AFFDL-TR-70-16
VOLUME II

FLOW OVER AIRFOILS IN THE TRANSONIC REGIME - COMPUTER PROGRAMS

Richard J. Magnus
William H. Gallaher

CONVAIR DIVISION OF GENERAL DYNAMICS

TECHNICAL REPORT AFFDL-TR-70-16, VOLUME II

1 March 1970

Distribution of this Document is Unlimited

Air Force Flight Dynamics Laboratory
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Distribution of this Document is Unlimited.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AFFDL-TR-70-16
VOLUME II

FLOW OVER AIRFOILS IN THE TRANSONIC REGIME - COMPUTER PROGRAMS

Richard J. Magnus
William H. Gallaher

Distribution of this Document is Unlimited

FOREWORD

The present study was undertaken by the Convair Division of General Dynamics, P. O. Box 1128, San Diego, C lifornia under Air Force Contract F33615-69-C-1180 and Project No. 1366. The contract monitor was Captain Lowell Keel of the Air Force Flight Dynamics Laboratory (FDMM), Wright-Patterson Air Force Base, Ohio.

The principal investigator was Dr. H. Yoshihara. Dr. R. Magnus carried out the development of the inviscid procedure, while Mr. W. Gallaher carried out the computer programming for the viscous portion.

This report covers work undertaken from 15 November 1968 to 12 December 1969, and was submitted to the Air Force Flight Dynamics Laboratory on 12 December 1969. The report is in two volumes. Volume I contains the technical aspects and the results, while Volume II covers the details of the inviscid and viscous computer programs.

Philip P. Antonatos
Chief, Flight Mechanics Division
Air Force Flight Dynamics Laboratory

ABSTRACT

In this volume, Convair computer program Number P2586, for the problem setup and subsequent calculation of the compressible transonic flow about a blunted, lifting airfoil, is described. Input, output and operating instructions are given. Detailed flow charts and complete FORTRAN listings of the programs are also given. Also, the computer program for analysis of the separated flow bubble caused by interaction of a normal shock with a boundary layer is described.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF SYMBOLS

a	Speed of sound
e	Internal energy per unit mass
E	Internal energy plus kinetic energy per unit volume
F, G	Vector functions of the primary conservation variables
k	Diffusive damping coefficient
M	Mach number
n	Number of time planes to propagate a signal forward a distance Δx against the free stream
p	Pressure
Rh	Inverse of radius in polar coordinate system
t	Time
u	Velocity aligned with x coordinate
u_s	Signal velocity against the free stream
U	Total velocity
v	Velocity aligned with y coordinate direction
W	Dependent variables (vector) of system of conservation equations
x, y	Cartesian coordinates aligned and normal to airfoil chordline
α	(1) Angle of attack
	(2) A constant (Eqs. 10-12)
	(3) Matrix of $\frac{\partial F_i}{\partial W_j}$ (Eqs. 15-17)
β	Matrix of $\frac{\partial G_i}{\partial W_j}$

γ	Ratio of specific heats of gas
θ	Angular coordinate of polar coordinate system
ρ	Gas density

<u>Subscripts</u>

1	Along the x coordinate direction
2	Along the y coordinate direction
0	Reference value
∞	Free stream conditions
A	Conditions ahead of plane unsteady wave
B	Conditions behind plane unsteady wave

SECTION I

INTRODUCTION

This volume documents the computer programs used to calculate the transonic flow about two-dimensional airfoils. The approach used is discussed in Volume I. Since the numerical approach used in the inviscid flow region may be of interest to the investigator operating the computer programs, a portion is repeated in this volume. Sections II and III discuss the inviscid flow problem, the numerical approach, and some results of exercising the inviscid flow computer program. Section IV and V describe the setup and inviscid flow field programs.

The approaches used for the boundary layer on the transonic airfoil are discussed in Volume I. These approaches have existed for some time and therefore were not repeated in this volume. Section VI presents the computer program used to calculate the boundary layer on a two-dimensional airfoil.

These computer programs are operational on the CDC-6400 machine located at Convair Division of General Dynamics. Attempts were made to call library functions and subroutines that would be operational on other digital computers. A complete FORTRAN listing of program and SC-4020 generated flow charts are presented in the appendixes.

BLANK PAGE

SECTION II

DESCRIPTION OF THE INVISCID FLOW PROBLEMS

The program is designed to calculate planar compressible, transonic flows about blunted, lifting airfoils at angle-of-attack. Starting from an arbitrary initial guess of the flow field a two-step finite-difference scheme is used to numerically integrate the exact unsteady Euler equations. The large time or asymptotic solution is taken to represent the steady flow solution. The flow field is represented by three principal dependent variables at a number of nodes of cartesian coordinate mesh regions and an outer polar coordinate system. Free stream conditions are maintained at infinity by use of a transformation of the polar coordinate system.

1. BASIC EQUATIONS

Using cartesian coordinates, the unsteady inviscid equations for compressible flow are (from Chapter 7 of Liepmann and Roshko, Reference 1):

Continuity

$$\frac{\partial \rho}{\partial t} = \frac{\partial (-\rho u)}{\partial x} + \frac{\partial (-\rho v)}{\partial y} \tag{1}$$

Momentum

$$\frac{\partial (\rho u)}{\partial t} = \frac{\partial}{\partial x} (-\rho u^2 - p) + \frac{\partial}{\partial y} (-\rho u v) \tag{2}$$

$$\frac{\partial (\rho v)}{\partial t} = \frac{\partial}{\partial x} (-\rho u v) + \frac{\partial}{\partial y} (-\rho v^2 - p) \tag{3}$$

Energy

$$\frac{\partial E}{\partial t} = \frac{\partial}{\partial x} \left[- u (E + p) \right] + \frac{\partial}{\partial y} \left[- v (E + p) \right] \tag{4}$$

where $$E = \rho \left[e + \frac{1}{2} (u^2 + v^2) \right] \tag{5}$$

In addition, to make a consistent system, an equation of state is necessary.

$$p = (\gamma - 1) \rho e \tag{6}$$

The equations, (1) - (6), are applicable to flows of perfect gases at any Mach number. We will simplify the system by assuming that shocks are weak and the flow is, therefore, isentropic. Hence, (4) - (6) will be replaced by:

$$\left(\frac{p}{p_o}\right) = \left(\frac{\rho}{\rho_o}\right)^{\gamma} \tag{7}$$

where p_o and ρ_o are reference conditions.

The free stream properties have been adopted for the reference quantities and a system of non-dimensional variables has been used. The fundamental units chosen were the free stream density, the free stream sound speed, and the airfoil chord. Free stream properties then have the following numerical values:

Speed of sound	$a_\infty = 1.0$
Velocity	$U_\infty = M_\infty$
Density	$\rho_\infty = 1.0$
Pressure	$p_\infty = \frac{1}{\gamma}$
Internal energy	$e_\infty = \frac{1}{\gamma (\gamma - 1)}$
Total energy	$E = \rho_\infty (e_\infty + \frac{1}{2} U_\infty^2) = \frac{1}{\gamma (\gamma - 1)} + \frac{1}{2} M_\infty^2$
Total enthalpy	$H_o = \frac{1}{\gamma - 1} + \frac{1}{2} M_\infty^2$

Equations (1) - (3) are in conservation form and will be regarded as a coupled system which can be represented in vector form:

$$W_t = F_x + G_y \tag{8}$$

The functions W, F, and G are three component column vectors:

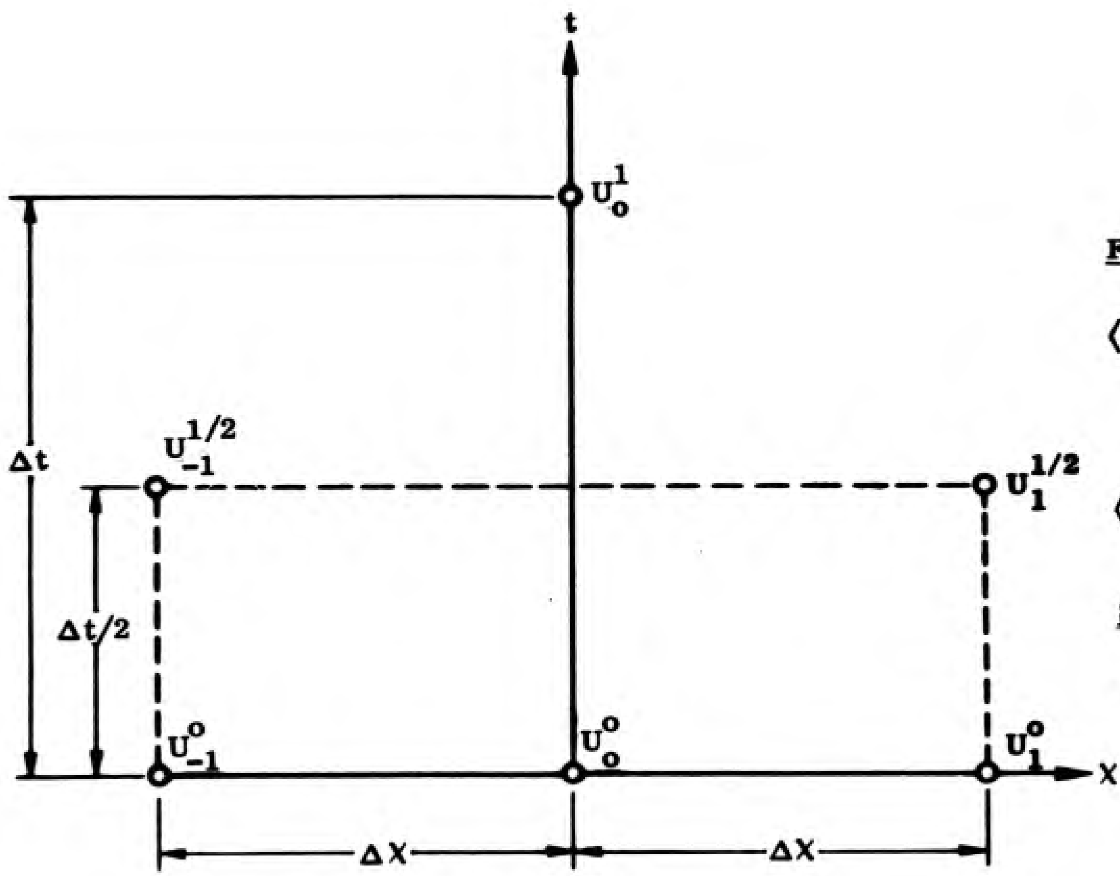
$$W = \begin{bmatrix} \rho u \\ \rho v \\ \rho \end{bmatrix} \qquad F = -\begin{bmatrix} \rho u^2 + p \\ \rho uv \\ \rho u \end{bmatrix} \qquad G = -\begin{bmatrix} \rho uv \\ \rho v^2 + p \\ \rho v \end{bmatrix} \tag{9}$$

2. BOUNDARY CONDITIONS

For the boundary conditions far from the airfoil, the flow is required to approach a uniform flow with the conditions $u = U_\infty \cos \alpha$, $v = U_\infty \sin \alpha$, and $\rho = \rho_\infty$. For convenience in calculating flows over a given airfoil at several angles of attack the computer program maintains the x axis aligned aft along the airfoil chordline. The origin is at the airfoil nose and y positive is upward. For all positive times the component of the velocity vector normal to the profile surface is required to be zero, and the Kutta condition is fulfilled at the sharp trailing edge. Initially the profile is assumed to be sufficiently "leaky" that a uniform flow at the free stream condition persists everywhere. At zero time the leakiness is impulsively turned off by suddenly imposing the zero normal velocity and Kutta conditions. Eq. (8) is then used to determine the subsequent subsidence of the flow to the desired steady state.

3. FINITE DIFFERENCE PROCEDURE AT FIELD POINTS

The system of partial differential equations given by (8) is replaced by a system of difference equations by first replacing the continuous x, y, t, space by a lattice of nodal points, and then substituting (for the partial derivatives) partial differences expressed in terms of the values of the dependent variables at the lattice points. A rectangular lattice with the



DIFFERENTIAL EQUATION

$$\frac{\partial U}{\partial t} = \alpha \frac{\partial U}{\partial X}$$

FIRST STEP

$$\langle U_1^{1/2} \rangle = U_1^o + \frac{\alpha \Delta t}{2} \left(\frac{3U_1^o - 4U_o^o + U_{-1}^o}{2\Delta X} \right)$$

$$\langle U_{-1}^{1/2} \rangle = U_{-1}^o + \frac{\alpha \Delta t}{2} \left(\frac{-U_1^o + 4U_o^o - 3U_{-1}^o}{2\Delta X} \right)$$

SECOND STEP

$$\langle U_o^1 \rangle = (1-k)\, U_o^o + k \left(\frac{U_1^o + U_{-1}^o}{2} \right) + \alpha \Delta t \left(\frac{\langle U_1^{1/2} \rangle - \langle U_{-1}^{1/2} \rangle}{2\Delta X} \right)$$

Figure 1. Illustration of application of difference scheme to a simple one-dimensional unsteady problem.

spacing Δx, Δy, and Δt in the respective coordinate directions is used. For the partial derivatives a modification of the explicit Lax-Wendroff second order differencing scheme has been chosen. The scheme may be illustrated in principle by applying it to the simple equation:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial u}{\partial x} \tag{10}$$

where α is a constant. Considering the basic lattice element shown in Fig. 1, the marching procedure for determining the value of u_o^1 knowing the values u_o^o, u_1^o, and u_{-1}^o at the initial time plane will be described; here we have used the notation $u\ (m\Delta x,\ n\Delta t) = u_m^n$.

The differencing scheme is carried out in two steps. As the first step, construction points $u_1^{1/2}$ and $u_{-1}^{1/2}$ on an auxiliary time plane at $t = \frac{\Delta t}{2}$ are defined by means of the equations:

$$u_1^{1/2} = u_1^o + \frac{\alpha \Delta t}{2} \left(\frac{3u_1^o - 4u_o^o + u_{-1}^o}{2\Delta x} \right)$$

$$u_{-2}^{1/2} = u_{-1}^o + \frac{\alpha \Delta t}{2} \left(\frac{-u_1^o + 4u_o^o - 3u_{-1}^o}{2\Delta x} \right) \tag{11}$$

Here the quantities in the parentheses are the finite difference approximation for $\frac{\partial u}{\partial x}$. Eqs. 11 therefore represent linear Taylor expansions. The second and final step of the two step scheme gives the sought value u_o^1 in terms of the known initial values by the Taylor expansion:

$$u_o^1 = \left[(1 - k)\ u_o^o + k \left(\frac{u_1^o + u_{-1}^o}{2} \right) \right] + \alpha \Delta t \left(\frac{u_1^{1/2} - u_{-1}^{1/2}}{2\Delta x} \right) \tag{12}$$

where the values of $u_{1}^{1/2}$ and $u_{-1}^{1/2}$ are given by (11), and k is a constant such that $0<k<1$. Note here that the linear Taylor coefficient is evaluated on the auxiliary time plane at $\Delta t/2$. In the term in the square brackets of (12) use is made of the equivalence of u_{0}^{0} with the average of u_{1}^{0} and u_{-1}^{0}. The bracketed term thus represents a weighted average of these two equivalents with the weighting constant k. The constant k is called the diffusive damping coefficient in the remainder of this volume. When the coefficient k is zero or is of order Δx, the above procedure leads to an approximation with a truncation error of the order of Δt^3 or Δx^3, otherwise the errors will be of second order.

The significance of the diffusive damping constant k and the generation of an artificial viscosity by the use of the above second order difference scheme can both be shown by considering the resulting difference equation. Assuming that the various terms in this equation are continuous functions of their arguments, and expanding each of the terms in a Taylor series terminating each series consistently after the linear terms (see Hirt Ref. 2). The result will be a differential equation "equivalent" to the difference equation, containing the original differential Equation (10) with however two additional terms both representing a diffusion effect. One of the diffusion terms contains a space derivative of second order with a "coefficient of viscosity" proportional to k, while the second term contains a third derivative with a "coefficient of viscosity" proportional to Δt. The latter viscosity arises from the use of the second order differencing scheme, and its similarity to the artificial viscosity of Von Neumann and Richtmyer led Lax to label it also as an artificial viscosity. The above diffusive terms also represent the truncation error which arises due to the use of the difference scheme. The "equivalent" differential equation as derived from the difference scheme is especially useful since it provides an insight into the numerical process. One may see, for example, from the "equivalent" differential equation that the undesirable truncation error terms are in fact responsible for the indispensable damping necessary for stability[1]. In a particular calculation, examination of the diffusive terms would also identify local flow regions where non-linear instabilities would be expected by noting where these terms vanished.

---

[1] It would therefore be unwise to reduce the truncation error any more than is required to obtain results of meaningful accuracy.

The above difference procedure differs from the Lax-Wendroff procedure in two aspects. The first is the introduction of the diffusive damping coefficient k ($k = 0$ in the Lax-Wendroff procedure), and the second is in the positioning of the points on the auxiliary time plane at $\pm\Delta x$ instead of the $\pm\Delta x/2$. The latter modification has the convenience of not having to introduce points at the half-mesh locations in the space plane[2]. The former modification for non-zero values of k has an essential advantage of enchancing the artificial viscosity beyond that arising by the use of the second order scheme when such additional damping is required locally or temporarily to maintain stability. This flexibility however, possibly is obtained at the expense of a decreased order of accuracy depending on the relative magnitude of k.

The previous paragraphs have illustrated the finite difference analogue for the case of the simplified Equation (10). If the same procedure is applied instead to the unsteady Euler equation given by Equation (1) all of the considerations arising for the simplified equation would apply. The additional partial derivatives in y and the coefficients for the space derivatives which are, in general, functions of u, v, and ρ must be incorporated. To retain the second order accuracy these latter coefficients must be evaluated consistently; that is, at appropriate points in the initial time plane when applying the first step, and in the auxiliary time plane (at $t = \Delta t/2$) when applying the second step. The addition of the y-derivatives causes no essential difficulties. The difference scheme is shown in Fig. 2.

In the first step, the values of the components of the vectors $W^{1/2}_{1,o}$, $W^{1/2}_{-1,o}$, $W_{o,1}$, and $W^{1/2}_{o,-1}$ would be calculated using equations:

$$W^{1/2}_{1,o} = W^{o}_{1,o} + \frac{\Delta t}{2}\left[\frac{3F^{o}_{1,o} - 4F^{o}_{o,o} + F^{o}_{-1,o}}{2\Delta x} + \frac{G^{o}_{1,1} - G^{o}_{1,-1}}{2\Delta y}\right]$$

---

[2] This is of no particular advantage in a problem using cartesian coordinates but is an advantage when a problem using orthogonal curvilinear coordinates is being solved because the components of the metric tensor need only be known at the ordinary mesh points.

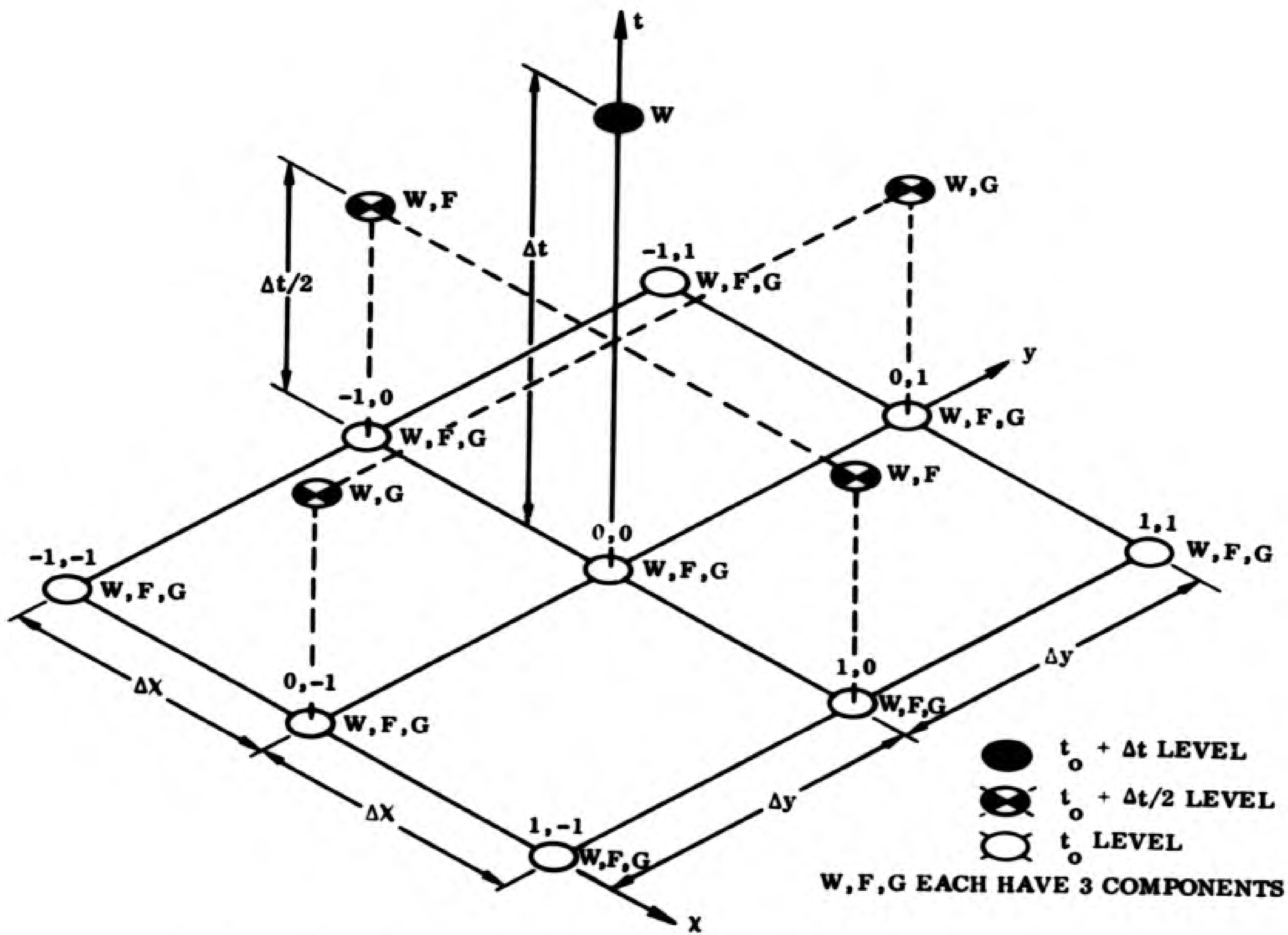


Figure 2. Construction points used by symmetric difference scheme for advancing solution in Cartesian mesh.

$$W^{1/2}_{-1,o} = W^{o}_{-1,o} + \frac{\Delta t}{2}\left[\frac{-3F^{o}_{-1,o} + 4F^{o}_{o,o} - F^{o}_{1,o}}{2\Delta x} + \frac{G^{o}_{-1,1} - G^{o}_{-1,-1}}{2\Delta y}\right]$$

$$W^{1/2}_{o,1} = W^{o}_{o,1} + \frac{\Delta t}{2}\left[\frac{F^{o}_{1,1} - F^{o}_{-1,1}}{2\Delta x} + \frac{3G^{o}_{o,1} - 4G^{o}_{o,o} + G^{o}_{o,-1}}{2\Delta y}\right]$$

$$W^{1/2}_{o,-1} = W^{o}_{o,-1} + \frac{\Delta t}{2}\left[\frac{F^{o}_{1,-1} - F^{o}_{-1,-1}}{2\Delta x} + \frac{-3G^{o}_{o,-1} + 4G^{o}_{o,o} - G^{o}_{o,1}}{2\Delta y}\right] \qquad (13)$$

The components of $W^{1/2}_{1,o}$ and $W^{1/2}_{-1,o}$ are converted to $F^{1/2}_{1,o}$ and $F^{1/2}_{-1,o}$ and the $W^{1/2}_{o,1}$ and $W^{1/2}_{o,-1}$ converted to $G^{1/2}_{o,1}$ and $G^{1/2}_{o,-1}$ using the algebraic relations implied in (9).

