SPIN COATING SIMULATION USING FINITE ELEMENT METHOD

by

Ramah Sutardja

Memorandum No. UCB/ERL M88/70

14 November 1988

SPIN COATING SIMULATION USING FINITE ELEMENT METHOD

by

Ramah Sutardja

Memorandum No. UCB/ERL M88/70

14 November 1988

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

SPIN COATING SIMULATION USING FINITE ELEMENT METHOD

by

Ramah Sutardja

Memorandum No. UCB/ERL M88/70

14 November 1988

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

SPIN COATING SIMULATION USING FINITE ELEMENT METHOD

by

RAMAH SUTARDJA

ABSTRACT

A spin coating simulator is developed and incorporated into the general purpose process simulator CREEP. The spin model is based on the mechanics of viscous creep-flow and it's finite element formulation is solved in cylindrical coordinates. The analysis also accounts for inertia and surface tension effect. Nonlinear terms in the Navier-Stokes equations are included to account for the eddy flow in the fluid. It is found that evaporation has an important role in determining the final film thickness. At present, Newtonian fluids are being studied.

Acknowledements

I wish to thank Professor William G. Oldham for his encouragement and support throughout the course of this work. Discussions with Pantas Sutardja, who is the author of the general purpose simulator program called CREEP, helped tremendously. Special thanks to Dr. Yosi Shacham and the staff in our microfabrication lab, especially Thomas Booth, Katalin Voros, Marilyn Krushner and Robin Rudell, for their technical help in the use of the laboratory facilities.

Special thanks goes to Ginetto Addiego for proofreading the initial version of this report. Last, but not least, I would like to thank the Semiconductor Research Corporation for its financial support during the course of this project.

Table of Contents

Chapter 1

Introduction

Planarization is a factor limiting the refinement of feature resolution (critical dimension). The increasing importance of planarization has fostered many studies of the planarizing capabilities of commercial photoresist and and polyimide. This project addresses the problem of spin-on coating profiles over arbitrary topography. Most theories of spin-on film thickness deal only with flat surfaces. However, there is no literature on the simulation of spin-on resist over steps and irregular topography.

Several spin models have been proposed in recent years. For example, L. K. White [1] approximates spin-on film on complex topography by considering the spin-on film as a low pass frequency filter. However, the first detailed hydrodynamic analysis of spin coating of a Newtonian liquid was given by Emslie *et al* [2] They assumed that the local centrifugal force per unit volume is uniform across the thinning film and is balanced solely by viscous shear across the film due to the radial liquid flux. These assumptions are valid when the Reynolds number $Re = \rho VH / \mu$ is much less than unity (H being a characteristic length scale for the thinning film, μ the fluid viscosity, ρ the fluid density, and V the characteristic velocity of the fluid. However, their work focussed mainly on films over flat substrate surfaces. Our project is to study the spin problem over any kind of substrate topography.

The formulation of our model is based on the mechanics of viscous creeping flow and it is solved in cylindrical coordinates using the finite element method. We have also included the nonlinear terms in the Navier-Stokes equations to account for the eddy flow in the fluid. The parameters which have been included in the model are:

1) spin time

2) spin speed

3) film viscosity

4) film density

5) surface tension of the film

Evaporation is not included in the model. However, it can be readily incorporated by assuming certain volumetric shrinkage rate of the film during spinning or after spinning.

Chapter 2

Modeling of the Spin Process

Photoresist may be modeled as a viscous fluid. Thus we formulated the spin problem using the viscous fluid model. The virtual work principle is used for the finite element formulation of the problem. The following formulation is very similar to the one derived by P. Sutardja[3] for oxidation.

1. Virtual Work Principle

Let the primary variable of the flow equation be the velocity. The following is the statement of the virtual work principle.

> The total work done by an arbitrary (infinitesimal) variation of velocities about the actual values, assuming all the external and internal forces being held unchanged, should equate to zero.

In mathematical form, this is given by

$$W_e = W_i \tag{1}$$

where

$$W_e = \int_{\Gamma_t} \mathbf{f}\cdot\delta\mathbf{v}\, d\Gamma + \int_{\Omega} \mathbf{b}\cdot\delta\mathbf{v}\, d\Omega\,, \tag{2}$$

$$W_i = \int_{\Omega} \delta\dot{\varepsilon}\cdot\sigma\, d\Omega\,, \tag{3}$$

Ω = body of interest,

Γ_t = portion of boundary where surface traction is applied,

$\mathbf{f}$ = surface traction on Γ_t ,

$\mathbf{b}$ = body force per unit volume, including acceleration effects,

$\mathbf{v}$ = velocity,

$\dot{\varepsilon}$ = strain rate (appropriate for velocity formulation),

σ = stress in the body.

In terms of matrix notation, this can be written as

$$\int_{\Gamma_t} \delta\mathbf{v}^T\mathbf{f}\, d\Gamma + \int_{\Omega} \delta\mathbf{v}^T\mathbf{b}\, d\Omega = \int_{\Omega} \delta\dot{\varepsilon}^T\sigma\, d\Omega \tag{4}$$

Before we start the discretization process, we set $f=0$ initially. By considering the problem as axisymmetric, we then have (in cylindrical coordinates),

$$\mathbf{v}=\begin{bmatrix} v_r \\ v_z \end{bmatrix} \quad , \quad \mathbf{b}=\begin{bmatrix} b_r \\ b_z \end{bmatrix} \quad , \quad \dot{\boldsymbol{\varepsilon}}=\begin{bmatrix} \dot{\varepsilon}_{rr} \\ \dot{\varepsilon}_{\theta\theta} \\ \dot{\varepsilon}_{zz} \\ \dot{\gamma}_{rz} \end{bmatrix} \quad , \quad \boldsymbol{\sigma}=\begin{bmatrix} \sigma_{rr} \\ \sigma_{\theta\theta} \\ \sigma_{zz} \\ \sigma_{rz} \end{bmatrix} \tag{5}$$

In our model, we have $b_r = \rho\omega^2 r$. We assume $b_z = 0$.

2. Constitutive Relations

The stress and strain vectors can each be split into a shear and compressive component:

$$\boldsymbol{\sigma}=\boldsymbol{\sigma}'+\boldsymbol{\sigma}'' \tag{6}$$

$$\dot{\boldsymbol{\varepsilon}}=\dot{\boldsymbol{\varepsilon}}'+\dot{\boldsymbol{\varepsilon}}'' \tag{7}$$

where

$$\boldsymbol{\sigma}'=\begin{bmatrix} \sigma'_{rr} \\ \sigma'_{\theta\theta} \\ \sigma'_{zz} \\ \sigma_{rz} \end{bmatrix} \quad , \quad \boldsymbol{\sigma}''=\begin{bmatrix} \sigma_p \\ \sigma_p \\ \sigma_p \\ 0 \end{bmatrix} \quad , \text{and} \tag{8}$$

$$\dot{\boldsymbol{\varepsilon}}'=\begin{bmatrix} \dot{\varepsilon}'_{rr} \\ \dot{\varepsilon}'_{\theta\theta} \\ \dot{\varepsilon}'_{zz} \\ \dot{\gamma}_{rz} \end{bmatrix} \quad , \quad \dot{\boldsymbol{\varepsilon}}''=\frac{1}{3}\begin{bmatrix} \dot{\varepsilon}_v \\ \dot{\varepsilon}_v \\ \dot{\varepsilon}_v \\ 0 \end{bmatrix} \quad , \tag{9}$$

where

$$\sigma_p = 1/3\,(\sigma_{rr}+\sigma_{\theta\theta}+\sigma_{zz}) \quad , \quad \sigma'_{rr}=\sigma_{rr}-\sigma_p \quad , \quad \sigma'_{\theta\theta}=\sigma_{\theta\theta}-\sigma_p \quad , \quad \sigma'_{zz}=\sigma_{zz}-\sigma_p \quad , \text{and} \tag{10}$$

$$\dot{\varepsilon}_v = 1/3\,(\dot{\varepsilon}_{rr}+\dot{\varepsilon}_{\theta\theta}+\dot{\varepsilon}_{zz}) \quad , \quad \dot{\varepsilon}'_{rr}=\dot{\varepsilon}_{rr}-\dot{\varepsilon}_v \quad , \quad \dot{\varepsilon}'_{\theta\theta}=\dot{\varepsilon}_{\theta\theta}-\dot{\varepsilon}_v \quad , \quad \dot{\varepsilon}'_{zz}=\dot{\varepsilon}_{zz}-\dot{\varepsilon}_v \tag{11}$$

It can be easily recognized that $\sigma_p = -P = -$pressure and $\dot{\varepsilon}_v$ = volumetric strain rate. We shall now make the assumption that the fluid is incompressible. Hence $\dot{\varepsilon}_v = 0$ and $\dot{\boldsymbol{\varepsilon}} = \dot{\boldsymbol{\varepsilon}}'$.

The stress - strain relationship for viscous fluid is given by

$$\boldsymbol{\sigma}'=\eta\mathbf{D}\dot{\boldsymbol{\varepsilon}}'=\eta\mathbf{D}\dot{\boldsymbol{\varepsilon}} \tag{12}$$

where η is the coefficient of viscosity and

$$\mathbf{D}=\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

We can now rewrite the constitutive relation for incompressible viscous flow as

$$\sigma = \sigma' - \mathbf{m}P = \eta \mathbf{D}\dot{\varepsilon} - \mathbf{m}P \tag{14}$$

where $\mathbf{m}^T = [\,1 \;\; 1 \;\; 1 \;\; 0\,]$.

