Spring 2019

An Examination of Kinetic Monte Carlo Methods with Application to a Model of Epitaxial Growth

Dylana Ashton Wilhelm

An Examination of Kinetic Monte Carlo Methods with Application to a Model of Epitaxial Growth

by

Dylana Ashton Wilhelm

Bachelor of Science
James Madison University 2017

Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Arts in

Mathematics

College of Arts and Sciences

University of South Carolina

2019

Accepted by:

Yi Sun, Director of Thesis

Xinfeng Liu, Reader

Cheryl L. Addy, Vice Provost and Dean of the Graduate School

Dedication

To Boomer and Nemo for the lasting imprint you have each left on my life.

Acknowledgments

I would like to thank my thesis advisor Dr. Yi Sun and my reader Dr. Xinfeng Liu for taking the time to review this work. I am also particularly grateful for the professors throughout my undergraduate study who fostered my mathematical abilities and encouraged me to pursue research as well as a graduate degree in mathematics. Lastly, I would like to express my profound gratitude to my parents and sisters for providing me with unwavering support and encouragement throughout my life. My accomplishments would not have been possible without them standing behind me.

Abstract

Through the assembly of procedural information about physical processes, the kinetic Monte Carlo method offers a simple and efficient stochastic approach to model the temporal evolution of a system. While suitable for a variety of systems, the approach has found widespread use in the simulation of epitaxial growth. Motivated by chemically reacting systems, we discuss the developments and elaborations of the kinetic Monte Carlo method, highlighting the computational cost associated with realizing a given algorithm. We then formulate a solid-on-solid bond counting model of epitaxial growth which permits surface atoms to advance the state of the system through three events: hopping, evaporation, and condensation. Finally, we institute the kinetic Monte Carlo method to describe the evolution of a crystalline structure and to examine how temperature influences the mobility of surface atoms.

Table of Contents

List of Figures

Chapter 1

Background on Epitaxial Growth

Epitaxy refers to the process by which a crystalline film grows on an underlying crystalline substrate. The term is derived from the Greek prefix *epi*, meaning "upon" or "above", and the suffix *taxis*, meaning "arrangement" or "order." As suggested by such a designation, the process of epitaxy results in the deposition of atoms onto an existing crystal in an ordered manner. The deposited film, or epitaxial layer, assumes an orientation with accordance to the pre-existing morphology of the surface. The epitaxial film may be of the same or different chemical structure as the substrate. In order to precisely describe the effect of composition on surface growth, epitaxy is categorized into two main types: homoepitaxy and heteroepitaxy. In homoepitaxy, the deposited film is composed of the same material as the substrate whereas in heteroepitaxy, the crystalline film grows on a substrate of a different material. The variation in chemical compositions in heteroepitaxy introduces a lattice mismatch between film and substrate, which creates a strain and thereby presents an additional influence on the mechanism of crystal growth.

Attention to the dependance of growth on various factors, such as interfacial energy, temperature, and lattice misfit, is of great importance as semiconductor and nanotechnology fabrication is a common application of epitaxial growth. Integrated circuits, for example, consist of a complex layering of semiconductor wafers and act as an essential component for many electronic and photonic devices. The performance and lifetime of such devices is rooted in their crystallinity; if growth is not well controlled or a significant lattice mismatch exists, unfavorable surface features may

form and alter the mechanical, electrical, and thermal properties of the device. To achieve optimal performance, structural perfection is thus desired. Models of epitaxial growth for technological applications therefore aim to identify the ideal conditions in the fabrication process that will minimize structural defects.

The physical form of an epitaxial layer is largely attributed to the principal growth mode and the atomic processes occurring along the surface. Figure 1.1 illustrates the various processes that may take place on the surface during epitaxial growth: deposition, evaporation, diffusion, nucleation of islands, attachment to an island, diffusion along a step edge, diffusion to a lower terrace, and detachment from an island.

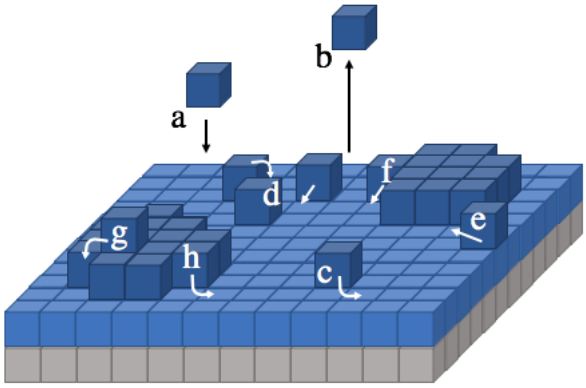

Figure 1.1: Atomic processes during growth: (a) deposition, (b) evaporation, (c) diffusion, (d) nucleation of islands, (e) attachment to an island, (f) diffusion along a step edge, (g) diffusion to a lower terrace, and (h) detachment from an island

Beginning in the late 1950s, experimental observations obtained through transmission electron microscopy led to the establishment of three distinct modes of epitaxial growth: Frank van der Merwe, Volmer-Weber, and Stranski-Krastanov (Figure 1.2). Determined through energy considerations of the film and substrate, the mode of growth characterizes the nucleation of the deposit and influences the surface morphology as growth proceeds, developments explored in detail by Pashley [8]. To

express the criterion for each growth mode, define γ_S to be the surface energy of the substrate, γ_F to be the surface energy of the film, and γ_I to be the interfacial energy between the film and substrate.

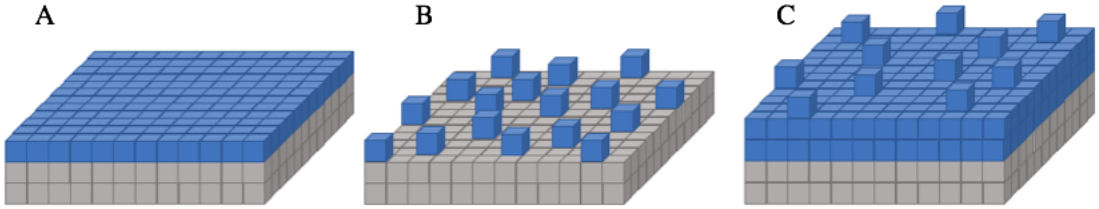

Figure 1.2: Classical modes of epitaxial growth: (A) Frank van der Merwe, (B) Volmer-Weber, (C) Stranski-Krastanov

In the Frank van der Merwe growth mode, each layer is formed by the nucleation of islands which coalesce to create a continuous film. A complete monolayer must be formed prior to the nucleation of a new layer. This growth mode occurs as the result of adsorbed atoms, known as adatoms, having a stronger interaction with the substrate than amongst themselves, a condition which can be described by the following inequality:

$$\gamma_F + \gamma_I \leq \gamma_S.$$

Complete wetting of the surface ensues as the growing layer reduces the surface energy. With smaller misfits producing a lower γ_I, similar structures in parallel orientation tend to result in this layer-by-layer growth method.

When the total surface energy of the film interfaces exceeds that of the substrate (i.e. $\gamma_F + \gamma_I > \gamma_S$), the growth of the film is characterized by the formation of three-dimensional clusters, or nuclei. This mechanism of growth is referred to as Volmer-Weber growth. The growing layer balls up on the surface to minimize interaction with the substrate. In order for the deposited atoms to migrate over the surface, a high diffusion coefficient is required. Once the formation of new nuclei ceases, the adatom clusters develop to form rough multi-layer features along the surface; thus, growth by the Volmer-Weber mode typically results in many imperfections in the epitaxial film.

Stranski-Krastanov growth is an intermediary mode of growth in which monolayer growth (Frank van der Merwe growth) is followed by the formation of clusters (Volmer-Weber growth). This method of growth occurs when a significant lattice misfit exists between the film and substrate. The mismatch creates a strain which results in the elastic energy being much greater than that of the surface energies. The initial layer of the film is strained to match the lattice of the substrate. The adsorbed atoms then form smooth, continuous monolayers until a critical thickness is achieved. Such a thickness is a function of the misfit of a system; a higher misfit requires a smaller number of monolayers. Once the appropriate thickness is reached, the accumulated elastic energy relaxes the strain and allows for the growth of three-dimensional clusters to take place. The transition in the growth mode is believed to occur in order to attain more energetically favorable conditions; the formation of clusters demands less energy than the continued growth of monolayers.

As discussed in the structural details of the three classical modes of epitaxial growth, the morphology of a film is largely dependent on the interaction of the deposited atoms and the lattice configurations of the film and substrate. Since the deposit and the substrate are composed of different materials in heteroepitaxy, consideration of the lattice spacings of the chosen materials is necessary to achieve a desired form. Little to no misfit results in smooth, layer-by-layer growth whereas a large misfit or a significant interfacial energy produces rough surface features. Therefore, if structural perfection of the epitaxial growth is sought, homoepitaxy or a substrate with zero misfit should be utilized.

To study the evolution in the surface morphology of a crystal, a model must account for how adatoms interact with each other and the substrate as well as the effects of various environmental factors. The kinetic Monte Carlo method is a prevalent approach in exploring epitaxial growth. The premise of this method is that adatoms occupy a position on a lattice and may only interact with neighboring atoms. The po-

tential energy of each adatom is determined by examining the number of bonds shared with neighbors and is incorporated into probabilistic rules which govern deposition, diffusion, and other atomistic processes. The kinetic Monte Carlo method simulates the growth of a crystal by using the transition rates depending on the energy barrier between states and by progressing time in relation to the chemical kinetics of the system.

In Chapter 2, we will discuss the selection of stochastic modeling to simulate chemical kinetics and introduce the terminology that will be utilized in the development of algorithms. In Chapter 3, we will present the history of the kinetic Monte Carlo method and formulate algorithmic tools to model chemical reactions. In Chapter 4, we will apply these tools to a simple model of epitaxial growth.

Chapter 2

Stochastic Formulation

2.1 Introduction

The time evolution of a broad range of systems in biology, chemistry, and physics can be modeled in two fundamental ways: deterministically and stochastically. In a deterministic approach, time evolution is treated as a continuous process which is represented by a set of ordinary differential equations. In a stochastic approach, time evolution takes the form of a Markovian random walk which can be described by a differential-difference equation. As the algorithms presented in Chapter 3 simulate the evolution of chemical systems, we will discuss the two approaches in this context.

Consider a system of chemical reactions in which the number of each chemical species is represented by X_i, $i = 1, \ldots, N$. Although molecular populations in a chemical system change only by discrete integer amounts, the traditional approach to simulating a reaction network involves constructing a set of coupled ordinary differential equations of the form

$$\frac{dX_i}{dt} = f_i(X_1, \ldots, X_N), \quad i = 1, \ldots, N \tag{2.1}$$

where the functions f_i result from the stoichiometric forms and rate constants of each reaction. Solving the system of differential equations produces the molecular population levels as a function of time. Furthermore, solution of

$$0 = f_i(X_1, \ldots, X_N), \quad i = 1, \ldots, N \tag{2.2}$$

provides the equilibrium concentrations of the chemical system.

