INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 9, 1469–1486 (1989)

A NEWTON'S METHOD SCHEME FOR SOLVING FREE-SURFACE FLOW PROBLEMS

DAVID S. DANDY

Combustion Research Facility, Sandia National Laboratories, Livermore, CA 94551, U.S.A.

AND

L. GARY LEAL

Department of Chemical Engineering, California Institute of Technology, Pasadena, CA 91125, U.S.A.

SUMMARY

A Newton's method scheme is described for solving the system of non-linear algebraic equations arising when finite difference approximations are applied to the Navier–Stokes equations and their associated boundary conditions. The problem studied here is the steady, buoyancy-driven motion of a deformable bubble, assumed to consist of an inviscid, incompressible gas. The linear Newton system is solved using both direct and iterative equation solvers. The numerical results are in excellent agreement with previous work, and the method achieves quadratic convergence.

KEY WORDS Free-surface Bubble Intermediate Reynolds number Newton's method

INTRODUCTION

Recent advances in both computer hardware and numerical methods for free-boundary problems have contributed to a present state in which computational fluid dynamics can now play a significant role as a third fundamental investigative branch, with an essentially experimental flavour not based upon theoretical solutions of the problem. This approach can be used with particular advantage when it is desired to explore fundamental physical mechanisms via one-at-a-time variation of independent parameters, or when the length or time scales of resolution required are too small for conventional experimental measurement. One important class of problems that particularly fits this description is the dynamics of bubbles and drops, where experimental observation of shapes is often possible (though extreme measures may be required to achieve adequate resolution[1]) but details of the velocity or pressure fields are very difficult to obtain.

There are basically two fundamental classes of problems, and there are two corresponding solution techniques. When creeping motion or potential flow approximations can be applied, so that the governing partial differential equations are linear and only the boundary conditions are non-linear (owing to the fact that the boundary shape is unknown), quite powerful numerical techniques have been developed, based upon the boundary integral formulation, for studying problems involving highly deformed, time-dependent bubbles or drops.[2,3] At intermediate Reynolds numbers, however, we must retain the complete Navier–Stokes equations, and free-boundary problems can only be solved via an adaptation of the numerical methods developed for

0271–2091/89/121469–18$09.00

Received 15 August 1988
Revised 31 January 1989

finite Reynolds number flows. Much of this work has centred around finite element techniques, due, in part, to the belief of early investigators that finite difference techniques were only well suited to problems with fixed and simple boundary shapes coinciding with a co-ordinate line or surface in one of the known analytical co-ordinate systems. More recently, finite difference methods have enjoyed a resurgence of popularity for applications involving more complex boundary shapes, owing to the development of methods for numerical generation of boundary-fitted co-ordinate grids.[4] Evidentally, these methods should also be applicable for free-boundary problems, and the present paper is the first of a series in which we address the development and application of an iterative technique, based upon Newton's method, for this class of problems.

Previous applications of boundary-fitted co-ordinate techniques for solution of free-surface or free-boundary flow problems have in fact been limited to problems involving the motions of bubbles or drops in viscous fluids at intermediate Reynolds numbers. The classic problem of buoyancy-driven translation of a bubble through a quiescent fluid was solved using an orthogonal, boundary-fitted co-ordinate system by Ryskin and Leal,[5] and with a non-orthogonal grid by Christov and Volkov.[6] Later, the present authors solved for translation of a deformable drop[7] using a generalization of the orthogonal mapping method of Ryskin and Leal.[8] Ryskin and Leal[9] also considered bubble deformation in a steady, uniaxial extensional flow, while Kang and Leal[10,11] developed a technique for solution of transient moving-boundary problems. In all of these studies,[5,7-11] the numerical technique used was successive approximation (using the alternating direction implicit method of Peacemann and Rachford[12]), with the calculation of interface shape and boundary-fitted co-ordinate mapping decoupled from the flow variables. Although this method has proven to be versatile and useful for the class of problems considered, the decoupled successive approximations can require many iterations to converge, particularly when extreme under-relaxation is required to insure stability. Further, these methods are not optimal for solution of problems where we expect to find limit points, multiple solutions and instabilities (bifurcations) to other solution branches. In such cases it is difficult with the successive approximation technique to distinguish between difficulty with the numerics and a true approach to a singularity.

An alternative to the decoupled successive approximation methods of the previous studies is a global iteration in which we simultaneously iterate on all of the unknown variables, including the flowfield, the boundary shape and the co-ordinate mapping. Such a formulation would seem to be ideally suited to using Newton's method, but nothing of this nature has yet been developed for the finite difference approach to the solution of free- or moving-boundary problems. On the other hand, successful implementation of Newton's method in the context of a finite element formulation for free-boundary problems has already been reported,[13,14] and this was seen as a further advantage of the finite element approach. In the present work we show that a global Newton's scheme can also be applied to the finite difference, boundary-fitted co-ordinate technique for the solution of free-boundary flow problems, and we apply the resulting algorithm to re-investigate the dynamics of a rising gas bubble in a quiescent fluid. Newton's method is shown to approach quadratic convergence even for initial conditions that are far from the final steady-state solution. By application of standard techniques for computer-aided analysis of non-linear problems, we are able to explore the stability of steady, axisymmetric bubble shapes along the solution branch which starts from a sphere for zero Weber number.

FORMULATION

We begin by considering the formulation of governing equations and boundary conditions. To be concrete, we consider the specific problem of a deformable inviscid gas bubble (a void) that

undergoes a steady, buoyancy-driven motion through an outer, quiescent liquid. We do not mean to imply by this choice that the basic solution methodology cannot be applied to other free-surface or moving-boundary flow problems. On the contrary, the necessary modifications will generally be completely transparent, and this is especially true for other problems involving axisymmetric motions of gas bubbles or drops where the necessary changes are essentially only the boundary conditions in the far-field. The outer liquid is characterized by a constant viscosity μ and density ρ. The interface is assumed to be characterized completely by the surface tension γ, which is constant. As shown in Figure 1, the geometry of the system is conveniently represented in terms of cylindrical co-ordinates (z, σ, ϕ). We assume that the motion of the drop is rectilinear and thus that the shape and flowfield are both axisymmetric; therefore all quantities are independent of ϕ. It is also assumed that both the dispersed and continuous phases are incompressible, and the volume of the bubble is therefore constant.

Governing equations

To actually solve the governing equations and boundary conditions describing the flow and bubble shape, it is necessary to approximate them at discrete points in space, and these points are to be determined by generating a boundary-fitted co-ordinate grid. It is intuitive to think of the solution as involving a two-step procedure: first a grid is prescribed, and then discrete solutions for the flow and shape are found on this grid. However, information regarding the shape of the bubble is a boundary condition used in generating the grid, and while it is necessary to have a grid on which to discretize the governing equations, we cannot produce a grid until we know the solution to these same equations. The resolution of this apparent dilemma is to simultaneously solve the equations governing not only the flow and shape, but also those describing the co-ordinate mapping.

