INTAKE: A NUMERICAL PROGRAM TO
CALCULATE FLUID VELOCITY PROFILES
NEAR A RECTANGULAR INLET IN A
STREAM WITH CROSS FLOW

L. Dresner

V. R. Cain

W. Davis, Jr.

MASTER

OAK RIDGE NATIONAL LABORATORY

OPERATED BY UNION CARBIDE CORPORATION • FOR THE U.S. ATOMIC ENERGY COMMISSION

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Contract No. W-7405-eng-26

Reactor Division

INTAKE: A NUMERICAL PROGRAM TO CALCULATE FLUID VELOCITY PROFILES
NEAR A RECTANGULAR INLET IN A STREAM WITH CROSS FLOW

L. Dresner

V. R. Cain

W. Davis, Jr.

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

JULY 1973

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
U.S. ATOMIC ENERGY COMMISSION

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

CONTENTS

	<u>Page</u>
Foreword	v
Symbols	vii
Abstract	1
1. Introduction	1
2. Theory	5
3. Operation of Intake	17
4. Illustrative Calculations and Discussion	22
Footnotes	31
Appendix - Source Listing of INTAKE.F4 and ISOMET	33

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

FOREWORD

Oak Ridge National Laboratory is providing technical assistance to the U. S. Atomic Energy Commission Directorate of Licensing in its analyses of the environmental impact of power reactors as required by the National Environmental Policy Act. The assessment of the environmental impact of a nuclear station has required the development of new analytical techniques and the expansion of many existing ones.

This document is one of the spin-off benefits of this effort by the Commission and its National Laboratories to protect the environment. The work reported here, while originally completed in support of the analysis of a particular nuclear plant, has wide application in other areas.

Other documents in this series are listed below:

1. L. Dresner, Steady Temperature Distributions in the Far-Field Region Obtained by Solution of the Equation of Convective Diffusion in Two Dimensions, USAEC Report ORNL-TM-4119, Oak Ridge National Laboratory.
2. D. A. Pilati, Transient Cooling of a One-Dimensional Thermal Plume and its Application for Determining Cold Shock, USAEC Report ORNL-TM-4160, Oak Ridge National Laboratory.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

SYMBOLS

a (A)	depth of river
b (B)	half-width of the intake in the vertical direction
c (C)	half-width of the intake in the horizontal direction
DS	time step in the Runge-Kutta integration
F, G	functions defined, respectively, by Eqs. (7) and (12)
NPNT	integer naming point on a streamline
NSTR	integer naming a streamline
Q	volumetric rate of withdrawal by the intake
u (U)	uniform flow velocity in the absence of withdrawal
$u_{1,2,3}$	x, y, z-components of flow velocity induced by withdrawal by the intake
x, y, z	coordinates shown in Fig. 1
x_1, x_2, y	coordinates shown in Fig. 1 used by isometric plotting routines
x_3 (X3)	x-coordinate of the center point of the intake
y_3 (Y3)	y-coordinate of the center point of the intake
$X4, Y4, Z4$	x, y, z-coordinates of the current point on the streamline being computed by INTAKE

INTAKE: A NUMERICAL PROGRAM TO CALCULATE FLUID VELOCITY PROFILES
NEAR A RECTANGULAR INLET IN A STREAM WITH CROSS FLOW

L. Dresner

V. R. Cain

W. Davis, Jr.

ABSTRACT

A conversational program named INTAKE for the PDP-10 Timesharing System is described. INTAKE calculates fluid flow streamlines near a rectangular intake in a uniformly deep stream with uniform flow. It may be used to estimate the entrainment of nonuniformly distributed plankton by the intake. The program produces isometric projections of selected streamlines on the Calcomp curve plotter. The theory of the computation is based on potential flow and is fully described. The report contains detailed instructions for operating the program. Illustrative results are exhibited and discussed in detail.

Keywords: Power plant, nuclear, intake, hydraulic analysis, flow stability, model, analytical model, computer program.

1. INTRODUCTION

One of the ways in which power plants affect the bodies of water on which they may be situated is by drawing in plankton with their cooling water. The plankton may be killed by the sudden temperature rise in the condensers, and the steady kill of plankton may in turn affect the numbers of all the organisms in the planktonic food chain.

The first step in assessing the extent of such changes is to estimate the rate of entrainment of plankton in the cooling water. If the plankton were uniformly distributed, and if the plankton did not swim or swam weakly, the entrainment could be calculated simply from the rate of withdrawal of cooling water. Many planktonic species spend different parts of the day near the top and bottom of the water, so that to calculate their entrainment it is necessary to know from what part of the body of water the cooling water is withdrawn. This means that the three-dimensional velocity field near the inlet must be calculated.

One typical situation for which velocity profiles may be desired is shown in Fig. 1. The box ABCDEFG represents a section of a river in which there is a uniform flow of velocity u in the direction parallel to AD. The plane ADFG is the bottom of the river; the plane BCE is the surface of the river. The depth a of the river is constant. The plane ABCD represents the left bank of the river, looking downstream. The rectangle HIJK in the plane ABCD, the sides of which are respectively parallel and perpendicular to the bottom of the river, represents a cooling water intake. The dimensions of the intake are $2b$ and $2c$ in the vertical and horizontal directions, respectively. The total volumetric rate of withdrawal through the intake is Q .

The program INTAKE calculates the velocity field in the vicinity of the intake on the basis of steady potential flow, treating the intake HIJK as a sink of uniform strength. Because the streamlines of the flow are space curves with both curvature and torsion, they are difficult to visualize without a graphic aid. Such an aid is provided by INTAKE in the form of isometric projections of selected streamlines shown in the box of Fig. 1. The selection of the streamlines is under the control of the user as will be explained in detail later. Typical examples of plots produced by INTAKE are shown in Figs. 7, 9, 10, and 11. Perpendiculars to the base plane are shown, allowing the coordinates of any point to be obtained directly from the isometric projection. The tick marks along the x-axis are separated by an interval equal to the length of the intake. The intake is located in the box so that one intake length is downstream of it and three are upstream of it. The tick marks on the y-axis

ORNL-DWG 73-2985

Fig. 1. Definition sketch.

divide the depth of the river into four equal parts. The interval between the tick marks on the z-axis equals $Q/3au$, and represents a compromise value that in most cases will insure that the interesting streamlines will fall in or near the box.

With some very minor reprogramming, INTAKE may also be used to visualize the velocity field created by a vertical pipe open at the top in a uniform ocean current. How this can be done is discussed at the end of Section 5.

2. THEORY

We establish the following set of coordinates in Fig. 1. We choose the direction of the main river flow to be the x-direction, we choose the vertical direction to be the y-direction, and we choose the z-direction to be perpendicular to the left bank. With these choices, x, y, and z form a right-handed set of coordinates as shown in Fig. 1. We choose the origin of coordinates to be at the bottom of the river directly under the upstream edge HK of the intake. The bottom of the river is thus the plane $y = 0$ and the left bank is the plane $z = 0$. The surface of the river is the plane $y = a$. We denote by y_3 the height of the center L of the intake above the bottom of the river; the coordinates of L are thus $(c, y_3, 0)$.

Having established these coordinates, we are ready to begin our calculation of the velocity field around the intake. Before studying the problem in its full generality, however, let us study a special case in which the flow field may easily be obtained by simple analytic techniques. The knowledge we gain from this special case will guide us in solving the general case. Let us imagine that b increases until it equals $a/2$, that c decreases to 0, while Q remains fixed at some specified value. The intake then becomes a line sink along the y-axis stretching from the bottom to the top of the river. As noted by Milne-Thomson,¹ such a straight line sink between two parallel planes can be regarded as a two-dimensional point sink. Milne-Thomson has solved the problem of such a point sink in a uniform stream and describes a graphical method due to Rankine of constructing the streamlines of the flow. The solution is symmetric around the line that passes through the point sink in the direction of flow. Therefore, as Milne-Thomson notes, this line can be replaced by a rigid wall, which in our case is the bank of the river.

Shown in Fig. 2 are streamlines constructed by Rankine's method for an intake withdrawing 3000 cfs from a 60-foot-deep river in which the uniform flow without the withdrawal is 1 fps. Fig. 2 is the view one would have looking at the box in Fig. 1 from underneath. The important feature of this flow field is the existence of a dividing streamline that

Fig. 2. Streamlines constructed by Rankine's method.

ends on a stagnation point on the bank. The existence of such a dividing streamline is connected with the fact that fluid particles moving downstream can suffer only two mutually exclusive fates: they are either withdrawn by the sink, or they pass forever downstream. Fluid particles on streamlines below the dividing streamline are withdrawn by the sink, fluid particles on streamlines above the dividing streamline pass downstream.

The streamlines spread apart sharply in the neighborhood of the stagnation point, those above the dividing streamline heading towards $x = \infty$, those below the dividing streamline entering the sink, as we move with the fluid. Conversely, as we move oppositely to the fluid, streamlines that are widely separated in the neighborhood of the stagnation point draw together sharply. Suppose now that we had a formula from which we could calculate the velocity components at any point in the plane. How then could we calculate the trajectory of the dividing streamline? If we knew the approximate location of the stagnation point, it would suffice to integrate the velocity components from a point nearby in the upstream direction. This would produce a streamline that would be quite close to the dividing streamline upstream of the stagnation point. How good an approximation this streamline would be to the dividing streamline could be checked by comparing the total flux under it with the withdrawal rate of the sink.

Even if we didn't know the approximate position of the stagnation point we could use the following variation of this procedure to approximate the dividing streamline. A streamline which is slightly above the dividing streamline upstream of the stagnation point will be slightly above the x -axis far downstream of the stagnation point. Hence if we choose a point far downstream of the stagnation point just off the x -axis and integrate the velocity components in the upstream direction, we should obtain a streamline just above the dividing streamline upstream of the stagnation point. (It is perhaps worth noting at this point that choosing the initial point of the integration on the x -axis creates a slight theoretical difficulty, for the z -component of velocity is zero on the x -axis and in the numerical integration one encounters difficulty in turning the corner at the stagnation point. However, choosing the initial point of the

integration just off the x -axis never creates any practical difficulty in approximating the dividing streamline as closely as we desire.)

