JOURNAL OF COMPUTATIONAL PHYSICS 44, 262–276 (1981)

Hele–Shaw Flow with a Cusping Free Boundary*

GUNTER H. MEYER

School of Mathematics, Georgia Institute of Technology, Atlanta, Georgia 30332

Received January 5, 1981

Hele–Shaw flow with suction for an incompressible fluid in a porous medium is introduced as a nontrivial model problem for two-dimensional transient free boundary problems. In this model an initially smooth free boundary can move with unbounded velocities and develop cusps. Two analytic solutions are worked out for a special geometry with which numerical results can be compared. It is then shown that the method of lines can reasonably well reproduce the analytic solutions, although at the expense of long computer times. Some strategies for improving the efficiency of the method of lines, including a multi-grid method, are described.

1. INTRODUCTION

In past years a number of methods have been proposed for the numerical solution of Stefan and related free interface and boundary problems. Depending on the model equations such methods can be based on an enthalpy formulation, on variational inequalities, or on front tracking [11]. Numerical experiments with all these methods invariably show good agreement with closed-form solutions where available, or with equivalent computations based on alternate methods. While the last word on the efficient computation of Stefan problems may not have been spoken it is clear that any one particular problem of this type can be considered solvable with current numerical methods.

There are, however, practical free boundary problems which formally show much the same structure as Stefan problems but for which the performance of numerical methods is not routinely predictable. For example, the classical Muskat problem in porous medium flow with an unfavorable mobility ratio leads to viscous fingering which is difficult to simulate numerically [10]. Similarly, dendritic growth on the solidification front of a molten solid cannot yet be adequately modelled numerically [3]. It is our aim here to examine the performance of front tracking based on a method-of-lines discretization for the numerical solution of Hele–Shaw flow in a porous medium. Although this problem does not appear as difficult as the Muskat or dendrite problem it does show several complicating features absent in the Stefan problem, notably rapid free surface movement and the development of nonsmooth

* This research was supported by the U.S. Army Research Office under Contract DAAG-79-C-0145.

0021-9991/81/120262–15$02.00/0

free boundaries. We shall demonstrate that the time-implicit method-of-lines approach, which was presented in [8] as a general purpose algorithm for free boundary problems, does a creditable job (within limits) of solving Hele–Shaw flow. In addition, we show that a multi-grid algorithm for the method-of-lines approximation provides a modest acceleration of the SOR iteration employed in [8]. Since computer run times depend heavily on the number of lines in the discretization, any decrease in run times allows more lines and thus a better numerical resolution, which is essential for problems with irregular boundaries. Last but not least, we expand the comments of [9] and present an essentially analytic solution of a two dimensional transient problem with a cusping free boundary with which our numerical results can be compared. Since Hele–Shaw flow still presents sizable problems this solution may serve as a benchmark against which other numerical methods yet to be developed or examined can also be compared.

2. The Model

The equations for Hele–Shaw flow can be written as

$$\Delta u = 0, \qquad (r,\theta) \in D(t), \quad t>0, \tag{2.1a}$$

$$u = f(\theta, t), \qquad (r,\theta) \in \partial D_1(t), \quad t>0, \tag{2.1b}$$

$$u = 0, \tag{2.1c}$$

$$\qquad (r,\theta) \in \partial D_2(t), \quad t>0.$$

$$\nabla u = -L\frac{dR}{dt}, \tag{2.1d}$$

Here u denotes the fluid pressure in an incompressible fluid in a reservoir $D(t)$ between the inner well boundary $\partial D_1(t)$, usually taken as the circle $r = r_0$, and the moving free outer boundary $\partial D_2(t)$ which is identified with the zero isobar ($u = 0$). The second free boundary condition

$$\nabla u = -L\frac{dR}{dt} \equiv -L\left(\frac{dr}{dt}, r\frac{d\theta}{dt}\right)$$

determines the moverment of the position vector $R(t)$ from the reservoir center at $r = 0$ to any point of the free boundary. It is Darcy's law for flow in a porous medium in which L describes the fluid and reservoir properties. Finally, the initial reservoir $D(0)$ is assumed given.

In [9] the expression

$$\nabla u \cdot \nabla u - Lu_t = 0, \qquad (r,\theta) \in \partial D_2(t), \tag{2.2}$$

is given instead of (2.1d). If the solution u of (2.1) is sufficiently smooth then it necessarily satisfies (2.2) because it follows from

$$u(r(t), \theta(t), t) = 0, \qquad (r, \theta) \in \partial D_2(t),$$

that $du/dt = u_r\, dr/dt + u_\theta\, d\theta/dt + u_t \equiv 0$. Substitution of (2.1d) for dr/dt and $d\theta/dt$ yields (2.2). We prefer to work here with (2.1d) because the absence of u_t in the equations somewhat simplifies the numerical method.

In the above model the well $\partial D_1(t)$ is usually interpreted as an approximation to a point source or sink. Thus we think of (2.1b) as

$$u(r_0, \theta, t) = K \ln r_0, \qquad r_0 \ll 1. \tag{2.3}$$

If $K < 0$ then the model describes a reservoir into which fluid is injected. If the fluid were slightly compressible then (2.1a) would have to be replaced by $\Delta u - \alpha u_t = 0$ and the resulting problem is identical to the usual one phase Stefan problem. Therefore, the Hele–Shaw injection problem is not considered challenging. On the other hand, if we choose $K > 0$ then fluid is extracted and we speak of Hele–Shaw flow with suction. For a slightly compressible fluid Hele–Shaw flow with suction would be equivalent to a Stefan problem with negative latent heat. We also note that the suction problem is identical with the electrodeposition problem where a cathode is plated with a metal during electrolysis. The injection problem is equivalent to electrochemical machining, where the anode is eroded during electrolysis [7].

