Time-dependent Monte Carlo studies of Surface Diffusion
by Ann Marie Bowler

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Chemistry

Montana State University

© Copyright by Ann Marie Bowler (1990)

Abstract:

The fabrication of microelectronic devices is of universal interest. However, relatively little is known about the competing microscopic surface reactions involved in these thin layer growth processes. The microscopic details of the growth processes cannot be deduced from current experimental techniques because the measurements require time scales which are longer by orders of magnitude than the microscopic reaction time scales. Because many interesting and useful semiconductor devices are thermodynamically unstable, optimization of the governing kinetic parameters is desirable. We examine surface diffusion and ordering in our efforts to understand the epitaxial growth of semiconductor devices.

We examine diffusion using numerical algorithms which incorporate a kinetic treatment in conjunction with a time-dependent Monte Carlo formalism. Our method is based upon a probabilistic description of adparticle jump events. The diffusion rate is determined by the energetics of adparticle interactions on the lattice. The rare event problem associated with other theoretical techniques is overcome by our highly efficient algorithms. Consequently, we are able to observe events, including ordering and island formation, which occur on time scales which are longer by orders of magnitude than those for simple adsorbate diffusion.

We have developed algorithms to examine surface defects, steps, adparticle interactions, concentrations, etc. Our approach is readily applied to a wide range of substrate/overlayer systems. We conclude that the experimental techniques currently employed to examine diffusion are fundamentally flawed in that they fail to deduce anything about the microscopic diffusion mechanisms. Furthermore, the relatively simple picture of diffusion on which experimental analyses are based are grossly inappropriate in many situations. Even small concentrations of surface defects radically alter diffusion rates and mechanisms.

Our most recent studies have been of the III-V semiconductor (110) surfaces. We have coupled our time-dependent Monte Carlo algorithm to a simple kinematic model for LEED profiles and find that the results of our TDMC analyses are in close agreement with projections based on the macroscopic experimental behavior of these systems. We can now begin more in depth efforts to simultaneously describe the predominant competing surface reactions in epitaxial growth processes.

TIME-DEPENDENT MONTE CARLO STUDIES
OF SURFACE DIFFUSION

by
Ann Marie Bowler

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

of
Doctor of Philosophy
in
Chemistry

MONTANA STATE UNIVERSITY
Bozeman, Montana

December 1990

D378
B676

ii

APPROVAL

of a thesis submitted by

Ann Marie Bowler

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

21 December 1990
Date

Eric S. Ford
Chairperson, Graduate Committee

Approved for the Major Department

December 21 1990
Date

Edwin H. Abbott
Head, Major Department

Approved for the College of Graduate Studies

January 3, 1991
Date

Henry J. Parsons
Graduate Dean

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a doctoral degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library. I further agree that copying of this thesis is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for extensive copying or reproduction of this thesis should be referred to University Microfilms International, 300 North Zeeb Road, Ann Arbor Michigan 48106, to whom I have granted "the exclusive right to reproduce and distribute copies of the dissertation in and from microfilm and the right to reproduce and distribute by abstract in any format."

Signature Ann Bowler

Date 12-21-90

TABLE OF CONTENTS

	Page
INTRODUCTION-----	1
TIME-DEPENDENT MONTE CARLO STUDIES OF PERFECT SURFACES-----	10
Time-Dependent Monte Carlo Method for Perfect Surfaces-----	10
Representative Results for Perfect Surfaces-----	21
No Interactions and Nearest Neighbor Interactions-----	21
Nearest and Next-Nearest Neighbor Interactions-----	37
TIME-DEPENDENT MONTE CARLO STUDIES OF DIFFUSION/DESORPTION-----	49
Time-Dependent Monte Carlo Method for Diffusion/Desorption-----	49
Representative Results for Diffusion/Desorption----	50
Conclusions for Diffusion/Desorption-----	53
TIME-DEPENDENT MONTE CARLO STUDIES OF COADSORBATES-----	54
Time-Dependent Monte Carlo Model for Coadsorbates-----	54
Representative Results for Coadsorbate Systems----	64
Conclusions for Coadsorbate Diffusion-----	76

TABLE OF CONTENTS--Continued

	Page
TIME-DEPENDENT MONTE CARLO STUDIES OF STEPPED SURFACES-----	78
Time-Dependent Monte Carlo Model for Stepped Surfaces-----	78
Representative Results for Stepped Surfaces-----	82
Conclusions for Diffusion on Stepped Surfaces-----	108
COMPARISON BETWEEN PERFECT SURFACES, COADSORBATES, AND STEPPED SURFACES-----	110
TIME-DEPENDENT MONTE CARLO STUDIES OF METALS ON III-V SEMICONDUCTOR (110) SURFACES-----	113
Time-Dependent Monte Carlo Model for Metals on III-V Semiconductor (110) Surfaces-----	113
Generation of Theoretical-Low Energy Electron Diffraction Profiles-----	124
Theoretical LEED Profiles for Bi/GaAs (110) and Bi/InSb (110)-----	127
Analysis of Experimental LEED Profiles for Bi/GaAs (110)-----	134
Time-Dependent Monte Carlo Simulations of Bi/GaAs (110)-----	149
Conclusions for Bi/GaAs (110)-----	150
Analysis of Experimental LEED Profiles for Bi/InSb (110)-----	152
Time-Dependent Monte Carlo Simulations of Bi/InSb (110)-----	161
Conclusions for Bi/InSb (110)-----	164
CONCLUSIONS-----	165
LITERATURE CITED-----	168

TABLE OF CONTENTS--Continued

APPENDICES-----	174
APPENDIX A--TDMC ALGORITHM FOR PERFECT SURFACES---	175
TDMC Algorithm for Perfect Surfaces-----	176
Time Evolution of System-----	187
Testing and Modification of Code for Perfect Surfaces-----	192
Running the Diffusion Simulation for Perfect Surfaces-----	202
APPENDIX B--TDMC ALGORITHM FOR COADSORBATES-----	215
TDMC Algorithm for Coadsorbates-----	216
Running the Diffusion Simulation for Coadsorbates-----	219
APPENDIX C--TDMC ALGORITHM FOR STEPPED SURFACES---	230
TDMC Algorithm for Stepped Surfaces-----	231
Running the Diffusion Simulation for Stepped Surfaces-----	235
APPENDIX D--TDMC ALGORITHM FOR III-V (110) SURFACES-----	246
Calculation of Migration Barriers for a Substrate with Unequal Binding Energies-----	247
TDMC algorithm for Bi/GaAs (110) and Bi/InSb (110)-----	250
Running the Diffusion Simulation for Bi/GaAs-----	255

LIST OF TABLES

Table	Page
I. Diffusion Coefficients ($\times 10^{-10}$) versus time for various interaction energies. Actual diffusion coefficients can be obtained by multiplying these values by the square of the lattice constant for an appropriate system.-----	34
II. Transition Times for Particle Hops on a Stepped Surface.-----	86
III. Shifts in Maximum Intensity of Fractional Order LEED Peaks for Bi/GaAs (110) in the [1-10] Direction.-----	149
IV. Chi-Square Statistics for Generation of 500 Random Sets of Coordinates on a Rectangular Lattice.-----	194
V. Chi-Square Values for Generation of 10,000 Random Numbers.-----	195
VI. Chi-Square Statistics for Single Particle Hopping with VAX Intrinsic Generator.-----	196
VII. Best Seeds for Monte Carlo Random Number Generators.-----	196

LIST OF FIGURES

Figure	Page
1. Counting scheme for correlation functions. A-H denote equivalent sets of neighbors with increasing distance from the central particle.-----	20
2. Root mean square displacement vs. square root of time for systems with no interactions and nearest neighbor interactions. Squares denote no interactions, circles denote repulsive interactions, and triangles denote attractive interactions.-----	22
3. Initial (left) and final (right) particle maps for 0.25 ML initial coverage with no particle interactions.-----	24
4. Initial (left) and final (right) particle maps for 0.25 ML initial coverage with repulsive nearest neighbor interactions of -1.0 RT.-----	25
5. Initial (left) and final (right) particle maps for 0.25 ML initial coverage with attractive nearest neighbor interactions of 1.0 RT.-----	26
6. Site blocking study for crowded initial configurations with no interactions.-----	27
7. Diffusion coefficient vs. time for initial coverage of 0.25 ML and nearest neighbor interactions. Squares denote no interactions, circles denote repulsive interactions, and triangles denote attractive interactions.-----	29
8. Final particle map for system with an initial coverage of 0.25 ML and no interactions.-----	31
9. Diffusion coefficient vs. time for initial coverage of 0.50 ML and nearest neighbor interactions. Squares denote no interactions, circles denote repulsive interactions, and triangles denote attractive interactions.-----	34
10. Correlation functions for a) no interactions, b) nearest neighbor repulsions, and c) nearest neighbor attractions.-----	36

LIST OF FIGURES-Continued

Figure	Page
11. Initial (left) and final (right) particle maps for an initial coverage of 0.50 ML with nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 0.5 RT.-----	38
12. Initial (left) and final (right) particle maps for an initial coverage of 0.50 ML with nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 1.0 RT.-----	39
13. Initial (left) and final (right) particle maps for an initial coverage of 0.50 ML with nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 1.5 RT.-----	40
14. Initial (left) and final (right) particle maps for an initial coverage of 0.50 ML with nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 2.0 RT.-----	41
15. Diffusion coefficient vs. time for an initial coverage of 0.50 ML and nearest and next-nearest neighbor interactions. All nearest neighbor repulsions are -1.0 RT. Empty squares denote next-nearest neighbor attractions of 0.5 RT, empty circles 1.0 RT, solid squares 1.5 RT, and solid circles 2.0 RT.-----	42
16. Correlation functions for nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of a) 0.5 RT, b) 1.0 RT, c) 1.5 RT, and d) 2.0 RT.-----	47
17. Recombinatory desorption study: a) collision frequency vs. time, b) coverage vs. time, c) collision frequency vs. coverage, and d) diffusion coefficient vs. time.-----	52
18. Transition type assignment scheme for coadsorbates A and B. -----	56
19. Energy window approximation scheme for coadsorbates. -----	62
20. Initial dilute configuration of particles and traps for diffusion study.-----	66

LIST OF FIGURES-Continued

Figure	Page
21. Final configuration of particles and traps at attractive nearest and next-nearest neighbor adparticle-trap interactions of 2.0 RT, repulsive nearest neighbor adparticle interactions of -1.0 RT, and attractive next-nearest neighbor adparticle interactions of 1.5 RT.-----	67
22. Final configuration of particles and traps at attractive nearest and next-nearest neighbor adparticle-trap interactions of 2.0 RT, repulsive nearest neighbor adparticle interactions of -1.0 RT, and attractive next-nearest neighbor adparticle interactions of 2.0 RT.-----	68
23. Initial crowded configuration of particles and traps for diffusion study.-----	69
24. Final configuration of particles and traps at attractive nearest and next-nearest neighbor adparticle-trap interactions of 2.0 RT, repulsive nearest neighbor adparticle interactions of -1.0 RT, and attractive next-nearest neighbor adparticle interactions of 1.0 RT.-----	70
25. Final configuration of particles and traps at attractive nearest and next-nearest neighbor adparticle-trap interactions of 2.0 RT, repulsive nearest neighbor adparticle interactions of -1.0 RT, and attractive next-nearest neighbor adparticle interactions of 1.5 RT.-----	71
26. Final configuration of particles and traps at attractive nearest and next-nearest neighbor adparticle-trap interactions of 2.0 RT, repulsive nearest neighbor adparticle interactions of -1.0 RT, and attractive next-nearest neighbor adparticle interactions of 2.0 RT.-----	72
27. Particle-trap correlation functions for initially dilute system at a) attractive interactions of 1.5 RT and b) 2.0 RT.-----	72
28. Particle-trap correlation functions for initially crowded system at a) attractive interactions of 1.5 RT and b) 2.0 RT.-----	73

LIST OF FIGURES-Continued

Figure	Page
29. Particle-particle correlation functions for dilute system at a) attractive interactions of 1.5 RT and b) 2.0 RT.-----	74
30. Particle-particle correlation functions for crowded system at a) attractive interactions of 1.5 RT and b) 2.0 RT.-----	74
31. Diffusion coefficients for initially dilute trap systems.-----	75
32. Diffusion coefficients for initially crowded trap systems.-----	76
33. Particle map comparison for perfect and stepped surfaces.-----	83
34. Diffusion coefficients for perfect and stepped surfaces.-----	84
35. Potential particle transitions near a step.-----	85
36. Initial (left) and final (right) particle maps for diffusion on a perfect surface.-----	88
37. Particle map for stepped surface with no adparticle interactions.-----	89
38. Particle map for stepped surface with repulsive nearest neighbor interactions of -1.0 RT and attractive next-nearest neighbor interactions of 1.0 RT.-----	90
39. Particle map for stepped surface with repulsive nearest neighbor interactions of -1.0 RT and attractive next-nearest neighbor interactions of 1.5 RT.-----	91
40. Particle map for stepped surface with repulsive nearest neighbor interactions of -1.0 RT and attractive next-nearest neighbor interactions of 2.0 RT.-----	92
41. Particle map for stepped surface with no adparticle interactions.-----	94

LIST OF FIGURES-Continued

Figure	Page
42. Particle map for stepped surface with repulsive nearest neighbor interactions of -1.0 RT and attractive next-nearest neighbor interactions of 1.0 RT.-----	95
43. Particle map for stepped surface with repulsive nearest neighbor interactions of -1.0 RT and attractive next-nearest neighbor interactions of 1.5 RT.-----	96
44. Particle map for stepped surface with repulsive nearest neighbor interactions of -1.0 RT and attractive next-nearest neighbor interactions of 2.0 RT.-----	97
45. Initial (left) and final (right) particle maps for diffusion on a perfect surface.-----	98
46. X, Y, and total diffusion coefficients for parallel configurations in the presence of adparticle interactions.-----	100
47. X, Y, and total diffusion coefficients for perpendicular configurations in the presence of adparticle interactions.-----	101
48. X, Y, and total diffusion coefficients for perfect and stepped surfaces with no interactions.-----	102
49. X, Y, and total diffusion coefficients for perfect and stepped surfaces with nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 1.0 RT.-----	103
50. X, Y, and total diffusion coefficients for perfect and stepped surfaces with nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 1.5 RT.-----	104
51. X, Y, and total diffusion coefficients for perfect and stepped surfaces with nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 2.0 RT.-----	105

LIST OF FIGURES-Continued

Figure	Page
52. Diffusion coefficients vs. time for step concentration study. Ten, fifteen, twenty, and twenty-five refer to the number of terrace sites per step.-----	107
53. Comparison of diffusion rates on perfect surfaces, stepped surfaces, and surfaces with traps for nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 1.0 RT.-----	110
54. Comparison of diffusion rates on perfect surfaces, stepped surfaces, and surfaces with traps for nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 1.5 RT.-----	111
55. Comparison of diffusion rates on perfect surfaces, stepped surfaces, and surfaces with traps for nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 2.0 RT.-----	112
56. Side view of clean GaAs (110) surface. -----	114
57. Side view of Sb covered GaAs (110) surface.-----	115
58. Symmetry of GaAs surface defined by two atom basis. Hash marks between Ga and As atom pairs illustrate symmetry.-----	116
59. Distorted hexagonal transition model for III-V semiconductor (110) surfaces. Neighbor pairs around Ga and As sites are numbered 1-3.-----	117
60. Nearest and next-nearest neighbor distances for GaAs (110).-----	119
61. Unit cell model for diffusion simulations.-----	128
62. Theoretical LEED [1-10] profiles for perfect chains.-----	129
63. Theoretical LEED [1-10] profiles for longer vacancy lengths and mixtures of chain/vacancy lengths.-----	130
64. Theoretical LEED [1-10] profiles for 11 atom chains.-----	131
65. Theoretical LEED profiles for [001] direction.-----	133

LIST OF FIGURES-Continued

Figure	Page
66. (1X1) LEED pattern for clean GaAs (110). -----	135
67. Direct and reciprocal lattice vectors for GaAs (110).-----	136
68. (6X1) LEED pattern for Bi/GaAs (110).-----	137
69. Representative Voigt functional fits for experimental LEED profiles for Bi/GaAs (110).-----	143
70. FWHM vs. temperature for a) (5/6,1) and (1,1) LEED spots [1-10] direction and b) (5/6,1) spot [001] direction for 0.7 ML Bi on GaAs (110).-----	144
71. FWHM vs. temperature for a) (5/6,1) and (1,1) LEED spots [1-10] direction and b) (5/6,1) spot [001] direction for 1.0 ML Bi on GaAs (110).-----	145
72. FWHM vs. temperature for a) (5/6,1) and (1,1) LEED spots [1-10] direction and b) (5/6,1) spot [001] direction for 1.5 ML Bi on GaAs (110).-----	146
73. Time-dependent Monte Carlo generated surface map for Bi/GaAs (110). All adparticle interactions are attractive.-----	151
74. Experimental LEED profiles for Bi/InSb (110) as a function of temperature. -----	153
75. Auger data for an initial coverage of 0.8 ML Bi on InSb (110).-----	154
76. Possible structural interpretation for Bi on InSb (110) LEED data.-----	155
77. FWHM for (1,0) beam vs. temperature for Bi on InSb (110).-----	157
78. FWHM for (1,-1/2) beam vs. temperature for Bi on InSb (110).-----	159
79. FWHM for (1,-1/3) beam vs. temperature for Bi on InSb (110).-----	160
80. Possible chain ordering/disordering mechanism for Bi on InSb (110).-----	160

LIST OF FIGURES-Continued

Figure	Page
81. Surface map from time-dependent Monte Carlo simulation of Bi diffusion on InSb (110).-----	162
82. Theoretical LEED profile for TDMC map of Bi on InSb (110).-----	163
83. Transition probability assignment scheme.-----	183
84. Sites affected by adparticle hops.-----	185
85. Counting scheme for directional diffusion studies.-----	189
86. Seed study comparisons for a) no interactions, b) attractions, c) repulsions, and d) repulsive-attractive interactions.-----	201
87. Time-dependent Monte Carlo data file breakdown for perfect surface.-----	212
88. Time-dependent Monte Carlo data file breakdown for perfect surface restart.-----	213
89. Complete TDMC data file for perfect surface.-----	214
90. Complete TDMC data file for perfect surface restart.-----	214
91. Time-dependent Monte Carlo data file breakdown for coadsorbates.-----	227
92. Time-dependent Monte Carlo data file breakdown for coadsorbate restart.-----	228
93. Complete TDMC data file for coadsorbates.-----	229
94. Complete TDMC data file for coadsorbate restart.-----	229
95. Time-dependent Monte Carlo data file breakdown for surface steps.-----	243
96. Time-dependent Monte Carlo data file breakdown for surface step restart.-----	244
97. Complete TDMC data file for surface steps.-----	245
98. Complete TDMC data file for surface step restart.----	245

LIST OF FIGURES-Continued

Figure	Page
99. Time-dependent Monte Carlo data file breakdown for Bi/GaAs.-----	264
100. Time-dependent Monte Carlo data file breakdown for Bi/GaAs restart.-----	265
101. Complete TDMC data file for Bi/GaAs.-----	266
102. Complete TDMC data file for Bi/GaAs restart.-----	266

ABSTRACT

The fabrication of microelectronic devices is of universal interest. However, relatively little is known about the competing microscopic surface reactions involved in these thin layer growth processes. The microscopic details of the growth processes cannot be deduced from current experimental techniques because the measurements require time scales which are longer by orders of magnitude than the microscopic reaction time scales. Because many interesting and useful semiconductor devices are thermodynamically unstable, optimization of the governing kinetic parameters is desirable. We examine surface diffusion and ordering in our efforts to understand the epitaxial growth of semiconductor devices.

We examine diffusion using numerical algorithms which incorporate a kinetic treatment in conjunction with a time-dependent Monte Carlo formalism. Our method is based upon a probabilistic description of adparticle jump events. The diffusion rate is determined by the energetics of adparticle interactions on the lattice. The rare event problem associated with other theoretical techniques is overcome by our highly efficient algorithms. Consequently, we are able to observe events, including ordering and island formation, which occur on time scales which are longer by orders of magnitude than those for simple adsorbate diffusion.

We have developed algorithms to examine surface defects, steps, adparticle interactions, concentrations, etc. Our approach is readily applied to a wide range of substrate/overlayer systems. We conclude that the experimental techniques currently employed to examine diffusion are fundamentally flawed in that they fail to deduce anything about the microscopic diffusion mechanisms. Furthermore, the relatively simple picture of diffusion on which experimental analyses are based are grossly inappropriate in many situations. Even small concentrations of surface defects radically alter diffusion rates and mechanisms.

Our most recent studies have been of the III-V semiconductor (110) surfaces. We have coupled our time-dependent Monte Carlo algorithm to a simple kinematic model for LEED profiles and find that the results of our TDMC analyses are in close agreement with projections based on the macroscopic experimental behavior of these systems. We can now begin more in depth efforts to simultaneously describe the predominant competing surface reactions in epitaxial growth processes.

INTRODUCTION

Historically, investigations of surface diffusion have played a fundamental role in the understanding of a number of significant phenomena including the catalysis, crystallization, and annealing processes. A number of excellent reviews on the topic of surface diffusion are available.¹⁻⁸ Surface diffusion may, in many situations, dictate the natural time scale for the interaction between chemical reactants and, thus, control reaction rates.¹ The first significant evidence for the existence of lateral interactions between adsorbed species came with the observation of adsorbate ordering on single crystal metal surfaces.² Such ordering in the adsorbed layer may have a profound influence on adsorption and desorption kinetics.² Furthermore, the preference for specific adsorption sites illustrates that adsorbate migration is an activated process. Therefore, an examination of adparticle migration can provide information about the topography of the potential energy contours of a surface.² More recently, surface diffusion has been probed in an effort to gain a fundamental understanding of processes such as molecular beam epitaxial growth. This process plays a crucial role in the fabrication of semiconductor devices; but relatively

little is known about the microscopic details of the problem.¹

A variety of experimental techniques have been, and still are, used to examine surface diffusion. Among these are field emission microscopy (FEM),^{1,3,5,9-11} field ionization microscopy (FIM),^{1, 5, 12-16} laser induced thermal desorption (LITD),¹⁷⁻³⁴ low energy electron diffraction (LEED),³³⁻³⁴ and, most recently, scanning tunneling microscopy (STM).³⁵⁻⁴³ The primary experimental objective is to determine the Fick's Law diffusion coefficient for the system under investigation. This is generally done by introducing a sharp, initial concentration gradient of adsorbed gas on the substrate and subsequently monitoring the time evolution of the system. The diffusion coefficient is then obtained from an acceptable solution to Fick's second law:¹

$$\frac{dc}{dt} = \frac{d}{dx} \left(-D \frac{dc}{dx} \right) \quad (1)$$

Field emission microscopy^{1,3,5,9-11} and field ionization microscopy^{1,5,12-16} are powerful methods for examining the motion of individual atoms or atomic clusters on a surface. However, these methods provide little, if any, information about ordering in a diffusion system.⁶⁸ Similarly, low energy electron diffraction (LEED) can provide much information about whether or not an adsorbate is ordered;^{33,34}

but little information about the actual kinetics of the ordering phenomena can be ascertained. In order to obtain information about the kinetics of adsorbate ordering using LEED techniques, the sample must be repeatedly heated and cooled in rapid succession. The time scale for such a heating cycle is much longer than the time scale for diffusive motion and ordering.⁷⁶ Consequently, much information about the diffusion process is lost.

One of the more common recent probes of surface diffusion is the laser induced thermal desorption method (LITD) where an adsorbate is deposited on a substrate. A laser beam is used to rapidly desorb the adparticles from a region of the surface which has a simple, well defined geometric shape. This creates a sharp concentration gradient; and particles begin to migrate into the vacated region. After a predetermined time period, the same region of the lattice is heated once again with the laser beam. The adsorbate is desorbed once again. The amount of adsorbate which has diffused into the region of interest can be determined; and a diffusion coefficient for the process can be calculated.

Although the LITD method is currently extensively used to measure diffusion rates, it is also limited in that no specific information about lateral interactions or individual particle motion can be obtained.⁷⁵ Laser induced thermal desorption measures the combined effects of

adparticle interactions, surface steps, surface traps, and other lattice defects. Therefore, no direct, quantitative information about lateral interactions can be deduced. In fact, surface defects which significantly alter diffusion phenomena can be introduced by the repeated laser heating of the surface.

Scanning tunneling microscopy (STM) provides precise information about individual particle positions; but, once again, no information about the kinetics of ordering can be obtained from this approach due to the relatively long experimental time scale involved. We can, however, use our diffusion model to extract information about the microscopic details of the process which leads to the configuration of adparticles observed by STM experiments.

Theoretical diffusion studies employ a wide variety of techniques. These range from thermodynamic descriptions,^{1,67} to molecular dynamics simulations⁴⁴⁻⁵⁵ to transition state theory^{4,56-65} to Monte Carlo methods.⁶⁶⁻⁷⁵ Some characteristic limitations of these theoretical models are included here for purposes of comparison.

Many authors have attempted to describe surface diffusion using various thermodynamic approaches.¹ This is true of the Metropolis Monte Carlo method where only the energy differences between the initial and final states are considered in the description of adparticle hopping.⁶⁷ Such treatments ignore the fact that surface migration is an

activated process. Although the method has been somewhat successful, the true picture of the thermally activated diffusion process can hardly be assessed from such a perspective. Other thermodynamic approaches assume that diffusion (flux) occurs in response to a gradient in the chemical potential, rather than to the concentration gradient specified in Fick's first law:^{1,72}

$$J = -D \frac{dc}{dx} \quad (2)$$

Other authors have used molecular dynamics techniques utilizing classical equations of adparticle motion with appropriate interatomic potentials.⁴⁴⁻⁵⁵ This analysis, in essence, reduces to the calculation of the velocity autocorrelation function from which the diffusion constant can be ascertained. These methods have also been somewhat successful; but some major limitations are inherent. More specifically, the application of these methods is restricted to systems of only a few particles. Much computation time is spent "watching" events that never materialize into adparticle hops. This computational obstacle is known as the "rare event" problem; and it is present in a similar form in other time-dependent Monte Carlo (TDMC) techniques where much computation time is spent addressing particles which don't move during the time interval of the current

computational step.⁶⁶⁻⁶⁸

All of these methods are also fundamentally limited by the inability to provide information about adsorbate ordering or island formation. Only very small model systems have been described using these methods. Consequently, much of the predominant physics of diffusing systems has entirely evaded analysis. We have developed a unique time-dependent Monte Carlo (TDMC) approach which enables us to examine both the adatom migration and ordering processes.

Many of the difficulties inherent in the methods described above are resolved by our approach. Most importantly, the "rare event" problem has been overcome; and processes that occur over much longer time scales than diffusion, specifically overlayer ordering and growth phenomena, can be described in microscopic detail. We are also able to study systems of thousands of interacting particles. Therefore, we are able to study one of the largest theoretical model systems to date. We are able to observe both the diffusion and ordering processes. In addition, we are able to deduce the microscopic diffusion mechanisms in systems with a variety of adparticle interactions.

Our investigations begin with diffusion on perfect surfaces. Although the results for perfect surfaces are quite valuable, they describe only the most simple model for surface diffusion. In reality, the best surfaces used in

experimental studies of diffusion contain a significant amount of steps and other defects. We have extended our TDMC method to more complicated systems which include surface steps and traps or poisons.

Specific geometries for the desorption area in laser induced thermal desorption experiments are selected because the mathematical solutions to the diffusion equations are known; and a diffusion coefficient can be readily calculated. Typically, the mathematical solutions possess radial symmetry. We will demonstrate that such an analysis is inappropriate for systems with surface steps even at relatively low step concentrations.

We also examine systems of particles diffusing in the presence of a minority of traps, or entities which are strongly bound to the surface and have a significant affinity for the diffusing particles. Impurities or poisons such as these have a large influence on diffusion rates and ordering phenomena in the systems we examine.

We also have the ability to monitor the diffusion behavior of two distinct adparticle types at comparable concentrations. Studies of coadsorbate systems such as these can provide considerable insight into catalysis and chemical reactions between heterogeneous adsorbed species on a solid substrate. In addition, we examine the diffusion/recombinatory desorption process which is involved in these and many other chemical processes.

At an even greater level of complexity, we have extended our approach to examine diffusion in more complicated systems such as the (110) surfaces of the III-V semiconductors. We have developed algorithms which examine the ordering of bismuth on the GaAs and InSb (110) surfaces. The ordering and growth mechanisms in these metal/semiconductor systems are of great importance in the fabrication of microelectronics.

The primary goal of molecular beam epitaxy is to prepare a wide variety of non-equilibrium surface morphologies. This indicates that, under a wide range of epitaxial conditions, the growth of the III-V semiconductor surfaces is kinetically, rather than thermodynamically controlled. Epitaxial growth of these surfaces is controlled by fundamental chemical processes (i.e. physical adsorption, migration, dissociative chemisorption, incorporation, and desorption.) There is no comprehensive understanding of these competing microscopic processes. The details of the growth process can only be ascertained from a microscopic, theoretical analysis such as the one we have developed.

Although we only begin the process here, our goal is to understand the growth processes for the III-V semiconductor surfaces and their associated mechanisms. We will explore the relationships between adparticle interactions, reactant concentration, and gas and surface temperature. We wish, also, to determine the relationships between the growth

mechanism and the kinetic parameters.

Using the time-dependent Monte Carlo method developed here, we can begin to examine the mechanisms of crystal growth phenomena and ultimately make theoretical predictions that will allow us to optimize the growth and ordering processes.

The first step in our analysis of metals on III-V semiconductor surfaces is to extract information from the experimental LEED profiles for metals adsorbed onto these substrates. Both the position and width of the LEED profiles provide information about overlayer growth and ordering. We examine experimental LEED profiles for Bi on GaAs (110) and InSb (110). These data provide one link between experiment and theory. We have utilized an algorithm which incorporates the most simple model of LEED diffraction. Thus, we are able to compute LEED spectra for the overlayer structures produced in our TDMC analysis and use this information to understand the overlayer ordering phenomena.