In the second step,diffusive damping may be incorporated by choice of a non-zero value for parameter k:

$$W^{1}_{o,o} = \left\{1 - \frac{k}{2}\left[1 + \left(\frac{\Delta x}{\Delta y}\right)^{2}\right]\right\}W^{o}_{o,o} + \frac{k}{4}\left[(W^{o}_{1,o} + W^{o}_{-1,o}) + \left(\frac{\Delta x}{\Delta y}\right)^{2}(W^{o}_{o,1} + W^{o}_{o,-1})\right]$$

$$+ \Delta t\left[\frac{F^{1/2}_{1,o} - F^{1/2}_{-1,o}}{2\Delta x} + \frac{G^{1/2}_{o,1} - G^{1/2}_{o,-1}}{2\Delta y}\right] \qquad (14)$$

Using the usual approach for linearized stability and accuracy analyses the equations (8) may be rewritten as:

$$W_t = \alpha W_x + \beta W_y \tag{15}$$

where α, β are matrices whose elements are:

$$\frac{\partial F_i}{\partial W_j} \text{ and } \frac{\partial G_i}{\partial W_j} \text{ respectively.}$$

Then regarding the elements of α and β to be locally constant the equivalent of the centered two-step scheme (13, 14) is:

$$W^1_{o,o} - W^o_{o,o} = \frac{k}{4} (\Delta x)^2 \left[\left(W_{xx} + \frac{\Delta x^2}{12} W_{xxxx} + \ldots\right) + \left(W_{yy} + \frac{\Delta y^2}{12} W_{yyyy} + \ldots\right) \right]$$

$$+ \Delta t \left[\alpha \left(W_{\underline{x}} + \frac{\Delta x^2}{6} W_{xxx} + ..\right) + \beta \left(W_{\underline{y}} + \frac{\Delta y^2}{6} W_{yyy} + ..\right) \right]$$

$$+ \frac{\Delta t^2}{2} \left[\alpha^2 \left(W_{\underline{xx}} + \frac{\Delta x^2}{12} W_{xxxx} + ..\right) + \right.$$

$$+ (\alpha\beta + \beta\alpha) \left(W_{\underline{xy}} + \frac{\Delta x^2}{6} W_{xxxy} + \frac{\Delta y^2}{6} W_{xyyy} + \ldots\right)$$

$$\left. + \beta^2 \left(W_{\underline{yy}} + \frac{\Delta y^2}{12} W_{yyyy} + ..\right) \right] \tag{16}$$

Whereas, by comparison, a series expansion of the functions in the vicinity of the reference point would give:

$$W^1_{o,o} - W^o_{o,o} = \Delta t \left[\alpha W_x + \beta W_y\right] + \frac{\Delta t^2}{2}\left[\alpha^2 W_{xx} + (\alpha\beta + \beta\alpha) W_{xy} + \beta^2 W_{yy}\right]$$

$$+ \frac{\Delta t^3}{6}\left[\alpha^3 W_{xxx} + (\alpha\alpha\beta + \alpha\beta\alpha + \beta\alpha\alpha) W_{xxy} + (\alpha\beta\beta + \beta\alpha\beta + \beta\beta\alpha) W_{xyy}\right.$$

$$\left. + \beta^3 W_{yyy}\right] + .. \quad (17)$$

The underlined terms in the difference scheme (16) faithfully reproduce terms in the series expansion (17). The remaining terms may be regarded as the discretization errors due to use of finite steps Δx, Δy, and Δt and intentionally added damping terms but, as a stability analysis would demonstrate, at least some of these terms are necessary to prevent divergence in repeated applications of the scheme.

As a stationary state is approached after repeated applications of the difference scheme, the right hand side of (16) must approach zero whereas the steady flow equations would require:

$$\alpha W_x + \beta W_y = 0,$$

$$\alpha^2 W_{xx} + (\alpha\beta + \beta\alpha) W_{xy} + \beta^2 W_{yy} = 0, \text{ etc.}$$

It is evident that the stationary descriptions of αW_x and βW_y obtained by repeated applications of the difference scheme will not agree exactly with the steady flow solution because of the presence of the non-zero discretization error and stabilization terms in (16).

It is to be expected that the errors in the desired terms are not random. The higher order derivatives tend to be continuous over extensive regions of the field and, therefore the nature (signs at least) of the discretization errors tend to be similar over broad regions of the field. Further, the general dissipative nature of the stabilization terms might be expected to result in deficiencies of the conservation variables at loci remote from the field boundary (at infinity) at which conditions are being absolutely maintained because of accumulation of the non-random errors in the individual cells along the paths from the boundary.

Experience gained in working on the present transonic airfoil problem has shown that the stationary numerical solution of the difference equations does tend to contain errors in total enthalpy (or stagnation pressure) which seems to be directly dependent on distance (number of meshes) from the free stream boundary at infinity. Although a second order scheme is used which tends to limit the errors within a mesh, the accumulation of errors over the approximately 20 or more meshes between infinity and the airfoil is serious. On a typical problem it was found, for example, that the stagnation point pressure coefficient was low by about 1/3 of the expected value; too much to be tolerated in an engineering solution.

Theoretically the error could be halved by doubling the number of mesh points between the airfoil and the free stream boundary but this is impractical because it would require roughly 4 times the data storage and 8 times the computing cost to solve the problem. Since each plane of the numerical solution process can be regarded as a refined guess as to the flow field, the present computer program contains a means for occasionally resetting the total enthalpy at all field points. Arbitrarily, it was chosen to make the adjustment by preserving Mach number and flow direction and adjusting the pressure (so that the stagnation point is treated properly). The stationary state for the numerical solution using the difference equations plus total enthalpy adjustment avoids some of the unpalatable manifestations of the accumulations of discretization errors.

4. FINITE DIFFERENCE PROCEDURE AT BOUNDARIES

Use of the procedure described in the previous section enables one to march in time for interior points where all of the points involved lie within the flow domain. For lattice points at or near the boundaries, where the regular configuration is no longer possible, a modified procedure must be used. Moreover, at the boundaries a method to impose the boundary conditions must be chosen. The procedure could be significantly simplified if the lattice could be continued in a regular manner into the interior of the profile. However the analytical continuation of the flow into the profile interior, primarily in the nose region, might contain severe gradients in the flow variables and even be inflicted with singularities in the form of limiting lines (or regression lines), where the flow is supersonic. With such a severe flow in the interior of the profile it would be imprudent to extend the lattice into this region.

Thus to calculate the points in the vicinity of and on the profile we superimpose on the underlying basic rectangular network of points a sequence of rotated nine point lattice elements, one of which is shown in Fig. 3. The superimposed lattice elements are identical to the underlying basic elements but are rotated such that the bottom triplet of points forms a tangent to the airfoil surface at the middle point. Each rotated lattice element is used to advance the solution at the center point as well as the boundary point. For the latter point a suitable asymmetric difference scheme is used. Sufficient overlap of this rotated sequence of lattice elements with the underlying regular network is maintained to obtain the necessary continuity between the two systems of lattice points.

The required initial values at the nine points of the rotated lattice element are obtained from the calculations at a prior time step by a suitable interpolation of the results from the basic as well as the rotated lattice elements. In the first step of the two step scheme, values of $W_k^{1/2}$ are obtained above loci k = 1, 2, 3, 4, 5, 6 and 8 using simple diffusion stabilized difference operators of the type illustrated by (13). The W components are converted to F or G components as needed and the symmetric scheme given in (14) is used to advance the solution at Pt. 5. The second step for obtaining the value W^1 at locus 2 (the wall point) which is used in the present computer program is:

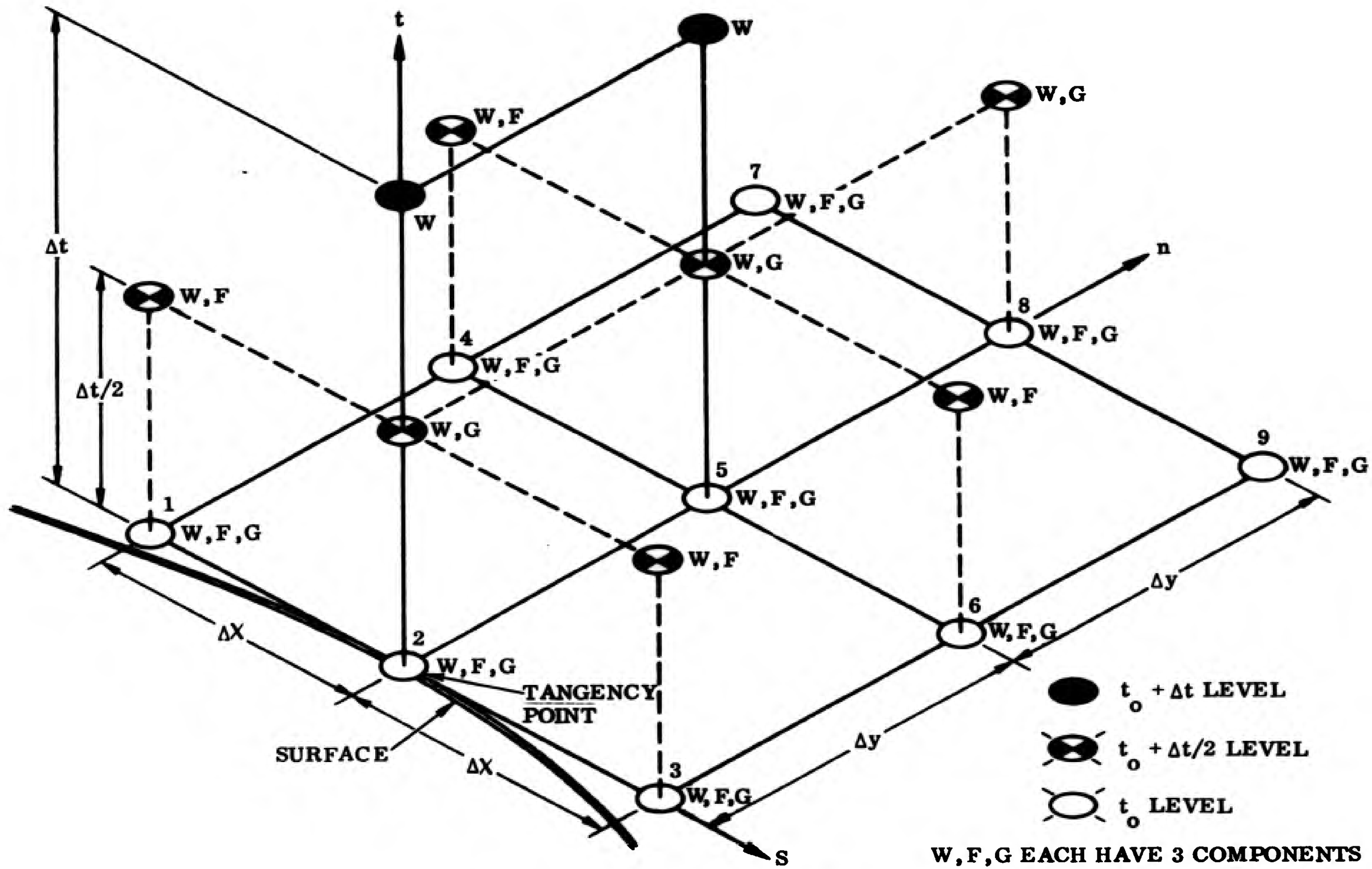


Figure 3. Construction points used by difference scheme for advancing solution in local rotated mesh tangent to airfoil surface.

$$W_2^1 = \left\{ 0.5\ (1.0 - k) + 0.5 \left[1.0 - \left(\frac{\Delta x}{\Delta y}\right)^2 k \right] + 0.25 \left(\frac{\Delta x}{\Delta y}\right)^2 k \right\} W_2^o$$

$$+ 0.25\ k\ (W_1^o + W_3^o) + 0.5k \left(\frac{\Delta x}{\Delta y}\right)^2 W_5^o - 0.25\ k \left(\frac{\Delta x}{\Delta y}\right)^2 W_8^o$$

$$+ \frac{\Delta t}{2} \left\{ \frac{F_3^{1/2} - F_1^{1/2}}{2\Delta x} + \frac{-3G_2^{1/2} + 4G_5^{1/2} - G_8^{1/2}}{2\Delta y} \right\} \qquad (18)$$

The weights attached to the various W_k^o in the equation were chosen to provide equal diffusive properties in the x and y directions when $\Delta x \neq \Delta y$. Here, in this locally rotated mesh, the clockwise tangent to the airfoil surface is regarded as the x direction and the outward surface normal is regarded as the y direction. Within the computer program the primary dependent variables for the local rotated meshes are written in terms of velocity components along the clockwise tangent and outward normal. The resulting velocity normal to the airfoil is not necessarily zero. A plane unsteady wave is introduced at Pt. 2 tangent to the airfoil at Pt. 2 of an appropriate strength and velocity which reduces the normal velocity to zero. Specifically, if the values of the W components at the wall obtained after application of (18) are:

$$W_{(1A)} = \rho_A\, u_{tA}$$

$$W_{(2A)} = \rho_A\, U_{nA}$$

$$W_{(3A)} = \rho_A \qquad (19)$$

the values obtained after insertion of the unsteady wave used to satisfy boundary conditions are:

$$W_{(3B)} = \left[\left(W_{(3A)} \right)^{\frac{\gamma-1}{2}} - \frac{\gamma-1}{2}\ \frac{W_{(1A)}}{W_{(3A)}} \right]^{\frac{2}{\gamma-1}}$$

$$W_{(1B)} = \frac{W_{(1A)}}{W_{(3A)}}\ W_{(3B)}$$

$$W_{(2B)} = 0 \qquad (20)$$

This plane unsteady wave is simply an artifice to introduce the influence of those points in the interior of the profile that fall in the domain of dependence of the boundary point in question. The velocity component tangent to the surface is kept invariant. The necessary wave strength, its velocity as well as the density behind the wave are obtained by the "jump conditions" for a moving wave.

For the fulfillment of the conditions at infinity we first represent the far field by a polar coordinate system and map the region exterior to a circle of a given large radius (taken to be 1.4 chords in our example) into the interior of a circle by an inverse transformation such that infinity in the physical plane is mapped to the origin of the transformed domain. A lattice configuration conforming to a polar coordinate representation is then introduced, and the calculation in the far field region is carried out using the transformed difference equations imposing the free stream conditions at the origin. Sufficient overlap of this far field lattice configuration with the rectangular system is maintained to obtain a proper patching of the two flow domains. The conversion to an inverted polar coordinate system has been carried out only with respect to the spatial independent variables; the components of the dependent conservation variables have the same (cartesian) momentum components that are used in the cartesian mesh region. This was done in order to eliminate drift in free stream properties due to truncation errors of a finite difference scheme used in a curvilinear coordinate system.

$$Rh = \frac{r_o}{r} = \frac{1.4}{\sqrt{x^2 + y^2}}$$

$$\tan\theta = \frac{y}{x} \qquad (21)$$

The Rh space was divided into 8 increments ranging from zero (representing an infinite radius) to unit (representing a circle of 1.4 chords radius). The θ space was divided into 40 sectors each 9 degrees wide. The equations in the polar coordinate region are obtained by straightforward rewriting of the original equations (8).

$$W_t = \left[F_{Rh}\frac{\partial Rh}{\partial x} + F_\theta \frac{\partial\theta}{\partial x}\right] + \left[G_{Rh}\frac{\partial Rh}{\partial y} + G_\theta \frac{\partial\theta}{\partial y}\right] \qquad (22)$$

In the rectangular lattice region a refinement of the lattice is also introduced where large flow gradients are expected such as near the leading edge of the profile. A fine mesh is embedded in a coarser mesh such that the size of the coarser mesh is an integral multiple of the fine mesh. Suitable overlap of the two lattice systems is again provided for proper matching of the two systems. In the overlap region adequate enhanced damping is incorporated to maintain stability, and the diffusive damping coefficients for the coarse and fine mesh systems are adjusted to try to maintain an approximate "viscosity match" between the two systems. In the example presented there are 5 cartesian mesh systems ranging in size from 0.0025 chord squares at the nose up to 0.20 chord squares for outer portions of the flowfield. A considerable part of the computer program logic is concerned with exchange of information along the boundaries of the various mesh systems.

5. ALLOWABLE TIME STEP

An explicit difference scheme, such as has been described, generally is less complicated (as to amount of computer logic) than an implicit scheme and may require fewer cells to store working variables but has the disadvantage that the "solution" will diverge unless the time step sizes are smaller than some strictly definable upper limit.

For the scheme described the time step size criterion is:

$$\Delta t \le \frac{\Delta x \, \Delta y}{u_1 \, \Delta y + u_2 \, \Delta x + a \left[(\Delta x)^2 + (\Delta y)^2\right]^{1/2}} \tag{23}$$

where a is the sound speed;

$$a = a_\infty \left(\frac{\rho}{\rho_\infty}\right)^{\frac{\gamma-1}{2}} \tag{24}$$

The criterion is roughly equivalent to the Courant-Freidrichs-Lewy criterion; referring to Figure 2, the backward Mach cone through point $(x_o, y_o, t_o + \Delta t)$ should intersect the t_o plane within the confines of the lozenge having apexes at $(x_o \pm \Delta x, y_o)$ and $(x_o, y_o \pm \Delta y)$. Since, when the time step is to be determined the flow properties are unknown at $(x_o, y_o, t_o + \Delta t)$, the properties at (x_o, y_o, t_o) are used in (23) and a safety factor is applied so that Δt is less than the stability limit. The regions with a coarser space mesh are permitted to march with a larger time step. Information is exchanged between regions after each computational step even though the time steps are different.

It should be noted that (23) is the time step size criterion for the scheme with negligible additional diffusive damping. A complete analysis has not been carried out for the case of strong diffusive damping but there are indications that a factor $\sqrt{1 - 2k}$ should be included in the numerator of (23) and that $k > 0.5$ will result in instability.

6. TIME TO REACH STEADY FLOW

The general approach used in the computer program of altering an initial guess as to the flowfield in a manner based on the unsteady equations makes it desirable to have some estimate of the amount of time which might accumulate before the flow would be reasonably steady. The question is complicated by the use of meshes of different sizes in various parts of the field and by the procedure of using time steps which are near the allowable in each mesh. However, a useful estimate of the number of planes (that is passes through the field executed by the computer) needed for the first chance of a steady flow may be given.

The crucial source of signals for adjusting the flow to the steady condition may be taken as the trailing edge where the Kutta condition is being applied. It may be argued that these signals must propagate forward to affect the location of the stagnation point and the general flowfield about the airfoil. The signals will travel forward across a mesh at a velocity roughly equal to $u_s = (a - u)$ and since the time increments used are roughly $\Delta t = k\Delta x/(a + u)$ the number of time steps necessary for the signal to progress forward a distance Δx is about:

$$n = \frac{\Delta x}{u_s \, \Delta t} = \frac{a + u}{k\,(a - u)} \qquad (25)$$

Then, taking the number of Δx increments between the leading and trailing edges to be m, we could expect that the number of planes to be calculated before the effect of the Kutta condition would be felt at the nose would be about

$$N = \frac{m\ (a + u)}{k\ (a - u)} \tag{26}$$

For example, if there were m = 50 mesh increments along the airfoil surface and the flow was Mach 0.8 and a safety factor k = 0.67 was being used we would expect that:

$$N \doteq \frac{50\ (1.0 + 0.8)}{0.67\ (1.0 - 0.8)} = 675$$

This sort of estimate should not be taken as anything more than a minimum number of passes through the field before the flow begins to have a resemblance to the steady state. Actually, local variations in velocity and sound speed and the need to adjust the flow several chords away from the airfoil may make it necessary to calculate several times the number of planes given by (26) before an acceptably steady (from an engineer's point of view) solution is obtained.

SECTION III

APPLICATION TO THE TRANSONIC FLOW OVER AN AIRFOIL

The present procedure was applied to the calculation of transonic flow over a NACA 64A-410 profile at an angle-of-attack of 4° and at a Mach number of 0.72. This section will describe the airfoil, mesh regions used to represent the flow field, and results of applying the present procedure.

1. AIRFOIL DESCRIPTION

The airfoil computed is a NACA 64A-410 profile with coordinates given by Reference 3. The present setup computer program used this basic set of information and fitted the points with a series of spline functions.

2. SPATIAL FIELD DESCRIPTION

The system of lattice points used to calculate the present example of transonic flow about a 64A-410 airfoil, is shown as Figure 4. Six different lattice regions were used, which were represented by approximately 3700 points with about 45 points on the upper surface of the airfoil.

Details of the notation used in the mesh arrangement are given in Appendix A, however, general information will be given in this paragraph. Each mesh region consists of rows and columns with the first letter of the variable name denoted by N and M respectively. The second letter in column-row descriptor denotes the region, given by the following:

Region	Identifier	Mesh Description
1	Y	Outer polar
2	C	Coarse cartesian
3	F	Fine cartesian
4	H	Airfoil cartesian
5	M	Outer nose
6	T	Inner nose

For the three-letter descriptors, the third letter either indicates limits or names important columns or rows. For an example, looking at region 6 (inner nose mesh), the rows are bounded by N = NTA to NTT while the columns are bounded by M = MTA to MTT. These bounding columns/rows contain information obtained by interpolation on assignment from the next larger mesh region. In general, the solution in a mesh region is advanced at all points bounded by the columns/rows one mesh spacing in from the mesh boundary; or for region 6, N = NTB to NTS and M = MTB to MTS. Using b to denote one of the four regions adjoining the airfoil, the mesh column passing through the airfoil nose is given by MbAN and that passing through the airfoil tail by MbAT. The mesh row which passes through the airfoil chord line is given by NbCL.

Local rotated meshes are arranged around airfoil surface normals which pass through regular cartesian mesh points near the airfoil boundary.

3. RESULTS OF CALCULATIONS

The resulting pressure distribution is shown in Fig. 5 where a comparison is made with the experimental results obtained by Stivers (Ref. 4). There is fairly good agreement on the lower side of the profile, but on the upper surface there is a pronounced discrepancy in the nose region, and in the vicinity of the shock. The discrepancies in the nose region and near the shock might both be attributed to viscous effects which are not taken into account in the calculations. The pressure pattern near the shock is due definitely to the appearance of a lambda shock as the result of boundary layer-shock wave interaction; in the leading edge region, the difference is probably due to an appearance of a short bubble separation which prevents the flow from attaining the full leading edge expansion. (It may be recalled that in Stiver's tests the transition was natural with a Reynolds number based on a six inch chord of approximately 10^6).

In Figs. 6a and 6b the resulting constant Mach number lines are plotted, with the latter figure illustrating the nature of the mesh fineness in order to properly resolve details in the nose region. The latter plot has been obtained by an automatic plotter where for simplicity straight lines have been used to connect the interpolated points obtained by linear interpolation of the computed values.

Approximately 830 time planes were required to attain a reasonably steady result for an accumulated expenditure of computer time of approximately 3.5 hours on the CDC 6400 computer.

