



A faster optimal solver for thin film flows

M.A. Aljohani^{a,b}, P.K. Jimack^{a,*}, M.A. Walkley^a

^a School of Computing, University of Leeds, United Kingdom

^b School of Mathematics, Taibah University, Saudi Arabia

ARTICLE INFO

Article history:

Received 16 March 2022

Received in revised form 21 September 2022

Accepted 23 October 2022

Available online 26 October 2022

Keywords:

Thin film flow

Nonlinear systems

Multigrid

Preconditioned Newton-Krylov

ABSTRACT

A new, multigrid-based, algorithm is proposed for the solution of the discretized lubrication equations which are widely used to model a broad class of thin film flow phenomena. The approach is based upon the use of a Newton-Krylov solver for the nonlinear algebraic systems of equations that arise following mesh-based spatial discretizations and implicit time discretizations. The novel contribution is to propose a block-based preconditioner that includes two applications of algebraic multigrid (AMG) as a key component: thus allowing AMG techniques to be applied in the solution of thin film flow problems for the first time. An implementation of this preconditioned solver is demonstrated for a typical thin film scenario of free-surface flow down a non-smooth inclined plane, considering both steady-state and time-dependent configurations. Systematic computational testing is undertaken in comparison with two other state-of-the-art multigrid methods in order to demonstrate the substantial computational advantages of the proposed algorithm.

© 2022 The Author(s). Published by Elsevier B.V. on behalf of IMACS. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Thin film flow problems are widely studied due to their great importance across a broad range of applications. In recent years numerical methods for the simulation of such flows, based upon the lubrication (or long-wave) approximation, have advanced significantly, and a number of “optimal” solvers have been developed [21,30]. In this context, “optimal” means that the computational work (and therefore the execution time) grows linearly with the number of degrees of freedom being used in the spatial discretization. This is a significant achievement since the underlying system of partial differential equations (PDEs) is parabolic and fourth order, which means it cannot be solved explicitly in time without a prohibitive time-step size restriction due to instability. Furthermore, the problem is nonlinear, which leads to large systems of nonlinear algebraic equations arising at each implicit time step. The key to solving these systems optimally has been the development of multigrid methods: either designed specifically for the nonlinear system [21,30] or applied in conjunction with linearization schemes such as Newton’s method [27,50].

This paper describes an improved optimal solution method for the discretized systems of PDEs modeling thin film flows. The technique that we propose is based upon a new preconditioner for the Jacobian matrix that arises from a Newton linearization of the discrete nonlinear system at each implicit time step. At the heart of this preconditioner, which we contrast favorably to the application of more traditional multigrid techniques to this problem, is a black-box algebraic multigrid solver: thus making the solution strategy straightforward to implement and apply in practice.

* Corresponding author.

E-mail addresses: masjohani@taibahu.edu.sa (M.A. Aljohani), P.K.Jimack@leeds.ac.uk (P.K. Jimack), M.A.Walkley@leeds.ac.uk (M.A. Walkley).

In the remainder of this introduction we provide some further, brief, background to thin film flows and multilevel solvers. The following two sections then describe the thin film PDEs (and their discretization) and the relevant multigrid solution algorithms (both traditional and our proposed preconditioned Newton-Krylov approach) in more detail. These are followed by a discussion of numerical simulations that have been undertaken and some conclusions.

Understanding the flow of thin liquid films is of enormous importance for the description, explanation and control of a vast array of natural and engineering phenomena and processes. Occurrences in nature include thin layers of fluid on the eye [10,16] or lungs [24,28], which have enormous impact upon the function of these organs, as well as the effectiveness of therapeutic interventions [25]. Elsewhere in biology such films are essential for the locomotion of some animals [29] or the survival of bio-films [17]. Furthermore, these flows also control natural phenomena at much larger length scales, such as glaciation [41] or the formation of lava fields [37,49].

Similarly, in engineering applications thin films occur at a wide range of length scales, where they are drivers across multiple problem domains. Perhaps the most frequently occurring is in the coating of surfaces, where a multitude of processes have been developed, each of which requires the understanding and control of thin film flows. Examples include: roll coating, which uses rollers to apply liquid films onto continuous substrates [52]; spin coating, which employs centrifugal force [42]; gravure coating, which uses engraved rollers to transfer liquid onto a moving substrate [43]; spray coating, typically used to distribute particle-laden drops onto the substrate [13]; and drop casting, which uses a needle-like applicator [47]. Thin liquid films also occur in many other engineering applications however, such as in the lubricating film between mechanical elements [3,12] or even controlling rain-wind-induced vibrations of cable-stayed bridges [33].

As a consequence of the importance and diversity of the applications of thin film flows there has been significant research undertaken in recent years into the development of efficient numerical methods for such problems. These include, for example, the use of mesh adaptivity [3,30], and of parallel computing [20,31]. The focus of this paper however is the development of a computational algorithm with optimal complexity: that is, a solution time that grows linearly with the number of degrees of freedom in the spatial discretization. Such techniques are typically based upon multigrid or multilevel methods, which are briefly introduced in the next subsection.

Multigrid (MG) methods have been successfully applied to the solution of elliptic PDEs and systems for many years [9,11]. Initially developed for linear problems, the approach was soon generalized to nonlinear equations via the FAS (full approximation scheme) and Newton-multigrid methods [48].

The high-level idea behind the technique is to solve the discrete equations on a fine grid (using, for example, a finite difference or a finite element discretization) by first eliminating the high frequency components of the error cheaply and quickly. Following this *pre-smoothing* stage a *coarse grid correction* is undertaken to remove the lower frequency components of the error on a coarser discretization: this second step is implemented recursively, by applying a smoother and further correction, until a coarsest level is reached. Finally, a *post-smoother* is applied at each level following each coarse grid correction. Importantly, it is possible to generalize this idea to a purely algebraic formulation in which the coarser grids do not need to be generated explicitly. This algebraic multigrid (AMG) approach is most commonly applied as a preconditioner rather than a solver [34,38] however, up until now, AMG has not been applied to the solution of the equations of thin film flow.

More recent extensions have seen MG applied directly to systems of PDEs (elliptic and parabolic with implicit time-stepping) [21,39], or applied as components of more complex preconditioners in mixed formulations. Examples of the latter include fast solvers for the incompressible Navier-Stokes equations [7,18] or generalizations of these to multiphase flow problems [5]. A similar approach is taken in this paper, where we propose, implement and test a new block-preconditioner for the systems of equations arising from finite difference discretizations of the thin film equations that are introduced in the following section.

2. Thin film flows

As highlighted in the Introduction, there are many important problems which are controlled by the flow of a thin liquid film. In this section we briefly describe the standard simplification to the incompressible Navier-Stokes equations that allows the system to be reduced to the Reynolds' equation, which is a fourth order nonlinear parabolic PDE with the film thickness as its dependent variable. In practice this is expressed as a pair of coupled second order equations which may be discretized in space and time: we illustrate this via a five-point finite difference scheme and the simplest implicit time stepping scheme. This section concludes with a discussion of the resulting system of nonlinear algebraic equations that must be solved at each time step.

