NUMER. FUNCT. ANAL. AND OPTIMIZ., 16(1&2), 219–231 (1995)

On Computation of Solution Curves for Semilinear Elliptic Problems

Philip Korman
Institute for Dynamics and
Department of Mathematical Sciences
University of Cincinnati
Cincinnati, Ohio 45221-0025
and
Tiancheng Ouyang
Department of Mathematics
Brigham Young University
Provo, Utah 84602

Abstract

We consider computation of solution curves for semilinear elliptic equations. In case solution is stable, we present an algorithm with monotone convergence, which is a considerable improvement of the corresponding schemes in [4] and [5]. For the unstable solutions, we show how to construct a fourth-order evolution equation, for which the same solution will be stable.

AMS Subject classification. 65L10, 65N20.

1 Introduction

We are interested in computation of the solution $u = u(x)$ for two point boundary value problems depending on a parameter λ

$$u'' + \lambda f(x,u) = 0 \text{ on } (a,b), \ u(a) = u(b) = 0, \tag{1.1}$$

and the corresponding Dirichlet problem

$$\Delta u + \lambda f(x,u) = 0 \text{ for } x \in \Omega, \ u = 0 \text{ for } x \in \partial\Omega. \tag{1.2}$$

(Here $\Omega \subset R^n$ is a smooth domain, and typically $n = 2$.)

For simplicity, we present our results only for the problem (1.1), although all our results hold for (1.2) also, and we report on computations in both one and two dimensions. In section 2 we consider computation of stable solutions. (Stability means that solution of corresponding evolution equation tends to $u(x)$). It is well-known that if one solves the corresponding evolution equation with initial data being subsolution (supersolution) of (1.1) then its solution will tend increasing (decreasing) to the steady-state, which is a solution of (1.1). This fact was used in C.U. Huy, P.J. McKenna and W. Walter [4] to develop both implicit and explicit schemes with monotone convergence. In P. Korman [5] and G. Choudury and P. Korman [3] it was shown that an explicit scheme with monotone convergence can be developed even for fully nonlinear equations. In V. Barbu and P. Korman [1] a similar scheme was used for obstacle problems with nonlinear forcing term. In section 2 we introduce a modification of the explicit scheme, similar to Gauss-Seidel modification of Jacoby iteration. We prove that this modification preserves monotonicity. (The advantages of monotone schemes are discussed e.g. in [1]). Also, from our experiments it appears that the resulting scheme is much more stable than the explicit scheme, i.e. one can take much larger time steps. The new scheme is easier to program than the explicit scheme in [4], while it appears to be faster (CPU time) than the implicit scheme in [4]. (The last statement, of course, depends on the implementation.)

In section 3 we take up computation of unstable solutions of (1.1), i.e. the steady-states which cannot be reached in practical computations by the solutions of the corresponding evolution equations (the stable manifold is typically one-dimensional). Variational techniques, based on the mountain pass lemma can be typically used to compute unstable solutions. This approach was taken in [2] and independently in [6], where a different implementation, not requiring recomputation of the entire path, is presented. Here we present a more systematic way to compute the unstable solutions. We introduce a fourth-order evolution problem for which solution of (1.1) is now a stable steady-state, so that the unstable solution of (1.1) can be computed by using (say) an explicit scheme for the fourth-order equation. Here we present an heuristic justification of the scheme, a more rigorous analysis will be given by one of the authors in a paper in preparation. Our scheme is very easy to program. While convergence for fourth-order equations can be slow, one has to "pay the price" only once: once solution of (1.1) is computed for a particular $\lambda = \lambda_0$, we use very efficient continuation techniques to compute the solutions for other λ. Combination of our techniques in sections 2 and 3 allow efficient computation of all solutions (including multiple solutions for fixed λ) for any problem of type (1.1) or (1.2), for which one understands the structure of solutions.

2 Computation of the stable solutions

In this section we shall consider computations of stable solutions for the problems (1.1) and (1.2). We recall that solution $u(x)$ of (1.1) is called stable if the principle eigenvalue of the linearized problem

$$-v'' - \lambda f_u(x,u)v = \lambda v \text{ on } (a,b), \; v(a) = v(b) = 0 \tag{2.1}$$

is positive. This means that $u(x)$ is a stable steady-state solution for the corresponding parabolic problem (here $u = u(x,t)$)

$$u_t - u_{xx} = f(x,u) \;\; a < x < b, \;\; t > 0 \;\; u(a,t) = u(b,t) = 0. \tag{2.2}$$

On the other hand, the solution of (1.1) is unstable if the principal eigenvalue of (2.1) is negative, or it is an unstable steady-state solution of (2.2).

