Simulation of Quantum Dot Formation Using Finite Element Method
ME-599 Term Project
Winter 07

Abstract

In this work, we employed the finite element method to study various effects on a thin film system such as the surface energy contribution, the elastic strain energy contribution, and the bulk free energy contribution. We found that the surface energy causes the film to remain flat while the strain energy term promotes an increase in amplitude of surface modulation. The elastic energy also causes evaporation and the bulk free energy term was introduced to model vapor deposition. With these contributions, we were able to simulate the quantum dot behavior during their formation.

Introduction

Quantum dots are nanometer-sized particles or islands of a semiconductor material embedded in another semiconductor material. They have electronic properties different from that of the bulk due to quantum confinement, and thus hold a promise for nanotechnology applications such as LED's, detectors, data memory devices, lasers, and single electron transistors. However, fabricating a regular, perfectly aligned dot structure is still a challenging issue. Rather than using an accurate positioning device, such as a focused ion beam, it is preferable to use the technique of self-organizing/assembling growth via a strain relaxation mechanism. This is done by depositing the vapor of a material onto a substrate made of a material with a different lattice parameter. It is found experimentally in many material systems that heteroepitaxial growth results in spontaneous self-organization and assembly of islands. [1,2]

The driving force for the quantum dot formation is the reduction of the total energy with a contribution from the elastic strain energy. The elastic strain arises by the lattice misfit between the film and the substrate material. Initially, the combination of the surface/interfacial energies and the strain energy are such that the system favors wetting. Therefore, the film material forms a flat wetting layer over the substrate. As the film material is deposited, the thickness of the film increases and the strain relaxation mechanism will result in island formation, despite an increase in the surface energy. At the early stage of surface roughening, the material is more relaxed at the crest than at the valley. In other words, the area at the crest has a lower chemical potential. Consequently, the film material diffuses from the valley to the crest, which eventually leads to island formation.[3]

In this project, we examine the formation of quantum dots by employing the finite element method formulated by Ref. 4, 5.

Formulation

Based on method developed by Ref. 4 and Ref. 5 and Ref. 6, we used the model for surface diffusion and evaporation/condensation. The weak statement can be written as

$$\int \left[\frac{J.\delta I}{M} + \frac{j\delta i}{m} \right] dA = -\delta G \qquad (1)$$

where J is the mass flux, I is the mass displacement, M is the mobility of atom on the surface, j is the volume of matter added per unit area of the solid surface per unit time, i is the volume of matter added per unit area and m is the evaporation-condensation rate, and δG is the energy change due to matter relocation and exchange on the surface. The first term in the integral associates with surface diffusion and the second term associates with the condensation/evaporation process.

The quantity δG can be obtained from the free energy consideration. Here we consider a) the surface energy, γ_s, b) the elastic energy, w, due to a misfit between the film and the substrate, and c) free energy per unit volume of atoms in a bulk solid phase, g, to model deposition from the vapor phase. The expression δG is

$$\delta G = \gamma_s \delta l + \int (w + g)\delta r_n dA , \qquad (2)$$

where l is the length of the surface, δr_n is the normal displacement of the surface.
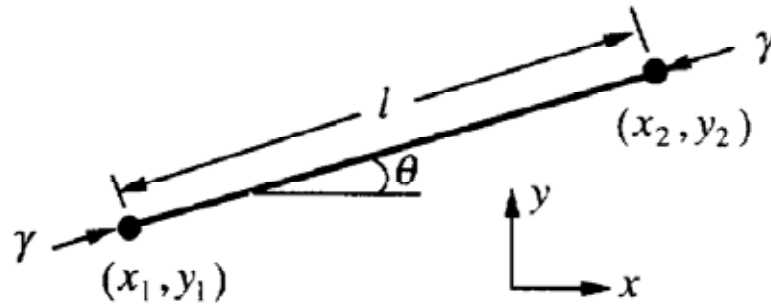
Computational Method

