2014

Combining Monte Carlo Transport and Level Set Surface Evolution for Modeling Vapor Phase Deposition of Thin Films over Sub-Micron Features

John Lewis Smith
Rose-Hulman Institute of Technology, smithjl@rose-hulman.edu

Combining Monte Carlo Transport and Level Set Surface Evolution for Modeling Vapor Phase Deposition of Thin Films over Sub-Micron Features

A Thesis

Submitted to the Faculty

of

Rose-Hulman Institute of Technology

by

John Lewis Smith

In Partial Fulfillment of the Requirements for the Degree

of

Masters of Science in Chemical Engineering

May 2014

ROSE-HULMAN INSTITUTE OF TECHNOLOGY

Final Examination Report

John Smith
Name

Chemical Engineering
Graduate Major

Thesis Title Combining Monte Carlo Transport and LevelSet Surface Evolution for Modeling Thin-Film Deposition at the Feature Scale

DATE OF EXAM: October 11, 2013

EXAMINATION COMMITTEE:

	Thesis Advisory Committee	Department
Thesis Advisor:	Daniel Coronell	CHE
	Adam Nolte	CHE
	Allen Holder	MA

PASSED X

FAILED ____________

ACKNOWLEDGEMENTS

I would like to thank the Chemical Engineering Department for their help and guidance in the transition from the military to student life. My life has been enriched by those whom I have had classes and personal correspondences.

I would like to thank my advisor, Dr. Coronell, for his mentorship and guidance. I was given free-reign to choose the direction of this work and I believe that I learned much by pursuing a topic of my choice. My correspondence with him is unforgettable.

I was also like to thank Dr. Holder, who introduced me to mathematical analysis. Any future work in mathematics will undoubtedly be the result of his commitment to push me through Principles of Mathematical Analysis.

Lastly, I thank my wife and family for their everlasting support in continuing my education.

ABSTRACT

Smith, John

M.S Chem Eng.

Rose-Hulman Institute of Technology

May 2014

Combining Monte Carlo Transport and Level Set Surface Evolution for Modeling Vapor
Phase Deposition of Thin Films over Sub-Micron Features

Thesis Advisor: Dr. Daniel G. Coronell

A hybrid scheme is used to model the vapor phase deposition of thin films at the feature scale. The transport of the chemical species to the substrate surface is modeled with a Collisionless Direct Simulation Monte Carlo (DSMC) method. The Level Set Method is used to model the growth of the thin-film on the substrate. The convergence criteria for these methods were not found in literature.

The governing equations for the Level Set Method are, in general, non-linear partial differential equations. The coupling of the DSMC Method with the Level Set Method results in a set of non-Gaussian stochastic non-linear partial differential equations. Developing general convergence criteria proved exceedingly difficult, and only qualitative results are presented to support our convergence criteria. Simulation results are in qualitative agreement with experiments and other results from literature.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Background/History

The fabrication of microelectronic devices involves vapor phase deposition of thin films over sub-micron features such as a trench [1,2]. Feature size reduces as more components are packaged in a single device. In the case of a trench, the aspect ratio increases and under the same physical process conditions will lead to less coverage in the trench [3,6]. In Figure 1.1 the trench on the right has a void in which no material was deposited. This creates undesirable effects in the device, and limits how small the device can be. For instance, if the material to be deposited were a conductor, the presence of the void limits the amount of current by effectively increasing the resistance in the trench. Step coverage is a performance measure used to characterize the amount of material is deposited inside the trench versus the substrate surface.

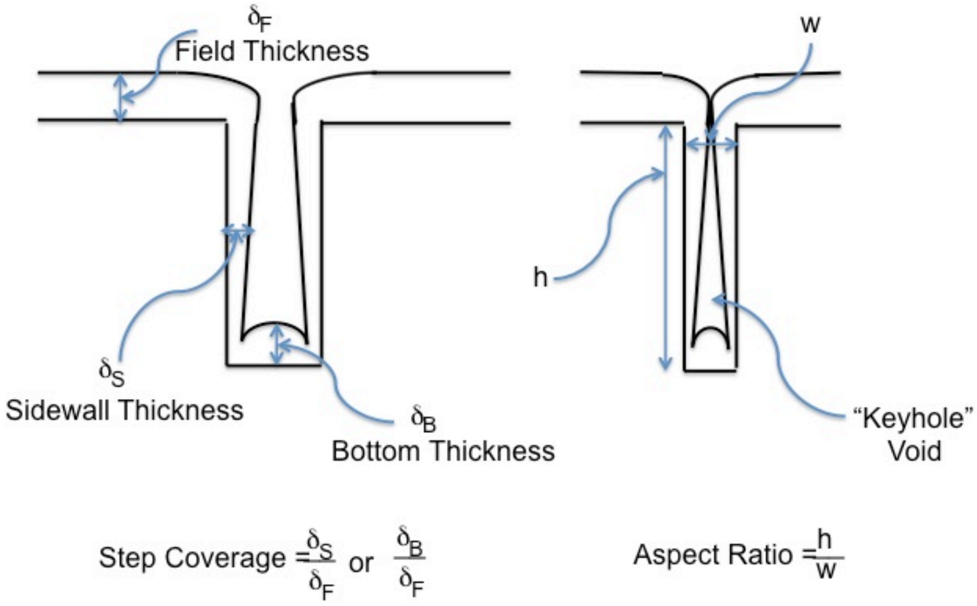

Figure 1.1 Illustration of thin-film deposition in a trench. Higher aspect ratios lead to smaller step coverage and potential keyhole-shaped voids.

1.2 Overview of Scope of Research

The scope of research is to determine the feasibility of combining the Collisionless DSMC Method for transport of species to the film/wafer surface and the Level Set Method for modeling the evolution of the film. The DSMC method allows one to introduce non-equilibrium gas transport not easily described by continuum approach at the expense of computational efficiency. The Level Set Method can be utilized on the surface to increase computational efficiency over discrete methods. The coupling of these methods occurs at the gas-solid interface of the thin film.

