Computing 51, 327–334 (1993)

Computing

An Algorithm for Computing Unstable Solutions of Semilinear Boundary Value Problems

P. Korman, Cincinnati

Received June 14, 1993

Abstract — Zusammenfassung

An Algorithm for Computing Unstable Solutions of Semilinear Boundary Value Problems. We present a partially interactive algorithm for accurate computation of unstable solutions of semilinear Dirichlet boundary value problems.

Key words: Unstable solutions, mountain-pass lemma.

Ein Algorithmus zur Berechnung instabiler Lösungen halblinearer Randwertprobleme. Wir stellen einen teilweise interaktiven Algorithmus zur genauen Berechnung instabiler Lösungen von halblinearen Dirichlet-Randwertproblemen vor.

1. Introduction

We are interested in the numerical computation of the unstable solutions of semilinear boundary value problems ($u = u(x)$)

$$-u'' = f(x,u) \qquad \text{for all } a < x < b, \qquad u(a) = u(b) = 0. \tag{1}$$

We recall that solution is called unstable if the principal eigenvalue of the linearized problem

$$-v'' - f_u(x,u)v = \lambda v \qquad a < x < b, \qquad v(a) = v(b) = 0 \tag{2}$$

is negative (see D. Henry [3]). This means that $u(x)$ is an unstable steady-state for the corresponding parabolic equation (here $u = u(x,t)$)

$$u_t - u_{xx} = f(x,u) \qquad a < x < b, \qquad t > 0; \qquad u(a,t) = u(b,t) = 0. \tag{3}$$

For example, consider an important Emden-Fowler equation

$$-u'' = a(x)u^p \text{ for } a < x < b, \qquad u(a) = u(b) = 0, \qquad a(x) > 0, \qquad p > 0. \tag{4}$$

It occurs in nuclear physics, gas dynamics, astronomy and other fields, see e.g. C. D. Luning and W. L. Perry [6] and the references therein. The positive solution of (4) is necessarily unstable. Indeed, if $u(x) > 0$ is a solution of (4), then it follows immediately that $\gamma u(x)$ is a subsolution of (4) for any constant $\gamma > 1$, and it is a supersolution if $0 < \gamma < 1$. This is another equivalent definition of instability (see D. H. Sattinger [8], where one can also find the definition of super- and subsolutions).

Computation of an unstable solution is a nontrivial task, as for example the well-known monotone iteration method cannot possibly work (see [8]). Also, one cannot obtain the solution of (1) as a limit when $t \to \infty$ of the solution of (2), as was done for stable solutions in [4, 5, 2]. Continuation algorithms can be used, see e.g., E. L. Allgower, C. S. Chien and K. Georg [1] or M. Smiley [8], however such algorithms appear to be not easy to apply, particularly getting onto a branch of solutions might be hard. For Emden-Fowler equations an ingenious algorithm with monotone convergence is proposed in Luning and Perry [6], however their technique does not extend to other equations. One can use shooting method for (1), however it has its limitations, and does not extend to PDE.

We present here a partially interactive algorithm based on the fact that unstable solutions of (1) are typically saddle points of the corresponding "energy" functional defined on $H_0^1(a,b)$

$$J(u) = \int_a^b \left(\frac{1}{2}u'^2 - F(x,u)\right)dx, \qquad F(x,u) \equiv \int_0^u f(x,z)dz. \tag{5}$$

We shall assume that

$$f(x,0) = 0 \qquad \text{for all } a \le x \le b, \tag{6}$$

i.e. $u = 0$ is a solution of (1). Notice that $J(0) = 0$. If we also assume

$$\lim_{u \to 0} \frac{F(x,u)}{u^2} = 0 \qquad \text{uniformly in } a \le x \le b, \tag{7}$$

then in view of Poincaré's inequality we have $J(u) > 0$ for sufficiently small u (see e.g., [7]). If there is a function $e(x)$ such that $J(e) < 0$, then assuming a technical condition (p_4) in [7], p. 9, which is satisfied in all our examples, the mountain pass lemma guarantees existence of a nontrivial solution. That is the solution that we compute.

