FACILITY FORM 602

N68-37733 (ACCESSION NUMBER) | (THRU)
45 (PAGES) | 1 (CODE)
CR-97311 (NASA CR OR TMX OR AD NUMBER) | 19 (CATEGORY)

GRAPHICAL APPROXIMATION TO THE DOMAIN OF ATTRACTION FOR SECOND ORDER SYSTEMS

Ryuzo Yokoyama

Edwin Kinnen

GPO PRICE $ ______

CFSTI PRICE(S) $ ______

Hard copy (HC) _

Microfiche (MF) _

ff 653 July 65

August 1968

This research was supported by the National Aeronautics and Space Administration under Grant NsG-574.

THE UNIVERSITY OF ROCHESTER
COLLEGE OF ENGINEERING AND APPLIED SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
ROCHESTER, NEW YORK

GRAPHICAL APPROXIMATION TO THE DOMAIN OF ATTRACTION FOR SECOND ORDER SYSTEMS

Ryuzo Yokoyama

Edwin Kinnen

College of Engineering and Applied Science

University of Rochester

Rochester, New York 14627

August 1968

This research was supported by the National Aeronautics and Space Administration under Grant NsG-574.

GRAPHICAL APPROXIMATION TO THE

DOMAIN OF ATTRACTION

FOR SECOND ORDER SYSTEMS

CONTENTS

Graphical Approximation to the

Domain of Attraction

For Second Order Systems

Ryuzo Yokoyama

Edwin Kinnen

University of Rochester

Rochester, New York 14627

Abstract

A method is described for developing an approximate domain of attraction for a singular point of the ordinary autonomous second order differential equation $\ddot{x} + f(x,\dot{x})\dot{x} + g(x,\dot{x}) = 0$, where f and g are finite order polynomials in x and $\dot{x}$. Reverse solution trajectories are calculated for domain boundaries from a selective set of end point conditions known to belong to the domain of attraction. The latter is found initially from an arbitrarily chosen Liapunov function. The process is arranged for machine computation and is particularly effective for equations with solution trajectories not exhibiting limit cycles.

GRAPHICAL APPROXIMATION TO THE DOMAIN OF ATTRACTION FOR SECOND ORDER SYSTEMS

I. INTRODUCTION

The problem of establishing stability characteristics of a singular point is fundamental to the analysis of nonlinear systems. A more difficult problem arises, however, if the domain of attraction about a singular point found to be asymptotically stable is to be determined. The significant practical interest in resolving this latter problem justifies an investigation into the use of computer assisted solutions. The domain of attraction, or the region of asymptotic stability, defines those points in state space around a stable singular point corresponding to initial conditions of all solution trajectories which approach the singular point as t becomes increasingly more positive. A computer assisted procedure is described for systematically developing an arbitrary close approximation to the domain of attraction for a class of second order nonlinear autonomous ordinary differential equations, based on the concepts of an invarient set and limiting set from stability theory [1].

Methods which have been suggested for estimating the domain of attraction are directed generally to finding alternate Liapunov functions which describe more inclusive portions of the domain [2,3,4]. This procedure is based instead on the calculation of selective reverse solution trajectories. It is convenient to use a Liapunov function for the system in an arbitrarily small region about the singular point to start the procedure. The results are independent of the choice of this Liapunov function, however.

While the development has been restricted to one class of nonlinear second order equations, it appears possible to extend the ideas to more complex equations.

The problem is described in Section II, along with some characteristics of solution trajectories of the class of nonlinear differential equations. A theorem is stated in Section III to support the construction procedure. Details of the method and some practical considerations are given in Section IV. The next section describes a machine program for constructing an approximation to the domain of attraction. Three examples with computer assisted solutions appear in Section VI.

II. STATEMENT OF THE PROBLEM

Consider a second order system

$$\begin{cases} \dot{x} = y \\ \dot{y} = N(x,y), \end{cases} \tag{1}$$

where $N(x,y)$ is assumed to be a finite order polynomial in x and y of order n, say

$$N(x,y) = a_{00} + (a_{10}x + a_{11}y) + (a_{20}x^2 + a_{21}xy + a_{22}y^2) + \cdots$$

$$\cdots + (a_{n0}x^n + a_{n1}x^{n-1}y + \cdots + a_{n,n-1}xy^{n-1} + a_{nn}y^n)$$

$$= \sum_{m=0}^{n} \sum_{i=0}^{m} a_{mi}x^{m-i}y^i. \tag{2}$$

Writing (2) as

$$N(x,\dot{x}) = -f(x,\dot{x})\dot{x} - g(x,\dot{x}), \tag{3}$$

equation (1) is equivalently given as

$$\ddot{x} + f(x,\dot{x})\dot{x} + g(x,\dot{x}) = 0. \tag{4}$$

As singular points of (1) are defined for $\dot{x} = \dot{y} = 0$, all y coordinates are zero and the x coordinates are real roots of the algebraic equation

$$N(x,0) = 0$$

or $$a_{00} + a_{10}x + \cdots + a_{n0}x^n = 0. \tag{5}$$

Consider $k(k \leq n)$ distinct real roots of (5) such that the singular points of (2) are

$$(\alpha_1,0),\ (\alpha_2,0),\ \cdots,\ (\alpha_k,0). \tag{6}$$

By a parallel shift of the y axis, the position of each point can be moved to the origin of a new coordinate system. It is convenient to assume, therefore, that an asymptotically stable singular point is located at the origin and, without loss of generality, to consider the problem of finding the domain of attraction of the origin [5]. This assumption requires that the coefficient a_{00} in (5) is zero. The origin is also assumed to be an isolated singular point, i.e., at least one of the other coefficients in (5) is not zero.

Restated, the problem is the following: for a given system

$$\begin{cases} \dot{x} = y \\ \dot{y} = N(x,y), \end{cases} \tag{7}$$

where (i) $N(x,y)$ is an n^{th}-order polynomial in x and y,

$$N(x,y) = \sum_{m=1}^{n} \sum_{i=0}^{m} a_{mi} x^{m-i} y^i, \quad (n \text{ finite}) \tag{8}$$

and (ii) at least one a_{10}, a_{20}, $\cdots$, a_{n0} is nonzero; obtain an approximation to the domain of attraction of the origin, say D, by another domain entirely within D and arbitrarily close to D.

It is pertinent to recall some characteristics of the solution to (7).

Remark 1: The satisfaction of the Lipschitz condition in a domain R of the state plane for (7) guarantees the existence of a unique solution for any initial state and the nonintersection of trajectories except at singular points. Furthermore, all trajectories are directed to the right in the upper half state plane and to the left in the lower half state plane. On the x axis, trajectories are directed perpendicularly up if $N(x,0) > 0$ and down if $N(x,0) < 0$. If $N(x,0) = 0$, the point is a singular point by definition.

The following theorem provides a necessary condition for the nonlinearity $N(x,0)$ in (7) if the origin is asymptotically stable.

Theorem 1: If the origin of (7) is asymptotically stable, there exists an $\varepsilon > 0$ such that

$$x \cdot N(x,0) < 0 \tag{9}$$

for all $0 < |x| \leq \varepsilon$.

Proof: Consider the contrary, that there is a $\delta_1 > 0$ or a $\delta_2 > 0$ such that

$$N(x,0) > 0 \qquad \text{for all } 0 < x \leq \delta_1 \tag{10}$$

or $$N(x,0) < 0 \qquad \text{for all} -\delta_2 \leq x < 0. \tag{10'}$$

For (10), choose an initial state

$$(x_0,0), \tag{11}$$

where $$0 < x_0 < \delta_1. \tag{12}$$

The trajectory extends from (11) into the first quadrant perpendicular at the x axis and is then directed to the right becoming more distant from the origin. For the trajectory to approach to the origin as $t \to +\infty$, it must necessarily enter the fourth quadrant.

Since, by (10), this cannot occur through an intersection of the segment between the origin and $(\delta_1, 0)$, the trajectory must become more distant from the origin than $(\delta_1, 0)$, independent of the magnitude of x_0. Because of this independence of x_0, the assumption of asymptotic stability of the origin is contradicted. The proof is similar if (10') is assumed.

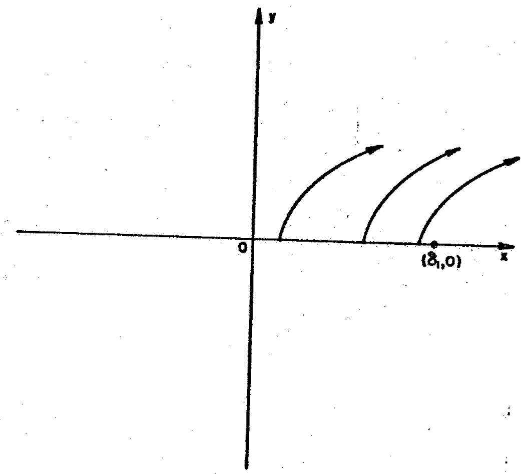

Figure 1

Theorem 2: Let R be a closed, simply connected, bounded region in the state plane of (7). If R does not include any singular point, then a trajectory of (7) starting from a point in R must reach the boundary of R in finite time.

Outline of the proof: A trajectory starting from a point in R can stay in R for all subsequent time if and only if:

(i) R is unbounded, or

(ii) the closure of R, i.e., $\bar{R}$, includes a singular point, or

(iii) $\bar{R}$ completely includes a limit cycle.

The possibility of (i) or (ii) is denied by the boundedness and the closedness of R respectively. The possibility of (iii) is denied by the simple connectedness and closedness, that is, if there exists a limit cycle in R = R, there must exist a singular point inside the limit cycle [6] and in R. Consequently under the assumptions no trajectory can stay in R for all time.

Identifying function: Consider a set of functions of class C_1 such that

$$h(x,y) = K, \tag{13}$$

where K is a parameter. Equation (13) is a contour field in the state plane of (7). Noting the value of the time derivative of (13) under (7) at some point (x_a,y_a) is

$$\dot{h}(x,y)\Big|_{(x_a,y_a)} = \{\frac{\partial h}{\partial x}y_a + \frac{\partial h}{\partial y}N(x_a,y_a)\}, \tag{14}$$

the following statements are evident.

(i) If the value of (14) is positive, the trajectory at (x_a,y_a) is directed so as to climb the contour, toward increasing values of K of (13).

(ii) If the value of (14) is negative, the trajectory at (x_a,y_a) is directed so as to descend the contour, toward decreasing values of K of (13).

(iii) If the value of (14) is zero, the trajectory at (x_a,y_a) is tangent to the contour.

Equation (13) is called an identifying function.

Remark 2: Suppose that (x_0,y_0) is a regular point in a domain of attraction of the origin. Consider a trajectory T starting from a point (x_1,y_1) such that it reaches (x_0,y_0) and the time interval for this transition of the solution is finite. Then any point on T, including (x_1,y_1), is a point in the domain of attraction due to the uniqueness of the solution and the definition of asymptotic stability. For this purpose, it is necessary and sufficient that T includes no singular point between (x_0,y_0) and (x_1,y_1), and and that the distance between these two points is finite along T.

III. FUNDAMENTAL THEOREM

Assume that Ω is a know subset of D.

