Electronic Transactions on Numerical Analysis.
Volume 60, pp. 20–39, 2024.
Copyright © 2024, Kent State University.
ISSN 1068–9613.
DOI: 10.1553/etna_vol60s20

ETNA
Kent State University and
Johann Radon Institute (RICAM)

MULTI-SCALE SPECTRAL METHODS FOR BOUNDED RADIALLY SYMMETRIC CAPILLARY SURFACES[*]

JONAS HAUG[†] AND RAY TREINEN[†]

Abstract. We consider radially symmetric capillary surfaces that are described by bounded generating curves. We use the arc-length representation of the differential equations for these surfaces to allow for vertical points and inflection points along the generating curve. These considerations admit capillary tubes, sessile drops, and fluids in annular tubes as well as other examples. We present a multi-scale pseudo-spectral method for approximating solutions of the associated boundary value problems based on interpolation by Chebyshev polynomials. The multi-scale approach is based on a domain decomposition with adaptive refinements within each subdomain.

Key words. capillarity, spectral methods

AMS subject classifications. 76B45, 65N35, 35Q35, 34B60

1. Introduction.

In a recent work, the second author introduced an adaptive Chebyshev spectral method for radially symmetric capillary surfaces. These radially symmetric capillary surfaces give rise to boundary value problems for systems of nonlinear ODEs. Treinen [15] found approximations of the solutions to these nonlinear systems by using Newton's method to generate a sequence of linear problems, and the corresponding sequence of solutions, when convergent, approach the solution of the nonlinear problem. The solutions of the linear problems were computed with Chebyshev spectral methods. There were tests put in place to ensure that the Newton steps were converging to a solution of the nonlinear problem within a prescribed tolerance, and if that was not achieved, then the resolution of the underlying approximation scheme was adaptively increased, repeating until convergence was achieved within the requested tolerance. Overall the performance was fast and robust. However, in that work some cases were identified where the adaptive step required a rather ungainly number of points, and the performance suffered. There were also instances where the algorithm did not converge. These cases are the subject of the current study. We have implemented multi-scale approaches for these problems, where the problematic regions are isolated and treated with separate adaptive approaches.

We use Matlab with Chebfun [4] to facilitate some aspects of our work, though we will not be using all of the automation implemented in Chebfun. We do use Chebfun to generate our differentiation matrices and for plotting our generating curves with barycentric interpolation.

We will focus on two prototype problems:

- **P1** simply connected interfaces that are the image of a disk, and
- **P2** doubly connected interfaces that are the image of an annulus.

Within these two prototype problems we investigate subsets of problems that either have been computationally expensive to solve or where convergence fails altogether. Before we discuss our multi-scale approach for specific problems, we give a brief background on the underlying mathematical problems and summarize the algorithm from [15].

We have released Matlab implementations of these algorithms under an open source license. These files are available in a GitHub repository at
https://github.com/raytreinen.

[*] Received June 20, 2023. Accepted October 24, 2023. Published online on January 26, 2024. Recommended by Tom DeLillo.

[†] Department of Mathematics, Texas State University, 601 University Dr., San Marcos, TX 78666 ({qcd5,rt30}@txstate.edu).

2. Mathematical framework for the prototype problems. The mean curvature of a capillary surface is proportional to its height u, where the height is measured over some fixed reference level. We will restrict our attention to embedded solutions of the mean curvature equation

$$2H = \kappa u,$$

where H is the mean curvature of the surface, $\kappa = \rho g/\sigma > 0$ is the capillary constant with ρ defined to be the difference in the densities of the fluids, g the gravitational constant, and σ the surface tension. When the height of the interface u is a graph over the base domain, the mean curvature H is given by

$$2H = \nabla \cdot \frac{\nabla u}{\sqrt{1+|\nabla u|^2}}.$$

If the interface is not a graph over the base domain, then H can be defined locally in the sense that the height of the interface forms a surface that can be interpreted as a manifold. See Finn [7]. This equation is sometimes known as the Young-Laplace equation or the capillary equation.

A radial symmetric capillary surface can be described by a system of three nonlinear ordinary differential equations, parameterized by the arc-length s:

$$\frac{dr}{ds} = \cos\psi, \tag{2.1}$$

$$\frac{du}{ds} = \sin\psi, \tag{2.2}$$

$$\frac{d\psi}{ds} = \kappa u - \frac{\sin\psi}{r}, \tag{2.3}$$

where r is the radius, u is the height of the interface, and the inclination angle ψ is measured from the corresponding generating curve that is described by $(r(s), u(s))$. The associated boundary values will come from the natural boundary conditions for each problem and will correspond to prescribing the inclination angle(s) at one or more radii; the details are described in what follows.

We defer to Finn [7] and Treinen [15] for background and further references. However, there are some perspectives and examples that give some insight into the surfaces that we compute in this paper. The computationally challenging cases we consider here have ψ prescribed on the boundary with $|\psi| > \pi/2$. Figure 2.1 displays a section of a radially symmetric sessile drop, and this physical configuration can be represented by an extension of the solution curve for a capillary surface in a tube. Finn and many others took the approach of studying (2.1)–(2.3) as an initial value problem with the initial height given by u_0, where $r = 0$ at $s = 0$. From the regularity results for these elliptic problems, one is able to deduce that $\psi = 0$ there. It follows that there is a one-parameter family of solutions of these equations parameterized by u_0, and one finds the unique u_0-value that attains a prescribed indication angle ψ_b at a prescribed radius b. This process results in a generating curve for a radially symmetric capillary surface in a tube, and this is also a big theme in [7].

While we do not use this approach in the current work, it could give insight to the reader. In [7, Section 3.1], Finn describes the correspondence principle that there is a one-to-one correspondence between sessile drops resting on a plate and capillary surfaces in a tube. For general sessile drops, this requires a proof that one may extend solutions of (2.1)–(2.3) past $\psi = \pi/2$. Once the solution has been extended past the vertical point, one stops at a

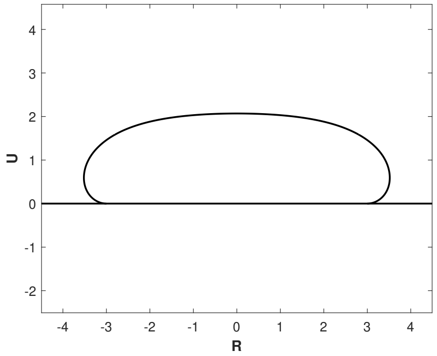

FIG. 2.1. *A radially symmetric sessile drop rests on a plate, and a section of the configuration is shown. The fluid interface intersects with the plate at radius $b = 3$, and the contact angle γ is π, giving an extremely hydrophobic drop.*

prescribed inclination angle ψ_b that corresponds to the physical contact angle. With this in hand, the correspondence with the physical configuration is achieved by the transformation $v = -u - \lambda/\kappa$, where the height v describes a sessile drop given by

$$\nabla \cdot \frac{\nabla v}{\sqrt{1 + |\nabla v|^2}} = \kappa v + \lambda,$$

with the Lagrange multiplier λ for satisfying the physical volume constraint. We have illustrated this in Figure 2.2.

When one considers (2.1)–(2.3), it is also possible to continue the solutions to these initial value problems for any arc-length $s > 0$, and the solutions are known to be analytic. They do, however, have self-intersections in this global setting of arbitrary $s > 0$. A classification of all of these global solutions and an exploration of the asymptotic behavior appears in the work of Bagley and Treinen [2]. For physical applications one is restricted to selecting connected pieces of these curves that avoid these self-intersections. Later in this paper we will use this global perspective to explain why one encounters some of the difficult problems studied here; Figure 5.1 shows examples of this self-intersection.