For instance, consider a chemical system posed by Paulsson et al. [9] that is given by the following reactions

$$\emptyset \xrightarrow{k_1} I \xrightarrow{k_2} P \xrightarrow{k_3} \emptyset \tag{2.3}$$

$$I + S \xrightarrow{k_4} S \tag{2.4}$$

$$\emptyset \underset{k_6}{\overset{k_5}{\rightleftarrows}} S. \tag{2.5}$$

In this system, $\emptyset$ acts as a source to the molecule I and a sink to the molecule P. Equation (2.3) describes how I is produced with rate k_1 and how P arises from I with rate k_2 but degrades with rate k_3. This reaction is coupled with equation (2.4) in which the degradation of I is catalyzed by a signal molecule S with rate k_4. The production and degradation of S with rates k_5 and k_6 respectively is provided in equation (2.5). By the law of mass action, the time evolution of the concentrations of I, P, and S can be described by the system of ordinary differential equations

$$\frac{dI}{dt} = k_1 - k_2 I - k_4 SI, \tag{2.6}$$

$$\frac{dP}{dt} = k_2 I - k_3 P, \tag{2.7}$$

$$\frac{dS}{dt} = k_5 - k_6 S. \tag{2.8}$$

To determine the equilibrium points of the system, we set equations (2.6)–(2.8) equal to zero and solve for I, S, and P; this yields the number of each species in terms of the rate constants of the reactions:

$$I = \frac{k_1 k_6}{k_2 k_6 + k_4 k_5},\ S = \frac{k_5}{k_6},\ P = \frac{k_1 k_2 k_6}{k_3(k_2 k_6 + k_4 k_5)}.$$

Although a deterministic formulation such as this is sufficient for some chemically reacting systems, there are many cases in which the differential reaction-rate equations fail to accurately describe the fluctuations in the molecular species. Collisions in a system of molecules occur in a seemingly random manner, a behavior which is more appropriately accounted for in a stochastic formulation. In this approach, the

reaction constants are viewed as probabilities per unit time and the time evolution of the system is governed by a single differential-difference equation constructed according to the laws of probability. Details of the stochastic framework are made clear following the introduction of necessary terminology.

2.2 Definitions and Notation for Stochastic Formulation

Consider a chemical system with N species X_j, $j = 1, \ldots, N$ and denote the population levels of X_j molecules at time t as $X_j(t)$. Suppose the molecular species may interact through a set of M chemical reactions and let R_i refer to the ith reaction where $1 \leq i \leq M$. The occurrence of a chemical reaction in a system effects change in the molecular population values. For a given reaction R, define the set of reactant molecules as Depends(R) and the set of all species that have their copy numbers changed when the reaction is executed as Affects(R). To update the molecular population values following a reaction, let $\boldsymbol{\nu}_i \in \mathbb{R}^N$ be a vector of coefficients that adjusts the copy numbers of each species with respect to the reaction R_i. Furthermore, define the state vector $\mathbf{v}$ to contain the number of molecules of each species present in the system at a given time t.

The reaction R_i is also characterized by its propensity a_i, which is determined by the mesoscopic rate constant of the reaction and the state of the system. Assuming the system is in thermal equilibrium, the molecules are uniformly distributed and collisions between molecules occur at random. A collision takes place, however, only if the center-to-center distance between two molecules is no greater than the sum of their radii. The propensity of a reaction thus involves the probability of reactant molecules colliding as well as the probability that a reaction occurs upon this collision. We therefore define the propensity function $a_i(\mathbf{v})$ such that $a_i dt$ is the probability that an R_i reaction will occur in the infinitesimal time interval $(t, t+dt)$ given that the system was in state $\mathbf{v}$ at time t. This definition is often regarded as the fundamental

hypothesis of stochastic chemical kinetics [4,5,7].

To provide an example of the definitions highlighted above, consider the chemical system presented in equations (2.3) through (2.5). We have that the state vector is given by the population of each molecular species, $\mathbf{v} = (I(t), S(t), P(t))$. The table below gives the propensity and state-change vector $\boldsymbol{\nu}$ for each reaction.

Table 2.1: Application of definitions to a chemical reaction system

Reaction	**Propensity**	**Depends**	**Affects**	$\boldsymbol{\nu} = (\Delta I, \Delta S, \Delta P)$
$\emptyset \xrightarrow{k_1} I$	k_1	–	$\{I\}$	$(1, 0, 0)$
$I \xrightarrow{k_2} P$	$k_2 I(t)$	$\{I\}$	$\{I, P\}$	$(-1, 0, 1)$
$P \xrightarrow{k_3} \emptyset$	$k_3 P(t)$	$\{P\}$	$\{P\}$	$(0, 0, -1)$
$I + S \xrightarrow{k_4} S$	$k_4 I(t) S(t)$	$\{I, S\}$	$\{I\}$	$(-1, 0, 0)$
$\emptyset \xrightarrow{k_5} S$	k_5	–	$\{S\}$	$(0, 1, 0)$
$S \xrightarrow{k_6} \emptyset$	$k_6 S(t)$	$\{S\}$	$\{S\}$	$(0, -1, 0)$

2.3 Chemical Master Equation

Traditionally, the time evolution of a chemical system is constructed through the solution of a differential-difference equation. Termed the master equation, the equation concerns the function $P(\mathbf{v}, t|\mathbf{v}_0, 0)$, which is the probability that the system is in state $\mathbf{v}$ at time t given that it was in state $\mathbf{v}_0$ at time $t = 0$. To derive the equation, we consider the number of ways in which the system can arrive at state $\mathbf{v}$ in the time interval $(t, t + dt)$. The possible routes from state $\mathbf{v}_0$ to $\mathbf{v}$ include no reaction occurring in $(t, t + dt)$, exactly one reaction occurring, and more than one reaction occurring. All of these events are mutually exclusive and so the probability that any of them occur can be calculated by summing their individual probabilities [6], which gives

$$\begin{aligned} P(\mathbf{v}, t + dt|\mathbf{v}_0, 0) = P(\mathbf{v}, t|\mathbf{v}_0, 0) \left[1 - \sum_{i=1}^{M} a_i(\mathbf{v})dt + o(dt)\right] \\ + \sum_{i=1}^{M} P(\mathbf{v} - \boldsymbol{\nu}_i, t|\mathbf{v}_0, 0)[a_i(\mathbf{v} - \boldsymbol{\nu}_i)dt + o(dt)] + o(dt). \end{aligned} \tag{2.9}$$

By subtracting $P(\mathbf{v}, t|\mathbf{v}_0, 0)$ from both sides of equation (2.9), dividing throughout by dt, and taking the limit as dt approaches 0, we obtain the chemical master equation:

$$\frac{\partial}{\partial t}P(\mathbf{v}, t|\mathbf{v}_0, 0) = \sum_{i=1}^{M}[a_i(\mathbf{v} - \boldsymbol{\nu}_i)P(\mathbf{v} - \boldsymbol{\nu}_i, t|\mathbf{v}_0, 0) - a_i(\mathbf{v})P(\mathbf{v}, t|\mathbf{v}_0, 0)]. \tag{2.10}$$

For the Paulsson system in equations (2.3)–(2.5), the chemical master equation is given by

$$\begin{aligned}\frac{\partial p(I,S,P)}{\partial t} &= k_1 p(I-1,S,P) - k_1 p(I,S,P) + k_2(I+1)p(I+1,S,P-1) \\ &\quad - k_2 I p(I,S,P) + k_3(P+1)p(I,S,P+1) - k_3 P p(I,S,P) \\ &\quad + k_4(I+1)S p(I+1,S,P) - k_4 I S p(I,S,P) + k_5 p(I,S-1,P) \\ &\quad - k_5 p(I,S,P) + k_6(S+1)p(I,S+1,P) - k_6 S p(I,S,P).\end{aligned}$$

Although the master equation exactly describes the time evolution of a system, its formulation and solution is not always plausible. For example, a model of the virus lambda phage contains 75 equations in 57 species [3]; writing down the master equation for such a large system, let alone solving it, would be a tremendous undertaking. Stochastic simulation algorithms, which will be discussed in Chapter 3, offer a more feasible approach; these algorithms generate the exact probability of a system following a given trajectory as predicted by the master equation without the need for such an equation to be explicitly expressed.

Chapter 3

Kinetic Monte Carlo Methods

3.1 History of KMC

Named after the district of Monaco renowned for gambling, the Monte Carlo method refers to a technique through which random numbers are used to solve problems. The inception of the method emerged in a game of solitaire in which Stan Ulam considered the use of successive random operations to estimate the probability of a successful outcome. With new advancements in electronic computing occurring at the time, Ulam envisioned the application of such a statistical sampling approach to a variety of problems in mathematical physics and proposed the method to John von Neumann in a 1946 correspondence. The method initially boasted the appeal of being an efficient means to approximate integrals which are unable to be solved analytically. With the establishment of several variations, the Monte Carlo method has since progressed to form a class of algorithms with the ability to address complex problems in a wide range of disciplines [1,13].

Beginning in the 1960s, the Monte Carlo method developed to study systems that evolve dynamically from state to state, a behavior which coincides with the kinetics of a Markov random walk. Through examining stochastic sequences of the fundamental physical transitions of a process, the approach precisely models the time evolution of a system, achieving longer time scales than previously attainable by other methods. Since being coined the kinetic Monte Carlo (KMC) method in the 1990s, the approach has garnered widespread use in simulating processes of a stochastic nature, including

those found in statistical physics, atomic diffusion, chemical reactions, biochemical networks, cellular growth, and irradiation [1,13]. The foundation of the kinetic Monte Carlo method arises from the consideration that the modeled system transitions from one state to another in a way which can be determined probabilistically. Random variate generation from a dynamic probability distribution allows the kinetic Monte Carlo method to choose an event and its associated timestep to advance the system accordingly. With the transition rates tied to the underlying state space of the system, kinetic Monte Carlo simulations produce a temporal evolution that corresponds to the solution of the space-time partial differential equation [10].

In 1976, Daniel T. Gillespie devised an algorithm that provides an exact solution to the chemical master equation using Monte Carlo methods. Named the stochastic simulation algorithm, the formulation is equivalent to the classical KMC algorithm referred to as the Bortz-Kalos-Lebowitiz (BKL) algorithm, or n-fold way, although it was discovered independently. Gillespie proposed two variants of his stochastic simulation algorithm, the direct method and the first reaction method [4,5,7]; these developments of the kinetic Monte Carlo method are presented in sections 3.2 and 3.3 respectively. Improving upon the computational cost associated with Gillespie's algorithms, Michael A. Gibson and Jehoshua Bruck published the next-reaction method in 2000 [3]. The algorithm more efficiently selects a reaction in each iteration of the simulation by using an indexed priority queue or binary tree; the details of this approach are provided in section 2.4. In 2008, Alexander Slepoy, Aidan P. Thompson, and Steven J. Plimpton offered a further refinement to the stochastic simulation algorithm, achieving a computational cost independent of the number of reactions [11]. This constant-time algorithm, which institutes a composition and rejection scheme, is discussed in section 2.5. To conclude the chapter, we explore how the kinetic Monte Carlo method may be implemented to study the evolution of a crystalline structure.