Theorem 3: Let (a) L be a line segment in Ω, (b) P be a trajectory not necessary within Ω such that it reaches a point (x_0,y_0)

on L, and (c) $h(x,y) = k$ be an identifying function which intersects both L and P at ℓ and ρ, as in figure 2. Assume all distances between intersecting points along each of the segments are finite. Denote the domain surrounded by these three segments plus the boundary as D_s, a closed domain. Then (i) if $\dot{h}(x,y)$ under (7) is sign definite for all points on the segment between ρ and ℓ and including ρ, and (ii) D_s does not include any singular point, then

$$D_s \subset D \tag{15}$$

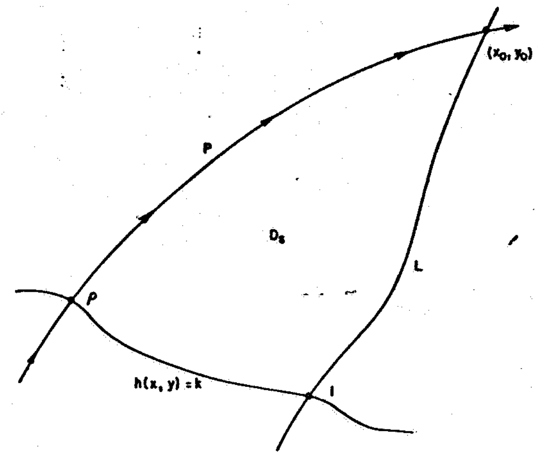

Figure 2

Proof: By assumption, D_s satisfies the conditions of Theorem 2. Hence any trajectory in D_s must reach some boundary point of D_s in a finite time. But no trajectory can leave D_s at a point on P between ρ and (x_0,y_0) except possibly at (x_0,y_0). Also no trajectory can leave D_s at a point on the identifying function between ℓ and ρ due to the sign definiteness of the derivative of the identifying function. Thus any trajectory which includes points in D_s must also include a point on the segment L between ℓ and (x_0,y_0). As L is included in D, and from Remark 2, $D_s \subset D$.

Consider a modification of Theorem 3 for a similarly defined L and $h(x,y) = k$. Let D_s' be defined as a domain surrounded by L and $h(x,y) = k$, and two trajectories P_1 and P_2 starting from different points ρ_1 and ρ_2 on $h(x,y) = k$ and reaching points ℓ_1 and ρ_2 on L, as shown in figure 3. All distances between intersecting points ρ_1, ρ_2, ℓ_1 and ℓ_2 along each of the segments are assumed finite.

Lemma: (i) If $\dot{h}(x,y)$ is sign definite at all points on $h(x,y) = k$ between ρ_1 and ρ_2 including ρ_1 and ρ_2, and (ii) if D_s' does not include any singular point, then

$$D_s' \subset D.$$

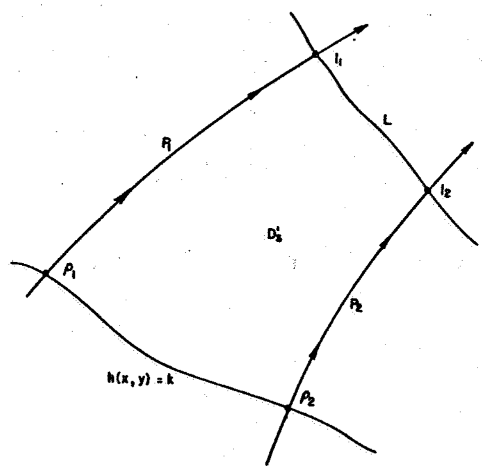

Figure 3

IV. METHOD OF APPROXIMATING THE DOMAIN OF ATTRACTION

Theorem 3 and the Lemma can be used routinely to develop D for a given system (7). The process is described as finding or constructing:

1. a line segment L which is a subset of D,
2. a trajectory P (or P_1 and P_2) to reach a point on L,
3. an identifying function $h(x,y) = k$ to construct a domain D_s (or D_s'); then
4. the condition of Theorem 3 or the Lemma is checked for

$$D_s \subset D \qquad (\text{or } D_s' \subset D),$$

and the process is repeated for a new D_s.

It is necessary for the nonlinearity of (7) to statisfy the condition of Theorem 1; if this fails, the origin cannot be asymptotically stable.

Remark 3: To identify a line segment L in D, it is sufficient to find a Liapunov function V(x,y) to prove the asymptotic stability of the origin. This defines a region Ω

$$\Omega; \quad V(x,y) \leq \alpha \qquad (\alpha > 0: \text{const}), \tag{16}$$

where $\dot{V}$ is negative definite. Define

$$L_v; \quad V(x,y) = \alpha.$$

Then

$$L_v \subset D \tag{17}$$

and can be used initially as a segment L. Using L_v for L also simplifies later calculations, as all trajectories reach L_v from the external side of Ω.

To find V(x,y), any of the many suggested construction methods for an autonomous system can be employed. For machine computation, Rodden's method [3], based on a Zubov theorem, may be used if necessary. As Ω is only required to initiate the procedure, the magnitude of Ω is of little consequence.

Remark 4: Relative to step 2, finding a trajectory to reach a point ℓ on L_v, a reverse time trajectory from some point on L_v is appropriate. The reverse time system of (7) is defined by

$$\begin{cases} \dot{x} = -y \\ \dot{y} = -N(x,y). \end{cases} \tag{18}$$

Remark 5: Simple identifying functions are worth considering. Observe the cases when

$$h_1(x,y) = x = k, \text{ a constant}, \tag{19}$$

and $$h_2(x,y) = y = 0. \tag{20}$$

From (7) and Remark 1, (19) becomes

$$\dot{h}_1(x,y)\Big|_{x=k} = \dot{x} = y > 0 \quad \text{in the upper half plane,} \tag{21}$$

and $$< 0 \quad \text{in the lower half plane.} \tag{21'}$$

Alternately (20) is

$$\dot{h}_2(x,y)\Big|_{y=0} = \dot{y}\Big|_{y=0} = N(x,0). \tag{22}$$

which is sign definite on each segment of the x axis between singular points. If the origin is asymptotically stable, (22) is negative on the x axis between the origin and the nearest singular point to the right and positive on the x axis between the origin and the nearest singular point to the left (Theorem 1). Therefore the existence of appropriate identifying functions such as (19) or (20) is sufficient to satisfy part of the conditions of Theorem 3 on the domain $D_s \subset D$.

To systematically construct domains D_s using identifying functions (19) and (20), suppose a Ω and L_v have been found as shown in figure 4. Define in this figure

C_0: the intersection between L_v and the x axis in the right half plane.

q_0: the intersection between L_v and the x axis in the left half plane.

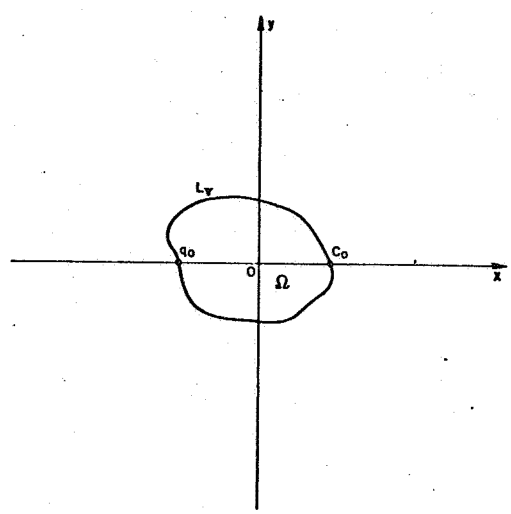

Figure 4

As Ω is a subset of D, it cannot include any singular point, except the origin. By Theorem 1,

$$N(x,0) < 0 \quad \text{for all } x \text{ between the origin and } C_0, \tag{23}$$

$$N(x,0) > 0 \quad \text{for all } x \text{ between the origin and } q_0. \tag{23'}$$

(1) Consider a reversed time trajectory T_1 from C_0. T_1 necessarily becomes more negative than q_0 in the upper half plane, by (23) and Remark 1. As shown in figure 5, define D_s^1 as the domain surrounded by L_v, T_1, $x = k < q_0$, and the x axis between q_0 and $(k,0)$, where k is otherwise an arbitrary value. If there exists no singular point between q_0 and $(k,0)$ on the x axis, then Theorem 3 insures

$$D_s^1 \subset D.$$

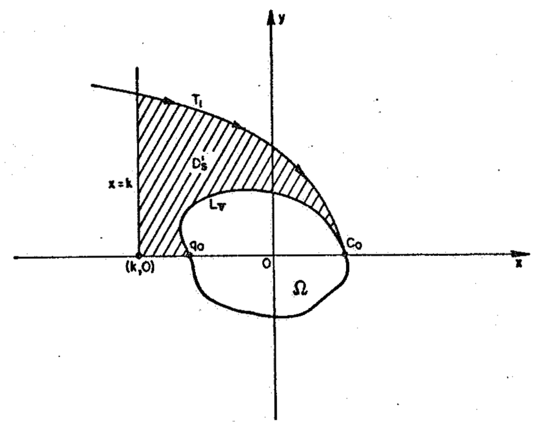

Figure 5

(2) As long as the reverse time trajectory T_1 remains in the upper half plane, the domain D_s^1 can be extended until a singular point appears between q_0 and $(k,0)$. Assume such a singular point at $(\alpha_1,0)$, as in figure 6. Then the domain surrounded by L_v, T_1, $x = \alpha_1 + \varepsilon_1$ and the x axis between q_0 and $(\alpha_1 + \varepsilon_1,0)$ can be identified as D_s^1, where ε_1 is a small positive value. In the vicinity of the singular point $(\alpha,0)$, there occur two possible initial states of reverse time trajectories of a new subdomain of D. These are $(\alpha_1 + \varepsilon_1,\delta_1)$ and $(\alpha_1 + \varepsilon_1,0)$, where δ_1 is defined in the following.