Our algorithm breaks naturally into two parts (and two separate computer programs). In the first part of the algorithm we interactively compute an approximation to the minimax of J. In the second part we use this approximation to begin a defect minimizing descent.

2. Description of the Algorithm

For the problem (1) we assume conditions (6) and (7) to hold, and let $e(x)$ be a known function, such that $J(e) < 0$. Typically we tried $e(x) = Ax(1 - x)$ with a constant

$A > 0$. 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) by its finite difference version

$$-\frac{u_{k+1} - 2u_k + u_{k-1}}{h^2} = f(x_k, u_k) \qquad k = 1, \ldots, N-1, \qquad u_0 = u_N = 0. \quad (8)$$

Set $U = (u_0, u_1, \ldots, u_N)$. Using numerical integration we represent $J = J(U)$. Define

$$d_k = -\frac{u_{k+1} - 2u_k + u_{k-1}}{h^2} - f(x_k, u_k) \qquad k = 1, \ldots, N-1,$$

$$p = p(U) = \sum_{k=1}^{N-1} d_k^2; \qquad \mathit{defmax}(U) = \max_{1 \le k \le N-1} |d_k|. \quad (9)$$

Solving (8) is equivalent to finding a zero of p, i.e. to the minimization of the nonnegative function $p(u)$. Since p has more than one minimum (including the trivial one), we must get close to the nontrivial solution of (8) before using a minimization algorithm for $p(U)$.

Part I. Interactive search for the minimax.

Step 1. With $e_k = e(x_k)$, $E = (e_0, e_1, \ldots, e_N)$, $\tau = \frac{1}{2N_1}$, N_1 an integer parameter at our disposal, we join E to the origin in R^{N+1} by a straight line, i.e. we define

$$V_i = (1 - i\tau)E, \qquad i = 0, 1, \ldots, 2N_1.$$

Assume $J(V_{i_0}) = \max_{0 \le i \le 2N_1} J(V_i)$. Then $U^1 = V_{i_0}$ is our first guess. Using (8) we compute $\mathit{defmax}(U^1)$. If $\mathit{defmax}(U^1)$ is judged to be small (usually less than one), then U^1 can be passed as the initial guess for Part II. If a smaller value of *defmax* is desired then, since the desired solution is a saddle point of J, it is natural to try to get closer to it on a path minimizing J.

Step 2. Compute the Frechet derivative and use integration by parts,

$$J'(u)v = \int_a^b (u'v' - f(x,u)v)dx = -\int_a^b (u'' + f(x,u))v\,dx.$$

Hence $J'(u)v < 0$ for $v = -\tau_1(-u'' - f(x,u))$, $\tau_1 > 0$. I.e., if we define $U^{1,0} = U^1$, and $(U_k = u_k)$

$$U_k^{1,i+1} = U_k^{1,i} - \tau_1 d(U_k^{1,i}), \qquad i = 1, \ldots, m_1, \qquad k = 1 \ldots N-1, \quad (10)$$

with $U_0^{1,i+1} = U_N^{1,i+1} = 0$, we expect J to decrease for τ_1 small. The constants $\tau_1 > 0$ and the number of steps m_1 are at our disposal. While iterating (10), we monitor $\mathit{defmax}(U_k^{1,i})$. If that quantity does not decrease, we keep m_1 small and pass to the next step.

Step 3. We join U^{1,m_1} by straight lines to E and to the origin, i.e., we let (τ and N_1 as in Step 1)

$$V_{1,i} = (1 - 2i\tau)E + 2i\tau U^{1,m_1}, \qquad i = 0, 1, \ldots, N_1;$$
$$\bar{V}_{1,i} = (1 - 2i\tau)E, \qquad i = 0, 1, \ldots, N_1.$$

Among all $V_{1,i}$ and $\bar{V}_{1,i}$ we select the one maximizing J, and call it U^2.

We then return to Step 2 to produce $U^{2,1}, \ldots, U^{2,m_2}$. We continue to alternate between steps 2 and 3, terminating the process when $defmax(U^{s,m_s})$ is suitably small. We could always accomplish that in our experiments.

