Meccanica (2014) 49:239–252
DOI 10.1007/s11012-013-9788-6

Adaptive mesh refinement for simulation of thin film flows

Yibao Li · Darae Jeong · Junseok Kim

Received: 9 April 2013 / Accepted: 11 July 2013 / Published online: 3 August 2013

Abstract We present a robust and accurate numerical method for simulating gravity-driven, thin-film flow problems. The convection term in the governing equation is treated by a semi-implicit, essentially non-oscillatory scheme. The resulting nonlinear discrete equation is solved using a nonlinear full approximation storage multigrid algorithm with adaptive mesh refinement techniques. A set of representative numerical experiments are presented. We show that the use of adaptive mesh refinement reduces computational time and memory compared to the equivalent uniform mesh results. Our simulation results are consistent with previous experimental observations.

Keywords Adaptive mesh refinement · Nonlinear diffusion equations · Thin film · Nonlinear multigrid method

1 Introduction

Coating flows are fluid flows that generate thin liquid films on surfaces as a result of external forces such as inertia, viscosity, gravity, and surface tension [1]. These coating flows are useful for covering a large surface area with one or more thin, uniform liquid layers [2].

The behavior of thin film flows has been studied experimentally [3, 4], analytically [5–18], and numerically [19–43]. Sun et al. [34] presented a detailed implementation of an adaptive finite element method, and demonstrated its performance on the thin-film problem with a moving contact line. Lee et al. [32, 36] solved a thin film flow over a plane containing well-defined single and grouped topographic features using a full approximation storage multigrid algorithm and employing automatic mesh adaptivity.

In this paper, we develop an adaptive finite difference method for a nonlinear time-dependent gravity-driven thin liquid film equation.

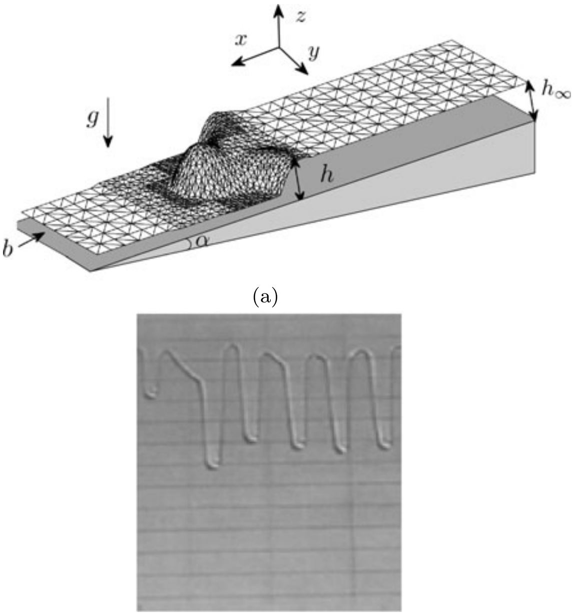
The rest of the paper is organized as follows. In Sect. 2, we briefly describe the governing equation for gravity-driven thin liquid film flow with a lubrication model. In Sect. 3, the fully discrete and nonlinear multigrid scheme and adaptive mesh refinement (AMR) method for the governing equation are given. In Sect. 4, we present numerical results. Conclusions are given in Sect. 5.

2 Governing equation

We consider the dynamics of a thin liquid layer on an inclined surface driven by gravity [24].

Y. Li
Department of Computational Science and Engineering,
Yonsei University, Seoul 120-749, Republic of Korea
e-mail: yibaoli@korea.ac.kr

D. Jeong · J. Kim (✉)
Department of Mathematics, Korea University,
Seoul 136-713, Republic of Korea
e-mail: cfdkim@korea.ac.kr



(a)

(b)

Fig. 1 (**a**) A schematic diagram of the thin film flow. (**b**) A photo of fluid flowing down an inclined plane: Figure reprinted with permission from L. Kondic, SIAM Review, **45**, 95–115 (2003) [24]. Copyright (2012) SIAM. Note that a similar physical experiment was also performed by H.E. Huppert [6]

Figures 1(a) and 1(b) show a schematic illustration and real experimental result for s gravity-driven thin film [24], respectively. The coordinate x increases in the flow direction, y measures distance in the transverse direction, and z is the elevation perpendicular to the substrate. Let $z = h(x, y, t)$ be the thickness of the thin liquid layer. The motion of the fluid is governed by the time-dependent lubrication equation [24]:

$$\frac{\partial h}{\partial t} = \frac{-1}{3\eta}\nabla \cdot \left[\tau h^3 \nabla \Delta h + \rho g h^3 (\sin\alpha \mathbf{e}_x - \cos\alpha \nabla h)\right], \tag{1}$$

where η is viscosity, τ is surface tension, ρ is density, g is the gravitational constant, α is the angle of inclination of the substrate from the horizontal, and $\mathbf{e}_x = (1, 0)$.

To non-dimensionalize Eq. (1), we employ nondimensional variables (denoted by hats) by defining the characteristic variables in terms of balancing terms.

$$h = H\hat{h}, \qquad x = L\hat{x}, \qquad y = L\hat{y}, \qquad \text{and} \qquad t = T\hat{t}. \tag{2}$$

Here H is the thickness of the flat upstream part of the film. L and T are then given by

$$L = \left(\frac{\tau H}{\rho g \sin\alpha}\right)^{1/3} \quad \text{and} \quad T = \frac{3\eta L}{\rho g H^2 \sin\alpha}. \tag{3}$$

After substituting Eqs. (2) and (3) into Eq. (1) and dropping the ‘ˆ’, we obtain the dimensionless thin film equation as

$$h_t + \left(h^3\right)_x = \nabla \cdot \left(h^3 \nabla (Dh - \Delta h)\right), \tag{4}$$

where $D = (\frac{H^2 \rho g \sin\alpha}{\tau})^{1/3} \cot\alpha$ measures the size of the normal component of gravity.

3 Numerical method

First, we split the fourth order Eq. (4) into a system of second order equations as follows.

$$h_t + f_x(h) = \nabla \cdot \left(f(h)\nabla \mu\right), \qquad h = h(x, y, t), \tag{5}$$

$$\mu = Dh - \Delta h, \quad \text{for } (x, y) \in \Omega = (0, L_x) \times (0, L_y), \tag{6}$$

where $f(h) = h^3$. Boundary conditions are given by

$$h(0, y, t) = h_\infty, \qquad h(L_x, y, t) = b,$$
$$h(x, 0, t) = h(x, L_y, t),$$
$$\mu_x(0, y, t) = \mu_x(L_x, y, t) = 0,$$
$$\mu(x, 0, t) = \mu(x, L_y, t),$$

where h_∞ is a constant upstream height and b is a precursor film thickness.

3.1 Discretization of the proposed scheme

We now present fully discrete schemes for Eqs. (5) and (6) in the two-dimensional domain $\Omega = (0, L_x) \times (0, L_y)$. Let N_x and N_y be positive even integers, $\Delta x = L_x/N_x = L_y/N_y$ be the uniform mesh size, and $\Omega_{\Delta x} = \{(x_i, y_j) : x_i = (i - 0.5)\Delta x, y_j = (j - 0.5)\Delta x, 1 \le i \le N_x, 1 \le j \le N_y\}$ be the set of cell-centers. Let h_{ij}^n and μ_{ij}^n be approximations of $h(x_i, y_j, n\Delta t)$ and $\mu(x_i, y_j, n\Delta t)$, respectively. Here, $\Delta t = T/N_t$ is the time step, T is the final time, and N_t is the total number of time steps. Then, a semi-implicit time and centered difference space discretization of Eqs. (5) and (6) is

$$\frac{h_{ij}^{n+1}-h_{ij}^{n}}{\Delta t} = \nabla_d\cdot\big(f(h)_{ij}^{n+1}\nabla_d\mu_{ij}^{n+1}\big) - \frac{f_x(h_{ij}^{n+1})}{2} - \frac{f_x(h_{ij}^{n})}{2}, \tag{7}$$

$$\mu_{ij}^{n+1} = Dh_{ij}^{n+1} - \Delta_d h_{ij}^{n+1}, \tag{8}$$

where the superscripts denote discrete time steps. Here, we use five points to discretize the terms $\Delta_d h_{ij}^{n+1}$ and $\nabla_d\cdot(f(h)_{ij}^{n+1}\nabla_d\mu_{ij}^{n+1})$ as

$$\Delta_d h_{ij}^{n+1} = \frac{h_{i,j+1}^{n+1}+h_{i,j-1}^{n+1}+h_{i+1,j}^{n+1}+h_{i-1,j}^{n+1}-4h_{ij}^{n+1}}{h^2},$$

$$\nabla_d\cdot\big(f(h)_{ij}^{n+1}\nabla_d\mu_{ij}^{n+1}\big) = \frac{f_{i+\frac{1}{2},j}^{n+1}(\mu_{i+1,j}^{n+1}-\mu_{ij}^{n+1}) - f_{i-\frac{1}{2},j}^{n+1}(\mu_{ij}^{n+1}-\mu_{i-1,j}^{n+1})}{h^2} + \frac{f_{i,j+\frac{1}{2}}^{n+1}(\mu_{i,j+1}^{n+1}-\mu_{ij}^{n+1}) - f_{i,j-\frac{1}{2}}^{n+1}(\mu_{ij}^{n+1}-\mu_{i,j-1}^{n+1})}{h^2},$$

where $f_{i+\frac{1}{2},j}^{n+1} = f((h_{ij}^{n+1}+h_{i+1,j}^{n+1})/2)$, and the other terms are similarly defined. $f_x(h_{ij}^n)$ and $f_x(h_{ij}^{n+1})$ are treated using an explicit and a semi-implicit essentially non-oscillatory (ENO) type scheme [44], respectively.