The simple TDMC method described here is applicable to multitudes of diffusion systems. We are able to describe the effects of surface imperfections, adparticle interactions, concentration, temperature, kinetics, and energetics. Thus, we assert that the TDMC models described below are superior in many respects to previous theoretical techniques.

TIME-DEPENDENT MONTE CARLO STUDIES OF PERFECT SURFACES

Time-Dependent Monte Carlo Method for Perfect Surfaces

Our analysis begins with adparticles bound to a four-fold symmetric lattice. Our initial choice of substrate surface geometry is arbitrary. Our TDMC method has also been applied to more complex surface geometries with equal ease. A predetermined number of particles can be deposited either randomly or in a prescribed initial configuration on the surface. The binding energy of each adsorbed species to the surface is influenced by both the number and magnitude of its nearest and next-nearest neighbor interactions. Thus, the presence of adparticle interactions becomes of paramount importance in surface diffusion kinetics. The interactions, which are assumed to be pairwise additive, can either be attractive or repulsive. Therefore, the binding energy of a particular adatom may be higher or lower than the binding energy of an isolated adatom to the surface. Equation 3 gives the binding energy of a typical particle to the surface.

$$E_{ij} = E_b^0 - i \cdot E_{nn} - j \cdot E_{nnn} \quad (3)$$

Indices i and j are the number of nearest and next-nearest neighbor adparticles, and E_{nn} and E_{nnn} are the nearest and next-nearest neighbor interaction energies, respectively. The binding energy of an isolated adatom to the substrate is given by E_b^0 .

In our lattice gas model, migration is allowed only to vacant nearest neighbor sites. An activation barrier to migration must be overcome to accomplish each particle hop. The assumption that particles diffuse by single lattice unit nearest neighbor jumps has been verified experimentally.⁵ In addition, recent work by Cohen and Voter⁵⁷ illustrates that multiple hops only have a significant impact on diffusion dynamics at higher temperatures than those used in our analysis. In addition, multiple hops should only be of significance in a relatively small number of chemical systems. Under ordinary conditions, the desorption rate will not only compete with but will begin to dominate multiple hop diffusion at these higher temperatures. The activation barrier, which is calculated from the intersection point between harmonic potential wells centered on adjacent lattice sites, is defined in Equation 4.

$$E_{ijkl} = E_m^0 + (E_{kl} - E_{ij})/2 + (E_{kl} - E_{ij})^2/16E_m^0 \quad (4)$$

Indices i and j , respectively, refer to the number of nearest and next-nearest neighbors in the initial configuration. Indices k and l , respectively, refer to the neighbor numbers in the final configuration. The migration barrier for an isolated adatom is given by E_m^0 . The curvature of the potential is related to the frequency of the frustrated translation of the adsorbed particle. The vibrational frequency associated with this frustrated translation mode can be determined experimentally for many systems.

The first step in the diffusion simulation process is to determine all possible initial and final states for the adsorbed species and then to assign a transition type to each potential adatom hop. Each unique initial and final state combination is referred to as a transition type. For a square lattice with nearest and next-nearest neighbor interactions, there are 25 possible initial states and 25 possible final states. Hypothetically, this leads to a total of 625 transition types; but, because certain of these can be eliminated, there are actually only 196 allowed transition types. For example, because migration is only allowed to vacant nearest neighbor sites and because the initial site from which a particle migrates is necessarily left empty, a particle can never have four nearest neighbors in the initial or final states.

It is also true that there are a significant number of particle neighborhoods that differ only in the arrangement of the neighboring particles around the particle of interest. For instance, in a configuration of two nearest neighbors, the neighboring particles may either be adjacent to one another or opposite to one another. This clearly represents two non-equivalent configurations. Because the transition type includes the initial and final particle configuration, these configurational differences are explicitly included in the transition type assignments.

As the calculation proceeds, each filled site on the lattice is addressed in sequence; and each particle is placed on the population lists of its corresponding transition types. Because there are four possible directions for migration, each particle may be placed on as many as four population lists. Each transition type is characterized by an activation energy. From this activation energy, an occurrence probability is computed. The occurrence probability is simply the product of the transition type occupation number and a Boltzmann factor as represented in Equation 5.

$$P_{ijkl} = N_{ijkl} e^{(-E_{ijkl}/kt)} \quad (5)$$

Now, each transition type is addressed with a frequency proportional to its probability of occurrence. Once a transition type is selected, a particle is chosen randomly from its population list to undergo migration. Individual particles are not addressed, nor are individual lattice sites. This is a dramatic difference between our method and those discussed previously. The probabilities as determined from potential surface energetics govern the diffusion rate. Thus, an adsorbed particle migrates in each time-dependent Monte Carlo (TDMC) cycle.

When a particle reaches its final destination, the lattice occupation map and the population lists of all affected transition types are updated. A transition rate is also calculated using Equation 6 where k_m , the rate constant, is assumed to be the same for all transition types. In all of our investigations described in this dissertation, the rate constant is given a value of $1 \cdot 10^{10}$.

$$W_{ijkl} = k_m P_{ijkl} \quad (6)$$

In actuality, the rate constant is dependent on the transition type or neighborhood of the migrating particle. However, we contend that different particle neighborhoods have rate constants that differ only by a "kinetic isotope effect." More specifically, changing a particle neighborhood only slightly changes the vibrational frequency

of the particle. We argue that this change is insignificant in comparison to the differences in Boltzmann factors. Therefore, k_m is assumed to be equivalent for all transition types. The transition time for a single adparticle hop is simply the inverse of the transition rate, as Equation 7 illustrates.

$$T_{ijkl} = 1/W_{ijkl} \quad (7)$$

The sequence begins again with the selection of a new transition type and particle. When the calculation is complete, the real time for the process is computed using Equation 8.

$$time = \sum T_{ijkl} / NT \quad (8)$$

Here NT represents the total number of adparticles on the substrate surface. The mean square displacement and the diffusion coefficient are then calculated assuming random walk and Fick's law behavior. The diffusion coefficient certainly has a spatial dependence which we could compute in the diffusion simulation. However, this spatial dependence cannot be measured experimentally.

The diffusion coefficient is also dependent on the value we select for the hopping frequency, or rate constant.

Although we don't know the precise value of this parameter, a simple scaling algorithm would allow us to compute the actual diffusion coefficient given the experimentally determined value for a particular system.

No current discussion of surface diffusion is complete without the inclusion of correlation effects. Our model explicitly includes some correlation effects in the transition type assignments. More specifically, each time a particle migrates, the neighborhoods of all affected particles are updated. The new occurrence probabilities contain the "memory" of the previous hop. Therefore, a particle may be more likely to move if a neighboring particle has recently moved. The activation barrier for a second hop is clearly influenced by the first. This correlated behavior, which parallels the phenomena observed by Landman, et al.,⁷⁷ in their studies of dimer diffusion, is clearly included in the model.

Tsong and Gao⁷⁶ have also proposed that full rows of atoms migrate simultaneously in the (1X1) to (1X2) surface reconstruction of fcc (110) planes. A phenomenon such as the simultaneous reconstruction of the fcc (110) planes could clearly not be modeled using our approach. However, the conceptual and computational difficulties encountered in applying the TDMC method to surface reconstruction may center fundamentally on semantics and the definition of the term simultaneous. It is certainly possible that the

reconstruction proceeds via a series of highly correlated hopping events that occur on a time scale much shorter than that observed by the experimental probe. Thus, the time scale for particle hopping must be considered here. Typical particle hops occur in our model on a time scale of roughly a tenth of a nanosecond; and experimental time scales may be on the order of several nanoseconds.⁷⁶ Therefore, it is possible that several individual hops lead to the surface reconstruction observed above.

We do not consider the sort of correlation where a single particle, driven by its momentum in a certain direction, makes multiple hops. In our model, the particle is assumed to thermally equilibrate with the lattice after each hop. In most systems of interest, heat transport between a particle and the lattice takes place on the order of a few picoseconds. This is far faster than the atomic hopping rate; therefore, our assumption should be valid. Tully, et al.,⁵¹ concluded from molecular dynamics studies that 70% of particle hops in a system at moderate temperature were individual, uncorrelated ones.

Voter and Doll⁵⁹ also propose that, in order for a second hop to be directionally oriented, it must take place within a few vibrational periods of the first; or the memory of the event is lost. We expect thermal equilibration to take place more rapidly than this at the temperatures we are examining. As mentioned above, multiple hop correlation is

not expected to play a significant role unless the temperature is large in comparison to the activation barrier for migration. This is not true for the systems under investigation. Furthermore, because we are dealing with significant concentrations, a particle is not likely to be able to make multiple hops before it feels the influence of other adparticles. Consequently, consecutive, multiple hop correlation does not play a significant role in the adsorbate overlayer systems under investigation.

An algorithm which incorporates this diffusion methodology on perfect surfaces has been developed and tested. We currently have the capacity to study systems of thousands of particles on a lattice composed of tens of thousands of lattice sites using a VAX 8550. We have been able to simulate diffusion in systems with a variety of adparticle interactions.

The coordinates of each particle on the lattice are saved at regular intervals throughout the calculation. Thus, maps of particle locations can be generated at any phase of the diffusion process. This allows us to directly observe the influence of adparticle interactions and long range ordering. These maps of overlayer structure can also be directly compared with scanning tunnelling microscope experiments.

Another powerful probe of adsorbate ordering is the correlation function. Tabulation of correlation values is

quite similar to computing the Fourier transform of the overlayer. In order to generate the correlation function, we address each filled site on the lattice in sequence. The number of neighbors of each "type" surrounding each particle are counted according to the scheme in Figure 1. A neighbor "type" is simply defined by its displacement from the particle of interest. There are either four or eight equivalent neighbors of each type on the square substrate. The neighbor types are numbered consecutively with increasing distance from the particle of interest. Neighbor types one through eight, or A through H as they are labeled in Figure 1, are included in the analysis.

When a particle is addressed, the occupation number of neighbors one through eight are tabulated. We then move to the next filled site and add the occupation number of each neighbor type surrounding the particle of interest to the previous total.

After all of the filled lattice sites have been addressed, we have the total number of neighbor types one, two, three, et cetera. These occupation numbers are normalized. The normalization constant for each neighbor type is simply the product of the total number of particles in the system and the number of equivalent sites for that neighbor type. The resulting normalized quantity is called the correlation function. The correlation function provides an additional method for describing ordering and island

formation in the diffusion system. The correlation functions can also uniquely illustrate the effects of adparticle interactions.

Figure 1. Counting Scheme for Correlation Functions. A-H Denote Equivalent Sets of Neighbors with Increasing Distance from the Central Particle.

Using Fick's second law, we can determine the diffusion coefficient from the adparticle displacements in time. The

initial and final particle positions and the lattice jump length are used in the calculation. The distance traveled by each particle is tabulated as a function of time. The mean square displacement and root mean square displacement for the entire system of particles are then accessible. Diffusion coefficients, which can be compared to experimental values, can then be calculated. The FORTRAN algorithm for perfect surfaces is described in greater detail in Appendix A.

Representative Results for Perfect Surfaces

No Interactions and Nearest Neighbor Interactions

The diffusion coefficient for a two dimensional random walk is given by Equation 9.⁴

$$\sqrt{\langle R^2 \rangle} = \sqrt{4Dt} \quad (9)$$

A plot of the root mean square displacement versus the square root of time for systems with no interactions and with nearest neighbor attractive or repulsive interactions at an initial coverage of 0.25 is illustrated in Figure 2. Initial and intermediate particle maps for these systems are presented in Figures 3 through 5. Particles begin migrating in these systems from the initially crowded configurations

on the left. The sharp concentration profiles used in these computer simulations closely model the experimental methods described previously.¹

Figure 2. Root mean square displacement vs. square root of time for systems with no interactions and nearest neighbor interactions. Squares denote no interactions, circles denote repulsive interactions, and triangles denote attractive interactions.

One of the most striking observations for these systems lies in the initial nonlinearity of the root mean square displacement versus square root of time curves. This

curvature corresponds to a reduced initial rate of diffusion. This nonlinear behavior persists in each system until the initial particle pack becomes dilute enough that random walk conditions are approximated. Furthermore, this curvature is entirely absent in an initially dilute system. One possible explanation for these observations is that the particles on the edge of the diffusion front initially act as a wall and hinder the migration of the particles farther back in the pack. Thus, we refer to this as a site blocking effect.

Additional computer simulations for a variety of adparticle coverages reveal that the initial curvature in D is not a function of the initial coverage. These data are presented in Figure 6. Consequently, it seems most reasonable that we are observing a column by column dissolution of the initial particle pack. In that case, the first column of the initial particle pack would diffuse away from the diffusion front, followed by the second column, et cetera. Thus, the number of particles in each column would be of little consequence; and we would not expect the curvature to be coverage dependent.

Figure 3. Initial (left) and final (right) particle maps for 0.25 ML initial coverage with no particle interactions.

Figure 4. Initial (left) and final (right) particle maps for 0.25 ML initial coverage with repulsive nearest neighbor interactions of -1.0 RT.

Figure 5. Initial (left) and final (right) particle maps for 0.25 ML initial coverage with attractive nearest neighbor interactions of 1.0 RT.

Figure 6. Site blocking study for crowded initial configurations with no interactions.

Although plots of the root mean square displacement versus square root of time provide a convenient measure of adparticle displacements in time, it is difficult to determine the actual diffusion coefficients based solely on this representation of diffusion. This computation difficulty, which is demonstrated by Equation 9, arises because the diffusion coefficient is related to the square of the slope of such a curve. Thus, we provide an alternate, more easily interpreted representation of the time-dependent behavior of interacting diffusing systems.

If we solve for the diffusion coefficient, D , in Equation 9, we find that it is related to the mean square displacement and the time. By monitoring the diffusion coefficient as a function of time, we can determine the magnitude and range of effects due to crowding and adparticle interactions in diffusing systems. All of the diffusion coefficients are expressed in arbitrary units. These values can be multiplied by the square of the lattice vector for an appropriate system to yield actual diffusion coefficients.

The influence of lateral interactions is dramatically emphasized in Figure 7. It is evident that repulsive nearest neighbor interactions significantly enhance the diffusion rate while attractive interactions significantly retard the diffusion rate. Distinct differences in the particle maps for these systems at intermediate times are also obvious. Figures 3 through 5 present initial and intermediate maps of the overlayer for a non-interacting system and for systems with attractive and repulsive interactions at initial coverages of 0.25 ML. Maps, such as those illustrated here, derived from our TDMC simulation allow us to clearly observe the effects of adparticle interactions and the time-evolution of ordering or disordering in the overlayer. In the system with repulsive nearest neighbor interactions, there are fewer adjacent particles than in the system with no interactions. One might predict that strong repulsions in the initial crowded configuration of particles would enhance

the dissolution of the particle pack. This proves to be the case.

Figure 7. Diffusion coefficient vs. time for initial coverage of 0.25 ML and nearest neighbor interactions. Squares denote no interactions, circles denote repulsive interactions, and triangles denote attractive interactions.

Conversely, when attractive interactions are present, small clusters or nearest neighbor islands are observed in the particle maps. An equilibrium configuration of islands lingers until long times; and the system with attractive interactions never achieves the maximum diffusion rate that the repulsively interacting system achieves. The system with

no interactions is intermediate between the two extreme cases.

Figure 7 also shows the effect of site blocking in the initial particle pack. The initial diffusion rates for the system with no interactions and the system with repulsive interactions are initially slowed. The rates increase as site blocking becomes less important. Surprisingly the diffusion coefficients pass through a maximum before they begin to approach the random walk limit at long times. As the initial particle pack begins to disperse, more empty sites become available. At these intermediate coverages, the repulsive interactions govern the behavior of the diffusing particles. The diffusion rate increases while the continuing dilution of the system reacts to reduce these interatomic repulsions. The diffusion rate reaches a maximum value at some critical concentration and then falls off as the concentration becomes more dilute. At long times, the system approaches the limit of infinite dilution where the adparticle interactions no longer play a role in the diffusion process.

It is interesting to note that there is also a maximum rate of diffusion in the system with no adparticle interactions. In fact, this maximum rate actually exceeds the random walk diffusion rate. Site blocking is known to reduce the initial diffusion rate. However, as the particles migrate onto the open lattice, we should tend toward the

maximum, or random walk, limit of diffusion. How, then, is it possible that the maximum rate of diffusion exceeds this limit? We have determined that this is an artifact of the initial configuration of particles. Random walk statistics apply only to radially symmetrical systems. We have discovered that our systems do not entirely behave in a symmetric fashion.

Figure 8. Final particle map for system with an initial coverage of 0.25 ML and no interactions.

We initially speculated that the infinite barrier on the left edge of the particle pack might break the radial symmetry of the process and, therefore, cause the increase in the diffusion rate. This hypothesis was tested by placing the initial strip of particles in the center of the lattice rather than against the edge of the lattice. The same behavior was observed in the diffusion coefficient. Thus, the infinite barrier on the left edge of the particle pack does not appear to be the culprit.

In these crowded initial configurations, there are available binding sites all along one or both edges of the pack. Particles are free to migrate onto a completely empty substrate in the horizontal direction. However, crowding almost entirely eliminates diffusion in the vertical direction. Examination of the diffusion coefficients for each direction indicate that diffusion onto the open lattice is considerably faster than diffusion in the direction perpendicular to it.

Instead of traveling in both directions initially, particles are essentially traveling in only one direction. Thus, they are displaced by larger distances. For instance, if a particle makes 7 hops in the horizontal direction, its displacement is 7 times the jump length. If the same particle makes 3 hops in the horizontal direction and 4 hops in the vertical direction, its net displacement is only 5 times the jump length. Therefore, this initial pseudo one

dimensional diffusion results in larger displacements and an unexpected increase in the diffusion rate as computed for two dimensional diffusion. As the concentration becomes more dilute, the diffusion coefficients in each direction show the same rate of change with time. The acceleration in the diffusion coefficient is entirely absent in a system which is initially very dilute.

Figure 8 shows the particle map for the non-interacting system in the long time limit. At long times, there are no qualitative differences in the particle maps for these systems. However, quantitative differences in the diffusion coefficients persist until very long times as the lateral interactions continue to influence the diffusion rates. At long times and infinite dilution, we expect the three curves in Figure 7 to converge. This convergence is indeed observed in the long-time simulation limit.

The results for initial coverages of 0.50 with the same interactions discussed here follow the same definitive trends with the exception that the islands in the system with attractive interactions are slightly larger at intermediate times. These results are presented in Figure 9.

Table I summarizes the results for diffusion under the variety of interactions examined in these studies.

TABLE I. Diffusion coefficients ($\times 10^{-10}$) versus time for various interaction energies. Actual diffusion coefficients can be obtained by multiplying these values by the square of the lattice constant for an appropriate system.

time (seconds)			1.0×10^{-8}	2.0×10^{-8}	4.0×10^{-8}	6.0×10^{-8}	8.0×10^{-8}
coverage	E_{nn}	E_{nnn}					
0.25	0	0	5.32	7.40	8.53	7.09	5.48
0.25	-RT	0	7.19	10.64	9.40	7.46	5.69
0.25	+RT	0	2.14	3.03	4.30	4.82	5.00
0.50	0	0	9.94	15.12	16.87	13.99	11.57
0.50	-RT	0	18.83	23.96	21.17	16.05	10.34
0.50	+RT	0	3.73	5.15	7.69	8.35	8.88

Figure 9. Diffusion coefficient vs. time for initial coverage of 0.50 ML and nearest neighbor interactions. Squares denote no interactions, circles denote repulsive interactions, and triangles denote attractive interactions.

One method of probing overlayer ordering is to tabulate correlation functions for the particle configuration on the lattice. This method is similar to a Fourier transform technique. Figure 10a illustrates a representative pattern that emerges in the correlation function in the absence of adparticle interactions. This function was derived from the particle map saved from the non-interacting, initially crowded system at 0.5 ML. The map was saved after a relatively small number of particle hops. Consequently, the result is illustrative of a relatively crowded system of particles. Figure 10b and 10c also illustrate representative correlation patterns for nearest neighbor attractive and repulsive interactions of 1.0 RT at an initial coverage of 0.5 ML.

When no adparticle interactions are present, there is no variation in the peak intensity for neighbors 1 through 8. This indicates that the particles are distributed in a completely random fashion across the lattice.

When repulsive nearest neighbor interactions are present, the magnitude of the first neighbor correlation peak is significantly reduced relative to the remaining peaks. Thus, we can clearly observe that the repulsive interactions are directing the particles away from configurations that leave them surrounded by nearest neighbors. The intensities of the first and second correlation peaks increase considerably with attractive

nearest neighbor interactions as the particles begin to coagulate. Large nearest neighbor islands should produce more intense peaks for the more distant neighbors. Consequently, we might speculate that the average size of the clusters in this attractive system is small. Only small nearest neighbor clusters are observed in the particle map for this system.

Figure 10. Correlation functions for a) no interactions, b) nearest neighbor repulsions, and c) nearest neighbor attractions.

The correlation intensities for the interacting systems begin to level off as the particle concentrations become more dilute. The influence of the interactions becomes less important with increasing distance between adparticles. At long times, there are no qualitative differences in the correlation functions for the interacting and non-interacting systems. The intensities of the correlation peaks in all of these systems tend toward zero at long times and infinite dilution. Therefore, these functions prove only to be instructive in relatively crowded adparticle configurations.

Nearest and Next-Nearest Neighbor Interactions

Next, systems with both nearest and next-nearest neighbor interactions were considered. As is the case in some real systems, the nearest neighbor interactions were given repulsive values; and the next-nearest neighbor interactions were assigned attractive values. The magnitude of the nearest neighbor repulsions was held constant at $-1.0 RT$ while the magnitude of the next-nearest neighbor interactions was gradually increase from $0.5 RT$ to $2.0 RT$. R is the ideal gas constant in calories; and T is the temperature in Kelvin. These systems, which began migrating from a perfect island configuration, appear in Figures 11 through 15.

Figure 11. Initial (left) and final (right) particle maps for an initial coverage of 0.50 ML with nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 0.5 RT.

Figure 12. Initial (left) and final (right) particle maps for an initial coverage of 0.50 ML with nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 1.0 RT.

Figure 13. Initial (left) and final (right) particle maps for an initial coverage of 0.50 ML with nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 1.5 RT.

Figure 14. Initial (left) and final (right) particle maps for an initial coverage of 0.50 ML with nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 2.0 RT.

As these figures indicate, the diffusion rate is reduced as the magnitude of the next-nearest neighbor interaction energy is increased. As the attractive forces between particles become larger, island formation becomes clearly visible in the lattice maps. The decrease in the diffusion coefficient is monotonic with the exception of the system diffusing with attractive interactions of 1.5 RT.

Figure 15. Diffusion coefficient vs. time for an initial coverage of 0.50 ML and nearest and next-nearest neighbor interactions. All nearest neighbor repulsions are -1.0 RT. Empty squares denote next-nearest neighbor attractions of 0.5 RT, empty circles 1.0 RT, solid squares 1.5 RT, and solid circles 2.0 RT.

In the long time limit, the system with interactions of 2.0 RT diffuses more rapidly than the system with interactions of 1.5 RT. A possible explanation for the crossing of the diffusion coefficient versus time curves for these systems is that island formation begins beyond the initial diffusion front in the system at 1.5 RT. This is clear from close examination of the particle map for this system. At the same time in the system at 2.0 RT, relatively few particles are found beyond the initial diffusion front--most are trapped within the large island domain. At 1.5 RT, many particles have migrated onto the open lattice. At some critical coverage, cluster formation begins beyond the initial diffusion front. The net result is a reduction in the diffusion rate. This phenomenon is also observed at 2.0 RT at longer times when the concentration of particles in the sea reaches the critical value for cluster formation. The somewhat intuitive nature of these results further substantiates the validity of the time-dependent Monte Carlo approach.

At next-nearest neighbor interaction energies of 2.0 RT, a meniscus effect is observed along the edges of the lattice. Pockets are also vacated along the diffusion front. Under these conditions, the diffusion mechanism is clearly not the column by column dissolution discussed earlier. At these high next-nearest neighbor attractive energies, it is most energetically favorable for the particles to stay in a

next-nearest neighbor configuration. As particles migrate away from the diffusion front, they continually rearrange to maintain as many next-nearest neighbors as possible. Once particles have moved beyond the initial diffusion front in a next-nearest neighbor fashion, those behind them can move easily into the recently vacated next-nearest neighbor positions. This leads to the formation of the next-nearest neighbor outcroppings or "fingers" observed in the particle maps. It can be as much as five times faster for a particle to move along a finger than it is for a particle to break away from the smooth diffusion front. It is also easier for a particle to break away from the tip of a finger than it is to break away from the smooth front because the particle is moving away from one attractive neighbor instead of two. Along the perfectly smooth diffusion front, a particle has at least two next-nearest neighbors in its initial configuration. When the particle breaks away, it is left with no next-nearest neighbors. Therefore, the effect of the attractive interactions is completely eliminated in the transition. This is not an energetically favorable hop. On the other hand, if a particle moves along the edge of a finger it can achieve a configuration where there is only one next-nearest neighbor. Breaking away from one attractive neighbor is less costly energetically. Consequently, we see a preference for transitions from the end of a finger or a particle outcropping rather than from

the smooth diffusion front. Consequently, this pocket, "finger," or droplet formation along the original diffusion front becomes quite efficient.

The correlation functions for these systems with repulsive and attractive interactions are also quite interesting. Figures 16a, 16b, 16c, and 16d present the correlation functions for the systems with attractive interactions of 0.5 RT, 1.0 RT, 1.5 RT, and 2.0 RT, respectively. The formation of large islands can more easily be detected in the correlation functions for these systems with next-nearest neighbor attractive interactions. These systems have repulsive nearest neighbor interactions of -1.0 RT. At the low attractive interactions in Figures 16a and 16b, the only definite feature is a reduction in the nearest neighbor correlation peak. This is certainly due to the repulsive nearest neighbor interactions. There is only a slight preference for next-nearest neighbor sites at these low attractions.

As the attractions become larger in Figures 16c and 16d, we begin to see a definite preference for next-nearest neighbor configurations. In fact, when the interactions are increased to 2.0 RT, we see exclusively next-nearest neighbor occupation. It is interesting to note the distinct pattern of peaks at neighbors 2,3,5,7, and 8 for this nearly perfect island. The particle map for this system at attractive interactions of 2.0 RT is comprised of one large

island and relatively few loose particles.

It is also interesting to note that the correlation function can provide a qualitative estimate of the concentration of defects "within" the islands. There are clearly peaks at neighbors 4 and 6 at attractions below 2.0 RT. There is little nearest neighbor occupation in any of these systems. Therefore, these peak intensities do not seem to correspond to atoms trapped within the next-nearest neighbor island domains. Instead, the intensities at positions 4 and 6 might be characteristic of domains of next-nearest neighbor occupation that are out of phase. The complete absence of defects within the island at 2.0 RT is partially an artifact of the initial perfect island configuration. At lower attractions, the particles diffuse more easily and can create these phase domains at a reduced energy penalty.

We can also get a qualitative estimate of island size at the higher attractive interactions. The correlation intensities for the more distant neighbors 7 and 8 are clearly greater at 2.0 RT than they are at lower interactions. However, it may not be appropriate to make quantitative comparisons between these data because the relative coverages are changing constantly. The systems at lower attractive interactions spread out onto the open lattice. The system at 2.0 RT remains relatively undisturbed in its initial configuration.

Figure 16. Correlation functions for nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of a) 0.5 RT, b) 1.0 RT, c) 1.5 RT, and d) 2.0 RT

Correlation functions provide a unique way to observe overlayer ordering. The influence of adparticle interactions is clearly represented. We can also obtain information about cluster and island formation. This information, however, is

primarily qualitative. The concentration in these systems is constantly changing; and all correlation functions tend toward zero at low concentrations. Therefore, it seems most appropriate to use correlation functions to observe short time adparticle configurations in these initially crowded systems. It may be less valuable to extract information from the correlation functions at longer times; and it is certainly inappropriate to attempt to derive quantitative information from these systems as the concentration constantly changes at a different rate in each.

TIME-DEPENDENT MONTE CARLO STUDIES OF DIFFUSION/DESORPTION

Time-Dependent Monte Carlo Method for Diffusion/Desorption

In order to examine the effects of surface migration on recombinatory desorption, the diffusion algorithm was modified to monitor surface "collisions" and to randomly remove particle pairs from the surface upon collision. Desorption proceeds when the particles begin to migrate; and the first collisions of moving particles occur.

After each particle hop, the number of nearest and next-nearest neighbors that the moving particle encounters are tabulated. A collision occurs when the migrating particle encounters a nearest neighbor. If one or multiple collisions occur, then the desorption subroutine is called. Each nearest neighbor particle pair at the collision site is assigned an equal probability window based on the total desorption probability. A random number is generated. If this number falls within one of the windows containing a particle pair, the pair is removed from the surface. If the random number does not fall within the probability region occupied by particle pairs, no desorption occurs.

The collision frequency is stored as a function of time throughout the calculation. The system presented here was composed initially of 1000 non-interacting particles. The lattice dimensions, which were 25 X 200 lattice units, remained constant throughout the simulation. A desorption probability of 0.0005 was used in the analysis.

The root mean square displacement is calculated for each time interval using the average number of particles on the lattice during that time interval. The displacement of particles that desorb during a given interval are only considered during that interval. The coordinates of these particles just prior to desorption are used in the calculation.

Representative Results for Diffusion/Desorption

As illustrated in Figure 17, the collision frequency decreases linearly as the coverage is depleted through desorption. The coverage also decreases linearly with time in this case. It appears, as well, as if the collision frequency has a direct concentration proportionality. This last result, obtained from a detailed Monte Carlo calculation, agrees well with the results obtained from the following simple, phenomenological model of the diffusion of an ideal gas confined to two dimensions. An adatom (hard sphere) on the two dimensional lattice with a diameter, d ,

travelling at a velocity, v , will sweep out a rectangular area of length, l , in time, t . Thus the area swept out in time t is given by Equation 10.

$$\text{area} = d \cdot v \cdot t \quad (10)$$

Any particle whose center is located in this rectangle will collide with the migrating particle. If the surface concentration of particles is c , then the collision frequency can be expected to be:

$$\text{collision frequency} = d \cdot v \cdot c \quad (11)$$

The collision frequency is directly proportional to the cross-sectional area of the particle projected onto the surface plane, the particle velocity, and the surface concentration of adsorbate.