Axisymmetry enables us to simplify the mapping problem from construction of a three-dimensional grid to a two-dimensional grid. For reasons described elsewhere,[7] we perform an inverse conformal mapping of the infinite flow domain to a finite auxiliary domain. We then solve

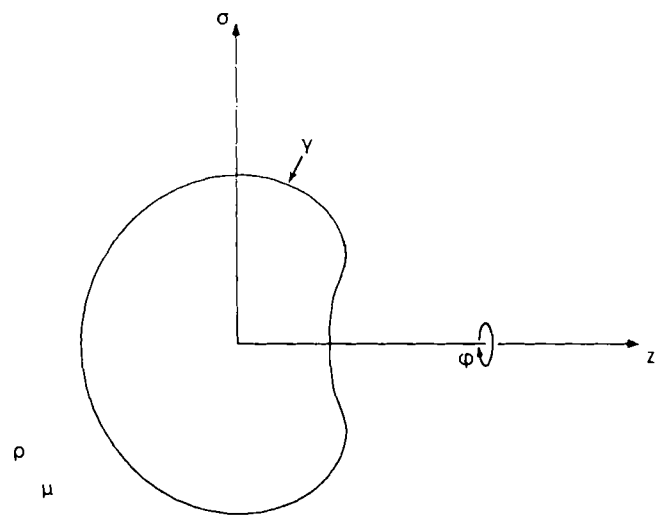

Figure 1. Schematic sketch of the problem

a pair of Laplace equations to obtain a discrete set $z^*(\xi, \eta)$ and $\sigma^*(\xi, \eta)$, where ξ and η are general curvilinear co-ordinate variables. Solution of the covariant Laplace equations provides us with a discrete mapping between the finite auxiliary space and a unit square in a computational (ξ, η) space. The absence of any explicit reference to the interface in the governing equations or boundary conditions allows us to use the strong constraint method.[8] The strong constraint method provides freedom to specify the distortion function $f(\xi, \eta)$. The governing equations for the mapping and the form chosen for the distortion function f are

$$f^2 z^*_{xx} + z^*_{yy} + f f_x z^*_x - f_y z^*_y / f = 0, \tag{1a}$$

$$f^2 \sigma^*_{xx} + \sigma^*_{yy} + f f_x \sigma^*_x - f_y \sigma^*_y / f = 0, \tag{1b}$$

$$f(\xi, \eta) = \pi \xi (1 - \beta \cos \pi \eta), \tag{2}$$

where $0 < \beta \leq 1$ is a parameter used to control the grid density in the η-direction. In equations (1) and for the duration of the paper a shorthand notation will be used: partial derivatives of the dependent variables with respect to the independent variable ξ will be denoted by the subscript x, and partial derivatives of the same variables with respect to η will be denoted by the subscript y. These subscripts are used to distinguish derivatives from components: $u_y = \partial u / \partial \eta$ whereas u_η is the component of velocity in the η-direction.

It is convenient to cast the flow equations in terms of the streamfunction and vorticity rather than the primitive variables $\mathbf{u}$, p. The process for arriving at the final equations is carried out in detail elsewhere,[7] but the steps will briefly be reviewed here. First the steady Navier–Stokes and continuity equations are non-dimensionalized using the radius a of an undeformed bubble as the characteristic length scale, the translation velocity of the bubble U_∞ as a characteristic velocity and a pressure scale $\frac{1}{2}\rho U_\infty^2$. The resulting equations contain the Reynolds number Re, which is defined as $Re = 2\rho a U_\infty / \mu$. The curl of the dimensionless Navier–Stokes equations is taken, and we obtain an equation in terms of the vorticity ω and velocity $\mathbf{u}$. Note that since this problem is axisymmetric, the vorticity vector has only one non-zero component. We then define the components of the velocities in the general (ξ, η) co-ordinate system using the streamfunction ψ:

$$u_\xi = -\frac{1}{\sigma h_\eta} \frac{\partial \psi}{\partial \eta}, \qquad u_\eta = \frac{1}{\sigma h_\xi} \frac{\partial \psi}{\partial \xi},$$

and substitute into the velocity–vorticity formulation to obtain

$$\mathscr{L}^2(\omega\sigma) - \frac{1}{2} \frac{Re}{h_\xi h_\eta} \left[\frac{u_\xi}{h_\xi} \frac{\partial}{\partial \xi} \left(\frac{\omega}{\sigma} \right) + \frac{u_\eta}{h_\eta} \frac{\partial}{\partial \eta} \left(\frac{\omega}{\sigma} \right) \right] = 0, \tag{3a}$$

$$\mathscr{L}^2 \psi + \omega = 0, \tag{3b}$$

where

$$\mathscr{L}^2 \equiv \frac{1}{h_\xi h_\eta} \left[\frac{\partial}{\partial \xi} \left(\frac{f}{\sigma} \frac{\partial}{\partial \xi} \right) + \frac{\partial}{\partial \eta} \left(\frac{1}{f\sigma} \frac{\partial}{\partial \eta} \right) \right].$$

The scale factors of the co-ordinate system are defined to be

$$h_\xi^2 = z_x^2 + \sigma_x^2, \qquad h_\eta^2 = z_y^2 + \sigma_y^2,$$

and their ratio is the distortion function

$$f = h_\eta / h_\xi.$$

When the operators in equations (3) are expanded, we obtain the following:

$$f^2\omega_{xx}+\omega_{yy}+f(f_x+f\sigma_x/\sigma)\omega_x+(\sigma_y/\sigma-f_y/f)\omega_y$$
$$-(1/\sigma^2)(f^2\sigma_x^2+\sigma_y^2)\omega-\tfrac{1}{2}Ref[(\tilde{u}_\xi\omega)_x+(\tilde{u}_\eta\omega)_y]=0, \tag{4a}$$

$$f^2\psi^*_{xx}+\psi^*_{yy}+f(f_x-f\sigma_x/\sigma)\psi^*_x-(f_y/f+\sigma_y/\sigma)\psi^*_y$$
$$-3f\xi\sigma(\tfrac{3}{2}f\xi\sigma_x+\tfrac{1}{2}\xi\sigma f_x+f\sigma)+h_\eta^2\sigma\omega=0. \tag{4b}$$

Since ψ asymptotically approaches infinite values as $\sigma\to\infty$, we have defined a modified streamfunction $\psi^*=\psi-\frac{1}{2}\sigma^2(1-\xi^3)$ which is bounded at infinity. With this modified streamfunction, we can write the velocity components as

$$\tilde{u}_\xi=h_\eta u_\xi=-\psi^*_y/\sigma-(1-\xi^3)\sigma_y,$$
$$\tilde{u}_\eta=h_\xi u_\eta=\psi^*_x/\sigma+(1-\xi^3)\sigma_x-\tfrac{3}{2}\sigma\xi^2.$$

Note that the dependent variables for the mapping appearing in equations (4) are z and σ, whereas the co-ordinate unknowns to be solved for are z^* and σ^*. Thus, everywhere that z, σ and their derivatives appear in the governing equations and boundary conditions, a conformal mapping relation must be used to write all of these variables in terms of the auxiliary mapping variables z^*, σ^* and their derivatives.

Boundary conditions

At infinity we require that there be uniform streaming flow, for which the velocity is constant and the flow is irrotational. Additionally, infinity in the physical (z, σ) co-ordinate system corresponds to the origin in the finite (z^*, σ^*) co-ordinate system. Thus we arrive at the following homogeneous condition at infinity:

$$\psi^*, \omega, z^*, \sigma^*=0 \quad \text{at } \xi=0. \tag{5}$$

The upstream and downstream axes of symmetry are considered to be zero streamlines, and the conditions at these boundaries are

$$\omega, \psi^*, z^*_y, \sigma^*=0 \quad \text{at } \eta=0, 1. \tag{6}$$

Any mapping satisfying equations (1) will be orthogonal in the interior of the domain; therefore the homogeneous Neumann condition on z^* insures that the grid is also orthogonal at the boundaries $\eta=0$ and $\eta=1$.

