

AFRL-ML-WP-TR-1999-4028

**THIN FILM GROWTH SIMULATION USING  
CELLULAR AUTOMATA STATE SPACE,  
AND NEURAL NETS METHODS**



A.G. Jackson, M. Benedict

AvXm Partnership  
5375 Oakvista Place  
Kettering, OH 45440-2634

February 1999

Final Report for Period 21 April 1998 - 01 October 1998

Approved for Public Release; Distribution is Unlimited.

19990507 042

Materials & Manufacturing Directorate  
Air Force Research Laboratory  
Air Force Materiel Command  
Wright-Patterson Air Force Base, Ohio 45433-7734

## NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

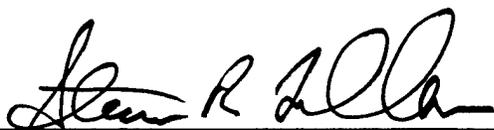
This report has been reviewed by the Office of Public Affairs (ASC/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.



---

CAPT DAVID CONRAD, USAF  
Project Engineer  
Materials Process Design Branch  
Manufacturing Technology Division



---

DR. STEVEN R. LeCLAIR  
Chief  
Materials Process Design Branch  
Manufacturing Technology Division

“If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFRL/MLMR, Bldg. 653., 2977 P St., Suite 6, W-PAFB, OH 45433-7739 to help us maintain a current mailing list.”

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

# REPORT DOCUMENTATION PAGE

FORM APPROVED  
OMB NO. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)		2. REPORT DATE February 1999	3. REPORT TYPE AND DATES COVERED Final 04/21/98 - 10/01/98
4. TITLE AND SUBTITLE Thin Film Growth Simulation Using Cellular Automata State Space, and Neural Nets Methods		5. FUNDING NUMBERS C F33615-98-C-5138 PE 65502F PR 3005 TA 05 WU AQ	
6. AUTHOR(S) A.G. Jackson, M. Benedict		7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AvXm Partnership 3575 Oakvista Place Kettering, OH 45440-2634	
8. PERFORMING ORGANIZATION REPORT NUMBER AvXm-99-1		9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES) Materials & Manufacturing Directorate Air Force Research Laboratory, Air Force Materiel Command Wright-Patterson AFB, OH 45433-7734 POC: CAPT David Conrad USAF, AFRL/MLMR, (937) 255-8786	
10. SPONSORING/MONITORING AGENCY REP NUMBER AFRL-ML-WP-TR-1999-4028		11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for Public Release; Distribution is Unlimited.		12b. DISTRIBUTION CODE	
13. ABSTRACT The objective of this research was to demonstrate simulation of multi-species thin film growth that is fast, displays on a desktop computer, and utilizes a unique combination of cellular automata, state space, and neural nets. Software to accomplish this objective was refined by developing computer code that implements a physical model of film growth. Neural nets were integrated into the software to reduce the time needed for computing the atom positions, and generation of exemplars needed for training and testing the neural nets was accomplished via visual basic code. Display of the simulation using a desktop computer (Macintosh Power PC) was achieved. Functionalities for live rotation and modification of the film are included, as well as the capability to record the simulation as a computer movie. Study of vacancy behavior using the simulation of about 30, believed to be a result of scaling issues in constraints used in the model. This software can be used to simulate film growth for molecular beam epitaxy and pulsed laser deposition processes. Physical and chemical model studies and process refinement studies can be accomplished using this software.			
14. SUBJECT TERMS Thin Films, Cellular Automata, Neural Nets, Simulations, Atomic Scale.		15. NUMBER OF PAGES 34	
16. PRICE CODE		17. SECURITY CLASSIFICATION OF REPORT Unclassified	
18. SECURITY CLASS OF THIS PAGE Unclassified		19. SECURITY CLASS OF ABSTRACT Unclassified	
20. LIMITATION ABSTRACT SAR			

Standard Form 298 (Rev 2-89)  
Prescribed by ANSI Std Z39-18  
298-102

## Table of Contents

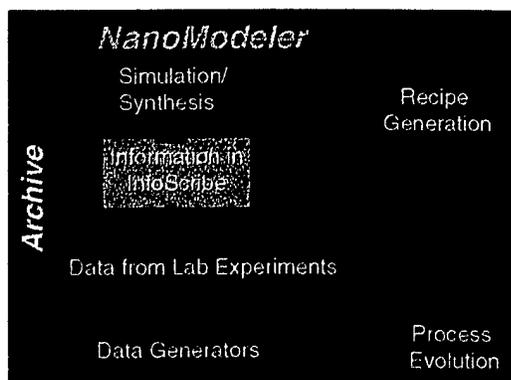
<b>1. Introduction</b>	<b>1</b>
<b>2. Growth Model of Thin Films</b>	<b>2</b>
<b>2.1. Mechanisms of Growth</b>	<b>2</b>
<b>3. Phase I Technical Objectives</b>	<b>3</b>
<b>4. NanoModeler Description</b>	<b>3</b>
<b>4.1. Physical Model Basis</b>	<b>4</b>
<b>4.2. Neighborhoods</b>	<b>4</b>
<b>4.3. Temperature</b>	<b>5</b>
<b>4.4. Adsorbed State</b>	<b>6</b>
<b>4.5 Surface Diffusion</b>	<b>6</b>
<b>5. Neural Nets</b>	<b>7</b>
<b>5.1. Generation of Exemplars for Neural Net Training</b>	<b>7</b>
<b>5.2. Arrhenius Equations and the Metropolis Algorithm</b>	<b>8</b>
<b>5.3. State Change Probabilities</b>	<b>10</b>
<b>6. Implementation of the Model</b>	<b>12</b>
<b>7. Results</b>	<b>12</b>
<b>7.1. Simulations</b>	<b>12</b>
7.1.1. Effect of flux ratio on number of vacancies	12
7.1.2. Time dependence of site occupancies	16
7.1.3. Substrate variations	16
7.1.4. Line Defect	18
7.1.5. Comparison of substrate interface to film with and without diffusion module active.	19
<b>8. Summary</b>	<b>23</b>
<b>9. Acknowledgments</b>	<b>24</b>
<b>10. References</b>	<b>24</b>
<b>Appendix</b>	<b>25</b>

## 1. Introduction

Simulation of thin film growth is an essential aspect of developing new materials engineered to specific properties as well as defining the processing sequence required to produce the film. Software is available that will simulate such growth, but the computer platforms are in the high end of cost and performance, and the programs are designed to be used by experts in thin film growth who are interested in studying the mechanisms of nucleation and growth. Identifying the processing parameters and the control parameters suited to producing the films is not a focus of these computer programs and applications.

In addition the computational tractiveness of such approaches is bound up in the first principles, molecular dynamics, monte carlo, and related methods. While these methods are powerful and essential, the time required to accomplish simulation of real films can be very long, extending from days to months depending on the complexity of the film.

Thus, the first issue is reducing the computational burden in simulation. The second issue is ease of use and suitability of display of the simulation in a timely manner for the user. In the following we present results of research effort directed toward these two issues.



Users who must design materials or processes for manufacturing the materials require data, properties, time constraints, cost boundaries, environmental data and constraints, safety issues, and health issues. The software described in this report, the NanoModeler™, addresses the first three explicitly, while the remainder can be dealt with from the decisions based on the simulations and verification experiments.

*Figure 1. Flow of data and information through the NanoModeler.*

Properties are decided upon by the end use of the material, whether this is a device, exploratory design, or process modification or new design. They drive the simulation by providing a user with the benchmarks for determining if the predicted film is close to the required goals.

Data is essential in this enterprise. The type of data required consists of two types, static and dynamic (time series). Static data is available on the properties of materials in handbooks, a few CDs, the Internet, the literature, and directly from researchers. This poses a number of operational problems when the data is dynamic, because storing such data in an easily accessible form can be troublesome. NanoModeler attacks this problem by using the InfoScribe Technologies InfoScribe™ data acquisition and archiving software. The flow of information/data is illustrated in Figure 1, which shows the NanoModeler system in terms of the flow of data from archive to simulation to 'recipe',

from which a process can be improved, leading to new data and new experiments. The loop is completed by feeding the data into the InfoScribe™ archive. Although not accomplished during this research, the file formats designed into Nanomodeler are suited to use by Infoscribe interfaces.

## **2. Growth Model of Thin Films**

Attachment of atoms to a surface at the atomic scale is complex, involving a number of interactions not easily described mathematically or physically. For our purposes, nucleation of films on substrates can be thought of in terms of a series of random, single interactions followed by competitive growth into grains and microstructures. In this model, the arrival of atoms at a surface interact locally with other atoms in the immediate neighborhood. These local interactions simplify conceptually the sequence that produces an island, step, ledge, or layer of film.

Mathematical models of film nucleation and growth have been developed that are based on quantum mechanics. These require details of interaction potentials, geometrical parameters such as bond distances and angles, and numerical estimates of constants used in the equations describing the interactions. Such models are very desirable, because they allow a richness of detail that provides for a wide range of situations found in film growth. The downside is that many of the needed parameters are not known and no means for rational estimates are available. In addition, the computational complexity is very high, requiring large computational facilities to handle realistic numbers of atoms.

Molecular dynamics approaches are better computationally, but they also require methods for rapid solutions of large sets of simultaneous equations, as well as estimates for parameters.

Stochastic approaches fare better when the interactions are confined to small numbers of atoms or molecules. Thus, monte carlo computations, while tedious, are excellent means for simulating atomic behavior. The difficulty here lies in applying MC methods to an array of atoms larger than about 50,000.

### **2.1. Mechanisms of Growth**

Qualitatively, what occurs when a 'bare' substrate is exposed to a flux of atoms? Some assumptions are needed here to bound the problem. We assume that the substrate is atomically flat and perfect crystallographically. The atoms in the flux are the same species as the substrate, and the flux is thermal, that is, no kinetic energy larger than thermal is permitted.

The flux has a flux rate of  $N$  atoms/cm<sup>2</sup>-sec.

