



EC210 – Solid State Electronics

Lab 1

Crystal Structure & Introduction to FreeMat

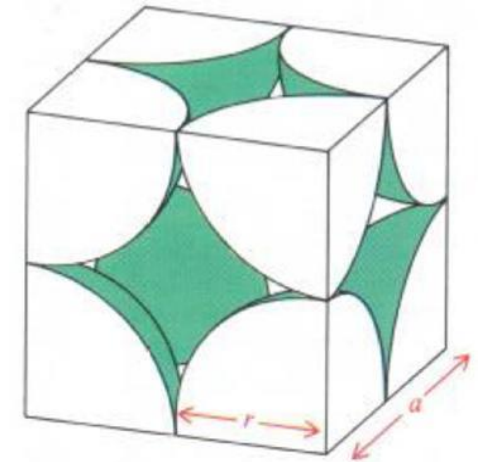
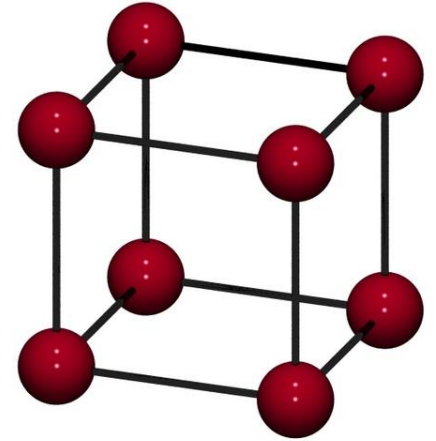
Notes Prepared by: Eng. Nourhan Hazem (GTA)
Class : Solid State Electronics - EC210, Spring 2015
Lecturer: Dr. Amr Bayoumi

Outline

- Introduction to cell structures
- Assignment on cell structures !
- Introduction to FreeMat
- FreeMat Layout
- Using FreeMat as a Calculator
- Creating Variables
- Creating Vectors and Matrices
- Strings
- Assignment on FreeMat !

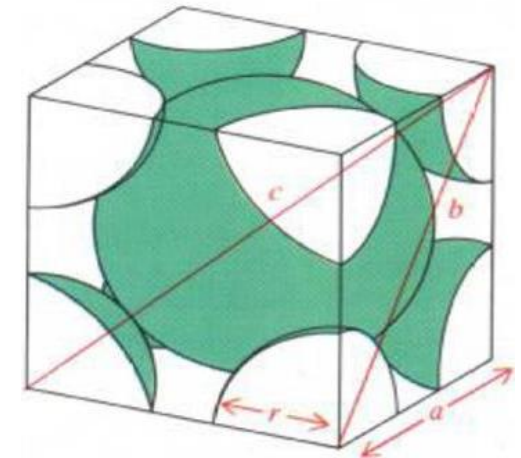
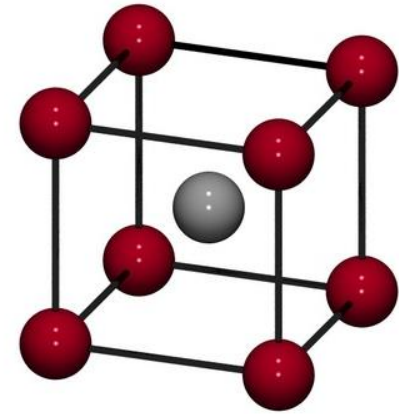
Simple Cubic Unit Cell (SC)

- Number of atoms per unit cell:
 - $n = 8 \text{ corners} * 1/8 = 1 \text{ atom/unit cell}$



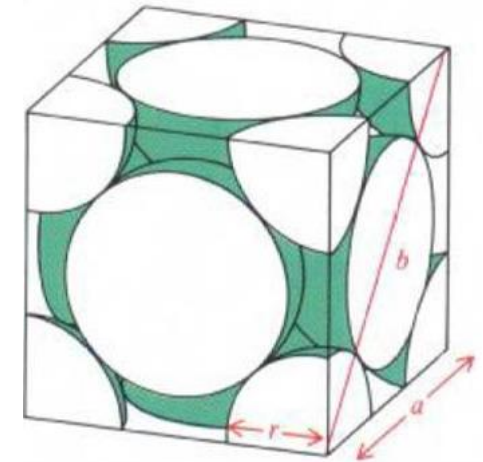
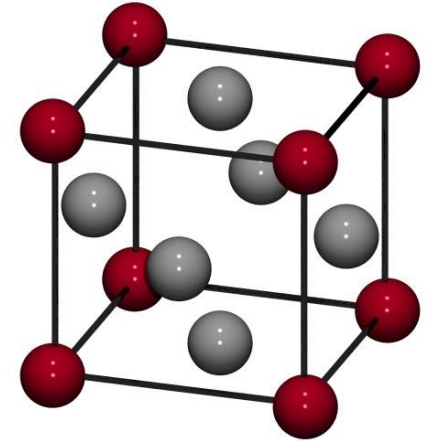
Body-Centered Cubic Unit Cell (BCC)

- Number of atoms per unit cell:
 - $n = 8 \text{ corners} * 1/8 + 1 \text{ atom}$
 $= 2 \text{ atoms/unit cell}$



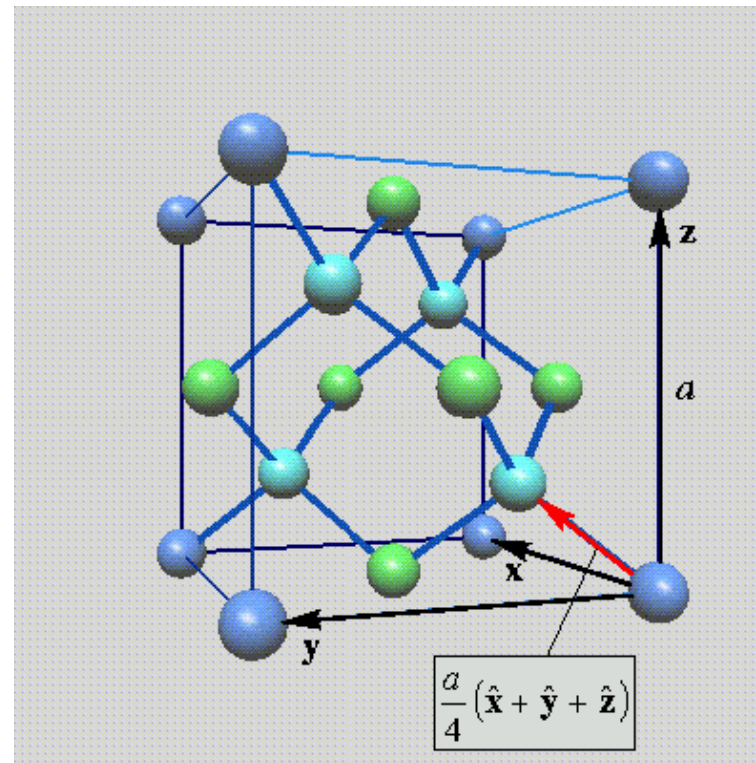
Face-Centered Cubic Unit Cell (FCC)

- Number of atoms per unit cell:
 - $n = 8 \text{ corners} \times 1/8 + 6 \text{ faces} \times 1/2$
 $= 4 \text{ atoms/unit cell}$



Diamond Cubic Cell Structure

- Number of atoms per unit cell:
 - $n = 8 \text{ corners} \times 1/8 + 6 \text{ faces} \times 1/2 + 4 \text{ atoms}$
 $= 8 \text{ atoms/unit cell}$



Assignment !!

- Design a prototype of the SC, BCC or FCC.
- To be submitted before: 24-Feb-15

Bonus !!

- Design a prototype of the diamond cubic cell structure.
- To be submitted before: 24-Feb-15

Introduction

- The name **FreeMat** stands for **Free MATrix** Library.
- **FreeMat** is a free open source numerical computing environment and programming language, similar to MATLAB.
- It has been commercially available since **2002**.



FreeMat Layout

The screenshot shows the FreeMat software interface. The main workspace (1) is empty. The File Browser (2) shows the directory structure. The History (3) panel displays the execution log. The Variables (4) panel shows the current state of variables. The Debug (5) panel is empty.

Filename	Size	Date Mod
.. (Parent F...		
bin	0	9 Feb 2011
help	0	9 Feb 2011
toolbox	0	9 Feb 2011
Uninstall.exe	162890	9 Feb 2011

```
x=[ 1 2 3 4 5 6]
y=[3 -1 2 4 5 1]
plot(x,y)
source 'D:/TA (AAS) EC210 Solid ...
clc
%% Mon 9. Feb 22:51:31 2015
clc
```

Name	Class	Value
ans	double	

Using FreeMat as a Calculator

```
--> 1+2*3
```

```
ans =  
7
```

```
--> sin(pi/4)
```

```
ans =  
0.7071
```

```
--> (7+5)/4
```

```
ans =  
3
```

```
--> 2/sqrt(2)
```

```
ans =  
1.4142
```

```
--> 1/ans
```

```
ans =  
0.7071
```

Creating Variables

- FreeMat variables are created with an assignment statement. **They do not need an initialization.**

`variable name = a value (or an expression)`

- For example:

```
>> x = 10
```

```
>> y = 1.3*42-5*x
```

```
>> z = 63/3+5*2;
```

```
>>
```

```
x =
```

```
y =
```

```
10
```

```
4.6000
```

Creating Vectors and Matrices

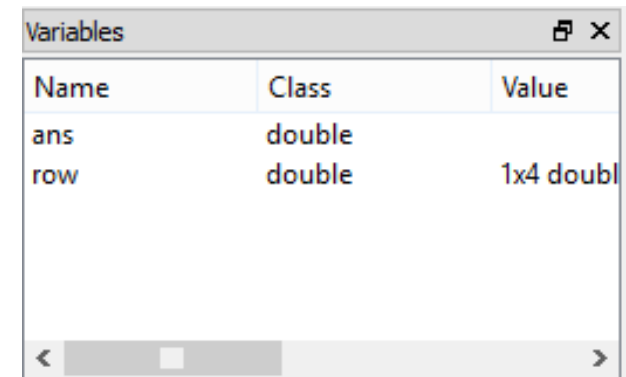
- The power of FreeMat lies within its ability to create vectors and matrices easily.
- **Row vectors** can be written as comma- or space-separated values between square brackets:

```
--> row=[1 2 3 4]

row =
  1 2 3 4

--> row= [1,2,3,4]

row =
  1 2 3 4
```



The screenshot shows the 'Variables' window in FreeMat. It contains a table with three columns: 'Name', 'Class', and 'Value'. The table lists two variables: 'ans' and 'row'. 'ans' is of class 'double' and has a value of '1x4 doubl'. 'row' is also of class 'double' and has a value of '1x4 doubl'. The window has a scroll bar at the bottom.

Name	Class	Value
ans	double	1x4 doubl
row	double	1x4 doubl

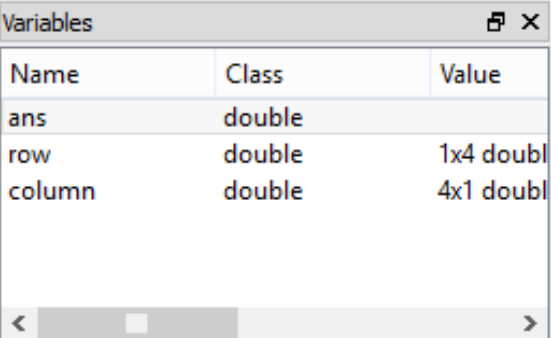
Creating Vectors and Matrices

- **Column vectors** are written as semicolon-separated values between square brackets.

```
--> column=[1;2;3;4]
```

```
column =
```

```
1  
2  
3  
4
```



The image shows a screenshot of the MATLAB 'Variables' window. It contains a table with three columns: 'Name', 'Class', and 'Value'. The table lists three variables: 'ans' (double), 'row' (double, 1x4 double), and 'column' (double, 4x1 double). The window has a title bar with a lock icon and a close button, and a scrollbar at the bottom.

Name	Class	Value
ans	double	
row	double	1x4 double
column	double	4x1 double

Creating Vectors and Matrices

- You can use the two functions 'size' and 'length' to display the vectors' dimensions.

```
>> size(row)
```

```
ans =
```

```
1 4
```

```
>> size(column)
```

```
ans =
```

```
4 1
```

```
>> length(row)
```

```
ans =
```

```
4
```

```
>> length(column)
```

```
ans =
```

```
4
```

Creating Vectors and Matrices

- **Matrices** are combination of row vectors and column vectors **concatenated** with each others.

```
>> a = [1 2;3 4]
```

```
a =
```

```
    1    2  
    3    4
```

```
>> x = [5 6];
```

```
>> y = [7 8];
```

```
>> z = [x;y]
```

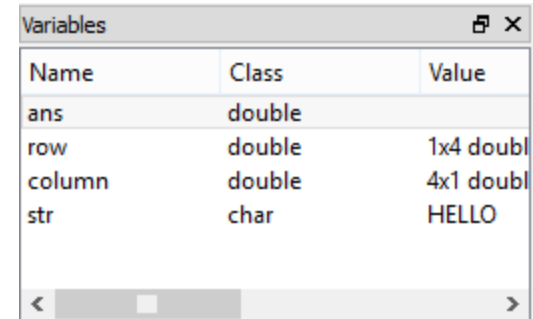
```
z =
```

```
    5    6  
    7    8
```

Strings

- A **string** is a vector consisting of characters.

```
--> str='HELLO'  
  
str =  
HELLO
```



The screenshot shows the 'Variables' window in RStudio. It contains a table with three columns: 'Name', 'Class', and 'Value'. The table lists several variables: 'ans' (double), 'row' (double, 1x4 double), 'column' (double, 4x1 double), and 'str' (char, HELLO). The 'str' variable is highlighted in blue.

Name	Class	Value
ans	double	
row	double	1x4 doubl
column	double	4x1 doubl
str	char	HELLO

Assignment !

1. Make the following variables:

$$a = \sqrt[4]{3 \times 5 + 1}$$

$$b = \frac{2e^6}{\log 37.6}$$

2. Then, evaluate the following expression:

$$c = \frac{\{\log_2(\sqrt{a} + \sqrt[21]{b})\}^\pi}{\tan\left(\frac{a}{be^5} \times \frac{\pi}{3}\right)}$$

3. Define a string with your name, ID and class.

• **Due date: 16-Feb-15**

Thank you for your attention