The topology of the streamlines is more complicated in three dimensions than in two. We can ease our burden of understanding the three-dimensional flow field by first studying the streamlines in the plane $z = 0$ of Fig. 1. Owing to the symmetry with respect to reflection about the (x,y) -plane, the z -component of flow velocity vanishes in the (x,y) -plane. Thus a streamline any point of which lies in the (x,y) -plane, lies entirely in the (x,y) -plane. The fluid elements which flow along streamlines in the (x,y) -plane past the line $x = 0$ suffer one of four fates: they pass above the sink and enter it, they pass above the sink and go downstream, they pass below the sink and enter it, or they pass below the sink and go downstream. These four different kinds of fluid elements are separated by two dividing streamlines which intersect at right angles at a stagnation point S (see sketch in Fig. 3). At the stagnation point S the flow velocity is zero: the x -component of flow velocity due to the suction of the intake just balances the downstream flow velocity of the main current, and the component of velocity due to the top part of the intake just balances the component of velocity due to the bottom part of the intake. Finally, S is the only stagnation point in the flow field, for only in the (x, y) -plane can the z -component of flow velocity be zero.

Let us now consider fluid elements whose motion is not confined to the (x,y) -plane. Such fluid elements can be divided into two classes according as they are ultimately withdrawn by the sink or pass forever downstream. Fluid elements of the two classes are separated by a dividing stream tube as shown in Fig. 4. Points on the dividing stream tube lie on streamlines that neither enter the sink nor pass downstream. These streamlines must then end on a stagnation point. The only stagnation point being S , all the streamlines of which the dividing stream tube is composed must end on S , as shown in Fig. 4.

Let us now consider a streamline just outside the dividing stream tube but very close to it. As we proceed in the direction of fluid flow, the streamline at first approaches the stagnation point, but eventually diverges from it, proceeding downstream close to the (x,y) -plane. The

ORNL-DWG 73-2987

Fig. 3. Sketch of the Streamlines in the (x,y) -plane.

ORNL-DWG 73-2988

Fig. 4. Sketch showing the dividing stream tube and some streamlines on it.

closer we choose the streamline to the dividing stream tube to begin with, the smaller its distance of closest approach to S and the smaller its asymptotic separation from the (x,y) -plane. Thus in order to approximate the dividing stream tube, we can choose points just off the (x,y) -plane far downstream of the intake, and integrate the velocity components in the upstream direction. Different choices of y for the initial point will give us different streamlines on the dividing stream tube. However, not all choices of y for the initial point will give us streamlines on the dividing stream tube. Some choices will give us stream lines lying near the (x,y) -plane above or below the dividing stream tube. Thus some trial and error may be necessary in the course of any practical calculation.

With this procedure it should be possible to approximate the curve PRQ in Fig. 4, which represents the boundary of the intersection of the dividing stream tube with a plane parallel to the (y,z) -plane far upstream. It is this curve which tells us where the water entering the intake comes from. The whole calculation can be checked by comparing the water flow through the section PRQ with the flow into the intake.

Having established the method by which we would approximate the dividing stream tube if we knew the flow velocity at every point, we must now calculate the flow velocity. The velocity potential we seek is the sum of the potential of a uniform flow in the x -direction and the potential due to the rectangular sink centered on the point L . The latter potential must be calculated taking into account the conditions that the y -component of flow velocity must vanish on the planes $y = 0$ and $y = a$, and that the z -component of flow velocity must vanish on the plane $z = 0$. These conditions can be taken into account by calculating the potential due to the sink at L (Fig. 1) as though it were in an infinite medium, and adding to it the infinite-medium potentials due to identical rectangular sinks centered on certain image points to be specified below.² The infinite-medium potential due to a rectangular sink can be calculated as the limit as $N \rightarrow \infty$ of the sum of the infinite-medium potentials due to a uniform rectangular array of N point sinks, each of strength Q/N . The latter potentials are proportional to the reciprocal distance from the sink.

Let us now imagine that the point lies not in the (x,y) -plane, but lies instead slightly above it in the positive z -direction. Let the

coordinates of L be (c, y_3, ϵ) . The images of L then lie at the points $(c, y_3, -\epsilon)$, $(c, 2a - y_3, \pm \epsilon)$, $(c, y_3 \pm 2na, \pm \epsilon)$, and $(c, 2a - y_3 \pm 2na, \pm \epsilon)$, where $n = 1, 2, 3, \dots$. All of the images of L are outside the region $0 \leq y \leq a$, $z \geq 0$, as they should be; and as is evident from the sketch of Fig. 5, the planes $y = 0$, $y = a$, and $z = 0$ are planes of reflection symmetry, on which the normal component of velocity vanishes. Owing to the fact that $\epsilon \neq 0$, the sink at L is not quite in the right place, so we must let $\epsilon \rightarrow 0$. When $\epsilon \rightarrow 0$, L and its images coalesce in pairs. This means that the potential of the rectangular sink that we seek is the sum of the infinite-medium potentials due to rectangular sinks of strength $2Q$ located at the points $(c, y_3 \pm 2na, 0)$ and $(c, 2a - y_3 \pm 2na, 0)$, $n = 0, 1, 2, 3, \dots$.

Let us now calculate the infinite-medium velocity potential of a rectangular sink of total strength $2Q$. For convenience, let us consider the sink to be centered on the origin and to lie in the (x, y) -plane. Later, when necessary, we can obtain the infinite-medium potential due to a sink centered on L or one of its images by performing a simple transformation of coordinates. The infinite-medium velocity potential $\phi(x, y, z)$ of a point sink of strength q located at the point (x', y', z') is

$$\phi(x, y, z) = - [q/4\pi] [(x-x')^2 + (y-y')^2 + (z-z')^2]^{-1/2} \quad (1)$$

The infinite-medium velocity potential of a rectangular sink of dimensions $2c \times 2b$ and strength $2Q$ lying in the (x, y) -plane and centered on the origin is then

$$\phi(x, y, z) = - [Q/8\pi bc] \int_{-c}^{+c} dx' \int_{-b}^{+b} dy' [(x-x')^2 + (y-y')^2 + z^2]^{-1/2} \quad (2)$$

The straightforward evaluation of ϕ using Eq. (2) will not avail us much for the following reason. The contribution to the potential at P: (x, y, z) due to a very remote image sink is roughly $-Q/2\pi r$, where r is the distance from P to the center of the image sink. The images of L, as we have seen above, consist of two sets of points, the points

ORNL-DWG 73-2989

Fig. 5. Sketch of the image system.

of each set being equidistantly spaced and stretching to infinity in either direction. The remote points of each set contribute a harmonic series to the sum-over-images, i.e., a sum to infinity of terms of the form $1/2na$. Such a series does not converge, which means that the expression obtained for the potential by summing Eq. (2) over all the images does not converge.

The lack of convergence of the series for the potential is not a drawback, however, since it is the velocity components rather than the potential that we seek. The velocity components due to the remote images fall off as r^{-2} rather than as r^{-1} , and the contribution of the remote points forms a convergent series. From (2) we obtain by differentiation

$$u_{\begin{Bmatrix} 1 \\ 2 \\ 3 \end{Bmatrix}} = [Q/8\pi bc] \int_{-c}^{+c} dx' \int_{-b}^{+b} dy' [(x-x')^2 + (y-y')^2 + z^2]^{-3/2} \begin{Bmatrix} x'-x \\ y'-y \\ -z \end{Bmatrix} \quad (3)$$

where u_1, u_2, u_3 are the x-, y-, and z-components of velocity, respectively, at the point (x, y, z) in an infinite medium due a rectangular sink of dimensions $2c$ and $2b$ and strength $2Q$ lying in the (x, y) -plane and centered on the origin. Let us first calculate u_1 . The y' -integration can be carried out with the help of the identity $\int (t^2 + k^2)^{-3/2} dt = (t/k^2) \times (t^2 + k^2)^{-1/2}$, and yields

$$u_1 = \frac{Q}{8\pi bc} \int_{-c}^{+c} dx' \frac{x'-x}{(x'-x)^2 + z^2} \left[\frac{b-y}{\sqrt{(x'-x)^2 + (b-y)^2 + z^2}} + \frac{b+y}{\sqrt{(x'-x)^2 + (b+y)^2 + z^2}} \right] \quad (4)$$

The x' -integration can be carried out with the help of the identity

$$\int \frac{t}{(t^2 + k^2) \sqrt{t^2 + \ell^2}} dt = \frac{1}{2 \sqrt{\ell^2 - k^2}} \ln \left[\frac{\sqrt{t^2 + \ell^2} - \sqrt{\ell^2 - k^2}}{\sqrt{t^2 + \ell^2} + \sqrt{\ell^2 - k^2}} \right] \quad (5)$$

and yields

$$u_1 = [Q/8\pi bc] [F(x,y,z,b,c) + F(x,-y,z,b,c) - F(-x,y,z,b,c) - F(-x,-y,z,b,c)] \quad (6)$$

where the function $F(x,y,z,b,c)$ is given by

$$F(x,y,z,b,c) = \frac{1}{2} \frac{y-b}{|y-b|} \ln \left[\frac{\sqrt{(x-c)^2 + (y-b)^2 + z^2} + |y-b|}{\sqrt{(x-c)^2 + (y-b)^2 + z^2} - |y-b|} \right] \quad (7)$$

