0.0592/2 \cdot \log\{Cu++\}$ (Nernst equation for the cathode)
(8) $E_{anode} = -0.763 - 0.0592/2 \cdot \log\{Zn++\}$ (Nernst equation for the anode)
It can be easily shown that eq(7) - eq(8) yields eq(9), so that $E_{eff} = E_{cathode} - E_{anode}$

Example 3 Let us calculate Eh of the following half-reaction :

$NO_3^-{}_{(aq)} + 4\ H^+_{(aq)} + 3\ e^- = NO_{(g)} + 2\ H_2O$

From the Nernst eq we have :

$$Eh = Eh^0 - \frac{0.0592}{4} \log \frac{P_{(NO)}}{\{NO3^-\}\{H^+\}^4}$$

if nitrogen pressure is unitary and we substitute concentrations for activities :

$$Eh = Eh^0 + 0.0148 \cdot \log([NO_3^-]\,[H^+]^4) = 0.92 + 0.0148 \cdot \log([NO_3^-]\,[H^+]^4) \text{ volt}$$

Chapter 13. Prevalence (or Pourbaix) diagrams for some common elements

Introduction

The aim of the following Octave scripts is to solve equilibria between chemical species with different oxidation states of the same element, as a function of pH and Eh**,** the redox potential of the solution relative to the Standard Hydrogen Electrode (SHE), defined by the Nernst equation. In this way, we obtain a **pH Eh** diagram, known as prevalence (or Pourbaix) diagram (M. Pourbaix, 1904-1998).

A **pH Eh** stability diagram shows in a comprehensive way how protons and electrons simultaneously shift the different equilibria under various conditions and may indicate which species predominates within a given range of pH and Eh

This diagram provides a key aid for the representation of equilibrium species in a certain solution, on the basis of the redox potentials which are available in the scientific literature. Indeed this diagrams plot the electrochemical stability for the different redox states of an element as a function of pH and Eh. It can be classified as a phase diagram, which maps the prevalence/stability regions, mainly in aqueous solutions.

The boundary lines in a Pourbaix diagram represent redox and acid-base reactions, and are the parts of the diagram where two species can exist in equilibrium. For example, in the Pourbaix diagram of Fe in fig.1, the red area on the top left represents the stability region of Fe^{+++}, which is only stable in a strong oxidizing and acid environment, the yellow area represents the stability region of Fe^{++} and the horizontal line between Fe^{+++} and Fe^{++} regions is described by the reaction, $Fe^{+++} + e^- \rightarrow Fe^{++}$ which has a standard potential of + 0.77 volt.

While we could use standard electrode potentials for all these lines (when all the concentrations of a soluble species are unitary), in practice, Pourbaix diagrams are plotted for lower ion concentrations (often 1 *mmol/L* or even lower), since they are more relevant for corrosion and electrochemical experiments. In the diagrams of this book, concentration has been chosen to be 0.01 *mol/L*, but this value can be adjusted by users in the scripts.

The first step in the diagram construction, for instance the Fe diagram, is to collect the relevant electrochemical potentials, which have been reported in Literature, elsewhere.

Fe++ + 2e- ---> Fe(s)	E° = - 0.4402 volt
Fe+++ + e- ---> Fe2+	E° = + 0.771 volt
Fe(OH)3 + 3 H+ + 3 e- ---> Fe(s) + 3 H2O	E° = + 0.059 volt
Fe(OH)2 + 2e- ---> Fe(s) + OH-	E° = - 0.877 volt
(FeOH)2+ + H+ + e- ---> Fe2+ + H2O	E° = + 0.914 volt

Usually the above potentials do not refer to a common oxidation state of the element, so it is necessary to transform them with reference to an oxidation state, which is equal to zero in the case of metallic elements.

For instance, the potential for *Fe*+++ is referred to that of *Fe*++ and not that of metallic iron, but but adding the reaction in row 1 to the reaction in row 2 (table below), we obtain the desired relation:

Fe++ + 2e- ---> Fe(s)	E° = - 0.4402 volt
Fe+++ + e- ---> Fe2+	E° = + 0.771 volt
Fe+++ + 3e- ---> Fe(s)	E° = (-0.4402•2 + 0.771)/3 = -0.0365 volt

The electrochemical potential are multiplied or divided by the number of transferred electrons. For the last half-reaction, the Nernst equation is written as:

$$Eh(Fe^{3+}/Fe) = -0.0365 - \frac{0.0592}{3}\log_{10}\frac{\{Fe\}}{\{Fe^3+\}} \quad .$$

Since, for solid Fe the activity is unitary, $\{Fe\}=1$, as for pure solid substances, we can replace activities with concentrations and rearrange as follows:

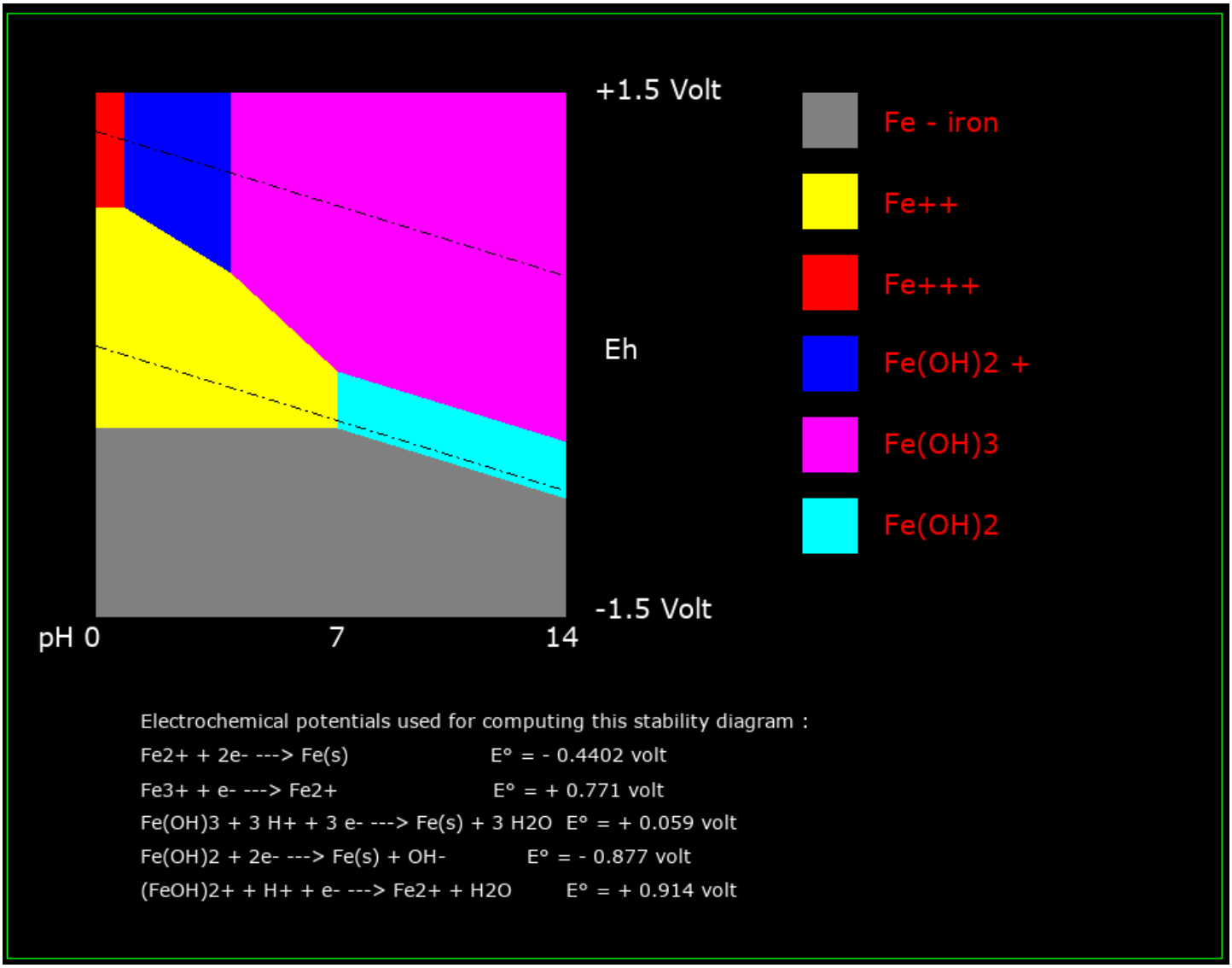
$$\log_{10} E_H(Fe^{3+}/Fe) = (E^0 + 0.0365)\frac{3}{0.0592} - \log_{10}[Fe^{3+}] = (E^0 + 0.0365)\frac{3}{0.0592} - \log_{10} c_1$$

where $c_1 = 0.01\,\text{mol/dm}^3$ denotes the chosen concentration of soluble species.

Repeating the procedure for every boundary line allows us to graphically plot the boundaries of every prevalence/stability domain, having a polygonal form, in a 2D diagram. The Matlab fill function, which requires to know the ordered set of the polygon vertices, is employed for coloring the different regions.

The Fe prevalence diagram

Once the concentration of *Fe* soluble species is fixed to the default value *0.01 mol/L* corresponding to *logConc = -2*, the script creates a blank background with the help of the matrix M. The matrix is then filled pixel by pixel by the color pertaining to the highest potential among the different forms of *Fe* compounds, which is detected by the Octave function max(x). Finally different comments and labels are added to the figure.



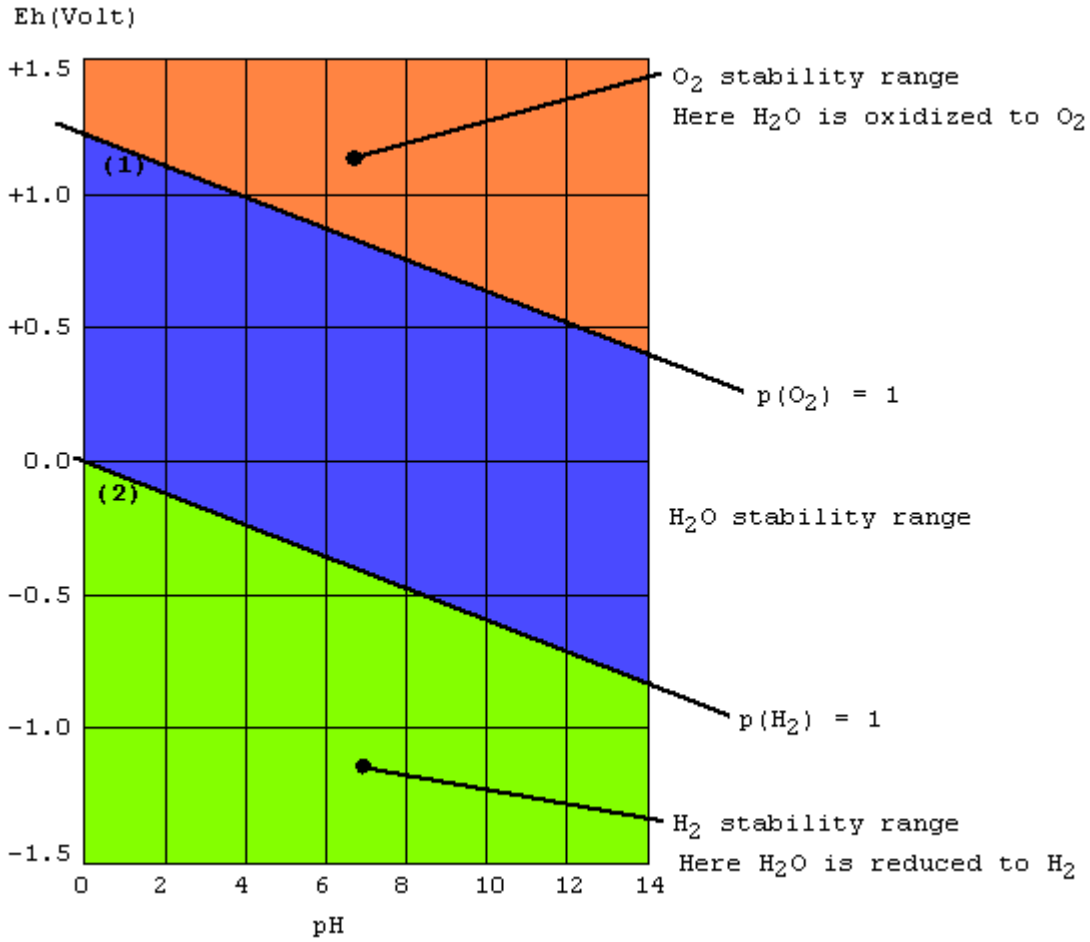

```
clear;clc;x = zeros (1,6);
logConc = -2; % [Fe ] in solution = 0.01 as reference
% prepare the M matrix
M = zeros(700,900,3,"uint8");
M(1:700,1,2) = 255;  % green rectangular frame
M(1:700,900,2) = 255;
M(1,1:900,2) = 255;
M(700,1:900,2) = 255;
% legend colors
M(60:100,590:630,1) = 128;M(60:100,590:630,2) = 128;M(60:100,590:630,3) = 128;
M(120:160,590:630,1) = 255;M(120:160,590:630,2) = 255;
M(180:220,590:630,1) = 255;
M(240:280,590:630,3) = 255;
M(300:340,590:630,1) = 255;M(300:340,590:630,3) = 255;
M(360:400,590:630,2) = 255;M(360:400,590:630,3) = 255;
% figure is prepared
```

```
figure(1,'position',[20,50,1161,860],'graphicssmoothing','off','resize','off','color',
[0,0,0]);
image(M);axis('off');
text(100,100,"please wait, I'm working it
out !",'color','r','fontname','verdana','fontsize',20);
% important to display the working figure
ak = kbhit(1);
% now the pixels are assigned to a color, accordin the chemical stability
for row = 60:447
  for column = 67:414
    pH = (column - 57)/24.786;
    Eh = (243.5 - row)/129;
    x(2) = (Eh + 0.4402)*2/0.0591 - logConc; % log Fe2+
    x(3) = (Eh + 0.0365)*3/0.0591 - logConc; % log Fe3+
    x(4) = (Eh - 0.0112)*3/0.0591 + 2*pH - logConc; % log Fe(OH)2+
    x(5) = (Eh - 0.059)*3/0.0591 + 3*pH;     % log Fe(OH)3
    x(6) = (Eh + 0.0496)*2/0.0591 + 2*pH;    % log Fe(OH)2
    [w,iw] = max(x);
    switch (iw)
      case 1;M(row,column,1:3) = [128,128,128];
      case 2;M(row,column,1:3) = [255,255,0];
      case 3;M(row,column,1:3) = [255,0,0];
      case 4;M(row,column,1:3) = [0,0,255];
      case 5;M(row,column,1:3) = [255,0,255];
      case 6;M(row,column,1:3) = [0,255,255];
    endswitch
  endfor
endfor
% the figure is displayed
image(M);axis('off');
% final text comments over the figure
A{1} = "Electrochemical potentials used for computing this stability diagram :";
A{2} = "Fe2+ + 2e- ---> Fe(s)                    E° = - 0.4402 volt";
A{3} = "Fe3+ + e- ---> Fe2+                      E° = + 0.771 volt";
A{4} = "Fe(OH)3 + 3 H+ + 3 e- ---> Fe(s) + 3 H2O  E° = + 0.059 volt";
A{5} = "Fe(OH)2 + 2e- ---> Fe(s) + OH-           E° = - 0.877 volt";
A{6} = "(FeOH)2+ + H+ + e- ---> Fe2+ + H2O        E° = + 0.914 volt";
for i = 1:6
  text(100,500 + 25*i,A{i},'color','w','fontname','verdana','fontsize',14);
```

```
endfor
A{1} = "Fe - iron";A{2} = "Fe++";;A{3} = "Fe+++";A{4} = "Fe(OH)2 +";A{5} =
"Fe(OH)3";A{6} = "Fe(OH)2"
for i = 1:6
  text(650,20 + 60*i,A{i},'color','r','fontname','verdana','fontsize',20);
endfor
text(24,464,"pH 0                          7
14",'color','w','fontname','verdana','fontsize',20);
A{1} = "+1.5 Volt";A{2} = " Eh";A{3} = "-1.5 Volt";
for i = 1:3
  text(436,192*i - 136,A{i},'color','w','fontname','verdana','fontsize',20);
endfor


line([67 414],[247 354],'color','k','linewidth',1,'linestyle','-.');
line([67 414],[88 195],'color','k','linewidth',1,'linestyle','-.');
```

The fundamental prevalence diagram: water

Natural waters are often in a highly dynamic state with regard to oxidation- reduction rather than in or near equilibrium. Most oxidation-reduction reactions have a tendency to be much slower than acid-base reactions, especially in the absence of suitable bio- chemical catalysis. Nonetheless, equilibrium diagrams can greatly aid attempts to understand the possible redox pattems in natural waters and in water technological systems.



The construction of Eh—pH diagrams may be introduced by considering the redox stability of water. H_2O can be oxidized to O_2 or reduced to H_2.

The pertinent redox equations can be written in different equivalent forms:

for reduction

1) $2\ H^+ + 2e^- = H_{2(g)}$ $\quad E^0 = 0.000$ volt (reference electrode)

2) $2\ H_2O + 2\ e^- = H_{2(g)} + 2\ OH^-$ $\quad E^0 = -0.828$ volt

for oxidation

3) $O_{2(g)} + 4\ H^+ + 4\ e^- = 2\ H_2O$ $\quad E^0 = +1.229$ volt

4) $O_{2(g)} + 2\ H_2O + 4\ e^- = 4\ OH^-$ $\quad E^0 = +0.401$ volt

Eq. (1) and (2) are equivalent , as are eq. (3) and (4)

Eq. (2) can be obtained from (1) by adding the formal equilibrium of water dissociation

$$2\ H_2O = 2\ OH^- + 2\ H^+$$

and eq (4) can be obtained from (3) by adding twice the same

$$4\ H_2O = 4\ OH^- + 4\ H^+$$

The difference in E^0 between eq.(1)-(2) and eq.(3)-(4) is readily explained by the Nernst equation taking into account the ionic product of water and hence that

$[OH^-] = 1 \cdot 10^{-14} / [H^+]$.

For example if $[H^+] = 1.000$ (standard conditions) then $[OH^-] = 1 \cdot 10^{-14}$ mol/L

If gas pressure (H_2 / O_2) are unitary, we can say from eq.(1) that

$$5) \quad Eh_{(1)} = E^0 - \frac{0.0592}{2} \cdot \log\left(\frac{1 \cdot 10^{-14}}{[H^+]}\right)^2 = 0.000 + 0.0592 \cdot \log[H^+] = -0.0592 \cdot pH$$

Therefore the standard E^0 of eq.(2), when $[OH^-] = 1$ and therefore $[H^+] = 1 \cdot 10^{-14}$ and pH = 14 will be : $Eh_{(2)} = -0.0592 \cdot pH = -0.828$ volt

If gas pressure (H_2 / O_2) are unitary, we can say from eq.(3) that

$$6) \quad Eh_{(3)} = E^0 - \frac{0.0592}{4} \cdot \log\left(\frac{1 \cdot 10^{-14}}{[H^+]}\right)^4 = 1.229 + 0.0592 \cdot \log[H^+] = 1.229 - 0.0592 \cdot pH$$

Therefore the standard E^0 of eq.(4), when $[OH^-] = 1$ and therefore $[H^+] = 1 \cdot 10^{-14}$ and pH = 14 will be : $Eh_{(4)} = 1.229 - 0.0592 \cdot pH = 1.229 - 0.828 = +0.401$ volt

The stability fields of our main substances (liquid H_2O, gaseous H_2 and O_2) are are separated by boundary lines (just like the borders of a country), when we extablish the boundaries, the stability fields are determined by consequence.