(3) Consider a second reverse time trajectory T_2, starting from $(\alpha_1 + \varepsilon_1,\delta_1)$, as shown in figure 6. Call the first inter-

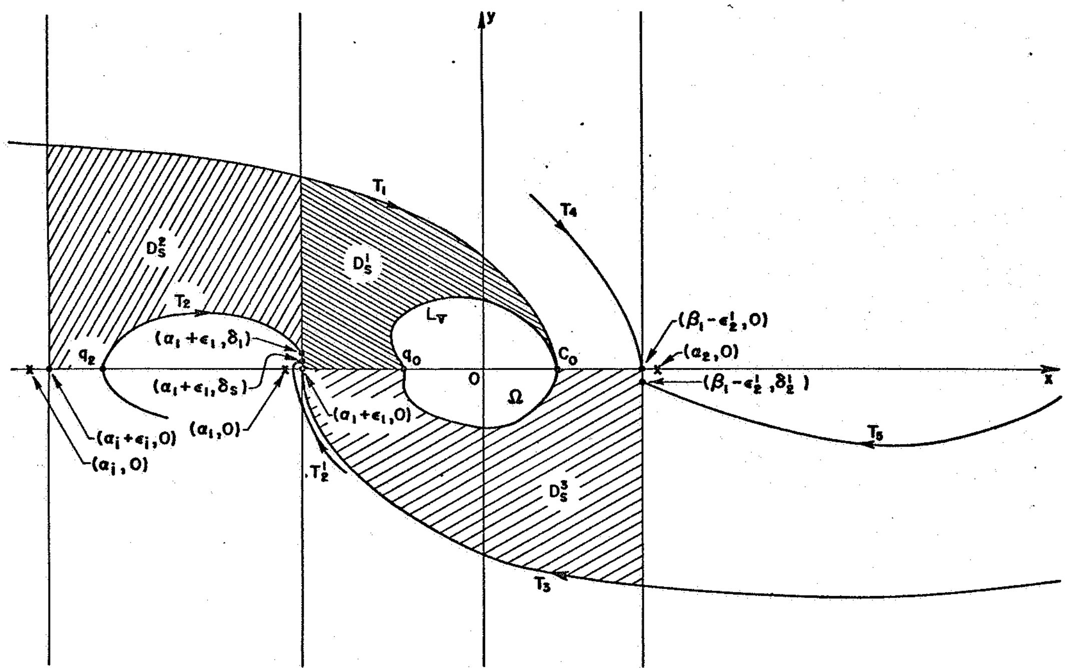

Figure 6

section of T_2 and the x axis q_2 (if it exists) and assume the nearest singular point to the left of q_2 to be at $(\alpha_i,0)$. If T_1 remains in the upper half plane above the singular point $(\alpha_i,0)$, consider the domain D_s^2 as that surrounded by T_1, T_2, $x = \alpha_1 + \varepsilon_1$, the segment of the x axis between q_2 and $(\alpha_i + \varepsilon_i,0)$ and $x = \alpha_i + \varepsilon_i$, where ε_i is an arbitrary small positive value. Since by assumption there exists no singular point on the x axis between q_2 and $(\alpha_i + \varepsilon_i,0)$, $\dot{y} = N(x,0)$ on the x axis between $(\alpha_i + \varepsilon_i,0)$ and q_2 is sign definite and the Lemma insures

$$D_s^2 \subset D.$$

If a very small value is chosen for δ_1, e.g., δ_s as in figure 6, the reverse time trajectory T_2' from $(\alpha_1 + \varepsilon_1,\delta_s)$ may enter the lower half plane without passing over the singular point $(\alpha_1,0)$. Then the construction of a new subdomain of D using T_1 will fail in the upper half plane. Therefore, δ_1 must be chosen so that the reverse time trajectory T_2 passes over the singular point and be a lower-bound positive value satisfying this restriction.

If, additionally, T_1 intersects the x axis at q_3, as in figure 7, and no singular point exists between q_2 and q_3, D_s^2 is reduced to the domain surrounded by T_1, T_2, $x = \alpha_1 + \varepsilon_1$, and the segment of the x axis between q_2 and q_3. If T_2 stays in the upper half plane forever, D_s^2 can be identified as an infinite strip partially surrounded by T_1, T_2 and $x = \alpha_1 + \varepsilon_1$, as shown in figure 8.

(4) For a reverse time trajectory which starts from $(\alpha_1 + \varepsilon_1, 0)$ and is directed to the right in the lower half plane, as shown in figure 6 as T_3, arguments similar to those considered in (1)-(3) can be applied. Let D_s^3 be the domain surrounded by T_3, the segments of the x axis between $(\alpha_1 + \varepsilon_1,0)$ and q_0, and between C_0 and $(\beta_1 - \varepsilon_1',0)$, L_v and $x = \beta_1 - \varepsilon_1'$. $(\beta_1,0)$ is assumed to be the nearest singular point to the right of the origin and ε_1' is a small arbitrary positive value. Then the Lemma insures

$$D_s^3 \subset D.$$

There occur two possible initial states for reverse time trajectories T_4 and T_5 in the vicinity of the singular point, i.e., $(\beta_1 - \varepsilon_1', -\delta_1')$ and $(\beta_1 - \varepsilon_1',0)$. δ_1' is a small positive so that T_5 passes below $(\beta_1,0)$ for the same reason given for T_2 in item (3).

(5) If T_1 in figure 5 enters the third quadrant at point $(q_1,0)$, as shown in figure 9, with no singular point between $(q_1,0)$ and the origin, then D_s^1 is the domain surrounded by L_v, T_1 and the segment of the x axis between q_0 and q_1, and $D_s^1 \subset D$.

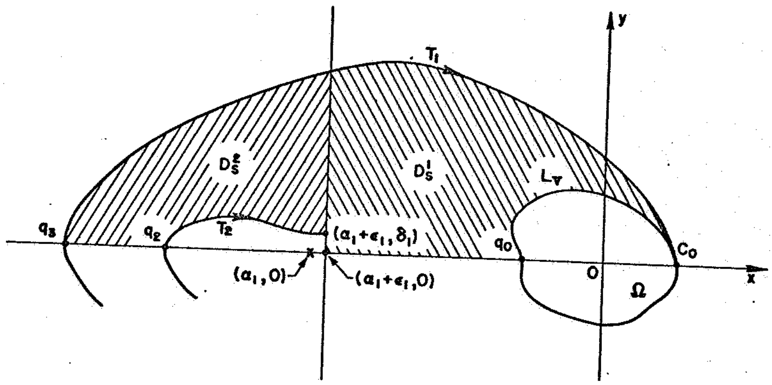

Figure 7

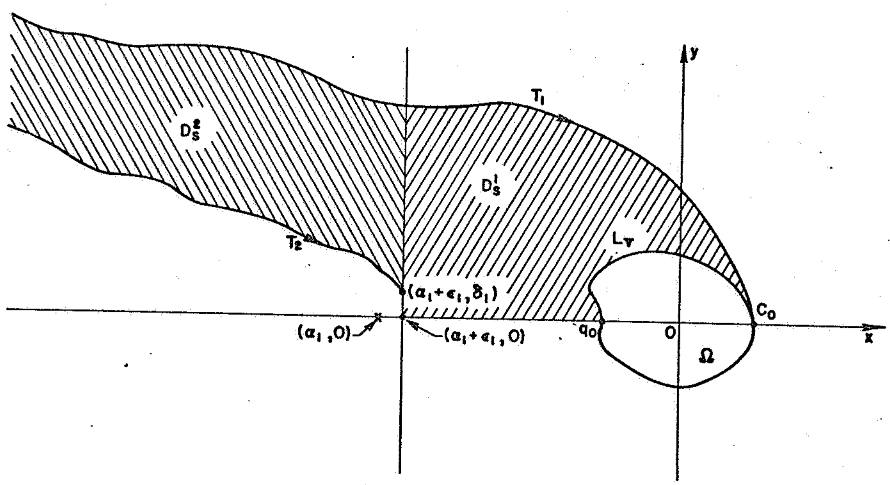

Figure 8

Continuing these steps (1)-(5), based on any known subdomain of D, new subdomains of D can be sequentially developed. Note that constructed subdomains using reverse time trajectories can include no singular points. It is this characteristic combined with the identifying functions (19) and (20) that essentially simplify the foregoing procedure for the machine computation described in the next section.

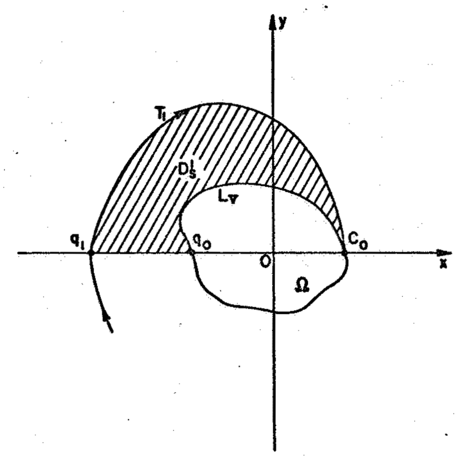

Figure 9

V. MACHINE COMPUTATION

The steps given in the last section can be programmed for machine computation and the direct plotting of an approximation to the domain of attraction. Initially it is necessary to check the condition of Theorem 1 and determine a known segment in D, e.g., a Liapunov function as stated. For a machine program it is sufficient to calculate and plot reverse time trajectories to approximate D from selected initial states. The approximated domain is then identified visually, referring to the considerations of the previous section. A program has been assembled to sequentially: (C_1) calculate reverse time trajectories from selected initial states, (C_2) develop initial conditions for additional trajectories and (C_3) end when all relevant trajectories are calculated. It is necessary to locate singular points of system (7). These exist on the x axis with x coordinates which are the real roots of

$$N(x,0) = 0. \tag{24}$$

Any standard method can be applied to solve this equation. In the program which follows, all singular points are assumed to be identified for the computation and inserted as data input. The flow chart of the program appears in figure 10.

For (C_1), the Runge-Kutta method was used to solve the time reversed trajectories approximated by connecting segments of infinitesimal time intervals. End points of a segment, say (x_n, y_n) for $t = t_n$ and (x_{n+1}, y_{n+1}) for $t = t_n + \Delta t$, are related [7] as

$$\begin{cases} x_{n+1} = x_n + \frac{1}{6}(K_o + 2K_1 + 2K_2 + K_3) \\ y_{n+1} = y_n + \frac{1}{6}(L_o + 2L_1 + 2L_2 + L_3), \end{cases} \tag{25}$$

where

$$\begin{cases} K_o = -\Delta t \cdot y_n \\ L_o = -\Delta t \cdot N(x_n, y_n) \\ K_1 = -\Delta t \cdot (y_n + \frac{L_o}{2}) \\ L_1 = -\Delta t \cdot N(x_n + \frac{K_o}{2}, y_n + \frac{L_o}{2}) \\ K_2 = -\Delta t \cdot (y_n + \frac{L_1}{2}) \\ L_2 = -\Delta t \cdot N(x_n + \frac{K_1}{2}, y_n + \frac{L_1}{2}) \\ K_3 = -\Delta t \cdot (y_n + L_2) \\ L_3 = -\Delta t \cdot N(x_n + K_2, y_n + L_2). \end{cases} \tag{26}$$

Each point (x_n, y_n) along the trajectories are punched out for subsequent machine plotting.

