AD 634404

THE BOEING COMPANY
AIRPLANE DIVISION - WICHITA BRANCH

CODE IDENT. NO. 81205

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DOCUMENT NUMBER D3-6961 DATE 29 December 1965

TITLE WATER DROPLET IMPINGEMENT PREDICTION FOR ENGINE INLETS BY TRAJECTORY ANALYSIS IN A POTENTIAL FLOW FIELD - FINAL REPORT

CONTRACT NO. NOw 65-0273-f Item 1 MODEL R&D

DISTRIBUTION LIMITATIONS - DDC AVAILABILITY NOTICE

- [X] ~~Qualified requesters may obtain copies of this report from DDC.~~
- [ ] Foreign announcement and dissemination of this report by DDC are not authorized.
- [ ] U. S. government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through The Boeing Company.
- [ ] U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through The Boeing Company.
- [ ] All distribution of this report is controlled. Qualified DDC users shall request through The Boeing Company.

DDC
JUN 30 1966
A

PREPARED BY William F Schmidt 12-29-65

-29-65

/29

/29

CLEARINGHOUSE
FOR FEDERAL SCIENTIFIC AND TECHNICAL INFORMATION

Hardcopy	Microfiche		
$6.00	$1.25	227 pp	as

ARCHIVE COPY

REV LTR: _

ISSUE NO. _ /

E-3039 R2

Best
Available
Copy

TABLE OF CONTENTS

		Page
1.0	Object	4
2.0	Summary	5
2.1	Development Summary	5
2.2	Descriptive Summary	5
3.0	Conclusions	6
4.0	Recommendations	7
5.0	Description	8
6.0	Method	9
6.1.1	Potential Flow Field Analysis	9
6.1.2	Potential Flow Computer Program	13
6.1.3	Potential Flow Computer Program Usage	17
6.2.1	Water Droplet Trahectory Analysis	22
6.2.2	Water Droplet Trajectory Computer Program	25
6.2.3	Water Droplet Trajectory Computer Program Usage	28
7.0	Results and Discussion	38
8.0	Symbols and References	40
8.1	Symbols	40
8.2	References	42
9.0	Plates	44
10.0	Appendices	56
10.1	Appendix 1 - Potential Flow Computer Program Flow Chart	56
10.2	Appendix 2 - Potential Flow Computer Program Listing	112

		Page
10.3	Appendix 3 - Water Droplet Trajectory Computer Program Flow Chart	153
10.4	Appendix 4 - Water Droplet Trajectory Computer Program Listing	206

1.0 OBJECT

This report is written to fulfill the requirements of Bureau of Naval Weapons contract NOw 65-0273-f Item 4. This contract was issued for development of a digital computer program to predict the anti-icing requirements for engine inlets by predicting water droplet impingement. The determination of the design requirements is made by establishing the potential flow field around the subject surface and then calculating the trajectory of the water droplet as it approaches and impinges on the surface.

During the execution of this contract, any new discoveries or inventions (subject inventions) were to be reported to the Government.

2.0 SUMMARY

2.1 Development Summary

With the signing of the development contract on March 9, 1965, the Contractor started work on the two basic digital computer programs to provide an accurate method of water droplet impingement. The potential flow computer program was updated and refined by the addition of variable distance incrementation, improved plotting and table preparation routines and more output.

The water droplet trajectory computer program was adapted to the improved potential flow program and expanded to include methods to calculate trajectories that included smaller droplets, droplet reversal, gravity and buoyancy effects and random input. Provisions for summing the results from a given droplet size distribution and for determining the tangent droplet trajectories were also included. Both computer programs were verified by comparison with theoretical and NACA analytical and test results.

In compliance with contract Section V (PATENT RIGHTS), the Contractor hereby certifies that there are no Subject Inventions relative to the fulfillment of this contract.

2.2 Descriptive Summary

The Boeing Company under contract to the Bureau of Naval Weapons has developed a digital computer program to predict the anti-icing requirements of engine inlets by calculating water droplet trajectories toward the particular inlet through a potential flow field. The incompressible potential flow field is determined by a separate computer program using known boundary conditions. The potential flow field results are combined with the droplet and flow physical data for input to a droplet trajectory computer program. This water droplet trajectory program is based on balancing the drag, buoyancy and weight forces on the water droplet with the rate of change of the water droplet momentum.

The water droplet computer program is extremely versatile because the actual design shape is used in the study and therefore the results do not need to be extrapolated.

Verification of the computer programs was accomplished by comparisons with theoretical and NACA test and analytical data.

3.0 CONCLUSIONS

Based on historical and current development experience as shown by the references and this report, it is concluded that:

1. This method of determining water impingement on cylindrical and plane objects is accurate, efficient and rapid.

2. The water droplet trajectory computer program can be used to predict the trajectories of large and small water droplets starting at arbitrary conditions.

3. The potential flow computer program is a useful tool to describe complex flow fields to assist in surface and duct pressure distribution studies.

4. The developed computer program does not contain any new inventions or discoveries that could be classed as a Subject Invention (contract section V).

4.0 RECOMMENDATIONS

Based on the results and conclusion described in this document, it is recommended that:

1. This method of water impingement prediction be adopted as a standard for anti-icing requirements determination.

2. Development of the water droplet trajectory program be continued to expand its utility, simplify its usage and increase its accuracy especially for heavier, larger particles.

3. The potential flow computer program be expanded to include compressibility effects.

5.0 DESCRIPTION

A preliminary water droplet trajectory calculation computer program was refined, expanded and verified during the completion of Item 1 of the subject contract. Specific effort was devoted to accomplishing the following.

(a) The potential flow field computer program was updated and refined for increased utility and versatility. Variable distance incrementation in both the horizontal and vertical (or radial) directions was provided to more adequately define a potential flow field by using fine increments in areas of special interest. The potential flow program output was also increased to include velocity, velocity ratio and pressure coefficients at coordinate points on selected streamlines.

(b) The water droplet trajectory computer program's capabilities were expanded to permit the calculation of smaller droplet trajectories and droplet direction reversal. These changes were accomplished by making the water droplet horizontal distance increments independent of those selected for the potential flow grid and by using an additional set of equations for vertical travel. For droplet direction change calculations, the droplet primary direction of travel is changed whenever the perpendicular distance traveled is greater than an element in that direction. The droplet direction is reversed whenever the droplet inertia is insufficient to carry it across an increment and the perpendicular velocity component is small.

(c) The water droplet trajectory computer program was modified to provide for variation in the water droplet's initial state with respect to the free stream flow (random input) and for the effects of gravity and buoyancy on the droplet trajectories. The random input feature permits starting the droplet trajectory from any point in the potential flow field with any velocity components. The inclusion of gravity and buoyancy effects destroys the symmetry of the cylindrical flow field trajectories and requires calculations with negative ordinate values. The capability of calculating below the centerline has been included.

(d) The water droplet trajectory computer program was verified by comparison of the program's impingement predictions with NACA analytical and test results. Analytical comparisons were made against the NACA cylinder, sphere and airfoil data and comparisons with the test data were made for the sphere, airfoil and supersonic engine inlet.

6.0 METHOD

This analytical method of water droplet impingement prediction utilizes a digital computer and a finite difference solution of the trajectory equations of particles in an incompressible potential flow field. The potential flow field is determined from known boundary stream function values by using the relaxation technique for solving the stream function equations derived from continuity and irrotationality.

The results of the potential flow computer program are used with additional water droplet and flow physical data as input to the water droplet trajectory computer program. The water droplet trajectory program calculations result in an accurate determination of water impingement intensity on a desired surface.

Flow charts and listing of the computer programs are presented in appendices 1 through 4.

6.1.1 Potential Flow Field Analysis

Since the computer program solves the stream function equations to obtain the potential flow field from boundary stream function values, these boundary values must be determined analytically. The generalized flow field for an engine inlet is shown on page 12. Using this figure with the y-values as radial lenghts, we can develop the boundary values as follows.

For two dimensional (plane) incompressible steady fluid flow, the velocity component in any direction is found by differentiating the stream function at right angles to that direction. Assuming the positive flow direction as upward (increasing y-values) and to the right (increasing x-values) the velocity components are

$$v_x = \frac{\partial \psi}{\partial y} \tag{1}$$

$$v_y = -\frac{\partial \psi}{\partial x} \tag{2}$$

In the case of axially-symmetric flow, such as that of an engine inlet the velocity components are

$$v_x = v_{axial} = \frac{\partial \psi}{y\,\partial y} \tag{3}$$

$$v_y = v_{radial} = -\frac{\partial \psi}{y\,\partial x} \tag{4}$$

The airflow toward the engine inlet is assumed to be from left to right with a value of U_∞ at the left boundary. At this boundary the flow velocity is uniform and horizontal. Therefore,

$$v_{y1} = v_y \text{ left} = 0 \qquad (5)$$

$$v_{x1} = v_x \text{ left} = \frac{\partial \psi}{y \ \partial y} = U_1 = U_\infty = k \ U'_\infty \qquad (6)$$

then $\int d\psi = \int k \ U'_\infty \ y dy$ (7)

and $\psi = k \ U'_\infty \ y^2/2 + C$ (8)

or $\psi_1 = k \ U'_\infty \ y_1^2/2 + C$ (9)

To evaluate the constants (k and c) the following boundary conditions are used.

(a) $\psi_1 = 0$ at $y_1 = 0$

(b) $\psi_1 = 1$ at $y_1 = R_\infty$ (stream tube radius)

Using (a) c is evaluated as zero.

Using (b) k is evaluated as $2/(U'_\infty R^2_\infty)$ and equation (9) can be rewritten as

$$\psi_1 = y_1^2 / R^2_\infty \qquad (10)$$

At the radial extremity of the control volume (y = y max)

$$\psi_1 = \psi_{max} = (y_{max}^2 / R^2_\infty)$$

Since y_{max} is a constant and assumed to be the ordinate of a horizontal stream line, ψ_{max} is a constant for all values of x.

The arbitrary selection of the stream function value of one (1.0) for the value at the stream tube radius on the left boundary establishes this value everywhere on the stream tube boundary including the cowl surface. Similarly, selection of the zero stream function value on the centerline establishes this value on the centerline and on the centerbody surface.

At the right boundary, uniform horizontal flow is assumed to exist within the inlet and also outside the cowl. These flows can (and generally do) have different velocities. With these assumptions the following basic equation can be developed and evaluated.

$$\psi = k \; U \; y^2/2 + c. \tag{11}$$

Inside the engine inlet

$$\psi = 0 \text{ at } y_r = y_{ocb} \quad \text{so} \quad c_2 = - k_2 \, U \, y^2_{ocb}/2.$$

$$\psi = 1.0 \text{ at } y_r = y_{ic} \quad \text{so} \quad k_2 = 2/(U\,(y^2_{ic} - y^2_{ocb})).$$

Then

$$\psi_{ri} = (y_r^2 - y_{ocb}^2) \,/\, (y_{ic}^2 - y_{ocb}^2) \tag{12}$$

This equation for the stream function is valid between the engine inlet centerbody and cowl at the right side of the control volume.

Outside the engine inlet equation (11) is evaluated with these conditions.

$$\psi = 1.0 \text{ at } y_r = y_{oc}$$

$$\psi = \psi_{max} = \frac{y_m^2}{R_\infty^2} \text{ at } y_r = y_{max}$$

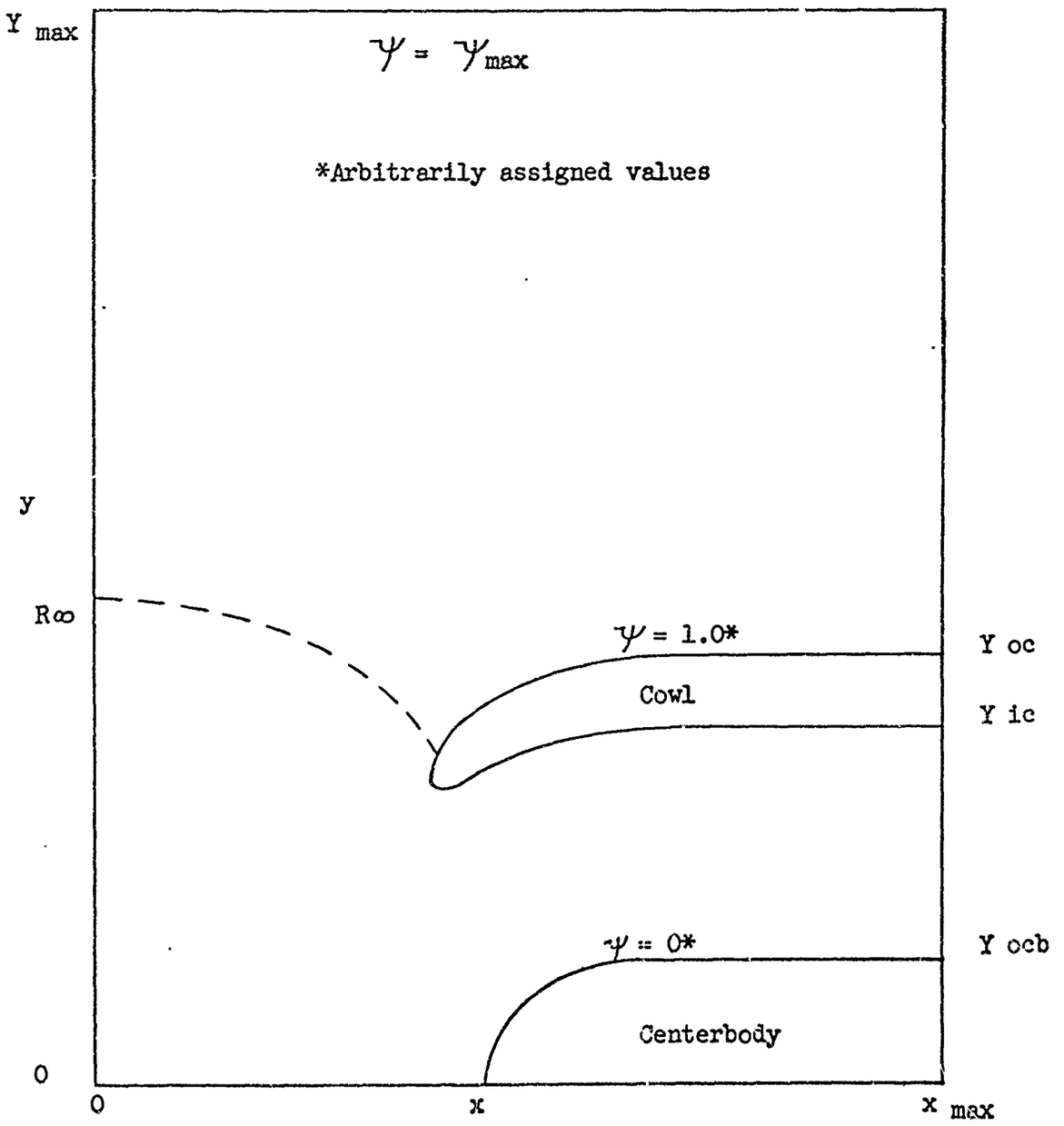
Then $c_3 = 1 - k_3 \; U \; y_{oc}^2 / 2$

$$k_3 = (2/U)\,(y^2_{max}/R^2_\infty - 1)/(y^2_{max} - y^2_{oc})$$

$$\psi_{ro} = 1 + (y_r^2 - y_{oc}^2)\;(y_{max}^2/R_\infty^2 - 1)/(y_{max}^2 - y_{oc}^2) \tag{13}$$

This equation is used to evaluate the stream function values on the right control volume boundary in the constant velocity region outside the engine inlet.

For a particular engine inlet configuration and engine power setting (airflow requirement), the input left and right boundary stream function values for the potential flow program are determined from equations (10), (12) and (13). The lower, upper and model surface stream function boundary values are arbitrarily assigned at zero, maximum and one, respectively.



CALC			REVISED	DATE	GENERALIZED ENGINE INLET FLOW FIELD	D3-6961
CHECK						
APR						
APR					THE BOEING COMPANY	
					WICHITA DIVISION WICHITA 1 KANSAS	PAGE 12
CONTRACT NO.						

E-1196 R1

6.1.2 Potential Flow Computer Program

Incompressible, inviscous and irrotational fluid flow problems in plane and axially-symmetric flow fields can be solved with the assistance of the IBM 7094/ FORTRAN digital computer program described herein.

Input data for the program includes a geometric description of the flow field boundaries and stream function values for the boundaries. Program results include a complete description of the potential flow field by enumerating stream function values at mesh intersections and streamline coordinates, as well as fluid velocities and pressure coefficients for selected streamlines.

The stream function, psi (ψ), for incompressible, steady, fluid flow is defined as the volume of fluid passing between a base streamline and the point in question. For a plane flow field the field depth is considered to be one unit and for an axially symmetric flow field, the depth varies with the radius.

The velocity component in any direction is the differential of the stream function at right angles to that direction. The usual convention of positive flow upward and to the right results in the following derivatives for velocity components.

Field	Plane	Axially Symmetric
Horizontal velocity, Vx	$\partial\psi/\partial y$	$(1/y)\ (\partial\psi/\partial y)$
Vertical velocity, Vy	$-\partial\psi/\partial x$	$(-1/y)\ (\partial\psi/\partial x)$

Limiting the flow to be irrotational introduces the following equation:

$$\partial V_Y/\partial X - \partial V_X/\partial Y = 0$$

or

$$\partial^2\psi/\partial X^2 + \partial^2\psi/\partial Y^2 = 0 \qquad (14)$$

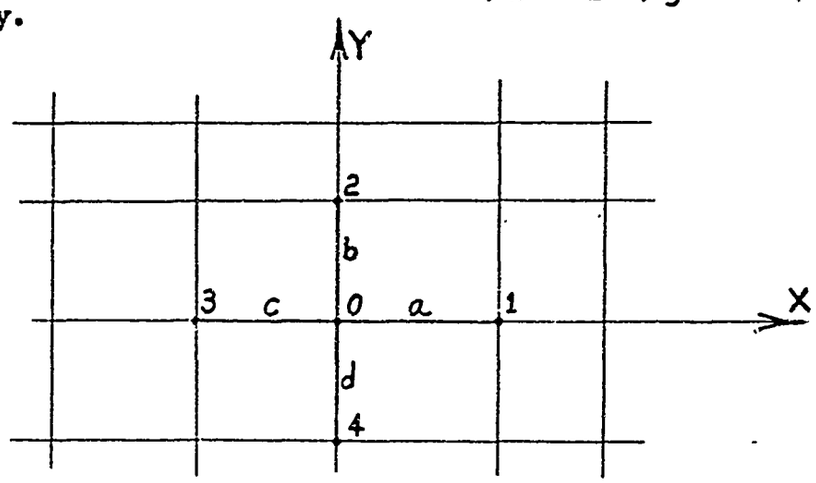
for plane flow. For axially symmetric flow this limitation yields

$$\partial/\partial x\ (-1/y\ \ \partial\psi/\partial x) - \partial/\partial Y(1/Y\ \ \partial\psi/\partial Y) = 0$$

$$\partial^2\psi/\partial X^2 + \partial^2\psi/\partial Y^2 - 1/Y\ \partial\psi/\partial Y = 0 \qquad (15)$$

Equations (14) and (15) define the stream function for incompressible, steady, inviscous and irrotational fluid flow in plane and axially symmetric flow fields. Numerical evaluation of these equations by the relaxation technique is the primary function of the potential flow computer program.

The potential flow computer program is divided into three distinct sections: potential flow field solution; derivative calculation and contour mapping. The potential flow field solution is accomplished by a finite difference approximation of equations (14) and (15). This solution is based on the use of a rectangular grid with the origin of coordinates temporarily at the point being considered. For a typical mesh point, ψ_0 is the stream function to be calculated and a, b, c and d are the distances to adjacent mesh points with stream function values of ψ_1, ψ_2, ψ_3 and ψ_4, respectively.



An example of a mesh point coordinate system showing variable mesh spacing and coordinate orientation is illustrated by this sketch. The average first partial derivaties of ψ with respect to x from the origin to points 1 and 3 are:

$$\partial\psi/\partial x)_{0,1} = \Delta\psi/\Delta x)_{0,1} = (\psi_1 - \psi_0)/a \qquad (16)$$

$$\partial\psi/\partial x)_{3,0} = \Delta\psi/\Delta x)_{3,0} = (\psi_0 - \psi_3)/c \qquad (17)$$

The average second partial derivative of ψ with respect to x to be used between the distance a/2 to the right of the origin to c/2 to the left of the origin is

$$\partial^2 \psi/\partial x^2)_0 = \left(\partial\psi/\partial x)_{0,1} - \partial\psi/\partial x)_{3,0}\right) \Big/ (a + c) / 2$$

$$= 2(\psi_1 - \psi_0)/(a\,(a+c)) - 2\,(\psi_0 - \psi_3)/\,(\,c\,(a+c)) \qquad (18)$$

By similar analysis in the vertical direction

$$\partial^2 \psi/\,\partial y^2)_0 = 2\,(\psi_2 - \psi_0\,)/\,(b(b+d)) - 2(\psi_0 - \psi_4)/(d(b+d)) \qquad (19)$$

The first partial derivative of ψ with respect to y at the origin is approximated by a linear interpolation of the average first partial derivaties in the adjacent intervals above and below the origin.

$$\partial\psi/\partial y)_0 = (d(\partial\psi/\partial y)_{0,2} + b\,(\partial\psi/\partial y)_{4;0})/\,(b + d)$$

$$= d\,(\psi_2 - \psi_0)\,/\,(b(b+d)) + b\,(\psi_0 - \psi_4)\,/\,(d(b+d)) \qquad (20)$$

Substitution of equations (18), (19) and (20) into equations (14) and (15) results in the following finite difference approximations for plane and axially symmetric flow, respectively.

$$\psi_0 = \frac{\dfrac{2\psi_1}{a(a+c)} + \dfrac{2\psi_2}{b\,(b+d)} + \dfrac{2\psi_3}{c\,(a+c)} + \dfrac{2\psi_4}{d(b+d)}}{\dfrac{2}{a(a+c)} + \dfrac{2}{b\,(b+d)} + \dfrac{2}{c(a+c)} + \dfrac{2}{d\,(b+d)}} \qquad (21)$$

$$\psi_0 = \frac{\dfrac{2\psi_1}{a(a+c)} + \dfrac{(2 - d/y)\psi_2}{b\,(b+d)} + \dfrac{2\psi_3}{c\,(a+c)} + \dfrac{(2 + b/y)\psi_4}{d(\,b + d)}}{\dfrac{2}{a(a+c)} + \dfrac{(2-d/y)}{b(b+d)} + \dfrac{2}{c(a+c)} + \dfrac{(2+b/y)}{d(b+d)}} \qquad (22)$$

In the computer program equations (22) is reduced to equation (21) for plane flow by making d/y and b/y equal zero.

The computer program constructs a finite difference solution at each mesh point and sweeps through them in a definite order. Repetitive calculations (sweeps) result in the stream function values converging to values consistent

with the specified boundary values. To improve convergence over-relaxation is used. That is, the calculated change in the stream function value at each point is multiplied by a relaxation factor which is greater than one. This result is added to the previous value and the sum is used for the new stream function value.

Sweeping is done in alternate directions, first by columns from lower left to upper right, and then, also by columns, from upper right to lower left. (This improves the rate of convergence somewhat for cases with normal derivative enforcement on the boundaries). A series of 40 sweeps (20 pairs of alternate direction sweeps) is carried out to establish the pattern of convergence; then the point with maximum error is located and followed through ten more sweeps. The change in the indicated error over this last sequence of eleven sweeps is used to define the ratio for a geometric series. The ratio is used to extrapolate the migration of the field over the sequence of eleven sweeps to an indicated limit, subject to a hedge factor and a maximum limit on the indicated extrapolation. This procedure has been found to accelerate the convergence of well behaved configurations by about 30% and the convergence of poorly behaved configurations by a factor of three.

The second part of the computer program operates on the potential flow field obtained above and calculates the partial derivatives of Ψ with respect to x and y at all mesh points. These derivatives are obtained by linear interpolation using the Ψ values at the desired point and at the adjacent points on each side at the same ordinate or abscissa value. Derivatives at points on boundaries are extrapolated linearly from the two points closest to the boundary at the same ordinate or abscissa value.

The absolute velocity at each mesh point is calculated as the square root of the sum of the squares of the x and y partial derivatives of Ψ at the point. For axially symmetric flow fields this calculation includes division by the y value (radius) to obtain the true velocity.

Contour mapping of selected stream lines or constant partial derivatives is accomplished in the third program section. The ordinate values of stream lines (stream function isolines) or contours of constant partial derivative values are determined at the input abscissa values from the potential flow field and/or either of the derivative fields by linear interpolation. The flow velocity at the stream line coordinates is also determined by interpolation.

The pressure coefficient is calculated at each point on the contour lines. The pressure coefficient is calculated as 1 -(local velocity/free stream velocity)^2.

6.1.3 Potential Flow Computer Program Usage

Input for the potential flow computer program is prepared on six field, 72 column floating decimal data cards as illustrated on the following Potential Flow Program Input Format. The individual input quantities are described as follows.

TITLE	Identification of the problem - used as a heading in the printout.
AXYM	Flow field type key - zero for a plane flow field and one for the axially symmetric flow field.
CVGS	Relaxation convergence tolerance - when the maximum difference between two successive sweeps PSI value is less than this number, the flow field iteration will stop. This value should be greater than 3×10^{-8} due to machine limitations.
NSWPS	Number of sweeps - maximum number of sweeps permitted. The iteration will stop after this number of sweeps.
PUNCH	Potential flow field table punch key - table will be punched if this value is one.
NDERIV	Normal derivative key - one to force the normal derivatives at boundaries to zero. Normally input as zero and Y0 and Y1 are ignored.
Y0	Y-value of the lower boundary on which the normal derivatives are forced to zero.
Y1	Y-value of the upper boundary on which the normal derivatives are forced to zero.
*XSCL	Scale factor in the X direction for plot of stream lines equal to the number of inches which correspond to a unit change in X.
*YSCL	Scale factor in the Y direction.
*XL1	Lower limit in the X direction of area to be plotted.
*XL2	Upper limit in the X direction of area to be plotted.
*YL1	Lower limit in the Y direction of area to be plotted.
*YL2	Upper limit in the Y direction of the area to be plotted.

* XSCLA, YSCLA, XL1A, XL2A, YL1A and YL2A are the corresponding parameters for a second plot. If only one plot is desired the sixth card should be blank.

Y1, Y2, Y3, ... Ym	First, second, third ... m'th selected ordinate mesh line values.
Xa, Xb, Xc, ... $X_{n/2}$	First, second, third ... n/2'th abscissa boundary values at a given ordinate value.
$\psi a, \psi b, \psi c, \ldots \psi_{n/2}$	Stream function boundary values at Xa, Xb, Xc, ... $X_{n/2}$ or at Ya, Yb, Yc, ... $Y_{n/2}$.
n	The number of boundary coordinates and stream function values along that particular mesh line.
X1, X2, X3, ... Xm	First, second, third ... m'th selected abscissa mesh line values.
Ya, Yb, Yc, ... $Y_{n/2}$	First, second, third ... n/2'th ordinate boundary value at a given abscissa value.
NSL	Number of stream lines of a type (see CSCH).
CSCH	Determines whether stream lines connect points of equal stream function value (CSCH = 2), derivative of PSI with respect to X (CSCH = 3) or derivative of PSI with respect to Y (CSCH = 4).
NNLNS	Number of types of SLINES. NSL, CSCH, and the SLINES must be input NNLNS times.
SLINES	Values of PSI (or derivatives of PSI with respect to X or Y) through which lines will be drawn.

FLOATING DECIMAL DATA FOR 7094

Programmer Potential Flow Program Input Format

FIELD 1		FIELD 2		FIELD 3		FIELD 4		FIELD 5		FIELD 6		
Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	ID
TITLE												
TITLE		(continued)										
AXYM		CVGS		NSWPS		PUNCH						
NDERIV		YO		Y1								
XSCL		YSCL		XL1		XL2		YL1		YL2		
XSCLA		YSCLA		XL1A		XL2A		YL1A		YL2A		
Y_1		n		Xa		ψ_a		Xb		ψ_b		
Xc		ψ_c		Xd		ψ_d						
Xe		ψ_e		Xf		ψ_f						
...		...		...		...						
Xn/2-1		$\psi_{n/2-1}$		Xn/2		$\psi_{n/2}$						
Y_2		n		X_a		ψ_a		Xb		ψ_b		
Xc		ψ_c		X_d		ψ_d						
...		...		...		...						
Xn/2-1		$\psi_{n/2-1}$		Xn/2		$\psi_{n/2}$						
Y3		n		X_a		ψ_a		X_b		ψ_b		
Xc		ψ_c		X_d		ψ_d						

Comments:

A-3322

FLOATING DECIMAL DATA FOR 7094

Programmer Potential Flow Program Input Format

FIELD 1		FIELD 2		FIELD 3		FIELD 4		FIELD 5		FIELD 6		
Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	ID
...		...		...		...						
Xn/2-1		$\psi_{n/2-1}$		Xn/2		$\psi_{n/2}$						
...		...		...		...		...		...		
Ym		..n		Xa		ψ_a		Xb		ψ_b		
Xc		ψ_c		Xd		ψ_d						
...		...		...		...						
Xn/2-1		$\psi_{n/2-1}$		Xn/2		$\psi_{n/2}$						
Blank Card												
X_1		1. n		Y_a		ψ_a		Y_b		ψ_b		
Y_c		ψ_c		Y_d		ψ_d						
...		...		...		...						
Yn/2-1		$\psi_{n/2-1}$		Y n/2		$\psi_{n/2}$						
X_2		n..		Y_a		ψ_a		Y_b		ψ_b		
Y_c		ψ_c		Y_d		ψ_d						
...		...		...		...						
Yn/2-1		$\psi_{n/2-1}$		Yn/2		$\psi_{n/2}$						
...		...		...		...						

Comments:

A-3322

FLOATING DECIMAL DATA FOR 7094

Programmer Potential Flow Program Input Format

Page 21 of

FIELD 1		FIELD 2		FIELD 3		FIELD 4		FIELD 5		FIELD 6		ID
Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	
X_m		ψ_n		Y_a		ψ_a		Y_b		ψ_b		
Y_c		ψ_c		Y_d		ψ_d						
...		...		...		...						
$Y_{n/2-1}$		$\psi_{n/2-1}$		$Y_{n/2}$		$\psi_{n/2}$						
		Blank card										
NSL		CSCH		NNLNS		XMAX		U'_∞				
SLINES(1)		SLINES (2)		SLINES (3)		SLINES (4)		SLINES (5)		SLINES (6)		
SLINES (7)		SLINES (8)		...		SLINES (NSL)						

Comments:

A-3328

Output from the potential flow computer program is self explanatory and is printed in the following order.

Input	Input is printed out to provide a permanent record.
Sweep history	Record of convergence is printed out with residue at point of largest deviation.
Potential flow field solution	- The stream function values, derivatives and flow velocity are printed out at each mesh point.
Contour lines	Stream line or constant partial derivative coordinate values, velocities and pressure coefficients at these points are printed.

On request, the program will also provide a plotter tape for a Gerber plotter of the contour lines. The potential flow field is punched on data cards in table form for use in the water droplet trajectory computer program.

6.2.1 Water Droplet Trajectory Analysis

The equations of motion for a water droplet in air are obtained by a summation of forces on the drop. Ignoring the gravitational force, the drag force on the water drop must be equal and opposite to the rate of change of the water drop momentum. The drag force in the x-direction is

$$D_x = C_D \left(\rho_a \mathcal{V}_{rel}^2 /2\right) \left(\pi a^2\right) \left(\mathcal{V}_x - \mathcal{V}_{xw}\right)/\mathcal{V}_{rel} \qquad (23)$$

where $\mathcal{V}_{rel} = \mathcal{V} - \mathcal{V}_w$.

The rate of change of drop momentum in the x-direction is

$$F_x = \text{mass x acceleration} = \left(4 \pi a^3 \rho_w /3\right) d\mathcal{V}_{xw}/dt. \qquad (24)$$

The drop Reynolds number is

$$R = \left(2 a \rho_a \mathcal{V}_{rel}\right)/\mu \qquad (25)$$

then $$D_x = C_D R \pi a \mu \left(\mathcal{V}_x - \mathcal{V}_{xw}\right)/ 4 \qquad (26)$$

and $$F_x = D_x \qquad (27)$$

$$\left(4 \pi a^3 \rho_w/3\right) d\mathcal{V}_{xw}/dt = \left(C_D R/4\right) \pi a \mu \left(\mathcal{V}_x - \mathcal{V}_{xw}\right) \qquad (28)$$

or $$\left(2 a^2 \rho_w U /9\mu\right) d\dot{\mathcal{V}}_{xw}/dt = \left(U C_D R/24\right) \left(\dot{\mathcal{V}}_x - \dot{\mathcal{V}}_{xw}\right) \qquad (29)$$

where $\mathcal{N}_{xw} = U\dot{\mathcal{N}}_{xw}$, $\mathcal{N}_x = U\dot{\mathcal{N}}_x$

$$d\mathcal{N}_{xw} = U d\dot{\mathcal{N}}_{xw}$$

$$d\dot{\mathcal{N}}_{xw}/\,dt = (UC_DR/24K)\,(\dot{\mathcal{N}}_x - \dot{\mathcal{N}}_{xw}) \tag{30}$$

where $$K = 2\,a^2\rho_w U/9\mu \tag{31}$$

and since $dt = dx/\dot{\mathcal{N}}_{xw}U$ (32)

$$\dot{\mathcal{N}}_{xw}\,d\dot{\mathcal{N}}_{xw} = (C_DR/24K)\,(\dot{\mathcal{N}}_x - \dot{\mathcal{N}}_{xw})\,dx. \tag{33}$$

Similarily in the y-direction

$$d\dot{\mathcal{N}}_{yw}/dt = UC_DR/24K\,(\dot{\mathcal{N}}_y - \dot{\mathcal{N}}_{yw}) \tag{34}$$

Equation (33) can be integrated using mean values of velocity over an interval.

$$\int_{(\dot{\mathcal{N}}_{xw})_n}^{(\dot{\mathcal{N}}_{xw})_{n+1}} \dot{\mathcal{N}}_{xw}\,d\dot{\mathcal{N}}_{xw} = (C_DR/24K)_{mean}\,(\dot{\mathcal{N}}_x - \dot{\mathcal{N}}_{xw})_{mean} \int_{X_n}^{X_{n+1}} dx \tag{35}$$

$$\left[(\dot{\mathcal{N}}_{xw})^2_{n+1} - (\dot{\mathcal{N}}_{xw})^2\right] \Big/ 2$$

$$= (C_DR/24K)_{mean}\,(\dot{\mathcal{N}}_x - \dot{\mathcal{N}}_{xw})_{mean}(X_{n+1} - X_n) \tag{36}$$

Equation (34) can be integrated in a similar manner.

$$\int_{(\dot{\mathcal{N}}_{yw})_n}^{(\dot{\mathcal{N}}_{yw})_{n+1}} d\dot{\mathcal{N}}_{yw} = (UC_DR/24K)_{mean}\,(\dot{\mathcal{N}}_y - \dot{\mathcal{N}}_{yw})_{mean} \int_{t_n}^{t_{n+1}} dt \tag{37}$$

$$(\dot{v}_{yw})_{n+1} - (\dot{v}_{yw})_n = (UC_DR/24K)_{mean} (\dot{v}_y - \dot{v}_{yw})_{mean}(t_{n+1}-t_n) \quad (38)$$

From equation (32) is obtained

$$t_{n+1} - t_n = 2(X_{n+1} - X_n) \Big/ \left[U(\dot{v}_{xwn+1} + \dot{v}_{xwn})\right] \quad (39)$$

and similarily

$$Y_{n+1} = Y_n + U \left[(\dot{v}_{yw})_{n+1} + (\dot{v}_{yw})_n\right] (t_{n+1} - t_n)/ 2. \quad (40)$$

By similar analysis using mean values of velocity over an interval in the y-direction the following equations are derived.

$$\left[(\dot{v}_{yw})^2_{n+1} - (\dot{v}_{yw})^2_n\right] \Big/ 2 = \left(\frac{C_DR}{24K}\right)_{mean} (\dot{v}_y - \dot{v}_{yw})_{mean} (Y_{n+1} - Y_n) \quad (41)$$

$$(\dot{v}_{xw})_{n+1} - (\dot{v}_{xw})_n = (UC_DR/24K)_{mean}(\dot{v}_x - \dot{v}_{xw})_{mean}(t_{n+1} - t_n) \quad (42)$$

$$t_{n+1} - t_n = 2(Y_{n+1} - Y_n) \Big/ U((\dot{v}_{yw})_{n+1} + (\dot{v}_{yw})_n) \quad (43)$$

$$X_{n+1} = X_n + U \left[(\dot{v}_{xw})_{n+1} + (\dot{v}_{xw})_n\right] (t_{n+1} - t_n)/2 \quad (44)$$

Equations (36), (38), (39) and (40) are the basic equations of the water droplet trajectory computer program for droplet travel that is predominately in the horizontal direction. For droplet travel in the vertical direction equations (41) through (44) are used. The use of equations for travel either horizontally or vertically permits the calculation of droplet trajectories in areas where a fixed interval is not traversed in the selected direction.

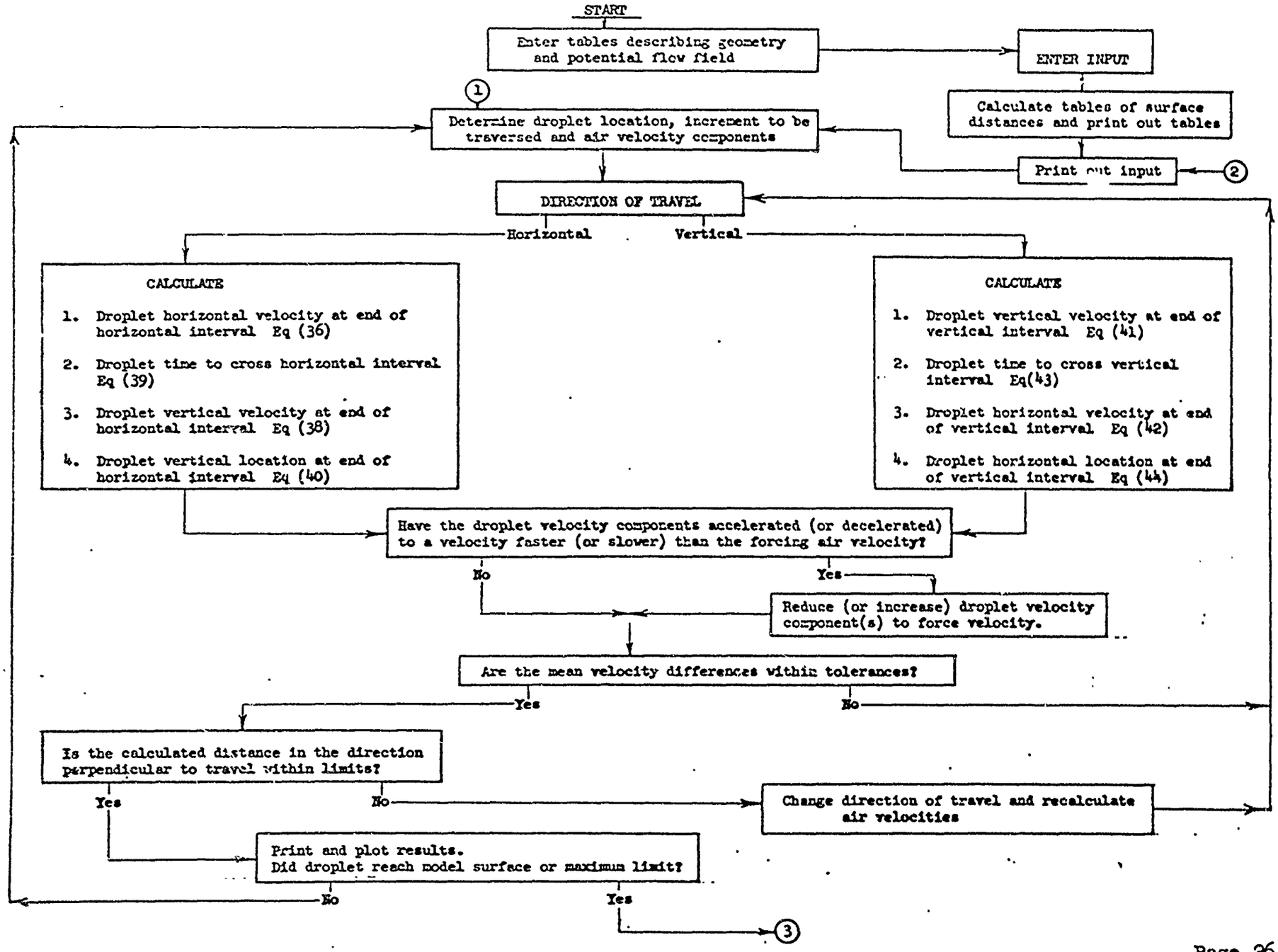
The inclusion of gravitational and buoyancy effects on the water droplet trajectories results in the following modifications to the trajectory equations. Equation (38) contains the additional term $-(g/\rho_w U)(\rho_w - \rho_a)(t_{n+1} - t_n)$ on the right side. Equation (41) must include the term $-(g/\rho_w U^2)(\rho_w - \rho_a)(Y_{n+1} - Y_n)$ on the right side.

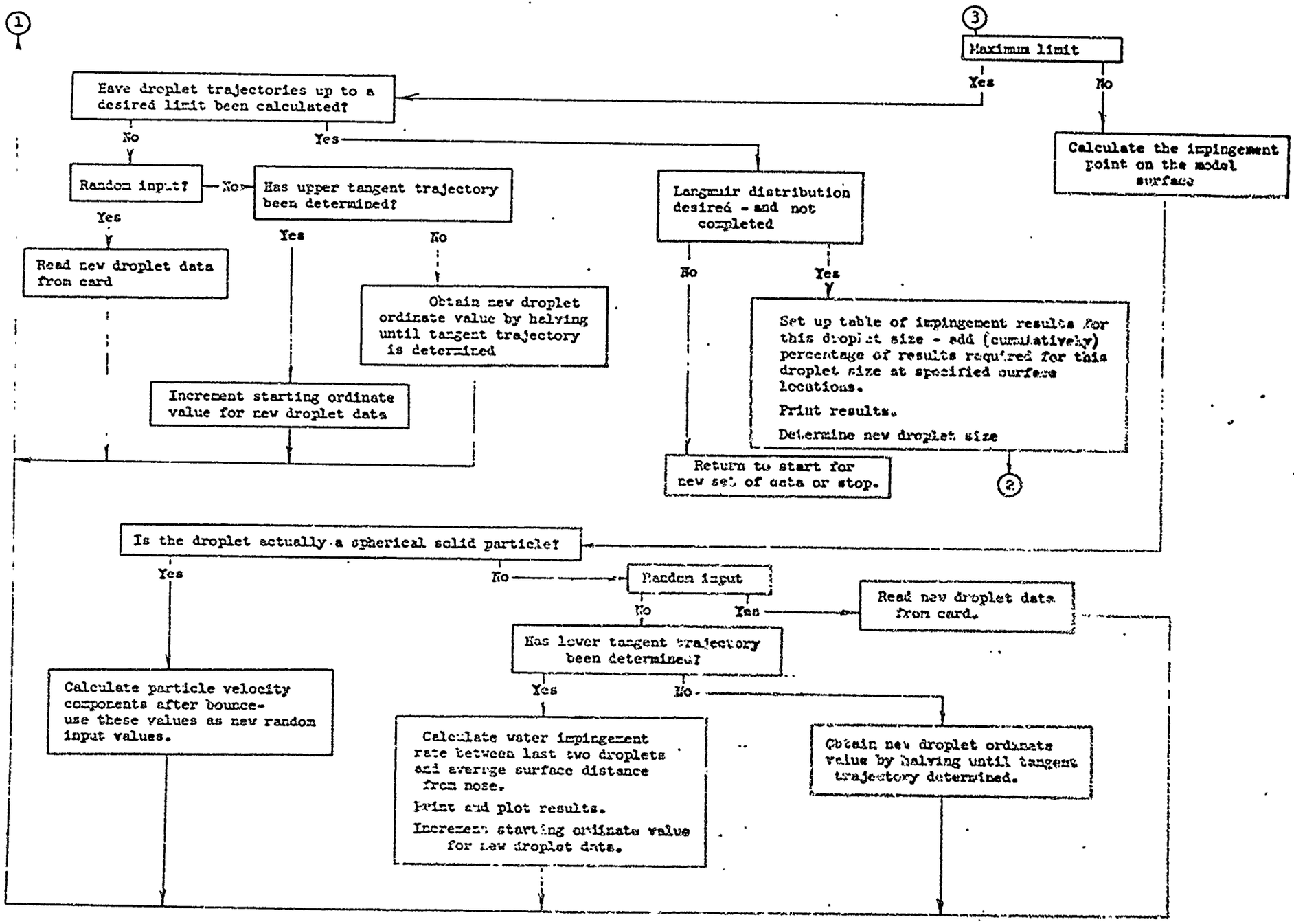
6.2.2 Water Droplet Trajectory Computer Program

The water droplet trajectory computer program traces the trajectories of individual drops across finite incremental distances through the incompressible potential flow field until they impinge on a surface or pass out of the area of interest. This program includes air velocity components and trajectory routines with appropriate tests as shown on the following abreviated flow chart and on the detail flow chart and listing of the appendix.

A droplet size distribution impingement intensity summation routine was added to the water droplet trajectory program as an option. This routine was added to provide a more accurate comparison of the program results with experimental data in artificial and natural icing conditions. The Langmuir "D" droplet size distribution of reference 1 was selected and incorporated in the program. Other droplet distributions could be added with small program changes. The Langmuir "D" distribution is as follows.

Liquid water in size group, percent	Average group drop radius/volume mean drop radius
5	0.31
10	0.52
20	0.71
30	1.00
20	1.37
10	1.74
5	2.22


START
Enter tables describing geometry and potential flow field
ENTER INPUT
Calculate tables of surface distances and print out tables
Print out input
2
1
Determine droplet location, increment to be traversed and air velocity components
DIRECTION OF TRAVEL
Horizontal
Vertical
CALCULATE
1. Droplet horizontal velocity at end of horizontal interval Eq (36)
2. Droplet time to cross horizontal interval Eq (39)
3. Droplet vertical velocity at end of horizontal interval Eq (38)
4. Droplet vertical location at end of horizontal interval Eq (40)
CALCULATE
1. Droplet vertical velocity at end of vertical interval Eq (41)
2. Droplet time to cross vertical interval Eq(43)
3. Droplet horizontal velocity at end of vertical interval Eq (42)
4. Droplet horizontal location at end of vertical interval Eq (44)
Have the droplet velocity components accelerated (or decelerated) to a velocity faster (or slower) than the forcing air velocity?
No
Yes
Reduce (or increase) droplet velocity component(s) to force velocity.
Are the mean velocity differences within tolerances?
Yes
No
Is the calculated distance in the direction perpendicular to travel within limits?
Yes
No
Change direction of travel and recalculate air velocities
Print and plot results.
Did droplet reach model surface or maximum limit?
No
Yes
3


1
3
Maximum limit
Yes
No
Have droplet trajectories up to a desired limit been calculated?
No
Yes
Calculate the impingement point on the model surface
Random input?
No
Has upper tangent trajectory been determined?
Langmuir distribution desired - and not completed
Yes
Read new droplet data from card
Yes
No
Obtain new droplet ordinate value by halving until tangent trajectory is determined
No
Yes
Set up table of impingement results for this droplet size - add (cumulatively) percentage of results required for this droplet size at specified surface locations.
Print results.
Determine new droplet size
Increment starting ordinate value for new droplet data
Return to start for new set of data or stop.
2
Is the droplet actually a spherical solid particle?
Yes
No
Random input
No
Yes
Read new droplet data from card.
Has lower tangent trajectory been determined?
Calculate particle velocity components after bounce- use these values as new random input values.
Yes
No
Calculate water impingement rate between last two droplets and average surface distance from nose.
Print and plot results.
Increment starting ordinate value for new droplet data.
Obtain new droplet ordinate value by halving until tangent trajectory determined.

6.2.3 Water Droplet Trajectory Computer Program Usage

Input for the water droplet trajectory computer program consists of geometric and potential flow descriptive tables and data cards. The tables are read first in numerical order from one through nine. Table arrangement is shown on the following Table Input Format. Symbols used in the Table Input Format are described as follows.

YNO	Number of ordinate values in the table plus one. (NOTE: YNO is 2. for a single function table.)
XNO	Number of abscissa values in the table plus one.
Table No.	Index number of table.
$Y_1, Y_2 \ldots Y_n$	First, second ... n'th ordinate values
$X_1, X_2 \ldots X_n$	First, second ... n'th abscissa values
$F(X_i, Y_j)$	Dependent variable value at (X_i, Y_j) where i has values 1, 2, ... m and j has values 1, 2, ... n.

The physical orientation of the individual tables is shown on the Water Droplet Trajectory Computer Program Table Key. A written description of these tables is as follows.

FLOATING DECIMAL DATA FOR 7094

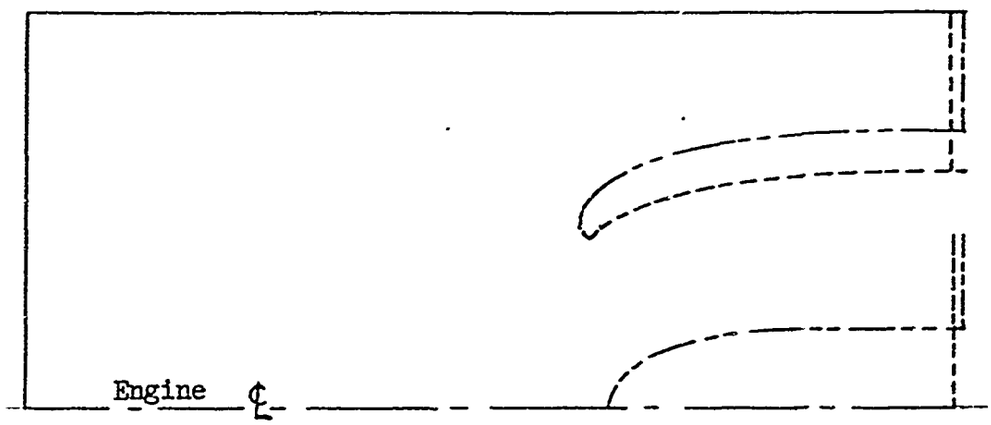
Table Input Format (Two dimensional)

Programmer _____

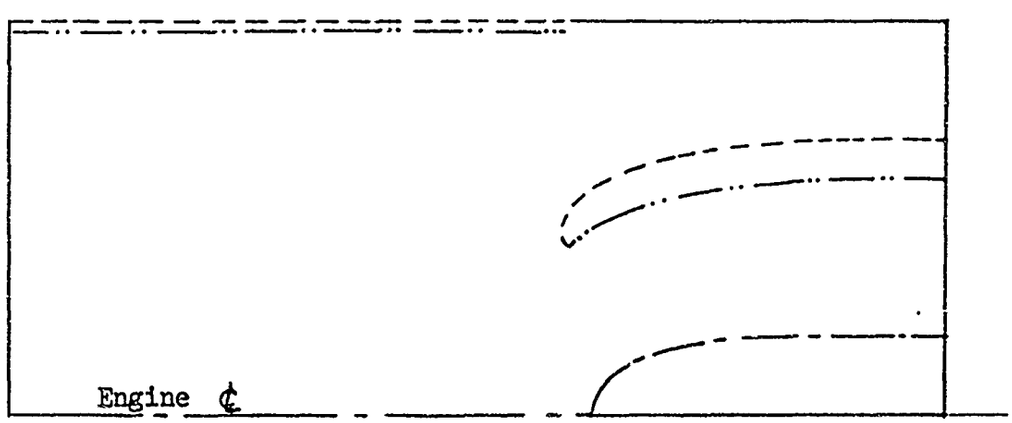
FIELD 1		FIELD 2		FIELD 3		FIELD 4		FIELD 5		FIELD 6		
Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	ID.
1.		YNO		XNO		Table No.						
0		Y_1		Y_2		Y_3		Y_4		Y_5		
Y_6		Y_7		. . .		Y_n		X_1		$F(X_1, Y_1)$		
$F(X_1, Y_2)$		$F(X_1, Y_3)$		. . .		$F(X_1, Y_n)$		X_2		$F(X_2, Y_1)$		
$F(X_2, Y_2)$		$F(X_2, Y_3)$		. . .		$F(X_2, Y_n)$		. . .		X_m		
$F(X_m, Y_1)$		$F(X_m, Y_2)$		$F(X_m, Y_3)$		. . .		$F(X_m, Y_n)$				

Comments:

Water Droplet Trajectory Computer
Program Table Key



——— XB2 Table 4
- - - - - XB3 Table 5



——— YB1 Table 8*
—··—·· YB2 Table 6**
- - - - - YB3 Table 7**

* Table 9 is generated from Table 8 from the centerbody highlight aft.

** Tables 1 and 2 are generated from Tables 6 and 7, respectively, from the cowl highlight aft.

Table Number	Variable Description
1	Lower cowl surface distance from highlight versus horizontal distance to right of highlight. A dummy table is input and the actual values are generated internally from Table 6 values. (SUP vs. X-XHL)
2	Upper cowl surface distance from highlight versus horizontal distance to right of highlight. A dummy table is input and the actual values are generated internally from Table 7 values. (SUP vs. X-XHL).
3	Stream function values versus X and Y. This is the only double function table used and it is generated by the potential flow program. The X and Y values used are the mesh points (ψ vs. X, Y)
4	First boundary value of X after left hand boundary at given Y value (XB2 vs. Y).
5	Second boundary value of X after left hand boundary at given Y value. (XB3 vs. Y)
6	First boundary value of Y above lower boundary at given X value. (YB2 vs. X).
7	Second boundary value of Y above lower boundary at given X value (YB3 vs. X)
8	Lowest boundary value of Y at given X value. This is normally an extension of the engine centerline and the centerbody surface (YB1 vs. X)
9	Centerbody surface distance from highlight versus horizontal distance to right of highlight. A dummy table is input and the actual values are generated from table 8 values. (SUP vs. X-XHL).

After the tables are read into the computer, the input data is called. These data are arranged on six field 72-column floating decimal cards as shown on the following Water Droplet Trajectory Computer Program Input Format. The individual input quantities are described as follows.

Identification	Problem title - used as a heading on printout and title on machine plot.
Tape No.	Index number of plotting tape unit (usually 6)
X scale 1	X-scale factor for first curve
Y scale 1	Y-scale factor for first curve
X set 1	Minimum X-value of first curve-inches
Y set 1	Minimum Y-value of first curve-inches
X max 2	Maximum X-value of second curve-inches
Y max 2	Maximum Y-value of second curve - inches
X scale 2	X-scale factor for second curve
Y scale 2	Y-scale factor for second curve
X set 2	Minimum X-value of second curve-inches
Y set 2	Minimum Y-value of second curve-inches
LWC	Liquid water content of air-lb/cu. inch
a	Droplet radius - ft.
U_∞	Actual remote free stream velocity - ft/sec.
U'_∞	Potential flow remote free stream velocity - relative
ρ_a	Air density - lb/cu. ft.
μ_a	Air viscosity - lb/ft. sec.
Xmax	Maximum value of X-inches
Ymax	Maximum value of Y -inches
Ydrop	Initial Y-value for first water droplet-inches
ΔYirop	Vertical increment between water droplets at initial X-value - inches
XL	Initial X -value for water droplets - inches
X_{HL1}	Centerbody highlight X-value- inches

Y_{SL}	Minimum model Y-value-inches
Y_{SM}	Maximum model Y-value for droplet calculations-inches
V_{XWIN}	Initial droplet velocity in the X-direction-ft/sec.
V_{YWIN}	Initial droplet velocity in the Y-direction-ft/sec.
DIRECT	Direction of droplet travel key - negative is for vertical travel and positive indicates horizontal travel.
RANDOM	Random input key - negative is for normal input (increment Y drop) and positive indicates random input.
X_{HL}	Cowl highlight X-value-inches
Y_{HL}	Cowl highlight Y-value-inches
ALAMDA	Flow field type key - zero indicates plane flow and one indicates axially symmetric flow.
RESIL	Solid particle-surface resilience factor-ratio of the particle normal velocity after impingement to the normal velocity before impingement (absolute values)
ρ_w	Particle density - lb/cu. ft.
Langmr	Water droplet distribution summation key -mean droplet size of Langmuir D distribution if desired, otherwise input zero.
GRAVTY	Gravity and buoyancy effects key - zero if effects neglected and one to include effects in the calculations.
$\Delta X1$	Coarse increment of X for calculations-inches
$\Delta X2$	Fine increment of X for calculations-inches
XREF1	X-value at change from coarse to fine X-increment-inches
XREF2	X-value at change from fine to coarse X-increment-inches

YTRAP1	Lower limit of particle trap - inches.
YTRAP2	Upper limit of particle trap - inches.
XTRAP1	Left limit of particle trap - inches.
XTRAP2	Right limit of particle trap - inches.
TABLE	Tables 1, 2 and 9 generation key - calculate if 1.0 and omit if zero.

FLOATING DECIMAL DATA FOR 7094

Programmer _____ Water Droplet Trajectory Computer Program Input Format

FIELD 1		FIELD 2		FIELD 3		FIELD 4		FIELD 5		FIELD 6		
Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	Co-efficient	Exp.	ID
Identification												
Tape No.		0		Xscale 1		Yscale 1		Xset 1		Yset 1		
Xmax 2		Ymax 2		Xscale 2		Yscale 2		Xset 2		Yset 2		
LWC		a		U_∞		U'_∞		ρ_a		μ_a		
Xmax		Ymax		Ydrop		ΔYdrop		XL		XHL1		
YSL		Y_{SM}		V_{XWIN}		V_{YWIN}		DIRECT		RANDOM		
XHL		YHL		ALAMDA		RESIL		ρ_w		LANGMR		
GRAVTY		X1		X2		XREF1		XREF2				
Y_{TRAP1}		Y_{TRAP2}		X_{TRAP1}		X_{TRAP2}		TABLE				
YDROP		XL		VXW		VYW						
YDROP		XL		VXW		VYW						
YDROP		XL		VXW		VYW						
YDROP		XL		VXW		VYW						

Random input cards - as many as required followed by a blank card to trigger a new data set.

Comments:

A-3332

REV

E-30

Output of the water droplet trajectory computer program consists of tabular print out and a plotting magnetic tape. The tabular print out includes a listing of the tables and input data, droplet starting conditions, a step-by-step trajectory record and impingement data. If a droplet size distribution summation is requested, the results are printed at the end of each droplet size calculation and at the completion of the total distribution. The tabular print out includes the following items.

XNN1	X-value at completion of calculation across an interval. This value is the XNN (starting) value of the next interval - inches.
YMN1	Y-value at completion of calculation across an interval. This value is the YM (starting) value of the next interval - inches.
DTW	Time interval required for droplet to cross interval from XNN to XNN1. Units of feet at remote free stream velocity.
CK	Drag coefficient parameter, $C_D R_{rel}/24$
REN	Relative Reynolds Number of water droplet to air, R_{rel}.
VDX	Normalized average horizontal air velocity between XNN and XNN1.
VDXWN1	Normalized horizontal water droplet velocity at XNN1.
VDYWN1	Normalized vertical water droplet velocity at XNN1.
VDY	Normalized average vertical air velocity between XNN and XNN1.
XKT	Iteration of mean velocity counter used to control non-converging solutions.
DIRECT	Direction of droplet travel key.
WATRAT	Rate of water impingement between last two drops, lb/min-sq. in.
AVS	Average surface distance from highlight for impingement of last two droplets - inches.
XP	X-value at droplet impingement - inches.

YP	Y-value at droplet impingement - inches.
WATER	Water contained in area between starting points of last two droplets lb/sec.- in. of depth.

7.0 RESULTS AND DISCUSSION

Comparison of the computer program results with theoretical and NACA analytical and test data for program verification is shown on Figures 1 through 8. Figures 1, 2 and 3 verify the accuracy of the potential flow computer program for both plane and axially-symmetric flow fields.

Stream lines calculated by the updated potential flow computer program around a cylinder at right angles to a uniform stream (plane flow) are shown on Figure 1. Theoretical stream line points are shown for comparison. This Figure indicates complete agreement between theory and the program results in the area that would affect water droplet impingement with a slight variation further from the cylinder.

Figure 2 presents the potential flow stream lines for a sphere in a uniform stream. These data for the axially-symmetric flow field compare favorably with the theoretical stream line points and are adequate for accurate water droplet trajectory calculations.

The surface velocity calculated by the potential flow program is compared to test results (reference 22), on Figure 3 for the front section of an NACA 65_1-212, 72 inch chord airfoil. These data show good agreement in the area of water droplet impingement. Disagreement on the aft section is due to intentionally abbreviated program input and could be improved if required.

Figures 4 through 8 verify the accuracy of the water droplet trajectory program for both plane and axially-symmetric flow fields.

The bounding (tangent) water droplet trajectories calculated by the contracted computer program and from Figure 2 of NACA report No. 1215 (reference 2), are superimposed on the cylinder-in-a-uniform-stream potential flow field on Figure 4. These data do not agree but the differences can probably be attributed to calculation differences. The NACA data do not consider the effects of gravity on the droplets and the differential equations of motion were linearized at distances greater than five radii ahead of the cylinder. The contracted digital computer solution considers gravitational effects and is computed from 20 radii ahead of the object. The difference in the bounding trajectory ordinate values at the origin results in the computer program calculating ten percent less impinged water than the NACA data indicate.

The water impingement efficiency comparison on the NACA 65_1-212 airfoil is shown on Figure 5. These data indicate very good agreement in curve shape and near agreement in magnitude between the NACA TN 3839 (reference 17), test values and the computer program results. A slight difference in droplet size distribution from the assumed Langmuir "D" distribution could account for the difference in the curve values.

The computed water impingement results for the 18 inch diameter sphere are shown with NACA test and calculated data on Figure 6. The computer program results for the 14.7 micron volume mean droplet diameter drops compare with the NACA test results as well as the NACA calculated data does, but they are on opposite sides. NACA TN 4092 (reference 12) noted that the accuracy of determining the test droplet size was ± 6% and that this variation would account for the apparent test and calculation results discrepancy. Refining the calculation increment (shorter time interval) to provide the most accurate finite difference solution of the differential equations of motion resulted in a lower water impingement efficiency. It is believed that the earlier NACA calculations were not refined as well as the contracted computer program and therefore were not as accurate.

The computed water impingement intensity on a supersonic engine inlet is compared to the NACA test results (reference 9) on Figures 7 and 8. The centerbody impingement intensity shown on Figure 7 indicates excellent agreement between the test and calculated data. These results show that the computer program provides an accurate determination of the water impingement intensity on an axially-symmetric model when the proper inputs are supplied.

Figure 8 illustrates the more difficult comparison of test and calculated impingement on the supersonic inlet cowl. These data show very good agreement in both magnitude and location. The computed results would definitely provide an adequate design solution.

Machine plotting results are illustrated by Figure 9 for the potential flow program and on Figures 10 and 11 for the water droplet trajectory program.

8.0 SYMBOLS AND REFERENCES

8.1 Symbols

Symbol	Description
a	water droplet radius
c	constant of integration
C_D	Drag Coefficient
D	Drag, lb.
F	Force, lb.
g	Gravitational acceleration, 32.174 ft/sec.-sec.
k	Proportionality constant between air velocity and stream function that is eliminated by arbitrary stream function values.
K	inertia parameter, $2\rho_w a^2 U_\infty / 9\mu_a$, ft.
r	radius, in.
R	Reynolds Number
t	time, sec.
U	free stream velocity, ft/sec.
U'_∞	potential flow free stream velocity
v	local velocity, ft/sec.
$\dot{v}$	normalized local velocity, v/U
x	horizontal distance, inch
y	vertical distance, inch
μ	viscosity lb/ft.-sec.
ρ	density - lb/cu. ft.
ψ	stream function, relative

Subscripts

a	air
i	inside
ic	inside cowl
l	left
N	value at current coordinate location
N + 1	value at next coordinate location
o	outside
oc	outside cowl
ocb	outside centerbody
r	right
rel	relative
w	water or water droplet
x	in horizonatl direction
y	in vertical (or radial) direction
∞	remote

8.2 References

1. Langmuir, Irving & Blodgett, Katherine B.; A Mathematical Investigation of Water Droplet Trajectories - Technical Report No. 5418, Air Material Command, AAF, Feb. 19, 1946. (Contract No. W-33-038-ac-9151 with General Electric Co.).

2. Brun, R. J.; Lewis, W.; Perkins, P. J. & Serafini, J. S.; Impingement of Cloud Droplets on a Cylinder and Procedure for Measuring Liquid-Water Content and Droplet Sizes in Supercooled Clouds by Rotating Multicylinder Method. NACA Report 1215, 1955.

3. Bureau of Naval Weapons Contract Number NOw 65-0273-f dated March 9, 1965.

4. Pope, Alan; Basic Wing and Airfoil Theory. McGraw-Hill Book Company, Inc. 1951.

5. Emmons, Howard W.; Fundamentals of Gas Dynamics. Princeton University Press, 1958.

6. Bergrun, Normal P ; A Method for Numerically Calculating the Area and Distribution of Water Impingement on the Leading Edge of an Airfoil in a Cloud. NACA TN 1397, August 1947.

7. Tribus, Myron; Modern Icing Technology, University of Michigan Project M992-E ARDC, USAF Contract AF 18(600)-51, E.O. No. 462 Br-1, January 1952.

8. Prince, David C.; Kushevics, Guntis & Edwards, Russell N.; Computer Evaluation of Ion Engines. Summary Report prepared by General Electric Company, Evendale, Ohio for NASA. Contract NASA 8-623, March 6, 1963.

9. Gelder, Thomas F.; Droplet Impingement and Ingestion by Supersonic Nose Inlet in Subsonic Tunnel Conditions. NACA TN 4268, May 1958

10. Brun, Rinaldo J.; Cloud-Droplet Ingestion in Engine Inlets with Inlet Velocity Ratios of 1.0 and 0.7. NACA Report 1317, 1957.

11. Torgeson, W. L. & Abramson A. E.; A Study of Heat Requirements for Anti-Icing Radome Shapes with Dry and Wet Surfaces. WADC Technical Report 53-284, September 1953.

12. Lewis, James P. & Ruggeri, Robert S. Experimental Droplet Impingement on Four Bodies of Revolution. NACA TN 4092, December 1957.

13. Brun, Rinaldo J. & Dorsch, Robert G.; Impingement of Water Droplets on an Ellipsoid with Fineness Ratio 10 in Axisymmetric Flow. NACA TN 3147, May 1954

14. Dorsch, Robert G.; Brun, Rinaldo J. & Gregg, John L.; Impingement of Water Droplets on an Ellipsoid with Fineness Ratio 5 in Axisymmetric Flow. NACA TN 3099, March 1954.

15. Lewis, William & Brun, Rinaldo J.; Impingement of Water Droplets on a Rectangular Half Body in a Two-Dimensional Incompressible Flow Field. NACA TN 3658, February 1956.

16. von Glahn, Uwe H.; Use of Truncated Flapped Airfoils for Impingement and Icing Tests of Full-Scale Leading Edge Section. NACA RM E56E11, July 1956.

17. Gelder, Thomas F.; Smyers, William H. Jr. & von Glahn, Uwe; Experimental Droplet Impingement on Several Two-Dimensional Airfoils with Thickness Ratios of 6 to 16 Percent. NACA TN 3839, December 1956.

18. Brun, Rinaldo J.; Gallagher, Helen M. & Vogt, Dorothea; Impingement of Water Droplets on NACA 65A004, Airfoil and Effect of Change in Airfoil Thickness from 12 to 4 percent at 4° Angle of Attack. NACA TN 3047, November 1953.

19. Brun, Rinaldo J.; Gallagher, Helen M. & Vogt, Dorothea E.; Impingement of Water Droplets on NACA 65A004, Airfoil at 8° Angle of Attack. NACA TN 3155, July 1954.

20. Brun, Rinaldo J. & Vogt, Dorothea E.; Impingement of Water Droplets on NACA 65A004 Airfoil at 0° Angle of Attack. NACA TN 3586, November 1955.

21. Gray, Vernon H.; Correlations Among Ice Measurements, Impingement Rates, Icing Conditions and Drag Coefficients for Unswept NACA-65A 004 Airfoil. NACA TN 4151, February 1958.

22. Brun, Rinaldo J.; Gallagher, Helen M. & Vogt, Dorothea E.; Impingement of Water Droplets on NACA 65_1 - 208 and 65_1 - 212 Airfoils at 4° Angle of Attack. NACA TN 2952, May 1953.

23. Brun, Rinaldo J.; Serafini, John S. & Moshos, George J.; Impingement of Water Droplets on an NACA 65_1 - 212 Airfoil at an Angle of Attack at 4°. NACA RME52B12, September 1952.