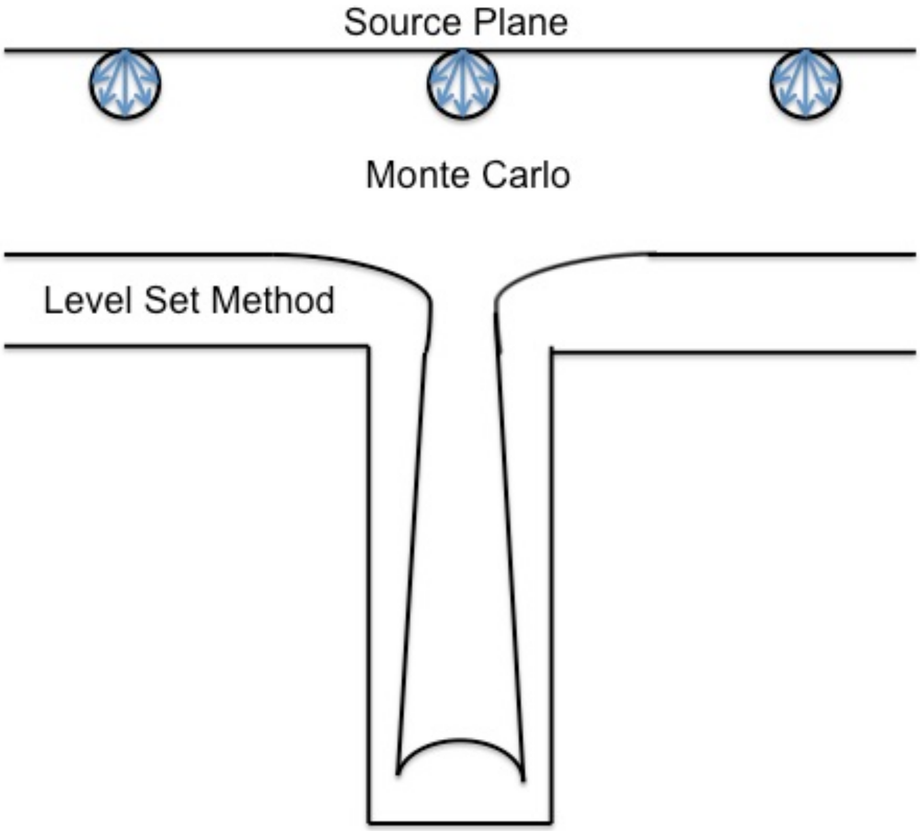

Figure 1.2 A Monte Carlo simulation models the transport of particles to the gas-solid interface. The Level Set Method handles the evolution of the interface. The Collisionless DSMC Method is used for transport to the interface.

Although additional chemistry in the gas phase can change the profile of the film deposition, the focus here is on the gas-solid interface. The important result is that the species in the gas phase has adsorbed on the surface. The additional chemistry, while valid, would only increase computation time and is beyond the scope of this research.

A convex Level Set Method is used to evolve the surface. Non-convex schemes smooth out shocks by adding numerical diffusion. The Level Set Method tracks surface evolution naturally by selecting the appropriate condition at shocks and rarefactions. Additional surface chemistry can also be added to the Level Set Method. Once again, this would not change the success of the Level Set. It is important to note that adding additional surface chemistry may necessitate changing to a non-convex scheme.

2. METHODOLOGY

2.1 Overview of Methodology

The coupling of the DSMC and Level Set is implemented as shown in Figure 2.1. All of the steps contained within the yellow box are implemented in the DSMC Method and all other steps are implemented in the Level Set Method. The DSMC Method generates the flux at the gas-solid interface at each time step. The Level Set Method then evolves the interface for the specified time interval. The details of each method are discussed in the following sections.

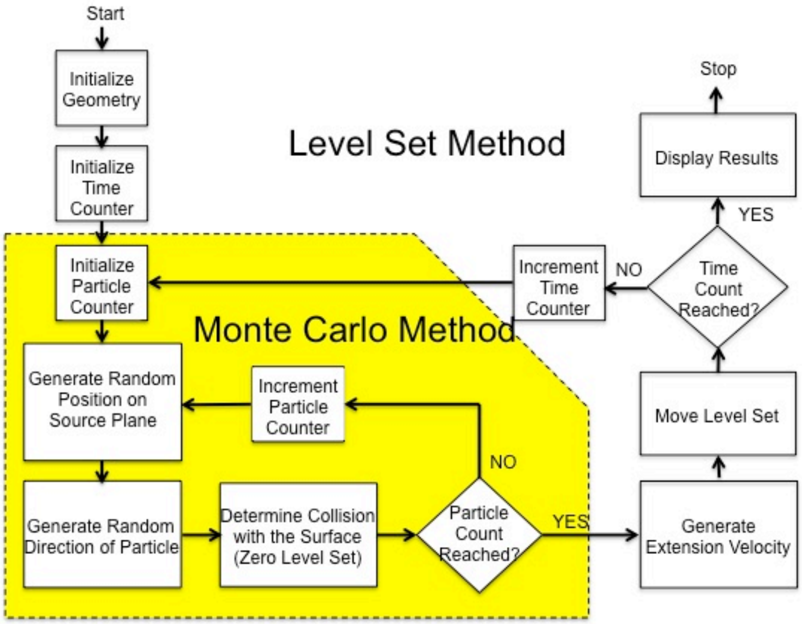
Start
Initialize Geometry
Initialize Time Counter
Level Set Method
Stop
Display Results
YES
Time Count Reached?
NO
Increment Time Counter
Initialize Particle Counter
Monte Carlo Method
Generate Random Position on Source Plane
Increment Particle Counter
NO
Move Level Set
Generate Random Direction of Particle
Determine Collision with the Surface (Zero Level Set)
Particle Count Reached?
YES
Generate Extension Velocity

Figure 2.1 Flow chart for the coupling of DSMC and Level Set methods.

2.2 Monte Carlo Method

The Monte Carlo Method utilized here is described in detail in Bird's book[5]. Particles are generated with a random direction according to a prescribed distribution. The particles are then transported to the surface where they collide and adsorb. In general, the transport to the surface would involve intermolecular collisions before the particle reached the surface. The mean free path of the gas species is significantly larger than the feature scale being modeled. The model is simplified to collisionless ballistic transport.

For ballistic transport, the trajectory of the particle starts at a point on the source plane and continues in straight line until it impacts the surface. The particle flux in a given direction is proportional to the projected area of the source onto the solid angle. The projected area is simply the cosine of the angle to the normal. Hence, there is an angular dependence on the flux emanating from the source plane. Figure 2.2 shows two angular distributions typically used, cosine and power-law cosine.