2.1. The long-wave approximation

Generally, thin film flows involve a liquid that is bounded between a solid substrate and a free surface with another fluid, such as air (though, in the case of lubricating flows, the film may separate two solid surfaces [3]). The distinguishing features of the applications being discussed here are that the motion and pressure gradient perpendicular to the substrate may be neglected relative to the flow parallel to it. More precisely, we assume that the flow is dominated by the viscous forces and that the ratio, ε , of the length scale of the thickness of the film to its span is small. This allows the Stokes' equations

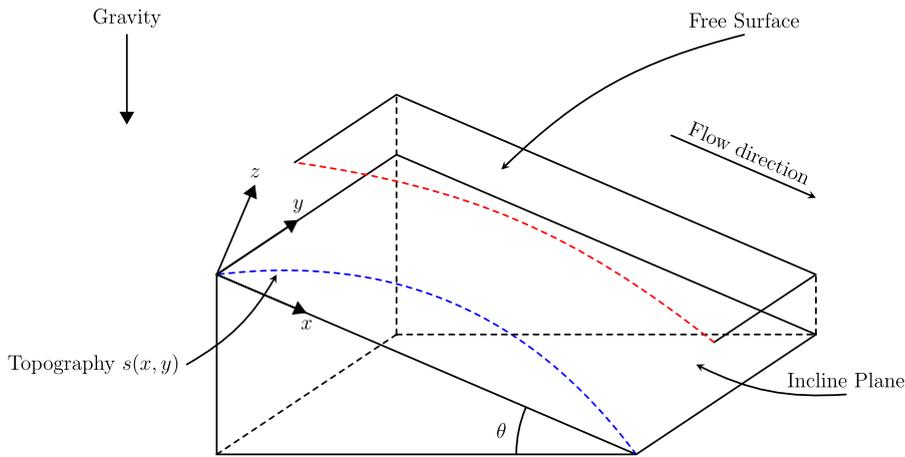


Fig. 1. Thin film flow over a bed topography with shape $s(x, y)$ on a substrate inclined at an angle θ .

to be approximated by terms that are leading order in ε , with higher order terms neglected: known as the long-wave (or lubrication) approximation.

For the sake of clarity, in this paper we consider the application of this long-wave approach to the specific case of a fully-developed flow down an inclined plane, as described in detail in [19] and illustrated in Fig. 1. We do not repeat the derivation of [19] here, however they describe a non-dimensionalization that yields the following system of PDEs for the film thickness, $h(x, y, t)$ and the pressure field, $p(x, y, t)$.

$$\frac{\partial h}{\partial t} = \frac{\partial}{\partial x} \left[\frac{h^3}{3} \left(\frac{\partial p}{\partial x} - 2 \right) \right] + \frac{\partial}{\partial y} \left[\frac{h^3}{3} \left(\frac{\partial p}{\partial y} \right) \right], \tag{1}$$

where

$$p = -6 \Delta(h + s) + 2\sqrt[3]{6} N(h + s). \tag{2}$$

Here $h(x, y)$ is the height of the film above the substrate, Δ is the two-dimensional Laplacian and $s(x, y)$ captures the substrate topology (i.e. variations in the height of the inclined plane). The constant $N = Ca^{\frac{1}{3}} \cot \theta$, where Ca is the Capillary number (which indicates the ratio of viscosity to surface tension) and θ is the angle of inclination of the substrate.

To further simplify the considerations in this paper, though without any loss of generality, we restrict our consideration to the case of flow down a vertical wall: hence $\theta = 90^\circ$, which means $N = 0$. We also assume that domain boundaries are sufficiently far from any topographic features that the flow is uniform at this “far field”.

Together, these assumptions mean that Equation (2) simplifies to

$$p = -6 \Delta(h + s), \tag{3}$$

and we may prescribe straightforward Dirichlet boundary conditions on the perimeter of the domain $(x, y) \in [X_1, X_2] \times [Y_1, Y_2]$. For our given non-dimensionalization (from [19]) these may take the following form:

$$\begin{aligned} h(X_1, y) &= h(X_2, y) = h(x, Y_1) = h(x, Y_2) = 1, \\ p(X_1, y) &= p(X_2, y) = p(x, Y_1) = p(x, Y_2) = 0. \end{aligned} \tag{4}$$

Finally, we define initial conditions for each variable as $h(x, y, t = 0) = h_0(x, y)$ and $p(x, y, t = 0) = p_0(x, y)$.

Note that it is equally common in the literature to present the system (1) and (2) as a single fourth order parabolic PDE for the dependent variable h . An advantage of expressing as a coupled system of second order equations is to allow greater flexibility in the spatial discretization: when using the finite element method, for example, only C^0 elements are required (as opposed to C^1 elements for a single fourth order PDE); and when approximating with finite differences, as in this paper, a more compact stencil is possible without sacrificing computational accuracy (see below).

2.2. Discretization

The system (1) and (3) may be solved through the application of a finite difference method in space and an implicit time-stepping scheme. In this example we choose the standard five-point stencil for the second order terms in (1) and (3) and the implicit Euler method for Equation (1). Of course it would also be possible to select a second order implicit scheme in time (to match the second order spatial discretization) and this would not substantively alter any of the principles in what follows.

Assuming that the spatial domain is divided into N intervals in the x -direction (with spacing Δx) and M intervals in the y direction (with spacing Δy), this discretization scheme leads to the following nonlinear system of algebraic equations at each grid point (i, j) ($2 \leq i \leq N$ and $2 \leq j \leq M$), and each new time level $n + 1$:

$$\begin{aligned} & \frac{h_{i,j}^{n+1} - h_{i,j}^n}{\Delta t} \\ & - \left(\frac{1}{\Delta x} \right) \left[\left(\frac{h_{i,j}^{n+1} + h_{i+1,j}^{n+1}}{2} \right)^3 \left(\frac{p_{i+1,j}^{n+1} - p_{i,j}^{n+1}}{\Delta x} \right) - 2 \right] - \left(\frac{h_{i,j}^{n+1} + h_{i-1,j}^{n+1}}{2} \right)^3 \left(\frac{p_{i,j}^{n+1} - p_{i-1,j}^{n+1}}{\Delta x} \right) - 2 \right] \\ & - \left(\frac{1}{\Delta y} \right) \left[\left(\frac{h_{i,j}^{n+1} + h_{i,j+1}^{n+1}}{2} \right)^3 \left(\frac{p_{i,j+1}^{n+1} - p_{i,j}^{n+1}}{\Delta y} \right) - 2 \right] - \left(\frac{h_{i,j}^{n+1} + h_{i,j-1}^{n+1}}{2} \right)^3 \left(\frac{p_{i,j}^{n+1} - p_{i,j-1}^{n+1}}{\Delta y} \right) - 2 \right] = 0, \end{aligned} \tag{5}$$

and

$$\begin{aligned} & p_{i,j}^{n+1} + \frac{6}{(\Delta x)^2} [(h_{i+1,j}^{n+1} + s_{i+1,j}) - 2(h_{i,j}^{n+1} + s_{i,j}) + (h_{i-1,j}^{n+1} + s_{i-1,j})] \\ & + \frac{6}{(\Delta y)^2} [(h_{i,j+1}^{n+1} + s_{i,j+1}) - 2(h_{i,j}^{n+1} + s_{i,j}) + (h_{i,j-1}^{n+1} + s_{i,j-1})] = 0. \end{aligned} \tag{6}$$

Here Δt is the time step size, $s_{i,j}$ denotes $s(x, y)$ evaluated at grid point (i, j) , all values with superscript $n + 1$ are unknowns and all values with superscript n are known from the previous time level (or the initial condition when $n = 0$).

Note that this discretization is justified in detail in prior work, such as [21] and [19]. In the former there is an explanation of the treatment of the cubic term in (1), whilst in the latter it is demonstrated that the discretization given by (5) and (6) compares very favorably with the finite element solution of the Navier-Stokes equations at low Reynolds number. In particular, it is demonstrated that the solution of this discrete system provides an accurate and robust approximation to the thin film flow across a range of topographical features. Subsequent research has shown this discretization approach to be applicable, and highly effective, across a number of similar thin film flow applications [1,2,32,44].

