

A Numerical Scheme for Particle-Laden Thin Film Flow in 2-D

Matthew R. Mata^{a,*}, Andrea L. Bertozzi^a

^a*Department of Mathematics, University of California Los Angeles, 520 Portola Plaza, Los Angeles, California, 90095-1555*

Abstract

The physics of particle-laden thin film flow are not fully understood, and recent experiments have raised questions with current theory. There is a need for fully 2-D simulations to compare with experimental data. To this end, a numerical scheme is presented for a lubrication model derived for particle-laden thin film flow in two dimensions with surface tension. The scheme relies on an ADI process to handle the higher-order terms, and an iterative procedure to improve the solution at each timestep. This is the first paper to simulate the 2-D particle-laden thin film lubrication model. Several aspects of the scheme are examined for a test problem, such as the timestep, runtime, and number of iterations. The results from the simulation are compared to experimental data. The simulation shows good qualitative agreement. It also suggests further lines of inquiry for the physical model.

Keywords: adaptive timestepping, alternating direction implicit, coupled system, fourth order, particle-laden, thin film

1. Introduction

In recent years, the problem of numerically solving gravity-driven thin film flow for clear fluids has had ample work done in both one and two dimensions. However, the case when the film contains particles suspended within it has received less attention, especially in two dimensions. The evolution of a clear fluid down an inclined plane is modeled using a single partial differential equation and numerical schemes have been derived using finite differences [15] and finite elements [31]. For similar equations, such as spreading thin films, there are methods for finite elements in one dimension [9, 10, 36] and for finite differences in two dimensions [34]. The incorporation of particles into such a flow leads to another variable in the model, namely the particle concentration, and an accompanying equation related to the evolution of the particles. The result is a system of equations that requires a different approach from the clear-fluid case to formulate a practical numerical scheme.

An active area of research in the last decade has been the development of numerical methods for higher-order thin film equations including complex fluids described by systems of equations. Related problems include methods for coupled systems of nonlinear parabolic equations [21, 25]. The scheme presented here is, in part, inspired by recent models for surfactants [33] and thin films [34]. We choose an Alternating Direction Implicit (ADI) scheme as a tractable method for implicit

*Principal corresponding author

Email addresses: matthewmata@math.ucla.edu (Matthew R. Mata), bertozzi@math.ucla.edu (Andrea L. Bertozzi)

18 timesteps, because surface tension introduces a severe restriction on the timestep in the case of
19 explicit schemes. While ADI schemes for numerically solving parabolic equations date back to the
20 1950's [26], their use in higher-order problems is rather new, e.g., [34], and not all that well-studied.
21 However, the ease of parallelization makes such schemes a very viable choice for multiprocessor
22 platforms. Since their inception, ADI schemes have been extended to handle parabolic problems
23 with mixed derivative terms [2, 8, 23, 29], variable coefficients [14, 34], and high-order terms [34].

24 The ideas present in these schemes can be combined to create an efficient way to numerically
25 solve the particle-laden thin film flow equations. The nonlinearity and higher-order terms are han-
26 dled in a similar manner to Witelski and Bowen [34], which dealt with thin-film equations, and
27 the remaining terms are treated as in Warner et al. [33], which devised a semi-implicit scheme
28 for surfactants. This combined approach is fine-tuned to draw out better efficiency, via adaptive
29 timestepping and an iterative procedure within each timestep. At the cost of the extra calcula-
30 tions due to the iterative nature of the scheme, the timestep needed for stability can be improved
31 over recent methods. The result is an efficient method to simulate the continuum model in two
32 dimensions.

33 The full physics of particle-laden thin film flow is not well understood. Recent experiments,
34 and their comparison to the model, have raised questions. We present such a comparison in this
35 paper, where the results show qualitative agreement. In particular, by performing 2-D simulations,
36 we are able to observe finger formation and compare directly with experiments. There is a need
37 for accurate, fully 2-D simulations of the model for comparison with experimental data, such as in
38 the case of mudslides and oil spills.

39 The paper is organized as follows: Section 2 presents the system of evolution equations for
40 the flow. In Section 3, the numerical scheme for this system is derived. Section 4 covers the
41 adaptive timestepping scheme implemented in the code. A complete explanation of the spatial
42 discretization is given in Section 5. The practicality and implementation of a moving reference
43 frame in the simulations are discussed in Section 6. Numerical simulations are presented in Section
44 7. We compare the results generated from the numerical scheme to an experiment using silicone
45 oil and glass beads in Section 8. Finally, in Section 9, we provide a discussion of the results and
46 future work.

47 2. Model

48 The results from experiments indicate that particle-laden thin film flows exhibit three distinct
49 regimes, based on the initial particle concentration and angle of inclination [35]. For low concentra-
50 tions and angles, the particles settle to the substrate with clear fluid flowing over top of them. The
51 behavior after sedimentation is similar to clear-fluid experiments, such as those done by Huppert
52 [13]. High concentrations and angles cause a particle-rich ridge to emerge at the front of the flow.
53 Medium concentrations and angles lead to a particle concentration which appears to stay well-
54 mixed throughout the duration of the experiment. Based on Cook [5], this behavior likely belongs
55 to one of the two previously mentioned regimes but may not have evolved to the point where this
56 distinction can be made.

57 The evolution equations for the flow are based on the regime where the inclination angle and
58 particle concentration are both high enough to induce the formation of a particle-rich ridge. It is
59 formulated in terms of the thickness of the film, h , and the particle concentration by volume, ϕ (see
60 Figure 1). The equations for modeling this regime were first derived in Zhou et al. [35]; re-derived
61 in Cook et al. [6], using conservation of volume rather than mass; and modified in Cook et al. [7],

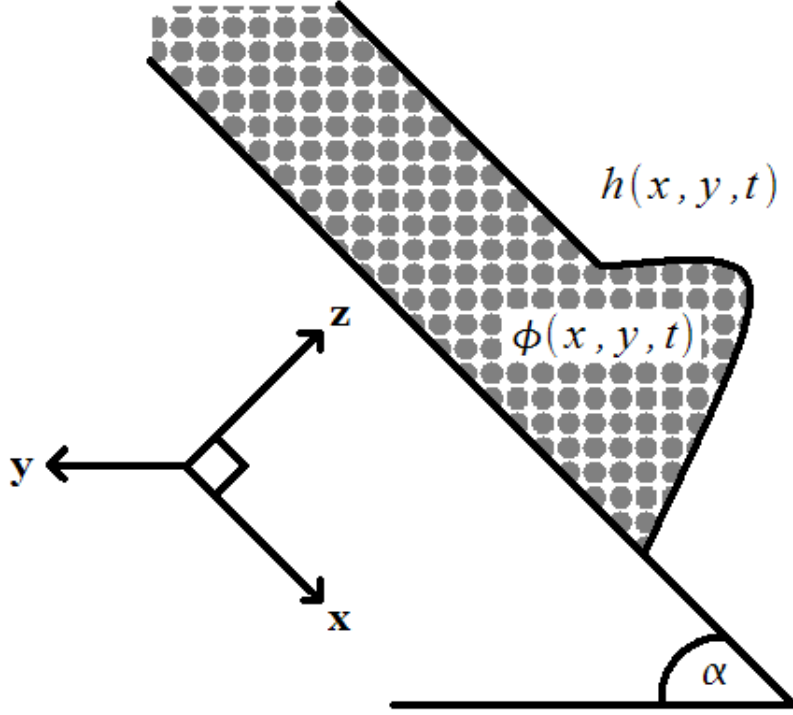


Figure 1: The coordinate system and variables considered in this problem.

62 adding in a shear-induced diffusion term to correct for an instability affecting ϕ . The dimensionless
 63 system [7] is

$$h_t + \nabla \cdot (h v_{av}) = 0, \quad (1)$$

$$(\phi h)_t + \nabla \cdot [\phi h (v_{av} + (1 - \phi)v_{rel}) - F_{diff}] = 0. \quad (2)$$

64 The orientation for (1)-(2) is such that \mathbf{x} lies in the plane and is parallel to the direction of the
 65 flow, \mathbf{y} is across the inclined plane and perpendicular to \mathbf{x} , and \mathbf{z} is normal to the plane.

66 The volume-averaged velocity of the liquid and the particles together is

$$v_{av} = \frac{h^2}{\mu(\phi)} \nabla \nabla^2 h - D(\alpha) \left[\frac{h^2}{\mu(\phi)} \nabla (\rho(\phi)h) - \frac{5}{8} \frac{h^3}{\mu(\phi)} \nabla (\rho(\phi)) \right] + \frac{\rho(\phi)}{\mu(\phi)} h^2 \hat{\mathbf{x}}, \quad (3)$$

67 where the terms in (3) come from surface tension, the effects of gravity normal to the inclined
 68 plane, and the effects of gravity parallel to the inclined plane.

69 The density of the fluid as a whole is $\rho(\phi) = 1 + \Delta\phi$; $\Delta = \frac{\rho_p - \rho_l}{\rho_l}$ is the difference in the densities
 70 between the particles and the liquid. The function $\mu(\phi) = (1 - \phi/\phi_{max})^{-2}$ [17, 30] is the effective
 71 fluid viscosity, where ϕ_{max} is the maximum packing fraction of particles, assuming the particles
 72 are spheres. For this problem, the maximum packing fraction has been empirically determined to
 73 be 0.58, while the theoretical value is 0.64 [32]. $D(\alpha) = (3Ca)^{1/3} \cot(\alpha)$ [3] is a modified capillary
 74 number, where Ca is the capillary number of the liquid and α is the angle of inclination of the

75 plane on which the fluid is flowing ($\alpha = 0$ corresponds to the plane being horizontal while $\alpha = \pi/2$
76 to vertical).

77 The settling velocity of the particles, relative to the velocity of the liquid, is a combination of
78 three factors, assumed to be multiplicative,

$$v_{rel} = V_s f(\phi) w(h) \hat{\mathbf{x}}. \quad (4)$$

79 The coefficient $V_s = \frac{2}{3} a^2 \Delta$ in (4) is the Stokes settling velocity of a single sphere settling in a
80 viscous liquid, where a is the dimensionless particle radius. A hindered settling function, in this
81 case the Richardson-Zaki function $f(\phi) = (1 - \phi)^5$ [28], accounts for the effect of sedimentation.
82 The particles settling parallel to the substrate is modeled using a wall effects function, $w(h) =$
83 $A(h/a)^2 / \sqrt{1 + (A(h/a)^2)^2}$ with $A = 1/18$. This function is an approximation to a method of
84 images solution to a single sphere falling parallel to a vertical wall [12]. This has the property that
85 it is near 0 for h small and near 1 for h large.

86 Since (3) contains higher-order terms but (4) does not, v_{rel} is not regularized. This leads to an
87 instability affecting the particle concentration in numerical simulations [7]. To correct for this, a
88 shear-induced diffusion term (5) was added in,

$$F_{diff} = \frac{3}{2} a^2 (3Ca)^{1/3} \hat{D}(\phi) \frac{h^2 \rho(\phi)}{\mu(\phi)} \nabla \phi. \quad (5)$$

89 This behavior can be seen in a 1-D example on the domain $x : 0 - 50$ with $\Delta x = 0.05$. The
90 initial film thickness is a jump, from 1 to 0.05, smoothed by hyperbolic tangent. The initial particle
91 concentration is taken to be $\phi = 0.3$. This simulation is similar to those described in Section 7, and
92 a moving reference frame is used, as discussed in Section 6. By time $t = 1000$, the solution without
93 the extra diffusion term has developed an instability (Figure 2) while the one with it is still stable
94 (Figure 3).

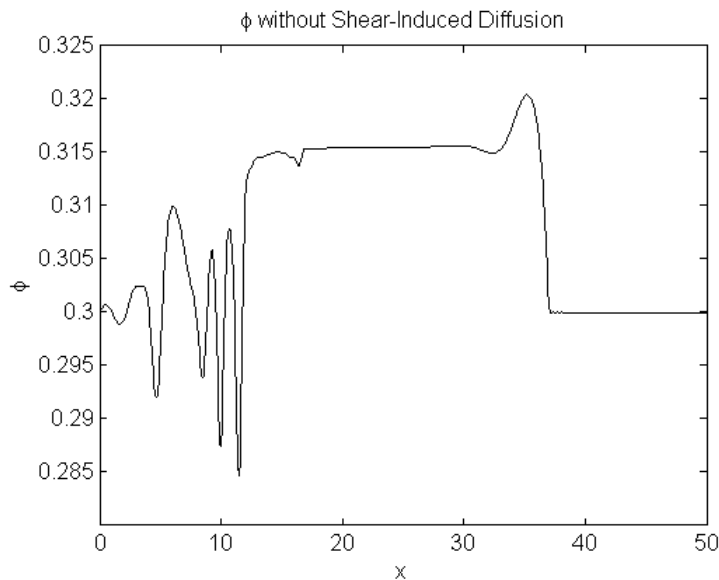


Figure 2: The numerical solution of ϕ at time $t = 1000$ without shear-induced diffusion. By this time, an instability has developed.