Molecular Beam Epitaxy (MBE) process produces a continuous flux of atoms, while Pulsed Laser Deposition (PLD) process produces a burst of atoms in microseconds followed by a few tens to hundreds of milliseconds pause until the next burst. These two

processes represent widely different film growth situations. In MBE, for realistic fluxes that produce growth rates of a few tenths of nanometers/second (roughly one layer per second), the arrival rate at the atomic scale is a gentle rain of atoms which arrive at random locations on the substrate and with velocities that are not high enough to damage the surface by ejecting substrate atoms. For PLD, on the other hand, the arrival rate is orders of magnitude higher than in MBE. Equivalent flux to produce 1 layer/sec rate is about 100 times larger for PLD than for MBE. In terms of models, MBE fluxes allow the time interval to be about 100 microseconds for atoms arriving at a surface in order to accommodate the number per  $\text{nm}^2$ . In PLD, all the atoms arrive in a few microseconds, followed by relaxation of this 'pile' of atoms to pseudo-equilibrium during the pause between pulses. In MBE neighborhood interactions are most important, while in PLD neighborhood interactions and surface diffusion, as well as bulk diffusion play a major role.

As an atom approaches the substrate the interaction region becomes more well defined relative to the atoms in the substrate surface. When the atom lands on the surface, it feels the effects of the atoms in the immediate neighborhood, that is the nearest and next nearest neighbors. The initial effect is to adsorb the atom to the neighborhood with weak van der Waals type bonding. This is physisorption. Depending on the atoms in the neighborhood, chemisorption (strong bonding) may occur, surface diffusion may take place, or the atom may evaporate. At its simplest these interactions are the starting situations possible. Hence, an algorithm to simulate this behavior must allow for this simple case and also the more complex behavior when defects and dopants are present. And, the effects of temperature must be built into the algorithm.

Quantitative models are available for simple cases, see for example Grimes et al. work on inert gas adsorption onto calcides. [2] Such studies are important to the modeling methods because they define the limits of various concepts, for example, the parameters used for various kinds of interaction potentials. Because such systems are chosen to test theories, they are not realistic in that engineering materials are not of interest (not tractable). The challenge, therefore, is to create a model that utilizes the concepts tested in the theoretical models and that allows for real engineering materials to be considered as well as the process used to create them.

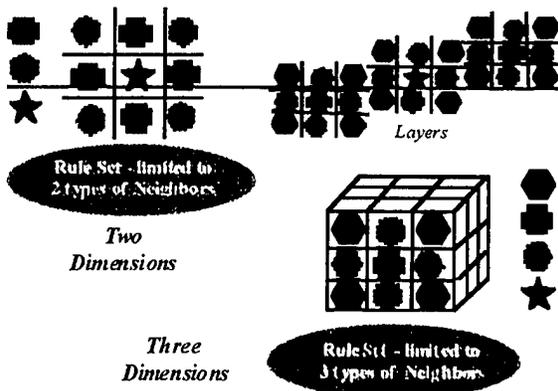
### **3. Phase I Technical Objectives**

The technical objective of this research was to demonstrate a simulation of simple multi-species thin film growth that is fast, displays on a desktop computer, and utilizes a unique combination of cellular automata, state space, and self-architecting neural nets, together with a user interface easy to apply.

### **4. NanoModeler Description**

The essential pieces of the software consist of (1) the physical model assumptions, in the context of cellular automata, and the parameters, equations, and states applied, (2) the

## Cellular Automaton



## 4.2. Neighborhoods

The operation of the system is similar to a standard cellular automaton, except that the rules governing the interactions of a neighborhood may vary from cell to cell. The relationship of the central atom to a neighbor cell is defined in terms of a transformation of the attributes of the central atom to those of the neighbor cell under a given set of rules. Such a transformation allows a wide range of

*Figure 2. Basic 2D and 3D Cellular Automata.*

possible changes, including quantitative, functional, geometric, and qualitative (Figure 3).

## Neighborhoods

*Interactions and Relations, Neighborhood Space*

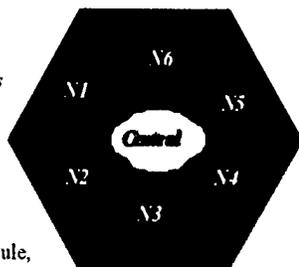
*Attributes of Neighborhoods*

*State:*

Occupied  
Unoccupied

*Properties:*

Name: Atom, Molecule,  
Spatial: Coordinates,  
Temporal: Position (time, energy),  
Energy: Bond Type,



This disentanglement of the neighborhood from the geometry generalizes the neighborhood idea to that of a neighborhood space in which each point contains a set of attributes that define the neighborhood. The interactions and relationships that may exist between a neighborhood and the central object being considered become

*Figure 3. Neighborhoods.*

the focus of attention, rather than the chemical and physical model equations that define the quantitative aspects of the system. This means that we can consider qualitative issues separately from numerical ones. And in point of fact when a model is established there are a series of qualitative decisions that are made before any numerical data are applied to equations. These decisions are part of the algorithm for deciding on which numerical data to use and the relevant equations. By generalizing the neighborhood concept to include this decision we allow a broader control of the model than otherwise possible, and we now can include the geometry as part of the attribute set. Put simply, we can make decisions about behavior without knowing specific numeric values. This is especially important when experimental or theoretical data are not available, either from the literature or from computations.

We are also free to decide the number of neighborhoods to use. We can start with a single neighborhood surrounding the central object. This would, for example, correspond to a uniform rule set. If there are several sets of rules or interactions or functional relationships or logical relationships, then the number of neighborhoods would increase accordingly, the exact number depending on the split of the attributes desired. The neighborhoods can coincide with the geometric neighborhoods if desired, but this is not required. The hierarchy of decisions starts with information about the objects involved, for example, names, states, background effects directly associated with the objects. Next specific information is used such as charge size, radius, properties (conductivities, electron configurations, valence state), crystal info (type, space group symbol, Wyckoff site

separately from numerical ones. And in point of fact when a model is established there are a series of qualitative decisions that are made before any numerical data are applied to equations. These decisions are part of the algorithm for deciding on which numerical data to use and the relevant equations. By generalizing the neighborhood concept to include this decision we allow a broader control of the model than otherwise possible, and we now can include the geometry as part of the attribute set. Put simply, we can make decisions about behavior without knowing specific numeric values. This is especially important when experimental or theoretical data are not available, either from the literature or from computations.

We are also free to decide the number of neighborhoods to use. We can start with a single neighborhood surrounding the central object. This would, for example, correspond to a uniform rule set. If there are several sets of rules or interactions or functional relationships or logical relationships, then the number of neighborhoods would increase accordingly, the exact number depending on the split of the attributes desired. The neighborhoods can coincide with the geometric neighborhoods if desired, but this is not required. The hierarchy of decisions starts with information about the objects involved, for example, names, states, background effects directly associated with the objects. Next specific information is used such as charge size, radius, properties (conductivities, electron configurations, valence state), crystal info (type, space group symbol, Wyckoff site symbols). Finally, the geometric information is applied, such as separation distances, coordinates, bond distance and angle, coordination. With such a hierarchy, the task is to identify the possible decisions that can be made using the information at each level in order to determine what path to numeric data must be taken.

For the simulation of thin film growth, we have assumed a simple cubic arrangement of neighborhoods in which the vapor, surface and substrate layers comprise the cube. The lowest is the substrate layer, containing 9 neighbors. The surface layer is next, containing 8 neighbors. This is the layer containing the central atom. The vapor layer is last with 9 neighbors. The neighbors are numbered starting with the substrate layer and continuing through the vapor layer. The numbering is used in the algorithms as labels to track the attributes of each neighbor. Vacancies are permitted in each layer, and binary species are permitted. Three or more species are not yet implemented, but expanding the species range poses no fundamental problem.

There are four states possible currently: AB bonded: no change in coordinates, only a change in state; AA bonded: only a change in state; evap: the atom has evaporated from the surface, changing the coordinates and the state; adsorbed: on the surface, able to interact and transform its attributes.

### **4.3. Temperature**

The temperature dependence of state changes is assumed to be Arrhenius type. Thus, there are activation energies for each possible type of state. Choice of the values for the energies is made based on any literature values that are available.

#### 4.4. Adsorbed State

Once an atom is adsorbed on the substrate surface, it is free to move on the surface and change its state to bonded, evaporated (vapor), or remain in the adsorbed state but in a new neighborhood. The reference atom is always symbolized by "A".

Each case represents an energy barrier with an associated energy that must be overcome in order to change state to a new state.

#### 4.5 Surface Diffusion

Sutton and Balluffi [2] note that the relative order of diffusion coefficients is Surface > grain > bulk > liquid. The type of bonding affects the details of diffusion. Metallic and covalent bonding effects are similar, and ionic bonding is different because it is dominated by charge effects. For example, the surface energy per bond-area for FCC is given by

$$E_s = -2(2h+1)e_b/(h^2 + k^2 + l^2)a^2$$

where, h,k,l = plane indices, a = lattice parameter, and  $e_b$  = energy/bond = (NZ/2) (Z= coordination number), which is a function of the cohesive energy per bond and the plane indices, whereas for ionic bonds, the expression is

$$E_s = aY/(4\pi^2),$$

where a = lattice parameter and Y = Young's modulus. This expression is independent of direction or surface plane indices. Hence, to apply the correct expression, the bond type must first be determined, and then the orientation if metallic or covalent. The lowest energy orientations are favored, because this results in the most stable configuration.

Vacancies in substrate and surface are permitted in the above algorithms. Diffusion of an atom on the surface of the substrate can now lead to filling vacancies or evaporation.

Long range diffusion on the surface is another problem. Surface diffusion occurs rapidly compared to bulk diffusion, because there are fewer barriers to overcome. So in the same time a bulk atom diffuses a few sites, a surface atom may diffuse many sites. How far in a given time period an atom diffuses is temperature dependent.

Now the first order approximation of diffusion distance is just  $(Dt)^{1/2}$  where D = the diffusion coefficient, t = time. Hence, the time to diffuse a distance x is just  $x^2/D$ . Thermal vibrations are on the order of  $10^{-12}$  to  $10^{-14}$  seconds for comparison. Ga diffusion on GaAs has a diffusion coefficient of about  $D = 10^{-12}$  cm<sup>2</sup>/sec, according to literature. Converting to nanometers, we arrive at the following. For diffusion of about a lattice parameter (about .4 nm), the time if  $D = 10^{-12}$  cm<sup>2</sup>/sec, is about 2 millisecs. Some values for D are larger, giving times of a few 10s of microseconds. For the simulations in which we allow

atoms to move about 1 nm or less the time frame is therefore about  $10^{-5}$  to  $10^{-3}$  seconds per step, far longer than thermal vibrations.

## **5. Neural Nets**

The rule sets for a realistic model will be very complex when qualitative-quantitative rule sets are created. The possible interactions among the atoms are many, necessitating a large number of logical constructions that are expensive computationally. The presence of many nested If-Then rules together with logical AND, OR, and NOT operations raises a concern about real time display of growth simulations. For simple models, the speed of desktops is fast enough to make this point moot, but as the model includes more and more possible interactions, the logical nesting becomes large, slowing down the real-time display options. Coupling to this many calls to exponential functions and other quantitative functions that may be added will result in unacceptable slowing of the engine.

