Computational Study of Thin Film Flow

Kieran Jennings

July 22, 2016

Contents

Chapter 1

Introduction

1.1 The Aim of this Research

The primary interest of this research is to study the solutions of the Thin Film Flow model, more specifically, testing for uniqueness of solutions. Efficient solvers have been developed to solve the Thin Film Flow Model in one dimension. The Thin Film Flow Model is a system of non-linear partial differential equations which can be made discrete using the Finite Difference Method.
Newton's Method can then be used to approximately solve the system of discrete equations at each point. Convergence of Newton's method relies heavily on a sufficiently accurate initial guess. Methods to gain a good initial guess before using Newton's Method are included in this report, using such methods tries to help Newton's Method converge, although using these methods still does not guarantee convergence. Newton's method will often preferentially converge to one stable state, even if multiple solutions exist. It may be interesting to compute these further solutions. Deflation can be used to solve this problem by eliminating the previous solution Newton's method has found, therefore a second solution may be found. Non-linear partial differential equations often can not be proved to have unique solutions, therefore several tests can be conducted to try find a second solution.

1.2 Outline of this Report

In this report we consider the derivation of thin film flow equations and then the method used to form approximations to derivatives using spatial discretisation. We also explain methods to find approximations to nonlinear systems of equations. As Newton's method is the numerical method used throughout this report, we also explain some methods to gain good initial guesses that can be used to start Newton's method. Deflation is also explained at the end of Chapter 2, this is an important aspect of this research as it is crucial to finding further solutions, if any exist.
This report is designed to give a good mathematical understanding of the methods used in finding the results presented in Chapter 3.

Chapter 2

Literature Review

2.1 Partial Differential Equations

Partial Differential Equations (PDEs) are differential equations with more than one independent variable, whereas Ordinary Differential Equations (ODEs) only have one independent variable. Common notation for ODEs is $\frac{d}{dx}$ or f'. There is a difference in notation for PDEs and this difference in notation is very important. Notation for PDEs is $\frac{\partial}{\partial x}$ or f_x. As there are more than one independent variable in PDEs, there will be a partial derivative for each independent variable. There will also be more than one possibility for the order in which we take the first and second derivative. For example, if we have independent variables x and y we could differentiate the function with respect to x and then with respect to y. The notation for this would be $\frac{\partial^2}{\partial x \partial y}$ or f_{xy}.

PDEs are normally harder to solve than ODEs and generally need computer algorithms in order to be solved. The general form of a second order partial differential equation in 2D is as follows [3],

$$a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial x \partial y} + c\frac{\partial^2 u}{\partial y^2} + d\frac{\partial u}{\partial x} + e\frac{\partial u}{\partial y} + fu + g = 0 \tag{2.1}$$

where a,b,c,d,e,f and g may be functions of the independent spatial variables x and y and, for non-linear problems, of the dependant variable u. There are three classifications for equation (2.1). Equation (2.1) is elliptic when $b^2 - 4ac < 0$, parabolic when $b^2 - 4ac = 0$ and hyperbolic when $b^2 - 4ac > 0$.

The thin film flow model is a system of non-linear parabolic partial differential equations, therefore this report will only consider parabolic PDEs.

2.1.1 Thin Film Flow Model

An example of a non-linear parabolic PDE is the thin film flow model. Thin film flow, when solved, provides the shape of a free surface over a topography and the pressure at each point of the free surface.

The thickness of the free surface must be small in comparison to the length of the domain i.e. $\epsilon = H_0/L_0$ must be small. Thin film flow is derived using the Navier-Stokes and continuity equations (see [2] for details). The derivation shown in [2] is as follows:

$$\frac{\partial^2 u}{\partial z^2} = \frac{\partial p}{\partial x} - 2, \tag{2.2}$$

$$\frac{\partial^2 v}{\partial z^2} = \frac{\partial p}{\partial y}, \tag{2.3}$$

$$\frac{\partial p}{\partial z} = -2\epsilon \cot\theta \tag{2.4}$$

where (x, z, y) are directional axes, p represents the pressure and (u, v) represents the fluid velocity.
We can solve these equations by assuming the no-slip condition $(u, v) = (0, 0)$ on the substrate $z = s(x)$ where s is the bed topography, and zero tangential stress at the film surface:

$$\frac{\partial u}{\partial z} = \frac{\partial v}{\partial z} = 0 \quad \text{at} \quad z = h + s \tag{2.5}$$

Integrating equations (2.2) and (2.3) twice with respect to z over the interval $z \in [s, h + s]$ yields

$$u = (\frac{\partial p}{\partial x} - 2)(z - s)(\tfrac{1}{2}(z - s) - h), \tag{2.6}$$

$$v = (\frac{\partial p}{\partial y})(z - s)(\tfrac{1}{2}(z - s) - h) \tag{2.7}$$