9.0 PLATES

Figure 1 Potential Flow Field for Cylinder

Figure 2 Potential Flow Field for Sphere

Figure 3 NACA 65_1-212 Airfoil Surface Velocity, 4° Angle of Attack

Figure 4 Water Droplet Trajectory Comparison on Cylinder

Figure 5 NACA 65_1-212 Airfoil Water Impingement, 4° Angle of Attack

Figure 6 NACA TN 4092 Sphere Water Impingement

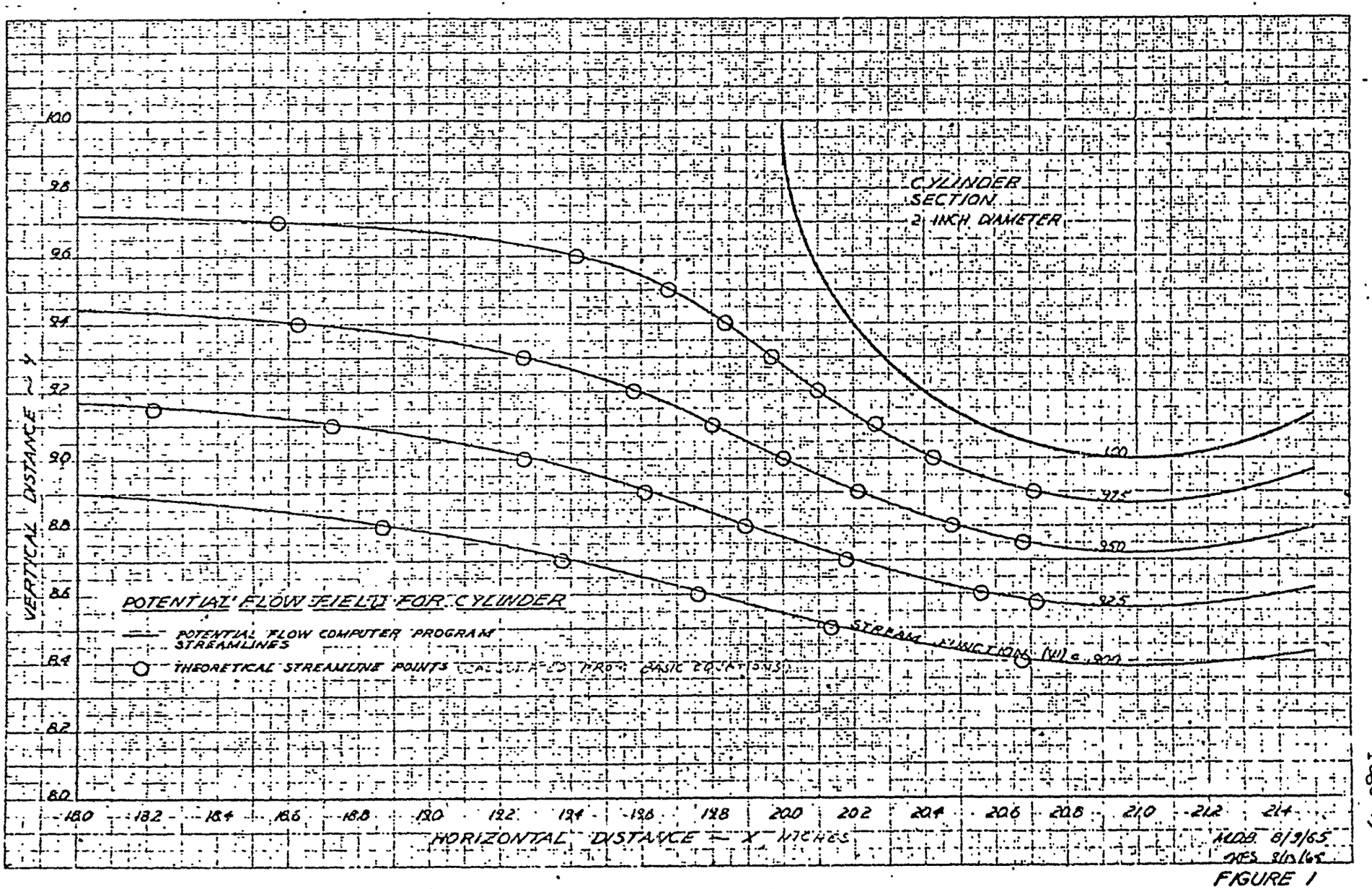
Figure 7 Supersonic Nose Inlet Centerbody Water Impingement Comparison

Figure 8 Supersonic Nose Inlet Cowl Water Impingement Comparison

Figure 9 Potential Flow Field for 18 inch Diameter Sphere

Figure 10 Water Droplet Trajectories - 5.77 Micron Drop

Figure 11 Water Droplet Trajectories - 18.6 Micron Drop



KDB 8/9/65
JES 8/10/65

FIGURE 1

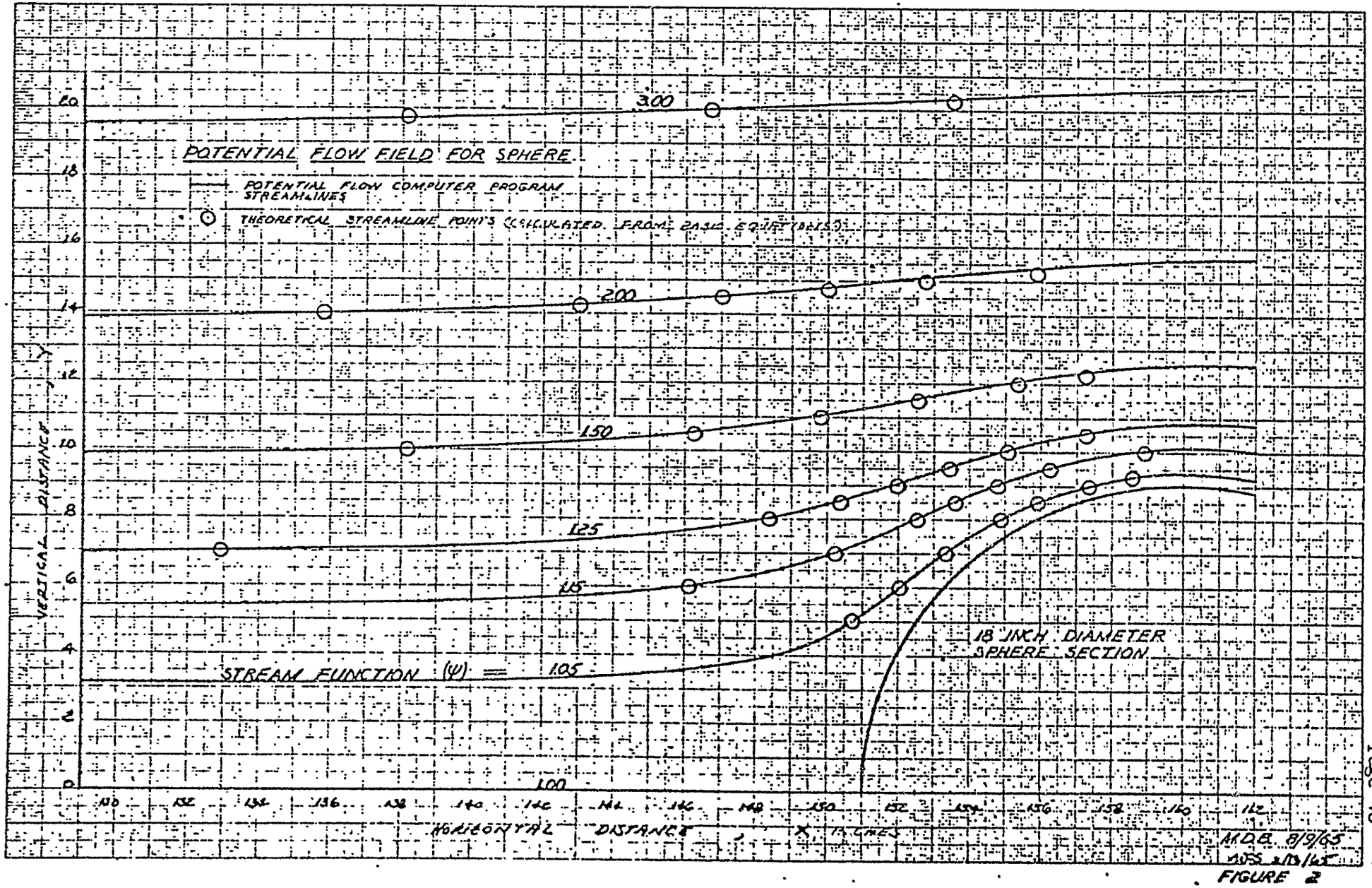
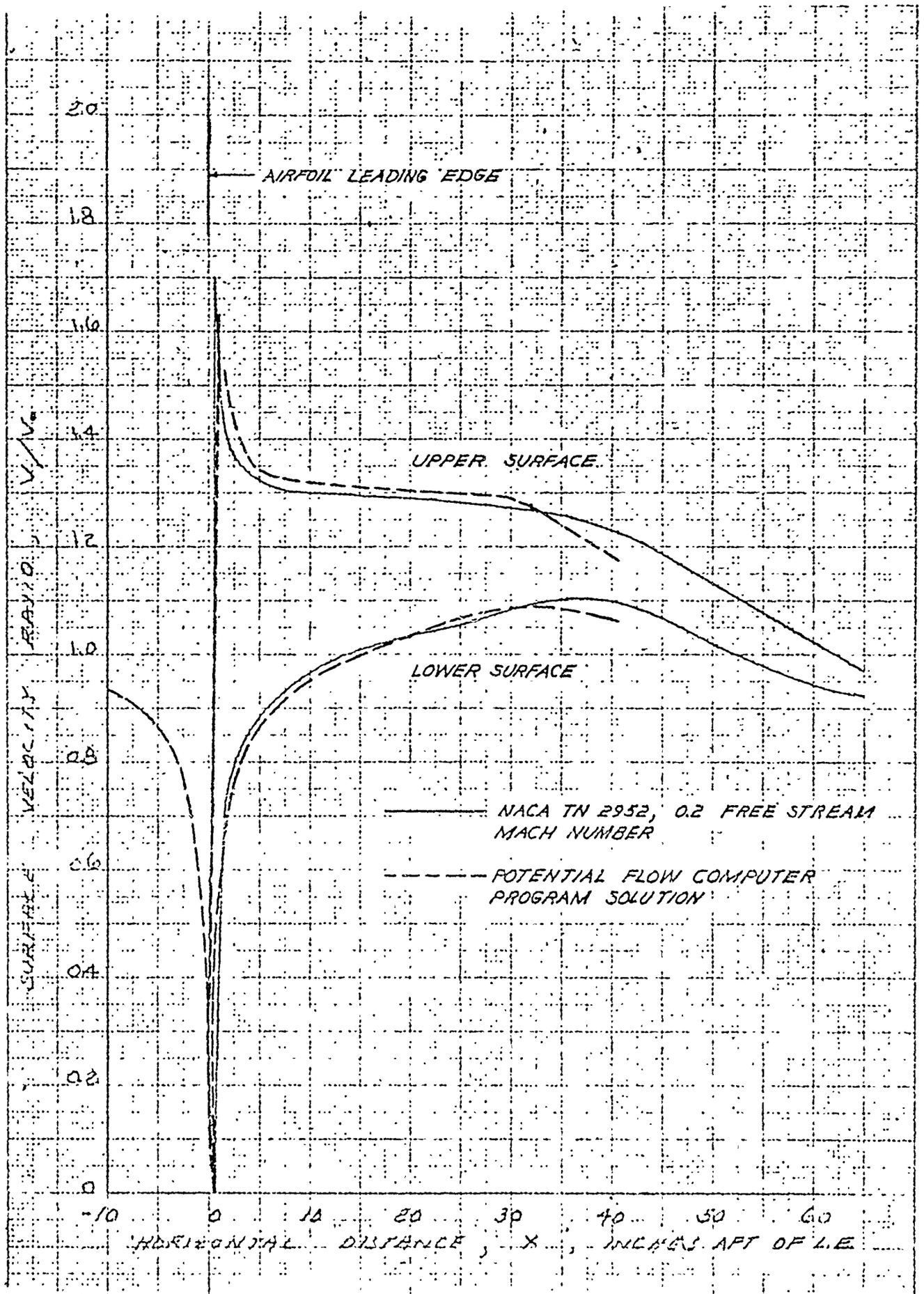

POTENTIAL FLOW FIELD FOR SPHERE
POTENTIAL FLOW COMPUTER PROGRAM STREAMLINES
THEORETICAL STREAMLINE POINTS (CALCULATED FROM BASIC EQUATIONS)
300
200
150
125
115
105
100
STREAM FUNCTION (ψ) =
18 INCH DIAMETER SPHERE SECTION
VERTICAL DISTANCE, Y
HORIZONTAL DISTANCE, X, INCHES
0
2
4
6
8
10
12
14
16
18
20
130
132
134
136
138
140
142
144
146
148
150
152
154
156
158
160
162
M.D.B. 8/9/65

FIGURE 2



CALC	M.D.B.	8/9/65	REVISED	DATE
CHECK	TFS	7/13/65		
APPD				
APPD				

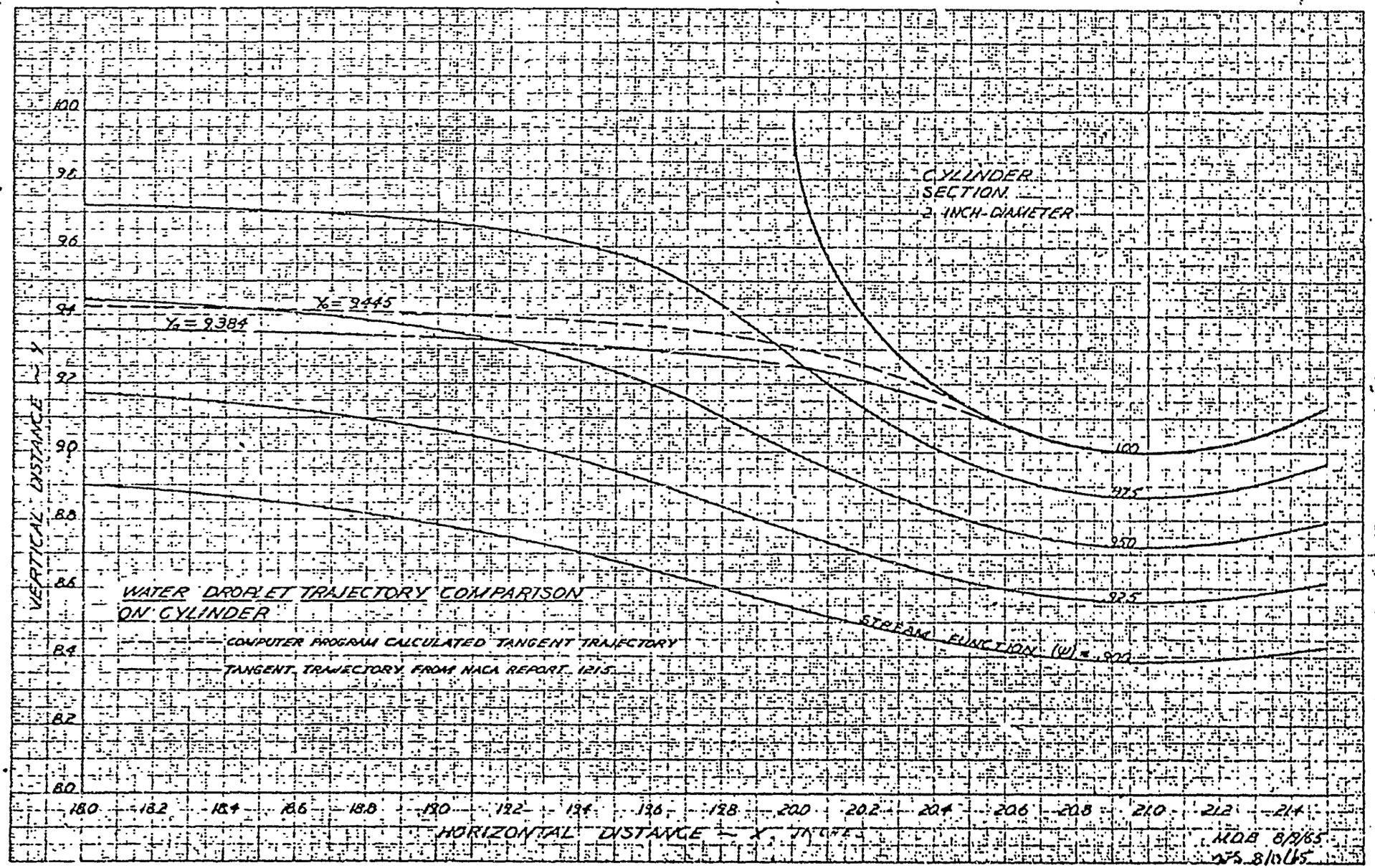
NACA 65,-212 AIRFOIL SURFACE VELOCITY, 4° ANGLE OF ATTACK

REV LTR:

NO. FIGURE 3

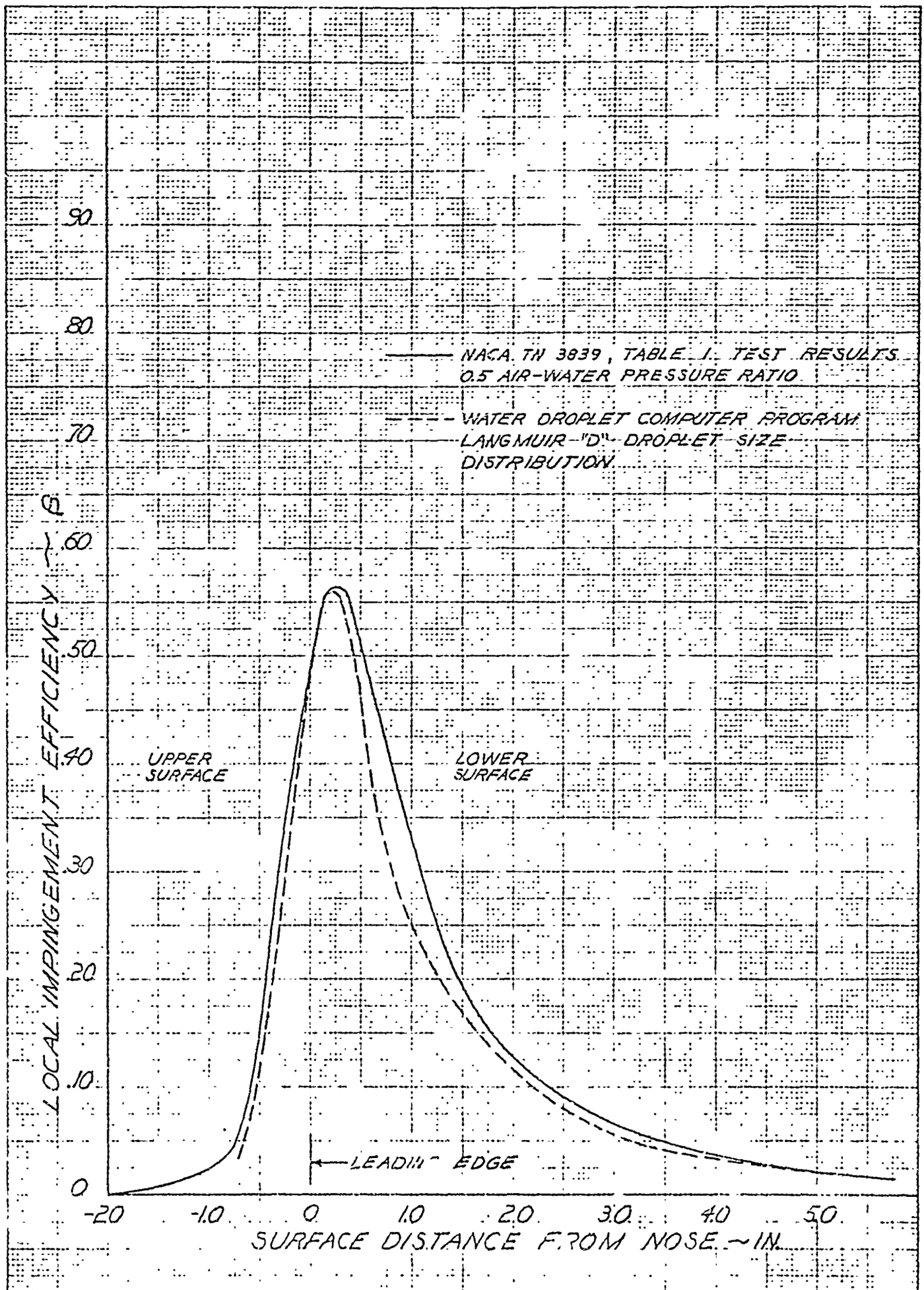
SECT

PAGE 47



MDB 8/9/85
WS 8/10/85

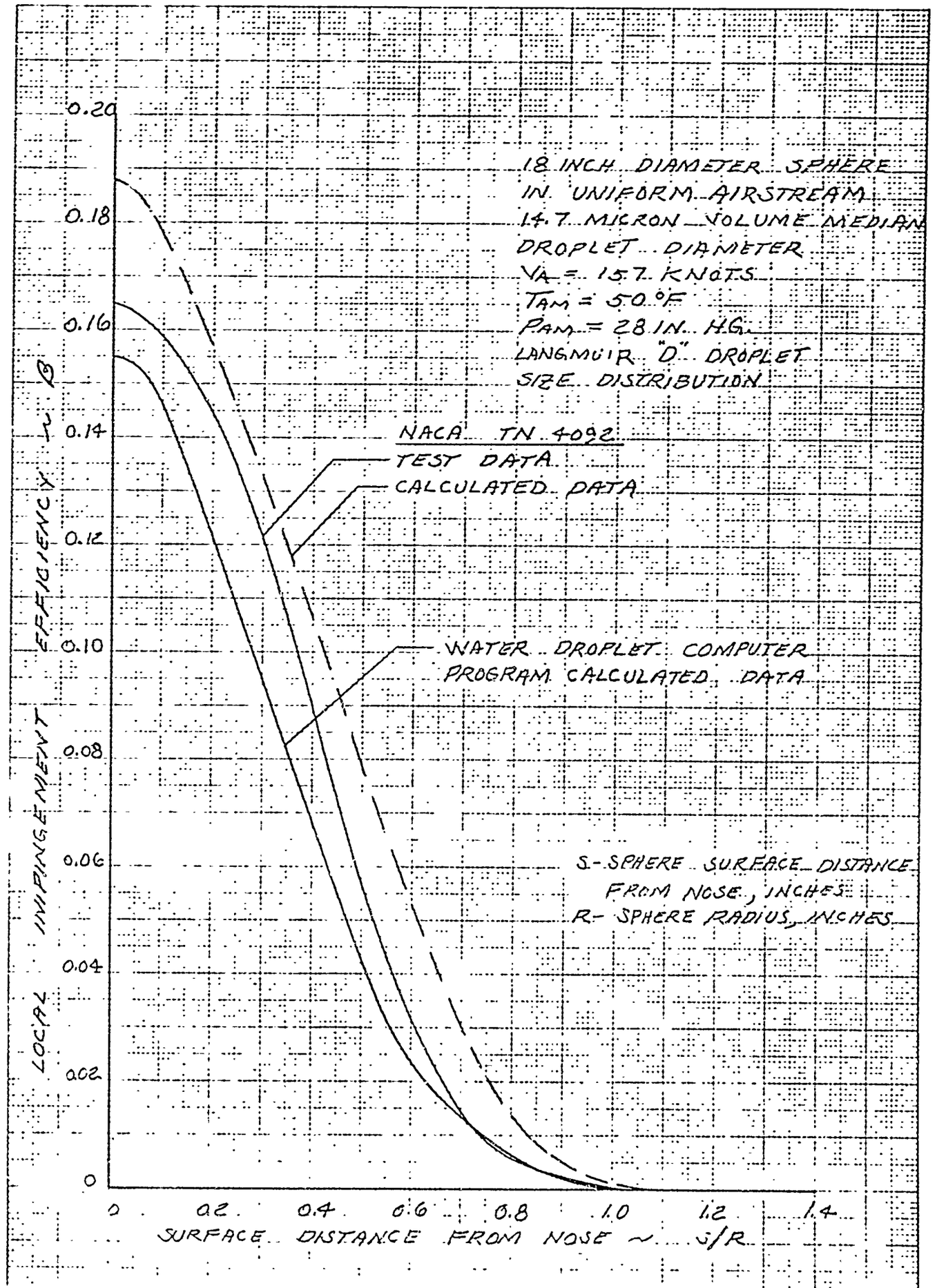
FIGURE 4



			REVISED	DATE	NACA 65,-212 AIRFOIL WATER IMPINGEMENT, 4° ANGLE OF ATTACK
CALC	MDB	8/2/			
CHECK	MES	8/13/05			
APPD					
APPD					

BOEING NO. FIGURE 5

REV I TR: SECT PAGE 49

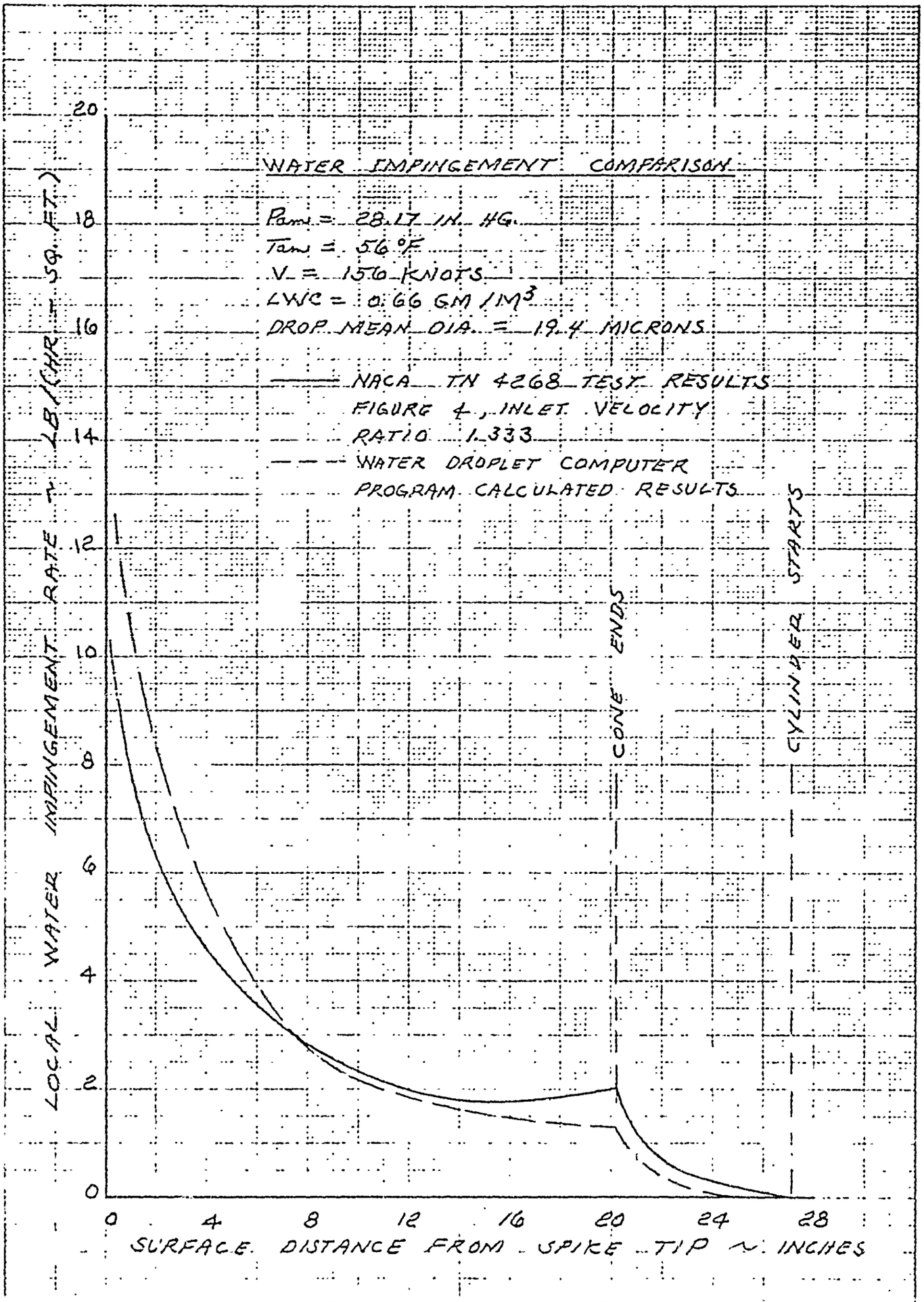


CALC	BFS	9/13/55	REVISED	DATE	NACA TN 4092 SPHERE WATER IMPINGEMENT
CHECK					
APPD					
APPD					

REV LTR

BOEING NO. FIGURE 6

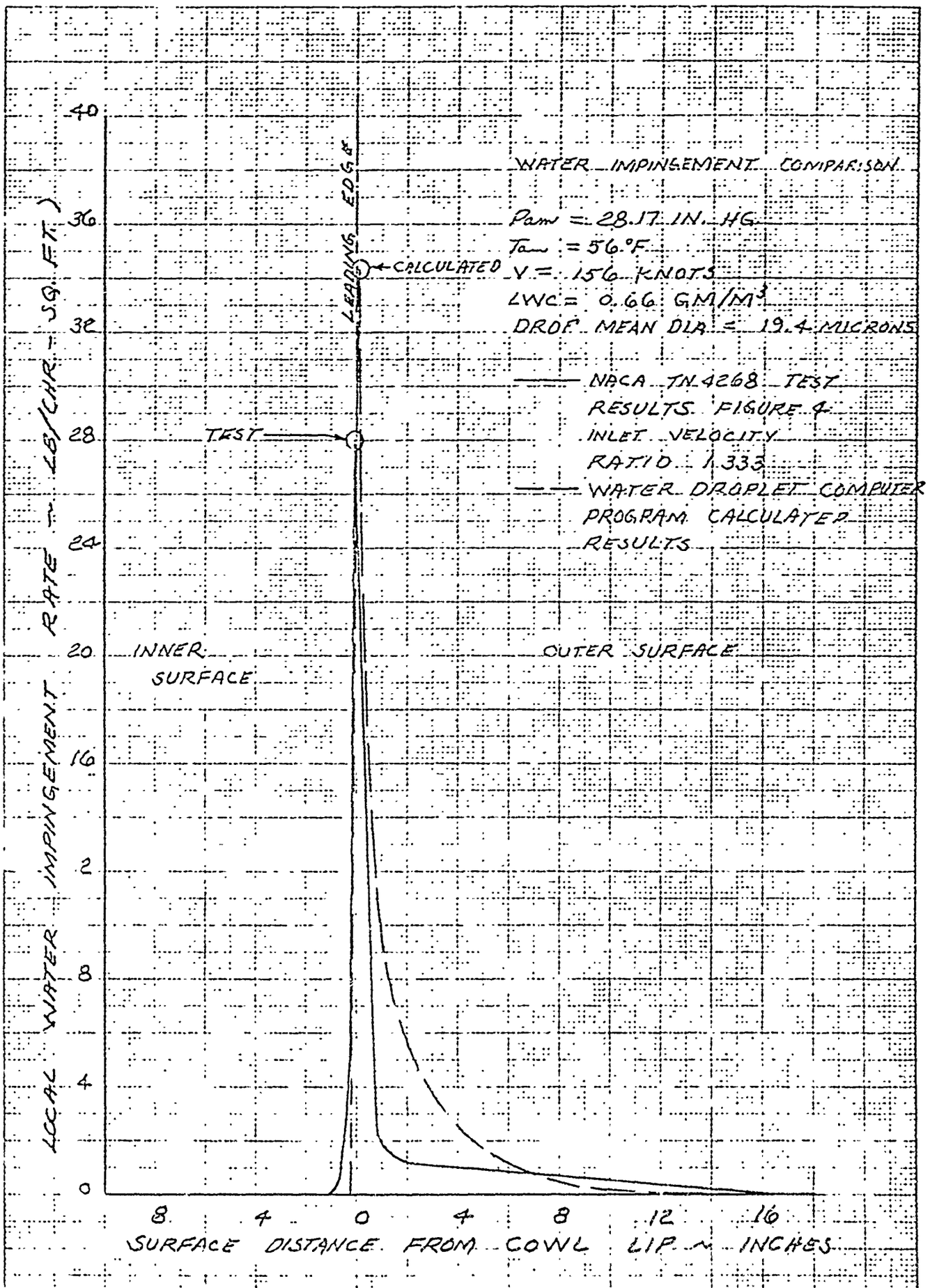
SECT PAGE 50



CALC	WFS		REVISED	DATE	SUPERSONIC NOSE INLET CENTERBODY
CHECK					
APPD					
APPD					

BOEING NO. FIGURE 7

REV LTR:



CALC			REVISED	DATE	SUPERSONIC NOSE INLET COWL
CHECK					
APPD					
APPD					

BOEING NO. FIGURE 8

SECT PAGE

REV LTR:

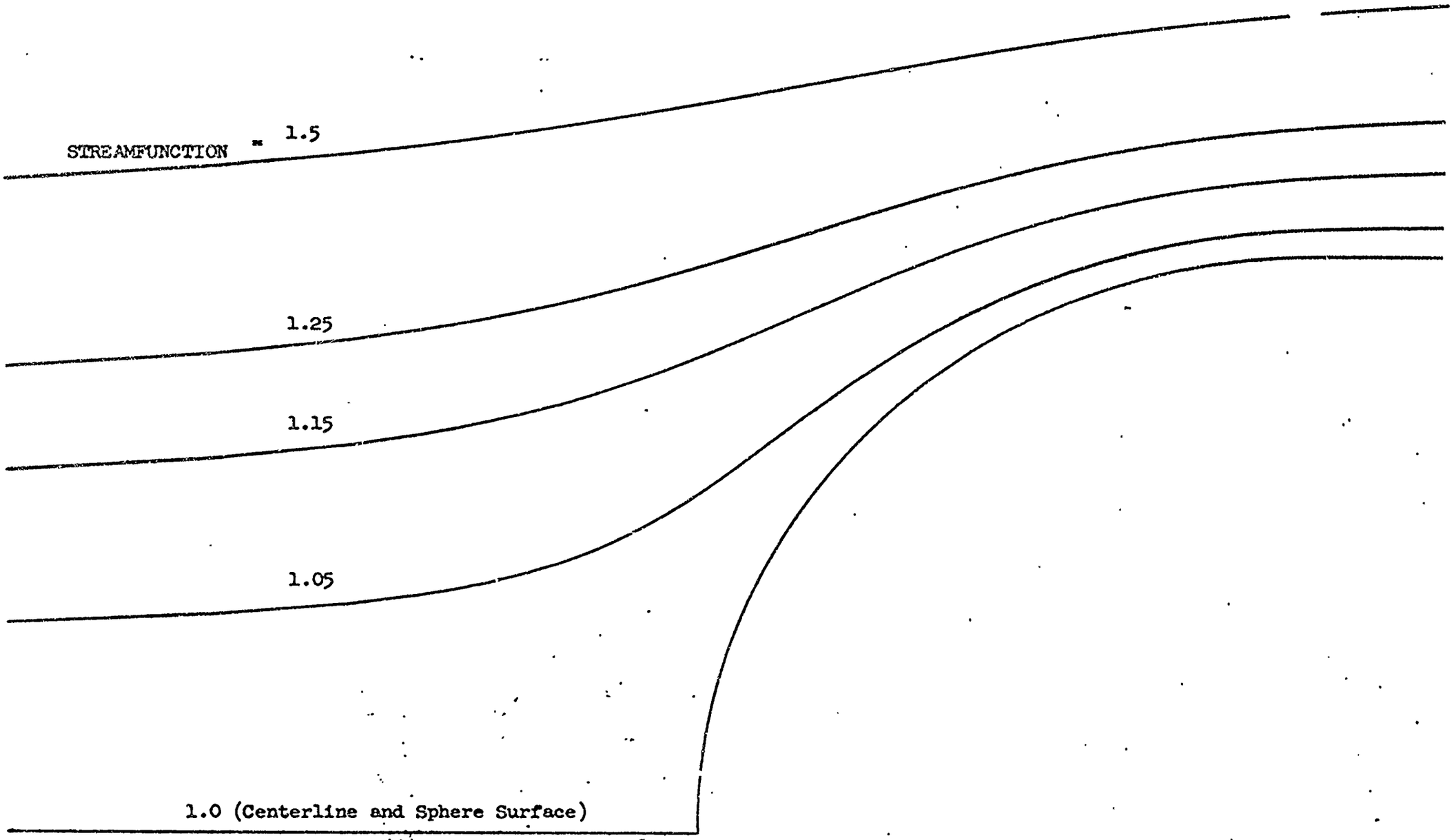


FIGURE 9. POTENTIAL FLOW FIELD FOR 18 INCH DIAMETER SPHERE

WATER DROPLET TRAJECTORIES

Ambient Pressure	28 in. Hg.
Ambient Temperature	50°F
Remote Air Velocity	157 Knots

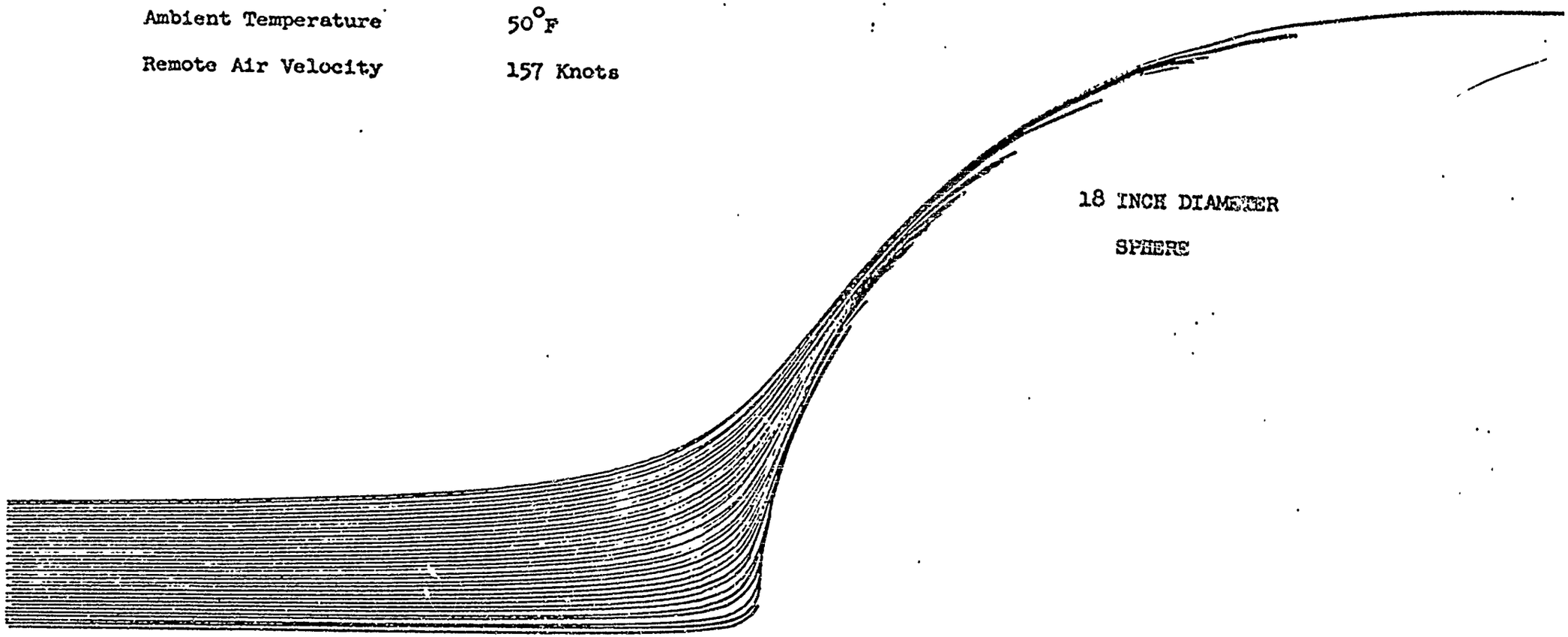


FIGURE 10. SPHERE 5.77 MICRON DROP

WATER DROPLET TRAJECTORIES

Ambient Pressure	28 in Hg.
Ambient Temperature	50°F
Remote Air Velocity	157 Knots

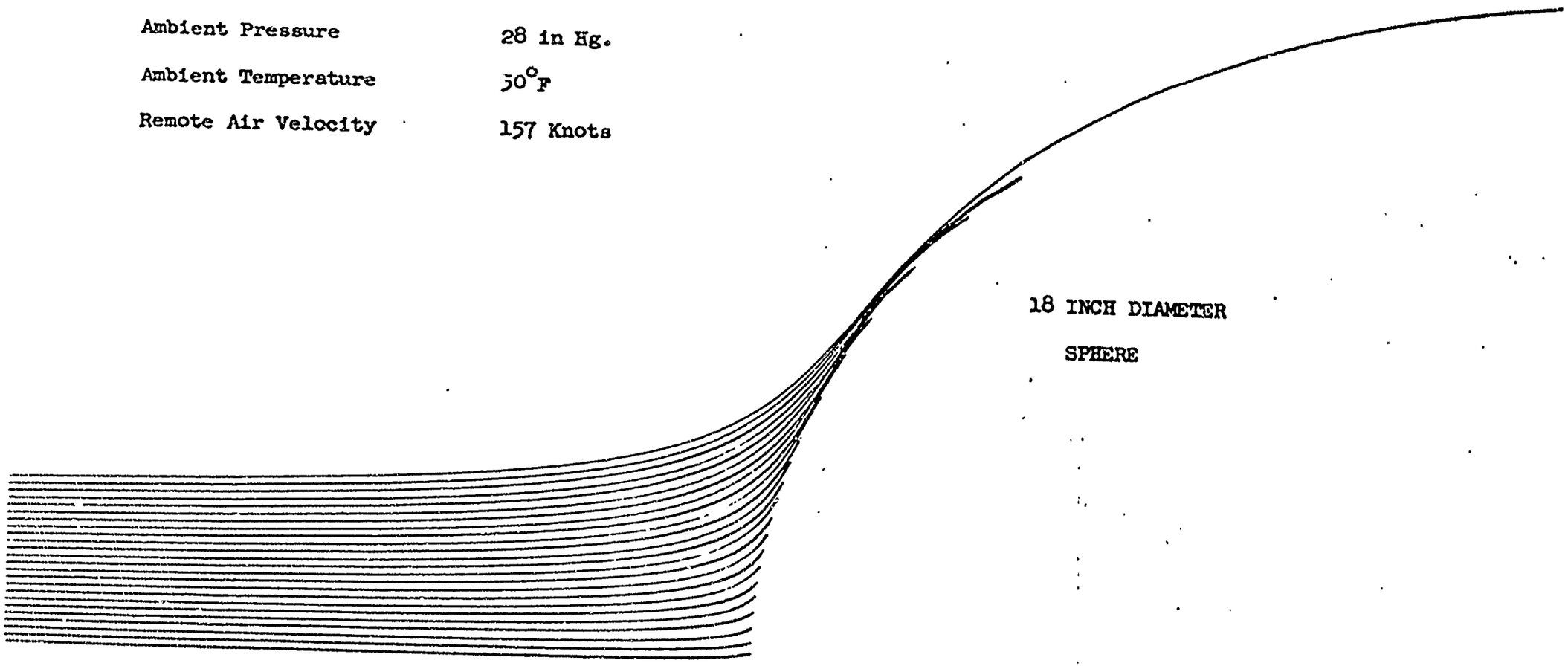


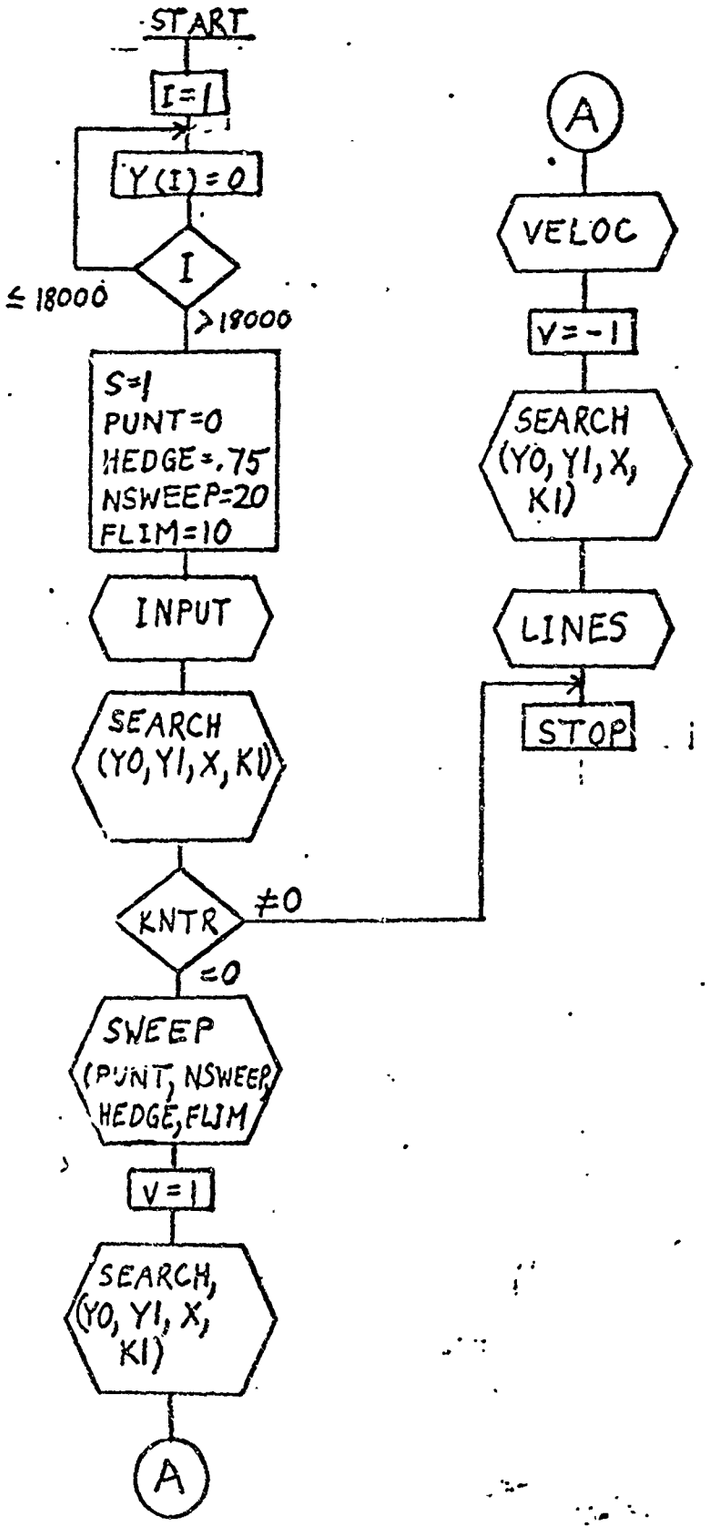
FIGURE 11.

SPHERE 18.6 MICRON DROP

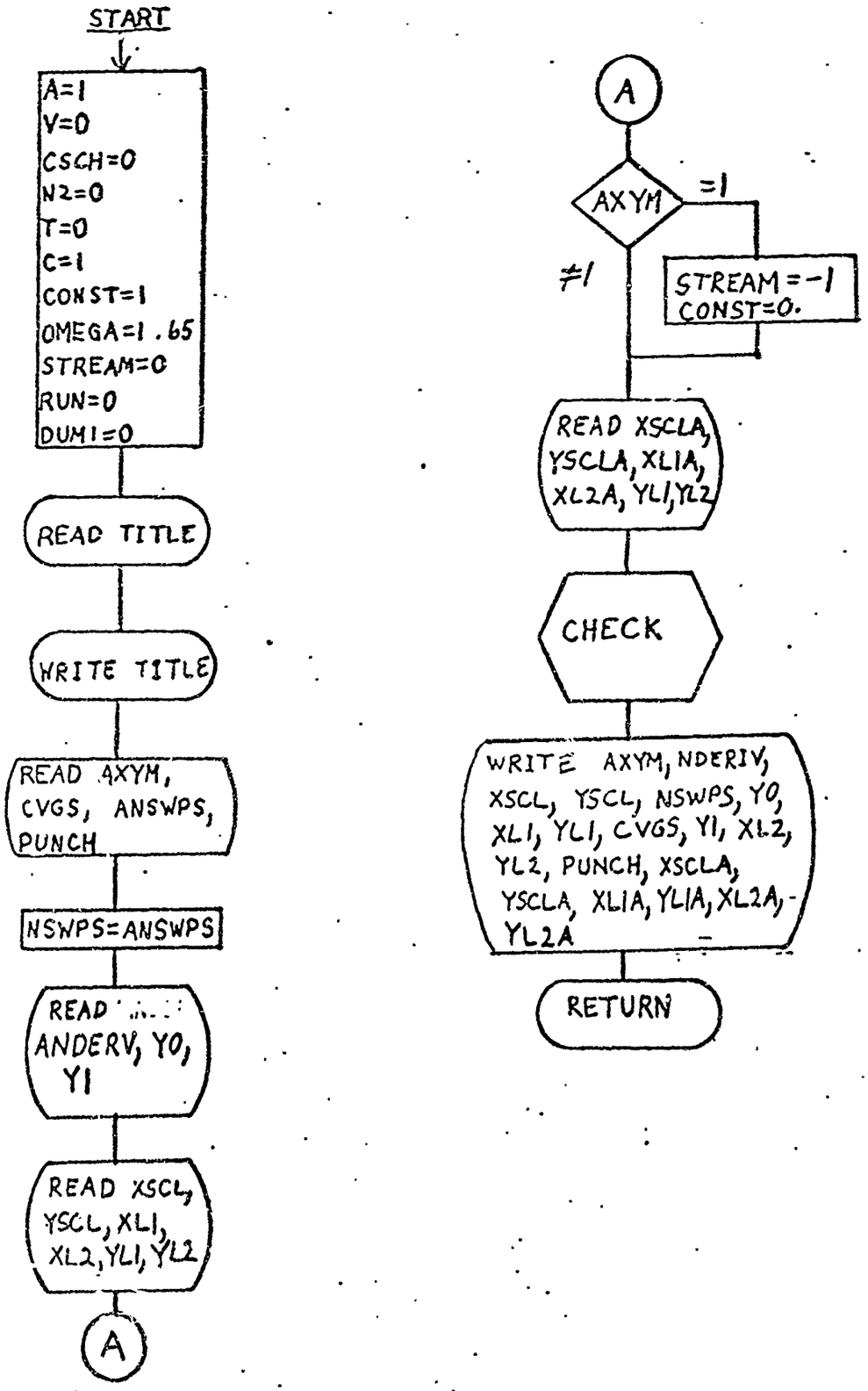
10.0 APPENDICES

10.1 Appendix 1 - Potential Flow Computer Program Flow Chart

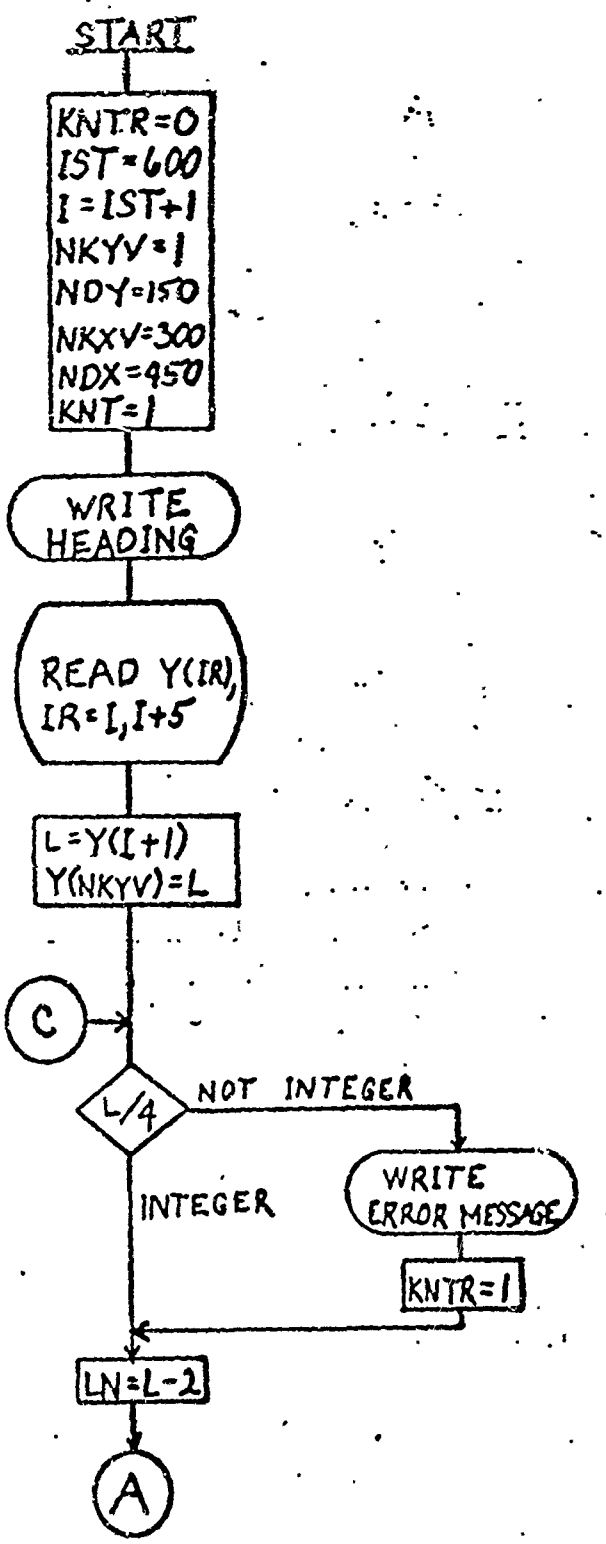
POTENTIAL FLOW PROGRAM

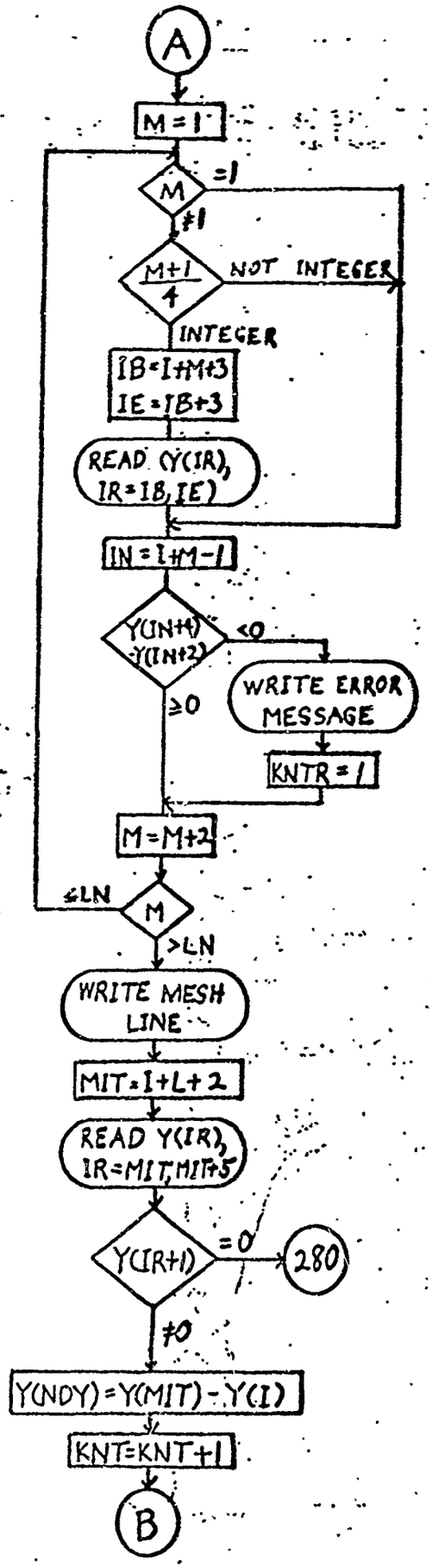


SUBROUTINE INPUT

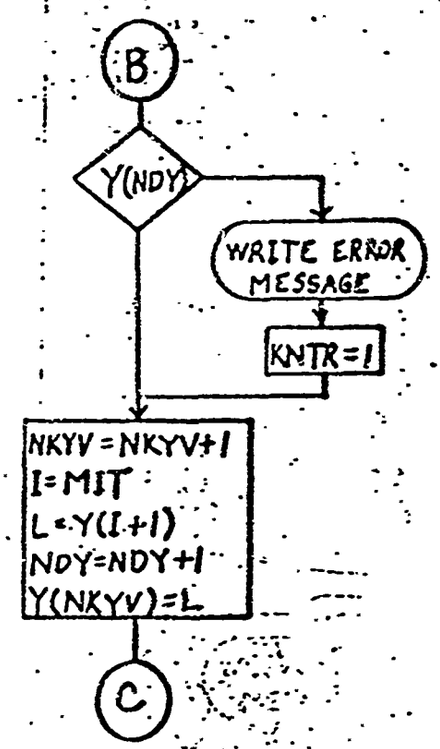


SUBROUTINE CHECK

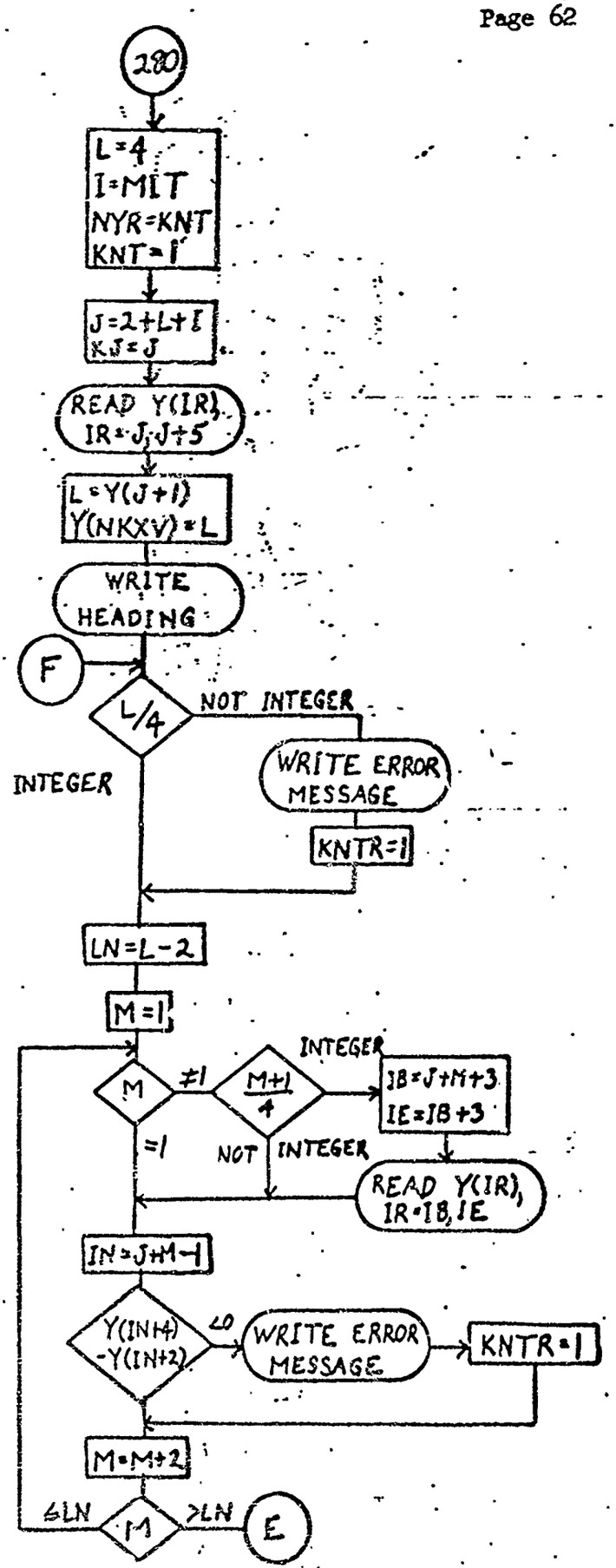



A
M=1
M
=1
≠1
(M+1)/4
NOT INTEGER
INTEGER
IB=I+M+3
IE=IB+3
READ (Y(IR), IR=IB, IE)
IN=I+M-1
Y(IN+1) - Y(IN+2)
<0
≥0
WRITE ERROR MESSAGE
KNTR=1
M=M+2
≤LN
M
>LN
WRITE MESH LINE
MIT=I+L+2
READ Y(IR), IR=MIT, MIT+5
Y(IR+1)
=0
280
≠0
Y(NDY)=Y(MIT)-Y(I)
KNT=KNT+1
B

CHECK 2

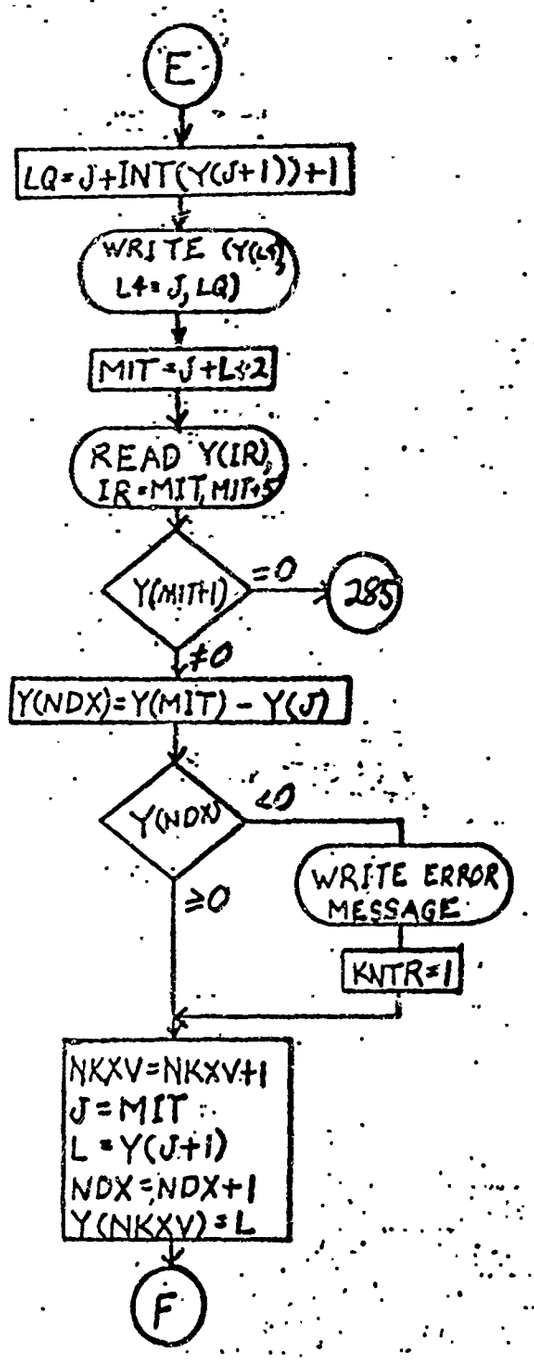

B
Y(NDY)
WRITE ERROR MESSAGE
KNTR=1
NKYV=NKYV+1
I=MIT
L=Y(I+1)
NDY=NDY+1
Y(NKYV)=L
C

CHECK 3



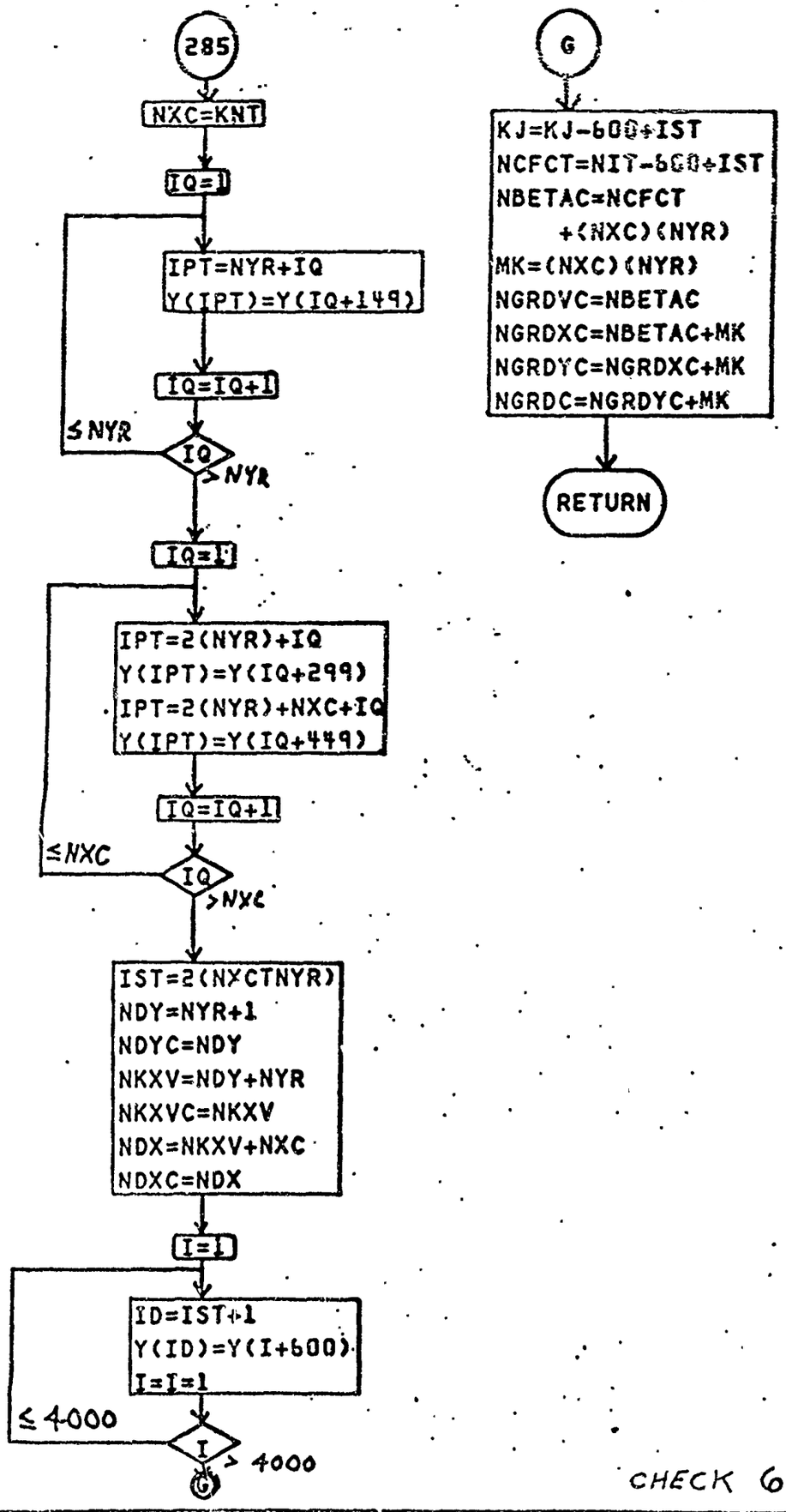
CHECK 4

REVISED



REVISED

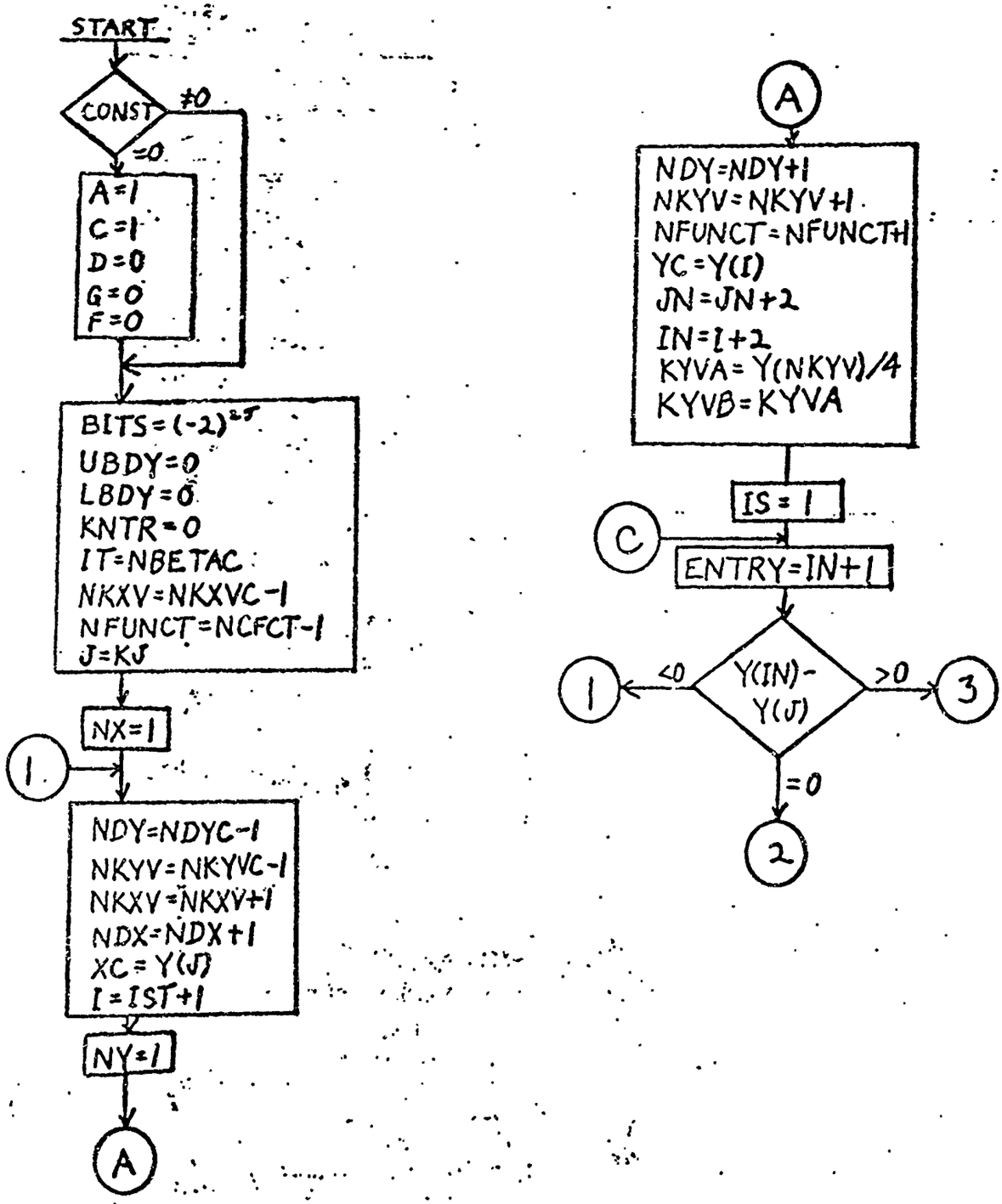
CHECK 5

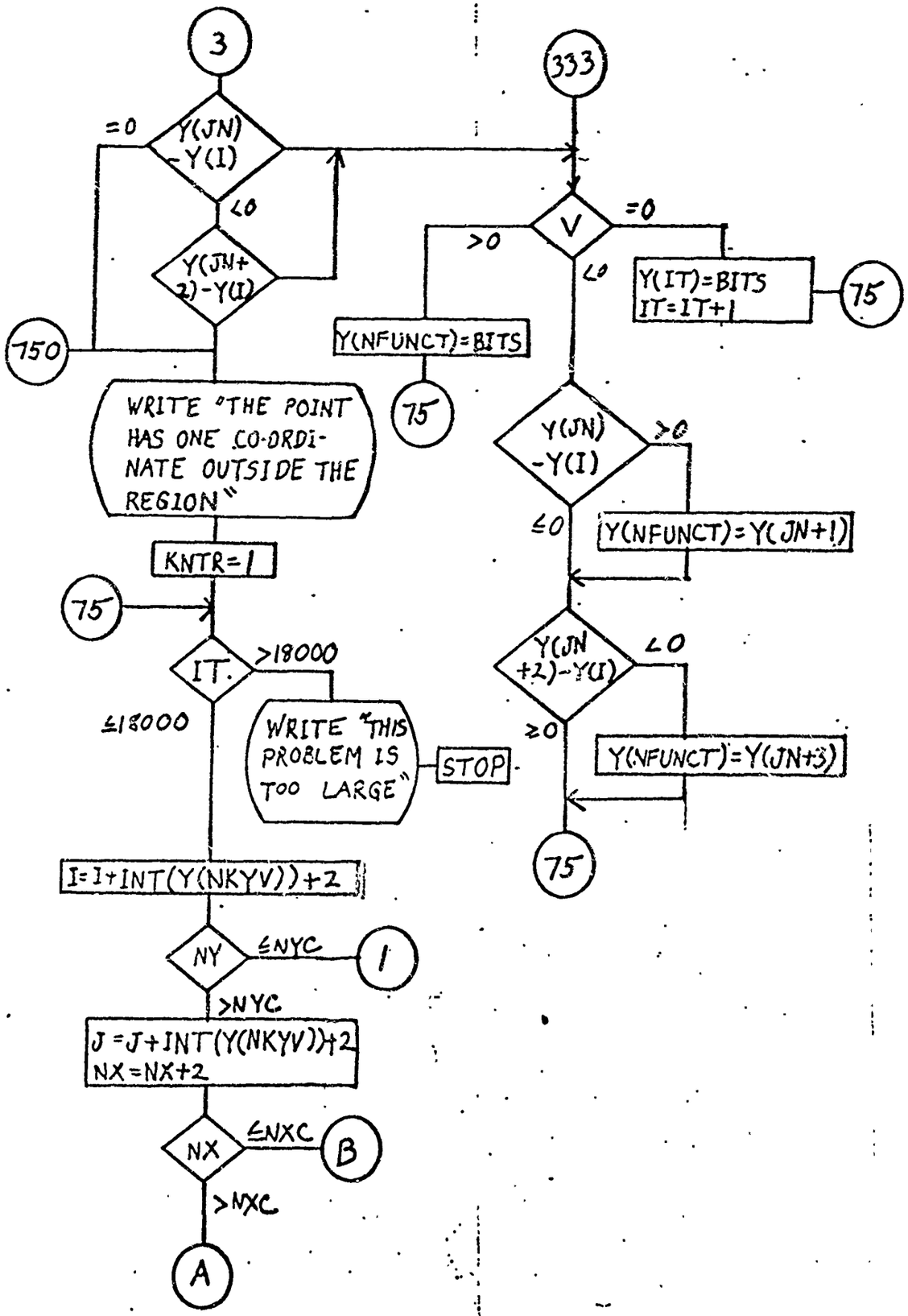


CHECK 6

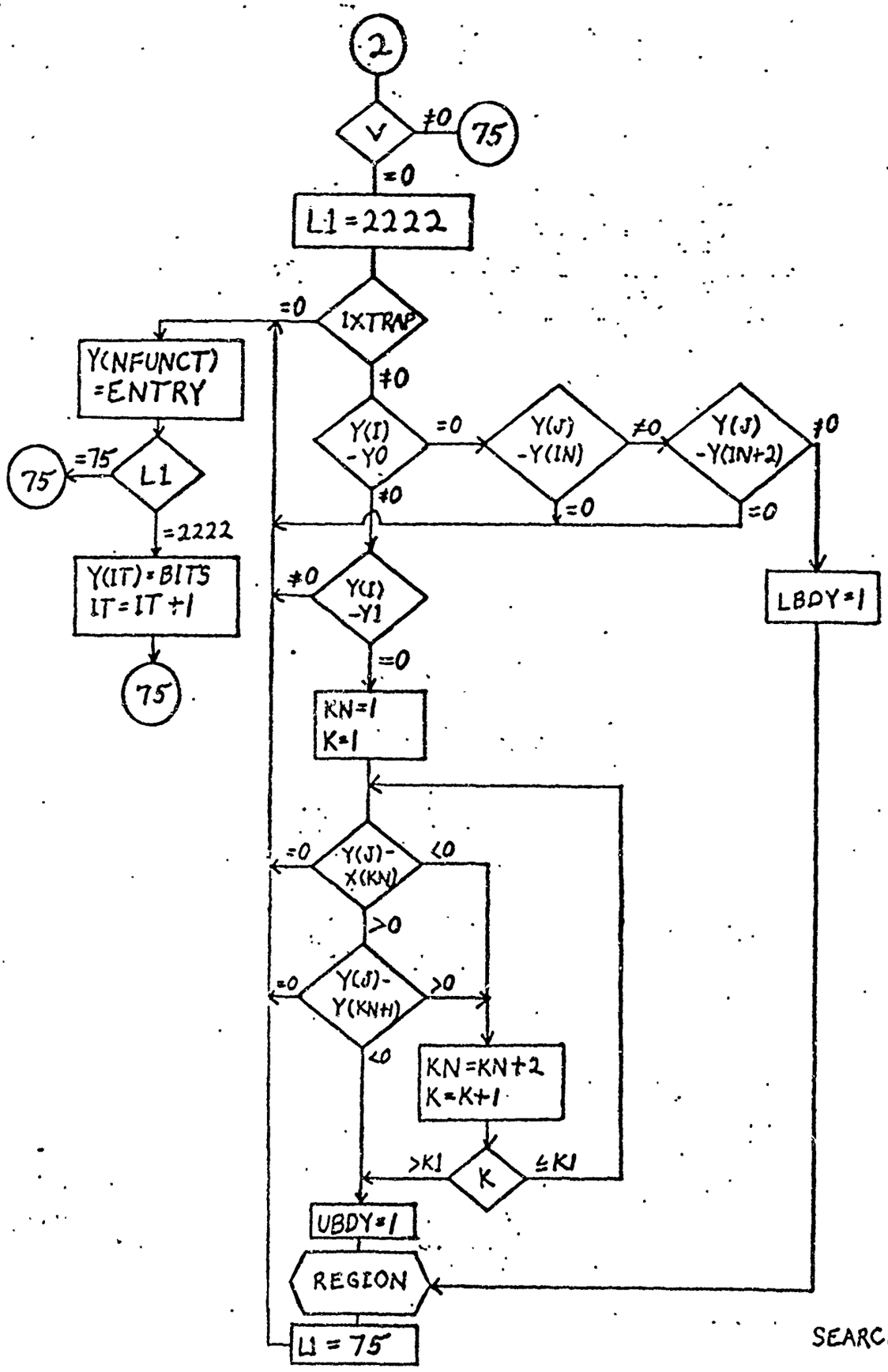
REVISED

SUBROUTINE SEARCH

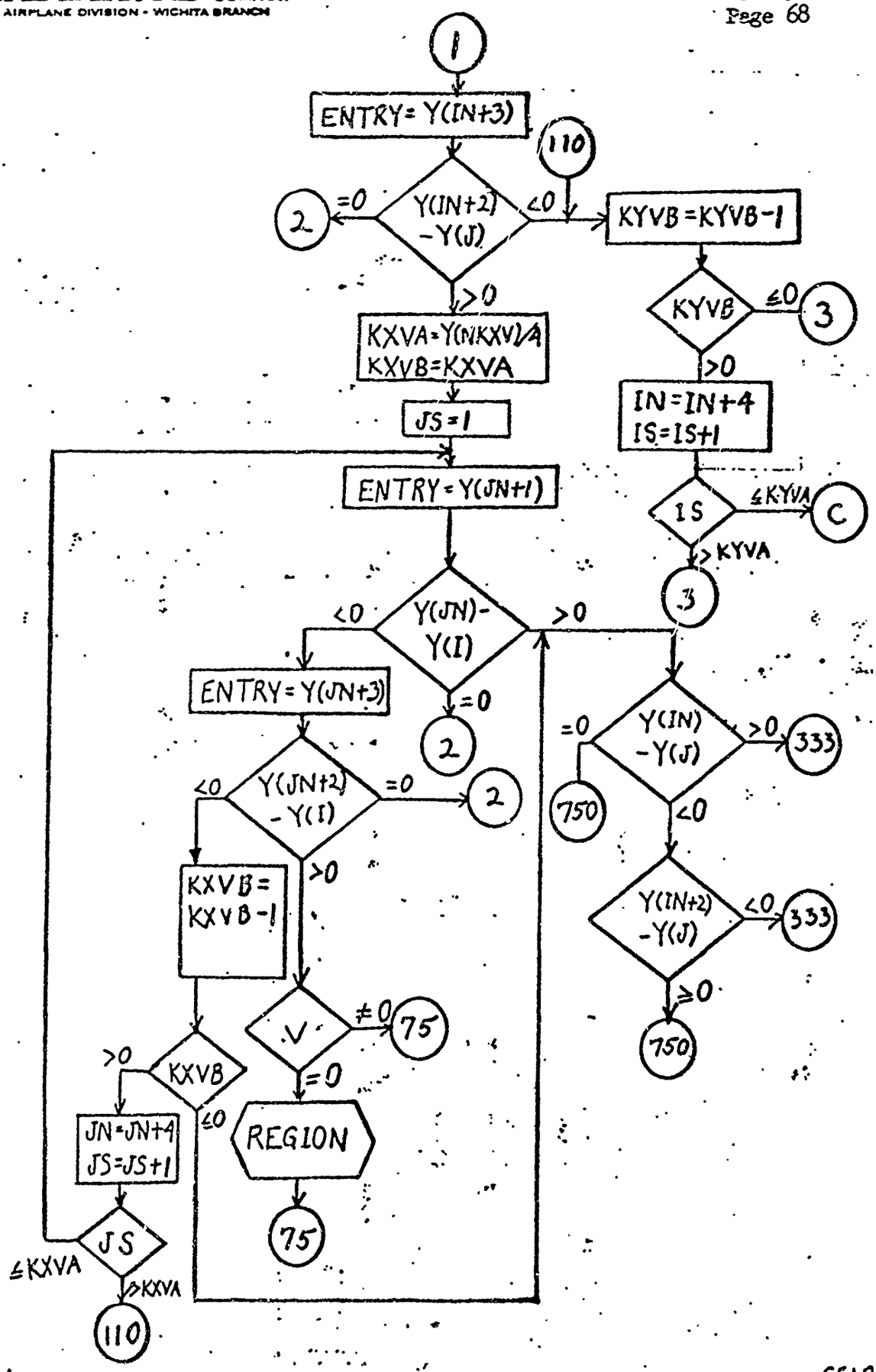



3
Y(JN) -Y(I)
=0
<0
Y(JN+2)-Y(I)
750
WRITE "THE POINT HAS ONE CO-ORDINATE OUTSIDE THE REGION"
KNTR=1
75
IT.
>18000
≤18000
WRITE "THIS PROBLEM IS TOO LARGE"
STOP
I=I+INT(Y(NKYV))+2
NY
≤NYC
1
>NYC
J=J+INT(Y(NKYV))+2
NX=NX+2
NX
≤NXC
B
>NXC
A
333
V
=0
>0
<0
Y(IT)=BITS
IT=IT+1
75
Y(NFUNCT)=BITS
75
Y(JN) -Y(I)
>0
≤0
Y(NFUNCT)=Y(JN+1)
Y(JN+2)-Y(I)
<0
≥0
Y(NFUNCT)=Y(JN+3)
75

SEARCH 2


2
V
≠0
75
=0
L1=2222
IXTRAP
=0
Y(NFUNCT)
=ENTRY
L1
=75
75
=2222
Y(IT)=BITS
IT=IT+1
75
≠0
Y(I)
-Y0
=0
Y(J)
-Y(IN)
≠0
Y(J)
-Y(IN+2)
≠0
=0
=0
≠0
LBDY=1
≠0
Y(I)
-Y1
=0
KN=1
K=1
=0
Y(J)-
X(KN)
<0
>0
=0
Y(J)-
Y(KN+1)
>0
<0
KN=KN+2
K=K+1
>K1
K
≤K1
UBDY=1
REGION
L1=75

