JOURNAL OF COMPUTATIONAL PHYSICS 52, 105–121 (1983)

A Numerical Method for the Nonlinear Neumann Problem

R. KANNAN*

Department of Mathematics, The University of Texas at Arlington, Arlington, Texas 76019

AND

W. PROSKUROWSKI†

Department of Mathematics, University of Southern California, Los Angeles, California 90089-1113

Received April 13, 1982; revised February 10, 1983

An iterative scheme for the nonlinear Neumann problem over a smooth bounded domain in R^2 is studied. An equivalent system of coupled problems is obtained by the method of "Alternative Problems." A combination of the capacitance matrix method and one-dimensional nonlinear solver is then applied to this system yielding an efficient numerical algorithm.

1. INTRODUCTION

In this paper, we study a numerical method for solving the nonlinear Neumann problem

$$\begin{aligned} -\Delta u + f(u) &= g(x) \qquad \text{in } \Omega, \\ \partial u/\partial n &= h(s) \qquad \text{on } \partial\Omega, \end{aligned} \tag{1.1}$$

where Ω is a smooth bounded domain in R^2 (or R'), $x \in \Omega$, and f is an increasing function. We develop an iterative scheme which gives rise to a coupled system of equations. This system of equations is obtained by utilizing the fact that the nullspace of the Laplacian operator together with Neumann boundary conditions is generated by the constant functions.

The nonlinear Neumann problem has been studied from the computational point of view in [10, 13]. In [10] Keller discusses monotone convergence and related methods for nonlinear Neumann problems. The problem has also been studied in [20] by using the M-matrix nature of the discrete problem. In this paper, we proceed in a

* Research partially supported by U.S. Army Research Grant DAAG20-80-C-0060.
† Supported in part by National Science Foundation Grant MCS-8003382 at the University of Southern California.

0021-9991/83 $3.00

different direction. Using the method of "Alternative Problems" we decompose the nonlinear problem (1.1) into a system of two equations: one on the range of $-\Delta u$ and the other on the kernel that is one dimensional. We then introduce the iterative procedure that follows the lines of [3]. As starting points for the iterative process we use upper and lower solutions of (1.1) and finding these in an integral part of the process. This iterative scheme enables us to use a fast-solver capacitane matrix method [19] for the equation on the range and a one-dimensional nonlinear solver [1] for the equation on the nullspace. Numerical experiments have demonstrated the computational efficiency of this procedure. In further sections we shall outline this method. It should be noted that a discretization by a finite difference method gives rise to a problem in R^n of the type

$$Au + F(u) = G,$$

where A is a $n \times n$ singular matrix and $F: R^n \to R^n$ is a nonlinear transformation. Solving the problem over the range and kernel separately enables us to verify the Fredholm consistency conditions at each stage of the iterative process. Finally, it must be noted that splitting the nonlinear problem into problems over the range and kernel gives rise to a linear problem on the range at each stage of the iterative process. We are now in a position to apply the capacitance matrix method to solve the nonlinear problem on irregular bounded domains.

The results of the numerical experiments are also compared with the results obtained with an extension of the scheme suggested by Keller [9] and Pennline [15] for ODEs, which uses the displacement of the operator d^2/dx^2. In this paper we illustrate only numerical examples and continue further theoretical investigations elsewhere.

2. Preliminaries

In this section we outline the four concepts used in the iterative schemes: (i) Alternative Problems, (ii) Capacitance matrix method, (iii) the ZEROIN procedure, and (iv) the scheme to improve the rate of convergence.

(i) The nonlinear problem (1.1) can be written as an operator equation

$$Eu = Nu,$$

where N is the nonlinear Nemitskii operator generated by $g(x) - f(u)$ and E is the linear differential operator generated by $-\Delta$ together with the Neumann boundary conditions. We assume, for the sake of simplicity, that f is a sufficiently smooth function. Clearly, if $u \in L^2(\bar{\Omega})$ is a solution to the above problem, then we must have

$$\int_{\Omega} [f(u) - g(x)]\, dx + \int_{\partial\Omega} h(s)\, ds = \varnothing. \tag{2.1}$$

This motivates the use of the method of "Alternative Problems" [13]. Thus, for any $u \in L^2(\bar{\Omega})$ we define

$$Pu = 1/|\Omega| \int_{\Omega} u(x)\, dx.$$

We then obtain a splitting of $L^2(\bar{\Omega})$ as the direct sum of two orthogonal closed linear subspaces S_0 and S_1 where P: $L^2(\bar{\Omega}) \to S_0$ is the idempotent projection operator defined above. Looking for a solution $u \in L^2(\bar{\Omega})$ of the nonlinear problem (1.1) is thus equivalent to finding $v \in S_0$, $w \in S_1$ such that $u = v + w$ satisfies

$$Ew = (I - P)\, N(v + w),$$
$$0 = PN(v + w).$$

In terms of (1.1) these two equations are thus reduced to the system

$$\begin{aligned} -\Delta w &= (I - P)[-f(v + w) + g(x)] \qquad \text{in } \Omega, \\ \partial w/\partial n &= h(s) \qquad \text{on } \partial\Omega, \end{aligned} \tag{2.2}$$

and

$$0 = P[-f(v + w) + g(x)]. \tag{2.3}$$

The existence of a solution of this equivalent system of equations and hence of problem (1.1) may be seen in [4].

