Department of Applied Analysis
and Computer Science
Technical Report CS-73-21
June, 1973

"A METHOD FOR THE NUMERICAL DETERMINATION OF BIFURCATION STATES OF NONLINEAR SYSTEMS OF EQUATIONS"

by

R.B. Simpson
University of Waterloo

This work was supported by the National Research Council of Canada under grant A8239.

ABSTRACT

A numerical technique is presented for determining a simple turning point in a branch of solutions of an algebraic system of equations depending on a scalar parameter. Results are given from testing the method on discrete versions of several mildly nonlinear boundary value problems to determine turning points in the positive solution branch.

§1. Introduction

We wish to describe a numerical technique for analysing a simple form of singular behaviour in the solutions of nonlinear algebraic systems of equations. Consider such a system of N equations for unknown N vector, u, which depends on a scalar parameter, λ.

$$A(u, \lambda) = 0 \tag{1.1}$$

A solution 'curve', $u(\lambda)$, which varies continuously with λ is called a branch of solutions. As a function of λ, $u(\lambda)$ will, in general, be multiple valued and can exhibit complex behaviour involving changing of its multiplicity (see [13] and its extensive bibliography). We shall treat a simple common case here, which we shall refer to as a turning point in the branch.

Definition $\bar{\lambda}$ is a turning point of (1.1) if for every sufficiently small neighbourhood of λ :

a) (1.1) has a continuous branch of solutions $(\lambda, u(\lambda))$ through $(\bar{\lambda}, u(\bar{\lambda}))$

b) for $(\lambda, u(\lambda))$ on the branch of a),

$\lambda \leq \bar{\lambda}$ (turning to the left) or

$\lambda \geq \bar{\lambda}$ (turning to the right).

The problem that we wish to tackle, then, is the numerical determination of the turning points of a solution branch.

If the Jacobian matrix of A with respect to u is denoted by $A_u(u, \lambda)$, then the eigenvalues of $A_u(w, \rho)$ can be designated by $\mu_i(w, \rho)$. It is well known ([8]) that if $\overline{\lambda}$ is a turning point of (1.1) then

$$(1.2) \qquad \mu_k(u(\overline{\lambda}), \overline{\lambda}) = 0 \qquad \text{for some } k$$

Hence, the problem of determining the turning points of branches of solutions of (1.1) can be viewed as a special case of solving (1.1) and (1.2) simultaneiously. This is the basis of the method described here, and we assume throughout that $A_u(u(\lambda), \lambda)$ is a symmetric matrix.

Such turning points occur in boundary value problems (with u being a function) for the equilibrium states of nonlinear systems (elasticity [1], [2], fluid mechanics [3], or nonlinear diffusion processes [4], [5]). They mark the loss (or gain) of dynamic stability of such states as the parameter, λ, varies, [5]. We intend the method to be able to analyse discretized versions of such problems and have tested it on problems with u being a function defined on a finite difference grid in the x-y plane and

$$(1.3) \qquad A(u, \lambda) = \Delta_h u + \lambda f(u)$$

Here Δ_h represents a finite difference version of the Laplacian and $f(y)$ is an increasing function of y. These problems are discrete versions of mildly nonlinear eigenvalue problems and the relation of their solution branches to the solution branches of their continuous counterparts has been studied by S. V. Parter in [6] and [7] and the author in [11] and [12].

Extensive numerical computations by a different approach to problems (1.3) have been reported by J. B. Rosen in [11]. The technique provides numerical solutions and error bounds for equations with a monotonicity property. This property holds for the minimal positive solution of a branch of positive solutions of (1.1) and is lost at $\lambda = \overline{\lambda}$. Hence upper bounds for $\overline{\lambda}$ were obtained in [11] by observing the λ values for which the technique failed. The problem of computing a solution branch in the presence of a turning point has also be treated by Anselone and Moore in [1], and by Thurston in [15], from the point of view of getting past the turning point to the rest of the solution branch.

§2. The Method

In this section, we describe a numerical method for computing turning points of branches of nonlinear systems as discussed in the preceding section. This is done in three subsections after presenting the viewpoint and background assumptions; two sections on a mathematical description of the method and one on some details of its implimentation for the experiments reported on.

We assume that rough quantitative knowledge of the turning point has been obtained, i.e. an estimate of $\overline{\lambda}$ and whether the branch turns to the left $(u(\lambda)$ is defined for $\overline{\lambda} - \varepsilon \leq \lambda \leq \overline{\lambda}$) or the right. Some analytic techniques are available for this purpose ([4], [5].) In the examples presented this was done by scanning the branch for a choice of λ values using Newton's method. We shall describe the method as applied to turning points turning to the left in this section.

It is based on an inverse interpolation technique for solving

$$\mu(u(\overline{\lambda}), \overline{\lambda}) = 0 \tag{2.1}$$

(here we have suppressed the subscript of the eigenvalue, μ ,) where $\mu(u(\lambda), \lambda)$ is a 'multi valued function' of λ as indicated by the solid line of Fig. 2.

§2.1 Modified inverse interpolation

For the jth step of the iterative procedure for solving (2.1), three current points on the graph of $\mu(u(\lambda), \lambda)$ are retained

(λ_j, μ_j), $(\lambda_{j+1}, \mu_{j+1})$, $(\lambda_{j+2}, \mu_{j+2})$. The inverse data (μ_i, λ_i) $i = j, j+1, j+2$ is interpolated by a quadratic, $P_2(\mu)$, and the next estimate of $\overline{\lambda}$, λ_{j+3} is calculated as

$$(2.2) \qquad \lambda_{j+3} = \alpha P_2(0) + (1 - \alpha)\lambda_j$$

for some parameter value, α , $0 < \alpha < 1$.

