JOURNAL OF COMPUTATIONAL PHYSICS 65, 448-461 (1986)

The Numerical Solution of Elliptic Free Boundary Problems Using Multigrid Techniques

ROLAND HUNT

Department of Mathematics, University of Strathclyde, Glasgow, Scotland

Received June 19, 1985; revised December 12, 1985

A multigrid algorithm is developed for the numerical solution of elliptic free boundary problems. The domain of the problem is mapped onto a rectangle and the governing equations discretized using finite differences. The resulting algebraic system is solved iteratively using a multigrid V cycle. For a convergent relaxation procedure it is necessary to use line iteration perpendicular to the free boundary simultaneously altering the values of the dependent variable and the position of the boundary, which is conveniently done using a single Newton iteration. Three problems are considered, a Poisson type problem, a steady state heat transfer problem, and one from electrochemical machining. The first two problems rapidly converge in a few multigrid cycles, the third converges less rapidly though adequately. Since full multigrid (FMG) is used, the results on the three finest grids could be combined to give accurate results of sixth order. © 1986 Academic Press, Inc.

1. INTRODUCTION

Free boundary problems, in which the position of one or more boundaries is unknown, arise in a variety of problems. For example, in Ockendon and Hogkins [1] there are presented applications from the steel and glass industries, chemistry, biology, and astrophysics, which involve heat flow and diffusion with phase changes, or involve chemical reaction and absorption. At the free boundary there are two boundary conditions rather than one and hence the boundary data is overprescribed. It is this extra boundary condition which provides the necessary additional information not only to solve the equations but also to locate the position of the unknown boundary. New independent variables transform the equations such that the domain of the problem is transformed into a rectangle. Since the boundary is unknown this transformation is unknown and has to be solved simultaneously with the problem. The equations are discretized in the usual way and, since we restrict ourselves to elliptic problems, the resulting algebraic system needs to be solved iteratively.

In recent years there have been proposed many new fast solver algorithms such as fast Fourier transformation, incomplete lower-upper decomposition, for example; but one of the most competitive in terms of efficiency has been the multigrid

0021-9991/86 $3.00

technique. The multigrid algorithm has successfully been applied to elliptic problems using finite differences or finite elements with fixed boundaries of arbitrary geometry, singular perturbation problems, and integral equations [2, 3]. In this paper we demonstrate that the application of multigrid techniques to free boundary problems results in a similarly efficient procedure. Finding a convergent relaxation procedure was the only real difficulty encountered. It was found necessary to relax the position of the free boundary simultaneously with the dependent variable. The best way of accomplishing this is to use line interation which includes a single nodal value of the boundary position. The algorithm is illustrated by the solution of three problems namely a Poisson type problem, a steady state heat transfer problem, and one from electrochemical machining.

The technique incorporates the full multigrid algorithm (FMG) which can be easily adapted to produce results at every grid level. By applying Richardson extrapolation to the results from the three finest grids it is possible to produce results with sixth-order accuracy.

2. Numerical Procedure

The problem which we consider is illustrated in Fig. 1i. The boundary $0ABC$ consists of fixed elements $0A$, $0C$, and AB given by $y=0$, $x=0$, and $x=L$ respectively and a free element CB given by $y=g(x)$, where g is unknown and is determined by the problem. The governing equations of the problem are typically of the form

$$\begin{aligned}
&Lu=f, && 0<y<g(x), 0<x<L\\
&Fu=d, && y=0, 0\leqslant x\leqslant L\\
& && x=0, 0<y<g(0)\\
& && x=L. 0<y<g(L)\\
&\left.\begin{aligned}F_1u=d_1\\ F_2u=d_2\end{aligned}\right\} && y=g(x), 0\leqslant x\leqslant L
\end{aligned}\tag{1}$$

where L, F, F_1, and F_2 are linear operators and f, d, d_1, and d_2 are known functions of x and y. The equations are assumed sufficient for the solutions of unknowns $u(x, y)$ and $g(x)$.

