

SOME GENERAL BIFURCATION TECHNIQUES*

RALPH BAKER KEARFOTT†

Abstract. The problem $H(y) = H(x, \lambda) = \theta$, where $H: \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ is considered. Numerical techniques for locating bifurcation points \bar{y} and for following arcs leading from \bar{y} are presented. These techniques are valid for primary and secondary bifurcation points, and at multiple bifurcation points, regardless of whether there is a change in the sign of the determinant or the Jacobi matrix H_x at \bar{y} ; they can also possibly be used when arcs intersect tangentially. The techniques do not require computation of second partial derivatives, although Jacobi matrices are computed using finite differences in neighborhoods of bifurcation points.

Details for incorporation into a derivative-free arc-following method, developed in a previous work, are given. Computational results for five test examples appear. Directions for further investigations and improvements are listed.

The stepsize control and Jacobi matrix update techniques may be improved for large, sparse problems, when first partial derivatives are easy to compute, or in the absence of bifurcation points.

Key words. arc-following method, bifurcation, numerical method, rank-1 updates, steplength algorithms

1. Introduction. We desire to quantitatively describe the solution set of

$$(1) \quad H(y) = H(x, \lambda) = \theta,$$

where $H: \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ and $y = (x, \lambda) \in \mathbb{R}^n \times \mathbb{R}$.

Solution sets of (1) usually consist of intersecting sets of codimension-1 manifolds in \mathbb{R}^{n+1} . Under sufficient smoothness and regularity assumptions, the solution set within the region $\{(x, \lambda), \|x\| \leq M, 0 \leq \lambda \leq 1\}$ consists of a finite number of smooth intersecting arcs. In that situation, the numerical problem can be resolved into the following components: (i) following the individual solution arcs of (1) accurately and efficiently; (ii) finding the bifurcation points (points where the arcs intersect); and (iii) successfully following one or more arcs away from a bifurcation point.

Much work on these problems has been done by H. B. Keller, W. C. Rheinboldt, and others. For a survey of arc-following techniques, see e.g. [2]. For a survey of techniques for bifurcation problems, see e.g. [16]. The reader may also consult [17], where application of simplicial methods to these problems is treated. Each of these surveys contains lists of further references.

Despite the richness of techniques and methods, further research is desirable. Improvements and further evaluation of both arc-following and bifurcation techniques are possible. Also, bifurcation techniques are of an ad hoc nature; primary, secondary and multiple bifurcation points are treated separately (cf. [16]); little work has been done in detecting "even order" bifurcation points (where an odd number of arcs intersect); and continuation away from a bifurcation point may fail when the arcs intersect tangentially. Other undesirable characteristics include the need for higher order derivatives or double iteration processes.

The purpose of this paper is to present some new and more general techniques for handling bifurcation problems. We implement the techniques with a derivative-free predictor-corrector-type arc-following method, which we review in § 2. We base

detection of each bifurcation point \bar{y} on analysis of the singular values of an approximate Jacobi matrix $H'(\bar{y})$; details are given in § 3.

We obtain directions for arcs emanating from \bar{y} by locating minima of $\|H(y)\|_2$ for y on the boundary of a small ball in the affine space containing \bar{y} and defined by the null space of an approximate $H'(\bar{y})$. This technique is most similar to that of H. B. Keller [14] in that direction vectors at the bifurcation point are given explicitly; it differs in the method of obtaining those directions. Also, Keller's directions are tangent to the arcs at \bar{y} , whereas our directions are approximately secant to the projections of the arcs onto the above affine space. Our approach and its implementation are detailed in § 4 below.

Results and analyses of computer runs are given in § 5. A summary, conclusions and directions for improvements are given in § 6.

2. The arc-following method. Since the basic arc-following method is explained in [11], only a general outline will be given here.

We assume that solution arcs of (1) can be parametrized with respect to arclength s , and we suppose $y_0 \in \mathbb{R}^n \times \mathbb{R}$ is such that $H(y_0) = \theta$. Then:

$$(2) \quad H'(y(s))y'(s) = \theta, \quad \|y'(s)\| = 1, \quad y(0) = y^{(0)}$$

may be integrated to find an arc $y(s)$ with $H(y(s)) = \theta$, where H' is the n by $n+1$ Jacobi matrix of H , and $y'(s)$ is the componentwise derivative of $y \in \mathbb{R}^{n+1}$ relative to s .

Our method is based on "predictor-corrector" techniques (cf. e.g. [1], [6], [15]). The predictor step corresponds to an implicit Euler step for (2) and is defined by:

$$(3) \quad z_0^{(k+1)} = y^{(k)} + \delta_k b^{(k)}$$

where $\delta_k \in \mathbb{R}$ is suitably small, $H(y^{(k)}) = \theta$, and $b^{(k)}, \|b^{(k)}\| = 1$ is approximately tangent to $y(s)$ at $y^{(k)}$. The next iterate $y^{(k+1)}$ is then obtained by solving:

$$(4) \quad G(z) = \begin{pmatrix} H(z) \\ b^{(k)T}(z - z_0^{(k+1)}) \end{pmatrix} = \theta$$

with a generalized secant method.

In [11] a derivative-free predictor-corrector method for (2) is given. There, H' is computed by updating with a least-change secant (Broyden) technique. Such techniques allow computation of a new approximate derivative matrix H' each time a new value of H is obtained, without any additional function or derivative evaluations. To assure that H' is approximately correct, Powell's idea of special correction updates is used ([19] and [11]); the effect is to supply a matrix H' which reflects, as a linear transformation, the action of the true H' in a region containing the past $2n+3$ points $z_j^{(i)}$ and $y^{(i)}$. (Powell corrections require, on the average, less than one additional function evaluation per step).

Stepsize algorithms for adaptively choosing δ_k are also given in [11], in addition to publications of several other researchers. In the absence of bifurcation points, such stepsize controls are usually based on keeping the number of corrector iterations required to solve (4) within bounds, or upon keeping the cosine of the angle between successive approximate tangent directions $b^{(k)}$ and $b^{(k+1)}$ within bounds; to do this, δ_{k+1} is set to δ_k , $2\delta_k$, or $\frac{1}{2}\delta_k$ as appropriate.

Numerical tests in [11] indicate that the predictor-corrector technique, combined with Broyden updates of H' , is competitive with general algorithms such as the Chow-Yorke algorithm in [25], provided one assumes evaluation of a Jacobi matrix from scratch would require n function evaluations. This fact has also been verified

* Received by the editors February 4, 1981 and in revised form March 29, 1982.

† Department of Mathematics and Statistics, The University of Southwestern Louisiana, Lafayette, Louisiana 70504.

by Kurt Georg [8], who independently investigated a similar predictor-corrector scheme with Broyden updates.

Our basic idea for handling bifurcation points, in addition to a framework for its implementation, are also given in [11]. Below, we expand, in refined form, these ideas; we then present the results of numerical tests and outline possible improvements.

3. Detection of the bifurcation points. If \bar{y} is a bifurcation point of (1), then the dimension of the null space $N(\bar{y})$ of $H'(\bar{y})$ must be greater than 1 (cf. [20]). In particular, the determinant $\det(H_x(x, \lambda))$ must vanish at the point $\bar{y} = (x^*, \lambda_*)$ with corresponding arclength s_* . Let $\{\tau_i(s)\}_{i=1}^{n+1}$ be the eigenvalues of the Jacobi matrix $G'(y(s))$, where G is as in (4). Then two or more $\tau_i(s)$ must vanish at $s = s^*$.

If an odd number of $\tau_i(s)$ cross the axis $\tau = 0$ at $s = s^*$, then $\det(G'(y(s)))$ must change sign at $s = s^*$. Some techniques for locating bifurcation points are based on detection of such sign changes (see the references in the surveys) or on detection of changes in a topological degree (cf. e.g. [10]). Here, we deal with the more difficult problem of detecting all bifurcation points.

In general, any quantity $q(s)$ which is a continuous function of arclength s and which equals zero at values of s for which $H'(y(s))$ is rank deficient may be used to detect bifurcation points. In the initial experiments presented here, we elect to examine the reciprocal of the condition number of H' relative to the 2-norm, expressed as the ratio of the smallest nonzero singular value $\sigma_n(s)$ to the largest nonzero singular value $\sigma_1(s)$ of $H'(y(s))$. To obtain this quantity, we actually did a singular value decomposition of the approximate $H'(y^{(k)})$; though not optimal from the point of view of overhead, this stable technique gave us valuable additional information concerning the behavior of our algorithms and test functions.

Bifurcation points are found by detecting minima of $\sigma_n(s)/\sigma_1(s)$ (or, more generally, minima of $|q(s)|$). Besides having the properties listed in § 2 of this paper, our predictor stepsize control should decrease δ_k before encountering a bifurcation point, in such a manner that the position of each bifurcation point may be efficiently bracketed and refined. If exact derivative matrices $H'(y^{(k)})$ are assumed, linear convergence to a bifurcation point can be achieved by allowing $|q(s_{k+1})| - |q(s_k)| > -\alpha q(s_k)$, for some $0 < \alpha < 1$, until $|q(s_k)|$ reaches some threshold value, after which the step δ_k is unchanged until a minimum of $|q(s)|$ is bracketed.

As was mentioned in § 2, the matrices H'_k , and hence the $q(s_k)$, formed from Broyden updates with Powell correction steps are not exact Jacobi matrices at $y^{(k)}$, but depend upon the action of the exact linear transformations $H'(y)$ in a region containing the past $2n + 3$ corrector iterates and points $y^{(i)}$. In our experiments, we handled this problem by setting δ_{k+1} to be a small fraction of δ_k whenever the average decrease in $|q(s)|$ over the past $2n + 3$ points $y^{(i)}$ exceeded $\alpha |q(s_k)|$; δ_{k+1} was not allowed to be greater than δ_k except after a cycle of $2n + 3$ predictor steps over which δ_k was not decreased due to $q(s)$.

The above step control in the neighborhood of bifurcation points is inefficient when n is large, since $2n + 3$ times as many points $y^{(k)}$ are computed as in the case where "exact" H' are available. A possible improvement, in later computations, would be to discard the Powell update procedure, use Broyden (or other quasi-Newton) updates only for the corrector steps, and compute H' using finite differences at each $y^{(k)}$. The H' could then be considered exact, with the step control threshold set according to the accuracy of the differencing scheme. (Of course, exact H' would be even better, if they are easy to obtain).

If we assume $H'_k = H'(y^{(k)})$ is an exact Jacobi matrix, minima of $q(s)$ are bracketed whenever $|q(s_{k-1})| < |q(s_{k-2})|$ and $|q(s_{k-1})| < |q(s_k)|$. If H'_k is the result of applying

quasi-Newton updates and Powell correction steps, the computed $q(s_k)$ will vary irregularly near a minimum of the actual $q(s)$. In our experiments, we handled this by comparing average values of $|q(s_k)|$ over the first third, middle third, and last third of the past $2n + 5$ iterates. As in stepsize control, this method is cumbersome and undesirable for large n , but worked reliably in our experiments.