One must ensure, of course, that $\lambda_{j+3} < \overline{\lambda}$, and this is the function of the parameter α . In Lemma 1, it is stated that the λ_j converge montonely to $\overline{\lambda}$ under suitable conditions. Strictly speaking, to get monotonicity, a reassignment of indices j+1, j+2, j+3 may be necessary after jth step. These conditions refer to $\lambda(\mu)$ as the inverse function to $\mu(u(\lambda), \lambda)$ and require

$$(2.3) \qquad \max \left| \frac{d^3\lambda(\mu)}{d\mu^3} \right| \leq K_3$$

$$\text{and} \qquad \max \left| \frac{d^2\lambda(\mu)}{d\mu^2} \right|^{-1} \leq M_2$$

where the maxima are taken for $|\mu(\lambda_1)| \leq |\mu| \leq |\mu(\overline{\lambda})|$.

Lemma 1 Let $k = K_3(2M_2)^{3/2}/6$ and λ_4 be defined by (2.2) with $\lambda_1 \leq \lambda_2 \leq \lambda_3 < \overline{\lambda}$. If

$$(2.4) \qquad k(\overline{\lambda} - \lambda_1)^{1/2} = \rho < 1$$

then for $0 < \alpha < 1/(1+\rho)$,

(2.5) $$\lambda_1 < \lambda_4 < \bar{\lambda}$$

The λ_j converge to $\bar{\lambda}$ with linear asymptotic convergence rate $= 1 - \alpha$. (Proof is in Appendix A.) One could vary α with j so that $\alpha_j \to 1$ as $j \to \infty$; however in the experiments tried a constant α was used, in the range .85 to .97 usually.

§2.2 Evaluation of $\mu(u(\lambda), \lambda)$

The process just described requires the evaluation of $\mu(u(\lambda), \lambda)$ for given λ . This involves solving

(2.6) $$A(u(\lambda), \lambda) = 0$$

for $u(\lambda)$ and calculating the relevant eigenvalue, μ , of $A_u(u(\lambda), \lambda)$. To accomplish this, we exploit some properties of Newton's method observed by Rall, [10], generalizing the well known behaviour of Newton's method for multiple roots of a scalar equation.

The results pertinent to this discussion can be qualitatively summarized as follows. Suppose that $A_u(w, \lambda)$ is symmetric; that λ is set to $\bar{\lambda}$ and hence (2.6) has $\overline{u(\lambda)}$ as a multiple solution; and suppose that Newton's method for (2.6) produces $\{u_n(\bar{\lambda})\}$ which converges to $\bar{u}(\bar{\lambda})$.

If we let e_n be the error in $u_n(\bar{\lambda})$, i.e.

(2.7) $$e_n = u_n(\bar{\lambda}) - \bar{u}(\bar{\lambda})$$

then the components of e_n in the directions of the null space of

$A_u(\bar{u}, \bar{\lambda})$ are reduced at a linear rate, while those in the orthogonal complement are reduced at a quadratic rate.

If we are using λ near $\bar{\lambda}$, we could expect that similar behaviour of the error in the Newton iterates would occur i.e. that as the error, e_n , decreases, its direction is predominantly in the eigenspace of $\mu(u(\lambda), \lambda)$,(the eigenvalue of $A_u(u(\lambda), \lambda)$ closest to zero). Although we would not know the error in the nth iterate, e_n , we could expect that

$$(2.8) \qquad c_n = u_{n+1}(\lambda) - u_n(\lambda) \cong e_n$$

i.e. that the correction to the nth iterate has predominantly the direction of the eigenspace referred to above. This vector is therefore put into the Rayleigh quotient for $A_u(u(\lambda), \lambda)$ to give an approximate value for $\mu(\lambda)$. In practice its direction could be improved somewhat by several steps of the inverse power technique for calculating eigenvalues.

In principle, the nth step of Newton's method for calculating $\mu(u(\lambda), \lambda)$ from a given $\lambda \cong \bar{\lambda}$ could be described by (2.9). In (2.9) we are using square brackets $[w, v]$ to denote the inner product of vectors w and v ,

$$(2.9) \qquad A_u(u_n(\lambda), \lambda)c_n = -A(u_n(\lambda), \lambda) \equiv r_n$$

$$\mu_n = [c_n, A_u(u_n(\lambda), \lambda)c_n]/[c_n, c_n] = [c_n, r_n]/[c_n, c_n]$$

$$u_{n+1}(\lambda) = u_n(\lambda) + c_n$$

with $u_n(\lambda)$ converging to $u(\lambda)$ and μ_n converging to $\mu(u(\lambda), \lambda)$

for a successful implementation.

§2.3 Implementation Details

In implementing (2.9) one might anticipate that if Newton's method is continued until c_n is quite small, or effectively stops decreasing, then the direction of c_n would be heavily contaminated by round off effects, and in particular, might not be primarily in the eigenspace of $\mu(u(\lambda), \lambda)$. Hence (2.9) was terminated while c_n was still several orders of magnitude larger than the machine resolution of u_n .