The boundary conditions at the interface $\xi=1$ are more complicated than those described above. First there is the kinematic condition, which states that the normal velocity of the interface $u_\xi(1, \eta)$ is zero at steady state. From the relation between the streamfunction and the velocity components it is seen that zero normal velocity is equivalent to requiring the streamfunction at $\xi=1$ to be independent of position along the interface. That is, the interface corresponds to a streamline

$$\psi^*=0 \quad \text{at } \xi=1. \tag{7}$$

There are two stress balances at the interface, the normal stress balance and the tangential stress balance. Since the bubble is assumed to be a void (zero viscosity and density), we find that there is zero tangential stress at the interface, and after some manipulation[5] the non-dimensional tangential stress balance reduces to

$$\omega-2\kappa_\eta u_\eta=0 \quad \text{at } \xi=1, \tag{8}$$

where κ_η is the principal radius of curvature in the η-direction and is given by

$$\kappa_\eta = (z_y \sigma_{yy} - z_{yy}\sigma_y)/h_\eta^3.$$

Similarly, manipulation of the normal stress balance yields the following non-dimensional form:[5]

$$\tau_{\xi\xi} - \frac{4}{W}(\kappa_\eta + \kappa_\phi) = 0, \tag{9}$$

where the Weber number is $W = 2\rho a U_\infty^2/\gamma$, and κ_ϕ is the principal radius of curvature in the azimuthal direction such that $\kappa_\phi = -f\sigma_x/\sigma h_\eta$. The component of the stress tensor appearing in equation (9) is

$$\tau_{\xi\xi} = -p - \frac{8}{Re\,\sigma h_\eta}(\sigma u_\eta)_y, \tag{10}$$

and the pressure is

$$p = \tfrac{3}{4}C_D z - u_\eta^2 - \frac{4}{Re}\int \frac{r}{\sigma}(\omega\sigma)_x \mathrm{d}\eta + \lambda, \tag{11}$$

where λ is an as-yet undetermined constant of integration, and the drag coefficient C_D is

$$C_D = 2\int_0^1 \sigma\sigma_y \left[u_\eta^2 + \frac{4}{Re}\left(\int^\eta \frac{f}{\sigma}(\omega\sigma)_x \mathrm{d}\hat{\eta} - \frac{2}{\sigma h_\eta}(\sigma u_\eta)_y \right) \right] \mathrm{d}\eta. \tag{12}$$

Finally, to force the mapping to be orthogonal at the interface, the following boundary condition is imposed:

$$z_x^* - \sigma_y^*/f = 0 \quad \text{at } \xi = 1. \tag{13}$$

Equation (13) is analogous to the Riemann condition in conformal mapping and is automatically satisfied in the interior of the domain, subject to the existence of a solution to equations (1).

As a final constraint on the system, necessary to close the system of equations and boundary conditions above, we require that the bubble be incompressible and thus conserve volume as it deforms. This constraint can be conceptually associated with the constant of integration λ in the expression for pressure, equation (11). In fact, in the global iteration scheme described here, λ is simply treated as one of the unknowns, which is to be determined simultaneously with all of the other unknowns, while the condition of constant volume provides one of the many equations that are to be solved. The integral equation imposed is

$$\int_0^1 \sigma^2(1, \eta) z_y(1, \eta) \mathrm{d}\eta + \tfrac{4}{3} = 0. \tag{14}$$

Algebraic equations

The unit square in computational (ξ, η) space is discretized by dividing both the ξ- and η-direction into N equally spaced nodes, yielding a grid of N^2 total nodes. The nodes in the ξ-direction will be denoted by the subscript i, such that $1 \le i \le N$, whereas the nodes in the η-direction will be represented by the subscript j, $1 \le j \le N$. For the results presented in this paper, $N = 61$.

To reduce equations (1) and (4) from differential equations to algebraic equations, it is a simple matter of using centred, second-order finite differences. Similarly, partial derivatives appearing in

the boundary conditions (8)–(13) are reduced to algebraic form using finite differences. Normal derivatives at the interface $\xi=1$ are approximated by second-order, one-sided differences. To approximate the integrals appearing in equations (11), (12) and (14), Simpson's rule was used for equations (12) and (14), while the trapezoidal rule was used for equation (11). The reason for using the two different integration schemes is that although Simpson's rule has higher accuracy than the trapezoidal rule, it requires that the number of points in the integration be odd—which is true in our case when we integrate over the surface from 0 to π since 61 points are used in the discretization—but when we integrate over only part of the surface (from 0 to η_j) the number of points alternates between odd and even as j varies from 2 to 61.

The governing equations are approximated at the $(N-2)^2$ interior grid points, and at the (i,j)th node the mapping equations are

$$f_{ij}[f_{ij}+\varepsilon(f_x)_{ij}]z^*_{i+1\,j}-2(1+f^2_{ij})z^*_{ij}+f_{ij}[f_{ij}-\varepsilon(f_x)_{ij}]z^*_{i-1\,j}+[1-\varepsilon(f_y)_{ij}/f_{ij}]z^*_{ij+1}+[1+\varepsilon(f_y)_{ij}/f_{ij}]z^*_{ij-1}=0, \tag{15a}$$

$$f_{ij}[f_{ij}+\varepsilon(f_x)_{ij}]\sigma^*_{i+1\,j}-2(1+f^2_{ij})\sigma^*_{ij}+f_{ij}[f_{ij}-\varepsilon(f_x)_{ij}]\sigma^*_{i-1\,j}+[1-\varepsilon(f_y)_{ij}/f_{ij}]\sigma^*_{ij+1}+[1+\varepsilon(f_y)_{ij}/f_{ij}]\sigma^*_{ij-1}=0, \tag{15b}$$

where $\varepsilon=h/2$ and h is the mesh size; $h=1/(N-1)$. Note that the form chosen for the distortion function in equation (2) gives

$$f_x=\pi(1-\beta\cos\pi\eta), \qquad f_y=\beta\pi^2\xi\sin\pi\eta.$$

The equations governing the flow are

$$(f^2_{ij}+\varepsilon q_{1_{ij}}+\varepsilon q_{5_{ij}}\tilde{u}_{\xi_{i+1\,j}})\omega_{i+1\,j}+(q_{3_{ij}}h^2-2f^2_{ij}-2)\omega_{ij},+(f^2_{ij}-\varepsilon q_{1_{ij}}-\varepsilon q_{5_{ij}}\tilde{u}_{\xi_{i-1\,j}})\omega_{i-1\,j}+(1+\varepsilon q_{2_{ij}}+\varepsilon q_{5_{ij}}\tilde{u}_{\eta_{ij+1}})\omega_{ij+1}+(1-\varepsilon q_{2_{ij}}-\varepsilon q_{5_{ij}}\tilde{u}_{\eta_{ij-1}})\omega_{ij-1}=0, \tag{16a}$$

$$(f^2_{ij}+w_{1_{ij}})\psi^*_{i+1\,j}-2(1+f^2_{ij})\psi^*_{ij}+(f^2_{ij}-\varepsilon w_{1_{ij}})\psi^*_{i-1\,j}+(1+\varepsilon w_{2_{ij}})\psi^*_{ij+1}+(1-\varepsilon w_{2_{ij}})\psi^*_{ij-1}+h^2 w_{4_{ij}}=0, \tag{16b}$$

where the coefficients are given by

$$q_{1_{ij}}=f_{ij}(f_x)_{ij}+f^2_{ij}(\sigma_x)_{ij}/\sigma_{ij},$$
$$q_{2_{ij}}=(\sigma_y)_{ij}/\sigma_{ij}-(f_y)_{ij}/f_{ij},$$
$$q_{3_{ij}}=-[f^2_{ij}(\sigma^2_x)_{ij}+(\sigma^2_y)_{ij}]/\sigma^2_{ij},$$
$$q_{5_{ij}}=-\tfrac{1}{2}fRe,$$

and

$$w_{1_{ij}}=f_{ij}(f_x)_{ij}-f^2_{ij}(\sigma_x)_{ij}/\sigma_{ij},$$
$$w_{2_{ij}}=-(f_y)_{ij}/f_{ij}-(\sigma_y)_{ij}/\sigma_{ij},$$
$$w_{4_{ij}}=-3f_{ij}\xi_i\sigma_{ij}[\tfrac{3}{2}f_{ij}\xi_i(\sigma_x)_{ij}+\tfrac{1}{2}\xi_i\sigma_{ij}(f_x)_{ij}+f_{ij}\sigma_{ij}]+h^2_{\eta_{ij}}\sigma_{ij}\omega_{ij}.$$