Part II. We use gradient descent to minimize $p(U)$. Compute

$$p_{u_1} = -2\left(\frac{-2d_1 + d_2}{h^2} + f_u(x, u_1)d_1\right)$$

$$p_{u_k} = -2\left(\frac{d_{k-1} - 2d_k + d_{k+1}}{h^2} + f_u(x, u_k)d_k\right), \qquad k = 2, \ldots, N-2,$$

$$p_{u_{N-1}} = -2\left(\frac{-d_{N-2} + 2d_{N-1}}{h^2} + f_u(x, u_{N-1})d_{N-1}\right),$$

i.e., if we define $d_0 = d_N = 0$, $d = (d_0, d_1, \ldots, d_N)$, then we can write

$$-\nabla p = 2\Delta_h d + 2f_u(x, u)d \qquad (\Delta_h d - \text{discrete version of } d'').$$

Starting with $U_0 = (u_{01}, \ldots, u_{0N-1}) = (U_1^{s,m_s}, \ldots, U_{N-1}^{s,m_s})$ (the boundary values are not used in Part II), we iterate

$$U_{i+1k} = U_{ik} - \tau_2 \nabla p(x_k, U_{ik}), \qquad i = 1, \ldots, M, \qquad k = 1, \ldots, N-1,$$

with suitably chosen $\tau_2 > 0$ and M. This allows us to improve the solution at the interior mesh points (keeping $u_0 = u_N = 0$).

3. Numerical Experiments

First we describe a modification of "Gauss-Seidel" type, which allowed us to obtain better approximations to the solution (smaller *defmax*) on Part I of the algorithm. One replaces (10) by

$$U_k^{r,i+1} = U_k^{r,i} - \tau_1 d(U_1^{r,i+1}, \ldots, U_{k-1}^{r,i+1}, U_k^{r,i}, \ldots, U_{N-1}^{r,i}) \tag{11}$$

for all r, i and k. We used this modification in the examples below, although the algorithm also works without it.

Example 1. $-u'' = 10u^2, 0 < x < 1, u(0) = u(1) = 0$. We took $h = 0.05$, $N_1 = 5$ (i.e. $\tau = \frac{1}{10}$), $e(x) = 9x(1-x)(J(e) < 0)$. On Step 1 the maximum value of $J \simeq 1.72$ was achieved at $0.6e(x)$. (The values of J were computed using the trapezoidal rule with $h = 0.05$, and they are not very accurate. However, what we need is relative values of J at various points). For steps 2 and 3 we took $\tau_1 = 0.001$, and $m_1 = m_2 = \cdots = m_p = 12$. We made several trial runs with various p (the number of repetitions as defined in Step 3). It was possible to obtain small values of *defmax*, and we also noticed that minimax value of $J \simeq 1.34$ (below this value *defmax* was increasing). The best result was achieved when the following modification was introduced in Step 2: when $J < 1.35$ then $\tau_1 := 0.0005$. Then with $p = 2$, $m_1 = 12$ and $m_2 = 375$ we reduced *defmax* to 0.002759.

Corresponding $U = (u_0, u_1, \ldots, u_N)$ is already a good approximation of the solution. We then passed this u to defect minimization program. We took $\tau_2 = 0.0000008$, and after $M = 109500$ steps we achieved *defmax* $= 0.000000$. The values of the solution at mesh points are given in the Table 1. Notice that the value of τ_2 has to be small (when we tried $\tau_2 = 0.0000009$, then *defmax* was increasing).

The computation of Part I took only a few seconds on SUN 386, while Part II took around 90 minutes. (One could use faster descent algorithms like PARTAN).