When (2.9) was terminated, a couple of steps of the inverse power method for computing μ were taken to improve the numerical value of μ somewhat. Typically this resulted in about a 10% correction in the computed value of μ . (See Table 1a)

In Newton's method, and the inverse power process, a linear system of equations must be solved with a sparse, symmetric matrix of coefficients which may not be positive definite. The system was solved using a Gaussian elimination program for band matrices which performed partial pivoting (GELB of the IBM Scientific Subroutine Package). In no runs did the program warning parameter signal the possible loss of significant digits.

Lemma 1 indicates that $\lambda_1 < \lambda_4 < \overline{\lambda}$, but does not give the relation between λ_2, λ_3 and λ_4 . In this implimentation, a three vector ($\lambda(1)$, $\lambda(2)$, $\lambda(3)$) is retained, but not in order. Instead, a variable, irepl, is maintained so that λ(irepl) is the least of the λ

values at each stage (or the greatest for turning to the right). After λ_4 is computed, it replaces $\lambda(\text{irepl})$ and, after $\mu(u(\lambda_4), \lambda_4)$ is computed, irepl is recomputed.

In the discussion of Newton's method in the preceding section, the topic of starting values was omitted. One could retain $u(\lambda_1)$, $u(\lambda_2)$ and $u(\lambda_3)$ and if, e.g. λ_2 were being replaced by λ_4 , use $u(\lambda_2)$ as the starting value for calculating $u(\lambda_4)$. Rather than retain $u(\lambda_i)$, $i = 1,2,3$, we made a least squares fit using two parameters,

$$A(\lambda_i)\phi_1 + B(\lambda_i)\ _2$$

for each $u(\lambda_i)$ and retained $A(\lambda_i)$ and $B(\lambda_i)$. The shape functions ϕ_1 and ϕ_2 are discussed below.

The program was written to identify vectors u and c as mesh functions defined on a subset of a square mesh of spacing h on the unit square $0 \leq x \leq 1$, $0 \leq y \leq 1$. This is a natural identification for boundary value problems in two independent variables; but, of course, other problems can be adapted to it. The submesh used is described by two variables, RBOTM and RTOP indicating that the lines $Y_{bot} = (\text{RBOTM} - 1)h$ and $Y_{top} = (\text{RTOP} - 1)h$ are the bottom and top mesh boundary rows; and by two functions of y , $f_\ell(y)$ and $f_r(y)$. The mesh subregion then is

$$\{(i-1)h,\ (j-1)h \mid i,j \text{ positive integers } \text{RBOTM} < j < \text{RTOP},$$
$$f_\ell((j-1)h) < (i-1)h < f_r((j-1)h)\} \ .$$

The shape functions, ϕ_1 and ϕ_2 , used in the least squares fitting process then are the restrictions to the mesh of

$$\Phi_i(x, y) = (y - (\mathrm{RBOTM} - 1)h)((\mathrm{RTOP} - 1)h - y) *$$
$$(x - f_\ell(y))(f_r(y) - x) \cos (\pi\alpha_i x) \cos(\pi\alpha_i y)$$

where α_i , i = 1, 2, are chosen to suit the symmetries of the problem at hand. The least squares parameters for the solutions are stored and used to generate initial guesses when required; they are printed out to provide a compact description of the solution branch (see Table 3), and they are used to generate initial guesses for subsequent runs. A schematic flow chart of the method as implimented, then, is given in Fig. 1.

§3. Some Experiments

The method was tested on some finite difference versions of the mildly nonlinear boundary value problem for plane regions D

$$(3.1) \qquad \begin{aligned} -\Delta u(x) &= \lambda f(u(x)) & x \in D \\ u(x) &= 0 & x \in \partial D \end{aligned}$$

The results reported on here are for:

A) D - unit square $(0 \le y \le 1;\ 0 = f_\ell(y) \le x \le f_r(y) = 1)$ and $f(u) = e^u$

B) D - six sided region bounded by $y = 0;\ y = 1;$ $x = f_\ell(y) \equiv \max(0, y-\frac{1}{2})$; $x = f_r(y) = \min(y+\frac{1}{2}, 1)$; and $f(u) = e^u$

C) D - unit square and $f(u) = 1 + (u + u^2/2)/(1 + u^2/100)$.

For the problems A) and C) involving the unit square, the nine point box form of the discrete Laplacian was used, with a nonlinear deferred correction as described in [9]. For problem B) the five point discrete Laplacian was employed.

Typical results for the relation of $\mu(u(\lambda), \lambda)$ to λ are shown in a graph on Fig. 2 composed of data from two runs for problem A using a mesh spacing of $h = 1/16$. In Table 1a an example of some computer output monitoring the progress of Newton's method and the inverse power method for evaluating $\mu(u(\lambda), \lambda)$ is shown. In the second half of the table (Table 1b) the later iterates of the modified inverse interpolation process for solving (2.1) is given (for this run

$\alpha = .9$). Runs of problems A and C using $h = 1/21$ showed no change in the accuracy of the results as given.

In Figure 3 is shown the part of the branches of positive solutions showing the turning point for problems A and B ($f(u) = e^u$, unit square and six sided domain). In Figure 4, the branch of positive solutions for problem C and also the variation of $\mu(u(\lambda), \lambda)$ with λ are shown. The branch shows two turning points, $\bar{\lambda}_1 = 7.957$ and $\bar{\lambda}_2 = 6.423$, it turns to the left at $\bar{\lambda}_1$ and to the right at $\bar{\lambda}_2$. As the branch turns upwards at $\lambda = \bar{\lambda}_1$, the critical eigenvalue becomes positive, indicating that the upper part of the branch has become dynamically unstable. As the branch turns back to the right at $\lambda = \bar{\lambda}_2$, the critical eigenvalue becomes negative once more. Hence this boundary value problem is a model for a system which has two stable steady states; a lower one existing for $0 < \lambda < \bar{\lambda}_1$ and an upper one existing for $\lambda > \bar{\lambda}_2$.