The region $0ABC$ is transformed onto a rectangle using the equations

$$\xi=x, \eta=\frac{y}{g(x)}\tag{2}$$

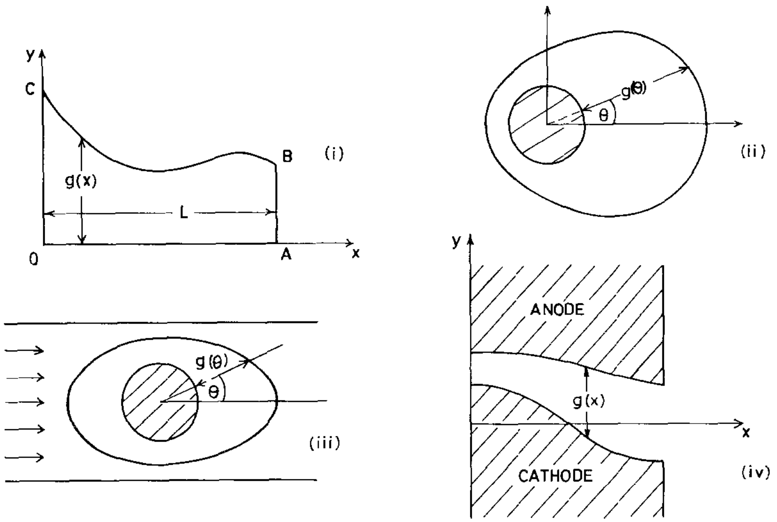

FIG. 1. (i) The typical general problem, (ii) the Poisson type problem, (iii) the flow past a refrigerated pipe, and (iv) the electrolytic cell used for machining the anode.

to give equations of the form

$$\begin{array}{ll} L'u=f' & 0<\eta<1,\, 0<\xi<L \\ F'u=d' & \eta=0,\, 0\leqslant\xi\leqslant L \\ & \xi=0,\, 0<\eta<1 \\ & \xi=L,\, 0<\eta<1 \\ \left.\begin{array}{l} F_1'u=d_1' \\ F_2'u=d_2' \end{array}\right\} & \eta=1,\, 0\leqslant\xi\leqslant L. \end{array} \tag{3}$$

where the primed quantities not only depend on ξ and η but also on g and its derivatives.

A mesh of dimension $n\times m$ is placed on the rectangle given by

$$\begin{array}{lll} \xi_i=ih, & i=0,\, 1,\ldots,\, n, & h=L/n \\ \eta_j=jk, & j=0,\, 1,\ldots,\, m, & k=1/m. \end{array} \tag{4}$$

If we let U_{ij} and G_i be approximations for $u(ih,jk)$ and $g(ih)$ respectively then Eqs. (3) are replaced by the numerical equations

$$L^{hk}U_{ij}=f_{ij}^k \qquad i=1,2,\ldots,n-1;\, j=1,2,\ldots,m-1 \tag{5a}$$

$$F^{hk}U_{ij} = d^h_{ij} \qquad \begin{array}{l} j=0,\ i=0,1,\dots,n \\ i=0,\ j=1,2,\dots,m-1 \\ i=n,\ j=1,2,\dots,m-1 \end{array} \tag{5b}$$

$$\left.\begin{array}{l} F_1^{hk}U_{ij} = d^h_{1,ij} \\ F_2^{hk}U_{ij} = d^h_{2,ij} \end{array}\right\} \qquad j=m,\ i=0,1,\dots,n. \tag{5c}$$

where L^{hk}, F^{hk}, F_1^{hk}, and F_2^{hk} are numerical operators and f^h, d^h, d_1^h, and d_2^h can depend on h via the numerical differencing of the derivatives of g.

Equations (5) can be solved using the usual multigrid techniques, the only difficulty is in obtaining a suitable relaxation procedure. The obvious procedure might involve the relaxation of U_{ij} using Eqs. (5a), (5b) and the first equation in (5c). A second sweep would then be used to relax G_i using the second equation in (5c). This procedure is, however, unreliable since changes in G_i introduced by the second sweep introduce high frequency errors to the residuals relaxed in the first sweep. Brandt and Dinar [4] introduced the notion of distributive relaxation to overcome this type of problem. This would involve changes in U_{ij} and G_i during the second sweep constructed in such a way that the residuals relaxed in the first sweep remain unchanged. An effect similar to that achieved by the distribution of residuals is readily obtained using line relaxation. For given i the unknown G_i appears in the coefficients of the difference equations (5) only at nodes on the lines $\xi = jh, j = i-1, i, i+1$. This suggests that the difference equations on $\xi = ih$ should be relaxed simultaneously and hence η-line relaxation is natural for the problem. Accordingly we solve the complete set of non-linear equations for

$$\{U_{ij}, j = 0, 1, \dots, n; G_i\} \tag{6}$$

at location i using Newton iteration. It is found in practice that one Newton iteration is sufficient during each sweep.

The multigrid procedure is then more or less standard. The coarse grid structure can use standard coarsening in which the grids have cell sizes $h \times k$, $2h \times 2k$, $4h \times 4k, \dots$, etc., or, since η-line relaxation is used, one can use ξ-coarsening in which the grids have cell sizes $h \times k$, $2h \times k$, $4h \times k, \dots$, etc. Restriction from fine to coarse may use simple injection, half or full weighting, and prolongation from coarse to fine uses the usual bilinear interpolation. As we will see the process is rapidly convergent using the V cycle, hence W and F cycles are unnecessary.