Example 2. $-u'' = u^2 + xu^3$, $0 < x < 1$, $u(0) = u(1) = 0$. We found it advantageous to rescale the problem by letting $u = 10v$, obtaining

$$-v'' = 10v^2 + 100xv^3 \qquad 0 < x < 1, \qquad v(0) = v(1) = 0.$$

This allowed us to take $e(x) = Ax(1 - x)$ with $A = 4$. For the original problem we would need to choose A considerably larger. This would necessitate taking large N_1, and also would make the vaues of J less accurate. We took $h = 0.1$, $N_1 = 10$. On Step 1 the maximum value of $J \simeq 0.304$ was achieved at 0.45 $e(x)$. For the Steps 2 and 3 we took $\tau_1 = 0.001$ and $m_1 = m_2 = \cdots = m_5 = 12$, i.e., we repeated the Step 2-Step 3 five times making 12 iterations each time. For the above five repetitions, *defmax* would typically decrease on the first few iterations of the Step 2, and then

Table 1

$u(0.00) = 0.0000000000$
$u(0.05) = 0.1646599276$
$u(0.10) = 0.3286420331$
$u(0.15) = 0.4899239992$
$u(0.20) = 0.6452053275$
$u(0.25) = 0.7900794086$
$u(0.30) = 0.9193478535$
$u(0.35) = 1.0274862874$
$u(0.40) = 1.1092315204$
$u(0.45) = 1.1602168902$
$u(0.50) = 1.1775496803$
$u(0.55) = 1.1602168902$
$u(0.60) = 1.1092315204$
$u(0.65) = 1.0274862874$
$u(0.70) = 0.9193478535$
$u(0.75) = 0.7900794086$
$u(0.80) = 0.6452053275$
$u(0.85) = 0.4899239992$
$u(0.90) = 0.3286420331$
$u(0.95) = 0.1646599276$
$u(1.00) = 0.0000000000$
defmax $= 0.000000$

increase again. On the sixth repetition of Step 2 *defmax* kept on decreasing, so we took $m_6 = 21$ (the optimal number), and obtained an approximation to the solution with *defmax* $\simeq 0.093$. For the defect minimization we took $\tau_2 = 0.00001$, and after 4500 steps we again obtained *defmax* $= 0.000000$. The entire computation took under 5 minutes on SUN 386. The solution is given in Table 2.

Example 3. $-u'' = \lambda u^2(1-u)$, $0 < x < 1$, $u(0) = u(1) = 0$. Multiplying the equation by u, integrating by parts and using Poincaré's inequality, we conclude that no positive solution exists for $\lambda \le 4\pi^2$. For $\lambda > \lambda_0$, where $\lambda_0 > 4\pi^2$ a critical value, two positive solutions exist, see [7]. The solutions are ordered $0 < u_1(x) < u_2(x)$ for $0 < x < 1$. 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. We obtained $u_2(x)$ using a monotone scheme developed in [5], i.e., as a limit when $t \to \infty$ of the corresponding evolution equation, with $u(x,0)$ being a supersolution (we took $u(x,0) = 2$). Compute:

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

It is easy to see that $u = 0$ is a point of local minimum of $J(u)$. The stable solution $u_2(x)$ is also a point of local minimum. We indicate the argument next. Suppose a function of $\psi(x)$ is close to $u_2(x)$ in $H_0^1(0,1)$, and $J(\psi) < J(u_2)$. Since $u_2(x)$ is stable, it follows that the solution of the corresponding evolution equaton, starting at $\psi(x)$, will tend to $u_2(x)$. But along that solution $J(u)$ is decreasing in t (see [3]), a contradiction. One can think of a "graph" of $J(u)$ as having two local minimums 0 and $u_2(x)$, and a saddle point between them corresponding to $u_1(x)$.

Next we describe computations for $\lambda = 60$. To compute $u_1(x)$ we took $u_2(x)$ as the starting function $e(x)$. For Part I we took $h = 0.05$, $N_1 = 5$, $\tau_1 = 0.001$, $m_1 = \cdots = m_7 = 12$, $m_8 = 50$, and obtained an approximation to the solution with *defmax* $\simeq 0.0093$. For Part II we took $\tau_2 = 0.0000008$ ($\tau_2 = 0.00000081$ led to overflow), and after $M = 243000$ steps obtained *defmax* $= 0.000000$. While computations of $u_2(x)$