It turns out that the stable solutions of (1.1) are generally easy to compute: it takes only minimal programming effort, and the result is obtained practically instantaneously on a 486 PC (using 20 mesh points and achieving stabilization of 6 decimal digits). We describe next such an algorithm, with convergence being monotone, proposed by C.U. Huy, P.J. McKenna and W. Walter [4], and independently by P. Korman [5], where more general problems were considered (see also [3] where fully nonlinear equations with general boundary conditions are treated). We divide the interval $[a,b]$ into N equal parts of length $h = \frac{b-a}{N}$ each, denote $x_0 = a$, $x_k = x_0 + kh$ for $k = 1,2,\ldots,N$, $u_k = u(x_k)$, and replace (1.1) by its finite difference version

$$\begin{aligned} -\frac{u_{k+1} - 2u_k + u_{k-1}}{h^2} &= \lambda f(x_k, u_k), \;\; k = 1,\ldots,N-1, \\ u_0 &= u_N = 0. \end{aligned} \tag{2.3}$$

We recall that a grid function $\varphi_k = \varphi$ is called a supersolution of (2.3) if

$$\begin{aligned} -\frac{\varphi_{k+1} - 2\varphi_k + \varphi_{k-1}}{h^2} &\geq \lambda f(x_k, \varphi_k), \;\; k = 1,\ldots,N-1, \\ \varphi_0 &\geq 0, \;\; \varphi_n \geq 0. \end{aligned} \tag{2.4}$$

A subsolution ψ_k is defined by reversing all the inequalities in (2.4).

To solve (2.3), one sets up iterations by discretizing (2.2), which in the case of explicit scheme is done as follows ($n = 1,2,\ldots$).

$$\begin{aligned} \frac{u_k^{n+1} - u_k^n}{\tau} &= \frac{u_{k+1}^n - 2u_k^n + u_{k-1}^n}{h^2} + \lambda f(x_k, u_k^n), \;\; k = 1,\ldots,N-1, \\ u_0^{n+1} &= u_N^{n+1} = 0. \end{aligned} \tag{2.5}$$

This allows us to compute u_k^{n+1} given u_k^n. It was shown in [4] and [5] that if one starts the iterations with $u_k^0 = \varphi_k$, then for sufficiently small τ the iterates u_k^n are decreasing in n for all k, and so if the u_k^n are bounded below they must converge to a solution of (2.3). One way convergence can be ensured, is when one assumes existence of both super- and sub-solutions, and $\psi_k \le \varphi_k$ for all k (see [4], [5] for details).

Next we present an improvement of the scheme (2.5), which is of "Gauss-Seidel" type. We replace (2.5) by

$$\begin{aligned} \frac{u_k^{n+1} - u_k^n}{\tau} &= \frac{u_{k+1}^n - 2u_k^n + u_{k-1}^{n+1}}{h^2} + \lambda f(x_k, u_k^n), \quad k = 1, \ldots, N-1, \\ u_0^{n+1} &= u_N^{n+1} = 0, \end{aligned} \tag{2.6}$$

i.e., we immediately use the computed values of u_k^{n+1}.

It is natural to expect that the scheme (2.6) is faster than (2.5). It also turns out to be more stable (one can use larger τ), and, as the following theorem shows, it preserves the monotonicity property of (2.5).