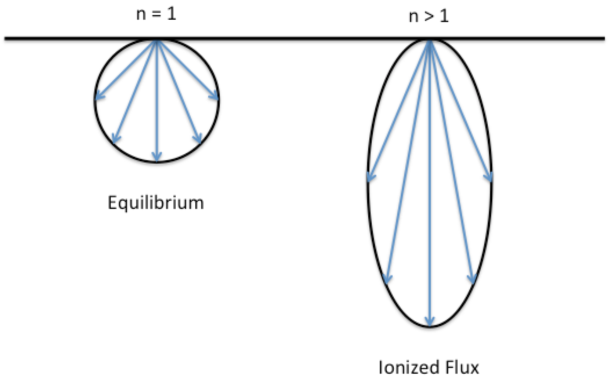

Figure 2.2 Cosine distribution and process conditions typically associated with each.

A cosine distribution (n=1 in Figure 2.2) is the prescribed distribution in the modeling of vapor deposition of an equilibrium gas. The distribution function and cumulative distribution function are as follows,

$$\boldsymbol{f(\gamma) = 2\cos(\gamma)\sin(\gamma)} \qquad \textbf{(2.1)}$$

$$F(\gamma) = 1 - \cos^2(\gamma) = \sin^2(\gamma)$$

$$\boldsymbol{F(\gamma) = \cos^2(\gamma) \quad (\text{convention}).} \qquad \textbf{(2.2)}$$

Note that although the cumulative distribution function is $\sin^2(\gamma)$, the convention is to define it as $\cos^2(\gamma)$ for convenience. The letter, γ, is used for the cone angle to avoid confusion with, ϕ, used in the Level Set Method.

A method is needed to generate γ with the prescribed distribution. Typical tools only have the capability of generating numbers from a uniform or normal distribution. The method here is described in Bird [5]. Generate a random value, ξ, between zero and one from a uniform distribution. If the value passes the acceptance criteria, set the value equal to the cumulative distribution and solve. Here the acceptance criterion is that the cosine is non-negative. For the cosine distribution, the value for γ is obtained solving the equation

$$\boldsymbol{\cos(\gamma) = \sqrt{\xi}.} \qquad \textbf{(2.3)}$$

A highly directional flux is sometimes needed to obtain coverage in high aspect ratios. Introducing a direct current bias and ionizing the gas phase species can achieve the directional flux needed. A power-law cosine distribution (n>1 in Figure 2.2) is used to describe this. The distribution and cumulative distribution function are given as

$$\boldsymbol{f(\gamma) = (n+1)\cos^{n}(\gamma)\sin(\gamma)} \tag{2.4}$$

$$F(\gamma) = 1 - \cos^{\mathrm{n}+1}(\gamma)$$

$$F(\gamma) = \cos^{\mathrm{n}+1}(\gamma) \quad (\text{convention}) \tag{2.5}$$

where the convention like the standard cosine distribution is considered $\cos^{\mathrm{n}+1}(\theta)$. Similarly, the value of θ can be determined from the equation

$$\boldsymbol{\cos(\gamma) = \sqrt[n+1]{\xi}.} \tag{2.6}$$

Spherical coordinates are a natural choice for these cosine distributions. The azimuthal angle, θ, has a uniform distribution and is found by generating a random number between 0 and 2π. The trajectory of the particle is represented as direction cosines,

$$\boldsymbol{\begin{aligned} dc_x &= \cos(\theta)\sin(\gamma) \\ dc_y &= \sin(\theta)\sin(\gamma) \\ dc_z &= \cos(\gamma). \end{aligned}} \tag{2.7}$$

These are used in the collision detection model for ballistic transport to the zero level set.

2.3 Level Set Method

The Level Set Method was first introduced by Sethian to describe the evolution of curves [8]. Much of the success of the Level Set Method is attributed to its ability to handle the evolution of curves naturally [8, 9]. Shocks and rarefactions need not be special cased in order for the method to correctly choose the proper solution. A typical approach for solving a level set is to define a signed distance function such that distances behind the front are negative and distances ahead of the front are positive. The signed distance function for the front itself is zero. Tracking the evolution of the boundary is just a matter of tracking contours of zero. This is commonly referred to as tracking the Zero Level Set. Figure 2.3 shows the evolution of an expanding circle. At each time interval, the Zero Level Set is found. This creates an implicit surface of the evolving front.

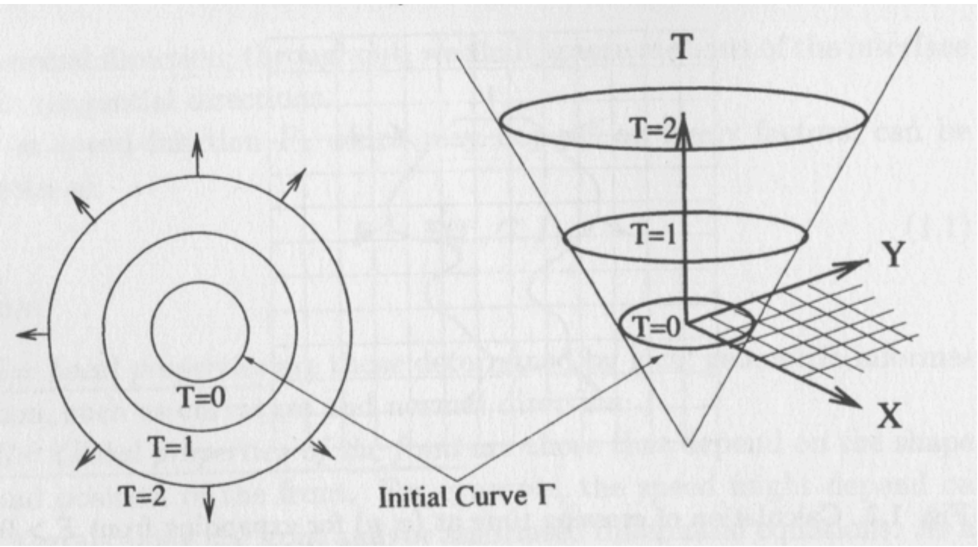

Figure 2.3 Tracking the Zero Level Set as the front moves outward [8].

The Level Set Method over an implicit surface naturally handles shocks and rarefactions. In Figure 2.4, the picture on the left demonstrates a common error when solving a surface evolution in the presence of a shock. The picture on the right uses a Level Set Method to correctly choose the proper solution.