2.1. The problem P1.

For radially symmetric and simply connected surfaces that are the image of a disk, we specify boundary conditions by the requirement that at some arc-length $\ell > 0$, the radius $r(\ell)$ attains a prescribed value $b > 0$ and the inclination angle $\psi(\ell)$ attains a prescribed value $\psi_b \in [-\pi, \pi]$. Figure 2.3 displays a typical example that appeared as a dashed curve in Figure 2.2. In our formulation the value of the arc-length ℓ is unknown, and we approach this unknown parameter by re-scaling the problem, defining $\tau = s/\ell$, or $s = \ell\tau$. Then we define

$$\begin{aligned} R(\tau) &:= r(\ell\tau) = r(s), \\ U(\tau) &:= u(\ell\tau) = u(s), \\ \Psi(\tau) &:= \psi(\ell\tau) = \psi(s). \end{aligned}$$

This scaling is suitable for the use of Chebyshev polynomials, as the natural domains of R, U, and Ψ are now $[-1, 1]$. Then, using the chain rule, from (2.1)–(2.3), we find

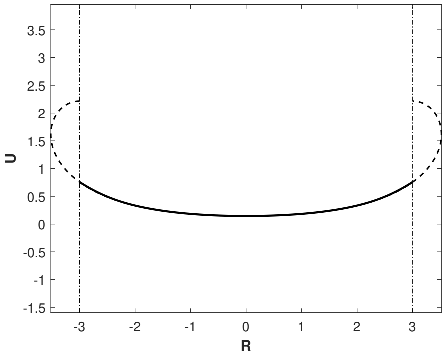

FIG. 2.2. *A capillary surface in a tube of radius $b = 3$ is shown in bold, and the dashed lines illustrate the extension of this solution up to the inclination angle $\psi_b = \pi$. The curves here are sections of the associated radially symmetric fluid interfaces.*

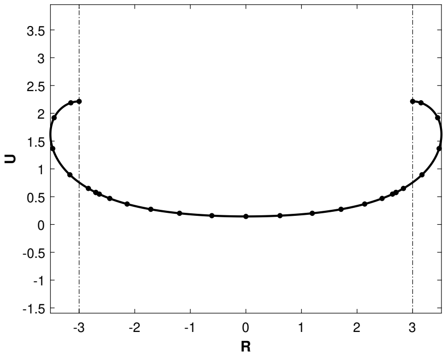

FIG. 2.3. *A capillary surface with radius $b = 3$ and inclination angle $\psi_b = \pi$. Here the dots along the interface are the Chebyshev gridpoints for three regions in our domain decomposition. In our figures we include vertical lines to indicate the radius (or radii) of interest.*

$$R'(\tau) - \ell \cos \Psi(\tau) = 0, \tag{2.4}$$

$$U'(\tau) - \ell \sin \Psi(\tau) = 0, \tag{2.5}$$

$$\Psi'(\tau) + \frac{\ell \sin \Psi(\tau)}{R(\tau)} - \kappa \ell U(\tau) = 0. \tag{2.6}$$

Setting the column vector $\mathbf{v} = [R\ U\ \ \Psi\ \ell]^T$, we can use (2.4)–(2.6) to define the nonlinear operator in the vector equation

$$\tilde{N}(\mathbf{v}) = \mathbf{0}. \tag{2.7}$$

We then use the boundary conditions

$$\begin{aligned} R(1) - b &= 0, & \Psi(1) - \psi_b &= 0 \\ R(-1) + b &= 0, & \Psi(-1) + \psi_b &= 0, \end{aligned} \tag{2.8}$$

so that we have some form of a two-point boundary value problem. Here we are prescribing the natural boundary conditions on a cylinder of radius b. See Figure 2.3. As we discussed previously, we also allow for the strictly parametric solution, where $\pi/2 < \psi_b$, and this then provides a model for the sessile drop wetting a disc of radius b. We append (2.7) with these boundary conditions (2.8) to form the system

$$N(\mathbf{v}) = \mathbf{0}.$$

The resulting solution $(R, U) = (r, u)$ is not a generating curve of the surface, but it is a section. The generating curve is given by $(R(\tau), U(\tau))$, for $0 \leq \tau \leq 1$, and the generating curve has total arc-length ℓ.

One final remark on the general nature of this problem is that (2.3) and (2.6) have singularities at $r = R = 0$. It is known that this singularity is removable, and the solution is analytic [7]. We will discuss our numerical approach for this singularity when we return to the results for this problem. For the time being, we present an alternative form of (2.6) that is suitable for reducing the numerical error from rounding, as exaggerated by dividing by a number close to zero. We multiply by R to get

$$R(\tau)\Psi'(\tau) + \ell \sin \Psi(\tau) - \kappa \ell R(\tau) U(\tau) = 0, \tag{2.9}$$

which we use when b is relatively small. When we use (2.9) in place of (2.6), we denote the changed N by N_1 (and F and L similarly below) when specifying that the difference is important, and we will refer to those objects without subscripts when no confusion is expected.

We will later approach this nonlinear problem with a Newton method, and we need to use the Fréchet derivative

$$F(\mathbf{v}) = \frac{dN}{d\mathbf{v}}(\mathbf{v}).$$

In our derivations we used the ideas presented in the nonlinear examples given by Aurentz and Trefethen [1], and that work also refers to Zeidler [16], which we have found to be a good place for further reading. Given that $\mathbf{v}$ has several components and since some of the computations in $F(\mathbf{v})$ involve derivatives with respect to τ, we introduce the differential operator

$$D = \frac{d}{d\tau},$$

which is applied in a block fashion to $\mathbf{v}$ so that $R'(\tau) = [D\ 0\ 0\ 0]\mathbf{v}$, for example. We also have need to use an operator version of function evaluation. We denote this operator by D^0_τ. For instance, we have $D^0_1 R = R(1)$ and $D^0_{-1}\Psi = \Psi(-1)$. Also, we employ an abuse of notation by writing R for the R-component of $\mathbf{v}$ and similarly for U, Ψ, and ℓ. With this in hand, we compute

$$F(\mathbf{v}) = \begin{bmatrix} D & 0 & \ell \sin \Psi & -\cos \Psi \\ 0 & D & -\ell \cos \Psi & -\sin \Psi \\ \frac{-\ell \sin \Psi}{R^2} & -\kappa \ell & D + \frac{\ell \cos \Psi}{R} & \frac{\sin \Psi}{R} - \kappa U \\ D^0_{-1} & 0 & 0 & 0 \\ D^0_1 & 0 & 0 & 0 \\ 0 & 0 & D^0_{-1} & 0 \\ 0 & 0 & D^0_1 & 0 \end{bmatrix} \begin{bmatrix} R \\ U \\ \Psi \\ \ell \end{bmatrix}. \tag{2.10}$$

We have need to solve linear systems based on the definition $F(\mathbf{v}) := L\mathbf{v}$. For F_1 and L_1, we merely change the third row to

$$\begin{bmatrix} D\Psi - \kappa \ell U & -\kappa \ell R & RD + \ell \cos \Psi & \sin \Psi - \kappa U R \end{bmatrix}. \tag{2.11}$$

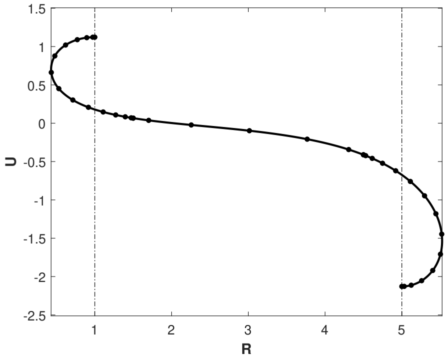

FIG. 2.4. *A generating curve for a radially symmetric annular capillary surface with radii $a = 1$ and $b = 5$ with inclination angles $\psi_a, \psi_b = -\pi$. The fluid is below or to the left of the curve. Restricting the region between the two vertical lines, this interface forms a symmetric capillary surface in an annular tube. The extended interface can be used as a component interface in more complicated physical examples. This configuration could be seen as a generating curve for a radially symmetric liquid bridge spanning the gap between two horizontal plates. In that usage, the contact with the lower plate is non-wetting, and the contact with the upper plate is wetting. The dots on the fluid interface are the locations of the Chebyshev grid points for the three regions in the domain decomposition used to generate this solution curve.*

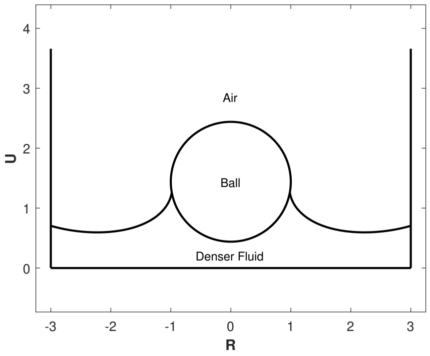

FIG. 2.5. *A ball floats at the fluid interface, and a section is shown of the radially symmetric configuration. Here the fluid-air interface is an annular capillary surface, and it is possible (or mathematically admissible) that there could be a vertical point on the solution curve near the ball, so the parametric solutions are required. The algorithm used to compute this configuration is based on a shooting method, which was the standard approach before the recent advances in using spectral methods. See McCuan and Treinen [10] for background on this problem.*