Once the minimum has been bracketed, it may be refined using successive quadratic or linear interpolation, etc. Standard techniques may be used, but linear combinations of previous approximate singular points \bar{y} are taken to produce new $\bar{y} = y(\bar{s})$; in this process, each new \bar{y} is corrected via formula (4) in order to lie on an arc $H(y(s)) = \theta$, until the condition of H' renders this inadvisable. In our experiments, relatively exact H' obtained from finite differences were used during such a "line search" (over arclength).

In our experiments, the smallest singular value $\sigma_n(s)$ typically had an absolute-value-type singularity in its derivative at points \bar{s} where $\sigma_n(\bar{s}) = 0$. Since we implemented a hybrid quadratic interpolation, linear interpolation line search, refinement of the point \bar{y} typically consisted of successive linear interpolation.

Occasionally, minima of $|q(s)|$ not corresponding to $q(s) = 0$ are computed. This must occur since we do not assume $q(s)$ changes sign at singular points $\bar{y} = y(\bar{s})$. These may be eliminated in the stepsize control (setting of δ_k), in the line search, or after each \bar{y} is located. In our experiments, we chose the conservative method of analyzing the null space of each $H'(\bar{y})$ once \bar{y} has been refined.

It is desirable to compute the singular points \bar{y} accurately. This is because minima of $\|H\|$ over a low-dimensional manifold are computed to obtain tangent directions leading from \bar{y} , and distinct minima are relatively close together when \bar{y} is given inaccurately. (See the discussion in the next section and Fig. 1.) Thus, the tolerance in the line search is set according to the accuracy to which $q(s)$ is being computed.

4. Determining arc directions at a bifurcation point. The ansatz and basic structure of the algorithm employed appear in the introduction and in [11, § 4]. Here, we give a summary, more details, and improvements.

We assume that the component arcs $y(s) \subset H^{-1}(\theta)$ are smooth functions of arclength s ; then any such arcs passing through a bifurcation point \bar{y} must have tangent vectors at $y = \bar{y}$ in the null space of $H'(\bar{y})$. Suppose $\{v^{(1)}, \dots, v^{(k)}\}$ is an orthonormal basis for this null space, and let Π be the affine space given by $\Pi = \{\bar{y} + \sum_{j=1}^k \alpha_j v^{(j)} \mid \alpha_j \in \mathbb{R}\}$. Let $\mathcal{A} \subset \Pi$ be a small region containing \bar{y} in its interior, and let $\{m^{(1)}, m^{(2)}, \dots, m^{(q)}\}$ be the locations of the minima of $\|H\| \mid \partial \mathcal{R}$. Then direction vectors for arcs intersecting at \bar{y} are given by a subset of $\{(m^{(i)} - \bar{y}) / \|m^{(i)} - \bar{y}\| \mid 1 \leq i \leq q\}$.

To proceed, three tasks must be undertaken: (i) determination of an orthonormal basis for the null space of $H'(\bar{y})$; (ii) choice of a size and shape for the region \mathcal{R} ; and (iii) choice of and execution of a method for finding the minima $m^{(i)}$ of $\|H\| \mid \partial \mathcal{R}$.

In the experiments reported here, we form a basis for the null space of $H'(\bar{y})$ from the right singular vectors of $H'(\bar{y})$ corresponding to singular values equal to zero. Less general techniques and exploitation of special structure in H' can improve the efficiency of this computation.

The region \mathcal{R} was chosen here to be a k -ball S_δ of radius δ centered at \bar{y} , so that $\partial \mathcal{R}$ is without boundary and an unconstrained minimizer can be used. We parametrize ∂S_δ in terms of the standard spherical angular coordinates: $\{(\phi_1, \dots, \phi_{k-2}, \psi) \mid -\pi/2 \leq \phi_i \leq \pi/2, 1 \leq i \leq k-2 \text{ and } -\pi \leq \psi \leq \pi\}$. The region $[-\pi/2, \pi/2]^{k-2} \times [-\pi, \pi]$ was divided into $(2p)^{k-1}$ subregions, defined by dividing

each of the spherical coordinate intervals into $2p$ subintervals. The unconstrained minimizer was then applied $(2p)^{k-1}$ times, with starts in each subregion.

Choice of the radius δ requires some thought for numerically difficult problems. If $y^{(i)} = \bar{y}$ is a bifurcation point, then approximations $z^{(i+1)}$ to the next iterate $y^{(i+1)}$ will be given by:

$$z^{(i+1)} = m^{(i)} = y^{(i)} + (m^{(i)} - y^{(i)})$$

for some i , $1 \leq i \leq q$. Thus, it is important that $H'(m^{(i)})$ be of rank n to within the computational precision of H' . Define:

$$\mathcal{M}_j = \{y \in \mathbb{R}^n \times [0, 1] \mid \sigma_j(y) = 0\}$$

for $1 \leq j \leq n$, where $\sigma_j(y)$ is the j th singular value of $H'(y)$. Then, under appropriate regularity conditions, \mathcal{M}_j is a finite collection of codimension-1 manifolds in $\mathbb{R}^n \times [0, 1]$, and $\bar{y} \in \mathcal{M}_j$, $n - k + 1 \leq j \leq n$. Also, if there are q' arcs in $H^{-1}(\theta)$ intersecting at \bar{y} , define $\mathcal{A}_{j'}$, $1 \leq j' \leq q'$, to be the projection of the j' th such arc onto the affine space Π corresponding to the null space of $H'(\bar{y})$. Then $H'(m^{(i)})$ can be made nonsingular by increasing δ provided the $\mathcal{A}_{j'}$ and the \mathcal{M}_j intersect sufficiently transversally at \bar{y} .

In our experiments, the following test was used to decide nonsingularity of H' . Suppose the machine can represent m decimal digits, and suppose H is computed by differences with relative stepsize equal to 10^{-5} , so that H is accurate to approximately $m - s$ digits. Then, if H' is computed with forward differences with optimal stepsize, H' will be accurate to approximately $(m - s)/2$ decimal digits. Let β be the number of decimal digits accuracy in σ_n , where σ_j is the j th singular value of H' , $1 \leq j \leq n$. then, since $\|H'\|_2 = \sigma_1$, we have

$$\beta \doteq (m - s)/2 - \log_{10}(\sigma_1/\sigma_n) = (m - s)/2 - L - l,$$

where $L = \log_{10} \sigma_1$ and $l = -\log_{10} \sigma_n$ (cf. e.g. [22, p. 321]). Since H' is singular when $\sigma_n = 0$, H' will be singular to within the computational precision whenever $l \geq \beta$. Thus, H' is considered singular whenever:

$$(5) \quad l \geq (m - s)/4 - L/2.$$

As an additional consideration in choosing δ large, all arcs emanating from \bar{y} can be detected only when \bar{y} is in the interior of S_δ ; this will not be the case if \bar{y} is given only approximately and δ is too small; numerically, \bar{y} relatively near ∂S_δ will make distinct arcs more difficult to detect (see Fig. 1).

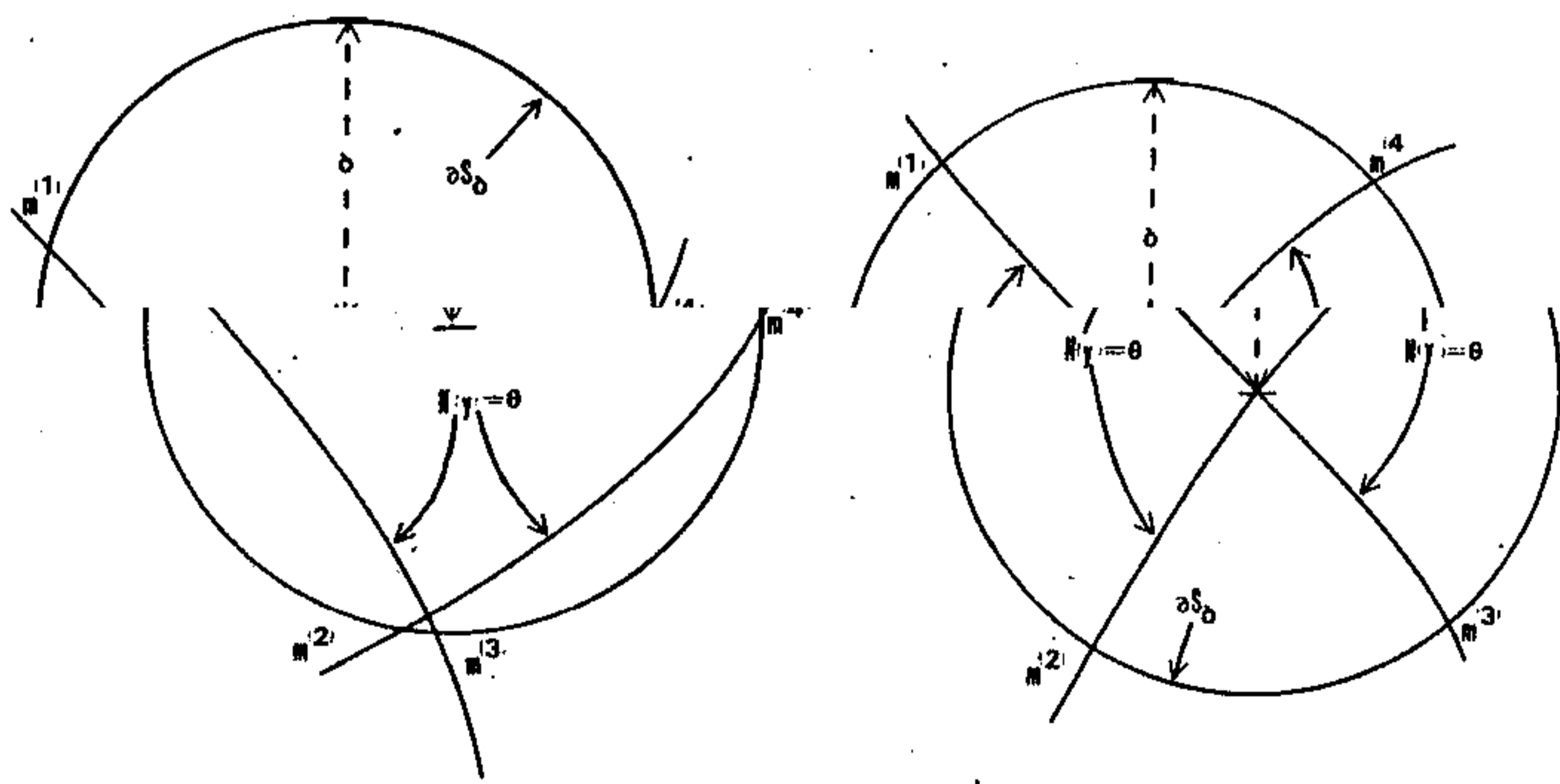


FIG. 1. Minima of $\|H\|$ on ∂S_δ are easier to distinguish when the center of S_δ is near the bifurcation point.

