AN IMPLICIT SURFACE TENSION ALGORITHM FOR PICARD SOLVERS OF SURFACE-TENSION-DOMINATED FREE AND MOVING BOUNDARY PROBLEMS

P. J. SLIKKERVEER, E. P. VAN LOHUIZEN AND S. B. G. O'BRIEN*

Philips Research Laboratories, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands

SUMMARY

One of the methods for solving a free or moving boundary problem is the use of Picard solvers which solve the geometry and the velocity field successively. When, however, the kinematic condition is used for updating the geometry in this technique, numerical stability problems occur for surface-tension-dominated flow. These problems are shown here to originate from the unstable integration of the local smoothing of the surface by surface tension. By an extension of the surface tension contribution to the flow field an implicit treatment of surface tension is obtained which overcomes these stability problems. The algorithm is applicable to both free and moving boundary problems, as will be shown by examples in this paper.

KEY WORDS surface tension; free boundary, moving boundary; implied algorithm

1. INTRODUCTION

Many industrial processes contain a combination of fluid flow and free or moving fluid surfaces. Examples of such processes are coating flows and viscous sintering. Since these fields are interesting both commercially and mathematically, many techniques have been used for their numerical simulation (see e.g. Reference 1). A limited number of algorithms has found their way into commercially available simulation software. Floryan and Rasmussen[2] give an overview of these algorithms. They note that for fluid surfaces with surface tension it is important to describe the surface geometry precisely. For this, two methods perform best, namely the updated Euler method and the arbitrary Lagrange Euler (ALE) method. These methods are in fact closely related to each other. Both methods generate a mesh on the fluid domain and adapt it to the change in surface geometry. The methods become equal for linear flow problems such as our examples where the time dependence is limited to the change in geometry. Two solution methods are used in this class of numerical methods, both having their merits.[2,3]

1. In the first of these methods, all unknowns, both velocity components and surface position, are combined in one vector.[3,5,6] This algorithm enables the use of a Newton iteration scheme for free boundaries. The implementation of this linearization involves a considerable amount of work, which results in a high order of convergence but a limited robustness.[3]
2. The second type of algorithm uses a Picard solving technique, in which the velocity components and the surface position are computed successively. This method is easier to implement since it does not require adaptation of the flow elements. It is more robust, but at the cost of a linear convergence for free surfaces.

* Department of Mathematics, University of Limerick, Limerick, Ireland.

CCC 0271–2091/96/090851–15

Received 19 August 1994
Revised 27 June 1995

Since the second technique is more robust, it is applicable to industrial problems. In free boundary problems which involve surface tension, however, one of two separate calculation schemes must be chosen depending on the nature of the problem; one is limited to surface-tension-dominated flow and the other to flow with only small surface tension effects.[2,3] The technique fails for problems which have a combined character, e.g. with only local surface tension domination. The Picard solution method also fails for moving boundary problems with surface tension domination.

In this paper we will show that the algorithm does not fail in these situations but is in fact hopelessly inefficient. We show that this difficulty can be resolved by an expansion of the surface tension term. To demonstrate this, we consider two-dimensional viscous flow (quasi-static Stokes flow) where surface tension dominates. The solution is worked out for the finite element method. The quasi-static Stokes flow was chosen for our examples, since here the surface tension is the only source of instability in the problem.

First we will introduce the basic differential equations and the Picard solution technique in more detail in the next section. In section 3 we will show that the surface tension problems orginate in unstable integration of the position of the free surface. To arrive at a solution to this problem, we will then derive the weak formulation of the set of differential equations as is implemented in the finite element method. Concentrating on the way the surface tension effect is represented in this formulation, we will derive in Section 5 an expansion to the surface tension contribution, which stabilizes the calculation in these situations. Section 6 will describe briefly the numerical implementation that is used for the calculation of the two examples in Section 7. These examples will show the effectiveness of the adapted surface tension implementation. The results will be discussed in Section 8.

2. DIFFERENTIAL EQUATIONS AND PICARD ITERATION SCHEME

Here we will consider the flow of an incompressible viscous liquid, which can be described mathematically with the combination of the continuity equation and the Stokes equations, together with the appropriate boundary conditions. The differential equations are

$$\begin{gathered} \boldsymbol{\nabla}\cdot\mathbf{u}=0, \\ \boldsymbol{\nabla}p-\boldsymbol{\nabla}\cdot\{\mu[(\boldsymbol{\nabla}\mathbf{u})+(\boldsymbol{\nabla}\mathbf{u})^{\mathrm{T}}]\}-\rho\mathbf{f}=\mathbf{0}, \end{gathered} \tag{1}$$

with p the hydrostatic pressure, $\mathbf{u}$ the velocity vector, ρ the specific mass and $\mathbf{f}$ an external body force such as gravity.

For the boundary conditions we will concentrate on the free/moving boundary. Here the stress is prescribed on the boundary and is supplemented by the so-called kinematic condition. The prescribed surface stress consists of two parts, namely a normal stress component originating from the atmospheric pressure (p_a) and the surface tension on a curved surface ($\gamma\boldsymbol{\nabla}_s\cdot\mathbf{n}$, where $\boldsymbol{\nabla}_s\cdot\mathbf{n}$ denotes the curvature of the surface and $\boldsymbol{\nabla}_s$ the surface gradient operator as defined in Appendix I) and a tangential stress component in the case of a surface tension gradient occurring along the surface ($\boldsymbol{\nabla}_s\gamma$):

$$\boldsymbol{\sigma}_{\mathrm{n}}=-(p_{\mathrm{a}}+\gamma\boldsymbol{\nabla}_{\mathrm{s}}\cdot\mathbf{n})\mathbf{n}, \tag{2a}$$

$$\boldsymbol{\sigma}_{\mathrm{t}}=\boldsymbol{\nabla}_{\mathrm{s}}\gamma. \tag{2b}$$

The kinematic condition is used to describe the movement of the free surface:

$$\frac{\mathrm{d}\mathbf{x}^{\Gamma}}{\mathrm{d}\tau}=\mathbf{u}^{\Gamma}, \tag{3}$$

where $\mathbf{x}^{\Gamma}$ and $\mathbf{u}^{\Gamma}$ denote the position of the free/moving boundary and the velocity at this boundary respectively and τ denotes the time. For a converged solution of the free surface this reduces to

$$\mathbf{u}^{\Gamma} \cdot \mathbf{n} = 0. \tag{4}$$

This set of equations can be solved using a Picard solution technique. As stated in the previous section, two schemes exist for solving this set of equations. For flows where surface tension is not dominant, the following algorithm is used.