3. Discretization

The discretization can now proceed. Let

$$\mathbf{v} = \mathbf{N}\mathbf{V} \tag{15}$$

where

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{v}_n \end{bmatrix} \quad , \qquad \mathbf{v}_i = \begin{bmatrix} v_{ir} \\ v_{iz} \end{bmatrix} \tag{16}$$

$$\mathbf{N} = [\mathbf{N}_1, \ldots\ldots, \mathbf{N}_n] \quad , \qquad \mathbf{N}_i = \begin{bmatrix} N_i(x,y) & 0 \\ 0 & N_i(x,y) \end{bmatrix} \tag{17}$$

Now, the $\mathbf{N}_i$'s are 2×2 matrices since v has r and z components. Hence we have

$$\delta \mathbf{v}^T = \delta \mathbf{V}^T \mathbf{N}^T \tag{18}$$

From small strain analysis,

$$\dot{\varepsilon} = \mathbf{L}\mathbf{v} \tag{19}$$

where L is the strain operator:

$$\mathbf{L} = \begin{bmatrix} \frac{\partial}{\partial r} & 0 \\ \frac{1}{r} & 0 \\ 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial z} & \frac{\partial}{\partial r} \end{bmatrix} \tag{20}$$

We can also write $\dot{\varepsilon} = \mathbf{B}\mathbf{V}$ where B=LN. Thus we have

$$\delta \dot{\varepsilon}^T = \delta \mathbf{V}^T \mathbf{B}^T \tag{21}$$

Applying the finite element approximation to the virtual work principle, and factoring out $\delta \mathbf{V}^T$, we get,

$$\delta \mathbf{V}^T \int_\Omega \mathbf{B}^T \sigma \, d\Omega = \delta \mathbf{V}^T \int_\Omega \mathbf{N}^T \mathbf{b} \, d\Omega \tag{22}$$

Since this is true for arbitrary variation $\delta \mathbf{V}$,

$$\int_\Omega \mathbf{B}^T(\sigma' - \mathbf{m}P)\, d\Omega = \int_\Omega \mathbf{N}^T \mathbf{b}\, d\Omega \tag{23}$$

$$\int_\Omega \mathbf{B}^T(\eta \mathbf{D}\dot{\varepsilon} - \mathbf{m}P)\, d\Omega = \int_\Omega \mathbf{N}^T \mathbf{b}\, d\Omega \tag{24}$$

Discretization is not complete if **P** is not discretized. If we discretize **P** in a similar manner as we did for the velocities, we wil have n unknowns for **P**. There will then be more variables than equations set up ($2n$ equations vs $3n$ unknowns). Indeed the missing equation is the incompressibility condition:

$$\dot{\varepsilon}_v = 0 \tag{25}$$

We can set up another n equations by premultiplying equation (17) by δP and integrating over Ω. However, this will result in a $3n \times 3n$ matrix with n zero diagonal entries, which is undesirable. We shall instead try to eliminate the pressure term to form 2n equations in 2n unknowns.

4. Penalty Function Method

A way of eliminating the pressure term is by introducing a limiting constraint of the form

$$P = -\alpha\dot{\varepsilon}_v \tag{26}$$

It is conceivable that if α is infinitely large, than $\dot{\varepsilon}_v$ must be zero for P to have any finite value. We shall use a large value for α in our simulation to approximate the incompressibility condition.

We can now write

$$\dot{\varepsilon}_v = \frac{v_r}{r} + \frac{\partial v_r}{\partial r} + \frac{\partial v_z}{\partial z} = \mathbf{m}^T\dot{\varepsilon} \tag{27}$$

Hence, we have

$$\int_\Omega \mathbf{B}^T(\eta \mathbf{D}\dot{\varepsilon} + \alpha \mathbf{m}\mathbf{m}^T\dot{\varepsilon})\, d\Omega = \int_\Omega \mathbf{N}^T \mathbf{b}\, d\Omega \tag{28}$$

$$\left[\int_\Omega \mathbf{B}^T(\eta \mathbf{D}\mathbf{B} + \alpha \mathbf{m}\mathbf{m}^T\mathbf{B})\, d\Omega\right]\mathbf{V} = \int_\Omega \mathbf{N}^T \mathbf{b}\, d\Omega \tag{29}$$

or

$$[\,\mathbf{K}_v + \mathbf{K}_p\,]\mathbf{V} = \mathbf{F} \tag{30}$$

where

$$\mathbf{K}_v = \eta \int_\Omega \mathbf{B}^T\mathbf{D}\mathbf{B}\, d\Omega \tag{31}$$

$$\mathbf{K}_p = \alpha \int_\Omega \mathbf{B}^T\mathbf{m}\mathbf{m}^T\mathbf{B}\, d\Omega \tag{32}$$

$$\mathbf{F} = \int_\Omega \mathbf{N}^T\mathbf{b}\, d\Omega \tag{33}$$

We thus have $2n \times 2n$ matrix equation of the form

$$\mathbf{KV} = \mathbf{F} \tag{34}$$

The above formulation is known as the penalty function approach in the finite element literature. The resulting matrix **K** is positive definite and can be solved by direct Gaussian elimination.

5. Justification of the Model.

The first order analytical approximation of the thickness of a film with time for the Newtonian fluid model is given by[4]

$$z_s = \sqrt{\frac{1}{\frac{1}{z_{s0}^2} + \frac{4}{3\eta}\rho\omega^2 t}} \tag{35}$$

where z_{s0} is the initial height at the surface of the film, and z_s is the height at the surface of the film after time t.

To verify our formulations, we simulated the spinning of a film on a flat surface and compared the results with eq. 36. Thickness data were obtained in a period from 0.05 sec to 2.2 sec. The log-log plot of the simulation results and the analytical solutions are shown in Fig. 1. Agreement is excellent.

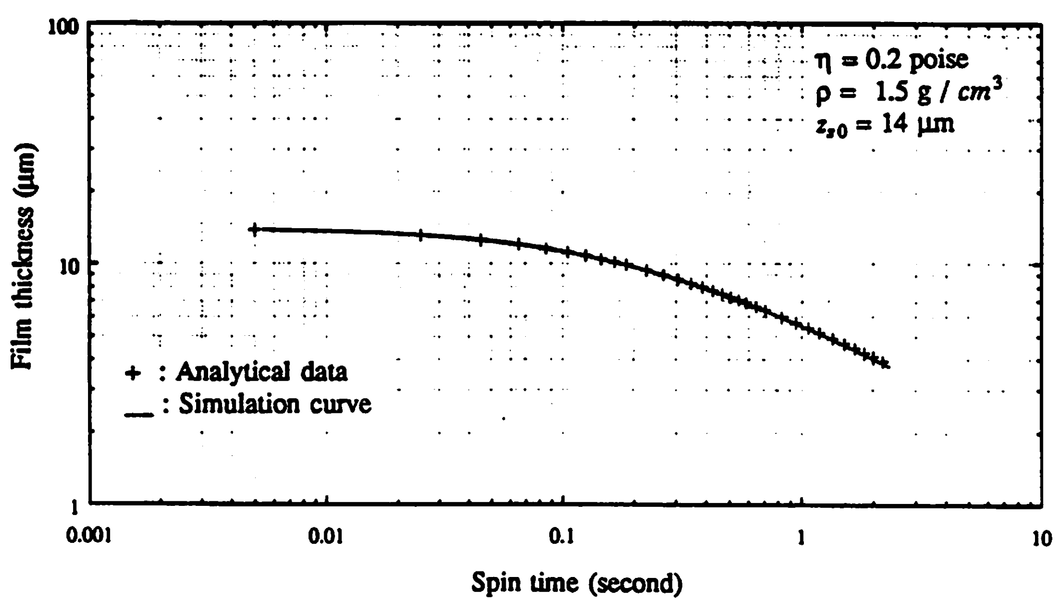


Fig. 1. Log-log plot of simulation results and analytical solutions.

Chapter 3
Boundary Conditions

The number of mesh points used has critical effect on the simulation time. It would be ideal to simulate only the region close to the topographic features, with artificial boundary conditions imposed on both ends of the domain as shown in Fig. 1.

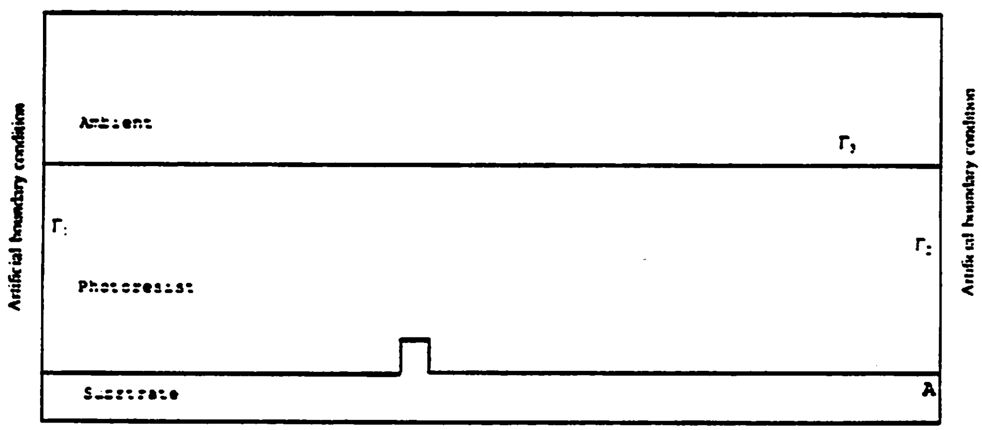


Fig. 1. Artificial boundary conditions imposed on both ends of the simulation domain.

Stillwagon *et al* [5] show that spin coating produces conformal film profiles over gaps on the substrate with widths greater than about 50 μm. Therefore, the regions sufficiently far away from the topographic features should behave like that of a flat substrate. We tried a number of methods in imposing the boundary conditions. One method is using the first order differential equation governing creep flow over a flat substrate:

$$\frac{d\sigma_{rz}}{dz} = -b_r \tag{1}$$

where σ_{rz} is the shear stress along the r-z plane, and b_r = force per unit volume in the r direction.

From this equation, we can derive the radial and the vertical velocities at any given height (z) and radius (r) in the fluid. The respective equations for the radial and the vertical velocities are

$$v_r = \frac{b_r}{2\eta}(-z + 2z_s)z \tag{2}$$

$$v_z = \frac{\rho\omega^2}{\eta}(z_s z^2 - z^3/3) \tag{3}$$

where z_s is the height at the surface, and η is the viscosity of the fluid.

This method was tried but it does not give satisfactory results. We believe it is because the slight discrepancies between the solutions on a flat substrate and the actual simulated domain produces errors which propagate throughout the fluid after several time steps.

