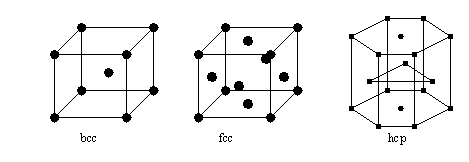
CRYSTAL LATTICE PLOTS

**PROBLEM FORMULATION**

We are well aware of the fact that the nature of metals has fascinated humans for many centuries, because these materials provided people with tools of unsurpassed properties both in war and in their preparation and processing. Pure gold and silver have been known to humans since the Stone Age. Lead and silver were fused from their ores as early as the fourth millennium BC.

The majority of common metals have either a Face Center Cubic Structure, Body Centered Cubic Structure, or a Hexagonal Close Packed structure. The major differences between these structures is the Unit Cell, the building block.



The different cells leads to different physical properties of bulk metals.

For example, FCC metals, Cu, Au, Ag, are usually soft and 'ductile', which means they can be bent and shaped easily. BCC metals are less ductile but stronger, e.g. iron, while HCP metals are usually brittle. Zinc is HCP and is difficult to bend without breaking, unlike copper. Many other features depend upon the crystal structure of metals, such as density, deformation processes, alloying behavior, and much more. Thus, it is important to understand metal structures.

Face Center Cubic consists of an atom at each cube corner and an atom in the center of each cube face. A hard sphere concept can be used to describe atomic packing in unit cells. For cubic crystals the lattice parameter is identical in all three crystal axes. If a corner atom of the FCC unit cell is removed, six atoms are revealed in a hexagonal array. These atoms are closed packed, i.e. they cannot be packed any tighter, and each atom touches its neighbor in any direction. Since a close packed plane such as this can be achieved by removing each of the eight corner atoms and because eight such planes form an octahedron, they are called the 'Octahedral' planes. Thus the FCC structure has four sets of two parallel planes. As parallel planes with the same atomic arrangement are equivalent the FCC structure has four equivalent close packed planes.

Some metals and metal alloys possess high structural strength per unit mass, making them useful materials for carrying large loads or resisting impact damage. Metal alloys can be engineered to have high resistance to shear, torque and deformation. However the same metal can also be vulnerable to fatigue damage through repeated use or from sudden stress failure when a load capacity is exceeded. The strength and resilience of metals has led to their frequent use in high-rise building and bridge construction, as well as most vehicles, many appliances, tools, pipes, non-illuminated signs and railroad tracks.

So now, having known the significance of the metals, it becomes equally important to know the area where we have ways to tackle with the defects in them. The first step before dealing with the defects is to find them. Now to achieve this we are choosing the lattice structure as our base model for the purpose of detection of the defects.

The defect in the metal used in real life applications is found by techniques which can generate the crystal lattice. Now this lattice is compared with one of the standard structures which depict as to how it should be like.

This helps out in finding the common defects like point defect, line defect etc.

What we will be doing here is - we are generating the structure, with which we can compare the result obtained from the device that generates the structure of existing metal lattice.

Now proceeding towards the execution one important thing to look for is, we can use to implement the lattice structure. As we know that a lattice is a 3-dimensional structure, which will have vectors. The mathematical implementation hence includes the vectors and arrays. And one of best possible alternative is to implement this using MATLAB .

**Why have we used MATLAB (Matrix Laboratory) for lattice implementation?**

MATLAB language is very convenient for working with vectors and arrays. It has various types of plots that can be used for visualization of lattices, surfaces and three-­‐dimensional shapes. This further eases the study and analysis along with solutions to the problems faced with the rectifying of metals and other substances.

Matlab Doesn’t require compiler to execute like C,C++. It just executes each sentence as it is written in code. This increase productivity and coding efficiency.

MATLAB has several advantages over other methods or languages:

Its basic data element is the matrix. A simple integer is considered a matrix of one row and one column. Several mathematical operations that work on arrays or matrices are built-in to the MATLAB environment. For example, cross-products, dot-products, determinants, inverse matrices, vectorized operations. Adding two arrays together needs only one command, instead of a ‘for’ or ‘while’ loop.

