Reduced Moment-Based Models for Oxygen Precipitates and Dislocation Loops in Silicon

Bart Trzynadlowski

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington
2013

Reading Committee:
Scott Dunham, Chair
Robert Darling
Jason Guo

Program Authorized to Offer Degree:
Electrical Engineering

University of Washington

Abstract

Reduced Moment-Based Models for Oxygen Precipitates and Dislocation Loops in Silicon

Bart Trzynadlowski

Chair of the Supervisory Committee:
Professor Scott T. Dunham
Electrical Engineering

The demand for ever smaller, higher-performance integrated circuits and more efficient, cost-effective solar cells continues to push the frontiers of process technology. Fabrication of silicon devices requires extremely precise control of impurities and crystallographic defects. Failure to do so not only reduces performance, efficiency, and yield, it threatens the very survival of commercial enterprises in today's fiercely competitive and price-sensitive global market.

The presence of oxygen in silicon is an unavoidable consequence of the Czochralski process, which remains the most popular method for large-scale production of single-crystal silicon. Oxygen precipitates that form during thermal processing cause distortion of the surrounding silicon lattice and can lead to the formation of dislocation loops. Localized deformation caused by both of these defects introduces potential wells that trap diffusing impurities such as metal atoms, which is highly desirable if done far away from sensitive device regions. Unfortunately, dislocations also reduce the mechanical strength of silicon, which can

cause wafer warpage and breakage. Engineers must negotiate this and other complex tradeoffs when designing fabrication processes. Accomplishing this in a complex, modern process involving a large number of thermal steps is impossible without the aid of computational models. In this dissertation, new models for oxygen precipitation and dislocation loop evolution are described.

An oxygen model using kinetic rate equations to evolve the complete precipitate size distribution was developed first. This was then used to create a reduced model tracking only the moments of the size distribution. The moment-based model was found to run significantly faster than its full counterpart while accurately capturing the evolution of oxygen precipitates. The reduced model was fitted to experimental data and a sensitivity analysis was performed to assess the robustness of the results. Source code for both models is included.

A moment-based model for dislocation loop formation from {311} defects in ion-implanted silicon was also developed and validated against experimental data. *Ab initio* density functional theory calculations of stacking faults and edge dislocations were performed to extract energies and elastic properties. This allowed the effect of applied stress on the evolution of {311} defects and dislocation loops to be investigated.

DEDICATION

To my beloved parents, Dorota and Andrzej, for all their sacrifices and unconditional love. These trying times have opened my eyes to the countless ways they have shaped my character.

To my sister, Nicole, who is an amazing young woman on the threshold of an exciting new chapter in life. I know she will reach great heights!

To *Dziadek*, who would have been so proud. His example will always guide me.

And to Janet, for her love, wisdom, and all the little ways she brightened each and every day 8+)

ACKNOWLEDGMENTS

I must first and foremost express my gratitude to my adviser, Prof. Scott Dunham, whose mentorship has made the past six-and-a-half years an enriching and formative experience. When I first arrived at the University of Washington in 2006, I could not tell a vacancy from a hole in the ground, but thanks to Scott's guidance and his impressive knowledge, I have learned more than I ever imagined possible about a dizzyingly complicated subject matter.

The Electrical Engineering faculty and staff are outstanding and have been a pleasure to interact with and learn from. I would especially like to thank Prof. Bruce Darling for his enormously helpful advice and agreeing to serve on my reading committee at the last minute; Frankye Jones, for all her help and genuine concern; and Bryan Crockett and Alex Llapitan, for helping resolve an 11th hour crisis. I also greatly appreciate all of my committee members, who had to endure a very long defense: Profs. "Anant" Anantram, Karl Böhringer, Jason Guo, and from the Mechanical Engineering department, Prof. Jaehyun Chung, who was willing to serve as my Graduate School Representative on short notice.

My trips to Asia opened a window in my mind and were truly a life-changing experience. I am thankful to both Karl and Scott for arranging an unforgettable summer at the University of Tokyo in 2008. I am also grateful to Scott for alerting me to the National Science Foundation's East Asia and Pacific Summer Institutes (EAPSI) program, which made possible my summer at National Tsing Hua University (NTHU) two years later, and to Prof. Yu-Lun Chueh for his hospitality there. I am especially thankful to Scott Hou for sacrificing what little time he had to help get me settled in at NTHU and the Taiwanese student hosts in the EAPSI program for going out of their way to arrange extraordinary outings.

I would like to thank all of my remarkable colleagues in the Nanotechnology Modeling Laboratory for their time, their patient explanations, and their friendship: Wenjun Jiang, Renyu Chen, Baruch Feldman, Kjersti Kleven, Haoyu Lai, Chihak Ahn, Jason Guo, and Armin Yazdani. I wish them all great success in their personal and professional lives. Each and every one of them has earned it!

Although unrelated to my graduate research, my internship at NVIDIA in the fall of 2012 helped me regain my confidence during the kind of low point that every graduate student

experiences. I thank Guy Peled, my manager, and Eric Chi, my mentor, for facilitating an outstanding experience there.

Funding for my research was generously provided by the Semiconductor Research Corp., the Silicon Wafer Engineering and Defect Science program, Sony, and the Silicon Solar Consortium. *Ab initio* calculations were performed on cluster nodes donated by Intel and AMD, and on a Pacific Northwest National Laboratory supercomputer.

Most of all, I am thankful for all the unconditional love, encouragement, and support I have received from my family: my mother, Dorota; father, Andrzej; and sister, Nicole. Even from afar, they were with me every step of the way, sharing not only in my small triumphs but in all my frustrations and anxieties. And finally, I would also like to say a very special "thank you" to Janet, who has been an unfaltering pillar of love, support, and wisdom these past two years. I hope that distance will never break the bond we formed.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER 1
INTRODUCTION

The objective of this dissertation is to develop computational models for two important defects that occur in silicon: oxygen precipitates and dislocation loops. Process models and technology computer-aided design (TCAD) tools have long been used by the integrated circuit (IC) and photovoltaic industries to optimize device structures, dopant profiles, and device performance. The motivation is obvious: simulation is cheaper, faster, and easier than iterative fabrication and testing. Unfortunately, predictive models are inherently difficult to develop. Empirical models have limited applicability while models based on physical laws often depend on parameters that are impossible to measure directly (e.g., atomic-scale hopping frequencies and binding energies). This results in models that often fall short of their promise.

In recent years the predicted end of Moore's Law scaling has spurred a renewed interest in TCAD for designing complex device structures and wringing additional performance out of existing processes. Silicon solar cells also provide fertile territory for such research. Improvements in efficiency are slow and incremental, and competition is intense. Some commercial products already approach theoretical maximum efficiencies achieved in the laboratory. Even modest reductions in unwanted defects or improvements in yield can be lucrative.

Phase transformations (e.g., oxygen precipitation) and extended defects (e.g., dislocation loops) are especially challenging to model. These phenomena are inherently complex, involving huge numbers of atoms and potential reaction pathways, and are governed by a large number of factors that are impossible to measure experimentally. Several models have been developed over the decades, each with their own strengths and weaknesses. This work demonstrates simplified models derived from fundamental precipitation kinetics. The criteria for success are accuracy, robustness, extensibility, and usability of the models.

Accuracy is judged by how well experimental results can be replicated using fitting parameters that have a physical interpretation. Models derived from physical equations have a higher likelihood of succeeding under new, untested scenarios and are useful for developing a deeper understanding of a system's behavior. Robustness means the models are stable and

produce sensible results over a wide range of conditions. Oxygen and dislocation loops are known to interact with other defects. In order to play a useful role in modern process development, precipitation models must be easy to extend and support coupling to other defect models. Last, but certainly not least, models must be usable in real-world research and development environments. They must be fast and produce results that are easy to interpret. Input parameters and constraints should be clearly understandable. Ideally, they should also be capable of integrating with existing commercial TCAD solutions and workflows.

1.1 OXYGEN IN SILICON

Nearly all silicon wafers used in IC manufacturing are prepared using the Czochralski (CZ) process. High-purity silicon is first melted in a crucible and then a precisely-oriented seed crystal is mounted on a rod and lowered into the melt. Over the course of several hours, the rod is slowly rotated and pulled out at a rate of a few tens of millimeters per hour. The silicon cools and crystallizes along the same orientation as the seed crystal, ultimately forming a large, cylindrical, single-crystal boule on the order of a couple of meters in length and 6 to 12 inches in diameter. Because the melting temperature of silicon is very high (1414 °C), the crucible is made from quartz (SiO_2), which has a higher melting point. The quartz surface introduces oxygen atoms on the order of 10^{18} cm^{-3} into the silicon.

Oxygen in silicon occurs in several forms: interstitial oxygen, as individual atoms positioned in regions between silicon lattice sites; small clusters, some of which can be electrically active (e.g., thermal double donors) or can bind to dopants; and oxygen precipitates, each of which constitute a separate SiO_2 phase within the silicon substrate and can grow to hundreds of nanometers and billions of atoms in size [1].

The presence of oxygen in silicon has long been known to have a beneficial effect on IC yield. It enhances the mechanical strength of silicon substrates [2] and, in precipitated form, can capture harmful metal impurities (a process known as *intrinsic gettering* or *internal gettering*) [3, 4]. Oxygen can also be detrimental. Metals gettered by precipitates can sharply degrade yield if located near the active regions of devices. Oxygen precipitates promote the formation of dislocation loops [5, 6, 7, 8], which act as gettering sites but also cause slip and warpage [9].

Oxygen can bind with boron, the most common p-type dopant, to form BO_2 clusters, which are strong recombination centers [10] and are a particular worry for solar cell manufacturers.

Negotiating these tradeoffs in a complex fabrication process is extremely difficult. Computational models can provide insight into how various process steps affect oxygen profiles and can be used to rapidly test optimization strategies without having to enter the clean room.

1.2 DISLOCATION LOOPS

Dislocation loops are an important class of defect that occurs in crystalline materials [11]. They represent a misalignment of the crystal lattice, as if an extra plane has been inserted or removed. An extra plane of atoms is called an *extrinsic stacking fault* and is an aggregation of silicon interstitials. The edge of this extra partial plane is the *dislocation core*.

Dislocations are responsible for the phenomenon of plasticity and the ductile properties of metals. In semiconductors, where extremely high quality crystals are required, dislocations are typically a nuisance, although applications for intentional dislocation engineering do exist [12]. Dislocations adversely impact device performance by acting as recombination centers [13], sources of scattering [14], and affect doping profiles due to preferential segregation of impurities to dislocation cores [15]. Most importantly, dislocations adversely affect the mechanical properties of silicon by causing slip and warpage [2, 9]. Dislocations act as effective gettering sites [16, 17, 18], which are desirable if they can be kept far away from the active regions of devices.

Dislocations are an increasingly important concern in nanoscale devices, where their effect on conductivity and impurity profiles is larger, and in solar cells, where they can introduce recombination centers that degrade efficiency [19]. Regions of high stress, introduced intentionally by strain engineering [20] or unintentionally as a consequence of film deposition, are known to enhance the formation and growth of dislocations. Dislocation formation is known to occur in the vicinity of growing oxygen precipitates [6, 7, 8].

Extended defects such as dislocation loops can be thought of as precipitates with interstitial silicon atoms serving as the solute. *Nucleation* – the initial formation of a precipitate from solute atoms – can be classified as being either *homogeneous* or *heterogeneous*. When only a high concentration of solute atoms (exceeding the *solubility* concentration) are required to

nucleate a precipitate, the process is said to be homogeneous. Heterogeneous nucleation requires the presence of other defects or attachment sites for precipitates to form around and begin growing. In this work, a simple heterogeneous nucleation model for dislocation loops is developed for use in the oxygen model and a more complex version involving both homogeneous and heterogeneous pathways is developed as an extension to an existing {311} defect model.

1.3 ORGANIZATION OF THIS WORK

Models of two different systems are presented in this work: oxygen precipitates in CZ silicon and dislocation loops in ion-implanted silicon. The oxygen model consists of both a full and a reduced model. The reduced model requires fewer equations to be solved by making assumptions about the precipitate size distribution. Both models were implemented as part of the same simulation code and are selectable at run-time. Both models also include a simple dislocation loop model. The model of dislocation loops in ion-implanted silicon exists only in reduced form and, unlike the simpler dislocation model within the oxygen model, assumes that dislocations form from {311} defects. It also models stress dependence based on data obtained with *ab initio* calculations, allowing applied stress to be input as a simulation condition.

Background material is discussed in Chapters 2 and 3. Chapter 2 provides a broad overview of oxygen precipitates and dislocation loops as well as defects that play an important role in their formation and evolution. Chapter 3 describes the physics of precipitation and introduces the generalized forms of the full and reduced precipitation models.

The oxygen model is described in Chapter 4 using the formalism introduced in Chapter 3. The full and reduced forms of the model are derived, compared with each other, and then the reduced model is validated against experimental data. Chapter 5 describes the model for dislocation loops in ion-implanted silicon and the *ab initio* calculations that were used to calibrate it. The model is validated against experimental data and the predicted effect of applied stress is investigated.

Lastly, concluding remarks and recommendations for future work are made in Chapter 6.

CHAPTER 2
OXYGEN PRECIPITATES AND DISLOCATION LOOPS IN SILICON

Semiconductor devices must be fabricated using extremely high quality crystals with precisely controlled defect concentrations. Perfect crystals cannot be produced, however, because defects are guaranteed to be present by the laws of thermodynamics at any non-zero temperature. This chapter provides a brief overview of the origin and behavior of two important and interdependent defects in silicon: oxygen precipitates and dislocation loops.

2.1 GROWN-IN DEFECTS

Most integrated circuits and high-efficiency solar cells are fabricated using crystalline silicon produced using the CZ growth process, described briefly in Section 1.1. Polycrystalline silicon is melted down in a crucible and a precisely oriented seed crystal is dipped into the melt. As the seed crystal is slowly rotated and pulled out, the molten silicon cools and crystallizes around the seed. The rotation speed and pulling rate determine the diameter of the resulting boule and affect the concentration of *grown-in* defects.

The silicon cools as it is withdrawn but the length of the process and the temperatures involved allow impurity diffusion, precipitation, and segregation to occur. Oxygen is the most abundant impurity, with concentrations on the order of 10^{18} cm^{-3}, followed by carbon and nitrogen, which are typically present at concentrations less than 10^{16} and 2×10^{14} cm^{-3}, respectively [21]. Concentrations can vary considerably and depend on the position in the boule and the parameters of the CZ process. This is demonstrated in Fig. 2.1, which shows the axial variation of interstitial oxygen in boules grown with different seed crystal and crucible rotation speeds. Precipitates also form during cooling and have been the subject of many experimental [22, 23, 24, 25] and theoretical studies [26, 27].

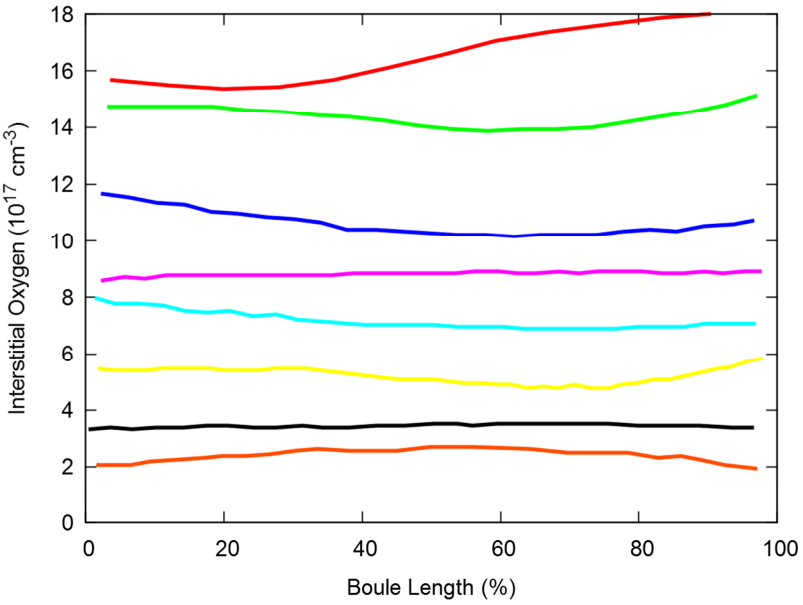

Figure 2.1. Interstitial oxygen concentration versus axial position for 8 CZ boules grown with different seed crystal and crucible rotation speeds. Reproduced from Ref. [21].

2.2 INTRINSIC DEFECTS IN SILICON

Intrinsic defects are defects comprised of the substrate material itself – in this case, silicon – as opposed to *extrinsic* defects, which involve foreign species. The most common intrinsic defects are *point defects*, so called because they are limited in size to approximately atomic dimensions. Point defects can accumulate into small clusters and grow into larger one- and two-dimensional *extended defects*, namely {311} defects and dislocation loops.

2.2.1 POINT DEFECTS

There are two types of intrinsic point defects: vacancies and interstitials. Vacancies are empty lattice sites and interstitials are silicon atoms situated between lattice sites. Both are illustrated in Fig. 2.2. In an infinite crystal, interstitials and vacancies can only be generated together (as *Frenkel pairs*) but the presence of surfaces allows point defects to be injected or absorbed independently. Point defects exist at non-zero temperatures because they increase entropy, thereby lowering the Gibbs free energy,

$$G = H - T \cdot S \tag{2.1}$$

where T is temperature. The perfect crystal has the lowest enthalpy, H, but defects increase the entropy, S.

Upon formation of a vacancy, the change in free energy of the system, ΔG_V, is

$$\Delta G_V = \Delta H_V^f - T\Delta S_V \tag{2.2}$$

where ΔH^f_V is called the formation enthalpy and ΔS_V consists of three components: ΔS^m_V, the entropy of mixing; ΔS^c_V, configuration entropy; and ΔS^f_V, the formation entropy, an entropy change due to vibrational modes, etc.

$$\Delta S_V = \Delta S_V^m + \Delta S_V^c + \Delta S_V^f \tag{2.3}$$

$$\Delta S_V^m = k_B \ln\left(\frac{C_S - C_V}{C_V}\right) \tag{2.4}$$

$$\Delta S_V^c = k_B \ln(\theta_V) \tag{2.5}$$

where k_B is Boltzmann's constant. The entropy of mixing is determined by the number of different ways vacancies can be placed into available sites, with C_V being the vacancy concentration and C_S being the concentration of possible sites. The configuration entropy depends on the number of different configurations, θ_V, of the vacancy.

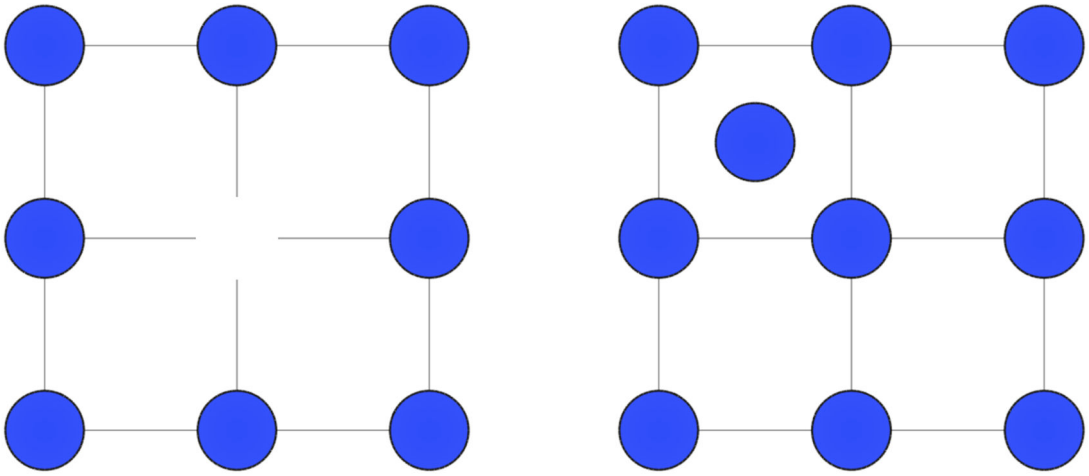

Figure 2.2. Schematic illustration of a vacancy (left) and interstitial (right) in a simple cubic lattice.

The thermal equilibrium concentration of vacancies, C_V*, is found by setting ΔG_V to zero and solving for $C_V = C_V$*.

$$\frac{C_V^*}{C_S - C_V^*} = \theta_V \exp\left(-\frac{\Delta H_V^f - T\Delta S_V^f}{k_B T}\right) \tag{2.6}$$

In general, for a defect type X,

$$\frac{C_X^*}{C_S - C_X^*} = \theta_X \exp\left(\frac{-\Delta G_X^f}{k_B T}\right) \tag{2.7}$$

where C_X represents the concentration of sites occupied by the defect, C_S-C_X is the concentration of free sites, and C_X* is the thermal equilibrium defect concentration. The formation enthalpy and entropy have been combined together into ΔG^f_X, the formation energy. In most cases, $C_S >> C_X$ (i.e., a dilute solution), allowing C_X* to be written as

$$C_X^* \cong \theta_X C_S \exp\left(\frac{-\Delta G_X^f}{k_B T}\right) \tag{2.8}$$

For a vacancy, the possible sites are simply the lattice sites, making C_S equal to C_{Si}, the silicon lattice site density.

Point defects are mobile and are characterized by their diffusivity,

$$D_X = d_X \exp\left(\frac{-\Delta H_X^m}{k_B T}\right) \tag{2.9}$$

where d_X is the diffusivity pre-factor and ΔH^m_X is the *migration barrier* (also called the *activation energy*) [28, 29].

Theoretical studies suggest several possible configurations for silicon interstitials [30]. Fig. 2.3 (b-d) shows the three lowest energy configurations: split-<110> (dumb-bell configuration), hexagonal, and tetrahedral.

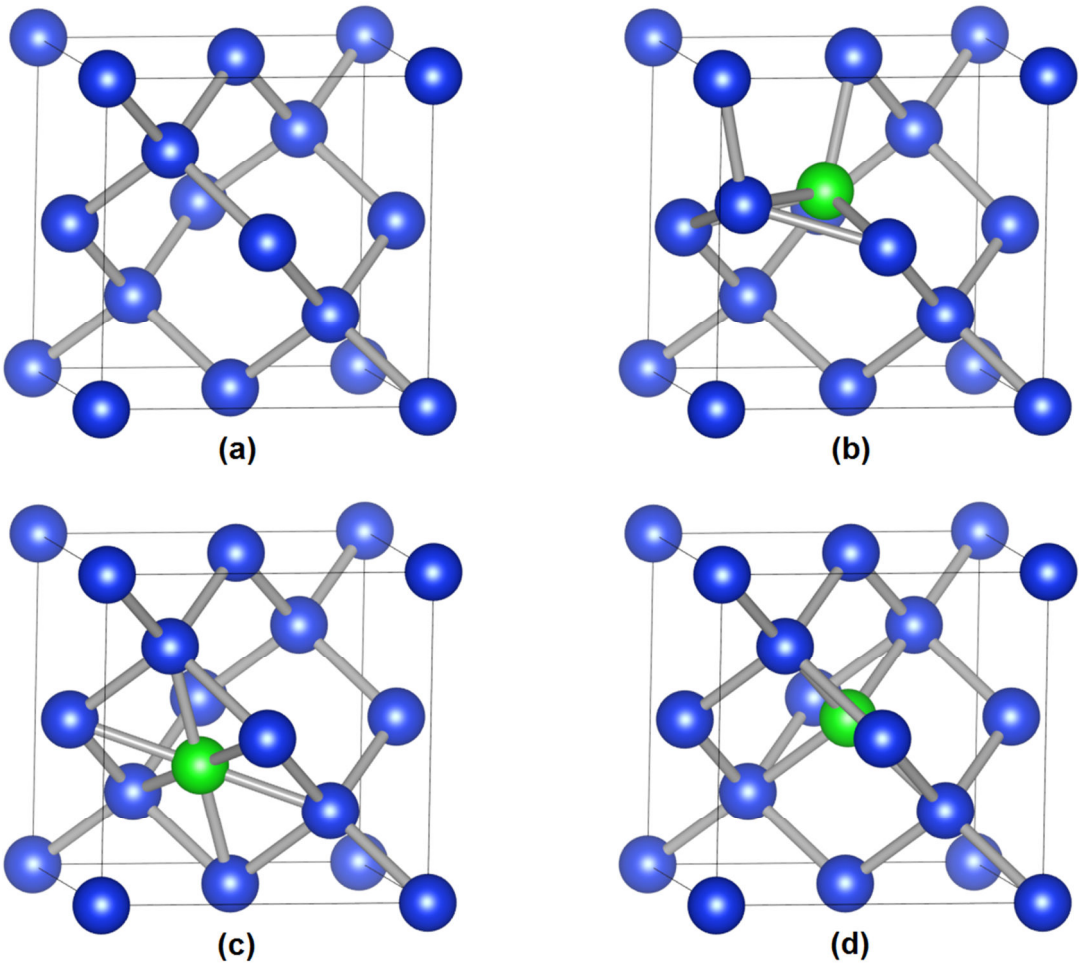

Figure 2.3. Energetically-favorable configurations of silicon interstitials (colored green): (a) perfect silicon, (b) split-<110>, (c) hexagonal, (d) tetrahedral.

2.2.2 {311} DEFECTS

Interstitial silicon atoms can accumulate together into small interstitial clusters [31] and {311} defects (sometimes called *rod-like* defects), so named because they lie in a {311} habit plane and are arranged as long, narrow chains [32]. The structure of a {311} defect is shown in Fig. 2.4.

The {311} defect forms in the presence of very high concentrations of interstitials, such as those generated by ion implantation [33, 34, 35]. Its formation and evolution can be considered a precipitation process with interstitials serving as the solute. Precipitation is discussed in more detail in Chapter 3.

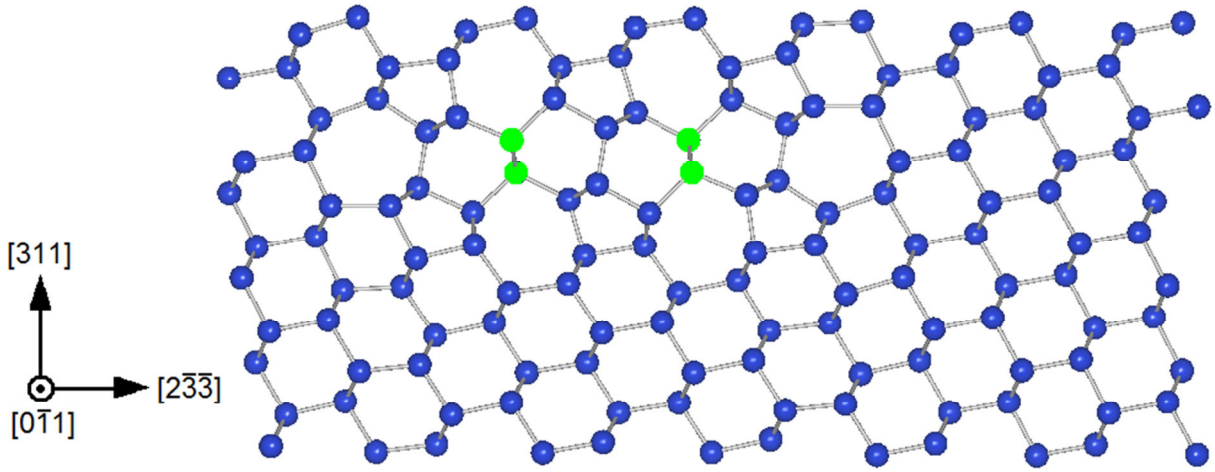

Figure 2.4. A {311} defect consisting of two interstitial chains (colored green) extending along the [0$\bar{1}$1] direction.

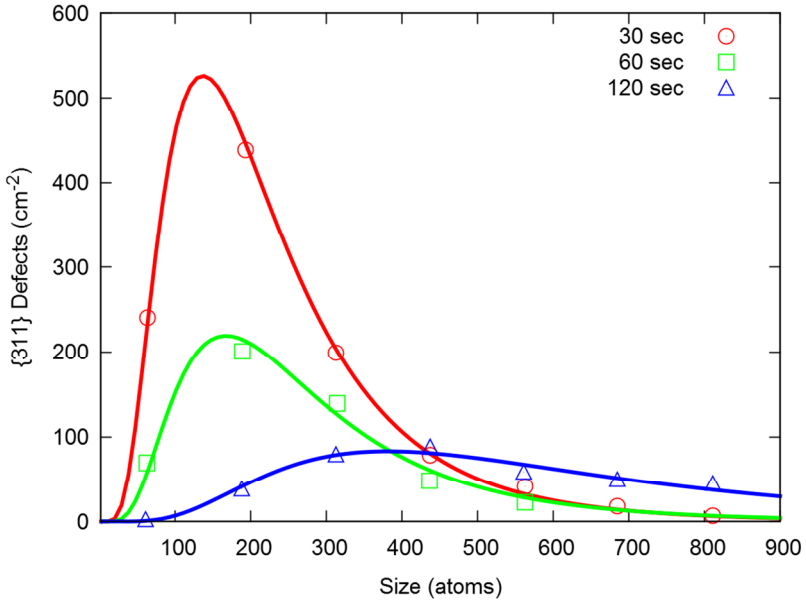

Figure 2.5. Size distributions of {311} defects fitted to Eq. (2.10) with $\sigma = 0.6$ [34, 36].

Pan and Tu found that the {311} size distribution is log-normal [34] with the form

$$f(n,t) = \frac{m_0(t)}{n\sigma\sqrt{2\pi}} \exp\left(-\frac{[\ln(n) - \mu(t)]^2}{2\sigma^2}\right) \tag{2.10}$$

where f is the density of defects comprised of n atoms at time t, m_0 is the density of all defects in the population, and μ and σ are parameters that determine the shape of the distribution (not to

be confused with the mean and standard deviation of the normal distribution function) [36]. Whereas m_0 and μ change over time as defects grow and dissolve, σ was found to remain relatively constant. Fig. 2.5 shows several size distributions observed after different annealing times fitted to Eq. (2.10) with constant σ.

2.2.3 DISLOCATION LOOPS

Dislocations are essentially a misalignment in the crystal lattice, as if caused by motion of the crystal along a cut, and have long been understood to affect the mechanical properties of materials. In metals, they are responsible for the phenomenon of plasticity. Dislocations are said to have edge, screw, or mixed character and are described by their *Burgers vector*, which describes the displacement of the defective crystal relative to its perfect form. The *dislocation core* is the location of the crystal mismatch and, in the case of edge dislocations, extends along a line perpendicular to the Burgers vector by definition. Its radius is defined arbitrarily, usually as a small multiple of the Burgers vector magnitude.

Fig. 2.6 is a schematic illustration of an edge dislocation formed by the termination of a half-plane in a simple cubic lattice. The *Burgers circuit*, outlined in blue, is an arbitrary closed path that passes through atoms in the unperturbed crystal. Following the insertion of the half-plane, the circuit no longer closes and the displacement required to do so is the Burgers vector (marked with a red arrow).

Dislocations can form due to mechanical stresses generated either internally, by precipitates or material interfaces, or externally, by film deposition or other sources of applied strain. This work is concerned exclusively with *Frank dislocation loops*, or *faulted dislocation loops*, which have an edge character and are formed from the aggregation of interstitials [11, 37]. Interstitials arrange themselves into an additional, out-of-order, partial plane called a *stacking fault*. Fig. 2.7 compares perfect and faulted silicon. When viewed along a <111> direction, crystalline silicon consists of a repeating series of three identical layers differing only by an offset along <112>. A stacking fault occurs when the layer ordering is incorrect and can be either *intrinsic*, when a layer is removed (accumulation of vacancies), or *extrinsic*, when an additional layer is inserted (due to interstitials). A dislocation exists where the partial plane terminates. Fig. 2.8 shows *ab initio* calculation results of a very small stacking fault and two edge dislocations.

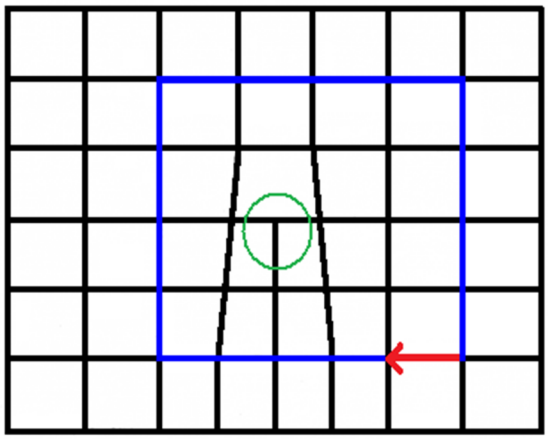

Figure 2.6. Schematic illustration of an edge dislocation showing the Burgers circuit (blue path) and the Burgers vector (red arrow) needed to complete it. The dislocation core itself is circled in green.

The kinetics of dislocation loop formation in silicon are poorly understood. It has been observed that {311} defects transform into dislocation loops through an unfaulting process [38, 39, 40] possibly involving an intermediate defect with a {111} habit plane [41].

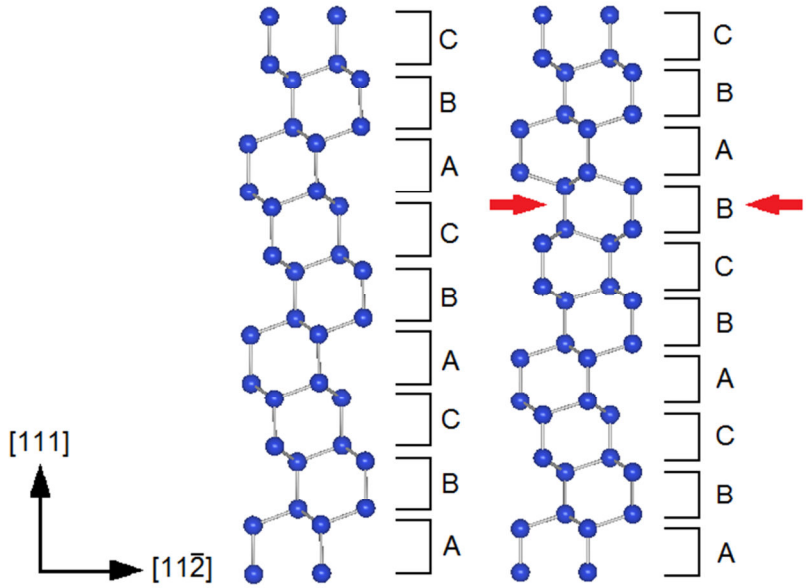

Figure 2.7. A stacking fault (marked with red arrows) caused by the insertion of an additional out-of-order "B" plane.

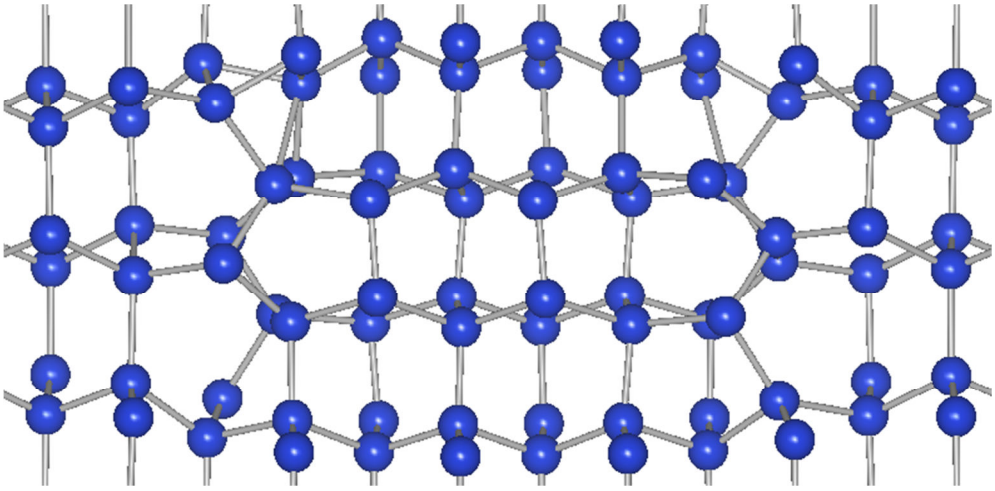

Figure 2.8. *Ab initio* calculation results of a small stacking fault and two edge dislocations.

Pan et al. found that Frank dislocation loops are approximately normally distributed in terms of radius [42] and can be fitted to

$$f(r,t)=\frac{m_0(t)}{\sigma(t)\sqrt{2\pi}}\exp\left(-\frac{[r-\mu(t)]^2}{2\sigma^2(t)}\right) \tag{2.11}$$

where f is the density of dislocations with radius r at time t, m_0 is the density of all dislocations, μ is the mean radius, and σ is the standard deviation. All the parameters evolve over time but σ remains proportional to μ. Fig. 2.9 shows several size distributions measured after different anneals fitted to Eq. (2.11).

The terms *dislocation* and *dislocation loop* are used interchangeably here and always imply the existence of a stacking fault. Other authors tend to be more precise and distinguish between stacking faults and dislocations. In the context of oxygen precipitation, *stacking fault* is the standard nomenclature but they will continue to be referred to as dislocations throughout this work.

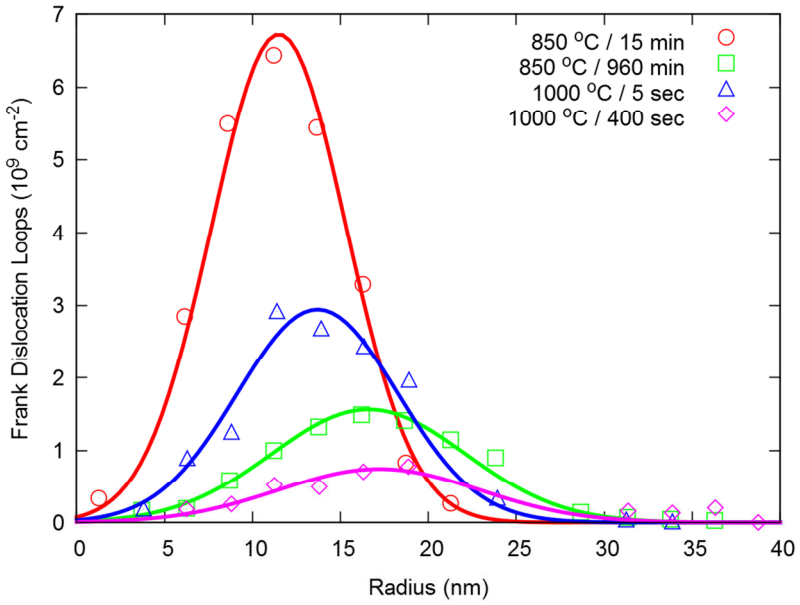

Figure 2.9. Size distributions of Frank dislocation loops fitted to Eq. (2.11) with $\sigma = \mu/3$ [42].

2.3 OXYGEN

Oxygen is an unavoidable impurity in CZ crystals and has been widely studied for several decades. In this section, a brief overview of its properties and significance is given.

2.3.1 INTERSTITIAL OXYGEN

Oxygen is most commonly present in silicon in the form of dispersed single atoms occupying interstitial sites. Characterization experiments have established the oxygen position as a bond-centered site with an Si-O bond length of approximately 1.6 angstroms (Å) and an Si-O-Si bond angle of 160° [43, 44], as depicted in Fig. 2.10. In this configuration, the valence bonds of the two silicon atoms and the oxygen atom are satisfied, making oxygen electrically inactive.

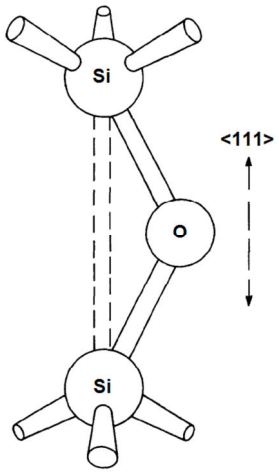

Figure 2.10. Configuration of interstitial oxygen in silicon. Reproduced from Ref. [44].

Fourier transform infrared spectroscopy (FTIR) is used to measure the concentration of interstitial oxygen, which has absorption peaks at two characteristic absorption bands in the infrared spectrum: 1107 cm^{-1} and 515 cm^{-1} [43, 45]. The relationship between the interstitial oxygen concentration, C_O, and absorption coefficient, a_0, is determined by a proportionality constant, χ, called the *calibration factor*.

$$C_O = \chi \cdot \alpha_0 \ \text{cm}^{-3} \tag{2.12}$$

Several different calibration standards exist, including two published by the American Society for Testing and Materials (ASTM) – the so-called "old" (1979) and "new" (1983) ASTM calibration factors [46, 47] – and a more recent attempt at a universal calibration standard referred to as IOC-88 (International Oxygen Coefficient 1988) [48]. Concentrations determined using different calibration factors can vary by up to a factor of 2. Therefore, it is important to understand which standard was used when interpreting experimental results so that data can be normalized to a common calibration factor. The most frequently used calibration standards are listed in Table 2.1.

Interstitial oxygen is known to inhibit the formation of dislocations through a process known as *dislocation locking* [2, 49], improving the mechanical properties of silicon wafers. This is one of the reasons that CZ silicon is often preferred over silicon grown by the floating-zone (FZ) method, which contains very little oxygen.

Table 2.1. FTIR calibration factors for determining interstitial oxygen content in silicon [50].

Calibration Standard	χ (ppma-cm)	χ (cm^{-2})
ASTM F121-79 (Old ASTM)	9.63	4.815×10^{17}
ASTM F121-83 (New ASTM)	4.90	2.45×10^{17}
ASTM F1188 (IOC-88), JEIDA 61	6.28	3.14×10^{17}
"JEIDA Coefficient (Original)"	6.06	3.03×10^{17}

2.3.2 SMALL OXYGEN CLUSTERS

At relatively low temperatures, oxygen atoms bind together to form various small cluster structures with similar binding energies that behave like single and double donors [51]. These are undesirable because they can change the resistivity of silicon beyond what is expected from precisely calibrated doping. As early as the 1950's, oxygen was suspected to play a role in the formation of donor-type defects during heat treatments between 350-500 °C [52, 53]. By the 1980's, it was established that these defects were double donors but their exact structure (and even whether or not they were really composed of oxygen atoms) remained a mystery [54]. Other oxygen-related donor defects have been discovered to form at 650-850 °C [55] and at 450 °C [51], the latter being shallow donors. A theoretical model based on *ab initio* calculations explaining the structure and formation of thermal double donors has been proposed by Lee et al. [56]

Fast-diffusing O_2 dimers with a binding energy of approximately 0.3 eV have been observed experimentally [57, 58]. *Ab initio* calculations suggest that these are electrically active and migrate by alternating between two different configurations: the so-called *square* and *staggered* structures. Both configurations have single and double positive charge states. The positively charged staggered structure creates a repulsive Coulomb barrier that slows hole trapping, resulting in a low recombination rate [10].

The efficiency of solar cells produced from CZ silicon has been observed to degrade by about one tenth when illuminated by sunlight [59, 60]. Studies have implicated BO_2 complexes as the culprit [61, 62, 63]. Theoretical calculations have led to a proposed structure consisting of an O_2 dimer trapped by substitutional boron [60] and a model for its electrical behavior that

explains the role of the boron atom in creating a strong recombination center [10]. An alternative theory holds that the degradation results from a complex of an *interstitial* boron atom and an O_2 dimer [64].

2.3.3 OXYGEN DIFFUSION

Interstitial oxygen diffusion has been characterized using numerous techniques at both high (T > 700 °C) and low (T < 400 °C) temperatures. By fitting these results, Mikkelsen produced the widely accepted expression for oxygen diffusivity [65],

$$D_O = 0.13 \cdot \exp\left(\frac{-2.53\ \text{eV}}{k_B T}\right)\ \text{cm}^2/\text{sec} \tag{2.13}$$

Oxygen diffusivity appears to be mostly insensitive to intrinsic point defects and dopants [66, 67, 68] but an enhancement effect in the presence of hydrogen [69, 70] and under electron irradiation [71] has been observed. *Ab initio* calculations reveal that oxygen migration between neighboring sites is a complex process involving coupled barriers with a saddle ridge [72].