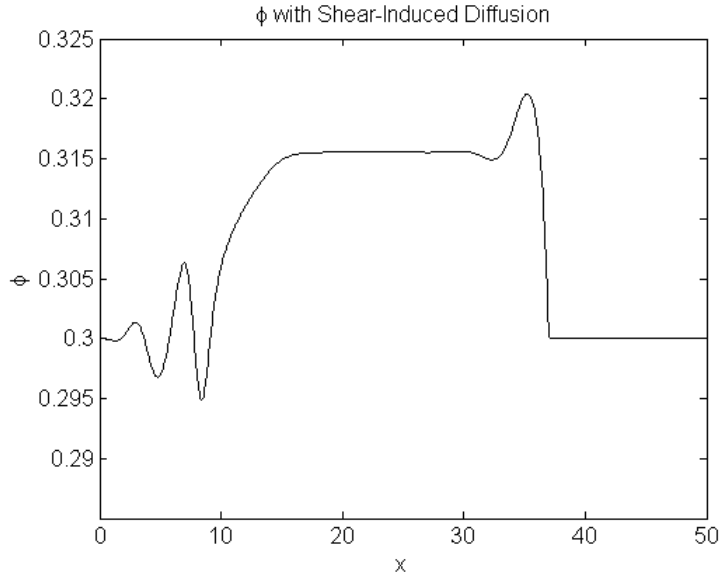


Figure 3: The numerical solution ϕ at time $t = 1000$ with shear-induced diffusion (5). The solution is still stable due to the extra term.

95 Equation (5) accounts for horizontal diffusion of particles in the fluid caused by horizontal gradients
 96 of ϕ and was derived based on results from Leighton [19] and Leighton and Acrivos [20]. The term
 97 $\hat{D}(\phi) = (1/3)\phi^2 (1 + (1/2)e^{8.8\phi})$ is a dimensionless diffusion coefficient.

98 3. Numerical Scheme

99 In the case of a gravity-driven clear fluid flow, the model reduces to a single equation [3] for the
 100 film thickness, h ,

$$h_t + (h^3)_x + \nabla \cdot (h^3 \nabla \nabla^2 h - D(\alpha) h^3 \nabla h) = 0. \quad (6)$$

101 Solving (6), and similar problems, numerically in one and two dimensions has been done using
 102 several different methods [1, 15, 22, 31, 34]. Including particles in the physics not only adds a
 103 second equation, but couples it to the equation for the film thickness. The particle-laden case
 104 has been solved numerically in one dimension with methods such as forward Euler with upwind
 105 differencing [35] and the Lax-Friedrichs method [6] when the high-order terms are omitted, and
 106 backward Euler with centered differencing [35] when the terms are included.

107 This system of PDE's in two dimensions poses numerical difficulties beyond those present in the
 108 clear-fluid problem. For both the clear and particle-laden cases, fully explicit schemes typically have
 109 the problem that an $O(\Delta x^4)$ timestep, assuming $\Delta x = \Delta y$, is needed for stability. One solution is
 110 to use an implicit scheme. For the clear-fluid and similar problems, the nonlinearity combined with
 111 an implicit scheme amounts to solving the problem at each timestep using an iterative process, such
 112 as Newton's method, to converge to the solution [34]. For the particle-laden case, using an implicit
 113 scheme typically requires that both equations be solved simultaneously, using an iterative process
 114 to account for the nonlinearity. This results in a linear algebra problem with twice the number
 115 of unknowns and a matrix that is twice as large in each dimension, compared to the clear-fluid
 116 problem. Therefore, solving the particle-laden case leads to larger linear algebra problems to solve

117 at each timestep and the matrix from Newton's method will have a more complex structure than
 118 for clear fluids.

119 The goal of the scheme presented here is to circumvent some of the aforementioned difficulties.
 120 The advantages of this approach, over a purely explicit scheme or implicit with Newton's method,
 121 is that the timestep is more lenient than for a fully explicit scheme and the linear algebra problem
 122 that results from the implicit part of the scheme is reduced to a series of smaller banded matrix
 123 solves, which can be done efficiently and independently for each equation.

124 The numerical scheme that we employ for the particle-laden thin film flow problem is inspired by
 125 the schemes presented in Witelski and Bowen [34] for higher-order parabolic PDE's and Warner et
 126 al. [33] for surfactants. In Witelski and Bowen, an ADI scheme is derived for solving the nonlinear
 127 PDE known as the thin film equation,

$$h_t + \nabla \cdot (f(h)\nabla\nabla^2h) = 0. \quad (7)$$

128 The ADI scheme for (7) is backward Euler in time and uses approximate values of h in the nonlinear
 129 and mixed-derivative implicit terms. It is suggested to start with approximations, such as time-
 130 lagged values, for evaluating these terms and calculating the numerical solution at the timestep.
 131 Then use this solution for the new approximate values within the same timestep and recalculate.
 132 This results in an iterative scheme at each timestep. However, for solving the thin film equation,
 133 it was noted that the iterations did not provide a noticeable improvement. Warner et al. use
 134 this method for a coupled system of nonlinear PDE's relating to surfactants. They handle the
 135 higher-order terms implicitly using Crank-Nicholson, and apply ADI to this. The remaining terms,
 136 which are at least second-order in space, are treated explicitly. For the nonlinearity and mixed-
 137 derivative terms, the values are time-lagged and the problem is solved only once per timestep. In
 138 the simulations, $\Delta x = \Delta y = \pi/100 \approx 0.0314$ required a timestep of $O(10^{-5})$.

139 Our approach is to handle applicable terms implicitly, using ADI, and treat the remaining terms
 140 explicitly, as we show below. Iterations within each timestep allow for a larger Δt to be taken at
 141 the cost of some extra calculations. In general, the increase in the size of the timestep outweighs
 142 the extra computational work, as shown in Section 7.

143 For equation (1), the terms

$$\nabla \cdot \left(\frac{h^3}{\mu(\phi)} \nabla \nabla^2 h + \frac{\rho(\phi)}{\mu(\phi)} h^3 \hat{\mathbf{x}} \right) \quad (8)$$

144 can be handled implicitly. This is because the spatial derivatives on these terms are applied to
 145 h . Including the first-order terms in the implicit treatment allows them to be discretized spatially
 146 using centered differencing to maintain stability. First discretize the terms in (8) in time with
 147 backward Euler, including the time derivative,

$$h^{n+1} + \Delta t \nabla \cdot \left(\frac{h^3}{\mu(\phi)} \nabla \nabla^2 h + \frac{\rho(\phi)}{\mu(\phi)} h^3 \hat{\mathbf{x}} \right)^{n+1} = h^n. \quad (9)$$

148 Write out the operators in (9) fully,

$$\begin{aligned} & h^{n+1} + \Delta t \left[\partial_x \left(\frac{h^3}{\mu(\phi)} h_{xxx} \right) + \partial_y \left(\frac{h^3}{\mu(\phi)} h_{yyy} \right) \right. \\ & \left. + \partial_x \left(\frac{\rho(\phi)}{\mu(\phi)} h^3 \right) \right]^{n+1} + \Delta t \left[\partial_x \left(\frac{h^3}{\mu(\phi)} h_{yyx} \right) + \partial_y \left(\frac{h^3}{\mu(\phi)} h_{xxy} \right) \right]^{n+1} = h^n. \end{aligned} \quad (10)$$

149 The idea behind the ADI approach is to reduce the implicit part of (10), with derivatives in
 150 both x and y , to a product of two operators, each with only derivatives in either x or y . To achieve
 151 this, the terms involving only x -derivatives and only y -derivatives are grouped together. Define the
 152 operators

$$D_x = \partial_x \left(\frac{h^3}{\mu(\phi)} \partial_{xxx} + \frac{\rho(\phi)}{\mu(\phi)} h^2 I \right)^{n+1}, \quad D_y = \partial_y \left(\frac{h^3}{\mu(\phi)} \partial_{yyy} \right)^{n+1}. \quad (11)$$

153 Then replacing the terms in (10) with the definitions in (11), we have

$$\begin{aligned} & h^{n+1} + \Delta t (D_x + D_y) h^{n+1} \\ & + \Delta t \left[\partial_x \left(\frac{h^3}{\mu(\phi)} h_{yyx} \right) + \partial_y \left(\frac{h^3}{\mu(\phi)} h_{xxy} \right) \right]^{n+1} = h^n. \end{aligned} \quad (12)$$

154 In order to obtain an ADI scheme from (12), note that $I + \Delta t D_x + \Delta t D_y = (I + \Delta t D_x)(I +$
 155 $\Delta t D_y) - (\Delta t)^2 D_x D_y$ and so the left-hand side, with the addition of an $O(\Delta t^2)$ term, can be written
 156 as a product of two one-dimensional operators.

$$\begin{aligned} & (I + \Delta t D_x)(I + \Delta t D_y) h^{n+1} - (\Delta t)^2 D_x D_y h^{n+1} \\ & + \Delta t \left[\partial_x \left(\frac{h^3}{\mu(\phi)} h_{yyx} \right) + \partial_y \left(\frac{h^3}{\mu(\phi)} h_{xxy} \right) \right]^{n+1} = h^n. \end{aligned} \quad (13)$$

157 To handle the nonlinear terms, which occur in front of derivatives, and mixed-derivative terms in
 158 (13), define them as approximate, denoted by a tilde (e.g., \tilde{h}^{n+1}). The approximate terms can be
 159 chosen in some reasonable manner, such as time-lagged or extrapolated. This will be discussed
 160 in more detail later. Subtract the mixed-derivative terms from and add the $O(\Delta t^2)$ term to both
 161 sides. This leaves a scheme in which all the terms operating on h^{n+1} are known, as is the entire
 162 right-hand side.

$$\begin{aligned} & (I + \Delta t \tilde{D}_x)(I + \Delta t \tilde{D}_y) h^{n+1} = h^n \\ & + \left\{ (\Delta t)^2 \tilde{D}_x \tilde{D}_y - \Delta t \left[\partial_x \left(\frac{\tilde{h}^3}{\mu(\tilde{\phi})} \partial_{yyx} \right) + \partial_y \left(\frac{\tilde{h}^3}{\mu(\tilde{\phi})} \partial_{xxy} \right) \right] \right\}^{n+1} \tilde{h}^{n+1}. \end{aligned} \quad (14)$$

163 For simplicity, define the operators in (14) as

$$\tilde{L}_x = I + \Delta t \tilde{D}_x, \quad \tilde{L}_y = I + \Delta t \tilde{D}_y.$$

164 Subtracting $\tilde{L}_x \tilde{L}_y \tilde{h}^{n+1}$ from both sides of (14), which cancels the $O(\Delta t^2)$ term, yields

$$\tilde{L}_x \tilde{L}_y \left(h^{n+1} - \tilde{h}^{n+1} \right) = - \left(\tilde{h}^{n+1} - h^n \right) - \Delta t \nabla \cdot \left(\frac{\tilde{h}^3}{\mu(\tilde{\phi})} \nabla \nabla^2 \tilde{h} + \frac{\rho(\tilde{\phi})}{\mu(\tilde{\phi})} \tilde{h}^3 \right)^{n+1}. \quad (15)$$

165 At this point, the implicit part of the scheme is complete and the explicit terms can be added back
 166 into (15) using forward Euler.

$$\begin{aligned} \tilde{L}_x \tilde{L}_y (h^{n+1} - \tilde{h}^{n+1}) = & - \left(\tilde{h}^{n+1} - h^n \right) - \Delta t \nabla \cdot \left(\frac{\tilde{h}^3}{\mu(\tilde{\phi})} \nabla \nabla^2 \tilde{h} + \frac{\rho(\tilde{\phi})}{\mu(\tilde{\phi})} \tilde{h}^3 \right)^{n+1} \\ & + \Delta t \nabla \cdot \left\{ D(\alpha) \left[\frac{h^3}{\mu(\phi)} \nabla (\rho(\phi)h) - \frac{5}{8} \frac{h^4}{\mu(\phi)} \nabla (\rho(\phi)) \right] \right\}^n. \end{aligned} \quad (16)$$

167 Define

$$u = h^{n+1} - \tilde{h}^{n+1},$$

168 which can be thought of as a correction term to the approximation of h^{n+1} , and (16) can be written
169 as a three-step process: two one-directional solves (17)-(18) and an update step (19).

$$\begin{aligned} \tilde{L}_x v = & - \left(\tilde{h}^{n+1} - h^n \right) - \Delta t \nabla \cdot \left(\frac{\tilde{h}^3}{\mu(\tilde{\phi})} \nabla \nabla^2 \tilde{h} + \frac{\rho(\tilde{\phi})}{\mu(\tilde{\phi})} \tilde{h}^3 \right)^{n+1} \\ & + \Delta t \nabla \cdot \left\{ D(\alpha) \left[\frac{h^3}{\mu(\phi)} \nabla (\rho(\phi)h) - \frac{5}{8} \frac{h^4}{\mu(\phi)} \nabla (\rho(\phi)) \right] \right\}^n, \end{aligned} \quad (17)$$

$$\tilde{L}_y u = v, \quad (18)$$

$$h^{n+1} \approx \tilde{h}^{n+1} + u. \quad (19)$$

170 Since the operators \tilde{L}_x and \tilde{L}_y involve at most fourth-order terms, the spatial discretization of
171 them will lead to a five-point stencil in the x - and y -direction, respectively. This discretization is
172 discussed fully in Section 5. Along each row/column of the discretized domain, this results in a
173 pentadiagonal linear algebra problem. This can be solved using a pentadiagonal solver, or a more
174 generic banded matrix solver.

175 To help with the inaccuracy in the nonlinear and mixed-derivative terms resulting from approx-
176 imation, an iterative procedure can be used at each timestep to improve the solution and size of the
177 timestep. This was first suggested for the ADI scheme in the context of thin film equations [34].
178 This procedure amounts to repeating the three-step process associated with solving each equation
179 at each timestep and updating the approximate solution with the most recent solution, until the
180 new and approximate solutions sufficiently converge. This is similar to fixed-point iteration.

