Low Energy Ion Beam Assisted Growth of Metal Multilayers

A Thesis Presented to
the faculty of the School of Engineering and Applied Science
University of Virginia

In Partial Fulfillment of
the requirements for the Degree
Doctor of Philosophy (Engineering Physics)

By

Junjie Quan
01/2007
UMI Number: 3239975

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI

UMI Microform 3239975
Copyright 2007 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
APPROVAL SHEET

The thesis is submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Engineering Physics)

[A Signature]

AUTHOR

This thesis has been read and approved by the examining Committee:

[H. Wadley]

Haydn N. G. Wadley (Thesis advisor)

[Leonid V. Zhigilei]

Leonid V. Zhigilei (Chairperson)

[J. Ruvalds]

John Ruvalds (Minor-Physics)

[R. Hull]

Robert Hull

[Stuart A. Wolf]

Stuart A. Wolf

Accepted for the School of Engineering and Applied Science

[J. H. Aylor]

James H. Aylor
Dean, School of Engineering and Applied Science

01/2007
# TABLE OF CONTENT

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENT</td>
<td>I</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>1</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>1</td>
</tr>
<tr>
<td>FIGURE CAPTIONS</td>
<td>I</td>
</tr>
<tr>
<td>CHAPTER 1: INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Goals of the Dissertation</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Structure of the Dissertation</td>
<td>4</td>
</tr>
<tr>
<td>CHAPTER 2: INTRODUCTION TO GIANT MAGNETORESISTANCE</td>
<td>6</td>
</tr>
<tr>
<td>2.1. Magnetoresistance</td>
<td>7</td>
</tr>
<tr>
<td>2.2. GMR Mechanisms</td>
<td>11</td>
</tr>
<tr>
<td>2.3. GMR Applications</td>
<td>16</td>
</tr>
<tr>
<td>2.3.1 Magnetic Read Head Sensors</td>
<td>16</td>
</tr>
<tr>
<td>2.3.2 Nonvolatile Magnetic Memory</td>
<td>18</td>
</tr>
<tr>
<td>2.4 Growth of GMR Films</td>
<td>20</td>
</tr>
<tr>
<td>2.4.1 Molecular Beam Epitaxy</td>
<td>20</td>
</tr>
<tr>
<td>2.4.2 Sputtering</td>
<td>21</td>
</tr>
<tr>
<td>2.4.3 Ion Beam Deposition</td>
<td>23</td>
</tr>
<tr>
<td>2.5 Desired GMR Structures</td>
<td>24</td>
</tr>
<tr>
<td>2.6 Improved Deposition Approaches</td>
<td>27</td>
</tr>
<tr>
<td>CHAPTER 3: ATOMICISTIC SIMULATION METHODS</td>
<td>29</td>
</tr>
<tr>
<td>3.1. Molecular Dynamics</td>
<td>31</td>
</tr>
<tr>
<td>3.2. Interatomic Potentials</td>
<td>34</td>
</tr>
<tr>
<td>3.3. Molecular Statics Calculations</td>
<td>37</td>
</tr>
</tbody>
</table>
6.4.2 Modulated Energy Ion Assistance ............................................. 110
6.4.3 Sequential Ion Assistance ....................................................... 112
6.5 Discussion ............................................................................. 115
6.6 Conclusion ........................................................................... 120

CHAPTER 7. BIASED TARGET ION BEAM DEPOSITION .................. 121
7.1. Insights from Simulations ....................................................... 121
7.2. Problems of Conventional Ion Beam Deposition .................... 126
7.3. Biased Target Ion Beam Deposition .......................................... 129
7.4. Ion Source ........................................................................... 132

CHAPTER 8. LOW ENERGY ION BEAM ASSISTED DEPOSITION OF A GMR SPIN VALVE .................................................................................. 134
8.1 Introduction ............................................................................ 135
8.2 Experimental Depositions ......................................................... 136
8.3 Results ................................................................................... 139
8.3.1 Unassisted Deposition ........................................................... 139
8.3.2 Simultaneous Assistance ...................................................... 142
8.3.3 Other Depositions ................................................................. 144
8.4 Discussion ............................................................................. 146
8.5 Conclusion ........................................................................... 147

CHAPTER 9. LOW ENERGY ION ASSISTED DEPOSITION OF Ta/Cu FILMS .................................................................................. 148
9.1 Introduction ............................................................................ 149
9.2 MD Simulations of Ta/Cu Vapor Deposition ......................... 152
9.2.1 Simulation Method ................................................................. 152
9.2.2 Results ............................................................................... 153
9.3 Experimental Growth Approach ............................................. 159
9.4 Experimental Results ............................................................... 161
9.4.1 Simultaneous Ion Assistance ................................................. 161
9.4.2 Modulated ion energy and sequential ion beam assistance methods .............................................. 168
ACKNOWLEDGMENTS

The completion of this dissertation would not have been possible without the encouragement and help of a number of people. I wish to thank my advisor, Professor Haydn N. G. Wadley, for his earnest and patient advice and guidance throughout this research program. I am grateful to Dr. Xiaowang Zhou for his guidance on simulation techniques and scientific writing. Above all, they served as a wonderful source of knowledge and experience that helped to shape the course of my work. I greatly appreciate all the members from 4Wave Inc., especially Drs. T. Hylton, S. Antrazi, and D. Baldwin for their generous help in experiments. I am grateful to Dr. D. Mauri from Hitachi Global Storage Technology and Dr. W. Egelhoff from the National Institute of Standards and Technology (NIST) for the discussions about experiments and help with characterizations. I also want to thank Dr. R. Hull and his student L. He for their generous help with TEM characterization. I especially appreciate the support of all the members of the University of Virginia Intelligent Processing Materials Laboratory (IPML). I would like to thank Joanna Murdick for her help on proofreading my dissertation. I also wish to thank the Department of Engineering Physics for creating an ideal environment to pursue my doctoral degree. I am grateful to the Defense Advanced Research Projects Agency and Office of Naval Research for support of my work (under ONR grant N00014-03-C-0288 monitored by Drs. Carey Schwartz and Julie Christodoulou). Finally, my greatest thank you goes to all members of my family for their understanding, motivation, and encouragement.
ABSTRACT

Vapor deposited metal multilayers have attracted a great deal of interest in recent years because they offer extraordinary strength, hardness, heat resistance, and unexpected new properties like high reflectivity and spin-dependent conductivity. The giant magnetoresistance effects discovered in Fe/Cr artificial superstructures in 1988 stimulated a large number of studies on the electronic transport properties of spintronic materials because of their important applications in highly sensitive magnetic sensors, nonvolatile random access memories, and the data storage industry in general. Magnetic multilayers allow exploitation of unique micromagnetic, magnetooptic, and magnetoelectronic phenomena that cannot be realized using conventional materials. For example, if ferromagnetic layers (such as CoFe) with a thicknesses of 5-7 nm are separated by a non-magnetic spacer (such as Cu or AlOx) of an appropriate thickness (1-3 nm), they can exhibit large changes in their electrical resistance when a magnetic field is applied. These changes are caused mainly by spin-dependent conduction electron scattering at magnetic multilayer interfaces. Many experimental and theoretical works have sought to promote a basic understanding of the effect of atomic structure in thin film multilayers upon spin dependent transport. It has been found that interfacial imperfections, such as interfacial roughness and interlayer mixing, dramatically reduce the properties exploited for spintronic applications.

A combination of computer modeling and experiments has been used to discover more effective ways to control the interfacial structures of metal multilayers. Earlier atomic simulations had indicated that it is very important to control adatom energy during
deposition in order to improve interface properties. Based on these ideas, this dissertation has investigated the effects of low energy ion assistance during metal multilayer deposition. Using molecular dynamics modeling, the effects of ion energy, ion species, ion fluence, and ion incidence angle has been investigated during low energy ion assisted vapor deposition of Cu/Co multilayers. Key mechanisms of atomic reconstruction in the low energy regime have been identified and investigated in detail using a molecular statics method. By changing the ion energy or modifying the time lag between metal and ion deposition, these simulations identified three different approaches for controlling the atomic assembly of multilayer: (i) simultaneous, (ii) modulated energy and (iii) sequential ion assistance. Each has been shown capable of enabling significant interfacial structure control during the growth of metallic multilayers.

A biased target ion beam deposition (BTIBD) system was designed and constructed to provide growth environments similar to those that resulted in the best simulated film structures. The BTIBD system has been successfully employed to deposit a variety of multilayers. Experimental investigations of low energy ion assisted growth of a representative spin valve (Ta/NiFe/Co/Cu/Co/FeMn/Cu) multilayer and a model Ta/Cu film indicate significantly improved interfacial structures when oblique, low energy, argon ion assistance was used. These results are fully consistent with the simulations and confirm that a BTIBD approach to multilayer deposition results in substantially improved films compared to those synthesized by traditional ion beam assisted deposition approaches where less effective control of atomic self assembly is possible.
FIGURE CAPTIONS

Figure 2.1 (a) Anisotropic magnetoresistance (AMR), and (b) application of an AMR sensor [59]. Normally, the resistivity of AMR material is smaller when the magnetization is perpendicular to the electric current direction as shown in (a). ...... 8

Figure 2.2 Resistivity of Fe/Cr multilayers at 4.2K as a function of the magnetic field [1]. ......................................................................................................................................................... 9

Figure 2.3 Electron transports in a multilayer structure. The white arrows show the magnetic moment of the ferromagnetic layers. Extra electron scattering occurs when the moments are antiparallel. Only interface scattering is shown in the diagram. ... 12

Figure 2.4 Room temperature magnetoresistance vs. Cu spacer layer thickness for a series of Co/Cu multilayers [61]. .................................................................................................................. 14

Figure 2.5 Schematic drawing of typical magnetic recording disk drive and spin-valve read head [92]. ......................................................................................................................... 17

Figure 2.6 Schematic representation of giant magnetoresistive spin valve structure. ...... 18

Figure 2.7 Cartoon of cross-point MTJ MRAM architecture: Magnetic tunnel junction storage cells are located cross-point of the word and bit lines. The (a) and (b) cartoons show the processes of reading and program modes [92]................................. 19

Figure 2.8 Schematic illustration of a planar magnetron sputtering system [114]......... 22

Figure 2.9 A schematic illustration of RF diode sputter deposition system. ................. 23

Figure 2.10 Schematic illustration of an ion beam deposition system with ion beam assistance....................................................................................................................................... 24
<table>
<thead>
<tr>
<th>Figure Caption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.11 An illustration of interface roughness effects on magnetic coupling. .......... 26</td>
</tr>
<tr>
<td>Figure 2.12 An illustration of interface mixing effects. ........................................... 27</td>
</tr>
<tr>
<td>Figure 3.1 Multiscale modeling tools for physical vapor deposition [130]. ...................... 30</td>
</tr>
<tr>
<td>Figure 4.1 (a) Atomic reassembly of a surface under kinetically constrained conditions showing terrace/asperity formation and shadowing effects and (b) example of atomic reassembly paths on a surface and vacancy annihilation in the subsurface lattice atoms. &quot;A&quot; jump corresponds to an &quot;on terrace&quot; hop, &quot;B&quot; is a jump &quot;from a ledge of terrace,&quot; &quot;C&quot; is a jump &quot;from a dimer,&quot; &quot;D&quot; is a jump &quot;over a terrace ledge,&quot; and &quot;E&quot; is a &quot;vacancy annihilation&quot; jump [43] ........................................... 42</td>
</tr>
<tr>
<td>Figure 4.2 A model rough surface structure: Co cluster on Cu film surface ....................... 49</td>
</tr>
<tr>
<td>Figure 4.3 Surface morphology after 1500 Xe$^+$ ion impacts. (a) Xe$^+$ energy 2 eV; (b) Xe$^+$ energy 4 eV; (c) Xe$^+$ energy 6 eV; and (d) Xe$^+$ energy 10 eV. ........................................... 51</td>
</tr>
<tr>
<td>Figure 4.4 Energy and species effects on (a) interfacial roughness parameter (fraction of atoms remaining above the first layer after ion bombardment,) and (b) interfacial mixing parameter (fraction of mixing atoms after ion bombardment). Solid lines are fitted curves .................................................................................................................. 53</td>
</tr>
<tr>
<td>Figure 4.5 Recoil atomic assembly mechanism at an ion energy of 1 eV. (a) Original configuration at time $t = 0$ ps, (b) $t = 0.4$ ps, (c) $t = 0.5$ ps, and (d) $t = 0.8$ ps........... 55</td>
</tr>
<tr>
<td>Figure 4.6 Exchange flattening mechanism during an 8 eV Ar ion impact. Selected atoms are marked with an open circle, square, and diamond to show their motion.............. 56</td>
</tr>
</tbody>
</table>
Figure 4.7 Flattening mechanisms during a 12 eV ion impact. Flattening occurred by the exchange mechanism for the "diamond" atom and the direct hop mechanism for the "circle" atom. ................................................................. 58

Figure 4.8 Flattening mechanisms during a 20 eV ion impact. All three top layer atoms are flattened by the direct hop flattening mechanism. An exchange mechanism is also seen for the "triangle" atom in the second layer ........................................... 59

Figure 4.9 Comparison of intermixing mechanism at different ion impact sites. (a) On the edge of a terrace, (b) one row away from the edge, (c) two rows away from the edge, (d) four rows away from the edge, and (e) on the infinity terrace. Energies for the onset of intermixing for each case are used ........................................... 63

Figure 4.10 Comparison of simulated Cu/Co/Cu trilayer structures deposited without and with ion assistance. (a) Deposition without ion assistance; (b) 6 eV Ar ion assistance. .................................................................................................................. 68

Figure 5.1 A model (111) Cu crystal surface partially coated with twelve, 10-atom Co clusters arranged in the form of three monolayer high pyramids has been used to investigate the effect of ion fluence and incident angle upon the atomic reassembly processes. An incidence angle, θ, is defined as the deviation from the \([\overline{1} \overline{1} T \overline{1}]\) direction in the \((\overline{1} \overline{1} 0)\) plane........................................................................................................ 75

Figure 5.2 Atomic configurations of a Co-on-Cu surface following 10 eV Ar ion impacts at an ion incident angle of zero degrees. (a) Ion fluence of 0.01 ions/\(\AA^2\); (b) ion fluence of 0.1 ions/\(\AA^2\); (c) ion fluence of 0.2 ions/\(\AA^2\); and (d) ion fluence of 1.0...
Figure Captions

ions/Å². Cut off distance of inert ion disappearance was increased in (d) in order to show high fluence of assisting ions. ................................................................. 77

Figure 5.3 (a) Surface roughness parameter as a function of ion fluence using different ion species at a fixed ion energy of 3.0 eV and a fixed ion incident angle of zero degree, and (b) intermixing parameter at a fixed ion incident energy of 15.0 eV and an ion incident angle of zero degree. Solid lines in the figure are the predictions of a two-body collision model. ................................................................. 79

Figure 5.4 (a) Surface flattening and (b) interfacial mixing of critical ion energies as a function of ion incident angle for different inert gas ion masses. Solid lines in the figure are the predictions of a two-body collision model. ......................................................... 85

Figure 5.5 Time-resolved images of an 8 eV Ar ion impact with a Co surface asperity on a (111) Cu surface at different ion incident angles of (a) 10°; (b) 30°; and (c) 60°. 88

Figure 5.6 Illustration of cluster evolution during an 8 eV Ar ion impact. (a) An initial 19 atom cobalt cluster on a (111) copper surface; (b) after an 8 eV argon ion impact at an incident angle of 10°, the cobalt atom A has dropped two atom levels and resides on the copper surface; (c) after an 8 eV argon ion impact at an incident angle of 30°, the cobalt atom A has pushed atom B onto a copper surface by substitution into its original position. ................................................................................... 89

Figure 5.7 Analysis of mixing mechanisms during an 8 eV Ar ion impact. (a) Incident ion angle of 10°; (b) incident ion angle of 30°; and (c) incident ion angle of 50°. Atoms involved in the mixing are shown using smaller balls. Their position is shown at a
sequential time interval of 0.2 ps while the full size atoms are shown in their initial positions. ................................................................................................................................. 91

Figure 5.8 Simulated Cu/Co/Cu multilayer structures deposited using simultaneous argon ion assistance: (a) ion/metal ratio = 1; (b) ion/metal ratio = 5; and (c) ion/metal ratio = 10. The metal atoms had an energy of 0.5 eV and were deposited at an incident adatom angle of 40°. The ion energy was 6 eV and the ion incident angle was 50°. The center part of each multilayer has been removed to better reveal atomic scale structures. ............................................................................................. 95

Figure 6.1 Atomic structures of a Cu/Co/Cu trilayer grown using simultaneous metal deposition and argon ion assistance. The substrate temperature was 300 K and ion/metal flux ratio was unity. Assisting argon ion energies were (a) 0 eV (no ion assistance), (b) 8 eV, and (c) 20 eV. The ions were introduced in the \( [\bar{1} \bar{1} \bar{1}] \) direction. .................................................................................................................. 103

Figure 6.2 (a) RMS roughness and (b) number of cobalt atoms mixed into the underlying copper layer as a function of argon ion energy. The ions were introduced in the \( [\bar{1} \bar{1} \bar{1}] \) direction. .................................................................................................................. 105

Figure 6.3 Effect of ion energy (E) and incident angle on the (a) roughness and (b) mixing indices using a simultaneous ion assistance scheme. The ion angle was defined by its deviation from the \( [\bar{1} \bar{1} \bar{1}] \) direction in the (110) plane. ......................... 106

Figure 6.4 Copper concentration profiles through a Cu/Co/Cu trilayer deposited using a simultaneous ion assistance and metal deposition scheme. Profiles are shown for
films grown without ion assistance (0 eV) and with 8 eV and 20 eV assistance applied perpendicular to the growth surface. ................................................................. 107

Figure 6.5 Cobalt mixing probability as a function of argon ion energy for different depths of the Co-on-Cu interface below the film surface. ........................................... 109

Figure 6.6 Modulated ion energy deposition of Co-on-Cu. (a) First half (five atomic layers) of the cobalt layer was deposited without ion assistance. The remainder of the cobalt layer was deposited with an argon ion energy of (b) 6 eV, (c) 15 eV, and (d) 20 eV. ............................................................................................... 111

Figure 6.7 Growth of a Cu/Co/Cu trilayer using modulated ion energy assistance with an argon ion energy of 20 eV. (a) First half (five atomic layers) of the copper layer was deposited without ion assistance. (b) The remainder of the copper layer was deposited with an argon ion energy of 20 eV. ......................................................... 112

Figure 6.8 Deposition of a Cu/Co/Cu trilayer using sequential ion assistance with an argon ion energy of 30 eV and an ion fluence of 0.5 ions/Å². (a) Structure after completion of first cobalt layer, (b) structure after ion irradiation of the cobalt surface at 30 eV, (c) structure after deposition of the top copper layer, and (d) structure after ion irradiation of the copper surface at 30 eV. ................................. 114

Figure 6.9 Effect of cobalt cluster size on cluster flattening following argon ion impacts at 12 eV and 25 eV. (a) 10-cobalt atom asperity, (b) 20-cobalt atom asperity, and (c) 35-cobalt atom asperity. ................................................................. 117

Figure 6.10 Cobalt interface mixing fraction contours on an ion energy-interface depth map. ........................................................................................................... 118
Figure 7.1. Atomic CoFe/Cu/CoFe/NiFe multilayers: (a) simulation at an adatom energy of 3.0 eV, and (b) atom probe experiment of a sputtered sample [28] .................. 122

Figure 7.2. Atomic CoFe/Cu/CoFe/NiFe multilayer obtained from simulation using an intralayer energy modulation scheme [28] .................................................. 124

Figure 7.3. Simulated Ni/Cu/Ni multilayer deposited at 1.0 eV and an oblique incident angle of 50°. (a) No substrate rotation and (b) substrate rotation rate 0.196 turns/ML [31] ........................................................................................................ 125

Figure 7.4. Conventional ion beam deposition approach ........................................ 127

Figure 7.5. Schematic illustration of a BTIBD system .............................................. 130

Figure 7.6. Schematic Layout of Mark II End-Hall ion source and HCES 5000 Hollow Cathode electron source ................................................................. 133

Figure 8.1 Schematic illustration of electromagnet. High field and high uniformity magnetic field is generated between the poles. Samples were placed at the center of the poles in the MR measurements ......................................................... 138

Figure 8.2 In-plane magnetoresistance properties were measured with a four point probe technique. The two outer probes were used to introduce a current, I, while the two inner probes were used to measure the potential difference between two positions along the current path. Both the sensing current and the magnetic field are in the plane of the film. Voltage drops, which can be used to deduce resistance, were measured as a function of magnetic field which arranged orthogonal (θ = 90°) or parallel (θ = 0°) to the current direction ......................................................... 139
Figure 8.3 Resistance vs. magnetic field curve for a spin valve of Ta (40 Å)/Ni$_{80}$Fe$_{20}$ (40 Å)/Co (15 Å)/Cu (33 Å)/Co (45 Å)/FeMn (100 Å)/Cu (20 Å) at room temperature.

Target voltage was 300 V and processing gas was Ar. ..................................................... 140

Figure 8.4 GMR ratio as a function of Cu spacer layer thickness for Ta (40 Å)/Ni$_{80}$Fe$_{20}$ (40 Å)/Co (15 Å)/Cu (t)/Co (45 Å)/FeMn (100 Å)/Cu (20 Å) spin valve deposited without ion assistance. ........................................................................... 141

Figure 8.5 GMR ratio and coupling field as a function of assisting argon ion energy.

Simultaneous ion assistance was only applied during deposition of the Co/Cu/Co part of the Ta/NiFe/Co/Cu/Co/FeMn/Cu spin valve................................................................. 143

Figure 8.6 Resistance vs. magnetic field curves for Ta/NiFe/Co/Cu/Co/FeMn/Cu spin valves with simultaneous ion assistance on Co/Cu/Co layers using 10 eV Ar ion. 143

Figure 8.7 Resistance vs. magnetic field curves for Ta/NiFe/Co/Cu/Co/FeMn/Cu spin valves with combination of different ion assistance schemes using parameters listed in Table 8.1. .................................................................................................................. 145

Figure 9.1 MD simulation results for Cu films deposited on a Ta buffer layer using a simultaneous ion assistance method: (a) Film packing density, (b) surface roughness, and (c) interfacial mixing as a function of assisting argon ion energies at a fixed ion-to-atom flux ratio and an ion incident angle of 45° ...................................................... 155

Figure 9.2 Simulation results for Cu films deposited on a Ta buffer layer using modulated energy and sequential ion assistance schemes: (a) Film packing density, (b) surface roughness, and (c) interfacial mixing and surface roughness as a function of assisting argon ion energies at an ion incident angle of 45°. ..................................................... 158
Figure Captions

Figure 9.3 Cu/Ta film properties as a function of assisting argon ion energy using simultaneous ion beam assisted deposition at an ion-to-atom ratio of 4 and incident angle of 45°: (a) film surface roughness and (b) film electrical resistivity. An empirical structure function, $U$, was deduced from simulation results and is shown as the solid curve in (b). .......................................................... 162

Figure 9.4 AFM film morphology dependence on assisting argon ion energy at an ion-to-atom ratio of 4 and incident angle of 45°: (a) no ion assistance, (b) ion energy of 7 eV, and (c) ion energy of 15 eV. .......................................................... 163

Figure 9.5 Cu/Ta film properties as a function of ion-to-atom flux ratio in simultaneous ion beam assisted deposition at an argon ion energy of 10 eV and incident angle of 45°: (a) film surface roughness, and (b) film electrical resistivity. Solid line in (b) is for structure function, $U$. .......................................................... 164

Figure 9.6 Measured auger electron spectrometer depth profiles of composition for Cu/Ta/substrate films deposited using simultaneous argon ion assistance deposition at an incident angle of 45° with three different ion assistance conditions: no ion assistance, 10 eV and ion-to-atom flux ratio of 5, and 20 eV and ion-to-atom flux ratio of 14. .......................................................... 166

Figure 9.7 High resolution TEM cross section images of the Ta/Cu bilayer interfaces deposited using different simultaneous ion assistance conditions at incident angle of 45°: (a) no ion assistance, (b) 10 eV and ion-to-atom flux ratio of 5, and (c) 20 eV and ion-to-atom flux ratio of 14. The CuTa region in (c) consists of a heavily mixed solid state solution. .......................................................... 168

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure Captions

Figure 9.8 Cu/Ta bilayer properties as a function of ion energy in modulated ion energy and sequential ion beam treatments: (a) film surface roughness, and (b) film electrical resistivity. The ion-to-metal flux ratio was 6 in the modulated ion energy scheme and an ion density of 1 mA/sec was applied for about 30 and 120 seconds to the completed tantalum and copper surfaces, respectively, in the sequential ion assistance. 170

Figure A.1 The computational cell used to simulate vapor deposition in the MD FORTRAN code. 186

Figure B.1 Map of functions for all the subroutines and fuctions with nanosim.f MD code. 199
CHAPTER 1. INTRODUCTION

1.1 Motivation

Nanoscale multilayers often exhibit special magneto-electrical properties that can significantly impact technology. In 1988, for instance, it was discovered that multilayers consisting of two thin (30 Å) Fe layers separated by a thin (9-18 Å) Cr layer exhibited a large drop in electrical resistance upon the application of an external magnetic field [1]. This giant magnetoresistance (GMR) was subsequently found in many other multilayers consisting of conductive ferromagnetic metal sandwiches [2-11]. The GMR phenomenon has quickly impacted technology; it is now widely used for highly sensitive magnetic field sensors in the read heads of hard disk drives [12-14], for galvanic isolators [15, 16], and for magnetic random access memories [12, 17] that promise nonvolatility, radiation hardness, low power consumption, high densities (comparable to dynamic random access memory), and higher access speeds [13, 14, 18-21]. The impact of these technologies has been significant. The introduction of GMR read heads in the hard disk drives is responsible for increases in aerial recording density of up to 100 Gb/in$^2$. The access time of magnetic access memory has also been decreased from milliseconds to 10 nanoseconds, which is might open up applications in a dynamic random access memory market in the near future.

The key GMR properties exploited in these applications include a high GMR ratio (the maximum electrical resistance change divided by the lowest electrical resistance at the saturated magnetic field) and a low switching magnetic field. These properties were
Chapter 1. Introduction

found to be extremely sensitive to the atomic scale structure of multilayers where layer thicknesses can be as little as 1-3 nm [22-25]. The Debye-Waller-like factor indicates that the interface roughness is very critical even at atomic length scales [26, 27]. Over the past several decades, extensive efforts have been made to grow chemically sharp, physically flat interfaces in metallic multilayers by optimizing the substrate temperature, deposition rate, background gas pressure, and by the use of surfactant-mediated growth conditions [2, 28-34]. As a result, the transport properties for these GMR multilayers have increased considerably [35-38]. Nonetheless, the best GMR properties achieved so far appear still to be well below theoretically predicted values [39]. Experimental studies indicate that all the existing deposition techniques result in films with noticeable interface imperfections [5, 22, 23, 35].

It is difficult to experimental characterize all the important atomic scale features of a metallic multilayer. This, combined with a limited understanding of the atomic assembly processes during metal multilayer synthesis, has been responsible for a slow convergence upon the ideal deposition processes for synthesizing thin film metallic multilayers. It has been suggested that a combination of computer modeling and carefully conducted experiments could provide a more effective way to discover the key factors controlling the interfacial structures of multilayers [30, 40-42]. In this dissertation, an improved deposition technique is developed using an integrated approach that combines both atomistic simulations and experiments. The growth of Co/Cu-based GMR multilayers is chosen for study because the Co/Cu system is one of the most promising
systems for GMR applications. However, the fundamental science revealed in this system is expected to be widely applicable to many other material systems.

1.2 Goals of the Dissertation

Molecular dynamics techniques are beginning to be used to understand atomic assembly mechanisms during the growth of thin metal films [28, 32, 43]. Atomistic simulations have been used to identify the effects of adatom energy [30], and adatom incident angle [31] on Ni/Cu GMR multilayer deposition. This dissertation will extend the use of this atomistic simulation approach to explore the effects of low energy ion assistance during the deposition of Co/Cu multilayers. The precise atomic structure requirement of GMR multilayers imposes stringent constraints upon the deposition conditions. The use of low energy, ion assisted vapor deposition processes is shown to provide flexibility for creating improved multilayer structures. The effects of ion energy, ion species, ion fluence, and ion incidence angle upon interfacial structures of Cu/Co multilayers are all investigated and the atomic assembly mechanisms that contribute to the final atomic scale structure are identified. Insights obtained from the simulations are used to design a novel, biased target ion beam deposition (BTIBD) method for the growth of spin valve multilayers. Experiments are conducted to grow and characterize the properties and atomic scale structures of multilayers using this approach. The results verify the predictions of the simulations and demonstrate significant improvements can be achieved in BTIBD grown multilayer films.
Chapter 1. Introduction

1.3 Structure of the Dissertation

The dissertation is organized to first identify the relations between multilayer structure and the performance of GMR devices. Atomic simulations are then used to simulate the growth of model structures under various ion assisted growth conditions. A practical approach for the implementation of the lessons learned from the simulations is described and this is used to grow real films.

Chapter 2 overviews GMR multilayer structures, describes their applications, and identifies the desired atomic scale structures. Chapter 3 describes atomistic simulation approaches that can be used for studying the atomic scale structures of vapor deposited GMR multilayers. Chapters 4, 5, and 6 use atomistic simulations to study the low energy ion assisted deposition of Co/Cu multilayers. The effects of ion energy and ion species are studied in Chapter 4, and the effects of ion fluence and ion incident angle are explored in Chapter 5. Chapter 6 focuses on ion assistance schemes that might be practically implemented. Chapter 7 describes a novel biased target ion beam deposition (BTIBD) technology developed using the atomistic simulation approach. Chapter 7 discusses the improvement of BTIBD grown multilayers over those fabricated by conventional ion beam deposition (IBD) techniques. Chapters 8 and 9 experimentally explore in detail the low energy ion assisted deposition of thin films using BTIBD and compares the results with those from simulations. Chapter 8 mainly discusses ion assistance effects on spin valve GMR multilayers, while Chapter 9 focuses on Cu/Ta bilayer films, which facilitate direct transmission electron microscopy (TEM) observation of the interfacial structures. The work presented in these two chapters also characterizes
film surface morphology, film thicknesses, film uniformity, interface roughness, and the interlayer mixing of deposited samples using a variety of techniques, such as profilometry, atomic force microscopy, auger electron spectroscopy and transmission electron microscopy. Chapter 10 consolidates the discoveries and observations from both the simulations and experiments, and summarizes them in a series of conclusions.
CHAPTER 2: INTRODUCTION TO GIANT MAGNETORESISTANCE

The development of electronics has been based upon the manipulation of electron charge and led to the development of a multitude of modern information technologies during the second half of the twentieth century. The emergence of new "spintronic" devices based upon the manipulation of the electron’s spin property now promises a new direction for microelectronics. For instance, the recent successful introduction of magnetic elements into semiconductors [39] and transparent oxides [44, 45], coupled with the observation of the prolonged spin lifetimes in some of these materials [46] has raised the possibility of eventually creating quantum computers with potentially orders of magnitude boosts in computational power [47-49]. Spin-related electron transport is also responsible for the large electrical resistance changes observed in some metallic multilayers when placed in small magnetic fields [1, 13]. These giant magnetoresistance (GMR) multilayers [1] have already been successfully used to create read head sensors for hard disk drives which in turn has enabled an increase in storage capacity by more than an order of magnitude [12, 19]. GMR materials are also being actively explored for a new generation of magnetic random access memory that is nonvolatile, radiation hard, low power consumptive, and is competitive in both access speed and capacity [17, 50]. GMR metal multilayer devices are thought to be the first of many spintronic devices exploiting metal multilayers [1], metal oxide films [51], and ferromagnetic semiconductors [39]. This dissertation is focused on the synthesis of GMR metal multilayers.
Chapter 2. Introduction to Giant Magnetoresistance

2.1. Magnetoresistance

Magnetoresistance (MR), which was discovered by William Thomson in 1857, refers to the change in electrical resistance of a material when placed in an external magnetic field. Many conducting materials exhibit small changes in resistivity when placed in a magnetic field because moving electrons receive force from external magnetic field. Much larger (~ 2%) changes occur in anisotropic magnetoresistance (AMR) materials such as many ferromagnetic metals and alloys [52, 53]. AMR originates from the orbital angular momentum of magnetic ions, the Lorentz force acting on conduction electrons, and a larger probability of s-d scattering of electrons in the direction of magnetic field. Normally, the resistivity is smaller when the magnetization is perpendicular to the electric current direction than when it is parallel, Fig. 2.1(a). Numerous applications of the phenomena have been explored in early devices, such as magnetometric positioning sensors [54, 55], rotary encoders [56], and read heads of digital audio tape recorders [57, 58] etc.. Fig. 2.1(b) shows an example device where a magnet is attached to a rotating disk and an AMR sensor is used to detect the speed of rotation from the resistance change of the MR sensor [59]. Efforts have also been made to apply AMR sensors to read information stored on magnetic recording devices since they allow a quick transformation of magnetic information to an electric signal. The storage capacity of the magnetic recording devices usually scales directly with the magnitude of the resistance change under a small field [60]. Unfortunately, the AMR effect typically gives a too small resistance change for modern applications of this type [61].
A breakthrough in magnetoresistance occurred in 1988 when Baibich et al. discovered the GMR effect [1]. Fig 2.2 shows their resistance-field curves for several Fe/Cr/Fe multilayer structures measured at 4.2 K. The magnetoresistance (MR) ratio can be quantified as the maximum resistance change divided by the lowest resistance. It can be seen that for the [3 nm Fe/0.9 nm Cr]_n multilayer, the MR ratio reached nearly 80%. The structure still retained an MR ratio of 17% at room temperature, which is about ten times larger than the AMR effect observed (~2%) [61]. This MR phenomenon was hence called "giant." It has now been established that when a pair of ferromagnetic (Fe) layers were separated by approximately 0.9 nm, exchange coupling occurred and their magnetic
moments became anti-parallel at zero external field. Electrons with any given spin directions were then scattered at one of the Fe/Cr interfaces and a high electrical resistance was observed. If a sufficiently large (~ 20 kOe in the structures studied by Baibich et al.) external magnetic field was applied, the magnetic moment of one of the Fe layers could be flipped so that the moments of the layers became parallel. The spin dependent scattering was then reduced and the materials resistance was significantly reduced.

![Graph showing the resistivity of Fe/Cr multilayers as a function of the magnetic field](image)

Figure 2.2 Resistivity of Fe/Cr multilayers at 4.2K as a function of the magnetic field [1].

GMR effects have since been observed in many material systems composed of two ferromagnetic metal layers separated by a conductive spacer metal layer, such as
Co/Cu/Co [5, 62] and NiFe(å)/Cu/NiFe(å) [63]. The key to simple GMR sandwiches is the existence of the exchange coupling so that the magnetic moments of the two magnetic layers are anti-parallel under an ambient magnetic field. Strong exchange coupling only occurs at discrete (periodic) spacer layer thicknesses [8]. As a result, the maximum GMR ratio is obtained at discrete exchange spacer layer thicknesses [8, 64].

However, if a device is conducted at one of the discrete spacing that maximizes the MR ratio, the coupling is usually so strong that a large external field is required to switch the magnetic moment of one of the magnetic layers in order to achieve the parallel magnetic moment alignment [65]. This is undesirable for magnetic sensing applications. The problem can be resolved using "spin-valve" multilayer structures where an antiferromagnetic layer is deposited before the first ferromagnetic layer. This strongly pins the magnetic moment of the magnetic layer [66]. Parallel and anti-parallel magnetic moment alignments of a pair of layers can then be achieved by rotating the moment of the free magnetic layer using the external field. A much thicker spacer layer can then also be used to deliberately avoid the strong exchange coupling resulting in a device that can sense quite weak magnetic fields [65, 67].

It should be pointed out that other phenomena similar to giant magnetoresistance have recently been discovered. For instance, an extremely large MR effect has been found in manganese perovskite oxides, and has been termed colossal MR (CMR) [68, 69]. Unfortunately, the magnetic field required to achieve the magnetoresistance change is currently too high (10 kOe) for most practical applications [70, 71]. Spin dependent tunneling junctions consisting of two ferromagnetic metal layers separated by a thin
oxides layer also exhibit a larger MR ratio than GMR multilayers [51, 72]. Remarkable progress has been made in the development of tunnel magnetoresistance (TMR) multilayers and they are being developed for nonvolatile magnetic random access memories [73]. Finally, many granular materials constructed from ferromagnetic metal systems also exhibit quite MR ratios [74, 75]. It appears to be a 3D manifestation of the 2D phenomenon found in thin film.

2.2. GMR Mechanisms

The excitement created by the discovery of the GMR effect has inspired extensive studies of its mechanism [8, 52, 61, 64]. Selective scattering of conduction electrons at the interfaces between the ferromagnetic and nonmagnetic layers and differing penetration of conduction electrons into ferromagnetic layers are believed to be the cause of the phenomenon [76]. This is schematically illustrated in Fig. 2.3 below. The spin of the majority electrons in ferromagnetic layers are oriented parallel to the magnetization vector, M, in Fig. 2.3.
Figure 2.3 Electron transports in a multilayer structure. The white arrows show the magnetic moment of the ferromagnetic layers. Extra electron scattering occurs when the moments are antiparallel. Only interface scattering is shown in the diagram.

When an electric field is applied, the conduction electrons are accelerated until they encounter a scattering center. The mean free path, or the average distance the conduction electrons migrate before scattering by phonons and other electron scattering defects, is of the order of 50 Å in the metals used in GMR multilayers [77]. This is comparable to the spacing between interfaces in GMR multilayers and so many electrons are likely to arrive at the interfaces before being scattered by other scattering centers.

Consider the transport of those electrons not subjected to intrinsic scattering in the multilayer plane, Fig. 2.3. When adjacent magnetic layers are magnetized in an antiparallel alignment, as shown in Fig. 2.3(a), the spin direction of any conduction electrons will not always match the majority spin direction in one of the magnetic layers. Significant electron scattering then occurs when conduction electrons try to enter such a
magnetic layer, resulting in a high resistance. When the adjacent magnetic layers are magnetized in a parallel manner, as in Fig. 2.3(b), at least 50% of the electrons in the spacer layer (assuming the spin up and spin down electrons are equally partitioned) have identical spin directions to the majority spin directions in both ferromagnetic layers. These electrons are likely to enter the adjacent ferromagnetic layers with negligible scattering when they arrive at the interfaces.

A GMR sandwich requires antiferromagnetic coupling between the two ferromagnetic layers. The existence of periodic antiferromagnetic coupling has been predicted by the Ruderman-Kittel-Kasuya-Yosida (RKKY) theory and confirmed in many systems [8, 61, 64]. For instance, the MR ratio of Co/Cu multilayers has been measured as a function of the Cu layer thickness [61]. The results are shown in Fig. 2.4. It can be seen that the MR ratio periodically varies with Cu layer thickness and peaks at discrete Cu layer thicknesses (10 Å, 20 Å ... etc). The saturation magnetic field (not shown) also peaks at the peak MR ratios. Parkin et al. found that oscillation of interlayer coupling with wavelengths of 1~1.5 nm occurs quite generally in a wide variety of magnetic multilayer systems including systems such as Co/Ru and Co/Cr [64].
Chapter 2. Introduction to Giant Magnetoresistance

Figure 2.4 Room temperature magnetoresistance vs. Cu spacer layer thickness for a series of Co/Cu multilayers [61].

Of the many various multilayer systems investigated, the Co/Cu/Co multilayer system has both a relatively low coercive field and a very high MR ratio. This is well suited for magnetic sensing applications and hence has been one of the most widely systems [5, 62]. The Co/Cu/Co system is also a well studied system because its mechanism of interlayer coupling appears to be simpler than that of other systems [78-81]. These simple sandwich systems provide a relatively straightforward opportunity to study the mechanism responsible for the GMR effect. For practical applications, Dieny et al. proposed more complicated spin-valve structures [67], which as mentioned above achieve anti-parallel coupling by first pinning the magnetic moment of one of the ferromagnetic layers by an antiferromagnetic layer (e.g., FeMn). The first spin-valve
structures consisted of a NiFe 15 nm/Cu 2.6 nm/NiFe 15 nm sandwich deposited on a 10 nm thick FeMn antiferromagnet. The unpinned NiFe layer is very soft. As a result, the relative orientations of the two magnetic layers are extremely sensitive to the external field. Due to the very high MR ratio achieved at a very low magnetic field [66], spin-valve systems have been successfully applied as magnetic field sensors [82, 83]. Larger GMR ratios have been reported when repeats of the simple GMR sandwiches are stacked together [1]. This approach, however, cannot be applied for spin-valves because of difficulties in getting antiparallel magnetic moments.

It should be pointed out that spin-valve devices utilized the GMR effect measured with the electrical current in the plane (CIP) of the layers. This geometry is easy to measure. The GMR effect also occurs when the current is perpendicular to the plane of the layers (CPP). Since electrons are forced to enter both magnetic layers, the CPP GMR ratio is usually higher than the CIP GMR ratio. For metal only GMR structure, it is difficult to make a CPP structure because measured device resistance is always controlled by contact resistance due to extremely low resistance of the GMR structures in vertical direction. Unlike the CIP, the measurement of the CPP GMR ratio requires patterning of the wafer. Nonetheless, the CPP geometry is essential for the ongoing efforts to develop magnetic random access memory devices [17, 50].

The existence of defects both within the metal and at multilayer interfaces can significantly affect GMR properties. For instance, interface roughness [84-86] and interlayer mixing [87] can cause undesired coupling and electron scattering. Their presence can greatly reduce the GMR ratio [88, 89]. On the other hand, atomically
smooth interfaces can act as "mirrors" where electrons are reflected specularly. Egelhoff et al. [36] and other recent papers [90] report that such interfaces resulted in a very high GMR ratio. By designing more sophisticated material structures to reduce these defects, numerous recent studies indicated that spin-valves with a very high GMR ratio could be created without stacking repeats of GMR sandwiches.

2.3. GMR Applications

GMR materials have many potential applications ranging from electrical motors and generators, to compasses and metal detectors [91]. Perhaps the most remarkable to date are their use for the read heads of magnetic recording devices and nonvolatile magnetic random access memories.

2.3.1 Magnetic Read Head Sensors

Spin-valve structures have been used to construct read head sensors for magnetic recording devices because of their good response to weak magnetic fields. A schematic read head sensor for magnetic recording hard disk devices is illustrated in Fig. 2.5. A magnetic recording medium (the hard disk) rotates with high speed and the head writes and reads digital signals in circular tracks on the disk. The write and read head is positioned on the front of a slider that moves across the radial coordinate of the rotating disk. The write head portion has a write pole composed of a high saturation field material (permalloy) and a write gap for signal writing using the magnetic flux generated by copper coils. The coils create a locally strong magnetic field which orient the moment of
the magnetic domains in the recording medium. The read head portion has an insulator (AlO₃), in which the GMR read head is placed. It is magnetically shielded from the write part of the head.

![Figure 2.5 Schematic drawing of typical magnetic recording disk drive and spin-valve read head [92].](image)

A typical spin-valve structure like that shown in Fig. 2.6 possesses a suitable combination of both GMR ratio and switching field for sensing alignment of magnetic domains in the read head. It can be seen that when the GMR read head scans over the disk, the magnetic information (field) stored in each of the magnetic bits is quickly converted to a corresponding electrical signal. The nano-scale size of a GMR sensor, its high sensitivity to magnetic field, and its low cost for production have allowed the area
density of recording devices to grow at between 100% and 200% per year since the initial introduction of spin-valve read heads by IBM in 1997 [93-95].

![Diagram of giant magnetoresistive spin valve structure](image)

Figure 2.6 Schematic representation of giant magnetoresistive spin valve structure.

2.3.2 Nonvolatile Magnetic Memory

Magnetic random access memory applications are very attractive primarily because of the nonvolatile property, i.e., the memory is maintained even when power is removed from the memory [96, 97]. A variety of MRAM technologies have been long explored, including macroscopic ferrite core memory [98] and magnetic bubble memory [99]. Ideally, arrays of nanoscale magnetic storage cells are desired for a high density of the memory. These cells must have two different measurable stable magnetic states, representing “0” and “1,” respectively. The significant advantages of GMR materials are their low cost, simple implementation, and a large signal (due to their high MR ratio). These combine to creation of new MRAM devices that might become competitive with other memory technologies [100].
A schematic illustration of a proposed MRAM architecture is shown in Fig. 2.7. GMR memory cells are located at the cross-points between a lower array of conductive “word” lines and an upper array of conductive “bit” lines. Using this structure, any given cell can be addressed by passing currents in the corresponding pair of the word and bit lines. During writing, the currents are adjusted so that the magnetic field created by either the word or the bit line alone is not sufficient to switch the magnetic moment of the magnetic layers in the cell, but their combined magnetic field is sufficient to achieve the switch. This allows the “writing” (switch of the magnetic moment) to be achieved only to the targeted memory cell. To read the state of a magnetic cell, a current is passed through that cell, which creates a corresponding electrical signal.

(a) Read mode

(b) Program mode

Figure 2.7 Cartoon of cross-point MTJ MRAM architecture: Magnetic tunnel junction storage cells are located cross-point of the word and bit lines. The (a) and (b) cartoons show the processes of reading and program modes [92].

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
2.4 Growth of GMR Films

GMR multilayers have been most successfully grown using physical vapor deposition methods (PVD). In a PVD process, vaporized materials are condensed on a wafer surface. Control of the atomic scale structures of the deposited multilayers can be very complicated. These structural features are sensitive to the characteristics of the vapor flux and a variety of atomic assembly and diffusion mechanisms on the growth surface. It is very important to identify the key growth conditions essential to good GMR properties. Although GMR multilayers were first grown using a thermal evaporation approach, (molecular beam epitaxy) [1], sputtering and ion beam deposition techniques have been found to produce much better GMR properties [2-5, 7, 9, 11, 29, 63]. This observation can be understood by comparing the processing conditions provided by each of the deposition methods.

2.4.1 Molecular Beam Epitaxy

With an ultra-high vacuum system and a precisely controlled low rate of deposition, molecular beam epitaxy (MBE) has been successfully used for the epitaxial growth of pure metals [101, 102], defect-free semiconductor multilayers [103], and even epitaxial oxides including high T_c-superconductors [104] and magnetic perovskites [105]. MBE systems use evaporation sources such as Knudsen cells to create a thermalized plume [106]. These sources provide stable pressure vapor plume by precise temperature control of the melt source. They allow constant flux vapor beam with a low flux and low (thermalized) vapor atom energy.
2.4.2 Sputtering

Two approaches based upon magnetron and radio frequency (RF) diode sputtering have been used to grow GMR films [107, 108]. A schematic diagram of magnetron sputtering is shown in Fig. 2.8. A plasma is initiated by application of an electric field. A low pressure is used to maintain a high energy inert gas ion flux which sputters atoms from a target. This creates a flux of high energy target vapor atoms. The growth of a thin film proceeds by condensation of these atoms on the growth surface. Scattering by the background gas during vapor atom transport from target to substrate, decreases the adatom energy, and the average vapor atom energy during condensation generally decreases as the chamber pressure and target-substrate distance are increased. The high background pressure results in a relatively low sputtering ion flux and therefore a low rate of film growth. In magnetron sputtering, a set of permanent magnets is used to confine the plasma near the target increasing the sputtering flux. The advantage is that it significantly increases the deposition rate without having to increase the supply of electrons [109, 110]. Magnetron sputtering has been quite successfully used to grow GMR films [111, 112]. Nonetheless, there are several issues. For instance, there are constraints on target materials and target dimensions, the nonuniform plasma density results in nonuniform target erosion, and it is difficult to create a uniform magnetic field around large and complex surfaces [110, 113].
A schematic diagram of a RF diode sputter deposition system with a plane parallel electrode geometry is shown in Fig. 2.9. An inert gas plasma is created and maintained in the chamber by an RF power source. The positive gas ions are accelerated across the plasma sheath potentials to strike both the target and the substrate. The accelerated ions bombard the target surface and result in sputtering of target atoms. These sputtered atoms are then transported to substrate to deposit films. Energies and fluxes of the ions are mainly determined by RF power applied to the plasma and background pressure. The
energy and flux of both metal and gas ions usually have potentially important effects
upon the resulting GMR properties [115].

Figure 2.9 A schematic illustration of RF diode sputter deposition system.

2.4.3 Ion Beam Deposition

Fig. 2.10 schematically illustrates a typical ion beam deposition system with
optional ion assistance. Instead of using a plasma to provide the energetic sputtering ions,
an ion beam deposition (IBD) process uses an ion source to create a collimated beam of
ions directed at the target for the sputtering [116]. In IBD, the ion energy and deposition
rate can be independently controlled and substrate heating from the plasma is reduced
[116]. A high deposition rate can be achieved at very low working gas pressures. As a
result of a low chamber pressure, the background gas scattering is reduced and adatoms
may have higher energies than that of magnetron sputtering. IBD has been successfully
used to grow GMR multilayers [117, 118]. However, IBD still suffers from many
problems, such as overspill contamination, a low deposition rate, nonuniform target
etching, damage from reflected neutrals, and difficulties in operating at low adatom
energy ranges, and oblique adatom incident angles [119, 120].

Figure 2.10 Schematic illustration of an ion beam deposition system with ion beam assistance.