Figure 17d illustrates the diffusion coefficient vs. time. There is a rapid decrease in the diffusion coefficient at very short times. Diffusion then proceeds at roughly the same rate for the duration of the simulation. The rapid decrease in D results from desorption of the most energetic adparticles in the system.

The particles with the lowest migration barriers diffuse most rapidly and are more likely to suffer collisions than the slower particles with high migration barriers. The more a particle collides with neighboring particles, the more

Figure 17. Recombinatory desorption study: a) collision frequency vs. time, b) coverage vs. time, c) collision frequency vs. coverage, and d) diffusion coefficient vs. time.

likely it is to desorb. Thus, we wind up depleting the population of the particles in the high energy portion of the Boltzmann distribution. The system is left with the slower moving particles which collide and eventually achieve a Boltzmann distribution of energies again. The sharp decline in D results from the rapid depletion of the "hot" particles.

Conclusions for Diffusion/Desorption Systems

The diffusion/desorption algorithm has allowed us to determine the coverage dependence of the collision frequency. Desorption certainly competes effectively with diffusion at elevated temperatures; and this algorithm can be used to examine several significant systems such as the diffusion of Bi on InSb (110). A temperature program has not been designed. However, a small amount of programming would allow us to increase the temperature of the surface and the desorption probability as a function of time. These additional modifications would allow us to more closely reproduce temperature-programmed thermal desorption spectra and monitor reaction rates where recombinatory desorption is involved.

TIME-DEPENDENT MONTE CARLO STUDIES OF COADSORBATES

Time-Dependent Monte Carlo Model for Coadsorbates

Coadsorbed systems provide another interesting avenue to explore in our investigation of surface diffusion. Poisons or traps can have a significant effect on the diffusion rate. The time-dependent Monte Carlo method certainly provides valuable insights into the diffusion mechanism in the presence of these impurities. Several significant modifications, which are described below, were required to adapt the original TDMC diffusion algorithm to these more complicated systems.

The most significant challenge that we encountered in developing the model for mixed adsorbates was the enumeration of the two-species transition types. In the original model, there are 196 distinguishable, allowed transition types when nearest and next-nearest neighbor interactions are considered. Each allowed type is referenced by four indices that correspond to the number of nearest and next-nearest neighbors in the initial and final states.

When two distinct adsorbed species are present on the lattice, eight indices are required to describe the initial and final states. Based on this initial indexing, there are

a total of 65,536 theoretical particle configurations. In order to determine which of these transition types were physically meaningful, two separate algorithms were developed. These algorithms were comprised of a series of logical statements designed to eliminate all, or at least the majority, of the impossible configurations.

The possibilities were first reduced by imposing the stipulation that the sum of the nearest neighbor A and B atoms in the initial state must be less than or equal to three. The number of next-nearest neighbor A and B atoms must be less than or equal to four. These criteria are also valid for the final configurations. The initial sweep of 256 possible distinct configurations ruled out all but 150. Thus we were left with a reduced total of 22,500 theoretical transition types.

The remaining configurations were further screened using the criteria developed from a visual model. For instance, if no nearest neighbor A atoms are present in the initial configuration, then a maximum of 2 next-nearest neighbor A atoms can be present in the final configuration, and so on. PROGRAM EXCLUDE contains the actual relationships between particle species and neighborhoods used in this elimination process. This second elimination process narrowed the list of allowed adparticle neighborhoods to 13,608. Because the energy of a transition depends on whether an A atom or a B atom is moving, the

Figure 18. Transition type assignment scheme for coadsorbates A and B.

actual number of transition types is 27,216.

VAX FORTRAN allows a maximum of seven dimensions for a single array. Eight indices would be required to identify the neighborhoods of the two species transition types by the previous method if both nearest and next-nearest neighbor interactions are included. Consequently, a new referencing system had to be developed. As stated above, there are 150

potential initial or final particle neighborhoods. However, not all of the combinations of these represent allowed transitions. The 150 unique configurations were numbered and stored in a matrix. The designated number for a configuration is indexed by the number of each species and their position with respect to the particle of interest. This number, which describes a particle neighborhood, is accessed as the first phase in the transition type assignment.

The 13,608 allowed configurations were scanned. The index, 1-150, corresponding to the neighbor numbers in the initial state was referenced. The index corresponding to the final state was also referenced. A transition type number was then assigned and stored in a 150 X 150 X 2 matrix. The numbers 1-13,608, as illustrated in Figure 18, correspond to A atom neighborhoods; and the numbers 13,609-27,216 correspond to B atom neighborhoods.

The particle counting scheme also takes a considerably more complex form in this version of the algorithm because the lattice must still be scanned for filled sites. In addition, the identity of each atom must also be determined. When particles are placed on the lattice, they are assigned numbers which identify them as species A or B. After each neighbor is counted and identified, the indices corresponding to the initial and final neighborhoods are referenced. The transition number is then determined. Error

messages have been included in the counting scheme to insure that the calculation will stop in the event that an allowed transition is missing from the list.

Because storage and data manipulation would be very restrictive and time consuming if each of the allowed transition types was addressed individually, a new procedure was developed for random particle selection. Instead of randomly selecting a particular transition type, the algorithm was designed to randomly select particles based on their energies. There are generally many degeneracies among the energies of transition types. Consequently, all transition types with identical, or nearly identical, energies can be placed on one energy list. Particles are subsequently selected from these energy lists. This is essentially the same criterion used in the original algorithm. The only major difference is that multiple transition types are included in each energy list.

Particles are still randomly selected from the energy lists. Now the particle can belong to any of the transition types on the energy list. Before the simulation begins, each transition type is assigned to an energy list. All of the particles of that type are placed on the list. The hopping process is then governed directly by energy, rather than transition type. When particle neighborhoods change, the particles are placed on the appropriate new energy lists. This process substantially reduces the amount of storage

space required for the simulation and greatly simplifies the particle and transition type bookkeeping schemes.

The energy subroutine also had to undergo major modifications to accommodate the coadsorbate energy expressions, and to comply with the energy selection procedure described above. The energies of A and B atom transitions must be computed separately due to the cross terms between different particle types. The activation energies must also be compiled separately to allow for different binding energies of A and B atoms to the substrate.

The energy expressions for particle hops and the activation energies for the two species are given by Equations 12-15. Indices i, j, k, and l represent the neighbor occupation numbers in the initial configuration. Indices m, n, o, and p characterize the final configuration. E_{MA} is the binding energy of species A to the substrate. E_{MB} is the binding energy of species B to the substrate. As before, negative values of the interaction energies correspond to repulsive interactions. Positive values correspond to attractive interactions.

$$EHOP_A = -(m \cdot A_{nn} + n \cdot AB_{nn}) - (o \cdot A_{nnn} + p \cdot AB_{nnn}) \\ + (i \cdot A_{nn} + j \cdot AB_{nn}) + (k \cdot A_{nnn} + l \cdot AB_{nnn}) \quad (12)$$

$$EACT_A = E_{MA} + EHOP_A/2 + EHOP_A^2/16 E_{MA} \quad (13)$$

$$EHOP_B = -(m \cdot B_{nn} + n \cdot AB_{nn}) - (o \cdot B_{nnn} + p \cdot AB_{nnn}) \\ + (i \cdot B_{nn} + j \cdot AB_{nn}) + (k \cdot B_{nnn} + l \cdot AB_{nnn}) \quad (14)$$

$$EACT_B = E_{MB} + EHOP_B/2 + EHOP_B^2/16 E_{MB} \quad (15)$$

As stated above, the large number of transition types prohibits the storage of exact transition energies and Boltzmann factors for each type individually. Therefore, the minimum and maximum energy transitions are used to define an energy regime for the system of interest. This energy interval is then divided into 100 smaller windows of equal dimension. Each transition type is assigned to the appropriate energy window. The allowed transition energies for A and B atoms are computed in two separate algorithms. Transitions for A atoms occupy energy lists 1-100; and B atom transitions occupy energy lists 101-200.

After initial testing, it became clear that enough error was introduced by the energy window approximation to invalidate the program timer. Not only was the clock affected by the error in the width of the energy windows, but also by the inclusion of multiple transition types

within each window. As it turns out, the occupation number of the entire energy window is not necessarily representative of the occupation number of a specific transition type. The calculated time for particle jump events was found to be inaccurate based on comparisons with previous exact results.

The error introduced by assigning the transition types to energy windows was not significantly improved by increasing the number of energy lists and, therefore, reducing the width of the energy window. Consequently, a more complicated scheme was devised to reduce the error. After transition types are assigned to the appropriate energy lists, the lists are scanned and the number of occupied lists is determined. If the total number of occupied lists is small, the lists are further subdivided. For instance, if 25 lists are occupied, each window can be broken into 4 smaller windows.

The new energy scale is not continuous. Instead, it is fashioned in the same manner as the transition probability scale. The endpoints of each list are referenced. Only the occupied lists are further subdivided. Once the new subdivisions have been made, the transition types are reassigned to smaller windows. Then the total occupation number is tabulated again. If the occupied number of energy lists is still small, the windows are divided again. In this manner, the error in the energy window approximation can be

reduced to almost any predetermined value. This scheme is illustrated in Figure 19.

Figure 19. Energy window approximation scheme for coadsorbates.

The utility of the method of successively subdividing the energy windows is contingent on the degeneracy of a particular system of particles. When multiple degeneracy exists and only a few of the energy lists are occupied, the energy windows can be made so small that they almost represent the exact transition energy values. If, on the other hand, most of the energy lists are occupied at the outset, the reduction in the energy error will not be as effective. More lists can be added to accommodate such systems; but, as stated above, the error reduction will be

The utility of the method of successively subdividing the energy windows is contingent on the degeneracy of a particular system of particles. When multiple degeneracy exists and only a few of the energy lists are occupied, the energy windows can be made so small that they almost represent the exact transition energy values. If, on the other hand, most of the energy lists are occupied at the outset, the reduction in the energy error will not be as effective. More lists can be added to accommodate such systems; but, as stated above, the error reduction will be considerably less satisfying in this case.

In order to correct the second source of error in the clock, the occupation number of each transition type must be tabulated throughout the calculation. A more accurate transition rate can then be calculated by multiplying the transition probability for the energy list by the fraction of particles of a specific type on the list, as shown in Equation 16: This avoids the possibility of calculating a very large transition rate when the occupation number of a specific type in the energy list is small. The time for the process is then calculated using Equation 17.

$$W = k_m \text{PROB}_{\text{ETYPE}} (ON_{\text{TTYPE}} / ON_{\text{ETYPE}}) \quad (16)$$

$$\text{time} = \Sigma T / \text{NPART} \quad (17)$$

behavior of each species and the interactions between different species on the lattice. A more complete description of the coadsorbate algorithm is provided in Appendix B.

Representative Results for Coadsorbate Systems

A trial run was conducted using the coadsorbate algorithm where the experiment was confined to a single adsorbate species. This allowed us to verify the validity of the energy approximation by comparing the results with those from the original, exact version of the diffusion simulator. We wanted to determine the magnitude of the error that results from incorporating the energy window approximation. The agreement between the coadsorbate algorithm and the algorithm for perfect surfaces is quite good, but not exact. We find that the "absolute" diffusion coefficients are slightly different. However, the time rate of change in the diffusion coefficients is the same. The small difference here probably results from the difference in the hopping order of the particles. The energy lists are occupied differently than transition type lists. Consequently, even if the same random number seeds are used in the two trials, particles will still be selected in a different order. This will slightly affect the initial diffusion coefficient. Thus, the difference here, which is quite small, is not believed to be a result of the energy

list approximation but, rather, the hopping order. What we actually see is a difference similar to those illustrated in Appendix A for different random number seeds.

Our initial investigations of coadsorbates involve adparticles, as before, and traps which have a high affinity for the adparticles. The traps are bound to the surface and, for all practical purposes, are immobile. Difficulties were encountered with the program timer due to the large binding energies of the poisons or traps relative to the adparticle binding energies. We discovered initially that we were not using binding energies sufficient to "stick" the traps to the surface, as desired. We also determined that, with a higher migration barrier, the movement of a trap is a very unlikely event. Therefore, it is not appropriate to add the time for a trap hop into the total time for the simulation unless we run the simulation for times which are orders of magnitude longer than those required to monitor adparticle diffusion. For instance, we are looking at total simulation times on the order of tens of nanoseconds for adsorbate diffusion. The time for a single trap hop would be on the order of one one hundredth of a second. A single trap hop would disguise any other information about adparticle diffusion. Thus, in the unlikely event that a trap should hop, we do not include the time for the hop in the data.

Figure 20 shows the initial configuration for a surface trap or poison experiment. A trap or poison, as discussed

here, is an entity which is bound very strongly to the surface, has an extremely low rate of migration, and has a relatively high affinity for migrating particles. The traps in these studies are the larger markers. Poisons dramatically reduce surface diffusion, and certainly chemical reaction rates, in some systems.

Figure 20. Initial dilute configuration of particles and traps for diffusion study.

Figures 21 and 22 show the final configurations for two different sets of adparticle interaction energies. The trap binding energy and affinity for adparticles are held

constant. The traps have attractive nearest and next-nearest neighbor attractions for adparticles. We have introduced repulsive nearest neighbor interactions and attractive next-nearest neighbor interactions between the adparticles. The attractions in Figure 21 are 1.5 RT; and in Figure 22 they are 2.0 RT.

Figure 21. Final configuration of particles and traps at attractive nearest and next-nearest neighbor adparticle-trap interactions of 2.0 RT, repulsive nearest neighbor adparticle interactions of -1.0 RT, and attractive next-nearest neighbor adparticle interactions of 1.5 RT.

We see in both figures that the traps act as nucleation sites. At attractive adparticle interactions of 1.5, the traps collect particles primarily as nearest neighbors. The adparticle attractions are insufficient to cause long range ordering; and diffusion proceeds as usual after trap titration. It is more obvious in Figure 22, where higher adparticle attractions are present, that the traps titrate and subsequently promote island formation. The ordering begins to extend far beyond the range of influence of the traps.

Figure 22. Final configuration of particles and traps at attractive nearest and next-nearest neighbor adparticle-trap interactions of 2.0 RT, repulsive nearest neighbor adparticle interactions of -1.0 RT, and attractive next-nearest neighbor adparticle interactions of 2.0 RT.

Figure 23 shows an initially crowded configuration of particles on a surface with randomly distributed traps. There are the same number of adparticles and traps here as there were in the previous case. In this case, we have attractive next-nearest neighbor interactions of 1.0 RT, 1.5 RT, and 2.0 RT. The final particle maps for these studies are displayed in Figures 24-26. We see that, at low attractive adparticle energies the traps become titrated, as before; but the adparticle interactions are insufficient to promote additional island growth.

Figure 23. Initial crowded configuration of particles and traps for diffusion study.

On the other hand, as we increase the attractions between particles to 2.0 RT, we find that we titrate the traps near the diffusion front to form small islands. Because the adparticles are in such close proximity, we see the development of very large islands.

Figure 24. Final configuration of particles and traps at attractive nearest and next-nearest neighbor adparticle-trap interactions of 2.0 RT, repulsive nearest neighbor adparticle interactions of -1.0 RT, and attractive next-nearest neighbor adparticle interactions of 1.0 RT.

Figure 25. Final configuration of particles and traps at attractive nearest and next-nearest neighbor adparticle-trap interactions of 2.0 RT, repulsive nearest neighbor adparticle interactions of -1.0 RT, and attractive next-nearest neighbor adparticle interactions of 1.5 RT.

Figures 27 and 28 show the adparticle-trap correlation functions for the initially dilute (Figures 21 and 22) and initially crowded (Figures 25 and 26) configurations at 1.5 RT and 2.0 RT. We see that, in the dilute configurations, there are more particles bound to traps than there are in the crowded configurations. This is consistent with the fact that the particles have more access to the traps in the initially dilute configuration.

Figure 26. Final configuration of particles and traps at attractive nearest and next-nearest neighbor adparticle-trap interactions of 2.0 RT, repulsive nearest neighbor adparticle interactions of -1.0 RT, and attractive next-nearest neighbor adparticle interactions of 2.0 RT.

Figure 27. Particle-trap correlation functions for initially dilute system at a) attractive interactions of 1.5 RT and b) 2.0 RT.

Figure 28. Particle-trap correlation functions for initially crowded system at a) attractive interactions of 1.5 RT and b) 2.0 RT.

Figures 29 and 30 present the particle-particle correlation functions for the dilute and crowded configurations. There is little difference between the dilute and crowded configurations at 1.5 RT because the adparticle interactions are insufficient for island formation. However, we clearly see the significance of island formation in the crowded configuration. There is also ordering beyond the range of influence of the traps in the dilute configuration. However, there is not a sufficient concentration of adparticles around any trap to promote long range ordering.

Figure 29. Particle-particle correlation functions for dilute system at a) attractive interactions of 1.5 RT and b) 2.0 RT.

Figure 30. Particle-particle correlation functions for crowded system at a) attractive interactions of 1.5 RT and b) 2.0 RT.

Figures 31 and 32 show the diffusion coefficients vs. time for the dilute and crowded configurations, respectively. We see a rapid decrease in the diffusion coefficients in Figure 31 as the particles begin to titrate the traps. Once the titration of most traps is complete, the diffusion coefficients level off. The diffusion coefficients for the dilute systems are significantly higher than those for the systems in Figure 32. Figure 32 also illustrates, once again, that the island formation induced by the traps at 2.0 RT has a profound influence on diffusion dynamics in this system.

Figure 31. Diffusion coefficients for initially dilute trap systems.

Figure 32. Diffusion coefficients for initially crowded trap systems.

Conclusions for Coadsorbate Diffusion

The studies presented here demonstrate that surface traps can have both short and long range effects on the diffusion dynamics. In the presence of adparticles with low attractive energies, the traps titrate quickly; and then diffusion proceeds as it might in a system with no surface traps. We also find that the initial configuration of adparticles in a system with traps has a profound influence on the diffusion rate and mechanism.

We predict that, in the limit of very, very long times, the systems in which the adparticles are initially crowded will become similar to those which are initially dilute. This poses another interesting question about the laser induced thermal desorption experiment. In this experiment, a sharp initial concentration gradient is introduced by desorbing adatoms from a section of the lattice. The diffusion coefficient is measured as the particles on the diffusion front migrate into the desorbed region. Our studies indicate that there must be careful consideration about the time scale for measurement. There are clearly different diffusion mechanisms and diffusion rates at short and long times.

TIME-DEPENDENT MONTE CARLO STUDIES OF STEPPED SURFACES

Time-Dependent Monte Carlo Model for Stepped Surfaces

Stepped surfaces provide another more realistic foundation on which to build additional understanding of surface diffusion phenomena. Although our results for perfect surfaces were quite valuable, they describe only the most simple model for surface diffusion. In reality, the best surfaces used in experimental studies of diffusion contain a significant concentration of surface steps and other defects. Consequently, we have also extended our time-dependent Monte Carlo method to treat these more realistic surfaces.

The same basic procedures described for coadsorbates were also used to adapt the diffusion algorithm to accommodate stepped surfaces. In this case, two maps of the surface are used to assign transition types. One map contains the particle locations in two dimensions, as before. The second contains a map of the topography, or Z coordinate, at each surface site. The Z coordinate at each value of X and Y is incorporated into this map before the calculation begins. The map of the Z direction, or steps, remains unchanged during the calculation. It is, however,

utilized each time a transition type is assigned. The step interval can be input from the data file if regular step intervals are used. Other algorithms can also be developed to produce different surface features.

Difficulties similar to those discussed for coadsorbates were also encountered in this case when transition types were enumerated. In order to keep the dimensions of the program at a realistic level, some restrictions on the geometry of the surface had to be imposed. The code can currently accommodate smooth parallel steps which run along the Y direction. These steps can either be spaced evenly or unevenly. Each step can also either be a step up or a step down. Because the initial configuration of particles can be placed in any fashion relative to the steps, we can actually monitor the effects of step features in just about any configuration relative to the diffusing particles. The restrictions imposed on the surface geometry exclude very jagged or discontinuous surface features. The code can, however, be used to explore many surface structures. More complicated geometries would require that (many) additional transition types be added to the current scheme.

While the counting scheme for the stepped surface was being designed, it became clear that some additional criteria had to be imposed to limit the number of transition types. Thus, an additional approximation was incorporated.

If filled and empty step sites have distinguishable influences on an adjacent particle, the transition type would be based on nearest neighbors, next-nearest neighbors, filled step sites, empty step sites, steps up, and steps down. The assignment of transition types under these conditions would become nearly impossible.

In order to further restrict the number of transition types, the following procedure was used. When the particle neighborhood is defined, surface steps have a higher priority than particles. The magnitude of the energy barrier for all step sites is the same, regardless of whether they are filled or empty. The particles are not allowed to move to filled step sites. They are aware of the presence of an adjacent step site regardless of whether it is occupied or not, though. This approximation will be valid as long as the magnitude of the step barrier is considerably larger than the magnitudes of the particle interactions and migration barriers. The transition type is now a function of nearest neighbors, next-nearest neighbors, and step sites. The steps can either go up or down. Step interactions have priority over particle interactions.

In an interesting experiment, we ran the diffusion algorithm with surface steps with no step barriers. We discovered that the result was not physically meaningful because the symmetry of the problem is broken by introducing this geometric feature with no associated energy barrier.

The TDMC model is based on energetics. We are unable to identify a geometric distortion without an associated energy. This result is consistent with the criterion in the previous paragraph that states that the model is valid only when the step barrier is larger than the magnitudes of the adparticle interactions.

The number of allowed transition types in this simplified model is 8,804. It should also be noted that, in keeping with the restrictions described above, there is no provision in the algorithm to explicitly allow for a change in the coordination number for a particle at the base of a step. This change of coordination or binding energy is, however, included explicitly in the activation barrier at the step site.

The energy window approximation was also incorporated in the step algorithm. The same scheme described previously is used to reduce the error introduced by using energy windows of finite width. The transition type occupation number is used in the program timer, as before. The energy for a particle transition is given by Equation 18 where the sum of the absolute values of indices i and j (and k and l) is less than or equal to four. E_{nn} is the nearest neighbor particle interaction energy. S_{nn} is the nearest neighbor step barrier. E_{nnn} is the next-nearest neighbor particle interaction energy. S_{nnn} is the next-nearest neighbor step barrier. The step barrier is always a positive number. The

surface geometry map dictates whether the energy increases or decreases at the step. As always, negative particle interactions are repulsive.

$$E_{ijkl} = -(i \cdot E_{nn} + j \cdot S_{nn}) - (k \cdot E_{nnn} + l \cdot S_{nnn}) \quad (18)$$

The root mean square displacement is calculated in three dimensions using the maps of particle positions and surface geometry. The diffusion coefficients for the stepped surfaces also have to be calculated in three dimensions. The two and three dimensional expressions for the diffusion coefficient differ only by a multiplicative constant. The correlation functions are calculated, as before, for the adsorbed particles. The number of particles at the bottom and top of the steps is also tabulated. A more thorough description of the algorithm for surface steps is presented in Appendix C.

Representative Results for Stepped Surfaces

A trial run with no particle interactions and no surface steps indicates, once again, that no significant error is introduced by the energy window approximation. The comparison between particles diffusing on a surface with no steps and a surface with ten terrace sites per step provides a more interesting observation. We adjusted the energy for

terrace and step hops such that the diffusion rate for terrace hops would be ten times the rate for step hops. Our expectation was that, overall, the particles on the stepped surface would migrate at roughly one half the rate of those on the flat surface. Figure 33 shows the particle maps for the two trials. The particles on the stepped surface make much less progress in a great deal more time. Figure 34 shows the diffusion coefficients for the two systems.

Figure 33. Particle map comparison for perfect and stepped surfaces.

Figure 34. Diffusion coefficients for perfect and stepped surfaces.

The particles on the stepped surface migrate at a much slower rate than we anticipated. The number of particles beyond the first step is surprisingly small. A more careful analysis of the stepped surface revealed that the tendency is for particles to stay at the base of a step. The steps influence the migration barrier for not one, but two, lattice sites. A particle "remembers" the step until it is two lattice units away.

Figure 35 illustrates some possible transitions for a single particle on a stepped surface. Table II contains the time constants for each of the labeled transitions. These numbers indicate that, even if the particle hops up a step, it is more likely to be pulled back into the potential well at the base of the step than to continue migrating across the lattice. Therefore, the rate at which a particle migrates beyond the step is slower than we originally anticipated.

Figure 35. Potential particle transitions near a step.

Perhaps a more interesting trend in these transition probabilities is the tendency for motion parallel to the steps. The time constants for motion parallel to the steps

are smaller than the times for both of the transition events that represent motion perpendicular to the steps. Motion parallel to the steps is favored regardless of whether the particle is at the top or the bottom of a step. Therefore, it appears that the most probable journey for a particle at the top of a step is migration along the edge of the step and an eventual return to the potential well at the base of the step. This trend explains the low concentration of particles beyond the first step in our study.

Table II. Transition Times for Particle Hops on a Stepped Surface.

Transition	Time ($\times 10^{10}$)
a	0.550
b	1.263
c	36.643
d	0.824
e	13,880.000
f	10.649
g	0.824
h	36.643
i	1.263
j	0.824
k	1.649

We would also expect particles to titrate the step sites and, if possible, to continue hopping parallel to the steps. Particle maps clearly illustrate this phenomenon in our model systems. This is precisely the behavior that occurs in real systems where one dimensional ordering

parallel to the surface steps is observed. We also expect steps to act as nucleation sites when particle interactions are significant enough to initiate island formation.

There is also a significant factor of two difference between two and three dimensional diffusion coefficients. The diffusion coefficient for the system without steps is calculated in two dimensions. The stepped surface is treated as three dimensional. However, a huge barrier to diffusion is present at the step sites; and diffusion in this direction cannot really be considered the same. In fact, when step concentrations are high, diffusion is reduced to one dimension.

Figure 36 presents the particle picture for diffusion on a perfect surface. Figures 37-40 present the particle maps for stepped systems with increasing attractive adparticle interactions. The steps are represented by dotted lines in the vertical direction. The particles begin migrating from a dense configuration which is initially parallel to the surface steps. This will be referred to as the parallel configuration in the ensuing discussion. Note that "parallel" describes only the orientation of the initial particle pack relative to the steps, not the primary direction of diffusion.

The particle maps illustrate that the step sites do, indeed, act as particle traps. Particles can easily be seen lining up at the bottoms of the step edges. There is also a

"corralling" or "herding" effect where particles are primarily confined to one region of the lattice. The steps act as the dividing lines between distinctly different concentration regimes. As the particle interactions increase, more ordering at the step edges becomes apparent. The steps can clearly act as nucleation sites for particle clusters.

Figure 36. Initial (left) and final (right) particle maps for diffusion on a perfect surface.

Figure 37. Particle map for stepped surface with no adparticle interactions.

Figure 38. Particle map for stepped surface with repulsive nearest neighbor interactions of -1.0 RT and attractive next-nearest neighbor interactions of 1.0 RT.

Figure 39. Particle map for stepped surface with repulsive nearest neighbor interactions of -1.0 RT and attractive next-nearest neighbor interactions of 1.5 RT.

Figure 40. Particle map for stepped surface with repulsive nearest neighbor interactions of -1.0 RT and attractive next-nearest neighbor interactions of 2.0 RT.

Figures 41-44 are quite similar. However, the particles began migrating in these systems from configurations that were initially perpendicular to the surface steps. This is referred to as the perpendicular particle configuration in the following discussion. Figure 45 shows the analog for the perfect surface. Trends similar to those observed for the parallel configuration are observed here. The exception is that there is more nucleation at the step edges in this configuration. The particles initially encounter more step edges than they do in the parallel configuration. Thus we would predict more nucleation.

Figures 46 and 47 show the directional and total diffusion coefficients for the parallel and perpendicular configurations as a function of adparticle interactions. The Y direction is the direction parallel to the surface steps. The X direction is perpendicular to the steps. Figures 48-51 compare the diffusion coefficients for the perfect surface with those for the parallel and perpendicular configurations for each set of interaction energies. A number of significant trends are evident.

Figure 41. Particle map for stepped surface with no adparticle interactions.

Figure 42. Particle map for stepped surface with repulsive nearest neighbor interactions of -1.0 RT and attractive next-nearest neighbor interactions of 1.0 RT.

Figure 43. Particle map for stepped surface with repulsive nearest neighbor interactions of -1.0 RT and attractive next-nearest neighbor interactions of 1.5 RT.

Figure 44. Particle map for stepped surface with repulsive nearest neighbor interactions of -1.0 RT and attractive next-nearest neighbor interactions of 2.0 RT.

Figure 45. Initial (left) and final (right) particle maps for diffusion on a perfect surface.

In the parallel configuration (Figure 46) at low interactions, the Y diffusion coefficient (the diffusion coefficient in the direction parallel to the steps) is accelerated relative to the X diffusion coefficient. This

trend is the exact opposite of what is observed for diffusion on the perfect surface. Diffusion in the Y direction is obviously enhanced by the surface steps. This agrees with our previous calculation which indicated that diffusion parallel to the steps would be enhanced at the step edges.

At higher interaction energies, when ordering in the initial configuration becomes prevalent, the differences between X and Y diffusion coefficients in Figure 46 are reduced. In fact, the diffusion rates in the two directions become almost equal. The diffusion coefficient decreases, as we have previously seen, as the interaction energies increase. The cross-over in diffusion coefficient that we observe between the systems with $E_{nnn}=1.5$ RT and $E_{nnn}=2.0$ RT on the perfect surface is absent here.

At low interaction energies in the parallel configuration, the diffusion coefficient is a sharply rising function initially. However, it abruptly "rolls over" and attains a constant value. This change presumably occurs when particles begin to encounter the first step. At higher interaction energies, this effect is absent, presumably due to the fact that fewer particles have advanced far enough to encounter a step. At higher interactions, the rate limiting step is escape from the large island domain. The steps have, for all practical purposes, not even participated in the diffusion process yet.