Use of neural nets offers the possibility of reducing this time problem by trading the on-line computation with off-line training. The live-time computation only involves the prediction computation of the net, which is fixed once the net has trained. Thus, no matter how complex the rules may become, the neural net always has about the same time requirement for prediction. The issue of prediction accuracy is a separate one.

To use a neural net requires that a large number of examples (the exemplars) be available in order for a set of net weights to be created that accurately predicts the behavior of the model. As mentioned above this is done off line by code that insures that a reasonable set of possible input conditions is included in the exemplars. Several problems arise at this stage that affect performance of the model. The inputs needed for the neural net must include those parameters that directly influence the final state of the atom at the substrate. The output was restricted to only the state of the atom, not the position or a new position. The parameters that were found to be most useful were the occupancies of the neighborhood atoms, the temperature, probabilities of bonding, evaporation, adsorption, and the random number used in the probability algorithm. With the inputs and outputs decided, generation of the exemplars was accomplished as described next.

### **5.1. Generation of Exemplars for Neural Net Training**

The essential features of the model were included in a Visual Basic macro that generated a spreadsheet containing the probabilities for bonding, vaporization, and adsorption, and the output state. The random probability used the random generator in Excel, which, while not the best random generator, served the purpose for the pattern generation.

The decision about the state of the central atom is made based on the physical-chemical properties assumed for the systems considered. The states are represented in terms of barriers that the atom must overcome. Thus, AB bonding is represented by a large energy barrier that represents the deep well associated with this bond. The AA bond is the next deepest well, followed by adsorption and then evaporation. This ranking is physical since the evaporation barrier is the weakest bond when referenced to an adsorbed atom, and the

two bond types are deeper wells relative to the adsorbed state. Put differently, the amount of energy required to break the bonds and change states is in the ranking  $AB > AA > \text{Adsorbed} > \text{Evap}$ .

## 5.2. Arrhenius Equations and the Metropolis Algorithm

The surface of the substrate represents a collection of sites at which a phase transformation in the vapor occurs when it comes in contact with the site. Hence, we are interested in changes in state at each site. The scale for the sites is atomic scale, i.e., we deal with atom-atom interactions initially. Changes of state in terms of the atom-atom interactions represent a free state of the atom in the vapor changing to some class of bound state at the surface. There may be a number of bound states because of the richness of possible interactions at the surface. For the simplest case, we consider only physisorption (weak bonding to the surface), chemisorption (strong bonding to the surface), adsorption (weak bonding to the surface that allows movement on the surface), and evaporation (breaking of surface bonds).

A barrier model for interactions can be built on the principles of energy requirements to bond or break bonds between an atom and another or a group of atoms. Such interactions are often represented in terms of rate constants in the form of Arrhenius equations:

$$\text{rate} = \text{constant} * \exp[-\text{energy}/kT].$$

Such rate expressions can be found in any chemistry text. Usage of this equation, however, extends beyond simple rate considerations, and it is used broadly to represent the barrier that is present in any change of state. The constant in the equation, called the frequency factor, represents the rate when no energy barrier is present. The Arrhenius form has been demonstrated to be a valid representation in numerous experiments and is widely accepted.

The Metropolis algorithm [3] is useful here, because it considers changes in terms of probabilities in a way that is similar to a barrier energy model.

The probability that a site will change its state from  $i$  to  $j$  is a transition probability,  $S_{ij}$ , and the probability that the state is in state  $i$  is  $p_i$ . The assumption that is made is that the product of the transition probability with the state probability is conserved, that is,

$$S_{ij}p_i = S_{ji}p_j.$$

If the barrier energy of the states can be represented as

$$\exp[-(E_j - E_i)/kT],$$

where  $E_j - E_i$  represents the barrier energy (going from  $i$  to  $j$ ), then the Metropolis method states that the ratio of probabilities is just this function:

$$S_{ij}/S_{ji} = p_j/p_i = \exp[-(E_j - E_i)/kT].$$

This equality implies that the transition probabilities are of the form

$$S_{ij} = \begin{cases} 1 & \text{if } E_j - E_i < 0 \\ \exp[-(E_j - E_i)/kT] & \text{if } E_j - E_i > 0. \end{cases}$$

The state change is determined by the condition that state  $i$  goes to state  $j$  if  $p_j$  is some arbitrary fraction of  $p_i$ . This rule is implemented by choosing a random number in  $[0,1]$  and multiplying it by  $p_i$  to set  $p_j$ . If  $p_j < \exp[-(E_j - E_i)/kT]$ , then the state changes to  $j$ .

This approach has been applied to the possible state changes at the surface when an atom comes into contact with a site on the surface. A collection of possible site configurations has been considered that contains those configurations with the highest potential for inducing a state change. Each configuration has a barrier energy associated with it and therefore it has a probability as well.

To create a rule that considers these configuration probabilities, we consider all the probabilities, sum them, and then normalize each one to the sum, producing a set of probabilities in the interval  $[0,1]$ . Next we create a set of probabilities derived from these that divides the  $[0,1]$  interval into sections. This is done by taking the first probability and marking its value as the first boundary above 0. The next boundary is generated by adding this first boundary value to one of the remaining probabilities. This is continued until all the probabilities are exhausted. The resulting set of regions represents probabilities for state changes corresponding to the barriers in the model. The algorithm for determining the state change is to choose a random number in  $[0,1]$  and compare it with the diagram just created. Depending on the value of the random number, a given state change takes place. The range in each region is determined by the barrier energy, e.g., a strong bond will cover a wide range in the diagram and so be more probable than a weak bond. In addition these boundaries will be temperature dependent through the exponential function.

This method is exemplified in the following example. Suppose there are three possible states with probabilities  $p_1$ ,  $p_2$ ,  $p_3$  calculated from barrier energies  $E_1$ ,  $E_2$ ,  $E_3$ . The conversion to the probability diagram is accomplished by the following:

$$p_n = p_1 + p_2 + p_3;$$

$$p'_1 = p_1/p_n, \text{ [first boundary]}$$

$$p'_2 = p_2/p_n + p'_1, \text{ [second boundary]}$$

$$p'_3 = p_3/p_n + p'_2. \text{ [third boundary; in this case = 1]}$$

Thus, an atom on the surface will change its state to configuration  $p_2$  if  $p'_1 < p_{\text{rand}} < p'_2$ .

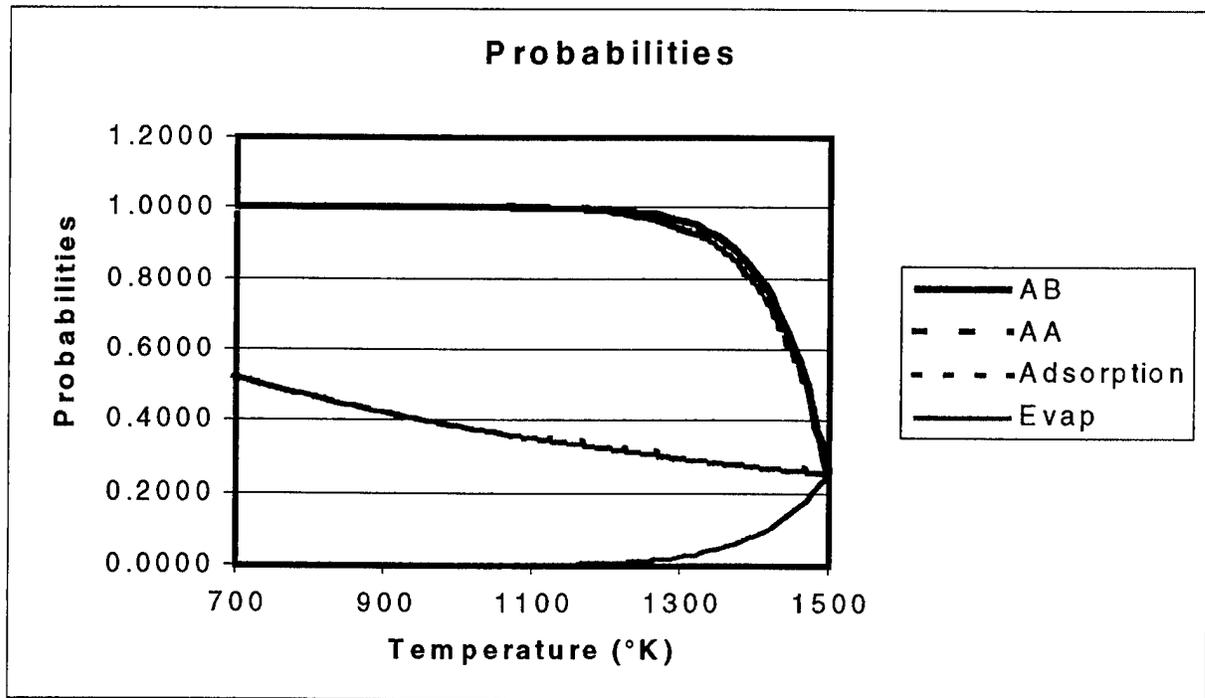


Figure 4. Plot of state change probabilities.

### 5.3. State Change Probabilities

State changes can be written in terms of a probability represented by a modified Arrhenius type equation in which the barrier energy and the temperature are included:

$$p = 1 - C \exp[-E/kT],$$

where C is a normalization constant to insure that  $0 \leq p \leq 1.0$ . Boundary conditions are that  $p(T_{\text{melt}}) \ll 1.0$ , and  $p(T_{\text{room}}) = 1.00000$ .

To relate the calculated p to the random  $p_{\text{rand}}$  chosen in the algorithm, we need to adjust the values of p by normalizing to the number of possible states. So we sum the probabilities for each state to get

$$\text{sum} = p_{AA} + p_{AB} + p_{\text{adsorbed}} + p_{\text{evap}}$$

Then the normalized and scaled value of probability for AA is given by

$$p_{aa}' = p_e' + p_{AA}/\text{sum}$$

and for AB

$$p_{ab}' = p_{aa}' + p_{AB}/\text{sum},$$

and for evaporation

$$p_e' = p_{\text{evap}}/\text{sum.}$$

These values set the boundaries in the  $p_{\text{rand}}$  plot shown in Figure 5. The order shown in the figure is somewhat arbitrary, and the order chosen is one of convenience.

