User’s Manual for LAMOCA
A Kinetic Lattice Monte Carlo Simulator for Study Point Defect Mediated Dopant Diffusion in Silicon

Zudian Qin and Scott T. Dunham
Department of Electrical Engineering
University of Washington

October 2002
## CONTENTS

2.3 Output Files ................................................................. 13
  2.3.1 Data File ................................................................. 13
  2.3.2 Status File ............................................................... 14

3 Examples ................................................................. 15
  3.1 Cluster Energy Dilemma .................................................. 15
  3.2 Arsenic Diffusion .......................................................... 16

4 Advanced Topics ......................................................... 18
  4.1 Acceleration to Lamoca .................................................. 18
    4.1.1 Problem ............................................................... 18
    4.1.2 Solution .............................................................. 19
  4.2 Fermi Level Modeling in Atomistic Simulations ...................... 20
    4.2.1 Carrier Distribution: Classical Solution ....................... 21
    4.2.2 Carrier Distribution: Quantum Solution ....................... 21
    4.2.3 Fermi Level .......................................................... 24
Chapter 1

Introduction

1.1 Introduction

Lamoca is a 3D atomistic simulator designed to investigate point defect-mediated dopant diffusion in silicon at elevated process temperatures. Some of its fundamental features include:

- It is based on a stochastic method.
- The domain it operates on is a silicon (diamond) lattice.
- Dopant atoms and point defects (vacancies and interstitialcies) are mapped onto an otherwise perfect silicon lattice.
- Only point defects are mobile; dopant diffuses via interaction with point defects.
- A Point defect diffuses via reaction with its first nearest neighbors (1NNs).
- A vacancy moves through exchanging its position with one of its four 1NNs.
- An interstitialcy moves by completing both a dissociation process that the interstitialcy breaks apart in its current site and an association process that one of the two constituents jumps into one of its 1NNs, resulting in most likely formation of a new interstitialcy.
- The reaction rate of these aforementioned processes are calculated based on the change of the system energy due to the process.
Note that traditional continuum model-based dopant diffusion simulations focus on macroscopic descriptions (e.g., dopant concentration) of the system and seek time evolution of these quantities in most cases by solving time-dependent coupled PDEs.

1.2 Species

In its current version, lamoca is capable of simulating arsenic and boron diffusion in silicon. The species that may exist in the system are defined in Table 1.1.

<table>
<thead>
<tr>
<th>No</th>
<th>Symbol</th>
<th>Defect Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Z</td>
<td>Silicon</td>
<td>A regular silicon atom site</td>
</tr>
<tr>
<td>2</td>
<td>V</td>
<td>Vacancy</td>
<td>A vacant lattice site</td>
</tr>
<tr>
<td>3</td>
<td>z</td>
<td>Interstitialcy</td>
<td>Two silicon atoms sharing a lattice site</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>Arsenic</td>
<td>A substitional arsenic atom site</td>
</tr>
<tr>
<td>5</td>
<td>a</td>
<td>Arsenic_interstitialcy</td>
<td>An arsenic and a silicon atom sharing a lattice site</td>
</tr>
<tr>
<td>6</td>
<td>B</td>
<td>Boron</td>
<td>A substitional born atom site</td>
</tr>
<tr>
<td>7</td>
<td>b</td>
<td>Boron_interstitialcy</td>
<td>A boron and a silicon atom sharing a lattice site</td>
</tr>
</tbody>
</table>

Table 1.1: Species in the system

1.3 Reactions

In lamoca, point defects and dopants diffuse via reactions of point defects with their 1N.Ns. Table 1.2 lists all reactions that may take place inside lamoca simulations and they each are described below as labelled in the table.

1. A vacancy exchanges its position with a neighboring silicon atom.
2. A vacancy exchanges its position with a neighboring substitional arsenic atom.
3. A vacancy exchanges its position with a neighboring substitional boron atom.
1.3. REACTIONS