2.5 Desired GMR Structures

Extensive evidence has been accumulated showing that the transport properties of
GMR multilayers are extremely sensitive to the thickness of the spacer layer. For spin-
valves, the optimal spacer layer thickness lies between 25 Å and 35 Å. Depending on the
design, the magnetic layer thickness can be between 20 Å and 100 Å. Under this
Chapter 2. Introduction to Giant Magnetoresistance

structural constraint, interfacial roughness on the scale from nanometers to tens of nanometers can lead to a magnetic coupling of a dipolar origin called “orange peel” coupling, Fig. 2.11 [121]. This coupling increases the difficulty of rotating the moment in one magnetic layer relative to that in the other. Rumpled layers also cause anisotropy dispersion [122, 123], since the local planes of the magnetic layers will rarely be parallel to the plane of the substrate. Interfacial roughness at the spacer/ferromagnetic interface has been found to sharply reduce the GMR ratio [2-4, 6, 7, 29]. X-ray diffraction data obtained for Co/Cu multilayer systems have revealed that reductions of the interfacial roughness enhanced GMR effects [87, 124]. Parkin et al. also found that the use of an Fe buffer layer can improve GMR because the Fe underlayer smoothed the interfaces. It has also been established that the majority electrons in Cu traveling nearly parallel to the layers can be totally reflected at Cu/Co interfaces in Co/Cu/Co multilayers because the majority Fermi surface of Co is slightly smaller than the Fermi surface of Cu. This occurs on both sides of a Cu spacer layer if the moments in the ferromagnetic layers are aligned but on only one side if they are anti-aligned. Because these electrons stay in the Cu layer where the resistivity is low, the parallel resistance is significantly decreased and the GMR is increased when this transport mechanism is maximized. However, this “wave guide” effect requires atomically smooth interfaces [125].
Atomic mixing at the magnetic and non-magnetic layer interfaces can significantly degrade the GMR properties in other ways [29]. For instance, if Ni diffuses into the Cu spacer layer, it can lose its moment alignment and become a center for spin-independent scattering that reduces the GMR [89]. Diffusion of Fe or Co into the Cu layer can have an even worse effect because these Fe or Co atoms maintain their moment and can cause spin-flip scattering [88]. Significant mixing can also result in a change in the “virtual” spacing and hence the magnetic coupling between the two magnetic layers due to the creation of magnetic dead-layers at the interfaces, Fig. 2.12. Because layer thicknesses in most GMR systems are only 1–2 nm, roughness and mixing must both be controlled at a high spatial resolution to prevent the formation of pinholes (which physically and magnetically connect the two magnetic layers). It is therefore essential to minimize both interfacial roughness and interlayer mixing to achieve the maximum GMR performance for GMR multilayers.
2.6 Improved Deposition Approaches

Interdiffusion and interface mixing can be minimized by using a low substrate temperature and reducing the high energy ion irradiation that accompanies some growth processes. Under such conditions, however, adatoms do not always have a high enough surface mobility to find the lowest energy positions leading to both increased surface and interfacial roughness [126, 127].

Thermodynamic analysis indicates that, at least for binary systems, nanoscale multilayers in which all interfaces are smooth are not stable. They therefore cannot be grown under equilibrium growth conditions [128, 129]. This in part accounts for the experimental observation that low temperature deposition methods such as magnetron sputtering and ion beam deposition (IBD) produce better GMR multilayers than those made using molecular beam epitaxy method [2-5, 7, 9, 11, 29, 63]. This occurs even though the latter can be carried out in a higher vacuum and therefore cleaner chamber. However, these deposition processes have other important differences, perhaps the most...
significant of which is the average kinetic energy of the depositing atoms and the presence of energetic assisting ions.

Extensive theoretical and experimental studies appear to indicate two approaches can potentially reduce interfacial structure in GMR multilayers. The first centers around the idea controlling the adatom energy [30]; the second is based upon a surfactant mediated growth [33, 34, 36]. A third approach investigated here is based upon the use of low energy ions to assist deposition.
CHAPTER 3: ATOMISTIC SIMULATION METHODS

Computer simulations are playing an increasingly important role in scientific research today. This is a reversal of the past when the advance of science often relied on experimental discoveries and utilized theories to interpret them. Despite the complexity of nature, theories often could only be validated to a few simple cases where solutions from mathematical equations were possible. In many cases, the complexities invariably associated with real world problems must be deliberatively removed for the problems to be solvable. Many extremely interesting physical problems encountered at surfaces, with clusters of atoms, and the study of phase transitions fell outside this scope.

The emergence of high speed computers in the 1950s opened a new research road: computer simulations (or the use of computers for experiments). Simulations have now become an important bridge connecting theories with experiments. Simulations are becoming extremely powerful tools not only to understand and interpret the experiments from atomistic to macroscopic length scales, but also to study regimes that are not easily accessible from experiments. For the complicated atomic assembly problems during physical vapor deposition, a multiscale modeling approach shown in Fig. 3.1 needs to be used.
Figure 3.1 Multiscale modeling tools for physical vapor deposition [130].

Molecular dynamics (MD) is one important atomistic simulation method. It has been successfully applied in many problems, including the study of viscosity [131], heat flow [132], defect properties [133], fracture [134], surface reconstruction [30, 130], melting [135], crystal faceting [136, 137], diffusion [138], roughening [139], and friction [140].

This chapter first discusses the MD method used here to simulate the growth of GMR multilayers. The embedded atom method interatomic potentials used to define the interatomic forces during MD simulations are then described. Finally, a molecular statics (MS) method used to determine the low energy crystal configurations is introduced.
3.1. Molecular Dynamics

Molecular dynamic (MD) simulation essentially determines the motion of all individual atoms in computational solids, liquids, and gases. Here motion is referred to as positions and velocities as a function of time. This allows the microstructure evolution in any materials systems to be resolved at the level of individual atoms. The earliest MD simulations in 1957 assumed a hard sphere model, and were used to investigate a phase diagram [141]. The first MD simulation that integrated atom positions based upon a continuous interatomic potential occurred in 1960 [142]. In that work, creation of defects during radiation damage in a 500-atom system was successfully studied. In 1964, Aneesur Rahman published his MD simulations of liquid Ar [143]. This greatly promoted the development of the field.

MD simulations can be applied to study different equilibrium and non-equilibrium systems. For instance, MD is typically applied to an isolated system. Such a system contains a constant number of atoms, $N$, a constant volume, $V$, and a constant total energy (sum of molecular kinetic and potential energies), $E$. The variables $N$, $V$, and $E$ specify the thermodynamic state of the system. Assume that the total potential energy, $V_N$, of the system is a sum of the pairwise potential energy, $\phi(r_{ij})$, between atoms $i$ and $j$ separated by a distance, $r_{ij}$,

$$\frac{1}{2} \sum_{i,j \neq i} \phi(r_{ij}), \quad i,j = 1,2,\ldots,N. \quad (3-1)$$

Then in the constant NVE MD, atom positions are obtained by solving Newton’s equations of motion:
\[ \overline{F_i}(t) = m_i \ddot{r}_i(t) = \sum_{j \neq i} \frac{d}{dr_{ij}} \phi(r_{ij}) \quad i = 1,2,\ldots,N, \quad (3-2) \]

where \( r_i \) and \( m_i \) are position and mass of atom \( i \), \( F_i \) is the force acting on atom \( i \) by the other atoms, and \( r_{ij} \) is the spacing between atoms \( i \) and \( j \).

The above equation gives snapshots of atomic configurations of the system as a function of time. Each of the atomic configurations is associated with a set of properties, such as the lattice constant, potential energy, and kinetic energy (temperature), etc. The corresponding experimental measured properties are the time average, \( \langle A \rangle \), of these properties.

\[ \langle A \rangle = \lim_{t \to \infty} \frac{1}{t} \int_0^t A(\tau) d\tau. \quad (3-3) \]

In addition to isolated systems, an external stress can be applied so that the volume and energy of the system are no longer constant. Such MD methods were first introduced in the early 1970s [144-146] to compute transport coefficients and have now been successfully used to obtain the shear viscosity, bulk viscosity, thermal conductivity, and diffusion coefficients [147].

MD is typically applied to systems containing from several hundred to a few hundred of thousands atoms [148]. Such small systems have associated artificial surface effects. For example, a 500-atom liquid cube can have an edge length scale of about 8.5 atomic diameters, well within the wall-fluid interaction range of between 4 and 10 atomic diameters from the wall. Surface effects could be minimized by using periodic boundary conditions. This means that the computational cells are periodically repeated in a three-dimensional space. However, periodic boundary conditions impose constraints on the
way the cell deforms and are hence not applicable to situations such as phase transformation that lose the periodicity. During MD simulations where external stresses are applied, the period lengths are allowed to change to reflect the response of the simulated system to the external stresses.

In this dissertation, the MD model (Appendix A and B) of thin film deposition developed by Zhou and Wadley was used [28, 30, 149]. The interatomic potentials are based on the embedded-atom (EAM) potentials developed by Johnson and his collaborators [150]. For simulation of an infinite film, periodic boundary conditions were used in the lateral directions (normal to the growth direction). Initial crystalline substrates were created by assigning positions to atoms at the ideal bulk lattice sites and velocities to atoms according to the Maxwell Boltzmann distribution. Growth was simulated by continuously injecting adatoms to the top (growth) surface at a frequency that gave rise to the desired deposition rate. Desired adatom energy and adatom incident angle were specified by the initial velocity vectors of adatoms. To prevent the shift of crystals due to impact of adatoms at the top surface, several bottom monolayers of atoms are fixed during simulations. A subsurface region was fixed at a constant substrate temperature using dragging forces. This also provided an effective energy sink to prevent the energy accumulation during adatom impacts. The positions of all atoms were then solved as a function of time. The numerical time step used was in the order of 1 fs (10^{-15} s) so that the atom oscillations were accurately simulated.
3.2. Interatomic Potentials

Many potentials have been proposed for MD studies of vapor deposition and for MS calculations to identify the minimum energy atomic configurations and minimum energy path for atomic diffusion during thin film growth. For the transition metals such as those needed in GMR applications, the embedded atom method (EAM) approach originally developed by Baskes and Daw [151, 152] is particularly effective. The EAM correctly predicts the elastic constants and vacancy formation energy, and has been successfully used to study a variety of surface and interface problems [152]. The standard embedded atom method (EAM) potential for binary Cu-Ni alloys developed by Foiles [153] can be employed to explore Ni/Cu multilayers. The EAM database has now been developed for at least 15 metals and their alloys: fcc Cu, Ag, Au, Ni, Pd, Pt, and Al; bcc Fe, Mo, Ta, and W; and hcp Mg, Co, Ti, and Zr. This EAM database has been successfully applied to explore the metal GMR multilayers with increasing chemistry complexities. This potential database is described in the following discussion.

Assume that the total potential energy of the system, $V_N$, can be distributed among atoms so that

$$V_N = \sum_i E_i, \quad i = 1, 2, \ldots N,$$

(3-4)

where $E_i$ is the energy of atom $i$. In EAM, $E_i$ is composed of an embedding energy, $F_i(\rho)$, for embedding the atom into its lattice site associated with a local electron density, $\rho$, and the pair energies between this atom and its neighbors, $\phi_{ij}(r_{ij})$: 
Chapter 3. Atomistic Simulation Method

\[ E_i = F_i(\rho_i) + \frac{1}{2} \sum_{i \neq j} \phi_{ij}(r_{ij}), \quad i, j = 1, 2, \ldots N. \]  
(3-5)

The electron density, \( \rho_i \), is a measure of the local environment. \( \rho_i \) can be calculated using

\[ \rho_i = \sum_{i \neq j} f_i(r_{ij}), \quad i, j = 1, 2, \ldots N, \]  
(3-6)

where \( f_i(r_{ij}) \) is the electron density at the site of atom \( i \) arising from atom \( j \) at a distance \( r_{ij} \) away from \( i \).

The three functions, \( \phi(r) \), \( f(r) \), and \( F(\rho) \) have been determined for a variety of metal elements. One problem in studying alloys using EAM potentials is that the elemental potentials cannot be simply combined to form the alloy potentials. In addition, the cutoff distances used for different elements are usually not consistent. An alloy EAM model [150] enabled the merge of elemental EAM potentials to study alloys. A universal cutoff procedure was also introduced to ensure the consistent cutoff of the potentials for different elements.

A Morse potential has been used for the pair potentials. It contains a short-range repulsive exponential and a long-range attractive exponential. For a smooth cut off of the potential, each term is multiplied by a factor that approaches unity as atomic separation, \( r \), decreases from the cutoff range and approaches 0 as \( r \) increases from the cutoff range. That is,

\[ \phi(r) = \frac{A \exp \left[ -\alpha \left( \frac{r}{r_e} - 1 \right) \right]}{1 + \left( \frac{r}{r_e} - \kappa \right)^n} - \frac{B \exp \left[ -\beta \left( \frac{r}{r_e} - 1 \right) \right]}{1 + \left( \frac{r}{r_e} - \lambda \right)^n}, \]  
(3-7)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
where \( r_e \) is the equilibrium spacing between nearest neighbors, \( A, B, \alpha, \) and \( \beta \) are four adjustable parameters, and \( m, n, \kappa, \) and \( \lambda \) are four additional parameters for the cutoff.

The electron density function is taken with the same form as the attractive term in the pair potential as follows:

\[
\rho_i = \frac{\rho_0 \exp \left[ -\beta \left( \frac{r}{r_e} - 1 \right) \right]}{1 + \left( \frac{r}{r_e} - \lambda \right)^n}.
\]  

(3-8)

The requirement of the embedding energy is that it has a zero value at \( \rho = 0 \) and is physically reasonable at a large \( \rho \). In the normalized form of the potential, we further require that the embedding function independently satisfies the equilibrium condition, with its slope equal to zero at the equilibrium electron density, \( \rho_e \). The following split function was used for \( F(\rho) \),

\[
F(\rho) = \sum_{i=0}^{3} F_i \left( \frac{\rho}{\rho_e} - 1 \right)^i, \rho < \rho_n, \rho_n = 0.85 \rho_e,  
\]  

(3-9)

\[
F(\rho) = \sum_{i=0}^{3} F_i \left( \frac{\rho}{\rho_e} - 1 \right)^i, \rho_n \leq \rho < \rho_0, \rho_0 = 1.15 \rho_e,  
\]  

(3-10)

\[
F(\rho) = F_e \left[ 1 - \ln \left( \frac{\rho}{\rho_e} \right)^n \right] \left( \frac{\rho}{\rho_e} \right)^n, \rho_0 \leq \rho.  
\]  

(3-11)

By fitting the cohesive energy, the lattice constant, the bulk modulus, the shear modulus, the elastic anisotropy ratio, the pressure derivative of the bulk modulus, and vacancy formation energy, we could determine all of the parameters for our desired metals.
3.3. Molecular Statics Calculations

Many kinetic processes at the atomic scale are determined by the activation energy barriers for atoms to jump from one site to another. The equilibrium crystal configurations are also given by the minimum energy states. Molecular statics (MS) calculations can be used to either relax a crystal by minimizing its energy or to determine the minimum energy path and the associated energy barrier [154, 155]. For the latter situation, a selected atom is moved in the jump direction in steps. The energy of the crystal is then minimized with the constraint that the moving atom is not allowed to move in the step direction, but its relaxation in the plane perpendicular to the step direction is free. In this way, the trajectory of the jumping atom is not forced to be a straight line. Instead, it follows a minimum energy path similar to the one during real crystal diffusion. The energy versus step distance curve then allows the determination of the energy barrier of diffusion. It should be noted that no thermal vibration is simulated in MS calculations. This means that the temperature is 0 K (see Appendix A and B for code). MS has proven to be a good method in estimating metallic crystal surface diffusion rates.
CHAPTER 4. LOW ENERGY ION ASSISTED ATOMIC ASSEMBLY OF METALLIC SUPERLATTICES

Metallic superlattices with planar, unalloyed (unmixed) interfacial structures are difficult to fabricate by all conventional vapor deposition methods. In this chapter, MD simulations have been used to explore the ways in which inert gas ions can be used to control the atomic assembly of a model Cu/Co metallic super lattice system. High energy, high atomic weight ions are shown to smooth rough interfaces but introduce undesirable intermixing at interfaces. Light ions with very low energies fail to flatten the rough surfaces that are naturally created during deposition at ambient temperature where surface atom mobility is kinetically constrained. The optimum energies for achieving the lowest combination of interfacial roughness and interlayer mixing have been found for each inert gas ion species and the key mechanisms of surface structure reorganization activated by ion impacts have been identified over the range of ion masses and energies studied. Optimum ion energies that maximize the interface structural perfection have been identified.

4.1. Introduction

As mentioned in previous chapters, metallic multilayers with nanoscale layer thicknesses (sometimes referred to as metallic superlattices) exhibit a number of unusual properties [1]. For instance, superlattices consisting of repeated Cr/Sc bilayers with layer thickness in the 5-15 Å range specularly reflect X-rays and can be used as mirrors [156] for X-ray microscopy [157, 158], astronomy [159, 160], lithography [161], and
Chapter 4. Low Energy Ion Assisted Atomic Assembly of Metallic Superlattices

microanalysis [162]. Multilayers consisting of repeated thin (~50 Å) ferromagnetic layers (such as Co, Fe, Ni, and their alloys) separated by thin (10-35 Å) conductive metal layers (such as Cu) exhibit large changes in electrical resistance upon the application of a magnetic field [29, 163]. These giant magnetoresistive (GMR) materials have been utilized to construct magnetic sensors [12]. Magnetic tunnel junction multilayers composed of a pair of ferromagnetic layers sandwiching a thin dielectric layer exhibit even larger resistance changes and are being explored for magnetic random access memory [17].

These unusual properties of metallic multilayers are highly sensitive to the atomic scale structure of their interfaces [7, 88, 89, 157, 164, 165]. For each of the applications above, performance is improved if the interfaces separating the different layers are chemically sharp and atomically smooth [88, 89, 156]. Such a structure is invariably metastable and cannot therefore be achieved under equilibrium growth conditions. Instead, the kinetic phenomena active during the atomic assembly of the interface are constrained in order to trap a desired structure. The basic approach involves maintaining the growth surface at a low temperature to impede surface kinetics and to then use the momentum, energy, or incident angle of the arriving atoms and/or an assisting inert gas ion flux to nudge surface atoms into locations that result in desired structures. Surfactants and surface alloying can also be exploited at times to preferentially bias atomic jumps that promote step flow growth leading to planar surfaces [34].

Recent MD simulations have shown that under kinetically constrained growth conditions, an increase in an adatom’s translational energy can be used to flatten a growth surface by impact-induced mechanisms [28, 30, 31]. In this process, the incident energy
is transferred to surface atoms resulting in either enhanced surface migration or removal of atoms from loosely bound surface sites. MD simulations have also shown that high incident energy atoms can cause interlayer mixing by an atomic exchange mechanism [28, 30, 31]. It is therefore important to optimize the adatom energy so that it produces the lowest combination of interfacial roughness and interlayer mixing.

Ion beam assisted deposition is also an effective growth approach for modifying surface assembly. Both computational and experimental studies of ion assisted deposition have identified strong ion energy effects upon film structure, stress, density, and growth morphology [166-168]. However, most of these studies have concentrated upon the effects of high energy ions (in the 100-5,000 eV range). Unless a glancing incident angle is used, these high energy impacts cause extreme interlayer mixing, and it would appear that they are poorly suited for creating the metallic superlattices used for X-ray mirror and magnetoeletronic applications [28, 30, 31].

The use of very low energy ion assistance appears to be a promising approach for manipulating multilayer interfaces. Experiments by Birch et al. have shown that the performance of Cr/Sc X-ray multilayers is improved using 9 eV argon ion assisted deposition [156]. A group at Veeco also investigated the effects of ion assistance on properties of GMR stacks [169]. In their studies, assisting ion fluxes were controlled using radio-frequency bias, gas cluster ion beams, and other ion beam treatments. They found that ion energies in the range 10-60 eV significantly improved the GMR ratio of their films. Kools also found that slight increases in kinetic energy (> 5 eV) could lead to biased diffusion and the suppression of defect formation [170].
The experiments described above have explored a narrow subset of the many ion energies, ion types, and ion incident angle combinations potentially accessible to film growers. These studies also shed little light upon the atomic assembly processes responsible for performance improvements. Computational approaches based upon MD have proven useful for exploring the assembly of thin multilayers. The predicted structures have been shown to often be in remarkably good agreement with those obtained by atomic resolution structure characterization methods, such as the three-dimensional atomic probe technique [28]. Previous atomistic simulations of low energy ion impacts with small atom clusters have shown that low energy (5~20 eV) ion assisted deposition can reduce interfacial roughness without causing interlayer mixing [171-173]. However, the precise role of the ion mass and energy upon the atomic scale reassembly mechanisms were not investigated in these earlier studies.

Roughness and intermixing at interfaces during the vapor deposition of metallic superlattices are controlled by many complicated atomic assembly mechanisms. Under the kinetically constrained growth condition that is used to prevent thermally assisted interdiffusion between layers of different compositions, randomly deposited atoms have insufficient mobility to always migrate to the lowest energy (most coordinated) sites. As a result, a high density of surface terraces form on the surface. Terraces formed on top of other terraces lead to asperities, which can preferentially intercept the incoming vapor flux and therefore grow at an accelerated rate. The adjacent (valley) regions are then flux "shadowed," further increasing the roughness, Fig. 4.1(a).
Figure 4.1 (a) Atomic reassembly of a surface under kinetically constrained conditions showing terrace/asperity formation and shadowing effects and (b) example of atomic reassembly paths on a surface and vacancy annihilation in the subsurface lattice atoms. "A" jump corresponds to an "on terrace" hop, "B" is a jump "from a ledge of terrace," "C" is a jump "from a dimer," "D" is a jump "over a terrace ledge," and "E" is a "vacancy annihilation" jump [43].

The crystal configurations resulting from deposition at low growth temperatures have very large surface areas (and therefore a high surface energy) and contain many (total energy increasing) lattice defects. If kinetically permitted, the system will evolve to a lower energy state where more of the atoms occupy highly coordinated lattice sites. This occurs by atomic jumps from sites of relatively higher energy (low coordination) to sites of lower energy. Fig. 4.1(b) shows examples of these jumps. A-type jumps allow atoms to diffuse to step edges (ledges) and extend the terrace, and D-type jumps may also allow atoms on terraces to diffuse through or over ledges to extend the terrace. E-type jumps, which fill vacancies in the interior of the crystal, can also occur depending on the
mobility of atoms around the vacancies. For elemental systems with isotropic surface energies, the lowest energy crystal configuration is a flat surface because it has the smallest surface area. This requires significant surface mobility, which in turn depends on the substrate temperature and the deposition rate. These determine the jump frequency and the time available for an atom to jump on the surface before it is buried (frozen) into a bulk lattice structure.

The energy barriers for atomic jumps are dependent upon the local atomic environment [174]. For instance, jumps on different crystallographic (but flat) surfaces, along a ledge, away from a ledge, down from a terrace, or up to a high terrace (see Fig. 4.1) all have different energy barriers impeding their jumps. In multilayers, the variation of the local composition in the vicinity of the jump path can significantly change the energy barriers [34]. As a result, thermally-activated evolution of a surface is essentially a process where a large number of surface atom jumps occur continuously, each with significantly different jump frequencies. During the growth of multilayered systems with nonzero solubility of the constituent metals, the mixing due to the thermally-activated mode of atomic assembly is reduced by depositing the multilayers at a low substrate temperature and high deposition rate [8]. However, such conditions then result in a rough surface. The ideal atomic assembly environment would enable high surface mobility with little or no atom transport vertically through the films. Assisting inert gas ions have the potential to facilitate this, but the detailed mechanistic insights need to be established.

In this chapter, we utilize an MD simulation approach to explore in detail the effects of inert gas ion mass and energy upon atomic assembly mechanisms during low energy
ion beam assisted vapor deposition of a metal multilayer system. We focus upon a model Co/Cu system, which has been the subject of extensive experimental study [5, 62], and restrict our study to ions that arrive perpendicular to the surface. We show that these assisting ions can have strong effects on the atomic scale reassembly processes on a surface. For each ion species (mass), an ion energy range is shown to exist where optimum multilayer interface properties can be achieved. We also show that atomic assembly mechanisms are highly sensitive to the ion mass, resulting in different film structures. We have quantified these phenomena to provide theoretical guidance to developers of practical deposition processes.

4.2 Simulation Methodology

4.2.1 MD Simulation

Details of MD simulations can be found in Chapter 3 and the Fortran code is appended (Appendix B). Briefly, a computational crystal is created by assigning the positions of atoms to an assembly of lattice sites. An interatomic potential is then used to calculate the forces between atoms, and Newton’s equations of motion are used to calculate the velocities and positions of all atoms. This approach dynamically solves for atom vibration around the occupied lattice sites. It also naturally incorporates the energy barriers for the various jump paths and correctly simulates the thermally-activated jumps of the vibrating surface atoms from one local energy minimum site to another. The various hyperthermal impact effects can also be accurately represented by Newton’s
equations of motion. Hence, MD captures all of the complicated atomic assembly phenomena described above.

The model system analyzed consisted of a Cu substrate made up of seventy-two (2 2 4) planes in the x direction, eight (1 1 1) planes in the y (growth) direction, and forty-two (2 0 0) planes in the z direction. To minimize the effect of the small crystal size, periodic boundary conditions were used in both the x and z directions so that the simulated surface was extended infinitely in these two directions. A free boundary condition was used for the y surface. To avoid a crystal shift due to the ion impacts with the top surface, the coordinates of the atoms in the bottom two (111) Cu layers were fixed during simulation. The crystal was kept at a constant film temperature of 300 K by applying damping forces to a region below the surface [30]. Inert gas ions of prescribed ion species and energy were randomly injected and vertically impacted the top y surface.

The numerical scheme of MD simulations requires the use of short time step ($10^{-15}$ sec.) to solve for the dynamics of lattice vibration. As a result, MD can only simulate a very short ($10^{-9} - 10^{-8}$ sec.) process. High ion/adatom fluxes must be used in order to simulate a sufficient number of ion/adatom impacts within the available computational time. For the ion impact simulations on the model rough surfaces, a fixed ion fluence of 0.5 ions/$\AA^2$ was used. For growth simulations, a fixed deposition rate of 10 nm/ns was used. Under these high rates of ion/adatom impacts, the time for the thermal diffusion of atoms on a free surface was significantly shortened. Nonetheless, the simulated conditions were carefully chosen to ensure that the time interval between ion or adatom arrival was at least 0.15 ps or above. This time interval allows the impact induced atomic
assembly to be realistically captured and should allow the effects of assisting ions to be accurately predicted.

The energy barrier for Co surface diffusion in a Co-rich environment was estimated to be approximately 0.1 eV. This is a relatively low energy barrier and surface diffusion is likely to occur given the diffusion time commonly encountered in experiments even at a relatively low deposition temperature of 300 K [149]. The shortened diffusion time used in the simulations did not allow such surface diffusion. It should be noted that under equilibrium conditions, Co tends to form clusters on a Cu surface because the binding between Co atoms is stronger than that between Co and Cu atoms. As a result, the surface diffusion of Co atoms on a Cu surface is likely to roughen the surface. Because of these, the neglect of the surface diffusion should not affect our analysis of the flattening effects of continuous assisting ion impacts on the Co-on-Cu growth surfaces.

It should be noted here that effects of strains in the film, roles of surface energy, polycrystalline grains and alloying all affect system equilibrium or the lowest energy state when considering different material systems. Therefore, some other systems could have different situation in smoothing interface because the sharpest interfaces and the smoothest surfaces sometimes are far away from the equilibrium.

4.2.2 Interatomic Potentials

MD simulations yield reliable results only when high fidelity interatomic potentials are used to calculate the interatomic forces. For closely packed metals such as the Cu and Co studied here, the embedded atom method (EAM) potential initially proposed by
Baskes and Daw [152] meets this requirement [28, 30]. EAM potentials use a pairwise energy term, and a term describing the binding energy associated with embedding an atom into a local electron density background. Introduction of the embedding energy term has been justified using ab initio theories [152]. By fitting to physical constants, it can capture the many-body effects of the atomic interactions. It also distinguishes the physical differences between bonding at a surface from bonding in the bulk. As a result, EAM is well suited for the study of surface problems, such as those encountered during vapor deposition on ion bombardment [28, 30].

The EAM potentials developed for elemental metals cannot be directly used to study alloys [150]. Alloy interactions are important and must be incorporated. Here we use an alloy EAM model that incorporates interactions between Cu and Co atoms [28, 150].

In previous simulations of low energy inert gas bombardment of a metal surface, interactions between inert gas ions and metal atoms were found to be well represented by a universal pair potential derived from experimental ion bombardment data [175]. This universal pair potential was used here to calculate the ion-ion and ion-metal interactions.

4.2.3 Molecular statics analysis

Many kinetic processes at the atomic scale are determined by the activation energy barriers for atoms to jump from one site to another. As mentioned in Chapter 3, MS has been used here to calculate the energy barriers of these surface atom assembly mechanisms. In this approach, a selected atom is incrementally moved in the jump
direction and the relaxed crystal energy is calculated by a conjugate gradient energy minimization procedure. During calculation, the jumping atom is only constrained to move in the plane perpendicular to the moving direction. In this way, the trajectory of the jumping atom is not forced to be a straight line and would be similar to that occurring during real surface diffusion. The energy versus step distance relation can then be obtained and the energy barrier for a jump path determined. The calculations were conducted at 0 K using the code appended in Appendix B.

4.3 Results and Discussion

4.3.1 Roughness and Intermixing at Interfaces

A Co-on-Cu interface (rather than Cu-on-Co) was chosen for this study because previous work indicated that this surface is more difficult to flatten without mixing [171]. To understand the mechanisms of the ion beam assisted deposition process at multilayer interfaces, Co clusters on a Cu film were used to imitate a rough surface created during the early deposition of cobalt on a copper film. Fig. 4.2 shows the simulated system. It consisted of twelve uniformly separated Co pyramids on the Cu surface. Each three-layer pyramid contained 10 Co atoms: 6 on the base layer, 3 on the mid layer, and a single atom on the peak. Various inert gas ion species (He\(^+\), Ne\(^+\), Ar\(^+\), and Xe\(^+\)) and ion impacting energies ranging from 0–25 eV were explored. For each ion species and energy, a total of 1500 ion impacts were simulated.

The model described in Fig. 4.2 assumes a relatively small amount of the surface asperities. It should be noted that the effects of ion impacts are sensitive to the initial
sizes of surface clusters. In general, higher ion energies are required for flattening bigger clusters [176]. We used small cluster size for the analysis because we intended to understand the effects of simultaneous ion impacts and adatom deposition within the limit of a high ion to adatom ratio where any nucleated roughness embryos are flattened promptly by the ions.

Figure 4.2 A model rough surface structure: Co cluster on Cu film surface.

The effect of ion energy upon the atomic scale structure of the model system after bombardment with 1500 Xe ions can be seen in Fig. 4.3. At a low ion energy of 2 eV, Fig. 4.3(a) shows that a relatively large fraction of Co atoms remained in a second Co monolayer or more above the Cu surface. This occurred even though most of the Co atoms had experienced several Xe ion impacts. The fraction of the Co atoms remaining in or above the second monolayer was reduced as the ion energy was increased to 4 eV, Fig. 4.3(b). It is also evident that below this energy, almost no intermixing occurred in the top copper surface layer. Increasing the ion energy to 6 eV resulted in fully flattened Co
clusters – all the Co atoms were distributed in a partial monolayer on the Cu surface, Fig. 4.3(c). Some Co atoms had penetrated into the topmost copper layer and four copper atoms can be seen sitting above this layer within the partial Co monolayer. This intermixing became more significant as the ion energy was increased to 10 eV, Fig. 4.3(d). In addition to the decrease in the vertical surface roughness and the increase in intermixing, Fig. 4.3 (a)-(d) indicate that increases in the ion energy also result in a coalescence of some of the initially separated Co islands. Ion impacts clearly promote lateral diffusion of Co atoms.
Figure 4.3 Surface morphology after 1500 Xe$^+$ ion impacts. (a) Xe$^+$ energy 2 eV; (b) Xe$^+$ energy 4 eV; (c) Xe$^+$ energy 6 eV; and (d) Xe$^+$ energy 10 eV.

A normalized roughness can be calculated as the fraction of Co atoms remaining on more than one monolayer above the copper surface. Likewise an intermixing parameter can be defined as the fraction of Co atoms present in the first layer of copper. The effects
of ion energy and ion species on these roughness and intermixing metrics are shown in Fig. 4.4 (a) and (b) for the four ion species studied. The ion species (mass) clearly has a significant effect on the ion energy needed to achieve smoothing without mixing. For each ion species, an optimum energy can be identified where both roughness and intermixing are minimized. For Xe, this energy is about 5 eV, which increases as the ion mass is reduced. For He, this optimum energy is around 18 eV.
Figure 4.4 Energy and species effects on (a) interfacial roughness parameter (fraction of atoms remaining above the first layer after ion bombardment) and (b) interfacial mixing parameter (fraction of mixing atoms after ion bombardment). Solid lines are fitted curves.
4.3.2 Atomic Reassembly Mechanisms

a) Roughness

The final atomic configurations shown in Fig. 4.3 represent the accumulation of many atomic assembly events. To gain insights into the most significant of these mechanisms, many impacts were examined using time resolved “snapshots” of the atomic configurations.

When the ion energy was very low, the assisting ions did not have enough energy to reconstruct the surface atoms and our study indicated that a simple recoil mechanism was the dominant mechanism of interaction. Fig. 4.5 shows an example of this recoil mechanism for a 1 eV Argon ion impact with a Co island. It shows that when a low energy ion impacts the top of the island, the binding interaction within the cluster was sufficiently strong that the island was able to retain its original shape until the Ar ion was recoiled. Careful inspection shows that when the ion first contacted the surface island (at about 0.4 ps), the island responded by changing its shape, Fig. 4.5 (b). The deformation, however, was elastic and at 0.5 ps the atoms began to bounce back, Fig. 4.5 (c). This resulted in the recoil of the Ar ion, Fig. 4.5 (d). The energy transferred to the island from the inert ion was calculated from the energy difference of the incoming and the recoiled ion. Approximately 200 meV (20% of the incident energy) was transferred to the island from the incoming ion. This energy was clearly insufficient to overcome the energy barriers impeding reassembly of the cluster. This recoil mechanism was also the dominant mechanism operative during low ion energy bombardment with all the other ions.
Figure 4.5 Recoil atomic assembly mechanism at an ion energy of 1 eV. (a) Original configuration at time t = 0 ps, (b) t = 0.4 ps, (c) t = 0.5 ps, and (d) t = 0.8 ps.

As the ion energy was increased, we observed an atomic exchange flattening mechanism. This is illustrated using a time sequence shown in Fig. 4.6. In this simulation series, an argon ion with an ion energy of 8 eV made a normal impact with the top of a truncated Co pyramid. Several selected atoms are marked with an open circle, a square, and a diamond to enable their motions to be tracked. After impact, the “diamond” atom in the second layer has been “pushed” into the underlayer, yielding its original position to the “circle” atom in the top layer. This exchange can be more clearly seen at 0.3 ps, Fig. 4.6 (c), and at 0.5 ps, Fig. 4.6 (d). After completion of this first exchange (at 0.5 ps), a second exchange involving the “square” atom occurred between 0.5 ps and 0.7 ps, Fig. 4.6 (d) and (e). Fig. 4.6 (f) shows the atomic structure at 1 ps, where all of the top three
layer atoms had descended to the second layer by this exchange mechanism. The energy transferred from the inert ion was approximately 4.512 eV, 56.4% of the original ion energy. This is significantly larger than the previous case. Extensive analysis of the simulations indicates that this cluster atom exchange mechanism is the dominant assembly process in the intermediate ion energy region. It is favored over a direct downward hop mechanism because it has a relatively lower Ehrlich-Schwoebel barrier for jumping over the terrace edge.

Figure 4.6 Exchange flattening mechanism during an 8 eV Ar ion impact. Selected atoms are marked with an open circle, square, and diamond to show their motion.
As the ion energy was further increased to 12 eV, we found that a direct hop mechanism began to frequently occur in addition to the exchange mechanism. Fig. 4.7 shows a typical time sequence for an impact that activated the direct hop mechanism. Again, we use open circles, squares, and diamonds to trace selected atoms in the structure during the reassembly process. It can be seen from Fig. 4.7 (a)-(f) that the “circle” atom in the top layer directly hopped over a double layer Ehrlich-Schwoebel barrier and came to rest in the first Co layer on the Cu surface. This mechanism required about 1 ps to occur. However, as this was occurring, the “diamond” atom in the top layer also reassembled and was transferred to the second layer by the exchange mechanism. This process began at 0.2 ps and was completed about 0.3 ps later, Fig. 4.7 (c)-(e). About 72.25% (8.67 eV) of the original ion energy (12 eV) was found to have been transferred to the surface. The activation combination of these direct hop and exchange flattening mechanisms correlates with the rapid decrease of roughness as the ion energy was increased, Fig. 4.4 (a).
Fig. 4.8 shows the impact of an argon ion with an energy of 20 eV. In this case, all three of the top layer atoms (marked with an open circle, a square, and a diamond) were fully flattened. This occurred by the direct hop mechanism within 0.3 ps of impact, Fig. 4.8 (a)-(d). In addition, the atom marked with an open triangle in the second layer moved to the first layer by the exchange mechanism during 0.3-1.0 ps, Fig. 4.8 (d)-(f). This
eV impact transferred about 85% (17 eV) of its energy to the surface. This process was typical of that activated in the high energy tail of the roughness versus ion energy relationship in Fig. 4.4 (a).

Figure 4.8 Flattening mechanisms during a 20 eV ion impact. All three top layer atoms are flattened by the direct hop flattening mechanism. An exchange mechanism is also seen for the “triangle” atom in the second layer.

Further insights can be gained by calculating the energy barriers for several of the jump paths. Molecular statics results are shown in Table 4.1. The barrier for a single Co
atom to jump on a flat Co surface ("A" jump in Fig. 4.1) is about 0.103 eV. On a flat Cu surface, it falls to 0.084 eV, which is in good agreement with experimental values [177]. The energy barrier for Co atoms to break away from a Co surface dimer (a "C" jump) and monolayer islands (a "B" jump) are larger, 0.322 eV and 1.887 eV respectively, because these jumps result in a reduction in nearest neighbor coordination. If we label "H₁" and "E₁," as the jumping of a top layer atom by either a direct hop or exchange flattening mechanisms, respectively, and "H₂" and "E₂" are similar jumps for second layer atoms, then Table 4.1 shows that the energy barriers for the exchange mechanisms ("E₁" and "E₂" jumps) are always lower than those for the direct hop mechanisms ("H₁" and "H₂" jumps). This explains why the exchange mechanism always appears first as the energy of an ion is increased. It accounts for our simulation observations that low energy ion beam bombardment is effective for flattening a Co on Cu surface. We also observed that Ehrlich-Schwoebel barriers for the top Co atom are generally smaller than those of Co atoms in the second monolayer of a pyramid cluster because they are less highly coordinated. These energy barriers and estimated transferred ion energies then account well for the results reported in Fig. 4.5-4.8. They show that the mechanism of reassembly can be "tuned" by selection of ion energy.
Table 4.1 Energy barriers for various jumps

<table>
<thead>
<tr>
<th>Jump Path</th>
<th>Energy Barriers (eV)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (Co on Co)</td>
<td>0.103</td>
</tr>
<tr>
<td>A (Co on Cu)</td>
<td>0.084</td>
</tr>
<tr>
<td>B</td>
<td>1.887</td>
</tr>
<tr>
<td>C</td>
<td>0.322</td>
</tr>
<tr>
<td>H&lt;sub&gt;1&lt;/sub&gt;</td>
<td>0.59</td>
</tr>
<tr>
<td>E&lt;sub&gt;1&lt;/sub&gt;</td>
<td>0.29</td>
</tr>
<tr>
<td>H&lt;sub&gt;2&lt;/sub&gt;</td>
<td>1.853</td>
</tr>
<tr>
<td>E&lt;sub&gt;2&lt;/sub&gt;</td>
<td>0.622</td>
</tr>
</tbody>
</table>

b) Intermixing

The intermixing of Co and Cu during an ion impact was found to be caused by an exchange mechanism between Co atoms and underlying Cu atoms. Molecular statics analysis of the exchange between a Co surface atom and an underlayer Cu atom on a flat surface was found to have a high energy barrier of about 3.24 eV, consistent with the significant mixing only at high ion energies, Fig. 4.4 (b). However, Fig. 4.4 (b) also indicates a small amount of mixing at significantly lower energies. To explore this, we simulated ion impacts with Co atoms located near surface defects. Fig. 4.9 shows argon ion impacts with a Co adatom sitting at different surface sites: (a) at the edge of the terrace, (b) at one row away from a terrace edge, (c) at two rows away from the terrace edge, (d) at four rows away from terrace edge, and (e) on a perfect terrace. The ion energies used in Fig. 4.9 are close to the minimum energies found to result in mixing for each of the different impact sites. Fig. 4.9 (a) and (b) clearly show that as little as 4 eV ion energy is needed to cause intermixing if the exchanging atoms are located near low atomic coordination sites, such as the edge of a terrace. A higher energy is required to cause mixing as the impact site moves away from the terrace edge, Fig. 4.9 (c)-(e). If an
impact occurs more than four rows away from the terrace edge, Fig. 4.9 (d), the ion energy required to cause the mixing is essentially the same as that for a flat surface, Fig. 4.9 (e). These "defects" provide a low resistance mechanism for the large dilations that must be accommodated during this exchange mixing process. These insights suggest that very high energy impacts would be needed to cause lattice interior exchange. Indeed, we find that at least 40 eV argon ion impacts are required to induce mixing between the second and third planes below a flat copper surface. The use of ion bombardment at the completion of deposition of a Cu layer to minimize surface defects would cause the average ion energy necessary to cause mixing to increase.
Figure 4.9 Comparison of intermixing mechanism at different ion impact sites. (a) On the edge of a terrace, (b) one row away from the edge, (c) two rows away from the edge, (d) four rows away from the edge, and (e) on the infinity terrace. Energies for the onset of intermixing for each case are used.
4.4 Collision Model

4.4.1 Roughness

The functional dependence of flattening and mixing upon ion energy and mass can be well captured by a simple two-body collision theory. If we assume that the incident ion energy is \( E_0 \), then the energy transferred to the film, \( E_T \), can be approximated from the two-body collision theory as

\[
E_T = \frac{4M_1M_2}{M_1 + M_2} E_0 \left( \sin \frac{\theta}{2} \right)^2 ,
\]

where \( M_1 \) is the mass of the impacting ion, \( M_2 \) is an effective total mass of surface atoms that interact with the ion, and \( \theta \) is the ion scattering angle. If we consider only the normal ion incident angle and assume that the ion is back scattered, the scattering angle can be taken as 180°. The transferred energy then becomes

\[
E_T = \frac{4M_1M_2}{M_1 + M_2} E_0 .
\]

An atom jumps from one local energy minimum site to another only when it acquires a kinetic energy that is higher than the energy barrier impeding movement along the transition path. Hence, a flattening mechanism can only begin when the ion energy transferred to the surface significantly exceeds the energy barrier for the corresponding flattening mechanisms. With a further increase of energy, additional flattening mechanisms can be activated and the roughness continuously decreases.

Suppose that the incremental decrease in roughness following \( dN \) ion impacts is denoted \( dR \). \( dR \) can be approximated as

\[
dR = -R P_1 P_2 \exp \left( \frac{-E_b}{E_T + kT_{\text{room}}} \right) dN ,
\]

where \( E_b \) is the energy barrier of the flattening mechanism, \( E_T \) is the transferred energy, \( N \) is the
number of impacting ions, $P_1$ is the probability of an ion impacting a surface asperity, and $P_2$ is the probability that an adatom jumps to another site after receiving enough energy to overcome flattening mechanism barriers. $P_1$ is a function of surface asperity area that is proportional to $\sqrt[3]{R^2}$, and $P_2$ is a function of surface atom jumping direction and ion impacting direction.

If we assume that the impact position is random and the impact direction is perpendicular to the surface, $P_2$ can be taken as a constant. Integration of the equation above with the condition that the roughness goes to zero when $E_T$ is large and has a value of unity when $E_T$ is zero gives an expression for roughness:

$$\frac{2}{3} R = A \left[ \exp \left( -\frac{E_b}{E_T + kT_{room}} \right) - \exp \left( -\frac{E_b}{kT_{room}} \right) \right] + 1. \quad (4-1)$$

Then parameter $A$ can be obtained by fitting to the simulated data. The curves defined in this relationship are plotted as the solid curves in Fig. 4.4 (a). They predict a trend that is consistent with the simulated data.

4.4.2 Intermixing

A mixing function can be derived in a similar way. Suppose that the number of unmixed Co atoms decreases by $dN_{CoUp} = -aN_{CoUp} \exp \left( -\frac{E_b}{E_T + kT_{room}} \right) dN$ during $dN$ ion impacts, where $N_{CoUp}$ is the number of Co atoms that remained unmixed, $E_b$ is the energy barrier for the exchange mixing mechanism, and $a$ is a coefficient that is the function of the exchange probability between layers at high ion energies. If coefficient, $a$, is
Independent of other parameters, then integration of the equation above with the condition that $N_{CoUp}$ is 120 at zero $E_T$ and roughly 56 (the lowest number in our simulations even under very high ion energies) at large $E_T$ gives the number of exchanged atoms as:

$$N_{CoUp} = \exp \left[ 0.7 \exp \left( \frac{-E_b}{E_T + kT_{room}} \right) + 4.8 \right]$$  \hspace{1cm} (4 - 2)

An intermixing parameter, IM, can then be calculated as $IM = 1 - \frac{N_{CoUp}}{N_{Co}}$, where $N_{Co}$ is the total number of Co atoms in the system.

As has been described above, the ion impact with a surface can be thought of as the interaction between the ion and a group of surface atoms with an effective total mass of $M_2$. Because the energy required for flattening generally differs from that for mixing, atoms that are active in the flattening may not coincide with those that are active in the mixing. As a result, the effective mass used in the flattening analysis can be assumed to be different from the one used in the mixing analysis. The effective mass varies as a function of incident ions. Heavy ions can transfer more impact energy to the surface. As a result, they can displace more surface atoms, resulting in a larger effective total mass. By fitting to MD results, the effective masses were found to be $4M_{Cu}$, $8M_{Cu}$, $12M_{Cu}$, and $20M_{Cu}$ for He, Ne, Ar, and Xe impacting ions, respectively. With these effective masses, the intermixing curves calculated using Eq. (2) are shown in Fig. 4.4 (b) as solid lines. They are seen to match well with the simulated data.
4.5 Multilayer Growth Simulations

Based on the results shown in Fig. 4.4, growth conditions for producing the multilayer with a low combination of interface roughness and intermixing can be easily determined. We found that for Ar ions, an ion energy of 6 eV produces near optimized film structures. Simulations of the growth of a Cu/Co/Cu multilayer with and without 6 eV ion assistance are shown in Fig. 4.10 (a) and (b). Clearly, without ion assistance, Fig. 4.10 (a), the film contains numerous defects such as voids, rough interfaces, and even a pinhole. On the other hand, low energy ion assisted growth of a Cu/Co/Cu trilayer structure using a near optimized argon ion energy resulted in a much improved structure, Fig. 4.10 (b).
Figure 4.10 Comparison of simulated Cu/Co/Cu trilayer structures deposited without and with ion assistance. (a) Deposition without ion assistance; (b) 6 eV Ar ion assistance.
4.6 Conclusions

MD simulations have been used to study the effects and mechanisms of low energy ion assistance on the interface structures of Cu/Co multilayers. The simulations have revealed that

1. Ion assistance with ion energy of less than 30 eV has significant effects on interface structures in the Co-on-Cu system. Increasing the ion energy results in a reduction in surface roughness. However, high ion energies/ion masses cause significant intermixing.

2. Studies of the atomic assembly mechanisms indicate that flattening of Co clusters on a Cu surface occur by exchange flattening mechanisms at relatively low energies and by a direct hop mechanism at higher energies.

3. An exchange mechanism of mixing at Co on Cu interfaces has been identified. Its activation is a sensitive function of the surrounding lattice perfection. The impact energy for an exchange event decreases near low coordinated surface sites, such as terrace edges.

4. The functional dependency of flattening and mixing upon ion energy and ion mass is well described by a simple two-body collision model.

5. For normal incidence, the optimal ion energy that flattens the Co cluster on a Co surface without significant intermixing is approximately 5 eV for Xe ions and rises to 18 eV for He ions.
CHAPTER 5. ION FLUENCE AND INCIDENCE ANGLE EFFECTS

Inert gas ion impacts can be used to manipulate atomic assembly processes during the growth of metallic superlattices, but the detailed mechanisms are not well understood. In this chapter, MD simulations are further used to investigate the effects of ion incident angle and fluence upon the reassembly and structure of a copper surface partially covered with cobalt asperities. In the low ion energy regime, increasing the ion fluence decreases the cobalt layer surface roughness while gradually leading to an increase in the degree of interfacial mixing. The flattening of asperities occurs by direct (athermal) ion activation of an Ehrlich-Schwoebel mechanism of atom jumping. Intermixing of cobalt surface atoms in an underlying copper layer is found to occur by a knock-on process, and the lowest energy barriers for this occur in low-index <110> and <112> crystal (channeling) directions. The mechanistic insights gained from the study are used to simulate the ion assisted growth of a Cu/Co/Cu multilayer system. Using ion parameters chosen to selectively activate atomic assembly mechanisms that promote flat, unmixed interfacial structures, it is shown that Cu/Co/Cu multilayer structures with high quality, smooth, and chemically sharp interfaces can be obtained by using oblique, low energy, moderate fluence ion assistance with an ion mass that is similar to the atomic mass of the metals.

5.1 Introduction

As described in previous chapters, ideal structures are thermodynamically unstable and experimentally difficult to make [128, 129]. Lattice mismatch strain energy and configurational entropy in epitaxially grown films drive inter-diffusion between the
layers [178, 179]. High interfacial energy can also cause topology breakdown (spheroidization) [180]. These effects are experimentally avoided by kinetically constraining the deposition process. Unfortunately, the metallic multilayers grown in this way contain bulk lattice defects and significant interfacial imperfections [129, 181, 182]. Efforts to improve structural perfection by increasing the growth temperature or by decreasing the deposition rate are then severely constrained by the thermodynamically driven processes discussed above [181, 182].