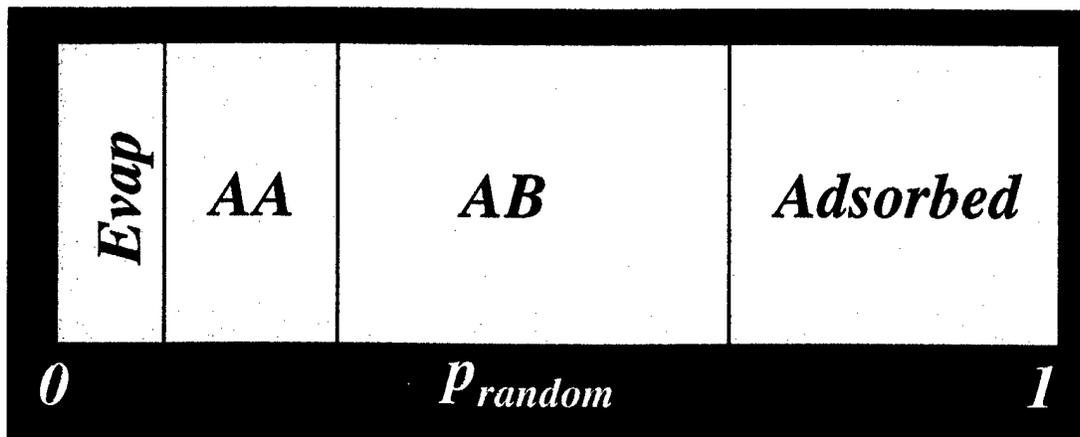


Figure 5. Schematic diagram of the regions for the four possible atom states. The choice of  $p_{\text{random}}$  determines which state is applied.

These boundaries are functions of temperature as well as the barrier energies, and this dependence is shown in Figure 6.

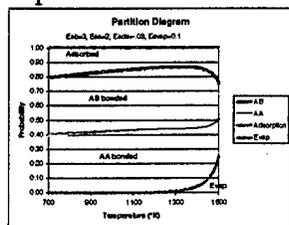


Figure 6. Temperature dependence of boundaries for each region.

## **6. Implementation of the Model**

Implementation of the model has been accomplished, but the process still requires strong user interaction.

The procedure consists of the following steps:

1. Choose the element/crystal system of interest. At present this choice is limited to binary systems; the substrate is the test substrate developed for debugging the NM engine and display.
2. Choose energy barrier values (eV) for AA, AB, adsorption, and evaporation.
3. Choose melting temperature (K).
4. Generate exemplars by using the exemplar macro (requires Excel98). At least 4000 exemplars are needed to provide a good sample of possible states.
5. Use the exemplars as inputs and target for Neuralyst neural net software. Train and test the net. This generates the weights file needed by NM.
6. Format the weights file.
7. Run NM, being sure to enter the correct energy and melt parameters.
8. Upon starting, set the time unit, check the diffusion check box, orient the substrate as desired.
9. Activate the start button.
10. Choose the process desired and follow instructions for each, e.g., set fluxes, temperatures, or other variables.
11. If a movie recording was selected, give the file a name and click OK.
12. Run the simulation for the desired time. Stop by pausing as needed to examine the film, change processes, change parameters.

The inputs received from the state change engine include a set of attribute values, which are fed to the model algorithms. The outputs consist of a set of attributes (the output 'state') with values determined by the model algorithms. At present there are four states allowed: 'vapor', 'AAbonded', 'ABbonded', 'adsorbed'. Code has been developed that allows for steps and ledges which results in an additional three states: 'walls', 'cliffs', and 'wallsandcliffs'. These latter three states are not used at this time.

## **7. Results**

### **7.1. Simulations**

#### **7.1.1. Effect of flux ratio on number of vacancies**

In this series of simulations, the effects of changing the ratio of impinging species is examined by varying the ratio of the 'atoms' used here, As/Ga, from 0.5, to 1, to 2. The figures show the change in the number of vacancies as a function of temperature for the three ratios. Each data point represents the average of 10 simulations in order to reduce

the variation in the vacancy count. Average of 10 simulations was used since a smaller number of simulations yielded a large spread in the standard deviation. The estimated uncertainty is about 10% in the number counted. Melting temperature in each case was set to 1500°K

The procedure was as follows:

A given set of energy and melting temperature parameters were selected and entered into the NM program dialog box upon starting the simulation. Each simulation was run in the 'seconds' time mode for 10 seconds simulation time (this took up to a minute in real time). The substrate atoms were made invisible in the display, the film was rotated so that the interface layer between the film and the substrate was exposed, and the number of vacancies (and unbound atoms) were counted. The substrate used was the standard 'checkerboard' substrate we created for test purposes.

Vacancy concentrations are represented by an activation energy of formation as

$$N(\text{vacancies}) = \text{const} * \exp[-E_{\text{vacancy}}/kT].$$

N should increase as temperature increases, and the rate of increase is expected to be very rapid as the melting temperature is approached.

In Figures 7 and 8 the curves show that the number of vacancies increase with temperature as expected, although the rate of increase is not exponential as implied by the N equation given above. However, we note that the behavior of vacancies in our model is not explicitly present. We have included no equations or model that deals with vacancies directly. Hence, to have behavior as shown is encouraging.

Adjustment of the energies affects the shape of the vacancy-temperature curves, as shown in Figure 9. The cause of the dip at 900K is not known, but we believe the negative value used for the adsorption energy distorts the behavior and produces this deviation from the shape shown in the previous two plots. Also, the uncertainty in the counts will reduce the dip somewhat.

Curve fitting these plots reveals an activation energy of about 0.03 eV, about 30 times too small, using estimates in the literature for metals of about 1 eV. Sources of this discrepancy most probably lie in the probability function constant used. Additional simulations to refine these estimates are planned.

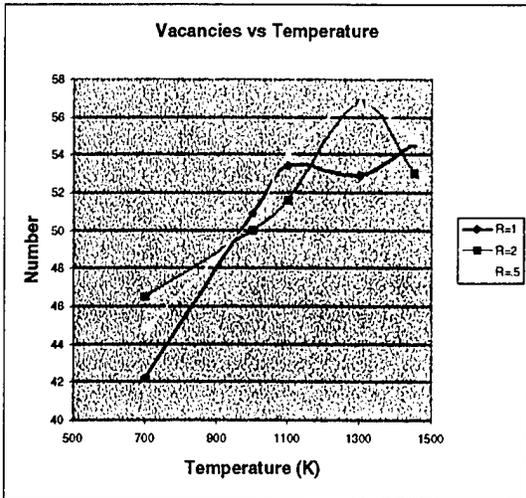


Figure 7. Vacancies vs temperature, Eads = 0.1

Averages

Eab = 3; Eaa = 2; Eads = 0.1; Eevap = 0.1

Temperature	Vacancies	Ratio
700	42.2	1
1000	50.9	1
1100	53.4	1
1300	52.9	1
1450	54.6	1
700	46.5	2
1000	50.0	2
1100	51.6	2
1300	56.7	2
1450	53.0	2
700	44.9	0.5
1000	50.6	0.5
1100	54.1	0.5
1300	56.7	0.5
1450	54.8	0.5

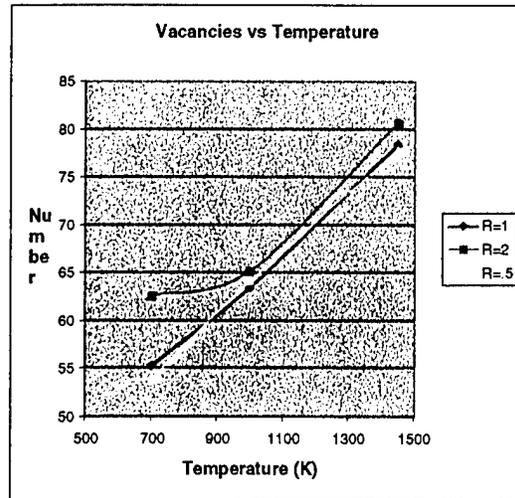


Figure 8. Vacancies vs temperature, Eads = 1

Averages

Eab = 3; Eaa = 2; Eads = 1; Eevap = 0.1

Temperature	Vacancies	Ratio
700	55.3	1
1000	63.4	1
1450	78.3	1
700	62.4	2
1000	65.0	2
1450	80.5	2
700	53.9	0.5
1000	64.1	0.5
1450	77.8	0.5

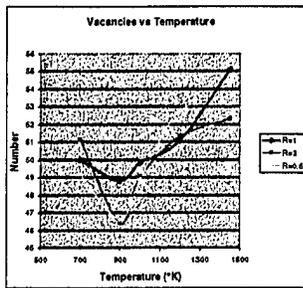


Figure 9. Vacancies vs temperature, Eads = -0.03, Eevap = 2.

#### Averages

Eab = 3; Eaa = 2; Eads = -0.03; Eevap = 2

Temperature	Vacancies	Ratio
700	50	1
900	48.8	1
1000	49.8	1
1200	51.1	1
1450	55.1	1
700	51.1	2
900	46.4	2
1000	49	2
1200	51.3	2
1450	52.3	2
700	49.2	0.5
900	46.6	0.5
1000	48.9	0.5
1200	52.8	0.5
1450	53.2	0.5

### 7.1.2. Time dependence of site occupancies

To determine the behavior of the number of unoccupied sites as a function of time, a simulation was run to enable a count of open sites to be made as a function of simulation time. The substrate temperature was held at 700K, flux ratio at As/Ga = 0.5, energies of  $E_{ab} = 3$ ,  $E_{aa} = 2$ ,  $E_{ads} = 1$ , and  $E_{evap} = .1$  eV,  $T_{melt} = 1500^{\circ}\text{K}$ . The process was MBE and the time scale was seconds. The plot in Figure 10 shows that the number of sites decreases exponentially as expected. The rate is, however, slower than desired by about a factor of about 100, assuming that coverage of 1 layer per second is desired. These results are in broad agreement with the behavior discussed in Example 1, in which the energies found are too small by about a factor of 30.

These results indicate that the model needs to be explored further to identify the source of the scale error in the rate and energy. Modifications to the probability functions are needed since these control the fundamental behavior of the model with temperature and energy. In addition, continuing efforts at code verification via debugging may also show some errors in computation in the code for the algorithm itself.