In Table 2 are presented the numerical estimates of the turning points as discussed.

Problem A	$\bar{\lambda} = 6.8082$		
Problem B	$\bar{\lambda} = 9.6384$		
Problem C	$\bar{\lambda}_1 = 7.957$	;	$\bar{\lambda}_2 = 6.423$

Table 2

The estimate of $\overline{\lambda} = 6.8082$ for problem A compares favourably with the upper bound of 6.81 for this problem obtained by J. B. Rosen [11].

As discussed in §2.3, the program computes least square coefficients, $A(\lambda)$, $B(\lambda)$, as brief summaries of $u(\lambda)$, the fitted forms being

$$u(x, y, \lambda) \overset{\sim}{=} \phi(x, y)(A(\lambda) + B(\lambda)\cos(a\pi x)\cos(a\pi y))$$

with $\phi(x, y) = (x - f_\ell(y))(f_r(y) - x)y(1 - y)$ and $a = 3$ for problems A and C and $a = 2$ for problem B . Some of these coefficients are given in Table 3 to provide a more quantitative description of the branches of positive solutions for these problems.

Appendix A (Proof of Lemma 1 §2.1)

The inverse interpolation method is effectively an iterative scheme for evaluating $\lambda(0) = \bar{\lambda}$. In §2.1, we let $P_2(\mu)$ be the quadratic polynomial interpolating the function $\lambda(\mu)$ at (μ_i, λ_i), $i = 1,2,3$. The improved estimate, λ_4, for $\bar{\lambda}$ based on this data then is given by (2.2) i.e.

$$\lambda_4 = \alpha P_2(0) + (1 - \alpha)\lambda_1 \quad . \tag{A.1}$$

Here we wish to show that for α chosen as in Lemma 1

$$\lambda_1 < \lambda_4 < \bar{\lambda} \quad . \tag{A.2}$$

Since $\mu = 0$ is a local maximum for $\lambda(\mu)$, we have (assuming $\lambda(\mu)$ to be three times continuously differentiable) that

$$\lambda_i = \lambda(\mu_i) = \bar{\lambda} + (d^2\lambda(S_i)/d\mu^2)\mu_i^2/2 \tag{A.3}$$

for some S_i, $\lambda_i < S_i < \bar{\lambda}$. Using

$$\bar{\lambda} - \lambda_1 \geq \bar{\lambda} - \lambda_j \qquad j = 2,3 \quad \text{and} \quad (2.3)$$

i.e.

$$\max_{\lambda_1 < S < \bar{\lambda}} |d^2\lambda(S)/d\mu^2|^{-1} \leq M_2$$

we get

$$\begin{aligned} \mu_i^2 &= 2(\lambda_i - \bar{\lambda})/(d^2\lambda(S_i)/d\mu^2) \\ &\leq 2(\bar{\lambda} - \lambda_1)M_2 \qquad\qquad i = 1,2,3 \ . \end{aligned} \tag{A.4}$$

Consequently, using the basic interpolation error formula and K_3 from (2.3),

$$(A.5) \qquad |\overline{\lambda} - P_2(0)| \leq K_3|\mu_1 \ \mu_2 \ \mu_3|/6$$

$$\leq (K_3(2M_2)^{3/2}/6)(\overline{\lambda} - \lambda_1)^{3/2}$$

Now

$$(A.6) \qquad \overline{\lambda} - \lambda_4 = (1 - \alpha)(\overline{\lambda} - \lambda_1) + \alpha(\overline{\lambda} - P_2(0))$$

so, using (A.5), and setting

$$k = K_3(2 \ M_2)^{3/2}/6$$

we get

$$(A.7) \quad (1 - \alpha)(\overline{\lambda} - \lambda_1) - \alpha k(\overline{\lambda} - \lambda_1)^{3/2} \leq \overline{\lambda} - \lambda_4 \leq (1 - \alpha)(\overline{\lambda} - \lambda_1) + \alpha k(\overline{\lambda} - \lambda_1)^{3/2} .$$

Hence, if $\rho = k(\overline{\lambda} - \lambda)^{1/2} < 1$ and $0 < \alpha < 1/(1 + \rho)$ (A.7) shows that $0 < \overline{\lambda} - \lambda_4$ and that as $\overline{\lambda} - \lambda_1$ approaches zero,

$$(A.8) \quad \overline{\lambda} - \lambda_4 = (1 - \alpha)(\overline{\lambda} - \lambda_1) + 0((\overline{\lambda} - \lambda_1)^{3/2}) \quad .$$

Bibliography

1. Anselone P.M. and Moore R.H. "An Extension of the Newton-Kantorovich-Method for Solving Nonlinear Equations with an Application to Elasticity," J. Math. Anal. Appl. Vol.13 (1966), pp. 476-501.

2. Bauer L., Reiss E.L. and Keller H.B. "Axisymmetric Buckling of Hollow Spheres and Hemispheres," Comm. Pure and Applied Math. Vol. 23 (1970), pp. 529-568.

3. Busse F.H. "On the Stability of Two Dimensional Convection in a Layer Heated from Below," J. of Math. and Physics (1968), pp. 140-150.