On the other hand, if δ is chosen too large, the arcs intersecting at \bar{y} do not lie approximately in Π . This can cause the surface of $\|H\|$ to be "flatter" near the $m^{(i)}$ and can also cause corrector iterations starting with $m^{(i)}$ to fail. In view of this, the smallest δ consistent with easy detection of the $m^{(i)}$ and with $H'(m^{(i)})$ being nonsingular should be used.

In our experiments, we initially set δ equal to the maximum of the minimum allowable predictor stepsize and $\sqrt{n+1}$ times ϵ_f , where ϵ_f is the relative accuracy to which \bar{y} has been located. As the $m^{(i)}$ were computed, inequality (5) was checked; if (5) was false for any i , δ was replaced by the minimum of $M\delta$ and the maximum allowable predictor stepsize. After increasing δ , the minimization process was restarted with $i = 1$. We mention that (5) and this process are very conservative, and improvements in efficiency are possible.

We have found the simplex method of Nelder and Mead [18] suitable for direct location of the $m^{(i)}$. A descent method similar to the method of steepest descent, the simplex method of Nelder and Mead begins with a starting simplex; this simplex is then changed one vertex at a time by "reflection", "expansion", and "contraction". The iteration ceases when the diameter of the resulting simplex has become smaller than some tolerance ϵ_d .

The values of the minimization search accuracy ϵ_d which are reasonable to demand depend upon which δ is selected, and both ϵ_d and δ depend on the machine accuracy ϵ_m and the order of accuracy in the values of H . It can be shown that a tolerance of ϵ_d in the $(k-1)$ -dimensional parameter space corresponds roughly to a distance of $\sqrt{k-1}\pi\delta\epsilon_d$ in \mathbb{R}^{n+1} . Since the relative accuracy of coordinates of $m^{(i)} \in \mathbb{R}^{n+1}$ cannot exceed the machine epsilon ϵ_m , it is prudent to have:

$$(6) \quad \epsilon_d > \frac{\|y\| \epsilon_m}{\sqrt{k-1}\pi\delta}$$

In practice, ϵ_d is chosen somewhat larger than the right-hand side of (6), since the computational precision is somewhat larger than ϵ_m and also since the minimization is less costly with larger stopping tolerances. In particular, ϵ_m in (6) may be replaced by a fraction of the arc-following tolerance. In the tests, ϵ_d was automatically doubled every time a location $m^{(i)}$ was found such that the scaled magnitude of $\|H(m^{(i)})\|$ was less than .1 times the corresponding tolerance in the arc-following method, and was decreased proportionally whenever δ was increased.

5. Numerical experiments. Here, we present results from several test examples; our goal is to demonstrate feasibility of the techniques as alternative approaches, to give an idea of their versatility, to describe the algorithms' behavior and to point out possible difficulties.

Five examples were treated. All involve special H of the form

$$(7) \quad H(x, \lambda) = \lambda g(x) + (1 - \lambda)g^{(0)}(x),$$

where $g: \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $g^{(0)}: \mathbb{R}^n \rightarrow \mathbb{R}^n$. The continuation method was begun at $(\theta, 0)$, and continuation along each individual arc was halted when either the $\lambda = 1$ hyperplane or $\lambda = 0$ hyperplane was reencountered. The goal of the algorithms was to detect all bifurcation points and follow all arcs.

In the first four examples,

$$(8) \quad g(x) = Ax - f(x),$$

where A is the matrix corresponding to discretization of the boundary value problem

$$(9) \quad -u'' = 0; u(0) = u(1) = 0,$$

with central differences and n internal meshpoints, and the i th component of f is given by

$$f_i(x) = x_i^3.$$

Examples 1, 2 and 3 correspond to $g(x)$ as in (8), with $g^{(0)}$ defined by: $g_i^{(0)}(x) = -x_i$, and with $n = 2, 4$ and 7 , respectively. In Example 4, we took $n = 3$ and we took g as in (8), but with $g^{(0)}(x) = -Ax$. In the final example, we took $n = 2$, we took $g^{(0)}$ to be the identity function, and we took the components of g to be

$$g_1(x) = 2x_1(x_1^2 + x_2^2) - .5x_1, \quad g_2(x) = x_2(x_1^2 + x_2^2) - .5x_2.$$

(A variant of this appears in [23] in an example of a multiple bifurcation point.)

For the first three examples, it has been shown in [10] that there are exactly n primary bifurcation points, occurring at

$$(10) \quad \lambda_i = 1 - m_i / (1 + m_i), \quad i = 1, 2, \dots, n.$$

Here, m_i is the i th eigenvalue of the matrix A and is given by

$$(11) \quad m_i = 2(n+1)^2 [1 + \cos(i\pi/(n+1))]$$

(cf. [9]). In these examples, the primary bifurcation points are all simple, and the arcs intersect transversally. Furthermore, it can be shown that $\sigma_n(s)$ is exactly linear immediately to the left and immediately to the right of each primary bifurcation point, where s is arclength. Thus, the successive linear interpolation scheme (cf. § 3) for refinement of \bar{y} will give, in theory, exact bifurcation points \bar{y} , without necessity of line searches. However, the primary bifurcation points are spaced unevenly and are very close together for n large (cf. formulas (10) and (11)). Thus, the stepsize control scheme near bifurcation points is tested.

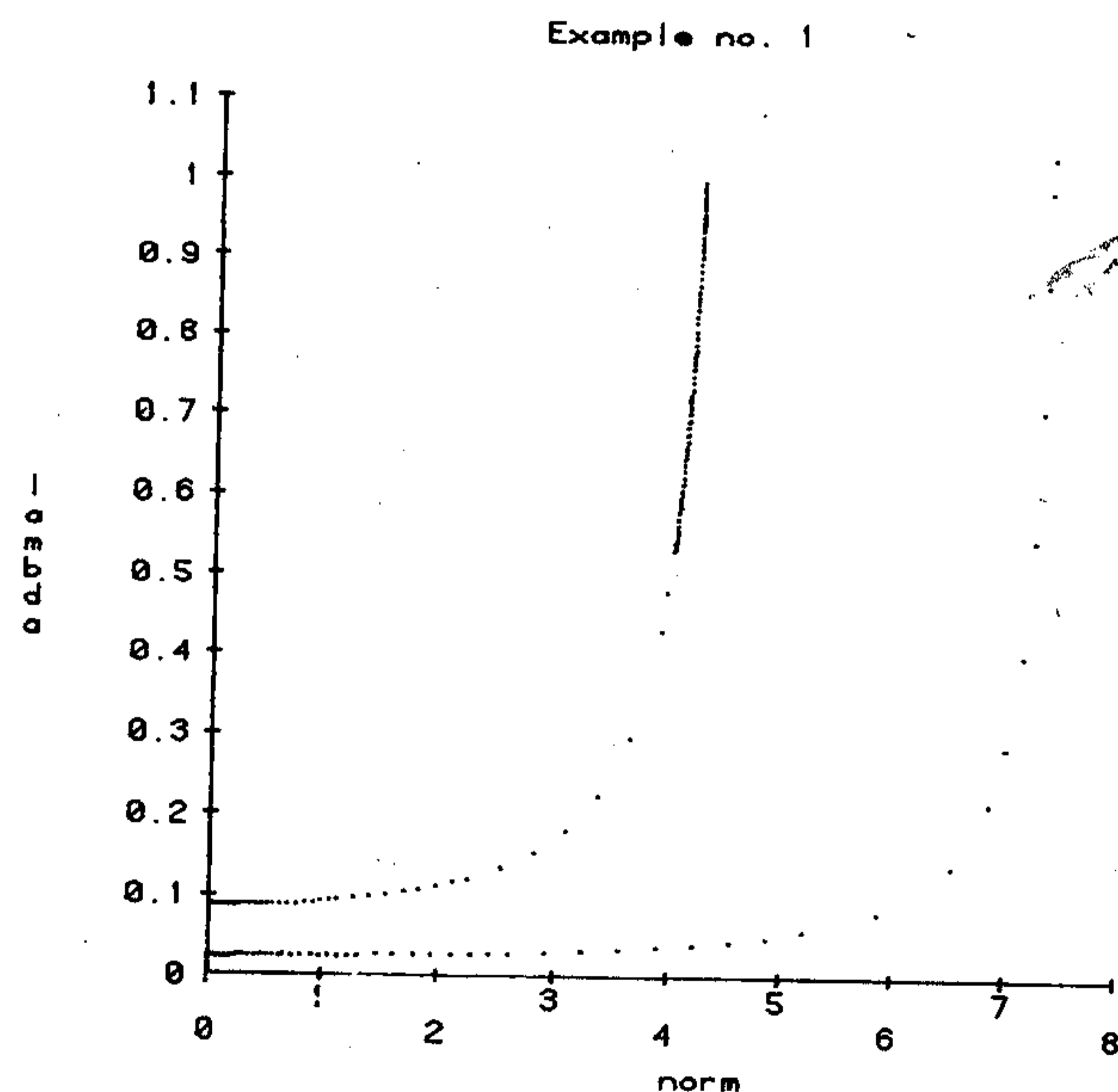


FIG. 2. Example 1.

In the fourth example, a single multiple bifurcation point occurs at $\lambda = \frac{1}{2}$ (cf. [10]), where H' equals the zero matrix, and four arcs intersect. Thus, it poses a test for the direction-finding algorithm in § 4.

In the fifth example, the unique primary bifurcation point for $\lambda \in [0, 1]$ occurs at $\lambda = \frac{2}{3}$. There, H' is the zero matrix and three arcs intersect.

The actual computations for Example 1 were straightforward, with no encountered difficulties. The norm and first coordinate of actual iterates are plotted with respect to λ in Figs. 2 and 3, respectively. Note that the step size on the arcs intersecting at λ_2 becomes small; this is because $\sigma_n(s)$ is decreasing and H' is singular on these arcs at $\lambda = 1$.