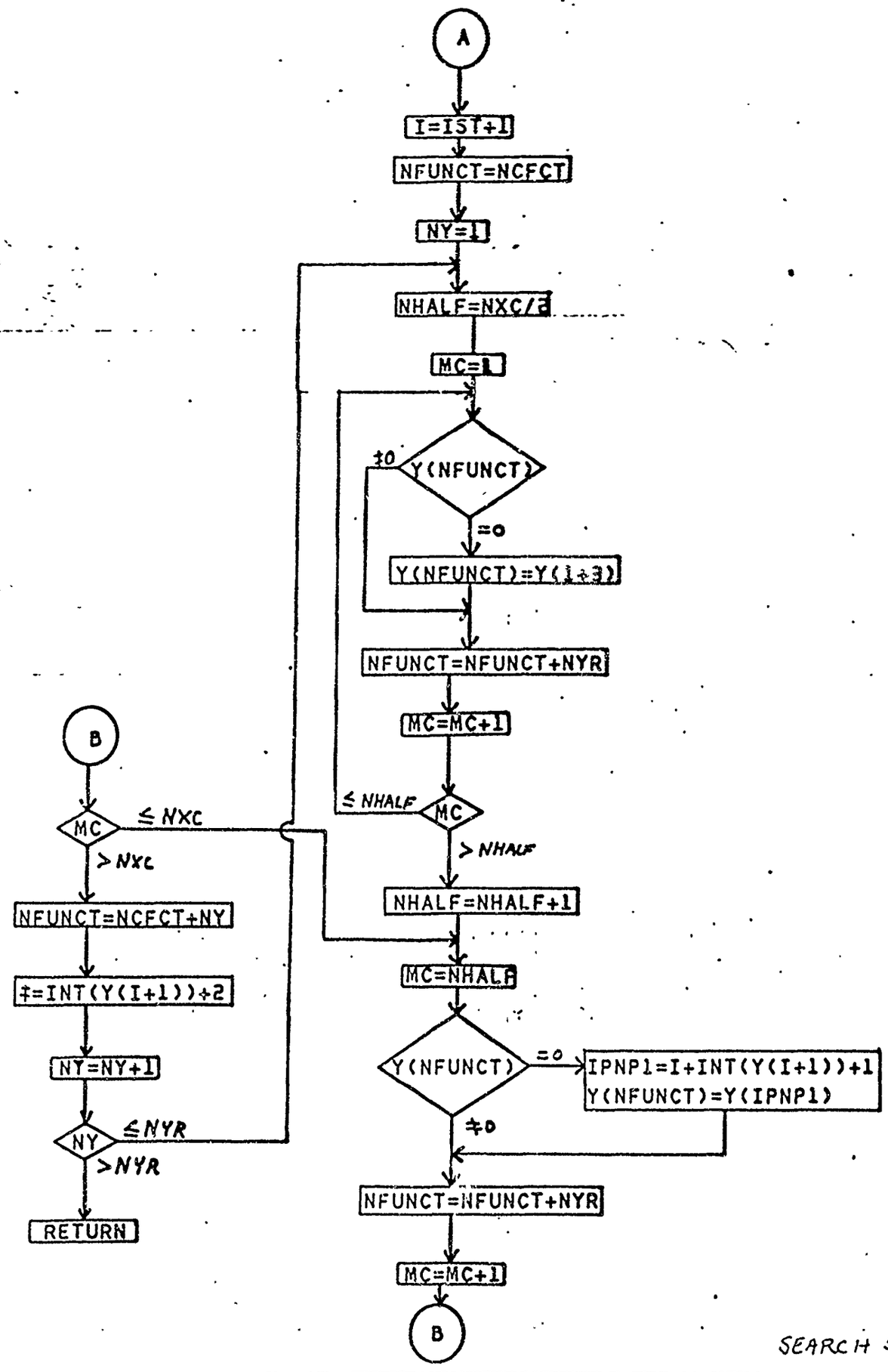
SEARCH 3



SEARCH 4

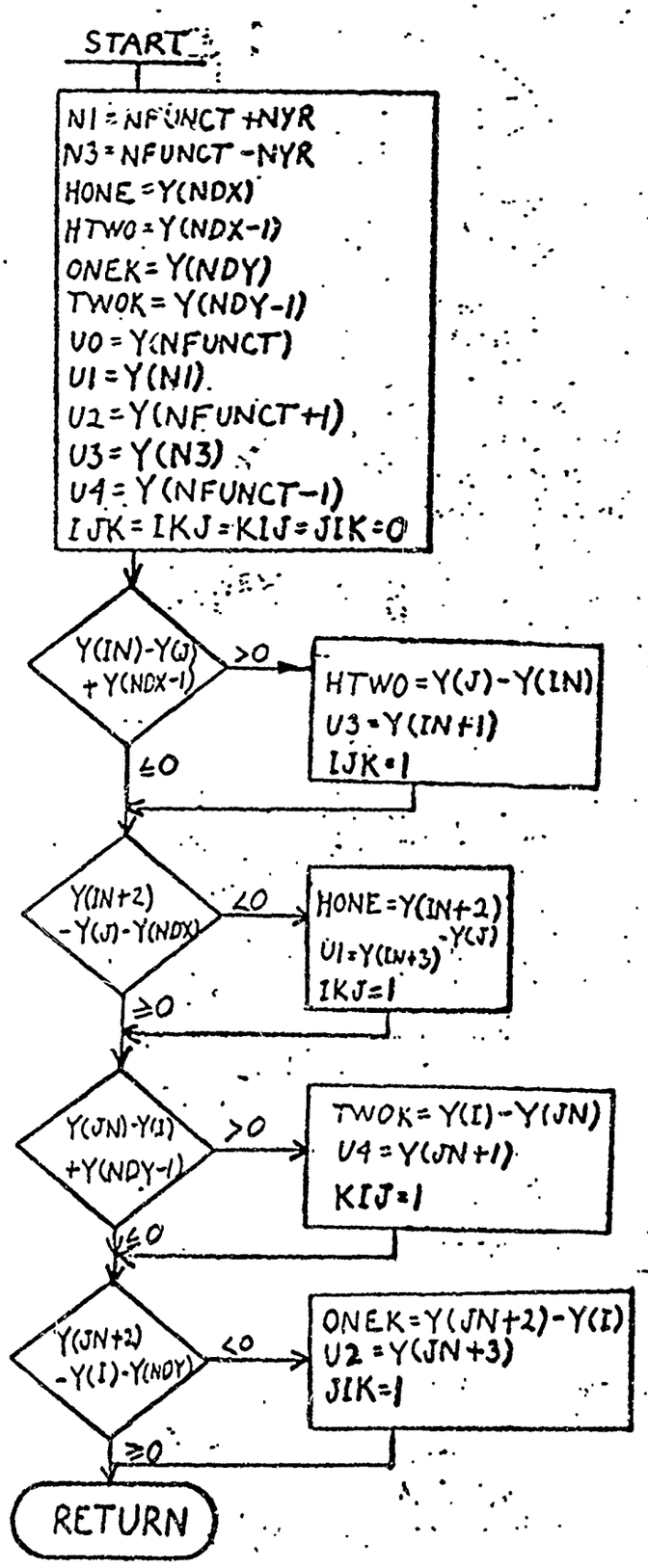
REVISED

E 1666 R8

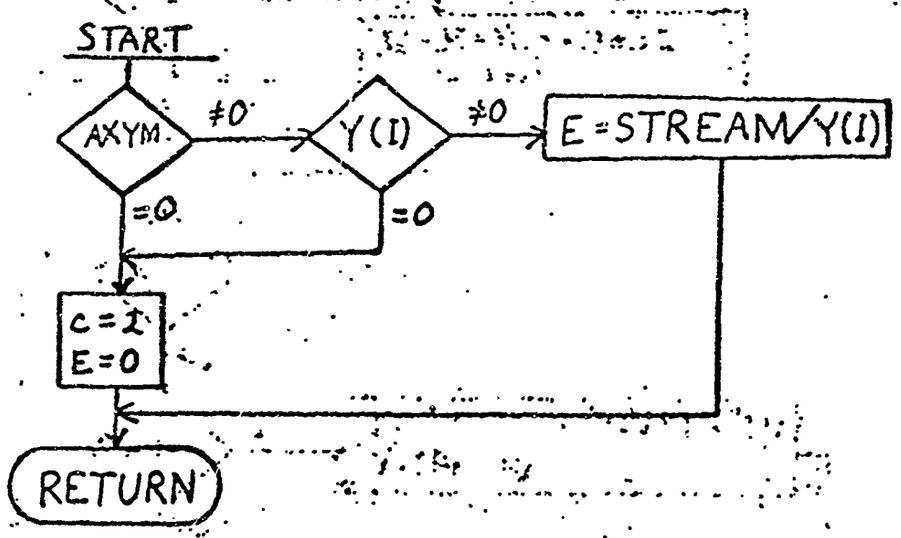

A
I=IST+1
NFUNCT=NCFCT
NY=1
NHALF=NXC/2
MC=1
Y(NFUNCT)
≠0
=0
Y(NFUNCT)=Y(I+3)
NFUNCT=NFUNCT+NYR
MC=MC+1
MC
≤ NHALF
> NHALF
NHALF=NHALF+1
MC=NHALF
Y(NFUNCT)
=0
IPNP1=I+INT(Y(I+1))+1
Y(NFUNCT)=Y(IPNP1)
≠0
NFUNCT=NFUNCT+NYR
MC=MC+1
B
B
MC
≤ NXC
> NXC
NFUNCT=NCFCT+NY
I=INT(Y(I+1))+2
NY=NY+1
NY
≤ NYR
> NYR
RETURN

SEARCH 5

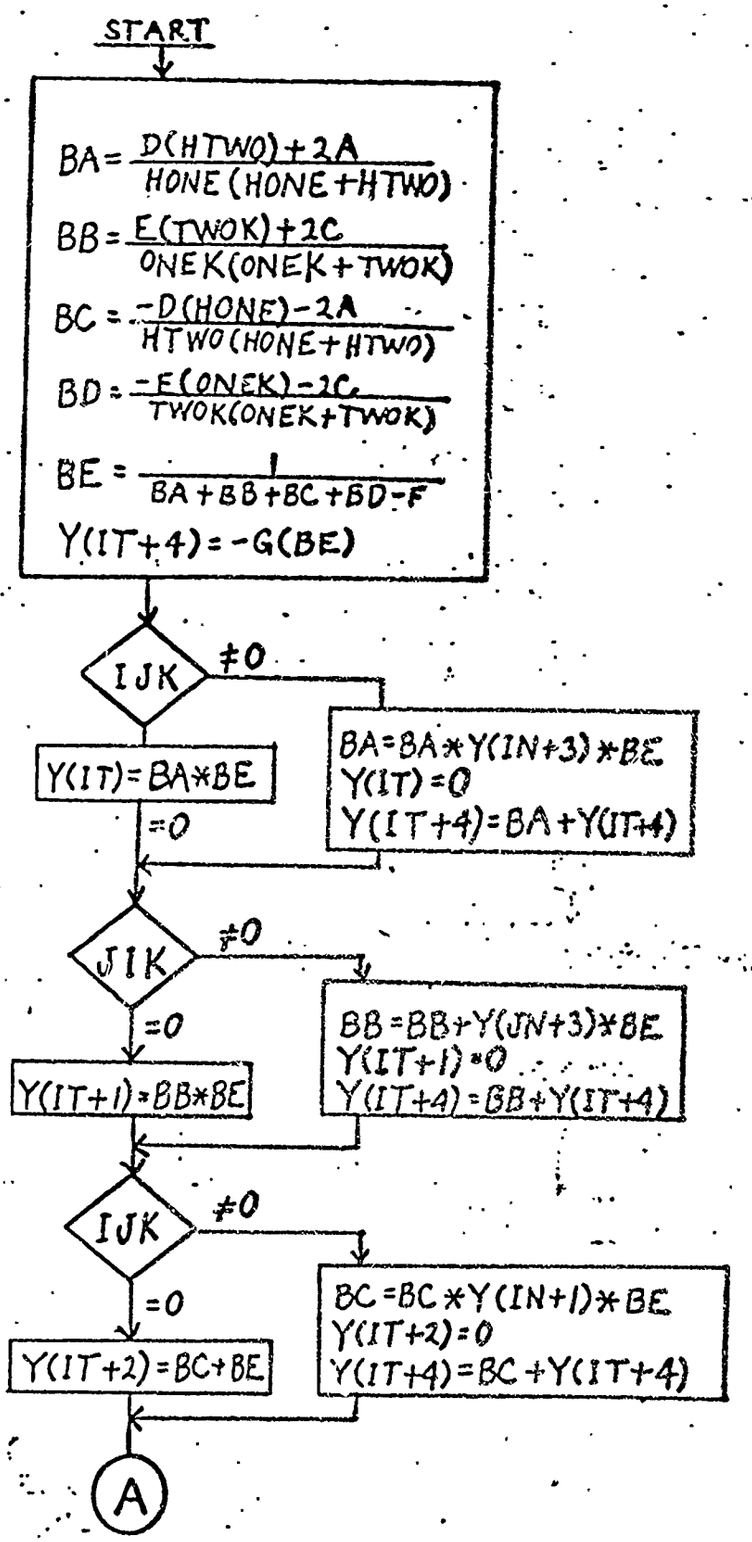
SUBROUTINE GRDSTL

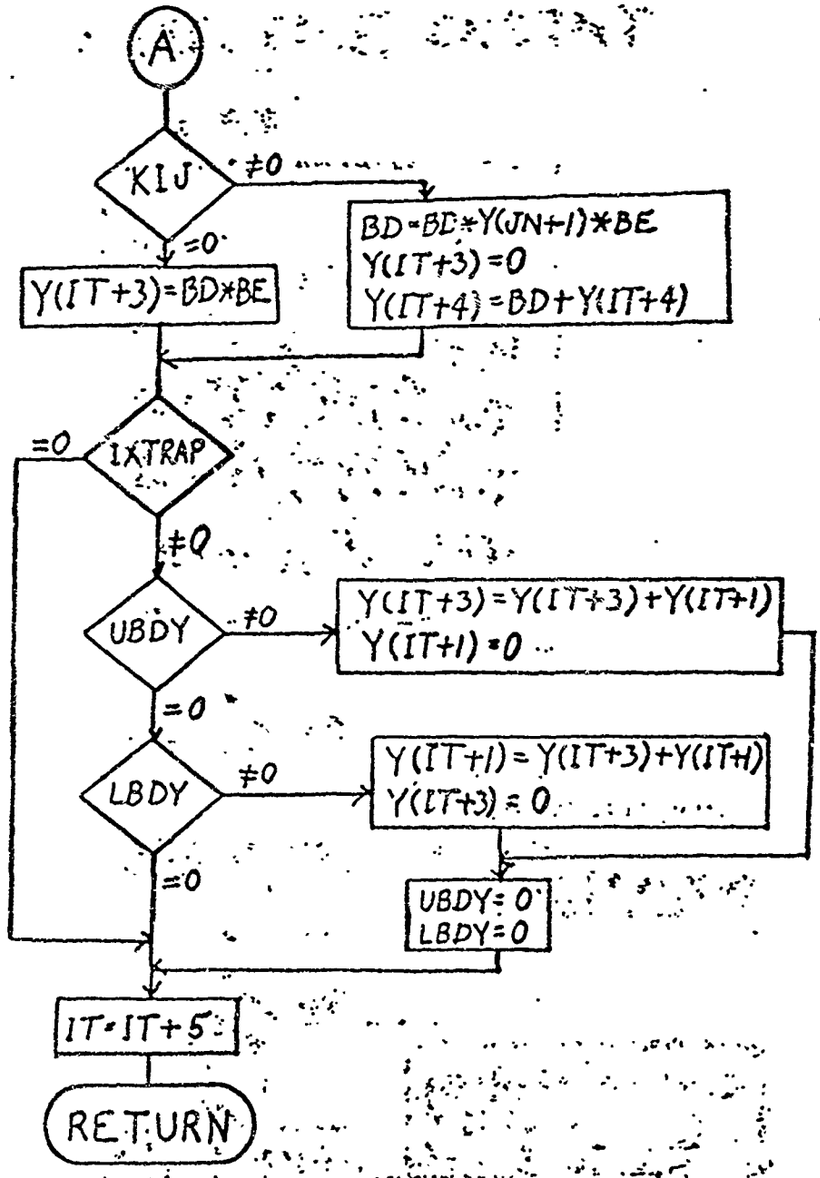


SUBROUTINE ACDEFG



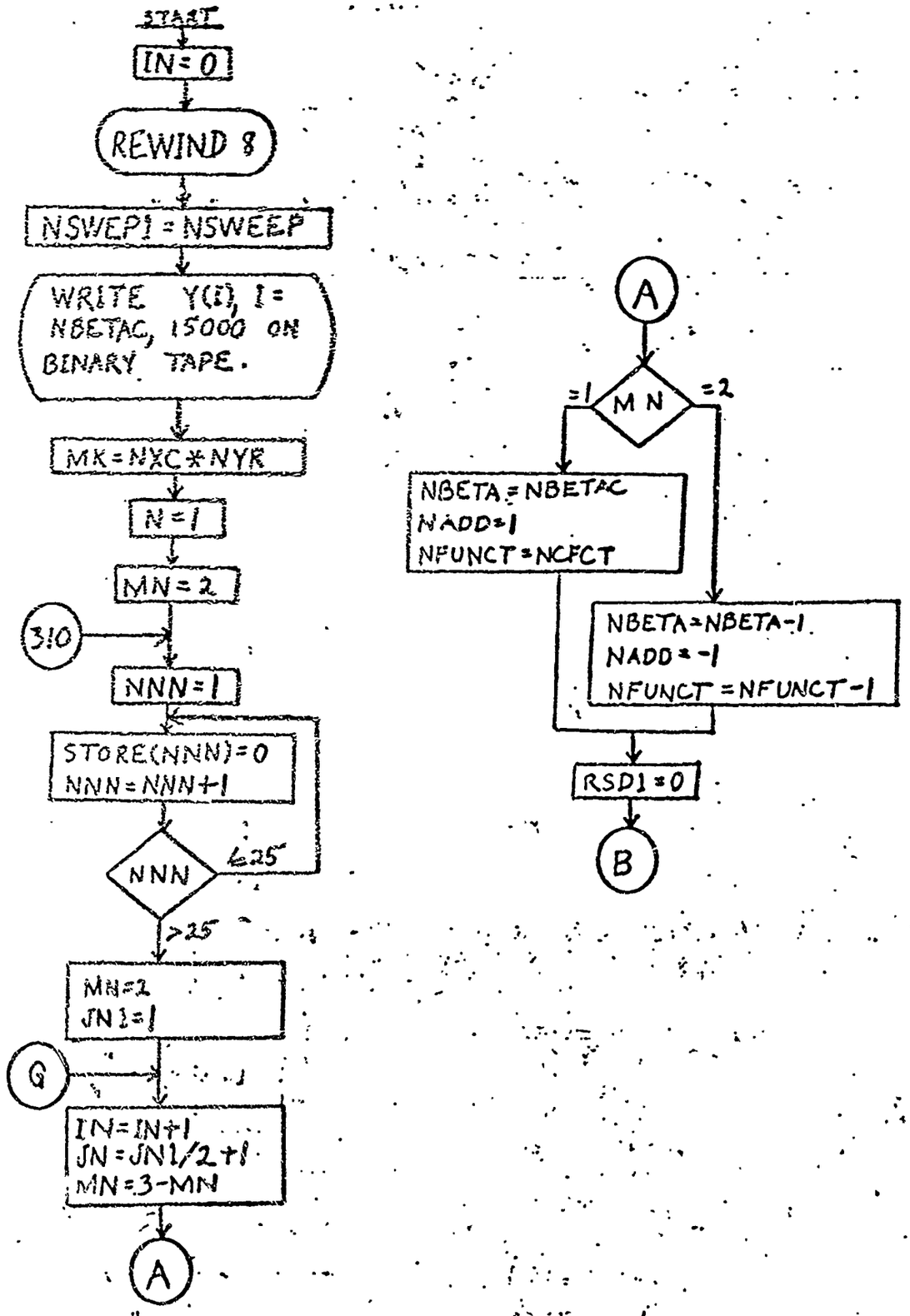
SUBROUTINE CNSTNT

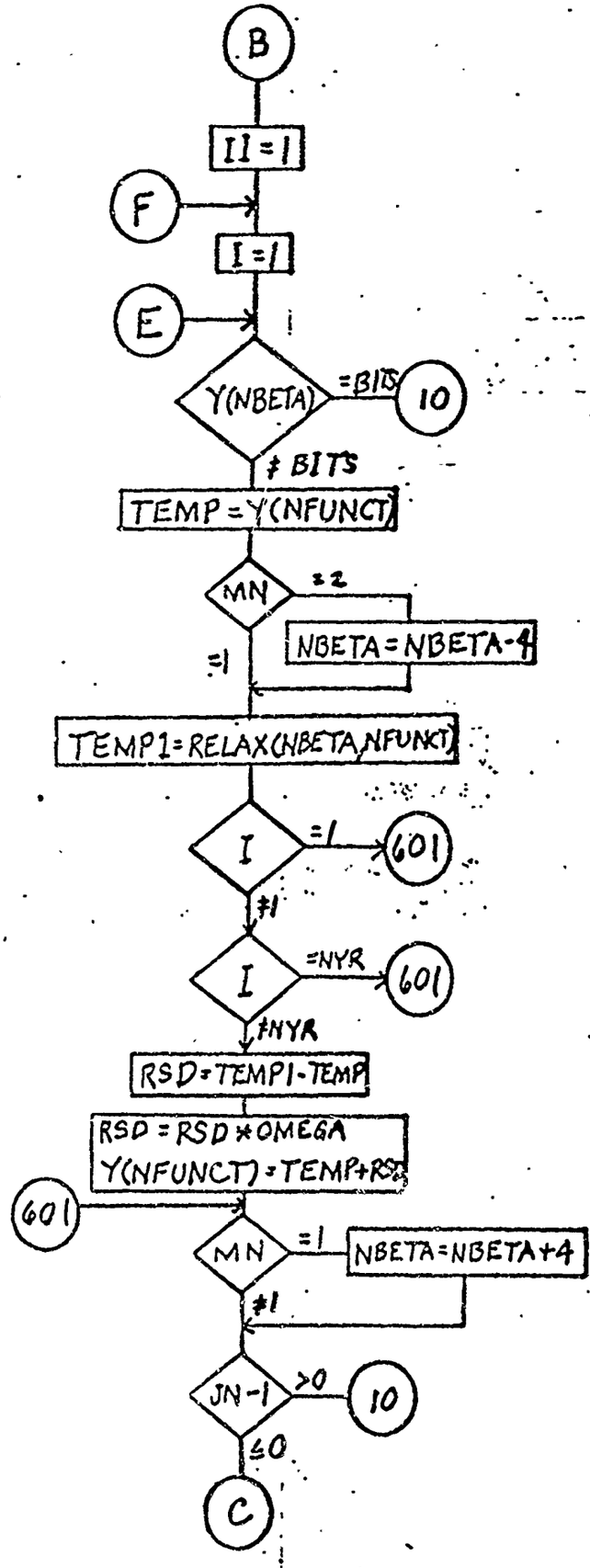



A
KIJ
≠0
=0
BD=BD*Y(JN+1)*BE
Y(IT+3)=0
Y(IT+4)=BD+Y(IT+4)
Y(IT+3)=BD*BE
IXTRAP
=0
≠0
UBDY
≠0
Y(IT+3)=Y(IT+3)+Y(IT+1)
Y(IT+1)=0
=0
LBDY
≠0
Y(IT+1)=Y(IT+3)+Y(IT+1)
Y(IT+3)=0
=0
UBDY=0
LBDY=0
IT=IT+5
RETURN

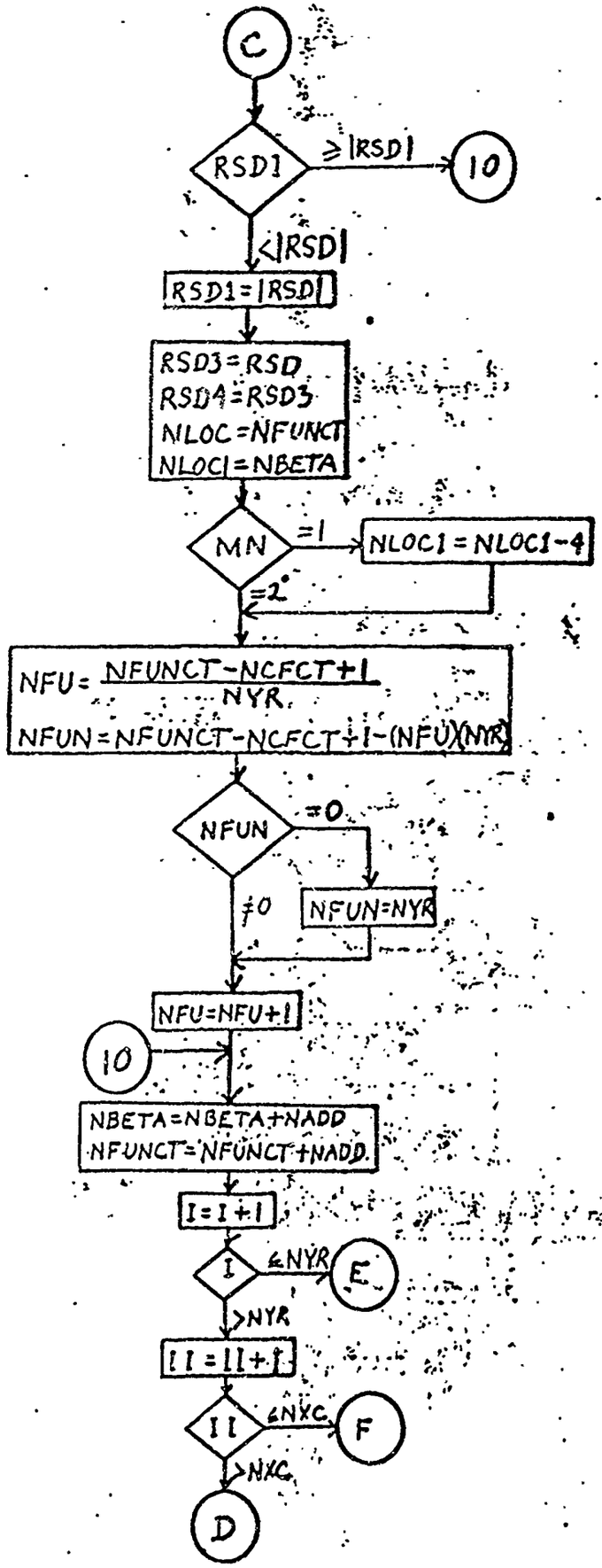
CNSTNT 2

SUBROUTINE SWEEP

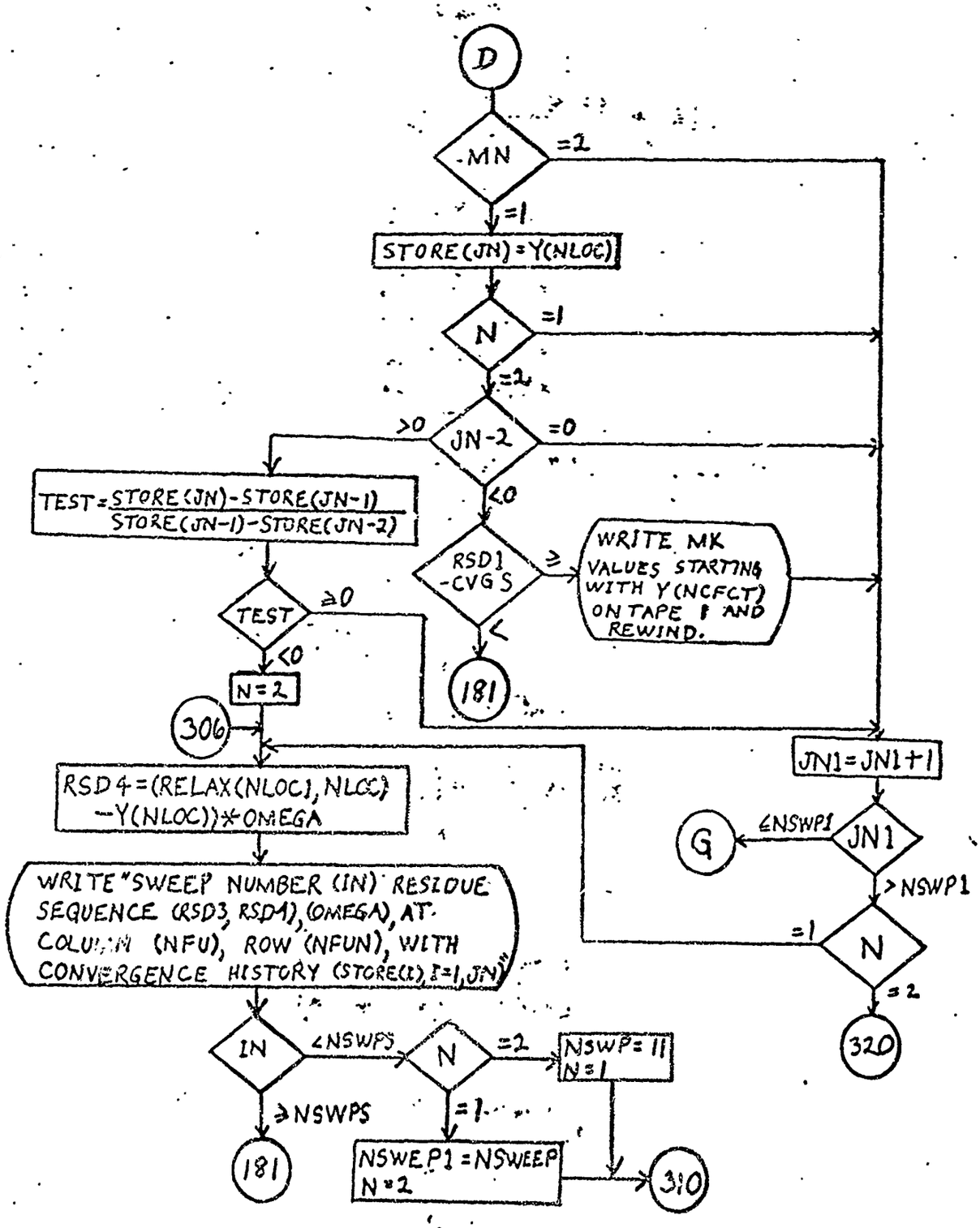



B
II=1
F
I=1
E
Y(NBETA)
=BITS
10
≠ BITS
TEMP=Y(NFUNCT)
MN
=2
NBETA=NBETA-4
=1
TEMP1=RELAX(NBETA,NFUNCT)
I
=1
601
≠1
I
=NYR
601
≠NYR
RSD=TEMP1-TEMP
RSD=RSD*OMEGA
Y(NFUNCT)=TEMP+RSD
601
MN
=1
NBETA=NBETA+4
≠1
JN-1
>0
10
≤0
C

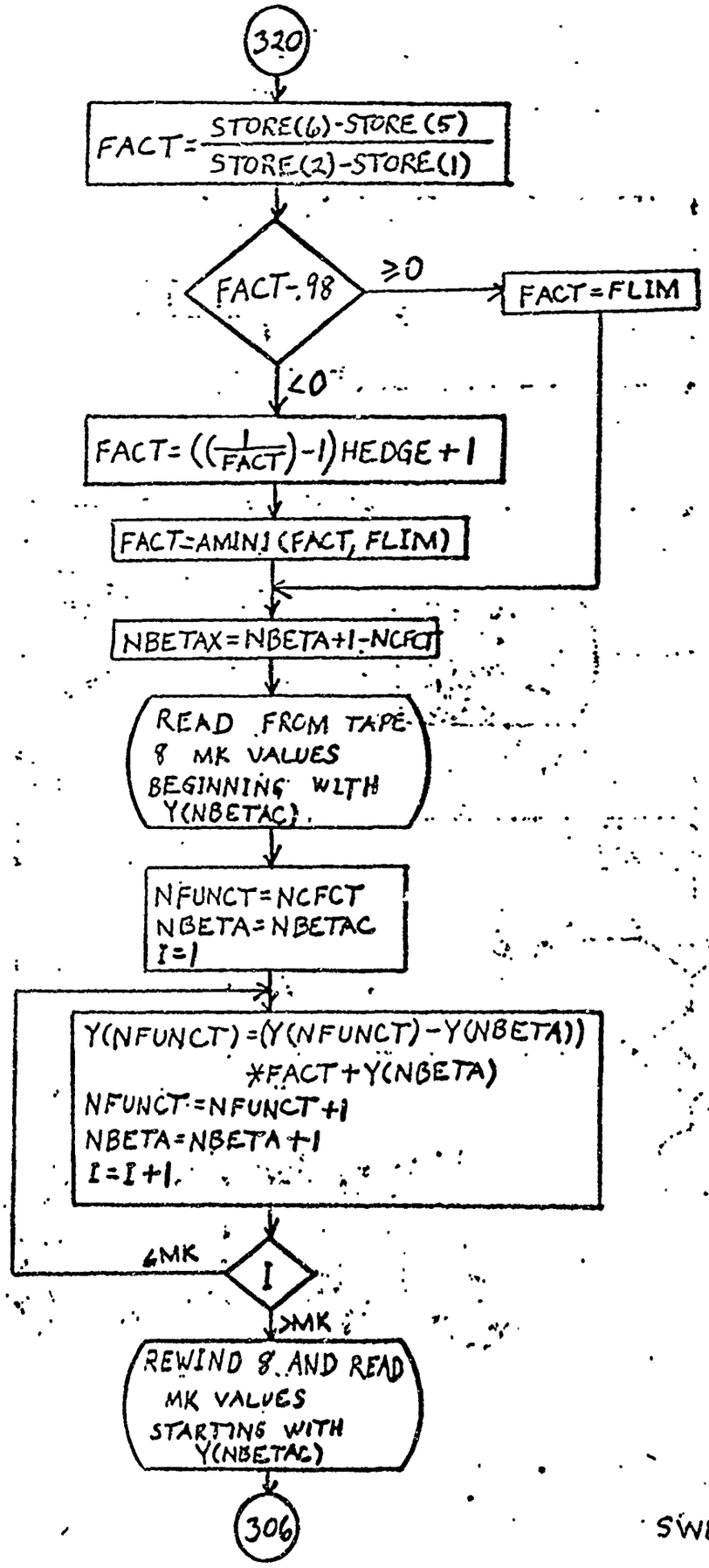
SWEEP 2


C
RSD1
≥|RSD|
10
<|RSD|
RSD1=|RSD|
RSD3=RSD
RSD4=RSD3
NLOC=NFUNCT
NLOC1=NBETA
MN
=1
NLOC1=NLOC1-4
=2
NFU= NFUNCT-NCFCT+1 / NYR
NFUN=NFUNCT-NCFCT+1-(NFU)(NYR)
NFUN
=0
NFUN=NYR
≠0
NFU=NFU+1
10
NBETA=NBETA+NADD
NFUNCT=NFUNCT+NADD
I=I+1
I
≤NYR
E
>NYR
II=II+1
II
≤NXC
F
>NXC
D

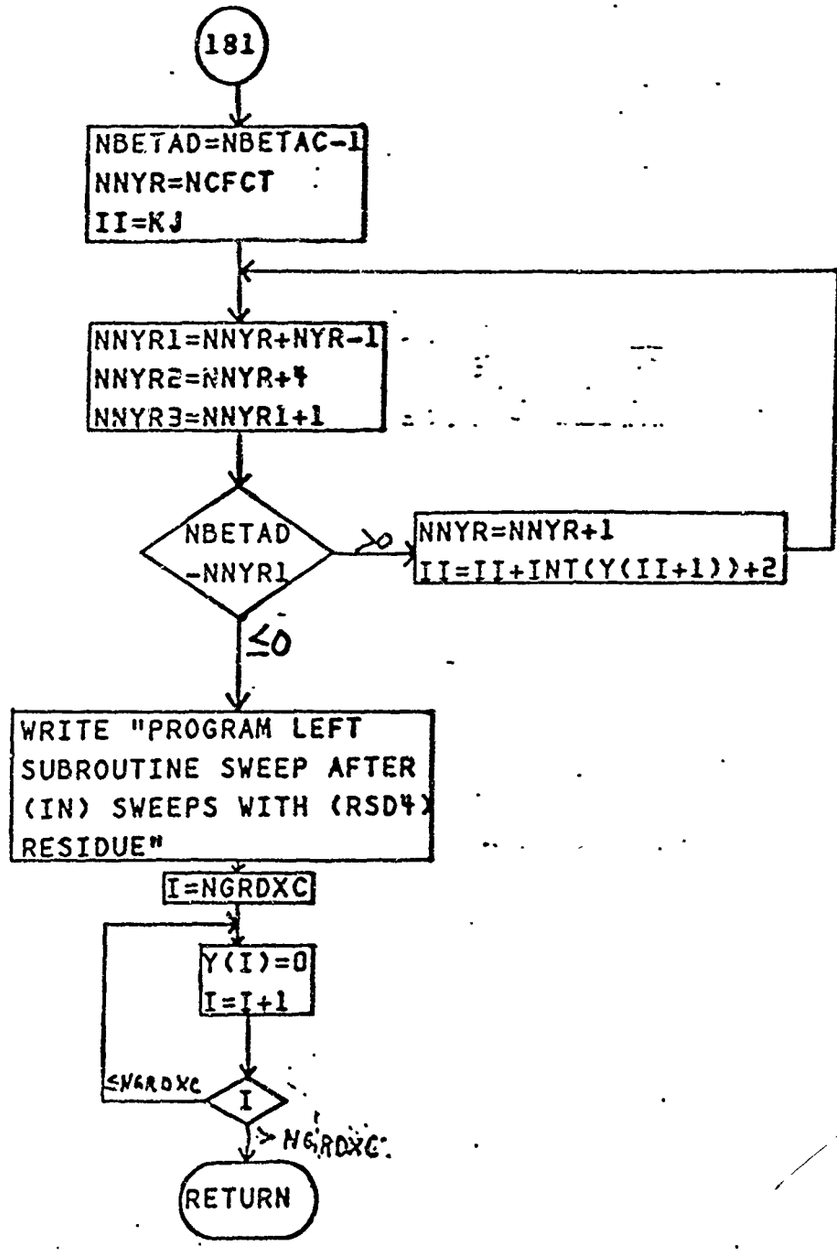
SWEEP 3


D
MN
=2
=1
STORE(JN)=Y(NLOC)
N
=1
=2
JN-2
>0
=0
<0
TEST=STORE(JN)-STORE(JN-1) / STORE(JN-1)-STORE(JN-2)
RSD1 -CVGS
≥
WRITE MK VALUES STARTING WITH Y(NCFCT) ON TAPE 8 AND REWIND.
<
181
TEST
≥0
<0
N=2
306
RSD4=(RELAX(NLOC), NLOC) -Y(NLOC))*OMEGA
WRITE "SWEEP NUMBER (IN) RESIDUE SEQUENCE (RSD3, RSD4), (OMEGA), AT COLUMN (NFU), ROW (NFUN), WITH CONVERGENCE HISTORY (STORE(I), I=1,JN)"
IN
<NSWPS
N
=2
NSWP=11
N=1
≥NSWPS
=1
181
NSWEP1=NSWEEP
N=2
310
JN1=JN1+1
JN1
<NSWP1
G
>NSWP1
N
=1
=2
320

SWEEP 4


320
FACT = (STORE(6)-STORE(5)) / (STORE(2)-STORE(1))
FACT-.98
≥0
FACT=FLIM
<0
FACT= ((1/FACT)-1) HEDGE+1
FACT=AMIN1 (FACT, FLIM)
NBETAX=NBETA+1-NCFCT
READ FROM TAPE 8 MK VALUES BEGINNING WITH Y(NBETAC).
NFUNCT=NCFCT
NBETA=NBETAC
I=1
Y(NFUNCT)=(Y(NFUNCT)-Y(NBETA)) *FACT+Y(NBETA)
NFUNCT=NFUNCT+1
NBETA=NBETA+1
I=I+1
<MK
I
>MK
REWIND 8 AND READ MK VALUES STARTING WITH Y(NBETAC)
306

SWEEP 5


181
NBETAD=NBETAC-1
NNYR=NCFCT
II=KJ
NNYR1=NNYR+NYR-1
NNYR2=NNYR+4
NNYR3=NNYR1+1
NBETAD
-NNYR1
>0
NNYR=NNYR+1
II=II+INT(Y(II+1))+2
≤0
WRITE "PROGRAM LEFT
SUBROUTINE SWEEP AFTER
(IN) SWEEPS WITH (RSD4)
RESIDUE"
I=NGRDXC
Y(I)=0
I=I+1
≤NGRDXC
I
> NGRDXC
RETURN

SWEEP 6

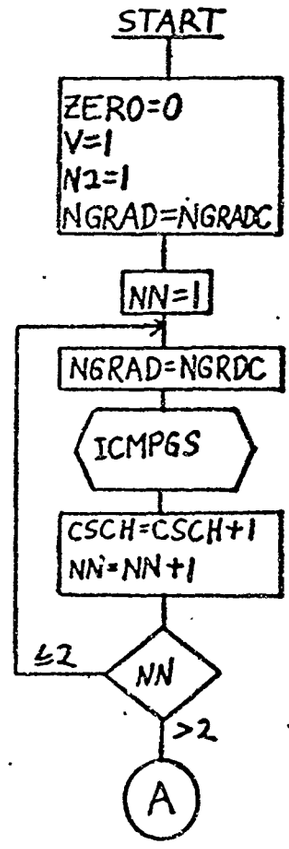
FUNCTION RELAX

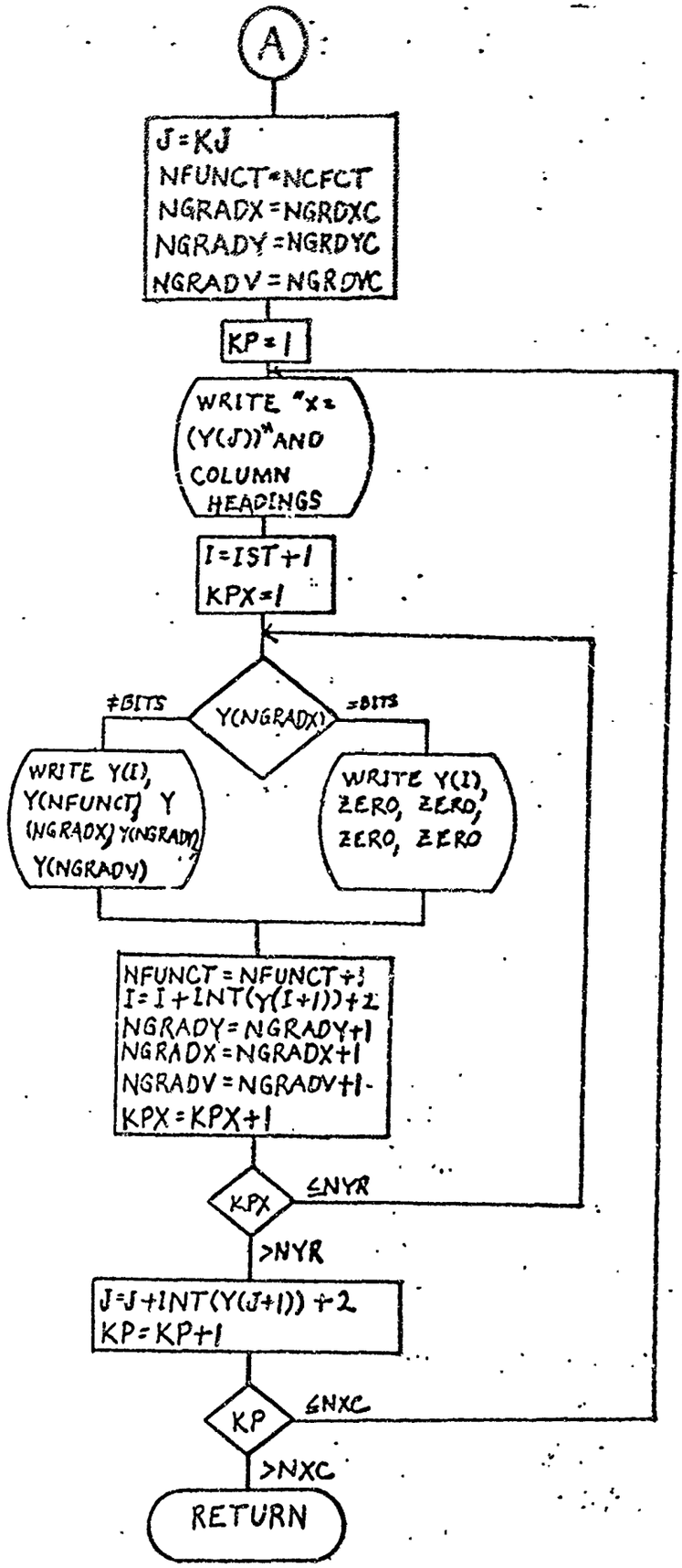
START

```
N1=NFUNCT+NYR
N3=NFUNCT-NYR
RELAX=Y(N1)*Y(NB)
      +Y(NFUNCT+1)*Y(NB+1)
      +Y(N3)*Y(NB+2)
      +Y(NFUNCT-1)*Y(NB+3)
      +Y(NB+4)
```

RETURN

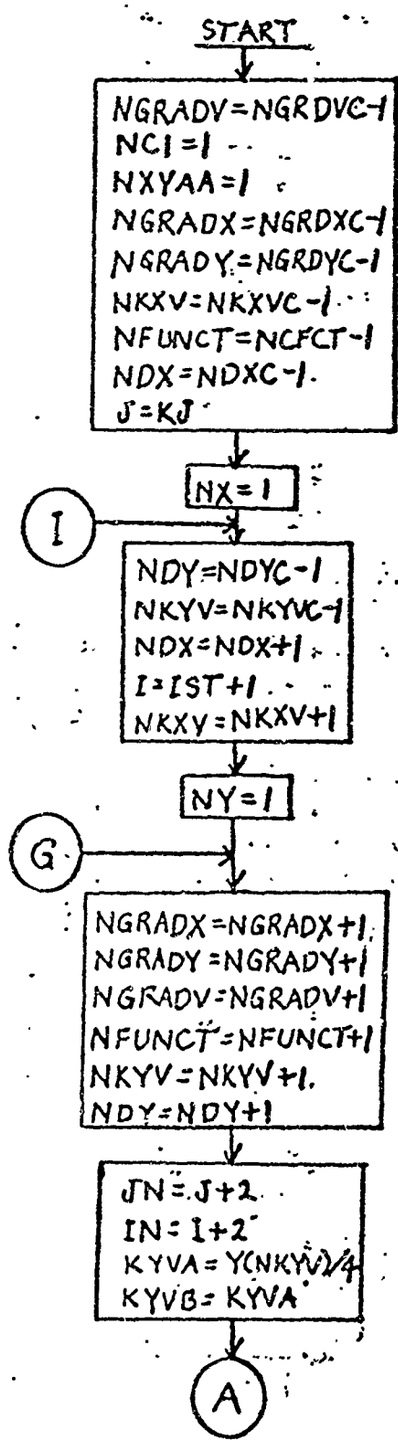
SUBROUTINE VELOC

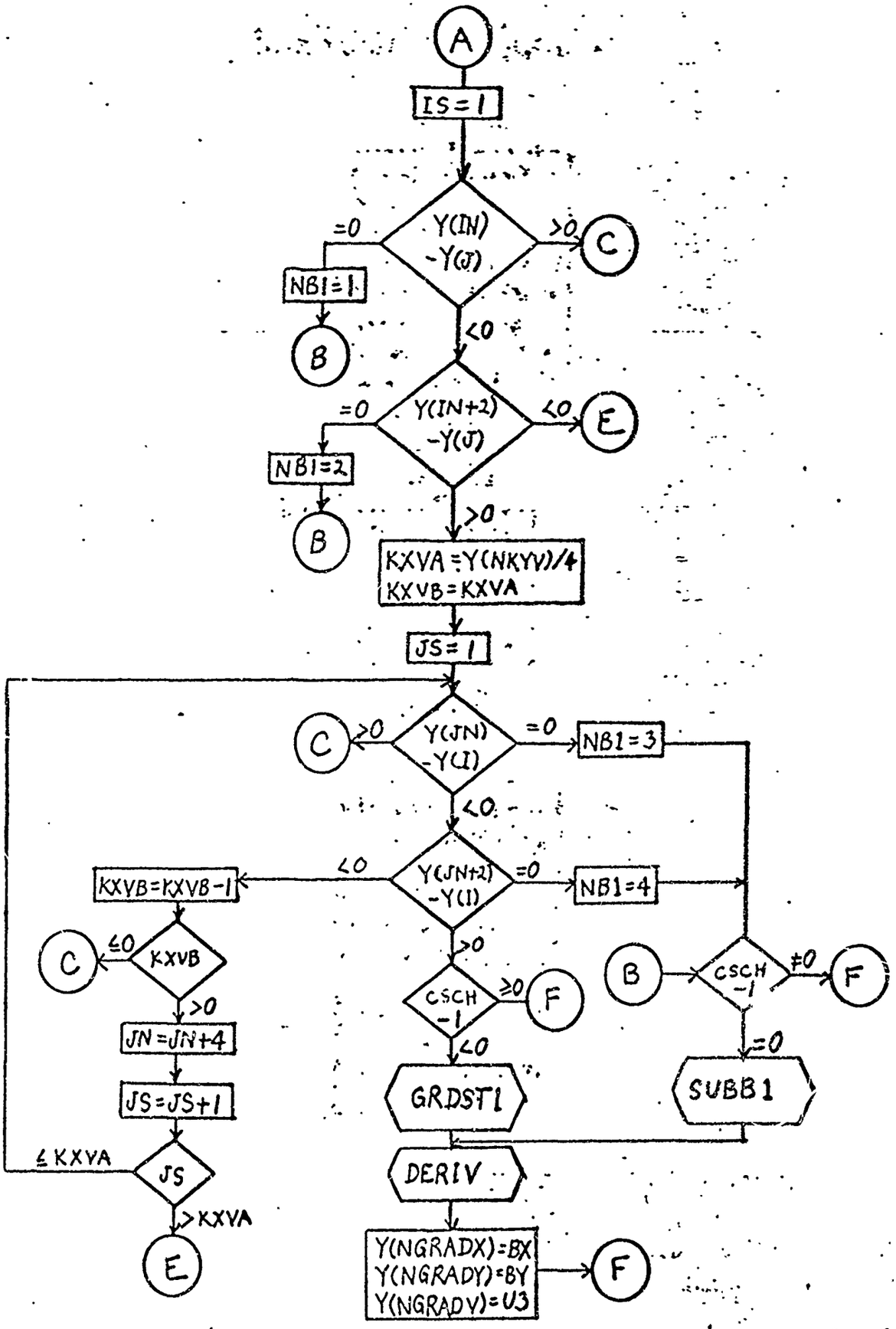



A
J=KJ
NFUNCT=NCFCT
NGRADX=NGRDXC
NGRADY=NGRDYC
NGRADV=NGRDVC
KP=1
WRITE "X=
(Y(J))" AND
COLUMN
HEADINGS
I=IST+1
KPX=1
Y(NGRADX)
≠BITS
=BITS
WRITE Y(I),
Y(NFUNCT), Y
(NGRADX), Y(NGRADY),
Y(NGRADV)
WRITE Y(I),
ZERO, ZERO,
ZERO, ZERO
NFUNCT=NFUNCT+3
I=I+INT(Y(I+1))+2
NGRADY=NGRADY+1
NGRADX=NGRADX+1
NGRADV=NGRADV+1
KPX=KPX+1
KPX
≤NYR
>NYR
J=J+INT(Y(J+1))+2
KP=KP+1
KP
≤NXC
>NXC
RETURN

VELOC 2

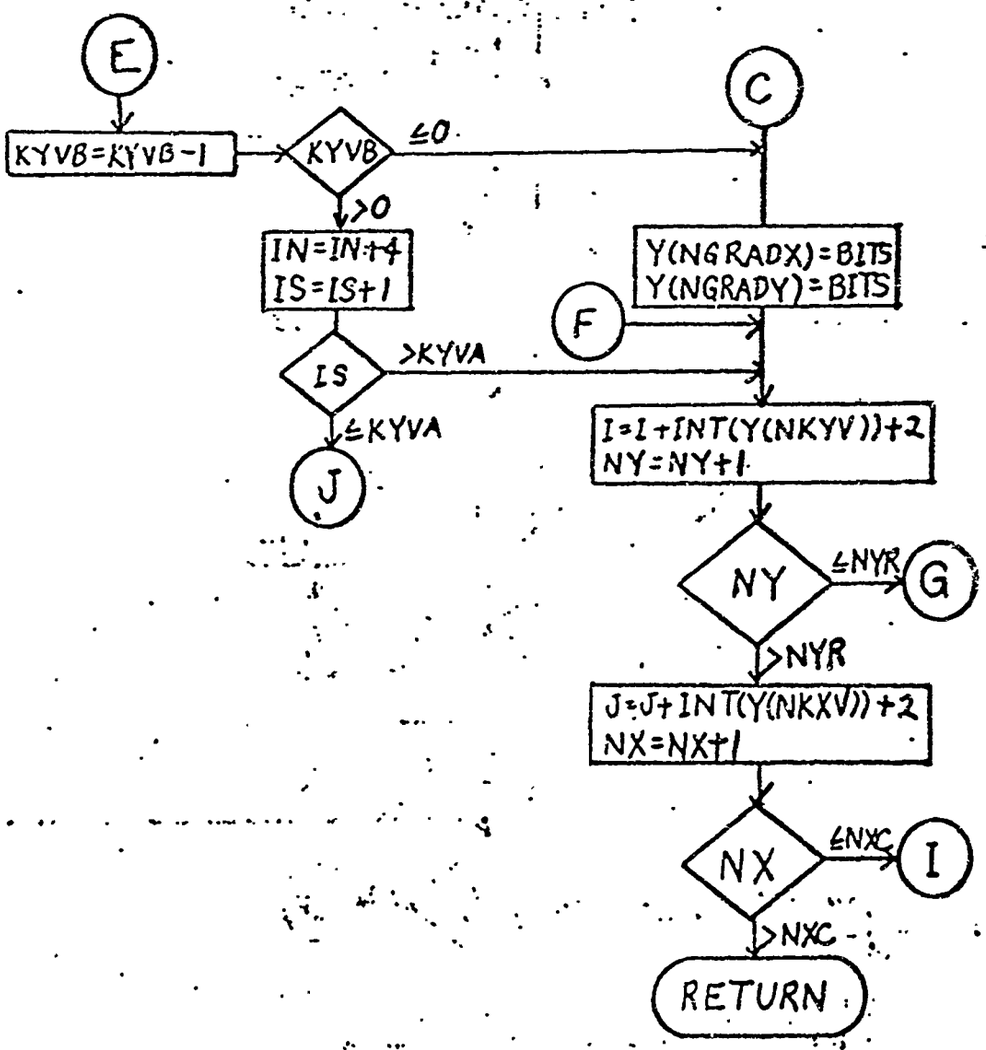
SUBROUTINE ICMPGS





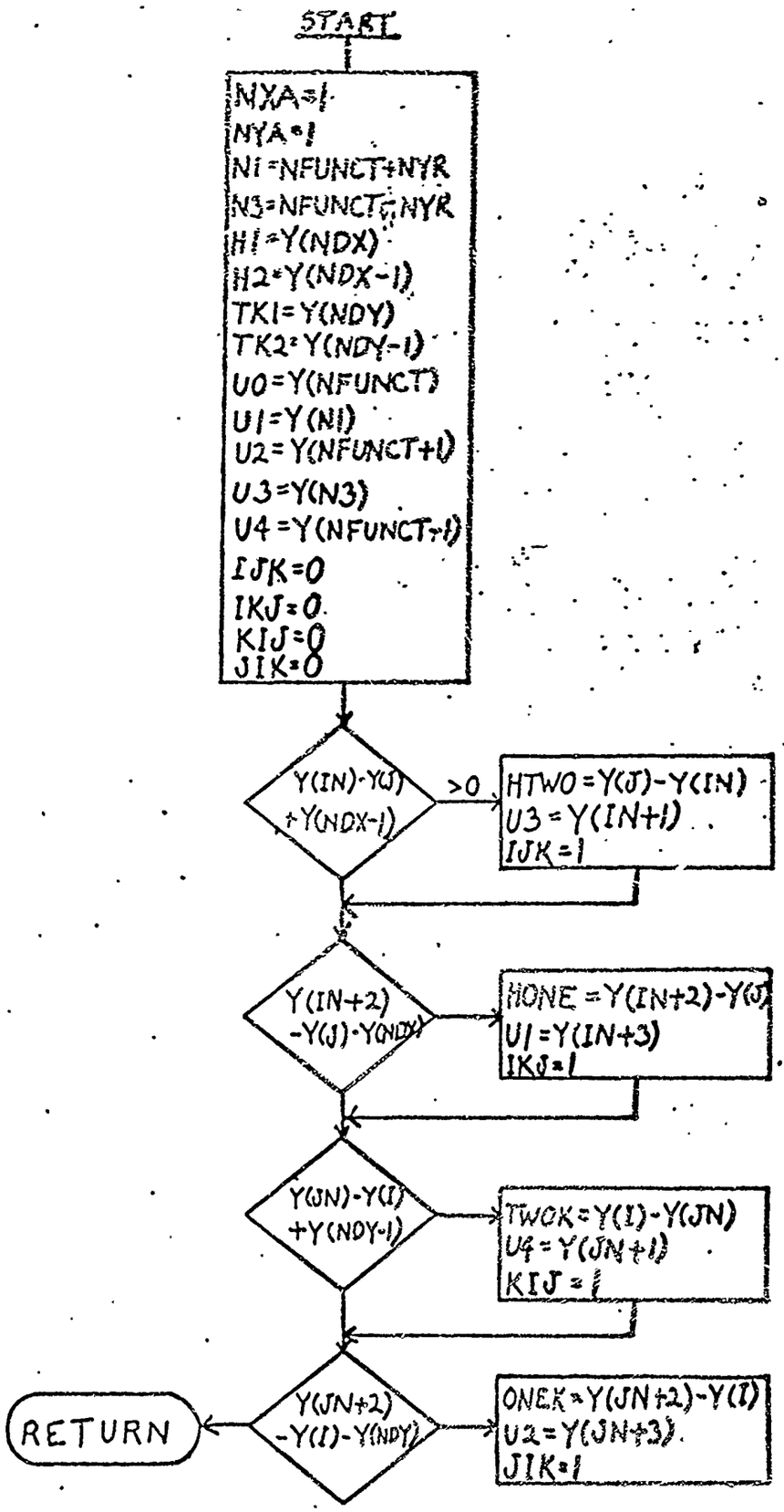
ICMPGS 2

REVISED

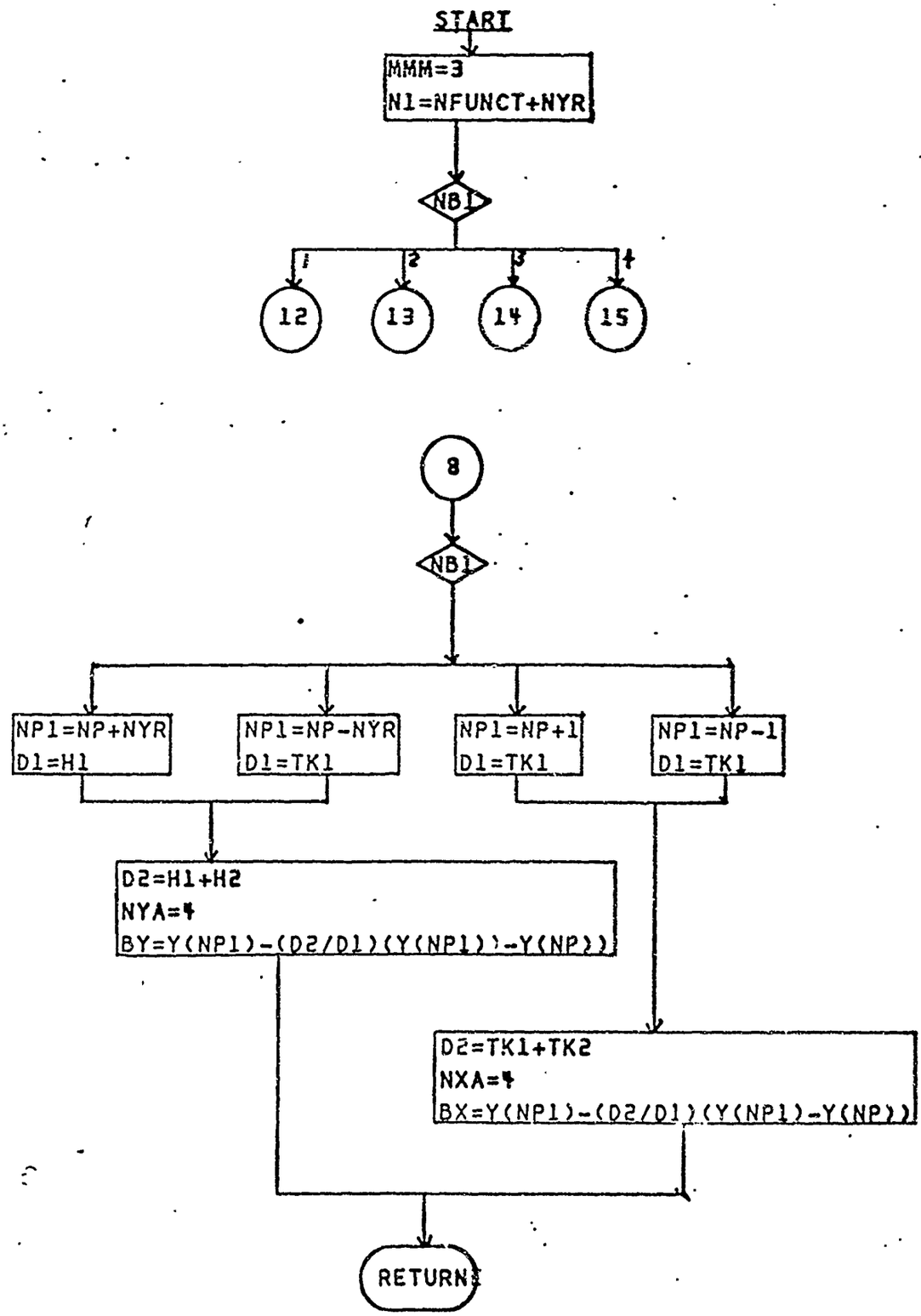

E
KYVB=KYVB-1
KYVB
≤0
>0
C
IN=IN+4
IS=IS+1
Y(NGRADX)=BITS
Y(NGRADY)=BITS
F
IS
>KYVA
≤KYVA
J
I=I+INT(Y(NKYV))+2
NY=NY+1
NY
≤NYR
G
>NYR
J=J+INT(Y(NKXV))+2
NX=NX+1
NX
≤NXC
I
>NXC
RETURN

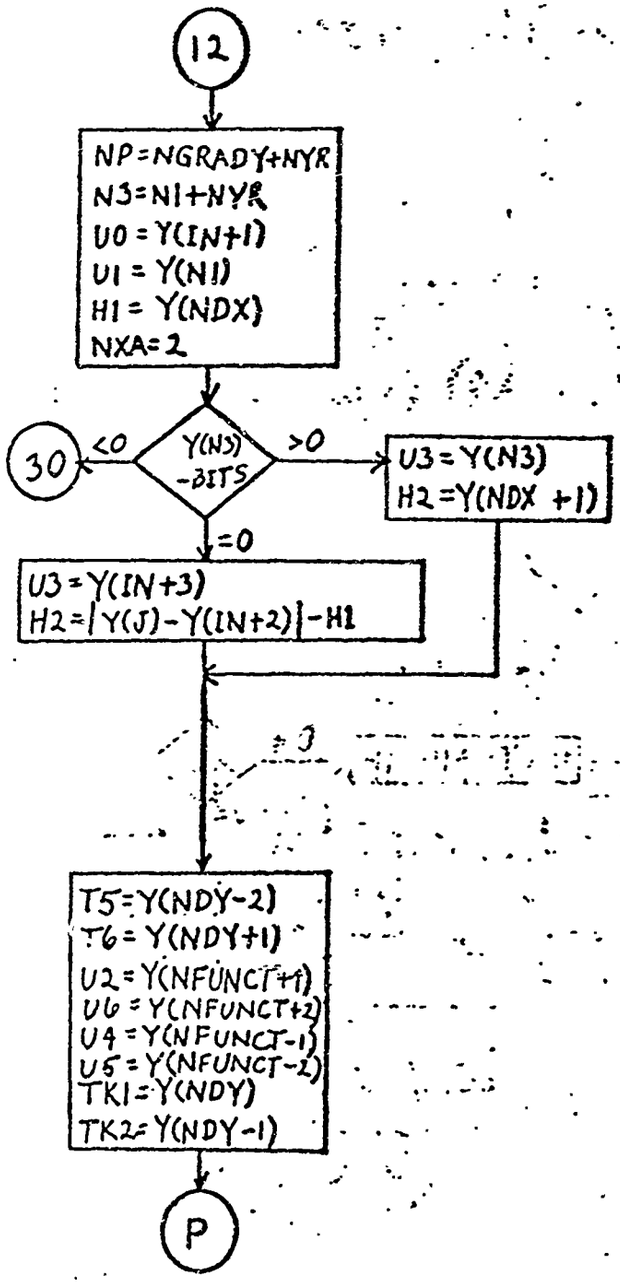
ICMPGS 3

SUBROUTINE GRDSTI

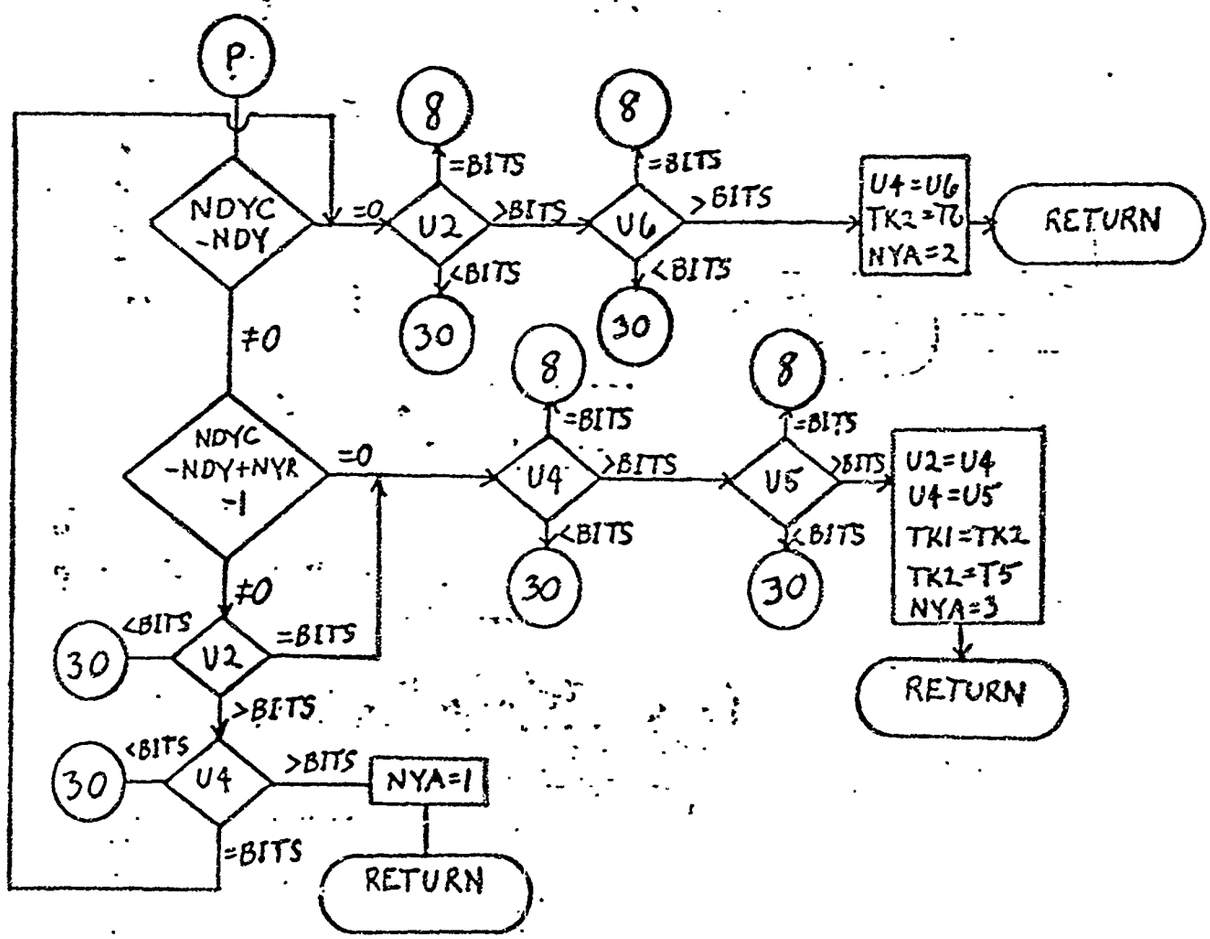


SUBROUTINE SUBB1

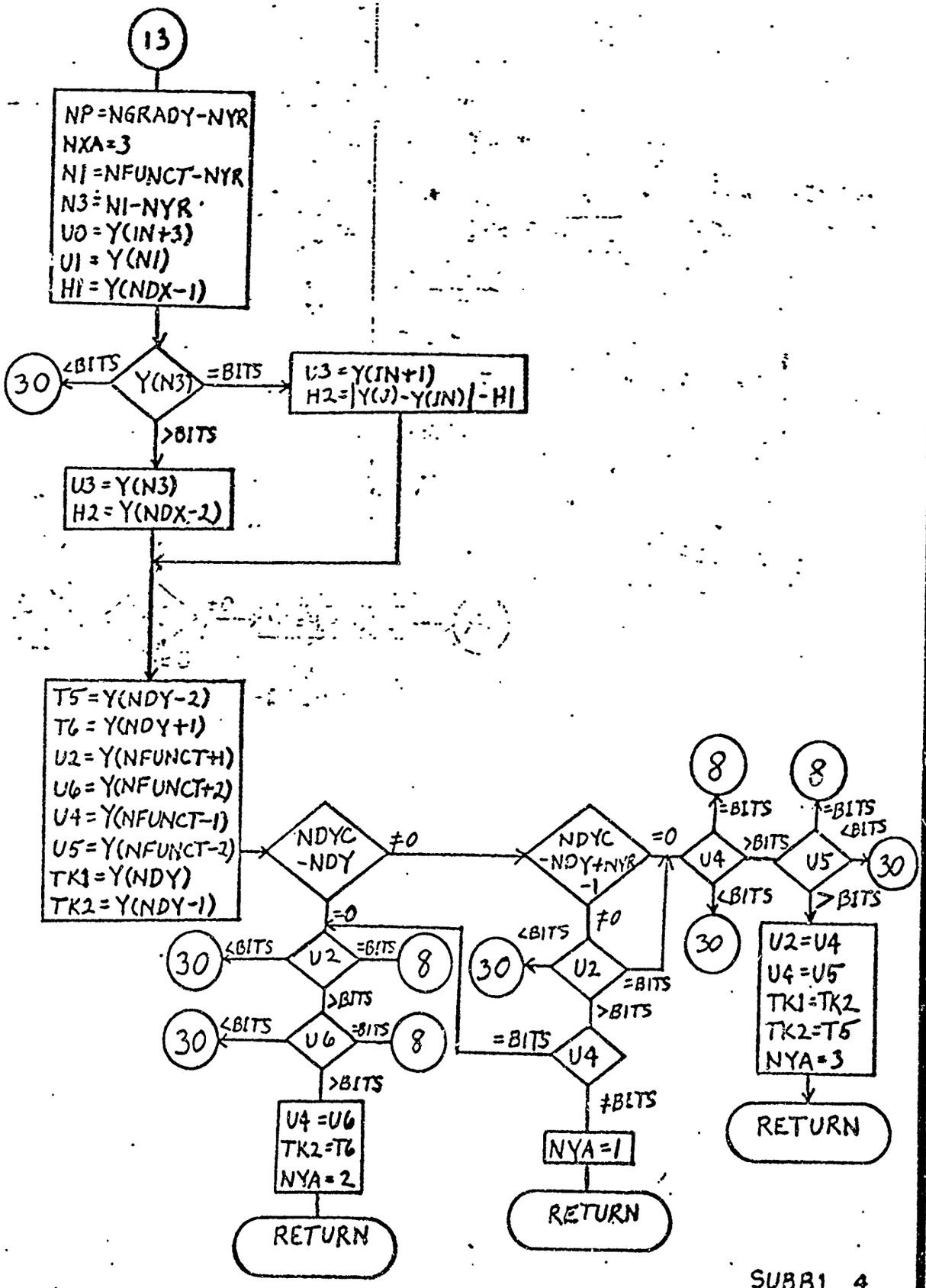



12
NP=NGRADY+NYR
N3=N1+NYR
U0=Y(IN+1)
U1=Y(N1)
H1=Y(NDX)
NXA=2
Y(N3) -BITS
<0
30
>0
U3=Y(N3)
H2=Y(NDX+1)
=0
U3=Y(IN+3)
H2=|Y(J)-Y(IN+2)|-H1
T5=Y(NDY-2)
T6=Y(NDY+1)
U2=Y(NFUNCT+1)
U6=Y(NFUNCT+2)
U4=Y(NFUNCT-1)
U5=Y(NFUNCT-2)
TK1=Y(NDY)
TK2=Y(NDY-1)
P

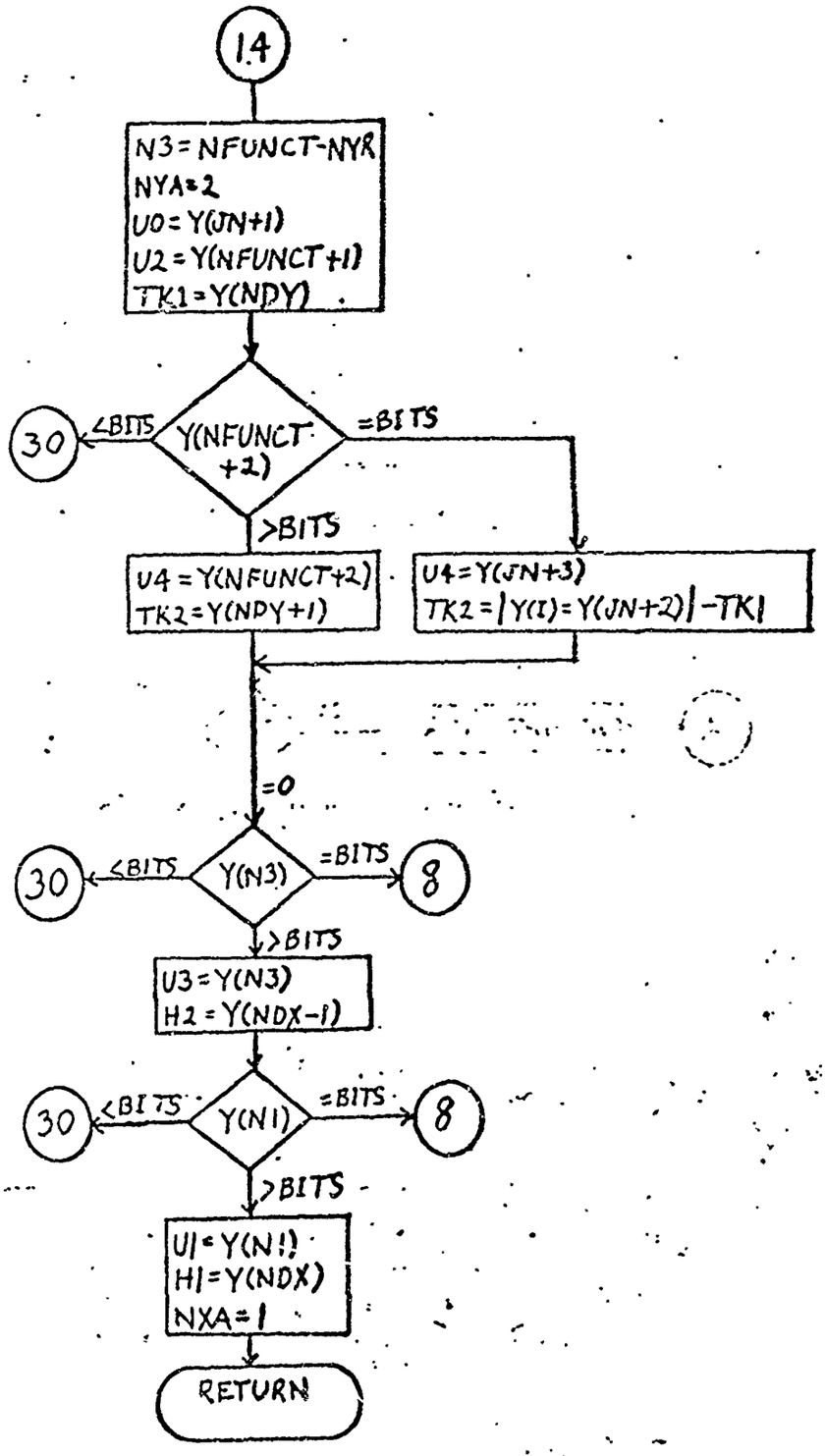
SUBB1 2



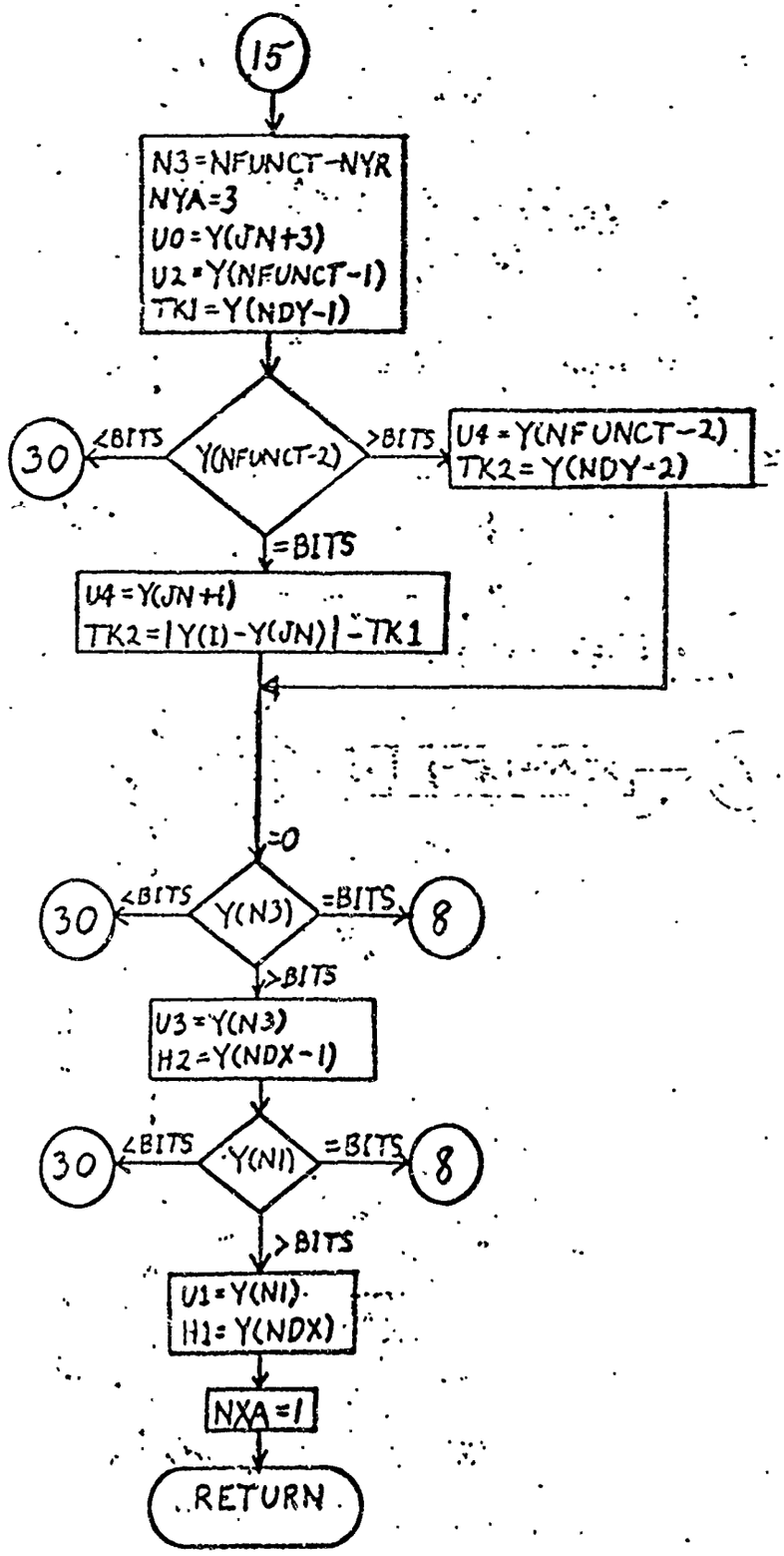
REVISED
E 1686 R8


13
NP=NGRADY-NYR
NXA=3
N1=NFUNCT-NYR
N3=N1-NYR
U0=Y(IN+3)
U1=Y(N1)
H1=Y(NDX-1)
Y(N3)
<BITS
=BITS
>BITS
30
U3=Y(IN+1)
H2=|Y(J)-Y(JN)|-H1
U3=Y(N3)
H2=Y(NDX-2)
T5=Y(NDY-2)
T6=Y(NDY+1)
U2=Y(NFUNCT+1)
U6=Y(NFUNCT+2)
U4=Y(NFUNCT-1)
U5=Y(NFUNCT-2)
TK1=Y(NDY)
TK2=Y(NDY-1)
NDYC-NDY
≠0
=0
NDYC-NDY+NYR-1
U2
U6
U4
U5
8
U4=U6
TK2=T6
NYA=2
RETURN
NYA=1
U2=U4
U4=U5
TK1=TK2
TK2=T5
NYA=3