<table>
<thead>
<tr>
<th>No</th>
<th>Reaction</th>
<th>Reactants</th>
<th>Resultants</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$V+Z \Rightarrow Z+V$</td>
<td>$V,Z$</td>
<td>$Z,V$</td>
</tr>
<tr>
<td>2</td>
<td>$V+A \Rightarrow A+V$</td>
<td>$V,A$</td>
<td>$A,V$</td>
</tr>
<tr>
<td>3</td>
<td>$V+B \Rightarrow B+V$</td>
<td>$V,B$</td>
<td>$B,V$</td>
</tr>
<tr>
<td>4</td>
<td>$z+V \Rightarrow Z+z$</td>
<td>$z,V$</td>
<td>$Z,z$</td>
</tr>
<tr>
<td>5</td>
<td>$z+Z \Rightarrow Z+z$</td>
<td>$z,Z$</td>
<td>$Z,z$</td>
</tr>
<tr>
<td>6</td>
<td>$z+A \Rightarrow Z+a$</td>
<td>$z,A$</td>
<td>$Z,a$</td>
</tr>
<tr>
<td>7</td>
<td>$z+B \Rightarrow Z+b$</td>
<td>$z,B$</td>
<td>$Z,b$</td>
</tr>
<tr>
<td>8</td>
<td>$a+V \Rightarrow A+Z$</td>
<td>$a,V$</td>
<td>$A,Z$</td>
</tr>
<tr>
<td>9</td>
<td>$a+V \Rightarrow Z+a$</td>
<td>$a,V$</td>
<td>$Z,a$</td>
</tr>
<tr>
<td>10</td>
<td>$a+Z \Rightarrow A+z$</td>
<td>$a,Z$</td>
<td>$A,z$</td>
</tr>
<tr>
<td>11</td>
<td>$a+Z \Rightarrow Z+a$</td>
<td>$a,Z$</td>
<td>$Z,a$</td>
</tr>
<tr>
<td>12</td>
<td>$a+A \Rightarrow A+a$</td>
<td>$a,A$</td>
<td>$A,a$</td>
</tr>
<tr>
<td>13</td>
<td>$a+B \Rightarrow A+b$</td>
<td>$a,B$</td>
<td>$A,b$</td>
</tr>
<tr>
<td>14</td>
<td>$b+V \Rightarrow B+Z$</td>
<td>$b,V$</td>
<td>$B,Z$</td>
</tr>
<tr>
<td>15</td>
<td>$b+V \Rightarrow Z+b$</td>
<td>$b,V$</td>
<td>$Z,b$</td>
</tr>
<tr>
<td>16</td>
<td>$b+Z \Rightarrow B+z$</td>
<td>$b,Z$</td>
<td>$B,z$</td>
</tr>
<tr>
<td>17</td>
<td>$b+Z \Rightarrow Z+b$</td>
<td>$b,Z$</td>
<td>$Z,b$</td>
</tr>
<tr>
<td>18</td>
<td>$b+A \Rightarrow B+a$</td>
<td>$b,A$</td>
<td>$B,a$</td>
</tr>
<tr>
<td>19</td>
<td>$b+B \Rightarrow B+b$</td>
<td>$b,B$</td>
<td>$B,b$</td>
</tr>
</tbody>
</table>

Table 1.2: Reactions in the system

5. An interstitialcy hop (dissociation of an interstitialcy and formation of a new one at the neighboring site).

6. An interstitialcy dissociation and an arsenic_interstitialcy formation.

7. An interstitialcy dissociation and a boron_interstitialcy formation.

8. Annihilation of an arsenic_interstitialcy-vacancy pair. The silicon atom hops to the neighboring site.


10. An arsenic_interstitialcy dissociation and formation of an interstitiacy at the neighboring site.
11. An arsenic\_interstitialcy hop (dissociation of an arsenic\_interstitialcy and formation of a new arsenic\_interstitialcy at the neighboring site.

12. Dissociation of an arsenic\_interstitialcy and formation of a new arsenic\_interstitialcy after a silicon atom hops to the neighboring site.

13. Dissociation of an arsenic\_interstitialcy and formation of a boron\_interstitialcy after a silicon atom hops to the neighboring site.


15. Annihilation of an boron\_interstitialcy-vacancy pair. The boron atom hops to the neighboring site.

16. An boron\_interstitialcy dissociation and formation of an interstitiacy at the neighboring site.

17. An boron\_interstitialcy hop (dissociation of an boron\_interstitialcy and formation of a new boron\_interstitialcy at the neighboring site.

18. Dissociation of an boron\_interstitialcy and formation of an arsenic\_interstitialcy after a silicon atom hops to the neighboring site.

19. Dissociation of a boron\_interstitialcy and formation of a new boron\_interstitialcy after a silicon atom hops to the neighboring site.

1.4 Reaction Rates

The reaction rate of each of the reactions described in the previous section is determined by the barrier height that exists between the two sites involved in a reaction. This number is calculated as:

\[ \nu = \nu_0 \exp \left( -\frac{E_m}{k_B T} \right) \exp \left( \frac{E_i - E_f}{2k_B T} \right) \]  (1.1)

where \( \nu_0 \) is a universal constant, \( E_m \) is the unbiased migration barrier, \( E_i \) and \( E_f \) represent the system energy before and after the reaction, respectively.
1.5 System Energy

In lamoca the energy of a system is determined by the system configuration in atomic scale. In other words, impurities, including dopants and point defects, and their arrangements on the lattice define the system energy level. In its current version, lamoca computes two types of energy that add up to the total system energy, binding energy and potential energy.

1.5.1 Binding Energy

Binding energy is a measure of the binding strength among a group of closely coupled dopants and defects. In lamoca binding energy is further classified as pair energy and cluster energy.