2.3. Nonlinear system

The nonlinear algebraic system of equations arising from the above discretization can be written as follows:

$$F_i(U) = 0, \quad i = 1, \dots, neq, \tag{7}$$

where $neq = 2 \times (N - 1) \times (M - 1)$ is the total number of unknowns in Equations (5) and (6). Furthermore,

$$F = \begin{pmatrix} F_p \\ F_h \end{pmatrix}, \tag{8}$$

where

$$\begin{aligned} & [F_h(U)]_{i,j} = \frac{h_{i,j}^{n+1} - h_{i,j}^n}{\Delta t} \\ & - \left(\frac{1}{\Delta x} \right) \left[\left(\frac{h_{i,j}^{n+1} + h_{i+1,j}^{n+1}}{2} \right)^3 \left(\frac{p_{i+1,j}^{n+1} - p_{i,j}^{n+1}}{\Delta x} \right) - 2 \right] - \left(\frac{h_{i,j}^{n+1} + h_{i-1,j}^{n+1}}{2} \right)^3 \left(\frac{p_{i,j}^{n+1} - p_{i-1,j}^{n+1}}{\Delta x} \right) - 2 \right] \\ & - \left(\frac{1}{\Delta y} \right) \left[\left(\frac{h_{i,j}^{n+1} + h_{i,j+1}^{n+1}}{2} \right)^3 \left(\frac{p_{i,j+1}^{n+1} - p_{i,j}^{n+1}}{\Delta y} \right) - 2 \right] - \left(\frac{h_{i,j}^{n+1} + h_{i,j-1}^{n+1}}{2} \right)^3 \left(\frac{p_{i,j}^{n+1} - p_{i,j-1}^{n+1}}{\Delta y} \right) - 2 \right], \end{aligned} \tag{9}$$

for each interior grid point (i, j) , and similarly

$$\begin{aligned} & [F_p(U)]_{i,j} = p_{i,j} + \frac{6}{(\Delta x)^2} [(h_{i+1,j} + s_{i+1,j}) - 2(h_{i,j} + s_{i,j}) + (h_{i-1,j} + s_{i-1,j})] \\ & + \frac{6}{(\Delta y)^2} [(h_{i,j+1} + s_{i,j+1}) - 2(h_{i,j} + s_{i,j}) + (h_{i,j-1} + s_{i,j-1})]. \end{aligned} \tag{10}$$

In Equation (7) U is a vector of size neq , ordered with the film thickness unknowns followed by the pressure unknowns:

$$U = \begin{pmatrix} h^{n+1} \\ p^{n+1} \end{pmatrix}. \tag{11}$$

Hence, U is a vector of all of the unknowns $h_{i,j}^{n+1}$ and $p_{i,j}^{n+1}$, representing the computed film thickness and pressure at grid point (i, j) at the end of time step n . For notational convenience we will drop the superscript $n + 1$ for these computed h and p values in the remainder of this paper.

In order to solve Equations (7) it will be necessary to evaluate the Jacobian of the system. This may be computed numerically, using finite differences for example, however, due to its sparsity, it is generally more computationally efficient to evaluate the non-zero entries analytically. In order to do this it is convenient to write the Jacobian matrix, J in terms of the following blocks:

$$J = \begin{bmatrix} \frac{\partial F_p}{\partial h} & \frac{\partial F_p}{\partial p} \\ \frac{\partial F_h}{\partial h} & \frac{\partial F_h}{\partial p} \end{bmatrix}. \tag{12}$$

Note that in the steady-state case the first term in Equation (5), which is the first term on the right-hand side of Equation (7), is not present. In this steady-state case let J_S represent the Jacobian matrix:

$$J_S = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}. \tag{13}$$

For the time dependent case we may express the Jacobian, J_T say, as a modification to J_S . This modification is required only for the J_{21} block, which represents the derivative of equation F_h with respect to h . We may therefore express J_T as follows:

$$J_T = \begin{bmatrix} J_{11} & J_{12} \\ (\frac{-1}{\Delta t})I + J_{21} & J_{22} \end{bmatrix}. \tag{14}$$

We generate the entries of the blocks in matrices (13) and (14), J_{11} , J_{12} , J_{21} and J_{22} , from Equations (5) and (6) by exact differentiation. By considering only the non-zero entries of each block it is possible to build and store the analytical Jacobian efficiently in a sparse matrix format. Furthermore, an important feature of the diagonal blocks J_{11} and J_{22} is that they are each the result of discretization (and, in the case of J_{22} , linearization) of second order operators. This has two important consequences: pertaining to stability/convergence; and to the development of the efficient preconditioner that is proposed in this paper. The latter point is expanded upon in detail in the following section, whilst the former is briefly discussed here in a less formal manner.

It is well known that, for fourth-order elliptic problems, mixed formulations, such as that considered in this work, have a number of advantages. As previously noted, this includes the ability to use simpler finite elements or finite difference stencils, however these advantages also extend to the resulting discrete systems. As discussed in [46], for example, the condition number in the mixed formulation is $O((\Delta x)^{-2})$ compared to $O((\Delta x)^{-4})$ when fourth order differences are used directly. On the other hand, the stability and convergence theory for the mixed formulation becomes more complex since we replace a convex energy functional by a non-convex saddle-point problem. Whilst we are not able to present a formal proof of stability for this nonlinear problem, we suggest that the presence of discrete second order operators in the diagonal blocks of (13) and (14) provides a strong analogy with the Ciarlet-Raviart formulation of the biharmonic equation for which stability and convergence results are long established ([14,22]). Combined with the empirical evidence of significant prior work, [1,2,19,21,32,44], we argue that this is an important, reliable and efficient discretization approach for this class of problem.

3. Solution algorithms

In this section we first describe a standard Newton multigrid approach to solving Equations (7). This is based upon the use of a geometric multigrid V-cycle to solve the linear system of equations at each Newton iteration. In the following subsection we then present our proposed solution technique for (7). This is also based upon Newton’s method but, instead of applying multigrid directly to each linear system, we use a Krylov subspace iterative method (GMRES) with a carefully designed preconditioner. This preconditioner is itself based upon the application of AMG solves. Finally, in Subsection 3.3, we illustrate how a nonlinear multigrid solver, the full approximation scheme (FAS), may be applied directly to solve Equations (7). This is based upon previously published work [21], and will be used as a baseline to demonstrate the improvements in solution efficiency using our proposed solver.

3.1. Newton-multigrid

Application of Newton’s method to the nonlinear equations (7) requires a linear system of the form

$$J \delta = -F \tag{15}$$

to be solved at each iteration. Using the orderings given by (8) and (11) leads the Jacobian to have the block structure noted in (12). Hence, (15) becomes:

Table 1

The maximum and minimum eigenvalues (or real part in the complex case) of the coefficient matrix J and the right-preconditioned matrix JP_0^{-1} (in this case, for the maximum, we quote the eigenvalue with the largest imaginary part since the real parts are indistinguishable).

