SIAM J. SCI. STAT. COMPUT.
Vol. 4, No. 4, December 1983

0196-5204/83/0404-0008 $01.25/0

MULTIGRID ALGORITHMS FOR THE SOLUTION OF LINEAR COMPLEMENTARITY PROBLEMS ARISING FROM FREE BOUNDARY PROBLEMS*

ACHI BRANDT† AND COLIN W. CRYER‡

Abstract. Several free boundary problems (including saturated-unsaturated flow through porous dams, elastic-plastic torsion and cavitating journal bearings) can be formulated as linear complementarity problems of the following type: Find a nonnegative function u which satisfies prescribed boundary conditions on a given domain and which, furthermore, satisfies a linear elliptic equation at each point of the domain where u is greater than zero. We show that the multigrid FAS algorithm, which was developed by Brandt to solve boundary value problems for elliptic partial differential equations, can easily be adapted to handle linear complementarity problems. For large problems, the resulting algorithm, PFAS (projected full approximation scheme) is significantly faster than previous single-grid algorithms, since the computation time is proportional to the number of grid points on the finest grid.

We then introduce two further multigrid algorithms, PFASMD and PFMG. PFASMD is a modification of PFAS which is considerably faster than PFAS. Using PFMG (projected full multigrid) it is possible to solve a linear complementarity problem to within truncation error using less work than the equivalent of seven Gauss–Seidel sweeps on the finest grid.

AMS (MOS) subject classifications. 35J65, 35R35, 65N99, 90C33

Key words. multigrid algorithms, free boundary problems, linear complementarity problems

1. Introduction. Several free boundary problems can be reformulated as an (infinite-dimensional) LCP (linear complementarity problem): Given a domain $\Omega \subset R^n$ with boundary $\partial\Omega$, and given functions f and g, find u (defined on Ω) such that (in an appropriate weak sense)

$$
\begin{aligned}
&\text{(a)} && \mathcal{L}u(x) \leqq f(x), && x \in \Omega,\\
&\text{(b)} && u(x) \geqq 0, && x \in \Omega,\\
&\text{(c)} && u(x)[\mathcal{L}u(x) - f(x)] = 0, && x \in \Omega,\\
&\text{(d)} && u(x) = g(x), && x \in \partial\Omega,
\end{aligned}
\tag{1.1}
$$

where $\mathcal{L}$ is a given second order elliptic operator. We do not write (1.1a) in the more usual form $-\mathcal{L}u(x) + f(x) \geqq 0$ because we wish to maintain compatibility with the notation in previous papers by Brandt.

Well-known examples of free boundary problems which can be written in the form (1.1) include porous flow through dams (a recent reference is Baiocchi [1978]), journal bearing lubrication (Cryer [1971a], Cimatti [1977]) and elastic-plastic torsion (Cea, Glowinski and Nedelec [1974], Lanchon [1974], Cryer [1980]). General references include: Duvaut and Lions [1976], Glowinski, Lions and Tremolieres [1976], Cryer [1977], Glowinski [1978], Cottle, Giannessi and Lions [1980], and Kinderlehrer and Stampacchia [1980].

* Received by the editors January 16, 1981, and in revised form July 15, 1982.

† Department of Applied Mathematics, the Weizmann Institute of Science, Rehovot, Israel. The research of this author was sponsored by the United States Army under contract DAAG29-80-C-0041 and contract DAJ37-79-C-0504.

‡ Computer Sciences Department and Mathematics Research Center, University of Wisconsin–Madison, Madison, Wisconsin 53706. Current address: Department of Applied Mathematics and Theoretical Physics, University of Cambridge. The research of this author was sponsored by the National Science Foundation under grant MCS77-26732 and by the United States Army under contract DAAG29-80-C-0041.

When (1.1) is approximated using finite differences on a grid G, one obtains a (finite-dimensional) LCP:

$$
\begin{aligned}
&\text{(a)} && LU(x) \leqq f(x), && x \in G,\\
&\text{(b)} && U(x) \geqq 0, && x \in G,\\
&\text{(c)} && U(x)[LU(x) - f(x)] = 0, && x \in G,\\
&\text{(d)} && U(x) = g(x) && x \in \partial G,
\end{aligned}
\tag{1.2}
$$

where $U(x)$ is an approximation to $u(x)$ at the grid points $x \in G \cup \partial G$ and where L is a difference operator which approximates $\mathscr{L}$. The coefficients of L are $O(h^{-2})$, where h denotes the grid length.

By multiplying (1.2) by h^2 and eliminating the known values of $U(x)$ on ∂G, the LCP (1.2) may be written in matrix form:

$$
\begin{aligned}
&\text{(a)} && AU \leqq b,\\
&\text{(b)} && U \geqq 0,\\
&\text{(c)} && U^T(AU - b) = 0,
\end{aligned}
\tag{1.3}
$$

where U is the N-vector of values of $U(x)$ on G, and A is an $N \times N$ matrix with coefficients which are $O(1)$. We will assume that A is symmetric and negative definite.

For example, if $\mathscr{L}$ is the Laplace operator in R^2, then a possible choice for L would be the classical five-point difference operator, in which case A would be a matrix with diagonal elements -4 and off-diagonal elements either 0 or 1.

There is an extensive literature on the (finite-dimensional) LCP (see Balinski and Cottle [1978]). In particular, if A is negative definite, as we assume, then there exists a unique solution to (1.2) and (1.3).

Since the LCP (1.3) arises from a free boundary problem, the matrix A has special properties which make it possible to use specialized algorithms which are particularly efficient. Such algorithms include projected SOR (Cryer [1971], Glowinski [1971]) the method of Cottle and Sacher [1977], and the modified block SOR (MBSOR) method of Cottle, Golub and Sacher [1978]; Cryer [1980a] summarizes these algorithms, and Cottle [1974] gives numerical comparisons between them.

Recently, it has been found (Brandt [1977], Brandt and Dinar [1979]) that multigrid algorithms are an effective tool for solving linear equations of the form

$$AX = b. \tag{1.4}$$

The basic idea of these multigrid algorithms is to compute on a sequence of nested grids. The computation proceeds on a particular grid until the error becomes smooth and the rate of convergence slows, at which point the computation is transferred to a coarser grid. When the error has been reduced on the coarser grid, the solution on the finer grid is corrected using interpolated values from the coarser grid.

In this paper, we show how the multigrid algorithm FAS of Brandt can be modified to solve the LCP (1.3). We find that the modified multigrid algorithm, PFAS, is substantially faster than previous single-grid algorithms.

The paper is organized as follows. In § 2, we describe PFAS, the projected full approximation scheme for solving (1.3): PFAS combines the concepts of multigrid algorithms with those of projected SOR. In § 3, we discuss the implementation of PFAS, and in § 4, we give numerical results obtained using PFAS. In § 5, we discuss

alternative implementations of PFAS, the last of which (PFASMD) leads to substantially improved convergence. We also include several less successful implementations because they are instructive. In § 6, we describe results obtained using PFMG, the projected full multigrid algorithm. The basic idea of PFMG is to compute the initial approximation on each grid by interpolating an accurate solution on the next coarsest grid. Using PFMG we are able to solve a problem to within truncation error using less work than the equivalent of seven Gauss–Seidel sweeps on the finest grid. Our results are summarized in § 7, and some possible extensions are mentioned.

Listings of the programs used in this paper are given in Brandt and Cryer [1980].

2. PFAS (projected full approximation scheme). Brandt [1977], [1980], and Brandt and Dinar [1979] give a detailed exposition of multigrid methods and their philosophy, and the reader is referred to these papers for background information. The algorithm described below, PFAS, is a modification of the FAS (full approximation scheme) which is considered in Brandt [1977, § 5] and in Brandt and Dinar [1979, § 2.2].

The domain $\Omega \subset R^n$ is approximated by a sequence of grids

$$G^1 \subset G^2 \subset \cdots \subset G^M \subset R^n,$$

with corresponding grid sizes

$$h_1 = 2h_2 = 4h_3 = \cdots = 2^{M-1}h_M.$$

Let F^k be the restriction of f to G^k,

$$F^k(x) = f(x), \qquad x \in G^k. \tag{2.1}$$

Then, on G^k the difference equations (1.2) approximating (1.1) take the form

$$\begin{aligned}
&\text{(a)} && L^k U^k(x) \leqq F^k(x) && \text{in } G^k,\\
&\text{(b)} && U^k(x) \geqq 0 && \text{in } G^k,\\
&\text{(c)} && U^k(x)[L^k U^k(x) - F^k(x)] = 0 && \text{in } G^k,\\
&\text{(d)} && U^k(x) = g(x) && \text{in } \partial G^k.
\end{aligned} \tag{2.2}$$

Let the points of G^k be ordered: $x_1^k, x_2^k, \cdots, x_{N_k}^k \in G^k$, and let U^k be the vector

$$U^k = \{U_j^k : 1 \leqq j \leqq N_k\} \equiv \{U^k(x_j^k) : 1 \leqq j \leqq N_k\}.$$

Then, (1.3) takes the form

$$\begin{aligned}
&\text{(a)} && A^k U^k \leqq b^k,\\
&\text{(b)} && U^k \geqq 0,\\
&\text{(c)} && (U^k)^T [A^k U^k - b^k] = 0
\end{aligned} \tag{2.3}$$

where

$$A^k = \{a_{ij}^k : 1 \leqq i, j \leqq N_k\} \tag{2.4}$$

is a known sparse symmetric negative definite matrix and $b^k = \{b_j^k\}$ is a known vector with components $b_j^k = h_k^2 F^k(x_j^k)$ except at points x_j^k adjacent to ∂G^k.

2.1. The projected Gauss–Seidel algorithm. It is possible to solve the LCP's (2.2) and (2.3) using the *projected Gauss–Seidel algorithm* which we now describe.

Let $u^{k,0}(x)$ be an approximate solution of (2.2) and (2.3). We compute recursively a sequence of approximations $u^{k,1}(x), u^{k,2}(x), \cdots$. Let $u^{k,s-1}(x)$ be given. From (2.2d),

the boundary values of $u^{k,s}(x)$ are equal to $g(x)$. The interior values of $u^{k,s}(x)$, which together comprise the vector

$$u^{k,s} = \{u_j^{k,s} : 1 \leqq j \leqq N_k\} = \{u^{k,s}(x_j^k) : 1 \leqq j \leqq N_k\}, \tag{2.5}$$

are obtained, point by point, by first applying the classical Gauss–Seidel method to (2.3) to obtain

$$u_j^{k,s-1/2} = u_j^{k,s-1} + \left[b_j^k - \sum_{l<j} a_{jl}^k u_l^{k,s} - \sum_{l \geqq j} a_{jl}^k u_l^{k,s-1} \right] \Big/ a_{jj}^k = u_j^{k,s-1} + \tilde{r}_j^{k,s} / a_{jj}^k, \tag{2.6}$$

say, and then *projecting*:

$$u_j^{k,s} = \max\{0, u_j^{k,s-1/2}\}. \tag{2.7}$$

The process of applying (2.6) and (2.7) for $1 \leqq j \leqq N_k$ to obtain $u^{k,s}$ from $u^{k,s-1}$ will be called a *G^k-projected Gauss–Seidel sweep*, or a *G^k-projected sweep*. The quantities $\tilde{r}_j^{k,s}$ will be called the *dynamic residuals*.

It is known (Cryer [1971], Glowinski [1971]) that $u^{k,s} \to U^k$ as $s \to \infty$.

When implementing the projected Gauss–Seidel method only the latest values of the solution are stored. We will, therefore, often suppress the iteration counter s and denote one projected Gauss–Seidel sweep applied to (2.2) and (2.3) by

$$u^k \leftarrow \text{projected Gauss–Seidel } [u^k : L^k, F^k]. \tag{2.8}$$

Similarly,

$$\nabla u^k = u^{k,s} - u^{k,s-1} \tag{2.9}$$

will denote the difference between the latest approximation u^k and its predecessor, while

$$\nabla u_{\text{old}}^k = u^{k,s-1} - u^{k,s-2} \tag{2.10}$$

denotes the previous difference.

2.2. Error estimates for the projected Gauss–Seidel algorithm. When implementing the projected Gauss–Seidel algorithm as part of a multigrid process, it is important to be able to estimate the error. In order to do so, we note that since, by assumption, $-A^k$ is symmetric and positive definite, there exists a coercitivity constant $\alpha_k > 0$ such that

$$w^T(-A^k)w \geqq \alpha_k w^T w, \tag{2.11}$$

for all $w \in R^{N_k}$.