Discretization of equations (8), (9), (13) and (14) yields another $3(N-2)+1$ algebraic equations, for a total of $4(N-2)^2+3(N-2)+1$ equations. Correspondingly, there are $(N-2)^2$ unknowns for ψ^*_{ij}, ω_{ij}, z^*_{ij} and σ^*_{ij}, for a total of $4(N-2)^2$ interior unknowns; at the interface $\xi=1$, there are $N-2$ unknowns for ω_{Nj}, z^*_{Nj} and σ^*_{Nj}, and the single unknown λ, for a grand total of $4(N-2)^2+3(N-2)+1$ unknowns. For the 61×61 mesh used in this work, there are 14 102 equations and unknowns.

DETAILS OF THE NUMERICAL SCHEME

Application of Newton's method to the solution of the system of non-linear algebraic equations generated in the previous section is straightforward: the vector set of equations is symbolically represented by $\mathbf{G}(\mathbf{x}; Re, W)=\mathbf{0}$, where $\mathbf{x}$ is the vector set of unknowns. If we start with a 'reasonable' initial guess $\mathbf{x}^{(0)}$, then at the $(n+1)$th iteration, or Newton step, the approximation to the solution is

$$\mathbf{x}^{(n+1)}=\mathbf{x}^{(n)}-(\mathbf{A}^{(n)})^{-1}\mathbf{G}^{(n)} \tag{17}$$

where $\mathbf{A}$ is the Jacobian matrix defined by $\mathbf{A}=\partial\mathbf{G}/\partial\mathbf{x}$. The difficulty comes in computing the elements of $\mathbf{A}$.

The elements of the Jacobian were computed by analytically differentiating the algebraic equations (15) and (16) as well as the discrete forms of the boundary conditions (8), (9), (13) and (14) with respect to all of the unknowns. In finite difference formulations such as this, however, an equation evaluated at the (i, j)th node depends only on the unknowns at that node and the node's nearest neighbours. Thus the Jacobian is sparse. The accuracy of the Jacobian elements was checked in two different ways. First, the computer algebra package SMP was used to compare the analytical form of each Frechet derivative. Secondly, numerical values of the elements during a run were computed using the analytic formulae and also by numerical differencing of the algebraic equations (15) and (16).

To achieve a tight band structure in the Jacobian matrix, we order the equations and unknowns in alternating groups of $N-2$, starting at $i, j=2$: for each i, in order of appearance, we write $N-2$ equations for ψ^*, ω, z^* and σ^*. After all of the governing equations (15) and (16) have been written, we include the boundary conditions in the following order: normal stress balance, tangential stress balance, the orthogonality condition on the mapping, and finally the volume constraint. The band structure arising from this ordering is shown in Figure 2. There is a high degree of density in the

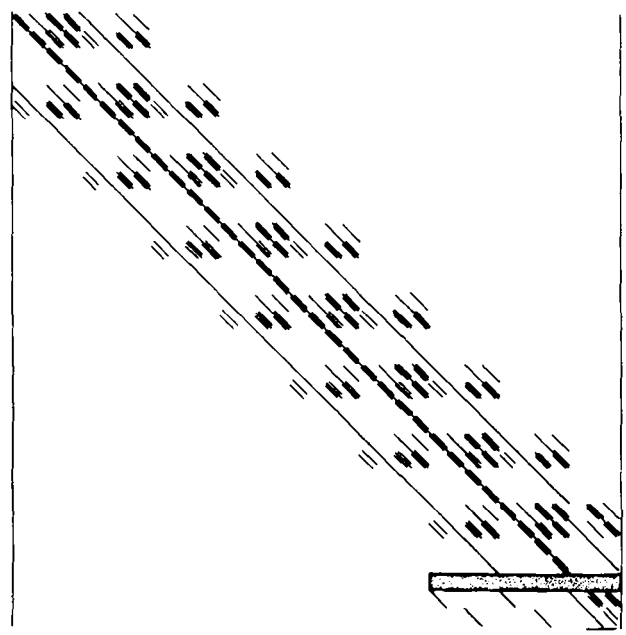

Figure 2. Band structure of the Jacobian matrix

normal stress balance owing to the integral terms. In this matrix of order 14 102 we have a bandwidth of 708.

Solution of the linear system (17) requires the use of a sparse matrix package that efficiently orders and stores the elements of the Jacobian and also provides a fast, accurate solution. We have examined four such sparse matrix library packages: SPARSPAK,[15] SMPAK (Yale), ILUPACK[16] and Harwell MA32. Both SPARSPAK and SMPAK proved unusable because neither makes any provision for pivoting, which is crucial for this type of problem. They also assume that the matrix is symmetric and positive definite, and this acts to reduce the efficiency of storage and elimination. ILUPACK was designed to efficiently solve large sparse unsymmetric linear systems by conjugate gradient-type iterative methods; additionally, ILUPACK uses incomplete LU factorization extensions to SPARSPAK. There are seven possible conjugate gradient methods for the user to choose from.

1. The ORTHOMIN(K) method of Vinsome,[17] in which K is an integer denoting the dimension of the subspace used for approximation.
2. The minimum residual method[18] is a simple descent method.
3. The generalized conjugate residual method GCR(K) is a restarted version of *ORTHOMIN*.
4. The generalized minimum residual method GMRES(K)[19] uses orthogonality of the basis vectors to construct a solution which minimizes the residual norm over the subspace.
5. USYMLQ[20] uses a subspace for computing approximate solution vectors which is built by using multiplications of the matrix and its transpose in turn. This method minimizes the Euclidean norm of the error over the subspace.
6. USYMQR is like USYMLQ except that it minimizes the residual norm over the subspace rather than the Euclidean norm.
7. LSPR[21] uses conjugate gradients applied to the normal equations.

We found that GMRES(K) out-performed all of the other six methods in tests run on our system. In actuality, only two of the other methods would even work on our problems, ORTHOMIN(K) and GCR(K); the other methods either 'blew up' or converged to incorrect finite solutions. One Newton step, using GMRES(K) as the solver, took 5·1 CPU seconds on a CRAY X-MP/24.* This time includes evaluation of the matrix elements and right-hand side, and all calls to ILUPACK. One disadvantage of ILUPACK is that it is an in-core matrix solver. Although the package efficiently orders and stores the elements of the sparse matrix, the workspace required by ILUPACK is nonetheless high—almost 3·5 million words for each problem. This is most of the core memory of the X-MP/24, and to use more than half is very expensive (since it effectively takes memory away from the second processor).

The package with which we have had the most success is MA32 from Harwell. MA32 solves large sparse systems by the frontal method, optionally using disc storage for the matrix factors; it is a direct solver employing partial pivoting. Aside from the obvious strength of being able to solve *very* large problems in a specified and relatively small amount of core storage, MA32 was written so that equations can be input by the user in one of two ways: (i) by elements, as is natural in finite element calculations, or (ii) by a row at a time, as is natural when using finite differencing. During elimination, the number of variables needed in core (or in the 'front') at any time is dependent on the bandwidth of the matrix, provided that the matrix has some regular pattern structure. MA32 accepts input by equation, and will commence elimination on a variable when that variable is fully

* The CRAY X-MP/24 was located at Boeing Computer Services.

summed, that is, the variable is available for use as a pivot in Gaussian elimination when it has appeared for the last time and does not occur in future equations.