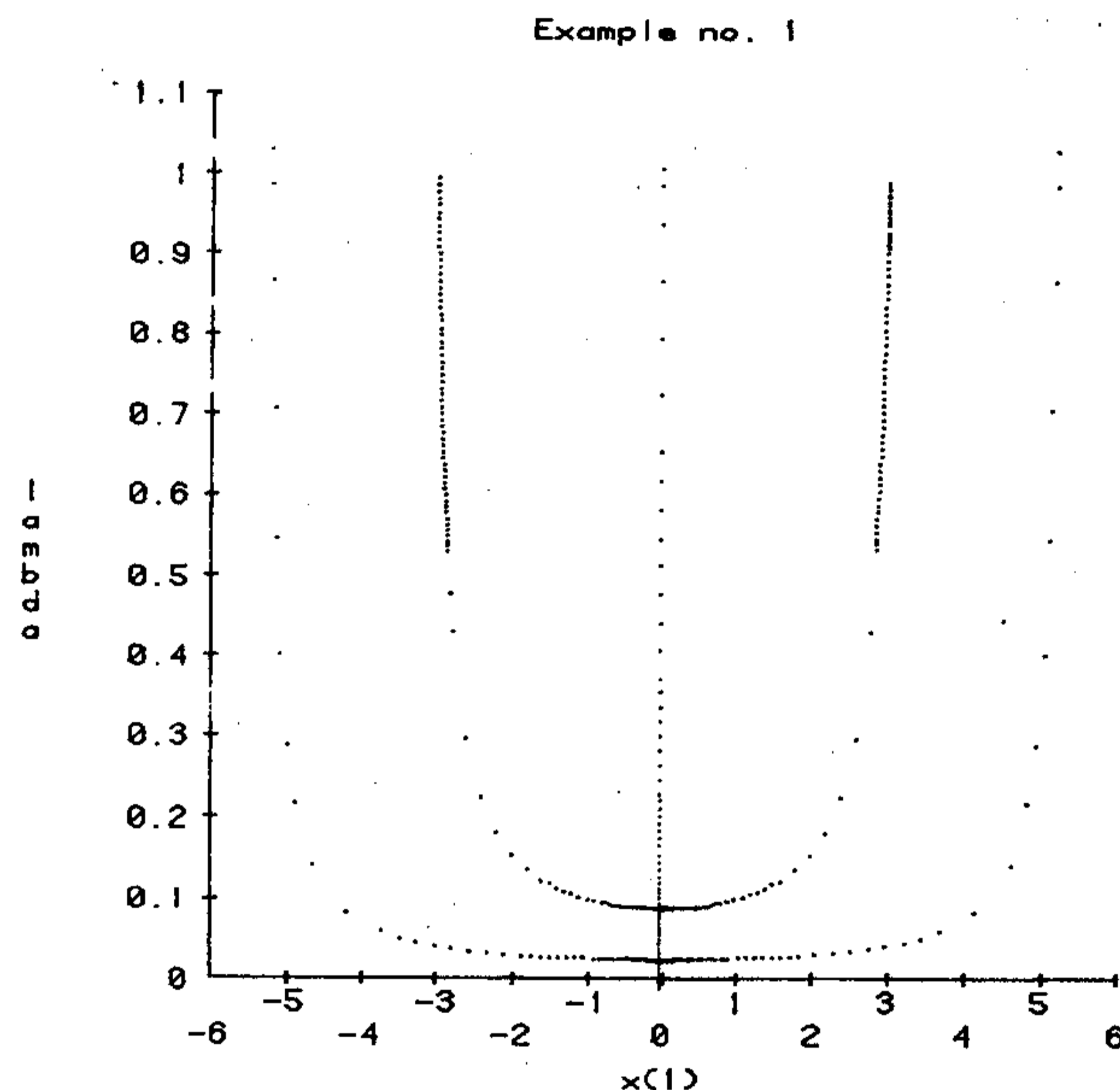


FIG. 3. Example 1.

The algorithms also performed satisfactorily for Example 2. However, secondary bifurcations occurred on the arcs intersecting at λ_2 . Also, minima of $\sigma_n(s)$ not corresponding to bifurcation points were detected on the arcs intersecting at λ_3 . See Figs. 4, 5 and 6 for graphs of the iterates.

Example 3 was straightforward, but the algorithms performed a sizable amount of computation. Numerous minima of σ_n not corresponding to bifurcation points were found.

The fourth example was by far the most difficult. The matrix H' was nearly singular in a large region, so proper automatic selection of the radius δ for S_δ was important. The hypersurface $\|H\|(y)$ seemed, in addition, to have numerous "wrinkles" near the bifurcation point; though actual tangent directions were computed accurately, the minimization gave numerous $m^{(i)}$ not corresponding to arcs $\|H\| = \theta$. These "imposters" were promptly detected when corrector iteration in the arc-following method failed, but a fine grid (i.e. p large and a large number of starts in the

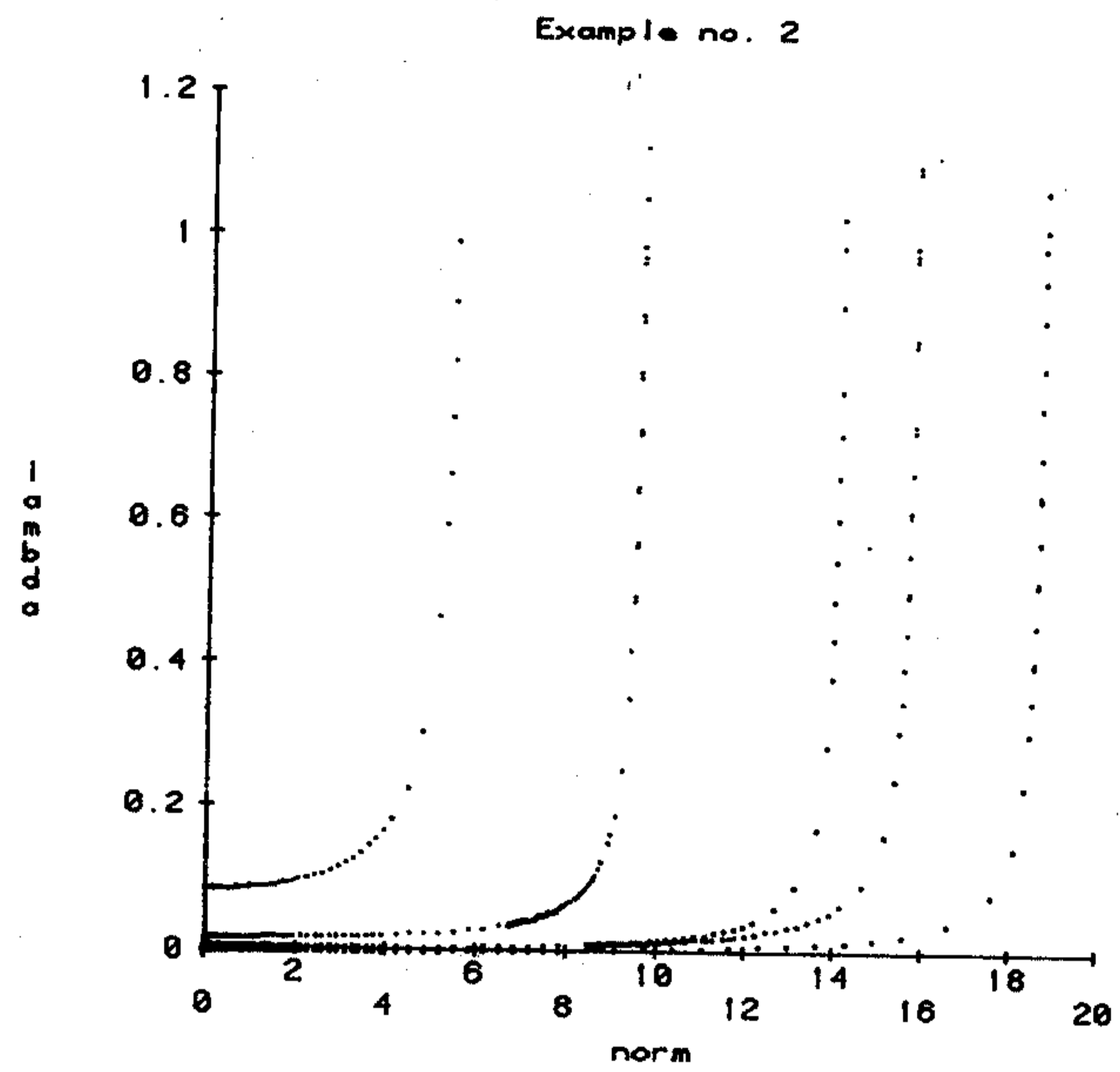


FIG. 4. Example 2.

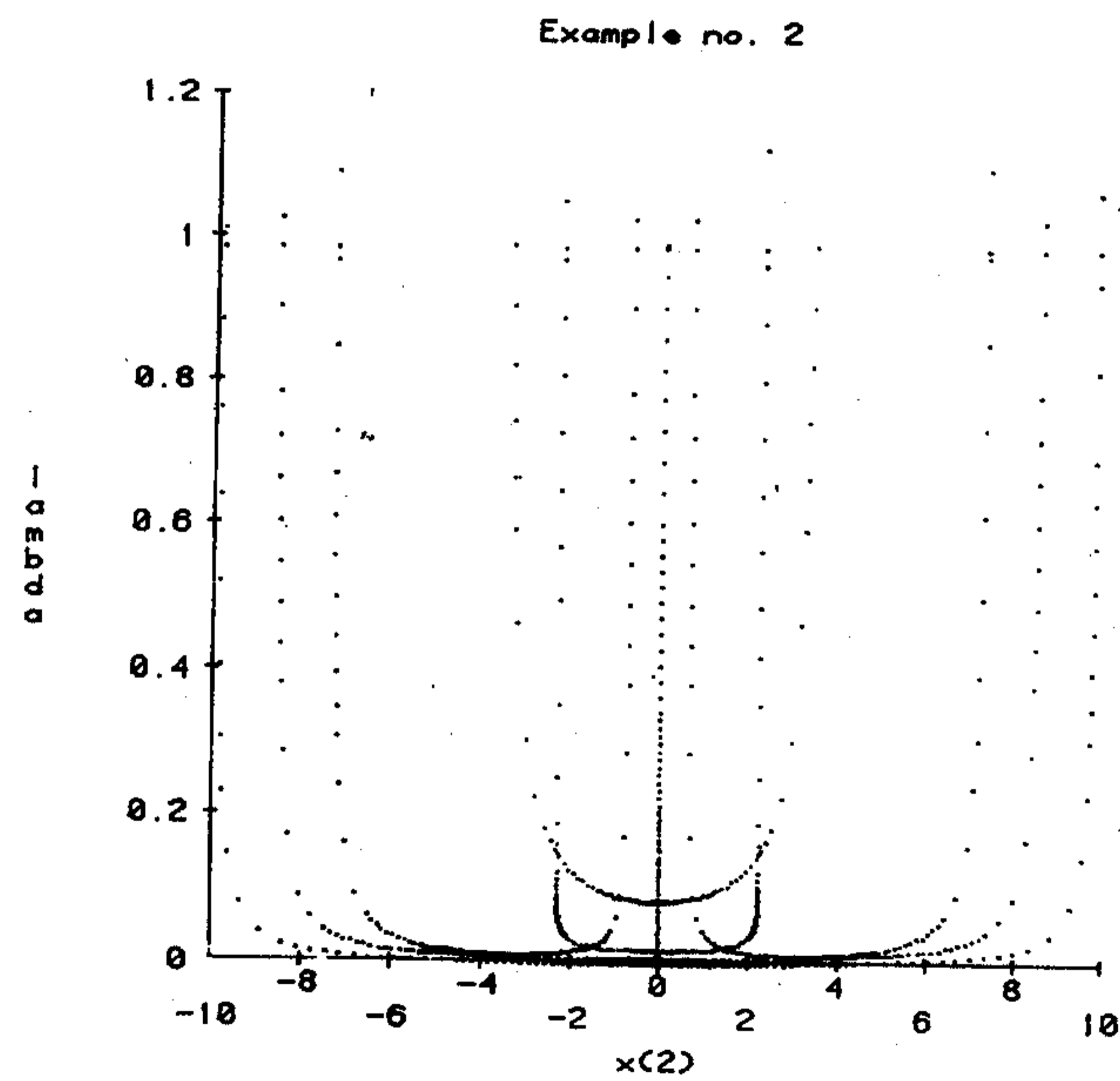


FIG. 5. Example 2.