$$f_x(h_{ij}^n) = f'(h_{ij}^n)\left(\frac{\bar{h}_{i+\frac{1}{2},j}^n - \bar{h}_{i-\frac{1}{2},j}^n}{\Delta x}\right).$$

The values $\bar{h}_{i+\frac{1}{2},j}^n$ are calculated as follows:

$$k = \begin{cases} i & \text{if } f'(h_{ij}) \geq 0 \\ i+1 & \text{otherwise.} \end{cases}$$

$$a = \frac{h_{kj}^n - h_{k-1,j}^n}{\Delta x}, \qquad c = \frac{h_{k+1,j}^n - h_{kj}^n}{\Delta x},$$

$$d_{kj} = \begin{cases} a & \text{if } |a| \leq |c| \\ c & \text{otherwise.} \end{cases}$$

$$\bar{h}_{i+\frac{1}{2},j}^n = h_{kj}^n + \frac{\Delta x}{2} d_{kj}\big(1-2(k-i)\big).$$

As $f'(h) = 3h^2 > 0$, then with $k = i$ we can define

$$f_x(h_{ij}^n) := f'(h_{ij}^n)\left(\frac{h_{ij}^n - h_{i-1,j}^n}{\Delta x}\right) + \Im(h_{ij}^n),$$

$$f_x(h_{ij}^{n+1}) := f'(h_{ij}^{n+1})\left(\frac{h_{ij}^{n+1} - h_{i-1,j}^{n+1}}{\Delta x}\right) + \Im(h_{ij}^n),$$

where $\Im(h_{ij}^n) = 0.5(d_{ij} - d_{i-1,j})f'(h_{ij}^n)$. We define the boundary condition as

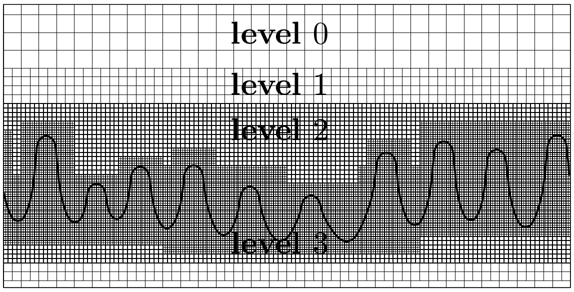


Fig. 2 Block-structured local refinement with four levels

$$h_{0j} = 2h_\infty - h_{1j}, \qquad h_{N_x+1,j} = 2b - h_{N_x j},$$
$$h_{i0} = h_{iN_y},$$
$$h_{i,N_y+1} = h_{i1}, \qquad \mu_{0j} = \mu_{1j}, \qquad \mu_{N_x+1,j} = \mu_{N_x j},$$
$$\mu_{i0} = \mu_{iN_y}, \qquad \mu_{i,N_y+1} = \mu_{i1}.$$

In this paper, we use AMR [45] to solve the nonlinear discrete system (7) and (8) at the implicit time level. Figure 2 shows a schematic diagram of block-structured local refinement with a four level AMR method. A pointwise Gauss–Seidel relaxation scheme is used as the smoother in the multigrid method. The next section gives further details of the AMR algorithm.

3.2 Dynamic adaptive mesh refinement algorithms

In the AMR algorithms, we consider a hierarchy of increasingly finer grids, $\Omega_{l+1},\ldots,\Omega_{l+l^*}$, which are restricted to smaller and smaller subdomains, while the last hierarchy of global grids is $\Omega_0,\Omega_1,\ldots,\Omega_l$. That is, we consider a hierarchy of grids, $\Omega_0,\Omega_1,\ldots,\Omega_{l+0}$, $\Omega_{l+1},\ldots,\Omega_{l+l^*}$. Here, we denote Ω_{l+0} as level zero, Ω_{l+1} as level one, and so on. Construction of the multilevel mesh begins at the zero-level grid. Finer resolution grids are added at level one to cover those grid points on the zero grid where refinement is flagged. This process continues in the same fashion until level l^* is reached. Moreover, the grid spacing Δx_k on level k is related to that of the next level $(k+1)$ as $\Delta x_k = 2\Delta x_{k+1}$. Figure 2 shows a schematic illustration of the set of finer grids with four levels ($l^* = 3$). The AMR is performed by uniformly subdividing each mother cell into four daughter cells.

3.3 Creation of the grid hierarchy

There are many possible criteria for deciding where refinement is necessary. In many physical problems,

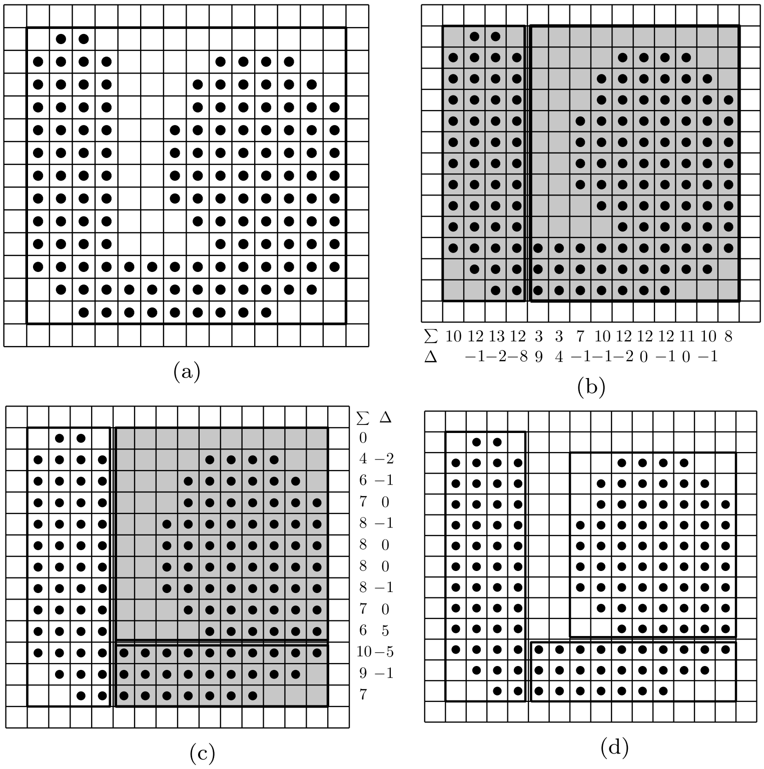

(a)
Σ 10 12 13 12 3 3 7 10 12 12 12 11 10 8
Δ −1 −2 −8 9 4 −1 −1 −2 0 −1 0 −1
(b)
Σ Δ
0
4 −2
6 −1
7 0
8 −1
8 0
8 0
8 −1
7 0
6 5
10 −5
9 −1
7
(c)
(d)

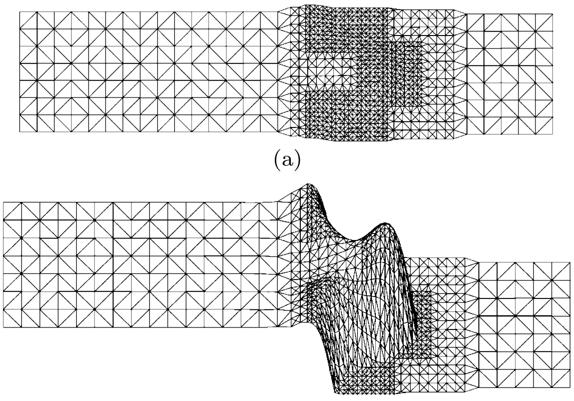
Fig. 3 Three steps in the regridding algorithm: (**a**) tag cells and enclose them in a box, (**b**) split the box into two boxes using a histogram of the column or row sums of tagged cells, (**c**) fit new boxes to each split box and repeat if the ratio of tagged to untagged cells is too small, and (**d**) the most efficient rectangles

physical quantities such as sharp density gradients or large charge distributions may provide indicators for refinement. In the current implementation, the grid is adapted dynamically based on the undivided gradient $|\nabla_u h|^k$, which is defined as

$$|\nabla_u h|^k_{ij} = \sqrt{\left(h^k_{i+1,j} - h^k_{i-1,j}\right)^2 + \left(h^k_{i,j+1} - h^k_{i,j-1}\right)^2},$$

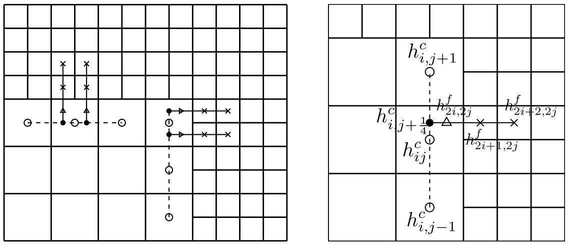
where h^k_{ij} are cell center values defined with respect to the level-k grids on the domain Ω_{l+k}. We then tag cells containing the front, i.e., those in which the undivided gradient of the film height is greater than a critical value *tol*, i.e., $|\nabla_u h|^k > tol$. Throughout this paper, we use $tol = 0.01$. Once we have decided which cells are to be refined, we use this information to create a hierarchy of levels. We use the algorithm of Berger and Rigoutsos [46], in which tagged points are clustered into efficient boxes. The efficiency of a grid is defined as the number of cells in the grid that were tagged for refinement divided by the total number of cells in the grid. The grid generator takes a list of tagged points and draws the smallest box around them. For efficiency, the boxes are not allowed to become too small. This procedure performs the following steps:

1. Fit a box to enclose the tagged cells.
2. Recursively sub-divide the box. Split the box in the longest direction at a position based on the histogram formed from the sum of the number of tagged cells per row or column.
3. After splitting the box, fit new bounding boxes to each half and repeat the process. Continue until the size of every box is not smaller than a given value and every box consists of a minimum number of non-tagged cells.
4. Compute the fill ratio, which is defined as the number of tagged cells divided by the size of the box. For example, the fill ratio in Fig. 3(c) is 135/182. If this grid does not satisfy an efficiency criterion (in the current implementation, fill ratio ≥ 0.75), the grid generator will look for the best way to subdivide the tagged points in order to create more efficient boxes, as shown in Fig. 3(d).