O_2 dimers are thought to diffuse much more quickly than interstitial oxygen [73]. First-principles studies suggest that O_2 dimer diffusion occurs through a sequence of carrier-recombination-assisted reconfigurations between square and staggered structures, as shown in Fig. 2.11 [10].

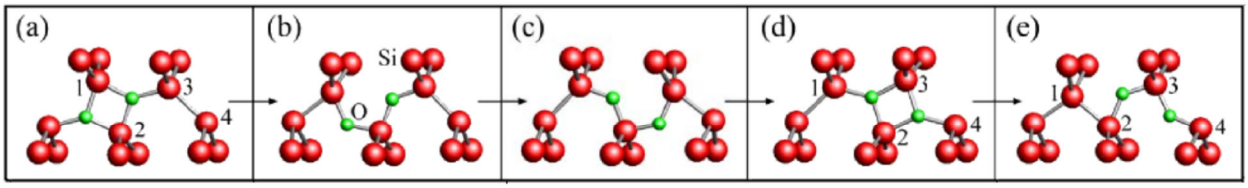

Figure 2.11. O_2 dimer (green) diffusion pathway along a {110} direction. Frames (a)-(d) show the key steps in the migration path from between silicon atoms 1 and 2 to between atoms 2 and 3. In (e), the first step in moving the dimer further to between atoms 3 and 4 is shown. Reproduced from Ref. [10].

2.3.4 OXYGEN PRECIPITATION

Perhaps the most important and widely studied oxygen-related defects are oxygen precipitates, sometimes more accurately called *oxide precipitates* because they are in fact composed of SiO_2 molecules. Precipitates form when the concentration of interstitial oxygen exceeds the solubility and their growth is a diffusion-limited process [74]. Once formed, oxygen precipitates are very stable and can only be dissolved at very high temperatures.

Nucleation of precipitates during crystal growth is a complex process that likely involves a number of different mechanisms. Numerous models have been proposed, some assuming the process is homogeneous [22, 75] and others suggesting heterogeneous nucleation involving other defects [23, 24, 76]. This work is primarily concerned with conditions that occur during thermal processing at temperatures between 600 and 1200 °C. A detailed treatment of the CZ growth process is beyond the scope of this dissertation and it is simply assumed that a grown-in initial distribution of oxygen precipitates always exists. Fig. 2.12 shows one such size distribution measured using infrared light scattering tomography [77].

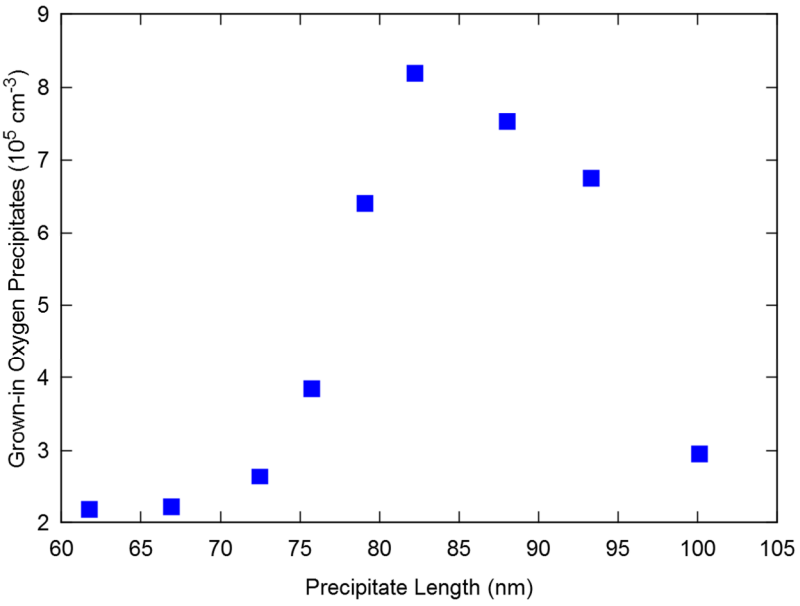

Figure 2.12. Measured distribution of the side lengths of polyhedral oxygen precipitates in an as-grown wafer [77]. The total concentration of precipitates is approximately 4.2×10^6 cm^{-3}.

The shape of oxygen precipitates depends strongly on the annealing temperature. There are three regimes. The first is at low temperatures, between 400 and 650 °C, where precipitates grow in an elongated needle-like shape comprised of high-pressure coesite SiO_2 because there is no mechanism for strain relief [78]. Fig. 2.13 shows needle-shaped precipitates observed after 650 °C annealing. In this regime, the strain energy dominates. At intermediate temperatures, from about 650 to 950 °C, precipitates take on a disk-shaped or platelet geometry that minimizes strain energy but increases the surface area [78, 79, 80], as shown in Fig. 2.14. Strain is relieved through the ejection of silicon interstitials and formation of dislocation loops, making the strain energy itself less important in determining the shape. Above 950 °C, all strain is easily relieved and precipitates take on an octahedron shape with 8 equivalent {111} faces to minimize their anisotropic surface energy [78, 80, 81, 82]. Fig. 2.15 shows octahedral precipitates imaged after high-temperature growth anneals.

Oxygen precipitation is normally studied using as-grown silicon samples subjected to one- or two-step anneals [83, 84]. The precipitation behavior of any multi-step process (e.g., CMOS) can be understood in terms of the simpler two-step sequence. The first step is conducted at an intermediate temperature between 650 and 950 °C, where the nucleation rate is largest but growth is slow, for no more than a few hours to nucleate precipitates. This is the *nucleation step*. Then, the temperature is raised above 950 °C, typically to between 1000 and 1100 °C, to allow nucleated precipitates to grow larger. This is the *growth step* and is much longer, usually between 8 and 24 hours.

The characteristic S-shaped curve that results from two-step treatments is shown in Fig. 2.16. It consists of three characteristic regions: no precipitation, partial precipitation, and full precipitation. When the concentration of oxygen exceeds the solubility level, precipitation can occur, and when the concentration is high enough, all the oxygen will eventually precipitate, resulting in a linear relationship between precipitated and initial oxygen concentrations in the *full precipitation region*. In the *partial precipitation region*, the oxygen concentration exceeds the solubility but growth kinetics and energy costs associated with the precipitate/matrix interface are the dominant factors in determining how much and how quickly precipitation will occur.

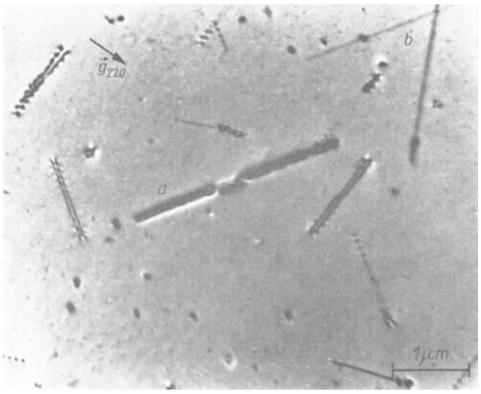

Figure 2.13. Electron microscope image of dislocation dipoles (a) and needle-shaped coesite oxygen precipitates (b) observed after a 100 hr anneal at 650 °C. Reproduced from Ref. [81].

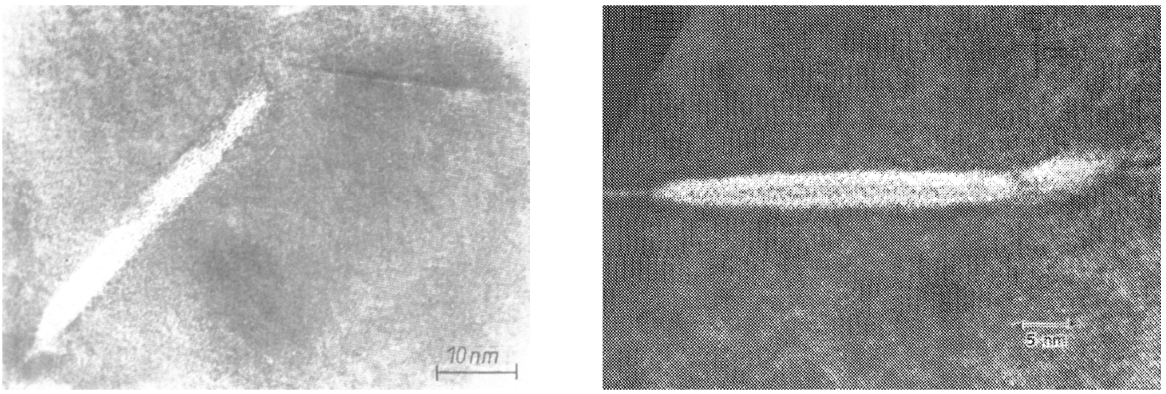

Figure 2.14. Electron microscope images of platelet oxygen precipitates formed during annealing between 750 and 900 °C. Platelets are parallel to {100} planes and are viewed edge-on. Reproduced from Refs. [81] and [78].

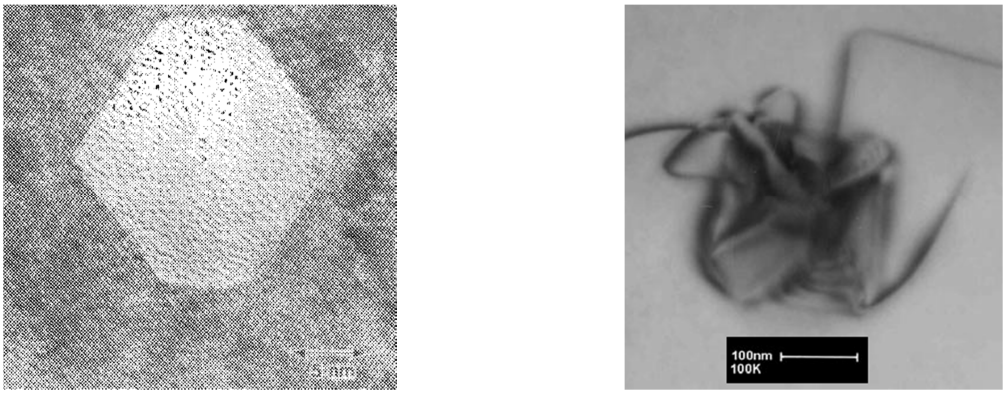

Figure 2.15. Octahedral oxygen precipitates imaged with transmission electron microscopy. Left: Precipitate after 100 hr, 750 °C nucleation and 64 hr, 1175 °C growth steps. Right: Precipitate with induced dislocations after 6 hr, 750 °C nucleation and 24 hr, 1050 °C growth steps. Reproduced from Refs. [78] and [82].

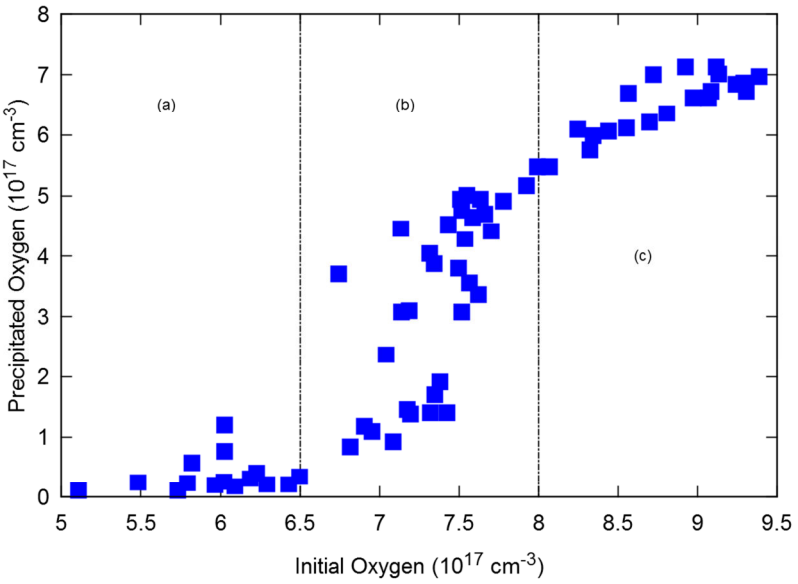

Figure 2.16. Dependence of precipitated oxygen on initial oxygen concentration in a two-step treatment. The three characteristic regions are shown: (a) no precipitation, (b) partial precipitation, and (c) full precipitation.

Numerous studies of oxygen solubility have been carried out and the best fit to the experimental data was obtained by Mikkelsen [65].

$$C_{SS} = 9\times10^{22} \exp\left(\frac{-1.52 \text{ eV}}{k_B T}\right) \text{cm}^{-3} \tag{2.14}$$

Apart from the initial oxygen concentration, precipitation is also highly dependent on other initial conditions (point defect concentrations and the grown-in precipitate distribution) and thermal history. Small oxygen precipitates formed during the CZ process can be annihilated by rapidly raising the temperature to 1200 °C or above [85]. New precipitates will still nucleate during subsequent low- and intermediate-temperature thermal steps but the total precipitation will be less because most of the grown-in precipitates will have been dissolved.

In device processing, it is desirable to have some oxygen precipitation occur in the bulk so that internal gettering of harmful impurities can occur far away from active device regions. To help achieve this, a relatively defect-free *denuded zone* is formed at the wafer surface (where devices exist) by the out-diffusion of oxygen at high temperatures (1000 to 1200 °C). Too much precipitation, however, is undesirable because the depletion of interstitial oxygen weakens the mechanical properties of the wafer [4]. Fig. 2.17 illustrates this trade-off. In terms of the S-

curve, the partial precipitation regime is generally the most desirable for device manufacturers. This requires precise control of the precipitation process – and accurate precipitation models!

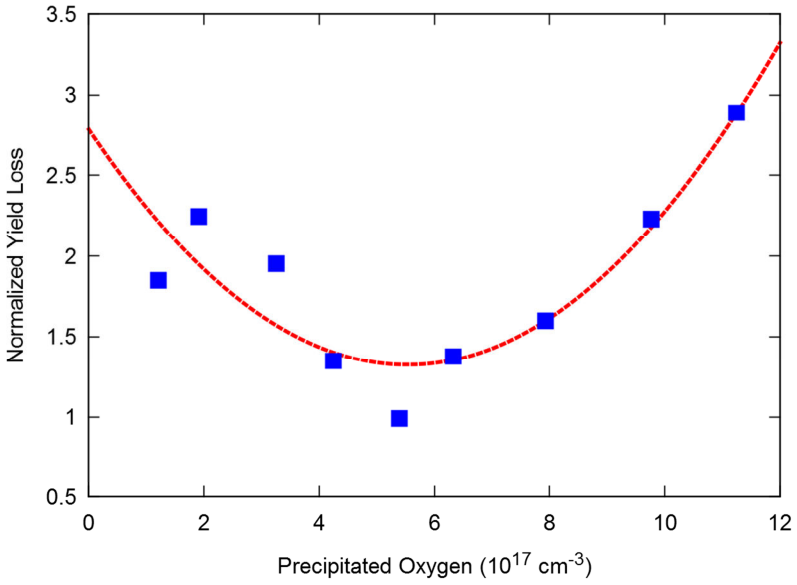

Figure 2.17. Dependence of normalized yield loss on precipitated oxygen concentration [4]. Incomplete gettering reduces yield below 5×10^{17} cm^{-3} whereas excessive dislocation generation reduces yield on the opposite end of the curve.

Several oxygen precipitation models have appeared in the literature since the 1980's. Early models modeled only growth and dissolution of existing precipitates and did not consider the nucleation of initial precipitates nor attempt to track the evolution of their size distribution over time [86, 85]. A series of sophisticated and highly influential computational models based on the kinetics of phase transformations were developed at the Vienna University of Technology beginning in the late 1980's. The first, by Schrems et al., used the Fokker-Planck equation to simulate the evolution of the precipitate size distribution [87]. This allowed the effects of thermal history to be studied and a number of one- and two-step experiments to be successfully reproduced in simulation. Subsequent versions of the model added interactions with point defects, dislocation loops, and the use of rate equations at small sizes instead of assuming an equilibrium distribution [88, 89]. A more complex model was later described by Senkader et al. [90, 21], also at the university. It included a dislocation loop model solved using the Fokker-Planck equation as well. This work seems to have influenced the Ko and Kwack, who produced a very similar model, albeit with simpler treatment of dislocation loops [91].

In the late 1990's and early 2000's, researchers at Sumitomo Metal Industries, Ltd. and later, Sumitomo Mitsubishi Silicon Corp. (now SUMCO Corp.), also developed computational models of oxygen precipitation. Kobayashi developed a model based on kinetic rate equations that included point defect interactions and used it to study nucleation during CZ crystal growth [26, 92]. Sueoka et al. developed a complex model simultaneously describing oxygen precipitates, dislocation loops, and vacancy clusters (also called crystal-originated particles) with separate Fokker-Planck equations [27]. Unlike prior models of oxygen precipitates, which treated them as spheres for simplicity, Sueoka et al. modeled their actual morphology.

Although these models appear to successfully match many experimental results, they still possess numerous shortcomings. Source code is not readily available and in some cases has been permanently lost. All were implemented with custom-written solvers and are difficult to reproduce. Models based on the Fokker-Planck equation are not easily implementable in commercial TCAD environments whereas those based on kinetic rate equations are easy to implement but computationally expensive. The procedures for generating initial conditions and fitting to experimental data are seldom accurately described in the literature and most of the models still depend on fitting parameters to match data, limiting their general applicability.

CHAPTER 3
MODELING PRECIPITATION

Precipitation is the process by which a dispersed solute species within a matrix material forms a separate phase. It is a kind of *phase transformation*. The formation of water droplets and rain from water vapor are familiar, everyday examples of precipitation. By a similar process, atoms diffusing through a crystal can also precipitate. An easily observable example in the context of semiconductor fabrication is the formation of copper precipitates on the surface of silicon, a process that occurs even at room temperature because of the very high diffusivity of copper [93, 94]. Extended defects – dislocation loops, {311} defects, voids – are also precipitates. In this chapter, the physical reasons for precipitation and common modeling approaches are discussed.

3.1 PHYSICS OF PRECIPITATION

The physics of precipitation are described by the classical theory of the kinetics of phase transformations, which was brought into its modern form by Becker and Döring [95], Volmer [96], Zeldovich [97], and Frenkel [98]. A detailed overview of the theory can be found in Ref. [99].

Precipitation occurs when the concentration of a solute species within a matrix becomes high enough that the system can lower its free energy by forming a separate solute-rich phase. Enthalpy and entropy are lowered when solute atoms are incorporated into the precipitate, providing a driving force for the process. The change in free energy, ΔG, upon adding a single solute atom to a precipitate is

$$\Delta G = \Delta G_P - k_B T \ln\left(\frac{C}{C_S}\right) \tag{3.1}$$

where ΔG_P is the formation energy consisting of enthalpy and entropy components. The entropy of mixing depends on the concentrations of solute, C, and sites that solute atoms can occupy, C_S. When $\Delta G < 0$, the precipitate will grow; when $\Delta G > 0$, it will shrink; and, if $\Delta G = 0$, the precipitation process has reached equilibrium and the precipitate will stop growing.

Solving for the solute concentration at the equilibrium point yields C_{SS}, the *solubility* (or *solid solubility*, hence the subscript).

$$C_{SS} = C_S \exp\left(\frac{\Delta G_P}{k_B T}\right) \tag{3.2}$$

At this concentration, the solute is saturated, and when C_{SS} is exceeded (*supersaturation*), a separate precipitate phase will eventually form.

In Eq. (3.1), the precipitate/matrix interface and other considerations (e.g., energy costs due to elastic deformation and other phenomena) are neglected, meaning that ΔG_P is the per-atom energy of an infinitely large precipitate. In reality, the effect of the interface and other factors must be considered and ΔG_P can be interpreted as the volume component of the formation energy. The free energy change upon forming a size n precipitate can be expressed more generally as

$$\Delta G_n = -nk_B T \ln\left(\frac{C}{C_S}\right) + \Delta G_n^f \tag{3.3}$$

where ΔG^f_n called the precipitate *formation energy*, is

$$\Delta G_n^f = n \cdot \Delta G_P + \Delta G_n^{exc} \tag{3.4}$$

The *excess energy*, ΔG_n^{exc}, includes the energy change caused by formation of the precipitate/matrix interface (called *surface energy*) and any other energy components (e.g., due to strain, point defect interactions, etc., depending on the system).

Usually, ΔG_n is expressed in terms of the solubility. Using Eq. (3.2), it can be written as

$$\Delta G_n = -nk_B T \ln\left(\frac{C}{C_{SS}}\right) + \Delta G_n^{exc} \tag{3.5}$$

This form is convenient because it explicitly shows that when $C < C_{SS}$, precipitate formation increases the energy of the system, making it thermodynamically unfavorable. When $C > C_{SS}$, the energy tends to be reduced and precipitation is likely to occur. The excess energy normally increases monotonically with size and adds an additional energy barrier that must be overcome. If only surface energy is considered, the excess energy of a spherical precipitate will be proportional to $n^{2/3}$ and for a disk-shaped precipitate (assuming solute atoms attach themselves

only at the perimeter), it will be proportional to $n^{1/2}$. Fig. 3.1 shows the change in free energy, ΔG_n, as a function of size for a spherical precipitate. The energy first increases until $n = n_{crit}$, the *critical size*, and then decreases. When $n > n_{crit}$, adding atoms decreases the free energy while removing them increases it, making growth favorable. The opposite occurs when $n < n_{crit}$ – the precipitate will dissolve.

The critical size can be determined analytically be differentiating the free energy with respect to n, setting the result to zero, and solving for $n = n_{crit}$.

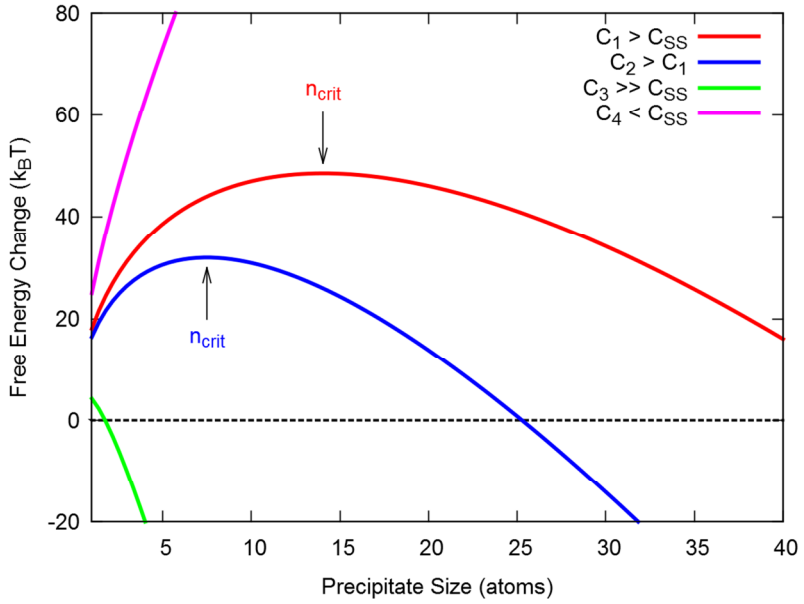

Figure 3.1. Precipitate free energy shown for different solute concentrations, C_1 through C_4, demonstrating supersaturated ($C > C_{SS}$) and undersaturated ($C < C_{SS}$) conditions. The critical size, n_{crit}, is labeled where visible.

3.2 FULL KINETIC PRECIPITATION MODEL

Precipitation can be modeled by solving a *kinetic rate equation* (KRE) for each possible precipitate size. This approach is referred to here as a *full kinetic precipitation model* (FKPM). The following common assumptions are made in its derivation:

- Precipitates grow and shrink one atom at a time as in the theory of Volmer [96].

- Nucleation occurs homogeneously, requiring only a supersaturation of the solute. The equations for heterogeneous nucleation, where pre-existing attachment sites are required, are mostly the same, differing primarily in how the smallest precipitates form.
- Precipitates are spaced sufficiently far apart that direct interactions between them can be ignored. When considering precipitate concentrations, precipitate volume is neglected and they are treated the same as point defects.
- Precipitates are immobile. This is frequently the case, such as for oxygen precipitation, but is not always true. For example, dislocations experience *glide* motion [11, 37] when subjected to stress.
- Small precipitates behave similarly to large precipitates and there are no alternative reaction pathways or phases. That is, the energies and kinetic rates of both small and very large (so-called *macroscopic*) precipitates have similar forms. In reality, the discrete effects of small clusters can be important to consider. The binding of solute atoms or small clusters (for example, BO_2 complexes) to other species may also need to be accounted for.
- All precipitates have the same morphology – in this work, disk-shaped and spherical precipitates are considered.

3.2.1 KINETIC RATE EQUATIONS

Two solute atoms cluster together to form a size two precipitate and then continue to either grow or dissolve one atom at a time. The KREs form a system of coupled differential equations that constitute the FKPM model:

$$\frac{\partial C}{\partial t} = D\nabla^2 C - \sum_{n=2}^{\infty} n \frac{\partial f_n}{\partial t} \tag{3.6}$$

$$\frac{\partial f_n}{\partial t} = R_n - R_{n+1} \qquad n = 2 \ \ldots \ \infty \tag{3.7}$$

where C is the solute concentration, D is its diffusivity, and ∇ is the spatial gradient operator. The concentration of precipitates containing n solute atoms is denoted by f_n. The net rate of

growth (or *flux*) from size n-1 to n, R_n, is expressed as a difference in growth and dissolution rates.

$$R_2 = g_1 C - d_2 f_2 \tag{3.8}$$

$$R_n = g_{n-1} f_{n-1} - d_n f_n \tag{3.9}$$

where g_n is the rate of growth from size n to n+1 and d_n is the rate of dissolution from size n to n-1.

3.2.2 GROWTH AND DISSOLUTION RATES

Growth occurs by solute atoms diffusing to the precipitate surface and incorporating themselves there. A derivation of the growth rate appears in Ref. [100]. It can be expressed as

$$g_n = D\lambda_n C \tag{3.10}$$

where λ_n is a kinetic factor that depends on the geometry and interface reaction rate. The dissolution rate can be obtained using the equilibrium condition, where growth and dissolution are balanced.

$$g_{n-1} f_{n-1}^* - d_n f_n^* = 0 \tag{3.11}$$

The equilibrium concentration of a size n precipitate, f_n^*, can be found by considering the precipitation process as a chemical reaction,

$$n \cdot X \Leftrightarrow Y_n \tag{3.12}$$

where X denotes an atom of the solute species and Y_n is a size n precipitate. In equilibrium, the free energies of the product and the reactants are equal and the reaction is halted.

$$n\Delta G_X + nk_B T \ln\left(\frac{C}{C_S}\right) = \Delta G_n^f + k_B T \ln\left(\frac{f_n^*}{C_S}\right) \tag{3.13}$$

It is important to note that all energies discussed throughout this dissertation are relative quantities, hence the purpose of the Δ symbol. The ground state relative to which energies are expressed is a perfect crystal matrix (i.e., no point defects) with solute atoms present. When oxygen precipitates are considered, the ground state includes interstitial oxygen atoms. For dislocation loops, where interstitial silicon atoms are the solute, the ground state is perfect

silicon. This makes $\Delta G_X = 0$ by definition. The equilibrium distribution can easily be solved for and is a function of the precipitate free energy of Eq. (3.3).

$$f_n^* = C_S \exp\left(\frac{-\Delta G_n}{k_B T}\right) \tag{3.14}$$

By substituting Eq. (3.14) into Eq. (3.11), the final expression for the dissolution rate is obtained.

$$d_n = \frac{g_{n-1} f_{n-1}^*}{f_n^*} = g_{n-1} \exp\left(\frac{\Delta G_n - \Delta G_{n-1}}{k_B T}\right) = \frac{g_{n-1}}{C} C_{SS} \exp\left(\frac{\Delta G_n^{exc} - \Delta G_{n-1}^{exc}}{k_B T}\right) \tag{3.15}$$

3.2.3 THE SELECTED POINTS METHOD

Precipitates can grow to billions of atoms in size or larger, making Eqs. (3.6) and (3.7) impossible to solve directly. Instead, two methods are commonly used: solution of the KREs at selected sample points (referred to here as the *selected points* method, after Kobayashi [26]) and solution of a discretized Fokker-Planck equation.

In the selected points method, KREs are solved only at a limited number of sample points. Because the adjacent sizes for a given sample point are no longer explicitly solved, the KREs are reformulated to use adjacent sample points instead.

$$\frac{\partial f_i}{\partial t} = R_i - R_{i+1} \qquad i = 2 \ \ldots \ i_{max} \tag{3.16}$$

In this notation, f_i represents the concentration of precipitates of size n_i, which is the size at the ith sample point. The maximum size, n_{max}, occurs at index i_{max} and depends on the requirements of the system. The flux, R_i, represents the net rate of growth of precipitates from size n_{i-1} to n_i.

$$R_i = g_{i-1}^s f_{i-1} - d_i^s f_i \tag{3.17}$$

The quantities g_i^s and d_i^s are functions that depend on the growth and dissolution rates and on sample points n_{i-1}, n_i, and n_{i+1}. A method for determining these quantities is provided by Kobayashi in Appendix B of Ref. [26]. His method is used to implement the oxygen precipitation model, described in Chapter 4.

At the smallest sizes, precipitate concentrations tend to approach their equilibrium values (although this is not true for short times or low temperatures) and the gradient of the distribution can become very steep. The discrete effects of small clusters and coupling to other defect species may also need to be considered. Therefore, a sample spacing of 1 is used from size 2 up to a size n_u (so that $n_i = i$ from 2 to n_u). Beyond size n_u, the spacing between adjacent samples can be non-uniform and is typically made to increase monotonically until the maximum size is reached. In this work, the following scheme is used:

$$n_i = \begin{cases} i & 2 \le i \le n_u \\ n_u + \sum_{j=n_u}^{i-1} S^{j-n_u} & n_u < i \le i_{max} \end{cases} \tag{3.18}$$

where S is the *sample discretization factor* and controls the rate at which sample spacing increases. A value of 1 results in unit spacing while larger values create fewer samples. Reasonable choices of S for an FKPM model are between 1.0 and 1.8. The value of n_u is typically chosen to be between 10 and 30 but may vary depending on the nature of the system [88, 90, 26, 91].

At the maximum precipitate size, n_{max}, there is no outgoing flux or any other boundary condition. The maximum size should be made much larger than the largest precipitates that will form during a simulation so that the concentration there remains negligible. In practice, values of 10^9 or 10^{10} atoms are adequate for oxygen models.

Examples of FKPM models using the selected points method are the arsenic and phosphorus precipitation models by Dunham [100], the oxygen model by Kobayashi [26, 92] and the heterogeneous iron precipitation model by Haarahiltunen et al. [101]. The primary disadvantage of this method is its slow performance.

3.2.4 THE FOKKER-PLANCK EQUATION

Treating n as a continuous variable allows the size distribution, f_n, to be approximated by a continuous function. This is performed by taking a first-order Taylor series expansion with respect to n of Eqs. (3.7) and (3.9), resulting in a partial differential equation of the form

$$\frac{\partial f_n}{\partial t} = -\frac{\partial I_n}{\partial n} \tag{3.19}$$

where

$$I_n = -B\frac{\partial f_n}{\partial n} + A \cdot f_n \tag{3.20}$$

$$A = g_n - d_n - \frac{\partial B}{\partial n} \tag{3.21}$$

$$B = \frac{g_n + d_n}{2} \tag{3.22}$$

As before, f_n is the precipitate size distribution function, although it is now continuous with respect to n. In general, f_n, g_n, and d_n are functions of n, space, and time. Eq. (3.19) is the *Fokker-Planck equation* (FPE), which was mentioned in Chapter 2. It is a drift-diffusion equation in size space. The first term on the right-hand side of Eq. (3.20) represents diffusion of precipitates from highly populated to less populated sizes due to random size fluctuations. The second term is the drift term, which describes the growth of precipitates to lower the total free energy of the system.

The FPE was first used in this form by Zeldovich [97] and Frenkel [98], and was later improved upon by Goodrich [102], whose formalism is often used in FPE-based precipitation models. The FPE is widely used in precipitation modeling because of its stability and relatively efficient performance. An efficient finite difference scheme satisfying the conditions of convergence and unconditional stability was developed by Chang and Cooper [103], who applied it to the modeling of electrons in plasma. A study of logarithmic and linear discretization schemes for the FPE applied to dopant precipitation appears in Ref. [104]. The favorable convergence properties and stability of the FPE allow for longer time steps and coarser sampling of the size distribution than is possible for an equivalent model using the selected points method.

At small sizes, the FPE is not valid and KREs are often used instead. The equations are linked by making the fluxes, R_n and I_n, at the transition size (e.g., $n \leq 30$) equal [90]. An alternative approach is to assume that the small sizes quickly form a quasi-equilibrium distribution, which may require the use of fitting parameters [87, 27, 105].

The FPE has been widely employed to model the precipitation of oxygen [87, 88, 89, 90, 91, 27] and other species [104, 105]. Despite the FPE's considerable advantages, a more

computationally efficient approach requiring fewer sample points in the size space remains highly desirable.

3.3 REDUCED KINETIC PRECIPITATION MODEL

Although the FKPM approach accurately models the evolution of the precipitate size distribution, the large number of equations or sample points that must be used (dozens to hundreds in typical cases) can become prohibitively time consuming, particularly when considering spatially inhomogeneous systems with fine meshing. An alternative approach is the *reduced kinetic precipitation model* (RKPM), which considers only the *moments* of the distribution [106, 36, 107, 108]. The ith moment is defined as

$$m_i = \sum_{n=k}^{\infty} n^i f_n \tag{3.23}$$

Precipitates below size k are solved using KREs as in the FKPM model. The zeroth moment, m_0, is the concentration of all precipitates of size k or larger; the first moment, m_1, is the concentration of all solute atoms held in those precipitates; and the second moment, m_2, contains information about the breadth of the distribution. Using Eq. (3.23), Eqs. (3.6) and (3.7) can be rewritten as

$$\frac{\partial C}{\partial t} = D\nabla^2 C - \frac{\partial m_1}{\partial t} - \sum_{n=2}^{k-1} n \frac{\partial f_n}{\partial t} \tag{3.24}$$

$$\frac{\partial m_0}{\partial t} = R_k \tag{3.25}$$

$$\frac{\partial m_1}{\partial t} = k \cdot R_k + d_k f_k + D \cdot m_0 \cdot \left(C \cdot \gamma_2 - \gamma_3 \right) \tag{3.26}$$

$$\frac{\partial m_2}{\partial t} = k^2 \cdot R_k + D \cdot m_0 \cdot \left(C \cdot \gamma_4 - \gamma_5 \right) \tag{3.27}$$

where

$$\gamma_2 = \frac{1}{D \cdot C \cdot m_0} \sum_{n=k}^{\infty} g_n f_n \tag{3.28}$$

$$\gamma_3 = \frac{1}{D \cdot m_0} \sum_{n=k}^{\infty} d_n f_n \tag{3.29}$$

$$\gamma_4 = \frac{1}{D \cdot C \cdot m_0} \sum_{n=k}^{\infty} (2n+1) \cdot g_n f_n \tag{3.30}$$

$$\gamma_5 = \frac{1}{D \cdot m_0} \sum_{n=k}^{\infty} (2n+1) \cdot d_{n+1} f_{n+1} \tag{3.31}$$

The second term in the right-hand side of Eq. (3.26) is required to correct for duplication between R_k and γ_3.

Ultimately, f_k and larger precipitates must be eliminated from the equations so that the entire state of the system is expressed only in terms of the moments, m_i, and precipitate concentrations, f_n, for $n < k$. It is impossible to recover the full distribution from a finite set of moments, however, and therefore some closure assumptions are required in order to approximate γ_i and the flux from size k-1 to k (which is required to compute R_k). The exact values of these quantities and the moments themselves can easily be extracted from FKPM simulations, allowing their dependence on other, known quantities in the system to be studied and models of their behavior to be constructed.

3.3.1 THE DELTA FUNCTION APPROXIMATION

The *delta function approximation* (DFA) models the precipitate size distribution as a Dirac delta function at the average size,

$$f_n = m_0 \cdot \delta\left(n - n_{avg}\right) \tag{3.32}$$

where

$$n_{avg} = \frac{m_1}{m_0} \tag{3.33}$$

Only the lowest two moments are used: m_0 and m_1. The physical justification for the DFA is that precipitate size distributions tend to be relatively sharply peaked at large sizes, as demonstrated in Fig. 3.2. The DFA also assumes that the marginal changes in the growth and

dissolution rates at large sizes are small, making it possible to treat precipitates larger than size k as if they were all of size n_{avg}, the average size.

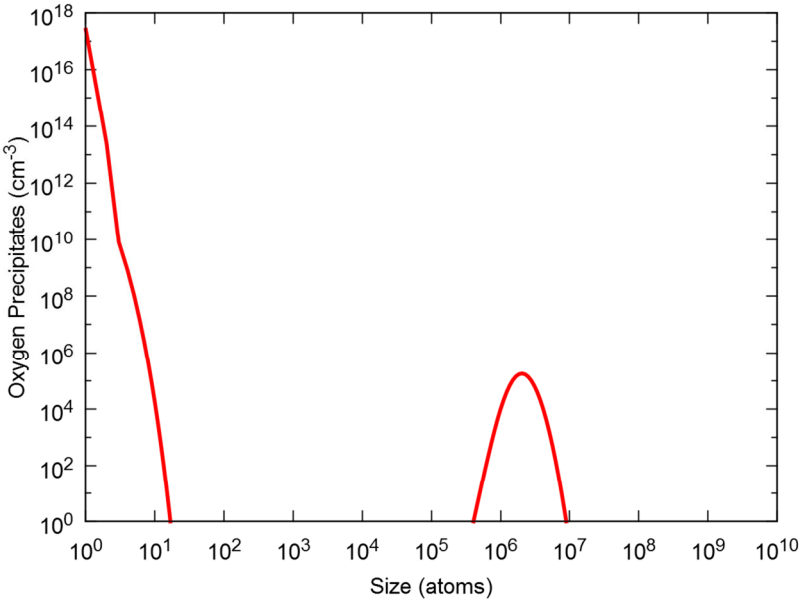

Figure 3.2. Typical oxygen precipitate size distribution showing sharp peak at large sizes.

Using Eq. (3.32), γ_2 and γ_3 can be written directly.

$$\gamma_2 = \frac{g_{n_{avg}}}{D \cdot C} \tag{3.34}$$

$$\gamma_3 = \frac{d_{n_{avg}}}{D} \tag{3.35}$$

This ability to express the RKPM model in terms of the same kinetic expressions and physical parameters as the FKPM model is one of the DFA's most appealing features.

Unfortunately, the DFA provides no help in handling the boundary condition between sizes k-1 and k. One way to resolve this problem is to devise an estimator for f_k, which cannot be solved explicitly because it is included in the definition of the moments. A second option is to model the flux from size k-1 to k (which is really what f_k is ultimately needed for) directly. There is no general procedure for either of these approaches, making the k-1/k boundary condition the most challenging aspect of constructing DFA-based RKPM models.

The applicability of the DFA to a particular precipitation problem can easily be tested by replacing the estimated k-1/k boundary condition with its actual counterpart as solved by the FKPM model. If the DFA is indeed suitable, the result will closely match the FKPM model and the accuracy of the RKPM model will depend primarily on the quality of the boundary condition approximation.

3.3.2 ANALYTICAL APPROXIMATION OF THE SIZE DISTRIBUTION

Even when the size distribution has been experimentally characterized and an expression for f_n is known, as is the case for dislocation loops and {311} defects [34, 42], closed-form expressions for the infinite summations of γ_i will virtually never exist and it may be impossible to express f_n analytically in terms of the moments. However, if it *can* be written analytically, it may be possible to perform a numerical calculation of γ_i using f_n directly and then construct an empirical function to fit the results. Failing that, approximating the distribution function with an analytical expression may simplify matters.

For example, consider the following distribution:

$$f_n = m_0 z_0 z_1^n \tag{3.36}$$

Using Eq. (3.23), a system of equations can be written and solved to determine z_0 and z_1:

$$\begin{aligned} 1 &= \sum_{n=k}^{\infty} z_0 z_1^n \\ n_{avg} &= \sum_{n=k}^{\infty} n z_0 z_1^n \end{aligned} \tag{3.37}$$

The solution when $k = 2$ is simply*

$$\begin{aligned} z_0 &= \frac{1 - z_1}{z_1^2} \\ z_1 &= \frac{n_{avg} - 2}{n_{avg} - 1} \end{aligned} \tag{3.38}$$

---

* The following identities are needed: $\sum_{n=0}^{\infty} z^n = \frac{1}{1-z}$ and $\sum_{n=1}^{\infty} n z^n = \frac{z}{(1-z)^2}$, if and only if $|z| < 1$.

The solution becomes more difficult to obtain for larger values of k. Subject to other constraints, namely on the form of the dissolution rate, d_n, Eq. (3.36) can be used to obtain parameterized forms of γ_i that can then be fitted to numerical calculations of the full summations. This approach is described in greater detail in Ref. [36].

3.3.3 ENERGY-MINIMIZING SIZE DISTRIBUTION

A size distribution that minimizes free energy, ΔG_n, is an option in situations where there is no knowledge of the actual size distribution. The total free energy of all precipitates in the system is minimized subject to Eq. (3.23) as a constraint in terms of m_0, m_1, and m_2. The resulting distribution has the form:

$$f_n = z_0 \exp\left(\frac{-\Delta G_n^{exc}}{k_B T} + z_1 n + z_2 n^2 \right) \tag{3.39}$$

where z_0, z_1, and z_2 are determined by solving the following system:

$$\begin{aligned} n_{avg} &= \frac{\sum_{n=k}^{\infty} n \cdot \exp\left(-\Delta G_n^{exc} + z_1 n + z_2 n^2\right)}{\sum_{n=k}^{\infty} \exp\left(-\Delta G_n^{exc} + z_1 n + z_2 n^2\right)} \\ \frac{m_2}{m_0} &= \frac{\sum_{n=k}^{\infty} n^2 \cdot \exp\left(-\Delta G_n^{exc} + z_1 n + z_2 n^2\right)}{\sum_{n=k}^{\infty} \exp\left(-\Delta G_n^{exc} + z_1 n + z_2 n^2\right)} \\ z_0 &= \frac{m_0}{\sum_{n=k}^{\infty} \exp\left(-\Delta G_n^{exc} + z_1 n + z_2 n^2\right)} \end{aligned} \tag{3.40}$$

Solving the system and calculating γ_i is time consuming but tables of solutions can be pre-computed and interpolated during run-time. A derivation of the energy-minimizing distribution and its application to RKPM models of arsenic and phosphorus clustering can be found in Ref. [106].

CHAPTER 4
OXYGEN MODEL

FKPM and RKPM models of oxygen precipitation in CZ silicon were developed and are described in this chapter. The FKPM model was used to construct the RKPM model. Model parameters and initial conditions were then fitted to experimental oxygen precipitation data using the RKPM model. A sensitivity analysis was also conducted to assess the robustness of the fits.

4.1 MODEL

The process of oxygen precipitation as it is modeled in this work is depicted in Fig. 4.1. Oxygen atoms diffuse from the silicon matrix to the surface of an oxygen precipitate and incorporate themselves there. The growing precipitate creates compressive strain within itself and in the surrounding matrix. Strain is relieved by consuming vacancies or ejecting interstitials to provide additional volume. The energy cost associated with point defect incorporation causes growth to slow down as the local interstitial supersaturation, C_I/C_I^*, increases. Eventually, the combination of high interstitial supersaturation and local strain causes {111}-oriented faulted dislocation loops to nucleate, which act as sinks for nearby interstitials and allow growth to continue.

To derive the oxygen model, the equations in Chapter 3 are reused with the following substitutions:

$$\begin{aligned} C &\rightarrow C_O \\ D &\rightarrow D_O \\ C_S &\rightarrow C_{Si} \end{aligned} \tag{4.1}$$

The solute species is interstitial oxygen. Its concentration is denoted by C_O and D_O is its diffusivity. The density of possible precipitation sites is assumed to be the silicon lattice site density, C_{Si}.