3.2 Gillespie's Algorithm

Suppose we have a spatially homogeneous system of N different chemical species which are subject to the reactions R_i, $i = 1, \ldots, M$. Assume that the system is restricted to a volume V in which the molecular species are in thermal equilibrium. Under this assumption, the molecules are uniformly distributed in V and their velocities follow a Boltzmann distribution; these characteristics permit the model to ignore nonreactive molecular collisions and simply be concerned with those that are reactive [7]. Let $X_j(t)$ denote the number of molecules of species j in the system at time t. To describe the change in the molecular populations at a given time, Gillespie's algorithm seeks to answer two questions: at what time does the next reaction occur and which reaction will it be?

The aforementioned questions may be resolved by determining the probability density $P(\tau, \mu)$ that the next reaction is R_μ and will occur in the time interval $(t + \tau, t + \tau + d\tau)$. This function is a joint probability density function on the space of two random variables, the time until the next reaction τ and the index of the next reaction μ. To derive an exact formula for $P(\tau, \mu)$, we apply the laws of probability to the fundamental hypothesis of chemical kinetics. Considering that a reaction's propensity a_i depends on the physical properties of the involved molecules and the temperature of the system, we define the probability that an R_μ reaction will take place in the next infinitesimal time interval as $a_\mu d\tau$. Furthermore, define $P_0(\tau)$ to be the probability that given that the system is in state $\mathbf{v}$ at time t, no reaction will occur in the time interval $(t, t + \tau)$. Using these definitions, we may write $P(\tau, \mu)d\tau$ as the product of the probability that no reaction occurs within τ time units and the probability that an R_μ reaction occurs in the next $d\tau$ time units:

$$P(\tau, \mu)d\tau = P_0(\tau)a_\mu d\tau. \tag{3.1}$$

To derive an analytical expression for $P_0(\tau)$, we divide the time interval $(t, t + \tau)$

into K subintervals of length $\epsilon = \tau/K$. By the fundamental hypothesis and the multiplication theorem of probability, the probability that no reaction occurs in the time interval $(t, t+\epsilon)$ is given by

$$\prod_{i=1}^{M} (1 - a_i\epsilon + o(\epsilon)) = 1 - \sum_{i=1}^{M} a_i\epsilon + o(\epsilon).$$

As this also provides the probability that no reaction occurs in subsequent subintervals, $P_0(\tau)$ can be expressed as

$$\begin{aligned} P_0(\tau) &= \left[1 - \sum_{i=1}^{M} a_i\epsilon + o(\epsilon)\right]^K \\ &= \left[1 - \sum_{i=1}^{M} a_i\tau K^{-1} + o(K^{-1})\right]^K. \end{aligned}$$

Taking the limit as K approaches infinity, we obtain the probability that none of the M reactions occur in any of the K subintervals:

$$\begin{aligned} P_0(\tau) &= \lim_{K\to\infty} \left[1 - K^{-1}\left(\sum_{i=1}^{M} a_i\tau + o(K^{-1})/K^{-1}\right)\right]^K \\ &= \exp\left[-\sum_{i=1}^{M} a_i\tau\right]. \end{aligned}$$

By inserting this expression into equation (3.1), we find that the reaction probability density function is given by

$$P(\tau, \mu) = a_\mu \exp\left[-a\tau\right], \tag{3.2}$$

where a is the sum of the individual reaction propensities

$$a = \sum_{i=1}^{M} a_i.$$

Note that equation (3.2) is defined for the continuous variable τ, $0 \leq \tau < \infty$, and the discrete variable μ, $\mu = 1, \ldots, M$; for all other values of τ and μ, we define $P(\tau, \mu) = 0$.

Providing the foundation for the stochastic simulation algorithm, $P(\tau, \mu)$ is a dynamic probability distribution based on the propensities and molecular populations

of all reactant species in a chemical system. In order to specify when the next reaction occurs and what reaction it will be, we need to establish a method for generating a sample of τ and μ according to the distribution. The Monte Carlo method offers a framework through which this can be accomplished. As the system takes the form of a continuous-time Markov chain, it exhibits a memoryless property; specifically, the transition probability depends only on the current configuration of the system and not on past history [13]. Through conditioning the probability density function, we obtain

$$P(\tau, \mu) = P_1(\tau) \cdot P_2(\mu|\tau). \tag{3.3}$$

In this formulation, $P_1(\tau)d\tau$ is the probability that the next reaction (of any type) takes place in the interval $(t + \tau, t + \tau + d\tau)$ and $P_2(\mu|\tau)$ is the probability that the reaction is R_μ provided that it occurs in the given time interval. We find $P_1(\tau)$ by summing the probabilities of each reaction, yielding

$$P_1(\tau) = \sum_{\mu=1}^{M} P(\tau, \mu). \tag{3.4}$$

Through solving for $P_2(\mu|\tau)$ in equation (3.3) and substituting in (3.4), we arrive at

$$P_2(\mu|\tau) = \frac{P(\tau, \mu)}{\sum_{i=1}^{M} P(\tau, i)}. \tag{3.5}$$

In the notation of equation (3.2), the distributions can be expressed as

$$P_1(\tau) = a \exp[-a\tau],$$
$$P_2(\mu|\tau) = \frac{a_\mu}{a}.$$

Through this representation, it becomes evident that the next reaction μ is chosen with probability a_μ/a and the time τ until this next reaction happens is exponentially distributed with mean $1/a$.

To choose τ and μ according to the distributions in (3.4) and (3.5), we will draw two uniformly distributed random numbers from the unit interval and employ the

Monte Carlo inversion technique [4]. In this method, x is said to be randomly drawn from a probability density function $P(x)$ if $x = F^{-1}(r)$, where r is a uniformly distributed random number number in the unit interval and $F(x)$ is the cumulative distribution function defined by $F(x) = \int_{-\infty}^{x} P(y)dy$. To derive the expressions by which τ and μ will be determined, define $F_1(\tau)$ to be the cumulative distribution function of $P_1(\tau)$ and $F_2(\mu)$ to be the cumulative distribution function of $P_2(\mu|\tau)$. Then, we have

$$\begin{aligned} F_1(\tau) &= \int_{-\infty}^{\tau} P_1(\gamma)d\gamma \\ &= \int_{0}^{\tau} \sum_i P(\gamma, i)d\gamma \\ &= \int_{0}^{\tau} \sum_i a_i \exp[-\gamma \sum_i a_i]d\gamma \\ &= -\exp[-\gamma \sum_i a_i]\Big|_0^{\tau} \\ &= 1 - \exp\left[-\tau \sum_i a_i\right]. \end{aligned}$$

Drawing a random number r from the uniform distribution in the unit interval, we obtain

$$\begin{aligned} F_1^{-1}(r) &= \frac{1}{\sum_i a_i} \log\left(\frac{1}{1-r}\right) \\ &= \frac{1}{a} \log\left(\frac{1}{r_1}\right), \end{aligned} \tag{3.6}$$

where a is the sum of the reaction propensities and $r_1 = 1 - r$ is a random number in the unit interval.

Since the reactions partition the unit interval according to the size of their propensity function, we can select the reaction by generating a second random number r_2 from the unit interval and taking μ to be the value which satisfies

$$F(\mu - 1) < r_2 \leq F(\mu).$$

Utilizing equations (3.2) and (3.5), we have

$$\begin{aligned} F_2(\mu) &= \int_{-\infty}^{\mu} P_2(\eta|\tau)d\eta \\ &= \int_{-\infty}^{\mu} \frac{P(\tau,\eta)}{\sum_i P(\tau,i)}d\eta \\ &= \int_{-\infty}^{\mu} \frac{a_\eta}{\sum_i a_i}d\eta \\ &= \frac{1}{\sum_i a_i}\int_{-\infty}^{\mu} a_\eta d\eta \\ &= \frac{1}{a}\int_{-\infty}^{\mu} a_\eta d\eta. \end{aligned}$$

Since μ is an integer random variable, the cumulative distribution function takes the form

$$F_2(\mu) = \frac{1}{a}\sum_{i=1}^{\mu} a_i.$$

Therefore, the reaction to occur at time $t+\tau$ is chosen by selecting the integer μ for which

$$\sum_{i=1}^{\mu-1} a_i < ar_2 \leq \sum_{i=1}^{\mu} a_i. \tag{3.7}$$

With a method now established for selecting τ and μ from the correct probability distributions, Gillespie's direct method for simulating the time evolution of a chemical system can be provided. The stochastic simulation algorithm, outlined in Figure 3.1, generates trajectories of the molecular populations in exact correspondence with the chemical master equation (2.10). The calculations of the evolution cease when the number of iterations of the algorithm, n, first exceeds the predetermined stopping time, denoted t_{max}.

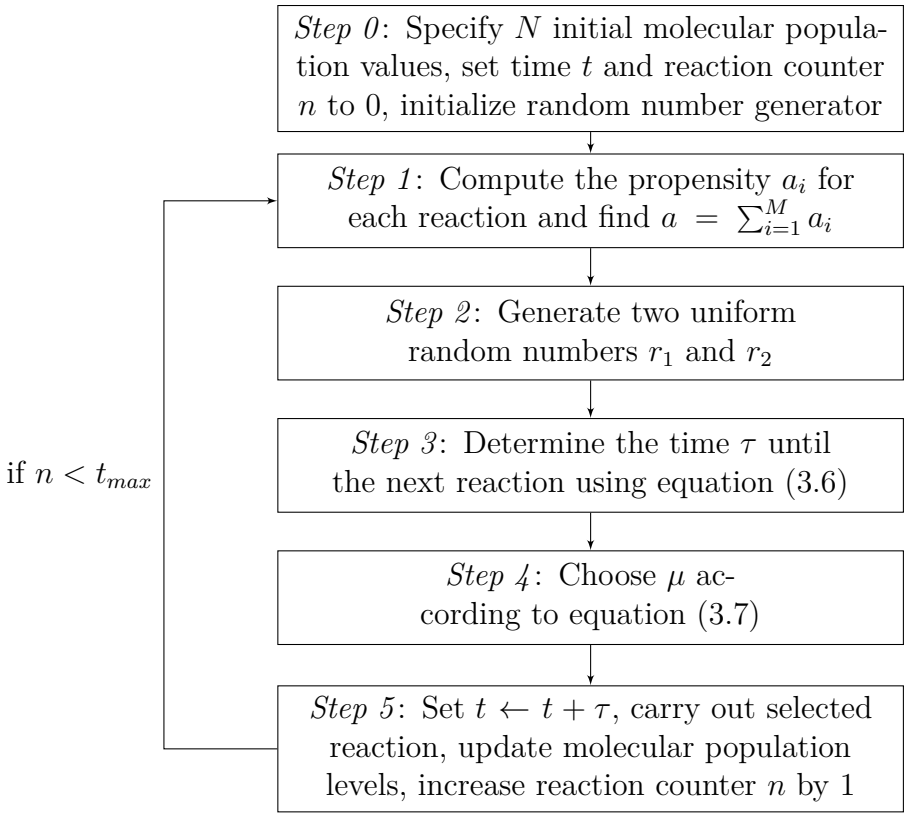

Figure 3.1: Gillespie's stochastic simulation algorithm

Although repeated execution of Gillespie's algorithm produces an average trajectory indicative of a system's behavior, the computational time is dependent upon the number of reactions, resulting in the algorithm being inefficient for large systems. To understand the overall computational cost, consider each step of the algorithm and recall that M provides the total number of different reactions that the system can undergo. In step 1, the calculation of the propensity for each reaction as well as computing the sum over all reactions takes $O(M)$ operations. The generation of the random number r_1 in step 2 allows for the computation of τ in step 3; utilizing equation (3.6), this occurs in $O(1)$ time. The second random number r_2 is used to determine the reaction μ to take place in the time step; as this selection occurs by searching the partitions created by the propensities of the reactions, the inequality given in (3.7) may not be satisfied until $\mu = M$. Finally, performing the selected re-