Theorem 2.1 *Assume that the problem (2.3) possesses a subsolution ψ_k and a supersolution φ_k with $\psi_k \le \varphi_k$ for all $k = 1, \ldots, N-1$. Assume that the function $f(x,u)$ is continuous in x and Lipschitz continuous in u uniformly in x. Then the problem (2.3) has a solution, which can be approximated using (2.6). Namely, letting $u_k^0 = \varphi_k$ ($u_k^0 = \psi_k$) and choosing τ sufficiently small, we get a decreasing (increasing) in n sequence u_k^n converging to a solution of (2.3).*

Proof: Letting $u_k^0 = \varphi_k$ and denoting $U^n = (u_1^n, \ldots, u_{N-1}^n)$, we rewrite (2.6) in matrix form as

$$\left(I - \frac{\tau}{h^2} A\right) U^{n+1} = \frac{\tau}{h^2} B U^n + D(U^n). \tag{2.7}$$

Here $A = (a_{ij})$ is an $(n-1) \times (n-1)$ matrix such that $a_{ij} = 1$ when $i = j+1$, $j = 1, \ldots, N-2$, $a_{ij} = 0$ otherwise; $B = (b_{ij})$ is an $(n-1) \times (n-1)$ matrix such that $b_{ij} = 1$ when $j = i+1$ and $i = 1, \ldots, N-2$ and D is a diagonal matrix with $d_{kk} = \left(1 - \frac{2\tau}{h^2}\right) u_k^n + \tau\lambda f(x_k, u_k^n)$. Denote $W^n = U^{n+1} - U^n$, $n = 0, 1, \ldots$. We claim that $W^0 \le 0$, i.e. $u_k^1 \le \varphi_k$. This is proved by induction in k. Indeed, by the definition of the supersolution

$$\frac{u_1^1 - \varphi_1}{\tau} = \frac{\varphi_2 - 2\varphi_1}{h^2} + \lambda f(x_1, \varphi_1) \le 0,$$

i.e., $u_1^1 \le \varphi_1$, and then

$$\frac{u_2^1 - \varphi_2}{\tau} = \frac{\varphi_3 - 2\varphi_2 + u_1^1}{h^2} + \lambda f(x_2, \varphi_2) \le \frac{\varphi_3 - 2\varphi_2 + \varphi_1}{h^2} + \lambda f(x_2, \varphi_2) \le 0,$$

i.e., $u_2^1 < \varphi_2$ and so on.

We now prove by induction that $W^n \le 0$ for all n. Indeed, from (2.6) we obtain (assuming $W^{n-1} \le 0$ and denoting by L the Lipschitz constant of $f(x,u)$)

$$\left(I - \frac{\tau}{h^2}A\right) W^n \le BW^{n-1} + \left(1 - \frac{2\tau}{h^2} - \tau L\right) W^{n-1}. \tag{2.8}$$

Writing $I - \frac{\tau}{h^2}A$ as a product of the elementary matrices, we see that the matrix $\left(I - \frac{\tau}{h^2}A\right)^{-1}$ is a lower triangular one, and all of its entries on and below the diagonal are positive. If we now choose τ so small that $1 - \frac{2\tau}{h^2} - \tau L > 0$, then $W^n \le 0$.

If we denote by v_k^n the iterates (2.6) starting with $v_k^0 = \psi_k$, then repeating the above arguments two more times, we show that for all $k = 1, \dots, N-1$,

$$\psi_k \le v_k^1 \le v_k^2 \le \cdots \le u_k^2 \le u_k^1 \le \varphi_k,$$

and the proof follows.

Our numerical experiments in one and two dimensions have shown the scheme (2.6) to be a substantial improvement over (2.5). The main reason is that one can choose larger time step τ for (2.6). (It is also faster for equal τ).

Example 1. We computed the positive solution of

$$u'' + u(5-u) = 0 \text{ on } (0,1), \ u(0) = u(2) = 0.$$

We took $N = 20$ (i.e. $h = 0.1$), $\tau = 0.009$, $u_k^0 = 6$ for all k, and after just 80 time steps we obtained the solution with six decimal digits stabilized (maximum value, $u(1) \simeq 2.948380$) using the scheme (2.6). By comparison, when the scheme (2.5) was used with the same N and u_k^0, the largest value of τ we could take was $\tau = 0.0026$ ($\tau = 0.0027$ led to overflow), and it took around 1300 time steps to achieve the same accuracy.