- **Pair Energy** In lamoca terminology pair energy is a measure of the binding strength between a pair of defects, also called pair binding energy. Practically, pair energy value is computed via comparing the energy of two systems, one has two defects separated far enough and the other has the same pair of defects come close. Suppose there is a system with a vacancy and an arsenic atom, if \( E_1 \) is the system energy when the vacancy and the arsenic atom are far apart and \( E_2 \) the system energy when the vacancy and arsenic are brought close, the difference \( E_1 - E_2 \) should represent the binding strength of the pair. Pair energy are classified in lamoca as 1NN, 2NN, 3NN... pair energy to represent the binding strength when the pair of defects are placed as first, second and third nearest neighbors. Pair energy up to 9NN can be defined, but in current version pair energy only up to 3NN has been implemented.

- **Cluster Energy** Unlike pair energy which measures the binding strength between a pair, cluster energy measures the binding strength among a closely coupled cluster of defects, namely the 1NN cluster. A good example of such clusters is a \( \text{As}_nV \) cluster, formed by \( n \) arsenic atoms and a vacancy with the vacancy sitting at the center of the cluster. The binding strength of such clusters may deviate significantly from the result achieved by simply adding up pair energies between the pairs of defects in the cluster. However, if this value is not supplied by user for a cluster, it will be derived from user-supplied pair energy values.

The system’s total binding energy is computed by:

\[
E_B = \sum_i \phi_i (1\text{NNs}) + \sum_{i \neq j} \varphi_{ij} (|r_i - r_j|) \tag{1.2}
\]
where \( \phi_i \) denotes the cluster energy of the 1NN cluster that has the \( i \)th defect as its center; and \( \varphi_{ij} \) represents the pair energy between the \( i \)th and the \( j \)th defect, which depends on the distance between the pair.

As indicated in the above equation, the simulator goes over each of all defects in the system in computing the system energy. So in practice, the user-supplied pair energy value (as will be discussed in Section 2.2.2) should be only one half of its real value, to avoid double counting. For instance, assuming that the pair energy of a 1NN pair of vacancy and arsenic atom is 1000 \( meV \), the user-specified pair energy value of ”VA1nn” and ”AV1nn” (these denotation will be explained in Section 2.2.2) should be 500 \( meV \) apiece. The assignment of cluster energy requires more involved effort. Section 3.1 gives an example of this type computation.

### 1.5.2 Potential Energy

The potential energy of the system is a summation of electrical potential energy of all ionized dopant atoms and charged point defects in the system:

\[
E_P = \sum_i z_i \times (E_F - E_{Fi}) \tag{1.3}
\]

where \( E_F \) and \( E_{Fi} \) donote the Fermi level and intrinsic Fermi level of the system, respectively; \( z_i \) is the charge state of the \( i \)th defect. In current version, the charge state of each type of defect is treated differently in \( n \)-type and \( p \)-type material, as in Table 1.3.

<table>
<thead>
<tr>
<th>No</th>
<th>Defect</th>
<th>( n )-type</th>
<th>( p )-type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vacancy</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Interstitialcy</td>
<td>0</td>
<td>+1</td>
</tr>
<tr>
<td>3</td>
<td>Arsenic</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>4</td>
<td>Arsenic_interstitialcy</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>5</td>
<td>Boron</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>6</td>
<td>Boron_interstitialcy</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

Table 1.3: Charge states of defects
1.6 Simulation Procedure

In lamoca the system evolves by virtue of reactions between point defects and their 1NNs. A step in lamoca terminology is defined as a completion of a reaction process. Reactions are carried out sequentially so simulations proceed step by step. Logically, each simulation follows the way described as below:

1. At start, calculate rates for all possible reactions in the system. According to Eq. 1.1, rate calculations involve computing system energy before and after a reaction.

2. Compare all the rates and choose one reaction from the possible set. The chance that a reaction may be selected is directly proportional to the value of the rate of the reaction.

3. Perform the selected reaction.

4. Increment the system time by the inverse of the sum of all rates:

   \[ \Delta t = \left( \sum_j \nu_j \right)^{-1} \]  

5. Calculate rates for new reactions and update rates for old reactions in case their rates have been affected by the previous reaction. As a result of a reaction, the system configuration is locally altered. This may have effect on the computation of energies of nearby potential reactions and for this reason, change their rates.

6. Repeat from step 2 to step 5 until user-defined conditions are met.
Chapter 2

Running the Program

2.1 Start It

To start the program, execute the command:

```
lamoca
```

Running lamoca invokes two input and two output files, implicitly. The two input files should be named as "parameter.inp" and "energy.inp" and must exist and reside in the current directory where the program is invoked. The two output files are automatically generated into the current working directory and have assigned names of "status.out" and "data.out". All these files are in plain text format. In the following sections we will examine the content of these files in more detail.