Mass, Q, is conserved throughout the domain, this means that

$$\nabla \cdot Q = 0, \tag{2.8}$$

where $Q = \int_s^{h+s} (u, v)^T dz$. Integrating equations (2.6) and (2.7) will give us Q, this can then be substituted into (2.8) to provide us with the 2D thin film flow model:

$$\frac{\partial}{\partial x}\left[\frac{h^3}{3}\left(\frac{\partial p}{\partial x} - 2\right)\right] + \frac{\partial}{\partial y}\left[\frac{h^3}{3}\left(\frac{\partial p}{\partial y}\right)\right] = 0 \tag{2.9}$$

where h represents the film thickness and p represents the pressure. The pressure can be obtained by integrating equation (2.4) with respect to z, where the constant of integration is determined by

$$-\frac{\epsilon^3}{Ca}\nabla^2(h + s) \quad \text{on } z = h + s, \tag{2.10}$$

therefore,

$$p = -\frac{\epsilon^3}{Ca}\nabla^2(h + s) + 2\epsilon(h + s - z)\cot\theta \tag{2.11}$$

The z-dependence in equation (2.11) due to the $2\epsilon z \cot\theta$ term does not have any influence on the film thickness h since its partial derivative with respect to both x and y is zero. Ca is the capillary number of the fluid,

$$Ca = \frac{\mu U_0}{\sigma} \tag{2.12}$$

where μ is viscosity, U_0 is the characteristic velocity and σ is the surface tension. Equation (2.11) can be simplified (see [2] for more details) to

$$p = -6\Delta(h+s) + 2\sqrt[3]{6}N(h+s), \tag{2.13}$$

where,

$$N = Ca^{\frac{1}{3}} \cot\alpha \tag{2.14}$$

where α is the angle the surface is inclined at.
In this report, the one-dimensional thin film flow model with $\alpha = 90°$ will be studied. The thin film flow for one dimension is as follows,

$$\frac{\partial}{\partial x}\left[\frac{h^3}{3}\left(\frac{\partial p}{\partial x} - 2\right)\right] = 0, \tag{2.15}$$

$$p = -6\frac{\partial^2}{\partial x^2}(h+s). \tag{2.16}$$

Finally, to complete the model of thin film flow, a topography, s, needs to be defined. A general topography for thin film flow is as follows,

$$s(x) = D[1 + \frac{1}{\pi}(-\tan^{-1}(\frac{x}{\delta}) + \tan^{-1}(\frac{x-w}{\delta}))], \tag{2.17}$$

where x is a Cartesian coordinate, D is the trench depth, w is the trench width, and δ is the steepness of the trench sides.

2.2 Introduction to Spatial Discretisation

In this section we discuss a method of spatial discretisation, the finite difference method (FDM). Spatial discretisation methods require a finite number of points. This is normally achieved by partitioning a spatial domain into a finite number of points. There are several spatial discretisation methods that can be used but the finite difference method (FDM) is the only method used in this report.

2.2.1 Finite Difference Method

Partial differential equations and ordinary differential equations do not always have solutions that are easy to find by analytic methods. In most cases an approximation is good enough. The finite difference method can approximate partial differential equations and ordinary differential equations. Finite difference methods approximate solutions to PDEs and ODEs by partitioning a spatial domain into a set of points. Uniformly spaced points are considered in this report, although variable spacing can be used. The spatial domain of the model $x \in [X_1, X_2]$ is approximated as a set of $N+1$ equally spaced points, $\{x_1, x_2, x_3, \ldots, x_{N+1}\}$, where

$$x_i = X_1 + (i-1)\Delta x, 1 \leq i \leq N+1, \tag{2.18}$$

$$\Delta x = \frac{X_2 - X_1}{N}, \tag{2.19}$$

where Δx is the mesh size. The functions can be approximated over this mesh at each point.

Taylor series expansions are used to formulate an approximation to derivative at each point, using the neighbouring points.
The Taylor series expansion of $f(x+\Delta x)$ is as follows:

$$f(x+\Delta x) = f(x) + \frac{f'(x)}{1!}\Delta x + \frac{f^{(2)}(x)}{2!}\Delta x^2 + \ldots + \frac{f^{(n)}(x)}{n!}\Delta x^n + \ldots \tag{2.20}$$

and for $f(x-\Delta x)$:

$$f(x-\Delta x) = f(x) - \frac{f'(x)}{1!}\Delta x + \frac{f^{(2)}(x)}{2!}\Delta x^2 - \ldots + \frac{f^{(n)}(x)}{n!}(-\Delta x)^n + \ldots \tag{2.21}$$

where $n!$ is the factorial of n. The forward difference approximation of the first derivative of f is derived from equation (2.20) as:

$$f'(x) \simeq \frac{f(x+\Delta x) - f(x)}{\Delta x} \tag{2.22}$$

As this method is only an approximation, a measurement of the accuracy is needed. This method provides a truncation error that is used to describe the accuracy of these expressions. For the first derivative,

$$\frac{f(x+\Delta x) - f(x)}{\Delta x} = f'(x) + \frac{f^{(2)}(x)}{2}\Delta x + \ldots \tag{2.23}$$

which gives,

$$\frac{f(x+\Delta x) - f(x)}{\Delta x} = f'(x) + O(\Delta x) \tag{2.24}$$

The $O(\Delta x)$ symbol is used to demonstrate the effect of the mesh spacing on the truncation error. The error of the approximation is proportional to Δx and tends to zero as $\Delta x \to 0$. Similarly, the backward difference approximation of the first derivative is derived from equation (2.21) as:

$$\frac{f(x) - f(x-\Delta x)}{\Delta x} = f'(x) + O(\Delta x) \tag{2.25}$$

Both approximations of f' are first-order accurate. Subtracting equations (2.20) and (2.21) obtains the central difference approximation as:

$$\frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} = f'(x) + O(\Delta x^2) \tag{2.26}$$