Table 2

$v(0.00) = 0.0000000000$
$v(0.10) = 0.0986561959$
$v(0.20) = 0.1962430650$
$v(0.30) = 0.2884672838$
$v(0.40) = 0.3651688652$
$v(0.50) = 0.4090577585$
$v(0.60) = 0.4019903688$
$v(0.70) = 0.3397872714$
$v(0.80) = 0.2385774459$
$v(0.90) = 0.1208119918$
$v(1.00) = 0.0000000000$
defmax $= 0.000000$

Table 3

$u1(x)$	$u2(x)$
$u(0.00) = 0.0000000000$	$u(0.00) = 0.0000000000$
$u(0.05) = 0.0357196888$	$u(0.05) = 0.1513223995$
$u(0.10) = 0.0712548296$	$u(0.10) = 0.2997297864$
$u(0.15) = 0.1060826497$	$u(0.15) = 0.4387005478$
$u(0.20) = 0.1394015116$	$u(0.20) = 0.5614673093$
$u(0.25) = 0.1702118007$	$u(0.25) = 0.6634972496$
$u(0.30) = 0.1974159897$	$u(0.30) = 0.7433064697$
$u(0.35) = 0.2199283046$	$u(0.35) = 0.8018420399$
$u(0.40) = 0.2367809905$	$u(0.40) = 0.8412667415$
$u(0.45) = 0.2472151690$	$u(0.45) = 0.8638404062$
$u(0.50) = 0.2507483433$	$u(0.50) = 0.8711733140$
$u(0.55) = 0.2472151690$	$u(0.55) = 0.8638404062$
$u(0.60) = 0.2367809905$	$u(0.60) = 0.8412667415$
$u(0.65) = 0.2199283046$	$u(0.65) = 0.8018420399$
$u(0.70) = 0.1974159897$	$u(0.70) = 0.7433064697$
$u(0.75) = 0.1702118007$	$u(0.75) = 0.6634972496$
$u(0.80) = 0.1394015116$	$u(0.80) = 0.5614673093$
$u(0.85) = 0.1060826497$	$u(0.85) = 0.4387005478$
$u(0.90) = 0.0712548296$	$u(0.90) = 0.2997297864$
$u(0.95) = 0.0357196888$	$u(0.95) = 0.1513223995$
$u(1.00) = 0.0000000000$	$u(1.00) = 0.0000000000$
$defmax = 0.000000$	$defmax = 0.000000$

and of Part I took only a few seconds, it took us close to 4 hours for Part II (SUN 386). Both positive solutions are given in Table 3.

Acknowledgement

It is a pleasure to thank D. Schmidt for constant help with computer software and hardware, and his interest in this work.

References

[1] Allgower, E. L., Chien, C. S., Georg, K.: Large sparse continuation problems. In: Mittelmann, H. D., Roose, D. (eds.) Continuation techniques and bifurcation problems, pp. 3–21. Boston: Birkhäuser, 1990.

[2] Choudury, G., Korman, P.: On computation of solutions of fully nonlinear elliptic problems. J. Comp. Appl. Math. *41*, 301–311 (1992).

[3] Henry, D.: Geometric theory of semilinear parabolic equations. Berlin, Heidelberg, New York: Springer 1981 (Lecture Notes in Mathematics, *840*).

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

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

[6] Luning, C. D., Perry, W. L.: Positive solutions of negatative exponent generalized Emden-Fowler boundary value problems. SIAM J. Math. Anal. *12* (6), 874–879 (1981).

[7] Rabinowitz, P. H.: Minimax methods in critical point theory with applications to differential equations, CBMS Reg. Conf. Ser. in Math. No. 65, Amer. Math. Soc., Providence, RI (1986).

[8] Sattinger, D. H.: Monotone methods in nonlinear elliptic and parabolic boundary value problems. Indiana Univ. Math. J. *21*, 979–1000 (1972).
[9] Smiley, M. W.: A numerical study of spontaneous bifurcation. J. Comp. Appl. Math. *19*, 179–188 (1987).

Philip Korman
Institute for Dynamics and
Department of Mathematical Sciences
University of Cincinnati
Cincinnati, OH 45221-0025
U.S.A.