action in step 5 scales as $O(1)$ as it is assumed that each reaction has a small bounded number of products. Through exploring the generation and update times, it is thus evident that the computational cost of Gillespie's algorithm scales linearly with the number of reactions in the system. As this limits the size of systems that can be efficiently simulated, improvements to the stochastic simulation algorithm have since been made to reduce the computational time.

3.3 Gillespie's First Reaction Method

A variant on Gillespie's direct method, the first reaction method provides the foundation for Gibson and Bruck's logarithmic scaling of the stochastic simulation algorithm. Gillespie's first reaction method differs from the algorithm in the previous section only in the implementation of the Monte Carlo step. Although utilizing the same probability distributions for τ and μ, the method proposes an alternate approach for determining which reaction occurs next, the details of which are to follow.

The foundation for the first reaction method lies in the consideration of $P(\tau, R_i)d\tau$, the probability that the reaction R_i occurs in the time interval $(t+\tau, t+\tau+d\tau)$ provided that no other reaction takes place first. In a derivation mirroring that of equation (3.2), the probability density function is given by

$$P(\tau, R_i)d\tau = a_i \exp[-a_i\tau]d\tau. \tag{3.8}$$

Drawing M uniformly distributed random numbers $r_1, \ldots, r_M$ from the unit interval, we can sample from this distribution by computing

$$\tau_i = \frac{1}{a_i}\log\left(\frac{1}{r_i}\right), \quad i = 1, \ldots, M. \tag{3.9}$$

The values of τ_i provide the relative time to the next occurrence of each reaction; it is these putative times that are the defining characteristic of the first reaction method. As the name alludes, the method selects μ to be the reaction that occurs first, that

is the reaction that has the least putative time. The time until the next reaction τ is then assigned to be τ_μ. Figure 3.2 presents the framework for the algorithm.

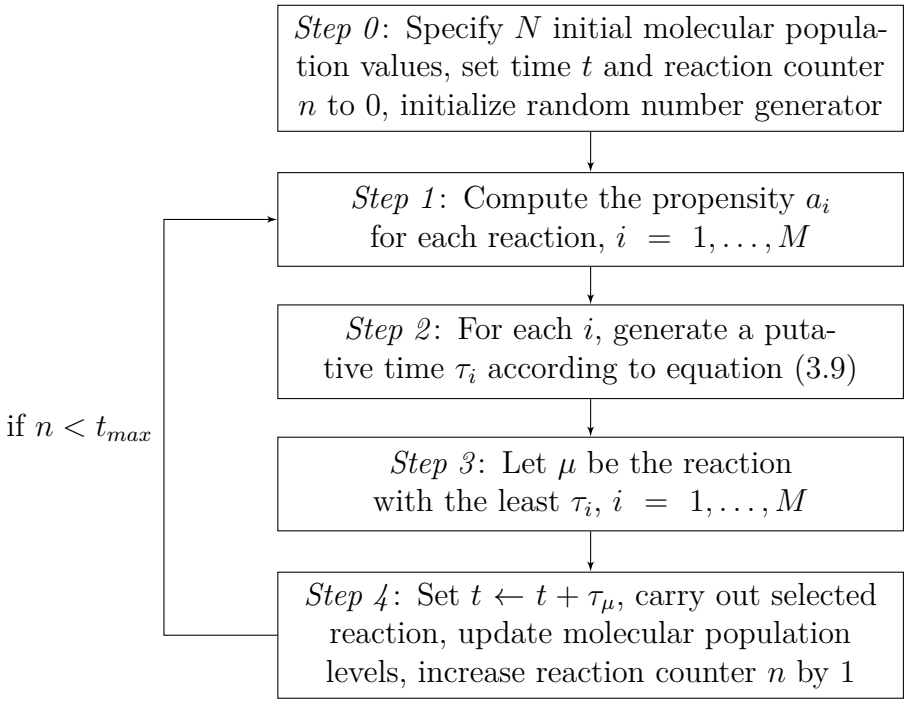

Figure 3.2: Gillespie's first reaction algorithm

The procedure for choosing τ and μ in the first reaction method is found to be an equivalent approach to generating a pair (τ, μ) from the probability density function (3.2). Utilizing M random numbers for the realization of a single chemical reaction, the method, however, does not improve upon the limitations of Gillespie's direct method. In fact, it not only takes time proportional to M to compute a putative time for each reaction but also to determine the smallest putative time; as a result, the first reaction method also scales as $O(M)$.

3.4 Gibson-Bruck Algorithm

The inefficiencies in Gillespie's first reaction method stem from the need in each iteration to update the reaction propensities, calculate a putative time for each reaction, and identify the smallest reaction time. Gibson and Bruck proposed an algorithm,

which they designated the next reaction method, to significantly diminish the computational costs associated with these operations. To more efficiently update propensities following the occurrence of a reaction and establish the time to the next reaction, the method employs two data structures: a dependency graph and an indexed priority queue.

A dependency graph is a directed graph with the vertex set consisting of all the reactions in the network. An edge exists between two reactions R_1 and R_2 if and only if the intersection of Affects(R_1) and Depends(R_2) is not empty. With this construction, a given reaction R_i has only edges oriented towards the reactions whose propensities would be impacted by the execution of R_i. Therefore, the recalculation of propensity functions is only necessary for the reactions contained in the set corresponding to the implemented reaction, leading to this step in the algorithm being of $O(1)$. Figure 3.3 depicts the dependency graph for the Paulsson system in equations (2.3)–(2.5), with the vertices labeled according to the index of the reaction's rate constant.

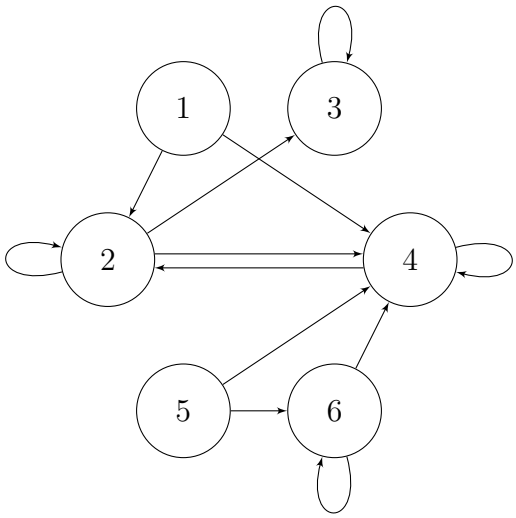

Figure 3.3: Dependency graph of Paulsson system

An indexed priority queue is a tree structure designed to minimize the computational cost associated with identifying the reaction with the lowest putative time. The

nodes of the tree contain the index of a reaction and the time when the next reaction of that type is expected to occur. The tree is then assembled so that the putative time of each parent node is lower than either of its children. As a result, the root of the queue holds the minimum reaction time, thereby supplying the index of the next reaction. Identification of the minimum putative time thus scales as $O(1)$. Updating the queue following the fulfillment of the reaction involves changing the value of the corresponding node and then percolating up or down the structure according to its value. This update procedure avoids the disposal of $M - 1$ reaction times as seen in the first reaction method and can be accomplished in logarithmic time.

The next reaction method differs further from the first reaction method in the use of absolute times rather than relative times. The transformation from exponentially distributed relative times to exponentially distributed absolute times merely arises by adding the elapsed time t to the putative time. Following the execution of the reaction μ, the putative time τ_μ is updated according to

$$\tau_\mu = \frac{1}{a_\mu} \log\left(\frac{1}{r}\right), \tag{3.10}$$

where r is a uniformly distributed random number in the unit interval. For those reactions affected by performing R_μ, the absolute times may be transformed to new values without the need of generating random numbers, a procedure formally justified by Gibson and Bruck [3]. Through consideration of the distributions of the impacted relative times, they propose the following transformation for the absolute time of an affected reaction R_α:

$$\tau_{\alpha,new} = \frac{a_{\alpha,old}}{a_{\alpha,new}}(\tau_{\alpha,old} - t) + t. \tag{3.11}$$

In the above computation, $a_{\alpha,old}$ corresponds to the propensity of R_α prior to amending the molecular population levels and $a_{\alpha,new}$ is the propensity afterwards. For the reactions unaffected by the occurrence of R_μ, modification to the absolute reaction time is not necessary.

With the foundation for updating reaction times and propensities established, the formulation of the stochastic simulation algorithm by Gibson and Bruck may now be provided (Figure 3.4).