Sputtering processes create energetic metal atoms, and film assembly on a surface is then largely controlled by the adatom energy [30]. When strong plasma field strengths and low pressure sputtering conditions are used, the sputtered metal atoms have high (10-20 eV) kinetic energies when they impact on a growth substrate. Smooth interfaces but with significant interfacial mixing have been grown using these conditions [28]. The best films are grown using sputtering processes where the incident atom energy is in the 1-5 eV range. In this regime, flattening occurs but the intermixing rates at interfaces remain relatively low [28, 30].

Sputtering and other ion assisted growth processes also enable the use of ion fluxes to provide a "non-kinetic" method for manipulating atomic scale structures during the growth of thin films [183, 184]. While a large body of experimental literature relates ion parameters such as ion mass, energy, and flux to monolithic film defect structures and properties [166], relatively few experimental or theoretical studies have addressed the atomic scale control of interfacial structures using this approach.
Several experimental studies have indicated that very low energy ion assistance, with ion energies in the 10-60 eV range, can significantly improve the optical and electron transport properties of some multilayer systems [156, 169, 170]. Carefully tailored ion assistance approaches therefore appear necessary for the control of interfacial structures in multilayer systems where the layers are only a few atomic layers thick. This is experimentally difficult to explore because there are few methods for creating controlled ion fluxes with very low energies and it is not trivial to conduct the atomic scale characterization required to determine the outcome of the growth process.

MD simulations of low energy inert gas ion impacts with a metal surface provide an alternative approach to tailoring ion assistance strategies [43]. Chapter 4 has shown that under normal incidence angle ion impact conditions, the ion mass and kinetic energy significantly influence the atomic reassembly of a system consisting of model Co asperities on a flat (111) Cu surface. These simulations indicate that atom recoil, atom exchange across interfaces, and direct (non-thermal) Ehrlich-Schwoebel barrier jumps could be activated by ion impacts. Simulations also indicate that very low energy ions could potentially be used to flatten an interface without causing interfacial mixing. The ion mass is found to have a significant effect upon both interface flattening and intermixing mechanisms. This arises because the ion mass directly controls momentum transfer to surface atoms and, therefore, the probability of initiating various atom movements as well. In fact, the relationships between interfacial structure, ion mass, and ion energy deduced from these simulations were found to be remarkably well approximated by a simple momentum transfer model [43].
During the ion assisted vapor deposition of metals, other ion parameters, such as the ion incidence angle and the ion fluence, must also affect the atomic scale surface reconstruction processes [28]. However, the relationships between these parameters and the resulting structures of interfaces have not been well resolved in the low ion energy assistance regime. Here, Chapter 4’s MD simulation approach is extended to investigate the effect of ion fluence and ion incidence angle upon the reconstruction of a model Co on Cu interface. The study again focuses on the atomic assembly mechanisms activated by the impact of ions with a wide range of ion masses. It is found that the incidence angle has a strong effect on the in-plane migration of surface atoms. We show that desirable atomic assembly mechanisms can be selected by appropriate choices of ion fluence and incident angle in the low ion energy regime using ions with an atomic mass similar to that of the metals. The results are used to guide the design of a simulated low energy ion assisted growth process. This tailored process is shown to result in a Cu/Co/Cu metallic superlattice with planar, chemically sharp interfaces and few lattice defects.

5.2 Simulation Methodology

The MD simulation methodology used in this chapter was identical to that utilized in Chapter 4 to study the effects of ion energy and mass upon the morphology of a model Co on Cu surface (details can be seen in Appendix A and B). Briefly, an embedded atom method potential database [28, 150] was used to calculate the interatomic forces between Cu and Co atoms. A universal pair potential was used to define the interatomic forces involving the inert gas (He, Ne, Ar, and Xe) atoms [175]. A copper face-centered-cubic
single crystal substrate containing seventy-two \( (22\bar{4}) \) planes in the \( x \)-direction, eight \( (111) \) planes in the \( y \)-(growth) direction, and forty-two \( (220) \) planes in the \( z \)-direction was created. Periodic boundary conditions were used in both the \( x \)- and \( z \)-directions to minimize the effects of small crystal size in these two directions. Inert gas ions with an incidence angle, \( \theta \), to the \( [1 \bar{1} 1] \) direction in the \( (1 \bar{1} 0) \) plane and with a prescribed initial kinetic energy, \( E_i \), were injected toward the surface, Fig. 5.1. To avoid a crystal shift during ion impact, the positions of the atoms in the bottom two \( (111) \) Cu layers were fixed during the simulations. To simulate the isothermal conditions present during experimental ion bombardment, the atoms in the two Cu monolayers above the fixed region were maintained at a constant temperature of 300 K. A sufficient time interval (at least 0.15 ps) between impacts was then used to enable full atom relaxation and dissipation of the ion impact energy so that the mechanisms activated by each impact could be considered as approximately independent.
Figure 5.1 A model (111) Cu crystal surface partially coated with twelve, 10-atom Co clusters arranged in the form of three monolayer high pyramids has been used to investigate the effect of ion fluence and incident angle upon the atomic reassembly processes. An incidence angle, $\theta$, is defined as the deviation from the $[\bar{1} \bar{1} 1]$ direction in the $(\bar{1} \bar{1} 0)$ plane.

The effects of the inert gas ion fluence and incident angle on interface roughness and chemical composition gradient were studied for a range of ion masses (from He to Xe). The effects of up to 3000 impacts with a surface (corresponding to a maximum fluence of 1.0 impacts/Å$^2$) were studied. Only Co clusters on a Cu surface were studied since the relatively low cohesive energy and the relatively large atomic size of Cu make the Co on Cu surface more difficult to flatten (without causing mixing) than the Cu on Co surface [28].
Chapter 5. Ion Fluence and Incidence Angle Effects

The Co clusters on the Cu surface model described above assume a relatively small size of the surface asperities, which was proven to be quite stable by calculating energy barriers for flattening [43]. It should be noted that the effects of ion impacts are sensitive to the initial sizes of surface clusters. In general, higher ion energies are required for flattening bigger clusters [176]. In this chapter, we again used only a small cluster size for the analysis because we intended to understand the effects of simultaneous ion impacts and adatom deposition where any nucleated roughness embryos are flattened promptly by the ions before growing too big.

5.3 Ion Fluence Effects

The atomic configurations of the model crystal after Ar ion bombardment of various fluences are shown in Fig. 5.2. The ion energy was 10.0 eV and the ion incident angle was normal to the surface (θ=0°). It can be seen that both cobalt surface roughness and interface mixing at the Co on Cu interface were highly dependent upon the ion fluence. At the lowest ion fluence of 0.01 ions/Å² corresponding to one ion impact per ten Co atom cluster, Fig. 5.2 (a), two of the Co pyramidal clusters were reduced to either one or two layer thick Co islands. However, the majority of the Co clusters were not impacted by an ion and they remained unchanged. When a 10 eV argon ion did impact a Co cluster, partial or complete flattening with a low probability of intermixing at the cluster-copper interface was found to occur.
Figure 5.2 Atomic configurations of a Co-on-Cu surface following 10 eV Ar ion impacts at an ion incident angle of zero degrees. (a) Ion fluence of 0.01 ions/Å²; (b) ion fluence of 0.1 ions/Å²; (c) ion fluence of 0.2 ions/Å²; and (d) ion fluence of 1.0 ions/Å². Cut off distance of inert ion disappearance was increased in (d) in order to show high fluence of assisting ions.
Chapter 5. Ion Fluence and Incidence Angle Effects

As the ion fluence was increased to 0.1 ions/Å², all of the three-layer Co pyramidal clusters had been impacted by at least one ion and all had undergone some degree of flattening, Fig. 5.2(b). Approximately a half of the clusters had been reduced to single Co monolayer islands. Further increases of the ion fluence resulted in more cluster flattening with the majority of the islands fully flattened at a fluence of 0.2 ions/Å² (this fluence corresponds roughly to two argon impacts per surface cobalt atom), Fig. 5.2 (c). However, atomic intermixing (between the underlying Cu and Co atoms sitting in the first cobalt atom plane) began to occur. As the ion fluence was increased to 1.0 ions/Å², Fig. 5.2 (d), the extent of this intermixing increased significantly.

As usual, a roughness parameter, R, can be defined as the fraction of Co atoms remaining above the first Co island monolayer. Likewise, an intermixing parameter, M, can be defined as the fraction of Co atoms present in the underlying Cu crystal. By these definitions, the roughness parameter, R, takes a value of unity for the initial pyramidal cluster structure and falls to zero when all the clusters are fully reduced to one monolayer thick islands. Likewise, the mixing parameter, M, takes a value of zero for the initial structure and a value of unity for the maximum possible amount of mixing.

The roughness and intermixing parameters were calculated as a function of ion fluence for a variety of inert gas ion species (He, Ne, Ar, and Xe) using a normal incidence angle (θ=0°) and a range of ion energies. It can be seen that at the relatively low ion energy of 3 eV, the roughness parameter fell with increasing ion fluence for all ion species, Fig. 5.3 (a). Reductions in ion mass resulted in a lower rate of flattening with fluence, consistent with earlier observations in Chapter 4. For example, the lightest He
ions caused only a minor roughness reduction over the ion fluence range studied in Fig. 5.3 (a). In comparison, the heaviest Xe ions with identical energy caused the roughness to drop to nearly zero at a fluence of only 0.4 ions/Å².

Figure 5.3 (a) Surface roughness parameter as a function of ion fluence using different ion species at a fixed ion energy of 3.0 eV and a fixed ion incident angle of zero degree, and (b) intermixing parameter at a fixed ion incident energy of 15.0 eV and an ion incident angle of zero degree. Solid lines in the figure are the predictions of a two-body collision model.
Chapter 5. Ion Fluence and Incidence Angle Effects

This observation of ion mass effect is consistent with the increased effectiveness of ion energy transfer to the clusters, thereby enabling their reorganization to flatten (more coordinated) structures. Fig. 5.3 (a) also shows that the reduction of surface roughness with ion fluence (for high ion masses) was initially rapid but then became slower, especially at ion fluences above 0.4 ions/Å². Although the rate at which the roughness decreased depended on the ion mass, the roughness appeared to eventually approach zero for all the ion masses.

Earlier studies of the effects of ion energy upon cluster reassembly processes have indicated that intermixing occurs when the energy of normally incident ions exceeds a (ion mass dependent) threshold energy for mixing [43]. These energy thresholds were approximately 4 eV for Xe, 6 eV for Ar, 8 eV for Ne, and 17 eV for He according to the study in Chapter 4. It can be seen from Fig. 5.3 (b) that for those ion species whose threshold energy for mixing was below the studied energy of 15 eV, mixing initially increased monotonically with fluence and then saturated at high fluences. The saturation intermixing value also depended on the ion mass: Higher saturation intermixing occurred for heavier ion species. Note that the mixing threshold energy for He is 17 eV and therefore 15 eV ion impacts failed to induce any intermixing regardless of the ion fluence investigated here.

A collision theory model has been used to interpret ion energy and ion mass effects upon roughness and mixing for the model Co/Cu system [43]. This model can also be used to interpret the effects of ion fluence and incident angle upon roughness and intermixing. Suppose that the roughness is changed by an amount, \(dR\), due to ion
irradiation to a small fluence, \( dF \). By estimating the energy transfer from an ion to the lattice, it can be shown (see Chapter 4):

\[
dR = -R \rho P_1 P_2 \exp \left( \frac{-E_f}{E_T + kT} \right) dF,
\]

where \( R \) is the roughness parameter, \( E_f \) is an energy barrier for flattening, \( k \) is Boltzmann’s constant, \( T \) refers to the temperature, \( F \) is the inert gas ion fluence, \( E_T \) is the transferred energy from the inert ion, \( P_1 \) is the probability of an ion impacting a cluster (this is proportional to \( R^{23} \)), and \( P_2 \) is the probability that an adatom jumps from one site to another after receiving sufficient energy to overcome the intervening energy barriers. \( P_2 \) probability is therefore a function of the atom jumping and the ion impacting directions.

We have simulated the behavior of a large number of impacts with random surface impact positions and find that \( P_2 \) can be reasonably approximated by an average value for the surfaces studied here. Integrating equation (1) with the condition that \( R = 1 \), at a fluence \( F = 0 \) and \( R \) approaches to zero at large fluence \( F \), gives an expression relating roughness to ion fluence:

\[
R(F) = \left[ A \exp \left( \frac{-E_f}{E_T + kT} \right) F + 1 \right]^{\frac{3}{2}},
\]

where \( A \) is a constant of integration.

In Eq. (5-2), the transferred energy from an ion impact, \( E_T \), can be calculated using the collision model [43]:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Chapter 5. Ion Fluence and Incidence Angle Effects

\[ E_r = \frac{4M_1 M_2}{M_1 + M_2} E_0 \left( \sin \frac{\varphi}{2} \right)^2, \]  

(5-3)

where \( M_1 \) is the ion mass, \( M_2 \) is an effective mass for the group of atoms that interact with the ion [43], and \( \varphi \) is the ion scattering angle in the center of the mass coordinate system.

The average energy transfer for the many random scattering events that were simulated can be obtained by assuming that the \( \left( \sin \frac{\varphi}{2} \right)^2 \) term in Eq. (5-3) is a constant.

The resulting roughness-fluence predictions of Eq. (5-2) are shown as solid lines in Fig. 5.3 (a) with the constant \( A \) fitted to the simulation data. It can be seen that the model reproduces the ion mass and fluence trends seen in simulations. This good agreement indicates that energy transfer controls surface structure modification during ion assistance. Note that as the fluence became large, \( R(F) \) tends to approach zero as seen in the simulations.

An analogous expression for intermixing as a function of ion fluence has also been developed. Suppose that the number of cobalt atoms on a copper surface is decreased by \( dN \) during exposure to an ion fluence, \( dF \). Then:

\[ dN = -CN \exp \left( \frac{-E_m}{E_r + kT} \right) dF, \]  

(5-4)

where \( N \) is the number of cobalt atoms remaining on the copper surface, \( E_m \) is an energy barrier for mixing, and \( C \) is a coefficient that is dependent upon the exchange probability between layers.
Integration of Eq. (5-4) with the initial condition that $N$ is equal to the total number of cobalt atoms in the system, $N_{Co}$, gives an expression relating $N$ to ion fluence,

$$N(F) = N_{Co} \exp \left[ -C \exp \left( \frac{-E_m}{E_F + kT} \right) F \right]. \quad (5-5)$$

The mixing parameter, $M$, is defined as $M = 1 - \frac{N}{N_{Co}}$. By fitting $M$ to the simulated mixing data, the parameter $C$ can be determined and the mixing parameter can then be calculated as a function of ion fluence. The results are shown by the solid lines in Fig. 5.3 (b). The model predictions are again generally consistent with the MD simulation results. Higher fluence always gives more possibilities of atomic exchange between layers activated by ion impacts. However, as ion fluence further increases, interlayer exchange is reduced as the number of Co atoms on the Cu surface is reduced. This results in an ion mass dependent plateau in the mixing versus fluence curves as shown in Fig. 5.3 (b).

### 5.4 Incident Angle Effects

Earlier studies of low energy ion irradiation of interfaces using a normal ion incident angle indicated that flattening and mixing occur when the ion energy exceeds an ion mass dependent critical energy [43]. These critical ion energies correspond to transferred energies that are sufficient to enable direct (athermal) atom jumps over Ehrlich-Schwoebel barriers during flattening and to overcome atom exchange energy barriers during mixing. A critical ion energy for flattening can be defined as the lowest ion energy to completely flatten the surface (i.e., $R < 0.02$) after a designated high fluence of ion irradiation, 0.5 ions/Å². Likewise, a critical ion energy for mixing can be defined.
Chapter 5. Ion Fluence and Incidence Angle Effects

as the lowest ion energy for the initial onset of mixing (i.e., $M > 0.02$) after the high ion fluence of 0.5 ions/$\text{Å}^2$. It should be pointed out that values of the flattening and mixing critical energies depend on the designated fluence, but this dependence is relatively weak when fluence is 0.5 ions/$\text{Å}^2$ or above.

To examine the role of incident ion angle, critical energies for flattening and mixing were calculated as a function of the ion incident angle for the different ion species. The results are shown in Fig. 5.4 for both flattening and intermixing. It can be seen that the critical energies are relatively independent of ion incident angle when the angle is small but increase rapidly once the ion incident angle increases above $60^\circ$. Furthermore, the critical energies of mixing are close to that of flattening near normal ion incident angle and become larger than the latter as the ion incident angle increases to around $55^\circ$. However, flattening critical energies dominate again when the incident angle becomes even larger. To grow improved metal multilayers using ion assistance, it is necessary to use an ion energy that is sufficient to flatten a surface but is significantly less than the threshold ion energy for intermixing. It is evident from Fig. 5.4 that this is difficult to achieve when the ions strike the surface at a normal incident angle ($\theta = 0^\circ$). However, Fig. 5.4 suggests that flattening without mixing can be achieved when a relatively high ion incident angle is used.
Figure 5.4 (a) Surface flattening and (b) interfacial mixing of critical ion energies as a function of ion incident angle for different inert gas ion masses. Solid lines in the figure are the predictions of a two-body collision model.
Chapter 5. Ion Fluence and Incidence Angle Effects

Critical flattening and intermixing energies were also calculated as a function of ion incident angle using the collision model. The flattening calculations were based on Eq. (5-2), but with the coefficient “A” allowed to be a function of incident angle, θ. We found that the $A(\theta)$ function is proportional to the effective area of the surface asperities projected in the direction of ion incidence. For the pyramidal asperities shown in Fig. 5.1, $A(\theta)$ was well approximated by $B\cos(\theta)$, where $B$ is a coefficient related to the area of roughness in the direction of ion incidence. Substituting $R = 0.02$ and $F = 0.5$ ions/Å² in Eq. (5-2) yields an expression for the critical flattening energy, $E_{FC}$:

$$E_{FC}(\theta) = \frac{M_1 + M_2}{4M_1M_2} \left[ \frac{-E_f}{\ln(R^{-2/3}) - 1 - \ln(A(\theta)F)} - kT \right].$$ \hspace{1cm} (5 - 6)

The critical intermixing energy calculations were based on Eq. (5-5). In this case, the component of an impact momentum perpendicular to the surface plays an important role in the interlayer exchange mechanism. The impact energy due to the perpendicular momentum component is proportional to $\cos^2(\theta)$. The critical mixing energy, $E_{MC}$, was then derived as:

$$E_{MC}(\theta) = b \frac{M_1 + M_2}{4M_1M_2 \cos^2 \theta} \left[ \frac{-E_a}{\ln(-\ln(1-IM)) - \ln(aF)} - kT \right],$$ \hspace{1cm} (5 - 7)

where $a$ and $b$ are constants. The predictions of Eqs. (6) and (7) are shown as solid lines in Fig. 5.4. They are seen to be in generally good agreement with the simulated results. Variations of momentum transfer by using different assisting ions and the effective impact area of islands by using different ion incident angles clearly affect the flattening and mixing atomic mechanisms.
Chapter 5. Ion Fluence and Incidence Angle Effects

Careful examination of mixing data in Fig. 5.4 (b) indicates a discrepancy between the simple model and the simulations at incident angles between 45° and 65°. The simple model, Eq. (5-7), overestimates the critical mixing energy in this incident angle range for Ne, Ar, and Xe ion impacts. It is shown below that the deviation of the critical energy for mixing from the model for incident angles ~ 50° is a consequence of the existence of low energy barrier (atom channeling) paths beneath the copper surface.

5.5 Atomic Assembly Mechanisms

Stop-action images of atomic structures just prior to and after an 8 eV Ar ion impacts with a Co asperity on a (111) Cu surface are shown in Fig. 5.5 at different ion incident angles. They enable the evolution of the asperity structure to be observed as a function of incident angles. It can be seen that each of the impacts resulted in flattening, consistent with the results in Fig. 5.4(a). In each case, cluster flattening occurred within a few hundred femtoseconds after an ion impact and was enabled by energy transfer to atoms near the impact site. The momentum acquired by the atoms at the top of a pyramidal asperity allowed them to jump over an Ehrlich-Schwoebel barrier to form a two monolayer thick island. The energy transferred from the impacting ion to the cluster is sensitive to ion incident angle (for the cases shown in Fig. 5.5, it was 3.492 eV for 10°, 2.101 eV for 30°, and 0.401 eV for 60°). These differences in transferred energy then significantly affected the jump path and jump probability for the flattening process.
To more clearly examine the details of the impact-induced flattening mechanisms, the initial and final atomic configurations of a three-layer pyramidal Co cluster before and after an impact of an 8 eV argon ion at 10° and 30° incident angles are shown in Fig. 5.6. The atoms involved in the reassembly are marked as A and B with atom A in the top layer and atom B in the second layer. When the ion incident angle was small (nearly parallel to the Cu surface normal), atom A received sufficient energy to overcome the Ehrlich-Schwoebel barrier for a double monolayer jump and directly landed on the Cu surface, Fig. 5.6 (b). This is similar to the mechanism of reassembly shown in Fig. 5.5 (a).
During this jump, atom $A$ was found to travel in either a $[\bar{1}21]$ or $[211]$ direction even though the incident direction of the ion was originally projected to the $[\bar{1} \bar{1} 2]$ direction, Fig. 5.6 (a). This arises because of the existence of lower energy barriers in these two directions. MS estimates of Ehrlich-Schwoebel energy barriers in the various jump directions indicates an energy barrier of 1.68 eV for jumps along either the $[\bar{1}21]$ or $[211]$ directions compared to 1.85 eV for a jump in the $[\bar{1} \bar{1} 2]$ direction.

(a) Initial $[\bar{1} \bar{1} 2]$  
(b) After impact ($10^\circ$)  
(c) After impact ($30^\circ$)

Figure 5.6 Illustration of cluster evolution during an 8 eV Ar ion impact. (a) An initial 19 atom cobalt cluster on a (111) copper surface; (b) after an 8 eV argon ion impact at an incident angle of 10°, the cobalt atom $A$ has dropped two atom levels and resides on the copper surface; (c) after an 8 eV argon ion impact at an incident angle of 30°, the cobalt atom $A$ has pushed atom $B$ onto a copper surface by substitution into its original position.

As the ion incident angle was increased to 30°, atom $A$ jumped by a different mechanism. In this case, it occupied the original position of the underlayer atom $B$ and, in doing this, displaced the B atom over a ledge barrier onto the Cu surface, Fig. 5.6 (c). This was similar to the mechanism responsible for reassembly of the asperity in Fig. 5.5 (b). The energy barrier for such a jump was calculated to be 0.62 eV. Again, atom $B$
jumped in either a \([121]\) or \([211]\) direction even though the projected incident direction of
the ion was originally in the \([\bar{1}\bar{1}2]\) direction. When the ion incident angle was increased
further to 60°, atom \(A\) was found to remain on the top of the asperity after impact, Fig. 5.5
(c). In this case, the transferred energy was insufficient to overcome any of the flattening
energy barriers.

Ion impact induced mixing of a Co atom in the Cu surface was found to be very
effective when the transferred momentum resulted in atom motion in a copper channeling
direction. Studies of channeling in fcc Cu have shown that the low-index \(<110>\), \(<111>\),
\(<100>\), and \(<112>\) crystal directions are preferred channeling directions [185-187]. To
examine this channeling aided process more carefully, the intermixing processes during 8
eV argon ion impacts are shown in Fig. 5.7 for incident ion angles of 10° [Fig. 5.7 (a)],
30° [Fig. 5.7 (b)], and 50° [Fig. 5.7 (c)]. In Fig. 5.7, the left-hand column corresponds to a
side view observed along the \([\bar{1}\bar{1}0]\) direction while the right-hand column corresponds to
a top projection looking along the \([\bar{1}\bar{1}\bar{1}]\) direction. In the figure, Ar, Co, and Cu atoms
are shown as dark gray, light gray, and white atoms, respectively. The trajectories of the
Co and Cu atoms involved in the mixing processes are indicated by using smaller balls to
designate their positions at 0.2 ps time intervals measured from the moment of ion impact.
The full size atoms represent the original positions of atoms that were not displaced.
Figure 5.7 Analysis of mixing mechanisms during an 8 eV Ar ion impact. (a) Incident ion angle of 10°, (b) incident ion angle of 30°, and (c) incident ion angle of 50°. Atoms involved in the mixing are shown using smaller balls. Their position is shown at a sequential time interval of 0.2 ps while the full size atoms are shown in their initial positions.

It can be seen that regardless of the original ion impact direction, Co atoms always tend to migrate into the Cu substrate along low-index directions. For example, all the atom injection trajectories shown in Fig. 5.7 are along the low index $\langle 1 \bar{1} 0 \rangle$ direction.
Chapter 5. Ion Fluence and Incidence Angle Effects

regardless of the original incident ion direction. The top view images shown in Fig. 5.7 also indicate that all the ejected atom trajectories are projected in a low-index $<121>$ direction. The energy barriers for Co atom penetration into a Cu (111) surface in a channeling direction were calculated using MS. These energy barriers were 2.56 eV for the $<110>$ directions, 2.62 eV along the $<111>$ directions, 2.76 eV along the $<100>$ directions and 2.68 eV in the $<112>$ directions. The availability of many mixing channeling paths with similar energy barriers facilitates the mixing. As a result, Fig. 5.4 (b) shows that the critical energy for mixing obtained in simulations is lower than that obtained from the energy transfer model at incident ion angles near $50^\circ$.

As the impacting ion angle exceeds $60^\circ$, the resolved momentum in a channeling direction decreases and the ion impacts are less likely to cause the Co atom to penetrate the crystal. As a result, a simple ion reflection mechanism dominates in the large angle regime. In this regime, the assisting ions increasingly transfer only a lateral momentum to the surface atoms before reflection from the surface. They do not efficiently transfer the momentum that activates the mixing mechanism. This is consistent with the steep increase of the critical energy of mixing seen in Fig. 5.4 (b).

5.6 Deposition of Cu/Co Multilayers

The results above show the effects of ion fluence and incident angle upon the interfacial structure at a Co on Cu interface subjected to low ion energy bombardment. They indicate that selection of the ion mass, energy, the angle of incidence, and the fluence enable creation of an interface with a controlled combination of interfacial
roughness and intermixing. These ion parameters enable selective activation of the atomic assembly mechanisms that promote step flow growth while avoiding those that enable atom exchange across the dissimilar metal interface. This surface modification study has identified the approximate functional relationships between the ion flux parameters and interfacial structure ion modification of a model system consisting of Co islands on a (111) Cu surface.

Many applications of metal multilayers require a minimum combination of interfacial roughness and mixing [88, 156, 188]. For the argon ions, Fig. 5.4 indicates that ion energies in the 5-7 eV range are above the critical energy of flattening and yet below that for intermixing at an ion incident angle between 50° and 60° and an ion fluence of 0.5 ions/Å². It is therefore interesting to test if this combination of ion assistance parameters results in optimal interfacial structures when used during the vapor deposition of a Cu/Co/Cu multilayer.

To investigate this, we have performed an MD simulation of the ion assisted growth of a Cu/Co/Cu multilayer. We used argon ions with an energy of 6 eV, an ion incident angle of 50°, and various ion/atom ratios between one and ten (corresponding to an ion fluence range between 0.1 and 1.0 ions/Å²). The metal deposition simulation method was identical to that described in Chapter 4. The substrate was an initially flat, defect free copper crystal with a (111) surface. A cobalt layer and then a copper layer were sequentially deposited onto the surface using an adatom energy of 0.5 eV. An adatom incident angle of 40° was used to create flux shadowing so that the deposited surface is expected to be rough without ion assistance. To simulate sufficient atoms with the
available computational resources, a high deposition rate of 1 nm/ns was used while the substrate temperature was maintained at 300 K throughout the growth process. Due to the need to use an accelerated deposition rate, the time for thermal diffusion of atoms on the surface was significantly shorter than that available in practical processes. However, the simulated conditions ensured that the time interval between ion or adatom arrivals was at least 0.15 ps. This allowed impact induced surface reconstructions to be realistically captured and allowed the individual effects of ion impact under kinetically constrained metal atom deposition conditions to be reasonably assessed [43]. During the simulations, both the metal adatom and assisting ion fluxes were simultaneously injected toward the substrate surface.

Representative multilayer structures for different Ar ion to metal atom ratios are shown in Fig. 5.8. The use of a very low ion to metal adatom ratio resulted in a very rough cobalt surface and an inherited very rough Cu on Co interface, Fig. 5.8 (a). This occurs because the low adatom mobility was insufficient to compensate the strong flux shadowing conditions that promote surface roughening. We also note that the low fluence resulted in very little intermixing at either of the interfaces in Fig. 5.8 (a).
Figure 5.8 Simulated Cu/Co/Cu multilayer structures deposited using simultaneous argon ion assistance: (a) ion/metal ratio = 1; (b) ion/metal ratio = 5; and (c) ion/metal ratio = 10. The metal atoms had an energy of 0.5 eV and were deposited at an incident adatom angle of 40°. The ion energy was 6 eV and the ion incident angle was 50°. The center part of each multilayer has been removed to better reveal atomic scale structures.
Chapter 5. Ion Fluence and Incidence Angle Effects

The use of a much higher ion/atom ratio of 10 ion impacts/atom, Fig. 5.8 (c) resulted in much smoother interfaces, but extensive mixing occurred at the Co on Cu interface. At an intermediate ion flux of 5 ions/metal adatom, Fig. 5.8 (b), the combination of flatness and chemical sharpness at both interfaces was significantly improved. These deposition simulations therefore indicate that under conditions where surface atom mobility is kinetically inhibited (low temperature and rapid deposition rates), low energy ion assistance methods can be used to create very high quality interfacial structures. This is accomplished by selective activation of atomic assembly processes that promote flattening without interlayer mixing.

5.7 Conclusions

In this chapter, an MD simulation approach has been used to investigate the effects of ion fluence and ion incident angle upon the atomic reassembly of a (111) copper surface partially coated with 12 cobalt clusters. The study extends an earlier analysis of the effects of ion energy and mass upon the same system in Chapter 4. These studies indicate:

1. For ions whose mass specific energy exceeds a critical value, increasing the ion fluence increases the probability of ion impact with atoms in surface asperities and results in a decrease in surface roughness. However, as the ion energy is increased above an ion mass dependent critical value, interlayer mixing increases with ion fluence. At high ion fluence, the surface roughness eventually approaches zero while interlayer mixing eventually saturates.
2. Different flattening mechanisms are observed at different ion incidence angles due to differences in the transferred momentum. When the ion incident angle is normal to the growth surface, atoms near the apex of asperities receive sufficient energy and momentum during the impact to overcome Ehrlich-Schwoebel barriers and then are able to directly jump on to the copper surface. At high incident angles, less energy transfer occurs and atoms on the top of asperities either remain on the same layer or jump by a lower activation barrier indirect exchange mechanism. It is therefore possible to select a flattening mechanism by the choice of the ion incident angle.

3. Cobalt atom mixing in underlying copper layers is found to occur by a knock-on process that is dependent upon the incident ion direction. A “channeling effect” has been observed during low energy ion impacts that transfer momentum in low-index crystal directions such as $<110>$ and $<112>$. These directions have “lower energy barrier” pathways for the knock-on process. The selection of an ion incidence angle that avoids these low-index directions can reduce interlayer mixing.

4. A simple energy transfer model for ion impact with the growth surface is found to account well for the simulation results. The model identified an optimum ion energy regime (above the flattening and below the intermixing critical energies), where Co-on-Cu surfaces with low roughness and intermixing can be grown.

5. The predicted optimum ion beam parameters have been used to simulate the ion assisted growth of a Cu/Co/Cu multilayer system. The results confirm that low
energy ion assisted deposition using an oblique ion incident angle, a moderate ion fluence, and ion masses that are comparable to the mass of the metals results in high quality interfacial structures.
CHAPTER 6. LOW ENERGY ION ASSISTED CONTROL OF INTERFACIAL STRUCTURES

A MD method again has been used to simulate the argon ion assisted deposition of Cu/Co/Cu multilayers and to explore ion beam assistance strategies that can be used during or after the growth of each layer to control interfacial structures. A low argon ion energy of 5-10 eV was found to minimize a combination of interfacial roughness and interlayer mixing (alloying) during the ion assisted deposition of multilayers. However, complete flattening with simultaneous ion assistance could not be achieved without some mixing between the layers when a constant ion energy approach was used. It was found that multilayers with lower interfacial roughness and intermixing could be grown by either modulating the ion energy during the growth of each metal layer or by utilizing ion assistance only after the completion of each layer’s deposition. In these latter approaches, relatively high energy ions could be used since the interface is buried and less susceptible to intermixing. The interlayer mixing dependence upon the thickness of the over layer has been determined to be a function of ion energy.

6.1 Introduction

In ion assisted vapor deposition approaches, ion beams [129] and plasmas [156, 189] are usually used to induce inert gas ion impacts with growth surfaces. This results in local energy transfer to surface atoms and enhanced surface atom mobility and enables use of a low epitaxial growth temperature to avoid interlayer diffusion in the subsurface region.
Chapter 6. Low Energy Ion Assisted Control of Interfacial Structures

[166, 167, 190]. However, conventional ion beam assisted deposition processes utilize high energy ion fluxes in the 50 eV to 5 KeV range during metal deposition. While these have been successfully used to create flat surfaces on thick monolithic films [166, 167], these ion energies exceed the damage threshold energy for multilayers with nanoscopic layer thicknesses and result in significant interlayer mixing [191, 192].

MD simulations have been used to explore the atomic assembly mechanisms during the vapor deposition of metallic multilayers. The structures predicted by these accelerated deposition rate simulations are surprisingly similar to those observed by three-dimensional atomic probe techniques [28]. The predicted misfit dislocation structures are also similar to those observed by high resolution transmission electron microscopy [41, 193]. Simulations of ion interactions with a model Co-on-Cu bilayer system indicate that low (5-10 eV) energy impacts can be highly effective at flattening small (10 atoms) cobalt clusters on (111) copper surfaces without inducing interfacial alloying [171, 191, 194]. Previous chapters also reveal that appropriate combinations of ion species (mass), ion energy, ion fluence, and ion incident angle can be used to selectively activate either atom recoil, atomic exchange, or direct jumping mechanisms to flatten surface asperities [191, 194]. Studies also indicate that nearly perfect multilayer structures can be vapor deposited using ion assistance if the ion energy is maintained between the critical energies for flattening and intermixing. These energies and the energy gap depend upon the ion incident angle and ion mass. The widest critical energy gap is achieved for an ion incident angle around 50° from surface normal (which avoids easy atom penetration in crystal channeling directions) using an ion mass comparable to that of the metals [194].
There are many ways, in which a low energy ion assistance method could be used to control the assembly of a metal multilayer thin film. In the example above, constant energy ion bombardment was used simultaneously with the deposition process. However, the ion energy could also be varied (modulated) during the metals deposition, or the ion bombardment could be applied only after each metal layer had been deposited (sequential assistance). In this chapter, we utilize MD simulations to investigate these three ion assistance approaches for controlling the interfacial structures of a model Cu/Co/Cu trilayer system. Several near-optimal processes for the growth of metallic superlattices with chemically sharp, planar interfaces are identified.

6.2 Simulation Methodology

The MD simulation methodology used for the previous chapters of the ion modification of Co/Cu multilayers was also used here. Briefly, deposition and ion assistance were simulated by injecting metal and argon ion atoms toward the surface at an angle, \( \theta \), to the \([\overline{1} \overline{1} \overline{1}]\) direction in the \((1 \overline{1} 0)\) plane. The incident particles were assigned an incident energy, \( E_i \). First a cobalt and then a copper layer were deposited onto an initially flat \((111)\) copper surface. For all simulations, a fixed deposition rate of 1 nm/ns and an adatom energy of 0.5 eV were used. Ion/adatom flux ratios of one and two were used for simulations of simultaneous and modulated energy ion assistance schemes, respectively. An ion fluence of 0.5 ions/Å² was used to assess sequential ion assistance schemes. The simulated conditions were carefully chosen to ensure that the time interval between ion or adatom arrivals was greater than 0.15 ps. This time interval allowed
impact-induced surface reconstructions to be realistically captured and the effects of assisting ions to be more accurately predicted [191].

To avoid a crystal shift following an impact, the positions of the atoms in the bottom two (111) copper layers were fixed. Atoms in a sub-surface region above the fixed layer were maintained at a constant temperature of 300 K using a thermostat algorithm [28]. The study was restricted to argon ion effects since it is the most commonly used inert gas in ion assisted vapor deposition experiments. The magnitude of ion assistance effects are dependent upon the ion mass and are described in the Chapter 4.

6.3 Simultaneous Ion Assistance with Metal Flux Deposition

In conventional ion beam assistance methods, continuous inert gas ion irradiation of the growth surface accompanies metal deposition [166]. Numerous simulations of a Cu/Co/Cu trilayer structure deposition with argon ion assistance were conducted with ion incident energies between 0 eV and 30 eV and different ion incident angles. Details of the atomic reassembly mechanisms using simultaneous ion assistance can be found in previous chapters. Fig. 6.1 shows examples of the atomic structures obtained using selected ion energies between 0 eV and 20 eV, a fixed ion/metal ratio of unity, and a normal ion incident angle. In the figure, copper, cobalt, and argon atoms are marked by white, gray, and black spheres, respectively. The simulations reveal the strong dependence of the atomic scale structure upon the incident ion energy. When there is no ion assistance, Fig. 6.1(a), the cobalt adatom surface mobility is insufficient to reconstruct the surface. As a result, a rough cobalt surface is formed, and when this is
subsequently buried by the copper layer, it becomes a rough Cu-on-Co interface. When the adatom energy was close to zero eV, very little intermixing between layers was observed, which is consistent with earlier studies [191].

Figure 6.1 Atomic structures of a Cu/Co/Cu trilayer grown using simultaneous metal deposition and argon ion assistance. The substrate temperature was 300 K and ion/metal flux ratio was unity. Assisting argon ion energies were (a) 0 eV (no ion assistance), (b) 8 eV, and (c) 20 eV. The ions were introduced in the \([1 \bar{1} 1]\) direction.
Chapter 6. Low Energy Ion Assisted Control of Interfacial Structures

When the ion energy was increased to 8 eV, Fig. 6.1 (b), the cobalt film surface and resulting interface roughness were both reduced. The previous study has shown that this reduction arises because the assisting ions are able to transfer sufficient momentum to surface cluster adatoms enabling them to migrate and find ledge/edge sites that are one or two monolayers below the cluster peak [191]. For ion energies below 10 eV, this process leads to a significant flattening of the interface without activating interdiffusion. However, if the ion energy is raised too high, Fig. 6.1 (c), the interface becomes heavily intermixed as it is flattened. Thus, there exists an optimum ion energy near 8 eV that minimizes a combination of surface roughness and interfacial intermixing. This optimum energy is a function of ion species and ion/atom flux ratio [191, 194].

A root-mean-square (RMS) deviation parameter can be used to quantify the roughness of the trilayer structure, and the number of cobalt atoms mixed into the underlying copper layer at a Co-on-Cu interface can be used to quantify intermixing. The RMS roughness and mixing as a function of ion energy for simultaneous ion assistance is shown in Fig. 6.2 as solid circles. Open circles and solid squares represent the results for the modulated energy and sequential ion assistance schemes discussed in next section. The results in Fig. 6.2 (a) indicate that the roughness is significantly decreased as the assisting ion energy is increased. For instance, the RMS roughness was approximately 6.3 Å without ion assistance but decreased to 4.5 Å using 10 eV argon ion assistance. Although ion energies above 10 eV further decrease the RMS roughness, Fig. 6.2 (b) indicates that this is accompanied by significant interlayer mixing using a simultaneous assistance approach.
Figure 6.2 (a) RMS roughness and (b) number of cobalt atoms mixed into the underlying copper layer as a function of argon ion energy. The ions were introduced in the [1 1 1] direction.
Chapter 6. Low Energy Ion Assisted Control of Interfacial Structures

The ion incidence angle also affects the mechanisms of atomic assembly and the energy for optimum interfacial structures. Fig. 6.3 shows a 3-dimensional plot of the roughness and mixing parameters as a function of ion energy and incident angle. These results indicate that both roughness and mixing parameter dependencies on ion energies are rather insensitive to ion incident angle for angles lower than 50°. However, the roughness or mixing increases or decreases very rapidly once the ion angle exceeds 50°. Further analysis indicates that the lowest combination of roughness and mixing occurs at an ion energy of 10 eV and an ion incident angle of 50°.

![Figure 6.3 Effect of ion energy (E) and incident angle on the (a) roughness and (b) mixing indices using a simultaneous ion assistance scheme. The ion angle was defined by its deviation from the \([\bar{1} \bar{1} 1]\) direction in the (110) plane.]

The amplitude of the interfacial roughness and width of intermixing at an interface are critical properties of a multilayer. To explore this further, the copper composition was
calculated as a function of film thickness for multilayers deposited using different ion energies and a fixed ion/metal ratio of unity, Fig. 6.4. The results indicate that the Co-on-Cu interface grown using 20 eV argon ion assistance is much more diffuse than that grown with 8 eV or without any ion assistance. This mixing has been shown elsewhere [191] to occur by a high energy ion impact induced exchange mechanism. It can also be seen that the Cu-on-Co interfaces grown using either no assistance or with 20 eV ion assistance are more diffuse than those obtained at 8 eV ion assistance. The diffuse Cu-on-Co interface formed using 20 eV ion assistance is again the result of an atomic exchange mechanism [191], whereas the apparent diffuse Cu-on-Co interface achieved without ion assistance is a manifestation of the rough interface (see Fig. 6.1 (a)).

![Graph depicting copper concentration profiles](image)

Figure 6.4 Copper concentration profiles through a Cu/Co/Cu trilayer deposited using a simultaneous ion assistance and metal deposition scheme. Profiles are shown for films grown without ion assistance (0 eV) and with 8 eV and 20 eV assistance applied perpendicular to the growth surface.
6.4 Other Ion Assistance Schemes

6.4.1 Effects of Interface Depth

It has been shown that quite chemically sharp, almost flat interfaces can be fabricated with simultaneous ion assisted deposition scheme using an optimized ion energy. However, the optimum ion energy is sensitive to underlying/overlying materials. Both Figs. 6.1 and 6.4 indicate that the mixing and roughness of the Cu-on-Co and Co-on-Cu interfaces are different when the same growth conditions are used. This arises because the energy barriers for cobalt migration over terrace edges and the atomic exchange of copper adatoms with underlying cobalt atoms are higher than those for copper atom migration as well as the exchange of a cobalt adatom with an underlying copper atom. Universal ion energy is therefore a compromise for the two materials and a better structure would be obtained using different constant energies for the two layers.

We also recognize that high energy ion induced mixing occurs only when an interface is near to the surface where ion impacts occur. The momentum transfer process is unable to drive the exchange processes more than a few monolayers below the surface. The degree of mixing can therefore be reduced by using the high energy ion assistance only after the interface has been buried deeply by deposition of the next material layer. As the interface becomes more deeply buried, it is reasonable to expect that more energetic ions could be used for surface flattening without incurring a higher probability of mixing. To explore this, a series of structures were created by depositing cobalt in
different thicknesses on a copper substrate. Ion impacts were then introduced and the intermixing probability examined.

The interlayer mixing probability as a function of ion energy and interface depth is shown in Fig. 6.5, which shows that increasing the interface depth significantly reduces interfacial mixing. It can also be seen that the use of ion energy in the "optimized energy region" for simultaneous ion assistance (8 eV) requires only one monolayer of coverage to avoid mixing. As the ion energy is increased to 20 eV, the intermixing probability at the interface drops from around 12% to 3% and then to 2% for one monolayer, two monolayers, or three monolayers of interface coverage. Almost no mixing was found for four and five monolayers of coverage at a 20 eV ion energy. If the ion energy is further increased to 30 eV, four monolayer coverage still results in only a 5% mixing probability, and five atomic layers result in almost no mixing at a 30 eV ion energy.

![Graph showing the relationship between interface depth and ion energy for different mixing probabilities.](image)

Figure 6.5 Cobalt mixing probability as a function of argon ion energy for different depths of the Co-on-Cu interface below the film surface.
6.4.2 Modulated Energy Ion Assistance

As discussed above, once a complete coverage of the underlying material by four or five atomic layers of new material has been achieved, the subsequent use of higher energy ion impacts to more effectively flatten the layer is much less likely to cause intermixing. It might then be possible to use a modulated ion energy scheme to promote flatness without inducing interlayer mixing. In such an approach, the first few atomic layers of a new material layer would be deposited with a very low energy (or no) ion assistance, and then the remaining layer deposition would be completed using higher energy assistance.

Simulations were performed to explore this modulated energy ion assistance strategy for the deposition of a cobalt layer on a copper substrate. The first five atomic layers of cobalt were deposited without ion assistance, Fig. 6.6 (a). The remainder of the cobalt layer was then deposited with ion assistance using various ion energies and a fixed ion/metal ratio of two. Fig. 6.6 (b)-(d) show typical structures obtained using 6 eV, 15 eV, and 20 eV argon ion energies. When the first half cobalt layer is deposited without ion assistance, there is essentially no intermixing at the interface. However, the surface roughness is high due to the lack of surface atom migration. Fig. 6.6 (b)-(d) indicate that the surface roughness is progressively decreased as the assisting ion energy is increased while the interlayer mixing remains relatively low. Very flat surfaces and interfaces are obtained using ion energies of around 20 eV. The roughness and interlayer mixing resulting from this modulated energy ion assistance method are summarized in Fig. 6.2.
The surface roughness progressively decreases with increasing the ion energy while interlayer mixing remains low. The minimum combination of mixing and roughness were achieved at an argon ion energy of around 20 eV for the Co-on-Cu interface.

Figure 6.6 Modulated ion energy deposition of Co-on-Cu. (a) First half (five atomic layers) of the cobalt layer was deposited without ion assistance. The remainder of the cobalt layer was deposited with an argon ion energy of (b) 6 eV, (c) 15 eV, and (d) 20 eV.

The modulated ion assistance technique was also used to deposit copper layers on the bilayer structure shown in Fig. 6.6 (d). Again, the first five atomic copper layers were
deposited on the cobalt surface without ion assistance, Fig. 6.7 (a). The remainder of the copper layer was then deposited with ion assistance at an ion energy of 20 eV and an ion/metal ratio of 2, Fig. 6.7 (b). It can be seen that the modulated energy ion assistance scheme resulted in multilayers with chemically sharp, physically flat interfaces.

(a) First half of Cu layer without ion assistance

(b) Second half of Cu layer with argon ion energy of 20 eV

Figure 6.7 Growth of a Cu/Co/Cu trilayer using modulated ion energy assistance with an argon ion energy of 20 eV. (a) First half (five atomic layers) of the copper layer was deposited without ion assistance. (b) The remainder of the copper layer was deposited with an argon ion energy of 20 eV.

6.4.3 Sequential Ion Assistance

The maximum ion energy that can be used in a modulated ion energy assistance scheme is limited by the distance between surface and interface at the onset of ion assistance. The highest ion energy (greatest smoothing rate) can be used once a layer has been fully deposited. A sequential ion assistance approach was therefore explored in
which a complete metal layer was first deposited without (or with very low energy) ion assistance, and then relatively high energy ion impacts were used to flatten the metal surface after the metal layer was deposited. The interfacial roughness and intermixing results for the sequential ion beam assistance scheme are summarized in Fig. 6.2. It can be seen that the mixing was always very low and the RMS roughness was decreased (to as low as 2.9 Å) as the ion energy was increased to 30 eV. The combination of roughness and mixing continued to decrease as the ion energy was increased to 30 eV.

A complete Cu/Co/Cu trilayer was grown using the sequential deposition scheme with an ion energy of 30 eV, Fig. 6.8. Fig. 6.8 (a) shows the cobalt layer deposited on copper without ion assistance. The surface was then irradiated with ions at an ion energy of 30 eV to an ion fluence of 0.5 ions/Å², Fig. 6.8 (b). This process was very effective at smoothing the surface. A copper layer was deposited onto this smoothed cobalt layer without ion assistance, Fig. 6.8 (c). Finally, the copper surface was irradiated with ions at an ion energy of 30 eV to an ion fluence of 0.5 ions/Å², Fig. 6.8 (d). The results show that the sequential ion assistance scheme resulted in a multilayer structure with planar interfaces and little interfacial mixing even when a moderately high ion energy of 30 eV was used.
Figure 6.8 Deposition of a Cu/Co/Cu trilayer using sequential ion assistance with an argon ion energy of 30 eV and an ion fluence of 0.5 ions/Å². (a) Structure after completion of first cobalt layer, (b) structure after ion irradiation of the cobalt surface at 30 eV, (c) structure after deposition of the top copper layer, and (d) structure after ion irradiation of the copper surface at 30 eV.

Examination of the three ion assistance approaches, Fig. 6.2, also indicates that at the same assisting ion energy, the simultaneous ion beam assistance approach usually
resulted in slightly lower surface roughness values but was accompanied by the highest probability of interfacial mixing. The sequential ion beam assistance in all but the 30 eV case resulted in the smallest flattening effect and the least interfacial mixing. Additional simulations (not shown) indicated that for all three ion assistance schemes, the roughness increased as the ion energy was increased above 30 eV due to the resputtering of the surface atoms. The effect of ion incident angle was also examined for the modulated energy and sequential ion assistance schemes. These simulations indicated very similar effects to those reported for the simultaneous ion assistance approach described above.

6.5 Discussion

The results discussed above indicate that optimum ion energy (to achieve the lowest combination of interfacial roughness and mixing) in a Cu/Co/Cu multilayer structure depends on the ion assistance method. For example, the optimized argon ion energy for simultaneous assistance is around 8 eV, whereas for modulated energy or sequential ion assistance, it is either 20 eV or more than 30 eV, respectively. This difference arises in part because the size of asperities that must be smoothed is different for the three ion assistance methods. During simultaneous ion assistance, the assisting ions arrive simultaneously with the metal atoms and therefore continuously reduce the small surface asperities that are nucleated during larger growth. As a result, relatively low energy (8 eV) ions are effective at flattening the surface. When the assisting ions are only applied after a certain layer thickness has been deposited, as in both modulated energy and sequential ion assistance processes, the average size of the surface asperities is larger, Fig. 6.6 (a)
and Fig. 6.8 (a). For a fixed ion fluence, higher energy ions are then needed to flatten the
more well developed roughness. This is the primary explanation of the best flattening
performance of the simultaneous ion assistance method in the low energy region, Fig. 6.2.