Fig. 1. A strait line element

Fig. 1 shows one element, with two nodes at the positions (x_1, y_1) and (x_2, y_2). The element has length l and slope θ, which relate to the nodal positions by the expression $l*\sin\theta = y_2-y_1$ and $l*\cos\theta = x_2-x_1$. We specify a point on the element by its distance from the mid-point of the element, s. When the two nodes change their positions by $(\delta x_1, \delta y_1)$ and $(\delta x_2, \delta y_2)$, the element become straightened, elongated, translated, or rotated, according to the nodal position changes. Consequently, the element moves in the normal direction by distance

$$\delta r_n = N_1\delta x_1 + N_2\delta y_1 + N_3\delta x_2 + N_4\delta y_2, \qquad (3)$$

with the linear interpolation coefficients being

$$N_1 = -\left(\frac{1}{2} - \frac{s}{l}\right)\sin\theta, \ N_2 = -\left(\frac{1}{2} - \frac{s}{l}\right)\cos\theta$$
$$N_3 = -\left(\frac{1}{2} + \frac{s}{l}\right)\sin\theta, \ N_4 = +\left(\frac{1}{2} - \frac{s}{l}\right)\cos\theta. \tag{4}$$

Also for velocity,

$$v_n = N_1\acute{X}_1 + N_2\acute{Y}_1 + N_3\acute{X}_2 + N_4\acute{Y}_2 \tag{5}$$

However, for flux,

$$J = Q_1 J_1 + Q_2 J_2 + Q_m J_m, \tag{6}$$

where J_1, J_2 and J_m, are the fluxes at the two nodes and the mid-point of the element, respectively, we used the quadratic interpolation coefficients,

$$Q_1 = \frac{s}{l}\left(1 - \frac{2s}{l}\right), \ Q_2 = \frac{s}{l}\left(1 + \frac{2s}{l}\right)$$
$$Q_m = 1 - \left(\frac{2s}{l}\right)^2 \tag{7}$$

Writing the position and mass displacement in a column matrix δq and the velocity and mass flux in a column matrix q'.

$$\delta q = [\delta x_1 \quad \delta y_1 \quad \delta I_1 \quad \delta x_2 \quad \delta y_2 \quad \delta I_2 \quad \delta I_m]^T \tag{8}$$

$$\delta q' = [\acute{X}_1 \quad \acute{Y}_1 \quad J_1 \quad \acute{X}_2 \quad \acute{Y}_2 \quad J_2 \quad J_m]^T \tag{9}$$

We can write the integral Eq. 1 as

$$(\delta q)^T H q', \tag{10}$$

where

$$H = \frac{1}{6ml}\begin{vmatrix} 2l^2S^2 & -2l^2SC & 5lS & l^2S^2 & -l^2SC & -lS & -4lS \\ -2l^2SC & 2l^2C^2 & -5lC & -l^2SC & l^2C^2 & lC & 4lC \\ 5lS & -5lC & 14+4\mu & lS & -lC & 2-\mu & -16+2\mu \\ l^2S^2 & -l^2SC & lS & 2l^2S^2 & -2l^2SC & -5lS & 4lS \\ -l^2SC & l^2C^2 & -lC & -2l^2SC & 2l^2C^2 & 5lC & -4lC \\ -lS & lC & 2-\mu & -5lC & 5lC & 14+4\mu & -16+2\mu \\ -4lS & 4lC & -16+2\mu & 4lS & -4lC & -16+2\mu & 32+16\mu \end{vmatrix} \quad (11)$$

where we have used the shorthand notation $\mu = (ml^2)/(5M)$, $S = \sin\theta$, $C = \cos\theta$.