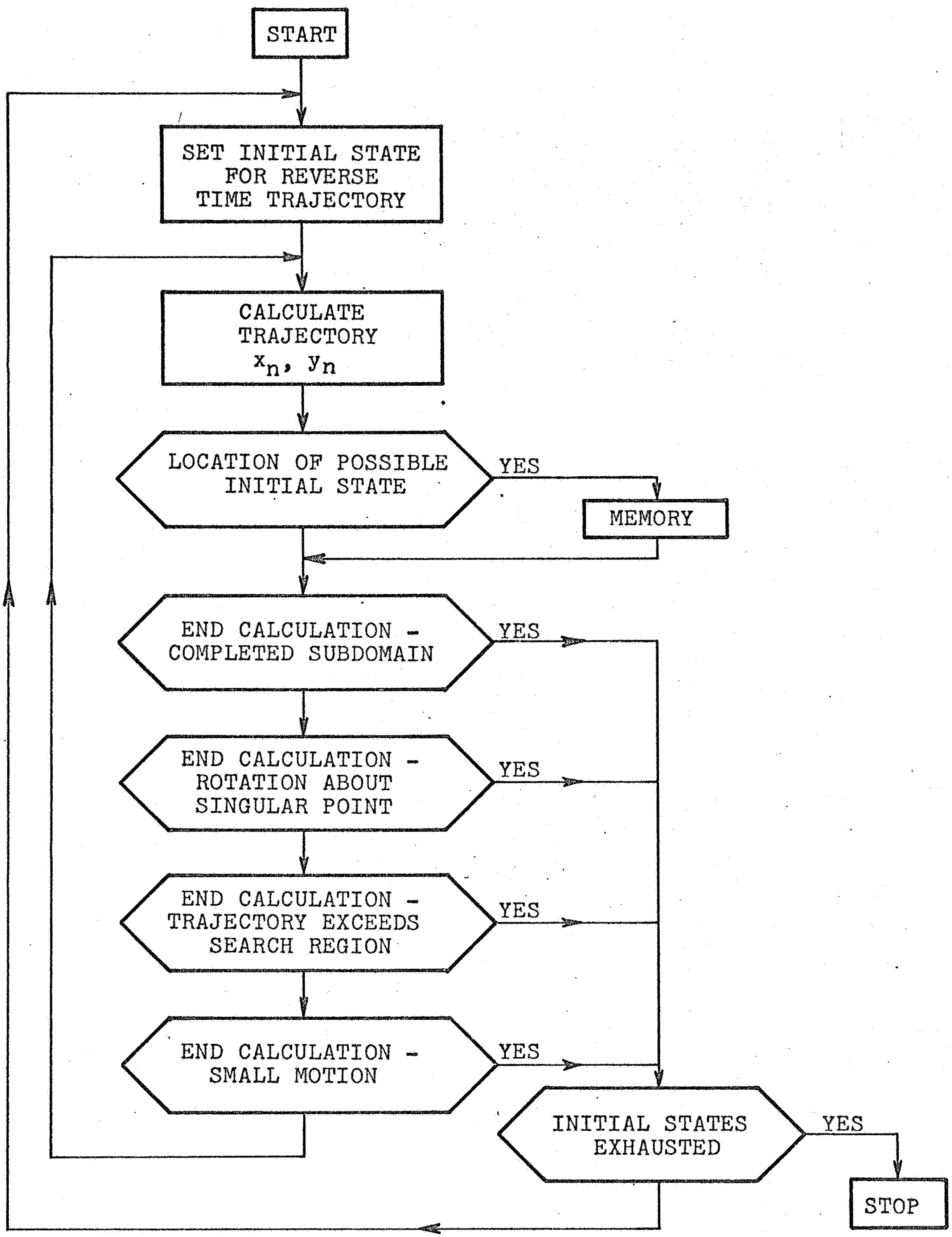
START
SET INITIAL STATE FOR REVERSE TIME TRAJECTORY
CALCULATE TRAJECTORY x_n, y_n
LOCATION OF POSSIBLE INITIAL STATE
YES
MEMORY
END CALCULATION - COMPLETED SUBDOMAIN
YES
END CALCULATION - ROTATION ABOUT SINGULAR POINT
YES
END CALCULATION - TRAJECTORY EXCEEDS SEARCH REGION
YES
END CALCULATION - SMALL MOTION
YES
INITIAL STATES EXHAUSTED
YES
STOP

Figure 10

For (C_2), the initial state for the first reverse time trajectory is predetermined as a point on L_v. Initial states for other trajectories depend upon subsequent results but exist only in the vicinity of singular points. Generally, for a singular point to the immediate left of a subdomain boundary in the upper half plane, as $(\alpha,0)$ in figure 6, two possible initial states exist, $(\alpha + \varepsilon,0)$ and $(\alpha + \varepsilon,\delta)$. ε is an arbitrary small positive number and δ is the lower-bound positive value so that the reverse time trajectory from $(\alpha + \varepsilon,\delta)$ passes above $(\alpha,0)$. Alternately, about a singular point $(\alpha,0)$ to the right of a subdomain boundary in the lower half plane, as $(\alpha,0)$ in figure 6, $(\alpha - \varepsilon,0)$ and $(\alpha - \varepsilon,-\delta)$ are possible initial states of reverse time trajectories which pass below $(\alpha,0)$. If a new possible initial state is found during the calculation from a trajectory passing over or under a singular point, this initial state is stored for the later calculations. For convenience, at the beginning of a reverse time trajectory calculation, an identifying trajectory number and the corresponding initial state are printed out.

The calculation of each trajectory is stopped when

(i) it is extended so as to establish a subdomain of D,

(ii) it is extended to a preselected limit value of the coordinates,

(iii) it rotates many times about a singular point indicating an approach to a limit cycle, or

(iv) its extension becomes infinitesimal as it approaches a singular point.

For (ii), the calculation is stopped when the trajectory reaches the limits of the region

$$|x| < P, \tag{27}$$

where P is chosen arbitrarily but so that $x = \pm P$ are not singular points. When the computation of a trajectory is stopped for exceeding this limit, the statement "CHANGE OF TRAJECTORY DUE TO EXCESS VALUE OF X" is printed out. For (iii), the number of changes of sign for y is counted along each reverse time trajectory and is printed out as "NO. OF ROTATIONS * $\frac{1}{2}$". When this number exceeds a preset value, the calculation is stopped and the statement "CHANGE OF TRAJECTORY DUE TO EXCESS ROTATION" is printed out. For (iv), the amount

$$\sum_{i}^{i+100} \max \{|x_i - x_{i-1}|, \ |y_i - y_{i-1}|\}$$

is accumulated along the trajectory and if the average value of the maximum changes becomes

$$\frac{\sum_{i}^{i+100} \max \{|x_i - x_{i-1}|, \ |y_i - y_{i-1}|\}}{100} \leq SK, \qquad (28)$$

where SK is a predetermined value, the calculation is stopped and the statement "CHANGE OF TRAJECTORY DUE TO STEADY STATE" is printed out.

When the possible initial states for reverse time trajectories are exhausted in the region indicated by (27) and the last trajectory is terminated, the entire computation is complete.

The Fortran program assembled for this computation and used for the examples of the next section is listed in the Appendix. The output of the calculation is both printed and punched out, the latter then used for a standard plotting routine for a graphical result.

VI. EXAMPLES

Three examples demonstrate the method and illustrate numerically calculated domains of attraction. The identifying functions used were

$$x = K \qquad (19)$$

and

$$y = 0. \qquad (20)$$

The region for the search was arbitrarily restricted to the x coordinate. If a reverse time trajectory rotated more than four times around the origin, the calculation was arranged to end, anticipating a limit cycle. Δt was assumed to be 0.01.

Example 1: The system is given as

$$\begin{cases} \dot{x} = y \\ \dot{y} = -x - 6y + 2y^3. \end{cases} \tag{29}$$

(i) As

$$x \cdot N(x,0) = -x^2$$

the condition of Theorem 1 is satisfied.

(ii) A singular point exists at the origin.

(iii) Assume

$$V(x,y) = x^2 + y^2$$

and the time derivative under (29) is

$$\dot{V}(x,y) = 2x\dot{x} + 2y\dot{y} = -4y^2(3 - y^2).$$

Therefore

$$\begin{aligned} \dot{V}(x,y) &< 0 \quad \text{if} \quad y^2 < 3 \quad \text{and} \quad y \neq 0, \\ &= 0 \quad \text{if} \quad y = 0. \end{aligned} \tag{30}$$

On the x axis,

$$\begin{cases} \dot{x} = 0 \\ \dot{y} = -x \end{cases}$$

and any trajectory terminates at the origin. From (30), it is possible to choose L_v as

$$x^2 + y^2 = (1.5)^2$$

which is completely included in D.

The computed result is shown in figure 11. Starting from (1.5,0) the calculation stopped after the reverse time trajectory

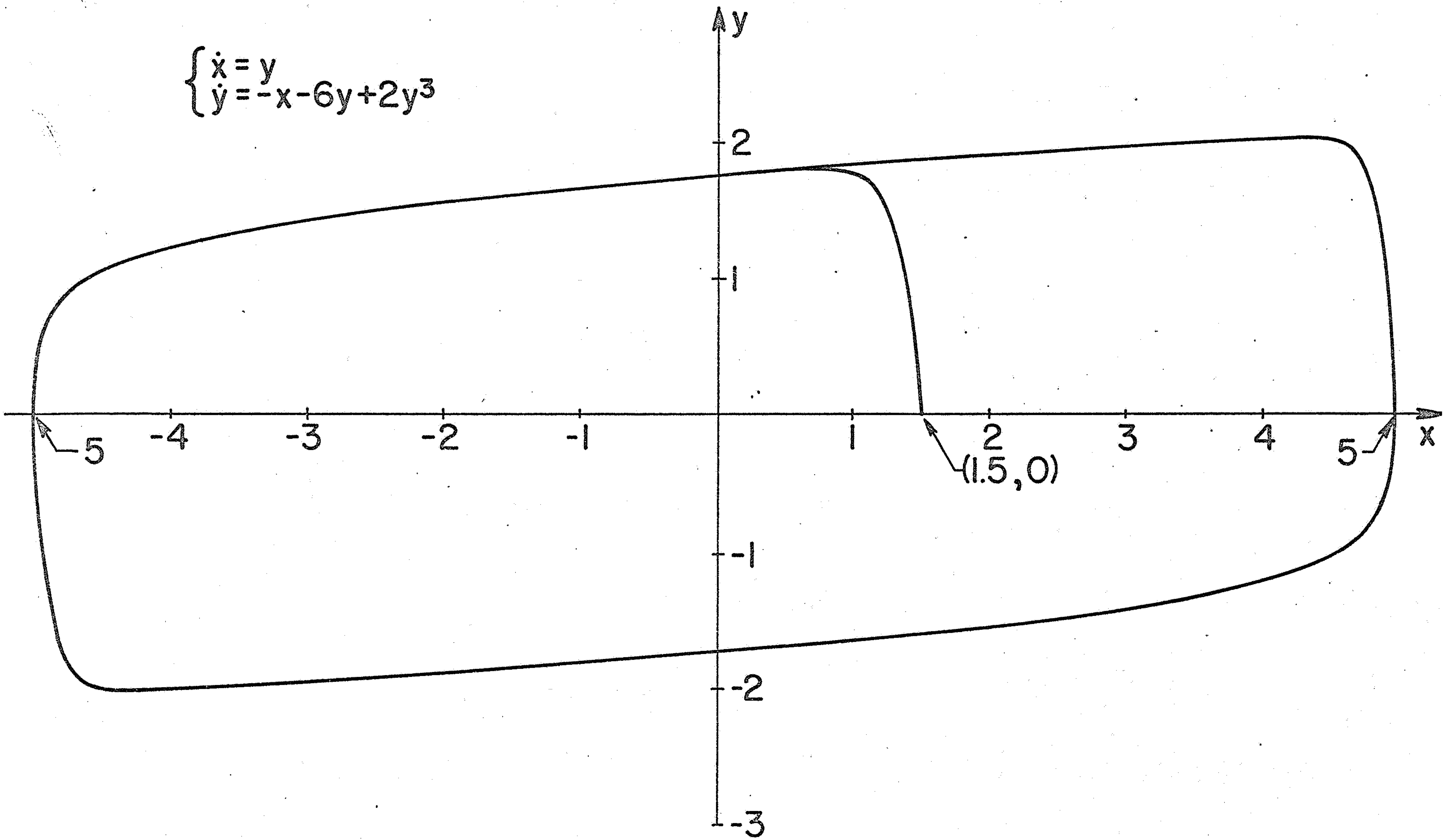
$\begin{cases} \dot{x} = y \\ \dot{y} = -x - 6y + 2y^3 \end{cases}$
y
x
2
1
-1
-2
-3
-5
-4
-3
-2
-1
1
2
3
4
5
(1.5,0)

Figure 11

rotated four times around the origin. Thus there exists a limit cycle around the origin and the domain inside the limit cycle is recognized as the domain of attraction. For this example, the procedure does no more, in effect, than determine the domain of attraction by plotting a reverse trajectory.