181 For the first equation, when entering the timestep, a choice must be made as to the value of \tilde{h}^{n+1}
182 and $(\tilde{\phi}\tilde{h})^{n+1}$. Using h as an example, two reasonable choices would be a time-lagged approximation,
183 h^n , which is a first-order accurate approximation in time, or an extrapolated approximation, $2h^n -$
184 h^{n-1} , which is second-order in time. For adaptive timestepping, this extrapolation is given by
185 $h^n + (\Delta t / \Delta t_{old})(h^n - h^{n-1})$, where Δt is the prospective timestep between t^n and t^{n+1} and Δt_{old}
186 is the timestep between t^{n-1} and t^n . While the second choice of an approximation is second-order,
187 it also requires storing an extra set of data, namely h^{n-1} . With this choice made, the three-step
188 process can be implemented, obtaining a solution, h^{n+1} . At this point, the approximation can
189 be redefined, $\tilde{h}^{n+1} = h^{n+1}$, and the process run again. This can be continued until convergence
190 between the approximate and new solution, or equivalently when the correction term u is small in
191 a chosen norm.

192 For (2), the ADI method is applied to ϕh as a whole, since the time derivative is on this term.
193 The applicable terms in the equation are

$$\nabla \cdot \left[-D(\alpha) \left(\Delta \frac{(\phi h) h^2}{\mu(\phi)} \nabla(\phi h) \right) + \phi h \left(\frac{\rho(\phi)}{\mu(\phi)} h^2 + (1 - \phi) V_s f(\phi) w(h) \right) \hat{\mathbf{x}} \right]. \quad (20)$$

194 As with (1), the time discretization of (20) is based on a backward Euler method

$$\begin{aligned} & (\phi h)^{n+1} + \Delta t \nabla \cdot \left[-D(\alpha) \left(\Delta \frac{(\phi h) h^2}{\mu(\phi)} \nabla(\phi h) \right) \right. \\ & \left. + \phi h \left(\frac{\rho(\phi)}{\mu(\phi)} h^2 + (1 - \phi) V_s f(\phi) w(h) \right) \hat{\mathbf{x}} \right]^{n+1} = (\phi h)^n. \end{aligned} \quad (21)$$

195 Writing out the operators in (21) explicitly,

$$\begin{aligned} & (\phi h)^{n+1} - \Delta t D(\alpha) \Delta \left[\partial_x \left(\frac{(\phi h) h^2}{\mu(\phi)} \partial_x(\phi h) \right) + \partial_y \left(\frac{(\phi h) h^2}{\mu(\phi)} \partial_y(\phi h) \right) \right]^{n+1} \\ & + \Delta t \partial_x \left[\phi h \left(\frac{\rho(\phi)}{\mu(\phi)} h^2 + (1 - \phi) V_s f(\phi) w(h) \right) \right]^{n+1} = (\phi h)^n. \end{aligned} \quad (22)$$

196 Define the operators in (22) involving only x -derivatives and only y -derivatives as \mathcal{D}_x and \mathcal{D}_y ,
197 respectively.

$$\begin{aligned} \mathcal{D}_x &= -D(\alpha) \Delta \partial_x \left(\frac{(\phi h) h^2}{\mu(\phi)} \partial_x \right)^{n+1} + \partial_x \left(\left[\frac{\rho(\phi)}{\mu(\phi)} h^2 (1 - \phi) V_s f(\phi) w(h) \right] I \right)^{n+1}, \\ \mathcal{D}_y &= -D(\alpha) \Delta \partial_y \left(\frac{(\phi h) h^2}{\mu(\phi)} h^2 \partial_y \right)^{n+1}. \end{aligned} \quad (23)$$

198 Using (23), the equation can be compactly written as

$$(\phi h)^{n+1} + \Delta t (\mathcal{D}_x + \mathcal{D}_y) (\phi h)^{n+1} = (\phi h)^n. \quad (24)$$

199 Note that there are no mixed-derivative terms to handle in (24). The left-hand side can be written
200 as the product of two one-dimensional operators, incurring an $O(\Delta t^2)$ term in the process.

$$(I + \Delta t \mathcal{D}_x) (I + \Delta t \mathcal{D}_y) (\phi h)^{n+1} - (\Delta t)^2 \mathcal{D}_x \mathcal{D}_y (\phi h)^{n+1} = (\phi h)^n. \quad (25)$$

201 Add the $O(\Delta t^2)$ term to both sides of (25), and make all terms that occur nonlinearly at time
202 t^{n+1} approximate, as before.

$$(I + \Delta t \tilde{\mathcal{D}}_x) (I + \Delta t \tilde{\mathcal{D}}_y) (\phi h)^{n+1} = (\phi h)^n + (\Delta t)^2 \tilde{\mathcal{D}}_x \tilde{\mathcal{D}}_y (\tilde{\phi} \tilde{h})^{n+1}. \quad (26)$$

203 Define

$$\tilde{\mathcal{L}}_x = I + \Delta t \tilde{\mathcal{D}}_x, \quad \tilde{\mathcal{L}}_y = I + \Delta t \tilde{\mathcal{D}}_y$$

204 and subtract $\tilde{\mathcal{L}}_x \tilde{\mathcal{L}}_y (\tilde{\phi} \tilde{h})^{n+1}$ from both sides of (26) to obtain

$$\begin{aligned} & \tilde{\mathcal{L}}_x \tilde{\mathcal{L}}_y \left((\phi h)^{n+1} - (\tilde{\phi} \tilde{h})^{n+1} \right) = - \left((\tilde{\phi} \tilde{h})^{n+1} - (\phi h)^n \right) \\ & - \Delta t \nabla \cdot \left[-D(\alpha) \left(\Delta \frac{(\tilde{\phi} \tilde{h}) \tilde{h}^2}{\mu(\tilde{\phi})} \nabla(\tilde{\phi} \tilde{h}) \right) + \tilde{\phi} \tilde{h} \left(\frac{\rho(\tilde{\phi})}{\mu(\tilde{\phi})} \tilde{h}^2 + (1 - \tilde{\phi}) V_s f(\tilde{\phi}) w(\tilde{h}) \right) \hat{\mathbf{x}} \right]^{n+1}. \end{aligned} \quad (27)$$

205 The remaining terms can be incorporated into (27) via forward Euler.

$$\begin{aligned}
& \tilde{\mathcal{L}}_x \tilde{\mathcal{L}}_y \left((\phi h)^{n+1} - (\tilde{\phi} \tilde{h})^{n+1} \right) = - \left((\tilde{\phi} \tilde{h})^{n+1} - (\phi h)^n \right) \\
& - \Delta t \nabla \cdot \left[-D(\alpha) \left(\Delta \frac{(\tilde{\phi} \tilde{h}) \tilde{h}^2}{\mu(\tilde{\phi})} \nabla (\tilde{\phi} \tilde{h}) \right) + \tilde{\phi} \tilde{h} \left(\frac{\rho(\tilde{\phi})}{\mu(\tilde{\phi})} \tilde{h}^2 + (1 - \tilde{\phi}) V_s f(\tilde{\phi}) w(\tilde{h}) \right) \hat{\mathbf{x}} \right]^{n+1} \\
& - \Delta t \nabla \cdot \left[\phi h \left(\frac{h^2}{\mu(\phi)} \nabla \nabla^2 h - D(\alpha) \left(\frac{h^2}{\mu(\phi)} \nabla h - \frac{5}{8} \frac{h^3}{\mu(\phi)} \nabla (\rho(\phi)) \right) \right) - F_{diff} \right]^n.
\end{aligned} \tag{28}$$

206 Define

$$w = (\phi h)^{n+1} - (\tilde{\phi} \tilde{h})^{n+1}.$$

207 Then (28) can be written out as the three-step process (29)-(31):

$$\begin{aligned}
& \tilde{\mathcal{L}}_x v = - \left((\tilde{\phi} \tilde{h})^{n+1} - (\phi h)^n \right) \\
& - \Delta t \nabla \cdot \left[-D(\alpha) \left(\Delta \frac{(\tilde{\phi} \tilde{h}) \tilde{h}^2}{\mu(\tilde{\phi})} \nabla (\tilde{\phi} \tilde{h}) \right) + \tilde{\phi} \tilde{h} \left(\frac{\rho(\tilde{\phi})}{\mu(\tilde{\phi})} \tilde{h}^2 + (1 - \tilde{\phi}) V_s f(\tilde{\phi}) w(\tilde{h}) \right) \hat{\mathbf{x}} \right]^{n+1} \\
& - \Delta t \nabla \cdot \left[\phi h \left(\frac{h^2}{\mu(\phi)} \nabla \nabla^2 h - D(\alpha) \left(\frac{h^2}{\mu(\phi)} \nabla h - \frac{5}{8} \frac{h^3}{\mu(\phi)} \nabla (\rho(\phi)) \right) \right) - F_{diff} \right]^n,
\end{aligned} \tag{29}$$

$$\tilde{\mathcal{L}}_y w = v, \tag{30}$$

$$(\phi h)^{n+1} \approx (\tilde{\phi} \tilde{h})^{n+1} + w. \tag{31}$$

208 The spatial operators in the $\tilde{\mathcal{L}}_x$ and $\tilde{\mathcal{L}}_y$ terms are at most second-order, and spatial discretization
209 leads to a three-point stencil in each direction. Similar to with the first equation, a tridiagonal
210 solver or banded matrix solver can be used to solve along each row/column.

211 Solving the system, as a whole, at each timestep can be then done by solving the first equation
212 for h^{n+1} , solving the second equation for $(\phi h)^{n+1}$, then recovering the particle concentration as
213 $\phi^{n+1} = (\phi h)^{n+1}/h^{n+1}$.

214 4. Adaptive Timestepping

215 We use an adaptive timestepping scheme to advance the solution. The scheme utilizes the
216 solution at consecutive timesteps t^{n-1}, t^n, t^{n+1} . Based on a measure of error, it decides whether or
217 not to accept the new solution, and if it is reasonable to increase the size of the timestep. This
218 is a modification of the scheme used in Bertozzi et al. [4], in which it serves as an estimate of a
219 dimensionless local truncation error in time. Consider the solution of the film thickness, h , at times
220 t^{n-1}, t^n , and t^{n+1} . Calculate $e^{n+1} = (h^{n+1} - h^n)/h^n$ and $e^n = (h^n - h^{n-1})/h^n$. The modification
221 from the original method is to divide by the value h^n at each point rather than $h_{max}^n = \max_{i,j} \{h_{i,j}^n\}$,
222 since it produces a better-working adaptive scheme for this problem. Denote the timestep going
223 from time t^n to t^{n+1} as Δt and from t^{n-1} to t^n as Δt_{old} . Then define

$$Error = \left\| e^{n+1} - \frac{\Delta t}{\Delta t_{old}} e^n \right\|. \tag{32}$$

224 This provides a dimensionless estimate of the local truncation error in time, accumulated over the
 225 grid. The solution will be accepted if this error is less than some tolerance, denoted Tol_1 . If the
 226 error is less than a smaller tolerance, $Tol_2 < Tol_1$, for a fixed number of steps, the timestep is
 227 increased by a scale factor. If the error is larger than Tol_1 , the maximum number of iterations
 228 within a timestep is surpassed, or the solution becomes negative, the timestep is reduced by a
 229 factor of 2.

230 Since (32) only takes into account one of the two variables, this error can be computed for
 231 ϕh as well. These two errors can be combined into an overall measure of the error by taking the
 232 maximum of the two, or by some other reasonable combination.

233 5. Spatial Discretization

234 We use centered finite differences for all spatial discretizations. Using the notation, $h_{i+1/2,j} \approx$
 235 $(h_{i,j} + h_{i+1,j})/2$, the fourth-order term in (1) is

$$\begin{aligned}
 & \nabla \cdot \left(\frac{h^3}{\mu(\phi)} \nabla \nabla^2 h \right)_{i,j} \\
 & \approx \left(\frac{h_{i+1/2,j}^3}{\mu(\phi_{i+1/2,j})} h_{xxx,i+1/2,j} - \frac{h_{i-1/2,j}^3}{\mu(\phi_{i-1/2,j})} h_{xxx,i-1/2,j} \right) / \Delta x \\
 & + \left(\frac{h_{i+1/2,j}^3}{\mu(\phi_{i+1/2,j})} h_{yyx,i+1/2,j} - \frac{h_{i-1/2,j}^3}{\mu(\phi_{i-1/2,j})} h_{yyx,i-1/2,j} \right) / \Delta x \\
 & + \left(\frac{h_{i,j+1/2}^3}{\mu(\phi_{i,j+1/2})} h_{xxy,i,j+1/2} - \frac{h_{i,j-1/2}^3}{\mu(\phi_{i,j-1/2})} h_{xxy,i,j-1/2} \right) / \Delta y \\
 & + \left(\frac{h_{i,j+1/2}^3}{\mu(\phi_{i,j+1/2})} h_{yyy,i,j+1/2} - \frac{h_{i,j-1/2}^3}{\mu(\phi_{i,j-1/2})} h_{yyy,i,j-1/2} \right) / \Delta y.
 \end{aligned} \tag{33}$$

236 Here, the third derivatives are calculated at half-grid points by differencing consecutive standard
 237 second-order approximations. Two representative examples are

$$h_{xxx,i+1/2,j} \approx (h_{i+2,j} - 3h_{i+1,j} + 3h_{i,j} - h_{i-1,j}) / \Delta x^3, \tag{34}$$

$$\begin{aligned}
 h_{xxy,i,j+1/2} & \approx ((h_{i+1,j+1} - 2h_{i,j} + 1 + h_{i-1,j+1}) / \Delta x^2 \\
 & - (h_{i+1,j} - 2h_{i,j} + h_{i-1,j}) / \Delta x^2) / \Delta y.
 \end{aligned} \tag{35}$$