Since we need velocities for more than one element, we have to generate global H matrix, which is diagonal and includes matrix for each element. And it also has overlapped part for the same node of two adjacent elements. Here is the part of MATLAB code for global matrix.

```
if i==1
    for j=1:7
        for k=1:7
            H(j,k)=He(j,k);
        end
    end
else
    for j=0:2
        for k=0:2
            H(4*i+j-4,4*i+k-4)=He(j+1,k+1)+H(4*i+j-4,4*i+k-4);
        end
    end
    for j=0:2
        for k=4:7
            H(4*i+j-4,4*i+k-4)=He(j+1,k);
            H(4*i+k-4,4*i+j-4)=He(k,j+1);
        end
    end
    for j=4:7
        for k=4:7
            H(4*i+j-4,4*i+k-4)=He(j,k);
        end
    end
end
```

The equation (2) can be express in term of force on the element

$$\delta G = (\delta q)f \quad (12)$$

where

$$f = \gamma_s \begin{bmatrix} \cos\theta \\ \sin\theta \\ -\cos\theta \\ -\sin\theta \end{bmatrix} + \frac{l}{2} \begin{vmatrix} ((2/3)w^1 + (1/3)w^2)\sin\theta \\ -((2/3)w^1 + (1/3)w^2)\cos\theta \\ ((1/3)w^1 + (2/3)w^2)\sin\theta \\ -((1/3)w^1 + (2/3)w^2)\cos\theta \end{vmatrix} + \frac{l}{2}g \begin{bmatrix} \sin\theta \\ -\cos\theta \\ \sin\theta \\ -\cos\theta \end{bmatrix} \qquad (13)$$

Equating Eq. 10 and Eq. 12 yields

$$Hq' = f \qquad (14)$$

This equation is a set of linear algebraic equation for the generalized velocities. We use Matlab to solve for the velocity matrix. Note that taking an inverse of H is not possible since H is close to being a singular matrix. One way of suppressing the singularity is to add a small mass to the diagonal term of H matrix. In our program we use the command "psudoinverse", which also suppresses the singularity.

Once solved, the nodal velocities can be used to update the nodal positions. Here we use Euler method for time evolution

$$x_{n+1} = x_n + v_n \Delta t, \qquad (15)$$

where x_n is the current position and x_{n+1} is the new position. The accuracy of this method depends on the magnitude of time step. Since Euler method is an explicit method, the magnitude of time step has to be small compared with the magnitude of l. In our simulation we have $dt/l = 0.1$.

Strain Energy Density Calculation

Calculating w can be a cumbersome task. Here, we use a finite element package, Abaqus, to automatically calculate w. We use Matlab as a main program to evolve the morphology and only call Abaqus when strain energy calculation is required.

We model the effect of the misfit as a constant force applied at the edge of the film. This is equivalent to a film on a stiff and infinitely large substrate. This assumption is reasonable when the substrate is much thicker than the film.

The workflow is as followed.
   a) Matlab generates initial positions of the nodes on the surface and write the data to a position-file.
   b) Matlab automatically calls a DOS batch file. In this batch file, Abaqus was called to read the position-file and a pre-programmed input file. It then generates meshes over the entire domain and, with specified boundary force, calculates the strain energy. Note that we add additional two 10×10 grids on both sides of the film so that the area of interest is far from the boundary. This avoids error due to edge effect.
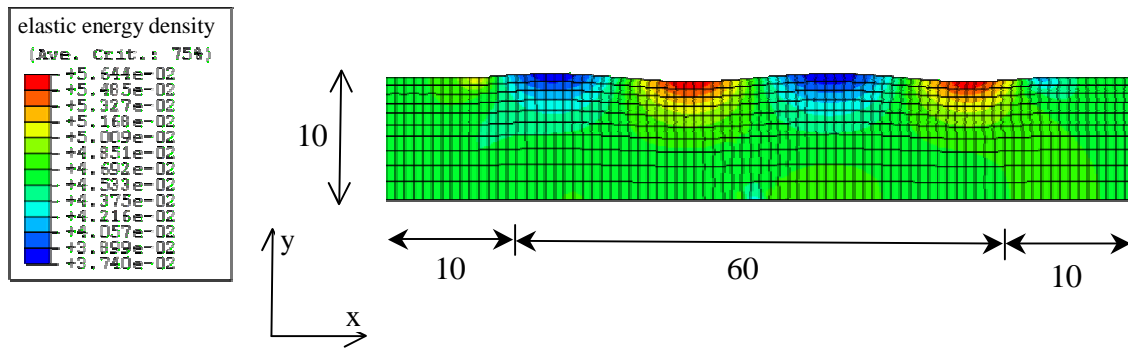
Fig. 2. Calculation domain and meshes generated by Abaqus. Our calculation domain is 60x10. We attached additional 10x10 grid to avoid edge effect. The color specifies the magnitude of the strain energy density.