Hele–Shaw flow with injection, electrochemical machining, and the one-phase Stefan problem have been studied extensively (see [5, 4, 6], resp.). The key tool has been the conversion of the free boundary problem into a variational inequality for an associated dependent variable. For example, in the injection problem it follows from the maximum principle that $\|\nabla u\| \neq 0$ and $u_t > 0$ on $\partial D_2(t)$. Hence the free boundary $u = 0$ can be expressed as $t = s(x)$, the domain $D(t)$ is growing and $t < s(x)$ for any x outside $\partial D_2(t)$. We can set $u \equiv 0$ outside $D(t)$ and formally introduce

$$w(x, t) = \int_0^t u(x, \tau)\, d\tau, \qquad 0 < t < T, \quad x \in \hat{D}, \tag{2.4}$$

where T is an arbitrary but fixed final time, and where $\hat{D}$ is a bounded smooth domain which contains $D(T)$ and which has the same well boundary $r = r_0$. A straightforward calculation (see [4]) shows that w satisfies the free boundary problem

$$\begin{aligned} \Delta w &= -L, \qquad x \in D(t) - D(0), \\ &= 0, \qquad x \in \overline{D(0)}; \end{aligned} \tag{2.5}$$

$$w(r_0, \theta, t) = Kt \ln r_0 > 0,$$

$$w = \partial w/\partial n = 0, \qquad x \in \partial D_2(t).$$

It follows from [6, p. 150] that any sufficiently smooth solution of the variational inequality

$$\int_{\hat{D}} \nabla w \cdot \nabla(v - w)\, dx \geqslant \int_{\hat{D}} f(x)(v - w)\, dx, \qquad w, v \in \mathbb{K}, \tag{2.6}$$

is necessarily a solution of (2.5). Here

$$\begin{aligned} f(x) &= L, \qquad x \notin D(0), \\ &= 0, \qquad x \in D(0), \end{aligned}$$

and $\mathbb{K} = \{v\colon v \in H^1(\hat{D});\ v \geqslant 0;\ v(r_0, \theta, t) = Kt \ln r_0;\ v = 0$ on the outer boundary of $\hat{D}\}$. The existence of a unique solution of (2.6) can be asserted, a convergent numerical scheme to compute w, and the equivalence of w and u can be established [4]. From a practical point of view it is especially of use to note that for any time t the variational inequality (2.6) can be solved without regard to the history of the problem in the interval $(0, t)$. The free boundary at time t is simply the set $\{x{:}w(x, t) = 0\}$.

In the suction problem the free boundary can again be expressed as $t = s(x)$, but now the domain is contracting and $t > s(x)$ for any x outside $\partial D_2(t)$. In this case the transformation (2.4) is no longer useful because w on $\partial D_2(t)$ will depend on $\partial D_2(t)$. A formal conversion into a variational inequality can be carried out with the transformation

$$w(x, t) = \int_t^T u(x, \tau)\, d\tau,$$

where as before $u \equiv 0$ outside $\partial D_2(t)$. However, this transformation is only meaningful if $\partial D_2(T)$ is known which amounts to a replacement of the initial value problem (2.1) by a final value problem which is equivalent to the injection problem. However, $\partial D_2(T)$ is not known, and in fact, not meaningful if the problem has only a local solution. Thus, at this time, the question of existence and uniqueness of a solution for (2.1) for all but a few special geometries must be assumed on physical grounds.

For comparison with numerical results two essentially analytic solutions of (2.1) are available. The first of these holds for the one-dimensional problem where $\partial_2 D(0)$ is a circle around the well, say $r = R$. It is straightforward to verify that the solution of (2.1) is given by

$$u(r, t) = -K \ln r_0 \left(\frac{\ln r - \ln r_0}{\ln s - \ln r_0}\right) + K \ln r_0, \tag{2.7}$$

where the free boundary $r = s(t)$ is determined from

$$\left.\frac{\partial u}{\partial r}\right|_{r=s} = -\frac{K \ln r_0}{s(\ln s - \ln r_0)} = -L\,\frac{ds}{dt}. \tag{2.8}$$

Integration shows that $s(t)$ must be a root of the equation

$$F(s) = (K \ln r_0)\, t - L \frac{s^2}{2}\left[\ln s - \ln r_0 - \frac{1}{2}\right] + L \frac{R^2}{2}\left[\ln R - \ln r_0 - \frac{1}{2}\right] = 0.$$

We see that $F'(s) = -Ls[\ln s - \ln r_0] \leqslant 0$ and $F''(s) = -L(1 + \ln s/r_0) < 0$ on $[r_0, R]$. Hence the root $r = s(t)$ can always be found with Newton's method. For this reason (2.7) may be considered a given function for any value of t. We observe from (2.8) that $ds/dt \to -\infty$ as $s \to r_0$. (We remark that $|ds/dt| < \infty$ in the constant-flow-rate problem, where (2.1b) is replaced by $\partial u/\partial r = \text{constant}$ on $r = r_0$.)