LEMMA 2.1. *Let U^k be the solution of the* LCP (2.3), *and let $u^k \geqq 0$ be an approximate solution. Let*

$$r^k = (r_j^k) = b^k - A^k u^k, \tag{2.12}$$

and $r_+^k = (r_{+j}^k)$, where

$$r_{+j}^k = \begin{cases} r_j^k & \text{if } u_j^k > 0, \\ \min\{0, r_j^k\} & \text{if } u_j^k = 0. \end{cases} \tag{2.13}$$

Then

$$(U^k - u^k)^T(-A^k)(U^k - u^k) \leqq (U^k - u^k)^T(-r_+^k). \tag{2.14}$$

Hence,

$$\|U^k - u^k\|_2 \leqq \alpha_k^{-1} \|r_+^k\|_2. \tag{2.15}$$

Proof. With r_+^k defined as above, we see that u^k satisfies the LCP:

$$\begin{aligned} &\text{(a)} && A^k u^k \leqq b^k - r_+^k, \\ &\text{(b)} && u^k \geqq 0, \\ &\text{(c)} && (u^k)^T (A^k u^k - b^k + r_+^k) = 0. \end{aligned} \tag{2.16}$$

Following Falk [1974], we multiply (2.3a) by the nonnegative vector $(u^k)^T$ and use the complementarity condition (2.3c) to obtain

$$(u^k - U^k)^T A^k U^k \leqq (u^k - U^k)^T b^k. \tag{*}$$

Similarly, multiplying (2.16a) by $(U^k)^T$ we obtain

$$(U^k - u^k)^T A^k u^k \leqq (U^k - u^k)^T (b^k - r_+^k). \tag{**}$$

Adding (*) and (**) and combining terms we obtain (2.14) and hence (2.15). □

LEMMA 2.2. *Let U^k be the solution of the* LCP (2.3), *and let $u^k \geqq 0$ be an approximate solution obtained after one or more G^k projected sweeps. Let*

$$A^k = (D^k - L^k - P^k) \tag{2.17}$$

where D^k is diagonal, and L^k and P^k are strictly lower and upper triangular matrices, respectively.

Then u^k satisfies the LCP

$$\begin{aligned} &A^k u^k \leqq b^k - P^k \nabla u^k, \\ &u^k \geqq 0, \\ &(u^k)^T (A^k u^k - b^k + P^k \nabla u^k) = 0. \end{aligned} \tag{2.18}$$

Hence,

$$\|U^k - u^k\|_2 \leqq \alpha_k^{-1} \|P^k\|_2 \|\nabla u^k\|_2. \tag{2.19}$$

Proof. Consider the projected Gauss–Seidel method defined by (2.6) and (2.7). For each point x_j^k we first compute the dynamic residual $\tilde{r}_j^{k,s}$. The new value of $u_j^{k,s}$ is chosen so as to reduce the residual. Denote the residual at the point x_j^k immediately after step (2.7) by $\hat{r}_j^{k,s}$, so that

$$\hat{r}_j^{k,s} = \bar{r}_j^{k,s} - a_{jj}^k (u_j^{k,s} - u_j^{k,s-1}). \tag{2.20}$$

Remembering that A^k is negative definite, and hence $a_{jj}^k < 0$, we see that there are two possibilities:

$$\begin{aligned} \text{either} \quad & u_j^{k,s} > 0 \text{ and } \hat{r}_j^{k,s} = 0, \\ \text{or} \quad & u_j^{k,s} = 0 \text{ and } \hat{r}_j^{k,s} \geqq 0. \end{aligned}$$

Thus, dropping the superscript s, and setting $\hat{r}^k = \{\hat{r}_j^k : 1 \leqq j \leqq N_k\}$,

$$u^k \geqq 0, \quad \hat{r}^k \geqq 0, \quad (u^k)^T \hat{r}^k = 0. \tag{2.21}$$

Let

$$r^k = b^k - A^k u^k.$$

It is readily seen from (2.17) that

$$r^k = \hat{r}^k + P^k(u^{k,s} - u^{k,s-1}) = \hat{r}^k + P^k \nabla u^k. \tag{2.22}$$

Combining (2.21) and (2.22) we obtain (2.18). Comparing (2.16) and (2.18) we see that the arguments which led to (2.15) from (2.16) may be applied to (2.18), with r_+^k replaced by $P^k \nabla u^k$, to obtain (2.19). □

As Lemmas 2.1 and 2.2 show, we can estimate the error in an approximate solution u^k in terms of the residual r^k or the difference ∇u^k; we will usually use ∇u^k to estimate the error, since this quantity is readily available during a G^k-projected sweep.

Remark 2.1. The reader may wonder why we bothered to introduce r_+^k in Lemma 2.1, since (2.15) holds with r_+^k replaced by r^k. The reason is that for the LCP (2.3) there may be large positive residuals at points x_j^k where $U^k(x_j^k) = 0$, but this does not mean that the error is large.

In multigrid algorithms it is necessary to compare norms on different grids. We, therefore, wish to introduce a norm which is not grid dependent. To do so, we proceed as follows.

We first note that, to a good approximation, the coercivity constant α_k for $-A^k$ satisfies

$$\alpha_k \doteq \alpha h^2,$$

where α is the smallest eigenvalue of $\mathscr{L}$.

Next, assume that the approximate grid function u^k has been extended to a function $u^k(x)$ on Ω approximating the solution $u(x)$ of (1.1). Then

$$\begin{aligned} \|u(x) - u^k(x)\|_{2,\Omega} = \left| \int_\Omega |u(x) - u^k(x)|^2 \, dx \right|^{1/2} &\doteq \left| \sum_{j=1}^{N_k} h_k^n |U_j^k - u_j^k|^2 \right|^{1/2} \\ &= h_k^{n/2} \|U^k - u^k\|_2 \\ &\leqq \frac{h_k^{n/2}}{\alpha_k} \|P^k\|_2 \|\nabla u^k\|_2 \\ &\doteq \frac{\|P^k\|_2}{\alpha} h_k^{n/2-2} \|\nabla u^k\|_2. \end{aligned}$$

The norms $\|P^k\|_2$ are essentially independent of k; for example, for the five-point formula, $\|P^k\|_2 \leqq 2$. Thus a measure for the error $\|u(x) - u^k(x)\|_{2,\Omega}$ is provided by

$$\|\nabla u^k\|_G \equiv h_k^{n/2-2} \|\nabla u^k\|_2, \tag{2.23}$$

and this norm will be used in the computations.

2.3. PFAS (projected full approximation scheme). PFAS (projected full approximation scheme) obtains an approximation $\bar{u}^M$ to the solution U^M on the finest grid G^M by recursively generating a sequence of approximations $\bar{u}^k$ on the grids G^k.

Each $\bar{u}^k$ is an approximate solution to an LCP of the form (2.2) with F^k replaced by a function $\bar{F}^k$ which is defined later. In general, $\bar{F}^k$ is different from F^k so that $\bar{u}^k$ is not an approximation to U^k. However, $\bar{F}^M = F^M$ and so $\bar{u}^M$ is an approximation to U^M.

We begin by initializing $\bar{u}^M$ to some suitable value. For example, we might set

$$\bar{u}^M(x) = \begin{cases} g(x) & \text{on } \partial G^M, \\ 0 & \text{in } G^M. \end{cases} \tag{2.24}$$

We also set

$$\|\nabla \bar{u}^M\|_G = 10^{30}, \qquad \varepsilon^M = \varepsilon \tag{2.25}$$

(where ε is the desired accuracy on the finest grid, and where the astronomical number 10^{30} ensures that at least two G^M projected sweeps are carried out),

$$\begin{aligned} \bar{F}^M(x) &= F^M(x) \quad \text{for } x \in G^M, \\ \bar{U}^M(x) &= U^M(x) \quad \text{for } x \in G^M. \end{aligned} \tag{2.26}$$

We now make a number of G^M projected sweeps,

$$\bar{u}^M \leftarrow \text{projected Gauss–Seidel } [\bar{u}^M : L^M, \bar{F}^M]. \tag{2.27}$$

After each sweep we test whether

$$\|\nabla \bar{u}^M\|_G \leqq \varepsilon^M. \tag{2.28}$$

If so, the accuracy criterion is satisfied, and we accept $\bar{u}^M$ as an accurate approximation to $U^M \equiv \bar{U}^M$ on G^M.

It is known that Gauss–Seidel iteration is a smoothing process: the error $\bar{U}^M(x) - \bar{u}^M(x)$ becomes smoother as the number of sweeps increases, while, at the same time, the rate of convergence slows down. We, therefore, carry out only a few G^M projected sweeps, stopping when either (2.28) is satisfied or convergence is slow:

$$\|\nabla \bar{u}^M\|_G \geqq \eta \|\nabla \bar{u}^M_{\text{old}}\|_G. \tag{2.29}$$

Here, η is a fixed parameter; in our work we have taken $\eta = .5$.

Suppose that (2.28) is not satisfied but that (2.29) is satisfied. This means on the one hand that the accuracy of $\bar{u}^M$ must be improved, and on the other hand that it is inefficient to continue iterating on G^M. The slow rate of convergence on G^M indicates that the error is smooth, so that the error can be represented satisfactorily on the next coarsest grid, G^{M-1}. We therefore move to G^{M-1}.

Since $\bar{U}^M(x)$ satisfies (2.2), with $k = M$ and $F^M = \bar{F}^M$, the error

$$V^M(x) = \bar{U}^M(x) - \bar{u}^M(x) \tag{2.30}$$

satisfies the LCP

$$\begin{aligned} & L^M V^M(x) \leqq \bar{r}^M(x) && \text{on } G^M, \\ & V^M(x) + \bar{u}^M(x) \geqq 0 && \text{on } G^M, \\ & [V^M(x) + \bar{u}^M(x)][L^M V^M(x) - \bar{r}^M(x)] = 0 && \text{on } G^M, \\ & V^M(x) = 0 && \text{on } \partial G^M, \end{aligned} \tag{2.31}$$

where the residual $\bar{r}^M$ is given by

$$\bar{r}^M(x) = \bar{F}^M(x) - L^M \bar{u}^M(x), \qquad x \in G^M. \tag{2.32}$$

As already observed, $V^M(x)$ is a smooth function and may, therefore, be accurately represented on G^{M-1}. Furthermore, comparing (2.31) and (1.1) we see that $V^M(x)$

is an approximation to the continuous solution $v(x)$ of the LCP

$$
\begin{aligned}
&\mathcal{L}v(x) \leqq \bar{r}^M(x), && x \in \Omega,\\
&v(x) + \bar{u}^M(x) \geqq 0, && x \in \Omega,\\
&[v(x) + \bar{u}^M(x)][\mathcal{L}v(x) - \bar{r}^M(x)] = 0, && x \in \Omega,\\
&v(x) = 0 && \text{on } \partial\Omega
\end{aligned} \tag{2.33}
$$

(where, by abuse of notation, $\bar{r}^M(x)$ and $\bar{u}^M(x)$ are defined on Ω by appropriate interpolation between the values of $\bar{r}^M$ and $\bar{u}^M$ on the gridpoints of G^M). Thus, a good approximation to $V^M(x)$ may be obtained by solving the finite difference approximation to (2.33) on G^{M-1}. That is, $V^M(x)$ is closely approximated on G^{M-1} by the solution $W^{M-1}(x)$ of the LCP,

$$
\begin{aligned}
&\text{(a)} && L^{M-1}W^{M-1}(x) \leqq S_M^{M-1}\bar{r}^M(x), && \text{on } G^{M-1},\\
&\text{(b)} && W^{M-1}(x) + I_M^{M-1}\bar{u}^M(x) \geqq 0, && \text{on } G^{M-1},\\
&\text{(c)} && [W^{M-1}(x) + I_M^{M-1}\bar{u}^M(x)][L^{M-1}W^{M-1}(x) - S_M^{M-1}\bar{r}^M(x)] = 0, && \text{on } G^{M-1},\\
&\text{(d)} && W^{M-1}(x) = 0, && \text{on } \partial G^{M-1}.
\end{aligned} \tag{2.34}
$$

Here I_M^{M-1} and S_M^{M-1} are operators taking grid functions on G^M into grid functions on G^{M-1}. (As an aid in memorization, note that in $I_M^{M-1}\bar{u}^M$ the subscript M and superscript M "cancel".)

The operators I_M^{M-1} and S_M^{M-1} can be defined in many ways. One way is to choose both I_M^{M-1} and S_{M-1}^M to be the injection operator:

$$
\text{Inj}_M^{M-1} w(x) = w(x), \qquad x \in G^{M-1}. \tag{2.35}
$$

If we were solving a linear boundary value problem, then condition (2.34b) would not apply, and it would be most efficient to solve for the correction W^{M-1} on G^{M-1}. Since we are solving inequalities the problem is nonlinear, and it is necessary to solve for a "full approximation" $\bar{U}^{M-1}$ on G^{M-1}.