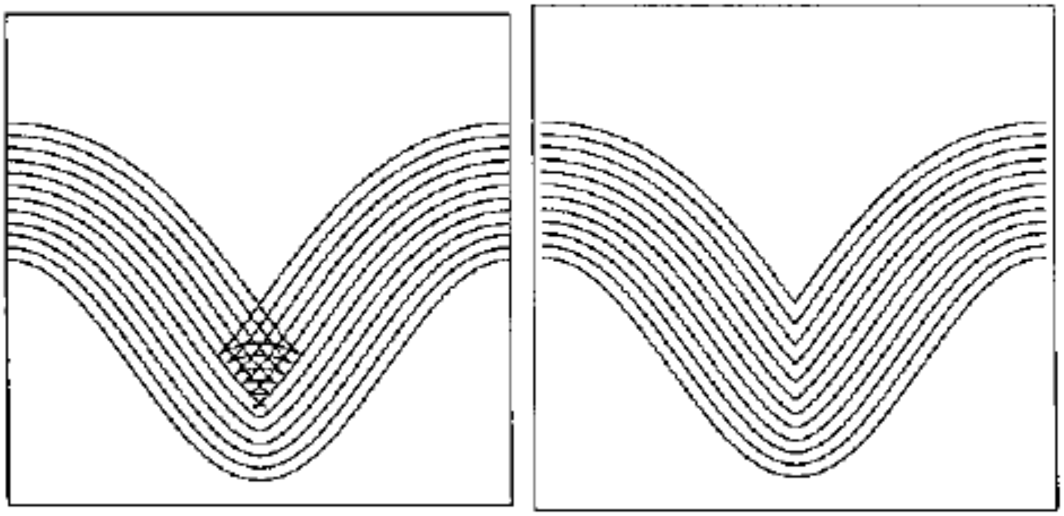

Figure 2.4 Swallowtail on the left and the entropy solution on the right **[8]**.

The Level Set Method is typically identified as solving the following initial boundary value problem [8, 9]:

$$\begin{aligned} &\boldsymbol{\phi_t + F|\nabla\phi| = 0} \\ &\boldsymbol{\phi(x, t = 0)}\ \textbf{known} \end{aligned} \tag{2.8}$$

where ϕ is the signed distance function. This assumes the moving boundary propagating in the direction of the positive values for positive speeds, *F*. The solution is found by tracking $\phi = 0$ throughout the duration of the simulation.

Before discussing the particular method utilized, there is some preliminary nomenclature that needs to be defined. The terms D_i^{-x}, D_i^{+x}, and $D_i^{0x}$are the backward, forward, and center differences defined in the typical sense,

$$\boldsymbol{D_i^{-x} \equiv \frac{\phi(x,t)-\phi(x-h,t)}{h}} \qquad \textbf{(2.9)}$$

$$\boldsymbol{D_i^{+x} \equiv \frac{\phi(x+h,t)-\phi(x,t)}{h}} \qquad \textbf{(2.10)}$$

$$\boldsymbol{D_i^{0x} \equiv \frac{\phi(x+h,t)-\phi(x-h,t)}{2h}} \qquad \textbf{(2.11)}$$

which are first order accurate. The variable x, refers to any spatial variable and subscript i to the particular point on the mesh of which the Level Set equations are to be solved. Extension into two and three-dimensions only requires adding an additional subscript. This notation is typically used in order to describe upwind-schemes. Upwind-scheme refers to choosing the proper values when valuating any parameters for the problem. For instance, a moving boundary propagating in the positive x direction would necessarily have to utilize a backwards difference with respect to x. This is strictly due to the physics of the problem, all the values behind the boundary have been in contact with the front and therefore have information about the front. The values in front of the moving boundary do not have any relevant information about it at its present position. More formally, values are necessarily chosen from the domain of dependence.

The values of the forward and backward derivatives help chose the proper upwind-scheme. As an example, Figure 2.5 shows a front moving horizontally with positive speed. The domain of dependence is behind the front (negative signed distances). This shows, which has been proven in [8], that the forward derivative must be negative and backward derivative must be positive for an upwind-scheme. The canonical form of this statement in two dimensions given in [8] is

$$\boldsymbol{D^x = \sqrt{\max(D^{-x},0)^2 + \min(D^{+x},0)^2}} \tag{2.12}$$

$$\boldsymbol{D^y = \sqrt{\max(D^{-y},0)^2 + \min(D^{+y},0)^2}.} \tag{2.13}$$

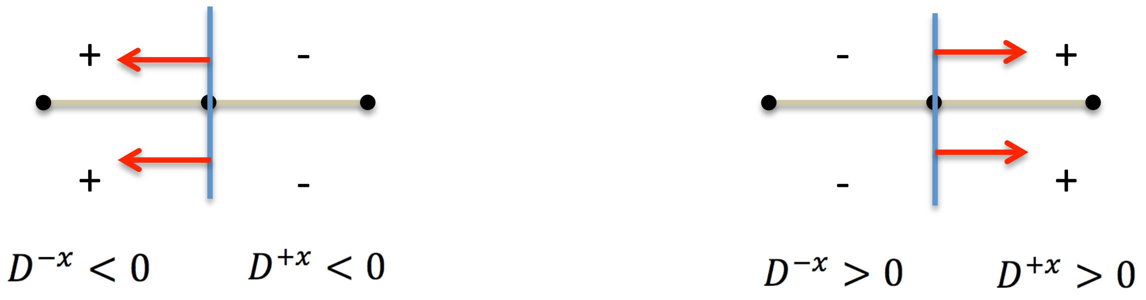

Figure 2.5 Signed-ness of derivatives used for determining upwind-schemes. The direction of the arrows is the direction the front is moving.

The magnitude of the gradient in two-dimensions is [8]

$$\boldsymbol{|\nabla\phi| = \sqrt{\max\,(D^{-x},0)^2 + \min(D^{+x},0)^2 + \max(D^{-y},0)^2 + \min\,(D^{+y},0)^2}} \tag{2.14}$$

and the finite difference equation that solves the Level Set equations is

$$\boldsymbol{\phi_{ij}^{t+1} = \phi_{ij}^{t} \;-\; \Delta t\, F_{ij}|\nabla\phi|.} \tag{2.15}$$

This method is first-order accurate in time and space. The speed function F_{ij} is a matrix containing values for the speed function at every location.