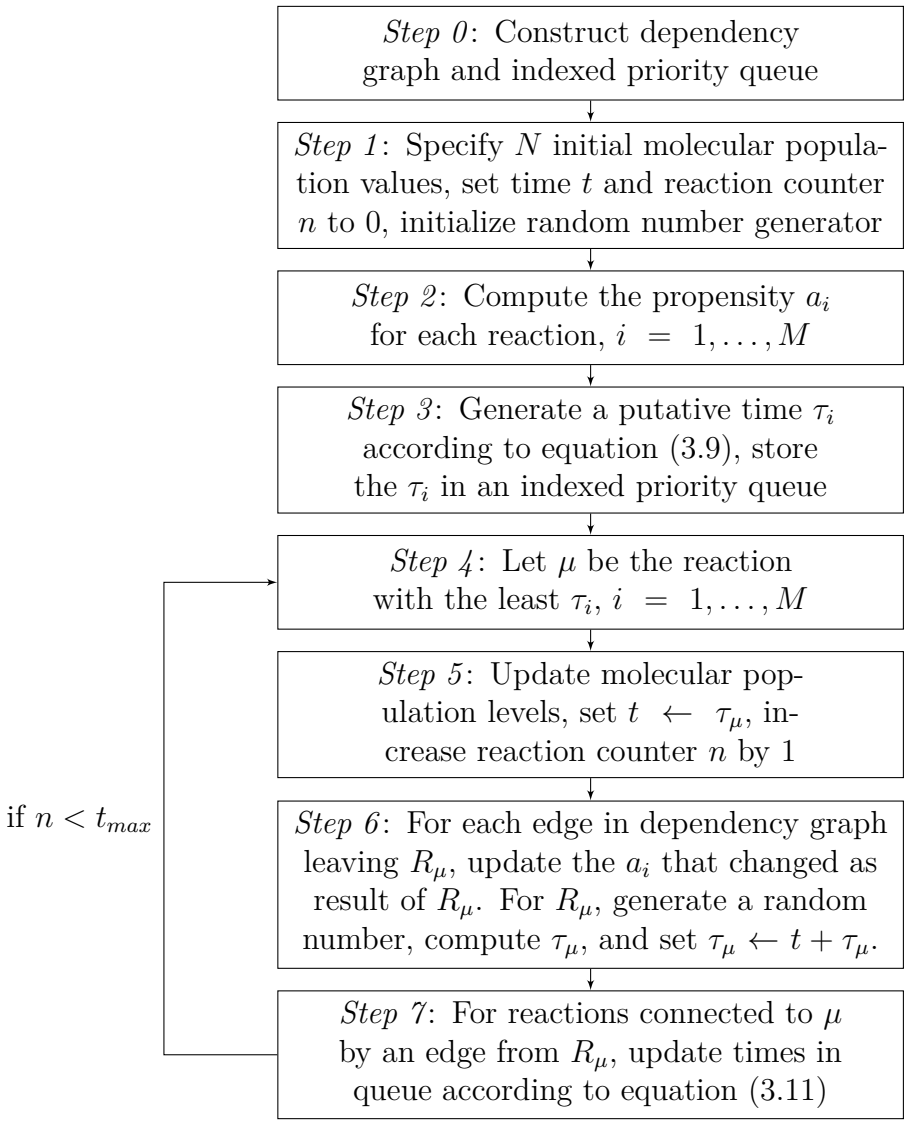

Figure 3.4: Gibson-Bruck next reaction algorithm

Through the addition of a dependency graph and an indexed priority queue, Gibson and Bruck's algorithm achieves a runtime preferable to that of Gillespie's first reaction method. Although the algorithm requires the use of M random numbers in the initialization of reaction times, it only generates one random number per iteration, an improvement upon both of Gillespie's algorithms. The single random number

is used in the computation of the absolute time until the selected reaction is next performed; this operation along with carrying out the reaction scales as $O(1)$. The inclusion of the dependency graph permits the update of propensities following the reaction to also occur in constant time. The algorithm then proceeds by instituting changes to the indexed priority queue in response to the executed reaction. In this tree structure, the number of nodes corresponds to the number of reactions M; since the dependency graph allowed for the recalculation of only the minimal number of propensities, percolating changes through the tree takes at most $\log_2(M)$ operations. The resulting structure contains the minimum τ as the top node, which enables the next reaction μ to be identified in constant time. While producing the same reaction as would result from Gillespie's algorithm, Gibson and Bruck's next reaction method improves the simulation of a single realization of the chemical process to logarithmic time.

3.5 Constant-time KMC

Offering a further enhancement to the stochastic simulation algorithm, Slepoy, Thompson, and Plimpton developed a composition and rejection algorithm with constant-time performance. With scaling independent of the number of reactions, the algorithm allows for efficient modeling of very large networks. The method accomplishes its computational efficiency through the use of a rejection scheme. The foundational data structure for the rejection idea is a bar graph in which the set of M reactions are listed along the x-axis and the y-axis designates the corresponding propensity for each reaction. An example of such a propensity graph is provided in Figure 3.5.

To randomly select a reaction, a rectangle of height $a_{\max}$ is first drawn to bound the M reaction propensities. Two uniform random numbers are then generated: a random integer k from the set $\{1, \ldots, M\}$ and a random number r from the interval $[0, a_{\max}]$. Consider the coordinate (k, r). If the coordinate falls within the bar representing

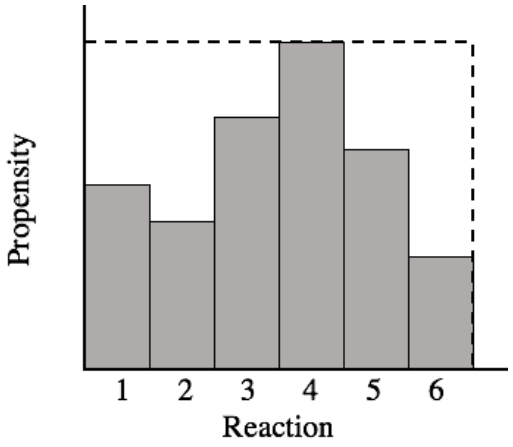

Figure 3.5: Illustration of reaction propensity bar graph

reaction R_k, then the given reaction is selected to be carried out. If the random point (k, r) instead falls outside a vertical bar, then the attempt is rejected. The process is then instituted until a point is obtained inside one of the bars, thereby specifying the reaction to be performed. If the set of bars does not cover a large fraction of the bounding rectangle's area, it is possible that this procedure would require several iterations before an attempt is successful. To limit the number of rejected points, a composition scheme is introduced.

The composition procedure involves grouping the reactions by their propensities. Define $a_{\min}$ to be the minimum propensity value and $a_{\max}$ to be the maximum propensity value of all reactions in the system. Let G be the number of groups. A reaction R_i is assigned to group j if the reaction's propensity a_i falls in the interval $[2^{j-1}a_{\min}, 2^j a_{\min})$. Denote a_g as the sum of all reaction propensities in the group g. In order to choose a group of reactions, draw a random number r_2 from the interval $[0, \sum_i a_i]$ and determine the integer $\hat{g}$ for which

$$\sum_{g=1}^{\hat{g}-1} a_g < r_2 \leq \sum_{g=1}^{\hat{g}} a_g. \tag{3.12}$$

With the group $\hat{g}$ chosen, a reaction is then selected from the group using the rejection algorithm with the height of the bounding rectangle given by $2^{\hat{g}}a_{\min}$. Figure 3.6

illustrates the idea of the composition and rejection scheme for a system of 9 reactions.

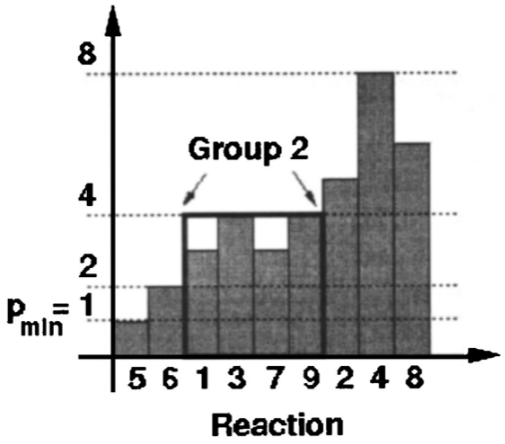

Figure 3.6: Scheme for grouping reactions by propensity from: Alexander Slepoy, Aidan P. Thompson, and Steven J. Plimpton. A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks. *The Journal of Chemical Physics*, 128(20): 205101, 2008, Figure 3.

Through grouping the reactions as described, the set of bars in a given group covers greater than half the area of the group's bounding rectangle. As a result, the selection of a reaction now arises from at most two iterations of the rejection scheme. With the number of groups G independent of the number of reactions M, the composition and rejection component enables reaction selection to scale as constant-time. As in each previous algorithm, the update of reaction propensities follows the execution of a given reaction; however, this scheme additionally entails the consideration of how performing the selected reaction impacts group assignments and each group propensity sum. Those reactions with the potential to change groups may be easily identified through the use of a dependency graph for the reaction system. By comparing the propensity of an affected reaction to its old value, it can be determined whether the reaction remains in the same group or changes groups. As the

number of dependent reactions is bounded, the process of modifying group sums is a constant-time operation.

Exactly equivalent to Gillespie's stochastic simulation algorithm, the composition and rejection algorithm as outlined in Figure 3.7 improves the generation and update of a reaction network to constant-time.

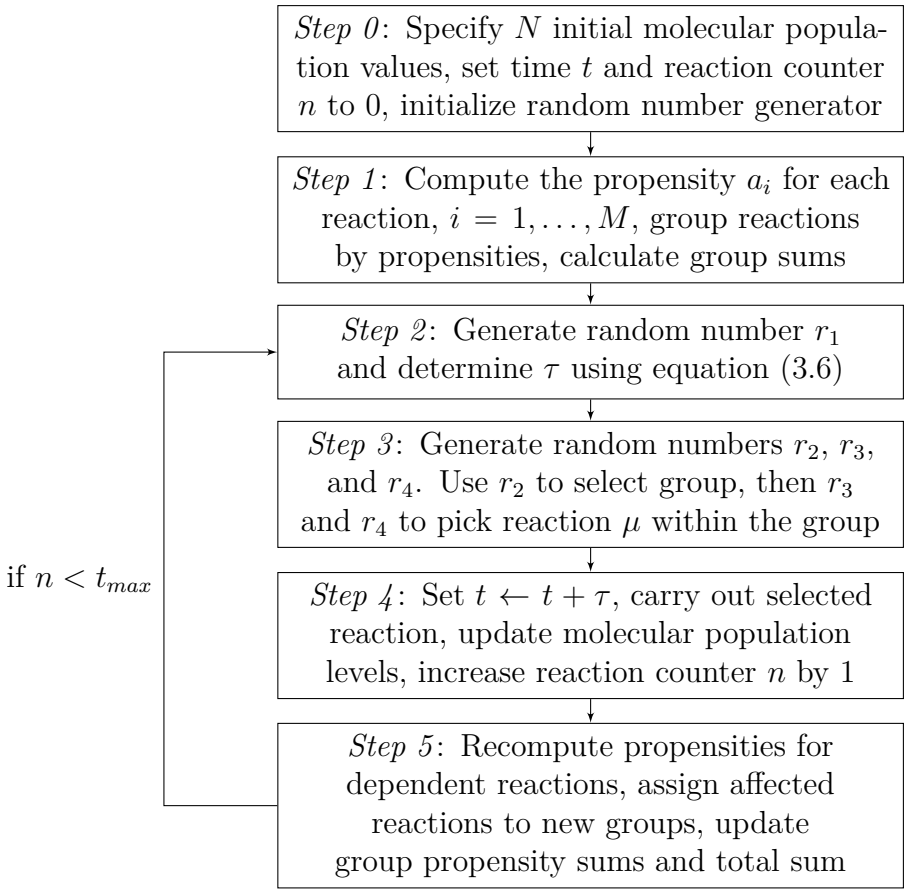

Figure 3.7: Composition and rejection algorithm

Before the first iteration of the algorithm, the initialization for the system is performed, the reactions are assigned to groups, and the group propensity sums are computed. In step 2, it is evident that the computation of the time to the next reaction τ remains unchanged from Gillespie's original algorithm, scaling as $O(1)$. Selecting the reaction μ to carry out in step 3 is accomplished through the use of three random numbers and is now independent of the number of reactions M. The

step may require additional random numbers if rejection occurs; however, the inclusion of the composition scheme enables selection of the reaction to take place in less than two attempts. This composition and rejection aspect of the algorithm scales as $O(1)$. As in each previous algorithm, the execution of the reaction in step 4 also scales as $O(1)$. To update the system after conducting the selected reaction, the propensities of affected reactions are computed, group assignments are evaluated, and group propensities sums are altered, each of which is done in $O(1)$ time. With an overall scaling of $O(1)$, the composition and rejection algorithm thus presents an efficient means for simulating the time evolution of large chemical networks.

