

# **ELECTRON-SOLID INTERACTION**

By

**AZLAN BIN ARIFFIN**

## **FINAL YEAR PROJECT REPORT**

**Submitted to the Electrical & Electronics Engineering Programme  
in Partial Fulfillment of the Requirements  
for the Degree  
Bachelor of Engineering (Hons)  
(Electrical & Electronics Engineering)**

**Universiti Teknologi Petronas  
Bandar Seri Iskandar  
31750 Tronoh  
Perak Darul Ridzuan**

**© Copyright 2007**

**by**

**Azlan bin Ariffin, 2007**

**CERTIFICATION OF APPROVAL**

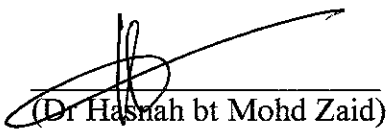
**Electron-Solid Interaction**

by

Azlan bin Ariffin (4083)

A project dissertation submitted to the  
Electrical & Electronics Engineering Programme  
Universiti Teknologi PETRONAS  
in partial fulfilment of the requirement for the  
BACHELOR OF ENGINEERING (Hons)  
(ELECTRICAL & ELECTRONICS ENGINEERING)

Approved by,



(Dr Hashnah bt Mohd Zaid)

Project Supervisor


UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

June 2007

## CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



---

AZLAN BIN ARIFFIN

## ABSTRACT

This final report is intended to be evaluated by the external examiner before the final presentation is carried out. The report covers all information that has been collected starting from the previous findings until the current stage of the project. The current project methodology, discussion and data that have been obtained are also included in this report. The literature review covers the background study of electron-solid interaction. Also, the method used to calculate the probability of interaction between the electron and the intended material is discussed. In addition, the properties of nuclei and findings on Rutherford scattering are added. Molecular geometry and molecular bonding theories is included as to further strengthen the understanding of molecules nature. The project methodology covers the methods that have been used to conduct research and its relevance with regards to the project, such as collecting information from various resources. The data and discussion will reflect on the formula derived to calculate the probability of interaction. In addition, electrostatic force and energy conservation calculation is done to represent the physics of the interaction between the electron and the material. The structure and properties of selected materials such as Graphite and Silicon is included to be discussed. Finally, the graphical user interface for the simulation is designed using Graphical User Interface Design Environment (GUIDE).

## ACKNOWLEDGEMENT

*In the name of ALLAH, the Most Graceful and Most Merciful,*

I am indebted to many individuals who help me throughout this final year project. First of all I would like to express my greatest gratitude to my supervisor, Dr. Hasnah bt Mohd Zaid for her expert guidance, attention and suggestions, support and advices regarding the project and difficulties faced during the project execution. I would also like to express a special thank to my family members for their priceless support, constant love, valuable advices and their understanding to me. Without their enormous encouragement, all my effort in preparing this project would have not successful. Last but not least, my appreciation goes to the individuals or groups that have helped me in any possible way to complete the project. Above all, I would like to thank God for making it possible for this project until this.

## TABLE OF CONTENTS

<b>Certification.....</b>	<b>i</b>
<b>Abstract.....</b>	<b>iii</b>
<b>Acknowledgement.....</b>	<b>iv</b>
<b>Table of Contents.....</b>	<b>v</b>
<b>List of Figures.....</b>	<b>vii</b>
<b>List of Equations.....</b>	<b>viii</b>
<b>Abbreviation &amp; Nomenclature.....</b>	<b>viii</b>
<b>Chapter 1: Introduction.....</b>	<b>1</b>
1.1 Background of Study.....	1
1.2 Problem Statement.....	2
1.2.1 Problem Identification.....	2
1.2.2 Significance of the Project.....	3
1.3 Objective and Scope of Study.....	4
1.3.1 The Relevancy of the Project.....	4
1.3.2 Feasibility of the Project within Scope and Time Frame.....	4
<b>Chapter 2: Literature Review.....</b>	<b>5</b>
2.1 Electron Scattering.....	5
2.2 Monte Carlo Method (Random Numbers and Probability Method).....	6
2.3 Properties of Nuclei.....	7
2.4 Rutherford Scattering.....	8
2.5 Molecular Geometries and Bonding Theories.....	10
<b>Chapter 3: Methodology.....</b>	<b>12</b>
3.1 Procedure Identification.....	12
3.2 Tools.....	13

<b>Chapter 4: Data and Discussion</b> .....	<b>14</b>
4.1 Structure and Properties of Material.....	14
4.1.1 Graphite.....	14
4.1.2 Silicon.....	16
4.2 Coulomb’s Law.....	17
4.3 Energy Conservation.....	18
4.4 Probability Algorithm.....	19
4.4.1 Atomic structure.....	19
4.4.2 Molecular structure.....	19
4.5 Getting used to GUIDE.....	21
4.5.1 GUI M-file.....	21
4.5.2 Opening Function Callback.....	22
4.5.3 Output Function Callback.....	24
4.5.4 Callbacks.....	25
4.6 Callback Functions.....	27
4.6.1 Types of callback.....	27
4.7 Creating Graphical User Interfaces (GUIs) using GUIDE.....	29
4.7.1 Designing steps.....	30
4.8 Properties of GUI component.....	32
<b>Chapter 5: Conclusion &amp; Recommendations</b> .....	<b>34</b>
5.1 Conclusion.....	34
5.2 Recommendations.....	35
<b>References</b> .....	<b>36</b>
<b>Appendices</b> .....	<b>37</b>
Appendix 1: MATLAB M-file.....	37

## LIST OF FIGURES

Figure 1: Monte Carlo simulation program .....	3
Figure 2: Repulsive scattering by a point particle.....	8
Figure 3: Plum pudding model.....	9
Figure 4: Shapes of some AB <sub>2</sub> molecules.....	10
Figure 5: Shapes of some simple AB <sub>3</sub> molecules.....	11
Figure 6: Five fundamental geometries of AB <sub>n</sub> molecules.....	11
Figure 7: Project methodology.....	13
Figure 8: Crystal structure of graphite .....	15
Figure 9: Face-centered cubic (FCC) crystal structure.....	16
Figure 10: Setting a limit on the radius of a nucleus.....	18
Figure 11: Flowchart of the events of electron-solid interaction .....	20
Figure 12: The resulting GUI.....	23
Figure 13: GUIDE Quick Start window.....	30
Figure 14: GUIDE Layout Editor window.....	30
Figure 15: GUI for the simulation in “Create Mode”.....	31
Figure 16: GUI for the simulation in “Run Mode”.....	31



## LIST OF EQUATIONS

Atomic mass number

$$A = Z + N$$

Electrostatic force

$$F = \frac{kq_1q_2}{r^2}$$

Energy conservation

$$E_i = \frac{1}{2}mv^2$$

$$E_f = \frac{k(Ze)q}{d}$$

$$r = (1.2 \times 10^{-15} \text{ m}) A^{\frac{1}{3}}$$

## ABBREVIATIONS & NOMENCLATURE

MC – Monte Carlo

FCC – Face-Centered Cubic

GUI – Graphical User Interface

GUIDE – Graphical User Interface Design Environment

# CHAPTER 1

## INTRODUCTION

### 1.1 Background of Study

Electrons have both wave and particle properties. Its wave properties enable a beam of electron to behave like a beam of radiation. Electrons have small wavelength when compared to other light microscopes, but it can still interact with a given material due to its electric charge. Its wavelength is dependent on its energy, thereby it can be adjusted accordingly by controlling its accelerating field.