2.4 Extension Velocities

One of the requirements of the Level Set Method is that the speed function is defined for all indices of the model [8]. The Monte Carlo Method only generates a flux at the gas-solid interface, not all indices. The vacant indices are filled with extension velocities. The extension velocities themselves need to solve the level set equations,

$$\phi_t + F_{ext}|\nabla\phi| = 0$$
$$\phi(x, t = 0) \text{ known.} \qquad \textbf{(2.16)}$$

One additional requirement that is convenient to place on the extension velocities is

$$\boldsymbol{\nabla F_{ext} \cdot \nabla \phi = 0}. \qquad \textbf{(2.17)}$$

Satisfying this requirement will maintain the signed distance function with a constant magnitude, $|\nabla\phi| = 1$. The scheme utilized here finds the minimum distance to the front, as suggested by Sethian [8].

3. SIMULATIONS OF VAPOR DEPOSITION OF A THIN-FILM

There were three simulations that were executed: pinhole deposition, deposition in narrow trench, and power-law deposition in a narrow trench. The pinhole deposition on flat wafer is essentially a validation that the correct distribution is being generated. The source of the particle is fixed in position and deposition is modeled accordingly. The deposition in a narrow trench was ran with a typical cosine distribution. Most of the computational and experimental references in this paper are from this type of distribution. The power-law deposition is used to demonstrate how a charged species can more properly fill a narrow trench.

3.1 Pinhole Deposition

This simulation should result in a profile that has a maximum in the middle where the pinhole is located. The edges should be statistically equal and the profile should be sufficiently smooth. Figure 3.1 shows the expected profile.

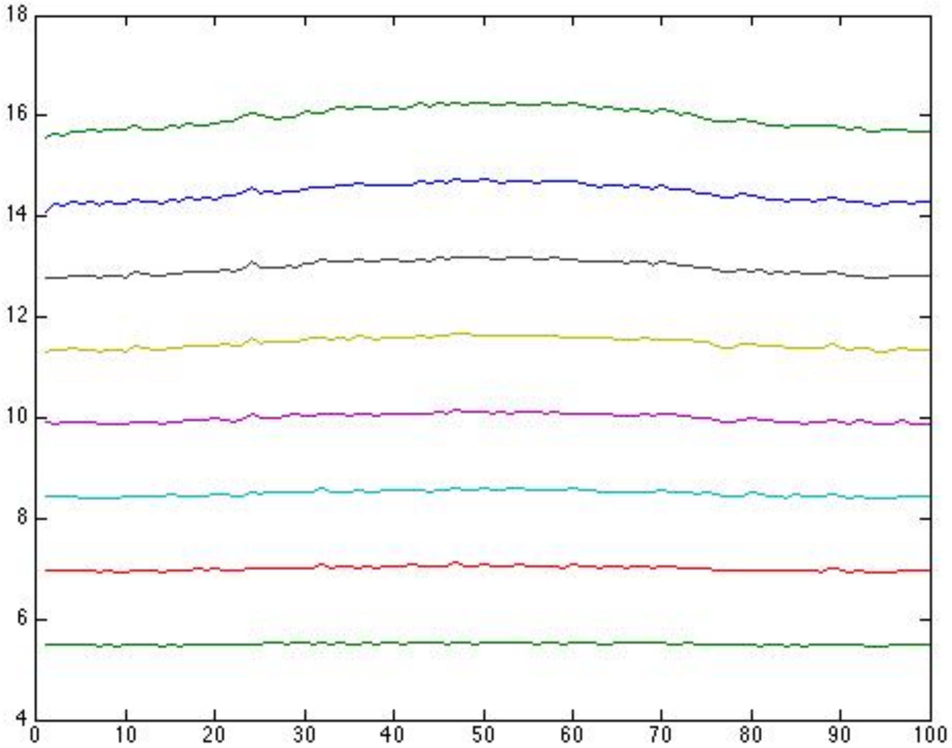

Figure 3.1 Deposition on flat plate with a pinhole source.

3.2 Deposition in a narrow trench

The source point of the particle is fixed at the top, source plane, and uniformly random in the x-direction. In Figure 3.2, the aspect ratio is too high to achieve any appreciable step coverage.

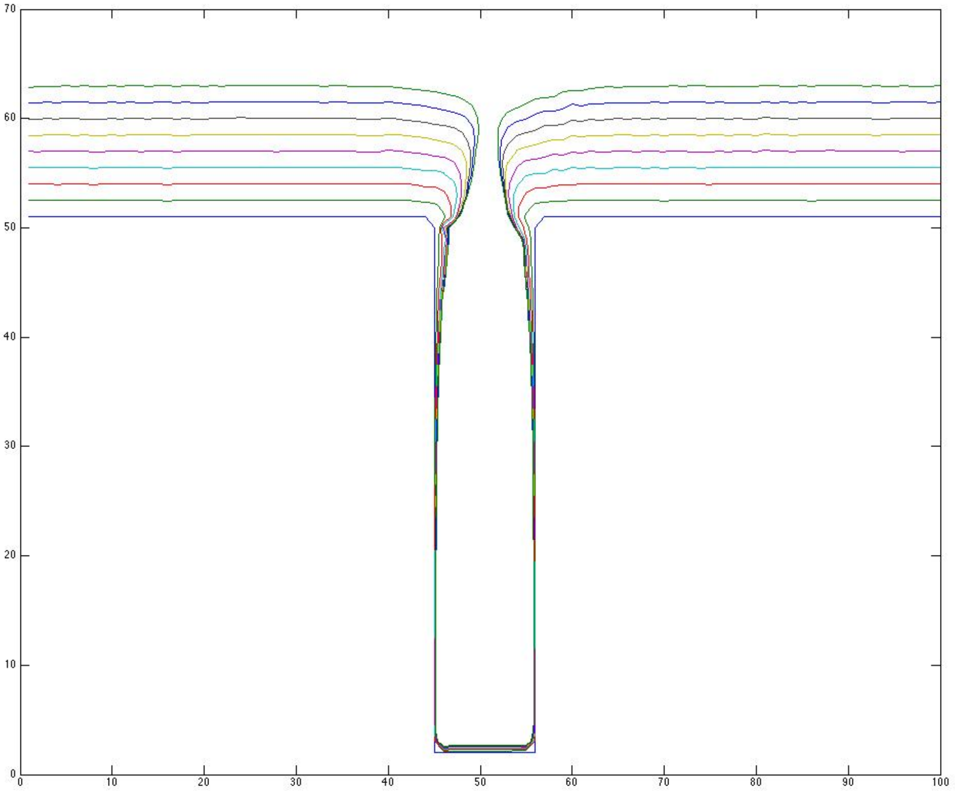

Figure 3.2 Deposition in a narrow trench with a cosine distribution flux. The aspect ratio, bottom step coverage, and side step coverage are 4.5, 0.05, and 0.03, respectively.