1. Generate a mesh on the initial geometry (Ω_0, with Γ_0 being the boundary of Ω_0) and compute the velocity field on this geometry.
2. Compute the new geometry (Ω_τ) by updating the co-ordinates at the free surface with the aid of (3) using $\mathbf{u}_{\tau-\Delta\tau}$ and regenerating the mesh:

$$\mathbf{x}_{\tau}^{\Gamma} = \mathbf{x}_{\tau-\Delta\tau}^{\Gamma} + \Delta\tau\, \mathbf{u}_{\tau-\Delta\tau}^{\Gamma}. \tag{5}$$

3. Calculate the velocity field ($\mathbf{u}_\tau$) using equation (1) at Ω_τ with boundary conditions (2) at the free boundary. Γ_τ is used in the calculation of the free surface curvature in equation (2a).
4. Repeat steps 2 and 3 for each time step.

This algorithm is called the kinematic update scheme because of its use of the kinematic condition for updating the surface geometry. A second scheme is used when surface tension dominates. This is called the normal stress update since it prescribes the kinematic condition (equation (4)) as a boundary condition to the flow field and uses the normal stress (equation (2a)) for the geometrical update.

The two schemes are necessary since the kinematic update is found to be unstable for surface-tension-dominated flow.[2,3] In its turn the normal stress update algorithm has been found unstable for flow with small surface tension effects.

For moving boundary problems there is no analogy for the normal stress update algorithm, since equation (4) is only valid for free boundaries and the form (3) of the kinematic condition cannot be used as a boundary condition because $\mathbf{u}^{\Gamma}$ is unknown.

3. ORIGIN OF STABILITY PROBLEMS FOR SURFACE-TENSION-DOMINATED FLOW

The origin of the instability in the kinematic update scheme for surface-tension-dominated flows is best illustrated by studying the update algorithm of the free surface, equation (5). When surface tension dominates, the shape of the surface determines the velocity field, so we may write schematically

$$\mathbf{u}_{\tau-\Delta\tau}^{\Gamma} = \mathbf{u}(x_{\tau-\Delta\tau}^{\Gamma}). \tag{6}$$

With this, equation (5) becomes

$$\mathbf{x}_{\tau}^{\Gamma} = \mathbf{x}_{\tau-\Delta\tau}^{\Gamma} + \Delta\tau\, \mathbf{u}(x_{\tau-\Delta\tau}^{\Gamma}). \tag{7}$$

This equation shows the explicit Euler algorithm that is used for updating the free/moving surface geometry.[4] This explicit algorithm is known to be conditionally stable: the time step must be a factor smaller than the smallest timescale present in the physical problem.

This brings us to the real problem. Using dimensional analysis, we estimate the timescale (τ_{smooth}) correlated with the smoothing of a sinusoidal wave on the surface with a wavelength λ and an amplitude a by surface tension forces to be

$$\tau_{\text{smooth}} \approx \frac{\mu R}{\gamma} = \frac{\mu}{\gamma}\frac{\lambda^2}{4\pi^2 a}, \tag{8}$$

with R being the curvature of the surface.

This result indicates that the timescale for smoothing goes to zero with the wavelength of the disturbance. The physical problem thus has as a minimum timescale $\tau_{\text{smooth}} = 0$.

In the numerical problem this is not the case, since the minimum wavelength of a disturbance equals that of the mesh size. However, since the physical problem usually contains a second timescale (e.g. that of the macroscopic change in the geometry), the numerical problem can become extremely stiff and becomes even stiffer when the mesh size is reduced. In view of the fact that numerical calculations always introduce errors and will do so in the surface geometry too, the smoothing process plays a part in any numerical problem involving surface tension.

It is thus the stiffness of the numerical problem that causes the stability problems in the application of the kinematic condition for surface-tension-dominated flow. For a stable application of this condition in the current algorithm the size of the time steps must be extremely small, which means that a large number of time steps will be required to obtain a result on a macroscopic timescale. Note that the introduction of an iteration per time step (iterative implicit Euler) does not overcome the problem, since the convergence criterion of the iteration per time step equals the stability criterion of the discussed algorithm.

The general solution for the integration of stiff problems is the use of an implicit algorithm. Although it is probably not possible to formulate the complete problem implicitly, we show that the kinematic update algorithm can be adapted in such a way that the surface tension terms are calculated implicitly, as will be shown in Section 4. To be able to do this, we will first have to derive the weak formulation for the flow problem.

4. WEAK FORMULATION OF FLOW EQUATIONS

The weak forms of the differential equations are

$$\begin{gathered}\int_\Omega \Psi_k \nabla\cdot\mathbf{u}\,\mathrm{d}\Omega = 0,\\ \int_\Omega \phi_i \nabla p\,\mathrm{d}\Omega - \int_\Omega \phi_i \nabla\cdot\{\mu[(\nabla\mathbf{u}) + (\nabla\mathbf{u})^{\mathrm{T}}]\}\,\mathrm{d}\Omega - \int_\Omega \phi_i \rho\mathbf{f}\,\mathrm{d}\Omega = \mathbf{0},\end{gathered} \tag{9}$$

where ϕ_i and Ψ_k are the test functions for the momentum equation and continuity equation respectively. After partial integration of the diffusion and the pressure term the momentum equation can be written as (see e.g. Reference 7)

$$\begin{gathered}-\int_\Omega p\nabla\phi_i\,\mathrm{d}\Omega + \int_\Omega \nabla\phi_i\cdot\{\mu[(\nabla\mathbf{u}) + (\nabla\mathbf{u})^{\mathrm{T}}]\}\,\mathrm{d}\Omega - \int_\Omega \phi_i\rho\mathbf{f}\,\mathrm{d}\Omega\\ -\int_\Gamma \left\{-\phi_i p\mathbf{n} + \phi_i\{\mu[(\nabla\mathbf{u}) + (\nabla\mathbf{u})^{\mathrm{T}}]\}\cdot\mathbf{n}\right\}\mathrm{d}\Gamma = 0.\end{gathered} \tag{10}$$