3. Test Problems

To examine the algorithm proposed in Section 2 we will consider three problems, first a simple problem with known analytical solution, second, a heat transfer problem [5] and finally a problem from electrochemical machining [6].

1. *Poisson Type Problem*

This problem is suggested in part by semi-conductor studies and is depicted in Fig. 1ii. The governing equations are

$$\begin{array}{ll} \dfrac{1}{r}\dfrac{\partial}{\partial r}\left(r\dfrac{\partial u}{\partial r}\right)+\dfrac{1}{r^2}\dfrac{\partial^2 u}{\partial \theta^2}=f(r,\theta) & \\ u(r,\theta)=\alpha(\theta) & r=1 \\ \left.\begin{array}{l} u(r,\theta)=\beta(\theta) \\ \dfrac{\partial u}{\partial n}(r,\theta)=\gamma(r,\theta) \end{array}\right\} & r=1+g(\theta). \end{array} \tag{7}$$

for which (r, θ) are polar coordinates and $\partial/\partial n$ denotes the normal derivative at the boundary. If the solution is given by

$$\begin{aligned} u(r,\theta)&=(r-2)^3\, e^{-\cos\theta} \\ g(\theta)&=4+\cos\theta \end{aligned} \tag{8}$$

then f, α, β, and γ are given by

$$\begin{aligned} f(r,\theta)&=\frac{(r-2)\,e^{-\cos\theta}}{r^2}\,[3r(3r-2)+(r-2)^3(\sin^2\theta+\cos\theta)] \\ \alpha(\theta)&=-e^{-\cos\theta}, \qquad \beta(\theta)=(2+\cos\theta)^3\,e^{-\cos\theta} \\ \gamma(r,\theta)&=\frac{(r-2)^2\,e^{-\cos\theta}}{r\sqrt{r^2+\sin^2\theta}}\,[3r^2+(r-2)\sin^2\theta]. \end{aligned} \tag{9}$$

2. *Steady State Heat Transfer*

The second problem describes the final build-up of an ice jacket around a refrigerated pipe of radius 1 and temperature -1 in a channel of width 4 delivering water at a constant rate (Fig. 1iii). The equations to be solved are

$$\begin{array}{ll} \dfrac{1}{r}\dfrac{\partial}{\partial r}r\dfrac{\partial u}{\partial r}+\dfrac{1}{r^2}\dfrac{\partial^2 u}{\partial \theta^2}=0 & \\ u(r,\theta)=-1 & \text{on} \quad r=1 \\ \left.\begin{array}{l} u(r,\theta)=0 \\ \dfrac{\partial u}{\partial n}(r,\theta)=\lambda\dfrac{(2-\cos\theta)}{2-r\,|\sin\theta|} \end{array}\right\} & r=1+g(\theta) \end{array} \tag{10}$$

where u is the ice temperature and λ the heat transfer coefficient. The outer boundary conditions contain a discontinuous derivative at $\theta = 0, \pi$; otherwise the problem is similar in type to the first problem.

Equations (7) and (10) can be transformed into the rectangle $0 \leqslant \xi \leqslant \pi$, $0 \leqslant \eta \leqslant 1$ using the equations

$$\begin{aligned} \theta &= \xi \\ r &= 1 + g(\xi)\,\eta \end{aligned} \tag{11}$$

to give

$$\begin{aligned} &[1 + 2g\eta + (g'^2 + g^2)]\frac{\partial^2 u}{\partial \eta^2} + [g + \eta(g + 2g'^2 - g'')]\frac{\partial u}{\partial \eta} \\ &\quad - 2\eta g g' \frac{\partial^2 u}{\partial \eta\, \partial \xi} + g^2 \frac{\partial^2 u}{\partial \xi^2} = g^2(1 + g\eta)^2 f(1 + g\eta, \xi) \\ &\qquad u = \alpha(\xi), \qquad \eta = 0, \qquad 0 \leqslant \xi \leqslant \pi \end{aligned}$$

$$\left.\begin{array}{ll} & u = \beta(\xi) \\ \text{Problem 1} & \dfrac{\partial u}{\partial \eta} = 3g(g-1)^2 e^{-\cos \xi} \\ \text{Problem 2} & \dfrac{\partial u}{\partial \eta} = \lambda \dfrac{(2 - \cos \xi)\, g(g+1)}{(2 - \sin \xi[g+1])([g+1]^2 + f'^2)^{1/2}} \end{array}\right\} \quad \eta = 1, 0 \leqslant \xi \leqslant \pi \tag{12}$$

$$\frac{\partial u}{\partial \xi} = 0, \ \xi = 0 \text{ and } \pi, \ 0 \leqslant \eta \leqslant 1$$

where the last equation is a result of the problems' being symmetrical about $\xi = 0$ and $\xi = \pi$. For Problem 1 f, α, and β are as given by Eq. (9) and for Problem 2 $f = 0$, $\alpha = -1$, and $\beta = 0$.