(a)

(b)

Fig. 4 Schematic illustration of our proposed adaptive meshes with different views



(a) (b)

Fig. 5 Interpolation on the coarse-fine boundary: Use quadratic interpolation through coarse cells (*open circles*) to get intermediate values (*solid circles*), then use intermediate value with fine cells (×'s) to get ghost cell values (*open triangles*) for computing coarse-fine fluxes (*arrows*)

A schematic illustration of our proposed adaptive meshes with different views is shown in Fig. 4. Note that our method is not based on triangular adaptive mesh refinement, we simply illustrate these with a triangular mesh plot using MATLAB 2012*a* [47].

3.4 Coarse-fine boundary interpolation

In order to discretize the Laplacian on a generic level grid, we use the standard nearest neighbor stencils in the definitions of the discrete derivatives to fill the ghost-layer values by interpolation.

The grid-to-ghost-layer exchange process is illustrated in Fig. 5(a). The main idea is to use quadratic interpolation through coarse cells (open circles) to get intermediate values (solid circles), then use intermediate values with fine cells (×'s) to get ghost cell values (open triangles) for computing the coarse-fine fluxes (arrows).

To obtain the value $h^c_{i,j+\frac{1}{4}}$ (solid circle), we use three points $h^c_{i,j-1}$, $h^c_{i,j}$, and $h^c_{i,j+1}$ with the quadratic polynomial approximation as shown in Fig. 5(b). Let the quadratic polynomial approximation be $h^c_i(t) = \alpha t^2+\beta t+\gamma$. And assume $h^c_i(-\Delta^c_x) = h^c_{i,j-1}$, $h^c_i(0) = h^c_{ij}$, and $h^c_i(\Delta^c_x) = h^c_{i,j+1}$. The parameters α, β, and γ can be calculated by the following equations.

$$\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} (\Delta^c_x)^2 & -\Delta^c_x & 1 \\ 0 & 0 & 1 \\ (\Delta^c_x)^2 & \Delta^c_x & 1 \end{pmatrix}^{-1} \begin{pmatrix} h^c_{i,j-1} \\ h^c_{ij} \\ h^c_{i,j+1} \end{pmatrix}.$$

Using α, β, and γ, we get $h^c_{i,j+\frac{1}{4}} = h^c_i(\Delta^c_x/4)$. To obtain $h^f_{2i,2j}$, we implement another quadratic interpolation with the values $h^c_{i,j+\frac{1}{4}}$, $h^f_{2i+1,2j}$, and $h^f_{2i+2,2j}$ in a similar fashion.

3.5 Numerical solution—adaptive nonlinear multigrid method

In this subsection, we will only describe the adaptive full approximation storage cycle to solve the nonlinear discrete system of Eqs. (7) and (8). The algorithm of the V-cycle multigrid method for solving the discrete system is as follows. First, let us rewrite Eqs. (7) and (8) as $N(h^{n+1}, \mu^{n+1}) = (\phi^n, \psi^n)$, where $N(h^{n+1}, \mu^{n+1}) = (\frac{h^{n+1}_{ij}}{\Delta t} - \nabla_d \cdot (f(h)^{n+1}_{ij} \nabla_d \mu^{n+1}_{ij}) + \frac{f'(h^{n+1}_{ij})}{2}(\frac{h^{n+1}_{ij}-h^{n+1}_{i-1,j}}{\Delta x}), \mu^{n+1}_{ij} - Dh^{n+1}_{ij} + \Delta_d h^{n+1}_{ij})$ and the source term is $(\phi^n, \psi^n) = (\frac{h^n_{ij}}{\Delta t} - \frac{f'(h^n_{ij})}{2}(\frac{h^n_{ij}-h^n_{i-1,j}}{\Delta x}) - \Im(h^n_{ij}), 0)$.

Using the above notation on all levels $k = 0, 1, \ldots, l, l+1, \ldots, l+l^*$, an adaptive multigrid cycle is formally written as follows [48]:

Adaptive cycle

First we calculate ϕ^n_k, ψ^n_k on all levels and set the previous time solution as the initial guess, i.e., $(h^0_k, \mu^0_k) = (h^n_k, \mu^n_k)$.

$$\bigl(h^{m+1}_k, \mu^{m+1}_k\bigr) = \mathit{ADAPTIVEcycle}\bigl(k, h^m_k, h^m_{k-1}, \mu^m_k, mu^m_{k-1}, N_k, \phi^n_k, \psi^n_k, \nu\bigr).$$

(1) *Presmoothing*

- Compute $(\bar{h}^m_k, \bar{\mu}^m_k)$ by applying ν smoothing steps to (h^m_k, μ^m_k) on Ω_k.

$$\bigl(\bar{h}^m_k, \bar{\mu}^m_k\bigr) = \mathit{SMOOTH}^\nu\bigl(h^m_k, \mu^m_k, N_k, \phi^n_k, \psi^n_k\bigr),$$

where one *SMOOTH* relaxation operator step consists of solving Eqs. (11) and (12) given below by 2 × 2 matrix inversion for each i and j. Note that a pointwise Gauss–Seidel relaxation scheme is used as the smoother in the multigrid method. Rewriting Eqs. (7) and (8), we get

$$\begin{aligned}
&\left(\frac{1}{\Delta t}+\frac{f'(h_{ij}^{n+1})}{2\Delta x}\right)h_{ij}^{n+1}\\
&\quad+\frac{f_{i+\frac{1}{2},j}^{n+1}+f_{i-\frac{1}{2},j}^{n+1}+f_{i,j+\frac{1}{2}}^{n+1}+f_{i,j-\frac{1}{2}}^{n+1}}{\Delta x^2}\mu_{ij}^{n+1}\\
&=\phi_{ij}^{n}+\frac{f_{i+\frac{1}{2},j}^{n+1}\mu_{i+1,j}^{n+1}+f_{i-\frac{1}{2},j}^{n+1}\mu_{i-1,j}^{n+1}}{\Delta x^2}\\
&\quad+\frac{f_{i,j+\frac{1}{2}}^{n+1}\mu_{i,j+1}^{n+1}+f_{i,j-\frac{1}{2}}^{n+1}\mu_{i,j-1}^{n+1}}{\Delta x^2}\\
&\quad+\frac{f'(h_{ij}^{n+1})h_{i-1,j}^{n+1}}{2\Delta x},
\end{aligned}\tag{9}$$

$$\begin{aligned}
&-\left(D+\frac{4}{\Delta x^2}\right)h_{ij}^{n+1}+\mu_{ij}^{n+1}\\
&=\psi_{ij}^{n}-\frac{h_{i+1,j}^{n+1}+h_{i-1,j}^{n+1}+h_{i,j+1}^{n+1}+h_{i,j-1}^{n+1}}{\Delta x^2}.
\end{aligned}\tag{10}$$

Next, we replace h_{kl}^{n+1} and μ_{kl}^{n+1} in Eqs. (9) and (10) with $\bar{h}_{kl}^{m}$ and $\bar{\mu}_{kl}^{m}$ if $k\le i$ and $l\le j$; otherwise we replace them with h_{kl}^{m} and μ_{kl}^{m}, i.e.,

$$\begin{aligned}
&\left(\frac{1}{\Delta t}+\frac{f'(h_{ij}^{m})}{2\Delta x}\right)\bar{h}_{ij}^{m}\\
&\quad+\frac{f_{i+\frac{1}{2},j}^{m}+f_{i-\frac{1}{2},j}^{m}+f_{i,j+\frac{1}{2}}^{m}+f_{i,j-\frac{1}{2}}^{m}}{\Delta x^2}\bar{\mu}_{ij}^{m}\\
&=\phi_{ij}^{n}+\frac{f_{i+\frac{1}{2},j}^{m}\mu_{i+1,j}^{m}+f_{i-\frac{1}{2},j}^{m}\bar{\mu}_{i-1,j}^{m}}{\Delta x^2}\\
&\quad+\frac{f_{i,j+\frac{1}{2}}^{m}\mu_{i,j+1}^{m}+f_{i,j-\frac{1}{2}}^{m}\bar{\mu}_{i,j-1}^{m}}{\Delta x^2}\\
&\quad+\frac{f'(h_{ij}^{m})\bar{h}_{i-1,j}^{m}}{2\Delta x},
\end{aligned}\tag{11}$$

$$\begin{aligned}
&-\left(D+\frac{4}{\Delta x^2}\right)\bar{h}_{ij}^{m}+\bar{\mu}_{ij}^{m}\\
&=\psi_{ij}^{n}-\frac{h_{i+1,j}^{m}+\bar{h}_{i-1,j}^{m}+h_{i,j+1}^{m}+\bar{h}_{i,j-1}^{m}}{\Delta x^2},
\end{aligned}\tag{12}$$