The second analytic solution applies to a truly two dimensional problem and can be derived following the ideas of [9]. Thus, let us suppose that $D(t)$ is a domain in the complex z plane, which is the image of the unit circle $|w| = 1$ in the complex w plane under the conformal transformation $z = a_1(t)\, w + a_2(t)\, w^2$. In general $D(t)$ will be a limaçon. The solution of Laplace's equation in $D(t)$ and vanishing on $\partial D_2(t)$ is given by

$$u(x, y, t) = \operatorname{Re} \phi(z, t) = \operatorname{Re} K \ln w, \tag{2.9}$$

where ϕ is an analytic function. The task is to determine a_1 and a_2 such that Darcy's law holds on the free boundary. We use the expression (2.2) and recall that $\nabla u \cdot \nabla u = |\partial\phi/\partial z|^2$ and $\partial\phi/\partial z = (\partial\phi/\partial w)(\partial w/\partial z)$. It follows that

$$\frac{\partial\phi}{\partial z} = \frac{K}{w(a_1 + 2a_2 w)};$$

furthermore, $u_t = \operatorname{Re}(\partial\phi/\partial w)(\partial w/\partial t) = \operatorname{Re}(-K/w(a_1 + 2a_2 w))(a_1'(t)\, w + a_2'(t)\, w_2)$. Substitution into (2.2) and simplification results in

$$K = L\left\{\frac{1}{2}\frac{d}{dt}|a_1|^2 + \frac{d}{dt}|a_2|^2 + \operatorname{Re} w(\bar{a}_1 a_2' + 2\bar{a}_1 a_2)\right\}.$$

This condition will hold everywhere on $\partial D_2(t)$, i.e., when $|w| = 1$, provided the last term vanishes. This is achieved if we set $a_1^2 a_2 \equiv 1$. Integration now shows that a_1 and a_2 must be chosen such that

$$|a_1|^2 + 2\,|a_2|^2 = -2(K/L)\, t + C. \tag{2.10}$$

C is an arbitrary constant at our disposal. For the numerical work below we shall choose $K = L = 1$ and replace $-2t + C$ by $-\tau$ where τ is increasing. If we set $a_2 = \rho$ then (2.10) can be rewritten as the cubic equation

$$2\rho^3 + 2\rho\tau + 1 = 0,$$

which for $\tau < \tau_c = -(17/16)^{1/3} \approx -1.190551$ has three real roots. We shall choose the root

$$\rho = 2\sqrt{|\tau|/3}\cos\left(\frac{\psi + 4\pi}{3}\right),$$

where ψ is determined from

$$\cos\psi = -3\sqrt{3}/4\sqrt{|\tau|^3}.$$

The image of the unit circle in the w plane becomes the contour

$$z = x + iy = \frac{1}{\sqrt{\rho}} e^{i\beta} + \rho e^{2i\beta}, \qquad \beta \in [0, 2\pi). \tag{2.11}$$

We note that $|z|^2 = 1/\rho + \rho^2 + 2\sqrt{\rho}\cos\beta$ so that the free boundary at time $\tau < \tau_c$ is a limaçon with minimum distance to the origin when $\beta = \pi$, i.e., when $y = 0$. Moreover, this boundary is smooth for $t < t_c$; in fact for $\tau < -1.5$ the free boundary encloses a convex domain and looks almost circular. At $\tau = -1.4$ the domain no longer is convex. As τ increases the indentation on $y = 0$ becomes more pronounced, and at $\tilde{t} = \tau_c$ the point $(x, 0)$ has become the vertex of a cusp. Since this cusp defines a reentrant corner for $D(\tau_c)$ the gradient and hence the radial speed of the free boundary are unbounded. Thus in Hele–Shaw flow with suction an initially smooth

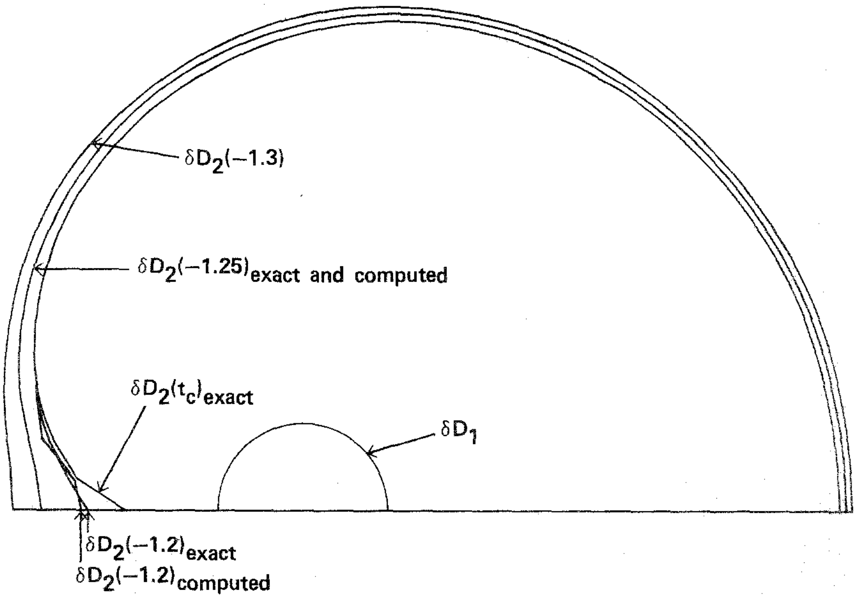

FIG. 1. Computed and exact free boundaries for Run I. For $t = -1.3$ and $t = -1.25$ the computed and correct free boundaries are indistinguishable. For $t = -1.2$ oscillations are beginning to appear near $\theta = \pi$. The innermost free boundary is $\partial D_2(t_c)$ with a cusp at $\theta = \pi$.