4. Joseph D.D. and Sparrow E.M. "Nonlinear Diffusion Induced by Nonlinear Sources," Quart. Appl. Math. Vol. 28 (1970), pp. 327-342.

5. Keller H.B. and Cohen D.S. "Some Positone Problems Suggested by Nonlinear Heat Generation," J. of Math. and Mech. Vol. 16 (1967), pp. 1361-1376.

6. Parter S.V. "Mildly Nonlinear Elliptic Partial Differential Equations and their Numerical Solution I," Numer. Math. Vol. 7 (1965), pp. 113-128.

7. Parter S.V. "Maximal Solutions of Mildly Nonlinear Elliptic Equations," Numerical Sol'n of Nonlinear Diff'l Equations, D. Greenspan ed. John Wiley, New York (1966), pp. 213-238.

8. Pimbley G.H. "Eigenfunction Branches of Nonlinear Operators, and their Bifurcations," Lecture Notes in Mathematics No. 104, (1969), Springer-Verlag.

9. Pererya V. "Iterated Deferred Corrections for Nonlinear Boundary Value Problems," Numer. Math. Vol. 11 (1968), pp. 111-125.

10. Rall L.B. "Convergence of the Newton Process to Multiple Solutions," Num. Math. Vol. 9 (1966), pp. 23-37.

11. Rosen J.B. "Approximate Solution and Error Bounds for Quasi-Linear Elliptic Boundary Value Problems," SIAM J. of Num. Anal. Vol. 7 (1970), pp. 80-103.

12. Simpson R.B. "Finite Difference Methods for Mildly Nonlinear Eigenvalue Problems," SIAM J. of Num. Anal. Vol. 8 (1971),pp.190-211.

13. Simpson R.B. "Existence and Error Estimates for Solutions of a Discrete Analog of Nonlinear Eigenvalue Problems," Math. of Comp. Vol. 26 (1972), pp. 359-375.

14. Stakgold I. "Branching of Solutions of Nonlinear Equations," SIAM Review, Vol. 13 (1971), pp. 289-332.

15. Thurston G.A. "Continuation of Newton's Method through Bifurcation Points," J. of Appl. Mech. Vol. 36 (1969), pp. 425-430.

```
          NEWTONS  METHOD FOR LAMBDA =  6.8082

ITERATION NUMBER   EIGENVALUE ESTIMATE   SIZE OF NEWTON CORRECTION

       1                 -153.9710                  1.560E-01

       2                    0.1004                  2.804E-02

       3                    0.1438                  1.862E-02


          INVERSE ITERATION PHASE FOR LAMBDA =  6.8082

ITERATION NUMBER   EIGENVALUE ESTIMATE

       1                    0.1438

       2                    0.1048

       3                    0.1048

 LAMBDA=    6.80817  UMAX=    1.2732


          NEWTONS  METHOD FOR LAMBDA =  6.8081

ITERATION NUMBER   EIGENVALUE ESTIMATE   SIZE OF NEWTON CORRECTION

       1                 -158.2972                  1.595E-01

       2                    0.0481                  2.421E-02

       3                    0.1429                  5.743E-03

       4                    0.1349                  5.554E-03


          INVERSE ITERATION PHASE FOR LAMBDA =  6.8081

ITERATION NUMBER   EIGENVALUE ESTIMATE

       1                    0.1349

       2                    0.1236

       3                    0.1236

 LAMBDA=    6.80814  UMAX=    1.2742
```

TABLE 1A

LAMBDA	MU	UMAX
6.80462	.7315	1.3065
6.80659	.5001	1.2943
6.80725	-.3128	1.2508
6.80779	.2703	1.2821
6.80802	.1869	1.2776
6.80807	-.0919	1.2627
6.80814	.1236	1.2742
6.80817	.1048	1.2732

6.80822 - Newton's method failed to converge

Table 1b

Problem A

λ	Minimal positive Solution		Upper positive Solution	
	$A(\lambda)$	$B(\lambda)$	$A(\lambda)$	$B(\lambda)$
5.50	10.6	-.04	34.1	6.45
6.00	12.6	.11	30.2	4.81
6.25	13.9	.24	29.1	3.99
6.75	18.6	.91	23.2	1.98

Problem B

λ	Minimal positive Solution		Upper positive Solution	
	$A(\lambda)$	$B(\lambda)$	$A(\lambda)$	$B(\lambda)$
8.80	15.7	10.1	35.1	11.1
9.00	16.7	10.6	33.6	11.6
9.50	*	*	28.4	12.5

Problem C

λ	Minimal positive Solution		Unstable positive solution		Upper positive Solution	
	$A(\lambda)$	$B(\lambda)$	$A(\lambda)$	$B(\lambda)$	$A(\lambda)$	$B(\lambda)$
6.8	15.7	.25	90.9	12.2	240.	17.0
7.0	16.9	.36	79.2	10.6	265.	16.0
7.5	21.0	.81	57.8	70.8	*	*

