

Implementation of Extended Systems Using Symbolic Algebra

K.A. Cliffe[†] and S.J.Tavener[‡]

July 13, 1999

Abstract

Following the pioneering work of Keller and others in the 1970s and '80s, numerical techniques for solving non-linear systems of equations that exhibit bifurcations have been developed to the point where they can potentially be applied to a wide range of problems arising in continuum mechanics. The central idea is to augment the discretised governing equations with one or more conditions so that the 'extended system' characterises a particular bifurcation point. By computing paths of singular points the behaviour of the system under investigation can be mapped out in a comprehensive fashion. Of considerable practical difficulty when implementing these methods is that they require the evaluation of derivatives of the discretised equations with respect to both the independent variables and the parameters. The higher the codimension of the singularity being sought, the higher the order of the derivatives required. Evaluating these derivatives is both tedious and error prone. An efficient method for computing the necessary derivatives for discretisations based on the Galerkin finite-element method will be presented that takes advantage of a symbolic algebra package. Our method makes it possible to deal with complicated non-linearities in a very straightforward manner. We demonstrate the complexity of systems that may be addressed by considering Marangoni convection in a two-dimensional domain with a deformable free surface.

1 Introduction

The equations governing the flow of most fluids are nonlinear, and as such can exhibit a variety of complex and interesting phenomena. Of particular interest here is the potential for bifurcations and the accompanying multiplicity of the solution set.

A landmark paper in the field of numerical computation of bifurcation problems by Keller [9] presented two key ideas which have provided the basis for much of the subsequent work in this area. The first idea is that of continuation and path following. Rather than focusing on the computation of single isolated solutions, *paths* of solutions are sought as a parameter is varied. In the case of incompressible flow, the Reynolds usually plays the role of this distinguished parameter. The method presented by Keller for computing such paths has subsequently been widely used.

The second key idea in Keller's paper is that of the extended system. Naïve path following techniques fail at points where the equations being solved are singular. Keller showed how to extend the system of equations under investigation by adding an additional parameter, the "pseudo arclength", and an additional equation controlling the pseudo arclength so that the combined system is nonsingular at points where the equations alone are singular, thereby allowing paths of solutions to be followed around limit points.

The ideas of Keller were subsequently developed and expanded by a number of authors in the late 1970s and 1980s. Extended systems were proposed to compute limit points [11], symmetry-breaking bifurcation points [13], Hopf bifurcation points [7] and other more complicated bifurcations [8]. All the proposed extended systems have the property that they are regular at their target bifurcation points. This property means that Newton's method, or one of its variants, can be used to solve the system and continuation methods can be used to compute paths of the bifurcation points.

At first these methods were applied to small systems of algebraic equations, and to equations arising from the discretisation of ordinary differential equations. However, it rapidly became apparent that they could be applied equally well to discretisations of partial differential equations, and in particular to the Navier-Stokes equations. In the early 1980s a number of papers appeared in which the ideas of continuation and path following, together with the use of extended systems, were applied to problems in fluid mechanics. Impressive agreement between numerically computed bifurcation structures and experimental observations was obtained, in particular for the

Taylor-Couette problem [12, 1, 2, 4, 3].

Continuation and path following methods are now widely used by computational scientists in general and by computational fluid dynamicists in particular. However, the more powerful methods to follow paths of bifurcation points in order to ‘map out’ the structure of the solution set, seem not to have been used widely in the computational fluid dynamics community despite the early successes of the methods. Part of the reason for this may be due to the fact that many flows of industrial importance do not exhibit the kind of multiplicity typical of the Taylor-Couette problem. Another, and perhaps more important factor, is that these methods are relatively difficult to implement. Extended systems for bifurcation points require the evaluation of various derivatives of the governing equations with respect to both the dependent variables and the parameters. These derivatives can be difficult to compute, particularly when additional physical phenomena besides the basic fluid flow, are being treated. If these derivatives have to be evaluated and implemented every time the discretisation scheme is changed or every time the physical phenomena being modelled are changed, the task rapidly becomes unmanageable.