3.3 Power-law deposition in a narrow trench

This simulation should show more deposition within the trench. The power-law cosine distribution corresponds to a more vertically directed distribution. The deposition on the surface should be essentially the same as the cosine distribution above since the relative size of the trench width to the surface. Figure 3.3 shows an increase in step coverage for the directional flux modeled by the power-law cosine.

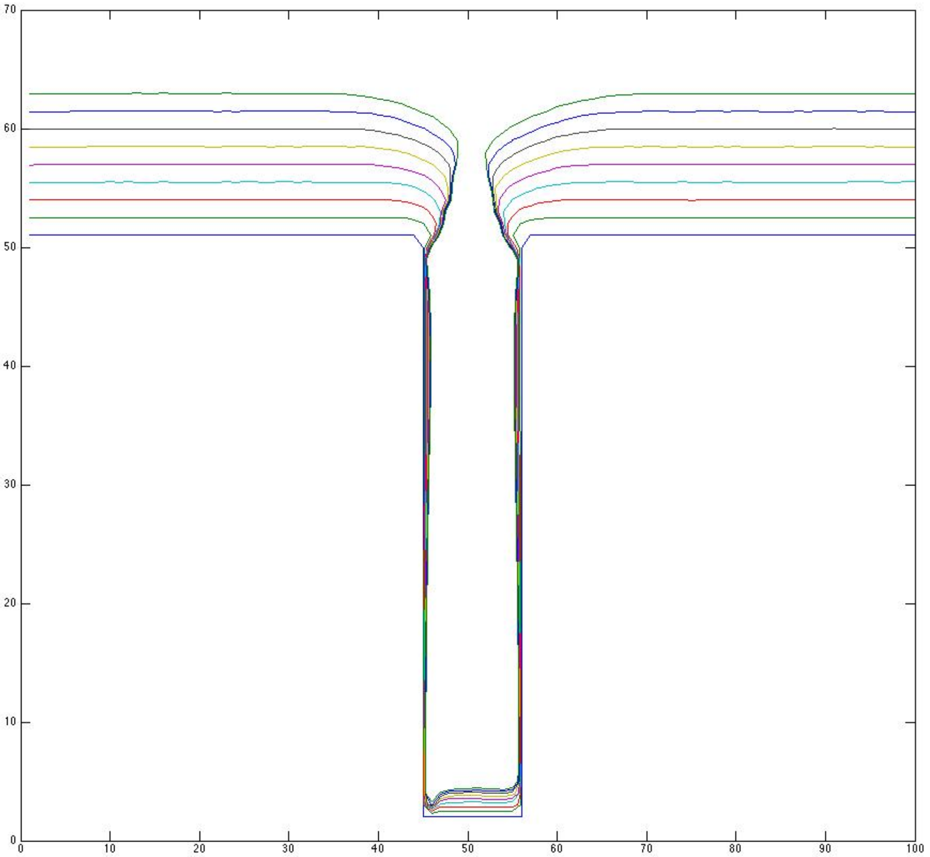

Figure 3.3 Deposition in a narrow trench with power-law cosine (power = 16). The aspect ratio, bottom step coverage, and side step coverage are 4.5, 0.2, and 0.05, respectively.

3.4 Overview of Results

The results obtained are consistent with those found in literature [2, 3, 6, 7, 10, 13]. Only the qualitative profiles were compared. These works typically involved additional gas-phase or surface chemistry not utilized in these simulations. At the time of writing, the work of Al-Mosshen[13] was found outside of the typical literature search. His work demonstrates similar results as the ones found here. The source code for Al-Mosshen's work was not published, only the results.

Difficulties with front evolution are discussed below. Some simulations below were created during the development and testing of the code. Although not all parameters may are consistent, they highlight particular features that need to be discussed. Some features are more easily seen with a different geometry (i.e. a wider trench). Parameters such as grid size and generated particles are not easily seen in the picture but are noted when important.

3.5 Discussion of Results

The profiles generated from the simulations generally follow the expected results noted in literature. There are abnormalities in the profile that are not due to the physics being modeled but due to the model or method itself. One example of this is the handling of shocks at the corners of the geometry as shown in Figure 3.4. A notch forms because of the scheme used to calculate the extension velocities. The notch starts small and grows because it is blocking particles from entering the void. A more robust scheme should be used for calculating extension velocities if this needs to be avoided.

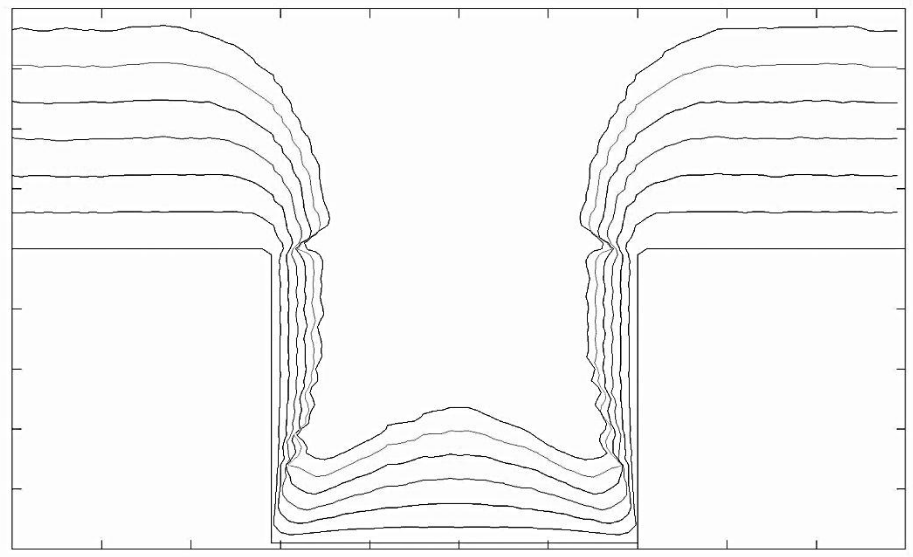

Figure 3.4 Effect of shocks at the corners of the trench.