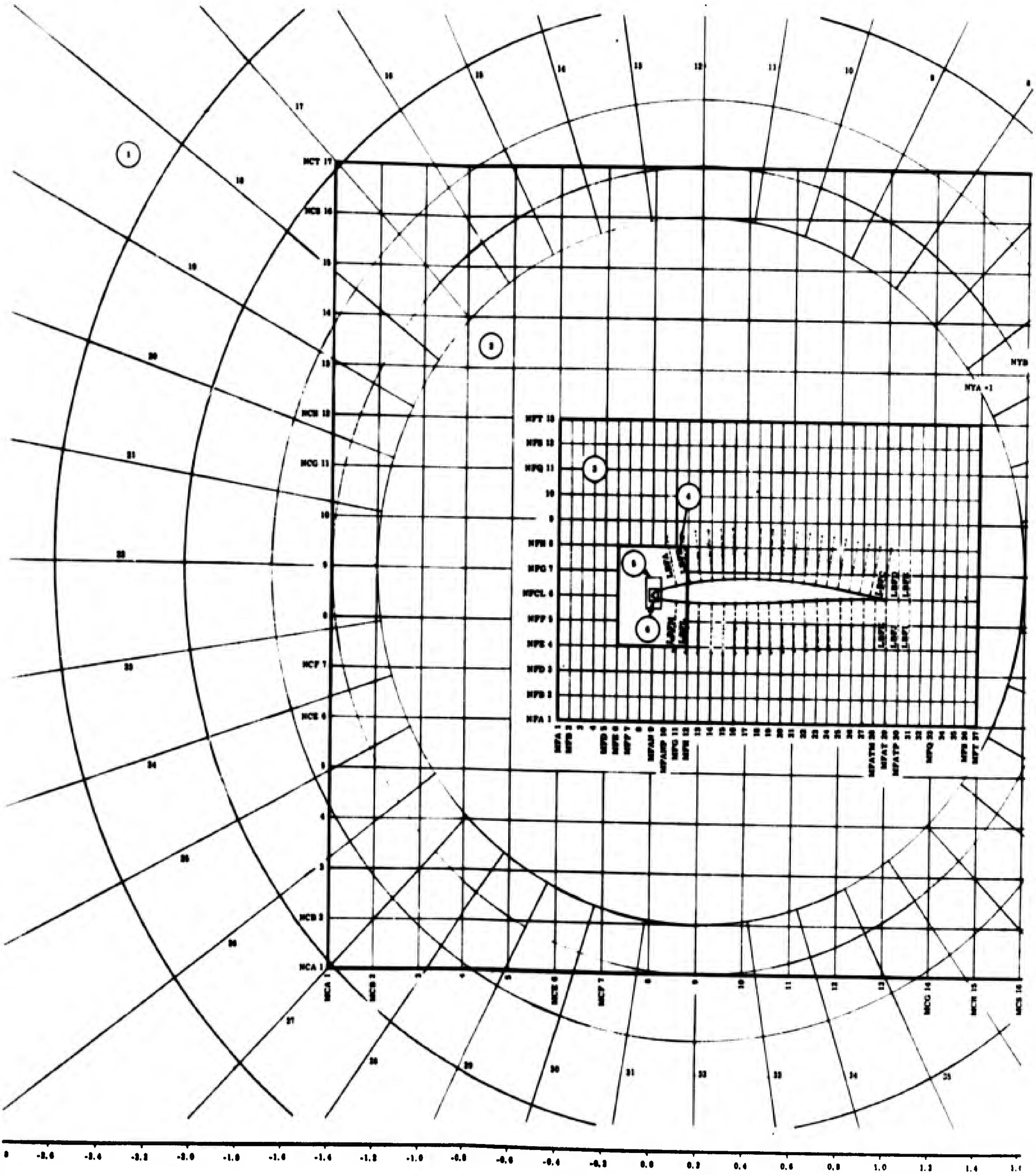


Figure 4. Arrangement o:
Airfoil. (a)

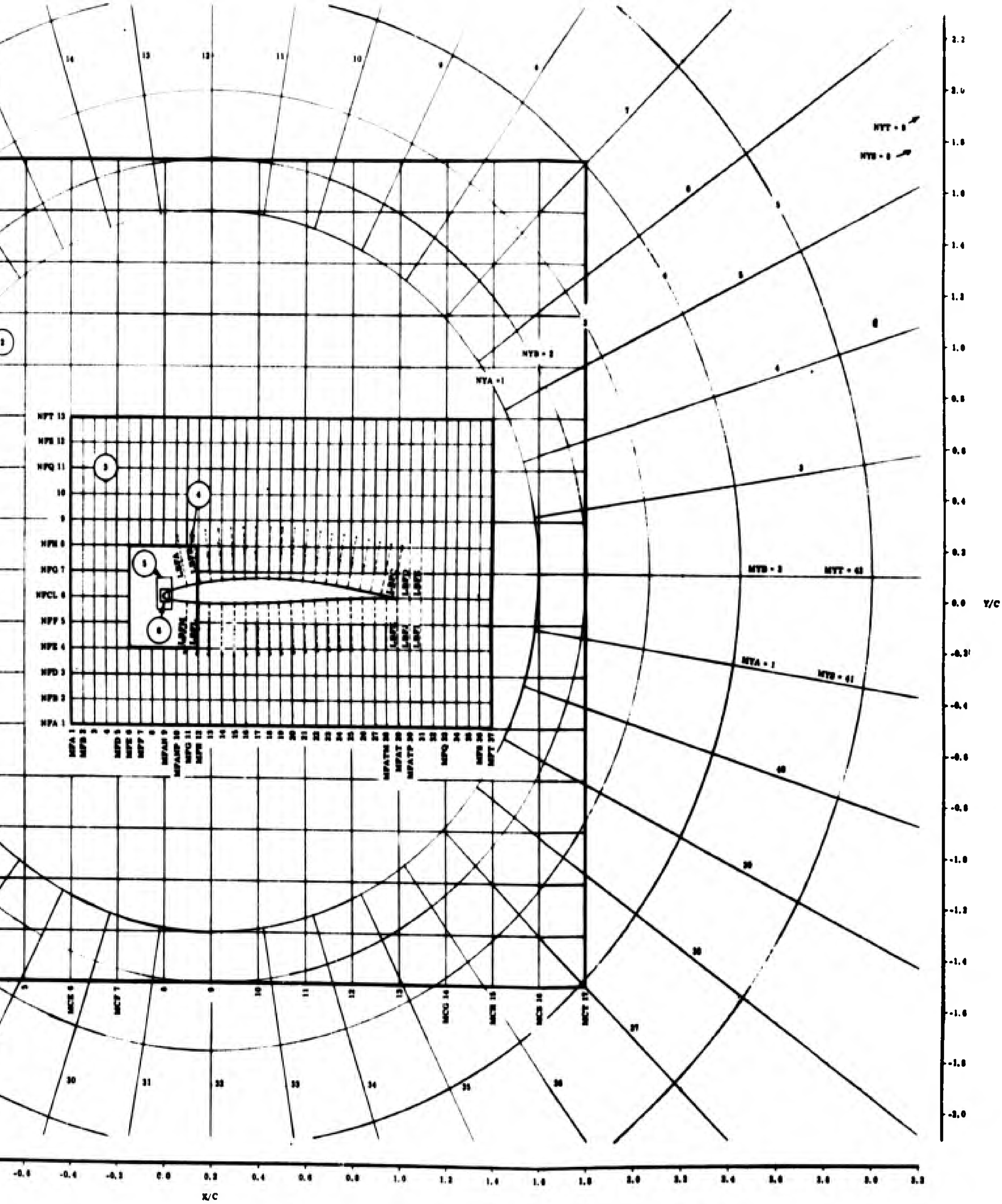


Figure 4. Arrangement of mesh regions around NACA 64A-410 Airfoil. (a) Outer regions.

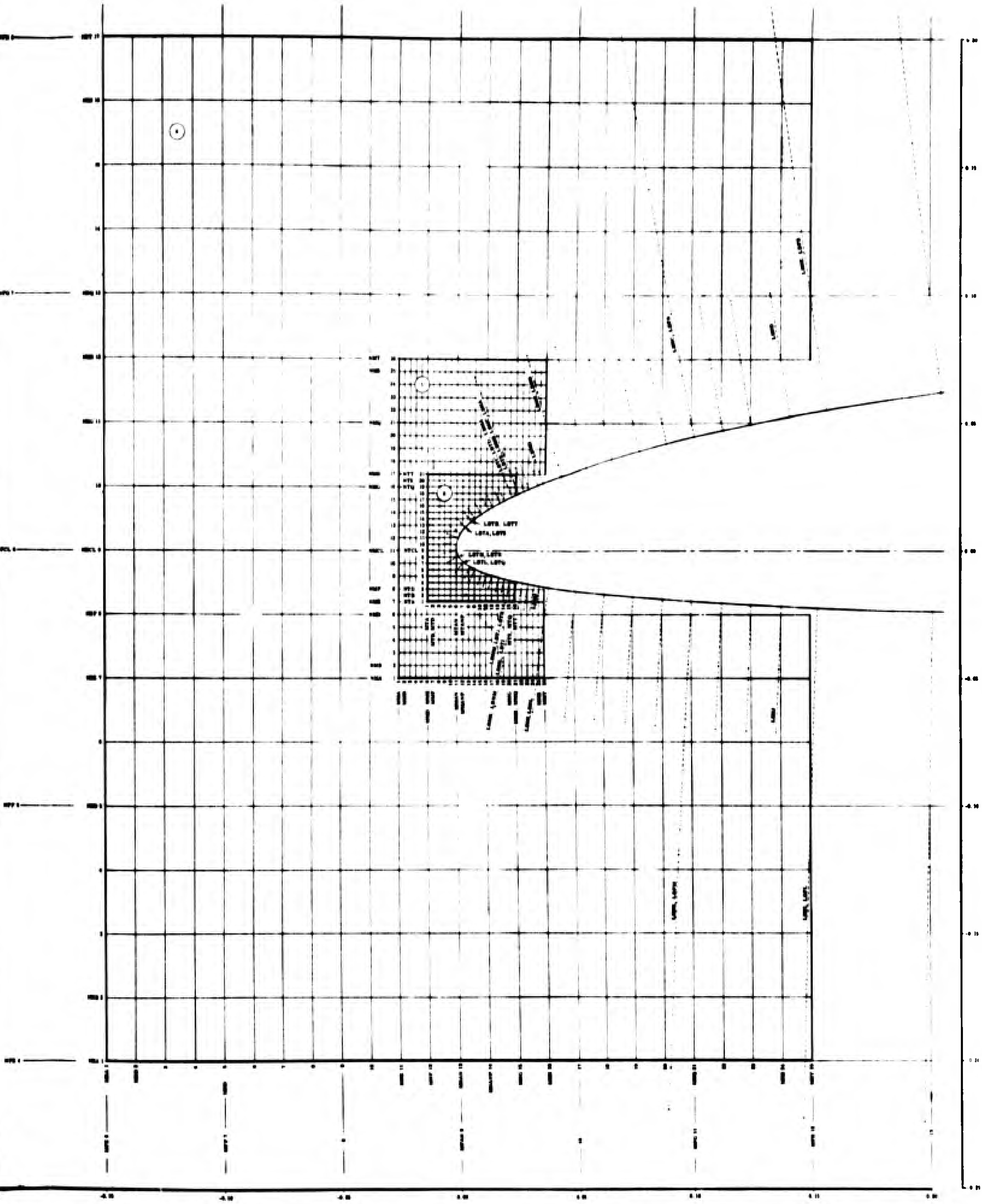


Figure 4 (concluded). Arrangement of mesh regions around NACA 64A-410 Airfoil. (b) Inner regions around nose.

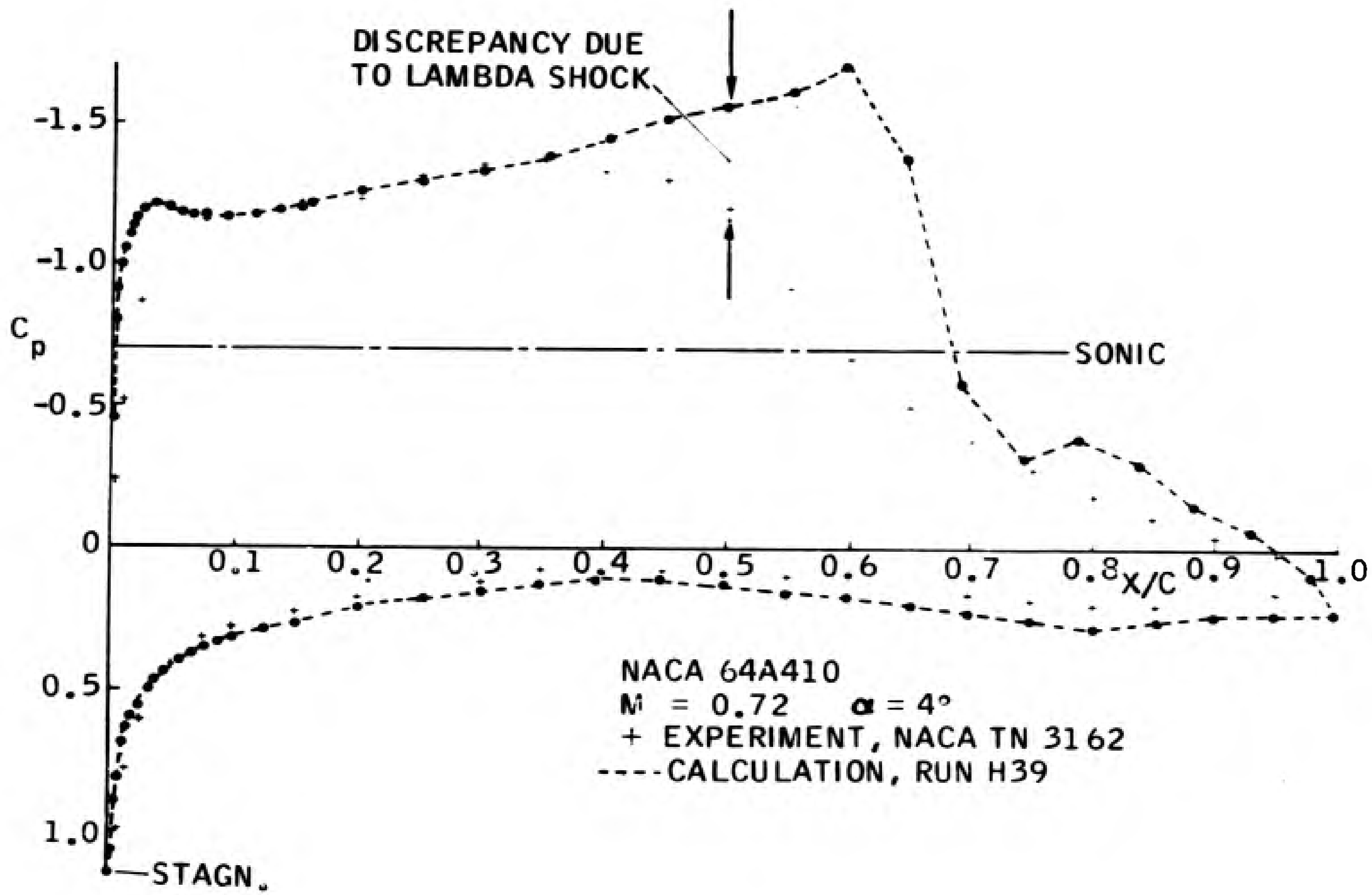


Figure 5. Pressure distribution on NACA 64A-410 Airfoil at $M = 0.72$, $\alpha = 4^{\circ}$.

BLANK PAGE

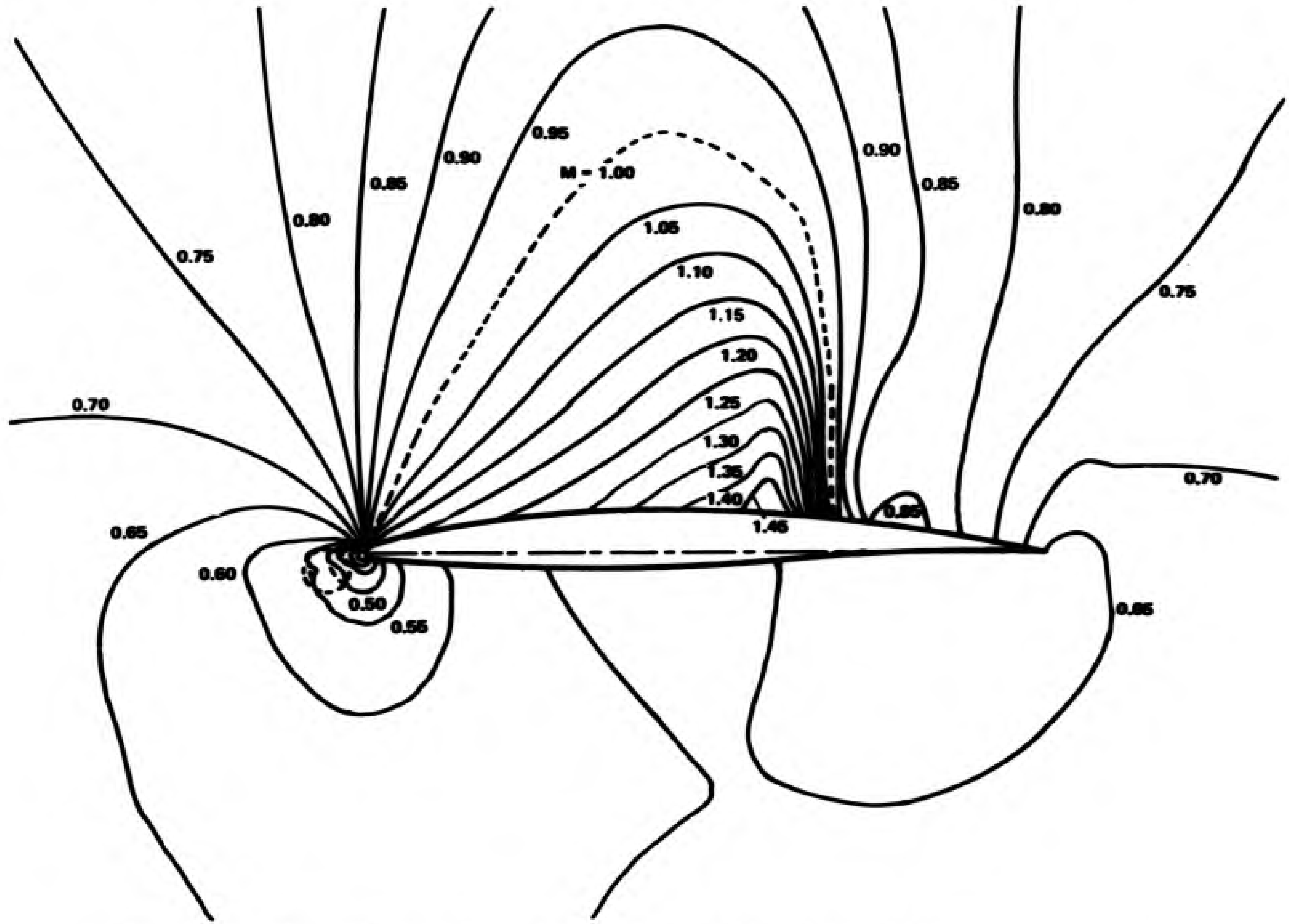


Figure 6. Mach number contours of the flowfield about a NACA 64A-410 Airfoil at $M = 0.72$, $\alpha = 4^{\circ}$. (a) General view.

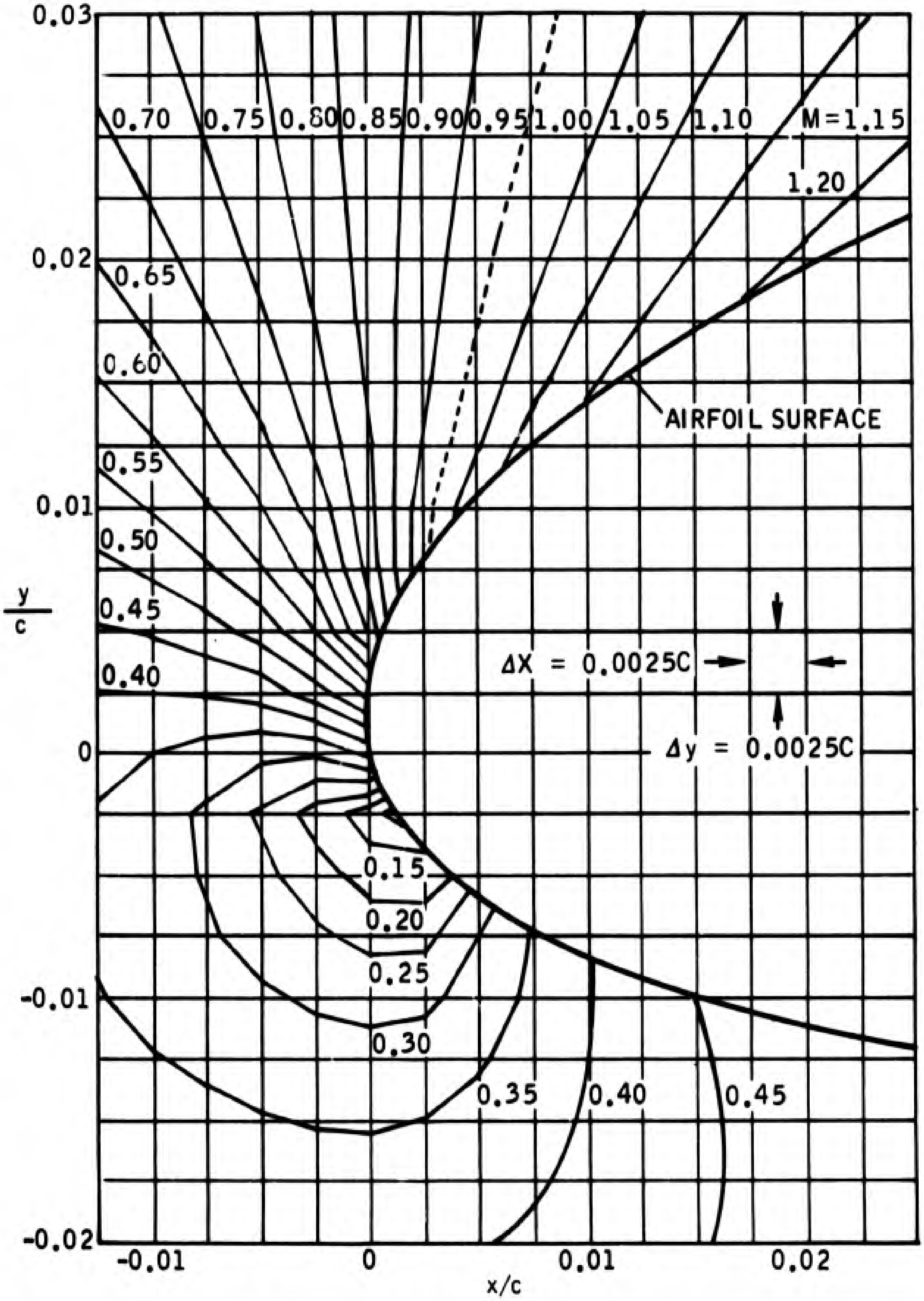


Figure 6 (concluded). Mach number contours of the flowfield about a NACA 64A-410 Airfoil at Mach 0.72, $\alpha = 4^\circ$. (b) Detail around nose.

The calculations were continued for approximately one more hour to 1070 planes with the only significant change being a slight aft movement of the shock closing the supersonic zone. The results presented in Figs. 5 and 6 are those after 1070 planes.

In this single example the progress of the calculations was monitored quite closely by inspection of printed data at 20-plane intervals. The primary effort was directed towards obtaining meaningful results with no attempt made to obtain economical results. Although it is felt that even the above amount of computer time required is not excessive for the nature of the results obtained, serious consideration should be given to methods for shortening the computer time. The large expenditure of computing time can be directly attributed to the large number of lattice points required to resolve the flow adequately, especially in the nose region, and to the slowness of obtaining a steady flow.