Example 2. We consider the problem (in two dimensions)

$$-\Delta u = -(u_{xx} + u_{yy}) = \lambda e^u \text{ in } \Omega, \ u = 0 \text{ on } \partial\Omega, \tag{2.9}$$

which is prominent in applications, particularly in combustion theory, see e.g. [9]. This equation is often referred to as Bratu or Gelfand equation. It is known that the branch of positive stable solutions (emanating from $\lambda = 0$, $u = 0$) bends back at a critical $\lambda_0 = \lambda_0(\Omega) > 0$. We took Ω to be the unit square $\Omega = (0,1) \times (0,1)$ and $\lambda = 6.5$, which is close to the critical λ_0. We took uniform square mesh of step size $h = 0.1$ (i.e. the solution was computed at $9^2 = 81$ interior points). We used the scheme (2.6) (with $u_{k,p} = u(kh, ph)$),

$$(2.10)\quad \frac{u_{k,p}^{n+1} - u_{k,p}^{n}}{\tau} = \frac{1}{h^2}\left[u_{k+1,p}^{n} + u_{k,p+1}^{n} + u_{k-1,p}^{n+1} + u_{k,p-1}^{n+1} - 4u_{k,p}^{n}\right] + \lambda \exp(u_{k,p}^{n})$$

for $k, p = 1, \ldots, 9$. We took $\tau = 0.0045$ and after just 80 time steps we obtained five decimal digits stabilized (the maximum value was $u(0.5, 0.5) = 1.00623$). The maximum over all k and p of the absolute values of the right hand sides in (2.10) was 0.0000000 (we shall refer to this below as defmax = 0.0000000). When we tried the scheme (2.5) for the same problem, the largest time step we could take was $\tau = 0.0013$, and the convergence was much slower.

3 Computation of unstable solution branches

For the unstable solutions of (1.1) or (1.2) neither the approach above nor the well-known method of monotone iterations can possibly work. In Korman [6] a partially interactive algorithm for computing unstable solutions was presented, see also [2]. Here we present a more direct and systematic approach.

Denote by $U = (u_1, \ldots, u_{N-1})$ and by $-\Delta_h u_k$ the left hand side of (2.3). Then solving (2.3) is equivalent to finding the zero minimum points of the nonnegative function

$$(3.1)\qquad F(U) = \sum_{k=1}^{N-1} (\Delta_h u_k + \lambda f(x_k, u_k))^2.$$

Similarly, solving (1.1) can be seen as looking for the minimum of the functional

$$(3.2)\qquad F(u) = \int_a^b (u'' + \lambda f(x,u))^2 dx.$$

In developing our algorithm we shall use the discrete and continuous versions of the equation (1.1) and of the functional (3.2) interchangeably, deferring rigorous justification of the method to a paper in preparation.

Assume that $u(x,t)$ is a solution of the following fourth-order problem, resembling the Cahn-Hilliard equation

$$(3.3)\qquad u_t = -(u_{xx} + \lambda f(x,u))_{xx} - \lambda f_u(x,u)(u_{xx} + \lambda f(x,u)) \quad \text{for } a < x < b,\ t > 0$$

$$(3.4)\qquad u(a,t) = u(b,t) = 0$$

$$u_{xx}(a,t) + \lambda f(a, u(a,t)) = u_{xx}(b,t) + \lambda f(b, u(b,t)) = 0 \tag{3.5}$$

$$u(x,0) = u_0(x). \tag{3.6}$$

(For the PDE version just replace double differentiation in x by Laplacian.) Condition (3.5) represents the equation (1.1) written at $x = a$ and $x = b$, which is a common computational device. Notice that this condition is linear in view of (3.4). Along the trajectories of (3.3-3.6) we compute, integrating by parts twice,

$$\begin{aligned} \frac{d}{dt}F(u) &= \int_a^b 2(u_{xx} + \lambda f(x,u))(u_{xxt} + \lambda f_u u_t)dx \\ &= -2\int_a^b u_t^2 dt < 0. \end{aligned}$$

Now assume that the problem (1.1) has a finite number of solutions, and that they are non-degenerate, i.e., the only solution of the linearized problem

$$-v'' - \lambda f_u(x,u)v = 0 \text{ on } (a,b), \ v(a) = v(b) = 0 \tag{3.7}$$

is zero. If the solution of (3.3-3.6) stays in a compact region, it can be expected to converge to a steady-state solution of (3.3). Comparing (3.3) and (3.5) with (3.7), we conclude that the steady-state solutions of (3.3) are solutions of (1.1) (denoting $v(x) = u_{xx} + \lambda f(x,u)$, we see that v satisfies (3.7), and hence $v = 0$). Finally, since each point of minimum of $F(u)$ has its own basin of attraction for the flow induced by (3.3-3.6), it follows that we can expect that if $u_0(x)$ is chosen close to the solution $\overline{u}(x)$ of (1.1) (stable or unstable), then the solution $u(x,t)$ of (3.3-3.6) will converge to $\overline{u}(x)$ as $t \to \infty$. (Actually, the basins of attraction are usually quite large.) We explain our approach on the following numerical examples.