The purpose of this paper is to present a complete and, we believe, elegant solution to the above problem. We shall describe a method that can be used to generate all the terms required for the solution of virtually any extended system, no matter how complicated. The method has been used in finite-element software for solving fluid mechanics problems, but such is its versatility that it can be used to study very general systems of second order elliptic partial differential equations. Most of our applications have been in two-dimensions, but the method is equally applicable to three-dimensional calculations. The main limitation in three dimensions is the high operation counts associated with direct methods. However, if robust and efficient alternatives to direct methods are available, our technique can be used in three dimensions.

The basic software required for our method is a set of routines that can compute element residual vectors and generalised element stiffness matrices. Virtually all finite-element software has these facilities and so we believe our methods could be used widely with relative ease.

The essential requirement is that all the quantities required for the extended systems be computed as sums of suitable element residual vectors and element matrices, and that these computations be implemented in a form that separates the specifics of the equations from the details of the

finite-element method. If this requirement is satisfied, we recognise that the part of the computation that depends on the governing equations requires subroutines (Fortran in our case) that can be produced automatically by a symbolic algebra package (REDUCE in our case) from a very simple description of the equations governing the problem being studied. The consequences of this observation are very significant. Once the code to solve the extended systems has been written it can then be used for a wide range of problems and can be used with a variety of finite-element methods. For example it has enabled electrohydrodynamic instabilities in nematic liquid crystals to be computed by directly solving the Ericksen-Leslie equations (see e.g. [10]) coupled to equations for the electric field and charge density, a problem that would otherwise have been intractable.

Computing the derivatives and the conversion of the formulae into code using a computer algebra system is much quicker and more accurate than performing the same operations by hand. New problems can be set up quickly and the full range of bifurcation techniques can be applied to them.

In the rest of this paper we describe our method in some detail. One representative extended system is introduced in section 2. Since this material is standard in the literature it is presented here only so that the various quantities that need to be computed can be identified. In section 3 the essential details of the method for computing the various derivatives are presented and are illustrated using a single scalar equation in one-space dimension. The method is of course applicable to systems of equations in two or three dimensions. In section 4 the method is applied to the problem of Marangoni convection with a deformable free surface. The free surface is handled using an orthogonal mapping technique. The governing equations comprise the balance of mass, momentum and energy together with a pair of elliptic equations that determine the mapping. This problem is already sufficiently complicated that it would, we believe, be very difficult without our methods. A few sample results, showing paths of bifurcations, are presented to illustrate the power of the method. More details can be found in the recent paper by Cliffe and Tavener [6]. Finally, in section 5 we present our conclusions.

2 Extended systems

In this section we introduce a typical extended system used to compute bifurcation points. The purpose is to illustrate the quantities that must be

computed in order to use extended system techniques to solve bifurcation problems. This material is now a well known and standard part of the literature and is presented here only for completeness.

We consider the following system of equations

$$\mathbf{F}(\mathbf{x}, \lambda, \mu) = \mathbf{0}, \quad \mathbf{F} : \mathbf{R}^N \times \mathbf{R} \times \mathbf{R} \mapsto \mathbf{R}^N. \quad (1)$$

These equations are assumed to arise from the discretisation of a physical fluid mechanical problem and will generally express the appropriate conservation laws – conservation of mass and momentum for example. The vector $\mathbf{x} \in \mathbf{R}^N$ is the vector whose components are the degrees of freedom used to approximate velocity, pressure, temperature, etc.

A simple limit point is defined to be a point $(\mathbf{x}_0, \lambda_0, \mu)$ such that

$$\mathbf{F}(\mathbf{x}_0, \lambda_0, \mu) = \mathbf{0} \quad (2)$$

$$\mathbf{F}_x(\mathbf{x}_0, \lambda_0, \mu) \vec{\phi}_0 = \mathbf{0} \quad (3)$$

$$\mathbf{l}^T \vec{\phi}_0 - 1 = 0 \quad (4)$$

with $\mathbf{F}_\lambda(\mathbf{x}_0, \lambda_0, \mu) \notin \text{Range}(\mathbf{F}_x(\mathbf{x}_0, \lambda_0, \mu))$. (Here and throughout, subscripts denote differentiation, e.g. $\mathbf{F}_\lambda = \partial \mathbf{F} / \partial \lambda$.)