Another method we have tried is fixing the boundary by imposing the first order approximation of force vectors along the cross-sectional surface. Unfortunately, this method has the same problem as the previous method.

Let us now consider the boundary Γ_2 in figure 1, with no boundary conditions imposed on it. As the fluid flow is pinned at point A, the simulation result will show the rotation of fluid around that point (Fig. 2). We are not interested in this region. To eliminate this problem, we can use a coarse mesh at this region. A coarse mesh will stiffen the structure by not allowing this kind of rotation. It gives us good planar solutions for the boundary Γ_3. in Fig. 1. As the new version of the mesh generator is incorporated into our program, we are able to dissect the simulated region into three regions of different densities (Fig. 3), and thus we can afford to increase the mesh density near the region of interest and use a very coarse mesh for the rest of the regions.

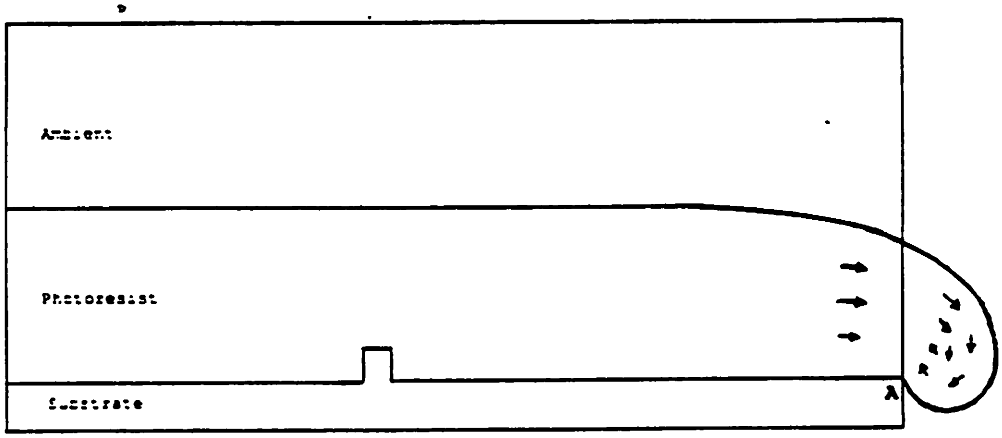


Fig. 2. Rotation of fluid around point A.

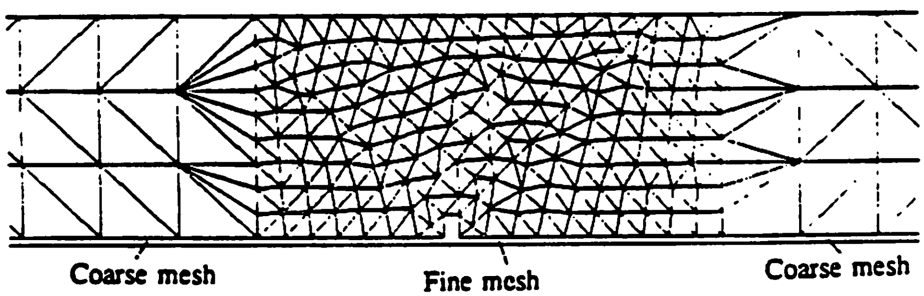


Fig. 3. The simulation domain is divided into 3 regions of different mesh densities.

Chapter 4

Reynold's Number and Numerical Instabilities

The Navier-Stokes equations describing the viscous flow is given by:[6]

$$\rho\frac{\partial u}{\partial t}+\rho(u\frac{\partial u}{\partial r}+v\frac{\partial u}{\partial z})=-\frac{\partial P}{\partial r}+\mu(\frac{\partial^2 u}{\partial r^2}+\frac{\partial^2 u}{\partial z^2}) \tag{1}$$

$$\rho\frac{\partial v}{\partial t}+\rho(u\frac{\partial v}{\partial r}+v\frac{\partial v}{\partial z})=-\frac{\partial P}{\partial z}+\mu(\frac{\partial^2 v}{\partial r^2}+\frac{\partial^2 v}{\partial z^2}) \tag{2}$$

where u and v are the velocities in the r and z direction respectively, and P is the fluid pressure. In the above equation, an incompressible fluid with no body force has been assumed.

When the Reynold's number is high, eddy flow[7] may occur, especially at large radii where the velocities are high. Consequently, the non-linear convective terms (the second terms at the left hand side of Equation 1 and Equation 2) may not be neglected.

As a rough guide, when the Reynold's number is much smaller than one, it is considered to have a low value. To include the above nonlinear terms in our simulation, Newton-Ralphson's iterative method is employed.

Furthermore, the solutions of the Newton-Ralphson's iterative method may not converge when the Reynold's number is high. One way of overcoming this problem is to use smaller size of finite elements. However, this method may be too costly in terms of computer time. A method which does not require finer mesh is the *upwind method*, which will not be discussed here. However, when the Reynold's number is larger than a critical value, the fluid flow become turbulent. Our model is not able to predict this kind of behaviour.

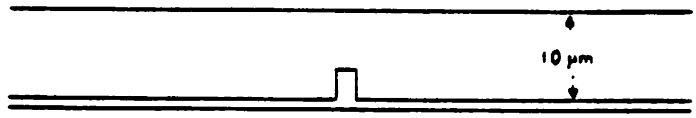
Chapter 5

Results

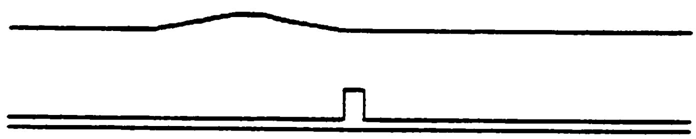
When a polymer film is dispensed on a wafer in a spin process, a layer of film with a thickness of several hundred micrometers is formed initially. To simulate the process starting from several hundred micrometers in order to study a structure with a dimension of a few micrometers, it will take the computer extremely long time to finish the simulation. This is very impractical and costly. By running many simulations and ignoring surface tension effect, we verified that the final profile is independent of the initial thickness of the film. Furthermore, the initial profile does not affect the final profile as long as sufficiently long simulation time has elapsed (see Fig. 1).. In Fig. 1(a), 1(b) and 1(c), the same values of spin speed, film viscosity and film density are used. The final profiles obtained from these three cases coincide after approximately 0.04 second (see Fig. 1(d)). This is in agreement with the first order equation of spinning of a flat film (eq. 2.35), which is:

$$z_s = \sqrt{\frac{1}{\frac{1}{z_{s0}^2} + \frac{4}{3\eta}\rho\omega^2 t}} \qquad (1)$$

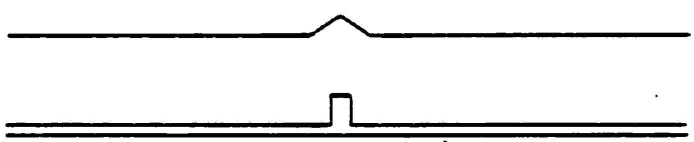
Notice that the term involving the initial height (z_{s0}) is generally negligible compared to the term involving viscosity and density, and that surface tension plays no role on planar surfaces.



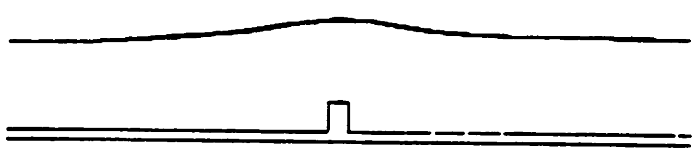
(a) Initial profile



(b) Initial profile



(c) Initial profile



(d) Final profile after a fraction of a second.

Fig. 1. Final profile is independent of initial profile after sufficiently long time.

1. Simulation of Profiles over Example Topographic Features

Polymers with Negligible Surface Tension

Figures 2, 3, 4 and 5 show the simulation results over various features. All simulations have the spin axis on the left edge of the profile. The following data are for the simulations performed for these figures.

Spin speed: 4700 rpm

Film density: 1.5 g/cm^3

Film viscosity: 0.27 poise (dyne-second/cm^2)

Simulations of step topography are shown in Figures 6 and 7. All features are 1 cm away from the center of the wafer. The input files of the above examples can be found in Appendix D.

b) Polymers with Surface Tension Value (~30 - 50 dyne/cm)

Simulation of spin-on polymer films with surface tension value between 30 dyne/cm and 50 dyne/cm were performed. The initial profiles are similar to those in Figures 2, 3, 4 and 5. The simulation results show that the final profiles are level after one-tenth of a milli-second.

The above simulation suggests that the surface tension effect overwhelms the centrifugal flow of polymer during spinning. In order to confirm the hypothesis, we performed the following two simulations.

First, a layer of polymer with an initial profile as shown in Fig. 8(a) is spun until a pseudo-steady state profile is obtained. Surface tension is ignored here. Fig. 8(b) shows the pseudo-steady state profile. Let H be the change in height as shown in Fig. 8(b). We define the time constant, τ_1, to be the time taken for the maximum height to reach $1/e$ (e = 2.7182818...) of the pseudo-steady state maximum height. Let the change in the maximum height after time τ_1 be δH.

Second, a surface tension value of 40 dyne/cm is introduced into the simulation of the profile shown in Fig. 8(b). The polymer is allowed to relax under the effect of surface tension until the film profile is almost flat (see Fig. 8(c)). Another time constant, τ_2, is defined here as the time taken for a change of height δH (loss in height) to occur.

From simulations, we obtained the following values:

$\tau_1 \sim 0.05$ second

$\tau_2 \sim 3 \times 10^{-6}$ second

We also estimated τ_1 and τ_2 for the simple feature shown in Fig. 8. We believe τ_1 should be on the order of S/V, and τ_2 on the order of $S\eta/\gamma$, where S is the feature width, V the average velocity at the top of the feature, η the viscosity of the polymer film, and γ the surface tension coefficient of the film. Using the values $S = 1 \times 10^{-4}\ cm$, $V = 1.4 \times 10^{-3}\ cm/sec$, $\eta = 0.27\ poise$, and $\gamma = 40\ dyne/cm$, we obtain,

$\tau_1 \sim 0.07$ second

$\tau_2 \sim 7 \times 10^{-7}$ second

The analytical predictions for the time constant do not differ significantly from the observation from simulation. Notice that τ_1 is four orders of magnitude larger than τ_2. Thus we conclude that surface tension is the dominant effect on the film profile during spinning.

c) Shrinkage

In practice, a flat film profile is never seen from photographs obtained from scanning electron microscopy (SEM). Therefore we suspect that shrinkage may be responsible for the final film profiles. Fig. 9 shows the simulation result obtained from shrinkage with no spinning.