Grid level	Min $\lambda(J)$	Max $\lambda(J)$	Min $\lambda(JP_0^{-1})$	Max $\lambda(JP_0^{-1})$
3	-0.1961 - 0.2156i	-29.9876	1.0000	1.0000 + 6.7314i
4	-0.2094 - 0.2153i	-122.1418	1.0000 - 0.0000i	1.0000 + 6.7168i
5	-0.2128 - 0.2151i	-490.7803	1.0000 - 0.0000i	1.0000 + 6.7129i

$$\underbrace{\begin{pmatrix} \frac{\partial F_p}{\partial h} & \frac{\partial F_p}{\partial p} \\ \frac{\partial F_h}{\partial h} & \frac{\partial F_h}{\partial p} \end{pmatrix}}_J \begin{pmatrix} \delta_h \\ \delta_p \end{pmatrix} = - \begin{pmatrix} F_p \\ F_h \end{pmatrix}. \tag{16}$$

Here, J and F are evaluated at $\begin{pmatrix} h^k \\ p^k \end{pmatrix}$, the latest Newton iterate, which is then updated by

$$\begin{pmatrix} h^{k+1} \\ p^{k+1} \end{pmatrix} = \begin{pmatrix} h^k \\ p^k \end{pmatrix} + \begin{pmatrix} \delta_h \\ \delta_p \end{pmatrix}. \tag{17}$$

For the Newton-MG algorithm, standard MG V-cycles are used to approximate the solution of Equation (15) [11,48]. To implement this a sequence of nested grids is first generated. Each Newton step begins by restricting the latest solution estimate on the finest grid, $\begin{pmatrix} h^k \\ p^k \end{pmatrix}$, to every other level and then using these restrictions to evaluate the Jacobian matrix on each grid. At each V-cycle a weighted red-black Gauss-Seidel iteration is used as the smoother (this is a pointwise iteration that updates δ_h and δ_p simultaneously at each node by solving a 2×2 linear system) and then the resulting residual is restricted to the next mesh for the coarse grid correction. Further implementation details may be found in [4].

3.2. Preconditioned Newton-Krylov

This section focuses on a Newton-Krylov algorithm with our new AMG-based preconditioner. The system (7) is again solved by Newton’s method however, at each Newton iteration (16) is solved using preconditioned GMRES, reflecting the fact that J is a sparse, non-symmetric matrix [40,51].

To develop our preconditioner, let us rewrite the linear system (16) in the following block matrix form:

$$\begin{pmatrix} \mathbf{K} & \mathbf{I} \\ \mathbf{B} & \mathbf{K}_\alpha \end{pmatrix} \begin{pmatrix} \delta_h \\ \delta_p \end{pmatrix} = \begin{pmatrix} F_p \\ F_h \end{pmatrix} \tag{18}$$

where, using the notation of Equations (12)-(14), $\mathbf{K} = J_{11}$, J_{12} is the identity matrix (see Equation (10)), $\mathbf{B} = J_{21} (+ \frac{1}{\Delta t} I)$ and $\mathbf{K}_\alpha = J_{22}$.

From [35], it is known that an ideal preconditioner (in the sense that it will converge in just two iterations) for the system (18) would take the form:

$$P = \begin{pmatrix} \mathbf{S} & \mathbf{0} \\ \mathbf{B} & \mathbf{K}_\alpha \end{pmatrix}, \tag{19}$$

where the Schur complement matrix $S = K - IK_\alpha^{-1}B$. However the cost of applying (19) would be prohibitive, and so simplifications of it must be considered. One such preconditioner could be

$$P_0 = \begin{pmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{B} & \mathbf{K}_\alpha \end{pmatrix}, \tag{20}$$

in which the approximation $K \simeq S$ has been made. An assessment of the eigenvalues of the right-preconditioned matrix JP_0^{-1} shows that they do not grow significantly as the finite difference grid is refined (unlike the eigenvalues of J), as illustrated in Table 1 for example. This behavior is consistent with an effective preconditioner provided it continues to hold as the finite difference grid is further refined [51].

Unfortunately, even the application of (20) is too expensive for it to be a candidate as a practical preconditioning matrix for the system (16) since the blocks K and K_α must be inverted. We may note however (from Equations (5) and (6)) that these diagonal blocks represent discretizations of second order differential operators. As such, they can be inverted approximately using a multigrid algorithm: and in this work we choose to do this using an algebraic multigrid (AMG) approach. Consequently, our proposed preconditioner, P_1 , is as follows:

$$P_1 = \begin{pmatrix} \mathbf{AMG(K)} & \mathbf{0} \\ \mathbf{B} & \mathbf{AMG(K}_\alpha) \end{pmatrix}. \tag{21}$$

Here **AMG(X)** represents the action of one AMG iteration applied to matrix X. We realize this using the software implementation that is available in the Harwell Subroutine Library (HSL), including the routines HSL-MI20 for the AMG method and HSL-MI24 for the GMRES solver [6,26].

3.3. FAS

In order to assess the performance of the approach described in the previous subsection we have also implemented the nonlinear multigrid FAS algorithm [11] as a solver for Equations (7). Our implementation closely follows [21], which makes use of a linear interpolation operator and a full-weighting restriction operator to transfer between grids. For the smoother we have implemented both a point-wise nonlinear Jacobi iteration and a point-wise nonlinear red-black Gauss-Seidel iteration (the red-black ordering is selected for the Gauss-Seidel smoother because we only need to update the Jacobian evaluation after each red and black sweep, making it much more efficient than other orderings when using the five-point stencil [36]). On the coarsest grid we use Newton’s method with a sparse direct solver [23].

To illustrate the smoother more clearly, we describe the Jacobi case in more detail here. In this case we define each nonlinear Jacobi iteration as an approximate solve for the i^{th} equation of the nonlinear system (7),

$$F_i(u_1^k, \dots, u_{i-1}^k, u_i^{k+1}, u_{i+1}^k, \dots, u_{nu}^k) = 0, \tag{22}$$

for $i = 1, \dots, nu$, where k and $k + 1$ denote the current and new approximations, u_i denotes the value of the (h, p) pair at the corresponding vertex of the grid, and nu is the number of non-Dirichlet grid points (so $2 \times nu = neq$). To approximately solve this pair of equations we take a single Newton step, which requires us to evaluate J_{bi} which is one of the (2×2) diagonal blocks of the analytical Jacobian matrix. Then, at each grid point $i = 1, \dots, nu$, we solve the linear system,

$$\underbrace{\begin{pmatrix} \frac{\partial F_{pi}}{\partial h_i} & \frac{\partial F_{pi}}{\partial p_i} \\ \frac{\partial F_{hi}}{\partial h_i} & \frac{\partial F_{hi}}{\partial p_i} \end{pmatrix}}_{J_{bi}} \begin{pmatrix} \delta_{hi} \\ \delta_{pi} \end{pmatrix} = - \begin{pmatrix} F_{pi} \\ F_{hi} \end{pmatrix}. \tag{23}$$

Having solved Equation (23) we may update h and p at the corresponding node $i = 1, \dots, nu$. We actually do this update in a weighted manner:

$$\begin{pmatrix} h_i^{k+1} \\ p_i^{k+1} \end{pmatrix} = \begin{pmatrix} h_i^k \\ p_i^k \end{pmatrix} + \omega \begin{pmatrix} \delta_{hi} \\ \delta_{pi} \end{pmatrix}, \tag{24}$$

where ω is a weighting parameter whose optimal value is determined as a part of this study.

4. Steady-state numerical results

We have performed extensive numerical experiments in order to optimize the parameters selected for the FAS, Newton-MG and Newton-Krylov nonlinear multilevel schemes [4]. We then compare our best choice of parameters for each nonlinear multilevel scheme to determine the best approach for both steady-state and time-dependent test problems.