The last integral in this formula is the effect of the surface traction vector and can be written as

$$\int_\Gamma (\sigma_{\mathrm{n}}\mathbf{n} + \boldsymbol{\sigma}_{\mathrm{t}})\phi_i\,\mathrm{d}\Gamma. \tag{11}$$

At the free surface the stress boundary conditions (2) can be substituted in this term to obtain

$$\int_\Gamma (\sigma_{\mathrm{n}}\mathbf{n} + \sigma_{\mathrm{t}}\mathbf{t})\phi_i\,\mathrm{d}\Gamma = \int_\Gamma [-p_{\mathrm{a}}\mathbf{n}\phi_i - \gamma(\nabla_{\mathrm{s}}\cdot\mathbf{n})\mathbf{n}\phi_i]\,\mathrm{d}\Gamma + \int_\Gamma \nabla_{\mathrm{s}}\gamma\phi_i\,\mathrm{d}\Gamma. \tag{12}$$

It is convenient to transform the curvature term in the integral through partial integration using the surface divergence theorem[8]

$$-\int_{\Gamma} \gamma(\nabla_s \cdot \mathbf{n})\mathbf{n}\phi_i \, \mathrm{d}\Gamma = \int_{\partial\Gamma} \gamma\phi_i\mathbf{m} \, \mathrm{d}(\partial\Gamma) - \int_{\Gamma} \nabla_s(\phi_i\gamma) \, \mathrm{d}\Gamma, \tag{13}$$

where $\partial\Gamma$ denotes the boundary of the free/moving surface (Γ) and $\mathbf{m}$ is the outward-pointing normal of the boundary $\partial\Gamma$ of the surface; $\mathbf{m}$ is perpendicular to the normal ($\mathbf{n}$) of the surface and perpendicular to the tangent of the line $\partial\Gamma$ ($\mathbf{ds}$):

$$\mathbf{m} = \mathbf{n} \times \mathbf{ds}. \tag{14}$$

With equation (13) above the last term in equation (10) becomes

$$\int_{\Gamma} (\sigma_n\mathbf{n} + \sigma_t\mathbf{t})\phi_i \, \mathrm{d}\Gamma = -\int_{\Gamma} p_a\mathbf{n}\phi_i \, \mathrm{d}\Gamma - \int_{\Gamma} \nabla_s(\phi_i\gamma) \, \mathrm{d}\Gamma + \int_{\Gamma} \nabla_s\gamma\phi_i \, \mathrm{d}\Gamma + \int_{\partial\Gamma} \gamma\phi_i\mathbf{m} \, \mathrm{d}(\partial\Gamma). \tag{15}$$

Applying the chain rule to the second term on the right-hand side, we obtain for the boundary integral over the free surface

$$\int_{\Gamma} (\sigma_n\mathbf{n} + \sigma_t\mathbf{t})\phi_i \, \mathrm{d}\Gamma = -\int_{\Gamma} p_a\mathbf{n}\phi_i \, \mathrm{d}\Gamma - \int_{\Gamma} \gamma\nabla_s\phi_i \, \mathrm{d}\Gamma + \int_{\partial\Gamma} \gamma\phi_i\mathbf{m} \, \mathrm{d}(\partial\Gamma). \tag{16}$$

5. IMPLICIT SURFACE TENSION CONTRIBUTION

Equation (16) shows that the effect of surface tension is limited to two terms, one containing the effect over the surface and one representing the contact line effects (the last term in equation (16)). We will assume that the contact line contribution is constant (which implies a constant contact angle and a constant surface tension at the contact point). In that case only the surface term changes with the geometry. We will focus on that term from this point on. When we restrict ourselves to two-dimensional Cartesian co-ordinates, this term reduces to

$$-\int_{\Gamma} \gamma\nabla_s\phi_i \, \mathrm{d}\Gamma = -\int_{\Gamma} \gamma\mathbf{t}\frac{\mathrm{d}\phi_i}{\mathrm{d}s} \, \mathrm{d}s. \tag{17}$$

Here $\mathbf{t}$ is the tangent vector to the curve. As we have shown in Section 2, this integral is normally calculated over the current surface geometry, e.g. using $\mathbf{t}(\tau)$. To obtain an implicit surface tension algorithm, we must calculate this term for the new (not yet existing) surface gometry. To this end we develop the term in a Taylor series in time:

$$\left[-\int_{\Gamma} \gamma\frac{\mathrm{d}\phi_i}{\mathrm{d}s}\mathbf{t} \, \mathrm{d}s\right]_{\tau+\Delta\tau} = \left[-\int_{\Gamma} \gamma\frac{\mathrm{d}\phi_i}{\mathrm{d}s}\mathbf{t} \, \mathrm{d}s\right]_{\tau} - \left[\frac{\mathrm{d}}{\mathrm{d}\tau}\int_{\Gamma} \gamma\frac{\mathrm{d}\phi_i}{\mathrm{d}s}\mathbf{t} \, \mathrm{d}s\right]_{\tau}\Delta\tau + O(\Delta\tau^2). \tag{18}$$

Since the tracing algorithm of the surface (equation 5)) is linear, only the first two terms in the series are needed. The first term on the right-hand side is just the contribution of the surface tension at time τ. We will concentrate on the second term on the right-hand side, giving us the effect of a linear update in time of the surface geometry on the surface tension.

Note that the equation above can be written as an integral along a reference curve l which does not change in time:

$$-\int_{\Gamma} \gamma\frac{\mathrm{d}\phi_i}{\mathrm{d}s}\mathbf{t}(s(\tau)) \, \mathrm{d}s = -\int_{\phi} \gamma \, \mathbf{t}(\phi_i, \tau) \, \mathrm{d}\phi_i = -\int_{l} \gamma \, \mathbf{t}(l, \tau)\frac{\mathrm{d}\phi_i}{\mathrm{d}l} \, \mathrm{d}l. \tag{19}$$

In the finite element method this reference curve l can be interpreted as a curve consisting of reference elements. The time dependence of this term now arises via the tangent vector $\mathbf{t}$ and the surface tension itself.