This approximation of f' is second-order accurate. We can also derive an approximation of the second derivative, f'', by adding equations (2.20) and (2.21), which gives

$$\frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{\Delta x^2} = f''(x) + O(\Delta x^2) \tag{2.27}$$

This is called the central difference approximation for the second derivative, whereas equation (2.9) is the central difference approximation for the first derivative. We can apply these approximations to each point, x_i, on our spatial grid. Equation (2.9) can be written as follows,

$$\frac{\partial f}{\partial x}|_i \simeq \frac{f_{i+1} - f_{i-1}}{2\Delta x} + O(\Delta x^2) \tag{2.28}$$

and the central difference equation for the second derivative, equation (2.26), can be written as follows,

$$\frac{\partial^2 f}{\partial x^2}|_i \simeq \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2} + O(\Delta x^2) \tag{2.29}$$

Central difference approximations are used in this report to approximate solutions to the thin film flow model. The thin film flow model has no analytic solution in general, therefore we apply FDM schemes to the model to gain numerical approximations to the system of non-linear parabolic partial differential equations. This method reduces the parabolic equations to an initial value system of nonlinear ordinary differential equations.

2.3 Nonlinear Systems

A nonlinear equation obtained from the discretisation of a nonlinear PDE can be written as follows:

$$\begin{aligned} f_1(x_1, x_2, \ldots, x_n) &= 0 \\ f_2(x_1, x_2, \ldots, x_n) &= 0 \\ f_3(x_1, x_2, \ldots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \ldots, x_n) &= 0 \end{aligned} \tag{2.30}$$

Vector notation is generally used to denote a system of equations. The notation used in this report to show the above system of equations is as below:

$$F(x) = 0, \tag{2.31}$$

where,

$$x = [x_1, x_2, \ldots, x_n]^T \tag{2.32}$$

2.3.1 Newton's Method

Newton's Method is a very popular method of solving nonlinear equations and systems of equations. Newton's method is an iterative approach to approximating the solution to an equation, with the use of a Jacobian matrix. Newton's method is normally preferred to other methods as it converges quadratically. A description of the workings of the method is given in this section.
For a single equation, f, Newton's method is as follows,

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{2.33}$$

When using Newton's method, a tolerance is normally used. This sets a stopping point for the algorithm. For a system of equations, F, Newton's method is slightly different. The method is as follows,

$$x_{k+1} = x_k - (\frac{\partial F}{\partial x}(x_k))^{-1} F(x_k) \tag{2.34}$$

Both methods are similar, the main difference being equation (2.34) uses a Jacobian matrix, $\frac{\partial F}{\partial x}(x_k)$. The Jacobian matrix is an n x n matrix of partial derivatives where

$$J_{ij} = \frac{\partial F_i}{\partial x_j}, i = 1, 2, \ldots, n, j = 1, 2, \ldots, n.$$

We can write the Jacobian matrix as:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \frac{\partial f_n}{\partial x_3} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{2.35}$$

To solve equation (2.34) we can split each iteration into two two steps as follows:

- We find a vector δ by solving the linear system

  $$J(x_k)\delta = -F(x_k) \tag{2.36}$$

  MATLAB has a standard linear solve that can be used to solve system (2.36).

- After finding δ we can use the equation

  $$x_{k+1} = x_k + \delta. \tag{2.37}$$

  Equation (2.37) updates the vector x_k at each iteration. Newton's method will keep iterating until a failure or convergence criteria is met. For example, a failure criteria could be the maximum number of iterations being reach and convergence criteria is normally when a solution reaches a value within a specified tolerance.
  Newton's method will only converge with a sufficiently good initial guess. Initial guesses are discussed in Section (2.4).

2.4 Initial Guess

Newton's method relies heavily on the initial guess. We can think of solutions as having contours around them showing the local gradient of the nonlinear function. If Newton's method does not start within these contours then it may not converge. Different solutions can have different distributions of contours meaning the initial guess may need to be closer for some solutions than others. By default we choose $h = 1, p = 0$ which is physically reasonable. There are several methods for producing an improved initial guess before applying Newton's method, two of which are described in this section.