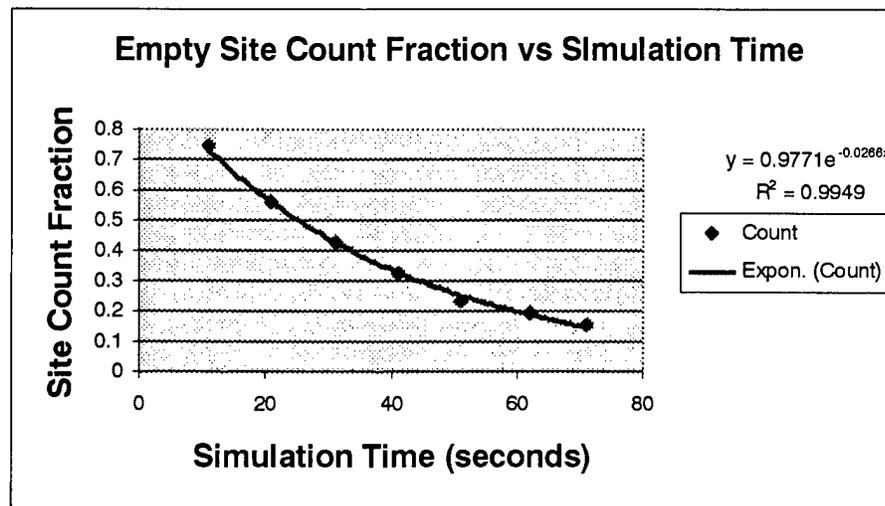


Figure 10. Plot of site count fraction vs simulation time in seconds.

### 7.1.3. Substrate variations

For test purposes a simple square checkerboard substrate was created. Tests of growth can be accomplished using this square format which matches the intrinsic square cell arrangement in the CA engine. For two-species models, this substrate offers test cases for a small (400 atoms), large (1520), very-large (3750), or really-large (10250) substrate. Effects of scaling can be examined, as well as boundary conditions. Variations in species ratio and variations in substrate temperature can be tested to determine refinements needed for the model.

Using the MBE process parameters with ratio of 1 and substrate temperature of about 1480K, the time differences for 10 second growth on the small to the really-large substrates was about a factor of 4, that is, real time of the simulation was about 30 seconds for the small substrate, and

about 125 seconds for the really-large substrate. So for 25 times as many atoms, the simulation time increased by about a factor of 4. Such a slow scaling factor is very promising for simulations involving large numbers of atoms.

The really-large substrate example corresponds to a flat array of atoms about 60x170 atoms in size. Allowing 0.5 nm between atoms, this corresponds to about 0.03x0.085 micron area, still relatively small, but large compared to monte carlo simulations (run on faster and more costly machines). The time to generate films several layers thick is at least 500 times faster than a typical monte carlo run. Also, this time does not include the time to create the display at each time step, which must be done separately. Nor does such simulation allow manipulation of the display during the simulation, nor generation of a movie file until after the simulation is complete. Comparisons with molecular dynamics and first-principles simulations are not possible, since such large arrays are not usually done because of time, cost, or machine constraints. However, times for these simulations are not faster than monte carlo simulations, and frequently they are considerably longer, perhaps factors of 2 to 5, making the time comparison estimate at least a factor of 1000. By improving rendering times, the simulation time for NM can be further reduced, although the amount of improvement is not known at this point in the research.

The really-large substrate (60x170 atom) provides a test for practical display limitations. On a 17" monitor, the display window occupies about a 6x6 inch (150x150 mm) area. The substrate itself occupies a smaller area on the screen, frequently resulting in a Moire pattern. That is, the single atoms are not visible to the eye, and the pixel size and colors interfere.

For a 0.26mm pitch resolution screen, this means that the number of atoms must be limited such that the display size does not fall below or is near the pitch value. For a 60 atom row, this means that the row must be  $60 \times 0.5\text{mm} = 30\text{ mm}$  in length or larger (allowing a factor of two times the pitch for the resolution factor). The size of the substrate in the display window can be increased using the command key and the cursor, but the result if the substrate is made too large is a loss of view of all of the substrate. For movie file generation, the user must decide what aspect of the simulation is to be recorded, and then he must adjust the size accordingly.

The issue of display artifacts is important to consider since film features include line defects. Display artifacts can easily appear to be line defects if the resolution is poor. Hence, care must be exercised in interpreting the display during the simulation. Once a movie file is made, of course, the resolution is fixed, further suggesting that some thought about the nature of the display be done before the simulation is run. Because the time for a simulation is so short, on the other hand, a few runs can be made at several substrate resolutions in order to find an acceptable display for the features of interest.

In any case, this issue of number of atoms also is related to scale changes that must be made as the film grows and grains are nucleated and grow. To simulate microstructure with a grain distribution, atomistic simulation is not desirable because of the number of atoms involved.

#### 7.1.4. Line Defect

In several runs using the 'verylarge' substrate a line defect appears in the film, as shown in the figures. Time for each run was 10 seconds, process: MBE; substrate temperature: 1450K; ratio = 1. The substrate appears to be defect free in the region in which the defect appears in the film. The source of this defect is not clear. Display artifact has been eliminated, since film atoms are absent. There may be a problem in the CA engine. The film may be exhibiting real self assembly behavior that produces the defect. The number of vacancies is very large, and this may be contributing to this defect.

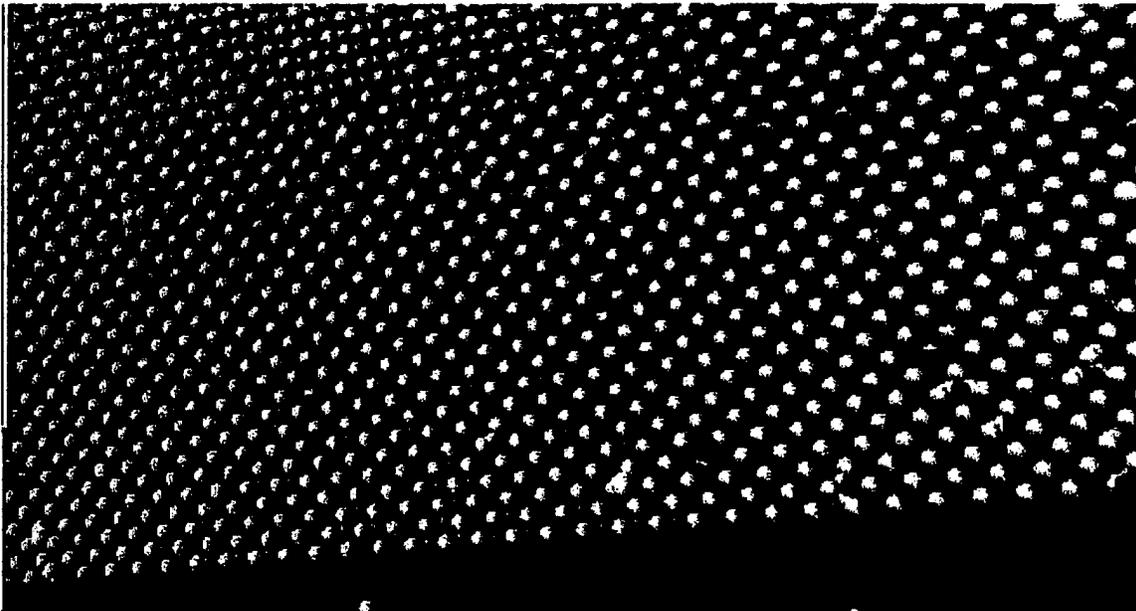


Figure 11. Substrate with no film. Defect appears near the center where three blue atoms appear.

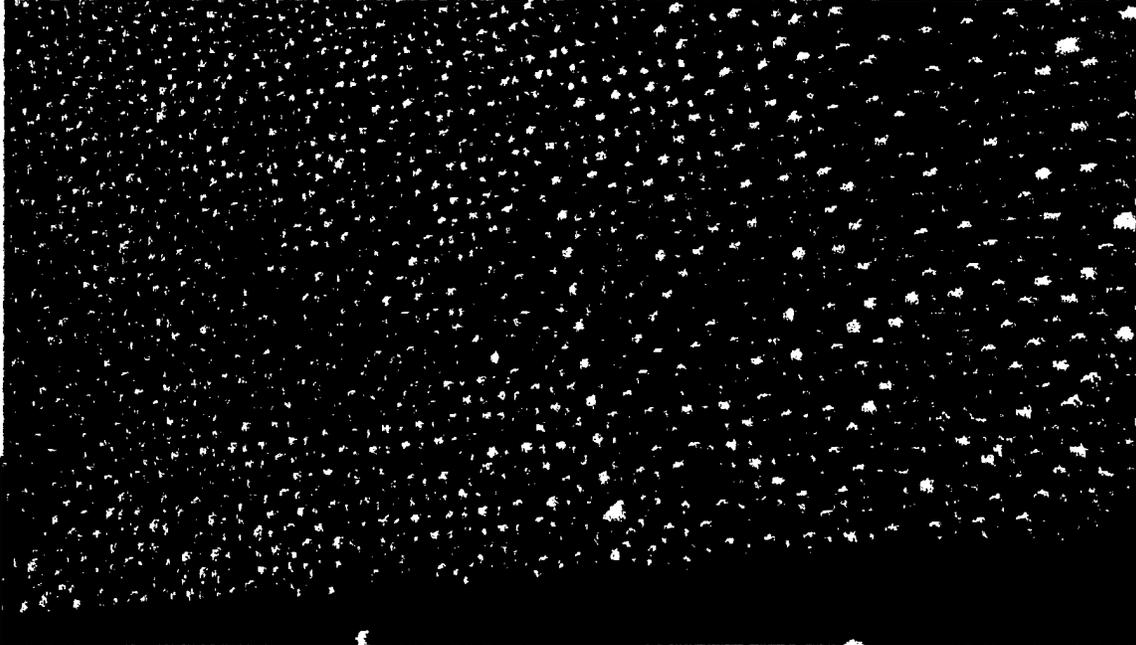
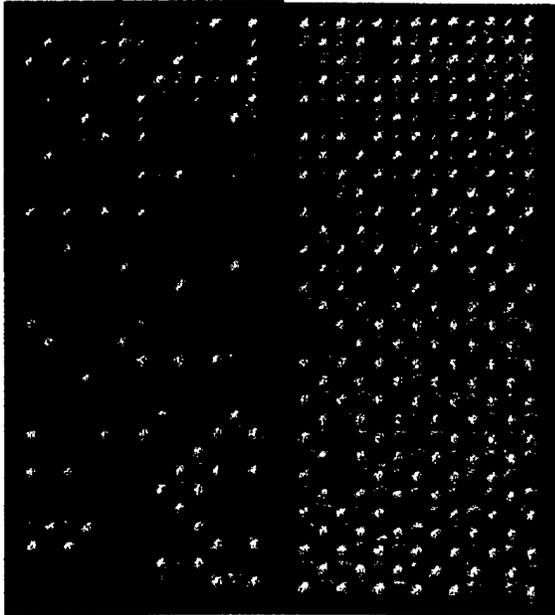


Figure 12. Substrate with film shown. Defect appears near the center where the three blue atoms can be seen.