(2) *Coarse-grid correction*

$$(\bar{h}_{k-1}^{m},\bar{\mu}_{k-1}^{m})=\begin{cases}I_k^{k-1}(\bar{h}_k^{m},\bar{\mu}_k^{m}) & \text{on } \Omega_{k-1}\cap\Omega_k\\ (h_{k-1}^{m},\mu_{k-1}^{m}) & \text{on } \Omega_{k-1}-\Omega_k.\end{cases}$$

$$(\phi_{k-1}^{n},\psi_{k-1}^{n})=\begin{cases}I_k^{k-1}\{(\phi_k^{n},\psi_k^{n})-N_k(\bar{h}_k^{m},\bar{\mu}_k^{m})\}\\ \quad+N_{k-1}I_k^{k-1}(\bar{h}_k^{m},\bar{\mu}_k^{m})\\ \quad\text{on } \Omega_{k-1}\cap\Omega_k\\ (\phi_{k-1}^{n},\psi_{k-1}^{n}) \quad \text{on } \Omega_{k-1}-\Omega_k.\end{cases}$$

- Compute an approximate solution $(\hat{h}_{k-1}^{m},\hat{\mu}_{k-1}^{m})$ of the coarse grid equation on Ω_{k-1}, i.e.,

$$N_{k-1}\big(h_{k-1}^{m},\mu_{k-1}^{m}\big)=\big(\phi_{k-1}^{n},\psi_{k-1}^{n}\big).\tag{13}$$

If $k=1$, we explicitly invert a 2 × 2 matrix to obtain the solution. If $k>1$, we solve Eq. (13) using $(\bar{h}_{k-1}^{m},\bar{\mu}_{k-1}^{m})$ as an initial approximation to perform an adaptive multigrid k-grid cycle:

$$\begin{aligned}
&\big(\hat{h}_{k-1}^{m},\hat{\mu}_{k-1}^{m}\big)\\
&\quad=\mathit{ADAPTIVEcycle}\big(k-1,\bar{h}_{k-1}^{m},\bar{\mu}_{k-1}^{m},h_{k-2}^{m},\\
&\qquad\qquad\mu_{k-2}^{m},N_{k-1},\phi_{k-1}^{n},\psi_{k-1}^{n},\nu\big).
\end{aligned}$$

- Compute the correction at $\Omega_{k-1}\cap\Omega_k$.

$$\big(\hat{u}_{k-1}^{m},\hat{v}_{k-1}^{m}\big)=\big(\hat{h}_{k-1}^{m},\hat{\mu}_{k-1}^{m}\big)-\big(\bar{h}_{k-1}^{m},\bar{\mu}_{k-1}^{m}\big).$$

- Set the solution at the other points of $\Omega_{k-1}-\Omega_k$.

$$\big(h_{k-1}^{m+1},\mu_{k-1}^{m+1}\big)=\big(\hat{h}_{k-1}^{m},\hat{\mu}_{k-1}^{m}\big).$$

- Interpolate the correction to Ω_k,

$$\big(\hat{u}_{k}^{m},\hat{v}_{k}^{m}\big)=I_{k-1}^{k}\big(\hat{u}_{k-1}^{m},\hat{v}_{k-1}^{m}\big).$$

- Compute the corrected approximation on Ω_k.

$$\big(h_k^{m,\text{after}\,CGC},\mu_k^{m,\text{after}\,CGC}\big)=\big(\bar{h}_k^{m}+\hat{u}_k^{m},\bar{\mu}_k^{m}+\hat{v}_k^{m}\big).$$

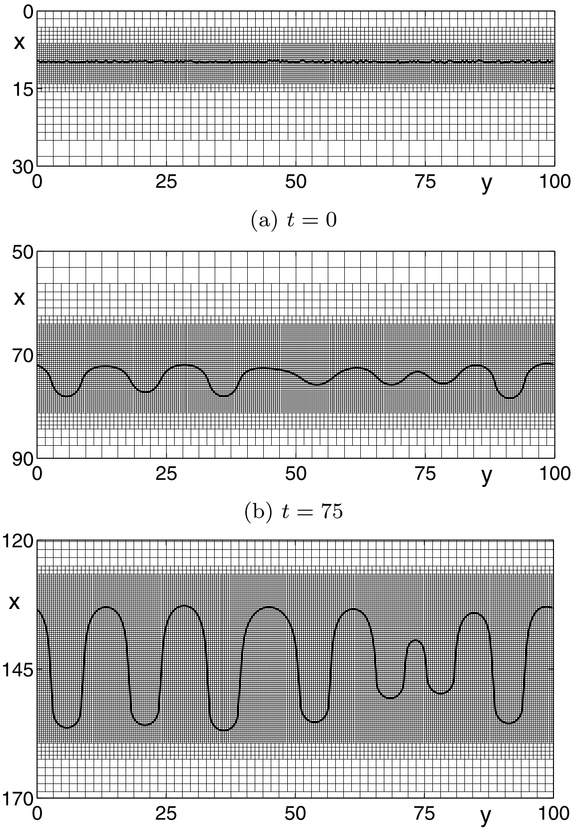
(3) *Postsmoothing*

$$\begin{aligned}
&\big(h_k^{m+1},\mu_k^{m+1}\big)\\
&\quad=\mathit{SMOOTH}^{\nu}\big(h_k^{m,\text{after}\,CGC},\mu_k^{m,\text{after}\,CGC},\\
&\qquad\qquad N_k,\phi_k^{n},\psi_k^{n}\big).
\end{aligned}$$

This completes the nonlinear ADAPTIVE cycle.

4 Numerical experiments

In this section, we perform numerical experiments to determine the evolution of a thin film with AMR, compare the uniform and adaptive mesh methods, study the effect of parameter values, compare the experimental and numerical data, and numerically simulate multiple thin films. Unless otherwise specified, we use $h_\infty=1$ and $b=0.01$ in this paper.



(a) $t = 0$

(b) $t = 75$

(c) $t = 150$

Fig. 6 Time evolution of the fluid front with the adaptive mesh method. The dimensionless times are shown below each figure. $l^* = 3$ levels and level zero domain, $\Omega_{l+0} = 64 \times 32$ are used on the computational domain $\Omega = (0, 200) \times (0, 100)$

4.1 Evolution of thin film with adaptive mesh refinement

We start with a numerical simulation for the temporal evolution of a thin film using AMR with the initial condition $h(x, y, 0) = 0.5[h_\infty + b - (h_\infty - b)\tanh(3(x - 10) + \mathrm{rand}(x, y))]$, where $\mathrm{rand}(x, y)$ is a random value between -1 and 1. In this numerical simulation, $D = 0.1$, $l^* = 3$ levels, and the level zero domain $\Omega_{l+0} = 64 \times 32$ are used on the computational domain $\Omega = (0, 200) \times (0, 100)$. The maximum mesh size is $\Delta x_{\max} = 3.13$ and the finer mesh sizes are 64×32, 128×64, 256×128, and 512×256. The calculation is run until time $T = 150$ with a time step $\Delta t = 0.2$. Figure 6 shows the evolution of the contour ($h = 0.5$) of the thin film at times $t = 0, 75$, and 150. It can be observed from the figures that gravity-driven fingers develop and become longer as time evolves. At the same time, the grid hierarchy structure dynamically adjusts itself to capture the developing finger feature.

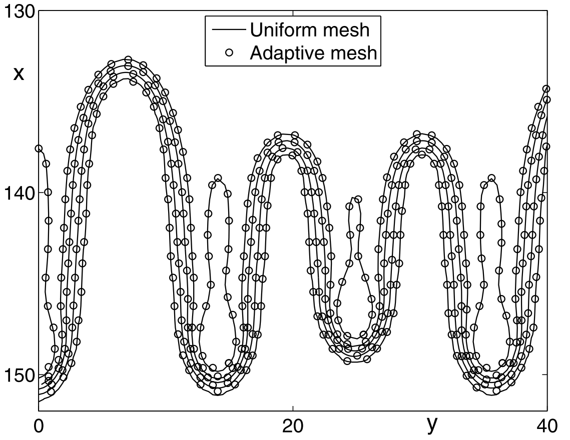


Fig. 7 Comparison between uniform mesh (*solid line*) and adaptive mesh method (*circle*) at $t = 150$. Here, contours of fluid are at $h = 0.1, 0.4, 0.7, 0.9$, and 1.2

4.2 Comparison between uniform and adaptive mesh

In this numerical simulation, we compare a sequence of results with different mesh sizes for a thin film simulation. We compare the numerical solutions obtained on uniform and adaptively refined meshes. The initial condition is $h(x, y, 0) = 0.5[h_\infty + b - (h_\infty - b)\tanh(3(x - 10) + \cos(\pi y/25))]$ on the computational domain $\Omega = (0, 200) \times (0, 100)$. We use a set of increasingly finer meshes $2^{6+n} \times 2^{5+n}$ in the uniform test and $l^* = n$ levels in the adaptive mesh method for $n = 3, 4$, and 5. The maximum grid spacing is $\Delta x_{\max} = 3.13$ and $D = 0.1$. With time step $\Delta t = 0.2$, the calculations are run until $T = 150$. Figure 7 compares the simulation results obtained on uniform and adaptively refined meshes. This indicates that they are in good agreement.