3. *Electrochemical Machining*

In the final problem the piece of metal which is to be shaped forms the anode of an electrolytic cell (Fig. 1iv) which dissolves when a current is passed between anode and cathode. The cathode, which is made of metal uncorruptable by the electrolyte, is moved towards the anode at a constant rate such that in a steady state the gap, given by $g(x)$, is independent of time. The shape of the cathode, the curve $y = f(x)$, determines the shape of the anode. Thus for given $f(x)$ we wish to determine $g(x)$ from which the shape of the anode is simply $f + g$. The governing equations are

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

$$\left.\begin{array}{lll} \dfrac{\partial u}{\partial x}=0 & \text{on} & x=0 \text{ and } x=1 \\ u=0 & \text{on} & y=f(x) \\ \left.\begin{array}{l} u=1 \\ \dfrac{\partial u}{\partial n}=v \end{array}\right\} & \text{on} & y=f(x)+g(x) \end{array}\right. \qquad (13)$$

where u is the electric potential and v is a constant related to the recession rate. For compatibility reasons we assume that

$$f'(x)=g'(x)=0 \qquad \text{at} \quad x=0 \text{ and } 1. \qquad (14)$$

Equations (13) and (14) can be transformed onto the square $0\leqslant\xi\leqslant 1$, $0\leqslant\eta\leqslant 1$ using the equations

$$\begin{aligned} x&=\xi \\ y&=f(\xi)+\eta g(\xi) \end{aligned} \qquad (15)$$

to give

$$\begin{gathered} [(f'+\eta g')^2+1]\frac{\partial^2 u}{\partial\eta^2}+[2f'g'-f''g+\eta(2g'^2-gg'')]\frac{\partial u}{\partial\eta} \\ -2(f'+\eta g')\,g\frac{\partial^2 u}{\partial\xi\,\partial\eta}+g^2\frac{\partial^2 u}{\partial\xi^2}=0 \\ u=0, \qquad \eta=0, \qquad 0\leqslant\xi\leqslant 1 \\ \left.\begin{array}{l} u=1 \\ \dfrac{\partial\phi}{\partial\eta}=\dfrac{vg}{(1+[f'+g']^2)^{1/2}} \end{array}\right\}, \qquad \eta=1.\ 0\leqslant\xi\leqslant 1 \\ \partial\phi/\partial\xi=0, \qquad \xi=0 \text{ and } 1, \qquad 0\leqslant\eta\leqslant 1. \end{gathered} \qquad (16)$$

4. Implementation

An $n\times n$ mesh is placed on the rectangle $0\leqslant\xi\leqslant L$, $0\leqslant\eta\leqslant 1$. where $L=\pi$ for Problems 1 and 2 and $L=1$ for Problem 3, with $n=2^m$ to facilitate the multigrid algorithm. A typical mesh point (ξ_i, η_j) is given by (ik, jh), $i, j=0, 1,..., n$, where $K=L/n$ and $h=1/n$. In order that we may use central differencing to represent derivative boundary conditions we includethe fictitious mesh points $(-k, jh)$, $(1+k, jh)$, $j=0, 1,..., n$, and $(ik, 1+h)$, $i=-1, 0,..., n+1$.

Equations (12) and (16) can be represented by

$$
\begin{aligned}
&A\frac{\partial^2 u}{\partial \eta^2}+B\frac{\partial u}{\partial \eta}+C\frac{\partial^2 u}{\partial \xi\,\partial \eta}+D\frac{\partial^2 u}{\partial \xi^2}+E=0 \\
&u=\alpha(\xi), \qquad \eta=0 \\
&\left.\begin{aligned} &u=\beta(\xi) \\ &\frac{\partial u}{\partial \eta}=\gamma(\xi, g, g') \end{aligned}\right\} \qquad \eta=1, \frac{\partial u}{\partial \xi}=0, \xi=0, L
\end{aligned}
\tag{17}
$$

where A, B, C, D, and E can depend on ξ, η, g, g', and g'' but not on u. All derivatives were represented by central differences and hence Eqs. (17) are replaced by the algebraic system