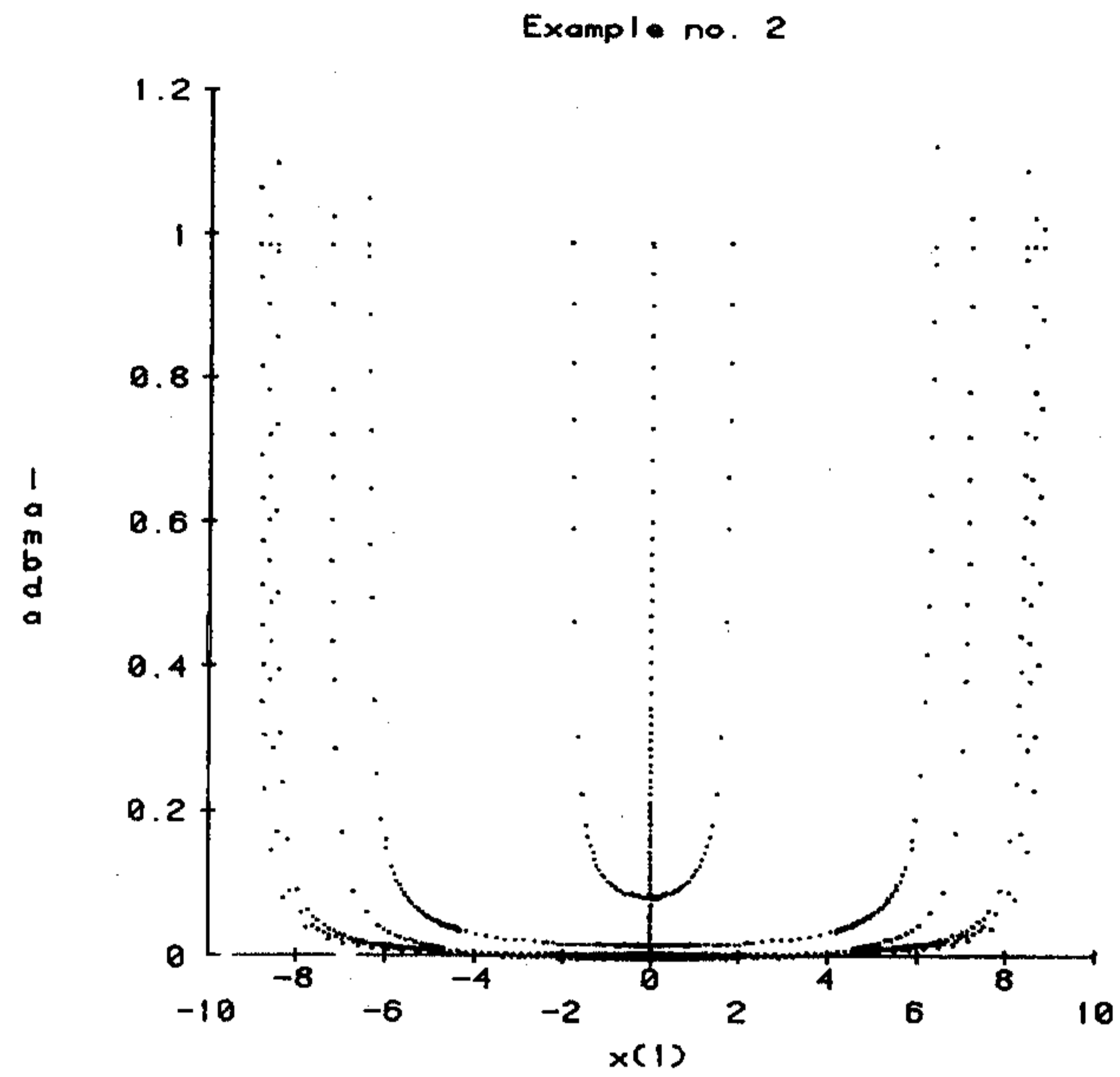


FIG. 6. Example 2.

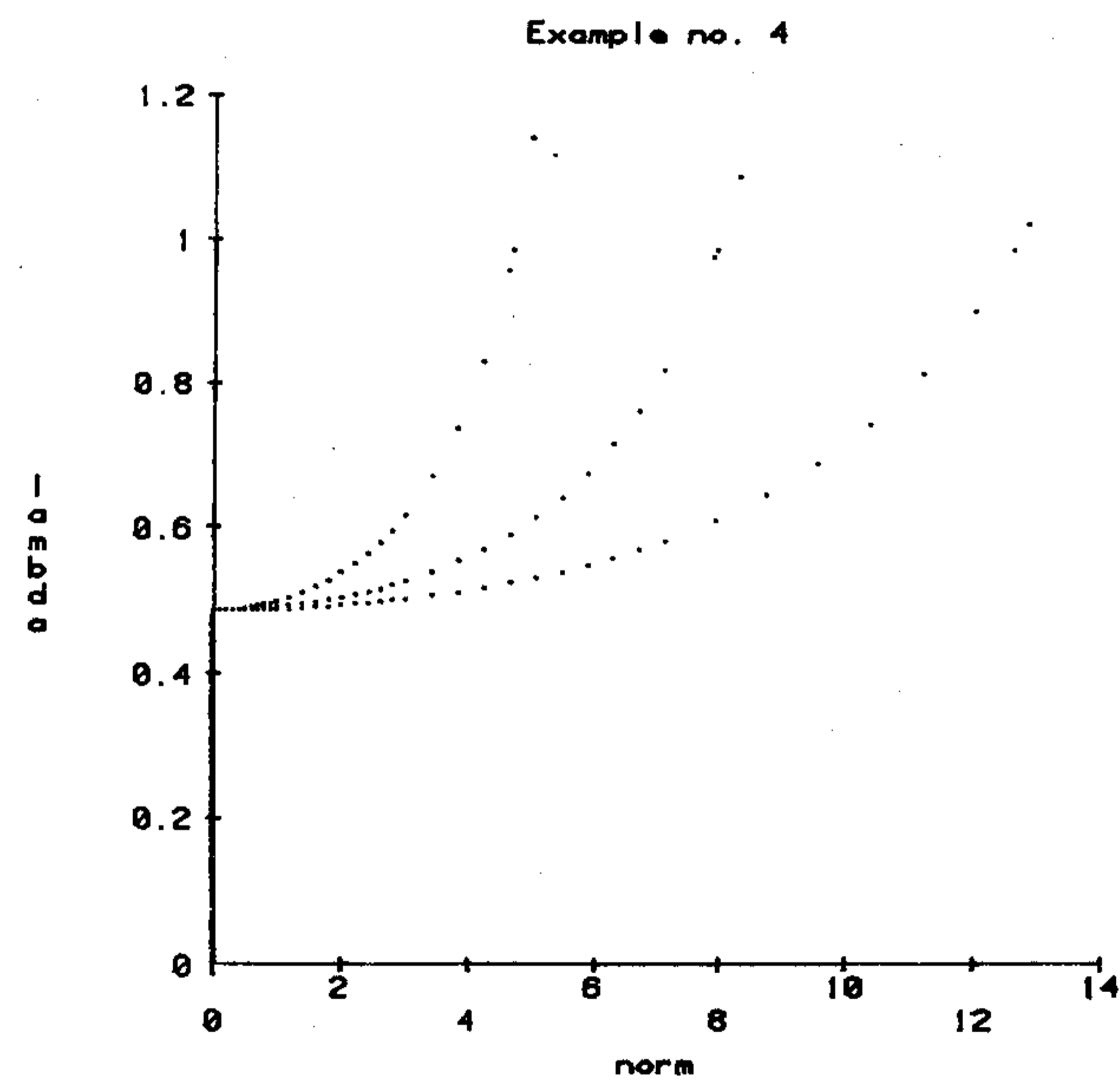


FIG. 7. Example 4.

Example no. 4

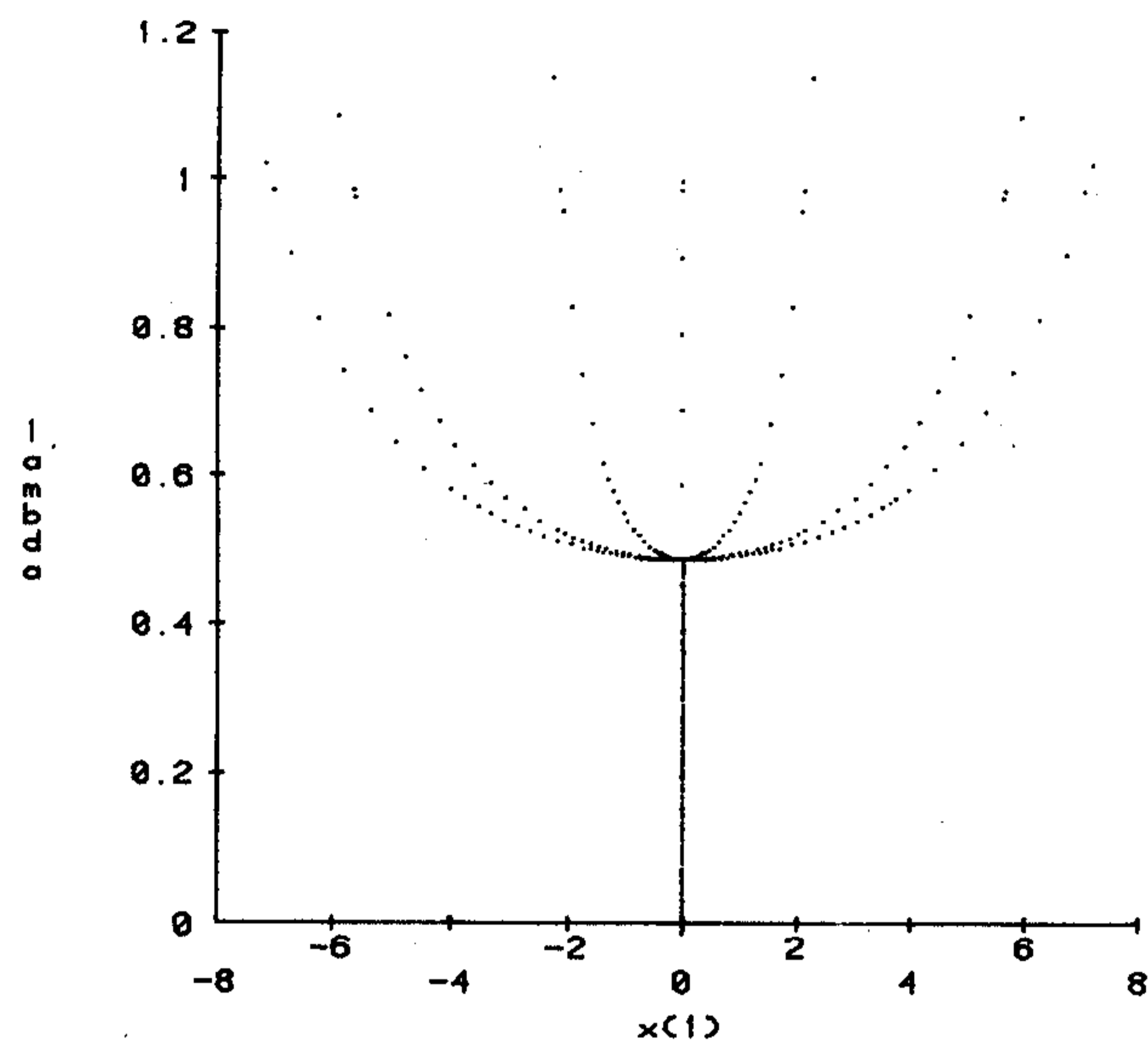


FIG. 8. Example 4.

minimization process) was required. With $p = 3$ (216 minimizations in 3-space), all 7 arcs proceeding away from the bifurcation point were found, but 14 false directions were also given. Plots are given in Figs. 7, 8 and 9.