In [10], Keller proves the existence of a solution of (1.1) by applying the method of upper and lower solutions. Thus, if p_0 and q_0 are upper and lower solutions, i.e.,

$$-\Delta p_0 \geqslant -f(p_0) + g(x),$$
$$\partial P_0/\partial n \geqslant h$$

(with similar, but reversed inequalities for q_0), then Keller [10] proves that (1.1) has at least one solution $u(x)$ satisfying $q_0(x) \leqslant u(x) \leqslant p_0(x)$, $x \in \bar{\Omega}$. In addition, various iterative schemes using the upper and lower solutions are derived in [10].

(ii) The iterative scheme for Eqs. (2.2) and (2.3) that is proposed in the next section involves solving linear Neumann problems on irregular bounded smooth domains. The method utilized in this paper for such problems is the capacitance matrix matrix method.

The capacitance matrix method (CMM) is an extension of the Fast Helmholtz Solvers (FHS) to arbitrary bounded regions. A FHS is a solver for the equation

$$-\Delta u + cu = f \qquad \text{in } \Omega,$$

with proper boundary conditions on $\partial\Omega$, where c is a constant. It is usually based on

Fast Fourier Transform or cyclic reductions (or both) and is used only for regions which are rectangular after a possible change of variables.

The FHS produces a solution at a cost proportional to $N \log N$ arithmetic operations and using N storage locations, where N is the member of mesh points inside the region.

The CMM can be viewed and described in two different ways: (a) algebraic and (b) potential-theoretic. In (a) the CMM solution is obtained as a solution to a problem which is a low rank modification of the problem on a rectangle in which the arbitrary region is embedded. The low rank modification is expressed in terms of the Woodbury formula and its generalizations [2]. On the other hand, in (b) the original problem (1.1) is reduced to solving a Fredholm integral equation of the second kind (of dimensionality one less than the original problem) [19]. Similar formulations are widely used in integral equation techniques. In contrast, in CMM one does not use expansive quadrature rules, but one works implicitly with the discrete Green's function generated by a FHS. At first the CMM was presented in terms of potentials and electrodes [6]. Then an algebraic development took place [2] followed by a series of papers in which the spectral structure of the integral operators was utilized [19]. For a recent survey of the iterative variants of CMM, see [17]. Such solvers have modest storage requirements (of $\sqrt{N}$ locations for the solution and several vectors of length N) and a total computational expense of the order of 10 calls of a FHS. The currently available package program for either the Dirichlet or Neumann problems is described in [18].

(iii) Our iterative scheme for Eq. (2.3) involves locating a zero of a single-valued real function. The solver utilized in this paper is ZEROIN [1, 5]. This program requires that two points be given where the function values are of opposite signs. The method is a combination of the bisection and secant methods. Besides being globally convergent, the method is at worst linearly convergent (bisection) and for smooth functions it has the speed of the secant method. The choice of the two starting points for our problem is discussed in Section 3.

(iv) We also study an extension to PDEs of a scheme discussed by Keller for ODEs [9], and its variant due to Pennline [15]. Briefly stated, this scheme involves displacing the linear part of (1.1) by cu and thereby generates an iterative scheme based on the contraction mapping theorem. Thus the iterative scheme reduces to

$$\begin{aligned} -\Delta u_{n+1} + cu_{n+1} &= -f(u_n) + cu_n + g(x) \qquad \text{in } \Omega, \\ \partial u_{n+1}/\partial n &= h(s) \qquad \text{on } \partial\Omega, \end{aligned} \tag{2.4}$$

and thus can be written as

$$u_{n+1} = L_c^{-1} N_c u_n,$$

where L is the operator $-\Delta + cI$ together with the boundary conditions. By an appropriate choice of c, the operator $L_c^{-1}N_c$ becomes a contraction over some suitable bounded closed convex set and thus the iterations (2.4) converge.

3. Iterative Procedures for the Nonlinear Problem

As in Section 2, the nonlinear problem (1.1) can be written as an equivalent system of Eqs. (2.2) and (2.3). In this section we will discuss a numerical procedure based on Eqs. (2.2) and (2.3).

We first note that the subspace S_0 of $L^2(\bar{\Omega})$ is one-dimensional. A natural iterative procedure generated by Eqs. (2.2) and (2.3) is as follows: let $v_0 \in S_0$ and $w_0 \in S_1$ be starting points for the sequences $\{v_n\}$, $\{w_n\}$. Then let $v_{n+1} \in S_0$ and $w_{n+1} \in S_1$ be given by

$$\begin{aligned} -\Delta w_{n+1} &= (I-P)[-f(w_n+v_n)+g(x)] \qquad \text{in } \Omega, \\ \partial w_{n+1}/\partial n &= h(s) \qquad \text{on } \partial\Omega, \end{aligned} \tag{3.1}$$

and

$$0 = P[-f(w_{n+1}+v_{n+1})+g(x)]. \tag{3.2}$$

We first discuss the existence of such a $v_{n+1} \in S$ and $w_{n+1} \in S_1$.

Note that by the Fredholm alternative, Eq. (3.1) is consistent and thus is uniquely solvable for $w_{n+1} \in S_1$. We now assume that f is an increasing function of its argument. Since $v_{n+1} \in R$ it follows that (3.2) would be solvable (and thus uniquely) provided we can find $a,, b \in R$ such that

$$P[-f(w_{n+1}+a)+g(x)] \qquad \text{and} \qquad P[-f(w_{n+1}+b)+g(x)] \tag{3.3}$$

are of opposite signs. This property, namely, the existence of $a, b \in R$, is required at each stage of the iterative process. Thus, if the nonlinear problem has upper and lower solutions then the iterative procedure by (3.1) and (3.2) is well defined. This may be seen from the fact that if $A(x)$ is such that $f(A) \geqslant g(x)$ for all x in the domain then $P[f(A)-g] \geqslant 0$. Hence, if $c \in R$ is such that $c \geqslant A(x)$, $x \in \bar{\Omega}$, then such a c can be utilized to generate upper and lower solutions at each stage of the iterative process, thereby finding the constants a and b required in (3.3).