If the grid size is too large or Monte Carlo samples are too few, the method becomes unstable. This could be an issue of violations of necessary conditions (i.e. Courant-Freidrichs-Lewy condition)[11]. The equation that is being solved, however, is a non-linear stochastic partial differential equation with a non-Gaussian distribution. No meaningful analysis could be performed on this. This issue was resolved by simply decreasing grid size or increasing Monte Carlo samples for a given time interval.

3.6 Verification and Validation of Results

All the profiles generated here, even with the numerical abnormalities, generally follow the shape in Figure 3.5. The photo compares experiential evidence with a well-known commercial package. The trench in the figure is much wider than most considered here.

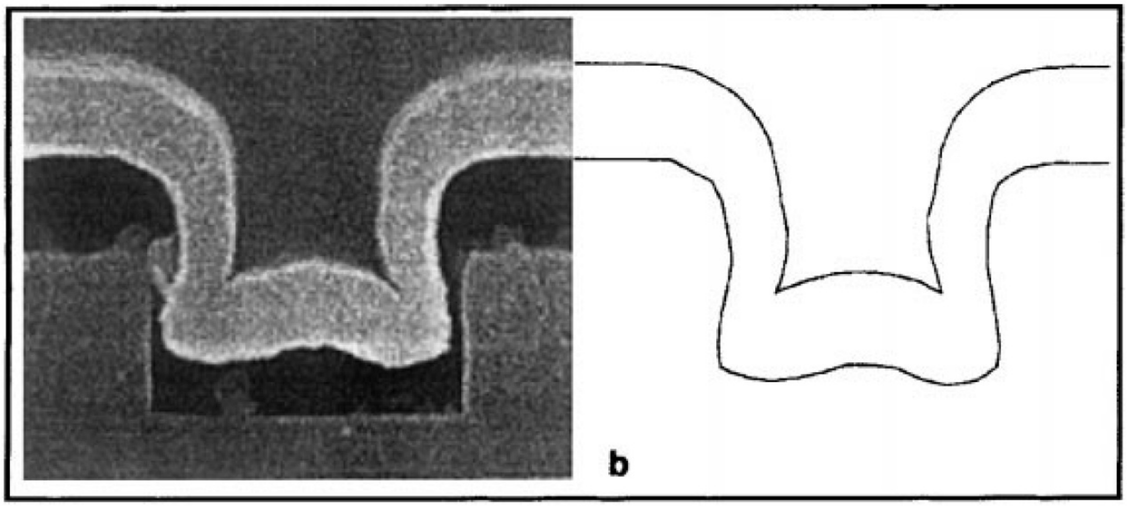

Figure 3.5 SEM image of SiO_2 on Al versus the commercial package EVOLVE **[10]**.

4. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

Developing convergence criteria for the general case proved to be difficult. The equation to be solved numerically is a stochastic non-linear partial differential equation. Convergence was verified empirically by analyzing the resulting profile. Despite the mathematical analysis challenges, coupling the two methods is feasible and does give expected results. Instability in the algorithm is typically due to either an insufficient grid size or a sampling error in the Monte Carlo flux calculation. These two methods, Level Set and Monte Carlo, show that the discrete and continuum approach can be utilized to solve thin-film deposition problems. The advantage of the discrete particle approach is the additional physics that can be easily incorporated into the model while the Level Set Method handles the evolution of the boundary naturally.

The original direction of this work was to determine the feasibility and convergence criteria for coupling a Monte Carlo transport model with a Level Set Method for surface evolution. The latter is a foregone result of this work, however, there are some interesting changes that may make it feasible to at least provide empirical relations for convergence. My general approach in determining convergence was to implement the Level Set Method without any additional complexities or computational shortcuts (such as the Narrow Band), determine convergence criteria, and then see if the work could be easily extended to the other cases. Outlined below are some of the changes recommended to anyone wishing to further this work in determining convergence criteria for the coupling of a Monte Carlo Method with the Level Set Method.

The most notable change to the algorithm is using a Reverse Monte Carlo approach as suggested by Howell [12]. The amount of samples needed in the simulation was only empirically determined. This was after several trial-and-error iterations of the particular aspect ratio. To produce the figures presented takes approximately 6 hours on an Intel i7 with 16GB of RAM. Reverse Monte Carlo provides a means to determine the number of samples needed for a desired accuracy. This may allow one to use the results of stochastic partial differential equations more readily.

For those who wish to add additional chemistry to the models, the Narrow Band formulation is recommended [8]. This eliminates iteration of the entire computational domain when advancing the surface. Implementation of the Narrow Band does require one to re-initialize once the approximated error of the advancing front becomes too large. This is covered by Sethian [8] and implemented in Al-Mosshen's work[13] although no code is provided.

LIST OF REFERENCES

[1] C. Liu, Foundations of MEMS, 2nd Edition, Upper Saddle River: Prentice Hall, 2011.

[2] D. G. Coronell, "Simulation and Analysis of Rarefied Gas Flows in Chemical Vapor Deposition Processes," MIT PhD Dissertation, Cambridge, 1993.

[3] C. T. S., G. T.H and R. G.B., "A fundamental feature scale model for low pressure deposition processes," *Journal of Vacuum Science and Technology A,* vol. 9, no. 3, pp. 524-529, 1991.

[4] J. O. Hirschfelder, C. F. Curtiss and R. B. Bird, Molecular Theory of Gases and Liquids, New York: John Wiley & Sons, Inc., 1954.

[5] G. A. Bird, Molecular Gas Dynamics and the Direct Simulation of Gas Flows, Oxford: Oxford University Press, 1994.

[6] M. K. Gobbert and T. S. Cale, "A kinetic transport and reaction model and simulator for rarefied gas flow in the transition regime," *Journal of Computational Physics,* vol. 213, no. 2, pp. 591-612, 2006.

[7] M. Meyyappan, Computational Modelling in Semiconductor Processing, Norwood: Artech House, Inc., 1995.

[8] J. A. Sethian, Level Set Methods and Fast Marching Methods, New York: Cambridge University Press, 1999.

[9] S. Osher and R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, New York: Springer-Verlag, 2003.

[10] T. S. Cale, B. Rogers, T. Merchant and L. Borucki, "Desposition and etch processes: continuum film evolution in microelectronics," *Computational Materials Science,* vol. 12, no. 4, pp. 333-353, 1998.