238 The two second-order terms are discretized as

$$\begin{aligned}
 & \nabla \cdot \left(\frac{h^3}{\mu(\phi)} \nabla (\rho(\phi) h) \right)_{i,j} \\
 & \approx \left(\frac{h_{i+1/2,j}^3}{\mu(\phi_{i+1/2,j})} (\rho(\phi_{i+1,j}) h_{i+1,j} - \rho(\phi_{i,j}) h_{i,j}) - \frac{h_{i-1/2,j}^3}{\mu(\phi_{i-1/2,j})} (\rho(\phi_{i,j}) h_{i,j} - \rho(\phi_{i-1,j}) h_{i-1,j}) \right) / \Delta x^2 \\
 & + \left(\frac{h_{i,j+1/2}^3}{\mu(\phi_{i,j+1/2})} (\rho(\phi_{i,j+1}) h_{i,j+1} - \rho(\phi_{i,j}) h_{i,j}) - \frac{h_{i,j-1/2}^3}{\mu(\phi_{i,j-1/2})} (\rho(\phi_{i,j}) h_{i,j} - \rho(\phi_{i,j-1}) h_{i,j-1}) \right) / \Delta y^2,
 \end{aligned} \tag{36}$$

$$\begin{aligned}
& \nabla \cdot \left(\frac{h^4}{\mu(\phi)} \nabla(\rho(\phi)) \right)_{i,j} \\
& \approx \left(\frac{h_{i+1/2,j}^4}{\mu(\phi_{i+1/2,j})} (\rho(\phi_{i+1,j}) - \rho(\phi_{i,j})) - \frac{h_{i-1/2,j}^4}{\mu(\phi_{i-1/2,j})} (\rho(\phi_{i,j}) - \rho(\phi_{i-1,j})) \right) / \Delta x^2 \\
& + \left(\frac{h_{i,j+1/2}^4}{\mu(\phi_{i,j+1/2})} (\rho(\phi_{i,j+1}) - \rho(\phi_{i,j})) - \frac{h_{i,j-1/2}^4}{\mu(\phi_{i,j-1/2})} (\rho(\phi_{i,j}) - \rho(\phi_{i,j-1})) \right) / \Delta y^2.
\end{aligned} \tag{37}$$

239 The advective term is discretized using a standard centered-differencing scheme.

240 The terms in (2) are discretized in the same manner since many of them are similar to those in
241 (1). The fourth- and second-order terms that come from v_{av} are discretized as in (33)-(37), with h
242 replaced by ϕh . Both advective terms are done via standard centered differencing.

243 The shear-induced diffusion term is discretized the same way as (36)-(37).

$$\begin{aligned}
& \nabla \cdot \left(\hat{D}(\phi) \frac{h^2 \rho(\phi)}{\mu(\phi)} \nabla \phi \right)_{i,j} \\
& \approx \left(\hat{D}(\phi_{i+1/2,j}) \frac{h_{i+1/2,j}^2 \rho(\phi_{i+1/2,j})}{\mu(\phi_{i+1/2,j})} (\phi_{i+1,j} - \phi_{i,j}) - \hat{D}(\phi_{i-1/2,j}) \frac{h_{i-1/2,j}^2 \rho(\phi_{i-1/2,j})}{\mu(\phi_{i-1/2,j})} (\phi_{i,j} - \phi_{i-1,j}) \right) / \Delta x^2 \\
& + \left(\hat{D}(\phi_{i,j+1/2}) \frac{h_{i,j+1/2}^2 \rho(\phi_{i,j+1/2})}{\mu(\phi_{i,j+1/2})} (\phi_{i,j+1} - \phi_{i,j}) - \hat{D}(\phi_{i,j-1/2}) \frac{h_{i,j-1/2}^2 \rho(\phi_{i,j-1/2})}{\mu(\phi_{i,j-1/2})} (\phi_{i,j} - \phi_{i,j-1}) \right) / \Delta y^2.
\end{aligned}$$

244 Centered differencing is not used for the moving reference frame, if one is employed. Instead, a
245 second-order upwind differencing scheme is used, which will be discussed in the next section.

246 6. Reference Frame

247 The area of interest in the simulations is near the front of the flow, where effects like the capillary
248 and particle-rich ridge occur. With a fixed reference frame, the spatial domain would need to be
249 taken as the entire area over which the flow would evolve, leading to large portions of the domain
250 where no change is occurring. This issue can be easily addressed by using a moving reference frame.

251 To implement a moving reference frame, we add an extra term to each equation, $-sh_x$ on the
252 left-hand side of (1) and $-s(\phi h)_x$ on (2). Here, s is the speed at which the moving reference frame
253 travels. Zhou et al. [35] approximate the front speed by removing all terms from the equations
254 which are higher than first order, leaving only the advective terms. They observe that these terms
255 capture the large scale dynamics, including the speed of the shocks, and the ridges that develop in
256 h and ϕ . This leaves a 2×2 system of scalar conservation laws of the form

$$\begin{aligned}
& h_t + [F(h, \phi h)]_x = 0, (\phi h)_t + [G(h, \phi h)]_x = 0, \\
& F(h, \phi h) = \frac{\rho(\phi)}{\mu(\phi)} h^3, G(h, \phi h) = \frac{\rho(\phi)}{\mu(\phi)} (\phi h) h^2 + (\phi h)(1 - \phi) V_s f(\phi) w(h).
\end{aligned} \tag{38}$$

257 The initial conditions for (38) are

$$h(x, 0) = \begin{cases} h_l, & x \leq 0, \\ h_r, & x > 0, \end{cases} \tag{39}$$

$$(\phi h)(x, 0) = \begin{cases} \phi_0 h_l, & x \leq 0, \\ \phi_0 h_r, & x > 0. \end{cases} \quad (40)$$

where h_l and h_r in (39) and (40) are the initial film thickness and the height of the precursor b , respectively, and ϕ_0 in (40) is the initial particle concentration of the fluid. These initial conditions specify a Riemann problem [18]. From the initial shock in both equations, an intermediate state emerges, $(h_i, (\phi h)_i)$. The weak form of this system produces two Rankine-Hugoniot jump conditions, which define the shock speeds, ahead and behind the intermediate states. For s_1 , the speed of the shock behind the intermediate state, and s_2 , the speed ahead, these conditions are given by

$$\begin{aligned} s_1 &= \frac{F(h_i, (\phi h)_i) - F(h_l, (\phi h)_l)}{h_i - h_l} = \frac{G(h_i, (\phi h)_i) - G(h_l, (\phi h)_l)}{(\phi h)_i - (\phi h)_l}, \\ s_2 &= \frac{F(h_r, (\phi h)_r) - F(h_i, (\phi h)_i)}{h_r - h_i} = \frac{G(h_r, (\phi h)_r) - G(h_i, (\phi h)_i)}{(\phi h)_r - (\phi h)_i}. \end{aligned} \quad (41)$$

This nonlinear system (41) of four equations and four unknowns, $h_i, \phi_i h_i, s_1$, and s_2 , can be solved via Newton's method. For the simulations shown in Section 7, our reference frame speed is an average of the two speeds, $s = (s_1 + s_2)/2$.

The discretization of the terms for the moving reference frame is done explicitly using forward Euler combined with second-order upwind-differencing,

$$-sh_x \approx -s \frac{-h_{i+2,j} + 4h_{i+1,j} - 3h_{i,j}}{2\Delta x}.$$

This was chosen for simplicity and that it visually produced better results than the other discretizations that were tried. The effects of this choice appear to be some minor dissipation and dispersion in some cases behind the particle-rich ridge in the concentration.

The moving reference frame can be used for both the 1-D and 2-D cases (see Figures 4 and 5). To demonstrate this, simulations were run under the same conditions as those in Section 7. The theory-based solution for the problem without higher-order terms (38)-(40) aligns well with the 1-D numerical solution for the full problem. The 2-D solution for the full problem with a perturbation to the initial film thickness leads to a finger that moves faster than the 1-D case and the troughs, to the sides of the finger, move slower.

This can be viewed more succinctly in Figures 6 and 9, where the contours of the 1-D and perturbed 2-D cases are overlaid. The position of the finger runs ahead of the 1-D front while the troughs lag behind. Figure 7 shows the average front position of the flow for both cases. The averaging of the front position was first done by Huppert [13] for experiments involving clear fluids. Both simulations start with the same volume and, after an initial transient, the average front positions of the film for the 1-D and perturbed 2-D cases (measured at $h = 0.5$) remains constant and close to each other. Figure 8 shows the position of the finger and the trough in the 2-D case over time.

7. Simulations

A rectangular domain is used with the x -direction oriented down the inclined plane and the y -direction across the inclined plane. In all cases, the particle concentration is initially taken to be

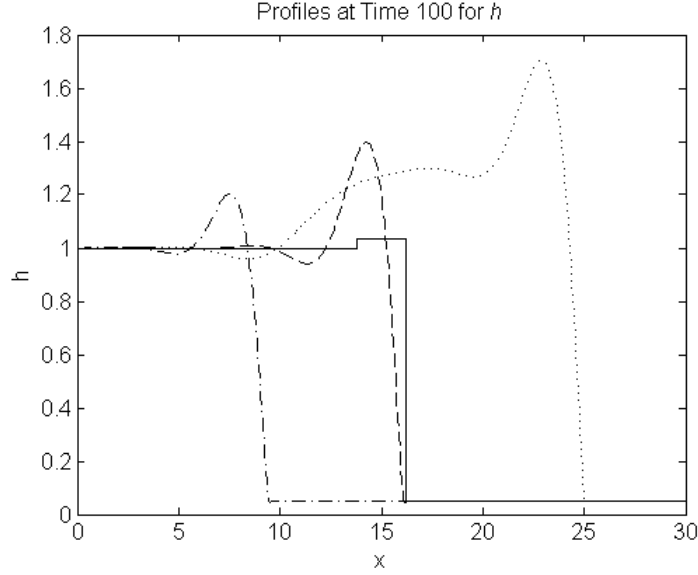


Figure 4: Comparison of theory and simulations at time $t = 100$ for the film thickness, h : theory without higher-order terms (solid line), 1-D solution to the full problem (dashed line), perturbed 2-D finger (dotted line), and perturbed 2-D trough (dot-dashed line).

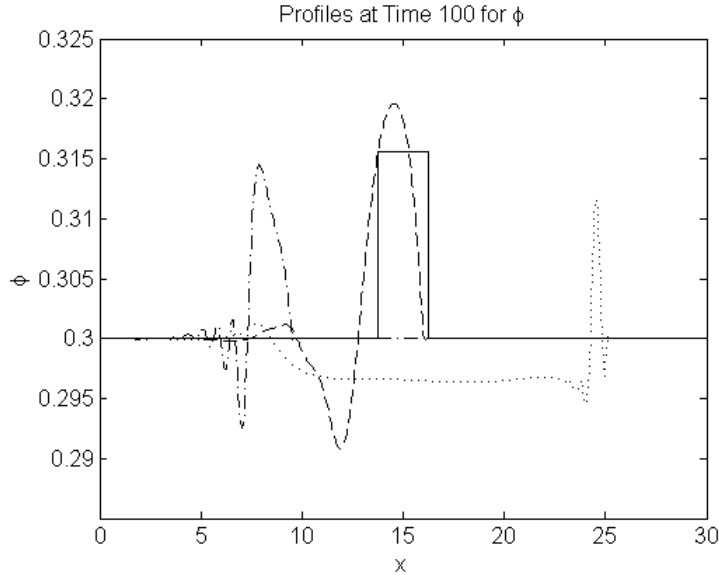


Figure 5: Comparison of theory and simulations at time $t = 100$ for the particle concentration, ϕ . The labels are the same as in Figure 4.

289 $\phi(x, y, 0) = \phi_0$, where $0 \leq \phi_0 \leq \phi_{max}$. This corresponds to having a well-mixed initial fluid. The
 290 film thickness far behind the contact line is set at $h(x, y, 0) = 1$ and ahead of the flow, a precursor
 291 of height $h(x, y, 0) = b$ is assumed. At the contact line, a perturbation to a linear front can be
 292 applied to induce behavior such as a fingering instability. The parameters in the model are taken
 293 to be: $a = 0.1, \Delta = 1.7, Ca = 10^{-3}, \alpha = \pi/4$. The constant ϕ_{max} is taken to be 0.67, in line with
 294 the simulations in Cook et al. [7]. The initial timestep is set to $\Delta t = 10^{-6}$.

295 For the model, two sources contribute to the height of the film thickness and particle concentra-

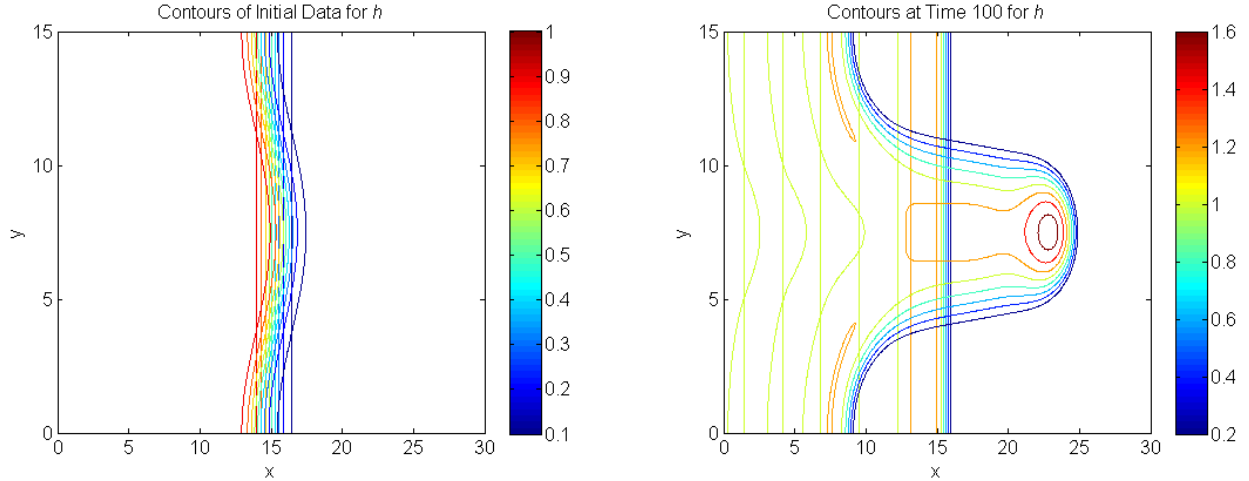


Figure 6: A contour plot of the simulation at times $t = 0$ and $t = 100$ for the film thickness, h , in the 1-D and perturbed 2-D cases. The perturbation in 2-D leads to a fingering instability not seen in the 1-D case.