In [10] Keller studied the iterative procedure

$$\begin{aligned} -\Delta u_{n+1} + f(u_n) &= g(x), \\ b(s)\, u_{n+1} + \partial u_{n+1}/\partial n &= h(s), \qquad b(s) \not\equiv 0, \end{aligned} \tag{3.4}$$

and established that under suitable smoothness and growth hypotheses on f, the corresponding iterates $u_n(x)$ satisfy

$$u_1 \leqslant \cdots \leqslant u_{2n+1} \leqslant \cdots \leqslant u \leqslant \cdots \leqslant u_{2n} \leqslant \cdots \leqslant u_0,$$

where $u(x)$ is the solution to (1.1). However, the author states that the convergence of the alternating sequence $\{u_n, (x)\}$ is not proved in general. In [11] the convergence of such a sequence was established for a specific problem. It must be noted that (3.4)

and the system generated by (2.2) and (2.3) differ in that (3.4) does not include the case $b(x) \equiv 0$, which is the problem studied in this paper. The convergence of the sequence $\{u_n\} = \{v_n + w_n\}$ can be established from the increasing nature of f. We present an outline of the proof here.

The starting point (v_0, w_0) for our scheme can be obtained from the upper solution $A(x)$ of the original problem. Thus, $v_0 = PA(x)$, $w_0 = (I - P)A(x)$. This choice of the starting point enables us to apply the variant of the maximum principle as derived in [10]. Following the same lines of proof as in [10] we can conclude that $w_1 \leqslant w_0$. Regarding the bifurcation equation, we use a nonlinear solver referred to as the Brent procedure [5]. In order to utilize this procedure, for every n we need to obtain constants a and b such that $P[-f(a + w_n) + g(x)]$ and $P[-f(b + w_n) + g(x)]$ are of opposite signs. As remarked before, one could use the upper and lower solutions of the nonlinear problem. However, in order to improve the procedure at each stage, we proceed as follows: we choose one endpoint $a = u$ such that

$$P[-f(a + w_n) + g(x)] \geqslant 0 \qquad (\text{or } \leqslant 0).$$

Then the other endpoint b may also be obtained from

$$b = a + P[-f(a + w_n) + g(x)].$$

That b is a lower solution, i.e., $P[-f(b + w_n) + g(x)] \leqslant 0$, can be concluded from the following result in [14, p. 452]: let $F: D \subset R^n \to R^n$ be order-convex and Gateaux-differentiable on the convex set $D_0 \in D$ and suppose that there is a nonnegative $C \in L(R^n)$ such that $F'(x)C \geqslant I$, $x \in D_0$. If $Fy_0 \geqslant 0$ and $x = y - CFy \in D_0$ then $Fx_0 \leqslant 0$. In our case, we treat $F: R \to R$ to be

$$F(v) = P[-f(v + w_n) + g(x)].$$

But this also implies that v_1, which is obtained as the solution of $P[-f(v + w_1) + g(x)] = 0$, satisfies $v_1 \leqslant v_0$ so that we can conclude $u_1 = v_1 + w_1 \leqslant u_0 = v_0 + w_0$. Proceeding similarly one can show as in [10] the alternating nature of the sequence $\{u_n\}$. In order to show convergence of the entire sequence $\{u_n\}$ we show that the sequence $\{u_{2n}\}$ (and thus $\{u_{2n-1}\}$) is convergent. Finally, we show that the limits of $\{u_{2n}\}$ and $\{u_{2n-1}\}$ are the same, thereby proving the convergence of $\{u_n\}$. Noting that $\{u_{2n}\}$ is bounded we can conclude that $\{-f(u_{2n}) + g\}$ is bounded. One can then show that $\{u_{2n}\}$ is bounded in $H_1(\Omega)$ and thus there is a convergent subsequence on $L^2(\Omega)$. But $\{u_{2n}\}$ being monotone, the entire sequence is convergent. Similarly $\{u_{2n-1}\}$ is also convergent. We can now use the hypothesis that f is increasing to see that any solution to (1.1) is unique. This can be seen by an application of the maximum principle. One can also see uniqueness from the fact that if u and v are two solutions of (1.1) then $-(\Delta u - \Delta v) + (f(u) - f(v)) = 0$. Taking L^2-inner product with $u - v$, the uniqueness follows.

We conclude the discussion on the above iterative scheme with a remark on the construction of upper and lower solutions. This problem in general is difficult to

resolve. However, in the specific examples of nonlinear functions that we study in this paper, we are able to obtain upper and lower solutions as follows: proceeding as in the literature on existence of solutions of these problems [4], we can show that there exist numbers R and r such that

$$\int_\Omega [-f(R + w(x)) + g(x)] \leqslant 0 \qquad \text{and} \qquad \int_\Omega [-f(-R + w(x)) + g(x)] \geqslant 0$$

for all $\|w\| \leqslant r$. Choosing $w = 0$, for large R, we have $-f(R) + g(x) \leqslant 0$ and $-f(-R) + g(x) \geqslant 0$. We can then treat R and $-R$ as upper and lower solutions since

$$-\Delta(R) + f(R) - g(x) \geqslant 0 \qquad \text{and} \qquad -\Delta(-R) + f(-R) - g(x) \leqslant 0.$$

It must be noted here that the upper solution so obtained belongs to the kernel of the operator $-\Delta$ together with Neumann boundary conditions. This method has been utilized in the numerical examples.