Table 3

Some Least Squares Parmaeters for Positive Solution Branches

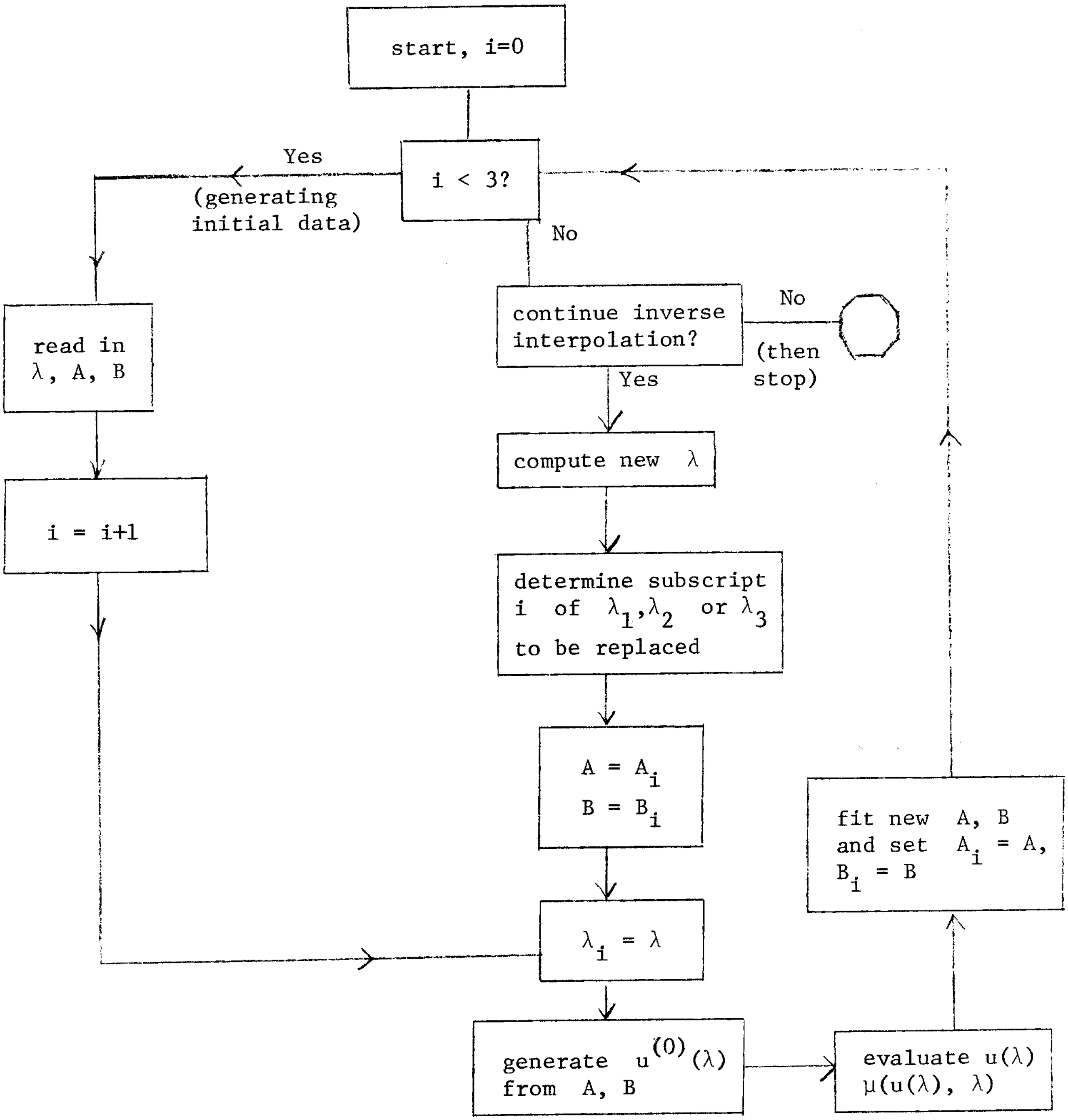
start, i=0
i < 3?
Yes
(generating initial data)
No
read in λ, A, B
i = i+1
continue inverse interpolation?
No
(then stop)
Yes
compute new λ
determine subscript i of λ_1,λ_2 or λ_3 to be replaced
A = A_i
B = B_i
λ_i = λ
generate $u^{(0)}(\lambda)$ from A, B
evaluate u(λ) μ(u(λ), λ)
fit new A, B and set A_i = A, B_i = B