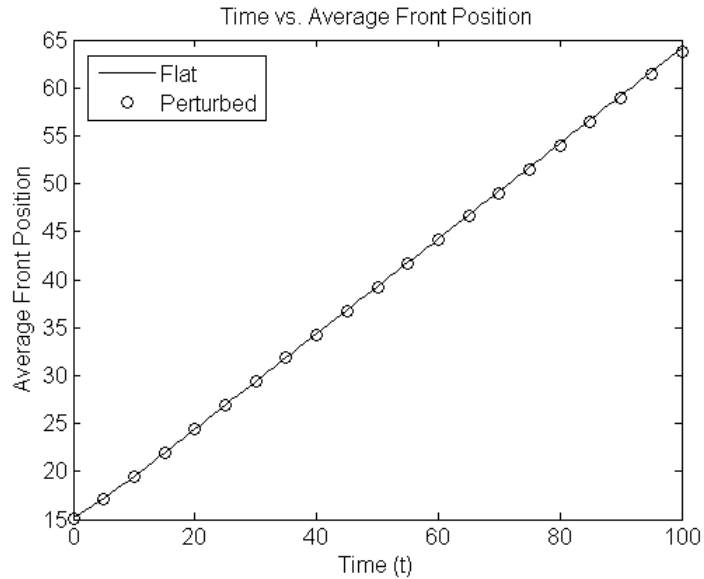


Figure 7: The average front position of the film thickness, h , of the 1-D and perturbed 2-D cases up to time $t = 100$. After an initial transient, the speeds stay close to constant and to each other.

296 tion near the front of the flow. The first is the higher-order terms, such as surface tension, which
 297 produce smooth ridges in both h and ϕ . Second, even without these terms, an intermediate state
 298 at the front emerges for both variables, higher than either of their respective left or right states.
 299 These heights are dependent on the precursor b .

300 The height of the precursor in the following simulations is chosen out of convenience, to keep the
 301 size of Δx close to the precursor height. In general, choosing a different precursor has a small effect
 302 on the speed of the flow, but a large effect on both the film thickness and particle concentration.
 303 To illustrate this, Table 1 shows the height of the intermediate states for both h and ϕ as well
 304 as the speeds of the trailing and leading shocks obtained from the theory-based solution to the
 305 system of scalar conservation laws (38)-(40) (see Section 6 for a more in-depth discussion). The

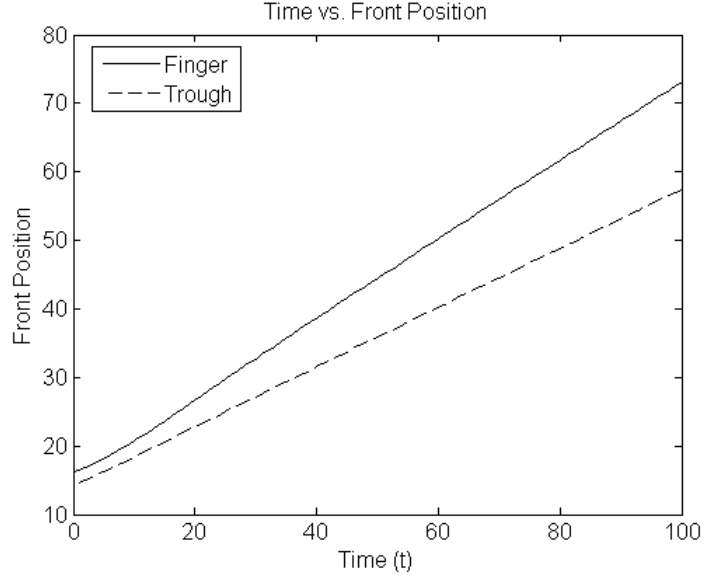


Figure 8: The front position of the film thickness, h , of the perturbed 2-D case up to time $t = 100$ along the finger and trough.

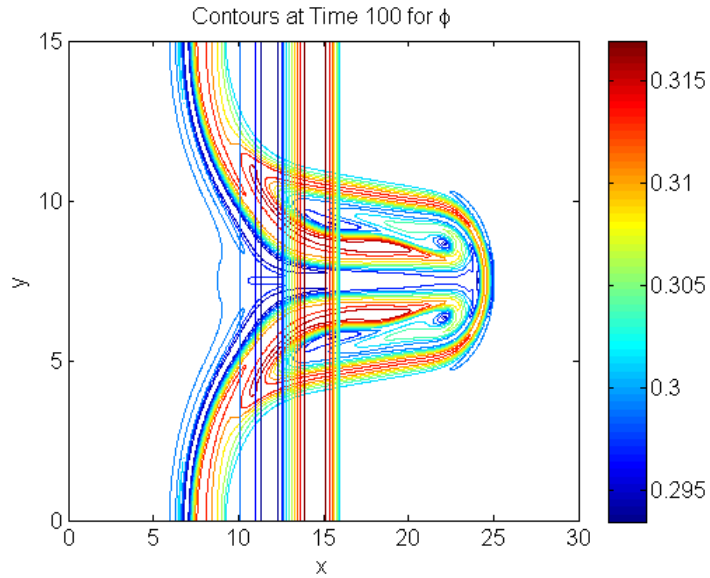


Figure 9: A contour plot of the simulation data at time $t = 100$ for the particle concentration, ϕ , in the 1-D and perturbed 2-D cases. The perturbation leads to a particle-rich ridge that outlines and begins to fill in the finger.

intermediate film thickness h_i and particle concentration ϕ_i increase as the height of the precursor b decreases. For the shock speeds, a smaller precursor leads to the trailing shock speed s_1 staying relatively the same, but the leading shock speed s_2 slows down and approaches s_1 . These results agree with the previous ones related to solving the system of scalar conservation laws [6, 35]. For this model, the smallest precursor for which a solution exists is $b \approx 9 \times 10^{-4}$ [6]. A precursor close to this case, $b = 0.001$, produces shocks speeds which are close together and an intermediate particle concentration near the maximum packing fraction. An alternative settling function that permits solutions with smaller precursors, $f_B(\phi) = (1 - \phi/\phi_{max})^5$, is examined in Cook et al. [6].

b	h_i	ϕ_i	s_1	s_2
0.1	1.01653	0.307566	0.459323	0.510221
0.05	1.03478	0.315538	0.459314	0.483782
0.025	1.07107	0.330331	0.459301	0.471418
0.0125	1.1427	0.356006	0.459289	0.465441
0.00625	1.28276	0.396078	0.459294	0.462488
\vdots	\vdots	\vdots	\vdots	\vdots
0.001	9.14247	0.635545	0.459788	0.459916

Table 1: The intermediate states and shock speed solutions from equation (41) based on the precursor thickness b . As the precursor decreases, both h_i and ϕ_i increase and the shock speeds converge.

314 The boundary conditions for h are Dirichlet in front and behind the flow and Neumann on the
315 sides. The same is done for ϕ . In addition, all third derivatives in h , normal to the boundary, are
316 set to 0. More specifically, for a rectangular domain with length X_0 and width Y_0 , the boundary
317 conditions are

$$\begin{aligned}
h(0, y) &= 1, \quad h_{xxx}(0, y) = 0, \quad h(X_0, y) = b, \quad h_y(X_0, y) = 0, \\
h_y(x, 0) &= 0, \quad h_{yyy}(x, 0) = 0, \quad h_y(x, Y_0) = 0, \quad h_{yyy}(x, Y_0) = 0, \\
\phi(0, y) &= \phi_0, \quad \phi(X_0, y) = \phi_0, \quad \phi_y(x, 0) = 0, \quad \phi_y(x, Y_0) = 0.
\end{aligned}$$

318 The simulations are all run using moving reference frames, with the speed of the frame determined
319 as in Section 6.

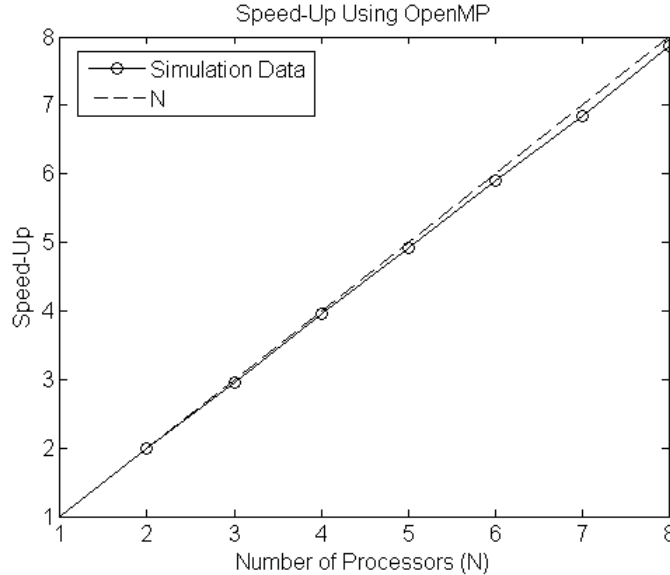


Figure 10: The speed-up gained by going from 1 to N processors using OpenMP. The line $y = N$ is shown as a point of reference.

320 The code is written in parallel using the C++ OpenMP package. This choice of parallelization
321 was made since the majority of calculations are done via *for* loops and OpenMP works well with

322 loop-heavy code. This includes the calculation of all finite differences and the solves along rows and
 323 columns associated with the ADI part of the scheme. This is especially useful since rows/columns
 324 can be solved independently of each other for each equation. In addition, writing special solvers
 325 for linear systems of equations across multiple processors [24, 27] is avoided by this approach. The
 326 speed-up attained using N processors is calculated by dividing the runtime for one processor by
 327 the runtime for N processors ($Speed - Up = Time(1 Processor)/Time(N Processors)$).

328 Based on Figure 10, the scaling seems close to linear up to 4 processors, with a small drop-off
 329 in performance as the number increases. This almost-linear behavior is a result of all of the code,
 330 outside of a few minor calculations and the recording of the data, being amenable to parallelization.

331 To test some preferences that need be chosen a priori in the simulation, we conducted short-
 332 time tests to gauge the effectiveness of each approach. The ones considered here are (a) whether to
 333 time-lag or extrapolate the approximate terms and (b) whether or not to perform iterations past
 334 a single solve to improve the approximate terms, and therefore the solution at each timestep (see
 335 Table 2).

(a) Approximate Terms	Time-Lag	Extrapolate
(b) Iterations	One Iteration	Multiple Iterations

Table 2: The two choices to be made when implementing the numerical scheme. One must choose whether to (a) time-lag or extrapolate the approximate terms and (b) whether to perform additional iterations past the initial solve.

336 Consider an initial condition of $\phi_0 = 0.3$ and a front perturbed from Riemann initial data,
 337 $h(x, y, 0) = 1$ behind far behind the front, $h(x, y, 0) = 0.05$ far ahead of the front. At the jump
 338 from fluid to precursor, the shape of the front given as $x_{front} = X_0/2 - \cos(2\pi y/Y_0)$. This initial
 339 data is then smoothed using hyperbolic tangent and matched to the boundary condition (see Figure
 340 13). This has the effect that the initial timestep can be taken more leniently.

341 We ran this initial simulation for each of the four combinations in Table 2 to time $t = 1$ and
 342 the maximum timestep allowed, average number of iterations per timestep, and the total runtime,
 343 in seconds, are listed in the table below (Table 3). This choice was made as the timestep changes
 344 dramatically over this time interval and can provide insight as to what methods seem practical
 345 for long-time runs. Since adaptive timestepping is utilized here, the tolerances are tuned so as to
 346 ensure that the simulation stays stable, not only to time $t = 1$ but for some time afterwards as well
 347 (it is taken up to $t = 100$ in this case, which is the length of the long-run simulations).

	Δt_{max}	Avg. Iter.	Runtime
Time-Lagged and One Iteration	0.000568341	1.0	518.2
Time-Lagged and Iterations	0.00183296	2.20997	601.468
Extrapolation and One Iteration	4.07743×10^{-5}	1.0	19596.1
Extrapolation and Iterations	0.00486338	1.29668	376.603

Table 3: Results for time $t = 1$ based on various choices for implementation.

348 Using *Iterations* does well for both choices of approximate terms in that the total runtimes are
 349 low, the maximum timesteps are large, and the number of iterations stays close to 1. Between

350 these two, *Extrapolation and Iterations* does best, with nearly one fewer iteration required per
 351 timestep, on average, and a runtime that is 37% shorter. Performing *One Iteration*, the runtime for
 352 *Time-Lagged* is in between the two cases with *Iterations*, but for *Extrapolation*, it performs poorly,
 353 producing a runtime that is 33 to 52 times worse than the other three options. This is due to the
 354 small maximum timestep that is associated with this approach, which is 14 to 119 times smaller
 355 than the other three. At this point, it makes sense to discard the *Extrapolation and One Iteration*
 356 approach due to its excessive runtime and explore the remaining ones.