SECTION IV

SETUP PROGRAM

The computer program for solving the physical problem was expected to be long, that is it would require a great number of FORTRAN statements, would require a great number of cells to store working variables, and would need to be used for several hundred passes through the field before the desired quasi-stationary solution was obtained. Consequently it was elected to break the program logic into two separate programs; the first (TEHAI) is used once to set up the problem and the second (HANE) is used repeatedly for the several hundred modifications of the description of the flow field.

As a general rule, any extensive operations which need be done only once during the work on the problem would be carried out by the setup program (TEHAI) and results of these operations would be written on magnetic tape for later use by the flow field program (HANE). Since several hours of calculation might be involved, HANE is designed to write intermediate solution results on tape so that printed results may be studied at leisure before continuing the solution.

Thus the setup program (TEHAI) is used to define the problem and prepare initial information for the flow field program which then advances the solution. This program was written in FORTRAN IV type language for the CDC 6400 computer using a SCOPE 3.1 version compiler. The system subroutines used are the normal square root, log, log base 10, sine and cosine and exponential routines.

1. DESCRIPTION OF PROGRAM

The setup program consists of a main routine called TEHAI and the following subroutines; SHAPE, CAXIS, PONDER, BUNTE, DISTW, WAIF and CARTW. Brief descriptions of the routines are given in the following sections. A complete listing of the program and glossary of principal FORTRAN terms are given in the Appendices.

1.1 MAIN ROUTINE TEHAI

This routine assigns the step sizes and operation index limits for each region. It also calculates interpolation constants for interchange of information between the mesh point regions.

Airfoil information is read in as a table of airfoil surface points by subroutine SHAPE which then fits a number of spline functions to these points. The resulting splines are used as interpolation functions to determine the airfoil coordinates and surface slopes at desired locations along the airfoil.

The setup routine TEHAI sets cartesian mesh operation limits depending upon airfoil location and index limit in each region. Next, airfoil surface normals are computed by dropping them from regular mesh points of the cartesian mesh near the surface. These surface normals serve as central axes for 9 point local rotated meshes tangent to the airfoil surface. Then the interpolation weights between the surface normals and local cartesian mesh are computed. Indexes and flags for interpolation and exchange of information between the different mesh regions are also assigned in this routine. The initial values of working variables (Wb and WSb) are then computed and assigned to each mesh point. This routine then writes out all the necessary information in binary form on a magnetic tape.

Most of the selections of reference arrays for interpolations and calculations of interpolation weights are carried out by calling subroutines PONDER, DISTW, CARTW, and WAIF.

1.2 SUBROUTINE SHAPE (NSHP, CLINE)

Subroutine SHAPE reads in arbitrary airfoil information given by a table of surface points and spline fits these points in three regions. The first fit is along the upper surface from the leading edge to the trailing edge. The second fit is along the lower surface from the leading edge to the trailing edge. The third fit covers the blunted nose regions from the lower surface to the upper surface. The last two surface points of the third fit coincide with the first two surface points of the first fit. Surface points 1 and 2 of the third fit coincide with points 2 and and 1 of the second fit. After a spline fit of the airfoil points exists, this subroutine has the options to use the resulting spline fit to obtain airfoil data at arbitrary intersections between mesh coordinates and airfoil surface.

The index NSHP is a flag to specify the path through subroutine SHAPE as follows:

NSHP = 1	Read the airfoil surface points and fit this information with a spline function. Print out the resulting fitted data.

NSHP = 2	Use the spline fit to find airfoil information along a vertical cut, getting both upper and lower surface.
NSHP = 3	Find airfoil information near leading edge using a horizontal cut.
NSHP = 4	Find airfoil information along lower surface using a vertical cut.
NSHP = 5	Same as 4 but along upper surface.

The variable CLINE is the value of x or y along which airfoil surface information is desired.

1.3 SUBROUTINE CAXIS (KR, MH, MS, MAN, NCL, LCU, LCL, IA, DA, SA, CA, LA, XY, SC, DS, LF)

This subroutine determines the geometry of the airfoil local surface normal. Index KR defines the mesh point region, MH and MS define the range of index M in region KR over which the local surface normals are desired. MAN is a reference value of index M which indicates the relative location of the leading edge. NCL is a reference value of index N and indicates the location of the airfoil chordline. Arguments LCU and LCL are used for synchronization of the index numbers for surface normals (on the upper and lower surfaces respectively) with cartesian column indexes M.

The index IA (1,M) indicates the last regular cartesian mesh intersection below the airfoil surface in column M while DA (1,M) is the distance of the intersection from the airfoil surface in fractions of Δy while SA and CA are the sine and cosine of the airfoil surface slope at the intersection of column M with the airfoil. Arguments IA(2,M), etc indicate corresponding quantities associated with the first regular cartesian mesh above the surface and LA indicates one of the dimensions of the IA, DA, SA, and CA arrays.

The arguments above are used in connection with subroutine SHAPE in an iterative procedure to locate the intersection with the airfoil of a surface normal dropped from the regular cartesian mesh mode closest to the airfoil surface. The XY array contains the coordinates of the base of the surface normal, SC the sines and cosines of the clockwise tangent to the airfoil surface at that locus, DS is the distance from the airfoil surface to the generating cartesian mesh node in fractions of unit Δy while LF indicates one of the dimensions of the XY, SC and DS arrays.

1.4 SUBROUTINE PONDER (KSPL, MLL, MUL, XTZ, YTZ, XYT, SCT, DST, DWT, DWTX, AWT, CWT, IWT, LBX, MTT)

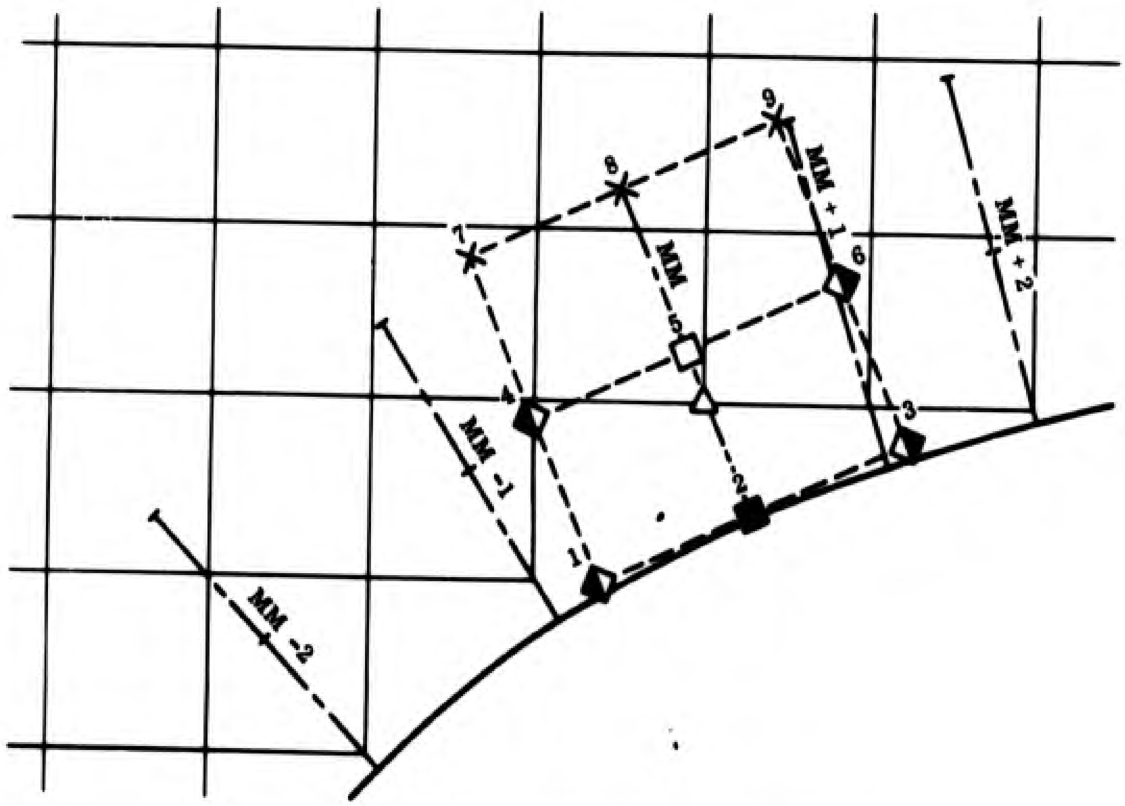
Subroutine PONDER directs the calculation of interpolation weights to be applied in transferring from the distorted mesh to adjacent local rotated mesh, also the weights for interpolating from the airfoil surface normals to those regular cartesian points adjacent to the airfoil surface and weights for interpolating from the cartesian grid to the upper three points of each local rotated mesh. A distorted mesh consists of 9 points on three adjacent surface normals. Subroutine DISTW is called on for the calculation of the weights for interpolating out of distorted meshes while CARTW is used for calculating weights for interpolating out of cartesian meshes.

The subroutine is entered with arguments on range of surface normals to be treated MLL to MUL, geometric information on the bases of the surface normals XYT, DST and SCT, and other arguments and returns interpolation weights.

Argument KSPL identifies occasions calling for special interpolation weights DWTX at extremes of the range of surface normals to be covered. Referring to Fig. 7, while attention is directed to surface normal MM, the 9 weights (DWT) for interpolating from the distorted mesh centered on MM to points 3 and 6 of the local rotated mesh centered on normal MM-1 and on to points 1 and 4 of the local rotated mesh centered on normal MM + 1 are calculated. At the left end of the range, however, there is no choice but to calculate the 9 weights (DWTX) for interpolating onto points 1 and 4 of the local rotated mesh centered on normal MM by use of the information available (see points B, Fig. 8); similarly, at the right end of the range and weights (DWTX) for interpolating onto points 3 and 6 of the local rotated mesh centered on MM would be calculated. Index KSPL has value 1 at the forward end of the range on the upper surface, 2 at the aft upper surface, 3 at aft lower surface, and 4 at forward lower surface.

The 3 weights (AWT) for interpolating from the 3 points on a surface normal to the cartesian mesh point which occasioned the construction of the surface normal are calculated from a knowledge of the offset DST.

The upper three points (7, 8, 9) of each local rotated mesh are obtained by interpolation from an underlying 3 by 3 regular cartesian mesh centered at a row and column designated in IWT. The 9 interpolation weights are given by CWT. Arguments XTZ and YTZ locate the lower left corner of the full cartesian array covering the region and are used as starting points in a counting procedure for locating the appropriate 3 by 3 groups to be used in



× INTERPOLATED FROM CARTESIAN MESH

⬗ INTERPOLATED FROM DISTORTED MESH COMPOSED OF NORMALS MM -2, MM -1, MM

⬖ INTERPOLATED FROM DISTORTED MESH COMPOSED OF NORMALS MM, MM +1, MM +2

△ INTERPOLATED FROM NORMAL MM ONTO CARTESIAN MESH

□ CALCULATED USING SYMMETRIC DIFFERENCE SCHEME

■ CALCULATED USING ASYMMETRIC DIFFERENCE SCHEME PLUS WALL BOUNDARY CONDITION

Figure 7. Relation of local rotated mesh to Cartesian mesh and distorted mesh.

the CWT interpolations. The indexing information is packed into IWT as a single word (100*M+N) where M and N are the column and row of the central point of the 3 by 3 group to be used in the interpolation. Argument MTT is the index of the last right hand column of the underlying cartesian region, this information is used in biasing choice of IWT to prevent interpolations CWT from using data from column MTT before that data is available.

Argument LBX sets dimensions on a number of the arrays used in the subroutine.

Subroutine PONDER is also called by TEHAI to calculate interpolation weights for interpolating from a 9 point distorted mesh (bridging a boundary of a finer imbedded cartesian mesh) onto some of the finer regular cartesian mesh boundary points; the weights are DWFH, DWHM, or DWMT depending on which pair of contiguous regions are involved (see points F of Fig. 8).

1.5 SUBROUTINE BUNTE (NE, NH, IFO, JFO, ID)

Subroutine BUNTE sets codes for the types of interpolations to be carried out in assigning data from a coarser cartesian region to points on the right hand column of an imbedded finer cartesian mesh region. Argument NE is the lower limit of the N index of the coarser region while NH-1 is the upper index. JFO defines the type of cell at N index location IFO and are used by the routine to detect interception of the column by the airfoil. The index ID returns the coded interpolation instructions at field boundaries as follows:

ID = 1	The mesh point is inert, no interpolation required.
ID = 2	Cubic interpolation should be carried out using reference information from 4 points straddling the point in question.
ID = 3	Parabolic interpolation should be used with 2 references points below and one reference point above the point in question.
ID = 4	Parabolic interpolation should be carried out using two reference points above and one below the point in question.

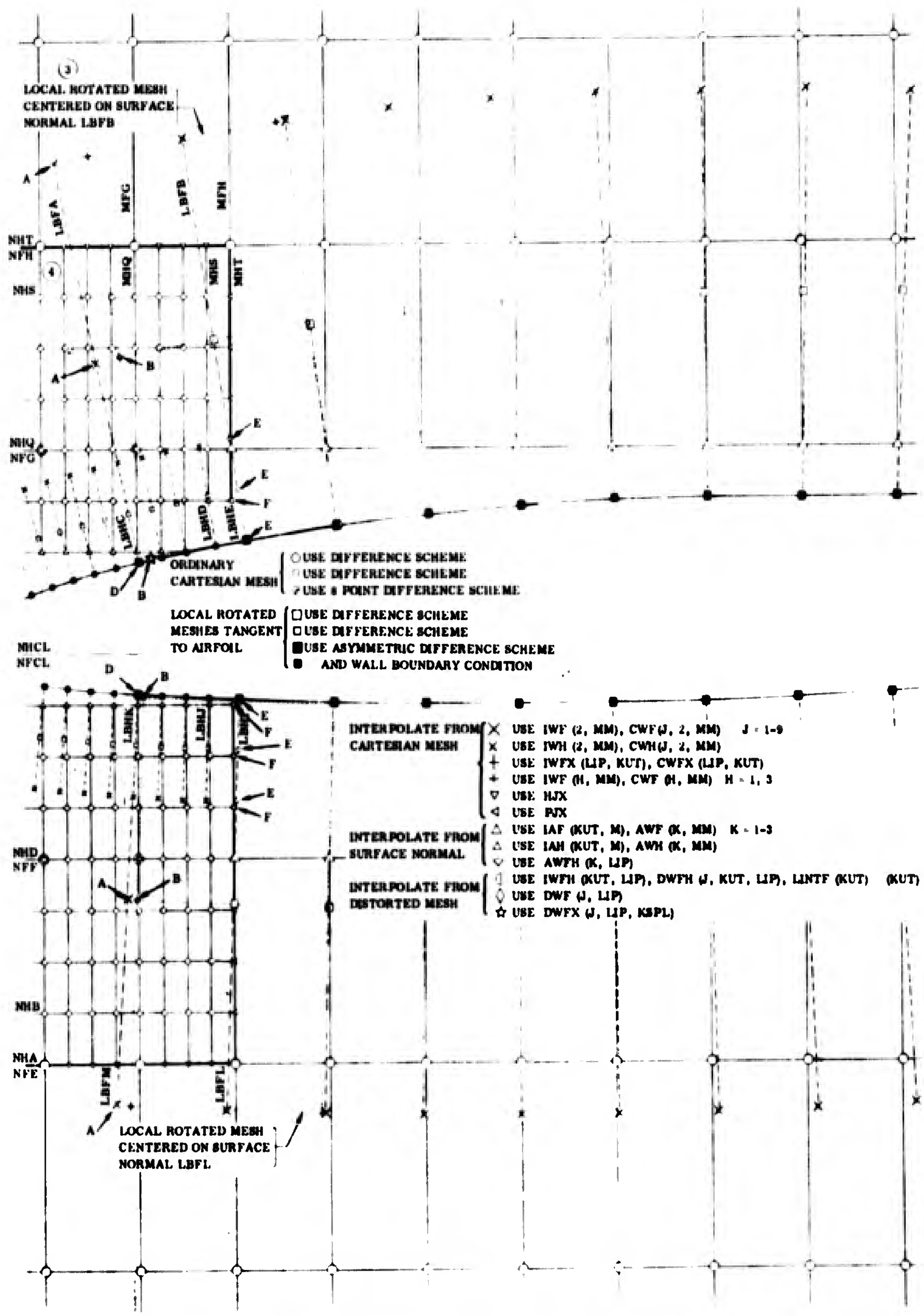


Figure 8. Special features of mesh interrelations where the airfoil penetrates a mesh size change.

1.6 SUBROUTINE DISTW (MD, XP, YP, TW, XYU, SCU, LDIW)

Subroutine DISTW calculates weights TW to be used on values of a function (f) at nine points in a distorted array in order to find interpolated values of F at a specific cartesian point (XP, YP). This subroutine is called first with MD = 1 which digests geometric information on the distorted array (XYU, SCU) consisting of 9 points on 3 adjacent surface normals. The function (f) will be assumed to have polynomial dependence on x, x^2, y, y^2, xy, x^2y, xy^2 and x^2y^2. Then setting MD = 2 will result in use of the basic fitted geometric information to calculate the weights (TW) to be applied to the values of f at the nine reference points in order to interpolate onto a particular point XP, YP. Argument LDIW sets one of the dimensions of the double subscripted variables XYU and SCU.

1.7 SUBROUTINE WAIF (KR, KUT, M, XTZ, YTZ, XYM, SCM, CWMX, IWMX, LBX)

Subroutine WAIF calculates interpolation weights for special auxiliary interpolations from the cartesian mesh to the upper 2 points of those surface normals which are part of a coarser mesh system but overlap and reside in a finer mesh region; see points marked A in Figure 8. The base point of the surface normal, being coincident with the base point of a surface normal of the finer mesh system is obtained by direct assignment (see points D in Fig. 8).

Arguments used in this subroutine are quite similar to those used in subroutine PONDER. Argument KR is the region containing the surface normal, M the index number of the surface normal and XYM and SCM are geometric data on the locus and attitude of the surface normal. When KUT = 1 we are dealing with an airfoil upper surface situation and KUT = 2 signifies a lower surface situation; bias is introduced into selection of the 3 by 3 cartesian array to be used in the interpolations in order to avoid use of cartesian mesh data which is not being properly handled because of too close proximity to the airfoil. The coordinates of the lower left corner of the cartesian mesh region are given by XTZ and YTZ and LBX is one of the dimensions of the arrays used by the routine. The column and row names of central points of the selected 3 by 3 cartesian reference arrays are packed into IWMX and the 9 weighting functions are CWMX.

1.8 SUBROUTINE CARTW (AP, BP, TW)

Subroutine CARTW calculates weights to be used in the interpolation from a 3 by 3 cartesian mesh. The TW are the weights based upon fractions AP and BP, the offsets from the central point of the nine point cartesian array. This subroutine assumes that the functions to be interpolated are fitted by a polynomial containing x, x^2, y, y^2, and xy terms only.

2. INPUT DATA

Card No.	Format	Cols.	Name	Definition
1	8A10	1-80	TITLE	Title card for particular airfoil
2	E10.3	1-10	CVRT	Convergence tolerance on X or Y on iterative determinations within subroutine SHAPE
	E10.3	11-20	EPSLN	Convergence tolerance on fitting of spline functions in subroutine SHAPE
3	3EID.3	1-30	NTE(J)	Number of table entries in airfoil region J.
----- FOR J = 1,2; REPEAT NTE (J) TIMES - - - - -				
4	E10.3	1-10	X	Airfoil x coordinate
	E10.3	11-20	y	y coordinate at the corresponding x.
----- FOR J = 3, REPEAT NTE (J) TIMES				
	E10.3	1-10	Y	Airfoil y coordinate
	E10.3	11-20	X	corresponding x coordinate

3. OUTPUT DATA

The setup program writes a binary tape and presently it is requested at 800 BPI. This program is designed to printout a significant amount of the information placed on the output tape so the programmer and engineer can check the results.

4. OPERATING INSTRUCTIONS

Operation of the setup program requires a significant amount of effort by the engineer in defining the problem. The engineer must select the number and location of mesh points that will best satisfy requirements on accuracy in calculating the flow field about the particular airfoil under investigation. Much of this grid sizing must be physically placed in the computer program in specific statements and array dimensions. The cartesian mesh operational constants and limits must be placed in the program: i.e., type of mesh cell and N index range for this type of cell. In general, the interpolation instructions and weights are calculated by the setup program. After the setup program has set the problem up by writing a magnetic tape acceptable to the flow field program, the engineer must look at the printed out information and check the field mesh setup and operational limits.

It has seemed worthwhile in the preliminary work carried out here to make fairly detailed drawings of the airfoil and the mesh systems to be utilized in the problem. The information printed out by the setup program could then be checked for reasonableness by reference to the drawings. Some of the logic in the setup program may be quite specialized, that is, accurate only on the specific problems undertaken and could be inadequate on new problems having different features.

In particular it has been found desirable to check a number of points as described below:

a. Airfoil Spline Fit. Examination should be made for smoothness of the printed data on second and third derivatives of spline functions fitted to the airfoil input data. Tabulated airfoil coordinates are not always free of typographical errors and also keypunch errors in the input cards might lead to embarrassingly inaccurate or non-smooth descriptions of the airfoil surface and later to flow fields containing waves.

b. Interpolated Airfoil Surface Points. The printed information on mesh column and row numbers for those cartesian mesh nodes immediately exterior to the airfoil surface should be checked by comparison with the drawings.

c. Operational Limits. The operational limits (say IFO and JFO; see Appendix A) for each column of each region should be checked to assure that the damping and cell type coding chosen is appropriate for the configuration of regular cartesian mesh points available at each location.

d. Surface Normals. The listing of surface normals should be inspected to assure that they are in proper sequence; overlap properly and coincide with surface normals of coarser or finer mesh regions at region boundaries; and that the orientation is proper for the tracing of clockwise tangents to the airfoil surface.

e. Special Interpolation Codes at Rightmost Fine Mesh Columns. The codes (IDFH say) for interpolation from coarser mesh to points on the right hand boundary column of a finer imbedded mesh should be checked.

f. Interpolation Addresses and Weights for Interchanges Between Local Rotated Meshes and Cartesian Mesh. All interpolation weights should be scanned to see that the sum of the weights is unity and that no single weight exceeds unity by very much without a valid explanation. Addresses (M and N) for 3 by 3 cartesian mesh arrays to be used in interpolating onto local rotated mesh points should be inspected to assure that legitimate data will be available when the interpolation is carried out. Particular care should be taken to inspect those special interpolation operations which are carried out where the airfoil crosses an interface between coarser and finer imbedded cartesian mesh systems.

The following detailed instructions apply to the CDC-6400 computer located at the Convair Division of General Dynamics.

4.1 RUN REQUEST

The run request information will in general contain the following information.

a. Name of programmer, charge number, etc.

b. Type of system; Scope 3.1.

c. Identifying name for output tape.

d. Time estimate and core size request. This run normally takes less than one minute of CP time with 140000_8 cells for loading and 131100_8 cells required for execution.

4.2 CONTROL CARDS

a. A REQUEST card is required for writing the binary tape output.

b. REWIND card for the requested tape.

c. SET(0) card.

d. INPUT (LC, 70000) calling for a maximum line limit of $70{,}000_8$.

e. UNLOAD card for requested tape.

f. EXIT and UNLOAD card for possible error caused termination of the run.

The control cards are separated from the binary deck by a 7/8/9 card, which also appears before the data cards. The data cards are followed by a 6/7/8/9 card.