When O_2 (P=1atm.) will be able to oxidize water and when not ? . The boundary lines in the Eh-pH diagrams are usually made by the points satisfying the thermodynamic equilibrium among the chemical species, for oxygen, hydrogen and water :

1) $2\ H^+ + 2e^- = H_{2(g)}$

3) $O_{2(g)} + 4\ H^+ + 4\ e^- = 2\ H_2O$

These equilibria are satisfied by the Nernst equation, in particular equations (5) and (6) allow to divide by straight lines the **Eh-pH** diagram; on these lines every point satisfies the condition of thermodynamic equilibrium for each pH value between 0 and 14. This means $p(H_2) = 1$ atm or p(O2) = 1 atm. Therefore the points in the diagram lying below line (5) correspond to an equilibrium of H_2/H_2O with a $p(H_2)>1$ atm, so hydrogen is the predominant phase (green area in fig.1). Above the upper line, water becomes an effective reductant (producing oxygen).

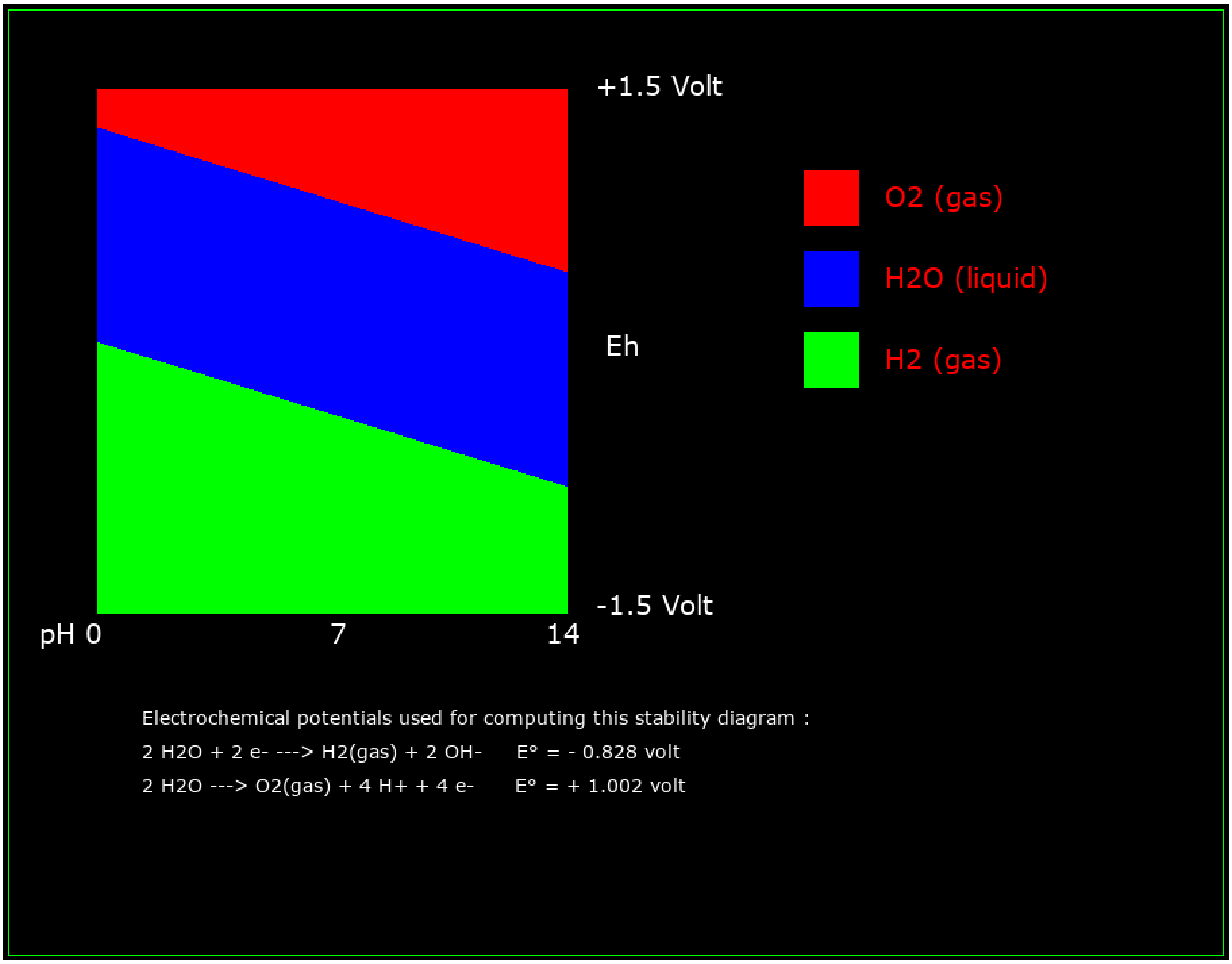
The points in the diagram lying above line (6) correspond to an equilibrium of O_2/H_2O with a $p(O_2)>1$ atm, so oxygen is the predominant phase (brown area in fig.1). Below this lower line, water is an effective oxidant (producing hydrogen).

Within the H_2O domain, O_2 acts as an oxidant and H_2 as a reductant. The lines of both equations have slopes -0.0592, and they intersect the ordinate, at pH = 0 (Eh = 1.229 and Eh = 0.000 V respectively) and pH = 14 (Eh = 0.401 and Eh = -0.828 V respectively)

```
clear all;clc;x = zeros (1,3);
```

```
% a 700 rows x 900 columns array is prepared, with a green frame
M = zeros(700,900,3,"uint8");
M(1:700,1,2) = 255;
M(1:700,900,2) = 255;
M(1,1:900,2) = 255;
M(700,1:900,2) = 255;
% legend colors
M(120:160,590:630,1) = 255;
M(180:220,590:630,3) = 255;
M(240:280,590:630,2) = 255;
% figure is prepared
figure(1,'position',[20,50,1161,860],'graphicssmoothing','off','resize','off','color',
[0,0,0]);
image(M);axis('off');
text(100,100,"please            wait,            I'm            working            it
out !",'color','r','fontname','verdana','fontsize',20);
ak = kbhit(1);% important to display the working figure


% now the pixels are assigned to a color, according to the chemical stability
for row = 60:447
  Eh = (243.5 - row)/129;
  for column = 67:414
    pH = (column - 57)/24.786;
    x(2) = - Eh*2/0.0591 - 2*pH;         % log H2
    x(3) = (Eh - 1.229)*4/0.0591 + 4*pH; % log O2
    [w,iw] = max(x);
    switch (iw)
      case 1;M(row,column,1:3) = [0,0,255];  %  blue  H2O
      case 2;M(row,column,1:3) = [0,255,0];  %  green  H2
      case 3;M(row,column,1:3) = [255,0,0];  %  red  O2
    endswitch
  endfor
endfor
% the figure is displayed
image(M);axis('off');hold on;
% final text comments over the figure
A{1} = "Electrochemical potentials used for computing this stability diagram :";
A{2} = "2 H2O + 2 e- ---> H2(gas) + 2 OH-     E° = - 0.828 volt";
A{3} = "2 H2O ---> O2(gas) + 4 H+ + 4 e-      E° = + 1.002 volt";
```

```
for i = 1:3
  text(100,500 + 25*i,A{i},'color','w','fontname','verdana','fontsize',14);
endfor
A{1} = "O2 (gas)";A{2} = "H2O (liquid)";;A{3} = "H2 (gas)";
for i = 1:3
  text(650,80 + 60*i,A{i},'color','r','fontname','verdana','fontsize',20);
endfor
text(24,464,"pH   0                                                    7
14",'color','w','fontname','verdana','fontsize',20);
A{1} = "+1.5 Volt";A{2} = " Eh";A{3} = "-1.5 Volt";
for i = 1:3
  text(436,192*i - 136,A{i},'color','w','fontname','verdana','fontsize',20);
endfor
```



Manganese prevalence diagram

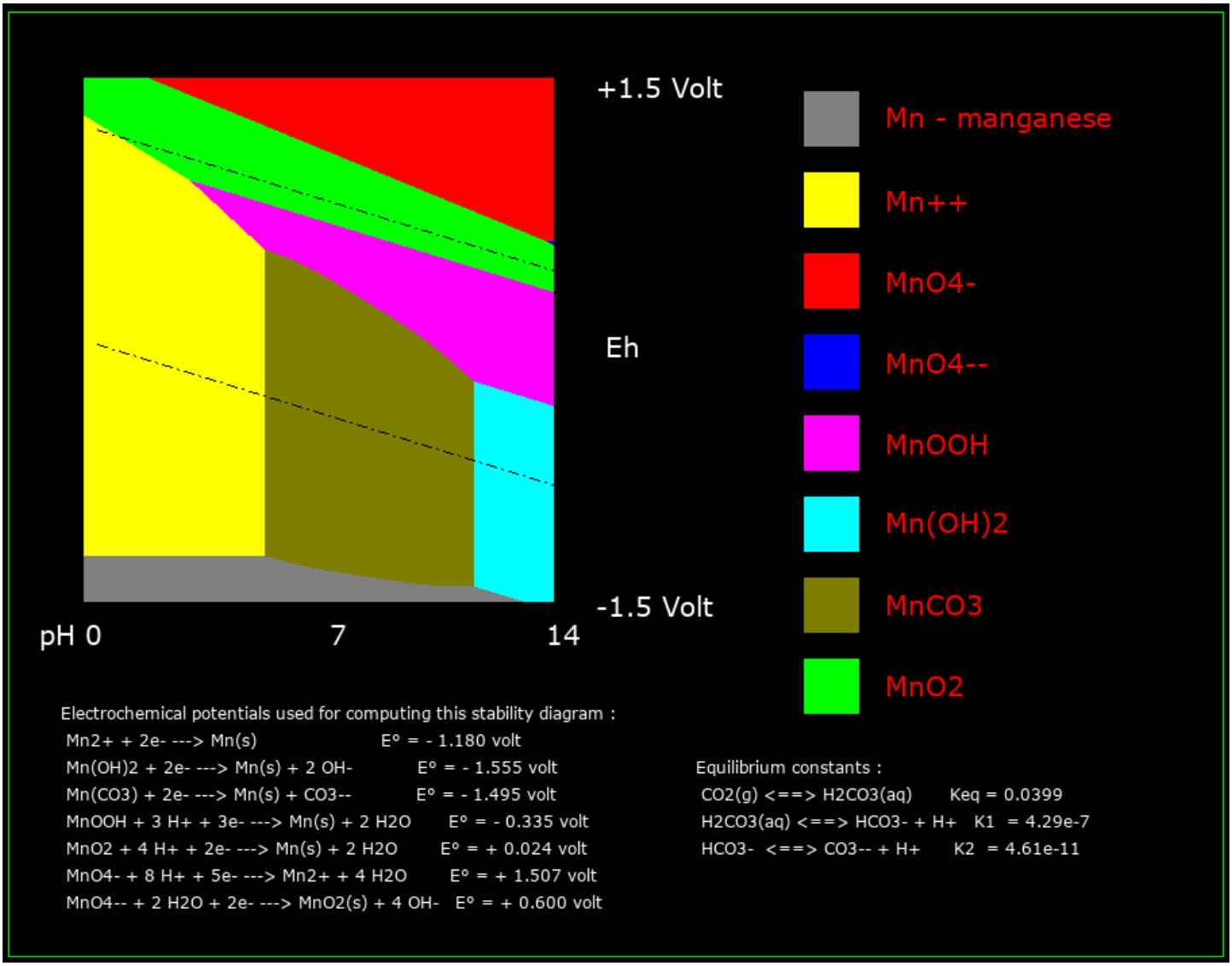
Manganese oxides and hydroxides cover a wide range of oxidation states, ranging from +2 to +7. The reactions to be considered are as follows.

```
Mn2+ + 2e- ---> Mn(s)                          E° = - 1.180 volt
Mn(OH)2 + 2e- ---> Mn(s) + 2 OH-               E° = - 1.555 volt
Mn(CO3) + 2e- ---> Mn(s) + CO3--               E° = - 1.495 volt
MnOOH + 3 H+ + 3e- ---> Mn(s) + 2 H2O          E° = - 0.335 volt
MnO2 + 4 H+ + 2e- ---> Mn(s) + 2 H2O           E° = + 0.024 volt
MnO4- + 8 H+ + 5e- ---> Mn2+ + 4 H2O           E° = + 1.507 volt
MnO4-- + 2 H2O + 2e- ---> MnO2(s) + 4 OH-      E° = + 0.600 volt
 Equilibrium constants :";
CO2(g) <==> H2CO3(aq)          Keq = 0.0399
H2CO3(aq) <==> HCO3- + H+      K1  = 4.29e-7
HCO3-  <==> CO3-- + H+         K2  = 4.61e-11
```

```
clear;clc;x = zeros (1,6);
logConc = -2; % [Mn ] in solution = 0.01 as reference
Ctot = 0.002; % Ctot = H2CO3 + HCO3- + CO3--  total of carbonate species
K1 = 4.2e-7; K2 = 4.8e-11; % first and second dissociation constants for H2CO3
% prepare the figure
M = zeros(700,900,3,"uint8");
M(1:700,1,2) = 255;  % green border
M(1:700,900,2) = 255;
M(1,1:900,2) = 255;
M(700,1:900,2) = 255;
% legend colors
M(60:100,590:630,1) = 128;M(60:100,590:630,2) = 128;M(60:100,590:630,3) = 128;
M(120:160,590:630,1) = 255;M(120:160,590:630,2) = 255;
M(180:220,590:630,1) = 255;
M(240:280,590:630,3) = 255;
M(300:340,590:630,1) = 255;M(300:340,590:630,3) = 255;
M(360:400,590:630,2) = 255;M(360:400,590:630,3) = 255;
M(420:460,590:630,1) = 125;M(420:460,590:630,2) = 125;
M(480:520,590:630,2) = 255;
% figure is prepared
```

```
figure(1,'position',[20,50,1161,860],'graphicssmoothing','off','resize','off','color',
[0,0,0]);

image(M);axis('off');

text(100,100,"please            wait,            I'm            working            it
out !",'color','r','fontname','verdana','fontsize',20);

% important to display the working figure

ak = kbhit(1);

% now the pixels are assigned to a color, accordin the chemical stability

for col = 57:404

  pH = (col - 57)/24.786;

  H = 10^(-pH);

  CO3 = Ctot*K1*K2/(K1*K2 + K1*H + H*H); logCO3 =  log10(CO3);

  for row = 50:437

    Eh = (243.5 - row)/129;

    x(2) = (Eh + 1.180)*2/0.0591 - logConc; % log Mn2+

    x(3) = (Eh - 0.739)*7/0.0591 + 8*pH - logConc;  % log MnO4-

    x(4) = (Eh - 0.768)*6/0.0591 + 8*pH - logConc;  % log MnO4--

    x(5) = (Eh + 0.335)*3/0.0591 + 3*pH;    % log MnOOH

    x(6) = (Eh + 0.727)*2/0.0591 + 2*pH;    % log Mn(OH)2

    x(7) = (Eh + 1.495)*2/0.0591 + logCO3;  % log MnCO3

    x(8) = (Eh - 0.024)*4/0.0591 + 4*pH;    % log MnO2

    [w,iw] = max(x);

    switch (iw)

      case 1;M(row,col,1:3) = [128,128,128];  %  gray     Mn(metallic)

      case 2;M(row,col,1:3) = [255,255,0];    %  yellow   Mn++

      case 3;M(row,col,1:3) = [255,0,0];      %  red      MnO4-

      case 4;M(row,col,1:3) = [0,0,255];      %  blue     MnO4--

      case 5;M(row,col,1:3) = [255,0,255];    %  magenta  MnOOH

      case 6;M(row,col,1:3) = [0,255,255];    %  light blue  Mn(OH)2

      case 7;M(row,col,1:3) = [125,125,0];    %  brown    MnCO3

      case 8;M(row,col,1:3) = [0,255,0];      %  green    MnO2

      endswitch

  endfor

endfor

% the figure is displayed

image(M);axis('off');


% final text comments over the figure

A{1} = "Electrochemical potentials used for computing this stability diagram :";
```

```
A{2} = " Mn2+ + 2e- ---> Mn(s)                          E° = - 1.180 volt";
A{3} = " Mn(OH)2 + 2e- ---> Mn(s) + 2 OH-               E° = - 1.555 volt";
A{4} = " Mn(CO3) + 2e- ---> Mn(s) + CO3--               E° = - 1.495 volt";
A{5} = " MnOOH + 3 H+ + 3e- ---> Mn(s) + 2 H2O          E° = - 0.335 volt";
A{6} = " MnO2 + 4 H+ + 2e- ---> Mn(s) + 2 H2O           E° = + 0.024 volt";
A{7} = " MnO4- + 8 H+ + 5e- ---> Mn2+ + 4 H2O           E° = + 1.507 volt";
A{8} = " MnO4-- + 2 H2O + 2e- ---> MnO2(s) + 4 OH-      E° = + 0.600 volt";
A{9} = "Equilibrium constants :";
A{10} = " CO2(g) <==> H2CO3(aq)          Keq = 0.0399";
A{11} = " H2CO3(aq) <==> HCO3- + H+      K1  = 4.29e-7";
A{12} = " HCO3-  <==> CO3-- + H+         K2  = 4.61e-11";


for i = 1:12
                  text(40         +         470*(i>8),500         +         20*i         -
120*(i>8),A{i},'color','w','fontname','verdana','fontsize',12);
endfor
A{1} = "Mn - manganese";A{2} = "Mn++";;A{3} = "MnO4-";A{4} = "MnO4--";A{5} =
"MnOOH";A{6} = "Mn(OH)2";...
A{7} = "MnCO3";A{8} = "MnO2";
for i = 1:8
  text(650,20 + 60*i,A{i},'color','r','fontname','verdana','fontsize',20);
endfor
text(24,464,"pH   0                                                                    7
14",'color','w','fontname','verdana','fontsize',20);
A{1} = "+1.5 Volt";A{2} = " Eh";A{3} = "-1.5 Volt";
for i = 1:3
  text(436,192*i - 136,A{i},'color','w','fontname','verdana','fontsize',20);
endfor


line([67 414],[247 354],'color','k','linewidth',1,'linestyle','-.');
line([67 414],[88 195],'color','k','linewidth',1,'linestyle','-.');
```



Lead, lead sulphate and carbonate

Lead oxides, sulphates and chlorides are usual found among corrosion products of lead manufacts, like pipes, ingots, vessels and so on, when exposed to natural waters. The stability/prevalence diagram is calculated in presence of atmospheric CO2 so as $PbCO_3$ (lead carbonate) formation is considered, together with $PBSO_4$.

The reactions to be considered are as follows.