Figure 46. X, Y, and total diffusion coefficients for parallel configurations in the presence of adparticle interactions.

Figure 47. X, Y, and total diffusion coefficients for perpendicular configurations in the presence of adparticle interactions.

Figure 48. X, Y, and total diffusion coefficients for perfect and stepped surfaces with no interactions.

Figure 49. X, Y, and total diffusion coefficients for perfect and stepped surfaces with nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 1.0 RT.

Figure 50. X, Y, and total diffusion coefficients for perfect and stepped surfaces with nearest neighbor repulsions of $-1.0 RT$ and next-nearest neighbor attractions of $1.5 RT$.

Figure 51. X, Y, and total diffusion coefficients for perfect and stepped surfaces with nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 2.0 RT.

The initial rapid rise in the Y diffusion coefficient is also absent in the perpendicular configuration in Figure 47. The step edges are encountered immediately in that case. In the perpendicular systems, there is more initial exposure of the particles to the step edges. A corresponding decrease in the initial diffusion coefficients relative to those for parallel systems is observed. This is illustrated in Figure 48. Once again, particles are observed to coagulate at the step edges. Diffusion in the direction of the steps is enhanced. It is not clear, however, whether this acceleration is due to increased mobility along the step edges or due to the fact that there is less site blocking in the Y direction in this initial configuration. There is certainly some contribution to the rate acceleration from both factors.

At the end of our simulation, the Y diffusion coefficients in Figures 48 and 49 are still rising in the perpendicular systems with low interactions. Although the overall diffusion coefficients for these systems are initially slower than their parallel counterparts, the Y diffusion coefficients continue rising as particles race along the step edges. There is a cross-over in the Y diffusion coefficients for the parallel and perpendicular cases. The Y, and overall, diffusion coefficients in the perpendicular cases are faster in the long run.

The diffusion coefficients are much slower in all of the stepped systems than they are for the corresponding systems on the perfect surface. Thus, surface steps prove to have a profound influence on diffusion phenomena. The final diffusion rates for the perpendicular systems tend to be faster than their parallel analogs at low interactions. This further illustrates the ability of the steps to directionally enhance the diffusion rate.

Figure 52. Diffusion coefficients vs. time for step concentration study. Ten, fifteen, twenty, and twenty-five refer to the number of terrace sites per step.

These effects are not observed at higher interactions where the overall diffusion rates for parallel and perpendicular systems become almost identical.

Figure 52 shows the diffusion rate as a function of step concentration. The particles were in the usual crowded configuration on the left edge of the lattice. The steps were placed at intervals ranging from ten to twenty-five terrace sites. Clearly, diffusion is rapid until the particles encounter the step edges. We observe that, in the long time limit, the diffusion coefficients converge.

Conclusions for Diffusion on Stepped Surfaces

Surface steps prove to significantly retard the diffusion rates relative to rates on perfect surfaces. The steps also directionally enhance the diffusion rate, corral particles, and act as nucleation sites. It is obvious from these studies that the laser induced thermal desorption experiment is flawed by its inability to treat surface steps. The interpretation of the experimental data assumes that the system possesses radial symmetry. The particles are assumed to diffuse into the vacated region at the same rate from any direction. The symmetry of the problem is clearly broken when surface steps are present. No treatment that fails to discuss this directionality in the diffusion coefficient is sufficient. There are clearly preferred

directions for diffusion from our step studies. There are also a large number of surface steps on even the most perfect surfaces available. Consequently, the effects of surface steps must be considered in any comprehensive analysis of surface diffusion.

COMPARISON BETWEEN PERFECT SURFACES, COADSORBATES, AND STEPPED SURFACES

It seems appropriate at this juncture to compare our results for diffusion under perfect and imperfect conditions. Figures 53-55 illustrate the diffusion coefficients for systems with repulsive nearest neighbor and attractive next-nearest neighbor interactions. The attractive interactions increase from 1.0 RT to 2.0 RT in going from Figures 53-55.

Figure 53. Comparison of diffusion rates on perfect surfaces, stepped surfaces, and surfaces with traps for nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 1.0 RT.

Diffusion is clearly strongly affected by the presence of both surface steps and traps. The diffusion rate on the perfect surface is always higher than the rate on the imperfect surfaces. At low attractive interactions, traps have a smaller influence on the diffusion rate than the surface steps. At high attractive interactions, steps and traps produce roughly the same rate of migration.

Figure 54. Comparison of diffusion rates on perfect surfaces, stepped surfaces, and surfaces with traps for nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 1.5 RT.

Of course the diffusion rates are influenced by the affinity of the traps for the adparticles and the step concentration and barrier height. We can however conclude that the influence of surface steps and traps must be considered in any description of surface diffusion.

Figure 55. Comparison of diffusion rates on perfect surfaces, stepped surfaces, and surfaces with traps for nearest neighbor repulsions of -1.0 RT and next-nearest neighbor attractions of 2.0 RT.

**TIME-DEPENDENT MONTE CARLO STUDIES OF METALS
ON III-V SEMICONDUCTOR (110) SURFACES**

Time-Dependent Monte Carlo Model for Metals
on III-V Semiconductor (110) Surfaces

Overlayer systems of metals diffusing on III-V semiconductor surfaces are quite interesting and lend themselves quite nicely to inspection by our time-dependent Monte Carlo approach. Our investigations of these systems provide a foundation on which we will build our understanding of the molecular beam epitaxial growth process. This section describes the development of diffusion algorithms which are specifically tailored to the study of these systems. We are particularly interested in the Bi/GaAs and Bi/InSb systems. However, because the behavior of Sb on GaAs has been characterized in far greater detail than that of Bi on GaAs or InSb, we will use Sb/GaAs as our prototype. The structure of the GaAs (110) surface and the nature of the antimony overlayer will be briefly discussed. We will then disclose the details appropriate to a TDMC model for this system. The development of the actual FORTRAN algorithms will be outlined in Appendix D.

The GaAs (110) surface has been the focus of much attention in the recent literature. In addition, the adsorption process for antimony on GaAs (110) has been

characterized quite well. The GaAs (110) surface is conjugated, as illustrated in Figure 56, with alternating rows of first and second layer gallium and arsenic atoms.

Figure 56. Side view of clean GaAs (110) surface.

When antimony is adsorbed onto the surface with subsequent thermal annealing, a highly ordered overlayer which produces a sharp (1x1) LEED pattern is observed. Antimony is strongly bound to the surface. However, it has very high mobility at elevated temperatures.^{78,79} The observed LEED pattern corresponds to the formation of zig-zag chains of antimony along the [1-10] direction. The adatoms lie directly above second layer gallium and arsenic atoms. The ordered antimony overlayer, which possesses the same symmetry as the surface, is illustrated in Figure 57.

Figure 57. Side view of Sb covered GaAs (110) surface.

At first glance, the symmetry required to produce a (1X1) LEED pattern seems to be absent in the zig-zag configuration of surface atoms. However, we can easily understand the origin of the symmetry in the LEED pattern if we recognize the two atom basis set illustrated in Figure 58.

There is no evidence for Sb-substrate disruption or interdiffusion. It is, however, known that the arsenic binding sites are preferred over gallium sites. Consequently, at submonolayer coverages, the truncated antimony chains on the surface are generally terminated at arsenic atoms.⁷⁸

Figure 58. Symmetry of GaAs surface defined by two atom basis. Hash marks between Ga and As atom pairs illustrate symmetry.

As we began to contemplate a theoretical model for this highly irregular surface, we proposed that binding sites in adjacent rows could be considered in hexagonal arrays, as illustrated in Figure 59. This hexagonal configuration provides the most simple model for adatom binding and surface migration below monolayer coverage. It also provides for both interchain and intrachain transitions. This six-fold coordination scheme provides the basis for the diffusion algorithms described in this chapter. It is also important to note that the entire scheme described below is valid for any systems with symmetrical, hexagonal geometries.

For the hexagonal binding scheme in Figure 59, there are two non-equivalent binding sites, Ga and As. In addition, there are three non-equivalent sets of "nearest" neighbors at each binding site. This situation leads to 27 possible nearest neighbor configurations. In turn, this leads to 729 theoretical, unique initial-final neighborhood combinations. However, the difference in binding sites significantly reduces the degeneracy of the problem. There are actually 2916 (729×4) theoretical transition types in the final analysis for this irregular geometry.

Figure 59. Distorted hexagonal transition model for III-V semiconductor (110) surfaces. Neighbor pairs around Ga and As sites are numbered 1-3.

If next-nearest neighbors were included in this counting scheme, there would be in excess of two million theoretical transition types. If it were even possible to enumerate all of the allowed transitions, the bookkeeping process would never be possible. Therefore, no next-nearest neighbor interactions have been incorporated into the III-V diffusion algorithms. We propose, however, that the "next-nearest" neighbor positions are distant enough from the particle of interest in this distorted hexagonal configuration that they do not significantly influence adparticle migration. Figure 60 illustrates the relative distances between a particle and its "nearest" and "next-nearest" neighbors on GaAs (110). The distances of the "next-nearest" neighbors are exaggerated by the distortions in this "hexagonal" lattice and are even more distant than they would be in the perfect hexagon. Furthermore, it is reasonable to conclude that the observed one dimensional antimony chain formation is exclusively facilitated by the "nearest" neighbor interactions included in this model. A particle actually feels the influence of particles that are one and two sites away along the chain. The particle two sites away might actually more appropriately be called a next-nearest neighbor.

Figure 60. Nearest and next-nearest neighbor distances for GaAs (110).

The elimination of impossible transition types was carried out by a procedure that was completely analogous to the ones described previously for coadsorbates and steps. "Generic" allowed final configurations were tabulated for each of the 27 initial neighborhoods. The term generic is used to describe the possible combinations of neighbors 1-3 in the initial and final states without regard to the identity of the binding site. A FORTRAN algorithm was developed to do the actual elimination and counting of these transition types. In the final analysis, there are 427 distinct configurations of neighbors. When the different

combinations of binding sites are considered, there are 1708, or four times as many, possible transition types for Sb on GaAs (110). Once again, appropriate error statements have been incorporated into the code to guard against missing transition types.

Two differences between this counting scheme and previous ones are that 1) the binding site (Ga or As) in the initial and final configuration must be determined and 2) the number of each of the three specific neighbor types at each site must also be determined to correctly assign a transition energy. Care must also be used when assigning interaction energies because the "neighborhood" is different at Ga and As sites. The numbering scheme for the three sets of non-equivalent neighbors around Ga and As binding sites is illustrated in Figure 59. The final distinction between this and previous counting schemes is that each particle may now be found on six transition type lists because there are six avenues for motion from each site. However, the "long" hops between different rows of binding sites will have higher energies and represent a minority of actual particle transitions. This seems to be consistent with the observed one dimensional diffusion mechanism. The predominant mechanism for diffusion should be motion along a single row of GaAs during which particle encounters lead to chain formation. Our model certainly accounts for such a mechanism.

The energy list formalism described previously has been used once again in this situation because the large number of transition types prohibits each one from being considered separately. Each combination of binding site transitions (As-As, Ga-Ga, As-Ga, Ga-As) is accounted for in its own energy algorithm. Each energy algorithm assigns transition types to 50 energy lists. Previous algorithms assigned transition types to 100 or 200 lists. The accuracy of the energy window approximation is reduced as the number of lists becomes smaller. If accuracy becomes an issue, however, the program can always be redimensioned to reduce the energy error.

The different binding sites on the substrate also necessitated that the migration barrier expression be re-evaluated to accommodate migration from one binding type to the other. This effort, which leads to a generalized form of the migration barrier equation, is presented in Appendix D. The limiting forms of these generalized equations produce the appropriate expression for the single binding energy which has been used until this point. Equations 19 and 20 give the binding energies for particles on As and Ga sites. E_{nAs} and E_{nGa} are the binding energies of isolated adatoms to As and Ga sites on the surface. Indices i, j, k , etc. refer to the numbers of neighbors 1-3 and As_{nn1} , As_{nn2} , etc. refer to the interaction energies. For two distinct binding energies, the expression for the migration barrier becomes

equal to the difference between the energies at the final and initial locations, as Equation 21 demonstrates.

$$E_1 = E_{mAs} + i \cdot As_{nn1} + j \cdot As_{nn2} + k \cdot As_{nn3} \quad (19)$$

$$E_2 = E_{mGa} + l \cdot Ga_{nn1} + m \cdot Ga_{nn2} + n \cdot G_{nn3} \quad (20)$$

$$E_m = E_2 - E_1 \quad (21)$$

When a particle is selected for motion, ten other particles are affected by the transition. In the four-fold symmetric model with nearest and next-nearest neighbors, there were twelve affected particles. Thus, in this scheme, particle moves should require less computation time. On the other hand, there are now six directions for motion on the GaAs (110) lattice. Transition type storage and energy list assignments will require more than the previous number of computational steps.

The distorted hexagonal geometry and the different binding sites do not permit trajectory lengths to be tabulated as easily as they were for the square geometry. A new matrix has been introduced into the code to keep track

of each particle's motion in each direction. The root mean square displacement can be calculated from this matrix if the appropriate jump lengths are known. The "raw" trajectories will also be output to a file. That way the diffusion coefficient can also be calculated for different geometries. The only time when a specific geometry need be assigned is when the displacements or diffusion coefficients are calculated. All other data manipulations, and the subsequent diffusion behavior, are based strictly on the energies of interaction between particles.

The final breakdown of our previous formalism occurs in the tabulation of a correlation function. It is not practical to calculate a correlation function for Sb on the GaAs (110) lattice due to the relative absence of symmetry. This lattice should, however, lend itself directly to LEED analysis. A simple model for calculating LEED spectra will be discussed in the next section.

In summary, the model described here is applicable to any system that can be portrayed as "hexagonal." The particle interactions account for distortions in the symmetry of such a geometry. There is also generally no restriction on the adsorbate. In the Bi/GaAs system, however, there is a limit on the length of the chains in the [1-10] direction. The model is also easily adapted to treat this system. More details about the FORTRAN code for these systems is provided in Appendix D.

Generation of Theoretical Low Energy
Electron Diffraction Profiles

An algorithm which allows us to generate LEED spectra from the maps of particle positions produced in our time-dependent Monte Carlo simulations is discussed in this section. It is necessary that we be able to generate LEED spectra for the particle maps derived from the TDMC simulation for our studies of the III-V semiconductor growth processes. The FWHM analysis of LEED profiles provides one link to examining the microscopic details of the migration process. In fact, the LEED analysis provides a very valuable link between experiment and theory. We ultimately hope to be able to determine the TDMC kinetic parameters through this analysis. The kinematic LEED analysis of surface geometry consists of measuring the Fourier transform of a periodic array of surface atoms. Theoretically, the parameter to be computed is the exponential of the dot product of the reciprocal lattice vector with the distance, as Equation 22 illustrates.

$$I(k) = \left[\sum_{x=0}^L e^{ikx} \right]^2 \quad (22)$$

$I(k)$ is the intensity, k is the reciprocal lattice vector, and x is the distance vector.

An algorithm that computes the Fourier transform of an array of adsorbate atoms on any surface has been developed previously. This algorithm has recently been used to examine Bi overlayers on the GaAs (110) and InSb (110) lattices.

We provide the particle coordinates and the incident beam energy. The computation consists of the following steps. First, values for the X and Y components of the wave vector are selected. We can use either degrees or inverse angstroms for the wave vectors. The value for the Z component of the wave vector is derived from the wave vectors in the X and Y directions and the incident beam energy. At each value of the wave vector, the dot product between the wave vector and the coordinates of each filled site on the lattice is computed. This dot product is exponentiated and added into a running total. After all filled sites have been addressed, this sum of exponentials is squared to get the intensity at the wave vector of interest. The intensity is stored; and a new value for the

wave vector begins the process again.

This algorithm describes only the most simple of scattering events. We do not include scattering from the substrate, geometrical structure factors, atomic form factors, or multiple scattering. Experimental LEED is certainly much more complicated; and we can't expect to develop a complete understanding of the experimental profiles under investigation.

However, we have examined several overlayer configurations on the GaAs (110) and InSb (110) surfaces. The simple theoretical model proves to be quite valuable in our surface diffusion studies.

The wave vector, k , in reciprocal space is related to the distance in real space as

$$k=2\cdot\pi/d \quad (23)$$

The GaAs lattice constants are 4.0 and 5.6 angstroms in the [1-10] and [001] directions, respectively. Thus, the integral order diffraction peaks in reciprocal space should appear at 1.57 inverse angstroms in the [1-10] direction and 1.11 inverse angstroms in the [001] direction. This proves to be the case for a monolayer coverage on the (110) surface.

Theoretical LEED Profiles for Bi/GaAs (110)
and Bi/Insb (110)

Bismuth is known to adsorb on the GaAs (110) surface and form zig-zag chains in the [1-10] direction with random nucleation. The clean GaAs (110) surface produces a 1X1 LEED pattern. Upon exposure to bismuth, a 6X1 LEED pattern appears. Scanning tunneling microscope experiments reveal that bismuth forms zig-zag chains with a repeat distance of 24 angstroms on the GaAs (110) surface in the [1-10] direction. Antimony also forms zig-zag chains on GaAs (110). There is, however, no constraint on the length of the Sb chains. The chain length for bismuth is limited due to the 9% increase in atomic radius of bismuth. The 6X1 LEED pattern is believed to evolve from a repeating chain structure with a unit cell vector that is six times the length of the conventional unit cell vector.

We began our analysis with perfect chains on the (110) substrate in the [1-10] direction. We studied a number of chain lengths with a number of vacancies and determined that the fractional order diffraction peaks appeared at the predicted wave vector values. Figure 61 illustrates three sets of truncated Bi chains with vacancies between them. As stated previously, the GaAs (110) surface is comprised of a two atom basis. The chains in Figure 61 are ten atoms long

this configuration is 24 angstroms. Therefore, we would expect a 6th order peak at $\pi/12$ inverse angstroms.

Figure 61. Unit cell model for diffusion simulations.

Figure 62c proves that this is the case. The entire series of fractional order peaks for an overlayer structure do not always appear in these spectra. Several other examples are illustrated in Figure 62. All of these produce fractional order peaks that correspond to the appropriate dimensions of the repeating unit cell length.

Figure 63 presents more LEED profiles. In cases a and b, we held the chain length constant and introduced more vacancies between chains. The LEED peaks still appear at the wave vectors corresponding to repeat length of the large unit cell. We have also produced spectra for mixtures of 10 atom chains and vacancies. In these cases, we see peaks corresponding to both unit cell lengths.

Figure 62. Theoretical LEED [1-10] profiles for perfect chains.

Figure 63. Theoretical LEED [1-10] profiles for longer vacancy lengths and mixtures of chain/vacancy lengths.

Figure 64. Theoretical LEED [1-10] profiles for 11 atom chains.

There is some controversy as to the exact configuration of Bi atoms that produces the 6th order diffraction peaks. The configuration of 10 atom Bi chains with two vacancies in Figure 61 produces an appropriate LEED pattern. It has also been proposed that 11 atom chains might produce 6th order peaks; so three configurations of 11 atom chains and vacancies were examined. These results are presented in Figure 64. The configuration in Figure 64a does, indeed, produce 6th order peaks. However, this configuration of atoms includes a Bi chain of indefinite length. This seems to be inconsistent with steric arguments.

The ordering in the [001] direction on GaAs (110) and InSb (110) is also quite interesting. Figure 65a illustrates the effect of misaligning the vacancies in the [001] direction on the intensity of the 6th order peaks in the [1-10] direction. Figures 65b-65d show the production of fractional order peaks when vacant rows are introduced in the [001] direction on InSb (110). We will examine ordering in both the [1-10] and [001] directions in the experimental LEED profiles for Bi on both GaAs and InSb (110).

These results indicate that the simple kinematic LEED model can be quite valuable in our investigations. We can produce theoretical LEED patterns for any overlayer geometry. We can also produce spectra to any predetermined level of statistical accuracy.

Figure 65. Theoretical LEED profiles for [001] direction.

Analysis of Experimental LEED Profiles for Bi/GaAs (110)

Previous sections have outlined in some detail the time-dependent Monte Carlo model that has been developed to examine the gallium arsenide (110) surface with a variety of overlayers. We have recently begun to examine the behavior of bismuth overlayers on the GaAs (110) surface in an effort to understand the dominant microscopic processes for overlayer growth and ordering. The first step in the analysis involves measuring the full width at half maximum (FWHM) of appropriate LEED (low energy electron diffraction) profiles at a variety of temperatures. This analysis, which provides a measure of overlayer ordering, will be discussed in this section.

The FWHM analysis of LEED profiles also provides a critical key to examining the microscopic details of the migration process. The experimental LEED profiles described here can be compared to the LEED profiles that are derived from the particle maps produced in the Monte Carlo simulation. Therefore, we will determine the degree of ordering in the overlayer by examining the LEED data. A method to produce theoretical LEED profiles was discussed in the previous section.

Bismuth is known to adsorb on the GaAs (110) surface and form zig-zag chains in the $[1-10]$ direction with random nucleation.⁸¹ The clean GaAs (110) surface produces the (1×1) LEED pattern illustrated in Figure 66. The direct and reciprocal lattice vectors for this system are illustrated in Figure 67. There are two atoms per unit cell. Although, at first glance, this surface seems to lack symmetry, the two atom basis in Figure 58 has symmetry consistent with the 1×1 LEED pattern for the clean surface.

Figure 66. (1×1) LEED pattern for clean GaAs (110).

Figure 67. Direct and reciprocal lattice vectors for GaAs (110).

Upon exposure to bismuth, the (6X1) LEED pattern in Figure 68 emerges.⁸² Scanning tunneling microscope experiments reveal that bismuth forms zig-zag chains with a repeat distance of 24 angstroms on the GaAs (110) surface in the [1-10] direction.⁸¹ Antimony also forms zig-zag chains on GaAs (110). There is, however, no constraint on the length of the Sb chains. The chain length for bismuth is limited due to the 9% increase in atomic radius relative to antimony. The 6X1 LEED pattern is believed to evolve from a repeating chain structure with a unit cell vector that is six times the length of the conventional unit cell.

Figure 68. (6X1) LEED pattern for Bi/GaAs (110).

The actual Bi chain length is the source of some controversy. Some have speculated that the chains are composed of 11 or 12 bismuth atoms.⁸³ We find that 11 atom chains separated by a single vacancy produce 6th order peaks in our theoretical LEED calculation. Twelve atom chains do not, however, produce an appropriate 6th order LEED spectrum. Theoretical LEED patterns are generated from maps detailing the positions of metal overlayer atoms using a simple kinematic model. The unit cell depicted in Figure 61 is certainly consistent with a 24 angstrom repeat length.

This configuration also produces a LEED pattern with 6th order spots. Presumably, the Bi chains are limited in length due to steric interactions. The series of 11 atom chains which produce 6th order peaks is comprised of rows of adjacent Bi atoms of undetermined length. This seems to defy the hypothesis that steric hindrance limits the chain length. Consequently, we base our analysis on the model in Figure 61. In order to determine whether the 11 or 10 atom model is correct, a more precise measurement of the surface coverage near one monolayer would be necessary.

Analysis of the 6th order LEED profiles at different temperatures provides information about the degree of ordering within and between chains. The 6th order spot we are analyzing appears to the immediate left of the (1,1) integral order spot. The reciprocal lattice coordinates for the (1,1) spot are (1.57,1.11) inverse angstroms. Our analysis involves LEED results at three coverages, 0.7, 1.0, and 1.5 monolayers. The experimental deposition procedure and subsequent analytical measurements produce coverages correct to within 10%.

Due to the nature of the experimental apparatus, the LEED profile is tilted at an angle of roughly 14 degrees from horizontal. Consequently, the first step in the LEED profile analysis is to rotate the experimentally derived matrix of intensities through an angle of 14 degrees. The length of the experimental LEED profile is measured in pixel

units. Each pixel corresponds to a particular position on the graphic display screen. The LEED spectra from these experiments are 422 pixels in the [001] direction and 619 pixels in the [1-10] direction.

PROGRAM SHIFT rotates the raw matrix through an angle of 14 degrees, bringing the data to a more convenient coordinate system. One potential problem in this phase of the procedure is that we compute the new X and Y coordinates in real numbers and then round to nearest integer values that correspond to matrix elements in the new array. Two rows of raw data may have the same X or Y coordinate when the results are rounded. In that case, the first row of data would be written over by the second row. Consequently, it is possible that one row of data may be lost in the rotation. There have, however, been no significant difficulties introduced by this phenomenon in the analysis.

Once the data have been rotated, the immediate region around the peak of interest is examined to determine the maximum intensity. When the coordinates of the point of maximum intensity are known, a "slice" is taken through the LEED spectrum in the direction under investigation. In the results reported here, we examine data for slices in both the [1-10] and [001] directions. In PROGRAM PROFILE, a window 100 pixels wide is chosen in the direction of interest. The center of the spot under investigation is in the center of this window. At each of the 100 window

positions, a profile is generated by taking the average over a 9 pixel band in the perpendicular direction. The center of this contains the point of maximum intensity. This averaging technique, which was required to reduce the amount of noise in the data, may affect the results. More specifically, averaging over several pixels diminishes the overall intensity of the peak, and therefore, certainly alters the value of the FWHM.

The background level in the averaged LEED profile is determined by picking one point on each side of the point of maximum intensity. The critical points are selected in regions that are relatively far away from the actual peak to insure that they represent only the background level. Once two points have been selected, the equation of a line containing the two points is determined. An estimate of the background level is then known at each point along the line. This background level can be subtracted to reveal the "actual" LEED profile.

We initially analyzed the Bi/GaAs data by fitting Gaussian functions to the LEED profiles. The LEED profiles under examination have a Gaussian shape at the top. However, they also have very wide and slowly decaying "wings" at lower intensities which are characteristic of Lorentzian profiles. We were unable to obtain Gaussian fits that included the "wings" of these curves. At low temperature, the Lorentzian character of the profiles is even more

pronounced. The Gaussian fits proved to not even be good estimates in these cases. We have finally settled on the use of an algorithm that fits the LEED peak profiles to Voigt functions (a convolution of Gaussian and Lorentzian functions). The Voigt algorithm has been coupled to a commercial algorithm which minimizes the square of the difference between the theoretical and experimental curves. Thus, we derive the best fit for each profile. This minimization algorithm is suitable for profiles in the [001] direction.

However, we decided that our analysis should include both the 6th order and integral order spots in the [1-10] direction. These peaks are in very close proximity to each other and overlap to varying degrees as a function of temperature. Consequently, we derived an algorithm that would simultaneously fit parameters to both peaks. In this fashion, we produced a curve that represented the best possible fit for each peak as well as the overlap region. Excellent fits were obtained in the manner.

The FWHM of each profile was obtained by drawing a line between the two intensity points that spanned the half maximum intensity. We derived the equation of the line that included the two intensity values and then used it to calculate the width at the half maximum. Figure 69 presents a representative sample of the curve fits for the integral order (1,1) and (5/6,1) spots in the [1-10] direction and

for the $(5/6,1)$ spot in the $[001]$ direction. The FWHM can be converted to inverse angstroms if the appropriate conversion is known. In this case, we used the raw data to compute the distance between two integral order peaks. The conversion becomes $118 \text{ pixels} = 1.57 \text{ inverse angstroms}$. This scaling factor places peaks at the appropriate theoretical positions.

Figures 70a through 72a show plots of the FWHM in the $[1-10]$ direction as a function of temperature for the three coverages under investigation. At 0.7 ML and low temperatures, the 6th order spot appears as a diffuse shoulder adjacent to the integral order spot. As the annealing temperature is increased, the peak narrows and becomes a distinct entity. This trend indicates that, as the thermal energy increases, the adatoms align along the $[1-10]$ direction. The (6×1) superstructure is formed in this fashion.

The FWHM values for the 6th order spot in the $[001]$ direction in Figure 70b indicate that there is also ordering between the chains. The ordering in the $[001]$ direction involves the alignment of the vacancies between the 10 atom bismuth chains. There is little change in the FWHM for the integral order peak with temperature.

Figure 69. Representative Voigt functional fits for experimental LEED profiles for Bi/GaAs (110).

Figure 70. FWHM vs. temperature for a) (5/6,1) and (1,1) LEED spots [1-10] direction and b) (5/6,1) spot [001] direction for 0.7 ML Bi on GaAs (110).

Figure 71. FWHM vs. temperature for a) (5/6,1) and (1,1) LEED spots [1-10] direction and b) (5/6,1) spot [001] direction for 1.0 ML Bi on GaAs (110).

Figure 72. FWHM vs. temperature for a) (5/6,1) and (1,1) LEED spots [1-10] direction and b) (5/6,1) spot [001] direction for 1.5 ML Bi on GaAs (110).

At 1.0 ML there appear to be three distinct regimes in the [1-10] direction. There is relatively little change in the FWHM of the 6th order spot up to a temperature of 10 degrees Celsius. From 10 to 110 degrees, there is a steady decline in the FWHM which corresponds to overlayer ordering. It is not clear whether this rapid decline corresponds to filling from second layer atoms or vacancy hopping. There is a less dramatic change in the FWHM from 110 to 200 degrees. In the [001] direction, the FWHM is significantly smaller than the FWHM at 0.7 ML. Once again, there is no change in the width of the integral order spot at this coverage.

There is a visible trend toward ordering in both directions at 1.5 ML. The changes are much smaller in this case. It should be noted that the minimum widths of the integral and 6th order spot tends toward the same values at 1.0 and 1.5 ML. We expect that, at saturation coverage, there is an upper bound on the amount of ordering due to the finite number of vacancies on the lattice. The trends in the FWHM for the 6th order peak are quite similar. It appears as if a substantial amount of adsorbate is deposited in a second layer at these higher coverages. As we heat the surface, we anneal both of these layers. The final values for the FWHM at 1.0 and 1.5 ML indicate that we arrive in both cases at a highly ordered or defect deficient overlayer structure.

We observe a narrowing of the fractional order peak in the [001] direction as a function of temperature at all three coverages. Ordering in the [001] direction involves alignment of the vacancies between bismuth chains.