For these test problems the domain is $X_1 = -10, X_2 = 10, Y_1 = -5, Y_2 = 5, dx = (X_2 - X_1)/(mx - 1)$, and $dy = (Y_2 - Y_1)/(my - 1)$. The grid size is $my = 2^{L_{MAX}} + 1$ and $mx = 2 * (my - 1) + 1$ in directions X and Y , where L_{MAX} denotes the maximum grid level. In all the tests in this paper we used Dirichlet boundary conditions. The number of unknown grid points is defined as $nu = nx \times ny$, where $nx = mx - 2$ and $ny = my - 2$. The number of nonlinear equations and variables is $neq = 2 \times nu$. For the topography of the bed we define:

$$S(x, y) = \min(\max(d, -1), 0), \tag{25}$$

where

$$\begin{aligned} d &= \max(dx, dy) \\ dx &= \max(x, -x - 4), \\ dy &= \max(y - 2, -y - 2). \end{aligned}$$

As shown in the bottom half of Fig. 2, this corresponds to a square trough of depth 1, centered at $(-2, 0)$, with vertical walls (which are approximated by the resolution on the finest grid). In each test, we compute the solution with a single nonlinear system solve based on the initial guess $h = 1; p = 0$.

Table 2 shows the different sizes for the grid levels that we have used to solve the thin film flow system in the following subsections. In order to choose the best parameters for FAS and Newton-MG algorithms we employ two different types of smoother: weighted Jacobi and the weighted red-black Gauss-Seidel; these smoothers are applied with varying values of the parameter ω . To compare the convergence of FAS or Newton iterations (outer iterations), we use the nonlinear residual tolerance $Tol = 1e - 08$. For the Newton-Krylov solver, inner tolerance values are varied in order to control the

Table 2
The grid level, grid size, and the number of equations $neq = 2 \cdot nu$ used in the thin film flow system of equations.

Grid level	Grid size	No. of equations
4	17×33	930
5	33×65	3906
6	65×129	16002
7	129×257	64770
8	257×513	260610
9	513×1025	1045506

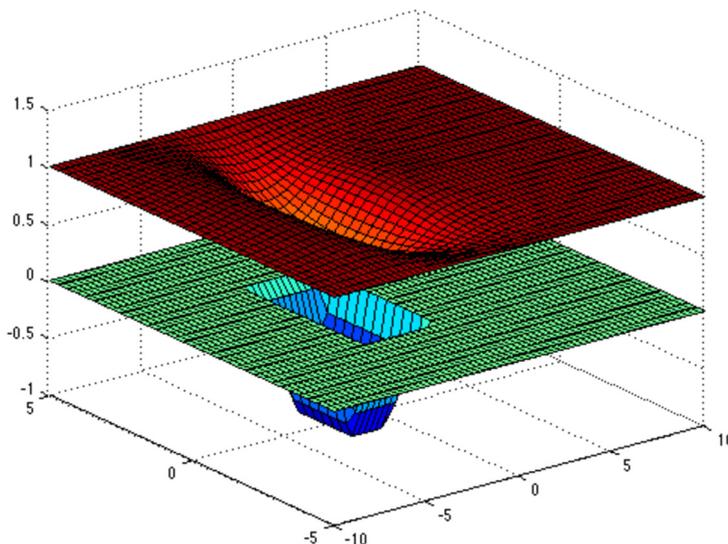


Fig. 2. Square hole bed shape s in blue (bottom) and the free surface (numerical solution) $h + s$ in red (top): note that the aspect ratio is distorted in this image in order to maximize its size. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 3
FAS performance for varying grid size for the thin film flow in the steady-state case with red-black G-S with $\omega = 1.2$, $(pre, post)_{smooth} = (1, 1)$, $Coarse\ Grid = 5$ and $Tol = 1e - 8$.

Input		Output	
Grid Size		FAS Solver	Time (sec)
Fine	Coarse	V-Cycle	-
6	5	7	2.2061
7	5	7	4.3632
8	5	7	14.3745
9	5	7	56.7912

convergence of GMRES iterations (inner iterations) and we consider different choices for the restart parameter (Restart) of GMRES iterations and the maximum number of GMRES iterations (Maxit) in each case. The results shown in the following subsections represent the best performance for each method across the parameters that we varied. All timings were taken based upon Matlab implementations running on a standard specification desktop computer with a single Intel Xeon CPU.

4.1. FAS results

For the FAS algorithm we have found that the best values of the parameters are: $\omega = 1.2$ with red-black G-S; the number of pre- and post-smooths $(pre, post) = (1, 1)$; and the coarse grid level is $G = 5$ (33×65). In Table 3, we demonstrate that run times perform optimally and the number of V-cycle iterations is fixed. This is consistent with previously published results, [21], since it implies that performance is independent of the problem size and that the FAS nonlinear multigrid algorithm executes with linear time complexity $O(N)$, where N is the total number of unknowns (which grows by a factor of 4 with each additional fine grid level).

Table 4

Newton-MG performance for a varying grid size for the thin film flow in the steady-state case with red-black G-S with $\omega = 1.2$, $(pre, post)_{smooth} = (1, 1)$ fixed V-cycle, the number of V-cycles = 3, Coarse Grid = 4 and the relative tolerance for Newton is $Tol = 1e - 8$.

Input			Output	
Grid Size		MG Solver	Newton Solver	Time (sec)
Fine	Coarse	V-Cycle	Iteration	-
6	4	3	6	1.3197
7	4	3	6	5.6554
8	4	3	7	27.3475
9	4	3	6	100.5532

Table 5

Newton-Krylov-AMG solver performance in steady-state case, using preconditioner P_1 with GMRES maximum iteration $Maxit = 300$, (the relative tolerance for GMRES is $Tol = 1e - 3$), $Restart = 20$, $(pre, post)_{smooth} = (1, 1)$ and the relative tolerance for Newton is $Tol = 1e - 8$.

Input	Output				Time (sec)
Grid size	Newton Solver	GMRES Solver			-
Fine	NNI	Min	Max	Average	-
6	6	10	13	11.16	0.7592
7	6	10	13	11.33	2.7394
8	6	10	13	11.16	11.0972
9	6	10	13	11.50	47.7340

4.2. Newton-multigrid results

For the Newton-MG algorithm we have found that the best value of the parameter ω is 1.2, the best smoother is red-black G-S, the best number for the pre- and post-smooths is $(pre, post) = (1, 1)$ and the best coarse grid size is $G = 4$. We use a fixed number of 3 V-cycles per linear solve, having tested a range of other values. As we can see from Table 4 the number of Newton iterations remains almost constant. Furthermore, since the work per Newton iteration grows linearly with the number of unknowns, we may observe that the overall execution time also grows linearly: thus implying that our Newton-MG implementation is also optimal since it executes with time complexity of approximately $O(N)$.

4.3. Preconditioned Newton-Krylov results

For our newly proposed preconditioner for the Newton-Krylov algorithm, and applying the AMG block solves using $(pre, post) = (1, 1)$, we obtained results that are presented in Table 5. As seen in this table, the computational time is once again very close to optimal and the needed number of Newton and GMRES iterations required is bounded as the grid is refined, which implies that our preconditioned Newton-Krylov algorithm also executes with approximate linear time complexity.

4.4. Discussion of steady-state results

As expected, we are able to demonstrate that all three solvers that we have considered are close to optimal in that their execution time grows approximately linearly with the total number of unknowns on the finest grid that is used. However, as illustrated in Fig. 3, there is a significant difference in the time required to complete each of these solves. For example, the application of our proposed preconditioned Newton-Krylov solver yields results of the same accuracy in approximately half the time needed when using a conventional Newton-multigrid solver.