Formulas (6) and (7) can be used to calculate u_2 as well as u_1 . If in Eq. (3) written for u_1 (first line) we interchange x and y , interchange b and c , rename the dummy variables of integration x' and y' with the new names y' and x' , respectively, and interchange the order of integration, we obtain Eq. (3) written for u_2 (second line). Thus,

$$u_2(x,y,z,b,c) = u_1(y,x,z,c,b) \quad (8)$$

To calculate u_3 , we first integrate the third line of Eq. (3) as we did the first line using the first identity given above. We obtain

$$u_3 = [Q/8\pi bc] (-z) \int_{-c}^{+c} \frac{dx'}{(x'-x)^2 + z^2} \left[\frac{b-y}{\sqrt{(x'-x)^2 + (b-y)^2 + z^2}} + \frac{b+y}{\sqrt{(x'-x)^2 + (b+y)^2 + z^2}} \right] \quad (9)$$

We perform the x' -integration using the identity

$$\int \frac{dt}{(t^2 + k^2) \sqrt{t^2 + \ell^2}} = \frac{1}{k \sqrt{\ell^2 - k^2}} \tan^{-1} \left(\frac{t \sqrt{\ell^2 - k^2}}{k \sqrt{t^2 + \ell^2}} \right) \quad (10)$$

and obtain

$$u_3 = [Q/8\pi bc] [G(x,y,z,b,c) + G(x,-y,z,b,c) + G(-x,y,z,b,c) + G(-x,-y,z,b,c)] \quad (11)$$

where

$$G(x,y,z,b,c) = \frac{z(y+b)}{|z(y+b)|} \tan^{-1} \left(\frac{(x-c) |y+b|}{|z| \sqrt{(x-c)^2 + (y+b)^2 + z^2}} \right) \quad (12)$$

Using the expressions given in Eqs. (6, 8, 11) for u_1 , u_2 , and u_3 , we can find points on any streamline by integrating the coupled ordinary differential equations

$$dx/dt = u_1(x,y,z) + u \quad (13a)$$

$$dy/dt = u_2(x,y,z) \quad (13b)$$

$$dz/dt = u_3(x,y,z) \quad (13c)$$

In the program INTAKE this integration is performed by the Runge-Kutta scheme.

3. OPERATION OF INTAKE

The reader is assumed to be familiar with the procedure for logging in on the ORNL PDP-10 Timesharing System. After logging in, the command `.RUN INTAKE` begins the operation of the program INTAKE. (The symbol `↵` means a carriage return on the teletype.) The program responds with the question "WANT TO RECOVER FOR27.DAT?" This question refers to the possibility of recovering data saved from previous calculations, as will be described in greater detail below. To begin entirely fresh, this question should be answered "NO". The program then responds with the question "A, B, C, Q, Y3, U=?" and the user supplies these numbers on one line, separated by commas, and terminated by a carriage return.* The numbers can be specified in any consistent set of units of length and time. This means, for example, that if the lengths A, B, C, and Y3 are specified in feet, and the velocity U in ft sec^{-1} , then the withdrawal rate Q of the sink must be specified in $\text{ft}^3 \text{sec}^{-1}$.

After accepting these numbers, the program responds with the statement "READY FOR STARTING POINT AND STEP SIZE". The user now responds by typing four numbers on one line, separated by commas, and terminated by a carriage return. The first three of these numbers, called X4, Y4, and Z4 in the program, are the coordinates of a point on the streamline we are about to compute. The last number, called DS in the program, is the time step in the Runge-Kutta integration of Eq. (13). If it is positive, the integration advances the streamline in the direction of flow; if it is negative, the integration advances the streamline in the direction opposite to that of the flow. After accepting these numbers, the program prints out the values of X4, Y4, and Z4 as well as two integers. The first of these integers, NSTR, labels the streamline we are on; the second, NPNT(NSTR), labels the points on the streamline. After printing these numbers the program pauses and waits for any one of the twelve acceptable commands which may be given to it. These twelve commands are displayed below in Table 1; an explanation follows the display.

* The numbers A, B, C, Q, Y3, and U, shown capitalized here to resemble teletype output as closely as possible, represent the variables a, b, c, q, y3, and u of the previous section.

Table 1. Twelve Commands Accepted by INTAKE

Command	Comment
$\rangle$	The symbol $\rangle$ represents a carriage return
I $\rangle$	I is any integer
DS = xx.xx $\rangle$	xx.xx is any number (G format)
X4 = xx.xx $\rangle$	xx.xx is any number (G format)
Y4 = xx.xx $\rangle$	xx.xx is any number (G format)
Z4 = xx.xx $\rangle$	xx.xx is any number (G format)
DISCARD $\rangle$	
BACK $\rangle$	
NEW $\rangle$	
SAVE $\rangle$	
PLOT $\rangle$	
QUIT $\rangle$	

The command $\rangle$ causes the Runge-Kutta integration to advance one step. After completing the integration step, the program prints out the new values of X4, Y4, Z4, NSTR, and NPNT(NSTR).

The command I $\rangle$ causes the command $\rangle$ to be executed I times.

The command DS = xx.xx $\rangle$ causes the number xx.xx to become the current value of DS. After accepting the number, the program echoes the command by typing "DS = xx.xx". This allows the user to change the time step in the Runge-Kutta integration during the course of computation of one streamline. This command can be used to save computing time since a larger time step can be used on the straighter portions of a streamline than on the more curved portions.

The commands X4 = xx.xx $\rangle$, Y4 = xx.xx $\rangle$, and Z4 = xx.xx $\rangle$ cause the respective numbers xx.xx to become the current values of X4, Y4, and Z4. After each command the program echoes the command. These commands are used in connection with the commands DISCARD $\rangle$, BACK $\rangle$, and NEW $\rangle$.

In order to understand the effects of the commands DISCARD), BACK), and NEW), it is necessary to know how the computed points on the various streamlines are stored by the program prior to plotting. The points on the various streamlines calculated by the Runge-Kutta integration are stored in three 50 x 20 arrays names XPLT, YPLT, and ZPLT. X_4 is stored in XPLT, Y_4 is stored in YPLT, and Z_4 is stored in ZPLT. If we think of corresponding portions of XPLT, YPLT, and ZPLT together, we can think of the calculated points as being stored in 20 groups of 50 each (see sketch in Fig. 6). Let us for the moment assume that each group of 50 points represents one streamline, and let us suppose that the last point computed by the Runge-Kutta integration scheme is stored at point P as shown in Fig. 6. The command DISCARD) then destroys all the points of the current streamline. The next computer point will then be stored in location P' as shown in Fig. 6. The command BACK) destroys the points of the current streamline as well as the previous one (except when NSTR = 1, when the effect of BACK) and DISCARD) is the same). The next computed point will be stored in P''. The command NEW) ends the computation of the current streamline and allows us to begin the next. The next computed point will be stored in P''.

It is usual to assign new values of X_4 , Y_4 , and Z_4 after executing one of the orders BACK), DISCARD), or NEW). However, if this is not done, the values of X_4 , Y_4 , and Z_4 , which are stored elsewhere in the program besides the arrays XPLT, YPLT, and ZPLT, are the values referring to P, the last point computed.

From this description it might appear that no streamline could be composed of more than 50 points. This apparent difficulty is handled in the following way. Whenever NPNT reaches 50, the coordinates of the last point computed are stored not only in the fiftieth location in the current group of 50 points, but also in the first location of the next group of 50 points. In this way a "new" streamline is created the first point of which is the same as the last point of the "old" streamline. When plotted, these two streamlines appear as a single smooth curve.

ORNL-DWG 73-2990

Fig. 6. Schematic diagram illustrating the commands BACK, DISCARD, and NEW.

The command `SAVE` can be used at any time to store on disk the values of `A`, `B`, `C`, `Y3`, `U`, the current values of `X4`, `Y4`, `Z4`, and `DS`, and the contents of the arrays `XPLT`, `YPLT`, and `ZPLT`. This data is saved in a file named `FOR27.DAT`. The command `SAVE` overwrites the file created by the previous `SAVE` command, so that the contents of the file `FOR27.DAT` are the data at the time of the last `SAVE` command. To recover the data from the file `FOR27.DAT` one should give the command `.RUN INTAKE` and answer the program's first question, "WANT TO RECOVER FOR27.DAT?", with `YES`. Thereafter one may give any of the twelve commands shown in Table 1. The `SAVE` command is intended to be used to protect the results of lengthy computations from inadvertent loss, for example, as might occur in the case of machine trouble in which the contents of the core are lost.

When the user judges sufficiently many streamlines have been calculated, he may cause an isometric projection to be drawn by the Calcomp Curve Plotter by giving the command `PLOT`. The isometric projection is drawn by a set of plotting subroutines written by W. Davis, Jr.³ The command `PLOT` causes the creation of a scratch plotting file `FOR28.DAT`. Directly following this command the user will be asked if he wishes to keep this plot; only the answer `Y` will cause the scratch file to be transferred to the permanent plotting file `FOR29.DAT`. A `PLOT` command overwrites previous `FOR28.DAT` files, but successive `Y` answers add successive scratch files to the permanent file `FOR29.DAT`. After leaving `INTAKE`, the user should give the command `.Q PLT: = FOR29.DAT/DISP:RENAME`, which will queue the permanent plotting file to the plotter queue and remove it from the user's disk area. For more information about Calcomp plotting, the user may give the command `.HELP PLOT` in the monitor mode.

The command `QUIT` should be used to end the running of `INTAKE`.

After each of the twelve commands except `QUIT`, the program pauses after executing the command and waits for another of the twelve commands. In the event that an incorrect command is given, the program returns the error message "DIDN'T UNDERSTAND, PLEASE RETYPE." and waits to accept one of the twelve commands.