SECTION V

FLOW FIELD PROGRAM

The flow field program (HANE) uses initial information from the setup program and advances the flow field solution in time-like steps. This program was written in FORTRAN IV type language for the CDC 6400 computer using a SCOPE 3.1 version compiler. The system subroutines used are the normal square root, log, log base 10, and exponential routines.

In general, it has been attempted to carry out operations which need only be done once in a problem in the setup program (TEHAI) and (because of the method for calculating computing machine costs applied at the facility where the operating program was developed) to tend to weight computational speed more highly than storage conservation when writing program logic.

1. DESCRIPTION OF PROGRAM

This program consists of a main routine called HANE and the following subroutines; REPAIR, OUCH, PUTOUT, AIROUT, YTRE, OOKII, SUSUMU, WOP, WOAP, HERI, JIKAN, ENGS, SAKAI, KAKOI, WAKU, KAKARU, and TREND. Brief descriptions of the routines are given in the following sections. A complete listing of the program and glossary of principal FORTRAN terms are given in the Appendices. Detailed flow charts of all the subroutines are contained in Appendix E.

1.1 MAIN ROUTINE HANE

Routine HANE reads in the necessary card input then reads a binary tape which contains the mesh regions, operating limits, and the working variables. A number of useful constants, weighting factors and damping factors are computed. In the main loop this routine first advances the solution in the outer polar mesh region, then continues working on the cartesian mesh regions down to the smallest mesh size region: coarse cartesian, fine cartesian, airfoil cartesian, outer nose, and ending with the inner nose mesh. After advancing the solution one step, information between mesh regions are interchanged. A test is then made to determine if any of the mesh points should be pushed toward the steady state value of free stream total enthalpy. A test is then made to see if printed output is desired. At the end of a run, an option exists to save useful information and the present solution, to be used in a continuation run after visual inspection of printed data from the present solution.

1.2 SUBROUTINE REPAIR

Subroutine REPAIR has no routine purpose in the program. Rather it is specially written and used to correct any errors which might show up; say in the operational limits generated by the setup program, etc. Alternatively this routine might contain the logic necessary to convert an existing solution to serve as starting conditions for a new problem at different Mach number or angle-of-attack; this routine is called by an input flag (IFIZ).

1.3 SUBROUTINE OUCH (JO)

Subroutine OUCH is used to select the type of printout desired depending upon index JO. This subroutine calls other routines PUTOUT and AIROUT which in turn perform the printout. The type of printout available is:

JO = 1	Printout everything, conservation and physical variables for all regions and the airfoil surface.
JO = 2	Printout physical variables for all regions and the airfoil surface.
JO = 3	Printout physical variables for all regions except for the "outer field" and the "coarse mesh" . All airfoil values are printed.
JO = 4	Print only airfoil surface physical variables.

1.4 SUBROUTINE PUTOUT (J, W, MAX, NAX)

Subroutine PUTOUT is used to printout the flow field conservation variables for J = 1 and the flow field physical variables for J = 2. The argument W represents the array of conservation variables, while MAX and NAX are two of the 3 array dimensions for W in the particular region being worked on.

The conservation variables are: $W1 = \rho u$, $W2 = \rho v$ and $W3 = \rho$. Physical variables which are printed include pressure (p/p_∞), density, u and v velocity components, Mach number, and total enthalpy. The values for the entire rectangular cartesian mesh region are printed regardless of whether or not the point might be inert because of being imbedded within the airfoil.

1.5 SUBROUTINE AIROUT (M1, M2, W, MAX, XY)

Subroutine AIROUT is used to printout physical flow variables at locations where surface normals intersect the airfoil surface.

Arguments M1 and M2 represent the limits on the range of surface normal index numbers to be printed. Argument XY is a double subscripted variable containing the airfoil coordinates with MAX as one of the dimensions. MAX is also a dimension for the array of conservation variables (W). This subroutine prints out the surface coordinates, pressure, density, velocity components tangential and normal to the surface, Mach number, and the total enthalpy.

1.6 SUBROUTINE YTRE

Subroutine YTRE is used to advance the solution in the inverted polar coordinate region which represents the outermost portions of the flow field.

The solution is advanced throughout the field using a 3 column wide moving array which converts 3 columns of WY to F and G, then advances the solution at points along a single (the central) column and places the advanced solution in a temporary storage array WAD. The F, G and WAD data in the 3 column array are then column shifted and a new set of WY from the next higher (M) column number is read in and converted to F and G. Another set of advanced values WAD are calculated and placed in the moving array and the older advanced values (WAD) which had been temporarily held are placed into the main WY array after the 3 column moving array has been shifted (to higher M). In this fashion, it is not necessary to reserve storage for two complete sets of WY, that is the reference values and the advanced values.

The allowable time step to be used at each mesh point is calculated in a relatively rough manner and is thought to be highly underestimated, especially on the rows representing rings of large radius.

The equations to be solved (22) are different from those used in cartesian regions and hence difference approximations to derivatives of F and G with respect to both Rh and θ must be obtained. It should be noted that, while a two-step multilevel difference scheme is employed, the long-term stability and accuracy of the scheme would be degraded if the metrics

$\frac{\partial Rh}{\partial x}$, $\frac{\partial \theta}{\partial x}$, $\frac{\partial Rh}{\partial y}$ and $\frac{\partial \theta}{\partial y}$ were not regarded as constants over the 9-point "footprint" while the advanced solution is being calculated at the central of the 9 points.

Values of the conservation variables at the row representing the circle of infinite radius are left invariant in order to enforce free stream boundary conditions. The routine, also, does not advance the solution at points on the row (N = 1) representing the smallest radius ring since data for this row is obtained later by interpolation from the underlying coarse cartesian mesh region.

1.7 SUBROUTINE OOKII

Subroutine OOKII is used to advance the solution in the coarse cartesian mesh for limits of MCB ≤ M ≤ MCS and NCB ≤ N ≤ NCS. Each mesh point is advanced in time according to the locally allowable limit.

The program is organized to advance the solution using a moving 3 column array which is shifted through the field from lower to higher values of M as has been described previously in the discussion of subroutine YTRE. A feature not in YTRE but incorporated in OOKII and also in subroutine SUSUMU (in a more complicated form) is the use of operational limits,in this case, ICO and JCO. In advancing the solution at the various rows (N) in a particular column (M) it is unnecessary to treat some of the points since they will be given closer attention when calculations are carried out in the imbedded finer mesh. Within subroutine OOKII loop limits on N are set for blocks of points in each column M by ICO. All points within each block are treated in a similar manner given in code by JCO; coded 5 if the solution is to be advanced and 0 if the points are to be regarded as inert.

In writing this routine it was judged (in view of the number of operations involved in switching) not economical to block out inert regions when converting W to F and G or in shifting columns within the moving array. This assumption should be reexamined if the program is rewritten for another machine or for production use.

It will be seen subsequently that subroutine SUSUMU performs essentially the same function as OOKII; it advances the solution within a cartesian mesh region subject to operational instructions like ICO and JCO. Subroutine SUSUMU, however, is written to handle a more complicated variety of cells (not all points are handled with absolutely symmetric difference schemes) because of the airfoil cutting through the mesh, and to incorporate enhanced damping near region boundaries. It was considered worthwhile to write OOKII as a separate subroutine to deal with the simpler coarse cartesian mesh region.

1.8 SUBROUTINE SUSUMU (KR, MRA, MRT, NRA, NRT, W, IRO, JRO, ITSMD)

Subroutine SUSUMU is used to advance the solution in those rectangular cartesian regions which are directly intersected by the airfoil. Argument

KR defines the region number, while MRA, MRT, NRA, NRT define the operating limits for the M and N index respectively. In addition MRT and NRT are dimensions for the array of conservation variables W and arguments IRO and JRO. JRO(KK,M) indicates the type of cell and damping that exists between the N column limits of IRO(KK,M) and IRO(KK+1,M) = 1. JRO is a packed word with the flag for the type of damping (JDR) obtained by JRO/100 and the type of cell (JCT) given by JRO-100 JDR. This subroutine makes use of two other routines WOAP and WOP for computing the half-step values used in the two-step scheme.

The solution is advanced over the rectangular field using a five-column moving array which is swept through the field from left-to-right, that is toward successively larger column index names (M). In the array of principal dependent variables, those locations within and to the right of the moving 5-column array contain the values from the previous time plane while locations to the left of the moving array contain data which have been advanced forward by one time step.

Ordinarily, that is, when the 5 column array is in the central portion of the field the sequence of operations for advancing the solution at field column M is as follows:

a. Variables W for field column M+1 are converted to functions F and G and stored in moving array column J = 4.

b. F and G data generated earlier from W variables of field columns M-1 and M and now resident in columns J = 2 and J = 3 of the moving array are used together with the data from J = 4 and W values in field columns M-1, M, and M+1 to obtain advanced solutions for all points in column M; these new solutions (obtained point-by-point for successively larger values of N in the column) are stored temporarily in an array WAD(K,J,N) where J = 3.

c. Values at WAD(K,1,N) replace old values of W(K,M-2,N); WAD(K,1,N) are replaced by WAD(K,2,N) and WAD(K,2,N) are replaced by WAD(K,3,N); also F(K,2,N) and G(K,2,N) are replaced by F(K,3,N) and G(K,3,N); and F(K,3,N) and G(K,3,N) are replaced by F(K,4,N) and G(K,4,N). After this shifting of temporarily stored primary dependent variables (WAD) and auxiliary functions (F and G) one column to the left in the moving array, field column index M is increased by one and the steps (a) to (c) are repeated.

Near the left and right boundaries of the field the routine described is suitably modified to handle the problem of getting started and to assure that all the advanced data gets stored properly when the field has been completely covered.

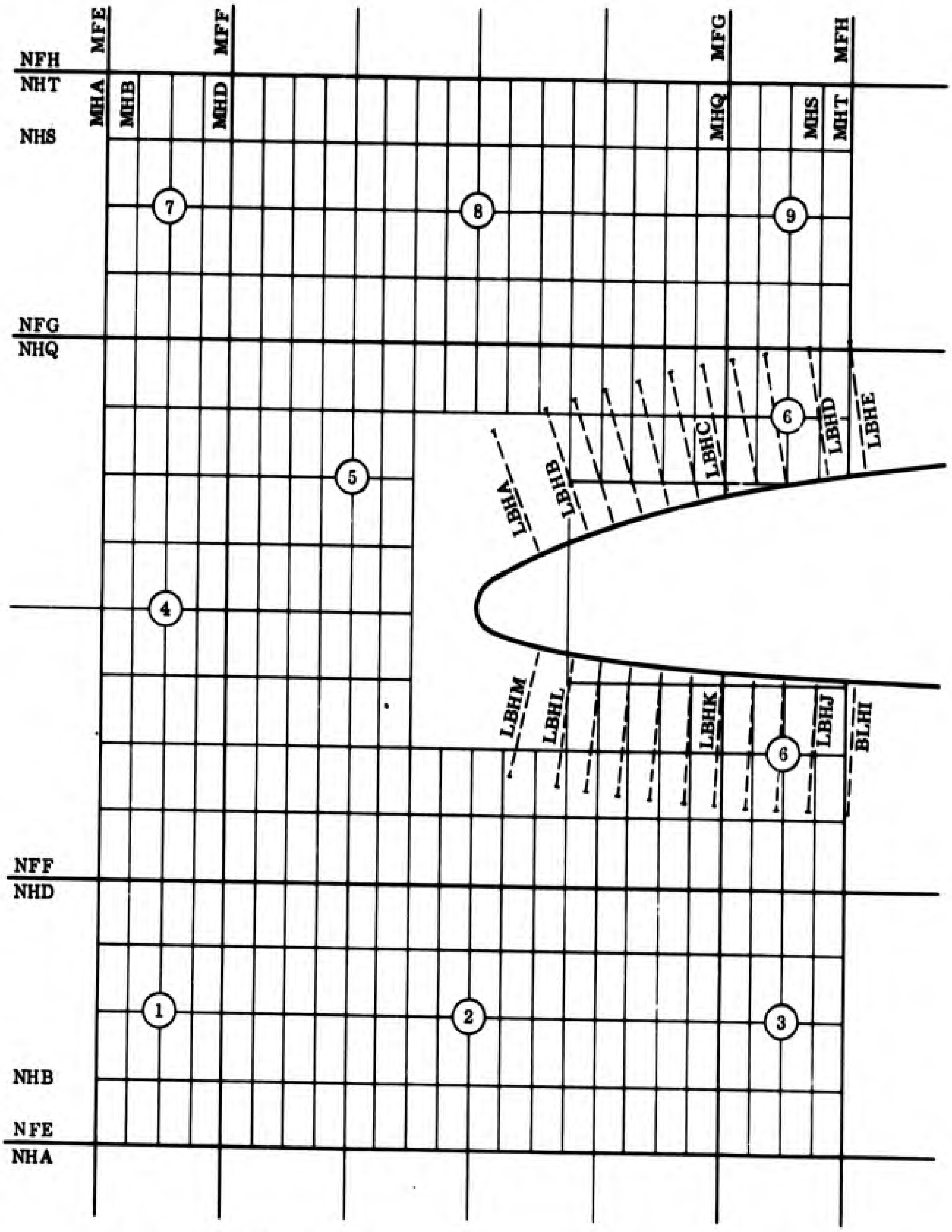
Control over the type of difference scheme to be applied to the individual points (indexed N) in a given column (M) is exercised with operational limits IRO(KK,M) and JRO(KK,M). All cells having N between and including IRO(KK,M) and IRO(KK+1,M)-1 are treated alike in a DO loop. The type of treatment to be given is coded in JRO(KK,M) in two parts:

a. JRO/100 is an index JDR indicating whether or not the normal intentional diffusive damping factor (k) should be increased in an attempt to match the effective damping in a surrounding coarser cartesian mesh; briefly, JDR = 5 requires no augmentation of damping while other JDR = 1,9 require amounts depending on mesh size ratios and the proximity of the point to the field boundaries; see Fig. 9. The amounts of augmentation to be applied are calculated in the main routine HANE and transmitted to SUSUMU in common block PEND.

b. JRO-100 JDR is an index JCT indicating the type of difference scheme to be applied to the available data in advancing the solution. If JCT = 0 the old data is reassigned, that is the point is considered inert. The present program makes use only of JCT = 5, 10, 11, 12, 13 cells. In each of these the solution is advanced at the central point of a 3 by 3 array but for cell types 10-13 the data is considered unuseable at one of the array corners; see Fig. 10. References to cell types 1-4 and 6-9 are vestigial their use having been eliminated during ontogenesis of the program. The logic for advancing the solution of a type 2 cell has been written into subroutine HERI which handles airfoil boundary points.

Argument ITSMD controls the manner of determining the time increment to be used in advancing the solution. If ITSMD = 1 the time step is calculated at each point based upon local conditions; if ITSMD = 2 the time step is determined externally by subroutine JIKAN, transmitted in common block CON1, and the same value is used for all points in the region.

Insofar as feasible, the logic has been arranged to minimize the switching needed to treat inert points and ordinary field points and more complicated switching paths may occur in treating extraordinary cells or cells near field boundaries.



5 ~ NO AUGMENTATION 4,6 MULTIPLY BY DMBX
2,8 MULTIPLY BY DMBY 1,3,7,9 MULTIPLY BY LARGER OF DMBX, DMBY

Figure 9. Typical arrangement of damping augmentation regions.

1.9 SUBROUTINE WOP (I, W, M, N)

Subroutine WOP is presently used in the half step portion of the overall scheme to advance the solution for mesh points that are surrounded by seven good reference points instead of eight. The conservation variable W is dimensioned W(3,M,N). Argument I indicates the type of cell:

I	JCT	IPX	IPY
1	10	1	1
2	11	2	3
3	12	3	2
4	13	4	4

and Figure 10 shows the mesh point distribution for the various types of cells using this subroutine.

Because there are only 8 points rather than the 9 used in the completely symmetric scheme (13,14) functions are assumed to be given by polynomials having x, x^2, y, y^2, xy, x^2y and xy^2 terms. The absence of an x^2y^2 term does not affect the order of accuracy of the difference scheme, rather it merely changes the numerical coefficients of the various discretization error terms.

1.10 SUBROUTINE WOAP (IIA, JJA, W, M, N)

Subroutine WOAP is used to calculate half-step values of the conservation variables and corresponding values of F and G. Referring to Fig. 10, subroutine WOAP calculates the half-step value at mesh point (IIA,JJA); therefore one must call this routine for each half step value required. Arguments M and N are the dimensions for conservation variable W.

Subroutine WOAP is relatively little used in the present version of the program, being called only for half-step values while treating types 10-13 cells.

1.11 SUBROUTINE HERI (KR, LBX, W, ITSMD)

Subroutine HERI advances the solution at airfoil surface points and at points Δy outboard along a normal from the surface using a locally rotated

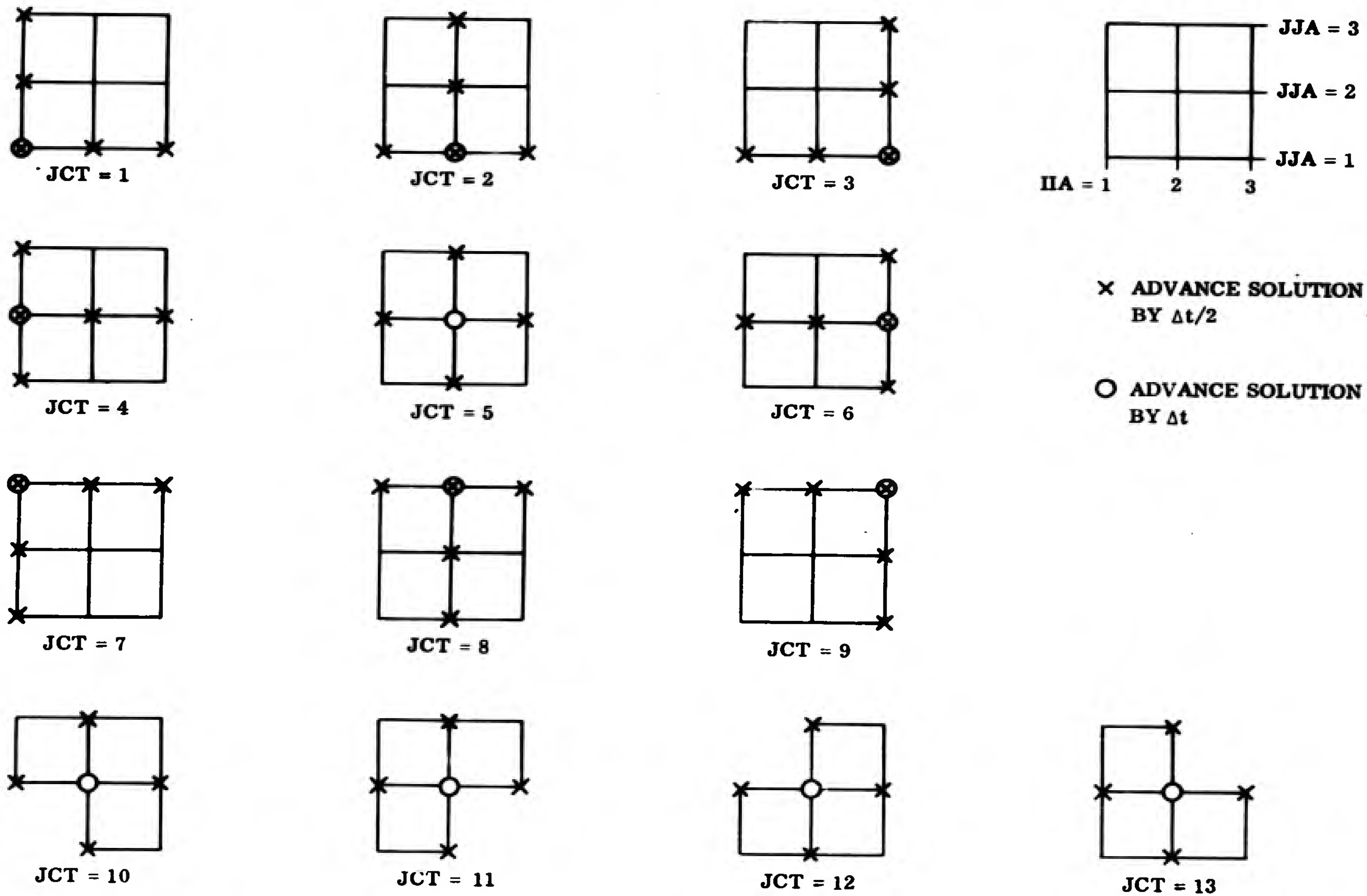


Figure 10. Cell types used in computer program.

3 by 3 cartesian mesh which is tangent to the airfoil surface. This routine assumes the wall point is the critical value for the allowable time step calculation. After advancing the solution, this routine assigns a no-normal-velocity boundary condition for the surface point. Index KR is the region number. Index LBX is the dimension for the local normal cartesian mesh region conservation variables W. Flag ITSMD indicates the type of time step to use:

ITSMD = 1	Use local conditions at the airfoil point to determine the allowable time step.
ITSMD = 2	Use a time step limit computed outside this routine by subroutine JIKAN.

Subroutine HERI sets, depending on the mesh region, loop limits determining index numbers of surface normals along which the solution is to be advanced. Also warnings are set to determine those cells at which the diffusive damping is to be augmented because of proximity of the cells to a region boundary.

In the main loop the values of W at the 9 points in the 3 by 3 local rotated array are converted to F and G functions, the allowable time step is determined and the solution is advanced a half step in time over points 1, 2, 3, 4, 5, 6, 8. Referring to Fig.10 it may be seen that these construction points are necessary to advance the solution a full step at points 2 and 5. The full step solution at point 2 is stored temporarily in an array WA and the wave method for satisfying boundary conditions is applied (Eqs. 19,20) before assigning the advanced solution to the main variable array at point 2. Depending on JTND (a program input supplied through common block ENTHP) subroutine ENGS may be called to adjust the total enthalpy at wall point 2.

1.12 SUBROUTINE JIKAN

Subroutine JIKAN calculates the time step allowable in the inner nose mesh region by sampling all cartesian mesh field points and airfoil surface point and choosing the minimum of the allowable values.

The criterion for allowable time step in a cell amounts to having the backward Mach cone from point 5 at $t_o + \Delta t$ fall within the lozenge 2-6-8-4-2 in the t_o plane. Assuming Mach number $\sqrt{2}$, there might be a factor of (as much as 1.2) larger allowable time step in a square cell having the velocity aligned with the 4-5-6 axis than in a similar cell having the velocity aligned parallel to a 1-5-9 cell diagonal. In the inner nose region where locally rotated meshes can be oriented 45 degrees with respect to the underlying regular cartesian mesh, the policy of using severely mismatched time steps and exchanging of data between rotated and regular meshes by interpolation could possibly cause an instability in the calculation. Hence it was elected to use time steps matched over the entire inner nose region.

In other flow regions the boundary meshes are more closely aligned with the underlying cartesian system so the use of a non-uniform time step was allowed. The need for a uniform time step in the inner nose region has not been tested by experiment.

1.13 SUBROUTINE ENGS (W)

Subroutine ENGS attempts to push the total enthalpy of airfoil surface points towards the steady state value of free stream total enthalpy. The present version of this routine holds the pressure and changes the velocity magnitude unless the local static enthalpy exceeds the desired total enthalpy; in that case the Mach number is retained while all three conservation values are altered.

Argument W is a set of 3 conservation variables at a single wall point. Subroutine ENGS is called subject to the control of program input JTND and reduces the discrepancy between local total enthalpy and free stream total enthalpy to (1-TENB) of the detected discrepancy; TENB is also a program input. Note that it has been assumed that $\gamma = 1.4$ in this subroutine.