Algorithm (2.2)–(2.3) has also been utilized in [8] and [12] to study numerical methods for nonlinear elliptic problems by using other hypotheses on f that guarantee local convergence.

We thus consider the iterative sequence (2.4). As in [15], the nonlinear function f is assumed to satisfy $0 \leqslant \delta \leqslant df/du \leqslant N$, i.e., $f(u)$ satisfies a two-sided Lipschitz condition with constants δ and N. One could then choose the displacement c to be $(\delta + N)/2$ and ensure that the linear transformation $-\Delta u + cu$ together with Neumann boundary conditions is invertible. Furthermore, the right-hand side of the above linear problem satisfies a Lipschitz condition. It is easy to see (cf. [7]) that for this choice of c we can (theoretically) transform the problem into an integral equation of the type

$$u = L^{-1}[cu - f(u) + g(x)] = Tu,$$

where L is the operator $-\Delta + cI$ together with Neumann boundary conditions. Then one can show that T is a strict contraction and this the sequence $\{u_{n+1}\}$ converges to a solution of the original nonlinear problem. We would like to remark, however, that it is not essential that $0 \leqslant \delta \leqslant df/du$. In other words, δ and N could be negative, but in this case one has to ensure that for the choice of $c = (\delta + N)/2$, the transformation L^{-1} can be defined. Finally, the linear problems at each stage of the iterative process can be treated efficiently with the use of fast solvers for Helmholtz equation, see [18, 22]. This is an important factor in handling nonlinear PDEs. Details of these and related discussions will be published elsewhere. Some numerical examples are provided in this paper.

4. Efficiency of the Numerical Scheme

The numerical scheme presented in the previous section is more general than its current implementation. Our choice was mainly guided by the available mathematical

software and simplicity of its application. Our aim was to develop a computationally efficient algorithm (without much effort to optimize it) which we now describe.

We imbed the irregular region D in a contiguous rectangle R and impose a uniform square mesh with the step size h. We denote the number of mesh points inside the rectangle by N, $N = nm$, where n and m are the number of mesh points in R in both coordinate directions. In each iteration we solve successively (3.1) and (3.2).

The discrete analogue of the auxiliary equation (3.1) has the same consistency properties as (3.1). Thus a properly chosen quadrature procedure for the projection operator P preserves the Fredholm consistency criterion for the discrete problem. The auxiliary equation (3.1) describes an iteration at each step of which one needs to solve a linear (Poisson) equation with the Neumann boundary condition, i.e., essentially in N unknowns. The nullspace of the Laplace operator with the Neumann boundary conditions is generated by constants. The discretization of this operator gives rise to a matrix that has a one-dimensional kernel. Therefore the bifurcation equation (3.2) is a nonlinear problem in only one variable. Consequently, the cost of solving (3.2) should be negligible if properly implemented, and the total computational expenses are dominated by Eq. (3.1). Nevertheless, the rate of convergence of the total system (3.1)–(3.2) can be slowed down considerably if (3.2) is solved inexactly, for example, by taking only one iteration

$$v_{n+1} = v_n + P[\, g(x, y) - f(v_n + w_{n+1})]. \tag{4.1}$$

Therefore, for every n, one needs to solve

$$F(v_{n+1}^k) = P[\, g(x, y) - f(v_{n+1}^k + w_{n+1})] = 0 \tag{4.2}$$

until the convergence test for the sequence of v's is satisfied. Clearly, if the starting point v is far away from the solution v, then the Picard iterations

$$v_{n+1}^{k+1} = v_{n+1}^k + F(v_{n+1}^k) \tag{4.3}$$

are too slow. Therefore, we have chosen the Brent solver ZEROIN [5] that does not require the derivative F' and combines reasonable speed of convergence with simplicity and robustness. In this way, we were able to achieve a rate of convergence for the system (3.1)–(3.2) that is virtually the same as for Eq. (3.1) alone used in cases with a zero nullspace, i.e., the Dirichlet boundary conditions. We refer to the next section for the experimental evidence.

The efficiency of solving the auxiliary equation (3.1) depends on the availability of Fast Helmholtz Solvers (FHS) on rectangular regions and their extensions to arbitrary regions by the capacitance matrix method (CMM). Neumann boundary conditions are handled by such solvers without difficulties. There exist two variants of the CMM: one in which the capacitance equations aresolved by a direct method, and the other in which they are solved by an iterative method, see [16]. The direct CMM is advantageous whenever one solves a sequence of problems that differ only in the right-hand sides, as in our case. Therefore the CMMEXP solver of [18] was

chosen. On relatively crude meshes, $N < 4{,}000$, except for the preprocessing stage, the cost of each step of the iteration (3.1) essentially is that of a FHS, i.e., proportional to $N \log N$ operations, as reported in the present paper (see the next section). On the other hand, for large meshes, $N > 4{,}000$, an iterative variant of the capacitance matrix method must be used, see Proskurowski [16, 18].