The effects of the surface asperity size upon the flattening process can be
illuminated using the simulation approach, Fig. 6.9. The islands in Fig. 6.9 (a)-(c) contain
10-atom, 20-atom, and 35-atom size asperities, respectively. Their structure after an
argon ion impact with energies of 12 eV and 25 eV is shown. It can be seen that the 12
eV ion transfers sufficient energy to reduce the smallest (10-atom) island to a one
monolayer thick island. However, the same impact only causes minor flattening of the
20-atom and 35-atom asperities. When the ion energy is increased to 25 eV, more
effective flattening of the larger islands occurs and accounts for the observation that both
modulated ion energy and sequential ion assistance methods require relatively higher ion
energies to achieve a similar level of flattening to the simultaneous assistance approach.
Figure 6.9 Effect of cobalt cluster size on cluster flattening following argon ion impacts at 12 eV and 25 eV. (a) 10-cobalt atom asperity, (b) 20-cobalt atom asperity, and (c) 35-cobalt atom asperity.

It is important to recognize that even using a modulated ion energy or sequential ion assistance scheme, mixing still occurs when high ion energies are used. The highest ion energy that does not result in intermixing depends on the distance between the surface and the underlying interface, as shown in Fig. 6.5. Numerous simulations were conducted to investigate the relationship between the interface depth, ion energy, and fraction of cobalt atom mixing at the Co-on-Cu interface. The results are summarized in Fig. 6.10 in the form of mixing fraction contours on an ion energy-interface depth map. The map
provides a convenient means identifying the selection of an ion energy and layer thickness to achieve a desired level of mixing. For example, use of an argon ion energy of 10 eV can be used in simultaneous ion assistance schemes to flatten a surface without causing significant mixing. If the smallest energy that can be used is 20 eV, then four monolayers of cobalt must be deposited before ion assistance is applied and a modulated or sequential assistance approach is preferred.

![Graph showing mixing fraction contours](image-url)

**Figure 6.10** Cobalt interface mixing fraction contours on an ion energy-interface depth map.

Finally, we note that both simultaneous and sequential ion assistance techniques have been widely used for vapor deposition processing. The techniques can be
differentiated by the method used to create the assisting ion flux [166, 195]. One approach utilizes ion beam gun technology, which can be used in many ion beam deposition [196-198], magnetron deposition [37], and evaporation techniques [192]. The assisting ion beam can be used either for simultaneous or sequential ion assistance. In order to focus ion beams on a substrate, the ion gun approach normally requires the use of a relatively high ion energy > 50 eV. The ion incident angle is easily varied by gun placement. This ion energy range is well suited for sequential ion assistance provided the layers of multistacks are more than five monolayers in thickness. Electrostatic deceleration schemes near the substrate in principle provide a means to reduce the ion energy into the range best suited for simultaneous or modulated ion assistance.

The second approach utilizes plasmas located near the substrate to provide assistance. Examples include high pressure RF diode and magnetron sputtering techniques [199-202]. These methods utilize a negative bias voltage applied to the substrate to attract ions from plasma to the growth surface. They are usually used for simultaneous or sequential ion assistance. Since the plasma is also used to cause the vapor flux to be deposited, the ion energies can be high, but collisions with background gas atoms in high pressure processes can reduce their energy into the range of interest for simultaneous assistance. Modulated ion assistance appears to be more difficult to implement using this approach. A sequel paper reports a comprehensive experimental study of the low energy (< 30 eV) ion assisted growth of thin films using a biased target ion beam deposition [203].
6.6 Conclusion

An MD method has been used to explore low energy argon ion assisted deposition approaches for the fabrication of Cu/Co multilayers. The study indicates:

1. During continuous ion assistance, the ion energy significantly effects the interfacial structure of multilayer structures. High ion energies promote migration of surface atoms over Ehrlich-Schwoebel barriers and promote flattening of the film surface. However, during multilayer growth, they can also result in interlayer mixing by an atom exchange process. A narrow “optimal” ion energy range (5-10 eV) exists for fabricating high structural quality interfaces using simultaneous metal deposition and ion assistance.

2. A modulated energy ion assisting method (which uses low ion energy or no assisting ions to deposit the first few atomic layers of each new material layer and higher ion energies for the remainder of that layer) is found to result in a lower combination of interfacial roughness and interfacial mixing than the simultaneous ion assistance approach.

3. A sequential assistance method, which deposits a complete metal layer with very low energy (or no) assisting ions, followed by high energy ion surface modification of the fully deposited metal layer, can result in very high quality interfacial structures and may be more easily implemented using ion beam deposition approaches.
CHAPTER 7. BIAS ED TARGET ION BEAM DEPOSITION

As mentioned in Chapters 1, 2, 4, and 5, GMR properties are extremely sensitive to atomic scale interfacial roughness and interlayer mixing across multilayer interfaces. A deposition process that can reduce both roughness and mixing is the key to expanding the applications for GMR multilayers. Many deposition methods have been tried to grow GMR multilayers. The most successfully used ones are based on sputter processes. They include magnetron sputtering, diode sputtering, and ion beam deposition (IBD). IBD is particularly interesting because of its low cost, low processing pressures, directional deposition flux, high adatom ener gies, and independent control of target and substrate environments. However, traditional IBD methods were not designed for optimum deposition conditions in order to grow GMR multilayers. Atomistic simulations indicated several key problems of IBD. The insights revealed from atomistic simulations helped the development of a new biased target ion beam deposition (BTIBD) technology that overcame the problems of traditional IBD. This chapter describes this new type of BTIBD technology.

7.1. Insights from Simulations

MD simulations have been used to study the effects of adatom energy on interfacial roughness and interlayer mixing [28]. Fig. 7.1 (a) shows a simulated CoFe/Cu/CoFe/NiFe multilayer deposited at typical sputter conditions with a normal adatom incident angle ($\theta = 0^\circ$), a substrate temperature of 300 K, a deposition rate of 1 nm/ns, and an adatom energy of 3.0 eV. Fig. 7.1 (b) is a three-dimensional atom probe (3DAP) experimental
image of the same structure. Similar configurations can be seen in the simulation and the experiment, both indicating a severely diffused CoFe-on-Cu interface, but little or no diffusion at the Cu-on-CoFe interface [28].

![Diagram](image)

Figure 7.1. Atomic CoFe/Cu/CoFe/NiFe multilayers: (a) simulation at an adatom energy of 3.0 eV, and (b) atom probe experiment of a sputtered sample [28].

Simulations successfully predicted important atomic assembly mechanisms. They revealed that the interfacial roughness and interlayer mixing are very sensitive to the adatom energy [30]. Increasing adatom energy enhances the reconstruction of local surface asperities during adatom impact resulting in surface flattening. A flat surface then evolves into a flat interface after a new overlayer is deposited. However, high-energy adatoms are more likely to exchange with underlying atoms upon impact, resulting in mixing when they are deposited on a surface of different species. Since Cu atoms have a lower cohesive energy and a larger size than Co or Fe atoms, they tend to segregate on the surface to reduce surface energy and surface tensile stress. Because of this, Co or Fe atoms impacting a Cu surface are more likely to exchange with the underlying Cu atoms.
than when Cu atoms impact a CoFe surface. As a result, the CoFe-on-Cu interface is more mixed than the Cu-on-CoFe interface in Fig. 7.1. The continuous exchange of the Cu atoms to the surface during deposition of the CoFe layer then results in Cu atoms becoming mixed into the later deposited CoFe layer. This impact exchange related diffusion differs from the thermally activated diffusion.

The trade-off between interfacial roughness and interlayer mixing in part accounts for the observation that the best GMR multilayers are obtained at intermediate adatom energy in rf diode sputtering [204], magnetron sputtering [29], and IBD [205]. Deposition of Cu/Ni multilayers has been simulated using a fixed normal adatom incident angle to identify the Cu-on-Ni interfacial roughness and the Cu mixing probability in a Ni layer as a function of incident energy [30]. It showed that the adatom energy that produces lowest roughness and mixing should be less than 10 eV.

The findings from simulations indicated an improved approach. If a low energy is used to deposit the first few monolayers of a new material layer, the mixing at the interface can be reduced. A high energy can then be used to deposit the remainder of the layer to flatten the surface without causing mixing. While no deposition system has been designed to apply such intralayer energy modulation, this scheme has been used in simulation to grow the same CoFe/Cu/CoFe/NiFe multilayer as shown in Fig. 7.1. An example of the deposited structure [28] is shown in Fig. 7.2, where the energy scheme is indicated, and other conditions are the same as those used to obtain Fig. 7.1 (a). Fig. 7.2 verifies that the modulated energy deposition has the potential to significantly further reduce both interfacial roughness and interlayer mixing.
Figure 7.2. Atomic CoFe/Cu/CoFe/NiFe multilayer obtained from simulation using an intralayer energy modulation scheme [28].

MD simulations have also been used to study the effects of adatom incident angle on the deposition of a Ni/Cu/Ni multilayer [31]. Fig. 7.3 shows typical results of the multilayer structures obtained with and without substrate rotation at an incident angle of 1.0 eV, an oblique incident angle of 50°, a substrate temperature of 300 K, and a deposition rate of 10 nm/ns. It can be seen that deposition at an oblique adatom incident angle could cause significant surface roughness during deposition of the Ni layer (corresponding to a rough Cu-on-Ni interface) due to a shadowing effect [31]. Substrate rotation can reduce the effect of shadowing. As a result, greatly reduced surface/interface roughness created during the oblique angle deposition condition.
Figure 7.3. Simulated Ni/Cu/Ni multilayer deposited at 1.0 eV and an oblique incident angle of 50°. (a) No substrate rotation and (b) substrate rotation rate 0.196 turns/ML [31].

MD simulations have been carried out to study sputtering [30]. The results indicate that the average energy of sputtered atoms is typically between 1 eV and 5 eV, depending on the sputtering ion mass and energy, target material, and system geometry. In addition, a fraction of the high-energy inert gas ions used for the sputtering can be reflected from the target as neutrals. These neutrals can have energies in excess of 50 eV. High energy particles bombarding the surface of the growing film cause significant interlayer mixing in the GMR multilayers. Simulations indicated several effective approaches to reduce this type of mixing. First, the use of heavier inert gas ions, such as Xe rather than Ar ions for the sputtering and normal ion incident angle at the target, can greatly reduce both the energy and flux of reflected neutrals. Secondly, increasing background gas pressure can increase the scattering during flux transport and hence reduce the energies of adatoms.
Chapter 7. Biased Target Ion Beam Deposition

and reflected neutrals at the substrate. However, practical constraints of IBD severely limit these options.

7.2. Problems of Conventional Ion Beam Deposition

The above discussions indicated that controllable adatom energy in a range between 1 eV and 5 eV is important for the growth of good GMR multilayers. To date, magnetron (and rf diode) sputtering and IBD are the methods most successfully used for depositing GMR multilayers [206]. However, not one of them is ideally suited for optimal processing of atomically engineered interfaces required by devices. For instance, magnetron and rf diode sputtering systems usually operate at high pressures, and the adatom flux is substantially scattered by the low energy background gas during transport from target to substrate. Adatoms hence have relatively low energies. These adatoms are well suited for deposition at interfaces for avoiding interlayer mixing but are poorly suited for deposition of other parts, as they cannot effectively flatten the surface. The conventional IBD, on one hand, can only be operated using relatively high sputtering ion energies and relatively low pressure with no substantial scattering from the background gas. As a result, the adatoms in IBD can have significantly higher energies. For instance, MD simulation of the sputtering of the (110) Ni surface by 600 eV Xe ions indicated that the average energy of the sputtered Ni atoms is around 15 eV. These atoms are better suited for flattening the surface; they are poorly suited for reducing the mixing at the interfaces. In order to explore ways to improve IBD by incorporating the knowledge
obtained from atomistic simulations, we analyze the IBD system using a schematic illustration shown in Fig. 7.4.

![Diagram of ion beam deposition](image)

Figure 7.4. Conventional ion beam deposition approach.

In the conventional IBD system shown in Fig. 7.4, a primary grid ion beam gun is used to generate high-energy inert gas ions (Ar\(^+\), Kr\(^+\), and Xe\(^+\), etc.) that are directed at a depositing metal target. These ions can sputter metal atoms off from the target surface, which are then transported to the substrate (as the depositing flux). Growth occurs when these atoms are condensed on the substrate surface. Multiple targets can be installed in the rotational target assembly and targets can be switched during deposition of multilayers. A second grid ion beam gun can be used to create low-energy inert gas ions that are directed at the substrate to assist the growth. During deposition, the substrate is allowed to rotate. Normally, the energy of sputtering ions, the energy of assisting ions,
the incident angle of sputtering ions at the target, and the incident angle of assisting ions at the substrate can all be controlled, but often within a limited range.

There are a number of drawbacks with the design shown in Fig. 7.4. First, the lowest inert gas ion energy normally achievable in commercially available grid ion beam guns is about 50 eV. This is significantly higher than the ideal energy desired for ion-assisted growth of GMR multilayers. Second, the primary sputtering ions must impact the target at an oblique incident angle that may cause high energy and flux of reflected neutrals. Third, while IBD systems are designed to contain all sputtering ions on the target, a fraction of the ions may miss the target in practice. These high-energy ions may sputter off undesired materials from some of the vacuum system hardware, resulting in “overspill” contamination of the growth film. In order to reduce beam overspill, high performance IBD systems generally employ large targets. Because the majority of the ions are focused only near the center of the target, it results in highly concentrated target wear at its center and poor utilization of the target materials. The concentrated sputtering at the target center also significantly reduces the deposition thickness uniformity that is essential to the GMR multilayers. Fourth, it is difficult to apply lower adatom energy desired for the interface deposition as described above. This is because at low beam voltages, it is more difficult to focus the sputtering ion beam, resulting in larger overspill contamination. In addition, deposition rates in IBD systems (typically 0.1-2 Å/sec) are lower compared to other sputtering deposition techniques. It is then critical to increase the deposition rate so that the contamination from residual gases in the vacuum chamber does not exceed the threshold value to affect the properties. This has been verified by the
Chapter 7. Biased Target Ion Beam Deposition

experimental observation that the GMR properties always increase as the deposition rate is increased during IBD deposition [182]. Processing at low beam voltages also results in low deposition rates and low wafer output. Fifth, in a conventional IBD process, an instantaneous change of deposition energy required by the intralayer energy modulation scheme cannot be achieved because a change of the sputtering ion beam gun energy will necessarily be accompanied by the adjustment of the corresponding change of ion beam plume.

7.3. Biased Target Ion Beam Deposition

Based upon the insights discussed above, a new biased target ion beam deposition (BTIBD) technology [120, 207] has been developed to overcome some of the problems of the conventional IBD method. The first such BTIBD system has been successfully assembled at 4Wave, Inc., and is now being operated at University of Virginia.

The concept of the BTIBD system is schematically shown in Fig. 7.5. Six different targets can be installed in the target assembly so that multilayers of up to six different layers can be deposited. The wafer stage is at the left side of the chamber facing the sputtering target. The wafer plane can be tilted so that the adatom incident angle can be adjusted from $0^\circ$ (normal incidence) to $90^\circ$. During deposition, the wafer is allowed to rotate along its axis. A magnetic field along the growth surface can be applied to deposit layers with desired magnetic orientation.
Figure 7.5. Schematic illustration of a BTIBD system.

The key to this BTIBD system is that instead of using the conventional grid ion beam gun, it uses novel low energy ion sources that combine an end-Hall ion source [208] and a hollow cathode electron source [209]. A special feature of this kind of ion source is that it can reliably produce a very high density of inert gas ions with a very low energy (from several eV). In Fig. 7.5, one such ion source is constructed to face the substrate stage and is used to create low energy assisting ions (0-50 eV), ideal for GMR multilayer
deposition. Another similar ion source is used to provide a low energy (0-50 eV), and high density of ions in front of the sputtering target. By applying a negative bias to the sputtering target, the inert gas ions can strike the target at a near-normal incidence. Depending on the bias voltage, the impact energy of the ions on the target can be easily controlled between 50 eV and 2000 eV, which is suitable for the sputtering. Since the other hardware (such as target shield and chamber wall, etc.) is grounded, most ions will strike the negatively biased target. Some ions miss the target; they will have only the un-accelerated low energy, which is not sufficient to cause overspill contamination. This not only eliminates the need for an effort to capture all of the ions on the target, but also allows the use of an ion beam that is much broader than the target. As a result, the entire target can be uniformly illuminated and both the target material utilization efficiency and deposition thickness uniformity are significantly improved. Because overspill contamination is reduced even when the ion beam is not focused, and a much higher ion density can compensate for the loss of deposition rate due to low sputtering yield, BTIBD allows the use of lower sputtering energies (i.e., low bias voltages). It also has the potential to increase the deposition rate desired for IBD deposition of GMR multilayers.

A plasma sheath develops at the surface of the negatively biased target. Because the sheath is very thin (~2mm) compared to the spacing between the ion source and target, the target bias has no substantial effect on the ion trajectories from source to target. Hence, for constant source operation, the illumination profile and the ion current reaching the target are nearly independent of the target voltage. The target shield to target spacing is closer than a shield distance at the edges of the target to prevent penetration of the
plasma. A large range of target voltages (~100 to > 1000 eV) and target current (0-500 mA) can be used while maintaining reasonable deposition rates. This facilitates a smooth control of target bias voltage and a means for the realization of intralayer energy modulation during deposition. In addition, the ion sources are capable of operating over a broad range of processing pressures (10^{-4}-5\times10^{-3} torr) allowing some control of the adatom scattering from the background gas. For these reasons, BTIBD provides a means of controlling adatom energies over a wide range (from several eV to several tens eV).

7.4. Ion Source

The key to the novel BTIBD method are the low energy high density ion sources. Here, a combination of a Mark II end-Hall ion source and an HCES 5000 hollow cathode electron source was used. These sources are shown in Fig. 7.6. Here we review briefly the operation of the source. The Mark II<sup>TM</sup> actually acts as anode. It is made with hard and soft magnetic materials with a gas inlet. The hollow cathode consists of a narrow tube, a “keeper” surrounding the cathode, and an emission end. Electrons emitted from the cathode flow toward the anode of the end-Hall source where ions are created. The electric and magnetic fields near the anode determine both ion and electron path lengths, control the ionization in the discharge, and accelerate the ions out of the anode. The most important advantage of this ion source is that it enables the ions to be controlled in a rather broad low energy range. This is in sharp contrast to the gridded ion source used in conventional IBD where ions must be narrowly distributed in a high energy range in order to focus the ion beam at the target. Another advantage of the end-Hall source is that
the anode current can be independently controlled regardless of the details of the ion beam. This means that the number of emitted ions can be independently optimized to achieve the best deposition conditions.

Figure 7.6. Schematic Layout of Mark II End-Hall ion source and HCES 5000 Hollow Cathode electron source.
CHAPTER 8. LOW ENERGY ION BEAM ASSISTED DEPOSITION OF A GMR SPIN VALVE

The spin-dependent electron transport in giant magnetoresistive (GMR) multilayers is significantly affected by the atomic scale structure of their interfaces. Devices with atomically flat and chemically sharp interfaces are preferred for magnetic sensor and memory applications. The atomic simulations of the atom-by-atom assembly of these devices indicate that near optimal interfacial structures can be created using low energy, ion assisted vapor deposition techniques with ion energies in the 5-10 eV range. The biased target ion beam deposition system described earlier was designed to experimentally test this hypothesis. Prototypical Ta/NiFe/Co/Cu/Co/FeMn/Cu spin valve structures were first grown using (simultaneous) argon ion assistance during deposition of the Co/Cu/Co trilayer portion of the spin valve multilayer. Assisting ion energies of around 10 eV resulted in structures with a 30% higher magnetoresistance ratio and significantly reduced coupling field compared to the our samples grown with no ion assistance or with ion energies above 15 eV. These results are consistent with the atomistic simulation predictions. Other promising ion assistance schemes identified by the simulations were then used to deposit the Ta, NiFe, FeMn, and the top copper layer. A near optimal strategy was identified that resulted in further improvements to the GMR ratio.
8.1 Introduction

Ion beam assistance during the vapor deposition of a monolithic thin film is often used to enhance its surface smoothness [166]. However, atomistic simulations using MD techniques indicate that if this ion assistance strategy is used during deposition of multilayers, it is essential that low energy (less than a few electron volt) atoms or ions are used to avoid intermixing at interfaces [28, 43]. When interlayer mixing does occur, it can lead to a dead magnetic layer in GMR structures and increase electron scattering, both of which degrade the GMR effect [89, 210]. An optimum ion mass specific energy must exist because the use of too low an ion energy usually results in rough surfaces and interfaces, which also degrade GMR properties.

MD simulations of the low energy ion assisted deposition of Co/Cu multilayers have identified near optimal ion assistance recipes for the growth of high quality interfacial structures with reduced roughness and intermixing [43, 194, 211]. The atomistic simulations reveal that the energy barriers for surface flattening and interlayer mixing mechanisms are both low [43]. However, the critical ion energy for flattening is usually less than that for mixing under oblique ion illumination conditions [43]. Ion assistance with energies above the flattening energy but below the mixing threshold therefore appear promising [194]. For ions with a similar atomic mass to that of the metal (e.g., argon ions for the Cu-Co system), the use of a fixed ion energy of 8 eV at an incident angle of around 50° during ion assisted (simultaneous) deposition was predicted to give the best interfacial structures [43, 194]. Other ion assistance approaches including modulated ion energy (where the ion energy was gradually increased as each layer's
Chapter 8. Low Energy Ion Beam Assisted Deposition of a GMR Spin Valve

thickness increased) and sequential ion beam assistance (where the ion assistance was used after deposition of each layer) also appeared equally promising and enabled the use of higher energies once an interface was buried by several monolayers of the new metal [211].

Here, these model-predicted low energy ion assistance strategies are experimentally investigated using a specially constructed biased target ion beam deposition (BTIBD) system designed to access the low energy ion assistance regime [120, 207]. This vapor deposition tool enabled independent control of the ion energy, incident angle, and flux and facilitated the study of all three of the ion assistance schemes identified above. We find the GMR ratio and coupling fields of a model spin valve structure can be significantly improved using these ion assistance concepts.

8.2 Experimental Depositions

GMR spin-valves with a Ta/NiFe/Co/Cu/Co/FeMn/Cu structure were deposited by a biased target ion beam deposition approach using various ion assisting strategies. The samples were grown on the Si/SiOx wafer at ambient temperature in the BTIBD system schematically illustrated in Fig. 7.5. All wafers were cut into 1x2 cm sample size before deposition. Each of the metal layers of a prototypical spin valve structure was sequentially deposited without breaking vacuum. Magnetic field (around 60 Oe) was applied along with sample long edge while depositing layers of NiFe, Co and FeMn. A rotatable target holder held the five metal sputtering targets. These were sequentially argon ion sputtered using a low voltage, high flux ion beam, which is created by utilizing
an end Hall ion source and secondary ionization by a low voltage electron beam. The ions were accelerated towards a target by applying a bias voltage to the target. This facilitated near normal incidence sputtering ion impacts, which greatly reduced the probability of high energy reflected neutral fluxes [120, 212] and the sample contamination that results from their impact with the interior surfaces of deposition equipment hardware [207].

The ion source used for ion assistance was designed to obliquely illuminate the face of the substrate stage and enabled production of low energy assisting ion flux in the 0-50 eV range. Details of the ion source characterization can be found in previous chapters and other papers [40, 120, 208]. For all of the depositions conducted here, the assisting ion flux obliquely impacted the sample surface at 45°. The assisting ion to metal atom flux ratio was 4 for all the experiments. The base pressure established prior to deposition of all the films was about $3 \times 10^{-8}$ Torr. The working pressure during growth was about $4.5 \times 10^{-4}$ Torr. Sputtering was achieved using a target bias voltage of 300 V. The film thickness was controlled by deposition time based upon prior calibrations using X-ray fluorescence spectroscopy to determine film thickness (for instance, the deposition rate was 0.57 Å/s for Cu with a deviation of less than 3%).

Magnetoresistance measurements of spin valves were carried out at room temperature with a standard four-point-probe method in an external magnetic field. The external magnetic field is provided by a GMW electromagnet, which can generate magnetic field up to 3 T, Fig. 8.1. Samples were place between the poles and magnetic field could be controlled by the coil current. Fig. 8.2 shows the experimental apparatus that was designed for sample MR measurements in a vacuum after the deposition. A
constant sensing current of 1 mA was used and the voltage drops were measured under the external field applied in the film plane. Gold contact probes, which should have no MR effects in the measurement of samples, were used as the contacts in the experiments. Both the sensing current and the magnetic field were in the plane of the film in the MR measurement. External magnetic field was designed to be always aligned with the sample long edge, which is the same direction of magnetic field while depositing. Direction of sensing current was tested in two cases: arranged orthogonal or parallel to the direction of external magnetic field. Measurements in these two cases (orthogonal and parallel) gave similar results and the average value was taken from several tests. Resistance changes deduced from voltage drop changes were plotted as a function of external magnetic field (typically from -1000 Oe to 1000 Oe) to understand sample MR properties. The MR measurements were repeated several times and all the results were reproducible for both GMR ratios (± 3%) and the coupling fields (± 2%).

![Diagram of electromagnet](image)

Figure 8.1 Schematic illustration of electromagnet. High field and high uniformity magnetic field is generated between the poles. Samples were placed at the center of the poles in the MR measurements.
Figure 8.2 In-plane magnetoresistance properties were measured with a four point probe technique. The two outer probes were used to introduce a current, I, while the two inner probes were used to measure the potential difference between two positions along the current path. Both the sensing current and the magnetic field are in the plane of the film. Voltage drops, which can be used to deduce resistance, were measured as a function of magnetic field which arranged orthogonal (θ = 90°) or parallel (θ = 0°) to the current direction.

8.3 Results

8.3.1 Unassisted Deposition

The effect of Cu spacer layer thickness (t) in a Ta (40 Å)/Ni_{80}Fe_{20} (40 Å)/Co (15 Å)/Cu (t)/Co (45 Å)/FeMn (100 Å)/Cu (20 Å) spin valve structure deposited without ion assistance was explored first. Fig. 8.3 shows an example of the GMR curve measured for an Ar sputtered spin-valve with a Cu layer thickness of 33 Å. In the figure, the GMR ratio is defined as usual, resistance change divided by saturation resistance, ΔR/R_s, where
Chapter 8. Low Energy Ion Beam Assisted Deposition of a GMR Spin Valve

\( R_s \) is the saturation resistance in large magnetic field. This device relies on the exchange biasing between top Co layer and FeMn layer to magnetically pin the top Co layer. As a result, the combination of an exchange biased ferromagnetic layer (top Co layer) and a simple ferromagnetic layer (NiFe and bottom Co layers) in our spin valve system made it possible to engineer the magnetic moments of the two layers to be either parallel or antiparallel to one another as a function of magnetic field without relying on interlayer exchange coupling even with small magnetic field. As can be seen in Fig. 8.3, when magnetic field is around zero or very high (above 150 Oe, for instance), the resistance of spin valve is low because both magnetic layers are parallel to each other. On the other hand, when magnetic layers are antiparallel to each other (with magnetic field of around 70 Oe in Fig. 8.3), the resistance of the system appears to be high because of the significant electron scattering at the interfaces between Cu and Co.

![Graph](image)

**Figure 8.3** Resistance vs. magnetic field curve for a spin valve of Ta (40 Å)/Ni_{80}Fe_{20} (40 Å)/Co (15 Å)/Cu (33 Å)/Co (45 Å)/FeMn (100 Å)/Cu (20 Å) at room temperature. Target voltage was 300 V and processing gas was Ar.
The GMR ratio is plotted as a function of copper spacer layer thickness in Fig. 8.4. It can be seen that the GMR ratio of devices grown in this manner is sensitive to the spacer layer thickness and is consistent with results for films grown by other techniques [8, 61]. A relatively high GMR ratio of 6.972% was obtained at spacer layer thickness of 33 Å (Fig. 8.3). It is also consistent with measurements of other groups who have shown that the maximum GMR ratio for this class of spin valve usually lies within the range of 6–8 % [213-218]. As an aside, we note that the use of Ar for the sputtering gas rather than higher atomic mass sputtering ions, such as Xe, is thought to produce this good GMR ratio because of the reduced reflected neutral energy of near normal ion impacts and the low ion energy due to low target voltage (300 V) [182, 212]. This BTIBD method therefore appears to provide a means for creating high quality spin valves using lighter sputtering gases [212].

![Graph](image_url)

Figure 8.4 GMR ratio as a function of Cu spacer layer thickness for Ta (40 Å)/Ni$_{80}$Fe$_{20}$ (40 Å)/Co (15 Å)/Cu (t)/Co (45 Å)/FeMn (100 Å)/Cu (20 Å) spin valve deposited without ion assistance.
Spin valves with a 33 Å spacer layer thickness were then deposited using argon ion assistance with ion energies that were varied up to 20 eV. This series of experiments utilized simultaneous ion assistance during deposition of the basic Co/Cu/Co part of the spin valve structure described earlier. The other layers were deposited without ion assistance. The GMR ratio and coupling field are plotted as a function of ion energy in Fig. 8.5. It can be seen that the use of an assisting ion energy of approximately 10 eV resulted in the highest GMR ratio and the lowest coupling field. This ion energy for best performance agrees well with the optimum energy predicted by earlier atomistic simulations for creating flat unmixed interfaces and is consistent with and some experiments [40, 43, 211]. The highest GMR measured ratio (9.06%), Fig. 8.6, represented a significant increase over that of structures made without assistance (6.97%). It is also interesting to note that higher ion assistance energies around 20 eV reduced the GMR effect presumably due to the mixing that has been predicted to then occur at these interfaces.
Figure 8.5 GMR ratio and coupling field as a function of assisting argon ion energy. Simultaneous ion assistance was only applied during deposition of the Co/Cu/Co part of the Ta/NiFe/Co/Cu/Co/FeMn/Cu spin valve.

Figure 8.6 Resistance vs. magnetic field curves for Ta/NiFe/Co/Cu/Co/FeMn/Cu spin valves with simultaneous ion assistance on Co/Cu/Co layers using 10 eV Ar ion.
8.3.3 Other Depositions

In the simultaneous ion assistance scheme used above, both the ion and adatom (metal) fluxes arrive at the film growth surface simultaneously with a fixed ion/adatom flux ratio. The best ion energy for depositing the Co/Cu/Co trilayer was around 10 eV. Simulations have indicated that higher ion assistance energies give slightly better results if they are used once an interface is well covered by several monolayers of a new metal [211]. This overlay thickness is critical in both sequential and modulated energy ion assistance schemes because their use of high energy assisting ions can otherwise induce interfacial mixing.

<table>
<thead>
<tr>
<th>layer</th>
<th>thickness</th>
<th>ion assistance scheme</th>
<th>ion energy</th>
<th>ion/adatom flux ratio</th>
<th>ion flux (ions/Å²/s)</th>
<th>over-layer thickness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ta</td>
<td>40 Å</td>
<td>sequential</td>
<td>20 eV</td>
<td>-</td>
<td>0.18</td>
<td>40 Å</td>
</tr>
<tr>
<td>NiFe</td>
<td>40 Å</td>
<td>sequential</td>
<td>20 eV</td>
<td>-</td>
<td>0.18</td>
<td>40 Å</td>
</tr>
<tr>
<td>Co</td>
<td>15 Å</td>
<td>simultaneous</td>
<td>10 eV</td>
<td>4</td>
<td>-</td>
<td>0 Å</td>
</tr>
<tr>
<td>Cu</td>
<td>33 Å</td>
<td>simultaneous</td>
<td>10 eV</td>
<td>4</td>
<td>-</td>
<td>0 Å</td>
</tr>
<tr>
<td>Co</td>
<td>45 Å</td>
<td>sequential</td>
<td>20 eV</td>
<td>-</td>
<td>0.18</td>
<td>45 Å</td>
</tr>
<tr>
<td>FeMn</td>
<td>100 Å</td>
<td>modulated</td>
<td>20 eV</td>
<td>4</td>
<td>-</td>
<td>30 Å</td>
</tr>
<tr>
<td>Cu</td>
<td>20 Å</td>
<td>simultaneous</td>
<td>10 eV</td>
<td>4</td>
<td>-</td>
<td>0 Å</td>
</tr>
</tbody>
</table>

Using insights gained from atomistic modeling of the three ion assistance schemes [211], a multi-assistance strategy shown in Table 8.1 was developed for depositing the spin valve structure. In this proposed approach, the simultaneous ion assistance scheme was used for the deposition of the very thin Co and Cu layers (those with less than 10 atomic layers as shown in Table 8.1). This was chosen to avoid high ion energy induced
interlayer mixing. A sequential ion assistance approach was used during deposition of the relatively thick Ta, NiFe, and Cu layers (these are 10-20 atomic layers thick), and the modulated ion energy ion assistance scheme was used for the deposition of the thickest (FeMn) layer. The modulated energy ion assistance is preferred over the sequential ion assistance for the thickest layer because the maintenance of an assisting flux during deposition of thick films can avoid island build-ups from the flux shadowing effect. Using this near optimal ion assistance scheme, Ta/NiFe/Co/Cu/Co/FeMn/Cu spin valve structures were grown and their measured GMR ratio was found to increase from 9.06% to 10.51%, Fig. 8.7.

Figure 8.7 Resistance vs. magnetic field curves for Ta/NiFe/Co/Cu/Co/FeMn/Cu spin valves with combination of different ion assistance schemes using parameters listed in Table 8.1.
8.4 Discussion

Atomistic simulations have shown that the impact of assisting ions with a sample surface transfers energy from the assisting ions to surface adatoms. This enables athermal adatom hopping to more coordinated (local minimum energy) sites on the surface and the growth of more atomically perfect structures. If ions are applied simultaneously with adatom condensation, a step flow mode of growth can be induced and the vertical growth of islands suppressed [43, 194]. This minimizes the development of rough surfaces and ensuing flux shadowing for oblique components of the condensing atom flux. Intermixing occurs by atom exchange, which is activated by the vertical component of the ion’s momentum during an impact. The use of oblique assisting ion fluxes can enhance the surface flattening processes and reduces the probability of intermixing at interfaces formed as a new metal is deposited on an earlier grown surface. The most recent simulations indicate that argon ions with energies of around 8 eV and incidence angles near 50° result in near optimal interfaces for GMR applications [194]. The results obtained here appear to corroborate this hypothesis.

The simultaneous ion assistance scheme gradually becomes less effective as the layer thickness and thus average size of islands grows. More ion energy then needs to be transferred to the structure for continuous value of flatness to be maintained. This becomes less problematic for interlayer mixing as the interface between the two layers becomes more deeply buried and therefore isolated from an ion impact. As a result, modulated ion energy and sequential ion assistance schemes are superior for thick films [40, 211].
Chapter 8. Low Energy Ion Beam Assisted Deposition of a GMR Spin Valve

The study conducted here indicates that the use of a low energy, simultaneous ion assistance scheme for very thin layers, combined with higher ion energy modulated and sequential ion assistance schemes for the thicker ones can improve the interfacial structures of a model spin valve and lead to significant increases in GMR ratio. Biased target ion beam deposition appears to provide an effective means for the implementation of this strategy.

Finally, it should be noted that efforts to directly observe spin valve interfacial structures using transmission electron microscopy have proven difficult because of the lack of contrast between the Co and Cu layers. Next chapter will study the role of ion assistance in Cu on Ta structures where the electron scattering contrast is greater [40].

8.5 Conclusion

The effects of low energy, inert gas ion assistance during the deposition of a Ta/NiFe/Co/Cu/Co/FeMn/Cu spin valve structure have been experimentally explored. It was found that oblique, low energy argon ion assistance with an energy around 10 eV during deposition of the Co/Cu/Co trilayer portion of the spin valve significantly improves spin valve properties. Atomistic simulations indicate that this results from improvements to the interfacial structures of these multilayer systems. By using modulated energy and sequential ion assistance for the other (thicker) layers of the spin valve structure, devices with more than 50% higher GMR ratio than our samples without ion assistance have been fabricated.
CHAPTER 9. LOW ENERGY ION ASSISTED DEPOSITION OF Ta/Cu FILMS

A combination of MD simulations and experiments has been used to investigate the use of various low energy, ion assisted vapor deposition approaches for controlling the interfacial structures of a model copper/tantalum multilayer system. Films were grown using argon ion beam assistance with either a fixed or modulated ion energy during metal deposition. The effect of sequential ion assistance (after a layer’s deposition) was also investigated. The argon ion energy was varied between 0 eV and 50 eV and the effect on the atomic scale structure of Ta/Cu film interfaces and electrical resistivity was studied. The use of simultaneous argon ion assistance with an ion energy of ~10 eV and an ion/metal atom flux ratio of ~6 resulted in atomically sharp interfaces with little intermixing, consistent with simulation predictions. Ion impacts in this range activated surface atom jumping and promoted a step flow film growth mode. Higher energies were also successful at interface flattening, but they caused significant intermixing between the layers and increased the film’s resistivity. This could be reduced using modulated ion energy and sequential ion beam assistance. This was again consistent with atomic scale simulations, which indicated that metal layers deposited over an interface before ion assistance was initiated impeded atom exchange across interfaces and therefore intermixing.
9.1 Introduction

An atomically flat interface can only be created if the film surface of a previously deposited layer is already flat at the start of the deposition of the next layer. To fabricate devices like those identified above, a smooth buffer layer is first deposited on a semiconductor substrate. This is then followed by the controlled deposition of the metallic multilayer system of interest. The atomic scale flatness of the buffer layer surface and the structure of the interface between it and a subsequently deposited metal (especially intermixing), can significantly effect the magnetic properties and X-ray reflectivity of the deposited structures [219, 220].

Two approaches can be used to create an assisting ion flux. One exploits ion beam sources, such as the Kaufman ion gun [198, 221], to generate energetic ions that impact a substrate. Ion polishing is a well-established example of the use of this kind of ion source [222, 223]. In this process, ions are accelerated in an ion beam gun up to several hundreds or thousands of electron volts. These ions then strike the film surface at an oblique angle and preferentially sputter the most weakly bound atoms from the surface (those with the fewest bonds to neighboring atoms). The surface then becomes smoother due to the preferential removal of material from convex areas of the surface. In “ion polishing,” the energy of the ions is usually well above the sputtering threshold for the film. For transition metals, this is in the 50 eV range [212, 224].

The second approach creates a plasma to which a negative bias voltage is applied between it and the substrate [166, 189]. The combination of the plasma potential and the applied bias accelerates the ions from the plasma toward the growth surface where they
typically make impact with a near normal incidence angle. A low background pressure reduces ion energy loss by binary scattering but also reduces the ion flux. Magnetic fields are often then used to enhance the ion concentration. This magnetron sputtering method has been widely used for the deposition of single layer metal films [201, 202]. The deposition process can to some extent be optimized by modification of the negative substrate bias voltage. In the traditional single layer film deposition process, a high bias voltage of several hundred volts has been experimentally found to produce the smoothest films [166]. Although ion impacts with an energy in the range of hundreds of electron volts leads to a significant reduction of surface roughness, such impacts can also create other defects and induce alloying at interfaces within multilayer structures [28, 171]. Higher pressure sputtering processes utilize binary ion-background gas atom collisions to reduce the ion impact energies and have been quite successfully utilized for the growth of some metallic multilayers [34].

Several experimental and MD simulation studies indicate that even very low energy (a few electron volts) atom or ion impacts with a growth surface can result in defect creation and mixing at the interfaces in metal multilayers [28, 32, 43, 171, 194]. These imperfections significantly affect thin films properties. For example, interlayer mixing in magnetic films creates a magnetic dead layer at the interface that degrades the performance of GMR multilayers [89, 210]. Amorphous layer formed by solid state alloying between Ta and Cu also reduces the conductivity of interconnects [225].

The analysis of MD simulations in previous chapters indicates that the critical ion energies needed to overcome energy barriers for flattening are in the 2 eV range. They
also show that the critical ion energy for intermixing can be higher, especially if oblique incidence angles are used [43]. These simulations suggest several promising deposition possibilities for reducing interfacial roughness and interlayer mixing using low energy ion assistance [43, 176, 194]. For example, the simultaneous use of very low ion energies during vapor deposition facilitates the growth of a smooth surface by athermally activating the lateral surface diffusion of the surface atoms. Provided the ion energy is below the (ion mass and ion incident angle dependent) critical energy for intermixing, this can be accomplished without a loss of chemical sharpness across interfaces. This latter issue becomes much less important if ion assistance is delayed during the deposition of a new layer or postponed until the layer’s deposition is complete. Three approaches for deposition of high quality multilayers therefore appear promising: (i) simultaneous ion assistance with deposition (using very low ion energies at oblique incidence angles), (ii) modulated ion energy assistance (where the ion energy is increased after an interface is buried by several layers of new atoms), and (iii) sequential ion assistance where higher ion energies can be used for flattening after a layer has been deposited [43].

Here, we experimentally explore the use of the ion assistance strategies to grow a model Ta/Cu bilayer. This system affords good electron scattering contrast in high resolution transmission electron microscopy allowing direct observations of interfacial structures. Copper has also been extensively used as an underlayer (seed layer) in magnetic structures [226-229] and for deep-submicron interconnects [219, 220, 230]. In these applications, copper can easily diffuse into the surrounding silicon structures [230],
so a Ta diffusion barrier layer is frequently used to protect the underlying silicon devices [231, 232].

We began by briefly examining new MD simulation results for the ion assisted growth of Cu on Ta, which show that the system has a similar ion assistance response to the Co/Cu system studies that have been extensively simulated in previous chapters. The simulations reconfirm the potential value of the three ion assistance approaches and identify preferred ion energies and flux ratios. We then used a biased target ion beam deposition approach to deposit Cu/Ta films using the best ion assistance strategies identified by modeling. Transmission electron microscopy is used to characterize the interfacial structures, and electrical resistivity measurements are used to provide insight into the electron scattering defects contained in these bilayers. The results confirm that the very low energy ion assisted growth hypotheses identified by atomistic simulations are practically realizable using the biased target ion beam deposition approach.

9.2 MD Simulations of Ta/Cu Vapor Deposition

9.2.1 Simulation Method

Details of the simulation methodology used to model the low energy ion assisted vapor deposition of multilayer systems can be found in Chapters 3 and 4. It was modified here for the Ta-Cu system by using an initial β-Ta substrate crystal created from 12 (100) planes stacked in the x direction, 12 (004) planes in the y (growth) direction, and 12 (010) planes in the z direction. An embedded atom method (EAM) potential was used
for the calculation of the interatomic forces between the metal atoms [28, 150, 233] and a universal pair potential was used to define the interatomic forces between inert gas (argon) atoms and between metal and inert gas atoms [175]. Periodic boundary conditions were used in both the $x$ and $z$ directions to minimize the effects of the relatively small crystal size in these two directions. Inert gas ions and adatoms with a prescribed incident angle, $\theta$, and incident kinetic energy, $E_i$, were continuously injected toward the surface. To uniquely define a three-dimensional incident direction with a single parameter, $\theta$, it was assumed that the ions were injected in the $x$-$z$ plane. To avoid a crystal shift during ion/atom impact, the bottom three (004) Ta layers were fixed during the simulations. The two Ta monolayers above the fixed region were kept at a constant temperature of 300 K to create quasi-isothermal growth conditions. Sufficient time (at least 0.15 ps) between (adatom or ion) impacts was used for relaxation and dissipation of the impact energy so that the effects of the impacts could be considered independent.

### 9.2.2 Results

The simultaneous ion assistance simulations revealed the formation of a variety of defects at the Cu on Ta interface and within the Cu film. These arose from shadowing effects, atomic exchange across the Cu-Ta interface and the large lattice constant mismatch (7.6 % in minimum) between the (004) $\beta$-Ta and (111) fcc Cu. To characterize the film’s structure, the packing density (which is inversely proportional to the vacancy defect concentration), surface roughness, and the degree of interlayer mixing of the simulated Cu films were calculated and are plotted in Fig. 9.1 as a function of ion energy.
and ion:atom flux ratio using an ion incident angle of 45 degrees. To characterize the roughness of the simulated structures, an interfacial roughness parameter, $R$, was defined that depended upon both the height and the width of surface asperities [30]:

$$R = \sum_{i=1}^{n} \frac{h_{il} + h_{ir}}{2} / \sum_{i=1}^{n} w_i,$$  \hspace{1cm} (9 - 1)

where $h_{il}$ and $h_{ir}$ are, respectively, the height measured from the left and right of the $i$th asperity and $w_i$ is the width of this asperity. Summation is conducted over the $n$ asperities in the $x$ direction.
Figure 9.1 MD simulation results for Cu films deposited on a Ta buffer layer using a simultaneous ion assistance method: (a) Film packing density, (b) surface roughness, and (c) interfacial mixing as a function of assisting argon ion energies at a fixed ion-to-atom flux ratio and an ion incident angle of 45°.
It can be seen from Fig. 9.1 (a) and (b) that when the ion energy and ion/atom flux ratio were both low, the film density was also very low and the film roughness was high. Many adatoms that were deposited on surface asperities remained there, and numerous vacancy type defects became trapped in the film during its growth. In this regime, the kinetic energy transferred to surface adatoms was insufficient to overcome the Ehrlich-Schwoebel energy barriers at the terrace edges [43]. These adatoms then remained adsorbed on a surface asperity without undergoing ion impact induced diffusion. This low energy, low ion:atom flux ratio regime is referred to as the surface adsorption dominated regime. Deposition in the surface adsorption regime was responsible for the growth of structure containing high defect concentrations, low film densities, and rough interfaces.

The film density increased and the surface roughness decreased as the ion energy and ion/atom flux ratio were increased, Fig. 9.1 (a) and (b). A relatively high film density and low roughness were obtained for ion energies above 10 eV. At these ion energies, surface adatoms acquired sufficient energy from ion impacts to migrate over the terrace edges and eliminate surface vacancies. Resputtering was found to begin to occur as the ion energy was increased above 30 eV. The most weakly bound surface atoms were preferentially removed, which led to additional smoothing.

Mixing at the interface between the layers increased as the ion/metal flux ratio and the ion energy were increased, Fig. 9.1 (c). For all ion/metal ratios, intermixing was negligible until the ion energy reached a value of 5 eV. Above this ion energy, the mixing probability rose steadily with ion energy. This intermixing became very significant for
ion energies above 10 eV and ion/metal flux ratios above 5. Quite low levels of intermixing were observed over the entire ion energy range studied, provided the ion/metal ratio remained below 1.0.

Simulations of modulated energy and sequential ion assistance schemes on the Cu/Ta system revealed very similar results to previous Cu/Co simulations [176]. Fig. 9.2 shows film density, surface roughness ratio, and interlayer mixing as a function of argon ion energy. For the modulated energy ion assistance simulations, the first half of the Cu layer was deposited without ion assistance and the remainder was then deposited using the ion energy shown at an ion/metal flux ratio of 6. For the sequential ion assistance simulations, an ion fluence of 1.0 ions/Å² was applied after completion of deposition of the Cu layer.
Figure 9.2 Simulation results for Cu films deposited on a Ta buffer layer using modulated energy and sequential ion assistance schemes: (a) Film packing density, (b) surface roughness, and (c) interlayer mixing and surface roughness as a function of assisting argon ion energies at an ion incident angle of 45°.
Fig. 9.2 (a) shows that the film's density increased with ion energy for both schemes. The modulated energy ion assistance scheme was marginally superior. It resulted in a slightly higher level of packing density for most of the simulations. Fig. 9.2 (b) and (c) show that both approaches resulted in significant surface flattening and very low interlayer mixing even for an ion energy as high as 30 eV. The film's surface roughness was simultaneously decreased by almost a half. Detailed analysis of simulations indicates that the modulated energy ion assistance method was slightly more effective than the sequential assistance approach because ion assistance was initiated before significant roughening of a surface had developed.

9.3 Experimental Growth Approach

A recently developed biased target ion beam deposition [120, 207] system was used to explore the ion assisted growth of Ta/Cu bilayer films at room temperature. A schematic illustration of the system is shown in Fig. 7.5. It enables independent control of the assisting ion energy, ion incident angle, and ion flux. When a negative bias voltage was applied to the sputtering target, the inert gas ions were attracted and made a near-normal incidence with the target surface at an easily controlled (but low) energy. During target sputtering, the low sputter yield per ion impact was off-set by the use of a high ion current. The near normal ion incidence for sputtering greatly reduced the creation of energetic neutrals [212]. The negative bias reduced ion impact with metal tooling thereby eliminating overspill contamination [212].
Chapter 9. Low Energy Ion Assisted Deposition of Ta/Cu Films

To provide low energy (0-100 eV) ion assistance, the ion assist source was located at the top of the chamber as shown in Fig. 7.5. Experimental studies have indicated that the mean ion energy is approximately 60% of the anode voltage while the ion beam current is approximately 20% of the anode current. The ion flux and energy parameters used below are all derived from these relationships.

Bilayer systems consisting of 5 nm (±1%) of tantalum and 30 nm (±2%) of copper were deposited on 6-inch wafers using the three ion assisting methods (simultaneous deposition with ion assistance, modulated ion energy, and sequential ion assistance) identified by simulations. Each series of experiments used oblique argon ion assistance (θ = 45°) during or after metal deposition. The effects of varying the assisting ion beam energy and ion flux were investigated for each of them. For all of the depositions, the base pressure was $3 \times 10^{-8}$ torr, the Ar working pressure was about $4.5 \times 10^{-4}$ torr, and the target bias voltage was 300 V.

After deposition, the in-plane sheet resistance of the bilayer system was measured by four-point probe using a 49-point scan on Prometric omnimap resistance measurement system. The film texture was characterized using X-ray diffraction (XRD). Surface morphology and roughness were characterized using a tapping mode atomic force microscope (AFM). Film thickness was controlled by the calibrated deposition rate by using X-ray fluorescence spectroscopy (XRF). It was found that for the deposition conditions used here, the deposition rate for Cu was 0.6 Å/sec while that for Ta growth was 0.2 Å/sec. Auger Electron Spectrometry (AES) was used to characterize the film
composition. The interfacial structures were also characterized by analytical transmission electron microscopy (TEM).