Using this, we find for the first expansion term of the surface tension in equation (18)

$$-\left[\frac{\mathrm{d}}{\mathrm{d}\tau}\int_{\Gamma} \gamma\mathbf{t}\frac{\mathrm{d}\phi_i}{\mathrm{d}s}\,\mathrm{d}s\right]_{\tau}\Delta\tau = -\Delta\tau\int_{l}\left.\frac{\mathrm{d}(\gamma\mathbf{t})}{\mathrm{d}\tau}\right|_{\tau}\frac{\mathrm{d}\phi_i}{\mathrm{d}l}\,\mathrm{d}l. \tag{20}$$

For the change in the tangent vector with time it can be shown that (see Appendix II)

$$\frac{\mathrm{d}\mathbf{t}}{\mathrm{d}\tau} = \left(\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}s}\cdot\mathbf{n}\right)\mathbf{n}. \tag{21}$$

Restricting our attention to the case of a constant surface tension and inserting this in equation (20), we obtain

$$-\gamma\,\Delta\tau\int_{l}\left.\frac{\mathrm{d}\mathbf{t}}{\mathrm{d}\tau}\right|_{\tau}\frac{\mathrm{d}\phi_i}{\mathrm{d}l}\,\mathrm{d}l = -\gamma\,\Delta\tau\int_{l}\left[\left(\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}s}\cdot\mathbf{n}\right)\mathbf{n}\right]_{\tau}\frac{\mathrm{d}\phi_i}{\mathrm{d}l}\,\mathrm{d}l. \tag{22}$$

Of course we can use equation (18) to rewrite this as an integral over the surface at time τ:

$$-\gamma\,\Delta\tau\int_{l}\left[\left(\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}s}\cdot\mathbf{n}\right)\mathbf{n}\right]_{\tau}\frac{\mathrm{d}\phi_i}{\mathrm{d}l}\,\mathrm{d}l = -\gamma\,\Delta\tau\int_{\Gamma}\left(\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}s}\cdot\mathbf{n}\right)\mathbf{n}\frac{\mathrm{d}\phi_i}{\mathrm{d}s}\,\mathrm{d}s. \tag{23}$$

We have thus found the term which accounts for the change in the surface tension contribution over one time step when the surface is updated with the kinematic condition. Note that this term is linear in the velocity vector $\mathbf{u}$. It can be taken up directly to the stiffness matrix.

To show the structure of the term, we will write it out for one element. Using numerical integration

$$\int_{e} f(x)\,\mathrm{d}s \approx \sum_{k=1}^{m} w_k f(x_k), \tag{24}$$

we find for this term

$$-\gamma\,\Delta\tau\int_{e}\frac{\mathrm{d}\phi_i}{\mathrm{d}s}u_{\mathrm{sn}}n_l\,\mathrm{d}s \approx -\gamma\Delta\tau\sum_{k=1}^{m} w_k\left.\frac{\mathrm{d}\phi_i}{\mathrm{d}s}\right|_{X_k}u_{\mathrm{sn}}(x_k)\,n_l(x_k), \tag{25}$$

where $l = 1, 2$ denote the components of the Stokes equation, m is the number of integration points over the element and u_{sn} is an abbreviation for the inner product between $\mathrm{d}\mathbf{u}/\mathrm{d}s$ and $\mathbf{n}$. For u_{sn} the following relation is found using the standard approximation of $\mathbf{u}$ over the element:

$$u_{\mathrm{sn}}(x_k) = \left.\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}s}\right|_{x_k}\cdot\mathbf{n}(x_k) \approx \sum_{j=1}^{q}\left(u_j\left.\frac{\mathrm{d}\phi_j}{\mathrm{d}s}\right|_{x_k}n_1(x_k) + v_j\left.\frac{\mathrm{d}\phi_j}{\mathrm{d}s}\right|_{x_k}n_2(x_k)\right), \tag{26}$$

where q is the number of nodal points in the surface element, e.g. $j = 1, 3$ for a quadratic element, and u_j, v_j and n_1, n_2 are the two components of the velocity vector and the normal vector respectively. The total structure of the term is thus

$$-\gamma\,\Delta\tau\sum_{k=1}^{m} w_k\left.\frac{\mathrm{d}\phi_i}{\mathrm{d}s}\right|_{x_k}u_{\mathrm{sn}}(x_k)\,n_l(x_k) = -\gamma\,\Delta\tau\sum_{j=1}^{q}\sum_{k=1}^{m} w_k\left.\frac{\mathrm{d}\phi_i}{\mathrm{d}s}\right|_{x_k}\left.\frac{\mathrm{d}\phi_j}{\mathrm{d}s}\right|_{x_k}n_l(x_k)[u_j n_1(x_k) + v_j n_2(x_k)]. \tag{27}$$

Since **u** is an unknown in the calculation of the velocity field, this term contributes to the element matrix. This contribution can be understood by using the following schematic representation of the element matrix:

$$\gamma\,\Delta t \begin{array}{c} \left(\begin{array}{c:c} \qquad\qquad & \qquad\qquad \\ \hline \qquad\qquad & \qquad\qquad \end{array}\right) \\ \begin{array}{cc} 1\ldots j\ldots q & \quad 1\ldots j\ldots q \\ p=1 & \quad p=2 \end{array} \end{array} \begin{array}{c} 1\ldots i\ldots q \\ l=1 \\ \\ 1\ldots i\ldots q \\ l=2 \end{array} \begin{pmatrix} u_1 \\ | \\ u_j \\ | \\ u_n \\ v_1 \\ | \\ v_j \\ | \\ v_n \end{pmatrix} \begin{array}{c} p=1 \\ \\ \\ p=2 \end{array} \qquad (28)$$

The component (i, j, p, l) of the contribution to the element matrix is

$$-\gamma\Delta\tau \sum_{k=1}^{m} w_k \left.\frac{\mathrm{d}\phi_i}{\mathrm{d}s}\right|_{x_k} \left.\frac{\mathrm{d}\phi_j}{\mathrm{d}s}\right|_{x_k} n_p(x_k)n_l(x_k). \qquad (29)$$

Bearing in mind the division of the element matrix as sketched in formula (28), the term

$$\left.\frac{\mathrm{d}\phi_i}{\mathrm{d}s}\right|_{x_k} \left.\frac{\mathrm{d}\phi_j}{\mathrm{d}s}\right|_{x_k} \qquad (30)$$

is recognized as a diffusion (viscous) term. It in fact dampens the overestimate of the update of the surface geometry by the explicit Euler algorithm. The term

$$n_p(x_k)n_l(x_k) \qquad (31)$$

limits the damping function to the direction perpendicular to the surface geometry.