It is also interesting to compare the performance of both Newton-based solvers against that of the FAS scheme for this problem. The preconditioned Newton-Krylov solver is approximately 20% faster than the FAS method, which is itself significantly faster than the Newton-MG solver. This last comparison is in contrast with the results of [8], which shows that the Newton-MG solver consistently outperformed FAS on the nonlinear second order problems that they considered. This suggests that it is not possible to draw absolute conclusions as to whether one scheme will be faster than the other across all nonlinear elliptic PDE problems - nevertheless, the clear outcome of the computations presented in this section is that the newly-proposed preconditioned Newton-Krylov approach is always the best choice for this family of thin film flow problems on the mesh sizes considered. Furthermore, even choosing non-optimal values for the solver parameters using this approach yields solutions in a shorter execution time than with the optimal choice of parameters for FAS or Newton-MG [4]. Detailed inspection of the lower two curves in Fig. 3 suggests that it is possible that the gap between the FAS and the preconditioned Newton-Krylov timings is slowly decreasing as the refinement level grows. However there is no reason to suspect any fundamental changes in their relative behavior beyond that already observed up to the finest grid considered here.

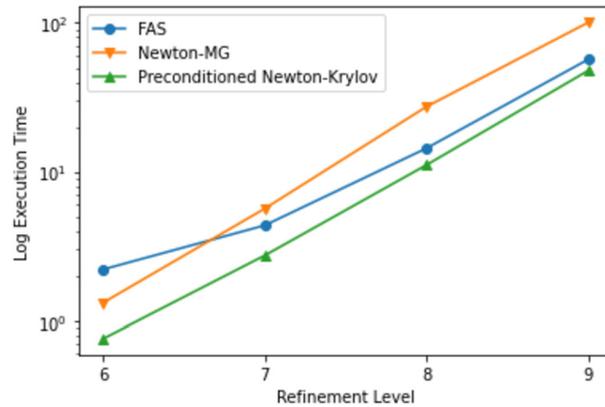


Fig. 3. Execution times for the steady-state test problem.

Table 6

FAS performance for varying grid size for thin film flow in time-dependent case with $\Delta t = 0.1$ and number of time steps = 10, with nonlinear red-black G-S with $\omega = 1$, $(pre, post)_{smooth} = (1, 1)$, Coarse Grid = 5 and the relative tolerance for FAS is $Tol = 1e - 8$.

Input		Output			
Grid Size		FAS Solver			Time (sec)
Fine	Coarse	Min	Max	Average	-
6	5	7	8	7.6	11.4158
7	5	8	9	8.8	43.3668
8	5	9	10	9.4	180.2312
9	5	9	10	9.6	746.0546

5. Time-dependent numerical results

As for the previous section, we have performed comprehensive numerical experiments to optimize the parameter selection for all three nonlinear multilevel schemes to solve this system in the time-dependent case. The computational results that are presented in this section have used the same bed shape and grid sizes that we used in the steady-state problem. Fig. 4a shows the initial conditions that we have used for the thin film flow system (specifically, that the film height above the substrate is constant). We display the numerical solutions for the thin film flow system at $T = 1$ (the end of the solution) in Fig. 4b. Throughout this subsection, the numerical results have been completed using the time step size $\Delta t = 0.1$ implying that the number of time steps taken to contrast the performance of the three nonlinear multilevel algorithms is always 10.

5.1. FAS results

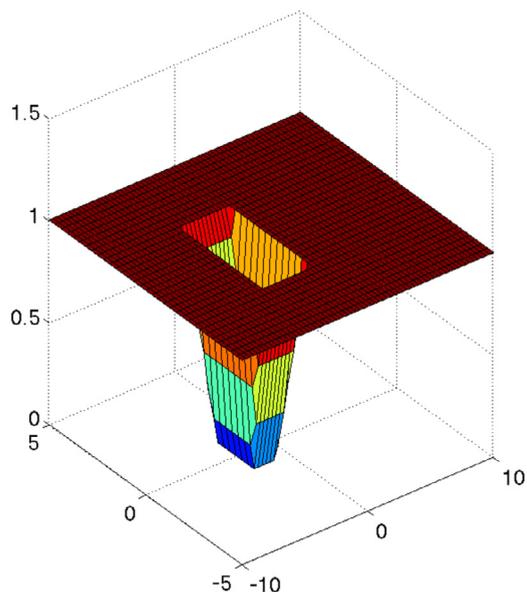
For the FAS algorithm, we found that the best value of the parameter ω is $\omega = 1$, the best smoother is red-black G-S, with the best value of the pre- and post-smooth $(pre, post) = (1, 1)$ and the best coarse grid size is $G = 5$. As shown in Table 6, we observed that the execution time grows by a factor of just over 4 as the problem size is grown by a factor of 4. This indicates that performance is almost independent of the problem size and our FAS nonlinear multigrid algorithm executes with linear time complexity that is approximately $O(N)$.

5.2. Newton-multigrid results

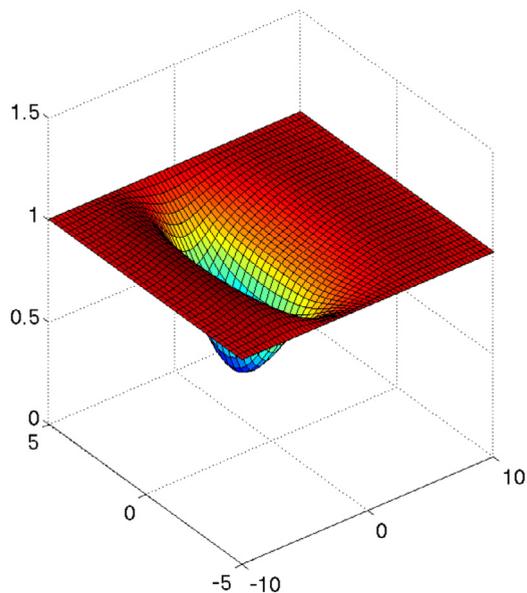
For the Newton-MG algorithm we determined that the best value of the parameter ω is $\omega = 1$, the best smoother is red-black G-S, the best value of the parameter pre- and post-smooth is $(pre, post) = (1, 1)$, and the best coarse grid size is $G = 6$. As shown in Table 7, the execution time increases by just a little more than a factor of 4 from one run to the next, which again demonstrates that our Newton-MG algorithm executes with close to linear time complexity.

5.3. Preconditioned Newton-Krylov results

For our newly proposed preconditioner applied with Newton-Krylov, the best results that we observed are presented in Table 8 using with GMRES $Tol = 1e - 4$. As we can see from this table, the required number of GMRES iterations is constant as the grid is refined, although the execution time grows very slightly worse than linearly. This implies that our Newton-MG algorithm also executes with close to linear time complexity.



(a) At Time $T=0$.



(b) At Time $T=1$.

Fig. 4. The free surface (numerical solution) $h + s$ at Time $T=0$ and $T=1$.

5.4. Discussion of time-dependent results

As with the steady-state case considered in the previous section, we again observe that the proposed preconditioned Newton-Krylov solver is clearly the best choice of the methods considered (see Fig. 5). Interestingly, all three methods have a time complexity that is close to linear but very slightly sub-optimal.

Unlike the steady-state case, there is much less to choose between the FAS and the Newton-MG schemes for this time-dependent family of problems. The key difference being that, in this case, the initial guesses at each time step (which

Table 7

Newton-MG performance for varying grid size for thin film flow in time-dependent case with $\Delta t = 0.1$ and number of time steps = 10, with $(pre, post)_{smooth} = (1, 1)$, red-black G-S with $\omega = 1$, coarse Grid = 6, fixed V-cycles (the number of V – cycle = 3) and the relative tolerance for Newton is $Tol = 1e - 8$.