9.4 Experimental Results

9.4.1 Simultaneous Ion Assistance

The surface roughness and in-plane electrical resistivity of films grown using simultaneous ion assistance are shown as a function of the assisting ion energy (for an ion-to-atom flux ratio of about four) in Figs. 9.3 (a) and (b). Several AFM images of selected samples are also shown in Fig. 9.4. It can be seen that film roughness and especially resistivity are sensitive to the assisting ion energy. The film surface roughness initially decreased as the ion energy was increased from 0 to 7 eV, which is obvious from the comparison between Fig 9.4 (a) and (b), and then remained constant with further increases in ion energy as shown in Fig. 9.3 (a). These results were in general agreement with simulations, Fig. 9.1 (b). Fig. 9.3 (b) shows that the film electrical resistivity at first decreased as the ion energy was increased from 0 to 10 eV but then started to increase as the ion energy was further increased. The electrical resistivity is effected by both the interface (and surface) roughness and the vacancy and impurity atom concentrations in the two layers (i.e., intermixing). The initial drop in resistivity is consistent with a decrease in roughness while the subsequent resistivity rise may be a manifestation of mixing. The solid line in Fig. 9.3 (b) is an interface structure function calculated from a combination of the simulated results (see Appendix C for its derivation). It combines the
atomic scale features of the structure observed in simulations in a form that approximates their effect upon the in-plane electrical resistivity.

Figure 9.3 Cu/Ta film properties as a function of assisting argon ion energy using simultaneous ion beam assisted deposition at an ion-to-atom ratio of 4 and incident angle of $45^\circ$: (a) film surface roughness and (b) film electrical resistivity. An empirical structure function, $U$, was deduced from simulation results and is shown as the solid curve in (b).
Figure 9.4 AFM film morphology dependence on assisting argon ion energy at an ion-to-atom ratio of 4 and incident angle of 45°: (a) no ion assistance, (b) ion energy of 7 eV, and (c) ion energy of 15 eV.

Film roughness and resistivity were also measured as a function of the assisting ion flux using an assisting ion energy of 10 eV. Fig. 9.5 (a) shows that an increase of the ion-to-atom ratio from 0 to 8 results in an ~50% reduction in surface roughness. The roughness then reached a near plateau as the flux ratio was increased above 10. This is again in good agreement with simulation studies (see Fig. 9.1 (b)). These showed that at a low ion flux, the assisting ions failed to move many of the adatoms on the surface and led to the accumulation of significant roughness. Simulations also concluded that a
significant ion fluence is needed to effectively flatten a surface by direct jumping and exchange mechanisms while still avoiding interlayer mixing (see Fig. 9.1 (c)).

Figure 9.5 Cu/Ta film properties as a function of ion-to-atom flux ratio in simultaneous ion beam assisted deposition at an argon ion energy of 10 eV and incident angle of 45°: (a) film surface roughness, and (b) film electrical resistivity. Solid line in (b) is for structure function, \( U \).
Fig. 9.5 (b) shows that the film resistivity initially decreased as the flux ratio was increased to around 6. A further increase in the ion flux then caused an increase in film resistivity. The initial decrease in resistivity is believed to result from impact induced adatom motion which helped to reduce vacancy type defects, Fig. 9.1. Again, solid line, corresponding to simulation results, predicted structure and captured the trends observed experimentally and will be discussed later.

Further insights into the effect of simultaneous ion beam assistance interfacial structures were obtained from AES measurements of the film composition as a function of depth. Three AES depth profiles were obtained corresponding to deposition with: (a) no ion assistance, (b) an ion energy, $E_{\text{ion}} = 10$ eV, and a flux ratio of 5, and (c) an ion energy $E_{\text{ion}} = 20$ eV and a flux ratio of 14, Fig. 9.6. In the study, the energies for the primary electron beam and the Ar beam used for the AES measurement were 5 and 3 keV, respectively. The data reveals that the sharpest interface was obtained using an ion energy of 10 eV and an ion-to-atom ratio of 5. The composition profile of the sample made without ion assistance was slightly wider consistent with interfacial roughness. The samples made using an ion assisting energy of 20 eV and an ion-to-atom ratio of 14 had the most diffuse interface, again consistent with simulation predictions of extensive intermixing during energetic ion assistance treatments.
Figure 9.6 Measured auger electron spectrometer depth profiles of composition for Cu/Ta/substrate films deposited using simultaneous argon ion assistance deposition at an incident angle of 45° with three different ion assistance conditions: no ion assistance, 10 eV and ion-to-atom flux ratio of 5, and 20 eV and ion-to-atom flux ratio of 14.

To further understand interfacial structures, TEM observations for the three samples with (i) no assistance, (ii) ion assistance with medium energy (10 eV) and flux ratio (5), and (iii) ion assistance with higher energy (20 eV) and high flux ratio (14) are shown in Fig. 9.7. Images were recorded on a JEOL 2010F operated at 200 kV in phase contract mode. The TEM analysis showed that copper had a strong (111) texture regardless of the ion assistance conditions. The Ta texture is harder to determine in bilayer structure because it is well below the surface and the film thickness is very small. Additional samples of single Ta layer were therefore grown using the same deposition conditions. Plain view TEM and X-ray diffraction studies all indicated that Ta films have a β (002) orientation in the growth direction. Examination of the interfacial structures for the
Si/Ta/Cu films grown with no assistance or with high energy and flux ratio assistance, Fig. 9.7 (a) and (c), indicates they had a more diffused interfacial region than the sample grown with medium energy (10 eV) assistance, Fig. 9.7 (b). We note that Fig. 9.7 (c) in particular exhibits a highly mixed interface following growth with high energy and high ion flux assistance (while one should consider the possibility that inter-mixed regions form during the 4 keV Ar+ milling that is standard in cross-sectional TEM sample preparation, we note that the samples in Figs 8 (a) and (c) were prepared under nominally identical conditions).
Figure 9.7 High resolution TEM cross section images of the Ta/Cu bilayer interfaces deposited using different simultaneous ion assistance conditions at incident angle of 45°: (a) no ion assistance, (b) 10 eV and ion-to-atom flux ratio of 5, and (c) 20 eV and ion-to-atom flux ratio of 14. The CuTa region in (c) consists of a heavily mixed solid state solution.

9.4.2 Modulated ion energy and sequential ion beam assistance methods

The modulated ion energy and sequential ion assistance approaches utilized ion assistance only after a layer had been partially or fully deposited. For the modulated ion
energy approach, 3 nm of each new layer were first deposited without any ion assistance. The remainder of the layer was then completed using simultaneous ion assistance. The ion-to-metal flux ratio was 6. For the sequential ion assistance experiments, an ion surface modification technique was used after each entire layer had been deposited. The ion current was 0.1 mA/sec and was applied for between 30 seconds and 120 seconds to the completed tantalum and copper surfaces.

The surface roughness and film electrical resistivity dependencies upon ion energy are shown in Fig. 9.8 (a) and (b). Both ion assistance methods exhibited similar trends with ion energy. The surface roughness at first decreased significantly as the ion energy was increased from 0 to about ~25 eV and then slightly increased for energies above 30 eV. This is again generally consistent with simulation results, Fig. 9.2 (b). Fig. 9.8 (b) shows that the film electrical resistivity quickly decreased as the ion energy was increased from 0 to 10 eV. It then remained roughly constant before increasing slightly above 30 eV. The resistivity decrease in Fig. 9.8 (b) is consistent with the surface roughness decrease in Fig. 9.8 (a), while the re-increase of resistivity is a result of the resputtering mechanism, which is confirmed by AFM studies of film thickness and morphology.
Figure 9.8 Cu/Ta bilayer properties as a function of ion energy in modulated ion energy and sequential ion beam treatments: (a) film surface roughness, and (b) film electrical resistivity. The ion-to-metal flux ratio was 6 in the modulated ion energy scheme and an ion density of 1 mA/sec was applied for about 30 and 120 seconds to the completed tantalum and copper surfaces, respectively, in the sequential ion assistance.
Chapter 9. Low Energy Ion Assisted Deposition of Ta/Cu Films

9.5 Discussion

The atomistic simulations indicate that when low energy argon ions impact Cu or Ta surfaces, they transfer energy to surface atoms, thereby helping them overcome energy barriers and find local minimum energy (more tightly bound) sites. During simultaneous adatom condensation and ion assistance in the 10 eV range, the transferred energy is sufficient to overcome energy barriers, and impact occurs frequently enough to suppress the rapid vertical growth of islands and promotes step flow growth [43, 194]. The reduced height of local surface asperities in turn minimizes surface shadowing during deposition.

The simulations of ion impacts identify three regimes of assistance. At low ion energies and low ion flux, the surface morphology is determined mainly by adatom adsorption. In this adsorption regime, the size of islands gradually increased during growth since the ion impacts were insufficient to migrate all surface atoms to empty lattice sites. At intermediate ion energy and ion flux, the film growth mode enters a densification regime as vacancy trapping is reduced and flattening starts to become significant. Finally, an intermixing regime occurs at high ion energy and ion flux conditions where significant vertical atom exchange occurs. The adsorption and densification regimes are responsible for the film surface roughness decrease shown at low ion energy/flux in Figs. 9.3 (a) and 9.5 (a).

Figs. 9.3 (b) and 9.5 (b) have shown that the film resistivity decreases as either the ion energy or flux ratio are increased. It eventually reaches a minimum and then increases as the ion energy/flux is further increased. These changes were not the result of variations
in the Cu/Ta fraction of the film. The Cu:Ta resistivity ratio for perfect materials is around 1:60. In our studies, only in-plane resistances were measured and it then can be assumed that almost all the current flowed through only the top Cu layer. The measured resistivity is therefore a function of the Cu layer’s structural properties.

Many studies [234, 235] have shown that resistivity of thin films is a sensitive indicator of the concentration of defects, such as vacancies and voids, and to the presence of impurity atoms like those introduced by intermixing at film interfaces. Surface and interfacial roughness can also contribute to increases in resistivity [236-240]. It is therefore anticipated that the resistivity changes observed in the experiments are related to changes in surface (and interfacial) roughness, the film’s packing density (vacancy and void content), and alloy scattering due to interlayer mixing. All were evident in the simulations and their combination reached a minimum at about the same ion energy observed for resistivity. The resistivity of a layer containing several different electron scattering defects can be estimated by combining models for electron scattering from the surfaces, vacancy type defects and impurities (see Appendix C). The resulting “structure function” combines these structural parameters in a similar functional form to their effect upon the resistivity and is given by:

\[
U = A \left(1 - \frac{\lambda_c \text{Erf} \left( \frac{4\pi h / \lambda_c}{8\sqrt{\pi}} \right)}{\lambda_c} \right) + Bn_i + C\delta \tanh \left( \frac{\Delta}{2\delta} \right),
\]

(9.2)

where \(h\) is surface roughness, \(n_i\) is the vacancy defect concentration, \(\delta\) is intermixing zone width, \(\Delta\) is film thickness related parameter, \(\lambda_c\) is Fermi wavelength, \(\text{Erf}\) is an error function and \(A, B,\) and \(C\) are fittable constants. The values of this \(U\)-function with substitution of all the structure parameters deduced from the simulations (Fig. 9.1) are
plotted as the solid lines in Figs. 9.3 (b) and 9.5 (b) using a normalized scale. It can be seen that the trend of $U$ with ion energy and flux matches well the experimental resistivity observations.

Comparison between Fig. 9.3 and Fig. 9.8 indicates that the ion energy required to flatten a surface using either the sequential or modulated ion energy ion treatments is higher than that needed using the simultaneous ion assistance approach (7 eV versus 20–30 eV). This arises because during the sequential and (to a lesser extent) modulated energy approaches, significant roughness developed prior to the ion irradiation. This roughness then required relatively high ion energy fluxes to achieve smoothing [176]. These higher energy ion treatments are, however, less likely to cause interlayer mixing because the interface was located well below the surface, Fig. 9.2. Simulations have previously indicated that interlayer mixing in thin multilayer films can be significantly suppressed by even just a few atomic layers of new material [176]. The resulting resistivity and structure function, $U$, changes in Fig. 9.8 in the 10-30 eV ion energy range are smaller in the sequential or modulated ion energy assistance approaches than those encountered with simultaneous ion assistance. When the ion beam energy became too high, significant resputtering did occur (the sputtering threshold energy for copper is approximately 20 eV). This resulted in both an increase in the film surface roughness and film resistivity at ion energies around 50 eV (the sputtering yield of copper is 1% per incident ion at an argon ion energy of 50 eV [224]), Fig. 9.8.
9.6 Conclusions

Low energy ion assisted vapor deposition of Ta/Cu bilayer films was explored using both experimental and simulation approaches. Low energy assisting ions were found to significantly effect film structures and electrical properties. It has been found:

1. During simultaneous ion beam assistance with metal deposition, film roughness and resistivity at first decrease as the ion energy/flux ratio increases. The use of ion energies around 10 eV and an ion flux ratio around 6 optimizes film structure properties. Further increases of the ion energy or flux ratio again increase film resistivity by inducing intermixing at the Cu/Ta interface.

2. AES and TEM studies have confirmed simulation results and indicate that the interface between Ta and Cu films is non-planar due to physical roughness when films are grown with no ion assistance and they are heavily intermixed when the ion energy and ion flux were high. The sharpest interfaces were achieved at intermediate ion energy and ion flux.

3. Atomistic simulations of vapor deposition have identified regimes of adsorption, densification, and intermixing during ion assisted deposition with increasing ion energy and flux ratio.

4. A structure function obtained from simulation is found to account well for the film resistivity dependence on ion energy and flux ratio.

5. Ion energy for minimum roughness in modulated ion energy and sequential ion assistance were found to be higher than that of a simultaneous ion assistance approach because the islands formed in the latter approaches are bigger. However,
the use of higher ion energies in these two schemes is less likely to introduce mixing between the layers.

6. Significant resputtering at an ion energy above 40 eV appears to be responsible for increases in roughness and resistivity when higher ion energy assisted deposition conditions are used.
CHAPTER 10. CONCLUDING REMARKS

Manipulation of the atomic scale structures of thin films during their growth by vapor deposition has resulted in many materials with exceptional functionality including giant magnetoresistance studied here [1], tunneling magnetoresistance [20], tunable optical emission/absorption and reflection [241], and ultralow thermal conductivity [242]. Development of these new materials has led to many new devices, such as magnetic sensors and memories [20], quantum dot lasers [243] and various detectors [244], and novel thermal/chemical protection systems [245]. The manipulation of layered, thin film structures at the nanoscale level is also at the heart of efforts to maintain microelectronic industry on its Moore’s law trajectory [246], to create other functional thin films [39], and to synthesize materials with higher thermoelectric figures of merit [247].

The introduction of many of the concepts above is paced by the development of vapor deposition processes that enable control of atomic assembly so that the appropriate atomic scale structures are created. Difficulties arise because many of the ideal nanostructures are thermodynamically unstable. At low homologous substrate temperature, vapor deposition processes create thermalized structures that are far from equilibrium. Therefore, the use of techniques that facilitate atom migration can manage precise control of the kinetic evolution during vapor deposition. However, many phenomena (temperature, atom/ion energy) can be used and so a large multidimensional space of many interrelated processing parameters exists for controlling structure. In some cases, existing experimental methods may not even access the most desired part of this processing space. Accelerating the convergence of a synthesis process for metal
multilayers can be obtained by integration of atomistic simulations with experimental efforts.

Experiments carried out over many years have indicated that sputtering-based methods, such as magnetron sputtering and IBD, produce better GMR multilayers than thermal evaporation methods such as MBE [42]. This arises because thin film deposition is basically a thermodynamically unstable process. Studies of the atomic assembly mechanisms indicated that adatom mobility plays a key role in thin metal layer deposition [30]. Low adatom mobility deposition processes, such as MBE, result in rough interfaces. However metal multilayers can be significantly improved by using hyperthermal energy deposition where the hot atoms promote flattening of the interfaces. However, high adatom energy facilitate adatom exchange with underlying atoms and therefore causes interlayer mixing [30]. Good nanoscale multilayer deposition process must therefore be optimized.

It has been recognized since the 1960s that bombardment of a growing thin film by energetic particles also strongly influences film properties [119, 248-251]. Energetic inert gas ions transfer their kinetic energy to atoms near the impact site and induce local heating. It is easy to understand that surface roughness will be reduced by increasing the assisting ion energy. However, use of too high ion energy during ion bombardment can also cause interfacial mixing in thin film deposition processes and results in the significant intermixing of cobalt atoms in copper [43, 194]. On the other hand, it is also difficult in many of the plasma-based ion assistance techniques to separate out the degree to which ion and neutral particle fluxes as well as ion energies affect resultant film
properties. An optimum ion assistance method that promotes interface sharpness in nanoscale films has yet to be developed.

This dissertation has presented a systematic series of computer simulations and experimental studies to investigate low energy, ion assisted vapor deposition of metal multilayer systems. The process conditions and methods used to deposit metal multilayers with sharp interface structures have been systematically investigated. An MD technique has been mainly used to investigate the ion assisted growth process and to identify the atomic assembly processes during vapor deposition. The results of simulations of the low energy assistance approach enabled a clearer understanding of the atomic assembly mechanisms during growth of thin metal films. The insights developed from the simulation also suggested alternative growth strategies for improved metal multilayers, including modulated ion energy and sequential ion assistance methods. They indicate that the interface structures can be protected from relatively high energy ions by providing several atomic layers' coverage over the interface.

A systematic series of experiments have been conducted to evaluate the dependence of film structure upon ion assistance and to compare these results with those of computer simulations. Model Cu/Co, spin valves and Ta/Cu thin films were grown under various ion assistance conditions and their surface morphology, magnetic, and electrical properties investigated. These investigations confirmed the simulation hypothesis and the importance of interface roughness and intermixing in metal multilayers. They further revealed that the interface structure in metal multilayers can be varied by modifications to assisting ion conditions. The transfer of translational energy from assisting inert gas ions
to atoms on the growth surface can be used to controllably activate atomic hopping under otherwise kinetically constrained growth conditions. These investigations confirmed the importance of interfacial structures in thin film multilayers and further emphasized its importance to ongoing efforts to create more sensitive magnetic field sensors and magnetic random access memory.

As a result of the work conducted for this dissertation, many specific conclusions can be drawn:

1. Studies of the use of ion assistance with a model system consisting of Co cluster on flat Cu surface show that ion energy plays a significant role upon surface cluster flattening. Increasing the ion energy reduces surface roughness, but high energy ions cause mixing at the Co/Cu interface.

2. The assisting ion mass also has a significant effect on both surface flattening and interlayer mixing. For instance, in the Co-on-Cu system, flattening ceases at assisting ion energies of 5 eV, 7 eV, 10 eV, and 20 eV for Xe, Ar, Ne, and He, respectively. Intermixing started at 4 eV for Xe, 6 eV for Ar, 8 eV for Ne, and 18 eV for He.

3. It is possible identify an optimal ion energy for a minimum combination of roughness and intermixing. It is approximately 5 eV for Xe ions and rises to 18 eV for He ions in the Co cluster on Cu surface system using ions that arrive perpendicular to the surface.

4. Molecular statics studies of the energy barriers impeding atomic assembly indicate that exchange mechanisms normally have a much lower energy barrier
than those of the direct hop mechanisms of the surface flattening. Recoil mechanism, exchange mechanisms, and direct hop mechanisms are the main mechanisms responsible for flattening and are sequentially activated as the assisting ion energy is increased.

5. When the ion energy is too high, the exchange mechanism at interface between two layers occurs. However, its activation is a sensitive function of the surrounding lattice perfection (i.e., exchange mechanism can occur at much lower energies in the vicinity of terrace edges).

6. Surface flattening and intermixing are also a sensitive function of the assisting ion fluence. The surface roughness eventually approaches zero with increase ion fluence provided the ion energy exceeds the critical value for flattening. Interlayer mixing tends to reach a fully alloyed saturation value when a high ion fluence, high energy ion assistance scheme is used.

7. Atomic reassembly mechanism studies indicate that the transferred momentum from assisting ions can also be adjusted by adjusting the ions incident angle. The incidence angle has different effects upon the various mechanisms of assembly and so it is therefore possible to select different mechanisms by choosing a combination of the ion incident angle, ion energy and ion mass.

8. A "channeling effect" plays particularly important role in atomic exchange between Co and Cu layers as the ion incident angle is changed. The energy barriers for atomic exchange are found to be lowest for the low-index crystal
directions, such as $<110>$ and $<112>$. Interlayer mixing can to some extent therefore be controlled by the ion incident angle.

9. A two-body collision model is found to account well for the functional dependency of flattening and mixing upon ion energy, mass, fluence, and incident angle found from molecular dynamics simulations.

10. The simulations indicates that optimum Cu/Co/Cu multilayer structures can be deposited using low energy ion assistance using a moderate ion energy, an oblique angle, a moderate ion fluence, and ion masses that are comparable to the mass of the metals.

11. The ion assistance simulation results indicate that modulated energy and sequential ion assistance methods are also effective methods that protect an interface from intermixing due to relatively high energy ions impacts. The modulated energy ion assistance approach uses low ion energy or no assisting ions to deposit the first few atomic layers of each new layer and higher ion energies for the remainder of that layer. The sequential ion assistance approach only applies high energy assisting ions after completion of the metal layer’s deposition. Simulations indicate that very low interlayer mixing was observed while keeping low surface roughness using either of these two methods.

12. Interlayer mixing-interface depth mapping provides a convenient means of identifying the selection of ion energy and layer thickness in both modulated energy and sequential ion assistance. For instance, four monolayers of cobalt must be deposited before any ion impact if the smallest ion energy available is 20 eV.
13. Based upon insights from simulations, a new biased target ion beam deposition technology has been developed and proved to have many advantages over existing techniques, including low overspill contamination, high uniformity, controllability of ion energy and flux, wide selection of adatom energy, variable ion incident angle, and clean operating conditions.

14. Application of low energy ion assistance during the deposition of a model Ta/NiFe/Co/Cu/Co/FeMn/Cu spin valve structure improved spin valve properties and confirmed the simulation hypotheses. Simultaneous Ar ion assistance during the deposition of Co/Cu/Co layers using an ion energy ~ 10 eV achieved a 30% improvement in GMR ratio. By combining three different ion assistance schemes on different thickness layers, including simultaneous, modulated energy and sequential ion assistance, devices with more than 50% higher GMR ratio than our samples without ion assistance were been fabricated. These improvements confirm the importance of interfacial structure upon magneto transport properties.

15. In order to obtain direct observation of the structural improvements, relative thick Ta/Cu bilayer films were grown using low energy ion assistance. This system affords good electron scattering contrast in high resolution transmission electron microscopy (HRTEM) allowing direct observations of interfacial structures.

16. Both auger electron spectroscopy and TEM observations confirmed the simulation results that the sharpest interfaces are achieved at intermediate ion energy and fluence. High ion energies or fluence caused significant mixing between Cu and Ta layers and created thick solid solution regions.
Chapter 10. Concluding Remarks

17. Measurement of Ta/Cu film resistivity indicated it to be a sensitive function of the ion energy and flux, and was well predicted with a "structure function" obtained from the simulations.

18. The simulations and experiments indicate that the modulated energy and sequential ion assistance schemes need some what higher ion energies to reach the minimum surface roughness compared to the simultaneous ion assistance approach. The modulated energy ion assistance resulted in a slightly better flattening effect than the sequential ion assistance approach in most cases studied. Both methods are less likely to introduce mixing between the layers than the simultaneous ion assistance method.

Finally, the combination of computational simulations and experimental works described in this dissertation showed a way to design the processes. As discussed in previous chapters, the materials that are being synthesized today are not well optimized, and have all been processed using first generation approaches. The proposed combination work is a high risk if the fundamental understanding of the inter-relationships between atomic scale structure and application relevant properties is absent for these processing. However, it is potentially very high payoff endeavor that promises to unravel the complex processes that lead to the growth of the structures, defect populations and internal stresses responsible for high performance films and allow us to design optimal processes for their synthesis. Predictive process modeling can play a significant role in material development where particularly large combinations of materials and conditions need to be explored. The modeling also allows us to quickly explore a wide range of
processing conditions, or even those that are not accessible with the currently available experimental settings. It is envisioned that these simulations will give insight for improving the processing methods.
APPENDIX A. MOLECULAR DYNAMICS INPUT FILE

A.1 Input File

In molecular dynamics (MD) simulation of deposition, the positions of the atoms are solved from Newton’s equations of motion. A computational crystal is created by assigning atom positions to lattice sites, see Fig. A.1. To extend the lateral crystal size, periodic boundary conditions are used in the \( x \) and \( z \) directions where atoms at one boundary interact with the images of the atoms at the other boundaries (by periodic translation). In the code, this is implemented by filling atoms into entire periodic length perub-perlb, where perub and perlb are the high and low boundaries. Free \( y \) surface is used for deposition. In the code, this is implemented by filling atoms into a small part of the periodic length so that atoms at the top surface do not see the images of the bottom surface (which will be translated to beyond the cut-off distance of the potential). Atoms and molecules with a given kinetic energy are randomly injected to the top \( y \) surface at incident directions defined by \( \theta \) and \( \phi \), as defined in Fig. A.1. The frequency at which adatoms are injected corresponds to the user defined deposition rate \( \nu_{\text{dep}} \). To prevent crystal shift during impacts, atom in the bottom region \( d_0 \) is fixed. A middle region \( d_1 \) is kept at a constant temperature for isothermal deposition. This leaves a free region at the top \( y \) surface to accurately simulate the process. To keep this free region approximately at a constant volume, the temperature controlled region expands during simulation. Because surface roughness may develop, the expanding rate of the temperature controlled region
TbndV is little slower than deposition rate vdep to prevent the isothermal region from exceeding any surface.

Figure A.1 The computational cell used to simulate vapor deposition in the MD FORTRAN code.

The a.i file is composed of a number of cards listed in a predesigned sequence. Each card pertains to a given category and enables users to define a set of variables to perform certain functions. The card starts with &cardname and ends with &end (see the example a.i). No sequence is required for the variables within each card. In the following, we explain the cards in the sequence in which they should be written in the a.i. Not all the variables that can be defined in each card are explained. Please reference the MD code in Appendix B to identify their functions.

**prntcard**: Define how the output is constructed. *Variables in this card:*
1. printf: a character string representing the name of an output file that contains information, such as energy of the crystal, stresses, volume, temperature, the conditions of the jobs, starting and finishing times of the runs, etc.

2. rstrtf: a character string representing the name of an output file that contains the final crystal (i.e., the positions and velocities of all the atoms as well as crystal boundaries). If rstrtf is set to "none," no rstrtf file will be created.

3. conff: a character string representing the name of a movie file that contains many intermediate crystals. The conff file is unformatted and can be converted to the rstrtf type of file. If conff is set to "none," no conff file will be created.

4. iconf: iconf is normally set to 1, Set to 2, 3, and 4 provides option to output more information (such as stresses, temperature etc) in conff.

headcard: Allow the user to give a title for the run. Variables in this card:

1. header: a character string representing a title for this run. This title will appear in the three output files.

funccard: Define the potential functions for one element. Users can include more funccard to do alloy simulations but will need an empty funccard to indicate the end of the element list. Variables in this card:

1. atomtype: a name of one of the elements defined in eam file. This element will be used in the simulation.

vaporcard: Define the vapor deposition conditions. Variables in this card:

1. ivapor: ivapor = 1 simulates vapor deposition. ivapor = 0 simulates the annealing of the crystal without vapor deposition.
2. threedim: normal is set to 1, representing three dimensional calculation.

3. type1: the code allows the user repeatedly deposit two different layers. Type1 contains five integers (may be changed in the source code) defining the species in the first layer during the multilayer deposition. Example, if type1 is set to 1, 2, 5, 0, 0 and the funccards have defined elements in such a sequence: Cu, Ni, Al, Co, Fe, Zn, then the first layer will contain first, second, and fifth element Cu, Ni, Fe.

4. fraction1: five float numbers, defining the relative content of the element. In the example above, if fraction1=0.65, 0.15, 0.20, 0.0, 0.0, then the first layer will contain 0.65Cu, 0.15Ni and 0.20Fe.

5. type2: similar to type1 but is used to define the second layer during multilayer deposition.

6. fraction2: similar to fraction1 but is used to define the second layer during multilayer deposition.

7. evapor: adatom energy in eV. The recommended value for the highest energy is between 0.1 and 5.0 eV (highest energy that can be achieved during simulation is evapor+devapor, see below). The need to use high fluxes and small crystals limits the energy range that can be simulated.

8. depdir: three float numbers, \(-\sin \theta \cos \varphi\), \(-\cos \theta\), and \(-\sin \theta \sin \varphi\), define the incident directions.

9. vdep: deposition rate, in units of mm/sec. Normally one can 5000.0-10000.0. Lower values always get better results but take longer time to calculate. Higher values may get unsatisfactory results.
10. TbndV: expanding rate for isothermal region. If the number is positive, 80–95% of the vdep.

11. devapor: a float number with default equal to 0.0 for constant energy deposition. Non-zero value allows the modulated energy deposition. For instance, if evapor is set to 0.1 and devapor is set to 4.9, then the first half of a given layer is deposited at 0.1 eV and the remaining half of that layer is deposited at evapor+devapor = 5.0 eV. After the layer is finished, the energy is reset to evapor = 0.1 eV for the deposition of a new layer.

12. subspin: a float number specifying the real experimental substrate rotation rate in turns/min.

13. vdepreal: a float number specifying the real experimental deposition rate in Å°/ps.

14. timelayer: contains two float numbers, defining the times (in picoseconds) used to deposit each layer. This will determine the thickness of the layers (thickness equals time multiplied by vdep).

_initcard:_ Used to define the initial crystal. *Variables in this card:*

1. initf: a character string, represents a file (in rstrf format, see a later section for an example) that contains the initial crystal. This allows the simulation to be continued from a crystal obtained from a previous calculation. If no previous crystal exists, initf can be set to “none.”

2. genlat: can be set to either .true. or .false.: .true. means a new crystal will be created (in which case the initial crystal defined in initf is ignored); .false. means no new crystal will be created.
3. genvel: can be set to either .true. or .false.. If it is set to .true., the crystal
temperature (i.e., atom velocities) will be reinitialized to the temperature defined
in velcard.

4. scale: three float numbers with default set to 1.0, 1.0, 1.0. The scale can be set to
other values to induce strains of the initial crystal in the x, y, and z directions.

latcard: Used to define the crystal if the lattice is to be created (see initcard). Variables
in this card:

1. latttype: a character string specifying lattice such as “fcc” and “bcc.”

2. aivec, bivec, civec: each contains three float numbers that specify the coordinates of
an atom in the primitive cell. Hence, aivec, bivec, civec can allow users to create a
crystal by defining the coordinates of all three atoms in a primitive cell of a
crystal. This overwrites the crystal defined with latttype, which is useful because
only fcc and bcc structures are currently available in the latttype mode.

3. alat: a float number for the lattice constant in angstrom.

4. xrot, yrot, zrot: each contains three float numbers, specifying the directions in the
x, y, and z axes.

5. perub, perlb: the high and low periodic boundaries. The recommended values for
perlb are −0.5, −0.5, −0.5. A slight shift from zero avoids the coincidence of an
atomic plane sitting on the boundary, hence giving a better view when the crystal
is visualized. At a given perlb, perub has to be chosen to account for the
periodicity of the crystal (in the x and z directions only). Assume that the crystal
structure is fcc and the x direction is [112]. In this direction, the (224) planes have
the shortest spacing \( d = \frac{a_{\text{lat}}}{\sqrt{2^2 + 2^2 + 4^2}} \), and the stacking sequence of the (224) planes is ABCDEF ABCDEF ..., perub(1)–perl(1) must therefore be a multiplicity of \( 6 \times d \). The recommended value for perub(2) is 900.0 for vapor deposition (a big enough value to create a free surface).

6. xbound, ybound, zbound: each with two float numbers for the low and high bounds within which atoms are filled. Default values are set to fill the entire periodic length. So for vapor deposition, only ybound needs to be specified. For instance, ybound(1) is set to the same value as perl(2), and ybound(2) is set to the value so that ybound(2)-ybound(1) is equal to the initial thickness of the substrate.

7. ncell: an integer to specify number of sublattices. Default is 1. In some cases, it may be desirable to create more sublattices. For example, diamond structure consists of two interweaving fcc lattices.

8. rcell: consists of 30 (may be changed in the source code) float numbers, with three consecutive numbers specifying the coordinates of the origin of each of a total of 10 sublattices.

9. ccell: consists of nelmax ×10 float numbers, where the value of nelmax can be found in the include file of the source code (see Appendix I). The consecutive nelmax numbers can completely define the composition in one sublattice (similar to the fraction1 and fraction2 in vaporcard). A total of 10 sublattice compositions can be defined.
Appendix A. Molecular Dynamics Input File

10. iseed: a variable to hold the random number seed for statistically varying the calculations.

velcard: Used to define the initial temperature when the genvel in the initcard is set to .true.. Variables in this card:

1. temp: the initial temperature in K.

bndcard: Define the boundary mode. Variables in this card:

1. ibdtyp: when set to 1, the periodic length is fixed during simulation. When set to 2, the periodic length can change, which relaxes the stress.

2. dpress: external hydrostatic stress.

3. dstress: three float numbers, specifying the external normal stresses in the three coordinate directions.

4. idynper: three integers with default values of 1, 1, 1. When ibdtyp is set to 2, the user can turn individual periodic length into fixed length by setting the corresponding idynper to 0.

neicard: Define the way the neighbors of atoms are found. Variables in this card:

1. dradn: just a numerical value that can change the calculation efficiency (try 0.1–1 angstrom range or maybe beyond to see if the calculation is faster).

defcard: Used to create some defects. Users can include more defcard, but need an empty defcard to indicate the end of the list. Variables in this card:

1. xmin, xmax, ymin, ymax, zmin, zmax: specify a region within which some changes are to be made to the atoms. Default is the entire periodic cell.
2. oldtype: an integer, further indicating that in the range defined above, changes are
made only to the atoms that are the oldtype in the funcard (default is all atoms).

3. delpos: three float numbers indicate the displacement magnitude of these atoms in
the three coordinate directions.

4. newtype: an integer indicating that those atoms types will be changed to newtype,
as defined in the funcard. If newtype is set to 0, those atoms will be removed.

**fixcard**: Used to define how atoms are fixed. Users can include more fixcard, but need an
empty fixcard to indicate the end of the list. **Variables in this card:**

1. xmin, xmax, ymin, ymax, zmin, zmax: specify the region to apply the fix. Default
   is the entire periodic cell. In order to apply the fix shown in Fig. K.1, only ymax
   needs to be defined: ymax = perl2 + d0.

2. vector: three float numbers defining a direction.

3. mode: an integer. mode = 0 means that the atoms are not fixed. mode = 1 means
   that atoms are fixed in the vector (defined above) direction but are allowed to
   move in a plane normal to this direction (i.e., takes away one degree of freedom).
   mode = 2 means that atoms can only move in the vector direction but are fixed in
   the plane normal to that direction (i.e., takes away two degrees of freedom). mode
   = 3 means that atoms are completely fixed (takes away all three degrees of
   freedom). For the case shown in Fig. K.1, mode = 3.

**tmpcard**: Used to define how the temperature is controlled. **Variables in this card:**

1. ifxtmp: an integer from 1 to 3 that activates one of the three numerical methods
   for controlling temperatures. We have used ifxtmp = 1 extensively in the past.
2. follow: if follow = .false., then the temperature control algorithm is applied to regions defined in regcard below. This is also the case shown in Fig. K.1. If follow = .true., then the temperature control algorithm is applied to atoms. Note that although these atoms are initially determined in a defined region, they can move out of this region.

**regcard**: Used to define the temperature control regions. Users can include more regcards but need an empty regcard to indicate the end of the list. *Variables in this card*:

1. xmin, xmax, ymin, ymax, zmin, zmax: define the region to apply the temperature control.

2. destmp: the desired temperature in K for that region.

3. tmptim: determine how fast temperature equilibrium is achieved. The default value is recommended unless a problem arises.

**avecard**: Define how the average properties are calculated. The properties are averaged over outputs at an outtim (defined in intcard below) interval. *Variables in this card*:

1. eqtim: the average properties are calculated only after a period defined by eqtim to allow the system to reach equilibrium.

**intcard**: Define how the numerical calculations are done. *Variables in this card*:

1. tottim: total simulation time in ps. In the sample a.i file, 10000 ps is used as the simulation time, taking about three weeks to finish the run.

2. outtim: time interval to output the data to the conff file.

3. dt: time step to numerically integrate Newton’s equations of motion is typically 0.001–0.004 ps. Use small values for higher rate and energy deposition.
4. inte: an integer value, specifying the mode of calculation. inte = 1 initiates molecular dynamics simulations. inte = -1 initiates the molecular statics calculations and simply minimizes the energy without deposition.

A2. Example

```
&printcard
  printf='p'
  rstrtf='r'
  conff='c'
  iconf=1
@end
&headcard  header='Same As Previous Header' &end
&funcard
  atomtype='Cu'
@end
&funcard
  atomtype='Co'
@end
&funcard
  atomtype='Xe'
@end
&funcard
ivapor=0
threadlm=1.0
type=1,2,3,0,0
fraction=1.0,0.0,0.0,0.0,0.0,0.0
fraction2=0.65,0.20,0.15,0.0,0.0
evapor=1.0
devapor=0.0
depdir=0.0,-1.0,0.0
dep=5000.0
Tbndv=4850.0
timelayer=400.0,400.0
ratio=2.0
einert=2.0
@emailcard
  initcard initf='test/cb',genvel=.true.,genlat=.false. &end
&latcard
  latypetype='fcc'
  avec=1.0,1.0,0.0
  bvec=0.0,1.0,1.0
  cvec=1.0,0.0,1.0
  alat=3.52
  xrot=0.0,0.0,1.0
  yrot=1.0,-1.0,0.0
  zrot=1.0,1.0,0.0
  perub=17.1,16.923111,16.923111
  perlb=-0.5,-0.5,-0.5
  xbound=0.5,17.1
  ybound=-0.5,16.923111
  zbound=0.5,16.923111
  ncell=1
  rcell=0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
```
Appendix A. Molecular Dynamics Input File

A.3 Restart File

An example restart (or r0) file is included. This file has 4032 atoms (second line first column). Note that the surface notation used here is (001) because the y-axis is the growth direction. There are seven types of atoms (second column in second line) and previous simulation time has transpired (third column in second line). The upper and lower boundaries are defined for x, y, and z in the third and fourth lines. The atom types and masses (in units used by MD code) are defined next. Every set of three lines thereafter contains the x, y, and z positions and velocities of the atoms, while the last line contains the type of the atom. The entire file ends with a final line of three zeros.

Cu-Co Multi-layers

<table>
<thead>
<tr>
<th>Atom Type</th>
<th>X Position</th>
<th>Y Position</th>
<th>Z Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cu</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>Co</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>
### Appendix A. Molecular Dynamics Input File

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>69</td>
<td>70</td>
<td>71</td>
<td>72</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>77</td>
<td>78</td>
<td>79</td>
<td>80</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>81</td>
<td>82</td>
<td>83</td>
<td>84</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>89</td>
<td>90</td>
<td>91</td>
<td>92</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>93</td>
<td>94</td>
<td>95</td>
<td>96</td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>97</td>
<td>98</td>
<td>99</td>
<td>100</td>
</tr>
</tbody>
</table>

### Notes

- Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
APPENDIX B. MOLECULAR DYNAMICS CODE

B.1 FORTRAN Subroutine and Function Map

The calls between the functions and subroutines that make up nanosim MD code have been mapped in Fig. B.1. MD code starts from initializing the variables and reads in much of the input data from a.i and setups of simulation. Input potential parameters and tables is also included in this step. The initial atom positions are read in from r0 or can be created from parameters in a.i (Appendix A). It is important that the timing, periodicity, scaling, constraint, and temperature control are monitored and maintained during either static energy calculation or dynamics simulation. Therefore, next step is the subroutines and functions that govern molecular statics and molecular dynamics calculations by minimizing energies of system and integrating the equations of atom motions. Potential parameters or tables are needed to obtain the energies and forces of the atoms in the computational cell in this step.
Figure B.1 Map of functions for all the subroutines and functions with nanosim.f MD code.

B.2 Include File

The vap.inc file is included in the nanosim.f MD code.

\begin{verbatim}
parameter (natmax=20000,nelmax=300,ngrid=5000,ngridr=5000,
1    nelmax=8)
\end{verbatim}
Appendix B. Molecular Dynamics Code

parameter (nboxx=20, nboxy=120, nboxz=10, nboxx=100)
logical sortp,follow
common /bigwork/ dneighs(3,natmax,nmeax/2),
   nneighs(natmax), nlocal(natmax), jneighs(natmax*nmeax/2)
common /lattice/ perub(3), perlb(3), perien(3), alat",
   xbound(2), ybound(2), zbound(2), natoms
common /onlyoi, twoi, threei, fouri
common /images/ onlyoi, twoi, threei, fouri
common /particle/ rv(6,natmax), itype(natmax)
common /scaled/ y(6,natmax)
common /forces/ f(3,natmax), e(natmax), stress(3,3),
   slocal(3,3,natmax)
common /accels/ acc(3,natmax)
common /density/ cho(natmax), fp(natmax)
common /types/ amass(nelmax), lelement(nelmax), ntype(nelmax),
   ntypes
common /integ/ t0, tend, dt, tol, dmax, nsteps, noutp,
   inte, iaccur, ndegfr, mterm
common /errs/ detops
common /calls/ nqtlst, nneimx, nforce
common /interact/ rcutsq
common /grids/ frhoar(ngrida, nelmax), frhoar1(ngrida, nelmax),
   frhoar2(ngrida, nelmax), frhoar3(ngrida, nelmax),
   frhoar4(ngrida, nelmax), frhoar5(ngrida, nelmax),
   rhoar1(ngrida, nelmax), rhoar2(ngrida, nelmax),
   rhoar3(ngrida, nelmax), rhoar4(ngrida, nelmax),
   rhoar5(ngrida, nelmax), rhoar6(ngrida, nelmax),
   z2rar1(ngrida, nelmax), z2rar2(ngrida, nelmax),
   z2rar3(ngrida, nelmax), z2rar4(ngrida, nelmax),
   z2rar5(ngrida, nelmax), z2rar6(ngrida, nelmax),
   rh6oar, drar, nzhoar, nrar
common /grids2/ rdar, rzhroar
common /print/ typesp, xminp, xmaxp, ymip, ymaxp, zmip, zmaxp, sortp,
   int, ipater, ipatom, ipiter, irstart, iconf, iconst
common /cbound/ dpres, bndmax(3), bndneg(3), bndvel(3), dstress(3),
   ibdtype, idynper(3)
common /cneigh/ rctsqr, dradn, nneigh, nneips, nyt, dnyt,
   nnxlnx(0,natmax), nwlst, nwlst(3,natmax, 3*nmax), nboxx1
common /nnrn1/ rctsoot, nboonei
common /average/ avtemp, avpres, avvol, avpe, avke, avstrs(3,3),
   eqtm, avtempq0(0:6), neqill, nskipd, nave
common /fixit/ timefx, vectfx(3,0:100), dctfxf(3,0:100),
   workfx(natmax), postfx(3,natmax), modefx(0:100),
   ipstfx(natmax), nfixes, ifxal1, ifxal2
common /tmpfxt/ qreg(6), tmregq(0:6), xcenter(6), xwidth(6),
   ycenter(6), ywidth(6), zcenter(6), zwidth(6), acttemp(0:6),
   drag(0:6), timetemp, ifxtmp, nregs, ipntrf(natmax), ngrf(0:6),
   follow
common /vapor/ ivapor, threedim, typel(5), fraction1(5), type2(5),
   fraction2(5), evapvol, depdir(3), vdep, ycenter1, ywidth1, TbdnV
   .ratio, ylesst(natmax), inert, dtadd, density, ntopbox, marki, mark2,
   nmo, emo, emo0, einert, devapor, inertvar, depdtime, spinphi,
   layernum, emoti, numit, timelayer(2), timelayer(2)

B.3 FORTRAN Functions

MD code (nanosim.f) was modified from the dynamo code written by S. M. Foles
and M. S. Daw by X. W. Zhou for use in vapor deposition. The code was additionally
modified by J. J. Quan especially for the low energy ion assistance vapor deposition by

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Appendix B. Molecular Dynamics Code

adding impact of inert ions on film surface and potentials between inert ions and metal atoms.

nanosim.f

```fortran
program nanosim
  include 'vap.inc'

  character*80 printf, rstrtf, confff
  character*24 timestr
  data rstrtf, confff/2*"none"/, printf/"dynprint"/

  namelist /printcard/ ipinter, ipatoms, ipitera, printf, rstrtf, confff,
  1   iconst, xminp, yminp, zminp, xmaxp, ymaxp, zmaxp, typem, sortp, iconf
  ipinter=0
  ipatoms=0
  ipitera=-1
  rstrtf=0
  iconf=0
  iconst=0
  xminp=-1.e6
  yminp=-1.e6
  zminp=-1.e6
  xmaxp=1.e6
  ymaxp=1.e6
  zmaxp=1.e6
  typem=0
  nboxnei=0
  sortp=false.
  inert=100
  open(unit=5, file='a.i')

  calculate the initial timing

  call initsec()

  read(5, prntcard)

  open output file

  open(unit=6, file=printf, form='FORMATTED', status='UNKNOWN')

  write(6, 9100)
  9100 format('******** nanosim (unix) 2/26/96 ********')

  write(6, 9005) printf
  9005 format(/, 1x, 'opened print file named ', a80)

  call mytime(timestr)

  write(6, 9101) timestr
  9101 format(/, 1x, 'begin run: ', a24)

  open crystal files

  if (.not. (rstrtf(1:4).eq.'NONE'.or. rstrtf(1:4).eq.'none')) then
    rstrtf = 1
    open(unit=21, file=rstrtf, form='FORMATTED', status='UNKNOWN')
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Appendix B. Molecular Dynamics Code

```
9006 write(6,9006) rstrtf
  format(1x,'opened restart file named ',a80)
end if
if (.not.(conf(1:4).eq.'NONE'.or.conf(1:4).eq.'none')) then
  if (iconf.eq.0) iconf = 1
  open(unit=20,file=conf,form='UNFORMATTED',status='UNKNOWN')
  write(6,9007) conf,iconf
else
  iconf = 0
end if

c write the program
c
1000 continue
calculate final timing
c
t2tot = seconds()
c
write(6,9010) t2tot
format(1x,'/lx,'timing:',1f12.3)
call mytime(timestr)
write(6,9012) timestr
9102 format(1x,'end of run:',a24)
c
stop
end

dynamo

c*****************************************************************************************/
c this is the MD program
c
subroutine dynamo
include 'vap.inc'
character*80 header
data header /
  namelist /headcard/ header
  namelist /bndcard/ ibdtype,dpres,bndmas,bnddrg,dstress,idynper
  namelist /nicard/ dradn
  namelist /avecard/ eqtim
eqo=0
nmo=0
em2=0
nexit=0
ibdtype=1
dpres=0
bndmas(1)=100.0
bndmas(2)=100.0
bndmas(3)=100.0
bnddrg(1)=1.0
bnddrg(2)=1.0
bnddrg(3)=1.0
idynper(1)=1
idynper(2)=1
idynper(3)=1
dstress(1)=0
dstress(2)=0
dstress(3)=0
eqtim=0
dradn=1.0