c) Abaqus then writes the strain energy associated with the nodes in the surface to the data file.
d) Matlab reads the data file and evolves the node on the surface and write the updated nodal position to the new position-file.
e) Repeat step b)

In our simulation we didn't need to recalculate the strain energy density at every time step. If the position doesn't change too much, the strain energy from the previous step is still valid. We found that calculate strain energy at every 10 time steps is a compromise between speed and accuracy.

The command for Matlab to call batch file is:

system('callabaq.bat')

The command in the batch file to call Abaqus is:

CALL abaqus j=elasden.inp int,

where elasden.inp is the name of the pre-programmed input file.

Results and Discussions

From Eq. 2, we have four contributions to the reduction of the free energy, which are surface energy, evaporation/condensation, elastic strain energy, and bulk free energy. Therefore, we would like to study each effect separately and we can combine them to simulate the quantum dot.
To achieve this, we set the coefficient in front of term of interest in Eq. 13 to be non zero while multiplying other term with zero. However, to suppress the effect of evaporation-condensation, we have to further set $m\lambda^2/M \ll 1$, which make the effect of surface diffusion more prominent.

1. *Stress-free undulating surface under surface tension*

In this simulation, we studied the effect of the surface energy. We set the initial perturbation, $A = 0.03\sin(kx)$, no applied force, and $g = 0$.
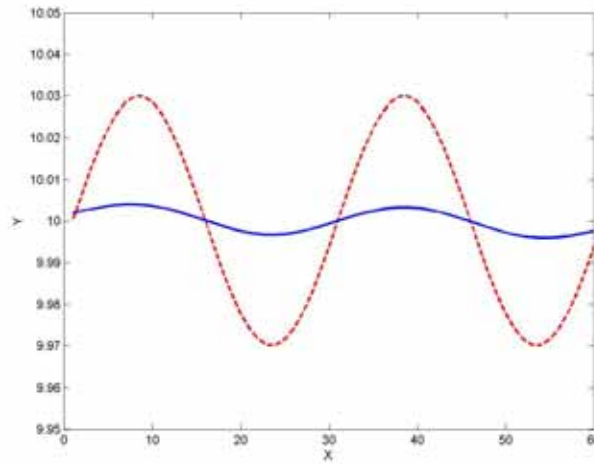

Fig. 3. The shape of the surface of the thin film. The red line indicate the original perturbation and the blue line indicate the shape after 500 iterations

Fig. 3 shows that with only surface energy contribution, the lowest energy state is the flat film. This is expected because the flat film has the lower surface area.

2. *Stressed undulating surface without surface tension*

In this simulation, we study the effect of stress by setting the perturbation, $A = 0.03\sin(kx)$. We added a constant force on the two ends and set $g = 0$, and $\gamma_s = 0$.
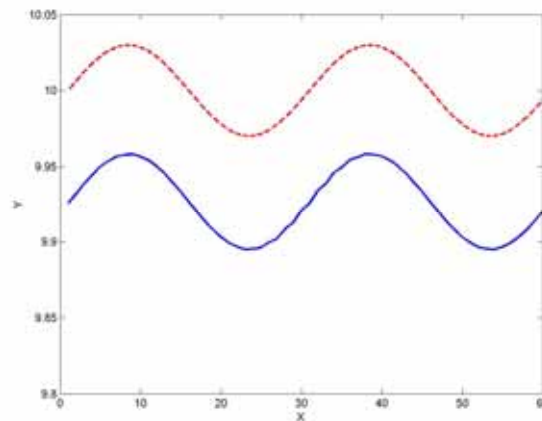