### 7.1.5. Comparison of substrate interface to film with and without diffusion module active.

Without the diffusion module active, the film shows complete filling of all sites. The number of adsorbed atoms is, however, very high. When bonding does occur, it is ordered, as expected. The generating algorithm needs to be modified to eliminate the large number of adsorbed atoms and replace them with bonded states. The second set is for a high temperature substrate case. The results are similar. See Figure 13.

Low temperature



High temperature

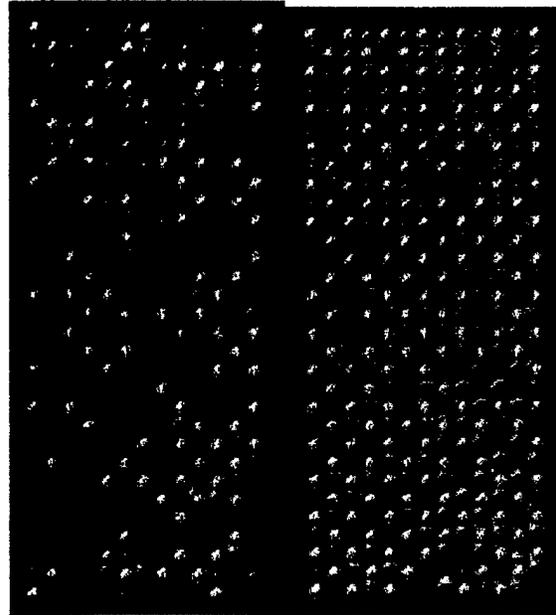


Figure 13. Low and high temperature simulations with and without diffusion.

In the next three figures are shown views of a simulated film grown in 1 second using MBE process on a 'cold' substrate.

Figures 14-15 show views of the substrate-film combination for MBE process for a 1 second growth. Film thickness ranges between 1 and 3 layers. As can be seen in the top view in figure 15, the layers are not complete, and hence the thickness must be averaged, in which case, about a monolayer has been added. This simulation highlights the advantage of being able to view the film from several directions in order to understand the meaning of an average thickness estimate.

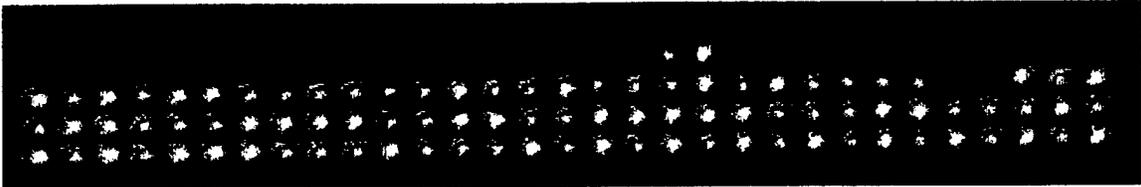


Figure 14. Side view of film. Growth method: MBE; time: 1 sec; Ga/As = 1; substrate temperature 325K. Blue = Ga, Yellow = As.

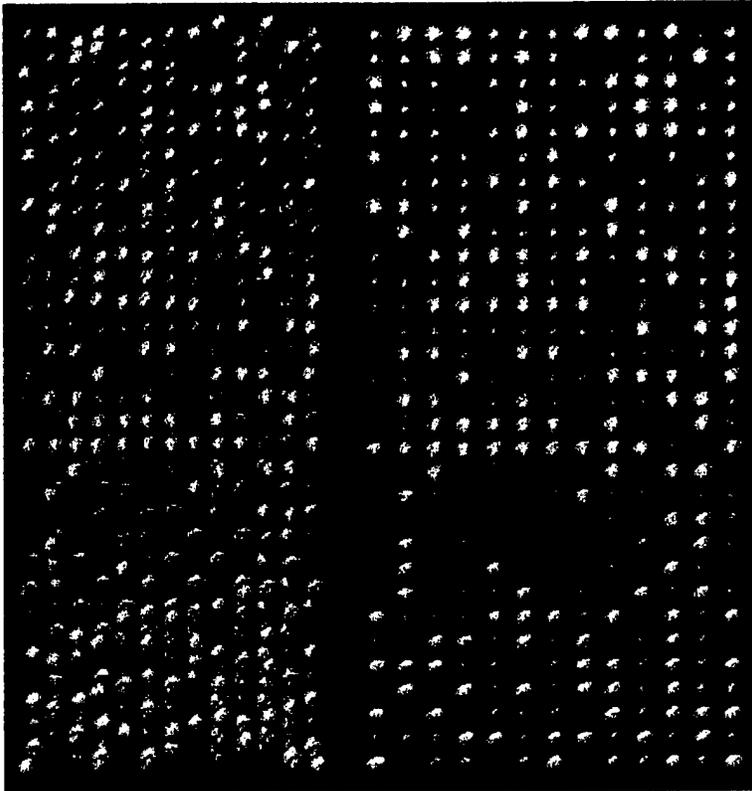


Figure 15. Top view, perspective (left) and orthographic (right). Purple = adsorbed of either species.

Figures 16-17 show the functionality available in viewing the film. In these figures the film atoms have been removed in order to view unbonded atoms present. In Figure 17, the interface between the substrate and the film is shown. This was accomplished by removing the substrate atoms and rotating the film to the substrate side. A detailed view of the film is therefore easily generated.

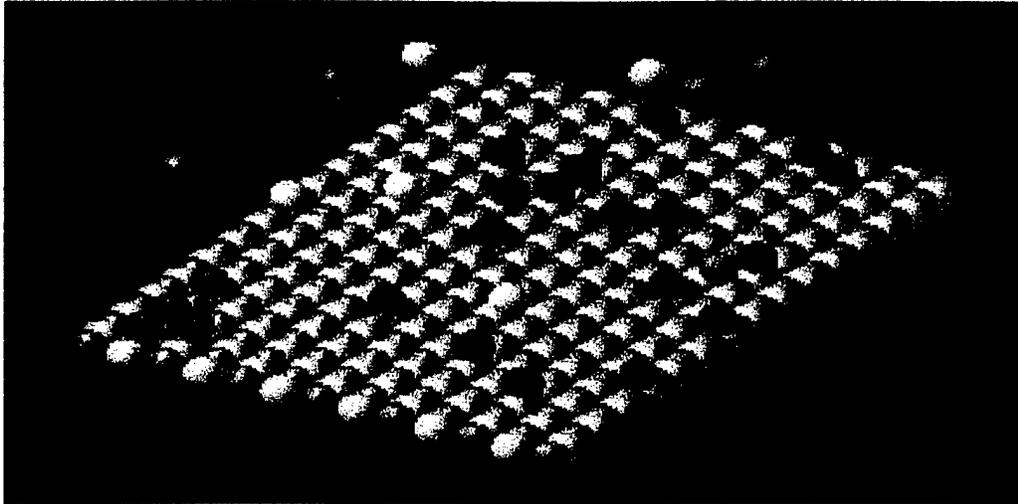


Figure 16. Film atoms removed to show the locations of the adsorbed (unbonded) atoms on the substrate surface. Additional vapor atoms are also shown.

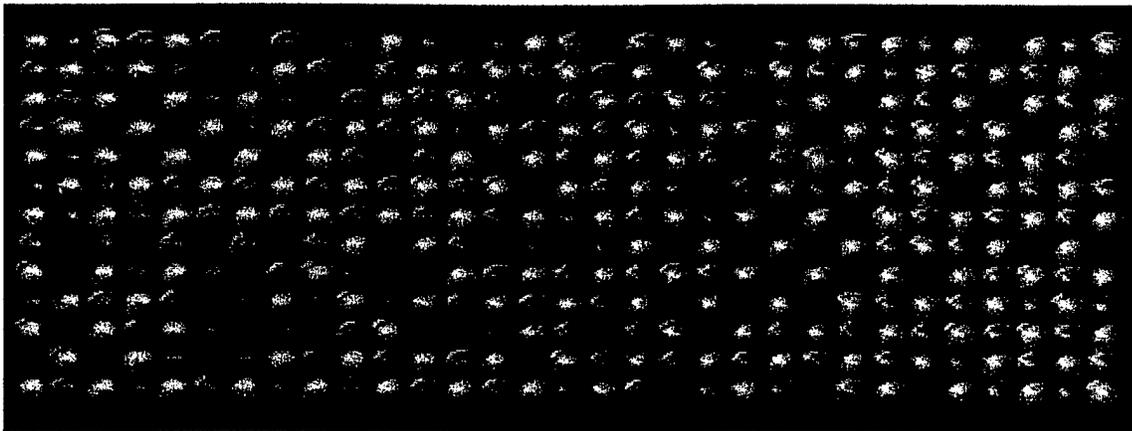


Figure 17. View of film from substrate side (the substrate atoms have been removed). Notice that adsorbed (unbonded) gas is present, vacancies are present, and the film is ordered, i.e., the film atoms are aligned over opposite atom species in the layer below.

Figures 18-20 show results for a Ga/As ratio of 1 and a high substrate temperature. Film thickness is not a large as in the low temperature case, and no unbonded atoms are observed.



Figure 18. Side view of film. Ga/As=1; substrate temperature = 1482 (high).

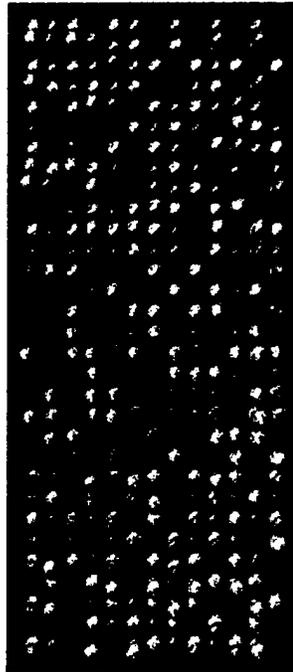


Figure 19. For this case, there are no adsorbed atoms present, the vacancies are large compared to the case for the low-temperature substrate, and the growth is epitaxial as before. The film is two layers thick, while the low substrate case the film is three layers thick.