To show the efficiency of our proposed AMR method, the CPU time required for the two methods is listed in Table 1. On a 2048×1024 grid, the uniform mesh method requires more than 37.4 hours. However, the adaptive mesh needs only 1056, which is about 128 times faster than the uniform mesh. The adaptive method uses 10240 nodes at $t = 0$ and 40960 nodes at $t = 150$, which is small compared to the 2048×1024 nodes in the uniform mesh.

Table 1 Comparison of CPU time for uniform and adaptively refined meshes

Grids	512 × 256	1024 × 512	2048 × 1024
Uniform mesh	5542 s	29559 s	134569 s
Adaptive mesh	79 s	261 s	1056 s
$\frac{\text{Uniform}}{\text{Adaptive}}$	70	113	128

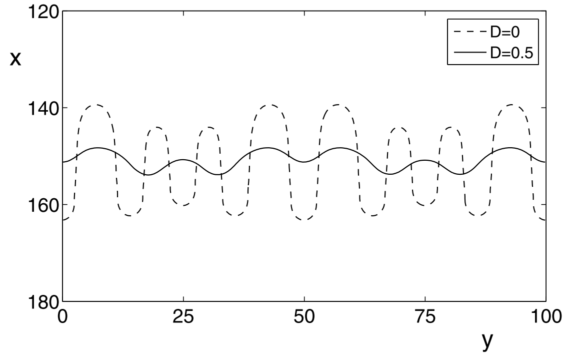


Fig. 8 Snapshots of the contact line profiles for the flow down a vertical plane $D = 0$ and an inclined plane $D = 0.5$

4.3 Effect of parameter D

The non-dimensional parameter D plays an important role in the thin film equation. By changing the film thickness (H) and inclination angle (α), we can adjust $D = (\frac{H^2 \rho g \sin\alpha}{\tau})^{1/3} \cot\alpha$ in the experiment. In general, we have $\frac{H^2 \rho g \sin\alpha}{\tau} \ll 1$ (see Refs. [49, 50]), therefore $|D| < |\cot\alpha|$.

To see the effect of D on the evolution of the thin liquid film, we take two different values, $D = 0$ ($\alpha = \pi/2$) and $D = 0.5$ ($0 < \alpha < \pi/2$). The initial condition is $h(x, y, 0) = 0.5[h_\infty + b - (h_\infty - b)\tanh(3(x - 10) + \cos(\pi y/25))]$. $l^* = 3$ and $\Omega_{l+0} = 64 \times 32$ are used on the computational domain $\Omega = (0, 200) \times (0, 100)$ with a time step $\Delta t = 0.2$. Figure 8 shows snapshots of the contact line profiles for the flow down a vertical plane $D = 0$ and an inclined plane $D = 0.5$ at $t = 160$. The results suggest that, as the value of the parameter D becomes smaller, the growth of the fingering patterns becomes faster.

4.4 Dynamic contact angle

When a liquid is brought into contact with a solid surface, the adhesion of the solid with the liquid and the cohesion of the liquid become interacting forces. The contact angle represents the balance between the cohesion forces of the liquid and the adhesion forces between the liquid and the solid [59]. The contact angle of a thin film can be defined as the angle between the tangent vector of the interface and the solid (see Fig. 9).

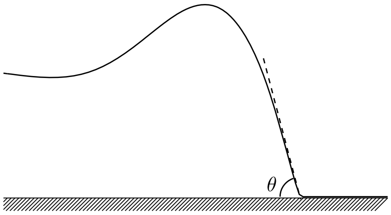


Fig. 9 Contact angle, defined as the angle between the *solid* and a *tangent line* to the interface of the thin film

The dynamic contact angle of thin-film flows has been extensively studied [18, 33, 56, 60, 61], where the authors considered the surface forces to study the wetting phenomena of a spreading liquid drop in contact with a solid surface. We will investigate the effect of b on the dynamic contact angle with two different models. One is a precursor film model, and the other is a slipping model. The difference between these two approaches is that the velocity at the bottom boundary is zero in the precursor model and nonzero in the slipping model. A schematic illustration of the two models is shown in Fig. 10. A detailed comparison of the two models is given in [18, 56, 61]. The different boundary condition leads to a modification to $f(h) = h^3 + Ch$ in Eq. (5). Here, C is a small positive parameter. Note that in the slipping model, b has no physical meaning, but is chosen to numerically preserve non-negativity of solutions. We use $b = 10^{-6}$ in the slipping model.

To calculate the dynamic contact angle, we use a method based on the quadratic polynomial approximation. Let $X_m = (x_m, y_m)$ for $m = 1, \ldots, M$ be uniformly distributed Lagrangian points such that $h(X_m) = 0.5$ (see '+' symbol in Fig. 11(a)). For each X_m, we consider the normal coordinate s with origin at X_m. Along the normal direction (solid line in Fig. 11(a)), we define a function $\bar{h}_m(s)$ by *interp*2, which is a $2D$ interpolation program in MATLAB 2012a [47]. Let k be an integer that satisfies $\bar{h}_m(k\Delta x) \geq 2b$ and $\bar{h}_m((k + 1)\Delta x) \leq 2b$. We define the quadratic polynomial, $\phi_m(s)$, passing three points $((k - 2)\Delta x, \bar{h}_m((k - 2)\Delta x))$, $((k - 1)\Delta x, \bar{h}_m((k - 1)\Delta x))$, and $(k\Delta x, \bar{h}_m(k\Delta x))$. Let s_* be the point that satisfies $\phi_m(s_*) = b$ and $s_* \geq$

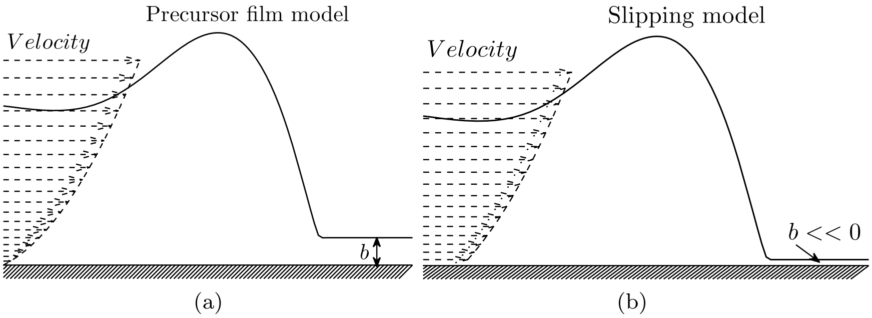


Fig. 10 Schematic illustration of (**a**) precursor film model and (**b**) slipping model

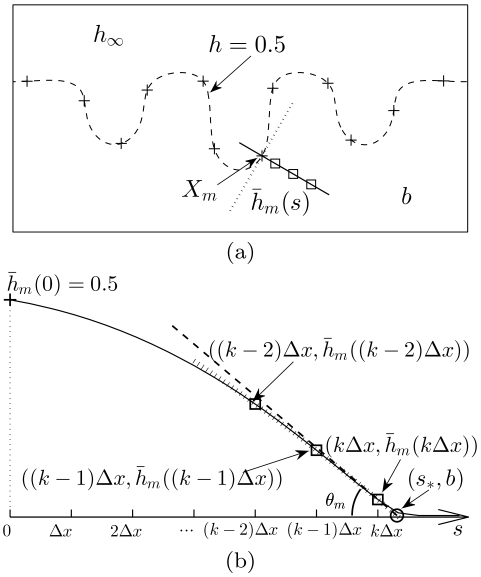


Fig. 11 Schematic illustration of dynamic contact angle computation. (**a**) The *solid line* is the normal direction at the front which is the contour at $h = 0.5$. (**b**) Computation of dynamic contact angle