6. IMPLEMENTATION

To test the implicit surface tension algorithm, the contribution of the surface tension to the matrix was added to the surface tension element of the Sepran Finite Element Toolbox.[9] The easy access to all parts of the code enabled a quick adaptation to the element.

The flow problems in the next section were computed using the Crouzier–Raviart $P^{2+}P^1$ available from Sepran.[7,9] This is an element with extended quadratic basic functions for the velocity and linear discontinuous basis functions for the pressure. It has the advantage that by an elimination the degrees of freedom at element level can be reduced to 12 for the velocity and one for the constant pressure. The remaining unknown for the pressure is later removed from the matrix using a penalty function.

An update Euler algorithm was used for the changing geometry. After computation of the velocity field the new boundaries were computed. To this end each point on the free/moving surface was updated using equation (5).

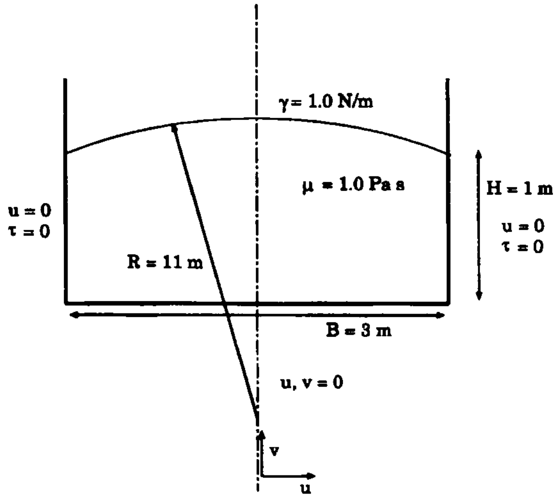

Figure 1. Definition of test problem for free surface flow: smoothing of fluid surface in a container

When required, a cubic spline was fitted through the newly computed co-ordinates. Along this spline the new boundary points were ordered in a suitable manner.

With the newly calculated boundaries a new mesh was created, either by generating a completely new topology or by stretching the old mesh. For the mesh stretching a standard procedure in Sepran was used. In this algorithm the immediate neighbours of each nodal point are identified. Following a Gauss–Seidel type of algorithm, each point is placed successively at the centre of its 'neighbouring' nodes.[9,11] This procedure is repeated until convergence.

7. RESULTS

Example of a free surface problem

To illustrate the effect of the implicit surface tension for free boundaries, we will consider the smoothing by surface tension of a surface in a container filled with fluid (see Figure 1). To prevent numerical problems originating in the singularities at the moving contact points[12] from masking the free surface instability, these singularities were removed by prescribing a slip boundary condition at the vertical walls.

At the initial time the free surface at the top of the fluid is curved. Since we assume a contact angle of 90°, the solution to the physical problem is a trivial horizontal straight line. The stiffness of the problem depends only on the number of elements along the free surface (the ratio of the smoothing timescales of the largest and smallest disturbances of the surface), since all other effects such as gravity have been neglected. As the implicit surface tension algorithm calculates smoothing implicitly, it should allow very large (infinite) time steps.

When the explicit algorithm is used with moderate time steps, wiggles develop in a few steps near the container walls. Figure 2 shows the typical one-up/one-down wiggle that develops near the container walls owing to the unstable integration of the smoothing of the smallest possible disturbances in the free surface. When the implicit surface tension algorithm is used, no such wiggles are present.

The calculations were performed at a constant time step. In Table I we compare the convergence of the two algorithms. For this we compute the deviation of the fluid surface from a straight line using the

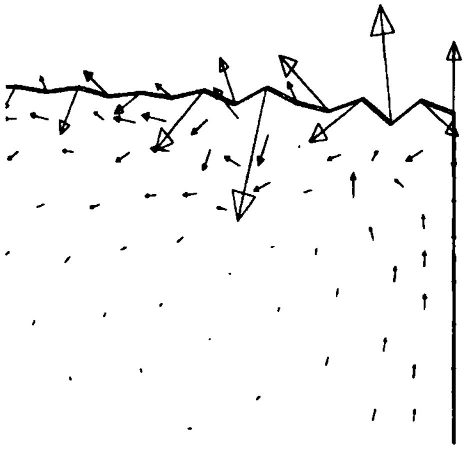

Figure 2. Typical one-up/one-down wiggle developing on unstable integration of surface tension smoothing

L^2 norm. Convergence is reached when this deviation is a factor of 10^{-3} less than that at the start of the computation.

Table I shows the characteristics of the explicit algorithm as discussed in Section 3. The maximum time step is related to the local mesh size, whereas the total time to convergence remains constant since it approximates the physical timescale for this. The implicit algorithm shows no mesh effects. The total time (60 s) does not have a physical meaning.

In the case of the explicit algorithm the time step is limited by the growth of wiggles as shown in Figure 2. The time step in the implicit algorithm is limited by problems near the contact point with the container wall. At large time steps the small tangential movement of the fluid near the surface just pushes nodal points through the wall. There is also some evidence for the development of a tangential motion instability in curved elements near the contact point in the case of very large time steps.

Since the implicit algorithm uses the surface tension contribution at the new time level, the calculated velocity field is affected by the size of the current time step. Figure 3 compares the calculated velocity field at the first time step using the explicit algorithm with that of the implicit algorithm with a time step of 20 s. Note that the vectors of the implicit calculation have been scaled up by a factor of 53 to obtain a comparable size with those of the explicit calculation. This illustrates the effective overestimation that is the basis for problems with the explicit algorithm.