Top view (perspective)



Figure 20. View from substrate side (substrate atoms removed). The distortion is due to slight rotation toward the viewer.

## **8. Summary**

Approaches to simulation of thin films are of value if they offer advantages as well as uniqueness. Computation of atomic behavior in detail for large numbers of atoms ( $>10^5$  atoms) requires significant computational capability when quantum mechanical, monte carlo, or molecular dynamics methods are used. The benefits obtainable from such simulations are strong enough, however, that a number of efforts are underway to accomplish these computations, even though the costs in time and personpower are high. Also, because of the intractability of calculations, such efforts are not considering optical, magnetic, or combinations of physical properties. On the other hand, the breadth of need for new materials with these specific properties is large. Simulation of micron sized thin films or time scales routinely encountered in processing films are also required.

Cellular automata approaches, extended and applied to atomic level simulation, offer one promising approach to modeling large atom arrays. Although not entirely quantitative because of the use of qualitative and logical equations, the simulations can be rendered in near-real-time. This is an important advantage for process development, because the user visually can identify the sensitivity of film structure and composition to changes in parameters considerably more rapidly than with conventional simulations. Because cellular automata rely on rules for their behavior, however, formulation of meaningful rule sets becomes a major task. Also, for real systems the number of rules and their logical complexity are very high. Addition of quantitative models to these rule sets offsets any speed advantage gained. This research indicates that cellular automata can be used and the gains obtained, thereby, maintained if neural networks are added to the computation. This requires dividing the computation task into off-line and on-line parts, a not impossible split. The upside is a tremendous gain in speed of the simulation, from hours to seconds. The downside is that a large training and testing set of examples is required. Generation of such exemplars may be time consuming. The online predictive capability of the cellular automaton-neural net algorithm, however, is so promising that the trade-off is worth the effort. This approach brings the ability to model atom arrays of millions of atoms that much closer, as well as the ability to model complex materials systems that come close to real materials. The potential for this software system as a tool in process and film design is large.

Results of simulations of simple films indicate that the software can be successfully operated using desktop computation. The model used, based on probability and energy barriers, appears to be meaningful physically. The example studied of vacancy density, approximates expected behavior based on consideration of well established ideas about vacancy concentrations, energy, and temperature dependence. A number of issues remain. Among these are numerical estimates of constants in the probability equations, the form of the equations themselves, some aspects of the rules used in the cellular automaton related to the how to choose directions based on the characteristics of the immediate neighborhood, and how to treat surface diffusion in this model.

This research has been focused on the development of the cellular automaton-neural net concept of stochastic modeling of film growth for real materials. In this respect, the research has been successful in producing software that functions well with a reasonable user interface, once the

neural net has been trained. Generation of the neural net weights file, however, is very user intensive, and this must be changed to be more user friendly.

Validation of the model against molecular dynamics or monte carlo methods is needed to establish a firm link between these methods and the cellular automaton-neural net method.

Expansion of the model to include more than two elements is essential, and should pose no technical barrier. Real crystal structures for substrates also poses no conceptual problem, and in fact code exists now as a supplement to the Nanomodeler software for generating substrates from any crystal system. But the computation engine must be modified to accommodate the openness present in any crystal system, i.e., the unoccupied Wyckoff sites in a crystal structure, for example, the interstitial sites in a cubic structure.

## **9. Acknowledgments**

We gratefully acknowledge the assistance of A. Hoffmaster, E. Barto, J. Reber, and P. Sisk in code generation, simulation runs, and for helpful suggestions.

## **10. References**

[1] R. Grimes, Imperial College website: <http://www.abulafia.mt.ic.ac.uk/USAF/index.html>

[2] A. P. Sutton and R. W. Balluffi, *Interfaces in Crystalline Materials*, Oxford Science Publications, New York, 1996.

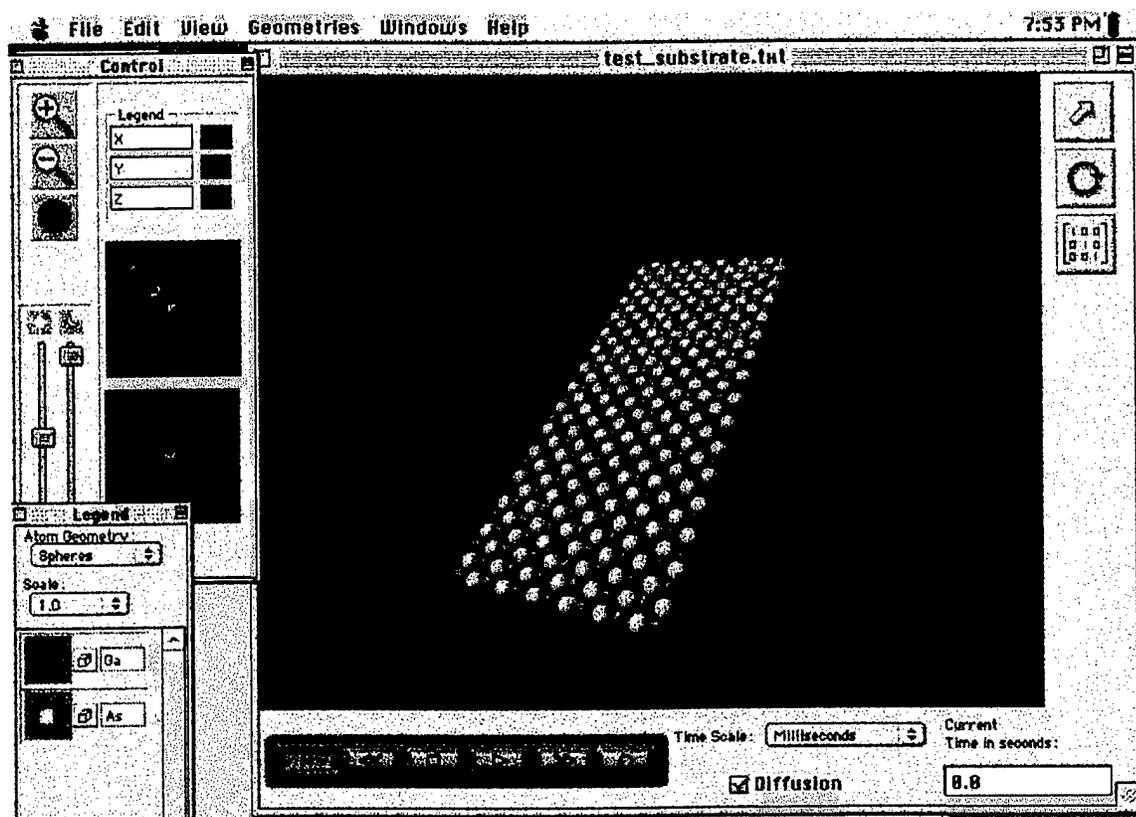
[3] Lecture notes, A. MacKinnon, Imperial College, London, 1997

## Appendix

### Features of Nanomodeler™

These features are for version beta1, February 1, 1999.

Display window with a substrate created. Zoom control is in upper left. Continuous zoom can be done by holding down the command key and moving the mouse in the window. Translation or rotation are done by clicking on the arrow or circular arrow at the upper right. Background colors and lighting can be changed using the controls on the upper left. The orientation is indicated in the small window with the red-green-blue coordinate arrows. The convention is right-handed and the order is  $R = x$ ,  $G = y$ ,  $B = z$ .



The video control bar is located at the bottom left of the display window. From left to right the controls are: pause, reverse, stop, start, forward, record.

<b>File</b>	<b>Edit</b>	<b>View</b>	<b>Geom</b>
<b>New</b>			<b>⌘N</b>
<b>Open ...</b>			<b>⌘O</b>
<b>Close</b>			<b>⌘W</b>
<b>Save</b>			<b>⌘S</b>
<b>Save As...</b>			
<b>Save As 3DMF...</b>			
<b>Revert</b>			
<b>Page Setup...</b>			
<b>Print...</b>			
<b>Run Automata</b>			<b>⌘R</b>
<b>Quit</b>			<b>⌘Q</b>

File menu:

New: create a new window

Open: open a new window from a file, usually a crystal file

Close: close the window

Save, Save As..., Save as 3DMF are not functional

Page setup... and Print... control printing

Run Automata: start the automaton engine

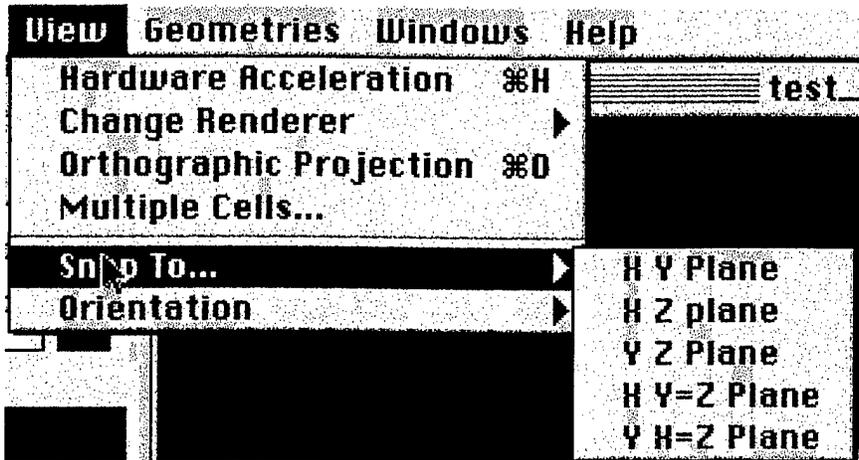
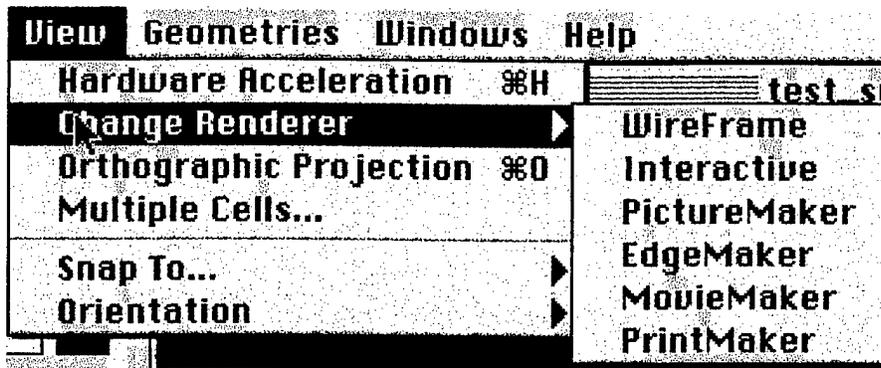
Quit: quit the application

<b>Edit</b>	<b>View</b>	<b>Geome</b>
<b>Can't Undo</b>		<b>⌘Z</b>
<b>Cut</b>		<b>⌘H</b>
<b>Copy</b>		<b>⌘C</b>
<b>Paste</b>		<b>⌘U</b>
<b>Clear</b>		
<b>Deselect All</b>		
<b>Select All</b>		<b>⌘A</b>
<b>Edit Object</b>		

Edit menu:

Only Select All or Deselect All are generally functional

Edit Object refers to a selected object in the window.