free boundary can evolve in finite time into a nonsmooth boundary moving with infinite speed. No equivalent behavior occurs in the classical Stefan problem. A plot of $\partial D_2(\tau_c)$ is shown in Fig. 1.

We remark that in this problem the point sink at the well is taken in the w plane. The condition on the actual well boundary $r = r_0$ is determined from the conformal transformation. A slight dependence of $u(r_0, \theta, t)$ on θ and t is observed.

3. The Numerical Method

The method-of-lines–SOR algorithm described in detail in [8] will be applied to the above Hele–Shaw flow model. For completeness, and to illustrate the changes made for this problem, we shall summarize the method.

The numerical method tracks the free boundary in time on specified rays coming from the origin. It applies directly to Eqs. (2.1) except that (2.1d) is replaced by

$$L\frac{\partial s}{\partial t} = -\left(1 + \left(\frac{\partial s}{\partial \theta}\Big/ s\right)^2\right)\frac{\partial u}{\partial r}, \tag{3.1}$$

where $r = s(\theta, t)$ is the equation of the free boundary and $\partial s/\partial t$ is the speed of a boundary point on the ray $\theta = \text{constant}$. [Equation (3.1) is easily derived from $r(t) = s(\theta(t), t)$, $u(r(\theta, t), \theta, t) = 0$, and (2.1d).] For the Stefan problems in [8] an implicit Euler method in time proved adequate. For the Hele–Shaw suction problem a Crank–Nicolson time discretization is necessary to correctly track the boundary in time. The method of lines with discrete θ and t and continuous r based on a central difference quotient in θ then leads to the following multi-point system at time $t = t_n$

$$u_i'' + \frac{1}{r}u_i' - \frac{2}{(r\Delta\theta)^2}u_i = -\frac{1}{(r\Delta\theta)^2}(u_{i+1} + u_{i-1}), \tag{3.2a}$$

$$u_i(r_0) = f(\theta_i, t_n), \tag{3.2b}$$

$$u_i(s_i) = 0, \tag{3.2c}$$

$$L\left(\frac{s_i - s_{i,n-1}}{\Delta t}\right) = -\frac{1}{2}\left\{\left(1 + \left(\frac{s_{i+1} - s_{i-1}}{s_i \Delta\theta}\right)^2 u_i'(s_i)\right)\right.$$
$$\left. + \left(1 + \left(\frac{s_{i+1,n-1} - s_{i+1,n-1}}{s_{i,n-1}\Delta\theta}\right)^2 u_{i,n-1}'(s_{i,n-1})\right)\right\} \tag{3.2d}$$

for $i = 0,..., N$ and $\theta_i = \theta_0 + (i/N)(\theta_N - \theta_0)$. All quantities involving the subscripts -1 and $N+1$ are determined either by symmetry or periodicity. For example, if $\theta = \theta_0$ is a symmetry boundary then $u_{-1} = u_1$ and $s_{-1} = s_1$. The subscript $(i, n-1)$ denotes the dependent variable on the ray $\theta = \theta_i$ at the preceding time level $t_{n-1} = t_n - \Delta t$. The values of $\{s_{i,0}\}_{i=0}^N$ are given, while $\{u_{i,0}\}_{i=0}^N$ is taken to be the solution of (3.2a–c) (i.e., the approximate solution to $\Delta u = 0$ with $u = 0$ on the given boundary $\partial D_2(0)$).

As in [8] the above multi-point free boundary problem is solved sequentially with a line SOR method. Starting with the solution at the preceding time level as an initial guess $\{u_i^0\}$ and $\{s_i^0\}$ we find an intermediate solution $(\tilde{u}_i, s_i^k\}$ in iteration k from the scalar two point free boundary problem obtained from (3.2) after the substitution

$$u_{i+1} \leftarrow u_{i+1}^{k-1}, \qquad s_{i+1} \leftarrow s_{i+1}^{k-1},$$
$$u_{i-1} \leftarrow u_{i-1}^{k}, \qquad s_{i-1} \leftarrow s_{i-1}^{k}.$$

The new iterate u_i^k is then determined as

$$u_i^k = u_i^{k-1} + \omega(\tilde{u}_i - u_i^{k-1})$$

for some suitable ω.