Electrons can be generated by thermionic discharge, where thermal vibrational energy overcomes the electrostatic force holding the electrons from the surface of a material. This process happens in the same manner as the cathode in cathode-ray tube. Electrons can be accelerated by an electric field and focused by electrical and magnetic field onto a sample. The resolution is far better compared to other light microscope due to its small wavelength. Diffraction effect is almost negligible even at nanoscale, enabling better resolution to be achieved. [1]

However, when an electron interacts with solid, scattering effects could cause electrons to reach areas that are not supposed to be exposed to the electrons. Therefore a study on the electron-solid interaction to predict the behavior of electrons is essential.

Electron beam is also used in lithography, which is a method used to fabricate semiconductor devices. The process involves transferring a pattern from a photomask (reticle) to the surface of a material. [2]

## **1.2. Problem Statement**

Electron-solid interaction is mainly used for photo imaging and lithography. Experimentation is usually done to see how the electrons actually interact with solid given various different parameters such as the firing energy of the electron, the types of solid used and the thickness of the solid. However, there are limitations to how much these experiments can provide data of the interaction. Firstly, the experiment is exposed to many external influences such as temperature. The experiment also cannot provide data in the shortest time since it usually takes up a lot of time to achieve the required result. Also, the conventional method needs a lot of funding to be carried out.

The current trend nowadays in researches is to develop a simulation program to replace the conventional method of experimentation. By doing so, all the workings will be done by a computer which uses algorithm to calculate and predict the outcome of a process. The electron-solid interaction can be simulated by a computer to see and predict the outcome of the interaction. This is more appropriate since the data can be saved and used for future references, in order to keep on improving the simulation program. It is not clear whether the results of the simulation are accurate but the program can be improved from time to time.

### **1.2.1 Problem Identification**

An actual experimentation on the electron-solid interaction would take a lot of time and the result may be inconclusive. To overcome this problem, a simulation program is used. The advantages of using a simulation program:

- Probability calculation is used in the program; therefore the electron characteristic can be studied with ease compared to the conventional method.
- A simulation program requires less cost compared to the conventional method
- The result from the simulation program can be used as a reference for the actual experimentation process

## 1.2.2 Significance of the Project

Currently, there are other simulation programs created for the studying the electron-solid interaction. One such example is the Monte Carlo simulation program.

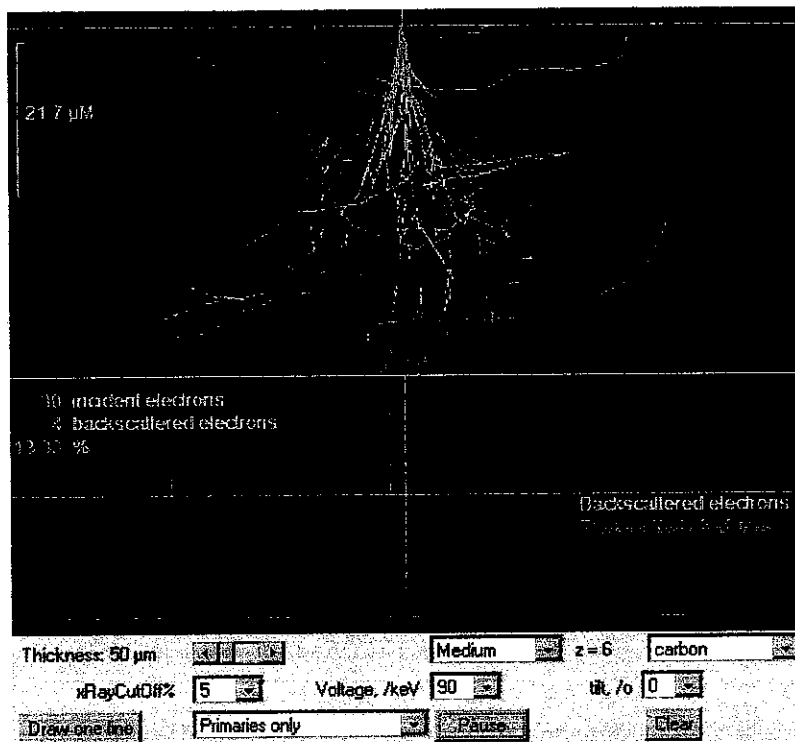


Figure 1: Monte Carlo simulation program [3]

For this project, it is desirable to create a much more comprehensive program compared to the existing one. The database of materials in the periodic table is always being updated from time to time as new discoveries are made. One of the major targets for this project is to include a comprehensive database. Furthermore, other functions or processes which occur during the interaction can be included in the development of the program.

### **1.3 Objective and Scope of Study**

The objective of this project is to:

- Study the processes involved in electron-solid interaction
- Collect database of certain element
- Derive probability equations to predict electron behavior during interaction
- Create a workable simulation program

#### **1.3.1 The Relevancy of the Project**

Competition makes for a faster development process. With this phrase in mind, the creation of a new program even though there are already other existing programs, will indirectly help to promote and fasten the research in electron field. The project itself is then considered as relevant in terms of aiding the development of new knowledge in electron behavior.

#### **1.3.2 Feasibility of the Project within Scope and Time Frame**

The FYP course is a two semester course. For this project, the first semester was dedicated to collecting data and research material which will be used in the simulation program. The second semester was reserved to develop the actual simulation program with the data and research material collected in the first semester. As long as this time frame is followed, the project can be completed within the allocated time.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Electron-Solid Interaction

A phenomenon which is associated with electron-solid interaction is electron scattering. When an electron passes through a solid, the electron can be scattered according to the structure of the solid. The electron may be scattered:

- Not at all
- Single scattering  
The electron undergoes only one scattering event as it passes through a specimen. It is the simplest scattering process.
- Plural scattering  
The electron undergoes more than one scattering event but less than 20 as it passes through a specimen.
- Multiple scattering  
The electron undergoes more than 20 scattering events as it passes through a specimen. This would only occur on rare occasion such as when very thick or dense specimens are used. [3] [4]

Each scattering event might be elastic or inelastic. Elastic event happens if a negligible amount of energy is lost by the primary electron in the process while for the inelastic event, a significant amount of energy is lost by the primary electron. The electron can be either forward scattered or back scattered.

A forward scattered electron is a high energy primary electron which suffers small angle ( $< 90^\circ$ ) scattering and re-emerges from the entry surface of a specimen. This is the most likely type of scattering to occur. A backscattered electron is a high energy primary electron which suffers large angle ( $> 90^\circ$ ) scattering and re-emerges from the entry surface of a specimen. [3] [4]

Various data are collected to visualize the simulation of the electron-solid interaction. The most important data that are needed is what happens during the interaction, whether the electron will be back-scattered or forward-scattered. The resulting effect of the interaction comes later.

## **2.2 Monte Carlo Method (Random Numbers and Probability Method)**

Monte Carlo (MC) methods are stochastic techniques which uses random numbers as inputs to a system and probability statistic to examine a problem. This method has been used in many fields such as economics and nuclear physics. The way that this method is applied varies from one field to another, but in a general view, any usage of random numbers to examine a problem can be called as Monte Carlo method.

This method is usually used to solve complex problems which require a high number of iteration to get the end result. For the project, it is desired to solve equations which describe the interactions for hundreds of thousands of atoms. By using Monte Carlo method, a large system can be modeled using random configurations and the data can be used to describe the system. [11]

“Hit and miss” integration is the simplest type of MC method. By assuming that there are only two conditions or variables used in an experiment, the hit and miss integration technique can be used to calculate the probability of occurrence as required.