Example 2: The system is given as

$$\begin{cases} \dot{x} = y \\ \dot{y} = -x - y - \frac{1}{4}x^2. \end{cases} \tag{31}$$

(i) As

$$x \cdot N(x,0) = -x^2(1 + \tfrac{1}{4}x),$$

the condition of Theorem 1 is satisfied, for example, in $|x| < 0.125$.

(ii) A singular point exists at the origin and $x = -4$.

(iii) Assume

$$V(x,y) = x^2 + y^2 + \frac{x^3}{6}, \tag{32}$$

where the last term is included to insure $\dot{V}$ negative definite. A rough sketch of contours of (32) is shown in figure 12. For $V < \frac{16}{3}$ they become concentric closed curves around the origin. The time derivative under (31) is

$$\dot{V}(x,y) = -2y^2 < 0 \text{ if } y \neq 0.$$

From (31), no solution terminates on the x axis except at the singular points. Thus (32) is a Liapunov function proving asymptotical stability of the origin [1]. Choose L_v somewhat arbitrarily as

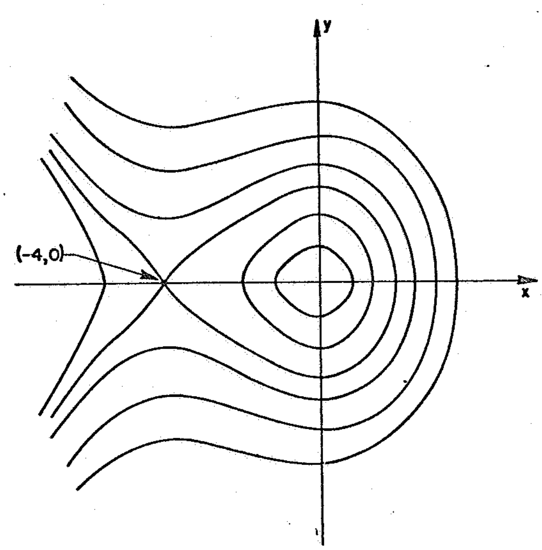

Figure 12

$$x^2 + y^2 + \frac{x^3}{6} = (0.3)^2,$$

which is completely included in D.

The result of a machine computation is shown in figure 13. The first reverse time trajectory, T_1, is started from $(-0.3,0)$. After this trajectory passes over the singular point at $(-4,0)$, the second trajectory, T_2, is started from $(-4+\varepsilon,0)$. When T_2 attains the preset maximum value of x, the third trajectory, T_3, is started from $(-4+\varepsilon,\delta)$. A magnification of T_2 and T_3 in the vicinity of $(-4,0)$ is shown in figure 14. Finally the region surrounded by T_2, T_3, $x = -4+\varepsilon$ and $x = -10.0$ is recognized to be a subset of D.

For comparison, figure 15 shows a set of trajectories of (31) calculated from the time reversed system with arbitrary selected initial states of

$$(0.01i,0), \quad i = 5,4,3,2,1,-1,-2,-3,-4,-5.$$

As an approximation to the domain of attraction evidently cannot be found by straight-forward calculations of reverse time trajectories in this example, the efficiency of this procedure is noticeable. Figure 16 is the union of figures 13 and 15.

Example 3: Given

$$\begin{cases} \dot{x} = y \\ \dot{y} = -4x - \frac{1}{2}y + \frac{1}{4}x^3. \end{cases} \tag{33}$$

(i) As

$$x \cdot N(x,0) = \frac{-x^2}{4}(16 - x^2),$$

the condition of Theorem 1 is satisfied, e.g., if $|x| \leq 1$.

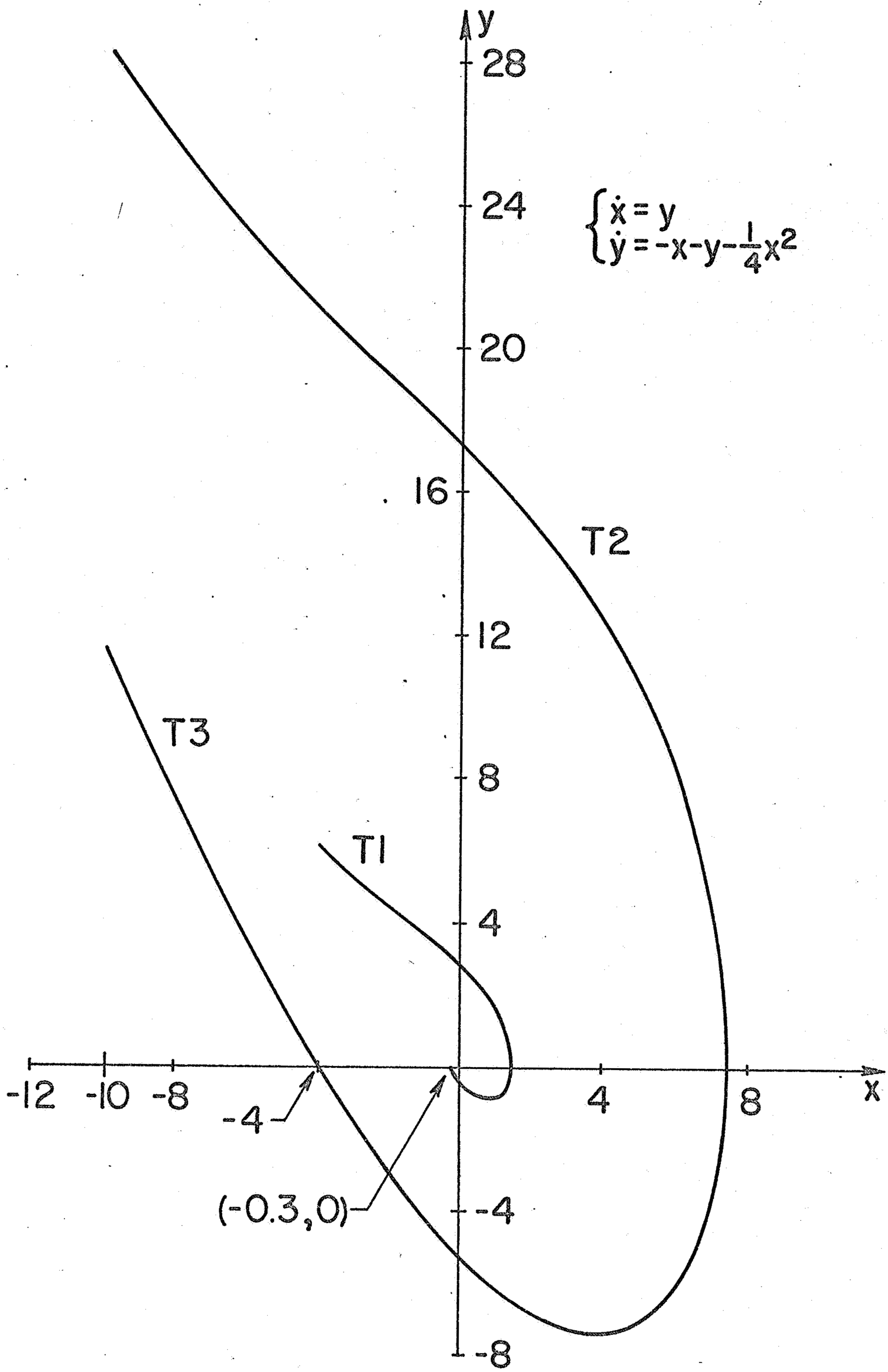
y
28
24
20
16
12
8
4
-4
-8
x
-12
-10
-8
-4
4
8
$\begin{cases} \dot{x} = y \\ \dot{y} = -x - y - \frac{1}{4}x^2 \end{cases}$
T1
T2
T3
(-0.3,0)

Figure 13

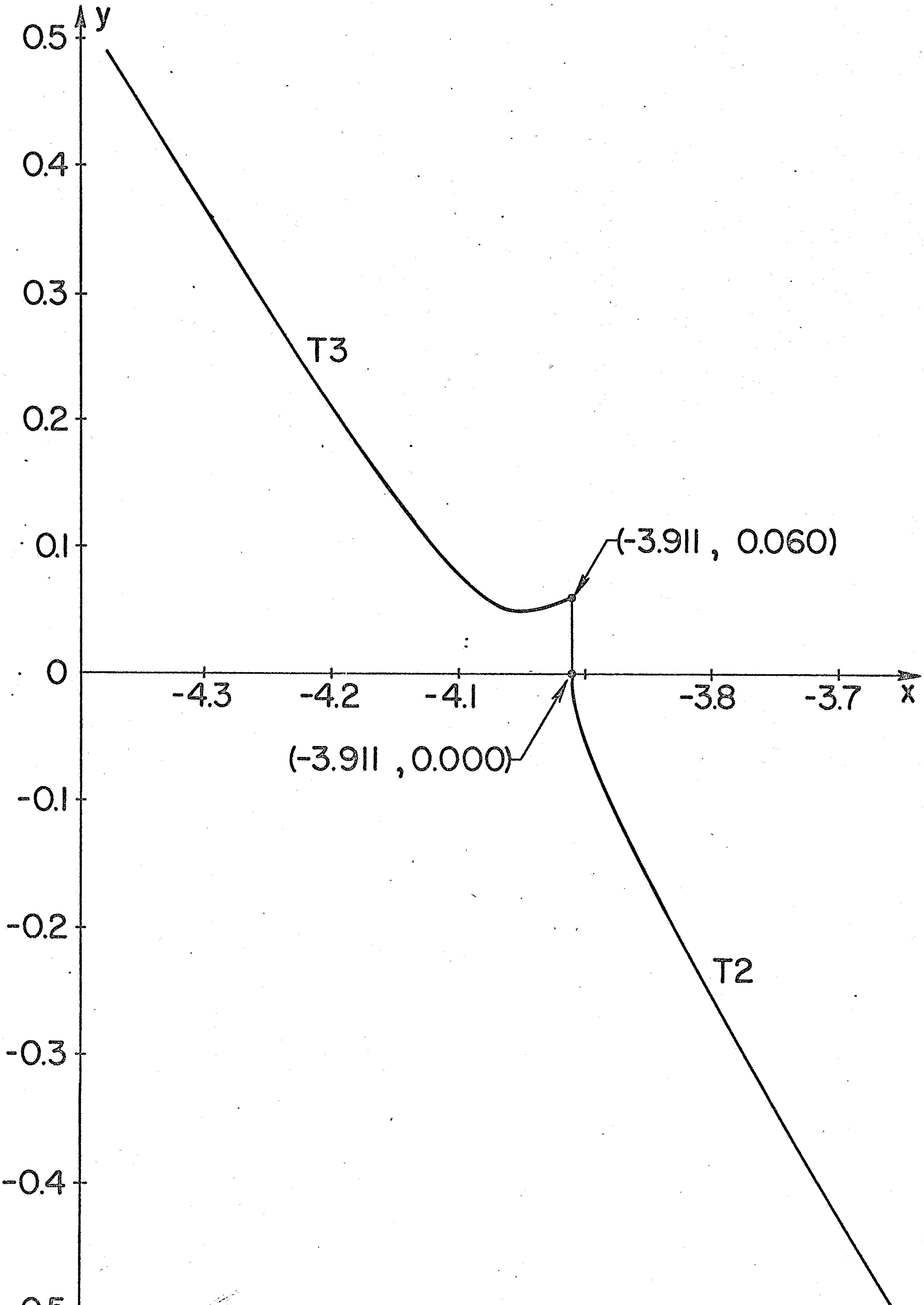
y
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-4.3
-4.2
-4.1
-3.8
-3.7
x
T3
T2
(-3.911 , 0.060)
(-3.911 , 0.000)