For the scalar problem we find the invariant imbedding approach convenient because the free boundary is easy to determine. The equations are developed in [8] and will be only listed here. We define $v_i \equiv \tilde{u}_i'$ and employ the Riccati transformation

$$\tilde{u}_i = R(r)\, v_i + w_i^k(r),$$

where

$$R' = 1 + \frac{1}{r} R - \frac{2}{(r\Delta\theta)^2} R^2, \qquad R(r_0) = 0$$

$$(w_i^i)' = -\frac{2R(r)}{(r\Delta\theta)^2} w_i^k + \frac{R(r)}{(r\Delta\theta)^2}(u_{i+1}^{k-1} + u_{i-1}^k), \qquad w_i^k(r_0) = f(\theta_i, t_n).$$

The free boundary s_i^k is found as the largest root on $[r_0, s_{i,n-1}]$ of

$$\phi_i^k(r) \equiv L\frac{r - s_{i,n-1}}{\Delta t} + \frac{1}{2}\left\{\left(1 + \left(\frac{s_{i+1}^{k-1} - s_{i-1}^k}{r\Delta\theta}\right)^2\right)\right]\left(-\frac{w_i^k(r)}{R(r)}\right)$$
$$+ \left(1 + \left(\frac{s_{i+1,n-1} - s_{i-1,n-1}}{s_{i,n-1}\Delta\theta}\right)^2 u_{i,n-1}'(s_{i,n-1})\right]\right\} = 0. \qquad (3.3)$$

Finally, we find v_i and hence $\tilde{u}_i$ by integrating

$$v_i' + \frac{1}{r} v_i - \frac{2}{(r\Delta\theta)^2}[R(r)\, v_i + w_i^k(r)] = -\frac{1}{(r\Delta\theta)^2}(u_{i+1}^{k-1} + u_{i-1}^k),$$

$$v_i(s_i^k) = -\frac{w_i^k(s_i^k)}{R(s_i^k)}.$$

If necessary, functions are extended quadratically beyond s_i^k.

The numerical solution of these equations is carried out as follows. For the Riccati equation the analytic solution

$$R(r) = \frac{r\Delta\theta}{\sqrt{2}} \tanh \frac{\sqrt{2}}{\Delta\theta} \ln \frac{r}{r_0}$$

is used. The linear equation for w_i^k is solved with the trapezoidal rule on a fixed $r-$grid which is common to all rays to allow easy communication from one ray to the next. As w_i^k becomes available the value of $\phi_i^k(r)$ is determined. If ϕ_i^k changes sign between the mesh points r_m and r_{m+1} in a neighborhood of $s_{i,n-1}$ then the root s_i^k is determined as the zero of the quadratic interpolant to $\phi_i^k(r)$ through $\phi_i^k(r_{m-1})$, $\phi_i^k(r_m)$, and $\phi_i^k(r_{m+1})$. The integration of v_i again is carried out with the trapezoidal rule. If necessary, linear interpolation is used to determine v_i' at $r=s_i^k$.

As observed in [8] the choice of Δr for the trapezoidal rule is constrained. To avoid artificial oscillations in the integration of w_i^k and v_i^k we require that

$$\Delta r = r_{m+1} - r_m < r_m^2 \Delta\theta^2/R(r_m) \sim \sqrt{2}\, r_m \Delta\theta.$$

This condition, as well as the rapid change of $R(r)$ near r_0, introduces a numerical boundary layer which requires a fine grid near r_0. One advantage of this approach is the flexibility in treating problems with a great variety of boundary conditions. For example, the injection problem is solved with these same equations provided t is decreasing rather than increasing. The dominant drawback of this method for multi-dimensional problems is the slow convergence of the SOR iteration as the number of rays increases. For this reason a small number of lines have generally been used before. However, for Hele–Shaw flow with suction a fairly fine angular resolution seems mandatory to trace the cusping surface. Three modifications of the basis research code of [8] were examined as a means of speeding up convergence.

The simplest improvement is an adaptive estimation of the relaxation parameter ω from one time step to the next based on the assumption that the optimum ω is nearly independent of time. We begin by choosing reasonable values ω_0 and ω_1 for the computation of the initial guess $\{u_{i,0}\}$ and the solution $\{u_{i,1}\}$ at time $t=t_1$. At $t=t_2$ we choose $\omega_2=\omega_1 \pm (\omega_1-\omega_0)$ where $+$ is choosen if the number of iteration I_1 at t_1 is less than the number I_0 at t_0. In general, given I_{n-2}, I_{n-1}, and I_n with the corresponding values ω_{n-2}, ω_{n-1}, ω_n we set $\omega_{n+1}=\omega_n+(\omega_n-\omega_{n-1})$ if $I(\omega)$ is monotone for $n-2$, $n-1$, n. If $I(\omega)$ is not monotone then we set $\omega_{n+1}=\omega_n-(\omega_n-\omega_{n-1})/2$. (As expected, only relative minima were observed when $I(\omega)$ was not monotone). This simple updating of ω appears to work very well.

Since the rate of convergence depends on the number of lines one can reduce the need for a large number of rays by using a fourth order quotient

$$u_{\theta\theta} \approx \frac{1}{12\,\Delta\theta^2}\left[-u_{i-2}+16u_{i-1}-30u_i+16u_{i+1}-u_{i+2}\right].$$

Together with the updating algorithm for ω this change improves the accuracy of the computed solution.We note that for this approximation small changes in the coefficients and source terms of the invariant imbeding equations become necessary. For example, in the Riccati equation the quantity $\sqrt{2}$ must be replaced by $\sqrt{30/12}$.