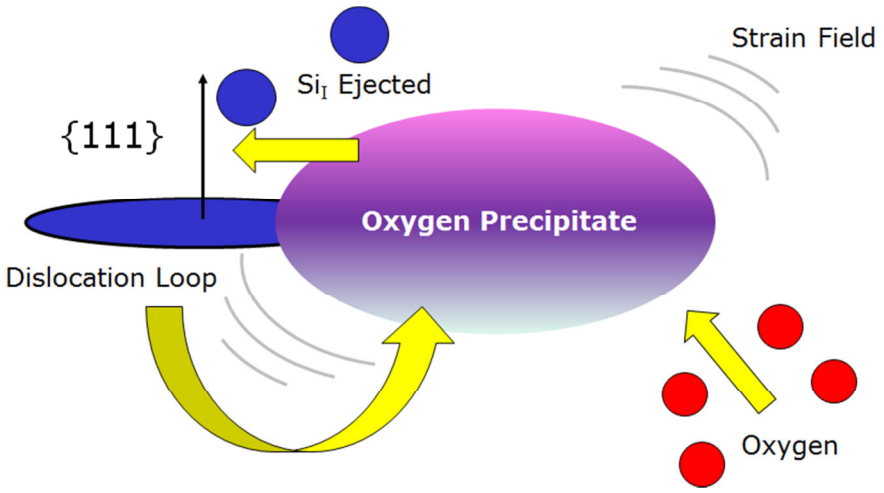

Figure 4.1. Schematic illustration of the oxygen precipitation process depicting interstitial ejection and eventual dislocation loop formation, resulting in a positive feedback loop allowing growth to proceed.

Oxygen precipitates are treated as having a spherical shape, as is commonly done, even though in reality, the precipitate morphology varies in order to minimize free energy and is influenced by point defects. The radius, r_n, of a sphere as a function of the number of oxygen atoms, n, is

$$r_n = \left(3n \frac{V_{SiO2}/2}{4\pi} \right)^{1/3} \tag{4.2}$$

where V_{SiO2} is the molecular volume of SiO_2. The small effect of residual strain on radius, described in the following section, is neglected.

From Eq. (3.10), the growth rate is

$$g_n = D_O \lambda_n^{sphere} C_O \tag{4.3}$$

The kinetic factor for a spherical precipitate, λ_n^{sphere}, is [100]

$$\lambda_n^{sphere} = \frac{4\pi \cdot r_n^2}{a + r_n} \tag{4.4}$$

where $a = D_O/k$ and k is the interface reaction rate. For a diffusion-limited reaction, a is on the order of the lattice constant. The dissolution rate can be written using Eq. (3.15).

4.1.1 STRAIN AND THE ROLE OF POINT DEFECTS

The formation of an oxygen precipitate can be expressed by the following reaction [26, 92]:

$$n \cdot O_i + p \cdot V \Leftrightarrow O_n V_{p,q} + q \cdot I + \text{strain energy} \tag{4.5}$$

where O_i denotes an interstitial oxygen atom, V represents a vacancy, and I is a silicon interstitial. The quantities p and q are the numbers of these defects involved in the reaction and $O_n V_{p,q}$ denotes an oxygen precipitate containing n oxygen atoms that has consumed p vacancies and ejected q interstitials.

Because the molecular volume of SiO_2 differs from the atomic volume of silicon by a factor of approximately 2.2, oxygen precipitates attempting to expand into the surrounding silicon matrix induce a considerable amount of strain. Additional volume to accommodate expanding precipitates can be supplied, the matrix and precipitate can become compressed and store energy in a strain field, or both. Additional volume is supplied by point defects: vacancies can be absorbed or interstitials can be ejected at a free energy cost. Elastic deformation of the precipitate and the surrounding silicon also consumes energy, which is stored in the resultant strain field. The optimum (i.e., minimum energy cost) solution is a balance between interacting with point defects and retaining some residual strain.

Fast interstitial/vacancy recombination is assumed so that

$$C_I C_V \cong C_I^* C_V^* \tag{4.6}$$

The silicon interstitial and vacancy concentrations are C_I and C_V, respectively; C_I^* and C_V^* are their thermal equilibrium values. This allows a simplified reaction to be used, as others have [109, 90, 91], involving only interstitials.

$$n \cdot O_i \Leftrightarrow O_n + m \cdot I + \text{strain energy} \tag{4.7}$$

Under the assumption of Eq. (4.6), this is equivalent to considering

$$n \cdot O_i + m \cdot V \Leftrightarrow O_n V_m + \text{strain energy} \tag{4.8}$$

The oxygen model tracks only the *net* interstitial concentration, N_I.

$$N_I = C_I - C_V \tag{4.9}$$

Using Eqs. (4.6) and (4.9), C_I and C_V can be expressed as functions of N_I, C_I^*, and C_V^*.

It is commonly assumed by most oxygen precipitation models that strain relief occurs primarily through interstitial ejection because dislocation loops are observed to form in the vicinity of growing precipitates, which is an indicator of high local interstitial supersaturations. In this work, the same assumption is made, although the added assumption of Eq. (4.6) allows one point defect species to be substituted for the other when deriving the energy and strain expressions.

4.1.2 ENERGY

To derive the components of the oxygen precipitate free energy, the vacancy-based form of the precipitation reaction, Eq. (4.8), is used. The energies of both sides of the reaction in equilibrium are written as in Eq. (3.13).

$$n \cdot \Delta G_{Oi}^{f} + nk_B T \ln\left(\frac{C_O}{C_{Si}}\right) + m \cdot \Delta G_V^{f} + mk_B T \ln\left(\frac{C_V}{C_{Si}}\right) = \Delta G_{n,m}^{f} + k_B T \ln\left(\frac{f_{n,m}^{*}}{C_{Si}}\right) \tag{4.10}$$

The formation energy change of interstitial oxygen, ΔG^{f}_{Oi}, is zero by definition because all formation energies are defined relative to perfect silicon with oxygen atoms located in interstitial positions. The vacancy formation energy is denoted by ΔG^{f}_{V} and $\Delta G^{f}_{n,m}$ is the oxygen precipitate formation energy. Unlike in the general derivation of the FKPM model, the number of incorporated vacancies, m, is used with the number of solute (oxygen) atoms, n, to track precipitates. The concentration of a size (n, m) precipitate is represented by $f_{n,m}$ and the thermal equilibrium concentration is $f_{n,m}$*. This is more general than considering only f_n but it will later be shown that tracking all possible vacancy configurations is unnecessary because only a narrow range of m values are energetically favorable.

The total free energy change, $\Delta G_{n,m}$, upon precipitate formation is

$$\Delta G_{n,m} = -nk_B T \ln\left(\frac{C_O}{C_{Si}}\right) - mk_B T \ln\left(\frac{C_V}{C_V^{*}}\right) + \Delta G_{n,m}^{f} \tag{4.11}$$

The precipitate formation energy can be separated into three components [90, 91]:

$$\Delta G_{n,m}^{f} = n \cdot \Delta G_P + \Delta G_{n,m}^{strain} + \Delta G_n^{surface} \tag{4.12}$$

$$\Delta G_n^{surface} = 4\pi \cdot r_n^2 \alpha \tag{4.13}$$

$$\Delta G_{n,m}^{strain} = 6\frac{4}{3}\pi \cdot r_n^3 \mu_{Si} \frac{e_T^2(n,m)}{1+4\mu_{Si}/\left(3K_{SiO2}\right)} \tag{4.14}$$

The atomic component of the precipitate formation energy, ΔG_P, is computed from the solubility using Eq. (3.2). Solubility has been measured experimentally [65] and can also be fitted to data from one- and two-step precipitation tests. The surface energy parameter, a, is usually reported in units of J/m^2. The *transformation* (or *linear misfit*) strain, e_T, is

$$e_T(n,m) = \left[\frac{V_{SiO2} \cdot n/2}{V_{Si} \cdot \left(n/2+m\right)}\right]^{1/3} - 1 \tag{4.15}$$

The transformation strain quantifies the mismatch relative to perfect silicon of a size (n, m) precipitate. The shear modulus of silicon is μ_{Si}, K_{SiO2} is the bulk modulus of SiO_2, and V_{Si} is the atomic volume of silicon (i.e., the inverse of C_{Si}).

Although a precipitate can in theory be formed by absorbing any number of vacancies, the energetics are unfavorable for all but a narrow range near the optimal size, m_{opt} [26]. This size is the value $m = m_{opt}$ that minimizes the energy, $\Delta G_{n,m}$. Unfortunately, the result is an equation without a closed-form solution.

To simplify the equation, the precipitate is assumed to be nearly relaxed, with only a small residual strain. Therefore, m_{opt} should be close to m_0, which is the number of vacancies needed to fully accommodate the excess precipitate volume, leaving zero residual strain. The zero strain point can be readily obtained by setting Eq. (4.15) to zero and solving for $m = m_0$.

$$m_0 = \frac{n}{2}\left(\frac{V_{SiO2}}{V_{Si}} - 1\right) \tag{4.16}$$

A linear approximation for e_T is then obtained by using a Taylor series expansion about m_0[†].

$$e_T(n,m) \cong e_T(n,m_0) + e_T'(n,m_0) \cdot \left(m - m_0\right) \tag{4.17}$$

where e_T' is the derivative of e_T with respect to m.

---

† An alternative approach is to take a second-order expansion of e_T^2 itself because this is how e_T is ultimately used in Eq. (4.14). The accuracy of the resulting expression for m_{opt} was found to be essentially equivalent to that of Eq. (4.19).

$$e_T'(n,m) = -\frac{1}{3}\cdot\left(\frac{V_{SiO2}\cdot n/2}{V_{Si}}\right)^{1/3}\cdot\left(\frac{1}{n/2+m}\right)^{4/3} \tag{4.18}$$

Fig. 4.2 is a comparison of the linear approximation of e_T with its true value. Using the linearized e_T, m_{opt} (a function of n) is found to be

$$m_{opt}(n) \cong z\cdot n \tag{4.19}$$

where z is

$$z = \frac{1}{2}\left[\frac{3}{4}k_B T\ln\left(\frac{C_V}{C_V^*}\right)\cdot\left(\frac{V_{SiO2}}{V_{Si}^2\mu_{Si}}\right)\cdot\left(1+\frac{4\mu_{Si}}{3K_{SiO2}}\right)+\frac{V_{SiO2}}{V_{Si}}-1\right] \tag{4.20}$$

Fig. 4.3 demonstrates the validity of this approach by showing the error of Eq. (4.19) relative to the true value of m_{opt}, obtained numerically along with that of precipitate energies calculated using this approximation. The error increases with increasing point defect supersaturation and temperature but the dependence of the precipitate energy on m is relatively weak so that even large errors in m_{opt} have only a modest effect on the energy.

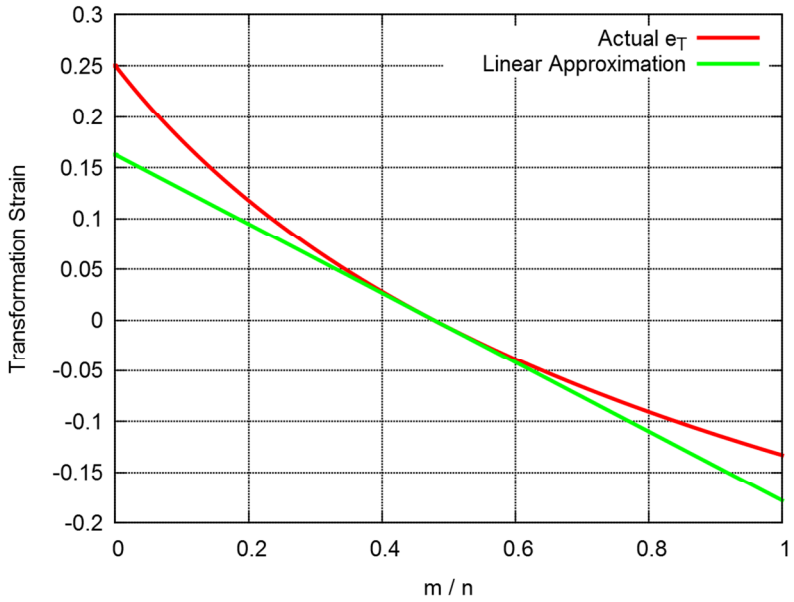

Figure 4.2. Comparison of the linear approximation of e_T, Eq. (4.17), with the true value, Eq. (4.15). The horizontal axis is the ratio of absorbed vacancies, m, to oxygen atoms, n.

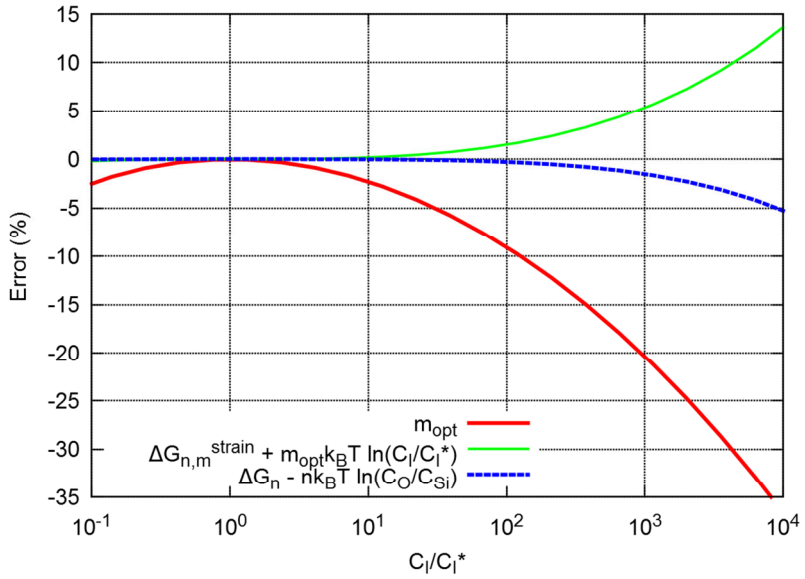

Figure 4.3. Error in m_{opt} and precipitate energy at T = 1050 °C and $n = 10^6$ when using Eq. (4.19) relative to the true value of m_{opt} computed numerically. Curves are virtually independent of n.

For all but the smallest precipitate clusters, it is assumed that only m_{opt} vacancies are incorporated because any other value will raise the formation energy and therefore be thermodynamically unfavorable.

$$f_n = \sum_m f_{n,m} \cong f_{n,m_{opt}} \tag{4.21}$$

The free energy, ΔG_n, as it appears in the model is simply $\Delta G_{n,m}$ at $m = m_{opt}$,

$$\Delta G_n = -nk_B T \ln\left(\frac{C_O}{C_{Si}}\right) + z \cdot nk_B T \ln\left(\frac{C_I}{C_I^*}\right) + \Delta G_n^f \tag{4.22}$$

where Eq. (4.6) has been used to rewrite the point defect energy in terms of interstitials and ΔG^f_n is $\Delta G^f_{n,m}$ evaluated at $m = m_{opt}$.

4.1.3 SMALL CLUSTERS

At the smallest sizes, the properties of oxygen clusters are expected to deviate from the larger, *macroscopic* precipitates described up to this point. The discrete effects of small oxygen clusters are modeled by estimating the formation energies ($\Delta G^f_{n,m}$) of likely O_nV_m clusters using a simple heuristic and *ab initio* calculation results of O_2, OV, O_2V, and V_2 (di-vacancy) clusters that were

obtained by Renyu Chen using VASP [110]. The *m*-states for a size *n* small cluster are assumed to be in relative thermal equilibrium with each other, allowing a single expression for total energy to be obtained using the following relationships:

$$f_n^* = C_{Si} \exp\left(\frac{-\Delta G_n}{k_B T}\right) = \sum_{m=0}^{n} f_{n,m}^* \tag{4.23}$$

$$f_{n,m}^* = C_{Si} \exp\left(\frac{-\Delta G_{n,m}}{k_B T}\right) \tag{4.24}$$

Only sizes 2 and 3 are implemented this way in the model. To enforce continuity from size 3 to size 4, from which point the macroscopic energy expressions are used, an offset is applied to the macroscopic energy so that it is equal to the corresponding small cluster energy when $n = 3$. The MATLAB source code (Appendix B) may be consulted for further detail on how small cluster energies are computed.

4.1.4 CRITICAL SIZE

The critical size, n_{crit}, is a useful analytic tool for understanding precipitation behavior. As discussed in Section 3.1, the critical size makes it possible to determine which precipitates will grow (those above the critical size) and which will shrink (those below), and whether precipitation is likely to happen at all given some initial size distribution. The critical size itself is not a parameter of the model that can be set independently to affect behavior; rather, it is a descriptive quantity computed from the instantaneous state of the model. Therefore, in order to accurately reflect model behavior, the critical size must be computed using the free energy expressions exactly as implemented in the model.

At the critical size, it is equally favorable for the precipitate to grow or shrink. In terms of the free energy, it is the size at which the energy does not change when *n* changes by one. Due to the discrete nature of the transitions, this condition can be expressed by either of the following equations:

$$\begin{aligned} \Delta G_{n+1} - \Delta G_n &= 0 \\ \Delta G_n - \Delta G_{n-1} &= 0 \end{aligned} \tag{4.25}$$

Solving for $n = n_{crit}$ will yield a slightly different result depending on which of the equations is used, which means there is a small (but inconsequential) amount of ambiguity in the definition of n_{crit}. To bypass this problem, n_{crit} is defined in terms of a centered discrete difference,

$$\Delta G_{n+1} - \Delta G_{n-1} = 0 \tag{4.26}$$

Because the discrete difference does not lead to a solution for n_{crit} that is easily expressed in closed form, a continuous derivative is used instead.

$$\frac{\partial \Delta G_n}{\partial n} = 0 \tag{4.27}$$

The result is

$$n_{crit} = \left[\frac{8\pi \cdot \alpha \cdot \rho^{2/3}}{3k_B T \ln\left(\frac{C_O}{C_{SS}} \left[\frac{C_V}{C_V^*} \right]^z \right) - \psi} \right]^3 \tag{4.28}$$

where ϱ and ψ are defined as follows:

$$\rho = \frac{r_n^3}{n} = \frac{3 \cdot \left(V_{SiO2} / 2 \right)}{4\pi} \tag{4.29}$$

$$\psi = \frac{\partial \Delta G_n^{strain}}{\partial n} = \frac{8\pi\rho\mu_{Si} \left[e_T(n, m_{opt}) \right]^2}{\left(1 + \frac{4\mu_{Si}}{3K_{SiO2}} \right)} \tag{4.30}$$

The error introduced by the continuous derivative in Eq. (4.27) is negligible for all but the very smallest sizes, as demonstrated by Fig. 4.4. The error below size 4 is mostly attributable to the fact that Eq. (4.28) was derived using the macroscopic form of the free energies, which are used when $n \geq 4$. Negative values of Eq. (4.28) indicate no precipitation is likely to occur, *not* that all precipitates will grow.

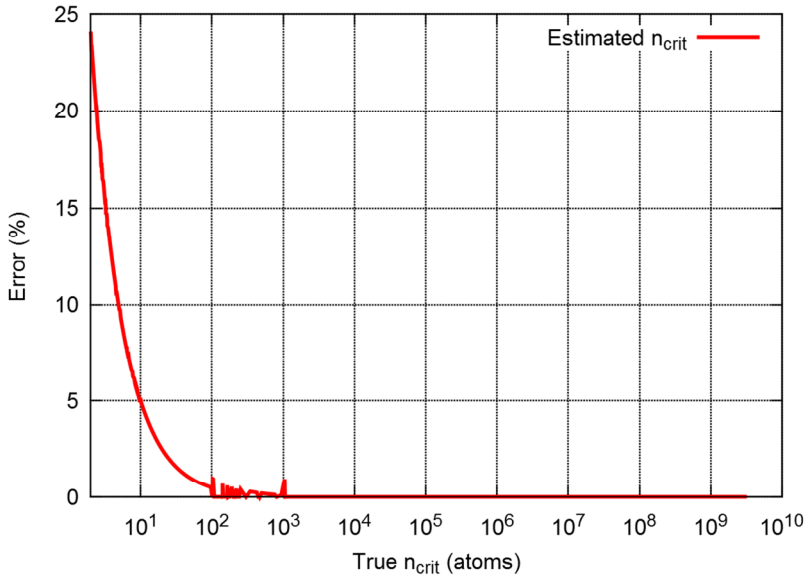

Figure 4.4. Deviation of the critical size estimated with Eq. (4.28) from its true value solved numerically.

4.1.5 POINT DEFECTS AND DISLOCATION LOOPS

Eq. (4.22) indicates that χ interstitials are ejected for each oxygen atom in a precipitate. To simplify the model implementation, 0.5 interstitials are ejected for each oxygen atom in a macroscopic precipitate, although χ is still used to compute energy. This is justified because χ was observed to remain within a small range centered about 0.5 under typical simulation conditions. The continuity equation for point defects is

$$\frac{\partial N_I}{\partial t} = D_I \nabla^2 C_I - D_V \nabla^2 C_V - R_I^{surface} - \frac{\partial m_1^{DL}}{\partial t} + 0.5\frac{\partial m_1}{\partial t} + \sum_{n=4}^{k-1} 0.5n\frac{\partial f_n}{\partial t} \tag{4.31}$$

where D_I and D_V are the interstitial and vacancy diffusivities, respectively, and $R_I^{surface}$ is the surface boundary condition.

Growing oxygen precipitates eject interstitials, leading to interstitial supersaturations that hinder further precipitation by increasing precipitate energy. When the supersaturation becomes sufficiently high, dislocation loops are formed, which then absorb interstitials into the stacking faults and lower C_I/C_I^*. By serving as sinks for ejected interstitials, dislocations provide a positive feedback mechanism that allows precipitation to proceed.

Dislocations are implemented with a simple DFA-based RKPM model. Nucleation occurs only at the critical size and once nucleated, dislocations are stable and will not dissolve. They may, however, grow or shrink. The equations for the dislocation loop moments, m_0^{DL} and m_1^{DL}, are

$$\frac{\partial m_0^{DL}}{\partial t} = g_{DL}(n_{crit}^{DL}) f_{DL}^*(n_{crit}^{DL}) \tag{4.32}$$

$$\frac{\partial m_1^{DL}}{\partial t} = n_{crit}^{DL} \cdot \frac{\partial m_0^{DL}}{\partial t} + g_{DL}(n_{avg}^{DL}) \cdot m_0^{DL} - d_{DL}(n_{avg}^{DL}) \cdot m_0^{DL} \tag{4.33}$$

The critical size, n_{crit}^{DL} is the value of n (here, the number of interstitial silicon atoms in the dislocation loop) that satisfies the equation

$$\frac{\partial \Delta G_n^{DL}}{\partial n} = 0 \tag{4.34}$$

where ΔG_n^{DL}, the free energy of a size n dislocation loop, is

$$\Delta G_n^{DL} = -nk_B T \ln\left(\frac{C_I}{C_I^*}\right) + n \cdot \Delta G_{SF} + n \cdot \Delta G_{SF}^{strain}\left(n_{avg}\right) + \Delta G_n^{self} \tag{4.35}$$

$$\Delta G_n^{self} = \frac{r_n^{DL}}{\sqrt{2}} \frac{\pi}{2} \frac{K_{DL}}{b^2} \left[\ln\left(8 \frac{r_n^{DL}}{r_{core}}\right) - 1 \right] \tag{4.36}$$

$$\Delta G_{SF}^{strain}(n) = -V_{Si} \Delta \varepsilon_{SF} \sigma_n \tag{4.37}$$

Here, ΔG_{SF} is the per-atom stacking fault formation energy obtained using VASP and ΔG_n^{self} is the dislocation elastic self-energy [37]. The energy pre-factor, K_{DL}, accounts for the anisotropy of silicon, b is the magnitude of the Burgers vector, and r_{core} is the dislocation core radius, which is set equal to b. The $1/\sqrt{2}$ factor appears in Eq. (4.36) because it is assumed that dislocations nucleated at the surface of an oxygen precipitate will take on a semi-circular geometry. The dislocation radius as a function of the number of atoms, n, is

$$r_n^{DL} = a_{Si} \cdot \left(\frac{n\sqrt{3}}{8\pi}\right)^{1/2} \tag{4.38}$$

where a_{Si} is the silicon lattice constant. Extrinsic stacking faults impart a considerable amount of compressive strain on the surrounding matrix and their formation energy is therefore highly

dependent on local stress. Tensile stress, generated at the surface of oxygen precipitates, promotes stacking fault formation. The per-atom change in energy of a stacking fault formed at the surface of a size n oxygen precipitate is $\Delta G_{SF}^{strain}(n)$. In Eq. (4.35), n_{avg} is the average oxygen precipitate size. The stacking fault induced strain, $\Delta\varepsilon_{SF}$, is the strain caused by a fully relaxed stacking fault along the direction of its {111} habit plane normalized by atomic volume. It was calculated using VASP and is discussed in more detail in Chapter 5. The stress tangential to the surface of an oxygen precipitate is [21]

$$\sigma_n = \frac{6\mu_{Si}K_{SiO2}}{3K_{SiO2}+4\mu_{Si}} \cdot e_T\left(n, m_{opt}\right) \tag{4.39}$$

A closed-form solution for n_{crit}^{DL} does not exist. To compute it, an initial guess (10^4) is made and the equation is then iterated 6 times. This computation can be written in inline form and performed very efficiently.

The equilibrium concentration of dislocations at the critical size is used as the amount nucleated. Nucleation occurs heterogeneously at the surface of oxygen precipitates. The equilibrium concentration of dislocations, f_{DL}*, is given by

$$f_{DL}^{*}(n) = 4\frac{2\pi \cdot r_n^{DL}}{a} \cdot \left(m_0 - m_0^{DL}\right) \cdot \exp\left(\frac{-\Delta G_n^{DL}}{k_B T}\right) \tag{4.40}$$

A factor of 4 is present to account for each possible {111} plane.

The growth and dissolution rates are

$$g_{DL}(n) = D_I \lambda_n^{disk} C_I \tag{4.41}$$

$$d_{DL}(n) = g_{DL}(n-1) \cdot \exp\left(\frac{\partial \Delta G_n^{DL} / \partial n}{k_B T}\right) \tag{4.42}$$

where λ_n^{disk} is a kinetic factor for the growth of a disk-shaped precipitate by attachment of atoms at the perimeter [111].

$$\lambda_n^{disk} = \frac{4\pi^2 a_{Si} \cdot \left(n\sqrt{3}/8\pi\right)^{1/2}}{\ln\left(\frac{8a_{Si}}{a_{DL}}\left(n\sqrt{3}/8\pi\right)^{1/2}\right)} \tag{4.43}$$

where a_{DL} is the capture radius for interstitials binding to the perimeter (the dislocation core), which is set to the silicon bond length.

$$a_{DL} = \frac{\sqrt{3}}{4} a_{Si} \tag{4.44}$$

4.1.6 REDUCED KINETIC PRECIPITATION MODEL

An RKPM model based on the DFA was implemented as described in Section 3.3. The FKPM model was used to calibrate the RKPM model even before the surface energy, a, was established and without a model for dislocation loops. The surface energy was initially estimated based on values reported in the literature and an artificial surface boundary condition for point defects was used to absorb excess interstitials at selected rates in order to allow precipitation to proceed. Both the FKPM and RKPM models were implemented in zero spatial dimensions (0D), simulating oxygen precipitation in the bulk of the wafer and eliminating all terms with a spatial derivative. The FKPM model was run under single- and two-step thermal cycles representative of those used in published experiments [83, 112], and the moments and γ_i were computed and extracted.

Based on the results, the value of k (the smallest size included in the moments) was chosen to be 72 because it appeared to work well with the DFA. The k-1/k boundary condition is handled using an empirical function that estimates f_k as a function of the moments.

$$f_k = \frac{m_0}{\left(\left[p_0 \cdot \left(n_{avg} - k\right)\right]^p + p_1\right) \cdot \left(n_{avg} - k\right) + 1} \tag{4.45}$$

where p_0, p_1, and p are fitting parameters. This function behaves correctly in the limiting cases:

$$\lim_{n_{avg} \to k} f_k = m_0, \quad \lim_{n_{avg} \to \infty} f_k = 0 \tag{4.46}$$

The predictor of Eq. (4.45) only contains information about the distribution *above* size k-1. A more accurate predictor would incorporate information from the low end of the distribution. However, despite investigating many different predictors based on m_0, m_1, f_{k-1}, f_{k-2}, and C_I/C_I^*, Eq. (4.45) remained the best. Fig. 4.5 plots the predictor alongside actual values of f_k obtained from a large number of FKPM simulations under different conditions. Table 4.1 lists the values of p_0, p_1, and p alongside all other model parameters.

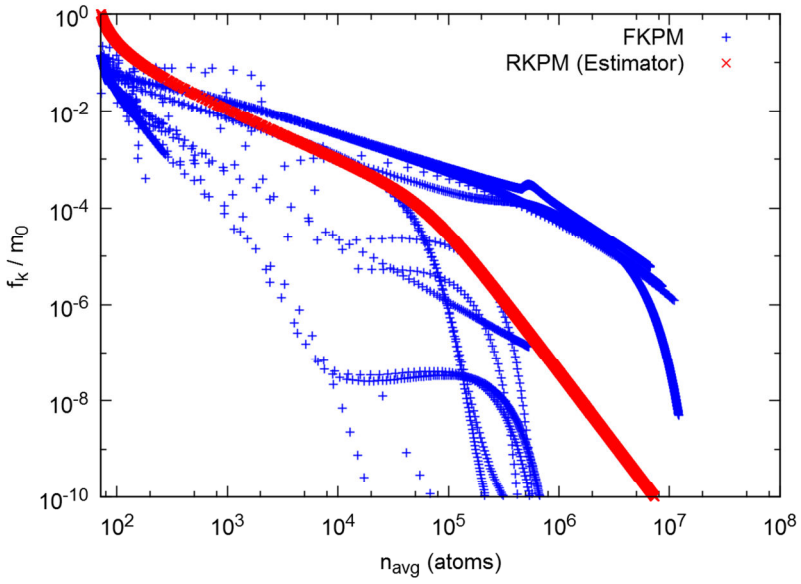

Figure 4.5. The f_k estimator, Eq. (4.45), compared to FKPM calculations of f_k for a wide range of representative conditions.

4.1.7 BOUNDARY CONDITIONS

In the finalized models, which assume a spatially homogeneous 0D system, a 0D surface boundary condition is used to estimate the diffusion of point defects to the surface and recombination there.

$$R_I^{surface} = \frac{12D_I}{L^2}\left(C_I - C_I^*\right) - \frac{12D_V}{L^2}\left(C_V - C_V^*\right) \tag{4.47}$$

where L is the wafer thickness. This is not normally reported for published experiments and it is set in the model to 400 μm, a realistic wafer thickness. Eq. (4.47) is derived in Appendix A. No such boundary condition is applied for oxygen because out-diffusion is assumed to be negligible.

4.1.8 SUMMARY

The following quantities are solved by the FKPM model:

- Interstitial oxygen concentration: C_O.
- Oxygen precipitate concentrations for sizes $n = 2 \ldots 10^9$ atoms: f_n. In the actual model implementation, the size space is sampled and the number of solved equations is on the order of 200 or less, depending on the sample discretization factor, S.
- Net silicon interstitial concentration: N_I.
- Concentration of dislocation loops (zeroth moment): m_0^{DL}.
- Concentration of interstitial silicon atoms within all dislocation loops (first moment): m_1^{DL}.

The equations that must be solved numerically are Eqs. (3.6), (3.7), (4.31), (4.32), and (4.33). The RKPM model reduces the number of precipitate equations solved by only tracking sizes $n = 2 \ldots k\text{-}1$, where $k = 72$. Less than 70 equations are required because the size space is sampled. The following additional quantities are solved:

- Concentration of oxygen precipitates (zeroth moment): m_0.
- Concentration of oxygen atoms within all oxygen precipitates (first moment): m_1.

Two additional equations are solved for the moments: Eqs. (3.25) and (3.26).

4.2 RESULTS

To validate the RKPM model, it is first compared to the FKPM model and then to experimental data. Lastly, a sensitivity analysis is performed to judge the robustness of the model. Unless otherwise noted, all simulation results were obtained using the finalized set of parameters of Table 4.1.

4.2.1 COMPARISON OF THE RKPM AND FKPM MODELS

The RKPM model is expected to differ from the FKPM model in two respects: formation of size k precipitates, which add to m_0, and growth, which affects m_1. Formation of size k

precipitates is governed by R_k in Eq. (3.25). Growth of existing precipitates is governed by γ_2 and γ_3, which in turn are obtained using the DFA. By comparing the behavior of both models under identical conditions, it is possible to assess the quality of the RKPM approximations and identify sources of mismatch.

Due to a lack of definitive information on initial precipitate size distributions, comparisons under conditions that replicate published experiments were not performed. A given set of initial m_0 and m_1 values, used to fit to experimental data, may correspond to many possible distributions in the full model. Therefore, comparisons between the models are made with no initial precipitates.

The validity of the DFA is easily tested by replacing the estimated f_k of Eq. (4.45) with the actual value computed by the FKPM model. The agreement is very good; the solute concentration, m_0, and m_1 track their full model counterparts very accurately, indicating that the DFA is a sound approach for modeling the growth of large precipitates.

Mismatch between the models is caused primarily by the inaccuracy of R_k, the flux across the k-1/k boundary. Figs. 4.6 and 4.7 demonstrate this by showing the evolution of C_O, m_0, and m_1 during a two-step process. Precipitates nucleate during the low-temperature step and then grow during the high-temperature step. The plots reveal that a large amount of dissolution occurs in the RKPM model when the temperature is ramped up, resulting in a diminished number of precipitates and therefore a retardation of the precipitation rate. Increasing the length of the nucleation step results in a larger number of precipitates and, consequently, proportionally fewer are lost to dissolution. The high initial solute concentration creates a large supersaturation, providing a strong driving force for precipitation to occur until it is no longer supersaturated.

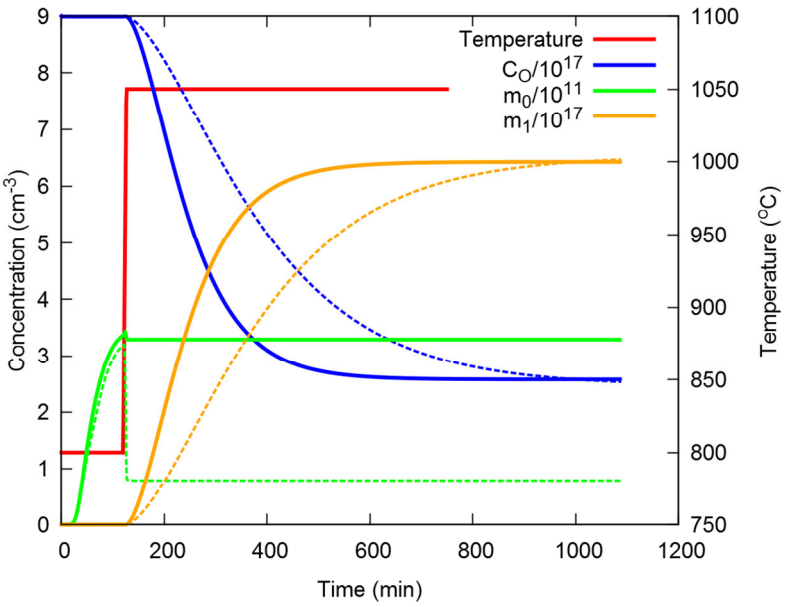

Figure 4.6. Comparison between the FKPM (solid lines) and RKPM (dashed lines) models for a two-step process: 800 °C for 2 hours, 1050 °C for 16 hours. Concentrations are scaled to enhance visibility.

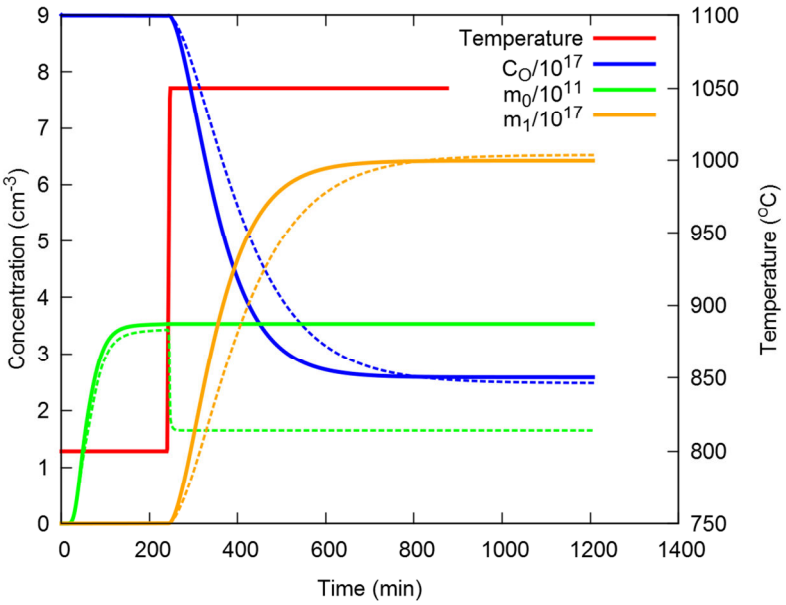

Figure 4.7. Comparison between the FKPM (solid lines) and RKPM (dashed lines) models for a two-step process: 800 °C for 4 hours, 1050 °C for 16 hours. The longer nucleation step here increases m_0 and reduces the error during the subsequent growth step.

4.2.2 COMPARISON OF THE RKPM MODEL TO EXPERIMENTS

Final calibration of the model parameters (surface energy, a, and solubility, C_{SS}) was performed using the RKPM model. They are reported in Section 4.2.3. The surface energy was allowed to have linear temperature dependence and is defined at two temperatures: 750 °C (a_{750}) and 1050 °C (a_{1050}). The temperature-dependent surface energy, $a(T)$, is

$$\alpha(T) = \frac{\alpha_{1050} - \alpha_{750}}{1050 - 750}(T - 750 - 273.15) + \alpha_{750} \tag{4.48}$$

where T is the temperature in Kelvin.

The significant effect on model behavior of small changes in the solubility concentration is demonstrated in Fig. 4.8. Although Eq. (2.14) is a widely-used estimate of the solubility, the uncertainty is high enough to justify fine-tuning this parameter by fitting to experimental data. The fitted value used in this work is given in Table 4.1 and shown in Fig. 4.9 alongside Eq. (2.14) and the experimental measurements Mikkelsen used to obtain it [65]. Mikkelsen used a non-standard calibration factor of 3.03×10^{17} cm^{-2} for the FTIR data whereas this work uses the new ASTM standard. Converting the FTIR data points in Fig. 4.9 to the new ASTM standard shifts them closer to the fitted solubility of this work but in order to keep the experimental values consistent with Mikkelsen's fit, no conversion was performed. Even without this adjustment, the solubility obtained here is well within the range of measured values.

Model validation was carried out by comparing to experiments with well-described process parameters, no oxidizing anneals (or very short oxidizing steps), and negligible carbon and nitrogen contamination. Precipitation experiments measuring changes in the interstitial oxygen concentration with FTIR spectroscopy were preferred because of the consistency and reliability of this technique. All FTIR data was normalized to the new ASTM calibration standard. Because all of the experiments were conducted on as-grown CZ silicon, initial conditions (m_0, average precipitate size, and the net interstitial concentration, none of which can be reliably measured) were treated as fitting parameters and were allowed to vary between different sets of experimental data but *not* between different experiments within the same study. The simulations replicated all thermal treatments, including temperature ramps, as they were described in the literature. The fitted initial conditions for each experiment are listed in Table 4.2.

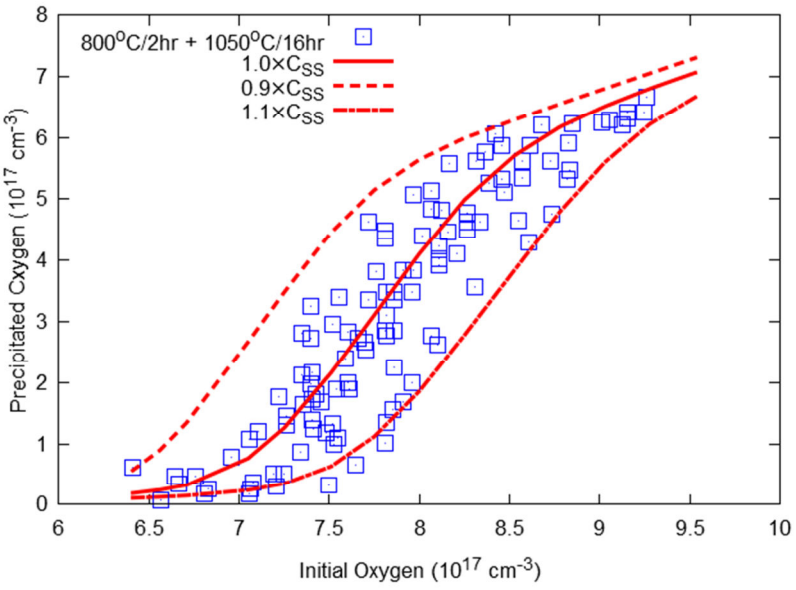

Figure 4.8. The effect of a 10% modification to the solubility (C_{SS}) on simulation results for a two-step precipitation experiment [83].

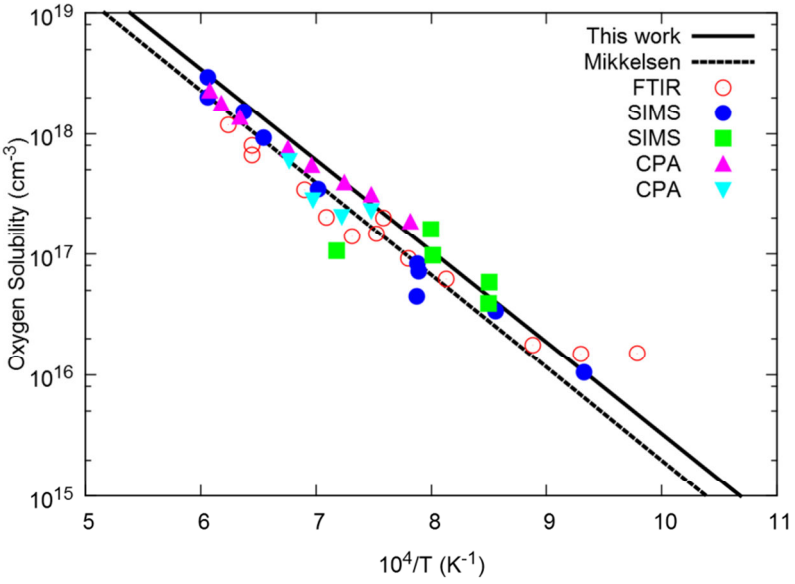

Figure 4.9. Fitted solubility of oxygen in silicon, C_{SS}, compared to data obtained with different measurement techniques and the best fit by Mikkelsen [65].

Two-step precipitation tests by Chiou and Shive [83], and Swaroop et al. [112], were used to calibrate the physical parameters. These simple experiments are designed to simulate the behavior of more complex processes, namely the CMOS process [84]. Precipitates are nucleated during the low-temperature step and then grow and ripen during the longer high-temperature step. The resultant characteristic S-shaped curve (precipitated oxygen concentration on the *y*-axis, initial interstitial oxygen concentration on the *x*-axis) allows the solubility to be easily extracted. The linear portion of the curve occurs in the regime of full precipitation and can be extrapolated to the *x*-intercept to obtain the solubility at the high temperature. Figs. 4.10 and 4.11 show good agreement with the experiments by Chiou and Shive, and Fig. 4.12 shows excellent agreement with the experiment by Swaroop et al., which has a longer nucleation step.