We seek solutions of the extended system

$$\mathbf{G}(\mathbf{y}, \mu) = \begin{pmatrix} \mathbf{F} \\ \mathbf{F}_x \vec{\phi} \\ \mathbf{l}^T \vec{\phi} - 1 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ 0 \end{pmatrix}, \quad (5)$$

where

$$\mathbf{y} = (\mathbf{x}^T, \vec{\phi}^T, \lambda)^T \in \mathbf{R}^N \times \mathbf{R}^N \times \mathbf{R}^1,$$

and

$$\mathbf{G} : \mathbf{R}^N \times \mathbf{R}^N \times \mathbf{R}^1 \times \mathbf{R}^1 \mapsto \mathbf{R}^N \times \mathbf{R}^N \times \mathbf{R}^1.$$

To solve the system (5) by Newton's method, its Jacobian is required, which is given by

$$\mathbf{G}_y = \begin{pmatrix} \mathbf{F}_x & 0 & \mathbf{F}_\lambda \\ \mathbf{F}_{xx} \vec{\phi} & \mathbf{F}_x & \mathbf{F}_{x\lambda} \vec{\phi} \\ 0 & \mathbf{l}^T & 0 \end{pmatrix}. \quad (6)$$

The details of the linear algebra involved in solving systems of equations with this Jacobian as the coefficient matrix can be found in [5]. The point we

wish to make here is simply that the various quantities appearing in equation (5) and the Jacobian (6) must be calculated, and these quantities can be particularly difficult to compute when the equations arise from discretisations of problems in fluid mechanics.

3 Use of symbolic algebra

In this section we show how the various quantities required for the solution of extended systems may be computed efficiently. The method we present may be used for systems of equations in one, two or three dimensions. However, in order to present the basic idea we will only consider the case of a scalar equation in one dimension.

Consider a scalar differential equation (boundary value problem) of the form

$$f(u, u_r) - \frac{d}{dr}(g(u, u_r)) = 0, \quad (7)$$

$$u(0) = u(1) = 0, \quad (8)$$

where

$$f : \mathbf{R} \times \mathbf{R} \mapsto \mathbf{R}, \quad \text{and} \quad g : \mathbf{R} \times \mathbf{R} \mapsto \mathbf{R}.$$

Assume that f and g are sufficiently smooth so that all the derivatives appearing in this paper are continuous. For simplicity we have suppressed the dependence of the equations on the parameters.

The starting point for the finite element discretisation of the boundary value problem (7) and (8) is the weak form, which consists of finding $u(r) \in H_0^1[0, 1]$ such that

$$\int_0^1 \left[f(u, u_r)v + g(u, u_r)\frac{dv}{dr} \right] dr \quad \forall v \in H_0^1[0, 1], \quad (9)$$

where $H_0^1[0, 1]$ is the usual Hilbert space of functions defined on $[0, 1]$ that are zero at either end of the interval and whose first derivatives are square integrable.

The problem is discretised by replacing $H_0^1[0, 1]$ in the weak form by a finite-dimensional finite-element space $X_h \subset H_0^1[0, 1]$, of dimension N_h . Let v_1, v_2, \dots, v_{N_h} be a basis for X_h . The discrete weak form then consists of

finding $u_h(r) \in X_h$ such that

$$\int_0^1 \left[f(u_h, u_{h,r}) v_i + g(u_h, u_{h,r}) \frac{dv_i}{dr} \right] dr \quad \forall v_i \in X_h. \quad (10)$$

Now, since $u_h(r)$ may be written in the form

$$u_h(r) = \sum_{j=1}^{N_h} x_j v_j(r), \quad (11)$$

the weak form is equivalent to solving the system of equations

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{F} : \mathbf{R}^{N_h} \mapsto \mathbf{R}^{N_h}, \quad (12)$$

where

$$F_i(\mathbf{x}) = \int_0^1 \left[f(u_h, u_{h,r}) v_i + g(u_h, u_{h,r}) \frac{dv_i}{dr} \right] dr. \quad (13)$$

In practice the integral in the discrete weak form is evaluated using Gaussian quadrature. If r_k and w_k denote the N_g Gauss points and weights respectively, and we define

$$v_i^k = v_i(r_k), \quad (14)$$

$$v_{i,r}^k = \frac{dv_i}{dr}(r_k), \quad (15)$$

$$u_h^k = \sum_{j=1}^{N_h} x_j v_j^k, \quad (16)$$

$$u_{h,r}^k = \sum_{j=1}^{N_h} x_j v_{j,r}^k, \quad (17)$$

and

$$f^k = f(u_h^k, u_{h,r}^k), \quad (18)$$

$$g^k = g(u_h^k, u_{h,r}^k), \quad (19)$$

then

$$F_i(\mathbf{x}) = \sum_{k=1}^{N_g} w_k \left[f^k v_i^k + g^k v_{i,r}^k \right]. \quad (20)$$