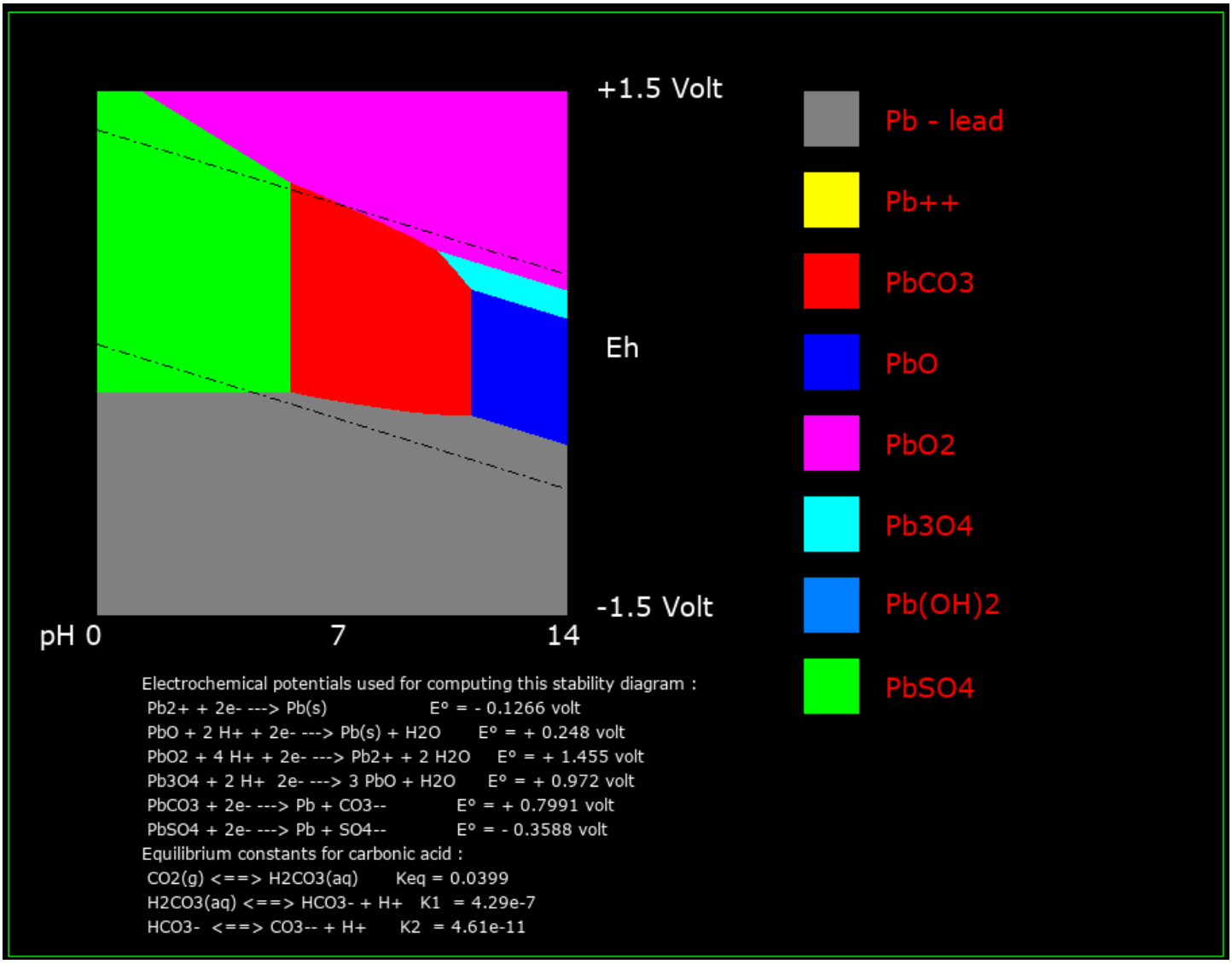
```
Pb2+ + 2e- ---> Pb(s)                    E° = - 0.1266 volt
PbO + 2 H+ + 2e- ---> Pb(s) + H2O        E° = + 0.248 volt
PbO2 + 4 H+ + 2e- ---> Pb2+ + 2 H2O      E° = + 1.455 volt
Pb3O4 + 2 H+  2e- ---> 3 PbO + H2O       E° = + 0.972 volt
PbCO3 + 2e- ---> Pb + CO3--              E° = + 0.7991 volt
PbSO4 + 2e- ---> Pb + SO4--              E° = - 0.3588 volt
Equilibrium constants for carbonic acid :
```

```
CO2(g) <==> H2CO3(aq)        Keq = 0.0399
H2CO3(aq) <==> HCO3- + H+    K1  = 4.29e-7
HCO3-  <==> CO3-- + H+       K2  = 4.61e-11
```

```
clear all;clc;x = zeros (1,6);
logConc = -2; logCl = -2;logSO4 = -2;Ctot = 0.002;
K1 = 4.2e-7; K2 = 4.8e-11; % first and second dissociation constants for H2CO3
% prepare the M matrix
M = zeros(700,900,3,"uint8");
M(1:700,1,2) = 255;
M(1:700,900,2) = 255;
M(1,1:900,2) = 255;
M(700,1:900,2) = 255;
% legend colors
M(60:100,590:630,1) = 128;M(60:100,590:630,2) = 128;M(60:100,590:630,3) = 128;
M(120:160,590:630,1) = 255;M(120:160,590:630,2) = 255;
M(180:220,590:630,1) = 255;
M(240:280,590:630,3) = 255;
M(300:340,590:630,1) = 255;M(300:340,590:630,3) = 255;
M(360:400,590:630,2) = 255;M(360:400,590:630,3) = 255;
M(420:460,590:630,2) = 128;M(420:460,590:630,3) = 255;
M(480:520,590:630,2) = 255;

% figure is prepared
figure(1,'position',[20,50,1161,860],'graphicssmoothing','off','resize','off','color',
[0,0,0]);
image(M);axis('off');
text(100,100,"please            wait,            I'm            working            it
out !",'color','r','fontname','verdana','fontsize',20);
% important to display the working figure
ak = kbhit(1);
% now the pixels are assigned to a color, accordin the chemical stability
for row = 60:447
  for column = 67:414
    pH = (column - 57)/24.786;
    H = 10^(-pH);
    CO3 = Ctot*K1*K2/(K1*K2 + K1*H + H*H); logCO3 =  log10(CO3);
```

```
    Eh = (243.5 - row)/129;
    x(2) = (Eh + 0.1266)*2/0.0591 - logConc;   % log Pb2+
    x(3) = (Eh + 0.515)*2/0.0591 + logCO3;     % log PbCO3
    x(4) = (Eh - 0.248)*2/0.0591 + 2*pH ;      % log PbO
    x(5) = (Eh - 0.6642)*4/0.0591 + 4*pH;      % log PbO2
    x(6) = (Eh - 0.429)*8/3/0.0591  + 8/3*pH;  % log Pb3O4
    x(7) = (Eh - 0.277)*2/0.0591 + 2*pH;       % log Pb(OH)2
    x(8) = (Eh + 0.3588)*2/0.0591 + logSO4;    % log PbSO4
    [w,iw] = max(x);
    switch (iw)
      case 1;M(row,column,1:3) = [128,128,128];
      case 2;M(row,column,1:3) = [255,255,0];
      case 3;M(row,column,1:3) = [255,0,0];
      case 4;M(row,column,1:3) = [0,0,255];
      case 5;M(row,column,1:3) = [255,0,255];
      case 6;M(row,column,1:3) = [0,255,255];
      case 7;M(row,column,1:3) = [0,128,255];
      case 8;M(row,column,1:3) = [0,255,0];
    endswitch
  endfor
endfor
% the figure is displayed
image(M);axis('off');
% final text comments over the figure

A{1} = "Electrochemical potentials used for computing this stability diagram :";
A{2} = " Pb2+ + 2e- ---> Pb(s)                  E° = - 0.1266 volt";
A{3} = " PbO + 2 H+ + 2e- ---> Pb(s) + H2O      E° = + 0.248 volt";
A{4} = " PbO2 + 4 H+ + 2e- ---> Pb2+ + 2 H2O    E° = + 1.455 volt";
A{5} = " Pb3O4 + 2 H+  2e- ---> 3 PbO + H2O     E° = + 0.972 volt";
A{6} = " PbCO3 + 2e- ---> Pb + CO3--            E° = + 0.7991 volt";
A{7} = " PbSO4 + 2e- ---> Pb + SO4--            E° = - 0.3588 volt";
A{8} = "Equilibrium constants for carbonic acid : ";
A{9} = " CO2(g) <==> H2CO3(aq)       Keq = 0.0399";
A{10} =" H2CO3(aq) <==> HCO3- + H+   K1  = 4.29e-7";
A{11} =" HCO3-  <==> CO3-- + H+      K2  = 4.61e-11";

for i = 1:11
```

```
  text(100,480 + 18*i,A{i},'color','w','fontname','verdana','fontsize',12);
endfor
A{1} = "Pb - lead";A{2} = "Pb++";;A{3} = "PbCO3";A{4} = "PbO";A{5} = "PbO2";A{6} =
"Pb3O4";A{7} = "Pb(OH)2";A{8} = "PbSO4";
for i = 1:8
  text(650,20 + 60*i,A{i},'color','r','fontname','verdana','fontsize',20);
endfor
text(24,464,"pH   0                                                                    7
14",'color','w','fontname','verdana','fontsize',20);
A{1} = "+1.5 Volt";A{2} = " Eh";A{3} = "-1.5 Volt";
for i = 1:3
  text(436,192*i - 136,A{i},'color','w','fontname','verdana','fontsize',20);
endfor

line([67 414],[247 354],'color','k','linewidth',1,'linestyle','-.');
line([67 414],[88 195],'color','k','linewidth',1,'linestyle','-.');
```



Sulphur prevalence diagram

This stability diagram includes the stable oxidation states of sulphur, namely 0, -2, +6 , which occur as elemental sulphur (nox=0) ,suphidric acid (nox = -2), a weak biprotic acid, and sulphuric acid (nox=+6) a strong acid for the first dissociation only.

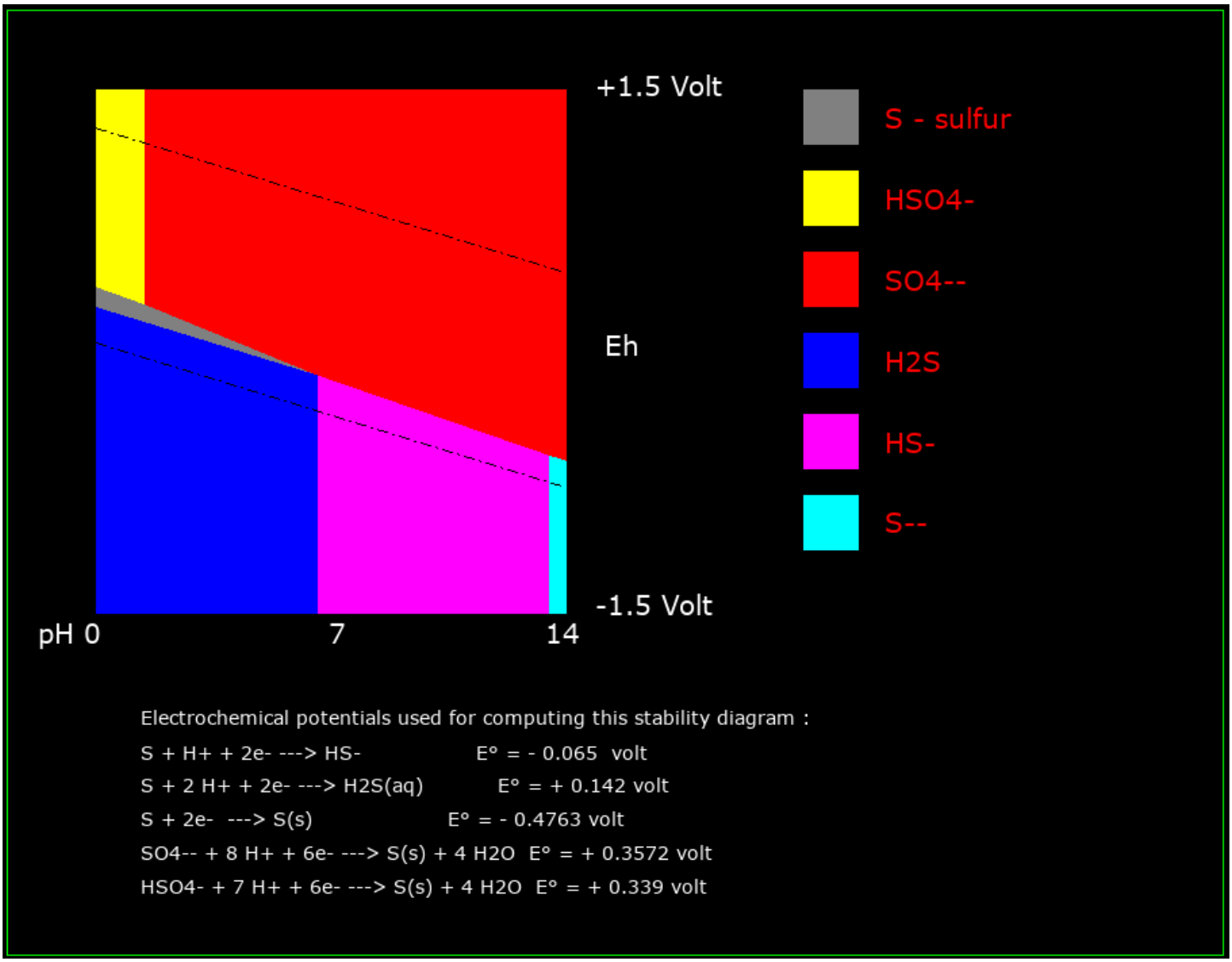
The reactions to be considered are as follows.

```
S + H+ + 2e- ---> HS-                    E° = - 0.065  volt";
S + 2 H+ + 2e- ---> H2S(aq)              E° = + 0.142 volt";
S + 2e-  ---> S(s)                       E° = - 0.4763 volt";
SO4-- + 8 H+ + 6e- ---> S(s) + 4 H2O  E° = + 0.3572 volt";
HSO4- + 7 H+ + 6e- ---> S(s) + 4 H2O  E° = + 0.339 volt";
```

```
clear;clc;x = zeros (1,6);
logConc = -2; % log S(solution)  = -2
% prepare the matrix M
M = zeros(700,900,3,"uint8");
M(1:700,1,2) = 255;
M(1:700,900,2) = 255;
M(1,1:900,2) = 255;
M(700,1:900,2) = 255;
% legend colors
M(60:100,590:630,1) = 128;M(60:100,590:630,2) = 128;M(60:100,590:630,3) = 128;
M(120:160,590:630,1) = 255;M(120:160,590:630,2) = 255;
M(180:220,590:630,1) = 255;
M(240:280,590:630,3) = 255;
M(300:340,590:630,1) = 255;M(300:340,590:630,3) = 255;
M(360:400,590:630,2) = 255;M(360:400,590:630,3) = 255;
% figure is prepared
figure(1,'position',[20,50,1161,860],'graphicssmoothing','off','resize','off','color',
[0,0,0]);
image(M);axis('off');
text(100,100,"please wait, I'm working it
out !",'color','r','fontname','verdana','fontsize',20);
% important to display the working figure
ak = kbhit(1);
```

```
% now the pixels are assigned to a color, accordin the chemical stability
for row = 60:447
  Eh = (243.5 - row)/129;
  for column = 67:414
    pH = (column - 57)/24.786;
    x(2) = (Eh - 0.339)*6/0.0591 + 7*pH - logConc;  % log HSO4-
    x(3) = (Eh - 0.3572)*6/0.0591 + 8*pH - logConc; % log SO4--
    x(4) = (0.142 - Eh)*2/0.0591 - 2*pH - logConc;  % log H2S
    x(5) = (-0.065 - Eh)*2/0.0591 - pH - logConc;   % log HS-
    x(6) = (-0.4763 - Eh)*2/0.0591 - logConc;       % log S--
    [w,iw] = max(x);
    switch (iw)
      case 1;M(row,column,1:3) = [128,128,128];
      case 2;M(row,column,1:3) = [255,255,0];
      case 3;M(row,column,1:3) = [255,0,0];
      case 4;M(row,column,1:3) = [0,0,255];
      case 5;M(row,column,1:3) = [255,0,255];
      case 6;M(row,column,1:3) = [0,255,255];
    endswitch
  endfor
endfor
% the figure is displayed
image(M);axis('off');
% final text comments over the figure
A{1} = "Electrochemical potentials used for computing this stability diagram :";
A{2} = "S + H+ + 2e- ---> HS-                E° = - 0.065  volt";
A{3} = "S + 2 H+ + 2e- ---> H2S(aq)          E° = + 0.142 volt";
A{4} = "S + 2e-  ---> S(s)                   E° = - 0.4763 volt";
A{5} = "SO4-- + 8 H+ + 6e- ---> S(s) + 4 H2O  E° = + 0.3572 volt";
A{6} = "HSO4- + 7 H+ + 6e- ---> S(s) + 4 H2O  E° = + 0.339 volt";
for i = 1:6
  text(100,500 + 25*i,A{i},'color','w','fontname','verdana','fontsize',14);
endfor
A{1} = "S - sulfur";A{2} = "HSO4-";;A{3} = "SO4--";A{4} = "H2S";A{5} = "HS-";A{6} =
"S--"
for i = 1:6
  text(650,20 + 60*i,A{i},'color','r','fontname','verdana','fontsize',20);
endfor
```

```
text(24,464,"pH   0                                                                    7
14",'color','w','fontname','verdana','fontsize',20);
A{1} = "+1.5 Volt";A{2} = " Eh";A{3} = "-1.5 Volt";
for i = 1:3
  text(436,192*i - 136,A{i},'color','w','fontname','verdana','fontsize',20);
endfor

line([67 414],[247 354],'color','k','linewidth',1,'linestyle','-.');
line([67 414],[88 195],'color','k','linewidth',1,'linestyle','-.');
```



Au/Cl in seawater

The Earth's oceans contain tiny concentrations of gold, about $10^{-30}\,g/km^3$. At this very low value the Earth's oceans would nevertheless hold 15000of gold [5].

In the past, a few people claimed to be able of economically recovering gold from sea water, but they were either mistaken or acted in an intentional deception. F. Haber (1868-1934) did research on the gold extraction from sea water in an effort to help Germany paying the World War I reparations [6]. Actually, his belief into a possible commercial extraction was founded on a largely overestimated analysis of seawater, and his efforts became unsuccessful.

In seawater, gold may be present as finely dispersed colloidal particles of metal and auric trihydroxide, $Au(OH)_3$ Gold may be also present in ionic form like chloroauric,,$AuCl_4^-$ and auric ions, Au+++. The prevalence diagram in provides a detailed picture of the abundance of these species relative to the approximate concentration [Cl-] = 0.6 mol/L of the chlorine ion in in seawater and to a total gold concentration of 0.01 mmol/L.

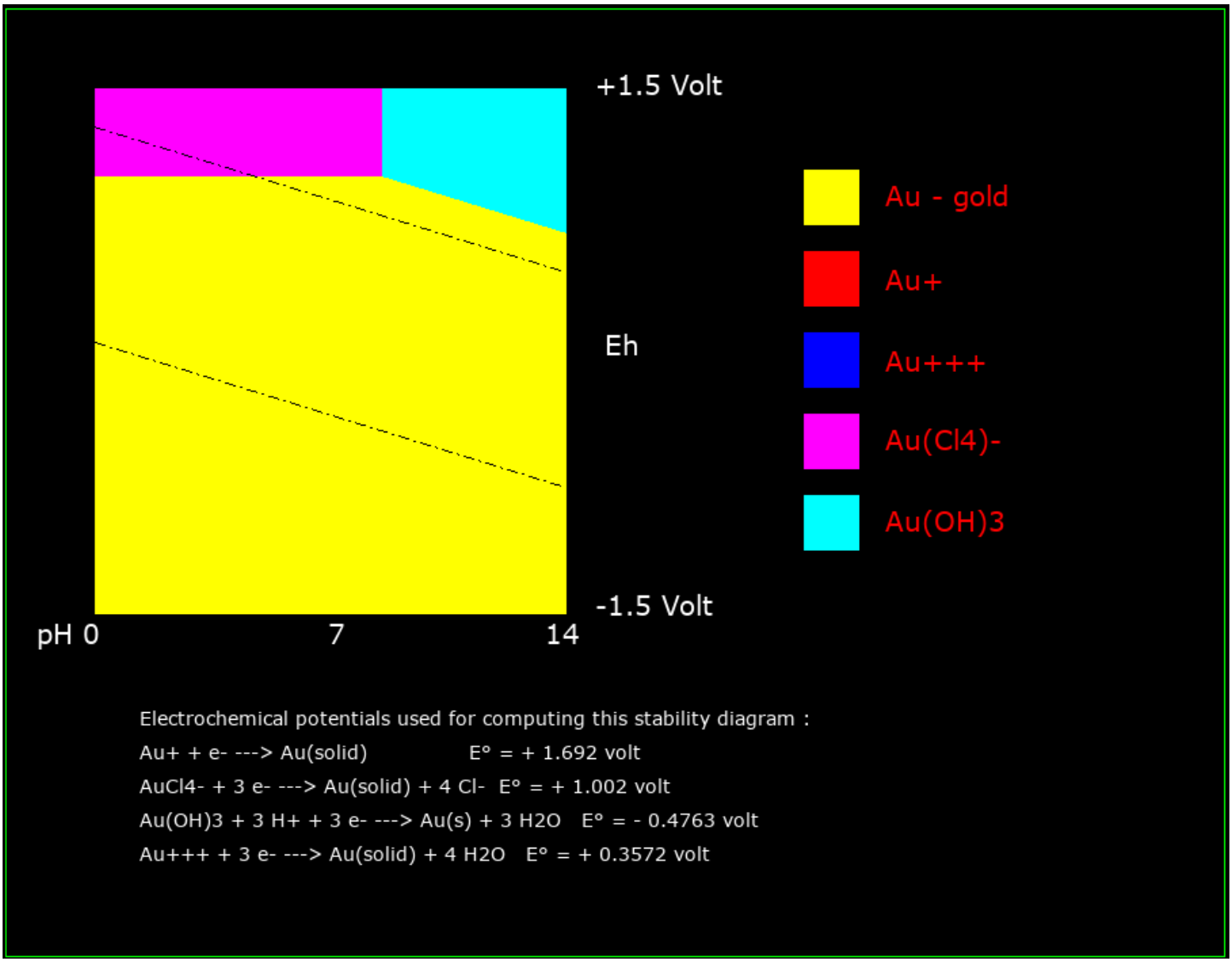
Reactions and electric potentials to be considered in order to build the stability diagram are the following ones.