2. Laboratory Experiment

The main difficulty of the experiment is to preserve the resist profile before taking SEM photographs. In order to take SEM photographs, the sample has to be hard-baked. However, hard bake will cause considerable reflow of the resist. It is learned that photoresist will harden due to the cross-linking of the molecules when exposed to ultra-violet (UV) radiation. We used this hardening method in the laboratory.

First, we spun a layer of KTI 820 photoresist (27 centi-poise, 32.5 % solute) on a wafer with some topography on it. Then we baked the wafer as short time as possible to avoid reflow of the resist (at 120 degrees Celcius for 1 minute). The wafer was then put into the nitrogen plasma chamber. Nitrogen plasma is known to produce UV radiation without etching the photoresist. The photoresist was UV cured for 10 minutes.

We examined the SEM photographs of the topographic features (single bump and double bumps) 0.75 cm away from the center of the wafer (Fig. 10 and Fig. 11). The corresponding simulation results (assuming shrinkage determines the final profile resist profile) are shown in Fig. 12 and Fig. 13. The simulation results look very similar to the laboratory results

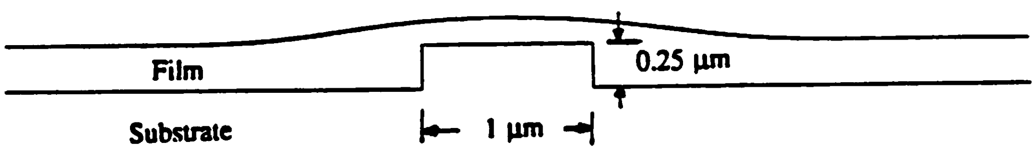


Fig. 2. A bump.

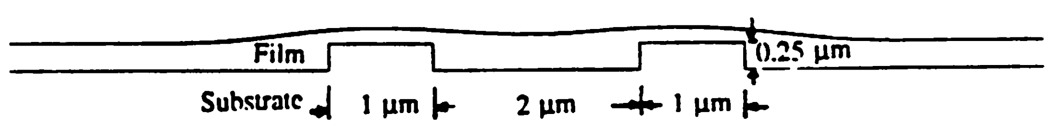


Fig. 3. Two bumps.

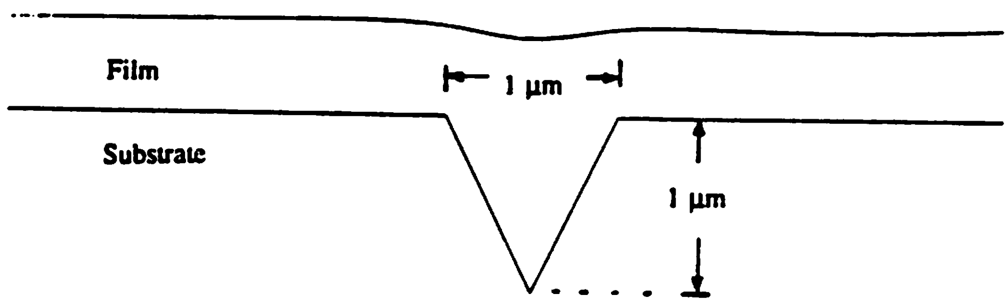


Fig. 4. A triangular groove.

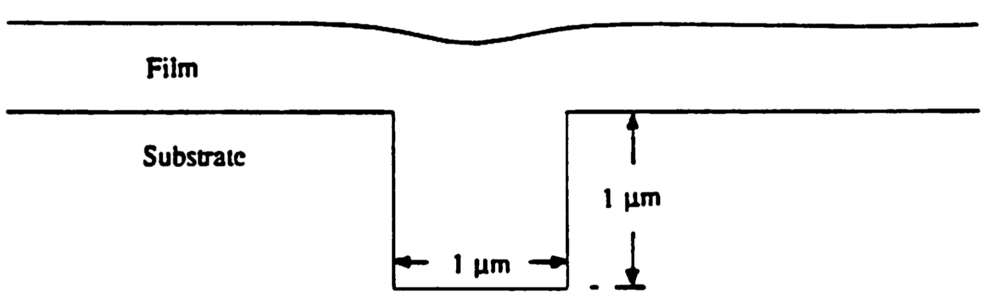


Fig. 5. A rectangular groove.

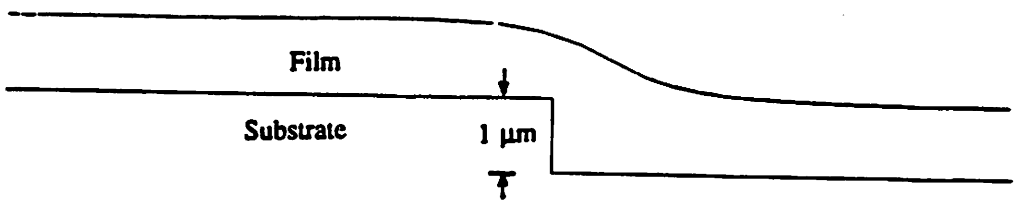


Fig. 6. A step.

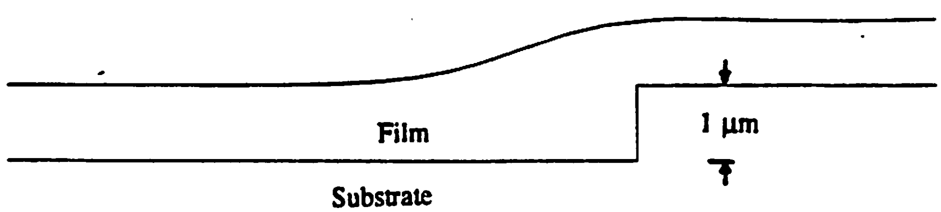


Fig. 7. A step (different orientation).

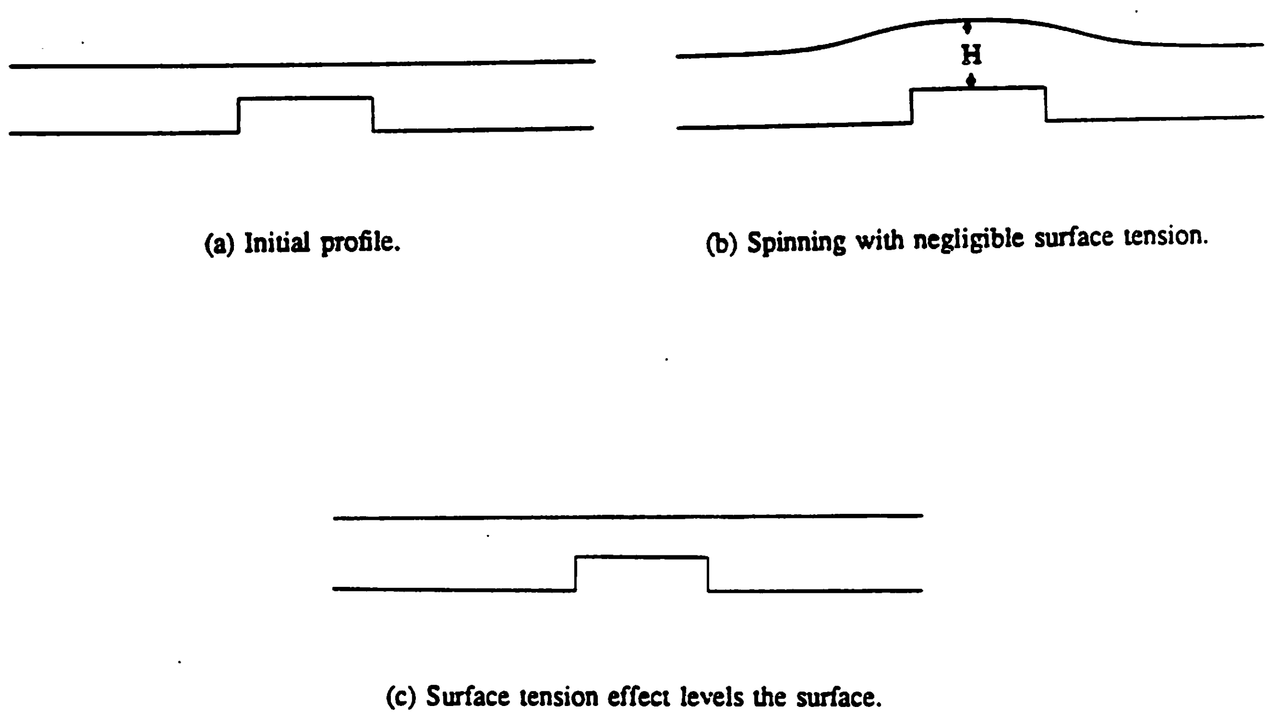


Fig. 8. To show the significance of surface tension in spin coating processes.

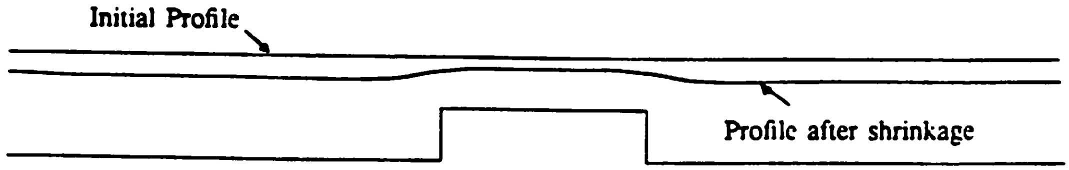


Fig. 9. Shrinkage with no spinning.