SUBB1 4

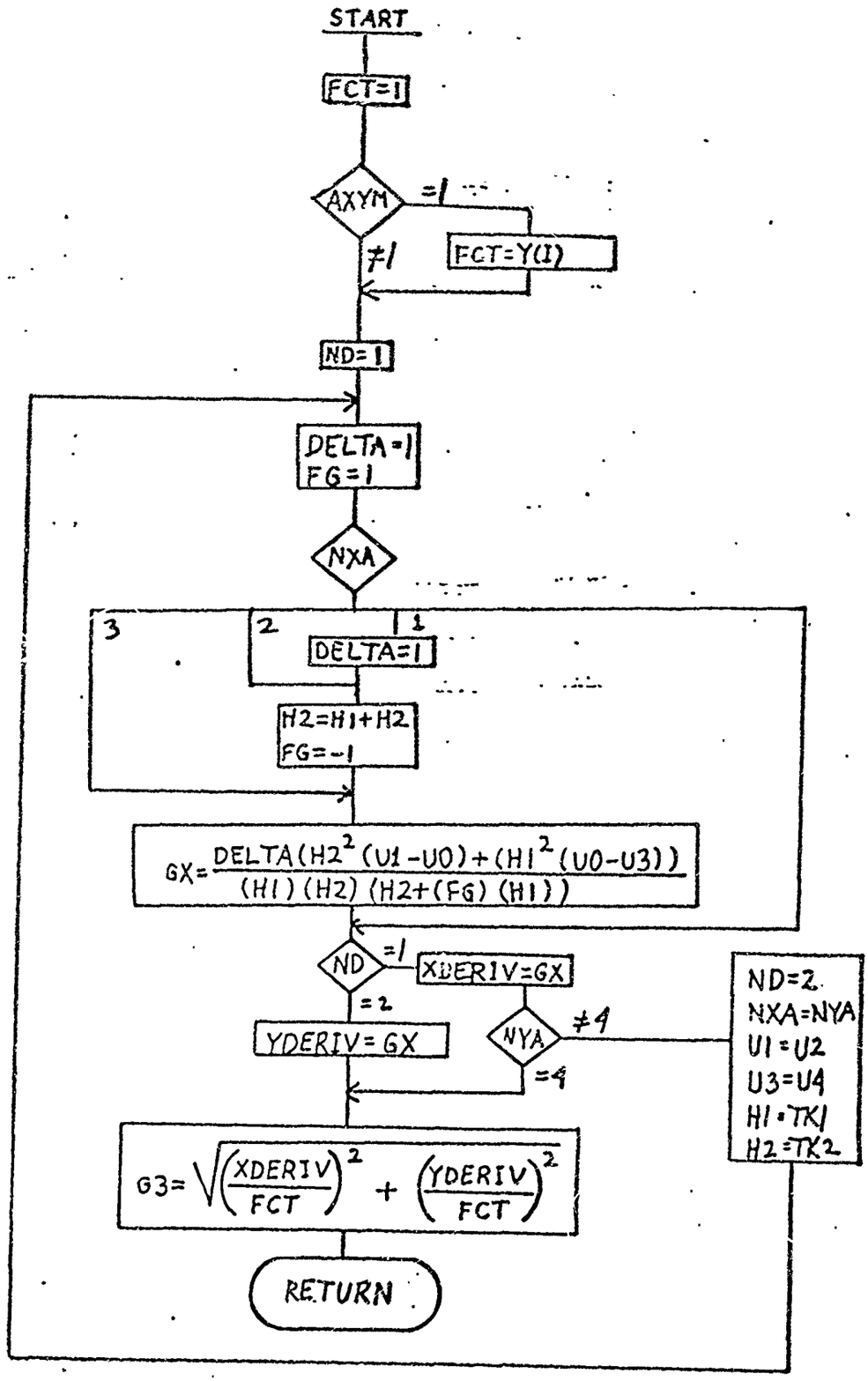

14
N3=NFUNCT-NYR
NYA=2
U0=Y(JN+1)
U2=Y(NFUNCT+1)
TK1=Y(NDY)
Y(NFUNCT+2)
<BITS
30
=BITS
>BITS
U4=Y(NFUNCT+2)
TK2=Y(NDY+1)
U4=Y(JN+3)
TK2=|Y(I)=Y(JN+2)|-TK1
=0
Y(N3)
<BITS
30
=BITS
8
>BITS
U3=Y(N3)
H2=Y(NDX-1)
Y(N1)
<BITS
30
=BITS
8
>BITS
U1=Y(N1)
H1=Y(NDX)
NXA=1
RETURN

SUBB1.5

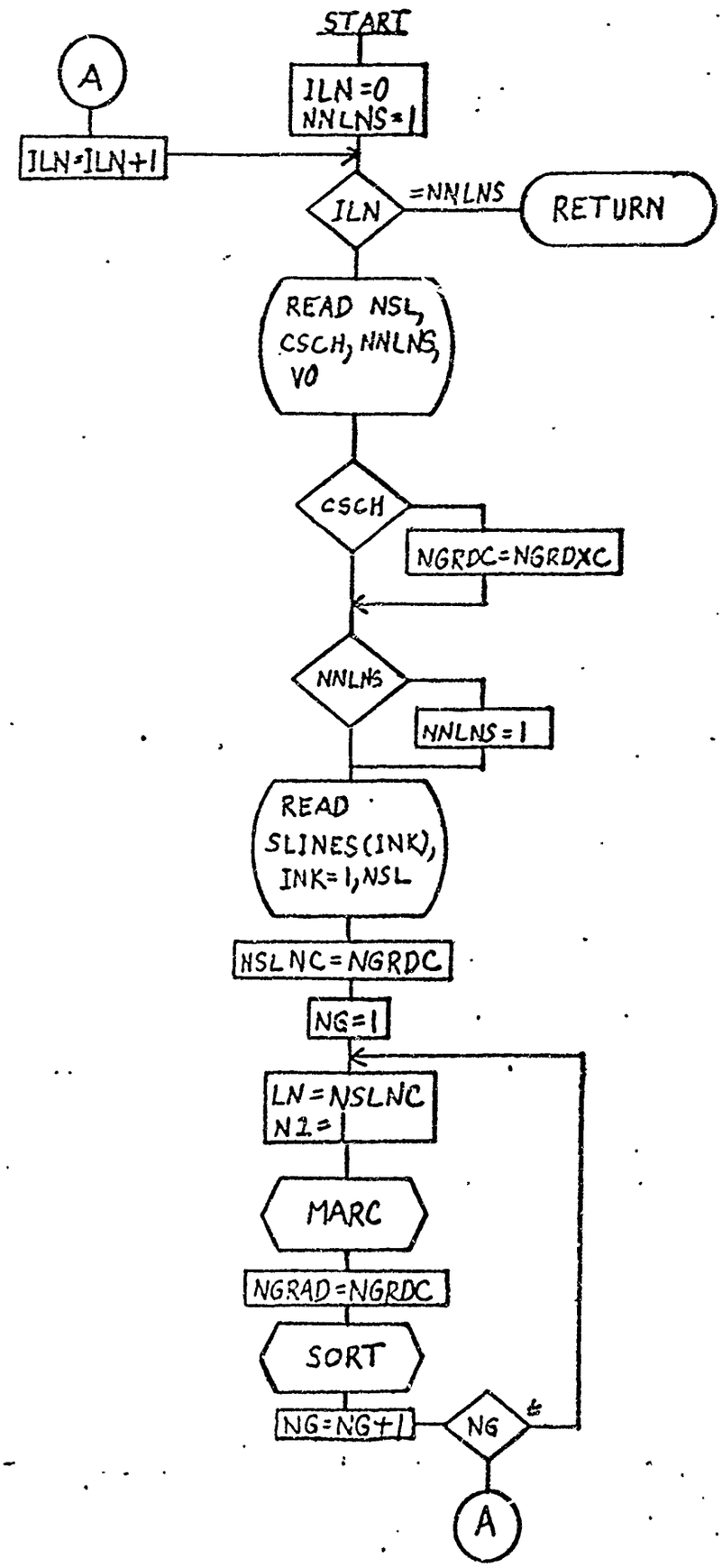

15
N3=NFUNCT-NYR
NYA=3
U0=Y(JN+3)
U2=Y(NFUNCT-1)
TK1=Y(NDY-1)
Y(NFUNCT-2)
<BITS
30
>BITS
U4=Y(NFUNCT-2)
TK2=Y(NDY-2)
=BITS
U4=Y(JN+1)
TK2=|Y(I)-Y(JN)|-TK1
=0
Y(N3)
<BITS
30
=BITS
8
>BITS
U3=Y(N3)
H2=Y(NDX-1)
Y(N1)
<BITS
30
=BITS
8
>BITS
U1=Y(N1)
H1=Y(NDX)
NXA=1
RETURN

SUBB1 6

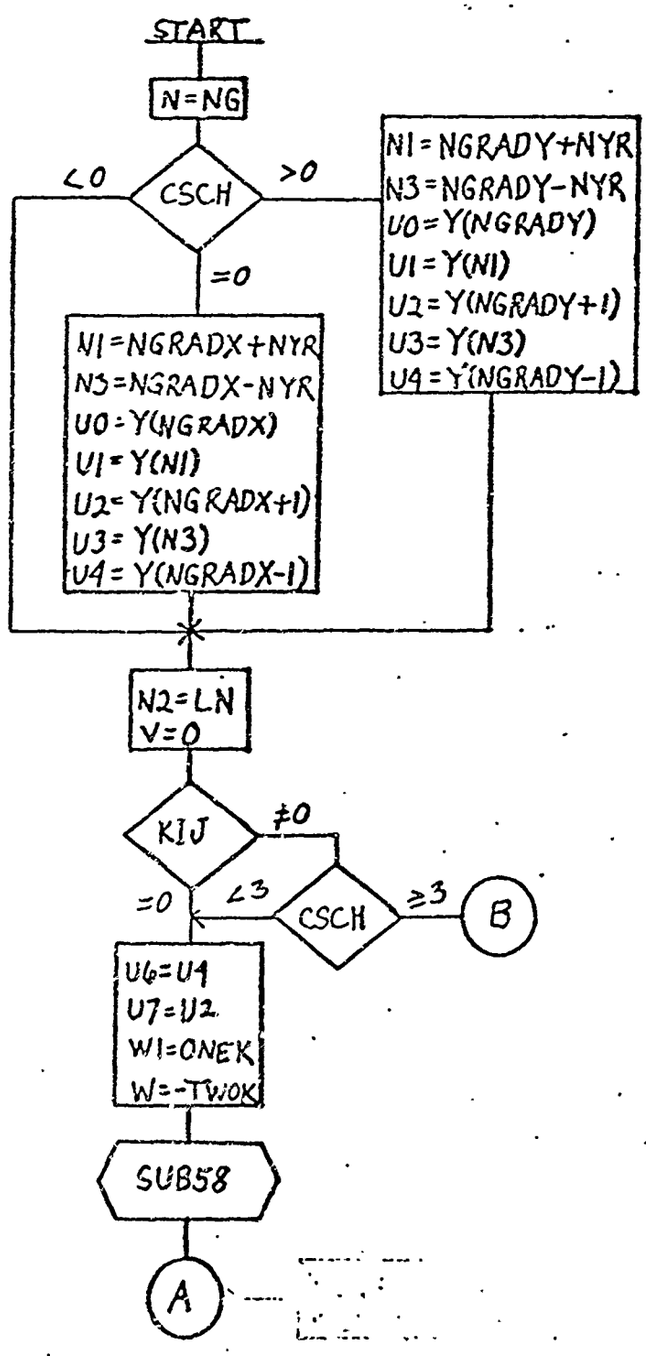
SUBROUTINE DERIV

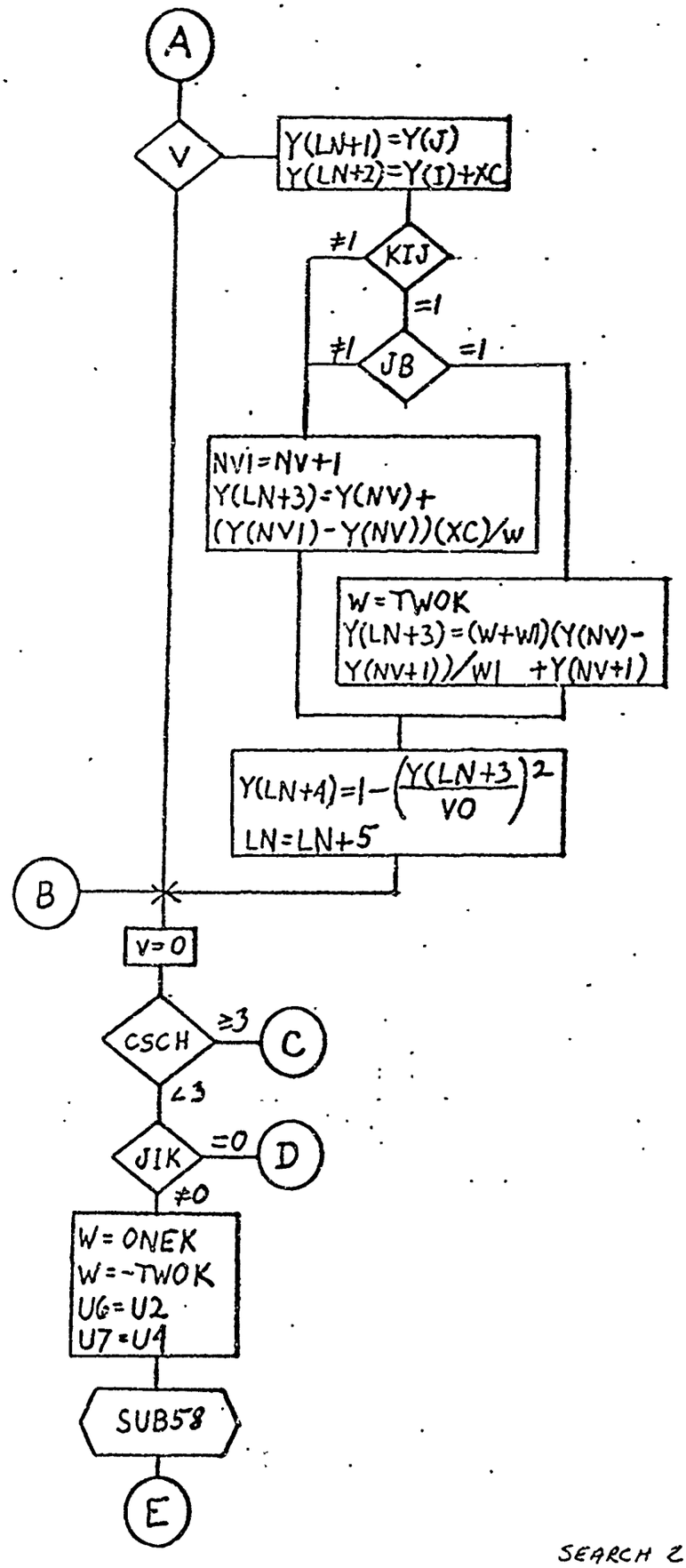


SUBROUTINE LINES

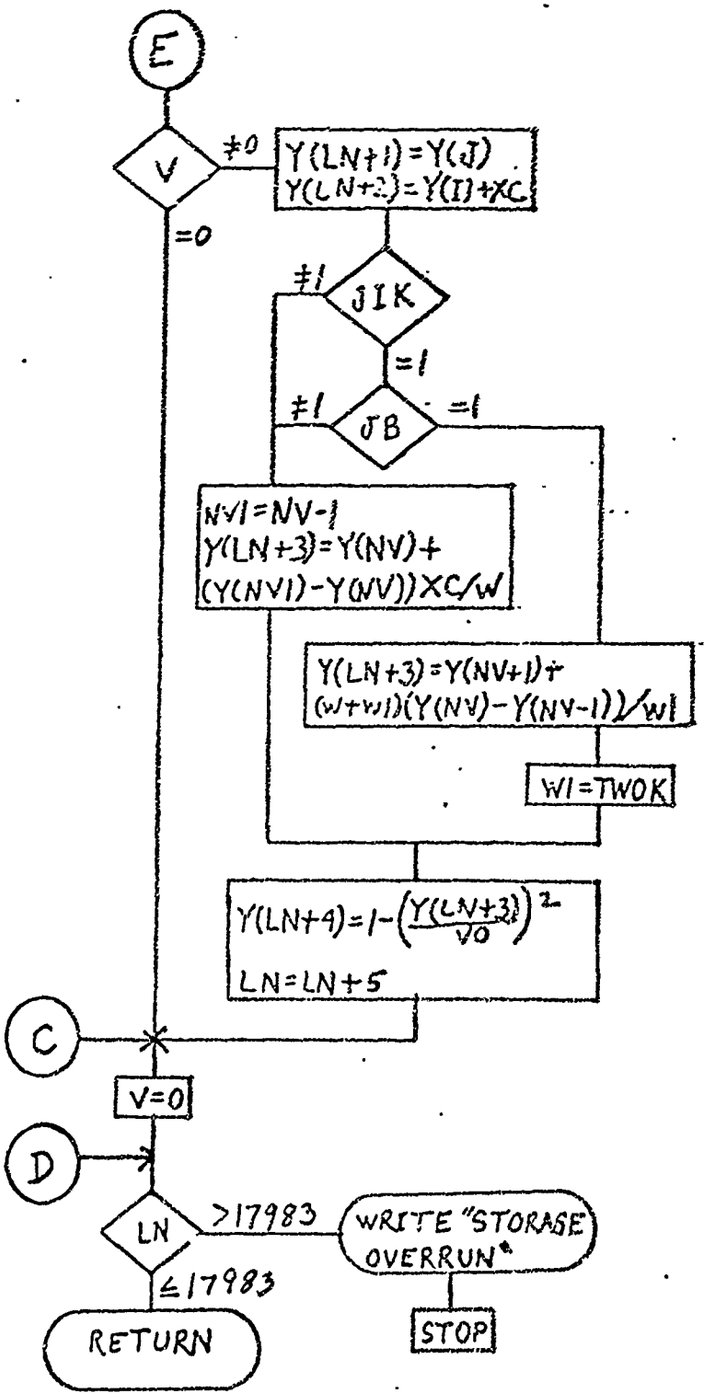


SUBROUTINE SGSRCH



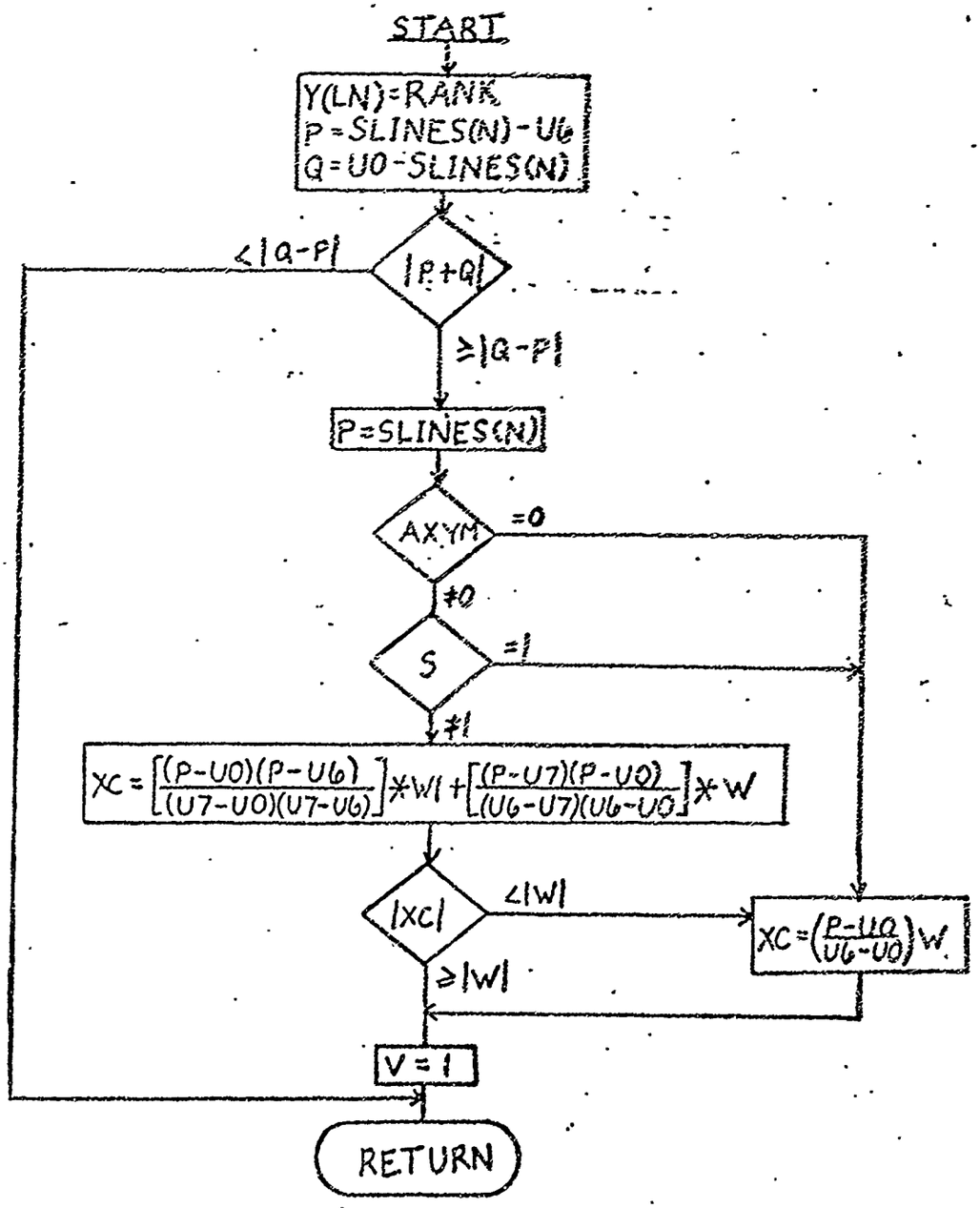

A
V
Y(LN+1)=Y(J)
Y(LN+2)=Y(I)+XC
≠1
KIJ
=1
≠1
JB
=1
NV1=NV+1
Y(LN+3)=Y(NV)+
(Y(NV1)-Y(NV))(XC)/W
W=TWOK
Y(LN+3)=(W+W1)(Y(NV)-
Y(NV+1))/W1 +Y(NV+1)
Y(LN+4)=1-(Y(LN+3)/VO)^2
LN=LN+5
B
V=0
CSCH
≥3
C
<3
JIK
=0
D
≠0
W=ONEK
W=-TWOK
U6=U2
U7=U4
SUB58
E

SEARCH 2

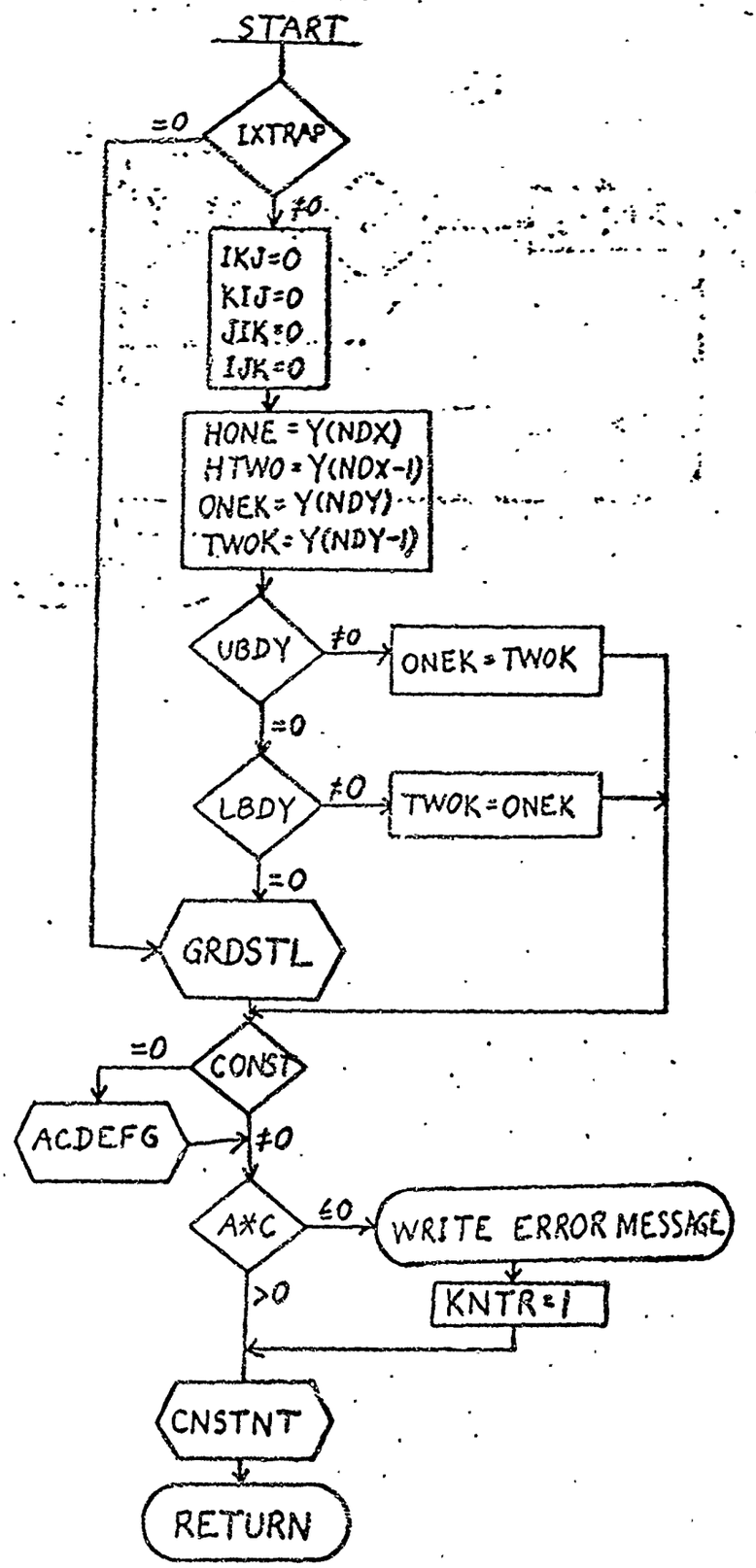

E
V
≠0
=0
Y(LN+1)=Y(J)
Y(LN+2)=Y(I)+XC
JIK
≠1
=1
JB
≠1
=1
NV1=NV-1
Y(LN+3)=Y(NV)+
(Y(NV1)-Y(NV))XC/W
Y(LN+3)=Y(NV+1)+
(W+W1)(Y(NV)-Y(NV-1))/W1
W1=TWOK
Y(LN+4)=1-(Y(LN+3)/VO)²
LN=LN+5
C
V=0
D
LN
>17983
WRITE "STORAGE OVERRUN"
≤17983
RETURN
STOP

SEARCH 3

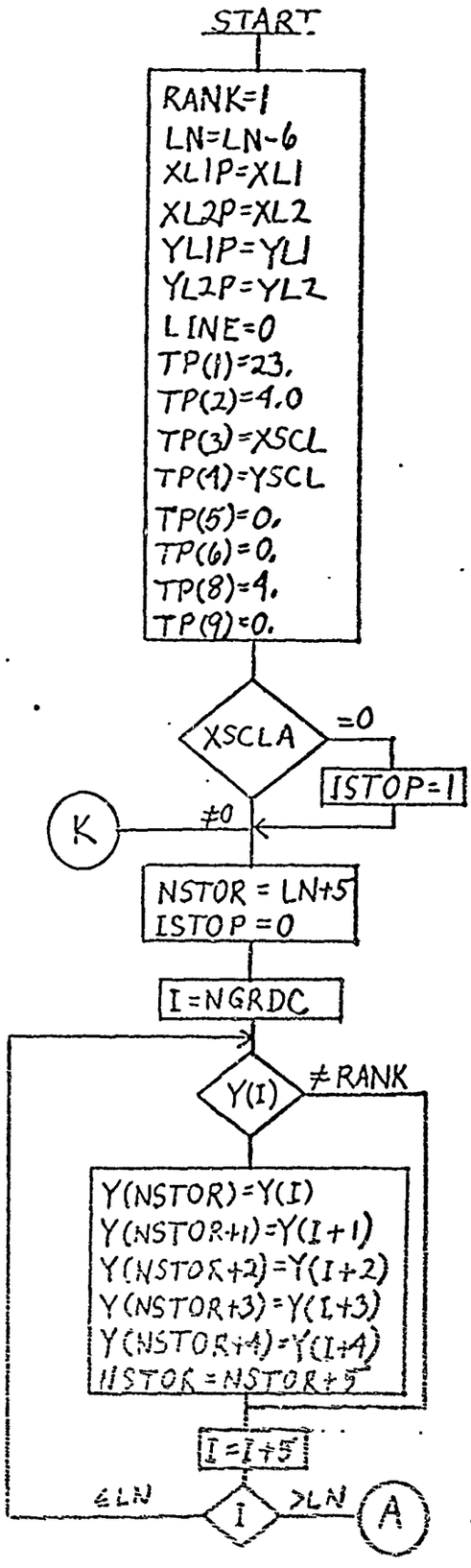
SUBROUTINE SUB58

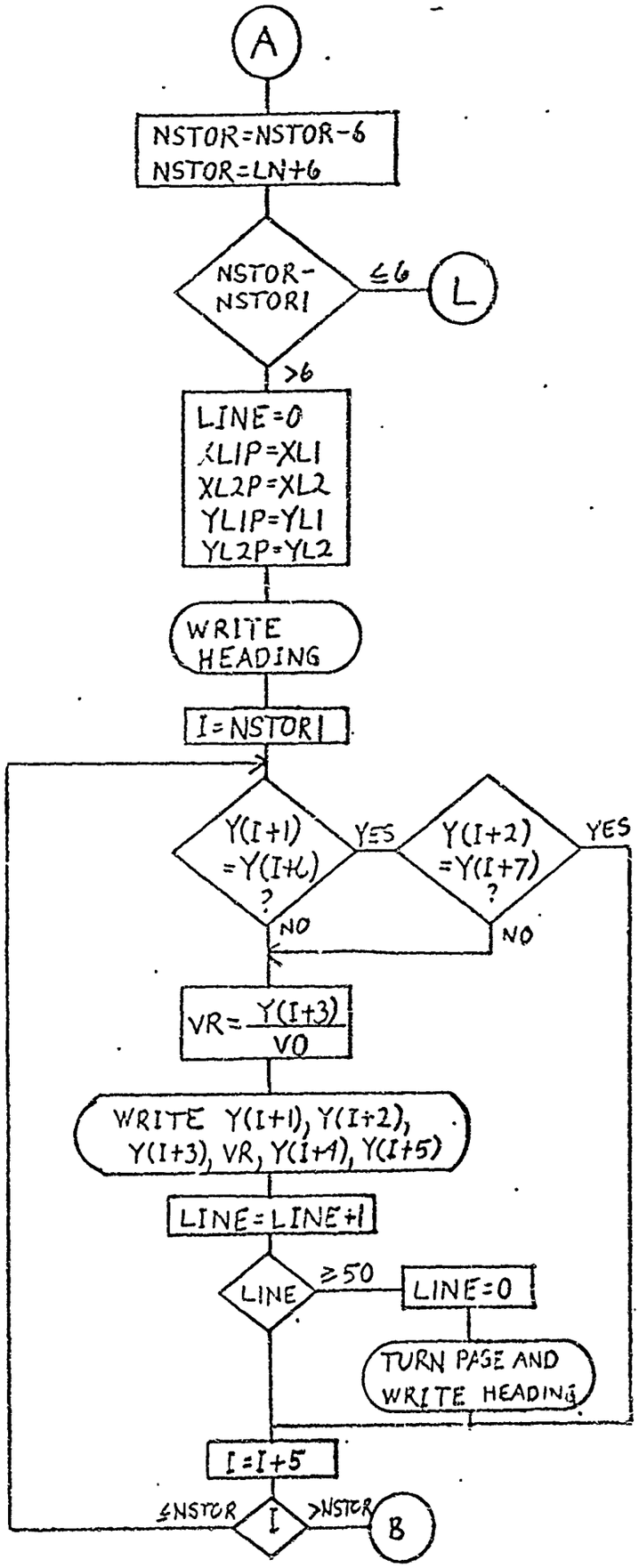


SUBROUTINE REGION

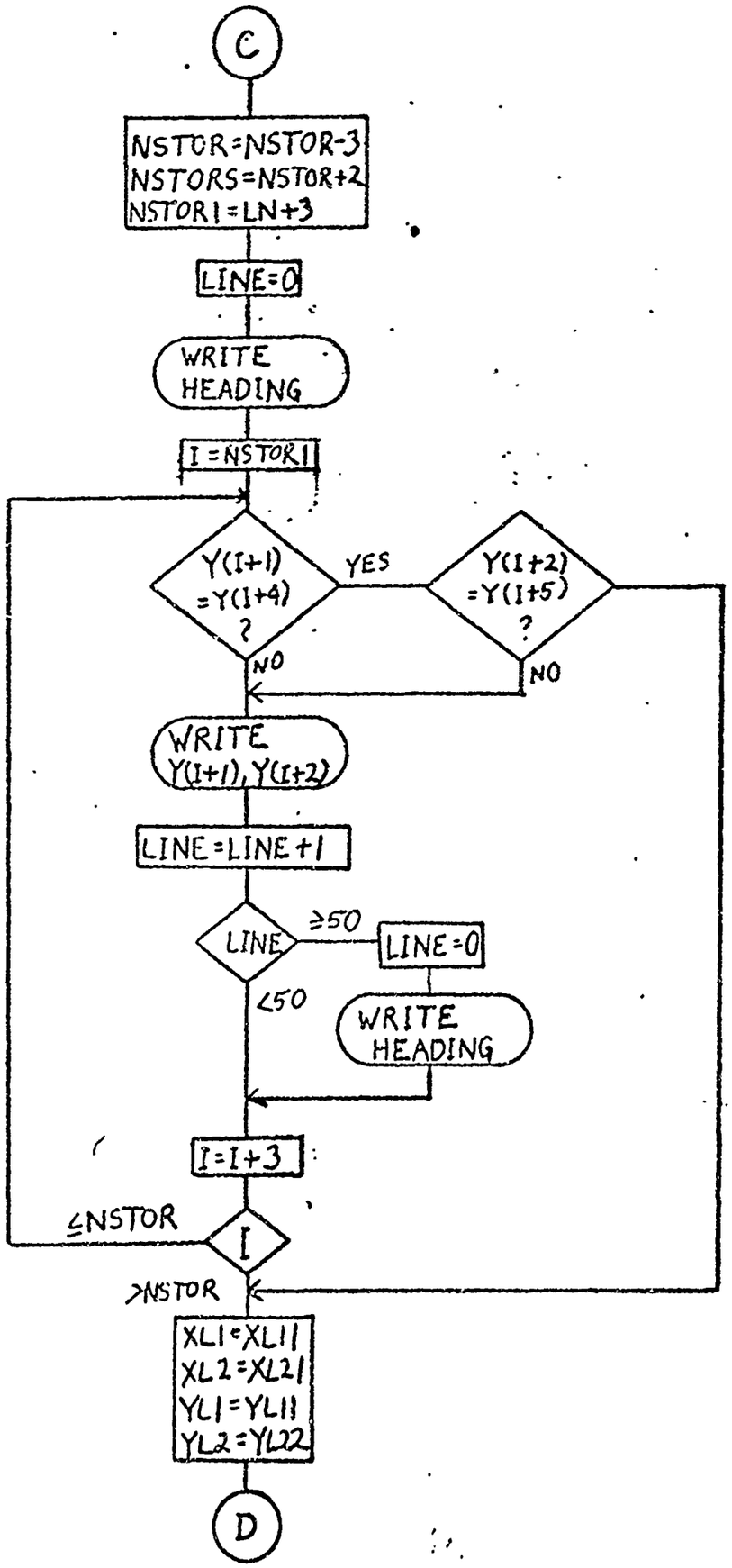


SUBROUTINE SORT

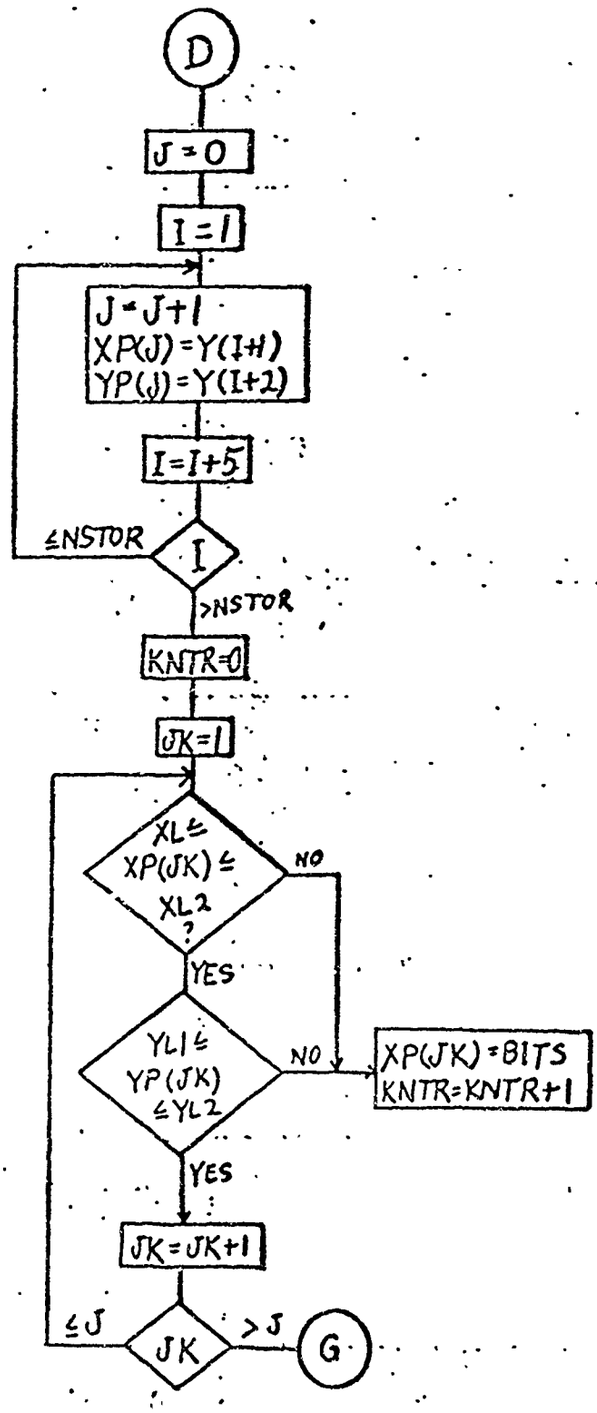



A
NSTOR=NSTOR-6
NSTOR=LN+6
NSTOR-NSTOR1
≤6
L
>6
LINE=0
XL1P=XL1
XL2P=XL2
YL1P=YL1
YL2P=YL2
WRITE HEADING
I=NSTOR1
Y(I+1) =Y(I+6) ?
YES
Y(I+2) =Y(I+7) ?
YES
NO
NO
VR= Y(I+3) / V0
WRITE Y(I+1), Y(I+2), Y(I+3), VR, Y(I+4), Y(I+5)
LINE=LINE+1
LINE
≥50
LINE=0
TURN PAGE AND WRITE HEADING
I=I+5
≤NSTOR
I
>NSTOR
B

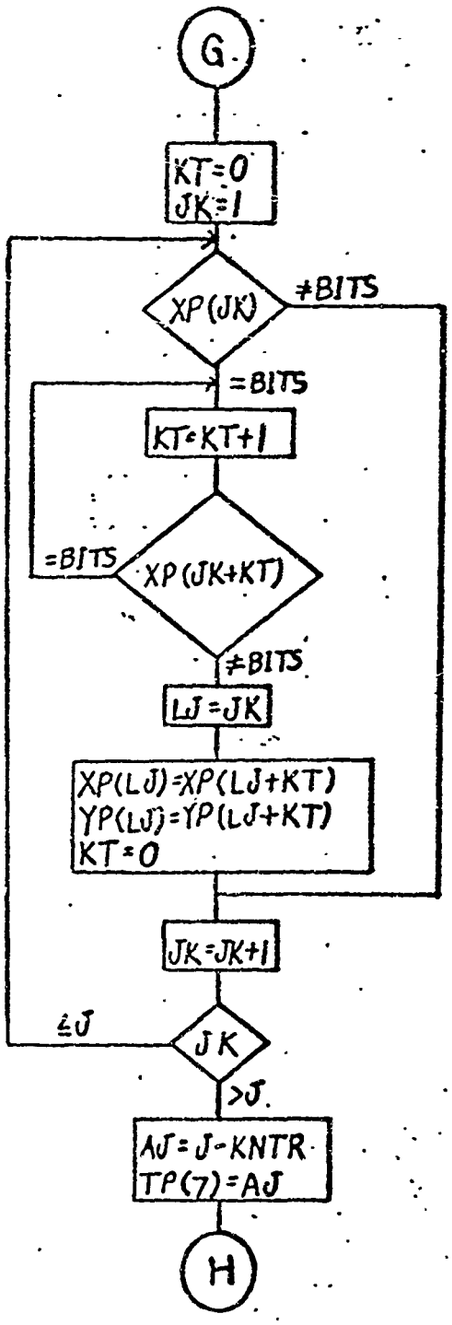
SORT 2


C
NSTOR=NSTOR-3
NSTORS=NSTOR+2
NSTOR1=LN+3
LINE=0
WRITE HEADING
I=NSTOR1
Y(I+1) =Y(I+4) ?
YES
Y(I+2) =Y(I+5) ?
NO
NO
WRITE Y(I+1), Y(I+2)
LINE=LINE+1
LINE
≥50
LINE=0
<50
WRITE HEADING
I=I+3
≤NSTOR
I
>NSTOR
XL1=XL11
XL2=XL21
YL1=YL11
YL2=YL22
D

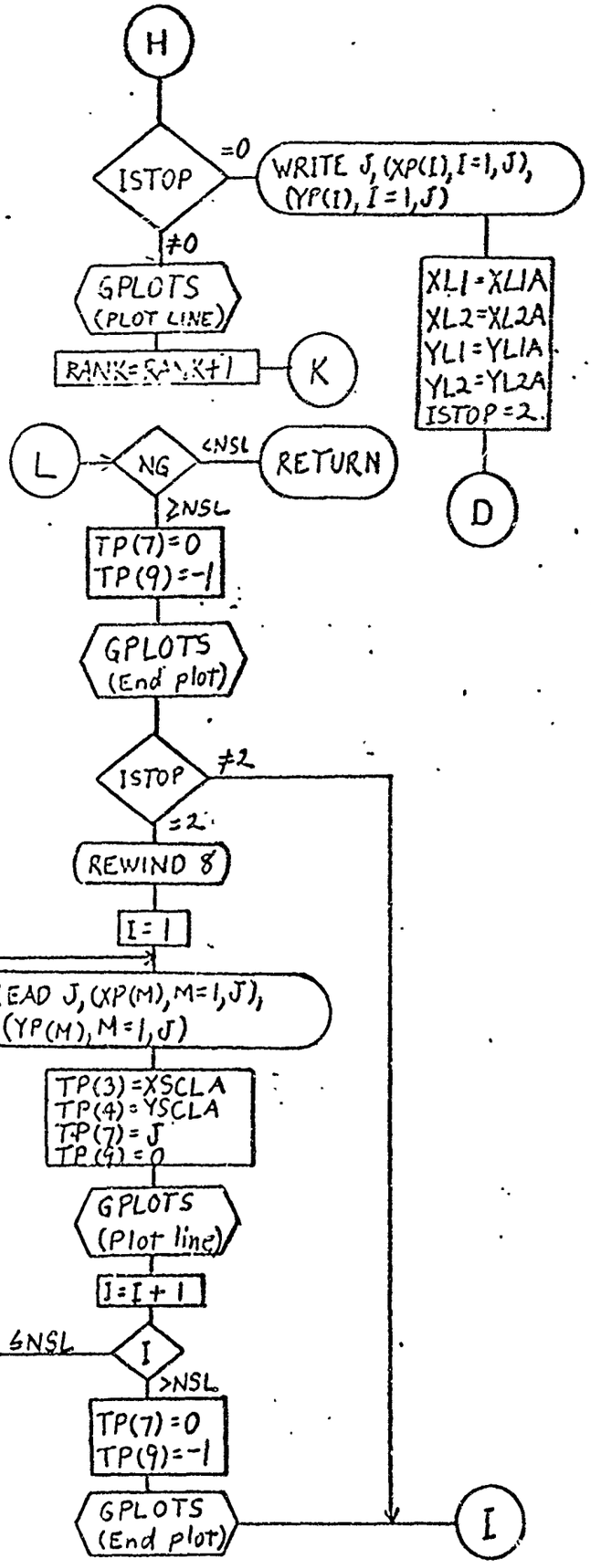
SORT 3


D
J = 0
I = 1
J = J+1
XP(J) = Y(I+1)
YP(J) = Y(I+2)
I = I+5
≤NSTOR
I
>NSTOR
KNTR=0
JK=1
XL ≤ XP(JK) ≤ XL2 ?
NO
YES
YL1 ≤ YP(JK) ≤ YL2
NO
XP(JK) = BITS
KNTR=KNTR+1
YES
JK = JK+1
≤J
JK
>J
G

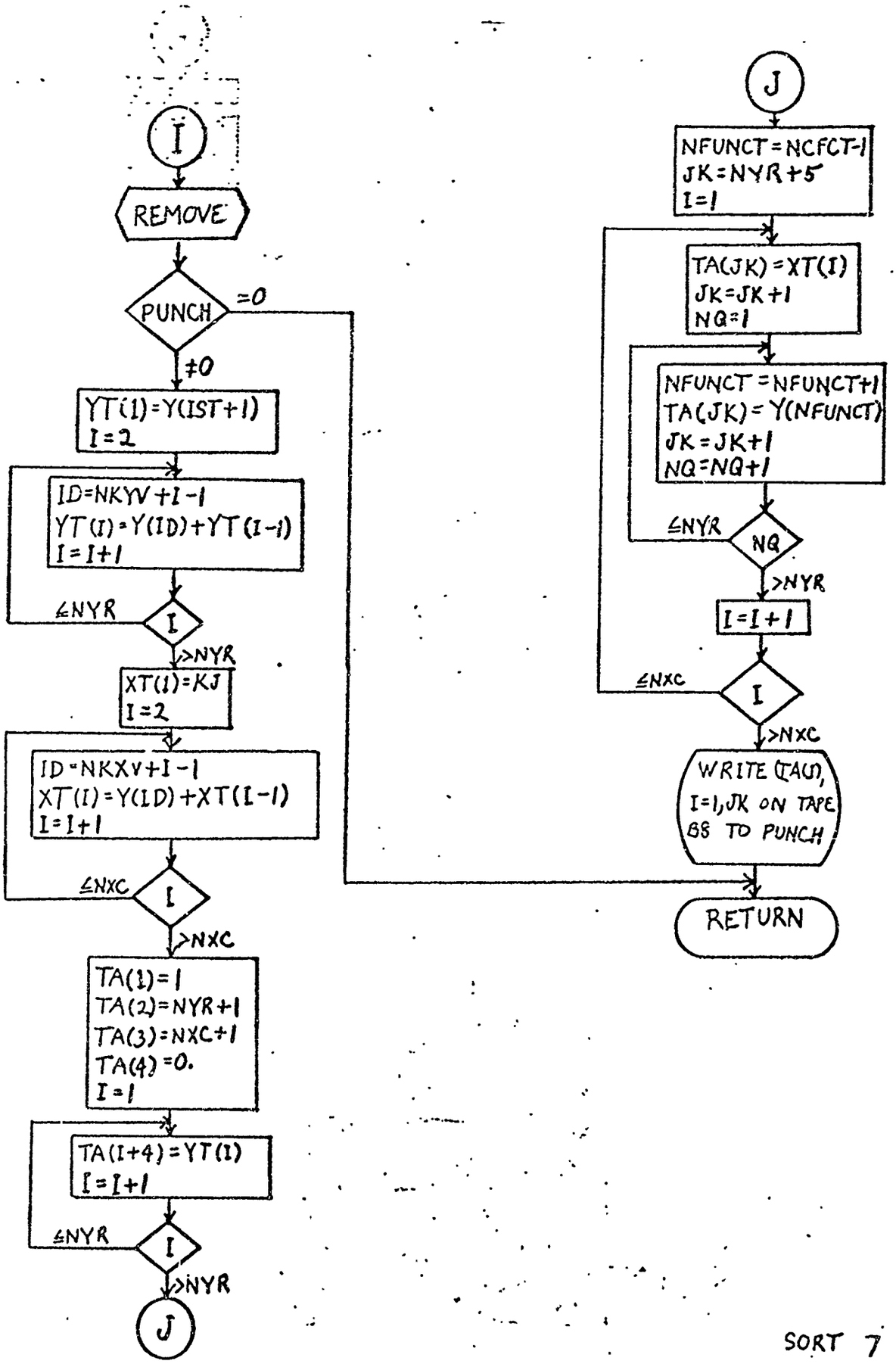
SORT 4



REVISED
E 1686 R8


H
ISTOP
=0
WRITE J, (XP(I), I=1,J), (YP(I), I=1,J)
≠0
GPLOTS (PLOT LINE)
RANK=RANK+1
K
XL1=XL1A
XL2=XL2A
YL1=YL1A
YL2=YL2A
ISTOP=2.
D
L
NG
<NSL
RETURN
≥NSL
TP(7)=0
TP(9)=-1
GPLOTS (End plot)
ISTOP
≠2
=2
REWIND 8
I=1
READ J, (XP(M), M=1,J), (YP(M), M=1,J)
TP(3)=XSCLA
TP(4)=YSCLA
TP(7)=J
TP(9)=0
GPLOTS (Plot line)
I=I+1
≤NSL
I
>NSL
TP(7)=0
TP(9)=-1
GPLOTS (End plot)
I

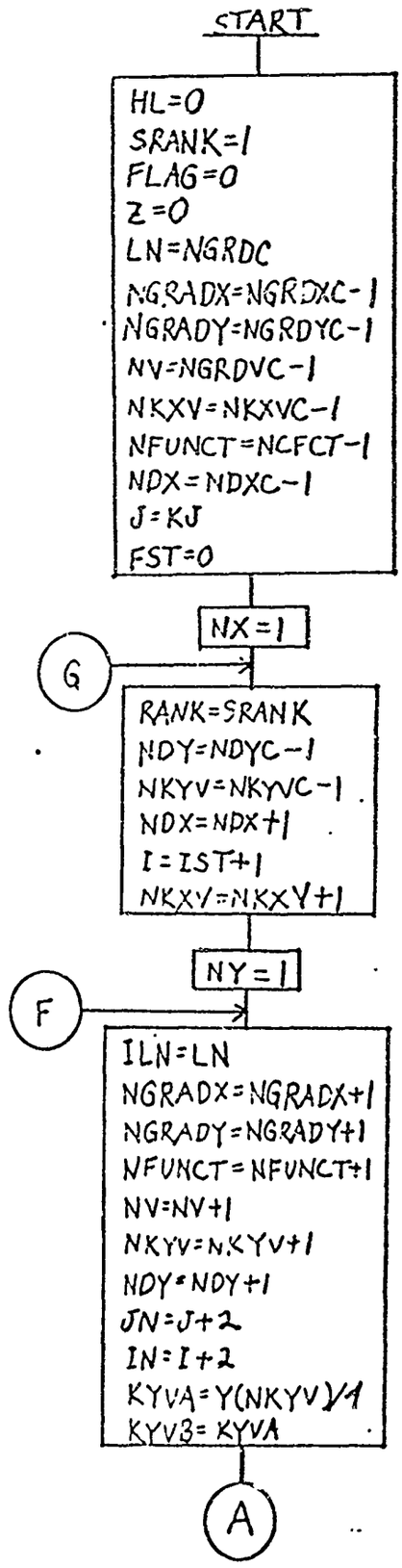
SORT 6

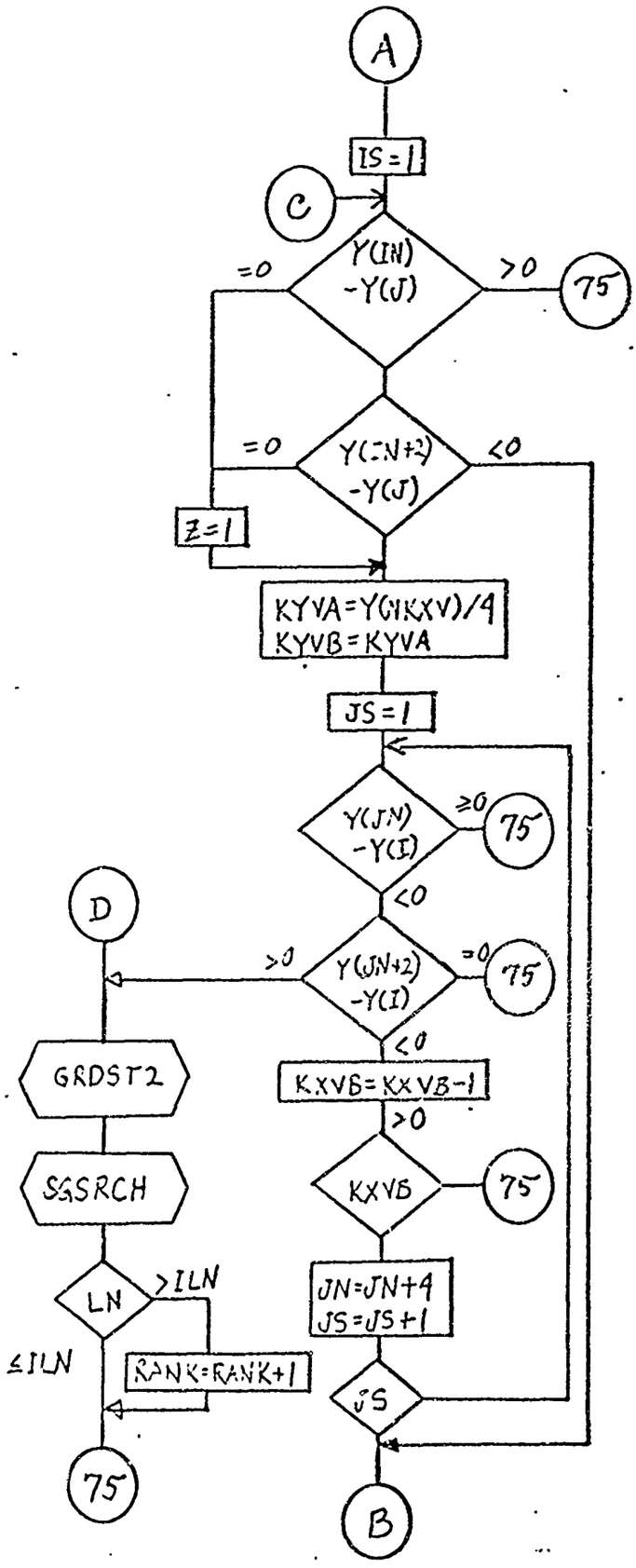


SORT 7

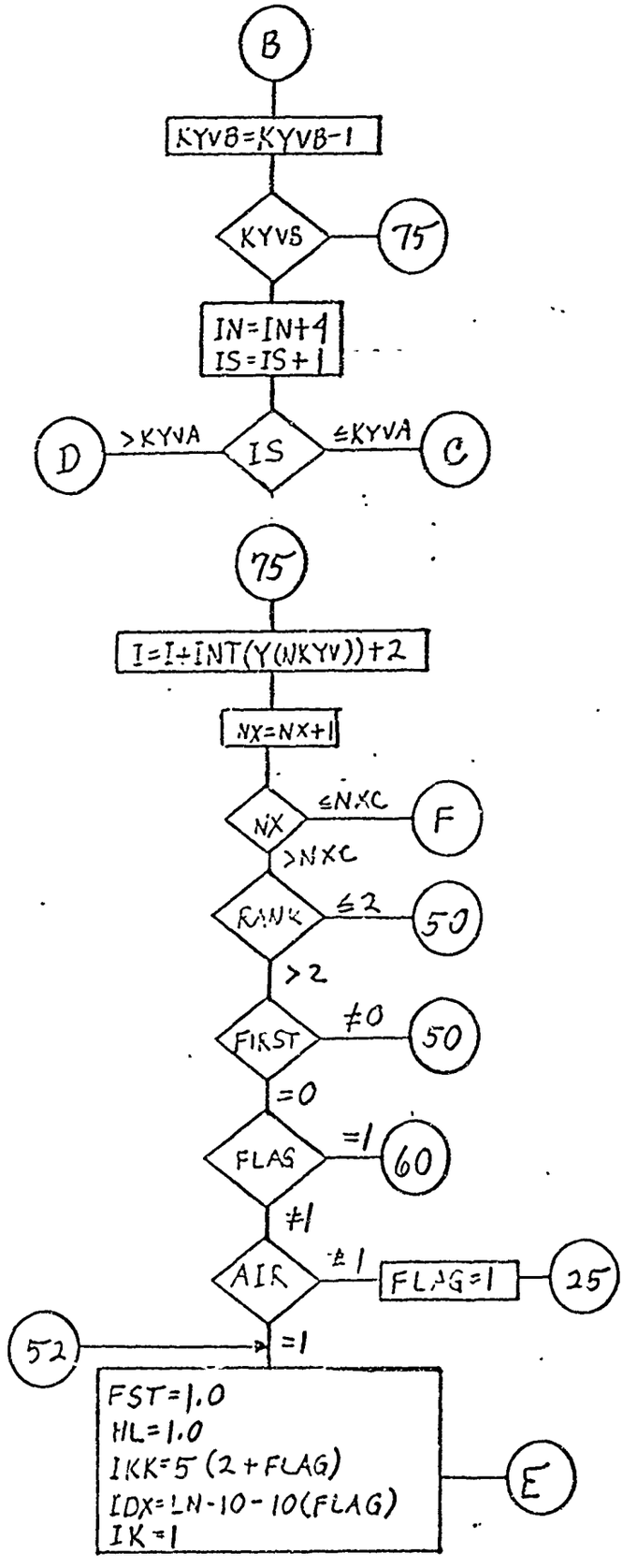
REVISED

SUBROUTINE MARC

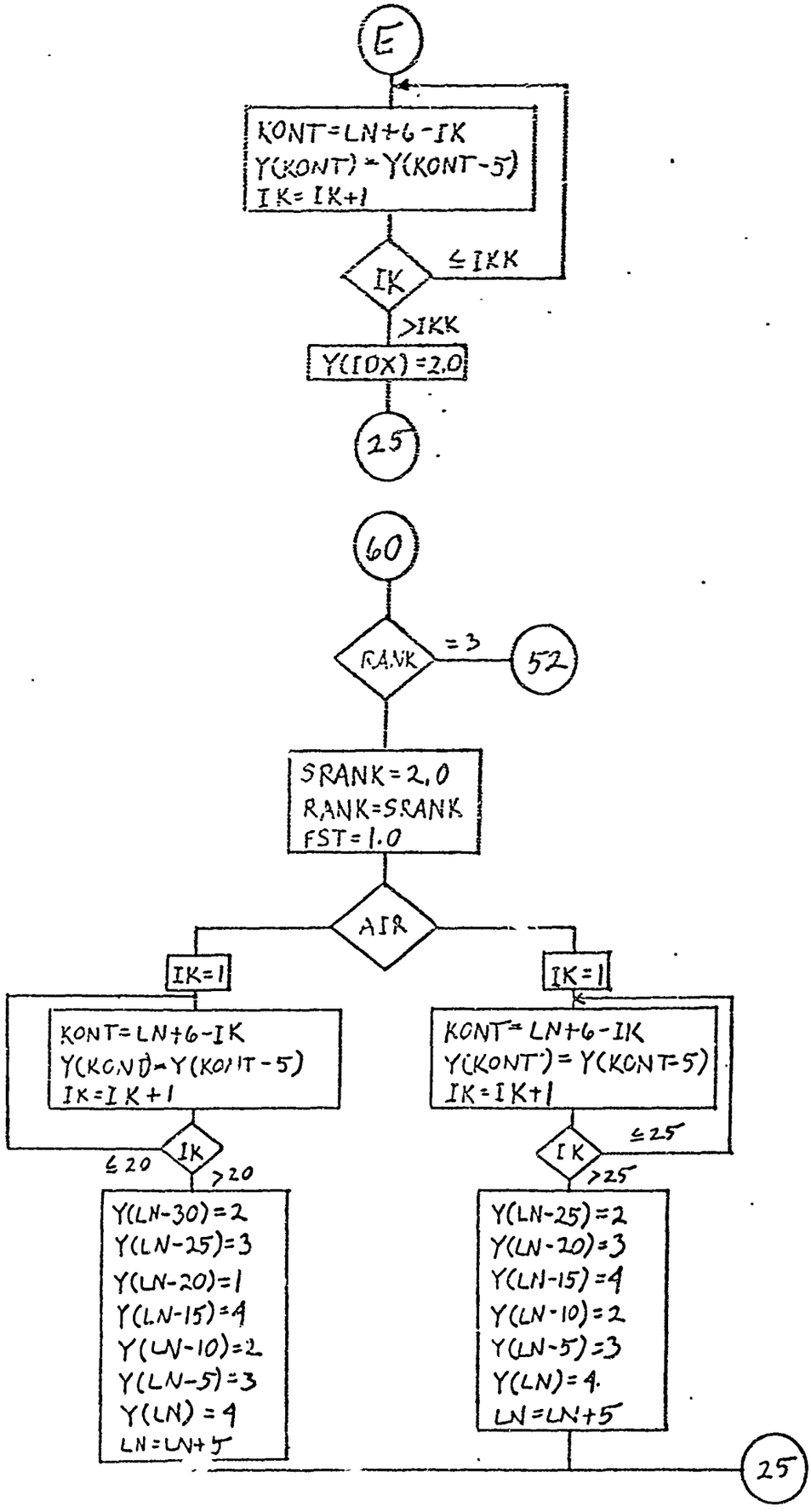




REVISED
E 1686 R8

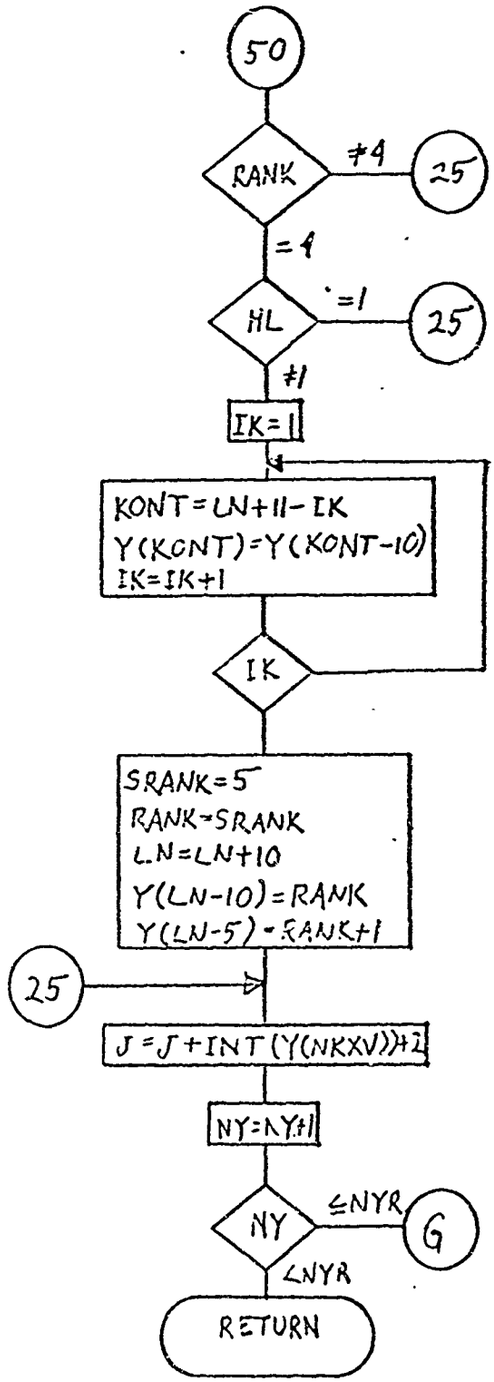

B
KYVB=KYVB-1
KYVB
75
IN=IN+4
IS=IS+1
IS
>KYVA
D
≤KYVA
C
75
I=I+INT(Y(NKYV))+2
NX=NX+1
NX
≤NXC
F
>NXC
RANK
≤2
50
>2
FIRST
≠0
50
=0
FLAG
=1
60
≠1
AIR
≠1
FLAG=1
25
52
=1
FST=1.0
HL=1.0
IKK=5 (2+FLAG)
IDX=LN-10-10(FLAG)
IK=1
E

MARC 3



MARC 4

REVISED



REVISED

MARC 5

10.2 Appendix 2 - Potential Flow Computer Program Listing

```
$IBFTC WAMAIN
C       MAIN PROGRAM
        DIMENSION B(1),Y(18000),SLINES(30)
        COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,
       1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y,     0(
       2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1,      0(
       3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,       0(
       4NGRADX,NGRADY,LN,CSCH,NG,NDERIV,SSRCH,NSWPS,OMEGA,CVGS,CONST,         0(
       5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,          0(
       6NLRCAL,Q,R,S,T,V,W,XC,YC,IH,NTAPE                                     0(
        COMMON /VEL/ NGRDVC
        COMMON /COM2/ XSCL,YSCL,PUNCH, ITLE,Y0,Y1,XL1,XL2,YL1,YL2,XSCLA,
       1YSCLA,XL1A,YL1A,XL2A,YL2A
        DIMENSION  ITLE(24)
        REWIND 23
      1 DO 10 I=1,18000
     10 Y(I)=0.
        S=1.
        PUNT = 0.                                                             0(
        HEDGE=.75                                                             0(
        NSWEEP=20                                                             0(
        FLIM=10.                                                              0(
        CALL INPUT
    902 CALL SEARCH(Y0,Y1,X,K1)                                               0(
C       THIS ROUTINE ALSO FINDS ERRORS IN THE INPUT DATA,GENERATES THE        0(
C       COEFFICIENTS OF THE DIFFERENCE ANALOGUE, AND ENFOFCES DERIVATIVES.    0(
        IF(KNTR)40,907,40
    907 CALL SWEEP(PUNT,NSWEEP,HEDGE,FLIM)                                    0(
        V=1.
        CALL SEARCH(Y0,Y1,X,K1)
        CALL VELOC
        V=-1.0
        CALL SEARCH (Y0,Y1,X,K1)
        CALL LINES
     40 STOP
        END
```

```
$IBFTC WALNK1
      SUBROUTINE INPUT
      DIMENSION Y(18000),SLINES(30),X(6)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y,      0(
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1,       0(
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,         0(
     4NGRADX,NGRADY,LN,CSCH,NG,NDERIV,SSRCH,NSWPS,OMEGA,CVGS,CONST,           0(
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,            0(
     6NLRCAL,Q,R,S,T,V,W,XC,YC,IH,NTAPE
      COMMON /COM2/ XSCL,YSCL,PUNCH, ITLE,Y0,Y1,XL1,XL2,YL1,YL2,XSCLA,
     1YSCLA,XL1A,YL1A,XL2A,YL2A
      COMMON /VEL/ NGRDVC
      DIMENSION  ITLE(24)
  501 FORMAT (6E12.6)
  502 FORMAT (6I12)
  503 FORMAT (12A6)
  600 FORMAT (7H AXYM =F2.0,11X,7HNDERIV=I2,  10X,5HXSCL=F8.3,4X,5HYSCL=
     1F8.3/   7H NSWPS=I6,   7X,7HY0    =F8.3, 4X,5HXL1 =F8.3,4X,5HYL1 =
     2F8.3/   7H CVGS =F9.7, 4X,7HY1    =F8.3, 4X,5HXL2 =F8.3,4X,5HYL2 =
     3F8.3/   7H PUNCH=F2.0,30X,6HXSCLA=F7.3,4X,6HYSCLA=F7.3/
     4     39X,5HXL1A=F8.3,4X,5HYL1A=F8.3/            39X5HXL2A=F8.3,4X,
     55HYL2A=F8.3//)
  610 FORMAT(1H1,12A6/1X,12A6///)
      XSCLA=0.
    2 A=1.                                                                    0(
      V=0.                                                                    0(
      CSCH=0.                                                                 0(
      N2=0                                                                    0(
      T=0.                                                                    0(
      C=1.                                                                    0(
      CONST=1.                                                                0(
      OMEGA=1.37
      STREAM = 0.                                                             0(
      RUN = 0.                                                                0(
      DUM1=0.                                                                 0(
      READ (5,503) ITLE
      WRITE (6,610)  ITLE
      READ (5,501) AXYM,CVGS,ANSWPS,PUNCH
      NSWPS=ANSWPS
      READ (5,501) ANDERV,Y0,Y1
      NDERIV=ANDERV
      READ (5,501) XSCL,YSCL,XL1,XL2,YL1,YL2
      READ (5,501) XSCLA,YSCLA,XL1A,XL2A,YL1A,YL2A
      IF (AXYM.EQ.1.) STREAM=-1.
      IF (AXYM.EQ.1.) CONST=0.
  900 CALL CHECK                                                              0(
      IF (KNTR.NE.0) STOP
      WRITE (6,600) AXYM,NDERIV,XSCL,YSCL,NSWPS,Y0,XL1,YL1,CVGS,Y1,XL2,
     1YL2,PUNCH,XSCLA,YSCLA,     XL1A,YL1A,    XL2A,YL2A
   42 RETURN
```

```
      END
60E
```

```
$IBFTC WACHCK
      SUBROUTINE CHECK                                                  0
C     THIS ROUTINE READS COORDINATE INPUT, CHECKS FOR ERRORS,AND PRINTS
C     COORDINATE INPUT.
      DIMENSION B(1),Y(18000),SLINES(30)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,   0
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y, 0
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1,  0
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,    0
     4NGRADX,NGRADY,LN,CSCH,NG,EXTRAP,SSRCH,NSWPS,OMEGA,CVGS,CONST,      0
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,       0
     6P,Q,R,S,T,V,W,XC,YC,IH,NTAPE                                       0
      COMMON /VEL/ NGRDVC
  227 FORMAT (4H Y1=F12.5/61H CHECK THE INPUT DATA CARD WHICH HAS THIS Y
     X AS ITS FIRST TERM/47H EITHER THE CONSTANT TERM IS NOT DIVISIBLE B
     XY 4/55H EXACTLY OR THE BOUNDARY COORDINATES ARE NOT MONOTONIC.//)
  228 FORMAT (4H Y1=F12.5/4H Y2=F12.5/41H THE COORDINATE VALUES ARE NOT
     XMONOTONIC.//)
  501 FORMAT (6E12.6)
  599 FORMAT (1H1)
  600 FORMAT (67H COORDINATES AND FUNCTION VALUES FOR BOUNDARY POINTS ON
     1 MESH ROWS  //8X,1HY,13X,1HN,13X,1HX,12X,6HF(X,Y),10X,1HX,11X,6HF(
     2X,Y)//)
  601 FORMAT (68H1COORDINATES AND FUNCTION VALUES FOR BOUNDARY POINTS ON
     1 MESH COLUMNS//8X,1HX,13X,1HN,13X,1HY,12X,6HF(X,Y),10X,1HY,11X,6HF
     2(X,Y)//)
  602 FORMAT (1X,6F14.7,/(29X,4F14.7))
  620 FORMAT (1X,6F14.7)
  686 FORMAT (1X,4I10)
      LINE=5
      KNTR = 0                                                          0
      IST=600
      I = IST+1                                                         0
C     Y(I)  IS THE LOCATION OF THE FIRST Y INPUT VALUE.                 0
      NKYV=1                                                            0
      NKYVC=NKYV                                                        0
C     NKYV AND NKXV GIVE THE BEGINNING OF KYV AND KXV                   0
C     IN Y ARRAY. THESE TWO VECTORS TELL HOW MANY BOUNDARY              0
C     VALUES ARE CONNECTED WITH EACH Y AND X COORDINATE.                0
      NDY=150
      NKXV=300
      NDX=450
      KNT=1
      WRITE (6,600)
      IE=I+5
      READ (5,501)(Y(IR),IR=I,IE)
      L = Y(I+1)                                                        0
      Y(NKYV)=L                                                         0.
   40 XL=L
      IF (AMOD(XL,4.))240,44,240
  240 WRITE (6,227) Y(I)
      KNTR = 1                                                          0
```

```
   46 LN = L-2                                                          0
C     LOOP  5 CHECKS MONOTONICITY OF X BOUNDARY COORDINATES              0
      DO 5  M = 1,LN,2                                                   0
      IF (M.EQ.1) GO TO 52
      XM=M+1
      IF (AMOD(XM,4.) .NE.0.) G  TO 52
      IB=J+M+3
      IE=IB+3
      READ (5,501) (Y(IR),IR=IB,IE)
   52 IN=J+M-1
      IF(Y(IN+4)-Y(IN+2))245,5,5                                         0
  245 WRITE (6,227) Y(J)
      KNTR = 1                                                           0.
    5 CONTINUE                                                           0
      LQ=J+INT(Y(J+1))+1
      WRITE (6,602)(Y(L4),L4=J,LQ)
      LINE=LINE+1
      IF (LINE.LT.25) GO TO 62
      LINE=0
      WRITE (6,601)
   62 MIT=J+L+2                                                          0.
      IE=MIT+5
      READ (5,501) (Y(IR),IR=MIT,IE)
      IF (Y(MIT+1).EQ.0.) GO TO 285
      KNT=KNT+1
      Y(NDX)=Y(MIT)-Y(J)                                                 0.
      IF(Y(NDX))243,8,8                                                  0.
C     IF STATEMENT CHECKS MONOTONICITY OF X COORDINATES                  0
  243 WRITE (6,228) Y(J),Y(MIT)
      KNTR = 1                                                           0
    8 NKXV = NKXV + 1                                                    0
      J=MIT                                                              0.
      L=Y(J+1)                                                           0.
      NDX = NDX + 1                                                      0
   26 Y(NKXV)=L                                                          0.
      GO TO 260
  285 NXC=KNT
      DO 290 IQ=1,NYR
      IPT=NYR+IQ
      Y(IPT)=Y(IQ+149)
  290 CONTINUE
      DO 291 IQ=1,NXC
      IPT=2*NYR+IQ
      Y(IPT)=Y(IQ+299)
      IPT=2*NYR+NXC+IQ
      Y(IPT)=Y(IQ+449)
  291 CONTINUE
      IST=2*(NXC+NYR)                                                    0
      NDY=NYR+1                                                          0
      NDYC=NDY                                                           0
C     NDYC GIVES LOCATION OF FIRST Y MESH VALUE IN Y ARRAY.              0
      NKXV=NDY+NYR                                                       0
```

```
   44 LN = L-2                                                          0
C     LOOP 16 CHECKS MONOTONICITY OF Y BOUNDARY COORDINATES              0
      DO 16 M=1,LN,2                                                     0
      IF (M.EQ.1) GO TO 51
      XM=M+1
      IF (AMOD(XM,4.) .NE.0.) GO TO 51
      IB=I+M+3
      IE=IB+3
      READ (5,501) (Y(IR),IR=IB,IE)
   51 IN = I+M-1
      IF(Y(IN+4)-Y(IN+2))241,16,16                                       0
  241 WRITE (6,227) Y(I)
      KNTR = 1                                                           0
   16 CONTINUE                                                           0
      LQ=I+INT(Y(I+1))+1
      WRITE (6,602)(Y(L4),L4=I,LQ)
      LINE=LINE+1
      IF (LINE.LT.25) GO TO 61
      LINE=0
      WRITE (6,599)
      WRITE (6,600)
   61 MIT = I+L+2                                                        0
      IE=MIT+5
      READ (5,501)  (Y(IR),IR=MIT,IE)
      Y(NDY)=Y(MIT)-Y(I)                                                 0
      IF (Y(MIT+1).EQ.0.) GO TO 280
      KNT=KNT+1
      IF(Y(NDY))244,24,24                                                0
C     IF'STATEMENT CHECKS MONOTONICITY OF Y COORDINATES                  0
  244 WRITE (6,228) Y(I),Y(MIT)
      KNTR = 1                                                           0
   24 NKYV=NKYV+1
      I=MIT                                                              0
      L=Y(I+1)                                                           0
      NDY = NDY + 1                                                      0
    4 Y(NKYV)=L                                                          0
      GO TO 40
  280 NYR=KNT
      KNT=1
      J=2+L+I                                                            0
      KJ=J                                                               0
C     Y(KJ) IS THE LOCATION OF THE FIRST X INPUT VALUE.                  0
      LINE=0
      IE=J+5
      READ (5,501) (Y(IR),IR=J,IE)
      L=Y(J+1)                                                           0
      Y(NKXV)=L                                                          0
      WRITE (6,601)
  260 XL=L
      IF (AMOD(XL,4.))242,46,242
  242 WRITE (6,227) Y(J)
      KNTR = 1                                                           0
```

```
      NKXVC=NKXV                                                        0.     $I
      NDX=NKXV+NXC                                                      0
      NDXC=NDX                                                          0      C
C     NDXC GIVES LOCATION OF FIRST X MESH VALUE IN Y ARRAY.             0      C
      DO 295 I=1,4000                                                          C
      ID=IST+I
      Y(ID)=Y(I+600)
  295 CONTINUE
      KJ=KJ-600+IST
      NCFCT=MIT-600+IST
C     Y (NCFCT) IS THE FIRST FUNCTION VALUE LOCATION.                   0
      NBETAC=NCFCT+(NXC*NYR)                                            0
C     Y(NBETAC)IS THE FIRST CONSTANT VALUE LOCATION.                    0
      MK=NXC*NYR
      NGRDVC=NBETAC
      NGRDXC=NBETAC+MK
      NGRDYC=NGRDXC+MK
      NGRDC=NGRDYC+MK
      IF (LINE.GT.22) WRITE (6,599)
   59 RETURN
      END                                                               0
```