2.2. The problem P2.

If we replace the symmetric and simply connected surfaces that are the image of the disk considered in Section 2.1 with doubly connected interfaces that are the image of an annulus, then the scaling argument is preserved. There are differences though, as the arc-length $s = 0$ does not correspond to the radius being zero. We reuse (2.4)–(2.6) and (2.10), but we have boundary conditions

$$\begin{aligned} R(1) - b &= 0, & \Psi(1) - \psi_b &= 0, \\ R(-1) - a &= 0, & \Psi(-1) - \psi_a &= 0, \end{aligned}$$

where we introduce $a \in (0, b)$ and $\psi_a \in [-\pi, \pi]$. See Figure 2.4. This changes the definition of N, but not that of F. Notice that in this case $R \neq 0$, so there is no singularity. If

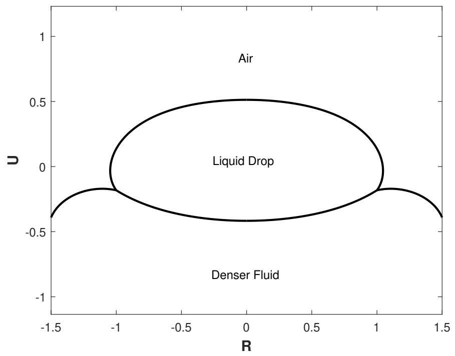

FIG. 2.6. *A radially symmetric liquid drop floats on the interface of a denser fluid, and a section of the configuration is shown. There are three component capillary surfaces used to build this configuration. Here, there are vertical points on the upper interface of the drop. It is conjectured that the annular component surface must be a graph over the base domain, but this has not yet been established. As with the floating ball example, the algorithm used to compute this is based on a shooting method for each interface. See Elcrat and Treinen [6] for details.*

$|\psi_a|, |\psi_b| \leq \pi/2$, then the resulting solution curves are graphs, and we have a so-called annular capillary surface in an annular tube, with the inner wall at the radius a and the outer wall at the radius b [5]. If this restriction of the inclination angles is removed, then the solution curve can pass through one or both of the "walls" [14]. In either case, these solution curves form generating curves with total arc-length 2ℓ. The applications of these surfaces are not only fluid surfaces in annular tubes but also multiple-component configurations. Figure 2.5 displays a ball floating on the surface of a liquid, and Figure 2.6 displays a liquid drop floating on the surface of a denser fluid. In both of these examples it is possible to have vertical points on the liquid interface, and so we include angles up to $|\psi_a|, |\psi_b| = \pi$ to allow for many applications like these.

3. The spectral method from [15]. We summarize the spectral method developed by Treinen [15], as it is the fundamental building block of our approach. We also point out that this approach has its roots in Trefethen's book [13], Driscoll and Hale [3], and Aurentz and Trefethen [1], and if the reader is not yet familiar with these works, he or she is strongly encouraged to read these first, perhaps starting with [1].

Here we present the basic ideas with the backdrop of $\mathbf{P1}$. The flowchart in Figure 3.1 gives the framework of the algorithm, and it applies to both $\mathbf{P1}$ and $\mathbf{P2}$. Here we will merely discuss the key points of this algorithm.

We will treat the nonlinearity of N with a Newton method. The core of the algorithm is to construct N and L, and then we solve the linear equation $L(\mathbf{v})\,d\mathbf{v} = -N(\mathbf{v})$ for $d\mathbf{v}$ to build the update $\mathbf{v}_{\texttt{new}} = \mathbf{v} + d\mathbf{v}$. This iterative process continues until some convergence criterion is met. This process needs an initial guess to begin, and if that initial guess is sufficiently close to the solution (and if some additional hypotheses hold), then the convergence is known to be quadratic. For discussions of the functional analysis in Banach spaces and the associated convergence, see Kantorovich and Akilov [9] as well as Ortega [12]; later Gragg and Tapia [8] gave optimal error bounds. For a discussion of the convergence rate in finite-dimensional problems, see Nocedal and Wright [11]. Here we do not verify the additional hypotheses needed in these results, as that analysis is outside the scope of this work.

We discuss the initial guesses for Newton's method when we consider specific problems below. However, we will not discuss the barriers constructed in [15], which were used to enforce convergence to physically correct configurations. These barriers are still used here.

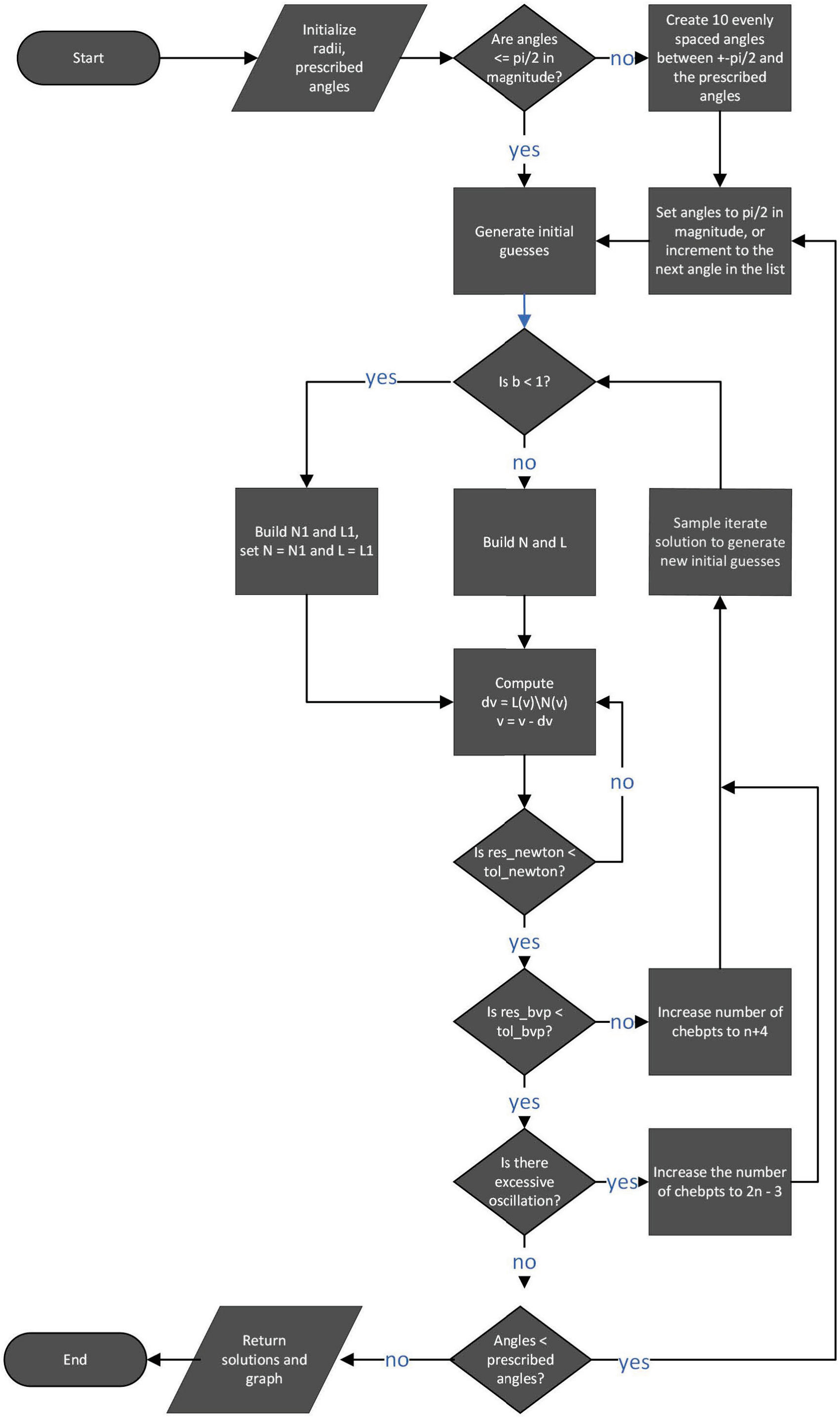
Start
Initialize radii, prescribed angles
Are angles <= pi/2 in magnitude?
no
Create 10 evenly spaced angles between +-pi/2 and the prescribed angles
yes
Generate initial guesses
Set angles to pi/2 in magnitude, or increment to the next angle in the list
yes
Is b < 1?
no
Build N1 and L1, set N = N1 and L = L1
Build N and L
Sample iterate solution to generate new initial guesses
Compute dv = L(v)\N(v) v = v - dv
no
Is res_newton < tol_newton?
yes
Is res_bvp < tol_bvp?
no
Increase number of chebpts to n+4
yes
Is there excessive oscillation?
yes
Increase the number of chebpts to 2n - 3
no
End
Return solutions and graph
no
Angles < prescribed angles?
yes

FIG. 3.1. *The flowchart describing the general algorithm.*

We will need Chebyshev differentiation matrices in what follows. These matrices can be realized as representing the linear transformation between two vectors of data corresponding to particular grid points, say $\mathbf{f}$ being mapped to $\mathbf{f}'$. The data at the grid points correspond to the interpolating polynomials for a function f and its derivative f', where the data are sampled at Chebyshev grid points $x_j = \cos(\theta_j) \in [-1, 1]$ with the angles θ_j equally spaced over $[0, \pi]$. Of course, these grid points do not need to be fixed, as multiple samplings can be used.

