


AN ABSTRACT OF THE THESIS OF

Uwe Schmid for the degree of Master of Science in Physics
presented on November 17, 1988.

Title: Monte Carlo Simulation of Atomic Diffusion in Si and GaAs
via Vacancies and Interstitials

Redacted for Privacy

Abstract approved: 

James A. Van Vechten

A computer simulation program, which runs efficiently on micro-computers, for the diffusion of point defects in the diamond and zincblende structure was developed. The simulated point defects comprise vacancies, host or impurity interstitials on any of three types of interstitials sites, and impurities. Diffusion via vacancy first and second neighbor hopping, the Frank-Turnbull (vacancy-interstitial) and the kick-out (host-impurity-interstitial) mechanism are simulated. The program is entirely controlled by the user, who may simulate various diffusion models with different parameters, such as activation energies and bond energies. Thus, the evolution of an initial defect configuration -

as fed to the program - can be monitored in space and time, giving insight into the simulated model. The program was applied to the diffusion of Au into Si. It gave U-shaped profiles, similar to those observed, for both of the previously proposed models, a simple Frank-Turnbull and a simple kick-out hypothesis. These ignore any effects due to the charge states of the various species, variation of the Fermi level across the sample and throughout the process, and any effect due to electron-hole recombination enhancement of the process together with the variation of the minority carrier lifetime as a result of this process. Whereas the data of the simple Frank-Turnbull could be fitted to a complementary error function, the kick-out yielded an exponential profile close to the surface. Neither of the models could account for the right time dependence of the central gold concentration, which is known experimentally to be a square root law. It can be concluded that neither suggested model contains the physics required to explain the experimental data.

Monte Carlo Simulation of Atomic Diffusion in
Si and GaAs via Vacancies and Interstitials

by

Uwe Schmid

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed November 17, 1988

Commencement June 1989

APPROVED:

Redacted for Privacy _____
PROFESSOR OF PHYSICS IN CHARGE OF MAJOR

Redacted for Privacy _____
Head of Department of Physics

Redacted for Privacy _____
Dean of Graduate School

Date thesis is presented November 17, 1988

Typed by Uwe Schmid for Uwe Schmid

ACKNOWLEDGMENT

This thesis is devoted to Ulrike for her love and support over thousands of miles during my entire stay in Oregon.

I am most grateful to Jim Van Vechten for giving me the opportunity to work on this challenging project, sharing his wisdom with me and for the excellent cooperation.

This project was considerably influenced by the excellent programming job of Nate Myers, who laid the foundations of this simulation with his work on vacancy migration.

I appreciate the kind help of Dick Arnold of the IBM PASC during my stay in Palo Alto, and Henri Jansen's perpetual interest in discussing computational problems in such depth.

Thanks are also due to the Fulbright commission for their scholarship.

Supported in part by the AFOSR and IBM.

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 Diamond and Zincblende Lattice	4
1.2 Point Defects in Si and GaAs	5
1.3 Concentration of Point Defects	8
1.4 Point Defect Models	9
2. ATOMIC DIFFUSION AND ITS MECHANISMS	11
2.1 Diffusion	11
2.2 Atomic Diffusion Theory	12
2.3 Simulated Diffusion Mechanisms	14
2.3.1 Vacancy Diffusion	15
2.3.2 Interstitial Diffusion Mechanisms	17
3. VIDSIM - THE MONTE CARLO SIMULATION PROGRAM	22
3.1 General Remarks	22
3.2 User Interface	26
3.2.1 Environment File	27
3.2.2 Defect Placement File	30
3.2.3 Occupant File	33
3.2.4 Inter-occupant File	35
3.3 The Simulation	37

3.4 Algorithm Checks	43
3.4.1 Random Number Generator	43
3.4.2 Random Walk	45
4. DIFFUSION OF GOLD INTO SILICON	49
4.1 Theoretical and Experimental Issues	49
4.2 Simulation Methods and Aspects	54
4.3 Results	60
4.3.1 Kick-out Mechanism	60
4.3.2 Frank-Turnbull Mechanism	75
4.4 Discussion	84
REFERENCES	88
APPENDICES	92

LIST OF FIGURES

1.1.	The diamond lattice.	5
1.2.	Simulated types of interstitials.	6
3.1.	Random walk by bond-centered interstitial.	47
3.2.	Random kick-out by bond-centered interstitial.	48
4.1.	Au concentration profiles of stimulated kick-out.	61
4.2.	Au concentration profile at $6.4 \cdot 10^{12}$ time units of stimulated kick-out.	64
4.3.	Stimulated kick-out: comparison with theory.	65
4.4.	Central Au concentration vs. time of stimulated kick-out.	67
4.5.	Au interstitial concentration profiles of spontaneous kick-out.	69
4.6.	Au concentration profiles of spontaneous kick-out.	70
4.7.	Au concentration profile at $3.8 \cdot 10^{12}$ time units of spontaneous kick-out.	72
4.8.	Spontaneous kick-out: comparison with theory.	73
4.9.	Central Au concentration vs. time of spontaneous kick-out.	74
4.10.	Vacancy concentration profile of spontaneous Frank-Turnbull.	77
4.11.	Au concentration profile at $5.3 \cdot 10^7$ time units of stimulated Frank-Turnbull.	80
4.12.	Central Au concentration vs. time of stimulated Frank-Turnbull.	81
4.13.	Au concentration profile at $8 \cdot 10^7$ time units of spontaneous Frank-Turnbull.	83

LIST OF TABLES

3.1.	VIDSIM keywords.	28
3.2.	Sample defect placement file.	32
3.3.	Sample occupant file.	34
3.4.	Sample inter-occupant file.	36
3.5.	The main simulation algorithm.	42
3.6.	Test of random number generators.	44
3.7.	Random walk check.	46
4.1.	Simulation energy parameters.	55
4.2.	χ^2 fit for exp() function of stimulated kick-out.	63
4.3.	χ^2 fit for exp() function of spontaneous kick-out.	71
A.1.	Properties of the diamond - and zincblende structure.	92
A.2.	Elements with the diamond - or zincblende structure.	92
A.3.	Calculated interstitial positions.	93
A.4.	Properties of the interstitial lattice.	94
A.5.	VIDSIM modules.	95

Monte Carlo Simulation of Atomic Diffusion in Si and GaAs via Vacancies and Interstitials

"The large amount of literature which is still devoted to diffusion in semiconductors is a very good illustration of the fact that nobody understands what is going on."¹

1. INTRODUCTION

Computational physics has become an important and exciting branch in modern science. The ability to model physical systems on computers is sometimes the only feasible approach when trying to solve complex systems such as many body systems or transport phenomena.

One of the most widely used techniques in computer simulation is the so-called "Monte Carlo method". It is based on the idea of creating statistical fluctuations with the help of "random numbers" and tracking the effects of this generated probability distribution on a complex system in both space and time.² Thus, one not only obtains detailed microscopic

information of the system to a degree which is inaccessible by other methods, but also insight into the problem.

This microscopic data can then be easily processed and compared to both existing theoretical models as well as to experimental results. Hence, it might help to clarify controversial research issues and perhaps serve as an impetus for new theoretical and experimental studies.

We hope that the Monte Carlo simulation program "VIDSIM", which will be introduced and described in chapter 3, will have exactly these stimulating effects on "the many puzzles and controversies that have been debated in the last fifteen years regarding the mechanisms of atomic diffusion in silicon"³ and GaAs.

It is evident, why the study of diffusion effects in Si and GaAs is technologically so important: dopant diffusion is used as an elementary process step in the fabrication of integrated circuits (IC's). Although silicon still has a predominant role as a base for IC's, the need for faster circuits seems to be better matched by III-IV compounds, such as GaAs.

In spite of its technological importance, our current knowledge of atomic diffusion is modest. At the present day, a huge variety of models exist which try to account for the

experimental diffusion data and reconcile them with other related data, such as impurity diffusion and quenching data.

Even the most fundamental questions concerning atomic diffusion, such as

- is the vacancy or the interstitial the dominant point defect which mediates atomic self diffusion at high temperatures ?
- what is the nature of these defects, are they simple or "extended" ?

are still a highly contentious matter.

With VIDSIM, we hope to provide a powerful and unbiased tool for the study of diffusion effects. The program supports various simple models and assumptions which can be "customized" by the user, allowing him (or her) to investigate the evolution of the diffusion process with the chosen defects, parameters and models.

As an application, we simulated a crucial and uttermost controversial experiment: the diffusion of gold in silicon. Various models were simulated and compared to the theories which try to explain the numerous experimental data. Details and results of this simulation can be found in chapter 4.

1.1 Diamond and Zincblende Lattice

The covalent crystals formed by the elements from column IV of the periodic table (C, Si, Ge) have structures in which each atom is tetrahedrally coordinated, corresponding to their sp^3 hybridized orbitals. The resulting lattice is called *diamond* or "A 4" structure. It can be regarded as two interpenetrating FCC lattices which are shifted by $(a/4)$ along the diagonal (Fig. 1.1). The coordination number is 4.

When the two interpenetrating FCC lattices are occupied by different atoms (such as in GaAs) the structure is called *zincblende* (sphalerite ZnS type), according to the German chemical term for ZnS, or "B 3" type.

Some essential properties of these two crystal structures are compiled in Table A.1. Elements of them are given in Table A.2.

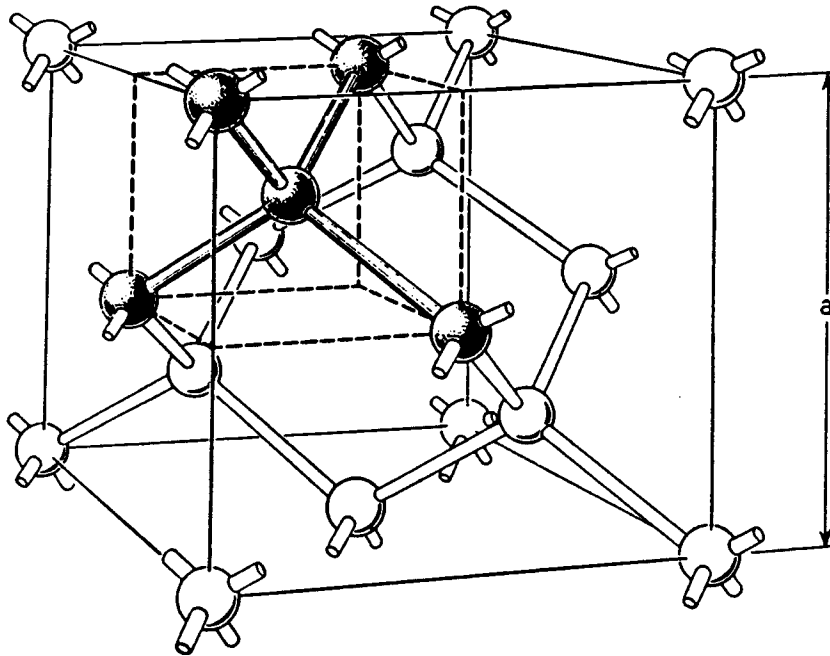


Fig. 1.1.
The diamond lattice
[from W. Shockley, *Electrons and Holes in Semiconductors* (Van Nostrand Reinhold, Princeton, New Jersey, 1950)].

1.2 Point Defects in Si and GaAs

Point defects can be subdivided into two categories: *native point defects* and *impurity related defects*; both are taken into account by VIDSIM.

Native point defects are structural, atomic imperfections. They govern self-diffusion in semi-conductors. There are three classes of interest:

(a) *Vacancy (V)*: a regular lattice site is empty. The missing atom migrated to the surface. This defect is also known as *Schottky defect*.

(b) *Interstitial (I)*: an atom is transferred from a lattice site to an interstitial position, a position not normally occupied by an atom. This is sometimes called a *Frenkel defect*. We will consider three possible interstitial positions of high symmetry in the Diamond and B 3 lattice (see Figure 1.2):

- (i) the tetrahedral,
- (ii) the hexagonal and
- (iii) the bond-centered interstitial.

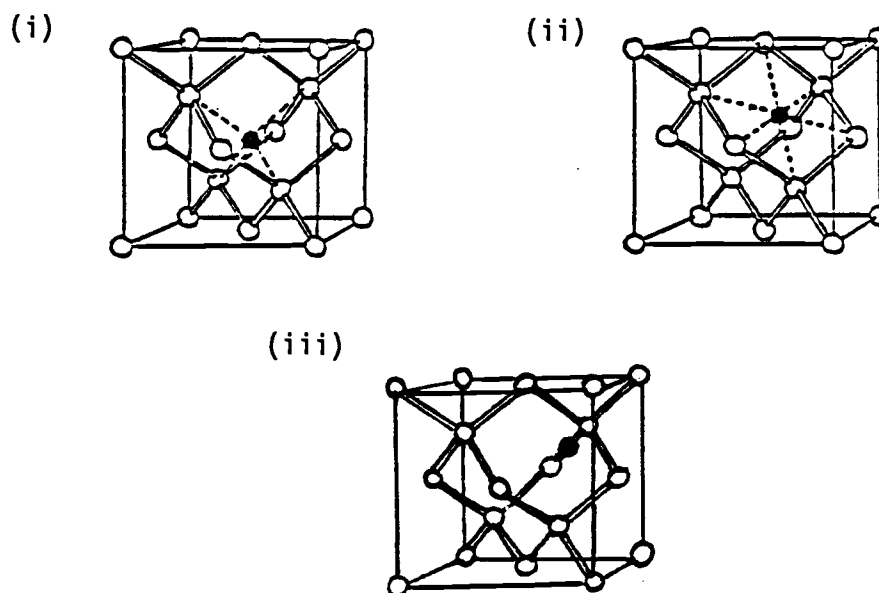


Fig. 1.2.
Simulated types of interstitials.

Their coordinates within the unit cell were calculated and are listed in Table A.3; some other geometrical properties are given in Table A.4.

For (i), an interstitial position was calculated by inverting a lattice point about its nearest lattice neighbor. The coordinates of hexagonal interstitials turned out to lie exactly in the middle between a lattice point and its third nearest neighbors at a certain distance ($\sqrt{11}a/4$). The positions of (iii) are trivial, of course, as they are right in between a lattice site and its nearest neighbor.

(c) *Antisite defect*: in a compound crystal, a site is occupied by the "wrong" atom, e. g. a Ga atom on an As site.

Impurities arise from the (deliberate) introduction of foreign atoms into the lattice, preferably Group III and IV elements. In this case, they are called *dopants*. There are two possibilities of incorporation:

(a) *substitution*, i. e. the impurity is located on a regular lattice site, and

(b) *interstitial incorporation*, i. e. the foreign atom is placed at an interstitial position. Again, the three positions mentioned above are possible.

The more complicated "*interstitialcy*" (Seitz, 1950)³, which forms bonds with the host, and the "*extended interstitial*" (Seeger and Chik, 1968)⁴ are no subject of this simulation.

1.3 Concentration of Point Defects

Point defects are present in the thermodynamic equilibrium at finite temperature due to the reduction of sample free energy that they produce by introducing entropy even at the cost of increasing the sample internal energy. During crystal growth or processing, the point defect concentration is often well above the equilibrium value. When the sample is suddenly heated, it may take time for the defect concentration to rise to the equilibrium value. Point defects must migrate into the bulk from sites, generally on the surface, where they are created.

In thermodynamic equilibrium, the number n_V of vacancies at a temperature T can be calculated for a crystal of N identical atoms by minimizing the sample free energy F , under the condition $V = \text{const}$. The calculations yields:

$$n_V = N \exp(-F_V^f/k_B T), \quad (1.1a)$$

$$F_V^f = E_V^f - T S_V^f, \quad (1.1b)$$

where E_V^f is the energy needed to form the vacancy, and S_V^f is the increase of entropy in formation.

Formulas identical to eqn. (1.1) hold true for other point defects, such as the interstitial, with the exception, of course, that E_V^f and S_V^f would be the energy and entropy of formation for the defect under consideration.

1.4 Point Defect Models

As shown above, there is always a certain concentration of interstitials C_I and vacancies C_V present in a crystal, which can be easily calculated for a specimen using eqn. (1.1a). Thus, the ratio of the two concentrations, C_V/C_I , determines which species dominates under equilibrium conditions. In a self-diffusion experiment, however, the measured quantity is:

$$D^{SD} = D_V C_V + D_I C_I , \quad (1.2)$$

where D_V and D_I are the diffusion constants (diffusivities) of the vacancies and interstitials, respectively (see chapter 2.1).

The following classifications for point-defect models are made in the literature⁶:

In the *pure vacancy models*⁷⁻⁹, it is asserted that $C_V \gg C_I$ and $D_V C_V \gg D_I C_I$, whereas in the *pure interstitial models*¹⁰, these inequalities have to be reversed. If the above terms are comparable, the corresponding models are called *combined vacancy and interstitials models*^{4,11}.

2. ATOMIC DIFFUSION AND ITS MECHANISMS

2.1 Diffusion

Mass transport in a crystal caused by the migration of point defects is called diffusion. The driving force behind this non-equilibrium process is a gradient in the concentration C of the particles. *Fick's first law* defines a diffusion coefficient D :

$$\mathbf{j} = - D \text{ grad } C , \quad (2.1)$$

where $\mathbf{j}(\mathbf{x})$ is the flux density and $C(\mathbf{x},t)$ the particle concentration. Eqn. (2.1) together with the *continuity equation*

$$\text{div } \mathbf{j} + \frac{\partial C}{\partial t} = 0 \quad (2.2)$$

yields *Fick's second law*:

$$\frac{\partial C}{\partial t} = D \nabla^2 C. \quad (2.3)$$

2.2 Atomic Diffusion Theory

Self-diffusion in a crystal is mediated through the migration of native point defects, in contrast to *impurity-diffusion*, which is a result of the motion of impurities. In the literature, two basic diffusion mechanisms are distinguished:

(a) *Vacancy diffusion*. An atom is hopping into an adjacent vacancy, leaving behind an unoccupied lattice site. Thus, the direction of the vacancy migration is opposite to the direction of the mass transport.

(b) *Interstitial diffusion*. An atom is exchanging interstitial sites.