3.6 Modifying KMC for Epitaxial Growth

Having established several variants of the stochastic simulation algorithm for chemically reacting systems, we now turn our attention to how kinetic Monte Carlo methods may be applied to epitaxial systems. In order to describe the evolution of a crystalline surface, we define the position of atoms according to a lattice structure and utilize a height array to indicate the surface configuration at any given state. Furthermore, a solid-on-solid approach is assumed in which the crystalline film grows such that one atom can only be accommodated by another atom, thereby prohibiting structural defects like overhangs and vacancies. The surface evolves from state-to-state through the movement of a single atom with the transition dependent on the local crystal configuration.

With state-to-state dynamics corresponding to a Markov walk, a stochastic procedure aptly models the transitions characteristic of an epitaxial system and achieves an efficiency that is orders of magnitude faster than molecular dynamics. The transition rates between states are derived from the system's energy in order to satisfy detailed balance. As the system must overcome an energy barrier to move from one state to another, the rate of the transition is established by the attempt frequency, activation

energy, and temperature, chemical characteristics which are typically related via an Arrhenius equation. According to the Arrhenius equation, a chemical reaction occurs at the rate

$$\kappa = \omega \exp\left[-\frac{E_D}{k_B T}\right], \tag{3.13}$$

where ω is an exponential pre-factor, E_D is the energy barrier for diffusion needed for the reaction to take place, k_B is the Boltzmann constant, and T is the temperature in Kelvin [1]. The parameter ω reflects the thermal vibration frequency of the system; specifically, it represents the number of collisions of atoms per second with the ability to react. The Arrhenius relation, equation (3.13), forms the basis for constructing the rates associated with the atomistic processes, such as deposition and diffusion, that an epitaxial system is experiencing. Additional specifications, however, will need to be introduced in order to precisely define the allowed movements and the nearest-neighbor attractions in each process, or event, considered.

As in previous kinetic Monte Carlo algorithms, propagating an epitaxial system forward in time requires event selection and updating of data structures in each iteration of the algorithm. With the atoms of an epitaxial system defined to occupy positions on a lattice, the KMC variant introduced by Bortz, Kalos, and Lebowitz [2], termed the n-fold way, provides an appropriate scheme. The n-fold way algorithm was formulated to simulate an Ising spin system, which is a d-dimensional lattice where each site is assigned a spin variable assuming the value of $+1$ or -1 . A given site moves to a new state by flipping its spin from one value to the other; this is accomplished by either reversing its own spin or by interchanging its spin with the unlike spin of a neighbor. In the case of an epitaxial system, the spin variables may indicate the presence $(+1)$ or absence (-1) of an atom in a given lattice site; a spin interchange would, therefore, correspond to the diffusion of an atom.

The n-fold way algorithm is constructed by determining the M possible events in the system and forming a list of the expected rates for the system to move from

one state to another. Let r_i denote the rate of event i, which coincides with the probability of a spin flip. The lattice structure of the system permits the grouping of events with the same probability to form n classes; it is this characteristic that lends to the algorithm's designation as the n-fold way. Suppose that n_j denotes the number of events in class j and P_j represents the probability of spin interchange shared by each event in the class. Then, the total of all possible possible outcomes in the next time interval is given by

$$R = \sum_{i=1}^{M} r_i = \sum_{j=1}^{n} n_j P_j. \tag{3.14}$$

Drawing a random number ρ_1 from the uniform distribution in the unit interval, the class α is selected for which

$$\sum_{j=1}^{\alpha-1} n_j P_j < \rho_1 R \leq \sum_{j=1}^{\alpha} n_j P_j. \tag{3.15}$$

An event from the class α is then chosen by generating a random integer ρ_e from the set $\{1, \ldots, n_\alpha\}$.

Choosing an event in this manner mirrors the method used to select a reaction in previous algorithms and guarantees that faster events are chosen with greater probability. After the event is selected and the system is modified to reflect the execution of the given event, the rate list must be updated. Since the change in the state of the system is limited to only the flipped site and the nearest neighbors, we can exploit the locality of rates to perform updates, requiring modification of only the minimal number of rates. The time for the chosen event to transpire following the previous event is determined by generating a random number ρ_2 in the unit interval and computing

$$\tau = \frac{1}{R} \log\left(\frac{1}{\rho_2}\right). \tag{3.16}$$

The overall simulation time is then advanced by the increment τ. The process of generating an event and the time for the event to occur as well as instituting the

appropriate changes to the data structures comprises one iteration of the n-fold way algorithm, which is presented in detail in Figure 3.8.

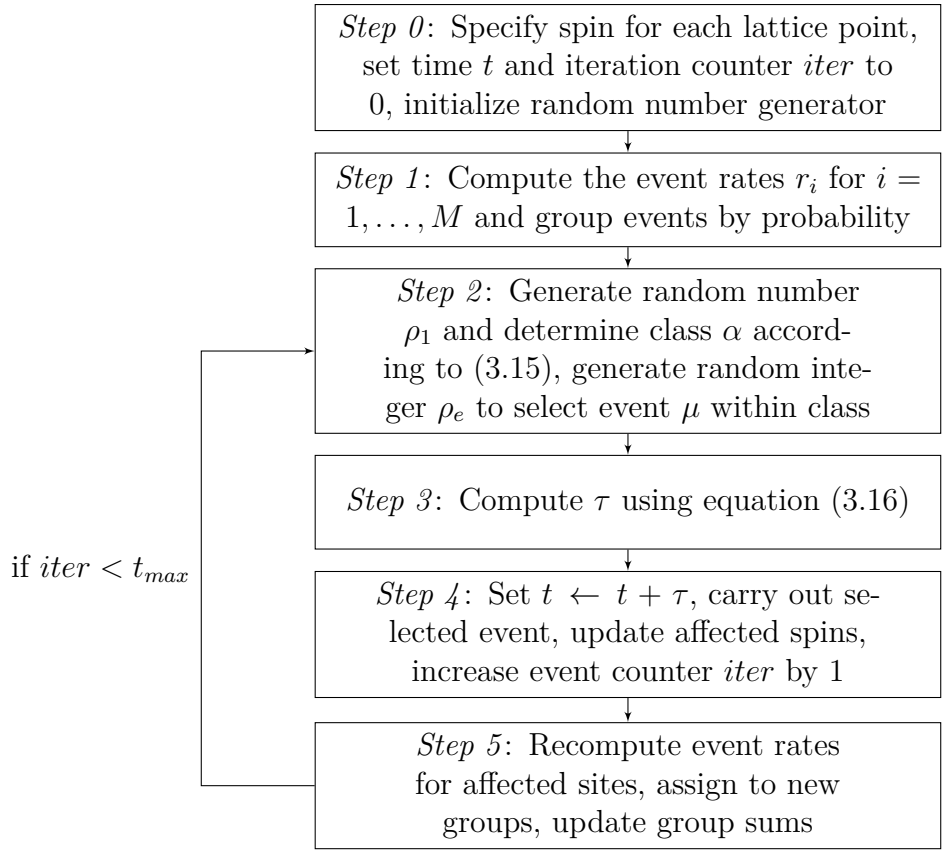

Figure 3.8: n-fold way for Ising spin systems

The n-fold way algorithm is said to be rejection-free as each iteration advances the system to a new state. In the case of epitaxial growth, repeated iteration of the algorithm produces a time evolution in the surface morphology of the modeled crystalline structure. The algorithm, as it appears in Figure 3.8, scales as $O(M)$. This computational cost results from the method by which the event class α is chosen; as this occurs through a probability weighted selection by partial sums (3.15), the operation scales as the uppermost limit of sums, which is the total number of events M. Selecting the particular event, computing τ, conducting the event, and updating the data structures each scale as $O(1)$.

The procedure detailed in this section offers a foundation for simulating the growth

of a crystalline film. As such, the framework will need to be built upon in order to incorporate the atomistic processes, dimensions, and materials one desires to model. Consideration should be provided, however, to the computational cost associated with realizing a given model; for instance, simulating a strained epitaxial system presents greater computational difficulty than a system without strain. In the next chapter, we will develop and implement a 1+1 dimensional model of homoepitaxial growth.

Chapter 4

Implementing a Model of Epitaxial Growth

4.1 Construction of Model

Using the foundation discussed in the last section of the preceding chapter, we will now implement the kinetic Monte Carlo method for a 1+1 dimensional model of epitaxial growth. We utilize a solid-on-solid bound counting scheme in which the atoms occupy positions on a simple cubic lattice and interact with nearest neighbors. The crystalline surface evolves through the execution of the atomistic processes of hopping, evaporation, and condensation. The rates of these processes are derived from transition state theory and thereby satisfy detailed balance. The kinetic Monte Carlo method provides a probabilisitic structure for selecting the lattice site and the event type that results in a change to the surface configuration of the crystal in the neighborhood of the chosen site.

To form the basis of the model, consider the cross-section of a homoepitaxial film which contains N atoms distributed across M active sites. Each lattice site is occupied by multiple atoms which is interpreted as layers on top of the substrate. The profile of the film is described by an array of heights $h = (h_0, h_1, \ldots, h_M)$, where h_i is the height at site i defined as the number of atoms exceeding the baseline height $h_0 = 0$ of the ghost site $i = 0$. Figure 4.1 provides a visual of the manner in which the heights of the structure are configured. The atoms which occupy the highest position at a given lattice site are surface atoms; it is these atoms which have the ability to move and change the surface configuration of the crystal. As the rates for hopping

and evaporation depend on the number of bonds that each surface atom possesses, the construction of heights permits the number of lateral bonds n_i and the number of surface bonds n_i^s at each site i to be easily determined. Each surface atom contains at minimum one surface bond due to the connection with an atom below; surface atoms, however, may be without any lateral bonds, in which they are termed adatoms. The lattice site $i = 2$ in Figure 4.1 provides an example of a surface atom which has no neighbors with which it shares bonds.