Table I. Comparison of convergences of implicit and explicit algorithms for the free surface example at different mesh sizes

Mesh size	Number of time steps	Size of time step	Size of time step limited by
Explicit			
$h = 0{\cdot}2$	80	0·11	Wiggles
$h = 0{\cdot}1$	127	0·07	Wiggles
$h = 0{\cdot}05$	302	0·03	Wiggles
Implicit			
$h = 0{\cdot}2$	3	20	Contact point problems
$h = 0{\cdot}1$	3	20	Contact point problems
$h = 0{\cdot}05$	3	20	Contact point problems

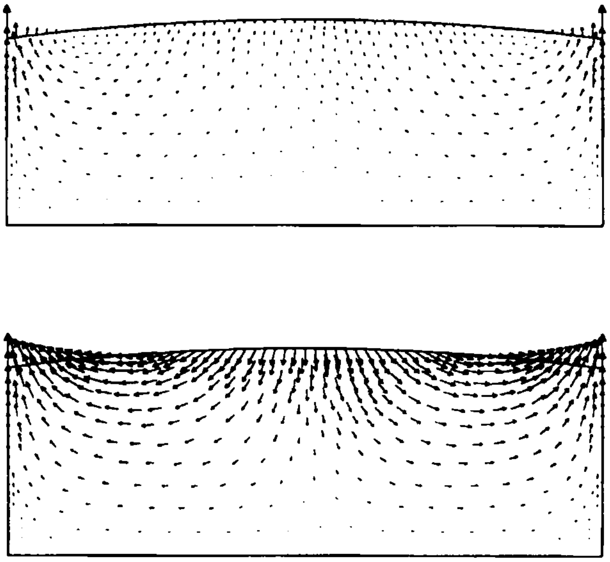

Figure 3. Effect of implicit surface tension term on calculated velocity field: (top) explicit; (bottom) implicit with $t = 20$ s. Note that the vectors of the implicit calculation have been scaled up by a factor of 53 in this figure

Example of a moving surface problem

To illustrate the moving boundary capability, we will consider the development of a two-dimensional drop from a disturbance in a thin fluid layer of water on a vertical wall (see Figure 4). Although this problem has a physical background, the solution does not exist physically, since the solution of the 2D problem is unstable and a 3D solution will develop.[13]

The 2D drop is a typical two-timescale problem. The interesting timescale is that on which the drop is developing under the influence of gravity, whereas the smoothing of local disturbances constitutes the second timescale. In the global formation of the drop, surface tension plays an unimportant role, but

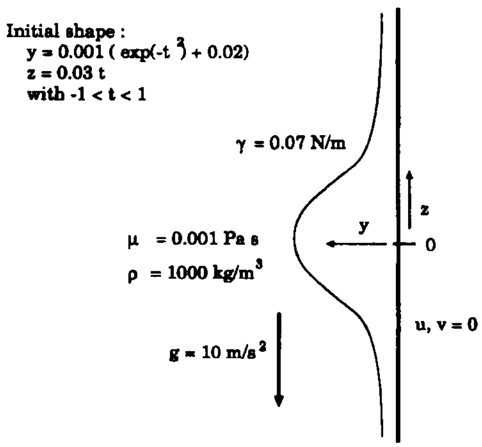

Figure 4. Definition of test problem for moving boundary flow: development of a 2D drop of water from a disturbance in a thin film on a vertical plate

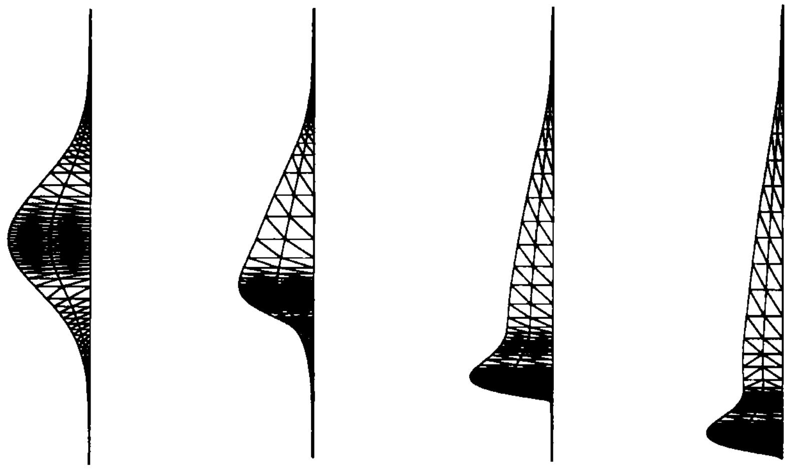

Figure 5. Distribution of elements along free surface as a function of boundary curvature by automatic remeshing scheme

without surface tension the formed shape will be unstable and will readily break.[14] Surface tension stabilizes the drop by a local dominance at the leading edge.

The flexibility in the numerical scheme is used here to structure the mesh automatically to where it is needed. The mesh along the free surface is distributed as a function of the curvature of the surface (see Figure 5). Note that the y-values are scaled by a factor of 10 in all plots.

Figure 6 shows the development of a drop of water. These results were calculated using the implicit algorithm at a constant step size. The total number of steps in the calculation was 200. The implicit algorithm was stable. The major difficulty in these calculations was the automatic remeshing algorithm (e.g. local overshoots of the splines and too strong a curvature with respect to element size).

The explicit algorithm was found to be impractical for this problem. When a constant mesh size and a variable time step (the maximum time step for stability) were used, the algorithm needed 900 time steps to reach time $=0{\cdot}0023$ s. Since the maximum time step decreased rather quickly with time, it was estimated that time $=0{\cdot}01$ s would be reached only after 17,000 time steps. Since the mesh also needs refinement near the leading edge, this must be seen as a best-case estimate; the actual number of time steps needed could easily be 50,000. In Figure 7 the computed free surface profiles of the explicit and the implicit algorithm are compared. The difference is only minor.

8. DISCUSSION

The results presented in the previous section demonstrate the stabilizing effect of the surface tension contribution to the matrix. The merit of this depends strongly on the stiffness of the problem considered. This stiffness may depend on the element size along the free surface, as was the case in the free surface test problem. It may also originate from the timescales of the different physical processes that play a part in the problem, as was the case in the moving boundary example. In both cases the use of the implicit surface tension algorithm drastically reduces the computation costs.

The stability with respect to the surface tension smoothing phenomena enables us to choose the time step correlated with the required accuracy only. In the case of free surfaces, accuracy in time is an issue as long as the iteration converges and the extra term disappears when the solution is approached.