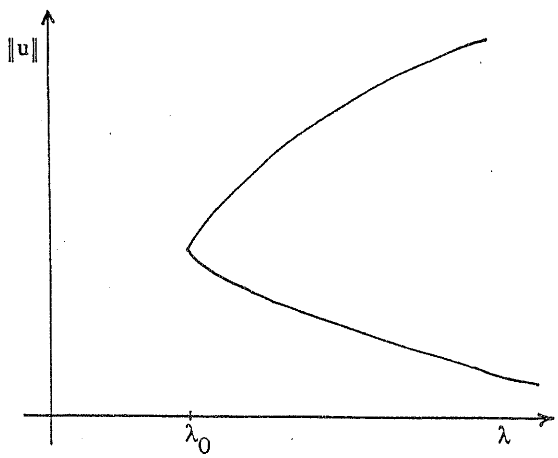
Example 1.

$$-u'' = \lambda u^2 \text{ on } (0,1), \ u(0) = u(1) = 0. \tag{3.8}$$

The problem (3.8) has exactly two solutions: a positive one, which is unstable, and a stable trivial solution (as can be seen by a phase plane analysis). Hence the functional $F(u)$ has only two minimums, and the basin of attraction of each one (in particular of the positive solution) can be expected to be large. Starting with $u_0(x) = 6$ for $0 < x < 1$, and using an explicit scheme for (3.3-3.6) we obtained solution of (3.8) for $\lambda = 1, 10, 15$. We took $h = 0.1$, $\tau = 0.000009$. Solutions are given in Table 1. By *defmax* we denote the $\max_k |\Delta_h u_k + \lambda f(x_k, u_k)|$, where $\Delta_h u_k$ is the left hand side of (2.3).

TABLE 1

Lambda = 1		Lambda = 10		Lambda = 15	
u(0.0) =	0.00000000	u(0.0) =	0.00000000	u(0.0) =	0.00000000
u(0.1) =	3.24789176	u(0.1) =	0.32478918	u(0.1) =	0.21652612
u(0.2) =	6.39029552	u(0.2) =	0.63902955	u(0.2) =	0.42601970
u(0.3) =	9.12434050	u(0.3) =	0.91243405	u(0.3) =	0.60828937
u(0.4) =	11.02584959	u(0.4) =	1.10258496	u(0.4) =	0.73505664
u(0.5) =	11.71166508	u(0.5) =	1.17116651	u(0.5) =	0.78077767
u(0.6) =	11.02584959	u(0.6) =	1.10258496	u(0.6) =	0.73505664
u(0.7) =	9.12434050	u(0.7) =	0.91243405	u(0.7) =	0.60828937
u(0.8) =	6.39029551	u(0.8) =	0.63902955	u(0.8) =	0.42601970
u(0.9) =	3.24789176	u(0.9) =	0.32478918	u(0.9) =	0.21652612
u(1.0) =	0.00000000	u(1.0) =	0.00000000	u(1.0) =	0.00000000
defmax =	0.000000	defmax =	0.0000000	defmax =	0.0000000



Pic. 1.

Example 2.

$$-u'' = \lambda u^2(1-u), \text{ on } (0,1), \; u(0) = u(1) = 0. \tag{3.9}$$

According to [8] there exists a critical $\lambda_0 > 0$, such the problem (3.9) has no solutions for $\lambda < \lambda_0$, exactly one solution for $\lambda = \lambda_0$, and for $\lambda > \lambda_0$ the problem (3.9) has exactly two positive solutions, $0 < u_1(x) < u_2(x)$. The upper one, $u_2(x)$, is stable, while the lower one $u_1(x)$ is unstable. Also there is a trivial solution $u = 0$, which is stable. The bifurcation diagram is given in Pic. 1. The stable solution $u_2(x)$ is easily computed using the scheme (2.6) with u_k^0 being a supersolution, e.g., $u_k^0 = 1$, $k = 1, \ldots, N-1$. Next we describe the computation of unstable solutions.