The graphical output is optimized for interaction. You can plot your data very easily, and then change colors, sizes, scales, etc by using the graphical interactive tools and compare the output with any sample.

**MATHEMATICAL MODEL USED:**

***BRAVAIS LATTICE:***

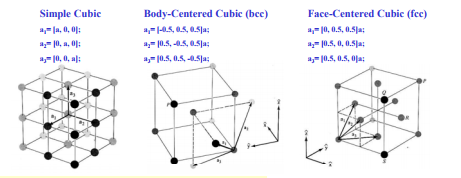
Bravais lattice is the common model of the crystals. It is a discrete periodic array of the points that shows spatial arrangement of the crystal unit cells. The points of the Bravais lattice are:



where a1, a2, a3 are primitive vectors translated through the space n1, n2, n3 times correspondingly. A unit cell can consist of one or more basis atoms. Positions of the basis atoms and primitive vectors generating their translations are the two things that define Bravais lattice.

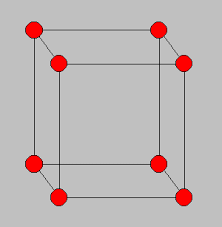
Unit cell consists of the smallest number of basis atoms. It is unique for the crystal.

Since Bravais lattice is infinite, people draw and do calculations with the supercell that consists of several repetitions of the unit cell, e.g. *NxNxN*. Supercell is not unique and it’s symmetry can be different from the symmetry of the unit cell. The number of the basis atoms of the supercell is *N*-­‐times larger than in the unit cell. Supercell translated along translational vectors covers the whole lattice without overlay.



**Simple cubic lattice**

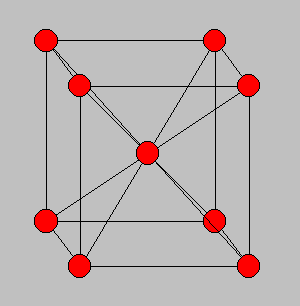
The simple cubic lattice consists of the lattice points identified by the corners of closely packed cubes.



The simple cubic lattice contains 1 lattice point per unit cell. The unit cell is the cube connecting the individual lattice points. The atoms in the picture are shown as an example and to indicate the location of the lattice points. The maximum packing density occurs when the atoms have a radius which equals half of the side of the unit cell. The corresponding maximum packing density is 52 %.

**Body centered cubic lattice**

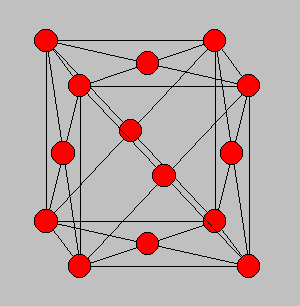
The body centered lattice equals the simple cubic lattice with the addition of a lattice point in the center of each cube.



The body centered cubic lattice contains 2 lattice point per unit cell. The maximum packing density occurs when the atoms have a radius which equals one quarter of the body diagonal of the unit cell. The corresponding maximum packing density is 68 %.

**Face centered cubic lattice**

The face centered lattice equals the simple cubic lattice with the addition of a lattice point in the center of each of the six faces of each cube.



The face centered cubic lattice contains 4 lattice point per unit cell. The maximum packing density occurs when the atoms have a radius which equals one quarter of the diagonal of one face of the unit cell. The corresponding maximum packing density is 74 %. This is the highest possible packing density of any crystal structure as calculated using the assumption that atoms can be treated as rigid spheres.

**LET US CREATE A PLOT FOR OUR LATTICES AND THE WE’LL COMPARE THEM WITH THE ACTUAL CUBIC CELLS (THEORETICALLY)**

Following is the matlab code used :

clear

clf

% Set primitive lattice vectors

a1 = [];

a2 = [];

a3 = [];

% Set basis

b1 = [0.0, 0.0, 0.0];