We have found that for the mapping, flow and full problems the minimum front size allowable is 313×469. For this front size, the disc space needed to store the UQ decomposition (in double precision words) is 3 068 928. The amount of real work space needed by MA32 varies from 250 000 to 500 000, depending on the front size and output buffer. All of the tests with MA32 have been performed on a CRAY X-MP/48.* Execution times for our cases are approximately 22·5 CPU seconds per Newton step. Unfortunately, an SSD (solid state storage device) is not available, so the temporary files are created on the local discs. Access to SSD for temporary storage would increase the I/O speed by a factor of 100–1000.

The convergence criteria that we have been using for the runs reported below is very simple; we require that the magnitudes of the elements of the residual vector **G** (that is, the vector of equations) satisfy the following:

$$e_\infty = \max_i G_i \leq \beta_\infty, \qquad e_2 = \frac{1}{M}\left(\sum_{i=1}^{M} G_i^2\right)^{1/2} \leq \beta_2.$$

We have found that reasonable values for these tolerances are $\beta_\infty = 10^{-8}$ and $\beta_2 = 10^{-10}$. Values of e_∞ versus Newton iteration for all of the cases presented here are found in Table I. It is apparent that the Newton scheme presented here achieves quadratic convergence.

One disadvantage of MA32—compared to the other sparse matrix libraries tested—is the necessity to give MA32 the locations of the non-zero elements twice, once for the package to determine when each variable is fully summed, and again for MA32 to perform the eliminations. Additionally, this process must be repeated twice for each Newton step. In other words, in libraries like SPARSPAK and ILUPAK, the user must give the package the non-zero indices only when the structure of the matrix changes, and for problems such as ours, this means we only have to give

Table I. The maximum norm e_∞ of the residual vector as a function of Newton iteration, for all of the results presented in this paper

	e_∞			
Re, *W*	1 step	2 steps	3 steps	4 steps
0·5, 0·5	$3{\cdot}8 \times 10^{-2}$	$1{\cdot}5 \times 10^{-3}$	$3{\cdot}1 \times 10^{-6}$	$1{\cdot}1 \times 10^{-11}$
2, 0·5	$4{\cdot}1 \times 10^{-3}$	$2{\cdot}0 \times 10^{-5}$	$2{\cdot}8 \times 10^{-10}$	$7{\cdot}2 \times 10^{-16}$
2, 2	$2{\cdot}2 \times 10^{-4}$	$6{\cdot}5 \times 10^{-8}$	$1{\cdot}1 \times 10^{-14}$	—
10, 1	$2{\cdot}3 \times 10^{-2}$	$5{\cdot}0 \times 10^{-4}$	$2{\cdot}9 \times 10^{-7}$	$1{\cdot}0 \times 10^{-13}$
10, 3	$8{\cdot}8 \times 10^{-4}$	$8{\cdot}2 \times 10^{-7}$	$7{\cdot}4 \times 10^{-13}$	$4{\cdot}3 \times 10^{-15}$
10, 8	$2{\cdot}1 \times 10^{-3}$	$5{\cdot}5 \times 10^{-6}$	$3{\cdot}6 \times 10^{-11}$	$6{\cdot}2 \times 10^{-13}$
50, 4	$2{\cdot}6 \times 10^{-2}$	$6{\cdot}7 \times 10^{-4}$	$5{\cdot}2 \times 10^{-7}$	$6{\cdot}0 \times 10^{-12}$
50, 5	$4{\cdot}2 \times 10^{-3}$	$4{\cdot}4 \times 10^{-5}$	$6{\cdot}9 \times 10^{-9}$	$7{\cdot}5 \times 10^{-12}$
50, 6	$5{\cdot}1 \times 10^{-3}$	$5{\cdot}6 \times 10^{-5}$	$8{\cdot}2 \times 10^{-9}$	$1{\cdot}2 \times 10^{-11}$
50, 8	$4{\cdot}8 \times 10^{-3}$	$5{\cdot}3 \times 10^{-5}$	$2{\cdot}3 \times 10^{-8}$	$4{\cdot}1 \times 10^{-11}$

* The CRAY X-MP/48 is located at the San Diego Supercomputer Center (SDSC).

the structure once, at the beginning of the run. However, MA32 loses the index information between each solution, so that the index information must be given to MA32 twice for each Newton step. Aside from the increased computational cost, this behaviour almost doubles the amount of Fortran source code, appreciably increasing the compilation time. Except for this inconvenience, MA32 has proved to be a versatile sparse matrix package.

CONTINUATION

Continuation can play a vital role in the application of Newton's method, particularly when the solution behaviour changes dramatically with small changes in one or more of the parameters. One of the simplest forms of continuation is Euler–Newton continuation,[22,23] where application of the chain rule from differential calculus generates a vector initial value problem (IVP). The independent variable in this linear IVP is the continuation parameter, and the coefficient matrix of the right-hand side is the Jacobian matrix **A**. The Euler–Newton algorithm breaks down when **A** is singular, that is, at a branch point. Because of this limitation, arc length continuation is used. The following is based on the continuation algorithm presented by Kubíček and Marek[24] and closely parallels the method developed by Rheinboldt.[25] We wish to generate a complete dependence $\mathbf{x}(Re)$ or $\mathbf{x}(W)$ which forms a continuous smooth curve in $(M+1)$-dimensional space $(x_1, x_2, \ldots, x_M, Re \text{ or } W)$. This continuation method gets its name because the arc length s of the solution curve is employed as a parameter of the method. First, the residuals are differentiated with respect to the arc length s

$$\frac{dG_i}{ds}=\frac{\partial G_i}{\partial Re}\frac{dRe}{ds}+\sum_{j=1}^{M}\frac{\partial G_i}{\partial x_j}\frac{\partial x_j}{ds}=0 \tag{18}$$

if continuation is performed in Re, and

$$\frac{dG_i}{ds}=\frac{\partial G_i}{\partial W}\frac{dW}{ds}+\sum_{j=1}^{M}\frac{\partial G_i}{\partial x_j}\frac{dx_j}{ds}=0 \tag{19}$$

if W is considered. With the addition of arc length s into the set of equations, one more constraint on the system is needed. This extra condition specifies the normalized length of the curve in solution space. Thus we require

$$\left(\frac{dx_1}{ds}\right)^2+\left(\frac{dx_2}{ds}\right)^2+\ldots+\left(\frac{dx_M}{ds}\right)^2+\left(\frac{dRe}{ds}\right)^2=1 \tag{20}$$

if we are again considering continuation in Re for example. Now, equation (18) forms a set of M linear algebraic equations in the $M+1$ unknowns dx_i/ds, $i=1,\ldots,M+1$, where $x_{M+1}\equiv Re$. The number of unknowns in equation (18) can be reduced by one if a new set of unknowns κ_i is defined such that

$$\frac{dx_i}{ds}=\kappa_i\frac{dx_k}{ds}, \quad i=1,2,\ldots,k-1,k+1,\ldots,M+1, \tag{21}$$

for some $\partial x_k/\partial s\neq 0$, $1\leq k\leq M+1$. Equation (18) then reduces to a system of M equations and M

unknowns, which can be solved for the κ_i. The coefficient matrix of this linear system is

$$\hat{\mathbf{A}} = \begin{pmatrix} \partial G_1/\partial x_1 & \cdots & \partial G_1/\partial x_{k-1} & \partial G_1/\partial x_{k+1} & \cdots & \partial G_1/\partial x_{M+1} \\ \partial G_2/\partial x_1 & \cdots & \partial G_2/\partial x_{k-1} & \partial G_2/\partial x_{k+1} & \cdots & \partial G_2/\partial x_{M+1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \partial G_M/\partial x_1 & \cdots & \partial G_M/\partial x_{k-1} & \partial G_M/\partial x_{k+1} & \cdots & \partial G_M/\partial x_{M+1} \end{pmatrix}.$$

The value of k was chosen using trial-and-error and was held fixed for the duration of a computation. This matrix is, in general, non-singular at limit points, making arc length continuation a valuable means of branching through limit points. With the values of κ_i, $\mathrm{d}x_k/\mathrm{d}s$ can be computed using equation (20). These equations comprise a set of $M+1$ coupled differential equations which can be solved by a variety of methods.