FIGURE 1

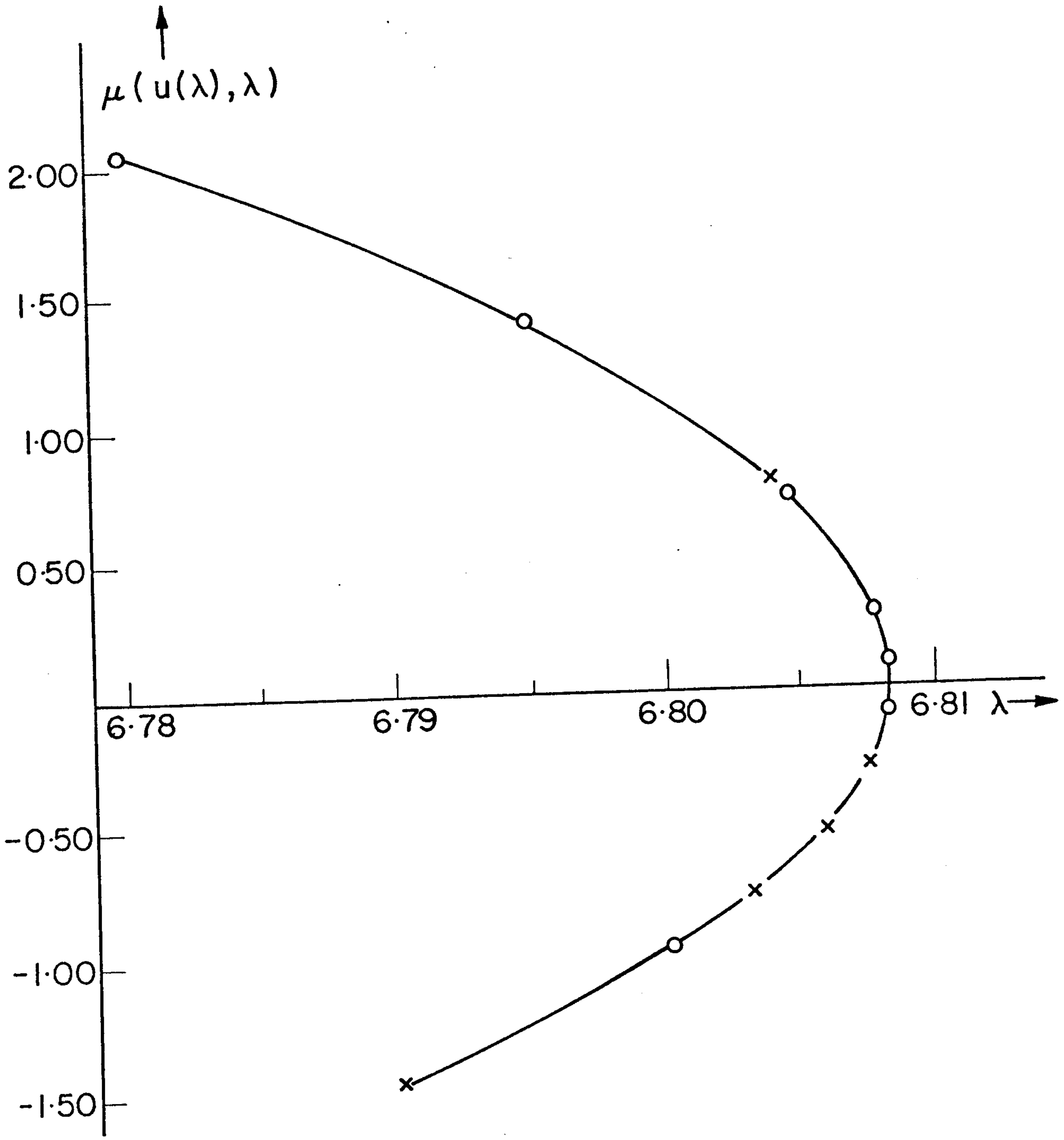

FIGURE 2

Variation of Critical Eigenvalue, $\mu(u(\lambda), \lambda)$ with λ
$f(u) = e^u$; unit square; (x, 0 denotes different runs)