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Appendix B. Molecular Dynamics Code

```c
// clear the force statistics variables
nforce = 0
nglst = 0
nneimx = 0

// read header information
read(5,headcard)
write(6,9110) header
format('****************************** ',//,
'ix,n80,,
'****************************** ')

// print out the units
write(6,9130)
format('/,, explanation of units,, positions in angstroms,,
 1 time in picoseconds,, velocities in angstroms/picosecond,,
 2 energies in ev,, temperature in degrees kelvin,,
 3 pressure in bars,,
 4 mass in ev-sec^2/angstrom^2,,)

// set up potentials

// call inter

// initialize the lattice

// call initlat

// read in the boundary conditions
read(5,bndcard)

if (ibdtype.eq.2.and.bnddrg(1).lt.0.0) then
  bnddrg(1) = sqrt(7.5*perlen(1)/bnmdas(1))
  bnddrg(2) = sqrt(7.5*perlen(2)/bnmdas(2))
  bnddrg(3) = sqrt(7.5*perlen(3)/bnmdas(3))
end if

if (ibdtype.eq.2) then
  trace = (dstress(1)+dstress(2)+dstress(3))/3.
  dpres = dpres + trace
dstress(1) = dstress(1) - trace
dstress(2) = dstress(2) - trace
dstress(3) = dstress(3) - trace
endif

write(6,9210)
format(/,, "***** boundary conditions")
if(ibdtype.eq.1)write(6,9215)
format(/,, 'constant volume (fixed periodic vectors ) ')
if(ibdtype.eq.2)then
write(6,9220)
format(/,, 'dynamic periodic lengths (fixed directions ) ')
write(6,9225)dpres,dstress,bnmdas,bnddrg
9225 format(' desired pressure:',g12.5,,
1 ' desired stresses:',g12.5,,
2 ' bondary mass:',g12.5,, ' boundary drag:',g12.5,,
write(6,9227) idynper
9227 format(' boundaries allowed to move:',g12.5),
endif

convert pressure and stresses from bar to internal units
dpress = dpres/1.e6
dstress(1) = dstress(1)/1.e6
dstress(2) = dstress(2)/1.e6
dstress(3) = dstress(3)/1.e6
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Appendix B. Molecular Dynamics Code

dstress(1) = stress(1)*perlen(2)*perlen(3)/perlen(1)
dstress(2) = stress(2)*perlen(1)*perlen(3)/perlen(2)
dstress(3) = stress(3)*perlen(1)*perlen(2)/perlen(3)
c
read(5,neicard)
c
nmind(0) = 0
nmind(1) = 0
newst = 1
if (dradn.lt.0.0) dradn = 0.1*sqrt(rcutsg)
rcroot=sqrt(rcutsg) + dradn
rcutsn = rcroot**2
if (ivapor .eq. 1) then
dnyt=rcroot
nyt=1000000
mark1=1
mark2=1
if (depdir(1) .lt. .0) mark1=-1
if (depdir(3) .lt. .0) mark2=-1
endif
write(6,9612) rcutsn,dradn
9612 format(1x,/, 'rcutsn=',g13.5,' dradn=',g13.5)
c c calculate energy of initial lattice
c c call calce(ekinl,eperf,etotl,temper,press,1,t0)
c c read in defects
c write(6,9140)
9140 format(/,/' ****** reading in defects ')
c call readeff(ndef)
c if (ndef.ne.0) newst = 1
write(6,9150)ndef
9150 format(' finished reading in ',i5,' defects ')
c c delete vacancies
c c call delvac
c c make sure that all atom types have been defined
c c call chktyp
c c make sure natoms is less than natmax
c if(natoms.gt.natmax) then
write(6,9160)natoms,natmax
9160 format(' natoms=',i10,' is greater than natmax=',i10)
stop
endif
c c read in the fix constraints
c c call setfix
c c calculate initial state
c if(ndef.ne.0) then
   call calce(ekinl,potl,etotl,temper,press,1,t0)
else
   potl = eperf
endif
volint = perlen(1)*perlen(2)*perlen(3)
write(6,9205)ekinl,potl,etotl,temper,press,volint
9205 format(' ****** initial state: ',i10,' 1' kinetic, potential and total energies ',3e17.7,/,
Appendix B. Molecular Dynamics Code

2 ' temperature:','gl3.6,' pressure:','gl3.6,' volume:','gl3.6)

write(6,9170)natoms
9170 format('(/,1x,i5,' particles ')
   if(ipatoms.eq.1.or.ipatoms.eq.3) call patoms

  c set up temperature control
  c call settemp
  c initialize the average variables
  c
  read(5,avecard)
  avvol = 0.0
  nave = 0
  nskipd = 0
  avpe = 0.0
  avke = 0.0
  avpres = 0.0
  avtemp = 0.0
  do 10 i = 1,3
    do 11 j = 1,3
      avstra(i,j) = 0.0
      do 11 ireg = 0,6
      avtmpreg(ireg) = 0.0
  11 enddo

  c setup movie files
  c
  if (iconf.ne.0) then
    if (ivapor .eq. 1 .and. inte .ne. -1) then
      write(20) header
    else
      write(20) header
      write(20) natoms
      write(20) (perub(i),i=1,3),(perlb(i),i=1,3)
      write(20) ntypes
      write(20) (amass(i),ielement(i),i=1,ntypes)
    endif
  endif

  c call subroutine newton
  c
call newton
  if (ivapor .eq. 1) then
    write(6,8041)
    write(6,8042)ycenter(l),ywidth(l)
  endif

8041 format(1x,'substrate temperature region ycenter and ywidth')
8042 format(2f10.4)

  c for energy minimization, output the possible configuration file
  c
  if (iconf.ne.0.and.inte.eq.-1) then

    c t = 0.0
    temp = 0.0
    if (iconf.eq.1) then
      write(20) t,temp,press,(perlen(i),bndvel(i),i=1,3)
      write(20) ((rv(i,j),i=1,3),itype(j),e(j),j=1,natoms)
    endif
    if (iconf.eq.2) then
      write(20) t,temp,press,(perlen(i),bndvel(i),i=1,3)
      write(20) ((rv(i,j),i=1,6),itype(j),e(j),j=1,natoms)
    endif
    if (iconf.eq.3) then
      write(20) t,temp,press,(perlen(i),bndvel(i),i=1,3)
      write(20) ((rv(i,j),i=1,3),itype(j),e(j),j=1,natoms)
      write(20) ((slocal(i,j,k),i=1,j),j=1,k,3),k=1,natoms)
end if
if (lconf.eq.4) then
  write(20) t, temp, press, (perlen(i), bndvel(j), i=1, 3)
  write(20) ((trv(i,j), i=1, 6), iftype(j), j=1, natoms)
  write(20) (((slocal(i,j), i=1, j), j=1, 3), k=1, natoms)
end if

if (inte.eq.2) goto 5000

if (ipatoms.ge.2) call patoms
  call calc(e, ekin2, pot2, etot2, temper, press, 1, tend)
  volume = perlen(1)*perlen(2)*perlen(3)
  write(6,9240) ekin2, pot2, etot2, temper, press, volume
9240 format(1x,'***** final state: ',/,
  1' kinetic, potential and total energies ',3e17.7,/,
  2' temperature:',g13.6,' pressure:',g13.6,' volume:',g13.6)

print out a warning if the energy minimization is terminated for any reason other than convergence
if (inte.eq.-1.and.nnterm.eq.1) then
  write(6,9232)
9232 format(1x,'warning************************************',/,
    1x,'minimization terminated due to nmax',/,
    1x,'warning************************************')
end if

work = 0.0
if (fxal2.ne.0) then
do 90 i = 1, natoms
  work = work + workfx(i)
if (work.ne.0.0) write(6,9242) work
9242 format(1x,'work done by external forces:',g15.6)
end if

calculate the changes in the energies
dekin=ekin2-ekin1
dpot=pot2-pot1
detot=etot2-etot1
ere=pot2-epot2
write(6,9245) dekin, dpot, detot
9245 format(1x,'changes in energies: kinetic, potential, and total ',
  1x,1x,'3e17.7')
write(6,9250) detot
9250 format(1x,'largest change in total energy reported by output: ',e17.7)
write(6,9260) erel
9260 format(1x,'energy of final lattice with defects',
  1x,'relative to initial lattice without defects =',e17.7)
write(6,9270) ((stress(i,j), i=1, 3), j=1, 3)
9270 format(1x,'stress tensor',/1x,3g15.5/1x,3g15.5/1x,3g15.5)
dvperc = 100.0*(volume-volum)/volum
write(6,9290) volume, dvperc
9290 format(1x,'final volume:',g14.7,
  1x,'percent volume change:',g12.5)

compute and output the average values.
if (nave.gt.0) then
  avtemp = avtemp/nave
  avpres = avpres/nave
  avke = avke/nave
  avpe = avpe/nave
  avvol = avvol/nave
  avden = natoms/avvol
write(6,9300) avtemp, avpres, avvol, avden, avke, avpe
Appendix B. Molecular Dynamics Code

9300 format(1x,'average values',/,
1 5x,'temperature:',g14.7,/,
2 5x,'volume:',g14.7,/,
3 5x,'density:',g14.7,/,
3 5x,'kinetic energy:',g14.7,/  'potential energy:',g14.7)
   do 100 i = 1,3
   do 100 j = 1,3
100  avstrs(i,j) = avstrs(i,j)/nave
   write(6,9301) (avstrs(i,j),j=1,3),i=1,3)
   format(5x,'average stress tensor',/,
3 3(7x,3g15.5,/,)
   if(iftxtmp.ne.0) then
   do 101  ireg = 0,nregs
   101  avtrmprg(ireg) = avtrmprg(ireg)/nave
   write(6,9302) (ireg,avtrmprg(ireg),ireg=0,nregs)
   format(' region:','i5,' average temperature:',g14.7)
   endif
   end if

output the restart file
c
   if (irstrt.eq.1) then
   write(21,9501) header
9501 format(a80)
   write(21,9502) natoms,nypes,tend
9502 format(2140,el15.8)
   write(21,9503) (perub(i),i=1,3), (perlb(i),i=1,3)
9503 format(3e25.16)
   write(21,9504) (amass(i),i=1,natypes)
9504 format(e25.16,i10)
   write(21,9505) ((rv(i,j),i=1,6),j=1,natoms)
9505 format(3e25.16/3e25.16/i10)
   write(21,9503) (bnlvel(i),i=1,3)
   end if

output the statistics on call to force
c
5000 continue
   write(6,9401)
9401 format(/,1x,***** statistics on FORCE and GNEIGH')
   write(6,9402) nforce
9402 format(1x,'total number of calls to FORCE:',i5)
   write(6,9403) ngtlist
9403 format(1x,'total number of neighbor list updates:',i5)
   write(6,9404) nneimx
9404 format(1x,'maximum number of actual neighbors found:',i5)
   if (nmo .gt. 0) then
   emo=emo/float(nmo)
   write(6,*) 'average inert gas energy before impact =',emo
eo=eo/float(nmo)
   write(6,*) 'average inert gas energy after impact =',eo
   write(6,*) 'number of inert gas atoms sampled =',nmo
   endif
   if (nomit .gt. 0) then
   eomit=eminit/float(nomit)
   write(6,*) 'average unsampled inert atom energy =',eomit
   write(6,*) 'number of unsampled inert atoms =',nomit
   endif
   return
end

******************************************************************************
prints out the atom positions etc.
c
   subroutine patoms
   include 'vap.inc'
dimension rmult(3), dum1(natmax), idum1(natmax), idum2(natmax)
c
do 1000 i = 1,natoms
1000  idum2(i) = i
   if (.not.sortp) goto 2000
Appendix B. Molecular Dynamics Code

```
permax = amax1(perlen(1),perlen(2),perlen(3))
if(permax.ge.1000.)then
  xbot = 1.e6
  ybot = 1.e6
  zbot = 1.e6
  do 1005 i = 1,natoms
    xbot = min(xbot,rv(i,1))
    ybot = min(ybot,rv(i,2))
    zbot = min(zbot,rv(i,3))
  continue
  call m0laaf(perlen,1,3,idum1,idum2,ifail)
  rmult(1) = 10**4*idum1(1)-3
  rmult(2) = 10**4*idum2(2)-3
  rmult(3) = 10**4*idum3(3)-3
  do 1010 i=1,natoms
    dum1(i) = rmult(1)*rv(i,1)-xbot
    dum2(i) = rmult(2)*rv(i,2)-ybot
    dum3(i) = rmult(3)*rv(i,3)-zbot
  continue
  else
    dum1(i) = rv(i,1)**2 + rv(i,2)**2 + rv(i,3)**2
  continue
eendif
  call m0laaf(dum1,natoms,idum1,idum2,ifail)
  if(ifail.ne.0)write(6,9020)ifail
9020 format(1x,' trouble in sorting routine ',i5)
2000 continue

9180 format(1x,'atomic data:',
  1   'number, type, positions, energy')
   write(6,9181)
9191 format(13x,'velocities, kinetic energy (if not minimization)',/
  2  13x,'forces, local F/N (in ev)'),/
  3  13x,'el. density, f(rho), f''(rho)'),/
  do 2100 i = 1,natoms
    j = idum2(i)
    if (rv(1,j).lt.xminp.or.rv(1,j).gt.xmaxp) goto 2100
    if (rv(2,j).lt.yminp.or.rv(2,j).gt.ymaxp) goto 2100
    if (rv(3,j).lt.zminp.or.rv(3,j).gt.zmaxp) goto 2100
    if (itype(j).ne.0.and.itype(j).ne.typep) goto 2100
    write(6,9190) j,itype(j),rv(1,j),rv(2,j),rv(3,j),e(j)
2100 format(1x,i15,1x,12,4g15.5)
9190 format(1x,i15,1x,12,4g15.5)
   if (int.e.-1) then
      skin = 0.5*amass(itype(j))*(rv(4,j)**2+rv(5,j)**2+rv(6,j)**2)
      write(6,9191) rv(4,j),rv(5,j),rv(6,j),skin
9191 format(9x,4g15.5)
   endif
   plocal = slocal(1,1)+slocal(2,2)+slocal(3,3))/3.
   plocal = plocal +
   $ amass(itype(j))*(rv(4,j)**2+rv(5,j)**2+rv(6,j)**2)/3.
   write(6,9192) (f(k,j),k=1,3),plocal
9192 format(9x,4g15.5)
   p = zho(j)/drhoar + 1.0
   k = p
   k = max(1,min(k,nrhoar-1))
   p = p - k
   ity = itype(j)
   embed = ((frhoar3(k,ity)*p+frhoar2(k,ity))*p+ 
   $ frhoar1(k,ity))*p+frhoar(k,ity)
   write(6,9193) rho(j),embed,fp(j)
9193 format(9x,3g15.5)
2100 continue

write(6,9200) (perlen(i),i=1,3)
9200 format(1x,'perlen:',3g15.7)
return
```
The inter subroutine is specific to the interatomic potential being used.

c*****************************************************************************
c this routine sets the parameters defining the interactions
c
subroutine inter
   include 'vap.inc'
   character*80 atomtype,atommatch
   namelist /funccard/ atomtype
   data commas'/1.0365e-4/
   common /pass/,re(16),fe(16),rhole(16),alpha(16),
      *beta(16),A(16),B(16),cai(16),rhol(16),
      *Fi1(16),Fi2(16),Fi3(16),
      *Fm0(16),Fm1(16),Fm2(16),Fm3(16),
      *fnn(16),Fn(16),rho(16),rhor(16)
   dimension blat(neimmax),lat(neimmax)

   ntypes=0
10  continue
       atomtype='none'
       read(5,funccard)
       if (atomtype .eq. 'none') goto 1200
       open(unit=10,file='eam',form='FORMATTED',status='OLD')
11       read(10,9501,end=1210)atommatch
       9501 format(a80)
       if (atomtype .eq. atommatch) then
            ntypes=ntypes+1
            if (inert .lt. ntypes) then
               write(6,'(a)')'only one inert gas element can be ',
               'defined, this element must be the last one'
               stop
            endif
            if (ntypes.gt.neimmax) then
               write(6,'(a)')'error: number of types greater than neimmax'
               stop
            endif
            if (atomtype .eq. 'He' .or.
               atomtype .eq. 'Ne' .or.
               atomtype .eq. 'Ar' .or.
               atomtype .eq. 'Kr' .or.
               atomtype .eq. 'Xe' .or.
               atomtype .eq. 'Rn') then
               read(10,*) ielement(ntypes)
               read(10,*) amass(ntypes)
               read(10,*) inert-ntypes
               else
               read(10,*) re(ntypes)
               read(10,*) Fe(ntypes)
               read(10,*) rhoe(ntypes)
               read(10,*) alpha(ntypes)
               read(10,*) beta(ntypes)
               read(10,*) A(ntypes)
               read(10,*) B(ntypes)
               read(10,*) cai(ntypes)
               read(10,*) rhol(ntypes)
               read(10,*) Fi1(ntypes)
               read(10,*) Fi2(ntypes)
               read(10,*) Fi3(ntypes)
               read(10,*) Fm0(ntypes)
               read(10,*) Fm1(ntypes)
               endif
         endif
      else
       1200 continue
   end
Appendix B. Molecular Dynamics Code

```
read(10,*) Pm2(ntypes)
read(10,*) Pm3(ntypes)
read(10,*) fnn(ntypes)
read(10,*) Fn(ntypes)
read(10,*) ielement(ntypes)
read(10,*) amass(ntypes)
blat(ntypes)=sqrt(2.0)*re(ntypes)
rhoin(ntypes)=0.85*rhoe(ntypes)
rhout(ntypes)=1.15*rhoe(ntypes)
endif
else
    if (atommatch .eq. 'He' .or.
      * atommatch .eq. 'Ne' .or.
      * atommatch .eq. 'Ar' .or.
      * atommatch .eq. 'Kr' .or.
      * atommatch .eq. 'Xe' .or.
      * atommatch .eq. 'Rn' ) then
        read(10,*) vtmp
        read(10,*) vtmp
    else
        do 1 i=1,21
        read(10,*)vtmp
    enddo
    goto 11
endif
goto 10
110
write(6,*)'error: atom type ',atomtype,' not found'
stop
1200 continue
ntype1=ntypes
if (inert .eq. ntypes) ntype1=ntypes-1
nrar=2000
nrhoar=2000
alatmax=blat(1)
rhoemax=rhoe(1)
do 2 i=2,ntype1
    if (alatmax .lt. blat(i)) alatmax=blat(i)
    if (rhoemax .lt. roe(i)) rhoemax=rhoe(i)
2 continue
rcutsq=sqrt(10.0)/2.0*alatmax
rst=0.5
drar=rcutsq/(nrar-1.0)
fmax=-1.0
do 3 il=1,ntype1
do 3 i2=il,il+!
    if (il .eq. i2) then
do 4 il=nrar
        r=(i-1.0)*drar
        if (r .lt. rst) r=rst
        call prof(il,r,fvalue)
        if (fmax .lt. fvalue) fmax=fvalue
        rhorr(il,i)=fvalue
        call pair(il,i2,r,psi)
z2rar(i,i2)=r*psi
4 continue
    else
        do 5 i=1,nrar
            r=(i-1.0)*drar
            if (r .lt. rst) r=rst
            call pair(il,i2,r,psi)
z2rar(il,i2)=r*psi
z2rar(i2,i)=z2rar(i,i2)
5 continue
endif
3 continue
rhom=fmax
drhoar=rhom/(nrhoar-1.0)
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Appendix B. Molecular Dynamics Code

do 6 i=1,ntypes
    do 7 i=1,nrhoar
        rhoar=(i-1.0)*drhoar
        call embed(it,rhof,emb)
        frhoar(i,it)=emb
    7 continue

    continue
    if (i eq .ne. ntypes) then
        do 331 i=1,ntypes
            do 332 i=1,nrar
                r=(i-1.0)*drar
                if (r .lt. rst) r=rst
                call pqrpull(i,ntypes,r,psi)
                z2rar(i,il,ntypes)=psi
            332 continue
        331 continue
    do 341 i=1,nrar
        rhorar(i,ntypes)=-0.0
    341 continue
    do 371 i=1,nrhoar
        frhoar(i,ntypes)=0.0
    371 continue
    endif

amass(i) = commas*amass(i)
rcutsq = rcutsq**2

rdrar = 1.0/drar
rdrhoar = 1.0/drhoar

do 500 i = 1,ntypes
    frhoar1(i,i) = frhoar(2,i)-frhoar(1,i)
    frhoar2(i,i) = 0.5*(frhoar(3,i)-frhoar(1,i))
    frhoar3(i,i) = 0.5*(frhoar(nrhoar,i)-frhoar(nrhoar-2,i))
    frhoar4(i,i) = frhoar(nrhoar,i)-frhoar(nrhoar-1,i)
    do 520 j = 3,nrhoar-2
        frhoar1(j,i) = ((frhoar(j-2,i)-frhoar(j+2,i))+8.*(frhoar(j+1,i)-frhoar(j-1,i)))/12.
    520 continue
    do 525 j = 1,nrhoar-1
        frhoar2(j,i) = 3.*(frhoar(j+1,i)-frhoar(j,i))
        -2.*frhoar1(j,i)-frhoar1(j+1,i)
        frhoar3(j,i) = frhoar1(j,i)+frhoar1(j+1,i)
        -2.*(frhoar(j+1,i)-frhoar(j,i))
    525 continue
    frhoar2(nrhoar,i) = 0.
    frhoar3(nrhoar,i) = 0.
    do 528 j = 1,nrhoar
        frhoar4(j,i) = frhoar1(j,i)/drhoar
        frhoar5(j,i) = 2.*frhoar2(j,i)/drhoar
        frhoar6(j,i) = 3.*frhoar3(j,i)/drhoar
    528 continue

rhorar1(i,i) = rhorar(2,i)-rhorar(1,i)
    rhorar2(i,i) = 0.5*(rhorar(3,i)-rhorar(1,i))
    rhorar3(i,i) = 0.5*(rhorar(nrar-1,i)-rhorar(nrar-2,i))
    rhorar4(i,i) = 0.
    do 530 j = 3,nrar-2
        rhorar1(j,i) = ((rhorar(j-2,i)-rhorar(j+2,i))+8.*(rhorar(j+1,i)-rhorar(j-1,i)))/12.
    530 continue
    do 535 j = 1,nrar-1
        rhorar2(j,i) = 3.*(rhorar(j+1,i)-rhorar(j,i))
        -2.*rhorar1(j,i)-rhorar1(j+1,i)
        rhorar3(j,i) = rhorar1(j,i)+rhorar1(j+1,i)
        -2.*(rhorar(j+1,i)-rhorar(j,i))
    535 continue
rhorar2(nrar,i) = 0.
Appendix B. Molecular Dynamics Code

```
rhorar3(nrar,i) = 0.
do 538 j = 1,nzar
   rhorar4(j,i) = rhorar1(j,i)/drar
   rhorar5(j,i) = 2.*rhorar2(j,i)/drar
   rhorar6(j,i) = 3.*rhorar3(j,i)/drar
   continue
   i1 = i
   do 540 i2 = 1,nr
      z2rar1(i1,i1,i2) = z2rar2(2,i1,i2) - z2rar(1,i1,i2)
      z2rar1(2,i1,i2) = 0.5*(z2rar3(3,i1,i2) - z2rar(1,i1,i2))
      z2rar1(nrar-1,i1,i2) = 0.5*(z2rar(nrar,i1,i2) - z2rar(nrar-2,i1,i2))
      z2rar1(nrar,i1,i2) = z2rar1(i1,i1,i2) + 8.*(z2rar(j1,i1,i2) - z2rar(j1-1,i1,i2))/12.
   continue
   do 560 j = 1,nrar-1
      z2rar2(j,i1,i2) = 3.*(z2rar(j+1,i1,i2) - z2rar(j,i1,i2)) - 2.*z2rar1(j,i1,i2) - z2rar1(j+1,i1,i2)
      z2rar3(j,i1,i2) = z2rar1(j,i1,i2) + z2rar1(j+1,i1,i2)
      z2rar2(nrar,i1,i2) = 0.
   do 568 j = 1,nzar
      z2rar4(j,i1,i2) = z2rar1(j,i1,i2)/drar
      z2rar5(j,i1,i2) = 2.*z2rar2(j,i1,i2)/drar
      z2rar6(j,i1,i2) = 3.*z2rar3(j,i1,i2)/drar
   continue
   do 540 i2 = 1,nr
      return
   end
```

This subroutine calculates the electron density.

```
subroutine prof(itl,r,f)
   common /pass/ re(16),fe(16),rhor(16),alpha(16),
   *beta(16),A(16),B(16),cai(16),ramda(16),
   *F10(16),F11(16),F12(16),F13(16),
   *FM0(16),FM1(16),FM2(16),FM3(16),
   *fnn(16),Fn(16),rhom(16),rhout(16)
   f=f(f)+exp(-beta(f)))*(r/re(f)-1.0)
   f=f/(1.0*(r/re(f)-rhom(f))**20)
   return
end
```

This subroutine calculates the pair potential.

```
subroutine pair(itl1,itl2,r,psi)
   common /pass/ re(16),fe(16),rhor(16),alpha(16),
   *beta(16),A(16),B(16),cai(16),ramda(16),
   *F10(16),F11(16),F12(16),F13(16),
   *FM0(16),FM1(16),FM2(16),FM3(16),
   *fnn(16),Fn(16),rhom(16),rhout(16)
   if (itl .eq. itl2) then
      psi=psi1*exp(-alpha(ctl1)*r/re(ctl1)-1.0)
      psi=psi1/(1.0+(r/re(ctl1)-cai(ctl1))**20)
   return
end
```
Appendix B. Molecular Dynamics Code

    psi12=B(iti)*exp(-beta(iti)*(r/re(iti)-1.0))
    psi12=psi12/(1.0+(r/re(iti)-ramda(iti))**20)
    psi=psi12-psi12
    else
    psi1=A(iti)*exp(-alpha(iti)*(r/re(iti)-1.0))
    psi1=psi1/(1.0+(r/re(iti)-cal(i1))**20)
    psi2=B(iti)*exp(-beta(iti)*(r/re(iti)-1.0))
    psi2=psi2/(1.0+(r/re(iti)-ramda(iti))**20)
    psi=psi1-psi2
    psi12=A(it2)*exp(-alpha(it2)*(r/re(it2)-1.0))
    psi12=psi12/(1.0+(r/re(it2)-cal(it2))**20)
    psi2=psi2/(1.0+(r/re(it2)-ramda(it2))**20)
    psi=psi12-psi2
    call prof(iti,r,f1)
    call prof(it2,r,f2)
    psi=0.5*(f2/f1*psia+f1/f2*psib)
    endif
    return
end

This subroutine calculates the pair energy with inert gas atoms.

    subroutine pairuniq(ii,i2,r,psi)
    include 'vap.inc'
    a0=0.5292
    a1=-0.8853*a0/(ielement(ii)**.23+ielement(i2)**.23)
    b1=0.181
    b2=0.5099
    b3=0.2802
    b4=0.02017
    c1=-3.2
    c2=-.9423
    c3=-.4029
    c4=-.2016
    y=r/maf
    phi=b1*exp(c1*y)+b2*exp(c2*y)+b3*exp(c3*y)+b4*exp(c4*y)
    psi=ielement(ii)*ielement(i2)*14.4*phi
    return
end

This subroutine calculates the embedding energy.

    subroutine embed(it,rho,emb)
    common /pass/ re(16),fe(16),rhoe(16),alpha(16),
    *beta(16),A(16),B(16),cal(16),ramda(16),
    *F10(16),F11(16),F12(16),F13(16),
    *Fm0(16),Fm1(16),Fm2(16),Fm3(16),
    *Fnn(16),Fn(16),rhoi(16),rhoout(16)
    if (rho .lt. rhoin(it)) then
      emb=F10(it)+
        *F11(it)*(rho/rhoi(it)-1.0)+
        *F12(it)*(rho/rhoi(it)-1.0)**2+
        *F13(it)*(rho/rhoi(it)-1.0)**3
    else if (rho .lt. rhoout(it)) then
      emb=Fm0(it)+
        *Fm1(it)*(rho/rhoi(it)-1.0)+
        *Fm2(it)*(rho/rhoi(it)-1.0)**2+
    end
Appendix B. Molecular Dynamics Code

*    Fm3(it)*(rho/rhoe(it)-1.0)**3
else
  emb=Fm(it)*(1.0-fnn(it)*log(rho/rhoe(it)))*
  (rho/rhoe(it))**fnn(it)
endif
return
end

C********************************************************************************************
C initlat initializes the positions and velocities of the atoms.

C
subroutine initlat
include 'vap.inc'
logical genlat,genvel,gendis,sort
character*50 initf
dimension scale(3),origin(3)
data genlat,genvel,gendis,sort/4*.false. /
data scale/3*-9999. /
data dismag/0.0/
data disxmin,disymin,diszmin/3*-1.6/,disxmax,disymax,diszmax/3*1.6/
namelist /vaporcard/ ivapor,threedim,type1,fraction1,type2,
fraction2,evapor,depdir,vdep,vbndV,ratio,inert,dttadd,
devapor,elnert,inertvary,subspin,vdepreal,timelayer
namelist /initcard/ initf,genlat,genvel,gendis,dismag,sort
data scale,disxmin,disymin,diszmin,disxmax,disymax,diszmax
ivapor=0
threedim=1.0
type1(1)=1
type2(1)=2
fraction1(1)=1.0
fraction2(1)=1.0
evapor=1.0
depdir(1)=0.
depdir(2)=0.
depdir(3)=0.
vdep=200000.0
vbndV=0.
ratio=0.
inert='none'
dttadd=0.
subspin=-1.
devapor=.0
elnert=.0
inertvary=0
vdepreal=6.0e-8
vdep=.0
depdirtme=.0
timelayer(1)=200.0
timelayer(2)=1.0
C
read(5,vaporcard)
timelayer(1)=.5*timelayer(1)
if (timelayer(2) .eq. -1.0) timelayer(2)=timelayer(1)
timelayer(2)=.5*timelayer(2)
if (ivapor .eq. 1) then
  do 88 i=2,5
    fraction1(i)=min(1.0,fraction1(i-1)+fraction1(i))
    fraction2(i)=min(1.0,fraction2(i-1)+fraction2(i))
  continue
  if (subspin .gt. .0) depdirtme=6.0*vdep/vdepreal*subspin*1.0e-12
  if (vbndV .gt. .0) vbndV=vbndV/100000.0
Appendix B. Molecular Dynamics Code

```fortran
if (TbndV.eq. -1.0) TbndV=vdep/100000.0
endif
read(5,initcard)

if (initf(1:4).eq.'NONE'.or.initf(1:4).eq.'none') then
    sort = .true.
genvel = .true.
genlat = .true.
else
    open(unit=22,file=initf,form='FORMATTED',
         status='OLD')
    write(6,9135)initf
79135 format(//'ix,........ using restart',/,
         ' reading in initial file named ',a80)
call restart
close(22)
endif

c generate a lattice if requested
if (genlat) then
call latgen
else
call latdm(density,ivapor,threedim)
endif

c check that the natoms is less than natmax
if (natoms.gt.natmax) then
    write(6,9160) natoms,natmax
9160 format(//'natoms=',i10,' is greater than natmax=',i10)
    stop
endif

c sort lattice if requested
if (sort) call sorter

c scale the lattice if requested
if (scale(1).eq.-9999.) scale(1) = 1.
if (scale(2).eq.-9999.) scale(2) = scale(1)
if (scale(3).eq.-9999.) scale(3) = scale(1)
if (scale(1).ne.1..or.scale(2).ne.1..or.scale(3).ne.1.) then
    write(6,9010) (scale(j),j=1,3)
9010 format(//,i2,'**********',/,' scaling lattice by ',3g13.5,
        ',',**********)
do 100 i = 1,3
    perlen(i) = scale(i)*(perub(i)-perlb(i))
    origin(i) = 0.5*(perlb(i)+perub(i))
    perub(i) = origin(i) + 0.5*perlen(i)
    perlb(i) = perub(i) - perlen(i)
do 100 j = 1,natoms
    rv(i,j) = origin(i) + scale(i)*(rv(i,j)-origin(i))
100 continue
110 continue
endif

c check periodicity
ccall chkper
c generate random displacements if requested
cif (gendis) call disgen(dismag,disxmin,disxmax,disymin,
        *disymax,diszmin,diszmax)
```
Appendix B. Molecular Dynamics Code

c generate the velocities if requested
  if (genvel) then
    call velgen
  else
    call veldum
  end if

c check that all atom types have been defined
  call chktyp

  if (ivapor .eq. 1) then
    if (density .le. .0) then
      write(6,*) 'density not defined'
      stop
    endif
  endif

  if (dtadd .equ .0) then
    if (threedim .ne. 1) then
      dtadd=(perub(1)-perlb(1))*vdep*density*1.0e-5
      dtadd=1.0/dtadd
    else
      dtadd=(perub(1)-perlb(1))*vdep*
      *density*1.0e-5
      dtadd=1.0/dtadd
    endif
    write(6,*)'dtadd=',dtadd
  endif
  return
end

********************************************************

c
This subroutine reads in the parameters required to restart.

c a run from fortran file 22
  subroutine restart
  include 'vap.inc'
  character*80 header

c
  read(22,9501,end=8000) header
9501  format(a80)  
    write(6,9001) header
9001  format(/, ' previous header: ',/3x,a80,/)  
    read(22,9502) natoms,ntyp,t0
9502  format(2i10,e15.8)

c
  check that natoms is less than natmax
  if (natoms.gt.natmax) then
    write(6,9901) natoms,natmax
9901  format(1x,'natoms=',i6,' is greater than natmax=',i6)
    stop
  endif

  write(6,9002) natoms,ntyp
9002  format(1x,i10, ' atoms',i10, ' particle types')
  write(6,9003) t0
9003  format(1x,'final time of previous run: ',f10.3)
    read(22,9503) (perub(i),i=1,3),(perlb(i),i=1,3)
9503  format(3e25.16)
    read(22,9504) (amass(i),i=element(i),i=1,ntyp)
9504  format(e25.16,i10)
Appendix B. Molecular Dynamics Code

9505 read(22,9505) (zw(i,j),i=1,6),itype(j),j=1,natoms)
9505 format(3(e25.16/3e25.16/110)
9505 read(22,9503,end=200) (bdvel(i),i=1,3)
9505 goto 1000
200 continue
210 do 210 i = 1,3
210 bdvel(i) = 0.0
210 write(6,9201)
9201 format(1x,'no bdvel available, zero assumed')
c c print out atom types
c c 1000 continue
1000 write(6,9100)
9100 format(' restart uses these types: ')
9102 format(' type element amass ',/',
9102 1                   -------- --------
9102 write(6,9103)(i,ielement(i),amass(i),i=1,natypes)
9103 format(1x,14,19,2x,gl1.4)
c
do 100 i = 1,3
100 perlenn(i) = perub(i) - perlbi
9031 write(6,9031)
9031 format(' periodicity ',15x,'x',14x,'y',14x,'z')
9032 write(6,9032)perlenn
9032 format(' lower periodic bound ',3g15.5)
9033 write(6,9033)perlenn
9033 format(' upper periodic bound ',3g15.5)
9034 format(' length ',8x,3g15.5)
8000 return
end

***********************************************************************
latgen

subroutine latgen
include 'vap.inc'
character*8 lattice
dimension aivec(3),bvec(3),cvec(3),xrot(3),yrot(3),zrot(3)
dimension xold(3),yold(3),zold(3),avec(3),bvecp(3),cvecp(3)
dimension roter(3,3),aperub(3),aperlb(3)
dimension axbnd(2),aybnd(2),azbnd(2)
dimension rcell(3,10),rcellp(nelm,10),rcellip(3,10)
common /loran/ iseed
data iseed/87654321/
c
data lattice/'none/',xrot/3*0./,yrot/3*0./,zrot/3*0./
data xold/1.,0.,0./,yold/0.,1.,0./,zold/0.,0.,1./
data avec/3*0.0./,bvec/3*0.0./,cvec/3*0.0./
data aperlb/3*9999.9999.,aperub/3*9999.9999.
data axbnd/9999.,9999.,aybnd/9999.,9999.,azbnd/9999.,9999.,9999.
data ncell/1/
data rcell/30*0./
namelist /latcard/ lattice,avec,bvec,cvec,alat,xrot,yrot,zrot,
iperub,perlbi,xbound,ybound,zbound,aperub,aperlb,axbnd,aybnd,
2azbnd,ncell,rcell,rcellp,iseed
alat=3.52
perlbi(1)=9999.0
perlbi(2)=9999.0
perlbi(3)=9999.0
perlbi(1)=9999.0
perlbi(2)=9999.0
perlbi(3)=9999.0

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Appendix B. Molecular Dynamics Code

xbound(1)=-9999.0
xbound(2)=9999.0
ybound(1)=-9999.0
ybound(2)=9999.0
zbound(1)=-9999.0
zbound(2)=9999.0
c
do 1 i = 1,nelmmax
do 1 j = 1,10
   ccell(i,j) = 0.
   continue
   ccell(i,1) = 1.
c
t0 = 0.0
c
determine the lattice parameters
c
   read(5,latcard)
   if (iseed .le. 0) iseed=iseed0
c
do 3 i = 1,3
   if (perlb(i).le.-9998..and.aperlb(i).le.-9998.) aperlb(i) = -2.0
   if (perub(i).ge.9998..and.aperub(i).ge.9998.) aperub(i) = 2.0
   if (perlb(i).le.-9998.) perlb(i) = alat*aperlb(i)
   if (perub(i).ge.9998.) perub(i) = alat*aperub(i)
3
c
set the default x,y,and z bounds
c
   if (axbnd(1).gt.-9998.) xbound(1) = alat*axbnd(1)
   if (aybnd(1).gt.-9998.) ybound(1) = alat*aybnd(1)
   if (azbnd(1).gt.-9998.) zbound(1) = alat*azbnd(1)
   if (axbnd(2).lt.9998.) xbound(2) = alat*axbnd(2)
   if (aybnd(2).lt.9998.) ybound(2) = alat*aybnd(2)
   if (azbnd(2).lt.9998.) zbound(2) = alat*azbnd(2)
   if (xbound(1).le.-9998.) xbound(1) = perlb(1) + 0.01
   if (xbound(2).ge.9998.) xbound(2) = perub(1) + 0.01
   if (ybound(1).le.-9998.) ybound(1) = perlb(2) + 0.01
   if (ybound(2).ge.9998.) ybound(2) = perub(2) + 0.01
   if (zbound(1).le.-9998.) zbound(1) = perlb(3) + 0.01
   if (zbound(2).ge.9998.) zbound(2) = perub(3) + 0.01
c
primdef = (avec(1)**2 +avec(2)**2 +avec(3)**2)**
1/(bvec(1)**2 +bvec(2)**2 +bvec(3)**2)*
2/(cvec(1)**2 +cvec(2)**2 +cvec(3)**2)
if (primdef.ne.0.0) then
   if (ivapor .eq. 1 .and. lattype .eq. 'none') then
      write(6,*h 'unknown lattype'
      stop
   else
      if (ivapor .ne. 1) lattype = 'none'
      endif
   endif
   if (lattype.eq.'none') lattype = 'fcc'
   call storelat(lattype,avec,bvec,cvec)
endif
if (ivapor .eq. 1) then
density=100.0
if (lattype .eq. 'fcc') density=4.0/alat**3
if (lattype .eq. 'bcc') density=2.0/alat**3
if (lattype .eq. 'sc') then
   write(6,*h 'Vapor Deposition not available in sc Materials'
   stop
endif
if (three.dim .ne. 1.0) density=density*alat/
sqrt(zrot(1)**2+zrot(2)**2+zrot(3)**2)
endif
if (lattype.ne.'none') then
Appendix B. Molecular Dynamics Code

```
write(6,9115) lattype
end if
9115  format(//' *     generating ',a8,' lattice ')
write(6,9116)avec,bvec,cvec
9116  format(//' primitive lattice vectors ',3f5.2,3x,3f5.2,3x,3f5.2)
xnorm = snrm2(3,xrot,1)
ynorm = snrm2(3,yrot,1)
znorm = snrm2(3,zrot,1)
c if((xnorm.ne.0.0).and.(ynorm.ne.0.0).and.(znorm.eq.0.0))
  1 call cross(xrot,yrot,zrot)
if((ynorm.ne.0.0).and.(znorm.ne.0.0).and.(xnorm.eq.0.0))
  1 call cross(yrot,zrot,xrot)
if((znorm.ne.0.0).and.(xnorm.ne.0.0).and.(ynorm.eq.0.0))
  1 call cross(zrot,xrot,yrot)
c if none or only one defined, set equal to no rotation
  1 land.(ynorm*xnorm.eq.0.0).and.(xnorm*znorm.eq.0.0).
    then
      call ccopy(3,xold,1,xrot,1)
      call ccopy(3,yold,1,yrot,1)
      call ccopy(3,zold,1,zrot,1)
    endif
c normalize the unit vectors and define rotation matrix
c xnorm = snrm2(3,xrot,1)
ynorm = snrm2(3,yrot,1)
znorm = snrm2(3,zrot,1)
do 5 i=1,3
  xrot(i) = xrot(i)/xnorm
  yrot(i) = yrot(i)/ynorm
  zrot(i) = zrot(i)/znorm
5 continue
  do 10 i=1,3
    do 10 j=1,3
      roter(i,j)= xold(i)*xrot(j) + yold(i)*yrot(j) + zold(i)*zrot(j)
    10 continue
  c rotate avec,bvec,cvec
  c do 20 i=1,3
    avecp(i)=0.0
    bvecp(i)=0.0
    cvecp(i)=0.0
    do 20 j=1,3
      avecp(i) = avecp(i) + roter(i,j)*avec(j)
      bvecp(i) = bvecp(i) + roter(i,j)*bvec(j)
      cvecp(i) = cvecp(i) + roter(i,j)*cvec(j)
    20 continue
    do 800 j=1,ncell
      do 800 i=1,3
        rcellp(i,j)=0.0
        do 800 j=1,3
          rcellp(i,j)=rcellp(i,j)+roter(i,j)*rcell(j,j)
        800 continue
      do 801 j=1,3
        rcellp(i,j)=rcellp(i,j)*0.5*alat
      801 continue
      do 30 i=1,3
        avecp(i) = avecp(i)
        bvecp(i) = bvecp(i)
        cvecp(i) = cvecp(i)
      30 continue
write(6,9216)avec,bvec,cvec
9216  format(//' rotated lattice vectors ',3f5.2,3x,3f5.2,3x,3f5.2)
c c compute perlen
```
Appendix B. Molecular Dynamics Code

c do 50 i = 1,3
   perlen(i) = perub(i) - perl(i)
c
   ahlc = 0.5*alat
   write(6,9030) ahlc
9030 format(' half-lattice constant = ',g15.5)
   write(6,9031)
9031 format(' x, y, z')
   write(6,9032)xbound(1),ybound(1),zbound(1)
9032 format(' lower cell bound ') implements 3g15.5)
   write(6,9033)xbound(2),ybound(2),zbound(2)
9033 format(' upper cell bound ') implements 3g15.5)
   write(6,9034)perlub
9034 format(' lower periodic bound ') implements 3g15.5)
   write(6,9035)perlub
9035 format(' upper periodic bound ') implements 3g15.5)
   write(6,9036)perlub
9036 format(' length ',x8,3g15.5)
   xlen = max(abs(avec(1)),abs(bvec(1)),abs(cvec(1)))*ahlc
   ylen = max(abs(avec(2)),abs(bvec(2)),abs(cvec(2)))*ahlc
   zlen = max(abs(avec(3)),abs(bvec(3)),abs(cvec(3)))*ahlc
   nsidex = int((xbound(2)+xbound(1))/xlen) + 2
   nsidex = int((xbound(2)+xbound(1))/ylen) + 2
   nsidex = max(nsidex,nsidex)
   nsidex = int((zbound(2)+zbound(1))/zlen) + 2
   nsidex = max(nsidex,nsidex)
   nsidex = 2*nsidex
   ncenx = int(0.6 + 0.5*float(nsidex))
   do 190 l = 1,ncell
     do 190 m = 2,ntypes
       ccell(m,l) = min(l,ccell(m-1,l) + ccell(m,l))
190 continue

   natoms = 0
   do 200 i=1,nsidex
     i = i - ncenx
     do 200 k=1,nsidex
       k = k - ncenx
     do 200 j=1,nsidex
       j = j - ncenx
       xc = ahlc*(ii*avec(1) + jj*bvec(1) + kk*cvec(1))
       yc = ahlc*(ii*avec(2) + jj*bvec(2) + kk*cvec(2))
       zc = ahlc*(ii*avec(3) + jj*bvec(3) + kk*cvec(3))
     do 200 l = 1,ncell
       xt = xc + rcell(1,l)
       yt = yc + rcell(2,l)
       zt = zc + rcell(3,l)
       if(xt.ge.xbound(2)) go to 200
       if(yt.ge.ybound(2)) go to 200
       if(zt.ge.zbound(2)) go to 200
       tst = ranl()
     if(tst.ge.xbound(1)) go to 200
     if(tst.ge.ybound(1)) go to 200
     if(tst.ge.zbound(1)) go to 200
     if(tst.ge.xbound(2)) go to 200
     if(tst.ge.ybound(2)) go to 200
     if(tst.ge.zbound(2)) go to 200
     tsp = ranl()
     it = 0
     do 201 m=ntypes,1,-1
     if(tsp.it.ccell(m,l)) it = m
       if (it.le.0.or.it.gt.ntypes) goto 200
       natoms = natoms + 1
       if(natoms.gt.natmax) go to 300
       rtype(natoms) = it
       rv(1,natoms) = xt
       rv(2,natoms) = yt
       rv(3,natoms) = zt
     200 continue
     return

   c
   300 write(6,9601)natmax

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Appendix B. Molecular Dynamics Code

9601 format(' number of atoms generated exceeds maximum dimension:'
1,i10)
return
end

C*******************************************************************************

Dummy program to read and discard the latcard.

C subroutine latdum(density, ivapor, threedim)
character*80 latttype
dimension avec(3), bvec(3), cvec(3), xrot(3), yrot(3), zrot(3)
dimension perub(3), perlb(3), xbound(2), ybound(2), zbound(2)
dimension sperub(3), sperlb(3), axbnd(2), aybnd(2), azbnd(2)
dimension rcell(3,10), ccell(8,10)
common /iran/, iseed
namei latcard/ latttype, avec, bvec, cvec, alat, xrot, yrot, zrot,
iperub, perlb, xbound, ybound, zbound, sperub, sperlb, axbnd, aybnd,
2azbnd, ncell, rcell, ccell, iseed
data iseed=87654321/
alat=3.52
zrot(1)=1.0
zrot(2)=1.0
zrot(3)=1.0
C
read(5, latcard)
if (iseed .le. 0) iseed=iseed0
if (ivapor .eq. 1) then
if (latttype .eq. 'none') then
write(6,*') 'unknown latttype'
stop
else
density=-100.0
if (latttype .eq. 'fcc') density=4.0/alat**3
if (latttype .eq. 'bcc') density=2.0/alat**3
if (latttype .eq. 'sc') then
write(6,*') 'Vapor Deposition not available in sc Materials'
stop
endif
endif
else
return
end
end
end
C*******************************************************************************
C subroutine cross(x, y)
dimension z(3), x(3), y(3)
z(1) = x(2)*y(3) - x(3)*y(2)
z(2) = x(3)*y(1) - x(1)*y(3)
z(3) = x(1)*y(2) - x(2)*y(1)
return
end

C*******************************************************************************

C subroutine storelat(latty, avec, bvec, cvec)
character*8 lstored(10), latti
dimension avecs(3,10), bvecs(3,10), cvecs(3,10)
dimension avec(3), bvec(3), cvec(3)
Appendix B. Molecular Dynamics Code

data listd/fcc ' , 'bcc ' , ' sc ' , ' 7 ' /
data avecs/1,1,0 , 1,1,-1 , 2,0,0 , 21*0.0 /
data bvecs/0,1,1 , -1,1,1 , 0,2,0 , 21*0.0 /
data cvecs/1,0,1 , 1,-1,1 , 0,0,2 , 21*0.0 /
c
primitive lattice vectors given in terms of half-lattice constants

i match = 0
do 10 i=1,10
   if(latty.ne.lstored(i)) go to 10
   i match = 1
     do 5 j=1,3
     a vec(j) = a vecs(j,i)
     b vec(j) = b vecs(j,i)
     c vec(j) = c vecs(j,i)
   5 continue
10 continue
   if(i match.eq.1) return
   write(6,15) latty
   format(' could not find lattice type ',a8,
   l . assume default type.')
c
assume default
i = 1
latty = lstored(i)
do 20 j=1,3
   a vec(j) = a vecs(j,i)
   b vec(j) = b vecs(j,i)
   c vec(j) = c vecs(j,i)
20 continue
return
end

c
The subroutine redef reads in the defcards.

c
subroutine redef(n def)
include ' vap . inc '
integer type , oldtype , newtype
dimension num(2),pos(3),delpos(3), vel(3)
namel ist / def card/ num , oldtype , pos , delpos , vel , newtype ,
1  xmin , xmax , ymin , ymax , zmin , zmax
c
n def = 0
100 continue
c
newtype = 99
oldtype = 99
type = 99
num(1) = -1
num(2) = -1
do 10 i = 1,3
   pos(i) = 9999.
   delpos(i) = 0.
   vel(i) = 9999.
10 continue
   xmin = -9999.
   xmax = 9999.
   ymin = -9999.
   ymax = 9999
   zmin = -9999.
   zmax = 9999.
c
c read in the next defcard and check to see if it is valid
   c read(5,defcard)
if (newtype.ne.99) type=newtype
if (num(1).ne.-1) goto 1000
if (type.ne.99) goto 1000
if (oldtype.ne.99) goto 1000
if (xmin.ne.-9999..or.xmax.ne.9999.) goto 1000
if (ymin.ne.-9999..or.ymax.ne.9999.) goto 1000
if (zmin.ne.-9999..or.zmax.ne.9999.) goto 1000
if (delspos(1).ne.0.) goto 1000
if (delspos(2).ne.0.) goto 1000
if (delspos(3).ne.0.) goto 1000

not valid so return
return
1000 continue
def = ndef + 1
write(6, defcard)

if (num(1).ne.-1) goto 1500
do 1100 i = 1,natoms
if (rv(1,i).lt.xmin.or.rv(1,i).gt.xmax) goto 1100
if (rv(2,i).lt.ymin.or.rv(2,i).gt.ymax) goto 1100
if (rv(3,i).lt.zmin.or.rv(3,i).gt.zmax) goto 1100
if (oldtype.ne.99.and.itype(i).ne.oldtype) goto 1100
if (type.ne.99) itype(i) = type
do 1110 j = 1,3
if (pos(j).ne.9999.) then
rv(j,i) = pos(j)
else
rv(j,i) = rv(j,i) + delpos(j)
endif
if (vel(j).ne.9999.) rv(3+j,i) = vel(j)
1110 continue
1100 continue
goto 100

1500 continue
if (num(2).eq.-1) num(2) = num(1)
creating a new atom
c
if (num(1).eq.0) then
natoms = natoms + 1
n = natoms
if (type.ne.99) then
itype(n) = type
else
itype(n) = 1
endif
do 1605 j = 1,3
rv(j,n) = pos(j)
if (vel(j).eq.9999.) then
rv(3+j,n) = 0.
else
rv(3+j,n) = vel(j)
endif
1605 continue
else
c
modify existing atoms
c
do 1600 i = num(1),num(2)
if (oldtype.ne.99.and.itype(i).ne.oldtype) goto 1600
if (type.ne.99) itype(i) = type
do 1610 j = 1,3

Appendix B. Molecular Dynamics Code

```c
if (pos(j).ne.9999.) then
  rv(j,i) = pos(j)
else
  rv(j,i) = rv(j,i) + dpos(j)
endif
if (vel(j).ne.9999.) then
  rv(3+j,i) = vel(j)
else
  rv(3+j,i) = 0.
endif
1610 continue
1600 continue
endif
goto 100
end

******************************************************************************

Deletes vacancies.