Fig. 4. The shape of the surface of the thin film. The red line indicate the original perturbation and the blue line indicate the shape after 500 iterations

From Fig. 4, the surface moves down almost uniformly. There exist a small amplitude increase but it is small and we are not sure whether this is due to the numerical error.

The face that the surface moves down is reasonable. When including stress, the energy of atoms in the solid becomes higher relative to that in the environment. Therefore, evaporation becomes more significant.

3. *Stressed undulating surface under surface tension*

In this simulation, we study the competition between stress and surface tension contributions. We used the same initial surface profile with both misfit and surface tension term non-zero. We set $g = 0$.
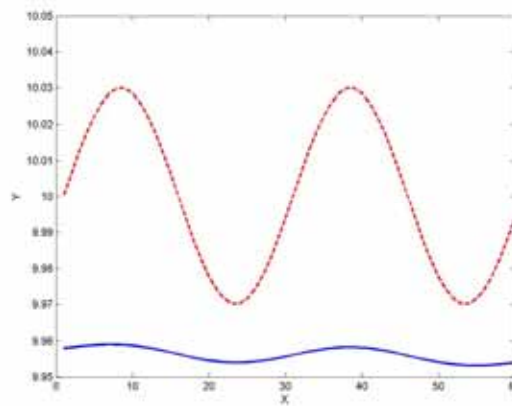

Fig. 5. The shape of the surface of the thin film. The red line indicate the original perturbation and the blue line indicate the shape after 500 iterations

Fig. 5. Shows that the surface moves downward as well as reduce the amplitude, which come from stress effect and surface tension effect, respectively.

4. *Undulating surface with phase difference only*

The remaining parameter is the bulk free energy, *g*. We use the same surface profile while setting the surface tension and stress contributions to zero. We experimented the value of g from negative to positive.

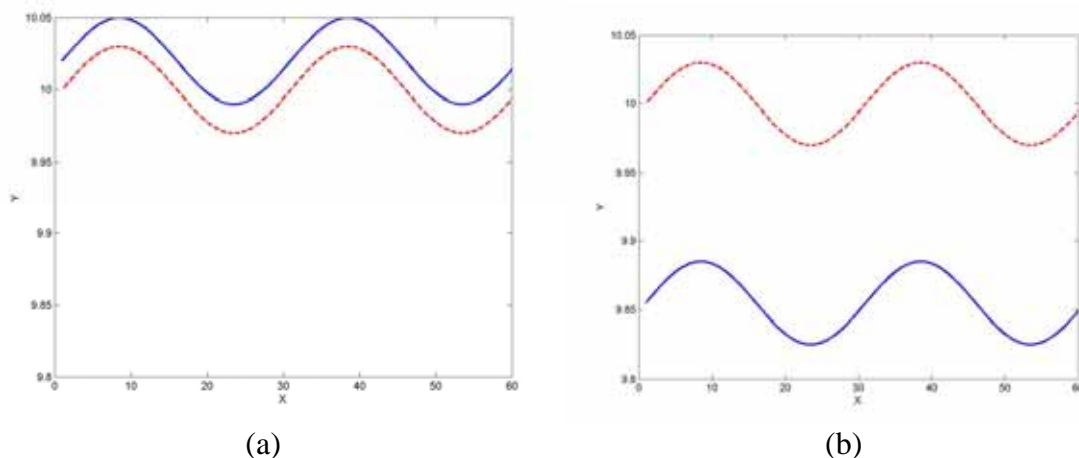(a)   (b)

Fig. 6. The shape of the surface of the thin film. The red line indicate the original perturbation and the blue line indicate the shape after 300 iterations. (a) g<0 (b) g>0.