The fifth example was perhaps the most straightforward. Here, there is no change in sign of the determinant of H_x at the bifurcation point on the trivial branch, since both eigenvalues pass from positive to negative; thus, the special capabilities of the stepsize control and bifurcation point refinement algorithms mentioned in § 3 are demonstrated (Figs. 10, 11).

As was indicated, the examples reveal behavior of the following facets of the algorithms: (i) the efficiency and reliability of deceleration and the "line search" when the arc-following method approaches and refines the bifurcation point; (ii) the efficiency and reliability with which directions for arcs intersecting at bifurcation points are computed, (iii) efficiency and reliability of the acceleration/deceleration scheme between and past bifurcation points. Performance data with regard to these qualities is given for Examples 1 through 5 in Tables 1 through 5, respectively.

The first part of each table deals with quality (iii): In the first column, the coordinates of the intersection of the arc in question with the $\lambda = 1$ hyperplane are given, while the coordinates of the last bifurcation point on this arc are given in the second column (BP). The total number of function evaluations required to follow the arc between these two points is given in the third column (FE). We note that this phase of the algorithms is extremely reliable; the discussion in [11] is for the most part valid, although the steps are increased somewhat more slowly here to assure that bifurcation points are not missed.

Example no. 4

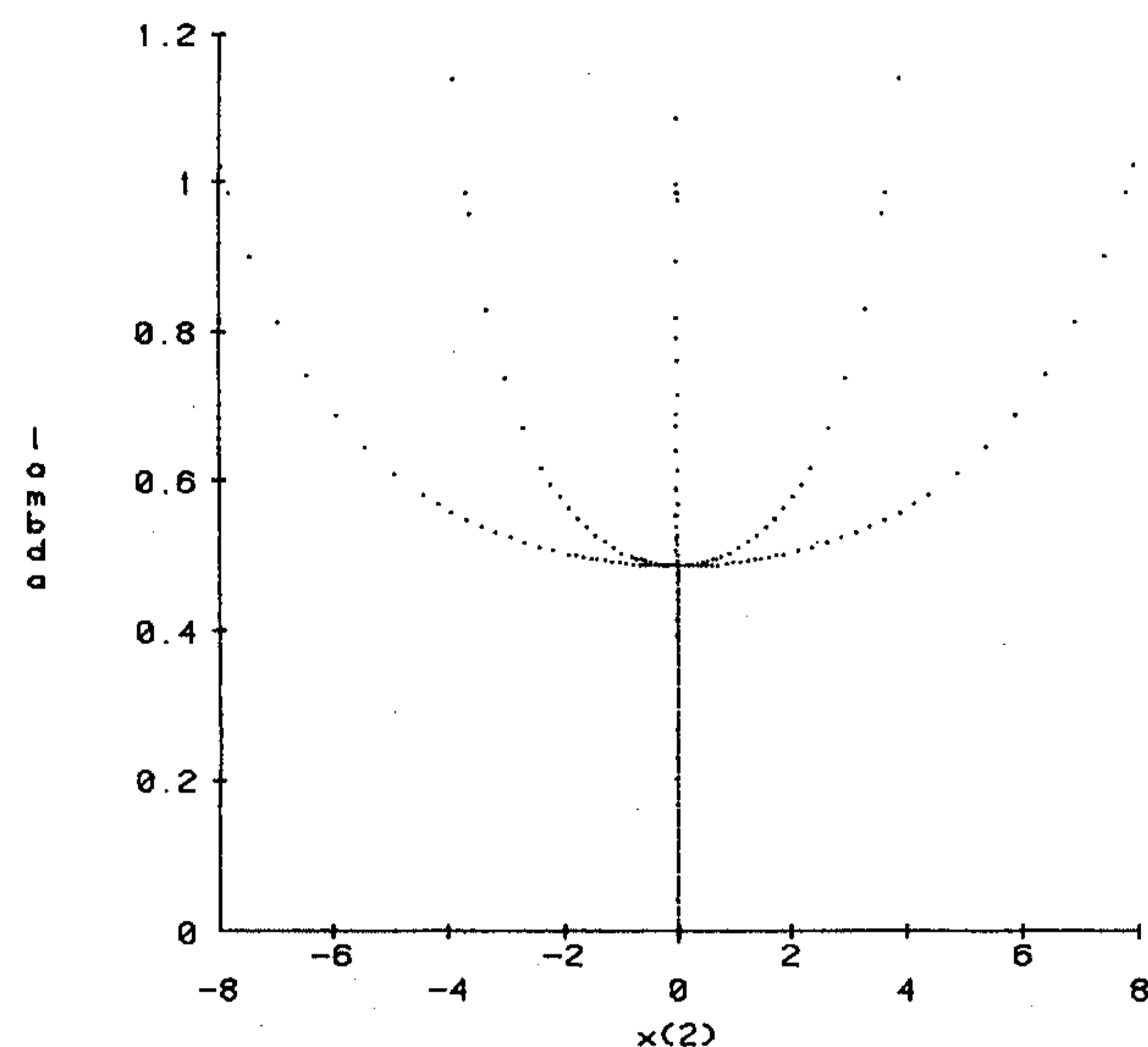


FIG. 9. Example 4.

Example no. 5

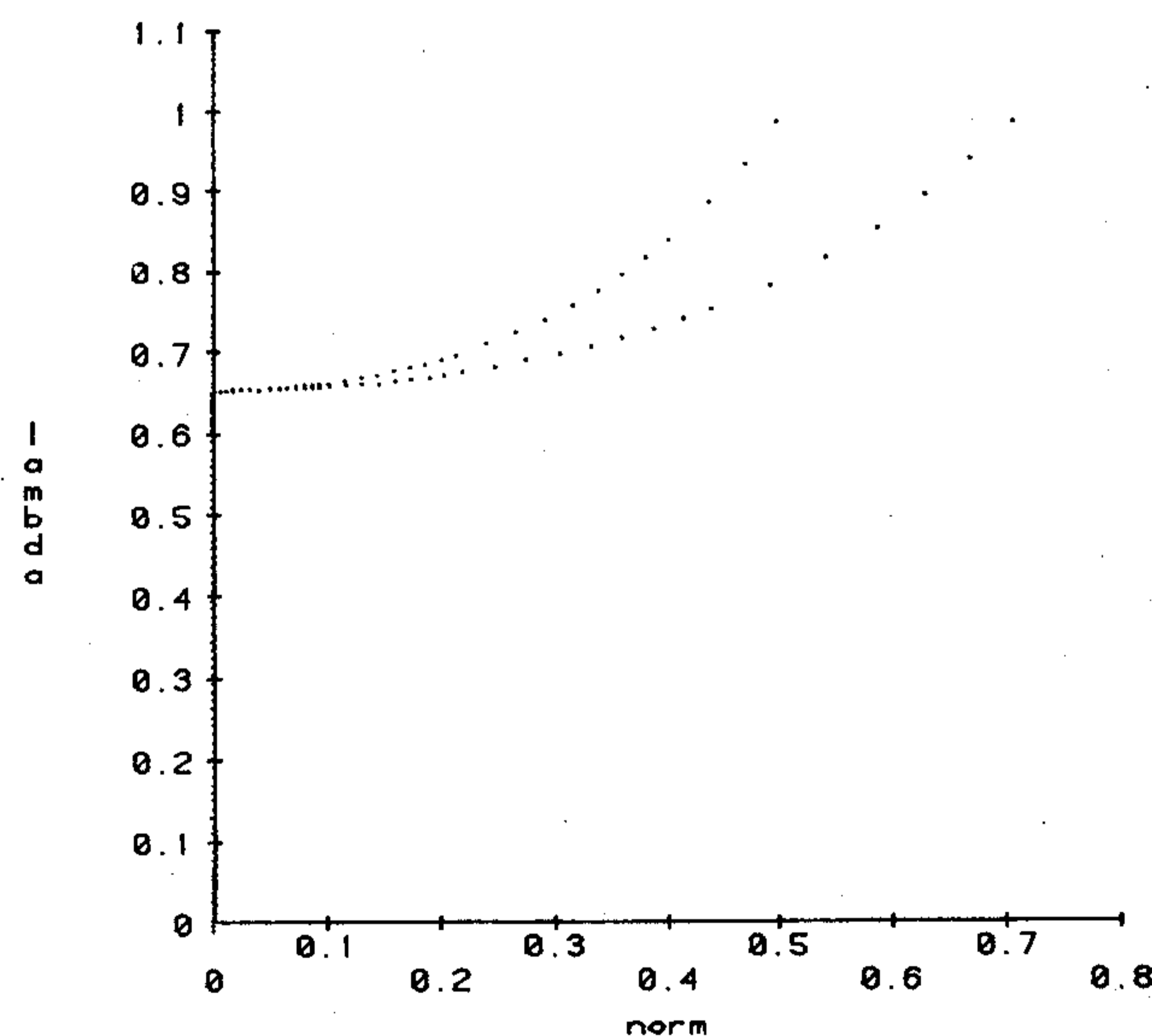


FIG. 10. Example 5.