The FWHM for the LEED spots at three coverages indicate that the overlayers order with increasing temperature. A closer examination of the LEED profiles reveals that the position of maximum intensity of the 6th, or other fractional order, peak shifts as a function of temperature also. Our theoretical LEED studies clearly reveal that the shift in the fractional order peaks is directly related to the average chain length. We hoped that the shift in the experimental data would allow us to predict the average chain length at each temperature. These shifts, which are presented in Table III, obviously don't provide the desired information. The fractional order peak shifts in both direction with annealing.

We find that, under optimal conditions, the maximum experimental resolution obtainable in the experiment is on the order of a few hundredths of an angstrom. Consequently, we conclude that the observed shifts in the data are not statistically significant.

In our theoretical LEED studies of this system, we see peak shifts ranging from roughly 1.30 to 1.40 inverse angstroms. Because these contrived overlayer configurations include systems at lower coverages, we don't expect that we

will see such a wide range of shifts in the experimental data. Configurations of chains separated by several vacancies are most likely impossible under these experimental conditions.

Table III. Shifts in Maximum Intensity of Fractional Order LEED Peaks for Bi/GaAs (110) in the [1-10] direction.

Coverage (ML)	0.7	1.0	1.5
Temperature (C)			
-110	1.23	1.35	1.35
-90	1.29	1.42	1.34
-70	-	1.42	-
-50	-	1.42	1.35
-30	1.29	1.42	-
-10	-	1.41	-
10	1.28	1.40	1.38
30	-	1.40	1.38
50	-	1.39	-
70	1.28	1.38	-
90	1.28	1.38	1.36
110	1.28	1.37	1.35
200	1.30	1.37	1.33

Time-Dependent Monte Carlo Simulations of Bi/GaAs (110)

We have examined the kinetics of overlayer ordering for bismuth atoms adsorbed on the GaAs (110) surface. The algorithm restricts the chain length to 10 atoms by introducing an infinite barrier for the formation of longer chains. A series of attractive interactions along and

between chains produces the particle map in Figure 73. We see here both the formation of zig-zag chains and the alignment of the vacancies along the [001] direction. We have produced LEED spectra for several of these systems at various coverages. We can resolve the theoretical LEED spectra to any predetermined level of accuracy. However, we have determined that the results of the LEED experiments which are currently available to us are not sufficient to describe the actual overlayer order on the surface.

Conclusions for Bi/GaAs (110)

We have developed a method to analyze experimental LEED profiles for Bi/GaAs (110). The profiles produced by this method provide some insight into the ordering of the overlayer. However, we find that the experimental resolution is insufficient to allow us to make accurate predictions about the average chain length at any given temperature. The TDMC algorithm produces highly ordered overlayers of zig-zag chains. The simulation also displays the predicted order or alignment of vacancies in the [001] direction. With more precise experimental data, we will certainly be able to make more specific statements about the meaning of the changes in the FWHM of the LEED profiles. We will also be able to discuss shifts in the fractional order LEED peaks as they relate to changes in the average chain length with annealing.

Figure 73. Time-dependent Monte Carlo generated surface map for Bi/GaAs (110). All adparticle interactions are attractive.

Analysis of Experimental LEED Profiles for Bi/InSb (110)

We have recently begun to examine the behavior of bismuth overlayers on InSb (110). This system is quite similar to the Bi/GaAs (110) system.⁸⁴ The (1X1) LEED pattern for clean InSb (110) is illustrated in the three dimensional plot in Figure 74. Upon deposition of 1.5 ML of bismuth at room temperature, the (1X1) pattern disappears, indicating that the bismuth overlayer is disordered. This behavior is observed for coverages ranging from 0.2 to 1.5 ML. Consequently, the bismuth overlayer is believed to be disordered at all coverages at room temperature. We believe that, upon annealing, bismuth forms the same sort of zig-zag chains in the [1-10] direction as those observed for bismuth on other III-V semiconductors.

Upon annealing to 100 C, a broad streak begins to appear in the [001], or interchain, direction as illustrated in Figure 74. Further annealing to 180 C produces the (1X2) pattern in Figure 74. With continued heating, the (1X2) pattern becomes diffuse and eventually gives rise to a sharp (1X3) pattern at 270 C. This is also illustrated in Figure 74. Finally, at 320 C the bismuth is completely desorbed; and the (1X1) pattern of the clean InSb surface returns.

Bismuth begins to desorb almost immediately as the sample is heated.

Figure 74. Experimental LEED profiles for Bi/InSb (110) as a function of temperature.

The Auger data displayed in Figure 75 provide an estimate of the bismuth coverage as a function of the temperatures examined. The (1X2) phase appears at approximately 0.5 ML. The (1X3) phase corresponds to a coverage of about 0.3 ML.

Figure 75. Auger data for an initial coverage of 0.8 ML Bi on InSb (110).

A proposed explanation for these data is that the LEED pattern in the [001] direction evolves as illustrated in Figure 76. Upon annealing to 180 C, it is believed that bismuth chains become spaced at a distance of two unit cells apart in the [001] direction. At higher temperatures where the (1X3) pattern is observed, it is believed that the bismuth chains are oriented at a distance of three unit cells apart. This model, which suggests a strong repulsive interchain interaction, is certainly consistent with the coverage estimates.

Figure 76. Possible structural interpretation for Bi on InSb (110) LEED data.

Our theoretical analysis consists of examining the FWHM of the appropriate LEED spots in the $[1-10]$ and $[001]$ directions. We have used a procedure which is entirely analogous to the one described in the section on Bi/GaAs. In this case, the background level of the LEED spectra was such that it completely sabotaged the analysis. In order to eliminate the effect of the highly asymmetric background, the LEED spectra were "folded" in half about the $(1,0)$ peak in the $[001]$ direction. It should be noted that this

analysis is not purely legitimate due to the fact that there is no rotational symmetry in the [001] direction of InSb. The fractional order peaks appear at the same locations on both sides of the integral order peak. They have different intensities, however. The background, which was decidedly non-linear, was successfully eliminated using this procedure.

There is one perplexing feature about the data generated by folding the LEED spectrum. It appears as if a (1X3) pattern emerges at low temperatures before the formation of the (1X2) pattern. This behavior would be contrary to the mechanism in Figure 76 unless much of the bismuth is trapped for a finite period of time in a second layer upon deposition.

We have again incorporated the algorithm that fits Voigt functions to the experimental data. The Voigt algorithm has been coupled to a commercial algorithm which minimizes the square of the difference between the theoretical and experimental curves. Once again, excellent fits were obtained. The FWHM was obtained by drawing a line between the two points that spanned the half maximum intensity. The equation of a line through these two points was then used to calculate the width at that point.

Figures 77 and 78 illustrate the changes in the FWHM as a function of temperature for the [1-10] and [001] directions of the integral and half order peaks,

respectively. It is our opinion that there were not enough data points of sufficient quality to make a thorough analysis for the 3rd order spots. The data that we do have are presented in Figure 79. It is interesting to note that Figure 77 indicates that there is little change in the length of the chains in the [1-10] direction as the temperature is increased. Although there are some subtle changes as the temperature increases, these differences are not believed to be of statistical significance.

Figure 77. FWHM for (1,0) beam vs. temperature for Bi on InSb (110).

Due to asymmetry of unknown origin, the LEED spectra could not be folded in half in the $[1-10]$ direction. Because there is rotational symmetry in this direction, such an analysis would be both appropriate and beneficial for reducing the amount of noise in the data. However, the $(1,0)$ beam appeared off the horizontal axis in the $[1-10]$ direction by a significant amount. Folding would have substantially altered the shape of the peak we were analyzing. This "off axis" behavior was initially believed to be due to over-rotation of the initial LEED spectrum; but the other peaks along the same axis in the $[001]$ direction had appropriate coordinates. We anticipate that averaging of the symmetry related spots in the $[1-10]$ direction would reduce the amount of scatter in the data presented in Figure 77. This analysis would probably show that there are no significant changes in the FWHM in the $[1-10]$ direction.

The results in Figure 77 suggest that any ordering on the surface is in the interchain, or $[001]$, direction. The change in the FWHM of the $(1,0)$ beam in the $[001]$ direction certainly supports this conclusion. In fact, there appears to be a pronounced trend toward ordering in the $[001]$ direction as the surface is heated from room temperature. There is little change in the FWHM after the 1×2 pattern emerges. As the 1×3 pattern develops, there is a significant trend toward disordering again. Figure 78, the analysis of the half order beam, also indicates that there is little

change in the chain length, but that there is slight ordering between chains as the temperature is increased.

A possible mechanism for these observed changes is shown in Figure 80. It is certainly consistent with a nearly constant chain length. In this proposed mechanism, ordering on the surface would be due to the alignment of the chains in the [001] direction. Disordering, corresponding to an increase in the FWHM, would be brought about by moving the chains out of registry. This mechanism is consistent with the fact that no change occurs in the [1-10] direction.

Figure 78. FWHM for (1, -1/2) beam vs. temperature for Bi on InSb (110).

Figure 79. FWHM for (1, -1/3) beam vs. temperature for Bi on InSb (110).

Figure 80. Possible chain ordering/disordering mechanism for Bi on InSb (110).

Time-Dependent Monte Carlo Simulations of Bi/InSb (110)

The proposed structural model indicates that there is a strong repulsive interchain interaction. This feature alone makes the Monte Carlo simulation of this system very exciting. Monte Carlo results at 0.5 ML for this system produce the particle map in Figure 81 that resembles the one depicted in Figure 77. This surface structure evolved from a random initial distribution of particles and a significant interchain repulsive interaction. Theoretical LEED profiles for contrived configurations with missing rows in the [001] direction were shown previously to produce fractional order peaks at the appropriate wave vectors. Theoretical LEED also produces a half order peak for the map produced in this Monte Carlo simulation. The LEED profile is presented in Figure 82.

Unfortunately, we are unable to extend our model to examine two missing rows of atoms along the [001] direction. This would require us to include next-nearest neighbor interactions in the basic hexagonal configuration. There are in excess of two million theoretical transition types in that case. We are currently unable to treat such long range interactions.

Figure 81. Surface map from time-dependent Monte Carlo simulation of Bi diffusion on InSb (110).

Figure 82. Theoretical LEED profile for TDMC map of Bi on InSb (110).

Conclusions for Bi/InSb (110)

We find that overlayer ordering in this system is primarily in the [001] direction. The missing row model is supported by theoretical LEED calculations for contrived surfaces. The result of a time-dependent Monte Carlo study also supports this mechanism for a missing row model produced by strong interchain repulsions.

CONCLUSIONS

The beauty of our time-dependent Monte Carlo approach lies not only in its relative simplicity, but also in the ability of the method to avoid the difficulties encountered in other theoretical approaches. The novelty of this approach begins with the kinetic treatment of the surface diffusion problem. Many investigators, as stated in the introduction, attempt to treat diffusion from a thermodynamic perspective when, in fact, it is the microscopic kinetic processes which actually govern the macroscopic behavior of a diffusion system.

Furthermore, the time-dependent Monte Carlo method treats successfully the "rare event" problem discussed above by incorporating the probabilistic formalism for describing adparticle jump events. Because a particle moves in each TDMC cycle, the algorithm is highly efficient. Thus, this approach implements a unique solution to the historical dilemma associated with analyzing dynamical processes which have widely varying time scales. There are at least several orders of magnitude difference between the ordering time scale and diffusion time scales. Therefore, a successful algorithm must be able to describe the long term process in relatively little "real" time. Our TDMC approach certainly accomplishes this objective.

For instance, a defect on the lattice may trap an adatom indefinitely. Other theoretical approaches would spend an unwarranted amount of computation time "watching" the inactive atom. In our TDMC method, a single incident of adatom trapping would have little effect on the overall behavior of the system; and little, if any, time would be spent addressing that particular atom. The most probable events are addressed with the highest frequency. Consequently, the predominant physics of the system is always portrayed via the maximum computational efficiency.

We have successfully examined diffusion on perfect surfaces and described the effects of coverage, interactions, and temperature on diffusion rates and mechanisms. Unlike other techniques, our TDMC method can monitor the diffusion process from the first adatom hop until long range overlayer ordering is observed in the particle maps.

The time-dependent Monte Carlo method developed here easily accommodates a wide variety of substrate geometries and overlayers. We find that widely varying diffusion mechanisms are operative as we change adparticle interactions, concentrations, surface defects, and substrate geometries. We have also derived from macroscopic experimental measurements the microscopic details of the diffusion process.

We further conclude that diffusion experiments, as they are currently performed and interpreted, are inappropriate in

many respects and fail to determine anything beyond the combined effects of adparticle interactions, concentration, and surface defects. Surface defects contribute in a very significant manner to diffusion rates, mechanisms, and overlayer ordering processes. Any analysis which fails to consider the effects of surface defects is fundamentally flawed.

Our analyses of metals on III-V semiconductor surfaces have proved to be a valuable beginning in our efforts to understand the molecular beam epitaxial growth process. Time-dependent Monte Carlo simulations coupled with a simple, theoretical low energy electron diffraction algorithm produce valuable insights which are entirely consistent with the macroscopic experimental observations. Consequently, we can begin to examine the other microscopic processes that are involved in the epitaxial growth processes.

The diffusion simulations described here have made a significant contribution to our understanding of microscopic surface diffusion phenomena. However, we have hardly applied the TDMC algorithms which have been developed here to their maximum potential. A multitude of questions can certainly be answered in future experiments. For instance, investigations of "mountainous" surfaces and of the effects of surface traps on the desorption process would surely be interesting. A number of interesting avenues for investigation lie in store; and certainly some intriguing discoveries lie ahead.

LITERATURE CITED

LITERATURE CITED

1. Ehrlich, G. and Stolt, K., Ann. Rev. Phys. Chem. 31 (1980) 603.
2. King, D.A., CRC Crit. Rev. Solid State Mater. Sci. 7 (1978) 167.
3. King, D.A., J. Vac. Sci. Technol. 17 (1980) 241.
4. Doll, J.D. and Voter, A.F., Ann. Rev. Phys. Chem. 38 (1987) 413.
5. Ehrlich, G., CRC Crit. Rev. Solid State Mater. Sci. 10 (1982) 391.
6. Tsong, T.T. and Cowan, P.L., CRC Crit. Rev. Solid State Mater. Sci. 7 (1978) 289.
7. Rhead, G.E., Surf. Sci. 47 (1975) 207.
8. Ehrlich, G., CRC Crit. Rev. Solid State Mater. Sci. 4 (1974) 205.
9. Gomer, R., Field Emission and Field Ionization, Harvard University Press, 1961.
10. Ehrlich, G. and Hudda, F.G., J. Chem. Phys. 35 (1961) 1421.
11. Gomer, R., CRC Crit. Rev. Solid State Mater. Sci. 8 (1979) 119.
12. Ehrlich, G., Phys. Today 34 (1981) 44.
13. Ehrlich, G., Surf. Sci. 63 (1977) 422.
14. Muller, E.W. and Tsong, T.T., Field Ion Microscopy, Principles and Applications, American Elsevier Publishing Co., Inc., 1969.
15. DiFoggio, R. and Gomer, R., Phys. Rev. B 25 (1982) 3490.
16. Tsong, T.T., Appl. Surf. Sci. 33/34 (1988) 15.
17. George, S.M. DeSantolo, A.M., and Hall, R.B., Surf. Sci. 159 (1985) L425.

18. Mullins, D.R., Roop, B., and White, J.M., Chem. Phys. Lett. 129 (1986) 511.
19. Roop, B., Costello, S.A., Mullins, D.R., and White, J.M., J. Chem. Phys. 86 (1987) 3003.
20. Seebauer, E.G. and Schmidt, L.D., Chem Phys Lett. 123 (1986) 129.
21. Seebauer, E.G., Kong, A.C.F., and Schmidt, L.D., Surf. Sci. 176 (1986) 134.
22. Burgess, D. ,Jr., Viswanathan, R., Hussla, I., Stair, P.C., and Weitz, E., J. Chem. Phys. 79 (1983) 5200.
23. Burgess, D.R. ,Jr., Hussla, I., Stair, P.C., Viswanathan, R., and Weitz, E., Rev. Sci. Instrum. 55 (1984) 1771.
24. Hall, R.B., J. Phys. Chem. 91 (1987) 1007.
25. Stair, P.C. and Weitz, E., J. Opt. Soc. Am. B 4 (1987) 255.
26. Brand, J.L and George, S.M., Surf. Sci. 167 (1986) 341.
27. Burgess, D., Jr., Stair, P.C., and Weitz, E., J. Vac. Sci. Technol. A 4 (1986) 1362.
- 28. Deckert, A.A., Brand, J.L., Arena, M.V., and George, S.M., J. Vac. Sci. Technol. A 6 (1988) 794.
- 29. Brand, J.L., Mak, C.H., Deckert, A.A., Koehler, B.G., and George, S.M., J. Vac. Sci. Technol. A 6 (1988) 842.
30. Seebauer, E.G., Kong, A.C.F., and Schmidt, L.D., J. Chem. Phys. 88 (1988) 6597.
31. Mak, C.H., Deckert, A.A., and George, S.M., J. Chem. Phys. 89 (1988) 5242.
32. Mak, C.H., Koehler, B.G., Brand, J.L., and George, S.M., J. Chem. Phys. 87 (1987) 2340.
33. LaGally, M.G., Wang, G.C. and Lu, T.-M., CRC Crit. Rev. Solid State Mater. Sci. 7 (1978) 233.
- ⊕ 34. Van Hove, M.A., "Low Energy Electron Diffraction: Experiment, Theory, and Structure Determination," Springer-Verlag, 1986.

35. Binning, G. and Rohrer, H., Rev. Mod. Phys. 59 (1987) 615.
36. Feenstra, R.M., and Martensson, P., Phys. Rev. Lett. 61 (1988) 447.
37. Feenstra, R.M., Thompson, W.A., and Fein, A.P., Phys. Rev. Lett. 56 (1986) 608.
38. Feenstra, R.M., Thompson, W.A., and Fein, A.P., J. Vac. Sci. Technol. A 4 (1986) 1315.
39. Binning, G. and Rohrer, H., Surf. Sci. 152/153 (1985) 17.
40. Hamers, R.J., Avouris, Ph., and Bozso, F., Phys. Rev. Lett. 59 (1987) 2071.
41. Stroscio, J.A., Feenstra, R.M., and Fein, A.P., Phys. Rev. Lett. 58 (1987) 1668.
42. Stroscio, J.A., Feenstra, R.M., and Fein, A.P., J. Vac. Sci. Technol. B 5 (1987) 923.
43. Becker, R.S., Swartzentruber, B.S., Vickers, J.S., Hybertsen, M.S., and Louie, S.G., Phys. Rev. Lett. 60 (1988) 116.
44. Tully, J.C., Ann. Rev. Phys. Chem. 31 (1980) 319.
45. Doll, J.D. and McDowell, H.K., Surf. Sci. 123 (1982) 99.
46. Doll, J.D. and McDowell, H.K., J. Chem. Phys. 78 (1983) 3219.
47. McDowell, H.K. and Doll, J.D., Surf. Sci. 121 (1982) L537.
48. Mruzik, M.R. and Pound, G.M., J. Phys. F11 (1981) 1403.
49. DeLorenzi, G., Jacucci, G., and Pontikis, V., Surf. Sci. 116 (1982) 391.
50. Voter, A.F. and Doll, J.D., J. Chem. Phys. 82 (1985) 80.
51. Tully, J.C., Gilmer, G.H., and Shugard, M., J. Chem. Phys. 71 (1979) 1630.
52. Garofalini, S.H. and Halicioglu, T.H., Surf. Sci. 104 (1981) 199.

53. DeLorenzi, G. and Jacucci, G., Surf. Sci. 116 (1982) 391.
54. Park, S.C. and Bowman, J.M., J. Chem. Phys. 80 (1984) 2191.
55. Lim, C. and Tully, J.C., J. Chem. Phys. 85 (1986) 7423.
56. Voter, A.F., Doll, J.D., and Cohen, J.M., J. Chem. Phys. 90 (1989) 2045.
57. Cohen, J.M. and Voter, A.F. Unpublished.
58. Voter, A.F. and Doll, J.D., J. Chem. Phys. 80 (1984) 5814.
59. Voter, A.F. and Doll, J.D., J. Chem. Phys. 80 (1984) 5832.
60. Ehrlich, G. and Kirk, C.F., J. Chem. Phys. 48 (1968) 1465.
61. Bassett, D.W. and Webber, P.R., Surf. Sci. 70 (1978) 520.
62. Wynblatt, P.W. and Gjostein, N.A., Surf. Sci. 22 (1970) 125.
63. Banavar, J.R., Cohen, M.H., and Gomer, R., Surf. Sci. 107 (1981) 113.
64. Doll, J.D. and McDowell, H.K., J. Chem. Phys. 77 (1982) 479.
65. Lauderdale, J.G. and Truhler, D.G., Surf. Sci. 164 (1985) 558.
66. Bowker, M. and King, D.A., Surf. Sci. 71 (1978) 583.
67. Bowker, M. and King, D.A., Surf. Sci. 72 (1978) 208.
68. Sadiq, A. and Binder, K., Surf. Sci. 128 (1983) 350.
69. Reed, D.A. and Ehrlich, G., Surf. Sci. 120 (1982) 179.
70. Oguz, E., Valls, O.T., Mazenko, G.F., Luscombe, L., and Heilig, S.J., Surf. Sci. 118 (1982) 572.
71. Tsong, T.T., Surf. Sci. 122 (1982) 99.
72. Reed, D.A. and Ehrlich, G., Surf. Sci. 102 (1981) 588.

73. Ray, L.A. and Baetzold, R.C., unpublished.
74. Mak, C.H., Andersen, H.C., and George, S.M., J. Chem. Phys. 88 (1988) 4052.
75. Hall, R.B., Upton, T.H., and Herbolzheimer, E., J. Vac. Sci. Technol. B 5 (1987) 1470.
76. Tsong, T.T. and Gao, Q., Surf. Sci. 182 (1987) L257.
77. Landman, U., Barnett, R.N., Cleveland, C.L., and Rast, R.H., J. Vac. Sci. Technol. A 3 (1985) 1574.
78. Drube, W., and Himpsel, F.J., "Unoccupied surface states of (1x1) Sb overlayers on GaAs(110) and InP(110)", Phys. Rev. B., 37(2) (1988) 855.
79. Schaffler, F., Ludeke, A., Taleb-Ibrahimi, A., Hughes, G., and Rieger, D., J. Vac. Sci. Technol., B5(4) (1987) 1048.
80. Bertoni, C.M., Calandra, F., Manghi, F., and Molinari, E., Phys. Rev. B, 27(2) (1983) 1251.
81. Skeath, P., Lindau, I., Su, C.Y., and Spicer, W.E., J. Vac. Sci. Technol., 19(3) (1981) 556.
82. McLean, A.B., Feenstra, A., Taleb-Ibrahimi, A., and Ludeke, R., Phys. Rev. B, 39 (1989) 12925.
83. Chang, S.L., Guo, T., Ford, W.K., Bowler, A., and Hood, E.S., in press.
84. T. Guo, K.J. Wan, and W.K. Ford, in press.

APPENDICES

APPENDIX A

TIME-DEPENDENT MONTE CARLO ALGORITHM FOR PERFECT SURFACES

Time-Dependent Monte Carlo Algorithm for Perfect Surfaces

The following paragraphs provide a comprehensive description of the VAX FORTRAN subroutines that simulate surface diffusion according to the time-dependent Monte Carlo formalism discussed in the previous section. Each subroutine has been designed to perform one, or more where appropriate, of the tasks described in the preceding section. These routines are called from the main program, DIFFUSE, which also contains all of the formats for reading data and writing the desired output arrays. The parameters that must be passed to each routine from the calling program are discussed below wherever appropriate.

The code was developed in its original form to accommodate 20 particles on a 10 X 10 lattice. This initial size limitation allowed us to subject the output from each subroutine to a manual check to insure that the data transfer schemes and output were valid at each phase of program development. The program is currently dimensioned to accommodate up to 1,000 particles on a 200 X 200 lattice. Further details about the size constraints on the simulation are discussed in relation to the pertinent program arrays as they are described in the following paragraphs.

The following is a list of critical criteria for a valid Time-Dependent Monte Carlo FORTRAN algorithm:

1. The following parameters must be constantly updated:
 - A. Current particle locations;
 - B. Occupation numbers for each transition type;
 - C. Probability associated with each transition type;
 - D. Occupation "map" of the surface.
2. The algorithm must do the following to satisfy the TDMC formalism:
 - A. Address transition types according to their associated probabilities;
 - B. Move particles randomly within transition types.

The simulation begins in SUBROUTINE RANMAP with the random generation of particle coordinates in a square or rectangular configuration. This routine incorporates the external function RANDOM to supply random coordinates. The number of particles, initial lattice dimension in each direction, and seed for the random number generator must be provided by the calling program. The seed should be a large integer of at least ten digits. Random numbers are generated in an $M \times N$ rectangle, where M and N correspond, respectively, to the lattice dimensions in the X and Y directions. More specifically, to place particles in a rectangle with the desired dimensions, a real random number between 0 and 1 is generated by RANDOM and multiplied by the lattice dimension in the appropriate direction. The resulting real number is then rounded to the nearest integer value which corresponds to a lattice site (array element). Two random numbers must be generated to obtain each set of

particle coordinates.

Each particle, which is assigned a number or "identity," can be monitored throughout the calculation. If the site selected by RANDOM is already occupied, a new set of random numbers is generated. This process continues until the desired coverage of particles is deposited. The initial map is saved; and another copy, the current map, is placed in a common storage block where it can be constantly updated during the migration process.

Thus far, we have begun most of the diffusion simulations from a randomly populated band, strip, or 'pack' of particles positioned at one edge of the lattice. Therefore, the coverage, which is defined by the initial dimensions of the lattice, begins to change as soon as the simulation proceeds. The crowded initial configuration is desirable and has been deemed appropriate because it most closely resembles the sharp initial concentration gradient required for the diffusion experiments discussed previously.

We have also found it necessary to examine the case where the narrow particle band is placed in the center of the lattice. A subroutine that generates any desired random initial configuration is easily derived from RANMAP in its present form and incorporated into the simulation. The simulation also possesses the flexibility to incorporate a predetermined initial configuration of particles. For instance, we have monitored migration from a perfect next-

nearest neighbor island configuration. The only criterion for the simulation is that a set, any set, of coordinates for the initial configuration be passed to the main program. There are no constraints on the initial positions of the particles.

In SUBROUTINE TRANSTYPE, each allowed transition type is assigned a number ranging from 1 to 196. Those types which are forbidden are assigned a value of zero. Each of the allowed transition type numbers, which were arbitrarily assigned, is referenced by indices which correspond to a specific particle neighborhood. When the program runs, the transition type numbers are read into a common storage block and retrieved using indices corresponding to the occupation number of the nearest and next-nearest neighbor sites. In order to insure that no allowed transition types were excluded from the list, an error statement is included in the code. The simulation is stopped and the error statement is printed in the event that an allowed transition type had been excluded from the list. The code has successfully run hundreds of times. Thus, it is certain that these 196 transition types describe the complete spectrum of adparticle hopping events on the four-fold symmetric lattice.

The energy barrier for migration, the nearest and next-nearest neighbor interaction energies, and the temperature are passed to SUBROUTINE ENERGY; and the

potential energy well depth for each possible neighborhood is calculated using Equation 3. According to this equation, negative values for the interaction energies correspond to repulsive interactions. Once the well depths are computed, the activation energy and Boltzmann factor associated with each transition type can be calculated from the appropriate equations. The Boltzmann factors are passed back to the calling program and repeatedly used to update the hopping probabilities.

SUBROUTINE TRANSITION counts the nearest and next-nearest neighbors for each particle given a map of the entire lattice. The final dimension of the lattice, which is assumed to be the same in both directions in the current version of the code, and the parameter ROUTE are passed from the calling program. After the neighbor numbers have been determined at its initial position, each particle is 'moved' in each vacant direction. The number of neighbors for each prospective final position are determined in this fashion. The counting scheme ignores the initial site. Consequently, the required vacancy will be left when a particle moves. The edge constraints that have been imposed to enforce particle conservation create an absorptive boundary. In other words, if a particle is at the edge of the lattice, motion in any direction that would place it off of the lattice is forbidden. No artifacts of this assumption have been detected in our results. In fact, the boundary condition

appears to have a completely negligible effect on diffusion dynamics.

Once the numbers of nearest and next-nearest neighbors for the initial and final positions are known, the number of the transition type which corresponds to each move is retrieved from the common block. The particle number and the final X and Y coordinates for each potential move are placed on the appropriate transition type (T) list. Each particle may appear on as many as four lists. Each time a particle is added to the end of a list, the number of the last element of the list is recorded. Each time a particle is removed from a list, a zero is recorded. The occupation number of the list is obtained by subtracting the number of zeros on a list from the number of the last occupied element of the list. Finally, the current position of the particle, its allowed transition types, and the position that it occupies on each T list are recorded in a matrix (PART) which is indexed by particle number.

TRANSITION has been developed to treat two special cases which are distinguished by the parameter ROUTE. In the first case, ROUTE 1, the entire lattice is swept; and the neighborhoods and transition types for all of the particles are determined. In the second case, ROUTE 2, only the twelve particles in the immediate vicinity of the moving particle are considered. The list of particles affected by a move is generated in SUBROUTINE MOVE and passed to TRANSITION

through a common storage block.

The probability associated with each transition type is simply the product of its Boltzmann factor and occupation number. After SUBROUTINE PROB calculates the occupation number of each transition type list, the probability associated with each type is computed. Once again, this subroutine is designed to either calculate the probabilities for all of the transition types or only for those types affected when a particle moves. The parameter ROUTE is used to distinguish between the two cases. The parameters passed from the main program include the ROUTE, the lists of transition types affected by a move, and the Boltzmann factors. The routine passes back the current probabilities for all 196 transition types.

In SUBROUTINE RANTYPE, the sum of the probabilities of all of the transition types is computed. Then each individual transition probability is divided by this normalization factor. If a transition type has a non-zero probability, then it is added into a running total. The value of the upper endpoint of this interval is stored in a matrix indexed by that transition type number. Consequently, each transition type encompasses an interval that is directly proportional to its magnitude, as illustrated in Figure 83. If the probability for a transition type is zero, then the matrix element for that type is also zero. A random number between 0 and 1 is generated. The index (transition

type) of the interval in which the number falls is then determined.