2.2 Input Files

Two input files, "parameter.inp" and "energy.inp" are designed for the program to receive user-defined parameters. The former tends to content parameter values that define the environments of the simulation, such as the size of the domain, the boundary conditions on the surfaces and etc. The latter is the place where users may supply energy values which are used in calculation of reaction rates.
2.2. INPUT FILES

2.2.1 Parameter File

The "parameter.inp" input file accepts formatted user input of the following parameters:

- The size of the domain on which the simulation is executed.
- The boundary conditions on domain surfaces.
- The system temperature type and its value.
- The initial impurity (dopant and point defect) type and its distribution within the domain.
- The timing factors of the simulation (e.g., process time).

Each input line within this file is either a comment line distinguished by a leading "#" sign or, a command line defining one of these aforementioned parameters. Each command line starts with a command followed by parameters pertinent to the command. The following is a description of all commands and their required parameters. The commands are shown in **boldcase** and the parameters are in *italic* style.

- **size x_size y_size z_size**
  **Description** This command defines the dimension of the simulation domain, measured by the silicon lattice constant. All supplied size values must be integers. How big the simulation domain can be is limited by the computing resources, especially the memory of the machine. For a machine with a one-gigabyte memory capacity, these size values should not exceed 100. The default domain size is $20 \times 20 \times 20$.

- **temperature t_type t_pre t_exp**
  **Description** This command defines the system temperature type and its value. The system temperature can be:
  - 0: Constant type. The temperature is constant across the whole system. The first parameter $t_{pre}$ supplies the temperature value.
  - 1: Linear type. The temperature varies linearly along the $z$-axis, the principal axis in the system. The second parameter, $t_{exp}$ supplies the proportionality so that the temperature varies in $z$ direction according to:
    \[ T = t_{pre} + t_{exp} \times z. \] (2.1)
By default, the system temperature assumes constant type and has a value 1000°C.

- **boundary** \( zl\_type \, zh\_type \, xl\_type \, xh\_type \, yl\_type \, yh\_type \)

  **Description** This command defines boundary conditions for all six surfaces of the simulation domain. Each of these six boundary conditions, represented by an integer can be:
  
  - 0: Periodic type.
  - 1: Reflection type.
  - 2: Sink type.

  All surfaces are by default periodically bounded and in current version of lamoca only periodic boundary condition is implemented.

- **defect** \( def\_name \, dd\_type \, dd\_val \, [dd\_peak \, dd\_sigma] \)

  **Description** This command initializes either a dopant or a point defect type in the system. Multiple dopant and/or point defect types can be introduced into the system by executing the command multiple times. The first parameter, \( def\_name \) specifies the name of the dopant or defect. And it can be (refer to Setion 1.2 for their definitions):

  - vacancy
  - interstitialcy
  - arsenic
  - boron
  - arsenic_interstitialcy
  - boron_interstitialcy

  The second parameter in the row, \( dd\_type \) is an integer that defines initial distribution type of the defect in the system. And it can be:

  - 0: Uniform distribution. The defect is initially uniformly distributed in the system. The third parameter, \( dd\_val \) supplies the number (an integer) of the defect that should be introduced into the system.
  
  - 1: Uniform distribution. The difference is, the third parameter, \( dd\_val \) at this time defines the density (a number between 0 and 1) of the defect. So the total number of the defect is this number multiplied by the total number of lattice sites in the system. In both cases of uniform distribution, the defect is randomly initialized in the system with equal opportunity of occupying any lattice site.
2.2. INPUT FILES

- 2: Gaussian distribution. The initial distribution profile is a Gaussian type along the $z$ direction (uniform in two other directions). If this type is specified, the fourth parameter, $dd\_peak$ and the fifth, $dd\_sigma$ must be supplied. The former defines the peak position of the Gaussian profile and the latter defines its width. The third parameter, $dd\_val$ defines the defect density at its peak position. So the defect density at any position is calculated as:

\[
d(z) = dd\_val \times \exp \left( -\frac{(z - dd\_peak)^2}{2 \times dd\_sigma^2} \right)
\]  

(2.2)

- 3: Undefined distribution type. Special procedures are necessary in order to initialize defect distributions other than predefined, or to input distribution profiles from other sources, such as output of ion implantation simulations.

- **timing** $t\_ts$ $t\_tt$ $t\_ls$

  **Description** This command can be used to control duration of the simulation and to define the periodicity on which the system information is written to the output files. The first parameter, $t\_ts$ defines the duration of a simulation in terms of step, while the second parameter, $t\_tt$ does so in terms of real time (seconds). A step in the simulation is defined as completion of a hop/exchange event. The third parameter, $t\_ls$ specifies a period in steps on which the system data is snapshot and saved to the output files.

A command and its parameters must be written within a single line. Commands and their parameters should be separated by blanks. Same command may be executed multiple times with the last superseding all previous ones.