Fig. 6. shows that the sign of g determine the movement of the surface. When g<0, the surface has lower bulk energy than that of the vapor and therefore, it is energetically favorable for the vapor phase to condense to the solid phase as in Fig. 6(a). The situation is the opposite in Fig. 6(b).

Recalling Eq (2), the free energy variation can be written as:

$$\delta G = \int \left(w + \gamma_s K + g \right) \delta r_n dA \qquad (16)$$

Because the free energy variation is associated with unit volume of solid grown on the surface, define a driving force:

$$p = -w - \gamma_s K - g \qquad (17)$$

Then, if $p > 0$, the solid gains mass from the environment, causing upward movement. If $p < 0$, the solid loses mass to the environment, resulting in downward movement. As a result, we can balance choose w and g to switch between cases where solid evaporates, vapor condensates, or no phase change occurs [6].

   5. *Stressed undulating surface with diffusion, and phase difference, stress dominated movement*

In this simulation, we increased the magnitude of stress contribution and adjust g to prevent phase change.
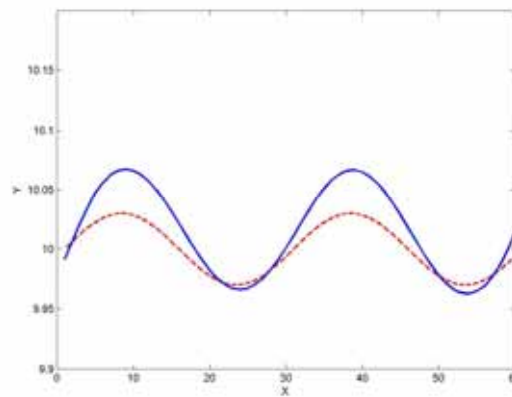

Fig. 6. The shape of the surface of the thin film. The red line indicate the original perturbation and the blue line indicate the shape after 500 iterations.

Fig. 6. shows the that with higher stress contribution, the amplitude of the film increases. The value of *g* was adjusted to prevent evaporation. This is the analogous to the formation of quantum dots.

Considering a dimensionless parameter which characterizing the relative significance of the elastic and surface energy:

$$s = \frac{\sigma_0^2 \lambda}{E \gamma_s} \tag{18}$$

If we set s large enough, the stress effect will dominate the surface movement, the amplitude of surface will increase. In this simulation, $s = 15$.

6. *Stressed undulating surface with diffusion and phase difference, diffusion dominated movement:*

If s is small enough, the surface tension dominate the surface movement, eventually, the surface will become flat. In this simulation, $s \approx 0.15$.
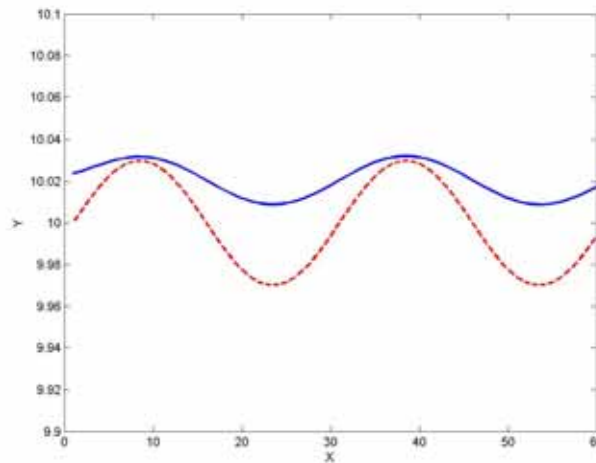

Fig. 7. The shape of the surface of the thin film. The red line indicate the original perturbation and the blue line indicate the shape after 900 iterations.