C

```
$IBFTC WASRCH
      SUBROUTINE SEARCH(Y0,Y1,X,K1)                                     0
C     THIS ROUTINE PLACES THE INPUT BOUNDARY VALUES IN THE PSI ARRAY
C     WHICH BEGINS WITH Y(NFUNCT). IT CHECKS FOR INCONSISTENCY IN INPUT
C     AND CALLS REGION FOR EACH MESH POINT INSIDE THE REGION.
      DIMENSION B(1),Y(18000),SLINES(30)
      DIMENSION X(6)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,   0
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y,  0
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1,   0
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,    0
     4NGRADX,NGRADY,LN,CSCH,NG,IXTRAP,SSRCH,NSWPS,OMEGA,CVGS,CONST,      0
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,       0
     6P,Q,R,S,T,V,W,XC,YC,IH,NTAPE                                       0
      COMMON UBDY,LBDY                                                   0
  225 FORMAT (4H Y1=F12.6/4H Y2=F12.6/56H THE POINT (Y1,Y2) HAS ONE COOR
     XDINATE OUTSIDE THE REGION//)
  620 FORMAT (1X,6F14.7)
  621 FORMAT (3I24)
  625 FORMAT (120H THIS PROBLEM IS TOO BIG. ELIMINATE AS MANY OF THE MES
     1H POINTS FOR WHICH PSI VALUES ARE CALCULATED AS THERE ARE OF SUCH-
     2/26H POINTS TO THE RIGHT OF X=F14.7)
  626 FORMAT (8F12.5)
  630 FORMAT (5F15.4)
  631 FORMAT (1H1)
      IF (CONST.EQ.1.)GO TO 8
      A=1.
      C=1.
      D=0.
      G=0.
      F=0.
    8 BITS=-2.**15
      UBDY=0.                                                            0
      LBDY=0                                                             0
      KNTR = 0                                                           0
      IT=NBETAC                                                          0
      NKXV=NKXVC-1                                                       0
      NFUNCT=NCFCT-1                                                     0
      NDX=NDXC-1                                                         0
      J=KJ                                                               0
C     LOOPS 75 AND 76 PERMIT CHECKING POINTS BY COLUMN                   0
      DO 76 NX=1,NXC                                                     0
      NDY=NDYC-1                                                         0
      NKYV=NKYVC-1                                                       0
      NKXV=NKXV+1                                                        0
    9 NDX=NDX+1                                                          0
      XC=Y(J)                                                            0
      I = IST+1                                                          0
      DO 755 NY=1,NYR
      NDY=NDY+1                                                          0
      NKYV=NKYV+1                                                        0
      NFUNCT=NFUNCT+1                                                    0
```

```
      YC=Y(I)                                                           0
      JN=J+2                                                            0
      IN=I+2                                                            0
      KYVA=Y(NKYV)/4.                                                   0
      KYVB = KYVA                                                       0
C     LOOPS 10 AND 112 CHECK ALL POINTS BETWEEN BOUNDARY COORDINATES    0
C     TO DETERMINE THEIR LOCATION IN THE FIELD                          0
      DO 10 IS=1,KYVA                                                   0
C     ENTRY PERMITS TRANFER OF BOUNDARY MESH POINTS TO FUNCTION ARRAY   0
      ENTRY=Y(IN+1)                                                     0
      IF(Y(IN)-Y(J))1,2,3                                               0
C     Y(IN) IS THE LOCATION OF THE LEFT X BOUNDARY COORDINATE           0
    1 ENTRY=Y(IN+3)                                                     0
      IF(Y(IN+2)-Y(J))110,2,7                                           0
    7 KXVA=Y(NKXV)/4.                                                   0
      KXVB=KXVA                                                         0
      DO 112 JS=1,KXVA                                                  0
      ENTRY=Y(JN+1)                                                     0
      IF(Y(JN)-Y(I))11,2,33                                             0
   11 ENTRY=Y(JN+3)                                                     0
      IF(Y(JN+2)-Y(I))12,2,13                                           0
C     SUBROUTINE REGION GENERATES MESH DISTANCES THAT                   0
C     ARE NOT FOUND IN THE CHECK SUBROUTINE. THESE ARE                  0
C     THE SHORT DISTANCES TO THE BOUNDARY.                              0
   13 IF (V) 75,113,75                                                  0
  113 CALL REGION                                                       0
      GO TO 75                                                          0
   12 KXVB=KXVB-1                                                       0
      IF(KXVB) 33,33,112                                                0
  112 JN=JN+4                                                           0
  110 KYVB = KYVB-1                                                     0
      IF (KYVB) 3,3,10                                                  0
   10 IN=IN+4                                                           0
C     Y(JN) IS THE LOCATION OF THE LEFT Y BOUNDARY COORDINATE           0
C     IF A POINT IS IN THE FIELD IN THE X-DIRECTION                     0
C     AND OUT IN THE Y-DIRECTION, THE ROUTINE GOES                      0
C     TO LOCATION 750 INDICATING INCORRECT INPUT DATA.                  0
    3 IF (Y(JN)-Y(I))101,750,333                                        0
  101 IF (Y(JN+2)-Y(I))333,750,750                                      0
  750 WRITE (6,225) Y(I),Y(J)
      KNTR = 1                                                          0
      GO TO 75                                                          0
C     MINUS ALL BITS ARE STORED IN COEFFICIENT ARRAY FOR                0
C     POINTS WHICH ARE NOT IN THE REGION DEFINED.                       0
   33 IF (Y(IN)-Y(J))171,750,333                                        0
  171 IF (Y(IN+2)-Y(J))333,750,750                                      0
  333 IF (V) 400,633,632                                                0
  633 Y(IT)=BITS                                                        0
      IT=IT+1                                                           0
      GO TO 75                                                          0
  400 IF (Y(JN).GT.Y(I)) Y(NFUNCT)=Y(JN+1)
      IF (Y(JN+2).LT.Y(I)) Y(NFUNCT)=Y(JN+3)
```

```
      GO TO 75
  632 Y(NFUNCT)=BITS                                                    0
      GO TO 75                                                          0
    2 IF (V) 75,22,75                                                   0
   22 ASSIGN 2222 TO L1                                                 0
      IF (IXTRAP)222,232,222                                            0
  232 CONTINUE
 2202 Y(NFUNCT)=ENTRY                                                   0
      GO TO L1,(2222,75)                                                0
 2222 Y(IT)=BITS                                                        0
      IT =IT+1                                                          0
      GO TO 75                                                          0
C     CHECK HERE TO FIND WHERE DERIVATIVES ARE TO BE ENFORCED           0
  222 IF (Y(I)-Y0)200,15,200                                            0
   15 IF (Y(J)-Y(IN))25,232,25                                          0
   25 IF (Y(J)-Y(IN+2))123,232,123                                      0
  200 IF (Y(I)-Y1)232,321,232                                           0
  321 KN=1                                                              0
      DO 201 K=1,K1                                                     0
      IF (Y(J)-X(KN))201,232,204                                        0
  204 IF (Y(J)-X(KN+1))206,232,201                                      0
  201 KN=KN+2                                                           0
  206 UBDY=1.                                                           0
      GO TO 207                                                         0
  123 LBDY=1.                                                           0
  207 CALL REGION                                                       0
      ASSIGN 75 TO L1                                                   0
      GO TO 2202                                                        0
   75 IF (IT.GT.18000) GO TO 320
  755 I=I+INT(Y(NKYV))+2                                              571
   76 J=J+INT(Y(NKXV))+2
      IF (V.NE.0.) GO TO 540
      IT=IT+1
      NFUNCT=NCFCT
      DO 310 NY=1,NYR
      NHALF=NXC/2
      DO 300 MC=1,NHALF
      IF (Y(NFUNCT).NE.0.)GO TO 300
      Y(NFUNCT)=Y(I+3)
  300 NFUNCT=NFUNCT+NYR
      NHALF=NHALF+1
      DO 305 MC=NHALF,NXC
      IF (Y(NFUNCT).NE.0.)GO TO 305
      IPNP1=I+INT(Y(I+1))+1
      Y(NFUNCT)=Y(IPNP1)
  305 NFUNCT=NFUNCT+NYR
      NFUNCT=NCFCT+NY
  310 I=I+INT(Y(I+1 ))+2
  540 RETURN                                                            0
  320 WRITE (6,625) Y(J)
      STOP
      END                                                               0
```

```
$IBFTC HAGRDS
C     THIS ROUTINE GETS PSI VALUES OF THE NEAREST POINTS IN FOUR
C     DIRECTIONS FROM THE POINT IN QUESTION AND THE DISTANCES TO THESE
C     POINTS
      DIMENSION B(1),Y(18000),SLINES(30)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y,
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1,
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,
     4NGRADX,NGRADY,LN,CSCH,NG,EXTRAP,SSRCH,NSWPS,OMEGA,CVGS,CONST,
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,
     6P,Q,R,S,T,V,W,XC,YC,IH,NTAPE
      N1 = NFUNCT+NYR
      N3=NFUNCT-NYR
      HONE=Y(NDX)
      HTWO=Y(NDX-1)
      ONEK=Y(NDY)
      TWOK=Y(NDY-1)
      U0=Y(NFUNCT)
      U1=Y(N1)
      U2=Y(NFUNCT+1)
      U3=Y(N3)
      U4=Y(NFUNCT-1)
      IJK=0
      IKJ=0
      KIJ=0
      JIK=0
      IF (Y(IN)-Y(J)+Y(NDX-1))51,51,150
  150 HTWO = Y(J)-Y(IN)
      U3=Y(IN+1)
      IJK=1
   51 IF(Y(IN+2)-Y(J)-Y(NDX))151,52,52
  151 HONE = Y(IN+2)-Y(J)
      U1=Y(IN+3)
      IKJ=1
   52 IF(Y(JN)-Y(I)+Y(NDY-1))53,53,152
  152 TWOK = Y(I)-Y(JN)
      U4=Y(JN+1)
      KIJ=1
   53 IF(Y(JN+2)-Y(I)-Y(NDY))153,540,540
  153 ONEK = Y(JN+2)-Y(I)
      U2=Y(JN+3)
      JIK=1
  540 RETURN
      END
```

```
$IBFTC WAACDE
      SUBROUTINE ACDEFG                                                 0
      DIMENSION B(1),Y(18000),SLINES(30)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,  0
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y, 0
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1, 0
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,   0
     4NGRADX,NGRADY,LN,CSCH,NG,EXTRAP,SSRCH,NSWPS,OMEGA,CVGS,CONST,     0
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,      0
     6NLRCAL,Q,R,S,T,V,W,XC,YC,IH,NTAPE                                 0
      IF (AXYM) 1,2,1                                                   0
    1 IF (Y(I)) 3,4,3                                                   0
    4 C=2.                                                              0
    2 E=0.                                                              0
      GOTO 7
    3 E=STREAM/Y(I)                                                     0
    7 RETURN                                                            0
      END                                                               0
```

```
$IBFTC WACNST
*  CNSTNT  SUBROUTINE FOR CALCULATING CONSTANTS OF DIFFFERENCE EQUATIONS    0
      SUBROUTINE CNSTNT                                                     0
C     THIS ROUTINE CALCULATES THE WEIGHTS TO BE GIVEN THE PSI VALUES OF
C     THE FOUR SURROUNDING POINTS IN THE RELAXATION. THESE CONSTANTS ARE
C     STORED BY SEARCH IN THE ARRAY THAT BEGINS WITH Y(NBETAC).
      DIMENSION B(1),Y(18000),SLINES(30)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,      0
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y,    0
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1,     0
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,       0
     4NGRADX,NGRADY,LN,CSCH,NG,IXTRAP,SSRCH,NSWPS,OMEGA,CVGS,CONST,         0
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,          0
     6P,Q,R,S,T,V,W,XC,YC,IH,NTAPE                                          0
      COMMON UBDY,LBDY                                                      0
C     THE CONSTANTS IKJ,JIK,IJK,KIJ ARE SET EQUAL TO 1 IF LESS THAN A       0
C     MESH DISTANCE AWAY FROM A BOUNDARY POINT.IN THIS CASE THE FUNCTION    0
C     VALUE AT THE                                                          0
C     BOUNDARY IS INCLUDED IN THE CONSTANT TERM Y(IT+4), AND THE            0
C     CORRESPONDING CONSTANT TERM IS SET EQUAL TO ZERO.                     0
  222 BA=(HTWO*D+2.*A)/(HONE*(HONE+HTWO))                                   0
      BB=(2.*C+TWOK*E)/(ONEK*(ONEK+TWOK))                                   0
      BC=(2.*A-HONE*D)/(HTWO*(HONE+HTWO))                                   0
      BD=(2.*C-ONEK*E)/(TWOK*(ONEK+TWOK))                                   0
      BE=-F+BA+BB+BC+BD                                                     0
      BE=1./BE                                                              0
      Y(IT+4)=-G*BE                                                         0
      IF(IKJ)202,201,202                                                    0
  202 BA = BA*Y(IN+3)*BE                                                    0
      Y(IT)=0.                                                              0
      Y(IT+4)=BA+Y(IT+4)                                                    0
      GO TO 210                                                             0
  201 Y(IT)=BA*BE                                                           0
  210 IF(JIK)204,203,204                                                    0
  204 BB = BB*Y(JN+3)*BE                                                    0
      Y(IT+1)=0.                                                            0
      Y(IT+4)=BB+Y(IT+4)                                                    0
      GO TO 211                                                             0
  203 Y(IT+1) =BB*BE                                                        0
  211 IF(IJK)206,205,206                                                    0
  206 BC=BC*Y(IN+1)*BE                                                      0
      Y(IT+2)=0.                                                            0
      Y(IT+4)=BC+Y(IT+4)                                                    0
      GO TO 212                                                             0
  205 Y(IT+2)=BC*BE                                                         0
  212 IF(KIJ)208,207,208                                                    0
  208 BD=BD*Y(JN+1)*BE                                                      0
      Y(IT+3)=0.                                                            0
      Y(IT+4)=BD+Y(IT+4)                                                    0
      GO TO 213                                                             0
  207 Y(IT+3)=BD*BE                                                         0
C     ANY DERIVATIVE ENFORCEMENT BEGINS WITH STATEMENT 213                  0
```

```
213 IF (IXTRAP) 217,227,217                                          0
217 IF (UBDY) 1,2,1                                                  0
  1 Y(IT+3)=Y(IT+3)+Y(IT+1)                                          0
    Y(IT+1)=0.                                                       0
    GO TO 225                                                        0
  2 IF (LBDY) 3,227,3                                                0
  3 Y(IT+1)=Y(IT+3)+Y(IT+1)                                          0
    Y(IT+3)=0.                                                       0
225 UBDY=0.                                                          0
    LBDY=0                                                           0
227 IT=IT+5                                                          0
    RETURN                                                           0
    END                                                              0
```

```
$IBFTC WASWEP
      SUBROUTINE SWEEP(PUNT,NSWEEP,HEDGE,FLIM)                          0
      DIMENSION STORE(25)                                               0
      DIMENSION B(1),Y(18000),SLINES(30)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,  0
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y, 0
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1, 0
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,   0
     4NGRADX,NGRADY,LN,CSCH,NG,EXTRAP,SSRCH,NSWPS,OMEGA,CVGS,CONST,     0
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,      0
     6P,Q,R,S,T,V,W,XC,YC,IH,NTAPE                                      0
      COMMON /VEL/ NGRDVC
  253 FORMAT (/13H SWEEP NUMBERI5, 9H RESIDUE=E15.7,
     110H AT COLUMN,I4,4H ROW,I4,25H WITH CONVERGENCE HISTORY//(6F18.6))
  255 FORMAT (36HOPROGRAM LEFT SUBROUTINE SWEEP AFTER,I8,12H SWEEPS WITH
     XF12.6,8H RESIDUE//)
  686 FORMAT (1X,4I10)
  750 FORMAT (1H1)
      WRITE (6,750)
      IN=0                                                              0
      REWIND 8
      NSWEP1=NSWEEP                                                     0
      WRITE(8)(Y(INM),INM=NBETAC,18000)
  316 MK=NXC*NYR                                                        0
      N=1                                                               0
  310 DO 311 NNN=1,25                                                   0
  311 STORE(NNN)=0.                                                     0
      MN=2                                                              0
      DO 112 JN1=1,NSWEP1                                               0
      IN=IN+1                                                           0
      JN=JN1/2+1                                                        0
      MN=3-MN                                                           0
  204 GOTO(1,2),MN                                                      0
    1 NBETA=NBETAC                                                      0
      NADD=1                                                            0
      NFUNCT=NCFCT                                                      0
      GO TO 3                                                           0
    2 NBETA=NBETA-1                                                     0
      NADD=-1                                                           0
      NFUNCT=NFUNCT-1                                                   0
    3 RSD1=0.                                                           0
      DO 12 II=1,NXC                                                    0
      DO 12 I=1,NYR                                                     0
      IF(Y(NBETA)-BITS)4,10,4                                           0
    4 TEMP=Y(NFUNCT)
      GO TO (6,5),MN                                                    0
    5 NBETA=NBETA-4                                                     0
    6 TEMP1=RELAX(NBETA,NFUNCT)
      IF((I.EQ.1).OR.(I.GE.NYR)) GO TO 601
      RSD=TEMP1-TEMP                                                    0
   19 RSD=RSD*OMEGA                                                     0
   17 Y(NFUNCT)=TEMP+RSD                                                0
```

```
601 GO TO (7,8),MN                                                    0
  7 NBETA=NBETA+4                                                     0
  8 IF(JN-1)9,9,10                                                    0
  9 IF (RSD1-ABS (RSD)) 99,10,10                                      0
 99 RSD1=ABS (RSD)
    RSD3=RSD                                                          0
    RSD4=RSD3                                                         0
    NLOC=NFUNCT                                                       0
    NLOC1=NBETA                                                       0
    GOTO(20,21),MN                                                    0
 20 NLOC1=NLOC1-4                                                     0
 21 NFU=(NFUNCT-NCFCT+1)/NYR                                          0
    NFUN=(NFUNCT-NCFCT+1)-NFU*NYR                                     0
    IF(NFUN)97,98,97                                                  0
98  NFUN=NYR                                                          0
97  NFU=NFU+1                                                         0
 10 NBETA=NBETA+NADD                                                  0
 12 NFUNCT=NFUNCT+NADD                                                0
400 GOTO(301,112),MN                                                  0
301 STORE(JN)=Y(NLOC)                                                 0
    GO TO(112,299),N                                                  0
299 IF(JN-2)302,112,303                                               0
302 IF(RSD1-CVGS)502,304,304                                          0
303 IF((STORE(JN)-STORE(JN-1))/(STORE(JN-1)-STORE(JN-2)))298,112,112  0
298 N=2                                                               0
    GOTO 306                                                          0
304 IEND=NCFCT+MK
    WRITE (8)  (Y(IJN),IJN=NCFCT,IEND)
318 BACKSPACE  8
112 CONTINUE                                                          0
    GO TO(306,320),N                                                  0
320 FACT=(STORE(6)-STORE(5))/(STORE(2)-STORE(1))                      0
    IF(FACT-.98)329,330,330                                           0
329 FACT=(1./(1.-FACT)-1.)*HEDGE+1.                                   0
    FACT=AMIN1(FACT,FLIM)                                             0
    GOTO 331                                                          0
330 FACT=FLIM                                                         0
331 NBETAX=NBETA+1-NCFCT                                              0
    IEND=NBETAC+MK
    READ (8) (Y(IJN),IJN=NBETAC,IEND)
322 NFUNCT=NCFCT                                                      0
    NBETA=NBETAC                                                      0
    DO 321 I=1,MK                                                     0
    Y(NFUNCT)=(Y(NFUNCT)-Y(NBETA))*FACT+Y(NBETA)                      0
    NFUNCT=NFUNCT+1                                                   0
321 NBETA=NBETA+1                                                     0
324 REWIND 8
    IEND=NBETAC+MK
    READ (8) (Y(IJN),IJN=NBETAC,IEND)
306 WRITE (6,253) IN,RSD3,      NFU,NFUN,(STORE(I),I=1,JN)
 16 IF(IN-NSWPS)307,502,502
307 GOTO(308,309),N                                                   0
```

```
308 NSWEP1=NSWEEP                                                       0
    N=2                                                                 0
    GOTO 310                                                            0
309 NSWEP1=11                                                           0
    N=1                                                                 0
    GOTO 310                                                            0
502 WRITE (6,255) IN,RSD4
    DO 105 I=NGRDXC,NGRDYC
105 Y(I)=0.
 61 RETURN
    END                                                                 0
```

```
$IBFTC WARELX
      FUNCTION RELAX(NB,NFUNCT)                                          0
      DIMENSION B(1),Y(18000),SLINES(30)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,   0
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y, 0
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1,  0
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,    0
     4NGRADX,NGRADY,LN,CSCH,NG,EXTRAP,SSRCH,NSWPS,OMEGA,CVGS,CONST,      0
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,       0
     6P,Q,R,S,T,V,W,XC,YC,IH,NTAPE                                       0
      N1=NFUNCT+NYR                                                      0
      N3=NFUNCT-NYR                                                      0
      RELAX=Y(N1)*Y(NB)+Y(NFUNCT+1)*Y(NB+1)+Y(N3)*Y(NB+2)+               0
     1Y(NFUNCT-1)*Y(NB+3)+Y(NB+4)                                        0
      RETURN                                                             0
      END                                                                0
```

```
$IBFTC WALNK2
      SUBROUTINE VELOC
      DIMENSION B(1),Y(18000),SLINES(30)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y,
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1,
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,
     4NGRADX,NGRADY,LN,CSCH,NG,NDERIV,SSRCH,NSWPS,OMEGA,CVGS,CONST,
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,
     6P,Q,R,S,T,V,W,XC,YC,IH,NTAPE
      COMMON /COM2/ XSCL,YSCL,PUNCH, ITLE,Y0,Y1,XL1,XL2,YL1,YL2,XSCLA,
     1YSCLA,XL1A,YL1A,XL2A,YL2A
      COMMON /VEL/ NGRDVC
      DIMENSION  ITLE(24)
  650 FORMAT (1H1,3X,2HX=F14.7//)
  651 FORMAT (11X,1HY,16X,3HPSI,12X,12HX-DERIVATIVE,6X,12HY-DERIVATIVE,
     19X,8HVELOCITY//)
  652 FORMAT (1X,5F18.7)
  686 FORMAT (1X,4I10)
      ZERO=0.
    7 V=1.
      N2=1
  112 DO 1 NN=1,2
      NGRAD=NGRDC
      CALL ICMPGS
    1 CSCH=CSCH+1.
   16 K=NGRDYC
      J=KJ
      NFUNCT=NCFCT
      NGRADX=NGRDXC
      NGRADY=NGRDYC
      NGRADV=NGRDVC
      DO 250 KP=1,NXC
      LINE=0
      WRITE (6,650) Y(J)
      WRITE (6,651)
      I=IST+1
      DO 251 KPX=1,NYR
      IF (Y(NGRADX).NE.BITS) GO TO 200
      WRITE (6,652) Y(I),ZERO,ZERO,ZERO,ZERO
      LINE=LINE+1
      IF (LINE.GT.52) GO TO 101
      GO TO 201
  200 WRITE (6,652) Y(I),Y(NFUNCT),Y(NGRADX),Y(NGRADY),Y(NGRADV)
      LINE=LINE+1
      IF (LINE.GT.52) GO TO 101
  201 NFUNCT=NFUNCT+1
      I=I+INT(Y(I+1))+2
      NGRADX=NGRADX+1
      NGRADY=NGRADY+1
  251 NGRADV=NGRADV+1
  250 J=J+INT(Y(J+1))+2
```

```
331 RETURN
101 LINE=0
    WRITE (6,650) Y(J)
    WRITE (6,651)
    GO TO 201
    END
```

```
$IBFTC WAICMP
      SUBROUTINE ICMPGS                                                 0
C     THIS ROUTINE IS CALLED TWICE. THE FIRST TIME IT CALLS  GROST1 AND
C     DERIV FOR INTERIOR POINTS. THE SECOND TIME IT CALLS SUBB1 AND
C     DERIV FOR BOUNDARY POINTS.
      DIMENSION B(1),Y(10000),SLINES(30)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,   0
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y, 0
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1,  0
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,   0
     4NGRADX,NGRADY,LN,CSCH,NG,EXTRAP,SSRCH,NSWPS,OMEGA,CVGS,CONST,     0
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,      0
     6P,Q,R,S,T,V,W,XC,YC,IH,JH,KH                                      0
      COMMON /VEL/ NGRDVC
*     INITIALIZE ALL SUBSCRIPTS TO PICK UP INFORMATION IN Y ARRAY.      0
*                                                                       0
      NC1=1
      NGRADX=NGRDXC-1
      NGRADY=NGRDYC-1                                                   0
      NGRADV=NGRDVC-1
      NKXV=NKXVC-1                                                      0
      NFUNCT=NCFCT-1                                                    0
      NDX=NDXC-1                                                        0
      J=KJ                                                              0
*                                                                       0
* SET LOOP TO PROGRESS FROM COLUMN TO COLUMN.                           0
*                                                                       0
      DO 76 NX=1,NXC                                                    0
* CHANGE SUBSCRIPTS AFFECTED BY A CHANGE OF COLUMN.                     0
*                                                                       0
      NDY=NDYC-1                                                        0
      NKYV=NKYVC-1                                                      0
      NDX=NDX+1                                                         0
      I = IST+1                                                         0
      NKXV=NKXV+1                                                       0
* SET LOOP TO PROGRESS ALONG A COLUMN.                                  0
      DO 75  NY=1,NYR
* CHANGE SUBSCRIPTS AFFECTED BY A CHANGE ALONG A COLUMN                 0
      NGRADX=NGRADX+1                                                   1
      NGRADY=NGRADY+1                                                   1
      NFUNCT=NFUNCT+1                                                   1
      NGRADV=NGRADV+1
      NKYV=NKYV+1                                                       1
      NDY=NDY+1                                                         1
*                                                                       1
* SET SUBSCRIPTS TO FIND BOUNDARY INFORMATION FROM THE INPUT ARRAY.     1
*                                                                       1
      JN=J+2                                                            1
      IN=I+2                                                            1
      KYVA=Y(NKYV)/4.                                                   1
      KYVB = KYVA                                                       1
*                                                                       1
```

```
* LOOP 10 AND 112 ARE NECESSARY FOR COMPLEX BOUNDARIES.
*
      DO 10 IS=1,KYVA
*
* LOCATE POSITION OF POINT IN THE ENTIRE SYSTEM.
*
* THE POINT WILL EITHER BE INTERIOR,EXTERIOR ORBOUNDARY.
*
      IF(Y(IN)-Y(J))1,2,3
    1 IF(Y(IN+2)-Y(J))110,22,7
    7 KXVA=Y(NKXV)/4.
      KXVB=KXVA
      DO 112 JS=1,KXVA
      IF(Y(JN)-Y(I))11,222,3
   11 IF(Y(JN+2)-Y(I))12,2222,13
*
* THE DERIVATIVES FOR INTERIOR POINTS ARE COMPUTED ON THE FIRST SWEEP.
   13 IF (CSCH-1.) 113,75,75
  113 CALL GRDST1(NXA,NYA)
      GO TO 90
   12 KXVB=KXVB-1
      IF (KXVB) 3,3,112
  112 JN=JN+4
  110 KYVB = KYVB-1
      IF (KYVB) 3,3,10
   10 IN=IN+4
      GO TO 75
*
* THE DERIVATIVES ON BOUNDARIES ARE COMPUTED ON THE SECOND SWEEP.
*
    2 NB1=1
      GO TO 202
   22 NB1=2
      GO TO 202
  222 NB1=3
      GO TO 202
 2222 NB1=4
  202 IF (CSCH-1.) 75,31,75
*
* SUBROUTINE SUBB1 SUPPLIES THE INFORMATION FROM
*
* WHICH THE BOUNDARY DERIVATIVES ARE COMPUTED.
*
   31 CALL SUBB1(NXA,NYA)
*
*DERIVATIVES ARE COMPUTED USING CENTRAL,FORWARD OR BACKWARD DIFFERENCING
*
   90 CALL DERIV (NXA,NYA,NXYAA,BX,BY,G3)
*
* ON RETURN FROM DERIV,BX AND BY ARE THE X AND Y DERIVATIVES .
      Y(NGRADX)=BX
      Y(NGRADY)=BY
```

```
      Y(NGRADV)=G3
      GO TO 75
    3 Y(NGRADX)=BITS
      Y(NGRADY)=BITS
* BITS ARE STORED IN THE X AND Y DERIVATIVE ARRAY FOR EXTRAPOLATION USES
   75 I=I+INT(Y(NKYV))+2                                                1073
   76 J=J+INT(Y(NKXV))+2
   26 RETURN
      END
```

```
$IBFTC WAGR01
      SUBROUTINE GRDST1(NXA,NYA)
      DIMENSION B(1),Y(18000),SLINES(30)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y,
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1,
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,
     4NGRADX,NGRADY,LN,CSCH,NG,EXTRAP,SSRCH,NSWPS,OMEGA,CVGS,CONST,
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,
     6P,Q,R,S,T,V,W,XC,YC,IH
      NXA=1
      NYA=1
      N1 = NFUNCT+NYR
      N3=NFUNCT-NYR
      H1=Y(NDX)
      H2=Y(NDX-1)
      TK1=Y(NDY)
      TK2=Y(NDY-1)
      U0=Y(NFUNCT)
      U1=Y(N1)
      U2=Y(NFUNCT+1)
      U3=Y(N3)
      U4=Y(NFUNCT-1)
      IJK=0
      IKJ=0
      KIJ=0
      JIK=0
      IF (Y(IN)-Y(J)+Y(NDX-1))51,51,150
  150 H2   = Y(J)-Y(IN)
      U3=Y(IN+1)
      IJK=1
   51 IF(Y(IN+2)-Y(J)-Y(NDX))151,52,52
  151 H1   = Y(IN+2)-Y(J)
      U1=Y(IN+3)
      IKJ=1
   52 IF(Y(JN)-Y(I)+Y(NDY-1))53,53,152
  152 TK2  = Y(I)-Y(JN)
      U4=Y(JN+1)
      KIJ=1
   53 IF(Y(JN+2)-Y(I)-Y(NDY))153,540,540
  153 TK1  = Y(JN+2)-Y(I)
      U2=Y(JN+3)
      JIK=1
  540 RETURN
      END
```

```
$IBFTC WASUB1
*SUBB1 00                                                                1
      SUBROUTINE SUBB1(NXA,NYA)                                          1
      DIMENSION B(1),Y(18000),SLINES(30)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,   1
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y, 1
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1,  1
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,    1
     4NGRADX,NGRADY,LN,CSCH,NG,EXTRAP,SSRCH,NSWPS,OMEGA,CVGS,CONST,      1
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,       1
     6P,Q,R,S,T,V,W,XC,YC,IH                                             1
  301 FORMAT (28H THERE IS GARBAGE IN MEMORY//)
* THIS SUBROUTINE FINDS DATA TO COMPUTE DERIVATIVES FOR BOUNDARIES.      1
* CHOOSE WHICH BOUNDARY IS BEING WORKED ON.                              1
      MMM=3
      N1=NFUNCT+NYR                                                      1
      GO TO (12,13,14,15),NB1                                            1
*                                                                        1
* LEFT  BOUNDARIES ARE IN QUESTION.                                      1
*                                                                        1
   12 NP=NGRADY+NYR                                                      1
      N3=N1+NYR                                                          1
      U0=Y(IN+1)                                                         1
      U1=Y(N1)                                                           1
      H1=Y(NDX)                                                          1
      NXA=2                                                              1
      IF (Y(N3)-BITS)30,2,1                                              1
    1 U3=Y(N3)                                                           1
      H2=Y(NDX+1)                                                        1
      GO TO 100
    2 U3=Y(IN+3)                                                         1
      H2=ABS (Y(J)-Y(IN+2))-H1                                           1
      GO TO 100
*                                                                        1
* RIGHT BOUNDARIES ARE IN QUESTION.                                      1
*                                                                        1
   13 NP=NGRADY-NYR                                                      1
      NXA=3                                                              1
      N1=NFUNCT-NYR                                                      1
      N3=N1-NYR                                                          1
      U0=Y(IN+3)                                                         1
      U1=Y(N1)                                                           1
      H1=Y(NDX-1)                                                        1
      IF (Y(N3)-BITS)30,20,31                                            1
   31 U3=Y(N3)                                                           1
      H2=Y(NDX-2)                                                        1
      GO TO 100
   20 U3=Y(IN+1)                                                         1
      H2=ABS (Y(J)-Y(IN))-H1                                             1
*                                                                        1
* SET UP DATA TO FACILITATE FINDING Y'DERIVATIVES                        1
  100 T5=Y(NDY-2)                                                        1
```

```
      T6=Y(NDY+1)
      U2=Y(NFUNCT+1)
      U6=Y(NFUNCT+2)
      U4=Y(NFUNCT-1)
      U5=Y(NFUNCT-2)
      TK1=Y(NDY)
      TK2=Y(NDY-1)
*
* IS THE BOUNDARY POINT A CORNER OF THE CIRCUMSCRIBED RECTANGULAR REGION
*
      IF(NDYC-NDY)  22,10,22
   22 IF(NDYC-NDY+NYR-1)23,7,23
   23 IF(U2-BITS)30,7,6
    6 IF(U4-BITS)30,10,90
*
* THE BOUNDARY POINT IS A TOP CORNER.
*
    7 IF(U4-BITS)30,8,11
   11 IF(U5-BITS)30,8,21
   21 U2=U4
      U4=U5
      TK1=TK2
      TK2=T5
      NYA=3
      GO TO 9
*
* THE POINT I SA BOTTOM CORNER OF THE RECTANGLE
*
   10 IF (U2-BITS)30,8,18
   18 IF (U6-BITS)30,8,120
  120 U4=U6
      TK2=T6
      NYA=2
      GO TO 9
*
* A LINEAR EXTRAPOLATION IS MADE WHEN DATA IS INSUFFICIENT.
*
    8 GO TO (112,113,114,115),NB1
  112 NP1=NP+NYR
      D1=H1
      D2=H1+H2
      GO TO 116
  113 NP1=NP-NYR
      D1=H2
      D2=H1+H2
  116 NYA=4
      BY=Y(NP1)-(D2/D1)*(Y(NP1)-Y(NP))
      GO TO 9
  114 NP1=NP+1
      D1=TK1
      D2=TK1+TK2
      GO TO 117
```

```
  115 NP1=NP-1
      D1=TK1
      D2=TK1+TK2
  117 NXA=4                                                          1
      BX=Y(NP1)-(D2/D1)*(Y(NP1)-Y(NP))
      GO TO 9                                                        1
   14 N3=NFUNCT-NYR                                                  1
      NYA=2                                                          1
      U0=Y(JN+1)                                                     1
      U2=Y(NFUNCT+1)                                                 1
      TK1=Y(NDY)                                                     1
      IF (Y(NFUNCT+2)-BITS)30,82,81                                  1
   81 U4 = Y(NFUNCT+2)                                               1
      TK2=Y(NDY+1)                                                   1
      GO TO 1000                                                     1
   82 U4=Y(JN+3)                                                     1
      TK2=ABS (Y(I)-Y(JN+2))-TK1                                     1
 1000 IF(Y(N3)-BITS)30,8,300                                         1
  300 U3=Y(N3)                                                       1
      H2=Y(NDX-1)                                                    1
      IF (Y(N1)-BITS)30,8,70                                         1
   70 U1=Y(N1)                                                       1
      H1=Y(NDX)                                                      1
      GO TO 91                                                       1
   15 N3=NFUNCT-NYR                                                  1
      NYA=3                                                          1
      U0=Y(JN+3)                                                     1
      U2=Y(NFUNCT-1)                                                 1
      TK1=Y(NDY-1)                                                   1
      IF(Y(NFUNCT-2)-BITS)30,42,41                                   1
   41 U4 = Y(NFUNCT-2)                                               1
      TK2=Y(NDY-2)                                                   1
      GO TO 110                                                      1
   42 U4=Y(JN+1)                                                     1
      TK2=ABS (Y(I)-Y(JN))-TK1                                       1
  110 IF (Y(N3)-BITS)30,8,43                                          1
   43 U3=Y(N3)                                                       1
      H2=Y(NDX-1)                                                    1
      IF (Y(N1)-BITS)30,8,47                                         1
   47 U1=Y(N1)                                                       1
      H1=Y(NDX)                                                      1
   91 NXA=1                                                          1
      GO TO 9                                                        1
   90 NYA=1                                                          1
      GO TO 9                                                        1
   30 WRITE (6,301)
      GO TO (81,82),MMM
    9 RETURN
      END                                                            1
```

```
$IBFTC WADERV
      SUBROUTINE DERIV (NXA,NYA,NXYAA,XDERIV,YDERIV,G3)
C     SUBROUTINE DERIV CALCULATES DERIVATIVES OF PSI WITH RESPECT TO X
C     AND Y, AND VELOCITIES.
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y,     0
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1,      0
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,        0
     4NGRADX,NGRADY,LN,CSCH,NG,NDERIV,SSRCH,NSWPS,OMEGA,CVGS,CONST,          0
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,           0
     6NLRCAL,Q,R,S,T,V,W,XC,YC,IH,NTAPE                                      0
      DIMENSION B(1),Y(18000),SLINES(30)
  600 FORMAT (1X,6F14.6)
*     IF NXA=1 THE X-DERIVATIVE IS FOUND BY CENTRAL DIFFERENCING             1
*     IF NXA=2 THE X-DERIVATIVE IS FOUND BY FORWARD DIFFERENCING             1
*     IF NXA=3 THE X-DERIVATIVE IS FOUND BY BACKWARD DIFFERENCING            1
*     THE ABOVE CONVENTION HOLDS FOR NYA AND Y-DERIVATIVES                   1
*     IF NXA=4 NO X-DERIVATIVE IS FOUND.                                     1
      FCT=1.
      IF (AXYM.EQ.1.) FCT=Y(I)
      ND=1                                                                   1
  100 DELTA=1.                                                               1
      FG=1.
      GO TO (3,2,1,7),NXA                                                    1
    1 DELTA=-1.                                                              1
    2 H2=H1+H2
      FG=-1.
    3 GX=DELTA*(H2**2*(U1-U0)+   H1**2*(U0-U3))/(H1*H2*(H2+FG*H1))
    4 GO TO (6,10),ND                                                        1
    6 XDERIV=GX                                                              1
      IF (NYA-4) 7,15,7                                                      1
    7 ND=2                                                                   1
      NXA=NYA                                                                1
      U1=U2
      U3=U4
      H1=TK1
      H2=TK2
      GO TO 100                                                              1
*     G3=GRADIENT,YDERIV=Y ER CROSS DERIVATIVE,XDERIV=X-DERIVATIVE           1
   10 YDERIV=GX                                                              1
   15 CONTINUE
      G3=SQRT((XDERIV/FCT)**2+(YDERIV/FCT)**2)
   13 RETURN                                                                 1
      END                                                                    1
```

```
$IBFTC WALNK3
      SUBROUTINE LINES
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,
     1HONE,HTWO,ONEK,TWOK,L,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y,
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1,
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,
     4NGRADX,NGRADY,LN,CSCH,NG,NDERIV,SSRCH,NSWPS,OMEGA,CVGS,CONST,
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,
     6P,Q,R,S,T,V,W,XC,YC,IH,NTAPE,TITLE1,NCHAIN,MONTH
      COMMON /COM2/ XSCL,YSCL,PUNCH, ITLE,Y0,Y1,XL1,XL2,YL1,YL2,XSCLA,
     1YSCLA,XL1A,YL1A,XL2A,YL2A
      COMMON /COM3/ YMAX
      COMMON/FRSCOF/ FIRST,V0,CO
      COMMON /VEL/ NGRDVC,AIR
      DIMENSION Y(18000),SLINES(30), ITLE(24)
      DIMENSION B(1)
  501 FORMAT (6E12.6)
  686 FORMAT (1X,4I10)
    2 ILN=0
      NNLNS=1
   34 IF (ILN.EQ.NNLNS) RETURN
   36 READ (5,501) ANSL,CSCH,NNLNS,CO  ,V0,AIR
      IF (CSCH.EQ.2.) NGRDC=NGRDXC
      IF (NNLNS.LT.1) NNLNS=1
      NSL=ANSL
      READ (5,501) (SLINES(INK),INK=1,NSL)
   12 NSLNC=NGRDC
      DO 61 NG=1,NSL
      LN=NSLNC
  111 N2=1
      CALL MARC
 1111 NGRAD=NGRDC
   60 CALL SORT (INT(CSCH)-1)
   61 CONTINUE
   27 ILN=ILN+1
      GO TO 34
      END
```

```
$IBFTC WASGSC
*SGSRCH                                                                 1
      SUBROUTINE SGSRCH                                                 1
          DIMENSION       Y(18000),SLINES(30)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,  1
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y, 1
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1, 1
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,   1
     4NGRADX,NGRADY,LN,CSCH,NG,EXTRAP,SSRCH,NSWPS,OMEGA,CVGS,CONST,     1
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,      1
     6P,Q,R,S,T,V,W,XC,YC,IH,NTAPE                                      1
      COMMON /ZCOM/ Z
      COMMON /LNK5/ NV
      COMMON /COM/ U7,W1
      COMMON/BDRY/ JB
      COMMON/PRSCOF/ FIRST,VO
  600 FORMAT (1X,F14.7)
  601 FORMAT (48H STORAGE OVERRUN WHILE CALCULATING STREAM LINES )
  602 FORMAT (1X,5F14.5)
      N=NG
      IF (CSCH-3.)  57,81,82                                            1
   81 N1=NGRADX+NYR                                                     1
      N3=NGRADX-NYR                                                     1
      U0=Y(NGRADX)                                                      1
      U1=Y(N1)                                                          1
      U2=Y(NGRADX+1)                                                    1
      U3=Y(N3)                                                          1
      U4=Y(NGRADX-1)                                                    1
      GO TO 57                                                          1
   82 N1=NGRADY+NYR                                                     1
      N3=NGRADY-NYR                                                     1
      U0=Y(NGRADY)                                                      1
      U1=Y(N1)                                                          1
      U2=Y(NGRADY+1)                                                    1
      U3=Y(N3)                                                          1
      U4=Y(NGRADY-1)                                                    1
   57 N2=LN                                                             1
      V=0.                                                              1
      IF (KIJ) 84,85,84                                                 1
   84 IF (CSCH-3.) 85,2,2                                               1
   85 U6=U4                                                             1
      U7=U2
      W1=ONEK
      W=-TWOK
      CALL SUB58 (U6,N)
      IF (V) 3,2,3                                                      1
    3 Y(LN+1)=Y(J)                                                      1
      Y(LN+2)=Y(I)+XC
      IF (KIJ.EQ.1.AND.JB.EQ.1) GO TO 31
      NV1=NV+1
      Y(LN+3)=Y(NV)+(Y(NV1)-Y(NV))*XC/W
      GO TO 32
```

```
 31 W=TWOK
    Y(LN+3)=(W+W1)*(Y(NV)-Y(NV+1))/W1+Y(NV+1)
 32 Y(LN+4)=1.-(Y(LN+3)/VO)**2
    LN=LN+5
  2 V=0.                                                          1
 12 V=0.                                                          1
    IF (CSCH-3.) 89,90,90
 89 IF (JIK) 22,73,22
 22 W=ONEK                                                        2
    W1=-TWOK
    U6=U2                                                         1
    U7=U4
    CALL SUB58 (U6,N)
    IF (V) 21,73,21                                               1
 21 Y(LN+1)=Y(J)                                                  1
    Y(LN+2)=Y(I)+XC                                               1
    IF (JIK.EQ.1.AND.JB.EQ.1) GO TO 40
    NV1=NV-1
    Y(LN+3)=Y(NV)+(Y(NV1)-Y(NV))*XC/W
    GO TO 41
 40 W1=TWOK
    Y(LN+3)=(W+W1)*(Y(NV)-Y(NV-1))/W1+Y(NV-1)
 41 Y(LN+4)=1.-(Y(LN+3)/VO)**2
    LN=LN+5
 90 V=0.
 73 CONTINUE                                                      1
 58 IF (LN.GT.17983) GO TO 100
    RETURN                                                        2
100 WRITE (6,601)
    STOP
    END
```

83

```
$IBFTC WAS858
      SUBROUTINE SU858 (U6,N)
      DIMENSION R(1),Y(18000),SLINES(30)
      COMMON /COM/ U7,W1
      COMMON /PLOT/ RANK
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,      1
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y,    1
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1,     1
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,       1
     4NGRADX,NGRADY,LN,CSCH,NG,EXTRAP,SSRCH,NSWPS,OMEGA,CVGS,CONST,         1
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,HI2,HK2,BY,BX,NKXV,NKYV,N3,          1
     6P,Q,R,S,T,V,W,XC,YC,IH                                                1
      Y(LN)=SLINES(N)                                                       1
      Y(LN)=RANK
      P=SLINES(N)-U6                                                        1
      Q=U0-SLINES(N)                                                        1
      IF (ABS(P+Q).LT.ABS(Q-P)) GO TO 2
      P=SLINES(N)
      IF (AXYM.EQ.0.0.OR.S.EQ.1.) GO TO 20
      XC=(P-U0)*(P-U6)/((U7-U0)*(U7-U6))*W1+(P-U7)*(P-U0)/((U6-U7)*(U6-
     1U0))*W
      IF (ABS(XC).LT.ABS(W1)) GO TO 21
   20 XC=(P-U0)/(U6-U0)*W
   21 V=1.                                                                  1
    2 RETURN.                                                               1
      END                                                                 1922
```

```
$IBFTC WAREGN
      SUBROUTINE REGION                                                 0.
      DIMENSION B(1),Y(18000),SLINES(30)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,  0
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y, 0
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1, 0.
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,   0.
     4NGRADX,NGRADY,LN,CSCH,NG,IXTRAP,SSRCH,NSWPS,OMEGA,CVGS,CONST,     0
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,      0
     6P,Q,R,S,T,V,W,XC,YC,IH,NTAPE                                      0
      COMMON UBDY,LBDY                                                  0
  226 FORMAT (3H A=F12.6/3H C=F12.6/3H E=F12.6/4H Y1=F11.6/4H Y2=F11.6/5
     X3H THE DIFFERENTIAL EQUATION IS NOT ELLIPTIC AT (Y1,Y2)/57H IF SUB
     XROUTINE ACDEFG IS USED. OTHERWISE CHECK A,C,AND F.//)
C     STATEMENT 112 TO 4 SET MESH VALUES FOR DERIBATIVE ENFORCEMENT      0
      IF (IXTRAP) 112,113,112                                           0
  112 IKJ=0                                                             0
      KIJ=0                                                             0
      JIK=0                                                             0
      IJK=0                                                             0
      HONE=Y(NDX)                                                       0
      HTWO=Y(NDX-1)                                                     0
      ONEK=Y(NDY)                                                       0
      TWOK=Y(NDY-1)                                                     0
      IF (UBDY) 1,2,1                                                   0
    1 ONEK=TWOK                                                         0
      GO TO 3                                                           0
    2 IF (LBDY) 4,113,4                                                 0
    4 TWOK=ONEK                                                         0
      GO TO 3                                                           0
  113 CALL GRDSTL                                                       0
    3 IF (CONST) 141,149,141                                            0
C     CONST, AN INPUT VALUE, CONTROLS USAGE OF ACDEFG.                  0
C     XC,YC ARE X AND Y COORDINATE VALUES.                              0
  149 CALL ACDEFG                                                       0
  141 IF (A*C)  147,147,43                                              0
C     STATEMENT 141 CHECKS ELLIPTICITY                                  0
  147 WRITE (3,226) A,C,E,Y(I),Y(J)
      KNTR = 1                                                          0
   43 CALL CNSTNT                                                       0
C     SUBROUTINE CNSTNT CALCULATES COEFFICIENTS OF DIFFERENCE ANALOGUE  0
   75 RETURN                                                            0
      END
```

```
$IBFTC WASORT
      SUBROUTINE SORT(KASE)
      DIMENSION XT(250),YT(250)
      DIMENSION XP(250),YP(250)
      DIMENSION TA(4000)
      DIMENSION TP(9)
      DIMENSION B(1),Y(18000),SLINES(30)
      DIMENSION    ITLE(24),HEAD(2,3)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y,
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1,
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,
     4NGRADX,NGRADY,LN,CSCH,NG,NDERIV,SSRCH,NSWPS,OMEGA,CVGS,CONST,
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,
     6P,Q,R,S,T,V,W,XC,YC,IH,NTAPE
      COMMON/PRSCOF/ FIRST,VO
      COMMON /COM2/ XSCL,YSCL,PUNCH, ITLE,YO,Y1,XL1,XL2,YL1,YL2,XSCLA,
     1YSCLA,XL1A,YL1A,XL2A,YL2A
      COMMON /COM3/ YMAX
      EQUIVALENCE (TA(1),Y(11000))
      EQUIVALENCE (XT(1),XP(1)),(YT(1),YP(1))
  600 FORMAT  (1H1,2A6,1H=F14.7//8X,1HX,13X,1HY,10X,8HVELOCITY,6X,
     18HVELOCITY,6X,8HPRESSURE/47X,5HRATIO,9X,11HCOEFFICIENT/)
  620 FORMAT (1X,5F14.2)
  630 FORMAT (1X,I3)
  640 FORMAT (1X//)
  700 FORMAT (5E14.8)
 1001 FORMAT (5F14.7)
      DATA (HEAD(I,1),I=1,2)/6H   PSI,6H       /,(HEAD(I,2),I=1,2)/6HX-DE
     XRI,6HVATIVE/,(HEAD(I,3),I=1,2)/6HY-DERI,6HVATIVE/
      RANK=1.
      LN=LN-5
      XL1P=XL1
      XL2P=XL2
      YL1P=YL1
      YL2P=YL2
      IF (XSCLA.EQ.0.) ISTOP=1
      LINE=0
      TP(1)=23.
      TP(2)=4.
      TP(3)=XSCL
      TP(4)=YSCL
      TP(5)=0.
      TP(6)=0.
      TP(8)=4.
      TP(9)=0.
    2 NSTOR =LN+5
      ISTOP=0
    5 DO 20  I=NGRDC,LN,5
      IF (Y(I).NE.RANK) GO TO 20
   10 Y(NSTOR)=Y(I)
      Y(NSTOR+1)=Y(I+1)
```

```
      Y(NSTOR+2)=Y(I+2)
      Y(NSTOR+3)=Y(I+3)
      Y(NSTOR+4)=Y(I+4)
      NSTOR=NSTOR+5
   20 CONTINUE
      GO TO 23
   22 RANK=RANK-1.0
   23 NSTOR=NSTOR-5
      NSTOR1=LN+5
      IF (NSTOR-NSTOR1.LE.6) GO TO 102
   46 LINE=0.
      XL1P=XL1
      XL2P=XL2
      YL1P=YL1
      YL2P=YL2
  971 WRITE (6,600) (HEAD(I,KASE),I=1,2),SLINES(NG)
      DO 775 I=NSTOR1,NSTOR,5
      IF (Y(I+1).EQ.Y(I+6).AND.Y(I+2).EQ. Y(I+7))  GO TO 775
      VR=Y(I+3)/VO
      WRITE (6,1001) Y(I+1),Y(I+2),Y(I+3),VR,Y(I+4)
      LINE=LINE+1
      IF (LINE.LT.50) GO TO 775
      LINE=0
      WRITE (6,600) (HEAD(J,KASE),J=1,2),SLINES(NG)
  775 CONTINUE
   72 J=0
      DO 100 I=NSTOR1,NSTOR,5
      J=J+1
      XP(J)=Y(I+1)
  100 YP(J)=Y(I+2)
      KNTR=0
      DO 65 JK=1,J
      IF (XP(JK).LT.XL1P.OR.XP(JK).GT.XL2P.OR.YP(JK).LT.YL1P.OR.YP(JK) .
     XGT.YL2P) GO TO 62
      GO TO 65
   62 XP(JK)=BITS
      KNTR=KNTR+1
   65 CONTINUE
      KT=0
      DO 68 JK=1,J
      IF (XP(JK).NE.BITS) GO TO 68
   66 KT=KT+1
      JKT=JK+KT
      IF (XP(JKT).EQ.BITS) GO TO 66
      DO 67 LJ=JK,J
      JKT=LJ+KT
      XP(LJ)=XP(JKT)
   67 YP(LJ)=YP(JKT)
      KT=0
   68 CONTINUE
      J=J-KNTR
      AJ=J
```

```
      TP(7)=AJ
      IF (ISTOP.NE.0) GO TO 80
      WRITE (8) J
      WRITE (8) (XP(I),I=1,J),(YP(I),I=1,J)
      XL1P=XL1A
      XL2P=XL2A
      YL1P=YL1A
      YL2P=YL2A
      ISTOP=2
      GO TO 72
   80 IF (ISTOP.EQ.1) GO TO 81
      TP (3)=XSCLA
      TP (4)=YSCLA
      TP(2)=4.
   81 CALL GPLOTS(TP,XP,YP)
      RANK=RANK+1.
      GO TO 2
  102 IF (NG.LT.NSL) RETURN
      TP(7)=0
      TP(9)=-1.
      TP(2)=4.
      CALL GPLOTS (TP,XP,YP)
      IF (ISTOP.NE.2) GO TO 120
      REWIND 8
      TP (3)=XSCL
      TP (4)=YSCL
      TP(9)=0.
      DO 101 I=1,NSL
      READ (8) J
      READ (8) (XP(M),M=1,J),(YP(M),M=1,J)
      TP(7)=J
      TP(2)=4.
  101 CALL GPLOTS (TP,XP,YP)
      TP(7)=0.
      TP(9)=-1.
      TP(2)=4.
      CALL GPLOTS (TP,XP,YP)
  120 CALL REMOVE(23,3)
      IF (PUNCH.EQ.0.) GO TO  99
      YT(1)=Y(IST+1)
      DO 123 I=2,NYR
      ID=NKYV+I-1
  123 YT(I)=Y(ID)+YT(I-1)
      XT(1)=Y(KJ)
      DO 124 I=2,NXC
      ID=NKXV+I-1
  124 XT(I)=Y(ID)+XT(I-1)
      TA(1)=1.
      TA(2)=NYR+1
      TA(3)=NXC+1
      TA(4)=0.
      DO 125 I=1,NYR
```

```
125 TA(I+4)=YT(I)
    NFUNCT=NCFCT-1
    JK=NYR+5
    DO 256 I=1,NXC
    TA(JK)=XT(I)
    JK=JK+1
    DO 255 NQ=1,NYR
    NFUNCT=NFUNCT+1
    TA(JK)=Y(NFUNCT)
    JK=JK+1
255 CONTINUE
256 CONTINUE
    JK=JK-1
    WRITE (7,700)(TA(I),I=1,3)
    WRITE (7,700) (TA(I),I=4,JK)
 99 RETURN
    END
```

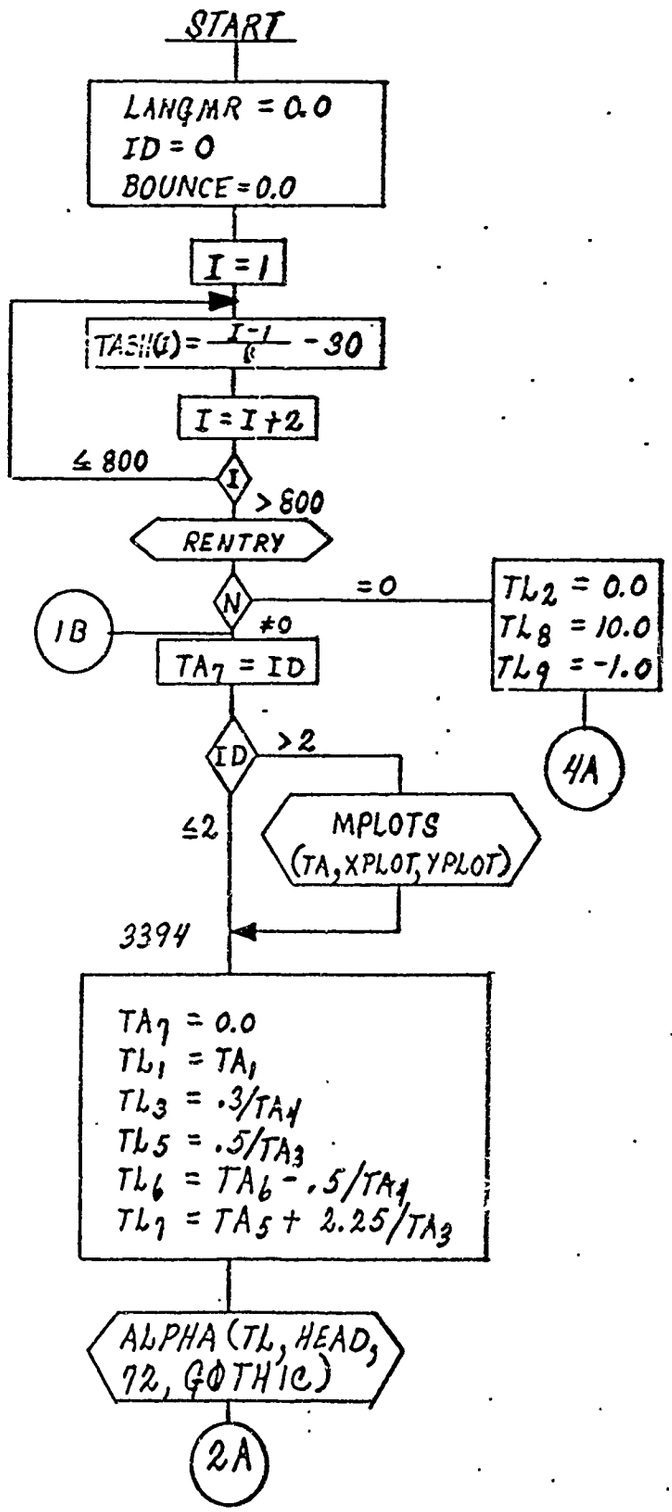
```
$IBFTC WAMARK
*MARC                                                                   1
      SUBROUTINE MARC                                                   1
* THIS SUBROUTINE PERFORMS THE FUNCTION OF SEARCH IN LINK 3.            1
      DIMENSION B(1),Y(18000),SLINES(30)
      COMMON B,NXC,NYR,NKXVC,NKYVC,NDXC,NDYC,NCFCT,NBETAC,KJ,BITS,IST,  1
     1HONE,HTWO,ONEK,TWOK,I,IN,J,JN,IKJ,KIJ,JIK,IJK,IT,NDX,NDY,NFUNCT,Y, 1
     2SLINES,H1,H2,TK1,TK2,NB1,NB2,NB3,NB4,NC1,NC2,NC3,NC4,N1,N2,U0,U1, 1
     3U3,U2,U4,IPRIN,NRSDUE,NSL,NSLNC,NGRDC,NGRDXC,NGRDYC,AXYM,NGRAD,   1
     4NGRADX,NGRADY,LN,CSCH,NG,EXTRAP,SSRCH,NSWPS,OMEGA,CVGS,CONST,     1
     5STREAM,KNTR,A,C,D,E,F,G,HK1,HK12,H12,HK2,BY,BX,NKXV,NKYV,N3,      1
     6P,Q,R,S,T,V,W,XC,YC,IH,JH,KH                                      1
      COMMON /VEL/ NGRDVC,AIR
      COMMON/BDRY/ JB
      COMMON /ZCOM/ Z
      COMMON /LNK5/ NV
      COMMON /PLOT/ RANK
  600 FORMAT (I12)
  620 FORMAT (1X,6F14.2)
*     INITIALIZE ALL SUBSCRIPTS TO PICK UP INFORMATION IN Y ARRAY.      1
*                                                                       1
      HL=0.0
      SRANK=1.0
      FLAG=0.0
      Z=0.
      LN=NGRDC
      NGRADX=NGRDXC-1                                                   1
      NGRADY=NGRDYC-1                                                   1
      NV=NGRDVC-1
      NKXV=NKXVC-1                                                      1
      NFUNCT=NCFCT-1                                                    1
      NDX=NDXC-1                                                        1
      J=KJ                                                              1
*                                                                       1
* SET LOOP TO PROGRESS FROM COLUMN TO COLUMN.                           1
*                                                                       1
      FST=0.
      DO 76 NX=1,NXC                                                    1
* CHANGE SUBSCRIPTS AFFECTED BY A CHANGE OF COLUMN.                     1
      RANK=SRANK
*                                                                       1
      NDY=NDYC-1                                                        1
      NKYV=NKYVC-1                                                      1
      NDX=NDX+1                                                         1
      I = IST+1                                                         1
      NKXV=NKXV+1                                                       1
*                                                                       1
* SET LOOP TO PROGRESS ALONG A COLUMN.                                  1
      DO 75  NY=1,NYR
* CHANGE SUBSCRIPTS AFFECTED BY A CHANGE ALONG A COLUMN                 1
*                                                                       1
      ILN=LN
```

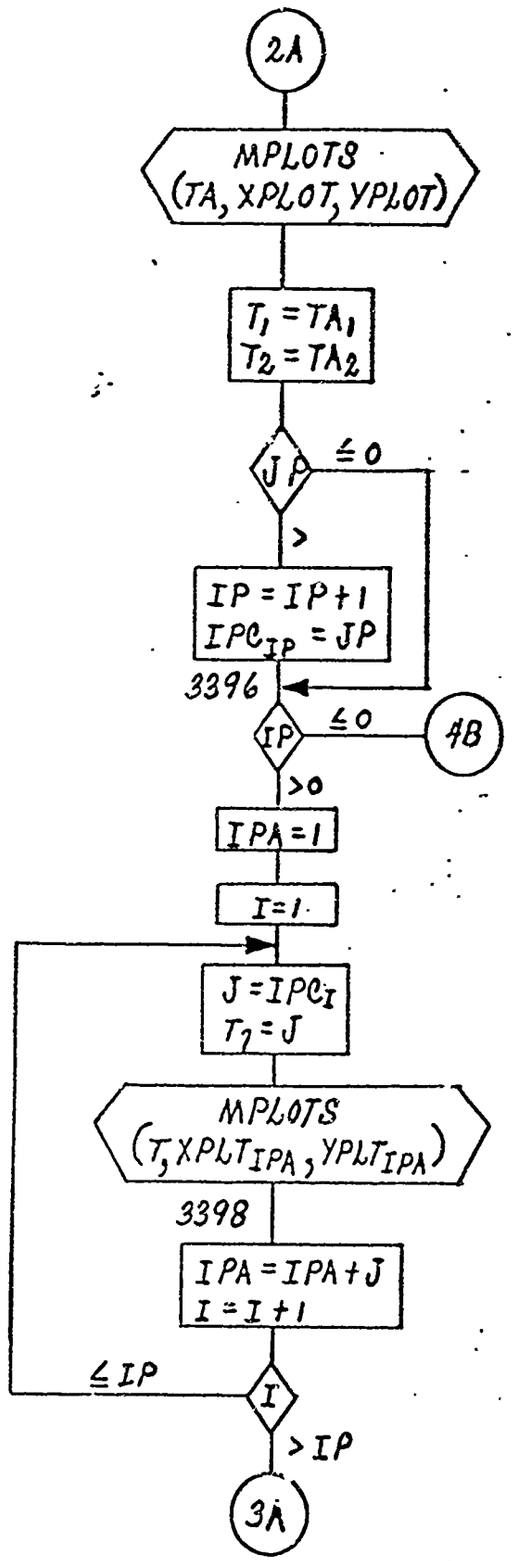
```
      NGRADX=NGRADX+1                                                   1
      NGRADY=NGRADY+1                                                   1
      NFUNCT=NFUNCT+1                                                   1
      NV=NV+1
      NKYV=NKYV+1                                                       1
      NDY=NDY+1                                                         1
*                                                                       1
* SET SUBSCRIPTS TO FIND BOUNDARY INFORMATION FROM THE INPUT ARRAY.     1
*                                                                       1
      JN=J+2                                                            1
      IN=I+2                                                            1
      KYVA=Y(NKYV)/4.                                                   1
      KYVB = KYVA                                                       1
*                                                                       1
* LOOP 10 AND 112 ARE NECESSARY FOR COMPLEX BOUNDARIES.                 1
*                                                                       1
      DO 10 IS=1,KYVA                                                   1
*                                                                       1
* LOCATE POSITION OF POINT IN THE ENTIRE SYSTEM.                        1
*                                                                       1
* THE POINT WILL EITHER BE INTERIOR,EXTERIOR ORBOUNDARY.                1
      IF(Y(IN)-Y(J))1,80,75
    1 IF(Y(IN+2)-Y(J))110,80,7
    7 KXVA=Y(NKXV)/4.                                                   1
      KXVB=KXVA                                                         1
      DO 112 JS=1,KXVA                                                  1
      IF(Y(JN)-Y(I))11,75,75                                            1
   11 IF(Y(JN+2)-Y(I))12,75,13                                          1
   12 KXVB=KXVB-1                                                       1
      IF (KXVB) 75,75,112                                               1
  112 JN=JN+4                                                           1
  110 KYVB = KYVB-1                                                     1
      IF (KYVB) 75,75,10                                                1
   10 IN=IN+4                                                           1
   13 CALL GRDST2
   14 CALL SGSRCH
      IF (LN.GT.ILN) RANK=RANK+1.
      Z=0.
   75 I=I+INT(Y(NKYV))+2
      IF (RANK.LE.2.0.OR.FST.NE.0.) GO TO 50
      IF (FLAG.EQ.1.0) GO TO 60
      IF (AIR.NE.1.0) GO TO 55
   52 FST=1.0
      HL=1.0
      IKK=5.0*(2.0+FLAG)
      IDX=FLOAT(LN)-(10.0+FLAG*10.0)
      DO 57 IK=1,IKK
      KONT=LN+6-IK
   57 Y(KONT)=Y(KONT-5)
      Y(IDX)=2.0
      GO TO 50
   55 FLAG=1.0
```

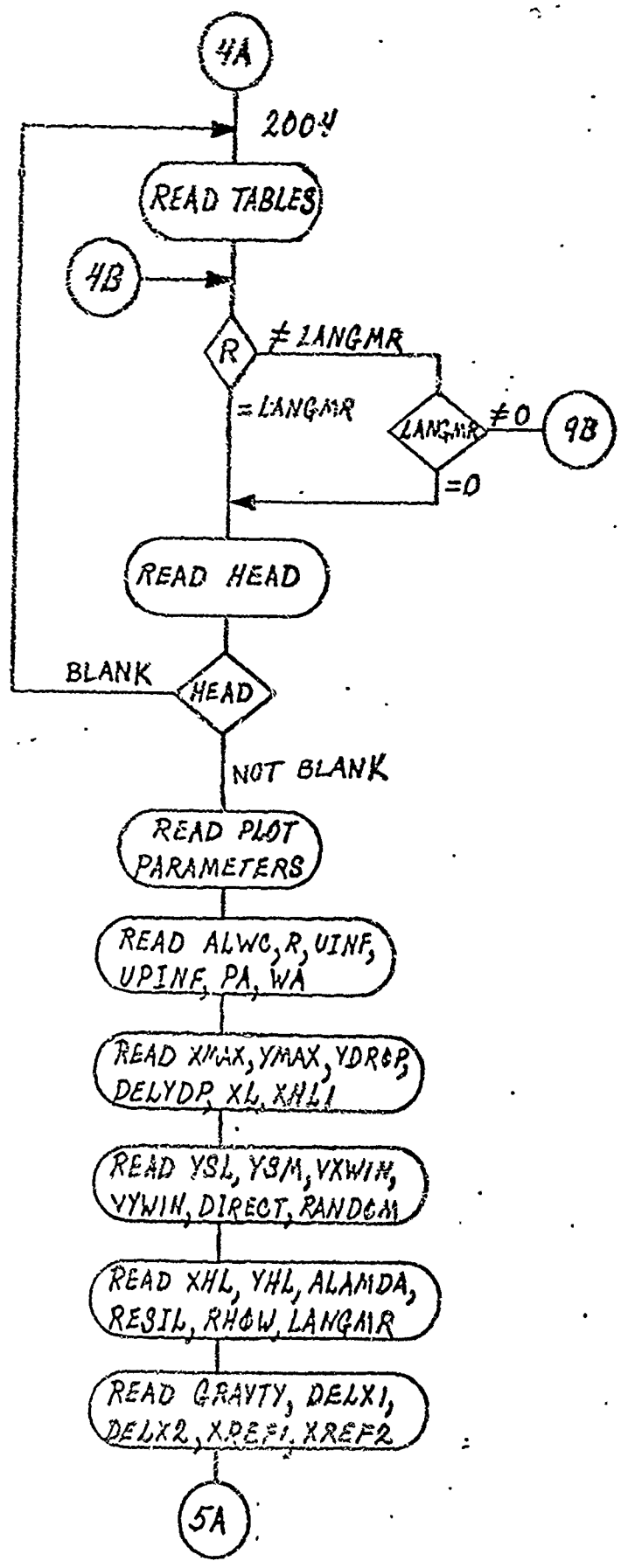
```
      GO TO 25
   60 CONTINUE
      IF (RANK.EQ.3.) GO TO 52
      SRANK=2.0
      RANK=SRANK
      FST=1.0
      IF (AIR.LT.1.0) GO TO 74
      DO 65 IK=1,20
      KONT=LN+6-IK
   65 Y(KONT)=Y(KONT-5)
      Y(LN-30)=2.0
      Y(LN-25)=3.0
      Y(LN-20)=1.0
      Y(LN-15)=4.0
      Y(LN-10)=2.0
      Y(LN-5)=3.0
      Y(LN)=4.0
      LN=LN+5
      GO TO 25
   74 DO 745 IK=1,25
      KONT=LN+6-IK
  745 Y(KONT)=Y(KONT-5)
      Y(LN-25)=2.0
      Y(LN-20)=3.0
      Y(LN-15)=4.0
      Y(LN-10)=2.0
      Y(LN-5)=3.0
      Y(LN)=4.0
      LN=LN+5
      GO TO 25
   50 CONTINUE
      IF (RANK.NE.4.0) GO TO 25
      IF (HL.EQ.1.0) GO TO 25
      SRANK=5.0
      RANK=SRANK
      DO 24 IK=1,10
      KONT=LN+11-IK
   24 Y(KONT)=Y(KONT-10)
      LN=LN+10
      Y(LN-10)=RANK
      Y(LN-5)=RANK+1.0
   25 CONTINUE
   76 J=J+INT(Y(NKXV))+2                                          1686
   26 RETURN
   80 Z=1.
      GO TO 7
      END
```

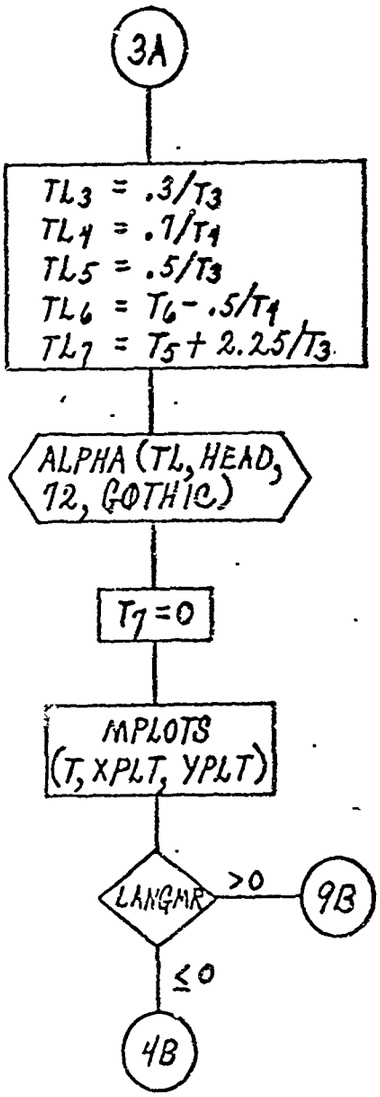
10.3 Appendix 3 - Water Droplet Trajectory Computer Program Flow Chart

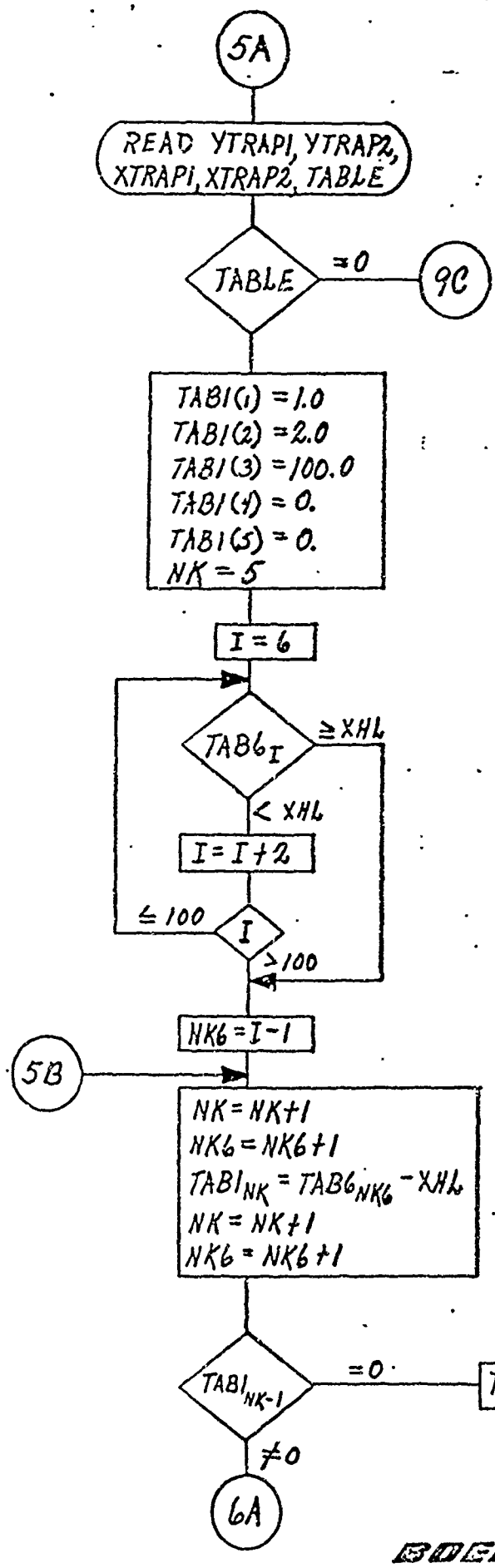
Water Droplet Trajectory Program

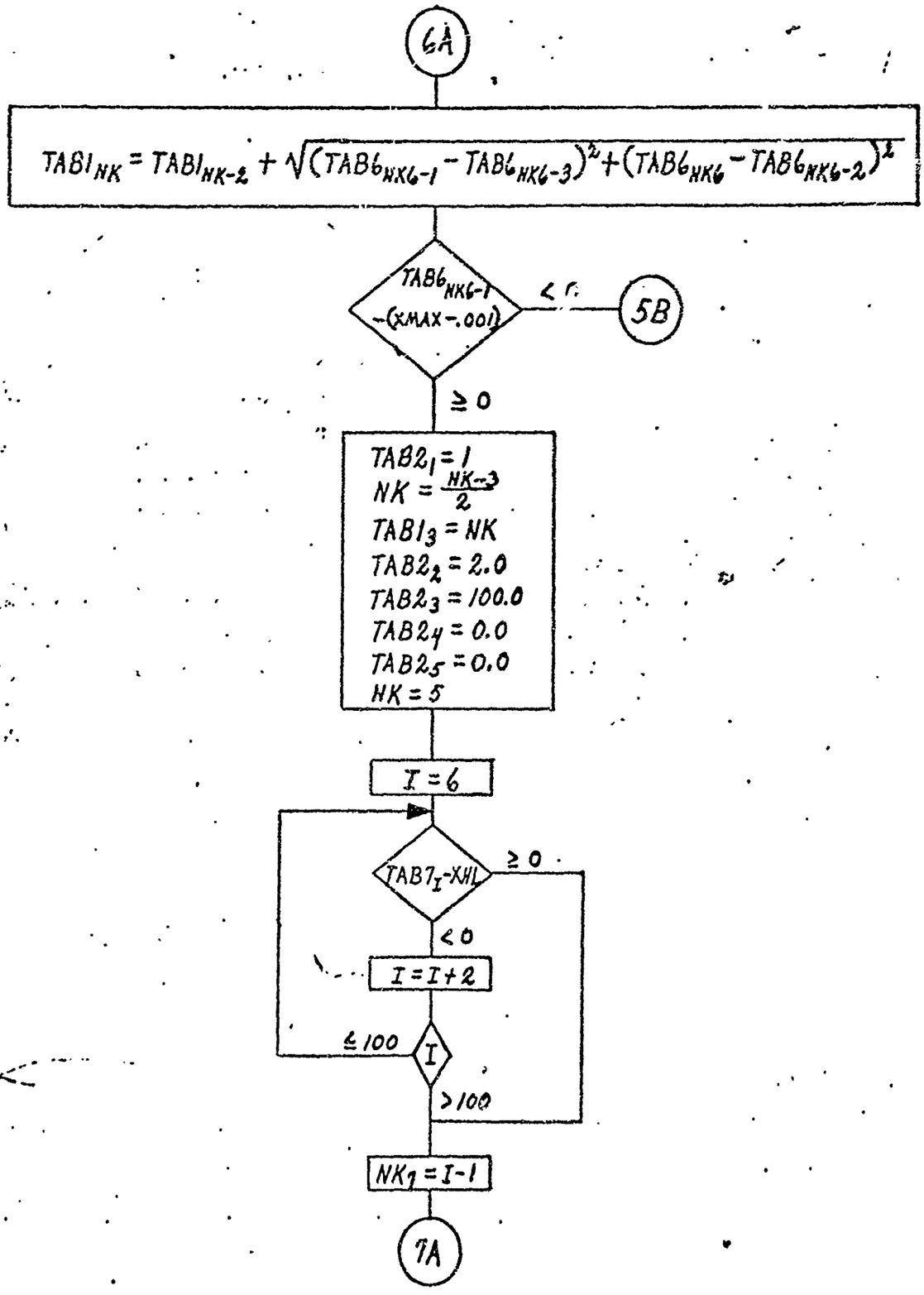


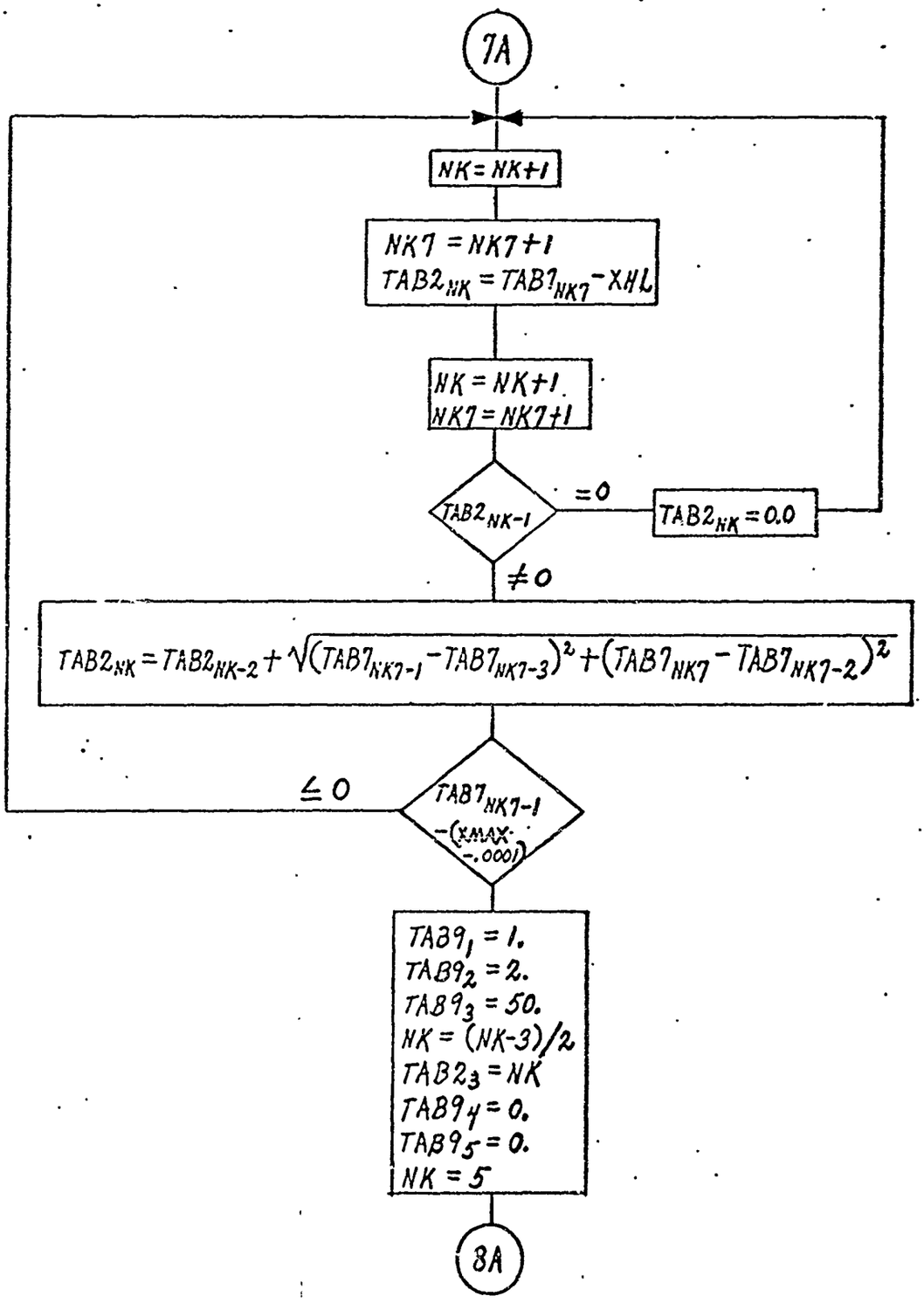

2A
MPLOTS
(TA, XPLOT, YPLOT)
T1 = TA1
T2 = TA2
JP
≤ 0
>
IP = IP + 1
IPC_IP = JP
3396
IP
≤ 0
4B
> 0
IPA = 1
I = 1
J = IPC_I
T1 = J
MPLOTS
(T, XPLT_IPA, YPLT_IPA)
3398
IPA = IPA + J
I = I + 1
≤ IP
I
> IP
3A