### 2.3 Properties of Nuclei

Each and every element has its own properties that characterize its structure. One easy way to recognize an element is by its nucleus structure. Inside nuclei, there are protons and neutrons, which are collectively called as nucleons. Proton is an electrically positive charge which is about 1836 times heavier than an electron. Neutron is an electrically neutral charge with its mass slightly higher than a proton.

$$A = Z + N$$

A = Mass number; total number of nucleons in a nucleus

Z = Atomic number; number of protons in a nucleus

N = Neutron number; number of neutrons in a nucleus

Nuclei are characterized by the number and type of nucleons they contain. From the formula above, the mass number is the sum of the atomic number and the neutron number. [13]



## 2.4 Rutherford Scattering

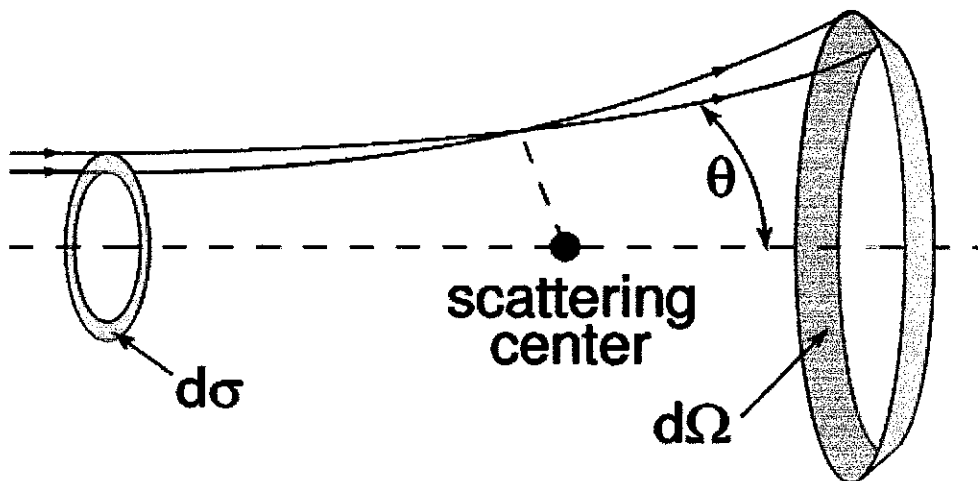


Figure 2: Repulsive scattering by a point particle [5]

For further understanding of the electron-solid interaction process, a comparison can be made to Rutherford scattering.

The experiment was done by Hans Geiger and Ernest Marsden in 1909 under the direction of Rutherford. The foil experiment was done by firing a beam of alpha particles (helium nuclei) at layers of gold leaf which is only a few atoms thick.

At that time, the atom is proposed to be analogous to a plum pudding model by J.J. Thompson. The negative charges (the plum) is spread throughout a positive sphere (the pudding). If this model is correct, the fired alpha beam should only be deflected by small angles as it passed through the gold leaf.

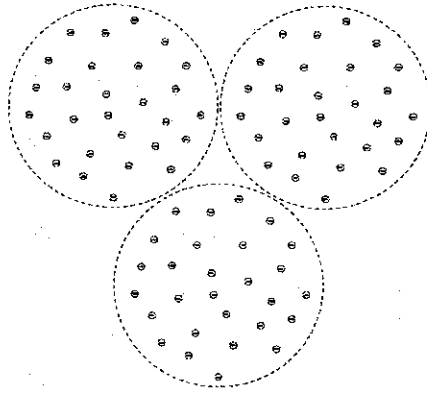


Figure 3: Plum pudding model [6]

However, the experiment done by Rutherford showed otherwise. The results showed that around 1 in 8000 alpha particles were deflected by very large angle (more than  $90^\circ$ ). Rutherford concluded that positively charged region is concentrated at the center of the atom while being surrounded by negatively charged electrons. The alpha particle is positively charged, therefore when it approached the nucleus, it was repelled strongly enough to be rebounded at a high angle. The small number of repelled alpha particles shows that the size of the nucleus is small. [6] [14]

## 2.5 Molecular Geometries and Bonding Theories

Molecular geometries and bonding theories is studied so that the basic structure of molecules for the selected materials can be analyzed with better understanding.

The overall shape of a molecule is determined by its bond angle and its bond length. Bond angle is the angle made by the lines joining the nuclei of the atoms in the molecule while bond length is the distance between the nuclei of the atoms in the molecule.

The general bonding of a molecule is the bonding of a central atom to two or more atoms of the same type. The general formula for this bonding is  $AB_n$ , where  $n$  number of B atoms is bonded to the central A atom.

From the  $AB_n$  formula, several general shapes can be observed. This includes  $AB_2$  and  $AB_3$  molecules. For  $AB_2$  molecules, it can be either linear (bond angle =  $180^\circ$ ) or bent (bond angle  $\neq 180^\circ$ ). Such examples would be  $CO_2$  (linear) and  $SO_2$  (bent).  
[13]

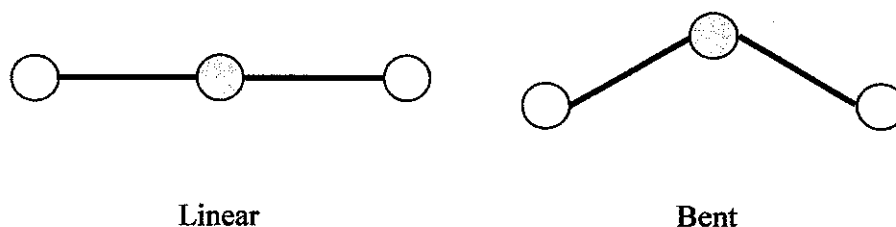


Figure 4: Shapes of some  $AB_2$  molecules [13]

For AB<sub>3</sub> molecules, there are two common shapes which are trigonal planar and trigonal pyramidal. Trigonal planar has the A atom which lies in the same plane as the B atoms, while trigonal pyramidal has the A atom which lies above the plane of the B atoms. These two shapes place the B atoms at the corner of an equilateral triangle.

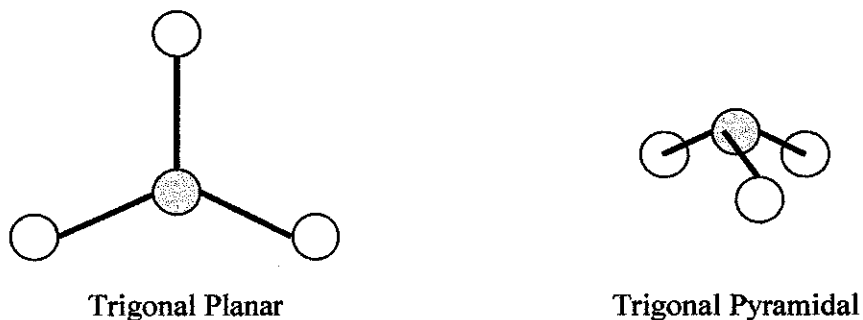


Figure 5: Shapes of some simple AB<sub>3</sub> molecules [13]

The shape for any particular AB<sub>n</sub> molecule can usually be derived from one of the following five basic geometric structures:

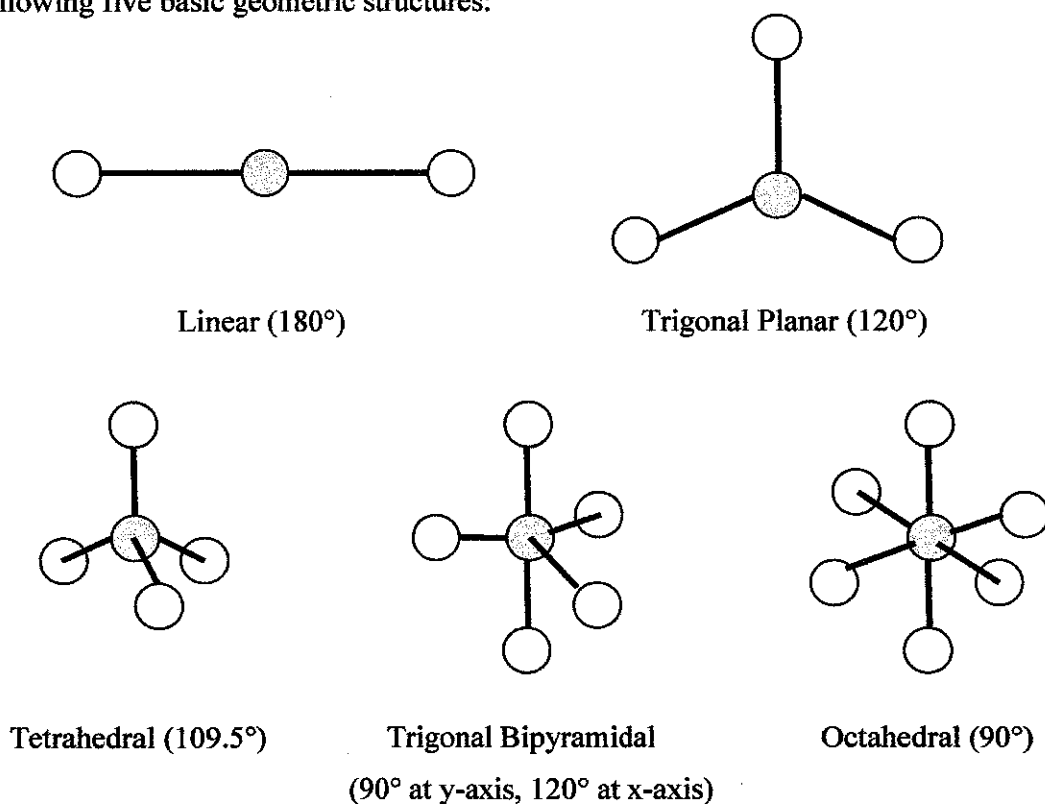


Figure 6: Five fundamental geometries of AB<sub>n</sub> molecules [13]

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 Procedure Identification**

In order to grasp the main concept concerning the project, a basic understanding of the structure of materials is required. For such purpose, graphite and silicon is selected as the material to be used in the simulation. Their crystal structures and parameters such as the radius of the atoms are collected. A research is carried out with respect to the electron-solid interaction process. The electron scattering event which happens in the process is studied and identified. The forces which apply to the electron and atoms of the solid is observed and applied to derive a suitable probability algorithm.

The simulation program coding comes after the initial procedure is done. A basic understanding of the intended program; MATLAB, is required to implement a working simulation program. By taking an example of the existing simulation program; Monte Carlo simulation, a new program will be created. The simulation is done using GUIDE (Graphical User Interface Design Environment) which is a tool under MATLAB used to create GUIs.

Evaluation and troubleshooting completes the overall procedure needed to complete the project. During this stage, any defects and correction are to be done to produce the best result.

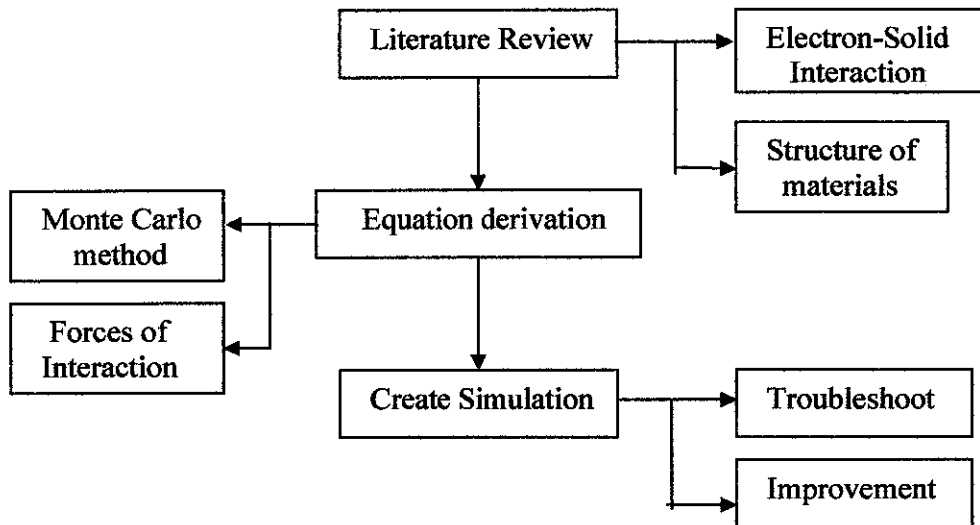


Figure 7: Project methodology

### 3.2 Tools

For this project, the following tools used:

- **MATLAB software**  
 MATLAB is a high-performance language for technical computing. It integrates computation, visualization and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. The software is used to create the simulation program.
- **GUIDE tools**  
 GUIDE (Graphical User Interface Design Environment) provides the tool to create the GUI (Graphical User Interface) for the simulation.
- **Computer hardware**  
 A high end computer is used to run the MATLAB software.

## **CHAPTER 4**

### **DATA AND DISCUSSION**

#### **4.1 Structure and Properties of materials**

Earlier in the project, various materials were to be included in the simulation program. However, due to time constrain and relevancy of the selected material, only two materials will be selected for simulation for the time being. As for further development of the simulation program, additional material can be added from time to time. Graphite and silicon were chosen due to their applicability as substrates in lithography.

##### **4.1.1 Graphite**

Graphite is an allotrope of carbon, besides the other carbon allotrope such as diamond. It is an electrical conductor, which is quite unique for a carbon allotrope. Also, graphite has the most stable form of solid carbon and can be considered as the highest grade of coal. It is mainly used for:

- Marking material (lead) in common pencils
- Reentry shields for missile nosecones
- Carbon fiber reinforced plastic and heat-resistant composites such as RCC reinforced carbon-carbon
- Firestops, which in particular are the plastic pipes and gaskets fitted around the parameter of a fire door.

Each carbon atom is covalently bonded to three other carbon atoms which form into hexagonal structures. These structures exist in layers where the layers are connected together by weak forces called van der Waals forces.

The unit cell dimensions are  $a = b = 2.456 \text{ \AA}$  (245.6 pm) and  $c = 6.694 \text{ \AA}$  (669.4 pm). The carbon-carbon bond length in the bulk form is  $1.418 \text{ \AA}$  (141.8 pm), and the interlayer spacing is  $c/2 = 3.347 \text{ \AA}$  (334.7 pm).

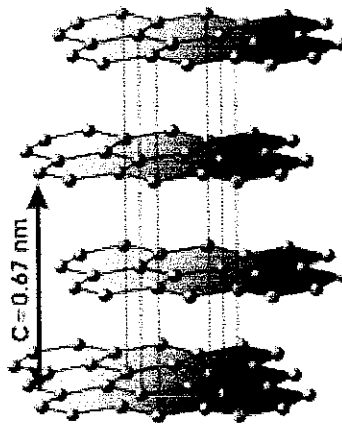


Figure 8: Crystal structure of graphite [7]

The bond between the atoms within a layer is stronger than the bond of diamonds, but the force between the layers is weak. Therefore, layers of it can slip over each other making it soft. [7]



### 4.1.2 Silicon

Silicon has the chemical symbol Si with atomic number of 14. It is a tetravalent metalloid which is less reactive compared to carbon. Silicon exists as pure element in nature, but it is widely distributed in dusts and also various forms of silicon dioxide or silicate. Silicon has many industrial usages such as:

- Principal component of most semiconductor devices such as integrated circuits (microchips)
- Main ingredient in glasses, cements and ceramics. It is also a component of silicones, a class-name for various synthetic plastic substances.

The material/crystal structure of silicon is face-centered cubic, with an atomic radius of 110 pm, covalent radius of 111 pm and Van der Waals radius of 210 pm.

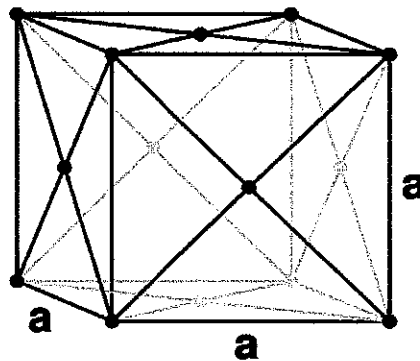


Figure 9: Face-centered cubic (FCC) crystal structure [9]

The face centered cubic lattice has lattice points on the faces of the cube, giving a total of 4 lattice points per unit cell. [8]

$$\left[ \left( \frac{1}{8} \right) \times 8 + \left( \frac{1}{2} \right) \times 6 \right] = 4 .$$

## 4.2 Coulomb's Law

Electric charges exist everywhere, even in our own body. Every atom contains both positive and negative charges which are held together by an attractive force. In an atom, the negative charge is represented by an electron while the positive charge is represented by a nucleus which contains both neutrons and protons. Suppose that an electron of magnitude  $q_1$  is separated by a distance  $r$  from a nucleus of magnitude  $q_2$ . Coulomb's law states that the magnitude of the electrostatic force between these charges is proportional to the product of the charges and inversely proportional to the square of the distance between them:

$$F = \frac{kq_1q_2}{r^2}$$

where  $q_1, q_2$  = charges

$r$  = distance between  $q_1$  and  $q_2$

$k = 8.99 \times 10^9 \text{ Nm}^2/\text{C}^2$

In the atom structure, the negative charge (electron) is a moving particle while the positive charge (nucleus) is a stationary particle. [12]

### 4.3 Energy Conservation

Using energy conservation, the size of nucleus can be estimated. Consider a case where a particle of charge  $+q$  and mass  $m$  approaching a nucleus of charge  $+Ze$  with a speed of  $v$ . The approach is assumed to be head-on. At some distance  $d$ , from the centre of the nucleus, the incoming particle comes to rest instantaneously before turning around. It can be concluded that the radius of the nucleus is less than  $d$ .

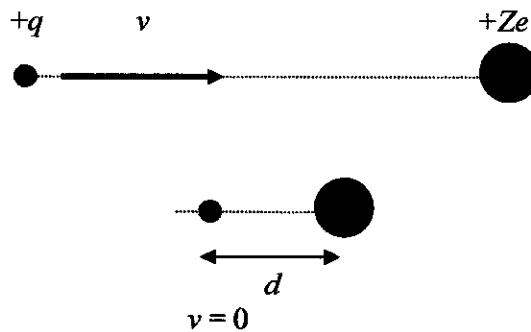


Figure 10: Setting a limit on the radius of a nucleus [12]

Expression for the initial energy of the system:

$$E_i = \frac{1}{2}mv^2$$

Expression for the final energy of the system:

$$E_f = \frac{k(Ze)q}{d}$$

Further research shows that the average radius of a nucleus of mass  $A$  is given approximately by the following expression:

$$r = (1.2 \times 10^{-15} \text{ m}) A^{\frac{1}{3}}$$

## 4.4 Probability Algorithm

The atomic structure and molecular structure is discussed by stages.

### 4.4.1 Atomic Structure

It is required to identify the nuclei and electrons structure in an atom as this will affect the chances of the electron passing through or being either forward-scattered or back-scattered. The diameter of the atom can be calculated by using the energy conservation equation:

$$r = (1.2 \times 10^{-15} \text{ m}) A^{\frac{1}{3}}$$
$$d = r \times 2$$

With the information of the atomic structure and its diameter, the probability of interaction between an electron and a material can be predicted. The probability algorithm is actually a ratio of the electron and nuclei position in an atom along with the whole diameter of the atom.

### 4.4.2 Molecular Structure

The molecular structure determines the electron-solid interaction as a whole compared to the electron structure level. The atoms arrangement in the molecule affects the probability of the interaction. The first stage of interaction is when the electron hits or passes through the atoms in the molecular level. If the electron hits the atom, then the atomic level comes up. The electron is then predicted whether it will pass through the atom or hits either the electron or nuclei of the atom. The resulting effect of the interaction is then observed, whether the electron would be back-scattered or forward scattered.

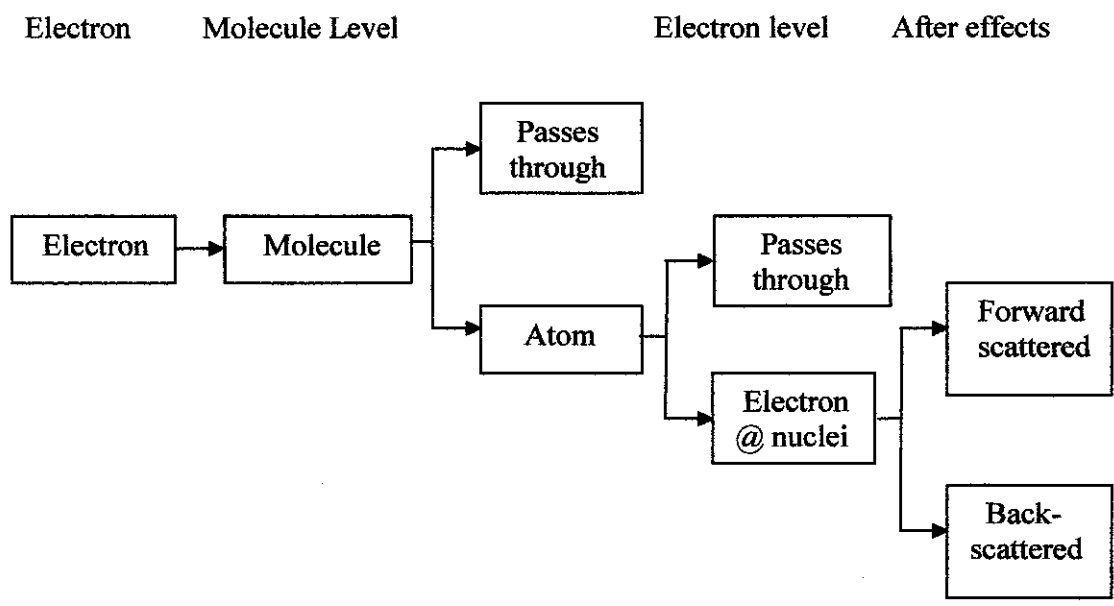


Figure 11: Flowchart of the events of electron-solid interaction

## **4.5 Getting used to GUIDE**

MATLAB is a powerful software which is widely used for its mathematical and modeling functions. One of the tools used is the GUIDE tools, which is used to create GUI's for simulation program. GUI stands for Graphical User Interface. The GUIDE tool makes it easy for designers to design the GUI to their preferences.

### **4.5.1 GUI M-file**

The GUIDE tool generates its own GUI M-file which controls the GUI and determines how it responds to a user's action, such as pressing a push button or selecting a menu item.

The M-file contains all the code needed to run the GUI, including the most commonly used callbacks for the GUI components. A callback is a function that executes when a user performs a specific action such as clicking a push button, selecting a menu item, or pressing a keyboard key, or when a component is created or deleted. Each component and menu item has properties that specify its callbacks. The callbacks, which are subfunctions of the M-file that initially consist of a framework that contains just a function definition, have to be programmed to control the operation of the GUI. The M-file also contains some initialization code, an opening function callback, and an output function callback. [15]

## 4.5.2 Opening Function Callback

The `opening` function is the first callback in every GUI M-file. It can be used to perform tasks that need to be done before the user has access to the GUI, for example, to create data or to read data from an external source. The code in the opening function is executed just before the GUI is made visible to the user, but after all the components have been created.

GUIDE names the opening function with the name of the M-file prefixed to `_OpeningFcn`. For example, the M-file is named `simple_gui.m`, so that the opening function is named `simple_gui_OpeningFcn`.

GUIDE will auto generate codes for the GUI. However, for the callback to function, codes have to be added. For example:

```
% --- Executes just before simple_gui is made visible.
function simple_gui_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to untitled (see VARARGIN)
```

```
% Create the data to plot
handles.peaks=peaks(35);
handles.membrane=membrane;
[x,y] = meshgrid(-8:.5:8);
r = sqrt(x.^2+y.^2) + eps;
sinc = sin(r)./r;
handles.sinc = sinc;
handles.current_data = handles.peaks;
surf(handles.current_data)
```



Auto generated



Added codes

Added codes:

- The first six executable lines create the data using the MATLAB functions `peaks`, `membrane` and `sinc` to generate the data.
- The next line, `handles.current_data = handles.peaks`, sets the `current_data` field of the `handles` structure equal to the data for `peaks`. The value of `handles.current_data` changes each time a user selects a different plot from the pop-up menu
- The last line displays the surf plot for `peaks`, which appears when the GUI is first opened.
- GUIDE automatically generates two more lines of code in the opening function, which follow the added codes:

`handles.output = hObject` saves the handle to the GUI for later access by the output function. While this command is not necessary in this example, it is useful if you want to return the GUI handle to the command line.

`guidata(hObject,handles)` = saves the handles structure [15]

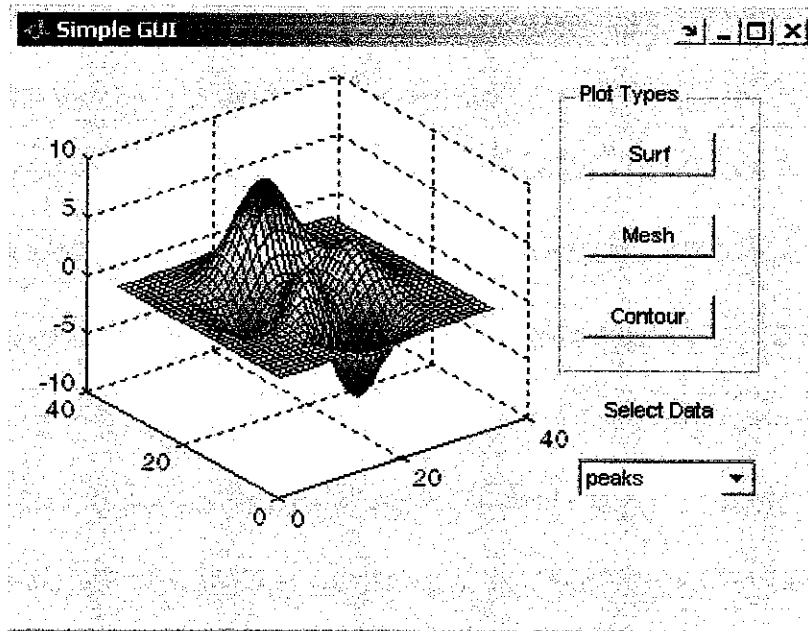


Figure 12: The resulting GUI [15]



### 4.5.3 Output Function Callback

The output function returns output arguments to the command line. This is particularly useful when a variable is needed to be returned to another GUI.

GUIDE generates the following lines of code in the output function:

```
% --- Outputs from this function are returned to the command line.
function varargout = my_gui_OutputFcn(hObject, eventdata,
handles)
% Get default command line output from handles structure
varargout{1} = handles.output;
```

For example, to make the GUI return a different output, such as pressing a push button:

- The command “uiwait;” is added to the opening function to make the M-file halt execution until a user activates a component in the GUI.
- For each component of the GUI where a user response is expected, the callback is programmed to update the value of handles.output, and execute “uiresume”. [15]

#### 4.5.4 Callbacks

When a user activates a component of the GUI, the GUI executes the corresponding callback. The name of the callback is determined by the component's Tag property and the type of callback. For example, the name of the Surf push button callback is `surf_pushbutton_Callback`.

Push button callback:

- Each of the push buttons creates a different type of plot using the data specified by the current selection in the pop-up menu. Their callbacks get data from the handles structure and then plot it.

```
% --- Executes on button press in surf_pushbutton.  
function surf_pushbutton_Callback(hObject, eventdata, handles)  
% hObject    handle to surf_pushbutton (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)
```

Auto generated

```
% Display surf plot of the currently selected data  
surf(handles.current_data);
```

Added code

Pop-up menu callback:

- The pop-up menu enables users to select the data to plot. Every time a user selects one of the three plots, the pop-up menu callback reads the pop-up menu Value property to determine what item is currently displayed and sets `handles.current_data` accordingly. [15]

```
% --- Executes on selection change in data_popup.  
function plot_popup_Callback(hObject, eventdata, handles)  
% hObject    handle to surf_pushbutton (see GCBO)  
% eventdata  reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)
```

Auto generated

```
val = get(hObject, 'Value');  
str = get(hObject, 'String');  
switch str(val);  
case 'peaks' % User selects peaks  
    handles.current_data = handles.peaks;  
case 'membrane' % User selects membrane  
    handles.current_data = handles.membrane;  
case 'sinc' % User selects sinc  
    handles.current_data = handles.sinc;  
end  
guidata(hObject, handles)
```

Added codes

## 4.6 Callback Functions

The callbacks control the operation of the GUI. Therefore, it is important that the callbacks functions are thoroughly define to understand its usage.

### 4.6.1 Types of Callback

Callback properties for all graphics objects (define callback routines):

- “ButtonDownFcn”  
MATLAB executes this callback when users click the left mouse button and the cursor is over the object or within a five-pixel border around the object.
- “CreateFcn”  
MATLAB executes this callback when creating the object.
- “DeleteFcn”  
MATLAB executes this callback just before deleting the object.

Callback properties for figures:

- “CloseRequestFcn”  
MATLAB executes the specified callback when a request is made to close The figure (by a close command, by the window manager menu, or by quitting MATLAB).
- “KeyPressFcn”  
MATLAB executes the specified callback when users press a key and the cursor is within the figure window and no component has focus.
- “ResizeFcn”  
MATLAB executes the specified callback routine when users resize the figure window.
- “WindowButtonDownFcn”  
MATLAB executes the specified callback when users click the mouse button and the cursor is within the figure, but not over an enabled uicontrol.

- “WindowButtonMotionFcn”  
MATLAB executes the specified callback when users move the mouse button within the figure window.
- “WindowButtonUpFcn”  
MATLAB executes the specified callback when users release the mouse button, after having pressed the mouse button within the figure.

Callbacks for specific components:

- “Callback”  
MATLAB executes the specified callback when a user activates a user interface control (uicontrol) or menu (uimenu) object, for example, when a user presses a push button or selects a menu item.
- “KeyPressFcn”  
MATLAB executes the specified callback when a user presses a key and the callback's component has focus.
- “ResizeFcn”  
MATLAB executes the specified callback routine when users resize a panel (uipanel) or button group (uibuttongroup) object.
- “SelectionChangeFcn”  
MATLAB executes the specified callback routine when a user selects a different radio button or toggle button in a button group component. [15]

## **4.7 Creating Graphical User Interfaces (GUIs) using GUIDE**

GUIDE, the MATLAB Graphical User Interface development environment, provides a set of tools for creating graphical user interfaces (GUIs). These tools greatly simplify the process of designing and building GUIs. The GUIDE tools can be used to

- **Lay out the GUI**

Using the GUIDE Layout Editor, the GUI layout can be easily done by clicking and dragging GUI components such as panels, buttons, text fields, sliders, menus, and so on into the layout area.

- **Program the GUI**

GUIDE automatically generates an M-file that controls how the GUI operates. The M-file initializes the GUI and contains a framework for all the GUI callbacks, which are the commands that are executed when a user clicks a GUI component. Using the M-file editor, codes can be added to the callbacks to perform the required functions. [15]

### 4.7.1 Designing Steps

1. The GUIDE program is initiated by typing “GUIDE” in the command window of MATLAB.
2. The GUIDE Quick Start window appears after the command has been entered. The option “Blank GUI (default)” is selected.

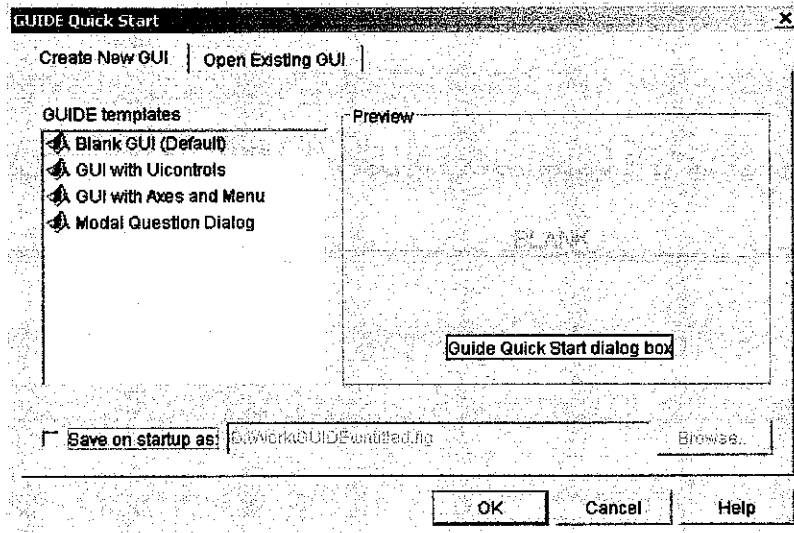


Figure 13: GUIDE Quick Start window [15]

3. The GUIDE Layout Editor appears and the GUI designing stage starts. The component palettes are dragged over into the layout area and placed accordingly as designed earlier.

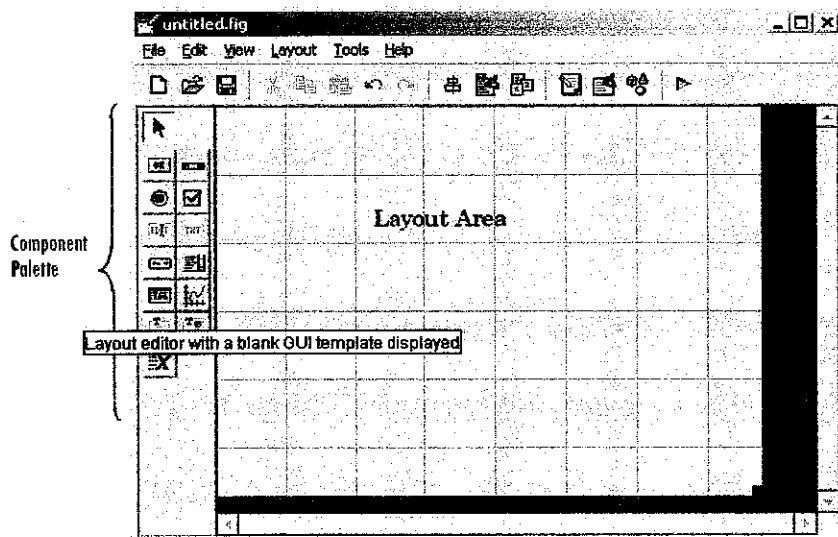


Figure 14: GUIDE Layout Editor window [15]

4. The final product design is in “Create Mode”. The following figure shows the initial design of the GUI in Create Mode.

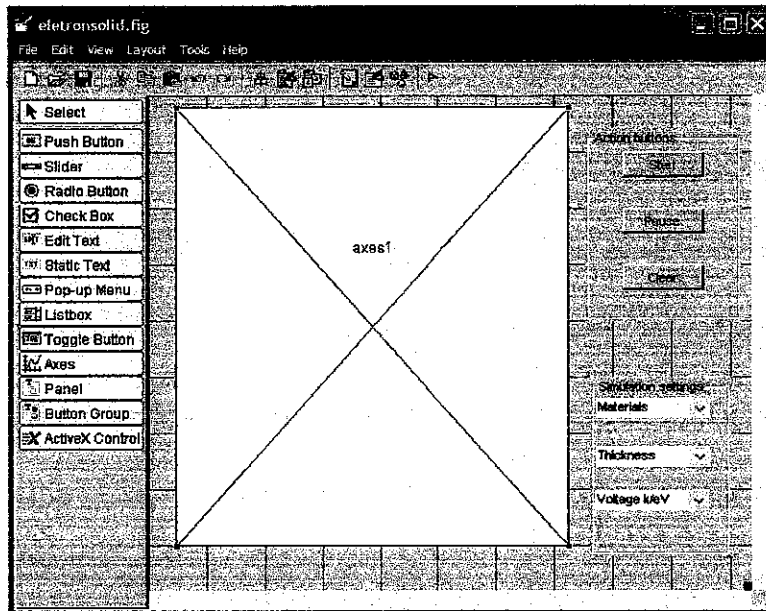


Figure 15: GUI for the simulation in “Create Mode”