### 2.2.2 Energy File

The "energy.inp" input file provides a place where users can supply energy values to be later used in calculations of hopping rate. There are a few types of energy values:

- **Pair Energy**

  **Description** See Section 1.5.1.

- **Cluster Energy**
CHAPTER 2.-running the program

**Description** See Section 1.5.1.

- **Migration Barrier**

  **Description** In lamoca mobile species (i.e., point defects) may hop to or exchange its position with its nearest neighbor. This process is normally completed by overcoming a barrier between the two sites whose height determines the rate of the process. If we at the moment ignore the adjustment of the barrier height due to the system energy change before and after the process, then this barrier height is what we call the unbiased migration barrier. Notice that migration barrier is a reaction-dependent value, i.e., each reaction involving different mobile species should have a unique migration barrier value.

- **Reaction Energy**

  **Description** In lamoca the change in system energy after a reaction can normally be calculated using pair energy and cluster energy values based on the change in system configurations. However, for some reactions where only one defect is involved, pair energy and cluster energy will be unavailable. It is under such circumstances that the reaction energy concept enters. A good example of such reaction is the annihilation of a vacancy-interstitial pair:

  $$V + z \leftrightarrow Z + Z.$$  
  \hfill (2.3)

  Here, $V$, $z$, and $Z$ are symbols for vacancy, interstitial, and silicon, respectively. Notice in this reaction binding energy values are unable to account for the amount of energy released by the system relaxation.

We now describe how to define these energy values within the "energy.inp" file.

Each line in the file is a data entry to be used to supply an energy value, except for empty lines. Each data entry consists of a keyword and a numerical integer which defines the energy value in millielectronvolt. The keywords should be constructed by following the rules accordingly.

- **Pair Energy:** The keyword for pair energy entry consists of exactly five letters as "×××nn". The first two letters should be the symbols of the defect pair and the third one should be a numerical number denoting the distance between the pair classified as $1NN, 2NN, 3NN...$. For instance, the pair energy for a vacancy and arsenic pair separated by the third nearest neighbor distance should have a keyword constructed as "$V.A3n$", where $V$ and $A$ are symbols for vacancy and arsenic, respectively, and 3 marks the third nearest neighbor distance between the pair. Refer to Table 1.1 for a complete list of the defect symbols.
2.3. OUTPUT FILES

- **Cluster Energy**: The keyword for cluster energy also consists of five letters. The first letter should be the symbol of the species sitting at the center. And the rest four letters should represent the four nearest neighbors. So, for example, the keyword for the cluster energy of a $As_3V$ should be "VAAAAZ", where a vacancy is surrounded by three arsenic atom plus a silicon atom.

- **Migration Barrier**: The keyword for specifying migration barrier should be constructed with reference to the reaction involved. For example, the keyword for migration barrier for reaction $z + A \rightleftharpoons Z + a$ is $ZAZam$. Understandably, the first two letters are symbols of the reactant, and the two subsequent letters are symbols of the resultant, leaving the fifth letter "m" to mark that this is migration barrier keyword. Caution must be taken not to change the order of the two resultant symbols.

- **Reaction Energy**: The keyword for specifying reaction energy should be very much like one for the migration barrier, except that the fifth letter is "r" instead of "m". So for the same reaction mentioned in the last paragraph, the keyword should be "$ZArA$".

**Note**: By default, all the pair and cluster energy values are zero.

2.3 Output Files

Two output files are generated automatically as a simulation is started, named as "data.out" and "status.out" and both saved in the current working directory.

2.3.1 Data File

Lamoca writes critical information about dopant locations in the system to the "data.out" file as the simulation proceeds. The frequency of this writing process is determined by the user-supplied parameter, $tt_ls$ (refer to command **timing** in Section 2.2.1). This saved information can be very useful for future analysis. The file looks like:

```
000.000 000.000
2
A 3
```
The two number on the first line are the step of the simulation and time of the system, respectively. They indicate when the writing is carried out. The number on the second line describes how many of dopant types are in the system. The first letter on the third line specifies the dopant type and it is followed by the total number of this type of dopant in the system. The subsequent line(s) gives detailed information on coordinates of these dopant. The fifth line in the example lists another type of dopant and its number and followed by their coordinates. The second paragraph has the same information of the system at a later instant of the simulation.

2.3.2 Status File

Lamoca saves its conditions to the "status.out" file during the simulation. This includes the environments of the simulation and possibly, error messages. Lamoca also reports effective dopant diffusivities periodically and the information is saved in this file. Effective dopant diffusivities are determined by:

\[ D = \frac{\langle \Delta x^2 \rangle}{6 \times \Delta t}, \]  

(2.4)

where \( \langle \Delta x^2 \rangle \) is the mean square distance that the dopant has diffused; \( \Delta t \) denotes the time duration for the diffusion.

Lamoca also displays this information on the screen.
Chapter 3

Examples

3.1 Cluster Energy Dilemma

As acknowledged in Section 1.5.1, the user-supplied pair energy value for a pair of defects should be only half of its real value. We gave an example in that section to show why this is so. In this section, we will learn how to obtain the cluster energy that the user should specify in the "energy.inp" file, from the known real cluster energy of a cluster.