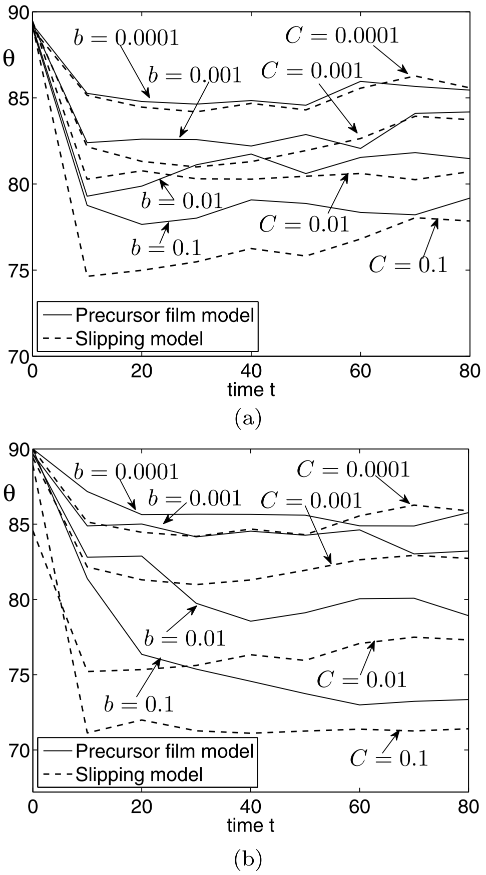
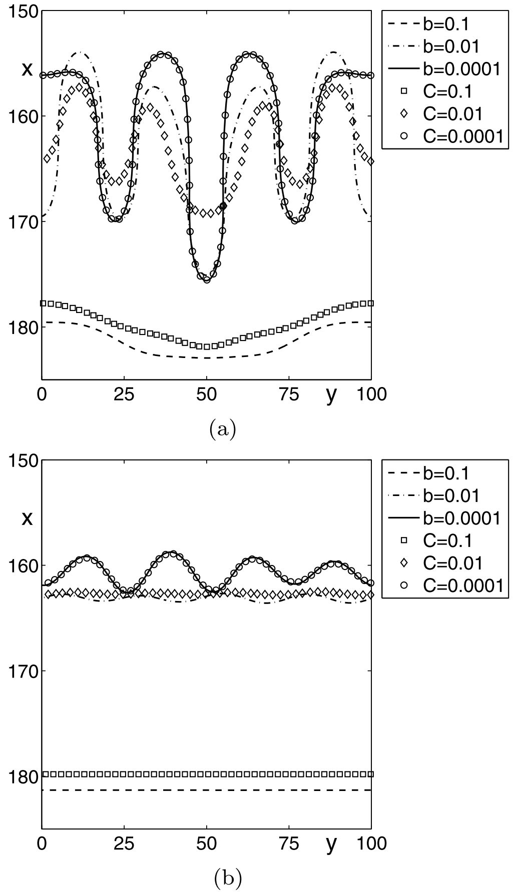


Fig. 12 Evolution of average dynamic contact angle with precursor film model and slipping model. (**a**) The initial profile is $h(x, y, 0) = 0.5[h_\infty + b - (h_\infty - b)\tanh(3(x-5) + \cos(\pi y/25))]$. (**b**) The initial profile is $h(x, y, 0) = 0.5[h_\infty + b - (h_\infty - b)\tanh(3(x-5))]$

$k\Delta x$ (see Fig. 11(b)). The dynamic contact angle at X_m is then defined as $\theta_m = -\tan^{-1}\frac{d\phi_m(s_*)}{ds}$. The average dynamics contact angle is given as $\theta = \sum_{m=1}^{M}\theta_m/M$. We perform a numerical experiment with two initial conditions, $h(x, y, 0) = 0.5[h_\infty + b - (h_\infty - b)\tanh(3(x-5) + \cos(\pi y/25))]$ and $h(x, y, 0) = 0.5[h_\infty + b - (h_\infty - b)\tanh(3(x-5))]$. $\Omega_{l+0} = 64 \times 32$ and $l^* = 3$ are used on $\Omega = (0, 100) \times (0, 50)$. We choose $\Delta t = 0.2$, and $D = 0$. In the precursor model, we use a set of different precursor film heights $b = 0.1$, 0.01, 0.001, and 0.0001. The slipping model uses $C = 0.1$, 0.01, 0.001, and 0.0001 with a fixed $b = 10^{-6}$. In Figs. 12(a) and 12(b), we show the evolution of the average dynamic contact angle obtained by the two models with and without initial perturbation, respectively. As the values of b and



(a)

(b)

Fig. 13 Comparison of profiles (contour at $h = 0.5$) at $T = 80$ with precursor film model and slipping model. (**a**) The initial profile is $h(x, y, 0) = 0.5[h_\infty + b - (h_\infty - b)\tanh(3(x - 5) + \cos(\pi y/25))]$. (**b**) The initial profile is $h(x, y, 0) = 0.5[h_\infty + b - (h_\infty - b)\tanh(3(x - 5))]$

C decrease, the dynamic contact angle increases. In Figs. 13(a) and 13(b), we compare the profiles (contour at $h = 0.5$) given by each model at $T = 80$ with and without initial perturbation, respectively. As can be seen, smaller values of b or C give a faster evolution of the thin-film front. We can also observe that the two methods generate more developed finger shapes when larger values b or C are used.

4.5 Comparison between simulation and experiment of gravity-driven fingering

We compare our numerical simulation results with the experimental data in [24], where silicon oil (surface tension: $\tau = 21$ dyn/cm and density: $\rho = 0.96$ g/cm^3) was used. Since the film thickness H is 0.5–1 mm [24] and the precursor film is thinner than 10^{-3} mm [56]. We use $b = 10^{-4}$ to fit the experimental condition. The computational results were in good agreement with the experimental data for wetting drops using $b \le 10^{-4}$ [57, 58]. $D = (H^2 \rho g \sin\alpha/\tau)^{1/3} \cot\alpha = 0.265$ with $\alpha = \pi/3$, $g = 9.8$ m/s^2, and $H = 0.5$ mm.

To define the initial configuration, we take a part of the experimental photo data from [24] (see top of Fig. 14(a)). Next, we detect the fluid front by an image segmentation technique [51–55]. The goal is to partition a given image into several regions, each of which has homogeneous intensity. For a given image $f_0(\mathbf{x})$ on the image domain Ω, we find the level-set function $\phi(\mathbf{x})$ that satisfies

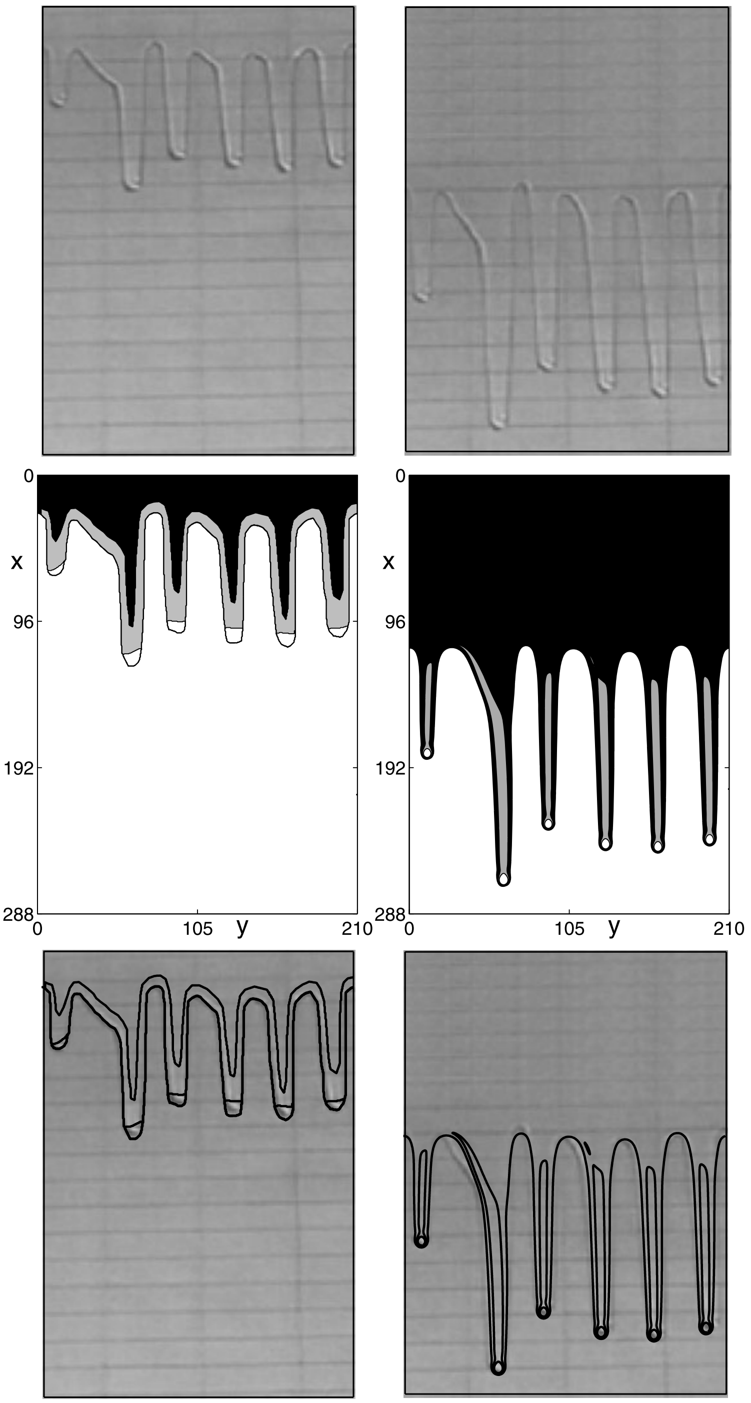
$$\phi(\mathbf{x}) \begin{cases} > 0 & \text{if } \mathbf{x} \in \text{inside } \Gamma, \\ = 0 & \text{if } \mathbf{x} \in \Gamma, \\ < 0 & \text{if } \mathbf{x} \in \text{outside } \Gamma, \end{cases}$$

where Γ is the segmenting curve. The geometric active contour model based on the mean curvature motion is given by the following evolution equation:

$$\phi_t = g(f_0)\left(-\frac{F'(\phi)}{\epsilon^2} + \Delta\phi\right) + \lambda g(f_0) F(\phi), \qquad (14)$$

where $g(f_0)$ is an edge stopping function, $F(\phi) = 0.25(1 - \phi^2)^2$, and λ is a parameter. For more details, please refer to [53].