$$
\begin{aligned}
\phi_{ij} \equiv{}& \frac{A_{ij}}{h^2}(u_{i,j+1}-2u_{ij}+u_{i,j-1})+\frac{B_{ij}}{2h}(u_{i,j+1}-u_{i,j-1}) \\
&+\frac{C_{ij}}{4hk}(u_{i+1,j+1}-u_{i-1,j+1}-u_{i+1,j-1}+u_{i-1,j-1}) \\
&+\frac{D_{ij}}{k^2}(u_{i+1,j}-2u_{ij}+u_{i-1,j})+E_{ij}=0. \\
&j=1, 2,..., n, \qquad i=0, 1,..., n
\end{aligned}
\tag{18}
$$

$$
\left.\begin{aligned}
&u_{i0}=\alpha(\xi_i)\equiv\alpha_i \\
&u_{in}=\beta(\xi_i)\equiv\beta_i \\
&\phi_{i,n+1}\equiv\frac{1}{2h}(u_{i,n+1}-u_{i,n-1})-\gamma\left(\xi_i, g_i, \frac{1}{2h}(g_{i+1}-g_{i-1})\right)=0 \\
&u_{-1,j}=u_{1j}, \qquad u_{n+1,j}=u_{n-1,j}, \qquad j=0, 1,..., n+1
\end{aligned}\right\} \quad i=0, 1,..., n
\tag{19}
$$

where $g_i \equiv g(\xi_i)$ and A_{ij}, B_{ij}, C_{ij}, D_{ij}, and E_{ij} are A, B, C, D, and E evaluated using ξ_i, η_j, g_i, $(1/2h)(g_{i+1}-g_{i-1})$, and $(1/h^2)(g_{i+1}-2g_i+g_{i-1})$, respectively.

Equations (18) are solved using η-line relaxation, that is, for a given i the variables $u_{i,j}$, $j=0, 1,..., n+1$, and g_i are solved simultaneously using Newton's iteration, i.e.,

$$
\begin{aligned}
&u_{ij}^{(s+1)}=u_{ij}^{(s)}+\Delta u_{ij}^{(s)} \qquad s=0, 1, 2,..., \\
&g_i^{(s+1)}=g_i^{(s)}+\Delta g_i^{(s)}
\end{aligned}
\tag{20}
$$

where $\Delta u_{ij}^{(s)}$ and $\Delta g_i^{(s)}$ are found by solving the matrix equation.

$$\begin{bmatrix} b_1 & c_1 & & & & d_1 \\ a_2 & b_2 & c_2 & & 0 & d_2 \\ \ddots & \ddots & \ddots & & & \vdots \\ & a_{n-2} & b_{n-2} & c_{n-2} & & d_{n-2} \\ & & a_{n-1} & b_{n-1} & 0 & d_{n-1} \\ & 0 & & a_n & c_n & d_n \\ & & & a_{n+1} & c_{n+1} & d_{n+1} \end{bmatrix} \begin{bmatrix} \Delta u^{(s)}_{(\imath)} \\ \Delta u^{(s)}_{i2} \\ \vdots \\ \Delta u^{(s)}_{i,n-2} \\ \Delta u^{(s)}_{i.n-1} \\ \Delta u^{(s)}_{i,n+1} \\ \Delta g^{(s)}_i \end{bmatrix} = - \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{n-2} \\ \phi_{n-1} \\ \phi_n \\ \phi_{n+1} \end{bmatrix} \tag{21}$$

where

$$\left.\begin{aligned} a_j &\equiv \frac{\partial \phi_{ij}}{\partial u_{j+1}} = \frac{A_{ij}}{h^2} + \frac{B_{ij}}{2h} \\ b_j &\equiv \frac{\partial \phi_{ij}}{\partial u_j} = -2\left(\frac{A_{ij}}{h^2} + \frac{D_{ij}}{k^2}\right) \\ c_i &\equiv \frac{\partial \phi_{ij}}{\partial u_{j-1}} = \frac{A_{ij}}{h^2} - \frac{B_{ij}}{2h} \end{aligned}\right\} \qquad j = 1, 2, ..., n \tag{22}$$

$$a_{n+1} = -c_{n+1} = \frac{1}{2h}$$

$$d_j = \frac{\partial \phi_{ij}}{\partial g_i}, \qquad j = 1, 2, ..., n+1;$$

u_{i0} and u_{in} have been replaced by α_i and β_i and hence $\Delta u^{(s)}_{i0}$ and $\Delta u^{(s)}_{in}$ do not appear in the system (18). Rather than working out the d_j analytically which can prove very tedious and error prone, it is easier to calculate it numerically using

$$\frac{\partial \phi_{ij}}{\partial g_\imath} \simeq \frac{1}{\varepsilon} \left[\phi_{ij}(g_i + \varepsilon, \mathbf{w}_{ij}) - \phi_{ij}(g_i, \mathbf{w}_{ij})\right] \tag{23}$$

where $\mathbf{w}_{ij}$ represents all other variables contained in ϕ_{ij} other than g_i. The value of ε is chosen to be sufficiently small to ensure reasonably accuracy of (23) without incurring any significant round-off error, in practice ε was set to 10^{-6}. Solutions of Eqs. (21) can be found very efficiently since the system is nearly tridiagonal.