5. The following figure shows the GUI in “Run Mode”

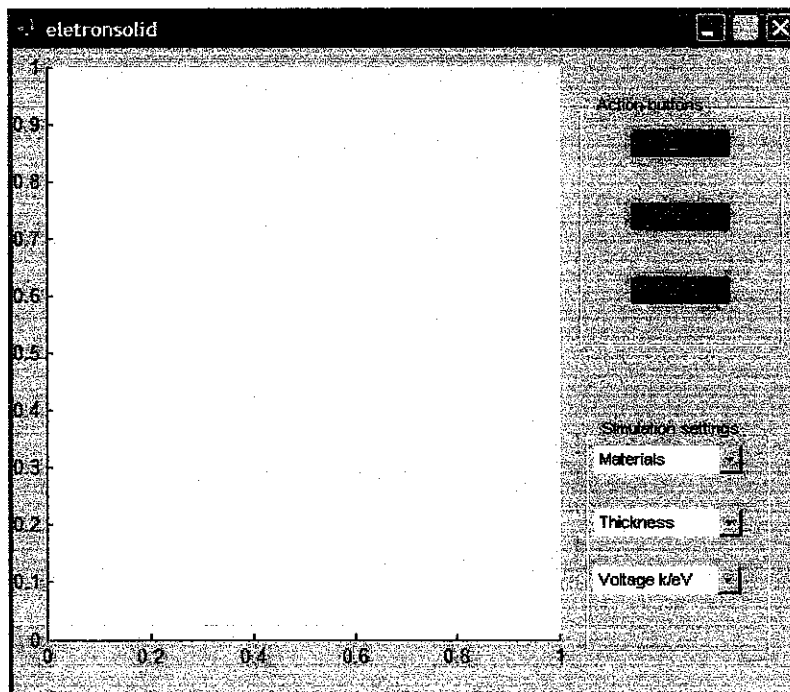


Figure 16: GUI for the simulation in “Run Mode”



## 4.8 Properties for GUI Component

Property Inspector is used to set the properties for each GUI components. The Property Inspector enables the setting of properties of the components in a layout. It provides a list of all settable properties and displays the current value. Each property in the list is associated with an editing device that is appropriate for the values accepted by the particular property.

There are many types of properties for each component:

- **Name Property**  
The value of a figure's "Name Property" is the title that displays at the top of the GUI. Initially, GUIDE sets the value of Name Property to the name of the FIG-file. The value of the Name Property can be renamed to the desired string once the GUI is saved.
- **Title Property**  
A panel's "Title property" controls the title that appears at the top or bottom of the panel. The position of the title can be rearranged by using the Title Position property.
- **String Property**  
The string property controls the label in some of the user interface controls such as push buttons, static text and pop-up menus.
- **Callback Property**  
A component can have many callback properties, but the most common one is the "Callback Property". The code provided for the Callback Property performs the primary work of the component. It executes, for example, when a user presses a push button, moves a slider, or selects a menu item.

- **Tag Property**

The Tag property provides a string as a unique identifier for each component. GUIDE uses this identifier to construct unique callback names for the different components in the GUI. When a component is added to a layout, GUIDE sets the value of Tag to a default string such as *pushbutton1*. If the component has a Callback property, GUIDE also sets the value of Callback to the string *%automatic*.

When the GUI is saved or run, GUIDE generates an M-file that includes stubs for the Callback functions for each component that has one. GUIDE creates a unique function name for each Callback function in the M-file by prefixing the value of the Tag property to the string *\_Callback*, for example; *pushbutton1\_Callback*. GUIDE also changes the value of the Callback property to a string that is a calling sequence for the callback. For example, if the name of the GUI M-file is *simple\_gui*, the new value of the Callback property becomes

```
untitled(pushbutton1_Callback',gcbo,[],guidata(gcbo))
```

The value of Tag can be redefined to be more descriptive, but the value of each Tag property must be unique for a given GUI. [15]

## CHAPTER 5

### CONCLUSION & RECOMMENDATION

#### 5.1 Conclusion

A series of internal and external forces accumulates when an electron hits a solid, particularly the atom of the solid. The electrostatic force between the negative (electron) and positive charges (nucleus) attracts each other. Also, when negative charges (fired electron and electron of the atom) meet each other, they repel. Magnetic force and electric field within the atom focuses the entry of the fired electron on the solid.

Different parameters such as energy of the fired electron and the thickness of the materials affect the outcome of the interaction. Larger energy will cause the electron to have multiple scattering. The same thing applies when thick material is used. More than 20 scattering will occur in both cases, although the resulting pattern is different for graphite and silicon.

Simulation program is much more accessible and easier to use compared to conventional experiment methods. With the advancement of technologies, simulation programs can be further enhanced and improved for future usage.

## **5.2 Recommendations**

The simulation only includes two materials. For future work, it is desirable to include more materials in the database of the simulation. Also, the simulation program can be further fine-tuned to give a better simulation result. The GUI of the program can be added with other parameters such as effects on secondary electron and stopping power of the electron.

## REFERENCES

- [1] [http://www.en.wikipedia.org/wiki/Transmission\\_electron\\_microscope](http://www.en.wikipedia.org/wiki/Transmission_electron_microscope)
- [2] <http://en.wikipedia.org/wiki/Photolithography>
- [3] [http://www.matter.org.uk/tem/electron\\_scattering.htm](http://www.matter.org.uk/tem/electron_scattering.htm)
- [4] <http://www.matter.org.uk/glossary/index>
- [5] [http://en.wikipedia.org/wiki/Rutherford\\_scattering](http://en.wikipedia.org/wiki/Rutherford_scattering)
- [6] <http://www.uwsp.edu/Chemistry/chemproj/fipse>
- [7] <http://en.wikipedia.org/wiki/Graphite>
- [8] <http://en.wikipedia.org/wiki/Silicon>
- [9] [http://en.wikipedia.org/wiki/Face\\_centered\\_cubc](http://en.wikipedia.org/wiki/Face_centered_cubc)
- [10] Handbook of Microlithography, Micromachining and Microfabrication; Volume 1: Microlithography; P. Rai-Choudhury; SPIE
- [11] An Introduction to MC Methods, Joy Woller, University of Nebraska-Lincoln, 1996
- [12] Physics; James S. Walker; Prentice Hall
- [13] Chemistry: The Central Science; Theodore L. Brown, H. Eugene Lemay, Bruce E. Bursten, Julia R. Burdge; Prentice Hall
- [14] An Introduction to the Physics of Nuclei and Particles; Richard A. Dunlap; Thomson, Brooks/Cole
- [15] MATLAB help documents

## APPENDICES

### Appendix I: MATLAB M-file

```
function varargout = eletronsolid(varargin)
% ELETRONSOLID M-file for eletronsolid.fig
%   ELETRONSOLID, by itself, creates a new ELETRONSOLID or raises the
existing
%   singleton*.
%
%   H = ELETRONSOLID returns the handle to a new ELETRONSOLID or the
handle to
%   the existing singleton*.
%
%   ELETRONSOLID('CALLBACK',hObject,eventData,handles,...) calls the local
function named CALLBACK in ELETRONSOLID.M with the given input
arguments.
%
%   ELETRONSOLID('Property','Value',...) creates a new ELETRONSOLID or
raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before eletronsolid_OpeningFunction gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to eletronsolid_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help eletronsolid

% Last Modified by GUIDE v2.5 11-Oct-2006 13:28:07

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
                  'gui_Singleton', gui_Singleton, ...
                  'gui_OpeningFcn', @eletronsolid_OpeningFcn, ...
                  'gui_OutputFcn', @eletronsolid_OutputFcn, ...
                  'gui_LayoutFcn', [], ...
                  'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before eletronsolid is made visible.
function eletronsolid_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to eletronsolid (see VARARGIN)

% Choose default command line output for eletronsolid
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes eletronsolid wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = eletronsolid_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu1 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu2 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu2

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.

```



```

if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu3 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from popupmenu3

% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```