Thus at each iteration of (3.1) one is solving a Helmholtz equation with constant coefficients (the same is true if the scheme includes an operator displacement to improve the rate of convergence as in (2.4)). This expense, we repeat, is proportional to $N \log N$ operations. This compares very favorably with the cost of the Newton iterations solved by an adaptive SOR solver. In the examples that we ran, the rate of convergence of our scheme was sufficiently fast, and its simplicity outweighed the speedup of the possible improved schemes.

5. Numerical Experiments

In this section we report the results of numerical experiments carried out on the DEC 10 computer at the University of Southern California. They are divided into: preliminary investigations on simple one-dimensional models, extensive tests on square regions, and, finally, experiments on circular regions using the capacitance matrix method. We studied first the rate of convergence of the alternative method (3.1)–(3.2) on a simple 1D model. We chose the problem

$$-u'' + f(u) = g(x) \qquad \text{in } [0, 1] \text{ with } f(u) = u^3, \tag{5.1}$$

and the homogeneous Neumann boundary conditions at $x = 0$ and $x = 1$. Function $g(x)$ was chosen so that the exact solution was $u(x) = \cos(\pi x)$. For this solution the projection Pu was identical to zero on $[0, 1]$. For any function u such that $v = Pu = 0$ one is able to solve (3.1) alone by setting $v_n = 0$, for all $n = 1, 2, \ldots$ (if Pu is nonzero then (3.1) alone converges to a wrong solution). This we will call Method 1 and we shall use it for comparison purposes only. Our aim then became to design a method such that the convergence rate of the entire iteration (3.1)–(3.2) was close to that of Method 1. Iteration (3.1)–(3.2) in which v is computed as according to (4.1) is called Method 2, and iteration (3.1)–(3.2) in which v in (4.2) is computed with the help of the nonlinear solver ZEROIN is called Method 3.

In Table I we have collected results for these three methods using as initial guess random numbers uniformly distributed in $[0, 1]$. The iterations were terminated when the l_2-norm of the residuals had dropped below $1e\text{-}6$. One should also note that Eq. (3.1) is singular and its solution is not unique. In all experiments presented in this section we chose that solution w_n which had a zero projection, $Pw_n = 0$, and thus belongs to the space S_1. As the results in Table I indicate, Method 2 is very inefficient whereas the rate of convergence for Method 3 is almost identical with that of Method 1. One can conclude that in this implementation the bifurcation equation (3.2) does not influence the convergence properties of the auxiliary equation (3.1).

TABLE I

Dependence of the Rate of Convergence on the Mesh Size $h = 1/n$ for the Three Methods Described in the Text

	Method 1		Method 2		Method 3		All
	k	$\|r\|_2$	k	$\|r\|_2$	k	$\|r\|_2$	$\|e\|_2$
$n = 10$	16	0.68e−6	Diverges		16	0.72e−6	0.12
$n = 20$	12	0.98e−6	60	0.91e−6	12	0.97e−6	0.59e−1
$n = 40$	11	0.62e−6	31	0.99e−6	11	0.66e−6	0.29e−1
$n = 80$	11	0.29e−6	25	0.82e−6	11	0.32e−6	0.14e−1
$n = 160$	10	0.83e−6	23	0.62e−6	10	0.79e−6	0.72e−2

Note. Here k is the number of iterations, the residual $r = g(x) - f(v + w)$, and the error $e = u(\text{exact}) - u(\text{computed})$.

Therefore in all the subsequent experiments presented in this section we have used the scheme described above as Method 3.

In Table Ia we have compared the results for the previous problem where function $g(x)$ was chosen so that the exact solution was A: $u(x) = \cos(\pi x)$, and B: $u(x) = 1 + \cos(\pi x)$, i.e., in the case B the projection Pu was nonzero. As initial guess we used the upper solution $u_2 = 2.0$. This time we required the accuracy in residuals to be 1e-3. The rate of convergence for our scheme is almost identical for both cases. Comparing Table I (Method 3) and Table Ia (case A) one can see clearly that the discretization error—and not the convergence tolerance—is the dominating source of the total error in the solution.

The following experiment was modelled on Steuerwalt's example [21]. One seeks the positive solution of the problem

$$-u''(x) + c1(u(x)^4 - c2^4) = g(x) \qquad \text{in}[0, 1], \tag{5.2}$$

TABLE Ia

Comparison of the Rate of Convergence for Solutions with Zero Projection (Case A) and with Nonzero Projection (Case B)

	A			B		
	k	$\|r\|_2$	$\|e\|_2$	k	$\|r\|_2$	$\|e\|_2$
$n = 10$	8	0.87e−3	0.12	8	0.62e−3	0.18
$n = 20$	7	0.33e−3	0.59e−1	7	0.53e−3	0.90e−1
$n = 40$	6	0.47e−3	0.29e−1	7	0.31e−3	0.44e−1
$n = 80$	6	0.28e−3	0.14e−1	6	0.97e−3	0.22e−1
$n = 160$	6	0.24e−3	0.72e−2	6	0.89e−3	0.11e−1

with homogeneous Neumann boundary conditions at $x = 0$ and $x = 1$. In analogy with [21] the constants were $c1 = 0.2e\text{-}8$ and $c2 = 10$, while the function $g(x)$ was chosen so that the exact solution was $u(x) = 650 + 250\cos(\pi x)$. The lower and upper solutions were taken as $u1 = c2 = 10$ and $u2 = (c2^4 + z/c1)^{1/4} = 0.117e + 4$, where $z = \max \text{abs}(g(x))$ in $[0, 1]$. As initial guess we used the upper solution u_2. This time we required the accuracy in residuals to be $1e\text{-}2$. The results in Tables II and IIa demonstrate the ability of the method to solve the problem fairly accurately (the last column in Table II shows the relative errors) in just a few iterations. The computed solution for $n = 10$ is displayed in Table IIb. It should be mentioned that the consecutive iterates converge in an oscillatory manner and not monotonically.