We took $h = 0.1$ and started with $\lambda = 60$. We took $u_k^0 = 0.5$ for all k, $\tau = 0.000009$, and computed $u_1(x)$ using the explicit scheme for (3.3-3.6) for all k. The initial guess $u_k^0 = 0.5$ turned out to be in the basin of attraction of $u_1(x)$, even though it is relatively far from the solution. A systematic way to obtain an approximation to the unstable solution is described in [6]. The convergence was rather slow. However, this computation gave us a point on the branch of unstable solutions. Using the standard predictor-corrector method (with two Newton steps) we were able to trace very quickly the unstable branch of solutions, obtaining very accurate solutions for both increasing and decreasing λ (and the step in λ does not have to be small). For a "predictor" at $\lambda + \Delta\lambda$, we used the formula $u(x, \lambda + \Delta\lambda) \simeq u(x, \lambda) + u_\lambda(x, \lambda)\Delta\lambda$, where u_λ is the solution of the linear problem

TABLE 2

Lambda =	61.0	umax =	0.24315565
Lambda =	62.0	umax =	0.23784542
Lambda =	63.0	umax =	0.23278223
Lambda =	64.0	umax =	0.22794765
Lambda =	65.0	umax =	0.22332517
Lambda =	66.0	umax =	0.21890000
Lambda =	67.0	umax =	0.21465876
Lambda =	68.0	umax =	0.21058939
Lambda =	69.0	umax =	0.20668091
Lambda =	70.0	umax =	0.20292332
Lambda =	71.0	umax =	0.19930751
Lambda =	72.0	umax =	0.19582511
Lambda =	73.0	umax =	0.19246844
Lambda =	74.0	umax =	0.18923046
Lambda =	75.0	umax =	0.18610463
Lambda =	76.0	umax =	0.18308494
Lambda =	77.0	umax =	0.18016582
Lambda =	78.0	umax =	0.17734209
Lambda =	79.0	umax =	0.17460894
Lambda =	80.0	umax =	0.17196191

$$u_\lambda'' + \lambda(2u - 3u^2)u_\lambda = -u^2(1-u) \text{ on } (0,1), \quad u_\lambda(0) = u_\lambda(1) = 0.$$

The critical value of the parameter turned out to be $\lambda_0 \simeq 40.8$. This value can be accurately predicted by the following heuristic argument. Consider $\varphi(\mu) = J(\mu\varphi_1)$, where J is the energy functional of the problem (3.9),

$$J(u) = \int_0^1 \left(\frac{1}{2}u'^2 - \lambda\frac{u^3}{3} + \lambda\frac{u^4}{4} \right) dx,$$

and $\varphi_1 = \sqrt{2}\sin \pi x$, the normalized principal eigenfunction, $\int_0^1 \varphi_1^2 dx = 1$, corresponding to $\lambda_1 = \pi^2$. Then one verifies that for λ small the function $\varphi(\mu)$ has only one critical point, while for λ large $\varphi(\mu)$ has three critical points. The transition occurs at $\lambda = 4\lambda_1 \frac{\int_0^1 \varphi_1^4 dx}{\left(\int_0^1 \varphi_1^3 dx\right)^2} \simeq 41.1$. In Table 2 we give the maximum values of some solutions from the lower branch. For all of these solutions the *defmax* (defined above) was ≤ 0.0000000002.

We remark that the solution curve in Example 2 could also be traced starting from the stable branch, although "turning the corner" would require extra effort. In Example 1, however, the entire curve of solutions is unstable, so that this alternative is not available.

Table 3.

$u_1(x)$

0.01079	0.02147	0.03107	0.03793	0.04045	0.03793	0.03107	0.02147	0.01079
0.02147	0.04312	0.06299	0.07744	0.08278	0.07744	0.06299	0.04312	0.02147
0.03107	0.06299	0.09291	0.11500	0.12322	0.11500	0.09291	0.06299	0.03107
0.03793	0.07744	0.11500	0.14301	0.15349	0.14301	0.11500	0.07744	0.03793
0.04045	0.08278	0.12322	0.15349	0.16483	0.15349	0.12322	0.08278	0.04045
0.03793	0.07744	0.11500	0.14301	0.15349	0.14301	0.11500	0.07744	0.03793
0.03107	0.06299	0.09291	0.11500	0.12322	0.11500	0.09291	0.06299	0.03107
0.02147	0.04312	0.06299	0.07744	0.08278	0.07744	0.06299	0.04312	0.02147
0.01079	0.02147	0.03107	0.03793	0.04045	0.03793	0.03107	0.02147	0.01079