It is important to note that in this expression for F_i , the precise form of the equation to be solved appears only through f^k and g^k , and, in particular, does not depend on the details of the finite element method. This means that it is possible to write the program in such a way that the problem specific part is confined to a subroutine, or set of subroutines, that compute the f^h and g^k given the values of u_h^k and $u_{h,r}^k$.

The computation of the Jacobian matrix for the system of equations $\mathbf{F} = \mathbf{0}$ is a straightforward application of the chain rule, namely

$$\frac{\partial F_i}{\partial x_j} = \sum_{k=1}^{N_g} w_k \left[\left(\frac{\partial f^k}{\partial u} \frac{\partial u_h^k}{\partial x_j} + \frac{\partial f^k}{\partial u_r} \frac{\partial u_{h,r}^k}{\partial x_j} \right) v_i^k + \left(\frac{\partial g^k}{\partial u} \frac{\partial u_h^k}{\partial x_j} + \frac{\partial g^k}{\partial u_r} \frac{\partial u_{h,r}^k}{\partial x_j} \right) v_{i,r}^k \right], \quad (21)$$

$$= \sum_{k=1}^{N_g} w_k \left[\left(\frac{\partial f^k}{\partial u} v_j^k + \frac{\partial f^k}{\partial u_r} v_{j,r}^k \right) v_i^k + \left(\frac{\partial g^k}{\partial u} v_j^k + \frac{\partial g^k}{\partial u_r} v_{j,r}^k \right) v_{i,r}^k \right], \quad (22)$$

using equations (14) to (17).

The computation of the Jacobian matrix times a vector is also straightforward. Suppose $\mathbf{F}_x \mathbf{y}$ is required. Let

$$\phi_h(r) = \sum_{j=1}^{N_h} y_j v_j(r), \quad (23)$$

then

$$\sum_{j=1}^{N_h} \frac{\partial F_i}{\partial x_j} y_j = \sum_{k=1}^{N_g} w_k \sum_{j=1}^{N_h} \left[\left(\frac{\partial f^k}{\partial u} y_j v_j^k + \frac{\partial f^k}{\partial u_r} y_j v_{j,r}^k \right) v_i^k + \left(\frac{\partial g^k}{\partial u} y_j v_j^k + \frac{\partial g^k}{\partial u_r} y_j v_{j,r}^k \right) v_{i,r}^k \right], \quad (24)$$

$$= \sum_{k=1}^{N_g} w_k \left[\left(\frac{\partial f^k}{\partial u} \phi_h^k + \frac{\partial f^k}{\partial u_r} \phi_{h,r}^k \right) v_i^k + \left(\frac{\partial g^k}{\partial u} \phi_h^k + \frac{\partial g^k}{\partial u_r} \phi_{h,r}^k \right) v_{i,r}^k \right], \quad (25)$$

where

$$\phi_h^k = \sum_{j=1}^{N_h} y_j v_j^k, \quad (26)$$

and

$$\phi_{h,r}^k = \sum_{j=1}^{N_h} y_j v_{j,r}^k. \quad (27)$$

It is important to note that the expression (25) for $\mathbf{F}_x \mathbf{y}$, is very similar to the expression (20) for \mathbf{F} itself, and can be implemented in a similar way. The other point to note is that the evaluation of both \mathbf{F}_x and $\mathbf{F}_x \mathbf{y}$ requires only the evaluation of expressions involving f , g and their derivatives. Furthermore, it is easy to see that any of the quantities arising from extended systems also fall into this category. The evaluation of the derivatives of f and g is a task for which symbolic algebra systems are ideally suited. For this work the REDUCE package was used, which can produce its output in the form of FORTRAN source code. This means that it is relatively easy to write a preprocessor that takes as its input the REDUCE description of the functions f and g and produces as its output the FORTRAN source required for all the extended systems.