1.14 SUBROUTINE SAKAI

Subroutine SAKAI interchanges information between adjoining regions by assignment of common points, interpolation and, in some cases, setting up limits before calling other subroutines to perform the interpolations. This routine assigns the Kutta condition before returning control to the main routine.

The interchanges of data are performed in a particular order with the objective of avoiding or minimizing the number of interpolations or assignments based upon incomplete or obsolete data.

First, points on the inner ring (N=1) of the outer polar coordinate mesh (Region 1) are obtained by interpolation from the coarse cartesian mesh (Region 2). Addresses of the central points of the 3 by 3 cartesian meshes to be used in the interpolations are given by IYB while offsets from the central point (in fractions of Δx and Δy) are given by DYB. The weighting functions are calculated by the program.

Next, data from ray M=41 of the polar mesh assigned to ray M=1 and data from ray M=2 is assigned to ray M=42. The solution is advanced along only 40 rays when the polar field is being treated as a rectangular array and columns 1 and 42 are auxiliaries obtaining advanced data by direct assignment.

The advanced data along the perimeter of the coarse cartesian mesh (Region 2) is obtained by interpolation from the polar mesh (Region 1); the ICB pick out the central points of the 3 by 3 reference arrays and the DCB are the offsets.

The meshes in the various cartesian mesh regions have all been chosen so that an integral number of the smaller meshes equals one larger mesh. Hence, the program assigns advanced values of field variables calculated in a finer mesh to the coincident points of coarser meshes. The values calculated in the inner nose region (Region 6) are assigned to the coincident points of Region 5; then values at points of Region 5 are assigned to coincident points of Region 4 and so on up to the coarsest cartesian mesh (Region 2).

Next, interchange of information between the cartesian mesh (Regions 4-6) and their respective arrays of local rotated meshes aligned on airfoil surface normals is undertaken. Subroutine KAKOI is called and interpolations from the cartesian mesh onto the upper point of each surface normal within the interior of the region. The interpolated data is obtained from a 3 by 3 cartesian array whose address is coded in IWb(2,MM) using interpolation weights CWb(K,2,MM) where b depends on the particular region, MM is the index numberof a specific surface normal and K=1,9 for the 9 reference points. The interpolated data is placed at WSb(KK,8,MM) where KK=1,3 for the 3 principal dependent (conservation variables). Subroutine KAKOI also interpolates, by using weights AWb(K,MM) and reference data from the 3 points of the local surface normal (points 2, 5 and 8 of the local rotated mesh) onto that cartesian mesh point which served to generate the surface normal. The proper cartesian mesh point to receive the data is identified by IAb If the cartesian mesh receiving the data happens to also be a point of the next coarser cartesian mesh, the data is assigned to the coarser mesh also.

In the fine mesh (Region 3), those local rotated meshes aft of the airfoil trailing edge coincide with the underlying cartesian grid and have advanced data assigned directly to them from the cartesian grid.

In both the cartesian mesh systems and their associated collection of local rotated meshes the solution is actually advanced (using a difference scheme) within the interior of the region; advanced data along the margins of the finer regions are obtained by interpolation and assignment from coarser regions. Conversely, if a coarser region surrounds a finer mesh region, the solution is advanced (by a difference scheme) only as far inboard as the perimeter of the finer mesh region using (as auxiliary data) information obtained (mainly by assignment) from points in the interior of the finer mesh region. A part of the functions of SAKAI is to make these data interchanges at region boundaries.

Data at the base points of those coarser mesh surface normals which stand in a finer mesh region and coincide with finermesh surface normals (marked D in Fig. 8) are assigned from the finer mesh surface normals.

Data on those finer mesh surface normals which stand in a coarser mesh and coincide with coarser mesh surface normals (marked E in Fig. 8) are obtained by three point interpolation from the coarser mesh surface normal using interpolation weights AWbc.

Next, data is interpolated and assigned from coarser cartesian mesh to points on the perimeter of finer cartesian mesh. On the left column and the top and bottom rows the interpolations are made using cubics fitted to 4 straddling reference points as data. The weights are listed as data to subroutine SAKAI in arrays GJX, HJX, RJX, SJX; the proper array is chosen for the integral relation between the coarser and finer mesh sizes. On the right hand column boundary the interpolations may also use the weighting arrays mentioned but, since the airfoil cuts through the mesh it may be necessary to make a parabolic interpolation based on three reference points if illegitimate reference data is to be excluded. The choice is controlled by operational codes IDbc and interpolation weights from data arrays PJX, QJX, TJX are used for these parabolic interpolations.

Advanced data for the upper two points on a coarser mesh surface normal which lies in a finer mesh region and coincides (at its base) with a finer mesh surface normal (points marked A in Fig. 8) are obtained by interpolation from the coarser cartesian mesh. Subroutine KAKARU carries out the operations using IWbX for the addresses of the 3 by 3 cartesian reference arrays and CWbX for weighting functions.

The advanced data at the remaining six points of each local rotated mesh (points 1, 4, 7 and 3, 6, 9) are obtained by interpolations using subroutine WAKU. For a given surface normal MM the data at points 7 and 9 are obtained from the underlying cartesian mesh using IWb(J,MM) J=1 and 3 to locate the reference 3 by 3 cartesian mesh and CWb for the interpolation weights. The data at points 1 and 4 is obtained by interpolation out of the distorted array consisting of surface normals MM-2, MM-1 and MM using interpolation weights DWb(K,J,MM-1); K=1,9; J=1,2. Similarly, data at points 3 and 6 is obtained by interpolation out of the distorted array consisting of surface normals MM, MM+1, and MM+2 using interpolation weights DWb(K,J,MM+1); K=1,9; J=3,4. At the ends of the ranges of surface normals (where normals MM-2 or MM+2 would not be available) the data at points 1, 4 or 3,6 would be obtained by interpolation from the distorted array consisting of surface normals M-1, M, and M+1 using weighting functiond DWbX.

When the attention of subroutine WAKU is directed to a coarser mesh surface normal which crosses the perimeter of a finer cartesian mesh, interpolations are made from the distorted array onto those finer cartesian mesh boundary points lying between the crossing of the surface normal by the finer mesh perimeter and the airfoil surface (points marked F on Fig. 8). Interpolation weights DWbc are used; IWbc identifies the finer cartesian mesh points which are to receive the interpolated data; and LINTb fixes the number of such interpolations to be carried out.

The final operations performed in SAKAI are for enforcement of the KUTTA condition. The density at the cartesian mesh point lying at the trailing edge has been taken as the average of the densities at the upper and lower surface points nearest the trailing edge. The horizontal mass flow (ρu) at the cartesian mesh point has been taken to be the average of the surface mass flows at the upper and lower surface points nearest the trailing edge. The vertical mass flow (ρv) at the cartesian mesh point representing the trailing edge has been set at zero.

1.15 SUBROUTINE KAKOI (WT,MTT,NTT,WM,MMT,NMT,WST,IWT,CWT,AWT,SCT,LBX,IAT,MNX)

Subroutine KAKOI interpolates from the cartesian mesh grid onto the top point of the airfoil local surface normal. In addition, an interpolation is made from the local surface normal to the cartesian mesh point on the surface normal and nearest the airfoil surface. If this cartesian mesh point coincides with a point of the next coarser cartesian mesh the interpolated values are assigned to the coarser mesh point too.

The array of principal dependent variables in the cartesian mesh region being worked on is WT(3,MTT,NTT) and of the next coarser cartesian region is WM(3,MMT,NMT). The array of dependent variables in local rotated meshes associated with surface normals is WST(3,9,LBX) and SCT(2,LBX) gives information on the orientation of the local rotated mesh with respect to the underlaying cartesian mesh. The array IWT(3,LBX) is used to locate the individual

3 by 3 cartesian meshes from which interpolations are to be made onto the uppermost points of surface normals using interpolation weights CWT(9,3,LBX). The interpolations from surface normals onto the cartesian mesh points nearest the surface, identified by IAT(2,MNX), are made using weights AWT(3,LBX).

Note that the principal dependent variables for cartesian mesh points are ρu, ρv, and ρ where u and v are cartesian mesh velocity components. In local rotated meshes the principal dependent variables are ρu_t, ρu_n, and ρ; that is velocities are given in directions parallel and normal to the airfoil surface. Consequently, in interchanges of information between cartesian and local rotated meshes the mass flow vector W_1, W_2 must be properly resolved by use of the orientation information SCT(1,MM) = sin θ and SCT(2,MM) = cosine θ where θ is the angle between a clockwise tangent to the airfoil surface and the positive cartesian x axis.

1.16 SUBROUTINE WAKU (IWT, CWT, WST, SCT, DWT, DWTX, LBX, WT, MTT, NTT, WU, MUT, NUT, DWTU, IWTU, LINTU)

As has been explained previously in discussion of subroutine SAKAI, subroutine WAKU interpolates from cartesian mesh points to points 7 and 9 of the local rotated meshes. It also interpolates from the distorted mesh consisting of 3 adjacent surface normals onto points 3 and 6 of the adjacent local rotated mesh to the left and points 1 and 4 of the adjacent local rotated mesh to the right. The routine will also interpolate onto points 1 and 4 or 3 and 6 for local rotated meshes at the ends of a mesh region and interpolate onto cartesian mesh points lying on the border of a finer mesh.

The dependent variables in the coarser cartesian mesh are given in array WT(3,MTT,NTT) and in the finer cartesian mesh by WU(3,MUT,NUT). The dependent variables in the local rotated meshes associated with the coarser of the two cartesian mesh regions are in array WST(3,9,LBX) and the orientation information is stored as SCT(2,LBX).

The interpolations from cartesian to local rotated meshes are carried out using locators IWT(3,LBX) and weights CWT(9,3,LBX). Interpolations from distorted meshes to local rotated meshes are carried out using weights DWT(9,4,LBX) and, at the ends of a mesh region, using weights DWTX(9,2,6). To interpolate from distorted mesh onto cartesian mesh border points identified by IWTU(2,4) weights DWTU(9,2,4) are used; LINTU(2) controls the number of such interpolations which should be carried out.

Note that, since the basic data for interpolations from a distorted array is drawn from three adjacent surface normals having different orientations, the mass flow vectors are resolved into the Cartesian components and the data is stored in an array WINT (3,9) before interpolating.

1.17 SUBROUTINE KAKARU (KUT, LU, IWMX, CWMX, WM, MMT, NMT, WSM, SCM, LBX)

Subroutine KAKARU interpolates from a coarser cartesian mesh to the upper two points of those coarser mesh surface normals at the ends of the array of surface normals standing inside a finer cartesian mesh region, see points A, Fig. 8.

The array of dependent variables in the coarser cartesian mesh is WM(3,MMT,NMT) and the array in local rotated meshes is WSM(3,9,LBX). The orientations of the local rotated meshes are given by SCM(2,LBX). Only one surface normal, indexed LU, needs be handled on the upper surface of the airfoil (KUT=1) and one on the lower surface (KUT=2). The appropriate 3 by 3 cartesian reference arrays are identified by IWMX(2,2) and interpolation weights CWMX(9,2,2) are used.

1.18 SUBROUTINE TREND

This routine attempts to push the total enthalpy at all mesh points (cartesian and local rotated meshes) toward steady state free stream total enthalpy. Discrepancies in total enthalpies are reduced to (1.0-TEND) of their calculated values, where TEND is an input quantity. The use of this subroutine is controlled by input quantity JTND and the subroutine is called only when MOD(L,JBCT)=0; JBCT is also a program input.

The total enthalpy is adjusted by preserving the Mach number and adjusting density and velocity magnitudes.

2. INPUT DATA

Card No.	Format	Cols.	Variable	Definition
1	8A10	1-80	TITLE	Title card from the calculation
2	I5	1-5	LZ	Initial plane number for run
	I5	6-10	LMAX	Termination plane number for run
	I5	11-15	LPRINT	Print interval
	I5	16-20	JOUT	Type of printout desired. = 1 printout all W's and physical variables

Card No.	Format	Cols.	Variable	Description
				= 2 printout physical variables for all regions and airfoil
				= 3 printout physical variables for all regions and airfoil except the outer two mesh regions
				= 4 printout only the airfoil physical variables.
2	I5	21-25	JTND	Flag for the type of enthalpy correction desired.
				= 1 no forcing of the field toward the steady state total enthalpy
				= 2 adjust boundary points by amount TENB
				= 3 adjust all field points each computational step using factor TEND
				= 4 adjust all field points based upon interval defined by JBCT. Once again, use factor TEND
	I5	26-30	IBUG	A debugging printout flag
				= normal printout
				= printout results of each computational step for the fine mesh region
2	I5	31-35	INIT	Flag to call for a tape save at end of run
				= 1 do not save information on tape
				= 2 save results of the last computational plane on tape

Card No.	Format	Cols.	Variable	Description
		36-40	IFIZ	Flag to call for subroutine REPAIR = 2 call for REPAIR ≠ 2 do not call REPAIR
		41-45	JBCT	Flag indicates interval for adjusting the discrepancies in total enthalpy
		46-50	JOFB	Flag used to indicate regions to be computed = 1 advance the solution in all regions = 2 advance the solution in all regions except the polar coordinate region = 3 advance the solution in all regions except the polar and coarse cartesian mesh regions
3	8E10.3	1-80	SAFE(I)	Safety factor reducing the time step used for the various regions, presently I = 1,6
4	8E10.3	1-80	DAMP(I)	Diffusive damping factor for the various regions, presently I = 1, 6
5	E20.3	1-10	TEND	Adjust all field points towards steady-state free stream total enthalpy, reducing discrepancies by the amount (1.0-TEND)
		11-10	TENB	Same as TEND, but for the airfoil boundary points.

3. OUTPUT DATA

The flow field program is designed to printout the conservation variables W along columns (index N) in each region or, if preferred, the physical variables along columns will be printed. The physical variables printed are; pressure, density, velocity in the x direction, velocity in the y direction, Mach number, and total enthalpy. At airfoil surface points velocities are given in components tangent and normal to the surface.

4. OPERATING INSTRUCTIONS

Mechanically, control of the operation of the flow field program is quite simple. The programmer must make sure to request the proper continuation tape and update the computational time limits on the card input. Normally, a run consists of from 10 to 60 computational planes depending upon the computer availability and confidence in the existing solution. After a run, the engineer should look at the solution progress. This usually means examining the printed data, plotting the airfoil pressure or Mach number distribution and pressure distribution ahead, above and behind the airfoil. The engineer should also look at the interchange of information at mesh region boundaries. A Mach number contour plot of selected mesh regions should be made to follow progress of the solution. Convair Division of General Dynamics has a contour plotting routine that accepts the magnetic tape output of the flow field program and plot density and Mach number contours. This contour plotting program accounts for the airfoil surface passing through the rectangular mesh cells at odd angles.

Once the engineer is satisfied that the solution is progressing properly, a continuation run is made using information from the last computational plane (stored on magnetic tape) as input. It is a good policy to save the output tapes from flow field program since most errors that occur normally build up in time and after a correction, one may want to back up and repeat the previous one or two runs. The errors that normally occur are incorrect interpolation and operation instructions from the setup program or instabilities due to poor choice of the amount of damping used in the various regions.

The following detailed instructions apply to the CDC-6400 computer located at Convair Division of General Dynamics.

4.1 RUN REQUEST

The run request information will contain the following information:

a. Name of programmer, charge number, etc.

b. Type of system, Scope 3.1.

c. Identifying name for output tape and reel number for input tape.

d. Time estimate and core size request. The time required to run 30 computational planes with the field size discussed in Section 3 is less than 7 minutes and the core required for execution is 65500_8 cells.

4.2 CONTROL CARDS

a. REQUEST cards are required for reading the binary input tape and writing the binary output tape.

b. REWIND cards are required for the requested tapes.

c. SET (0) card to zero out core.

d. INPUT (LC, 100000) calling for a maximum line limit of 100,000 lines.

e. UNLOAD card for the requested tapes.

f. A EXIT and UNLOAD cards for possible error caused termination of the run.

A 7/8/9 card separates the control cards and the binary deck. The card also appears before the data cards. The data cards are followed by a 6/7/8/9 card.

SECTION VI

BOUNDARY LAYER CALCULATION PROGRAM

This program calculates the boundary layer on an airfoil for an arbitrary surface velocity distribution. The program will continue calculations through turbulent boundary layer separation, but one must view these results in the separation region as questionable, since the correlations for the shape factors and entrainment in the separated region are based on existing experimental tests which were not two dimensional. It is expected that as more test data becomes available, modification to this program will enhance the predictions in this separation region.

Thwaites' (see Ref. 5) incompressible expression for the momentum thickness is used for the laminar portion of the boundary layer. Simple integrated expressions of the continuity and momentum equations are used to solve for the turbulent boundary layer.

1. PROGRAM DESCRIPTION

The computer program for calculating boundary layers on an airfoil consists of a main program called TURBL and the following subroutines: LAMINAR, TRANS, SLOPED, VEL, GAUSSEN, SAVEQ, SOLVE, INTER, INTERP, PARFIT, and SIMP. In addition, a number of functions are used: SKINF, HFACT, XINCOM and VISCOS. A brief description of each subroutine and function are given in the following paragraphs. A complete FORTRAN IV listing and SC-4020 plotted flow charts are presented in Appendix VI and Appendix VII.

1.1 PROGRAM TURBL

Program TURBL is the driver program and performs most of the computations. This program reads in the required input, prints this information and then proceeds to calculate the required starting information. If a laminar boundary layer is required, program TURBL will call subroutine LAMINAR which provides starting conditions for the turbulent boundary layer. Program TURBL then proceeds by calculating the turbulent boundary layer quantities using compressible integral continuity and momentum equations. These equations are integrated using a second order Runge-Kutta approximation. Resulting boundary layer properties are printed and then tests are made to determine if separation has occurred. Once separation has occurred (zero skin friction or a shape factor greater than an input value) the program will continue to calculate through the separated region, however one must remember that the existing expressions are not correct for separation. Reattachment is tested for by comparing the shape factor with an inputed reattachment value.

1.2 SUBROUTINE LAMINAR (THETO, XT, DEL30)

Subroutine LAMINAR uses Thwaites' (see Ref. 5) expression for incompressible momentum thickness for the laminar boundary layer. All the input compressible quantities outside the boundary layer are transformed to incompressible values before being used in this subroutine.

The laminar shape factor and skin friction are assumed to be given by Ref. 5. Empirical expressions for transition and separation are used in this routine. After transition to turbulent flow has occurred, an empirical expression due to Truckenbrot (see Ref. 6) is used for the jump in shape factor from laminar to turbulent. Before returning to the main program, this subroutine computes the compressible turbulent boundary layer starting conditions. Definition of arguments:

THETO = Starting value of momentum thickness. Input value in feet, nondimensionized by the airfoil chord in this subroutine. The subroutine returns the starting momentum thickness from the turbulent boundary layer calculation.

XT = Location of transition from laminar to turbulent flow given as a compressible value.

DEL30 = Starting laminar value of mass flow thickness Δ in feet, also returns the turbulent starting value.

1.3 SUBROUTINE TRANS (DELX, THETAO, DEL30, HSTART, XSTART)

This subroutine takes the compressible input values of conditions "outside" the boundary layer and transforms these into incompressible quantities for use in the laminar boundary layer calculation. Definition of arguments:

DELX = Compressible boundary layer step size, input value.

THETAO = Starting compressible momentum thickness, input value which is then transformed to an incompressible value and returned to the calling subroutine.

DEL30 = Starting compressible mass flow thickness Δ with the incompressible value returned to the calling subroutine.

HSTART = Starting compressible shape factor, H, returned as incompressible value.

XSTART = Starting X coordinate for the laminar boundary layer calculation. After transition to turbulent flow is indicated, this variable is returned as the location of transition.

1.4 SUBROUTINE SLOPED (X, H12, H32, DEL1, DEL2, DEL3, DD2DX, DD3DX)

This subroutine calculates the derivatives at location X for use in the second-order Runge-Kutta integration scheme. One has a choice of the incompressible or compressible equations. Normally the compressible expressions (IDE=2) should be used. Definition of arguments:

X = Location of desired derivatives

H12 = Incompressible shape factor

H32 = Mass flow shape factor

DEL1 = Displacement thickness δ^*

DEL2 = Momentum thickness θ

DEL3 = Mass flow thickness Δ

DD2DX = Momentum equation derivative

DD3DX = Mass flow equation derivative.

1.5 SUBROUTINE VEL(X)

This subroutine interpolates the input conditions outside the boundary layer at the desired location x. A subroutine called INTER determines the approximate location, in the input array, of the argument x. A linear or parabolic interpolation is carried out for the desired properties outside the boundary layer, including the velocity derivative with x. The isentropic relationships are then used to calculate pressure, temperature, and Mach number at location x. This subroutine calls PARFIT to determine a parabolic interpolation at location x using the three surrounding points.

1.6 SUBROUTINE GAUSSEN (NPTS, NO, IKIND, ISIGN, X, Y, WORD, Z)

This subroutine will fit a selected degree of polynomial NO (up to 19) to NPTS number of data points given as x and y arrays. For our present application, IKIND ≡ 2. The argument ISIGN = 1 calls for negative exponent, while 2 calls for positive exponent. The argument WORD is an 80-column title for the curve-fitted profile. The argument z represents the resulting coefficients.

1.7 SUBROUTINE SOLVE (A, X, C, NO)

This subroutine is called by subroutine GAUSSEN and solves a set of simultaneous linear equations by Gauss's elimination scheme.

1.8 SUBROUTINE INTER (WNO, WOMEG, JIND, WOW, NW, JI)

This subroutine performs a search of array WOMEG to find the location JI such that WNO is between WOMEG (JI-1) and WOMEG (JI). JIND is set = 1000 if interpolation is necessary or JI if the value of WNO is equal to WOMEG (JI), in which case no interpolation is necessary. Argument WOW indicates the linear fraction between WNO and two bounding values of WOMEG. Argument NW is the length of array WOMEG.

1.9 SUBROUTINE INTERP (WNO, WOMEG, HLAM, NW, ANS, DERV, INTER)

Similar to Subroutine INTER, but will perform a linear (ITER=1) or parabolic (ITER=2) fit to determine the interpolated value (ANS) of array HLAM for location WNO in the WOMEG array. The derivative (DERV) of HLAM with respect to WOMEG at WNO is also computed.

1.10 SUBROUTINE PARFIT (Y, D, COEF, YY, ANS, DANSDY)

This subroutine provides the parabolic coefficients (COEF), interpolated value (ANS) and the derivative (DANSDY) at location YY for a parabolic fit of D as a function of Y.

1.11 SUBROUTINE SIMP (MAX, DELU, SUB, C, ANS)

This subroutine performs a Simpson integration over an even number of intervals MAX with an interval step size of DELU. Array C is the value of the integral with SUB representing the existing value of the integration at the first location of the present interval and ANS is the integrated value at the end of integration.

1.12 FUNCTION SKINF (IS, H12, H32, DEL2, DEL1)

This function performs a calculation of the skin friction and entrainment function. If argument IS=3 the skin friction is set equal to zero, since this represents the separated region. A flag ICF (named common argument) is used to indicate the type of skin friction equation used; ICF=1 calls for the Ludwieg-Tillmann (see Ref. 7) expression while ICF=2 calls for the Felsch (see Ref. 8) modification to the Ludwieg-Tillmann expression which includes a ratio of an assumed separation value of shape factor and the shape factor.

Argument IFE (named common argument) allows one to select one of two entrainment expressions: IFE=1 calls for Head's (see Ref. 9) original expression, while IFE=2 calls for Green's (see Ref. 10) modification attempting to account for compressibility. The other function arguments have been defined in subroutine SLOPED.

1.13 FUNCTION HFACT (ISEP, HS, HSEP)

This function calculates the incompressible shape factor (HFACT) and the compressible shape factor denoted by H in named common XSKIN. Function argument HS represents the mass flow shape factor.