4. ILLUSTRATIVE CALCULATIONS AND DISCUSSION

Figure 7 shows three streamlines calculated for a typical intake, 26' (vert) x 80' (hor), withdrawing 3000 cfs from a river 60' deep. The stream velocity in the absence of withdrawal is 1 fps. The top of the intake is at the surface of the river so that for this case $y_3 = 47'$. From the streamlines given in Fig. 7 one may determine the curve of intersection of the dividing stream tube with a plane parallel to the (y,z) -plane far upstream. The furthest upstream position shown is $x = -240'$. The three streamlines shown pierce the plane $x = -240'$ in the points $(-240', 6', 2.5')$, $(-240', 14', 39')$, and $(-240', 60', 63')$. These three points are shown plotted in the (y,z) -plane in Fig. 8. Owing to the symmetry of the situation shown in Fig. 7, the dividing stream tube must intersect the planes $y = a$ and $z = 0$ at right angles. These requirements plus the three points allow us to plot the curve shown fairly unambiguously. A simple integration shows that the area between the curve and the y -axis is 2800 ft^2 . Further calculations using INTAKE modified to print out the velocity components u_1 , u_2 , and u_3 gives the result that at the nine points marked with crosses in Fig. 8, $u_1 = 1.055$ fps within one part in a thousand. The total volumetric flux through this area is then 2950 cfs. The total flux of course 3000 cfs in this example; the error of about 1.6% is unquestionably due to the crudeness of the graphical procedure.

Figure 9 shows the same intake in a river twice as deep, everything else the same as in Fig. 7. Figure 10 shows streamlines for the same intake in a river with twice the uniform flow, everything else the same as in Fig. 7. Figure 11 shows an intake twice as long and half as wide in the same river as that of Fig. 7. Summarized in Fig. 12 are the parameters of all four cases together with the intersections of the respective dividing stream tubes with the plane $x = -240'$. For convenience, the ordinate in Fig. 12 is the depth below the surface, $a-y$ rather than the height y above the floor. Comparison of cases 7 and 11, for example, shows the effect of a change in the shape of the intake; the effect is slight. Comparison of cases 7 and 9 show the effect of deepening the river. The effect on the dividing stream

ORNL DWG 73-5467

Fig. 7. Illustrative streamlines calculated for the following parameters: $A = 60$ ft, $B = 13$ ft, $C = 40$ ft, $Q = 3000$ cfs, $U = 1$ fps.

ORNL-DWG 73-2991

Fig. 8. Intersection of the dividing stream tube of
Fig. 7 with the plane $x = -240'$.

Fig. 8. Intersection of the dividing stream tube of Fig. 7 with the
plane $x = -240'$.

Fig. 9. Illustrative streamlines calculated for the following parameters: $A = 120$ ft, $B = 13$ ft, $C = 40$ ft, $Q = 3000$ cfs, $U = 1$ fps.

ORNL DWG 73-5469

Fig. 10. Illustrative streamlines calculated for the following parameters: $A = 60$ ft, $B = 13$ ft, $C = 40$ ft, $Q = 3000$ cfs, $U = 2$ fps.

ORNL DWG 73-5470

Fig. 11. Illustrative streamlines calculated for the following parameters: $A = 60$ ft, $B = 6.5$ ft, $C = 80$ ft, $Q = 3000$ cfs, $U = 1$ fps.

ORNL-DWG 73-2992

Fig. 12. Intersection of the dividing stream tube with the plane $x = -240'$ for the illustrative calculations of Figs. 7, 9, 10, and 11.

tube is comparatively small, although in case 9, virtually no water is drawn from the bottom half of the river. Comparison of cases 7 and 10 shows the effect of the uniform flow velocity on the dividing stream tube; interestingly, raising the velocity has less effect on the vertical extension of the dividing stream tube than on its lateral extension.

It would be interesting to inquire what the effect of decreasing Q would be. We need not do any further calculations to answer this question because the answer is already given by comparison of cases 7 and 10, as we can see from the following dimensional argument. The only dimensionless groups that can be formed from the independent parameters a , b , c , Q , and u ($y_3 = a - b$ in cases 7, 9, 10, and 11) are b/a , c/a , and Q/a^2u . Hence the equation of the dividing stream tube must have the form $F(x/a, y/a, z/a, b/a, c/a, Q/a^2u) = 0$. If we keep all the lengths the same, changing u by a given factor is the same, as changing Q by the reciprocal factor. Thus case 10 has the same dividing stream tube as if Q were 1500 cfs and u were 1 fps.

Since gravity is nowhere involved in the theory, Figs. 7, 9, 10, and 11 apply as well if the plane $y = a$ is interpreted as the bottom of the river and the plane $y = 0$ is interpreted as the top of the river.

These observations extend considerably the information we can derive from a particular calculation.

The following gives a simple example of how these calculations may be used to estimate biological entrainment. The intake structure of Fig. 7 draws only 16% of its cooling water from below a depth of 40' (i.e., from below $y = 20'$). Thus the fractional depletion per unit time of benthic organisms uniformly distributed in the lowest 20 feet of the river is only $(60/20) \times 16\% = 48\%$ of the fractional depletion per unit time of organisms uniformly distributed throughout the river. In the river of case 9, none of such benthic organisms is entrained.

As a final word, we discuss the application of INTAKE to another situation in which biological entrainment may be important, namely, an oceanic intake consisting of a vertical pipe open at the top in a uniform oceanic current. We can idealize the sink created by the intake pipe as

a point sink. This point sink can be approximated by making b and c small when INTAKE is used. However, a number of changes have to be made. In the first place, in the oceanic case the plane $z = 0$ is not a rigid boundary. This means that we do not need the images at $z = -\epsilon$ mentioned just before Eq. (1). Since these images coalesce in the limit with those at $z = +\epsilon$, the source strength in the oceanic case will be too large by a factor of 2. Hence, in the oceanic case, the value entered for Q should be half of the volumetric withdrawal rate.

The second change that needs to be made concerns the relation of the dimensions of the intake structure and the box in which the isometric projection is displayed. For riverside intakes, the box has a length in the x -direction of five times the intake length ($10c$), with three intake lengths ($6c$) upstream of the intake structure itself. If c is made small to mock up a point sink, this arrangement is no longer useful. In order to make a better arrangement, the programming listed in lines 50, 52, and 163 of INTAKE.F4 may be changed. $X3$ is the x -coordinate of the center of the rectangular sink and is presently set equal to c . $DELX2$ is the interval between tick marks on the x -axis and is presently set at $2c$. $X3$ is reckoned with respect to the origin shown at 0. The plotting routines imagine the origin to be at the point A in Fig. 1. The translation given in line 163 presently places 0 a distance $6c$ to the right of A. ($X1$, $X2$, and Y are the coordinates given to the plotting routine; they are related to the coordinates x , y , and z as shown in Fig. 1.) One may also change the programming given in line 51 if desired; $DELX1$ is the interval between tick marks on the z -axis. With these few changes INTAKE may easily be adapted for use with an oceanic intake.

FOOTNOTES

1. Theoretical Hydrodynamics, L. M. Milne-Thomason, 5th ed., Macmillan, New York, 1968, p. 209 ff.
2. A detailed discussion of the method of images can be found in ref. 1, p. 219 ff. The method is based on the following ideas. Suppose we have a potential flow field in an infinite medium in which there is a surface S on which the component of flow velocity normal to S is zero. The potential flow field on either side of S is then in no way affected if we replace S by a solid boundary. Planes of reflection symmetry are surfaces like S on which the normal component of velocity is zero. Since Laplace's equation for the potential is linear, we can sometimes superimpose infinite-medium solutions to create planes of reflection symmetry where we want solid boundaries. If we succeed in this, and if no source but the real one lies in the region bounded by the planes of symmetry, the sum of the infinite-medium solutions gives the correct solution in the region bounded by the planes of symmetry, the latter being considered as solid boundaries.
3. W. Davis, Jr., A Fortran Program to Draw Isometric-Type Projections with the Calcomp Plotter, USAEC Report ORNL-TM-4137, Oak Ridge National Laboratory, August 1967.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

APPENDIX

SOURCE LISTING OF INTAKE.F4 AND ISOMET

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK


```

C*****
C THIS PROGRAM CALCULATES THE STREAMLINE PASSING THROUGH
C A GIVEN POINT X4,Y4,Z4 IN THE SEMI-INFINITE CHANNEL BOUNDED
C BY THE PLANES Z=0, Y=0, Y=A. A STEADY FLOW EXISTS IN THE
C X-DIRECTION, THE VELOCITY OF WHICH IS U, SUPERIMPOSED ON
C THIS FLOW IS THE FLOW DUE TO A RECTANGULAR SINK OF TOTAL
C STRENGTH Q IN THE PLANE Z=0. THE SIDES OF THE
C SINK ARE PARALLEL TO THE X- AND Y-AXES. THE LENGTH OF THE
C SIDE PARALLEL TO THE X-AXIS IS 2*C, THE LENGTH OF
C THE SIDE PARALLEL TO THE Y-AXIS IS 2*B. THE CENTER
C OF THE SINK IS AT X3,Y3,0. THE TIME STEP
C SEPARATING SUCCESSIVE POINTS ON THE STREAMLINE IS D.
C INPUT PARAMETERS: A,B,C,Q,X3,Y3,U,X4,Y4,Z4,D.
C OUTPUT PARAMETERS: X4,Y4,Z4.
C TEMPORARIES USED: F1-F4,G1-G4,H1-H4.
      DIMENSION AT(8), DATA(1200), IDGE(12), NDRAW(12), ANSRS(12)
      COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,BT(3),XB(4),YB(4),YQ(4)
      DIMENSION NPNT(20), XPLT(50,20), YPLT(50,20), ZPLT(50,20)
      DATA ANSRS/' ','DS','X4','Y4','Z4','NEW','DIS','PLOT','QUIT',
8 'SAV','BACK',0/
      DATA HEIGHT, WIDTH, ANG1, ANG2, RX1X2, TICKL, BMARGIN
8 /6.0, 7.0, 40.0, 20.0, .5, .1, 3.0/
      DATA NX1, DELX1, NX2, DELX2, NINTY1, DELY1, LTND
8 /4,20,0, 5, 0.0, 4, 0.0, 1/
      DATA IDGE /1,1,1,0,1,1,0,1,1,0,1,1/
      DATA NDRAW /1,1,1,0,1,1,0,1,1,0,1,1/
      COMMON /DAT/ A,B,C,Q,X3,Y3,U
      CALL SETROD('INPUT',5)
      TYPE 47
47 FORMAT(' WANT TO RECOVER FOR27,DAT?'/)
      CALL XTRACT
      CALL NEXTASCII(WHERE,NP,5)
      IF (IACOMP(WHERE,'NO',3),EQ, 0) GO TO 49
      READ (27,6) HEIGHT, WIDTH, ANG1, ANG2, RX1X2, TICKL, BMARGIN
      READ (27,7) NX1, DELX1, NX2, DELX2, NINTY1, DELY1, LTND
      READ (27,8) IDGE
      READ (27,8) NDRAW
      READ (27,6) A,B,C,Q,X3,Y3,U,X4,Y4,Z4,D
      READ (27,8) NSTR, NPNT
      DO 17 J=1,NSTR
      NT = NPNT(J)
17 READ (27,6) (XPLT(I,J),I=1,NT), (YPLT(I,J),I=1,NT),
8 (ZPLT(I,J),I=1,NT)
      CALL RELEASE(27)
      GO TO 15
49 TYPE 100
100 FORMAT(' A,B,C,Q,Y3,U=?'/)
      ACCEPT 101, A,B,C,Q,Y3,U
101 FORMAT(6G)

```

```

X3 = C
DELX1 = Q/(3.*A*U)
DELX2 = 2*C
DELY1 = A/NINTY1
NSTR = 1
NPNT(NSTR) = 1
TYPE 3
3  FORMAT (' READY FOR STARTING POINT AND STEP SIZE'/)
4  ACCEPT 5, X4,Y4,Z4,D
5  FORMAT (4G)
430 TYPE 10, X4,Y4,Z4,NSTR,NPNT(NSTR)
10  FORMAT (3G,2I5,/)
    ITIMS = ITIMS + 1
    IF (ITIMS.GT. 0) GO TO 85
15  CALL XTRACT
    CALL NEXTASCII(WHERE,NP,5)
    DO 50 I=1,12
    IF (IACOMP(WHERE,ANSRS(I),2).EQ. 0)
8  GO TO (85,55,50,65,70,75,80,1234,1000,4321,78,51), I
50  CONTINUE
51  ITIMS = NEXTIN(NP)
    IF (ITIMS.GT. 0) GO TO 85
    TYPE 52
52  FORMAT (' DIDN'T UNDERSTAND, PLEASE RETYPE.',/)
    GO TO 15
55  D = FOLLREAL(NP)
    WRITE (5,57) ANSRS(I), D
57  FORMAT ('+',A2,' = ',F8,3,/)
    GO TO 15
60  X4 = FOLLREAL(NP)
    WRITE (5,57) ANSRS(I), X4
    GO TO 15
65  Y4 = FOLLREAL(NP)
    WRITE (5,57) ANSRS(I), Y4
    GO TO 15
70  Z4 = FOLLREAL(NP)
    WRITE (5,57) ANSRS(I), Z4
    GO TO 15
75  NT = NPNT(NSTR)
    XPLT(NT,NSTR) = X4
    YPLT(NT,NSTR) = Y4
    ZPLT(NT,NSTR) = Z4
    NSTR = NSTR + 1
    IF(NSTR.GT.20) NSTR=20
    NPNT(NSTR) = 1
    GO TO 15
78  IF(NSIR.GT.1)NSTR=NSTR-1
80  NPNT(NSTR) = 1
    GO TO 15

```



```

85  NT = NPNT(NSTR)
    XPLT(NT,NSTR) = X4
    YPLT(NT,NSTR) = Y4
    ZPLT(NT,NSTR) = Z4
    NPNT(NSTR) = NPNT(NSTR) + 1
    IF (NPNT(NSTR) .LE. 50) GO TO 87
    NSTR = NSTR + 1
    NPNT(NSTR) = 1
    GO TO 85
87  XT=X4
    YT=Y4
    ZT=Z4
    CALL CALC(XT,YT,ZT,U1,U2,U3)
    F1=(U+U1)*D
    G1=U2*D
    H1=U3*D
    XT=X4+.5*F1
    YT=Y4+.5*G1
    ZT=Z4+.5*H1
    CALL CALC(XT,YT,ZT,U1,U2,U3)
    F2=(U+U1)*D
    G2=U2*D
    H2=U3*D
    XT=X4+.5*F2
    YT=Y4+.5*G2
    ZT=Z4+.5*H2
    CALL CALC(XT,YT,ZT,U1,U2,U3)
    F3=(U+U1)*D
    G3=U2*D
    H3=U3*D
    XT=X4+F3
    YT=Y4+G3
    ZT=Z4+H3
    CALL CALC(XT,YT,ZT,U1,U2,U3)
    F4=(U+U1)*D
    G4=U2*D
    H4=U3*D
    X4=X4+(F1+2*F2+2*F3+F4)/6
    Y4=Y4+(G1+2*G2+2*G3+G4)/6
    Z4=Z4+(H1+2*H2+2*H3+H4)/6
    GO TO 430
1234 CALL PLOTS(DATA,1200,LTNO)
     NT = NPNT(NSTR)
     XPLT(NT,NSTR) = X4
     YPLT(NT,NSTR) = Y4
     ZPLT(NT,NSTR) = Z4

```

```

CALL BOX(HEIGHT,WIDTH,ANG1,ANG2,RX1X2,BMARGIN,NX1,DELY1,NX2,
1 DELX2,NINTY1,DELY1,IDGE,LTNO,AT)
CALL MARK(NDRAW,NX1,DELY1,NX2,DELY1,TICKL)
X2T = 6*C
YTT = Y3 + B
CALL CURVE(0,0,X2T,YTT,1,1,2)
YTT = Y3 - B
CALL CURVE(0,0,X2T,YTT,1,2,2)
X2T = 8*C
CALL CURVE(0,0,X2T,YTT,1,3,2)
YTT = Y3 + B
CALL CURVE(0,0,X2T,YTT,1,3,2)
X2T=6*C
CALL CURVE(0,0,X2T,YTT,1,3,2)
DO 800 J=1,NSIR
NT = NPNT(J)
DO 800 I=1,NT
X1 = ZPLT(I,J)
X2 = XPLT(I,J) + 6,*C
Y = YPLT(I,J)
CALL CURVE(X1,X2,Y,1,I,2)
CALL CURVE(X1,X2,0,,1,I,2)
800 CALL CURVE(X1,X2,Y-1,I,2)
CALL ADVANCE(AT)
GO TO 15
1000 CALL EXIT
4321 CALL OFILE(27,'FOR27')
WRITE(27,6) HEIGHT, WIDTH, ANG1, ANG2, RX1X2, TICKL, BMARGIN
6 FORMAT(1P7E11,3)
WRITE(27,7) NX1, DELX1, NX2, DELX2, NINTY1, DELY1, LTNO
7 FORMAT(3(1P,1PE11,3))
WRITE(27,8) IDGE
8 FORMAT(12I5)
WRITE(27,8) NDRAW
WRITE(27,6) A,B,C,Q,X3,Y3,U,X4,Y4,Z4,D
WRITE(27,8) NSTR, NPNT
DO 14 J=1,NSTR
NT = NPNT(J)
14 WRITE(27,6) (XPLT(I,J),I=1,NT), (YPLT(I,J),I=1,NT);
8 (ZPLT(I,J),I=1,NT)
GO TO 15
END

```

```

C!*****
SUBROUTINE CALC(XT,YT,ZT,U1,U2,U3)
C!THIS SUBROUTINE CALCULATES THE COMPONENTS U1,U2,U3
C!OF FLOW VELOCITY AT X,Y,Z DUE TO A RECTANGULAR SINK IN THE
C!(X,Y)-PLANE OF TOTAL STRENGTH Q WHOSE SIDES ARE PARALLEL TO
C!THE X- AND Y-AXES. THE LENGTH OF THE SIDE PARALLEL TO THE
C!X-AXIS IS 2*C AND THE LENGTH OF THE SIDE PARALLEL
C!TO THE Y-AXIS IS 2*B, THE CENTER IS LOCATED AT
C!X3,Y3,0, Z=0, Y=0, AND Y=A ARE RIGID SURFACES.
C!INPUT PARAMETERS: X3,Y3,B,C, (X,Y,Z,A,Q).
C!OUTPUT PARAMETERS: U1,U2,U3
C!TEMPORARIES USED: N,L,M,X2,Y2,Z2,D1,D2,D3,T6.
COMMON /DAT/ A,B,C,Q,X3,Y3,U
U1=0
U2=0
U3=0
DO 1080 N=0,10200
D1=0
D2=0
D3=0
DO 990 L=-1,1,2
DO 990 M=-1,1,2
X2=XT-X3
Y2=YT-L*Y3+2*M*N*A
Z2=ZT
D1=D1+X(X2,Y2,Z2,B,C)
D2=D2+Y(X2,Y2,Z2,B,C)
D3=D3+Z(X2,Y2,Z2,B,C)
990 CONTINUE
U1=U1+.5*(1+SGN(N))*D1
U2=U2+.5*(1+SGN(N))*D2
U3=U3+.5*(1+SGN(N))*D3
T6=1,E=3
IF (ABS(D1) .LT. T6 .AND. ABS(D2) .LT. T6 .AND.
& ABS(D3) .LT. T6) GO TO 1100
1080 CONTINUE
TYPE 1090
1090 FORMAT (' ALL IMAGES USED,')
1100 U1=L1*Q/(25.1328*B*C)
U2=L2*Q/(25.1328*B*C)
U3=L3*Q/(25.1328*B*C)
RETURN
END

```

```

FUNCTION F(X1,Y1,Z1,B,C)
R1=SQRT((X1-C)**2+(Y1-B)**2+Z1**2)
R2=ABS(Y1-B)
IF (R2 ,EQ, 0.) GO TO 100
IF ((R1-R2)/R2,GE,1.E-3) GO TO 100
R4=((X1-C)**2+Z1**2)/(2*R2)
F=SGN(Y1-B)*ALOG((R1+R2)/R4)/2.
RETURN
100 F=SGN(Y1-B)*ALOG((R1+R2)/(R1-R2))/2
END

```

```

FUNCTION X(X1,Y1,Z1,B,C)
X=F(X1,Y1,Z1,B,C)+F(-X1,Y1,Z1,B,C)+F(X1,-Y1,Z1,B,C)
X=X-F(-X1,-Y1,Z1,B,C)
END

```

```

FUNCTION Y(X1,Y1,Z1,B,C)
Y=X(Y1,X1,Z1,C,B)
END

```

```

FUNCTION G(X1,Y1,Z1,B,C)
R3=SQRT((X1-C)**2+(Y1+B)**2+Z1**2)
G=SGN(Y1+B)*ATAN(ABS((Y1+B)/Z1)*(X1-C)/R3)
END

```

```

FUNCTION Z(X1,Y1,Z1,B,C)
Z=G(X1,Y1,Z1,B,C)+G(-X1,Y1,Z1,B,C)+G(X1,-Y1,Z1,B,C)
Z=Z+G(-X1,-Y1,Z1,B,C)
END

```

```

FUNCTION SGN(X)
IF (X) 10,20,30
10 SGN = -1.0
RETURN
20 SGN = 0.0
RETURN
30 SGN = 1.0
RETURN
END

```

IIIIIIII	SSSSSSSSSS	00000000	MMM	MMM	EEEEEEEEEEEE	TTTTTTTTTTTT
IIIIIIII	SSSSSSSSSS	00000000	MMM	MMM	EEEEEEEEEEEE	TTTTTTTTTTTT
IIIIIIII	SSSSSSSSSS	00000000	MMM	MMM	EEEEEEEEEEEE	TTTTTTTTTTTT
III	SSS	000	MMMMMM	MMMMMM	EEE	TTT
III	SSS	000	MMMMMM	MMMMMM	EEE	TTT
III	SSS	000	MMMMMM	MMMMMM	EEE	TTT
III	SSS	000	MMM	MMM	EEE	TTT
III	SSS	000	MMM	MMM	EEE	TTT
III	SSS	000	MMM	MMM	EEE	TTT
III	SSSSSSSS	000	MMM	MMM	EEEEEEEEEEEE	TTT
III	SSSSSSSS	000	MMM	MMM	EEEEEEEEEEEE	TTT
III	SSSSSSSS	000	MMM	MMM	EEEEEEEEEEEE	TTT
III	SSS	000	MMM	MMM	EEE	TTT
III	SSS	000	MMM	MMM	EEE	TTT
III	SSS	000	MMM	MMM	EEE	TTT
III	SSS	000	MMM	MMM	EEE	TTT
III	SSS	000	MMM	MMM	EEE	TTT
III	SSS	000	MMM	MMM	EEE	TTT
III	SSS	000	MMM	MMM	EEE	TTT
III	SSS	000	MMM	MMM	EEE	TTT
IIIIIIII	SSSSSSSSSS	00000000	MMM	MMM	EEEEEEEEEEEE	TTT
IIIIIIII	SSSSSSSSSS	00000000	MMM	MMM	EEEEEEEEEEEE	TTT
IIIIIIII	SSSSSSSSSS	00000000	MMM	MMM	EEEEEEEEEEEE	TTT

REQUEST CREATED: 29-APR-73 09146112
 FILE: OSKB01180MET[6534,17] CREATED: 14-FEB-73 13119100 <155> PRINTED: 29-APR-73 29146151
 QUEUE SWITCHES: /PRINT:ARROW /FILE:ASCII /COPIES:1 /SPACING:1 /LIMIT:53
 FILE WILL BE RENAMED TO <055> PROTECTION

```

SUBROUTINE BOX(HT,WIDTH,ANG1,ANG2,RX1X2,BMARGN,NX1,DELY1,
1.  NX2,DELY2,NINTY1,DELY1,IDGE,LTNO,A)
DIMENSION A(6),DATA(1200),IDGE(12)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
IF (IOP, EQ, 11897) GO TO 4
IOP = 11897
CALL PLOTS(DATA,1200,LTNO)
4 IF(WIDTH)151,5,5
5 SIN1 = SIN(ANG1*6.2831853/360,0)
SIN2 = SIN(ANG2*6.2831853/360,0)
COS1 = COS(ANG1*6.2831853/360,0)
COS2 = COS(ANG2*6.2831853/360,0)
D2 = WIDTH/(RX1X2+COS1+COS2)
D1 = RX1X2*D2
XB(1) = 0,0
XB(2) = D1*COS1
XB(4) = D2*COS2
XB(3) = XB(2) + XB(4)
YB(1) = D1*SIN1
YB(2) = 0,0
YB(3) = D2*SIN2
YB(4) = YB(1) + YB(3)
D3 = HT - YB(4)
B(1) = D1/(DELY2*NX1)
B(2) = D2/(DELY2*NX2)
B(3) = D3/(DELY1*NINTY1)
DO 11 I=1,4
11 YB(I)=YB(I)+BMARGN
DO 21 I=1,4
21 YT(I) = YB(I) + D3
IF(IDGE(1)=1)31,25,31
25 CALL PLOT(XB(1),YB(1),3)
CALL PLOT(XB(2),YB(2),2)
31 IF(IDGE(2)=1)41,35,41
35 CALL PLOT(XB(3),YB(3),3)
CALL PLOT(XB(4),YB(4),2)
41 IF(IDGE(3)=1)51,45,51
45 CALL PLOT(XB(3),YT(3),3)
CALL PLOT(XB(4),YT(4),2)
51 IF(IDGE(4)=1)61,55,61
55 CALL PLOT(XB(1),YT(1),3)
CALL PLOT(XB(2),YT(2),2)
61 IF(IDGE(5)=1)71,65,71
65 CALL PLOT(XB(1),YB(1),3)
CALL PLOT(XB(4),YB(4),2)
71 IF(IDGE(6)=1)81,75,81
75 CALL PLOT(XB(2),YB(2),3)
CALL PLOT(XB(3),YB(3),2)
81 IF(IDGE(7)=1)91,85,91
85 CALL PLOT(XB(2),YT(2),3)
CALL PLOT(XB(3),YT(3),2)
91 IF(IDGE(8)=1)101,95,101

```

```

95 CALL PLOT(XB(1),YT(1),3)
   CALL PLOT(XB(4),YT(4),2)
101 IF(IDGE(9)=1)111,105,111
105 CALL PLOT(XB(1),YB(1),3)
   CALL PLOT(XB(1),YT(1),2)
111 IF(IDGE(10)=1)121,115,121
115 CALL PLOT(XB(2),YB(2),3)
   CALL PLOT(XB(2),YT(2),2)
121 IF(IDGE(11)=1)131,125,131
125 CALL PLOT(XB(3),YB(3),3)
   CALL PLOT(XB(3),YT(3),2)
131 IF(IDGE(12)=1)141,135,141
135 CALL PLOT(XB(4),YB(4),3)
   CALL PLOT(XB(4),YT(4),2)
141 W = ABS(WIDTH)
   A(5) = 0,0
   A(6) = W+1,0
151 RETURN
   END

```

```

SUBROUTINE MARK(NDRAW,NX1,DELX1,NX2,DELX2,NINTY1,DELY1,TICKL)
DIMENSION NDRAW(12)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
IF(NDRAW(1)-1) 31, 11, 21
11 CALL TICX11(NX1,DELX1,TICKL)
GO TO 31
21 CALL GRDX11(NX1,DELX1)
31 IF(NDRAW(2)-1) 61, 41, 51
41 CALL TICX12(NX1,DELX1,TICKL)
GO TO 61
51 CALL GRDX12(NX1,DELX1)
61 IF(NDRAW(3)-1) 91, 71, 81
71 CALL TICX13(NX1,DELX1,TICKL)
GO TO 91
81 CALL GRDX13(NX1,DELX1)
91 IF(NDRAW(4)-1) 121, 101, 111
101 CALL TICX14(NX1,DELX1,TICKL)
GO TO 121
111 CALL GRDX14(NX1,DELX1)
121 IF(NDRAW(5)-1) 151, 131, 141
131 CALL TICX21(NX2,DELX2,TICKL)
GO TO 151
141 CALL GRDX21(NX2,DELX2)
151 IF(NDRAW(6)-1) 181, 161, 171
161 CALL TICX22(NX2,DELX2,TICKL)
GO TO 181
171 CALL GRDX22(NX2,DELX2)
181 IF(NDRAW(7)-1) 211, 191, 201
191 CALL TICX23(NX2,DELX2,TICKL)
GO TO 211
201 CALL GRDX23(NX2,DELX2)
211 IF(NDRAW(8)-1) 241, 221, 231
221 CALL TICX24(NX2,DELX2,TICKL)
GO TO 241
231 CALL GRDX24(NX2,DELX2)
241 IF(NDRAW(9)-1) 271, 251, 261
251 CALL TICKY1(NINTY1,DELY1,TICKL)
GO TO 271
261 CALL GRIDY1(NINTY1,DELY1)
271 IF(NDRAW(10)-1) 301, 281, 291
281 CALL TICKY2(NINTY1,DELY1,TICKL)
GO TO 301
291 CALL GRIDY2(NINTY1,DELY1)
301 IF(NDRAW(11)-1) 331, 311, 321
311 CALL TICKY3(NINTY1,DELY1,TICKL)
GO TO 331
321 CALL GRIDY3(NINTY1,DELY1)
331 IF(NDRAW(12)-1) 361, 341, 351
341 CALL TICKY4(NINTY1,DELY1,TICKL)
GO TO 361
351 CALL GRIDY4(NINTY1,DELY1)
361 RETURN
END

```



```

SUBROUTINE POINT(X1,X2,Y,NORIGN,ITYPE,SYMBHT,THETA,NPOINT,IOP,
1          NPERP)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
IF(IOP = 1) 11, 11, 21
11 N = -1
C PLOT POINTS BUT DO NOT CONNECT THEM
GO TO 42
21 IF(NPOINT = 1) 31, 11, 31
31 IF(IOP = 1) 41, 11, 41
41 N = -2
C PLOT POINTS AND CONNECT THESE WITH A STRAIGHT LINE.
45 IF(NORIGN = 2) 51, 41, 71
51 YPB = YB(1) + X1*B(1)*SIN1 + X2*B(2)*SIN2
YP = YPB + Y*B(3)
XP = XB(1) + X1*B(1)*COS1 + X2*B(2)*COS2
GO TO 101
61 YPB = YB(2) + X1*B(1)*SIN1 + X2*B(2)*SIN2
YP = YPB + Y*B(3)
XP = XB(2) + X1*B(1)*COS1 + X2*B(2)*COS2
GO TO 101
71 IF(NORIGN = 3) 81, 81, 91
81 YPB = YB(3) + X1*B(1)*SIN1 - X2*B(2)*SIN2
YP = YPB + Y*B(3)
XP = XB(3) + X1*B(1)*COS1 - X2*B(2)*COS2
GO TO 101
91 YPB = YB(4) + X1*B(1)*SIN1 - X2*B(2)*SIN2
YP = YPB + Y*B(3)
XP = XB(4) + X1*B(1)*COS1 - X2*B(2)*COS2
101 CALL SYMBOL(XP,YP,SYMBHT,ITYPE,THETA,N)
IF(NPERP)111,121,111
111 CALL PLOT(XP,YP,2)
C DRAW A LINE PERPENDICULAR TO THE BASE PLANE THROUGH THE POINT IF
C NPERP IS A NON-ZERO INTEGER
CALL PLOT(XP,YP,1)
121 CALL PLOT(XP,YP,3)
RETURN
END

```

```

SUBROUTINE CURVE(X1,X2,Y,NORIGIN,NPOINT,K)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
IF(K.EQ.3) GO TO 102
IF(NORIGIN=2)51,61,71
51 YP = YB(1) + X1*B(1)*SIN1 + X2*B(2)*SIN2 + Y*B(3)
XP = XB(1) + X1*B(1)*COS1 + X2*B(2)*COS2
GO TO 101
61 YP = YB(2) + X1*B(1)*SIN1 + X2*B(2)*SIN2 + Y*B(3)
XP = XB(2) + X1*B(1)*COS1 + X2*B(2)*COS2
GO TO 101
71 IF(NORIGIN=3)81,81,91
81 YP = YB(3) + X1*B(1)*SIN1 - X2*B(2)*SIN2 + Y*B(3)
XP = XB(3) + X1*B(1)*COS1 - X2*B(2)*COS2
GO TO 101
91 YP = YB(4) + X1*B(1)*SIN1 - X2*B(2)*SIN2 + Y*B(3)
XP = XB(4) + X1*B(1)*COS1 - X2*B(2)*COS2
101 J=K+1
GO TO (141,132,104)J
141 ZP=YP
XF=XP
132 RETURN
104 ZP=YP
XF=XP
102 IF(NPOINT=2)111,121,131
111 CALL PLOT(XF,ZP,3)
RETURN
121 CALL PLOT(XF,ZP,2)
RETURN
131 CALL PLOT(XF,ZP,1)
RETURN
END
SUBROUTINE TICX11(NX1,DETX1,TICKL)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NX1T = NX1 - 1
DO 21 J = 1,NX1T
XPLT = XB(1) + J*DETX1*COS1*B(1)
YPLT = YB(1) + J*DETX1*SIN1*B(1) + TICKL
CALL PLOT(XPLT,YPLT,3)
YPLT = YPLT - TICKL
CALL PLOT(XPLT,YPLT,2)
XPLT = XPLT + TICKL*COS2
YPLT = YPLT + TICKL*SIN2
21 CALL PLOT(XPLT,YPLT,1)
RETURN
END

```

```

SUBROUTINE TICX12(NX1,DELX1,TICKL)
COMMON/L8COM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),VT(4)
NX1T = NX1 - 1
DO 21 J = 1,NX1T
  XPLT = XB(4) + J*DELX1*COS1*B(1) - TICKL*COS2
  YPLT = YB(4) - J*DELX1*SIN1*B(1) - TICKL*SIN2
  CALL PLOT(XPLT,YPLT,3)
  XPLT = XPLT + TICKL*COS2
  YPLT = YPLT + TICKL*SIN2
  CALL PLOT(XPLT,YPLT,2)
  YPLT = YPLT + TICKL
21 CALL PLOT(XPLT,YPLT,1)
RETURN
END

```

```

SUBROUTINE TICK13(NX1,DELX1,TICKL)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NX1T = NX1 - 1
DO 21 J = 1,NX1T
  XPLT = XB(4) + J*DELX1*COS1*B(1)
  YPLT = YT(4) - J*DELX1*SIN1*B(1) - TICKL
  CALL PLOT(XPLT,YPLT,3)
  YPLT = YPLT + TICKL
  CALL PLOT(XPLT,YPLT,2)
  XPLT = XPLT + TICKL*COS2
  YPLT = YPLT - TICKL*SIN2
21 CALL PLOT(XPLT,YPLT,1)
RETURN
END

```

```

SUBROUTINE TICX14(NX1,DELX1,TICKL)
COMMON/LB COM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NX1T = NX1 - 1
DO 21 J = 1,NX1T
  XPLT = XB(1) + J*DELX1*COS1*B(1) + TICKL*COS2
  YPLT = YT(1) + J*DELX1*SIN1*B(1) + TICKL*SIN2
  CALL PLOT(XPLT,YPLT,3)
  XPLT = XPLT + TICKL*COS2
  YPLT = YPLT + TICKL*SIN2
  CALL PLOT(XPLT,YPLT,2)
  YPLT = YPLT - TICKL
21 CALL PLOT(XPLT,YPLT,1)
RETURN
END

```

```

SUBROUTINE TICX21(NX2,DELX2,TICKL)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NX2T = NX2 = 1
DO 21 J = 1,NX2T
  XPLT = XB(1) + J*DELX2*COS2*E(2)
  YPLT = YB(1) + J*DELX2*SIN2*E(2) + TICKL
  CALL PLOT(XPLT,YPLT,3)
  YPLT = YFLT = TICKL
  CALL PLOT(XPLT,YPLT,2)
  XPLT = XFLT = TICKL*COS1
  YPLT = YFLT = TICKL*SIN1
21 CALL PLOT(XPLT,YPLT,1)
RETURN
END

```

```

SUBROUTINE TICK22(NX2,DELX2,TICKL)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NX2T = NX2 = 1
DO 21 J = 1,NX2T
  XPLT = XB(2) + J*DELX2*COS2*B(2) = TICKL*COS1
  YPLT = YB(2) + J*DELX2*SIN2*B(2) + TICKL*SIN1
  CALL PLOT(XPLT,YPLT,3)
  XPLT = XPLT + TICKL*COS1
  YPLT = YPLT + TICKL*SIN1
  CALL PLOT(XPLT,YPLT,2)
  YPLT = YPLT + TICKL
21 CALL PLOT(XPLT,YPLT,1)
RETURN
END

```

```

SUBROUTINE TICX23(NX2,DELX2,TICKL)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NX2T = NX2 = 1
DO 21 J = 1,NX2T
  XPLT = XB(2) + J*DELX2*COS2*B(2)
  YPLT = YT(2) + J*DELX2*SIN2*B(2) = TICKL
  CALL PLOT(XPLT,YPLT,3)
  YPLT = YPLT + TICKL
  CALL PLOT(XPLT,YPLT,2)
  XPLT = XPLT + TICKL*COS1
  YPLT = YPLT + TICKL*SIN1
21 CALL PLOT(XPLT,YPLT,1)
RETURN
END

```



```

SUBROUTINE TICX24(NX2,DELX2,TICKL)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NX2T = NX2 = 1
DO 21 J = 1,NX2I
  XPLT = XB(1) + J*DELX2*COS2*B(2) + TICKL*COS1
  YPLT = YT(1) + J*DELX2*SIN2*B(2) - TICKL*SIN1
  CALL PLOT(XPLT,YPLT,3)
  XPLT = XPLT + TICKL*COS1
  YPLT = YPLT + TICKL*SIN1
  CALL PLOT(XPLT,YPLT,2,
  YPLT = YPLT - TICKL
21 CALL PLOT(XPLT,YPLT,1)
RETURN
END

```

```

SUBROUTINE TICKY1(NINTY1,DELY1,TICKL)
COMMON/LBCOM1/S,N1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NINT = NINTY1 - 1
DO 21 J = 1,NINT
  XPLT = XB(1) + TICKL*COS2
  YPLT = YB(1) + J*DELY1*B(3) + TICKL*SIN2
  CALL PLOT(XPLT,YPLT,3)
  XPLT = XPLT + TICKL*COS2
  YPLT = YPLT + TICKL*SIN2
  CALL PLOT(XPLT,YPLT,2)
  XPLT = XPLT + TICKL*COS1
  YPLT = YPLT + TICKL*SIN1
21 CALL PLOT(XPLT,YPLT,1)
RETURN
END

```

```

SUBROUTINE TICKY2(NINTY1,DELY1,TICKL)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NINT = NINTY1 - 1
DO 21 J = 1,NINT
  XPLT = XB(2) - TICKL*COS1
  YPLT = YB(2) - J*DELY1*B(3) + TICKL*SIN1
  CALL PLOT(XPLT,YPLT,3)
  XPLT = XPLT + TICKL*COS1
  YPLT = YPLT - TICKL*SIN1
  CALL PLOT(XPLT,YPLT,2)
  XPLT = XPLT + TICKL*COS2
  YPLT = YPLT + TICKL*SIN2
21 CALL PLOT(XPLT,YPLT,1)
RETURN
END