Let us suppose there is an isolated $As_4V$ cluster. The real binding energy of the cluster is known as 6500 meV and the real 1NN and 2NN pair energy of an $AV$ (an arsenic and a vacancy) pair is also known to be 1000 and 500, respectively. According to Eq. 1.2, the total binding energy of this cluster is computed in the program as: (for an explanation of these denotations, see Section 2.2.2)

$$E_B = VAAAA + AVZZZ + AVZZZ + AVZZZ + AVZZZ + 12 \times AA2nn. \quad (3.1)$$

There are five cluster terms as underlined on the right side of the above equation. The first term $VAAAA$ is our interest; The rest four terms are identical, each corresponding a cluster that has an arsenic atom at center and a vacancy and three regular silicon atoms as 1NNs. These simple clusters are nothing but an isolated $AV$ pair so their energy value each is equal to the 1NN $AV$ pair energy, 500 meV and in total, 2000 meV. The last term in Eq. 3.1 shows that there are in total 12 2NN AA pair energies should be counted for a total of six 2NN AA pairs. The right size terms, except the first one now sum up to $12 \times (500/2) + 4 \times (1000/2) = 5000$ meV. Since the total binding
energy is 6500 meV, the cluster energy of VAAAA should be 1500 meV indeed.

3.2 A simple Vacancy-Mediated Arsenic Diffusion Experiment

In this example we will learn how to build two inputs files for a simulation of vacancy mediated arsenic diffusion. The parameters are:

- The domain size is $40 \times 40 \times 40$.
- The system temperature is constant type of $1000^\circ\text{C}$.
- Periodic boundary condition on all surfaces.
- There are 1024 arsenic atoms and a single vacancy in the system.
- The simulation runs 1000000 steps and the log period is also 1000000.

The "parameter.inp" file should look like:

```
#Example: vacancy mediated arsenic diffusion
#
size 40 40 40
temperature 0 1000 0
defect arsenic 0 1024
defect vacancy 0 1
timing 1000000 1e38 1000000
#
```

Note: There should be a "0" after the temperature value, otherwise the program will crash. It has no meaning other than a place holder. The last argument in "timing" should only be specified in integer form. If specified in scientific form the program will fail in reading this input value.

The "energy.inp" file looks like:

```
AV1nn 300
VA1nn 300
```
3.2. ARSENIC DIFFUSION

\begin{align*}
AV2nn & 200 \\
VA2nn & 200 \\
AV3nn & 100 \\
VA3nn & 100
\end{align*}

In this example we only supply pair energy values and leave the program itself to compute the cluster energy based on the supplied pair energy values (by doing that we assume that the cluster energy is a simple summation of pair energies). Fig. 3.1 plots the mean square distance that arsenic atoms have diffused against the supplied 1NN AV pair energy, based on results from multiple simulation. VA2nn and VA3nn are assumed to be equal to two third and one third of VA1nn, respectively.

![Graph showing mean square distance vs. VA1nn (milli-eV)](image)

Figure 3.1: Simulation results of vacancy mediated arsenic diffusion. Plotted is the mean square distance that arsenic atoms have diffused during a 1000000 steps of simulation as a function of the pair energy of an arsenic vacancy pair.
Chapter 4

Advanced Topics

4.1 Acceleration to Lamoca

4.1.1 Problem

The reason that molecular dynamics cannot be used in general to study dopant diffusion in silicon is that most of the computational effort goes into modeling the vibrations of atoms, and these are rapid processes that do not contribute to long range dopant diffusion. Lamoca simulations allow us to ignore these rapid processes and focus on the much slower processes that do contribute to dopant diffusion. However, other unwanted rapid processes that take up most of the computer time, but which do not contribute to diffusion, can still occur within lamoca simulations. A simple example might be a vacancy bound in a $\text{As}_nV$ cluster, for which the bond is so strong that the vacancy can hardly escape. The simulations spend enormous amount of time on just repeatedly exchanging positions of the vacancy and its neighboring arsenic atoms, which of course do not lead to arsenic diffusion in long range.

Fig. 4.1 represents a generalized description of the problem. On the figure there are four states labelled from 0 to 3 with state 1 and 2 assuming significantly lower energies than the other two states. The fast process in the case is the interchanging of state 1 and 2, while jumping from state 1 to 0 and that from 2 to 3 are much slower processes. Understandably, the computer will spend most of its time on exchanging state 1 and 2.
4.1. ACCELERATION TO LAMOCA

Figure 4.1: Schematic of states with fast processes. State 1 and 2 are low energy states and the transition process between them is much faster than the processes of jumping out of them, either from state 1 to 0 or from state 2 to 3.

4.1.2 Solution

To solve the problem, these fast processes aforementioned must be removed from simulations. One way to do this is to calculate the average time it takes to move from the states with fast processes (e.g. state 1 and 2 in Fig.) to their neighboring states (e.g. state 0 and 3). This allows a simulation to immediately jump out of these states with fast processes without wasting much of its time on processes that do not contribute to long range diffusion.