```
c
  subroutine delvac
    include 'vap.inc'
    j = 0
    do 30 i=1,natoms
      if (itype(i).eq.0) go to 30
      j = j + 1
      itype(j) = itype(i)
      do 20 k=1,6
        rv(k,j) = rv(k,i)
      enddo
      20 continue
      ndel = natoms - j
      natoms = j
    write(6,9000) ndel, natoms
  9000 format(/*' deleted ',110,' vacancies'/
             1,110,' atoms left'*/)
    return
  end

******************************************************************************

Sorts atoms.

```
c
  subroutine sorter
    include 'vap.inc'
    dimension idum1(natmax), idum2(natmax), idum3(natmax),
             idum(natmax)
    dimension rmult(3)
    newlst = 1
    order according to non-periodic directions first
    permmax = amax1(perlen(1), perlen(2), perlen(3))
    if (permmax.ge.1000.) then
      call m01aaf(perlen,1,i3,idum1,idum2,ifail)
      rmult(1) = 10**(4*idum1(1)-3)
      rmult(2) = 10**(4*idum1(2)-3)
      rmult(3) = 10**(4*idum1(3)-3)
      do 40 i=1,natoms
        duml(1) = rmult(1)*(rv(1,i)-zboun(l)) +
                   rmult(2)*(rv(2,i)-ybound(l)) +
                   rmult(3)*(rv(3,i)-zbound(l))
        40 continue
    endif
```
write(6,9010)duml(1),iduml(2),iduml(3)
9010 format(1/,****** sorting according to following hierarchy:',
 1/,x,y,z',
 2/1x,316)
else
  do 45 i=1,natoms
    duml(i) = rv(1,i)**2 + rv(2,i)**2 + rv(3,i)**2
  continue
  write(6,9015)
endif
call moleaf(duml,1,natoms,iduml,idum2,ifail)
if(ifail.ne.0)write(6,9020)ifail
9020 format(1/, trouble in sorting routine 'i5
  do 50 i = 1,natoms
  idum3(iduml(i)) = itype(i)
  do 52 i = 1,natoms
    itype(i) = idum3(i)
  do 60 k = 1,6
    do 62 i = 1,natoms
      duml(iduml(i)) = rv(k,i)
    do 64 i = 1,natoms
      rv(k,i) = duml(i)
  continue
  return
end

chktyp checks that all types used have been defined.

c
subroutine chktyp
  include 'vap.inc'
  do 90 i = 1,nemax
    netype(i) = 0
    do 100 i = 1,natoms
      ityp = itype(i)
      if (ityp.le.0.or.ityp.gt.netypes) then
        write(6,9001) ityp
      endif
        format(1x,'** undefined type found. itype = ',i5,' **')
      stop
      continue
    netype(ityp) = netype(ityp) + 1
    continue
  write(6,9101) (netype(i),i=1,netypes)
  format(1x,'number of atoms of each type: ',10i6)
  return
end

chkper

appendix b. molecular dynamics code
Appendix B. Molecular Dynamics Code

```fortran
check periodicity
rcut = sqrt(rcutsq)
permin = 2.*rcut
istop = 0
do 100 i=1,2
if(perlen(i).ge.permin)go to 100
write(6,9230)i
9230 format(' periodicity is too short in the ',i2,' direction')
write(6,9240)permin,perlen(i)
9240 format(' permin = ',e15.5,' periodicity = ',e15.5)
istop = 1
100 continue
if(istop.eq.1)stop
if(perlen(3).lt.2.*rcut)then
  onlyo1=.false.
  twoi=.true.
endif
if(onlyo1)go to 500
if(perlen(3).lt.rcut)threei=.true.
if(perlen(3).lt.2.*rcut/3.)fouri=.true.
if(perlen(3).lt.0.5*rcut)then
  permin = 0.5*rcut
  write(6,9230)i
  write(6,9240)permin,perlen(i)
  stop
endif
500 continue
nx = 1
ny = 1
if(fouri)nz = 4
if(threei.and..not.fouri)nz = 3
if(twoi.and..not.threei)nz = 2
if(onlyo1)nz = 1

print out the number of images used on the first call
if (ncalls.eq.0)then
  write(6,9035)nx,ny,nz
9035 format(' # of periodic images ',i10,5x,i10,5x,i10)
nx0 = nx
ny0 = ny
nz0 = nz
ncalls = 1
endif

print out notice if periodic images changes
if(nx.ne.nx0.or.ny.ne.ny0.or.nz.ne.nz0)then
  nx0 = nx
  ny0 = ny
  nz0 = nz
  write(6,9040)
9040 format('// *** changing # of periodic images ')
  write(6,9035)nx,ny,nz
endif
return
end
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Appendix B. Molecular Dynamics Code

This routine produces random velocities with a boltzmann distribution appropriate to a given temperature.

c subroutine velgen
  include 'vap.inc'
dimension vcm(3)
data boltz/8.617e-5/
namelist /velcard/ temp

c temp = 0.0
read(5,velcard)
c
tempin = temp * boltz

c vcm(1)=0.0
vcm(2)=0.0
vcm(3)=0.0
tmass = 0.0
do 110 i=1,natoms
  it = intype(i)
  vnorm = sqrt(tempin/amass(it))
do 105 jc=1,3
  rv(jc,i) = vnorm*rgauss()
vcm(jc-3) = vcm(jc-3) + amass(it)*rv(jc,i)
105 continue
  if (threedim .eq. .0) rv(6,i)=0.0
  if (threedim .eq. 1.0) then
    rv(6,i)=vnorm*rgauss()
    vcm(3) = vcm(3) + amass(it)*rv(6,i)
  endif
  tmass = tmass + amass(it)
110 continue
  if (threedim .eq. .0) vcm(3)=0.0
c
c ensure that the center of mass is not drifting
c
do 200 jc=1,3
  vcm(jc)=vcm(jc)/tmass
do 300 jc=4,6
do 300 i=1,natoms
  rv(jc,i)=rv(jc,i)-vcm(jc-3)
c
c calculate the kinetic energy
c
ekin=0.0
do 320 i=1,natoms
  v2 = rv(4,i)**2 + rv(5,i)**2 + rv(6,i)**2
320  ekin = ekin + amass(itype(i))*v2
  ekin = ekin*0.5
c
c calculate the actual temperature
c
tempact = 7736.6*ekin/float(natoms)
  if (threedim .eq. .0) then
    tempact=tempact*1.5
  endif

c scale the velocities to get the exact temperature
c
ekin = 7736.6*ekin/float(natoms)
  if (tempact.gt.0.0) then
    scalev = sqrt(temp/tempact)
  else
    if (temp.le.0.0) then
      scalev = 0.
  endif
Appendix B. Molecular Dynamics Code

```fortran
else
   write(6,*)'error in velgen'
stop
endif
endif
do 350 jc=4,6
do 350 i=1,natoms
   350 rv(jc,i) = rv(jc,i)*scalev
   c set the initial boundary velocities
   c
do 400 i = 1,3
   400 bndvel(i) = 0.0
   return
end

Dummy program to read and discard the velcard.

c subroutine veldum
   namelist /velcard/ temp
   read(5,velcard)
   return
end

This routine produces random displacement.

c subroutine disgen(dismag,disxmin,disxmax,disymin,disymax,
   'disxmin,disxmax)
   include 'vap.inc'
c
do 50 i=1,natoms
   if (rv(1,i) .le. disxmin) goto 50
   if (rv(1,i) .ge. disxmax) goto 50
   if (rv(2,i) .le. disymin) goto 50
   if (rv(2,i) .ge. disymax) goto 50
   if (threedim .eq. 1.0) then
      if (rv(3,i) .le. diszmin) goto 50
      if (rv(3,i) .ge. diszmax) goto 50
      endif
   do 51 ic=1,3
      if (ic .eq. 3 .and. threedim .eq. 0) goto 51
   rv(ic,i) = rv(ic,i) + 2.*dismag*(ranl()-0.5)
   51 continue
50 continue
return
end

Newton

```
Appendix B. Molecular Dynamics Code

subroutine newton
  include 'vap.inc'

  data tottim, outtim/0.1, 0.05/
  data nmax/100000000, stmax/0.01/
  namelist /intcard/ tottim, outtim, iaccur, dt, tol, inte, nmax, stmax
dt=0.001
tol=1.0e-9
iaccur=1
inte=1
read(5, intcard)

if (inte.eq.2) return
if (inte.eq. 1 .and. tottim.eq.0) return
if (inte.eq.-1 .and. nmax.eq.0) return
write(6, 9010)
9010 format(1, '****** solving problem')
nsteps = tottim/dt
if (inte.gt.0) ipitera = iabs(ipitera)
if (inte.eq.-1) then
  nsteps = nmax
dxmax = stmax
endif
if (outtim.gt.tottim .and. inte .eq. 1) then
  write(6, 9020) outtim, tottim
9020 format(1, ' output time=', g15.5, ' is greater than tottim=', g15.5, ' will set outtim=tottim')
  outtim = tottim
endif
noutp = nint(outtim/dt)
nequil = eqtim/outtim

c determine the number of degrees of freedom
c
if (ibdtype.eq.1) then
  ndegfr = natoms
else if (ibdtype.eq.2) then
  ndegfr = natoms + 1
else
  write(6, 9030) ibdtype
9030 format(1, 'undefined boundary type: ibdtype=', i3)
  stop
endif
c
check that ndegfr does not exceed natmax
c
if (ndegfr.gt.natoms) then
  write(6, 9901) ndegfr, natmax
9901 format(1, 'ndegfr=', i6, ' is greater than natmax=', i6)
  stop
endif
c compute the scaled coordinates
c
  call scale
c
  call the requested solver
c
if (inte.lt.0) then
  write(6, 9100)
9100 format(1, ' minimizing energy')
else
  write(6, 9110)
9110 format(1, ' solving newton's equation')
endif
if (inte.eq.-1) call minimize
Appendix B. Molecular Dynamics Code

```fortran
if (int.eq. 1) call nordint
write(6,9120)
9120 format(' ****** solution finished')
c unscale the coordinates
c call coleot
call uscale
return
end
```

The following several subroutines are for Molecular statics calculation. This routine
minimizes the energy using conjugate gradients methods.

```fortran
subroutine minimize
include 'vap.inc'
data df/-.1/
c eps = -tol
do 5 i=1,ndegfr
do 5 j=4,6
5 y(j,i) = 0.0
c if(dxmax.eq.0.0) then
distnn = alat/2.
dxmax = 0.1*distnn
endif
c n = 3*ndegfr
c write(6,9110) nsteps, eps, dxmax
call va08a(n,ener,df,eps,nsteps,ipitera,
1 dxmax,iterm)
c mnterm = iabs(iterm)
25 continue
9110 format(' calling minimizer (va08a)  nfmax = ',i10,
1 '  eps = ',g15.5,'  dxmax = ',g15.5)
return
end
```

Main subroutine for energy minimization.

```fortran
subroutine va08a(n,fl,dfn,epps,mmaxn,iprint,dsmax,
1 iterm)
include 'vap.inc'
dimension x(6*nmax)
equivalence (x(1),y(1,1))
common /mincom/ g(3*natmax), s(3*natmax)
c ntry = 0
nmax = 0
iterm=0
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Appendix B. Molecular Dynamics Code

if (iprint.ne.0) write (6,250) n, fl, dfn, epps, maxfn
eps=abs(epps)
z=fl
itn=0
call vafunc(fl)
fast=fl
smax=0.
fy=1.0e+6
imax=0
igrad=1
ifn=1
df=dfn
if (dfn.eq.0.) df=dfl-z
if (dfn.lt.0.) df=abs(df+fl)
if (dfn.le.0.) df=1.
continue
110
continue
do 120 i=1,n
120
s(i)=0.
smaxx=0.
gg=1.
do 220 icyc=1,n
itn=itn+1
gglast=gg
130
gg=sdot(n,g,1,g,1)
rg=gg/gglast
if (iprint.eq.0) go to 140
if (mod(itn,lab(i)).ne.0) go to 140
write (6,260) icyc, itn, ifn, fl, gg, imax, smax
if (iprint.lt.0) go to 140
write (6,270) (x(i), i=1,n)
write (6,270) (g(i), i=1,n)
continue
if (z.ne.0.) go to 150
if (igrad.eq.1) go to 240
call vafunc(fl)
igraf=1
ifn=ifn+1
go to 130
150
continue
if (abs(s(i)) .gt. abs(s(imax))) imax=i
160
continue
gs=sdot(n,g,1,s,1)
if (gs.ge.0.) go to 110
if (epps.lt.0.) then
ss=sdot(n,s,1,s,1)
alphmx=dmaxx/sqrt(ss)
endif
30
gs0=gs
if ( (fast-fl).gt.0. ) df=fast-fl
fast=fl
alpha=-2.*df/gs
alpha=0.
170
continue
if (ifn.ge.maxfn) go to 240
ntry = ntry + 1
if (alpha.ge.alphmx) nmax = nmax + 1
if (epps.lt.0.) alpha=amin1(alphmx, alpha)
alpha=alpha0+alpha
call saxpy (n,df, alpha, s,3,x,6)
call saxpy (n,df, alpha, s,2,x,6)
call saxpy (n,df, alpha, s,3,x,6)
call vafunc(fy)
ifn=ifn+1
igraf=1
icon=0
if (abs(alpha*s(imax)) .gt. eps) icon=1

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Appendix B. Molecular Dynamics Code

```haskell
reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
```
This routine sets up the gradient of the energy for va08a.

c    subroutine vafunc(ener)
    include 'vap.inc'
    common /mincom/ grad(3*natmax),s(3*natmax)

    c unscale the coordinates
    call colect
    call uscale
    if (ibdtype.ne.1) call chkper
    c compute the forces
    call force(t0)
    c determine the energy and the gradients
    if (ibdtype.eq.1) then
        ener = 0.0
        do 10 i = 1,natoms
        ener = ener + e(i)
        grad(1+3*(i-1)) = -f(1,i)
        grad(2+3*(i-1)) = -f(2,i)
        grad(3+3*(i-1)) = -f(3,i)
        10 continue
    else ener = 0.0
        do 20 i = 1,natoms
        ener = ener + e(i)
        grad(1+3*(i-1)) = -perlen(1)*f(1,i)
        grad(2+3*(i-1)) = -perlen(2)*f(2,i)
        grad(3+3*(i-1)) = -perlen(3)*f(3,i)
        20 continue
        ener = ener + e(natoms+1)
        volume = perlen(1)*perlen(2)*perlen(3)
        grad(3*natoms+1) = 1
        idynper(1)*((dpress-stress(1,1)/volume)*perlen(2)*perlen(3)
        + perlen(1)*dstress(1))
        grad(3*natoms+2) = 1
        idynper(2)*((dpress-stress(2,2)/volume)*perlen(1)*perlen(3)
        + perlen(2)*dstress(2))
        grad(3*natoms+3) = 1
        idynper(3)*((dpress-stress(3,3)/volume)*perlen(1)*perlen(2)
        + perlen(3)*dstress(3))
    endif
    return
end

This routine calls the nordsieck integrator.

c    subroutine nordint
    include 'vap.inc'

c    t=t0
    write(6,9001)dt
7001 format(' calling integrator (nordsieck): dt=',e15.5)
    x1 = seconds()
Appendix B. Molecular Dynamics Code

```c
ctime = dt*noutp
   call nords(t,ndegfr,ctime)

x2 = seconds() - x1
   write(6,9101)x2
9101 format(' time for call to integrator ',e15.5)
   write(6,9102)nforce
9102 format(' number of calls to subroutine force ',i20)
   return

******************************************************************************

Main MD subroutine.

```c
subroutine nords(t,n,otime)
   include 'vap.inc'
   common /uu/ u1(3,natmax),u2(3,natmax),u3(3,natmax),
   *u4(3,natmax),u5(3,natmax)
   common /add/ natadd,ityadd(100)
   dimension phi(3,natmax)
   data c0,c1,c2,c3,c4,c5/0.1875,0.697222222,1.,0.61111111111,1.
   *0.1666666667,0.01666666667/
   dtz = 0.5*dt**2
dtrec = 1.0/dt
dt1 = dt
   nstrt=0
   natadd=0
   tshift=0.
   layernumber=1
   tinit = t
   tfinal = tinit + nsteps*dt - dt/100.
   timest = seconds()
   neinc=0
   if (iomega .eq. 1) then
      do 300 i=1,natoms
      300   yest(i)=9000.0
   c   the program assumes a real positive number ratio
   c
   dtadd1=dtadd
dtadd2=dtfinal
   else
dtadd2=dtadd/ratio
   endif
   c  initialize the higher derivatives to zero
   c
   do 100 j = 1,3
do 100 i = 1,n
   u3(j,i) = 0.0
   u4(j,i) = 0.0
   u5(j,i) = 0.0
100  continue
c  set the velocity and acceleration
   c
   call accel(t)
do 200 j = 1,3
do 200 i = 1,n
   u1(j,i) = dt*y(j+3,i)
u2(j,i) = dt2*acc(j,i)
```
Appendix B. Molecular Dynamics Code

200 continue
C
1000 continue
C
  t = t + dt1
C
C compute the estimated values at the next time step.
C
  do 2000 j = 1,3
  do 2000 i = 1,n
  y(j,i) = y(j,i) + u1(j,i) + u2(j,i) + u3(j,i) +
  u4(j,i) + u5(j,i)
  u1(j,i) = u1(j,i) + 2.0*u2(j,i) + 3.0*u3(j,i) +
  4.0*u4(j,i) + 5.0*u5(j,i)
  y(j+3,i) = dtrec*u1(j,i)
  u2(j,i) = u2(j,i) + 3.0*u3(j,i) + 6.0*u4(j,i) + 10.0*u5(j,i)
  u3(j,i) = u3(j,i) + 4.0*u4(j,i) + 10.0*u5(j,i)
  u4(j,i) = u4(j,i) + 5.0*u5(j,i)
 2000 continue
C
C determine if vapor atoms are to be added
C
  if (ivapor .eq. 1) then
    if (t .gt. eqtim .and. nstrt .eq. 0) then
      nstrt=1
      dtrun1=0
      dtrun2=0
    endif
    if (nstrt .eq. 1) then
      dtrun1=dtrun1+dt1
      dtrun2=dtrun2+dt1
      tshift=tshift+dt1
      if (tshift .ge. timelayer(nlayer) .and. neinc .eq. 0) then
        evapor=evapor+devapor
        neinc=1
      endif
      if (tshift .ge. timelayer(nlayer)) then
        if (nlayer .eq. 1) then
          nlayer=2
        else
          nlayer=1
        endif
      evapor=evapor-devapor
      tshift=.0
      neinc=0
      endif
      if (dtrun1 .ge. dtadd1) then
        dtrun1=dtrun1-dtadd1
        natadd=natadd+1
        layadd(layadd)=1
        if (dtrun1 .ge. dtadd1) goto 1
      endif
      if (dtrun2 .ge. dtadd2) then
        dtrun2=dtrun2-dtadd2
        if (inertvary .eq. 1 .and. neinc .eq. 0) goto 142
        natadd=natadd+1
        layadd(natadd)=inert
      endif
      if (dtrun2 .ge. dtadd2) goto 2
    endif
  endif
142 continue
if (dtrun2 .ge. dtadd2) goto 2
endif
endif
if (natadd .gt. 0 .and. depdttime .gt. 0)
  *spinphi=depdttime*(t-tinit)
endif
C
C compute the actual accelerations at this point
C
  call accel(t)
Appendix B. Molecular Dynamics Code

```fortran
! compute the displacement function phi
do 3000  i = 1,n
   do 3000  j = 1,3
      phi(j,i) = dt2*acc(j,i) - u2(j,i)
   3000 continue
!
! correct the predicted values
do 4000  j = 1,3
   do 4000  i = 1,n
      y(j,i) = y(j,i) + c0*phi(j,i)
      u1(j,i) = u1(j,i) + c1*phi(j,i)
      y(j+3,i) = dt2*uc(j,i)
      u2(j,i) = u2(j,i) + c2*phi(j,i)
      u3(j,i) = u3(j,i) + c3*phi(j,i)
      u4(j,i) = u4(j,i) + c4*phi(j,i)
      u5(j,i) = u5(j,i) + c5*phi(j,i)
   4000 continue
!
! if (ivapor .eq. 1) then
   if (yest(i) .gt. y(2,i)) yest(i)=y(2,i)
!
! call chkvap
endif
!
! output the results
fiout=mod(t-tinit+0.5*dtl,otime)-0.5*dtl
!
! if (fiout .lt. 0.5*dtl .and. fiout .ge. -0.5*dtl) then
   if (t<ftime) goto 1000
!
! call output(t,iaccurr)
   if (t<ftime .and. .5*otime) call output(t,iaccurr)
!
9000 format(1x,'undefined boundary type: ibdtype =',i3)
!
```

**accel**

```fortran
! subroutine accel(time)
 include 'vap.inc'
!
! unscale the coordinates
!
! call colect
! call uscale
!
! if (ibdtype.ne.1) call chaper
!
! compute the forces
!
! call force(time)
!
! goto (1000,2000) ibdtype
!
! write(6,9010) ibdtype
!
9010 format(1x,'undefined boundary type: ibdtype =',i3)
stop
```
Appendix B. Molecular Dynamics Code

c ibdtype = 1
c 1000 continue
do 1101 j = 1,natoms
do 1100 i = 1,3
   acc(i,j) = f(i,j)/amass(iotype(j))
1100 continue
1101 continue
   return
c c ibdtype = 2
c 2000 continue
do 2101 j = 1,natoms
do 2100 i = 1,3
   acc(i,j) = (f(i,j)/amass(iotype(j)))
   1 - 2.*y(i+3,ndegfr)*y(i+3,j))/y(i,ndegfr)
2100 continue
2101 continue
   volume = y(1,ndegfr)*y(2,ndegfr)*y(3,ndegfr)
do 2200 i = 1,3
   acc(i,ndegfr) = idynper(i)*(1 + stresses(i,i)-dpress*volume)/y(i,ndegfr)
   2 - y(i,ndegfr)*dstress(i))/bnmas(i)
   3 - bnddrg(i)*y(3*i,ndegfr))
2200 continue
   return
end

*****************************************************************************************
output

subroutine output(t,iclfor)
include 'vap.inc'
data ifirst/0/,ipcnt/-1/
c c compute the unscaled coordinates
c call colect
c call uscale
c c compute the energy, temperature and pressure
c call calce(ekin,pot,etot,temp,press,iclfor,t)
c c determine the maximum change in total energy
c if(ifirst.eq.0)then
   ifirst=1
eold=etot
detops = 0.0
else
de=abs(etot-eold)
detops = amax1(detops,de)
eold=etot
endif
c c print out energy, temperature and pressure
c ipcnt = mod(ipcnt + 1,ipitera)
if(ipcnt.eq.0)then
   write(6,9001) t,ekin,pot,etot
9001 format(' output: time,kin,pot,tot','f7.3,3e13.5')
   write(6,9002)temp,press
9002 format(9x,'temperature: ',g12.5,5x,'pressure: ',g12.5)
   if (ifxtmp.ne.0) then
Appendix B. Molecular Dynamics Code

write(6,9004)(ireg,natrg,ireg),actttmp(ireg),ireg=0,nregs
format(15,' region:',i5,' number of atoms:',i10,' temperature:',g12.5)
if (lpadm.eq.1) write(6,9981)ycenter(1),ywidth(1)
format(9x,' ycenter:',g12.5,5x,' ywidth:',g12.5)
endif
if (libdtypeeq.2) then
write(6,9003)(perlen(i),i=1,3)
derif
format(9x,'perlen:',3g17.8)
derif

increment the average variables
c
nskip = nskipd + 1
if (nskip.gt.nequl) then
nave = nave + 1
avpres = avpres + press
avtemp = avtemp + temp
avvol = avvol + perlen(1)*perlen(2)*perlen(3)
avke = avke + ekin
avpe = avpe + pot
do 100 i = 1,3
do 100 j = 1,3
100 avstrs(i,j) = avstrs(i,j) + streseq(i,j)
do 101 ireg = 0,nregs
avtempd(ireg) = avtempd(ireg) + actttmp(ireg)
derif

coutput to file 20
c
if (lpadm.eq.1 .and. iconf.ne. 0) then
write(20) natoms,ntypes
write(20) (perub(i),i=1,3), (perlb(i),i=1,3)
write(20) (amass(i),i=1,3), (i0element(i),i=1,ntypes)
derif
if (iconf.eq.1) then
write(20) t, temp, press, (perlen(i),bndvel(i),i=1,3)
write(20) ((rv(i,j),i=1,3),itype(j),e(j),j=1,natoms)
derif
if (iconf.eq.2) then
write(20) t, temp, press, (perlen(i),bndvel(i),i=1,3)
write(20) ((rv(i,j),i=1,6),itype(j),e(j),j=1,natoms)
derif
if (iconf.eq.3) then
write(20) t, temp, press, (perlen(i),bndvel(i),i=1,3)
write(20) ((rv(i,j),i=1,3),itype(j),e(j),j=1,natoms)
write(20) ((slocal(i,j,k),i=1,l),j=1,3,k=1,natoms)
derif
if (iconf.eq.4) then
write(20) t, temp, press, (perlen(i),bndvel(i),i=1,3)
write(20) ((rv(i,j),i=1,6),itype(j),e(j),j=1,natoms)
write(20) ((slocal(i,j,k),i=1,l),j=1,3,k=1,natoms)
derif
return
derif
c******************************************************************************************
calce
c
subroutine calce(ekin,pot,etot,temp,press,icf,for,time)
include 'vap.inc'
data bolts/8.617e-5/
c
ccompute the kinetic energy
c
ekin=0.0
do 20 i=1,natoms
Appendix B. Molecular Dynamics Code

```
v2 = rv(4,i)**2 + rv(5,i)**2 + rv(6,i)**2
20 ekin = ekin + amass(iotype(i))*v2
      ekin = ekin*0.5
      c
      compute the temperature
      c
temp = 7.7366e+3*ekin/float(natoms)
      if (threedim .eq. .0) then
         temp*temp*1.5
         endif
      c
      compute temperatures of the regions
      c
      if (ifxtmp.ne.0) then
         do 40 ireg=0,nregs
            natrg(ireg) = 0
            40 acttmp(ireg) = 0.
            do 50 i=1,natoms
               ireg = i*natrg
               natrg(ireg) = natrg(ireg) + 1
               acttmp(ireg) = acttmp(ireg) +
               amass(iotype(i))*(rv(4,i)**2 + rv(5,i)**2 + rv(6,i)**2)
            50 continue
            do 60 ireg=0,nregs
               if (natrg(ireg).ne.0).acttmp(ireg) = acttmp(ireg)/
               (3.*boltz*float(natrg(ireg)))
               if (threedim .eq. .0) then
                  acttmp(ireg) = acttmp(ireg)*1.5
                  endif
               60 continue
      c
      if (iclfor.eq.1) call force(time)
      c
      compute the potential energy and pressure
      c
      pot=0.0
      do 300 i=1,natoms
      300 pot=pot+e(i)
      volume = perlen(1)*perlen(2)*perlen(3)
      press = (stresst(1,1)+stresst(2,2)+stresst(3,3))/(3.0*volume)
      press = 1.602e6*press
      c
      etot=ekin+pot
      return
      end
```

---

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Subroutine setfix determines the constraints on each atom.

```fortran
subroutine setfix
integer type
include 'vap.inc'
dimension num(2),vector(3),dvecd(3)
namelist /fixcard/ num,mode,vector,dvecd,
1 xmin,xmax,ymin,ymax,zmin,zmax

timefx = t0
nread=1

nfixes = 0
ifixall = 0
ifixal2 = 0

clear the fixcard data

do 100 i = 0,100
    modefix(i) = 0
    vectfix(1,i) = 0.
    vectfix(2,i) = 0.
    vectfix(3,i) = 0.
    dvectfix(1,i) = 0.
    dvectfix(2,i) = 0.
    dvectfix(3,i) = 0.
100 continue
1000 continue

num(1) = 0
num(2) = 0

mode = -9999
xmin = -9999.
xmax = 9999.
ymin = -9999.
ymax = 9999.
zmin = -9999.
zmax = 9999.
vector(1) = 9999.
vector(2) = 9999.

read the fix card
```
Appendix B. Molecular Dynamics Code

```c
read(5,fixcard)
if (nread .eq. 1) then
if (threedim .eq. .0) then
  mode=1
  vector(1)=.0
  vector(2)=.0
  vector(3)=1.0
endif
nread=2
endif

see if anything is to be fixed
if (mode.eq.-9999) goto 2000
nfixes = nfixes + 1
endif
if not, done
if (ifxall.eq.0) then
  ifxall = 1
  do 300 i = 1,natmax
    ipntfx(i) = 0
  enddo
endif
if (ifxal2.eq.0.and.mode.eq.4) then
  ifxal2 = 1
  do 400 i = 1,natmax
    workfx(i) = 0.
    posfx(1,i) = 0.
    posfx(2,i) = 0.
    posfx(3,i) = 0.
    continue
  enddo
write(6,fixcard)
if vector refers to a direction, normalize it.
if (mode.eq.1.or.mode.eq.2) then
  vnorm = sqrt(vector(1)**2 + vector(2)**2 + vector(3)**2)
  vector(1) = vector(1)/vnorm
  vector(2) = vector(2)/vnorm
  vector(3) = vector(3)/vnorm
endif
store the fixcard data
modefx(nfixes) = mode
if(vector(1).ne.9999.) vectfx(1,nfixes) = vector(1)
if(vector(2).ne.9999.) vectfx(2,nfixes) = vector(2)
if(vector(3).ne.9999.) vectfx(3,nfixes) = vector(3)
if(dvecdt(1).ne.9999.) dvecfx(1,nfixes) = dvecdt(1)
if(dvecdt(2).ne.9999.) dvecfx(2,nfixes) = dvecdt(2)
if(dvecdt(3).ne.9999.) dvecfx(3,nfixes) = dvecdt(3)
if (num(1).ne.0) goto 1500
if specified by position
  do 1100 i = 1,natoms
    if (rv(1,i).lt.xmin.or.rv(1,i).gt.xmax) goto 1100
    if (rv(2,i).lt.ymin.or.rv(2,i).gt.ymax) goto 1100
    if (rv(3,i).lt.zmin.or.rv(3,i).gt.zmax) goto 1100
    if (type(i).ne.0.and.itype(i).ne.type) goto 1100
    ipntfx(i) = nfixes
    if (mode.eq.4) then
      posfx(1,i) = rv(1,i)
      posfx(2,i) = rv(2,i)
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Appendix B. Molecular Dynamics Code

```fortran
posfx(3,i) = rv(3,i)
endif
1100 continue
goto 1000
c
c specified by number
c 1500 continue
if (num(2).eq.0) num(2) = num(1)
do 1600 i = num(1),num(2)
iptfx(i) = nfixes
if (mode.eq.4) then
posfx(1,i) = rv(1,i)
posfx(2,i) = rv(2,i)
posfx(3,i) = rv(3,i)
endif
1600 continue
goto 1000
2000 continue
if (nfixes.eq.0) return
c c set the velocities of any fixed atoms to 0 in the appropriate
c direction
c do 2100 i = 1,natoms
iii = iptfx(i)
if (modefxy(iii).eq.0.or.modefxy(iii).eq.4) goto 2100
if (modefxy(iii).eq.1) then
vvec = rv(4,i)*vectfx(1,iii) + rv(5,i)*vectfx(2,iii)
1 + rv(6,i)*vectfx(3,iii)
rv(4,i) = rv(4,i) - vvec*vectfx(1,iii)
rv(5,i) = rv(5,i) - vvec*vectfx(2,iii)
rv(6,i) = rv(6,i) - vvec*vectfx(3,iii)
endif
if (modefxy(iii).eq.2) then
vvec = rv(4,i)*vectfx(1,iii) + rv(5,i)*vectfx(2,iii)
1 + rv(6,i)*vectfx(3,iii)
rv(4,i) = vvec*vectfx(1,iii)
rv(5,i) = vvec*vectfx(2,iii)
rv(6,i) = vvec*vectfx(3,iii)
endif
if (modefxy(iii).eq.3) then
rv(4,i) = 0.0
rv(5,i) = 0.0
rv(6,i) = 0.0
endif
2100 continue
c c if(icnst.eq.1)then
write(6,9010)
9010 format(1x,'****** the following constraints apply')
write(6,9020)
9020 format(1x,' num type mode',15x,'vector',35x,'dvecdt')
do 2200 i = 1,natoms
if (modefxy(iptfx(i)).eq.0) goto 2200
write(6,9030) i,itype(i),modefxy(iptfx(i)),
1 (vectfx(j,iptfx(i)),j=1,3),
2 (dvectfx(j,iptfx(i)),j=1,3)
9030 format(1x,3i6,3g12.4,5x,3g12.4)
2200 continue
write(6,9040)
9040 format(1x,'end of constraints',/,,lx)
endif
return
end
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.


fixfor

subroutine fixfor(time)
    include 'vap.inc'
    dimension delta(3)
    if no constraints, return
    if (nfixes.eq.0) return
    increment the vectors with time
    do 100 i = 1,nfixes
        vectfx(1,i) = vectfx(1,i) + (time-timefx)*dvctfx(1,i)
        vectfx(2,i) = vectfx(2,i) + (time-timefx)*dvctfx(2,i)
        vectfx(3,i) = vectfx(3,i) + (time-timefx)*dvctfx(3,i)
    100 continue
    implement the constraints
    do 1000 i = 1,natoms
        if (ipntfx(i).eq.0) goto 1000
        goto (1100,1200,1300,1400) modefx(ipntfx(i))
    1000 if (modefx(ipntfx(i)).eq.0) goto 1000
    write(6,9101) modefx(i)
    9101 format(1x,'undefined modefx',i6)
    stop
    constrain to a plane
    dotprd = f(1,i)*vectfx(1,ipntfx(i)) + f(2,i)*vectfx(2,ipntfx(i))
    1 l = f(3,i)*vectfx(3,ipntfx(i))
    f(1,i) = f(1,i) - dotprd*vectfx(1,ipntfx(i))
    f(2,i) = f(2,i) - dotprd*vectfx(2,ipntfx(i))
    f(3,i) = f(3,i) - dotprd*vectfx(3,ipntfx(i))
    goto 1000
    constrain to a line
    dotprd = f(1,i)*vectfx(1,ipntfx(i)) + f(2,i)*vectfx(2,ipntfx(i))
    1 l = f(3,i)*vectfx(3,ipntfx(i))
    f(1,i) = f(1,i) - dotprd*vectfx(1,ipntfx(i))
    f(2,i) = f(2,i) - dotprd*vectfx(2,ipntfx(i))
    f(3,i) = f(3,i) - dotprd*vectfx(3,ipntfx(i))
    goto 1000
    constrain to a point
    f(1,i) = 0.0
    f(2,i) = 0.0
    f(3,i) = 0.0
    goto 1000
    add a force
    f(1,i) = f(1,i) + vectfx(1,ipntfx(i))
    f(2,i) = f(2,i) + vectfx(2,ipntfx(i))
    f(3,i) = f(3,i) + vectfx(3,ipntfx(i))
    delta(1) = rv(1,i)-posfx(1,i)
    delta(2) = rv(2,i)-posfx(2,i)
    delta(3) = rv(3,i)-posfx(3,i)
delta(1) = delta(1) - perlen(1)*nint(delta(1)/perlen(1))
delta(2) = delta(2) - perlen(2)*nint(delta(2)/perlen(2))
delta(3) = delta(3) - perlen(3)*nint(delta(3)/perlen(3))
workfx(i) = workfx(i) +
1 vecrnx(i,ipntrf(i))*delta(1) +
2 vecrnx(i,ipntrf(i))*delta(2) +
3 vecrnx(i,ipntrf(i))*delta(3)
e(i) = e(i) - workfx(i)
posfx(1,i) = rv(1,i)
posfx(2,i) = rv(2,i)
posfx(3,i) = rv(3,i)
1000 continue
C reset the last time called variable
C
timefx = time
return
end
C******************************************************************************

settmp

subroutine settmp
include 'vap.inc'
dimension qdefault(3)
data qdefault/1730.,40.,0.1/,destmp/300.,/tmptim/1.0/
namelist /tmpcard/ ifxtmp, follow
namelist /regcard/ q, tmptim, destmp, xmin, xmax, ymin, ymax, zmin, zmax
ifxtmp=0
follow=.false.
C read the temperature control card
C
drag(0)=.0
read(5,tmpcard)
C
write(6,9229)
9229 format(///,'***** energy/temperature conditions '
C
if (ifxtmp.eq.0) then
write(6,9230)
9230 format(///,' constant energy',/)
C read in regcards (to be ignored)
C
100 destmp = -9999.
read(5,regcard)
if(destmp.eq.-9999.)return
goto 100
endif
C set time variable for fixtmp
C
tiemeqp = t0
tmpreg(0) = 0.
do 300 i = -1,nlatmax
300 ipntrg(i) = 0
C
reppmass = amass(1)
do 305 ityp=1,nitypes
305 reppmass = amass1(reppmass,amass(ityp))
C
nregs = 0
C
1000 q = qdefault(ifxtmp)
Appendix B. Molecular Dynamics Code

```c

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
fixtmp

subroutine fixtmp(time)
  include 'vap.inc'
  data boltz/8.617e-5/

  c if (ifxtmp.eq.0) return

  c if(.not. follow) then
  if (ivapor.eq.1) then
    ycenter(i) = ycenter(i) + 5.*TbndV*time
    ywidth(i) = ywidth(i) + 5.*TbndV*time
  endif
  do 200 i=1,natoms
    ipntrg(i) = 0
  do 201 ireg=1,nregs
    if(abs(rv(1,i)-xcenter(ireg)).gt.xwidth(ireg)) goto 201
    if(abs(rv(2,i)-xcenter(ireg)).gt.ywidth(ireg)) goto 201
    if(abs(rv(3,i)-xcenter(ireg)).gt.zwidth(ireg)) goto 201
  ipntrg(i) = ireg
  201 continue
  200 continue
  endif

  c deltat = time-timextmp
  c standard drag
  c if(ifxtmp.eq.1) then

  c calculate the temperature of each region
  c do 299 ireg=0,nregs
    natrg(ireg) = 0.
    do 300 i=1,natoms
      ireg = ipntrg(i)
      natrg(ireg) = natrg(ireg) + 1
      acttmp(ireg) = acttmp(ireg) +
        amass(itype(i))*(rv(4,i)**2 + rv(5,i)**2 + rv(6,i)**2)
    300 continue
    do 305 ireg=0,nregs
      if(natrg(ireg).ne.0)acttmp(ireg) = acttmp(ireg)/
        (3.*boltz*float(natrg(ireg)))
      if(threedim.eq.0) then
        acttmp(ireg) = acttmp(ireg)*1.5
      endif
    305 continue
  c add drag term to force
  c do 306 ireg=1,nregs
    drag(ireg) = (acttmp(ireg)-tmpreg(ireg))/qreg(ireg)
  306 continue
  do 307 i=1,natoms
    do 307 j = 1,3
    f(j,i) = f(j,i) - drag(ipntrg(1))*rv(j+3,i)
  307 f(j,i)

  c hoover drag
  c if(ifxtmp.eq.2) then

  c calculate the temperature of each region
  c do 399 ireg=0,nregs
    natrg(ireg) = 0
Appendix B. Molecular Dynamics Code

399          acttmp(ireg) = 0.
400          do 401 i=1,natoms
401          ireg = ipntrg(i)
402          natrg(ireg) = natrg(ireg) + 1
403          acttmp(ireg) = acttmp(ireg) +
404            .      amass(iotype(i))*(rv(4,i)**2 + rv(5,i)**2 + rv(6,i)**2)
405          continue
406          do 405 ireg=0,nregs
407          if(natrg(ireg).ne.0) acttmp(ireg) = acttmp(ireg)/
408              (3.*boltz*float(natrg(ireg)))
409          if(threedim .eq. .0) then
410          acttmp(ireg) = acttmp(ireg)*1.5
411          endif
412          continue
413          c
414          c add drag term to force
415          c
416          do 420 ireg=1,nregs
417          dragdot = (acttmp(ireg)-tmpreg(ireg))/greg(ireg)
418          drag(ireg) = drag(ireg) + dragdot*deltat
419          continue
420          do 421 i=1,natoms
421          j = 1,3
422          f(j,i) = f(j,i) - drag(ipntrg(i))*rv(j+3,i)
423          endif
424          c
425          c langevin
426          c
427          if( (ifxtmp.eq.3).and.(deltat.ne.0.0) ) then
428          do 530 ireg=1,nregs
429          drag(ireg) = greg(ireg)
430          do 540 i=1,natoms
431          j = 1,3
432          stoch = sqrt( 2.*drag(ipntrg(i))*boltz*tmpreg(ipntrg(i))/
433            deltat ) * xgauss()
434          f(j,i) = f(j,i) - drag(ipntrg(i))*rv(j+3,i) + stoch
435          endif
436          c
437          c timetmp = time
438          c
439          return
440          end

******************************************************************************

force

subroutine force(time)
include 'vap.inc'
common /wrkspc/ rneigh(neimax),dneigh(3,neimax),k(natmax),
  .jneigh(neimax)
common /nelbob/ nboxx,nboxy,nboxz,nboxx,nboxy,nboxz,nboxx,nboxy,
  nboxz,nxt,nzt,dnxt,dnzt
dimension p(natmax),z2(neimax),z2p(neimax),rhoip(neimax),
  .rhop(neimax),phi(neimax),phiip(neimax),psip(neimax),
  .fpn(neimax)
common /add/ natadd,ityadd(100)
c
500          nforce = nforce + 1
510          call chkdis
520          c add vapor atoms as required

c
Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
if (inte .eq. 1 .and. natadd .gt. 0) call addvl

clear the arrays

do 200 i = 1,natoms
r(i) = 0.
f(1,i) = 0.
f(2,i) = 0.
f(3,i) = 0.
slocal(1,1,i) = 0.
slocal(2,1,i) = 0.
slocal(3,1,i) = 0.
slocal(1,2,i) = 0.
slocal(2,2,i) = 0.
slocal(3,2,i) = 0.
slocal(1,3,i) = 0.
slocal(2,3,i) = 0.
slocal(3,3,i) = 0.
200 continue
do 210 kl = 1,3
stress(kl,1) = 0.0
stress(kl,2) = 0.0
stress(kl,3) = 0.0
210 continue

calculate total density at each atom

neitot = 0
neilind(0) = neitot
if (newlst .ne. 0) then
if (natadd .lt. 0) then
natadd=0
goto 664
endif
nboxnei=nboxnei+1
nxt=perlen(k)/rcrroot
dnxt=perlen(k)/float(nxt)
if (threedim .ne. 0) then
nxt=perlen(3)/rcrroot
dnxt=perlen(3)/float(nxt)
if (nxt .ne. 0) then
dnxt=perlen(3)
endif
endif
if (ivapor .ne. 1) then
nyt=perlen(2)/rcrroot
donyt=perlen(2)/float(nyt)
if (nyt .gt. nboxy) then
write(6,'*) 'nboxy is too small'
stop
endif
endif
if (nxt .gt. nboxx) then
write(6,'*) 'nboxx is too small'
stop
endif
if (nzt .gt. nboxz) then
write(6,'*) 'nboxz is too small'
stop
endif
if (ivapor .eq. 1) ntopbox=0
do 663 i=1,natoms
nw1=int((rv(i,1)-perlb(1))/dnxt)+1
nw2=int((rv(i,2)-perlb(2))/dnxt)+1
if (threedim .ne. 0) nw3=int((rv(i,3)-perlb(3))/dnz)+1
if (threedim .eq. 0) nw3=1
if (nbox(nw1,nw2,nw3,nboxx) .ne. nboxnei) then
nbox(nw1,nw2,nw3,nboxn1)=0
nbox(nw1,nw2,nw3,nboxn1)=nboxnei
if (itvapor.eq.1.and.nw2.gt.ntopbox.and.
  *itype(i).ne.inert) ntopbox=nw2
endif
nbox(nw1,nw2,nw3,nboxn1)=nbox(nw1,nw2,nw3,nboxn1)+1
nbox(nw1,nw2,nw3,nbox(nw1,nw2,nw3,nboxn1))=1
663 continue
endif
664 continue
do 1000 i=1,natoms
c c obtain the information about the neighbors of atom i
c c call gneih(i)
c c compute the contribution to rho(i) from particle j
c
do 1300 j=1,nneips
  rneih(j) = sqrt(rneih(j))
p(j) = rneih(j)*drar + 1.0
  k(j) = p(j)
k(j) = min0(k(j),nrar-l)
p(j) = p(j) - k(j)
p(j) = amin1(p(j),1.0)
1300 continue
do 1400 j=1,nneips
  rho(i) = rho(i) +
  $ (rhorarl(k(j),itype(jneih(j))))*p(j) +
  $ rhorar2(k(j),itype(jneih(j))))*p(j) +
  $ rhorar3(k(j),itype(jneih(j))))*p(j) +
  $ rhorar4(k(j),itype(jneih(j))))
1400 continue
c c compute the contribution to rho(j) from particle i
c c ity = itype(i)
do 1600 j=1,nneih
  rho(jneih(j)) = rho(jneih(j)) +
  $ ((rhorarl(k(j),ity)*p(j)+rhorar2(k(j),ity)*p(j)+
  $ rhorar3(k(j),ity)*p(j)+rhorar4(k(j),ity)
1600 continue
nneighs(i) = nneih
do 1700 j=1,nneips
  nelot = nelot + 1
  jneighs(neilotal) = jneih(j)
dneighs(1,neilotal) = dneigh(1,j)
dneighs(2,neilotal) = dneigh(2,j)
dneighs(3,neilotal) = dneigh(3,j)
1700 continue
neilind(i) = nelot
1000 continue
c c now find f and fp for each particle
c c do 2000 i=1,natoms
  p(i) = rho(i)*xdrhoar + 1.0
  k(i) = p(i)
k(i) = max1(min(k(i),nrhoar-1))
p(i) = p(i) - k(i)
2000 continue
c c store fsui in e(i) and the derivative of fsui in fp
c c do 2100 i=1,natoms
  e(i) = ((frhoar2(k(i),itype(i)))*p(i) +
  $ frhoar2(k(i),itype(i)))*p(i) +
Appendix B. Molecular Dynamics Code

\$ \text{frhoar1}(k(i), \text{itype}(i)) \text{p(i)} + \\
\$ \text{frhoar}(k(i), \text{itype}(i)) \\
\text{fp}(i) = (\text{frhoar6}(k(i), \text{itype}(i)) \text{p(i)} + \\
\$ \text{frhoar5}(k(i), \text{itype}(i)) \text{p(i)} + \\
\$ \text{frhoar4}(k(i), \text{itype}(i)) \\

2100 \text{continue} \\
\text{c} \ \ \ \ \text{add up energy and forces} \\
\text{c} \\
\text{do 3000} \ i = 1, \text{natoms} \\
\text{c} \\
\text{nneips} = \text{neind}(i) - \text{neind}(i-1) \\
\text{nneigh} = \text{nneighs}(i) \\
\text{do 3100} \ j = 1, \text{nneips} \\
\text{j} = \text{neind}(i-1) + j \\
\text{jneigh}(j) = \text{jneighs}(jj) \\
\text{dneigh}(1, j) = \text{dneighs}(1, jj) \\
\text{dneigh}(2, j) = \text{dneighs}(2, jj) \\
\text{dneigh}(3, j) = \text{dneighs}(3, jj) \\
\text{rneigh}(j) = \sqrt{\text{dneigh}(1, j)^{**2} + \text{dneigh}(2, j)^{**2} + \\
\text{dneigh}(3, j)^{**2}} \\
\text{3100} \text{continue} \\
\text{do 3200} \ j = 1, \text{nneips} \\
\text{p}(j) = \text{rneigh}(j)^{rdarar + 1.0} \\
\text{k}(j) = \text{p}(j) \\
\text{k}(j) = \text{min}(\text{p}(j), \text{nrar}-1) \\
\text{p}(j) = \text{p}(j) - \text{k}(j) \\
\text{p}(j) = \text{amin}(\text{p}(j), 1.0) \\
\text{3200} \text{continue} \\
\text{c} \\
\text{find rhoj} \\
\text{c} \\
\text{ity} = \text{itype}(i) \\
\text{do 3400} \ j = 1, \text{nneips} \\
\text{rhoj}(j) = (\text{zhorar6}(k(j), \text{ity}) \text{p(j)} + \text{zhorar5}(k(j), \text{ity})) \text{p(j)} + \\
\$ \text{zhorar4}(k(j), \text{ity}) \\
\text{3400} \text{continue} \\
\text{c} \\
\text{find rhoj} \\
\text{c} \\
\text{find z**2, (z**2)-prime} \\
\text{c} \\
\text{do 3500} \ j = 1, \text{nneips} \\
\text{jty} = \text{itype}(\text{jneigh}(j)) \\
\text{z2}(j) = (\text{z2rar3}(k(j), \text{ity}, \text{jty}) \text{p(j)} + \\
\$ \text{z2rar2}(k(j), \text{ity}, \text{jty}) \text{p(j)} + \\
\$ \text{z2rar1}(k(j), \text{ity}, \text{jty}) \text{p(j)} + \\
\$ \text{z2rar}(k(j), \text{ity}, \text{jty}) \\
\text{z2p}(j) = (\text{z2rar6}(k(j), \text{ity}, \text{jty}) \text{p(j)} + \text{z2rar5}(k(j), \text{ity}, \text{jty})) \text{p(j)} + \\
\$ \text{z2rar4}(k(j), \text{ity}, \text{jty}) \\
\text{rhojp}(j) = (\text{zhorar6}(k(j), \text{ity}, \text{jty}) \text{p(j)} + \text{zhorar5}(k(j), \text{ity}, \text{jty})) \text{p(j)} + \\
\$ \text{zhorar4}(k(j), \text{ity}, \text{jty}) \\
\text{fpn}(j) = \text{fp}(\text{jneigh}(j)) \\
\text{3500} \text{continue} \\
\text{c} \\
\text{compute the contribution to the energy, pressure and forces at} \\
\text{c} \\
\text{particle i due to particle j} \\
\text{c} \\
\text{do 3700} \ j = 1, \text{nneips} \\
\text{p(j)} = 1.0 / \text{nneigh}(j) \\
\text{dneigh}(1, j) = \text{dneigh}(1, j) + \text{p(j)} \\
\text{dneigh}(2, j) = \text{dneigh}(2, j) + \text{p(j)} \\
\text{dneigh}(3, j) = \text{dneigh}(3, j) + \text{p(j)} \\
\text{phi}(j) = \text{z2}(j) \text{p(j)} \\
\text{phip}(j) = \text{z2p}(j) \text{p(j)} - \text{phi}(j) \text{p(j)} \\
\text{e(i)} = \text{e(i)} + 0.5 \text{phi}(j) \\
\text{psip}(j) = \text{fp}(i) \text{rhojp}(j) + \text{fpn}(j) \text{rhojp}(j) + \text{phip}(j)
Appendix B. Molecular Dynamics Code

\[ f(1,i) = f(1,i) - psip(j) \cdot dneigh(1,j) \]
\[ f(2,i) = f(2,i) - psip(j) \cdot dneigh(2,j) \]
\[ f(3,i) = f(3,i) - psip(j) \cdot dneigh(3,j) \]