The multigrid algorithm employed uses η-zebra line relaxation employing a single Newton iterate. Coarsening is in the ξ-direction only, i.e., the coarse grids have mesh size $2^i \times 2^m$, $i = 1, 2, ..., m$ with m usually set at 6 giving a 64×64 fine mesh. This coarsening permits us to use centered differencing along the $\eta = 1$ boundary at all levels. Restriction uses full weighting and prolongation bilinear interpolation. V cycles were used throughout with convergence after 6–12 iterations depending on the problem. The full multigrid algorithm was employed starting from a 4×4 grid, the initial state of the next finest grid being effected from the current grid using bilinear interpolation.

Problems 2 and 3 only converged if $\lambda \geqslant 1$ or $\nu \leqslant 2$. For other values of λ and ν it was necessary to use continuation which was accomplished in the following manner. For Problem 2 solutions were sought for, $\lambda = 10^d$, where $d = 1, \frac{1}{2}, \ldots, -2$. The solution for $\lambda = 10$ is convergent and is obtained as described above. The case $\lambda = 10^{d-1/2}$ can be obtained from the result for $\lambda = 10^d$ by dividing the current value of λ by $10^{-1/12}$ and applying a single V cycle. This was applied 6 times bringing the value of λ to $10^{d-1/2}$. The V cycle was then applied a further 6 to 12 times to obtain convergence. For Problem 3 solutions were sought for $\nu = 2^d$, $d = 0, \frac{1}{2}, \ldots, 3$ and a similar continuation process employed starting from $\nu = 1$.

5. Results

In each case the V cycle of the multigrid iteration performed a single relaxation sweep between fine to coarse and coarse to fine transfers (i.e., $\nu_1 = \nu_2 = 1$ in the usual notation). The error was monitored by calculating the root mean square change in u_{ij} resulting from the last relaxation sweep of the V cycle which occurs at the finest level. Let RMS(r) denote the error after r multigrid iteration cycles then we define the rate of convergence ρ by

$$\rho = \frac{\text{RMS}(N)}{\text{RMS}(N-1)} \tag{24}$$

where N is the number of cycles performed. The value of N lies in the range $6 \leqslant N \leqslant 12$ and is chosen such that RMS(N) is of the order 10^{-8}. A value of ρ equal to 0.1 indicates that each V cycle gives an extra decimal place accuracy in the computed results.

Problem 1. Results were obtained for mesh sizes 4×4, $8 \times 8, \ldots, 64 \times 64$, and the values of ρ obtained are shown in Table I. They are typically of the order of 0.1 and hence the multigrid algorithm gives satisfactory rapid convergence for the solution

TABLE I

Convergence Rates for Problem 1 with Error in g at $\theta = \pi/2$

Mesh size	ρ	Error in $g(\pi/2)$
4×4	0.060	0.1674
8×8	0.101	0.0380
16×16	0.115	0.0093
32×32	0.120	0.0023
64×64	0.111	0.0006

of the difference equations (18). Since the exact solution is known (Eqs. (8)) we can find the error in the numerical results, the errors for $g(\theta)$ at $\theta = \pi/2$ have been given in Table I. It will be noticed that these errors vary as h^2 which is as expected since the differencing is second order. In fact since central differences were used throughout the error dependence on h can be represented as a power series in h^2 and hence one can use the results on different grid sizes to obtain highly accurate results using Richardson's extrapolation. By extrapolating the results at common locations from the 16×16, 32×32, and 64×64 meshes we obtain solutions of sixth-order accuracy. For example, the results for $g(\theta)$ at $\theta = 0$, $\pi/2$, and π can be set in a Romberg type table as follows:

5.011845712		
	4.999983015	
5.002948689		5.000000005
	4.999998943	
5.000736379		
4.009260721		
	3.999980554	
4.002300596		4.000000001
	3.999998785	
4.000574238		
3.011140293		
	2.999990320	
3.002777814		3.000000012
	2.999999406	
3.000694008		

Since the exact solutions are 5, 4, and 3 respectively we have 8 decimal place accuracy and the average error over all 17 locations is 4.5×10^{-9}. Since the full multigrid algorithm (FMG) obtains results for various mesh spacings it is natural to employ this type of extrapolation to obtain results of high accuracy. Another possibility for improving the accuracy is to use τ-extrapolation [3].