The basic building blocks of both the nonlinear equation $N(\mathbf{v}) = \mathbf{0}$ and the linearized equation $Ld\mathbf{v} = -N$ are based on D^0 and D, implemented by the Chebfun commands

```
D0 = diffmat([n-1 n],0,X);
D1 = diffmat([n-1 n],1,X);
```

where $X = [-1, 1]$, n is the number of Chebyshev points we are using, and the input 0 or 1 indicates the number of derivatives. Since `D0` is rectangular, it becomes an $(n-1) \times n$ identity matrix interpreted as a dense "spectral down-sampling" matrix implemented as interpolating on an n-point grid followed by sampling on an $(n-1)$-point grid. That is, if we have represented a function f by the n-sample Chebyshev points stored in the vector $\mathbf{f}$, then $D0 * \mathbf{f}$ is a vector of length $n-1$ that corresponds to a sampling f at the $n-1$ Chebyshev points. We sparsely construct our operators N and L using these components in a block fashion. These building blocks are found by multiplying a linear operator times the vector $\mathbf{v}$ that contains the solution components. Then we use these building blocks to put together N and the square matrix L. If b is small, then we construct N_1 and L_1, where we have no excessive numerical error due to rounding near the singularity.

We always take care that the number of Chebyshev points is chosen so that there is no evaluation of $r = 0$, following Trefethen's treatment of the radial form of the Laplace equation [13]. The basic loop is

```
while res_newton > tol_newton
      dv = L(v)\N(v);
      v = v - dv;
      res_newton = norm(dv,'fro')/norm(v,'fro');
end
```

The tolerance `tol_newton` must be met, and we use the relative error measured by the Frobenius norm. For all of the examples in this paper, we used `tol_newton = 1e-13`.

If the Newton's method fails to converge within a specified maximum number of iterations, then we increase the number of Chebyshev points, we sample the current state of the iteration onto the new Chebyshev points, reinitialize the operators N and L as above, and enter another loop for Newton's method. Further, we include this process in an outer loop that also tests the relative error of the iterates even if the Newton's method converges. We use the Frobenius norm to compute `res_bvp` as $||N(v)||/||v||$. If this residual is larger than the prescribed tolerance, then we also increase the number of Chebyshev points. We used a prescribed tolerance of `tol_bvp = 1e-12` in all of the examples in this paper. It may also be that there is excessive polynomial oscillation across the Chebyshev points and that there are not enough of these points to accurately resolve the solution. In this case we again increase the number of Chebyshev points; however, this increase is more aggressive.

If the boundary conditions are specified so that the solution curve will not be a graph over a base-domain, such as when $\psi_b > \pi/2$, then we use a method of continuation. This applies to most of the problems considered in this paper. We harvest the sign of ψ_b, and in place of that boundary condition, we solve ten problems with linearly spaced boundary angles between $\pi/2$ and $|\psi_b|$ using the converged solution at each step as an initial guess for the next step. We preserve the number of needed Chebyshev points from the adaptive algorithm while moving from one step to the next. In $\mathbf{P2}$, if needed, we do this for the inclination angles at a and b simultaneously.

4. A multi-scale approach via domain decomposition.

We partition the arc-length domains into two or three subdomains, depending on the problem. The general approach is to identify a parameter that marks the boundary between the subdomains and then to put conditions in place so that the solutions on the subdomains extend across this artificial boundary as a solution for the global problem.

For many of the problems we choose the parameter to be a radius. We use the setting of **P1** to describe the details of the process for that choice of parameter. For some problems we find the parameter of the inclination angle to be a better choice for marking the boundary of the subdomains, and we use the setting of **P2** to describe the process for that choice of parameter.

In either case, we build a vector $\mathbf{v}$ and construct appropriate operators $N(\mathbf{v})$ and $F(\mathbf{v}) = L(\mathbf{v})\mathbf{v}$ for the partitioned domains with the matching conditions included. Then we follow the basic algorithm as described in the flowchart in Figure 3.1, where the relative errors and adaptive steps are taken corresponding to each subdomain.

It should be noted that while this is simple enough to describe, the computer code needed to implement these concepts becomes somewhat lengthy and care must be taken in the adaptive procedures for generating $\mathbf{v}$, N, and L. We will provide more detail in what follows.

4.1. P1-study.

For this case we require that $\psi = \psi_b$ at $r = b$, and, by symmetry, $\psi = -\psi_b$ at $r = -b$. The base code [15] works well in general. However, we are able to improve the performance if $\psi_b > \pi/2$ and $b \gg 1$. We then choose a $\delta > 0$ and pick the parameters to mark the subdomain boundaries at $r = \delta - b$ and $r = b - \delta$. This partitions the arc-length domain into three subdomains. Denote those domains by Ω_1, Ω_2, and Ω_3, respectively, as one moves from left to right along the solution curve. We also use these subscripts to denote the restrictions of solutions (r, u, ψ) on those subdomains. The arc-lengths at the boundary points are denoted by ℓ_1 and ℓ_2.

Physically, the height u and inclination angle ψ on Ω_2 will both be approximately zero, and the radius will be approximately measured by a translation of the arc-length. Thus, relatively few Chebyshev points are needed to resolve the solution to within the tolerance requested. On both Ω_1 and Ω_3 this is no longer the case, and the adaptive method of continuity described in Section 3 is used to resolve the solutions there. The conditions over the boundaries are

$$\begin{aligned} u_1(\ell_1) &= u_2(\ell_1), & u_2(\ell_2) &= u_3(\ell_2), \\ \psi_1(\ell_1) &= \psi_2(\ell_1), & \psi_2(\ell_2) &= \psi_3(\ell_2). \end{aligned}$$

Some explanation of the initial guesses for the underlying Newton's method is in order, although we keep this brief as [15] contains the formulas for the functions we need. On Ω_1 we use a circular arc that meets the boundary conditions at $r = -b$ and is horizontal at $r = -b + \delta$. On Ω_2 we use a line with zero height. On Ω_3 we again use a circular arc that meets the boundary conditions at $r = b$ and is horizontal at $r = b - \delta$. We also prescribe the heights of these circular arcs so that the heights match the boundaries of the subdomains. Since we are primarily interested in problems where $|\psi_b| > \pi/2$, we use these guesses to produce a converged solution for $|\psi_b| = \pi/2$, and then we use the method of continuation to feed that converged solution to problems with a sequence of increasing boundary angles up to $|\psi_b|$.