Figure 83. Transition probability assignment scheme.

Once a transition type has been randomly selected and the number of occupied elements on the list has been computed, another random number is generated and multiplied

by the occupation number. This value is rounded to the nearest integer, N ; and the particle at the N th filled position on the T list is selected to move. The transition type list is scanned until the N th filled position is located. Then the actual position on the list is recorded. Finally, the transition type and position on the list are passed to the subroutine that moves the particles. Because two random numbers corresponding to the transition list and position on the list are generated in this routine, two seeds must be passed from the calling program. Once again, these seeds are large integers.

Each time a transition type is selected, the transition rate for the move is computed by multiplying the transition probability by a prefactor, or rate constant, for migration which is assumed to be the same for all transition types. The validity of this assumption has been discussed previously. The transition time for the move is the inverse of the transition rate. The total time is given by the sum of the times for each transition divided by the total number of particles. SUBROUTINE CLOCK calculates the time for each hop and sums each transition time to obtain the total time for the process.

In SUBROUTINE MOVE, the randomly selected particle is moved to its final destination; and a list of neighboring particles that are affected by the move is generated. The calling program passes the randomly selected transition type

and position on the T list to MOVE. The final lattice length must also be provided to this routine. After the number of the particle and its current position on the lattice are referenced, a sweep of the twelve neighboring sites is made. The direction of the sweep depends on the direction of motion as illustrated in Figure 84. The numbers of the affected particles are placed in the array TMAP and passed to TRANSITION where their new neighborhoods and transition types are determined.

Figure 84. Sites affected by adparticle hops.

SUBROUTINE MOVE also keeps track of particle displacements. Each time a particle moves, its total trajectory length, or hop number, is increased by one. The 'absolute' displacement of each particle is also computed using the convention that moves which are in the positive X and Y directions have a value of positive one; and moves which are in the negative X and Y directions have a value of negative one. The total number of hops and the net displacement for each particle are thus known at the end of the simulation. Consequently, the number of hops made by all of the particles can be summed. The resulting number, which should equal the number of TDMC cycles specified in the data file, provides proof that the run is valid.

SUBROUTINE CLEAR removes the particles affected by a move from the transition type lists and replaces them with zeros. This routine also records the numbers of the T lists which are affected; so the probabilities for those types can be updated. Similarly, ADDT is called after the new transition types for the affected particles are determined. The numbers of the T lists to which particles are added are also recorded. This procedure eliminates the need to calculate the probabilities for all of the transition types after each move because only the numbers of the affected types are passed to PROB.

Time Evolution of the System

The foregoing was a description of the subroutines that comprise the time-dependent Monte Carlo portion of the simulation. A series of subroutines has also been developed to monitor the time evolution of the system and to produce data that describe the macroscopic processes of diffusion and ordering within the adsorbate overlayer. Data is written to files at various times throughout the simulation. Thus, we can monitor the time evolution of the system with any desired frequency. FORTRAN code that incorporates DISSPLA graphics software has been developed to plot the pertinent output.

In SUBROUTINE CORRELATE, each filled site on the lattice is addressed. The number of neighbors of each "type" surrounding each particle are counted according to the scheme in Figure 1. When a particle is addressed, the occupation number of neighbors one through eight are tabulated. We then move to the next filled site and add the occupation number of each neighbor type surrounding the particle of interest to the previous total.

After all of the filled lattice sites have been addressed, we have the total number of neighbor types one, two, three, et cetera. These occupation numbers are then normalized. The normalization constant for each neighbor

in the system and the number of equivalent sites for that neighbor type. The resulting normalized quantity is called the correlation, or radial distribution, function. The correlation function provides a method for describing ordering and island formation in the system. These correlation data are written to a file and plotted using PROGRAM PLOT CORR.

SUBROUTINE COUNT divides the lattice into sections that either run parallel or perpendicular to the diffusion boundary as illustrated in Figure 85. This routine is called after a predetermined number of TDMC cycles; and the number of particles in each section is recorded. Diffusion profiles and information regarding diffusion mechanisms can be generated by monitoring the time evolution of particle concentrations in specific regions of the lattice.

The initial and current particle positions and the lattice jump length are input into SUBROUTINE RMS. The distance traveled by each particle is then calculated. The mean square displacement for the system is calculated by summing the squares of the particle displacements and dividing by the total number of particles. The root mean square displacement is simply the square root of the mean square displacement. The lattice is also divided into sections that either run parallel or perpendicular to the diffusion boundary; and the RMS displacement is calculated for each section. This provides an additional method to

for each section. This provides an additional method to detect diffusion patterns as a function of particle concentration.

Figure 85. Counting scheme for directional diffusion studies.

SUBROUTINE RMST performs the same basic function as SUBROUTINE RMS. RMST calculates the root mean square displacement as a function of time throughout the simulation. This routine is called after a specified number of cycles. Consequently, the time increment is not precisely the same between each set of data points. These data, which are written to a separate file, can be used to test random walk conditions at low coverages where a plot of root mean square displacement versus square root of time should be linear. The data are plotted by PROGRAM PLOT RMS. These data are also used to calculate the diffusion coefficient as a function of time.

The particle coordinates are also saved at regular intervals throughout the calculation. Thus, maps of particle locations can be generated at each phase of the calculation. These data, which are saved in the PART array, are plotted in PROGRAM PLOT PART. The effects of crowding in densely populated regions of the lattice will not be detectable in the two dimensional diffusion coefficients produced by RMS and RMST. However, the diffusion coefficient in each direction can be obtained from the coordinates in this file.

It is desirable to be able to restart the diffusion calculation if the output analysis reveals that additional time evolution data is needed. Therefore, the main diffusion program and data storage schemes have been developed to facilitate program restarts. The parameters needed for

restarts are written into the file with the particle matrix and into the file with the program printout.

The previous paragraphs described the subroutines that comprise the diffusion simulator. A list of the parameters that are included in the data file for a run, a breakdown of the data file, a command file, and an output file are presented below. The following is a brief description of a typical TDMC diffusion run.

When the main program is executed, the run conditions are input from a data file and then written directly into an output file. The initial map is generated and saved. The correlation function for the initial configuration is produced; and the transition type matrix is initialized. The Boltzmann factors and activation energies for the specified neighboring interactions are computed before the transition types for each particle on the lattice are determined. After the probabilities for the initial configuration are calculated, the process of randomly selecting particles for motion begins.

Each time a TDMC loop is executed, a particle is chosen at random for motion. The time required for the move is recorded. An affected particle list is generated; and the transition types and probabilities for those particles are updated. At specified intervals, the particle concentrations in each section of the lattice are tabulated. The root mean square displacement is also calculated at various times

throughout the simulation. After the desired number of hops have been made, the TDMC loop is exited. The total time for the process is computed. When the TDMC routine is complete, the correlation function for the final configuration is generated. The mean square and root mean square displacements are computed. Finally, the coverage vs. time and displacement vs. time data are written to a file.

Testing and Modification of Code for Perfect Surfaces

The VAX intrinsic random number generator, which was originally used in our code to generate particle maps, produced configurations in which there were considerable fluctuations in the number of particles per unit area on the lattice. Because these fluctuations might have altered the initial appearance of the diffusion profiles, a statistical examination of the initial particle maps was deemed necessary. This observation also provoked us to verify the random nature of transition type selection. Selection of particles for migration must be completely random in order to satisfy the criteria for the time-dependent Monte Carlo model.

We elected to use chi-square testing to examine the data produced by the intrinsic random number generator that we were using. We also examined results from the functional form of a random number generator that became available.

This same statistic, which is defined by Equation 24, was also used to verify the randomness of transition type selection.

$$\text{Chi-Square} = \sum_{i=1}^N \frac{(N(\text{observed}) - N(\text{expected}))^2}{N(\text{expected})} \quad (24)$$

The intrinsic random number generator produced results that failed to fall within the designated limit ($\alpha=0.05$, 4 degrees of freedom, $\text{chi-square} < 9.49$) when used to generate particle coordinates. The functional form of the generator did, however, produce configurations that met with acceptable criteria for a random, uniform distribution. The reason for this discrepancy between the two generators is not well understood. However, it may be interesting to note that two random numbers are solicited from the same generator to produce one set of particle coordinates. Perhaps mapping the results from one generator into two dimensions contributes to the observed fluctuations in particle concentrations. On the other hand, if this were a factor, we would also expect the functional form of the generator to be affected.

In any event, the functional form of the generator was incorporated into the map code in place of the intrinsic generator. The configurations produced in ten trial runs using the the functional form of the generator are presented

in Table IV. Although it may appear that this discussion raises serious questions about the configuration of particles that we begin the simulation with, we should remember that the memory of this initial configuration is soon lost as the energetics of the system dictate the hopping sequence. The simulation will produce the most energetically favorable configuration after a sufficient number of hops, regardless of the initial particle configuration. As stated above, however, it may be inappropriate to emphasize trends in the short time diffusion profiles derived from a contrived initial configuration of particles.

Table IV. Chi-Square Statistics for Generation of 500 Random Sets of Coordinates on a Rectangular Lattice.

Trial Section	1	2	3	4	5	6	7	8	9	10
0	112	106	107	97	118	100	97	105	95	99
1	112	87	99	106	92	97	95	98	103	105
2	106	99	104	110	105	105	109	101	101	111
3	85	109	106	94	102	106	104	102	117	113
4	85	99	84	93	83	92	95	94	84	72
Chi-Square	6.3	2.9	3.6	2.3	7.1	1.3	1.6	0.7	5.8	11.0

A similar statistical experiment was conducted for transition type selection. Table V shows the results for the generation of 10,000 random numbers using the two generators. The 10,000 random numbers were placed in ten

'boxes'. The parameters $\alpha=0.05$, 9 degrees of freedom, and $\chi^2 < 16.9$ were used to assess the data.

In this case, the intrinsic form of the random number generator appeared to produce a better series of results than the functional form of the generator. However, the relative reliability of the two generators was less straight forward in this case. A single particle was placed on the lattice and allowed to make 100 hops which were directed by the intrinsic random number generator. The total number of hops in each of the four directions was tabulated in ten trials; and χ^2 statistics were generated. The results are shown in Table VI. The results clearly fall within the acceptable limit of $\chi^2 < 7.81$ at $\alpha=0.05$ with 3 degrees of freedom. Consequently, it was concluded that the selection of the direction of motion of the particle is random and that the intrinsic random number generator is suitable for use in selecting transition types.

Table V. Chi-Square Values for Generation of 10,000 Random Numbers.

INTRINSIC GENERATOR	FUNCTIONAL FORM OF GENERATOR
8.36	11.77
7.43	9.21
6.65	6.34
9.63	9.47
11.15	15.39
15.30	13.36
15.76	22.86
11.81	18.91
23.09	18.61
12.39	21.79

Table VI. Chi-Square Statistics for Single Particle Hopping with VAX Intrinsic Generator.

Trial	+Y	+X	-Y	-X	Chi-Square
1	28	31	28	13	7.92
2	24	21	29	26	1.36
3	23	21	36	20	6.64
4	26	22	22	30	1.76
5	28	21	27	24	1.20
6	27	23	25	25	0.32
7	31	26	19	24	2.96
8	17	20	33	30	7.12
9	24	21	33	22	3.60
10	28	26	23	23	0.72

When the statistical analysis above was completed, the seeds that generated the lowest chi-square values were selected for use in the diffusion simulation. A list of acceptable seeds for generating the map and selecting transition types is presented in Table VII.

Table VII. Best Seeds for Monte Carlo Random Number Generators.

MAP	TRANSITION TYPES
1516781432	1516781432
1516781979	1508061965
1531719631	1501301962
1508061965	1599547523
1501301962	1531061785

To further verify the validity of random transition type selection, we examined the "characteristic constants" for transition types. When the simulation is run with no neighboring interactions, the energies of all transition types are equal. The possibility for multiple transition types still exists, however, because the transition type number is assigned based on the neighborhood of the particle. The probability of selecting a particular transition type is directly proportional to its occupation number.

We incorporated the following scheme into the code. Each time a transition type is chosen, its address frequency is increased by one. The occupation number of the T list at that time is added into a running total. After a predetermined number of cycles, the average occupation number for each type is computed and stored. The simulation is divided into several such intervals. The address frequency and average occupation number for each type during each interval are recorded. These data are used to test the randomness of transition type selection based on the principle that with each transition type there is an associated constant given by:

$$\text{ONAV}_{\text{TYPE}} \cdot \text{BOLTZ}_{\text{TYPE}} / \text{ADDRESS}_{\text{TYPE}} = \text{CONST}_{\text{TYPE}} \quad (25)$$

$\text{ONAV}_{\text{TYPE}}$ is the average occupation number, $\text{BOLTZ}_{\text{TYPE}}$ is

the Boltzmann factor (characteristic transition time), and $ADDRESS_{TYPE}$ is the address frequency. This equation is simply a restatement of the criterion that transition types are chosen on the basis of their probabilities. The constant for each type should be the same during each interval of the calculation if the simulation is functioning as it should. This indeed proves to be the case.

The diffusion simulation was run at a coverage of 0.05 with no interactions. Data was collected during specified equal time intervals. The constant associated with each transition type was calculated. The results were consistent with Equation 25. The selection frequency of each occupied transition type remained constant throughout the calculation. Because Equation 25 is only valid when no interactions are present and because the time required to calculate these constants was non-negligible, this section of code was eliminated after the results were verified.

When a single particle is placed on the open lattice, there is only one possible transition type. Therefore, each hop should require a time interval equal to the inverse of the transition rate. The total time produced by the simulation for 100 random hops by a single particle was exactly 100 times the inverse of the transition rate, as expected. Thus, we concluded that the timer functioned according to our expectations.

However, we encountered a problem with the program timer

in our more complex initial trials. We compared the results for diffusion in systems with nearest and next-nearest neighbor interactions and found that the results were not physically meaningful. More specifically, as the next-nearest neighbor attractions were increased and island formation began to dominate, the algorithm produced larger diffusion coefficients. We know from experiment that island formation dramatically reduces the diffusion rate.

After a significant amount of searching, it became apparent that the normalized probabilities that were being used to calculate the time for particle hops were the source of the problem. Originally, the clock was designed to calculate the hop time from the normalized probabilities used in transition type selection. Unfortunately, the normalization process altered the relative time for each system. Although the data were internally consistent, no comparisons could be made between different systems. The system with high interactions was most severely affected because the squared term in the activation energy expression led to very small probabilities for jump events. Therefore, the normalization had a more profound influence on the time for each hop. We corrected this defect. The program now calculates a physically meaningful time for the diffusion process.

One of our concerns about the previously reported diffusion phenomena was that the results might be strongly

dependent on the seeds selected for the three, independent random number generators in the algorithm. Large fluctuations in the data could, in theory, make the diffusion coefficients for systems with different interactions indistinguishable. Several comparison runs have been obtained. The agreement between runs with different random number seeds is quite good.

When nearest neighbor or no interactions are considered, the agreement between the data from trials with different random number seeds is almost exact, as illustrated in Figure 86. We don't even see significant differences at short times when site blocking is prevalent. This reinforces our belief that the initial particle configuration does nothing to affect the long term diffusion behavior in these systems.

Excellent agreement was also observed for different seeds when next-nearest neighbor interactions were present, as the final example in Figure 86 demonstrates. Therefore, we conclude that the diffusion data for all of the systems under investigation are very reproducible and independent of the selected seeds.

Figure 86. Seed study comparisons for a) no interactions, b) attractions, c) repulsions, and d) repulsive-attractive interactions.

Running the Diffusion Simulation for Perfect Surfaces

The TDMC diffusion simulation is executed using a data file with the desired diffusion parameters and a command file to direct file management. The following is a list of the parameters as they are named in the FORTRAN algorithm. Data are saved at specified intervals throughout the calculation. Therefore, the simulation can be restarted where the previous run finished. The parameters needed for the restart are also included in the list. Data types are indicated in parentheses.

DATA	data file name (CHARACTER*60)
OUTPUT	output file name (CHARACTER*60)
TDMC	number of times T lists are rewritten (INTEGER)
ITDMC	number of hops before rewriting T lists (INTEGER)
LENGTH1	initial X lattice length (INTEGER)
LENGTH2	initial Y and final X and Y lattice length (INTEGER)
THETA	initial coverage (REAL)
DELL	interval length (INTEGER)
DELC	PART data save parameter (INTEGER)
DELR	RMS, COUNT, CORR data save parameter (INTEGER)
SEED	map seed (INTEGER)
SEED1	type seed (INTEGER)
SEED2	position seed (INTEGER)
EM	migration barrier for isolated adatom (REAL*16)
ENN	nearest neighbor interaction energy (REAL*16)
ENNN	next-nearest neighbor interaction energy (REAL*16)
KM	rate constant (REAL)
TEMP	temperature (REAL)
LAMBDA	jump length (REAL)
CTDMC	final value of TDMC from previous run
TIME	current time

TDMC and ITDMC determine the total number of hops executed, or the duration of the simulation. The transition

type lists have to be periodically updated or reinitialized. There are ITDMC hops before the lists are reinitialized. The lists are reinitialized TDMC times during the calculation. The total number of hops is the product ITDMC*TDMC. The value of ITDMC depends on the dimension of the transition type (T) lists, as discussed below.

The algorithm currently assumes a square diffusion system. The initial particle pack, however, can be deposited in a thin, rectangular strip. LENGTH1 is the initial occupied length in the X direction. LENGTH2 is the initial and final length in the Y direction. LENGTH1 and LENGTH2 can range from 0 to 199. LENGTH1 must be less than or equal to LENGTH2. In our investigations, the initial particle configuration has generally been deposited over an area that spans 10 X 200 lattice sites. Therefore, LENGTH1 is 9 (0-9 is ten lattice sites); and LENGTH2 is 199. The final lattice dimensions have been 200 X 200 sites. The final lattice dimension is assumed to be the same in both directions and equal to LENGTH2. THETA is the initial coverage. Thus, LENGTH1 times LENGTH2 times THETA should give the total number of particles. The algorithm can currently accommodate 1,000 particles.

DELL determines the length of the lattice divisions defined for diffusion profiles generated in SUBROUTINES COUNT and RMS. The PART matrix containing particle coordinates and trajectory lengths is saved at intervals of DELC, where DELC is some fraction of TDMC. For example, if TDMC is 10,000 and

ITDMC is 250, the total number of cycles, or particle hops, is 2,500,000. If DELC is 2,500, PART will be saved after every 2,500 times 250, or 625,000 hops. PART will be saved four times ($10,000/2,500$) during the calculation.

Correlation functions, displacements, and profiles are saved at intervals of DELR. DELR is defined in the same fashion as DELC. The quotient TDMC/DELR cannot exceed 20. In other words, these data can be saved 20 times during the simulation. DELC and DELR should be defined such that DELC is some multiple of DELR. This way, the data needed for the restart will be recorded in the output file. The PART matrix is generally saved fewer times during the simulation than other data due to limitations on available disk space.

SEED, SEED1, and SEED2 are large integers for the random number generators for the map, the transition type list, and the position of the particle on the list.

EM is the binding energy of an isolated adatom in units of RT where R is the ideal gas constant in kilocalories and T is the temperature in Kelvin. ENN is the nearest neighbor interaction energy in the same units. ENNN is the next-nearest neighbor interaction energy in the same units. TEMP is the temperature in Kelvin.

KM is the rate constant in inverse seconds. It is given a value of $1.0 \cdot 10^{10}$ in our investigations. LAMBDA is the jump length of the system of interest. This parameter is used to calculate the real particle displacements. In our

investigations, we have elected to give this quantity a value of 1.0. This leaves the displacements in somewhat arbitrary "lattice units" rather than in angstroms or nanometers. Our results can easily be multiplied by the lattice constant for an appropriate system to obtain realistic quantities.

Two additional parameters, which are not used in the first file for a run, are required to restart the calculation. CTDMC, the current cycle number, for the restart is the final value of TDMC from the preceding run. TIME is the time when the PART matrix was last saved. The calculation will resume from this point. All of the PART data for a run is written into the same file. Consequently, to restart a run, this file must be edited; and the initial and current PART matrices must be placed into separate files. The current values of the seeds for transition type selection are written into the output file (unit 6) adjacent to the time. Their current values must also be used in the restart file. The map seed is not needed because the initial configuration is read from the PART file.

Figures below illustrate data file creation. To start a new run, a file consisting of seven lines is needed. The position of each of the necessary input parameters described above are illustrated here. To restart a run, a data file consisting of eight lines with two additional parameters, the current cycle number and time, is needed. The actual FORTRAN, including formats, to read the data files is also given

below.

FORTTRAN FORMATS TO BEGIN A RUN:

```

      READ (5,1) , DATA
      READ (5,1) , OUTPUT
1     FORMAT (A60)
      READ (5,2) , TDMC, ITDMC
2     FORMAT (I5, I10)
      READ (5,3) , LENGTH1, LENGTH2, THETA, DELL, DELC, DELR
3     FORMAT (2I5, F10.5, 4I10)
      READ (5,4) , SEED, SEED1, SEED2
4     FORMAT (3I15)
      READ (5,5) , EM, ENN, ENNN, TEMP
5     FORMAT (4F10.6)
      READ (5,6) , KM, LAMBDA
6     FORMAT (E10.5, F10.4)

```

FORTTRAN FORMATS TO RESTART A RUN:

```

      READ (5,1) , DATA
      READ (5,1) , OUTPUT
1     FORMAT (A60)
      READ (5,2) , CTDMC, TDMC, ITDMC
2     FORMAT (3I10)
      READ (5,3) , TIME
3     FORMAT (E12.6)
      READ (5,4) , LENGTH1, LENGTH2, THETA, DELL, DELC, DELR
4     FORMAT (2I5, F10.5, 4I10)
      READ (5,5) , SEED, SEED1, SEED2
5     FORMAT (3I15)
      READ (5,6) , EM, ENN, ENNN, TEMP
6     FORMAT (4F10.6)
      READ (5,7) , KM, LAMBDA
7     FORMAT (E10.5, F10.4)

```

The following are examples of the command files for initial and restart runs. Correlation functions are written to unit 10; root mean square displacements are written to unit 11; and particle coordinates and trajectories are written to unit 13. To restart a run, the initial and current matrices of particle coordinates must be re-read. These matrices, which are obtained by editing and separating them

from the PART file, are assigned units 2 and 20, respectively.

COMMAND FILE TO BEGIN A RUN:

```
$ ASSIGN [UCHEHAB.RUN]RUN7490.DAT FOR005
$ ASSIGN [UCHEHAB.RUN]RUN7490.OUT FOR006
$ ASSIGN [UCHEHAB.RUN]CORR7490.DAT FOR010
$ ASSIGN [UCHEHAB.RUN]RMST7490.DAT FOR011
$ ASSIGN [UCHEHAB.RUN]PART7490.DAT FOR013
$ RUN [UCHEHAB.CODE]MAIN2101
```

COMMAND FILE TO RESTART A RUN:

```
$ ASSIGN [UCHEHAB.RUN]PART7490I.DAT FOR002
$ ASSIGN [UCHEHAB.RUN]RUN7490R.DAT FOR005
$ ASSIGN [UCHEHAB.RUN]RUN7490R.OUT FOR006
$ ASSIGN [UCHEHAB.RUN]CORR7490R.DAT FOR010
$ ASSIGN [UCHEHAB.RUN]RMST7490R.DAT FOR011
$ ASSIGN [UCHEHAB.RUN]PART7490R.DAT FOR013
$ ASSIGN [UCHEHAB.RUN]PART7490C.DAT FOR020
$ RUN [UCHEHAB.CODE]MAIN2101R
```

In addition, all of the input parameters and all of the output, with the exception of the data in the PART file, are written to unit 6.

As an aside, the PART file is composed of twelve columns. Each line contains all pertinent data for the particle whose number corresponds to that line of the array. The first and second columns give the particle's current X and Y coordinates, respectively. Columns 3 through 10 contain the transition types and position of the particle on each list. The sequence goes transition list, position on list, transition list, position on list, et cetera. Each particle can occupy as many as four transition lists. Column 11 contains the total trajectory length, or number of hops, for the particle. Column 12 contains the absolute trajectory

length, or actual displacement in lattice units, of the particle.

The following paragraphs are included here because the dimension of the transition type (T) array may prevent the simulation from linking or running depending on the available system memory. The program may have to be redimensioned to extend the analysis beyond those systems we have already examined or to run on other computer systems. Redimensioning will probably be the only modification necessary to examine a wide range of diffusion experiments. Therefore, a discussion regarding the pertinent program arrays is included at this juncture.

The critical size limiting factor in the simulation is certainly the number of particles in the system because this parameter directly dictates the length of the transition type lists. The size of the lattice can easily be extended without difficulty. However, the more particles there are in the system, the larger the transition type (T) lists must be to accommodate all potential particle hops. Each particle may have as many as four directions of motion, or transition types. Because there are 196 of these T lists, obtaining the required memory can rapidly become a formidable task as particle numbers increase. Because all of the lists must have the same dimension regardless of whether they are populated or not, a single highly populated transition type can dictate an unmanageable size for the diffusion program.

As the simulation proceeds, new potential transitions are added to the ends of the T lists; and old transitions are replaced by zeros. Thus, the lists must be of sufficient length to accommodate some predetermined number of particle hops, as well as their initial occupation. After the predetermined number of hops, we wipe the lists clean and sweep the lattice again to refill them. This eliminates the zero's, or vacancies, on the lists and prevents them from growing beyond a predetermined limit. Consequently, the dimension of the T lists will be reduced if the reinitialization process is carried out more frequently. The penalty for the reduction in memory requirements is certainly some increase in computation time. However, it is also time consuming to search through very long lists. Although there is clearly some optimum list length and reinitialization schedule, we have not specifically determined the "best" values for these parameters.

Therefore, the process of determining the necessary length of the transition type lists and the frequency for their reinitialization remains totally arbitrary. Each particle can hypothetically be on as many as four lists. Some transition types are also destined to be more highly populated than others. There is simply no way to predetermine precisely what the minimum size requirement for the T lists for the system of interest will be. Consequently, an estimate is required. There is really nothing to lose in making an

inappropriate estimate since the simulation will automatically stop if a transition list becomes filled during the migration process. The only penalty will be the loss of some computation time because the simulation will have to be restarted from the time of the last data save. The point is that, if memory constraints become a factor, there is certainly some latitude to be gained in experimenting with the lengths of these T lists.

The current lattice dimensions are 200 X 200. To change the lattice size, change `IMAP(0:199,0:199)` and `CMAP(0:199,0:199)`. To change the maximum number of particles, change `PART(X,12)` where X is the particle number. Also change arrays `IPART(X,12)` and `DIS(X)` when changing the particle number. To change the length of the transition type lists, change `T(196,Y,3)` where Y is the length of the lists. The current length is 10,000. These parameters appear in common blocks. Thus, many subroutines must be modified when the program is redimensioned. This procedure must be done very cautiously to avoid mysterious run-time errors.

The other potentially necessary modification of the code would be to convert from VAX FORTRAN to FORTRAN IV to run on other systems. The conversion would primarily consist of changing a few variable names and assigning statement labels wherever "END DO" appears.

The following command files can be used to compile and link the program. MAIN2101 is the main program for beginning

a run. MAIN2101R is the restart version of the main program. The diffusion data are saved at equal time intervals, rather than after some specified number of hops, in MAINTIME and MAINTIMER. The data file has precisely the same format. However, the parameter DELTAT, the time interval, must be assigned where it appears in the program. MAINTIME then has to be compiled and linked again.

COMMAND FILE TO COMPILE DIFFUSION ALGORITHM

```
$ FOR MAIN2101
$ FOR MAIN2101R
$ FOR TYPE
$ FOR ENERGY
$ FOR T2T2101
$ FOR MOVE2101
$ FOR ADD2101
$ FOR CLEAR2101
$ FOR PROB2101
$ FOR TYPE2101
$ FOR RAN210
$ FOR CLOCK
$ FOR CORR2
$ FOR RMS2101
$ FOR RMST2101
$ FOR COUNT2
```

COMMAND FILE TO LINK DIFFUSION ALGORITHM

```
$ LINK MAIN2101,TYPE,ENERGY,T2T2101,MOVE2101,ADD2101,-
CLEAR2101,PROB2101,TYPE2101,RAN210,CLOCK,CORR2,RMS2101,-
COUNT2,RMST2101
```

12345678901234567890123456789012345678901234567890										COLUMN
										LINE
RUN7490.DAT ← DATA FILE NAME										1
RUN7490.OUT ← OUTPUT FILE NAME										2
10000 250 ↑ ↑ TDMC ITDMC										3
9 199 0.5000 10 2000 1000 ↑ ↑ ↑ ↑ ↑ ↑ LENGTH1 LENGTH2 THETA DELL DELC DELR										4
1512191938 1501301962 1599547523 ↑ ↑ ↑ SEED SEED1 SEED2										5
0.1987 0.0000 0.0000 200.0 ↑ ↑ ↑ ↑ EM ENN ENNN TEMP										6
1.0E+10 1.0 ↑ ↑ KM LAMBDA										7

Figure 87. Time-dependent Monte Carlo data file breakdown for perfect surface.

12345678901234567890123456789012345678901234567890										COLUMN
										LINE
RUN7490R.DAT ← DATA FILE NAME										1
RUN7490R.OUT ← OUTPUT FILE NAME										2
10000 110000 250 ↑ ↑ ↑ CTDMC TDMC ITDMC										3
9 199 0.5000 10 2000 2000 ↑ ↑ ↑ ↑ ↑ ↑ LENGTH1 LENGTH2 THETA DELL DELC DELR										4
0.637157E-08										5
TIME										
0 600046420 1600076037 ↑ ↑ ↑ SEED SEED1 SEED2										6
0.1987 0.0000 0.0000 200.0 ↑ ↑ ↑ ↑ EM ENN ENNN TEMP										7
1.0E+10 1.0 ↑ ↑ KM LAMBDA										8

Figure 88. Time-dependent Monte Carlo data file breakdown for perfect surface restart.