To formulate the idea, we generalize our concept of a state to include a group of states which approaches local equilibrium due to extensive sampling via fast processes. More specifically, these states linked by fast processes are replaced by a single effective state with an effective energy:

\[ E_K = -k_B T \ln \left( \sum_{k \in K} \exp \left( -\beta E_k \right) \right) \]  

(4.1)

where \( K \) denotes the set of states. The jumping rate to a neighboring state \( j \) outside the set is calculated as:

\[ \nu(K \rightarrow j) = \sum_{k \in K} p_k \nu(k \rightarrow j) = \sum_{k \in K} \exp \left( -\frac{E_k - E_K}{k_B T} \right) \nu(k \rightarrow j) \]  

(4.2)

where \( p_k \) is the probability of that the state \( k \) is occupied within the set. Since we assume the
system has reached local equilibrium, the relative probability of finding the system in one of the set of states is just given by its Boltzmann factor as in Eq. 4.2.

4.2 Fermi Level Modeling in Atomistic Simulations

The intrinsic Fermi level variations within the system, as a result of point-like property of source charges and possibly nonuniform doping in the region, in return effect on dopant diffusion in silicon. The effect manifests in two ways:

1. Ionized dopant atoms as well as charged point defects experience electrical fields due to the spatial variations in the intrinsic Fermi level:

\[
\vec{E} = \frac{\nabla E_{Fi}}{e} = -\frac{k_B T_e}{e} \times \frac{\nabla n}{n}.
\]  

(4.3)

Here, \( E_{Fi} \) denotes the intrinsic Fermi level and \( n \) is the conduction electron density.

2. The population of dopant diffusion mediator (i.e., point defects) depends on local Fermi level.

\[
C_{x-} \propto \left( \frac{n}{n_i} \right), \quad C_{x=} \propto \left( \frac{n}{n_i} \right)^2, \quad C_{x+} \propto \left( \frac{n}{n_i} \right)^{-1}, \quad C_{x++} \propto \left( \frac{n}{n_i} \right)^{-2}
\]  

(4.4)

According to Boltzmann equation, the free carrier concentrations define local electrical potential level. Traditionally, these quantities have been calculated from dopant profiles using the charge neutrality assumption. Unfortunately, at atomic scale dopant atoms are considered to be discrete and point-like, thus dopant concentration is no longer a valid measurement. To obtain free carrier concentrations, we are forced to solve the Poisson equation in the presence of discrete point charges.

To clearly define the problem, we consider a system with a defined uniform background carrier (electron or hole) concentration. To maintain charge neutrality in the region, an equal, but opposite dopant (donor or acceptor) concentration is also assumed. A pair of charges is then introduced into the system with the positive (negative) charge held at the origin (point-like) and the other mobile, negative (positive) charge released in the system. The challenge is then to solve the redistribution of mobile charges in the neighborhood of the point charge at the origin under consideration of the background carrier screening.
4.2. FERMI LEVEL MODELING IN ATOMISTIC SIMULATIONS

4.2.1 Carrier Distribution: Classical Solution

There exists a classical solution to the problem, best known in its potential form as the screened Coulomb potential. The classical approach applies Boltzmann statistics and assumes the validity of linearization of the Boltzmann equation [3]. The solved charge distribution has the same form as the screened Coulomb potential:

\[
\rho(r) = -\frac{e}{4\pi l_D^2} \exp\left(-\frac{r}{l_D}\right),
\]

where \(l_D\) is the Debye length

\[
l_D = \sqrt{\frac{k_S \varepsilon_0 k_B T}{e^2 (n + p)}}.
\]

Here, \(n\) and \(p\) are the background carrier concentrations. This solution has a simple analytic form, but diverges as \(r\) approaches zero, which is physically incorrect and causes a problem in the carrier concentrations derivation. To overcome this characteristic of the classical solution, a quantum approach is necessary.

4.2.2 Carrier Distribution: Quantum Solution

In order to solve the problem in a practical way, two assumptions are made. The first assumption regards the system. We assume that the conduction band has a parabolic shape near its minima and conduction electrons are free. The second assumption is made on the approach, where we assume the quantum perturbation method is applicable to this problem.