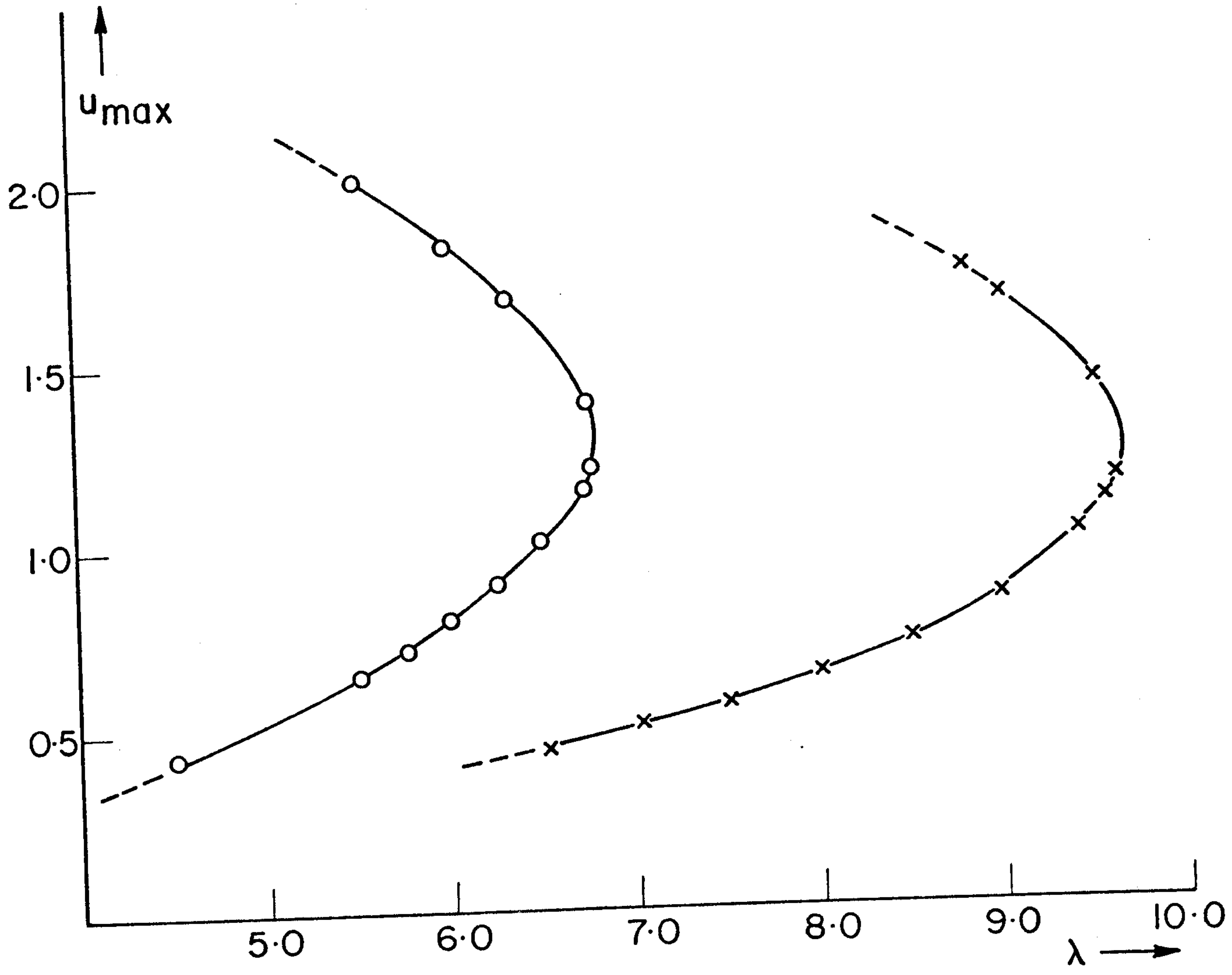

FIGURE 3

Positive Solution Branches for $f(u) = e^u$

—○— unit square (9 point discrete Laplacian)

—×— six sided region (5 point discrete Laplacian)

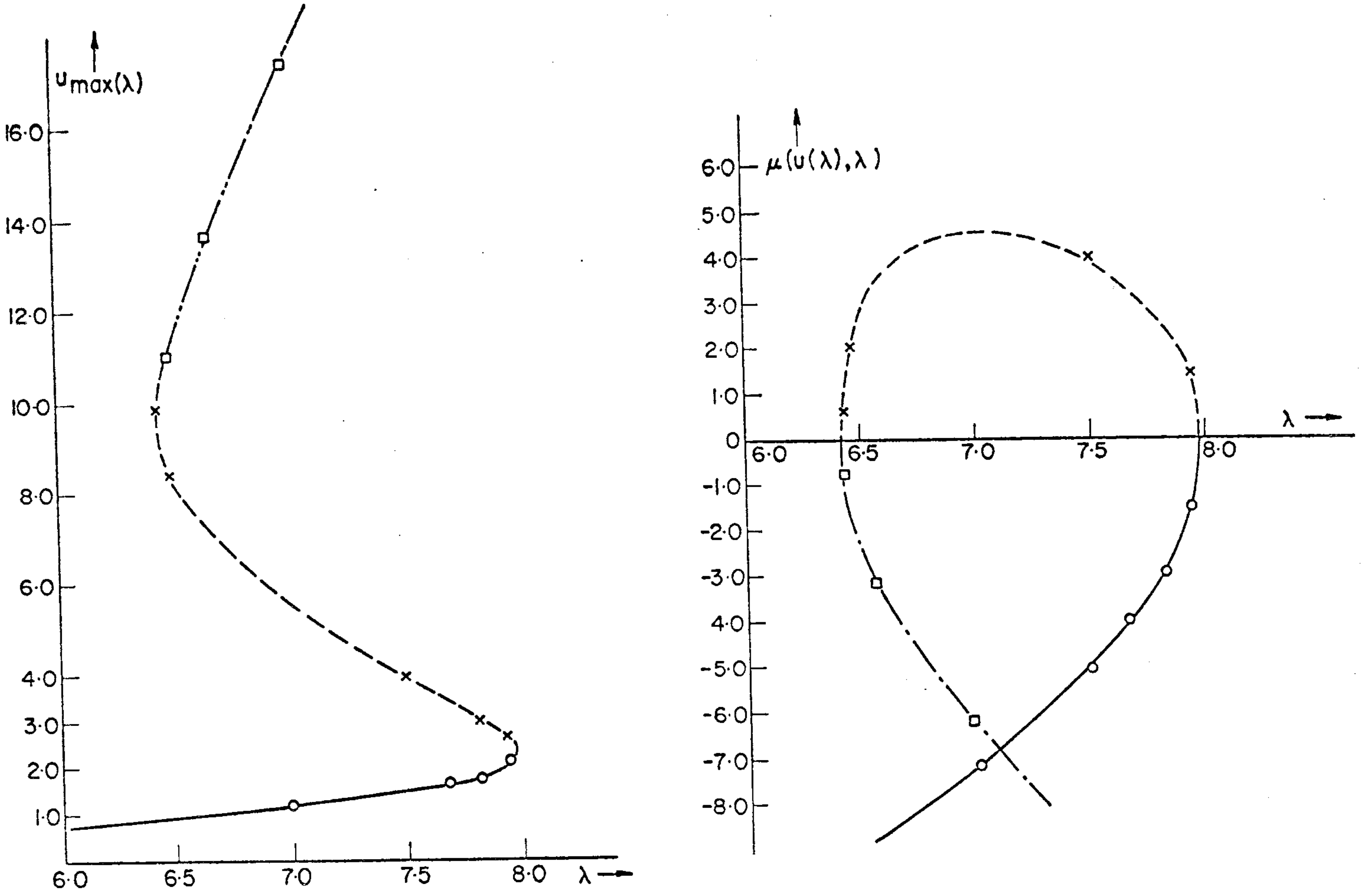

Figure 4

Left - Positive Solution Branch for $f(u) = 1 + (u + u^2/2)/(1 + u^2/100) - \lambda$ versus u max

Right - Variation of Critical Eigenvalue, $\mu(u(\lambda), \lambda)$, with λ

— — × — — unstable part of branch

———○——— minimal positive solution (stable)

-□— - - — upper·positive stable solution