Problem 2. Results were obtained for three mesh sizes 16×16, 32×32, and 64×64 for seven values of λ lying in the range $0.01 \leqslant \lambda \leqslant 10$ (for the case $\lambda = 0.01$ it was necessary to use Newton line iteration to convergence rather than employing a single iterate). The values of ρ are given in Table II and are on average about 0.1 and in every case less than 0.3. Thus we have a reasonably rapid convergence rate in spite of having discontinuous derivatives in the boundary condition at $\theta = 0$ and $\theta = \pi$. The results are shown graphically in Fig. 2 and are in agreement with those

TABLE II

Convergence Rates for Problem 2 for Various Values of λ

Mesh size	10	3.162	1	0.3162	0.1	0.03162	0.01
16×16	0.033	0.039	0.068	0.103	0.150	0.163	0.152
32×32	0.042	0.046	0.083	0.119	0.177	0.220	0.261
64×64	0.075	0.076	0.099	0.122	0.184	0.240	0.288

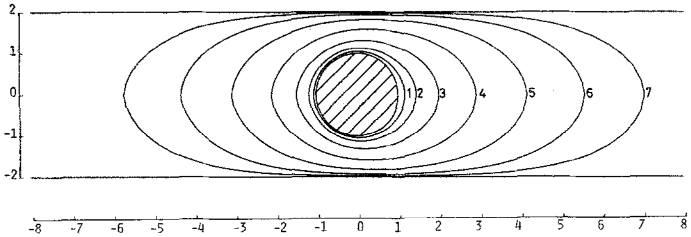

FIG. 2. Ice surfaces around a refrigerated pipe. Curves $k = 1, 2, \ldots, 7$ are for values of the heat transfer coefficient $\lambda = 10^{(3-k)/2}$.

calculated by Meyer [5]. The extrapolation process described for Problem 1 can be applied to the results from the three grids giving results of high accuracy. However, at or near $\theta = 0$ or $\theta = \pi$, this high accuracy is impaired because of the discontinuous derivative in the boundary condition. By considering differences between entries in the extrapolation tables (25) the accuracy in the results away from the boundary is about 8 decimal places and 4 decimals at the boundary.