The third, and most promising improvement of the code of [8] is the coupling of a multi-grid algorithm (see, e.g. [2]) with the method of lines. Use of a multi-grid technique seems reasonable because during the SOR iteration we observe the usual behavior of initially fast convergence of $\{u_i^k\}$ and $\{s_i^k\}$ with k, followed by increasingly smaller changes from one iteration to the next. Based on the observation that most time of the computation is spent on solving the differential equation $\Delta u = 0$, while the boundary correction appears to have little influence we shall use the multi-grid method in the following sense. On the original fine grid the problem (2.1) is solved as a free boundary problem.The residual equations then are solved on coarser grids as a fixed boundary problem on $D^k(t_n)$, where $D^k(t_n)$ is the domain obtained from the last fine grid calculation. Specifically, let $\{u_i^k, s_i^k\}$ be the solution of (3.2) obtained with $\omega = 1$ in the kth iterations. u_i^k satisfies

$$u_i^{k''} + \frac{1}{r} u_i^{k'} - \frac{2}{(r\Delta\theta)^2} u_i^k = -\frac{1}{(r\Delta\theta)^2} (u_{i+1}^{k-1} + u_{i-1}^k),$$

$$u_i^k(r_0 = f(\theta_i, t_n), \qquad u_i^k(s_i^k) = 0.$$

We wish to improve $\{u_i^k\}$ with the multi-grid method on he domain defined by the computed boundary points $\{s_i^k\}$. If we define the residual $z_i^k = u_i - u_i^k$ and arbitrarily set $z_i^k(s_i^k) = 0$ then $\{z_i^k\}$ is an approximate solution to be fixed boundary problem:

$$\begin{aligned} \Delta z &= S(r, \theta) && \text{on} \quad D^k(t_n), \\ z &= 0 && \text{on} \quad r = r_0, \\ z &= 0 && \text{on} \quad \partial D_2^k(t_n), \end{aligned}$$

where $D^k(t_n)$ is a smooth approximation to $D(t_n)$ defined by the boundary points $\{s_i^k\}$, and where $S(r, \theta)$ for $\theta = \theta_i$ and $r \in (r_0, s_i^k)$ has the form

$$S(r, \theta_i) = -\frac{1}{r^2 \Delta\theta^2} (u_{i+1}^k - u_{i+1}^{k-1}).$$

This boundary value problem again is solved with the method of lines on a new grid with half as many lines and mesh points per line. The same invariant imbedding equations can be used except that the source terms must be updated. Once $\{z_i^k\}$ has been found on this new grid its values at the remaining mesh points of the old grid are computed by linear interpolation and

$$u_i = u_i^k + z_i^k, s_i^k$$

serve as new data for the $(k+1)$st Gauss–Seidel iteration on the original free boundary problem.

The actual computation preceeds as follows. Starting with the solution from the proceeding time level as initial guess we compute $\{u_i^k, s_i^k\}$, $k = 1, 2, \ldots$ until convergence slows down. If we define $e^k = \max_i |u_i^k - u_i^{k-1}|$ then convergence is considered slow whenever $e^k/e^{k-1} > 0.65$. We then compute the residual problem on the coarser grid until $\max_i |z_i^{k,l} - z_i^{k,l-1}| < 10^{-4}$, where $\{z_i^{k,l}\}$ are the Gauss–Seidel iterates approximating $\{z_i^k\}$. The new approximation of u_i is found and the iteration on the original grid is resumed. A cycling between the original and the residual problem results until $e^k < 10^{-6}$ and $\max_i |s_i^k - s_i^{k-1}| < 10^{-6}$, when the iteration is considered to have converged.

The solution of the residual equation for z is itself accelerated in the usual way by computing its residual on a yet coarser grid. Altogether, up to four grids, the finest with 24 lines and 200 points per line, are used in our multi-grid code.

While we do not have any theoretical justification for the success of the multi-grid method in this free boundary setting the results obtained so far seem to justify this approach and make further research into this method desirable.

4. Numerical Results

The case of a circular free boundary concentric with the well boundary $r = r_0$ is a convenient test problem for debugging the code and for obtaining some feel for the Hele–Shaw suction problem. (Note that even if only one ray is chosen the problem is still solved iteratively because (3.2) (for $\omega = 1$) reduces to

$$u_0^{k''} + \frac{1}{r} u_0^{k'} - \frac{2}{(r\,\Delta\theta)^2} u_0^k = -\frac{2}{(r\,\Delta\theta)^2} u_0^{k-1}, \qquad k = 1, 2, \ldots.$$

(Of course, if in fact a one dimensional problem is to be solved then the linear and the source terms can be eliminated and no iteration is necessary.) We have collected some representative results both for the suction ($t \geqslant 0$) and injection ($t \leqslant 0$) problem. We also provide a comparison with the results obtained when the free surface is explicitly predicted with the formula

$$s_{i,n+1} = s_{i,n} - \Delta t\, u'_{i,n}(s_{i,n}) \qquad (L = K = 1)$$

and the resulting fixed boundary problem is solved with the above invariant imbedding method. A time-implicit method appears to be essential for the suction but not for the injection problem. However, in a line SOR method the only additional cost of an implicit over an explicit method lies in the evaluation of the function $\phi_i^k(r)$ in (3.3), which is a small price to pay for the improved performance. The poor performance of the explicit method raises doubt that the predictor corrector methods common for Stefan problems will succeed for the suction problem since even for the Stefan problem convergence may have to be forced through underrelaxation [1]. All further results were obtained with the implicit Crank–Nicolson method.