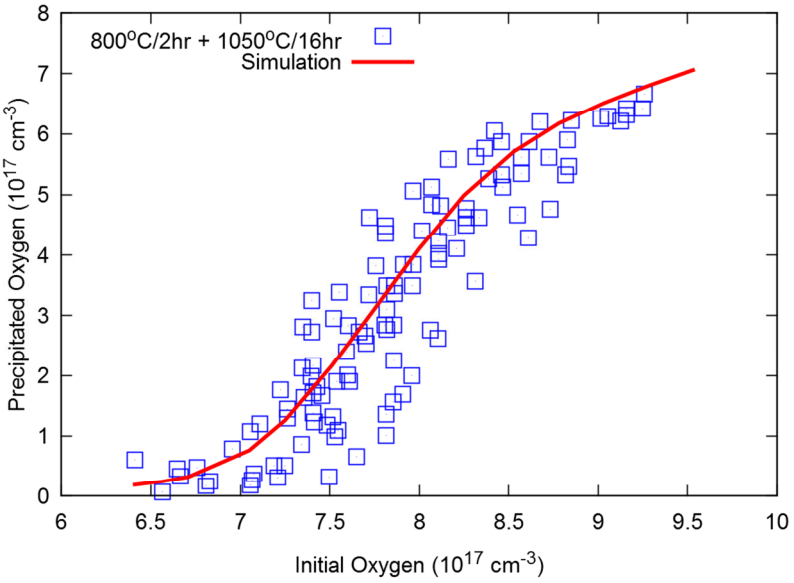

Figure 4.10. Comparison of the RKPM model to the long-duration two-step precipitation test by Chiou and Shive [83].

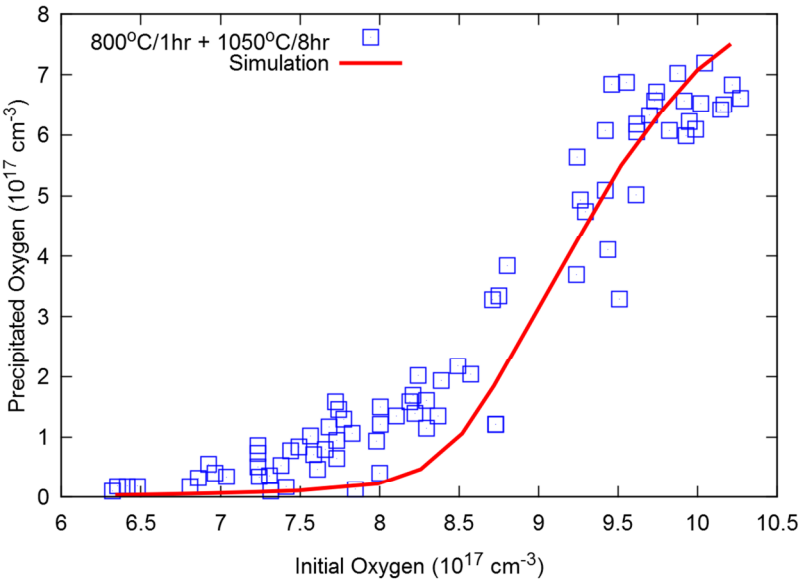

Figure 4.11. Comparison of the RKPM model to the short-duration two-step precipitation step by Chiou and Shive [83].

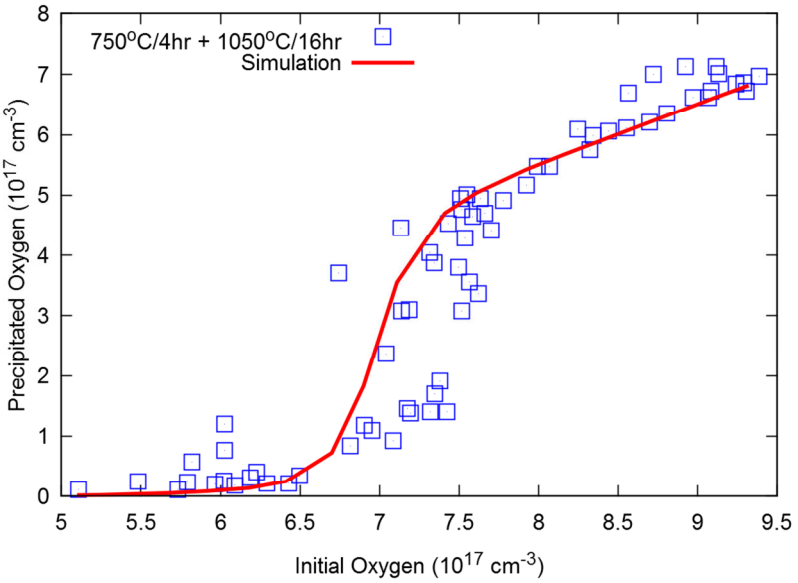

Figure 4.12. Comparison of the RKPM model to the two-step precipitation test by Swaroop et al. [112]

A large number of excess interstitials accumulate in the vicinity of growing oxygen precipitates, which slow down precipitate growth until dislocation loops form to absorb them. There is very little reliable quantitative data on dislocation loop nucleation and growth associated with precipitates, unfortunately, making it difficult to verify whether the behavior of the

dislocation model is accurate. Figs. 4.13 and 4.14 show the behavior of oxygen precipitates and dislocation loops as calculated by the model for the first 600 minutes of the two-step precipitation test by Swaroop et al. for an initial oxygen concentration of 9×10^{17} cm^{-3}, which would lie firmly in the full precipitation region of Fig. 2.16. Nucleation of oxygen precipitates causes a sharp increase in the interstitial supersaturation. Once dislocations appear, they quickly absorb the excess interstitials and grow proportionally to oxygen precipitates.

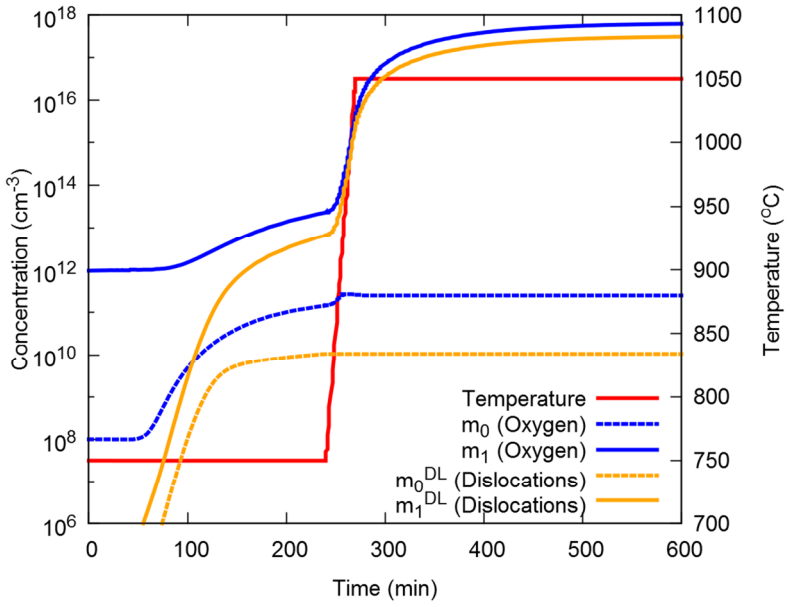

Figure 4.13. Evolution of oxygen precipitates and dislocation loops during the first 600 minutes of a two-step process (750 °C, 4 hr, and 1050 °C, 6 hr) with initial $C_O = 9\times10^{17}$ cm^{-3}. The interstitials ejected by oxygen precipitates nucleated between 50 and 150 minutes lead to dislocation loop formation.

The behavior of the dislocation loop model has its strongest effect in the partial precipitation regime where oxygen precipitation is limited by growth kinetics. In the model, dislocations primarily act as interstitial sinks that are activated when the interstitial supersaturation becomes high enough and ensure that oxygen precipitation can continue. The positive feedback they provide appears to be too strong, however, because it has been observed that once nucleated, larger interstitial supersaturations simply accelerate their growth rather than eventually slowing down oxygen precipitation, as is observed experimentally.

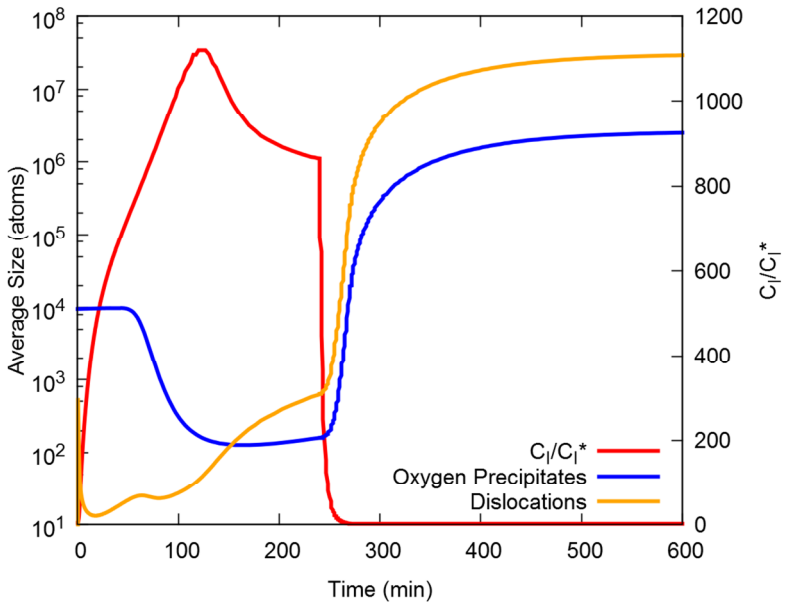

Figure 4.14. Interstitial supersaturation (C_I/C_I^*) and average defect sizes over time during the process described in Fig. 4.13.

For example, the injection of interstitials caused by oxide film growth is known to inhibit precipitation [113] but this effect would not occur with the current dislocation model. One possible reason for this is that the model nucleates dislocations only when it is favorable to do so, at which point the interstitial supersaturation is already high enough to drive their growth. Rather than nucleating at a single critical size, nucleating over a range of sizes, especially *below* the critical size, might help absorb enough interstitials earlier in the process to prevent runaway dislocation growth.

Sueoka et al. assume in their model that dislocations are nucleated during crystal growth between 1000 and 900 °C, and multiply the dislocation elastic self-energy term by a fitting parameter to ensure that this occurs [27]. Senkader et al. assume dislocations nucleate homogeneously and calculate their evolution using an FKPM model implemented with the Fokker-Planck equation [90, 21]. Ko and Kwack assume a constant dislocation loop density [91].

Experiments conducted by Kennel [114] to investigate the effects of nucleation time and temperature on precipitation behavior were also simulated. These experiments consisted of a nucleation step at either 650 or 750 °C, lasting between 0 and 32 hours, followed by a growth step at 1100 °C, from 0 to 20 hours. The simulations show good agreement with the data for 0

and 2-hour 750 °C nucleation anneals, shown in Fig. 4.15, but poorer agreement at 8 and 32 hours. Simulation results for the 650 °C nucleation treatments do not match Kennel's measurements. These mismatches are most likely caused by the nucleation rate being too low due to the assumption of spherical precipitate geometry when in fact needle- or platelet-shaped precipitates are observed below 950 °C [79, 78, 80]. Sueoka et al. report that assuming a spherical shape causes the strain energy to be overestimated, which then causes nucleation rates to be underestimated [27].

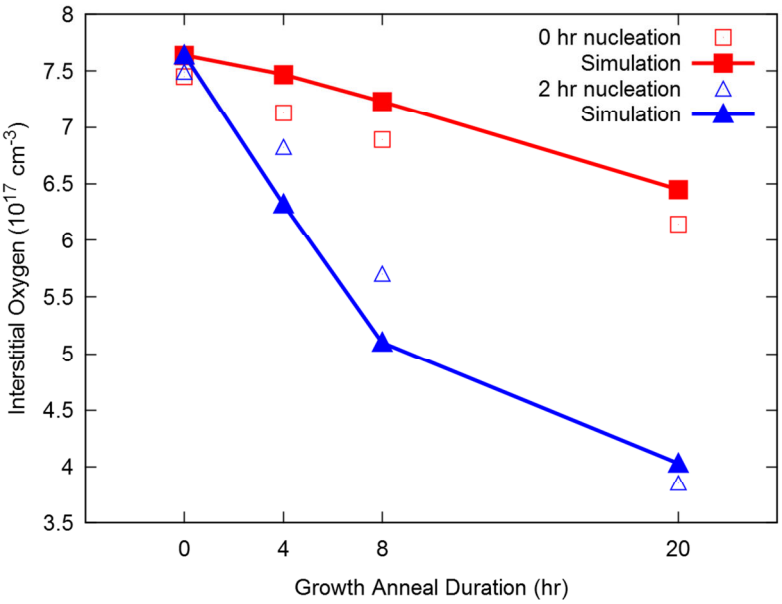

Figure 4.15. Experimental [114] and simulation results of the effect of varying the durations of 750 °C nucleation and 1100 °C growth anneals on the final interstitial oxygen concentration.

The model is able to replicate precipitate growth during a single-step anneal as observed in an experiment by Stewart et al. [115] Small-angle neutron scattering (SANS) measurements of oxygen precipitates were made at various points during a 750 °C anneal of as-grown CZ silicon. SANS measurements work by directing a beam of neutrons at a sample and measuring the amount of deflection that occurs due to elastic scattering from interactions with atomic nuclei. The number and volume of scattering centers, presumed to be oxygen precipitates, are measured and from that, the number of oxygen atoms is computed using the molecular volume of SiO_2.

Fig. 4.16 shows the loss of interstitial oxygen to precipitation during a 750 °C anneal compared with data from Ref. [115]. The results were found to be relatively insensitive to the

initial precipitate concentration and were fitted by adjusting the initial point defect concentration, N_I. The lower value of the initial data point (at 24 hours) is not possible and was probably caused by measurement uncertainty. Stewart et al. point out that precipitates become more difficult to measure at small sizes because the magnitude of the scattering cross-section is proportional to n_{avg}^2, where n_{avg} is the average number of atoms in a precipitate. The simulation predicts that the average precipitate size at the first point is more than an order of magnitude smaller than at the next sample (72 hours).

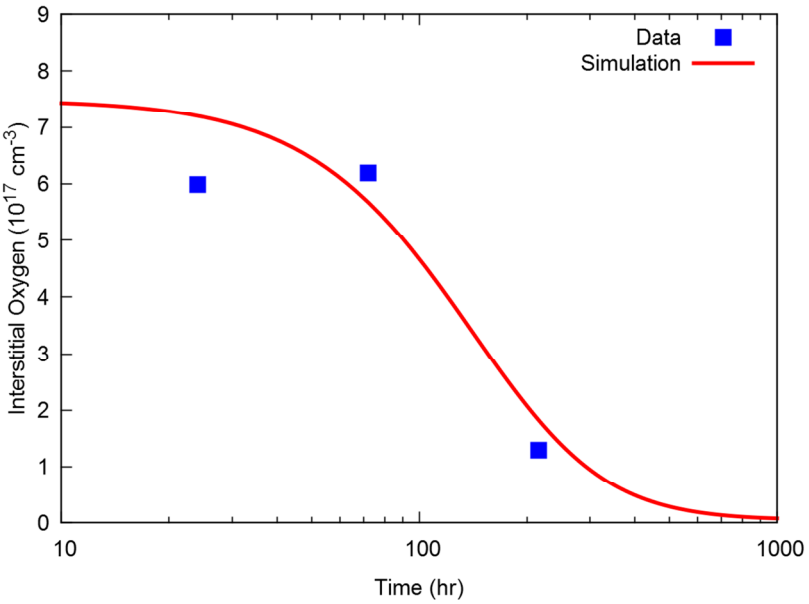

Figure 4.16. Simulation results of interstitial oxygen concentration over the course of a long-duration 750 °C anneal compared to SANS measurements [115]. The initial oxygen concentration was determined by FTIR to be 7.49×10^{17} cm^{-3}.

High-temperature (1100 °C and above) experiments by Hawkins and Lavine [85], Isomae [116], and Abe et al. [113] were also studied but could not be replicated. Whereas these experiments indicate that growth of nucleated precipitates occurs during the high temperature steps, a substantial amount of dissolution occurs in the RKPM model. This appears to be caused by the f_k estimator but may also be influenced by the increase in surface energy due to its linear dependence on temperature. Allowing the surface energy to change with temperature can be used to account for some of the effects of varying precipitate morphology but a simple linear model is difficult to justify. Ko and Kwack solved this problem by fitting the surface energy at several different temperatures [91].

Ko and Kwack claim to have replicated the single-step anneals by Abe et al. [113] but neither their paper [91] nor the doctoral dissertation by Ko [117] make any mention of whether or not experimental FTIR data was normalized to a common calibration standard. The oxygen concentrations in Fig. 3 of Ref. [91] (which is a comparison to the experiment by Chiou and Shive [83]) use the new ASTM standard but Fig. 8a (a comparison to data from Abe et al.) uses the old ASTM standard, which results in concentrations that are nearly twice as large. The data from Abe et al. can easily be fitted with the RKPM model using the old ASTM values because the initial oxygen concentration interpreted this way is extremely high: 1.65×10^{18} cm^{-3} . This makes it possible for precipitates present at the beginning of the process to grow larger at even the highest temperatures but is incorrect because the model parameters are valid only for concentrations normalized to the new ASTM calibration standard. The source code to Ko's model has been lost (B. G. Ko, personal communication, November 17, 2011), making it impossible to verify whether normalization was actually performed. If it was, the data points in the figures would have had to have been converted back to the original calibrations of their respective sources.

Senkader et al. [90] explicitly state that all data in their work was normalized to the IOC-88 standard but Fig. 8 of Ref. [90] is expressed using the old ASTM standard, which is how the data was originally measured by Hawkins and Lavine [85]. This error also appears in Senkader's doctoral dissertation [21]. If these values are used directly, the RKPM model can also reproduce the two-step experiment (excluding the effect of rapid thermal anneal steps), but this is not a valid result. Unfortunately, Senkader's simulation code has also been lost (S. Senkader, personal communication, November 28, 2011), making it impossible to verify exactly how the simulations were conducted.

4.2.3 PARAMETERS AND INITIAL CONDITIONS

The final set of physical parameters is shown in Table 4.1. Based on tests conducted with both models, it was found that sample discretization factors (S) between 1.05 and 1.2 provide acceptable accuracy. The final value chosen was 1.106954, which is the closest value to 1.1 that ensures k is 72.

Table 4.1. Parameters of the FKPM and RKPM oxygen models.

Parameter	Value	Units	Reference
D_O	0.13 exp(-2.53 eV/k_BT)	cm²/sec	[65]
a	5.0×10^{-8}	cm	
C_{SS}	1.17×10^{23} exp(-1.499 eV/k_BT)	cm^{-3}	
D_I	51.4 exp(-1.77 eV/k_BT)	cm²/sec	[118, 119]
C_I^*	2.9×10^{24} exp(-3.18 eV/k_BT)	cm^{-3}	[118, 119]
D_V	3.07 exp(-2.12 eV/k_BT)	cm²/sec	[118, 119]
C_V^*	1.4×10^{24} exp(-2.44 eV/k_BT)	cm^{-3}	[118, 119]
C_{Si}	5.0×10^{22}	cm^{-3}	[120]
a_{Si}	5.431×10^{-8}	cm	[120]
V_{Si}	2.0×10^{-23}	cm^3	[120]
μ_{Si}	64.9	GPa	[121]
V_{SiO2}	4.35×10^{-23}	cm^3	[122, 123]
K_{SiO2}	36.9	GPa	[124]
b	$a_{Si}\sqrt{3}/3$	cm	
r_{core}	b	cm	
K_{DL}	72	GPa	[125]
ΔG_{SF}	0.0152	eV	
$\Delta\varepsilon_{SF}$	0.996		
L	400	μm	
p_0	5×10^{-6}		
p_1	0.1		
p	2		
a_{750}	0.1915	J/m²	
a_{1050}	0.2565	J/m²	
k	72		
n_u	10		
n_{max}	10^9		
S	1.106954		

The best fit for a was found to increase with temperature, unlike the reported values in Ref. [87], but consistent with the model by Ko and Kwack [91]. However, unlike this work, the temperature dependence of a in the Ko and Kwack model is not linear.

Initial conditions used to fit experimental data are reported in Table 4.2.

Table 4.2. Fitted initial conditions for experimental data.

Experiment	**Initial m_0** (cm^{-3})	**Initial n_{avg}**	**Initial N_I** (cm^{-3})
Chiou and Shive [83]	9.71×10^{8}	7.80×10^{3}	-4.91×10^{12}
Swaroop et al. [112]	1.0×10^{8}	9.53×10^{3}	-3.97×10^{11}
Kennel [114]	1.0×10^{8}	4.05×10^{3}	-3.15×10^{12}
Stewart et al. [115]	1	73	-1.51×10^{14}

4.2.4 SENSITIVITY ANALYSIS

The results of Section 4.2.2 appear to be quite good but they provide no indication of the robustness of the fits. In order to understand which factors have the biggest impact on the precipitation process and to judge the quality of the model itself, an analysis of the model's sensitivity to its parameters is carried out.

Characterization experiments are usually repeated several times in order to capture natural variations that occur in initial conditions and during processing. A model exhibiting extreme sensitivity to a particular set of conditions is unlikely to be correct unless the same degree of sensitivity is confirmed by experimental observations. Unfortunately, sensitivity analyses for oxygen precipitation models do not appear to exist in the literature, although the work of Sueoka et al. indicates that models can be highly sensitive to initial point defect concentrations [27].

In order to assess the sensitivity of the model to changes in physical parameters (C_{SS}, a) and initial conditions (m_0, n_{avg}, N_I), a one-at-a-time sensitivity analysis was performed. Each parameter was varied within a physically reasonable range while keeping the others fixed at their best-fit values for a given experiment. The fitness metric, F, used to measure the quality of the fit, is a scaled sum-of-squares error.

$$F = \sum_i \left(\frac{y_i}{10^{18}} - \frac{x_i}{10^{18}} \right)^2 \tag{4.49}$$

where x_i is the experimental result at the ith sample point (i.e., precipitated oxygen concentration) and y_i is the corresponding simulation result. A perfect fit is indicated by $F = 0$.

The longer of the two Chiou and Shive experiments (800 °C for 2 hr, 1050 °C for 16 hr) [83] was selected for the sensitivity analysis because it was the primary experiment used to calibrate the physical parameters of the oxygen model. Figs. 4.17 and 4.18 show the effect of the surface energy and solubility on the fit quality. Both parameters strongly affect the accuracy of the model. There is a narrow range about which the surface energy can be varied with only a small effect on simulation results. The solubility, on the other hand, is much more important and offers very little room for modification.

The sensitivity to initial conditions is shown in Figs. 4.19-4.21. The model is most sensitive to the initial point defect concentration, which is a key factor in determining precipitate energy. It is less sensitive to the initial concentration of precipitates and their size. In all the replicated experiments, the fitted initial precipitate concentrations and average sizes are low, consistent with the findings of other theoretical [26, 92, 91, 27] and experimental studies [77]. Nevertheless, the presence of sufficiently large precipitates exceeding the critical size at the start of the process can influence simulation results. This is reflected in the sensitivity analysis, which shows insensitivity to initial precipitate conditions only to a certain point.

The initial point defect concentration was assumed to be a multiple of C_I^*-C_V^* at T = 1200 °C. Point defect concentrations in silicon can vary widely and the range examined in Fig. 4.21 is by no means exhaustive.

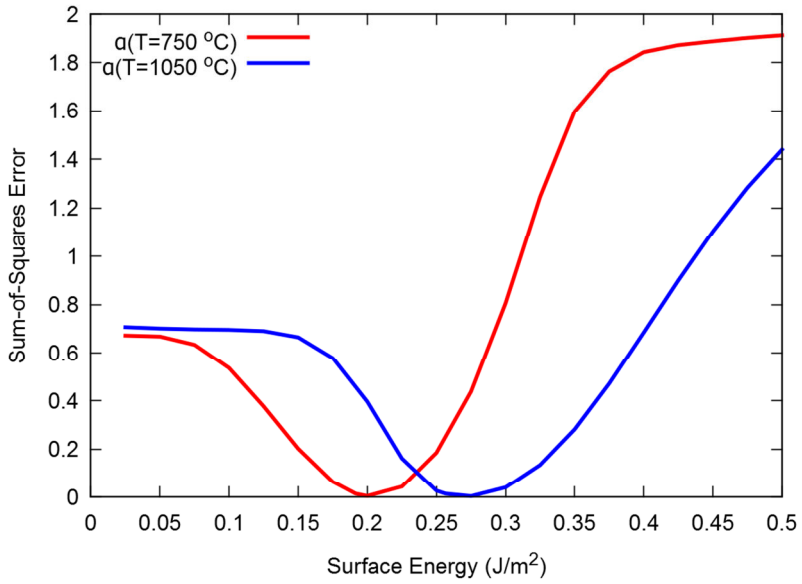

Figure 4.17. Effect of surface energy on fit quality. Only one of the two components of Eq. (4.48), a_{750} or a_{1050}, is varied at a time.

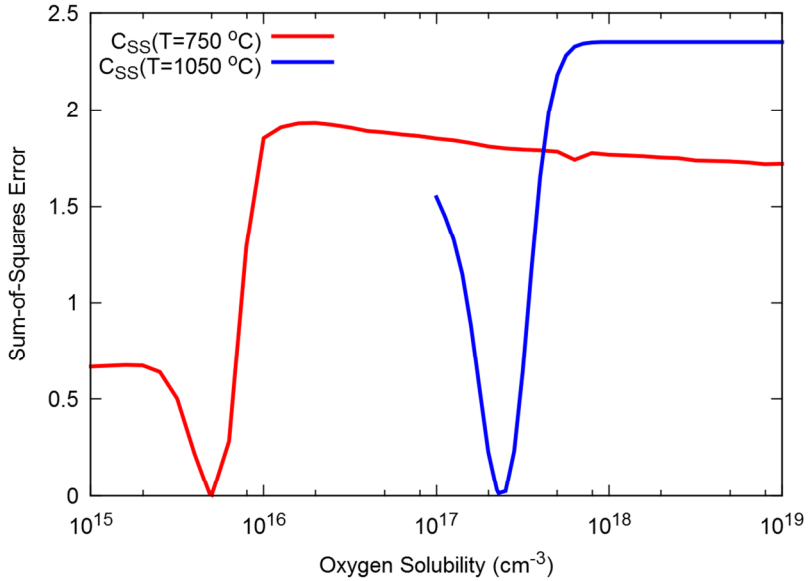

Figure 4.18. Effect of solubility on fit quality. The value of C_{SS} at either 750 or 1050 °C is varied while keeping the other fixed at its best-fit value. Both values are used to compute $C_{SS}(T)$.

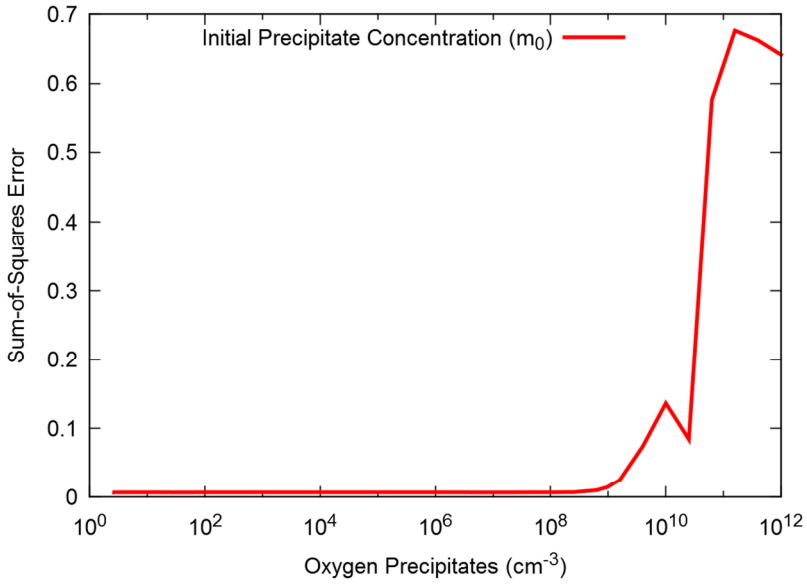

Figure 4.19. Effect of initial precipitate concentration on fit quality.

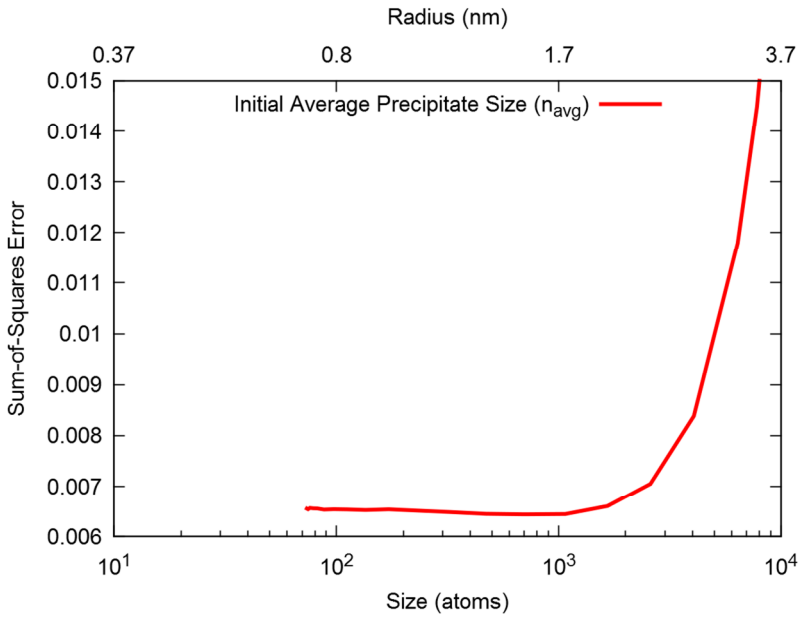

Figure 4.20. Effect of initial average size on fit quality.

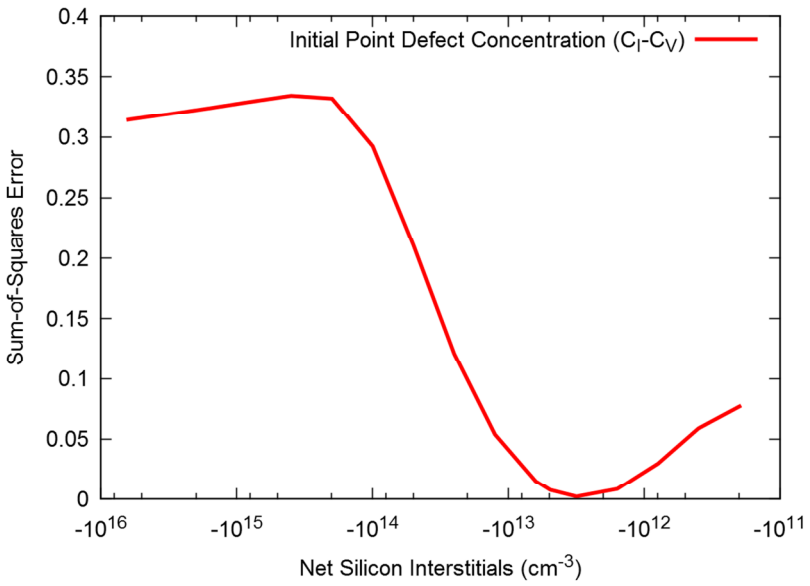

Figure 4.21. Effect of initial point defect concentration on fit quality.

4.2.5 PERFORMANCE AND CONVERGENCE

Table 4.3 reports the run-time performance of the FKPM and RKPM models using different sample discretization factors. A two-step process (800 °C for 2 hr, 1050 °C for 16 hr) was simulated using 12 different initial oxygen concentrations and no initial precipitates for a total of 12 simulations. All simulations were conducted using MATLAB Version 7.0.1.24704 on a workstation equipped with a 3.2 GHz Intel Core i5-2500K processor running Windows 7. The results are depicted visually in Figs. 4.22 and 4.23.

Table 4.3. Run-time performance and number of equations for the FKPM and RKPM models using different discretization ratios.

Sample Discretization Factor (*S*)	**No. of Eqns., FKPM**	**No. of Eqns., RKPM**	**Run Time, FKPM** (sec)	**Run Time, RKPM** (sec)
1.10695	197	36	9173	470
1.20604	118	29	3240	305
1.31681	87	27	1997	282
1.45487	69	24	1691	229
1.72905	53	23	1296	217

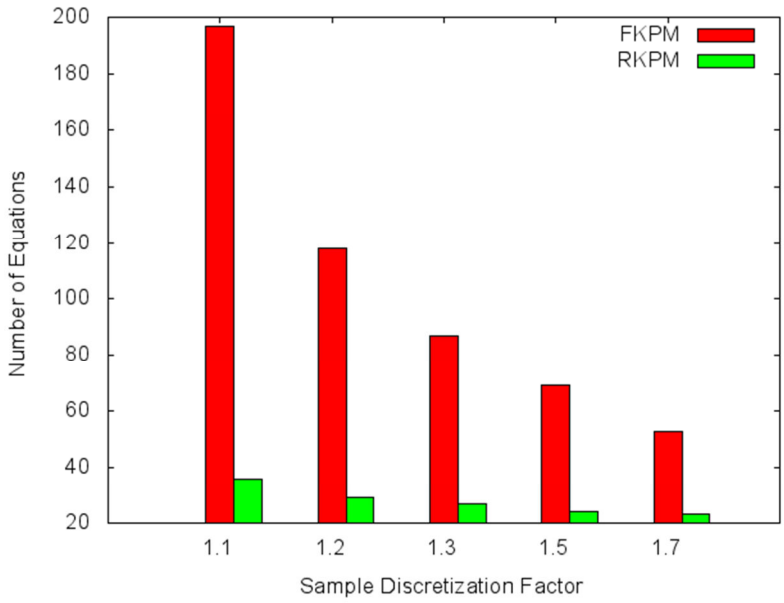

Figure 4.22. Number of equations in the FKPM and RKPM models for different sample discretization factors, *S*.

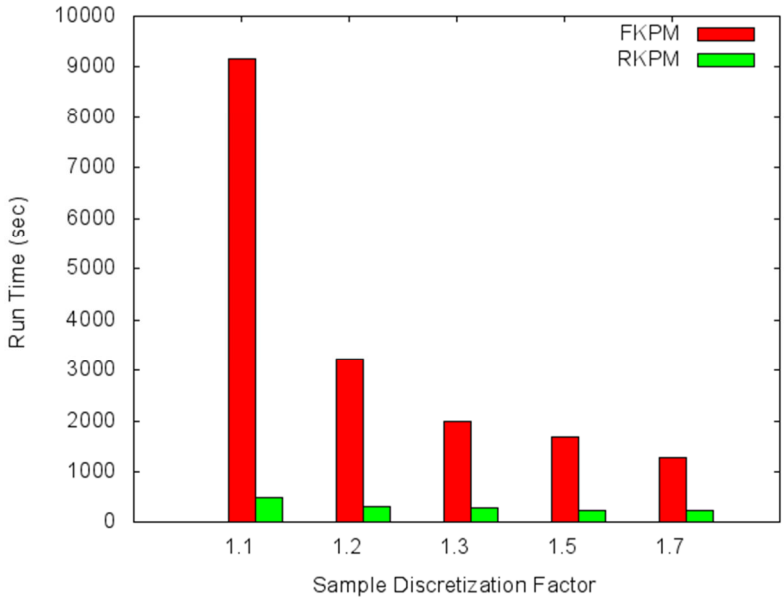

Figure 4.23. Model performance. Run times for a performance test consisting of 12 simulations of a two-step process (800 °C for 2 hr, 1050 °C for 16 hr), each with a different initial oxygen concentration.

Increasing *S* reduces the number of precipitate equations that must be solved (meaning fewer sizes of f_n to solve) and greatly improves run-time performance. This comes at a cost to accuracy, as shown in Figs. 4.24 and 4.25. The two-step experiments are relatively robust with

respect to the number of sample points but the Kennel experiment [114] suffers substantial degradation in accuracy with coarser sampling. The single-temperature anneal by Stewart et al. is not pictured because there is virtually no discernible change over the range of *S* values tested.

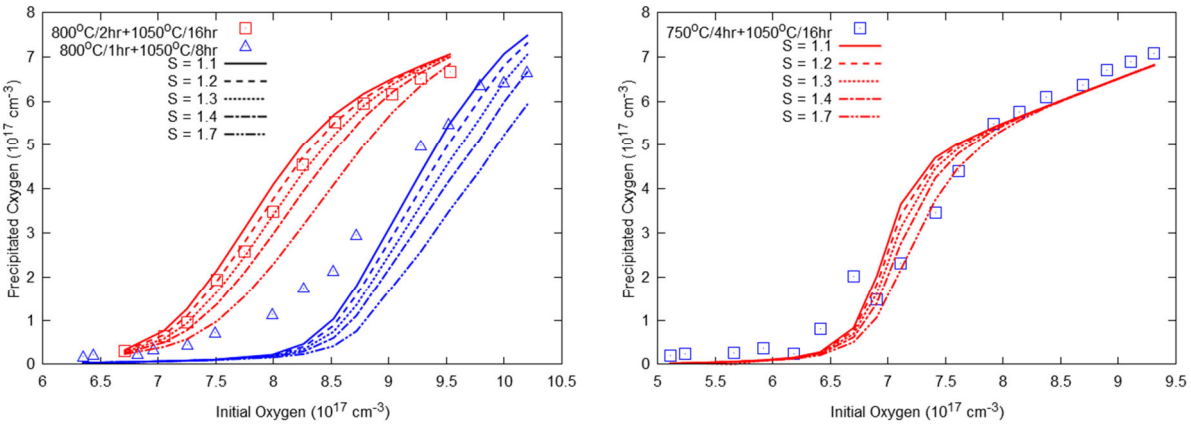

Figure 4.24. Effect of different sample discretization factors (*S*) on RKPM simulations of two-step experiments by Chiou and Shive [83] (left) and Swaroop et al. [112] (right).

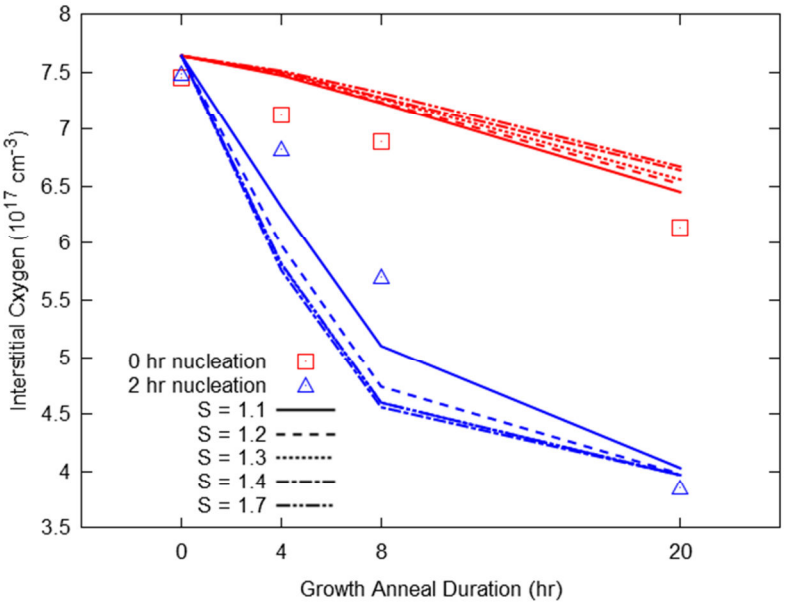

Figure 4.25. Effect of different sample discretization factors (*S*) on RKPM simulations of the experiments by Kennel [114].

The RKPM model is significantly faster than the FKPM version – a factor of 20 faster when S = 1.1. The performance and convergence tests show that S can be increased up to 1.2 without a significant loss in accuracy. It is difficult to compare these results with published oxygen models because their performance and convergence behavior is not discussed in the literature. A study of the impact of parallel computing on an oxygen precipitation model implemented with the Fokker-Planck equation was conducted by Karoui et al. [126].

CHAPTER 5
DISLOCATION LOOP MODEL

A stress-dependent, DFA-based RKPM model for dislocation loop formation from {311} defects in ion-implanted silicon is described in this chapter. It is built atop an RKPM model for {311} defects developed by Guo [108, 127], which is also based on the DFA. *Ab initio* calculations of stacking faults and edge dislocations in silicon were carried out to determine their formation energies and elastic properties. The results were then used to calibrate an RKPM model implemented in a commercial TCAD simulator: Sentaurus Process. The use of atomistic, first principles calculations to provide inputs to higher-level models is an example of the *process modeling hierarchy* described in Refs. [128] and [129].

Dislocation cores are known to trap metals, making them effective for gettering [16, 17, 18]. *Ab initio* calculations of the binding energies of several metals to edge dislocation cores have been carried out and are reported here.

5.1 *AB INITIO* CALCULATIONS

The Latin term *ab initio* means "from the beginning." It describes any analysis that is conducted from first principles, relying only on fundamental physical laws to make predictions without additional assumptions or models. *Density functional theory* (DFT) is a powerful and popular *ab initio* framework for performing quantum mechanically correct calculations of systems involving many atoms and electrons. It is used here to investigate stacking faults and dislocation loops at the atomic scale.

DFT is a quantum mechanical method for obtaining the ground state properties of a system without having to solve the prohibitively complex many-electron time-independent Schrödinger equation. Instead of considering all electron-electron interactions, the problem is reformulated in terms of the electron density. The solution is obtained by minimizing an energy functional of the electron density. Hohenberg and Kohn proved the remarkable fact that there exists a universal energy functional independent of the external potential which, when minimized, yields the *exact* ground state electron density and energy [130]. Unfortunately, the

true form of the energy functional has not been discovered. The most widely used form was suggested by Kohn and Sham [131].

DFT codes further simplify the many-body electron problem by distinguishing between valence electrons and inner core electrons. The electrons in the inner shells are strongly bound and are not relevant to chemical interactions between atoms. They are considered along with the atomic nucleus to form an ionic core and their interaction with the valence electrons is modeled with the use of a *pseudopotential*. In this work, VASP [110], a popular DFT code, and the generalized gradient approximation pseudopotential, GGA-PW91 [132, 133], were used to perform calculations.

5.1.1 STACKING FAULT STRUCTURE

Along <111> directions, crystalline silicon is comprised of a repeating series of identical layers differing only by an offset along <112>. In order to simplify the generation of stacking faults and minimize simulation cell sizes, a local coordinate system with the *z*-axis along [111] is used for all calculations described here. The perpendicular axes are along $[1\bar{1}0]$ (*x*) and $[11\bar{2}]$ (*y*). The unit cell in this orientation, shown in Fig. 5.1, consists of 12 atoms and has dimensions $a_{Si}/\sqrt{2} \times a_{Si}\sqrt{6}/2 \times a_{Si}\sqrt{3}$.

The extrinsic stacking fault depicted in Fig. 2.7 was generated within a supercell one unit cell wide in the *x* dimension, five cells in the *y* dimension, and three in the *z* dimension (1×5×3 cells, 180 atoms). Initially, there were 9 layers, as in the column of perfect silicon on the left side of Fig. 2.7, each occupying $1/9^{th}$ of the total supercell height. Each layer was compressed to $1/10^{th}$ of the supercell height and the layers above the insertion point were then shifted up by $1/10^{th}$ of the height to accommodate a 10^{th} layer of 20 atoms. The system was simulated with volume relaxation enabled and a *k*-point sampling of 1×3×1.

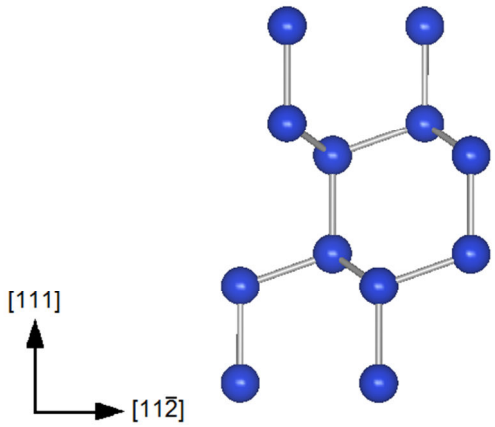

Figure 5.1. Silicon unit cell in the local coordinate system for *ab initio* dislocation calculations.

5.1.2 STACKING FAULT FORMATION ENERGY

The per-atom formation energy, ΔG_{SF}, was found to be very low:

$$\Delta G_{SF} = 0.01525 \text{ eV} \tag{5.1}$$

which is close to experimental estimates [134]. This is due to the fact that the bonding of atoms in the stacking fault to their first nearest neighbors is identical to perfect silicon.