[11] R. Courant, K. Freidrichs, H. Lewy, "On the Partial Difference Equations of Mathematical Physics," *IBM Journal*, pp. 215-234, March 1967.

[12] J. R. Howell, R. Siegel, M. P. Menguc, Thermal Radiation Heat Transfer, 5th Edition, Boca Raton: CRC Press, 2011.

[13] H. A. Al-Mohssen, "Chemical Vapor Deposition Models Using Direct Simulation Monte Carlo with Non-Linear Chemistry and Level Set Profile Evolution," MIT Master of Science Thesis, Cambridge, 2003.

All simulations were written in Matlab. No special functions were utilized outside of the core product. There are two files that make up the simulation: first_order_level_set.m and createSpeedFunction.m. All the Level Set Method code is contained within the first_order_level_set.m file with the exception of calculation of extension velocities. The DSMC and extension velocities are all contained within createSpeedFunction.m. This easily enables one to switch the method by which the speed is calculated by only modifying one line of code in the first_order_level_set.m.

APPENDIX A.

MATLAB Code for Level Set Method

first_order_level_set.m

```
old=0;
N = 100;
T = 1200;

Fmax = 1;
dx = 1;
dy = 1;
dt = .01;

r = Fmax*dt/dx;

%Initial Conditions
phi = zeros(N,N,2);
h = .5*N;
wi = .45*N;
wm = .1*N;

for i = 1:N
    for j=1:N
        if i<=h
            if ( j<=wi | j > wi+wm )
                phi(i,j,1) = h-i;
            elseif j < wi+wm/2+1
                phi(i,j,1)= sqrt((h-i)^2+(wi-j)^2);
            else
                phi(i,j,1)= sqrt((h-i)^2+(wi+wm+1-j)^2);
            end
        elseif i==N
            if j < wi
                phi(i,j,1) = -sqrt((N-1-i)^2+(wi-j)^2);
            elseif j > wi+wm
                phi(i,j,1) = -sqrt((N-1-i)^2+(wi+wm+1-j)^2);
            else
                phi(i,j,1) = N-1-i;
            end
        else
            if j <= wi
                phi(i,j,1)=-min(abs(h-i),abs(wi-j));
            elseif j > wi+wm
                phi(i,j,1)=-min(abs(h-i),abs(wi+wm+1-j));
            else
                phi(i,j,1) = min(min(abs(wi-j),abs(wi+wm+1-j)),abs(N-1-i));
            end
        end
    end
end

C = contourc(flipud(phi(:,:,1)),[0 0]);
C(:,1)=[];


plot(C(1,:),C(2,:))
hold all

bdx = 0;
fdx = 0;
bdy = 0;
fdy = 0;

t0=0;
t1=0;

%First order convex scheme
for t=1:T

    if(mod(t,2)==0)
        t0=2;
        t1=1;
    else
        t0=1;
        t1=2;
    end

    F = createSpeedFunction(Fmax,N,phi(:,:,t0));
    for i=1:N
        for j=1:N

            %x derivatives
            if j==1
                bdx = (phi(i,1,t0)-phi(i,N,t0))/dx;
                fdx = (phi(i,2,t0)-phi(i,1,t0))/dx;
            elseif j==N
                bdx = (phi(i,N,t0)-phi(i,N-1,t0))/dx;
                fdx = (phi(1,1,t0)-phi(1,N,t0))/dx;
            else
                bdx = (phi(i,j,t0)-phi(i,j-1,t0))/dx;
                fdx = (phi(i,j+1,t0)-phi(i,j,t0))/dx;
            end

            %y derivatives
            if i == 1
                bdy = 1/dy;
                fdy = (phi(i+1,j,t0)-phi(i,j,t0))/dy;
            elseif i == N
                bdy = (phi(i,j,t0)-phi(i-1,j,t0))/dy;
                fdy = -1/dy;
            else
                bdy = (phi(i,j,t0)-phi(i-1,j,t0))/dy;
                fdy = (phi(i+1,j,t0)-phi(i,j,t0))/dy;
            end

            phi(i,j,t1) = phi(i,j,t0) -
dt*F(i,j)*sqrt(max(bdx,0)^2+min(fdx,0)^2+max(bdy,0)^2+min(fdy,0)^2);
        end
    end

    progress = t/T*100

    if(mod(t,150)==0)
        C = contourc(flipud(phi(:,:,t1)),[0 0]);

        if numel(C) == 0
            break;
        else
            C(:,1)=[];
        end

        plot(C(1,:),C(2,:))
        hold all
    end
end
```

APPENDIX B.

MATLAB Code for Monte Carlo Flux and Extension Velocities

createSpeedFunction.m

```
function F = createSpeedFunction(Fmax,N,phi)

    F=zeros(N,N);
    pnum = 10*N*N;
    psize = Fmax*N/pnum;

    maxflux = 0;

    for p = 1:pnum
        srcx = randi(N);
        srcy = 1;
        theta = 2*pi*rand(1);
        gam = acos(sqrt(rand(1)));

        if cos(gam) <= 0
            p = p-1;
        else
            maxsteps = (N-srcy)/cos(gam)*2;

            for k = 1:maxsteps
                di = round(srcy+k/2*cos(gam));
                dj = round(srcx+k/2*cos(theta)*sin(gam));
                [di, dj] = correctBounds(di, dj, N);

                if phi(di,dj) <= 0
                    break
                end
            end

            F(di, dj) = F(di, dj)+psize;
            maxflux = max(F(di, dj), maxflux);

        end
    end

    maxflux
    F = buildExtensionVelocity(F,phi,N);

end

function Fext = buildExtensionVelocity(F,phi,N)
```

```
    Fext = zeros(N,N);

    [row,col,value] = find(F);
    var(value)
    nrow = length(row);
    dist=realmax*ones(nrow,1);

    for i=1:N
        for j=1:N

            if F(i,j) > 0
               Fext(i,j)=F(i,j);
            else
                for k=1:nrow
                    dist(k) = (row(k)-i)^2 + (col(k) - j)^2;
                end

                [dsq, ind] = min(dist);
                Fext(i,j) = value(ind);
            end
        end
    end
end

function [di, dj] = correctBounds(di, dj, N)
    if di < 1
        di=1;
    elseif di > N
        di=N;
    end

    if dj > N
        dj = mod(dj,N);
    elseif dj < 1
        dj = mod(dj,N);
    end

    if dj==0
        dj=N;
    end
end
```