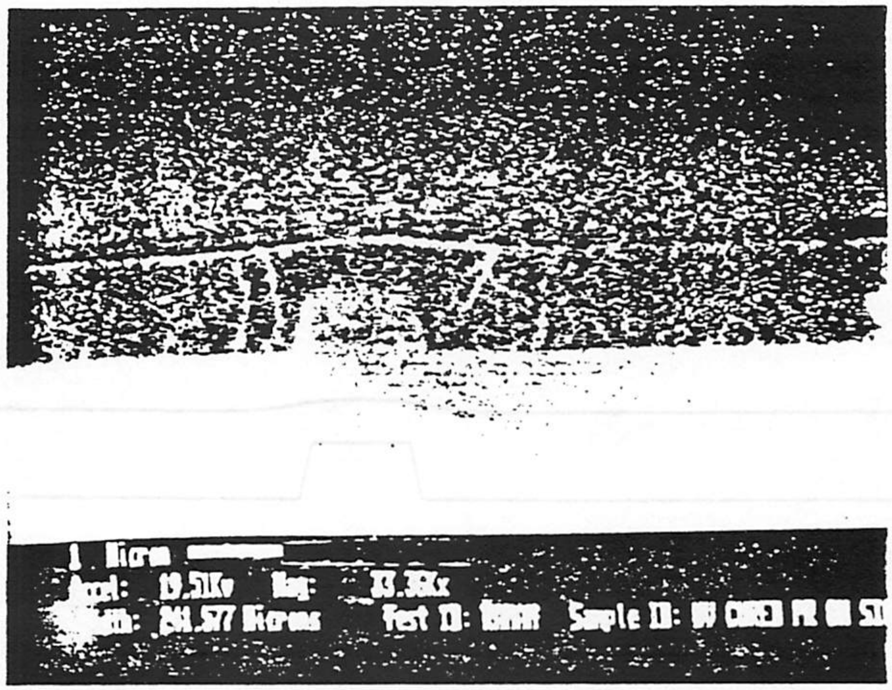


Fig. 10. SEM photograph of a bump.

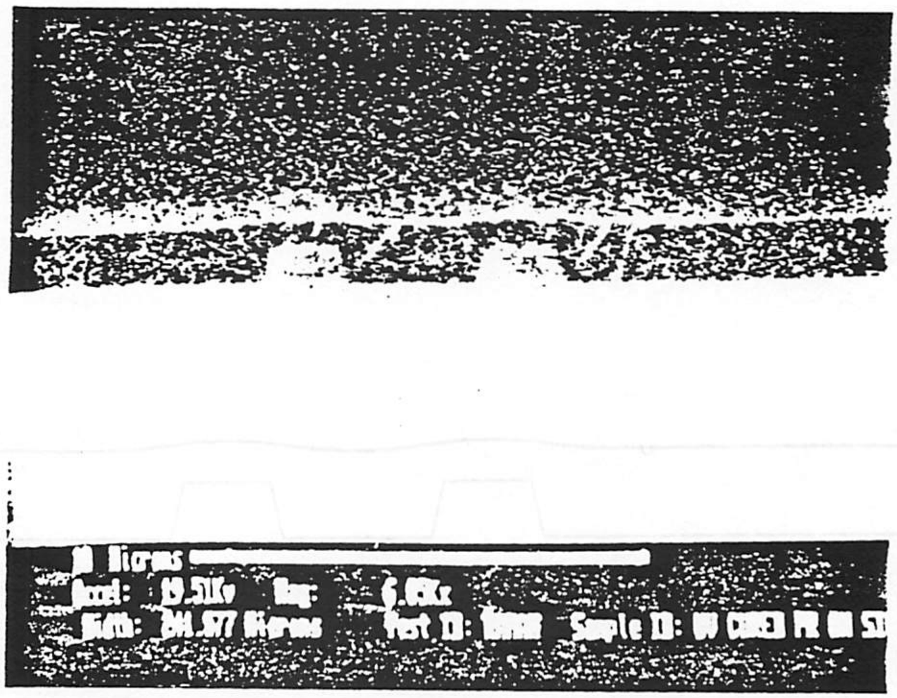


Fig. 11. SEM photograph of two bumps.

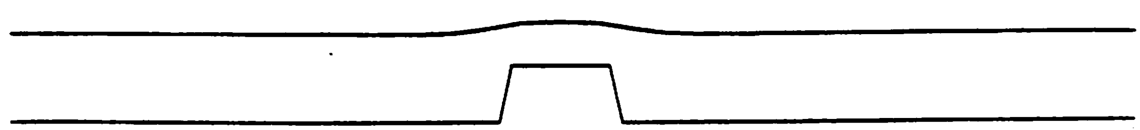


Fig. 12. Simulation result corresponding to Fig. 10.

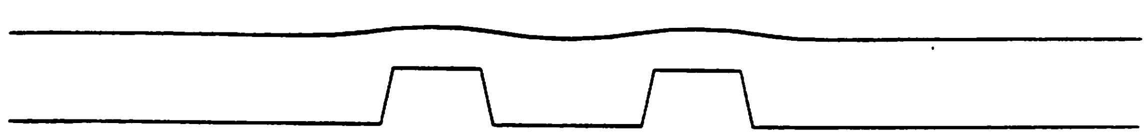


Fig. 13. Simulation result corresponding to Fig. 11.

Chapter 6

Future Work

There are a number of improvements that can be made to the model:

1) Evaporation

2) Non-Newtonian fluid behavior

3) Viscoelastic behavior

4) Acceleration

5) Variable concentration (inhomogeneous material properties)

6) Instability during spinning

The first improvement is to include solvent evaporation during spinning. Flack *et al.* [8] has shown that solvent evaporation controls the film thickness. Meyerhofer[9] predicted that evaporation rate is proportional to the square root of the spin speed. Flack also suggested that photoresist may not exhibit Newtonian behavior at high spin speed. We need to do more studies to investigate the accuracy of the Newtonian-fluid assumption.

The assumption of viscous flow of fluid may be invalid when most of the solvent has been evaporated. We believe the film will become viscoelastic towards the end of spinning. Hence, a viscoelastic model may be required to account for the change of fluid properties.

Evaporation also results in inhomogeneous material properties, especially in the vertical direction. To account for this phenomenon, a more elaborate physical model is needed. Furthermore, transient analysis of the problem is required.

Acceleration at the start of the spin may affect the final profile. The kinds of instabilities in the flow described by Damon[10] ("waves of liquid moving form the center out") can be observed when the acceleration is not rapid enough (of the order of 0.2s or larger) or when the solution is applied very nonuniformly, and particularly at low spin speeds. This suggests that the term $\frac{\partial P}{\partial t}$ in equations 4.1 and 4.2 has to be included in the simulation program. However, solving this kind of transient problem at this stage is too prohibitive in terms of the CPU time.

Chapter 7

Conclusions

We have introduced a spin model and incorporated it into CREEP, a general process simulator. The simulation results show reasonable film profiles over several different types of topography. Complete simulation starting from a thick initial film is still impractical at this moment. However, it is found that we need not start from a thick film of photoresist material in order to get the profile when the film is spun to a thin layer. The surface-tension effect is significant and requires a very small time step to produce stable simulation results. However, from simulation results, we believe the final film profile is determined by the shrinkage of the film. We suspect a non-constant viscosity also has an important effect on the surface tension of the film. In the future, we hope to include a solvent-evaporation model to account for the variation in film viscosity.

References

1. L. K. White, "Approximating Spun-on, Thin Film Planarization Properties on Complex Topography," *Electrochem. Soc. : Solid-State Science and Technology*, p. 168, RCA Laboratories, David Sarnoff Research Center, Princeton, New Jersey, Jan 1985.

2. A. G. Emslie, F. J. bonner, and L. G. Peck, *Applied Physics*, vol. 29, p. 858, 1958.

3. P. Sutardja, *Finite Element Method in Oxidation Process Simulation*, Master Thesis (UC Berkeley), May 1985.

4. W. J. Daughton and F. L. Givens, "An Investigation of the Thickness Variation of Spun-on Thin Films Commonly Associated with the Semiconductor Industry," *Electrochem. Soc. : Solid-State Science and Technology*, p. 173, Texas Instruments Incorporated, Dallas, Texas, Jan 1982.

5. L. E. Stillwagon, R. G. Larson, and G. N. Taylor, "Planarization of Substrate Topography by Spin Coating," *Electrochem. Soc. : Solid-State Science and Technology*, p. 2030, AT&T Bell Laboratories, August 1987.

6. William F. Hughes, *An Introduction to Viscous Flow*, p. 50, Hemisphere Publishing Corporation, Washington, 1979.

7. Noboru Kikuchi, *Finite Element Methods in Mechanics*, p. 301, Cambridge University, Cambridge, 1986.

8. Warren W. Flack, David S. Soong, Alexis T. Bell, and Dennis W. Hess, "A Mathematical Model for Spin Coating of Polymer Resists," *Applied Physics*, vol. 56(4), p. 1199, University of California, Berkeley, August 1984.

9. Detrich Meyerhofer, "Characteristics of Resist Film Produced by Spinning," *Applied Physics*, vol. 49(7), p. 3993, RCA Laboratories, David Sarnoff Research Center, Princeton, New Jersey, July 1978.

10. G. F. Damon, *Proceedings of the Second Kodak Seminar on Microminiaturization (Eastman Kodak Co., Rochester, New York, 1967)*, p. 36.

Appendix A

Derivation of Stiff Matrix.

The matrix equation for pressure-velocity formulation of fluid flow is repeated below.

$$[\,\mathbf{K}_v + \mathbf{K}_p\,]\mathbf{V} = \mathbf{F}$$

where

$$\mathbf{K}_v = \eta \int_\Omega \mathbf{B}^T \mathbf{D} \mathbf{B}\, d\Omega$$
$$\mathbf{K}_p = \alpha \int_\Omega \mathbf{B}^T \mathbf{m}\mathbf{m}^T \mathbf{B}\, d\Omega$$
$$\mathbf{F} = \int_\Omega \mathbf{N}^T \mathbf{b}\, d\Omega$$

$$\mathbf{B}_i = \mathbf{L}\mathbf{N}_i$$

$$\mathbf{B}_i = \begin{bmatrix} N_{i,r} & 0 \\ \frac{1}{r}N_i & 0 \\ 0 & N_{i,z} \\ N_{i,z} & N_{i,r} \end{bmatrix}$$

then

$$(\mathbf{B}^T\mathbf{D}\mathbf{B})_i = \begin{bmatrix} 2N_{i,r}N_{j,r} + \frac{2}{r^2}N_iN_j + N_{i,z}N_{j,z} & N_{i,z}N_{j,r} \\ N_{i,r}N_{j,z} & 2N_{i,z}N_{j,z} + N_{i,r}N_{j,r} \end{bmatrix}$$

$$(\mathbf{B}^T\mathbf{m}\mathbf{m}^T\mathbf{B})_i = \begin{bmatrix} (N_{i,r} + \frac{1}{r}N_i)(N_{j,r} + \frac{1}{r}N_j) & N_{j,z}(N_{i,r} + \frac{1}{r}N_i) \\ N_{i,z}(N_{j,r} + \frac{1}{r}N_j) & N_{i,z}N_{j,z} \end{bmatrix}$$

Appendix B

Surface Tension Effect

The model for surface tension is given by

$$E_s = \gamma A_s \tag{1}$$

where E_s = surface energy,

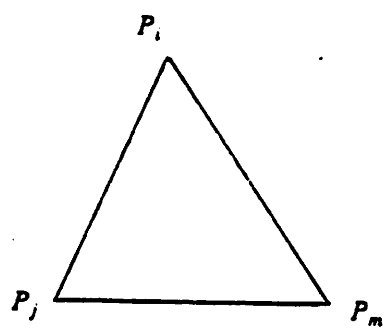
γ = surface tension coefficient,

A_s = surface area.

and the rate of change of surface energy is

$$\dot{E}_s = \gamma \, \delta \dot{A}_s \tag{2}$$

Consider the following triangular element.



For this element, the total suface energy is

$$E_s = E_{s\,ij} + E_{s\,jm} + E_{s\,mi} \tag{3}$$

Now, let us consider the surface energy contribution from each segment. Using segment i-j as an example, we have

$$E_{s\,ij} = \int_{P_i}^{P_j} r \, dl \tag{4.1}$$

$$= \int_{r_i}^{r_j} r\sqrt{1+\alpha^2}\, dr \qquad \textit{where } \alpha = \frac{z_i - z_j}{r_i - r_j} \tag{4.2}$$

$$= \tfrac{1}{2}\sqrt{1+\alpha^2}\,(r_j^2 - r_i^2) \tag{4.3}$$

$$= \frac{1}{2}(r_i+r_j)\sqrt{(r_i-r_j)^2+(z_i-z_j)^2} \tag{4.4}$$

Therefore,

$$E_{s\,ij} = \frac{1}{2}(r_i+r_j)\,l_{ij} \tag{5}$$

where

$$l_{ij} = \sqrt{(r_i-r_j)^2+(z_i-z_j)^2}$$

By partial differentiation of equation 5, we have

with respect to r_i:

$$\frac{1}{2}(l_{ij} + \frac{r_i^2-r_j^2}{l_{ij}}) \tag{6}$$

with respect to z_i:

$$\frac{1}{2}(r_i + r_j)(\frac{z_i-z_j}{l_{ij}}) \tag{7}$$

Appendix C

Spinner User Manual

The spinner module contains the following commands. For other commands of CREEP, please refer to the Ph.D. thesis of P. Sutardja (*"Finite-Element Methods for Process Simulation Application to Silicon Oxidation"*, UCB, May, 1988).

spin — This is the command to perform the spinning process, reflow and shrinkage, depending on input parameters.

time — A **float** (floating point) variable (initially set to 30.0 second). This is the total spin time, which can be subdivided into smaller time intervals by the command variable **t_divide**. If the spin speed is zero, then **time** is by default divided into 30 equal intervals unless the command **t_divide** is specified with a positive value. If **t_divide** is not specified, then the program will automatically evaluate the time steps which will move the points at the vicinity of the topographic feature a distance approximately equal to half of the typical mesh length close to the feature for each time step.

t_divide — A **float** variable (initially set to -1.0) which divides the total spin time into equal interval of **time/t_divide** seconds. This is only effective when the value is positive (Refer to the command **time** above).

del_t — A **float** variable which gives the size of the time step. Its value is readable, but it can only be changed indirectly by specifying the command **t_divide**.

surf_tension — A **float** variable (initially set to 0.0). This is used to set the surface tension at the oxide/ambient interface (in dyne/cm). Negative value means that the surface energy is higher than the bulk energy (ie, contraction force).

div_v	A float variable (initially set to 0.0) used to set the rate of divergence of the velocity for shrinkage simulation (in per second unit). Set a negative number for shrinkage, and positive value for expansion.
mesh_density	A float variable (initially set to 1.0). The automatic mesh generator generates a quasi-uniform mesh. The nominal mesh-length will be approximately 1/mesh_density (in micron). Currently, the simulation domain can be cut into 3 or 5 regions. In the case of three subregions, three mesh densities should be specified, i.e., **mesh1_density** (or **mesh_density**), **mesh2_density** and **mesh3_density**. For five subregions, additional parameters, **mesh4_density** and **mesh5_density**, have to be specified.
mesh1_density	Refer to **mesh_density**.
mesh2_density	Refer to **mesh_density**.
mesh3_density	Refer to **mesh_density**.
mesh4_density	Refer to **mesh_density**.
mesh5_density	Refer to **mesh_density**.
monitor_mesh	An **int** (integer) variable used as a flag to tell the spin command to show the finite-element mesh generated at every time-step of the computation. It is initially set to 1 (true).
newton_method	An **int** variable used as a flag to tell the program to solve the problem using Newton-Ralphson's iterative method if this value is 1 (default value). If other value is used, then the nonlinear terms of the Navier-Stokes equation will be forced to equal to zero (no iteration).

tol A float variable (initially set to 10.0) used as a criterion for the convergence of the Newton-Ralphson's iterative method. If the square of the magnitude of the maximum error force vector is greater than **tol**, the iteration process stops.

iterate_count An **int** variable used to tell the Newton-Ralphson's iteration process to stop after this number of iterations. The default value is 30. If this value is exceeded, the program will terminate, giving an error message.

spin_rpm A float variable (initially set to 0.0) for specifying the spin speed in revolutions per minute.

res_visc A float variable (initially set to 0.0) for specifying the viscosity of the fluid. The unit is dyne second / cm^2.

res_den A float variable (initially set to 0.0) for specifying the density of the fluid. The unit is g / cm^3.

rcut1 A float variable (initially set to 0.0) for specifying the position at which a region is to be split into two parts. If three different mesh densities are to be used, then **rcut1** and **rcut2** have to be specified. If five different mesh densities are to be used, then an additional variables **rcut3** and **rcut4** have to be specified. The latter two corresponds to the cuts in between the former two. i.e., for five regions, the order of cuts from left to right is **rcut1**, **rcut3**, **rcut4**, **rcut2**.

rcut2 Refer to **rcut1**.

rcut3 Refer to **rcut1**.

rcut4 Refer to **rcut1**.

vel_vector An int variable (initially set to 0) used to specify if the velocity vectors at the nodes should be displayed graphically. A value of 0 means no velocity vectors will be displayed. A value of 1 will display the velocity vectors normalized to the maximum velocity in the whole simulation domain. If it is desired to display all the velocity vectors at equal magnitude for the sake of clarity, then a value of 2 should be used. The default magnitude in this case is approximately 1 μm on the screen.

vel_scale A float variable (initially set to 1.0) used to change the magnitude of the velocity vectors by a factor of this value.

Appendix D

Examples of Input Files

The following are some examples of spin coating simulations over aligned marks. The simulation results are shown in chapter 5.

Example 1.

```
# See Fig. 1. of chapter 5.
# Topography: A bump.

sig_intr

path ds_mod plotter
spinner

# Mesh densities for three subregions of polymer film.
mesh_density = 0.0009 ;
mesh2_density = 0.3 ;
mesh3_density = 0.004 ;

monitor_mesh = on ;
plot_range : 9950 0 10050 50 ;
plot_window : 10 10 600 300 ;

rcut1 = 9980
rcut2 = 10040

newton_method = off ;

struct 1a.st ;  # for Fig. 1(a) in chapter 5.
#struct 1b.st ; # for Fig. 1(b) in chapter 5.
#struct 1c.st ; # for Fig. 1(c) in chapter 5.
draw

surf_tension = -0.005 ;  # surface tension in dyne / cm
res_visc = 0.2  ;        # film viscosity in dyne-second / cm^2
res_den = 1.5 ;          # film density in g / cm^3
spin_rpm = 4000 ;        # spin speed in rpm

time = 0.0002 * 15       # total spin time
t_divide = 15 ;          # divide time into 10000 equal intervals

double t ;
int count ;
count = 0 ;

while time > 0.0
```

```
   spin
   t = t + del_t ;
   !echo
   !echo total time elapsed:
   p t
   !echo
   count = count + 1
   !echo Number of steps:
   p count
   !echo
   cl
   draw
end
interactive
```

Structure for 1a.st

```
nodes
     0  0     0
     1  11000 0
     2  11000 15
     3  0     15
     4  0     1
     5  10004 1
     6  10004 4
     7  10006 4
     8  10006 1
     9  11000 1
     10 11000 10
     11 0     10
segments
 0.   0 1 0 2
 1    1 9 0 2
 2    9 10 0 9
 3    10 2 0 1
 4    2 3 0 1
 5    3 11 0 1
 6    11 4 0 9
 7    4 0 0 2
 8    4 5 2 9
 9    5 6 2 9
 10   6 7 2 9
 11   7 8 2 9
 12   8 9 2 9
 13   10 11 1 9
```

Structure for 1b.st

```
nodes
     0  0     0
     1  11000 0
     2  11000 15
```

```
     3  0      15
     4  0      1
     5  10004 1
     6  10004 4
     7  10006 .4
     8  10006 1
     9  11000 1
     10 0      10
     11 9985  10
     12 9993  11.5
     13 9996  11.5
     14 9998  11
     15 10004 10
     16 11000 10
segments
 0  0 1 0 2
 1  1 9 0 2
 2  9 16 0 9
 3  16 2 0 1
 4  2 3 0. 1
 5  3 10 0 1
 6  10 4 0 9
 7  4 0 0 2
 8  4 5 2 9
 9  5 6 2 9
 10 6 7 2 9
 11 7 8 2 9
 12 8 9 2 9
 13 10 11 9 1
 14 11 12 9 1
 15 12 13 9 1
 16 13 14 9 1
 17 14 15 9 1
 18 15 16 9 1
```

Structure for 1c.st

```
nodes
     0  0      0
     1  11000 0
     2  11000 15
     3  0      15
     4  0      1
     5  10004 1
     6  10004 4
     7  10006 4
     8  10006 1
     9  11000 1
     10 0      10
     11 10002 10
     12 10005 12
     13 10008 10
     14 11000 10
```

```
segments
 0   0 1 0 2
 1   1 9 0 2
 2   9 14 0 9
 3   14 2 0 1
 4   2 3 0 1
 5   3 10 0 1
 6   10 4 0 9
 7   4 0 0 2
 8   4 5 2 9
 9   5 6 2 9
 10  6 7 2 9
 11  7 8 2 9
 12  8 9 2 9
 13  10 11 9 1
 14  11 12 9 1
 15  12 13 9 1
 16  13 14 9 1
```

Example 2.

```
# See Fig. 2. of chapter 5.
# Topography: A bump.

sig_intr

path ds_mod plotter
spinner

mesh_density = 0.0004 ;
mesh2_density = 6 ;
mesh3_density = 0.007;

monitor_mesh = on ;
plot_range : 9997 0 10003 3 ;
plot_window : 10 10 600 300 ;

rcut1 = 9998 ;
rcut2 = 10003 ;

newton_method = off ;

struct bump1.st ;
draw

surf_tension = -0.003 ;
res_visc = 0.27 ;
res_den = 1.5 ;
spin_rpm = 4700 ;

time = 0.0002 * 15
t_divide = 15 ;

double t ;
int count ;
count = 0 ;

while time > 0.0
   spin
   t = t + del_t ;
   !echo
   !echo total time elapsed:
   p t
   !echo
   count = count + 1
   !echo Number of steps:
   p count
   !echo
   cl
   draw
end
interactive
```

Structure for bump1.st

```
nodes
     0  0 0
     1  10500    0
     2  10500    5
     3  0        5
     4  0        1
     5  10000    1
     6  10000    1.25
     7  10001    1.25
     8  10001    1
     9  10500    1
     10 0        1.26
     11 9999.3   1.26
     12 9999.8   1.4
     13 10001.2  1.4
     14 10001.7  1.26
     15 10500    1.26
segments
 0    0 1 0 2
 1    1 9 0 2
 2    9 15 0 9
 3    15 2 0 1
 4    2 3 0 1
 5    3 10 0 1
 6    10 4 0 9
 7    4 0 0 2
 8    4 5 2 9
 9    5 6 2 9
 10   6 7 2 9
 11   7 8 2 9
 12   8 9 2 9
 13   10 11 9 1
 14   11 12 9 1
 15   12 13 9 1
 16   13 14 9 1
 17   14 15 9 1
```

Example 3.

```
# See Fig. 3. of chapter 5.
# Topography: 2 bumps separated by 2 µm.

sig_intr

path ds_mod plotter
spinner

mesh_density = 0.0004 ;
mesh2_density = 6 ;
mesh3_density = 0.007;

monitor_mesh = on ;
plot_range : 9997 0 10007 5 ;
plot_window : 10 10 600 300 ;

rcut1 = 9998 ;
rcut2 = 10006 ;

newton_method = off ;

struct bump2.st ;
draw

surf_tension = -0.003 ;
res_visc = 0.27 ;
res_den = 1.5 ;
spin_rpm = 4700 ;

time = 0.0002 * 20 ;
t_divide = 20 ;

double t ;
int count ;
count = 0 ;

while time > 0.0
   spin
   t = t + del_t ;
   !echo
   !echo total time elapsed:
   p t
   !echo
   count = count + 1
   !echo Number of steps:
   p count
   !echo
   cl
   draw
end
interactive
```

Structure for bump2.st

```
nodes
     0  0 0
     1  10500    0
     2  10500    5
     3  0        5
     4  0        1
     5  10000    1
     6  10000    1.25
     7  10001    1.25
     8  10001    1
     9  10003    1
     10 10003    1.25
     11 10004    1.25
     12 10004    1
     13 10500    1
     14 0        1.26
     15 9999.3   1.26
     16 9999.8   1.4
     17 10001.3 1.4
     18 10001.8 1.3
     19 10002.2 1.3
     20 10002.7 1.4
     21 10004.2 1.4
     22 10004.7 1.26
     23 10500    1.26
segments
 0   0 1 0 2
 1   1 13 0 2
 2   13 23 0 9
 3   23 2 0 1
 4   2 3 0 1
 5   3 14 0 1
 6   14 4 0 9
 7   4 0 0 2
 8   4 5 2 9
 9   5 6 2 9
 10  6 7 2 9
 11  7 8 2 9
 12  8 9 2 9
 13  9 10 2 9
 14  10 11 2 9
 15  11 12 2 9
 16  12 13 2 9
 17  14 15 9 1
 18  15 16 9 1
 19  16 17 9 1
 20  17 18 9 1
 21  18 19 9 1
 22  19 20 9 1
 23  20 21 9 1
 24  21 22 9 1
 25  22 23 9 1
```

Example 4.

```
# See Fig. 4. of chapter 5.
# Topography: A triangular pit.

sig_intr

path ds_mod plotter
spinner

mesh_density = 0.0004 ;
mesh2_density = 5.5 ;
mesh3_density = 0.007;

monitor_mesh = on ;
plot_range : 9997 0 10003 3 ;
plot_window : 10 10 600 300 ;

rcut1 = 9997 ;
rcut2 = 10003 ;

newton_method = off ;

struct pit1.st ;
draw

surf_tension = -0.0001 ;
res_visc = 0.27 ;
res_den = 1.5 ;
spin_rpm = 4700 ;

time = 2

double t ;
int count ;
count = 0 ;

while time > 0.0
   spin
   t = t + del_t ;
   !echo
   !echo total time elapsed:
   p t
   !echo
   count = count + 1
   !echo Number of steps:
   p count
   !echo
   cl
   draw
end
interactive
```

Structure for pit1.st

```
nodes
     0  0 0
     1  10500    0
     2  10500    5
     3  0        5
     4  0        2
     5  10000    2
     6  10000.5  1
     7  10001    2
     8  10500    2
     9  0        2.5
     10 10500    2.5
segments
 0    0 1 0 2
 1    1 8 0 2
 2    8 10 0 9
 3    10 2 0 1
 4    2 3 0 1
 5    3 9 0 1
 6    9 4 0 9
 7    4 0 0 2
 8    4 5 2 9
 9    5 6 2 9
 10   6 7 2 9
 11   7 8 2 9
 12   9 10 9 1
```

Example 5.

```
# See Fig. 5. of chapter 5.
# Topography: A rectangular pit.

sig_intr

path ds_mod plotter
spinner

mesh_density = 0.0004 ;
mesh2_density = 5 ;
mesh3_density = 0.007;

monitor_mesh = on ;
plot_range : 9997 0 10003 3 ;
plot_window : 10 10 600 300 ;

rcut1 = 9997 ;
rcut2 = 10003.5 ;

newton_method = off ;

struct pit2.st ;
draw

surf_tension = -0.0001 ;
res_visc = 0.27 ;
res_den = 1.5 ;
spin_rpm = 4700 ;

time = 2

double t ;
int count ;
count = 0 ;

while time > 0.0
   spin
   t = t + del_t ;
   !echo
   !echo total time elapsed:
   p t
   !echo
   count = count + 1
   !echo Number of steps:
   p count
   !echo
   cl
   draw
end
interactive
```

Structure for pit2.st

```
nodes
      0   0 0
      1   10500    0
      2   10500    5
      3   0        5
      4   0        2
      5   10000    2
      6   10000    1
      7   10001    1
      8   10001    2
      9   10500    2
      10  0        2.5
      11  10500    2.5
segments
 0    0 1 0 2
 1    1 9 0 2
 2    9 11 0 9
 3    11 2 0 1
 4    2 3 0 1
 5    3 10 0 1
 6    10 4 0 9
 7    4 0 0 2
 8    4 5 2 9
 9    5 6 2 9
 10   6 7 2 9
 11   7 8 2 9
 12   8 9 2 9
 13   10 11 9 1
```

Example 6.

```
# See Fig. 6. of chapter 5.
# Topograghy: A step

sig_intr

path ds_mod plotter
spinner

mesh_density = 0.0004 ;
mesh2_density = 2.6 ;
mesh3_density = 0.007;

monitor_mesh = on ;
plot_range : 9994 0 10004 7 ;
plot_window : 10 10 600 300 ;

rcut1 = 9993 ;
rcut2 = 10007 ;

newton_method = off ;

struct step1.st ;
draw

surf_tension = -0.001 ;
res_visc = 0.27 ;
res_den = 1.5 ;
spin_rpm = 4700 ;

time = 0.0002 * 20 ;
t_divide = 20 ;

double t ;
int count ;
count = 0 ;

while time > 0.0
   spin
   t = t + del_t ;
   !echo
   !echo total time elapsed:
   p t
   !echo
   count = count + 1
   !echo Number of steps:
   p count
   !echo
   cl
   draw
end
interactive
```

Structure for step1.st

```
nodes
     0  0 0
     1  10500   0
     2  10500   5
     3  0       5
     4  0       1
     5  10000   1
     6  10000   2
     7  10500   2
     8  0       2
     9  9998    2
     10 10001   3
     11 10500   3
segments
 0    0 1 0 2
 1    1 7 0 2
 2    7 11 0 9
 3    11 2 0 1
 4    2 3 0 1
 5    3 8 0 1
 6    8 4 0 9
 7    4 0 0 2
 8    4 5 2 9
 9    5 6 2 9
 10   6 7 2 9
 12   8 9 9 1
 13   9 10 9 1
 14   10 11 9 1
```

Example 7.

```
# See Fig. 7. of chapter 5.
# Topograghy: A step

sig_intr

path ds_mod plotter
spinner

mesh_density = 0.0004 ;
mesh2_density = 2.6 ;
mesh3_density = 0.007;

monitor_mesh = on ;
plot_range : 9994 0 10004 7 ;
plot_window : 10 10 600 300 ;

rcut1 = 9993 ;
rcut2 = 10007 ;

newton_method = off ;

struct step2.st ;
draw

surf_tension = -0.001 ;
res_visc = 0.27 ;
res_den = 1.5 ;
spin_rpm = 4700 ;

time = 0.0002 * 20 ;
t_divide = 20 ;

double t ;
int count ;
count = 0 ;

while time > 0.0
   spin
   t = t + del_t ;
   !echo
   !echo total time elapsed:
   p t
   !echo
   count = count + 1
   !echo Number of steps:
   p count
   !echo
   cl
   draw
end
interactive
```

Structure for step1.st

```
nodes
     0  0 0
     1  10500   0
     2  10500   5
     3  0       5
     4  0       2
     5  10000   2
     6  10000   1
     7  10500   1
     8  0       3
     9  9999    3
     10 10002   2
     11 10500   2
segments
 0    0 1 0 2
 1    1 7 0 2
 2    7 11 0 9
 3    11 2 0 1
 4    2 3 0 1
 5    3 8 0 1
 6    8 4 0 9
 7    4 0 0 2
 8    4 5 2 9
 9    5 6 2 9
 10   6 7 2 9
 12   8 9 9 1
 13   9 10 9 1
 14   10 11 9 1
```

Example 8.

```
# See Fig. 8. of chapter 5.
# To study the significance of surface tension.

sig_intr

path ds_mod plotter
spinner

mesh_density = 0.0004 ;
mesh2_density = 6 ;
mesh3_density = 0.007;

monitor_mesh = on ;
plot_range : 9997 0 10003 3 ;
plot_window : 10 10 600 300 ;

rcut1 = 9997 ;
rcut2 = 10004 ;

newton_method = off ;

struct 8.st ;
draw

surf_tension = -0.00001 ;
res_visc = 0.27 ;
res_den = 1.5 ;
spin_rpm = 4700 ;

time = 0.001 * 1000 ;
t_divide = 1000 ;

double t ;
int count ;
count = 0 ;

while time > 0.0
   spin
   t = t + del_t ;
   !echo
   !echo total time elapsed:
   p t
   !echo
   count = count + 1
   !echo Number of steps:
   p count
   !echo
   cl
   draw
end
interactive
```

Structure for 8.st

```
nodes
     0  0 0
     1  10500    0
     2  10500    5
     3  0        5
     4  0        1
     5  10000    1
     6  10000    1.25
     7  10001    1.25
     8  10001    1
     9  10500    1
     10 0        1.50
     11 9999.3   1.50
     12 9999.8   1.50
     13 10001.2  1.50
     14 10001.7  1.50
     15 10500    1.50
segments
 0    0 1 0 2
 1    1 9 0 2
 2    9 15 0 9
 3    15 2 0 1
 4    2 3 0 1
 5    3 10 0 1
 6    10 4 0 9
 7    4 0 0 2
 8    4 5 2 9
 9    5 6 2 9
 10   6 7 2 9
 11   7 8 2 9
 12   8 9 2 9
 13   10 11 9 1
 14   11 12 9 1
 15   12 13 9 1
 16   13 14 9 1
 17   14 15 9 1
```

Example 9.

```
# See Fig. 9. of chapter 5.
# Shrinkage with no spinning.

sig_intr

path ds_mod plotter
spinner

mesh_density = 0.0004 ;
mesh2_density = 6 ;
mesh3_density = 0.007;

monitor_mesh = off ;
#plot_range : 0 0 10200 5 ;
plot_range : 9997 0 10003 3 ;
plot_window : 10 10 600 300 ;

rcut1 = 9998 ;
rcut2 = 10003 ;

newton_method = off ;

struct 9.st ;
draw

# Shrinkage
div_v = -10 ;
res_visc = 0.27 ;
res_den = 1.5 ;
spin_rpm = 0 ;

time = 0.003 * 60 ;
t_divide = 60 ;

double t ;
int count ;

count = 0 ;

while time > 0.0
   spin
   t = t + del_t ;
   !echo
   !echo total time elapsed:
   p t
   !echo
   count = count + 1
   !echo Number of steps:
   p count
   !echo
   draw
   monitor_mesh = 0 ;
```

```
end
interactive
```

Structure for 9.st

```
nodes
     0  0 0
     1  10500    0
     2  10500    5
     3  0        5
     4  0        1
     5  10000    1
     6  10000    1.25
     7  10001    1.25
     8  10001    1
     9  10500    1
     10 0        1.50
     11 9999.3   1.50
     12 9999.8   1.50
     13 10001.2  1.50
     14 10001.7  1.50
     15 10500    1.50
segments
 0    0 1 0 2
 1    1 9 0 2
 2    9 15 0 9
 3    15 2 0 1
 4    2 3 0 1
 5    3 10 0 1
 6    10 4 0 9
 7    4 0 0 2
 8    4 5 2 9
 9    5 6 2 9
 10   6 7 2 9
 11   7 8 2 9
 12   8 9 2 9
 13   10 11 9 1
 14   11 12 9 1
 15   12 13 9 1
 16   13 14 9 1
 17   14 15 9 1
```

Example 10.

```
# See Fig. 12. of chapter 5.
# Compare with SEM photograph of Fig. 10.
# Topography: 1 bump
# Shrinkage

sig_intr

path ds_mod plotter
spinner

mesh_density = 0.0004 ;
mesh2_density = 1.7 ;
mesh3_density = 0.007;

monitor_mesh = on ;
plot_range : 7490 0 7510 10 ;
plot_window : 10 10 600 300 ;

rcut1 = 7490.5 ;
rcut2 = 7510.5 ;

newton_method = off ;

struct shrink1.st ;
draw

div_v = -10 ;
res_visc = 0.27 ;
res_den = 1.5 ;
spin_rpm = 0 ;

time = 0.003 * 60 ;
t_divide = 60 ;

double t ;
int count ;

count = 0 ;

while time > 0.0
   spin
   t = t + del_t ;
   !echo
   !echo total time elapsed:
   p t
   !echo
   count = count + 1
   !echo Number of steps:
   p count
   !echo
   draw
   monitor_mesh = 0 ;
```

```
end
interactive
```

Structure for shrink1.st

```
nodes
     0  0 0
     1  8000     0
     2  8000     5
     3  0        5
     4  0        1
     5  7500     1
     6  7500.2   1.9
     7  7501.8   1.9
     8  7502     1
     9  8000     1
     10 0        2.8
     11 7499.3   2.8
     12 7499.8   2.8
     13 7502.2   2.8
     14 7502.7   2.8
     15 8000     2.8
segments
 0    0 1 0 2
 1    1 9 0 2
 2    9 15 0 9
 3    15 2 0 1
 4    2 3 0 1
 5    3 10 0 1
 6    10 4 0 9
 7    4 0 0 2
 8    4 5 2 9
 9    5 6 2 9
 10   6 7 2 9
 11   7 8 2 9
 12   8 9 2 9
 13   10 11 9 1
 14   11 12 9 1
 15   12 13 9 1
 16   13 14 9 1
 17   14 15 9 1
```

Example 11.

```
# See Fig. 13. of chapter 5.
# Compare with SEM photograph of Fig. 11.
# Shrinkage
# Topography: 2 bumps

sig_intr

path ds_mod plotter
spinner

mesh_density = 0.0004 ;
mesh2_density = 1.8 ;
mesh3_density = 0.007;

monitor_mesh = on ;
plot_range : 7493 0 7513 10 ;
plot_window : 10 10 600 300 ;

rcut1 = 7493 ;
rcut2 = 7515 ;


newton_method = off ;

struct shrink2.st ;
draw

div_v = -10 ;
surf_tension = 0 ;
res_visc = 0.27 ;
res_den = 1.5 ;
spin_rpm = 0 ;

time = 0.003 * 60 ;
t_divide = 60 ;

double t ;
int count ;

count = 0 ;

while time > 0.0
   spin
   t = t + del_t ;
   !echo
   !echo total time elapsed:
   p t
   !echo
   count = count + 1
   !echo Number of steps:
   p count
   !echo
```

```
   draw
   monitor_mesh = 0 ;
end
interactive
```

Structure for shrink2.st

```
nodes
     0  0 0
     1  8000     0
     2  8000     5
     3  0        5
     4  0        1
     5  7500     1
     6  7500.2   1.9
     7  7501.6   1.9
     8  7501.8     1
     9  7504.2     1
     10 7504.4   1.9
     11 7505.8   1.9
     12 7506     1
     13 8000     1
     14 0        2.7
     15 7499.3   2.7
     16 7499.8   2.7
     17 7502.3   2.7
     18 7502.8   2.7
     19 7503.2   2.7
     20 7503.7   2.7
     21 7506.2   2.7
     22 7506.7   2.7
     23 8000     2.7
segments
 0   0 1 0 2
 1   1 13 0 2
 2   13 23 0 9
 3   23 2 0 1
 4   2 3 0 1
 5   3 14 0 1
 6   14 4 0 9
 7   4 0 0 2
 8   4 5 2 9
 9   5 6 2 9
 10  6 7 2 9
 11  7 8 2 9
 12  8 9 2 9
 13  9 10 2 9
 14  10 11 2 9
 15  11 12 2 9
 16  12 13 2 9
 17  14 15 9 1
 18  15 16 9 1
 19  16 17 9 1
```

```
20 17 18 9 1
21 18 19 9 1
22 19 20 9 1
23 20 21 9 1
24 21 22 9 1
25 22 23 9 1
```