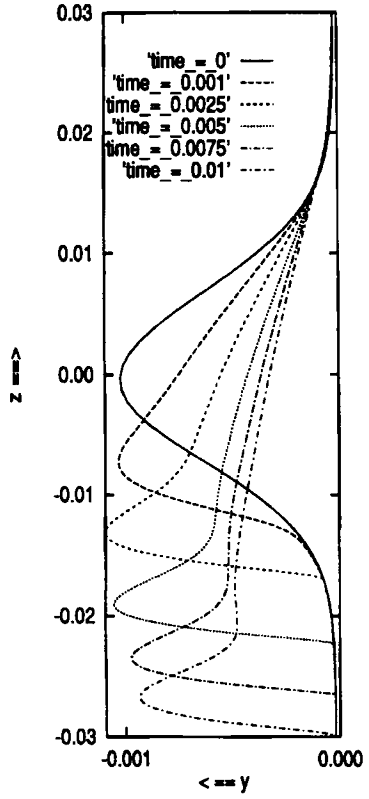

Figure 6. Moving surface geometry of drop being formed as a function of time

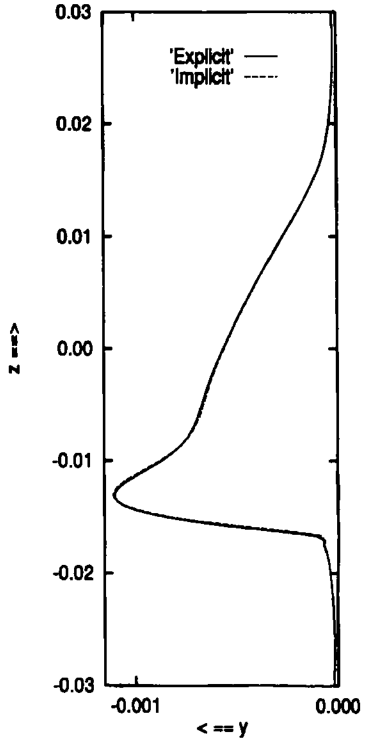

Figure 7. Comparison of drop geometry calculated according to implicit algorithm (50 time steps) and explicit algorithm (900 time steps) at time = 0·0024 s

Although we expected the implicit surface algorithm to follow this rule, combining boundary condition (4) and equation (23), we find that the extra term does not disappear at the steady state solution, since the following term remains:

$$\gamma\,\Delta\tau\int_{\Gamma}\left(\mathbf{u}\cdot\frac{\mathrm{d}\mathbf{n}}{\mathrm{d}s}\right)\mathbf{n}\frac{\mathrm{d}\phi_i}{\mathrm{d}s}\,\mathrm{d}s. \tag{32}$$

This term can be interpreted as being proportional to the distance a material particle has travelled over one time step ($\mathbf{u}\,\Delta\tau$) and to the curvature of the free boundary ($d\mathbf{n}/\,\mathrm{d}s$). It should be noted that our first example is not disturbed by this error, since the converged surface shape is not curved. The remaining term also disappears for linear surface elements (straight line segments), but for isoparametric elements the steady state solution is spoiled by the term above, making it dependent on the iteration procedure (time step size) used for obtaining this solution. From a theoretical viewpoint this is not acceptable.

The origin of this error is found in the mixed implicit–explicit algorithm developed in this paper. To repair the surface tension instability, we have chosen to compute the surface tension term implicitly, whereas the viscous terms are still treated explicitly. In the steady state this translates into the surface tension forces being calculated at the new position of the material particle along the boundary, while the viscous forces are calculated at the old position. This is illustrated by the proportionality with $\mathbf{u}\,\Delta\tau$.

The error can be removed by employing the normal velocity ($\mathbf{u}_n$) both in equation (3) and in the substitution from equation (40) to (41) instead of using the velocity vector itself to trace out the boundaries:

$$\mathbf{u}_n = (\mathbf{u} \cdot \mathbf{n})\mathbf{n}. \tag{33}$$

Usually the choice between **u** and $\mathbf{u}_n$ for tracing the surface is equivalent. Here the normal velocity, however, limits the distance between the old and new positions of the material points at the surface, thus reducing the error introduced by the implicit–explicit algorithm. Using $\mathbf{u}_n$, we derive for the extra term of the implicit surface tension algorithm (equivalent to equation (23))

$$-\gamma\,\Delta\tau \int_\Gamma \frac{\mathrm{d}(\mathbf{u}\cdot\mathbf{n})}{\mathrm{d}s}\mathbf{n}\frac{\mathrm{d}\phi_i}{\mathrm{d}s}\,\mathrm{d}s. \tag{34}$$

It is clear that the term now disappears for a steady state solution for a free boundary.

In moving boundary problems the error introduced by using **u** for surface tracing is greatly reduced, since the time step size must be chosen for accuracy in such a way that the geometry changes only slightly over one time step. As is illustrated in Figure 7, therefore, accuracy was not a problem in the case of our moving boundary problem. There is, however, a second source of error. The implicit Euler time integration is known from differential equation theory to add some additional diffusivity to the problem. This extra smoothing is directly correlated with the size of the time steps. In cases where this affects the accuracy of the calculation, the diffusivity can be limited by evaluating the surface tension at time level $n + \theta$ by adding a prefactor θ to the extra term. With $\theta = 0{\cdot}5$ the modified theta method regenerates the Crank–Nicholson rule, which is second-order-accurate in time (surface tension only!) and does not have the diffuse character of the implicit Euler integration, but at a cost of limited robustness. Usually $\theta = 0{\cdot}6$–$0{\cdot}7$ is recommended, which is a compromise between robustness and accuracy.

We tried these adaptations on our moving boundary problem. We found no advantage, since the accuracy was already good for the Euler time integration. However, a decrease in robustness did occur. During the reordering of the points along the surface the strong curvature at the leading edge of the drop introduced some errors in our spline representation of the surface. These 'larger' errors gave rise to instability problems. Since we did not need extra accuracy, we opted for the robustness of the implicit Euler method.

The two examples shown here were limited to 2D quasi-static Stokes flow. This type of flow was chosen because it is the 'most stable' type of flow. In this case the 'instability' caused by the explicit surface trace algorithm is most clear. Since it originates in the combination of the stiffness of the physical problem and an explicit surface trace algorithm, the 'instability' is not limited to that class of flows. Neither will the solution to the problem presented here be limited to such flows.

For 2D Navier–Stokes flow, where inertia effects play a role, the extra term in the surface tension is expected to resolve the surface tension instability. The merit of the implicit surface tension algorithm will be less apparent than with Stokes flow, since the maximum time step will be limited by convection-induced instabilities (Courant number).

The linearization performed here to obtain the surface tension contribution to the matrix can also be performed for 3D geometries. The equations become more complex. In the linearization, terms can be recognized which are the equivalent of equation (21) in 3D and terms which can be interpreted as representing the blow-up of the surface over time (increase or decrease in surface over the time step). Possibly the 3D equivalent of the term found here will suffice to stabilize the smoothing-related instability. So far we have not had the opportunity to test the 3D generalization of the algorithm presented in this paper, but we will make this the subject of a follow-up paper.