Input			Output			
Grid Size		MG Solver	Newton Solver			Time (sec)
Fine	Coarse	V-Cycle	Min	Max	Average	-
7	6	3	3	5	3.8	39.8166
8	6	3	4	5	4.2	182.4501
9	6	3	4	6	4.4	793.0276

Table 8

Newton-Krylov-AMG with AMG preconditioned solver in the time-dependent case, using P_1 preconditioner with $\Delta t = 0.1$ and the number of time step = 10, GMRES with maximum iteration $Maxit = 300$, $Tol = 1e - 4$ and $Restart = 20$ with $(pre, post)_{smooth} = (1, 1)$ and the relative tolerance for Newton is $Tol = 1e - 8$.

Input		Output					Time (sec)	
Grid size		Newton Solver			GMRES Solver			-
Fine	Coarse	Min	Max	Average	Min	Max	Average	
6	3	5	3.9	15	20	18.79	7.0630	
7	4	5	4.1	16	20	18.82	25.9074	
8	4	5	4.2	16	20	18.82	109.0045	
9	4	5	4.2	16	20	18.73	476.1476	

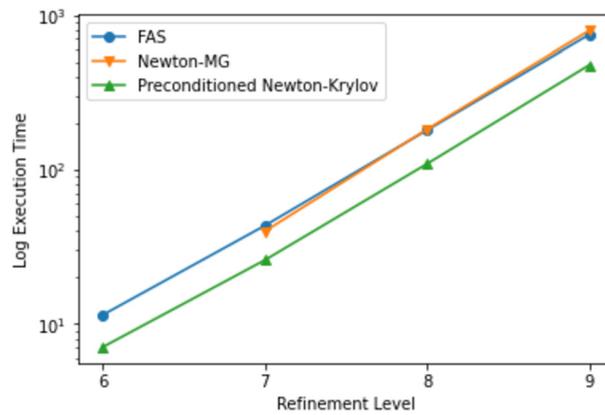


Fig. 5. Execution times for the transient test problem.

are based upon the solution at the previous time level) are far closer to the converged solution than in the steady-state examples.

Once again, we are able to conclude that, even with a non-optimal set of algorithm parameters, the preconditioned Newton-Krylov solver is faster than either of the other solvers with their optimally-tuned choice of parameters. This suggests that, for this family of thin film flows at least, it should be the preferred approach.

6. Discussion

In this paper, we have undertaken a careful investigation of the numerical solution of the standard model of thin film flow using the three different nonlinear multilevel algorithms for the discrete systems of equations representing the steady-state and implicitly-solved time-dependent problems. We have demonstrated that all three nonlinear multilevel algorithms are optimal, or very close to optimal, when used to solve the discretized equations on a sequence of finer and finer grids. Specifically, this means that FAS, Newton-MG and our new preconditioned Newton-Krylov solver all show almost linear time complexity for both steady-state and time-dependent cases.

Although we have achieved close to an optimal numerical solution in each case, the absolute execution times differ significantly. Furthermore, for each scheme there are a multitude of algorithmic parameters, the choice of which can have a notable impact on the overall solution times. Consequently, we have undertaken extensive numerical testing in order to identify the optimal combination of parameters in each case. We then made systematic quantitative comparisons between the solvers, using the “best” parameter choices for each, in both steady-state and time-dependent cases. The conclusions

from these tests clearly show that, in all cases considered, our proposed preconditioned Newton-Krylov algorithm is the best choice.

Throughout this research the development and testing of the proposed AMG-based preconditioner has only considered sequential implementation on a single CPU. Consequently, comparisons against FAS and Newton-MG have also been made only on a single CPU. An important future extension of this work would be to consider parallel implementations for different architectures. The FAS approach has been shown to work effectively on this problem in parallel across multiple CPU cores, [20,31], and has recently been applied successfully in other fluid flow applications on GPUs [45]. Whilst the AMG implementation used in this work is inherently sequential, [6,26], parallel AMG algorithms and software have been developed in recent years [15]. In principle therefore the preconditioner proposed here could be implemented in parallel too, however this would require more significant structural changes to the code organization than the relatively straightforward domain decomposition approaches used in [20,31].

In conclusion, to the authors' knowledge, this is the first paper to demonstrate how AMG may be used as the key component of the solution process when solving thin film flow problems. Furthermore, we have been able to demonstrate significant computational improvements from using this approach relative to the existing state-of-the-art methods.

Acknowledgements

Mashaël Aljohani was funded by a PhD scholarship from Taibah University, Saudi Arabia.