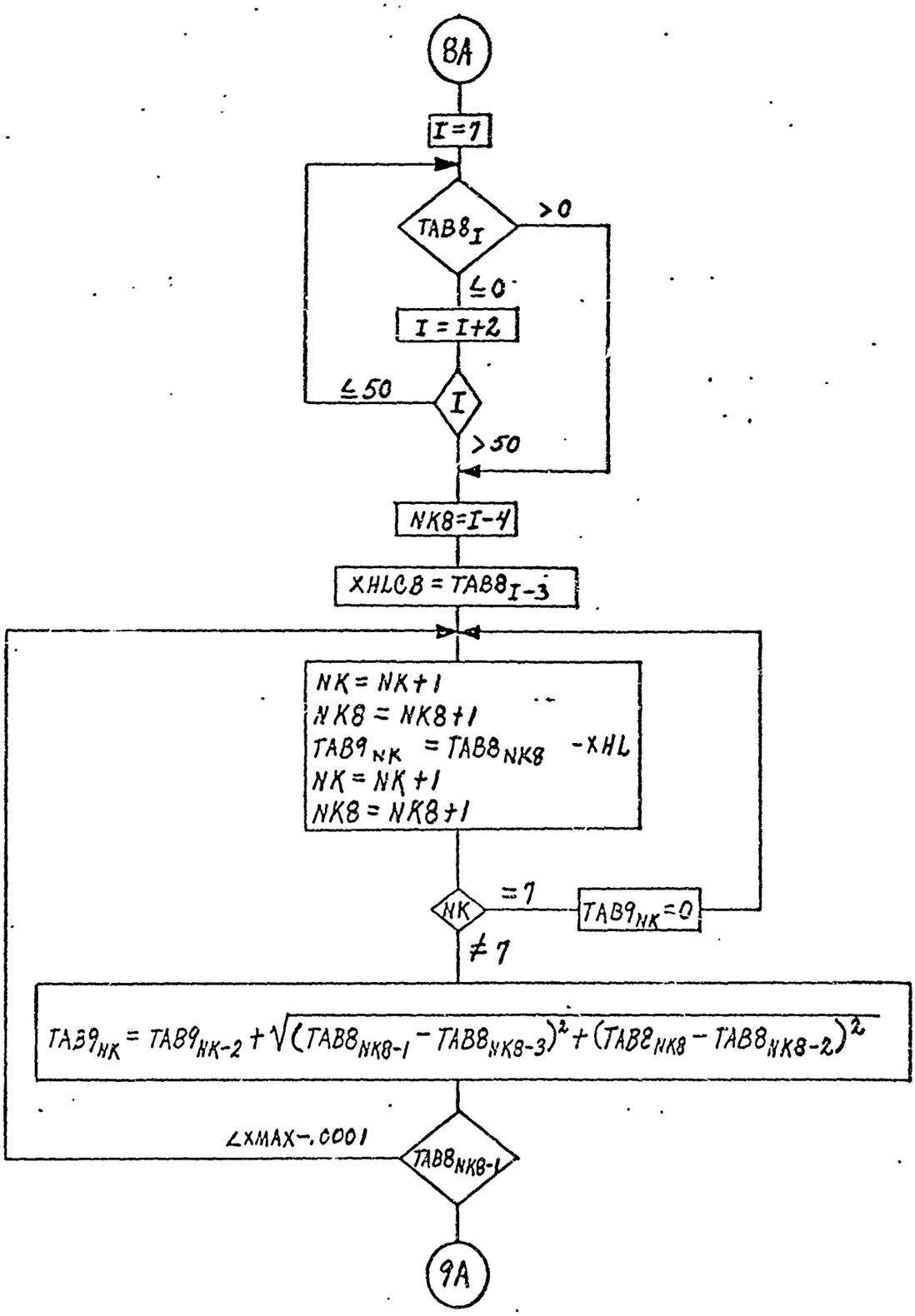

4A
2004
READ TABLES
4B
R
≠ LANGMR
= LANGMR
LANGMR
≠ 0
9B
= 0
READ HEAD
BLANK
HEAD
NOT BLANK
READ PLOT PARAMETERS
READ ALWC, R, UINF, UPINF, PA, WA
READ XMAX, YMAX, YDROP, DELYDP, XL, XHL1
READ YSL, YSM, VXWIN, VYWIN, DIRECT, RANDOM
READ XHL, YHL, ALAMDA, RESIL, RHOW, LANGMR
READ GRAVTY, DELX1, DELX2, XREF1, XREF2
5A

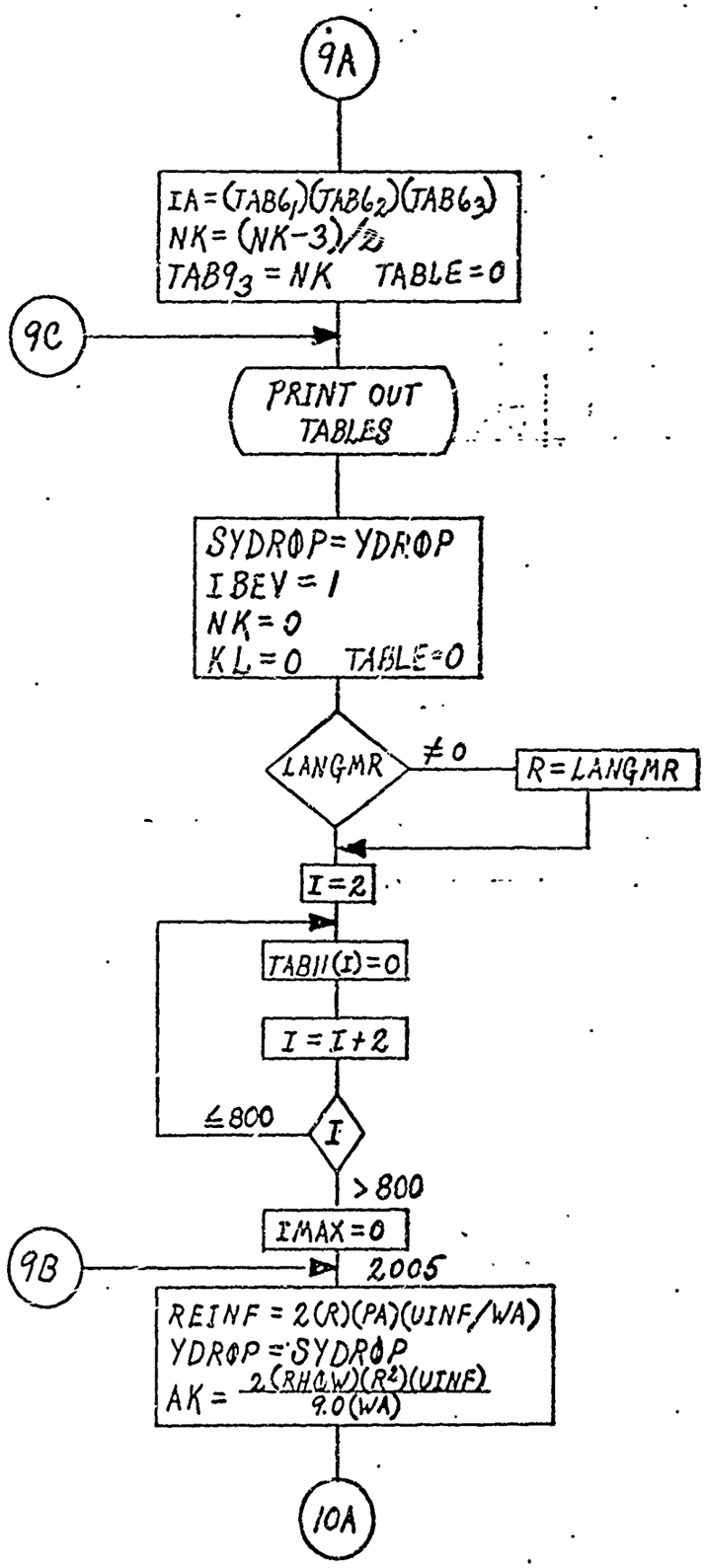

3A
TL3 = .3/T3
TL4 = .7/T4
TL5 = .5/T3
TL6 = T6 - .5/T4
TL7 = T5 + 2.25/T3
ALPHA (TL, HEAD, T2, GOTHIC)
T7 = 0
MPLOTS (T, XPLT, YPLT)
LANGMR
>0
9B
≤0
4B

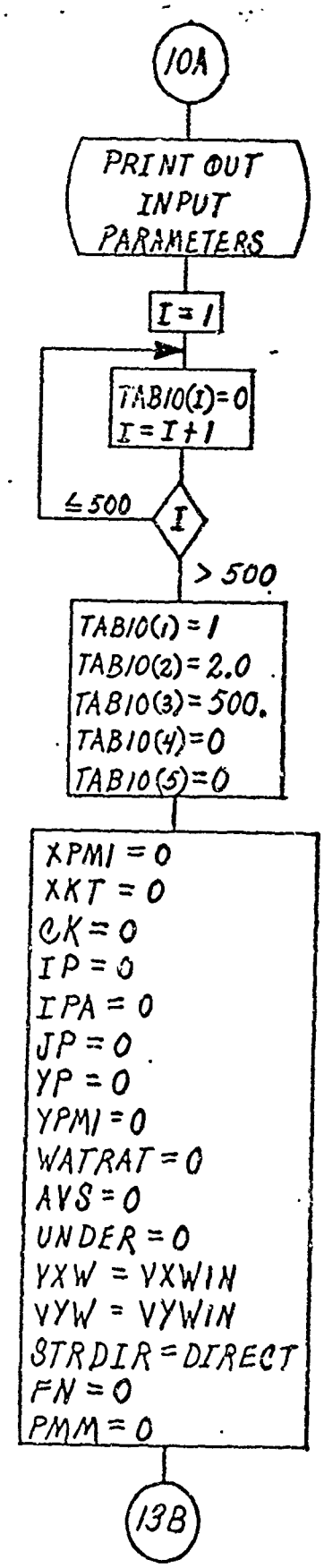

5A
READ YTRAP1, YTRAP2, XTRAP1, XTRAP2, TABLE
TABLE
=0
9C
TAB1(1) = 1.0
TAB1(2) = 2.0
TAB1(3) = 100.0
TAB1(4) = 0.
TAB1(5) = 0.
NK = 5
I = 6
TAB6_I
≥ XHL
< XHL
I = I + 2
≤ 100
I
≥ 100
NK6 = I - 1
5B
NK = NK+1
NK6 = NK6+1
TAB1_NK = TAB6_NK6 - XHL
NK = NK+1
NK6 = NK6+1
TAB1_NK-1
=0
TAB1_NK = 0.0
≠0
5B
6A

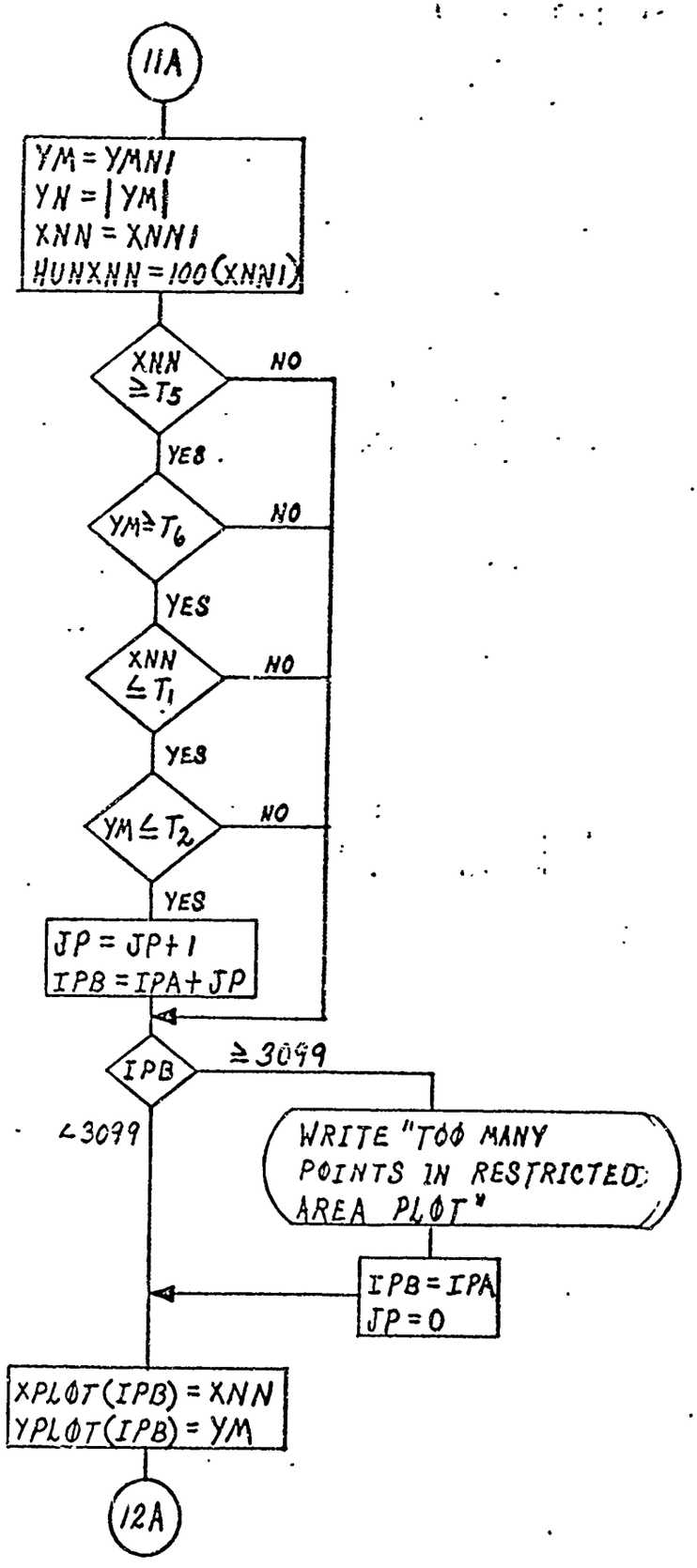

6A
$TAB1_{NK} = TAB1_{NK-2} + \sqrt{(TAB6_{NK6-1} - TAB6_{NK6-3})^2 + (TAB6_{NK6} - TAB6_{NK6-2})^2}$
$TAB6_{NK6-1} - (XMAX - .001)$
< 0
5B
≥ 0
$TAB2_1 = 1$
$NK = \frac{NK-3}{2}$
$TAB1_3 = NK$
$TAB2_2 = 2.0$
$TAB2_3 = 100.0$
$TAB2_4 = 0.0$
$TAB2_5 = 0.0$
$NK = 5$
$I = 6$
$TAB7_I - XHL$
≥ 0
< 0
$I = I + 2$
≤ 100
I
> 100
$NK_1 = I - 1$
7A

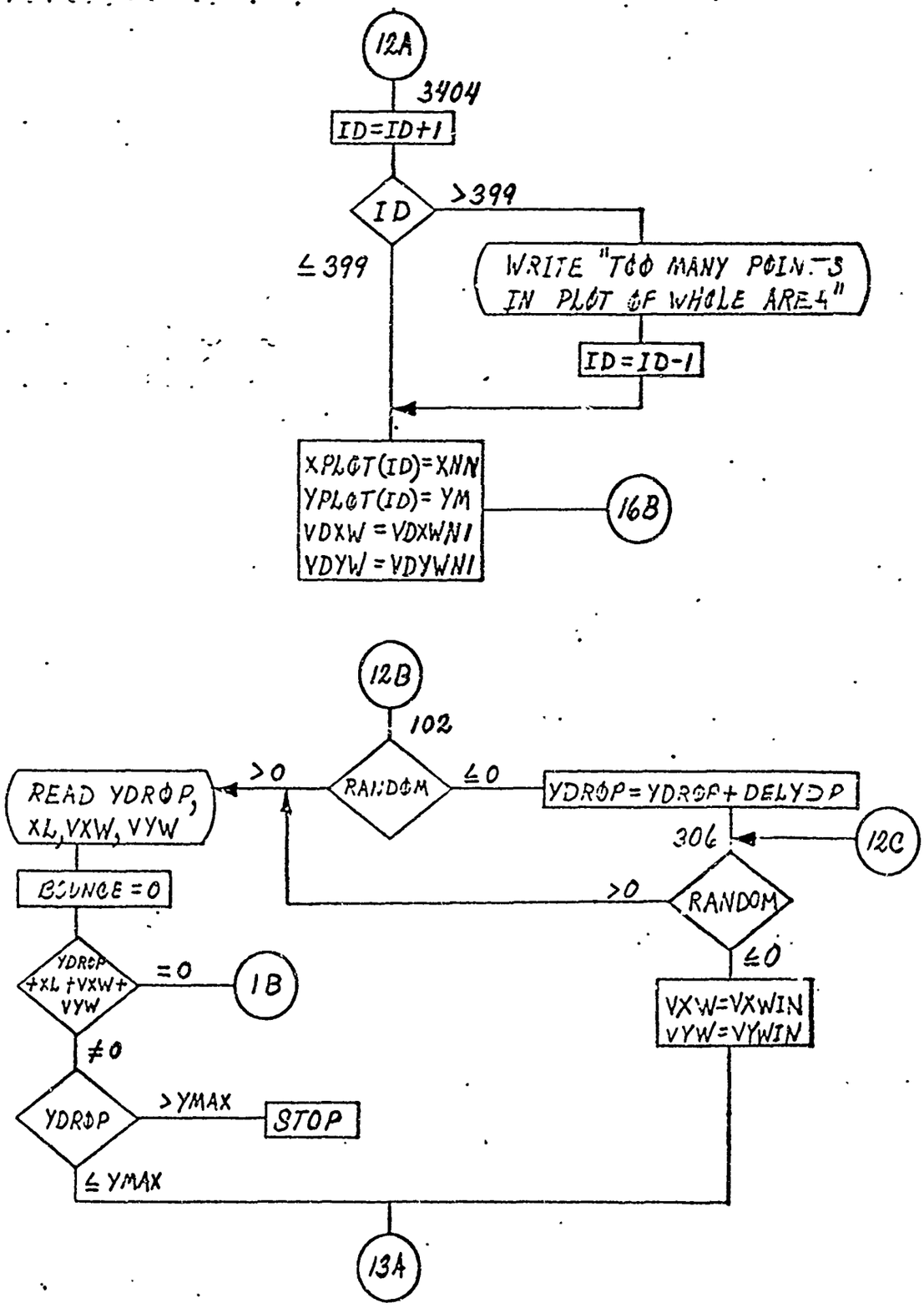

7A
NK = NK+1
NK7 = NK7+1
TAB2NK = TAB7NK7 - XHL
NK = NK+1
NK7 = NK7+1
TAB2NK-1
=0
TAB2NK = 0.0
≠0
TAB2NK = TAB2NK-2 + √((TAB7NK7-1 - TAB7NK7-3)² + (TAB7NK7 - TAB7NK7-2)²)
≤0
TAB7NK7-1 -(XMAX -.0001)
TAB91 = 1.
TAB92 = 2.
TAB93 = 50.
NK = (NK-3)/2
TAB23 = NK
TAB94 = 0.
TAB95 = 0.
NK = 5
8A

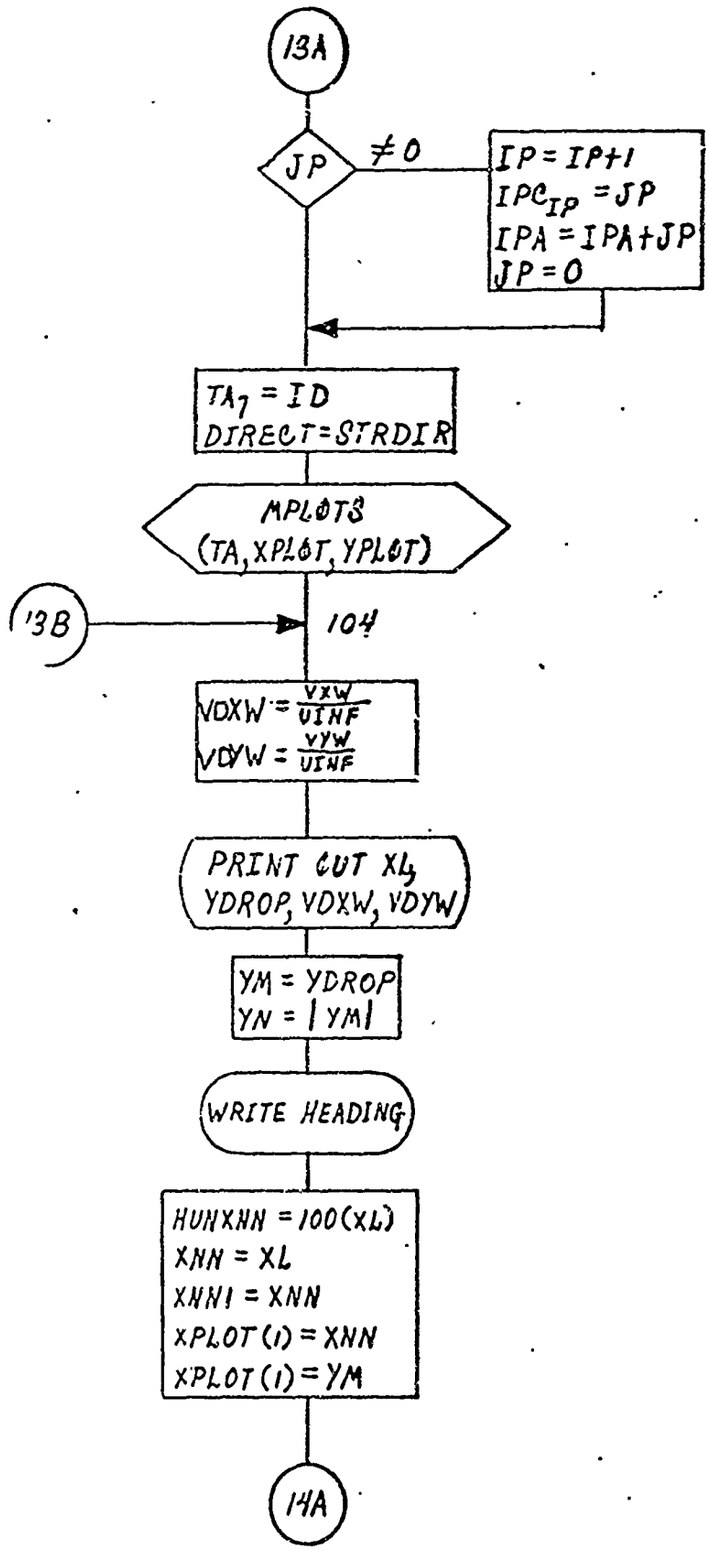

8A
I=7
TAB8_I
>0
≤0
I = I+2
≤50
I
>50
NK8=I-4
XHLCB = TAB8_I-3
NK = NK+1
NK8 = NK8+1
TAB9_NK = TAB8_NK8 -XHL
NK = NK+1
NK8 = NK8+1
NK
=7
TAB9_NK=0
≠7
TAB9_NK = TAB9_NK-2 + √((TAB8_NK8-1 - TAB8_NK8-3)² + (TAB8_NK8 - TAB8_NK8-2)²)
<XMAX-.0001
TAB8_NK8-1
9A

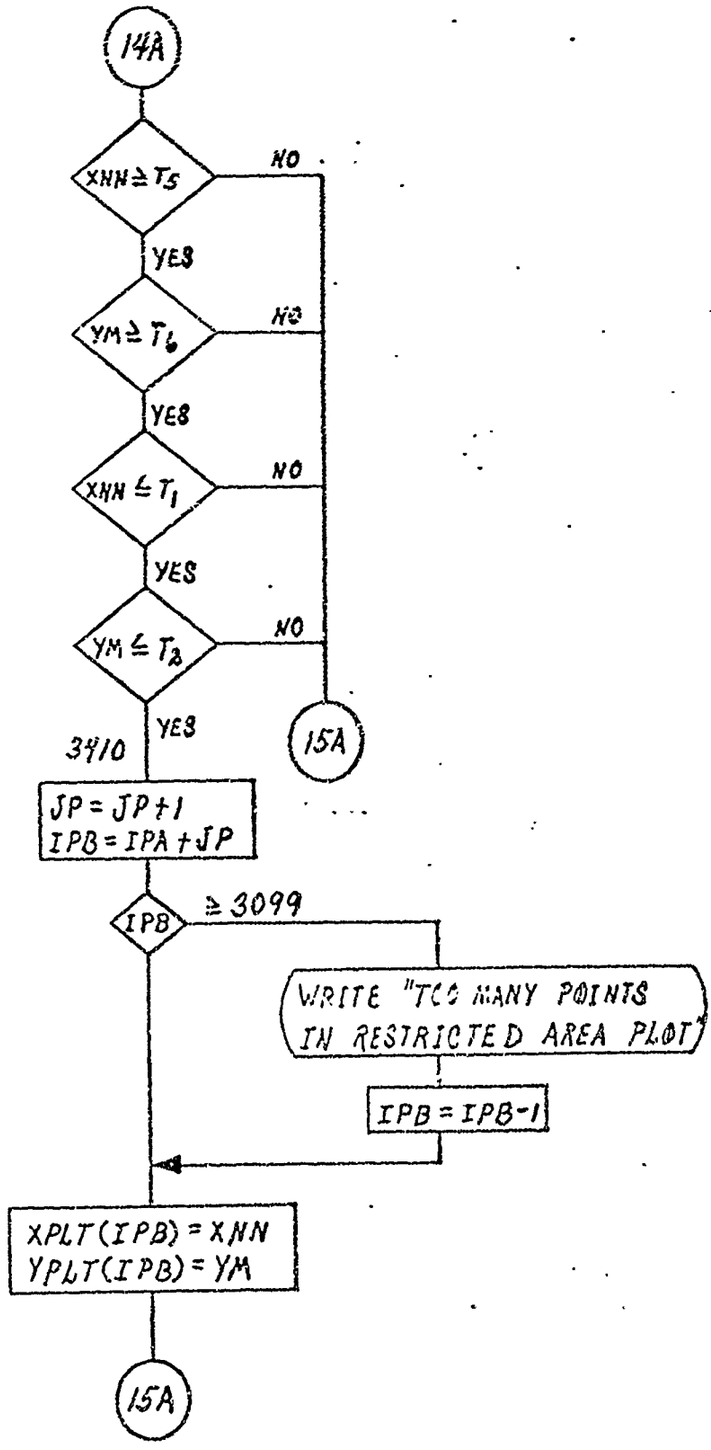

9A
IA=(TAB6₁)(TAB6₂)(TAB6₃)
NK=(NK-3)/2
TAB9₃ = NK TABLE=0
9C
PRINT OUT TABLES
SYDRØP=YDRØP
IBEV=1
NK=0
KL=0 TABLE=0
LANGMR
≠0
R=LANGMR
I=2
TAB11(I)=0
I=I+2
≤800
I
>800
IMAX=0
9B
2005
REINF = 2(R)(PA)(UINF/WA)
YDRØP = SYDRØP
AK = 2(RHOW)(R²)(UINF) / 9.0(WA)
10A

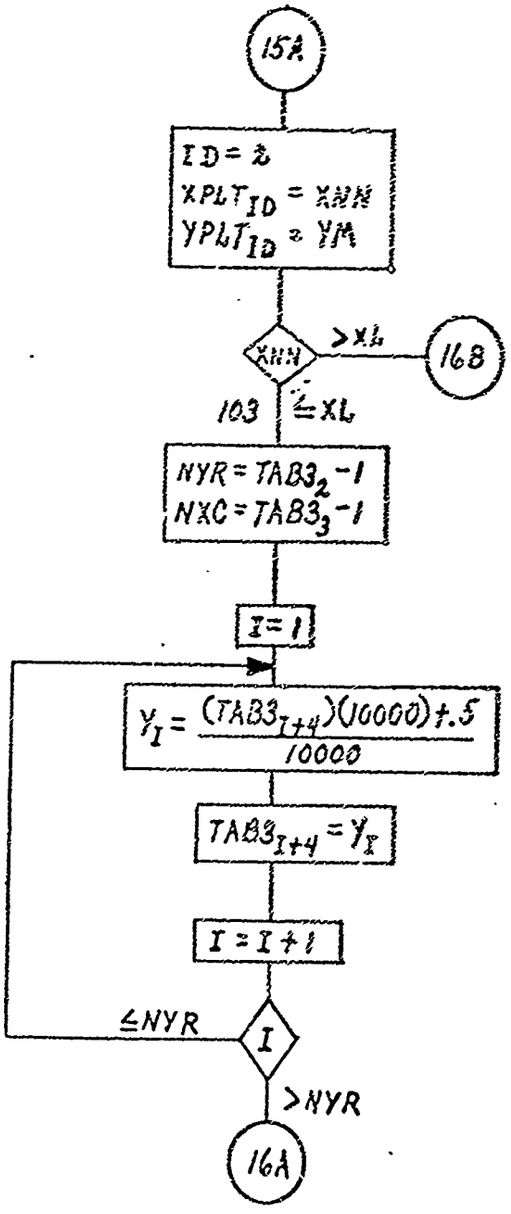

10A
PRINT OUT INPUT PARAMETERS
I = 1
TAB10(I) = 0
I = I + 1
≤ 500
I
> 500
TAB10(1) = 1
TAB10(2) = 2.0
TAB10(3) = 500.
TAB10(4) = 0
TAB10(5) = 0
XPMI = 0
XKT = 0
CK = 0
IP = 0
IPA = 0
JP = 0
YP = 0
YPMI = 0
WATRAT = 0
AVS = 0
UNDER = 0
VXW = VXWIN
VYW = VYWIN
STRDIR = DIRECT
FN = 0
PMM = 0
13B

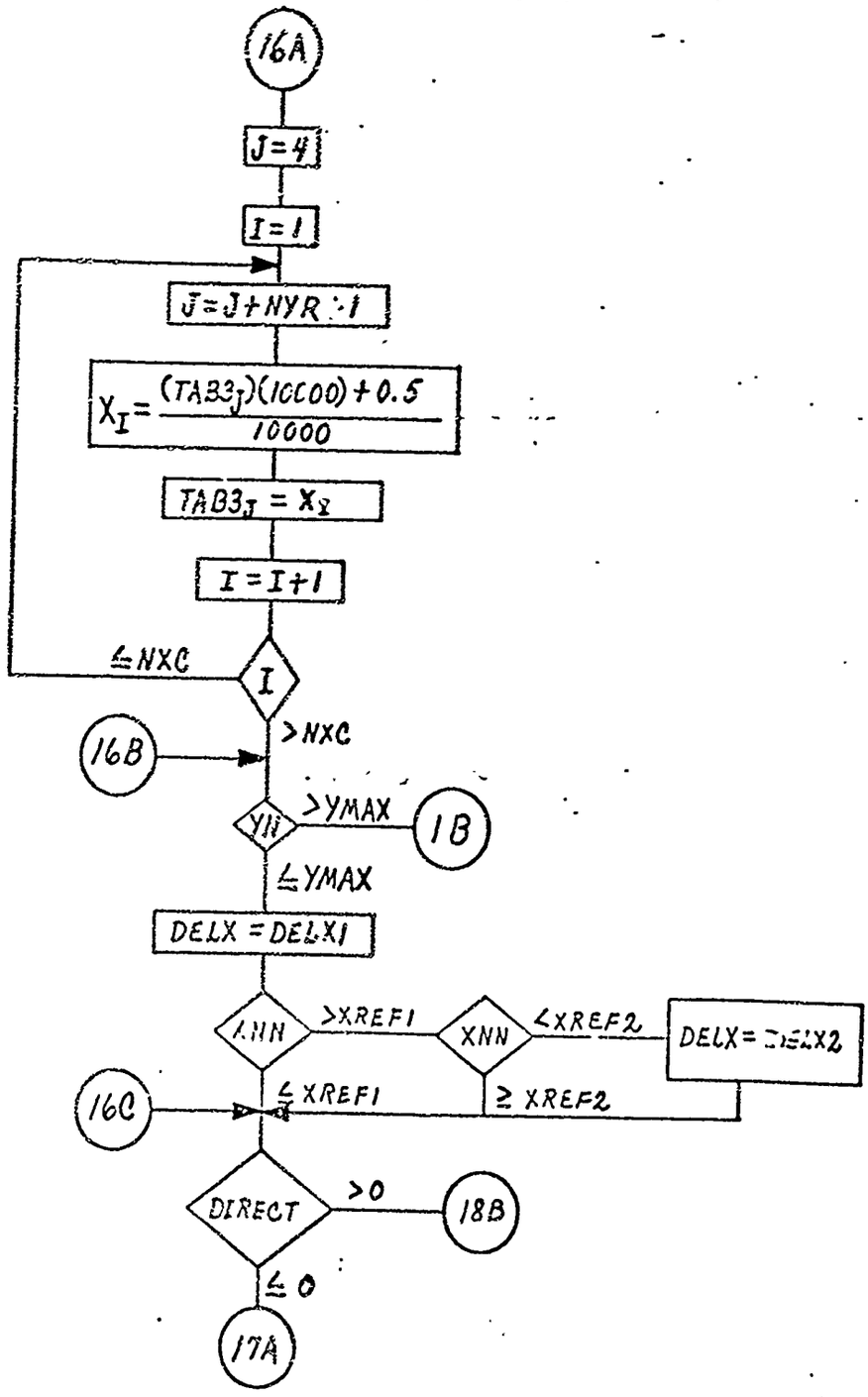

11A
YM = YMNI
YN = |YM|
XNN = XNNI
HUNXNN = 100(XNNI)
XNN ≥ T5
NO
YES
YM ≥ T6
NO
YES
XNN ≤ T1
NO
YES
YM ≤ T2
NO
YES
JP = JP + 1
IPB = IPA + JP
IPB
≥ 3099
< 3099
WRITE "TOO MANY POINTS IN RESTRICTED AREA PLOT"
IPB = IPA
JP = 0
XPLOT(IPB) = XNN
YPLOT(IPB) = YM
12A

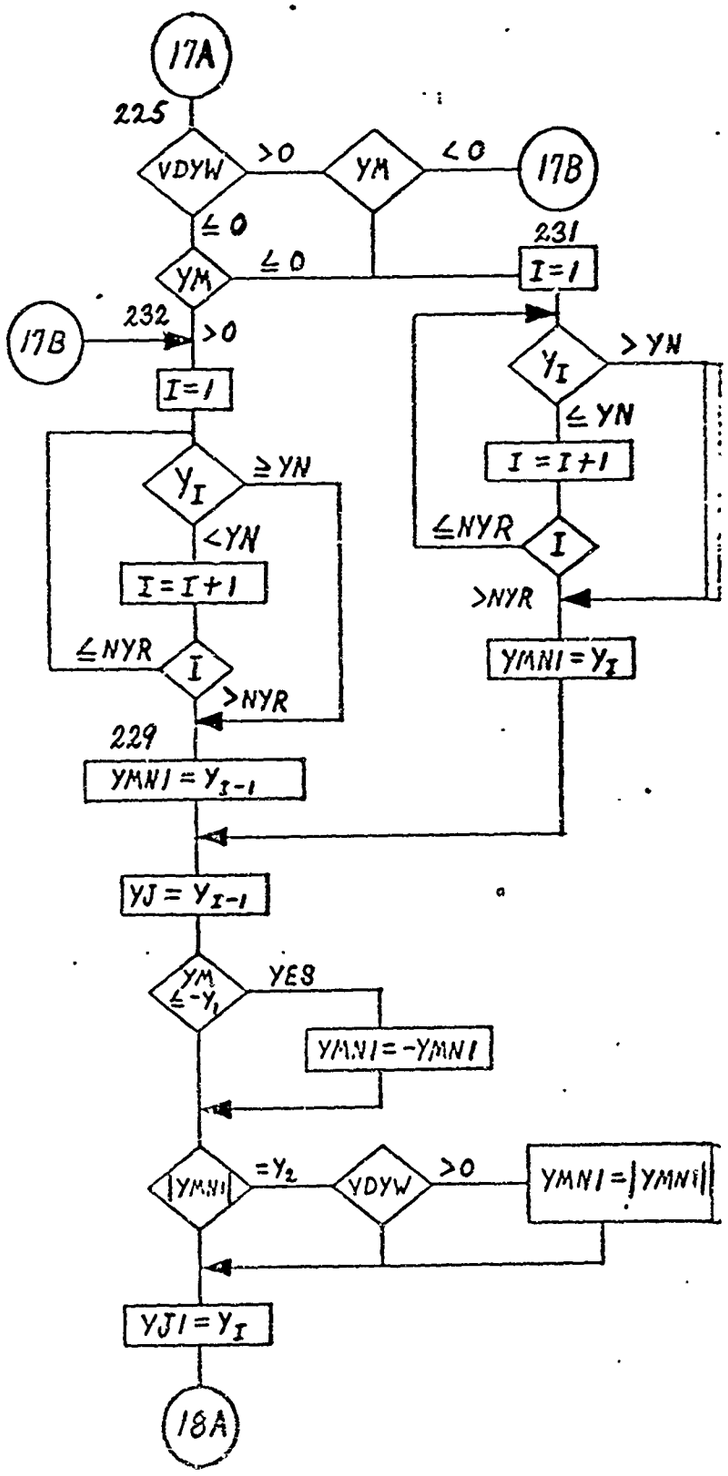

12A
3404
ID=ID+1
ID
>399
≤399
WRITE "TOO MANY POINTS IN PLOT OF WHOLE AREA"
ID=ID-1
XPLOT(ID)=XNN
YPLOT(ID)= YM
VDXW = VDXWNI
VDYW = VDYWNI
16B
12B
102
RANDOM
>0
≤0
READ YDROP, XL, VXW, VYW
YDROP=YDROP+DELYDP
306
12C
BOUNCE=0
>0
RANDOM
≤0
YDROP +XL +VXW+ VYW
=0
1B
VXW=VXWIN
VYW=VYWIN
≠0
YDROP
>YMAX
STOP
≤YMAX
13A

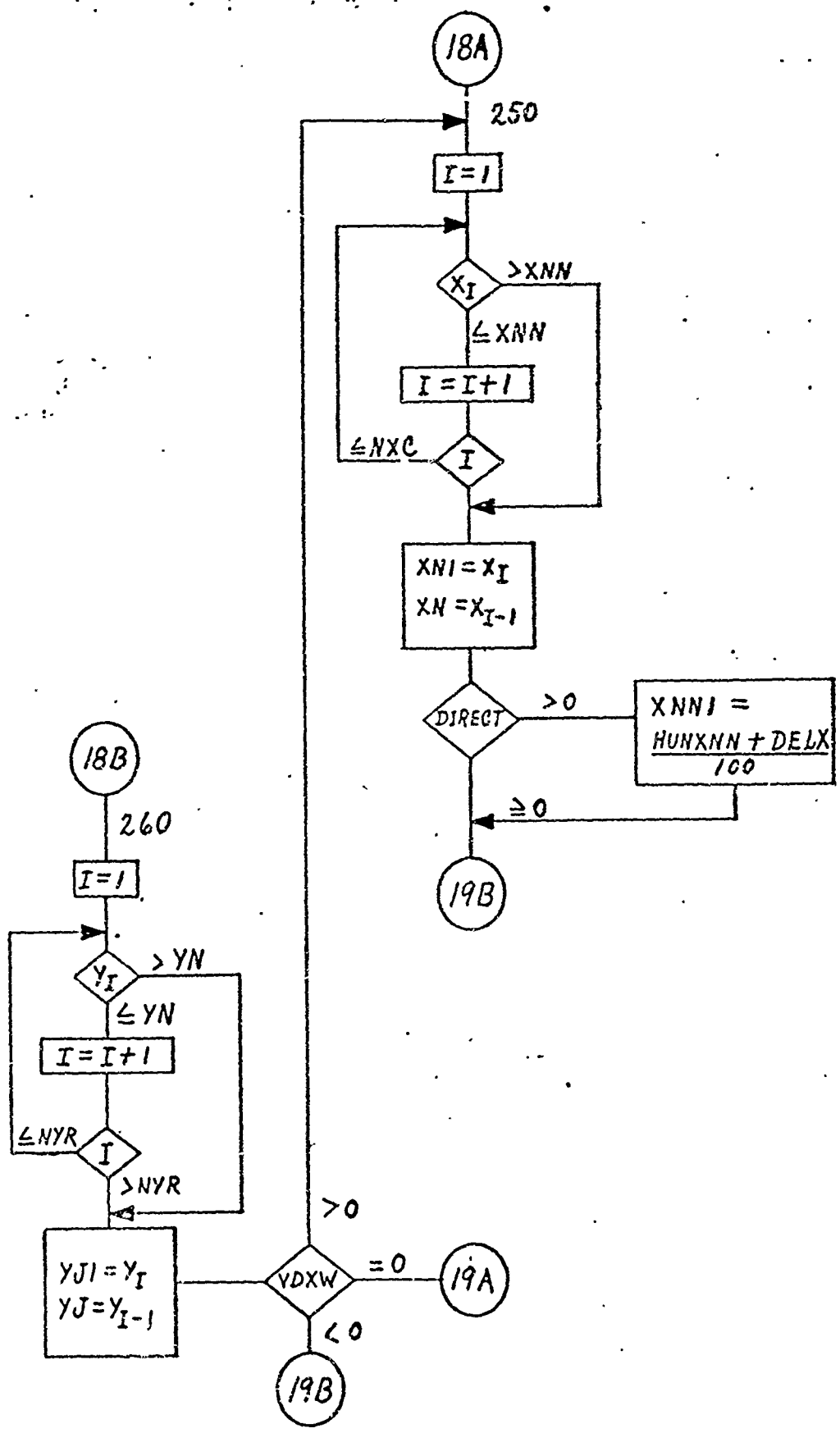

13A
JP
≠0
IP = IP+1
IPC_IP = JP
IPA = IPA+JP
JP = 0
TA7 = ID
DIRECT=STRDIR
MPLOTS
(TA, XPLOT, YPLOT)
13B
104
VDXW = VXW/UINF
VDYW = VYW/UINF
PRINT OUT XL, YDROP, VDXW, VDYW
YM = YDROP
YN = |YM|
WRITE HEADING
HUNXNN = 100(XL)
XNN = XL
XNN1 = XNN
XPLOT(1) = XNN
XPLOT(1) = YM
14A

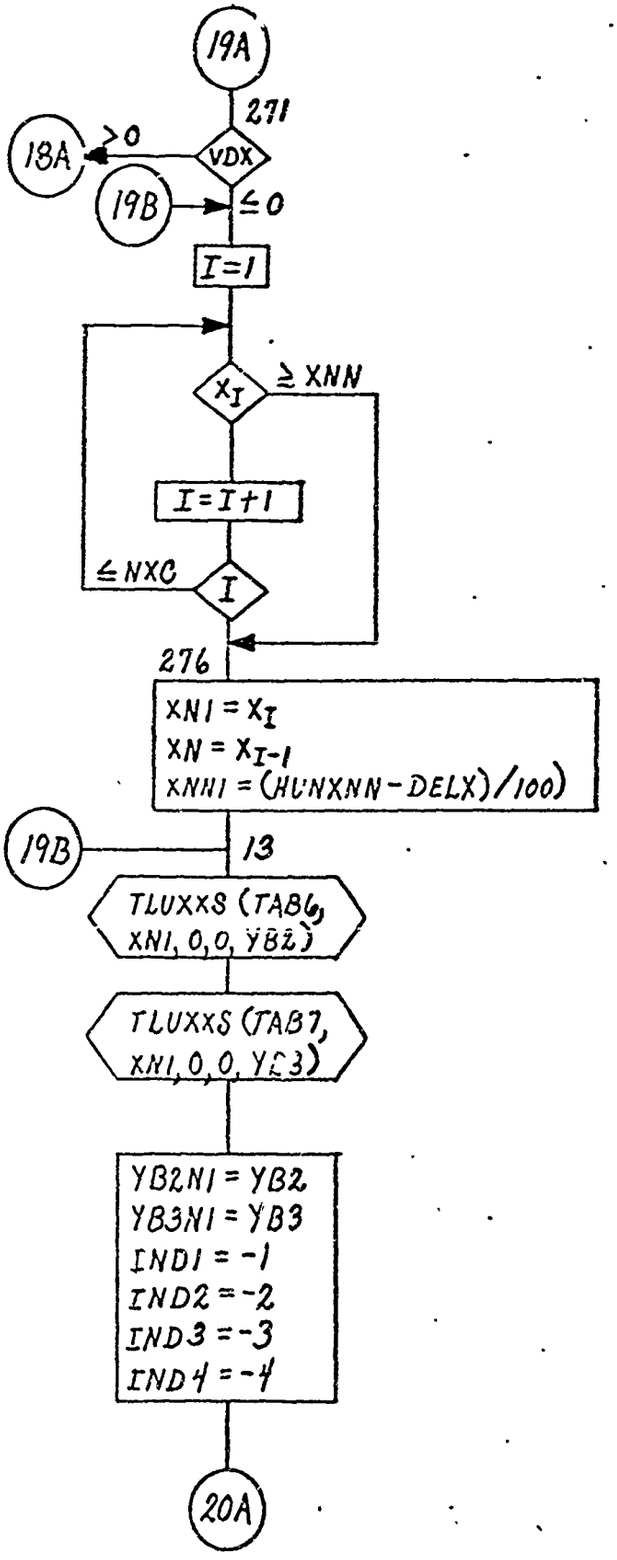

14A
XNN ≥ T5
NO
YES
YM ≥ T6
NO
YES
XNN ≤ T1
NO
YES
YM ≤ T2
NO
YES
15A
3410
JP = JP+1
IPB = IPA+JP
IPB
≥ 3099
WRITE "TOO MANY POINTS IN RESTRICTED AREA PLOT"
IPB = IPB-1
XPLT(IPB) = XNN
YPLT(IPB) = YM
15A

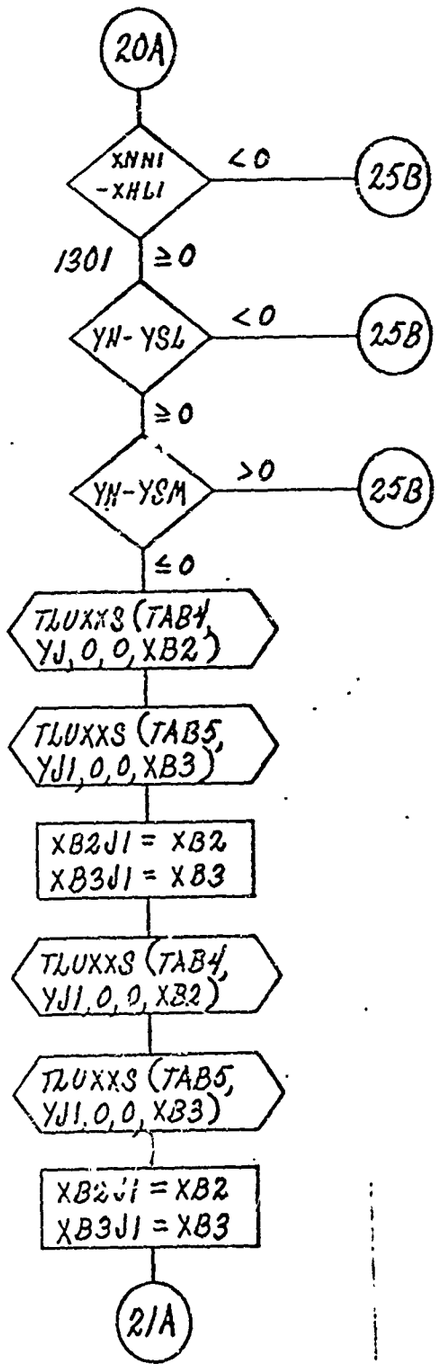

15A
ID = 2
XPLT_ID = XNN
YPLT_ID = YM
XNN
>XL
168
103
≤XL
NYR = TAB3_2 - 1
NXC = TAB3_3 - 1
I = 1
Y_I = (TAB3_I+4)(10000) + .5 / 10000
TAB3_I+4 = Y_I
I = I + 1
≤NYR
I
>NYR
16A

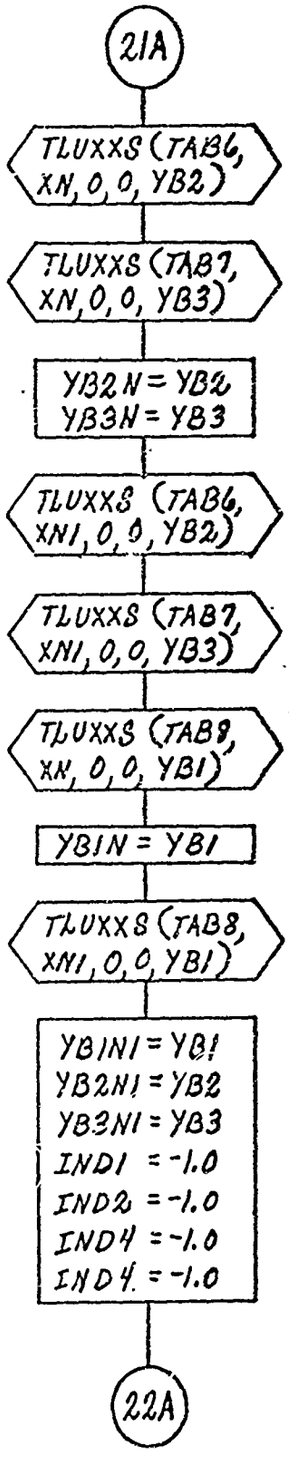

16A
J=4
I=1
J=J+NYR-1
$X_I = \frac{(TAB3_J)(10000)+0.5}{10000}$
$TAB3_J = X_I$
I=I+1
≤NXC
I
>NXC
16B
YN
>YMAX
1B
≤YMAX
DELX=DELX1
XNN
>XREF1
XNN
<XREF2
DELX=DELX2
16C
≤XREF1
≥XREF2
DIRECT
>0
18B
≤0
17A

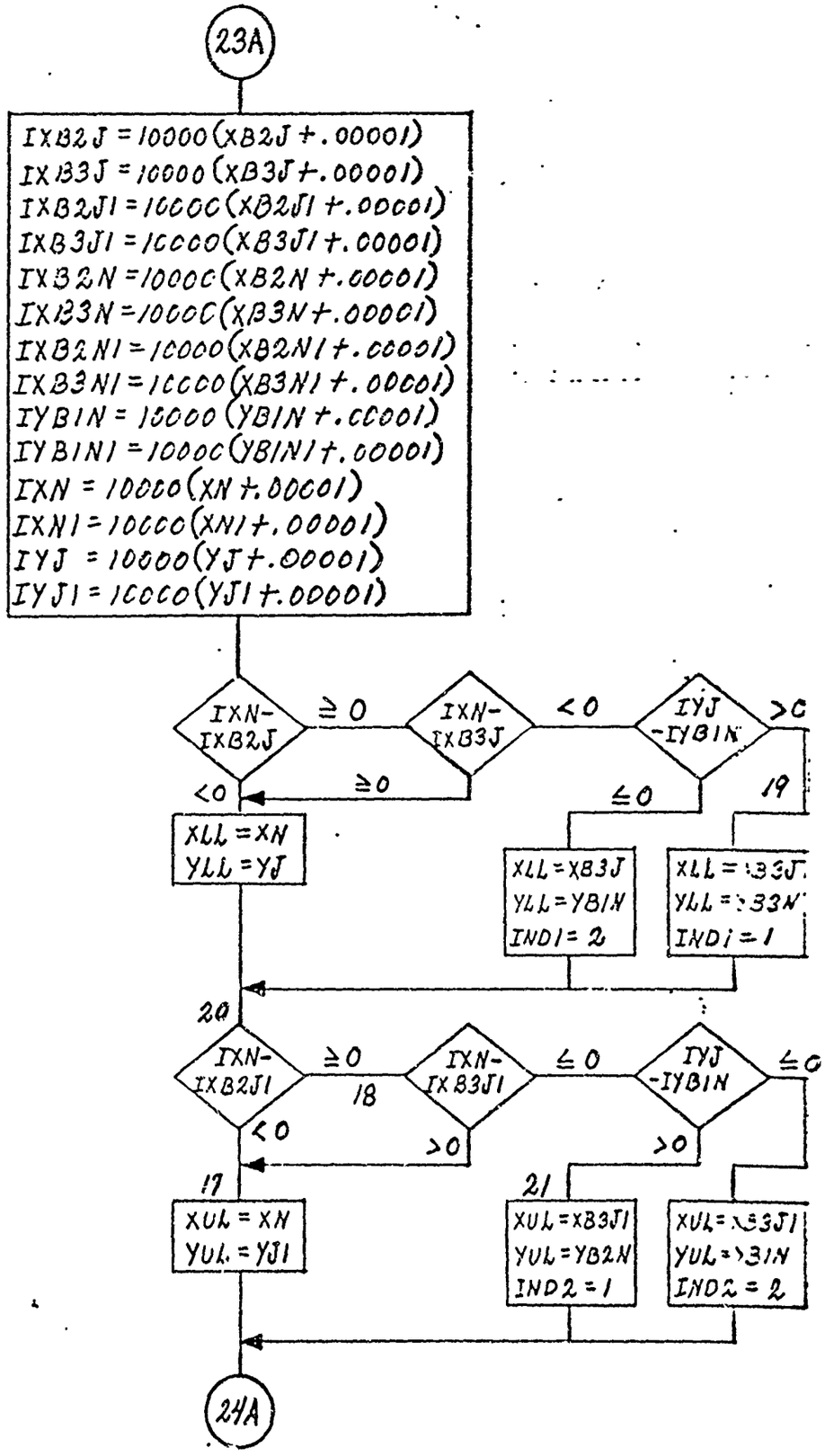

17A
225
VDYW
>0
YM
<0
17B
≤0
231
YM
≤0
I=1
17B
232
>0
I=1
Y_I
>YN
≤YN
I=I+1
≤NYR
I
>NYR
YMNI=Y_I
Y_I
≥YN
<YN
I=I+1
≤NYR
I
>NYR
229
YMNI=Y_{I-1}
YJ=Y_{I-1}
YM ≤$-Y_1$
YES
YMNI=-YMNI
|YMNI|
=Y_2
YDYW
>0
YMNI=|YMNI|
YJI=Y_I
18A

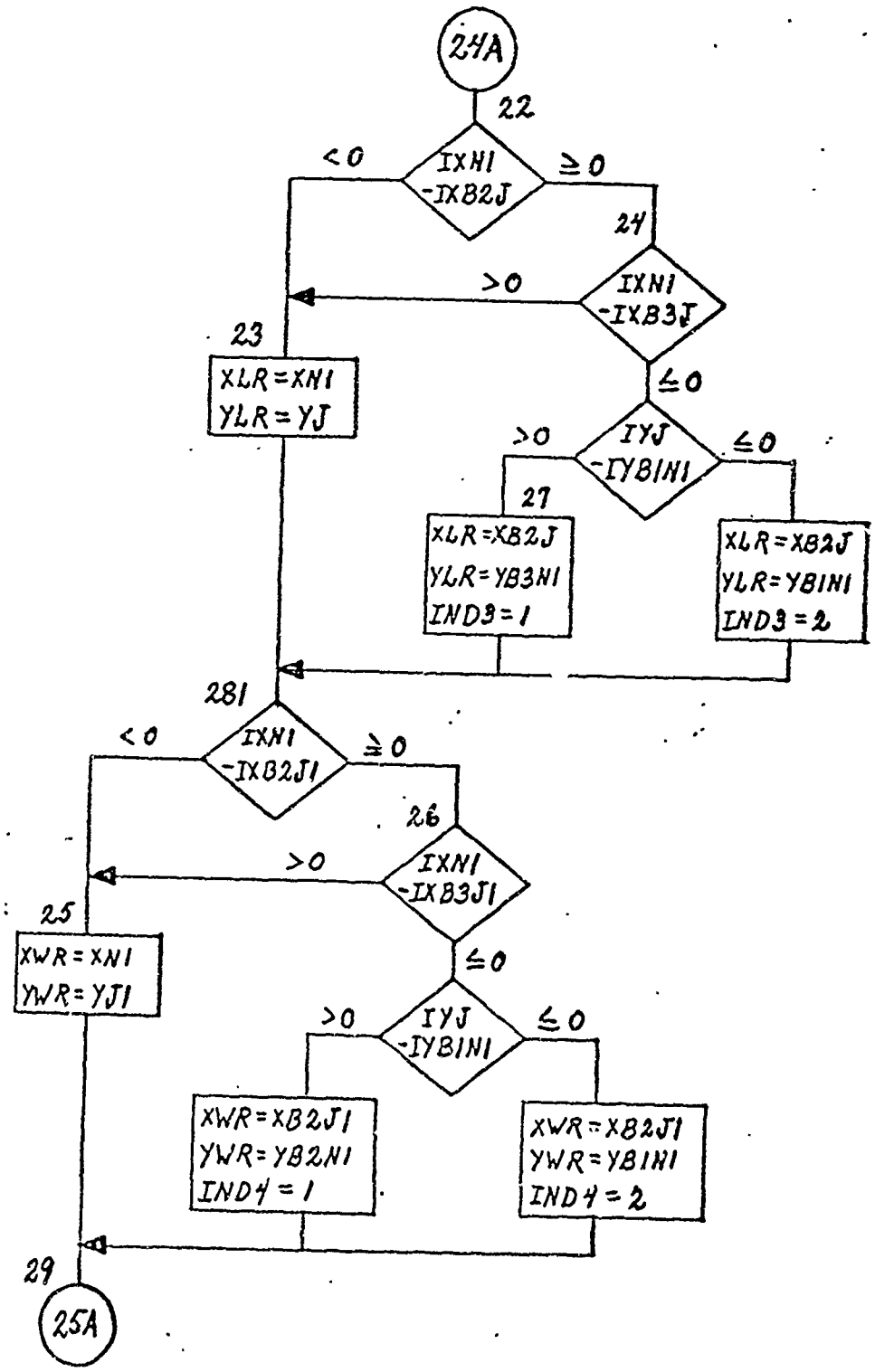

18A
250
I=1
X_I
>XNN
≤XNN
I=I+1
≤NXC
I
XNI=X_I
XN=X_{I-1}
DIRECT
>0
XNNI = (HUNXNN + DELX)/100
≥0
19B
18B
260
I=1
Y_I
>YN
≤YN
I=I+1
≤NYR
I
>NYR
YJI=Y_I
YJ=Y_{I-1}
VDXW
>0
=0
19A
<0
19B

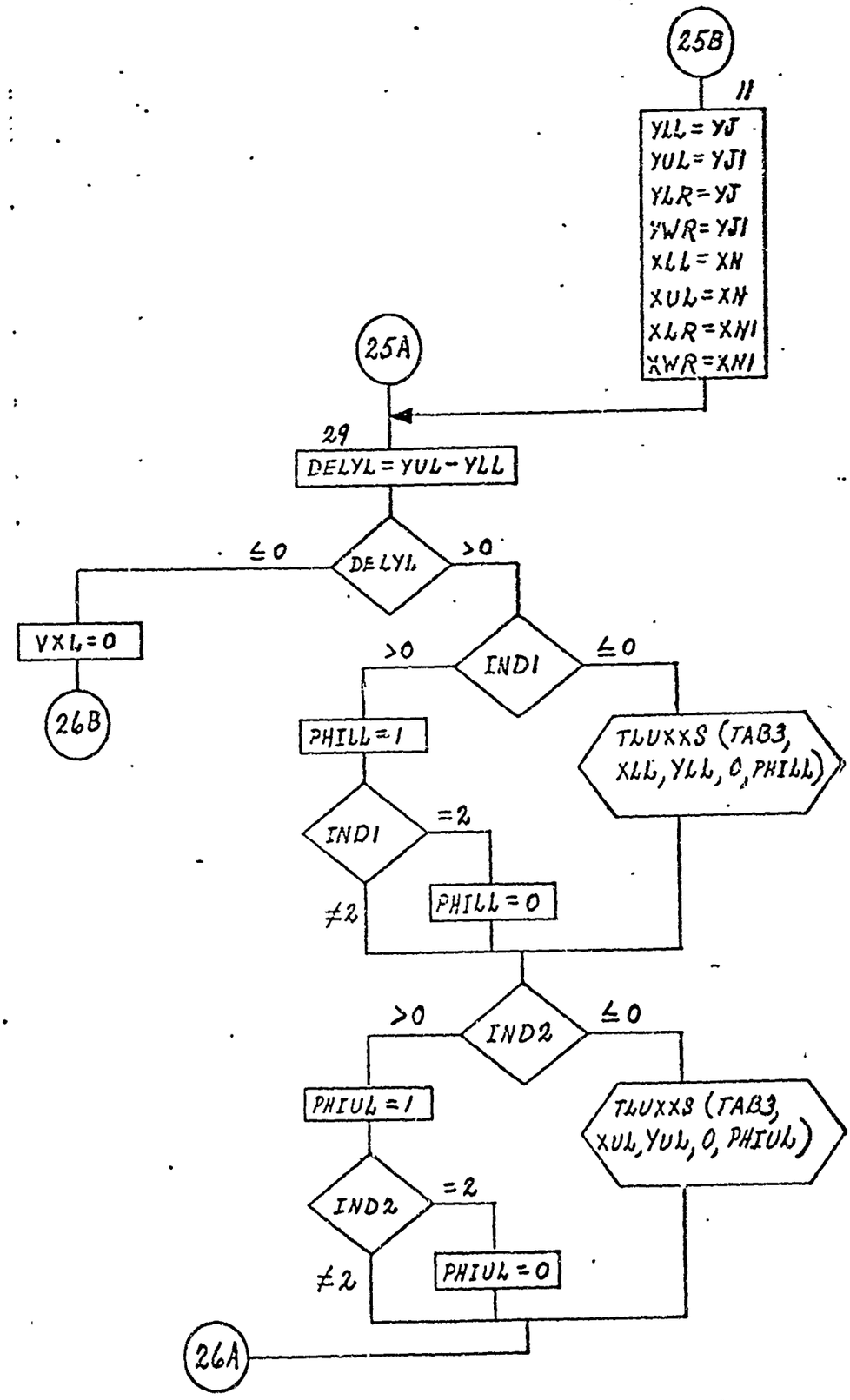

19A
271
VDX
>0
18A
19B
≤0
I=1
X_I
≥ XNN
I=I+1
≤NXC
I
276
XN1 = X_I
XN = X_{I-1}
XNN1 = (HUNXNN-DELX)/100)
19B
13
TLUXXS (TAB6, XN1, 0,0, YB2)
TLUXXS (TAB7, XN1, 0,0, YB3)
YB2N1 = YB2
YB3N1 = YB3
IND1 = -1
IND2 = -2
IND3 = -3
IND4 = -4
20A

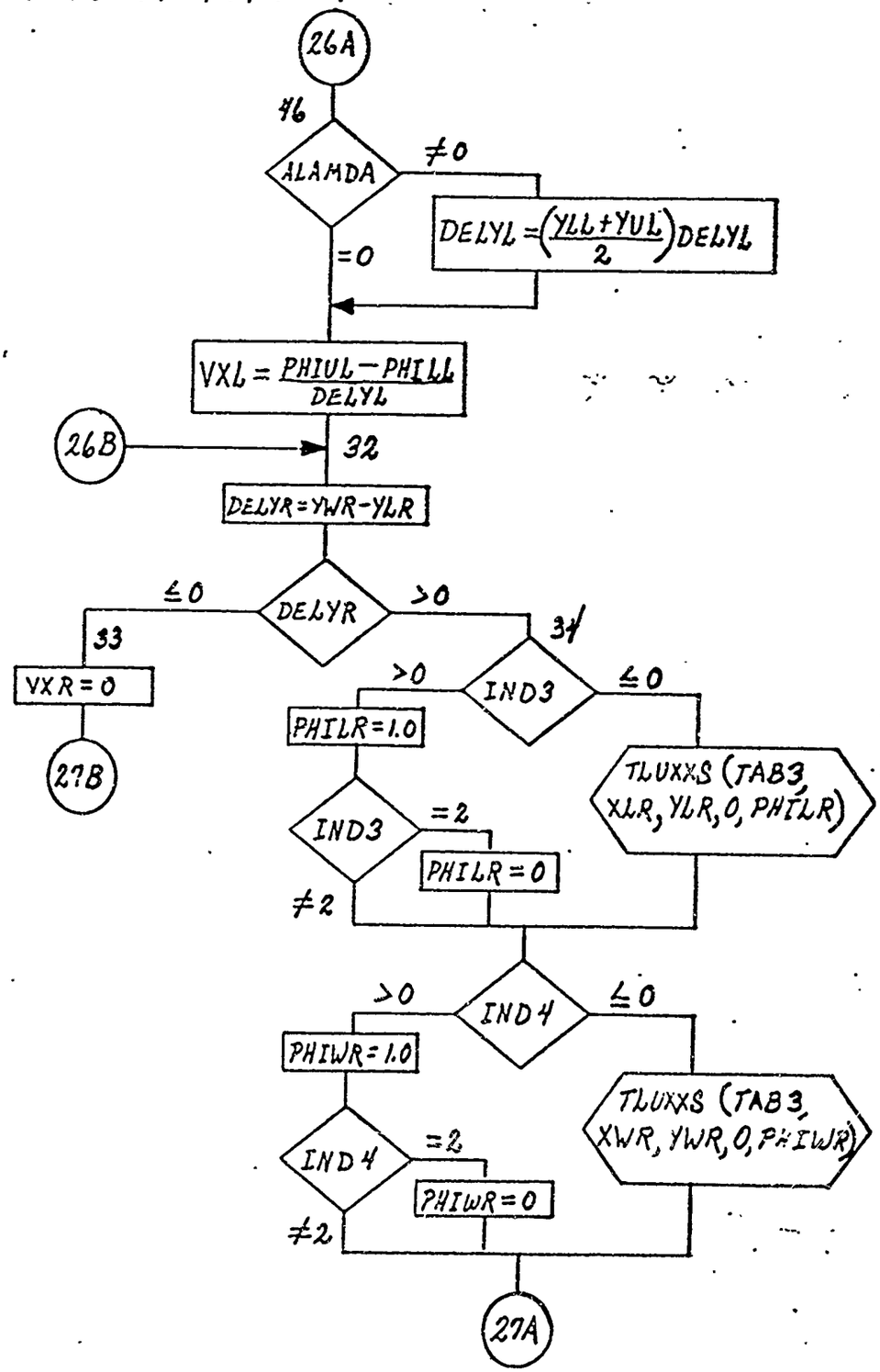

20A
XNNI -XHLI
< 0
25B
1301
≥ 0
YN- YSL
< 0
25B
≥ 0
YN-YSM
> 0
25B
≤ 0
TLUXXS (TAB4, YJ, 0, 0, XB2)
TLUXXS (TAB5, YJ1, 0, 0, XB3)
XB2J1 = XB2
XB3J1 = XB3
TLUXXS (TAB4, YJ1, 0, 0, XB2)
TLUXXS (TAB5, YJ1, 0, 0, XB3)
XB2J1 = XB2
XB3J1 = XB3
21A

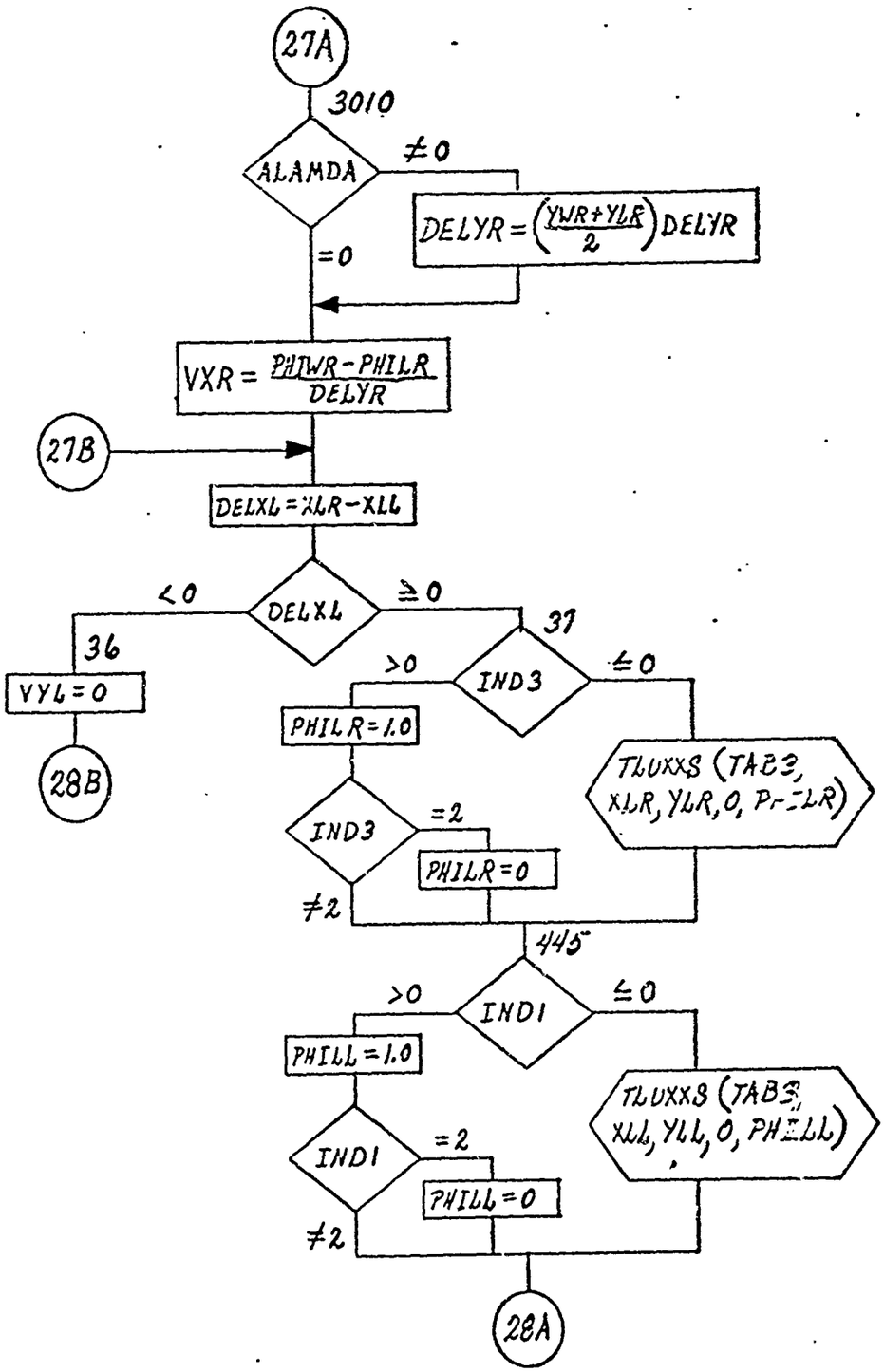

21A
TLUXXS (TAB6, XN, 0, 0, YB2)
TLUXXS (TAB7, XN, 0, 0, YB3)
YB2N = YB2
YB3N = YB3
TLUXXS (TAB6, XN1, 0, 0, YB2)
TLUXXS (TAB7, XN1, 0, 0, YB3)
TLUXXS (TAB8, XN, 0, 0, YB1)
YB1N = YB1
TLUXXS (TAB8, XN1, 0, 0, YB1)
YB1N1 = YB1
YB2N1 = YB2
YB3N1 = YB3
IND1 = -1.0
IND2 = -1.0
IND4 = -1.0
IND4 = -1.0
22A

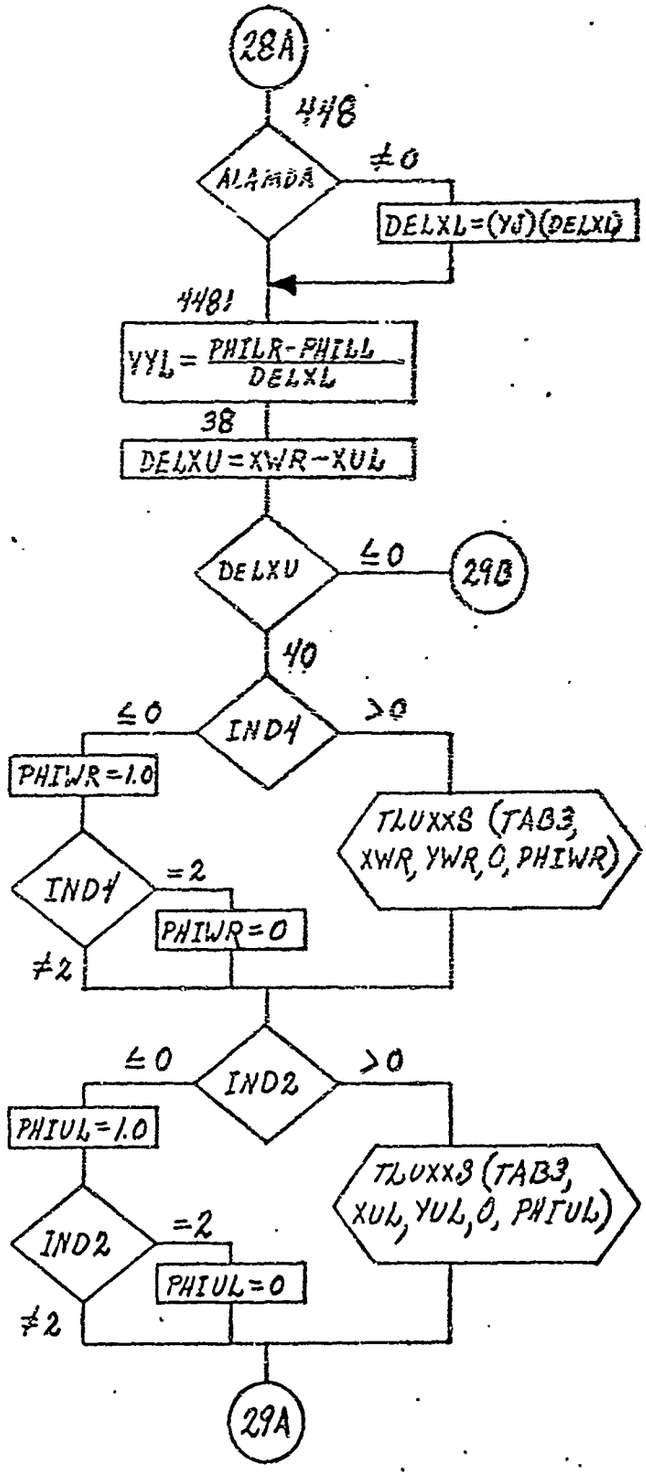

23A
IXB2J = 10000(XB2J + .00001)
IXB3J = 10000(XB3J + .00001)
IXB2J1 = 10000(XB2J1 + .00001)
IXB3J1 = 10000(XB3J1 + .00001)
IXB2N = 10000(XB2N + .00001)
IXB3N = 10000(XB3N + .00001)
IXB2N1 = 10000(XB2N1 + .00001)
IXB3N1 = 10000(XB3N1 + .00001)
IYBIN = 10000(YBIN + .00001)
IYBIN1 = 10000(YBIN1 + .00001)
IXN = 10000(XN + .00001)
IXN1 = 10000(XN1 + .00001)
IYJ = 10000(YJ + .00001)
IYJ1 = 10000(YJ1 + .00001)
IXN-IXB2J
≧ 0
IXN-IXB3J
< 0
IYJ-IYBIN
> 0
< 0
≧ 0
≦ 0
19
XLL = XN
YLL = YJ
XLL = XB3J
YLL = YBIN
IND1 = 2
XLL = XB3J
YLL = YB3N
IND1 = 1
20
IXN-IXB2J1
≧ 0
18
IXN-IXB3J1
≦ 0
IYJ-IYBIN
≦ 0
< 0
> 0
> 0
17
21
XUL = XN
YUL = YJ1
XUL = XB3J1
YUL = YB2N
IND2 = 1
XUL = XB3J1
YUL = YBIN
IND2 = 2
24A