Au+ + e- ---> Au(s)	E° = + 1.692 volt
AuCl4- + 3 e- ---> Au(s) + 4 Cl-	E° = + 1.002 volt
Au(OH)3 + 3H+ + 3e- ---> Au + 3 H2O	E° = + 1.45 volt
Au3+ + 3e- ---→ Au(s)	E° = + 1.49 volt

```
clear all;clc;x = zeros (1,5);  % log Au  = 0 as reference
logConc = -5;       % concentration of Au soluble species = 0.00001 M
logCl = log10(0.6); % concentration of Cl- in seawater = 0.6 M
% prepare the matrix M
M = zeros(700,900,3,"uint8");
M(1:700,1,2) = 255;
M(1:700,900,2) = 255;
M(1,1:900,2) = 255;
M(700,1:900,2) = 255;
% legend colors
M(120:160,590:630,1) = 255;M(120:160,590:630,2) = 255;
M(180:220,590:630,1) = 255;
M(240:280,590:630,3) = 255;
M(300:340,590:630,1) = 255;M(300:340,590:630,3) = 255;
M(360:400,590:630,2) = 255;M(360:400,590:630,3) = 255;
% figure is prepared
figure(1,'position',[20,50,1161,860],'graphicssmoothing','off','resize','off','color',
[0,0,0]);
image(M);axis('off');
```

```
text(100,100,"please wait, I'm working it
out !",'color','r','fontname','verdana','fontsize',20);
% important to display the working figure
ak = kbhit(1);
% now the pixels are assigned to a color, according to the chemical stability
for row = 60:447
  Eh = (243.5 - row)/129;
  for column = 67:414
    pH = (column - 57)/24.786;
    x(2) = (Eh - 1.692)/0.0591 - logConc;                % log Au+
    x(3) = (Eh - 1.498)*3/0.0591 - logConc;              % log Au3+
    x(4) = (Eh - 1.002)*3/0.0591 - logConc + 4*logCl;    % log Au(Cl4)-
    x(5) = (Eh - 1.450)*3/0.0591 + 3*pH;                 % log Au(OH)3
    [w,iw] = max(x);
    switch (iw)
      case 1;M(row,column,1:3) = [255,255,0];    %  yellow   Au(metallic)
      case 2;M(row,column,1:3) = [255,0,0];      %  red      Au+
      case 3;M(row,column,1:3) = [0,0,255];      %  blue     Au3+
      case 4;M(row,column,1:3) = [255,0,255];    %  magenta  Au(Cl4)-
      case 5;M(row,column,1:3) = [0,255,255];    %  cyan     Au(OH)3
    endswitch
  endfor
endfor
% the figure is displayed
image(M);axis('off');
% final text comments over the figure
A{1} = "Electrochemical potentials used for computing this stability diagram :";
A{2} = "Au+ + e- ---> Au(solid)               E° = + 1.692 volt";
A{3} = "AuCl4- + 3 e- ---> Au(solid) + 4 Cl-  E° = + 1.002 volt";
A{4} = "Au(OH)3 + 3 H+ + 3 e- ---> Au(s) + 3 H2O   E° = + 1.45 volt";
A{5} = "Au+++ + 3 e- ---> Au(solid) + 4 H2O   E° = + 1.49 volt";
for i = 1:5
  text(100,500 + 25*i,A{i},'color','w','fontname','verdana','fontsize',14);
endfor
A{1} = "Au - gold";A{2} = "Au+";;A{3} = "Au+++";A{4} = "Au(Cl4)-";A{5} = "Au(OH)3";
for i = 1:5
  text(650,80 + 60*i,A{i},'color','r','fontname','verdana','fontsize',20);
endfor
```

```
text(24,464,"pH   0                                                                    7
14",'color','w','fontname','verdana','fontsize',20);
A{1} = "+1.5 Volt";A{2} = " Eh";A{3} = "-1.5 Volt";
for i = 1:3
  text(436,192*i - 136,A{i},'color','w','fontname','verdana','fontsize',20);
endfor

line([67 414],[247 354],'color','k','linewidth',1,'linestyle','-.');
line([67 414],[88 195],'color','k','linewidth',1,'linestyle','-.');
```



References

[1] W. Stumm and J.J. Morgan, *Aquatic Chemistry, Chemical Equilibria and Rates in Natural Waters, Third Edition*, Wiley Interscience, 1996.

[2] R. H. Petrucci, F. G. Herring, J. D. Madura and C. Bissonnette, *General Chemistry Principles and Modern Applications*, Pearson Prentice Hall, 2010.

[3] Wikipedia, *Pourbaix diagrams,* from https://en.wikipedia.org/wiki/Pourbaix_diagram, 2021.

[4] M. Pourbaix, *Atlas of electrochemical equilibria in aqueous solution*, National Association of Corrosion Engeneers, Huston, Texas, 1974, pp.644.

[5] K. Kenison Falkner and J. Edmond, Gold in seawater, *Earth and Planetary Science Letters*. Vol. 98, No. 2, 1990, pp. 208–221, doi:10.1016/0012-821X(90)90060-B.

[6] F. Haber, Das Gold im Meerwasser, *Zeitschrift für Angewandte Chemie,* Vol. 40, No. 11, 1927, pp. 303–314, doi:10.1002/ange.19270401103.

[7] Wikipedia, *Standard electrode potential (data page),* https://en.wikipedia.org/wiki/Standard_electrode_potential_(data_page).

[8] Chemistry LibreTexts, *Electrochemistry*. 2020, August 14. Retrieved April 3, 2021, from https://chem.libretexts.org/@go/page/125386.

[9] Wikipedia, *Standard state*, https://en.wikipedia.org/wiki/Standard_state

Chapter 14. Data interpolation

Parabolic trend of Atmospheric CO2 (Mauna Loa Observatory)

Sun Spot Number from 1749, according to SISLO

Sea Surface Temperature and its frequency analysis

Strong correlation between Sun Spot Number (Silso) and neutron Flux (OULU). Sunspots number anticipates by about 5 months the neutron flux

Octave simulation of an Air Parcel (10x10x10 m) rising up in the earth atmosphere from sea level.Three cases are considered : isothermal, adiabatic with dry air, adiabatic with wet air.

Koehler curve

14.1 Parabolic trend of Atmospheric CO2 (Mauna Loa Observatory)

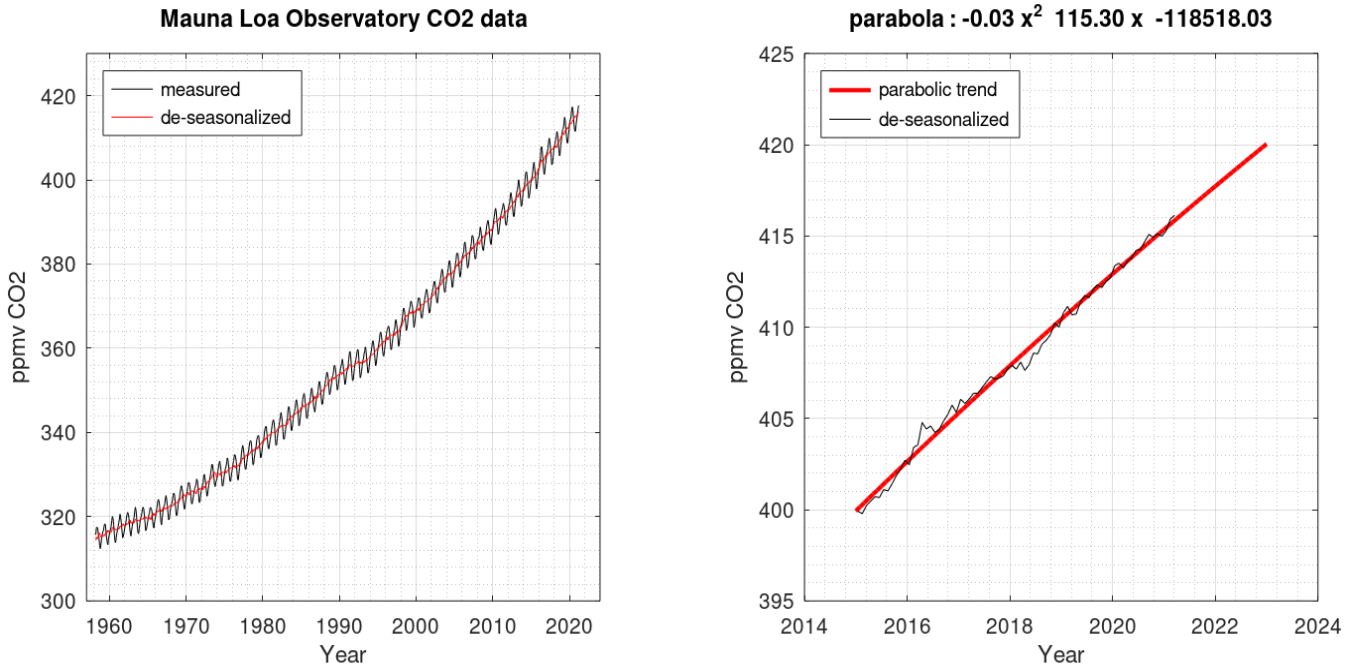


Fig.1 CO_2 ppmv values, monthly and detrended from march,1958 (left). Detrended from january,2015 and the same interpolated.Last value march,2021

Monthly amount of atmospheric CO_2 since March 1958 is regularly measured at the Mauna Loa Observatory, Hawaii. Data are reported as a dry mole fraction defined as the number of molecules of carbon dioxide divided by the number of molecules of dry air (water vapour removed), multiplied by one million (ppm).
The comprehensive monthly data are plotted in fig.1 left, indicating an apparent exponential rise from 316 ppm of 1958.Black monthly values (column 4, see below) and red de-seasonalized data(column 5, see below). The last six years interpolation of the de-seasonalized data shows a parabola with a downward opening. This is why the second degree coefficient is negative. This indicates a decrease in the slope of the Mauna Loa curve in the last six years.

First step is the download of the data into a local file (the name is 'co2_mm_mlo.txt' or whatever you prefer)

```
f =
urlwrite('https://www.esrl.noaa.gov/gmd/webdata/ccgg/trends/co2/co2_mm_mlo.txt','co2_mm_mlo.txt')
```

Once the file is downloaded the above instruction is no more needed, unless a new version is available (normally in the first days of each month). In the Octave here below is transformed into a comment by preceding the line with '%'.

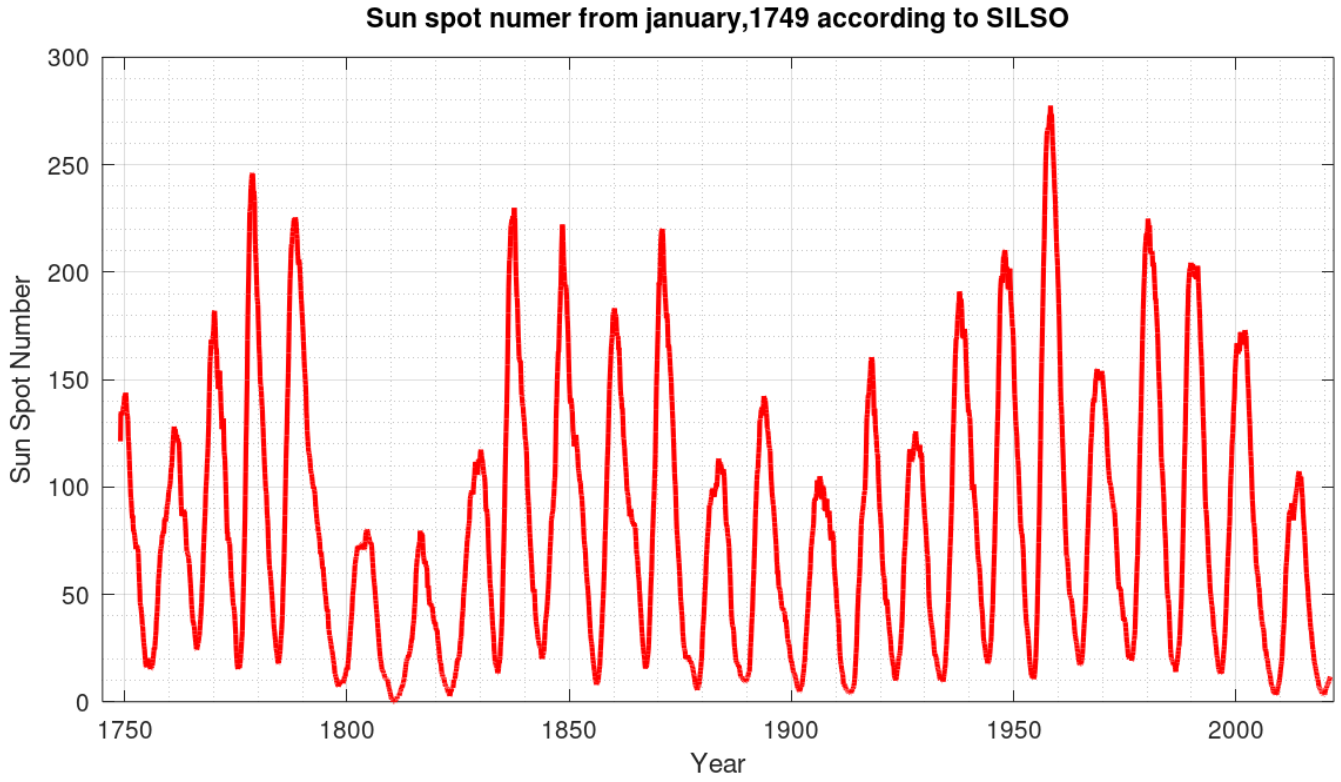
Data are preceded by useful informations, that the user should read.
Data section is dived in 8 columns, respectively :
1 = year, 2= month, 3= decimal date, 4= monthly average, 5= de-seasonalized, 6= number of days per month
7 = standard deviantion of days, 8 = uncertanity of mothly mean.

Data from March 1958 through April 1974 were obtained by C. David Keeling of the Scripps Institution of Oceanography (SIO) and they do not contain 7,8 values. Monthly mean CO2 is constructed from daily mean values. In order to obtain fig.1, we use in the script only column 3,4,5.

The file **'co2_mm_mlo.txt'** is stored in a character string with the instruction **fileread**. The data are preceded by comments, so we have to find data start and end using **index**. Two data starts are here selected, namely 1958,march (that corresponds to the beginning of records) and january,2015. The procedure happens therefore twice; a new character string **'M'** is extracted and then transformed in a numeric matrix **'X'** by the function **str2num()**. Of the matrix **'X'** we use the third column (decimal date), the fourth (monthly averaged data) and the fifth (same data as fouth, but de-seasonalized). **xYear** vector contains the decimal year format, **y** and **yd** store the CO2 concentration data. In fig. 1(a) the monthly and de-seasonalized data are shown. In fig.1(b) some more data elaboration is done, in order to calculate the trend from 2015 until do-day, using de-seasonalized data. The two mighty functions **polyfit()** and **polyval()** do the job of interpolate the points. The first function requires x,y vectors and a degree for the polynomial (in this case 2, but it can be varied), the second calculates the polynomial values according to abscissa xYear and the coefficients stored in vector **p**.
As it can be seen from the negative sign of the second order coefficient, the parabola is downward opening, indicating a decrease in the rate of increment of the CO2 concentration(fig.1).

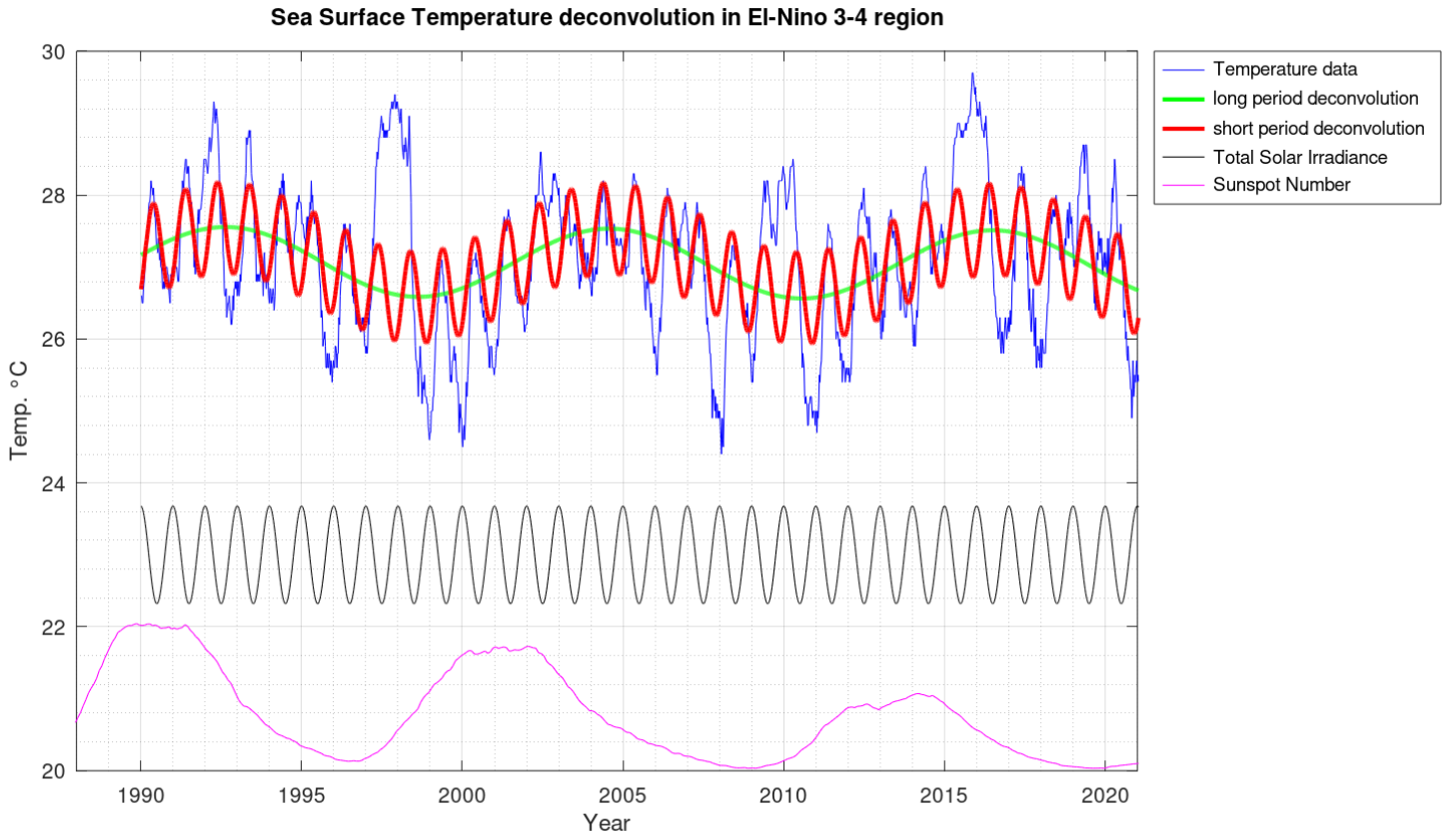
```
clear;clc;format short;format compact;
% f =
urlwrite('https://www.esrl.noaa.gov/gmd/webdata/ccgg/trends/co2/co2_mm_mlo.txt','c
o2_mm_mlo.txt')
S = fileread('co2_mm_mlo.txt');     % co2_mm_mlo.txt is the file 'as downloaded'
from web site
a1 = index(S,' 1958    3');         % find data start, whole set
a2 = index(S,' 2015');              % find data start, from january,2015
% from march,1958
M = S(a1:end);                      % M is a 'reduced' string only containing data
X = str2num(M);                     % the row vector is converted into a proper
matrix
xYear = X(':',3);    % decimal date
y = X(':',4);        % monthly average ppmv CO2 in dry air
yd = X(':',5);       % de-seasonalized data
subplot(1,2,1);plot(xYear,y,'k',xYear,yd,'r');grid on;grid minor on;
axis([1957,2024,300,430]);xlabel('Year');ylabel('ppmv CO2');
legend('measured','de-seasonalized','location','northwest');
title('Mauna Loa Observatory CO2 data');
% from january,2015
M = S(a2:end);                      % M is a 'reduced' string only containing data
X = str2num(M);                     % the row vector is converted into a proper
matrix
xYear = X(':',3);    % decimal date
yd = X(':',5);       % de-seasonalized data
p = polyfit(xYear,yd,2)  % displays the coefficients of parabole starting from
x^2, then x and const
x = (2015:1/12:2023);
y = polyval(p,x);
subplot(1,2,2);plot(x,y,'r','LineWidth',2,xYear,yd,'k');grid on;grid minor on;
axis([2014,2024,395,425]);xlabel('Year');ylabel('ppmv CO2');
legend('de-seasonalized','parabolic trend ','location','northwest');
title(['parabola : ',num2str(p(1),'%.2f'),' x^2  ',num2str(p(2),'%.2f'),' x
',num2str(p(3),'%.2f')]);
```