ISEP = 1 calls for Head's (see Ref. 9) incompressible relationship for $H = H(H_1)$.

ISEP = 2,3 calls for a fit of Seddon's (see Ref. 11) basic separation data.

ISEP = 4 represents a fit to Green's (see Ref. 10) reattachment data.

1.14 FUNCTION XINCOM(X)

This function takes the compressible value of X and performs a parabolic interpolation of the array representing the incompressible values of X and returns XINCOM, the corresponding incompressible value.

1.15 FUNCTION VISCOS(T)

This function returns the viscosity of air, VISCOS, using the Sutherland formula based upon temperature T in degrees Rankine.

1.16 SUBROUTINE SAVEQ (INDEX, X, D1, D2, D3, H12, H32, CF)

A subroutine which allows saving of seven variables at up to 100 stations. A summary printing of these variables occurs before job termination.

2. PROGRAM INPUT

Input required for this program is given below.

Card	Variable	Format	Description
1	WORD	8A10	Title card for calculation
2	NO	I5	Degree of polynomial to be fitted to the entire velocity distribution. A maximum of 19 exists. For interpolation of the input velocity distribution set NO = 0.
	IKIND	I5	Set IKIND=2
	ISIGN	I5	Sign on exponents when entire velocity profile is curve fitted; ISIGN=1, negative ISIGN=2, positive.
	IPAR	I5	Set IPAR=1 for a parabolic interpolation of the input velocity distribution. IPAR ≠ 1 calls for a linear interpolation.
3	ILAM	I5	Set ILAM=1 for a starting laminar boundary layer.
	XVIS	5X,E10.3	Linear viscosity ratio constant
	C	E10.3	Airfoil chord or reference length, ft.
4	ISEP	I5	Indicates the state of the starting turbulent boundary layer: ISEP=1 normal turbulent boundary layer ISEP=2 shock wave-boundary layer interaction region. ISEP=3 separated flow region ISEP=4 reattachment region.

Card	Variable	Format	Description
	ISKIN	I5	Flags the starting turbulent boundary layer skin friction relationships with region defined as per ISEP above.
	IFE	I5	Flag for the entrainment function: IFE=1 Head's incompressible IFE=2 Green's compressible
	ICF	I5	Flag for the turbulent skin friction law. ICF=1, use Ludwieg-Tillmann ICF=2, modification to the Ludwieg-Tillmann expression which allows one to assume a separation shape factor.
	HSEP	E10.3	Assumed separation shape factor
	HREAT	E10.3	Assumed reattachment shape factor
5	INCR	I5	Not used in present version
	IBUG	I5	Set IBUG=1 for a printout at each iteration
6	GAM	E10.3	Ratio of specific heats
	XSTART	E10.3	Starting value of X, ft.
	XEND	E10.3	Final desired X, ft.
	DELX	E10.3	Desired step size in the X direction, ft.
	DELK	E20.3	Starting displacement thickness, ft.
	HSTART	E10.3	Starting incompressible shape factor
	THETAO	E10.3	Starting momentum thickness, ft.

Card	Variable	Format	Description
	DEL30	E10.3	Starting mass flow thickness, ft.
7	ERROR	20X, E20.3	Factional error limit for the iteration on mass flow thickness
	ITER	I5	Maximum number of iterations allowed
	IPROF	I5	If IPROF=1, read in pressure coefficient distribution and calculate velocity
	ICOMP	I5	ICOMP=0 incompressible flow
			ICOM =1 compressible flow
	IDE	I5	IDE =1 use incompressible differential equations
			IDE =2 use compressible differential equations
8	XMIF	E10.3	Free stream Mach number
	XTIF	E10.3	Free stream temperature, ^{o}R
	PIF	E10.3	Free stream pressure, psia
9	IU	I5	Number of points to be read describing the conditions outside the boundary layer
If	IPROF = 1 read the following set of cards I=1, IU		
10	DISX(I)	E20.3	X-location of information, ft.
	CP(I)	E10.3	Pressure coefficient
	BETA(I)	E10.3	Local Mach angle outside the boundary, radians
If	IPROF ≠ 1 read the following set of card I=1, IU		
10	DISX(I)	E10.3	X-location of information, ft.

Card	Variable	Format	Description
	UINF(I)	E10.3	Velocity outside the boundary layer, ft/sec.
	PINF(I)	E10.3	Pressure outside boundary layer, psia
	MINF(I)	E10.3	Mach number
	TINF(I)	E10.3	Temperature, ^{o}R
	BETA(I)	E10.3	Local Mach angle, radians

APPENDIX I

DISCUSSION OF PRINCIPAL FORTRAN VARIABLES

A large number of FORTRAN variables are used in both the setup and flow field programs. A considerable number of these variables are associated with defining the proper mesh region and labeling columns/rows in a particular mesh region. Some shorthand notation in labeling these variables will be presented before listing a glossary of terms. On a machine of large capacity and using a language able to accommodate variables with more than 3 subscripts it would have been more convenient to use 4 subscripts rather than coding region number by the letter scheme (see AI.1 for variable names).

A discussion of interpolation weights is presented in Section AI.4.

AI.1 FIELD NOTATION

Presently, six mesh regions are used to represent the flow about the airfoil of interest shown by Figure 4. The notation for these regions are:

Region	Identifier	Description
1	Y	Outer polar
2	C	Coarse cartesian
3	F	Fine cartesian
4	H	Airfoil cartesian,
5	M	Outer nose cartesian
6	T	Inner nose cartesian

AI.2 SHORTHAND NOTATION FOR VARIABLE NAMES

We shall use a lower case letter in a FORTRAN variable name to signify that it substitutes for any of a set of letter code region identifiers. The identifiers denoted by lower case alphabetic characters are given below.

Identifier	Description
a	All six fields, Y, C, F. H, M, T
b	The four fields which adjoin the airfoil surface, F, H, M, T

Identifier	Description
c	Mesh column/row location
	= A for first col/row in mesh region
	= B for second col/row in mesh region
	= D for first interior col/row which is coincident with col/row of outer (coarser) mesh
	= E for first interior col/row coincident with boundary of inner (finer) mesh
	= F for first interior col/row coincident with an interior col/row of inner mesh
	= G for last interior col/row coincident with an interior col/row of inner mesh
	= H for last interior col/row coincident with boundary of inner mesh
	= Q for last col/row coincident with col/row of outer mesh
	= S for next-to-last col/row
	= T for last col/row
	= AN for airfoil nose
	= AT for airfoil tail
	= CL for airfoil chordline
d	Surface normal indicator
	= A for first (auxiliary) surface normal in the mesh (upper surface)
	= B for first working surface normal (upper surface)

= C for last surface normal coincident with surface normal of outer (coarser) mesh (upper surface)

= D for last working surface normal (upper surface)

= E for last (auxiliary) surface normal (upper surface)

= I for first (aux) surface normal (lower surface) (continuing clockwise from upper surface)

= J for first working surface normal (lower surface)

= K for first surface normal coincident with surface normal of outer mesh (lower surface)

= L for last working surface normal (lower surface)

= M for last (aux) surface normal (lower surface)

e All cartesian meshes, C, F, H, M, T

f Cartesian meshes H and M

As an example, let "a" represent the set of identifiers for the six meshes above (i.e., a may be replaced by any of the letters Y, C, F, H, M, or T). Then the variable name Na actually represents six variable names (NY, NC, NF, NH, NM, NT).

AI.3 GLOSSARY OF FORTRAN VARIABLE NAMES

ALPHA Airfoil angle-of-attack

ATSF Constant used in time step calculation ,

$$\sqrt{1 + (DY/DX)^2}$$

ATSFY Constant used in polar coordinate region time step calculation,

$$\sqrt{1 + (PRMSHY)^2}$$

CAb(K,M) Cosine of the airfoil surface slope at column M, K=1 is lower surface, K=2 is upper surface.

CAN(N) Cosine of airfoil surface for horizontal intersection of row N with airfoil surface.

CFTT	$\frac{1}{2}$ allowable time step in polar coordinate region
CJX(3,3)	Constant containing the weighting factors for derivatives of working variables.
DAb(K,M)	Displacement of intersection of column M and airfoil surface from the point identified by IAb(K,M), a fraction of Δy. K=1 is lower surface, K=2 is upper surface.
DAN(N)	Displacement of row N intersection with airfoil surface in the nose region.
DCB(I,K,L)	Displacements of the boundary point of the coarse cartesian mesh from the point in the polar mesh identified by ICB(I,K,L). Given as a fraction of the polar mesh spacing. I=1 is radial spacing, I=2 angular spacing.
DCX(3,6)	Constants used for diffusive damping at field mesh points in the x coordinate direction.
DCY(3,6)	Constants used for diffusive damping at field mesh points in the y coordinate direction.
DHD(3,3)	Constants containing diffusive damping factor DAMP(8)
DMBX	Multiplier for diffusive damping coefficient to be applied near vertical boundaries of finer cartesian mesh regions.
DMBY	Multiplier for diffusive damping coefficient to be applied near horizontal boundaries of finer cartesian mesh regions.
DOGMU	$2/(\gamma-1)$
DX(I)	Mesh size in the x-coordinate direction. I denotes region number per Section A.1.
DY(I)	Mesh size in the y-coordinate direction. I denotes region number per Section A.1.
DYB(I,L)	Displacements of the inner point on ray L of the polar mesh from the point in the cartesian mesh identified by IYB(2,L). Expressed as a fraction of the cartesian mesh spacing. I=1 is x spacing, I=2 is y spacing.

GAM	Ratio of specific heats, γ.
GMU	$(\gamma-1)$
HGMU	$(\gamma-1)/2$
HGPU	$(\gamma+1)/2$
HGPV	$(\gamma-1)/4$
HGPW	$(\gamma-3)/6$
HINF	Free stream total enthalpy, $1/(\gamma-1) + M_\infty^2/2$
IAb(K,M)	Row identifier for the nodes on column M nearest the airfoil surface. K=1 is lower surface, K=2 is upper surface.
IAN(N)	Column identifier for the nodes of horizontal row N nearest the airfoil surface in the nose region.
ICB(I,K,L)	Node identifier for the point in the polar mesh nearest the boundary point in the coarse cartesian mesh. K = 1 left boundary column = 2 right boundary column = 3 bottom boundary row = 4 top boundary row I = 1 is column, I = 2 is row L = point number.
I$\underline{e}$O(J,M)	Operational limit parameter identifying the points along each mesh column where the computing process changes. Up to 8(J=1,8) such points can be prescribed along each column. (M=1, M$\underline{e}$T).

IYB(I,L)	Node identifiers for the point in the coarse cartesian mesh nearest to the inner point on ray L(L=1,MYT) of the polar mesh. I=1 is a column, I=2 is a row.
J_e_O(J,M)	Operational parameter defining how the computing process changes at the points defined by I_e_O(J,M) (J=1,8) (M=1, M_e_T). Information is packed into J_e_O as follows $J_eO = 100 n_1 + n_2$ n_1 denotes the type of damping to be applied. (See Figure 9). n_2 indicates how the point is to be advanced. (See Figure 10).
KR	indicates mesh region
L	Computational plane number
LBbd	Surface normal identifier
LBbMAX	Total number of surface normals in mesh b.
LBIX(KR)	Integer ratio between cartesian Δx mesh size, DX(KR-1)/DX(KR).
LBIY(KR)	Integer ratio between cartesian Δy mesh size, DY(KR-1)/DY(KR).
LINTM(K) LINTH(K)	The number of boundary points in the embedded fine mesh requiring interpolation from the coarse mesh. K=1 is upper surface, K=2 is lower surface.
M_ac_	Column identifier (e.g MTT is the last column in mesh T).
M_b_ AN	Mesh column passing through airfoil nose (ℓ,e.)
M_b_ANP	= M_b_AN + 1
M_b_AT	Mesh column passing through airfoil tail (t.e)

MbATP	= MbAT + 1
MbATM	= MbAT - 1
MSL,MST	Column identifiers bounding the portion of the inner nose mesh region where the airfoil surface slope is between -1 and +1.
Nac	Row identifier
NbCL	Mesh row through airfoil chord line
NSL, NST	Row identifiers same as for MSL, MST
PRMSH(KR)	Ratio, DY(KR)/DX(KR)
PRMSHI(KR)	Ratio, DX(KR)/DY(KR)
PRMSHY	Ratio of radius mesh spacing to angular mesh spacing for the polar coordinate region, DX(1)/DY(1).
PRMYI	Ratio, 1/PRMSHY
QDNYI	$1/(M_\infty^2/2)$
RADC	Radius of Curvature
RADCL	Lower surface radius of curvature.
RADCU	Upper surface radius of curvature.
RBNDRY	Actual radius of inner ring of polar mesh region.
RBNDY	Nondimensional radius of inner ring of polar mesh region.
RH(I)	Nondimensional inverse radius of ring I in the polar mesh region, I=1,9.
SAb(K,M)	Sine of the airfoil surface slope at column M intersection with airfoil. K=1 is lower surface, K=2 is upper surface.

SAN(N) Sine of airfoil slope at intersection of horizontal row N with nose of airfoil.

SCb(I,L) Sine and cosine of clockwise tangent to airfoil surface at the base of the surface normals. I=1 is sine, I=2 is cosine. L=1, LBb MAX.

SCY(I,J) Sine and cosine of inclination angles of the rays in the polar mesh region. I=1 is sine, I=2 is cosine. J=1,42.

TSN(KR) Constant, 1/2 times the input safety factor for region KR.

UMACH Free stream Mach number.

UNOG Constant, $1/\gamma$

UNOGMU Constant, $1/(\gamma-1)$

Wa(I,M,N) Dependent conservation variable for region a.

I = 1 is x-mass flow

I = 2 is y-mass flow

I = 3 is density

M = 1, MaT, while N = 1, NaT.

WSb(I,J,L) Dependent conservation variable for local airfoil rotated meshes.

I = 1 is tangential mass flow

I = 2 is normal mass flow

I = 3 is density

J = 1,9 for the nine nodes in each of local rotated array L. L=1, LBb MAX.

WSu(I) Conservation variables at free stream conditions, I same as for Wa.

XY$\underline{b}$(I,L)	x and y coordinates of points on the airfoil surface at the base of the surface normals for mesh $\underline{b}$ I=1 is x, I=2 is y, L=1, LBb MAX
YLS	Airfoil y-coordinate for lower surface.
YPNOSE	Slope at airfoil nose.
YPSL	Airfoil lower surface slope.
YPSU	Airfoil supper surface slope.
YUS	Airfoil y-coordinate for upper surface.

AI.4 INTERPOLATION WEIGHTS

Several types of interpolations are performed in the program. For each type of interpolation information is required regarding the point for which the interpolation is being made, the point which identifies the location from which the interpolation is being made, and the interpolation weights (or coefficients) which are used to perform the interpolation. This information is established in the setup program and stored.

Interpolation types and the associated FORTRAN variables:

a. Polar to Coarse Cartesian Mesh

Interpolation is required to define dependent variables at the four boundaries of the coarse cartesian mesh.

ICB(I,K,L) — Node identifier for the point in the polar mesh nearest the boundary point in the coarse cartesian mesh

I identifies the point in the polar mesh

= 1 ring no.

= 2 ray no.

K identifies the boundary of the coarse cartesian mesh

= 1 left boundary column

= 2 right boundary column

= 3 bottom boundary row

= 4 top boundary row

L identifies the point on the coarse Cartesian mesh boundary

= 1 - MCT
(This assumes that the coarse cartesian mesh has the same number of rows and columns; i.e. MCT = NCT)

DCB(I,K,L) Displacements of the boundary point of the coarse cartesian mesh from the point in the polar mesh identified by ICB(I,K,L)

I = 1 is radial spacing

= 2 is angular spacing

(Expressed is fractions of the polar mesh spacing)
K and L have the same meaning as for ICB.

b. Coarse Cartesian Mesh to Polar Mesh

Interpolation is required to define dependent variables on the innermost ring of the polar mesh.

IYB(I,L) Node identifier for the point in the coarse cartesian mesh nearest to the innerpoint on ray L of the polar mesh (L=1 - MYT)

I = 1 is column no.

= 2 is row no.

DYB(I,L) Displacements of the inner point on ray L of the polar mesh from the point identified by IYB(I,L)

I = 1 is x spacing

= 2 is y spacing

(Expressed as fractions of the coarse cartesian mesh spacing).

c. Cartesian Mesh to Local Rotated Mesh

Interpolation is required to define dependent variables at the upper three points in the local rotated array (see Figure 7)

IWb (I,L) — Node identifier for the point in the cartesian mesh nearest the point in the local rotated array

I = 1 Point 7

= 2 Point 8

= 3 Point 9

The column and row members are "packed" into IWb (I,L),i.e.

IWb = xx (Col. No.) xx (Row No.)

CWb(9,I,L) — Interpolation weights for the 9 point array in the cartesian mesh centered about the point indicated by IWb(I,L).

Special Case: For a cartesian mesh with an embedded (finer) cartesian mesh, the auxiliary surface normals require special interpolations for the points labeled A in Figure 8.

IWfX(I,J) — Node identifier for points in the cartesian mesh

I = 1,2 for the two points A in Figure 8.

J = 1 is upper surface

J = 2 is lower surface

Column/row numbers are packed into IWfX

IWfX = xx (Col. No.) xx (Row No.)

CWfX(9,I,J) Interpolation weights corresponding to points indicated by IWfX(I,J).

d. Distorted Mesh to Local Rotated Mesh

Interpolation is required to define independent variables for the local rotated mesh associated with each surface normal. (Distorted mesh is the collection of 3 adjacent surface normals).

DWb(9,I,L) Interpolation weights for 9 point distorted array used to define dependent variables for points in the local rotated mesh.

I identifies the point in the local rotated mesh, see Figure 7

= 1 point no. 1

= 2 point no. 3

= 3 point no. 4

= 4 point no. 6

L identifies the surface normal (L=1 LBb MAX)

Special Case: For the surface normals at the ends of the operating ranges, the interpolation from distorted to local rotated meshes requires special interpolation weights.

DWbX(9,I,J) Interpolation weights for the distorted array used to define dependent variables for points in the local rotated mesh at the ends of the operating ranges. See points marked B in Fig. 8.

I = 1 for point nearest surface

= 2 for next point away from surface

J = 1 for surface normal at the forward operating limit (upper airfoil surface)

= 2 for surface normal at the aft operating limit (upper airfoil surface)

= 3 for aft operating limit (lower airfoil surface)

= 4 for forward operating limit (lower airfoil surface)

= 5 for lower operating limit (blunt part of nose region).

= 6 for upper operating limit (blunt part of nose region).

e. From Surface Normals to Cartesian Auxiliaries

Interpolation is required to define dependent variables at the cartesian mesh points nearest to the airfoil surface.

AWb(3,L) — Interpolation weights for the three points along a surface normal used to define values at cartesian auxiliaries.

L = 1 — LBb MAX

f. Field Boundaries for "E" Points

Special interpolation is required at field boundaries to assign values for embedded (finer) mesh surface normals (Points labeled E in Figure 8).

AWHM (3,3), AWMT (3,3) — Interpolation weights applied to the 3 values along a surface normal in the outer mesh to define values for the 3 points required for the inner mesh.

g. Field Boundaries for "F" Points

Special interpolation is required to assign values to boundary points for an inner (finer) mesh near the airfoil surface where interpolation cannot be performed between boundary points in the outer (coarser) mesh. (See points labeled "F" in Figure 8).

IWHM(I,J), IWMT(I,J) — Node identifiers for the fine mesh points requiring special interpolation

I = 1 upper surface

= 2 lower surface

	J-2	4 individual boundary points onto which data is to be interpolated. (Assumes there will be no more than 4 or that mesh size ratio is no more than 5).
DWHM(9,I,J) DWMT(9,I,j)		Interpolation weights for points identified by IWHM( ) and IWMT( ). I and J have same meanings.