357 Under the same conditions, we ran a longer simulation, this time to $t = 100$. Using the best
 358 remaining options, we can glean some idea as to which one(s) will work best for a longer simulation.

	Δt_{max}	Avg. Iter.	Runtime
Time-Lagged and One Iteration	0.00107169	1.0	17811.3
Time-Lagged and Iterations	0.00329173	2.95498	13153.8
Extrapolation and Iterations	0.0106161	2.01204	3364.93

Table 4: Results for time $t = 100$.

359 Comparing Tables 3 and 4, the maximum timestep for each approach has increased. Using
 360 *Iterations*, the average number has gone up in for both *Time-Lagged* and *Extrapolations*. However,
 361 the average number of iterations per timestep for *Extrapolation* is approximately one iterations fewer
 362 than for *Time-Lagged*. Also the runtime takes about 2.9 times longer for *Time-Lagged* compared to
 363 *Extrapolation*. One can see the benefit of performing iterations instead of using a smaller timestep in
 364 comparing the results for *Time-Lagged and One Iteration* and *Time-Lagged and Iterations*. *Time-*
 365 *Lagged and One Iteration* advances the solution approximately the same time forward with three
 366 timesteps as *Time-Lagged and Iterations* does with one timestep and three iterations. However,
 367 doing two extra timesteps costs more than two extra iterations, as seen in their respective runtimes.
 368 This is because the explicit terms do not need to be re-calculated for each iteration while they do
 369 for each timestep. Therefore, the only two options which make sense to use are the ones involving
 370 *Iterations*. Of these, *Extrapolation* is the clear favorite.

371 In Figure 11, we see that by time $t = 8$, all three approaches have settled into a respective
 372 timestep. The timestep for *Extrapolations and Iterations* does best, followed by *Time-Lagged and*
 373 *Iterations* and *Time-Lagged and One Iteration*. The timestep for *Extrapolation and Iterations* is
 374 3.2 times better than *Time-Lagged and Iterations* and 9.9 times better than *Time-Lagged and One*
 375 *Iteration*. The benefit of the larger timestep for both approaches with *Iterations* is partially offset
 376 by the need for extra calculations related to the iterations.

377 Figure 12 shows the number of iterations required throughout the simulation. For *Extrapolation*
 378 *and Iterations*, the increase in iterations approximately between times $t = 20$ and $t = 30$ corresponds
 379 to the finger forming and stretching out ahead of the flow in the film thickness and the particle-rich
 380 ridge growing higher and outlining the finger. While the number of iterations jumps once to 3 and
 381 then back down to 2 for *Extrapolation and Iterations*, it remains constant at 3 for *Time-Lagged*
 382 *and Iterations*. The cost of storing extra data and performing a small computation to find the
 383 extrapolated approximations seems a small price to pay to save one iteration per timestep, which
 384 includes recalculating values involving the approximate terms and performing the ADI solves.

385 Using the simulation data up to $t = 100$, we can examine the effects of the initial perturbation
 386 graphically. For the film thickness, a small capillary ridge forms in the center of the perturbation

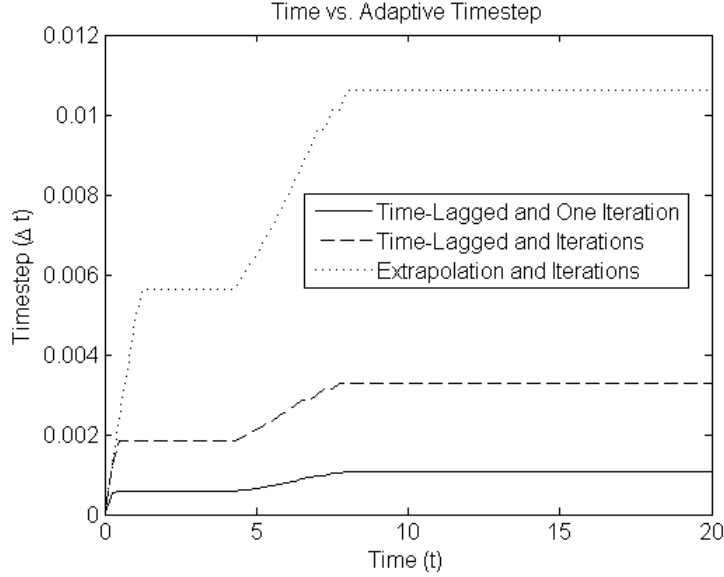


Figure 11: The adaptive timestep up to time $t = 20$. The timestep, Δt , is recorded in intervals of 0.25 for the three cases. *Extrapolation and Iterations* has a significantly larger timestep than either *Time-Lagged and One Iteration* or *Time-Lagged and Iterations*.

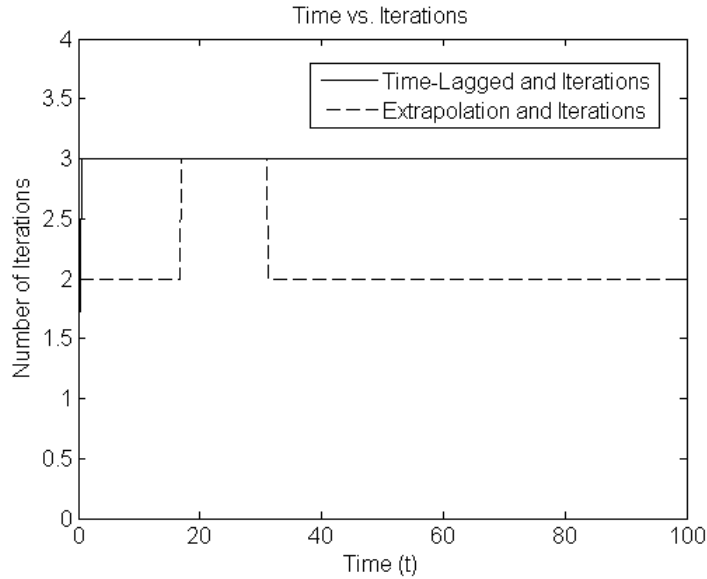


Figure 12: The number of iterations up to time $t = 100$. The iterations are recorded in intervals of 0.25 for the two cases. Using *Extrapolation and Iterations* does better than *Time-Lagged and Iterations* in terms of fewest number of iterations.

387 (Figure 14) and begins to stretch out ahead of the bulk flow (Figures 15 and 16). This is the well-
 388 known fingering instability present in thin-film flows. For the particle concentration, a particle-
 389 rich ridge initially forms at the contact line (Figure 17) and, as the fingering instability evolves,
 390 outlines the shape of the finger (Figures 18 and 19). Directly behind the ridge, a pocket of lower
 391 concentration forms. The interior of the finger is slowly encroached upon by the particles that have
 392 accumulated near the back and sides of the finger. This can be seen in Figure 19 as an interior

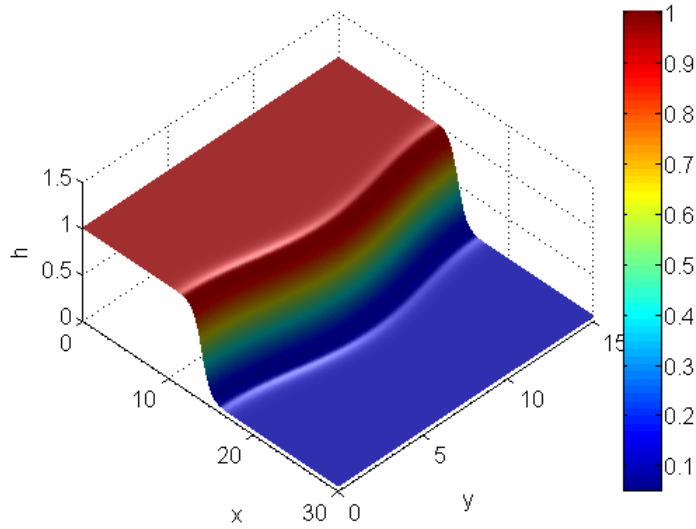


Figure 13: The initial film thickness. It is perturbed by a cosine wave along y and smoothed along x by hyperbolic tangent.

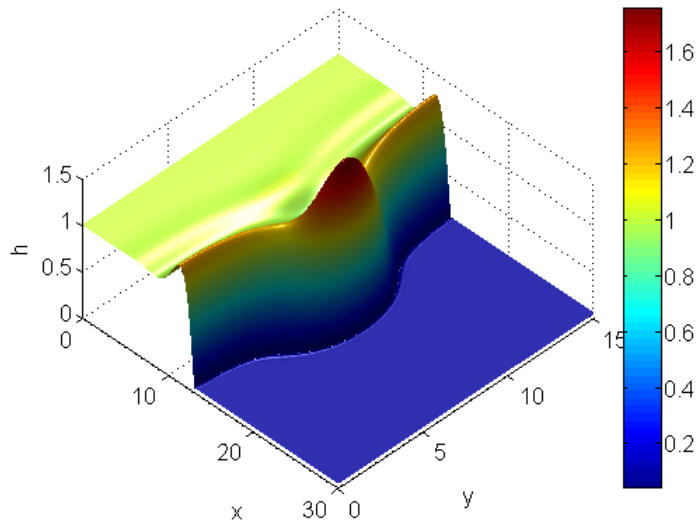


Figure 14: Film thickness at time $t = 25$. A small capillary ridge forms in the center of the flow.

393 layer along the inside of the particle-rich ridge. It is possible that this phenomenon is not physical,
 394 and may be a result of the current model not containing all of the necessary physics.

395 8. Comparison to Experiments

396 Experiments for particle-laden thin film flows have been compared in one dimension to the
 397 solution, both analytically and numerically, for clear-fluid flows. The average front position for
 398 clear fluids is given by a power law, where the location of the front scales like $t^{1/3}$ [13]. Ward et al.
 399 [32] compare the average front position of the flow to this scaling. Grunewald et al. [11] compare

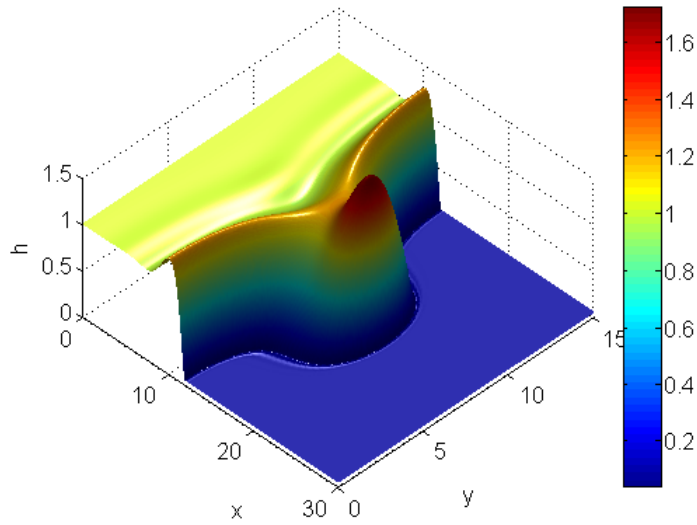


Figure 15: Film thickness at time $t = 50$. A fingering instability begins to develop from the ridge.

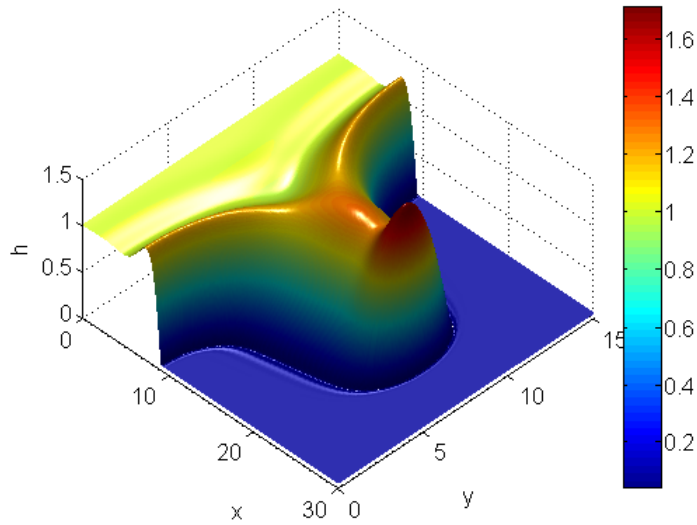


Figure 16: Film thickness at time $t = 100$. The finger stretches out ahead of the bulk flow.

400 the average front position to a re-derived 1-D model, based on results from Huppert [13] with a
 401 precursor, and to numerical solutions of the 1-D problem. Both find some agreement between the
 402 experiments and the scaling for the 1-D front position for clear fluids. We seek to compare the
 403 numerical solution in two dimensions to images of experiments, taking into account that variations
 404 occur across the front of the flow.

405 We use 1000 cSt Polydimethylsiloxane (PDMS), a silicone oil, for the liquid component of the
 406 fluid. For the particles, glass beads with diameters in the range of $250 - 425 \mu\text{m}$ are used. The
 407 two components are then well-mixed and released down an inclined plane from a reservoir. This
 408 corresponds to a constant-volume experiment, whereas our simulations are constant-flux.