We have tested the arc length continuation algorithm and found that it works well in the regions of parameter space considered (moderate Re and W). Specifically, with the initial guess generated using continuation, we were always able to obtain a solution at the new parameter value with Newton's method.

DISCUSSION

To demonstrate the performance and accuracy of Newton's method, we have obtained results at Reynolds numbers of 0·5, 2, 10 and 50 for a variety of Weber numbers. The results for $Re=0{\cdot}5$, $W=0{\cdot}5$ may be seen in Figure 3. At these low values of Re and W, deformation is very slight, and the flowfield is nearly fore–aft symmetric. (The flow is from left to right around the bubble.) The upper figure corresponds to lines of constant vorticity, while the lower figure shows the

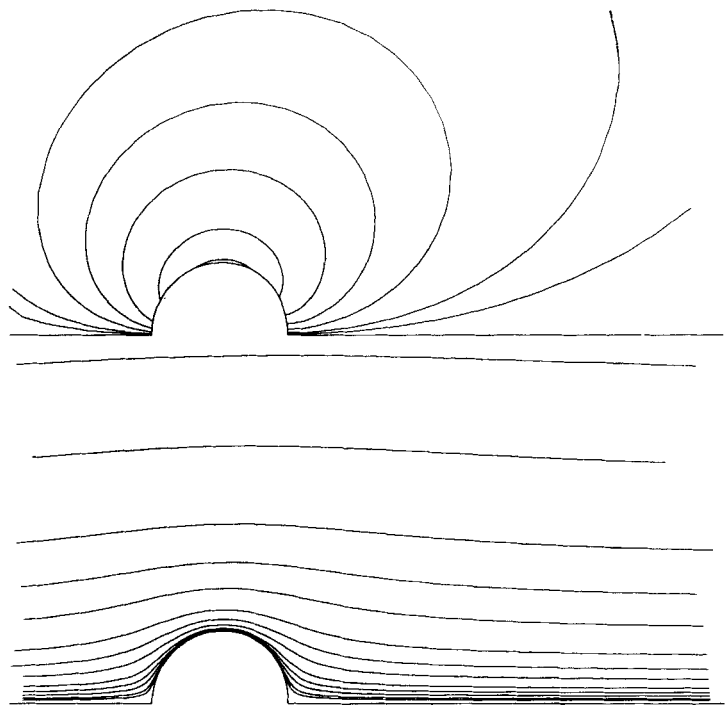

Figure 3. Vorticity lines and streamlines at $(R, W)=(0{\cdot}5, 0{\cdot}5)$. Flow is left-to-right around the bubble

Table II. Comparison of the present results for the drag coefficient C_D with the results of Ryskin and Leal and with the low-Re, W asymptotic results of Taylor and Acrivos

	C_D		
Re, W	Present work	Ryskin and Leal[5]	Taylor and Acrivos[20]
0·5, 0	33·6	33·6	33·7
0·5, 0·5	34·7	34·7	35·1
2, 0·5	9·62	9·62	—
2, 2	10·6	10·6	—
10, 1	2·66	2·67	—
10, 3	3·18	—	—
10, 8	4·02	4·00	—
50, 4	1·22	1·23	—
50, 5	1·40	—	—
50, 6	1·65	1·64	—
50, 8	2·20	2·18	—

streamlines. To obtain this result, the following initial condition was used:

$$z_{ij}^{*}=\xi_i\cos(\pi\eta_j), \qquad \sigma_{ij}^{*}=\xi_i\sin(\pi\eta_j), \quad 1\leq i,j\leq N,$$

$$\psi_{ij}^{*}=\omega_{ij}=0, \quad 1\leq i,j\leq N,$$

and

$$\lambda=1.$$

Convergence to the solution shown in Figure 3 was reached in four Newton steps. In Table II we present a comparison of drag coefficients computed in this work with the low-Re, W asymptotic analysis of Taylor and Acrivos[26] and the earlier numerical work of Ryskin and Leal,[5] who used an alternating direction implicit technique to compute solutions to this problem. Table II also summarizes all comparisons made between results of this work and Ryskin and Leal for all values of Re and W considered.

Figures 4 and 5 show results for $Re=2$ and 10, and $0{\cdot}5\leq W\leq 8$. As W increases so does deformation, and the bubble forms a dimple on the downstream side at $(Re, W)=(10, 8)$. The initial condition for $(Re, W)=2, 0.5)$ was obtained by taking the solution for the case (0·5, 0·5) and performing continuation in Re; The initial guess at (10, 0.5) was obtained by starting with the solution at (2, 0.5). Similarly, the initial guess for (2, 2) was found by starting with the solution at (2, 0·5) and performing continuation in W.

Figure 6 shows results for $Re=50$ and $W=4$, 5, 6 and 8. At this Reynolds number, as deformation increases with W, the upstream side of the bubble deforms, essentially to the same extent as the downstream side, causing the bubble to become oblate ellipsoidal in shape. An attached recirculating wake forms for $4<W<5$, and the dimensions of this wake grow larger as the bubble deformation increases. The presence of an attached eddy is accompanied by a thin region of oppositely signed vorticity.

We have demonstrated with these examples that Newton's method works well for free-surface flow problems when applied to solving the non-linear algebraic equations arising from finite difference approximations to the Navier–Stokes equations and their boundary conditions. Although the system of equations generated is quite large, they comprise a sparse system when linearized and are well suited for solution on large vector machines.