With the image segmentation technique, we can divide the whole domain into four regions, as shown in the middle of Fig. 14(a). As the initial condition, we set $h(x, y, 0) = h_\infty$ in the black region, $0.8h_{\max}$ in the gray domain, $h_{\max}$ in the nail part of the fingers, and b in the precursor region. Here, $h_{\max}$ is the maximum thickness of the thin film. To obtain $h_{\max}$, we perform a typical simulation with the initial condition $h(x, y, 0) = 0.5[h_\infty + b - (h_\infty - b)\tanh(3(x - 10) + \text{rand}(x, y))]$. Here, $h_\infty = 1$, $b = 10^{-4}$, and $D = 0.265$ are chosen. $l^* = 4$ levels and the level zero domain $\Omega_{l+0} = 64 \times 32$ are used on the computational domain $\Omega = (0, 420) \times (0, 210)$. The calculation is run until $T = 300$ with a time step $\Delta t = 0.1$. We obtain a value of $h_{\max} = 1.61$ and use this to define the initial condition shown in the middle of Fig. 14(a). Figures 14(a) and 14(b) compare the results at $t = 0$ and $t = 65$, respectively. From top to bottom, they show the experimental data, numerical simulation (filled contours in $h = b$, h_∞, $0.8h_{\max}$, and $h_{\max}$), and a superposition of the experimental data and numerical simulation result. Figure 14(b) indicates that the numerical result gives a qualitatively good agreement with the experimental data.



(a) (b)

Fig. 14 Comparison between numerical simulation and experimental data: (**a**) and (**b**) show comparisons at $t = 0$ and $t = 65$, respectively. From top to bottom, they are experimental data, numerical simulation (filled contours in $h = b$, h_∞, $0.8h_{\max}$, and $h_{\max}$), and a superposition of the experimental data and numerical simulation result. Experimental figures are reprinted with permission from L. Kondic, SIAM Review, **45**, 95–115 (2003)

4.6 Multiple thin film simulation

We consider the temporal evolution of multiple thin films. The initial condition is $h(x,y,0) = 0.5[h_\infty + b + (h_\infty - b)\tanh(3(x-10) + 3\cos(\pi y/25))]$ if $x < 10 + l_1 +$. Otherwise, $h(x,y,0) = 0.5[h_\infty + b - (h_\infty - b)\tanh(3(x - 10 - l_1 - l_2) + 3\cos(\pi y/25))]$ on the computational domain $\Omega = (0,800) \times (0,100)$. Here, l_1 is the length between two initial bumps and l_2 is the size of the preceding bump. $\Omega_{l+0} = 128 \times 16$, $l^* = 4$, $\Delta t = 0.1$, and $D = 0$ are used.

Figures 15(a)–15(d) are the evolution of multiple thin-film for $(l_1, l_2) = (100, 150)$, $(100, 300)$, $(200, 150)$, and $(200, 150)$, respectively. From left to right, each column is a snapshot at $t = 0$, 300, 360, 420, and 540. The filled contours are at the levels $h = 0.1$, 0.5, 1, and 1.6. In Fig. 15(a), the evolution of the preceding part becomes slower due to the limited volume of fluid. On the other hand, for the behind bump, fluid is kept supplied through the Dirichlet boundary condition. The front velocity is dependent on a jump of fluid film height across the front, i.e., $(f(h_+) - f(h_-))/(h_+ - h_-)$, where h_+ and h_- are behind and ahead fluid heights of the front, respectively [62, 63]. These lead the following fluid to catch and merge with the preceding fluid front.

To see the effect of the volume of the preceding fluid, we consider a larger l_2, as shown in Fig. 15(b). At time $t = 300$, the finger profile of the following fluid is similar to the previous reference case (a). In this case, it takes more time to catch the preceding fluid front. After the two fronts merge, the fingering profile is similar to the reference case, except for slightly smaller finger sizes. Next, we consider the effect of l_1 on the fingering dynamics (see Fig. 15(c)). At $t = 300$, the fingering pattern of the preceding fluid front is similar to the reference case. However, the finger of the following fluid is more developed compared to the reference case. This phenomenon continues until it merges to the preceding front. After merging, the finger size of the behind front is reduced. The final numerical experiment is a phase shifted initial condition (Fig. 15(d)). That is, for the preceding fluid, $h(x,y,0) = 0.5[h_\infty + b - (h_\infty - b)\tanh(3(x - 10 - l_1 - l_2) + 3\sin(\pi y/25))]$ for $x > 10 + l_1$. Until $t = 480$, the fingering pattern is similar to the case in Fig. 15(c). However, during the merging process, the fronts disrupt each other, which leads to a smaller finger pattern.

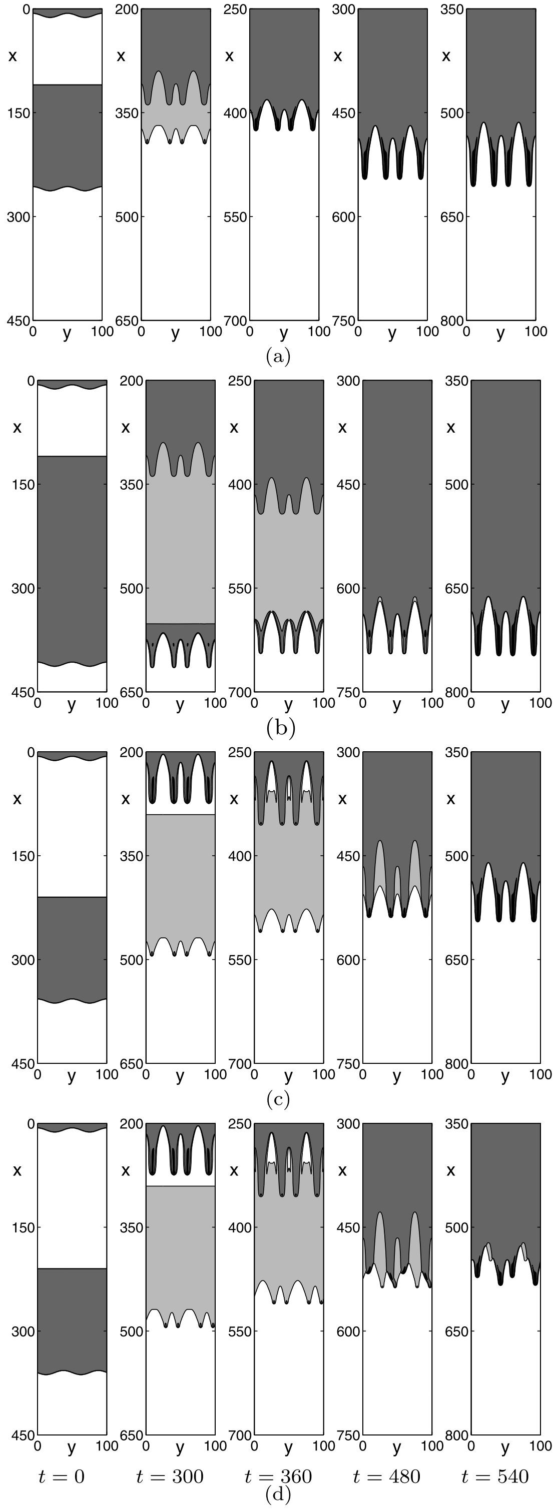


Fig. 15 (**a**)–(**d**) are the evolution of multiple thin-film for $(l_1, l_2) = (100, 150)$, $(100, 300)$, $(200, 150)$, and $(200, 150)$, respectively. The *filled contours* are at the levels $h = 0.1$, 0.5, 1, and 1.6. The times are shown below each column

5 Conclusions

In this paper, we presented a robust and accurate finite difference method for simulating a gravity-driven thin film through numerical investigations with an implicit essentially non-oscillatory scheme. The associated time-dependent governing equation was solved using a nonlinear full approximation storage multigrid algorithm by employing AMR. A set of representative numerical experiments were presented. By comparing the AMR results with those from a uniform mesh method, it was shown that the adaptive multigrid offers increased flexibility together with a significant reduction in memory requirement. A further comparison of our numerical solution with experimental data demonstrated the efficiency of our method.

Acknowledgements The corresponding author (J.S. Kim) was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2011-0023794). The authors greatly appreciate the reviewers for their constructive comments and suggestions, which have improved the quality of this paper.