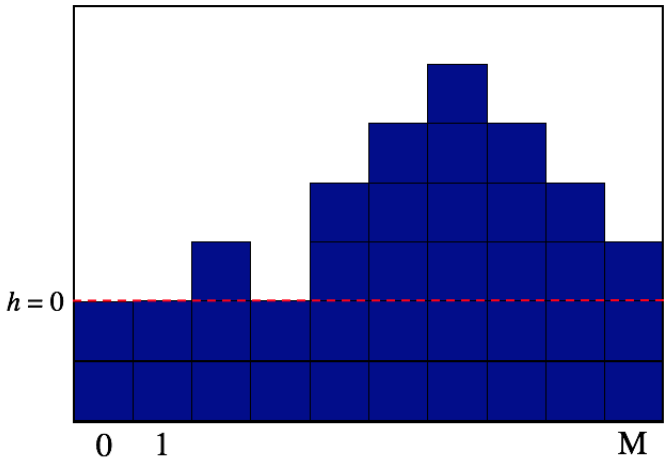

Figure 4.1: Diagram of lattice structure and height configuration

Surface atoms in the described system may undergo one of three events: hopping, evaporation, or condensation. The Arrhenius relation, equation (3.13), provides the foundation for the rates at which these transitions occur. Hopping refers to the lateral movement, or diffusion, of a surface atom. Without a boundary, a surface atom hops to the left or to the right with equal probability. In order to diffuse to a neighboring site, a surface atom must surmount the energy dividing the current state from the transition state. Let E_B denote the difference in energy between the current and transition states and ΔE denote the difference in energy between the current and final states. Then $E_B = -\Delta E - E_0$, where E_0 arises from considering the additional energy needed for the atom to be completely dismissed from the system [12]. Since

hopping requires the breaking and then the formation of bonds, the rate at site i is dependent on the number of surface bonds held by the atom. Defining γ to be the energy per unit area released by the formation of a bond, it follows that $\Delta E = -\gamma n_i^s$. With surface atoms at sites $i = 2, \ldots, M-1$ having the ability to hop to the left or to the right, the hopping rate is scaled by a factor of 2. Incorporating each of these considerations, the hopping rate of the ith surface atom for $i = 1, \ldots, M$ is given by

$$r_i = 2\omega \exp\left[-\frac{E_B}{k_B T}\right] = 2\omega \exp\left[\frac{E_0 - \gamma n_i^s}{k_B T}\right]. \tag{4.1}$$

Recognizing that $E_0 = -E_D + \gamma$ and that the number of lateral bonds is one less than the number of surface bonds ($n_i = n_i^s - 1$), the hopping rate can be more simply expressed as

$$r_i = 2\omega \exp\left[-\beta(E_D + \gamma n_i)\right], \tag{4.2}$$

where $\beta = \frac{1}{k_B T}$. If the surface atom at site i is an adatom, the hopping rate further reduces to $r_i = 2\omega \exp[-\beta E_D]$.

Evaporation results from a surface atom breaking bonds with neighboring atoms and withdrawing from the surface to enter a reservoir of atoms. The rate at which evaporation of the ith surface atom occurs is given by

$$v_i = \omega_E \exp[-\beta \gamma n_i], \tag{4.3}$$

in which the pre-factor ω_E incorporates activation energy. Condensation involves the deposition of atoms from the reservoir onto the crystalline surface. As atoms in the reservoir have not yet formed bonds with atoms on the surface, the parameter γ and term n_i are absent from the condensation rate. To describe the chemical potential between the reservoir and the crystalline surface, we define the parameter μ; an atom must overcome this potential in order to deposit onto the surface. The condensation rate for lattice site i, $i = 1, \ldots, M$, is thus provided by

$$c_i = \omega_E \exp[-\beta \mu]. \tag{4.4}$$

To model the surface evolution of a crystalline structure based on the prescribed atomistic processes, each iteration of the kinetic Monte Carlo algorithm requires event selection. For the model presented, this includes the selection of one of the M lattice sites as well as the process to occur at the given site. Consider the sum of all rate processes across all sites:

$$Z = \sum_{i=1}^{M} (r_i + v_i + c_i). \tag{4.5}$$

Drawing a random number ρ_1 from the uniform distribution in the unit interval, choose the site α which satisfies

$$\sum_{i=1}^{\alpha-1} (r_i + v_i + c_i) < \rho_1 Z \leq \sum_{i=1}^{\alpha} (r_i + v_i + c_i). \tag{4.6}$$

The event type is then randomly selected with probability proportional to the rates at the indicated site.

The selected event determines the changes required to the surface configuration at site α. If the chosen event is evaporation, the surface atom at site α is simply removed, and if the chosen event is condensation, an atom is added at the site. If the selected event is hopping, the following scheme is used to randomly determine whether the atom hops to the left, to the right, or remains in place:

1. If $2 \leq \alpha \leq M - 1$, then the atom hops with probability 0.5 to either site $\alpha + 1$ or site $\alpha - 1$.
2. If $\alpha = 1$, then the atom hops to the right with probability 0.5 or does not hop with probability 0.5.
3. If $\alpha = M$, then the atom hops to the left with probability 0.5 or does not hop with probability 0.5.

If a hopping event is executed, an atom is removed from site α and one is added to the chosen neighboring site. After updating the height array to reflect the selected event, the crystal's energy is computed as

$$E(h_1, \ldots, h_M) = \gamma N - \gamma \sum_{i=1}^{M} \min_i (h_i, h_{i-1}). \tag{4.7}$$

The change in the crystal's energy is indicative of the process that the surface underwent. Energy is embodied by the number of bonds within the crystal; forming new bonds releases energy whereas breaking bonds requires added energy.

4.2 Model Algorithm

Given an initial configuration of N atoms distributed across M sites, the evolution in the surface structure of an epitaxial film can be modeled through the repeated implementation of the algorithm presented in Figure 4.2.

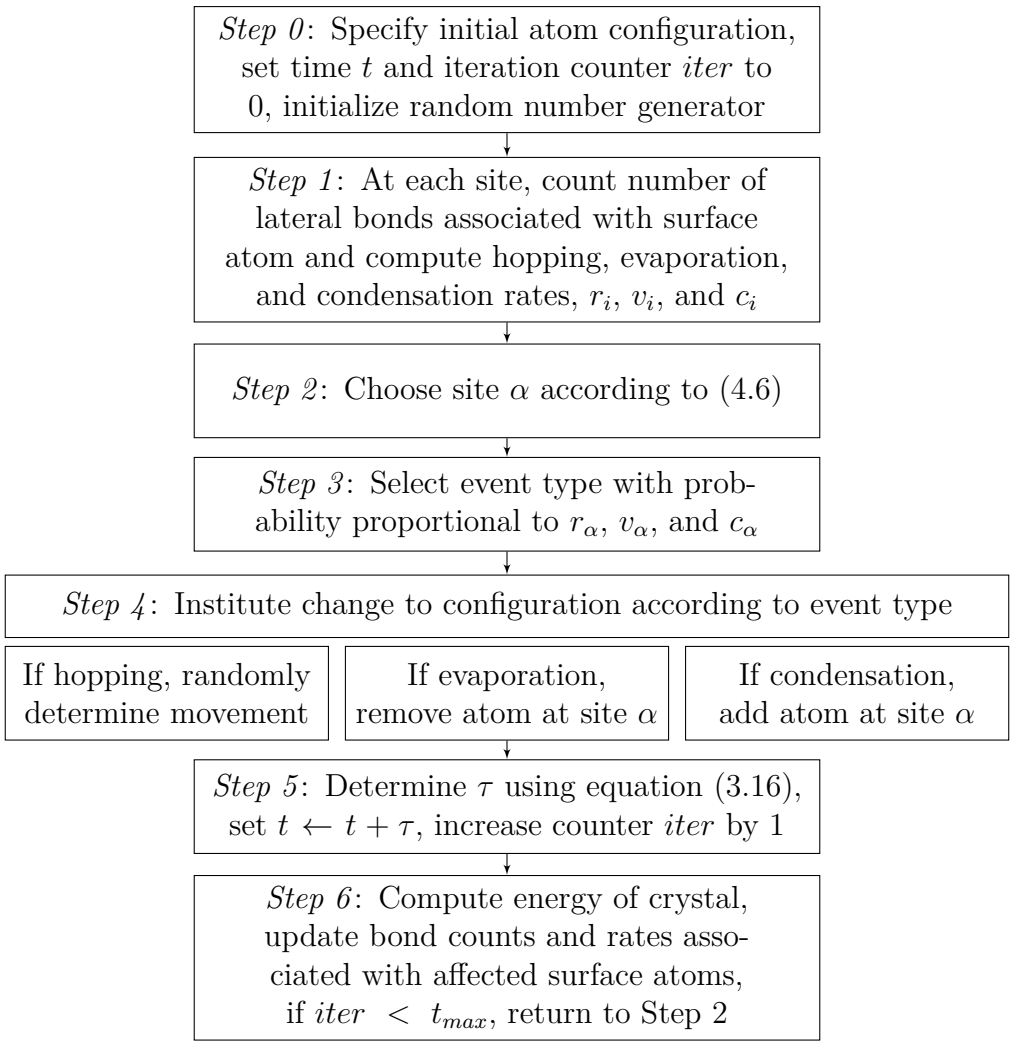

Figure 4.2: KMC algorithm for modeling 1+1 dimensional epitaxial growth with the events of hopping, evaporation, and condensation

The computational cost of the update stage (step 6) can be reduced by exploiting the locality of rates. The events of evaporation and condensation impact only the bond counts and event rates of the selected site α and the two neighboring sites, $\alpha - 1$ and $\alpha + 1$. The execution of a hop, however, can influence the rates of up to four sites: the site α, the site $\alpha - 1$, the site $\alpha + 1$, and either the site $\alpha - 2$ or $\alpha + 2$ depending on the direction of the hop. If the selected site α was either site 1 or site M, then hopping to the interior site requires updates to only three sites and failure to hop yields no changes.

4.3 Results of Model

To illustrate the execution of the aforementioned algorithm, the parameter values in Table 4.1 were utilized in the rate expressions. Aside from the Boltzmann constant, these values represent the chemical characteristics of silicon, an element widely used in the computer and microelectronic industries. The model can be altered to simulate the growth of a different material by updating the appropriate parameter values.

Table 4.1: Parameter values for use in model implementation [14]

Parameter	**Symbol**	**Value**
Number of collisions per second	ω	$1\ \text{s}^{-1}$
Energy barrier for diffusion	E_D	1.2 eV
Bond energy	γ	2.22 eV
Boltzmann constant	k_B	8.6173303×10^{-5} eV/K
ω with activation energy	ω_E	$1\ \text{s}^{-1}$
Chemical potential	μ	2γ eV

As the form of an epitaxial layer is dependent on growth temperature, we explore such an influence in simulations of the model by varying the value of T, the temperature parameter which contributes to each of the rate processes. To allow for comparison in the resulting surface morphology, we perform the simulations using

the same beginning configuration of atoms. This initial configuration, Figure 4.3, was obtained by randomly distributing 700 atoms across 50 lattice sites.