3700 continue

compute the contribution to the energy, forces, and stress
tensor and forces at particle j due to particle i

do 3800 j=1,nneigh
\[ e(jneigh(j)) = e(jneigh(j)) + 0.5 \cdot \phi(j) \]
\[ f(1,jneigh(j)) = f(1,jneigh(j)) + psip(j) \cdot dneigh(1,j) \]
\[ f(2,jneigh(j)) = f(2,jneigh(j)) + psip(j) \cdot dneigh(2,j) \]
\[ f(3,jneigh(j)) = f(3,jneigh(j)) + psip(j) \cdot dneigh(3,j) \]

3800 continue

compute the stress tensor

\[ \text{stress}(1,1) = \text{stress}(1,1) + \text{amass}(i\text{type}(i)) \cdot rv(4,i) \cdot rv(4,i) \]
\[ \text{stress}(1,2) = \text{stress}(1,2) + \text{amass}(i\text{type}(i)) \cdot rv(4,i) \cdot rv(5,i) \]
\[ \text{stress}(1,3) = \text{stress}(1,3) + \text{amass}(i\text{type}(i)) \cdot rv(4,i) \cdot rv(6,i) \]
\[ \text{stress}(2,2) = \text{stress}(2,2) + \text{amass}(i\text{type}(i)) \cdot rv(5,i) \cdot rv(5,i) \]
\[ \text{stress}(2,3) = \text{stress}(2,3) + \text{amass}(i\text{type}(i)) \cdot rv(5,i) \cdot rv(6,i) \]
\[ \text{stress}(3,3) = \text{stress}(3,3) + \text{amass}(i\text{type}(i)) \cdot rv(6,i) \cdot rv(6,i) \]

do 3910 j = 1,nneips
\[ psip(j) = psip(j) \cdot dneigh(j) \]

the 1/2 is factored in during loop 4000

\[ \text{slocal}(1,1,i) = \text{slocal}(1,1,i) - psip(j) \cdot dneigh(1,j) \cdot dneigh(1,j) \]
\[ \text{slocal}(1,2,i) = \text{slocal}(1,2,i) - psip(j) \cdot dneigh(1,j) \cdot dneigh(2,j) \]
\[ \text{slocal}(1,3,i) = \text{slocal}(1,3,i) - psip(j) \cdot dneigh(1,j) \cdot dneigh(3,j) \]
\[ \text{slocal}(2,2,i) = \text{slocal}(2,2,i) - psip(j) \cdot dneigh(2,j) \cdot dneigh(2,j) \]
\[ \text{slocal}(2,3,i) = \text{slocal}(2,3,i) - psip(j) \cdot dneigh(2,j) \cdot dneigh(3,j) \]
\[ \text{slocal}(3,3,i) = \text{slocal}(3,3,i) - psip(j) \cdot dneigh(3,j) \cdot dneigh(3,j) \]

3910 continue

do 3920 j = 1,nneigh
\[ \text{slocal}(1,1,jneigh(j)) = \text{slocal}(1,1,jneigh(j)) - psip(j) \cdot dneigh(1,j) \cdot dneigh(1,j) \]
\[ \text{slocal}(1,2,jneigh(j)) = \text{slocal}(1,2,jneigh(j)) - psip(j) \cdot dneigh(1,j) \cdot dneigh(2,j) \]
\[ \text{slocal}(1,3,jneigh(j)) = \text{slocal}(1,3,jneigh(j)) - psip(j) \cdot dneigh(1,j) \cdot dneigh(3,j) \]
\[ \text{slocal}(2,2,jneigh(j)) = \text{slocal}(2,2,jneigh(j)) - psip(j) \cdot dneigh(2,j) \cdot dneigh(2,j) \]
\[ \text{slocal}(2,3,jneigh(j)) = \text{slocal}(2,3,jneigh(j)) - psip(j) \cdot dneigh(2,j) \cdot dneigh(3,j) \]
\[ \text{slocal}(3,3,jneigh(j)) = \text{slocal}(3,3,jneigh(j)) - psip(j) \cdot dneigh(3,j) \cdot dneigh(3,j) \]

3920 continue

3000 continue

sum up the local contributions to the stress tensor to get the total stress
tensor and compute the other half of the symmetric matrix

do 4000 i = 1,natoms
\[ \text{slocal}(1,1,i) = 0.5 \cdot \text{slocal}(1,1,i) \]
\[ \text{slocal}(1,2,i) = 0.5 \cdot \text{slocal}(1,2,i) \]
\[ \text{slocal}(1,3,i) = 0.5 \cdot \text{slocal}(1,3,i) \]
\[ \text{slocal}(2,2,i) = 0.5 \cdot \text{slocal}(2,2,i) \]
\[ \text{slocal}(2,3,i) = 0.5 \cdot \text{slocal}(2,3,i) \]
\[ \text{slocal}(3,3,i) = 0.5 \cdot \text{slocal}(3,3,i) \]
\[ \text{slocal}(2,1,i) = \text{slocal}(1,2,i) \]
\[ \text{slocal}(3,1,i) = \text{slocal}(1,3,i) \]
\[ \text{slocal}(3,2,i) = \text{slocal}(2,3,i) \]
Appendix B. Molecular Dynamics Code

```plaintext
stress(1,1) = stress(1,1) + slocal(1,1,1)
stress(1,2) = stress(1,2) + slocal(1,2,1)
stress(1,3) = stress(1,3) + slocal(1,3,1)
stress(2,2) = stress(2,2) + slocal(2,2,1)
stress(2,3) = stress(2,3) + slocal(2,3,1)
stress(3,3) = stress(3,3) + slocal(3,3,1)

4000 continue
stress(2,1) = stress(1,2)
stress(3,1) = stress(1,3)
stress(3,2) = stress(2,3)

if (ibdtype.eq.2)
e(natoms+1) = dpres*pertlen(1)*pertlen(2)*pertlen(3)
    + 0.5*(dstress(1)*pertlen(1)**2
     + dstress(2)*pertlen(2)**2
     + dstress(3)*pertlen(3)**2)

add the temperature constraints

call fixtmp(time)

add the constraints

call fixfor(time)
return
end

******************************************************************************

gneigh

subroutine gneigh(i)
include 'vap.inc'
common /wrkspc/ rneigh(neimax),dneigh(3,neimax),k(natmax),
.jneigh(neimax)
common /neibob/ nboxx,nboxy,nboxz,nboxn,nxt,nzt,dntdzt,
dimension nsearch1(neimax),nsearch2(neimax),nsearch3(neimax)
dimension dis(3,natmax),r(natmax)

if (newlst.ne.0) goto 2500

if here then use the neighbors found earlier

jend = nnindex(i) - nnindex(i-1)
do 2108 jtmp = 1,jend
   k(jtmp) = nnlist(jtmp+nnindex(i-1))
   dis(1,jtmp) = rv(1,k(jtmp))
   dis(2,jtmp) = rv(2,k(jtmp))
   dis(3,jtmp) = rv(3,k(jtmp))
2108 continue

do 2110 jtmp = 1,jend
   dis(1,jtmp) = rv(1,i) - dis(1,jtmp)
   dis(1,jtmp) = dis(1,jtmp) - pertlen(1)*nint(dis(1,jtmp)/pertlen(1))
   dis(2,jtmp) = rv(2,i) - dis(2,jtmp)
   dis(2,jtmp) = dis(2,jtmp) - pertlen(2)*nint(dis(2,jtmp)/pertlen(2))
   dis(3,jtmp) = rv(3,i) - dis(3,jtmp)
   dis(3,jtmp) = dis(3,jtmp) - pertlen(3)*nint(dis(3,jtmp)/pertlen(3))
   r(jtmp) = dis(1,jtmp)**2 + dis(2,jtmp)**2 + dis(3,jtmp)**2
2110 continue

rneigh = 0

do 2100 jtmp = 1,jend

determine which pairs are separated by less than rcut
and store the needed information about these pairs
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Appendix B. Molecular Dynamics Code

```fortran
if (r(jtmp).gt.rcutsq) go to 2100
        j = k(jtmp)
        nneigh = nneigh + 1
        kneigh(nneigh) = r(jtmp)
        jneigh(nneigh) = j
        do 2120 kcoord = 1,3
            dneigh(kcoord,nneigh) = dis(kcoord,jtmp)
            if(onlyi) go to 2100
            disz = -sign(perlen(3),dis(3,jtmp))
            rim = r(jtmp) + 2.*dis(3,jtmp)*disz + disz**2
            if (rim.gt.rcutsq) go to 2100
            nneigh = nneigh + 1
            kneigh(nneigh) = rim
            jneigh(nneigh) = j
            dneigh(1,nneigh) = dis(1,jtmp)
            dneigh(2,nneigh) = dis(2,jtmp)
            dneigh(3,nneigh) = dis(3,jtmp) + disz
            if(.not.threei) go to 2100
            disz = -disz
            rim = r(jtmp) + 2.*dis(3,jtmp)*disz + disz**2
            if (rim.gt.rcutsq) go to 2100
            nneigh = nneigh + 1
            kneigh(nneigh) = rim
            jneigh(nneigh) = j
            dneigh(1,nneigh) = dis(1,jtmp)
            dneigh(2,nneigh) = dis(2,jtmp)
            dneigh(3,nneigh) = dis(3,jtmp) + disz
            if(.not.fouri) go to 2100
            disz = -2.*sign(perlen(3),dis(3,jtmp))
            rim = r(jtmp) + 2.*dis(3,jtmp)*disz + disz**2
            if (rim.gt.rcutsq) go to 2100
            nneigh = nneigh + 1
            kneigh(nneigh) = rim
            jneigh(nneigh) = j
            dneigh(1,nneigh) = dis(1,jtmp)
            dneigh(2,nneigh) = dis(2,jtmp)
            dneigh(3,nneigh) = dis(3,jtmp) + disz
            continue
        2100
        nneips = nneigh
        if(threei) then
            nneips = nneips + 1
            disz = perlen(3)
            rim = disz**2
            rneigh(nneips) = rim
            jneigh(nneips) = i
            dneigh(1,nneips) = 0.0
            dneigh(2,nneips) = 0.0
            dneigh(3,nneips) = disz
            nneips = nneips + 1
            disz = -disz
            rneigh(nneips) = rim
            jneigh(nneips) = i
            dneigh(1,nneips) = 0.0
            dneigh(2,nneips) = 0.0
            dneigh(3,nneips) = disz
        endif
        if(nneips.gt.neimax) then
            write(6,9212) nneips,neimax
            format(' number of neighbors',i5,' exceeds array bound ',i5)
            stop
        endif

9212 c compute the maximum number of neighbors seen so far

    neimax = max0(neimax,nneips)
    return
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
end of neighbor finding when using old neighbor list

determine new neighbor list while getting the neighbors

nnindx(i) = nnindx(i-1)
nneigh = 0
niz=int((rv(1,i)-perlb(1))/dnxt)+1
niy=int((rv(2,i)-perlb(2))/dnyt)+1
if (threedim .ne. .0) then
  if (nzt .le. 3) then
    nzaa=1
    nzbb=nzt
  else
    nzaa=niz-1
    nzbb=niz+1
  endif
  endif
  if (threedim .eq. .0) then
    nzaa=1
    nzbb=1
  endif
  if (nxt .le. 3) then
    nkaa=1
    nkbb=nxt
  else
    nkaa=niz-1
    nkbb=niz+1
  endif
  if (nyt .le. 3) then
    nyya=1
    nymb=nyt
  else
    nyya=niy-1
    nymb=niy+1
  endif
 endif
  nsearch1(neimax)=0
  do 2600  j1=nkaa,nkbb
    k1=j1
    if (k1 .lt. 1) k1=nxt
    if (k1 .gt. nxt) k1=1
    do 2601  j2=nyya,nymb
      k2=j2
      if (nyt .gt. nboxy) then
        if (k2 .lt. 1) goto 2601
        if (k2 .gt. nboxy) then
          write(*,'(a)') 'nboxy is too small'
          stop
        endif
      endif
      if (k2 .lt. 1) k2=nyt
      if (k2 .gt. nyt) k2=1
      endif
      do 2602  j3=nzaa,nzbb
        k3=j3
        if (k3 .lt. 1) k3=nzt
        if (k3 .gt. nzt .and. k3 .ne. 1) k3=1
        if (nbox(k1,k2,k3,nboxn) .ne. nboonei) goto 2602
        do 2603  j4=1,nbox(k1,k2,k3,nboxn1)
          j=nbox(k1,k2,k3,j4)
          if (j .ge. i) goto 2603
          if (itype(j) .eq. inert) goto 2603
          dis(i,j) = rv(1,i) - rv(1,j)
          dis(i,j) = dis(i,j) - perlen(1)*nint(dis(i,j)/perlen(1))
Appendix B. Molecular Dynamics Code

\[ \text{dis}(2,j) = \text{rv}(2,1) - \text{rv}(2,j) \]
\[ \text{dis}(2,j) = \frac{\text{dis}(2,j) - \text{perlen}(2) \times \text{nint}(\text{dis}(2,j)/\text{perlen}(2))}{\text{perlen}(2)} \]
\[ \text{dis}(3,j) = \text{rv}(3,1) - \text{rv}(3,j) \]
\[ \text{dis}(3,j) = \frac{\text{dis}(3,j) - \text{perlen}(3) \times \text{nint}(\text{dis}(3,j)/\text{perlen}(3))}{\text{perlen}(3)} \]
\[ r(j) = \text{dis}(1,j)^{**2} + \text{dis}(2,j)^{**2} + \text{dis}(3,j)^{**2} \]
if \( r(j) < \text{gt.rctsq} \) goto 2603
\[ \text{nsearch}1(\text{neimax}) = \text{nsearch}1(\text{neimax}) + 1 \]
\[ \text{nsearch}1(\text{nsearch}1(\text{neimax})) = j \]
2603 continue
2602 continue
2601 continue
2600 continue
\[ \text{nub} = \text{nsearch}1(\text{neimax}) \]
\[ \text{call molaaf}(\text{nsearch}1(1), \text{nub}, \text{nsearch}3, \text{nsearch}2, \text{ifail}) \]
do 2700 j=1, \text{nsearch}1(\text{neimax})
\[ j = \text{nsearch}1(\text{nsearch}2(j)) \]
\[ j = \text{nsearch}1(j) \]
store the index of the particle
\[ \text{nnindx}(i) = \text{nnindx}(i) + 1 \]
\[ \text{nnlist}(\text{nnindx}(i)) = j \]
determine which pairs are separated by less than rcut
and store the needed information about these pairs
if \( r(j) > \text{gt.rctsq} \) go to 2700
\[ \text{nneigh} = \text{nneigh} + 1 \]
\[ \text{nneigh}(\text{nneigh}) = r(j) \]
\[ \text{jneigh}(\text{nneigh}) = j \]
do 2750 kcoord = 1,3
\[ \text{dneigh}(\text{kcoord}, \text{nneigh}) = \text{dis}(\text{kcoord}, j) \]
if (onlyol) go to 2700
\[ \text{disz} = -2 \times \text{sign}(\text{perlen}(3), \text{dis}(3,j)) \]
\[ \text{rim} = r(j) + 2 \times \text{dis}(3,j) \times \text{disz} + \text{disz}^{**2} \]
if (rim > \text{gt.rctsq}) go to 2700
\[ \text{nneigh} = \text{nneigh} + 1 \]
\[ \text{nneigh}(\text{nneigh}) = \text{rim} \]
\[ \text{jneigh}(\text{nneigh}) = j \]
\[ \text{dneigh}(1, \text{nneigh}) = \text{dis}(1,j) \]
\[ \text{dneigh}(2, \text{nneigh}) = \text{dis}(2,j) \]
\[ \text{dneigh}(3, \text{nneigh}) = \text{dis}(3,j) + \text{disz} \]
if (.not. threei) go to 2700
\[ \text{disz} = -\text{disz} \]
\[ \text{rim} = r(j) + 2 \times \text{dis}(3,j) \times \text{disz} + \text{disz}^{**2} \]
if (rim > \text{gt.rctsq}) go to 2700
\[ \text{nneigh} = \text{nneigh} + 1 \]
\[ \text{nneigh}(\text{nneigh}) = \text{rim} \]
\[ \text{jneigh}(\text{nneigh}) = j \]
\[ \text{dneigh}(1, \text{nneigh}) = \text{dis}(1,j) \]
\[ \text{dneigh}(2, \text{nneigh}) = \text{dis}(2,j) \]
\[ \text{dneigh}(3, \text{nneigh}) = \text{dis}(3,j) + \text{disz} \]
if (.not. fouri) go to 2700
\[ \text{disz} = -2 \times \text{sign}(\text{perlen}(3), \text{dis}(3,j)) \]
\[ \text{rim} = r(j) + 2 \times \text{dis}(3,j) \times \text{disz} + \text{disz}^{**2} \]
if (rim > \text{gt.rctsq}) go to 2700
\[ \text{nneigh} = \text{nneigh} + 1 \]
\[ \text{nneigh}(\text{nneigh}) = \text{rim} \]
\[ \text{jneigh}(\text{nneigh}) = j \]
\[ \text{dneigh}(1, \text{nneigh}) = \text{dis}(1,j) \]
\[ \text{dneigh}(2, \text{nneigh}) = \text{dis}(2,j) \]
\[ \text{dneigh}(3, \text{nneigh}) = \text{dis}(3,j) + \text{disz} \]
2700 continue
\[ \text{nneilps} = \text{nneigh} \]
if (threei) then
\[ \text{nneilps} = \text{nneilps} + 1 \]
\[ \text{dis} = \text{perlen}(3) \]
Appendix B. Molecular Dynamics Code

```fortran
rim = disz**2
rneigh(nneips) = rim
jneigh(nneips) = i
dneigh(1,nneips) = 0.0
dneigh(2,nneips) = 0.0
dneigh(3,nneips) = disz
nneips = nneips + 1
disz = -disz
rneigh(nneips) = rim
jneigh(nneips) = i
dneigh(1,nneips) = 0.0
dneigh(2,nneips) = 0.0
dneigh(3,nneips) = disz
endif
if(nneips.gt.neimax)then
  write(6,9222)nneips,neimax
9222  format(' number of neighbors','i5,' exceeds array bound ','i5)
  stop
endif

c compute the maximum number of neighbors seen so far
c
  neimax = max0(nneimx,nneips)
c
  if this is the last call to gneigh, then set newlst=0 to indicate
c  that the current list can be used and also increment ngltlst
c
  if (i.eq.natoms) then
    newlst = 0
    ngltlst = ngltlst + 1
  endif
  return
end

c*******************************************************************************

chkdis

subroutine chkdis
include 'vap.inc'
dimension dis(3,natmax),r(natmax)
common /mm3rold/ rold(3,natmax)
c
  if (newlst.eq.1) goto 2500
c
  compare the new positions with the old ones
c
  do 2100 i=1,natoms
    dis(1,i) = rold(1,i) - rv(1,i)
    dis(2,i) = rold(2,i) - rv(2,i)
    dis(3,i) = rold(3,i) - rv(3,i)
    dis(1,i) = dis(1,i) - perlen(1)*nint(dis(1,i)/perlen(1))
    dis(2,i) = dis(2,i) - perlen(2)*nint(dis(2,i)/perlen(2))
    dis(3,i) = dis(3,i) - perlen(3)*nint(dis(3,i)/perlen(3))
    r(i) = dis(1,i)**2 + dis(2,i)**2 + dis(3,i)**2
  2100 continue
c
c  determine the maximum displacement and compare with dradm
c
drmax1 = 0.0
drmax2 = 0.0
do 2200 i = 1,natoms
  tmp = amin1(drmax1,r(i))
  drmax1 = amin1(drmax1,r(i))
  drmax2 = amin1(drmax2,tmp)
  2200 continue
```
Appendix B. Molecular Dynamics Code

```fortran
  dmax = sqrt(dmax1) + sqrt(dmax2)
  if (dmax.gt.dradd) goto 2500
  c
  c if here the old neighbor list can be used
  c
  newlst = 0
  return
  2500 continue
  c
  c if here, a new neighbor list is needed so store the current coordinates
  c
  newlst = 1
  do 2600 j = 1, natoms
    2600 r0ld(j,i) = rv(j,i)
  return
  end

c********************************************************************

gauss

  function rgauss()
  implicit real(a-h,o-z)
  data twopi/6.283185308/
  10 continue
    a1 = ran1()
    if (a1.eq.0.) goto 10
    a2 = ran1()
    rgauss = sqrt(-2.*log(a1))*cos(twopi*a2)
  return
  end

  c********************************************************************

rand1

  c random number functions

  real function ran1()
  common /ran/ iseed
  iseed=iseed+125
  iseed=iseed-2796203*(iseed/2796203)
  ran1=abs(iseed/2796203.0)
  return
  end

  c********************************************************************

sdot

  function sdot(n,x,incx,y,incy)
  implicit real(a-h,o-z)
  dimension x(n),y(n)
  tot = 0.0
  do 100 i = 1,n
    ix = 1 + (i-1)*incx
    iy = 1 + (i-1)*incy
    tot = tot + x(ix)*y(iy)
  100 continue
  sdot = tot
  return
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Appendix B. Molecular Dynamics Code

saxpy

subroutine saxpy(n,a,x,incx,y,incy)
  implicit real(a-h,o-z)
  dimension x(n),y(n)
  do 100 i = 1,n
    ix = 1 + (i-1)*incx
    iy = 1 + (i-1)*incy
    y(iy) = a*x(ix) + y(iy)
  100 continue
  return
end

snrm2

function snrm2(n,x,incx)
  dimension x(1)
  ilast = n*inc
  sum = 0.
  do 100 i = 1,ilast,inc
    sum = sum + x(i)**2
  100 continue
  snrm2 = sqrt(sum)
  return
end

scopy

subroutine scopy(n,x,incx,y,incy)
  dimension x(1),y(1)
  do 100 i = 0,n-1
    jx = 1 + i*incx
    jy = 1 + i*incy
    y(jy) = x(jx)
  100 continue
  return
end

m01aaf

subroutine m01aaf(a,ii,jj,ip,ia,ifail)
  c
  dimension a(jj),ia(jj),ip(jj),iu(16),il(16)
  ifail = 0
  if (ii.gt.jj) ifail = 1
  if (ii.lt.1) ifail = 2
  if (jj.lt.1) ifail = 3
  if (ifail.ne.0) return
  do 2 i = ii,jj
    2 ia(i) = i
    m = 1
    j = ii
    k = i
    if (i.ge.j) goto 70
  10 k = i
    ij = (i+j)/2
t = a(ij)
it = ia(ij)
if (a(i).le.t) goto 20
a(ij) = a(i)
a(i) = t
t = a(ij)
ia(ij) = ia(i)
ita = ia(ij)
20     l = j
if (a(j) .ge. t) goto 40
a(ij) = a(j)
a(j) = t
t = a(ij)
ia(ij) = ia(j)
ita = it
if (a(l) .le. t) goto 40
a(ij) = a(i)
a(i) = t
t = a(ij)
ia(ij) = ia(i)
ita = it
goto 40
30   a(l) = a(k)
a(k) = tt
itia = ia(k)
itia = itt
40     l = l - 1
if (a(l) .gt. t) goto 40
tt = a(l)
itta = ia(l)
50     k = k + 1
if (a(k) .lt. t) goto 50
if (k .le. l) goto 30
if (l-l .le. j-k) goto 60
ia(m) = i
ia(m) = l
i = k
m = m + 1
goto 80
60   ia(m) = k
ia(m) = j
j = l
m = m + 1
goto 80
70     m = m - 1
if (m .eq. 0) goto 1000
i = il(m)
70     m = m - 1
80 if (j-i .ge. i) goto 10
if (i .eq. i) goto 5
i = i - 1
90     i = i + 1
if (i .eq. j) goto 70
it = ia(i+1)
t = a(i+1)
if (t .lt. a(k)) goto 100
a(k+1) = t
ia(k+1) = ia(k)
k = k - 1
if (t .le. a(k)) goto 100
a(k+1) = t
ia(k+1) = it
goto 90
100     continue
do 1100 i = ii,jj
1100 ip(ia(i)) = i

C
C do 2000 k = ii,jj
C if (ia(k).lt.0) goto 2000
C kl = k
C ik1 = ia(kl)
C temp = a(kl)
2100 continue
C ia(kl) = -ia(kl)
C store = a(ik1)
C a(ik1) = temp
C temp = store
C kl = ik1
C ik1 = ia(kl)
C if (ik1.gt.0) goto 2100
2000 continue
C do 2200 k = ii,jj
2200 ia(k) = -ia(k)
C return
C end

C******************************************************************************

Following several subroutines are for timing.

C timing routines
C
C subroutine initsec()
C common /seccom/ start
C start=mclock()/100.0
C return
C end
C******************************************************************************

C function seconds()
C common /seccom/ start
C seconds=mclock()/100.0-start
C return
C end
C******************************************************************************

C subroutine mytime(timestr)
C integer*4 time
C character*24 timestr
C call ctime_(timestr,time())
C return
C end
C******************************************************************************

C The following timing functions run on sparc stations
C******************************************************************************

subroutine initsec()
implicit real(a-h,o-z)
real*4 tarray(2),start,etime
common /seccom/ start
start = etime(tarray)
return
end

C******************************************************************************

function seconds()
implicit real(a-h,o-z)
real*4 tarray(2),start,etime
common /seccom/ start
t = etime(tarray) - start
seconds = t
return
end
C******************************************************************************
Appendix B. Molecular Dynamics Code

subroutine mytime(timestr)
  character*4 timestr,ctime
  integer*4 time
  timestr = ctime(time())
  return
end

******************************************************************************

addv1

subroutine addv1
  include 'vap.inc'
  common /uu/ u1(3,natmax),u2(3,natmax),u3(3,natmax),
  *u4(3,natmax),u5(3,natmax)
  common /add/ natadd,ityadd(100)
  common /nm3r3d/ rold(3,natmax)
  common /neibob/ nbox(nxbo,nybox,nboxx,nboxn),nxt,nzt,dnxt,dnze
  if (depdirtme .gt. 0.0) then
    ptheta=sqrt(1-depdir(2)*depdir(2))
    depdir(1)=ptheta*cos(dir(2))
    depdir(3)=ptheta*sin(dir(2))
    mark1=1
    mark2=1
    if (depdir(1) .lt. 0.0) mark1=-1
    if (depdir(3) .lt. 0.0) mark2=-1
    endif
    nboxnei=nboxnei+1
    ntopbox=0
    if (ldv (eq. 2) then
      nxt=perlen(1)/rctroot
      dnxt=perlen(1)/float(nxt)
    if (threedim .ne. 0.0) then
      nxt=perlen(3)/rctroot
      if (nzt .ne. 0.0) dnz=perlen(3)/float(nzt)
    if (nzt .eq. 0.0) then
      nzt=1
      dnz=perlen(3)
    endif
  endif
  if (nzt .gt. 0.0) then
    write(6,*) 'nbox is too small'
    stop
  endif
  if (nzt .gt. 0.0) then
    write(6,*) 'nbox is too small'
    stop
  endif
  n1=natoms+1
  n2=n1+natadd
  do 1 i=1,3
    y(i,n2)=y(i,n1)
    y(i+3,n2)=y(i+3,n1)
    u1(i,n2)=u1(i,n1)
    u2(i,n2)=u2(i,n1)
    u3(i,n2)=u3(i,n1)
    u4(i,n2)=u4(i,n1)
    u5(i,n2)=u5(i,n1)
 1 continue
  do 663 i=1,natoms
    nw1=int((rv(i,1)-perlb(1))/dnxt)+1
    nw2=int((rv(i,1)-perlb(2))/dnxt)+1
    if (threedim .ne. 0) nw3=int((rv(i,1)-perlb(3))/dnzt)+1
    if (threedim .eq. 0) nw3=1
    if (nbox(nw1,nw2,nw3,nboxn) .ne. nboxnei) then
      nbox(nw1,nw2,nw3,nboxn)=0
 663 continue
Appendix B. Molecular Dynamics Code

```plaintext
nbox(nw1,nw2,nw3,nboxn)=nboxnei
if nw2 .gt. ntopbox .and. itype(i) .ne. inert) ntopbox=nw2
endif
nbox(nw1,nw2,nw3,nboxn)=nbox(nw1,nw2,nw3,nboxn)+1
nbox(nw1,nw2,nw3,nbox(nw1,nw2,nw3,nboxn))=i
continue
    do 2 i=1,natadd
        xemit=rnsl()*(perlen(1)+perl(1))
yemit=dnyt*(ntopbox+2.0)+perl(2)
        if (threeedim .ne. 1.0) then
            zemit=0.0
        else
            zemit=rnsl()+perlen(3)+perl(3)
        endif
        natoms=natoms+1
        if (ityadd(i) .eq. inert) then
            itype(natoms)=ityadd(i)
            velo=sqrt(2.0*einert/amass(inert))
        else
            tst=rnsl()
            it=0
            if (layernumber .eq. 1) then
                do 201 mm=ntypes,1,-1
                if (tst .lt. fraction1(mm)) it=typel(mm)
            endif
            if (layernumber .eq. 2) then
                do 202 mm=ntypes,1,-1
                if (tst .lt. fraction2(mm)) it=typet2(mm)
            endif
            if (it .eq. 0) it=1
            itype(natoms)=it
            velo=sqrt(2.0*evapor/amass(it))
        endif
        velx=velo*depdir(1)
        vely=velo*depdir(2)
        velz=velo*depdir(3)
        shortest=1000.0
        more=0
        nxaal=1000
        nxbbl=1000
        nyal=1000
        nybb=1000
        nzaal=1000
        nzbb=1000
        xx1=xemit
        yy1=yemit
        zz1=zemit
        nxx=(xx1-perlb(1))/dnxt
        pmi=dnxt*nxx+perl(1)
        if (pmi .eq. xx1) then
            xx0=xx1
        else
            if (xx1 .gt. perl(1)) then
                xx0=dnxt*nxx+perl(1)
            else
                xx0=dnxt*nxx+perl(1)
            endif
        endif
        yy0=yy1
        if (threeedim .eq. 0) then
            zz0=0.0
        else
            nzz=(zz1-perlb(3))/dnzt
            pmi=dnzt*nzz+perl(3)
            if (pmi .eq. zz1) then
                z0=zz1
```
Appendix B. Molecular Dynamics Code

else
  if (zz1.gt.perlb(3)) then
    zz0=dnzt*nzz+perlb(3)
    if (mark2.eq.-1) zz0=zz0+dnzt
  endif
endif
endif

3 if (deppdir(1).eq.0) then
  shift1=1000.0
else
  xx2=xx0+dnxt*mark1
  shift1=(xx2-xx1)/deppdir(1)
endif

yy2=yy0-dnyt
shift2=(yy2-yy1)/deppdir(2)
if (deppdir(3).eq.0) then
  shift1=1000.0
else
  zz2=zz0+dnzt*mark2
  shift3=(zz2-zz1)/deppdir(3)
endif
shift=amin1(shift1,shift2,shift3)

xx3=xx1
yy3=yy1
zz3=zz1
xx1=xx1+shift*deppdir(1)
yy1=yy1+shift*deppdir(2)
zz1=zz1+shift*deppdir(3)
if (shift.eq.shift1) xx0=xx2
if (shift.eq.shift2) yy0=yy2
if (shift.eq.shift3) zz0=zz2

xx4=.5*(xx1+xx3)
yy4=.5*(yy1+yy3)
zz4=.5*(zz1+zz3)
nxx=(xx4-perlb(1))/dnxt
pmi=dnxt*nxx+perlb(1)
if (pmi.eq.xxx) then
  nxxa=nxx
  nxbb=nxx+1
else
  if (xxx.gt.perlb(1)) then
    nxxa=nxx
    nxbb=nxx+2
  else
    nxxa=nxx-1
    nxbb=nxx+1
  endif
endif

nyy=(yy4-perlb(2))/dnyt
nyaa=nyy
nybb=nyy+2
if (nyaa.lt.1) nyaa=1
if (nybb.gt.ntopbox) nybb=ntopbox
if (threedim.eq.0) then
  nzaa=1
  nzbb=1
else
  nzz=(zz4-perlb(3))/dnzt
  pmi=dnzt*nzz+perlb(3)
  if (pmi.eq.zz4) then
    nzaa=nzz
    nzbb=nzz+1
  else
    if (zz4.gt.perlb(3)) then
      nzaa=nzz
    endif
    endif
  endif
endif
Appendix B. Molecular Dynamics Code

```
nzbb=nzz+2
else
nzaa=nzz-1
nzbb=nzz+1
endif
endif
do 4 jl=nzaa,nzbb
nshift1=0
if (jl .gt. nxt) nshift1=(jl-1)/nxt
if (jl .lt. 1) nshift1=-1+jl/nxt
kl=1-nshift1*nxt
do 5 j2=nyaa,nybb
do 6 j3=nzaa,nzbb
if ((jl .ge. nxaa .and. j1 .le. nxbb1) .and. 
  *(j2 .ge. nyaal .and. j2 .le. nybb1) .and. 
  *(j3 .ge. nzaal .and. j3 .le. nzbb)) goto 6
nshift3=0
if (threedim .ne. .0) then
if (j3 .gt. nzt) nshift3=(j3-1)/nzt
if (j3 .lt. 1) nshift3=-1+j3/nzt
endif
k3=j3-nshift3*nzt
if (nbox(kl,j2,k3,nboxnl) .ne. nboxn) goto 6
7
j4=1,nbox(kl,j2,k3,nboxnl)
k4=nbox(kl,j2,k3,j4)
if (itype(k4) .eq. inert) goto 7
xpos=rv(1,k4)+perlen(1)*nshift1
ypos=rv(2,k4)
zpos=rv(3,k4)+perlen(3)*nshift3
disl=xpos-xemit
disz=ypos-yemit
disz=zpos-zemit
disl2=disl**2+disz2**2+dis3**2
s1=disl*depdir(1)+dis2*depdir(2)+dis3*depdir(3)
if (s1 .lt. .0) goto 7
s12=s1**2
RR=dis-s12
if (RR .gt. rcutsq) goto 7
shrun=s1-sqrt(rcutsq-RR)
if (shrun .lt. shortest) shortest=shrun
7 continue
6 continue
5 continue
4 continue
nxaa=nxaa
nxbb=nxbb
nyaal=nyaa
nybb=nybb
nzaa=nzaa
nzbb=nzbb
if (shortest .eq. 1000.0) goto 3
if (more .eq. 0) then
more=1
goto 3
endif
rv(1,natoms)=xemit+shortest*depdir(1)
if (rv(1,natoms) .lt. perlb(1)) rv(1,natoms)=rv(1,natoms)+ 
  *(1.0-int((rv(1,natoms)-perlb(1))/perlen(1)))*perlen(1)
if (rv(1,natoms) .ge. perub(1)) rv(1,natoms)=rv(1,natoms)- 
  int((rv(1,natoms)-perlb(1))/perlen(1)))*perlen(1)
rv(2,natoms)=yemit+shortest*depdir(2)
if (threedim .eq. .0) then
rv(3,natoms)=zemit
else
rv(3,natoms)=zemit+shortest*depdir(3)
endif
rv(3,natoms)=rv(3,natoms)+ 
  *(1.0-int((rv(3,natoms)-perlb(3))/perlen(3)))*perlen(3)
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
if (rv(3,natoms) .ge. perub(3)) rv(3,natoms)=rv(3,natoms)-
  *int((rv(3,natoms)-perlb(3))/perlen(3))*perlen(3)
endif

rv(4,natoms)=velx
rv(5,natoms)=vely
rv(6,natoms)=velz
yest(natoms)=9000.0
do 8 j=1,3
if (iDdtyp .eq. 1) then
  y(j,natoms)=rv(j,natoms)
y(j+3,natoms)=rv(j+3,natoms)
else
  y(j,natoms)=(rv(j,natoms)-perlb(j))/perlen(j)
y(j+3,natoms)=rv(j+3,natoms)/perlen(j)
endif
u1(j,natoms)=dt*y(j+3,natoms)
u2(j,natoms)=.0
u3(j,natoms)=.0
u4(j,natoms)=.0
u5(j,natoms)=.0
8 continue
nw1=int((rv(1,natoms)-perlb(1))/dnxt)+1
nw2=int((rv(2,natoms)-perlb(2))/dnyt)+1
if (nw2 .gt. nboxy) then
  write(6,*), 'nboxy is too small'
  stop
endif
if (threedim .ne. .0) then
  nw3=int((rv(3,natoms)-perlb(3))/dnzt)+1
else
  nw3=1
endif
if (nbox(nw1,nw2,nw3,nboxn) .ne. nboxnei) then
  nbox(nw1,nw2,nw3,nboxn)=nboxnei
  nbox(nw1,nw2,nw3,nboxn1)=0
  if (nw2 .gt. ntboxo .and. _itype(natoms) .ne. inert) stopbox=nw2
endif
  nbox(nw1,nw2,nw3,nboxn1)=nbox(nw1,nw2,nw3,nboxn)+1
  nbox(nw1,nw2,nw3,nbox(nw1,nw2,nw3,nboxn1))=natom
if (iDdtyp .eq. 1 .and. newlst .eq. 0) then
  rold(1,natoms)=rv(1,natoms)
  rold(2,natoms)=rv(2,natoms)
  rold(3,natoms)=rv(3,natoms)
nmindx(natomi)-nnindx(natomi-1)
if (nxt .le. 3) then
  nxaa2=1
  nxbb2=nxt
else
  nxaa2=nw1-1
  nxbb2=nw1+1
endif
if (threedim .ne. .0) then
if (nzt .le. 3) then
  nxaa=1
  nxbb=nzt
else
  nxaa=nw3-1
  nxbb=nw3+1
endif
else
  nxaa=1
  nxbb=1
endif
do 9 j1=nxaa2,nxbb2
  k1=1
if (k1 .lt. 1) k1=k1+nxt
if (k1 .gt. nzt) k1=k1-nzt
do 10 j2=nw2-1,nw2+1
Appendix B. Molecular Dynamics Code

```c++
if (j2 .lt. 1 .or. j2 .gt. ntopbox) goto 10
do 11 j3=n2aa,n2bb
11 k3=j3
if (k3 .lt. 1) k3=k3+nzt
if (k3 .ne. 1 .and. k3 .gt. nzt) k3=k3-nzt
if (nbox(k1,j2,k3,nboxn).ne. nboxnei) goto 11
do 12 j4=1,nbox(k1,j2,k3,nboxn)
j=nbox(k1,j2,k3,j4)
if (j .eq. natoms) goto 12
if (itype(j) .eq. inert) goto 12
disi=rold(i,natoms)-rold(1,j)
disci=disi-perlen(i)*nint(disi/perlen(1))
2 disi=rold(1,natoms)-rold(i,j)
disci=disi-perlen(1)*nint(disi/perlen(1))
3 disi=rold(2,natoms)-rold(1,j)
disci=disi-perlen(1)*nint(disi/perlen(1))
4 disi=rold(3,natoms)-rold(3,j)
disci=disi-perlen(3)*nint(disi/perlen(3))
5 disi=disi**2+disi**2+disi**2
if (disi .gt. rctsgn) goto 12
nnindx(natoms)=nnindx(natoms)+1
nnindx(nnindx(natoms))=j
12 continue
11 continue
10 continue
9 continue
end
endif
2 continue
ndegfr=natoms
if (ibdtype .eq. 2) then
newlst=1
ndegfr=ndegfr+1
endif
natadd=0
if (newlst .ne. 0) natadd=-1
return
end
```

chkvap

```fortran
subroutine chkvap
include 'vap.inc'
common /nm3rold/ rold(3,natmax),
common /u1/ u1(3,natmax),u2(3,natmax),u3(3,natmax),
* u4(3,natmax),u5(3,natmax)
dimension njud(natmax)
do 51 i=1,natoms
njud(i)=0
51 continue
ni=0
n2=0
if (ibdtype .eq. 1) yline=5.0
if (ibdtype .eq. 2) yline=5.0/y2,ndegfr
do 1 i=1,natoms
if (itype(i) .ne. inert) then
1 ul(2,i) = ul(2,i) +u1(2,i)
if (y(2,i)-yline) yline+=yline+1
endif
endif
if (itype(i) .eq. inert .and. ((rv(5,i) .lt. 0
* .and. rho(i) .eq. 0 .or. f(2,i) .lt. 0)) then
if (rho(i) .eq. 0) then
emu=emu+1
else
emu=emu+1
endif
endif
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
\texttt{ni=ni+1} \\
\texttt{n2=n2+nnindx(i)-nnindx(i-1)} \\
\texttt{njud(i)=1} \\
\texttt{goto 1} \\
\texttt{endif} \\
\texttt{if (ni .eq. 0) goto 1} \\
\texttt{do 2 j=nnindx(i-1)+1,nnindx(i)} \\
\texttt{kk=nnlst(j)} \\
\texttt{if (njud(kk) .gt. 0) then} \\
\texttt{n2=n2+1} \\
\texttt{goto 2} \\
\texttt{endif} \\
\texttt{nnlst(j-n2)=kk-n1} \\
\texttt{2 continue} \\
\texttt{do 52 i=1,3} \\
\texttt{u1(i,i-n1)=u1(1,i)} \\
\texttt{u2(i,i-n1)=u2(1,i)} \\
\texttt{u3(i,i-n1)=u3(1,i)} \\
\texttt{u4(i,i-n1)=u4(1,i)} \\
\texttt{u5(i,i-n1)=u5(1,i)} \\
\texttt{52 continue} \\
\texttt{do 3 i=1,6} \\
\texttt{y(1,i-n1)=y(1,i)} \\
\texttt{3 continue} \\
\texttt{yest(i-n1)=yest(i)} \\
\texttt{if (modefx(ipntfx(i)) .eq. 4) then} \\
\texttt{workfx(i-n1)=workfx(i)} \\
\texttt{posfx(1,i-n1)=posfx(1,i)} \\
\texttt{posfx(2,i-n1)=posfx(2,i)} \\
\texttt{posfx(3,i-n1)=posfx(3,i)} \\
\texttt{if (ni .ne. 0) then} \\
\texttt{workfx(i)=0} \\
\texttt{posfx(1,i)=0} \\
\texttt{posfx(2,i)=0} \\
\texttt{posfx(3,i)=0} \\
\texttt{endif} \\
\texttt{if (follow) then} \\
\texttt{ipntrg(i-n1)=ipntrg(i)} \\
\texttt{if (ni .ne. 0) ipntrg(i)=0} \\
\texttt{endif} \\
\texttt{ipntfx(i-n1)=ipntfx(i)} \\
\texttt{if (ni .ne. 0) ipntfx(i)=0} \\
\texttt{rold(1,i-n1)=rold(1,i)} \\
\texttt{rold(2,i-n1)=rold(2,i)} \\
\texttt{rold(3,i-n1)=rold(3,i)} \\
\texttt{itype(i-n1)=itype(i)} \\
\texttt{nnindx(i-n1)=nnindx(i)-n2} \\
\texttt{1 continue} \\
\texttt{if (ni .eq. 0) return} \\
\texttt{natoms=natoms-nl} \\
\texttt{if (ibdtype.eq.1) ndegfr=natoms} \\
\texttt{if (ibdtype.eq.2) then} \\
\texttt{ndegfr=natoms+1} \\
\texttt{do 6 i=1,6} \\
\texttt{y(i,ndegfr)=y(i,ndegfr+nl)} \\
\texttt{6 continue} \\
\texttt{endif} \\
\texttt{return} \\
\texttt{end}
APPENDIX C. MULTILAYER RESISTIVITY

The resistivity of vapor deposited multilayers is expected to be higher than that of perfectly bonded, flat slabs of the various constituent layers because of electron scattering at rough surfaces, by vacancies, voids, dislocations, and other stacking defects in each layer, and by electron-impurity scattering resulting from interdiffusion between the layers. In the Ta/Cu bilayer system studied here, the resistivity of Ta is much greater than that of Cu (180 $\mu\Omega$cm and 3 $\mu\Omega$cm, respectively). In the case of our in-plane resistance measurements, the flow of current can be assumed to be only in the top Cu layer considering the big differences between their resistivities ($\rho_{Cu}:\rho_{Ta}=1:60$) and thicknesses ($d_{Cu}:d_{Ta}=6:1$). The changes in resistivity measured experimentally are therefore dominated by changes in electron scattering in the Cu layer. Matthiessen’s rule then provides a simple means for combining the many electron-scattering mechanisms in the film to estimate its electrical resistivity [252]. The resistivity, $\rho_T$, of a thin film in which electrons are scattered within the bulk lattice and at surface obeys this rule [252]: $\rho_T=\rho_s+\rho_b$, where $\rho_s$ is the resistivity induced by electron-surface scattering and $\rho_b$ is the bulk (perfect lattice) resistivity. In vapor deposited multilayers, other scattering mechanisms contribute to the resistivity and in this case the resistivity can be approximated as $\rho_T=\rho_s+\rho_b+\rho_{int}+\rho_{def}$, where $\rho_{def}$ is the resistivity induced by lattice defects (e.g., vacancies) and $\rho_{int}$ is the resistivity induced by imperfections at the interface. If it is assumed that the perfect bulk lattice resistivity is constant in the experiments conducted here, then $\rho_b$ can be ignored in consideration of resistivity changes. It is then
possible to define a structure-dependent function, $U$, to describe resistivity changes in vapor deposited films,

$$U = \rho_S + \rho_{\text{mt}} + \rho_{\text{def}}.$$  

(C1)

The Fuchs-Schodheimer model can be used to describe the scattering of electrons at the surface of a thin film. In this model, a specularity coefficient, $p$, is used to describe a fraction of the electrons that are scattered elastically at the solid surface. The resistivity ratio of film, $\rho_f$, to that of the bulk metal, $\rho_0$, is then described by [236, 240]:

$$\frac{\rho_f}{\rho_0} = \frac{\Phi_p(k)}{k},$$  

(C2)

where

$$\frac{1}{\Phi_p(k)} = \frac{1}{k} - \frac{3}{2k^2} (1-p) \int \left( \frac{1}{t^2} - \frac{1}{t^2} \right) \frac{1-e^{-kt}}{1-p e^{-kt}} dt,$$  

(C3)

and $k$ is the ratio of film thickness to the bulk solid electron mean free path. Here $p$ is a function of the electron incident angle relative to the surface normal, $\theta$ [253]:

$$p = \exp[-(4\pi\gamma)^2 u^2],$$  

(C4)

where $\gamma = h/\lambda_e$, $h$ is the surface roughness, $\lambda_e$ is the Fermi wavelength, and $u = \cos \theta$. In the case of large $k$, Eq. A2 has the approximate form:

$$\frac{\rho_f}{\rho_0} = \int du \left( 1 + \frac{3}{8k} (1-p) \right).$$  

(C5)

The surface induced resistivity can then be written:

$$\rho_s = \int du \left( \frac{3}{8k} \rho_0 (1-p) \right).$$  

(C6)
Appendix C. Multilayer Resistivity

The resistivity induced by scattering of defects (vacancies and impurities) within the film can be expressed as [254]:

$$\rho_{\text{def}} = \frac{n_0 \tau_0}{\sigma_0 n(d) \tau(d)}, \quad (C7)$$

where $\sigma_0 = n_0 e^2 \nu_0 \frac{m^*}{m}$ ($m^*$ is the carrier effective mass) is the Drude conductivity, $n(d)$ is the average density of electrons, and $\frac{1}{\tau(d)} = 2\pi U_0^2 n_i g(\varepsilon_F) m$ ($U_0$ is the strength of impurity potential, $g(\varepsilon_F)$ is the density of state, $n_i$ is the defect density and $m$ is the coefficient related to density function). Therefore, Eq. (A6) becomes:

$$\rho_{\text{def}} = \frac{2\pi U_0^2 n_i g(\varepsilon_F) mm^*}{n(d) e^2}. \quad (C8)$$

To account for interdiffusional effects on the resistivity near an interface between dissimilar materials, Skomski et al. [239] have proposed a phenomenological expression in the thick film limit as:

$$\rho_{\text{int}} = \frac{2\delta \rho_M}{\Delta} \tanh \left( \frac{\Delta}{2\delta} \right), \quad (C9)$$

where $\Delta$ is a film thickness related parameter and $\rho_M$ is the resistivity of an intermixed interfacial region of width $2\delta$ that exists between films a and b. The structure function, $U$, can then be expressed as a combination of Eqs. (A6), (A8), and (A9):

$$U = \int_{u=0}^{1} \left( \frac{3\rho_0}{8k} (1 - p) \right) + \frac{2\pi U_0^2 n_i g(\varepsilon_F) mm^*}{n(d) e^2} + \frac{2\delta \rho_M}{\Delta} \tanh \left( \frac{\Delta}{2\delta} \right). \quad (C10)$$

If the film resistivity is assumed to be only a function of the surface root mean square roughness, $h$, vacancy concentration, $n_i$, and interdiffusion width, $\delta$, and all other
Appendix C. Multilayer Resistivity

contributions remain constant, then the change in resistivity of a bilayer thin film (consisting of a good conductor on top of a poor one) can be approximated by:

\[ U = A \left( 1 - \frac{\lambda_e \text{Erf} \left( \frac{4 \pi h}{\lambda_e} \right)}{8 \sqrt{\pi} h} \right) + Bn_i + C \delta \tanh \left( \frac{\Delta}{2 \delta} \right) \]

(C11)

where A, B, and C are constants to be fitted and \( \text{Erf} \) is an error function.
BIBLIOGRAPHY

Bibliography

Bibliography

Bibliography

Bibliography

Bibliography

Bibliography

Bibliography