% One can use function quiver to draw a single vector

quiver3(b1(1),b1(2),b1(3),a1(1),a1(2),a1(3),'g','LineWidth',1.2)

hold on

quiver3(b1(1),b1(2),b1(3),a2(1),a2(2),a2(3),'r','LineWidth',1.2)

hold on

quiver3(b1(1),b1(2),b1(3),a3(1),a3(2),a3(3),'b','LineWidth',1.2)

hold on

N = 0

% Calculate positions of the atoms in the supercell

for n1 = 0:2

for n2 = 0:2

for n3 = 0:2

N = N+1

R(N,:) = n1.\*a1+n2.\*a2+n3.\*a3;

end

end

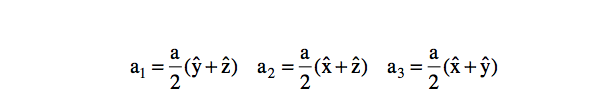
end

% Plot the atoms

scatter3(R(:,1),R(:,2),R(:,3),500,'MarkerFaceColor',[0 .75 .75])

Here, the vectors a1,a2 and a3 are kept empty. For simulation purposes,we feed the primitive vectors for each lattice structure into this code , to get a lattice plot of FCC,BCC or simple cubic.

**PROBLEM 1 a): A face-­centered cubic (fcc) lattice has one basis atom (0, 0, 0) and primitive vectors:**

**Take a=1. Plot these vectors. Then, draw fcc crystal supercell 2x2x2. Insert arrows showing translational vectors for this supercell.**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

PROGRAM CODE WITH EXPLANATION:

clear

clf

% Set primitive lattice vectors

a1 = [0.0 0.5 0.5];

a2 = [0.5 0.0 0.5];

a3 = [0.5 0.5 0.0];

% Set basis

b1 = [0.0, 0.0, 0.0];

% One can use function quiver to draw a single vector

quiver3(b1(1),b1(2),b1(3),a1(1),a1(2),a1(3),'g','LineWidth',1.2)

hold on

quiver3(b1(1),b1(2),b1(3),a2(1),a2(2),a2(3),'r','LineWidth',1.2)

hold on

quiver3(b1(1),b1(2),b1(3),a3(1),a3(2),a3(3),'b','LineWidth',1.2)

hold on

N = 0

% Calculate positions of the atoms in the supercell

for n1 = 0:2

for n2 = 0:2

for n3 = 0:2

N = N+1

R(N,:) = n1.\*a1+n2.\*a2+n3.\*a3;

end

end

end

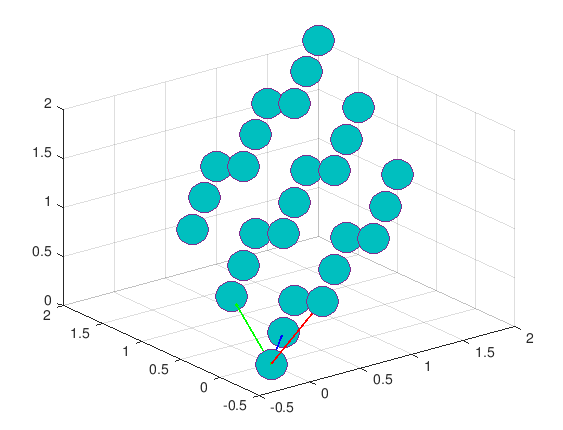
% Plot the atoms

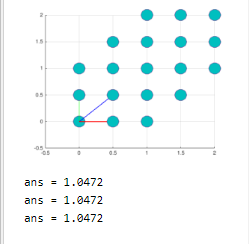
scatter3(R(:,1),R(:,2),R(:,3),500,'MarkerFaceColor',[0 .75 .75])

atan2(norm(cross(a1,a2)),dot(a1,a2))

atan2(norm(cross(a2,a3)),dot(a2,a3))

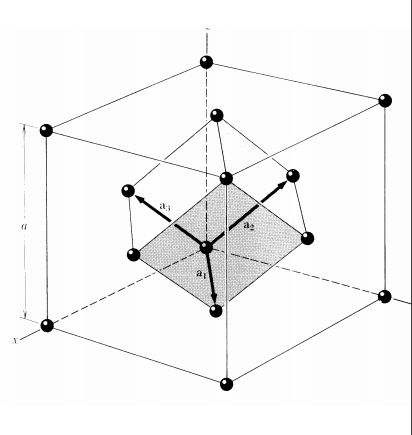
atan2(norm(cross(a3,a1)),dot(a3,a1))





These angles are in radians. 1.0472 radians= 60.0001403 degrees which is equal to the desired output.