Setting

$$
\bar{U}^{M-1}(x) = W^{M-1}(x) + I_M^{M-1}\bar{u}^M(x), \tag{2.36}
$$

it follows that $\bar{U}^{M-1}(x)$ satisfies the LCP

$$
\begin{aligned}
&\text{(a)} && L^{M-1}\bar{U}^{M-1}(x) \leqq \bar{F}^{M-1}(x) && \text{in } G^{M-1},\\
&\text{(b)} && \bar{U}^{M-1}(x) \geqq 0 && \text{in } G^{M-1},\\
&\text{(c)} && \bar{U}^{M-1}(x)[L^{M-1}\bar{U}^{M-1}(x) - \bar{F}^{M-1}(x)] = 0 && \text{in } G^{M-1},\\
&\text{(d)} && \bar{U}^{M-1}(x) = g(x) && \text{on } \partial G^{M-1},
\end{aligned} \tag{2.37}
$$

where

$$
\begin{aligned}
\bar{F}^{M-1}(x) &= S_M^{M-1}\bar{r}^M(x) + L^{M-1}I_M^{M-1}\bar{u}^M(x)\\
&= S_M^{M-1}[\bar{F}^M(x) - L^M\bar{u}^M(x)] + L^{M-1}I_M^{M-1}\bar{u}^M(x).
\end{aligned} \tag{2.38}
$$

Finally, we set

$$
\varepsilon^{M-1} = \delta\|\nabla\bar{u}^M\|_G, \tag{2.39}
$$

and

$$\bar{u}^{M-1} = I_M^{M-1}\bar{u}^M, \tag{2.40}$$

where δ is a constant; in our computations δ has been set equal to .15.

To recapitulate, starting with initial values of $\bar{u}^M$, ε^M, and $\bar{F}^M$, we first carry out G^M projected sweeps until convergence slows down. We then introduce a subsidiary problem on G^{M-1} with known $\bar{F}^{M-1}$ and ε^{M-1} and initial approximation $\bar{u}^{M-1}$. The process can be repeated, so that at any one stage of the computation we have a sequence of grid approximations $\bar{u}^M, \bar{u}^{M-1}, \cdots, \bar{u}^{k-1}$ (approximating $\bar{U}^M, \bar{U}^{M-1}, \cdots, \bar{U}^{k-1}$, respectively), tolerances $\varepsilon^M, \varepsilon^{M-1}, \cdots, \varepsilon^{k-1}$, and right-hand sides $\bar{F}^M, \bar{F}^{M-1}, \cdots, \bar{F}^{k-1}$.

In the general case, $\bar{U}^k$ is the solution of the LCP

$$\begin{aligned}
&\text{(a)} && L^k\bar{U}^k(x) \geqq \bar{F}^k(x) && \text{in } G^k,\\
&\text{(b)} && \bar{U}^k(x) \geqq 0 && \text{in } G^k,\\
&\text{(c)} && \bar{U}^k(x)(L^k\bar{U}^k(x) - \bar{F}^k(x)) = 0 && \text{in } G^k,\\
&\text{(d)} && \bar{U}^k(x) = g(x) && \text{on } \partial G^k,
\end{aligned} \tag{2.41}$$

or equivalently,

$$\begin{aligned}
&\text{(a)} && A^k\bar{U}^k \geqq \bar{b}^k,\\
&\text{(b)} && \bar{U}^k \geqq 0,\\
&\text{(c)} && (\bar{U}^k)^T(A^k\bar{U}^k - b^k) = 0.
\end{aligned} \tag{2.42}$$

This LCP is solved approximately using G^k projected sweeps until the latest approximation $\bar{u}^k$ satisfies either

$$\|\nabla\bar{u}^k\|_G \leqq \varepsilon^k \tag{2.43}$$

or

$$\|\nabla\bar{u}^k\|_G \geqq \eta\|\nabla\bar{u}^k_{\text{old}}\|_G. \tag{2.44}$$

If (2.44) holds but (2.43) does not, then a new problem on G^{k-1} is defined by setting

$$\bar{F}^{k-1} = S_k^{k-1}[\bar{F}^k - L^k\bar{u}^k] + L^{k-1}I_k^{k-1}\bar{u}^k, \tag{2.45}$$

$$\varepsilon^{k-1} = \delta\|\nabla\bar{u}^k\|_G, \tag{2.46}$$

$$\bar{u}^{k-1} = I_k^{k-1}\bar{u}^k, \tag{2.47}$$

$$\bar{U}^{k-1} = W^{k-1} + I_k^{k-1}\bar{u}^k, \tag{2.48}$$

$$V^k = \bar{U}^k - \bar{u}^k, \tag{2.49}$$

where W^{k-1} is an approximation to V^k on G^{k-1}. Unless otherwise indicated, I_k^{k-1} and S_k^{k-1} will be taken to be the injection operator Inj_k^{k-1}.

At some stage the latest approximation $\bar{u}^{k-1}$ must satisfy (2.43):

$$\|\nabla\bar{u}^{k-1}\|_G \leqq \varepsilon^{k-1}, \tag{2.50}$$

if for no other reason than that when $k-1=1$ we cannot introduce any more subsidiary problems and must iterate until (2.50) is satisfied. Having found an approximation $\bar{u}^{k-1}$ of sufficient accuracy, we return to G^k. To do so, we first determine an

approximation w^{k-1} to W^{k-1} from (2.48), namely,

$$w^{k-1} = \bar{u}^{k-1} - I_k^{k-1}\bar{u}^k. \tag{2.51}$$

Next, let I_{k-1}^k be an interpolation operator taking grid functions on G^{k-1} into grid functions on G^k. One choice for I_{k-1}^k is the bilinear interpolation operator L_{k-1}^k defined as follows. If P_1, P_2, P_3 and P_4 are the corners of a square in G^{k-1} (see Fig. 2.1), then

$$L_{k-1}^k w^{k-1}(P_i) = \begin{cases} w^{k-1}(P_i), & 1 \leqq i \leqq 4, \\ (w^{k-1}(P_1) + w^{k-1}(P_2))/2, & i = 5, \\ (w^{k-1}(P_1) + w^{k-1}(P_4))/2, & i = 6, \\ \left(\sum_{i=1}^{4} w^{k-1}(P_i)\right)\Big/ 4, & i = 7. \end{cases} \tag{2.52}$$

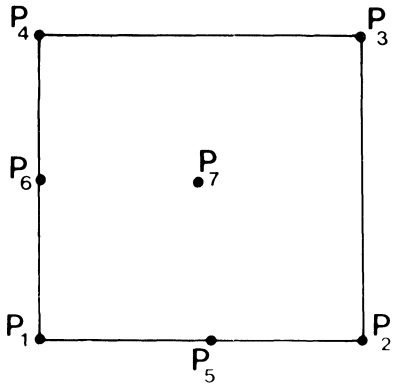

FIG. 2.1. *Bilinear interpolation from* G^{k-1} *to* G^k.

Since W^{k-1} is an approximation to V^k on G^{k-1},

$$I_{k-1}^k w^{k-1} = I_{k-1}^k [\bar{u}^{k-1} - I_k^{k-1}\bar{u}^k] \tag{2.53}$$

is an approximation to V^k, and, noting (2.49),

$$\tilde{u}^k = \bar{u}^k + I_{k-1}^k w^{k-1} \tag{2.54}$$

is an improved approximation to $\bar{U}^k$. However, because of the nonnegativity constraint upon $\bar{U}^k$, we allow somewhat greater generality, and replace $\bar{u}^k$ as follows:

$$\bar{u}^k \leftarrow \varphi(\tilde{u}^k; \bar{u}^k) = \varphi(\bar{u}^k + I_{k-1}^k w^{k-1}; \bar{u}^k). \tag{2.55}$$

Initially we set

$$\varphi(\tilde{u}^k; \bar{u}^k) = \tilde{u}^k, \tag{2.56}$$

but other choices will be considered later.

PFAS is described by (2.24) through (2.56). A flowchart is given in Fig. 3.1, and the implementation is discussed in § 3. If the algorithm converges, we will eventually obtain an approximation $\bar{u}^M$ satisfying the required accuracy condition (2.28), and the algorithm will terminate.

3. Implementation of PFAS. The flowchart for PFAS is given in Fig. 3.1. PFAS has been implemented as a FORTRAN subroutine for the case when Ω is a rectangle in R^2, $\mathscr{L}$ is the Laplacian operator, L is the five-point difference operator, I_k^{k-1} and S_k^{k-1} are injections (see (2.35)), and I_{k-1}^k is bilinear interpolation (see (2.52)). The subroutine PFAS, which is listed in Brandt and Cryer [1980] as part of the program for solving the porous flow free boundary problem described in § 4, is a modification of an earlier program, FAS Cycle C, of Brandt.

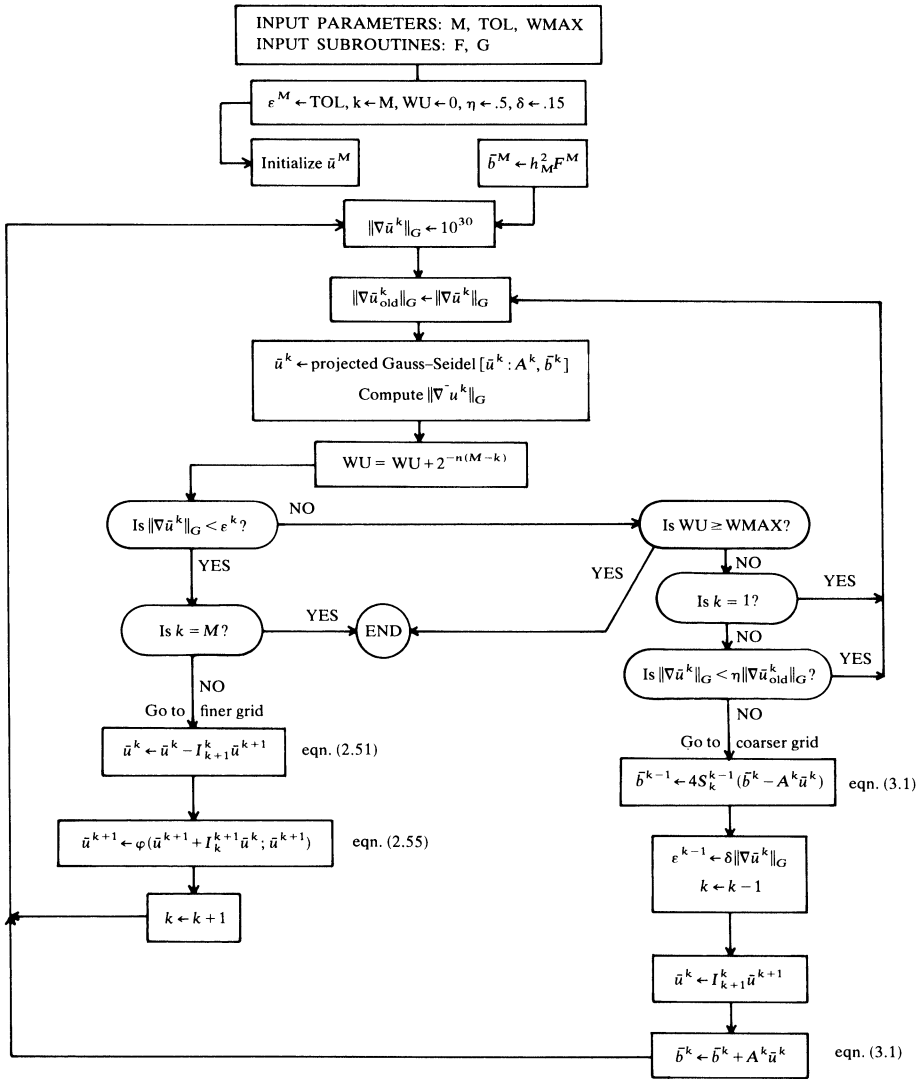

FIG. 3.1. *Flow chart for* PFAS.

PFAS is very easy to implement: the subroutine, with profuse comment cards, requires only 280 FORTRAN statements. It may also be remarked that many other interesting free boundary problems (for example, elastic-plastic torsion problems and cavitating journal bearing problems) are formulated on simple polygonal regions, and the program could easily be modified to handle these problems.

The following comments arise.