4 Marangoni Convection

The approach described above was employed to compute bifurcations of a fluid flow in a two-dimensional domain with a deformable free surface. An orthogonal mapping technique, used to compute the location of the free surface produced an additional three elliptic partial differential equations to be solved together with the Navier-Stokes and energy equations. The algebraic complexity introduced by the mapping technique made evaluating the derivatives required for extended systems and writing the subroutines to evaluate them a daunting and error prone task. Further, eight non-dimensional parameters appeared in our mathematical description of the problem.

Consider steady flows of a Newtonian fluid in a 2D domain that is heated from below. Assume that the Boussinesq approximation is valid, that the surface tension depends linearly with temperature along the deformable free upper surface and that the two contact angles are known and are equal.

The unknown location of the free surface was determined by an orthogonal mapping technique. Let $\Omega(x, y)$ be the physical domain with four sides Γ_L , Γ_B , Γ_R and Γ_F , the left, bottom, right and upper (free) sides respectively, and let $(\psi(x, y), \phi(x, y))$ be a mapping from Ω onto a reference domain Ω' where

$$\Omega' = \{(\psi, \phi) : (\psi, \phi) \in [-1/2, 1/2] \times [0, 1]\},$$

such that the level curves of $\psi(x, y)$ and $\phi(x, y)$ are orthogonal, and the boundaries Γ_L , Γ_B , Γ_R and Γ_F are mapped onto $\psi = -1/2$, $\phi = 0$, $\psi = 1/2$ and $\phi = 1$ respectively. This coordinate transformation is *orthogonal* if

$$\nabla\psi \cdot \nabla\phi = 0 \quad \text{for all } (x, y) \in \Omega.$$

Swapping the roles of dependent and independent variables, it can easily be shown that $x(\psi, \phi)$ and $y(\psi, \phi)$ must satisfy the coupled elliptic equations

$$\frac{\partial}{\partial\phi} \left(\frac{1}{\lambda} \frac{\partial x}{\partial\phi} \right) + \frac{\partial}{\partial\psi} \left(\lambda \frac{\partial x}{\partial\psi} \right) = 0, \quad (28)$$

$$\frac{\partial}{\partial\phi} \left(\frac{1}{\lambda} \frac{\partial y}{\partial\phi} \right) + \frac{\partial}{\partial\psi} \left(\lambda \frac{\partial y}{\partial\psi} \right) = 0, \quad (29)$$

in Ω' , for some $\lambda(\psi, \phi)$.

Boundary conditions for $x(\psi, \phi)$ and $y(\psi, \phi)$ were provided by the fixed boundaries of the physical domain Ω , from the orthogonality condition applied along the boundaries of Ω , from the kinematic boundary condition along the free surface Γ_F , and from the imposed contact angles.

For simplicity, we required $\lambda(\psi, \phi)$ to be constant in Ω' and enforced this by solving

$$\frac{\partial^2 \lambda}{\partial\psi^2} + \frac{\partial^2 \lambda}{\partial\phi^2} = 0, \quad (30)$$

with zero normal derivative on the boundary of Ω' . The constant value of λ was determined by invoking the orthogonality conditions at an interior point of Ω' .

The Navier-Stokes and energy equations were coupled to equations (28), (29) and (30). In non-dimensional form they are

$$\nabla \cdot \mathbf{u} = 0, \quad (31)$$

$$\frac{M}{Pr} \frac{D\mathbf{u}}{Dt} = \nabla \cdot \boldsymbol{\tau} + \left(\frac{RT}{M} - \frac{G}{MCa} \right) \mathbf{j}, \quad (32)$$

$$M \frac{DT}{Dt} = \nabla^2 T, \quad (33)$$

where

$$\tau_{ij} = -p\delta_{ij} + (u_{i,j} + u_{j,i}).$$

Note that all derivatives in (31)–(33) are with respect to x and y . Nonslip boundary conditions were imposed along on Γ_L, Γ_B and Γ_R . The temperature was required to be constant along Γ_B , and insulating conditions were imposed along Γ_L and Γ_R . Along the free surface the conditions

$$\mathbf{u} \cdot \mathbf{n} = 0, \quad (34)$$

$$n_i \tau_{ij} n_j = K \left(\frac{1}{M Ca} - T \right), \quad (35)$$

$$t_i \tau_{ij} n_j = -\mathbf{t} \cdot \nabla T, \quad (36)$$

$$\mathbf{n} \cdot \nabla T + LT = 0 \quad (37)$$

were imposed, corresponding respectively to the kinematic, normal and tangential shear stress conditions and to the Newton law of cooling. Here \mathbf{n} and \mathbf{t} are the unit normal and tangential vectors at the free surface and K is the curvature of the free surface.