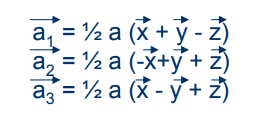
Let us consider the actual face-centered cubic cell.



The angle between adjacent edges is 60 degrees (Theoretically).

Therefore, our plot is almost similar to the theoretical cube.

**PROBLEM 1 b): A body­‐centered cubic (bcc) lattice has one basis atom (0, 0, 0) and primitive vectors:**



**Take a=1. Plot these vectors. Then, draw bcc crystal supercell 2x2x2. Insert arrows showing translational vectors for this supercell.**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

clear

clf

% Set primitive lattice vectors

a1 = [-0.5 0.5 0.5];

a2 = [0.5 -0.5 0.5];

a3 = [0.5 0.5 -0.5];

% Set basis

b1 = [0.0, 0.0, 0.0];

% One can use function quiver to draw a single vector

quiver3(b1(1),b1(2),b1(3),a1(1),a1(2),a1(3),'g','LineWidth',1.2)

hold on

quiver3(b1(1),b1(2),b1(3),a2(1),a2(2),a2(3),'r','LineWidth',1.2)

hold on

quiver3(b1(1),b1(2),b1(3),a3(1),a3(2),a3(3),'b','LineWidth',1.2)

hold on

N = 0

% Calculate positions of the atoms in the supercell

for n1 = 0:2

for n2 = 0:2

for n3 = 0:2

N = N+1

R(N,:) = n1.\*a1+n2.\*a2+n3.\*a3;

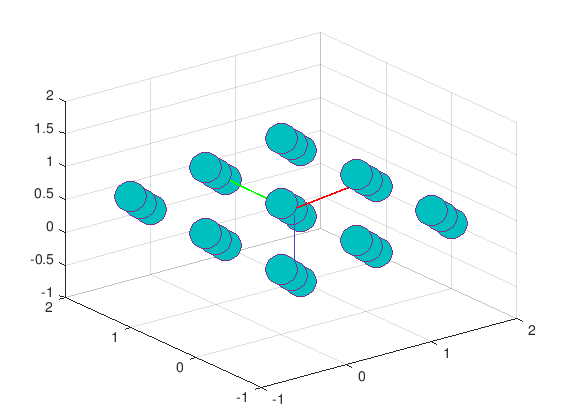
end

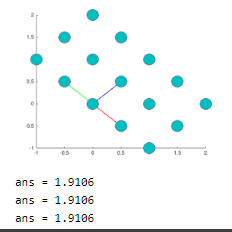
end

end

% Plot the atoms

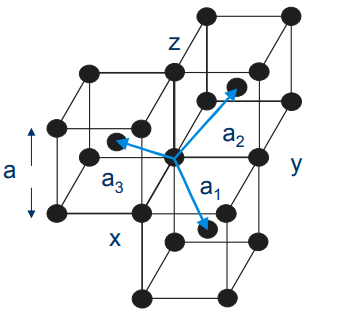
scatter3(R(:,1),R(:,2),R(:,3),500,'MarkerFaceColor',[0 .75 .75])





These angles are in radians. 1.9106 radians= 109.469316 degrees which is equal to the desired output.

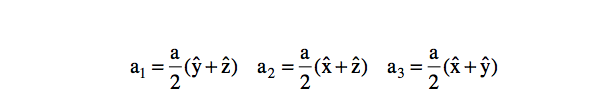
Let us consider the actual body-centered cubic cell.



The angle between adjacent edges is 109.3 degrees (Theoretically).

Therefore, our plot is almost similar to the theoretical cube.