The macroscopic diffusion constant can be related to the lattice properties using a model, in which the transition rate of an atom jumping to an adjacent position is determined by the frequency ν , with which the atom tries to overcome the energy barrier E_{act} , and the probability p that a fluctuation exists which provides the defect an energy greater than E_{act} . Thus, the atomic jump frequency Γ is

$$\Gamma = \nu p = \nu \exp(-E_{act}/k_B T). \quad (2.4)$$

The attempt frequency ν is about the highest frequency of lattice vibrations and can be approximated by the Debye frequency, which is about $1.35 \cdot 10^{13}$ for Si. Together with the density

gradient of defects at a separation distance a , a simple calculation yields:

$$j = - g R a^2 \Gamma \text{grad } C, \quad (2.5)$$

where g is a geometrical factor which is determined by the structure of the crystal. For the diamond lattice g was found to be 0.5. R is the so-called correlation factor and differs from unity, if a hop sets up a geometric configuration in a way, that the second hop is not independent of the first one. The two subsequent hops are said to be correlated.

From a comparison with eqn. (2.1), together with eqn. (1.1), we obtain the Arrhenius equation for the diffusion coefficient:

$$D = D_0 \exp(-Q/k_B T), \quad (2.6a)$$

with the total activation energy

$$Q = E_{act} + E_X^f \quad (2.6b)$$

and the preexponential factor

$$D_0 = g R a^2 \nu \exp(S_X^f/k_B), \quad (2.6c)$$

with the index $x = V$ denoting a simple vacancy mechanism, and $x = I$ an interstitial mechanism. Thus we have a strong exponential increase of D with the temperature T . Measurements of the temperature dependence are usually presented by a logarithmic Arrhenius plot. One of the most acute problems in the study of diffusion is, however, that the slope Q of such a plot does not yield separate information on E_{act} and E_x^f . Some researchers have such strong feelings about this problem, that they would quit the diffusion business as soon as this riddle is solved¹².

2.3 Simulated Diffusion Mechanisms

As pointed out in the last section (compare eqn. (2.4)), the probability p for a fluctuation greater than E_{act} , which is of course the probability for an event to occur, is one of the most important terms for the diffusion process. Our "Vacancy and Interstitial Diffusion Simulator" program (VIDSIM) is based on calculating p for each mobile defect in the simulated sample.

To accomplish that, the activation energy for each defect has to be given. The program computes E_{act} from a set of parameters, which the user has to "feed" to the program. The way this is done and the syntax which has to be followed is described in chapter 3. In the following section, we want to introduce the

simulated diffusion mechanisms and describe, how the activation energy is calculated for each of them.

During a simulation, a net flux of defect migration, which is proportional to eqn. (2.5), will occur. The geometrical factor g and the separation distance a are automatically taken into account to during the course of the simulation due to the geometric constellation. Thus, with the known Debye frequency for a given crystal, the atomic jump frequency Γ and j can be calculated.

2.3.1 Vacancy Diffusion

(a) Vacancy nearest neighbor hopping (NNH)

This mechanism was already explained above. In the simulation, we take the activation energy for NNH to be parametrized in the following form:

$$E_{act}^V = E_{kin} + non-add \sum E_j^{bond} + bond-add_j \quad (2.7)$$

where

- E_{kin} is the kinetic energy of the atom to hop into the vacancy. In the "*Ballistic Model*" (Van Vechten, 1975)¹³⁻¹⁸, this

is the threshold for the hop into a given zone-boundary cycle. For other models, E_{kin} simply designates the main activation energy parameter.

- $non-add$ is a parameter to adjust the variation of the bond strength, i. e. it allows for deviations from the bond-additivity approximation. The default value for it is 1.0.

- $\sum E_j^{bond}$ is the sum of bond energies between the atom which is to hop into the vacant site and all of its atoms to which it is bonded, i. e. its nearest neighbors. Thus, it is necessary for the simulation to be provided with a list of all bond energies between every possible pair of occupant types.

- $bond-add_j$ is an adjustment for vacancy neighbors. The j^{th} component of the bond-additivity vector is used, if the atom that hops has j vacancy neighbors ($0 \leq j \leq 4$). This parameter can be useful to take divacancy binding energies into account. $(0, 0, 0, 0, 0)$ is default.

(b) Vacancy second nearest neighbor hopping (SNNH)

In this case, no antisite defects will be created. Apart from an additional factor $expan_{SNNH}$ in front of E_{kin} , which is an energy adjustment for the longer hop, the activation energy looks exactly as described in eqn. (2.7). In the *Ballistic Model*, $expan_{SNNH}$ is a simple geometrical factor of 8/3, which is also the default value.

(c) Divacancy NNH and SNNH

These mechanisms have been proposed in the literature^{19, 20}. No special arrangements were provided to simulate them individually, as they are already considered by the *bond-add_j* parameter in eqn. (2.7) and by the decreasing activation energy, if a vacancy is a (second) nearest neighbor of the atom to hop, provided that the bond energy between this atom and the vacancy is chosen appropriately.

2.3.2 Interstitial Diffusion Mechanisms

Apart from simple, direct interstitial diffusion, there are two more complicated mechanisms.

(a) Frank-Turnbull (dissociative diffusion) mechanism

The following reaction between an atom X on an interstitial site X_I , a vacancy V and an atom X on a substitutional site X_S was originally discussed by Frank and Turnbull²¹:



From the point of view of the interstitial X_I , the situation looks as follows: it migrates via different mechanisms through the crystal, until a vacancy is its nearest neighbor. Then it needs the activation energy

$$E_{act}^{FT} = E_{kin}' + \sum E_i^{bind} \quad (2.9)$$

to jump into the vacancy and creates either

- a substitutional site X_S , if X is a dopant
- an antisite defect, if X is one of the host elements of a compound on the "wrong" sublattice, or simply
- an ordinary lattice site, if X is either a constituting element of the compound on the right sublattice or the one of an elemental crystal.

- E_{kin}' in eqn. (2.9) is the kinetic (or dominant) energy parameter and has to be supplied for every interstitial type of every occupant.

- $\sum E_i^{bind}$ is the sum of pairwise energy terms (interstitial binding energy) between the interstitial under consideration and its nearest lattice neighbors. This parameter allows for the variation of E_{act}^{FT} , when the interstitial has lattice neighbors of different types.

If reaction (2.8) proceeds in the other (\leftarrow) direction, things look different. In this case, we have to consider an atom X on a lattice site. Again, there are two possibilities:

(i) X is on a regular site. That means, that there are almost as many candidates for this reaction in the crystal as there are regular sites (N). Under the assumption that the number of impurities and possible antisites in the crystal is small compared to N (approx. $10^2 : 10^{12}$ for a normal simulation), we can neglect them. Then, the total probability $p_{tot,r}^{ivf}$ that any of the regular sites forms an interstitial-vacancy pair according to the reverse Frank-Turnbull mechanism (2.8), is $N/2$ times the probability that it is formed by an atom on the A sublattice, plus $N/2$ times for an atom on the B sublattice. The activation energy for such a single process can be described by:

$$E_{act}^{ivf} = E_f , \quad (2.10)$$

with E_f being the energy of formation parameter of the occupant under consideration. In case we have a diamond lattice, then we can think of it as a zincblende structure with the same occupant on the A and B sublattice.

(ii) X is an impurity or a possible antisite. As the jump probability depends exponentially on the activation energy, this can not be neglected. Even for a small concentration of impurities, the sum of the hopping probabilities over all impurities $p_{tot,i}^{ivf}$ can exceed $p_{tot,r}^{ivf}$. The probability for

such an inverse Frank-Turnbull mechanism has to be calculated for each impurity and antisite, and the activation energy corresponds (2.10), with, however, different E_f for different occupants!

The total probability for such an event anywhere in the sample (as a sum of $p_{tot,r}^{ivf}$ and $p_{tot,i}^{ivf}$) is calculated at the beginning of the simulation and has to be adjusted only, if impurities or antisites are created or annihilated.

(b) Kick-out mechanism

This mechanism was first proposed by Seeger and co-workers (for a detailed discussion see ⁶, ²²). Its reaction equation is:



Similar to (2.8) in the '--->' direction, an interstitial X_i hops onto a lattice site, but this time the site is occupied by the substitutional atom Y_S , and not by a vacancy. The kicked out occupant is displaced to the interstitial position Y_i , its original position is taken over by X_S . This is a symmetric reaction which involves the kinetic energy E_{kin}' of X , the interstitial formation energy for Y_I , the energy gain of the transfer of X_i onto a substitutional site $-E_f^X$, and the breaking

of all bonds around Y_S with the energy contribution $\sum E_i^{bond,Y}$; thus the activation energy yields:

$$E_{act}^{KO} = E_{kin}^{X'} + E_f^Y - E_f^X + \sum E_i^{bond,Y} . \quad (2.12)$$

(c) Direct interstitial-interstitial mechanism

This is the basic diffusion mechanism mentioned above. In this case, the direct migration from one interstitial position to the other does not involve any exchange reactions. Thus, we have one simple parameter for this mechanism, the interstitial migration energy:

$$E_{act}^{DII} = E_{mig} . \quad (2.13)$$

3. VIDSIM - THE MONTE CARLO SIMULATION PROGRAM

3.1 General Remarks

VIDSIM is a general purpose program for the Monte Carlo simulation of the atomic diffusion of point defects in diamond and zincblende lattices via vacancies and interstitials. The program was developed on an IBM AT clone in the C language, but great care was taken to make the program as portable as possible. In fact, **VIDSIM** has already been ported to and ran successfully on other systems, such as UNIX systems with the pcc compiler and the IBM 3090 supercomputer of the IBM Palo Alto Scientific Center.

In order to run the program on personal computers such as the IBM PS/2, AT and compatibles, and to overcome the memory restrictions which apply to MS-DOS, extensive use of dynamic memory allocation during run-time has been made. As a result of this programming technique, **VIDSIM** will require only as much memory as absolutely necessary to simulate the current defect configuration, which is determined by the number of defects, the types of interstitials and the diffusion mechanisms being simulated. To accomplish these requirements together with a high efficiency in speed, which is of course an imperative for numerical intensive calculations, full advantage of the pointer concepts available in the C language has been taken. In fact, there are such concepts as pointers to dynamically allocated

arrays of pointers, whose elements again point to structures, which are dynamically allocated, too. This approach has been chosen for two reasons: although it sounds very complicated, it is one of the fastest things that can be done in 'C', and it also circumvents the use of static arrays, which would have prevented the programming of this software on personal computers. As a result of this technique, a representative simulation will request only about 120-200 KBytes of RAM, which allows to run the simulation continually in the background with one of the multi-tasking programs available for MS-DOS.

As **VIDSIM** performs a lot of floating point arithmetic, it is advised to use the program only on microcomputers which are equipped with a math-coprocessor.

VIDSIM is programmed in a modular way, which means that procedures which act on logically related components of the program are combined in one file. The main diffusion simulator program consists of about 6200 lines of source code (~163 Kbytes), which are split up into 15 modules (source files), and another 800 lines of data declarations (~23 Kbytes) in 11 so-called "header"-files. The modules and their function are shown in Table A.5. In addition, there is a variety of utility programs for processing data, sorting the defect files and producing profiles. The present author is responsible for about half of this code, expanding an old version of the program which basically performed

vacancy migration. The old version was written by Nathan C. Myers, whose superb programming job exerted a considerable influence on the code that followed.

For the program development, different compilers and integrated development environments were used. The most satisfying results were obtained with the *Microsoft C 5.1 Optimizing Compiler*, both as far as the efficiency of the produced code is concerned (it was 25 - 45 % faster than the rest), and the capacity of its debugger.

Depending on the size of the sample and the simulated diffusion mechanisms, the simulation runs at a rate of about 40 to 100 events (hops) per second in real (computer) time on an IBM AT compatible (12 MHz), equipped with a 80287 math co-processor. After a run, this rate is displayed, together with a small event statistics, which shows how many of each hopping mechanisms has been performed.

The dimensions of the simulated sample, which can be set by the user, are limited by the "natural" integer representation of the compiler and the computer, respectively. If the integer type occupies N bits, then the maximum dimension (in one direction) will be 2^{N-3} . As there are 8 atoms in a unit cell of a unit volume, this restricts the number of the atoms in the simulated

crystal to 2^{3N-6} . For a 16 bit machine, this is approximately $4.4 \cdot 10^{12}$.

As a practical limit for the mobile defects, fewer than 200 vacancies and interstitials and less than 1000 defects overall may be treated with the IBM AT compatible implementation.

The program can also be run under different 'debug'-levels to provide various degrees of information about the calculated properties of the simulation and the assumptions it makes. In the highest debug-level (3), the maximum of information is displayed, including the hopping probabilities of each single mobile defect, followed by the hop which was finally performed. This is especially useful to check the assumptions one made during the start phase of a planned simulation. The logical step is then to reduce the debug level to 1, which still gives the run-time statistics.

For each occupant, one of the three types of interstitials (as described in chapter 1.2) may be selected for the simulation. Thus, in a GaAs crystal it would be possible to simulate bond-centered Ga and hexagonal As interstitials, for example, disregarding the impurities.

There is also the concept of 'computational periodic boundary conditions' which has to be explained: if a mobile defect

migrates beyond a surface in one direction, it will appear again on the other side of the crystal. This is very useful for excluding surface effects and conserves the number of mobile defects.

3.2 User Interface

VIDSIM is fundamentally controlled by the contents of its *input files*, which are:

- environment file
- occupant file (occ-file)
- inter-occupant file (ocx-file)
- defect placement file (dfp-file).

It will write the *dfp-out-file*. The function and syntax of each of these files will be described below.

For the syntax description throughout this chapter the **EBNF**-notation (Extended Bacus-Naur Form) is used, in which "|" separates alternatives, "[" and "]" surround optional parts and "{" and "}" enclose optional parts, which can be used multiply. A ";" finishes a rule, and reserved words are written in bold.

3.2.1 Environment File

This is the main control file of the simulation. It contains the names of all the other files and controls the migration mechanisms and various modes of the program, such as the conditions for terminating the simulation. In addition, some decisive simulation-constants, as for example the temperature, are defined in this file. The name of the environment file has to be specified as an argument in the command line when starting the program:

```
C:\SIM> vidsim test.env
```

starts the simulation with *test.env* as its environment-file.

The syntax of the environment file in EBNF notation is:

```
env-file:  
    { comment-line | keyword {arguments} };
```

A *comment-line* is any line that starts with the '#'-character. Table 3.1 lists all keywords which are defined in VIDSIM version 2.1; their numerical arguments are given with default values which are set, if the corresponding line is commented out.

Table 3.1.
VIDSIM keywords.

vacancy-simulation	
kick-out	
Frank-Turnbull	
boundaries	4000 4000 4000
real-timelimit	86400.0
virtual-timelimit	0.0
eventlimit	10000
debug	1
occupant-file	test.occ
occx-file	test.ocx
def-in-file	test_in.dfp
def-out-file	test_out.dfp
temperature	273.15
bond-additivity	0.0 0.0 0.0 0.0 0.0
non-additivity	0.0
numdefects	1297
compound	
2NN-hopping	
randseed	385047
snapshot	10000
total-time	0.0

The first three keywords (*vacancy-simulation*, *kick-out* and *Frank-Turnbull*) select the migration mechanisms which are to be simulated. If commented out, they are not in effect.

boundaries ix iy iz sets the periodic boundaries of the simulated crystal to $(-ix \leq x < ix)$ etc. in units of the lattice constant. If omitted, the maximum possible values (depending on the integer representation of the compiler) are used.

real-timelimit, *virtual-timelimit* and *eventlimit* set the conditions (upper limits) for terminating the program.

Setting a limit to zero imposes no restriction with respect to this parameter. Thus, in our example (Table 3.1), the simulation will run until 10000 events will have been simulated or the real (computer) time exceeds one day (86400 seconds), whichever conditions occurs first. There is no restriction with regard to the virtual time.

debug i ($0 \leq i \leq 3$) controls the amount of information displayed by the program, as explained above.

occupant-file and *occx-file* take the name of the files, which are to be used as occupant and inter-occupant files, as their argument. If necessary, the full path can be given.

def-in-file and *def-out-file* specify the defect placement files to read from (input file) and to write to (output file).

temperature x is the simulation temperature in degrees Kelvin.

bond-additivity and *non-additivity* are the parameters for vacancy simulation as used in eqn. (2.7).

numdefects i with *i* a prime number: size of the defect hash-table, which should be greater than the maximum number of defects expected.

2NN-hopping has to be set, if SNNH for vacancy migration is desired.

randseed i is the seed for the random number generator. Useful, if the computations should be repeatable, as a random seed which depends on the system time is used, if this parameter is not set.

snapshot i is the number of events after which the defect configuration is written into the *def-out-file* to ensure that a interruption of the simulation won't result in a total loss of data.

total-time x is the sum of the virtual time of previous simulations. The virtual simulation time of the current run will be added to this value and will be written into the *def-out-file*, from where it will be added up consecutively.

3.2.2 Defect Placement File

The defect placement files contain all the defects and their coordinates of a simulation. The data in the *def-in-file* defines the initial defect configuration of the simulated crystal, whereas the configuration as calculated during the course of a simulation

will be written (after each snapshot) to the *def-out-file* (note that these two files have to be itemized in the *environment-file*).

The syntax is:

```

dfp-file:
    { comment-line | defect-line };

defect-line:
    defect coord coord coord [tag];

defect:
    substitutional | mobile;

substitutional:
    occupant;

mobile:
    interstitial | vac;

interstitial:
    occupant separator i-type;

i-type:
    tetra | hex | bc;

separator:
    -;

coord:
    float;

tag:
    short integer;

```

Again, a *comment-line* starts with the pound-character ('#'). In the case of the *def-out-file*, comments are created by the program to identify some of the conditions under which the data was generated, the virtual time it needed for this single run and the total amount of virtual time (as the sum of previous runs). In

debug level 1 or higher, it will also write some event statistics in this file.

A *defect-line* consists of the identification for the defect, its cartesian coordinates (in units of the lattice constant a), and an optional tag (an integer between ± 32000) to keep track of it.

Table 3.2 shows a sample file. It is a good illustration of the rule for specifying mobiles, which are, in this case, *si* and *au* interstitials of the type (*i-type*) *bc* (bond-centered).