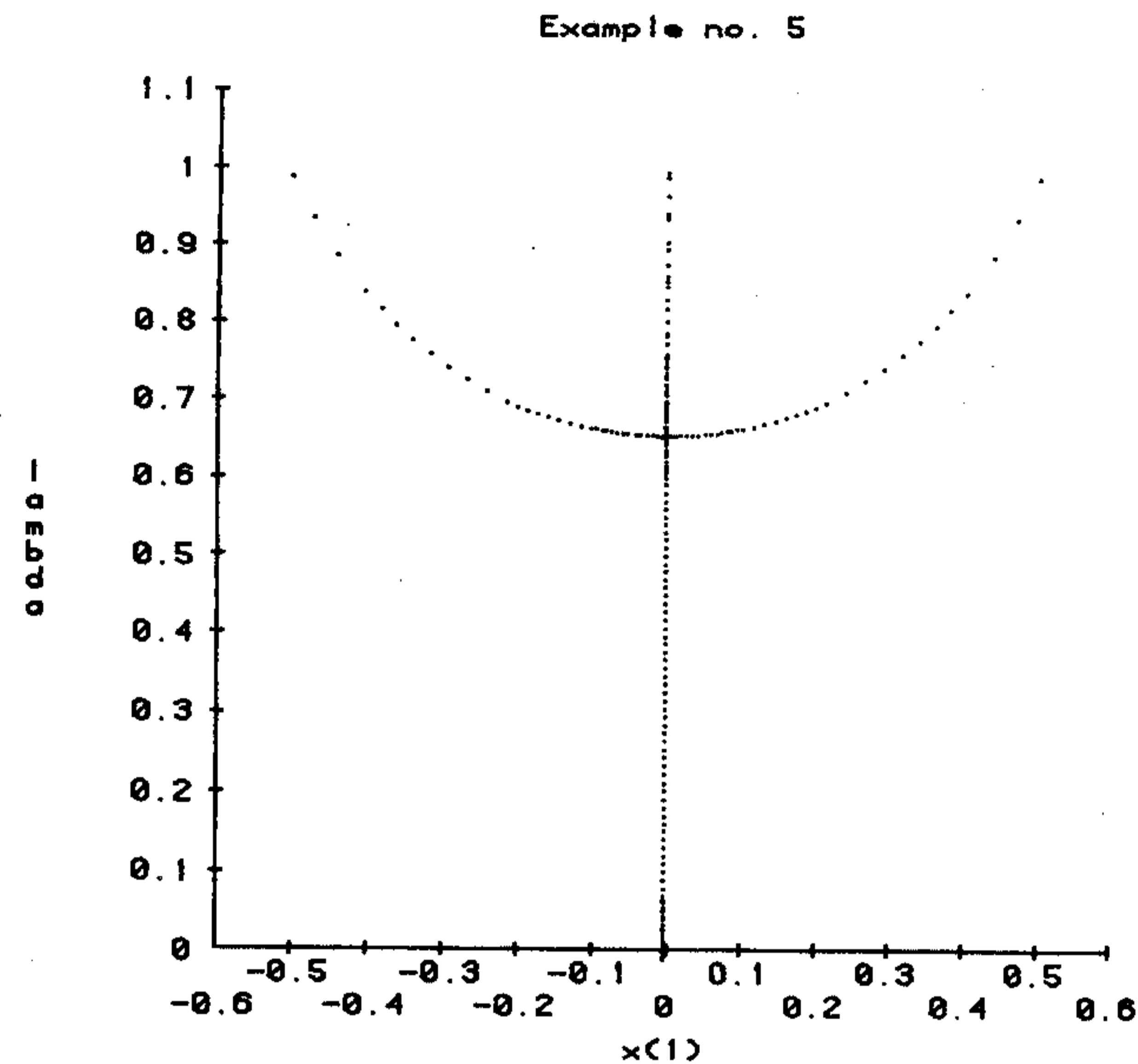


FIG. 11. Example 5.

The second part of each table deals with qualities (i) and (ii). The first and second columns (labeled BP and PBP, respectively) list the coordinates of the located bifurcation point and the coordinates of the previous bifurcation point (or origin), respectively. The third and fourth columns (FEP and PITL) give the total number of function evaluations required to obtain the bifurcation point from the previous one and the corresponding number of predictor steps, respectively. The final column (FEB) gives the number of function evaluations required by the minimization process to locate the $m^{(i)}$. Finally, the total number of function evaluations for the entire example, the total number of arcs intersecting at $\lambda = 1$ and the average number of function evaluations per arc are listed.

In all cases, the arc-following tolerance (ϵ_y in [11, Algorithm 2.1]) was set to 10^{-8} , and δ_{\max} was set equal to 1. In Examples 1, 2 and 3, $p = 6$ (cf. § 4) was adequate; $p = 3$ was adequate in Examples 4 and 5. The initial predictor step size (δ_0 in [11, Algorithm 2.1]) at $(\theta, 0)$ was in all cases set to the minimum δ_{\min} . In Examples 1, 2, 4 and 5, δ_{\min} was set to 10^{-5} ; in example 6, δ_{\min} was set to 10^{-6} . (The minimum stepsize δ_{\min} was judged small enough if it was less than the minimum distance between bifurcation points divided by $10(2n+3)$.) The tolerance ϵ_d in the simplex method of Nelder and Mead was set to $\max(\|\bar{y}\|, 1)\epsilon_y/(10\delta)$ where δ is the radius of the ball S_δ . All other tolerances were set as in [11].

The Fortran programs were run on a Honeywell Multics 68/80 system. The Fortran programs are in experimental form, and there were some unnecessary redundant function evaluations, included for convenience in I/O, etc. Thus, the values in Tables 1 through 5 should be considered somewhat large.

TABLE 1(a)

#	root	BP	FE
1	(0, 0, 1)	(0, 0, .1)	158
2	(3, 3, 1)	"	556*
3	(-3, -3, 1)	"	556*
4	(-5.2, 5.2, 1)	(0, 0, .0357)	385
5	(5.2, -5.2, 1)	"	377

* H' was singular at $\lambda = 1$ on these arcs.

TABLE 1(b)

#	BP	PBP	FEP	PITL	FEB
1	(0, 0, .0357)	(0, 0, 0)	340	219	675
2	(0, 0, .1)	(0, 0, .0357)	315	180	180

Total number of function evaluations, 4033. Total number of predictor steps, 1144. Number of function evaluations per arc, 806.

TABLE 2(a)

#	root	BP	FE
1	(0, 0, 0, 0, 1)	(0, 0, 0, 0, .0948)	223
2	(1.83, 3.41, 3.41, 1.83, 1)	"	581
3	(-1.83, -3.41, -3.41, -1.83, 1)	"	588
4	(-6.41, -2.3, 2.3, 6.41, 1)	(0, 0, 0, 0, .0281)	1706*
5	(6.41, 2.3, -2.3, -6.41, 1)	"	1706*
6	(-8.47, 7.34, 7.34, -8.47, 1)	(-4.83, 3.62, 3.62, -4.83, .0216)	381
7	(-8.65, 8.56, .67, -7.23, 1)	"	475
8	(-7.23, .67, 8.56, -8.65, 1)	"	503
9	(8.47, -7.34, -7.34, 8.47, 1)	(4.83, -3.62, -3.62, 4.83, .0216)	408
10	(7.23, -.67, -8.56, 8.65, 1)	"	490
11	(8.65, -8.56, -.67, 7.23, 1)	"	471
12	(-8.83, 9.87, -9.87, 8.83, 1)	(0, 0, 0, 0, .0109)	745
13	(8.83, -9.87, 9.87, -8.83, 1)	"	738

* Local minima of σ_n not corresponding to a singular H' were located on these arcs.

TABLE 2(b)

#	BP	PBP	FEP	PITL	FEB
1	(0, 0, 0, 0, .0109)	(0, 0, 0, 0, 0)	500	303	724
2	(0, 0, 0, 0, .0150)	(0, 0, 0, 0, .0109)	265	112	701
3	(0, 0, 0, 0, .0281)	(0, 0, 0, 0, .0150)	373	253	643
4	(0, 0, 0, 0, .0948)	(0, 0, 0, 0, .0281)	475	262	691
5	(-4.83, 3.61, 3.61, -4.83, .0216)	(0, 0, 0, 0, .0150)	1168	198	1186*
6	(4.83, -3.61, -3.61, 4.83, .0216)	"	1165	198	1192*

Total number of function evaluations, 18,188. Total number of predictor iterations, 3341. Number of function evaluations per arc, 1399.

* These are secondary bifurcation points.

TABLE 3(a)

#	root	BP	FE
1	(0, 0, 0, 0, 0, 0, 1)	(0, 0, 0, 0, 0, 0, .0931)	302
2	(-1.19, -2.34, -3.3, -3.7, -3.3, -2.34, -1.19, 1)	"	903
3	(1.19, 2.34, 3.3, 3.7, 3.3, 2.34, 1.19, 1)	"	830
4	(-4.26, -7.31, -4.26, 0, 4.26, 7.31, 4.26, 1)	(0, 0, 0, 0, 0, 0, .0260)	1037
5	(4.26, 7.31, 4.26, 0, -4.26, -7.31, -4.26, 1)	"	1043
6	(-10.1, -4.2, 2.82, 9.49, 2.82, -4.2, -10.1, 1)	(0, 0, 0, 0, 0, 0, .0125)	2989*
7	(10.1, 4.2, -2.82, -9.49, -2.82, 4.2, 10.1, 1)	"	2946*
8	(11.3, 0, -11.3, 0, -11.3, 0, 11.3, 1)	(0, 0, 0, 0, 0, 0, .00775)	2265*
9	(-11.3, 0, 11.3, 0, -11.3, 0, 11.3, 0)	"	1367*
10	(-13.8, 13.7, .0977, -11.8, .0977, 13.7, -13.8, 1)	(0, 0, 0, 0, 0, 0, .00562)	2969†
11	(13.8, 13.7, -.0977, 11.8, -.0977, -13.7, 13.8, 1)	"	2947†
12	(-14.1, 15.6, -14.1, 0, 14.1, 15.6, -14.1, 1)	(0, 0, 0, 0, 0, 0, .00456)	4577†
13	(14.1, -15.6, 14.1, 0, -14.1, 15.6, -14.1, 1)	"	4577†
14	(14.1, -15.8, 15.9, -16, 15.9, -15.8, 14.1, 1)	(0, 0, 0, 0, 0, 0, .00404)	2006†
15	(-14, 15.8, -15.9, 16, -15.9, 15.8, -14.1, 1)	"	1986†

* Local minima not corresponding to singular H' were found on these arcs.

† Due to a too-large tolerance, local minima not corresponding to a singular H' were identified as bifurcation points on these arcs; the minimization process correctly gave only one direction.

TABLE 3(b)

#	BP	PBP	FEP	PITL	FEB
1	(0, 0, 0, 0, 0, 0, .00404)	(0, 0, 0, 0, 0, 0, 0)	878	531	643
2	(0, 0, 0, 0, 0, 0, .00455)	(0, 0, 0, 0, 0, 0, .00404)	450	197	617
3	(0, 0, 0, 0, 0, 0, .00561)	(0, 0, 0, 0, 0, 0, .00455)	567	278	605
4	(0, 0, 0, 0, 0, 0, .00775)	(0, 0, 0, 0, 0, 0, .00561)	611	250	632
5	(0, 0, 0, 0, 0, 0, .0125)	(0, 0, 0, 0, 0, 0, .00775)	681	336	649
6	(0, 0, 0, 0, 0, 0, .0260)	(0, 0, 0, 0, 0, 0, .0125)	704	365	701
7	(0, 0, 0, 0, 0, 0, .0931)	(0, 0, 0, 0, 0, 0, .0260)	768	419	721

Total number of function evaluations 40,467. Total number of predictor steps 8,358. Number of function evaluations per arc: 809.