1. In PFAS, the LCP for $\bar{U}^k$ is solved in the form (2.42) rather than (2.41), but the values of $\bar{u}^k$ on ∂G^k are also stored. Thus, $\bar{b}^k = h_k^2 \bar{F}^k$ is stored instead of $\bar{F}^k$. In going from G^k to G^{k-1} we have, from (2.45), since $h_{k-1} = 2h_k$,

$$
\begin{aligned}
\bar{b}^{k-1} &= h_{k-1}^2 \bar{F}^{k-1} \\
&= h_{k-1}^2 (S_k^{k-1}[\bar{F}^k - L^k \bar{u}^k] + L^{k-1} I_k^{k-1} \bar{u}^k) \\
&= h_{k-1}^2 (S_k^{k-1} h_k^{-2}[\bar{b}^k - A^k \bar{u}^k] + L^{k-1} I_k^{k-1} \bar{u}^k) \\
&= 4 S_k^{k-1}[\bar{b}^k - A^k \bar{u}^k] + A^{k-1} I_k^{k-1} \bar{u}^k.
\end{aligned} \tag{3.1}
$$

2. A G^k-*work-unit* is the work required for one G^k-projected sweep. The work for one G^k-projected sweep is approximately $2^{-n(M-k)} G^M$-work-units, and WU denotes the total number of G^M-work-units. When no confusion is possible we write "work unit" instead of "G^M-work-unit".

3. The asymptotic speed of convergence is measured by the *asymptotic convergence factor* $\mathring{\mu}$, which is defined by

$$
\mathring{\mu} = \lim_{WU \to \infty} [\|\nabla \bar{u}^M\|_G]^{1/WU}. \tag{3.2}
$$

4. All the numerical computations were performed on the Univac 1180 at the University of Wisconsin–Madison. The programs were written in ASCII FORTRAN and compiled and executed using full optimization.

The Univac 1180 single-precision arithmetic has approximately eight decimals. The residuals usually decrease quite rapidly at the beginning of a computation so the round-off threshold is quickly reached. For example, for the problem considered in § 4 with $M = 5$, $\|U^M\|_G$ is about 2×10^3, and the single precision algorithm went into a loop when $\|\nabla \bar{u}^M\|_G$ reached 5×10^{-6} after a mere 50 work units.

In the numerical experiments we were particularly interested in measuring the asymptotic convergence factor $\mathring{\mu}$. To eliminate round-off effects, all the computations reported on here used double precision arithmetic. Of course, this is not normally necessary. Furthermore, even if very accurate solutions of the discrete problem (2.2) were required, it would suffice to store $\bar{u}^M$ in double precision and all other quantities in single precision.

The execution times quoted are those provided by the Univac 1180 Exec. System. As is often the case on timesharing systems, the times are only reproducible to within about 10%.

Because of its word length, the UNIVAC 1180 can only directly access 64K words of storage. When $M \geqq 7$, more than 64K words of storage are needed by PFAS, and there is a significant degradation in performance.

5. To measure $\mathring{\mu}$, the iterations were continued for the first 100 work units, unless the residuals vanished before. In practice, one usually iterates only for about 30 work units.

We also used several values of M in order to measure the dependence of $\mathring{\mu}$ upon M.

The computations starting at a level-M–level-$(M-1)$ junction and continuing until the next level-M–level-$(M-1)$ junction are called a *cycle*.

While minor variations do arise, a cycle often consists of a sequence of 2 sweeps at each of levels $M-1, M-2, \cdots, 1$, followed by 2 sweeps at each of levels $2, \cdots, M-1$, terminating with 2 or 3 sweeps at level M. If this pattern is followed

with 3 sweeps at level M, then the average number of work units per cycle is

$$3+4[2^{-n}+2^{-2n}+\cdots]=3+4/(2^n-1), \tag{3.3}$$

and the average number of work units per G^M projected sweep is $1+4/(3(2^n-1))$.

Of course, very irregular patterns are observed when the round-off threshold is reached.

6. It is usually found that $\|\nabla \bar{u}^M\|_G$ decreases steadily but not very regularly, in part because of slight variations in the number of sweeps at each level. To evaluate the algorithm, we have used two quantities:

$$r_f = \|\nabla \bar{u}^M_{\text{final}}\|_G = \text{ the value of } \|\nabla \bar{u}^M\|_G \text{ at the end of the last complete cycle before 100 work units,} \tag{3.4}$$

$$\mathring{\mu}_f = [\|\nabla \bar{u}^M_{\text{final}}\|_G / \|\nabla \bar{u}^M_{\text{initial}}\|_G]^{1/[\text{WU}_{\text{final}}-\text{WU}_{\text{initial}}]}. \tag{3.5}$$

where $\|\nabla \bar{u}^M_{\text{initial}}\|_G$ is the value of $\|\nabla \bar{u}^M\|_G$ after the first G^M sweep; $\mathring{\mu}_f$ is an estimate for the asymptotic convergence factor $\mathring{\mu}$.

We usually only quote r_f to one decimal place and $\mathring{\mu}_f$ to two decimal places, since this is quite adequate for our purposes.

7. In all the experiments reported here, the parameters δ and η (see (2.29) and (2.39)) were given by $\delta = .5$ and $\eta = .15$. According to Brandt [1977], the rate of convergence is not very sensitive to changes in these parameters, and this was confirmed in a few experiments.

In a few cases, but never for $\delta = .5$ and $\eta = .15$, the program "hunted": that is, the program went down from G^M to G^1, up to G^k for $k < M$, and then down again to G^1 instead of continuing up to G^M. This might happen several times before G^M was reached again.

4. Numerical results for porous flow through a dam. Calculations were performed on the well-known free boundary problem describing the flow of water through a porous dam. The geometry is shown in Fig. 4.1. Water seeps from a reservoir of height y_1 through a rectangular dam of width a to a reservoir of height y_2. Part of the dam is saturated and the remainder of the dam is dry. The wet and dry regions are separated by an unknown free boundary which must be found as part of the solution. For an introduction to the problem see Bear [1972] or Cryer [1976].

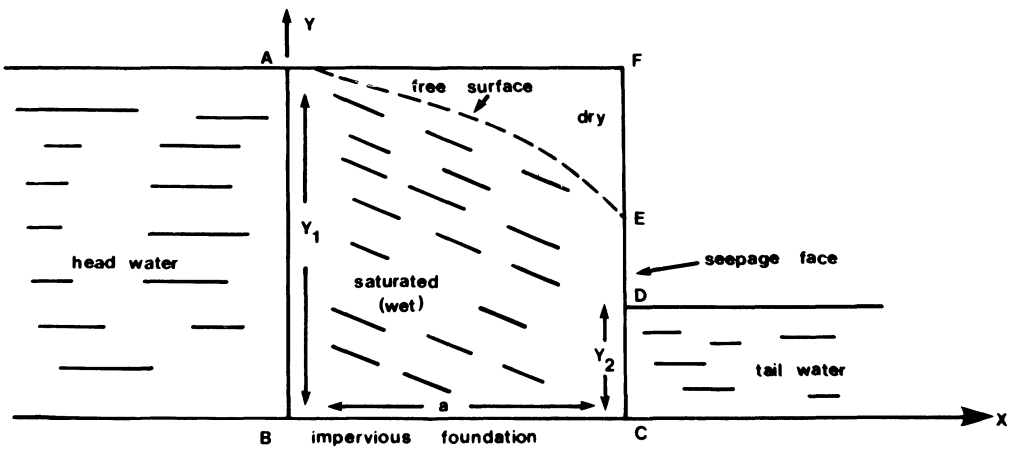

FIG. 4.1. *Seepage through a simple rectangular dam.*

As shown by Baiocchi [1971], the problem can be formulated as follows: Find u on the rectangle $\Omega = ABCF$ such that

$$\begin{aligned} &u_{xx}+u_{yy} \leqq 1 && \text{in } \Omega, \\ &u \geqq 0 && \text{in } \Omega, \\ &u(u_{xx}+u_{yy}-1)=0 && \text{in } \Omega, \end{aligned} \tag{4.1}$$

$$u = g = \begin{cases} (y_1-y)^2/2 & \text{on } AB, \\ (y_2-y)^2/2 & \text{on } CD, \\ [y_1^2(a-x)+y_2^2(x)]/2a & \text{on } BC, \\ 0 & \text{on } DFA, \end{cases} \tag{4.2}$$

which is in the form (1.1).

This problem was solved using PFAS. The initial values of $\bar{u}^M$ were obtained by interpolating the boundary values of u linearly in the x direction. A listing of the program is given in Brandt and Cryer [1980].

We considered the well-known case, $y_1 = 24$, $y_2 = 4$ and $a = 16$. In all computations G^1 was a $(2+1)\times(3+1)$ grid with $h_1 = 8$. The finest grid used was G^7 with $(128+1)\times(192+1) = 24897$ grid points.

To give the reader an idea of the solution, the solution U^2 of (2.2) is given to four decimal places in Table 4.1.

TABLE 4.1

U^2 for the dam problem.

y \ x	0	4	8	12	16
24	0	0	0	0	0
20	8	2.5371	0	0	0
16	32	18.1486	6.7841	0	0
12	72	47.2732	24.9879	7.9120	0
8	128	89.9564	53.9823	22.6601	0
4	200	146.5702	94.3247	44.7462	0
0	288	218.0000	148.0000	78.0000	8

TABLE 4.2

Solution of the dam problem using PFAS.

M	2	3	4	5	6	7
G^M	5×7	9×13	17×25	33×49	65×97	129×193
r_f	0*	4(−17)*	1(−13)	1(−8)	1(−10)	1(−7)
$\mathring{\mu}_f$	.404	.607	.726	.813	.778	.81
Execution time for 100 work units (seconds)	.114	.428	1.04	3.55	13.39	†
ρ_{SORopt}	.18	.49	.71	.84	.92	.96

* Reached round-off level before 100 work units.

† Required 70K workspace so extended storage facility invoked, and timing not compatible.

The numerical results, for different values of M and $\varepsilon^M = \text{TOL} = 0$, are given in Table 4.2. The most important conclusions are that convergence always occurs and that the convergence factor $\mathring{\mu}_f$ is always less than .81.

We now compare the convergence factors $\mathring{\mu}_f$ in Table 4.2 with those for single-grid methods of solving the LCP (2.2).

A popular single-grid method of solving the LCP (1.3) is G^M-projected SOR (point SOR with projection) which has also been called "modified SOR" by Cottle.

When using G^M-projected SOR, it is observed experimentally that the values of $\bar{u}^M$ settle down quite quickly into positive values and zero values. Thereafter G^M-projected SOR is equivalent to using point SOR on the subset $G_+^M = \{x \in G^M : U^M(x) > 0\}$. Thus the asymptotic convergence factor for G^M projected SOR is in general equal to the asymptotic convergence factor for point SOR on G_+^M. It is known (Varga [1962, p. 294]) that for a region of area A and for the finite difference equations corresponding to the five-point difference approximation to Laplace's equation with stepsize h, the convergence factor for the optimum choice of overrelaxation parameter ω is approximated quite well by

$$\rho_1(h) = \frac{2}{1+3.015[h^2/A]^{1/2}} - 1. \tag{4.3}$$

In the present case we do not know the area of G_+^M, but, as a rough guide, the area of G_+^M is approximately equal to the area of Ω, which is about 80% of the area of the rectangle $ABCF$. Therefore, for our present purposes the asymptotic convergence factor for G^M-projected SOR with optimum choice of ω may be taken to be

$$\rho_{\text{SORopt}} \doteq \frac{2}{1+3.015[h^2/(.8\times 16\times 24)]^{1/2}} - 1 \doteq \frac{2}{1+.172h} - 1; \tag{4.4}$$

these values are given at the bottom of Table 4.2.

As Table 4.2 shows, for large problems, PFAS is faster than G^M-projected SOR. On G^7, for example, the increase in speed (measured in work units) is $\ln .81/\ln .96 \doteq 5.2$. Against this, two factors must be borne in mind: (1) PFAS is more complicated and requires more overhead per work unit; (2) PFAS requires somewhat more storage. We discuss these two factors below, but before doing so we wish to emphasize that although these factors reduce the advantage in speed of PFAS, the measured execution times for PFAS are much smaller than those for G^M-projected SOR.

1. *Overhead.* To obtain an indication of the additional overhead required by PFAS, we compared execution times for $M = 5$. We first used PFAS with $\varepsilon^M = 2 \cdot 10^{-8}$. This required 96.156 work units and took 3.40 seconds. We then modified PFAS so that only the grid $k = M$ was used and so that overrelaxation was used with the overrelaxation parameter ω given by equation (4.4). We were thus using G^M-projected SOR with a nearly optimum ω. To reduce $\|\nabla \bar{u}^M\|_G$ to $\varepsilon^M = 2 \cdot 10^{-8}$ required 146 work units and took 4.82 seconds. Since

$$(3.40/96.156)/(4.82/146) \doteq 1.07,$$

we conclude that, in this application, the additional overhead required by PFAS only increases the computation time per G^M-work-unit by about 10%.

2. *Storage.* As implemented here, PFAS keeps the solutions and residuals on all the grids, and therefore requires storage for $2[1+4^{-1}+4^{-2}+\cdots] = 8/3G^M$ grids. In contrast, G^M-projected SOR requires storage for only one G^M grid.