12345678901234567890123456789012345678901234567890	COLUMN LINE
RUN7490.DAT	1
RUN7490.OUT	2
10000 250	3
9 199 0.5000 10 2000 1000	4
1512191938 1501301962 1599547523	5
0.1987 0.0000 0.0000 200.0	6
1.0E+10 1.0	7

Figure 89. Complete TDMC data file for perfect surface.

12345678901234567890123456789012345678901234567890	COLUMN LINE
RUN7490R.DAT	1
RUN7490R.OUT	2
10000 110000 250	3
9 199 0.5000 10 2000 1000	4
0.637157E-08 0 600046420 1600076037	5
0.1987 0.0000 0.0000 200.0	6
1.0E+10 1.0	7
	8

Figure 90. Complete TDMC data file for perfect surface restart.

APPENDIX B

TIME-DEPENDENT MONTE CARLO ALGORITHM FOR COADSORBATES

Time-Dependent Monte Carlo Algorithm for Coadsorbrates

This section provides a brief description of the FORTRAN subroutines involved in the diffusion simulation for coadsorbed species. Instructions for starting and continuing a run are provided later in this Appendix. A representative output file is also included there. The main program in this case is called COAD. All of the initial diffusion parameters are read by COAD. The main program is further comprised of all of the appropriate calls to subroutines and all of the format instructions for the output files.

The simulation begins in SUBROUTINE COADMAP where the initial random deposition of species A and B on the lattice is made. The number of atoms of each species, the dimensions of the initial region of deposition, and a seed for function RANDOM are supplied by the calling program. The dimensions of the regions which are initially occupied by A and B atoms can be different. All A atoms are deposited first and identified by number. B atoms are then numbered and placed on the lattice.

The first set of numbers used to identify transition types are read from a file by SUBROUTINE MIXTYPE. The second array to identify the transition types is then created and placed in common memory. The energy for each allowed transition type is computed in SUBROUTINE ENMIXA or ENMIXB.

Separate subroutines are required to allow for different binding energies of A and B atoms. The binding energies and all of the adparticle interactions must be passed from the calling program. There are A-A, B-B, and A-B interactions for both nearest and next-nearest neighbor positions. The procedure described above is used to reduce the error in the energy approximation. The energy window width, or error, is written to the output file. Each transition type is placed on the appropriate energy list.

Transition types for the initial and migrating systems are produced in SUBROUTINE TRANSITION. The lattice dimension, route, and an index that distinguishes between particle species are required. The particle of interest is first identified as species A or B. The transition type is then determined by counting the number of each species surrounding this particle. All potential particle hops are recorded in the PART matrix. Particle numbers and coordinates are placed on the appropriate energy lists.

The probability associated with each list of potential transitions is computed from the Boltzmann factors produced in the energy routines. SUBROUTINE EPROB either computes probabilities for all 200 energy lists or just for those lists affected by a move. Particles are randomly selected, according to their occurrence probabilities, from the lists for migration. As before, two seeds are required for the random number generators in SUBROUTINE RANEN. The selected

energy list and the position of the migrating particle on that list are passed back to the calling program.

SUBROUTINE MOVE directs the particle to its final destination and determines the direction of motion. A list of affected particles is generated; and particle displacements are recorded. SUBROUTINE CLOCK computes the time for each hop and the total time for the simulation. The total number of particles, the number of the selected energy list, the particle number, the direction of motion, the rate constant, and the probabilities must be provided to CLOCK. New transition types, energy lists, and probabilities are determined for all affected particles. The affected energy lists are recorded by SUBROUTINES ADD and CLEAR and corrected after each move.

At specified intervals throughout the simulation, time evolution data is recorded. SUBROUTINE COUNT monitors the concentration of A and B atoms in predetermined regions of the lattice as a function of time. The index which determines the particle species, the final lattice dimension, and the interval length must be provided. SUBROUTINE CORRMIX uses the number of each species, the total particle number, and the lattice length to compute correlation functions at specified intervals. The correlation function for each species and the total correlation function are computed.

SUBROUTINES RMSALL and RMSTALL compute the displacements for individual species and for all particles combined as a function of time. Displacements are also computed for particles occupying specified intervals of predetermined length. The number of each species, lattice dimension, initial particle positions, and jump length are used in these calculations.

Running the Diffusion Simulation for Coadsorbates

The time-dependent Monte Carlo diffusion simulation for coadsorbates is executed using a data file with the desired diffusion parameters and a command file to direct file management. The following is a list of the parameters as they are named in the FORTRAN algorithm. Data are saved at specified intervals throughout the calculation. Therefore, the simulation can be restarted where the previous run finished. The parameters needed for the restart are also included in the list. Data types are indicated in parentheses.

DATA	data file name (CHARACTER*60)
OUTPUT	output file name (CHARACTER*60)
TDMC	number of times ET lists are rewritten (INTEGER)
ITDMC	number of hops before rewriting ET lists (INTEGER)
LENGTH1	initial X lattice length for species A (INTEGER)
LENGTH2	initial Y and final X and Y lattice length (INTEGER)
LENGTH3	initial X lattice length for species B (INTEGER)

SEED	map seed (INTEGER)
SEED1	energy seed (INTEGER)
SEED2	position seed (INTEGER)
DELL	interval length (INTEGER)
DELC	PART data save parameter (INTEGER)
DELT	RMSALL, COUNTALL, CORRMIX data save parameter (INTEGER)
ATHETA	initial coverage of species A (REAL)
BTHETA	initial coverage of species B (REAL)
EMA	migration barrier for isolated A adatom (REAL*16)
EMB	migration barrier for isolated B adatom (REAL*16)
ANN	nearest neighbor interaction energy A-A (REAL*16)
ANNN	next-nearest neighbor interaction energy A-A (REAL*16)
BNN	nearest neighbor interaction energy B-B (REAL*16)
BNNN	next-nearest neighbor interaction energy B-B (REAL*16)
ABNN	nearest neighbor interaction energy A-B (REAL*16)
ABNNN	next-nearest neighbor interaction energy A-B (REAL*16)
KM	rate constant (REAL)
TEMP	temperature (REAL)
LAMBDA	jump length (REAL)
CTDMC	final value of TDMC from previous run (INTEGER)
TIME	current time (REAL*16)

TDMC and ITDMC determine the total number of hops executed, or the duration of the simulation. The energy lists have to be periodically updated or reinitialized. There are ITDMC hops before the lists are reinitialized. The lists are reinitialized TDMC times during the calculation. The total number of hops is the product ITDMC*TDMC. The value of ITDMC depends on the dimension of the energy (ET) lists, as discussed in Appendix A.

The algorithm currently assumes a square diffusion system. The initial region occupied by each species, however, can be a thin, rectangular strip. LENGTH1 is the initial length in the X direction occupied by species A.

LENGTH3 is the initial length in the X direction occupied by species B. LENGTH2 is the initial and final length in the Y direction. LENGTH1, LENGTH2, and LENGTH3 can range from 0 to 199. LENGTH1 and LENGTH3 must be less than or equal to LENGTH2. In our investigations, the initial configuration of the predominant species has generally been deposited over an area that spans 10 X 200 lattice sites. Therefore, LENGTH1 is 9 (0-9 is ten lattice sites); and LENGTH2 is 199. The minority species has occupied the entire lattice. This implies a value of 199 for LENGTH3, also. The final lattice dimensions have been 200 X 200 sites. The final lattice dimension is assumed to be the same in both directions and equal to LENGTH2.

ATHETA is the initial coverage of species A. BTHETA is the initial coverage of species B. Thus, LENGTH1 times LENGTH2 times ATHETA should give the total number of A particles. LENGTH3 times LENGTH2 times BTHETA should give the total number of B particles. The algorithm can currently accommodate a total of 1,000 particles.

DELL determines the length of the lattice divisions defined for diffusion profiles generated in SUBROUTINES COUNTALL and RMSALL. The PART matrix containing particle coordinates and trajectory lengths is saved at intervals of DELC, where DELC is some fraction of TDMC. For example, if TDMC is 10,000 and ITDMC is 250, the total number of cycles, or particle hops, is 2,500,000. If DELC is 2,500, PART will

be saved after every 2,500 times 250, or 625,000 hops. PART will be saved four times ($10,000/2,500$) during the calculation.

Correlation functions, displacements, and profiles are saved at intervals of DELT. DELT is defined in the same fashion as DELC. The quotient TDMC/DELT cannot exceed 20. In other words, these data can be saved 20 times during the simulation. DELC and DELT should be defined such that DELC is some multiple of DELT. This way, the data needed for the restart will be recorded in the output file. The PART matrix is generally saved fewer times during the simulation than other data due to limitations on available disk space.

SEED, SEED1, and SEED2 are large integers for the random number generators for the map, the transition type list, and the position of the particle on the list.

EMA is the binding energy of an isolated A adatom in units of RT where R is the ideal gas constant in kilocalories and T is the temperature in Kelvin. EMB is the binding energy of an isolated B atom. ANN is the nearest neighbor interaction energy between two A atoms in units of RT . ANNN is the next-nearest neighbor interaction energy between two A atoms in the same units. The parameters for B-B interactions and A-B interactions are defined in precisely the same manner. TEMP is the temperature in Kelvin.

KM is the rate constant in inverse seconds. It is given a value of $1.0 \cdot 10^{10}$ in our investigations. LAMBDA is the jump

length of the system of interest. This parameter is used to calculate the real particle displacements. In our investigations, we have elected to give this quantity a value of 1.0. This leaves the displacements in somewhat arbitrary "lattice units" rather than in angstroms or nanometers. Our results can easily be multiplied by the lattice constant for an appropriate system to obtain realistic quantities.

Two additional parameters, which are not used in the first file for a run, are required to restart the calculation. CTDMC, the current cycle number, for the restart is the final value of TDMC from the preceding run. TIME is the time when the PART matrix was last saved. The calculation will resume from this point. All of the PART data for a run is written into the same file. Consequently, to restart a run, this file must be edited; and the initial and current PART matrices must be placed into separate files. The current values of the seeds for transition type selection are written into the output file (unit 6) adjacent to the time. Their current values must also be used in the restart file. The map seed is not needed because the initial configuration is read from the PART file.

Figures below illustrate data file creation. To start a new run, a file consisting of twelve lines is needed. The position of each of the necessary input parameters described above are illustrated here. To restart a run, a data file consisting of thirteen lines with two additional parameters,

the current cycle number and time, is needed. The actual FORTRAN, including formats, to read the data files is also given below.

FORTRAN FORMATS TO BEGIN A RUN:

```

      READ (5,1) , DATA
      READ (5,1) , OUTPUT
1     FORMAT (A60)
      READ (5,2) , TDMC, ITDMC
2     FORMAT (2I10)
      READ (5,3) , LENGTH1, LENGTH2, LENGTH3
3     FORMAT (3I5)
      READ (5,4) , SEED, SEED1, SEED2
4     FORMAT (3I15)
      READ (5,5) , DELL, DELC, DELT
5     FORMAT (3I5)
      READ (5,6) , ATHETA, BTHETA
6     FORMAT (2F10.5)
      READ (5,7) , EMA, EMB
7     FORMAT (2F10.5)
      READ (5,8) , ANN, ANNN
      READ (5,8) , BNN, BNNN
      READ (5,8) , ABNN, ABNNN
8     FORMAT (2F10.5)
      READ (5,9) , KM, TEMP, LAMBDA
9     FORMAT (E10.5, 2F10.5)

```

FORTRAN FORMATS TO RESTART A RUN:

```

      READ (5,1) , DATA
      READ (5,1) , OUTPUT
1     FORMAT (A60)
      READ (5,2) , TDMC, ITDMC, CTDMC
2     FORMAT (3I10)
      READ (5,3) , TIME
3     FORMAT (E12.6)
      READ (5,4) , LENGTH1, LENGTH2, LENGTH3
4     FORMAT (3I5)
      READ (5,5) , SEED, SEED1, SEED2
5     FORMAT (3I15)
      READ (5,6) , DELL, DELC, DELT
6     FORMAT (3I5)
      READ (5,7) , ATHETA, BTHETA
7     FORMAT (2F10.5)
      READ (5,8) , EMA, EMB
8     FORMAT (2F10.5)
      READ (5,9) , ANN, ANNN

```

```

      READ(5,9), BNN,BNNN
      READ(5,9), ABNN,ABNNN.
9      FORMAT(2F10.5)
      READ(5,10), KM,TEMP,LAMBDA
10     FORMAT(E10.5,2F10.5)

```

The following are examples of the command files for initial and restart runs. Correlation functions are written to unit 10; root mean square displacements are written to unit 11; and particle coordinates and trajectories are written to unit 13. Units 3 and 4 are for transition type initialization. To restart a run, the initial and current matrices of particle coordinates must be re-read. These matrices, which are obtained by editing and separating them from the PART file, are assigned units 2 and 20, respectively.

COMMAND FILE TO BEGIN A RUN:

```

$ ASSIGN [UCHEHAB.COADS]INDEX.DAT FOR003
$ ASSIGN [UCHEHAB.COADS]TNUM.DAT FOR004
$ ASSIGN [UCHEHAB.RUNC]CRUN18.DAT FOR005
$ ASSIGN [UCHEHAB.RUNC]CRUN18.OUT FOR006
$ ASSIGN [UCHEHAB.RUNC]CCORR18.DAT FOR010
$ ASSIGN [UCHEHAB.RUNC]CRMST18.DAT FOR011
$ ASSIGN [UCHEHAB.RUNC]CPART18.DAT FOR013
$ RUN [UCHEHAB.COADS]COADM

```

COMMAND FILE TO RESTART A RUN:

```

$ ASSIGN [UCHEHAB.COADS]INDEX.DAT FOR003
$ ASSIGN [UCHEHAB.COADS]TNUM.DAT FOR004
$ ASSIGN [UCHEHAB.RUNC]CPART18I.DAT FOR002
$ ASSIGN [UCHEHAB.RUNC]CRUN18R.DAT FOR005
$ ASSIGN [UCHEHAB.RUNC]CRUN18R.OUT FOR006
$ ASSIGN [UCHEHAB.RUNC]CCORR18R.DAT FOR010
$ ASSIGN [UCHEHAB.RUNC]CRMST18R.DAT FOR011
$ ASSIGN [UCHEHAB.RUNC]CPART18R.DAT FOR013
$ ASSIGN [UCHEHAB.RUNC]CPART18C.DAT FOR020

```

```
$ RUN [UCHEHAB.COADS]COADMR
```

In addition, all of the input parameters and all of the output, with the exception of the data in the PART file, are written to unit 6. The command files for compiling and linking the code are presented below. COADM is the main program to begin a run. COADMR is the restart version of the main program.

COMMAND FILE TO COMPILE COADSORBATE CODE

```
$ FOR COADM  
$ FOR COADMR  
$ FOR ADD  
$ FOR CLEAR  
$ FOR CLOCK  
$ FOR COADMAP  
$ FOR CORRMIX  
$ FOR ELIST  
$ FOR EWINA  
$ FOR EWINB  
$ FOR EPROB  
$ FOR MIXTYPE  
$ FOR MOVE  
$ FOR RANEN  
$ FOR RMSALL  
$ FOR RMSTALL  
$ FOR COUNTALL
```

COMMAND FILE TO LINK COADSORBATE CODE

```
$ LINK COADM,ADD,CLEAR,CLOCK,COADMAP,CORRMIX,ELIST,-  
EWINA,EWINB,EPROB,MIXTYPE,MOVE,RANEN,RMSALL,RMSTALL,-  
COUNTALL
```

1234567890123456789012345678901234567890	COLUMN LINE
CRUN18.DAT ← DATA FILE NAME	1
CRUN18.OUT ← OUTPUT FILE NAME	2
2000 250 ↑ ↑ TDMC ITDMC	3
9 199 199 ↑ ↙ ↘ LENGTH1 LENGTH2 LENGTH3	4
1512191938 1501301962 1599547523 ↑ ↑ ↑ SEED SEED1 SEED2	5
10 500 200 ↑ ↑ ↑ DEL1 DELC DELT	6
0.45000 0.00250 ↑ ↑ ATHETA BTHETA	7
0.19870 1.98700 ↑ ↑ EMA EMB	8
-0.39740 0.59610 ↑ ↑ ANN ANNN	9
-0.79480 0.99350 ↑ ↑ BNN BNNN	10
7.94800 9.93500 ↑ ↑ ABNN ABNNN	11
1.0E+10 200.00000 1.00000 ↑ ↑ ↑ KM TEMP LAMBDA	12

Figure 91. Time-dependent Monte Carlo data file breakdown for coadsorbates.

Figure 92. Time-dependent Monte Carlo data file breakdown for coadsorbate restart.

12345678901234567890123456789012345678901234567890	COLUMN LINE
CRUN18.DAT	1
CRUN18.OUT	2
2000 250	3
9 199 199	4
1512191938 1501301962 1599547523	5
10 500 200	6
0.45000 0.00250	7
0.19870 1.98700	8
-0.39740 0.59610	9
0.79480 0.99350	10
7.94800 9.93500	11
1.0E+10 200.00000 1.00000	12

Figure 93. Complete TDMC data file for coadsorbates.

12345678901234567890123456789012345678901234567890	COLUMN LINE
CRUN18R.DAT	1
CRUN18R.OUT	2
5000 250 2000	3
0.594919E-08	4
9 199 199	5
0 1501301962 1599547523	6
10 500 200	7
0.45000 0.00250	8
0.19870 1.98700	9
-0.39740 0.59610	10
0.79480 0.99350	11
7.94800 9.93500	12
1.0E+10 200.00000 1.00000	13

Figure 94. Complete TDMC data file for coadsorbate restart.

APPENDIX C

TIME-DEPENDENT MONTE CARLO ALGORITHM FOR STEPPED SURFACES

Time-Dependent Monte Carlo Algorithm for Stepped Surfaces

This section provides a brief description of the FORTRAN subroutines involved in the diffusion simulation for stepped surfaces. Instructions for starting and continuing a run are provided in Appendix C. A representative output file is also included there. The main program in this case is called STEPSURF.

The series of logical statements used to rule out the impossible transition types are contained in file TNUM.FOR. As stated previously, the list of allowed transition types for these stepped systems is substantially reduced by requiring that the steps be smooth and run the length of the lattice in the Y direction. The transition type is a function of nearest and next-nearest neighbor particles and steps sites. Thus, four indices are used to uniquely define each of the initial and final neighborhoods.

If the surface steps are spaced at equal intervals, the simulation begins in SUBROUTINE STEP MAP. The lattice length and step interval length are provided by the calling program. This algorithm can be modified to create steps that are either increasing or decreasing in height as we move from left to right. If steps are placed at irregular intervals or

if the step barrier alternates from positive to negative, the appropriate Z coordinates will have to be derived in some other fashion. It is only mandatory that some initial map of the surface topography be supplied to the main program. The details of the topography are unimportant as long as they fall within the constraints described previously.

We next proceed to SUBROUTINE RANMAP where an initial random configuration of particles on the lattice is produced. There are no constraints on the initial configuration of particles beyond those which have already been described. For a random distribution of particles, we need to supply the number of particles, the dimensions of the initial occupied region, and a seed for RANDOM. We are also still free to place particles on the lattice in any predescribed initial pattern.

The sets of four indices which were produced by PROGRAM ILIST to identify transition types are read from a file by SUBROUTINE MIXTYPE. The four indices describe 100 unique particle neighborhoods on the stepped surface. The second, two dimensional array of allowed transition types is then created by eliminating the physically meaningless combinations of these 100 initial and final states. The allowed transitions are placed in common memory. The energy for each allowed transition type is computed in SUBROUTINE ENMIX given the binding energy, the neighboring adparticle interactions, and the step barriers. The error in the energy

approximation is reduced here by successively reducing the width of the occupied energy lists. The final energy window width, or error, is written to the output file. Each allowed transition type is also assigned to the appropriate energy list.

Transition types for the initial and migrating systems are produced in SUBROUTINE TRANSITION. The lattice final dimension and the route are required, as before. The number of adjacent particles and steps at the site of interest is tabulated in a fairly complicated fashion. The energy at the initial and final sites is always a function of the adjacent step barrier, regardless of whether or not that site is filled. The particle is not, however, allowed to move to a step site if it is occupied. The counting scheme is significantly simplified by restricting the steps to be smooth and continuous and to run in the Y direction. All potential particle hops are recorded in the PART matrix. The occupation number of each transition type is tabulated for use in the program timer. Finally, particle numbers and coordinates are placed on the appropriate energy lists.

The probability associated with each list of potential transitions is computed from the Boltzmann factors produced in the energy routine. SUBROUTINE EPROB either computes probabilities for all 200 energy lists or just for those lists affected by a move. Particles are randomly selected, according to their occurrence probabilities, from the lists

for migration. As before, two seeds are required for the random number generators in SUBROUTINE RANEN. The selected energy list and the position of the migrating particle on that list are passed back to the calling program.

SUBROUTINE MOVE directs the particle to its final destination and determines the direction of motion. A list of affected particles is generated; and particle displacements are recorded. Actually, only the total trajectory length, or hop number, for each particle is counted in this case. It is now too time consuming to keep track of the absolute particle displacement after each hop due to the three dimensional nature of the substrate. The total particle displacement is computed at the end of the simulation. Thus, no information is lost in excluding this step.

SUBROUTINE CLOCK computes the time for each hop and the total time for the simulation. The total number of particles, the number of the selected energy list, the particle number, the direction of motion, the rate constant, and the probabilities must be provided to CLOCK. New transition types, energy lists, and probabilities are determined for all affected particles. The affected energy lists are recorded by SUBROUTINES ADD and CLEAR and corrected after each move.

At specified intervals throughout the simulation, time evolution data is recorded. SUBROUTINE COUNT monitors the concentration of atoms in predetermined regions of the lattice as a function of time. These results should be

particularly useful as we began to determine the directional effects that the steps have on diffusion. The final lattice dimension, and the interval length must be provided to COUNT. SUBROUTINE CORRT uses the number of particles, the lattice length, and the surface topography map to compute correlation functions at specified times throughout the calculation. The correlation function for the particles is determined, as before. In addition, the particle concentrations near the step edges are also computed.

SUBROUTINES RMS and RMST compute the displacements in three dimensions for the entire system of particles. Displacements are also computed for collections of particles occupying specified intervals of predetermined length. The number of particles, lattice dimension, initial and current particles positions, and jump length are used in these calculations.

Running the Diffusion Simulation for Stepped Surfaces

The time-dependent Monte Carlo diffusion simulation for stepped surfaces is also executed using a data file with the desired diffusion parameters and a command file to direct file management. The following is a list of the parameters as they are named in the FORTRAN algorithm. Data are saved at specified intervals throughout the calculation. Therefore, the simulation can be restarted where the previous run

finished. The parameters needed for the restart are also included in the list. Data types are indicated in parentheses.

DATA	data file name (CHARACTER*60)
OUTPUT	output file name (CHARACTER*60)
TDMC	number of times ET lists are rewritten (INTEGER)
ITDMC	number of hops before rewriting ET lists (INTEGER)
LENGTH1	initial X lattice length (INTEGER)
LENGTH2	initial Y and final X and Y lattice length (INTEGER)
SEED	map seed (INTEGER)
SEED1	energy seed (INTEGER)
SEED2	position seed (INTEGER)
DELL	interval length (INTEGER)
DELC	PART data save parameter (INTEGER)
DELS	step interval length
DELT	RMS, COUNT, CORRT data save parameter (INTEGER)
THETA	initial coverage (REAL)
EM	migration barrier for isolated adatom (REAL*16)
ENN	nearest neighbor interaction energy (REAL*16)
ENNN	next-nearest neighbor interaction energy (REAL*16)
TNN	nearest neighbor step barrier (REAL*16)
TNNN	next-nearest neighbor step barrier (REAL*16)
KM	rate constant (REAL)
TEMP	temperature (REAL)
LAMBDA	jump length (REAL)
CTDMC	final value of TDMC from previous run (INTEGER)
TIME	current time (REAL*16)

TDMC and ITDMC determine the total number of hops executed, or the duration of the simulation. The energy lists have to be periodically updated or reinitialized. There are ITDMC hops before the lists are reinitialized. The lists are reinitialized TDMC times during the calculation. The total number of hops is the product ITDMC*TDMC. The value of ITDMC depends on the dimension of the energy (ET) lists, as discussed in Appendix A.

The algorithm currently assumes a square diffusion system. The initial region occupied by particles, however, can be a thin, rectangular strip. LENGTH1 is the initial length in the X direction occupied by particles. LENGTH2 is the initial and final length in the Y direction. LENGTH1 and LENGTH2 can range from 0 to 199. LENGTH1 must be less than or equal to LENGTH2. In our investigations, the initial configuration of particles has generally been deposited over an area that spans 10 X 200 lattice sites. Therefore, LENGTH1 is 9 (0-9 is ten lattice sites); and LENGTH2 is 199. The final lattice dimensions have been 200 X 200 sites. The final lattice dimension is assumed to be the same in both directions and equal to LENGTH2.

THETA is the initial coverage of adatoms. LENGTH1 times LENGTH2 times THETA should give the total number of particles. The algorithm can currently accommodate a total of 1,000 particles.

DELL is the interval length used in RMS and COUNT. The PART matrix containing particle coordinates and trajectory lengths is saved at intervals of DELC, where DELC is some fraction of TDMC. For example, if TDMC is 10,000 and ITDMC is 250, the total number of cycles, or particle hops, is 2,500,000. If DELC is 2,500, PART will be saved after every 2,500 times 250, or 625,000 hops. PART will be saved four times (10,000/2,500) during the calculation.

Correlation functions, displacements, and profiles are

saved at intervals of DELT. DELT is defined in the same fashion as DELC. The quotient TDMC/DELT cannot exceed 20. In other words, these data can be saved 20 times during the simulation. DELC and DELT should be defined such that DELC is some multiple of DELT. This way, the data needed for the restart will be recorded in the output file. The PART matrix is generally saved fewer times during the simulation than other data due to limitations on available disk space.

SEED, SEED1, and SEED2 are large integers for the random number generators for the map, the transition type list, and the position of the particle on the list.

EM is the binding energy of an isolated adatom in units of RT where R is the ideal gas constant in kilocalories and T is the temperature in Kelvin. ENN is the nearest neighbor interaction energy in the same units. ENNN is the next-nearest neighbor interaction energy. TNN is the interaction energy between a particle and an adjacent step. TNNN is the interaction energy between a particle and step sites at next-nearest neighbor positions. TNN and TNNN are always positive. ENN and ENNN are negative for repulsive interactions. TEMP is the temperature in Kelvin.

KM is the rate constant in inverse seconds. It is given a value of $1.0 \cdot 10^{10}$ in our investigations. LAMBDA is the jump length of the system of interest. This parameter is used to calculate the real particle displacements. In our investigations, we have elected to give this quantity a value

of 1.0. This leaves the displacements in somewhat arbitrary "lattice units" rather than in angstroms or nanometers. Our results can easily be multiplied by the lattice constant for an appropriate system to obtain realistic quantities.

Two additional parameters, which are not used in the first file for a run, are required to restart the calculation. CTDMC, the current cycle number, for the restart is the final value of TDMC from the preceding run. TIME is the time when the PART matrix was last saved. The calculation will resume from this point. All of the PART data for a run is written into the same file. Consequently, to restart a run, this file must be edited; and the initial and current PART matrices must be placed into separate files. The current values of the seeds for transition type selection are written into the output file (unit 6) adjacent to the time. Their current values must also be used in the restart file. The map seed is not needed because the initial configuration is read from the PART file.

Figures below illustrate data file creation. To start a new run, a file consisting of eleven lines is needed. The position of each of the necessary input parameters described above are illustrated here. To restart a run, a data file consisting of twelve lines with two additional parameters, the current cycle number and time, is needed. The actual FORTRAN, including formats, to read the data files is also given below.

FORTRAN FORMATS TO BEGIN A RUN:

```

      READ (5,1), DATA
      READ (5,1), OUTPUT
1     FORMAT(A60)
      READ (5,2), TDMC, ITDMC
2     FORMAT(2I10)
      READ (5,3), LENGTH1, LENGTH2
3     FORMAT(2I5)
      READ (5,4), SEED, SEED1, SEED2
4     FORMAT(3I15)
      READ (5,5), DELL, DELC, DELS, DELT
5     FORMAT(4I5)
      READ (5,6), THETA
6     FORMAT(F10.5)
      READ (5,7) EM
7     FORMAT(F10.5)
      READ (5,8), ENN, ENNN
      READ (5,8), TNN, TNNN
8     FORMAT(2F10.5)
      READ (5,9), KM, TEMP, LAMBDA
9     FORMAT(E10.5, 2F10.5)

```

FORTRAN FORMATS TO RESTART A RUN:

```

      READ (5,1), DATA
      READ (5,1), OUTPUT
1     FORMAT(A60)
      READ (5,2), TDMC, ITDMC, CTDMC
2     FORMAT(3I10)
      READ (5,3), TIME
3     FORMAT(E12.6)
      READ (5,4), LENGTH1, LENGTH2
4     FORMAT(2I5)
      READ (5,5), SEED, SEED1, SEED2
5     FORMAT(3I15)
      READ (5,6), DELL, DELC, DELS, DELT
6     FORMAT(4I5)
      READ (5,7), THETA
7     FORMAT(F10.5)
      READ (5,8) EM
8     FORMAT(F10.5)
      READ (5,9), ENN, ENNN
      READ (5,9), TNN, TNNN
9     FORMAT(2F10.5)
      READ (5,10), KM, TEMP, LAMBDA
10    FORMAT(E10.5, 2F10.5)

```

The following are examples of the command files for initial and restart runs. Correlation functions are written to unit 10; root mean square displacements are written to unit 11; and particle coordinates and trajectories are written to unit 13. Units 3 and 4 are for transition type initialization. To restart a run, the initial and current matrices of particle coordinates must be re-read. These matrices, which are obtained by editing and separating them from the PART file, are assigned units 2 and 20, respectively.