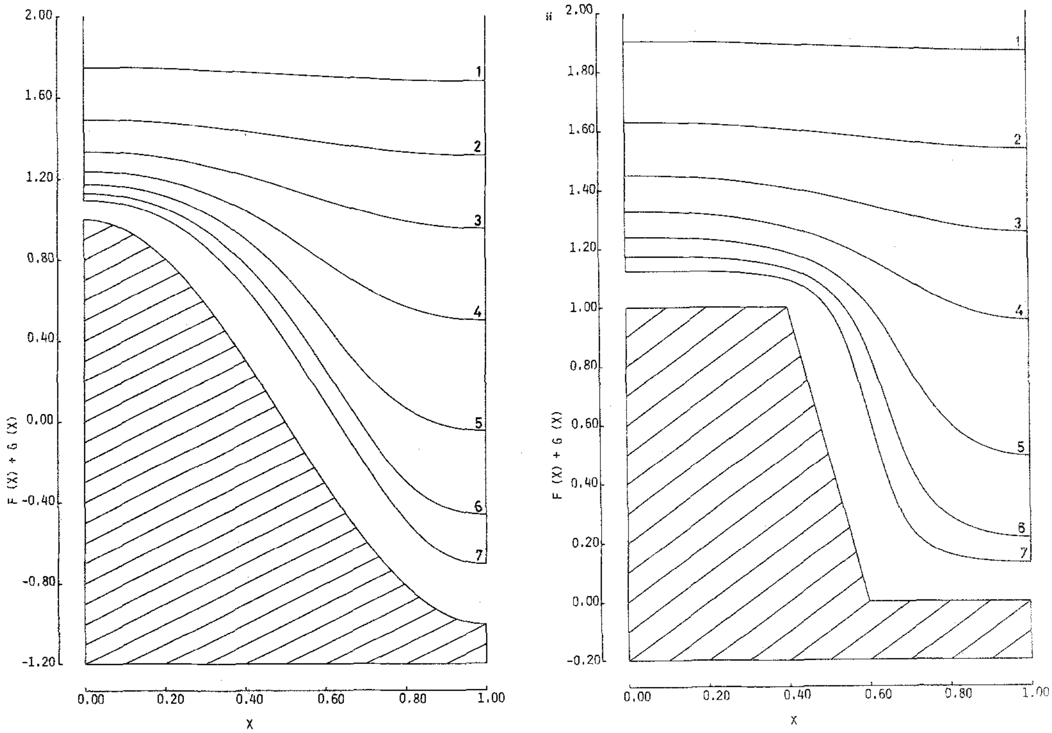

FIG. 3. (i) Anode surfaces for a cathode given by $f(x) = \cos x$. Curves $k = 1, 2, \ldots, 7$ are for values of $\nu = 2^{(k-1)/2}$. (ii) Anode surfaces for a cathode given by $f(x) = 1$, $0 \leqslant x < 0.4$; $f(x) = 3 - 5x$, $0.4 \leqslant x \leqslant 0.6$; $f(x) = 0$, $0.6 < x \leqslant 1$. Curves $k = 1, 2, \ldots, 7$ are for values of $\nu = 2^{(k-1)/2}$.

TABLE III

Convergence Rates for Problem 3 for Various Values of ν

	Mesh size	1	1.414	2	2.828	4	5.657	8
(i)	16 × 16	0.243	0.219	0.250	0.387	0.441	0.364	0.206
	32 × 32	0.394	0.379	0.381	0.440	0.514	0.472	0.364
	64 × 64	0.492	0.465	0.467	0.534	0.635	0.611	0.537
(ii)	16 × 16	0.274	0.373	0.556	0.512	0.445	0.207	0.149
	32 × 32	0.344	0.500	0.645	0.681	0.527	0.442	0.373
	64 × 64	0.549	0.545	0.679	0.734	0.604	0.653	0.581

PROBLEM 3. For this problem the shape of the cathode $f(x)$ needs to be specified. Results are obtained for

(i) $f(x) = \cos \pi x$, $0 \leqslant x \leqslant 1$

(ii) $f(x) = 1$, $0 \leqslant x < 0.4$; $= 3 - 5x$, $0.4 \leqslant x \leqslant 0.6$; $= 0$, $0.6 < x \leqslant 1$;

representing a continuous function and one with sharp corners (derivatives undefined at $x = 0.4$ and $x = 0.6$). The results, shown graphically in Figs. 3, have been obtained for seven values of ν lying in the range $1 \leqslant \nu \leqslant 8$ for three mesh sizes 16×16, 32×32, and 64×64. The results for $f(x) = \cos \pi x$, $\nu = 1$ are in agreement with those obtained by Sloan [5]. The values of ρ are given in Table III and give a much poorer rate of convergence than for Problems 1 and 2 with maximum values for ρ of 0.635 for (i) and 0.734 for (ii) with average values of 0.419 for (i) and 0.494 for (ii). Thus typically three iterations are required to reduce the error by an order of magnitude compared to one iteration in Problems 1 and 2. However since the full multigrid algorithm was used, only 12 multigrid iterations at the finest level was required to produce an RMS error of order 10^{-6} and 22 iterations for 10^{-9} which is acceptable on modern computers. The difficulty probably lies in the problem; Sloan using global Newton iteration incorporating continuation found the convergence rate very slow. The results from the three grids can be extrapolated to produce high accuracy results. However, only for case (i) for $\nu = 1$, $\sqrt{2}$, 2, $2\sqrt{2}$ could be extrapolated results be regarded as reliable and for these values of ν the accuracy was about 6 decimal places.

6. CONCLUSIONS

Using Newton line relaxation it has been demonstrated that elliptic free boundary problems can be solved successfully using multigrid techniques, the rate of convergence being "good" for Problems 1 and 2 ($\rho \sim 0.1$) and "acceptable" for Problem 3 ($\rho \sim 0.5$). Since the full multigrid algorithm produces results having grids of various mesh spacings it is natural to extrapolate the results from these grids to produce results of high accuracy.

Acknowledgment

I should like to express my gratitude to Dr. D. M. Sloan for introducing me to this problem and his constant encouragement throughout.

References

1. J. R. Ockendon and W. R. Hodgkins, *Moving Boundary Problems in Heat Flow and Diffusion* (Oxford Press (Clarendon), London/New York, 1975)
2. A. Brandt, *Math. Comp.* **31** (1977), 333.
3. W. Hackbusch and U. Trottenberg, *Multigrid Methods*, Lecture Notes in Mathematics Vol. 960. (Springer-Verlag, Berlin/Heidelberg/New York, 1982).
4. A. Brandt and N. Dinar, *Numerical Methods in Partial Differential Equations*, edited by J. H. Bramble (Academic Press, London/New York, 1979).
5. G. H. Meyer, *J. Inst. Math. Appl.* **20** (1977), 317.
6. D. M. Sloan, *J. Comput. Appl. Math.* **14** (1986), 279.