TABLE 4(a)

#	root	BP	FE
1	(0, 0, 0, 1)	(0, 0, 0, .5)	16
2	(7.05, -7.8, 7.05, 1)	"	161
3	(2.13, 3.66, 2.13, 1)	"	156
4	(-7.05, 7.8, 2.13, 1)	"	161
5	(5.66, 0, -5.66, 1)	"	160
6	(-5.66, 0, 5.66, 1)	"	157
7	(-2.13, -3.66, -2.13, 1)	"	132

TABLE 4(b)

#	BP	PBP	FEP	PITL	FEB
1	(0, 0, 0, .5)	(0, 0, 0, 0)	662	437	235, 732

Total number of function evaluations, 240,624. (Some function evaluations occurred while rejecting false directions at the bifurcation point.) Total number of predictor iterations, 1194. Number of function evaluations per branch, 34,374.

TABLE 5(a)

#	root	BP	FE
1	(0, 0, 1)	(0, 0, .667)	57
2	(0, .707, 1)	"	205
3	(.5, 0, 1)	"	193
4	(0, -.707, 1)	"	204
5	(-.5, 0, 1)	"	185

TABLE 5(b)

#	BP	PBP	FEP	PITL	FEB
1	(0, 0, .667)	(0, 0, 0)	468	315	54, 492

Total number of function evaluations, 55,804. Total number of predictor steps, 535. Number of function evaluations per arc, 11,160.

6. Summary, conclusions and possible improvements. We have explained various techniques for locating general bifurcation points and for following all arcs intersecting at such bifurcation points. These techniques were tried on five test examples.

The test results indicate the acceleration/deceleration scheme (§ 3) reliably finds bifurcation points, regardless of whether there is a change in the determinant of H_x , yet does not force excessive computation to be done where it is unnecessary. The efficiency can undoubtedly be further improved with a better choice of multiplication parameters for increasing and decreasing δ , etc.

The direction-finding algorithm (explained in § 4) seemed to work well on simple bifurcation points (where only two arcs intersect), but comparison with other methods would be desirable. Larger numbers of function evaluations were required when the dimension of the null space of $H'(\bar{y})$ was greater than 2, but arc directions were given very accurately, and restarting the arc following method caused no problems. Perhaps the technique of adjusting the radius of S_δ will allow computation of arc directions when the arcs intersect tangentially; this needs more investigation.

The excessive numbers of function evaluations in finding the $m^{(i)}$ when the dimension of the null space of H' is greater than 2 are due partially to the small tolerance ϵ_d used in the minimization routine; additional experiments are necessary to determine the effects of choosing ϵ_d larger. Perhaps a better way of finding all minima of a function of a small number of variables could be implemented.

There was an intrinsic difficulty in Example 4: numerous minima of $\|H\|$ not corresponding to arcs occurred on ∂S_δ . This phenomenon needs further study.

As presented above, each predictor-corrector step of the arc-following method will run in $O(n^3)$ algebraic operations; this is because solution of the system in formula

(4) and a singular value decomposition are both required. Appropriate use of matrix factorization and updating techniques will allow solution of the system in $o(n^2)$ operations [24]. Also, it may be possible to find minima of determinants related to this factorization in place of finding minima of σ_n , so that only $o(n^2)$ operations are required overall in a predictor-corrector step. The special structure of the problem can also be used to give more accurate H' without additional computations [24]. This will be reported on in the future.

Finally, we emphasize that the examples in § 5 were meant merely as tests of the techniques; not much can be said about the underlying differential equation when using a discretization with $n = 7$.

REFERENCES

- [1] E. ALLGOWER, *A survey of homotopy methods for smooth mappings*, 1980 (in [17]).
- [2] E. ALLGOWER AND K. GEORG, *Simplicial and continuation methods for approximating fixed points and solutions to systems of equations*, SIAM Rev., 22 (1980), pp. 28–85.
- [3] P. A. BUSINGER AND G. H. GOLUB, *Singular value decomposition of a complex matrix*, Algorithm 358, Comm. ACM, 12 (1960), pp. 564–565.
- [4] J. E. DENNIS, JR. AND R. B. SCHNABEL, *Least change secant updates for quasi-Newton methods*, SIAM Rev., 21 (1979), pp. 443–459.
- [5] A. EIGER, K. SIKORSKI AND F. STENGER, *A method of bisections for solving n nonlinear equations*, ACM Trans. Math. Software, submitted.
- [6] C. B. GARCIA AND T. Y. LI, *On a path-following method for systems of equations*, MRC Tech. Summ. Rep. 1983, Univ. Wisconsin, Madison, 1979.
- [7] C. B. GARCIA AND W. I. ZANGWILL, *Finding all solutions to polynomial systems and other systems of equations*, Math. Programming, 16 (1979), pp. 159–176.
- [8] K. GEORG, *On tracing an implicitly defined curve by quasi-Newton steps and calculating bifurcation by local perturbations*, this Journal, 2 (1981), pp. 35–50.
- [9] E. ISAACSON AND H. B. KELLER, *Analysis of Numerical Methods*, John Wiley, New York, 1966.
- [10] H. JÜRGENS, H. O. PEITGEN AND D. SAUPE, *Topological perturbations in the numerical nonlinear eigenvalue and bifurcation problems*, Rep. 7, Forschungsschwerpunkt Dynamische Systeme, Fachbereich Mathematik, Universität Bremen, 1979.
- [11] R. B. KEARFOTT, *A derivative-free arc continuation method and a bifurcation technique*, 1980 (in [17]).
- [12] ———, *An efficient degree-computation method for a generalized method of bisection*, Numer. Math., 32 (1979), pp. 109–127.
- [13] ———, *An improved program for generalized bisection*, to appear.
- [14] H. B. KELLER, *Numerical solution of bifurcation and nonlinear eigenvalue problems*, in Applications of Bifurcation Theory, P. H. Rabinowitz, ed., Mathematics Research Center Publication 8, Academic Press, New York, 1977.
- [15] T. Y. LI AND J. A. YORKE, *A simple reliable numerical algorithm for following homotopy paths*, in Analysis and Computation of Fixed Points, S. M. Robinson, ed., Academic Press, New York, 1979.
- [16] H. D. MITTLEMAN AND H. WEBER, *Numerical methods for bifurcation problems, a survey and classification*, in Bifurcation Problems and Their Numerical Solution, H. D. Middleman and H. Weber, eds., Birkhäuser, Basel, 1980.
- [17] E. L. ALLGOWER, K. GLASHOFF AND H.-O. PEITGEN, eds, *Numerical Solution of Nonlinear Equations*, Lecture Notes in Mathematics 878, Springer, New York, 1981.
- [18] J. A. NELDER AND R. MEAD, *A simplex method for function minimization*, Comput. J., 7 (1965), pp. 308–313.
- [19] M. J. D. POWELL, *A Fortran subroutine for solving systems of nonlinear algebraic equations*, in Numerical Methods for Nonlinear Algebraic Equations, P. Rabinowitz, ed., Gordon and Breach, New York, 1970.
- [20] P. H. RABINOWITZ, *Some aspects of nonlinear eigenvalue problems*, Rocky Mountain J. Math., 3 (1973), pp. 161–202.
- [21] F. STENGER, *Computing the topological degree of a mapping in R^n* , Numer. Math., 25 (1975), pp. 23–38.
- [22] G. W. STEWART, *Introduction to Matrix Computations*, Academic Press, New York, 1973.
- [23] N. M. TEMME, ed. *Nonlinear Analysis*, vol. 1, Mathematisch Centrum, Amsterdam, 1976.
- [24] MICHAEL TODD, private correspondence.
- [25] L. T. WATSON, *A globally convergent algorithm for computing fixed points of C^2 maps*, Appl. Math. Comput., 5 (1979), pp. 297–311.

AN EXACT PENALTY METHOD FOR CONSTRAINED, DISCRETE,
LINEAR l_∞ DATA FITTING*

BARRY JOE† AND RICHARD BARTELS†

Abstract. This paper presents an algorithm for solving linearly constrained, discrete, linear l_∞ approximation problems which makes use of a penalty linear programming approach. An implementation for small dense problems has been prepared and tested against two other codes, one published and the other not. Results of this testing are given. The paper is concluded with a short summary.

Key words. l -infinity, Chebyshev, data fitting, regression, exact penalty method, optimization, linear programming, constraints.

1. Introduction and formulation. We wish to solve the following problem:

Given a vector $c = [c_1, c_2, \dots, c_n]^T$, an $m \times n$ matrix A , an $m \times p$ matrix E , a $m \times q$ matrix G , a p -vector f and a q -vector h ,

$$(1.1) \quad \begin{aligned} & \text{minimize} && \|c - A^T y\|_\infty \\ & \text{subject to} && E^T y = f \\ & \text{and} && G^T y \geq h, \end{aligned}$$

where $y = [y_1, y_2, \dots, y_m]^T$ is a vector of unknown parameters. We will not be assuming anything about the nonnegative integers n, m, p or q .

It is assumed that all vectors and matrices are real. None of the matrices A, E or G are required to be of full rank—indeed, E and/or G may be vacuous—nor is it assumed that vectors y satisfying the constraints actually exist, so the problem posed is completely general.

Problem (1.1) can be formulated as the linear programming problem:

$$(1.2) \quad \begin{aligned} & \text{minimize} && e_1^T \begin{bmatrix} y_0 \\ y \end{bmatrix} \\ & \text{subject to} && \begin{bmatrix} e & A^T \\ e & -A^T \\ 0 & E^T \\ 0 & -E^T \\ 0 & G^T \end{bmatrix} \begin{bmatrix} y_0 \\ y \end{bmatrix} \geq \begin{bmatrix} c \\ -c \\ f \\ -f \\ h \end{bmatrix} \\ & \text{and} && y_0 \geq 0, \end{aligned}$$

where y_0 is a variable representing $\|c - A^T y\|_\infty$, e_1 is the vector $[1, 0]^T$ of dimension $m + 1$, i.e. the first unit vector, and e is the n -vector containing a 1 in all component

Problem (1.2) can be converted to its dual linear programming form:

$$(1.3) \quad \text{maximize} \quad [c^T \quad -c^T \quad f^T \quad -f^T \quad h^T \quad 0 \quad 0^T] \begin{bmatrix} t \\ u \\ v \\ w \\ z \\ s_0 \\ s \end{bmatrix}$$

* Received by the editors November 9, 1980, and in revised form April 12, 1982.

† Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada N2L 3C1