14.2 Sun Spot Number from 1749, according to SISLO




```
clear;clc;format long;format compact;
%f = urlwrite('http://sidc.oma.be/silso/DATA/SN_m_tot_V2.0.txt','silso.txt')
S = fileread('silso.txt'); % silso.txt is the downloaded text-file.
M = strrep(S,' *','')
X = str2num(M);
xYear = X(':',3);ySpot = X(:,4);
yS = movmean(ySpot,20); %<<< User may vary '20'
plot (xYear,yS,'r','LineWidth',2);grid on;grid minor on;axis([1745,2022,0,300]);
xlabel('Year');ylabel('Sun Spot Number');
title('Sun spot numer from january,1749 according to SILSO');
```

14.3 Sea Surface Temperature and its frequency analysis



National Oceanic and Atmospheric Administration mantains records for global and regional surface temperature. Weekly records of sea surface temperature can to be obtained from the web-address listed in the script ('urlwrite' function). One of the most interesting area is the so-called El-Nino 3-4 region, which stretches around the equator in the pacific ocean, 5N-5S in latitude and 150W-90W in longitude. This area is located in the tropical eastern Pacific Ocean (figure 2) and presents an irregular periodic variation in winds and sea surface temperatures, measured usually by the ENSO index (El Niño–Southern Oscillation).

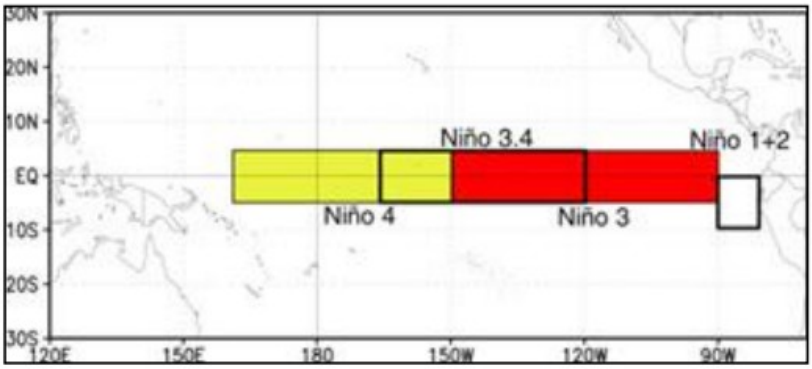


Fig.2 Geographic location of the four El Niño areas.

These variations affect the climate of much of the tropics and subtropics. The warming phase of the sea temperature is known as El Niño and the cooling phase as La Niña. Differently from the most Authors, here we examine not the ENSO index but a more direct parameter, the temperarure of the sea surface.

Data are obtained by the web site ponted out in the script in form of a weekly database from januar 1990. These data are deconvoluted firstly with a large period sinewave(from 550 to 650 weeks) , then the sinewave is detracted and the residuals are again deconvoluted with a sinewave but with a smaller period (from 50 to 60 weeks). The two periods converged to the values of

period 1 = 622.49 weeks = 11.93 years (1 year = 365.24 days)
period 2 = 52.186 weeks = 1.000 years (1 year = 365.24 days)

It is evident that the two periodicity are in close agreement with the solar spot cycle and the TSI (Total Tolar Irradiance), the last varying along with the sun-earth distance in the elliptic earth orbit, with an exact period of 1 year. The solar spot number is in close relationship with solar activity and hence the TSI.
The yearly variation of TSI is to be related to the earth-sun distance along the orbit. We use an approximate formula for this,

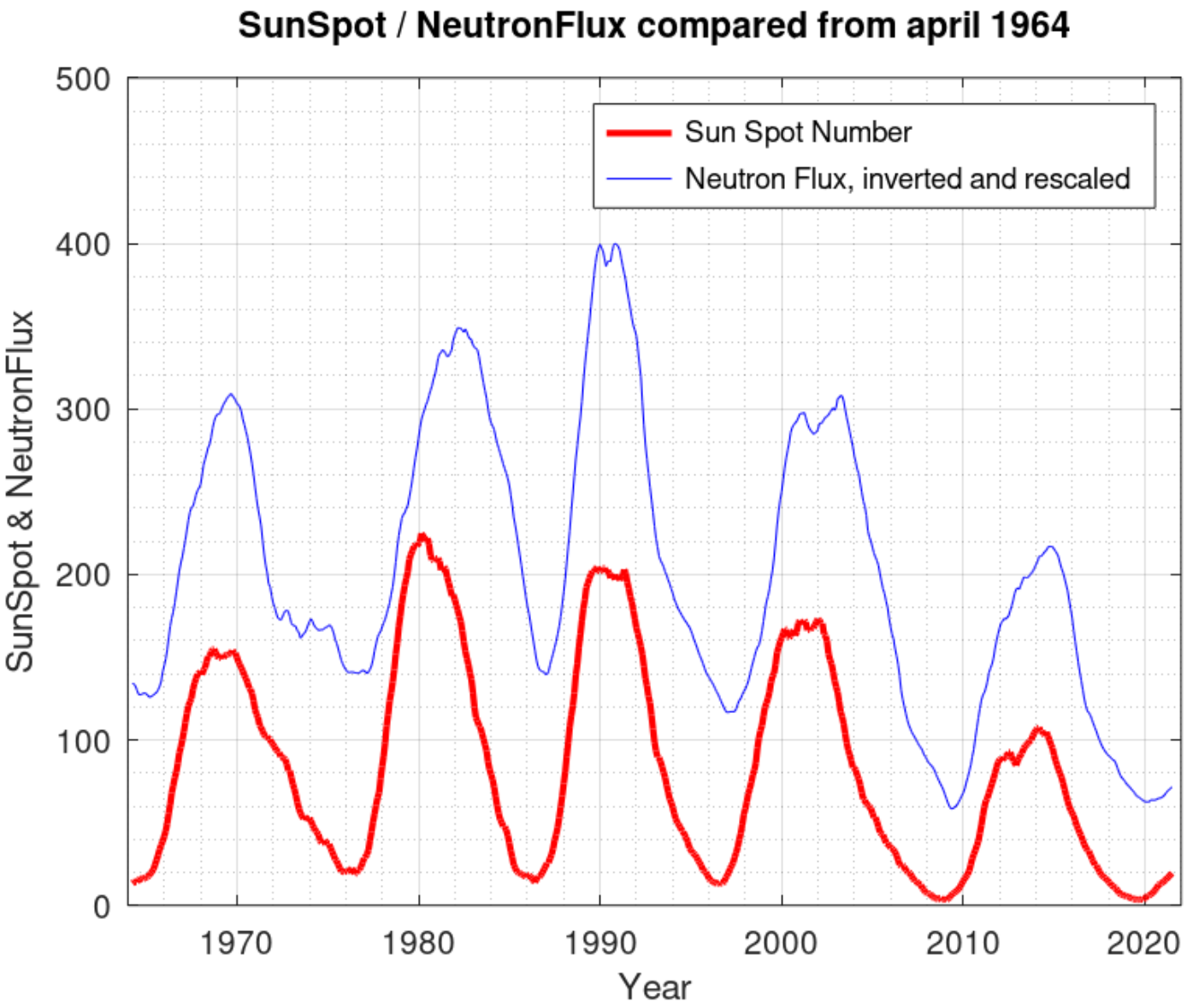
$$Q = S_0\ [1 + 0.034 \cdot \cos(2\pi \cdot \frac{n}{365.25})] \qquad S_0 = 1367\, W/m^2 \quad n = number\ of\ days$$

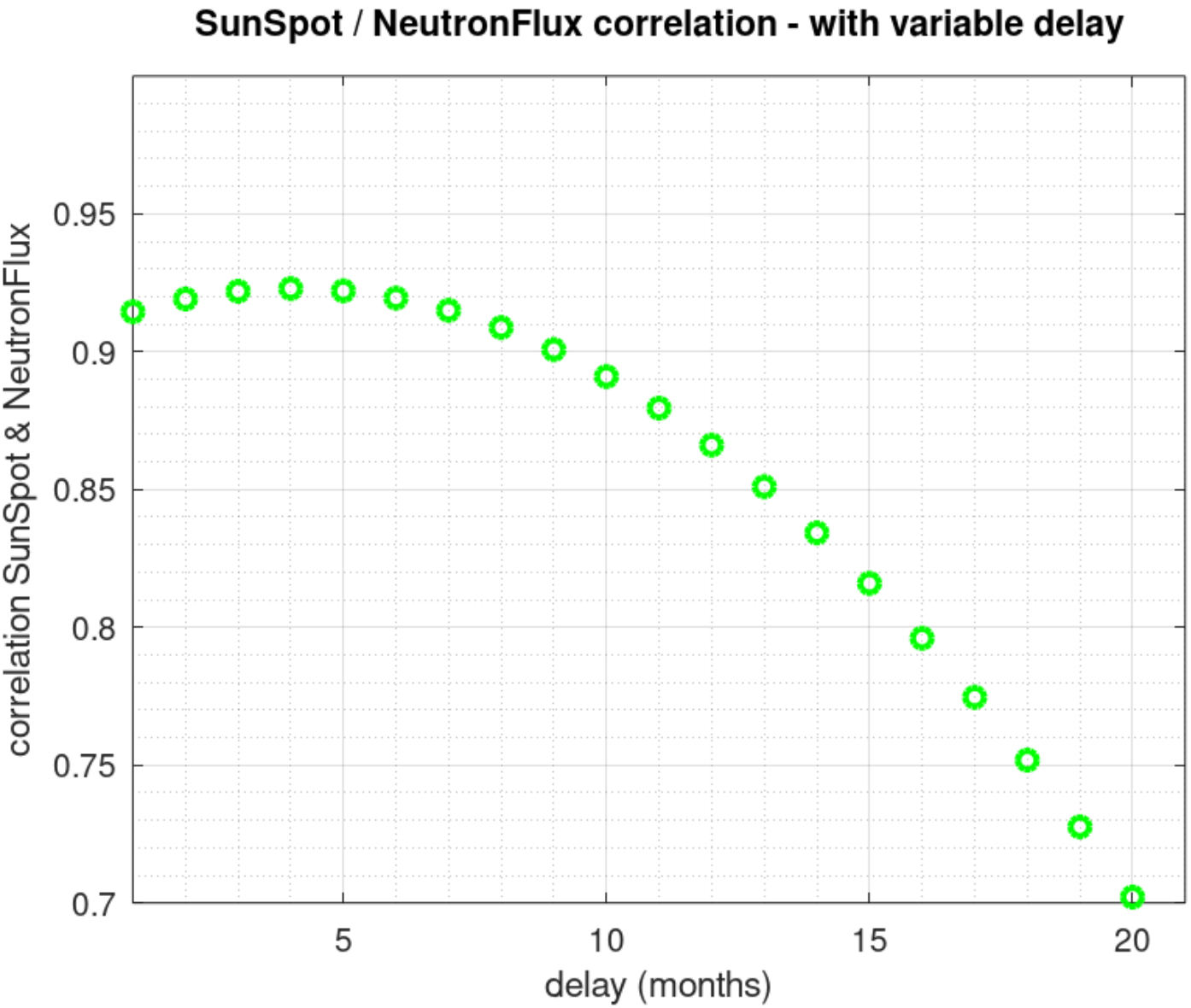
where Q is the TSI along the year. This formula takes advantage from the perihelion (closest approach to the sun) being in the first 2-4 days of january, approximated to 0.
The sunspot number is downloaded from the web site of SILSO, Sunspot Index and Longterm Solar Observations / Royal Observatory of belgium, Brussels, according to the usual procedure of 'urlwrite' function in Octave.
The calculated delay results about 4 months for the short cycle and 3 years for the long one. The heat capacity of the surface layer of the ocean waters could possibly explain a 4 moths delay. The longest delay of 3 years could be explained by the longer chain of events, from the increased solar spots to solar activity and finally to ocean's warming.
The whole script follows :

```
clear;clc;format short;format compact;
global Ud yd xd n;
% differently from Matlab, Octave functions are defined at the very beginning of
the program !
function ydRes = best(p) % ------ minimizer
  global Ud yd xd n;
  Ud(:,3) = sin(2*pi/p*n);   % sin
  Ud(:,4) = cos(2*pi/p*n);   % cos
  G = inv(Ud.' * Ud);
  xd(:,1) = G*Ud.'*yd;
  ydR = yd - Ud*xd(:,1); % residuals
  ydRes = sum(abs(ydR)); % sum of residuals
endfunction              % ------ end of minimizer
%-------------------------------------------------------
% Climate Prediction Center, National Weather Service, NOAA-USA
% f =
urlwrite('https://www.cpc.ncep.noaa.gov/data/indices/wksst8110.for','el_nino_2.txt
')
% the above line to be activated once
S = fileread('el_nino_2.txt');
a1 = index(S,'03JAN1990');
M = S((a1-1):end);
M = strrep(M,'JAN',' 0 '); % January is == 0
M = strrep(M,'FEB',' 1 ');
M = strrep(M,'MAR',' 2 ');
M = strrep(M,'APR',' 3 ');
M = strrep(M,'MAY',' 4 ');
M = strrep(M,'JUN',' 5 ');
M = strrep(M,'JUL',' 6 ');
```

```
M = strrep(M,'AUG',' 7 ');
M = strrep(M,'SEP',' 8 ');
M = strrep(M,'OCT',' 9 ');
M = strrep(M,'NOV',' 10 ');
M = strrep(M,'DEC',' 11 '); % December is == 11
M = strrep(M,'-',' -');
X = str2num(M);
xYear = X(:,1)./365 + X(:,2)./12 + X(:,3); % date in decimal year
yd = X(':',8);  %  El Nino 3 + 4 real ocean temp. 5N-5S 150W-90W
nTot = length(yd);
disp ([X(1,1),X(1,2)+1,X(1,3)]);          % start day month year
disp ([X(nTot,1),X(nTot,2)+1,X(nTot,3)]); % end day month year
Ud = zeros(nTot,4);
Ud(1:nTot,1) = 1;                        % constant plateau
Ud(1:nTot,2) = linspace(-1,1,nTot);    % linear trend (orthogonal)
n = linspace(1,nTot,nTot);
xd = zeros(4,1);
pMin = fminbnd(@best,550,650); % 1st call to the minimizer
x2 = best(pMin);pMin         % once found , pMin is given again to the best()
function
y3 = Ud*xd(:,1);              % y3 is the first SINEWAVE
y4 = yd;
yd = yd - y3;
pMin1 = fminbnd(@best,50,60); % 2nd call to the minimizer
x2 = best(pMin1);pMin1
y5 = Ud*xd(:,1) + y3;        % y5 is the second SINEWAVE
plot (xYear,y4,'b',xYear,y3,'g','LineWidth',2,xYear,y5,'r','LineWidth',2);grid on;
grid minor on;axis([1988,2021,20,30]);hold on;
%-------------------------------------------------------
% Total Solar Irradiance. An Estimate from ithaca web-site
% https://www.itacanet.org/the-sun-as-a-source-of-energy/part-2-solar-energy-
reaching-the-earths-surface/#2.1.-The-Solar-Constant
TSI = 3 + 20*(1 + 0.034*cos(2*pi*xYear));
plot (xYear,TSI,'k');hold on
%-------------------------------------------------------
% SILSO Sunspot Index and Longterm Solar Observations / Royal Observatory of
belgium, Brussels
%f = urlwrite('http://sidc.oma.be/silso/DATA/SN_m_tot_V2.0.txt','silso.txt')
S = fileread('silso.txt'); % silso.txt is the downloaded text-file.
M = strrep(S,' *','');
X = str2num(M);
xYearS = X(':',3);ySpot = X(:,4)./100 + 20;
yS = movmean(ySpot,20);
plot (xYearS,yS,'m'); title ('Sea Surface Temperature deconvolution in El-Nino 3-4
region');
xlabel('Year');ylabel('Temp. °C');
legend ('Temperature data','long period deconvolution','short period
deconvolution','Total Solar Irradiance',...
'Sunspot Number','Location','bestoutside');hold off
```

14.4 Strong correlation between Sun Spot Number (Silso) and neutron Flux (OULU). Sunspots number anticipates by about 5 months the neutron flux