```

```

SUBROUTINE TICKY3(NINTY1,DELY1,TICKL)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NINT = NINTY1 - 1
DO 21 J = 1,NINT
  XPLT = XB(3) - TICKL*COS2
  YPLT = YB(3) + J*DELY1*B(3) - TICKL*SIN2
  CALL PLOT(XPLT,YPLT,3)
  XPLT = XPLT + TICKL*COS2
  YPLT = YPLT + TICKL*SIN2
  CALL PLOT(XPLT,YPLT,2)
  XPLT = XPLT - TICKL*COS1
  YPLT = YPLT + TICKL*SIN1
21 CALL PLOT(XPLT,YPLT,1)
RETURN
END

```

```

SUBROUTINE TICKY4(NINTY1,DELY1,TICKL)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NINT = NINTY1 = 1
DO 21 J = 1,NINT
  XPLT = XB(4) + TICKL*COS1
  YPLT = YB(4) - J*DELY1*B(3) - TICKL*SIN1
  CALL PLOT(XPLT,YPLT,3)
  XPLT = XPLT + TICKL*COS1
  YPLT = YPLT + TICKL*SIN1
  CALL PLOT(XPLT,YPLT,2)
  XPLT = XPLT + TICKL*COS2
  YPLT = YPLT + TICKL*SIN2
21 CALL PLOT(XPLT,YPLT,1)
RETURN
END

```

```

SUBROUTINE GRDX11(NX1,DELX1)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NX1T = NX1 = 1
DO 21 J = 1,NX1T
  XPLT = XB(1) + J*DELX1*COS1*B(1)
  YPLT = YT(1) - J*DELX1*SIN1*B(1)
  CALL PLOT(XPLT,YPLT,3)
  YPLT = YB(1) - J*DELX1*SIN1*B(1)
21 CALL PLOT(XPLT,YPLT,2)
RETURN
END

```

```

SUBROUTINE GRDX12(NX1,DELX1)
COMMON/BCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NX1T = NX1 = 1
DO 21 J = 1,NX1T
  XPLT = XB(1) + J*DELX1*COS1*B(1)
  YPLT = YB(1) + J*DELX1*SIN1*B(1)
  CALL PLOT(XPLT,YPLT,3)
  XPLT = XB(4) + J*DELX1*COS1*B(1)
  YPLT = YB(4) + J*DELX1*SIN1*B(1)
21 CALL PLOT(XPLT,YPLT,2)
RETURN
END

```

```

SUBROUTINE GRDX13(NX1,DELX1)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NX1T = NX1 - 1
DO 21 J = 1,NX1T
  XPLT = XB(4) + J*DELX1*COS1*B(1)
  YPLT = YB(4) - J*DELX1*SIN1*B(1)
  CALL PLOT(XPLT,YPLT,3)
  YPLT = YT(4) - J*DELX1*SIN1*B(1)
21 CALL PLOT(XPLT,YPLT,2)
RETURN
END

```



```

SUBROUTINE GRDX14(NX1,DELX1)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NX1T = NX1 - 1
DO 21 J = 1,NX1T
  XPLT = XB(4) + J*DELX1*COS1*B(1)
  YPLT = YT(4) - J*DELX1*SIN1*B(1)
  CALL PLOT(XPLT,YPLT,3)
  XPLT = XB(1) + J*DELX1*COS1*B(1)
  YPLT = YT(1) - J*DELX1*SIN1*B(1)
21 CALL PLOT(XPLT,YPLT,2)
RETURN
END

```

```

SUBROUTINE GRDX21(NX2,DELX2)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NX2T = NX2 - 1
DO 21 J = 1,NX2T
  XPLT = XB(1) + J*DELX2*COS2*B(2)
  YPLT = YT(1) + J*DELX2*SIN2*B(2)
  CALL PLOT(XPLT,YPLT,3)
  YPLT = YB(1) + J*DELX2*SIN2*B(2)
21 CALL PLOT(XPLT,YPLT,2)
RETURN
END

```

```

SUBROUTINE GRDX22(NX2,DELX2,
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NX2I = NX2 - 1
DO 21 J = 1,NX2I
  XPLT = XB(1) + J*DELX2*COS2*B(2)
  YPLT = YB(1) + J*DELX2*SIN2*B(2)
  CALL PLOT(XPLT,YPLT,3)
  XPLT = XB(2) + J*DELX2*COS2*B(2)
  YPLT = YB(2) + J*DELX2*SIN2*B(2)
21 CALL PLOT(XPLT,YPLT,2)
RETURN
END

```

```

SUBROUTINE GRD423(NX2,DELX2)
COMMON/LBGM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NX2T = NX2 - 1
DO 21 J = 1,NX2T
  XPLT = XB(2) + J*DELX2*COS2*B(2)
  YPLT = YB(2) + J*DELX2*SIN2*B(2)
  CALL PLOT(XPLT,YPLT,3)
  YPLT = YT(2) + J*DELX2*SIN2*B(2)
21 CALL PLOT(XPLT,YPLT,2)
RETURN
END

```

```

SUBROUTINE GRDX24(NX2,DELX2)
COMMON/LB3COM1/SIN1,SIN2,COS1,COS2,D1,D2,C3,B(3),XB(4),YB(4),YT(4)
NX2T = NX2 = 1
DO 21 J = 1,NX2T
  YPLT = YT(2) + J*DELX2*SIN2*B(2)
  XPLT = XB(2) + J*DELX2*COS2*B(2)
  CALL PLOT(XPLT,YPLT,3)
  XPLT = XB(1) + J*DELX2*COS2*B(2)
  YPLT = YT(1) + J*DELX2*SIN2*B(2)
21 CALL PLOT(XPLT,YPLT,2)
RETURN
END

```

```

SUBROUTINE GRIDY1(NINTY1,DELY1)
COMMON/LB30M1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NINT = NINTY1 - 1
DO 21 J = 1,NINT
    YPLT = YB(4) + J*DELY1*B(3)
    CALL PLOT(XB(4),YPLT,3)
    YPLT = YB(1) + J*DELY1*B(3)
21 CALL PLOT(XB(1),YPLT,2)
RETURN
END

```

```

SUBROUTINE GRIDY2(NINTY1,DELY1)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NINT = NINTY1 = 1
DO 21 J = 1,NINT
    YPLT = YB(1) + J*DELY1*B(3)
    CALL PLCT(XB(1),YPLT,3)
    YPLT = YB(2) + J*DELY1*B(3)
21 CALL PLCT(XB(2),YPLT,2)
RETURN
END

```

```

SUBROUTINE GRIDY3(NINTY1,DELY1,
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,B(3),XB(4),YB(4),YT(4)
NINT = NINTY1 = 1
DO 21 J = 1,NINT
    YPLT = YB(2) + J*DELY1*B(3)
    CALL PLOT(XB(2),YPLT,3)
    YPLT = YB(3) + J*DELY1*B(3)
21 CALL PLOT(XB(3),YPLT,2)
RETURN
END

```



```

SUBROUTINE GRIDY4(NINTY1,DELY1)
COMMON/LBCOM1/SIN1,SIN2,COS1,COS2,D1,D2,D3,E(3),XB(4),YB(4),YT(4)
NINT = NINTY1 = 1
DO 21 J = 1,NINT
  YPLT = YB(3) + J*DELY1*B(3)
  CALL PLOT(XB(3),YPLT,3)
  YPLT = YB(4) + J*DELY1*B(3)
21 CALL PLOT(XB(4),YPLT,2)
RETURN
END

```

```
SUBROUTINE ADVANC(A)
  DIMENSION A(6)
  X = A(6) + 2,
3 CALL PLOT(X,0,,-3)
  RETURN
END
```

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

INTERNAL DISTRIBUTION

1. S. E. Beall	48. D. A. Pilati
2. F. T. Binford	49. A. S. Quist
3. W. E. Browning	50. R. R. Rickard
4. R. H. Bryan	51. R. C. Robertson
5. R. H. Busey	52. J. W. Roddy
6-10. V. R. Cain	53. M. W. Rosenthal
11. W. B. Cottrell	54-103. T. H. Row
12. C. C. Coutant	104. R. M. Rush
13. W. C. Cox	105. T. Shapiro
14. F. L. Culler	106. M. R. Sheldon
15-19. W. Davis, Jr.	107. S. Siegel
20-24. L. Dresner	108. M. Siman-Tov
25. C. Easterly	109. O. Sisman
26. A. Eraslan	110. M. J. Skinner
27. A. P. Fraas	111. W. T. Snyder
28. P. M. Garrett	112. I. Spiewak
29. C. P. Goodyear	113. W. G. Stockdale
30. J. A. Hafford	114. E. G. Struxness
31. R. F. Hibbs	115. D. A. Sundberg
32. R. M. Hill	116. E. H. Taylor
33. H. W. Hoffman	117. R. E. Thoma
34. P. R. Kasten	118. D. B. Trauger
35. G. W. Keilholtz	119. W. C. Tunnell
36. S. S. Kirsulis	120. F. Vaslow
37. O. H. Klepper	121. A. M. Weinberg
38. R. N. Lyon	122. G. D. Whitman
39. R. E. MacPherson	123-130. R. P. Wichner
40. H. C. McCurdy	131. M. M. Yarosh
41. J. R. McWherter	132. W. C. Yee
42. A. J. Miller	133-135. Central Research Library
43. M. Mitchell	136. Document Reference Section
44. D. J. Nelson	137-139. Laboratory Records Department
45. B. L. Nichols	140. Laboratory Records Department - RC
46. A. M. Perry	141. EISO Library (Bldg. 2028)
47. E. Pickelsimer	

EXTERNAL DISTRIBUTION

- 142-159. Manager, Technical Information Center, AEC (ACRS Members)
- 160. Research and Technical Support Division, AEC, ORO
- 161. Patent Office, AEC, ORO
- 162-163. Technical Information Center, AEC