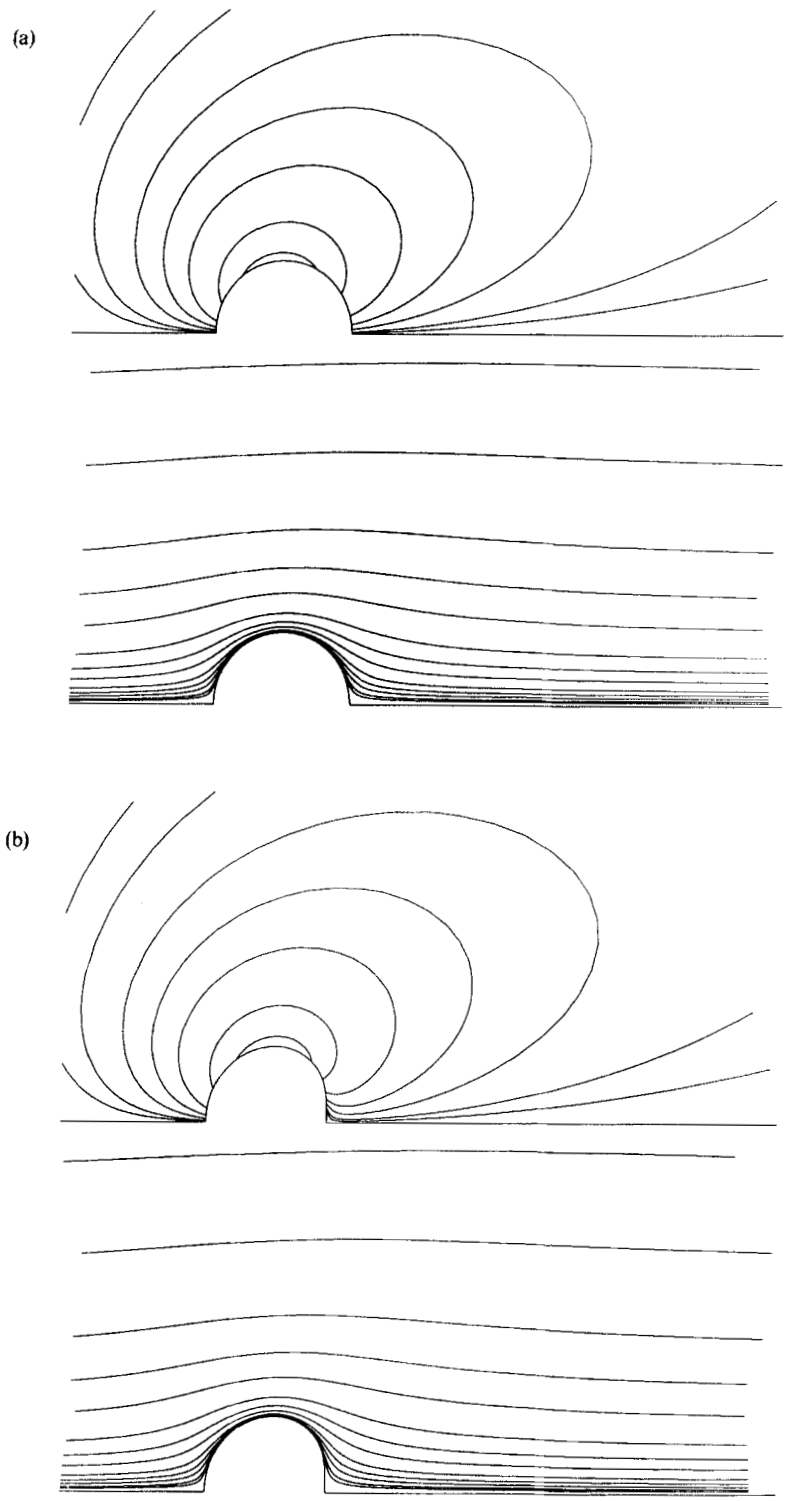

Figure 4. Vorticity lines and streamlines at $(R, W)=(2, 0{\cdot}5)$ and $(2, 2)$

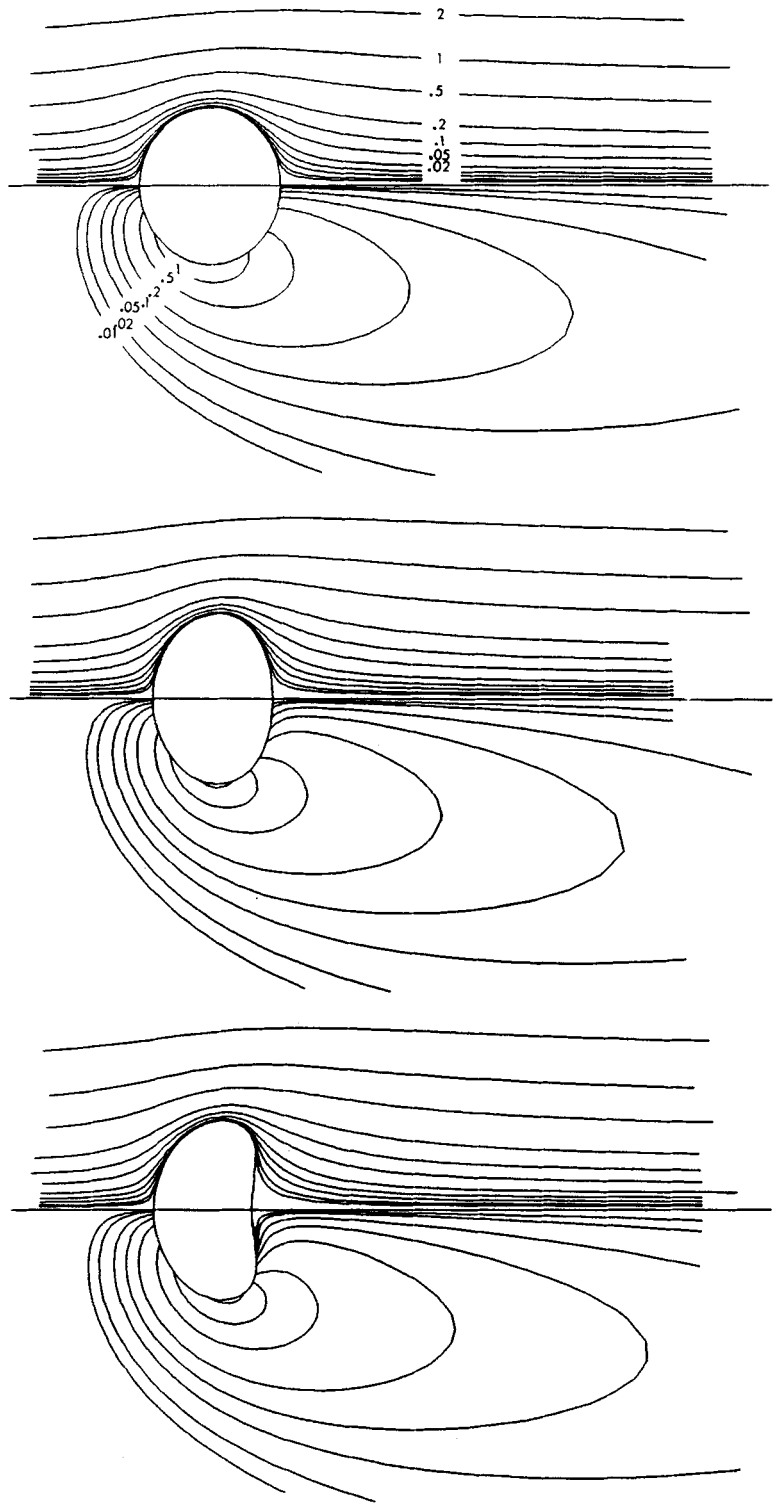

Figure 5. Vorticity lines and streamlines at $(R, W)=(10, 1)$, $(10, 3)$ and $(10, 8)$