7. *Stressed rough surface with diffusion, and phase difference*

In experiments, quantum dots form by vapor depositing film material on the substrate. Any seemingly flat substrate is rough when looking at the atomic scale. Therefore, we would like to use simulate the quantum dot from a rough substrate. We introduce random perturbation on the surface and use the same parameter as that of the previous simulation.
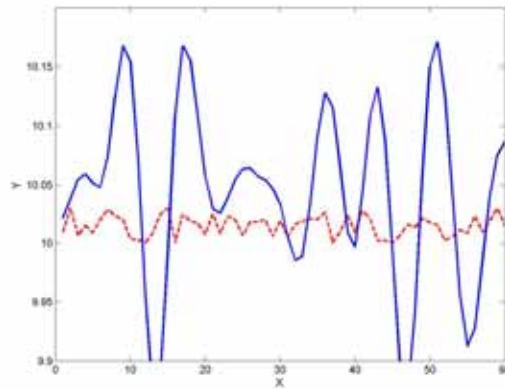
Fig. 8. The shape of the surface of the thin film. The red line indicate the original perturbation and the blue line indicate the shape after 500 iterations.

From Fig. 8, the amplitude of the surface increase, resulting in formation of 6 dots. The dots still look very rough since this is still very early in the evolution. Nevertheless, the simulation shows that the simulation of quantum dots is possible.

We summarized the important contributions in each simulation in the table below.

Runs	Diffusion	Stress	Phase difference
1	*Yes*	*No*	*No*
2	*No*	*Yes*	*No*
3	*Yes*	*Yes*	*No*
4	*No*	*No*	*Yes*
5	*Yes*	*Yes (dominating)*	*Yes*
6	*Yes(dominating)*	*Yes*	*Yes*
7 (random Perturbation)	*Yes*	*Yes (dominating)*	*Yes*

Table 1. Summary of significant contribution in each simulation

Summary

We have employed the finite element method to study the contribution from the surface tension, the strain energy, and the bulk free energy as well as simulated quantum dot formation. With the surface tension dominating, the system will try to reduce the energy by reducing the surface area, which results in a flat surface. The role of the strain energy is to promote an increase in the amplitude of modulation. At the same time, the stress affects an evaporation rate, causing the surface to move downward. In order to simulate the quantum dot formation, the downward movement is counterbalanced by the bulk energy term which promotes the formation the solid phase from the vapor phase. We show that it is possible to simulate the quantum dot behavior although longer simulation time is needed.

Future Work

1) In our simulation we only study each parameter without making a connection to physical parameters. It is very useful to employ non-dimensionalization scheme in order to better simulate experimental system.
2) We could improve our time evolution scale by using an implicit method. One efficient method is Runge-Kutta which is 4th-orderd accurate.
3) Our program doesn't have flexibility in eliminating note or adding node to properly represent the surface. This adaptation scheme is useful when simulating for a long time.
4) If there is electrical current going across the film, the currency-induced electromigration will change the shape of surface. Simulating surface change due to electromigration by using the combination of Abaqus and Matlab will be an interesting topic. This will help us understand the interaction effects among surface diffusion, elastic stressing and electrical current stressing.
5) We could also look at the case where the surface energy is dependent on surface orientation. This results in facets in which are present in many quantum dot systems.

Reference

[1] P. Liu, Y. W. Zhang, and C. Lu, Phys. Rev. B 68 (2003).
[2] Y. Ni, A. K. Soh, and L. H. He, Journal Of Crystal Growth 269, 262 (2004).
[3] S. M. Wise, J. S. Lowengrub, J. S. Kim, and W. C. Johnson, Superlattices Microstructures 36, 293 (2004).
[4] B. Sun, Z. Suo, and W. Yang, Acta Materialia 45, 1907 (1997).
[5] B. Sun and Z. Suo, Acta Materialia 45, 4953 (1997).
[6] J. H. Prevost, T. J. Baker, J. Liang, and Z. Suo, Int. J. Solids Structures 38, 5185 (2001).