Let us now turn to the two dimensional suction problem. As the preliminary experiment in [8] showed, the method of lines will produce a cusping free boundary when non-concentric circles are the initial boundaries. This is qualitatively the correct solution. Tables II and III below indicate quantitatively how well the free boundary (2.11) can be reproduced. In Table II only the closest point $r = s(\pi, t)$ is shown which at $\tau_c \approx 1.190557$ becomes the vertex of the cusp. Table III is included to

TABLE I

Radial Hele–Shaw Suction and Injection Problem

n	t_n	$\frac{t_n - t_{n-1}}{\Delta t}$	$s(t_n)$	Relative error (%) in the free boundary position	
				Time-implicit method	Time-explicit method
0	0	—	0.5	0	0
1	0.075	30	0.2452	−0.02	+2.57
2	0.090	30	0.1121	−0.20	+17.6
3	0.094	20	0.0644	−0.85	+54.8
4	0.042	18	0.0408	−2.92	+114.4
−1	−0.1	20	0.6894	−0.003	+0.23
−2	−0.2	20	0.8294	−0.006	+0.23
−3	−0.3	20	0.9451	0.007	+0.22

Note. $s(0) \equiv R = 0.5$; $\Delta r = 0.0025$ for the suction problem; $\Delta r = 0.0025$ on $[0.01, 0.1]$, $\Delta r = 0.005$ on $[0.1, 0.2]$, $\Delta r = 0.8/140$ on $[0.2, -1]$ for the injection problem. The percent relative error is defined as $100 \cdot [s_{0,n} - s(t_n)]/s(t_n)$, where $s(t)$ is the analytic solution. Note: $[t_n - t_{n-1}]/\Delta t$ indicates the number of time steps between the display times t_{n-1} and t_n.

TABLE II

Two Dimensional Hele–Shaw Flow with Suction

Run	n	t_n	$\frac{t_n - t_{n-1}}{\Delta t}$	$s(t_n)$	$S_{N,n}$	Maximum relative error in (%)
I	0	−1.3	—	1.0170	—	—
	1	−1.25	5	0.9142	0.9147	+0.3
	2	−1.12	10	0.7426	0.7803	+5.1
II	0	−1.3	—	1.0170	—	—
	1	−1.25	5	0.9142	0.9147	+0.3
	2	−1.2	10 or 20	0.7426	0.7661	+3.2
III	0	−2.0	—	1.7076	—	—
	1	−1.3	14	1.0170	1.0149	−0.2
	2	−1.25	5	0.9142	0.9144	+1.1
	3	−1.2	10	0.7426	0.7980	+9.1
IV	0	−1.2	—	0.7926	—	—
	1	−1.199	1	0.7364	0.7384	+0.3
	2	−1.198	1	0.7299	0.7340	+0.6
	3	−1.197	1	0.7230	0.7297	+0.9
	4	−1.196	1	0.7154	no convergence of the SOR iteration	

TABLE III

Analytic and Computed Free Boundaries for Hele–Shaw Flow with Suction at $\tau = -1.2$

Ray	Analytic solution at $\tau = -1.2$	Run I	Run II	Run III	Analytic solution at τ_c
0	1.8937	1.8936	1.8936	1.8934	1.8899
1	1.8916	1.8915	1.8915	1.8911	1.8878
2	1.8856	1.8854	1.8854	1.8851	1.8817
3	1.8755	1.8754	1.8754	1.8750	1.8716
4	1.8614	1.8612	1.8612	1.8610	1.8575
5	1.8433	1.8431	1.8431	1.8427	1.8393
6	1.8212	1.8210	1.8210	1.8206	1.8172
7	1.7952	1.7950	1.7949	1.7945	1.7911
8	1.7652	1.7699	1.7649	1.7644	1.7610
9	1.7313	1.7310	1.7310	1.7305	1.7270
10	1.6935	1.6932	1.6932	1.6927	1.6891
11	1.6519	1.6515	1.6515	1.6507	1.6473
11	1.6064	1.6060	1.6060	1.6054	1.6016
13	1.5570	1.5566	1.5566	1.5559	1.5521
14	1.5039	1.5034	1.5034	1.5025	1.4987
15	1.4469	1.4463	1.4463	1.4454	1.4413
16	1.3860	1.3854	1.3853	1.3849	1.3800
17	1.3211	1.3204	1.3203	1.3181	1.3146
18	1.2520	1.2513	1.2513	1.2552	1.2450
19	1.5786	1.1777	1.1769	1.1679	1.1706
20	1.1002	1.1001	1.1014	1.1219	1.0911
21	1.0163	1.0134	1.0096	0.9942	1.0053
22	0.9254	0.9439	0.9411	1.0163	0.9108
23	0.8257	0.8063	0.8209	0.7569	0.8021
24	0.7426	0.7803	0.7661	0.7980	0.6303

Note. $r_0 = 0.3$; $\Delta r = 0.0025$ on [0.3, 0.4], $\Delta r = 0.01$ on [0.4, 2]; $\Delta\theta = \pi/24$. Ray #i is given by $\theta = \theta_i = i\pi/24$. The maximum relative error is $\max_i\{|s_{i,n} - s(\theta_i, t_n)|/s(\theta_i, t_n)\} \cdot 100$.

encourage a comparison of the exact solution with numerical results for this nontrivial model problem. For the computation the line $y = 0$ is used as a symmetry boundary.