9. CONCLUSIONS

We have shown that the lack of stability of the Picard solution method using the kinematic condition for geometrical updates is caused by the stiffness of the physical problem combined with the explicit time integration of the surface. Through a modification of the boundary element for the surface tension an implicit algorithm is obtained which does not display such stability problems. This extends the Picard solution algorithm to surface-tension-dominated flows with moving boundaries. It also removes the need for two separate iteration schemes for free boundary problems. In all these cases the Picard solution method can be applied using the kinematic update algorithm with the implicit surface tension contribution.

APPENDIX I: DEFINITION OF SURFACE GRADIENT

The surface gradient is an operator that gives the change in a parameter caused by a change in the co-ordinates along the surface. Using a local co-ordinate system along the surface, with s_1 and s_2 being the co-ordinates in the orthogonal directions $\mathbf{t}_1$ and $\mathbf{t}_2$ respectively, we can write

$$\nabla_s p = \frac{\partial p}{\partial s_1}\mathbf{t}_1 + \frac{\partial p}{\partial s_2}\mathbf{t}_2. \tag{35}$$

APPENDIX II: CHANGE IN TANGENT TO A SURFACE WITH TIME

A curve in a two-dimensional Cartesian or a three-dimensional axisymmetrical co-ordinate system can be written by means of a parameter representation in parameter p as[10]

$$\mathbf{x} = \mathbf{x}(p). \tag{36}$$

The tangent vector along the curve is defined as the direction of the vector

$$\mathbf{r} = \frac{d\mathbf{x}}{dp}. \tag{37}$$

The tangent vector is thus

$$\mathbf{t} = \frac{\mathbf{r}}{|\mathbf{r}|} = \frac{\mathbf{r}}{\surd(\mathbf{r}\cdot\mathbf{r})}. \tag{38}$$

The length of the vector $\mathbf{r}$ can be expressed in terms of the length s along the curve as

$$|\mathbf{r}|^2 = \mathbf{r}\cdot\mathbf{r} = \left(\frac{ds}{dp}\right)^2. \tag{39}$$

The change in the tangent to the curve with time is now found by differentiating equation (38):

$$\frac{d\mathbf{t}}{d\tau} = \frac{d}{d\tau}\left(\frac{\mathbf{r}}{\surd(\mathbf{r}\cdot\mathbf{r})}\right) = \frac{1}{\surd(\mathbf{r}\cdot\mathbf{r})}\frac{d\mathbf{r}}{d\tau} - \frac{\mathbf{r}}{\surd[(\mathbf{r}\cdot\mathbf{r})^3]}\left(\mathbf{r}\cdot\frac{d\mathbf{r}}{d\tau}\right). \tag{40}$$

By now substituting equation (36) and keeping in mind that the parameter p is not dependent on time, we obtain with $\mathbf{u} = d\mathbf{x}/d\tau$ that

$$\frac{d\mathbf{t}}{d\tau} = \frac{1}{\surd(\mathbf{r}\cdot\mathbf{r})}\frac{d\mathbf{u}}{dp} - \frac{\mathbf{r}}{\surd[(\mathbf{r}\cdot\mathbf{r})^3]}\left(\mathbf{r}\cdot\frac{d\mathbf{u}}{dp}\right). \tag{41}$$

We now take the parameter p to be equal to the curve distance s at time τ. Following relation (39), we find $\mathbf{r} = \mathbf{t}$ and $\mathbf{r} \cdot \mathbf{r} = 1$. The equation above now reduces to

$$\frac{\mathrm{d}\mathbf{t}}{\mathrm{d}\tau} = \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}s} - \left(\mathbf{t} \cdot \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}s}\right)\mathbf{t} = \left(\mathbf{n} \cdot \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}s}\right)\mathbf{n}, \tag{42}$$

where $\mathbf{n}$ denotes the normal to the surface.

REFERENCES

1. K. H. Hoffmann and J. Sprekels (eds), *Free Boundary Problems: Theory and Applications*, Longman, Harlow/Wiley, New York, 1990.
2. J. M. Floryan and H. Rasmussen, 'A discussion of numerical methods for viscous free surface flows', in K. H. Hoffmann and J. Sprekels (eds), *Free Boundary Problems: Theory and Applications*, Longman, Harlow/Wiley, New York, 1990, pp. 778–787.
3. *FIDAP 7.0, Theory Manual*, Fluid Dynamics International, Evanston, IL, 1993.
4. S. B. G. O'Brien and P. J. Slikkerveer, 'Numerical and asymptotic solutions to a model for waterproofing of telecommunication cables', *Int. J. Eng. Sci.*, **32**, 1283–1301 (1994).
5. N. P. Kruyt, C. Cuvelier, A. Segal and J. van der Zanden, 'A total linearization method for solving viscous free boundary flow problems by the finite element method', *Int. j. numer. methods fluids*, **8**, 351–363 (1988).
6. H. Saito and L. E. Scriven, 'Study of coating flow by the finite element method', *J. Comput. Phys.*, **42**, 53–76 (1982).
7. C. Cuvelier, A. Segal and A. A. van Steenhoven, *Finite Element Methods and Navier Stokes Equations*, Riedel, Dordrecht, 1986.
8. C. E. Weatherburn, *Differential Geometry in Three Dimensions*, Cambridge University Press, London, 1972.
9. *Sepran Finite Element Toolbox*, Ingenieursbureau Sepra, Leidschendam, 1993.
10. L. Kuipers and R. Timman, *Handboek der Wiskunde*, Scheltema and Holkema, Amsterdam, 1966.
11. N. Praagman, personal communication, 1993.
12. E. B. Dussan, 'On the spreading of liquids on solid surfaces: static and dynamic contact lines', *Ann. Rev. Fluid Mech.*, **11**, 371 (1979).
13. S. W. Joo and S. H. Davis, 'Instabilities of three dimensional viscous falling films', *J. Fluid Mech.*, **242**, 529–647 (1992).
14. S. B. G. O'Brien, 'On Marangoni drying: non-linear kinematic waves in a thin film', *J. Fluid Mech.*, **254**, 649–670 (1993).