The computational vector is $\mathbf{v} = [R_1\, R_2\, R_3\, U_1\, U_2\, U_3\, \Psi_1\, \Psi_2\, \Psi_3\, \ell_1\, \ell_2\, \ell_3]^T$, and the mathematical description of N is given by

$$N(\mathbf{v}) = \begin{bmatrix} R_1'(\tau_1) - \ell_1 \cos\Psi_1(\tau_1) \\ R_2'(\tau_2) - \ell_2 \cos\Psi_2(\tau_2) \\ R_3'(\tau_3) - \ell_3 \cos\Psi_3(\tau_3) \\ U_1'(\tau_1) - \ell_1 \sin\Psi_1(\tau_1) \\ U_2'(\tau_2) - \ell_2 \sin\Psi_2(\tau_2) \\ U_3'(\tau_3) - \ell_3 \sin\Psi_3(\tau_3) \\ \Psi_1'(\tau_1) + \frac{\ell_1 \sin\Psi_1(\tau_1)}{R_1(\tau_1)} - \kappa\ell_1 U_1(\tau_1) \\ \Psi_2'(\tau_2) + \frac{\ell_2 \sin\Psi_2(\tau_2)}{R_2(\tau_2)} - \kappa\ell_2 U_2(\tau_2) \\ \Psi_3'(\tau_3) + \frac{\ell_3 \sin\Psi_3(\tau_3)}{R_3(\tau_3)} - \kappa\ell_3 U_3(\tau_3) \\ R_1(-1) + b \\ R_1(1) + b - \delta \\ \Psi_1(-1) + \psi_b \\ \Psi_1(1) - \Psi_2(-1) \\ R_2(-1) + b - \delta \\ U_1(1) - U_2(-1) \\ R_2(1) - b + \delta \\ \Psi_2(1) - \Psi_3(-1) \\ R_3(-1) - b + \delta \\ U_2(1) - U_3(-1) \\ R_3(1) - b \\ \Psi_3(1) - \psi_b \end{bmatrix}.$$

For $i = 1, 2, 3$ we allocate in each of these subdomains an independent variable τ_i, and at the appropriate step of the algorithm, the number of grid-points in each τ_i will be adaptively increased if the tolerances for the residual of N restricted to Ω_i is not met, as indicated in Figure 3.1. We scale the problem as in Section 3 to determine these arc-lengths ℓ_i, for $i = 1, 2, 3$, in the process of solving the problem. Then, L is given by the block matrix

$$L = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \\ M_{31} & M_{32} \end{bmatrix}, \tag{4.1}$$

where $M_{11} = I \otimes D$, using the Kronecker product and where I is the identity on $\mathbb{R}^{6\times 6}$, and

$$M_{12} = \begin{bmatrix} \ell_1 \sin\Psi_1 & 0 & 0 & -\cos\Psi_1 & 0 & 0 \\ 0 & \ell_2 \sin\Psi_2 & 0 & 0 & -\cos\Psi_2 & 0 \\ 0 & 0 & \ell_3 \sin\Psi_3 & 0 & 0 & -\cos\Psi_3 \\ -\ell_1 \cos\Psi_1 & 0 & 0 & -\sin\Psi_1 & 0 & 0 \\ 0 & -\ell_2 \cos\Psi_2 & 0 & 0 & -\sin\Psi_2 & 0 \\ 0 & 0 & -\ell_3 \cos\Psi_3 & 0 & 0 & -\sin\Psi_3 \end{bmatrix},$$

$$M_{21} = \begin{bmatrix} \frac{-\ell_1 \sin\Psi_1}{R_1^2} & 0 & 0 & -\kappa\ell_1 & 0 & 0 \\ 0 & \frac{-\ell_2 \sin\Psi_2}{R_2^2} & 0 & 0 & -\kappa\ell_2 & 0 \\ 0 & 0 & \frac{-\ell_3 \sin\Psi_3}{R_3^2} & 0 & 0 & -\kappa\ell_3 \end{bmatrix},$$

$$M_{22} = \begin{bmatrix} D + \frac{\ell_1 \cos\Psi_1}{R_1} & 0 & 0 & -\kappa U_1 & 0 & 0 \\ 0 & D + \frac{\ell_2 \cos\Psi_2}{R_2} & 0 & 0 & -\kappa U_2 & 0 \\ 0 & 0 & D + \frac{\ell_3 \cos\Psi_3}{R_3} & 0 & 0 & -\kappa U_3 \end{bmatrix},$$

and the blocks corresponding to the boundary conditions and matching conditions are given by

$$M_{31} = \begin{bmatrix} D^0_{-1} & 0 & 0 & 0 & 0 & 0 \\ D^0_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & D^0_{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & D^0_1 & -D^0_{-1} & 0 \\ 0 & D^0_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D^0_{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D^0_1 & -D^0_{-1} \\ 0 & 0 & D^0_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad M_{32} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ D^0_{-1} & 0 & 0 & 0 & 0 & 0 \\ D^0_1 & -D^0_{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & D^0_1 & -D^0_{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D^0_1 & 0 & 0 & 0 \end{bmatrix}.$$

With these objects in hand we use the general algorithm from the flowchart in Figure 3.1.

For measures of performance, we are primarily interested in the size of the linear systems used and the number of Newton iterations needed for convergence. To give an approximation of the sizes of these problems, let n_v be the length of the computational vector $\mathbf{v}$, giving the size of the linear system. We denote the number of Newton iterations by n_N, and this roughly indicates how many times the algorithm solves a system of size $n_v \times n_v$. Note that since the size of $\mathbf{v}$ is potentially growing in the adaptive phases, these numbers only give an upper bound for the complexity of the problem.

While it would be possible to compute the exact numbers, we find that this approach illuminates the process well enough without the extra detail. Of course we are also interested in the configurations where the multi-scale algorithm converges when the program from [15] fails to meet the same tolerances. Some examples of these experiments are collected in Table 4.1. Figure 4.1 displays the last example from the table. We focused on problems with large radii b, so the approach indicated by (2.11) was not used. In the example computed for Figure 2.3, the condition numbers for L started at 2,169.05 and decreased to 1,555.35 over the 141 iterations needed. In the example computed for Figure 4.1, the condition numbers for L started at 1,843 and increased to 669,032 with the adaptive addition of grid-points over the 139 iterations needed. While this ending condition number starts getting large, the converged solution satisfies the tolerance of the boundary value problem, and the extra points that cause the increase in the condition number were only added when the converged Newton steps did not lead to a configuration that met the tolerance of the BVP.

TABLE 4.1

Solutions that are the image of a disk with radius b are collected for a selection of inclination angles. The size of the corresponding linear system is given by n_v, for the length of the computational vector $\mathbf{v}$, while the number of Newton iterations is given by n_N. We label the columns corresponding to the multi-scale code by MS. The value of δ used for each experiment is included. If the algorithm did not converge, then it is labeled as dnc.

b	ψ_b	base code n_v	base code n_N	MS code n_v	MS code n_N	δ
11	π	154	53	90	130	2
29	$3\pi/8$	298	16	246	132	3
29	π	586	58	246	132	3
40	π	dnc		270	139	5

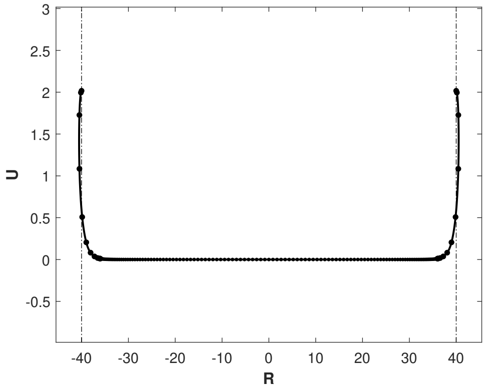

FIG. 4.1. *A capillary surface with an inclination angle of $\psi_b = \pi$ and an extremely large radius of $b = 40$. The vertical axis has been scaled to make the vertical displacement visible in this example. Here, the density of the grid-points in the middle zone of the domain decomposition is roughly one grid-point per unit length, and the upward swings of the interface are completely captured by the left and right zones. Without the domain decomposition, the density in the center would be much larger in order to accurately resolve the upswing behavior in the outer zones.*

4.2. P2-study. For the class of problems **P2** that are the image of an annulus, given $0 < a < b < \infty$, we require that $\psi = \psi_a$ at $r = a$ and $\psi = \psi_b$ at $r = b$. We use our approach for two types of configurations: first, where $a \ll \infty$ and b is somewhat large, and second, where $0 < a \ll 1$ and b is moderately sized. To accomplish this we use three different domain decompositions.

We first consider a case with three radial zones or subdomains, and refer to it as 3RZ below. We choose a $\delta > 0$ and pick our parameters to mark the boundaries at $r = a + \delta$ and $r = b - \delta$ to partition the arc-length domain into three subdomains just as we did in Section 4.1. The primary difference in this case is that the last 12 rows of $N(\mathbf{v})$ corresponding to the boundary conditions and matching conditions are replaced with

$$\begin{bmatrix} R_1(-1) - a \\ R_1(1) - a + \delta \\ \Psi_1(-1) - \psi_a \\ \Psi_1(1) - \Psi_2(-1) \\ R_2(-1) - a + \delta \\ U_1(1) - U_2(-1) \\ R_2(1) - b + \delta \\ \Psi_2(1) - \Psi_3(-1) \\ R_3(-1) - b + \delta \\ U_2(1) - U_3(-1) \\ R_3(1) - b \\ \Psi_3(1) - \psi_b \end{bmatrix}.$$

and L is unchanged. If $a < 1$, then the row replacement of (2.11) is used in Ω_1, and we illustrate those details in the following cases. Table 4.2 and Table 4.3 include some experiments that highlight the performance here. Table 4.2 has cases with $\psi_b < 0$ and Table 4.3 has cases with $\psi_b > 0$. Figure 4.2 displays the last example from Table 4.2.

For the next case under consideration, we split the domain into two regions as described in Section 4, where we again use the radius as the parameter that marks the boundary between the two subdomains. We use Ω_1 for the arc-lengths on the left of $r = a + \delta$ and Ω_2 for the

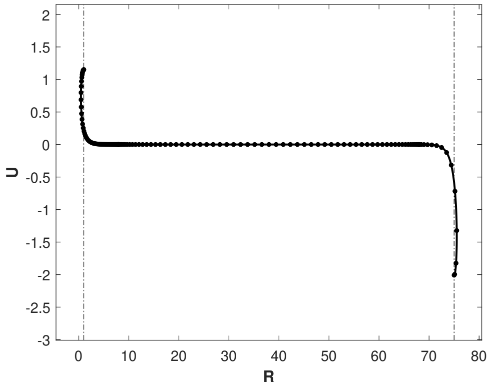

FIG. 4.2. *An annular capillary surface with radii $a = 1$ and $b = 75$ with inclination angles $\psi_a, \psi_b = -\pi$. The vertical axis has been scaled like in the last example. The character of the density and the zones capturing the regions of height change are both the same here. Note that the adaptive algorithm places more grid-points on the left upward swing of the interface in comparison to the right downward swing despite the fact that the total displacement is less on the left.*

TABLE 4.2
Solutions that are the image of an annulus with radii a and b, a selection of inclination angles focused on $\psi_b = -\pi$, and the total number of points as before.

a	b	ψ_a	ψ_b	base n_v	base n_N	3RZ n_v	3RZ n_N	2RZ n_v	2RZ n_N	δ
1	5	$-\pi/2$	$-\pi/2$	46	8	114	69	80	6	0.4
1	5	$-15\pi/16$	$-15\pi/16$	46	58	114	175	80	117	0.4
1	5	$-\pi$	$-\pi$	82	63	114	189	80	138	0.4
1	44	$-\pi/2$	$-\pi/2$	586	24	198	67	248	21	3
1	44	$-15\pi/16$	$-15\pi/16$	1162	71	198	158	476	134	3
1	44	$-\pi$	$-\pi$	1174	73	198	171	464	152	3
1	47	$-\pi/2$	$-\pi/2$	dnc		198	69	dnc		5
1	47	$-\pi$	$-\pi$	dnc		246	183	dnc		5
1	75	$-\pi$	$-\pi$	dnc		462	186	dnc		7

arc-lengths to the right of $a + \delta$. We refer to this case as 2RZ for two radial zones. Here we are primarily interested in small values of a and modest values of b, as we can see from the performance indicated in Table 4.2. We have $\mathbf{v} = [R_1\, R_2\, U_1\, U_2\, \Psi_1\, \Psi_2\, \ell_1\, \ell_2]^T$ and

$$N(\mathbf{v}) = \begin{bmatrix} R_1'(\tau_1) - \ell_1 \cos\Psi_1(\tau_1) \\ R_2'(\tau_2) - \ell_2 \cos\Psi_2(\tau_2) \\ U_1'(\tau_1) - \ell_1 \sin\Psi_1(\tau_1) \\ U_2'(\tau_2) - \ell_2 \sin\Psi_2(\tau_2) \\ R_1(\tau_1)\Psi_1'(\tau_1) + \ell_1 \sin\Psi_1(\tau_1) - \kappa\ell_1 R_1(\tau_1)U_1(\tau_1) \\ \Psi_2'(\tau_2) + \frac{\ell_2 \sin\Psi_2(\tau_2)}{R_2(\tau_2)} - \kappa\ell_2 U_2(\tau_2) \\ R_1(-1) - a \\ R_1(1) - a - \delta \\ \Psi_1(-1) + \psi_b \\ \Psi_1(1) - \Psi_2(-1) \\ R_2(1) - b \\ \Psi_2(1) - \psi_b \end{bmatrix}.$$

TABLE 4.3
*Solutions that are the image of an annulus with radii a and b, a selection of inclination angles focused on $\psi_b = \pi$, and the total number of points as before. For the base code example marked with *, we increased the initial number of Chebyshev points n from 15 to 19 to eliminate excessive oscillation.*

a	b	ψ_a	ψ_b	base n_v	base n_N	3RZ n_v	3RZ n_N	2RZ n_v	2RZ n_N	δ
1	5	- π	π	58*	56*	114	151	80	135	0.4
1	18	- π	π	298	65	126	157	152	137	2
1	44	- π	π	1174	808	210	174	dnc		3
1	47	- π	π	dnc		318	179	dnc		3
1	75	- π	π	dnc		510	177	dnc		5

Then we compute the Fréchet derivative $F(\mathbf{v}) = L\mathbf{v}$ to find the following expression for L:

$$\begin{bmatrix} D & 0 & 0 & 0 & \ell_1 \sin\Psi_1 & 0 & -\cos\Psi_1 & 0 \\ 0 & D & 0 & 0 & 0 & \ell_2 \sin\Psi_2 & 0 & -\cos\Psi_2 \\ 0 & 0 & D & 0 & -\ell_1 \cos\Psi_1 & 0 & -\sin\Psi_1 & 0 \\ 0 & 0 & 0 & D & 0 & -\ell_2\cos\Psi_2 & 0 & -\sin\Psi_2 \\ \Psi_1' - \kappa\ell_1 U_1 & 0 & -\kappa\ell_1 R_1 & 0 & D + \ell_1\cos\Psi_1 & 0 & \sin\Psi_1 - \kappa R_1 U_1 & 0 \\ 0 & \frac{-\ell_2 \sin\Psi_2}{R_2^2} & 0 & -\kappa\ell_1 & 0 & D + \frac{\ell_2\cos\Psi_2}{R_2} & 0 & \kappa U_2 \\ D^0_{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ D^0_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D^0_{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D^0_1 & -D^0_{-1} & 0 & 0 \\ 0 & D^0_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D^0_1 & 0 & 0 & 0 \end{bmatrix}.$$

The initial guesses for the underlying Newton's method are somewhat different in this case and the next. For the problems considered up to now, we used knowledge of the expected geometry of the solution to guide the initial guesses. In the problems with small a and modest b, where $|\psi_a|, |\psi_b| > \pi/2$, we do not have particularly helpful estimates for the solution. So we use the base code from [15] to generate a solution where the angles are sgn$(\psi_a)\pi/2$ at $r = a$ and sgn$(\psi_b)\pi/2$ at $r = b$. Then we interpolate this solution on our subdomains to generate initial guesses for the process as we begin the method of continuation, increasing the magnitude of the angles incrementally up to $|\psi_a|, |\psi_b|$.

Some examples of these experiments are collected in Tables 4.2–4.4 and can be compared with the three subdomain results there. Figures 4.3 and 4.5 contain the last example from Table 4.4. Figure 4.4 has a smaller value of b so that the hook near a is more visible.

This leads us to our final case. Our goal here is to provide a refinement for the configuration pictured in Figure 4.3. We use the domain decomposition we just described based on two subdomains Ω_1 and Ω_2 as a starting place. Then we mark some $\bar{\psi}$, and we divide the subdomain Ω_1 into two regions Ω_α and Ω_β; however, the starting point of these algorithms will not in general have angles that would fall in Ω_α. We proceed with the method of continuation to increase the magnitude of the prescribed angle at $r = a$ through 21 evenly spaced values starting with magnitude $\pi/2$ and ending at $|\psi_a|$, which is presumably a value near π. We use the two radial subdomain formulation up to but not including the last step in this continuation. On that last step we pick $\bar{\psi} = -\pi/2$ when $\psi_a < 0$, and we split Ω_1 into Ω_α and Ω_β. This gives us three subdomains, and the boundary between Ω_α and Ω_β is marked by $\bar{\psi}$, while the boundary between Ω_β and Ω_2 is marked by $r = a + \delta$. Then, in this setting we have the arc-length $\bar{\ell}$ for Ω_α and the arc-length over Ω_2 is $\ell_1 - \bar{\ell}$. The matching conditions at $\bar{\psi}$ are given by

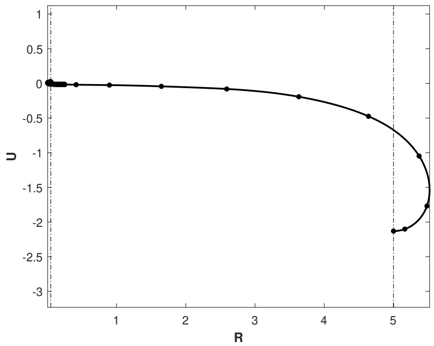

FIG. 4.3. *An annular capillary surface with radii $a = 0.05$ and $b = 5$ and inclination angles $\psi_a = -31\pi/32$ and $\psi_b = -\pi$ using 2RZ. This is the first of our "hook" examples. The left zone is requiring quite a few grid-points to pass through the vertical point and continue up to ψ_a.*

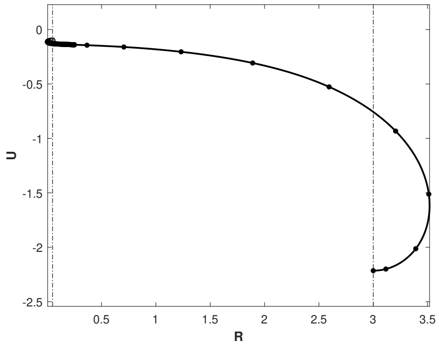

FIG. 4.4. *An annular capillary surface with radii $a = 0.05$ and $b = 3$ and inclination angles $\psi_a = -31\pi/32$ and $\psi_b = -\pi$ using A2RZ. Here the radius b is less than the previous example, and the domain decomposition based on a partition of the inclination angles is used. The number of grid-points near the left vertical point is still large, and the visibility of the small hook is poor. We will zoom in on this feature in the next figures.*

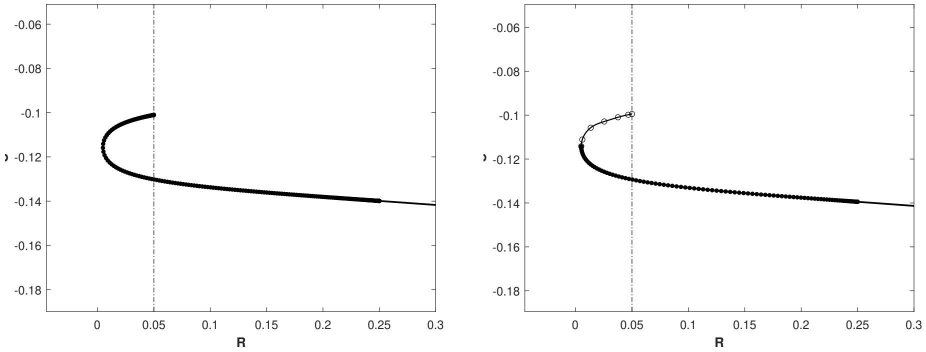

FIG. 4.5. *Zooming in on the hook with radii $a = 0.05$ and $b = 5$ and inclination angles $\psi_a = -31\pi/32$ and $\psi_b = -\pi$. The display on the left uses 2RZ and the display on the right uses A2RZ. While there are significantly fewer grid-points used in the second approach here, the angle $\psi_a = -\pi$ is still elusive.*

$$\begin{aligned} r_\alpha(\bar{\ell}) &= r_\beta(\bar{\ell}), \\ u_\alpha(\bar{\ell}) &= u_\beta(\bar{\ell}). \end{aligned}$$

We then roughly proceed as before. We first generate

$$N(\mathbf{v}) = \begin{bmatrix} R'_\alpha(\tau_\alpha) - \ell_\alpha \cos\Psi_\alpha(\tau_\alpha) \\ R'_\beta(\tau_\beta) - \ell_\beta \cos\Psi_\beta(\tau_\beta) \\ R'_2(\tau_2) - \ell_2 \cos\Psi_2(\tau_2) \\ U'_\alpha(\tau_\alpha) - \ell_\alpha \sin\Psi_\alpha(\tau_\alpha) \\ U'_\beta(\tau_\beta) - \ell_\beta \sin\Psi_\beta(\tau_\beta) \\ U'_2(\tau_2) - \ell_2 \sin\Psi_2(\tau_2) \\ \Psi'_\alpha(\tau_\alpha)R_\alpha(\tau_\alpha) + \ell_\alpha \sin\Psi_\alpha(\tau_\alpha) - \kappa\ell_\alpha R_\alpha(\tau_\alpha)U_\alpha(\tau_\alpha) \\ \Psi'_\beta(\tau_\beta)R_\beta(\tau_\beta) + \ell_\beta \sin\Psi_\beta(\tau_\beta) - \kappa\ell_\beta R_\beta(\tau_\beta)U_\beta(\tau_\beta) \\ \Psi'_2(\tau_2) + \frac{\ell_2 \sin\Psi_2(\tau_2)}{R_2(\tau_2)} - \kappa\ell_2 U_2(\tau_2) \\ R_\alpha(-1) - a \\ R_\alpha(1) - R_\beta(-1) \\ \Psi_\alpha(-1) - \psi_a \\ \Psi_\alpha(1) - \bar{\psi} \\ U_\alpha(1) - U_\beta(-1) \\ R_\beta(1) - a - \delta \\ \Psi_\beta(-1) - \bar{\psi} \\ \Psi_\beta(1) - \Psi_2(-1) \\ R_2(-1) - a - \delta \\ U_\beta(1) - U_2(-1) \\ R_2(1) - b \\ \Psi_2(1) - \psi_b \end{bmatrix}.$$

Then the blocks of L in (4.1) that need to be updated for this case are given by

$$\begin{aligned} M_{21} &= \begin{bmatrix} \Psi'_\alpha - \kappa\ell_\alpha U_\alpha & 0 & 0 & -\kappa\ell_\alpha R_\alpha & 0 & 0 \\ 0 & \Psi'_\beta - \kappa\ell_\beta U_\beta & 0 & 0 & -\kappa\ell_\beta R_\beta & 0 \\ 0 & 0 & \frac{-\ell_2 \sin\Psi_2}{R_2^2} & 0 & 0 & -\kappa\ell_2 & 0 \end{bmatrix}, \\ M_{22} &= \begin{bmatrix} D + \ell_\alpha \cos\Psi_\alpha & 0 & 0 & -\kappa R_\alpha U_\alpha & 0 & 0 \\ 0 & D + \ell_\beta \cos\Psi_\beta & 0 & 0 & -\kappa R_\beta U_\beta & 0 \\ 0 & 0 & D + \frac{\ell_2 \cos\Psi_2}{R_2} & 0 & 0 & -\kappa U_2 \end{bmatrix}, \end{aligned}$$

and the blocks corresponding to boundary conditions and matching conditions are given by

$$M_{31} = \begin{bmatrix} D^0_{-1} & 0 & 0 & 0 & 0 & 0 \\ D^0_1 & -D^0_{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & D^0_1 & -D^0_{-1} & 0 \\ 0 & D^0_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D^0_{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & D^0_1 & -D^0_{-1} \\ 0 & 0 & 0 & 0 & 0 & D^0_1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \; M_{32} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ D^0_{-1} & 0 & 0 & 0 & 0 & 0 \\ D^0_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & D^0_{-1} & 0 & 0 & 0 & 0 \\ 0 & D^0_1 & -D^0_{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D^0_1 & 0 & 0 & 0 \end{bmatrix}.$$

TABLE 4.4
Annular-type solutions with radii a and b, a selection of inclination angles, and the total number of points. In this table all cases had $\delta = 0.2$. The 3RZ cases that do not converge are due to the poor suitability of the standard initial guesses for these problems.

a	b	ψ_a	ψ_b	base n_v	base n_N	3RZ n_v	3RZ n_N	2RZ n_v	2RZ n_N
0.05	1	$-15\pi/16$	$-15\pi/16$	1162	78	dnc		332	198
0.05	1	$-31\pi/32$	$-\pi$	2314	82	dnc		620	250
0.05	3	$-15\pi/16$	$-15\pi/16$	2650	134	114	333	80	150
0.05	5	$-\pi/2$	$-\pi/2$	154	18	126	157	80	5
0.05	5	$-31\pi/32$	$-\pi$	dnc		426	358	620	228

TABLE 4.5
Solutions that are the image of an annulus with radii a and b, a selection of inclination angles, and the total number of points as before. In this table all cases had $\delta = 0.2$.

a	b	ψ_a	ψ_b	3RZ n_v	3RZ n_N	2RZ n_v	2RZ n_N	A2RZ n_v	A2RZ n_N
0.05	2	$-15\pi/16$	$-15\pi/16$	366	535	332	184	210	20
0.05	2	$-61\pi/64$	$-\pi$	654	551	620	197	354	25
0.05	2	$-31\pi/32$	$-\pi$	654	577	620	227	642	39
0.05	3	$-15\pi/16$	$-15\pi/16$	114	359	80	150	138	31
0.05	3	$-61\pi/64$	$-\pi$	366	404	332	193	138	38
0.05	3	$-31\pi/32$	$-\pi$	654	447	620	236	354	36
0.05	5	$-15\pi/16$	$-15\pi/16$	126	284	80	145	138	29
0.05	5	$-61\pi/64$	$-\pi$	378	322	332	186	138	37
0.05	5	$-31\pi/32$	$-\pi$	426	358	620	228	210	34

With these objects in hand, we again use the algorithm from the flowchart in Figure 3.1. These results are compared to the previous approaches in Table 4.5, and in each of the cases considered here, the base code failed.

In the example computed for Figure 4.2, the condition numbers for L started at 2,323.54 and increased to 191,742 with the adaptive addition of grid-points over the 186 iterations needed. In the example computed for Figure 4.3, the condition numbers for L started at 13,913.6 and increased to 36,681,500 with the adaptive addition of grid-points over the 228 iterations needed. In the example computed for Figure 4.4, the condition numbers for L increased from 257,701 to 10,298,600 over the 316 iterations needed. However, in this case some components saw a decrease in the condition number, and the large condition number is in the final increase of points. In general, the increases in the condition number along the iterations is primarily due to the adaptive algorithm. While these ending condition numbers do get large, Newton's method converges, and more importantly, the converged solutions satisfy the tolerance for the boundary value problem. This illustrates how demanding the underlying mathematical problem is.

5. Conclusions and closing remarks.

We have shown how several domain decompositions can improve performance over the adaptive Chebyshev spectral method in [15]. These new codes are primarily aimed at addressing problematic configurations identified in that work. The nonlinearities in the equations lead to complicated behavior that is nontrivial to classify, and as such it is difficult to determine beforehand which approach will lead to success. For the general scope of the annular problems, the existence theory is only partially completed, and the uniqueness of solutions is completely open [14]. With this perspective, it may take

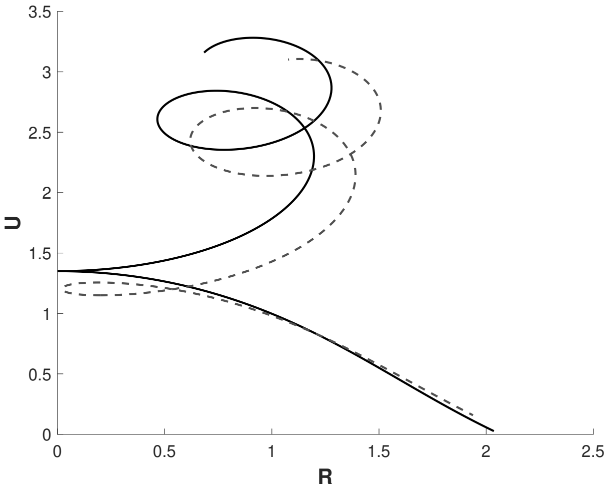

FIG. 5.1. *Two curves from the families of global solutions of* (2.1)–(2.3)*. One curve has a value of $r = 0$ with two component curves horizontally tangent there, and the other has a minimum value of r that corresponds to a loop containing a vertical point. If one takes the limit of this positive minimum r value of the loop to $r = 0$, then there is a singular pinching off of the loop, and the curve is extremely difficult to resolve for very small values of this loop radius. Also, while these are solutions of the capillary equations, the points of self-intersection are just artifacts of the global solutions to the mathematical equations with large arc-length $s > 0$. Physical fluids may be described by connected pieces of these curves as needed for an application so long as they do not contain any self-intersection points. At the beginning of the paper there were examples of physical configurations that could be built up from these connected pieces of the curves using both annular spans and components spanning $r = 0$.*

some experimenting with the number of subdomains and the location of the corresponding boundary points to achieve success.

For example, in the configurations where $\psi_a, \psi_b = -\pi$, when $0 < a \ll 1$ and b is moderately sized, the performance of the base code [15] is poor or fails completely, depending on how small a is. These configurations correspond to a singular limiting process that was explored by Bagley and Treinen [2]. In that work the global solutions of (2.1)–(2.3) were considered for all arc-lengths. These global solutions were classified into families of solutions, and the leftmost vertical point of the interface was a parameter used in this classification. When the radius of that leftmost vertical point is positive, the curve forms a loop there. As the radius of that vertical point goes to zero, the loop collapses into two curves that are tangent at $r = 0$, as indicated in Figure 5.1. (A rigorous study of this phenomenon has not yet been undertaken.) This behavior highlights the difficulty of the problem. While our attempts to use this multi-scale approach for this problem has improved the performance, we are still unable to find solutions up to $\psi_a = -\pi$ in some cases.

Our 3RZ code is designed to address $a \ll \infty$ and b somewhat large. Table 4.2 shows that this is working well. It does not function particularly well when b is closer to a, as seen in Table 4.4. This could be improved by using the initial guess scheme introduced for the 2RZ case.

Our 2RZ code is designed to address $0 < a \ll 1$ with moderate values of b, and while it does well with this difficult problem, getting to $\psi_a = -\pi$ is quite difficult. The A2RZ case is an attempt to resolve this, and while there is further improvement, getting all the way to $\psi_a = -\pi$ in all cases is still elusive.

This paper has focused on certain choices of inclination angles at particular radii. The opposite choices for those inclination angles lead to reflections of the surface over the radial axis, and work equally well. In general there are four parameters to explore, and it is possible that there are other difficult cases that we have not yet encountered. The programs we have written are available at the GitHub site referenced in the introduction and could be modified to treat these new cases as they are discovered.

We would like to thank the anonymous referees for their constructive and helpful comments.

REFERENCES

[1] J. L. AURENTZ AND L. N. TREFETHEN, *Block operators and spectral discretizations*, SIAM Rev., 59 (2017), pp. 423–446.

[2] Z. BAGLEY AND R. TREINEN, *On the classification and asymptotic behavior of the symmetric capillary surfaces*, Exp. Math., 27 (2018), pp. 215–229.

[3] T. A. DRISCOLL AND N. HALE, *Rectangular spectral collocation*, IMA J. Numer. Anal., 36 (2016), pp. 108–132.

[4] T. A. DRISCOLL, N. HALE, AND L. N. TREFETHEN, eds., *Chebfun Guide*, Pafnuty Publications, Oxford, 2014.

[5] A. ELCRAT, T.-E. KIM, AND R. TREINEN, *Annular capillary surfaces*, Arch. Math. (Basel), 82 (2004), pp. 449–467.

[6] A. ELCRAT AND R. TREINEN, *Numerical results for floating drops*, Discrete Contin. Dyn. Syst., (2005), pp. 241–249.

[7] R. FINN, *Equilibrium Capillary Surfaces*, Springer, New York, 1986.

[8] W. B. GRAGG AND R. A. TAPIA, *Optimal error bounds for the Newton-Kantorovich theorem*, SIAM J. Numer. Anal., 11 (1974), pp. 10–13.

[9] L. V. KANTOROVICH AND G. P. AKILOV, 2nd ed., *Functional Analysis*, Pergamon Press, Oxford-Elmsford, 1982.

[10] J. MCCUAN AND R. TREINEN, *Capillarity and Archimedes' principle of flotation*, Pacific J. Math., 265 (2013), pp. 123–150.

[11] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, 2nd ed., Springer, New York, 2006.

[12] J. M. ORTEGA, *The Newton-Kantorovich Theorem*, Amer. Math. Monthly, 75 (1968), pp. 658–660.

[13] L. N. TREFETHEN, *Spectral Methods in MATLAB*, SIAM, Philadelphia, 2000.

[14] R. TREINEN, *Extended annular capillary surfaces*, J. Math. Fluid Mech., 14 (2012), pp. 619–632.

[15] ———, *Spectral methods for capillary surfaces described by bounded generating curves*, Appl. Math. Comput., 450 (2023), Paper No. 127886, 17 pages.

[16] E. ZEIDLER, *Nonlinear Functional Analysis and its Applications. I*, Springer, New York, 1986.