The values of $s(\theta_i, t_n)$ are obtained by transferring 100 evenly spaced boundary points on the upper half circle in the w plane onto the z plane and by interpolating linearly in the z plane to obtain the free boundary on the specified rays $\theta = \theta_i$, $i = 0,..., 24$.

The results of run I are obtained with a second order approximation of $u_{\theta\theta}$. They show a maximum relative error of less than one percent until the last five time steps when the error increases to 5.1%. Figure 1 shows the exact and computed free boundaries of run I at $\tau = -1.3$, $\tau = -1.25$, and $\tau = -1.2$. The oscillation of the computed free boundary about the true position at $\tau = -1.2$ is a typical indication in these experiments that the solution is beginning to break down. This behavior is especially pronounced in the long time run III. The oscillations appear to be due to the approximation of $u_{\theta\theta}$ on a domain which is developing a reentrant corner. The maximum error is somewhat reduced and the oscillations virtually disappear when run I is repeated with a fourth order approximation of $u_{\theta\theta}$ as shown by the results of run II. Moreover, a doubling of the number of time steps over the initial interval $[-1.25, -1.20]$ had practically no effect on the computed solution of II. Run IV in Table II shows the performance of the method with a second order approximation to $u_{\theta\theta}$ when the initial free boundary is a well-developed limaçon. As the boundary speed increases the method of lines can no longer cope even if the computation is restarted with a smaller time step.

Our results show that it is indeed possible to track the free boundary in a Hele–Shaw suction problem with the method of lines. In particular, run III shows that an initially convex domain at $\tau = -2$ can be followed well into the stage where convexity is lost (say $\tau = -1.25$) with reasonably good accuracy, especially when a fourth order method of lines approximation can be used. These results are particularly encouraging when one considers that the time steps of Table II correspond to some fairly long real times in some applications. For example, for oil flow in a typical reservoir the constant in (2.1d) after nondimensionalizing can be shown to be

$$L = -\frac{\mu R^2}{k u_R},$$

where μ is the oil viscosity, k the reservoir permeability, and where u_R and R are a reference pressure drop and a reference reservoir dimension. If we choose $\mu = 2$ cp, $k = 500$ md, $R = 100$ m and $u_R = 2$ atm then the results of Table II apply roughly to a reservoir with an intial drainage radius between 100 and 200 m and a pressure drop of about 1.4 atm between the well and the free boundary. Since $L = 1$ in the above computation the real time τ and the computer time t are linked by

$$\tau = \frac{\mu R^2}{k u_R} t = 2 \times 10^8 t.$$

A time step of 0.01 in run I corresponds to a real time step of about 23 days which is quite long.

The dominant problem of slow convergence and long run times, however, has not been overcome. For 24 rays with 200 mesh points the computation of the initial condition $\{u_{i,0}\}$ alone takes about 60 CPU seconds on the Cyber 174 with the mutligrid code; the computation at any future time level takes about 30 CPU seconds.

The improvement in the rate of convergence due to the multigrid algorithm is about four-fold when compared to a straight Gauss–Seidel iteration; however, the improvement is much more modest when compared with an SOR iteration which utilizes a near optimum relaxation parameter. For example, run I with the multi-grid code restricted to one level but with a near optimum relaxation factor of $\omega = 1.62$

required 725 CPU seconds. The same run with four multi-grid levels (the coarsest with three rays and 25 mesh points per ray) took 582 CPU seconds. Since this a new multi-grid code in an unfamiliar setting, further improvement is expected as we gain experience with it. It is doubtful, however, that it will be economically possible to have sufficiently fine time and space discretizations in order to track the free boundary completely until the critical time with even the best implementation of the method of lines approximation. Thus Hele–Shaw flow with suction remains a challenge.

References

1. R. Bonnerot and P. Jamet, *J. Comput. Phys.* **41** (1981), 357–388.
2. A. Brandt, *Math. Comp.* **31** (1977), 333–390.
3. R. T. Delves, *in* "Crystal Growth" (B. R. Pamplin, Ed.), Pergamon Press, Oxford, 1975.
4. C. M. Eliott, *J. Inst. Math. Appl.* **25** (1980), 121–131.
5. C. M. Elliott and V. Janovský, *in* "1978 Proceedings of Mafelap." (J. R. Whiteman, Ed.), Academic Press, London, 1979.
6. D. Kinderlehrer and G. Stampacchia, "An Introduction to Variational Inequalities and Their Applications," Academic Press, New York, 1980.
7. J. A. McGeough, "Principles of Electrochemical Machining," Chapman & Hall, London, 1974.
8. G. H. Meyer, *SIAM J. Numer. Anal.* **18** (1981), 150–164.
9. J. R. Ockendon, *in* "Moving Boundary Problems" (D. G. Wilson, A. D. Solomon, and P. T. Boggs, Eds.), Academic Press, New York, 1978.
10. A. E. Scheidegger, "The Physics of Flow Through Porous Media," Univ. of Toronto Press, Toronto, 1960.
11. D. G. Wilson, A. D. Solmon, and P. T. Boggs (Eds.), "Moving Boundary Problems," Academic Press, New York, 1978.