2.4.1 Homotopy

Homotopy continuation is a method of finding a good initial guess. To understand how homotopy works we must look at the mathematics behind it. Firstly, we can define a new equation as follows:

$$H(t, U) = tF(U) + (1 - t)(F(U) - F(U_0)) \tag{2.38}$$

where $t \in [0, 1]$ and $F(U) = 0$. $H(t, U)$ is defined such that when $t = 0$, $H(t, U) = F(U) - F(U_0)$ and when $t = 1$, $H(t, U) = F(U)$.
We now solve

$$H(t, U(t)) = 0 \tag{2.39}$$

for $U(t)$ with initial condition $U(0) = U_0$.
As $H(t, U) = 0$ is solved along the continuation path, the total variation of H with t is zero,

$$\frac{dH}{dt} = \frac{\partial H}{\partial t} + \frac{\partial H}{\partial U}\frac{\partial U}{\partial t} = 0 \tag{2.40}$$

which defines a system of differential equations for $U(t)$.
We can rewrite equation (2.38) as:

$$H(t, U) = F(U) + (t - 1)F(U_0) \tag{2.41}$$

This provides us with the following partial derivatives:

$$\frac{\partial H}{\partial U} = \frac{\partial F}{\partial U} = J(U) \tag{2.42}$$

$$\frac{\partial H}{\partial t} = F(U_0) \tag{2.43}$$

From equation (2.40):

$$F(U_0) + J(U)\frac{\partial U}{\partial t} = 0 \tag{2.44}$$

therefore,

$$\frac{\partial U}{\partial t} = -J^{-1}(U)F(U_0) \tag{2.45}$$

We can derive a two-step iterative algorithm as follows:

$$J(U_k)\delta = -F(U_0) \tag{2.46}$$

$$U_{k+1} = U_k + \Delta t\delta \tag{2.47}$$

For a fixed number of steps of step size Δt on the interval $t \in [0, 1]$.
$U(t = 1)$ approximates the solution to $F(U) = 0$. We can then use this approximation as the initial guess to Newton's method and this is the method that is used the most in my results.

2.4.2 Gradient Descent

Gradient descent can be used as a solution algorithm but is normally used as a minimisation process. Convergence of gradient descent is only linear therefore it is a lot slower than Newton's method. The benefit of gradient descent is that it is more robust than Newton's method and that there is no linear system to solve at each iteration. The derivation of gradient descent is not given is this report. We start with an initial condition, U_0, and we update U at each step as follows:

$$d = -2J^T(U)F(U) \tag{2.48}$$

$$U = U + \lambda d$$

where d is the direction towards the solution and λ is how far away the solution is. We must find λ using the line-search algorithm. A line-search parameter, λ, is set equal to 1. We then test the following:

$$|F(x_k + \lambda d)| < |F(x_k)| \tag{2.49}$$

If equation (2.49) fails then we repeat with $\lambda = \frac{\lambda}{2}$. We must also limit the number of line-search iterations to ensure a maximum number of rejections.
Gradient descent is often used as a method of getting a good initial guess for Newton's method. Although gradient descent can be used to give an approximate solution, as it is only linearly convergent, Newton's method is preferred to find the final solution.

2.5 Deflation

Exploring the uniqueness of thin film flow is the goal of this research. The problem with using the methods explained in the previous sections is that Newton's method will nearly always find the same solution. Deflation is a method that eliminates solutions that have previously been found [1]. When a first solution, U^*, is found we can define a new nonlinear system as follows:

$$G(U) = \frac{F(U)}{||U - U^*||^p} + \alpha F(U) \tag{2.50}$$

This produces a system of equations such that as $U \to U^*$, $G(U) \to \infty$. Newton's method will therefore not treat U^* as a solution, but every other solution of $F(U)$

will also be a solution of $G(U)$.
Parameters p and α can be selected to yield slightly different variations of G. The conditions for selecting p and α are, $p \in \mathbb{R} \geq 1$ and $\alpha \geq 0$. Selecting a larger p will result in a larger region around U^* with a large gradient, whereas α will just shift the equation which may help find solutions.

Chapter 3

Computational Results

In this chapter we will use MATLAB to look for a second solution of thin film flow using the methods described in Chapter 2. MATLAB is very useful when maths based programming is needed as there are a lot of predefined functions that help solve systems of equations.
An efficient implementation for solutions of one dimensional thin film flow already existed before this research was conducted. This implementation did not include any of the initial guess finding methods or deflation.
We computed solutions for a range of different topographies. Various extra examples are shown in sections (3.1 - 3.6).

3.1 Thin Film Flow Model

In this section we introduce some results of thin film flow using Newton's method with an initial guess we specify. Varying topographies are needed to give a variety of different results.
Unless otherwise stated we assume a vertically aligned plane ($\alpha = 90°$) and take initial guess $h = 1, p = 0$ in the whole domain.
Below is our first example, Figure 3.1 is thin film flow applied over 81 points

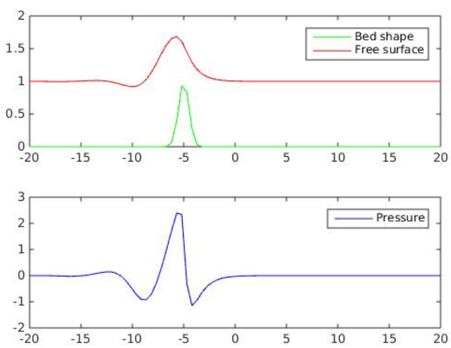

Figure 3.1: Gaussian hill

The topography is defined as follows:

$$s(x) = e^{(-2(x+5)^2)} \tag{3.1}$$

When approximating a function over a discrete domain, the number of points we use can also have an effect on the results. For example, in figure 3.2 below, we use 161 points.

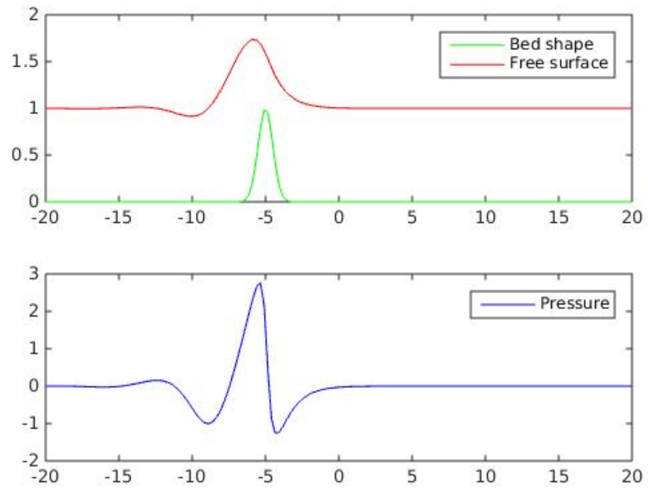

Figure 3.2: Gaussian hill - 161 points

Figure 3.2 has a much smoother bed shape, free surface and pressure, when compared to figure 3.1, this is due to doubling the points. The next page has some more examples of a free surface over different topographies, each using 81 points.

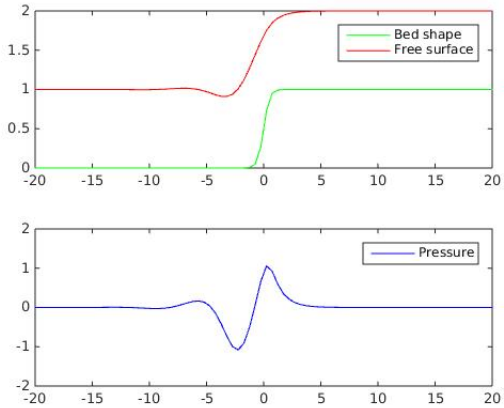

Figure 3.3: Step up: $s(x) = -\dfrac{1}{2}(\tanh(-2x) - 1)$

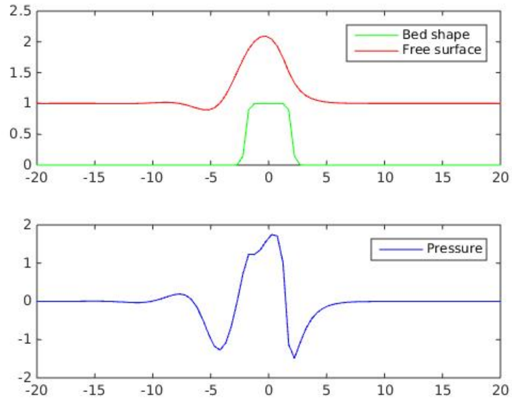

Figure 3.4: Sharp ridge: $s(x) = \dfrac{1}{2}(\tanh(-4(x-2)) - \tanh(-4(x+2)))$

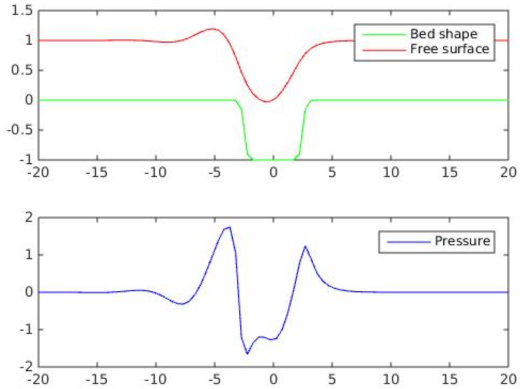

Figure 3.5: Sharp trench: $s(x) = -\dfrac{1}{2}(\tanh(-4(x-2.5)) - \tanh(-4(x+2.5)))$

3.2 Deflation

In this section we apply the deflation method from section 2.5. Firstly, we solve the thin film flow as we did in the previous section to obtain a solution. We then try to look for another solution using the deflation method with different values for deflation parameters p and α.
When we apply deflation to the solution we got in figure 3.5, we get the following:

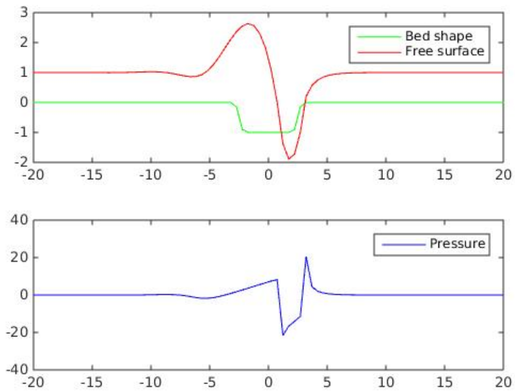

Figure 3.6: At $p = 1.1$ and $\alpha = 0.1$

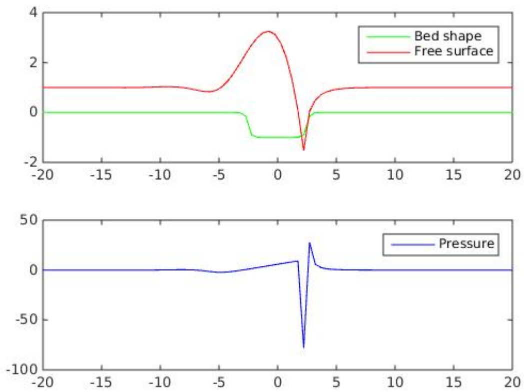

Figure 3.7: At $p = 1.2$ and $\alpha = 0.0$

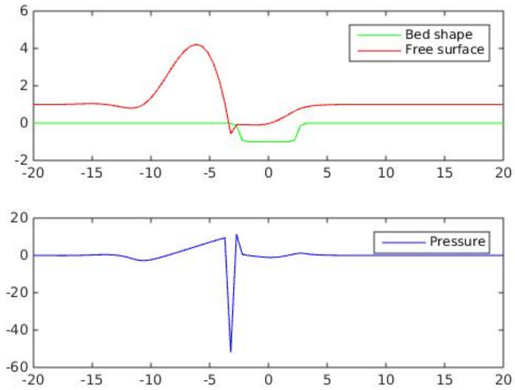

Figure 3.8: At $p = 1.2$ and $\alpha = 0.3$

These solutions are not physical solutions as the free surface cannot fall below the bed. Non-physical solutions are also interesting but to determine whether they are actual solutions we must increase the number of points used.
At 161 points we still obtain the solution shown in figure 3.8. When we double the points again to 321, the only solution that is found is the first physical solution.
Approximating discrete solutions over a finite grid can cause solutions to appear when they are not actual solutions of the continuous PDE system. Increasing the number of grid points is a way to test if these solutions are real or an artefact of the grid resolution.

3.3 Homotopy

In this section we look at the results produced when homotopy continuation is used to gain an initial guess. We use this initial guess to start the deflation method. MATLAB comes with several ODE solvers that approximate solutions to differential equations. We will use one of these to approximate a solution to equation (2.45).

Differential equation solvers in MATLAB can also take a set of options as one of its arguments. We used two options when applying the solver, which are 'RelTol' and 'OutputFcn'. RelTol is used to set a tolerance for when the ODE solver can stop, OutputFcn displays a graph as the ODE is solved at each point. An example of the graph output is given in figure (3.9).

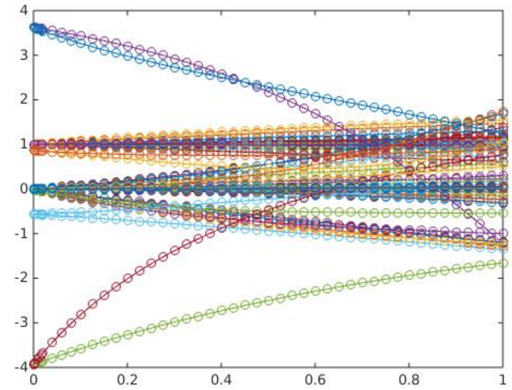

Figure 3.9: ODE solver

Deflation can push Newton's method away from other possible solutions, continuation is a way round this. Using continuation to find an initial guess for Newton's method means it may converge to a solution not previously found. Below are some solutions found using continuation.

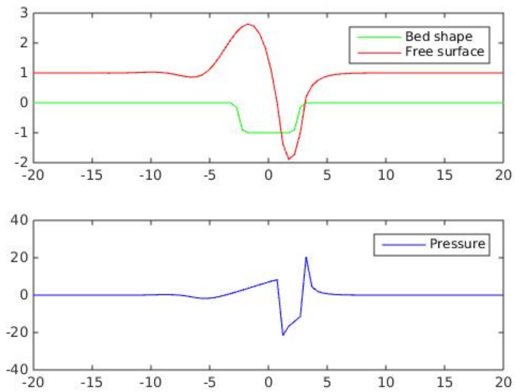

Figure 3.10: p = 1.1 and alpha = 0.1

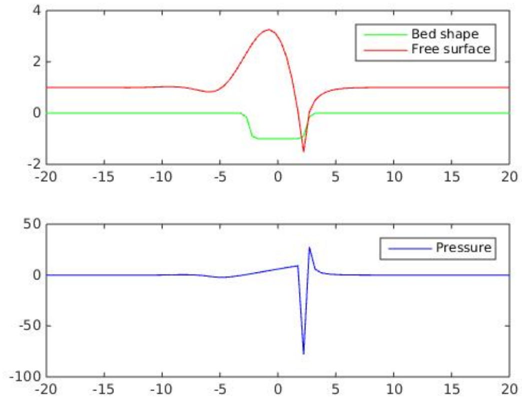

Figure 3.11: p = 1.2 and alpha = 0.0

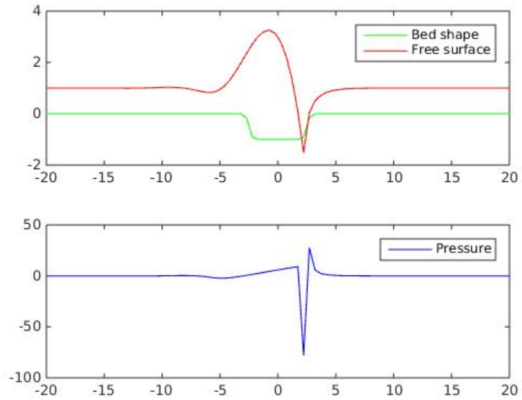

Figure 3.12: p = 1.2 and alpha = 0.0

All solutions with this bed topography and using continuation are the same solutions as those found in the previous section.
Several different topographies have been tested using continuation to look for second solutions. Examples of these tests are given in section 3.6.

3.4 Interpolation

When we find solutions using a lower number of grid points, we want to try and replicate the solution with a higher number of grid points. Interpolation is a method we can use for the solutions we find to produce a better initial guess for a higher number of grid points.
If we have a solution x that is a column vector, we can interpolate this solution onto a new solution x^* by placing every element $x(i)$ in to $x^*(2i-1)$. For example, $x^*(1) = x(1), x^*(3) = x(2), \dots, x^*(2n) = x(n)$. We can then fill in the missing points by averaging the points before and after the point we want to fill in. The result of x^*

can then be used as an initial guess for a solution with higher grid points.

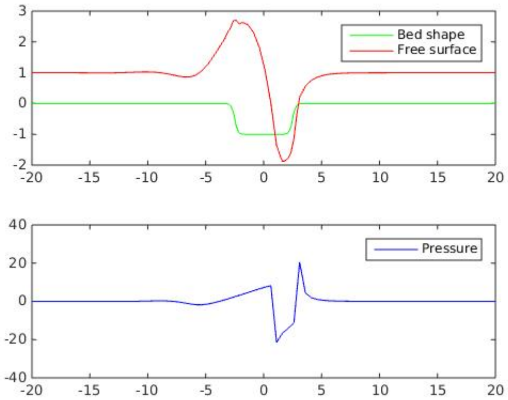

Figure 3.13: Interpolation

Figure (3.13) is the interpolation of figure (3.10) onto a higher number of grid points. Interpolation gives a very accurate approximation to the higher grid solution, therefore it maximises the chance of that solution being found again.

We applied interpolation to each solution we found in the examples above. Newton's method did not find any second solutions with interpolation applied.

3.5 Conclusion

In this report we have explained several methods of trying to find a second solution to the thin film flow model using Newton's method. We then applied these methods using computational simulations with various topographies. If we found a solution, which we did on several occasions, we then tried to repeat the solution with a higher number of grid points.

We started with 81 points originally then doubled the number of points if a solution was found. No second solution was reproducible over 161 points. If we had found a second solution then we could definitively say that thin film flow does not produce unique solutions, whereas there is no known way to prove that solutions are unique. We have exhausted all the possible ways of trying to find a second solution, as a result we can say it is likely, but not definite, that solutions of thin film flow are unique.

3.6 Additional Examples

In this section we will give additional examples of different topographies we tried. This will show the number of different examples we used to try find a second solution and come to our conclusion.

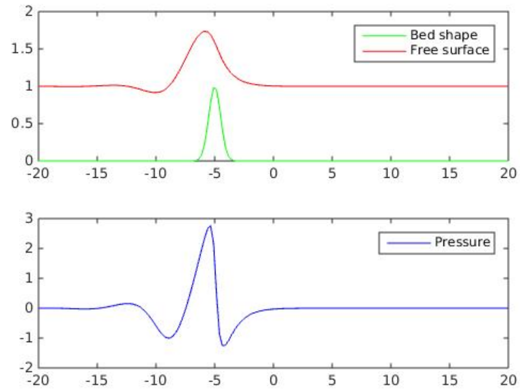

Figure 3.14: Gaussian Hill - $s(x) = e^{(-2(x+5)^2)}$

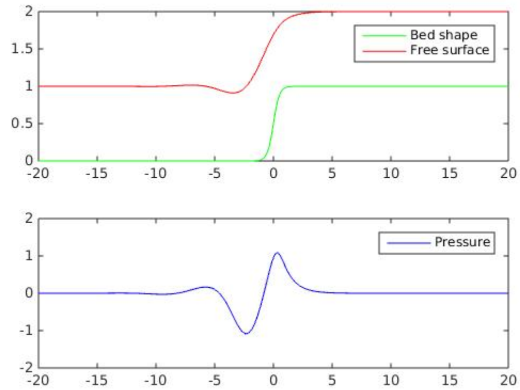

Figure 3.15: Step Up - $s(x) = -\frac{1}{2}(\tanh(-2x) - 1)$

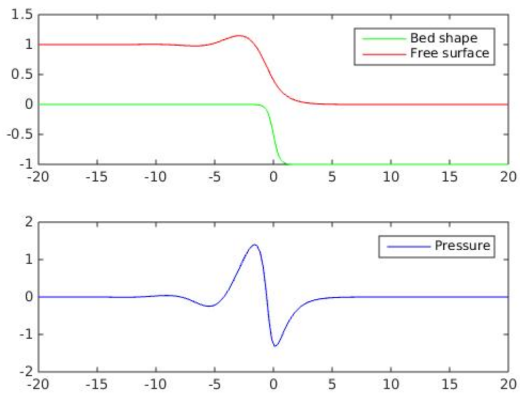

Figure 3.16: Step Down - $s(x) = \frac{1}{2}(\tanh(-2x) - 1)$

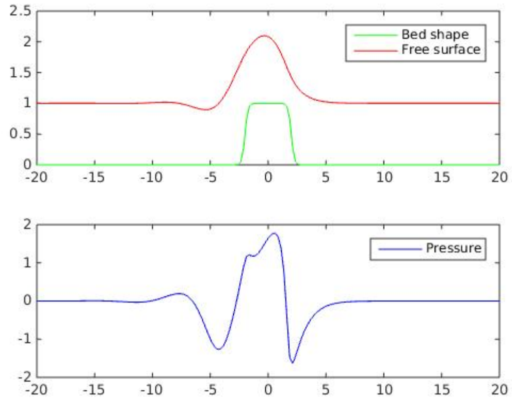

Figure 3.17: Ridge - $s(x) = \frac{1}{2}(\tanh(-4(x-2)) - \tanh(-4(x+2)))$

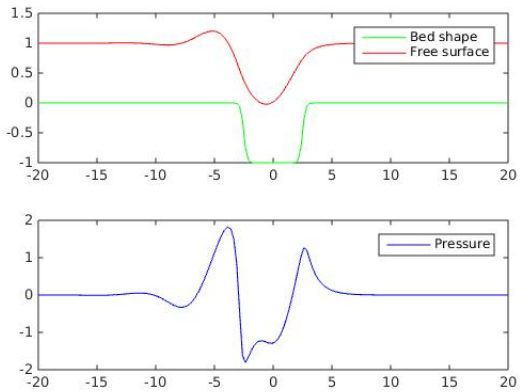

Figure 3.18: Trench - $s(x) = -\frac{1}{2}(\tanh(-4(x-2.5)) - \tanh(-4(x+2.5)))$

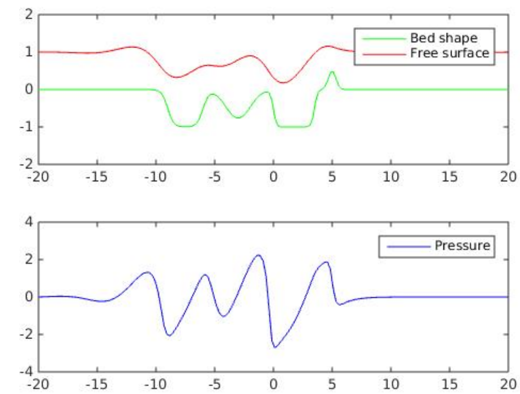

Figure 3.19: $s(x) = -\frac{1}{2}(\tanh(-4(x-3.5)) - \tanh(-4(x)) + \tanh(-1(x+2)) - \tanh(-1(x+4)) + \tanh(-2(x+6)) - \tanh(-2(x+9)) - e^{(-5(x-5)^2)})$

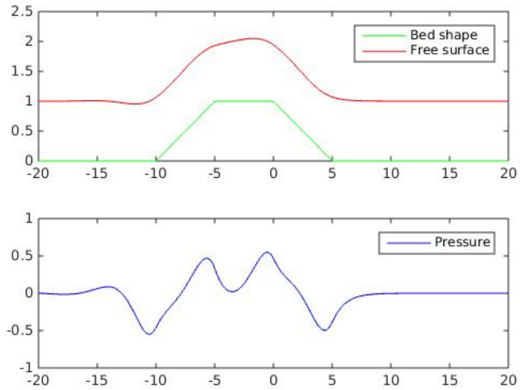

Figure 3.20: Straight line equations are used to make up the topography. This means the topography is fully resolved at a small number of grid points.

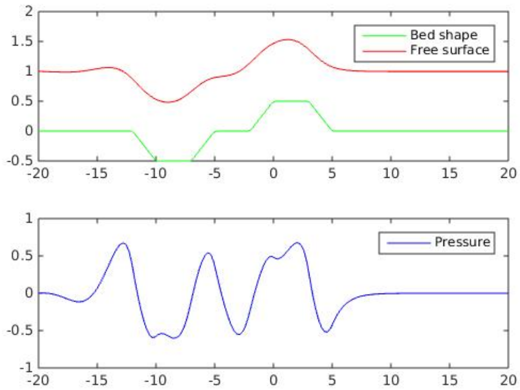

Figure 3.21: Straight line equations are used to make up the topography. This means the topography is fully resolved at a small number of grid points.

Bibliography

[1] P E Farrell, A Birkisson, and S W Funke. *Deflation techniques for finding distinct solutions of nonlinear partial differential equations*. Tech. rep. University of Oxford, 2014.

[2] P H Gaskell et al. "Gravity-driven flow of continuous thin liquid films on non-porous substrates with topography". In: *Journal of Fluid Mechanics* (2004).

[3] Walter A Strauss. *Partial differential equations: An introduction*. John Wiley & Sons Ltd, 2008.