5.1.3 STACKING FAULT INDUCED STRAIN

The stress dependence of the formation energy can be modeled using the concept of *induced strain*, which is the amount of strain that must be applied in order to minimize the formation energy of a defect [129]. Or, put another way, it is the amount of strain that the defect would like to induce in order to reach the minimum energy state. Using the general tensor form of Hooke's Law, the energy of a material system can be expressed as

$$E = E_0 + \frac{\Omega}{2}\sum_{j=1}^{6}\sum_{i=1}^{6}\left(\varepsilon_i - \kappa\Delta\varepsilon_i\right)\cdot\left(C_{ij} + \kappa\Delta C_{ij}\right)\cdot\left(\varepsilon_j - \kappa\Delta\varepsilon_j\right) \tag{5.2}$$

where E_0 is the minimum energy of the system and Ω is its volume. The 6 components of the vector form of the applied stress tensor, ε_i, are derived from the more general 3×3 matrix form with components ε_{ij}.

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{pmatrix} = \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{23} \\ 2\varepsilon_{31} \\ 2\varepsilon_{12} \end{pmatrix} \tag{5.3}$$

In the matrix form, subscripts *i* and *j* each range from 1 to 3 (the *x*, *y*, and *z* axes, respectively).

The *elasticity tensor* (also referred to as the *stiffness tensor*), **C**, is in general a tensor of rank 4 with elements denoted by C_{ijkl}. In cubic materials such as silicon, symmetries allow it to be expressed as a 6×6 matrix with only three unique components:

$$\mathbf{C} = \begin{pmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{pmatrix} \tag{5.4}$$

Localized modifications to the elasticity tensor caused by a defect are expressed as **ΔC**, which was found to be negligible in the case of stacking faults.

The induced strain is given by $\kappa\boldsymbol{\Delta\varepsilon}$, where κ is a normalization factor,

$$\kappa = \frac{I}{N} \tag{5.5}$$

with *I* being the number of interstitials comprising the defect in a supercell containing *N* lattice sites. The *normalized induced strain* is **Δε** and represents the induced strain scaled to atomic dimensions. The normalized induced strain can easily be used to estimate the induced strain for any defect size simply by applying the correct scaling factor.

The induced strain was estimated by comparing the supercell volumes before and after relaxation. VASP provides a coordinate system matrix of the form

$$\mathbf{R} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \tag{5.6}$$

The **R** matrix is interpreted as three column vectors that describe the orientation and length of the simulation supercell's *x*, *y*, and *z* dimensions and which need not be orthogonal. Given an *orthogonal* initial coordinate system (all off-diagonal components set to zero), $\mathbf{R_0}$, and a *non-orthogonal* resultant coordinate system, $\mathbf{R_1}$, the components of the induced strain tensor, $\boldsymbol{\varepsilon}^{\mathbf{induced}}$, are computed as follows:

$$\kappa\Delta\varepsilon_{ii} = \varepsilon_{ii}^{induced} = \frac{R_{1,ii} - R_{0,ii}}{R_{0,ii}} \tag{5.7}$$

$$\kappa\Delta\varepsilon_{ij} = \varepsilon_{ij}^{induced} = \frac{1}{2}\left(\frac{R_{1,ij} - R_{0,ij}}{R_{0,jj}} + \frac{R_{1,ji} - R_{0,ji}}{R_{0,ii}}\right) \tag{5.8}$$

The 'induced' superscript is used to prevent the induced strain from being mistaken for the applied strain, $\boldsymbol{\varepsilon}$, in Eq. (5.2) but is omitted for the sake of readability when writing the normalized induced strain, $\boldsymbol{\Delta\varepsilon}$.

Table 5.1 reports the normalized induced strain in the local coordinate system. The strain is primarily along the *z*-axis and has a magnitude of nearly 1.0, which is expected because it is caused by the insertion of an extra plane of atoms. Table 5.2 shows the normalized induced strain in the standard coordinate system for all possible orientations of the stacking fault.

Table 5.1. Normalized induced strain tensor ($\boldsymbol{\Delta\varepsilon}$) of the {111} stacking fault in its local coordinate system. $\Delta\varepsilon_{ij} = \Delta\varepsilon_{ji}$.

$\Delta\varepsilon_{11}$	$\Delta\varepsilon_{22}$	$\Delta\varepsilon_{33}$	$\Delta\varepsilon_{12}$	$\Delta\varepsilon_{13}$	$\Delta\varepsilon_{23}$
-0.0356	-0.0232	0.996	0.086	0.0	-0.00665

Table 5.2. Normalized induced strain tensor reported in the standard coordinate system for all possible orientations of the stacking fault.

Habit Plane	$\Delta\varepsilon_{11}$	$\Delta\varepsilon_{22}$	$\Delta\varepsilon_{33}$	$\Delta\varepsilon_{12}$	$\Delta\varepsilon_{13}$	$\Delta\varepsilon_{23}$
(111)	0.356851	0.257546	0.322803	0.342799	0.291649	0.390953
$(\bar{1}11)$	0.257546	0.356851	0.322803	-0.342799	-0.390953	0.291649
$(1\bar{1}1)$	0.257546	0.356851	0.322803	-0.342799	0.390953	-0.291649
$(11\bar{1})$	0.356851	0.322803	0.257546	0.291649	-0.342799	-0.390953

5.1.4 EDGE DISLOCATION STRUCTURES AND FORMATION ENERGIES

Terminating the extrinsic stacking fault results in an edge dislocation. A*b initio* calculations were utilized to explore possible dislocation core structures. In order to keep supercell sizes small and computationally tractable, terminations were made only along high-symmetry directions: the x and y axes. A dislocation dipole (two edge dislocations located on either end of the stacking fault plane) was formed, which is the smallest configuration possible in a system with periodic spatial boundary conditions. Although rectangular loops can easily be generated, they require much larger supercells.

The procedure previously described for generating a stacking fault was used to insert an extra plane of atoms only in select vertical columns while leaving the rest of the supercell unperturbed, creating a partial stacking fault that does not extend to the full width of the supercell. The edges of the stacking fault are the locations of the two edge dislocations.

In order to find probable dislocation core structures, a relaxation of the supercell was conducted in three steps:

1. Local relaxation with a fixed supercell volume. VASP attempts to find the minimum energy position of all atoms in the system while holding the supercell volume and shape fixed.
2. Full relaxation using the result from step 1. The supercell volume is now allowed to change in addition to the positions of the atoms.
3. Local relaxation using the result from step 2 with a fixed supercell volume. This step is necessary because the energy obtained in the previous step is inaccurate due to the volume change.

The relaxation algorithm for ionic steps was alternated between quasi-Newton and conjugate gradient minimization as necessary to achieve convergence.

The Burgers vector for this dislocation is

$$\mathbf{b} = \frac{a_{Si}}{3}\langle \mathbf{111} \rangle \tag{5.9}$$

Two different dislocation core structures were discovered along $[11\bar{2}]$: Structure A, shown in Fig. 5.2, and Structure B, shown in Fig. 5.3. Structure B was found to have lower

formation energy under stress-free conditions and also appears to be more stable (easier to reproduce within different supercell volumes). Fig. 5.4 is an image of a complete supercell (Si324D16).

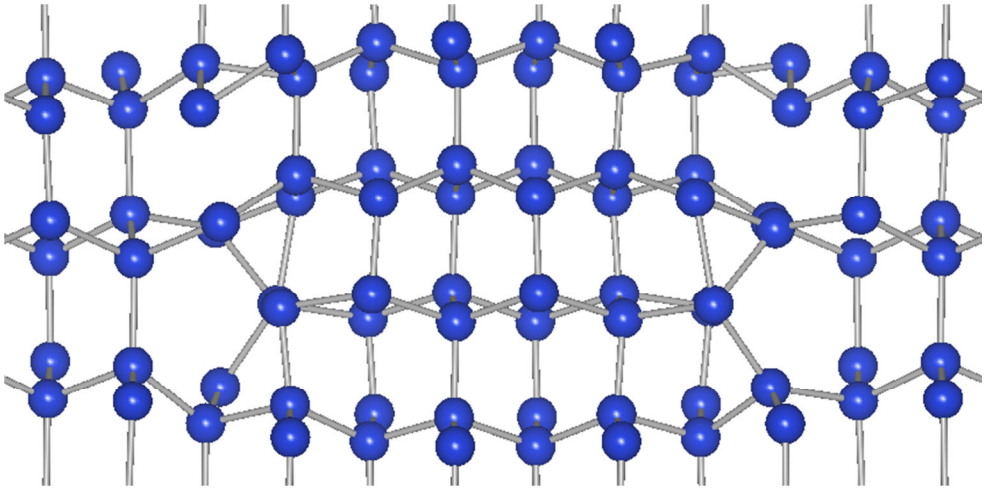

Figure 5.2. A dislocation dipole with core Structure A, viewed along [11$\bar{2}$].

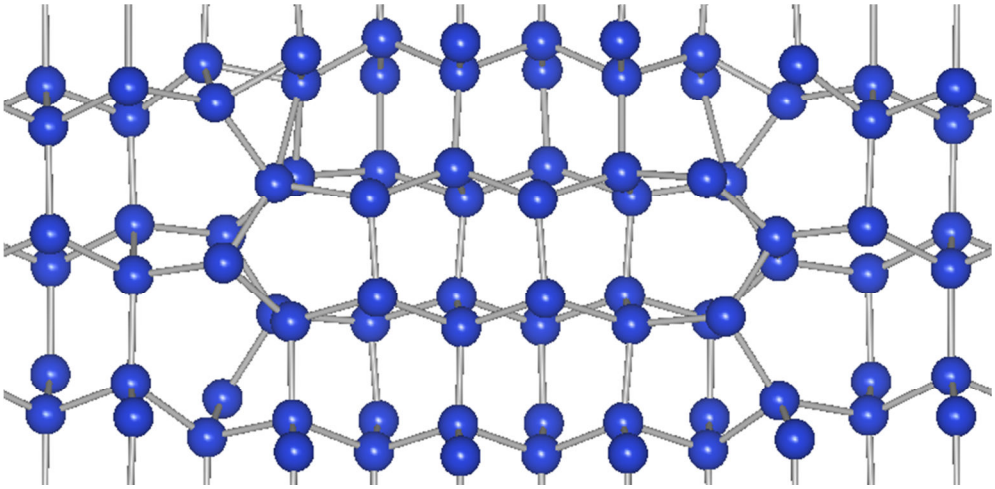

Figure 5.3. A dislocation dipole with core Structure B.

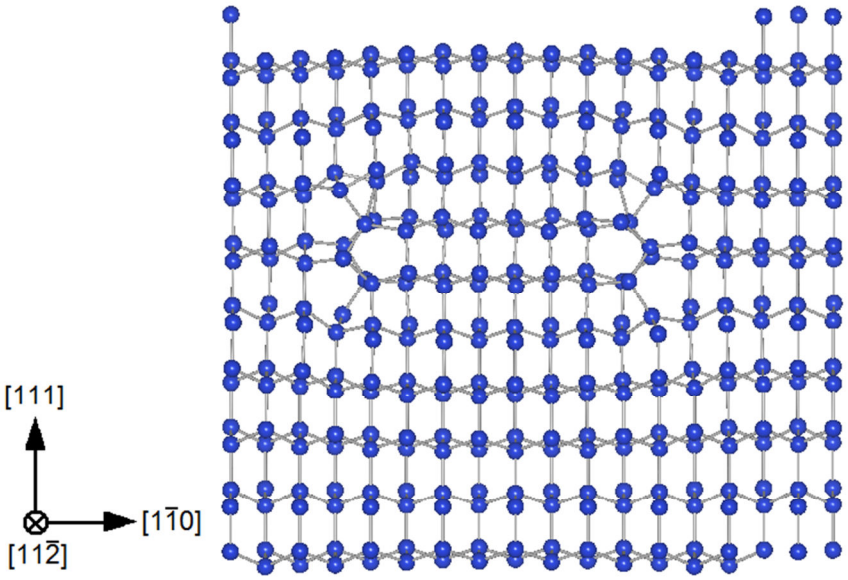

Figure 5.4. Complete supercell with 324 lattice sites and 16 interstitial atoms forming a stacking fault and dislocation dipole. Periodic boundary conditions have caused some atoms to shift from the top row to the bottom.

5.1.5 EDGE DISLOCATION INDUCED STRAIN

Under the assumption of linear elasticity, strain fields can be treated with linear superposition. The induced strain of a dislocation dipole can therefore be separated into two components: a planar (stacking fault) component and an edge (dislocation core) component. The normalization factor for the stacking fault contribution is, as before, simply I/N, where N is the total number of lattice sites in the system (which does *not* include interstitial atoms forming the stacking fault) and I is the number of interstitial atoms. For example, the Si324D16 system of Fig. 5.4 contains 340 atoms of which 16 are interstitials. The normalization factor is $\varkappa = 16/324$.

For the dislocation cores, it is not clear which of the interstitial atoms should be counted as belonging to the core. The choice is somewhat arbitrary. Here, all interstitial atoms are counted toward the planar induced strain (and the normalization can be thought of in terms of atomic volumes) but the core strain is normalized based on dislocation line length in angstroms divided by N. Given the normalized induced strain components, $\mathbf{\Delta\varepsilon}^{\mathrm{SF}}$ for the stacking fault and $\mathbf{\Delta\varepsilon}^{\mathrm{core}}$ for the core, the total induced strain of the entire dipole system is

$$\boldsymbol{\varepsilon}^{\mathrm{induced}} = \frac{I}{N}\mathbf{\Delta\varepsilon}^{\mathrm{SF}} + \frac{2L}{N}\mathbf{\Delta\varepsilon}^{\mathrm{core}} \tag{5.10}$$

where L is the length of the dislocation line. The dislocation core component is multiplied by 2 because a dipole system contains two dislocations.

The stacking fault normalized induced strain was found from the full stacking fault calculation to be 0.996 along the z axis with negligible x and y components. By fitting Eq. (5.10) to the dipole systems with Structure B, the z component of $\boldsymbol{\Delta\varepsilon}^{\mathbf{core}}$ was found to be 0.147 Å^{-1}.

A circular loop containing I interstitial atoms in a volume of N lattice sites has an induced strain

$$\boldsymbol{\varepsilon}^{\mathbf{induced}} = \frac{I}{N}\boldsymbol{\Delta\varepsilon}^{\mathbf{SF}} + \frac{2\pi \cdot r_I^{DL}}{N}\boldsymbol{\Delta\varepsilon}^{\mathbf{core}} \tag{5.11}$$

where r_I^{DL} is the radius of a dislocation loop containing I atoms, given by Eq. (4.38). The area per interstitial atom is

$$A = \frac{\sqrt{3}}{8}a_{Si}^2 \tag{5.12}$$

The total area of the circular loop is

$$A \cdot I = \pi \cdot \left(r_I^{DL}\right)^2 \tag{5.13}$$

The total induced strain can then be written as

$$\boldsymbol{\varepsilon}^{\mathbf{induced}} = \frac{I}{N}\boldsymbol{\Delta\varepsilon}^{\mathbf{SF}} + \frac{a_{Si}}{N}\left(\frac{I\pi\sqrt{3}}{2}\right)^{1/2}\boldsymbol{\Delta\varepsilon}^{\mathbf{core}} \tag{5.14}$$

The fraction of induced strain contributed by the dislocation core decreases as the loop size increases. This can be seen by taking the ratio of the core contribution to the stacking fault contribution:

$$\frac{\frac{a_{Si}}{N}\left(\frac{I\pi\sqrt{3}}{2}\right)^{1/2}\boldsymbol{\Delta\varepsilon}^{\mathbf{core}}}{\frac{I}{N}\boldsymbol{\Delta\varepsilon}^{\mathbf{SF}}} = a_0\left(\frac{\pi\sqrt{3}}{2I}\right)^{1/2}\frac{\boldsymbol{\Delta\varepsilon}^{\mathbf{core}}}{\boldsymbol{\Delta\varepsilon}^{\mathbf{SF}}} \tag{5.15}$$

This ratio reaches a maximum value of 1.36 when I = 1, falls to 0.14 when I = 100, and at I = 1000 is only 0.04. Typically, dislocation loops are many thousands of atoms in size (and larger). Therefore, the induced strain of edge dislocations is neglected in the dislocation loop model.

5.1.6 METAL DECORATION

Metal *decoration* refers to the binding of metal atoms to dislocation cores. A preliminary study of metal binding to dislocation cores was conducted using a very small dislocation dipole (Structure B) consisting of only 16 interstitial atoms in a cell with 240 lattice sites. The cell size along the $[11\bar{2}]$ direction was twice as large as in the other calculations in order to minimize interactions between the metal atoms and their periodic images. Calculations were performed using only a single k-point sample (Γ point). Most calculations were carried out with neutrally charged metal atoms, with the exceptions of copper and cobalt, for which the simulation supercell was given the same net charge as the most favorable charge state of the interstitial metal ion. The supercell was allowed to relax its volume in all dimensions to minimize free energy.

The binding energy is defined as

$$E_B^X = E_{Si240D16X} + E_{Si240} - \left(E_{Si240X} + E_{Si240D16}\right) \tag{5.16}$$

where X denotes the metal species. The free energy of a 240-site supercell with a 16-atom dislocation dipole and a single metal atom is denoted by $E_{Si240D16X}$; $E_{Si240D16}$ is the free energy of the equivalent system lacking the metal. The ground state is considered to be perfect silicon with a metal atom in its lowest energy configuration, which is a tetrahedral site. The free energy of a 240-atom supercell of perfect silicon is given by E_{Si240} and E_{Si240X} is the free energy of the same system with a metal atom present in a tetrahedral site.

The most favorable attachment sites near the dislocation core are interstitial positions where the silicon bonds are elongated and distorted. Fig. 5.5 shows the resulting structure of molybdenum bound to the dislocation core. Table 5.3 lists the binding energies for different metal species. Some of the results, especially the peculiarly high binding energy of titanium, are poorly converged and may not be accurate. More favorable sites along different dislocation line directions, for which the dislocation core structure has not been explored in this work, may also exist.

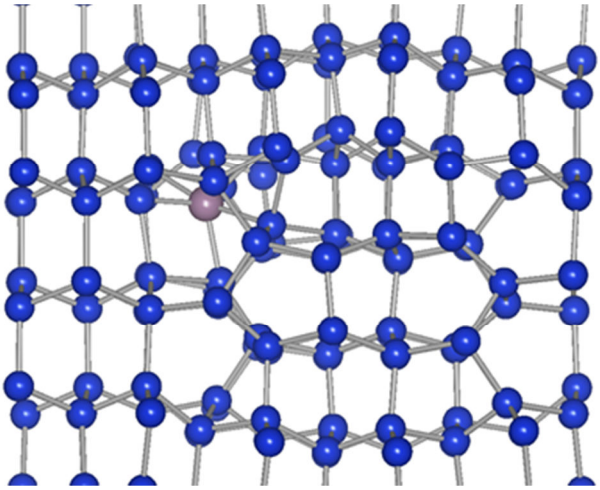

Figure 5.5. Binding of molybdenum to the dislocation core.

Table 5.3. Calculated binding energies of metals to the dislocation core.

Metal Species	**Binding Energy** (eV)
Cu^{+}	-0.82
Cr	-1.14
Mo	-1.07
Ti	-3.40
W	-1.50
Fe	-1.85
Co^{-}	-1.52

These results were not used in the dislocation loop model and are reported here only for the sake of posterity.

5.2 MODEL

The RKPM dislocation loop model is implemented atop a {311} defect and small interstitial cluster model by Guo [108, 127]. An earlier combined {311} defect and dislocation loop model based on the RKPM approach was developed by Gencer and Dunham [36]. It tracks both {311} defects and dislocation loops using the same set of moments and assumes all precipitates are {311} defects below a fixed critical size (1000) and dislocation loops above. The Guo model

treats {311} defects and small interstitial clusters as separate, parallel reaction pathways. This model is extended with a third pathway for dislocation loop evolution, allowing them to nucleate homogeneously as well as from existing {311} defects [38, 39, 40], depicted schematically in Fig. 5.6. More information about the kinetics of {311} defects can be found in Ref. [107].

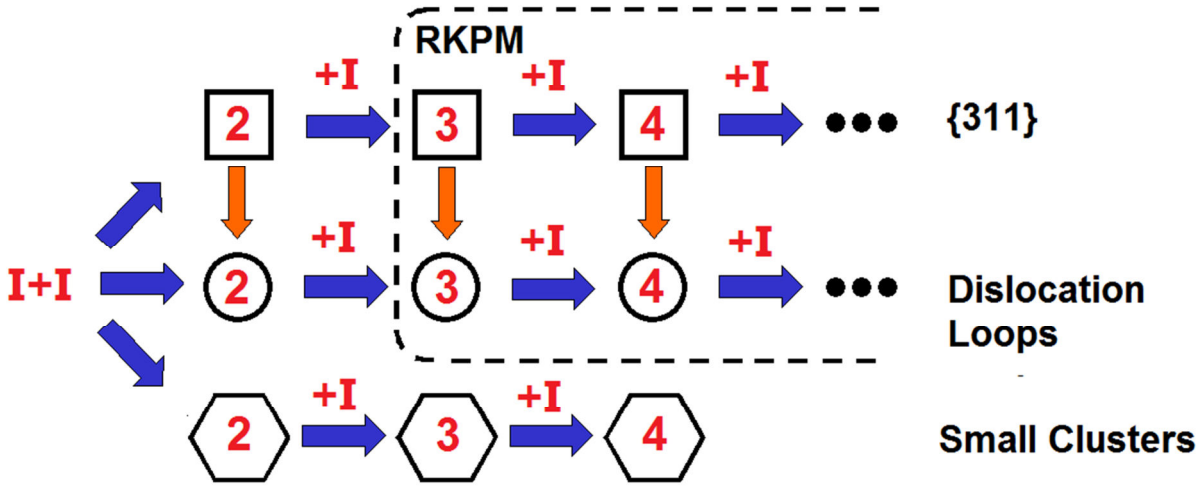

Figure 5.6. Reaction pathways for interstitial aggregates. Dislocation loops and {311} defects are tracked by separate RKPM equations coupled together by a transformation rate.

The model presented in this chapter is distinct from the simple dislocation model used in the oxygen precipitation model. Many of the same variable names will therefore be reused and care must be taken to avoid conflating them with their counterparts in the oxygen model.

5.2.1 ENERGY OF DISLOCATION LOOPS

The free energy of a dislocation loop, ΔG_n^{DL}, is modeled as consisting of several components:

$$\Delta G_n^{DL} = -nk_B T \ln\left(\frac{C_I}{C_I^*}\right) + n \cdot \Delta G_{SF} + 2\pi \cdot r_n^{DL} \Delta G_{core} + \Delta G_n^{self} + n \cdot \Delta G_{SF}^{strain}(\boldsymbol{\sigma}) \qquad (5.17)$$

where ΔG_{core} is the core energy per unit length (eV/Å), ΔG_n^{self} is the elastic self-energy, and ΔG_{SF}^{strain} is the change in free energy due to applied stress, $\boldsymbol{\sigma}$. The first two terms can be combined by defining an effective solubility for dislocation loops, which is based on the difference in the energy of the stacking fault phase and dispersed interstitial atoms.

$$\Delta G_n^{DL} = -nk_B T \ln\left(\frac{C_I}{C_{SS}^{DL}}\right) + 2\pi \cdot r_n^{DL} \Delta G_{core} + \Delta G_n^{self} + n \cdot \Delta G_{SF}^{strain}(\boldsymbol{\sigma}) \tag{5.18}$$

where

$$C_{SS}^{DL} = C_I^* \exp\left(\frac{\Delta G_{SF}}{k_B T}\right) \tag{5.19}$$

The need for a modified solubility can be explained by the fact that all energies are defined relative to perfect silicon, meaning that the interstitial formation energy must be considered as well, which is why C_I^* appears in the energy expression.

The elastic self-energy for a dislocation loop with radius r in an isotropic crystal, W_{iso}, is derived in Ref. [37]. Assuming isotropy, the result for an edge dislocation loop in silicon is

$$W_{iso} = r_n^{DL} \frac{\mu_{Si} b^2}{2(1-\upsilon_{Si})}\left[\ln\left(\frac{8r_n^{DL}}{r_{core}}\right) - 1\right] \tag{5.20}$$

The constants μ_{Si} and υ_{Si} are the shear modulus and Poisson's ratio in silicon, respectively. Because silicon is in fact an anisotropic crystal, the isotropic elastic constants must be replaced with an energy pre-factor, K_{DL}, that accounts for the crystal structure and the orientation of the edge dislocation [37]:

$$\frac{\mu_{Si}}{1-\upsilon_{Si}} \rightarrow K_{DL} \tag{5.21}$$

A reasonable estimate for modeling purposes appears in Ref. [125]:

$$K_{DL} = 72 \text{ GPa} \tag{5.22}$$

The anisotropic elastic self-energy, W_{aniso}, which is the energy used for ΔG_n^{self}, is

$$W_{aniso} = \Delta G_n^{self} = r_n^{DL} \frac{K_{DL} b^2}{2}\left[\ln\left(\frac{8r_n^{DL}}{r_{core}}\right) - 1\right] \tag{5.23}$$

The elastic self-energy corresponds to the amount of work done to deform the lattice. It is only valid beyond the core radius, where linear elasticity applies. The portion of the energy attributable to the region within the core radius, where atomistic effects dominate, is described by ΔG_{core}. The core energy can be extracted from atomistic simulations after correcting for the

effects of periodic boundary conditions, which create interactions with an infinite array of virtual dislocation dipoles. A method for this is described in Refs. [135] and [136]. Unfortunately, the method could not be successfully replicated and therefore, the core energy was estimated by taking the result from Ref. [125].

$$\Delta G_{core} = 0.75 \text{ eV/Å} \tag{5.24}$$

Although this estimate is for a Frank partial dislocation, it is expected to be of similar magnitude. Given the uncertainty in defining the core radius, this is arguably a justifiable approximation for modeling purposes.

The change in formation energy due to applied stress can be computed using Hooke's Law,

$$\Delta G_{SF}^{strain}(\boldsymbol{\sigma}) = -\frac{V_{Si}}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\Delta\varepsilon_{ij}\sigma_{ij} \tag{5.25}$$

where V_{Si} is the atomic volume of silicon, as in the oxygen model, and $\boldsymbol{\Delta\varepsilon}$ is the normalized induced strain of the stacking fault. Tensile stress normal to the stacking fault habit plane (<111> directions) reduces the formation energy and enhances the formation of dislocation loops.

5.2.2 TRANSFORMATION OF {311} DEFECTS INTO DISLOCATION LOOPS

Dislocations become thermodynamically favorable over {311} defects at large sizes. Fig. 5.7 compares the free energies of dislocations and {311} defects. Beyond the *crossover point*, the energy of dislocation loops becomes lower than that of {311} defects and transformation of {311} defects into dislocations will occur. The crossover point is likely to be much smaller than the actual size at which the transformation happens because a substantial rearrangement of interstitial atoms is involved. Little is known about the physics of this process but the energy barrier is probably significant. Once formed, dislocation loops are very stable and do not transform back into {311} defects.

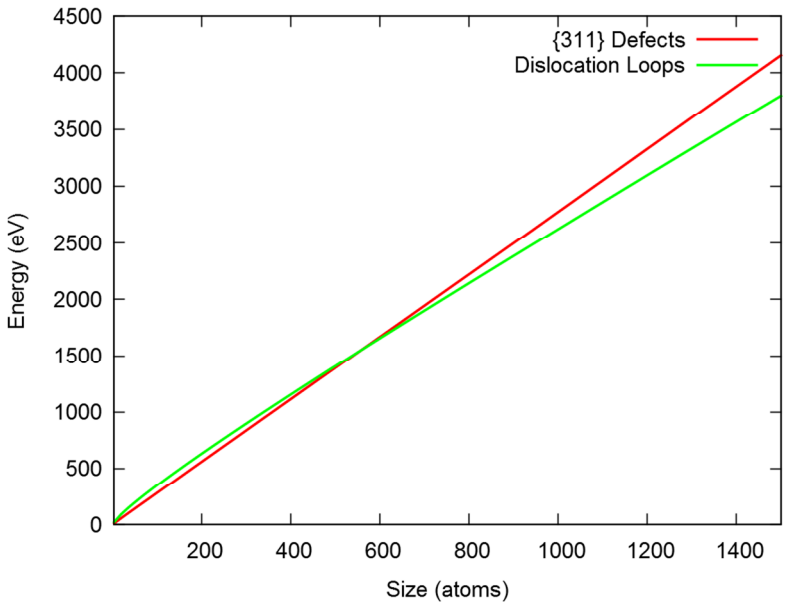

Figure 5.7. Free energy of extended defects as a function of size. The common factor of $nk_BT\ln(C_I)$ has been removed from both energies.

To estimate the transformation rate, it is assumed that the process is dependent on an attempt frequency, ν_0 (sec^{-1}), modified by a transition barrier, E_b, and driven by the difference between the energies of both defects:

$$R_{\{311\}\to DL}(n) = \nu_0 \exp\left(\frac{-E_b}{k_B T}\right)\exp\left(\frac{\Delta G_n^{\{311\}} - \Delta G_n^{DL}}{2k_B T}\right) \tag{5.26}$$

The transition barrier is almost certainly dependent on defect geometry but absent an understanding of the transformation kinetics, it was left as a constant. The transformation rate for each defect must be summed in order to obtain the total transformation rate, T_0, for all {311} defects. This rate and the corresponding rate for atoms, T_1, is

$$T_i = \sum_{n=k}^{\infty} n^i R_{\{311\}->DL}(n) \cdot \frac{f_n^{\{311\}}}{m_0^{\{311\}}} \tag{5.27}$$

where i is either 0 (T_0) or 1 (T_1) and $f_n^{\{311\}}$ is the concentration of size n {311} defects. As was discussed in Section 2.2.2, the size distribution of {311} defects is known to be log-normal. The rate is normalized by the total number of defects, $m_0^{\{311\}}$. In an RKPM model based on the DFA, computing the transformation rate only at the average size would be inaccurate because it would neglect the tail of the distribution. Even though the transformation rate at a given

average size may be relatively small, defects located in the tail of the distribution may be sufficiently numerous to cause a significant overall transformation rate.

Beyond the crossover point, T_0 grows extremely rapidly, so much so that the validity of this simple model must be questioned. Based on experimentation, it was discovered that more flexibility in setting the crossover point was needed to achieve a good fit to experimental data. It was also reasoned that at that point, the transition would likely be rapid; after all, simple models that assume instantaneous transformation beyond a fixed size have been used with a high degree of success. The following approximation was devised for T_0:

$$T_0\left(n_{avg}^{\{311\}}\right) = \omega(T)\cdot\left(\frac{n_{trans}}{n_{avg}^{\{311\}}}+1\right)^{-12} \tag{5.28}$$

where $n_{avg}^{\{311\}}$ is the average {311} defect size computed from the moments, as in Eq. (3.33). The size at which T_0 increases rapidly is called the *transformation size* and is set by the parameter n_{trans}. A second parameter, ω, which is a function of temperature alone, is used to tune the magnitude of the transformation rate. In Fig. 5.8, T_0 is shown for a few different values of ν_0 and E_b along with the above approximation.

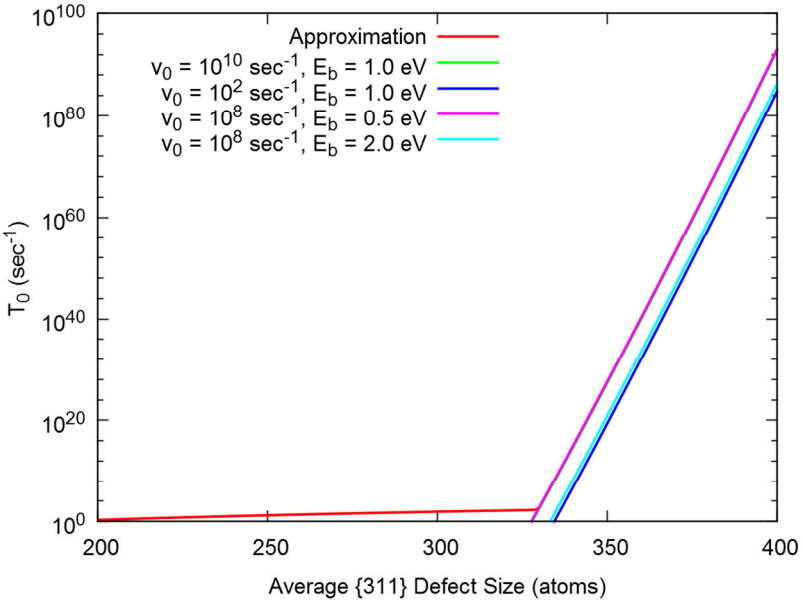

Figure 5.8. The transformation rate of {311} defects into dislocation loops, T_0, for different values of the attempt frequency, ν_0, and barrier energy, E_b, alongside the approximation of Eq. (5.28) at T = 1000 °C.

Because the DFA assumes the defect size distribution is sharply peaked around $n_{avg}^{\{311\}}$, T_1 is estimated simply as

$$T_1\left(n_{avg}^{\{311\}}\right) = n_{avg}^{\{311\}} \cdot T_0\left(n_{avg}^{\{311\}}\right) \tag{5.29}$$

5.2.3 REDUCED KINETIC PRECIPITATION MODEL

The dislocation loop model was implemented only as an RKPM model – an FKPM model was not developed. The transition size k was chosen to be 3 and the DFA was used to model the size distribution. As with the oxygen model, the RKPM formulation is essentially the same as is described in Chapter 3 with the following substitutions:

$$\begin{aligned} C &\rightarrow C_I \\ D &\rightarrow D_I \\ C_S &\rightarrow C_{Si} \end{aligned} \tag{5.30}$$

Interstitial silicon, C_I, is the solute.

The equations for the moments include the transformation rate terms, T_i. For {311} defects, they are

$$\frac{\partial m_0^{\{311\}}}{\partial t} = R_3^{\{311\}} - m_0^{\{311\}} \cdot T_0 \tag{5.31}$$

$$\frac{\partial m_1^{\{311\}}}{\partial t} = 3R_3^{\{311\}} + D_I \cdot m_0^{\{311\}} \cdot \left(C_I \cdot \gamma_2^{\{311\}} - \gamma_3^{\{311\}}\right) - m_0^{\{311\}} \cdot T_1 \tag{5.32}$$

Likewise, the dislocation loop equations are

$$\frac{\partial m_0^{DL}}{\partial t} = R_3^{DL} + m_0^{\{311\}} \cdot T_0 \tag{5.33}$$

$$\frac{\partial m_1^{DL}}{\partial t} = 3R_3^{DL} + D_I \cdot m_0^{DL} \cdot \left(C_I \cdot \gamma_2^{DL} - \gamma_3^{DL}\right) + m_0^{\{311\}} \cdot T_1 \tag{5.34}$$

Vacancy interactions were also included but are omitted here for the sake of brevity. The growth rate is the same as in Eq. (4.41) but the dissolution rate differs slightly from Eq. (4.42) in that a discrete difference between adjacent free energies is used rather than a continuous derivative.

$$d_{DL}(n) = g_{DL}(n-1) \cdot \exp\left(\frac{\Delta G_n^{DL} - \Delta G_{n-1}^{DL}}{k_B T}\right) \tag{5.35}$$

The k-1/k boundary condition is handled with the same functional form for f_k as is used in the oxygen model: Eq. (4.45).

5.2.4 STRESS DEPENDENCE

Applied stress changes both the energy and diffusivity of defects. The model allows an applied stress tensor, **σ**, to be defined. Using induced strain data obtained from *ab initio* calculations for stacking faults, {311} defects, small interstitial clusters, and interstitials, the change in free energy is computed, which in turn affects growth and dissolution rates, making some reaction pathways more favorable than others.

In the model, the stress dependences of the defect species are folded into the definitions of their respective solubilities and equilibrium concentrations.

$$C_{SS}^{X} \rightarrow C_{SS}^{X} \exp\left(\frac{\Delta G_X^{strain}(\boldsymbol{\sigma})}{k_B T}\right) \tag{5.36}$$

$$C_{X}^{*} \rightarrow C_{X}^{*} \exp\left(\frac{-\Delta G_X^{strain}(\boldsymbol{\sigma})}{k_B T}\right) \tag{5.37}$$

In order to account for the asymmetry of defects, it is assumed that all possible defect orientations are in relative equilibrium with each other. Therefore, weighted average values are used.

The $D_I C_I^*$ product is known to change anisotropically with strain [129]. Along each axis, $D_I C_I^*$ responds differently to parallel and perpendicular stresses. This modification can be described by the following equations:

$$\frac{D_I^x C_I^*(\boldsymbol{\sigma})}{D_I^x C_I^*(0)} = \exp\left\{\frac{-V_{Si}\left(\Delta\varepsilon_{parallel}\sigma_{11} + \Delta\varepsilon_{perpendicular}\sigma_{22} + \Delta\varepsilon_{perpendicular}\sigma_{33}\right)}{2k_B T}\right\} \tag{5.38}$$

$$\frac{D_I^y C_I^*(\boldsymbol{\sigma})}{D_I^y C_I^*(0)} = \exp\left\{\frac{-V_{Si}\left(\Delta\varepsilon_{perpendicular}\sigma_{11} + \Delta\varepsilon_{parallel}\sigma_{22} + \Delta\varepsilon_{perpendicular}\sigma_{33}\right)}{2k_B T}\right\} \tag{5.39}$$

$$\frac{D_I^z C_I^*(\boldsymbol{\sigma})}{D_I^z C_I^*(0)} = \exp\left\{\frac{-V_{Si}\left(\Delta\varepsilon_{perpendicular}\sigma_{11} + \Delta\varepsilon_{perpendicular}\sigma_{22} + \Delta\varepsilon_{parallel}\sigma_{33}\right)}{2k_B T}\right\} \tag{5.40}$$

where D_I^x is the interstitial diffusivity along the x-axis, etc., and $\Delta\varepsilon_{parallel}$ and $\Delta\varepsilon_{perpendicular}$ are the normalized induced strains parallel and perpendicular to the direction of motion, respectively.

5.3 RESULTS

The RKPM model was compared to a series of experiments involving {311} defects and dislocation loops. The parameters of the transformation rate, Eq. (5.28), and f_k, Eq. (4.45), were determined at different temperatures and then fitted to temperature-dependent forms.

5.3.1 COMPARISON TO EXPERIMENT

A series of experiments by Pan et al. [42] studied the formation and growth of dislocation loops at two different temperatures: 850 °C furnace anneals and 1050 °C rapid thermal anneals. CZ silicon wafers were implanted with Si^+ ions at 50 keV and a dose of 10^{16} cm^{-2}. This was high enough to produce an amorphous region at the wafer surface. The implant damage (silicon atoms dislodged from their lattice sites by collisions with ions) and the silicon ions themselves created a high interstitial supersaturation. During the subsequent thermal processing, dislocation loops formed and grew.

The ion implant was replicated in Sentaurus Process using the Crystal-TRIM kinetic lattice Monte Carlo model with a manually-tuned amorphization depth. Fig. 5.9 compares the simulation results with the observed density of interstitials bound to dislocation loops. The average dislocation loop size over time is shown in Fig. 5.10.

Initially, interstitials accumulate into {311} defects and then transform into dislocation loops or dissolve. The population of interstitials bound to loops saturates when no more free interstitials exist but the average loop size continues to increase. As dislocation loops form and begin to grow, the concentration of free interstitials falls, which causes the critical size to rise. This in turn causes smaller dislocations to dissolve and the released interstitials become absorbed by larger loops. This phenomenon is known as *Ostwald ripening*. Over very long periods of time, the loops slowly begin to shrink by releasing interstitials that then diffuse away and become

absorbed at the oxide/silicon interface before they can be recaptured by other loops or {311} defects. Fig. 5.11 helps illustrate this process by explicitly showing how both types of extended defect evolve over time.

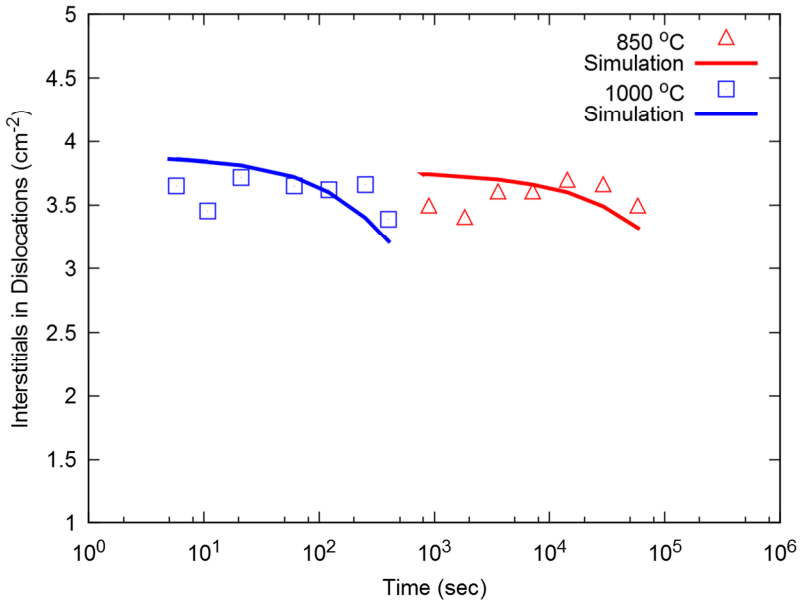

Figure 5.9. Interstitials bound to dislocation loops during single-step post-implant annealing [42].

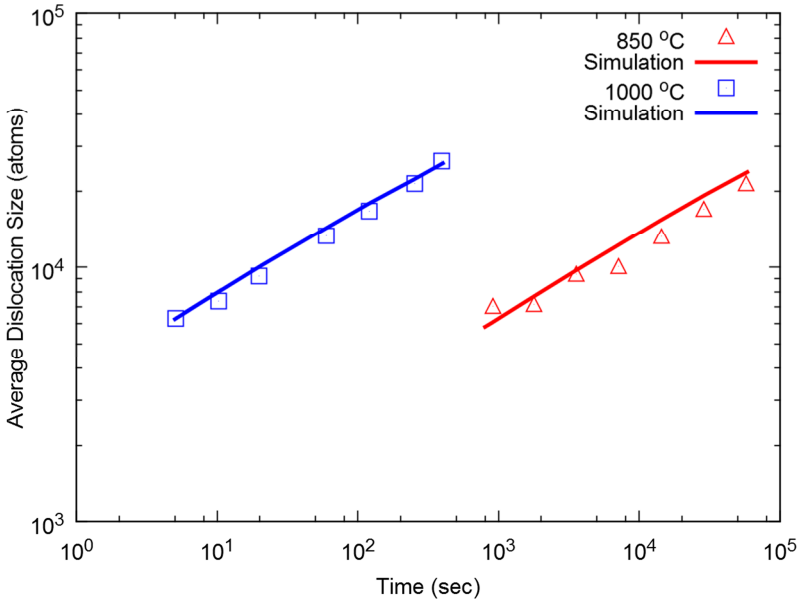

Figure 5.10. Growth and ripening of dislocation loops during annealing [42].

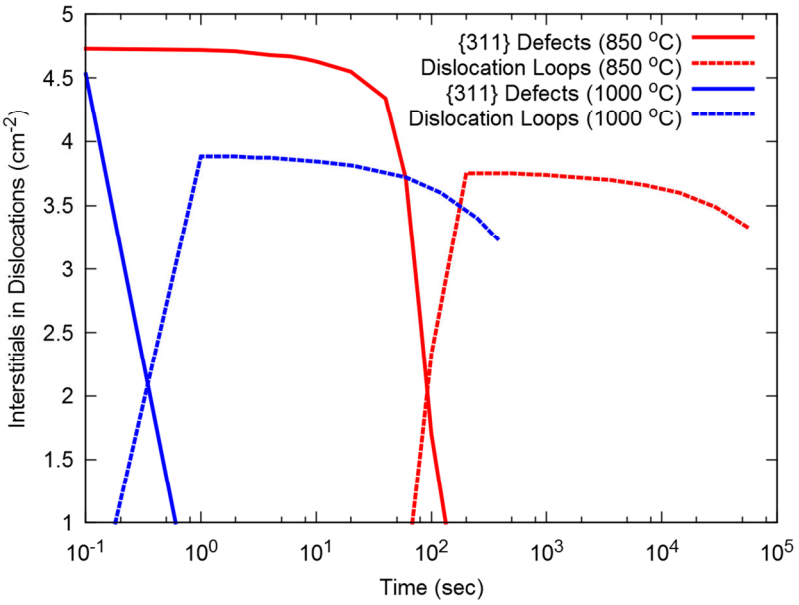

Figure 5.11. Evolution of extended defects during annealing [42].

The parameters of the transformation rate, T_0, and f_k were fitted separately at each temperature. The results are listed in Table 5.4.

Table 5.4. Parameters of the approximated transformation rate of Eq. (5.28) and the f_k predictor, Eq. (4.45) fitted to data from Ref. [42].

Parameter	Value at T = 850 °C	Value at T = 1000 °C
ω	5×10^{8} sec^{-1}	9×10^{11} sec^{-1}
n_{trans}	800	800
p_0	6×10^{6}	8×10^{4}
p_1	0	0
p	2	2

To allow the parameters to be extrapolated to other temperatures, they were fitted to an Arrhenius function for lack of a better understanding of their temperature dependence. The equations used in the model are

$$\omega(T) = 2.13\times10^{36}\exp\left(\frac{-6.16\ \text{eV}}{k_B T}\right)\ \text{sec}^{-1} \tag{5.41}$$

$$p_0(T) = 7.30 \times 10^{-10} \exp\left(\frac{3.55\ \text{eV}}{k_B T}\right) \tag{5.42}$$

The continued functioning of the {311} defect model was verified by comparing to an experiment by Eaglesham et al. [33] that mimics processing implants of the kind typically used for the lightly doped drain regions of MOSFET devices. Ion implant damage creates an interstitial supersaturation that quickly leads to {311} defect formation. These then dissolve (or transform into loops) at a rate that is strongly dependent on the annealing temperature. The agreement between simulation and experiment is excellent in this case, as shown by Fig. 5.12.

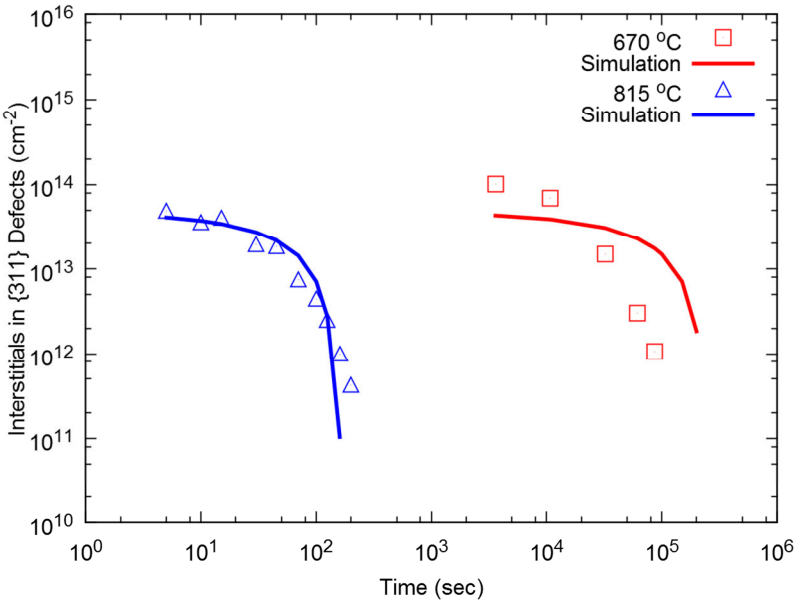

Figure 5.12. Interstitials bound to {311} defects versus annealing time in the experiment by Eaglesham et al. [33]

5.3.2 PREDICTED EFFECTS OF APPLIED STRESS

Due to the anisotropic induced strains of extended defects, applied strain breaks the degeneracy in formation energy of defects lying in differently-oriented habit planes. Depending on the nature of the applied stress, the overall formation of defects may be suppressed or enhanced.

To test the effect of applied stress, a biaxial strain of 1.5% parallel to the wafer surface was applied. Biaxial stress commonly occurs as a result of film deposition and lattice mismatch in heterojunctions. Fig. 5.13 shows the effect of biaxial strain on the evolution of {311} defects.

Compressive strain appears to inhibit their formation, and therefore increases dissolution, while tensile strain has the opposite effect. Average defect sizes are larger under compressive strain because of a smaller number of them are formed. Given that the induced strain of {311} defects (and dislocation loops) is compressive, this result is unsurprising.

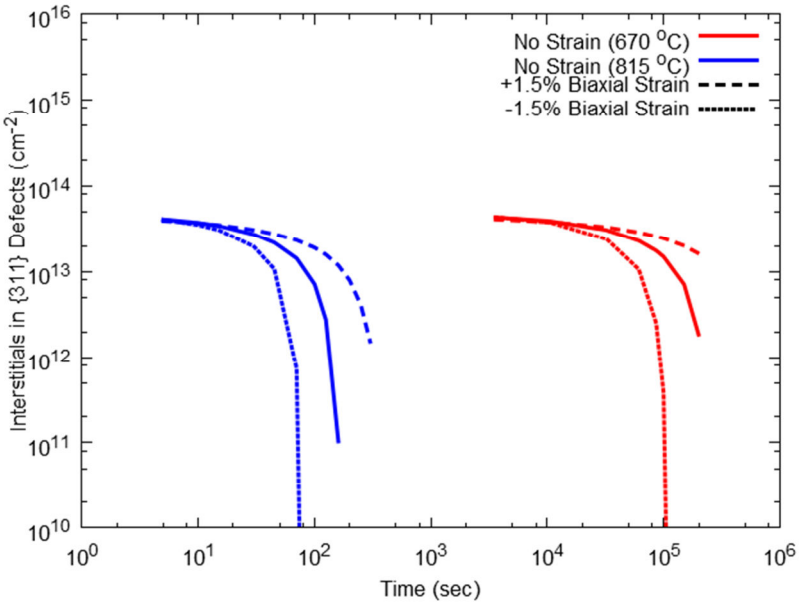

Figure 5.13. Effect of 1.5% biaxial strain on the formation and dissolution of {311} defects.

The effect on dislocations is shown in Figs. 5.14 and 5.15. Like {311} defects, dislocation loops have compressive induced strain and therefore are expected to behave similarly. Applied tensile stress enhances the formation of dislocation loops by reducing the work they must perform to create the dislocation strain field, thereby lowering their formation energy.

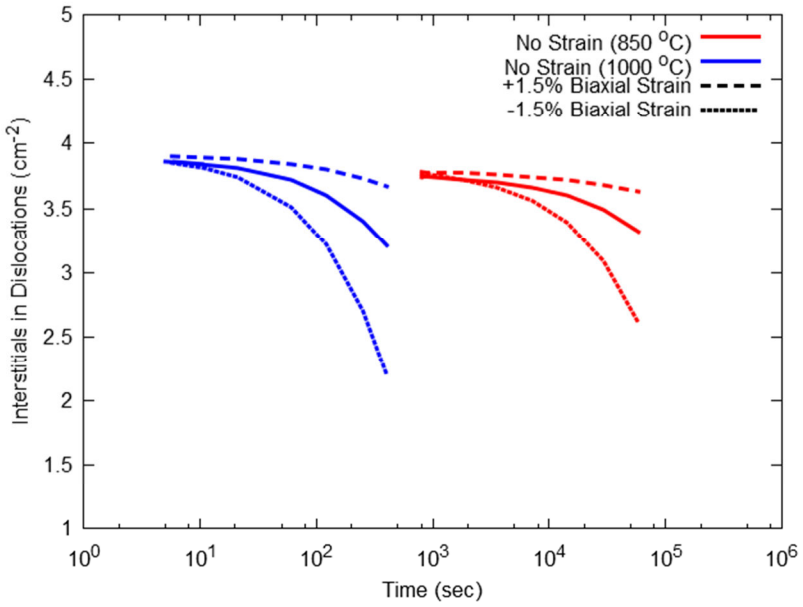

Figure 5.14. The effect of 1.5% biaxial strain on dislocation loops.

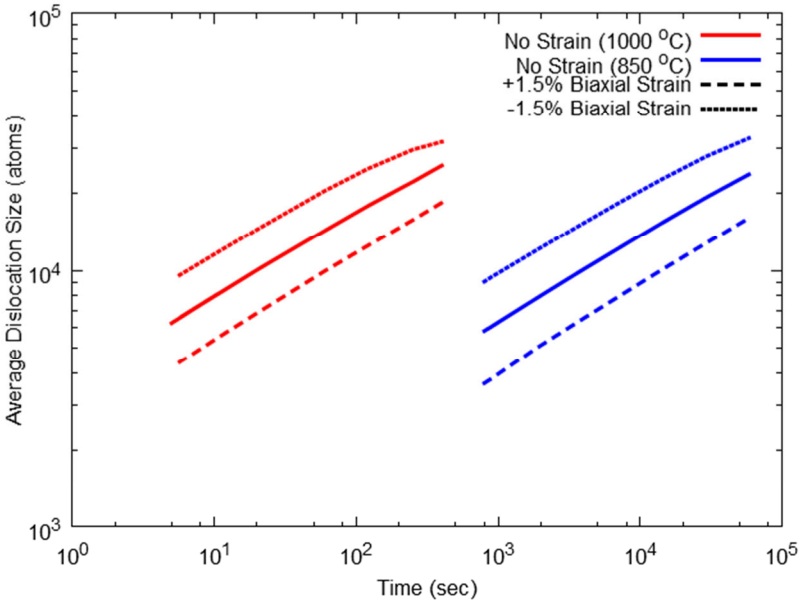

Figure 5.15. The effect of 1.5% biaxial strain on dislocation loop growth.

CHAPTER 6
CONCLUSION AND RECOMMENDATIONS

The objective set forth at the beginning of this dissertation was to develop accurate, robust, extensible, and usable models of oxygen precipitates and dislocation loops in silicon. These criteria have mostly been satisfied. The RKPM models developed here are certainly usable – they are fast, implemented with commonly-used commercial software tools, and require only a few physically meaningful initial conditions as fitting parameters. They are extensible – the dislocation models demonstrate how multiple precipitation models can easily be coupled and the treatment of small clusters in the oxygen model would easily allow, for example, BO_2 clusters to be modeled, if such a model is ever developed. They are robust – both RKPM models tend to converge easily over a wide range of input conditions and thermal conditions. The sensitivity analysis of the oxygen model demonstrated that the fitted initial conditions used to replicate experimental data are reasonable and that the model's response to perturbations in these conditions is sensible.

Although both models were able to successfully replicate experimental data, it would not be truthful to say that they are very accurate. There is much room for improvement and further research. However, a number of precise suggestions and recommendations to improve the accuracy of the oxygen model can be made:

- **Improve the *k*-1/*k* boundary condition.** A better estimator for f_k that depends on both forward-looking (i.e., moments) and backwards-looking (e.g., f_{k-1}, etc.) parameters is required. Alternatively, the flux across the boundary could be modeled in an entirely different manner and size k itself could perhaps be allowed to vary with simulation conditions.
- **Improve the dislocation loop model.** Investigations into the behavior of the dislocation model indicate that it may be providing too much positive feedback for the precipitate growth process. The handling of dislocation nucleation warrants further development.
- **Treat point defects independently.** Both defects must be considered to accurately model strain relief, precipitate morphology, and the widest possible range of real-world

processing conditions, including crystal growth itself. Having separate continuity equations for interstitials and vacancies, rather than only the net interstitial concentration, would be an improvement and would also allow the model to be coupled to other TCAD models.

- **Model precipitate morphology.** The assumption of spherical precipitate geometry is inadequate and may be causing nucleation rates to be underestimated. Modeling the precipitates as oblate spheroids would allow needle-like, platelet, and octahedral geometries to be approximated with a single functional form. It is suspected that these effects are presently hidden inside the temperature-dependent surface energy parameter.

The standalone RKPM dislocation loop model successfully demonstrates how dislocations and {311} defects can be treated with separate, simultaneous reaction pathways. A better model for the transformation rate and inclusion of heterogeneous dislocation loop nucleation at the end-of-range implant region would lead to a more generally applicable model of dislocation loops in silicon.

The primary novel result of this work is a reduced moment-based model of oxygen precipitation in silicon. With future refinements, this moment-based approach has great potential for producing oxygen models that are suitable for use in commercial simulation environments, helping to improve the yield and performance of ICs and silicon solar cells. Unlike previous works, the model source code is freely available and redistribution is encouraged in the hope that it will stimulate further development.

BIBLIOGRAPHY

[1] A. Borghesi, B. Pivac, A. Sassella and A. Stella, "Oxygen precipitation in silicon," *Journal of Applied Physics,* vol. 77, no. 9, pp. 4169-4244, 1995.

[2] I. Yonenaga, K. Sumino and K. Hoshi, "Mechanical strength of silicon crystals as a function of the oxygen concentration," *Journal of Applied Physics,* vol. 56, no. 8, pp. 2346-2350, 1984.

[3] F. Shimura, H. Tsuya and T. Kawamura, "Thermally Induced Defect Behavior and Effective Intrinsic Gettering Sink in Silicon Wafers," *Journal of The Electrochemical Society,* vol. 128, no. 7, pp. 1579-1583, 1981.

[4] L. Jastrzebski, R. Soydan, J. McGinn, R. Kleppinger, M. Blumenfeld, G. Gillespie, N. Armour, B. Goldsmith, W. Henry and S. Vecrumba, "A Comparison of Internal Gettering during Bipolar, CMOS, and CCD (High, Medium, Low Temperature) Processes," *Journal of The Electrochemical Society,* vol. 134, no. 4, pp. 1018-1025, 1987.

[5] J. R. Patel, K. A. Jackson and H. Reiss, "Oxygen precipitation and stacking-fault formation in dislocation-free silicon," *Journal of Applied Physics,* vol. 48, no. 12, pp. 5279-5288, 1977.

[6] W. Patrick, E. Hearn, W. Westdorp and A. Bohg, "Oxygen precipitation in silicon," *Journal of Applied Physics,* vol. 50, no. 11, p. 7156, 1979.

[7] F. A. Ponce, T. Yamashita and S. Hahn, "Structure of thermally induced microdefects in Czochralski silicon after high-temperature annealing," *Applied Physics Letters,* vol. 43, no. 11, p. 1051, 1983.

[8] T. Ono, G. A. Rozgonyi, C. Au, T. Messina, R. K. Goodall and H. R. Huff, "Oxygen Precipitation Behavior in 300 mm Polished Czochralski Silicon Wafers," *Journal of The Electrochemical Society,* vol. 146, no. 10, pp. 3807-3811, 1999.

[9] B. Leroy and C. Plougonven, "Warpage of Silicon Wafers," *Journal of The Electrochemical*

Society, vol. 127, no. 4, pp. 961-970, 1980.

[10] M.-H. Du, H. M. Branz, R. S. Crandall and S. B. Zhang, "Bistability-Mediated Carrier Recombination at Light-Induced Boron-Oxygen Complexes in Silicon," *Physical Review Letters,* vol. 97, no. 25, p. 256602, 2006.

[11] D. Hull and D. J. Bacon, Introduction to Dislocations, 4th ed., Oxford: Elsevier, 2001.

[12] M. A. Lourenço, M. S. A. Siddiqui, R. M. Gwilliam, G. Shao and K. P. Homewood, "Efficient silicon light emitting diodes made by dislocation engineering," *Physica E,* vol. 16, pp. 376-381, 2003.

[13] V. Kveder, M. Kittler and W. Schröter, "Recombination activity of contaminated dislocations in silicon: A model describing electron-beam-induced contrast behavior," *Physical Review B,* vol. 63, p. 115208, 2001.

[14] D. L. Dexter and F. Seitz, "Effects of Dislocations on Mobilities in Semiconductors," *Physical Review,* vol. 86, p. 964, 1952.

[15] J. Xia, T. Saito, R. Kim, T. Aoki, Y. Kamakura and K. Taniguchi, "Boron segregation to extended defects induced by self-ion implantation into silicon," *Journal of Applied Physics,* vol. 85, p. 7597, 1999.

[16] W. C. Dash, "Copper Precipitation on Dislocations in Silicon," *Journal of Applied Physics,* vol. 27, no. 10, pp. 1193-1195, 1956.

[17] B. Shen, T. Sekiguchi, J. Jablonski and K. Sumino, "Gettering of copper by bulk stacking faults and punched-out dislocations in Czochralski-grown silicon," *Journal of Applied Physics,* vol. 76, no. 8, pp. 4540-4546, 1994.

[18] J. Wong-Leung, E. Nygren and J. S. Williams, "Gettering of Au to dislocations and cavities in silicon," *Applied Physics Letters,* vol. 67, no. 3, pp. 416-418, 1995.

[19] M. Imaizumi, T. Ito, M. Yamaguchi and K. Kaneko, "Effect of grain size and dislocation

density on the performance of thin film polycrystalline silicon solar cells," *Journal of Applied Physics,* vol. 81, p. 7635, 81.

[20] M. L. Lee, E. A. Fitzgerald, M. T. Bulsara, M. T. Currie and A. Lochtefeld, "Strained Si, SiGe, and Ge channels for high-mobility metal-oxide-semiconductor field-effect transistors," *Journal of Applied Physics,* vol. 97, p. 011101, 2004.

[21] S. Senkader, *Physical Modeling and Simulation of Oxygen Precipitation in Silicon,* Vienna University of Technology, 1996.

[22] N. Inoue, J. Osaka and K. Wada, "Oxide Micro-Precipitates in As-Grown CZ Silicon," *Journal of The Electrochemical Society,* vol. 129, no. 12, pp. 2780-2788, 1982.

[23] K. Wada, H. Nakanishi, H. Takaoka and N. Inoue, "Nucleation temperature of large oxide precipitates in as-grown Czochralski silicon crystal," *Journal of Crystal Growth,* vol. 57, no. 3, pp. 535-540, 1982.

[24] H. Furuya, I. Suzuki, Y. Shimanuki and K. Murai, "Formation of Nuclei of Oxygen Precipitates in CZ Silicon Crystals during Crystal Growth Process," *Journal of The Electrochemical Society,* vol. 135, no. 3, pp. 677-682, 1988.

[25] A. Borghesi, B. Pivac and A. Sassella, "Oxygen precipitates in short-time annealed Czochralski silicon," *Journal of Crystal Growth,* vol. 126, no. 1, pp. 63-69, 1993.

[26] S. Kobayashi, "A model for oxygen precipitation in Czochralski silicon during crystal growth," *Journal of Crystal Growth,* vol. 174, pp. 163-169, 1997.

[27] K. Sueoka, M. Akatsuka, M. Okui and H. Katahama, "Computer Simulation for Morphology, Size, and Density of Oxide Precipitates in CZ Silicon," *Journal of The Electrochemical Society,* vol. 150, no. 8, pp. G469-G475, 2003.

[28] C. Wert and C. Zener, "Interstitial Atomic Diffusion Coefficients," *Physical Review,* vol. 76, no. 8, pp. 1169-1175, 1949.

[29] G. H. Vineyard, "Frequency factors and isotope effects in solid state rate processes," *Journal of Physics and Chemistry of Solids,* vol. 3, pp. 121-127, 1957.

[30] R. J. Needs, "First-principles calculations of self-interstitial defect structures and diffusion paths in silicon," *Journal of Physics: Condensed Matter,* vol. 11, no. 50, pp. 10437-10450, 1999.

[31] J. Kim, F. Kirchhoff, J. W. Wilkins and F. S. Khan, "Stability of Si-Interstitial Defects: From Point to Extended Defects," *Physical Review Letters,* vol. 84, p. 503, 2000.

[32] J. Kim, J. W. Wilkins, F. S. Khan and A. Canning, "Extended Si {311} defects," *Physical Review B,* vol. 55, no. 24, pp. 16186-16197, 1997.

[33] D. J. Eaglesham, P. A. Stolk, H.-J. Gossmann, T. E. Haynes and J. M. Poate, "Implant damage and transient enhanced diffusion in Si," *Nuclear Instruments and Methods in Physics Research B,* vol. 106, p. 191, 1995.

[34] G. Z. Pan and K. N. Tu, "Transmission electron miscroscopy on {113} rodlike defects and {111} dislocation loops in silicon-implanted silicon," *Journal of Applied Physics,* vol. 82, no. 2, pp. 601-608, 1997.

[35] I. Avci, M. E. Law, E. Kuryliw, A. F. Saavedra and K. S. Jones, "Modeling extended defect ({311} and dislocation) nucleation and evolution in silicon," *Journal of Applied Physics,* vol. 95, no. 5, pp. 2452-2460, 2004.

[36] A. H. Gencer and S. T. Dunham, "A combined model for {311} defect and dislocation loop evolution: Analytical formulation of kinetic precipitation model," *Journal of Applied Physics,* vol. 91, no. 5, pp. 2883-2889, 2002.

[37] J. P. Hirth and J. Lothe, Theory of Dislocations, New York: Wiley, 1982.

[38] J. Li and K. S. Jones, "{311} defects in silicon: The source of loops," *Applied Physics Letters,* vol. 73, no. 25, p. 3748, 1998.

[39] F. Cristiano, N. Cherkashin, X. Hebras, P. Calvo, Y. Lamrani, E. Scheid, B. de Mauduit, B. Colombeau, W. Lerch, S. Paul and A. Claverie, "Ion beam induced defects in crystalline silicon," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms,* vol. 216, pp. 46-56, 2004.

[40] P. Calvo, A. Claverie, N. Cherkashin, B. Colombeau, Y. Lamrani, B. de Mauduit and F. Cristiano, "Thermal evolution of {113} defects in silicon: transformation against dissolution," *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms,* vol. 216, pp. 173-177, 2004.

[41] S. Boninelli, N. Cherkashin, A. Claverie and F. Cristiano, "Evidences of an intermediate rodlike defect during the transformation of {113} defects into dislocation loops," *Applied Physics Letters,* vol. 89, no. 16, p. 161904, 2006.

[42] G. Z. Pan, K. N. Tu and A. Prussin, "Size-distribution and annealing behavior of end-of-range dislocation loops in silicon-implanted silicon," *Journal of Applied Physics,* vol. 81, no. 78, pp. 78-84, 1997.

[43] W. Kaiser, P. H. Keck and C. F. Lange, "Infrared Absorption and Oxygen Content in Silicon and Germanium," *Physical Review,* vol. 101, no. 4, pp. 1264-1268, 1956.

[44] J. W. Corbett, R. S. McDonald and G. D. Watkins, "The configuration and diffusion of isolated oxygen in silicon and germanium," *Journal of Physics and Chemistry of Solids,* vol. 25, no. 8, pp. 873-879, 1964.

[45] H. J. Hrostowski and R. H. Kaiser, "Infrared Absorption of Oxygen in Silicon," *Physical Review,* vol. 107, no. 4, pp. 966-972, 1957.

[46] ASTM International, *Test Method for Interstitial Atomic Oxygen Content of Silicon by Infrared Absorption,* West Conshohocken, PA, 1979.

[47] ASTM International, *Test Method for Interstitial Atomic Oxygen Content of Silicon by Infrared Absorption,* West Conshohocken, PA, 1983.

[48] A. Baghdadi, W. M. Bullis, M. C. Croarkin, Y.-Z. Li, R. I. Scace, R. W. Series, P. Stallhofer and M. Watanabe, "Interlaboratory Determination of the Calibration Factor for the Measurement of the Interstitial Oxygen Content of Silicon by Infrared Absorption," *Journal of The Electrochemical Society,* vol. 136, no. 7, pp. 2015-2024, 1989.

[49] S. Senkader, K. Jurkschat, D. Gambaro, R. J. Falster and P. R. Wilshaw, "On the locking of dislocations by oxygen in silicon," *Philosophical Magazine A,* vol. 81, no. 3, pp. 759-775, 2001.

[50] Semiconductor Equipment and Materials International, *Guide to Conversion Factors for Interstitial Oxygen in Silicon,* San Jose, CA, 2002.

[51] P. Wagner and J. Hage, "Thermal Double Donors in Silicon," *Applied Physics A,* vol. 49, no. 2, pp. 123-138, 1989.

[52] C. S. Fuller, J. A. Ditzenberger, N. B. Hannay and E. Buehler, "Resistivity changes in silicon single crystals induced by heat treatment.," *Acta Metallurgica,* vol. 3, no. 1, pp. 97-99, 1955.

[53] W. Kaiser, "Electrical and Optical Properties of Heat-Treated Silicon," *Physical Review,* vol. 105, no. 6, pp. 1751-1756, 1957.

[54] R. C. Newman, "Thermal donors in silicon: oxygen clusters or self-interstitial aggregates," *Journal of Physics C: Solid State Physics,* vol. 18, no. 30, pp. L967-L972, 1985.

[55] V. Cazcarra and P. Zunino, "Influence of oxygen on silicon resistivity," *Journal of Applied Physics,* vol. 51, no. 8, pp. 4206-4211, 1980.

[56] Y. J. Lee, J. von Boehm, M. Pesola and R. M. Nieminen, "Aggregation Kinetics of Thermal Double Donors in Silicon," *Physical Review Letters,* vol. 86, no. 14, pp. 3060-3063, 2001.

[57] H. J. Stein and J. W. Medernach, "Oxygen-related vibrational modes produced in Czochralski silicon by hydrogen plasma exposure," *Journal of Applied Physics,* vol. 79, no. 5,

pp. 2337-2342, 1996.

[58] L. I. Murin, T. Hallberg, V. P. Markevich and J. L. Lindström, "Experimental Evidence of the Oxygen Dimer in Silicon," *Physical Review Letters,* vol. 80, no. 1, pp. 93-96, 1998.

[59] J. Knobloch, S. W. Glunz, D. Biro, W. Warta, E. Schaffer and W. Wettling, "Solar cells with efficiencies above 21% processed from Czochralski grown silicon," in *Proceedings of the 25th IEEE Photovoltaic Specialists Conference*, New York, 1996.

[60] J. Adey, R. Jones and D. W. Palmer, "Degradation of Boron-Doped Czochralski-Grown Silicon Solar Cells," *Physical Review Letters,* vol. 93, no. 5, p. 055504, 2004.

[61] J. Schmidt and A. Cuevas, "Electronic properties of light-induced recombination centers in boron-doped Czochralski silicon," *Journal of Applied Physics,* vol. 86, no. 6, pp. 3175-3180, 1999.

[62] J. Schmidt, K. Bothe and R. Hezel, "Formation and annihilation of the metastable defect in boron-doped Czochralski silicon," in *Proceedings of the 29th IEEE Photovoltaic Specialists Conference*, New York, 2002.

[63] J. Schmidt and K. Bothe, "Structure and transformation of the metastable boron- and oxygen-related defect center in crystalline silicon," *Physical Review B,* vol. 69, no. 2, p. 024107, 2004.

[64] V. V. Voronkov and R. Falster, "Latent complexes of interstitial boron and oxygen dimers as a reason for degradation of silicon-based solar cells," *Journal of Applied Physics,* vol. 107, no. 5, p. 053509, 2010.

[65] J. C. Mikkelsen Jr., "The Diffusivity and Solubility of Oxygen in Silicon," in *MRS Proceedings*, 1985.

[66] J. Gass, H. H. Müller, H. Stüssi and S. Schweitzer, "Oxygen diffusion in silicon and the influence of different dopants," *Journal of Applied Physics,* vol. 51, no. 4, pp. 2030-2037,

1980.

[67] W. Wijaranakula, J. H. Matlock and H. Mollenkopf, "Oxygen diffusion in antimony-doped silicon," *Journal of Applied Physics,* vol. 53, no. 12, pp. 1068-1070, 1988.

[68] M. Pagani, "Secondary ion mass spectroscopy determination of oxygen diffusion coefficient in heavily Sb doped Si," *Journal of Applied Physics,* vol. 68, no. 7, pp. 3726-3728, 1990.

[69] S. K. Estreicher, "Interstitial O in Si and its interactions with H," *Physical Review B,* vol. 41, pp. 9886-9891, 1990.

[70] R. B. Capaz, L. V. C. Assali, L. C. Kimerling, K. Cho and J. D. Joannopoulos, "Mechanism for hydrogen-enhanced oxygen diffusion in silicon," *Physical Review B,* vol. 59, no. 7, pp. 4898-4900, 1999.

[71] R. C. Newman, J. H. Tucker and F. M. Livingston, "Radiation-enhanced diffusion of oxygen in silicon at room temperature," *Journal of Physics C: Solid State Physics,* vol. 16, pp. L151-L156, 1983.

[72] M. Ramamoorthy and S. T. Pantelides, "Coupled-Barrier Diffusion: The Case of Oxygen in Silicon," *Physical Review Letters,* vol. 76, no. 2, pp. 267-270, 1996.

[73] U. Gösele, K.-Y. Ahn, B. P. R. Marioton, T. Y. Tan and S.-T. Lee, "Do oxygen molecules contribute to oxygen diffusion and thermal donor formation in silicon?," *Applied Physics A,* vol. 48, no. 3, pp. 219-228, 1989.

[74] K. Wada, N. Inoue and K. Kohra, "Diffusion-limited growth of oxide precipitates in czochralski silicon," *Journal of Crystal Growth,* vol. 49, no. 4, pp. 749-752, 1980.

[75] J. Ōsaka, N. Inoue and K. Wada, "Homogeneous nucleation of oxide precipitates in Czochralski-grown silicon," *Applied Physics Letters,* vol. 36, no. 4, pp. 288-290, 1980.

[76] N. I. Puzanov and A. M. Eidenzon, "The effect of thermal history during crystal growth

on oxygen precipitation in Czochralski-grown silicon," *Semiconductor Science and Technology,* vol. 7, no. 3, pp. 406-413, 1992.

[77] K. Sueoka, N. Ikeda and T. Yamamoto, "Morphology and size distribution of oxide precipitates in as-grown Czochralski silicon crystals," *Applied Physics Letters,* vol. 65, no. 13, pp. 1686-1688, 1994.

[78] W. A. Tiller, S. Hahn and F. A. Ponce, "Thermodynamic and kinetic considerations on the equilibrium shape for thermally induced microdefects in Czochralski silicon," *Journal of Applied Physics,* vol. 59, no. 9, pp. 3255-3266, 1986.

[79] W. L. Bond and W. Kaiser, "Interstitial versus substitutional oxygen in silicon," *Journal of Physics and Chemistry of Solids,* vol. 16, pp. 44-45, 1960.

[80] J. Vanhellemont and C. Claeys, "A theoretical study of the critical radius of precipitates and its application to silicon oxide in silicon," *Journal of Applied Physics,* vol. 62, no. 9, pp. 3960-3967, 1987.

[81] H. Bender, "Investigation of the oxygen-related lattice defects in Czochralski silicon by means of electron microscopy techniques," *Physica Status Solidi A,* vol. 86, pp. 245-261, 1984.

[82] Z. Xi, D. Yang, J. Chen, J. Xu, Y. Ji, D. Que and H. J. Moeller, "Influence of copper precipitation on oxygen precipitation in Czochralski silicon," *Semiconductor Science and Technology,* vol. 19, pp. 299-305, 2004.

[83] H.-D. Chiou and L. W. Shive, "Test Methods for Oxygen Precipitation in Silicon," *VLSI Science and Technology/1985,* pp. 429-435, 1985.

[84] H.-D. Chiou, "Oxygen Precipitation Behavior and Control in Silicon Crystals," *Solid State Technology,* vol. 30, no. 3, pp. 77-81, 1987.

[85] G. A. Hawkins and J. P. Lavine, "The effect of rapid thermal annealing on the precipitation of oxygen in silicon," *Journal of Applied Physics,* vol. 65, no. 9, pp. 3644-3654,

1989.

[86] K. Wada and N. Inoue, "Growth kinetics of oxygen precipitates in Czochralski silicon," *Journal of Crystal Growth,* vol. 71, no. 1, pp. 111-117, 1985.

[87] M. Schrems, T. Brabec, M. Budil, H. Pötzl, E. Guerrero, D. Huber and P. Pongratz, "Simulation of Oxygen Precipitation in Czochralski Grown Silicon," *Materials Science and Engineering, B4,* pp. 393-399, 1989.

[88] M. Schrems, P. Pongratz, M. Budil, H. W. Pötzl, J. Hage, E. Guerrero and D. Huber, "Computer simulation of oxygen precipitation in CZ-silicon during rapid thermal anneals," in *Proceedings of the 20th European Solid State Device Research Conference*, Nottingham, 1990.

[89] M. Schrems, M. Budil, G. Hobler and H. Pötzl, "Oxygen Precipitation in CZ-Silicon During Multi-Step Annealing Cycles," in *Simulation of Semiconductor Devices and Processes Vol. 4*, Zurich, 1991.

[90] S. Senkader, J. Esfandyari and G. Hobler, "A model for oxygen precipitation in silicon including bulk stacking fault growth," *Journal of Applied Physics,* vol. 78, no. 11, pp. 6469-6476, 1995.

[91] B. G. Ko and K. D. Kwack, "Growth/dissolution model for oxygen precipitation for oxygen based on the kinetics of phase transformations," *Journal of Applied Physics,* vol. 85, no. 4, pp. 2100-2107, 1998.

[92] S. Kobayashi, "Mathematical modeling of grown-in defects formation in Czochralski silicon," *Journal of Crystal Growth,* vol. 180, pp. 334-342, 1997.

[93] M. B. Shabani, T. Yoshimi and H. Abe, "Low-Temperature Out-Diffusion of Cu from Silicon Wafers," *Journal of The Electrochemical Society,* vol. 143, no. 6, pp. 2025-2029, 1996.

[94] A. A. Istratov and E. R. Weber, "Physics of Copper in Silicon," *Journal of The Electrochemical Society,* vol. 149, no. 1, pp. G21-G30, 2002.

[95] R. Becker and W. Döring, "Kinetische Behandlung der Keimbildung in übersättigten Dämpfen," *Annalen der Physik,* vol. 24, no. 5, pp. 719-752, 1935.

[96] M. Volmer, Kinetik der Phasenbildung, Dresden: Theodor Steinkopff, 1939.

[97] Y. B. Zeldovich, "On the theory of new phase formation, cavitation," *Acta Physicochimica URSS,* vol. 18, pp. 1-22, 1943.

[98] J. Frenkel, Kinetic Theory of Liquids, New York: Oxford, 1946.

[99] J. W. Christian, The Theory of Transformations in Metals and Alloys (Part I), 3rd ed., Oxford: Pergamon, 2002.

[100] S. T. Dunham, "Modeling of the Kinetics of Dopant Precipitation in Silicon," *Journal of The Electrochemical Society,* vol. 142, p. 2823, 1995.

[101] A. Haarahiltunen, H. Väinölä, O. Anttila, E. Saarnilehto, M. Yli-Koski, Storgårds and J. Sinkkonen, "Modeling of heterogeneous precipitation of iron in silicon," *Applied Physics Letters,* vol. 87, p. 151908, 2005.

[102] F. C. Goodrich, "Nucleation Rates and the Kinetics of Particle Growth. II. The Birth and Death Process," *Proceedings of the Royal Society A,* vol. 277, pp. 167-182, 1964.

[103] J. S. Chang and G. Cooper, "A Practical Difference Scheme for Fokker-Planck Equations," *Journal of Computational Physics,* vol. 6, no. 1, pp. 1-16, 1970.

[104] H. O. Dokumaci, *Analysis and Modeling of Arsenic Activation and Deactivation in Silicon,* University of Florida, 1997.

[105] A. Haarahiltunen, H. Väinölä, O. Anttila, M. Yli-Koski and J. Sinkkonen, "Experimental and theoretical study of heterogeneous iron precipitation in silicon," *Journal of Applied Physics,* vol. 101, no. 4, p. 043507, 2007.

[106] I. Clejan and S. T. Dunham, "A reduced moment-based model for precipitation kinetics

and application to dopant activation in silicon," *Journal of Applied Physics,* vol. 78, no. 12, pp. 7327-7333, 1995.

[107] A. H. Gencer and S. T. Dunham, "A predictive model for transient enhanced diffusion based on evolution of {311} defects," *Journal of Applied Physics,* vol. 81, no. 2, pp. 631-636, 1996.

[108] H.-W. Guo, S. T. Dunham, C.-L. Shih and C. Ahn, "Modeling of Defect Evolution and TED under Stress based on DFT Calculations," in *SISPAD 2006*, Monterey, 2006.

[109] W. J. Taylor, U. M. Gösele and T. Y. Tan, "Precipitate strain relief via point defect interaction: models for SiO2 in silicon," *Materials Chemistry and Physics,* vol. 34, pp. 166-174, 1993.

[110] G. Kresse and J. Furthmüller, "Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set," *Physical Review B,* vol. 54, no. 16, pp. 169-186, 1996.

[111] S. T. Dunham, "Growth kinetics of disk-shaped extended defects with constant thickness," *Applied Physics Letters,* vol. 63, pp. 464-466, 1993.

[112] R. Swaroop, N. Kim, W. Lin, M. Bullis, L. Shive, A. Rice, E. Castel and M. Christ, "Testing for Oxygen Precipitation in Silicon Wafers," *Solid State Technology,* pp. 85-89, March 1987.

[113] H. Abe, I. Suzuki and H. Koya, "The Effect of Hydrogen Annealing on Oxygen Precipitation Behavior and Gate Oxide Integrity in Czochralski Si Wafers," *Journal of The Electrochemical Society,* p. 1, 144 306-311 1997.

[114] H. W. Kennel, *Physical Modeling of Oxygen Precipitation, Defect Formation, and Diffusion in Silicon,* Stanford University, 1991.

[115] R. J. Stewart, S. Messoloras and S. Rycroft, "A small angle neutron scattering study of oxygen precipitation in silicon," in *Proceedings of the NATO Advanced Research Workshop*,

Exeter, UK, 1996.

[116] S. Isomae, "Computer-aided simulation for oxygen precipitation in silicon," *Journal of Applied Physics,* vol. 70, no. 8, pp. 4217-4223, 1991.

[117] B.-G. Ko, *Modeling and Analysis of Oxygen Precipitation in CZ Silicon,* Hanyang University, 1999.

[118] H. Bracht, N. A. Stolwijk and H. Mehrer, "Properties of intrinsic point defects in silicon determined by zinc diffusion experiments under nonequilibrium conditions," *Physical Review B,* vol. 52, no. 23, p. 16542, 1995.

[119] H. Bracht, H. H. Silvestri, I. D. Sharp and E. E. Haller, "Self- and foreign-atom diffusion in semiconductor isotope heterostructures. II. Experimental results for silicon," *Physical Review B,* vol. 75, no. 3, p. 035211, 2007.

[120] S. K. Ghandhi, VLSI Fabrication Principles: Silicon and Gallium Arsenide, New York: John Wiley & Sons, 1983.

[121] R. Hull, Properties of crystalline silicon, vol. 20, Institute of Engineering and Technology, 1999.

[122] A. Hössinger, *Simulation of Ion Implantation for ULSI Technology,* Vienna University of Technology, 2000.

[123] B. El-Kareh, Fundamentals of Semiconductor Processing Technologies, Boston: Kluwer, 1995.

[124] A. Polian, V.-T. Dung and P. Richet, "Elastic properties of a-SiO2 up to 2300 K from Brillouin scattering measurements," *Europhysics Letters,* vol. 57, no. 3, pp. 375-381, 2002.

[125] F. Liu, M. Mostoller, V. Milman, M. F. Chisholm and T. Kaplan, "Electronic and elastic properties of edge dislocations in Si," *Physical Review B,* vol. 51, no. 23, p. 17192, 1995.

[126] F. S. Karoui, A. Karoui and G. A. Rozgonyi, "Simulation Of Point Defect Clustering In Cz-Silicon Wafers On The Cray T3E Scalable Parallel Computer: Application To Oxygen Precipitation," in *Technical Proceedings of the 2000 International Conference on Modeling and Simulation of Microsystems*, San Diego, 2000.

[127] H.-W. Guo, *Diffusion and Precipitation Models for Silicon Gettering and Ultra Shallow Junction Formation,* University of Washington, 2008.

[128] S. T. Dunham, P. Fastenko, Z. Qin and M. Diebel, "The Process Modeling Hierarchy: Connecting Atomistic Calculations to Nanoscale Behavior," *IEICE Transactions on Electronics,* Vols. E86-C, p. 276, 2003.

[129] M. Diebel, *Application of Ab-initio Calculations to Modeling of Nanoscale Diffusion and Activation in Silicon,* University of Washington, 2004.

[130] P. Hohenberg and W. Kohn, "Inhomogeneous Electron Gas," *Physical Review,* vol. 136, no. 3B, pp. B864-B871, 1964.

[131] W. Kohn and L. J. Sham, "Self-Consistent Equations Including Exchange and Correlation Effects," *Physical Review,* vol. 140, no. 4A, pp. A1133-A1138, 1965.

[132] J. P. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh and C. Fiolhais, "Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation," *Physical Review B,* vol. 46, p. 6671, 1992.

[133] J. P. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh and C. Fiolhais, "Erratum: Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation," *Physical Review B,* vol. 48, p. 4978, 1993.

[134] M. Y. Chou, M. L. Cohen and S. G. Louie, "Theoretical study of stacking faults in silicon," *Physical Review B,* vol. 32, p. 7979, 1985.

[135] W. Cai, V. V. Bulatov, J. Chang, J. Li and S. Yip, "Periodic image effects in dislocation modeling," *Philosophical Magazine A,* vol. 83, p. 539, 2003.

[136] W. Cai, *Atomistic and Mesoscale Modeling of Dislocation Mobility,* Massachusetts Institute of Technology, 2001.

APPENDIX A
ZERO-DIMENSIONAL POINT DEFECT BOUNDARY CONDITION

Growing oxygen precipitates eject a large number of interstitials, creating local supersaturations that inhibit further growth. Interstitials are removed by the generation of dislocation loops and diffusion to the wafer surfaces. It is important to consider both mechanisms. To model the effect of the wafer surfaces on point defect concentrations in 0D (single spatial point) simulations, the following assumptions are made:

- Point defects are generated throughout the wafer at a constant rate, G, caused by the oxygen precipitation process.
- The system is in steady state.
- Point defects recombine quickly at the surfaces so that $C = C^*$, where C is C_I for interstitials and C_V for vacancies, and C^* is the appropriate thermal equilibrium concentration.
- The wafer thickness is L.

The steady state point defect concentration in 1D will be derived first and the result will be used to construct a 0D expression that removes point defects from the system at the same rate.

The continuity equation in 1D is

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} + G \tag{A.1}$$

where D is the diffusivity. In steady state,

$$0 = D\frac{\partial^2 C}{\partial x^2} + G \tag{A.2}$$

The solution is

$$C(x) = -\frac{1}{2}\frac{G}{D}x^2 + K_1 x + K_2 \tag{A.3}$$

The boundary conditions are

$$\begin{aligned} C(0) &= C^* \\ C(L) &= C^* \end{aligned} \tag{A.4}$$

From these, K_1 and K_2 are determined to be

$$\begin{aligned} K_1 &= \frac{GL}{2D} \\ K_2 &= C^* \end{aligned} \tag{A.5}$$

Therefore, the steady state point defect distribution in the wafer in the 1D case is

$$C_{1D}(x) = -\frac{1}{2}\frac{G}{D}x^2 + \frac{GL}{2D}x + C^* \tag{A.6}$$

The 0D form of the point defect continuity equation is

$$\frac{dC}{dt} = G - k \cdot \left(C - C^*\right) \tag{A.7}$$

A reaction term has been added with a reaction constant, *k*. In the 1D case, this would be expressed in terms of boundary conditions describing the flux at each surface. The steady state solution is

$$C_{0D} = \frac{G}{k} + C^* \tag{A.8}$$

The objective is to determine the reaction constant, *k*, that causes the same amount of material to be removed from the wafer as in the 1D case. To do this, the solutions are integrated over the entire wafer. The 0D case is integrated by assuming that the solution is uniform at all positions.

$$\int_0^L C_{1D}(x) \cdot dx = \int_0^L C_{0D}(x) \cdot dx \tag{A.9}$$

Integrating each side results in an equation that can be solved for *k*.

$$-\frac{1}{6}\frac{G}{D}L^3 + \frac{1}{4}\frac{G}{D}L^3 + C^*L = \frac{G}{k}L + C^*L \tag{A.10}$$

The result is

$$k = \frac{12D}{L^2} \tag{A.11}$$

Using this value of k with Eqs. (A.7) and (4.9) allows the rate term, $R_I^{surface}$, of Eq. (4.47) to be obtained.

APPENDIX B
OXYGEN MODEL SOURCE CODE

The MATLAB source code for the oxygen model (`OModel.m`) is listed on the following pages. Also included is a script that simulates all the experiments discussed in Section 4.2.2 (`OxygenExperiments.m`).

```
%%
%% ===================================
%%  OModel.m
%%  by Bart Trzynadlowski, 2011-2013
%% ===================================
%%
%% Model for oxygen precipitation in silicon. Includes both the full and reduced
%% kinetic precipitation models described in my doctoral dissertation ("Reduced
%% Moment-Based Models for Oxygen Precipitates and Dislocation Loops in
%% Silicon," University of Washington, 2013).
%%
%% For usage instructions, please read the description of the OModel() function,
%% below. Further questions may be directed to me, Bart Trzynadlowski
%% (bart.trzy@gmail.com), or Prof. Scott Dunham (dunham@ee.washington.edu).
%%
%%
%% Acknowledgments
%% ---------------
%% This model was developed at the Nanotechnology Modeling Laboratory
%% (University of Washington, Dept. of Electrical Engineering) under the
%% supervision of Prof. Scott Dunham. VASP calculations of small oxygen clusters
%% and point defects were performed by Renyu Chen. Funding was generously
%% provided by SiWEDS member companies (namely Texas Instruments Inc.), Sony
%% Corp., and the Semiconductor Research Corp.
%%
%%
%% Version History
%% ---------------
%%
%% 2013.03.20:
%%  - Initial release included in doctoral dissertation.
%%

%
% [n, t, temp, CO, m0, m1, ncrit, SI, m0DL, m1DL, ncritDL, f, idxK, r] =
%   OModel(model, r, CO, m0, nAvg, netCI, waferThickness, 'steps',
%                                                          steps)
%   OModel(model, r, CO, m0, nAvg, netCI, waferThickness, 'steps',
%                                                          steps, params)
%   OModel(model, r, CO, m0, nAvg, netCI, waferThickness, 'points',
%                                                          temps, times)
%   OModel(model, r, CO, m0, nAvg, netCI, waferThickness, 'points',
%                                                          temps, times, params)
%
%   Runs the oxygen model. Outputs are optional; none of them, all of them, or
%   any number in between may be specified but their order cannot be changed.
%   Please pay careful attention to the use of units for input parameters, which
%   is not always consistent.
%
%   The default model parameters were fitted to experiments conducted with FTIR
%   spectroscopy. All interstitial oxygen concentrations were interpreted using
%   the "new" (1983) ASTM calibration factor (ASTM F121-83): 4.90 ppma-cm or
%   2.45e17 cm^-2. Oxygen concentrations should be normalized to this standard
%   for best results.
%
%
%   Inputs:
%
%       model           Selects the model type: 'rkpm' for the reduced model and
%                       'fkpm' for the full model. If you are unsure, use
%                       'rkpm'. The full model is not necessarily more accurate
%                       and some initial conditions (m0 and nAvg) cannot be
%                       represented in it, resulting in undefined behavior for
%                       anything other than values near 0 for those quantities.
%
%       r               Rediscretization factor. This is called the "sample
%                       discretization factor", S, in the dissertation. Controls
%                       the spacing between samples in size space. A value of 1
%                       corresponds to unit spacing, which in the full model
%                       generates an impossibly large number of equations but is
%                       valid in the RKPM model. Default physical parameters
```

```
%                       were fitted assuming a value of 1.1, which is
%                       recommended. Values in the range of 1.05-1.3 are most
%                       reasonable.
%
%                       The value passed will be adjusted to ensure that the
%                       sample point corresponding to size k = 72. Sometimes
%                       this fails in which case a slightly different r should
%                       be tried.
%
%       CO              Initial interstitial oxygen concentration [cm^-3].
%
%       m0              In the RKPM model, the initial concentration of oxygen
%                       precipitates [cm^-3]. Avoid values less than 1.
%
%       nAvg            In the RKPM model, the average size of initial oxygen
%                       precipitates [oxygen atoms]. Avoid values less than 73.
%
%       netCI           Initial point defect concentrations specified as the net
%                       silicon interstitial concentration: CI - CV [cm^-3].
%                       This is the only quantity that may be negative.
%
%       waferThickness  Wafer thickness [um]. The model is zero-dimensional and
%                       solves for a single spatial sample point that can be
%                       interpreted as either existing in the bulk or as the
%                       average value over the entire wafer. Oxygen out-
%                       diffusion is assumed negligible and not simulated but
%                       the diffusion of point defects to the wafer surface and
%                       recombination there is approximated. If unknown,
%                       reasonable values are 400 or 500 [um].
%
%       steps           When the temperature schedule format is 'steps', a
%                       vector of process steps and ramp rates defines the
%                       schedule. Its format is:
%
%                           [temp1, time1]
%                           [temp1, time1, ramp12, temp2, time2]
%                           [temp1, time1, ramp12, temp2, time2, ramp23, temp3,
%                            time3, ...]
%
%                       At least a single temperature [C] and time [hr] must be
%                       specified. Subsequent steps can be added in triplets
%                       consisting of the ramp rate [C/min] from the previous
%                       step, temperature, and time.
%
%       temps           When the temperature schedule format is 'points', a
%                       sequence of temperatures and corresponding absolute
%                       times define the schedule. The temperature vector is
%                       specified in units of [C]. Temperatures are interpolated
%                       linearly between adjacent points.
%
%       times           When the temperature schedule format is 'points', this
%                       defines the absolute time [sec] of each temperature.
%                       The temperature and time vectors must be of equal
%                       lengths. The first time must be 0, subsequent times must
%                       appear in ascending order, and a minimum of two points
%                       are required. Duplicate time values (i.e., zero-length
%                       steps) are not allowed.
%
%       params          Allows the physical parameters to be specified as a
%                       vector:
%
%                           [Alpha750, Alpha1050, Css750, Css1050]
%
%                       Alpha750 is the precipitate surface energy [J/m^2] at
%                       T = 750 C and Alpha1050 is the value at T = 1050 C. The
%                       two points define a linear relationship between the
%                       surface energy and temperature. Similarly, Css750 and
%                       Css1050 are the solubility concentrations [cm^-3] of
%                       oxygen in silicon at T = 750 C and 1050 C, respectively.
%                       They are used to construct an Arrhenius function:
%
```

```
%                           Css = Css0 * exp(-Ea / kBT)
%
%                       where Css0 is the prefactor and Ea is the activation
%                       energy.
%
%                       The default parameters are the best-fit values from my
%                       dissertation:
%
%                           Alpha750  = 0.1915 [J/m^2]
%                           Alpha1050 = 0.2565 [J/m^2]
%                           Css750    = 4.85551e15 [cm^-3]
%                           Css1050   = 2.2888e17 [cm^-3]
%
%
%   Outputs:
%
%       n               Sizes of each solution term. Values of 0 at the front of
%                       the vector correspond to non-precipitate solution terms
%                       (e.g., point defects, dislocation loops). CO is size 1.
%
%       t               Time points at which solutions are available [sec].
%
%       temp            Temperature profile [C].
%
%       CO              Interstitial oxygen concentration [cm^-3].
%
%       m0              Concentration of all oxygen precipitates of size 72 or
%                       larger [cm^-3]. The RKPM model solves this directly. The
%                       FKPM model computes this as a post-processing step by
%                       integrating the precipitate size distribution at each
%                       available time step.
%
%       m1              Concentration of oxygen atoms in all precipitates of
%                       size 72 or larger [cm^-3]. Solved directly by the RKPM
%                       model and integrated during post-processing by the FKPM
%                       model.
%
%       ncrit           Critical size of oxygen precipitates [oxygen atoms].
%                       This is a valuable diagnostic aid. Precipitates larger
%                       than this will tend to grow while smaller ones will
%                       shrink. Negative values are equivalent to an infinitely
%                       large critical size (all precipitates dissolve).
%
%       SI              Silicon interstitial supersaturation: CI/CI*, where CI*
%                       is the thermal equilibrium concentration of
%                       interstitials at the current temperature. Because fast
%                       recombination is assumed, CV/CV* = 1/SI.
%
%       m0DL            Concentration of all dislocation loops [cm^-3]. Faulted
%                       dislocation loops are nucleated at the surface of oxygen
%                       precipitates due to high interstitial supersaturations
%                       and local precipitate-induced strain.
%
%       m1DL            Concentration of silicon interstitials in all
%                       dislocation loops [cm^-3].
%
%       ncritDL         Critical size of dislocation loops [interstitial atoms].
%
%       f               Complete solution vector [cm^-3]. Non-zero values of
%                       n(i) are the size, in oxygen atoms, of the precipitate
%                       given by f(:,i).
%
%       idxK            Index of size k (72) in the solution vector.  f(1,idxK)
%                       would therefore be f(k) (using my dissertation's
%                       notation) at the very first time step.
%
%       r               Actual rediscretization factor used in the simulation.
%                       This is not in general the same as the value specified
%                       but is instead adjusted to ensure k = 72.
%
%
```

```
%   Examples:
%
%       1. A single-step process, 750 C for 10 hr.
%
%           OModel('rkpm', 1.1, 9e17, 1, 73, -4e12, 400, 'steps', [750 10])
%
%       2. As above but specified with 'points'.
%
%           OModel('rkpm', 1.1, 9e17, 1, 73, -4e12, 400, 'points', ...
%                  [750 750], [0 10*3600])
%
%       3. A two-step process, 800 C for 4 hr, 1050 C for 16 hr, with a 10 C/min
%          ramp rate between steps.
%
%           OModel('rkpm', 1.1, 9e17, 1e5, 1e4, -4e12, 400, 'steps', ...
%                  [800 4 10 1050 16])
%
%       4. As above but specified with 'points'.
%
%           OModel('rkpm', 1.1, 9e17, 1e5, 1e4, -4e12, 400, 'points', ...
%                  [800 800 1050 1050],                                ...
%                  [0, 4*3600, 4*3600+1500, 4*3600+1500+16*3600])
%
%       5. Obtaining the first 4 output vectors.
%
%           [n, t, temp, CO] = OModel(...);
%
%       6. Obtaining the first 7 output vectors.
%
%           [n, t, temp, CO, m0, m1, ncrit] = OModel(...);
%
function varargout = OModel(model, r, CO, m0, nAvg, netCI, waferThickness, format, varargin)
    % Default parameters (Alpha750, Alpha1050, Css750, Css1050)
    parameters = [ 0.1915 0.2565 4.85551e15 2.2888e17 ];

    % Clear outputs so MATLAB doesn't complain if we abort early
    varargout = cell(1, nargout);

    %
    % Validate model type
    %
    global modelType;   % 'f' for full model, 'r' for RKPM

    if (model(1) == 'f') || (model(1) == 'F')
        modelType = 'f';
    elseif (model(1) == 'r') || (model(1) == 'R')
        modelType = 'r';
    else
        fprintf('Error: Invalid model selection. Use either ''full'' or ''rkpm''.\n');
        return;
    end

    %
    % Validate schedule format, fetch parameters, and construct temperature
    % schedule. The temperature schedule is described by two vectors,
    % constructed here:
    %
    % 1. tempProfile: a vector of temperatures [C]
    % 2. timeProfile: a vector of corresponding times [sec] to interpolate
    %    temperatures between.
    %
    if strcmpi(format, 'steps')
        % Get thermal steps
        if nargin < 9
            fprintf('Error: Step vector argument is missing.\n');
            return;
        end

        steps = varargin{1};
        if length(steps) < 2
            fprintf(['Error: Step vector must contain at least two elements: ', ...
```

```
                'temperature [C], time [hr].\n']);
        return;
    end

    % Get optional model parameters
    if nargin == 10
        parameters = varargin{2};
    elseif nargin > 10
        fprintf('Error: Too many arguments for ''steps'' format.\n');
        return;
    end

    %
    % The step vector specifies the process as a series of temperature
    % steps consisting of a temperature and time, connected with a ramp
    % rate. The format is:
    %
    %       temp1, time1, [ramp1, temp2, time2, ...]
    %
    % where [] indicates optional elements. Temperatures are in [C],
    % times in [hr], and ramp rates in [C/min]. Each additional step
    % requires a ramp rate, temperature, and time. The ramp rate is
    % used to raise/lower the temperature linearly over time until the
    % next temperature is reached. Then, that temperature will be held
    % constant for the specified amount of time.
    %

    if length(steps) < 2
        fprintf(['Error: The minimum step vector size is 2 elements: ', ...
                 'temperature [C], time [hr].\n']);
        return;
    end

    % Begin building temperature profile
    tempProfile = [ steps(1) steps(1) ];    % temperature 1
    timeProfile = [ 0 steps(2) ] * 3600;    % from t = 0 to time1

    % Remaining elements must be triplets of: ramp rate, temp, time
    num = length(steps) - 2;    % elements remaining in vector
    idx = 3;
    if mod(num, 3) ~= 0
        fprintf(['Error: Additional temperature steps require 3 elements: ', ...
                 'ramp rate [C/min], temperature [C], time [hr].\n']);
        return;
    end

    % Construct the rest of the temperature profile
    for i = 1:(num/3)
        rampRate = steps(idx);
        temp = steps(idx+1);
        time = steps(idx+2);
        idx = idx + 3;

        % Ramp up to next temperature
        rampTime = abs(60*((temp-tempProfile(end))/rampRate));  % [sec]
        tempProfile(end+1) = temp;
        timeProfile(end+1) = timeProfile(end) + rampTime;

        % Hold for specified time
        tempProfile(end+1) = temp;
        timeProfile(end+1) = timeProfile(end) + time*3600;
    end
elseif strcmpi(format, 'points')
    %
    % This mode takes a temperature list and a corresponding time list
    % (in seconds) to define the temperature schedule.
    %

    % Get temperatures
    if nargin < 9
        fprintf('Error: Temperature vector argument is missing.\n');
```

```
        return;
    end
    tempProfile = varargin{1};
    if length(tempProfile) < 2
        fprintf('Error: Temperature vector must specify at least two temperatures.\n');
        return;
    end

    % Get times
    if nargin < 10
        fprintf('Error: Time vector argument is missing.\n');
        return;
    end
    timeProfile = varargin{2};
    if length(timeProfile) ~= length(tempProfile)
        fprintf('Error: Temperature and time vectors must have equal lengths.\n');
        return;
    end
    if timeProfile(1) ~= 0
        fprintf('Error: First time must be 0.\n');
        return;
    end

    % Get optional model parameters
    if nargin == 11
        parameters = varargin{3};
    elseif nargin > 11
        fprintf('Error: Too many arguments for ''points'' format.\n');
        return;
    end
else
    fprintf(['Error: Invalid temperature schedule format. ', ...
             'Use either ''steps'' or ''points''.\n']);
    return;
end

%
% Physical parameters (temperature-independent)
%
InitConstants(parameters, waferThickness);

%
% Global temperature callback. Interpolates between steps in the time
% profile.
%
global Temperature; % function callback to compute temperature [K] as a function
                    % of time [sec]
Temperature = @(t) interp1(timeProfile, tempProfile, t) + 273.15;

%
% Solution vector layout and sample points
%
global n idxNetCI idxM0DL idxM1DL idxM0Hi idxM1Hi idxSizeOne idxK;
[error, r] = InitSolutionLayout(r);
if error == true
    return;
end

%
% Initialize solutions
%
UpdateTemperature(0);                                   % set initial temperature
fInit = zeros(1,length(n));                             % row vector of solutions
fInit(idxSizeOne) = CO;
fInit((idxSizeOne+1):idxK) = ones(1,idxK-idxSizeOne);   % precipitates up to size k
                                                        % initialized to 1
fInit(idxM0Hi) = m0;
fInit(idxM1Hi) = nAvg*fInit(idxM0Hi);
fInit(idxNetCI) = netCI;

%
```

```
% Run the simulation
%
PrintSimulationSettings(r, tempProfile, timeProfile, fInit, waferThickness);
OutputCallback(0, fInit, 'init');   % initialize progress indicator
[t, f] = ode15s(@ODECallback, [0 timeProfile(end)], fInit);
OutputCallback(0, squeeze(f(end,:)), 'done');

%
% Return all results that were requested
%

% n: sample point vector
if nargout >= 1
    varargout{1} = n;
end

% t: time points vector
if nargout >= 2
    varargout{2} = t;
end

% temp: temperature vector [C]
if nargout >= 3
    varargout{3} = t;
    for i = 1:length(t)
        varargout{3}(i) = Temperature(varargout{3}(i)) - 273.15;
    end
end

% CO: vector of interstitial oxygen
if nargout >= 4
    varargout{4} = squeeze(f(:,idxSizeOne));
end

% m0: vector of m0 (precipitates w/ n >= k)
m0 = [];
m1 = [];
m2 = [];    % unused
if nargout >= 5
    if modelType == 'f'
        % Full model, compute all moments at each time step
        for i = 1:length(t)
            [m0(end+1), m1(end+1), m2(end+1)] = ...
                ComputeMoments(squeeze(f(i,:)), idxK, length(n));
        end
    else
        % RKPM model, copy moments from solution vector
        m0 = squeeze(f(:,idxM0Hi));
        m1 = squeeze(f(:,idxM1Hi));
    end

    varargout{5} = m0;
end

% m1: vector of m1 (all oxygen atoms in precipitates w/ n >= k)
if nargout >= 6
    varargout{6} = m1;
end

% ncrit: vector of critical sizes of oxygen precipitates
SI = [];        % also compute SI (temperature-dependent) here
ncrit = [];
ncritDL = [];
if nargout >= 7
    global CIstar;
    for i = 1:length(t)
        UpdateTemperature(t(i));    % recomputes CI* for this temperature
        SI(end+1) = ComputeCI(squeeze(f(i,:))) / CIstar;
        ncrit(end+1) = CriticalSizeEstimated(f(i,idxSizeOne), 1/SI(end));
        ncritDL(end+1) = LoopCriticalSize(m1(i)/m0(i), 1/SI(end));
    end
```

```
        varargout{7} = ncrit;
    end

    % SI: vector of CI/CI*
    if nargout >= 8
        varargout{8} = SI;
    end

    % m0DL: vector of dislocation loop m0
    if nargout >= 9
        varargout{9} = squeeze(f(:,idxM0DL));
    end

    % m1DL: vector of dislocation loop m1
    if nargout >= 10
        varargout{10} = squeeze(f(:,idxM1DL));
    end

    % ncritDL: vector of critical sizes of dislocation loops
    if nargout >= 11
        varargout{11} = ncritDL;
    end

    % f: solution vector (2D -- time, solution)
    if nargout >= 12
        % If RKPM model, fill in the estimator
        if modelType == 'r'
            for i = 1:length(t)
                k = n(idxK);
                nAvg = max(m1(i)/m0(i),k);
                f(i,idxK) = FkEstimator(m0(i), nAvg);
            end
        end
        varargout{12} = f;
    end

    % idxK: index in solution vector of size k (note: not solved in RKPM model)
    if nargout >= 13
        varargout{13} = idxK;
    end

    % r: sample discretization factor (may have been recomputed)
    if nargout >= 14
        varargout{14} = r;
    end
end

%
% status = OutputCallback(t, f, flag)
%
% Progress indicator. Called by the solver for each time step to display
% information.
%
function status = OutputCallback(t, f, flag)
    persistent  step;
    global      modelType Temperature n idxNetCI idxM0DL idxM1DL idxM0Hi idxM1Hi ...
                idxSizeOne idxK Css CIstar CVstar;

    if isempty(flag)
        % Only print every 100 steps
        step = step + 1;
        if mod(step, 100) == 0
            tempC = Temperature(t) - 273.15;
            CO = f(idxSizeOne);
            SI = ComputeCI(f)/CIstar;
            if modelType == 'f' % compute moments for full model (time consuming!)
                [m0, m1, m2] = ComputeMoments(f, idxK, length(n));
            else                % RKPM model solves moments
                m0 = f(idxM0Hi);
                m1 = f(idxM1Hi);
            end
```

```
            nAvg = m1/m0;
            ncrit = CriticalSizeEstimated(CO, 1/SI);
            m0DL = f(idxM0DL);
            m1DL = f(idxM1DL);

            fprintf(['Step: %d  T=%g C  t=%1.2f sec (%1.1f hr)  CO=%1.2e ', ...
                     '(CO/Css=%1.1f)  CI/CI*=%1.1f  m0=%1.2e  m1=%1.2e  ', ...
                     'nAvg=%1.2e (ncrit=%1.2e)  m0DL=%1.2e  m1DL=%1.2e\n'], ...
                     step, tempC, t, t/3600, CO, CO/Css, SI, m0, m1, nAvg, ...
                     ncrit, m0DL, m1DL);
        end
    elseif flag == 'init'
        step = 0;
        fprintf('Solver Progress\n');
        fprintf('---------------\n');
    else    % flag == 'done'
        fprintf('\nResults\n');
        fprintf('-------\n');

        CO = f(idxSizeOne);
        CI = ComputeCI(f);
        CV = ComputeCV(f);
        m0 = f(idxM0Hi);
        m1 = f(idxM1Hi);
        nAvg = m1 / m0;
        m0DL = f(idxM0DL);
        m1DL = f(idxM1DL);
        nAvgDL = m1DL / m0DL;
        fprintf('CO     = %g cm^-3 (CO/Css = %1.2f)\n', CO, CO/Css);
        fprintf('m0     = %g cm^-3\n', m0);
        fprintf('m1     = %g cm^-3\n', m1);
        fprintf('nAvg   = %g (Avg. radius: %1.1f nm)\n', nAvg, Radius(nAvg)/1e-7);
        fprintf('CI     = %g cm^-3 (CI/CI* = %1.2f)\n', CI, CI/CIstar);
        fprintf('CV     = %g cm^-3 (CV/CV* = %1.2f)\n', CV, CV/CVstar);
        fprintf('m0DL   = %g cm^-3\n', m0DL);
        fprintf('m1DL   = %g cm^-3\n', m1DL);
        fprintf('nAvgDL = %g (Avg. radius: %1.1f nm)\n', nAvgDL, LoopRadius(nAvgDL)/1e-7);

        fprintf('\nFinished.\n\n');
    end

    status = 0;
end

%
% PrintSimulationSettings():
%
% Prints a descriptive summary of the simulation settings.
%
function PrintSimulationSettings(r, tempProfile, timeProfile, fInit, waferThickness)
    global kB n idxSizeOne idxM0Hi idxM1Hi CIstar CVstar Css Css0 CssEa Alpha750 Alpha1050
modelType;

    % Basic settings
    fprintf('Simulation Settings\n');
    fprintf('-------------------\n');
    fprintf('Model:                ');
    if modelType == 'f'
        fprintf('Full\n');
    else
        fprintf('RKPM\n');
    end
    fprintf('Sampling Factor:      %g (%d equations)\n', r, length(n)-4);
    fprintf('Wafer Thickness:      %g um\n', waferThickness);

    % Model parameters
    fprintf('Surface Energy:       %g J/m^2 (T = 750 C), %g (T = 1050 C)\n', ...
            Alpha750, Alpha1050);
    fprintf('Css:                  %g cm^-3 (T = 750 C), %g (T = 1050 C)\n', ...
            Css0*exp(-CssEa/(kB*(750+273.15))), Css0*exp(-CssEa/(kB*(1050+273.15))))
```

```
    % Temperature profile
    fprintf('Temperature Schedule: \n');
    for i = 2:length(tempProfile)
        deltaTime = timeProfile(i) - timeProfile(i-1);
        temp1 = tempProfile(i-1);
        temp2 = tempProfile(i);
        if temp1 == temp2
            UpdateTemperature(timeProfile(i));
            fprintf('\t%g C (%1.2f hr)\n', temp1, deltaTime/3600);
            fprintf('\t\tCss = %g cm^-3\n', Css);
            fprintf('\t\tCI* = %g cm^-3\n', CIstar);
            fprintf('\t\tCV* = %g cm^-3\n', CVstar);
        else
            fprintf('\t%g -> %g C (%g C/min)\n', temp1, temp2, ...
                    abs(temp2-temp1)/(deltaTime/60));
        end
    end

    % Initial conditions
    UpdateTemperature(0);           % reset to initial temperature
    CO = fInit(idxSizeOne);
    CI = ComputeCI(fInit);
    CV = ComputeCV(fInit);
    m0 = fInit(idxM0Hi);
    m1 = fInit(idxM1Hi);
    nAvg = m1 / m0;
    fprintf('CO   = %g cm^-3 (CO/Css = %1.2f)\n', CO, CO/Css);
    fprintf('m0   = %g cm^-3\n', m0);
    fprintf('m1   = %g cm^-3\n', m1);
    fprintf('nAvg = %g (Avg. radius: %1.1f nm)\n', nAvg, Radius(nAvg)/1e-7);
    fprintf('CI   = %g cm^-3 (CI/CI* = %1.2f)\n', CI, CI/CIstar);
    fprintf('CV   = %g cm^-3 (CV/CV* = %1.2f)\n', CV, CV/CVstar);
    fprintf('\n');
end

%
% InitSolutionLayout(r):
%
% Given the sample discretization factor, r, defines all components of the
% solution vector, generates the sample points (n), and defines size k.
%
% Sets error to true if an unrecoverable error occurred and the simulation
% should be aborted. Returns the new, adjusted discretization factor in newR.
%
function [error, newR] = InitSolutionLayout(r)
    % Pointers: indices of solutions in the overall solution vector, f
    global idxNetCI idxM0DL idxM1DL;    % point defects and dislocation loops
    global idxM0Hi idxM1Hi;             % oxygen precipitate moments (n >= k)
    global idxSizeOne;                  % interstitial oxygen
    global idxChangeOver;               % sample spacing becomes > 1 after this
    global idxK;                        % size k precipitate (not solved in RKPM)

    % Samples
    global  n;                      % size n(i) of each solution vector element i
    global  maxSmallClusterSize;    % small cluster/macroscopic transition
                                    % (between 3 and nChangeOver)

    % Imported...
    global  modelType;

    newR = r;

    %
    % Size distribution and layout of solution variables is described below.
    %
    %   Index   Description                                     Pointer
    %   1       NetCI (silicon interstitials minus vacancies)   idxNetCI
    %   2       m0DL (m0 for dislocation loops)                 idxM0DL
    %   3       m1DL (m1 for dislocation loops)                 idxM1DL
    %   4       --
    %   5       --
```

```
%   6       --
%   7       --
%   8       m1 (DFA approximation of m1 in RKPM case)       idxM1Hi
%   9       m0 (0th moment)                                 idxM0Hi
%   10      CO (oxygen interstitials; i.e., the solute)     idxSizeOne
%   11      CO2 (O2, all point defect states in relative equilibrium)
%   12      f3 (size 3 precipitate)
%   ...
%
% The small cluster/macroscopic transition point, maxSmallClusterSize,
% indicates where macroscopic energies begin. The transition from
% maxSmallClusterSize-1 -> maxSmallClusterSize uses small cluster energies
% for both, maxClusterSize -> maxClusterSize + 1 uses macro energies for
% both.
%
% The change-over from discrete (single spacing) to interpolated equations,
% at index idxChangeOver, must be between size maxSmallClusterSize and index
% idxK.
%
% For the moment-based model, the maximum size is k, beyond which a moment-
% based approximation is used. For the full model, solutions are tracked to
% a much larger size.
%
idxNetCI = 1;
idxM0DL = 2;
idxM1DL = 3;
idxM0Hi = 9;
idxM1Hi = 8;
idxSizeOne = 10;

%
% Set up boundaries of different regions (set these carefully)
%
nChangeOver = 10;           % size beyond which sample spacing can be > 1
maxSmallClusterSize = 3;    % sizes 2 to here are small clusters
k = 72;                     % size at which moments begin in moment model
if modelType == 'f'
    maxSampleSize = 1e9;    % maximum sample size for full model
else
    maxSampleSize = k;      % in RKPM mode, moment model begins at k
                            % (the sample at k won't actually be solved)
end

if (nChangeOver <= maxSmallClusterSize) || (nChangeOver >= k)
    fprintf('Internal error: Discrete change-over size is invalid. Please change it.\n');
    error = true;
    return;
end

%
% Generate samples with logarithmic spacing such that a sample at size k
% exists.
%
idxChangeOver = idxSizeOne-1+nChangeOver;
if r == 1.0
    idxK = k-nChangeOver;
else
    idxK = log((k-nChangeOver)*(r-1)+1)/log(r); % index at which size k currently exists
                                                % (not likely to be integral)
end
idxK = floor(idxK);                                     % make it integral
newR = fzero(@(x) x^idxK - (k-nChangeOver)*(x - 1) - 1, r); % compute ratio necessary for
                                                        % n(idxK) = k
idxK = idxK + idxChangeOver;

% Leading zeros are for non-oxygen solutions that have no 'size'
n = [ zeros(1,idxSizeOne-1) Samples(maxSampleSize, nChangeOver, newR) ];

% Sometimes the math above fails...
if abs(n(idxK)/k-1) > 0.0001
    fprintf(['Internal error: Unable to generate sample point for k = %d. ' ...
```

```
                'Please choose a different rediscretization factor, r.\n'], k);
        error = true;
        return;
    end

    % Success!
    error = false;
end

%
% n = Samples(maxSize, changeOverSize, ratio)
%
% Generate a size space sample vector, n(i). Sample spacing is 1 up to the
% change-over size and then increases by a factor of "ratio" for each sample
% point. The spacing between the change-over size and the following sample is
% guaranteed to be 1.
%
% I didn't know about MATLAB's logspace() when I wrote this :) This could be
% made much simpler...
%
function n = Samples(maxSize, changeOverSize, ratio)
    % Spacing is 1 for [1:changeOverSize]
    n = 1:1:changeOverSize;

    % Logarithmic samples
    delta = 1;
    i = changeOverSize + 1;
    while n(i-1) < maxSize
        n(i) = n(i-1) + delta;
        delta = delta * ratio;
        i = i + 1;
    end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Parameters and Constants
%%
%% Physical constants and parameters are defined here. A global callback,
%% Temperature(t), which returns the temperature in K at time t, must be
%% defined.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% InitConstants(params, waferThickness)
%
% Initializes all temperature-independent global physical constants and
% parameters. Wafer thickness is supplied in microns.
%
function InitConstants(params, waferThickness)
    global kB CSi VolSiO2 VolSi uSi KSi KSiO2;
    global reactDist Alpha750 Alpha1050 Css0 CssEa;
    global HfOVRelaxed HfO2Relaxed HfO2VRelaxed HfV;
    global EbVV EbOV EbOO EbOOnVm;
    global aSi StrainOi StrainOn StrainV;
    global b coreRadius HfSF HfCore KDL indEpsSF;
    global surfReactDist;

    % Conversion factor for [Pa] -> [eV/cm^3]
    PaToLocal       = 6.24151e12;       % 1 [Pa] = 6.24150974e12 [eV/cm^3]

    % Physical constants
    kB              = 8.62e-5;          % Boltzmann constant [eV/K]
    CSi             = 5e22;             % silicon lattice site density [cm^-3]
    VolSiO2         = 4.35e-23;         % volume of SiO2 molecule [cm^3]
                        % (Ref: Hoessinger,
                        % http://www.iue.tuwien.ac.at/phd/hoessinger/node28.html)
    VolSi           = 2.00e-23;         % volume of Si atom [cm^3]
    uSi             = 6.49e10*PaToLocal;% shear modulus of Si [Pa] -> [eV/cm^3]
                                        % (Ref: Hull, Properties of Crystalline Silicon)
    KSi             = 9.78e10*PaToLocal;% bulk modulus of Si [Pa] -> [eV/cm^3] (Ref: Hull)
```

```
    KSiO2           = 3.69e10*PaToLocal;% bulk modulus of SiO2 [Pa] -> [eV/cm^3]
                                        % (Ref: Europhys. Lett., 57 (3), pp. 375-381 2002)

    % Precipitates
    reactDist       = 5e-8;             % interface reaction distance [cm]
    Alpha750        = params(1);        % surface energy at 750C [J/m^2]
    Alpha1050       = params(2);        % surface energy at 1050C [J/m^2]
    Css750          = params(3);        % solid solubility at 750C [cm^-3]
    Css1050         = params(4);        % solid solubility at 1050C [cm^-3]
    T750            = 750 + 273.15;
    T1050           = 1050 + 273.15;
    CssEa           = log(Css750/Css1050)*kB/(1/T1050-1/T750);
                                        % solubility activation energy [eV]
    Css0            = Css750*exp(log(Css750/Css1050)/(T750/T1050-1));
                                        % solubility prefactor [cm^-3]

    % Small cluster formation enthalpies (relative to perfect Si and
    %interstitial O) without strain (fully relaxed, no strain)
    HfOVRelaxed     = 2.02;             % OV [eV]
    HfO2Relaxed     = -0.39;            % O2 [eV]
    HfO2VRelaxed    = 0.51;             % O2V [eV]

    % Vacancy formation enthalpy
    HfV             = 3.5;              % [eV]

    % Binding energies, estimated from ab initio formation energies
    EbVV            = -1.7;             % [eV], literature estimates are from 1.5-2.0
    EbOV            = HfOVRelaxed-HfV;  % [eV]
    EbOO            = HfO2Relaxed;      % [eV]
    EbOOnVm         = EbOO;             % assume same

    % Estimates of small cluster linear (transformational, in radial direction)
    % strain components. From ab initio and valid only for a volume of
    % (2*aSi)^3.
    aSi             = 5.431e-8;         % silicon lattice constant [cm]
    StrainOi        = 0.002325;         % first Oi
    StrainOn        = 0.001146;         % each additional Oi
    StrainV         = -0.004344;        % each V

    % Dislocation loops
    b               = sqrt(3)/3;        % Burgers vector magnitude [aSi]
    coreRadius      = b;                % dislocation core radius [aSi]
    HfSF            = 0.0152;           % stacking fault energy [eV/atom]
    HfCore          = 0;                % core energy [eV/aSi]
    KDL             = 72/sqrt(2);       % dislocation energy prefactor (sqrt(2) was for semi-
                                        %circular geometry) [GPa]
    indEpsSF        = 0.996;            % normalized induced strain of a stacking fault

    % 0D surface boundary condition distance factor
    surfReactDist   = ((waferThickness * 1e-4)^2)/12;
end

%
% UpdateTemperature(t):
%
% This should be called at each time step of the solver. Computes the
% temperature at time t and recalculates all global temperature-dependent
% parameters.
%
function UpdateTemperature(t)
    global kBT DO Css Alpha HAtomic DI DV CIstar CVstar;        % exported
    global Temperature kB CSi Css0 CssEa Alpha750 Alpha1050;    % imported

    % Get current temperature
    T = Temperature(t);

    % Physical constants
    kBT = kB*T;

    % Oxygen parameters
    DO = 0.13*exp(-2.53/kBT);       % oxygen diffusivity [cm^2/sec]
```

```
    Css = Css0 * exp(-CssEa/kBT);   % oxygen solid solubility [cm^-3]
    HAtomic = kBT*log(Css/CSi);     % per-atom formation energy (i.e., Gp) [eV]
    Alpha = 6.24151e14 * interp1([750 1050]+273.15, [Alpha750 Alpha1050], T, 'linear', ...
                                 'extrap'); % surface energy [eV/cm^2]

    % Silicon interstitials and vacancies
    DI = 51.4*exp(-1.77/kBT);       % interstitial diffusivity [cm^2/sec]
    DV = 3.07*exp(-2.12/kBT);       % vacancy diffusivity [cm^2/sec]
    CIstar = (2980/51.4)*5e22*exp((-4.95+1.77)/kBT);    % interstitial equilibrium [cm^-3]
    CVstar = (86/3.07)*5e22*exp((-4.56+2.12)/kBT);      % vacancy equilibrium [cm^-3]
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Dislocation Model
%%
%% Functions for the dislocation model: energetics, etc. The formation energy
%% is repeatedly used here in different forms. Care must be taken that all of
%% these forms are consistent with each other.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% z = LoopStrainEnergy(stress)
%
% Given a stress in [Pa], returns the stress energy per atom [eV] based on the
% induced strain of the stacking fault.
%
function z = LoopStrainEnergy(stress)
    global VolSi indEpsSF;
    u = 1.602e-19;          % unit conversion factor [J/eV]
    v = VolSi * 1e-6;       % atomic volume [cm^3] -> [m^3] because strain is [Pa] = [N/m^2]
    z = -v*indEpsSF*stress; % strain energy per SF atom [J]
    z = z / u;              % [eV]
end

%
% z = PrecipitateStress(n, SV)
%
% Computes the tangential stress (sigma_theta,theta; sigma_phi,phi [Pa]) at the
% very edge of an oxygen precipitate, in the silicon matrix. Equation B.7 from
% the Senkader thesis is used: S. Senkader, "Physical Modeling and Simulation of
% Oxygen Precipitation in Silicon", Vienna University of Technology, 1996. The
% sign convention used by Senkader seems to be opposite of what is expected, so
% the equation is inverted here.
%
function z = PrecipitateStress(n, SV)
    global uSi KSiO2;
    mOpt = MOptimal(n,SV);
    u = uSi / 6.24151e12;   % [Pa = N/m^2]
    K = KSiO2 / 6.24151e12; % [Pa = N/m^2]
    A = -3*K*eT(n,mOpt)/(3*K+4*u);
    z = -2*u*A;             % stress at r = rp (w/ inverted sign)
end

%
% ncrit = LoopCriticalSize(nOx, SI)
%
% Dislocation loop critical size as a function of CI/CI* near an oxygen
% precipitate of size nOx (e.g., m1/m0). Dislocation loop formation energy takes
% the form:
%
%   GtotDL(n) = A*n + B*sqrt(n) + C*sqrt(n)*[log(D*sqrt(n))-1]
%
% Differentiating with respect to n and setting to 0 allows ncrit to be solved:
%
%   0 = A + 0.5*B/sqrt(n) + 0.5*C/sqrt(n)*[log(D*sqrt(n))-1]
%   0 = A*sqrt(n) + 0.5*B + 0.5*C*[log(D*sqrt(n))-1]
%
% Setting x = sqrt(n), we get:
%
%   x = -(0.5*B + 0.5*C*[log(D*x)-1])/A
```

```
%
% Iterating this a few times provides a solution.
%
function ncrit = LoopCriticalSize(nOx, SI)
    global kB kBT HfSF HfCore KDL b coreRadius;

    SV = 1/SI;

    rf = sqrt(sqrt(3)/(8*pi));  % radius factor [aSi/sqrt(atoms)]
    u = .9998;                  % unit conversion factor0 [GPa*aSi^3] -> [eV]

    % Energy equation: Gtot(n) = A*n + B*sqrt(n) + C*sqrt(n)*[log(D*sqrt(n))-1]
    A = -kBT*log(SI) + HfSF + LoopStrainEnergy(PrecipitateStress(nOx, SV));
    B = 2*pi*rf*HfCore;
    C = u*0.5*KDL*b*b*rf;
    D = rf*(8/coreRadius);

    % Iterate
    x = sqrt(1e4);              % initial guess for sqrt(n)
    for i = 1:6
        x = -0.5*(B+C*(log(D*x)))/A;
    end
    ncrit = x*x;
end

%
% z = LoopCnStar(n, nOx, SV)
%
% Equilibrium concentration of interstitials with a size n dislocation loop near
% an oxygen precipitate of size nOx.
%
function z = LoopCnStar(n, nOx, SV)
    global kB kBT CIstar HfSF HfCore KDL b coreRadius;

    % dGtotDL/dn = A + 0.5*B/sqrt(n) + 0.5*C/sqrt(n)*[log(D*sqrt(n))-1]
    % Here we compute the enthalpy portion of this (log(SI) term not included).
    rf = sqrt(sqrt(3)/(8*pi));  % radius factor [aSi/sqrt(atoms)]
    u = .9998;                  % unit conversion factor0 [GPa*aSi^3] -> [eV]
    A = HfSF + LoopStrainEnergy(PrecipitateStress(nOx, SV));
    B = 2*pi*rf*HfCore;
    C = u*0.5*KDL*b*b*rf;
    D = rf*(8/coreRadius);
    dHdn = A + 0.5*B/sqrt(n) + 0.5*C/sqrt(n)*(log(D*sqrt(n))-1);

    % Cn* = CI* * exp{dHtotDL(n)/dn) / kBT}
    z = CIstar * exp(dHdn/kBT);
end

%
% z = LoopFnStar(n, m0DL, m0Ox, nAvgOx, SI)
%
% Equilibrium concentration of size n dislocation loops assuming
% heterogeneous nucleation at oxygen precipitate sites.
%
function z = LoopFnStar(n, m0DL, m0Ox, nAvgOx, SI)
    global kB kBT b HfSF HfCore KDL coreRadius reactDist CSi;

    if n == 0
        z = 0;
        return;
    end

    SV = 1/SI;

    % Compute energy terms
    u = .9998;                  % unit conversion factor [GPa*aSi^3] -> [eV]
    r = sqrt(n*sqrt(3)/(8*pi)); % radius [aSi]
    Gperim = 2*pi*r*HfCore;
    Gself = u*0.5*KDL*b*b*r*(log(8*r/coreRadius)-1);    % elastic self-energy
    G = -n*kBT*log(SI) + n*(HfSF + LoopStrainEnergy(PrecipitateStress(nAvgOx, SV))) ...
        + Gperim + Gself;
```

```
    % fn*, the 4*2*pi*r/a0 factor is because there are 4 <111> planes
    z = (8*pi*Radius(n)/reactDist)*(m0Ox-m0DL)*exp(-G/kBT);
end

%
% z = LoopRadius(n)
%
% Dislocation loop radius [cm].
%
function z = LoopRadius(n)
    global aSi;
    z = aSi * sqrt(n*sqrt(3)/(8*pi));
end

%
% z = LoopLambda(n)
%
% Kinetic growth factor [cm] for dislocation loops.
%
function z = LoopLambda(n)
    global aSi;
    r = sqrt(n*sqrt(3)/(8*pi));
    c = sqrt(3)/4;
    z = 4*pi*pi*aSi*r/log(8*r/c);
end

%
% z = LoopGRate(n, CI)
%
% Dislocation loop growth rate.
%
function z = LoopGRate(n, CI)
    global DI;
    if n == 0
        z = 0;
    else
        z = DI*LoopLambda(n)*CI;
    end
end

%
% z = LoopDRate(n, nOx, SV)
%
% Dislocation loop dissolution rate near a size nOx oxygen precipitate.
%
function z = LoopDRate(n, nOx, SV)
    global DI;
    if n == 0;
        z = 0;
    else
        z = DI*LoopLambda(n)*LoopCnStar(n, nOx, SV);
    end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Energies of Small Oxygen Clusters
%%
%% Energies and optimal number of point defects for discrete, small oxygen
%% clusters, including effective quantities based only on the number of oxygen
%% atoms, n, assuming all possible point defect states, m, are in relative
%% equilibrium.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% z = HStrainDiscrete(n,m)
%
% Computes an estimate of the strain energy for discrete oxygen clusters based
% on ab initio calculations. Only valid for sizes n << 64.
%
```

```
function z = HStrainDiscrete(n,m)
    global uSi KSiO2 aSi StrainOi StrainOn StrainV;

    % eT(n,m): misfit strain, computed from ab initio estimates
    eT = StrainOi + (n-1)*StrainOn + m*StrainV;
    eC = eT / (1+4*uSi/(3*KSiO2));

    % Strain energy. Volume is based on 2x2x2 cell used in VASP for strain
    % calculations: (8*aSi)^3. Because of this fixed volume, these discrete
    % strain results become invalid as the precipitate size approaches the
    % volume of a 64 silicon atom cell.
    z = ((8*aSi)^3)*6*uSi*eT*eC;
end

%
% z = HfDiscrete(n,m)
%
% Computes the formation enthalpy of a small n, m cluster. A very simple
% heuristic based on limited ab initio data is used.
%
% OV is included here, unlike in the solved model. Not intended for use at
% run-time but can be used to pre-compute discrete cluster values for sizes
% 3 and above.
%
function z = HfDiscrete(n,m)
    global HfOVRelaxed HfO2Relaxed HfO2VRelaxed HfV EbVV EbOO EbOV EbOOnVm;

    % Strain energy term
    StrainEnergy = HStrainDiscrete(n,m) - n*HStrainDiscrete(1,0);

    % Handle cases for which we have ab initio data
    if (n == 1)
        if (m == 0)
            z = 0;  % Oi is 0 by definition
            return;
        elseif (m == 1)
            z = HfOVRelaxed+StrainEnergy;
            return;
        end
    elseif (n == 2)
        if (m == 0)
            z = HfO2Relaxed+StrainEnergy;
            return;
        elseif (m == 1)
            z = HfO2VRelaxed+StrainEnergy;
            return;
        end
    end

    %
    % Compute energy by first binding Oi's to dangling bonds caused by V's.
    % Then, any remaining Oi should be paired to the precipitate.
    %

    % Formation energy of all vacancies
    Hf = m*HfV;

    % Reduce energy by number of paired V's
    freeO = n;
    numVV = floor(m/2);
    numDanglingBonds = 4*m-2*numVV;
    Hf = Hf + EbVV*numVV;

    % Each Oi can satisfy two dangling bonds
    if (freeO >= (numDanglingBonds/2))
        Hf = Hf + EbOV*numDanglingBonds/2;
        freeO = freeO - numDanglingBonds/2;
    else
        Hf = Hf + EbOV*freeO;
        freeO = 0;
    end
```

```
    % Remaining Oi bind to each other and then precipitate
    if (freeO >= 2)
        Hf = Hf + EbOO;
        freeO = freeO - 2;
    end
    Hf = Hf + EbOOnVm*freeO;

    % Add in discrete strain energy, less n*HStrain(Oi) (which is our reference)
    z = Hf + StrainEnergy;
end

%
% z = DeltaGfvSmallCluster(n, SV)
%
% Change in energy of small cluster formation due to enthalpy of formation and
% vacancy entropy of mixing terms. All possible m states (i.e., number of point
% defects incorporated) are assumed to be in relative thermal equilibrium:
%
%       fn* = f*n,0 + f*n,1 + ... + f*n,n
%
% From this, we can compute a single DeltaGTotal for a size n small cluster:
%
%       DeltaGTotal(n) = -n*kBT*log(CO/CSi)
%                        - kBT*log(exp{-Hf_n,0/kBT} + SV*exp{-Hf_n,1/kBT} + ...)
%
% And finally, the definition of DeltaGfv is just:
%
%       DeltaGfv = -kBT*log(exp{-Hf_n,0/kBT} + SV*exp{-Hf_n,1/kBT} + ...)
%
% This term is defined because it is not convenient to break down the effective,
% aggregate energy for small clusters of size n any further. It is this term
% that must be used to enforce continuity between small clusters and
% macroscopic precipitates.
%
function z = DeltaGfvSmallCluster(n, SV)
    global kBT;

    % Compute term inside log first
    z = 0;
    for m = 0:n
        z = z + (SV^m)*exp(-HfDiscrete(n,m)/kBT);
    end

    % Return the final expression
    z = -kBT*log(z);
end

%
% z = DeltaGTotalSmallCluster(n, CO, SV)
%
% Total change in formation energy upon small cluster formation. This energy is
% continuous with the small cluster form at the transition point.
%
function z = DeltaGTotalSmallCluster(n, CO, SV)
    global kBT CSi;

    z = -n*kBT*log(CO/CSi) + DeltaGfvSmallCluster(n,SV);
end

%
% UpdateSmallClusterEnergies(SV)
%
% As an optimization, computes DeltaGfvSmallCluster/kBT for all sizes up to
% the transition point, storing them in a global array.
%
% This must be called each time step before computing any quantities that depend
% on precipitate energies (including dissolution rates)!
%
function UpdateSmallClusterEnergies(SV)
    global maxSmallClusterSize kBT;     % imported
```

```
    global DeltaGfvSmallClusterNoKBT;   % exported

    % Compute unitless formation energy factors
    DeltaGfvSmallClusterNoKBT = zeros(maxSmallClusterSize,1);
    for n = 1:maxSmallClusterSize
        DeltaGfvSmallClusterNoKBT(n) = DeltaGfvSmallCluster(n, SV)/kBT;
    end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Energies of Macroscopic Oxygen Precipitates
%%
%% Energies and optimal number of point defects for macroscopic (large size
%% limit) precipitates.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% z = eT(n, m)
%
% Translational strain (linear strain along radial direction) of an oxygen
% precipitate.
%
function z = eT(n, m)
    global VolSiO2 VolSi;
    z = (VolSiO2*n/(2*VolSi*(m+n/2)))^(1/3) - 1;
end

%
% z = MOptimal(n, SV)
%
% Optimal number of point defects (vacancies/interstitials) to eject. That
% is, the number that minimizes the free energy of a size n precipitate. Full
% derivation appears in dissertation. SV is the vacancy supersaturation.
%
function z = MOptimal(n, SV)
    global kBT VolSiO2 VolSi uSi KSiO2;

    % Value of m for 0 strain (point about which eT is linearized)
    m0 = n*((VolSiO2/(2*VolSi)) - 1/2);

    % Optimal value of m based on linear approximation of  eT(n,m)
    z = 0.5 * n * ((3/4)*kBT*log(SV)*(VolSiO2/(VolSi*VolSi*uSi))*(1+4*uSi/(3*KSiO2)) ...
        + (VolSiO2/VolSi) - 1);
end

%
% z = HStrain(n,m)
%
% Precipitate strain energy.
%
% NOTE: Minimum strain energy, HStrain(n,mOpt), should probably be computed in a
% more optimal fashion than calling this directly.
%
function z = HStrain(n,m)
    global VolSi VolSiO2 uSi KSiO2;

    % eT(n,m): misfit strain
    eT = (VolSiO2*n/(2*VolSi*(m+n/2)))^(1/3) - 1;

    % eC
    eC = eT / (1+4*uSi/(3*KSiO2));

    % Strain energy
    z = (6*4*pi/3)*(Radius(n)^3)*uSi*eT*eC;
end

%
% z = GSurface(n)
%
% Surface energy component of a precipitate (temperature-dependent).
```

```
%
function z = GSurface(n)
    global Alpha;
    z = 4*pi*Alpha*Radius(n)^2;
end

%
% z = GfMacro(n, SV)
%
% Returns the formation energy of a size n precipitate using the macroscopic
% form of the equations. This is the formation energy with entropy of mixing
% terms for oxygen interstitials and vacancies excluded.
%
%       Gf(n) = n*HAtomic + GSurface(n) + HStrain(n,mOpt(n))
%
% Note that this term alone is not continuous at the transition point between
% macroscopic precipitates and small clusters.
%
function z = GfMacro(n,SV)
    global HAtomic;
    if (n <= 1)
        z = 0;  % by definition
    else
        mOpt = MOptimal(n,SV);
        z = n*HAtomic+GSurface(n)+HStrain(n,mOpt);
    end
end

%
% z = DeltaGfvMacro(n, SV)
%
% Change in energy upon precipitate formation due to formation energy and
% vacancy entropy of mixing terms. Continuity at the transition size between
% small clusters and macroscopic precipitates is enforced by applying an
% offset.
%
function z = DeltaGfvMacro(n,SV)
    global kBT;

    DeltaGfvMacro3 = -MOptimal(3,SV)*kBT*log(SV) + GfMacro(3,SV);
    DeltaGOffset = DeltaGfvSmallCluster(3,SV) - DeltaGfvMacro3;
    z = -MOptimal(n,SV)*kBT*log(SV) + GfMacro(n,SV) + DeltaGOffset;
end

%
% z = DeltaGTotalMacro(n, CO, SV)
%
% Total change in formation energy upon macroscopic precipitate formation. This
% energy is continuous with the small cluster form at the transition point.
%
function z = DeltaGTotalMacro(n, CO, SV)
    global kBT CSi;

    z = -n*kBT*log(CO/CSi) + DeltaGfvMacro(n,SV);
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Growth and Dissolution Rates for Oxygen Precipitates
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% z = Radius(n)
%
% Radius of a large ("macroscopic") oxygen precipitate in [cm].
%
function z = Radius(n)
    global VolSiO2;
    z = ( 3 * n * ((VolSiO2/2) / (4*pi)) )^(1/3);
end
```

```
%
% z = RatePrefactor(n)
%
% Common prefactor for growth/dissolution rates.
%
function z = RatePrefactor(n)
    global reactDist DO;

    r = Radius(n);
    z = 4 * pi * (r^2) * (DO/(reactDist+r));
end

%
% z = CnStar(n, SV)
%
% Concentration of oxygen interstitials at equilibrium with a size n macroscopic
% precipitate (near the interface).
%
% Cn* is obtained by differentiating the total formation energy change with
% respect to n, setting it to 0, and solving for CO = Cn*:
%
%       Cn* = Css * (CI/CI*)^gammaI * exp{(Gf(n)-Gf(n-1))/kBT}
%
% Where gammaI = mOpt(n) - mOpt(n-1).
%
% Note that this is intended to be used with DRate(n), hence it looks backwards
% when performing the differentiation. When using this in the growth rate, we
% must be consistent, so passing in n+1 is a good idea.
%
function z = CnStar(n, SV)
    global kBT Css;

    mOpt_n = MOptimal(n,SV);
    mOpt_p = MOptimal(n-1,SV);
    deltaM = mOpt_n-mOpt_p;
    deltaGfExc = GSurface(n)+HStrain(n,mOpt_n)-GSurface(n-1)-HStrain(n-1,mOpt_p);
    SI = 1/SV;

    z = Css*(SI^deltaM)*exp(deltaGfExc/kBT);
end

%
% z = GRate(n, CO)
%
% Growth rate from size n to n+1.
%
function z = GRate(n, CO)
    z = RatePrefactor(n)*CO;
end

%
% z = DRate(n, CO, SV)
%
% Dissolution rate of a precipitate from size n to n-1. SV = CV/CV*.
%
% NOTE: Very important to keep this code in sync with the solver callback's
% code for small precipitates.
%
function z = DRate(n, SV)
    global maxSmallClusterSize DeltaGfvSmallClusterNoKBT CSi;

    %
    % There are a few different size regimes to handle (and the transitions
    % between them). Sizes 1 and 2 are special and handled directly in the
    % solver callback.
    %
    if (n < 3)
        z = 0;  % error, DRate() cannot be used with n < 3

    %
    % Dissolution of Small Cluster -> Small Cluster
```

```
    % ------------------------------------------
    %
    % Dissolution of size 3 -> size 2:
    %
    %       d(3) = g(2)*(f2*)/(f3*) = g(2)*exp{(Gtot_3-Gtot_2)/kBT}
    %
    %       Gtot_n,m = -n*kBT*log(CO/CSi) - m*kT*log(CV/CV*) + Hf_n,m
    %       fn,m*    = CSi*exp(-Gtot_n,m/kBT) = CSi*(CO/CSi)^n*(CV/CV*)^m *
    %                                           exp{-Hf_n,m/kBT}
    %       fn*      = fn,0*+fn,1*+...+fn,mMax* = CSi*exp(-Gtot_n/kBT)
    %
    % The overall energy term for size n, Gtot_n, can be computed from the
    % above relations.
    %
    %       Gtot_n = -kT*log((fn,0*+...+fn,mMax*)/CSi)
    %              = -kT*log((CO/CSi)^n*((CV/CV*)^0*exp{-Hf_n,0/kBT}+
    %                                    (CV/CV*)^1*exp{-Hf_n,1/kBT}+ ...
    %                                    (CV/CV*)^mMax*exp{-Hf_n,mMax/kBT}))
    %
    % Size 2 only has two m-states and size 3 has three, resulting in:
    %
    %       d(3) = g(2)*(CSi/CO)^3*(CO/CSi)^2 *
    %              exp{-log(exp{-Hf_3,0/kBT}+(CV/CV*)*exp{-Hf_3,1/kBT}+
    %                       (CV/CV*)^2*exp{-Hf_3,2/kBT})
    %                   - -log(exp{-Hf_2,0/kBT}+(CV/CV*)*exp{-Hf_2,1/kBT})}
    %
    % Easier and equivalent way is to factor out CO/CSi terms leaving the
    % DeltaGfv terms (vacancy entropy of mixing and formation energy) and take
    % the difference of those.
    %
    elseif (n <= maxSmallClusterSize)
        % Note: this deltaG is unitless (kBT is factored out and cancels with
        % the denominator of the exponential, which is why it is omitted).
        deltaGf = DeltaGfvSmallClusterNoKBT(n) - DeltaGfvSmallClusterNoKBT(n-1);
        z = RatePrefactor(n-1)*CSi*exp(deltaGf);

    %
    % Dissolution of Macroscopic Precipitates
    % ---------------------------------------
    %
    % Dissolution of a size n (macro) -> n-1 (macro):
    %
    %   d(n) = g(n-1)*(fn*)/(f(n-1)*) = g(n-1)*exp{(Gtot_n-Gtot_n-1)/kBT}
    %
    % Where
    %
    %   Gtot_n  = -n*kT*log(CO/CSi) - mOpt(n)*kT*log(CV/CV*) + Gf(n)
    %   Gf(n)    = n*Gp + Gsurf(n) + Hstrain(n,mOpt(n))
    %
    % Resulting in:
    %
    %   d(n) = g(n-1)*(CSi/CO)*(CV*/CV)^(DeltaM)*exp{(Gf(n)-Gf(n-1))/kBT}
    %   DeltaM = mOpt(n)-mOpt(n-1)
    %
    % Note that CO will cancel with CO in g(n-1).
    %
    else
        z = RatePrefactor(n-1) * CnStar(n, SV);
    end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Diagnostics
%%
%% Functions that compute quantities useful for diagnostic and model development
%% purposes.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% z = FnStar(n, CO, SV)
```

```
%
% Equilibrium concentration of size n precipitate, for debugging purposes.
%
function z = FnStar(n, CO, SV)
    global maxSmallClusterSize kBT CSi;

    if (n <= maxSmallClusterSize)
        Gtot = DeltaGTotalSmallCluster(n, CO, SV);
    else
        Gtot = DeltaGTotalMacro(n, CO, SV);
    end

    z = CSi*exp(-Gtot/kBT);
end

%
% z = CriticalSizeEstimated
%
% This is a new expression derived on 14 Nov 2012. It relies on the fact
% that the minimum strain, eT(n,mopt), is independent of n.
%
% A negative result means precipitates cannot form.
%
function z = CriticalSizeEstimated(CO, SV)
    global Alpha KSiO2 VolSi VolSiO2 uSi kBT Css;

    R = 3*(VolSiO2/2)/(4*pi);
    X = (3/2) * kBT * log(SV) * (VolSiO2/(4*VolSi*VolSi*uSi)) * (1+(4*uSi)/(3*KSiO2)) ...
        + (VolSiO2/(2*VolSi)) - (1/2);
    eTmin = (VolSiO2/(2*VolSi*((1/2)+X)))^(1/3) - 1;
    HsPrime = 8*pi*R*uSi*eTmin*eTmin/(1+(4*uSi)/(3*KSiO2));

    z = (4*pi*Alpha*(2/3)*(R^(2/3)) / (kBT*log((CO/Css)*(SV^X))-HsPrime))^3;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Differential Equations
%%
%% The complete set of kinetic rate equations that comprise the core of the
%% model are defined here. The ODE solver callback evaluates the right-hand side
%% of each equation at a particular time step.
%%
%% NOTE: Because of small oxygen clusters, which are a special case, the ODE
%% callback and supporting function must be kept in careful sync with the code
%% which computes dissolution rates (based on precipitate energies).
%%
%% O2 and O2V are both assumed to be present in f2. O3, O3V, and O3V2 are
%% assumed for f3. f1 only includes Oi (i.e., CO), not OV.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%
% z = ComputeCI(f)
%
% CI and CV are assumed to be in equilibrium: CICV=CI*CV*. Only interstitial
% emission is tracked for oxygen precipitation, so it makes sense to only
% consider "net CI". This results in two simultaneous equations, allowing CI
% and CV to be computed on demand:
%
%       CI*CV = (CI*)*(CV*)
%       NetCI = CI - CV
%
function z = ComputeCI(f)
    global idxNetCI CIstar CVstar;
    NetCI = f(idxNetCI);
    z = 0.5*(NetCI+sqrt(NetCI^2+4*CIstar*CVstar));
end

%
% z = ComputeCV(f)
%
```

```
% Computes CV based on the current NetCI, CI*, and CV*.
%
function z = ComputeCV(f)
    global idxNetCI;
    NetCI = f(idxNetCI);
    z = ComputeCI(f) - NetCI;
end

%
% z = ComputeIEjectionRate(reactTerm, df, maxIdx)
%
% Computes the rate of change of silicon interstitials due to precipitate growth
% for this time step.
%
% maxIdx is a parameter to allow us to avoid counting anything that is already
% accounted for by the moments. This function sums from size 2 up until maxIdx.
%
function z = ComputeIEjectionRate(df, maxIdx)
    global n idxSizeOne;

    % Small clusters
    % ... we assume no interstitial ejection for these ...
    z = 0;

    %
    % Macroscopic precipitates: assume fixed ratio of interstitials. This is
    % justified because mOpt/n typically stays within a small range about 0.5.
    % Given the inaccuracies in the dislocation loop model, an error in the
    % exact number of ejected interstitials ejected is presumed to be less
    % important than an error in the strain energy, which effects precipitate
    % growth rates. Therefore, mOpt is used to compute energies but 0.5*n is
    % used to emit interstitials.
    %
    for i=(idxSizeOne+3):maxIdx
        if (i == length(n))
            sampFactor = (2*(n(i)-n(i-1))/2);
        else
            sampFactor = (n(i+1)-n(i-1))/2;
        end

        ejected = 0.5*n(i);
        z = z + sampFactor*ejected*df(i);
    end
end

%
% [m0, m1, m2] = ComputeMoments(f, startIdx, endIdx)
%
% Computes the moments of the distribution between the start and ending indices.
% The input solution vector, f, must be one-dimensional (i.e., no time
% components).
%
function [m0, m1, m2] = ComputeMoments(f, startIdx, endIdx)
    global n;

    m0 = 0;
    m1 = 0;
    m2 = 0;
    for i = startIdx:endIdx
        if (i == length(n))
            sampFactor = (2*(n(i)-n(i-1))/2);
        else
            sampFactor = (n(i+1)-n(i-1))/2;
        end

        m0 = m0 + sampFactor*f(i);
        m1 = m1 + n(i)*sampFactor*f(i);
        m2 = m2 + n(i)*n(i)*sampFactor*f(i);
    end
end
```

```
%
% [a, b, c, d] = Discretize(np, nc, nn, CO, SV)
%
% Computes the Kobayashi discretization coefficients for the solution variable
% at index i (size nn, with size np being the previous sample point and nn being
% the next). The solution will have the form:
%
%   df(i)/dt = (a*f(i-1)-b*f(i)) - (c*f(i)-d*f(i+1))
%
% The first term, a*f(i-1)-b*f(i) corresponds to the flux from n(i-1) -> n(i).
% The second corresponds to the flux from n(i) -> n(i+1). When i is the end
% point, c and d are invalid and should be ignored. The form of the equation is
% then:
%
%   df(i)/dt = a*f(i-1) - b*f(i)
%
% Alternatively, it may be possible to just assume df(i)/dt = 0 because care
% should have been taken to make the maximum end point large enough that the
% solution drops to 0 there.
%
% This function should only be used from i = idxChangeOver onwards.
%
% The derivation of the coefficients appears in Appendix B of: S. Kobayashi,
% Journal of Crystal Growth, vol. 174, pp. 163-169 (1997). The same notation is
% used here.
%
function [a, b, c, d] = Discretize(np, nc, nn, CO, SV)
    global n;

    mW = np;    % previous sample point
    mP = nc;    % current (center)
    mE = nn;    % next

    delta_mP = (mE-mW)/2;
    delta_mw = mP-mW;
    delta_me = mE-mP;

    % PP and QP (roughly the inflow and outflow, relative to current point, nc)
    pP = GRate(mP,CO);
    qE = DRate(mE,SV);
    PP = (pP-qE)/(1-(qE/pP)^delta_me);
    %PP = (pP^delta_me)*(pP-qE)/(pP^delta_me-qE^delta_me);

    pW = GRate(mW,CO);
    qP = DRate(mP,SV);
    QP = (pW-qP)/((pW/qP)^delta_mw-1);
    %QP = (qP^delta_mw)*(pW-qP)/(pW^delta_mw-qP^delta_mw);

    % PW (analogous to PP above, w/ changes: P -> W, E -> P, delta_me -> delta_mw
    PW = (pW-qP)/(1-(qP/pW)^delta_mw);
    %PW = (pW^delta_mw)*(pW-qP)/(pW^delta_mw-qP^delta_mw);

    % QE (analogous to QP above, w/ changes: W -> P, P-> E, delta_mw -> delta_me
    QE = (pP-qE)/((pP/qE)^delta_me-1);
    %QE = (qE^delta_me)*(pP-qE)/(pP^delta_me-qE^delta_me);

    % Return coefficients
    a = PW / delta_mP;
    b = QP / delta_mP;
    c = PP / delta_mP;
    d = QE / delta_mP;
end

%
% z = FkEstimator(m0, nAvg)
%
% Empirical f(k) estimator.
%
function z = FkEstimator(m0, nAvg)
    p = 2;
    p0 = 5e-6;
```

```
    p1 = 0.1;
    z = m0/(((p0*(nAvg-72))^p+p1)*(nAvg-72)+1);
end

%
% df = ODECallback(t, f)
%
% ODE solver callback. Computes df/dt at time t for all solution variables.
%
function df = ODECallback(t, f)
    global modelType kBT n idxChangeOver idxNetCI idxSizeOne idxM0DL idxM1DL idxM0Hi ...
           idxM1Hi idxK surfReactDist reactDist CSi DO Css DI CIstar DV CVstar;

    % Update temperature and parameters
    UpdateTemperature(t);

    % Clear solution
    df = zeros(length(n),1);

    %
    % Silicon interstitials and vacancies:
    %
    % There is no explicit I/V recombination term and we assume that
    % CICV = CI*CV*.
    %
    % What is actually solved for is NetCI = CI - CV.
    %
    CI = ComputeCI(f);
    CV = ComputeCV(f);

    % Point defect supersaturations
    SI = CI/CIstar;
    SV = CV/CVstar;

    % Precompute small cluster formation energies for this time step
    UpdateSmallClusterEnergies(SV);

    %
    % Size 2 is special. O2 is mobile and an O2V state (one point defect
    % incorporated) exists. It is treated the same as other, larger small
    % clusters: the O2 and O2V states are assumed to be present in
    % proportion to their relative equilibrium concentrations.
    %
    % Relevant reactions are:
    %
    %   O + O   <-> O2 (f2 contains O2 and O2V)
    %   f2 + O  <-> f3 (all O3Vm precipitates)
    %   f3 + O  <-> f4
    %
    % Note that O+O only forms the O2 (not O2V) state. The dissolution rate
    % for size 2 is adjusted to only take into account the clusters in the
    % O2 state:
    %
    %   f2* = f2,0* + f2,1*
    %
    % Where f2,m* = CSi*exp{-Gtot_2,m/kBT} = CSi*(CO/CSi)^2 *
    %                                        (CV/CV*)^m*exp{-Gf_2,m/kBT}
    %
    %   f2,0*/f2* = exp{-Gf_2,0/kBT}/(exp{-Gf_2,0)/kBT}+
    %                                 (CV/CV*)*exp{-Gf_2,1/kBT})
    %

    % Remaining solute
    CO = f(idxSizeOne);

    % Size 2 energies
    HfO2  = HfDiscrete(2,0);
    HfO2V = HfDiscrete(2,1);

    % O + O <-> O2
    FractionO2 = 1/(1+SV*exp((HfO2-HfO2V)/kBT));
```

```
Rate_O_O_To_f2 = 4*pi*reactDist*DO*(CO*CO ...
                 - CSi*exp(HfO2/kBT)*FractionO2*f(idxSizeOne+1));

% f2 + O <-> f3
Rate_f2_O_To_f3  = GRate(2,CO)*f(idxSizeOne+1) - DRate(3,SV)*f(idxSizeOne+2);

% Sizes 2 and 3
df(idxSizeOne+1) = Rate_O_O_To_f2 - Rate_f2_O_To_f3;
df(idxSizeOne+2) = Rate_f2_O_To_f3 - (GRate(n(idxSizeOne+2),CO)*f(idxSizeOne+2) ...
                                    - DRate(n(idxSizeOne+3),SV)*f(idxSizeOne+3));

% Conservation of solute
df(idxSizeOne)   = - 2*df(idxSizeOne+1) - 3*df(idxSizeOne+2);

%
% Sizes 4 to nChangeOver-1: Discrete chemical rate equations (sample
% spacing is 1)
%
for i = (idxSizeOne+3):(idxChangeOver-1)
    df(i) = GRate(n(i)-1,CO)*f(i-1)-DRate(n(i),SV)*f(i) ...
            - (GRate(n(i),CO)*f(i)-DRate(n(i)+1,SV)*f(i+1));
    df(idxSizeOne) = df(idxSizeOne) - n(i)*df(i);
end

%
% Sizes nChangeOver and above: Interpolated rate equations.
%
% For full model, generate equations up to maximum size. For RKPM model,
% only up to size k-1 (k is special and has to be estimated).
%
if modelType == 'f'
    maxIdx = length(n);
else
    maxIdx = idxK - 1;
end
for i = idxChangeOver:maxIdx
    % Compute reaction rate and change in solute
    if (i == length(n))
        % Compute interpolation coefficients assuming a fictitious sample
        % point same distance out from previous one
        [a, b, c, d] = Discretize(n(i-1), n(i), n(i)+(n(i)-n(i-1)), CO, SV);
        %df(i) = 0;
        df(i) = a*f(i-1) - b*f(i);
        df(idxSizeOne) = df(idxSizeOne) - (n(i)*2*(n(i)-n(i-1))/2) * df(i);
    else
        % Compute interpolation coefficients
        [a, b, c, d] = Discretize(n(i-1), n(i), n(i+1), CO, SV);

        % Equations
        if (i == (idxK-1)) && (i == maxIdx)
            % This occurs only in RKPM. The expression computed here is
            % incomplete and lacks the reverse term (outbound flux to f(k)).
            % It will be filled in later when dm0/dt is computed.
            df(i) = a*f(i-1) - b*f(i);
            % When df(k) is filled in later, df(idxSizeOne) will be updated
        else
            df(i) = a*f(i-1) - b*f(i) - (c*f(i) - d*f(i+1));
            df(idxSizeOne) = df(idxSizeOne) - (n(i)*(n(i+1)-n(i-1))/2) * df(i);
        end
    end
end

%
% Moments of high distribution
%
m0 = max(0, f(idxM0Hi));
m1 = f(idxM1Hi);
k = n(idxK);
nAvg = max(m1/m0, k);

%
```

```
% Full model case
%
if modelType == 'f'
    %
    % m1 (high part of distribution)
    %
    for i = idxK:length(n)
        if (i == length(n))
            sampFactor = (2*(n(i)-n(i-1))/2);
        else
            sampFactor = (n(i+1)-n(i-1))/2;
        end
        df(idxM1Hi) = df(idxM1Hi) + n(i)*sampFactor*df(i);
    end

    %
    % m0
    %
    [a, b, c, d] = Discretize(n(idxK-1), n(idxK), n(idxK)+1, CO, SV);
    Ik = a*f(idxK-1) - b*f(idxK);
    sampFactorK = (n(idxK)+1-n(idxK-1))/2;
    df(idxM0Hi) = sampFactorK*Ik;

    if (f(idxM0Hi) < 1e-10) && (df(idxM0Hi) < 0)
        df(idxM0Hi) = 0;
    end

%
% RKPM model case
%
else
    %
    % Size k is special. There is no solution at idxK+1 so we must estimate
    % f(k).
    %
    [a, b, c, d] = Discretize(n(idxK-1), n(idxK), n(idxK)+1, CO, SV);
    sampFactorK = (n(idxK)+1-n(idxK-1))/2;
    f(idxK) = FkEstimator(f(idxM0Hi), nAvg);

    %
    % Delta function approximation
    %

    % Compute gammas. gamma3 includes adjustment factor for when nAvg -> k.
    gamma2 = RatePrefactor(nAvg)/DO;
    gamma3 = DRate(nAvg,SV)/DO - DRate(k,SV)*f(idxK)/(DO*m0);

    % Compute dm1/dt
    % To-do: why is n(idxK)+n(idxK)-n(idxK-1) unstable for discretization?
    [a, b, c, d] = Discretize(n(idxK-1), n(idxK), n(idxK)+1, CO, SV);
    sampFactorK = (n(idxK)+1-n(idxK-1))/2;
    Ik = a*f(idxK-1) - b*f(idxK);   % flux from k-1 -> k
    df(idxM1Hi) = sampFactorK*k*Ik + DO*m0*(CO*gamma2-gamma3);

    %
    % m0 and connection back to f(k-1)
    %
    % The use of sampFactorK here is in order to be consistent with
    % ComputeMoments(), otherwise, plugging in the full f(k) here will
    % not result in the same moments as those obtained from a full
    % simulation where ComputeMoments() is invoked at each time step.
    %
    df(idxM0Hi) = sampFactorK*Ik;

    % Recompute Ik interpolation as for outgoing flux at k-1 -> k
    [a, b, c, d] = Discretize(n(idxK-2),n(idxK-1),n(idxK),CO,SV);
    Ik = c*f(idxK-1)-d*f(idxK);

    % Connect it back to df(k-1) equation partially computed earlier
    df(idxK-1) = df(idxK-1) - 1*Ik;
```

```
        %
        % Conservation of solute (m1 and f(k-1))
        %
        df(idxSizeOne) = df(idxSizeOne) - (n(idxK-1)*(n(idxK)-n(idxK-2))/2) * df(idxK-1) ...
                         - df(idxM1Hi);
    end

    %
    % Dislocation loop nucleation and growth
    %
    nc = LoopCriticalSize(nAvg, SI);
    if (imag(nc) ~= 0) || (nc == NaN)
        nc = 0;
    end
    m0DL = max(0, f(idxM0DL));
    m1DL = max(0, f(idxM1DL));
    if m0DL ~= 0
        m1DLhat = m1DL/m0DL;
    else
        m1DLhat = 0;    % loop growth rate for n = 0 will return 0
    end
    if (nc > 1e3)
        df(idxM0DL) = 0;
        df(idxM1DL) = (LoopGRate(m1DLhat, CI)*m0DL - LoopDRate(m1DLhat, nAvg, SV)*m0DL);
    else
        df(idxM0DL) = LoopGRate(nc, CI)*LoopFnStar(nc, m0DL, m0, nAvg, SI);
        df(idxM1DL) = nc*df(idxM0DL) + LoopGRate(m1DLhat, CI)*m0DL ...
                      - LoopDRate(m1DLhat, nAvg, SV)*m0DL;
    end

    %
    % Point defects
    %

    % 0D boundary condition for point defects (approximation of diffusion to and
    % absorption at wafer surfaces.
    ISurfaceTerm = -((DI/surfReactDist)*(CI-CIstar)-(DV/surfReactDist)*(CV-CVstar));

    % Oxygen precipitates eject interstitials, loops consume them
    df(idxNetCI) = ISurfaceTerm + ComputeIEjectionRate(df, maxIdx) - df(idxM1DL);
    if modelType == 'r'
        df(idxNetCI) = df(idxNetCI) + 0.5*df(idxM1Hi);
    end

    % For some reason, calling the progress indicator here manually is faster
    % than using a solver callback passed to ode15s!
    OutputCallback(t, f, []);
end
```

```
%%
%% ====================================
%%  OxygenExperiments.m
%%  by Bart Trzynadlowski, 2012-2013
%% ====================================
%%
%% Using OModel(), replicates all experimental results discussed in my doctoral
%% dissertation ("Reduced Moment-Based Models for Oxygen Precipitates and
%% Dislocation Loops in Silicon," University of Washington, 2013).
%%
%% Version History
%% ---------------
%%
%% 2013.03.20:
%%  - Initial release included in doctoral dissertation.
%%

%
% OxygenExperiments()
%
% Runs the RKPM oxygen model with the best-fit parameters for a series of
% oxygen precipitation experiments and plots the results of each. Experiments
% are run one by one. The full set takes a long time to complete.
%
function OxygenExperiments
    global NetCI1200 IOCToASTM ppmaToConc;

    % CI*-CV* at T=1200 C is used to express the initial NetCI in a more
    % convenient form
    NetCI1200 = ((2980/51.4)*5e22*exp((-4.95+1.77)/(8.62e-5*(1200+273.15)))- ...
                (86/3.07)*5e22*exp((-4.56+2.12)/(8.62e-5*(1200+273.15))));

    % Conversion factor for IOC-88 -> new ASTM (1983) calibration standard
    IOCToASTM = 2.45 / 3.14;

    % Conversion factor for ppma -> cm^-3
    ppmaToConc  = 5e22 / 1e6;

    % Run all the experiments
    tic;
    ExperimentStewart();
    ExperimentChiou();
    ExperimentSwaroop();
    ExperimentKennel();
    time = toc;

    fprintf('All simulations finished. Run time: %1.1f minutes.\n', time/60);
end

%
% ExperimentStewart()
%
% SANS study of precipitation during a single-step 750 C anneal.
%
% Reference:
%
%   R. J. Stewart, S. Messoloras and S. Rycroft, "A small angle neutron
%   scattering study of oxygen precipitation in silicon," in Proceedings of the
%   NATO Advanced Research Workshop, Exeter, UK, 1996.
%
function ExperimentStewart
    global NetCI1200 IOCToASTM;

    % Best-fit initial conditions
    InitialM0       = (1e8)^0;
    InitialSize     = 72 + (1e4)^0;
    InitialNetCI    = NetCI1200 * (1e-3)^0.54;

    %
    % Data from Stewart et al.
    %
```

```
    % The initial oxygen concentration was measured using FTIR with the IOC-88
    % calibration standard. Precipitated oxygen was measured using SANS, which
    % should not need to be re-calibrated. Coincidentally, however, I discovered
    % that scaling the SANS measurements does preduce a much better fit, but
    % that is not done here.
    %

    COInit      = IOCToASTM * 9.6e17;               % FTIR measurement
    Times       = [ 24 72 216 ];                    % [hr]
    COData      = COInit - 1e17 * [ 1.5 1.3 6.2 ];  % SANS measurements

    % Simulate
    [n, t, temp, CO, m0, m1] = OModel('rkpm', 1.1, COInit, InitialM0, InitialSize, ...
                                      InitialNetCI, 400, 'steps', [ 750, 1000 ]);

    % Plot
    figure;
    semilogx(t/3600, CO, '-', Times, COData, 's');
    axis([10 1000 0 1e18]);
    legend('Simulation', 'Data', 'Location', 'NorthEast');
    title('SANS Experiment by Stewart et al. (1996)');
    xlabel('Time (hr)');
    ylabel('Interstitial Oxygen (cm^{-3})');
end

%
% ExperimentSwaroop()
%
% Two-step precipitation experiment: 750 C / 4 hr + 1050 C / 16 hr, 10 C/min
% ramp rate. The data here is a statistical reduction of a larger set of
% measurements that appeared in the original paper by Swaroop et al.
%
% Reference:
%
%   R. Swaroop, N. Kim, W. Lin, M. Bullis, L. Shive, A. Rice, E. Castel and M.
%   Christ, "Testing for Oxygen Precipitation in Silicon Wafers," Solid State
%   Technology, pp. 85-89, March 1987.
%
function ExperimentSwaroop
    global NetCI1200 ppmaToConc;

    % Best-fit initial conditions
    InitialM0       = (1e8)^1;
    InitialSize     = 72 + (1e4)^0.9939;
    InitialNetCI    = NetCI1200 * (1e-3)^1.4;

    % Statistically reduced data from Swaroop et al. Initial vs. precipitated
    % oxygen. FTIR data, new ASTM (1983) calibration factor.
    COInit = ppmaToConc * [ 10.2261 10.47 11.3236 11.8461 12.3515 12.8209 13.3938 ...
                            13.7954 14.2295 14.8199 15.219 15.8443 16.2794 16.7318 ...
                            17.3759 17.7934 18.2113 18.6116 ];
    COLoss = ppmaToConc * [ 0.392663 0.464778 0.503442 0.719044 0.471481 1.6131 4.03753 ...
                            2.93464 4.61048 6.9281 8.81756 10.9572 11.5287 12.2072 ...
                            12.7438 13.4221 13.7799 14.1375 ];

    % Run each point
    for i = 1:length(COInit)
        [n, t, temp, CO] = OModel('rkpm', 1.1, COInit(i), InitialM0, InitialSize, ...
                                  InitialNetCI, 400, 'steps', [ 750, 4, 10, 1050, 16 ]);
        SimLoss(i) = COInit(i)-CO(end);
    end

    % Plot
    figure;
    axis([5e17 10e17 0 10e17]);
    plot(COInit, SimLoss, '-', COInit, COLoss, 's');
    legend('Simulation (750 ^oC / 4 hr + 1050 ^oC /16 hr)', 'Data', 'Location', 'NorthWest');
    title('Two-Step Precipitation Experiment by Swaroop et al. (1987)');
    xlabel('Initial Oxygen (cm^{-3})');
    ylabel('Precipitated Oxygen (cm^{-3})');
end
```

```
%
% ExperimentChiou()
%
% Two-step precipitation experiments by Chiou & Shive:
%
%   1. Set A: 800 C / 2hr + 1050 C / 16 hr, 40 C/min ramp rate
%   2. Set B: 800 C / 1hr + 1050 C / 8 hr, 40 C/min ramp rate
%
% Reference:
%
%   H.-D. Chiou and L. W. Shive, "Test Methods for Oxygen Precipitation in
%   Silicon," VLSI Science and Technology/1985, pp. 429-435, 1985.
%
function ExperimentChiou
    global NetCI1200 ppmaToConc;

    % Best-fit initial conditions
    InitialM0       = (1e8)^1.1234;
    InitialSize     = 72 + (1e4)^0.972;
    InitialNetCI    = NetCI1200 * (1e-3)^1.036;

    %
    % Statistically reduced FTIR data from Chiou & Shive (Fig. 5), new ASTM
    %

    % Set A: 800 C / 2 hr + 1050 C / 16 hr
    COInitA = ppmaToConc * [ 13.42363 14.10402 14.49531 15.0197 15.5011 16.0044 16.5077 ...
                             17.0547 17.558 18.0613 18.5646 19.0678 ];
    COLossA = ppmaToConc * [ 0.586008 1.296908 1.9181 3.862636572 5.15996605 6.973240236 ...
                             9.101962048 11.08297337 11.880519 12.30062409 13.03411303 ...
                             13.33950589 ];

    % Set B: 800 C / 1 hr + 1050 C / 8 hr
    COInitB = ppmaToConc * [ 12.7002 12.8753 13.6411 13.9256 14.4945 14.9759 15.9825 ...
                             16.5295 17.0328 17.4486 18.5646 19.046 19.5711 20.0525 ...
                             20.4683 ];
    COLossB = ppmaToConc * [ 0.331459384 0.422737441 0.463782308 0.68762132 0.887194537 ...
                             1.459914686 2.267171456 3.480235626 4.287054927 5.929382049 ...
                             9.98575922 10.95526322 12.73558444 12.88391971 13.33950589 ];

    % Run set A
    for i = 1:length(COInitA)
        [n, t, temp, CO] = OModel('rkpm', 1.1, COInitA(i), InitialM0, InitialSize, ...
                                  InitialNetCI, 400, 'steps', [ 800, 2, 40, 1050, 16 ]);
        SimLossA(i) = COInitA(i)-CO(end);
    end

    % Run set B
    for i = 1:length(COInitB)
        [n, t, temp, CO] = OModel('rkpm', 1.1, COInitB(i), InitialM0, InitialSize, ...
                                  InitialNetCI, 400, 'steps', [ 800, 1, 40, 1050, 8 ]);
        SimLossB(i) = COInitB(i)-CO(end);
    end

    % Plot
    figure;
    axis([5e17 10e17 0 10e17]);
    plot(COInitA, SimLossA, '-', COInitA, COLossA, 's', COInitB, SimLossB, '-', ...
         COInitB, COLossB, 's');
    legend('Simulation (800 ^oC / 2 hr + 1050 ^oC /16 hr)', 'Data', ...
           'Simulation (800 ^oC / 1 hr + 1050 ^oC / 8 hr)', 'Data', ...
           'Location', 'NorthWest');
    title('Two-Step Precipitation Experiments by Chiou & Shive (1985)');
    xlabel('Initial Oxygen (cm^{-3})');
    ylabel('Precipitated Oxygen (cm^{-3})');
end

%
% ExperimentKennel()
%
```

```
% Precipitation experiments with variable-length nucleation (750 C) and growth
% (1100 C) anneals.
%
% Reference:
%
%   H. W. Kennel, "Physical Modeling of Oxygen Precipitation, Defect Formation,
%   and Diffusion in Silicon," doctoral dissertation, Stanford University, 1991.
%
function ExperimentKennel
    global NetCI1200 IOCToASTM;

    % Best-fit initial conditions
    InitialM0       = (1e8)^1;
    InitialSize     = 72 + (1e4)^0.9;
    InitialNetCI    = NetCI1200 * (1e-3)^1.1;

    %
    % Data from Kennel Fig. 3.5
    %

    COInit =        IOCToASTM * 9.8e17; % initial oxygen
    AnnealTime =    [ 0 4 8 20 ];       % growth step anneal times [hr]

    % CO remaining after anneal for 0 hr and 2 hr nucleation times
    COData0 =   IOCToASTM * [ 9.5522E+17 9.1429E+17 8.8331E+17 7.87055E+17  ];
    COData2 =   IOCToASTM * [ 9.6019E+17 8.74485E+17 7.3074E+17 4.93526E+17 ];

    % Run 0 hr nucleation data
    for i = 1:length(AnnealTime)
        if AnnealTime(i) ~= 0
            [n, t, temp, CO] = OModel('rkpm', 1.1, COInit, InitialM0, InitialSize, ...
                                      InitialNetCI, 400, 'steps', ...
                                      [ 750, 1/3600, 20, 800, 10/3600, 7, 1100, ...
                                        AnnealTime(i) ]);
        else
            [n, t, temp, CO] = OModel('rkpm', 1.1, COInit, InitialM0, InitialSize, ...
                                      InitialNetCI, 400, 'steps', ...
                                      [ 750, 1/3600, 20, 800, 10/3600 ]);
        end
        COSim0(i) = CO(end);
    end

    % Run 2 hr nucleation data
    for i = 1:length(AnnealTime)
        if AnnealTime(i) ~= 0
            [n, t, temp, CO] = OModel('rkpm', 1.1, COInit, InitialM0, InitialSize, ...
                                      InitialNetCI, 400, 'steps', ...
                                      [ 750, 2, 20, 800, 10/3600, 7, 1100, ...
                                        AnnealTime(i) ]);
        else
            [n, t, temp, CO] = OModel('rkpm', 1.1, COInit, InitialM0, InitialSize, ...
                                      InitialNetCI, 400, 'steps', ...
                                      [ 750, 2, 20, 800, 10/3600 ]);
        end
        COSim2(i) = CO(end);
    end

    % Plot
    figure;
    axis([0 10 2.5e17 1e18]);
    plot(AnnealTime, COSim0, '-', AnnealTime, COData0, 's', AnnealTime, COSim2, '-', ...
         AnnealTime, COData2, 's');
    legend('Simulation (No nucleation step)', 'Data', ...
           'Simulation (2 hr 750 ^oC nucleation)', 'Data', 'Location', 'SouthWest');
    title('1100 ^oC Annealing Experiments by Kennel (1991)');
    xlabel('Anneal Time (hr)');
    ylabel('Interstitial Oxygen (cm^{-3})');
end
```