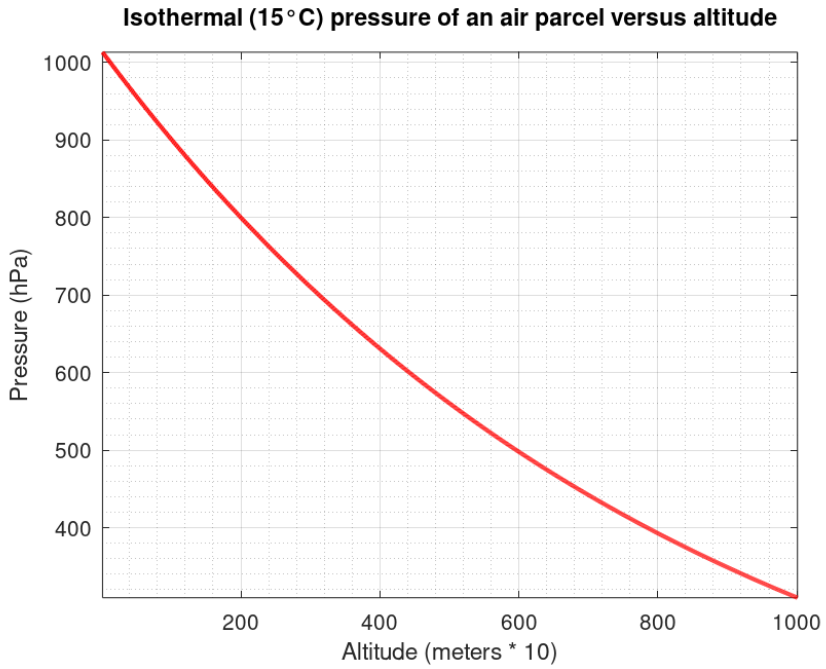
delay (months) = 5 correlation(max) = 0.92291

```
clear all;clc;format short;format compact;
% SILSO Sunspot Index and Longterm Solar Observations / Royal Observatory of
belgium, Brussels
% http://sidc.oma.be/silso/DATA/SN_m_tot_V2.0.txt
% f = urlwrite('http://sidc.oma.be/silso/DATA/SN_m_tot_V2.0.txt','silso.txt')
S = fileread('silso.txt');
a1 = index(S,'1964 04');
M = S(a1:end);
M = strrep(M,' *','');
X = str2num(M);
xYear = X(':',3) - 1/24;yS = X(:,4);% january = 0.5/12
yS = movmean(yS,20); %<<< User may vary '20'

% Neutron Monitor database query - [OULU, DOMC, DOMB NM detectors]-
http://cosmicrays.oulu.fi/
% Station: OULU NEUTRON MONITOR
S = fileread('OULU_monthly.txt');
a1 = index(S,'1964.04.01');
a2 = rindex(S,'CORR') - 1;
M = S(a1:a2);
M = strrep(M,'.',' ');
M = strrep(M,':',' ');
X = str2num(M);
yN = (7000 - X(:,10))/4; % Linerly modified monthly neutron Flux
```

```
% yN = modified neutron Flux = (7000 - NeutronFlux)/4
yN = movmean(yN,20); %<<< User may vary '20'
plot (xYear,yS,'r','LineWidth',2,xYear,yN,'b');grid on;grid minor
on;axis([1964,2022,0,500]);
xlabel('Year');ylabel('SunSpot & NeutronFlux');
title('SunSpot / NeutronFlux compared from april 1964')
legend('Sun Spot Number','Neutron Flux, inverted and rescaled');

% delayed correlation
j = 0;
for i = -10:10
  ++j;
  if i >= 0
    c(j) = corr(yN(1:(end-i)),yS((1+i):end));
  else
    c(j) = corr(yN((1-i):end),yS(1:(end+i)));
  endif
  disp([j,c(j)])
endfor
[w,iw] = max(c);disp(['delay (months) = ',num2str(9-iw),'    correlation(max) =
',num2str(w)]);
figure;
plot(c,'og','LineWidth',2);grid on;grid minor on;axis([1,21,0.7,1]);
xlabel('delay (months)');ylabel('correlation SunSpot & NeutronFlux');
title('SunSpot / NeutronFlux correlation - with variable delay')
```

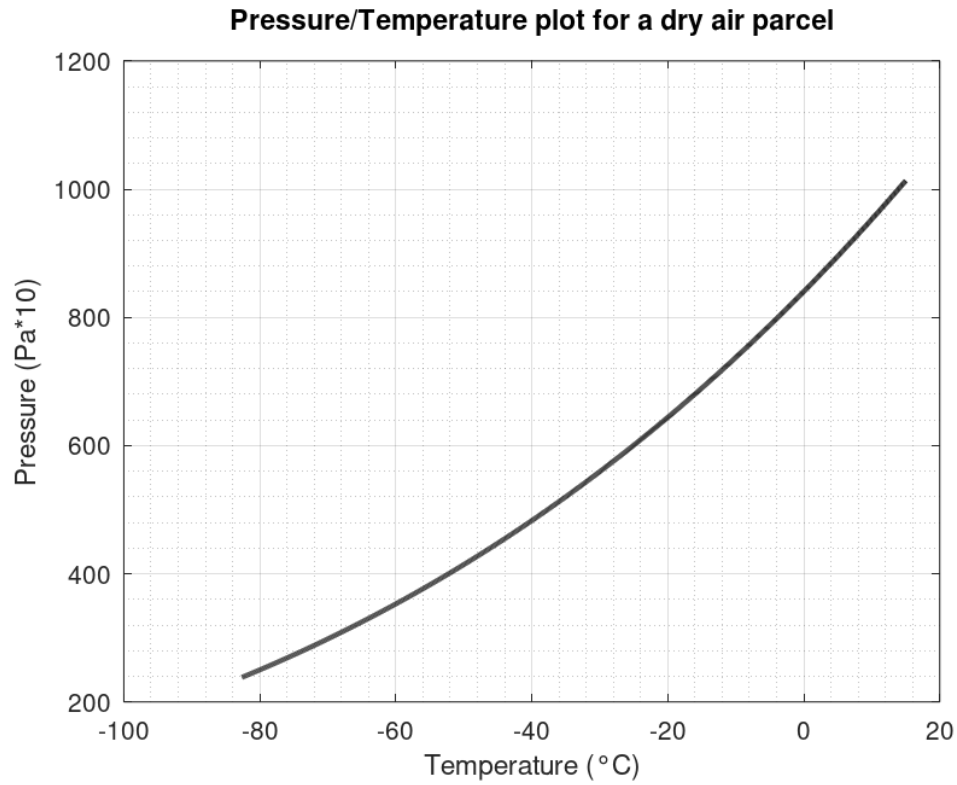
14.5 Octave simulation of an Air Parcel (10x10x10 m) rising up in the earth atmosphere from sea level.Three cases are considered : isothermal, adiabatic with dry air, adiabatic with wet air.

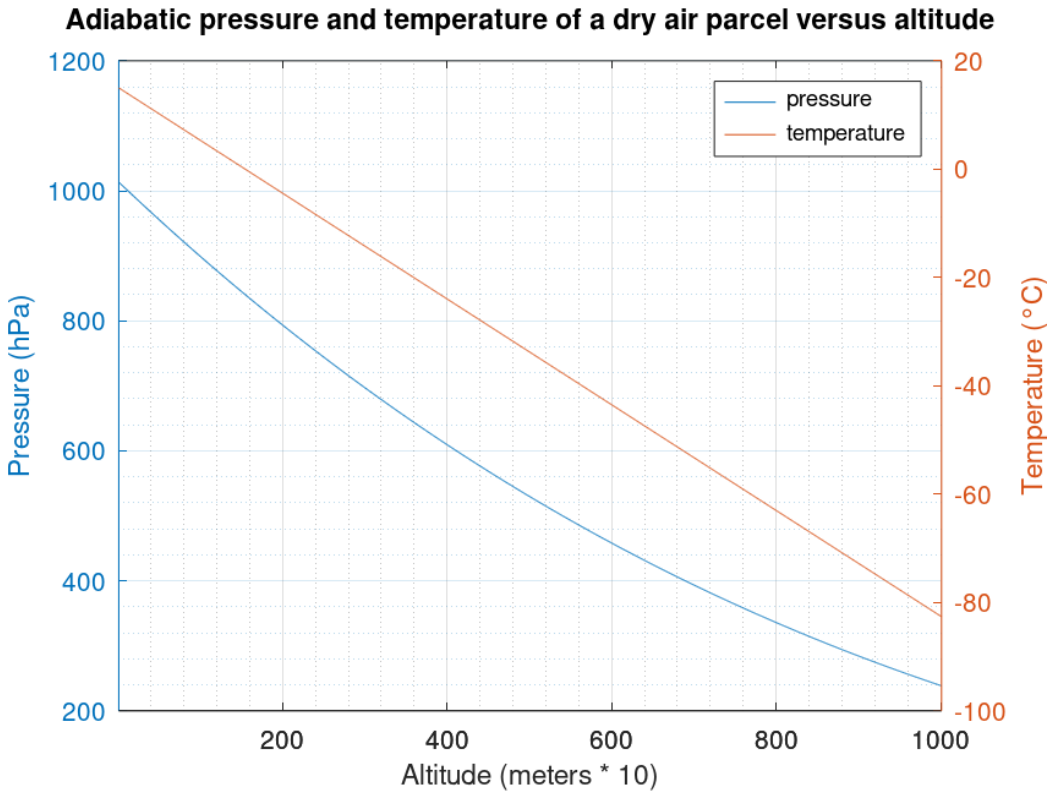
14.5.1- dry air at a constant temperature (15°C)




```
clc;clear all;format short;format compact;
% Air Composition in mol% : N2 78.084,O2 20.9476,Ar 0.9365,CO2 0.0319
% Universal Constants (S.I. Units) as follows
R  = 8.314472;  % Gas Constant
M  = 0.0289697; % Molar Mass of dry air , Kg/mol
g0  = 9.807;    % Gravity acceleration in m/s^2 on equatorial earth surface
Re = 6.38e06;   % Equatorial earth radius
T  = 273 + 15;  % start temperature of Air Parcel

% ----> Start Simulation : 1000 m^3 of air are considered as 'AIR PARCEL' 10 x 10
x 10 m <----

P = zeros(1000,1); P(1) = 101325;% start pressure @ sea level in Pa , 1 atm.
for h = 1:1000
  % the mass of 1000.0 m^3 of dry air is calculated according to pressure p(h); PV
= nRT  ; mass = PVM/(RT)  [kg]
  m1 = 1000*P(h)*M/R/T;
  % the gravity acceleration decreases with altitude, a new is calculated
  g = g0*Re^2/(Re + 10*h)^2;
  % this mass is transformed in force acting on 1.00 m^2
  f1 = m1*g/100;
  % this force is detracted from the pressure, and a new one is calculated
  P(h+1) = P(h) - f1;
endfor

% ----> End of simulation and graphics <----

plot (P/100,'r','LineWidth',2);grid on;grid minor on;axis("tight");
xlabel('Altitude (meters * 10)');ylabel('Pressure (hPa)');
title('Isothermal (15°C) pressure of an air parcel versus altitude');
```

14.5.2 – dry air parcel adiabatic expansion

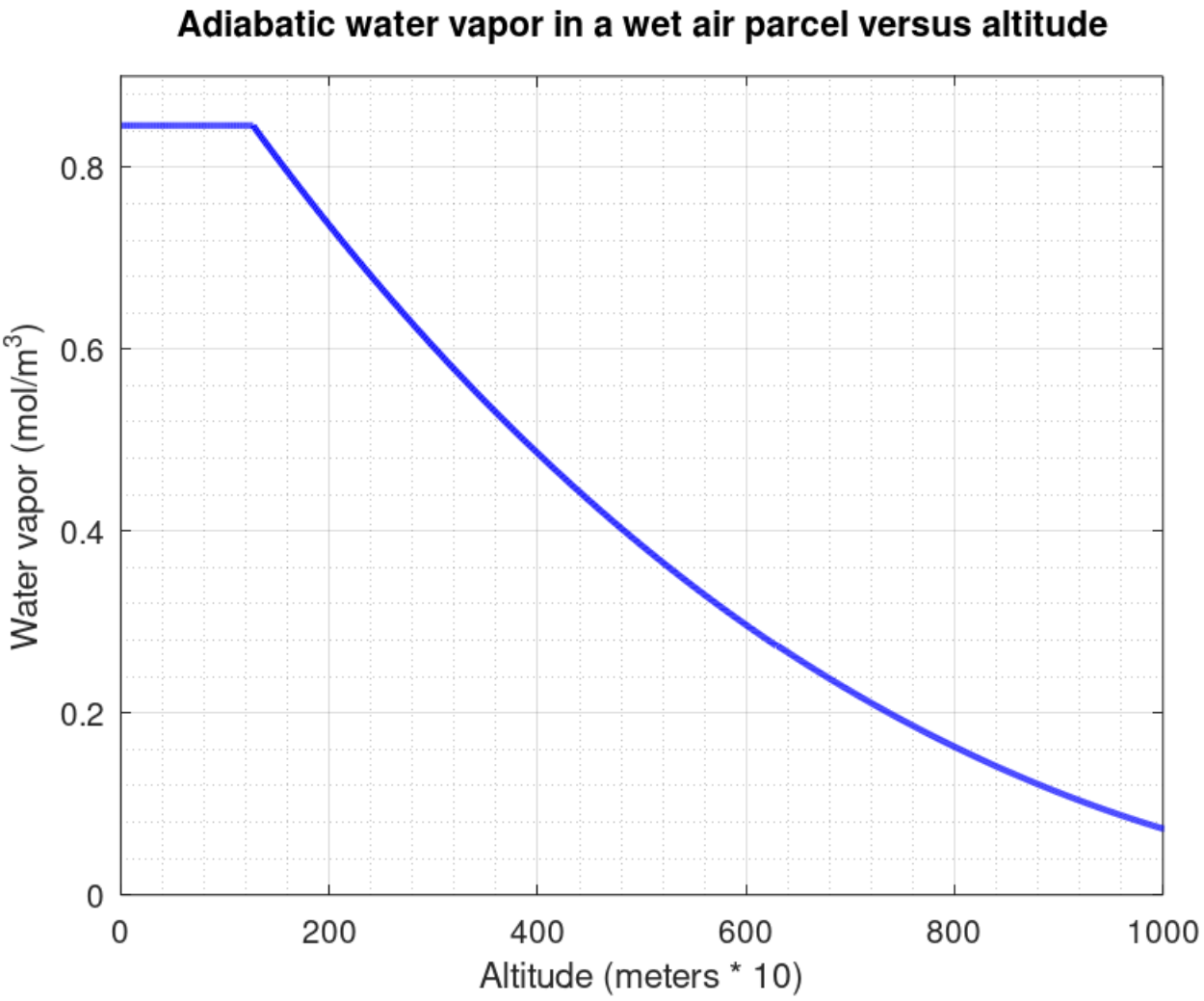


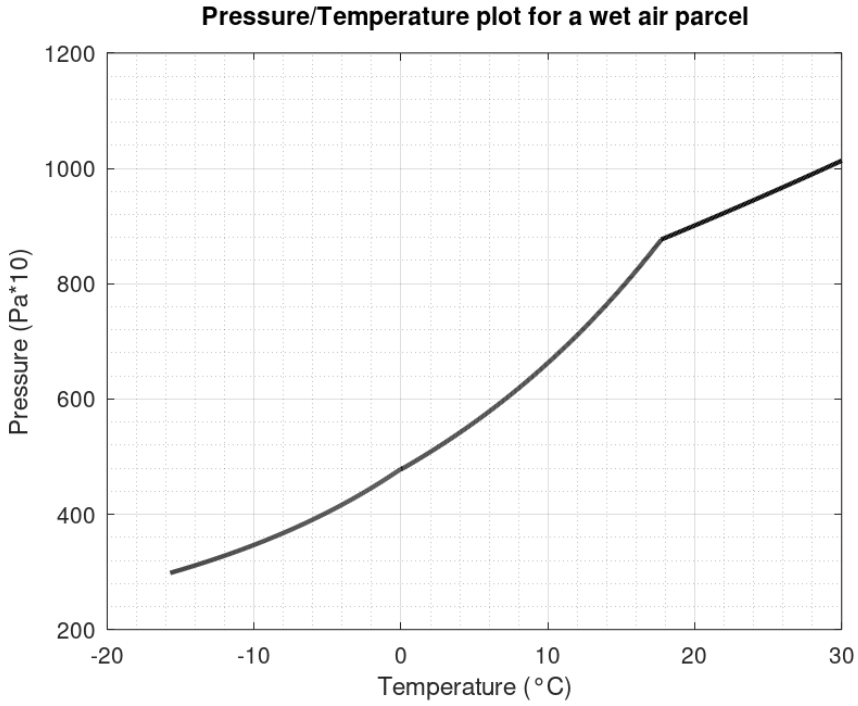


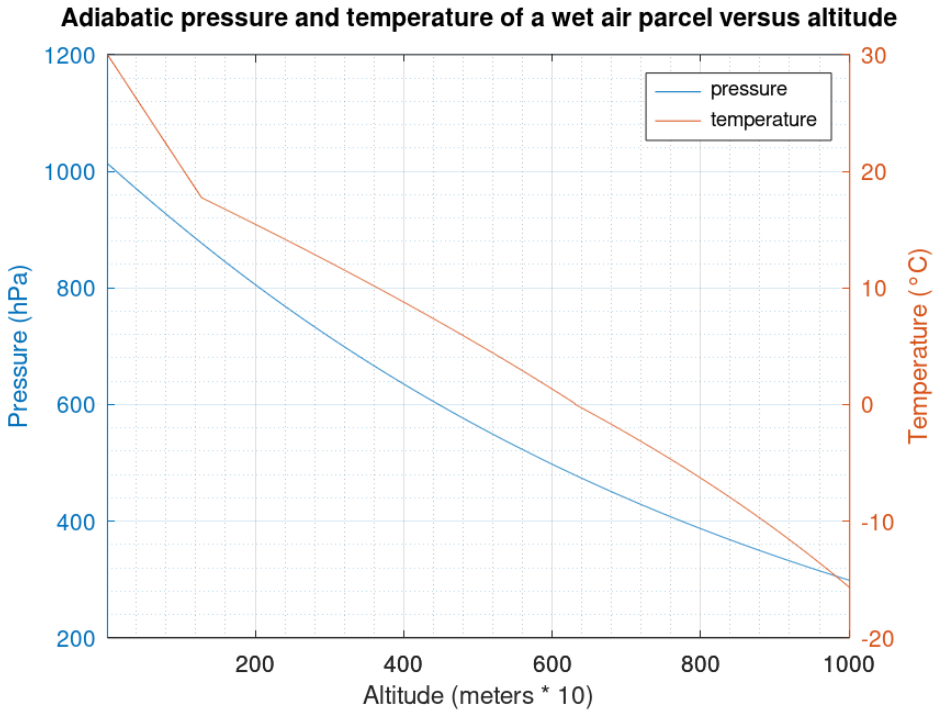

```
clc;clear all;format short;format compact;
% Air Composition in mol% : N2 78.084,O2 20.9476,Ar 0.9365,CO2 0.0319
% Universal Constants (S.I. Units) as follows
R  = 8.314472;  % Gas Constant
M  = 0.0289697; % Molar Mass of dry air , Kg/mol
g0 = 9.807;     % Gravity acceleration in m/s^2 on equatorial earth surface
Re = 6.38e06;   % Equatorial earth radius
T  = 273 + 15;  % start temperature of Air Parcel
% NASA thermodynamic coefficient for dry air
dry_air = [ 1.009950160e+04,-1.968275610e+02,5.009155110,-5.761013730e-
03,1.066859930e-05,...
           -7.940297970e-09,2.185231910e-12,-1.767967310e+02,-3.921504225e+00];
```

```
% ----> Start Simulation : 1000 m^3 of air are considered as 'AIR PARCEL' 10 x 10
x 10 m <----

p = zeros(1001,1); P(1) = 101325;% start pressure @ sea level in Pa , 1 atm.
T = zeros(1001,1); T(1) = 288; % start temperature 15°C
for h = 1:1000
  % the mass m1 and moles of 1000.0 m^3 of dry air is calculated according to
pressure p(h); PV = nRT  ; mass = n*M [kg]
  n1 = 1000*P(h)/R/T(h); m1 = n1*M ; % n1 is the number of moles in the air parcel
  % the gravity acceleration decreases with altitude, a new is calculated
  g = g0*Re^2/(Re + 10*h)^2;
  % this mass is transformed in force acting on 1.00 m^2
  f1 = m1*g/100;
  % this force is detracted from the pressure, and a new one is calculated
  P(h+1) = P(h) - f1;
  % Cp is calculated and dT due to expansion adiabatic work is calculated
  Tx = [T(h)^-2,T(h)^-1,1,T(h),T(h)^2,T(h)^3,T(h)^4];
  Cp = R*sum(Tx.*dry_air(1:7));
  deltaT = 1000*(P(h) - P(h+1))/Cp/n1;
  T(h+1) = T(h) - deltaT;
endfor

% ----> End of simulation ; graphics <----
x = linspace(1,1001,1001);
ax = plotyy (x,P/100,x,T-273);grid on;grid minor on;
xlabel("Altitude (meters * 10)");
ylabel (ax(1), "Pressure (hPa)");
ylabel (ax(2), "Temperature (°C)");
title("Adiabatic pressure and temperature of a dry air parcel versus altitude");
legend("pressure","temperature");
figure
plot(T-273,P/100,'k','LineWidth',2);grid on;grid minor on;
xlabel("Temperature (°C)");
ylabel("Pressure (Pa*10)");
title('Pressure/Temperature plot for a dry air parcel');
```