The depth of an undeformed fluid layer of equal volume was chosen as the length scale d , i.e. $d = \text{volume}/l$ where l was the length of the domain, and the ‘‘Marangoni’’ velocity $V_M = (\sigma_1 \beta d)/\mu$ was chosen as the velocity scale. Temperatures were nondimensionalised according to $T = (T^* - T_0)/(\beta d)$ where T_0 was the reference temperature far from the free surface. The seven non-dimensional parameters appearing in (31)–(37) are

$$\text{Aspect ratio, } \eta = \frac{l}{d}, \quad (38)$$

$$\text{Biot number, } L = \frac{hd}{k_0}, \quad (39)$$

$$\text{Bond number, } G = \frac{\rho_0 g d^2}{\sigma_0}, \quad (40)$$

$$\text{Capillary number, } Ca = \frac{\mu \kappa}{\sigma_0 d}, \quad (41)$$

$$\text{Marangoni number, } M = \frac{\sigma_1 \beta d^2}{\mu \kappa}, \quad (42)$$

$$\text{Prandtl number, } Pr = \frac{\mu}{\rho_0 \kappa}, \quad (43)$$

$$\text{Rayleigh number, } R = \frac{\rho_0 \alpha \beta g d^4}{\mu \kappa}. \quad (44)$$

Here k_0 is the thermal conductivity of the fluid and h is the surface conductance, σ_0 is the surface tension at the reference temperature T_0 and σ_1 is the rate of change of surface tension with temperature. The average vertical temperature gradient is β , ρ_0 is the density at the reference temperature T_0 , α is the coefficient of thermal expansion and κ is the coefficient of thermal diffusion. Finally, μ is the molecular viscosity and g is the acceleration due to gravity. It is important to remember that all derivatives appearing in (31)–(37) are with respect to x and y , but that all computations of $\mathbf{u}(\mathbf{x}(\psi, \phi))$, $p(\mathbf{x}(\psi, \phi))$ and $T(\mathbf{x}(\psi, \phi))$ are performed on Ω' , and so these derivatives must be rewritten in terms of ψ and ϕ using the chain rule.

For illustrative purposes we chose $G = R = 0$ and $Pr = L = 1$. With this choice of parameters, convection is driven by surface tension gradients alone and buoyancy plays no role. A non-conducting “trivial” solution exists for all values of the Marangoni number Ma , if both contact angles are specified to be 90 degrees. Under these circumstances the free surface is an isothermal surface. At a critical value of Ma that depends upon the aspect ratio and Capillary number, a pair of two-cell flows bifurcates from the trivial solution at a transcritical bifurcation point. This bifurcation is disconnected for all other values of the contact angles. For contact angles other than 90 degrees there is always a temperature gradient and thus a surface tension gradient along the free surface and convection is always present. A purely conducting solution cannot occur and the transcritical bifurcation to two-cell flows is disconnected. When the contact angle is less than 90 degrees and the free surface is convex, the centre of the free surface is closer to the hot bottom than the free surface next to the walls and there is a surface tension gradient directed from the centre outwards. Flows with up-welling at the middle arise with continuous increase in Marangoni number and are the so-called “primary” flows. Two-cell flows with down-welling at the middle arise as disconnected secondary flows that are stable only above a critical Marangoni number. The sense of convection in the primary and secondary flows is reversed for contact angles greater than 90 degrees. The locus of limit points for an aspect ratio $\eta = 1.7$ at which the disconnected secondary flow loses stability is plotted as a function of contact angle in figure 1.