References

- [1] G. Ahmed, M. Sellier, Modeling the effects of absorption on spreading dynamics, *Transp. Porous Media* 112 (2016) 637–663.
- [2] G. Ahmed, M. Sellier, M. Jerny, M. Taylor, Modeling the effects of contact angle hysteresis on the sliding of droplets down inclined surfaces, *Eur. J. Mech. B, Fluids* 48 (2014) 218–230.
- [3] S. Ahmed, C.E. Goodyer, P.K. Jimack, An adaptive finite element procedure for fully-coupled point contact elastohydrodynamic lubrication problems, *Comput. Methods Appl. Mech. Eng.* 282 (2014) 1–21.
- [4] M.A. Aljohani, Multilevel numerical algorithms for systems of nonlinear parabolic partial differential equations, Ph.D. thesis, University of Leeds, 2019.
- [5] A. Alrehaili, M.A. Walkley, P.K. Jimack, M.E. Hubbard, An efficient numerical algorithm for a multiphase tumour model, *Comput. Math. Appl.* 78 (8) (2019) 2734–2745.
- [6] J. Boyle, M. Mihajlovic, J. Scott, HSL MI20: an efficient AMG preconditioner, Tech. Rep., Department of Computational and Applied Mathematics, Rutherford Appleton Laboratory, 2007.
- [7] J. Boyle, M. Mihajlović, J. Scott, HSL_MI20: an efficient AMG preconditioner for finite element problems in 3D, *Int. J. Numer. Methods Eng.* 82 (1) (2010) 64–98.
- [8] K. Brabazon, M. Hubbard, P. Jimack, Nonlinear multigrid methods for second order differential operators with nonlinear diffusion coefficient, *Comput. Math. Appl.* 68 (12) (2014) 1619–1634.
- [9] A. Brandt, Multi-level adaptive solutions to boundary-value problems, *Math. Comput.* 31 (138) (1977) 333–390.
- [10] R.J. Braun, Dynamics of the tear film, *Annu. Rev. Fluid Mech.* 44 (2012) 267–297.
- [11] W.L. Briggs, S.F. McCormick, V.E. Henson, *A Multigrid Tutorial*, SIAM, 2000.
- [12] J.W. Cahn, J.E. Hilliard, Free energy of a nonuniform system. I. Interfacial free energy, *J. Chem. Phys.* 28 (2) (1958) 258–267.
- [13] V. Charitatos, T. Pham, S. Kumar, Droplet evaporation on inclined substrates, *Phys. Rev. Fluids* 6 (8) (2021) 084001.
- [14] P.G. Ciarlet, A. Raviart, A mixed finite element method for the biharmonic equation, in: C. de Boor (Ed.), *Mathematical Aspects of Finite Elements in Partial Differential Equations*, Academic Press, New York, 1974, pp. 125–145.
- [15] Pasqua D'Ambra, Fabio Durastante, Salvatore Filippone, AMG preconditioners for linear solvers towards extreme scale, *SIAM J. Sci. Comput.* 43 (5) (2021) S679–S703.
- [16] T.A. Driscoll, R.J. Braun, J.K. Brosch, Simulation of parabolic flow on an eye-shaped domain with moving boundary, *J. Eng. Math.* 111 (1) (2018) 111–126.
- [17] H.J. Eberl, R. Sudarsan, Exposure of biofilms to slow flow fields: the convective contribution to growth and disinfection, *J. Theor. Biol.* 253 (4) (2008) 788–807.
- [18] H.C. Elman, D.J. Silvester, A.J. Wathen, *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, Numer. Math. Sci. Comput. (2014).
- [19] P. Gaskell, P. Jimack, M. Sellier, H. Thompson, M. Wilson, Gravity-driven flow of continuous thin liquid films on non-porous substrates with topography, *J. Fluid Mech.* 509 (2004) 253–280.
- [20] P. Gaskell, P. Jimack, Y.-Y. Koh, H. Thompson, Development and application of a parallel multigrid solver for the simulation of spreading droplets, *Int. J. Numer. Methods Fluids* 56 (8) (2008) 979–989.
- [21] P.H. Gaskell, P.K. Jimack, M. Sellier, H.M. Thompson, Efficient and accurate time adaptive multigrid simulations of droplet spreading, *Int. J. Numer. Methods Fluids* 45 (11) (2004) 1161–1186.
- [22] R. Glowinski, O. Pirroneau, Numerical methods for the biharmonic equation and for the two-dimensional Stokes problem, *SIAM Rev.* 21 (1979) 167–212.
- [23] N.I. Gould, J.A. Scott, Y. Hu, A numerical evaluation of sparse direct solvers for the solution of large sparse symmetric linear systems of equations, *ACM Trans. Math. Softw.* 33 (2) (2007) 10.
- [24] J.B. Grotberg, *Respiratory fluid mechanics*, *Phys. Fluids* 23 (2) (2011) 021301.
- [25] E. Hermans, M.S. Bhamla, P. Kao, G.G. Fuller, J. Vermant, Lung surfactants and different contributions to thin film stability, *Soft Matter* 11 (41) (2015) 8048–8057.
- [26] HSL, *A collection of Fortran codes for large-scale scientific computation*, <http://www.hsl.rl.ac.uk>, 2007.
- [27] G. Jouvét, E. Bueler, C. Gräser, R. Kornhuber, A nonsmooth Newton multigrid method for a hybrid, shallow model of marine ice sheets, in: *Recent Advances in Scientific Computing and Applications*, in: *Contemporary Mathematics*, vol. 586, 2013, pp. 197–205.
- [28] D. Kang, M. Chugunova, A. Nadim, A. Waring, F. Walther, Modeling coating flow and surfactant dynamics inside the alveolar compartment, *J. Eng. Math.* 113 (1) (2018) 23–43.
- [29] S. Lee, J.W. Bush, A. Hosoi, E. Lauga, Crawling beneath the free surface: water snail locomotion, *Phys. Fluids* 20 (8) (2008) 082106.
- [30] Y. Lee, H. Thompson, P. Gaskell, Three-dimensional thin film and droplet flows over and past surface features with complex physics, *Comput. Fluids* 46 (1) (2011) 306–311.

- [31] Y.C. Lee, D. Emerson, P. Gaskell, X. Gu, H. Thompson, Parallel computing as a vehicle for engineering design of complex functional surfaces, *Adv. Eng. Softw.* 42 (5) (2011) 228–236.
- [32] Y.C. Lee, H.M. Thompson, P.H. Gaskell, Three-dimensional thin film and droplet flows over and past surface features with complex physics, *Comput. Fluids* 46 (2011) 306–311.
- [33] C. Lemaître, P. Hémon, E. de Langre, Thin water film around a cable subject to wind, *J. Wind Eng. Ind. Aerodyn.* 95 (9–11) (2007) 1259–1271.
- [34] S.P. MacLachlan, L.N. Olson, Theoretical bounds for algebraic multigrid performance: review and analysis, *Numer. Linear Algebra Appl.* 21 (2) (2014) 194–220.
- [35] M.F. Murphy, G.H. Golub, A.J. Wathen, A note on preconditioning for indefinite linear systems, *SIAM J. Sci. Comput.* 21 (6) (2000) 1969–1972.
- [36] J.M. Ortega, W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, vol. 30, SIAM, 1970.
- [37] A. Osipov, A self-similar solution to the problem of lava dome growth on an arbitrary conical surface, *Fluid Dyn.* 39 (1) (2004) 47–60.
- [38] F.H. Pereira, S.L.L. Verardi, S.I. Nabeta, A fast algebraic multigrid preconditioned conjugate gradient solver, *Appl. Math. Comput.* 179 (1) (2006) 344–351.
- [39] J. Rosam, P. Jimack, A. Mullis, An adaptive, fully implicit multigrid phase-field model for the quantitative simulation of non-isothermal binary alloy solidification, *Acta Mater.* 56 (17) (2008) 4559–4569.
- [40] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* 7 (3) (1986) 856–869.
- [41] C. Schoof, R.C. Hindmarsh, Thin-film flows with wall slip: an asymptotic analysis of higher order glacier flow models, *Q. J. Mech. Appl. Math.* 63 (1) (2010) 73–114.
- [42] L. Schwartz, R. Roy, Theoretical and numerical results for spin coating of viscous liquids, *Phys. Fluids* 16 (3) (2004) 569–584.
- [43] L. Schwartz, P. Moussallp, P. Campbell, R. Eley, Numerical modelling of liquid withdrawal from gravure cavities in coating operations, *Chem. Eng. Res. Des.* 76 (1) (1998) 22–28.
- [44] M. Sellier, J.W. Grayson, L. Renbaum-Wolff, A.K. Bertram, Estimating the viscosity of a highly viscous liquid droplet through the relaxation time of a dry spot, *J. Rheol.* 59 (2015) 733.
- [45] Xiaolei Shi, Tanmay Agrwal, Chao-An Lin, Feng-Nan Hwang, Tzu-Hsuan Chiu, A parallel nonlinear multigrid solver for unsteady incompressible flow simulation on a multi-GPU cluster, *J. Comput. Methods Phys.* 414 (2020) 109447.
- [46] Gilbert Strang, George J. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall, Englewood Cliffs, NJ, 1973.
- [47] U. Tahir, M.A. Kamran, M.H. Jang, M.Y. Jeong, Thin-film coating on cylinder for fabrication of cylindrical mold: roll-to-roll nano-imprint lithography, *Microelectron. Eng.* 211 (2019) 5–12.
- [48] U. Trottenberg, C.W. Oosterlee, A. Schüller, *Multigrid*, Academic Press, Orlando, 2001.
- [49] E. Vedeneva, Lava spreading during volcanic eruptions on the condition of partial slip along the underlying surface, *Fluid Dyn.* 50 (2) (2015) 203–214.
- [50] N. Wang, S. Chang, H. Huang, Comparison of iterative methods for the solution of compressible-fluid Reynolds equation, *J. Trib. Trans. ASME* 133 (2) (2011) 021702.
- [51] A.J. Wathen, Preconditioning, *Acta Numer.* 24 (2015) 329–376.
- [52] B. Willinger, A. Delgado, Analytical prediction of roll coating with counter-rotating deformable rolls, *J. Coat. Technol. Res.* 11 (2014) 31–37.