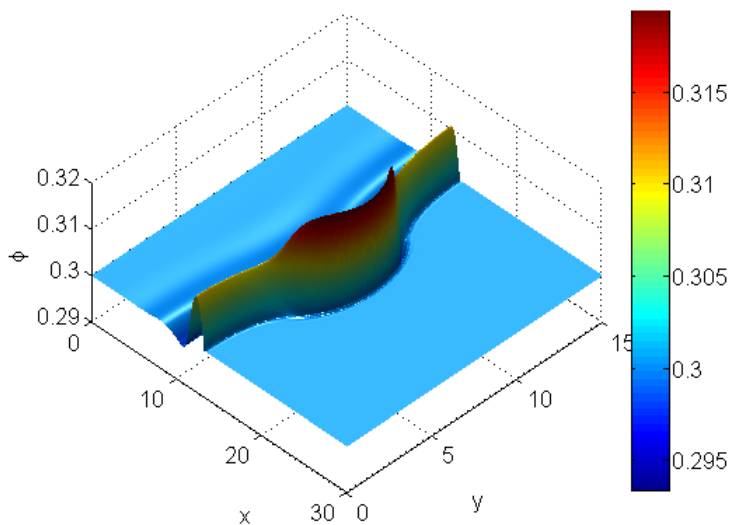


Figure 17: Particle concentration at time $t = 25$. A small particle-rich ridge forms at the front of the flow with a slightly higher concentration in the center.

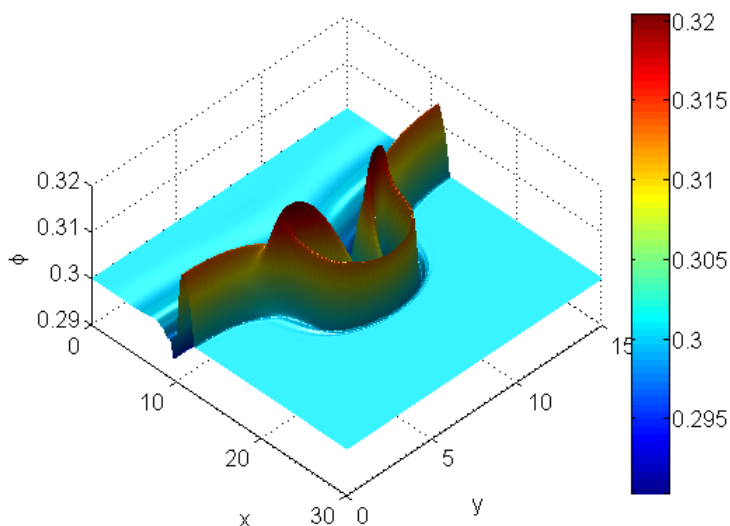


Figure 18: Particle concentration at time $t = 50$. The particle-rich ridge increases in concentration and has a higher concentration in and around the fingering instability.

409 The experiment, which we will compare to simulation, is a fluid of approximately 90 cm^3
 410 containing a volume which is 35.9% particles. The plane is inclined at a 32-degree angle. The fluid is
 411 allowed to flow down the plane, which is 14 cm across and 90 cm down. In the experiments, the flow
 412 starts out close to uniform across the front, away from the edges, and over time develops instabilities,
 413 in the form of fingers stretching out ahead of the bulk flow. Since, for simulations, starting with a
 414 uniform front along the y -direction leads to a uniform solution, we start the simulation some time
 415 after the start-time to add a perturbation to the initial data, which induces the type of behavior
 416 seen in the latter stages of the experiments.

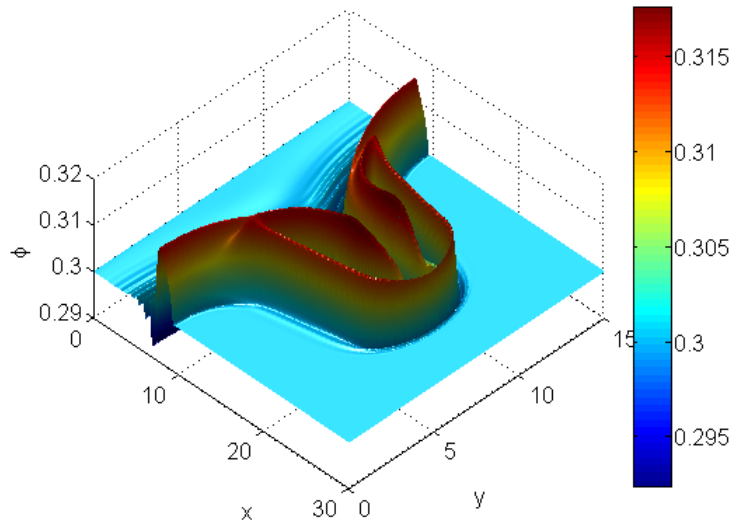


Figure 19: Particle concentration at time $t = 100$. The particle-rich ridge is composed of two parts: one outlining the finger and one entering the finger from the troughs.

417 In order to avoid simulating the problem over the entire domain, we truncate the solution near
 418 the front and treat the problem locally as being constant-flux. We are interested in the dynamics
 419 of finger formation during which time the film thickness only changes by at most 20% , so a local
 420 approximation by constant-flux is reasonable.

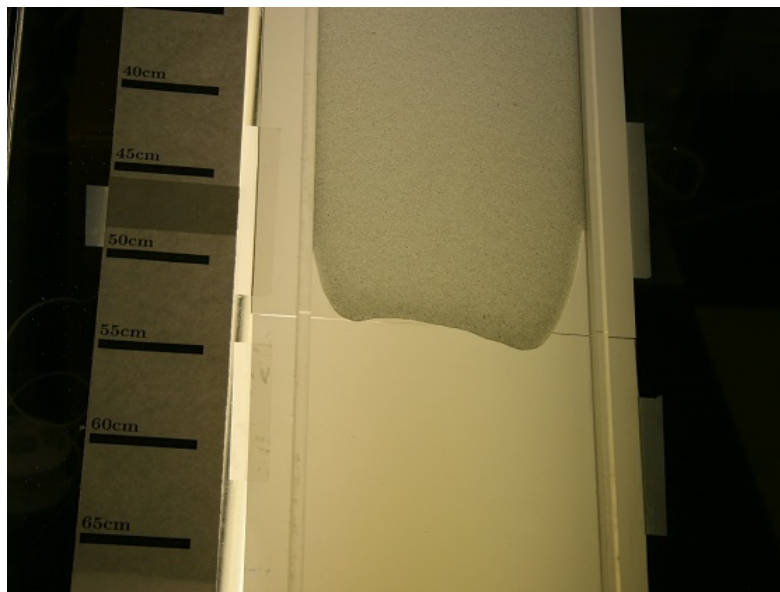


Figure 20: The initial condition of the experiment, used for comparing with the simulation. At this point, the front of the flow has begun to develop perturbations, which will lead to fingering instabilities.

421 We use two images, taken three minutes apart, to compare with the simulation. The first image
 422 is taken when the front of the flow has reached approximately 53 *cm* down the plane. The shape
 423 of the front is parabolic-like with two large perturbations at either end of the front. In between,

424 smaller perturbations exist which lead to fingering instabilities. The two outer perturbations lead
 425 to longer and thicker fingers than the smaller inner perturbations. We take a front similar to this
 426 in our simulation.

427 The scales for a constant-flux problem can be taken from Cook et al. [6], which are the same
 428 as for the clear-fluid case. The height scale is taken to be $h_0 = 1 \text{ mm}$. The length scale is
 429 $x_0 = (l^2 h_0)^{1/3}$, where the capillary length, l , is $l = \sqrt{\gamma/\rho_l g_{\parallel}}$. The constants are γ , the coefficient
 430 of surface tension; ρ_l , the liquid density; and g_{\parallel} , the component of gravity parallel to the inclined
 431 plane. The time scale is $t_0 = (3\mu_l/\gamma)x_0 l^2/h_0^2$, where μ_l is the dynamic liquid viscosity. The capillary
 432 number is given by $Ca = \mu_l x_0/\gamma t_0 = h_0^2/3l^2$.

433 The scales, given these parameters, are $h_0 = 0.001 \text{ m}$, $x_0 = 0.00161396 \text{ m}$, $l = 0.00205041 \text{ m}$,
 434 $t_0 = 0.93235 \text{ s}$, and $Ca = 0.0792863$. Using this, we can construct an initial condition which
 435 resembles the experiment and will produce similar results. This is done by measuring the features
 436 of the initial image and creating a similar condition. While the flow in the experiment is asymmetric,
 437 we take a symmetric initial condition in the simulation which has features that are approximately,
 438 in both location and size, the same as in the experiment. The track is taken to be 86.75 units
 439 wide (rounded up to the nearest 0.05 increment), which corresponds to the 14 *cm* wide track. The
 440 precursor in the simulation is set to $b = 0.05$, as in the previous simulations.

441 A moving reference frame is used since this is taken to be a constant-flux problem locally. The
 442 speed of the moving reference frame is approximately $s = 0.343198$, calculated as in Section 6.
 443 Running a simulation over the course of three minutes leads to a distance traveled for the frame of
 444 approximately 10.69 *cm*, where the actual displacement, based on experiments, is around 12 *cm*,
 445 so using the constant-flux assumption seems to produce a decent approximation of the distance the
 446 fluid will flow.

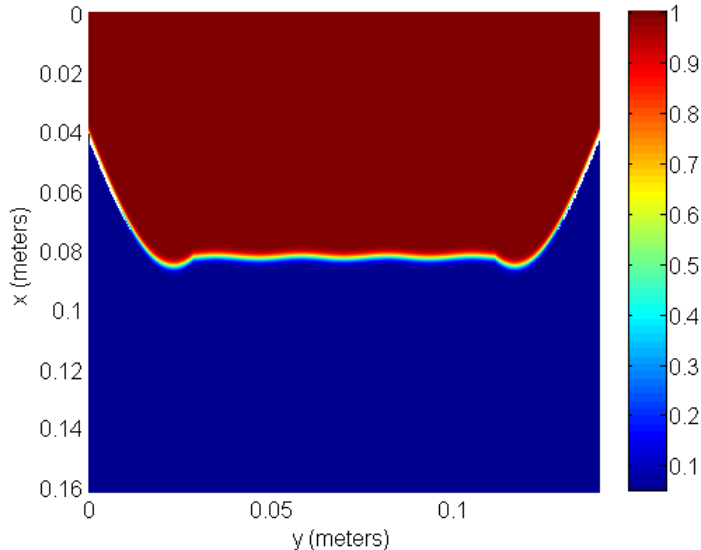


Figure 21: The initial condition for the film thickness, h , used in the simulation. This is an artificially-created starting condition to be representative of the state shown for the experiment. The height is in *mm*.

447 The initial data is generated using a sine wave to form the two large perturbations and the
 448 space away from the edges. The three fingers that develop between these two perturbations are
 449 simulated with a cosine wave of small amplitude, 0.25 in dimensionless units. The simulation is

450 run to $t = 193.06$, the equivalent of three minutes of real-time.

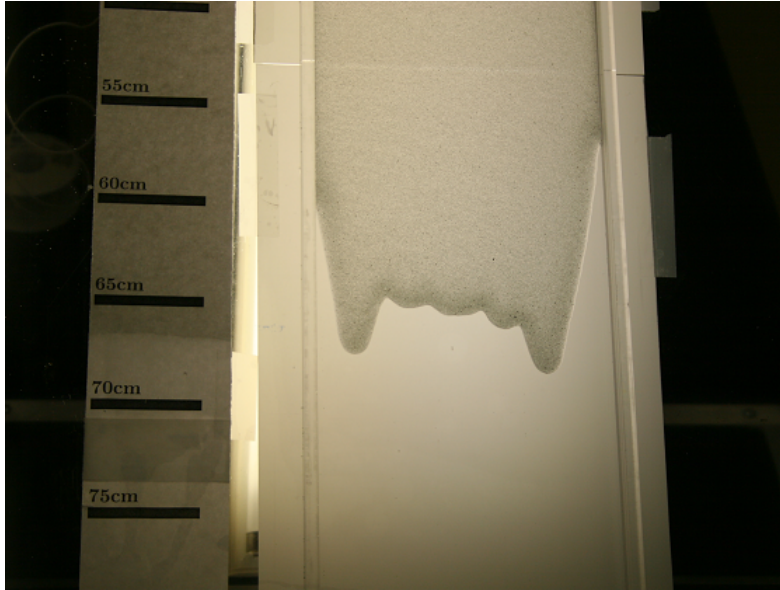


Figure 22: The evolution of the experiment after three minutes. The fingering instability starts to form at the front.

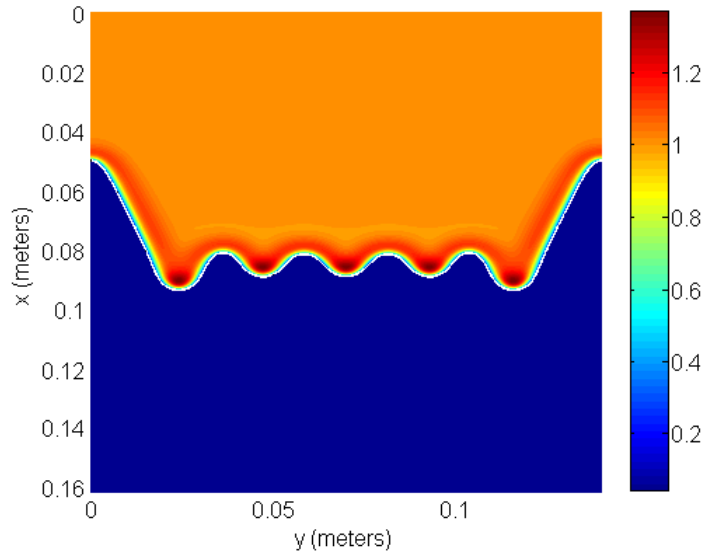


Figure 23: The evolution of the film thickness, h , in the simulation after three minutes. Both the experiment and simulation exhibit a fingering instability, but the instability in the simulation is less pronounced. The height is in mm .

451 Over the course of the three minutes, the exterior of the outer fingers in the experiments go
452 from 4 cm and 6.5 cm on the left and right, respectively, to 7.5 cm and 12 cm . The interior of these
453 fingers go from less than 1 cm on each side to about 3 cm . The interior fingers are not discernable
454 in the initial image. The flow as a whole, measured from where the fluid touches the walls, has
455 moved about 11 cm down the plane. The interior fingers in the experiment, extend about 0.5 cm
456 ahead of the flow.