If storage is at a premium, the residuals on G^M need not be stored, and PFAS requires only 5/3 times as much storage as G^M-projected SOR. If $\bar{u}^M$ is stored to double precision, but $\bar{u}^k$ and $\bar{b}^k$ are stored to single precision for $k < M$, only 4/3 times as much storage is needed. If $F(x)$ were not the constant 1, but a complicated function, then either the function values or the residuals would have to be stored for G^M-projected SOR, and PFAS would require at most 33% more storage.

Another possible single-grid algorithm for solving the LCP (1.3) is the MBSOR (Modified Block SOR) algorithm of Cottle, Golub and Sacher [1978]. This algorithm is based upon the solution of a sequence of "one-dimensional" LCP's in much the same way as line SOR is based upon solving a sequence of "one-dimensional" equations. We used MBSOR to solve the dam problem (4.1), (4.2) for the case $M = 5$. The program was kindly provided by Professor Sacher. We tried a few values of the overrelaxation parameter ω, and found that 1.8 gave the best results. With $\omega = 1.8$, MBSOR required 114 iterations to reduce $\|\nabla u^M\|_G$ to below $2 \cdot 10^{-8}$ and took 13.13 seconds. The following comments arise.

(i) In numerical experiments on the dam problem, Cottle [1974] found that MBSOR was about 20% faster than "modified point SOR", that is, G^M-projected SOR. This is consistent with the fact that, for equations, the convergence ratio for line SOR is only faster by a factor of $\sqrt{2}$ than point SOR, while there is more computation per iteration. This is also consistent with the present results, since G^M-projected SOR required 146 iterations to reduce the residual to $2 \cdot 10^{-8}$ while MBSOR required only 114.

(ii) The poor execution time of MBSOR (13.13 seconds) compared to PFAS (3.40 seconds) can be explained in part by two factors: (a) MBSOR requires more computation per iteration than is needed by PFAS for a single work unit; (b) the MBSOR program was written for the case of general coefficients, while the PFAS program takes advantage of the properties of the five-point difference operator.

(iii) It must also be borne in mind that Cottle, Golub and Sacher [1978] found that MBSOR was three times as fast as G^M-projected SOR for the journal bearing problem where the solution is zero at a high percentage of the gridpoints.

We conclude from Table 4.2 and from the above discussion, that for the dam problem (4.1), (4.2), PFAS is faster than G^M-projected SOR and modified block SOR for $M \geqq 5$, that is, for grids of dimension 33×49 or greater. Furthermore, we also conclude that the values of $\mathring{\mu}_f$ and ρ_{SORopt} in Table 4.2 provide a reasonably accurate guide to the relative performance of PFAS and G^M-projected SOR. We believe that PFAS will be faster than both G^M-projected SOR and MBSOR for a wide range of problems. Indeed, as shown by Table 4.2, the asymptotic convergence factor $\mathring{\mu}$ for PFAS is approximately equal to .8 for all values of M. Consequently, the amount of work required to reduce the residuals on G^M to below a given threshold ε is $O(N)$, where N is the number of gridpoints in G^M. In contrast, both G^M-projected SOR and modified block SOR have computation times which are $O(N^{3/2})$.

5. Alternative implementations of PFAS. In this section we discuss alternative implementations of PFAS, the best of which achieves substantially improved performance.

The improvement in PFAS which might be possible is suggested by considering the asymptotic convergence ratio, $\mathring{\mu}_{\text{FAS}}$ say, for FAS for Poisson's equation. For FAS, the error reduction per G^M-sweep is .5. If each G^M-sweep is accompanied by, on average, one G^k-sweep for $1 \leqq k \leqq M-1$, then the number of work units per

G^M-sweep is

$$1+2^{-2}+2^{-4}+\cdots=4/3,$$

and the convergence ratio is $(.5)^{3/4}=.595$, as stated by Brandt [1977, p. 351]. In the present case, as observed in § 3, the average number of work units per G^M-sweep is

$$1+4/[3(2^n-1)]=13/9,$$

so that

$$\mathring{\mu}_{\mathrm{FAS}}=(.5)^{9/13}=.6188. \tag{5.1}$$

This value of $\mathring{\mu}_{\mathrm{FAS}}$ is observed experimentally. The worst observed value of $\mathring{\mu}_f$ for the PFAS results quoted in § 3 was $\mathring{\mu}_f=.81$. Thus, FAS (for equations) is faster than PFAS (for LCP's) by a factor of $\ln .6188/\ln .81=2.28$.

Plausible reasons why PFAS is slower than FAS include the following difficulties:

D1: *Negative components of* $\bar{u}^k$. The inequality (2.41b) requires that $\bar{U}^k$ be nonnegative. In each G^k-projected sweep the step (2.7) ensures that $\bar{u}^k$ is nonnegative. Furthermore, if I_k^{k-1} is the injection operator, the initial approximation $\bar{u}^{k-1}$ defined by (2.47) is also nonnegative. However, (2.54) does not preserve nonnegativity: in returning to G^k from G^{k-1}, the initial approximation $\bar{u}^k$ may have negative components, and this is often observed. Of course, any negative components are removed in the first subsequent G^k-projected sweep, but nevertheless the introduction of negative components must retard convergence.

D2: *Large residuals near the free boundary.* At a point $x\in G^k$ where $\bar{U}^k(x)=0$ the corresponding residual

$$\bar{R}^k(x)=\bar{F}^k(x)-L^k\bar{U}^k(x) \tag{5.2}$$

must be nonnegative because of the inequality (2.41a) but need not be small.

D3: *Influence of the discrete interface.* The *discrete interface* $\Gamma^k\subset R^2$ is the interface between the set of points where $\bar{U}^k>0$ and the set of points where $\bar{U}^k=0$. Γ^k approximates the *continuous interface*, or *free boundary*, Γ separating the points where the solution $u(x)$ is positive from the points where $u(x)$ is zero.

In special cases it may happen that $\Gamma^k=\Gamma$ for all k, in which case PFAS converges as fast as FAS. An example is given by problem (5.3), (5.4) below with $R=2$, for which Γ is the line $y=5-2x$; it is found experimentally that $\Gamma^k=\Gamma$ for $k\leqq 6$.

In general, Γ^k and Γ differ by $O(h_k)$, and Γ^k and Γ^{k-1} differ by $O(h_k)$. In particular, it may happen that $\bar{U}^k(x)>0$ while $\bar{U}^{k-1}(x)=0$. Furthermore, near Γ^{k-1} the residuals may be less smooth because of the projection (2.7) and because of the irregular shape of Γ^k and Γ^{k-1}. This introduces errors in the coarse grid corrections (2.55), thereby slowing the rate of convergence. Finally, the injection operator (2.35) is not adequate if the data to which it is applied is not smooth.

To test the influence of the relationship between Γ and Γ^k on the convergence of PFAS, computations were made not only for the dam problem (4.1), (4.2) but also for the LCP:

$$\begin{aligned}
&\text{(a)} && u_{xx}+u_{yy}\leqq f(x,y) && \text{in } \Omega,\\
&\text{(b)} && u\geqq 0 && \text{in } \Omega,\\
&\text{(c)} && u=g && \text{on } \partial\Omega,\\
&\text{(d)} && u(u_{xx}+u_{yy}-f)=0 && \text{in } \Omega,
\end{aligned} \tag{5.3}$$

where $\Omega = [0, 3] \times [0, 2]$, and f and g are chosen so that the exact solution is

$$u = [\cos(x+y)+2][\max\{0; 2.5R - Rx - y\}]^2. \tag{5.4}$$

Here, R is a parameter which is chosen close to the value 2. Note that $u \in C^2(\Omega)$ and $u = 0$ above the line $y = R(2.5 - x)$. By changing the value of R we can force gridpoints to lie very close to the exact free boundary; this may be expected to cause PFAS difficulty, because if $\bar{U}^k(x)$ is positive but very small for some $x \in G^k$ then it will take PFAS a large number of iterations to determine whether $\bar{U}^k(x)$ is zero or positive.

Multigrid algorithms can often be speeded up by modifying the operators I_k^{k-1}, S_k^{k-1} and I_{k-1}^k. We have tried a number of modifications of the corresponding PFAS subroutines which were intended to address the difficulties D1 to D3 mentioned above.

Our first modifications to the auxiliary subroutines of PFAS were not very successful, but they were very instructive and we briefly summarize them. In all cases, the results are for the dam problem with $M = 5$.

M1. PFAS was modified so as to enforce nonnegativity of $\bar{u}^k$ immediately after returning from G^{k-1}. This was done by defining φ in (2.55) by

$$\varphi(\tilde{u}^k; \bar{u}^k) = \max\{0, \tilde{u}^k\}. \tag{5.5}$$

This modification converged slightly faster than PFAS with $\mathring{\mu}_f = .803$.

M2. The usual situation in which the nonnegativity of $\bar{u}^k$ is violated is as follows. Let $\bar{u}^k(x) = 0$, where $x \in G^k$ but $x \notin G^{k-1}$. Let $y \in G^{k-1}$ be a neighbor of x, such that $\bar{u}^k(y) > 0$. It may then happen that $W^{k-1}(y) < 0$. As a result, $(I_{k-1}^k w^{k-1})(x)$ may be negative, and if so the updated value of $\bar{u}^k(x)$ will be negative.

To avoid this, PFAS was modified so that the operator I_k^{k-1} became:

$$I_k^{k-1}\bar{u}^k(y) = \begin{cases} \bar{u}^k(y) & \text{if } \bar{u}^k(x) > 0 \text{ for all eight} \\ & \text{neighbors } x \text{ of } y \text{ in } G^k, \\ 0 & \text{otherwise.} \end{cases} \tag{5.6}$$

Remembering from (2.48) that

$$\bar{U}^{k-1} = W^{k-1} + I_k^{k-1}\bar{u}^k,$$

we see from (5.6) and (2.41b) that the restraint $W^{k-1}(y) \geqq 0$ is enforced for every point $y \in G^{k-1}$ with a neighbor $x \in G^k$ such that $\bar{u}^k(x) = 0$.

This modification converged slightly more slowly than PFAS, with $\mathring{\mu}_f = .817$.

M3. PFAS was modified so that if the current value of $\bar{u}^M(x)$ was zero, then $\bar{u}^k(x)$ was forced to be zero for $k < M$. In effect, (2.7) was followed by a further operation:

$$\text{If } k < M \quad \text{and} \quad \bar{u}^M(x_j^k) = 0, \quad \text{then } \bar{u}_j^{k,s} = 0. \tag{5.7}$$

This modification converged, but much more slowly than PFAS, with $\mathring{\mu}_f = .887$.

M4. Brandt [1977, p. 378] has found residual weighting useful when the coefficients of the differential equation are changing rapidly. We, therefore, changed the algorithm so that S_k^{k-1} became:

$$4S_k^{k-1}r^k(x) = \sum_{\Delta} \rho(\Delta) r^k(x + \Delta h_k), \tag{5.8}$$

where $\Delta = (\Delta_1, \Delta_2)$ for integers Δ_1, Δ_2 and the only nonzero $\rho(\Delta)$ are

$$\begin{aligned} &\rho(0,0) = 1 \\ &\rho(0,1) = \rho(1,0) = \rho(0,-1) = \rho(-1,0) = \tfrac{1}{2}, \\ &\rho(1,1) = \rho(1,-1) = \rho(-1,1) = \rho(-1,-1) = \tfrac{1}{4}. \end{aligned} \tag{5.9}$$

This modification cycled between G^1 and G^2, as did the further modification for which I_k^{k-1} was also defined by (5.8), (5.9).