Based on the two above assumptions, the generalized susceptibility of the system, characterizing the systems response to an external sinusoidal perturbation can be calculated as a function of spatial frequency of the perturbation [4]:

\[
\chi(\vec{q}) = -\frac{6 m^*_e}{\pi^3 \hbar^2} \int_0^\infty \frac{f(E_{\vec{k}})}{2k \cdot \vec{q} + q^2} d^3k,
\]

where \(q\) represents the spatial frequency of the sinusoidal perturbation, \(m^*_e\) is the effective mass of electrons (0.33\(m_e\)), and the factor of six accounts for multiple conduction band minima. The background carrier concentrations and system temperature enter as parameters through \(f(E_{\vec{k}})\), the Fermi-Dirac distribution function, with the Fermi energy chosen to match the defined background carrier concentrations. Eq. 4.7 characterizes the response of conduction electrons. Had the
Figure 4.2: Excess electron distributions in the neighborhood of a positive point charge, calculated using quantum perturbation method for (a) three distinct background carrier (electron) concentrations of $1 \times 10^{18}$, $1 \times 10^{19}$ and $5 \times 10^{19}$ cm$^{-3}$ at temperature of 1000°C. As the carrier concentration increases, the distribution profile drops off more rapidly due to stronger screening; (b) three different system temperatures of 800, 900 and 1000°C with the background carrier (electron) concentration set to be $1 \times 10^{19}$ cm$^{-3}$. Distribution profiles show only a weak dependence on temperature.

response of holes been sought, $m_e^*$ must be replaced by the effective mass of hole $m_h^*$ ($0.51 m_e$) and the factor of six should change to two.

Given the susceptibility function of the system from Eq. 4.7, the induced charge distribution is calculated in response to an external point charge, which is represented by its Fourier components [4]:

$$\rho(\vec{r}) = -e \int_0^\infty \left[ \frac{k_{Si}}{\varepsilon(q)} - 1 \right] \exp(i\vec{q} \cdot \vec{r}) \frac{d^3\vec{q}}{(2\pi)^3},$$

where $\varepsilon(q)$ is the dielectric function of the system

$$\varepsilon(q) = k_{Si} - \frac{\varepsilon_0^2}{\varepsilon_0 q^2} \chi(q).$$

Eq. 4.8 was evaluated numerically and the results are depicted in Fig. 4.2. Fig. 4.2(a) shows results for three distinct background carrier concentrations: $5 \times 10^{18}$, $1 \times 10^{19}$ and $5 \times 10^{19}$ cm$^{-3}$. Clearly, the charge distribution drops off more rapidly with an increasing background carrier concentration, indicating a stronger screening effect. Fig. 4.2(b) shows results for three systems with temperature of 800, 900 and 1000°C, respectively. As seen, within the process temperature regime the charge distribution only weakly depends on temperature.
4.2. **FERMI LEVEL MODELING IN ATOMISTIC SIMULATIONS**

Figure 4.3: Excess electron distributions in the neighborhood of a positive point charge from both the classical and quantum solutions for a system with a background carrier (electron) concentration of $1 \times 10^{19}$ cm$^{-3}$ and a temperature of 1000° C. The classical Debye length in this case is 2.67 nm. The quantum solution matches the classical solution at long range. Also plotted is the fitted curve $1.24 \times 10^{19} \exp(-r/2.67) / \sqrt{r^2 + 0.75^2}$, which shows excellent agreement with the quantum solution over the entire range.

Upon examination of numerical results for different background carrier concentrations and temperatures, we find that the quantum solution can be modeled by:

$$\rho(r) = \rho(0) \frac{r_0}{\sqrt{r^2 + r_0^2}} \exp\left(-\frac{r}{l_D}\right),$$  \hspace{1cm} (4.10)

where $\rho(0)$ is the charge density at the origin, and $r_0$ is a parameter representing best fitting to the numerical results. In fact, $r_0$ is only a weak function of the background carrier concentration and temperature, and is approximately 0.8 nm. In Fig. 4.3, both the classical and quantum solutions are plotted, along with Eq. 4.10, which shows excellent agreement with the quantum solution over the entire range. The quantum solution matches the classical solution at long range, but deviates from it in the short range. Most significantly, it gives a finite value at the origin, where the classical solution diverges. All previous calculations are carried out over background carrier concentrations up to $3 \times 10^{20}$ cm$^{-3}$. Beyond this concentration limit, Friedel oscillations [5] appear and Eq. 4.10 no longer accurately predicts the numerical results.
4.2.3 Fermi Level

Given the solution of charge distribution in the neighborhood of a single point charge, the overall carrier concentrations can be, to first approximation, determined by summing contributions from all ionized dopant atoms and charged defects, with a positive point charge (e.g., an ionized donor) inducing an electron cloud in its neighboring region and a negative charge (e.g., an ionized acceptor) inducing a hole cloud. Nevertheless, the production of carrier distributions over the whole system is inherently an iterative process, since the Debye length appeared in Eq. 4.10 is itself a function of carrier concentrations, as shown by Eq. 4.6. To start, the carrier concentration is set to be equal to the dopant concentration averaged over a broad range. The Debye length is then calculated using Eq. 4.6 and the carrier concentration recalculated using Eq. 4.10. Subsequent processes repeat the pattern of recalculation of Debye length and carrier concentration, until the carrier concentrations converge to a specified criterion. The local electrical potential level is then derived from the achieved carrier concentrations by the Boltzmann equation.