defmax = 0.000000

$u_2(x)$

0.20984	0.38489	0.48727	0.53335	0.54593	0.53335	0.48727	0.38489	0.20984
0.38489	0.66019	0.78738	0.83471	0.84636	0.83471	0.78738	0.66019	0.38489
0.48727	0.78738	0.90372	0.94142	0.94997	0.94142	0.90372	0.78738	0.48727
0.53335	0.83471	0.94142	0.97346	0.98036	0.97346	0.94142	0.83471	0.53335
0.54593	0.84636	0.94997	0.98036	0.98679	0.98036	0.94997	0.84636	0.54593
0.53335	0.83471	0.94142	0.97346	0.98036	0.97346	0.94142	0.83471	0.53335
0.48727	0.78738	0.90372	0.94142	0.94997	0.94142	0.90372	0.78738	0.48727
0.38489	0.66019	0.78738	0.83471	0.84636	0.83471	0.78738	0.66019	0.38489
0.20984	0.38489	0.48727	0.53335	0.54593	0.53335	0.48727	0.38489	0.20984

defmax = 0.000000

Example 3. With $\Delta u = u_{xx} + u_{yy}$, we solved a two-dimensional problem

$$(3.10) \quad -\Delta u = \lambda u^2(1-u) \text{ on } \Omega = (0,1)\times(0,1), \ u = 0 \text{ on } \partial\Omega.$$

The exact multiplicity for (3.10) appears to be an open question, although we believe that the situation here is similar to that of the problem (3.9), i.e. the bifurcation diagram is given in Pic. 1, with an unstable solution $u_1(x,\lambda)$ and stable solution $u_2(x,\lambda)$, and $u_1(x,\lambda) < u_2(x,\lambda)$ for all $\lambda > \lambda_0$ and $x \in \Omega$.

We took uniform square mesh of step size $h = 0.1$. In Table 3 we present both solutions for $\lambda = 50$ at the interior mesh points. The stable solution $u_2(x,\lambda)$ was quickly computed, using the scheme (2.6), and also it could be computed using (3.3-6) with $u_0(x) = 1$. To compute the unstable solution $u_1(x,\lambda)$ we were using (3.3-6) with $u_0(x) = 0.1$. The critical value of the parameter λ turned out to be $\lambda_0 \simeq 21.3$.

References

[1] V. Barbu and P. Korman, *Approximating optimal controls for elliptic obstacle problem by monotone iteration schemes*, Numer. Funct. Anal. and Optimiz., 12 (5 & 6), 429-442 (1991).

[2] Y.S. Choi and P.J. McKenna, *A mountain pass method for the numerical solution of semilinear elliptic problems*, Nonlinear Anal. Theory Methods Appl. 20, 417-437 (1993).

[3] G. Choudury and P. Korman, *On computation of solutions of fully nonlinear elliptic problems*, Journal of Computational and Applied Mathematics, 41, 301-311 (1992).

[4] C.U. Huy, P.J. McKenna and W. Walter, *Finite difference approximations to the Dirichlet problem for elliptic systems*, Numer. Math. 49, 227-237 (1986).

[5] P. Korman, *On computation of solutions of elliptic systems*, Numer. Funct. Anal. and Optimiz. 10 (9 & 10) 977-990 (1989).

[6] P. Korman, *An algorithm for computing unstable solutions of semilinear boundary value problems*. Submitted.

[7] P. Korman and T. Ouyang, *Multiplicity results for two classes of boundary value problems*, To appear in SIAM J. of Math Analysis.

[8] S.-H. Wang and N.D. Kazarinoff, *Bifurcation and stability of positive solutions of two-point boundary value problem*, J. Austral. Math. Soc. (Series A) 52, 334-342 (1992).

[9] F.A. Williams, Combustion Theory, Addison-Wesley, Massachusetts, 1969.

Received: July 28, 1993
Accepted: November 9, 1994