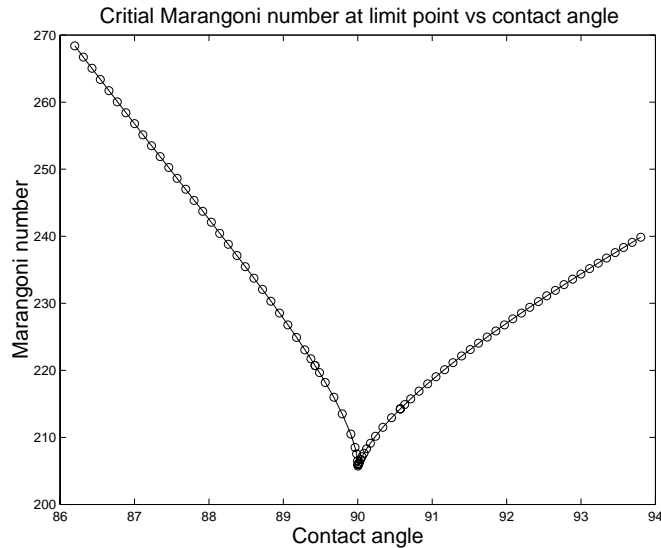


Figure 1: Locus of limit points on the secondary two-cell branch as a function of contact angle.

5 Conclusions

Numerical bifurcation methods can play a significant role in helping to develop our understanding of fluid dynamics. Although powerful methods have been available since the early 1980's, they have not been applied as widely to fluid dynamics as one might have expected. One of the reasons for this may be the difficulty in producing computer programs that make use of the extended system techniques for complicated fluid dynamical problems.

In the paper we have suggested an approach that makes use of symbolic algebra to implement extended system techniques in a general manner. The real advantage of the approach advocated here is that once the basic extended systems have been implemented, the work involved in applying them to new physical systems with new sets of governing equations is minimal. In fact the tedious and error prone task of computing the derivatives required and converting the results into computer code is carried out by the symbolic algebra package, REDUCE in our case, in a wholly automatic fashion.

We have illustrated the approach with reference to the computation of Marangoni convection with a deformable free surface. The free surface is

computed using an orthogonal mapping to transform the domain to a fixed and known domain on which the equations are solved by the finite-element method. It is our view that these calculations would have been too difficult and time consuming to have been performed without the use of the approach advocated in this paper.

References

- [1] T.B. Benjamin and T. Mullin. Notes on the multiplicity of flows in the Taylor experiment. *J. Fluid Mech.*, 121:219–230, 1982.
- [2] K.A. Cliffe. Numerical calculations of two-cell and single-cell Taylor flows. *J. Fluid Mech.*, 135:219–233, 1983.
- [3] K.A. Cliffe. Numerical calculations of the primary-flow exchange process in the Taylor problem. *J. Fluid Mech.*, 197:57–79, 1988.
- [4] K.A. Cliffe and T. Mullin. A numerical and experimental study of anomalous modes in the Taylor experiment. *J. Fluid Mech.*, 153:243–258, 1985.
- [5] K.A. Cliffe, A. Spence, and S.J. Tavener. The numerical analysis of bifurcation problems with application to fluid mechanics. In A. Iserles, editor, *Acta Numerica 2000*. CUP, 2000.
- [6] K.A. Cliffe and S.J. Tavener. Marangoni-Bénard convection with a deformable free surface. *J. Comp. Phys.*, 145:193–227, 1998.
- [7] A. Griewank and G. Reddien. The calculation of Hopf points by a direct method. *IMA J. Numer. Anal.*, 21:176–185, 1984.
- [8] A.D. Jepson and A. Spence. The numerical solution of nonlinear equations having several parameters, I: Scalar equations. *SIAM J. Numer. Anal.*, 22:736–759, 1985.
- [9] H.B. Keller. Numerical solutions of bifurcation and nonlinear eigenvalue problems. In P.H. Rabinowitz, editor, *Applications of Bifurcation Theory*, pages 359–384. Academic Press, 1977.

- [10] F.M. Leslie. Continuum theory for nematic liquid crystals. *Continuum Mech. Thermodyn.*, 4:167–175, 1992.
- [11] G. Moore and A. Spence. The calculation of turning points of nonlinear equations. *SIAM J. Numer. Anal.*, 17:567–576, 1980.
- [12] G.I. Taylor. Stability of a viscous liquid contained between two rotating cylinders. *Phil. Trans. R. Soc. London*, 223:289–343, 1923.
- [13] B. Werner and A. Spence. The computation of symmetry breaking bifurcation points. *SIAM J. Numer. Anal.*, 21:388–399, 1984.