The nonconvergence of Modification M4 requires explanation, and this is provided by

LEMMA 5.1. *Let φ be defined by* (2.56). *For $1 \leqq k \leqq M$ let $\bar{U}^k$ be the solution of the* LCP (2.41), *where $\bar{F}^k$ satisfies* (2.45). *Finally, let I_{k-1}^k satisfy*

$$(I_{k-1}^k(z^{k-1}) = 0) \Rightarrow (z^{k-1} = 0) \quad \textit{for all } z^{k-1} \in R^{N_{k-1}}. \tag{5.10}$$

Then for PFAS *to converge it is necessary that*

$$S_k^{k-1}[\bar{F}^k - L^k\bar{U}^k] \geqq 0, \tag{5.11}$$

$$I_k^{k-1}\bar{U}^k \geqq 0, \tag{5.12}$$

$$[I_k^{k-1}\bar{U}^k]^T S_k^{k-1}[\bar{F}^k - L^k\bar{U}^k] = 0. \tag{5.13}$$

Proof. We apply PFAS by setting $\bar{u}^k = \bar{U}^k$, and forming the LCP (2.41) on G^{k-1}:

$$\begin{aligned} &L^{k-1}\bar{U}^{k-1} \leqq \bar{F}^{k-1}, \\ &\bar{U}^{k-1} \geqq 0, \\ &(\bar{U}^{k-1})^T(L^{k-1}\bar{U}^{k-1} - \bar{F}^{k-1}) = 0. \end{aligned} \tag{***}$$

Solving this exactly so that $\bar{u}^{k-1} = \bar{U}^{k-1}$, we then return to G^k. Since PFAS converges, the new value of $\bar{u}^k$ given by (2.55) must be equal to $\bar{U}^k$. That is,

$$I_{k-1}^k w^{k-1} = I_{k-1}^k[\bar{U}^{k-1} - I_k^{k-1}\bar{U}^k] = 0,$$

which, from (5.10), implies that

$$\bar{U}^{k-1} = I_k^{k-1}\bar{U}^k.$$

Substituting into (***) and noting (2.45), we obtain (5.11) through (5.13). □

The following remarks follow from Lemma 5.1.

1. Lemma 5.1 brings out an interesting difference between multigrid methods for equations and for inequalities. For equations, $\bar{F}^k - L^k\bar{U}^k = 0$ and conditions (5.11)–(5.13) are satisfied for any reasonable choice of S_k^{k-1} and I_k^{k-1}, but this is not true for inequalities.

2. Since $\bar{U}^k$ solves (2.41), inequalities (5.11) and (5.12) will certainly hold if S_k^{k-1} and I_k^{k-1} map nonnegative vectors into nonnegative vectors. In particular, this will be the case if S_k^{k-1} and I_k^{k-1} take linear combinations of values with nonnegative weights.

3. If S_k^{k-1} and I_k^{k-1} are injections, then (5.13) is implied by (2.41c).

4. If S_k^{k-1} is defined by (5.8) and (5.9) while I_k^{k-1} is injection, then (5.13) does not hold in general. This is because in general there will be points $x, y \in G^k$ such that $x \in G^{k-1}$, $\bar{U}^k(x) > 0$, $\bar{U}^k(y) = 0$, y is a neighbor of x in G^k and $(\bar{F}^k - L^k\bar{U}^k)(y) > 0$. Then

$$I_k^{k-1}\bar{U}^k(x) = \bar{U}^k(x) > 0$$

and

$$(S_k^{k-1}(\bar{F}^k - L^k\bar{U}^k))(x) \geqq \tfrac{1}{4}(\bar{F}^k - L^k\bar{U}^k)(y) > 0,$$

so that (5.13) does not hold. This explains why Modification M4 of PFAS did not converge.

We now describe two further modifications of PFAS which were tried:

M5. Bearing Lemma 5.1 in mind, it is possible to introduce weighted sums for which (5.13) does hold. One choice uses weighted residuals only near the boundary:

$$4S_k^{k-1}r^k(x) = \begin{cases} 4r^k(x) & \text{if } \bar{u}^k(x) = 0 \text{ or if } \bar{u}^k(y) > 0 \\ & \text{for all eight neighbors } y \in G^k \text{ of } x, \\ \sum_{\Delta} \rho(\Delta) r^k(x + \Delta h_k) \text{ signum } [\bar{u}^k(x + \Delta h_k)] & \text{otherwise,} \end{cases} \tag{5.14}$$

where

$$\text{signum } \alpha = \begin{cases} 1 & \text{if } \alpha > 0, \\ 0 & \text{if } \alpha = 0, \end{cases}$$

and where the weights $\rho(\Delta)$ are as in (5.9). This modification converged more slowly than with PFAS, and it was found that $\mathring{\mu}_f = .854$.

M6. As mentioned in D1 and D3 above, if $\bar{u}^k(x) = 0$ then it may happen that $\bar{u}^k(x) = \bar{u}^k(x) + I_{k-1}^k w^{k-1}(x)$ is not zero. It can be argued that changes of $\bar{u}^k(x)$ from or to zero should only be done on G^k. We, therefore, modified PFAS so that in (2.55) φ was defined by

$$\varphi(\tilde{u}^k(x); \bar{u}^k(x)) = \begin{cases} \tilde{u}^k(x) & \text{if } \bar{u}^k(x) > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5.15}$$

I_k^{k-1} and S_k^{k-1} were injections. This program was called PFASMD.

We solved (4.1), (4.2) with $M = 5$ using PFAS and PFASMD. In each case, the computations were terminated when $\|\nabla \bar{u}^M\|_G \leqq 2 \cdot 10^{-8}$. The results are summarized in Table 5.1.

In Table 5.2 we compare PFAS and PFASMD for the problem (5.3), (5.4). As in Table 5.1 we iterated until $\|\nabla u^{(k)}\|_G \leqq 2 \cdot 10^{-8}$ on G^5.

We conclude from the results given in Tables 5.1 and 5.2 that PFASMD is substantially faster than PFAS.

Finally, in Table 5.3 we extend Table 4.2 by comparing the measured execution times for the projected SOR method and PFASMD for the dam problem for various values of M. In each case, the iterations were continued until $\|\nabla \bar{u}\|_G \leqq 2 \cdot 10^{-8}$.

TABLE 5.1

Solution of (4.1), (4.2) *with* $M = 5$ *and* $\varepsilon^M = 2.10^{-8}$ *using* PFAS *and* PFASMD.

	Method	
	PFAS	PFASMD
Work units	96.15	42.81
Execution time (seconds)	3.40	1.63
$\mathring{\mu}_f$	.815	.623

TABLE 5.2

Solution of (5.3), (5.4) *with* $M=5$, $R=32/15$ *and* $\varepsilon^M=2\cdot 10^{-8}$ *for* PFAS *and* PFASMD.

	Method	
	PFAS	PFASMD
Work units	73.62	56.96
Execution time (seconds)	3.09	2.58
$\mathring{\mu}_f$	.731	.669

TABLE 5.3

Comparison of G^M *projected* SOR *and* PFASMD *for the dam problem with* $\varepsilon^M=2\cdot 10^{-8}$.

	$M=$	2	3	4	5	6
	$G^M=$	5×7	9×13	17×25	33×49	65×97
G^M projected SOR	G^M iterations	19	34	69	146	295
	Execution time (seconds)	.02	.09	.60	4.88	39.37
PFASMD	G^M work units	23	30.5	38.7	42.8	45.7
	Execution time (seconds)	.04	.12	.41	1.64	6.57

As can be seen from Table 5.3, PFASMD is better than projected SOR except for very small grids.

6. PFMG (projected full multigrid algorithm). In this section we describe PFMG (projected full multigrid algorithm), which is a modification of the full multigrid algorithm of Brandt. The flowchart for PFMG is given in Fig. 6.1. PFMG has been implemented as a FORTRAN subroutine for the case when Ω is a rectangle in R^2, and $\mathscr{L}$ is the Laplacian operator. This subroutine is listed in Brandt and Cryer [1980] as part of a program for solving the porous flow free boundary problem of § 4, and the problem (5.3), (5.4).[1]

PFMG differs from PFAS in the following respects:

I. Instead of beginning on G^M, one begins on a coarser grid G^{LIN} and gradually works up to G^M. The computations begin on the initial grid G^l, $l=\mathrm{LIN}$, with an initial approximation $\bar{u}^l$. $\bar{u}^l$ is computed to the required accuracy using grids G^1 through G^l as in PFAS, except that, as will be discussed below, the decision to move to a different grid is based on slightly different criteria.

Once $\bar{u}^l$ has been found to sufficient accuracy, the initial approximation $\bar{u}^{l+1}$ is obtained from

$$\bar{u}^{l+1}=J_l^{l+1}\bar{u}^l, \tag{6.1}$$

where J_l^{l+1} is an interpolation operator taking grid functions on G^l into grid functions on G^{l+1}. It is known (Brandt [1977, p. 377]) that J_l^{l+1} should be more accurate than I_l^{l+1} in order to preserve the smoothness of $\bar{u}^l$.

[1] There are two errors in the program as listed in Brandt and Cryer [1980]. On line 1127 change (ITAU.EQ.1) to (ITAU.EQ.1 · AND · T.NE.0). Card 1123 (TAUGNM = TAUGNM + T*T) should be placed after card 1126 (*Q(IP+JK) . EQ.0) $T=0$).

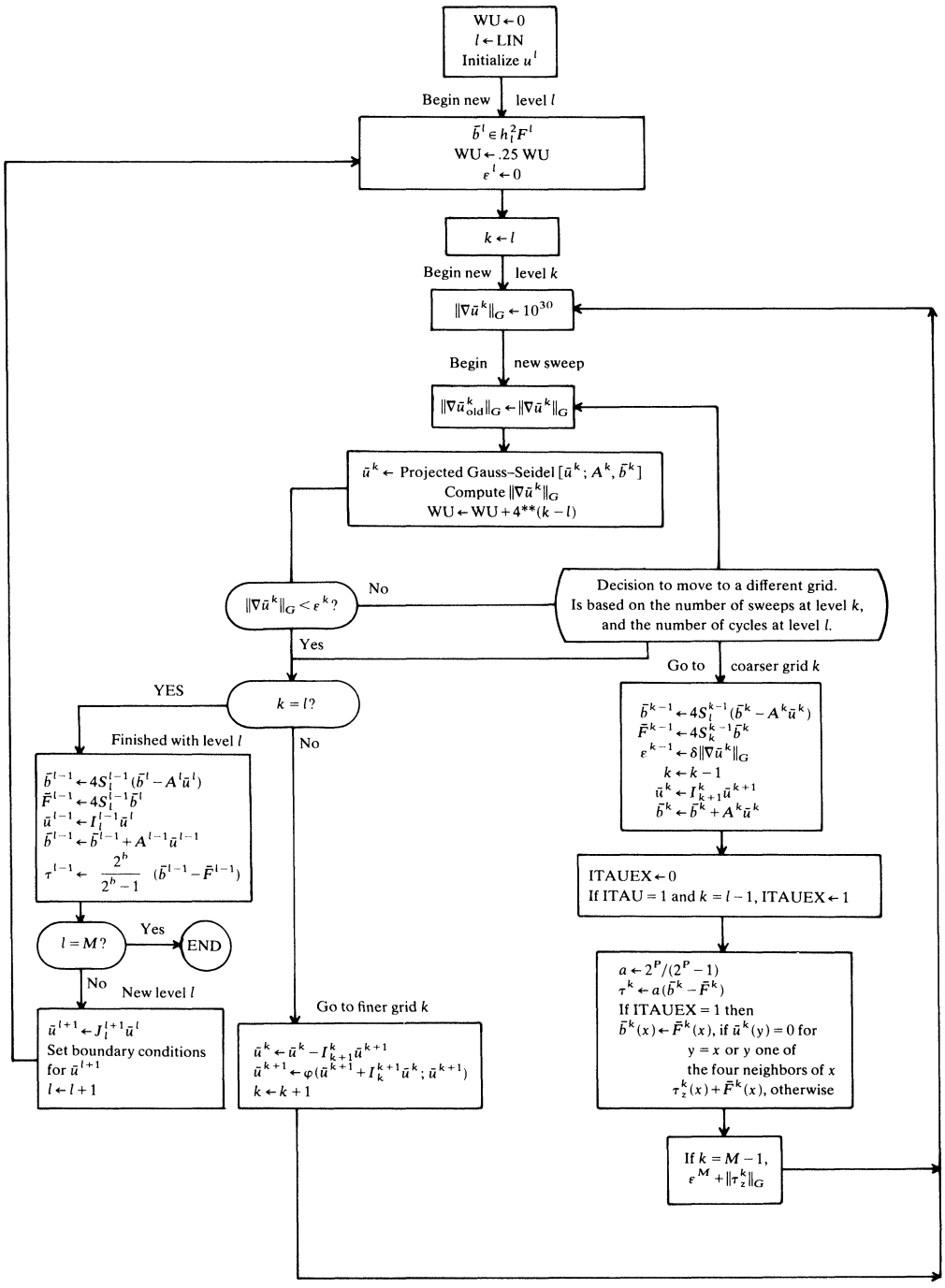

FIG. 6.1. *Flow chart for* PFMG.

In PFMG, J_l^{l+1} is based upon repeated use of the cubic interpolation formulas

$$f(\tfrac{1}{2}) = [-f(-1) + 9f(0) + 9f(1) - f(2)]/16, \tag{6.2}$$

$$f(\tfrac{3}{2}) = [f(-1) - 5f(0) + 15f(1) + 5f(2)]/16. \tag{6.3}$$

Repeating this process, we finally obtain an initial approximation $\bar{u}^M$ on G^M. Thereafter, the computation proceeds essentially as in PFAS.