View menu:

Hardware Acceleration: not functional

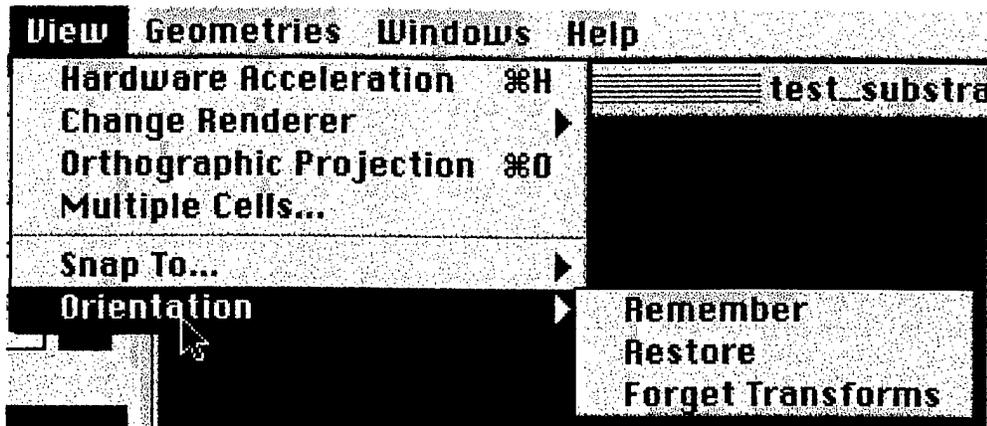
Change Renderer: Wireframe and Interactive are functional. The others are not.

Orthographic Projection: changes from perspective to orthographic projection

Multiple Cells...: Functional for a crystal file

Snap to...: Snap the orientation to the planes indicated

Orientation: Resets the coordinates to whatever orientation currently is shown and makes these the reference coordinates.



**Windows Help**

- ✓ **Control Window**
- Thermal Activity**

Windows menu:

Control Window: The main window

Thermal Activity: Allows thermal effects (disorder) to be added to substrate if the substrate file is a crystal file. Otherwise not functional.

**Geometries Wind**

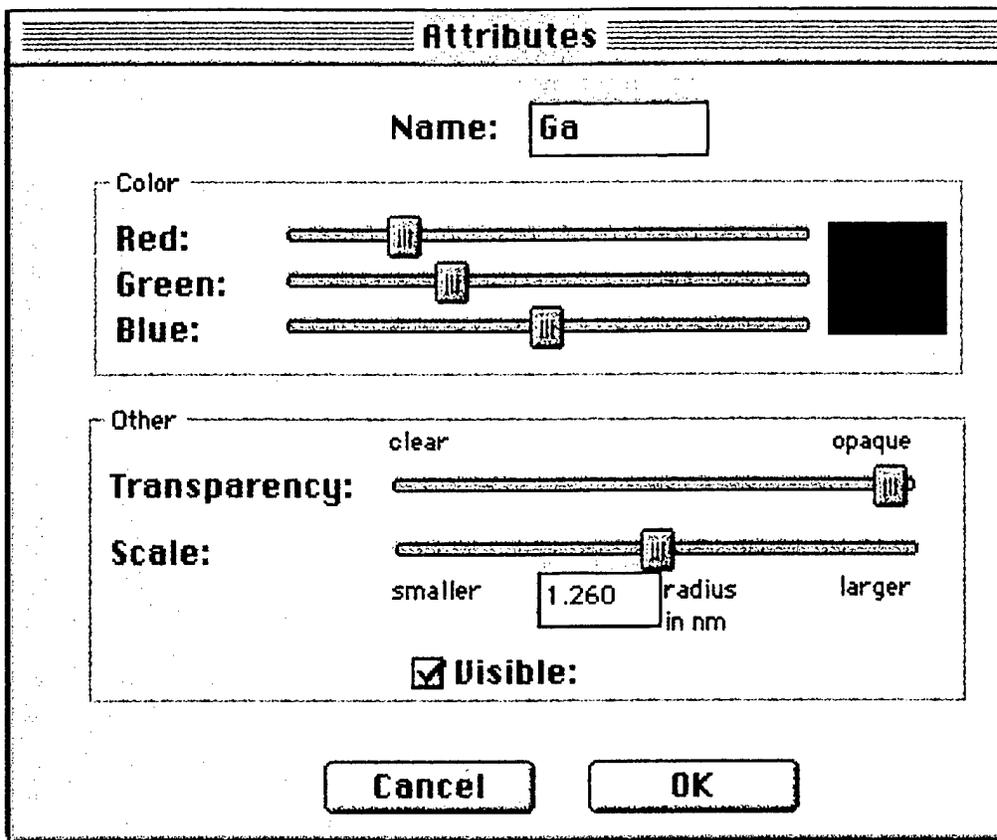
- Create Plane**
- Create Line**

Geometries menu:

Onle active for crystal files.

Create Plane: Create a plane by highlighting three objects in the crystal

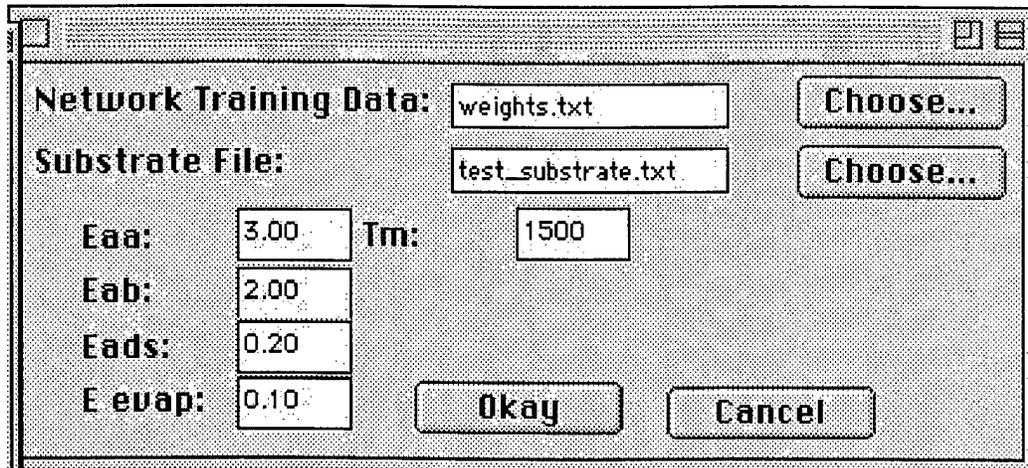
Create Line: Create a line by highlighting two objects in the crystal.



Legend Dialog:

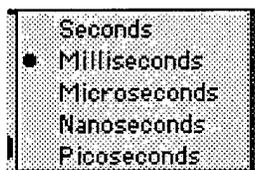
The colors of the object can be changed as desired by sliding the slider bars. The color is shown in the box. The radius of the object is indicated in the box and can be changed by sliding the bar. Transparency is nonfunctional. The object can be made invisible by unchecking the Visible check box.

The colors of the object can be changed as desired by sliding the slider bars. The color is shown in the box. The radius of the object is indicated in the box and can be changed by sliding the bar. Transparency is nonfunctional. The object can be made invisible by unchecking the Visible check box.

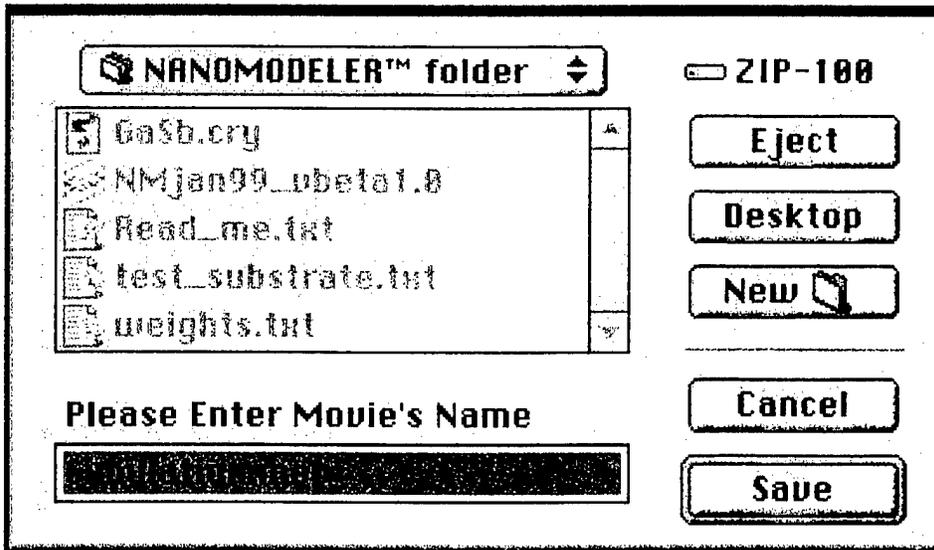


Automata Start Dialog:

Parameters may be changed as needed. The values are those used to generate the weights file. Default values for the default weights-file are shown. Click on the Network Training Data 'Choose' button. Select the appropriate file and click OK from that dialog. Next Click on the Substrate File 'Choose' button to select the substrate file. Select and click OK in that dialog. Click OK to generate the substrate.



Time pop-up window. Select the desired time interval using the mouse.



Movie Dialog: Appears when the 'Record' button is selected in the video control bar. Enter a file name and click SAVE.

**Process Control**

---

Knudsen Cell # 1

Species # 1 : **As**

Cell Temperature in degrees Kelvin

**900**

Knudsen Cell # 2

Species # 2 :

Cell Temperature in degree **Ga**

**900**

Substrate Controls

Substrate Temperature in degrees Kelvin

**900**

**325 K**                      **1900 K**

**Okay**

MBE Process Dialog: appears when the start button on the video control bar is selected. Choose the parameter values by sliding the sliders on each bar. Click OK when done to start the simulation.