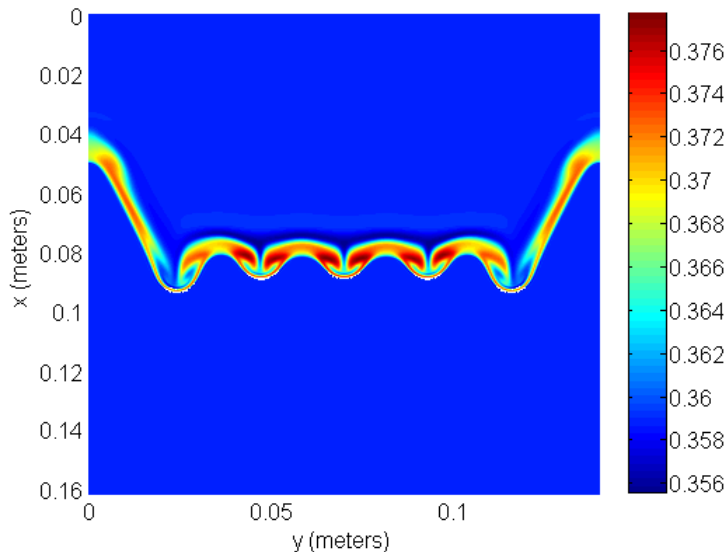


Figure 24: Particle concentration, ϕ , for the film thickness in Figure 23.

457 In the simulation, the moving reference frame accounts for 10.69 *cm* of movement, so the position
 458 where the fluid touches the edges has moved approximately 7.5 *cm*. The evolution of the fingers
 459 in the simulation is slightly less pronounced than in the experiments. This is likely due to the
 460 simulation initially undergoing a transient state where the fluid travels slower than at later times,
 461 while the transient in the experiment has occurred prior to this three-minute interval. The exterior
 462 of the outer fingers is approximately 4.2 *cm* and interior 1.2 *cm*. The interior fingers extend ahead
 463 of the flow about 0.8 *cm*. The tip of the longest finger in the experiments has moved 15 *cm* while in
 464 the simulations, it has advanced approximately 11.4 *cm*. The tips of the fingers, in the *z*-direction,
 465 reach up to 1.37 *mm*.

466 The particle concentration cannot be determined accurately at the particle-rich ridge in the
 467 experiment, but the increased opacity at the leading edge of the flow indicates an increase in the
 468 concentration, relative to the ambient concentration. This change in shade is approximately 1 *cm*
 469 long in the direction of the flow. In the simulations, the thickness of the ridge ranges from 0.6 to
 470 1.1 *cm*, which is consistent with the experiments.

471 9. Discussion

472 Schemes originally derived for numerically solving high-order parabolic problems have recently
 473 been extended to high-order systems, such as the case of surfactants and particle-laden thin films.
 474 Handling the higher-order terms in a practical way is necessary for fast and efficient computation.
 475 The scheme we have presented here for particle-laden thin film flow provides an easy-to-program
 476 and effective way to solve this high-order coupled system. This scheme can provide a blueprint for
 477 approaches to solving similar problems.

478 The numerical scheme developed for particle-laden thin film flow has several nice attributes.
 479 The timestep required for this scheme is in the range of $O(\Delta x^2)$, which is much better than the
 480 $O(\Delta x^4)$ for a fully explicit scheme. The structure of the scheme allows for the possibility of solving
 481 each equation with its own unique timestep for better efficiency, as the particle concentration is

482 typically the equation that fails the timestep restriction criteria. The linear algebra problem that
483 results from an implicit time discretization along with the nonlinearity is reduced to a series of tri-
484 and pentadiagonal solves, which can be done in parallel along the rows/columns of the grid.

485 The parallelization of the code is straightforward using OpenMP. The loops for computing the
486 explicit and approximate terms as well as the solves along rows and columns can be done in parallel,
487 leading to a code that scales close to linearly for up to 8 processors, getting close to 8 times speed-
488 up. Adding OpenMP implementation to C++ code on any multicore machine is easy to implement,
489 as it only requires adding a few lines of code to existing *for* loops and needs no managing of the
490 movement of data on the programmer's part. Since the code is predominantly such loops, it is easy
491 to parallelize and is highly effective in getting better runtimes.

492 Implementing *Iterations* within each timestep, which is first presented in Witelski and Bowen
493 [34], but not used in Warner et al. [33], seems to work best for this problem, in terms of allowing
494 for a larger timestep and producing an accurate solution. Among the choices for the approximate
495 terms when performing *Iterations*, *Extrapolation* seems to produce the best runtime and fewest
496 iterations. Implementation requires only storing an extra set of data used in extrapolating the
497 approximate terms but, using the adaptive timestepping discussed here, this data is stored anyway.

498 The choice of *Extrapolation and Iterations* may work best for this problem, but for other prob-
499 lems or initial conditions, another choice may fare better. It is recommended, as in this case, that
500 a short-term simulation be performed for the different choices of approximate terms and whether
501 or not to perform extra iterations. The small cost of these short runs may allow for a more efficient
502 run for actual simulations. It is also recommended that one examines the results to make sure that
503 the scheme is not only fast with the choice, but sufficiently accurate.

504 The numerical solution agrees reasonably well with the behavior seen in experiments. This is in
505 part because the model was derived for the case when a particle-rich ridge forms. This is seen in the
506 experiments for high angles of inclination and high concentrations, but will occur in model for all
507 concentrations and angles. The particle-rich ridge in the simulations is two thin layers of particles,
508 one which originates at the front of the flow and the other from the troughs of the emerging fingers,
509 which may not be physical.

510 The current model assumes a constant, or average, particle concentration throughout the fluid
511 layer in the z -direction. The same is true for the velocity, which is averaged in the z -direction.
512 Theory exists for the vertical movement of the particles [5], whether they will settle to the inclined
513 plane or form a ridge, and incorporating this behavior into a new model is the current research of
514 the authors. It is hoped that the current numerical scheme will be adaptable to this new model.

515 Acknowledgements

516 The authors would like to thank Chris Anderson, Nebojsa Murisic, Joseph Teran, and Tom
517 Witelski for their helpful comments and discussions. The images from experiments are courtesy of
518 Joyce Ho, Vincent Hu, Paul Latterman, Trystan Koch, and Kanhui Lin, and were performed during
519 the Research Experience for Undergraduates (REU) program at the University of California, Los
520 Angeles during the summer of 2009. This research was supported by NSF grants DMS-0601395
521 and DMS-1048840 and a grant from the UC Lab Fees Research fund 09-LR-04-116471-BERA.

522 [1] J.W. Barrett, J.F. Blowey, J. Garcke, Finite element approximation of a fourth order nonlinear
523 degenerate problem, Numer. Math. 80 (1998) 525-556.

- 524 [2] R.M. Beam, R.F. Warming, Alternating direction implicit methods for parabolic equations
525 with a mixed derivative, *SIAM J. Sci. Stat. Comp.* 1 (1980) 131-159.
- 526 [3] A.L. Bertozzi, M.P. Brenner, Linear stability and transient growth in driven contact lines,
527 *Phys. Fluids* 9 (1997) 530-539.
- 528 [4] A.L. Bertozzi, M.P. Brenner, T.F. Dupont, L.P. Kadanoff, Singularities and similarities in
529 interface flows, *Trends and Perspectives in Applied Mathematics*, L. Sirovich, ed., Springer-
530 Verlag Applied Mathematical Sciences, New York, 1994, pp. 155-208.
- 531 [5] B. Cook, A theory for particle settling and shear-induced migration in thin film flow, *Phys.*
532 *Rev. E Stat. Nonlin. Soft Matter Phys.* 78 (2008).
- 533 [6] B.P. Cook, A.L. Bertozzi, A.E. Hosoi, Shock solutions for particle-laden thin films, *SIAM J.*
534 *Appl. Math.* 68 (2008) 760-783.
- 535 [7] B. Cook, O. Alexandrov, A. Bertozzi, Linear stability of particle-laden thin films, *Eur. Phys.*
536 *J. Spec. Top.* 166 (2009) 77-81.
- 537 [8] I.J.D. Craig, A.D. Sneyd, An alternating-direction implicit scheme for parabolic equations
538 with mixed derivatives, *Comput. Math. Applic.* 16 (1988) 341-350.
- 539 [9] G. Grun, M. Rumpf, Nonnegativity preserving convergent schemes for the thin film equation,
540 *Numer. Math.* 87 (2000) 113-152.
- 541 [10] G. Grun, M. Rumpf, Simulations of singularities and instabilities arising in thin film flow,
542 *Euro. J. Appl. Math.* 12 (2001) 293-320.
- 543 [11] N. Grunewald, R. Levy, M. Mata, T. Ward, A.L. Bertozzi, Self-similarity in particle-laden
544 flows at constant volume, *J. Eng. Math.* 66 (2010) 53-63.
- 545 [12] H. Happel, H. Brenner, *Low Reynolds number hydrodynamics with special applications to*
546 *particulate media*, Prentice-Hall, Englewood Cliffs, NJ, 1965.
- 547 [13] H. Huppert, Flow and instability of a viscous current down a slope, *Nature* 300 (1982) 427-429.
- 548 [14] S. Karaa, A high-order ADI method for parabolic problems with variable coefficients, *Int. J.*
549 *Comput. Math.* 86 (2009) 109-120.
- 550 [15] L. Kondic, Instabilities in gravity driven flow of thin fluid films, *SIAM Rev. Soc. Ind. Appl.*
551 *Math.* 45 (2003) 95-115.
- 552 [16] J.K. Kowalski, A finite-difference method for parabolic differential equations with mixed
553 derivatives, *Math. Comput.* 25 (1971) 675-698.
- 554 [17] I.M. Krieger, Rheology of monodisperse lattices, *Adv. Colloid Interface Sci.* 3 (1972) 111-136.
- 555 [18] P.D. Lax, *Hyperbolic systems of conservation laws and mathematical theory of shock waves*,
556 *CBMS-NSF Regional Conference Series in Applied Mathematics* 11 (1973).
- 557 [19] D. Leighton, Ph.D. thesis, Stanford Univ., Stanford, California, 1985.

- 558 [20] D. Leighton, A. Acrivos, The shear-induced migration of particles in concentrated suspensions,
559 J. Fluid Mech. 181 (1987) 415-439.
- 560 [21] X-Z. Liu, X. Cui, J-G. Sun, FDM for multi-dimensional nonlinear coupled system of parabolic
561 and hyperbolic equations, J. Comput. Appl. Math. 186 (2006) 432-449.
- 562 [22] A.V. Lyushnin, A.A. Golovin, L.M. Pismen, Fingering instability of thin evaporating liquid
563 films, Phys. Rev. E Stat. Nonlin. Soft Matter Phys. 65 (2002).
- 564 [23] S. McKee, Alternating direction methods for a system of parabolic equations in two space
565 dimensions with a mixed derivative, J. Inst. Math. Appl. 8 (1971) 376-385.
- 566 [24] M. Naumov, A.H. Sameh, A tearing-based hybrid parallel banded linear system solver, J.
567 Comput. Appl. Math. 226 (2009) 306-318.
- 568 [25] C.V. Pao, Numerical analysis of coupled systems of nonlinear parabolic equations, SIAM J.
569 Numer. Anal. 36 (1999) 393-416.
- 570 [26] D.W. Peaceman, H.H. Rachford, Jr., The numerical solution of parabolic and elliptic differen-
571 tial equations, J. Soc. Ind. Appl. Math. 3 (1955) 28-41.
- 572 [27] E. Polizzi, A.H. Sameh, A parallel hybrid banded system solver: The SPIKE algorithm, Parallel
573 Comput. 32 (2006) 177-194.
- 574 [28] J.F. Richardson, W.N. Zaki, Sedimentation and fluidization: Part I, Trans. Inst. Chem. Eng.
575 32 (1954) 35-53.
- 576 [29] Satteluri, R.K. Iyengar, M.K. Jain, Comparative study of two and three level ADI methods for
577 parabolic equations with a mixed derivative, Int. J. Numer. Methods Eng. 10 (1976) 1309-1315.
- 578 [30] J.J. Stickel, R.L. Powell, Fluid mechanics and rheology of dense suspensions, Annu. Rev. Fluid
579 Mech. 37 (2005) 129-149.
- 580 [31] P. Sun, R.D. Russell, J. Xu, A new adaptive local mesh refinement algorithm and its applica-
581 tions on fourth order thin film flow problem, J. Comput. Phys. 224 (2007) 1021-1048.
- 582 [32] T. Ward, C. Wey, R. Glidden, A.E. Hosoi, A.L. Bertozzi, Experimental study of gravitation
583 effects in the flow of a particle-laden thin film on an inclined plane, Phys. Fluids 21 (2009).
- 584 [33] M.R.E. Warner, R.V. Craster, O.K. Matar, Fingering phenomena associated with insoluble
585 surfactant spreading on thin liquid films, J. Fluid Mech. 510 (2005) 169-200.
- 586 [34] T. Witelski, M. Bowen, ADI schemes for higher-order nonlinear diffusion equations, Appl.
587 Numer. Math. 45 (2003) 331-351.
- 588 [35] J. Zhou, B. Dupuy, A.L. Bertozzi, A.E. Hosoi, Theory for shock dynamics in particle-laden
589 thin films, Phys. Rev. Lett. 94 (2005).
- 590 [36] L. Zhornitskaya, A.L. Bertozzi, Positivity-preserving numerical schemes for lubrication-type
591 equations, SIAM J. Numer. Anal. 37 (2000) 523-555.