II. $\bar{u}^k$ is used to estimate the local truncation error on G^{k-1}. Suppose that the difference approximations are of order p and that $\bar{u}^k$ can be extended to a smooth function on Ω. Then on G^{k-1},

$$A^{k-1} I_k^{k-1} \bar{u}^k \doteq h_{k-1}^2 \mathscr{L} \bar{u}^k + \tau^{k-1} \tag{6.4}$$

and

$$S_k^{k-1} A^k \bar{u}^k \doteq h_k^2 \mathscr{L} \bar{u}^k + 2^{-(p+2)} \tau^{k-1}, \tag{6.5}$$

where the *local truncation error* τ^{k-1} depends upon the derivatives of $\bar{u}^k$. Eliminating the unknown $\mathscr{L}\bar{u}^k$ we obtain

$$\tau^{k-1} \doteq \frac{2^p}{2^p - 1} [A^{k-1} I_k^{k-1} \bar{u}^k - 4 S_k^{k-1} A^k \bar{u}^k] \tag{6.6}$$

$$= \frac{2^p}{2^p - 1} [\{4 S_k^{k-1} (\bar{b}^k - A^k \bar{u}^k)\} + \{A^{k-1} I_k^{k-1} \bar{u}^k\} - \{4 S_k^{k-1} \bar{b}^k\}]. \tag{6.7}$$

The estimate (6.7) is not accurate near the discrete interface, and so PFMG computes τ_z^{k-1} where

$$\tau_z^{k-1}(x) = \begin{cases} \tau^{k-1}(x), & \text{if } \bar{u}^{k-1}(x) > 0, \\ 0 & \text{if } \bar{u}^{k-1}(x) = 0. \end{cases} \tag{6.8}$$

Because of the lack of smoothness of the solution near the free boundary, it is not entirely clear what the value of p should be. It is known (Brezzi and Sacchi [1976]) that the convergence of the finite difference approximations is probably only $O(h^1)$ in the $W^{1,2}(\Omega)$ norm, and Nitsche [1975] has proved $O(h^2 \ln h)$ convergence in the infinity norm. However, these are global error bounds, while we are concerned with the asymptotic behavior of the local truncation error τ. Except in a neighborhood of the discrete interface Γ^l, p is clearly equal to 2. Since the choice of p may vary over Ω, we could perhaps set $p = 1$ near Γ^l, but the values of τ near Γ^l are not very accurate and so, for simplicity, we have taken $p = 2$ everywhere.

III. As usual in numerical analysis, the estimate (6.7) for τ^{k-1} can be used in two ways:

(a) To estimate the error $\bar{u}^k - u$. Since $\tau^k \doteq 2^{-2-p} \tau^{k-1}$, and remembering that G^k has four times as many points as G^{k-1} but $h_{k-1} = 2h_k$, we see from (2.23) that

$$\|\tau_z^k\|_G \doteq \|\tau_z^{k-1}\|_G / 2^p. \tag{6.9}$$

Combining (6.7), (6.8) and (6.9) we obtain an estimate for $\|\tau_z^k\|_G$.

In the previous sections we were concerned with asymptotic convergence. That is, we were concerned with the rate of convergence of $\bar{u}^k$ to $\bar{U}^k$ over a very large number of iterations. However, if we want an approximation to the solution u of (1.1), it is only necessary to iterate until the residual on G is small compared with the truncation error, that is, until

$$\|\nabla \bar{u}^k\|_G = O(\|\tau_z^k\|_G). \tag{6.10}$$

Once (6.10) holds, further computation will improve the accuracy of $\bar{u}^k$ as a solution of the finite difference equations, but will not improve its accuracy as an approximation to u. Noting (6.9), we see that (6.10) will certainly be true if

$$\|\nabla \bar{u}^k\|_G \leqq \|\tau_z^{k-1}\|_G. \tag{6.11}$$

(b) Improvement of accuracy of $\bar{u}^{k-1}$. Once an estimate for the truncation error τ_z^{k-1} is available, it can be used to improve the accuracy of the difference approximation on G^{k-1} by replacing $F^{k-1}(x)$ by $F^{k-1}(x)+\tau_z^{k-1}(x)$ (see (6.4)). This is only done at points $x \in G^{k-1}$ such that $\bar{u}^{k-1}(y)>0$ for all four neighbors $y \in G^{k-1}$ of x since the value of τ_z^{k-1} is not accurate elsewhere.

Of course, this is only meaningful when $\|\tau_z^{k-1}\|_G$ is small compared to $\|\nabla \bar{u}^k\|_G$: if the iterations are continued for a long time, then convergence will not occur because the conditions of Lemma 5.1 will be violated; but PFMG is never used in this way. In fact, experience with equalities indicates that when τ-extrapolation is used, the best procedure is to avoid relaxation after returning for the last time to the finest grid.

IV. As already mentioned, the logic of PFMG is more complicated than that of PFAS. Several parameters are introduced, and this enables one to control explicitly the number of G^k-projected sweeps at any level k, and the number of cycles at level l. In the computations reported on here, in each cycle on grid G^l two G^k-projected sweeps are carried out for $1<k \leqq l$ as we descend from G^l to G^1, and one G^k-projected sweep is carried out as we ascend from G^1 to G^l. For $l=\mathrm{LIN}$, up to three G^l cycles are allowed, so that a good initial approximation can be obtained. For $\mathrm{LIN}<l<M$, only one G^l cycle is allowed, while up to 10 G^M cycles are allowed.

We now describe numerical results obtained using PFMG to solve the dam problem (4.1), (4.2). In all cases, G^1 is a $(2+1)\times(3+1)$ grid and $\mathrm{LIN}=2$.

PFMG includes the option of computing $\|\bar{u}^l-u\|_\infty$ and $\|\bar{u}^l-u\|_G$, where u is the exact solution. For the dam problem, it is possible to compute u analytically using elliptic integrals (Cryer [1976]), but this has not yet been done: we therefore took u to be the most accurate approximation known to us, namely the approximation $\bar{u}^7$ computed in double precision on a $(128+1)\times(192+1)$ grid as described in § 4. For problem (5.3), (5.4) the exact solution is given by (5.4).

We first performed a number of experiments with $M=2$, 3, 4, and 5:

1. τ-extrapolation (with $p=2$) gave slightly worse results for the dam problem and slightly better results for problem (5.3), (5.4). (This is explained in part by the observed behavior of τ as a function of h, as discussed below.)

2. In contrast to our experience with PFASMD, the use of equation (5.15) (modification M6) had only a slight effect. (This may be explained by the observation that the slow convergence of PFAS is caused by the existence of gridpoints near the discrete interface at which (without modification M6) the computed values of $\bar{u}^k$ fluctuate between zero and small positive quantities. This is a delicate asymptotic matter below the level of the truncation error, and hence does not trouble FMG.)

3. It was thought that convergence might be improved by multiplying the difference $\nabla \bar{u}^k(x)$ by h for points x near the free boundary before computing $\|\nabla \bar{u}^k(x)\|_G$. This was found to have negligible effect.

All the results given below are for the case of no τ-extrapolation and no modification.

The results for the dam problem for different values of M are shown in Table 6.1.

Since we only have estimates for τ^{M-1}, it is not possible to obtain rigorous error bounds. Nevertheless, it is interesting to apply the error bounds of § 2.

TABLE 6.1

Solution of the dam problem using PFMG.

	M		
	3	4	5
G^M work units	8.75	6.67	6.41
Execution time (seconds)	.0675	.145	.404
$\|\bar{u}^M - \bar{u}^7\|_\infty / \|u\|_\infty$	.000665	.000168	.0000532
$\|\bar{u}^M - \bar{u}^7\|_G / \|\bar{U}^M\|_G$	.000810	.000145	.0000388
$\|\nabla \bar{u}^M\|_G$	.0666	.0557	.0339
$\|\tau_z^{M-1}\|_G$	0.0771	0.0795	0.0383

Let $\bar{U}^M$ denote the vector obtained by evaluating the solution $u(x)$ on G^M. Then, from (6.4), (1.1), (2.2), (2.3), (2.13) and (3.1),

$$A^M \bar{U}^M \leqq b^M + \tau^M, \tag{6.12}$$

so that, from Lemma 2.1,

$$\|\bar{U}^M - U^M\|_2 \leqq \frac{1}{\alpha_M} \|\tau_+^M\|_2. \tag{6.13}$$

On the other hand, from Lemma 2.2,

$$\|U^M - \bar{u}^M\|_2 \leqq \frac{1}{\alpha_M} \|P^M\|_2 \|\nabla \bar{u}^M\|_2.$$

For the dam problem, P is an upper triangular matrix with at most two nonzero elements per row, and $\|P^M\|_2 \leqq 2$. Thus,

$$\|U^M - \bar{u}^M\|_2 \leqq \frac{2}{\alpha_M} \|\nabla \bar{u}^M\|_2. \tag{6.14}$$

Combining these inequalities we obtain

$$\|\bar{U}^M - \bar{u}^M\|_2 \leqq \frac{1}{\alpha_M} [\|\tau_+^M\|_2 + 2\|\nabla \bar{u}^M\|_2],$$

or, equivalently,

$$\|\bar{U}^M - \bar{u}^M\|_G \leqq \frac{1}{\alpha_M} [\|\tau_+^M\|_G + 2\|\nabla \bar{u}^M\|_G]. \tag{6.15}$$

Using (6.8) and (6.9), we conclude that

$$\|\bar{U}^M - \bar{u}^M\|_G \doteqdot \frac{1}{\alpha_M} \left[\frac{1}{2^p} \|\tau_z^{M-1}\|_G + 2\|\nabla \bar{u}^M\|_G\right]. \tag{6.16}$$

Next, we note that for the dam problem

$$\alpha_M \doteqdot \alpha h_M^2, \tag{6.17}$$

where

$$\alpha = \left(\frac{\pi}{16}\right)^2 + \left(\frac{\pi}{24}\right)^2 \doteqdot .055 > 14/256 \tag{6.18}$$

and

$$h_M = 16 \cdot 2^{-M}.$$

Thus, finally, for the dam problem,

$$\|\bar{U}^M - \bar{u}^M\|_G \lessapprox \frac{2^{2M}}{14}\left[\frac{1}{2^p}\|\tau_z^{M-1}\|_G + 2\|\nabla \bar{u}^M\|_G\right]. \tag{6.19}$$

For example, for $M = 5$ we obtain, using Table 6.1, that

$$\|\bar{U}^5 - \bar{u}^5\|_G / \|\bar{U}^5\|_G \lessapprox \frac{2^{10}}{14}[\tfrac{1}{4}(0.0383) + 2(.0339)]/(5.9\ 10^3) \doteq .000959; \tag{6.20}$$

the observed value quoted in Table 6.1 is .0000388.

In Table 6.2 we repeat the computations of Table 6.1 for the problem (5.3), (5.4).

TABLE 6.2

Solution of problem (5.3), (5.4) *using* PFMG.

	M		
	3	4	5
G^M work units	6.75	5.672	5.414
Execution time (seconds)	.101	.271	.861
$\|\bar{u}^M - \bar{U}^M\|_\infty / \|u\|_\infty$	.000985	.000266	.0000645
$\|\bar{u}^M - \bar{U}^M\|_G / \|\bar{U}^M\|_G$	.00122	.000376	.0000956
$\|\nabla \bar{u}^M\|_G$	.241	.121	.0764
$\|\tau_z^{M-1}\|_G$	1.56	.509	.147

The error estimate (6.19) also holds for the problem (5.3), (5.4), since we are using the Laplace operator on a rectangle with sides in the ratio 2:3. Applying (6.19) we obtain

$$\|\bar{U}^5 - \bar{u}^5\|_G / \|\bar{U}^5\|_G \lessapprox \frac{2^{10}}{14}|\tfrac{1}{4}(.147) + 2(.076)|/(1 \cdot 2\ 10^4) \doteq .00115$$

the observed value quoted in Table 6.2 is .0000645.

The behavior of the global error $\bar{u}^M - u$ can be checked using Tables 6.1 and 6.2. From Table 6.1 we have

$$\left[\frac{\|\bar{u}^5 - \bar{u}^7\|_\infty}{\|\bar{u}^3 - \bar{u}^7\|_\infty}\right]^{1/2} = \left[\frac{.0000532}{.000665}\right]^{1/2} \doteq \frac{1}{2^{1.82}}.$$

From Table 6.2,

$$\left[\frac{\|\bar{u}^5 - u\|_\infty}{\|\bar{u}^3 - u\|_\infty}\right]^{1/2} = \left[\frac{.0000645}{.000985}\right]^{1/2} \doteq \frac{1}{2^{1.96}}.$$

These results strongly suggest that the global error is $O(h^2)$.