**PROBLEM 1 c): A simple cubic lattice has one basis atom (0, 0, 0) and primitive vectors:**

**Take a=1. Plot these vectors. Then, draw bcc crystal supercell 2x2x2. Insert arrows showing translational vectors for this supercell.**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

clear

clf

% Set primitive lattice vectors

a1 = [0.5 0.0 0.0];

a2 = [0.0 0.5 0.0];

a3 = [0.5 0.0 0.5];

% Set basis

b1 = [0.0, 0.0, 0.0];

% One can use function quiver to draw a single vector

quiver3(b1(1),b1(2),b1(3),a1(1),a1(2),a1(3),'g','LineWidth',1.2)

hold on

quiver3(b1(1),b1(2),b1(3),a2(1),a2(2),a2(3),'r','LineWidth',1.2)

hold on

quiver3(b1(1),b1(2),b1(3),a3(1),a3(2),a3(3),'b','LineWidth',1.2)

hold on

N = 0

% Calculate positions of the atoms in the supercell

for n1 = 0:2

for n2 = 0:2

for n3 = 0:2

N = N+1

R(N,:) = n1.\*a1+n2.\*a2+n3.\*a3;

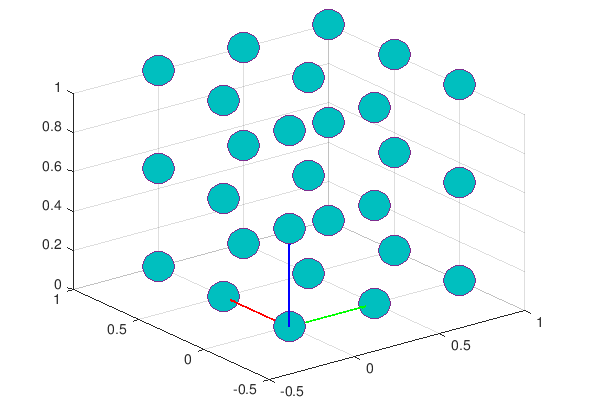
end

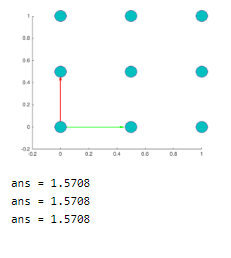
end

end

% Plot the atoms

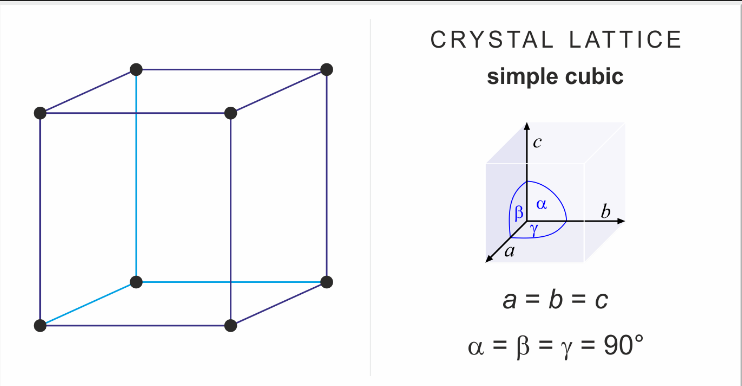
scatter3(R(:,1),R(:,2),R(:,3),500,'MarkerFaceColor',[0 .75 .75])





These angles are in radians. 1.5708 radians= 90.0002105 degrees which is equal to the desired output.

Let us consider the actual simple cubic cell.



The plot obtained above is similar to what we can obtain theoretically.

**VERIFICATION AND VALIDATION**

For the above Problems a), b) and c), we created a 2-D plot of the respective lattices and the edges obtained with (r,g,b) match the vectors of the theoretical plot. We also calculated the angle between the three primitive vectors in a 3-D plot.

As we can infer, the angles matched the actual angles in the theoretical (actual) plots.

Thus, our model is validated as well as verified.

**CONCLUSION**

Large metal components, installed in many big projects such as railways, processing plants need to carry out the quality check for  the metal components(for any inholes)  which can be carried out easily by comparing their lattice with the actual lattice of that particular element and the metal defects can be easily detected.

Thus our model for crystal lattice plot can be used for comparison with the lattice of metal structures for the detection of metals.