14.5.3 – wet air parcel adiabatic expansion (with water condensation)


Adiabatic water vapor in a wet air parcel versus altitude
Water vapor (mol/m3)
0
0.2
0.4
0.6
0.8
0
200
400
600
800
1000
Altitude (meters * 10)


Pressure/Temperature plot for a wet air parcel
Pressure (Pa*10)
200
400
600
800
1000
1200
-20
-10
0
10
20
30
Temperature (°C)




```
clc;clear all;format short;format compact;
% Universal Constants (S.I. Units) as follows
R  = 8.314472;  % Gas Constant
Ma = 0.0289697; % Molar Mass of dry air , Kg/mol
Mw = 0.018016;  % Molar mass of water , Kg/mol
g0 = 9.807;     % Gravity acceleration in m/s^2 on equatorial earth surface
Re = 6.38e06;   % Equatorial earth radius

load -binary 'interp.bin' m;
P = zeros(1001,1);P(1) = 101325;   % start pressure in Pa
T = zeros(1001,1);T(1) = 273 + 30; % start temperature
H2O = zeros(1001,1);               % mol H2O(vap) in Air Parcel
Pvap = exp(polyval(m(:,4),T(1)));  % water vapor partial pressure in Pa at T(1),
supposed T>273
StartH2O = 50;                     % start relative humidity
H2O(1) = StartH2O/100*1000*Pvap/R/T(1); % mole water at T(1) in the whole Air
Parcel

% <---- Start simulation ---->

for h = 1:1000
  % saturated water pressure Pvap and heat of condensation dH are calculated
  if T(h) <= 273.15                              % SOLID ICE <--> VAPOR
    Pvap = exp(polyval(m(:,2),T(h)));
    dH = polyval(m(:,1),T(h));
  else                                           % LIQUID WATER <--> VAPOR
    Pvap = exp(polyval(m(:,4),T(h)));
    dH = polyval(m(:,3),T(h));
  end
  % number of water moles at saturation
  H2Osat = 1000*Pvap/R/T(h);
  %..does water vapor condense ?
  if H2Osat < H2O(h)        % water vapor condenses, however 50% of due, in a
single step (kinetics)
    nC = (H2O(h) - H2Osat)*0.5 % nC is the number of water moles condensing
    H2O(h) = H2O(h) - nC;    % refresh of water vapor
  else
```

```
    nC = 0;                 % water doesn't condense
  endif
  H2O(h+1) = H2O(h);
  % the total moles nT of 1000.0 m^3 of wet air is calculated according to
pressure P(h); PV = nRT
  nT = 1000*P(h)/R/T(h);
  % the mass of the Air Parcel is calculated
  m1 = Mw*H2O(h) + Ma*(nT - H2O(h));
  % the gravity acceleration decreases with altitude, a new one is calculated
  g = g0*Re^2/(Re + 10*h)^2;
  % this mass is transformed in force acting on 1.00 m^2
  f1 = m1*g/100;
  % this force is detracted from the pressure, and a new one is calculated
  P(h+1) = P(h) - f1;
  % Cp is calculated and dT due to expansion adiabatic work
  Cp = polyval(m(:,5),T(h));
  deltaT = (1000*(P(h) - P(h+1)) - nC*dH)/Cp/nT;% by now Cp refers to dry air
  T(h+1) = T(h) - deltaT;
endfor

% ----> End of simulation ; graphics <----

x = linspace(1,1001,1001);
ax = plotyy (x,P/100,x,T-273);grid on;grid minor on;
xlabel("Altitude (meters * 10)");
ylabel (ax(1), "Pressure (hPa)");
ylabel (ax(2), "Temperature (°C)");
title('Adiabatic pressure and temperature of a wet air parcel versus altitude');
legend('pressure','temperature');
figure
plot(x,H2O/1000,'b',"LineWidth",2);grid on;grid minor on;axis([0,1000,0,0.900]);
xlabel("Altitude (meters * 10)");
ylabel("Water vapor (mol/m^3)");
title('Adiabatic water vapor in a wet air parcel versus altitude');
figure
plot(T-273,P/100,'k',"LineWidth",2);grid on;grid minor on;
xlabel("Temperature (°C)");
ylabel("Pressure (Pa*10)");
title('Pressure/Temperature plot for a wet air parcel');
```

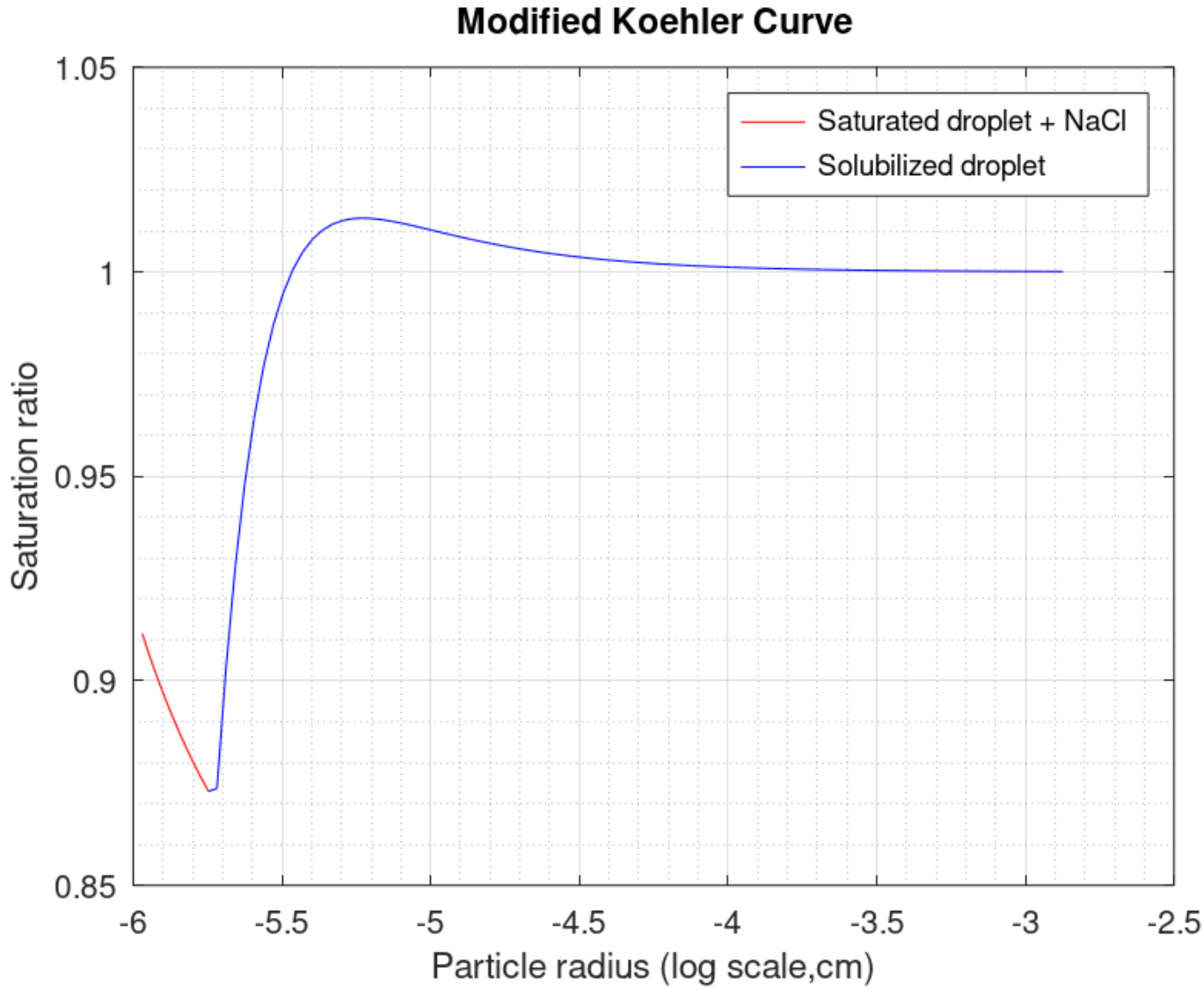
File **interp.bin** interpolates themodynamic data from NASA-CEA database with a third order polynomial

```
clc;clear all;format short;format compact;
% Universal Constants (S.I. Units) as follows
R  = 8.314472;  % Gas Constant
M  = 0.0289697; % Molar Mass of dry air , Kg/mol
% NASA thermodynamic coefficients
% solid H2O, valid 200 to 273.15 K
H2O_cr = [-4.026777480e+05,2.747887946e+03,5.738336630e+01,-8.267915240e-
01,4.413087980e-03,...
          -1.054251164e-05,9.694495970e-09,-5.530314990e+04,-1.902572063e+02];
% liquid H2O, valid 273.15 to 373.15 K
H2O_liq = [1.326371304e+09,-2.448295388e+07,1.879428776e+05,-
7.678995050e+02,1.761556813,...
          -2.151167128e-03,1.092570813e-06,1.101760476e+08,-9.779700970e+05];
% H2O gas, valid 200 to 600 K
H2O_gas = [-3.947960830e+04,5.755731020e+02,9.317826530e-01,7.222712860e-03,-
7.342557370e-06,...
            4.955043490e-09,-1.336933246e-12,-3.303974310e+04,1.724205775e+01];
```

```
% dry air
dry_air = [ 1.009950160e+04,-1.968275610e+02,5.009155110,-5.761013730e-
03,1.066859930e-05,...
           -7.940297970e-09,2.185231910e-12,-1.767967310e+02,-3.921504225e+00];
Rx1 = H2O_gas - H2O_cr; % H2O(solid) <==> H2O(gas) 200 --> 273.15
Rx2 = H2O_gas - H2O_liq;% H2O(liquid) <==> H2O(gas) 273.15 --> 373.15

i = 0;
for T=180:273
  TxH = [-1/T,log(T),T,T^2/2,T^3/3,T^4/4,T^5/5,1,0];
  TxS = [-1/T^2/2,-1/T,log(T),T,T^2/2,T^3/3,T^4/4,0,1];
  % SOLID ICE <--> VAPOR
  DeltaH = R*sum(TxH.*Rx1);DeltaS = R*sum(TxS.*Rx1);DeltaG = DeltaH - T*DeltaS;
  Pvap = 101325*exp(-DeltaG/R/T); % Pvap is the H2O vapor pressure in Pa!!
  ++i;x1(i) = T;y1(i) = DeltaH;y2(i) = log(Pvap);
endfor

i = 0;
for T=273:330
  TxH = [-1/T,log(T),T,T^2/2,T^3/3,T^4/4,T^5/5,1,0];
  TxS = [-1/T^2/2,-1/T,log(T),T,T^2/2,T^3/3,T^4/4,0,1];
  % LIQUID WATER <--> VAPOR
  DeltaH = R*sum(TxH.*Rx2);DeltaS = R*sum(TxS.*Rx2);DeltaG = DeltaH - T*DeltaS;
  Pvap = 101325*exp(-DeltaG/R/T); % Pvap is the H2O vapor pressure in Pa !!
  ++i;x2(i) = T;y3(i) = DeltaH;y4(i) = log(Pvap);
endfor

i = 0;
for T=210:330
  % Cp is calculated (specific molar heat for dry air)
  Tx = [T^-2,T^-1,1,T,T^2,T^3,T^4];
  Cp = R*sum(Tx.*dry_air(1:7));
  ++i;x3(i) = T;y5(i) = Cp;
endfor
m = zeros(4,5);
m(:,1) = polyfit(x1,y1,3);
m(:,2) = polyfit(x1,y2,3);
m(:,3) = polyfit(x2,y3,3);
m(:,4) = polyfit(x2,y4,3);
m(:,5) = polyfit(x3,y5,3);
save -binary 'interp.bin' m
```

14.6 Koehler curve

Koehler curve (see wikipedia '[Koehler theory](#)') does not account for the formation of a saturated salt solution in the aerosol droplet.
Moreover it employs a simplified Raoult equation , not suited for concentrated solutions and uses other unnecessary simplifications.
This simulation is on the contrary starts from a dry aerosol particle (in this example solid NaCl) and considers the growth of the particle by water condensation. The red curve refers to a saturated solution , while the blue to completely solubilized NaCl in the droplet.




```
clc;clear all;format short;format compact;
R = 8.314472;    % Gas Constant
T = 273;         % Temperature in (K)
load -binary 'interp.bin' m;
Ppure = exp(polyval(m(:,4),T)); % vapor pressure of pure liquid water at the
temperature 'T'
Ppure
r1 = 1e-6 ; % dry aerosol particle radius (cm)
            % NaCl mass therein contained is 9.09*r1^3 (grams)
            % 'm' is the mass of water forming the solution around the particle
            % being solubility for NaCl 36g in 100 ml H2O the mass of saturated
solution
```

```
            % around the particle is '1.36*m' and the amount of solubilized NaCl
is '0.36*m'
            % being 2.17 the density (g/cm^3) of solid Nacl and 1.202 the density
of saturated solution we have :
i = 0;ti = 0;
for m = logspace(-18,-8,100); % 'm' is the mass of water forming the solution
around the particle
  ++i;
  if m < (25.25*r1^3)
    ++ti;
    %disp('saturated')
    Va = (10.93*r1^3 + 2.518*m)/2.608; % 'Va' is the volume of the aerosol
particle
    r2 = (3*Va/4/pi)^(1/3); % 'r2' is the wet aerosol particle radius (cm)
    NaCl = 0.36*m; % 'NaCl' is the mass of dissolved NaCl in water around the
particle
    % ---------> All the above holds if the solution is saturated in NaCl, until
there is still some
    % NaCl inside the particle, (9.09*r1^3 - 0.36*m)>=0; m<= (25.25*r1^3) (g)
    % when m>(25.25*r1^3) then we apply the Koehler equation, being the NaCl
solution no more saturated
    % on assuming linear variation of the density of NaCl solution with
concentration we have
  else
    %disp('koehler')
    dens = 5.1*r1^3/m + 1; % 'dens' is the density of the aerosol particle, now
moniphasic (g/cm^3)
    Va = (9.09*r1^3 + m)/dens; % Va is the volume of the particle, now monophasic
    r2 = (3*Va/4/pi)^(1/3);
    NaCl = 9.09*r1^3;
  end

  % now we spot the Raoult equation (see wikipedia for example)
  molefractionH2O = (m/18)/(m/18 + 2*NaCl/58.44); % H2O=18 ; NaCl = 58.44
molecular weights ; i(van't Hoff) = 2
  Pvap = Ppure*molefractionH2O;

  % now we spot the Kelvin equation (see wikipedia ---> ln(P/Pvap)
=2*gamma*Vm/(r2*R*T)   )
  P(i) = Pvap * exp(2*100*0.0728*18e-6/(r2*R*T)); % (r2 in (cm), all the other in
S.I. units)
  radius(i) = log10(r2);
end

plot(radius(1:ti),P(1:ti)./Ppure,'r',radius(ti:end),P(ti:end)./Ppure,'b');grid on;
grid minor on;
xlabel("Particle radius (log scale,cm)");
ylabel("Saturation ratio");
title('Modified Koehler Curve');
legend('Saturated droplet + NaCl','Solubilized droplet')
```

Chapter 15. Frequency analysis

15.1 Introduction

When you have a time sequence of equally (or even unequally) time-sampled data, sometimes your eye can see repeated patterns in it. Sometimes your eye can be fooled, though, things that seem periodic might not be, and it may seem like you have a noisy light curve, but it may turn out that you are sampling many repeated cycles of a periodic variation.

In order to look for periodic signals, there are several mathematical tools. A periodogram is one of them. For a quick introduction the reader is addressed to the wiki page, however there are lots of good textbooks on this subject.

A periodogram calculates the significance of different frequencies in time-series data to identify any intrinsic periodic signals. In this respect it is similar to the Fourier Transform, but is optimized for unevenly time-sampled data, and for different shapes in periodic signals. Unevenly sampled data is particularly common in astronomy, where your target might rise and set over several nights, or you have to stop observing with your spacecraft to download the data.

Many different frequencies and candidate periodic signals are evaluated, and the statistical significance of each frequency is computed based upon a series of algebraic operations that depend on the particular algorithm used and periodic signal shape assumed.

As a result, any physical waveform can be decomposed into a number of discrete frequencies, or a spectrum of frequencies over a certain range. The study of the way general functions may be represented or approximated by sums of simpler trigonometric functions is known in mathematics as Fourier analysis , named after Joseph Fourier (1768-1830), who showed that representing a function as a sum of trigonometric functions greatly simplifies the study of heat transfer.

In sciences and engineering, the process of decomposing a function into oscillatory components is often coupled with the operation of rebuilding the function from its components, known as Fourier synthesis. The time-evolution of a physical process often contains essential information about the nature of its driving force(s). This chapter describes the results obtained by applying the Periodogram analysis (or deconvolution) to temperature records, mainly obtained by satellite data, in the last decades. The records may refer to lower troposphere, lower stratosphere, sea surface temperatures and others, and can be relative to selected areas or to the whole earth.

Methods and scripts described in here can be applied to a much wider collection of physical processes, provided a complete dataset for them is available as a function of time, space or other variables.

15.2 The Periodogram procedure based on a least square refinement

Let us start with an example of deconvolution from a series of data, generated by two sinewave functions with different periods (3, 20 years) in a certain interval, say 0 to 100 years, step 0.5. To the data a random noise is added. The lines are evidenced in orange .