The behavior of the local error τ can also be checked using Tables 6.1 and 6.2. From Table 6.1,

$$[\|\tau_z^4\|_G / \|\tau_z^2\|_G]^{1/2} = [.0383/.0771]^{1/2} \doteq 1/2^{.50},$$

while, from Table 6.2,

$$[\|\tau_z^4\|_G/\|\tau_z^2\|_G]^{1/2} = [.147/1.56]^{1/2} \doteq 1/2^{1.7},$$

so that $\tau = O(h^q)$ with $q \in (.50, 1.7)$. This explains why τ-extrapolation with $p = 2$ did not reduce the computational effort for the dam problem. The essential difficulty is of course that the irregularity of the discrete interface makes it difficult to obtain accurate estimates for τ.

Finally, in Table 6.3 we repeat the computations of Table 5.3 for a tolerance $\varepsilon^M = .0339$, the value of $\|\nabla \bar{u}^5\|_G$ in Table 6.1. We are thus comparing the performance

TABLE 6.3

Solution of the dam problem for $M = 5$ *and* $\varepsilon^M = .0339$ *using* PFASMD (*modification* M6), PFMG, *and projected* SOR.

	Method		
	PFMG	PFASMD	Projected SOR
Work units	6.41	9.64	60.0
$\|\nabla \bar{u}^M\|_G$	.0339	.0239	.0296
Execution time (seconds)	.404	.447	2.07

of PFAS (with Modification M6), PFMG and projected SOR for comparable errors. From Table 6.3, we see that PFMG is faster than projected SOR even when only low accuracy is required. PFAS and PFMG require comparable times, but PFMG gives much more information and is, therefore, preferable. PFMG also uses fewer work units than PFAS. This is significant because the number of work units used is independent of the computer. Furthermore, on the basis of experience with many problems, it can be said that the number of work units used does not vary greatly with the problem: for most operators $\mathscr{L}$, FMG requires only 5.4 work units.

We conclude this section with some remarks on the implementation of PFMG.

1. From Table 6.3 we see that the execution time per work unit of PFMG is greater than the comparable quantity for PFAS by a factor

$$\frac{.404}{6.41} \Big/ \frac{.447}{9.64} = 1.36.$$

This additional overhead is probably due to the cubic interpolation used by J_{k-1}^k, and could perhaps be reduced by better programming. When $\mathscr{L}$ is complicated, the additional overhead required by PFMG is relatively much less significant: it is only with a very simple operator like the 5-point Laplacian that the additional overhead is so expensive.

2. In PFMG one often need not have *any* storage for the finest grid G^M—not even external storage. The algorithm visits G^M only twice: at the beginning of the last cycle and at the end of the last cycle.

At the beginning of the cycle, the following operations are performed: interpolate (J_{M-1}^M); two G^M-projected sweeps; and residual transfer (I_M^{M-1} and S_M^{M-1}). All these operations can be made in one passage over G^M in such a way that only four columns of G^M are held in memory at one time. Each time a new column, say column i, is

created (by interpolation), a relaxation can be made in column $i-1$; then the second relaxation can already be made in column $i-2$ and the residuals from column $i-3$ can be transferred back to the coarse grid. Column $i-4$ can simultaneously be discarded (i.e., replaced by column i). After this visit to G^M, all the information is available (in $\bar{F}^{M-1}$ and $\bar{u}^{M-1}$) to solve the G^{M-1} problem to the truncation level of G^M.

The final return to G^M (which would require the storage of the previous values of U^M) is made in order to obtain the solution on G^M rather than on G^{M-1}, but it does not improve its pointwise accuracy. If one is interested only in knowing some functionals of the solution, these can be calculated without having the final solution on G^M. To approximate a functional $\mathscr{H}(U)$, for example, one computes $\mathscr{H}(\bar{u}^{M-1})+\sigma_M^{M-1}$, where $\sigma_M^{M-1}=\mathscr{H}(\bar{u}^M)-\mathscr{H}(I_M^{M-1}\bar{u}^M)$, $\bar{u}^{M-1}$ is the final solution on G^{M-1}, and $\bar{u}^M$ is the last solution on G^M before switching back to G^{M-1}. Clearly, σ_M^{M-1} can be calculated during the above-mentioned passage on G^M. Note that σ_M^{M-1} is a "relative truncation correction", similar to τ_M^{M-1}. It makes the approximation $\mathscr{H}(\bar{u}^{M-1})+\sigma_M^{M-1}$ correct to the G^M truncation level. $\mathscr{H}$ need not be a linear functional.

7. Conclusions and recommendations.

1. Multigrid methods can easily be adapted to handle linear complementarity problems arising from free boundary problems.

2. Multigrid methods are superior to projected SOR and modified block SOR (see Tables 5.3 and 6.3).

3. For high accuracy solutions of the discrete LCP, one should use PFASMD (see Tables 5.1 and 5.2).

4. For solutions which are accurate to within truncation error, one should use PFMG with no modifications (see Tables 6.1, 6.2, and 6.3).

Finally, we conclude with some comments suggesting possible future applications of multigrid methods to complementarity problems:

1. For equalities, experience has shown that multigrid methods are as efficient for problems where $\mathscr{L}$ is nonlinear as for problems where $\mathscr{L}$ is linear.

2. Experience from equalities indicates that with similar efficiency (just a few more work units), one can solve much more difficult problems, such as problems in which the coefficients of $\mathscr{L}$ vary by orders of magnitude (e.g., large variations in the diffusivity of the dam). In such cases SOR and other methods converge very slowly. See Alcouffe et al. [1980].

3. The truncation error near a discrete interface cannot be reduced by using higher order approximations, because the second derivatives are usually discontinuous. A good way to improve the approximation would be to use finer mesh sizes near the discrete interface. This can be combined very effectively effectively with the multigrid process (see Brandt [1979, § 3]). In fact, a vast improvement is to be expected if τ-extrapolation is used *together* with local refinements. Fine levels will then be used only near the interface.

4. It would be possible to use a parallel processor, in which case the Gauss–Seidel iterations would be performed using the red–black ordering of the grid points (Brandt [1980a], Foerster et al. [1980], Cryer et al. [1981]).

5. Although the numerical results for PFAS and PFMG are convincing, it would be desirable to obtain a rigorous proof of convergence, such as is available for the projected SOR method.

Acknowledgment. We thank Professor R. Sacher for making available a copy of his program for solving LCP's using the modified block SOR algorithm of Cottle and Sacher, and for his comments on an early version of this report.

REFERENCES

R. ALCOUFFE, A. BRANDT, J. E. DENDY, JR. AND J. W. PAINTER (1980), *The multi-grid methods for the diffusion equation with strongly discontinuous coefficients*, Report LA-UR-80-1463, Los Alamos Scientific Laboratory, Los Alamos, NM.

C. BAIOCCHI (1971), *Sur un problème à frontière libre traduisant le filtrage des liquides à travers des milieux poreux*, Comptes Rendus Acad. Sci. Paris Ser. A, 273, pp. 1215–1217.

——— (1978), *Free boundary problems and variational inequalities*, Technical Summary Report 1883, Mathematics Research Center, University of Wisconsin, Madison.

M. L. BALINSKI AND R. W. COTTLE (1978), *Complementarity and Fixed Point Problems*, North-Holland, Amsterdam.

J. BEAR (1972), *Dynamics of Fluids in Porous Media*, American Elsevier, New York.

A. BRANDT (1977), *Multi-level adaptive solutions to boundary value problems*, Math. Comp., 31, pp. 333–390.

——— (1979), *Multilevel adaptive techniques* (MLAT) *for singular-perturbation problems*, in Numerical Analysis of Singular Perturbation Problems, P. W. Hemker and J. J. H. Miller, eds., Academic Press, New York, pp. 53–142.

——— (1980), *Stages in developing multigrid solutions*, in Numerical Methods for Engineering, E. Absi, R. Glowinski, P. Lascaux and H. Veysseyre, eds., Dunod, Paris, pp. 23–44.

———(1980a), *Multigrid solvers on parallel computers*, in Elliptic Problem Solvers, M. Schulz, ed., Academic Press, New York.

A. BRANDT AND C. W. CRYER (1980), *Multigrid algorithms for the solution of linear complementarity problems arising from free boundary problems*, Technical Summary Report 2131, Mathematics Research Center, University of Wisconsin, Madison.

A. BRANDT AND N. DINAR (1979), *Multi-grid solutions to elliptic flow problems*, in Symposium on Numerical Solution of Partial Differential Equations, S. V. Parter, ed., Academic Press, New York, pp. 53–147.

F. BREZZI AND G. SACCHI (1976), *A finite element approximation of variational inequalities related to hydraulics*, Calcolo, 13, pp. 259–273.

J. CEA, R. GLOWINSKI AND J. C. NEDELEC (1974), *Application des méthodes d'optimisation, de differences et d'éléments finis à l'analyse numérique de la torsion élasto-plastique d'une barre cylindrique*, in Approximation et méthodes iteratives de resolution d'inéquations variationelles et de problèmes non linéaires, Cahier de l'IRIA, 12, pp. 7–138.

G. CIMATTI (1977), *On a problem of the theory of lubrication governed by a variational inequality*, Appl. Math. Optim., 3, pp. 227–242.

R. W. COTTLE (1974), *Computational experience with large-scale linear complementarity problems*, Technical Report SOL 74-13, Systems Optimization Laboratory, Department of Operations Research, Stanford Univ., Stanford, CA.

R. W. COTTLE, F. GIANNESSI AND J. L. LIONS, eds. (1980), *Variational Inequalities and Complementarity Problems*, John Wiley, New York.

R. W. COTTLE, G. H. GOLUB AND R. S. SACHER (1978), *On the solution of large, structured linear complementarity problems: The block partitioned case*, Appl. Math. Optim., 4, pp. 347–363.

R. W. COTTLE AND R. S. SACHER (1977), *On the solution of large, structured linear complementarity problems: The tridiagonal case*, Appl. Math. Optim., 3, pp. 321–340.

C. W. CRYER (1971), *The solution of a quadratic programming problem using systematic overrelaxation*, SIAM J. Control, 9, pp. 385–392.

——— (1971a), *The method of Christopherson for solving free boundary problems for infinite journal bearings by means of finite differences*, Math. Comp., 25, pp. 435–444.

——— (1976), *A survey of steady state porous flow free boundary problems*, Technical Summary Report 1657, Mathematics Research Center, University of Wisconsin, Madison.

——— (1977), *A bibliography of free boundary problems*, Technical Summary Report 1793, Mathematics Research Center, University of Wisconsin, Madison.

——— (1980), *The solution of the axisymmetric elastic-plastic torsion of a shaft using variational inequalities*, J. Math. Anal. Appl., 76, pp. 535–570.

——— (1980a), *Successive overrelaxation methods for solving linear complementarity problems arising from free boundary problems*, in Proceedings, Seminar on Free Boundary Problems, Pavia, October 1979, E. Magenes, ed., Istituto Nazionale di Alta Matematica Francesco Severi, Rome, vol. 2, pp. 109–131.

C. W. CRYER, P. M. FLANDERS, D. J. HUNT, S. F. REDDAWAY AND J. STANSBURY (1981), *The solution of linear complementarity problems on an array processor*, Technical Summary Report 2170, Mathematics Research Center, University of Wisconsin, Madison; J. Comput. Phys., to appear.

G. DUVAUT AND J. L. LIONS (1976), *Inequalities in Mechanics and Physics*. Dunod, Paris.

R. S. FALK (1974), *Error estimates for the approximation of a class of variational inequalities*, Math. Comp., 28, pp. 963–971.

H. FOERSTER, K. STUEBEN AND U. TROTTENBERG (1980), *Non-standard multigrid techniques using checkered relaxation and intermediate grids*, in Elliptic Problem Solvers, M. Schulz, ed., Academic Press, New York.

R. GLOWINSKI (1971), *La méthode de rélaxation*, Rend. Mat., 14, pp. 1–56.

——— (1978), *Finite elements and variational inequalities*, Technical Summary Report 1885, Mathematics Research Center, University of Wisconsin, Madison.

R. GLOWINSKI, J. L. LIONS AND R. TREMOLIERES (1976), *Analyse numérique des inéquations variationnelles*, Dunod, Paris.

D. KINDERLEHRER AND G. STAMPACCHIA (1980), *An Introduction to Variational Inequalities and Their Applications*, Academic Press, New York.

H. LANCHON (1974), *Torsion élastoplastique d'un arbre cylindrique de section simplement ou multiplement connexé*, J. Mécanique, 13, pp. 267–320.

J. A. NITSCHE (1975), *L-infinity convergence of finite element approximations*, in Mathematical Aspects of Finite Element Methods, Lecture Notes in Mathematics 606, Springer, Berlin.

R. S. VARGA (1962), *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ.