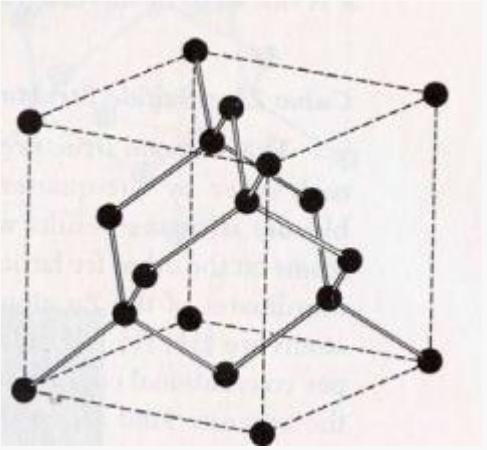
Chapter 2 Hands-on Excises

2.1 Using "vesta" to display crystal structures

1. Using existing "POSCAR" files to show the crystal structures



a) Si in diamond structure:

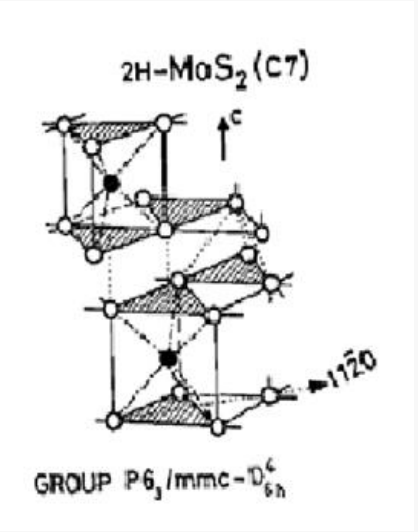
Examine the "POSCAR" file;

Use the "vesta.exe" to display the structure.

b) MoS_2 in 2H hexagonal structure:

Examine the "POSCAR" file;

Use the "vesta.exe" to display the structure.



2.2 Learn use “vi” to open, edit and save text files in linux/unix computing systems

a) Use “vi” to prepare a “POSCAR” file for Si in diamond structure and also MoS_2 in 2H hexagonal structure.
Verify the correctness of your files by displaying the structure using “vesta.exe”

b) Use “vi” to prepare a “POSCAR” file for Co in hcp structure and also $BaTiO_3$ in simple cubic structure (perovskite).
Verify the correctness of your files by displaying the structure using “vesta.exe”

2.3 Learn how to use other linux/unix commands

a) e.g., ls –l; cp; rm; mkdir; pwd; ps; etc .

b) e.g., fetch and send files to other computers using “sftp”, etc .

c) e.g., use “ssh” to logon to remote unix/linux computers.

2.4 Learn how to write, compile and run simple fortran programs.

a) Use “vi” to prepare a fortran or C program, e.g., “hello.f” or “hello.c”.

```
**********
* hello *
**********
c
      program hello              ! comment can be put here
!     character*15 name
      character*15  name
c
      print *, 'Hello, please enter your name'
      read *, name
      print *, name, 'you have a good day !'
c
      END
```

```
/*********
* hello *
*********/

#include <stdio.h>

main()
{
  char name[100];

  printf("Hello, please enter your name\n");

/*  scanf("%s",&name);  */  /* scanf only reads characters without space */
  gets(name);
  printf("%s, you have a good day !\n",name);
}
```

b) Verify the correctness of your code by compiling and executing it.

Home works

a) If your laptop computer is not unix/linux, set up “cygwin” under Window in your laptop computer.

b) If your laptop computer is not unix/linux, download and mount “putty” in your laptop computer.

c) Learn more fundamentals of linux/unix commands.

d) Learn more about preparation, compilation and execution of computer codes, e.g., hello*.f and hello*.c

Other examples of simple fortran programs

```
**********
* hello *
**********
C
      program hello
      real    x
C
100   print *, ' Please enter a real number'
      read *, x
      print *, 'The number you have enetered is', x
      go to 100
C
      END
```

```
**********
* hello *
**********
C
      program hello
      real*8  x
C
100   print *, ' Please enter a double precision real number'
      read *, x
      write(*,200) x
      go to 100
200   format('The number you have entered is ',E30.20)
C
      END
```

Appendix: Brief summary of fortran and C programs

fortran	C
Program structure	
program f function f(x) subroutine abc (x,y,z)	main() double f (double x) void abc (x,y,z)
Data type declarations	
real*8 x,y integer i,j real*8 z(100,100) data y/0.0/ character abc	double x,y; int i,j; double z[100][100]; # define y 0.0; char abc;
Operations	

x*y
x**y
x=y
do k = 1, kmax
x = x + k
x = y + k**2
enddo
do k = kmax, 1, -1
x = x +k
enddo
	x*y
	pow(x,y)
	x=y
	for[k=1,k<=kmax,k++]{
	x = x + k;
	y = y +pow(k,2);
	}
	for(k=kmax,k>=1,k--){
	x=x+k;
	}

fortran	C
Input and ouput to screen	
read(*,*) write(*,*) ‘Enter angle” write(*,*) “angle is’, x	scanf(“%lf”,x) printf(“Enter angle”) printf(“angle is %f”,x)
Input and ouput to files	
open(7,file=‘data.dat’,status=‘new’) write(7,f10.2) angle open(8,file=‘data.dat’,status=‘old’) read(8,*) x	FILE *fout, *fin; fout=fopen(“data.dat”,”w”); fin=fopen(“input.dat”,”r”); fscanf(fin,”%lf”,x);
Control structure: if … else	

if(ab.eq.0) then
print *,”Error!”
endif
if (ab.eq.0) then
print *,”Error!”
else
x=x+1.0
endif
	if(ab==0){
	printf (“Error!”);
	}
	if(ab==0){
	printf(“Error!”);
	}
	else{
	x=x+1.0;
	}

N.B. In fortran, all variables are case insensitive, but in C, everything is case sensitive, .e.g., Abc is not the same as abc.