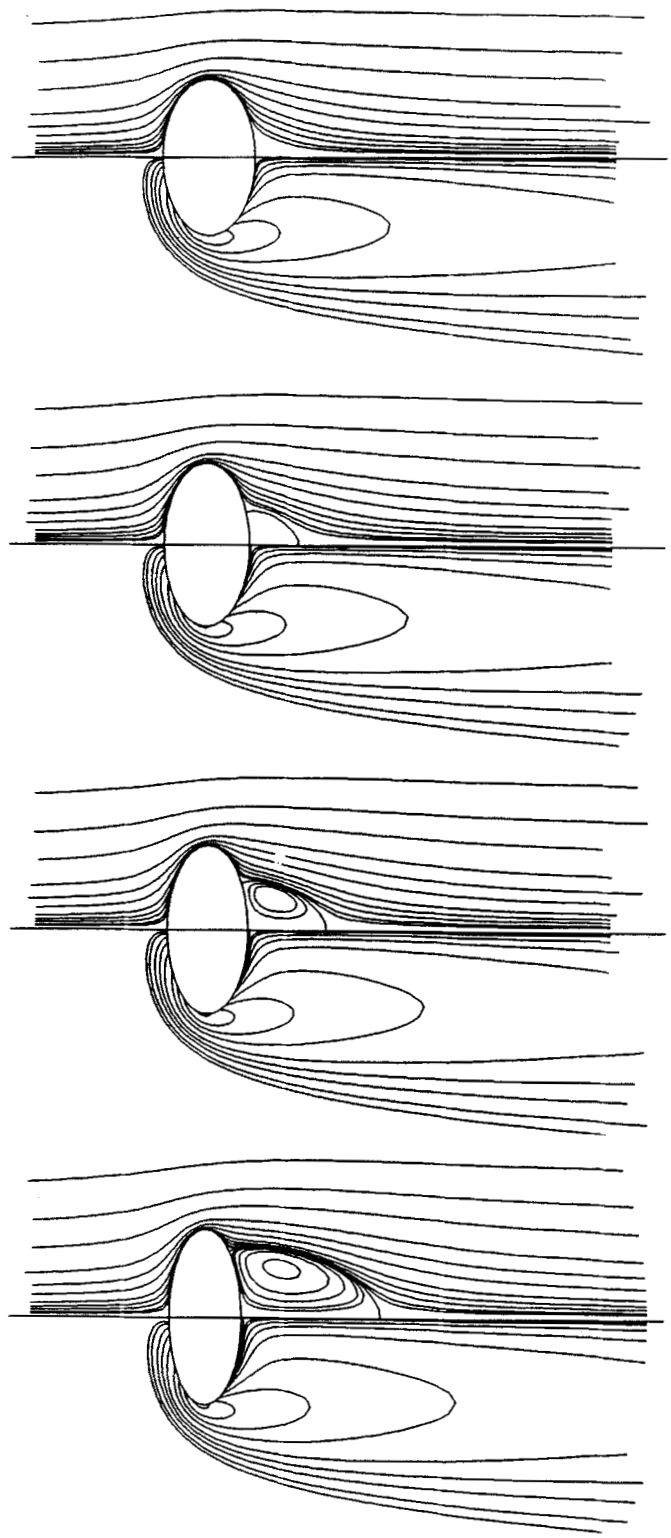

Figure 6. Vorticity lines and streamlines at $(R, W) = (50, 4)$, $(50, 5)$, $(50, 6)$ and $(50, 8)$

ACKNOWLEDGEMENTS

Computations using the library package ILUPACK were carried out on Boeing Computer Services' CRAY X-MP/24, and the authors wish to acknowledge the expert help and time spent by Boeing employee Dr. Horst Simon.

Computations using the Harwell library were carried out on the CRAY X-MP/48 at San Diego Supercomputer Center. We would like to thank Dr. Bob Leary for his help in getting started with MA32.

The authors wish to thank Prof. In Seok Kang for his suggestions regarding use of the strong constraint method in this problem.

This work was supported by grants from the Fluid Mechanics Program and the Office of Advanced Scientific Computing at the National Science Foundation.

REFERENCES

1. J. G. Hnat and J. D. Buckmaster, 'Spherical cap bubbles and skirt formation,' *Phys. Fluids* **19,** 182–194 (1976).
2. A. S. Geller, S. H. Lee and L. G. Leal, 'The creeping motion of a spherical particle normal to a deformable interface,' *J. Fluid Mech.* **169,** 27–69 (1986).
3. M. Miksis, J.-M. Vanden-Broeck and J. B. Keller, 'Axisymmetric bubble or drop in a uniform flow,' *J. Fluid Mech.* **108,** 89–100 (1981).
4. J. F. Thompson, Z. U. A. Warsi and C. W. Mastin *Numerical Grid Generation: Foundations and Applications.* Elsevier, 1985.
5. G. Ryskin and L. G. Leal, 'Large deformations of a bubble in axisymmetric steady flows. Part 2. The rising bubble,' *J. Fluid Mech.* **148,** 19–35 (1984).
6. C. I. Christov and P. K. Volkov, 'Numerical investigation of the steady viscous flow past a stationary deformable bubble,' *J. Fluid Mech.* **158,** 341–364 (1985).
7. D. S. Dandy and L. G. Leal, 'Buoyancy-driven motion of a deformable drop through a quiescent liquid at intermediate Reynolds numbers,' *J. Fluid Mech.*, in press (1989).
8. G. Ryskin and L. G. Leal, 'Orthogonal mappping,' *J. Comput. Phys.* **50,** 71–100 (1983).
9. G. Ryskin and L. G. Leal, 'Large deformations of a bubble in axisymmetric steady flows. Part 3. Uniaxial extensional flow,' *J. Fluid Mech.* **148,** 36–54 (1984).
10. I. S. Kang and L. G. Leal, 'Numerical solution of axisymmetric, unsteady, free-boundary problems at finite Reynolds number. I. Finite-difference scheme and its application to the deformation of a bubble in a uniaxial straining flow,' *Phys. Fluids* **30** (7), 1929–1940 (1987).
11. I. S. Kang and L. G. Leal, 'Numerical solution of axisymmetric, unsteady, free-boundary problems at finite Reynolds number. II. Deformation of a bubble in a biaxial straining flow,' *Phys. Fluids*, in press (1989).
12. D. W. Peacemann and H. H. Rachford, 'The numerical solution of parabolic and elliptic differential equations,' *J. Soc. Indust. Appl. Math.* **3,** 28–41 (1955).
13. H. M. Ettouney and R. A. Brown, 'Finite-element methods for steady solidification problems,' *J. Comput. Phys.* **49,** 118–150 (1983).
14. S. F. Kistler and L. E. Scriven, 'Coating flow theory by finite element and asymptotic analysis of the Navier–Stokes system,' *Int. j. numer. methods fluids* **4,** 207–229 (1984).
15. A. George, J. Liu and E. *Ng User Guide for SPARSPAK: Waterloo Sparse Linear Equations Package*, University of Waterloo, Waterloo, Ontario, Canada 1980.
16. H. D. Simon *User Guide for ILUPACK: Incomplete LU Factorization and Iterative Methods. Extensions to SPARSPAK.*, Boeing Computer Services, Tukwila, WA 1985.
17. P. K. W. Vinsome, 'Orthomin, an iterative method for solving sparse sets of simultaneous linear equations,' *SPE Proceedings of the Fourth Symposium on Reservoir Simulation*, 149–159 (1976).
18. H. C. Elman, 'Iterative methods for large, sparse, nonsymmetric systems of linear equations,' *Research Report No. 229,* Yale University, Dept. of Computer Sci. 1982.
19. Y. Saad and M. H. Schultz, 'GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,' *Research Report No. 254,* Yale University, Dept. of Computer Sci. 1983.
20. M. A. Saunders, H. D. Simon and E. L. Yip, 'Two conjugate-gradient-type methods for sparse unsymmetric linear equations,' *Report ETA-TR-18,* Boeing Computer Services 1984.
21. C. C. Paige and M. A. Saunders, 'LSQR: An algorithm for sparse linear equations and sparse least squares,' *ACM TOMS* **8,** 43–71 (1982).
22. R. Schreiber and H. B. Keller, 'Driven cavity flows by efficient numerical techniques,' *J. Comput. Phys.* **49,** 310–333 (1983).

23. H. B. Keller *Applications in Bifurcation Theory*, Academic Press, New York 1977.
24. M. Kubíček and M. Marek *Computational Methods in Bifurcation Theory and Dissipative Structures*, Springer-Verlag, New York 1983.
25. W. Rheinboldt, 'Solution fields of nonlinear equations and continuation methods,' *SIAM J. Numer. Anal.* **17,** 221–237 (1980).
26. T. D. Taylor and A. Acrivos, 'On the deformation and drag of a falling viscous drop at low Reynolds number,' *J. Fluid Mech.* **18,** 466–476 (1964).