At this point we turned to 2D problems. As a test problem we selected

$$-Lu + f(u) = g(x) \qquad \text{for} \quad x \in [0, 1; 0, 1], \tag{5.3}$$

where $f(u) = u^3$. L is the 2D Laplacian, with homogeneous Neumann boundary conditions on the sides of the unit rectangle.

Functions $g(x, y)$ were chosen so that the exact solutions were $u(x, y) = (\cos(\pi x)\cos(\pi y) + 1)\,\text{Amp}$, where the amplitude Amp is a constant 1, 2, and 5. As an initial guess we used the upper solution $u_2 = (\max(\text{abs}(g(x, y)))^{1/3}$ equal to 3.02 for Amp = 1, 4.69 for Amp = 2, and 10.32 for Amp = 5 (in all the cases the lower solution was $u_1 = 0.0$). The required accuracy in residuals was $1e - 3$. In our experiments tabulated below (see Table III) we have used the HWSCRT fast Poisson solver from the program package FISHPACK originated by Swarztrauber and Sweet, see [22]. In the experiments with Amp = 2 and Amp = 5, the iterations initially slowly oscillate towards the correct solution without reaching the prescribed accuracy, and then diverge sharply. Presumably the nature of this divergence is numerical, although we did not further investigate this problem.

The rate of convergence of the Picard iterations can be improved by the operator displacement procedure as formulated by Keller [9] for ODEs, and recently improved by Pennline [15]. It should be noted that the solution method employed in [9, 15] has been used for 1D problems only (knowledge of the Green's functions is required) and is entirely different from our own. In this approach one considers an equivalent form of the problem (5.3), namely,

$$-Lu + cu + f(u) = cu + g(x), \tag{5.4}$$

where the recommended displacement c (a positive constant) is chosen as

(K) $c = \max(df/du)$—according to [9], and

(P) $c = (\max(df/du) + \min(df/du))/2$—according to [15].

For our 2D test example with $f(u) = u^3$ and $u(x, y) = \text{Amp}(1 + \cos \pi x \cos \pi y)$ the values of these displacement parameters based on the known solution were: $c = 12$ and $c = 6$ for Amp = 1, $c = 48$ and $c = 24$ for Amp = 2, and $c = 300$ and $c = 150$ for Amp = 5. The displacements based on the upper and lower solutions and computed

TABLE II

Rate of Convergence for the Simplified Steuerwalt Example

	k	$\|r\|_2$	$\|e\|_2$	$\|e\|/\|u\|$
$n = 10$	9	0.23e−2	0.41e+2	0.62e−1
$n = 20$	8	0.49e−2	0.19e+2	0.29e−1
$n = 40$	8	0.35e−2	0.95e+1	0.14e−1
$n = 80$	8	0.30e−2	0.47e+1	0.70e−2
$n = 160$	8	0.28e−2	0.23e+1	0.35e−2

Note. Here k is the number of iterations, r the residuals, and e the errors.

TABLE IIa

Consecutive Iterations for $n = 10$

k	$\|r\|_2$	$\|e\|_2$	$\|e\|/\|u\|$
1	0.66e+3	0.12e+3	0.19
2	0.94e+2	0.29e+2	0.43e−1
3	0.15e+2	0.44e+2	0.66e−1
4	0.36e+1	0.40e+2	0.61e−1
5	0.83	0.41e+2	0.62e−1
6	0.19	0.41e+2	0.62e−1
7	0.44e−1	0.41e+2	0.62e−1
8	0.10e−1	0.41e+2	0.62e−1
9	0.23e−2	0.41e+2	0.62e−1
10	0.54e−3	0.41e+2	0.62e−1
11	0.13e−3	0.41e+2	0.62e−1
12	0.28e−4	0.41e+2	0.62e−1
13	0.99e−5	0.41e+2	0.62e−1

TABLE IIb

The Computed Solution $u(x)$ for $n = 10$

i	$u(i)$
0	929.528
1	906.589
2	861.197
3	796.219
4	716.635
5	629.032
6	540.917
7	459.912
8	392.950
9	345.549
10	321.248

TABLE III

Rate of Convergence of a 2D Problem with Amp = 1 on the Unit Square with the Uniform Mesh Size $h = 1/n$ in Both Coordinate Directions

	k	$\|r\|_2$	$\|e\|_2$
$n = m = 10$	1	0.113e+1	0.133
	2	0.148e	0.160e−1
	3	0.246e−1	0.916e−2
	4	0.529e−2	0.434e−2
	5	0.116e−2	0.530e−2
	6	0.254e−3	0.508e−2
$n = m = 20$	1	0.111e+1	0.123
	2	0.141	0.185e−1
	3	0.239e−1	0.544e−2
	4	0.531e−2	0.621e−3
	5	0.120e−2	0.143e−2
	6	0.270e−3	0.119e−2
$n = m = 40$	1	0.111e+1	0.120
	2	0.138	0.190e−1
	3	0.237e−1	0.463e−2
	4	0.536e−2	0.751e−3
	5	0.123e−e	0.516e−3
	6	0.283e−3	0.259e−3

Note. Here k is the number of iterations, r the residuals, and e the errors.

according to [15] were: $c = 13.7$ for Amp = 1, $c = 33.0$ for Amp = 2, and $c = 160.0$ for Amp = 5 (in addition some ad hoc values of the displacement were also used). A series of experiments similar to those in Table III were again run with the fast solver HWSCRT (the expenses for solving the Poisson and Helmholtz equations with constant coefficients are the same) but with different constants c, and the results are tabulated below. As initial guess we used the same upper solutions as above. In Tables IIIa, b, and c different values of the displacement c were used to improve the rate of convergence. Here k is the number of iterations, e the errors, and t the CPU time (in seconds) on the DEC 10 computer. The required accuracy in residuals was $1e-3$.

As expected the rate of convergence of the iterations slows down considerably when the amplitude of the solution is increased from 1 to 2 to 5. The results seem to indicate that even for our scheme the optimum value of the displacement parameter lies in the vicinity of that recommended by Pennline. It is important to note that the optimum is not sharp for $c > c$-optimal and thus even large errors in the overestimation of c do not produce damaging effects on the rate of convergence.

At this point we decided to make comparison tests using directly the scheme with the displacement of the operator, without applying the alternative approch (3.1)–(3.2), although utilizing our fast solver technique. The results for the problem

TABLE IIIa

Convergence Speed-Up for Amp = 1

$n = m$	$c = 0.0$	$c = 3.0$	$c = 6.0$	$c = 12.0$	$c = 13.7$	t (for $c = 6$)
10	6	4	5	7	7	0.64
20	6	4	5	7	7	2.00
40	6	4	5	7	7	7.72

TABLE IIIb

Convergence Speed-Up for Amp = 2

$n = m$	$c = 6.0$	$c = 12.0$	$c = 24.0$	$c = 33.0$	$c = 48.0$	$\|e\|_2$	t ($c = 24$)
10	14	8	9	12	14	7.9e−3	1.04
20	14	8	9	12	14	1.4e−3	3.49
40	14	8	9	12	14	0.25e−3	13.4

TABLE IIIc

Convergence Speed-Up for Amp = 5

$n = m$	$c = 100$	$c = 125$	$c = 150$	$c = 160$	$c = 300$	$\|e\|_2$	t ($c = 150$)
10	40	31	36	38	60	4.9e−3	4.75
20	[a]	29	34	35	56	0.94e−3	13.7
40	[a]	28	33	34	55	4.2e−3[b]	50.6

[a] The solution starts oscillating without reaching the accuracy in residuals.

[b] Here the discretization error is smaller than the error caused by the convergence tolerance; after 42 iterations the 2-norm of the error dropped to 0.3e−3.

(5.4) with the exact solution $u(x, y) = 5(\cos \pi x \cos \pi y + 1)$, initial guess (upper solution) = 10.32, $n = m = 10$, and the required accuracy in residuals $1e - 3$ are tabulated below, see Table IV. The comparison of the results in Tables IIIc and IV shows that the differences in the rate of convergence are rather small. On the other hand, although the computational cost of one iteration here is some 25% smaller than that of Table IIIc, the number of iterations as a function of the parameter c has a somewhat sharper minimum (the second divided difference of the number of iterations as a function of the displacement c is: 0.062 at $c = 150$ for Table IIIc, and 0.084 at $c = 125$ for Table IV). We additionally performed experiments with a fixed value of the displacement ($c = 150$) but changed our initial guess (upper solution u_2). For the scheme without the alternative problem approach, the number of iterations increased from 29 (for $u_2 = 10.32$) to 47 (for $u_2 = 15.48$), and the iterations diverged sharply after only three steps for $u_2 = 20.64$. We must draw the conclusion that the

TABLE IV

Rate of Convergence for the Iterations without the Alternative Problems Technique

$c=$	100	125	150	160	175	300	$t\ (c=150)$
$k=$	Diverges	50	29	33	37	59	2.95

Note. Here k is the number of iterations.

efficiency of this scheme is substantially affected by an initial guess that is too far off or by an inaccurate estimation of the derivative df/du (all we know is the upper limit of the solution and not the solution itself). On the other hand, our scheme based on the alternative problem is completely unaffected by the choice of initial guess (the rate of convergence remains exactly the same). This can easily be explained by observing that the projection operator puts all the residuals into the subspace of the kernel right away from the start.

The final series of experiments were performed for the problem

$$-Lu + cu + f(u) = cu + g(x) \qquad \text{in } D, \tag{5.5}$$

where L is the Laplacian, $f(u) = u^3$, with Neumann conditions at the boundary of the arbitrary bounded region D. At first the tests were carried out on a circular region D with the center at the origin and radius $d = 0.375$. In the previous example, on the unit square we chose $g(x, y)$ so that the exact solution was $u(x, y) = \cos(\pi x)\cos(\pi y) + 1$. An analogous solution in our circular region would be $u(x, y) = \cos(\pi z) + 1$, where $z = r/d$, $r^2 = x^2 + y^2$. Unfortunately in this case the compatibility condition (2.1) is not satisfied, and as a consequence the iterates (3.1)–(3.2) converge to a wrong solution. Therefore we chose the exact solution to be $u(x, y) = x + y + 1$, and computed $g(x, y)$ correspondingly. The compatibility condition is now satisfied although we now longer have the homogeneous Neumann boundary conditions. In fact at ∂D we have $u = (u - 1)/d$. This fact in no significant way influences the results, and thus our scheme generalizes to the nonhomogeneous Neumann boundary conditions. It should also be noted that for $u = x + y + 1$ the discretization error is zero, which explains the high accuracy in the solution. The experiments were run with the capacitance matrix program CMMEXP (see Proskurowski [18]), the parameters c were 0 and 1.53, the required accuracy in the residuals was $1e - 6$, and as initial guess we used the upper solution $u_2 = 1.53$ (the lower solution was $u_1 = 0{,}0$).

The rate of convergence is very similar for problems (5.3) and (5.5), as can be seen by comparing Tables III and V. In the absence of the discretization error, the limiting factor for the accuracy is (for $c > 0$) the machine tolerance. These results indicate that the number of iterations does not change for the mesh sizes under consideration. Moreover, an increase of the number of unknowns from 109 to 437 to 1789, i.e., by a

TABLE V

Rate of Convergence for the Problem (5.5) in the Circular Region

	$c = 0$		$c = 1.53$	
k	$\|r\|_2$	$\|e\|_2$	$\|r\|_2$	$\|e\|_2$
1	0.108e+1	0.320e−1	0.108e+1	0.137e−1
2	0.349e−1	0.297e−2	0.144e−1	0.667e−3
3	0.338e−2	0.423e−3	0.706e−2	0.385e−4
4	0.355e−3	0.105e−3	0.404e−4	0.220e−5
5	0.373e−4	0.130e−3	0.234e−5	0.149e−6
6	0.394e−5	0.127e−3	0.130e−6	0.638e−7
7	0.388e−6	0.128e−3	—	—

Note. Here the mesh size is $h = 1/n$, $n = 16$, k is the number of iterations, r are the residuals and e are the errors.

TABLE Va

Rate of Convergence with Different Mesh Sizes

		$c = 0$		$c = 1.53$		
	N	k	$\|e\|_2$	k	$\|e\|_2$	t ($c = 1.5$)
$n = m = 16$	109	4	0.105e−3	3	0.385e−4	1.28
$n = m = 32$	437	4	0.245e−3	3	0.480e−4	5.45
$n = m = 64$	1789	—	—	3	0.591e−4	26.34

Note. Here N is the number of mesh points inside the region, k the number of iterations to reach the 1e−3 accuracy in the residuals r, e is the error, and t is the CPU-time in seconds on the DEC 10 computer.

factor of 4.00 and 4.09, results in the increase of the CPU-time of the same order: from 1.28 to 5.45 to 26.34, i.e., by a factor of 4.26 and 4.83.

From the results in Tables III and Va one can conclude that for the mesh sizes considered and the tolerance in residuals of the order $1e - 3$, the total computational cost per unknown (or mesh point) on the DEC 10 computer is:

3 to 4 msec for problems in rectangular regions,
12 to 15 msec for problems in circular regions.

REFERENCES

1. R. P. BRENT, "Algorithms for Minimization without Derivatives," Prentice–Hall, Englewood Cliffs, N.J., 1973.
2. B. L. BUZBEE, F. W. DORR, J. A. GEORGE, G. H. GOLUB, *SIAM J. Numer. Anal.* **8** (1971), 722–736.

3. L. CESARI, *in* "Nonlinear Functional Analysis and Differential Equations" (Cesari, Kannan, and Schuur, Eds.), pp. 1–197, Dekker, New York, 1976
4. L. CESARI AND R. KANNAN, *Proc. Amer. Math. Soc.* **63** (1977), 221–225.
5. G. E. FORSYTHE, M. A. MALCOLM, AND C. B. MOLER, "Computer Methods for Mathematical Computations," Prentice-Hall, Englewood Cliffs, N. J., 1977.
6. R. W. HOCKNEY, *J. Appl. Phys.* **39** (1968), 4166–4170.
7. R. KANNAN AND J. LOCKER, *J. Differential Equations* **26** (1977), 1–8.
8. R. KANNAN AND K. J. MOREL, to appear.
9. H. B. KELLER, "Numerical Methods for Two-Point BVP," Ginn (Blaisdell), Boston, 1968.
10. H. B. KELLER, *Arch. Rat. Mech. Anal.* **35** (1969), 363–381.
11. H. B. KELLER AND E. REISS, *Comm. Pure Appl. Math.* **9** (1958), 273–292.
12. D. KU Ph.D. thesis, Univ. of Michigan, 1976.
13. W. E. OLMSTEAD, *J. Math. Mech.* **15** (1966), 899–908.
14. J. M. ORTEGA AND W. C. RHEIBOLDT, "Iterative Solution of Nonlinear Equations in Several Variables," p. 452, Academic Press, New York, 1970.
15. J. A. PENNLINE, *Math. Comp.* **37** (1981), 127–134.
16. W. PROSKUROWSKI, *ACM Trans. Math. Software* **5** (1979), 36–49.
17. W. PROSKUROWSKI, *in* "Advances in Computer Methods for PDEs–IV" (R. Vichnevetsky and R. S. Stepleman, Eds.), pp. 274–280, IMACS Publ.
18. W. PROSKUROWSKI, *ACM Trans. Math. Software* **9** (1983), 117–124.
19. W. PROSKUROWSKI AND O. WIDLUND, *Math. Comp.* **30** (1976), 433–468.
20. M. RAY, Ph.D. thesis, Univ. of Texas at Arlington, 1980.
21. M. STEUERWALT, *SIAM J. Numer. Anal.* **16** (1979), 402–420.
22. P. SWARZTRAUBER AND R. SWEET, Algorithm 541, *ACM Trans. Math. Software* **5** (1979), 352–364.