Table 3.2.
Sample defect placement file.

```
# Sun Aug 28 16:08:11 1988 seed:49669508
# in:kodiff2.dfp
# temperature:1368.5 events:7710000 time:8.45166e+010
# Event statistics:
#   interst.-interst.-hops: 7704305 (99.93 %)
#   kick-out: 5695 ( 0.07 %)
# total-time: 1.34175e+011
au      -999.000    -96.000    96.000
au       999.500     60.500   -36.000
vac     -976.000    -26.500   -33.500
vac      980.500    -82.000   -29.500
si-bc    981.625     44.875    -6.625    100
si-bc    981.125     77.625     37.625   -100
au-bc    985.875    -35.375   -47.625
au-bc   -971.375     54.625   -43.875
au-bc    975.625     95.125    41.625
```

The coordinates of the different defect types and the syntax will be checked and any inconsistencies will be reported.

3.2.3 Occupant File

The occupant file contains all data about each particular type of defect. Each line consists of several fields:

```

occ-file:
    { comment-line | occ-line [i-spec-line] };

occ-line:
    lattice-occ sublat display-symbol [ $E_{kin}$ ];

i-spec-line
    i-type display-symbol  $E_{kin}$   $E_f$  [ $E_{mig}$ ];

lattice-occ:
    occupant | vac;

occupant:
    char { char };

display-symbol:
    char;

sublat:
    A | B | X;

```

For each occupant, there is an obligatory occupant-line, followed by an optional interstitial-specification-line. Table 3.3 shows a sample file.

Table 3.3.
Sample occupant file.

```
# au_diff.occ: list of occ. & energies
#               (all energies in eV)
#
vac  X . 1.0
si   X . 1.2
      bc . 0.40 4.4 0.40
au   X . 2.5
      bc . 0.39 2.5 0.39
```

The *occupant*-name (any string can be used) is the same one as used in the *dfp-files*. The sublattice field indicates, that the following data applies to the named defect in the *A* sublattice, the *B* sublattice or both (*X*), respectively. If the *compound* flag of the *env-file* is disabled, then any data specified for the *A* sublattice is used for the *B* sublattice, and vice verse.

The display symbol is not yet used, but it is supported already for a planned graphical representation of the defects.

The E_{kin} parameter has to be specified only, if vacancy migration is to be simulated (*vacancy-simulation* flag must be set!); it is the energy needed in eqn. (2.7), and has to be specified in eV.

The order of entries is important: if vacancies occur in the bulk, then the *vac* entry has to be the first one. In an elemental crystal, the next entry defines the natural site occupant, whereas

in a compound, the two following entries classify, respectively, the *A* and *B* sublattice occupants. In either case, subsequent entries define the properties of impurities.

If a *i-spec-line* is encountered (which begins with an interstitial-type keyword), then the specified interstitial of the *occ-line* immediately preceding it will be simulated. In our example (see Table 3.3), the diffusion of bond-centered *si* and bond-centered *au* will be taken into account. Remember, that for each occupant there may be one interstitial type, although there may be different interstitial types for different occupants.

This time, the *display-symbol* is followed by E_{kin}' and E_f , which are the parameters as requested by eqns. (2.10) and (2.12). If the optional inter-interstitial migration energy E_{mig} (compare eqn. (2.13)) is declared, then this means that this type of interstitial-interstitial migration is allowed! E_{mig} acts quasi like a switch to enable/suppress this mechanism.

3.2.4 Inter-occupant File

The inter-occupant file describes the interaction between the different occupant. It contains

(a) the bond-energy E^{bond} (compare eqn. (2.7)) between every possible pair of occupant types

(b) the interstitial-lattice binding energy E^{bind} (see eqn. (2.9)) between every interstitial type of one species and all other species.

Table 3.4.
Sample inter-occupant file.

```
# au diff.ocx:
# all energies in eV!
# a)
si vac -1.0
au vac 0.0
si si 0.18
si au 0.0
au au 0.029
# b)
si-bc si 0.00
au-bc si 0.00
si-bc au 0.00
au-bc au 0.00
au-bc vac 0.00
si-bc vac 0.00
```

Table 3.4 shows a sample file. The corresponding syntax is:

```
occx-file:
    { comment-line | bond-line | bind-line };

bond-line:
    lattice-occ lattice-occ Ebond;

bind-line:
    interstitial lattice-occ Ebind;
```

The *bind-line* is only necessary, if the *Frank-Turnbull* switch is set, whereas the bond energies are required for the vacancy migration and the kick-out mechanism. Again, the file is checked for complete information, and any discrepancy is reported.

3.3 The Simulation

VIDSIM is started with the environment file as an argument. From the data contained in this file, it will configure the simulation according to the desired mechanisms and modes. Then it will continue the setup with the occupants and interstitial types and the pertaining energy parameters, as specified in the occ- and ocx-files. The next step is the processing of the *def-in-file*, which represents the initial defect configuration of the sample. All mobile defects, such as interstitials and, if desired, vacancies, are extracted and put together in a mobile list. Then, all information necessary for the neighbors of each of its elements is gathered and calculated. An interstitial, for example, has to "know" its nearest lattice neighbors and the nearest lattice neighbors of each of them, so that for the kick-out mechanism the bond energy terms of the kicked out site (eqn. (2.12)) can be calculated quickly. Thus, depending on the simulation configuration, a lot of complex data has to be processed.

All the defects, the mobile and static ones, are entered into the defect list. As it is important to decide, whether a specific site is already occupied or not, a hash-table has been implemented to ensure a fast look-up.

Now the genuine simulation can start. One main part of it is the calculation of the probability p_{tot} , that any of the mobile defects will hop, which is of course the sum of the hop probabilities of each mobile. To accomplish this, VIDSIM will go through the mobile list and check, whether the hop probability for each mobile has already been calculated and whether it is still valid. If not, it will recalculate this value, depending on the mobile, its vicinity, and the simulation configuration. If the mobile defect under consideration is a vacancy, the bond energies of all its neighbors have to be added up to get the activation energy as in eqn. (2.7).

In case we have an interstitial, things become a little more complicated. First of all, we have to consider the direct interstitial-interstitial migration, if allowed for this type, and calculate the probability for a hop of the interstitial into any of its nearest interstitial neighbors. Then, we have to take care of the interstitial-lattice hops, scanning all the direct lattice neighbors of the interstitial and calculate all the pertaining probabilities. If the lattice neighbor under consideration is a

vacancy, then this is a possible Frank-Turnbull mechanism, as given in eqns. (2.8) and (2.9). If the lattice site is unoccupied, however, we have to deal with the kick-out-mechanism (eqn. (2.11)), using the activation energy (2.12).

These computations are done for the whole mobile list, and the values are added to p_{tot} . Last, but not least, we still have to consider the reverse Frank-Turnbull mechanism (if operative). This is done by the methods described in chapter 2.3.2. To cover all impurities in the crystal, the defect list is processed at the start of the simulation, all impurities and antisites are extracted and $p_{tot,i}^{ivf}$ is calculated. During run-time, it only has to be updated when certain events occur.

Finally, we calculated p_{tot} , the total probability that any of the hops will occur. The average time until this happens, is then

$$\langle T \rangle = 1/p_{tot}. \quad (3.1)$$

We call this "*virtual time*", as it is proportional to the actual time the experiment would take in a laboratory to get this "real" time, we have to divide $\langle T \rangle$ by the Debye frequency of the crystal, as we conclude from eqn. (2.4). Thus, the natural time unit of our simulation is one phonon period.

To decide, which event (i.e. diffusion process performed by a mobile defect in a certain direction) has to occur after $\langle T \rangle$, we eventually use the random number generator, which provides a random number r in the range $(0 \leq r < 1)$. The quantity

$$p_{critical} = p_{tot} * r \quad (3.2)$$

determines, which event is to take place. This is accomplished by adding up all partial probabilities and subprobabilities, until the sum becomes greater than $p_{critical}$. The event with the probability which was added up last is the chosen one.

Then, this event has to be carried out. The affected defect(s) will be moved, and the mobile list updated, resulting in a recalculation only of the hop probabilities of the defects, which are located in the vicinity of the moving atom. This sounds very simple, but it becomes quite complicated, if coded in an efficient way, as some kind of indirect reference tables have to be installed for the following problem:

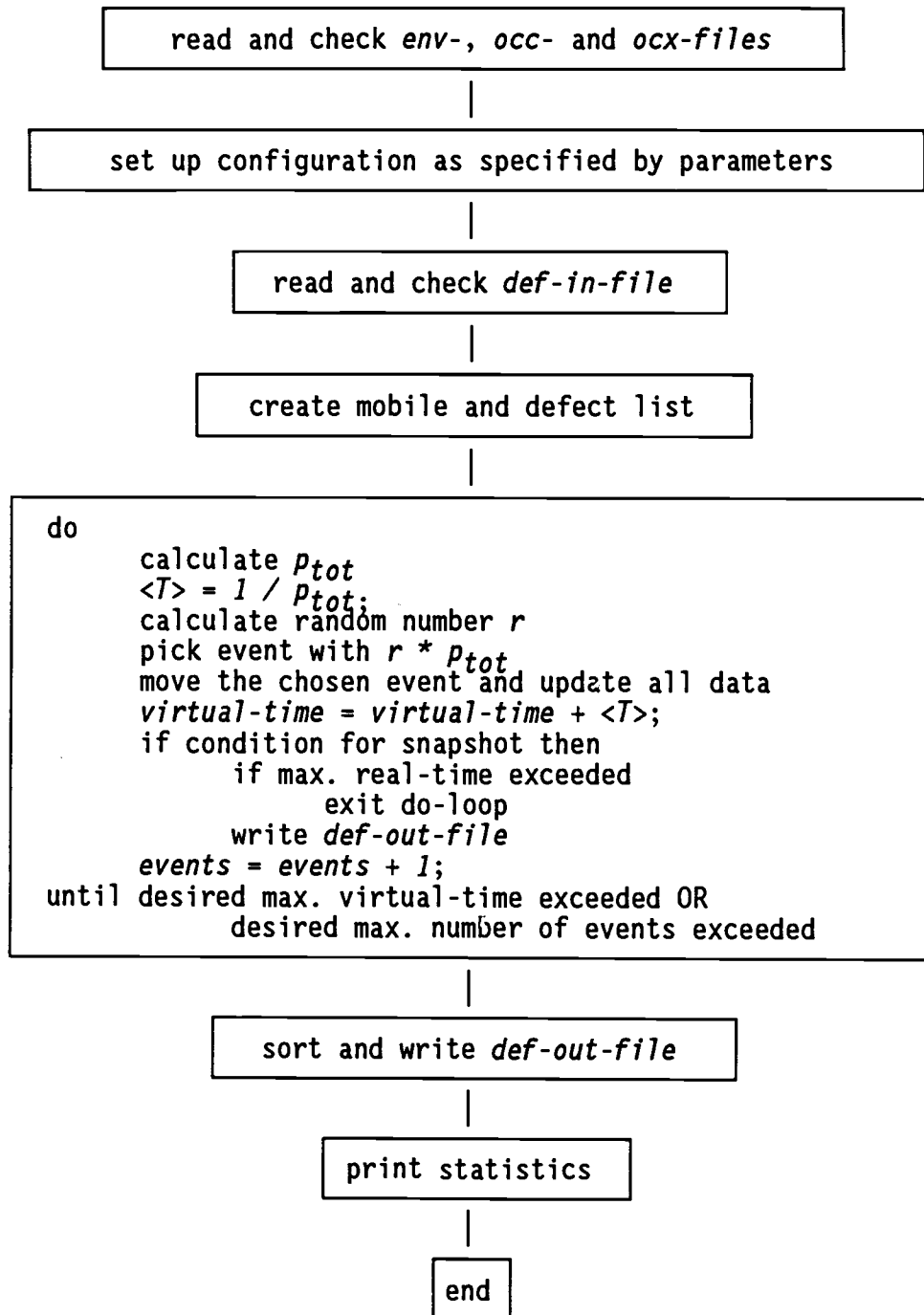
If a mobile X moved, then tell its neighbor Y of the direction d , that its direct neighbor in the reverse direction (X) has changed.

This is still complicated by the variety of mobile defects such as three types of interstitials, the geometry of the zincblende structure and the generality of the program.

A very condensed form of this algorithm is shown in Table 3.5. As we can see from it, the migration continues until one of the maximum parameters for the simulation, as defined in the environment file, is exceeded.

Thus the result after a certain number of events, will be a file with the updated defects and their coordinates and the virtual time in which the simulation was performed. Thus, we have information about the diffusion on a microscopic level, both in space and time.

Table 3.5. The main simulation algorithm.



3.4 Algorithm Checks

3.4.1 Random Number Generator

As we have seen in the last section, the simulation relies severely on the random number generator, which have to be excellent. In **VIDSIM**, there is two random number generators available: the original one (*drand48*) is working on the machine level by shifting bits and is adopted from UNIX System V.2. The second was added after some difficulties with *drand48* on the IBM 3090. It was decided to take a nighly portable and first class generator: *ranl*²³. It is based on the linear congruential method²³, which provides a practical infinite period and no sensible correlation.

To check the "quality" of these generators, two tests have been performed. The first involves the calculation of moments l ($l = 1, 2, 3, 4$) of the distribution:

$$s_l = (1/N) \sum r_i^l; \quad (3.3a)$$

and the second the correlation:

$$c_l = (1/N) \sum r_i r_{i+l}, \quad (3.3b)$$

where i runs from 1 to N . In our case, we took $N = 10^4$. This test was done for 10 different seeds. The relative deviations from the theoretically expected values,

$$s_j^{th} = 1/(1 + 1), \quad (3.4a)$$

$$c_j^{th} = 0.25, \quad (3.4b)$$

averaged over the 10 different seeds, are listed in Table 3.6. The occurring errors exceed the calculated ones only a few times by a minor amount, which can easily be explained by the finite accuracy of the digital representation of floating point numbers in a computer. The results of *ran1* seem to be a little bit better; both random number generators, however, gave extremely satisfying results.

Table 3.6.

Test of random number generators.

Shown are the deviations from the theoretical values.

1	rand		drand48		calculated	
	<Ds1>	<Dc1>	<Ds1>	<Dc1>	<Ds1>	<Dc1>
1	0.27%	0.60%	0.35%	0.70%	0.21%	0.42%
2	0.40%	0.47%	0.57%	0.71%	0.42%	0.42%
3	0.47%	0.58%	0.77%	0.66%	0.63%	0.42%
4	0.54%	0.61%	0.96%	0.78%	0.84%	0.42%

3.4.2 Random Walk

Another standard check is the random walk problem. A particle that performs n steps of the same length d in random directions will - according to the central limit theorem of Gauss - eventually move by the distance

$$D_n = \sqrt{n} d. \quad (3.5)$$

As this is a wildly fluctuating quantity, we decided to take advantage of some statistics and average over $N-n$ steps, where N is the maximum number of steps taken, and plot the following quantity:

$$R_n = \frac{1}{N-n} \sum_{i=1}^{N-n} |d_i - d_{i+n}| = \sqrt{n} d, \quad (3.6)$$

where d_i is the position of the atom after i hops.

This check was performed for $N = 5 \cdot 10^5$ and R_n plotted against \sqrt{n} . The results of the *interstitial-interstitial hop* (bond-centered type) are shown in Fig. 3.1. As expected, we have an almost straight line through the origin. A linear correlation calculation - forced through the origin - yielded 0.3597 for the slope, which deviates by only 1.7% from the 'true' value, the

distance of nearest bond-centered neighbors $d = \sqrt{2}/4$ (rf. Table A.4). The correlation coefficient R was found to be 0.9975.

Table 3.7.
Random walk check.

	direct inter- stitial hop	kick-out mechanism
slope	0.3597	0.3115
d	0.3536	0.3062
dev.	1.7%	1.7%
R	0.9975	0.9991

For the kick-out mechanism, the results are presented in Fig. 3.2. In Table 3.7, the calculated slope was compared to $\sqrt{2}$ times the distance of the bond-centered interstitial to its nearest lattice neighbor ($\sqrt{3}/8$). Again, very good accordance with the theoretical value was achieved.

The same test were done for the tetrahedral and hexagonal interstitials, and similar results were obtained.

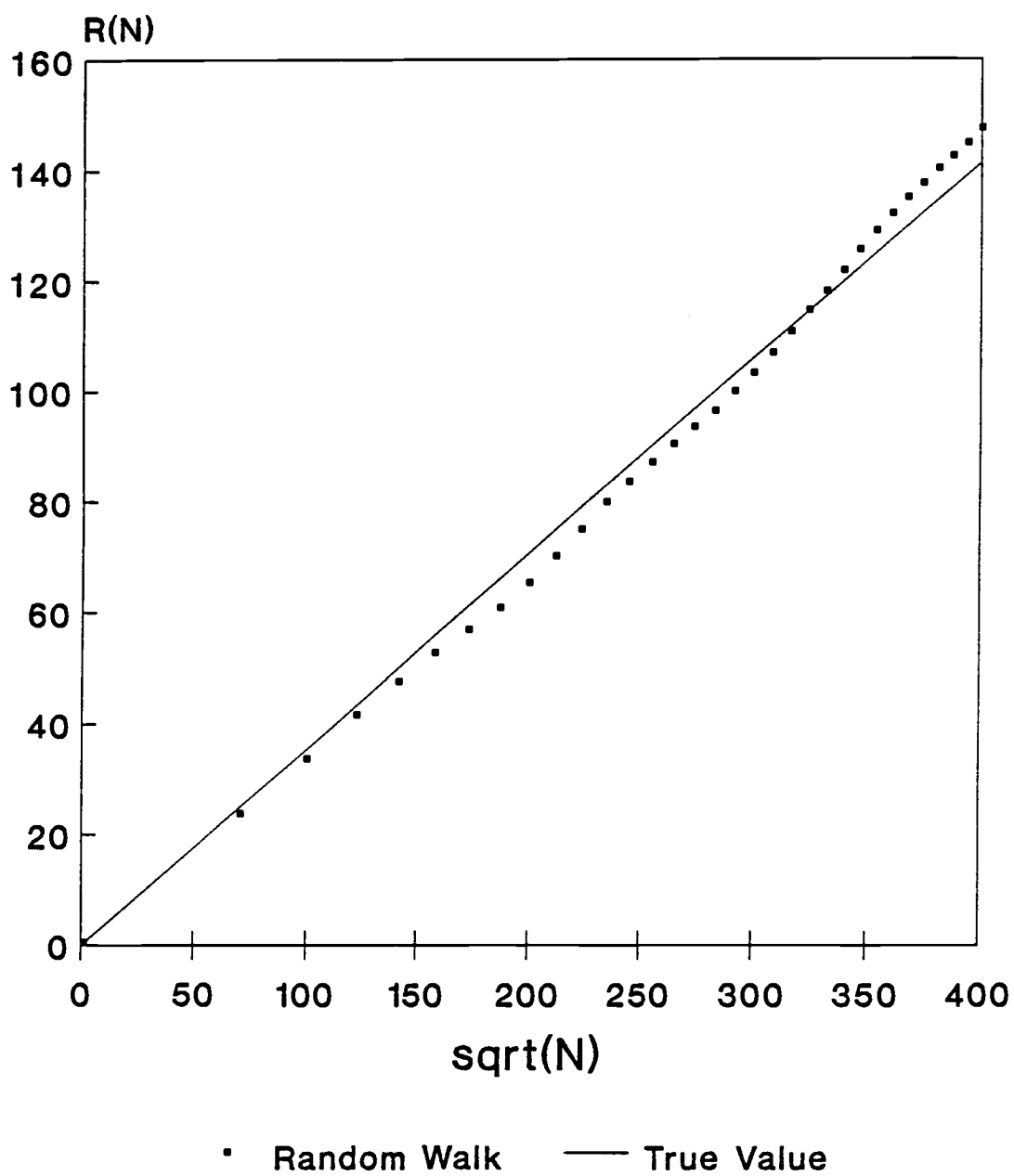


Fig. 3.1.
Random walk by bond-centered interstitial.

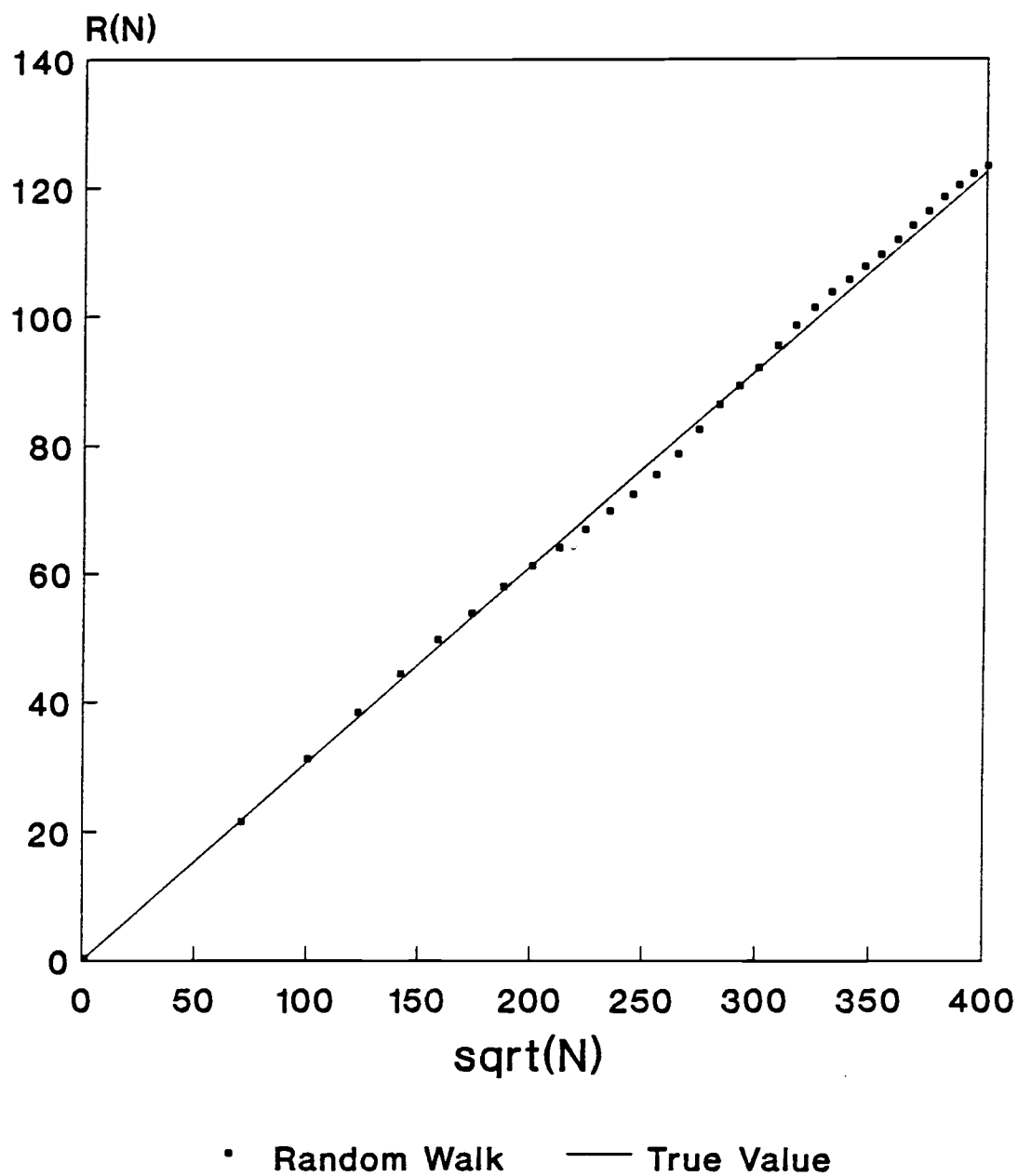


Fig. 3.2.
Random kick-out by bond-centered interstitial.

4. DIFFUSION OF GOLD INTO SILICON

4.1 Theoretical and Experimental Issues

Extensive literature exists on gold diffusion into silicon due to its considerable technological interest. Gold is used in high voltage silicon power devices to control the charge-carrier lifetime²⁴.

Under thermal equilibrium conditions, gold is dissolved in silicon either on substitutional (Au_S) or interstitial sites (Au_I). From experimental data^{25,26} it has been concluded for a long time that the solubility of Au_S is much higher than that of Au_I and that Au_S diffuses much slower than Au_I . It is also known that there is an exchange of Au between interstitial and substitutional sites, so that the direct interstitial-interstitial migration cannot be the only mechanism involved. The two competing mechanisms which can account for such a behavior are the Frank-Turnbull mechanism



and the kick-out mechanism



In the first case (reaction (4.1)), the diffusion of gold into silicon is limited by the in-diffusion of vacancies from the surface, which are annihilated via (4.2) by the highly mobile Au_I . The surface acts as a source both for the vacancies and Au_I , replenishing the resulting undersaturation. This effect is simulated by a special version of VIDSIM by creating a vacancy and an Au_S at random positions on the surface.

If reaction (4.2) is operative, then the resulting Au_S -profile is established by the in-diffusion of the Au_I and the diffusion of the self-interstitials to the surface, which acts as a sink for the self-interstitials produced by reaction (4.2) in supersaturation.

Experimentally, the following two properties of the profile resulting from the in-diffusion of Au into Si are well established:

(a) the diffusion of Au from both surfaces into a Si wafer yields a U-shaped gold concentration (C_S) profile

(b) the concentration of Au_S in the middle of the U-shaped profile (C_S^m) increases proportional to the square root of the diffusion time:

$$C_S^m \propto \sqrt{t} \quad (4.3)$$

over a long period of time.

Whereas in the 1960's and 70's Au diffusion was ascribed to the Frank-Turnbull mechanism²⁵⁻²⁹, Goesele et al.³⁰ and Seeger³¹ suggested in 1980 the kick-out mechanism as an explanation. Their theory is based on analytical calculations, i. e. a set of nonlinear partial differential equations for the concentration of the substitutional atoms, the vacancies and the self-interstitials, in which they claim that after some approximations the effective Au_S diffusivity has to be independent of the Au_S concentration in the case of the Frank-Turnbull mechanism; in the case of the kick-out mechanism, however, they calculated it to be *strongly concentration dependent*.

According to their calculations, which neglected among other things electrostatic interactions, the Au_S profile resulting from the Frank-Turnbull process is the error function complement (ERFC):

$$C_S = C_S^{eq} \operatorname{erfc}\{x/2(Dt)^{1/2}\} \quad (4.4)$$

in the case of a semi-infinite solid at $x \geq 0$ with an inexhaustible source for Au_S at its surface $x = 0$, which remains at its equilibrium concentration C_S^{eq} at all times. This result is not in accordance with the available experimental profiles. For the kick-out mechanism, the same profile results in the case of a

high density of internal sinks or sources, which is not within the scope of this thesis. For the Au diffusion into highly perfect wafers (low density of interstitial sinks), however, different results follow: Goesele et al.³⁰ calculate the solution for this problem with the boundary conditions

$$C_S(x=0, t) = C_S(x=d, t) = \infty \quad (4.5)$$

and the initial condition

$$C_S(0 < x < d, t=0) = 0 \quad (4.6)$$

to be³²

$$\operatorname{erf}[\{\ln(C_S/C_S^m)\}^{1/2}] = \left| \frac{d/2 - x}{d/2} \right|, \quad (4.7)$$

where d is the thickness of the wafer.

Seeger³¹, though, points out that solution (4.7) is a poor approximation near the wafer surfaces. He presents a solution which can be used near the surfaces, even for experimental conditions where C_S does not come close to its equilibrium value in the bulk:

$$C(\eta) \approx \frac{1}{1 + [-\ln(2|a_0|\pi^{1/2}C^0)]^{1/2}\eta} \quad (4.8a)$$

with

$$\eta = x/(D^*t)^{1/2}. \quad (4.8b)$$

Another very important result of these deductions is the prediction of the square root law (eqn. (4.3)) for the kick-out model, whereas for the Frank-Turnbull model a linear time dependence of C_S^m is calculated³⁰ - quite in contrast to Huntley and Willoughby^{27,28}, who obtained eqn. (4.3) in their calculations for the dissociative mechanism.

It is the goal of our Monte Carlo simulation to shed new light on these issues by simulating Au into Si diffusion with the two different models, remaining as closely as possible to the assumptions of the different authors. The results of the simulation will then be compared to the theoretical predictions to check, whether they really comply with them. In other words: a computer experiment is performed, adapted to the different mechanisms, defects and parameters of the two competing models. The experimental outcome will hopefully contribute to decide, whether these models are verifiably suitable for the explanation of diffusion processes.

4.2 Simulation Methods and Aspects

When setting up a **VIDSIM** simulation, one of the first things one has to do is to determine the size of the simulated crystal. As the envisaged diffusion simulation is basically a one dimensional process, one of the dimensions of the sample (let's call it the x direction) can be much larger than the other two - the periodic boundary conditions of **VIDSIM** will then guarantee almost infinite y and z proportions. After some tests we decided to take 2000 lattice constants a in the x direction, which is (rf. Table A.2) $1.09 \mu\text{m}$ in real space. The y and z direction, respectively, were taken smaller by a factor of ten, resulting in a simulated bulk of $6.4 \cdot 10^8$ atoms.

Secondly, the parameters for the simulation have to be determined. We decided to take a simulation temperature of 1368.5 K ($\approx 1095^\circ\text{C}$) because it is high enough to ensure a fast diffusion process and there is abundant data at this temperature region. The required activation energies for the kick-out mechanism were excerpted from the literature, in which the diffusivities were given. It has to be mentioned, however, that these values are estimates, based on numerous experiments and assumptions; the problem of gaining separate values for E_{act} and the enthalpy of formation E_x^f (see (eqn. 2.6)) is - as mentioned in chapter 2.2 - not yet completely solved. Table 4.1 shows the adopted values.

Table 4.1.
Simulation energy parameters.

Defect	Energy	Source
Au_I	$E_{mig} = E_{kin}' = 0.40 \text{ eV}$	Tan and Goesele ³³
	$E_f = 2.5 \text{ eV}$	Weber ³⁴
Si_I	$E_{mig} = E_{kin}' = 0.39 \text{ eV}$	Weber ³⁴
	$E_f = 4.4 \text{ eV}$	Tan and Goesele ³³

The corresponding values for the thermodynamic equilibrium concentration of the defects were also taken from Ref. 33, yielding 588 Au_S , 110 V , 48 Au_I and 4 Si_I for the size of our simulated bulk.

For the Frank Turnbull simulation, values from Van Vechten's long established *Ballistic Model*¹⁴⁻¹⁸ were calculated, according to

$$E_{kin} = \frac{1}{2} mv^2 = \frac{1}{2} (Fd\nu_D)^2, \quad (4.9)$$

where ν_D is the Debye frequency, d the distance between the two lattice sites, m the mass of the hopping atom and F a geometric constant equal to 0.9 in the case of the diamond lattice. The calculations gave 1.2 eV for the hopping Si atom. We chose 2.5 eV for Au as a reasonable value. Table 3.3 shows the activation

energies in the form of the occupant file, the way they were fed into the program.

Finally, the values for the inter-occupant files have to be specified. The bond energies between two Si and two Au atoms, respectively, were calculated from the latent melting heat to be 0.18 eV for Si-Si and 0.029 eV for Au-Au, respectively. The silicon-vacancy bond energy was set to -1.0 eV in accordance with the *Ballistic Model*, and the remaining parameter, such as the interstitial-lattice binding energies, were supposed to be of negligible values and were set to zero.

Because the advocates of an interstitial mechanism cannot specify which type of interstitial accounts for diffusion in silicon¹², we arbitrarily chose the bond-centered for simulational reasons, and in order to save CPU time.

Due to the poorly known boundary conditions and surface effects, we decided to simulate two cases for the kick-out, which differ in the way the injection of the Au interstitials into the surface is handled:

(a) In the *stimulated kick-out*, a Au_I is injected into the surface at a random position every time a Au_S is formed in the bulk via reaction (4.2). Physically, this could be explained by the release of energy of the kick-out, which could incite an injection. This method ensures a constant concentration of Au_I throughout the whole simulation and can be regarded as a good approximation of the surface condition (4.5). In the initial configuration, all of the Au_I calculated for the thermal equilibrium (48) were distributed along the two surfaces. The initial 588 Au_S were also distributed on the exterior of the bulk to take into account any effects resulting from an eutectic layer. Another surface effect which has to be considered is the annihilation of a Si_I , when it reaches the (y-z) surface, so that the surface can act as a sink for the supersaturated Au_I . In fact, the simulation was determined by the in-diffusion of the Au_I , which created a Au_S via the kick-out. As the resulting Si_I has a very high probability of inverting again that process in the next event via (4.2), we decided to modify the kick-out to a second neighbor event. Furthermore, to speed up the direct interstitial-interstitial migration in the diffusional direction, we added \pm one lattice constant to the hop in the $\pm x$ direction, which affects two thirds of all hops (one third does not hop in the $\pm x$ direction at all). This measure has two effects: First, it takes care that an Si_I which is created somewhere in the middle of the bulk does not take days of simulation time, until it is annihilated on the surface. As the direct self-interstitial migration is extremely

fast in virtual (laboratory) time and does practically not advance the virtual clock, this does not bias the result. Second, it boosts the Au_I at a faster migration speed into the bulk, so that they penetrate faster and deeper. This means that the resulting profile will be stretched out but still proportional to the original one. In particular the shape of the profile is not affected by this measure. Nonetheless, the overwhelming event was still the direct interstitial-interstitial hop of the Au_I , a fairly uninteresting event. We raised the migration energy for Au_I to 2.2 eV, so that for every three thousand direct interstitial hops we had one kick-out. The predominance of the direct interstitial hops (an experimental fact!) was not disturbed, but the time scale reduced by a factor of $4.6 \cdot 10^6$.

(b) In the *spontaneous kick-out*, it is assumed that the injection of the Au_I proceeds at a constant rate of time. We found the average time for a Au_I injection to be $2.8 \cdot 10^9$ time units from our experience with the stimulated kick-out. In contrast to the case (a), the initial dfp-file was empty, i. e. we simulated the evolution of crystal defects in time by a constant rate of Au_I injection. Apart from these two differences, this simulation was set up exactly the same way as the stimulated kick-out.

For the Frank-Turnbull mechanism, basically the same distinction was made:

(a) In the *stimulated Frank-Turnbull*, a creation of a Au_S via reaction (4.1) prompts the injection of one V . The initial 588 Au_S were again distributed at random positions along the surface, acting as an inexhaustible source for the Au_I and V according to the reverse dissociative mechanism (4.1). This way, they are providing "supplies" for the initial 48 Au_I , which had also been located near the surface.

(b) In the *spontaneous Frank-Turnbull*, every $6.8 \cdot 10^3$ time units (an experimental value) two V and one Au_I are formed on the surface, but, as all vacancies and interstitial reaching the surface are annihilated, most of them don't survive the first events as we assume a free surface and no eutectic layer. Again, there are no defects existing at the start of the simulation.

The different versions of this simulation were running for almost 4 months on up to 6 computers, including one PS/2 Model 70 and the IBM 3090 of the Palo Alto Scientific Center.

4.3 Results

4.3.1 Kick-out Mechanism

(a) Stimulated Kick-out

During the simulation we found that the Au_I diffused fast into the sample and remained at a high concentration near the surface, but were undersaturated in the middle of the sample, as they got eaten up during their migration to the center. The number of Si_I hardly ever exceeded the value 1.

Fig. 4.1 shows a logarithmic plot of the obtained Au concentration profiles after 3 different virtual times, up to $t = 6.4 \cdot 10^{12}$ time units (in phonon periods). In real time, this would be the equivalent of $0.1 \mu s$. The concentration was obtained by "virtual" cutting the crystal in the x direction into slices of 20 lattice constants and then counting the Au_S which are within one slice, and subsequent adding the numbers which result from the diffusion from both surfaces. Thus, to get a relative Au_S concentration, these values have to be divided by $1.28 \cdot 10^9$. To eliminate the stretching out of the profile which is due to the methods described in chapter 4.2, the distance from the surface would have to be reduced by a factor of 4.83.

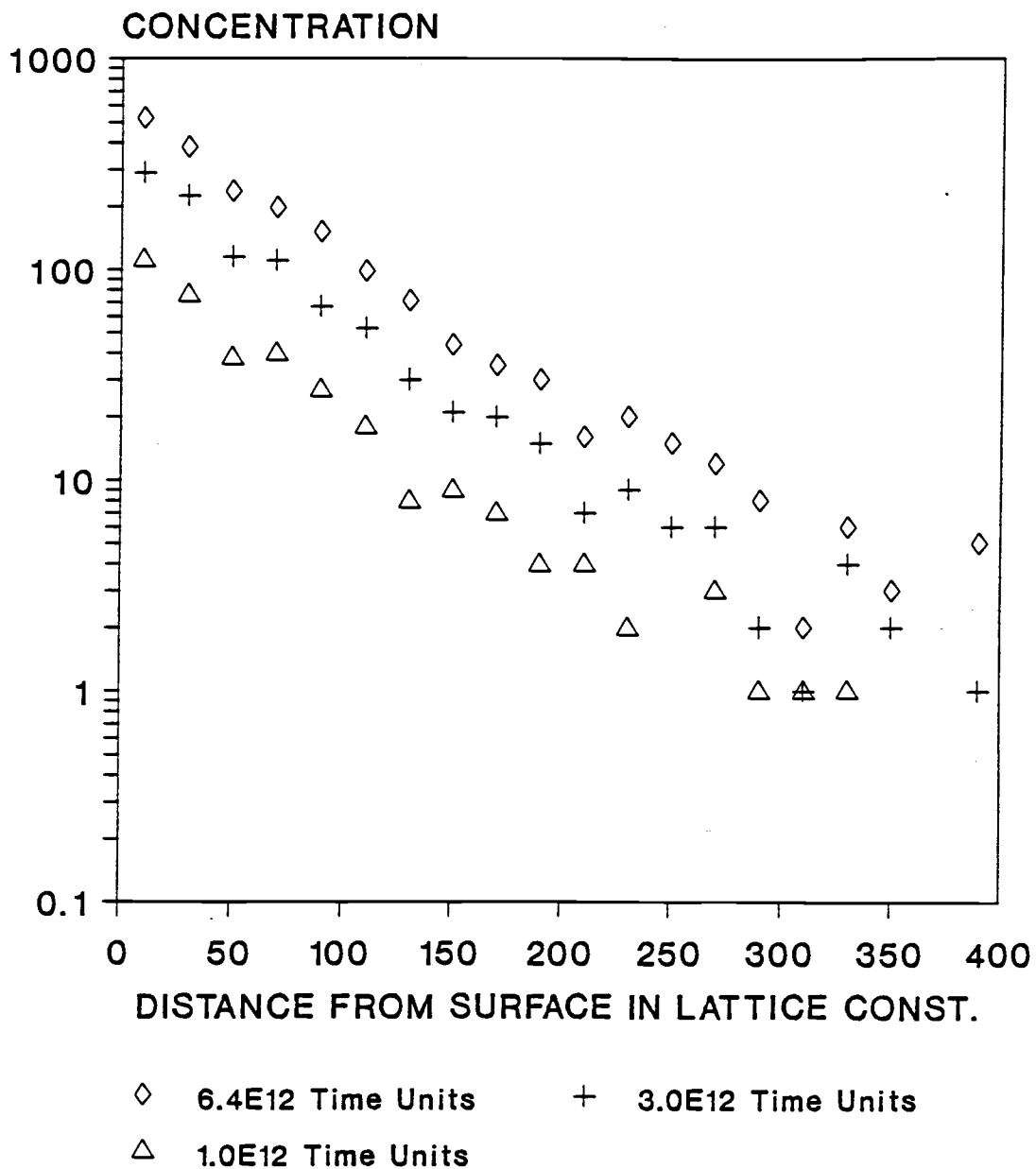


Fig. 4.1.
Au concentration profiles of stimulated kick-out.

All the three profile look very much the same and show a very nice exponential behavior until the penetration depth exceeds about 150 lattice constants, where they are flattening out a little bit. Ergo, on the scale of the whole length of the x direction, we have a U-shaped profile. Figure 4.2 shows an attempt to fit the data which we obtained at $6.4 \cdot 10^{12}$ time units to an exponential curve. The plotted error bars represent a \sqrt{n} error behavior which is typical for Monte Carlo problems and which we assumed to be valid in this case, too. Using these data and errors, we used a χ^2 fitting program for nonlinear models according to the Levenberg-Marquardt²³ method to calculate the parameters of the function $C_0 \exp(-x/\sigma)$ and to get an estimate of the goodness-of-fit. The main premise for such a fit, namely the normal distribution of the measurement errors, is certainly fulfilled for a Monte Carlo simulation. The calculated values for χ^2 never exceeded the degrees of freedom N (data points - adjustable parameters) of the fit by more than 10.5%, which is an indication for a good fit. Moreover, we calculated the statistical probability Q that, given the calculated parameters, this data set would have occurred plus or minus the given errors:

$$Q = \Gamma(0.5 N, 0.5 \chi^2), \quad (4.10)$$

where Γ is the incomplete Gamma function. As shown in Table 4.2, the Q values start off with an excellent value of 0.967 and abate with progressive time to an acceptable value of 0.300. This is an

indication that at the beginning of the diffusion process the profile commences with an accurately exponential shape. As time increases, however, the "tail" of the profile becomes consecutively smoother; this seems to be confirmed by the increase of the inverse slope of the semi-logarithmic plot, σ . The physical process that causes this change of the profile shape is the "real" diffusion by the inverse kick-out mechanism, i. e. the Au_S get displaced by interstitials, in contrast to the exponential contour, which is simply determined by the in-diffusion of the Au_I from the surface until they knock off a silicon substitutional. This characteristic has also been confirmed by "tagging" the Au_I , which transfer their tag to the Au_S in such a case. Furthermore, σ is identical to the square root of the probability that the interstitial performs a direct interstitial hop rather than a kick-out (53.7).

Table 4.2.
 χ^2 fit for exp() function of stimulated kick-out.

time	C_0	σ	χ^2	Q
$0.5 \cdot 10^{12}$	75.3 ± 2.4	49.6 ± 4.4	8.7	0.967
$1.0 \cdot 10^{12}$	126.7 ± 10.2	54.1 ± 3.3	12.3	0.832
$2.0 \cdot 10^{12}$	232.5 ± 13.3	57.9 ± 2.4	18.5	0.435
$3.0 \cdot 10^{12}$	341.9 ± 16.1	57.1 ± 2.0	19.9	0.530
$4.0 \cdot 10^{12}$	449.4 ± 18.4	57.7 ± 1.7	23.3	0.330
$5.0 \cdot 10^{12}$	534.4 ± 19.7	59.5 ± 1.6	23.8	0.534
$6.0 \cdot 10^{12}$	605.5 ± 20.5	60.9 ± 1.5	24.2	0.390
$6.4 \cdot 10^{12}$	643.6 ± 21.2	60.9 ± 1.5	26.0	0.300

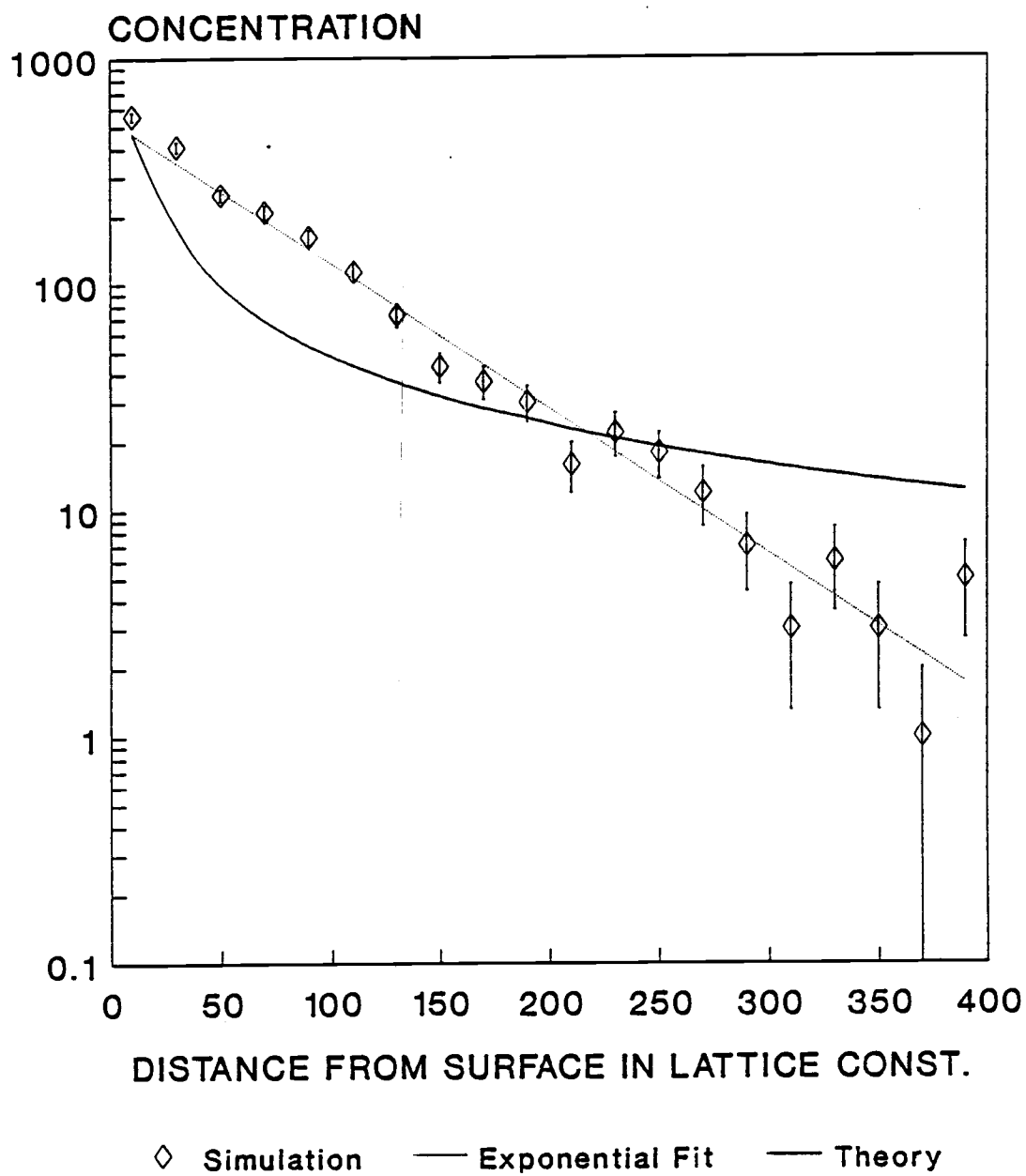


Fig. 4.2.
Au concentration profile at $6.4 \cdot 10^{12}$ time units of stimulated kick-out.

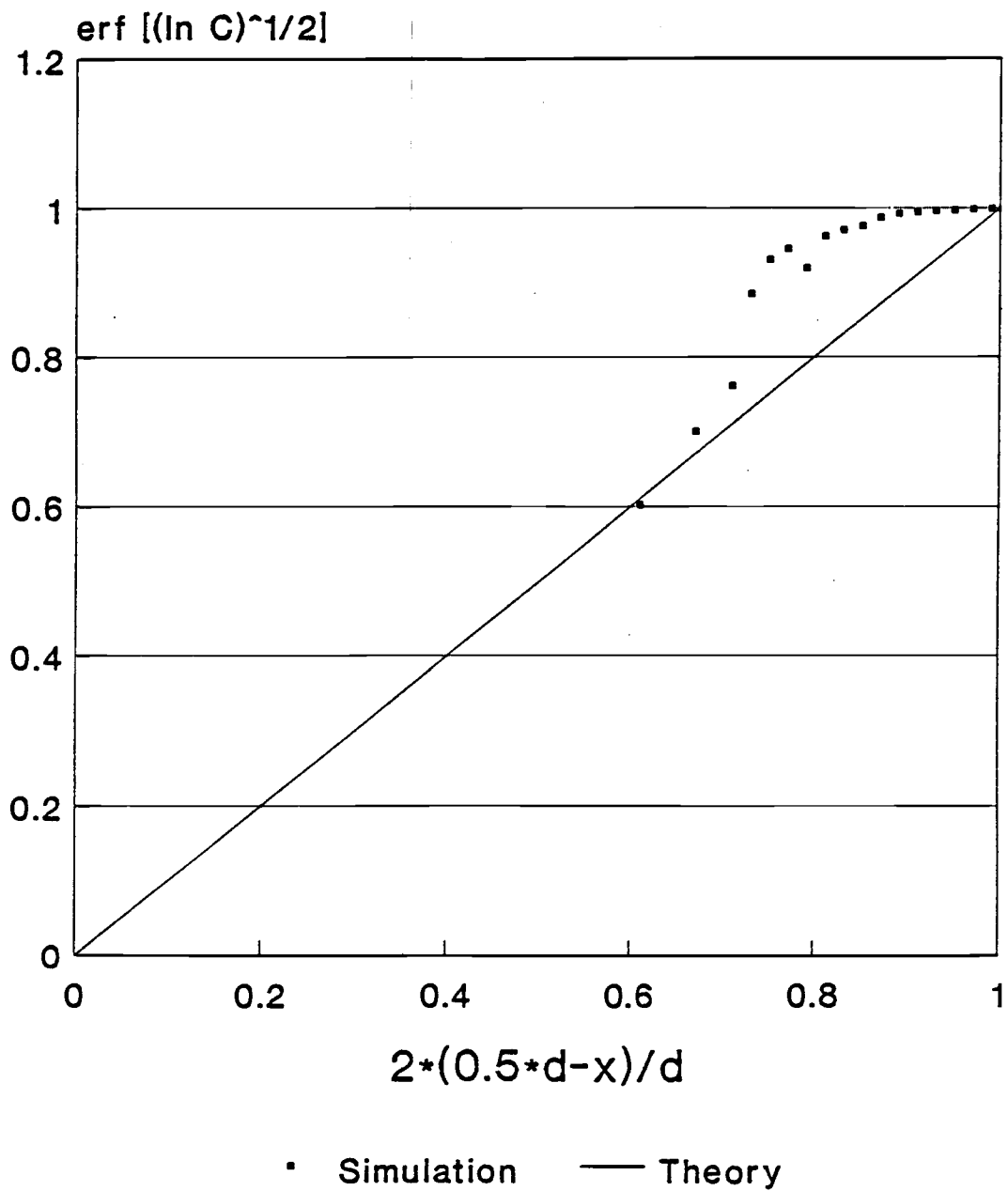


Fig. 4.3.
Stimulated kick-out: comparison with theory.

When we tried to fit the data to the predicted function (4.8), we obtained $Q = 0$. This attempt is plotted in Fig. 4.2, (solid line) and labeled as "theory". It seems to be evident that the simulation could not verify the theoretical predictions.

We also compared the kick-out results with eqn. (4.7), which is the alternative theoretical derivation. As illustrated in Fig. 4.3, the data points are way off the straight line, which eqn. (4.7) would assert. This theory cannot be supported either by our data, either.

Besides the shape of the diffusion profiles, the time dependence of the concentration in the middle of the bulk $C_S^m(t)$ is another important issue. To obtain C_S^m , we added all Au_S which penetrated deeper than 250 lattice constants into the bulk. The result is a nice linear plot, as shown in Fig. 4.4. A linear fit, which is drawn as a solid line, yielded a regression coefficient of $R = 0.9966$. The hypothesized square root behavior (4.3) is not in accordance with our result.

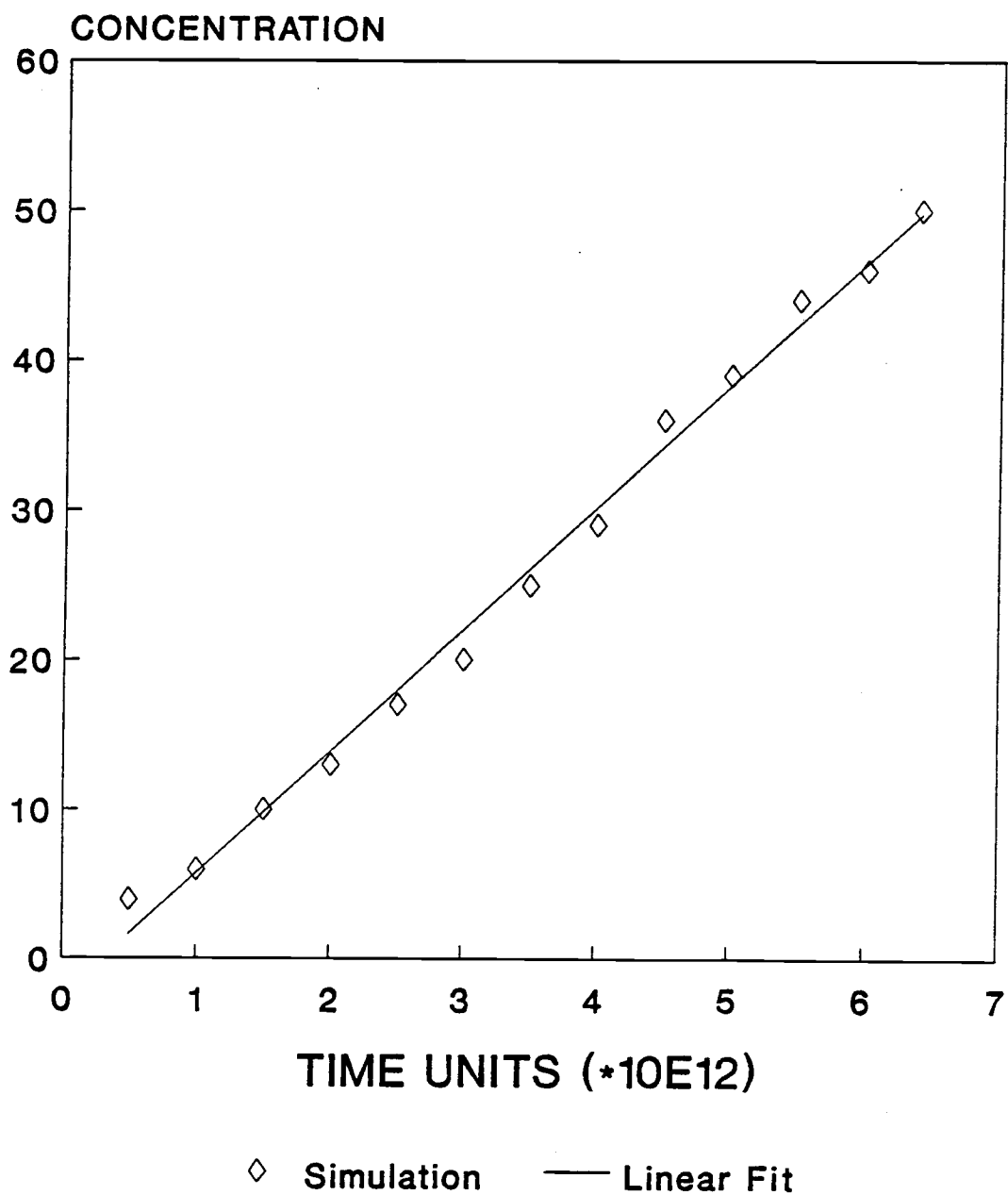


Fig. 4.4.
Central Au concentration vs. time of stimulated kick-out.

(b) Spontaneous Kick-out

Fig. 4.5 shows the concentration of the Au_I at $t = 1.9 \cdot 10^{12}$ and $t = 3.8 \cdot 10^{12}$ time units. Just like in the stimulated kick-out, there is a high and constant concentration of Au_I near the surface, which drops off exponentially after about 50 lattice constants. In both cases, the concentration of Au_I is almost zero beyond 180 lattice constants. The few interstitials beyond that point result from the inverse kick-out mechanism, a conclusion which accords with our observation of the flattening contour of the Au_S beyond 150 lattice constants. In addition, it seems that the proportion of these interstitials is increasing with time. This result is quite in contrast to what the protagonists of the kick-out assert³⁴ and assume for their calculations³⁰, namely a flat Au_I profile. At the end of the simulation, there were 60 Au_I in the sample, which is close to the expected value.

The Au_S profiles of this version are shown in Fig. 4.6. They are very similar to the previous ones. The differences can only be made transparent by statistical methods, as shown in Table 4.3.

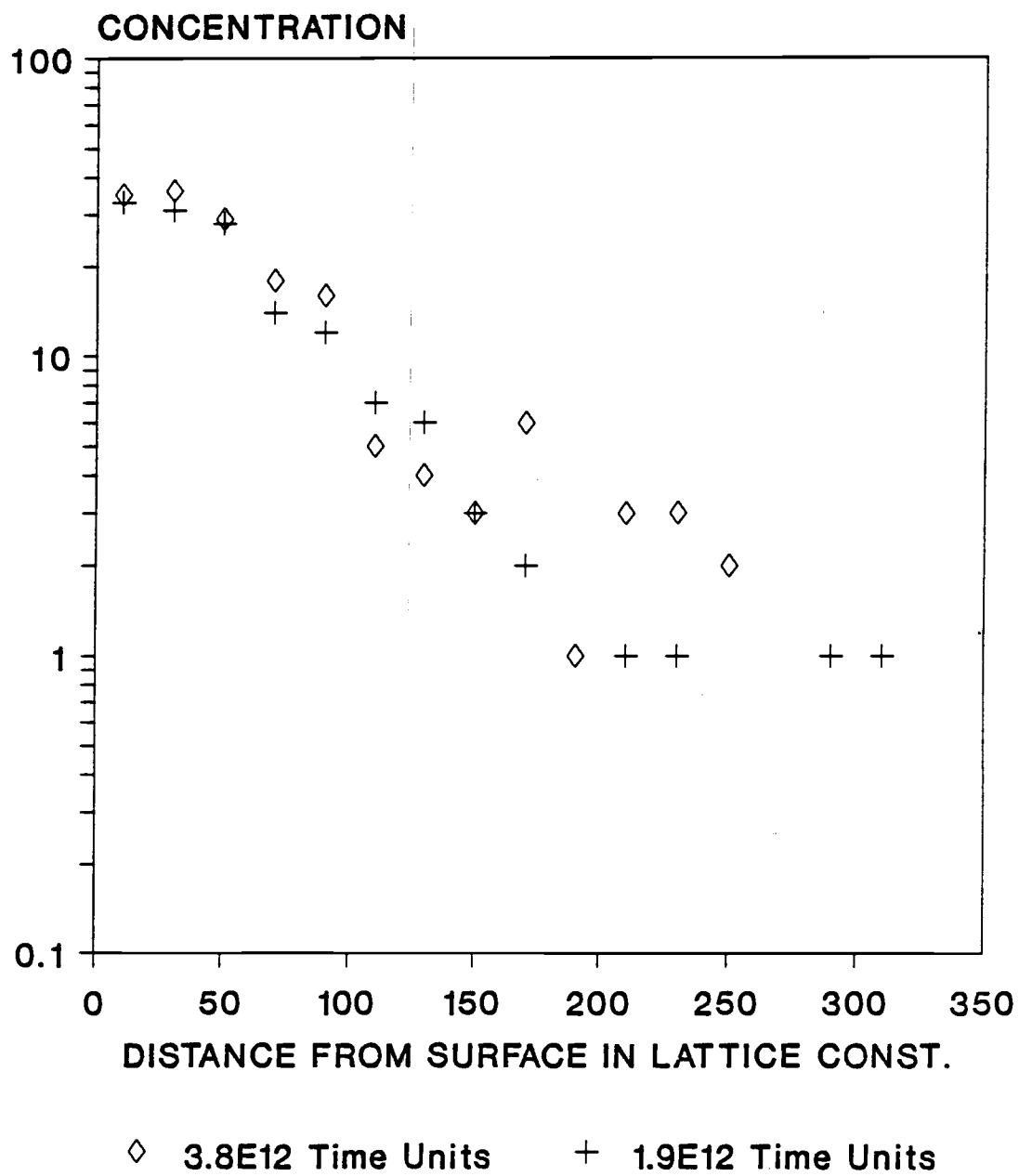


Fig. 4.5.
Au interstitial concentration profiles of spontaneous kick-out.

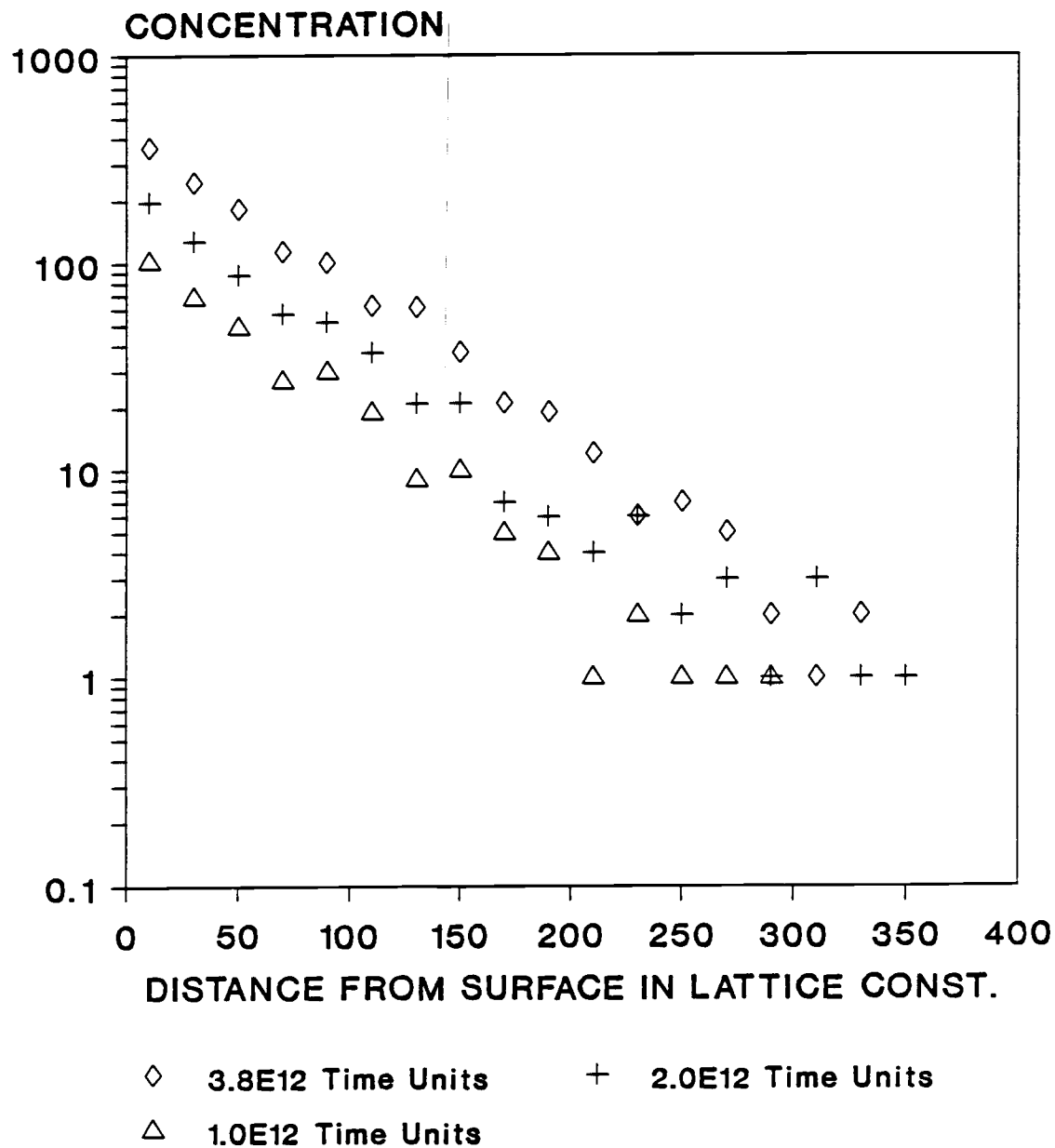


Fig. 4.6.
Au concentration profiles of spontaneous kick-out.

Table 4.3.
 χ^2 fit for $\exp()$ function of spontaneous kick-out.

time	C_0	σ	χ^2	Q
$0.5 \cdot 10^{12}$	$55,7 \pm 7.2$	49.6 ± 5.0	9.2	0.954
$1.0 \cdot 10^{12}$	122.6 ± 9.9	53.0 ± 3.1	7.4	0.986
$1.5 \cdot 10^{12}$	177.5 ± 11.9	52.7 ± 2.5	10.0	0.932
$2.0 \cdot 10^{12}$	227.2 ± 13.2	55.3 ± 2.3	11.4	0.978
$2.5 \cdot 10^{12}$	272.9 ± 14.0	58.8 ± 2.2	9.8	0.940
$3.0 \cdot 10^{12}$	327.7 ± 15.2	59.1 ± 2.0	9.2	0.955
$3.5 \cdot 10^{12}$	386.5 ± 16.5	58.6 ± 1.8	14.7	0.680
$3.8 \cdot 10^{12}$	422.9 ± 17.3	58.2 ± 1.7	13.1	0.784

The data indicate that at first the exponential behavior is very well fulfilled. Then, however, the exponential fit deteriorates again; a behavior, which is completely analogous to the stimulated kick-out. We may conclude from that there is basically no difference in the two cases. Even the slopes seem to be in exact agreement with each other.

This conclusion is reinforced by Fig. 4.7 and Fig 4.8, respectively, which speak for themselves. The obligatory linear fit for the time dependence of C_S^m yielded a correlation coefficient of $R = 0.9894$, a reasonable value. Fig. 4.9 shows the corresponding graph.

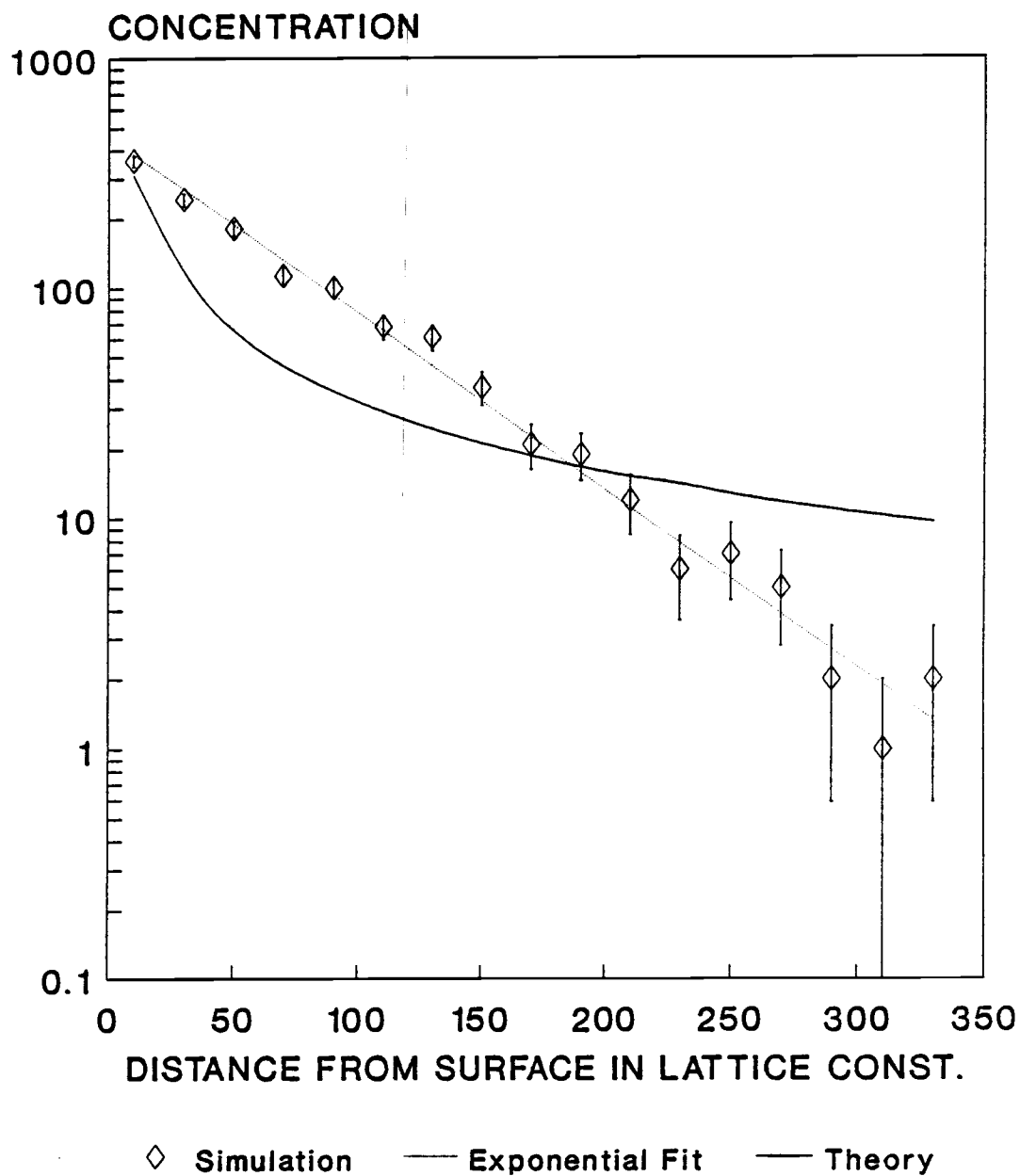


Fig. 4.7.
 Au concentration profile at $3.8 \cdot 10^8$ time units of spontaneous kick-out.

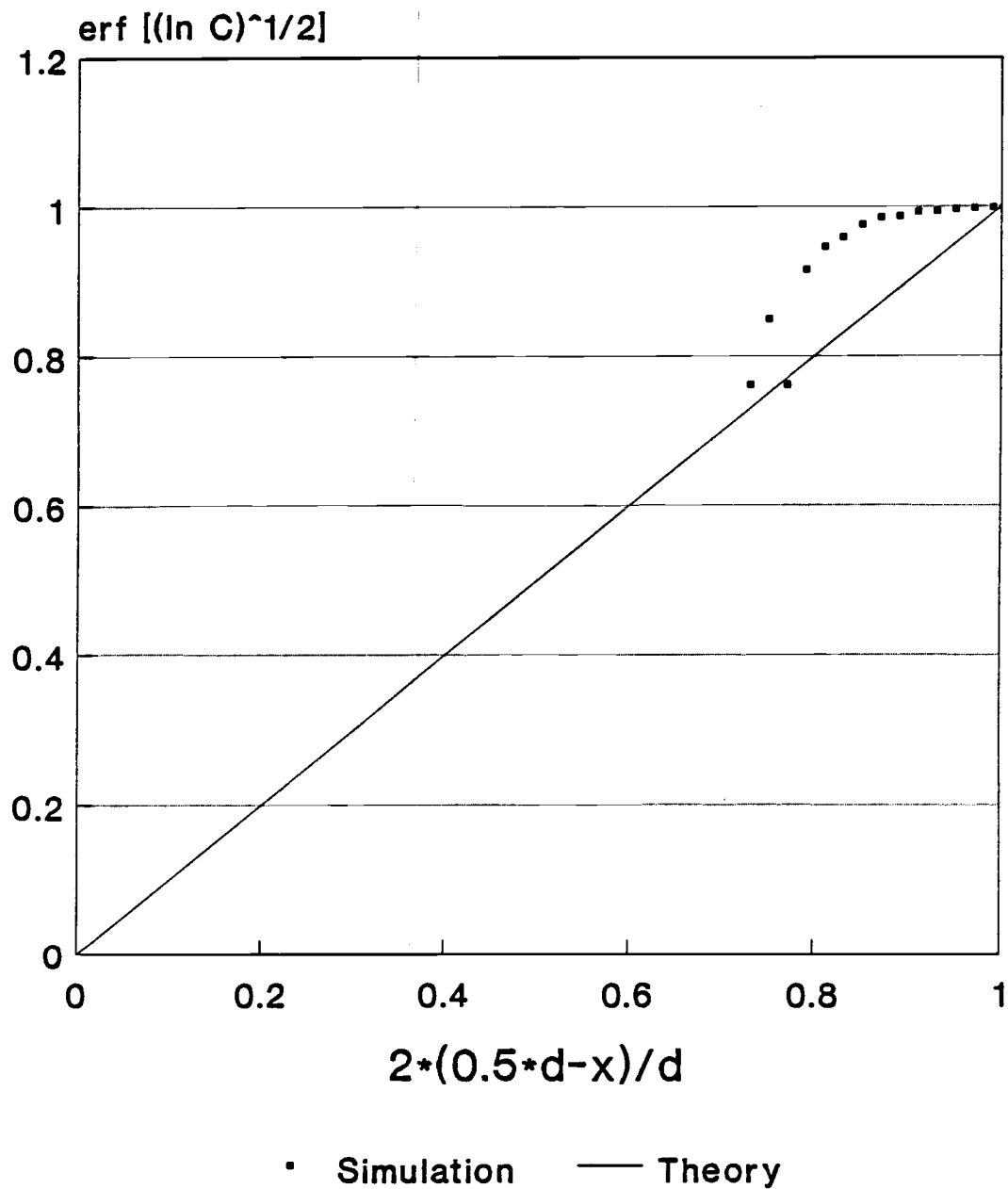


Fig. 4.8.
Spontaneous kick-out: comparison with theory.

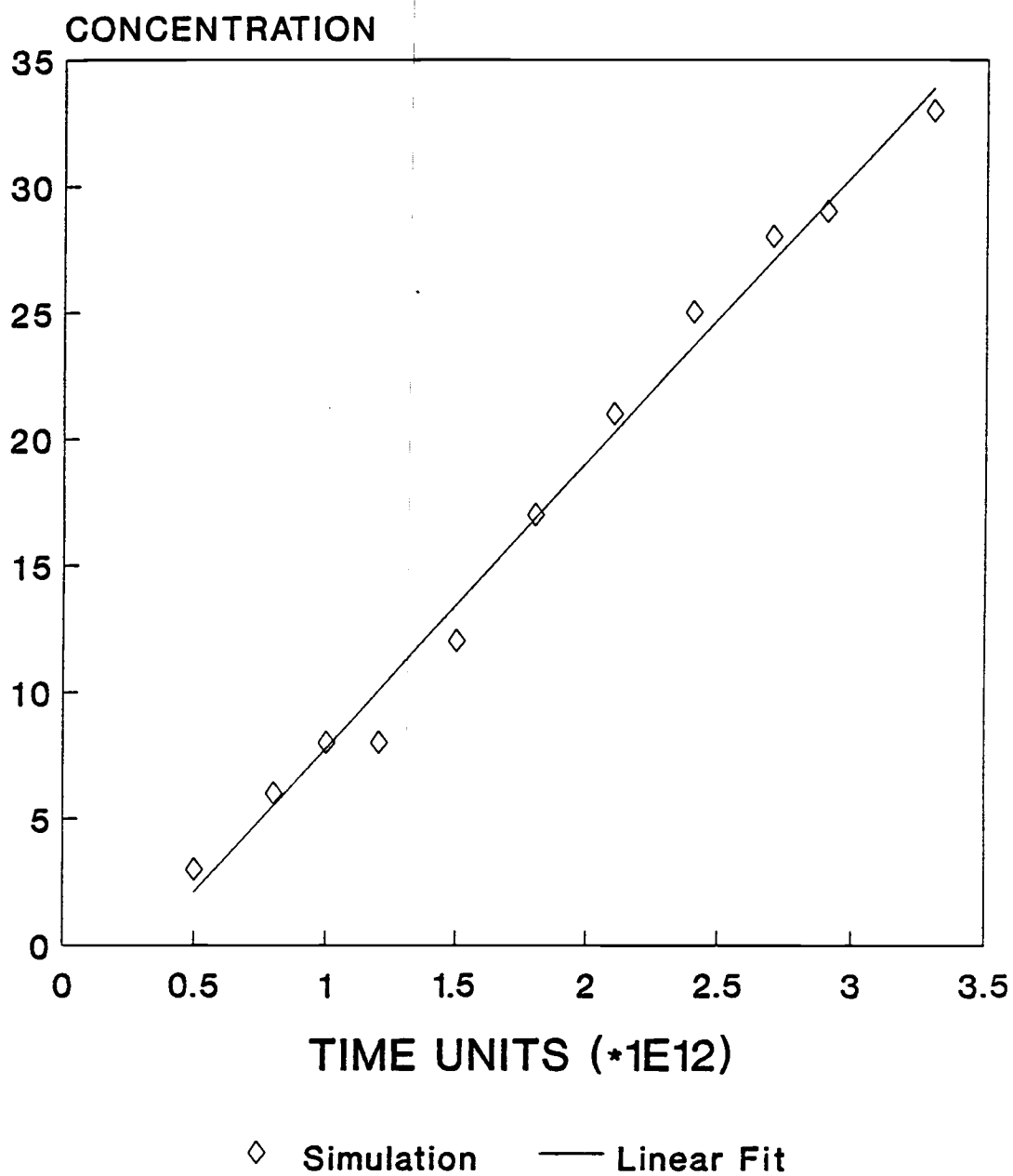


Fig. 4.9.
Central Au concentration vs. time of spontaneous kick-out.

4.3.2 Frank-Turnbull Mechanism

One basic problem which occurs for the Frank-Turnbull is the number of Au_S which are produced during a simulation. According to the law of mass action, this is proportional to the product of the components which are involved in the reaction. As it turns out, the product for the kick-out mechanism, $C_I \cdot C_{Si}$ (ref. reaction (4.2)), is about $5.8 \cdot 10^8$ greater than the corresponding product for the Frank-Turnbull mechanism, $C_I \cdot C_V$ (reaction (4.1)), resulting in a much lower concentration of Au_S in the Frank-Turnbull case. This is also the reason why the Au_I profile in the kick-out is not a flat one, as an Au_I will find during every hop from the surface on a Si, which it can knock off its site, quite in contrast to the Frank-Turnbull case, where the vacancies (and the Au_I) spend a lot of time diffusing into the bulk, before their reaction partner comes into their vicinity. Thus, in a first approximation, the vacancies can be regarded as free defects diffusing into the crystal with no interaction at all, which corresponds to the Fickian assumptions. In the case of a constant rate of injection the resulting profile is known to be of the ERFC type³⁶. This is exactly what is the result of our spontaneous Frank-Turnbull simulation, as shown in Fig. 4.10. Close to the surface, the V concentration was constant and in thermodynamic equilibrium. The χ^2 fit to the ERFC function yielded the diffusivity for vacancies,

$$D_V = (8.1 \pm 1.3) \cdot 10^{-6} \text{ cm}^2/\text{s}. \quad (4.11)$$

The probability for obtaining these data with the fitted parameters was $Q = 0.925$, an excellent value.

This is almost exactly the value which we calculated, using an equation which can be derived from the *Ballistic Model*⁹, $D_V = 8.5 \cdot 10^{-6} \text{ cm}^2/\text{s}$, if we include into the preexponential factor the enthalpy of formation⁹ of 4.1 k_B .

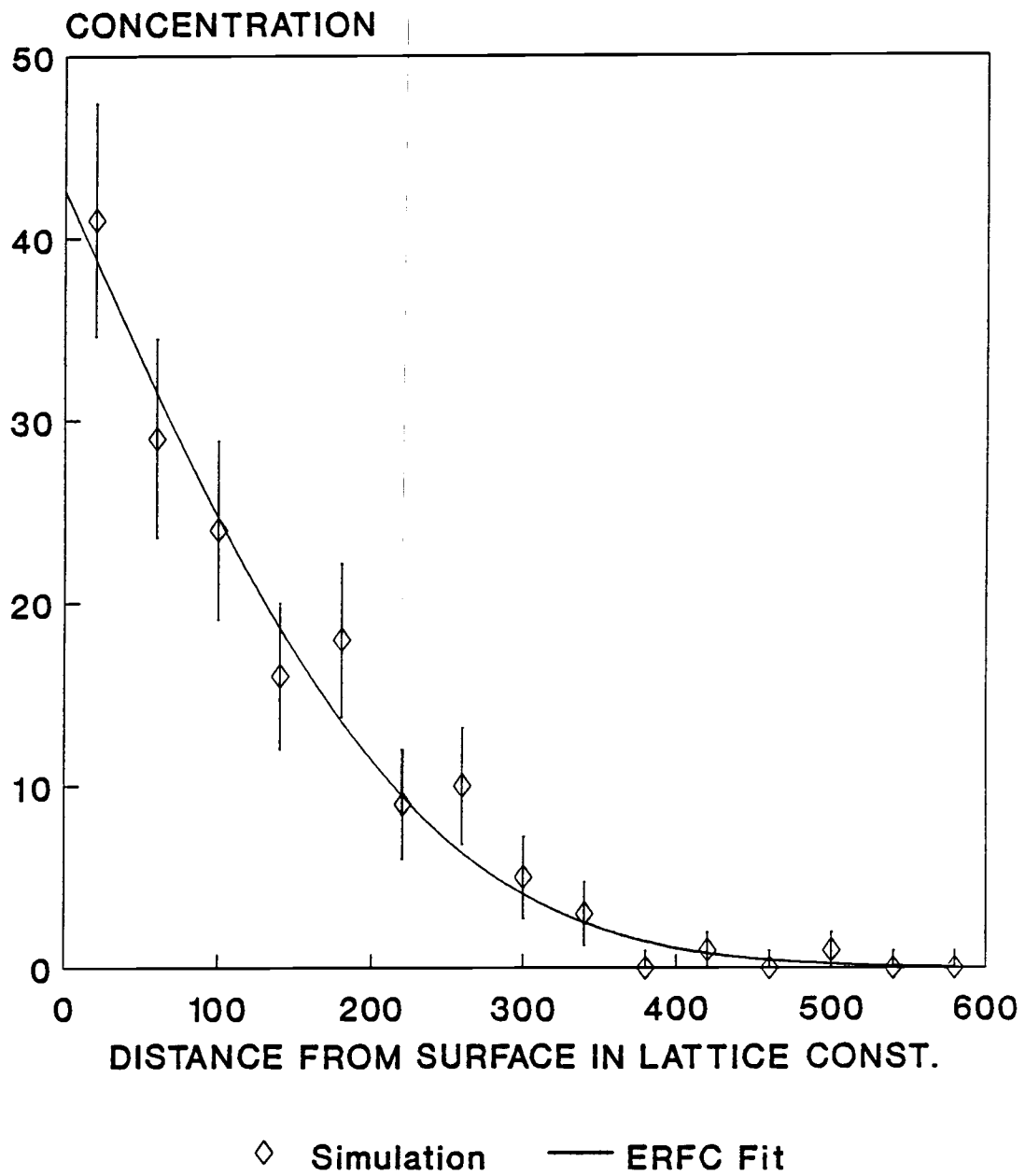


Fig. 4.10.
Vacancy concentration profile of spontaneous Frank-Turnbull.

(a) Stimulated Frank-Turnbull

As explained above, the number of occurring Au_5 was quite limited. To partly cure this problem, we ran the same simulation on two computers and added up the amount of Au_5 . The resulting profile after $5.3 \cdot 10^7$ time units (almost $4 \mu s$) is depicted in Fig. 4.11. The best fit we could obtain with these data was to the one to an ERFC function ($Q = 0.913$), which would be in accordance with theoretical reflections (compare eqn. (4.4)). However, some Au_5 seemed to occur at much higher level deep in the sample than what could accounted for by the ERFC function. As we also allowed the spontaneous formation of V via the reaction



(rf. chapter 2.3.2) in the bulk, there is an additional source of vacancies at random positions within the crystal, i. e. vacancies, which don't have to diffuse from the surface into the bulk and would be readily available for a Frank-Turnbull in the middle of the bulk. The calculated rate at which reaction (4.12) can happen is not enough to explain the observed deviation, as we hoped. This deviation from the ERFC profile seems to be a real physical effect.

The ERFC fit also yielded the diffusivity of the Au_S :

$$D_S = (3.4 \pm 1.3) \cdot 10^{-6} \text{ cm}^2/\text{s}. \quad (4.13)$$

Unfortunately, the experimental values for this property are hard to find and vary by several orders of magnitude, which makes it hard to compare our results. From Wilcox and La Chapelle²⁶ we find a value of about $1.0 \cdot 10^{-10} \text{ cm}^2/\text{s}$, which seems to be in accordance with values we extracted from the profiles measured by Huntley and Willoughby²⁷. It can be concluded that the pure Frank-Turnbull model does not account for the experimental data, either. It should be mentioned, that the previous analysis of the kick-out model did not yield any constant value for D_S , as the local rate of migration varies with the Au_S concentration. As a consequence of this, it was concluded that the profile ought not to be of the ERFC type.

Fig. 4.12 shows the central gold concentration. A straight line gives a reasonable fit to these data ($R = 0.963$), taken the few data points into account. The data seem to be in accordance with the theoretical predictions for the ERFC profile.

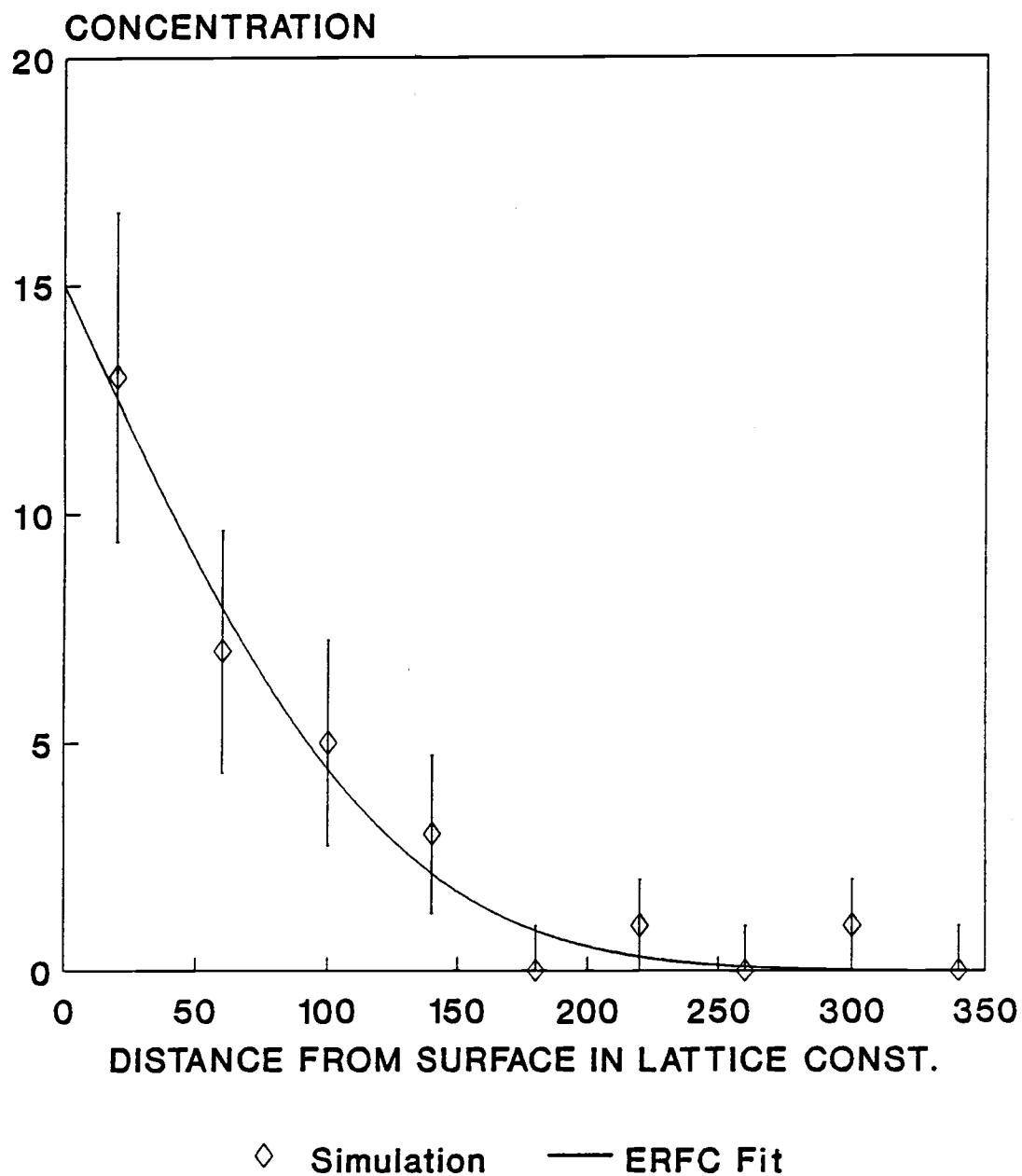


Fig. 4.11.

Au concentration profile at $5.3 \cdot 10^7$ time units of stimulated Frank-Turnbull.

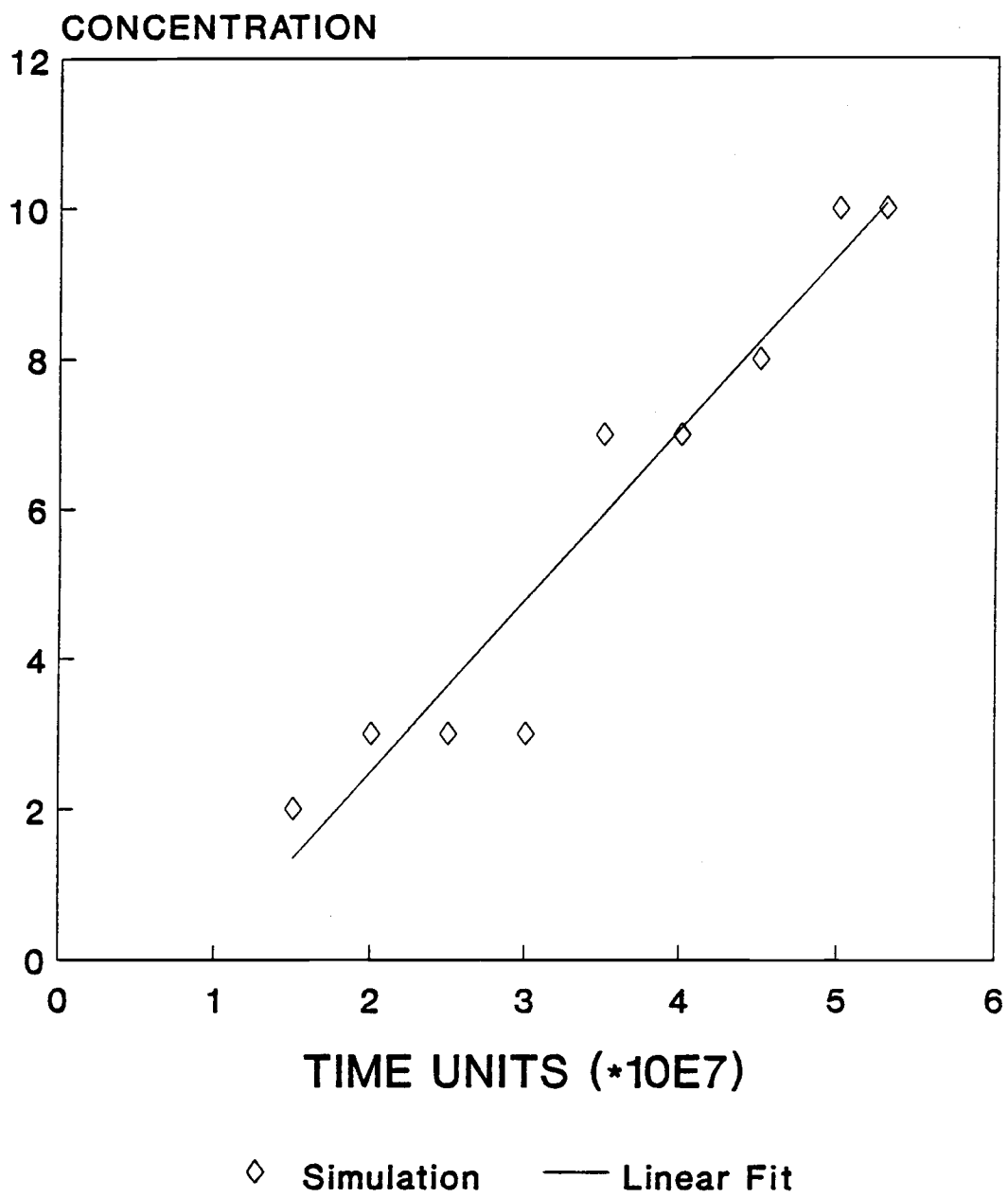


Fig. 4.12.
Central Au concentration vs. time of stimulated Frank-Turnbull.

(b) Spontaneous Frank-Turnbull

Here, we have even less data points. Fig. 4.13 indicates, however, a ERFC behavior, too. The fit is with $Q = 0.630$ not great, but better than any other trial function. The calculated value for the Au_S diffusivity is in this case

$$D_S = (4.0 \pm 2.4) \cdot 10^{-6} \text{ cm}^2/\text{s}, \quad (4.14)$$

which differs by only 10% from (4.13). In particular, the error limits of (4.13) and (4.14) overlap totally. Thus we can conclude, that the result of our simulation does not depend decisively on the injection conditions on the surface, something we saw already in the kick-out simulations. There also seems to be a slightly increased C_S in the middle of the bulk, as observed in the stimulated Frank-Turnbull.

The central gold concentration suggest a linear behavior in this case, too.

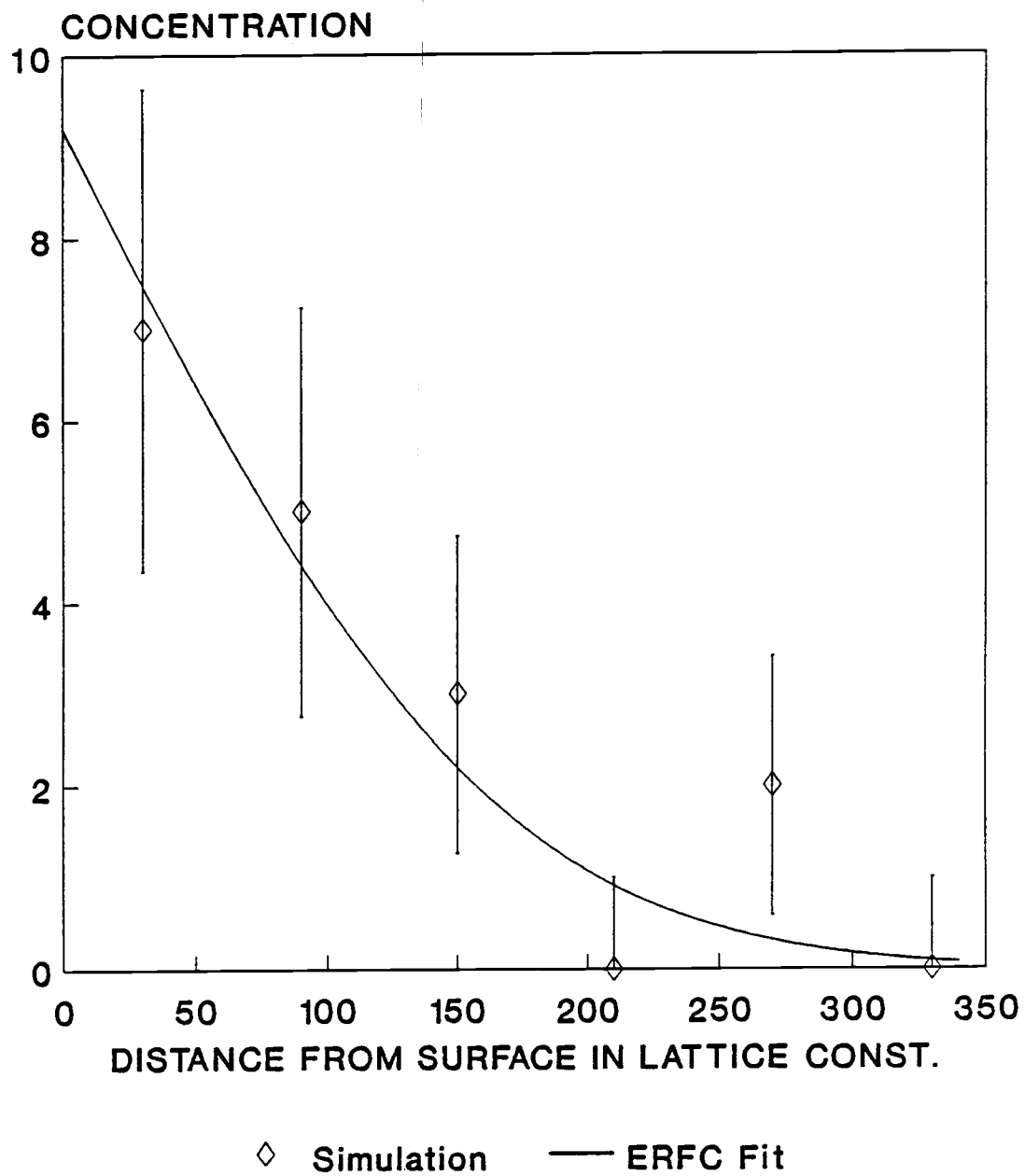


Fig. 4.13.
Au concentration profile at $8 \cdot 10^7$ time units of spontaneous Frank-Turnbull.

4.4 Discussion

With our application it was demonstrated, that VIDSIM can cope even with extremely complex problems, such as the diffusion of Au into Si. The main goal of this simulation, namely to gain insight into this puzzle by challenging two contradictory theories, has been fully achieved.

It was shown that both the kick-out and the Frank-Turnbull model are in qualitative agreement with the experimental data, as both yield U-shaped profiles.

The kick-out profiles, however, differ quantitatively from the predicted ones, and thus also from experimental profiles. The fact that the simulated crystal is thinner by a factor of about 500 compared to a typical experiment³² is almost compensated by the small diffusion times we could apply due to limited runtime on computers. One crucial advantage of a computer simulation is the fact, that surface effects can be excluded, in contrast to real wafer experiments, which can be severely limited by them, so that thicker samples have to be used in real life. After all, the physics of diffusion is not affected by the dimension of the sample. We can also monitor the diffusion process right from the beginning and don't have to take heating-up periods into account³². Thus, our profiles are exact even for extremely short times - something which is experimentally impossible. The fact

that the Au_S concentration decreased by about 2 orders of magnitude along the diffusion direction - a comparable amount to experiments - is also a hint that the applied virtual diffusion time was enough. An interesting question, however, is how the flattening of the initial exponential contour affects the profile with time overall - a problem, which will keep several micro-computers busy for quite a while.

Another interesting kick-out result is the Au_I profile and the behavior of the Au_I during their migration. The kick-out hypothesis obviously gives the Au_I a fairly high "cross-section", which results in a low penetration depth into the bulk; a "flat" distribution can't be explained with this model.

The profile obtained by testing the Frank-Turnbull hypothesis appear to be in quantitative agreement with the predicted theory, if we disregard the deviations in the middle of the bulk discussed earlier. During the end of the simulation, the number of inverse Frank-Turnbull effects (which create a V and a Au_I) was smaller by a factor of 4 than the inverse reaction. We can conclude from this that the simulation proceeded slowly towards an equilibrium state, in which the number of Au_S would be conserved. It is also shown that the process which determines the profile, namely the in-diffusion of vacancies, yields an ERFC shaped profile, which is what we expected for quasi-free defects (Fickian assumptions). Beyond that, we obtained an excellent value

for the diffusivity of the vacancies, a fact, that clearly restates the significance of the *Ballistic Model*.

It was observed that for both models the surface and injection conditions imposed on the simulation had only insignificant effects on the outcome.

As far as the central gold concentration is concerned, we could clearly show that neither of the two hypothesis yield a square root law, as experimentally observed. Though, the linear rise can't go on for ever, as inverse reactions of (4.1) and (4.2) will take effect and finally create an equilibrium, in which the creation and annihilate rate of Au_S are the same. It is easy to see that in both the Frank-Turnbull (4.1) and the kick-out mechanism (4.2) the rate of Au_S annihilation is proportional to the number of Au_S , whereas the rate of creation is, as we found, constant. In a local equilibrium the total change of the central concentration of Au_S can then be expressed by:

$$\frac{dC_S^m}{dt} = a - b C_S^m. \quad (4.15)$$

The solution of this differential equation is well known to be:

$$C_S^m = d \exp(-bt) + (a/b)\{1 - \exp(bt)\}, \quad (4.16)$$

which can be expanded into a Taylor series for short times:

$$C_S^m = d + (a - db) t, \quad (4.17)$$

yielding the observed linear time dependance. For longer times, however, there has to be saturation. By analyzing this simple differential equation it becomes evident that none of the two proposed models can give the right time dependence of the central gold concentration. In fact, there are many physical properties of the defects which neither of the two models takes into account:

- the Fermi level is affected by point defects and thus varies across the sample and throughout the process
- point defects are known to carry electric charge; electromagnetic interaction, however, has been disregarded³⁷
- electron-hole recombination-enhanced migration is observed in Si for positively charged vacancies³⁸.

Our results clearly show the importance of these effects, and we hope, that this study serves as an impetus for a thorough theoretical and experimental investigation of these effects, and their contribution towards diffusion in Si.

REFERENCES

1. J. C. Bourgoin, in: *Defects in Semiconductors*, ed. by L. C. Kimerling and J. M. Parsey, Jr. (AIME, New York, 1985).
2. For instance, see *Monte Carlo Methods in Statistical Physics*, ed. by K. Binder (Springer, Berlin, 1986).
3. A. Seitz, *Acta Cryst.*, 3, 346 (1950).
4. A. Seeger and K. P. Chik, *Phys. Stat. Sol.* 29, 455 (1968).
5. S. T. Pantelides, in: *Defects in Semiconductors*, ed. by L. C. Kimerling and J. M. Parsey, Jr. (AIME, New York, 1985).
6. W. Frank, A. Seeger and U. Goesele, in: *Defects in Semiconductors*, ed. by J. Narayan and T. Y. Tan (North-Holland, New York, 1981).
7. J. A. Van Vechten, *Phys. Rev. B* 17, 3197 (1978).
8. J. C. Bourgoin and M. Lanoo, *Rad. Eff.* 46, 157 (1980).
9. J. A. Van Vechten, in: *Defects in Semiconductors*, ed. by L. C. Kimerling and J. M. Parsey, Jr. (AIME, New York, 1985).

10. U. Goesele and H. Strunk, Appl. Phys. 20, 265 (1979).
11. A. J. R. De Kock and W. M. van de Wiggert, J. Cryst. Growth 49, 718 (1980).
12. U. Goesele, private communication, July 1988.
13. J. C. Phillips and J. A. Van Vechten, Phys. Rev. Lett. 30, 220 (1973).
14. J. A. Van Vechten, Phys. Rev. B 10, 1482 (1974).
15. J. A. Van Vechten, J. Electrochem Soc. 122, 419 (1975).
16. J. A. Van Vechten, in: *Handbook on Semiconductors*, ed. by S. P. Keller (North-Holland, Amsterdam, 1980), Vol. 3, Chap. 1.
17. J. A. Van Vechten, Phys. Rev. B 11, 3910 (1975).
18. J. A. Van Vechten, Phys. Rev. B 12, 1247 (1975).
19. D. Weiler and H. Mehrer, Phil. Mag. A 49, 309 (1984).
20. J. A. Van Vechten, J. Vac. Sci. Technol. B 2, 569 (1984).

21. F. C. Frank and D. Turnbull, Phys. Rev. **104**, 617 (1956).
22. A. Seeger and W. Frank, J. Electron. Mater. **14a**, 159 (1985).
23. W. H. Press et al., *Numerical Recipes in C* (Cambridge University Press, Cambridge, 1988).
24. W. M. Bullis, Solid-State Electron. **9**, 143 (1966).
25. W. C. Dash, J. Appl. Phys. **31**, 2275 (1980).
26. W. R. Wilcox, T. J. LaChapelle, J. Appl. Phys. **35**, 240 (1964).
27. F. A. Huntley, A. F. W. Willoughby, J. Electrochem. Soc. **120**, 414 (1973).
28. F. A. Huntley, A. F. W. Willoughby, Solid-State Electron. **13**, 1231 (1970).
29. W. R. Wilcox, T. J. LaChapelle and D. H. Forbes, J. Electrochem. Soc. **111**, 1377 (1964).
30. U. Goesele, W. Frank, A. Seeger, Appl. Phys. **23**, 361 (1980).
31. A. Seeger, Phys. Stat. Solidi (a) **61**, 521 (1980).

32. N. A. Stolwijk, J. Hoelzl, W. Frank, Appl Phys. A 39, 37 (1986).
33. T. Y. Tan, U. Goesele, Appl. Phys. A 37, 1 (1985).
34. E. R. Weber in: *Properties of Silicon*, EMIS Datareview (INSPEC, IIE London, 1988).
35. U. Goesele, private communication, November 1988.
36. J. Crank, *The Mathematics of Diffusion* (Clarendon Press, Oxford, 1975).
37. D. Mathiot, J. C. Pfister, J. Appl. Phys. 55, 3518 (1984).
38. J. A. Van Vechten, Phys. Rev. B 38, 9913 (1988).

APPENDICES

Table A.1.
Properties of the diamond - and zincblende structure.

Number of atoms per unit cell	Basis coordinates	Coordina- tion number	Distance of nearest neighbors
8	$ \begin{array}{l} 0\ 0\ 0; \begin{array}{l} 1\ 1 \\ -\ -\ 0; \end{array} \begin{array}{l} 1\ 1 \\ 2\ 2 \end{array} \begin{array}{l} 1\ 1 \\ 2\ 2 \end{array} \begin{array}{l} 1\ 1 \\ 2\ 2 \end{array} \\ 1\ 1\ 1\ 3\ 3\ 1\ 3\ 1\ 3\ 1\ 3\ 3 \\ -\ -\ -; -\ -\ -; -\ -\ -; -\ -\ - \\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4\ 4 \end{array} $	4	$ \frac{\sqrt{3}a}{4} $

Table A.2.
Elements with the diamond - or zincblende structure.

Element	structure	cube side a [Å]
C	A	3.57
Si	A	5.43
Ge	A	5.66
ZnS	B	5.42
GaAs	B	5.65

Table A.3.
Calculated interstitial positions.

Type	coordinates						
tetrahedral	1	1	1	1 1 1			
	0 0 -;	0 - 0;	- 0 0;	- - -;			
	2	2	2	2 2 2			
	1 1 3	1 3 1	3 1 1	3 3 3			
	- - -;	- - -;	- - -;	- - -;			
	4 4 4	4 4 4	4 4 4	4 4 4			
hexagonal	1 1 5	1 3 7	1 5 1	1 7 3	3 1 7	3 3 5	
	- - -;	- - -;	- - -;	- - -;	- - -;	- - -;	
	8 8 8	8 8 8	8 8 8	8 8 8	8 8 8	8 8 8	
	3 5 3	3 7 1	5 1 1	5 3 3	5 5 5	5 7 7	
	- - -;	- - -;	- - -;	- - -;	- - -;	- - -;	
	8 8 8	8 8 8	8 8 8	8 8 8	8 8 8	8 8 8	
	7 1 3	7 3 1	7 5 7	7 7 5			
	- - -;	- - -;	- - -;	- - -			
	8 8 8	8 8 8	8 8 8	8 8 8			
bond-centered	1 1 1	1 3 3	1 5 5	1 7 7	3 1 3	3 3 1	
	- - -;	- - -;	- - -;	- - -;	- - -;	- - -;	
	8 8 8	8 8 8	8 8 8	8 8 8	8 8 8	8 8 8	
	3 5 7	3 7 5	5 1 5	5 3 7	5 5 1	5 7 3	
	- - -;	- - -;	- - -;	- - -;	- - -;	- - -;	
	8 8 8	8 8 8	8 8 8	8 8 8	8 8 8	8 8 8	
	7 1 7	7 3 5	7 5 3	7 7 1			
	- - -;	- - -;	- - -;	- - -			
	8 8 8	8 8 8	8 8 8	8 8 8			

Table A.4.
Properties of the interstitial lattice.

Type	Number of interstitial neighbors of a lattice site	Number of lattice site neighbors of this interst.	Number of nearest interstitial neighbors
tetrahedral	4	4	4
hexagonal	12	6	6
bond-centered	4	2	6

Table A.4 continued.

Type	Distance of nearest lattice neighbors	Distance of nearest interstitial neighbors
tetrahedral	$\sqrt{3}a/4$	$\sqrt{3}a/4$
hexagonal	$\sqrt{11}a/8$	$\sqrt{2}a/4$
bond-centered	$\sqrt{3}a/8$	$\sqrt{2}a/4$

Table A.5.
VIDSIM modules.

Executable file:
vidsim.exe

Main program:
dosim.c The actual simulator program.

Sources:	(by module)
defect.h	Header for "defect" handling code
defect.c	Code to manage defect lists: add, remove, lookup
dfio.c	Code to read & write defects to/from files
dosim.c	main program for simulation
envir.h	
envir.c	Code to set up global environment for a run, and to clean up afterward. Opens and closes files, reads environment files.
event.h	
event.c	Opens, reads, writes event files
evio.c	
fundphys.h	
fundphys.c	Provides "fundamental physical" constants and functions.
global.h	
global.c	Contains definitions & initialization of objects and data types visible throughout all LM modules
mobile.h	
mobile.c	arrays & functions to operate on mobile items: vacancies and interstitials
occupant.h	
occupant.c	code to read an occupant file, and to perform services related to the lattice-site-occupant list
sim.h	
simrun.c	Code to calculate and choose a next event.
sim.c	Code to adjust the simulation state in preparation for the next event.
simdata.c	Initialized data structures for simulation.