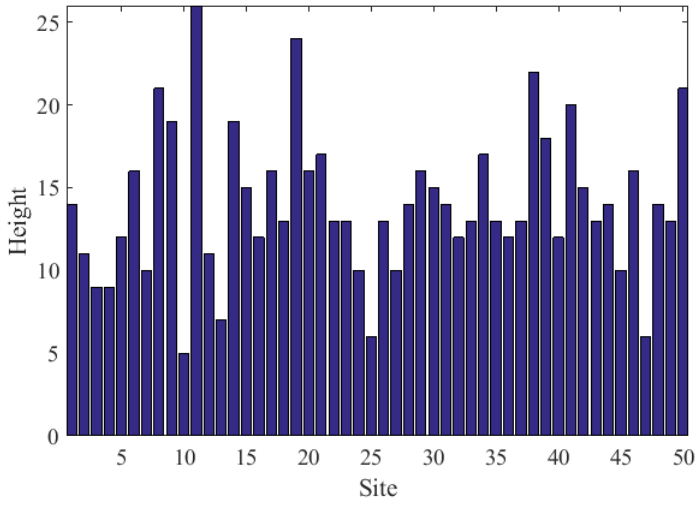

Figure 4.3: Initial configuration of atoms in simulation

Through the implementation of the algorithm outlined in Figure 4.2, we simulate the evolution of this initial crystal surface and the change in the system's energy brought about by each individual event. Ordered by decreasing temperature, Figures 4.4 – 4.6 display the atom configuration resulting after 500 iterations of the algorithm as well as a visual of the transformation in the structure's energy levels over the course of the simulation. With the computation of time determined by a random number and independent of the event chosen, the iterations of the algorithm do not represent equal spacings of time; the duration of the simulation may therefore correspond to time periods that vary according to growth temperature.

The initial configuration of atoms in Figure 4.3 resembles a rough, jagged crystalline surface. Implementing the kinetic Monte Carlo scheme at the melting point of silicon, the crystalline surface achieves a smooth, rounded appearance by the end of the simulation (Figure 4.4). At high temperatures, the hopping rate is elevated and atoms diffuse to more stable positions in order to reduce the system's energy, leading to the development of flatter surfaces with less structural defects. The resulting

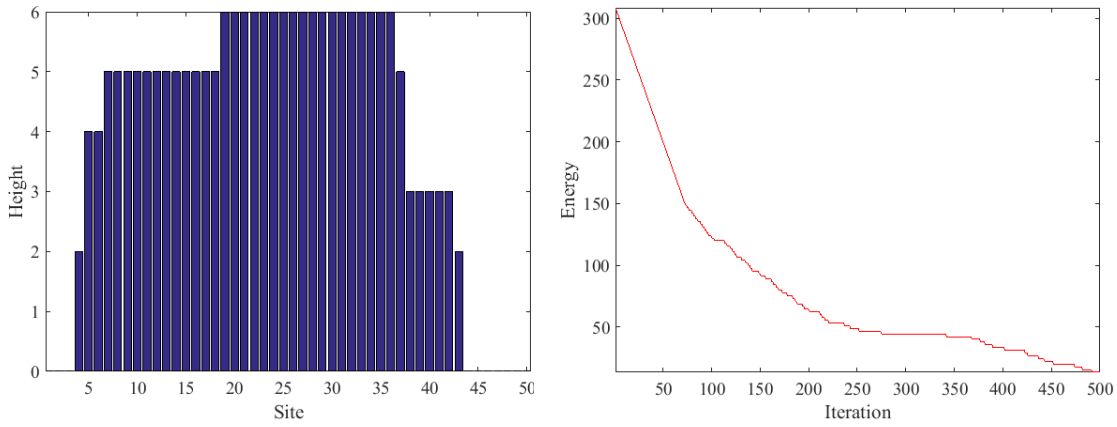

Figure 4.4: Configuration and energy levels for simulation at melting point

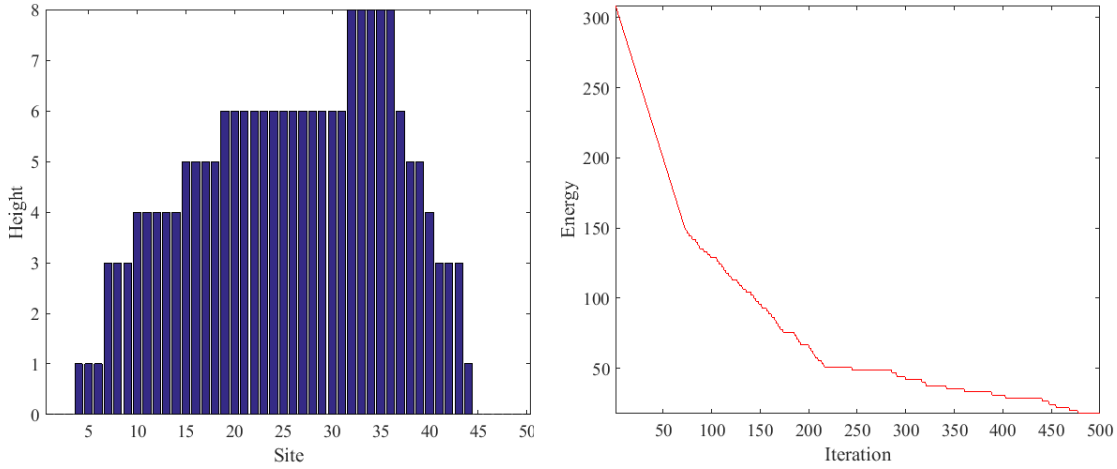

Figure 4.5: Configuration and energy levels for simulation at room temperature

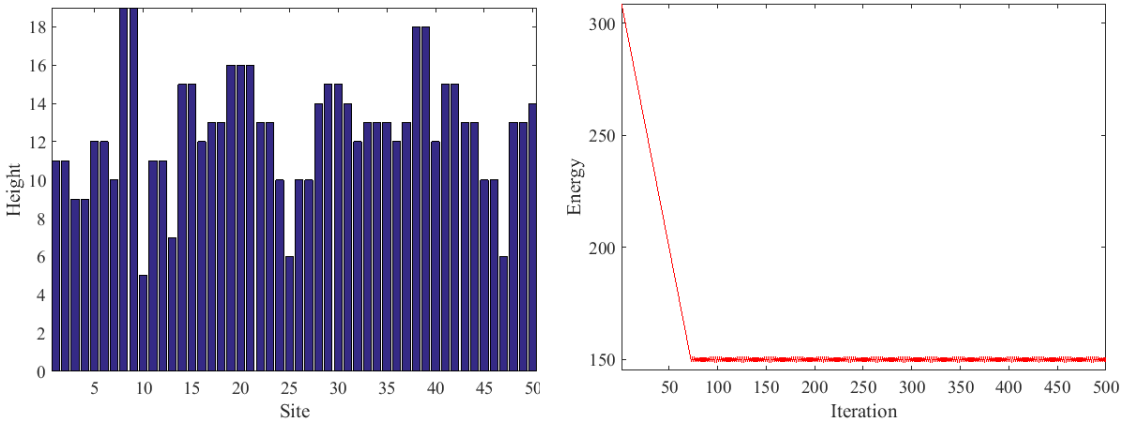

Figure 4.6: Configuration and energy levels for simulation at very low temperature

configuration and decreasing energy levels in Figure 4.4 mirror this behavior. The simulation producing Figure 4.5 was carried out at room temperature. The surface structure assumes a rounded orientation but with more frequent changes in height at the edges. Analyzing the energy levels over the course of the simulation, a decrease in energy is again realized; the leveling of the energy at the final iterations indicates the stabilization of the crystalline surface. By drastically reducing the growth temperature, a rounded surface structure is no longer obtained (Figure 4.6). Low temperatures inhibit the mobility of atoms, which results in little to no diffusion and gives rise to a rough surface. The plot of energy levels reflects the limited mobility of atoms as after approximately 75 iterations of the algorithm, the occurring kinetic processes are unable to further reduce the energy of system. The energy for the remainder of the simulation remains rather stagnant and at a value higher than that of the previous realizations.

The implementation of the kinetic Monte Carlo method in this section demonstrates the dependence of surface morphology on growth temperature. Considering the prevailing use of silicon as a semiconductor material, the exploration of the effect of various temperatures on the resulting structure offers an important application. Examining the growth conditions of other materials can easily be accomplished in a similar manner. The framework of the presented model can also be built upon to incorporate additional atomistic properties and rules for movement. Extending the model to include heteroepitaxial growth, however, poses greater difficulty; the lattice mismatch presented by the use of differing materials for film and substrate requires the incorporation of an elastic energy into the hopping rate. The computational cost associated with realizing such a model is orders of magnitude larger than any of the algorithms discussed herein.

Bibliography

[1] Corbett C. Battaile. The kinetic Monte Carlo method: foundation, implementation, and application. *Computer Methods in Applied Mechanics and Engineering*, 197(41): 3386–3398, 2008.

[2] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz. A new algorithm for Monte Carlo simulation of Ising spin systems. *Journal of Computational Physics*, 17(1): 10–18, 1975.

[3] Michael A. Gibson and Jehoshua Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The Journal of Physical Chemistry A*, 104(9): 1876–1889, 2000.

[4] Daniel T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4): 403–434, 1976.

[5] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25): 2340–2361, 1977.

[6] Daniel T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A*, 188(1): 404–425, 1992.

[7] Daniel T. Gillespie. Stochastic simulation of chemical kinetics. *Annual Review of Physical Chemistry*, 58(1): 35–55, 2007.

[8] D. W. Pashley. Epitaxy growth mechanisms. *Materials Science and Technology*, 15(1): 2–8, 1999.

[9] Johan Paulsson, Otto G. Berg, and Måns Ehrenberg. Stochastic focusing: fluctuation-enhanced sensitivity of intracellular regulation. *Proceedings of the National Academy of Sciences*, 97(13): 7148–7153, 2000.

[10] Tim P. Schulze. Efficient kinetic Monte Carlo simulation. *Journal of Computational Physics*, 227(4): 2455–2462, 2008.

[11] Alexander Slepoy, Aidan P. Thompson, and Steven J. Plimpton. A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks. *The Journal of Chemical Physics*, 128(20): 205101, 2008.

[12] Peter Smereka. Theory and application of simplified kinetic Monte Carlo models [presentation]. *Institute for Pure and Applied Mathematics*, 2012.

[13] Arthur F. Voter. Introduction to the kinetic Monte Carlo method. *Radiation Effects in Solids*, 235: 1–23, 2007.

[14] Dylana Wilhelm, Benjamin Dulaney, and Jenny Frank. Kinetic Monte Carlo simulations for one-dimensional homoepitaxial and heteroepitaxial crystalline structures. Unpublished manuscript, 2017.