Once the raw data are prepared, the first step in the analysis is to fit the most suitable sinewave function, in a certain range of low frequency, in this case from 10 to 31.6 years
This is done with a *for...endfor* cycle which recursively call the function *best()*. This function contains the powerful *ols* (ordinary least square estimation) function.
Once the best fit is found, the results of the search are displayed in a proper grapic output.
The low frequency deconvolution is then subtracted to the raw data and the resulting data are then analyzed for an higher frequency, in this case from 0.4 to 10 years. Again the results of the search are displayed in a graphic output. Finally, in a third graphic figure, the raw data and the sum of the two deconvoluted sinewaves, are plotted together

```
clear all;clc;format short;format compact;
global x y xd Ud;
function sd = best(p)       % ------ minimizer according to least square summation
  global x y xd Ud;
  Ud(:,1) = sin(2*pi/p*x);    % sin
  Ud(:,2) = cos(2*pi/p*x);    % cos
  [xd,sd,rd] = ols(y,Ud);     % ols = Ordinary Least Square refinement
endfunction                   % ------ end of minimizer

% generate some data in a 100 years period, 2 data available per year
x = linspace(0,100,201);  % a row vector is generated
x = x';                   % usually a column vector is preferred
p1 = 3;p2 = 20;           % these are the 2 periods in year
y = sin(2*pi/p1*x) + 0.5*cos(2*pi/p2*x); % y vector is obtained
g = randn(21); s = g(1:201);s = 0.3*s';  % random noise is added to it
y = y + s;
% find and detract low frequency component (search between 10 and 31.6 years)
k = 0;
for i = logspace(1,1.5,200);  % frequency is spanned by a logarithmic increment
  ++k;xL(k) = i;pL(k) = best(i);
endfor
[n,in] = min(pL);lF = xL(in)
plot (xL,pL,'b');grid on;grid minor on;
xlabel    ('Period    ,    years');ylabel    ('R-squared');title    ('Low    Frequency
Analysis');axis([10,32]);
x2 = best(lF);
y2 = y;
y = y - Ud*xd(:,1);
xd2(:,1) = xd(:,1);

% find high frequency component (search between 0.4 and 10 years)
figure;
k = 0;
for i = logspace(-0.4,1,200);  % frequency is spanned by a logarithmic increment
  ++k;xL(k) = i;pL(k) = best(i);
endfor
[n,in] = min(pL);hF = xL(in)
```

```
plot (xL,pL,'b');grid on;grid minor on;
xlabel ('Period , years');ylabel ('R-squared');title ('High Frequency
Analysis');axis([0,10]);
x2 = best(hF);
figure;

% 3) final graphics
yN = xd2(1)*sin(2*pi/lF*x) + xd2(2)*cos(2*pi/lF*x) + xd(1)*sin(2*pi/hF*x) +
xd(2)*cos(2*pi/hF*x);
plot (x,y2,'b','linestyle',':','linewidth',2,x,yN,'r','linewidth',2);grid on;grid
minor on;axis([0 100 -2 2]);hold on;
```

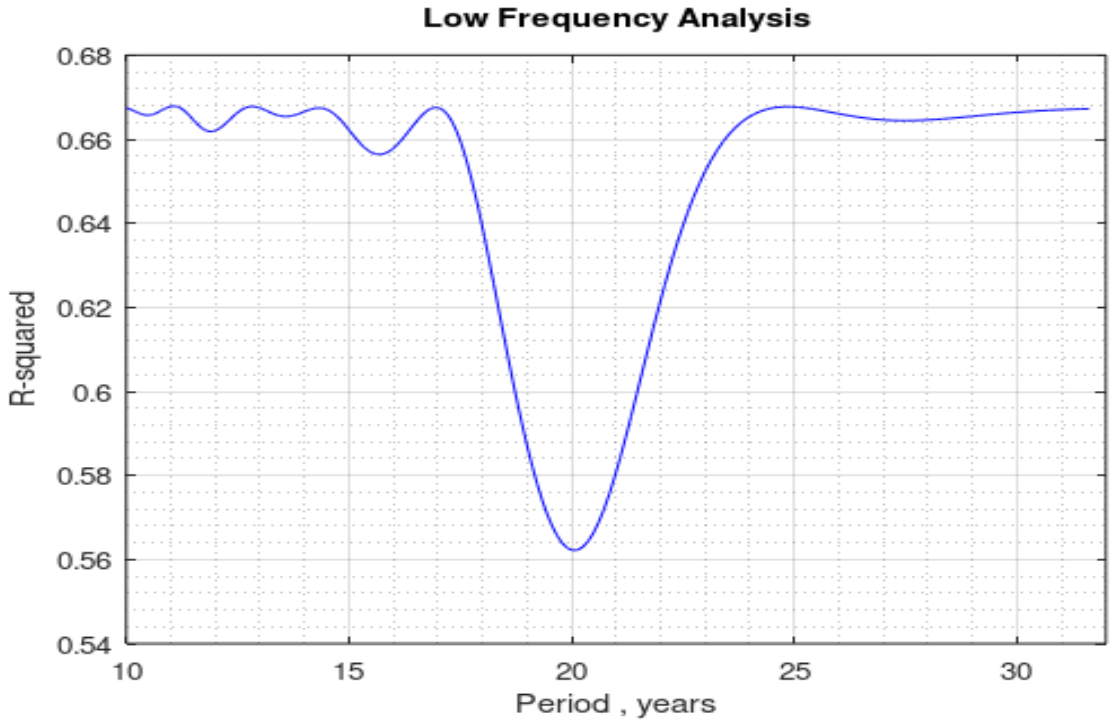


fig. 15.1

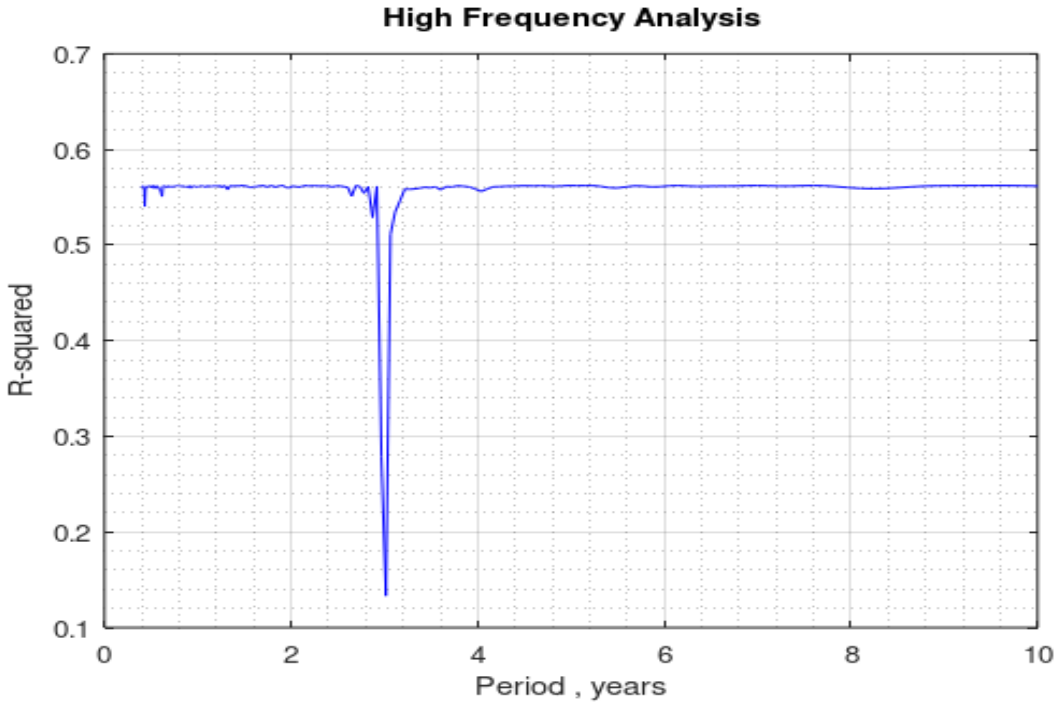


fig. 15.2

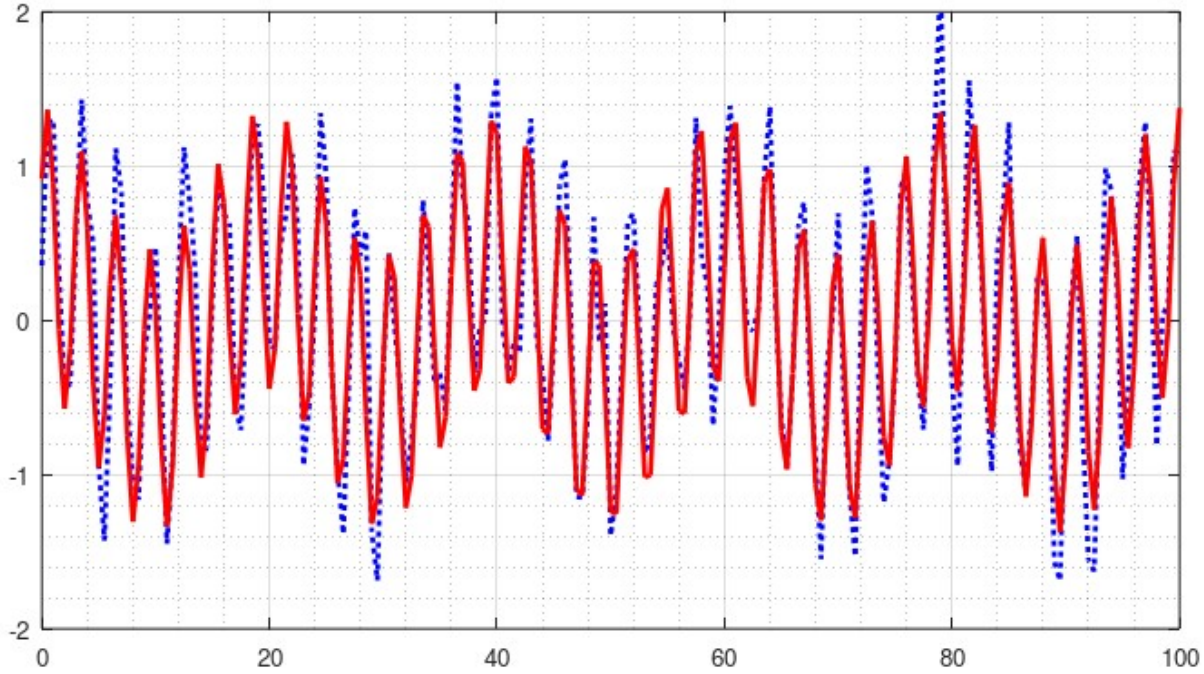


fig. 15.3 Raw data (blue dotted line) and deconvoluted two frequencies summation

15.3 The collection of data from web, following a simple example

Octave offers two different function to download data from a web source, namely *urlwrite* and *webread* . Using *urlwrite* two strings are needed, as in the following example. The first is the *url address* of the text file do be downloaded (in this case the lower-troposphere temperature data), the second being the file name to be assigned to that file, when downloaded (if not otherwise stated, in the current directory)

```
f=urlwrite('http://vortex.nsstc.uah.edu/data/msu/v6.0/tlt/
uahncdc_lt_6.0.txt','uahncdc_lt_6.0.txt');
```

Each time the program is executed, the download will be activated , so (unless new data are available) the line can be temporarly suppressed by a comment sign (%) .
As an example for the data download from the web and subsequent analysis according to a low frequency and an high frequency, we examine the UAH (University of Alabama in Huntsville) repository. Global monthly average temperature are reported on one of the most visited climate-site, the University of Alabama at Huntsville (UAH), USA. Other authoritative web-based sources for global temperature records are RSS(Remote Sensing System) and HadCRUT. The data are obtained by the National Oceanographic and Atmospheric Administration (NOAA) TIROS-N satellite, interpreted by Dr. Roy Spencer and Dr. John Christy, both at Global Hydrology and Climate Center of the same University.

Records start from december,1978 and are regularly updated. They refer to four layers of the planet's atmosphere:

1. lower troposphere
2. middle troposphere
3. tropopause
4. lower stratosphere

For each of the four regions of the atmosphere 24 regions of the globe are examined, namely :

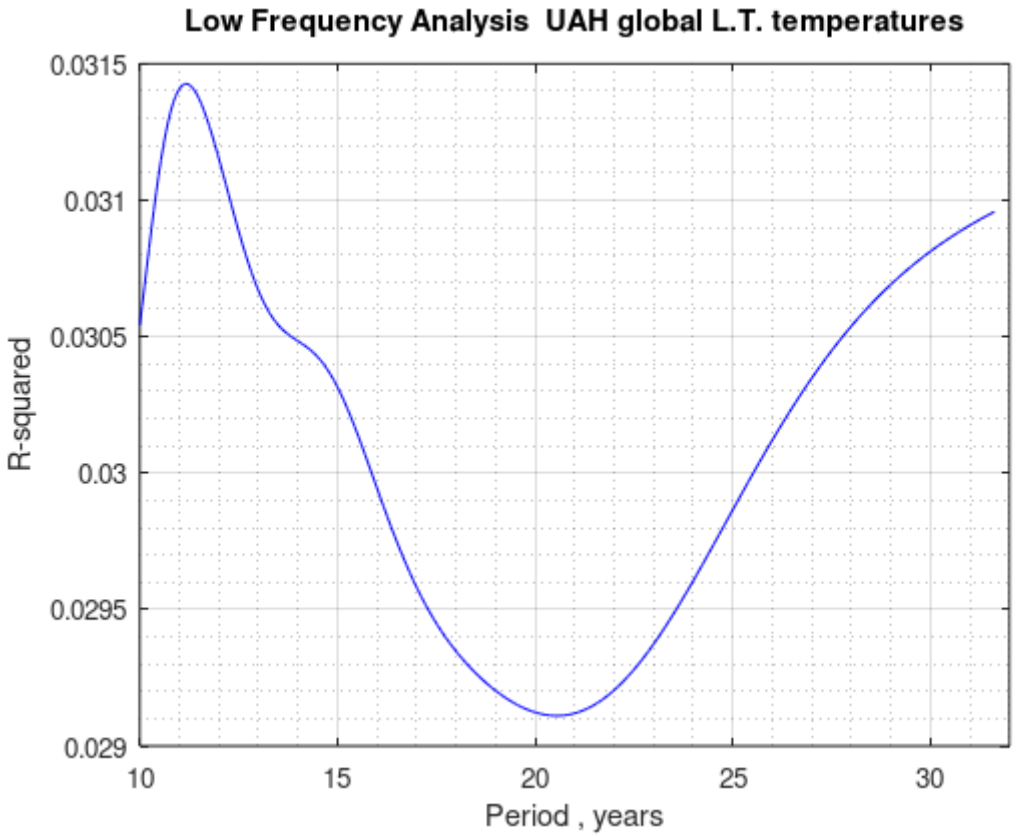
3 = Globe	*4 = Land*	*5 = Ocean*	*whole globe 90S-90N*
6 = Globe	*7 = Land*	*8 = Ocean*	*northern hemisphere 0-90N*
9 = Globe	*10 = Land*	*11 = Ocean*	*southern hemisphere 90S-0*
12 = Globe	*13 = Land*	*14 = Ocean*	*tropics 20S-20N*
15 = Globe	*16 = Land*	*17 = Ocean*	*northern extention 20N-90N*
18 = Globe	*19 = Land*	*20 = Ocean*	*southern extention 20N-90N*
21 = Globe	*22 = Land*	*23 = Ocean*	*north pole 60N-90N*
24 = Globe	*25 = Land*	*26 = Ocean*	*south pole 90S-60N*

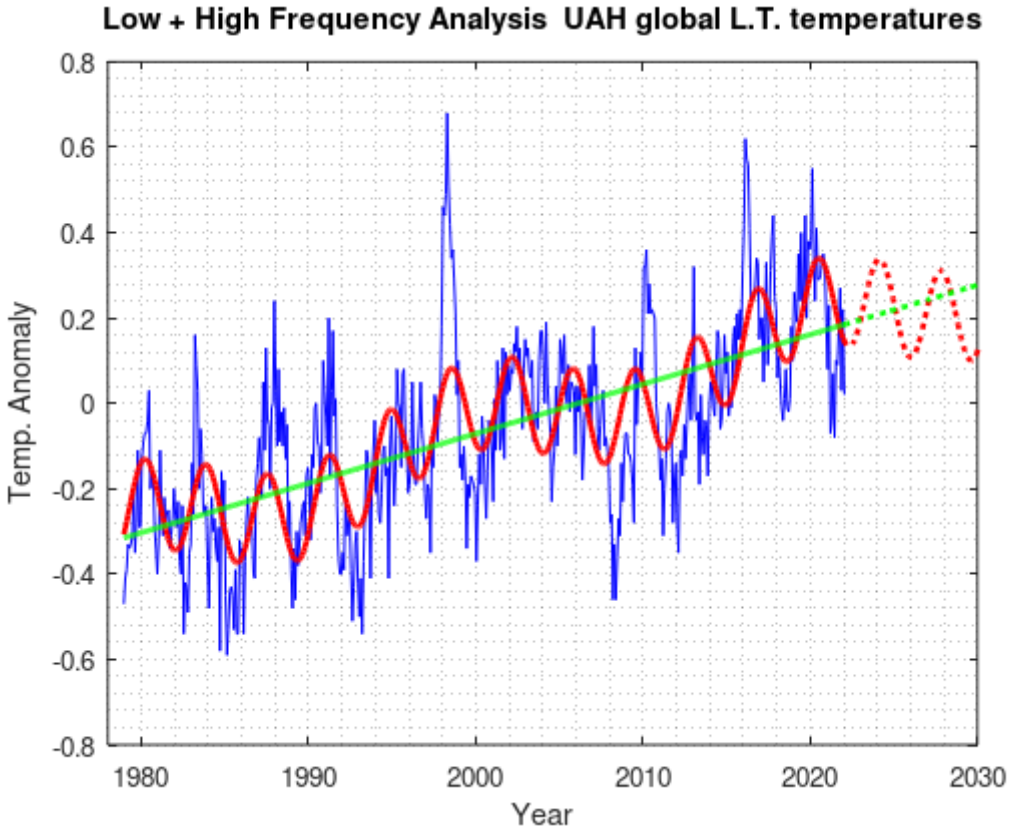
(numbers refer to the columns in the matrix)

The combination of the 4 layers with the 24 regions gives a total of 24x4 = 96 possible graphs; in the following we use combination 3(globe) + lower troposphere. As a further option the program calculates and subtract a linear trend. Indeed in may happen that for a relatively long time span the temperature data are subjected to a bias or linear trend, which must be accounted for. Still a further option allows to predict an estimate for the future , on the basis of the frequency analyses and linear trend

```
clear all;clc;format short;format compact;
global Ud xd yd2 x sumS;
function sd = best(p)   % ------ frequency minimizer
  global Ud yd2 xd x sumS;
  Ud(:,1) = sin(2*pi/p*x);     % sin
  Ud(:,2) = cos(2*pi/p*x);     % cos
  [xd,sd,rd] = ols(yd2,Ud);    % ols = Ordinary Least Square refinement
endfunction                    % ------ end of minimizer
tit = 'UAH global L.T. temperatures'

%f=urlwrite('http://vortex.nsstc.uah.edu/data/msu/v6.0/tlt/
uahncdc_lt_6.0.txt','uahncdc_lt_6.0.txt');

S = fileread('uahncdc_lt_6.0.txt');
choice = 5;  % 3=Globe  4=Land  5=Ocean ...
startYear = 1978;                                    % <<<< User
endYear = 2022;
a1 = index(S,num2str(startYear)) - 1;
a2 = rindex(S,num2str(endYear)) + 170;
M = S(a1:a2);X = str2num(M);
x = X(':',1) + (X(':',2)-0.5)./12;                   %  0.5 = Januar
yd = X(':',choice);
nTot = length(yd);

% 1) find and detract linear slope
pSlope = polyfit(x,yd,1)
y2 = polyval(pSlope,x);
yd2 = yd - y2;
Ud = zeros(nTot,2);xd = zeros(2,1);
%xNew = linspace(endYear,endYear+10,55);
%ySin = polyval(p2,x); % ySin è un vettore colonna
%yNew = polyval(p2,xNew);yNew = yNew.';

% 2) find and detract low frequency component (between 10 and 31.6 years)
k = 0;
for i = logspace(1,1.5,200);
  ++k;xL(k) = i;pL(k) = best(i);
endfor
[n,in] = min(pL);lF = xL(in)
plot (xL,pL,'b');grid on;grid minor on;
```

```
xlabel ('Period , years');ylabel ('R-squared');title (['Low Frequency Analysis
',tit]);axis([10,32]);
x2 = best(lF);
yd2 = yd2 - Ud*xd(:,1);
xd2(:,1) = xd(:,1);
figure;

% 3) find and detract high frequency component (between 0.4 and 10 years)
k = 0;
for i = logspace(-0.4,1,200);
  ++k;xL(k) = i;pL(k) = best(i);
endfor
[n,in] = min(pL);hF = xL(in)
plot (xL,pL,'b');grid on;grid minor on;
xlabel ('Period , years');ylabel ('R-squared');title (['High Frequency Analysis
',tit]);axis([0,10]);
x2 = best(hF);
figure;

% 4) future estimate
yN = y2 + xd2(1)*sin(2*pi/lF*x) + xd2(2)*cos(2*pi/lF*x) + xd(1)*sin(2*pi/hF*x) +
xd(2)*cos(2*pi/hF*x);
plot (x,yd,'b',x,yN,'r','Linewidth',2,x,y2,'g','Linewidth',2);hold on;
xNew = linspace(endYear,endYear+10,55);
yNew = polyval(pSlope,xNew);
yN = yNew + xd2(1)*sin(2*pi/lF*xNew) + xd2(2)*cos(2*pi/lF*xNew) +
xd(1)*sin(2*pi/hF*xNew) + xd(2)*cos(2*pi/hF*xNew);
plot(xNew,yN,'r','Linewidth',2,'Linestyle',":",xNew,yNew,'g','Linewidth',2,'Linest
yle',":");
grid minor on;axis([1978,2030]);
xlabel('Year');ylabel('Temp. Anomaly');title (['Low + High Frequency Analysis
',tit]);
```





Different other application of the above algorithms to Sea Surface Temperatures are to be found here.