References

1. Thoroddsen ST, Mahadevan L (1997) Experimental study of coating flows in a partially-filled horizontally rotating cylinder. Exp Fluids 23:1–13
2. Ruschak KJ (1985) Coating flows. Annu Rev Fluid Mech 17:65–89
3. Bertozzi AL, Münch A, Fanton X, Cazabat AM (1998) Contact line stability and “undercompressive shocks” in driven thin film flow. Phys Rev Lett 81:5169–5172
4. Sur J, Bertozzi AL, Behringer RP (2003) Reverse undercompressive shock structures in driven thin film flow. Phys Rev Lett 90:126105
5. Magdy AE, Alla AE, Shereen ME (2013) Stokes’ first problem for a thermoelectric Newtonian fluid. Meccanica 48:1161–1175
6. Huppert HE (1982) Flow and instability of a viscous current down a slope. Nature 300:427–429
7. Troian SM, Herbolzheimer E, Safran SA, Joanny JF (1989) Fingering instability of driven spreading films. Europhys Lett 10:25–30
8. Spaid MA, Homsy GM (1996) Stability of Newtonian and viscoelastic dynamic contact lines. Phys Fluids 8:460–478
9. Bertozzi AL, Brenner MP (1997) Linear stability and transient growth in driven contact lines. Phys Fluids 9:530–539
10. Kataoka DE, Troian SM (1997) A theoretical study of instabilities at the advancing front of thermally driven coating films. J Colloid Interface Sci 15:350–362
11. Diez JA, Kondic L (2001) Contact line instabilities of thin liquid films. Phys Rev Lett 86:632–635
12. Goddard BD, Nold A, Savva N, Pavliotis GA, Kalliadasis S (2012) General dynamical density functional theory for classical fluids. Phys Rev Lett 109:120603
13. Khayat RE, Kim KT, Delosquer S (2004) Influence of inertia, topography and gravity on transient axisymmetric thin-film flow. Int J Numer Methods Fluids 45:391–419
14. Aziz RC, Hashim L, Alomari AK (2011) Thin film flow and heat transfer on an unsteady stretching sheet with internal heating. Meccanica 46:349–357
15. Savva N, Kalliadasis S, Pavliotis A (2010) Two-dimensional droplet spreading over random topographical substrates. Phys Rev Lett 104:084501
16. Savva N, Kalliadasis S (2012) Influence of gravity on the spreading of two-dimensional droplets over topographical substrates. J Eng Math 73:3–16
17. Savva N, Pavliotis A (2009) Two-dimensional droplet spreading over topographical substrates. Phys Fluids 21:092102
18. Sibley DN, Savva N, Kalliadasis S (2012) Slip or not slip a methodical examination of the interface formation model using two-dimensional droplet spreading on a horizontal planar substrate as a prototype system. Phys Fluids 24:082105
19. Christov CI, Pontes J, Walgraef D, Velarde MG (1997) Implicit time-splitting for fourth-order parabolic equations. Comput Methods Appl Mech Eng 148:209–224
20. Karlsen KH, Lie KA (1999) An unconditionally stable splitting for a class of nonlinear parabolic equations. IMA J Numer Anal 19:609–635
21. Daniels N, Ehret P, Gaskell PH, Thompson HM, Decré M (2001) Multigrid methods for thin liquid film spreading flows. In: Proceedings of the first international conference on computational fluid dynamics, pp 279–284
22. Myers TG, Charpin JPF, Chapman SJ (2002) The flow and solidification of a thin fluid film on an arbitrary three-dimensional surface. Phys Fluids 14:2788–2803
23. Momoniat E (2011) Numerical investigation of a third-order ODE from thin film flow. Meccanica 46:313–323
24. Kondic L (2003) Instabilities in gravity driven flow of thin fluid films. SIAM Rev 45:95–115
25. Witelski TP, Bowen M (2003) ADI schemes for higher-order nonlinear diffusion equations. Appl Numer Math 45:331–351
26. Gaskell PH, Jimack PK, Sellier M, Thompson HM (2004) Efficient and accurate time adaptive multigrid simulations of droplet spreading. Int J Numer Methods Fluids 14:1161–1186
27. Gaskell PH, Jimack PK, Sellier M, Thompson HM, Wilson MCT (2004) Gravity-driven flow of continuous thin liquid films on non-porous substrates with topography. J Fluid Mech 509:253–280
28. Gaskell PH, Jimack PK, Sellier M, Thompson HM (2006) Flow of evaporating, gravity-driven thin liquid films over topography. Phys Fluids 18:031601
29. Kim J, Sur J (2005) A hybrid method for higher-order nonlinear diffusion equations. Commun Korean Math Soc 20:179–193
30. Myers TG, Lombe M (2006) The importance of the Coriolis force on axisymmetric horizontal rotating thin film flows. Chem Eng Process 45:90–98
31. Kim J (2006) Adaptive mesh refinement for thin-film equations. J Korean Phys Soc 49:1903–1907

32. Lee YC, Thompson HM, Gaskell PH (2007) An efficient adaptive multigrid algorithm for predicting thin film flow on surfaces containing localised topographic features. Comput Fluids 37:838–855
33. Vellingiri R, Savva N, Kalliadasis S (2011) Droplet spreading on chemically heterogeneous substrates. Phys Rev E 84:036305
34. Sun P, Russell RD, Xu J (2007) A new adaptive local mesh refinement algorithm and its application on fourth order thin film flow problem. J Comput Phys 224:1021–1048
35. Ha Y, Kim YJ, Myers TG (2008) On the numerical solution of a driven thin film equation. J Comput Phys 227:7246–7263
36. Lee YC, Thompson HM, Gaskell PH (2008) The efficient and accurate solution of continuous thin film flow over surface patterning and past occlusions. Int J Numer Methods Fluids 56:1375–1381
37. Sellier M, Lee YC, Thompson HM, Gaskell PH (2009) Thin film flow on surfaces containing arbitrary occlusions. Comput Fluids 38:171–182
38. Wang G, Rothmayer AP (2009) Thin water films driven by air shear stress through roughness. Comput Fluids 38:235–246
39. Sellier M, Panda S (2010) Beating capillarity in thin film flows. Int J Numer Methods Fluids 63:431–448
40. Veremieiev S, Thompson HM, Lee YC, Gaskell PH (2010) Inertial thin film flow on planar surfaces featuring topography. Comput Fluids 39:431–450
41. Lee YC, Thompson HM, Gaskell PH (2011) Dynamics of thin film flow on flexible substrate. Chem Eng Process 50:525–530
42. Li Y, Lee H-G, Yoon D, Hwang W, Shin S, Ha Y, Kim JS (2011) Numerical studies of the fingering phenomena for the thin film equation. Int J Numer Methods Fluids 67:1358–1372
43. Hu B, Kieweg SL (2012) The effect of surface tension on the gravity-driven thin film flow of Newtonian and power-law fluids. Comput Fluids 64:83–90
44. Shu CW, Osher S (1989) Efficient implementation of essentially non-oscillatory shock capturing schemes II. J Comput Phys 83:32–78
45. Almgren AS, Bell JB, Colella P, Howell LH, Welcome ML (1998) A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations. J Comput Phys 142:1–6
46. Berger MJ, Rigoutsos I (1991) An algorithm for point clustering and grid generation. IEEE Trans Syst Man Cybern 21:1278–1286
47. The Mathworks, Inc., Matlab. http://www.mathworks.com
48. Trottenberg U, Oosterlee C, Schüller A (2001) Multigrid. Academic Press, London
49. Goodwin R, Homsy GM (1991) Viscous flow down a slope in the vicinity of a contact line. Phys Fluids A 3:515–528
50. Lin TS, Kondic L (2010) Thin films flowing down inverted substrates: two dimensional flow. Phys Fluids 22:052105
51. Caselles V, Catte F, Coll T, Dibos F (1993) A geometric model for active contours in image processing. Numer Math 66:1–31
52. Chan T, Vese L (2001) Active contours without edges. IEEE Trans Image Process 2:266–277
53. Li Y, Kim J (2010) A fast and accurate numerical method for medical image segmentation. J Korea SIAM 14:201–210
54. Li Y, Kim J (2011) Multiphase image segmentation with a phase-field model. Comput Math Appl 62:737–745
55. Li Y, Kim J (2012) An unconditionally stable numerical method for bimodal image segmentation. Appl Math Comput 219:3083–3090
56. Diez JA, Kondic L, Bertozzi A (2000) Global models for moving contact lines. Phys Rev E 63:011208
57. Gratton R, Diez JA, Thomas LP, Marino B, Betelu S (1996) Quasi-self-similarity for wetting drops. Phys Rev E 53:3563–3572
58. Cazabat AM, Stua MAC (1986) Dynamics of wetting: effects of surface roughness. J Phys Chem 90:5845–5849
59. Mourik SV (2002) Numerical modelling of the dynamic contact angle. Master's thesis, University of Groningen
60. Bonn D, Eggers J, Indekeu J, Meunier J, Rolley E (2009) Wetting and spreading. Rev Mod Phys 81:739–805
61. Savva N, Kalliadasis S (2011) Dynamics of moving contact lines: a comparison between slip and precursor film models. Europhys Lett 94:64004
62. Bertozzi AL, Shearer M (2000) Existence of undercompressive traveling waves in thin film equations. SIAM J Math Anal 32:194–213
63. Levy R, Shearer M (2004) Comparison of two dynamic contact line models for driven thin liquid films. Eur J Appl Math 15:625–642