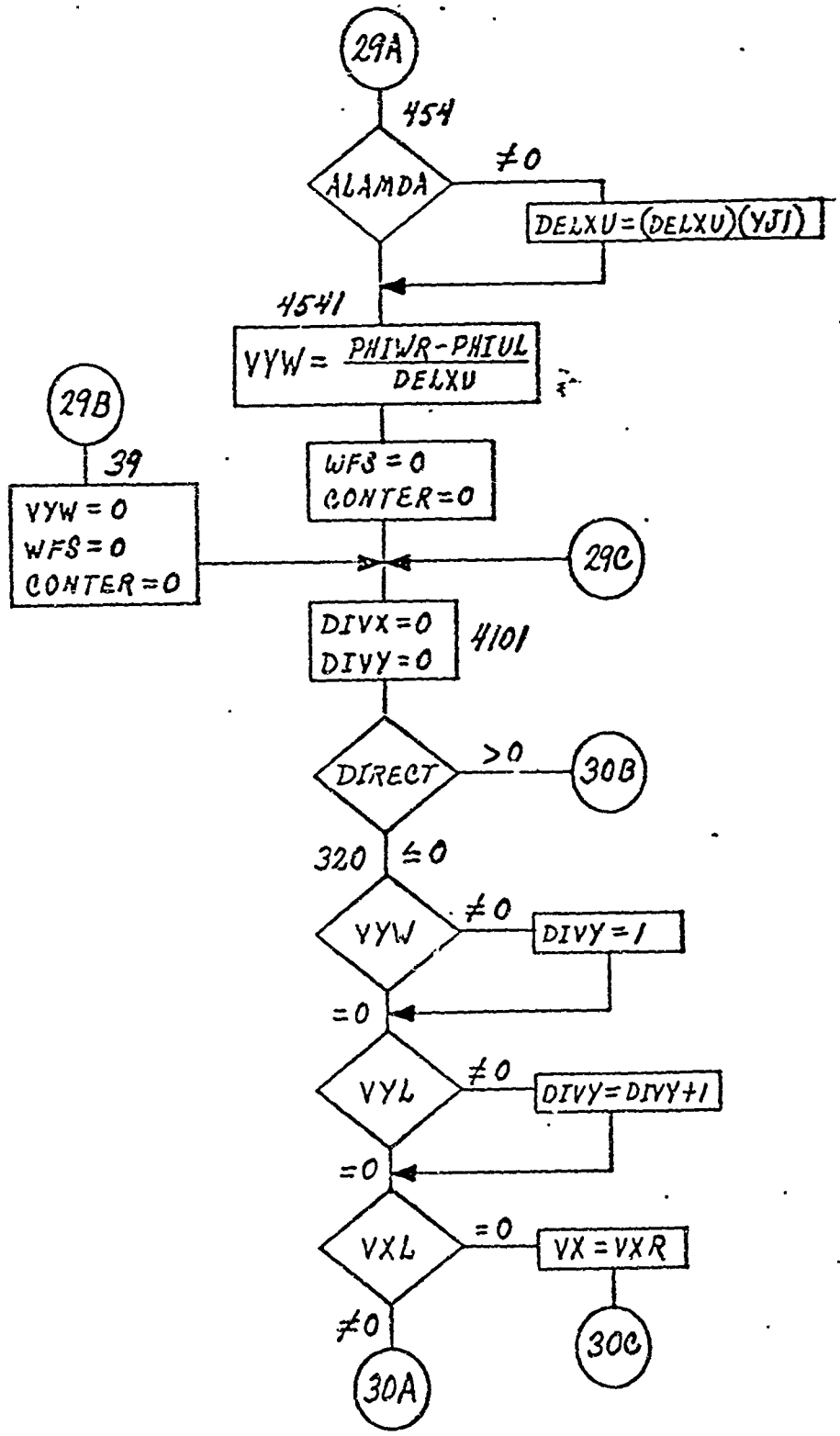

24A
22
IXN1 -IXB2J
< 0
≥ 0
24
IXN1 -IXB3J
> 0
23
XLR = XN1
YLR = YJ
≤ 0
IYJ -IYB1N1
> 0
≤ 0
27
XLR = XB2J
YLR = YB3N1
IND3 = 1
XLR = XB2J
YLR = YB1N1
IND3 = 2
281
IXN1 -IXB2J1
< 0
≥ 0
26
IXN1 -IXB3J1
> 0
25
XWR = XN1
YWR = YJ1
≤ 0
IYJ -IYB1N1
> 0
≤ 0
XWR = XB2J1
YWR = YB2N1
IND4 = 1
XWR = XB2J1
YWR = YB1N1
IND4 = 2
29
25A

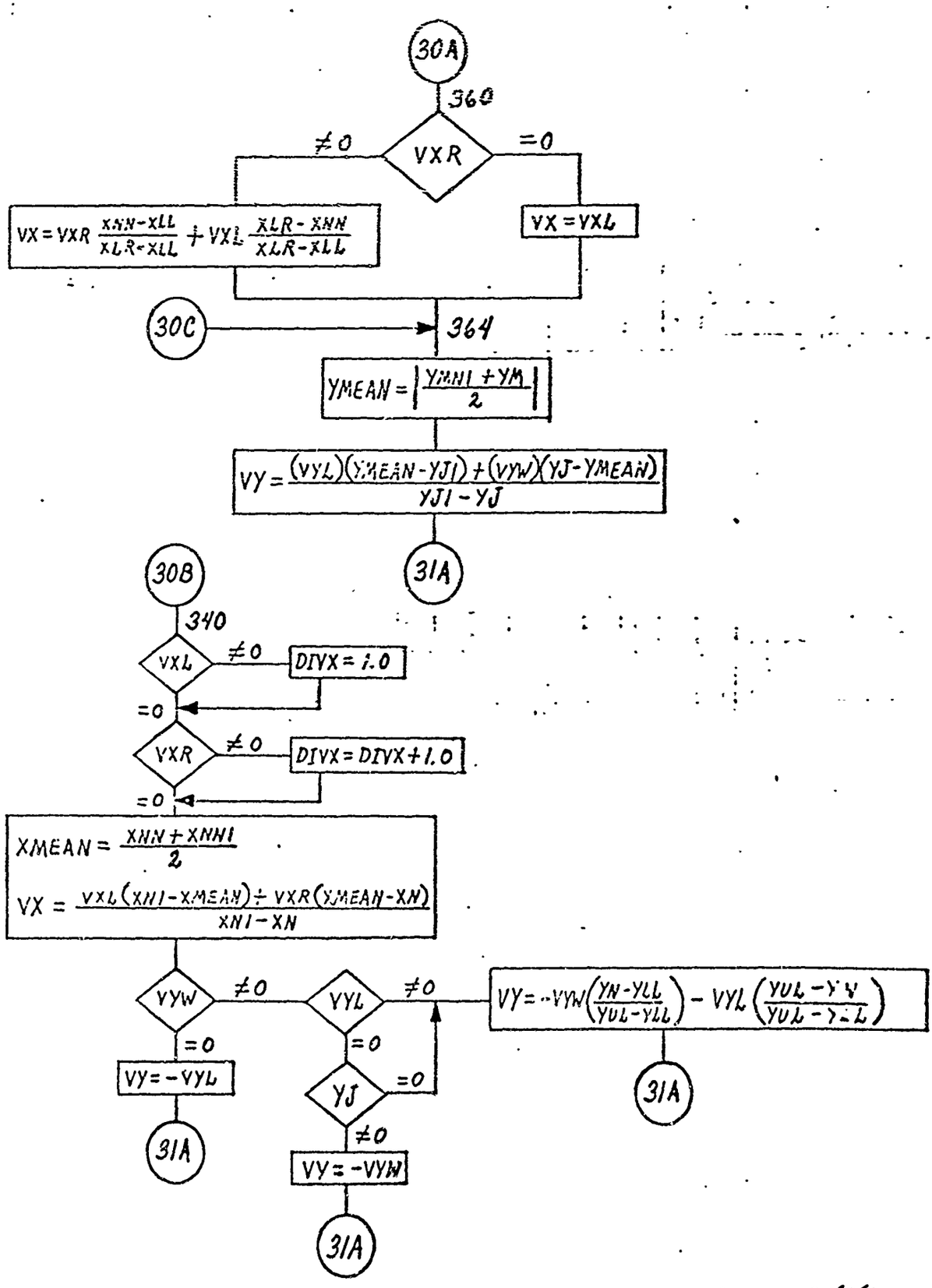

25B
11
YLL = YJ
YUL = YJ1
YLR = YJ
YWR = YJ1
XLL = XN
XUL = XN
XLR = XN1
XWR = XN1
25A
29
DELYL = YUL - YLL
DELYL
≤ 0
> 0
VXL = 0
26B
INDI
> 0
≤ 0
PHILL = 1
TLUXXS (TAB3, XLL, YLL, 0, PHILL)
INDI
= 2
≠ 2
PHILL = 0
IND2
> 0
≤ 0
PHIUL = 1
TLUXXS (TAB3, XUL, YUL, 0, PHIUL)
IND2
= 2
≠ 2
PHIUL = 0
26A

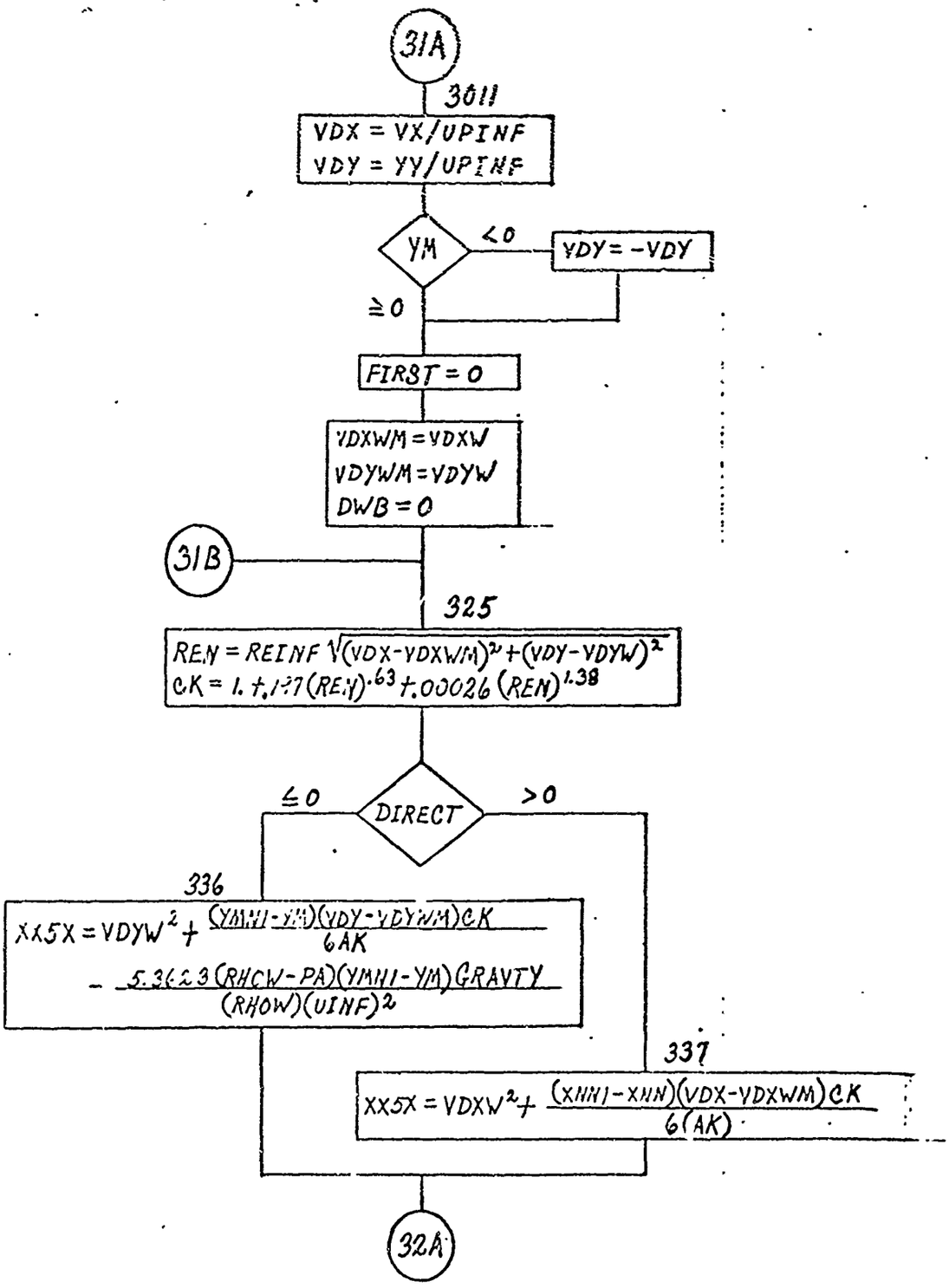

26A
46
ALAMDA
≠0
DELYL = ((YLL+YUL)/2) DELYL
=0
VXL = (PHIUL − PHILL)/DELYL
26B
32
DELYR=YWR-YLR
≤0
DELYR
>0
33
31
VXR=0
27B
>0
IND3
≤0
PHILR=1.0
TLUXXS (TAB3, XLR, YLR, 0, PHILR)
IND3
=2
PHILR=0
≠2
>0
IND4
≤0
PHIWR=1.0
TLUXXS (TAB3, XWR, YWR, 0, PHIWR)
IND4
=2
PHIWR=0
≠2
27A

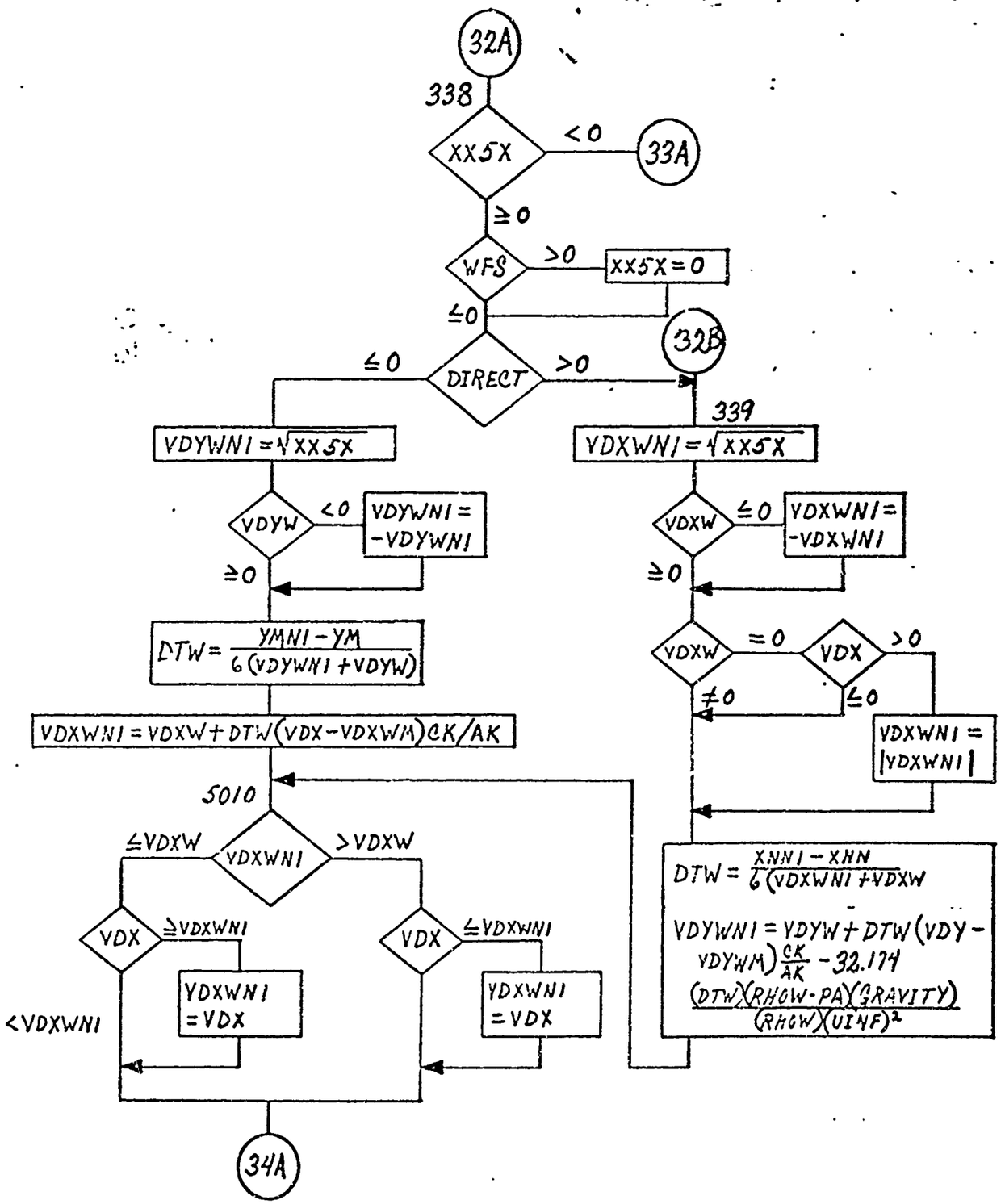

27A
3010
ALAMDA
≠0
DELYR = ((YWR+YLR)/2) DELYR
=0
VXR = (PHIWR-PHILR)/DELYR
27B
DELXL=XLR-XLL
DELXL
<0
≥0
36
VYL=0
28B
37
IND3
>0
≤0
PHILR=1.0
TLUXXS (TAB3, XLR, YLR, 0, PHILR)
IND3
=2
PHILR=0
≠2
445
IND1
>0
≤0
PHILL=1.0
TLUXXS (TAB3, XLL, YLL, 0, PHILL)
IND1
=2
PHILL=0
≠2
28A

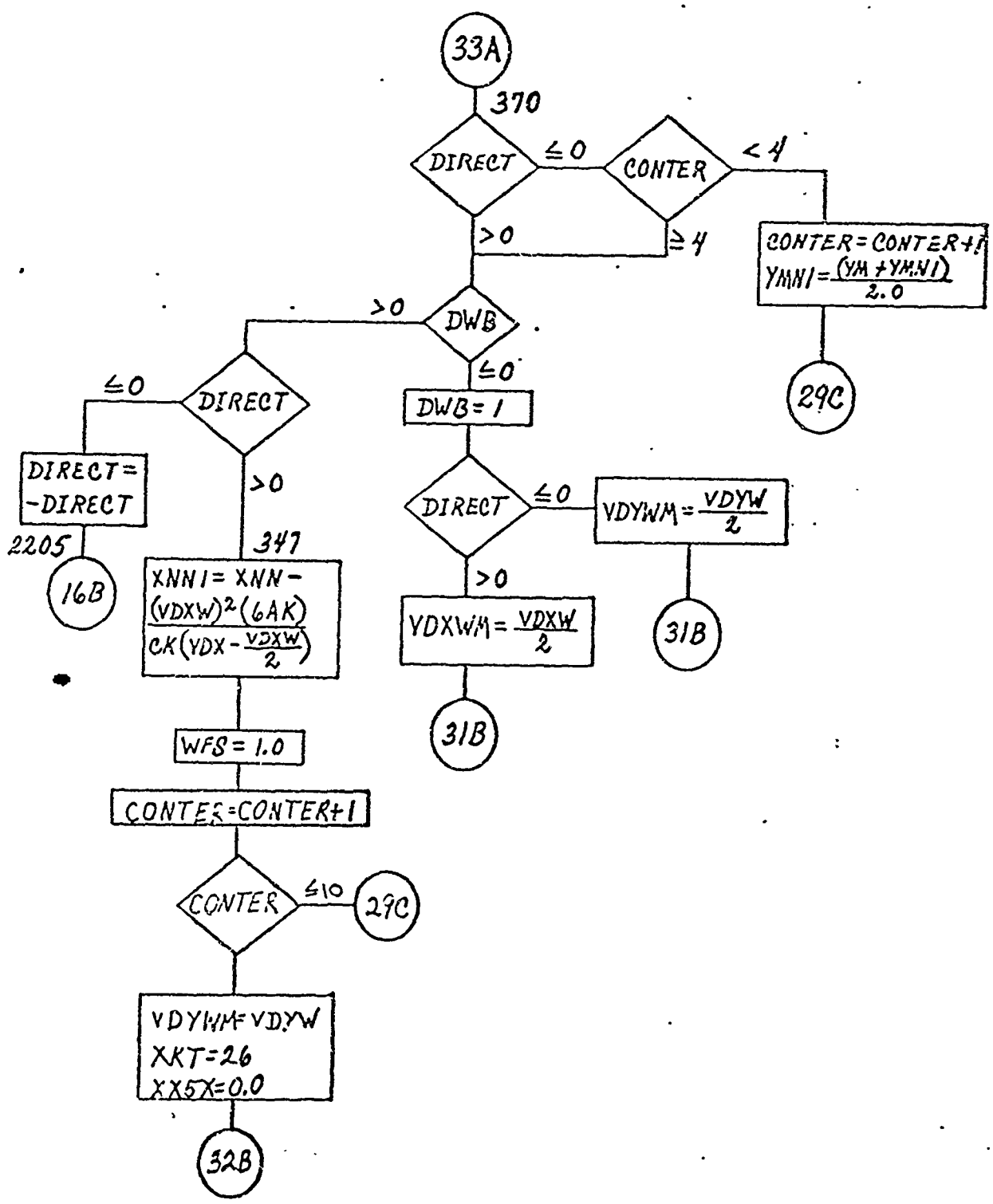

28A
448
ALAMDA
≠0
DELXL=(YJ)(DELXU)
4481
YYL = PHILR-PHILL / DELXL
38
DELXU=XWR-XUL
DELXU
≤0
29B
40
IND4
≤0
>0
PHIWR=1.0
TLUXXS (TAB3, XWR, YWR, 0, PHIWR)
IND4
=2
PHIWR=0
≠2
IND2
≤0
>0
PHIUL=1.0
TLUXXS (TAB3, XUL, YUL, 0, PHIUL)
IND2
=2
PHIUL=0
≠2
29A

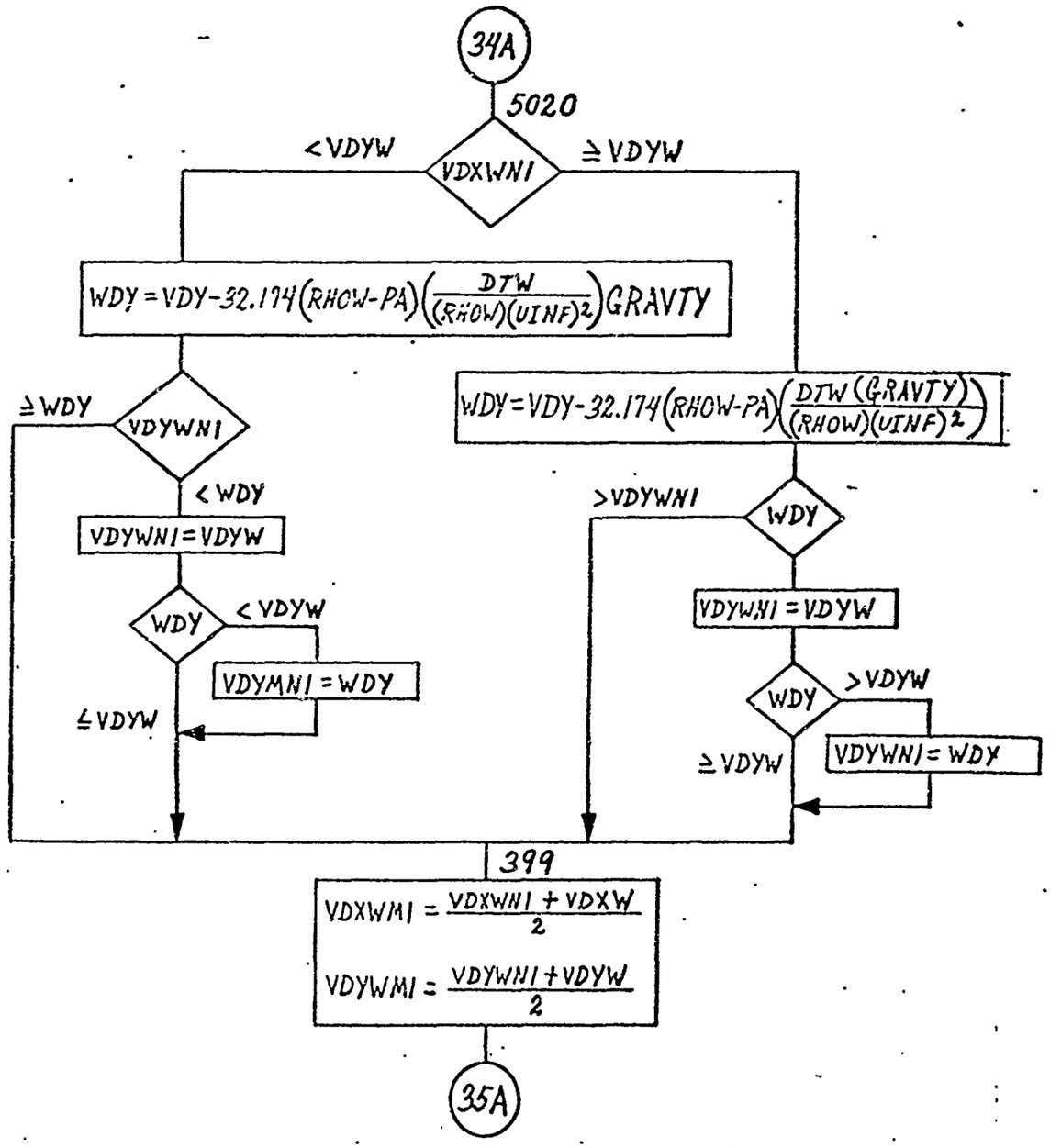

29A
454
ALAMDA
≠0
DELXU=(DELXU)(YJI)
4541
VYW = PHIWR-PHIUL / DELXU
29B
39
VYW = 0
WFS = 0
CONTER = 0
WFS = 0
CONTER = 0
29C
DIVX = 0
DIVY = 0
4101
DIRECT
>0
30B
320
≤0
VYW
≠0
DIVY = 1
=0
VYL
≠0
DIVY = DIVY+1
=0
VXL
=0
VX = VXR
≠0
30A
30C

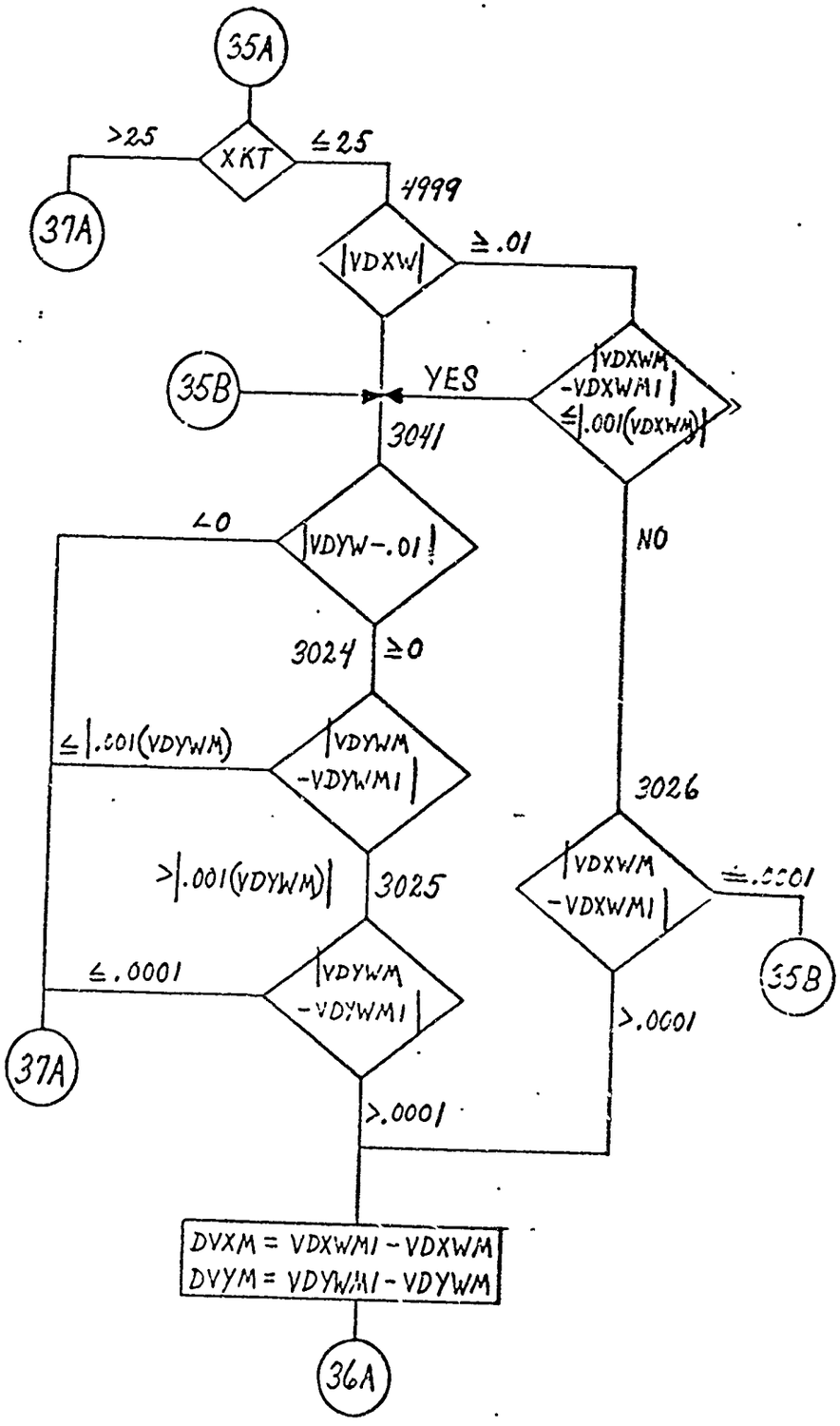

30A
360
VXR
≠0
=0
VX = VXR (XNN-XLL)/(XLR-XLL) + VXL (XLR-XNN)/(XLR-XLL)
VX = VXL
30C
364
YMEAN = |(YNNI + YM)/2|
VY = (VYL)(YMEAN-YJI) + (VYW)(YJ-YMEAN) / (YJI - YJ)
31A
30B
340
VXL
≠0
DIVX = 1.0
=0
VXR
≠0
DIVX = DIVX + 1.0
=0
XMEAN = (XNN + XNNI)/2
VX = (VXL(XNI-XMEAN) + VXR(XMEAN-XN)) / (XNI - XN)
VYW
≠0
VYL
≠0
VY = -VYW((YN-YLL)/(YUL-YLL)) - VYL((YUL-YN)/(YUL-YLL))
=0
VY = -VYL
31A
=0
YJ
=0
≠0
VY = -VYW
31A
31A

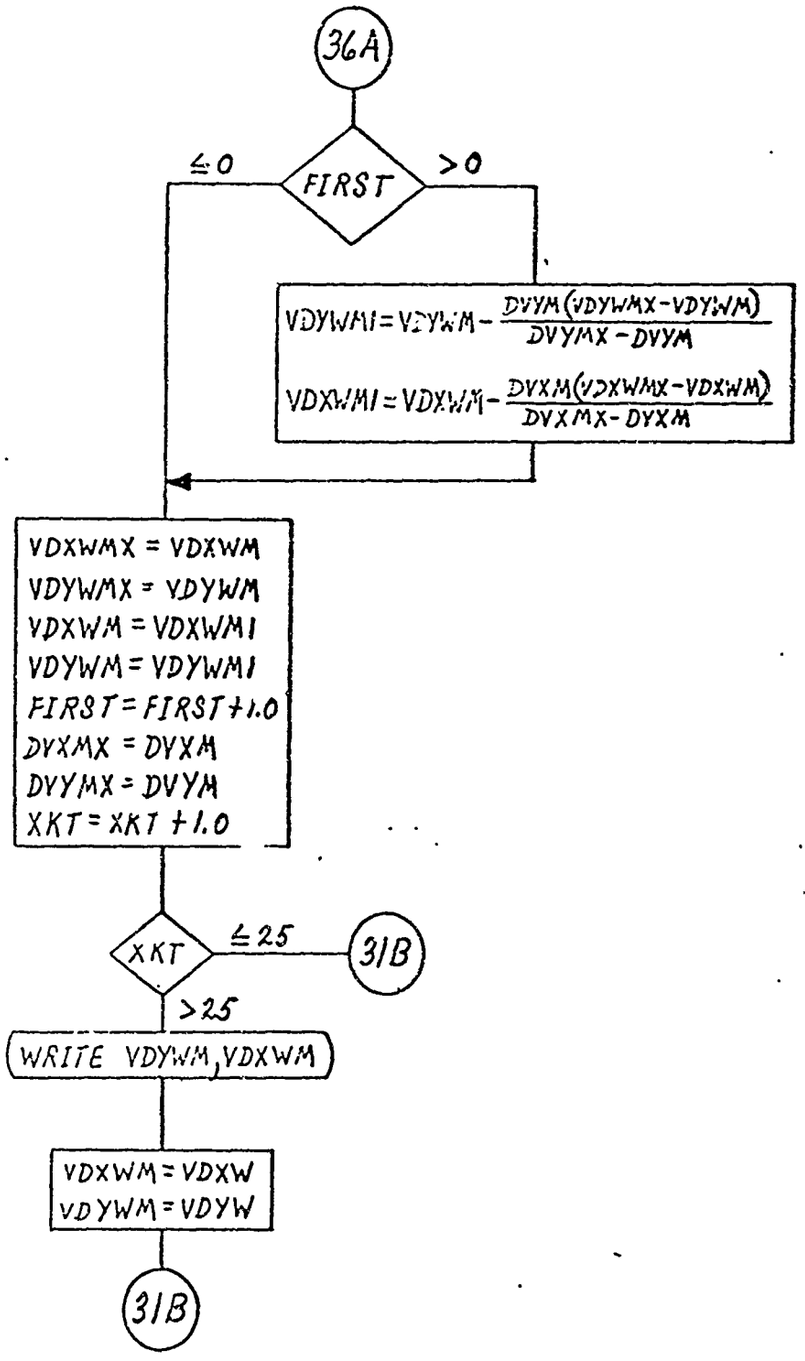

31A
3011
VDX = VX/UPINF
VDY = YY/UPINF
YM
<0
VDY = -VDY
≧0
FIRST = 0
VDXWM = VDXW
VDYWM = VDYW
DWB = 0
31B
325
REN = REINF √((VDX-VDXWM)² + (VDY-VDYW)²)
CK = 1. + .127 (REN)^.63 + .00026 (REN)^1.38
≦0
DIRECT
>0
336
XX5X = VDYW² + (YMHI-YM)(VDY-VDYWM)CK / 6AK
- 5.3623 (RHOW-PA)(YMHI-YM) GRAVTY / (RHOW)(UINF)²
337
XX5X = VDXW² + (XHHI-XHH)(VDX-VDXWM)CK / 6(AK)
32A

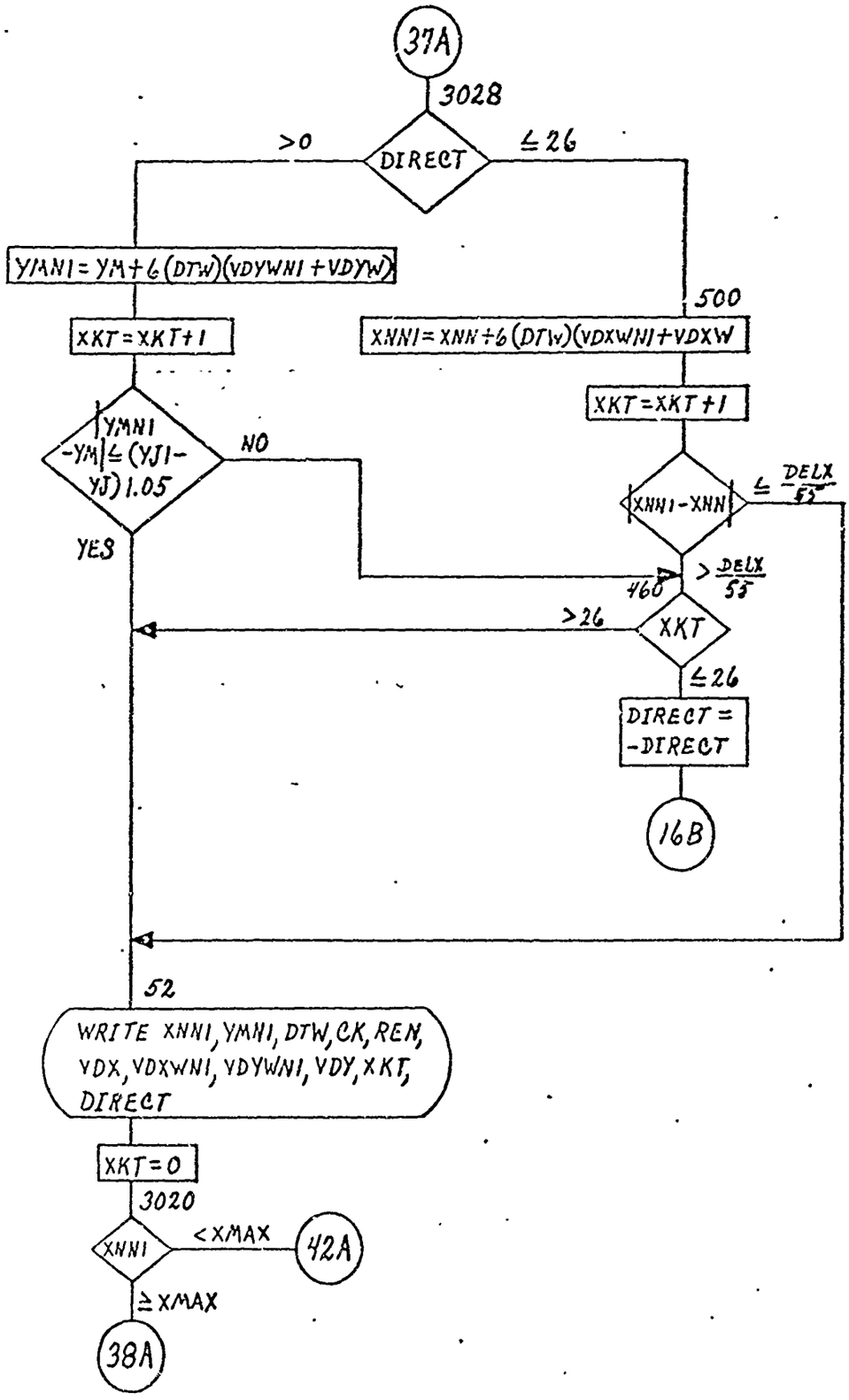

32A
338
XX5X
<0
33A
≥0
WFS
>0
XX5X=0
≤0
32B
≤0
DIRECT
>0
339
VDYWNI=√XX5X
VDXWNI=√XX5X
VDYW
<0
VDYWNI=
-VDYWNI
≥0
VDXW
≤0
VDXWNI=
-VDXWNI
≥0
DTW= (YMNI-YM)/6(VDYWNI+VDYW)
VDXW
=0
VDX
>0
≠0
≤0
VDXWNI=
|VDXWNI|
VDXWNI=VDXW+DTW(VDX-VDXWM)CK/AK
5010
≤VDXW
VDXWNI
>VDXW
DTW= (XNNI-XNN)/6(VDXWNI+VDXW
VDYWNI=VDYW+DTW(VDY-
VDYWM)CK/AK -32.174
(DTW)(RHOW-PA)(GRAVITY)/(RHOW)(UINF)²
VDX
≥VDXWNI
VDX
≤VDXWNI
<VDXWNI
VDXWNI
=VDX
VDXWNI
=VDX
34A

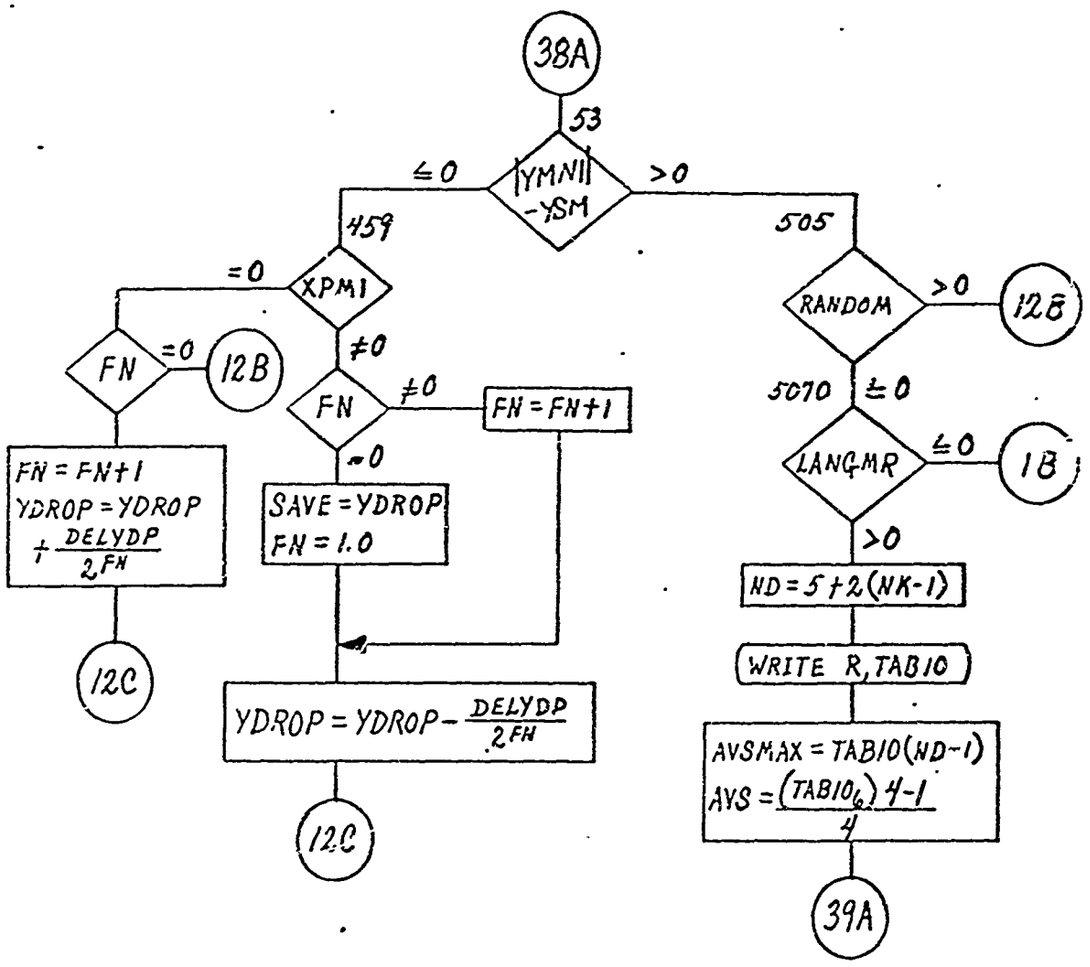

33A
370
DIRECT
≤0
CONTER
<4
>0
≥4
CONTER=CONTER+1
YMNI=(YM+YMNI)/2.0
29C
>0
DWB
≤0
DIRECT
≤0
DWB=1
DIRECT=
-DIRECT
2205
16B
>0
347
XNNI= XNN-
(VDXW)²(6AK)/CK(YDX-VDXW/2)
DIRECT
≤0
VDYWM=VDYW/2
31B
>0
VDXWM=VDXW/2
31B
WFS=1.0
CONTER=CONTER+1
CONTER
≤10
29C
VDYWM=VDYW
XKT=26
XX5X=0.0
32B

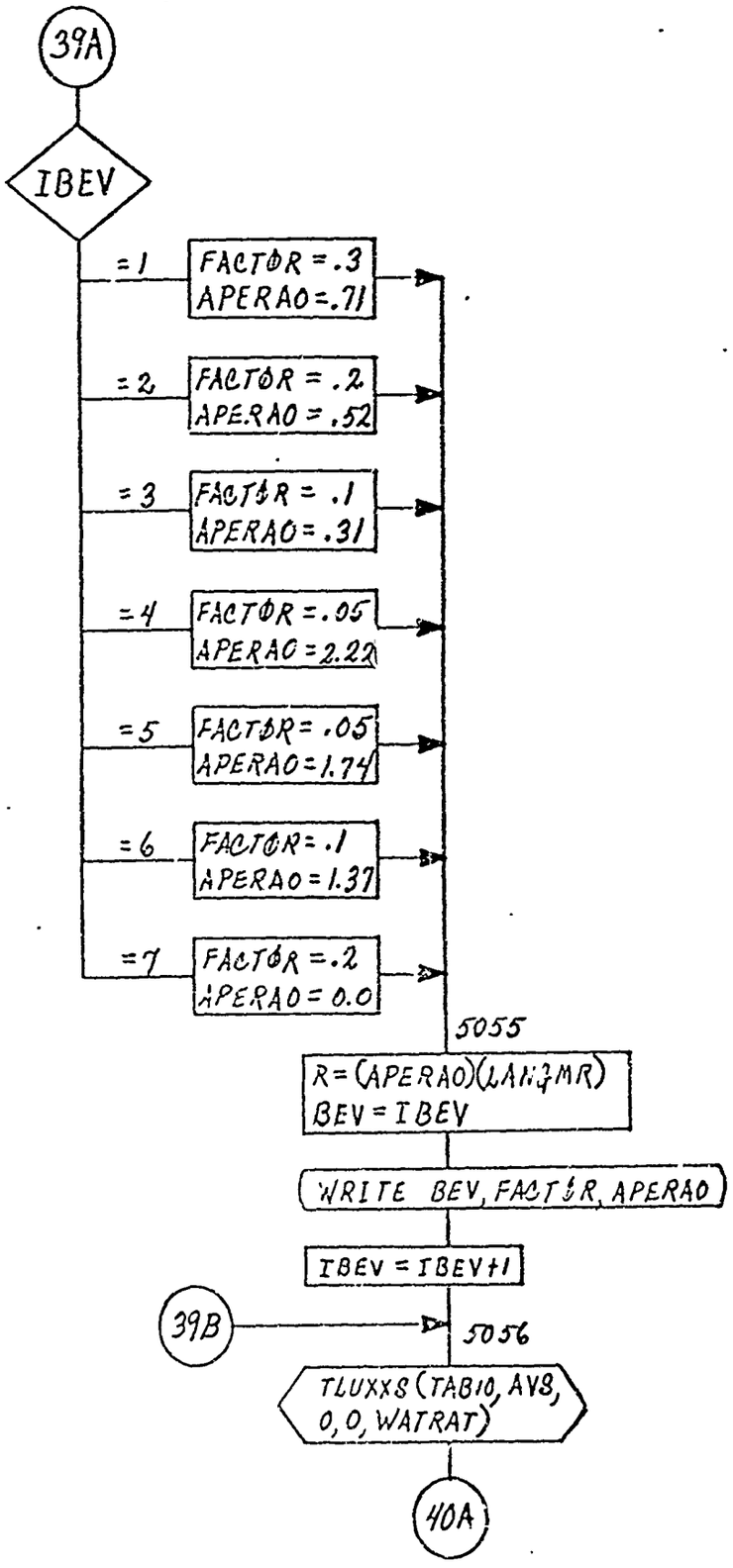

34A
5020
VDXWNI
< VDYW
≥ VDYW
WDY = VDY - 32.174 (RHOW - PA) (DTW / (RHOW)(UINF)²) GRAVTY
VDYWNI
≥ WDY
< WDY
VDYWNI = VDYW
WDY
< VDYW
VDYMNI = WDY
≤ VDYW
WDY = VDY - 32.174 (RHOW - PA) (DTW (GRAVTY) / (RHOW)(UINF)²)
WDY
> VDYWNI
VDYWNI = VDYW
WDY
> VDYW
VDYWNI = WDY
≥ VDYW
399
VDXWMI = (VDXWNI + VDXW) / 2
VDYWMI = (VDYWNI + VDYW) / 2
35A

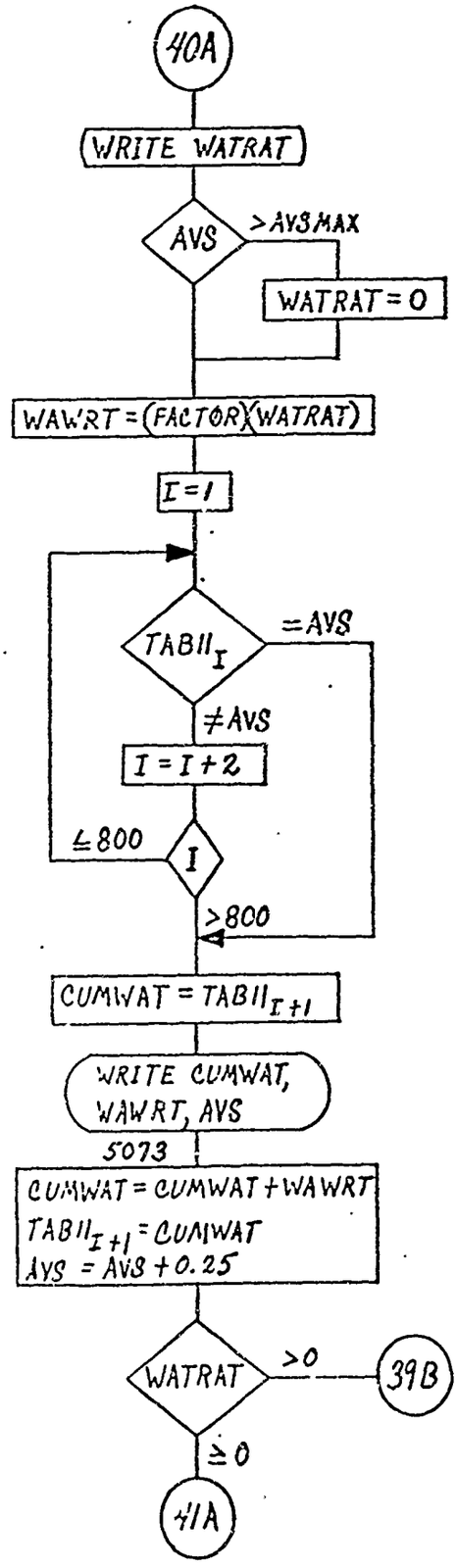

35A
XKT
>25
≤25
37A
4999
|VDXW|
≥.01
35B
YES
|VDXWM -VDXWM1| ≤|.001(VDXWM)|
3041
|VDYW-.01|
<0
NO
3024
≥0
|VDYWM -VDYWM1|
≤|.001(VDYWM)|
3026
|VDXWM -VDXWM1|
≤.0001
>|.001(VDYWM)|
3025
|VDYWM -VDYWM1|
≤.0001
>.0001
37A
35B
>.0001
DVXM = VDXWM1 - VDXWM
DVYM = VDYWM1 - VDYWM
36A

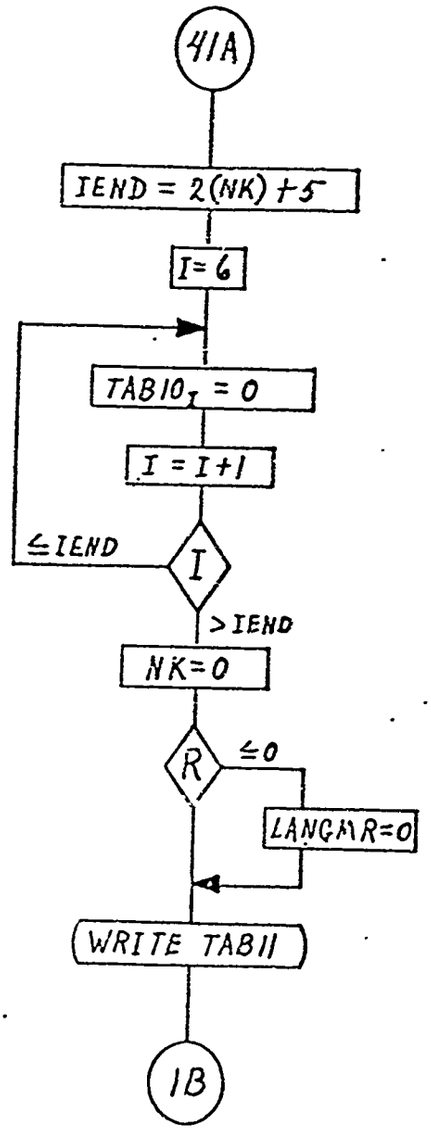

36A
FIRST
≤0
>0
VDYWMI = VDYWM − DVYM(VDYWMX − VDYWM) / DVYMX − DVYM
VDXWMI = VDXWM − DVXM(VDXWMX − VDXWM) / DVXMX − DVXM
VDXWMX = VDXWM
VDYWMX = VDYWM
VDXWM = VDXWMI
VDYWM = VDYWMI
FIRST = FIRST + 1.0
DVXMX = DVXM
DVYMX = DVYM
XKT = XKT + 1.0
XKT
≤25
31B
>25
WRITE VDYWM, VDXWM
VDXWM = VDXW
VDYWM = VDYW
31B

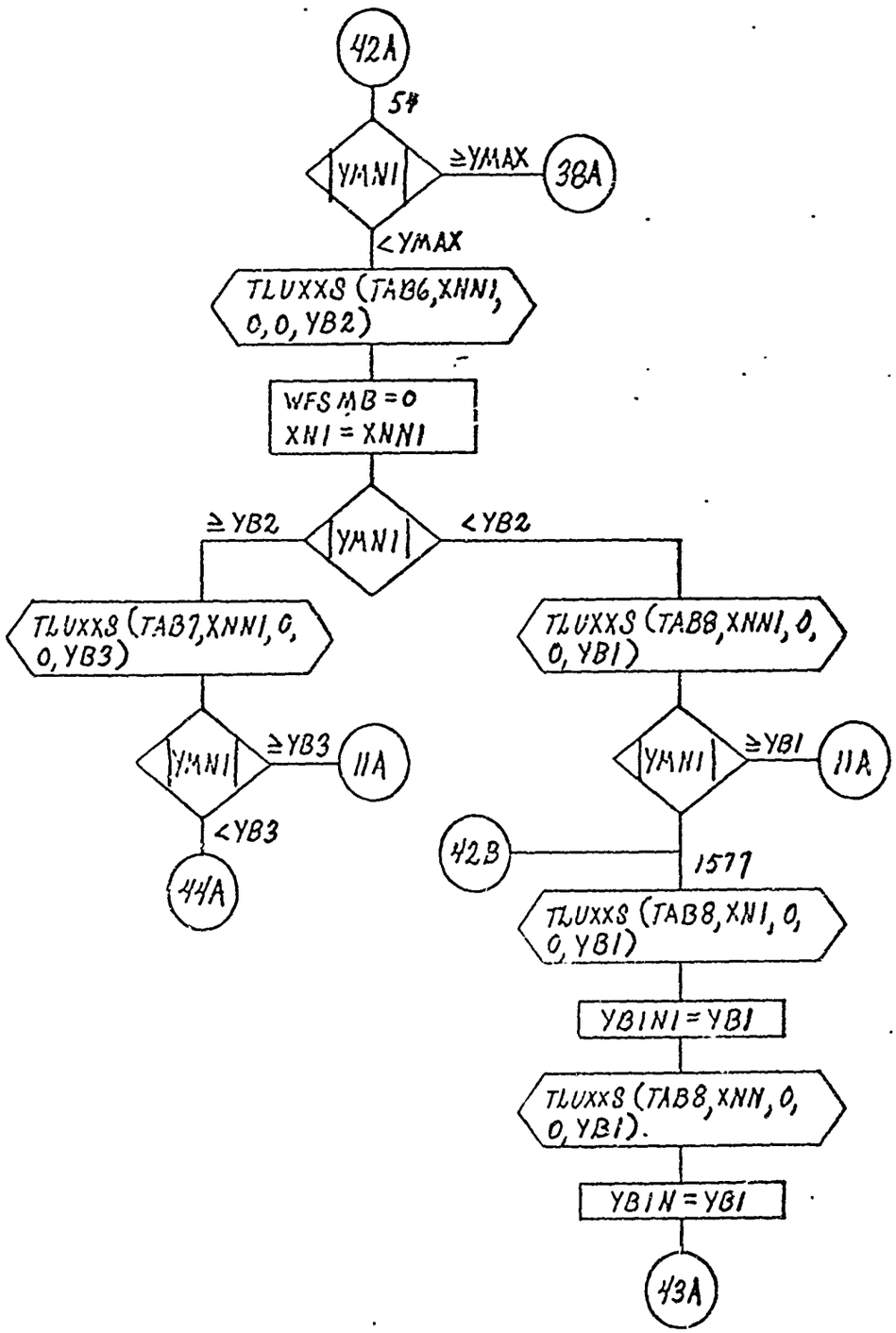

37A
3028
DIRECT
>0
≤26
YMNI = YM + 6 (DTW)(VDYWNI + VDYW)
XKT = XKT + 1
500
XNNI = XNN + 6 (DTW)(VDXWNI + VDXW
XKT = XKT + 1
|YMNI - YM| ≤ (YJ1 - YJ) 1.05
NO
YES
|XNNI - XNN| ≤ DELX/53
460
> DELX/53
>26
XKT
≤26
DIRECT = -DIRECT
16B
52
WRITE XNNI, YMNI, DTW, CK, REN, VDX, VDXWNI, VDYWNI, VDY, XKT, DIRECT
XKT = 0
3020
XNNI
< XMAX
42A
≥ XMAX
38A

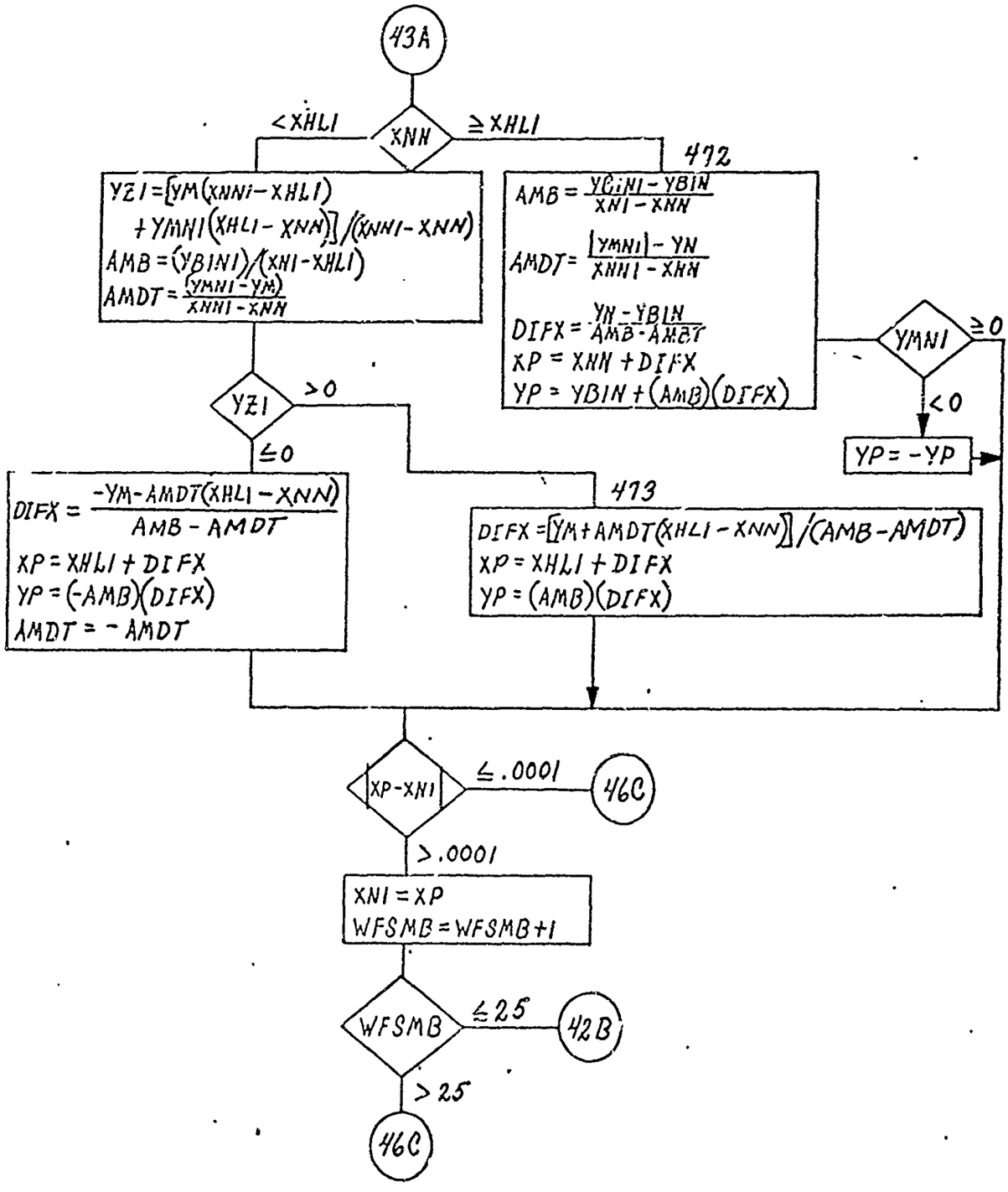

38A
53
|YMN1| -YSM
≤0
>0
459
505
XPM1
=0
≠0
FN
=0
12B
FN
≠0
FN = FN+1
=0
FN = FN+1
YDROP = YDROP + DELYDP / 2FN
SAVE = YDROP
FN = 1.0
12C
YDROP = YDROP - DELYDP / 2FN
12C
RANDOM
>0
12B
5070
≤0
LANGMR
≤0
1B
>0
ND = 5+2(NK-1)
WRITE R,TAB10
AVSMAX = TAB10(ND-1)
AVS = (TAB10_6) 4-1 / 4
39A

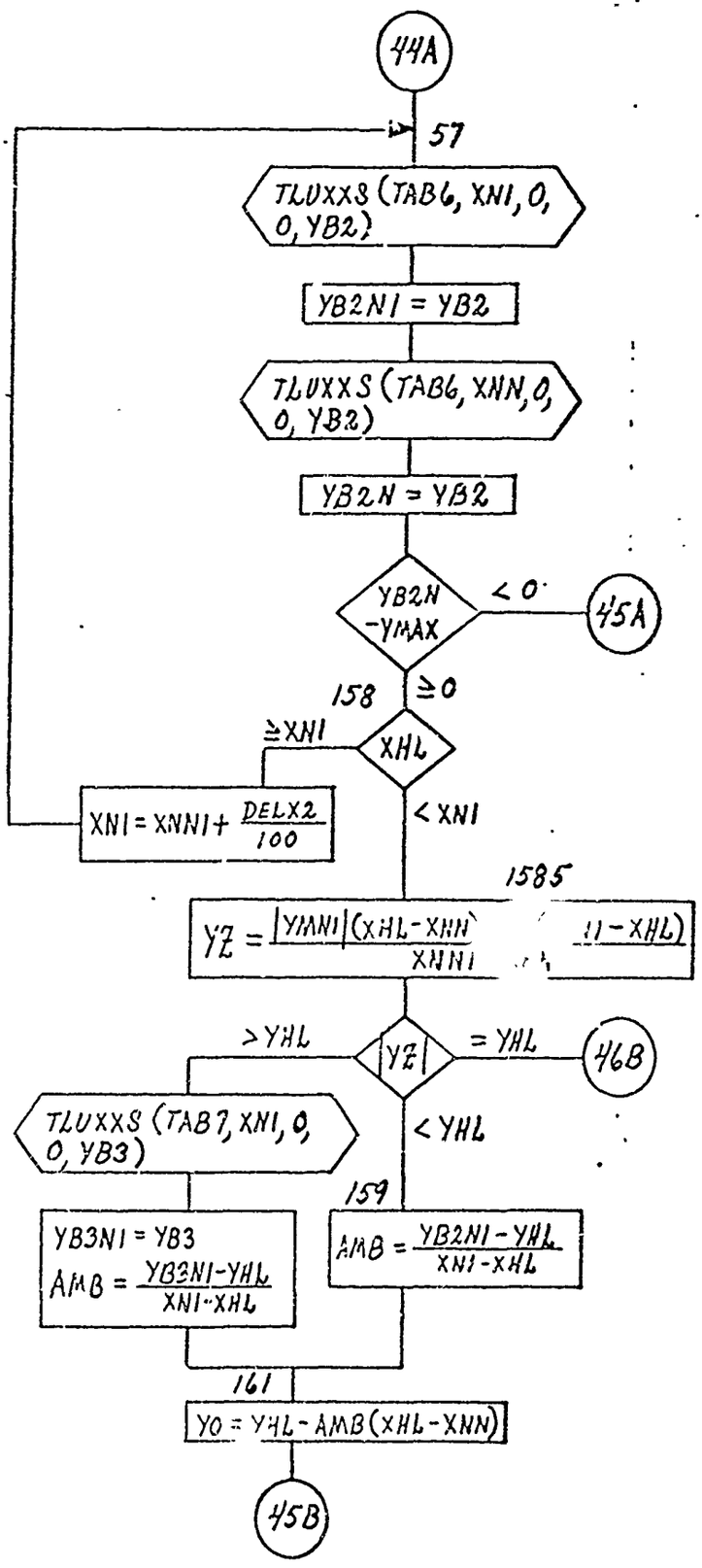

39A
IBEV
=1
FACTØR = .3
APERAO = .71
=2
FACTØR = .2
APERAO = .52
=3
FACTØR = .1
APERAO = .31
=4
FACTØR = .05
APERAO = 2.22
=5
FACTØR = .05
APERAO = 1.74
=6
FACTØR = .1
APERAO = 1.37
=7
FACTØR = .2
APERAO = 0.0
5055
R = (APERAO)(LAN3MR)
BEV = IBEV
WRITE BEV, FACTØR, APERAO
IBEV = IBEV+1
39B
5056
TLUXXS (TAB10, AV8, 0, 0, WATRAT)
40A

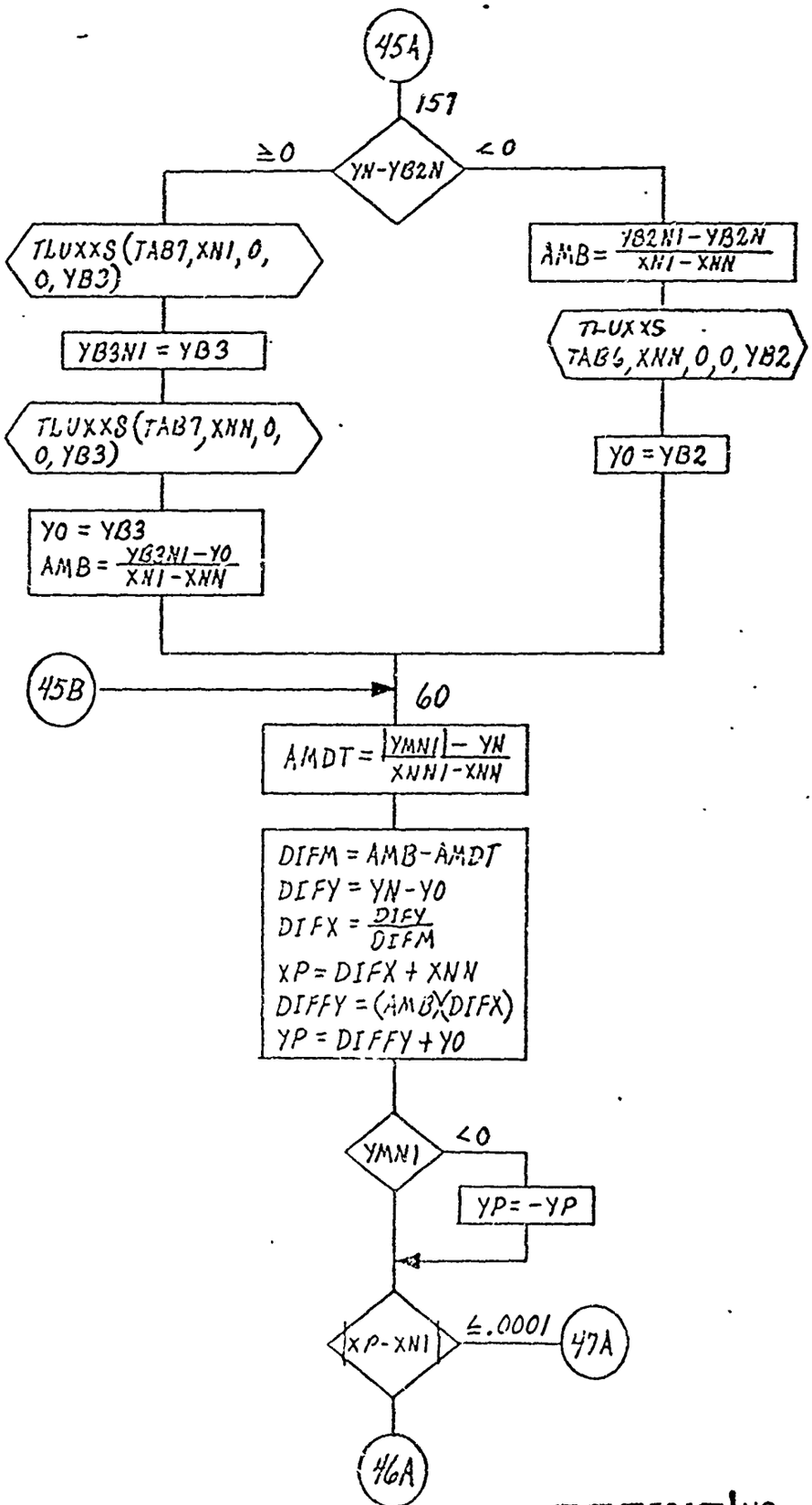

40A
WRITE WATRAT
AVS
>AVSMAX
WATRAT = 0
WAWRT = (FACTOR)(WATRAT)
I = 1
TABII_I
= AVS
≠ AVS
I = I + 2
≤ 800
I
≥ 800
CUMWAT = TABII_I+1
WRITE CUMWAT, WAWRT, AVS
5073
CUMWAT = CUMWAT + WAWRT
TABII_I+1 = CUMWAT
AVS = AVS + 0.25
WATRAT
>0
39B
≥ 0
41A

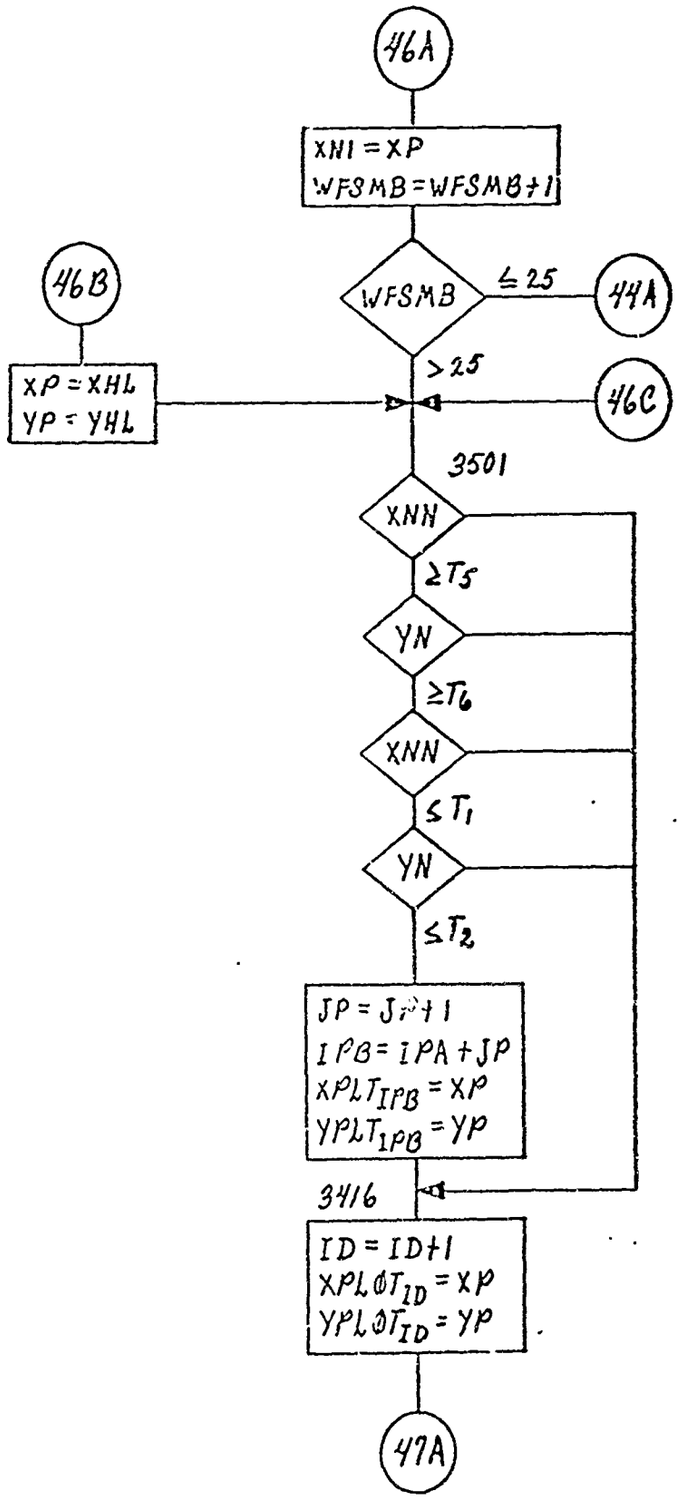

41A
IEND = 2(NK) + 5
I = 6
TAB10$_I$ = 0
I = I + 1
≤ IEND
I
> IEND
NK = 0
R
≤ 0
LANGMR = 0
WRITE TAB11
1B

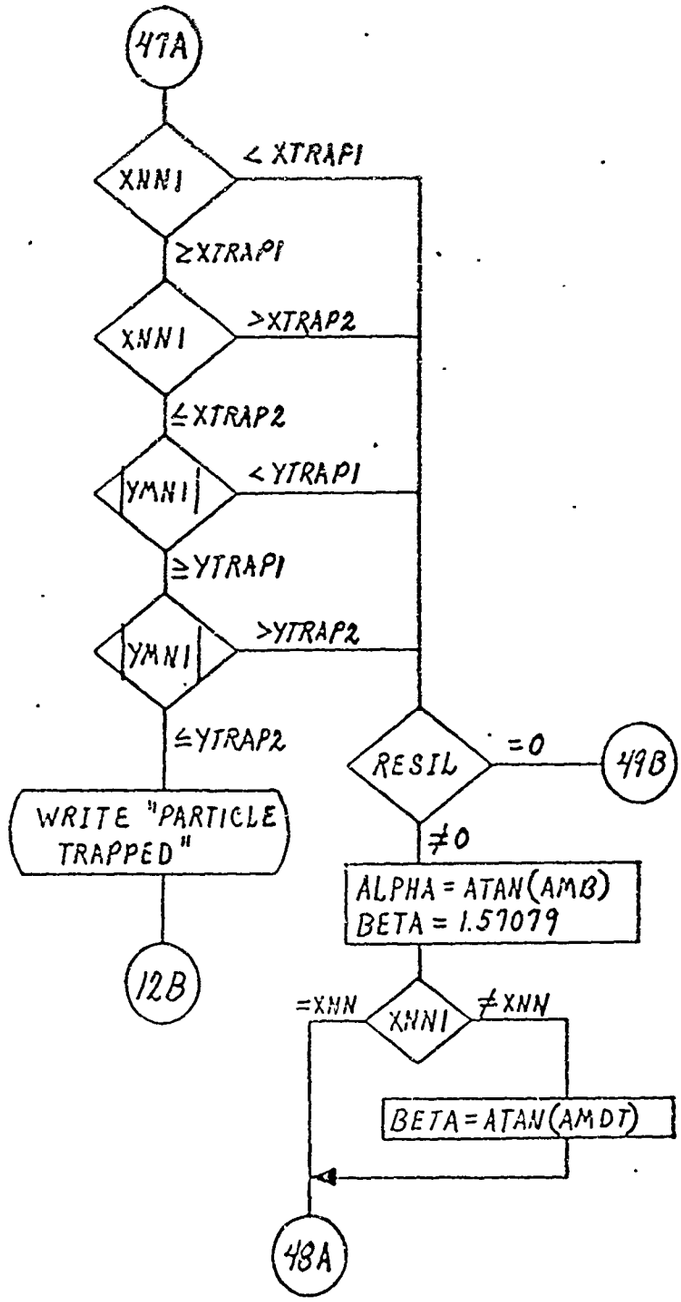

42A
54
|YMNI|
≥YMAX
38A
<YMAX
TLUXXS (TAB6,XNNI, 0,0,YB2)
WFSMB=0
XNI=XNNI
≥YB2
|YMNI|
<YB2
TLUXXS (TAB7,XNNI,0, 0,YB3)
TLUXXS (TAB8,XNNI,0, 0,YBI)
|YMNI|
≥YB3
11A
<YB3
44A
|YMNI|
≥YBI
11A
42B
1577
TLUXXS (TAB8,XNI,0, 0,YBI)
YBINI=YBI
TLUXXS (TAB8,XNN,0, 0,YBI).
YBIN=YBI
43A