Figure 14

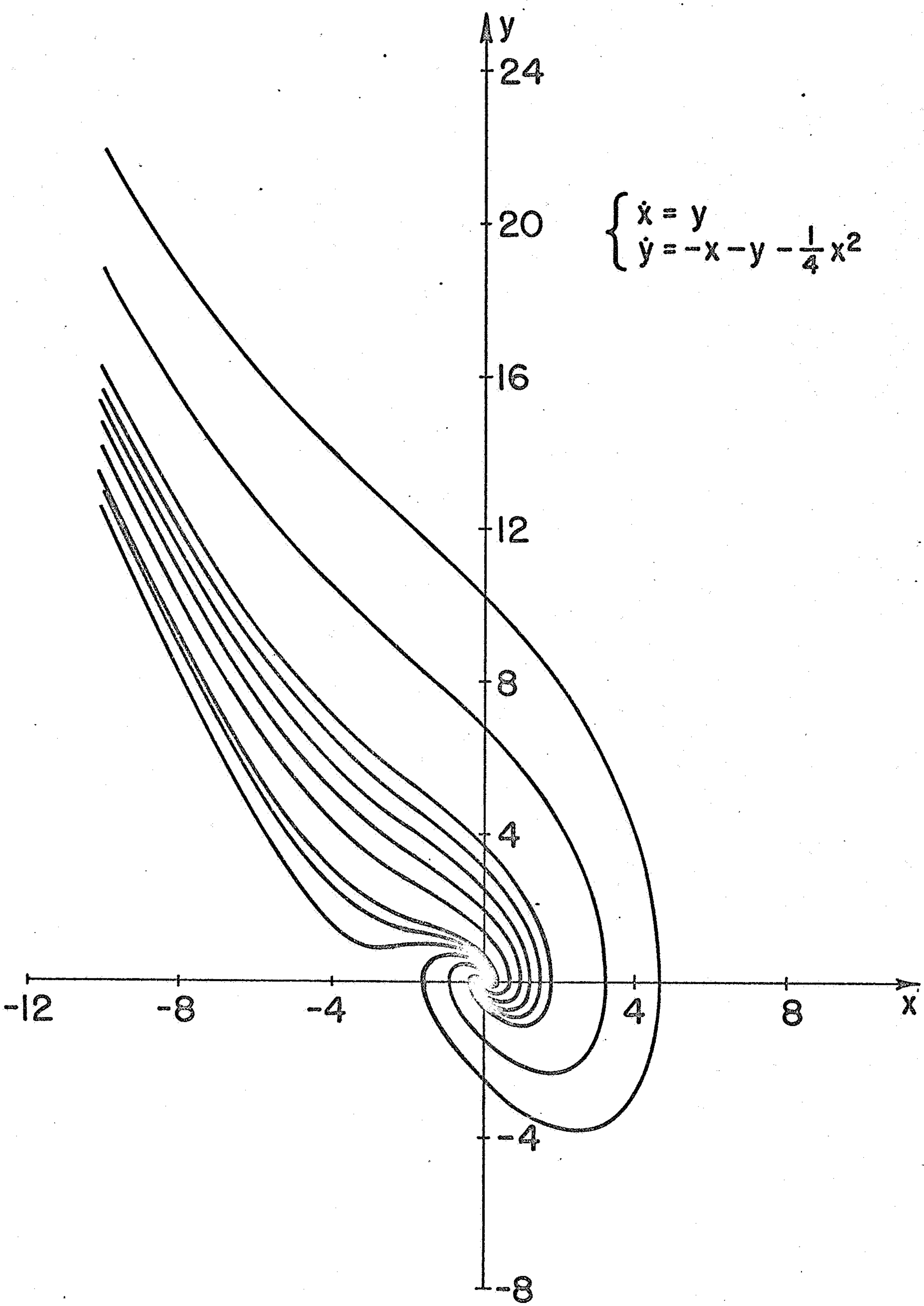
y
24
20
16
12
8
4
-4
-8
-12
-8
-4
4
8
x
$\begin{cases} \dot{x} = y \\ \dot{y} = -x - y - \frac{1}{4}x^2 \end{cases}$

Figure 15

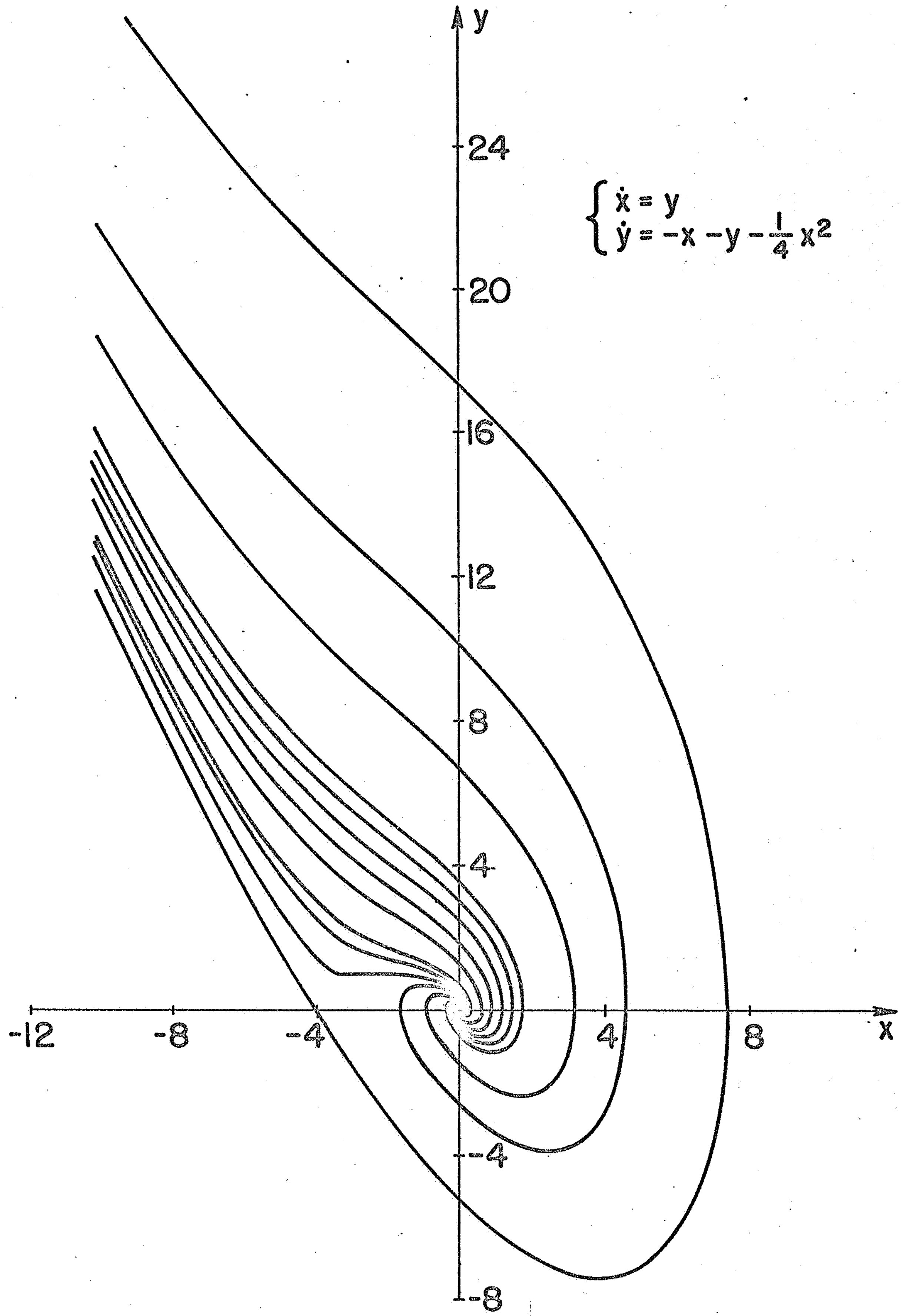
y
24
20
16
12
8
4
-4
-8
-12
-8
-4
4
8
x
$\begin{cases} \dot{x} = y \\ \dot{y} = -x - y - \frac{1}{4}x^2 \end{cases}$

Figure 16

(ii) Singular points are at $x = 4, 0, -4$.

(iii) Assume

$$V(x,y) = x^2 + \frac{y^2}{4} - \frac{1}{32}x^4. \tag{34}$$

A rough sketch of contours of (34) is shown in figure 17. For $V < 6$, these are closed curves around the origin. The derivative under (33) is

$$\dot{V}(x,y) = 2\dot{x}x + \frac{\dot{y}}{2}y - \frac{\dot{x}}{8}x^3$$

$$= -\frac{y^2}{4} < 0, \text{ if } y \neq 0. \tag{35}$$

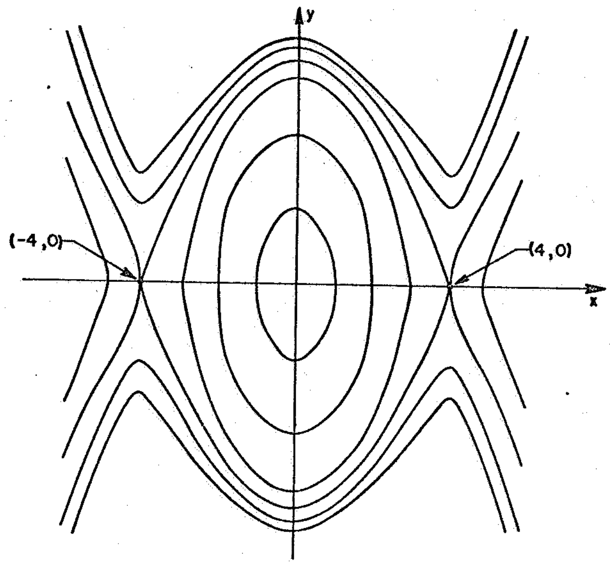

Figure 17

From (33), no solution trajectory terminates on the x axis except at the singular points. Therefore (34) is a Liapunov function proving asymptotic stability of the origin [1]. Choose L_V for convenience as

$$x^2 + \frac{y^2}{4} - \frac{1}{32}x^4 = \frac{31}{32},$$

which is completely included in D.

The result of the computation is shown in figure 18. The first reverse time trajectory, T_1, is started on L_V at $(-1.0, 0)$. After T_1 passes under the singular point at $(4,0)$, the second trajectory, T_2, is computed from $(4-\varepsilon, 0)$. Subsequently, T_2 passes over the singular point at $(-4,0)$. When T_2 attains a preset maximum, the third trajectory, T_3, is started at $(-4+\varepsilon, 0)$ and likewise terminates at a maximum value of x. The fourth and fifth trajectories are started at $(-4+\varepsilon, \delta)$ and $(4-\varepsilon, -\delta)$ respectively. Finally, the domain surrounded by $T_2 - T_5$, $x = 4-\varepsilon$, $x = -4+\varepsilon$ and $x = \pm 8$ is seen to be a subset of D.