COMMAND FILE TO BEGIN A RUN:

```
$ ASSIGN [UCHEHAB.STEP]INDEX.DAT FOR003
$ ASSIGN [UCHEHAB.STEP]TNUM.DAT FOR004
$ ASSIGN [UCHEHAB.RUNS]SRUN35.DAT FOR005
$ ASSIGN [UCHEHAB.RUNS]SRUN35.OUT FOR006
$ ASSIGN [UCHEHAB.RUNS]SCORR35.DAT FOR010
$ ASSIGN [UCHEHAB.RUNS]SRMST35.DAT FOR011
$ ASSIGN [UCHEHAB.RUNS]SPART35.DAT FOR013
$ RUN [UCHEHAB.STEP]STEPUP
```

COMMAND FILE TO RESTART A RUN:

```
$ ASSIGN [UCHEHAB.RUNS]SPARTIP.DAT FOR002
$ ASSIGN [UCHEHAB.STEP]INDEX.DAT FOR003
$ ASSIGN [UCHEHAB.STEP]TNUM.DAT FOR004
$ ASSIGN [UCHEHAB.RUNS]SRUN35R5.DAT FOR005
$ ASSIGN [UCHEHAB.RUNS]SRUN35R5.OUT FOR006
$ ASSIGN [UCHEHAB.RUNS]SCORR35R5.DAT FOR010
$ ASSIGN [UCHEHAB.RUNS]SRMST35R5.DAT FOR011
$ ASSIGN [UCHEHAB.RUNS]SPART35R5.DAT FOR013
$ ASSIGN [UCHEHAB.RUNS]SPART35C.DAT FOR020
$ RUN [UCHEHAB.STEP]STEPUPR
```

In addition, all of the input parameters and all of the output, with the exception of the data in the PART file, are

written to unit 6. The command files for compiling and linking the code are presented below. COADM is the main program to begin a run. COADMR is the restart version of the main program.

COMMAND FILE TO COMPILE STEP CODE

```
$ FOR STEPM  
$ FOR STEPMR  
$ FOR ADD  
$ FOR CLEAR  
$ FOR CLOCK  
$ FOR CORRT  
$ FOR COUNT  
$ FOR ELIST  
$ FOR EWIN  
$ FOR EPROB  
$ FOR MIXTYPE  
$ FOR MOVE  
$ FOR RANEN  
$ FOR RANMAP  
$ FOR RMS  
$ FOR RMST  
$ FOR STEPMA
```

COMMAND FILE TO LINK STEP CODE

```
$ LINK/EXE=STEPUP STEPM,ADD,CLEAR,CLOCK,CORRT,COUNT,-  
ELIST,EWIN,EPROB,MIXTYPE,MOVE,RANEN,RANMAP,RMS,RMST,STEPMA
```

1234567890123456789012345678901234567890	COLUMN
	LINE
SRUN35.DAT ← DATA FILE NAME	1
SRUN35.OUT ← OUTPUT FILE NAME	2
30000 250 ↑ ↑ TDMC ITDMC	3
9 199 ↑ ↙ LENGTH1 LENGTH2	4
1512191938 1501301962 1599547523 ↑ ↑ ↑ SEED SEED1 SEED2	5
10 2000 40 500 ↑ ↑ ↑ ↑ DELL DELC DELS DELT	6
0.50000 ↑ THETA	7
0.19870 ↑ EM	8
-0.39740 0.59610 ↑ ↗ ENN ENNN	9
0.39740 0.28100 ↑ ↗ TNN TNNN	10
1.0E+10 200.0 1.00000 ↑ ↗ ↑ KM TEMP LAMBDA	11

Figure 95. Time-dependent Monte Carlo data file breakdown for surface steps.

12345678901234567890123456789012345678901234567890	COLUMN LINE
SRUN35R5.DAT ← DATA FILE NAME	1
SRUN35R5.OUT ← OUTPUT FILE NAME	2
30000 ↑ TIMC	3
250 ↑ ITDMC	
24000 ↑ CTDMC	
0.683993E-07 ↑ TIME	4
9 199 ↑ ↙ LENGTH1 LENGTH2	5
SEED ↗ 0	
116223476 ↑ SEED1	6
1149027109 ↑ SEED2	
10 2000 40 500 ↑ ↑ ↑ ↑ DELL DELC DELS DELT	7
0.50000 ↑ THETA	8
0.19870 ↑ EM	9
-0.39740 0.59610 ↑ ↗ EMN EMNN	10
0.39740 0.28100 ↑ ↗ TMN TMNN	11
1.0E+10 200.0 1.00000 ↑ ↑ ↑ KM TEMP LAMBDA	12

Figure 96. Time-dependent Monte Carlo data file breakdown for surface step restart.

12345678901234567890123456789012345678901234567890	COLUMN LINE
SRUN35.DAT	1
SRUN35.OUT	2
30000 250	3
9 199	4
1512191938 1501301962 1599547523	5
10 2000 40 500	6
0.50000	7
0.19870	8
-0.39740 0.59610	9
0.39740 0.28100	10
1.0E+10 200.0 1.00000	11

Figure 97. Complete TDMC data file for surface steps.

12345678901234567890123456789012345678901234567890	COLUMN LINE
SRUN35R5.DAT	1
SRUN35R5.OUT	2
30000 250 24000	3
0.683993E-07	4
9 199	5
0 116223476 1149027109	6
10 2000 40 500	7
0.50000	8
0.19870	9
-0.39740 0.59610	10
0.39740 0.28100	11
1.0E+10 200.0 1.00000	12

Figure 98. Complete TDMC data file for surface step restart.

APPENDIX D

TIME-DEPENDENT MONTE CARLO ALGORITHM FOR III-V (110) SURFACES

Calculation of Migration Barriers for a Substrate with Unequal Binding Energies

For the square monatomic lattice, there exists a single binding energy which governs the motion of an isolated adatom. In the case of the III-V semiconductors, there are two different atomic binding sites with different binding energies. Thus we need to develop energy expressions which account for this. The two cases of interest are treated below. The first situation describes the migration of an isolated adatom on a surface with two distinct binding sites. The second case depicts the migration barrier for an arbitrary adatom which may or may not be surrounded by neighboring adatoms. E_1^0 is the binding energy for one substrate atom type. E_2^0 is the binding energy for the other substrate atom type. V_1 and V_2 are the potential wells on two adjacent binding sites. X' is the position of the minimum in V_1 . $X'+1$ is the minimum of V_2 . L is the lattice spacing and k is the force constant. X_c is the crossing point between the two potential energy wells which determines the activation barrier for migration.

In case 1, we have:

$$V_1(x) = E_1^0 + k(x - x')^2$$

$$V_2(x) = E_2^0 + k[x - (x' + 1)]^2$$

In case 2, we have:

$$V_1(x) = E_1 + k(x - x')^2$$

$$V_2(x) = E_2 + k[x - (x' + 1)]^2$$

Migration Barrier for an Isolated Adatom

In the first case, we need only consider the binding energies E_1^0 and E_2^0 for an isolated adatom. We need to find the crossing point x_c where $V_1(x_c) = V_2(x_c)$. The appropriate equality is:

$$E_1^0 + k(x_c - x')^2 = E_2^0 + k[(x_c - (x' - 1))]^2$$

Solving for the energy difference, $E_1^0 - E_2^0$, we get

$$E_1^0 - E_2^0 = k(-2x_c l + 2x' l + l^2) = \Delta E_m^0$$

$$\Delta E_m^0 / 2k = x_c l - x' l - l^2 / 2$$

$$\Delta E_m^0 / 2kl = (x_c - x' - l/2)$$

$$x_c = x' + l/2 + \Delta E_m^0 / 2kl$$

$$V_1(x_c) = E_1^0 + k[x' + l/2 + \Delta E_m^0 / 2kl - x']^2$$

$$E_m^0 = V_1(x_c) - V_1(x')$$

$$E_m^0 = E_1^0 + k[l^2/4 + \Delta E_m^0 / 2kl$$

$$+ (\Delta E_m^0)^2 / 4k^2 l^2] - E_1^0 - k(x' - x')^2$$

$$E_m^0 = k[l/2 + \Delta E_m^0 / 2kl]^2$$

Next, we need to solve for E_2 such that $V_1(x'+1) = V_2(x'+1)$.

This prevents the activation barrier from passing through a minimum and then increasing again.

$$E_1 = k[(x'+1) - x']^2 = E_2 + k[(x'+1) - (x'+1)]$$

$$E_2 = E_1 + kl^2$$

$$\Delta E_m = E_2 - E_1 = kl^2$$

Migration Barrier for Any Adatom

Here we need to find the crossing point x_c where $V_1(x_c) = V_2(x_c)$. This is the general case for two distinct binding energies and adparticles surrounded by neighboring adatoms.

$$x_c = 1/2l[\Delta E_m/k + 2x'l + l^2]$$

$$E_m = k[\Delta E_m/2lk + 1/2]^2$$

$$E_m = [\Delta E_m^2/4 + \Delta E_m + \Delta E_m/2 + \Delta E_m/4] = E_m = kl^2$$

$$E_m = E_2 - E_1$$

Time-Dependent Monte Carlo Algorithm for Bi/GaAs (110)
and Bi/InSb (110)

This section provides a brief description of the FORTRAN subroutines involved in the diffusion simulation for bismuth on the GaAs (110) surface. Instructions for starting and continuing a run are also provided in Appendix D. A representative output file is also included there. The main program in this case is called GAAS.

The series of logical statements used to rule out the impossible transition types are contained in PROGRAM LONGTYPE. The transition type is a function of the binding site and three sets of non-equivalent neighbors. Consequently, six indices are used to uniquely define each of the initial and final neighborhoods. The transition type is further characterized by an index that represents the initial and final binding sites. There are four possible combinations: Ga-Ga, As-As, Ga-As, and As-Ga.

The simulation begins with the random deposition of particles on a perfect hexagonal substrate in SUBROUTINE HEXMAP. The simulation is completely designed for a perfect hexagonal lattice. The distortions that represent the diffusion behavior on the III-V semiconductor (110) surface are completely introduced through the interaction energies. For Bi on GaAs, there is a limit of ten atoms on the length of the adatom chains. In reality, the chain length and

required pattern of vacancies can only be produced by limiting the chain length to 5 consecutive atoms on Ga or As binding sites. Consequently, each time a set of particle coordinates is generated, SUBROUTINE OCHAIN is called. The chain length limit is imposed in this fashion. If the chain will be too long, another set of coordinates is generated. This process continues until the desired coverage is deposited. The number of particles, the dimensions of the initially occupied region, and the random number seed must be supplied to HEXMAP. Once again, we can also deposit particles in any desired initial configuration.

The sets of six indices which were produced to identify transition types are read from a file by SUBROUTINE MIXTYPE. The six indices describe 27 unique particle neighborhoods on the (110) surface. It is also necessary to know the initial-final binding site combination. Thus, a four dimensional array of allowed transition types is created by eliminating the physically meaningless combinations of these 27 neighborhoods and discriminating between binding sites. The allowed transitions are placed in common memory.

The energies for allowed transition types are computed in four separate algorithms which are characterized by the the initial-final binding site combination. These four algorithms are called ENGAGA, ENGAAS, ENASGA, and ENASAS. The binding energies, the three distinct neighboring adparticle interactions, and the temperature must be provided. The error

in the energy window approximation is reduced here, as before, by successively reducing the width of the occupied energy lists. Each energy algorithm places transition types on 50 lists. The final energy window width, or error, is written to the output file. Each allowed transition type is also assigned to the appropriate energy list. In addition, we create two additional energy lists that are necessary to allow us to limit the bismuth chain length. Thus, we are left with 202 energy lists.

Transition types for the initial and migrating systems are produced in SUBROUTINE TRANSITION. The lattice dimensions and the route are required, as before. The binding site of the particle of interest is first determined. Then the numbers of each of three neighbor types are determined. The identity of the final binding site and the neighbor numbers in the final configuration are then determined. Chain length constraints must also be imposed here. Before a particle is assigned an allowed transition, we call SUBROUTINE CHAIN. We count the chain length for the potential move to insure that there won't be more than five consecutive atoms on Ga or As sites. If the chain will be longer than this, the transition is forbidden. This constant, but necessary, counting certainly significantly increases the computation time for this algorithm. All of the potential particle hops are recorded in the PART matrix. The occupation number of each transition type is tabulated for use in the program timer.

Finally, particle numbers and coordinates are placed on the appropriate energy lists. Each particle can be on as many as six lists.

The probability associated with each list of potential transitions is computed from the Boltzmann factors produced in the energy routines. SUBROUTINE EPROB either computes probabilities for all 202 energy lists or just for those lists affected by a move. Particles are randomly selected, according to their occurrence probabilities, from the lists for migration. As before, two seeds are required for the random number generators in SUBROUTINE RANEN. The selected energy list and the position of the migrating particle on that list are passed back to the calling program.

SUBROUTINE MOVE directs the particle to its final destination and determines the direction of motion. The appropriate trajectory length is also increased by one. There are twelve possible trajectories in this system. There are six trajectories for the distorted hexagon and two unique binding sites. The trajectories are different for hops from different binding sites. These "raw" trajectories are printed in the output file. They can be multiplied by appropriate distances to produce an estimate of the diffusion coefficient for a system of interest. A list of ten affected particles is also generated in MOVE.

SUBROUTINE CLOCK computes the time for each hop and the total time for the simulation. The total number of particles,

the number of the selected energy list, the particle number, the direction of motion, the rate constant, and the probabilities must be provided to CLOCK. New transition types, energy lists, and probabilities are determined for all affected particles. The affected energy lists are recorded by SUBROUTINES ADD and CLEAR and corrected after each move.

Because the LEED pattern for this system is proposed to arise from 10 atom chains of bismuth along the [1-10] direction, the number of these chains and the average chain length are monitored throughout the calculation. SUBROUTINE CHCOUNT has been designed to scan the lattice periodically and count the 10 atom chains. The routine actually counts the lengths of all chains on the lattice and stores the data in a matrix. In the current version of the code, only the number of 10 atom chains and the average chain length are printed. The frequency of other chain lengths can easily be printed also, if desired.

At specified intervals throughout the simulation, time evolution data is recorded. SUBROUTINES RMS and RMST compute the displacements for the particles on the GaAs substrate. The number of particles must be provided by the main program. The dimensions of the GaAs (110) unit cell appear explicitly in these routines. Consequently, the displacements are specific to this system and not in "lattice units" as before.

This algorithm is also different in that the data are saved at equal time intervals, rather than after a specified

number of hops. In addition, the system can be heated at some predetermined rate. Time and temperature increments are specified. The temperature starts at some initial value. When the specified length of time has passed, the temperature is increased. New energies, Boltzmann factors, and probabilities are computed for the next time interval. Thus, we can save data at any desired time or temperature. This scheme should provide an additional method of monitoring the effects of temperature on overlayer ordering. Of course the simulation can also be run at constant temperature.

The diffusion simulator for Bi/InSb is entirely analogous to the one described here for Bi/GaAs. The only differences are that there is no limit on the Bi chain length, there is no temperature ramp, and the dimensions of the InSb unit cell are used to calculate displacements. The InSb simulation is based on the number of hops, not the time. The procedures for data and command file creation are quite analogous to those described for Bi/GaAs. This algorithm could easily be modified to include a temperature ramp and a mechanism for diffusion with desorption.

Running the Diffusion Simulation for Bi/GaAs

The time-dependent Monte Carlo diffusion simulation for bismuth on gallium arsenide is executed using a data file with the desired diffusion parameters and a command file

to direct file management. The following is a list of the parameters as they are named in the FORTRAN algorithm. Data are saved at specified time intervals throughout the calculation. Therefore, the simulation can be restarted where the previous run finished. Several parameters are needed for the restart in this case. Data types are indicated in parentheses.

DATA	data file name (CHARACTER*60)
OUTPUT	output file name (CHARACTER*60)
TDMC	number of times ET lists are rewritten (INTEGER)
ITDMC	number of hops before rewriting ET lists (INTEGER)
TTIME	total simulation time (REAL*16)
DTIME	time interval between data saves (REAL*16)
PTIME	time interval for temperature increase (REAL*16)
TINC	temperature increment (REAL)
LENGTH1	initial X lattice length (INTEGER)
LENGTH2	initial Y lattice length (INTEGER)
THETA	initial coverage (REAL)
LENGTH3	final X lattice length (INTEGER)
LENGTH4	final Y lattice length (INTEGER)
SEED	map seed (INTEGER)
SEED1	energy seed (INTEGER)
SEED2	position seed (INTEGER)
EMGA	migration barrier for gallium sites (REAL*16)
E1GA	interaction energy for Ga neighbor 1 (REAL*16)
E2GA	interaction energy for Ga neighbor 2 (REAL*16)
E3GA	interaction energy for Ga neighbor 3 (REAL*16)
EMAS	migration barrier for arsenic sites (REAL*16)
E1AS	interaction energy for As neighbor 1 (REAL*16)
E2AS	interaction energy for As neighbor 2 (REAL*16)
E3AS	interaction energy for As neighbor 3 (REAL*16)
KM	rate constant (REAL)
TEMP	temperature (REAL)
CTDMC	final value of TDMC from previous run (INTEGER)
TDMC	total allowed number of hops (INTEGER)
PC	current value of ITDMC from previous run (INTEGER)
CTIME	current time (REAL*16)
TTIME	new total time (REAL*16)
CTEMP	current temperature (REAL)

TTIME, the real time from the simulation timer, determines the duration of the run. Because it may be difficult to predict the time for the diffusion process, TDMC is used to restrict the total number hops. The simulation will stop after $TDMC \cdot ITDMC$ hops even if the time designated by TTIME has not passed. TDMC also governs the frequency of energy list reinitialization in the previously described fashion. There are $ITDMC$ hops before the lists are reinitialized. The lists are reinitialized as many as TDMC times during the calculation. The total number of hops may no longer be the product $ITDMC \cdot TDMC$ because the time determines the stopping point. The final number of hops will be some random integer.

The algorithm currently assumes a "rectangular" diffusion system. The initial particle pack can still be deposited over a region of the lattice with smaller dimensions than those of the final lattice. LENGTH1 is the initial occupied length in the X direction. LENGTH2 is the initial length in the Y direction. LENGTH3 is the final length in the X direction. LENGTH4 is the final length in the Y direction. All lengths can range from 0 to 199. The simulation is based on binding sites, not actual distances. The X direction is the [1-10] direction. The Y direction is the [001] direction. Consequently, because the unit cell length is longer along the [001] direction, it may be desirable to make LENGTH2 and LENGTH4 smaller than LENGTH1

and LENGTH3. This will make the actual lattice dimensions more realistic. For instance, the LEED experiment covers roughly 200 X 200 angstroms. To model these dimensions, we give LENGTH1 a value of 49 and LENGTH2 a value of 35.

Although the lattice length can extend over 200 sites, the other dimensions of the program are not sufficient to accommodate a high coverage over a large lattice. THETA is the initial coverage. Thus, LENGTH1 times LENGTH2 times THETA should give the total number of particles. The algorithm can currently accommodate 4,000 particles.

The PART matrix containing particle coordinates is saved at intervals of DTIME, where DTIME is some fraction of TTIME. Displacements are also saved at intervals of DTIME. The quotient TTIME/DTIME cannot exceed 20. In other words, these data can be saved 20 times during the simulation. The PART matrix is saved as many times as the other time evolution data in this case. PTIME is the time interval between temperature jumps. When PTIME is 0.0, the simulation is run at constant temperature. TINC is the temperature jump per time interval. TINC/PTIME gives the heating rate.

SEED, SEED1, and SEED2 are large integers for the random number generators for the map, the energy list, and the position of the particle on the list.

EMGA is the binding energy of an isolated adatom on a Ga site. EMAS is the binding energy of an isolated adatom on an As site. These quantities are in units of RT where R is the

ideal gas constant in kilocalories and T is the temperature in Kelvin. $E1GA$, $E2GA$, and $E3GA$ are the interaction energies for sites surrounding Ga sites. These sites are numbered according to the scheme presented in the section on the model for this system. $E1AS$, $E2AS$, and $E3AS$ are the interaction energies for sites surrounding As sites. $TEMP$ is the initial temperature in Kelvin.

KM is the rate constant in inverse seconds. It is given a value of $1.0 \cdot 10^{10}$ in our investigations. The jump length for each hop has been incorporated into the algorithm. Consequently, the displacements produced by the algorithm are specifically computed for diffusion on GaAs (110). $CTEMP$ is the current temperature.

Several parameters are required to restart the calculation. $CTDMC$, the current cycle number, for the restart is the current value of $TDMC$ from the preceding run. Because the time dictates the number of hops, $TDMC$ will be some random integer. PC is the number of hops that were carried out within the smaller loop. PC is the current value of $ITDMC$ from the previous run. $TDMC$ still determines the upper limit on the number of hops. $CTIME$ is the time when the $PART$ matrix was last saved. The calculation will resume from this point. All of the $PART$ data for a run is written into the same file. Consequently, to restart a run, this file must be edited; and the initial and current $PART$ matrices must be placed into separate files. The values of $CTDMC$ and PC and

the time are written into the PART file. The current values of the seeds for energy type selection are written into the output file (unit 6) adjacent to the time. TTIME is the new total time for the simulation.

The figures below illustrate data file creation. To start a new run, a file consisting of ten lines is needed. The position of each of the necessary input parameters described above are illustrated here. To restart a run, another data file consisting of ten lines with the appropriate current parameters is needed. The actual FORTRAN, including formats, to read the data files is also given below. FORTRAN FORMATS TO BEGIN A RUN:

```

      READ(5,1), DATA
      READ(5,1), OUTPUT
1     FORMAT(A60)
      READ(5,2), TDMC,ITDMC
2     FORMAT(I10,I10)
      READ(5,3) TTIME,DTIME,PTIME,TINC
3     FORMAT(3E15.6,F10.4)
      READ(5,4), LENGTH1,LENGTH2,THETA
      READ(5,4), LENGTH3,LENGTH4
4     FORMAT(2I5,F10.5)
      READ(5,5), SEED,SEED1,SEED2
5     FORMAT(3I15)
      READ(5,6), EMGA,E1GA,E2GA,E3GA
      READ(5,6), EMAS,E1AS,E2AS,E3AS
6     FORMAT(4F10.6)
      READ(5,7), KM,ITEMP
7     FORMAT(E10.5,F10.4)
      READ(5,1), DATA
      READ(5,1), OUTPUT

```

FORTRAN FORMATS TO RESTART A RUN:

```

      READ(5,1), DATA
      READ(5,1), OUTPUT
1     FORMAT(A60)
      READ(5,2), CTDMC,TDMC,ITDMC,PC
2     FORMAT(4I10)
      READ(5,3) CTIME,TTIME,DTIME,PTIME,TINC

```

```

3      FORMAT(4E15.6,F10.4)
      READ(5,4), LENGTH1,LENGTH2,THETA
      READ(5,4), LENGTH3,LENGTH4
4      FORMAT(2I5,F10.5)
      READ(5,5), SEED,SEED1,SEED2
5      FORMAT(3I15)
      READ(5,6), EMGA,E1GA,E2GA,E3GA
      READ(5,6), EMAS,E1AS,E2AS,E3AS
6      FORMAT(4F10.6)
      READ(5,7), KM,CTEMP
7      FORMAT(E10.5,F10.4)

```

The following are examples of the command files for initial and restart runs. Root mean square displacements are written to unit 11. Raw trajectories are written to unit 12. Particle coordinates are written to unit 13. To restart a run, the initial and current matrices of particle coordinates must be re-read. These matrices, which are obtained by editing and separating them from the PART file, are assigned units 2 and 20, respectively. Units 3 and 4 are used to initialize the transition type array.

COMMAND FILE TO BEGIN A RUN:

```

$ ASSIGN [UCHEHAB.GAAS]NNTYPE.OUT FOR003
$ ASSIGN [UCHEHAB.GAAS]TYPEGAAS.OUT FOR004
$ ASSIGN [UCHEHAB.RGAAS]GAAS28.DAT FOR005
$ ASSIGN [UCHEHAB.RGAAS]GAAS28.OUT FOR006
$ ASSIGN [UCHEHAB.RGAAS]RMST28.DAT FOR011
$ ASSIGN [UCHEHAB.RGAAS]TRAJ28.DAT FOR012
$ ASSIGN [UCHEHAB.RGAAS]PART28.DAT FOR013
$ RUN [UCHEHAB.GAAS]GAAS

```

COMMAND FILE TO RESTART A RUN:

```
$ ASSIGN [UCHEHAB.RGAAS]PART28I.DAT FOR002
$ ASSIGN [UCHEHAB.GAAS]NNTYPE.OUT FOR003
$ ASSIGN [UCHEHAB.GAAS]TYPEGAAS.OUT FOR004
$ ASSIGN [UCHEHAB.RGAAS]GAAS28R.DAT FOR005
$ ASSIGN [UCHEHAB.RGAAS]GAAS28R.OUT FOR006
$ ASSIGN [UCHEHAB.RGAAS]RMST28R.DAT FOR011
$ ASSIGN [UCHEHAB.RGAAS]TRAJ28R.DAT FOR012
$ ASSIGN [UCHEHAB.RGAAS]PART28R.DAT FOR013
$ ASSIGN [UCHEHAB.RGAAS]PART28C.DAT FOR020
$ RUN [UCHEHAB.GAAS]GAASR
```

In addition, all of the input parameters and all of the output, with the exception of the data in the PART file, are written to unit 6.

The PART file in this case is composed of 14 columns. Each line contains all pertinent data for the particle who's number corresponds to that line of the array. The first and second columns give the particle's current X and Y coordinates, respectively. Columns 3 through 14 contain the six prospective transitions for the particle and the position of the particle on each energy list. The sequence goes energy list, position on list, energy list, position on list, et cetera.

The following command files can be used to compile and link the program. GAAS is the main program for beginning a run. GAASR is the restart version of the main program. The diffusion data are saved at equal time intervals.

COMMAND FILE TO COMPILE GAAS ALGORITHM

```
$ FOR ADD6  
$ FOR CLEAR6  
$ FOR CLOCK  
$ FOR COUNT  
$ FOR GAAS  
$ FOR ENASAS  
$ FOR ENASGA  
$ FOR ENGAAS  
$ FOR ENGAGA  
$ FOR HEXMAP  
$ FOR MIXTYPE  
$ FOR MOVEHEX  
$ FOR PROB  
$ FOR RANEN  
$ FOR RMSHEX  
$ FOR RMSTHEX  
$ FOR SCHAIN  
$ FOR SOCHAIN  
$ FOR TRANS
```

COMMAND FILE TO LINK GAAS ALGORITHM

```
$ LINK/EXE=GAAS GAAS,COUNT,ADD6,CLEAR6,CLOCK,SCHAIN,-  
HEXMAP,MIXTYPE,MOVEHEX,PROB,RANEN,RMSHEX,RMSTHEX,TRANS,-  
ENGAGA,ENGAAS,ENASGA,ENASAS,SOCHAIN
```

1234567890123456789012345678901234567890123456789012345										COLUMN
										LINE
GAAS28.DAT ← DATA FILE NAME										1
GAAS28.OUT ← OUTPUT FILE NAME										2
										3
50000		100								4
↑		↑								
TDMC		ITDMC								5
300.00000E-09		300.00000E-09		30.00000E-09		0.0		6		
↑		↑		↑		↑				
TTIME		DTIME		PTIME		TINC				
49		35		0.8000						7
↑		↑		↙						
LENGTH1		LENGTH2		THETA						
49		35								8
↑		↑								
LENGTH3		LENGTH4								
1586441319		1516781432		1599547523						9
↙		↑		↑						
SEED		SEED1		SEED2						
0.1987		5.0000		4.00000		3.0000				10
↑		↑		↑		↑				
EMGA		E1GA		E2GA		E3GA				
0.2478		3.0000		4.00000		5.0000				11
↑		↑		↑		↑				
EMAS		E1AS		E2AS		E3AS				
1.0E+10		450.0								12
↙		↑								
KM		TEMP								

Figure 99. Time-dependent Monte Carlo data file breakdown for Bi/GaAs.

123456789012345678901234567890123456789012345	COLUMN LINE
GAAS28R.DAT ← DATA FILE NAME	1
GAAS28R.OUT ← OUTPUT FILE NAME	2
3827 10000 100 43 ↑ ↑ ↑ ↑ CTDMC TDMC ITDMC PC	3
312.96000E-09 600.00000E-09 600.00000E-09 30.00000E-09 0.0 ↑ ↑ ↑ ↑ ↑ CTIME TTIME DTIME PTIME TINC	5
49 35 0.8000 ↑ ↑ ↙ LENGTH1 LENGTH2 THETA	6
49 35 ↑ ↑ LENGTH3 LENGTH4	7
SEED 0 -37933409 352000262 ↗ ↑ ↑ ↑ SEED1 SEED2	8
0.1987 5.0000 4.00000 3.0000 ↑ ↑ ↑ ↑ E1GA E1GA E2GA E3GA	9
0.2478 3.0000 4.00000 5.0000 ↑ ↑ ↑ ↑ E1AS E1AS E2AS E3AS	10
1.0E+10 450.0 ↗ ↑ KM TEMP	

Figure 100. Time-dependent Monte Carlo data file breakdown for Bi/GaAs restart.

123456789012345678901234567890123456789012345										COLUMN LINE
GAAS28.DAT										1
GAAS28.OUT										2
50000 100										3
300.00000E-09 300.00000E-09 30.00000E-09 0.0										4
49 35 0.8000										5
49 35										6
1586441319 1516781432 1599547523										7
0.1987 5.0000 4.00000 3.0000										8
0.2478 3.0000 4.00000 5.0000										9
1.0E+10 450.0										10

Figure 101. Complete data file for Bi/GaAs.

123456789012345678901234567890123456789012345										COLUMN LINE
GAAS28R.DAT										1
GAAS28R.OUT										2
3827 10000 100 43										3
312.96000E-09 600.00000E-09 600.00000E-09 30.00000E-09 0.0										4
49 35 0.8000										5
49 35										6
0 -37933409 352000262										7
0.1987 5.0000 4.00000 3.0000										8
0.2478 3.0000 4.00000 5.0000										9
1.0E+10 450.0										10

Figure 102. Complete data file for Bi/GaAs restart.

 