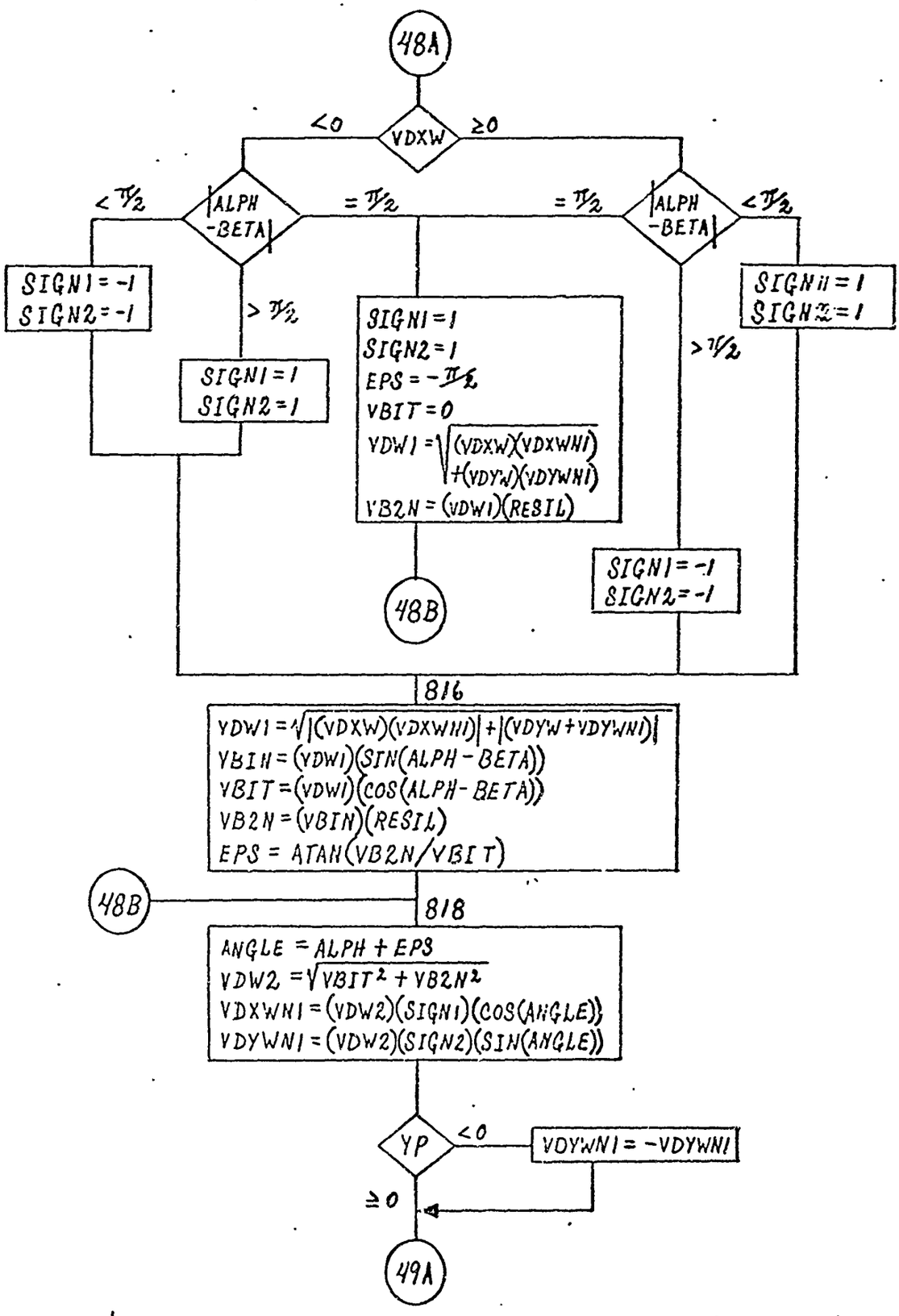

43A
XNN
< XHLI
≥ XHLI
YZI = [YM(XNNI − XHLI) + YMNI(XHLI − XNN)] / (XNNI − XNN)
AMB = (YBINI) / (XNI − XHLI)
AMDT = (YMNI − YM) / (XNNI − XNN)
472
AMB = (YBINI − YBIN) / (XNI − XNN)
AMDT = (YMNI − YN) / (XNNI − XNN)
DIFX = (YN − YBIN) / (AMB − AMDT)
XP = XNN + DIFX
YP = YBIN + (AMB)(DIFX)
YMNI
≥ 0
< 0
YP = −YP
YZI
> 0
≤ 0
DIFX = (−YM − AMDT(XHLI − XNN)) / (AMB − AMDT)
XP = XHLI + DIFX
YP = (−AMB)(DIFX)
AMDT = − AMDT
473
DIFX = [YM + AMDT(XHLI − XNN)] / (AMB − AMDT)
XP = XHLI + DIFX
YP = (AMB)(DIFX)
|XP − XNI|
≤ .0001
46C
> .0001
XNI = XP
WFSMB = WFSMB + 1
WFSMB
≤ 25
42B
> 25
46C

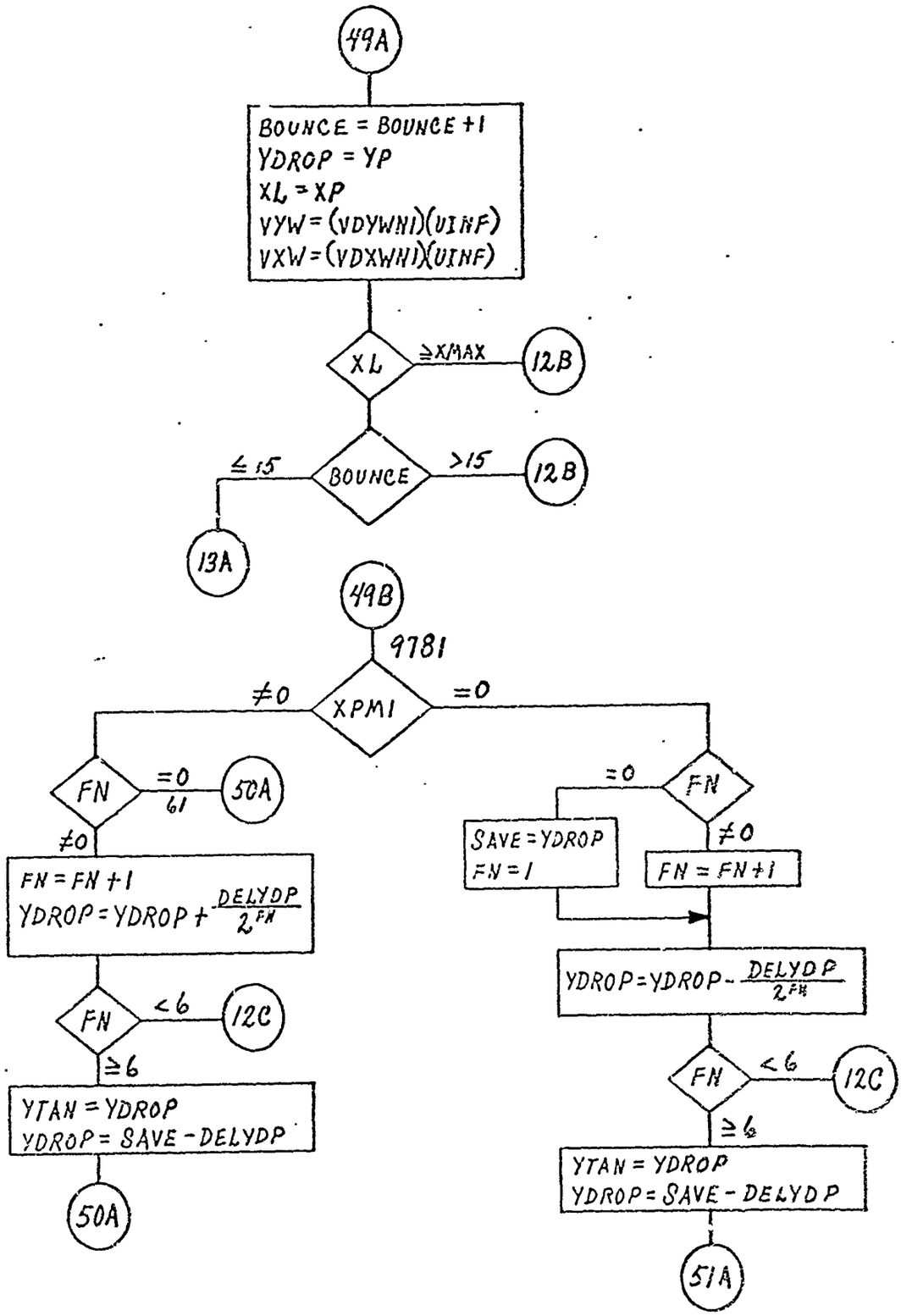

44A
57
TLUXXS (TAB6, XNI, 0, 0, YB2)
YB2NI = YB2
TLUXXS (TAB6, XNN, 0, 0, YB2)
YB2N = YB2
YB2N -YMAX
< 0
45A
≥ 0
158
XHL
≥ XNI
XNI = XNNI + DELX2/100
< XNI
1585
YZ = |YMNI| (XHL-XNN) / XNNI (1-XHL)
|YZ|
> YHL
= YHL
46B
< YHL
TLUXXS (TAB7, XNI, 0, 0, YB3)
YB3NI = YB3
AMB = (YB3NI-YHL)/(XNI-XHL)
159
AMB = (YB2NI-YHL)/(XNI-XHL)
161
YO = YHL - AMB (XHL-XNN)
45B

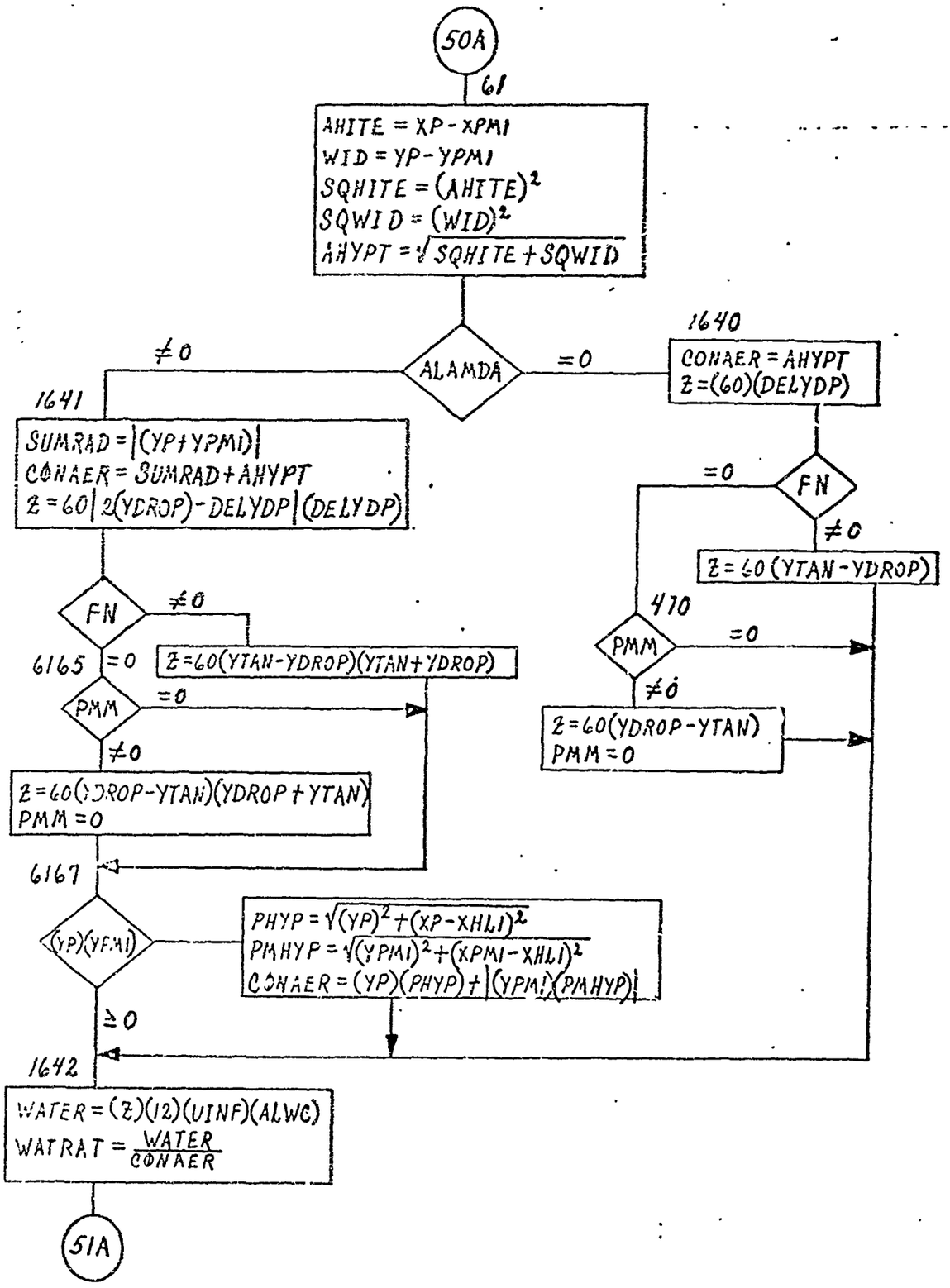

45A
157
YN-YB2N
≥0
<0
TLUXXS(TAB7,XN1,0,0,YB3)
YB3N1 = YB3
TLUXXS(TAB7,XNN,0,0,YB3)
YO = YB3
AMB = (YB3N1-YO)/(XN1-XNN)
AMB = (YB2N1-YB2N)/(XN1-XNN)
TLUXXS TAB6,XNN,0,0,YB2
YO = YB2
45B
60
AMDT = (|YMN1| - YN)/(XNN1-XNN)
DIFM = AMB-AMDT
DIFY = YN-YO
DIFX = DIFY/DIFM
XP = DIFX + XNN
DIFFY = (AMB)(DIFX)
YP = DIFFY + YO
YMN1
<0
YP = -YP
|XP-XN1|
≤.0001
47A
46A

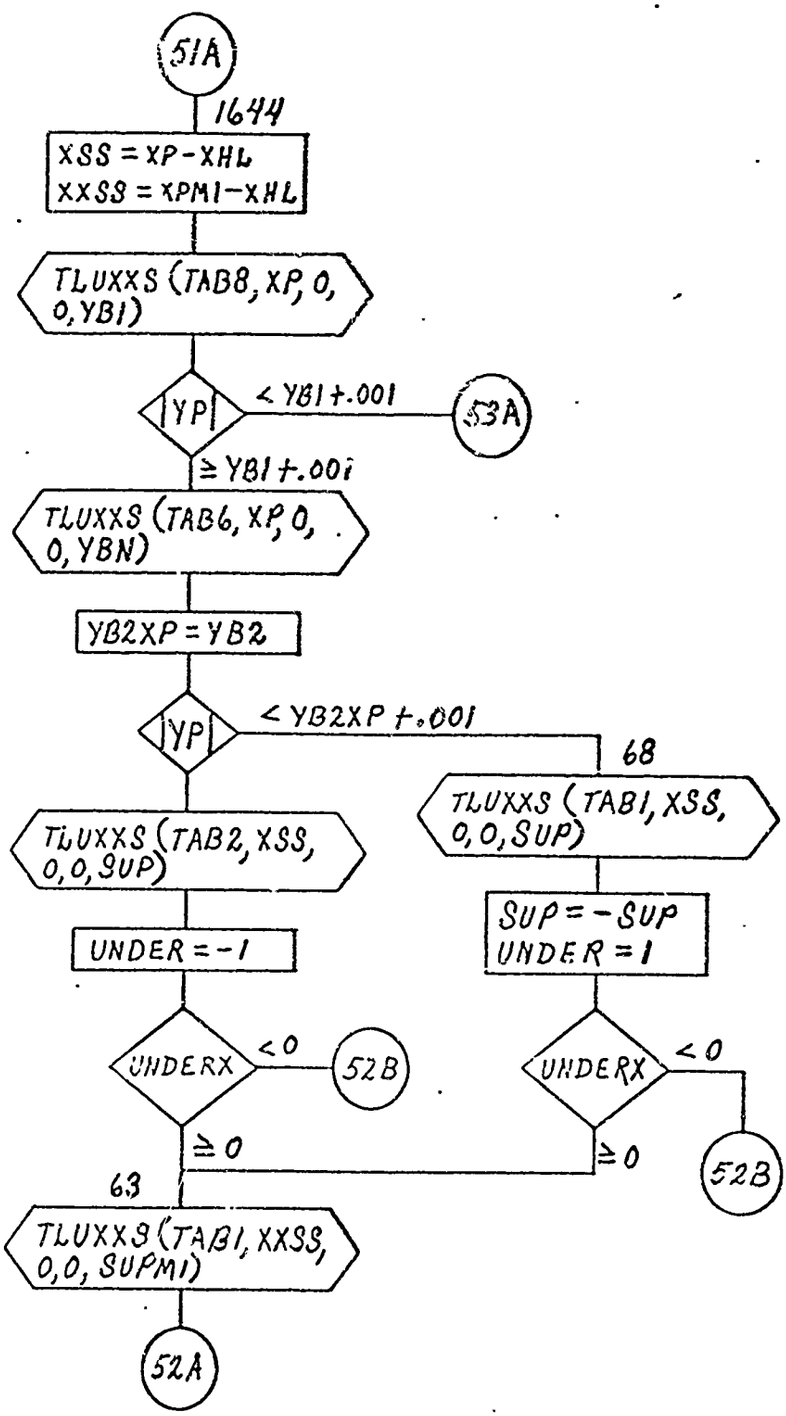

46A
XNI = XP
WFSMB = WFSMB + 1
WFSMB
≤ 25
44A
> 25
46C
46B
XP = XHL
YP = YHL
3501
XNN
≥ T5
YN
≥ T6
XNN
≤ T1
YN
≤ T2
JP = JP + 1
IPB = IPA + JP
XPLT IPB = XP
YPLT IPB = YP
3416
ID = ID + 1
XPLØT ID = XP
YPLØT ID = YP
47A

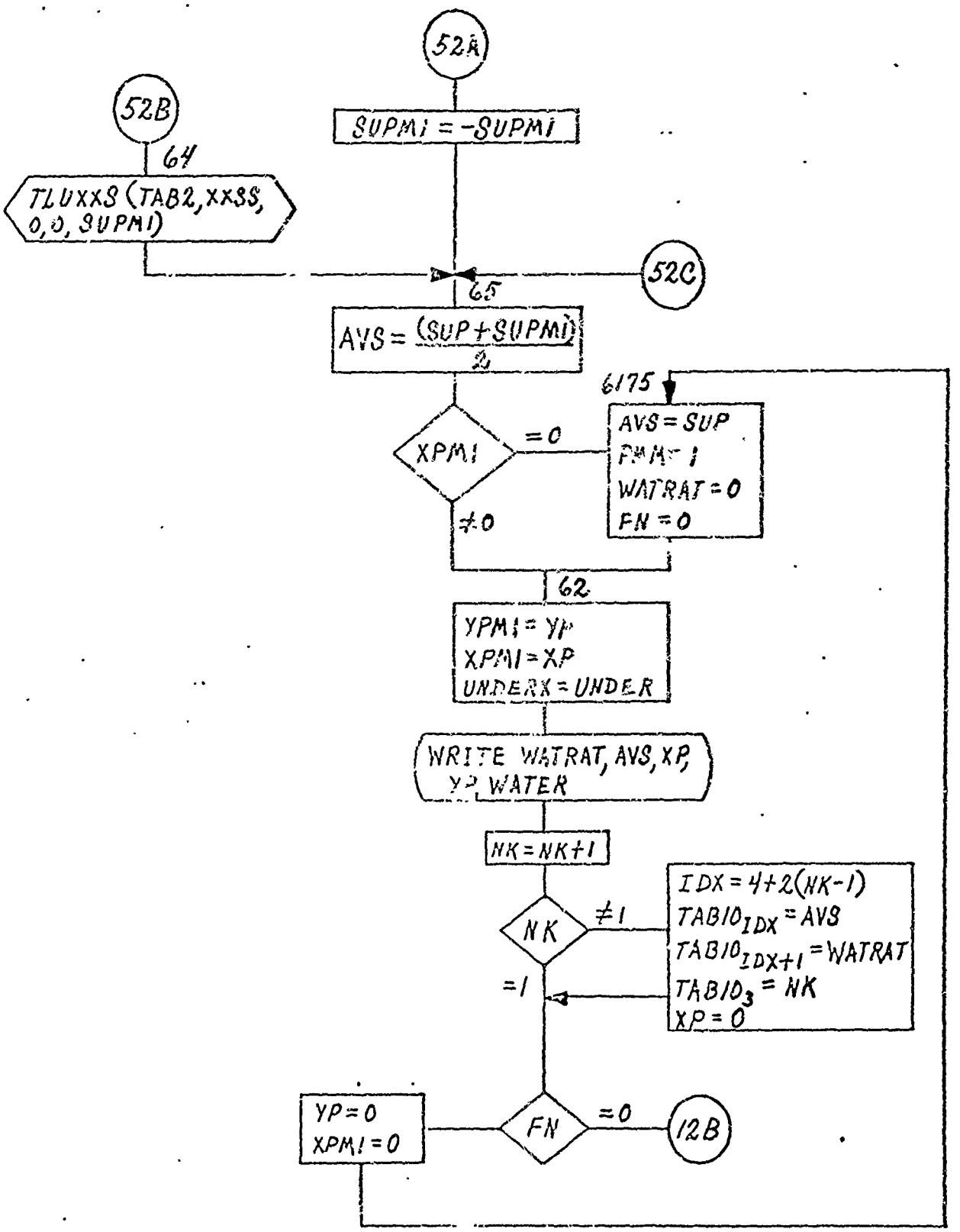

47A
XNN1
< XTRAP1
≥XTRAP1
XNN1
>XTRAP2
≤XTRAP2
|YMN1|
< YTRAP1
≥YTRAP1
|YMN1|
>YTRAP2
≤YTRAP2
WRITE "PARTICLE TRAPPED"
12B
RESIL
=0
49B
≠0
ALPHA = ATAN(AMB)
BETA = 1.57079
=XNN
XNN1
≠XNN
BETA=ATAN(AMDT)
48A

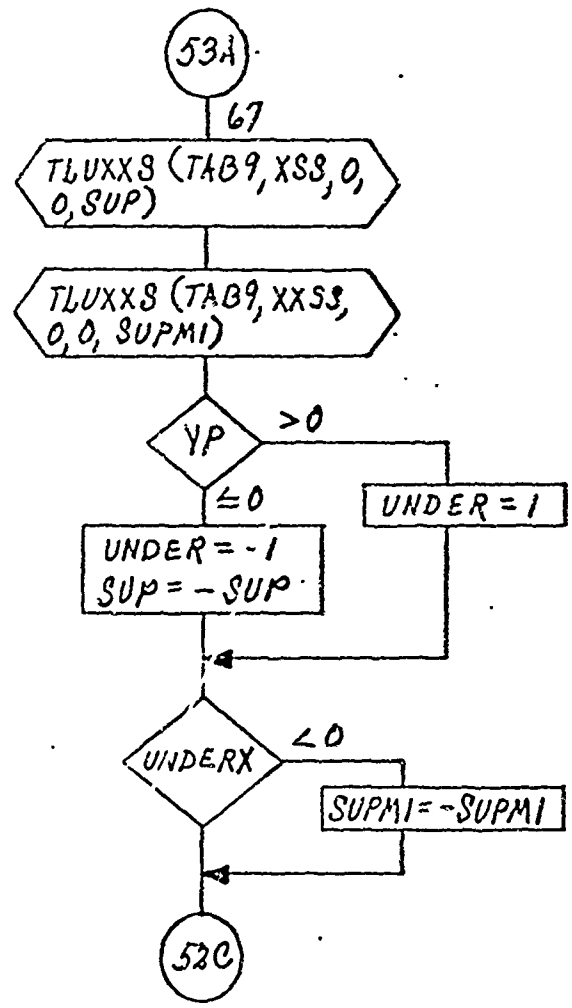

48A
VDXW
<0
≥0
|ALPH -BETA|
<π/2
=π/2
=π/2
|ALPH -BETA|
<π/2
SIGN1 = -1
SIGN2 = -1
>π/2
SIGN1 = 1
SIGN2 = 1
SIGN1 = 1
SIGN2 = 1
EPS = -π/2
VBIT = 0
VDW1 = √((VDXW)(VDXWN1) + (VDYW)(VDYWN1))
VB2N = (VDW1)(RESIL)
>π/2
SIGN1 = 1
SIGN2 = 1
SIGN1 = -1
SIGN2 = -1
48B
816
VDW1 = √|(VDXW)(VDXWN1)| + |(VDYW + VDYWN1)|
VBIN = (VDW1)(SIN(ALPH - BETA))
VBIT = (VDW1)(COS(ALPH - BETA))
VB2N = (VBIN)(RESIL)
EPS = ATAN(VB2N/VBIT)
48B
818
ANGLE = ALPH + EPS
VDW2 = √(VBIT² + VB2N²)
VDXWN1 = (VDW2)(SIGN1)(COS(ANGLE))
VDYWN1 = (VDW2)(SIGN2)(SIN(ANGLE))
YP
<0
VDYWN1 = -VDYWN1
≥0
49A


49A
BOUNCE = BOUNCE + 1
YDROP = YP
XL = XP
VYW = (VDYWNI)(UINF)
VXW = (VDXWNI)(UINF)
XL
≥XMAX
12B
≤15
BOUNCE
>15
12B
13A
49B
9781
≠0
XPMI
=0
FN
=0
61
50A
≠0
FN = FN + 1
YDROP = YDROP + DELYDP / 2^FN
FN
<6
12C
≥6
YTAN = YDROP
YDROP = SAVE - DELYDP
50A
=0
FN
SAVE = YDROP
FN = 1
≠0
FN = FN + 1
YDROP = YDROP - DELYDP / 2^FN
FN
<6
12C
≥6
YTAN = YDROP
YDROP = SAVE - DELYDP
51A


50A
61
AHITE = XP - XPMI
WID = YP - YPMI
SQHITE = (AHITE)²
SQWID = (WID)²
AHYPT = √SQHITE + SQWID
ALAMDA
≠0
=0
1640
CONAER = AHYPT
Z = (60)(DELYDP)
1641
SUMRAD = |(YP + YPMI)|
CONAER = SUMRAD + AHYPT
Z = 60 |2(YDROP) - DELYDP| (DELYDP)
FN
=0
≠0
Z = 60 (YTAN - YDROP)
FN
≠0
Z = 60(YTAN - YDROP)(YTAN + YDROP)
6165
=0
PMM
=0
≠0
Z = 60(YDROP - YTAN)(YDROP + YTAN)
PMM = 0
470
PMM
=0
≠0
Z = 60(YDROP - YTAN)
PMM = 0
6167
(YP)(YPMI)
PHYP = √(YP)² + (XP - XHLI)²
PMHYP = √(YPMI)² + (XPMI - XHLI)²
CONAER = (YP)(PHYP) + |(YPMI)(PMHYP)|
≥0
1642
WATER = (Z)(12)(UINF)(ALWC)
WATRAT = WATER / CONAER
51A


51A
1644
XSS = XP - XHL
XXSS = XPM1 - XHL
TLUXXS (TAB8, XP, 0, 0, YB1)
|YP|
< YB1 + .001
53A
≥ YB1 + .001
TLUXXS (TAB6, XP, 0, 0, YBN)
YB2XP = YB2
|YP|
< YB2XP + .001
68
TLUXXS (TAB1, XSS, 0, 0, SUP)
TLUXXS (TAB2, XSS, 0, 0, SUP)
SUP = - SUP
UNDER = 1
UNDER = -1
UNDERX
< 0
52B
UNDERX
< 0
≥ 0
≥ 0
52B
63
TLUXXS (TAB1, XXSS, 0, 0, SUPM1)
52A


52A
SUPMI = -SUPMI
52B
64
TLUXXS (TAB2, XXSS, 0, 0, SUPMI)
52C
65
AVS = (SUP+SUPMI)/2
6175
XPMI
=0
AVS = SUP
PMM = 1
WATRAT = 0
FN = 0
≠0
62
YPMI = YP
XPMI = XP
UNDERX = UNDER
WRITE WATRAT, AVS, XP, YP, WATER
NK = NK+1
NK
≠1
IDX = 4+2(NK-1)
TAB10IDX = AVS
TAB10IDX+1 = WATRAT
TAB10₃ = NK
XP = 0
=1
YP = 0
XPMI = 0
FN
=0
12B


53A
67
TLUXXS (TAB9, XSS, 0, 0, SUP)
TLUXXS (TAB9, XXSS, 0, 0, SUPMI)
YP
>0
≤0
UNDER = 1
UNDER = -1
SUP = -SUP
UNDERX
<0
SUPMI = -SUPMI
52C

10.4 Appendix 4 - Water Droplet Trajectory Computer Program Listing

```
$IBFTC DRCP
      DIMENSION  T(7),XPLOT(550),YPLOT(550),XPLT(3100),YPLT(3100)
      DIMENSION  TAB1(100),TAB2(100),TAB3( 5C00),TAB4(200),TAB5(200)
      DIMENSION TAB6(200),TAB7(200)
      DIMENSION HEAD(12),TA(7),IPC(100)
      DIMENSION TL(9)
      EXTERNAL  GOTHIC
      COMMON  TAB3
C     DIMENSION TAB6(200),TAB7(200)
C     DIMENSION  T(7),XPLOT(200),YPLOT(200),XPLT(2000),YPLT(2000)
C     DIMENSION HEAD(12),TA(7),IPC(100)
      DIMENSION TAB8(100),TAB9(100)
      INTEGER HEAD
      DIMENSION TAB10(500),TAB11(800)
C     DIMENSION TL(9)
      REAL MB,MDT,LANGMR
C     COMMON  TAB3
C     GOTHIC
 10CC FORMAT(6E12.6)
 8787 FORMAT(4X,4HDTW=E12.4,4X,5HVDYW=E12.4,4X,7HVDYWN1=E12.4/4X,5HVDXW=
     1E12.4,4X,7HVDXWN1=E12.4,4X,3HAK=E12.4/4X,3HCK=E12.4,4X,4HREN=E12.4
     2,4X,6HREINF=E12.4/4X,4HVDX=E12.4,4X,4HVDY=E12.4,4X,3HVX=E12.4/4X,
     33HVY=E12.4)
 1240 FORMAT(4X,7HWATRAT=E20.8,7X,4HAVS=E20.8/4X,3HXP=E20.8,11X,3HYP=E20
     1.8,9X,5HWATER ,F14.6)
      DIMENSION Y(150),X(150)
  6C0 FORMAT (6E18.5)
  601 FORMAT (1X)
  603 FORMAT (1X,4F14.5)
  604 FORMAT(/4H XP=F12.6/5H XN1=F11.6/)
  610 FORMAT (9H AHLSTEDT)
  630 FORMAT (1X,2HR=F12.6//6X,3HAVS,11X,6HWATRAT//(F14.6,2X,F14.6))
  635 FORMAT (///6X,3HAVS,11X,6HCUMWAT//(F14.6,2X,F14.6))
  640 FORMAT (1X,6F14.6)
  650 FORMAT (1X,8F9.3)
  660 FORMAT (17H PARTICLE TRAPPED)
  661 FORMAT (38H TOO MANY POINTS IN PLOT OF WHOLE AREA)
  662 FORMAT (40H TOO MANY POINTS IN RESTRICTED AREA PLOT)
  670 FORMAT (8H1TABLE 1/33H X=DISTANCE TO RIGHT OF HIGHLIGHT/45H Y=LOWE
     XR COWL SURFACE DISTANCE FROM HIGHLIGHT//10X,1HX,20X,1HY/)
  671 FORMAT (8H TABLE 2/33H X=DISTANCE TO RIGHT OF HIGHLIGHT/46H Y=UPPE
     XR COWL SURFACE DISTANCE FROM HIGHLIGHT//10X,1HX,20X,1HY/)
  605 FORMAT(1X////)
  673 FORMAT (8H TABLE 4/34H X=Y VALUE OF POTENTIAL FLOW FIELD/53H Y=FIR
     XST BOUNDARY VALUE OF X AFTER LEFT HAND BOUNDARY//10X,1HX,20X,1HY/)
  674 FORMAT (8H TABLE 5/34H X=Y VALUE OF POTENTIAL FLOW FIELD/54H Y=SEC
     XOND BOUNDARY VALUE OF X AFTER LEFT HAND BOUNDARY//10X,1HX,20X,1HY)
  675 FORMAT (8H TABLE 6/34H X=X VALUE OF POTENTIAL FLOW FIELD/50H Y=FIR
     XST BOUNDARY VALUE OF Y ABOVE LOWER BOUNDARY//10X1HX,20X,1HY/)
  676 FORMAT (8H TABLE 7/34H X=X VALUE OF POTENTIAL FLOW FIELD/50H Y=SEC
     XOND BOUNDARY VALUE OF Y ABOVE LOWER BOUNDARY//10X1HX,20X1HY/)
  677 FORMAT (8H TABLE 8/34H X=X VALUE OF POTENTIAL FLOW FIELD/41H CENTE
```

```
    XRLINE OR LOWEST BOUNDARY VALUE OF Y//10X,1HX,20X,1HY)
 678 FORMAT (8H TABLE 9/39H X=DISTANCE TO RIGHT OF HIGHLIGHT (XHL)/63H
    1Y=CENTERBODY SURFACE DISTANCE FROM CENTERBODY HIGHLIGHT (XHL1)//10
    2X,1HX,20X,1HY)
 680 FORMAT (10X,F10.5,10X,F10.5)
3100 FORMAT(12A6)
3101 FORMAT(1H1,12A6)
3102 FORMAT(1H0,10X4HALWC17X1HR14X4HUINF13X5HUPINF16X2HPA16X2HWA)
3104 FORMAT (1H0,11X,3HYSL,15X,3HYSM,13X,5HVXWIN,13X,5HVYWIN,13X,
    16HDIRECT,13X,6HRANDOM)
3105 FORMAT (1H0,11X,3HXHL,15X,3HYHL,12X,6HALAMDA,13X,5HRESIL,14X,4HRHO
    XW,12X,6HLANGMR)
3165 FORMAT (1H0,9X,5HREINF,16X,2HAK)
3106 FORMAT (1X,6(E15.7,3X))
3107 FORMAT(1H0)
3108 FORMAT (1H0,10X,4HXMAX,14X,4HYMAX,13X,5HYDROP,12X,6HDELYDP,16X,
    12HXL,14X,4HXHL1)
3109 FORMAT (2X,4HXNN1,5X,4HYMN1,6X,3HDTW,6X,2HCK,7X,3HREN,9X,3HVDX,
    17X,6HVDXWN1,6X,6HVDYWN1,8X,3HVDY,6X,3HXKT,3X,6HDIRECT)
3110 FORMAT (1H ,F7.3,F10.4,1X,F8.5,1X,E8.1,1X,E11.3,1X,E11.3,1X,E11.3,
    11X,E11.3,1X,E11.3,1X,3F5.1)
3111 FORMAT(8HOVDYWM =E12.4,5X7HVDXWM =E12.4)
3112 FORMAT (6E12.6)
3113 FORMAT (5E14.6)
3114 FORMAT (1H0,8X,6HGRAVTY,13X,5HDELX1,13X,5HDELX2,13X,5HXREF1,13X,
    15HXREF2)
3115 FORMAT (1H0,8X,6HYTRAP1,12X,6HYTRAP2,12X,6HXTRAP1,12X,6HXTRAP2,12X
    X,5HTABLE)
     DATA IBLANK/6H      /
     LANGMR=0.
     REWIND 23
     WRITE (23,610)
     ID=0
     DO 2009 I=1,800,2
     FILL=FLOAT(I-1)/8.-30.
2009 TABL1(I)=FILL
     BOUNCE=0.0
2001 CALL RENTRY (N)
     IF (N.NE.0) GO TO 6041
2000 CONTINUE
     CALL REMOVE (23,3)
     TL(2) = 0.0
     TL(6) = 10.0
     TL(9) = -1.0
     GO TO 2004
6041 X55=X55
2003 TA(7) = ID
     IF(ID - 2)3394,3394,3393
3393 CALL MPLOTS(TA,XPLOT,YPLOT)
3394 TA(7) = 0.0
     TL(1) = TA(1)
     TL(3)= .3/TA(3)
```

```
      TL(4) = .7/TA(4)
      TL(5) = .5/TA(3)
      TL(6)= TA(6)-.5/TA(4)
      TL(7) = TA(5) + 2.25/TA(3)
      CALL ALPHA(TL,HEAD,72,GOTHIC)
      CALL MPLOTS(TA,XPLOT,YPLOT)
      T(1) = TA(1)
      T(2) = TA(2)
      IF(JP)3396,3396,3395
 3395 IP = IP + 1
      IPC(IP) = JP
 3396 IF(IP)2002,2002,3397
 3397 IPA = 1
      DO 3398 I=1,IP
      J = IPC(I)
      T(7) = J
      CALL MPLOTS(T,XPLT(IPA),YPLT(IPA))
 3398 IPA = IPA + J
      TL(3) = .3/T(3)
      TL(4) =  .7/T(4)
      TL(5) =  .5/T(3)
      TL(6) = T(6) - .5/T(4)
      TL(7) =  T(5) + 2.25/T(3)
      CALL ALPHA(TL,HEAD,72,GOTHIC)
      T(7) = 0
      CALL MPLOTS(T,XPLT,YPLT)
      IF (LANGMR.GT.0.) GO TO 2005
      GO TO 2002
 2004 READ (5,3112) (TAB1(J),J=1,3)
      IA=TAB1(1)*TAB1(2)*TAB1(3)+3.0
      READ (5,3112) (TAB1(J),J=4,IA)
      READ (5,3112)(TAB2(J),J=1,3)
      IA=TAB2(1)*TAB2(2)*TAB2(3)+3.0
      READ (5,3112) (TAB2(J),J=4,IA)
      READ (5,3113)(TAB3(J),J=1,3)
      IA=TAB3(1)*TAB3(2)*TAB3(3)+3.0
      READ (5,3113) (TAB3(J),J=4,IA)
      READ (5,3112)(TAB4(J),J=1,3)
      IA=TAB4(1)*TAB4(2)*TAB4(3)+3.0
      READ (5,3112) (TAB4(J),J=4,IA)
      READ (5,3112)(TAB5(J),J=1,3)
      IA=TAB5(1)*TAB5(2)*TAB5(3)+3.0
      READ (5,3112) (TAB5(J),J=4,IA)
      READ (5,3112)(TAB6(J),J=1,3)
      IA=TAB6(1)*TAB6(2)*TAB6(3)+3.0
      READ (5,3112) (TAB6(J),J=4,IA)
      READ (5,3112)(TAB7(J),J=1,3)
      IA=TAB7(1)*TAB7(2)*TAB7(3)+3.0
      READ (5,3112)(TAB7(J),J=4,IA)
      READ (5,3112)(TAB8(J),J=1,3)
      IA=TAB8(1)*TAB8(2)*TAB8(3)+3.0
      READ (5,3112)(TAB8(J),J=4,IA)
```

```
      READ (5,3112)(TAB9(J),J=1,3)
      IA=TAB9(1)*TAB9(2)*TAB9(3)+3.0
      READ (5,3112)(TAB9(J),J=4,IA)
 2002 IF (R.NE.LANGMR.AND.LANGMR.NE.0.) GO TO 2005
      READ (5,3100) HEAD
      IF (HEAD(1).EQ.IBLANK) GO TO 2004
      READ (5,1000)(TA(I),I=1,6)
      READ (5,1000)(T(I),I=1,6)
      TA(1)=-23.
      READ (5,1000) ALWC,R,UINF,UPINF,PA,WA
      READ (5,1000) XMAX,YMAX,YDROP,DELYDP,XL,XHL1
      READ (5,1000) YSL,YSM,VXWIN,VYWIN,DIRECT,RANDCM
      READ (5,1000) XHL,YHL,ALAMDA,RESIL,RHOW,LANGMR
      READ (5,1000) GRAVTY,DELX1,DELX2,XREF1,XREF2
      READ (5,1000) YTRAP1,YTRAP2,XTRAP1,XTRAP2,TABLE
      SYDROP=YDROP
      IBEV=1
      NK=0
      KL=0
      IF (LANGMR.NE.0.) R=LANGMR
      DO 2007 I=2,800,2
 2007 TAB11(I)=0.
      IMAX=0
 2005 REINF=2.0*R*PA*UINF/WA
      IF (TABLE .EQ.0.0) GO TO 6055
      TAB1(1)=1.
      TAB1(2)=2.
      TAB1(3)=100.
      TAB1(4)=0.
      TAB1(5)=0.
      NK=5
      DO 701 I=6,100,2
  701 IF (TAB6(I).GE.XHL) GO TO 702
  702 NK6=I-1
  703 NK=NK+1
      NK6=NK6+1
      TAB1(NK)=TAB6(NK6)-XHL
      NK=NK+1
      NK6=NK6+1
      IF (TAB1(NK-1).EQ.0.) GO TO 705
      TAB1(NK)=TAB1(NK-2)+SQRT((TAB6(NK6-1)-TAB6(NK6-3))**2+(TAB6(NK6)-
     XTAB6(NK6-2))**2)
      IF (TAB6(NK6-1).LT.XMAX-.001) GO TO 703
      GO TO 706
  705 TAB1(NK)=0.
      GO TO 703
  706 TAB2(1)=1.
      NK=(NK-3)/2
      TAB1(3)=NK
      TAB2(2)=2.
      TAB2(3)=100.
      TAB2(4)=0.
```

```
      TAB2(5)=0.
      NK=5
      DO 710 I=6,100,2
  710 IF (TAB7(I).GE.XHL) GO TO 712
  712 NK7=I-1
  713 NK=NK+1
      NK7=NK7+1
      TAB2(NK)=TAB7(NK7)-XHL
      NK=NK+1
      NK7=NK7+1
      IF (TAB2(NK-1).EQ.0.) GO TO 715
      TAB2(NK)=TAB2(NK-2)+SQRT((TAB7(NK7-1)-TAB7(NK7-3))**2+(TAB7(NK7)-
     XTAB7(NK7-2))**2)
      IF (TAB7(NK7-1).LT.XMAX-.0001) GO TO 713
      GO TO 716
  715 TAB2(NK)=0.
      GO TO 713
  716 TAB9(1)=1.
      TAB9(2)=2.
      TAB9(3)=50
      NK=(NK-3)/2
      TAB2(3)=NK
      TAB9(4)=0.
      TAB9(5)=0.
      NK=5
      DO 720 I=7,50,2
  720 IF (TAB8(I).GT.0.) GO TO 722
  722 NK8=I-4
      XHLCB=TAB8(I-3)
  723 NK=NK+1
      NK8=NK8+1
      TAB9(NK)=TAB8(NK8)-XHL
      NK=NK+1
      NK8=NK8+1
      IF (NK.EQ.7) GO TO 725
      TAB9(NK)=TAB9(NK-2)+SQRT((TAB8(NK8-1)-TAB8(NK8-3))**2+(TAB8(NK8)-
     XTAB8(NK8-2))**2)
      IF (TAB8(NK8-1).LT.XMAX-.0001) GO TO 723
      GO TO 726
  725 TAB9(NK)=0.
      GO TO 723
  726 NK=(NK-3)/2
      TAB9(3)=NK
 6055 IA=TAB6(1)*TAB6(2)*TAB6(3)
      WRITE (6,670)
      IA=TAB1(1)*TAB1(2)*TAB1(3)+3.0
      WRITE (6,680) (TAB1(I),I=6,IA)
      WRITE(6,605)
      WRITE (6,671)
      IA=TAB2(1)*TAB2(2)*TAB2(3)+3.0
      WRITE(6,680) (TAB2(I),I=6,IA)
      WRITE(6,605)
```

```
      WRITE(6,673)
      IA=TAB4(1)*TAB4(2)*TAB4(3)+3.0
      WRITE (6,680) (TAB4(I),I=6,IA)
      WRITE (6,605)
      WRITE (6,674)
      IA=TAB5(1)*TAB5(2)*TAB5(3)+3.0
      WRITE (6,680) (TAB5(I),I=6,IA)
      WRITE (6,605)
      WRITE (6,675)
      IA=TAB6(1)*TAB6(2)*TAB6(3)+3.0
      WRITE (6,680) (TAB6(I),I=6,IA)
      WRITE (6,605)
      WRITE (6,676)
      IA=TAB7(1)*TAB7(2)*TAB7(3)+3.0
      WRITE (6,680) (TAB7(I),I=6,IA)
      WRITE (6,605)
      WRITE (6,677)
      IA=TAB8(1)*TAB8(2)*TAB8(3)+3.0
      WRITE (6,680) (TAB8(I),I=6,IA)
      WRITE (6,605)
      WRITE (6,678)
      IA=TAB9(1)*TAB9(2)*TAB9(3)*3.0
      WRITE (6,680) (TAB9(I),I=6,IA)
      WRITE (6,605)
      TABLE=0.0
      YDROP=SYDROP
      AK=2.0*RHOW*R**2*UINF/(9.0*WA)
      WRITE (6,3101)HEAD
      WRITE (6,3102)
      WRITE (6,3106)ALWC,R,UINF,UPINF,PA,WA
      WRITE (6,3108)
      WRITE (6,3106) XMAX,YMAX,YDROP,DELYDP,XL,XHL1
      WRITE (6,3104)
      WRITE (6,3106) YSL,YSM,VXWIN,VYWIN,DIRECT,RANDOM
      WRITE (6,3105)
      WRITE (6,3106) XHL,YHL,ALAMDA,RESIL,RHOW,LANGMR
      WRITE (6,3114)
      WRITE (6,3106) GRAVTY,DELX1,DELX2,XREF1,XREF2
      WRITE (6,3115)
      WRITE (6,3106) YTRAP1,YTRAP2,XTRAP1,XTRAP2,TABLE
      WRITE (6,3165)
      WRITE (6,3106) REINF,AK
      WRITE (6,3107)
      DO 2010 I=1,500
 2010 TAB10(I)=0.
      TAB10(1)=1.
      TAB10(2)=2.
      TAB10(3)=500.
      TAB10(4)=0.
      TAB10(5)=0.
      XPM1=0.
      XKT = 0.0
```

```
      CK = 0.0
      IP = 0
      IPA = 0
      JP = 0
      YP=0.0
      YPM1 = 0.0
      WATRAT = 0.0
      AVS = 0.0
      UNDER=1.0
      VXW=VXWIN
      VYW=VYWIN
      STRCIR=CIRECT
      FN=0.0
      PMM=0.0
      GO TO 104
  100 YM=YMN1
      YN=ABS(YM)
      XNN=XNN1
      HUNXNN=100.*XNN1
      IF(T(5) -XNN)3400,3400,3404
 3400 IF(T(6) - YM)3401,3401,3404
 3401 IF(T(1) -XNN)3404,3402,3402
 3402 IF(T(2) - YM)3404,3403,3403
 3403 JP = JP + 1
      IPB = IPA + JP

      IF (IPB.LT.3099) GO TO 6064
      WRITE(6,662)
      IPB=IPA
      JP=0
 6064 XPLT(IPB)= XNN
      YPLT(IPB)= YM
 3404 ID = ID + 1

      IF (ID.LT.549) GO TO 6065
      WRITE (6,661)
      ID=ID-1
 6065 XPLOT(ID) = XNN
      YPLOT(ID) = YM
      VDXW=VDXWN1
      VDYW=VDYWN1
      GO TO 2205
  102 IF (RANDOM) 305,305,310
  305 YDROP=YDROP+DELYDP
  306 IF (RANDOM.GT.0.0) GO TO 310
      VXW=VXWIN
      VYW=VYWIN
      GO TO 315
  310 READ (5,1000) YDROP,XL,VXW,VYW
      BOUNCE=0.0
      IF (YDROP+XL+VXW+VYW.EQ.0.0) GO TO 2003
      IF (YDROP.GT.YMAX) STOP
```

```
 315 IF(JP) 3405,3406,3405
3405 IP = IP + 1
     IPC(IP) = JP
     IPA = IPA + JP
     JP = 0
3406 TA(7) = ID
     DIRECT=STRDIR
     CALL MPLOTS(TA,XPLOT,YPLOT)
 104 VDXW=VXW/UINF
     VDYW=VYW/UINF
     WRITE (6,603) XL,YDROP,VDXW,VDYW
     YM=YDROP
     YN=ABS(YM)
     WRITE (6,3109)
     HUNXNN=100.*XL
     XNN=XL
     XNN1=XNN
     XPLOT(1) = XNN
     YPLOT(1) = YM
     IF(T(5) -XNN)3407,3407,3411
3407 IF(T(6) - YM)3408,3408,3411
3408 IF(T(1) -XNN)3411,3409,3409
3409 IF(T(2) - YM)3411,3410,3410
3410 JP = JP + 1
     IPB = IPA + JP
     XPLT(IPB) = XNN
     YPLT(IPB) = YM
3411 ID = 2
     XPLOT(ID) = XNN
     YPLOT(ID) = YM
     IF (XNN.GT.XL) GO TO 2205
 103 NYR=TAB3(2)-1.
     NXC=TAB3(3)-1.
     DO 210 I=1,NYR
     Y(I)=FLOAT(INT(TAB3(I+4)*10000.0+.5))/10000.0
     TAB3(I+4)=Y(I)
 210 CONTINUE
     J=4
     DO 220 I=1,NXC
     J=J+NYR+1
     X(I)=FLOAT(INT(TAB3(J)  *10000.0+.5))/10000.0
     TAB3(J)=X(I)
 220 CONTINUE
2205 IF (YN.GT.YMAX) GO TO 2003
     DELX=DELX1
     IF (XNN.GT.XREF1.AND.XNN.LT.XREF2) DELX=DELX2
 221 IF (DIRECT) 225,225,260
 225 IF (VDYW)226,226,235
 226 IF (YM.LE.0.) GO TO 231
 232 DO 228 I=1,NYR
     IF (Y(I).GE.YN) GO TO 229
 228 CONTINUE
```

```
229 YMN1=Y(I-1)
230 YJ=Y(I-1)
    IF(YM.LE.-Y(1)) YMN1=-YMN1
    IF (ABS(YMN1).EQ.Y(2).AND.VDYW.GT.0.0) YMN1=ABS(YMN1)
    YJ1=Y(I)
    GO TO 250
235 IF (YM.LT.0.) GO TO 232
231 DO 236 I=1,NYR
    IF (Y(I).GT.YN) GO TO 237
236 CONTINUE
237 YMN1=Y(I)
    GO TO 230
250 DO 255 I=1,NXC
    IF (X(I).GT.XNN) GO TO 256
255 CONTINUE
256 XN1=X(I)
    XN=X(I-1)
    IF (DIRECT.GT.0.) XNN1=(HUNXNN+DELX)/100.
    GO TO 13
260 DO 265 I=1,NYR
    IF (Y(I).GE.YN) GO TO 266
265 CONTINUE
266 IF (YN.EQ.Y(1)) I=2
    YJ1=Y(I)
    YJ=Y(I-1)
    IF (VDXW) 270,271,250
271 IF (VDX) 270,270,250
270 DO 275 I=1,NXC
    IF (X(I).GE.XNN) GO TO 276
275 CONTINUE
276 XN1=X(I)
    XN=X(I-1)
    XNN1=(HUNXNN-DELX)/100.
 13 CALL TLUXXS(TAB6,XN1,0,0,YB2)
    CALL TLUXXS(TAB7,XN1,0,0,YB3)
    YB2N1 =YB2
    YB3N1 =YB3
    IND1=-1
    IND2=-2
    IND3=-3
    IND4=-4
    IF (XNN1-XHL1) 11,12,12
 12 IF(YN-YSL ) 11,1301,1301
1301 IF(YN-YSM)  14, 14, 11
 14 CALL TLUXXS(TAB4, YJ,0,0,XB2)
    CALL TLUXXS(TAB5, YJ,0,0,XB3)
    XB2J=XB2
    XB3J=XB3
    CALL TLUXXS(TAB4,YJ1,0,0,XB2)
    CALL TLUXXS(TAB5,YJ1,0,0,XB3)
    XB2J1=XB2
    XB3J1=XB3
```

```
      CALL TLUXXS(TAB6,XN,C,0,YB2)
      CALL TLUXXS(TAB7,XN,C,0,YB3)
      YB2N=YB2
      YB3N=YB3
      CALL TLUXXS(TAB6,XN1,0,0,YB2)
      CALL TLUXXS(TAB7,XN1,0,0,YB3)
      CALL TLUXXS (TAB8,XN,0,0,YB1)
      YB1N=YB1
      CALL TLUXXS (TAB8,XN1,0,0,YB1)
      YB1N1=YB1
      YB2N1=YB2
      YB3N1=YB3
      IND1= -1.0
      IND2= -1.0
      IND3= -1.0
      IND4= -1.0
      IXB2J=10000.0*(XB2J+.00001)
      IXB3J=10000.0*(XB3J+.00001)
      IXB2J1=10000.0*(XB2J1+0.00001)
      IXB3J1=10000.0*(XB3J1+0.00001)
      IYB2N =10000.0*(YB2N +0.00001)
      IYB3N =10000.0*(YB3N +0.00001)
      IYB2N1=10000.0*(YB2N1+0.00001)
      IYB3N1=10000.0*(YB3N1+0.00001)
      IYB1N =10000.0*(YB1N +0.00001)
      IYB1N1=10000.0*(YB1N1+0.00001)
      IXN   =10000.0*(XN   +0.00001)
      IXN1  =10000.0*(XN1  +0.00001)
      IYJ   =10000.0*(YJ   +0.00001)
      IYJ1  =10000.0*(YJ1  +0.00001)
      IF (IXN-IXB2J) 15,16,16
   15 XLL=XN
      YLL=YJ
   20 IF (IXN-IXB2J1) 17,18,18
   16 IF (IXN.GE.IXB3J)GO TO 15
      IF (IYJ.GT.IYB1N) GO TO 19
      XLL=XB3J
      YLL=YB1N
      IND1=2
      GO TO 20
   19 XLL=XB3J
      YLL=YB3N
      IND1= 1.0
      GO TO 20
   17 XUL=XN
      YUL=YJ1
   22 IF (IXN1-IXB2J) 23,24,24
   18 IF (IXN.GT.IXB3J1) GO TO 17
      IF (IYJ.GT.IYB1N) GO TO 21
      XUL=XB3J1
      YUL=YB1N
      IND2=2
```

```
      GO TO 22
   21 XUL=XB3J1
      YUL=YB2N
      IND2= 1.0
      GO TO 22
   23 XLR=XN1
      YLR=YJ
  281 IF (IXN1-IXB2J1) 25,26,26
   24 IF (IXN1.GT.IXB3J) GO TO 23
      IF (IYJ.GT.IYB1N1)GO TO 27
      XLR=XB2J
      YLR=YB1N1
      IND3=2
      GO TO 281
   27 XLR=XB2J
      YLR=YB3N1
      IND3= 1.0
      GO TO 281
   25 XWR=XN1
      YWR=YJ1
      GO TO 29
   26 IF (IXN1.GT.IXB3J1) GO TO 25
      IF (IYJ.GT.IYB1N1) GO TO 28
      XWR=XB2J1
      YWR=YB1N1
      IND4=2
      GO TO 29
   28 XWR=XB2J1
      YWR=YB2N1
      IND4= 1.0
      GO TO 29
   11 YLL=YJ
      YUL=YJ1
      YLR=YJ
      YWR=YJ1
      XLL=XN
      XUL=XN
      XLR=XN1
      XWR=XN1
   29 DELYL=YUL-YLL
      IF(DELYL) 30,30,31
   30 VXL=0
   32 DELYR=YWR-YLR
      IF(DELYR) 33,33,34
   31 IF(IND1)  431, 431, 432
  432 PHILL = 1.0
      IF (IND1.EQ.2) PHILL=0.
      GO TO 433
  431 CALL TLUXXS(TAB3,XLL, YLL,0, PHILL)
  433 IF(IND2) 434, 434, 435
  435 PHIUL = 1.0
      IF (IND2.EQ.2) PHIUL=0.
```

```
      GO TO 46
  434 CALL TLUXXS(TAB3, XUL, YUL, 0, PHIUL)
   46 IF (ALAMDA.EQ.0.0) GO TO 436
      DELYL=((YLL+YUL)/2.)*DELYL
  436 VXL=(PHIUL-PHILL)/DELYL
      GO TO 32
   33 VXR=0
   35 CONTINUE
 3221 DELXL = XLR - XLL
      IF(DELXL) 36,36,37
   34 IF(IND3) 437, 437, 438
  438 PHILR =1.0
      IF (IND3.EQ.2) PHILR=0.
      GO TO 439
  437 CALL TLUXXS(TAB3,XLR,YLR,0, PHILR)
  439 IF(IND4) 440, 440, 441
  441 PHIWR =1.0
      IF (IND4.EQ.2) PHIWR=0.
      GO TO 3010
  440 CALL TLUXXS(TAB3,XWR,YWR,0,PHIWR)
 3010 IF (ALAMDA.EQ.0.0) GO TO 442
      DELYR=((YWR+YLR)/2.)*DELYR
  442 VXR = (PHIWR-PHILR)/DELYR
      GO TO 35
   36 VYL=0.0
   38 DELXU=XWR-XUL
      IF(DELXU) 39,39,40
   37 IF(IND3) 443, 443, 444
  444 PHILR =1.0
      IF (IND3.EQ.2) PHILR=0.
      GO TO 445
  443 CALL TLUXXS(TAB3,XLR,YLR,0,PHILR)
  445 IF(IND1) 446, 446, 447
  447 PHILL =1.0
      IF (IND1.EQ.2) PHILL=0.
      GO TO 448
  446 CALL TLUXXS(TAB3,XLL,YLL,0,PHILL)
  448 IF (ALAMDA.EQ.0.) GO TO 4481
      DELXL=YJ*DELXL
 4481 VYL= (PHILR-PHILL)/DELXL
      GO TO 38
   40 IF(IND4) 449, 449, 450
  450 PHIWR= 1.0
      IF (IND4.EQ.2) PHIWR=0.
      GO TO 451
  449 CALL TLUXXS(TAB3,XWR,YWR,0,PHIWR)
  451 IF(IND2) 452, 452, 453
  453 PHIUL= 1.0
      IF (IND2.EQ.2) PHIUL=0.
      GO TO 454
  452 CALL TLUXXS(TAB3,XUL,YUL,0,PHIUL)
  454 IF (ALAMDA.EQ.0.) GO TO 4541
```

```
      DELXU=DELXU*YJ1
 4541 VYW= (PHIWR-PHIUL)/DELXU
      WFS=0.0
      CONTER=0.0
      GO TO 4101
   39 VYW=0.0
      WFS=0.0
      CONTER=0.0
 4101 DIVX=0.0
      DIVY=0.0
      IF (DIRECT) 320,320,340
  320 IF (VYW.NE.0.) DIVY=1.
      IF (VYL.NE.0.) DIVY=DIVY+1.
      IF (VXL) 360,361,360
  361 VX=VXR
      GO TO 364
  360 IF (VXR) 363,362,363
  362 VX=VXL
      GO TO 364
  363 VX=VXR*((XNN-XLL)/(XLR-XLL))+VXL*((XLR-XNN)/(XLR-XLL))
  364 YMEAN=ABS((YMN1+YM)/2.0)
      VY=(VYL*(YMEAN-YJ1)+VYW*(YJ-YMEAN))/(YJ1-YJ)
      GO TO 3011
  340 IF(VXL) 42,43,42
   42 DIVX=1.0
   43 IF(VXR) 44,45,44
   44 DIVX=DIVX+1.0
   45 XMEAN =(XNN+XNN1)/2.
      VX=(VXL*(XN1-XMEAN)+VXR*(XMEAN-XN))/(XN1-XN)
      IF (VYW.NE.0.0.AND.VYL.NE.0.0) GO TO 365
      IF (VYW.NE.0.) GO TO 7001
      VY=-VYL
      GO TO 3011
 7001 IF (YJ.EQ.0.) GO TO 365
      VY=-VYW
      GO TO 3011
  365 VY = -VYW*((YN-YLL)/(YUL-YLL))-VYL*((YUL-YN)/(YUL-YLL))
 3011 VDX = VX/UPINF
      VDY=VY/UPINF
      IF(YM.LT.0.) VDY=-VDY
      FIRST=0.
      DWB=0.0
      VDXWM=VDXW
      VDYWM=VDYW
  325 REN=REINF*SQRT((VDX-VDXWM)**2+(VDY-VDYWM)**2)
      CK=1.+.197*REN**.63+.00026*REN**1.38
      IF (DIRECT) 336,336,337
  336 XX5X=VDYW**2+(YMN1-YM)*(VDY-VDYWM)*CK/(6.0*AK)-5.3623/(RHOW*UINF**
     X2)*(RHOW-PA)*(YMN1-YM)*GRAVTY
      GO  TO 338
  337 XX5X=VDXW**2+(XNN1-XNN)*(VDX-VDXWM)*CK/(6.0*AK)
  338 IF (XX5X.LT.0.) GO TO 370
```

```
      IF (WFS.GT.0.0) XX5X=0.0
      IF (DIRECT.GT.0.) GO TO 339
      VDYWN1=SQRT(XX5X)
      IF (VDYW.LT.0.) VDYWN1=-VDYWN1
      DTW=(YMN1-YM)/(6.0*(VDYWN1+VDYW))
      VDXWN1=VDXW+DTW*(VDX-VDXWM)*CK/AK
      GO TO 5010
  370 IF (DIRECT.GT.0.0) GO TO 348
      IF (CONTER.GE.4.0) GO TO 348
      CONTER=CONTER+1.0
      YMN1=(YM+YMN1)/2.0
      GO TO 4101
  348 IF (DWB.GT.0.0) GO TO 345
      DWB=1.0
      IF (DIRECT.GT.0.0) GO TO 346
      VDYWM=VDYW/2.0
      GO TO 325
  346 VDXWM=VDXW/2.0
      GO TO 325
  345 IF (DIRECT.GT.0.0) GO TO 347
      DIRECT=-DIRECT
      GO TO 2205
  347 XNN1=XNN-(VDXW*VDXW*6.0*AK)/(CK*(VDX-VDXW/2.0))
      CONTER=CONTER+1.0
      WFS=1.0
      IF (CONTER.LE.10.0) GO TO 4101
      VDYWM=VDYW
      XKT=26.0
      XX5X=0.0
      GO TO 339
  339 VDXWN1=SQRT(XX5X)
      IF (VDXW.LE.0.) VDXWN1=-VDXWN1
      IF (VDXW.EQ.0.0.AND.VDX.GT.0.0) VDXWN1=ABS(VDXWN1)
      DTW=(XNN1-XNN)/(6.0*(VDXWN1+VDXW))
      VDYWN1=VDYW+DTW*(VDY-VDYWM)*CK/AK-32.174*DTW*(RHOW-PA)/(RHOW*UINF*
     X*2)*GRAVTY
 5010 IF (VDXWN1.GT.VDXW) GO TO 4200
      IF (VDX.GE.VDXWN1) VDXWN1=VDX
      GO TO 5020
 4200 IF (VDX.LE.VDXWN1) VDXWN1=VDX
 5020 IF (VDYWN1.GE.VDYW) GO TO 4100
      WDY=VDY-32.174*(RHOW-PA)*(DTW)/(RHOW*UINF**2)*GRAVTY
      IF (VDYWN1.GE.WDY) GO TO 399
      VDYWN1=VDYW
      IF (WDY.LT.VDYW) VDYWN1=WDY
      GO TO 399
 4100 WDY=VDY-32.174*(RHOW-PA)*(DTW)/(RHOW*UINF**2)*GRAVTY
      IF (WDY.GT.VDYWN1) GO TO 399
      VDYWN1=VDYW
      IF (WDY.GT.VDYW) VDYWN1=WDY
  399 VDXWM1 = (VDXWN1 + VDXW)/2.0
      VDYWM1 = (VDYWN1 + VDYW)/2.0
```

```
      IF(XKT - 25.0)4999,4999,3028
 4999 IF(ABS(VDXW) - .01)3041,3040,3040
 3040 IF(ABS(VDXWM - VDXWM1) - ABS(.001 * VDXWM))3041,3041,3026
 3041 IF(ABS(VDYW) - .01)3028,3024,3024
 3024 IF(ABS(VDYWM - VDYWM1) - ABS(.001 * VDYWM))3028,3028,3025
 3025 IF(ABS(VDYWM - VDYWM1) - .0001 )3028,3028,3027
 3026 IF(ABS(VDXWM - VDXWM1) - .0001 )3041,3041,3027
 3027 DVXM=VDXWM1-VDXWM
      DVYM=VDYWM1-VDYWM
      IF (FIRST.LE.0.) GO TO 750
      VDYWM1=VDYWM-DVYM*((VDYWMX-VDYWM)/(DVYMX-DVYM))
      VDXWM1=VDXWM-DVXM*(VDXWMX-VDXWM)/(DVXMX-DVXM)
  750 VDXWMX=VDXWM
      VDYWMX=VDYWM
      VDXWM=VDXWM1
      VDYWM=VDYWM1
      FIRST=FIRST+1.
      DVXMX=DVXM
      DVYMX=DVYM
      XKT = XKT + 1.0
      IF(XKT-25.)325,325,5000
 5000 WRITE (6,3111)VDYWM,VDXWM
      VDXWM = VDXW
      VDYWM = VDYW
      GO TO 325
 3028 IF (DIRECT) 500,500,502
  500 XNN1=XNN+6.0*DTW*(VDXWN1+VDXW)
      XKT=XKT+1.0
      IF (ABS(XNN1-XNN).LE.DELX/55.0) GO TO 52
  460 IF (XKT.GT.26.0) GO TO 52
      DIRECT=-DIRECT
      GO TO 2205
  502 YMN1=YM+6.0*DTW*(VDYWN1+VDYW)
      XKT=XKT+1.0
      IF (ABS(YMN1-YM).LE.(YJ1-YJ)*1.05) GO TO 52
      GO TO 460
   52 WRITE (6,3110) XNN1,YMN1,DTW,CK,REN,VDX,VDXWN1,VDYWN1,VDY,XKT,
     XDIRECT
      XKT = 0.0
 3020 IF (XNN1-XMAX)54,53,53
   53 IF (ABS(YMN1)-YSM) 459,459,505
  459 IF (XPM1.EQ.0.0) GO TO 462
      IF (FN.NE.0.0) GO TO 471
      SAVE=YDROP
      FN=1.0
      GO TO 461
  471 FN=FN+1.0
  461 YDROP=YDROP-DELYDP/2.0**FN
      GO TO 306
  462 IF (FN.EQ.0.0) GO TO 102
      FN=FN+1.0
      YDROP=YDROP+DELYDP/2.0**FN
```

```
      GO TO 306
  505 IF (RANDOM) 5070,5070,102
 5070 IF (LANGMR.LE.0.) GO TO 2003
      ND=5+2*(NK-1)
      WRITE (6,630) R,(TAB10(I),I=6,ND)
      AVSMAX=TAB10(ND-1)
      AVS=FLOAT(INT(TAB10(6)*4.)-1)/4.
      GO TO(5040,5041,5042,5043,5044,5045,5046),IBEV
 5040 FACTOR=.3
 5048 APERAO=.71
      GO TO 5055
 5041 FACTOR=.2
 5049 APERAO=.52
      GO TO 5055
 5042 FACTOR=.1
 5050 APERAO=.31
      GO TO 5055
 5043 FACTOR=.05
 5051 APERAO=2.22
      GO TO 5055
 5044 FACTOR=.05
 5052 APERAO=1.74
      GO TO 5055
 5045 FACTOR=.1
 5053 APERAO=1.37
      GO TO 5055
 5046 FACTOR=.2
 5054 APERAO=0.0
 5055 R=APERAO*LANGMR
      BEV=IBEV
      WRITE (6,640)  BEV,FACTOR,APERAO
      IBEV=IBEV+1
 5056 CALL TLUXXS (TAB10,AVS,0,0,WATRAT)
      WRITE (6,640) WATRAT
      IF (AVS.GT.AVSMAX) WATRAT=0.
      WAWRT=FACTOR*WATRAT
      DO 5072 I=1,800,2
 5072 IF (TAB11(I).EQ.AVS) GO TO 5073
 5073 CUMWAT=TAB11(I+1)
      WRITE (6,640) CUMWAT,WAWRT,AVS
      CUMWAT=CUMWAT+WAWRT
      TAB11(I+1)=CUMWAT
      AVS=AVS+0.25
      IF (WATRAT.GT.0.0) GO TO 5056
      IEND=2*NK+5
      DO 5060 I=6,IEND
 5060 TAB10 (I)=0.
      NK=0
      IF (R.LE.0.) LANGMR=0.
      IF (I.GT.IMAX) IMAX=I
      WRITE (6,635) (TAB11(J),J=1,IMAX)
      GO TO 6041
```

```
   54 IF (ABS(YMN1)-YMAX)55,53,53
   55 CALL TLUXXS(TAB6,XNN1,0,0,YB2)
      WFSMB=0.
      XN1=XNN1
      IF (ABS (YMN1).LT.YB2) GO TO 1575
      CALL TLUXXS(TAB7,XNN1,0,0,YB3)
      IF (ABS(YMN1)-YB3) 57,100,100
 1575 CALL TLUXXS(TAB8,XNN1,0,0,YB1)
      IF (ABS(YMN1).GE.YB1) GO TO 100
 1577 CALL TLUXXS (TAB8,XN1,0,0,YB1)
      YB1N1=YB1
      CALL TLUXXS(TAB8,XNN,0,0,YB1)
      YB1N=YB1
      IF (XNN.GE.XHL1) GO TO 472
      YZ1=(YM*(XNN1-XHL1)+YMN1*(XHL1-XNN))/(XNN1-XNN)
      AMB=YB1N1/(XN1-XHL1)
      AMDT=(YMN1-YM)/(XNN1-XNN)
      IF (YZ1.GT.0.0) GO TO 473
      DIFX=(0.0-YM-AMDT*(XHL1-XNN))/(AMB+AMDT)
      XP=XHL1+DIFX
      YP=-AMB*DIFX
      AMDT=-AMDT
      GO TO 474
  473 DIFX=(YM+AMDT*(XHL1-XNN))/(AMB-AMDT)
      XP=XHL1+DIFX
      YP=AMB*DIFX
      GO TO 474
  472 AMB=(YB1N1-YB1N)/(XN1-XNN)
      AMDT=(ABS(YMN1)-YN)/(XNN1-XNN)
      DIFX=(YN-YB1N)/(AMB-AMDT)
      XP=XNN+DIFX
      YP=YB1N+AMB*DIFX
      IF (YMN1.LT.0.0) YP=-YP
  474 CONTINUE
      IF (ABS(XP-XN1 ).LE.0.0001) GO TO 3501
      XN1=XP
      WFSMB=WFSMB+1.0
      IF (WFSMB.LE.25.0) GO TO 1577
      GO TO 3501
   57 CALL TLUXXS (TAB6,XN1,0,0,YB2)
      YB2N1=YB2
      CALL TLUXXS (TAB6,XNN,0,0,YB2)
      YB2N=YB2
      IF(YB2N-YMAX) 157,158,158
  158 IF (XHL.LT.XN1) GO TO 1585
      XN1=XNN1+DELX2/100.
      GO TO 57
 1585 YZ=(ABS(YMN1)*(XHL-XNN)+YN*(XNN1-XHL))/(XNN1-XNN)
      IF (ABS(YZ)-YHL) 159,3500,160
  159 AMB=(YB2N1-YHL)/(XN1-XHL)
      GO TO 161
  160 CALL TLUXXS (TAB7,XN1,0,0,YB3)
```

```
      YB3N1=YB3
      AMB=(YB3N1-YHL)/(XN1-XHL)
  161 YO=YHL-AMB*(XHL-XNN)
      GO TO 60
  157 IF(YN-YB2N) 58, 59, 59
   58 AMB=(YB2N1-YB2N)/(XN1-XNN)
      CALL TLUXXS(TAB6,XNN,0,0,YB2)
      YO=YB2
      GO TO 60
   59 CALL TLUXXS (TAB7,XN1,0,0,YB3)
      YB3N1=YB3
      CALL TLUXXS(TAB7,XNN,0,0,YB3)
      YO=YB3
      AMB=(YB3N1-YO)/(XN1-XNN)
   60 AMDT=(ABS(YMN1)-YN)/(XNN1-XNN)
      DIFM=AMB-AMDT
      DIFY=YN-YO
      DIFX=DIFY/DIFM
      XP=DIFX+XNN
      DIFFY=AMB*DIFX
      YP=DIFFY+YO
      IF (YMN1.LT.0.) YP=-YP
      IF (ABS(XP-XN1).LE.0.0001) GO TO 3501
      XN1=XP
      WFSMB=WFSMB+1.
      IF (WFSMB.LE.25.) GO TO 57
      GO TO 3501
 3500 CONTINUE
      XP=XHL
      YP=YHL
 3501 IF(T(5) -XNN)3412,3412,3416
 3412 IF(T(6) - YN)3413,3413,3416
 3413 IF(T(1) -XNN)3416,3414,3414
 3414 IF(T(2) - YN)3416,3415,3415
 3415 JP = JP + 1
      IPB = IPA + JP
      XPLT(IPB) = XP
      YPLT(IPB) = YP
      IF (IPB.LT.3099) GO TO 3416
      JP=0
 3416 ID=ID+1
      XPLOT(ID)=XP
      YPLOT(ID) = YP
      IF (XNN1.LT.XTRAP1) GO TO 802
      IF (XNN1.GT.XTRAP2)GO TO  802
      IF (ABS(YMN1).LT.YTRAP1) GO TO 802
      IF (ABS(YMN1).GT.YTRAP2) GO TO 802
      WRITE (6,660)
      GO TO 102
  802 IF (RESIL.EQ.0.0) GO TO 9781
      ALPH=ATAN(AMB)
      BETA=1.57079
```

```
      IF(XNN1.NE.XNN) BETA=ATAN(AMDT)
      IF (VCXW.GE.0.0) GO TO 812
      IF (ABS(ALPHA-BETA)-1.57079) 810,804,808
  804 SIGN1=1.0
      SIGN2=1.0
      EPS=-1.57079
  806 VBIT=0.
      VDW1=SQRT(VDXW*VCXWN1+VDYW*VDYWN1)
      VB2N=VDW1*RESIL
      GO TO 818
  808 SIGN1=1.0
      SIGN2=1.0
      GO TO 816
  810 SIGN1=-1.0
      SIGN2=-1.0
      GO TO 816
  812 IF (ABS(ALPH-BETA)-1.57079) 808,814,810
  814 SIGN1=1.0
      SIGN2=1.0
      EPS=-1.57079
      GO TO 806
  816 VDW1=SQRT(ABS(VCXW*VCXWN1)+ABS(VDYW*VDYWN1))
      VBIN=VDW1*SIN(ALPH-BETA)
      VBIT=VDW1*COS(ALPH-BETA)
      VB2N=VBIN*RESIL
      EPS=ATAN(VB2N/VBIT)
  818 ANGLE=ALPH+EPS
      VDW2=SQRT(VBIT*VBIT+VB2N*VB2N)
      VDXWN1=VDW2*SIGN1*COS(ANGLE)
      VDYWN1=VDW2*SIGN2*SIN(ANGLE)
      IF (YP.LT.0.0) VDYWN1=-VDYWN1
      BOUNCE=BOUNCE+1.0
      YDROP=YP
      XL=XP
      VYW=VDYWN1*UINF
      VXW=VDXWN1*UINF
      IF (XL.GE.XMAX) GO TO 102
      IF (BOUNCE.GT.15.0) GO TO 102
      GO TO 315
 9781 X66=X66
      IF (XPM1.NE.0.0) GO TO 468
      IF (FN.EQ.0.0) GO TO 463
      FN=FN+1.0
      GO TO 464
  463 SAVE=YDROP
      FN=1.0
  464 YDROP=YDROP-DELYDP/2.0**FN
      IF (FN.LT.6.0) GO TO 306
      YTAN=YDROP
      YDROP=SAVE-DELYDP
      GO TO 1644
  468 IF (FN.EQ.0.0) GO TO 61
```

```
      FN=FN+1.0
      YDROP=YDROP+DELYDP/2.0**FN
      IF (FN.LT.6.) GO TO 306
      YTAN=YDROP
      YDROP=SAVE-DELYDP
      GO TO 61
   62 YPM1=YP
      XPM1=XP
      UNDERX=UNDER
      WRITE (6,3107)
      WRITE (6,1240)WATRAT,AVS,XP,YP,WATER
      WRITE (6,3107)
      NK=NK+1
      IF (NK.EQ.1) GO TO 475
      IDX=4+2*(NK-1)
      TAB10(IDX)=AVS
      TAB10(IDX+1)=WATRAT
      TAB10(3)=NK
      XP=0.0
  475 IF (FN.EQ.0.0) GO TO 102
      YP=0.0
      XPM1=0.0
      GO TO 6175
   61 AHITE=XP-XPM1
      WID=YP-YPM1
      SQHITE=AHITE**2
      SQWID=WID**2
      AHYPT=SQRT(SQHITE+SQWID)
      IF(ALAMDA) 1641, 1640, 1641
 1640 CONAER=AHYPT
      Z = 60.0* DELYDP
      IF (FN.EQ.0.0) GO TO 470
      Z=60.0*(YTAN-YDROP)
      GO TO 1642
  470 IF (PMM.EQ.0.0) GO TO 1642
      Z=60.0*(YDROP-YTAN)
      PMM=0
      GO TO 1642
 1641 SUMRAD=ABS((YP)+(YPM1))
      CONAER=SUMRAD*AHYPT
      Z=60.*(ABS(2.*YDROP-DELYDP))*DELYDP
      IF (FN.EQ.0.0) GO TO 6165
      Z=60.0*(YTAN-YDROP)*(YTAN+YDROP)
      GO TO 6167
 6165 IF (PMM.EQ.0.0) GO TO 6167
      Z=60.0*(YDROP-YTAN)*(YDROP+YTAN)
      PMM=0.0
 6167 IF (YP*YPM1.GE.0.0) GO TO 1642
      PHYP=SQRT(YP*YP+(XP-XHL1)**2)
      PMHYP=SQRT(YPM1**2+(XPM1-XHL1)**2)
      CONAER=(YP*PHYP)+ABS(YPM1*PMHYP)
 1642 WATER=Z*12.0*UINF*ALWC
```

```
      WATRAT=WATER/CONAER
 1644 XSS=XP-XHL
      XXSS = XPM1 - XHL
      CALL TLUXXS (TAB8,XP,0,0,YB1)
      IF (ABS(YP).LT.(YB1+.001)) GO TO 67
      CALL TLUXXS (TAB6,XP,0,0,YB2)
      YB2XP=YB2
      IF (ABS(YP).LT.(YB2XP+.001)) GO TO 68
      CALL TLUXXS (TAB2,XSS,0,0,SUP)
      UNDER=-1.0
      IF (UNDERX.LT.0.) GO TO 64
      GO TO 63
   68 CALL TLUXXS (TAB1,XSS,0,0,SUP)
      SUP=-SUP
      UNDER=1.0
      IF (UNDERX.LT.0.0) GO TO 64
   63 CALL TLUXXS(TAB1,XXSS,0,0,SUPM1)
      SUPM1=-SUPM1
      GO TO 65
   64 CALL TLUXXS (TAB2,XXSS,0,0,SUPM1)
   65 AVS = (SUP + SUPM1)/2.0
      IF (XPM1.NE.0.0) GO TO 62
 6175 AVS=SUP
      PMM=1.0
      WATRAT=0.0
      FN=0.0
      GO TO 62
   67 CALL TLUXXS (TAB9,XSS,0,0,SUP)
      CALL TLUXXS (TAB9,XXSS,0,0,SUPM1)
      IF (YP.GT.0.) GO TO 70
      UNDER=-1.
      SUP=-SUP
      GO TO 71
   70 UNDER=1.
   71 IF (UNDERX.LT.0.0) SUPM1=-SUPM1
      GO TO 65
      END
```