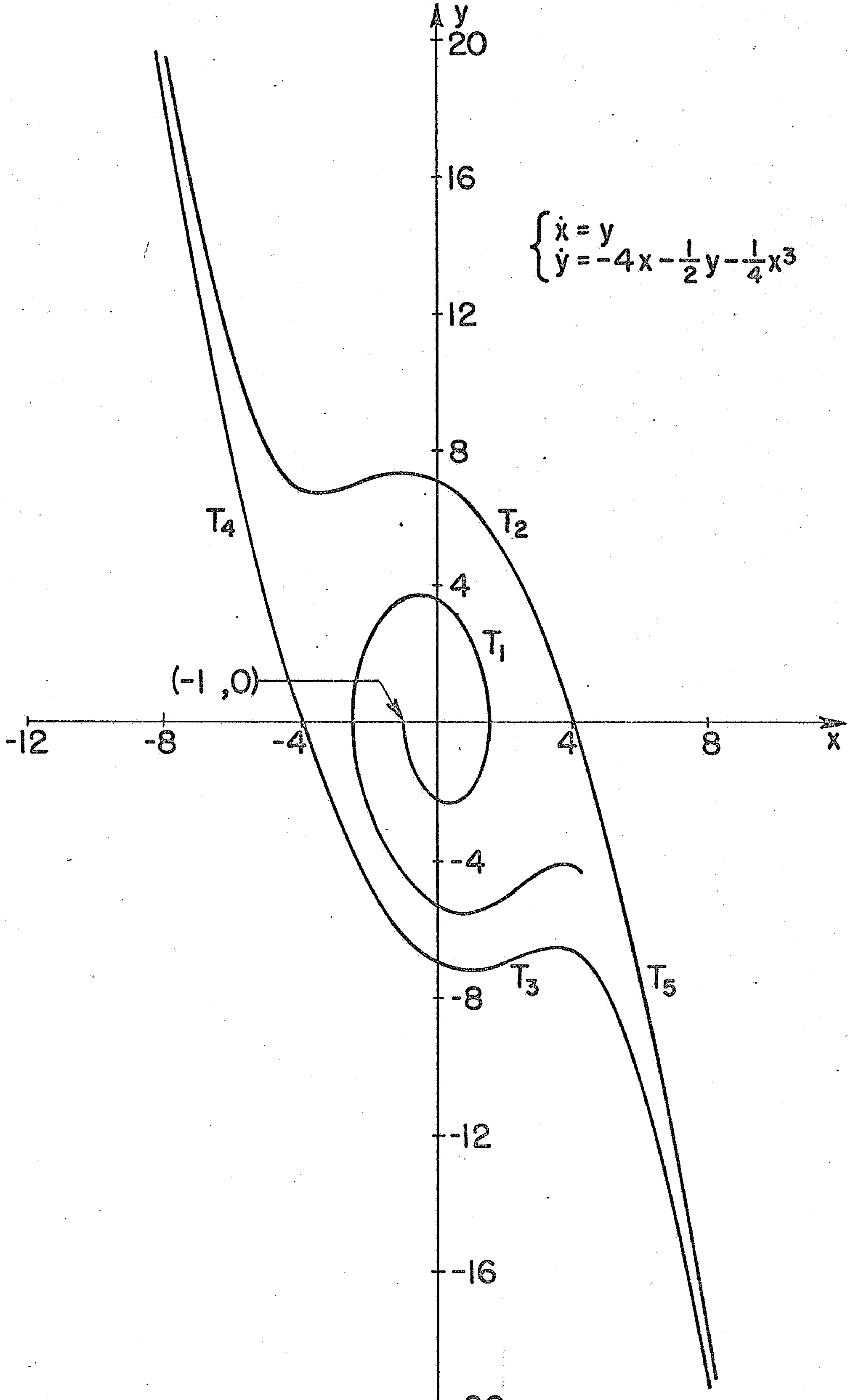
y
20
16
12
8
4
-4
-8
-12
-16
-20
-12
-8
-4
4
8
x
$\begin{cases} \dot{x} = y \\ \dot{y} = -4x - \frac{1}{2}y - \frac{1}{4}x^3 \end{cases}$
T_1
T_2
T_3
T_4
T_5
(-1 ,0)

Figure 18

VII. CONCLUSION

A procedure for determining an approximation to the domain of attraction of a general class of nonlinear differential equations (7) has been shown to be effective and efficient, and adaptable to machine assisted computation. This is illustrated by examples representing three types of domains of attraction.

The closeness of the approximation to the domain of attraction can be improved by choosing initial values for reverse time trajectories nearer to the singular points. This should be done with discretion, however, as the approximating domain can exceed the domain of attraction from accumulated errors in the trajectory computation, for a specific choice of Δt.

While the differential equations considered represent a limited class of second order equations, i.e., with the nonlinearity given as a finite order polynomial, it is expected that the procedure can be generalized somewhat. For other nonlinearities, the search for domains would have to be restricted to the regions where the uniqueness of the solution is guaranteed with the origin as an isolated singular point. The more general second order system

$$\begin{cases} \dot{x} = N_1(x,y) \\ \dot{y} = N_2(x,y) \end{cases}$$

could similarly be considered; however, the computation would become more complicated for singular point identification as the locations of singular points are no longer guaranteed on the x axis. Extensions to higher order nonlinear equations are limited in part by problems of representation.

VIII. REFERENCES

[1] LaSalle, J. and S. Lefschetz, Stability by Liapunov's Direct Method, Academic Press Inc., New York, N.Y., 1961, pp. 56-71.

[2] Mogolis, S.G. and W.G. Vogt, Control Engineering Applications of V.J. Zubov's Construction Procedure for Liapunov Functions, IEEE Trans. on Automatic Control, AC-8:104, 1963.

[3] Rodden, J.J., Applications of Liapunov Stability Theory, Doctoral Thesis, Stanford University, Stanford, California, 1964.

[4] Weissenberger, S., Stability Boundary Applications for Relay Control Systems Via a Steepest Ascent Construction of Liapunov Functions, Journal of Basic Eng., 88:419, 1966.

[5] Hahn, W., Theory and Application of Liapunov's Direct Method, Prentice Hall Inc., Englewood Cliffs, N.J., 1963.

[6] Minorsky, N., Nonlinear Oscillation, Van Nostrand, N.J., 1962, pp. 74-77.

[7] Weeg, G. and G. Weed, Introduction to Numerical Analysis, Blaisdall Pub. Co., Waltham, Mass., 1966, pp. 82-85.

APPENDIX

An IBM 360/50 was used for the computation described for the examples. The program, constructed according to the flow chart in figure 11, is listed starting on page 33.

The data statement requires

D: a small positive number for the trial value of δ in the vicinity of a singular point. As stated in item (3) of Section IV, the reverse time trajectory from $(\alpha-\varepsilon,\delta)$ [or $(\alpha+\varepsilon,-\delta)$] must pass over (or below) the singular point $(\alpha,0)$. To find the value of δ for each trajectory, a trial computation of a reverse time trajectory is made from $(\alpha-\varepsilon,D)$ [or $(\alpha+\varepsilon,-D)$] for 200 Δt segments. If the requirement for the trajectory cannot be satisfied, another trial computation is started from $(\alpha-\varepsilon,2D)$ [or $(\alpha+\varepsilon,-2D)$]. The trial trajectory computations are continued sequentially from $(\alpha-\varepsilon,kD)$ [or $(\alpha+\varepsilon,-kD)$] for 200 Δt segments until the requirements is satisfied, where k is a positive integer. Then δ is identified as the lower-bound k to satisfy the requirement. The number of 200 Δt segments for the trial trajectory is arbitrarily assumed.

E: a small positive number for ε.

H: a small positive number for Δt in equation (26).

P: a positive number to restrict the region of the state plane, as in (27).

SK: a small positive number for equation (28).

NUP: a positive integer for rotation of trajectories for stopping the trajectory calculation if a limit cycle is found.

NSN: the number of singular points, excluding the origin, in the left half plane of the search region.

NTS: the number of singular points, excluding the origin, in the search region plus 2.

SP(I): the x coordinate of the Ith singular point in the search region. I, beginning from 2, is numbered for each singular point except the origin from left to right sequentially. Thus SP(2) is the x coordinate of the left most singular point, SP(3) is that of the next to the right, etc. For purposes of the program, SP(1) = -P and SP(NTS) = P, although these are not singular points.

XO: the x coordinate of the initial state of the first reverse time trajectory.

YO: the y coordinate of the initial state of the first reverse time trajectory.

The compiling time of the program is about 4.50 seconds. Executing times of Examples 1-3 were as follows:

(i) Example 1: 20.5 seconds with $\Delta t = 0.01$ and $P = 10.0$.

(ii) Example 2: 26.8 seconds with $\Delta t = 0.01$ and $P = 10.0$.

(iii) Example 3: 25.3 seconds with $\Delta t = 0.01$ and $P = 8.0$.

```
*WATFOR
C     THE DOMAIN OF ATTRACTION FOR
C          X=Y
C          Y=(INSERT N(X,Y) OF THE EQUATION.)
C     THE CALCULATION CONDITIONS ARE
C          1. INCREMENT OF T = (INSERT INCREMENT OF T HERE.)
C          2. REGION OF SPACE , ABS(X) LESS THAN (INSERT P HERE.)
C     THE FIRST CARD OF EACH PROGRAM IS PUNCHED WITH F(X,Y)=-N(X,Y).
      F(X,Y)=(SEE COMMENT ABOVE.)
      F1(X)=-X
      DIMENSION SP(20)
      DIMENSION ISP(20)
      READ201,D,E,H,P,SK,NUP
201   FORMAT(5F15.7,I5)
      READ202,NSN,NTS
202   FORMAT(20I10)
      NTZ=NTS-1
      IF(NTS.EQ.2) GO TO 6
      READ203,(SP(I),I=2,NTZ)
203   FORMAT(20F15.7)
C     READ INITIAL STATE FOR FIRST TRAJECTORY.
6     READ204,XO,YO
204   FORMAT(2F15.7)
      SP(1)=-P
      SP(NTS)=P
      DO 10  I=1,NTS
      ISP(I)=6
10    CONTINUE
      LP=0
      NT=1
      J=NSN+1
C     SET INITIAL STATE AT (XO,YO) AND CALCULATE TRAJECTORY.
1     LP=LP+1
      S=0.0
      NS=0
      AP=XO
      AQ=YO
      NCY=0
      KTR=1
      PRINT211,LP,XO,YO
211   FORMAT(1HO,/////,5X,12H TRAJECTORY=,I3,5X,
     125H WITH THE INITIAL STATE (,E15.7,1X,E15.7,1X,2H).,///)
      PRINT212
```

```
212   FORMAT(1H0,17X,20HNO. OF ROTATIONS*0.5,8X,14HNO. OF SEGMENT,12X,
     110HX-POSITION,20X,10HY-POSITION,/)
      IF(NT.EQ.3) GO TO 2
      IF(NT.EQ.4) GO TO 3
      GO TO 4
2     TG1=X0
      GO TO 4
3     TJ1=X0
4     N=0
      IF(KTR.EQ.2) GO TO 5
      PRINT213,NCY,N,X0,Y0
213   FORMAT(1H,27X,I3,12X,I10,2E30.7)
      PUNCH220,LP,NCY,N,X0,Y0
220   FORMAT(2I5,I10,2F20.7)
5     N=N+1
      IF(KTR.EQ.2.AND.N.GE.200) GO TO 162
C     CALCULATION OF SEGMENTS OF TRAJECTORY
      P0X=H*F1(AQ)
      Q0Y=H*F(AP,AQ)
      P1X=H*F1(AQ+0.5*Q0Y)
      Q1Y=H*F(AP+0.5*P0X,AQ+0.5*Q0Y)
      P2X=H*F1(AQ+0.5*Q1Y)
      Q2Y=H*F(AP+0.5*P1X,AQ+0.5*Q1Y)
      P3X=H*F1(AQ+Q2Y)
      Q3Y=H*F(AP+P2X,AQ+Q2Y)
      XN=AP+(P0X+2.0*P1X+2.0*P2X+P3X)/6.0
      YN=AQ+(Q0Y+2.0*Q1Y+2.0*Q2Y+Q3Y)/6.0
      IF(KTR.EQ.2) GO TO 163
C     CHECK CHANGE OF SIGN OF Y.
      IF(AQ*YN.LT.0.0) GO TO 12
      NTR=1
      PRINT214,N,XN,YN
214   FORMAT(1H,42X,I10,2E30.7)
      GO TO 31
12    NTR=2
C     COUNT CHANGES OF SIGN OF Y FOR TRAJECTORY.
      NCY=NCY+1
      IF(NT.EQ.1) GO TO 18
      IF(NCY.EQ.1) GO TO 13
      IF(NCY.EQ.2.AND.NT.EQ.2) GO TO 16
      GO TO 18
13    IF(NT.EQ.2) GO TO 14
      IF(NT.EQ.3) GO TO 15
```

```
      TJ2=AP
      GO TO 18
14    TJ1=AP
      GO TO 18
15    TG2=AP
      GO TO 18
16    TJ2=AP
18    PRINT213,NCY,N,XN,YN
31    PUNCH220, LP,NCY,N,XN,YN
C     LOCATE POSSIBLE FUTURE INITIAL STATES.
      IF(XN.GT.SP(J).AND.SP(J+1).GE.XN) GO TO 51
      IF(AQ.GT.0.0) GO TO 19
      L=J+1
      GO TO 20
19    L=J
20    IF(NT.EQ.2) GO TO 32
      IF(LP.EQ.1) GO TO 35
      IF(NT.EQ.3.OR.NT.EQ.1) GO TO 36
      IF(NT.EQ.4) GO TO 38
      GO TO 45
32    IF(NCY.EQ.0.AND.ISP(L).EQ.5) GO TO 33
      IF(NCY.EQ.1.AND.X0*XN.GE.0.0) GO TO 39
      GO TO 45
33    IF(IS.EQ.1) GO TO 34
      ISP(L)=1
      GO TO 45
34    ISP(L)=2
      GO TO 45
35    ISP(L)=5
      GO TO 71
36    IF(ISP(L).EQ.6) GO TO 37
      GO TO 45
37    ISP(L)=5
      GO TO 45
38    IF(NCY.EQ.0.AND.X0*XN.GE.0.0) GO TO 39
      GO TO 45
39    IF(ISP(L).EQ.2.OR.ISP(L).EQ.1) GO TO 40
      GO TO 45
40    ISP(L)=0
45    IF(AQ) 46,51,47
46    J=J+1
      GO TO 48
47    J=J-1
```

```
48      IF(J.GT.O.AND.J.LT.NTS) GO TO 31
51      IF(NTR.EQ.1.OR.NT.EQ.1.OR.NCY.GE.3.OR.XO*XN.LT.0.0) GO TO 61
        IF(NT.EQ.2) GO TO 52
        IF(NCY.EQ.1) GO TO 67
        GO TO 61
52      IF(NCY.EQ.2) GO TO 67
C       CHECK END OF CALCULATION DUE TO EXCESS ROTATIONS.
61      IF(NCY.GT.NUP) GO TO 60
C       CHECK END OF CALCULATION DUE TO EXCESS VALUE OF X.
        IF(ABS(XN).LE.P) GO TO 69
        PRINT215
215     FORMAT(1H0,46HCHANGE OF TRAJECTORY DUE TO EXCESS VALUE OF X.,///)
        IF(NT.EQ.2.AND.NCY.EQ.0) GO TO 62
        GO TO 67
60      PRINT217
217     FORMAT(1H0,45HCHANGE OF TRAJECTORY DUE TO EXCESS ROTATIONS.,///)
        GO TO 67
62      IF(AQ.LT.0.0) GO TO 64
        DO 63 I=1,JK
        ISP(I)=0
63      CONTINUE
        GO TO 66
64      DO 65 I=JK,NTS
        ISP(I)=0
65      CONTINUE
66      IF(IS.EQ.1) GO TO 111
        GO TO 151
67      IF(NT.EQ.1) GO TO 71
        IF(IS.EQ.1) GO TO 68
        IF(NT.EQ.2) GO TO 121
        GO TO 131
68      IF(NT.EQ.2) GO TO 81
        GO TO 91
C       CHECK END OF CALCULATION DUE TO STEADY STATE.
69      IF(ABS(XN-AP).GT.ABS(YN-AQ)) GO TO 84
        S=S+ABS(YN-AQ)
        GO TO 85
84      S=S+ABS(XN-AP)
85      IF(N-100*NS.EQ.100) GO TO 86
        GO TO 88
86      NS=NS+1
        IF(S/(100*NS).LT.SK) GO TO 87
        S=0.0
```

```
      GO TO 88
87    PRINT216
216   FORMAT(1H0,41HCHANGE OF TRAJECTORY DUE TO STEADY STATE.,///)
      GO TO 67
88    AP=XN
      AQ=YN
      GO TO 5
C     FIND INITIAL STATE FOR NEXT TRAJECTORY.
71    IF(ISP(NSN+1).EQ.5) GO TO 72
      IF(ISP(NSN+2).EQ.5) GO TO 73
      IF(ISP(NSN+1).EQ.4) GO TO 75
      IF(ISP(NSN+2).EQ.3) GO TO 76
      GO TO 151
72    ISP(NSN+1)=4
      CE=1.0
      X0=SP(NSN+1)+CE*E
      GO TO 74
73    ISP(NSN+2)=3
      CE=-1.0
      X0=SP(NSN+2)+CE*E
74    J=NSN+1
      Y0=0.0
      NT=1
      KTR=1
      GO TO 1
75    ISP(NSN+1)=2
      K=NSN+1
      IS=1
      CE=1.0
      GO TO 77
76    ISP(NSN+2)=1
      K=NSN+2
      IS=2
      CE=-1.0
77    J=NSN+1
      JK=K
      X0=SP(K)+CE*E
      NT=2
      KTR=2
      GO TO 161
81    K=NSN+2
82    K=K-1
      IF(K.LT.1) GO TO 111
```

```
      IF(SP(K).LT.TJ1) GO TO 83
      GO TO 82
83    IF(ISP(K).NE.5) GO TO 111
      J=K
      KNT=K
      CE=1.0
      XO=SP(J)+CE*E
      YO=0.0
      NT=3
      ISP(J)=4
      KTR=1
      GO TO 1
91    IF(TJ2.GE.0.0) GO TO 95
      K=NSN+2
92    K=K-1
      IF(K.LE.0) GO TO 95
      IF(TG2.LE.0.0) GO TO 93
      IF(SP(K).GT.TJ2.AND.SP(K).LT.0.0) GO TO 94
      GO TO 92
93    IF(SP(K).GT.TJ2.AND.SP(K).LT.TG2) GO TO 94
      GO TO 92
94    IF(ISP(K).EQ.0) GO TO 92
      J=K-1
      CE=-1.0
      XO=SP(K)+CE*E
      ISP(K)=0
      NT=4
      IS=1
      KTR=2
      GO TO 161
95    J=KNT
      JK=J
      CE=1.0
      XO=SP(J)+CE*E
      ISP(KNT)=2
      NT=2
      IS=1
      KTR=2
      GO TO 161
111   IF(ISP(NSN+2).NE.3) GO TO 151
      K=NSN+2
      J=NSN+1
      JK=J
```

```
      CE=-1.0
      XO=SP(K)+CE*E
      ISP(K)=1
      NT=2
      IS=2
      KTR=2
      GO TO 161
121   K=NSN+1
122   K=K+1
      IF(K.GT.NTS) GO TO 151
      IF(SP(K).GT.TJ1) GO TO 122
123   IF(ISP(K).NE.5) GO TO 151
124   J=K-1
      CE=-1.0
      XO=SP(K)+CE*E
      ISP(K)=3
      YO=0.0
      NT=3
      IS=2
      KNT=K
      KTR=1
      GO TO 1
131   IF(TJ2.LE.0.0) GO TO 138
      K=NSN+1
132   K=K+1
      IF(K.GT.NTS) GO TO 138
      IF(TG2. LT.0.0) GO TO 133
      IF(SP(K).GT.TG2.AND.SP(K).LT.TJ2) GO TO 134
      GO TO 132
133   IF(SP(K).GT.0.0.AND.SP(K).LT.TJ2) GO TO 134
      GO TO 132
134   IF(ISP(K).EQ.0) GO TO 132
      J=K
      CE=1.0
      XO=SP(K)+CE*E
      ISP(KNT)=0
      NT=4
      IS=2
      KTR=2
      GO TO 161
138   J=KNT-1
      JK=KNT
      ISP(KNT)=1
```

```
      CE=-1.0
      XO=SP(KNT)+CE*E
      NT=2
      IS=2
      KTR=2
      GO TO 161
C     CHECK FOR REMAINING INITIAL STATES.
151   K=0
152   K=K+1
      IF(K.GT.NTS) GO TO 170
      IF(ISP(K).EQ.6.OR.ISP(K).EQ.0) GO TO 152
      IF(SP(K).GT.0.0) GO TO 153
      CE=1.0
      J=K
      IS=1
      ISP(K)=4
      GO TO 154
153   CE=-1.0
      J=K-1
      IS=2
      ISP(K)=3
154   XO=SP(K)+CE*E
      YO=0.0
      NT=3
      KNT=K
      KTR=1
      TJ2=P*CE
      GO TO 1
C     DETERMINATION OF VALUE FOR DELTA.
161   JC=0
162   JC=JC+1
      YO=CE*JC*D
      AP=XO
      AQ=YO
      GO TO 4
163   IF(CE*(XN-SP(K)).LE.0.0.AND.CE*YN.GT.0.0) GO TO 164
      IF(CE*(XN-SP(K)).GT.0.0.AND.CE*YN.GE.0.0) GO TO 5
      GO TO 162
164   KTR=1
      GO TO 1
170   PRINT218
218   FORMAT(1H0,24HCALCULATION IS COMPLETE.)
      STOP
```

```
      END
*DATA
            (INSERT DATA OF READ201.)
            (INSERT DATA OF READ202.)
            (INSERT DATA OF READ203.)
            (INSERT DATA OF READ204.)
*EOJ


              00000351
```