APPENDIX II

SETUP PROGRAM FORTRAN LISTING

```
      PROGRAM TEHAI (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE4,TAPE3)
C=====THIS VERSION FOR CAMBERED LIFTING AIRFOILS, OPERATIONAL LIMITS TO BE
C=====UMACH = MACH NUMBER
C=====GAM   = RATIO OF SPECIFIC HEATS
C=====ALPHA = ANGLE OF ATTACK, DEGREES
      COMMON/LIMITS/ MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT,
     C MCA, MCB, MCE, MCF, MCG, MCH, MCS, MCT, NCA, NCB, NCE, NCF,
     C NCG, NCH, NCS, NCT,
     E MFA, MFB, MFD, MFE, MFF, MFG, MFH, MFQ, MFS, MFT, MFAN,
     E MFANP, MEAT, MEATM, MFATP, NFA, NFB, NFD, NFE, NFF, NFG, NFH,
     E NFQ, NFS, NFT, NFCL,
     H MHA, MHB, MHD, MHE, MHF, MHG, MHH, MHQ, MHS, MHT, MHAN, MHANP,
     H NHA, NHB, NHD, NHE, NHF, NHG, NHH, NHQ, NHS, NHT, NHCL,
     M MMA, MMB, MMD, MME, MMF, MMG, MMH, MMQ, MMS, MMT, MMAN, MMANP,
     M NMA, NMB, NMD, NME, NMF, NMG, NMH, NMQ, NMS, NMT, NMCL,
     T MTA, MTB, MTD, MTQ, MTS, MTT, MTAN, MTANP, NTA, NTB, NTD,
     T NTQ, NTS, NTT, NTCL,
     S MSL, MST, NSL, NST
      COMMON/PLIMIT/  LBFA, LBFB, LBFC, LBFD, LBFE, LBFI, LBFJ, LBFK,
     F LBFL, LBFM, LBFMAX,
     H LBHA, LBHB, LBHC, LBHD, LBHE, LBHI, LBHJ, LBHK, LBHL, LBHM,
     H LBHMAX,
     M LBMA, LBMB, LBMC, LBMD, LBME, LBMI, LBMJ, LBMK, LBML, LBMM,
     M LBMMAX,
     T LBTA, LBTB, LBTC, LBTD, LBTE, LBTI, LBTJ, LBTK, LBTL, LBTM,
     T LBTQ, LBTR, LBTS, LBTT, LBTMAX
      COMMON/ZMESH/   DX(6), DY(6), RH(9)
      COMMON/ GEOMF/ SY(42), CY(42), DYB(2,42), DCB(2,4,17),
     A XYF(2,42), SCF(2,42), XYH(2,22), SCH(2,22),
     B XYM(2,14), SCM(2,14), XYT(2,43), SCT(2,43)
      COMMON/OPLM/   ICO(4,17), JCO(4,17), IFO(8,37), JFO(8,37),
     A IHO(8,25), JHO(8,25), IMO(8,26), JMO(8,26), ITO(8,16), JTO(8,16)
      COMMON/DUMA/   L
      COMMON/TERSE/  IYB(2,42), ICB(2,4,17), IDEH(10), IDHM(10),
     A IDMT(10), IWF(3,42), IWH(3,22), IWM(3,14), IWT(3,43),
     B LINTF(2), LINTH(2), LINTM(2), IWFX(2,2), IWHX(2,2), IWMX(2,2),
     C IAF(2,37), IAH(2,25), IAM(2,26), IAT(2,16), IAN(21),
     D IWFH(2,4), IWHM(2,4), IWMT(2,4)
      COMMON/DMOSA/  DWF(9,4,42), DWH(9,4,22), DWM(9,4,14),
     A DWT(9,4,43), AWF(3,42), AWH(3,22), AWM(3,14), AWT(3,43),
     B CWF(9,3,42), CWH(9,3,22), CWM(9,3,14), CWT(9,3,43),
     C DWFX(9,2,6), DWHX(9,2,6), DWMX(9,2,6), DWTX(9,2,6),
     D AWFH(3,3), AWHM(3,3), AWMT(3,3),
     E CWFH(9,2,4), CWHM(9,2,4), DWMT(9,2,4),
     F CWFX(9,2,2), CWHX(9,2,2), CWMX(9,2,2)
      COMMON/GAS/    GAM, HGMU, UNOG, DOGMU, [illegible], HINF, GMU, UNOGMU
      COMMON/FLOW/ UMACH, ALPHA, RBNDY
      COMMON/SPOT/M,KB
      COMMON/AIRF/YUS,YLS,YPSU,YPSL,XNOSE,YPNOSE,RADCU,RADCL,RADC
      COMMON          WY(3,42,9), WC(3,17,17), WF(3,37,13), WH(3,25,17),
     A WM(3,26,26), WT(3,16,21), WSF(3,9,42), WSH(3,9,22), WSM(3,9,14),
     B WST(3,9,43), WSU(3)
      DIMENSION             DAF(2,37), CAF(2,37), SAF(2,37),
     A                      DAH(2,25), CAH(2,25), SAH(2,25),
     B                      DAM(2,26), CAM(2,26), SAM(2,26),
```

```
     C             DAT(2,16), CAT(2,16), SAT(2,16),
     D            DAN(21), CAN(21), SAN(21),
     E DZZ(3), CSF(42), DSH(22), DSM(14), DST(43)
C-----NEW PROBLEM
  100 PI = 3.1415926536
      DTHET = PI/20.0
      DRHO = 0.125
      DX(1) = DRHO
      DX(2) = 0.2
      DX(3) = 0.05
      DX(4) = 0.0125
      DX(5) = 0.0025
      DX(6) = 0.0025
      DY(1) = DTHET
      DY(2) = 0.2
      DY(3) = 0.1
      DY(4) = 0.025
      DY(5) = 0.005
      DY(6) = 0.0025
C-----OUTER POLAR COORDINATE FIELD
      MYA = 1      $MYB = 2  $MYS = 41 $MYT = 42
      NYA = 1      $NYB = 2  $NYS = 8  $NYT = 9
C-----COARSE CARTESIAN MESH
      MCA = 1      $MCB = 2  $MCE = 6  $MCF = 7  $MCG = 14 $MCH = 15
      MCS = 16     $MCT = 17
      NCA = 1      $NCB = 2  $NCE = 6  $NCF = 7  $NCG = 11 $NCH = 12
      NCS = 16     $NCT = 17
C-----FINE CARTESIAN MESH
      MFA = 1      $MFB = 2  $MFD = 5  $MFE = 6  $MFF = 7  $MFG = 11
      MFM = 12     $MFQ = 33 $MFS = 36 $MFT = 37 $MFAN = 9 $MFANP = 10
      MFAT = 29    $MFATM = 28         $MFATP = 30
      NFA = 1      $NFB = 2  $NFD = 3  $NFE = 4  $NFF = 5  $NFG = 7
      NFM = 8      $NFQ = 11 $NFS = 12 $NFT = 13 $NFCL = 6
C-----OUTER NOSE REGION
      MHA = 1      $MHB = 2  $MHD = 5  $MHE = 11 $MHF = 12 $MHG = 15
      MHM = 16     $MHQ = 21 $MHS = 24 $MHT = 25 $MHAN = 13 $MHANP = 14
      NHA = 1      $NHB = 2  $NHD = 5  $NHE = 7  $NHF = 8  $NHG = 11
      NHM = 12     $NHQ = 13 $NHS = 16 $NHT = 17 $NHCL = 9
C-----INTERMEDIATE NOSE REGION
      MMA = 1      $MMB = 2  $MMD = 6  $MME = 6  $MMF = 7  $MMG = 20
      MMM = 21     $MMQ = 21 $MMS = 25 $MMT = 26 $MMAN = 11 $MMANP = 12
      NMA = 1      $NMB = 2  $NMD = 6  $NME = 7  $NMF = 8  $NMG = 16
      NMM = 17     $NMQ = 21 $NMS = 25 $NMT = 26 $NMCL = 11
      MTA = 1      $MTB = 2  $MTD = 2  $MTQ = 15 $MTS = 15 $MTT = 16
      MTAN = 6     $MTANP = 7
      NTA = 1      $NTB = 2  $NTD = 3  $NTQ = 19 $NTS = 20 $NTT = 21
      NTCL = 9
      BRNDY = 1.4
      THETA = -2.0*DTHET
      RH(1) = 1.0
      DO 101 NY = 2,NYT
  101 RH(NY) = RH(NY-1) + DRHO
C-----COARSE TO FINE INTERPOLATION CONSTANTS
      DO 136 L = 1,MYT
      THETA = THETA + DTHET
```

```
      SY(L) = SIN(THETA)
      CY(L) = COS(THETA)
      X = RBNDY*CY(L)
      Y = RBNDY*SY(L)
      J = 1
      XTR = -1.6
  102 IF(X - XTR) 106,112,104
  104 XTR = XTR + DX(2)
      J = J + 1
      GO TO 102
  106 XRF = (XTR - X)/DX(2)
      IF(XRF - 0.5) 110,108,108
  108 IYB(1,L) = J - 1
      DYB(1,L) = 1.0 - XRF
      GO TO 116
  110 IYB(1,L) = J
      DYB(1,L) = -XRF
      GO TO 116
  112 IYB(1,L) = J
      DYB(1,L) = 0.0
  116 J = 1
      YTR = -1.6
  122 IF(Y - YTR) 126,132,124
  124 YTR = YTR + DY(2)
      J = J + 1
      GO TO 122
  126 YRF = (YTR - Y)/DY(2)
      IF(YRF - 0.5) 130,128,128
  128 IYB(2,L) = J - 1
      DYB(2,L) = 1.0 - YRF
      GO TO 136
  130 IYB(2,L) = J
      DYB(2,L) = -YRF
      GO TO 136
  132 IYB(2,L) = J
      DYB(2,L) = 0.0
  136 CONTINUE
C-----FINE TO COARSE INTERPOLATION CONSTANTS
C-----LEFT COLUMN
      K = 1
      XSU = -1.6
      YSU = -1.6 - DY(2)
      LTP = NCT
      DXP = 0.0
      DYP = DY(2)
C-----COLUMN AND ROW ROUTINE
  140 DO 190 L = 1,LTP
      XSU = XSU + DXP
      YSU = YSU + DYP
      RHA = RBNDY/SQRT(XSU**2 + YSU**2)
      IF(XSU) 142,144,142
  142 THETA = ATAN2(YSU,XSU)
      IF(THETA.LT.0.0) THETA = THETA + 2.0*PI
      GO TO 150
  144 IF(YSU) 146,148,148
```

```
  146 THETA = 1.5*PI
      GO TO 150
  148 THETA = 0.5*PI
C-----M LOCATION
  150 J = 1
      TTR = -DTHET
  152 IF(THETA - TTR) 156,162,154
  154 TTR = TTR + DTHET
      J = J + 1
      GO TO 152
  156 TRF = (TTR - THETA)/DTHET
      IF(TRF - 0.5) 160,158,158
  158 ICB(1,K,L) = J - 1
      DCB(1,K,L) = 1.0 - TRF
      GO TO 170
  160 ICB(1,K,L) = J
      DCB(1,K,L) = -TRF
      GO TO 170
  162 ICB(1,K,L) = J
      DCB(1,K,L) = 0.0
C-----N LOCATION
  170 J = 1
      RTR = 1.0
  172 IF(RHA - RTR) 174,182,176
  174 RTR = RTR - DRHO
      J = J +1
      GO TO 172
  176 RRF = (RHA- RTR)/DRHO
      IF(RRF - 0.5) 180,180,178
  178 ICB(2,K,L) = J - 1
      DCB(2,K,L) = 1.0 - RRF
      GO TO 190
  180 ICB(2,K,L) = J
      DCB(2,K,L) = -RRF
      GO TO 190
  182 ICB(2,K,L) = J
      DCB(2,K,L) = 0.0
  190 CONTINUE
      GO TO (192,194,196,200), K
C-----RIGHT COLUMN
  192 K = 2
      XSU = 1.6
      YSU = -1.6 - DY(2)
      GO TO 140
C-----BOTTOM ROW
  194 K = 3
      XSU = -1.6 - DX(2)
      YSU = -1.6
      LTP = MCT
      DXP = DX(2)
      DYP = 0.0
      GO TO 140
C-----TOP ROW
  196 K = 4
      YSU = 1.6
```

```
      XSU = -1.6 - DX(2)
      GO TO 140
C-----PRINT INTERPOLATION CONSTANTS
  200 WRITE(6,23)
      WRITE(6,25)
      DO 204 MY = 1,MYT
  204 WRITE(6,10) MY,(IYB(L,MY), L = 1,2),(DYB(L,MY), L = 1,2)
      WRITE(6,27)
      WRITE(6,31)
      DO 206 NC = 1,NCT
  206 WRITE(6,10)NC,(ICB(L,1,NC),L=1,2),(DCB(L,1,NC),L=1,2)
      WRITE(6,28)
      WRITE(6,31)
      DO 208 NC = 1,NCT
  208 WRITE(6,10)NC,(ICB(L,2,NC),L=1,2),(DCB(L,2,NC),L=1,2)
      WRITE(6,29)
      WRITE(6,32)
      DO 210 NC = 1,MCT
  210 WRITE(6,10)NC,(ICB(L,3,NC),L=1,2),(DCB(L,3,NC),L=1,2)
      WRITE(6,30)
      WRITE(6,32)
      DO 212 NC = 1,MCT
  212 WRITE(6,10)NC,(ICB(L,4,NC),L=1,2),(DCB(L,4,NC),L=1,2)
C-----AIRFOIL DESCRIPTION
  250 CALL SHAPE(1,0.0)
      WRITE(6,33)
      WRITE(6,43)
      WRITE(6,34)
C-----FINE CARTESIAN MESH
      XAIR = 0.0 - DX(3)
      DO 270 M = MFAN,MFAT
      XAIR = XAIR + DX(3)
      CALL SHAPE(2,XAIR)
      WRITE(6,8) XAIR,YUS,YPSU,YLS,YPSL
      YSS = YLS
      YPSS = YPSL
      KAT = 1
      N = NFF
      YTEST = -CY(3)
  252 IF(YSS - YTEST) 256,258,254
  254 N = N + 1
      YTEST = YTEST + DY(3)
      GO TO 252
  256 IAF(KAT,M) = N - 1
      DAF(KAT,M) = 1.0 -(YTEST - YSS)/DY(3)
      GO TO 262
  258 IAF(KAT,M) = N - 1
      DAF(KAT,M) = 1.0
  262 CAF(KAT,M) = 1.0/SQRT(1.0 + YPSS**2)
      SAF(KAT,M) = YPSS*CAF(KAT,M)
      IF(KAT.EQ.2) GO TO 270
      YSS = YUS
      YPSS = YPSU
      YTEST = 0.0
      N = NFCL
```

```
      KAT = 2
  264 IF(YSS - YTEST) 266,267,265
  265 N = N + 1
      YTEST = YTEST + DY(3)
      GO TO 264
  266 IAF(KAT,M) = N
      UAF(KAT,M) = (YTEST - YSS)/DY(3)
      GO TO 262
  267 IAF(KAT,M) = N + 1
      UAF(KAT,M) = 1.0
      GO TO 262
  270 CONTINUE
C-----OUTER NOSE REGION
 2701 WRITE(6,51)
      WRITE(6,54)
      XAIR = 0.0 -DX(4)
      DO 2720 M = MHAN,MHT
      XAIR = XAIR + DX(4)
      CALL SHAPE(2,XAIR)
      WRITE (6,6) XAIR, YUS, YPSU, YLS, YPSL
      YSS = YLS
      YPSS = YPSL
      KAT = 1
      N = NHCL - 2
      YTEST = -2.0*DY(4)
 2702 IF(YSS - YTEST) 2706,2708,2704
 2704 N = N + 1
      YTEST = YTEST + DY(4)
      GO TO 2702
 2706 IAM(KAT,M) = N - 1
      UAM(KAT,M) = 1.0 -(YTEST - YSS)/DY(4)
      GO TO 2712
 2708 IAM(KAT,M) = N - 1
      UAM(KAT,M) = 1.0
 2712 CAM(KAT,M) = 1.0/SQRT(1.0 + YPSS**2)
      SAM(KAT,M) = YPSS*CAM(KAT,M)
      IF(KAT.EQ.2) GO TO 2720
      YSS = YUS
      YPSS = YPSU
      YTEST = 0.0
      N = NHCL
      KAT = 2
 2714 IF(YSS - YTEST) 2716,2717,2715
 2715 N = N + 1
      YTEST = YTEST + DY(4)
      GO TO 2714
 2716 IAM(KAT,M) = N
      UAM(KAT,M) = (YTEST - YSS)/DY(4)
      GO TO 2712
 2717 IAM(KAT,M) = N + 1
      UAM(KAT,M) = 1.0
      GO TO 2712
 2720 CONTINUE
C-----INTERMEDIATE NOSE REGION
 2801 WRITE(6,52)
```

```
      WRITE(6,34)
      XAIR = 0.0 -DX(5)
      DO 2820 M = MMAN,MMT
      XAIR = XAIR + DX(5)
      CALL SHAPE(2,XAIR)
      WRITE (6,6) XAIR, YUS, YPSU, YLS, YPSL
      YSS = YLS
      YPSS = YPSL
      KAT = 1
      N = NMCL - 5
      YTEST = -5.0*DY(5)
 2802 IF(YSS - YTEST) 2806,2808,2804
 2804 N = N + 1
      YTEST = YTEST + DY(5)
      GO TO 2802
 2806 IAM(KAT,M) = N - 1
      UAM(KAT,M) = 1.0 -(YTEST - YSS)/DY(5)
      GO TO 2812
 2808 IAM(KAT,M) = N - 1
      UAM(KAT,M) = 1.0
 2812 CAM(KAT,M) = 1.0/SQRT(1.0 + YPSS**2)
      SAM(KAT,M) = YPSS*CAM(KAT,M)
      IF(KAT.EQ.2) GO TO 2820
      YSS = YUS
      YPSS = YPSU
      YTEST = 0.0
      N = NMCL
      KAT = 2
 2814 IF(YSS - YTEST) 2816,2817,2815
 2815 N = N + 1
      YTEST = YTEST + DY(5)
      GO TO 2814
 2816 IAM(KAT,M) = N
      UAM(KAT,M) = (YTEST - YSS)/DY(5)
      GO TO 2812
 2817 IAM(KAT,M) = N + 1
      UAM(KAT,M) = 1.0
      GO TO 2812
 2820 CONTINUE
C-----INNER NOSE REGION
 2901 WRITE(6,44)
      WRITE(6,34)
      XAIR = 0.0 -DX(6)
      DO 2920 M = MTAN,MTT
      XAIR = XAIR + DX(6)
      CALL SHAPE(2,XAIR)
      WRITE (6,6) XAIR, YUS, YPSU, YLS, YPSL
      YSS = YLS
      YPSS = YPSL
      KAT = 1
      N = NTCL - 8
      YTEST = -8.0*DY(6)
 2902 IF(YSS - YTEST) 2906,2908,2904
 2904 N = N + 1
      YTEST = YTEST + DY(6)
```

```
      GO TO 2902
 2906 IAT(KAT,M) = N - 1
      DAT(KAT,M) = 1.0 -(YTEST - YSS)/DY(6)
      GO TO 2912
 2908 IAT(KAT,M) = N
      DAT(KAT,M) = 0.0
 2912 CAT(KAT,M) = 1.0/SQRT(1.0 + YPSS**2)
      SAT(KAT,M) = YPSS*CAT(KAT,M)
      IF(KAT.EQ.2) GO TO 2920
      YSS = YUS
      YPSS = YPSU
      YTEST = 0.0
      N = NTCL
      KAT = 2
 2914 IF(YSS - YTEST) 2916,2917,2915
 2915 N = N + 1
      YTEST = YTEST + DY(6)
      GO TO 2914
 2916 IAT(KAT,M) = N
      DAT(KAT,M) = (YTEST - YSS)/DY(6)
      GO TO 2912
 2917 IAT(KAT,M) = N
      DAT(KAT,M) = 0.0
      GO TO 2912
 2920 CONTINUE
C-----NOSE REGION HORIZONTAL CUTS
      WRITE(6,45)
      WRITE(6,36)
      KAT = 1
      NSL = IAT(1,MTT)
      NST = IAT(2,MTT)
      YAIR = FLOAT(NSL - 1 - NTCL)*DY(6)
  301 DO 314 N = NSL,NST
      YAIR = YAIR + DY(6)
      CALL SHAPE(3,YAIR)
      WRITE(6,8) YAIR,XNOSE,YPNOSE
      IF(N.EQ.NTCL) GO TO 309
      XSS = XNOSE
      M = MTAN - 4
      XTEST = -4.0*DX(6)
  302 IF(XSS - XTEST) 306,308,304
  304 M = M + 1
      XTEST = XTEST + DX(6)
      GO TO 302
  306 IAN(N) = M - 1
      DAN(N) = 1.0 - (XTEST - XSS)/DX(6)
      GO TO 312
  308 IAN(N) = M
      DAN(N) = 0.0
      GO TO 312
  309 IAN(N) = MTAN
      DAN(N) = 0.0
  312 CAN(N) = 1.0/SQRT(1.0 + YPNOSE**2)
      SAN(N) = YPNOSE*CAN(N)
  314 CONTINUE
```

```
C-----AIRFOIL INTERPOLATION OUTPUT
  515 WRITE(6,40)
      WRITE(6,43)
      WRITE(6,46)
      DO 516 M = MFA,MFT
  516 WRITE(6,10) M, (IAF(K,M), K = 1,2), (DAF(K,M), K = 1,2)
      WRITE (6,51)
      WRITE (6,46)
      DO 522 M = MHA,MHT
  522 WRITE(6,10) M, (IAH(K,M), K = 1,2), (DAH(K,M), K = 1,2)
      WRITE(6,52)
      WRITE(6,46)
      DO 518 M = MMA,MMT
  518 WRITE(6,10) M, (IAM(K,M), K = 1,2), (DAM(K,M), K = 1,2)
      WRITE (6,44)
      WRITE (6,46)
      DO 524 M = MTA,MTT
  524 WRITE(6,10) M, (IAT(K,M), K = 1,2), (DAT(K,M), K = 1,2)
      WRITE(6,45)
      WRITE(6,47)
      DO 520 N = NTA,NTT
  520 WRITE(6,10) N, IAN(N), IAN(N), DAN(N)
C-----SET CARTESIAN MESH OPERATIONAL LIMITS
C-----COARSE MESH, FIELD 2
      MLL = MCB
      MUL = MCE
      KL = 1
  320 DO 322 M = MLL,MUL
      ICO(1,M) = 2
      JCO(1,M) = 5
      ICO(2,M) = 17
  322 JCO(2,M) = 0
      GO TO (324,326),KL
  324 KL = 2
      MLL = MCH
      MUL = MCS
      GO TO 320
  326 DO 328 M = MCF,MCG
      ICO(1,M) = 2
      JCO(1,M) = 5
      ICO(2,M) = 7
      JCO(2,M) = 0
      ICO(3,M) = 12
      JCO(3,M) = 5
      ICO(4,M) = 17
  328 JCO(4,M) = 0
C-----FINE MESH, FIELD 3. ASSUMES AIRFOIL BETWEEN Y=0.1, -0.1
      DO 330 M = MFB,MFS
      IFO(1,M) = 2
  330 IFO(2,M) = 3
      MLL = MFB
      MUL = MFQ - 1
      DO 332 M = MLL,MUL
      JFO(1,M) = 105
      JFO(2,M) = 405
```

```
      IFQ(3,M) = 12
      IFQ(4,M) = 13
      JFQ(3,M) = 705
  332 JFQ(4,M) = 700
      MLL = MFQ + 1
      MUL = MFS
      DO 334 M = MLL,MUL
      JFQ(1,M) = 305
      JFQ(2,M) = 605
      IFQ(3,M) = 12
      IFQ(4,M) = 13
      JFQ(3,M) = 905
      IFQ(3,M) = 12
      IFQ(4,M) = 13
  334 JFQ(4,M) = 900
      KL = 1
      MLL = MFD
      MUL = MFE
  336 DO 338 M = MLL,MUL
      JFQ(1,M) = 205
      JFQ(2,M) = 505
      IFQ(3,M) = 12
      JFQ(3,M) = 805
      IFQ(4,M) = 13
  338 JFQ(4,M) = 800
      GO TO (340,342),KL
  340 KL = 2
      MLL = MFATP
      MUL = MFQ
      GO TO 336
  342 DO 344 M = MFF,MFAT
      JFQ(1,M) = 205
      JFQ(2,M) = 505
      IFQ(3,M) = 5
      JFQ(3,M) = 500
      IFQ(4,M) = 8
      JFQ(4,M) = 505
      IFQ(5,M) = 12
      JFQ(5,M) = 805
      IFQ(6,M) = 13
  344 JFQ(6,M) = 800
C-----OUTER NOSE REGION, FIELD 4
C     ASSUMES UPPER SURFACE SLOPE POSITIVE, LOWER SLOPE NEGATIVE
      DO 350 M = MHB,MHS
      IHQ(1,M) = 2
  350 IHQ(2,M) = 5
      MLL = MHB
      MUL = MHD - 1
      DO 352 M = MLL,MUL
      JHQ(1,M) = 105
      JHQ(2,M) = 405
      IHQ(3,M) = 14
      JHQ(3,M) = 705
      IHQ(4,M) = 17
  352 JHQ(4,M) = 700
```

```
      DO 354 M = MHD,MHS
      JHO(1,M) = 205
  354 JHO(2,M) = 505
      DO 356 M = MHD,MHE
      IHO(3,M) = 14
      JHO(3,M) = 805
      IHO(4,M) = 17
  356 JHO(4,M) = 800
      DO 358 M = MHE,MHG
      IHO(3,M) = 8
      JHO(3,M) = 500
      IHO(4,M) = 12
      JHO(4,M) = 505
      IHO(5,M) = 14
      JHO(5,M) = 805
      IHO(6,M) = 17
  358 JHO(6,M) = 800
      DO 364 M = MHH,MHS
      J = 3
      IF(IAH(1,M).EQ.IAH(1,M+1)) GO TO 360
      IHO(J,M) = IAH(1,M) - 1
      JHO(J,M) = 513
      J = 4
  360 IHO(J,M) = IAH(1,M)
      JHO(J,M) = 500
      J = J + 1
      IHO(J,M) = IAH(2,M) + 1
      IF(IAH(2,M).EQ.IAH(2,M+1)) GO TO 362
      JHO(J,M) = 511
      J = J + 1
      IHO(J,M) = IHO(J-1,M) + 1
  362 JHO(J,M) = 505
      J = J + 1
      IHO(J,M) = 14
      JHO(J,M) = 805
      J = J + 1
      IHO(J,M) = 17
  364 JHO(J,M) = 800
      MLL = MHG + 1
      MUL = MHS
      DO 366 M = MLL,MUL
      DO 366 J = 1,8
  366 JHO(J,M) = JHO(J,M) + 100
C-----INTERMEDIATE NOSE REGION, FIELD 5
C-----ASSUMES UPPER SURFACE SLOPE POSITIVE, LESS THAN 63 DEGREES,
C     LOWER SLOPE NEGATIVE
      DO 370 M = MMB,MMS
      IMO(1,M) = 2
      JMO(1,M) = 105
      IMO(2,M) = 6
  370 JMO(2,M) = 405
      DO 372 M = MMB,MME
      IMO(3,M) = 22
      JMO(3,M) = 705
      IMO(4,M) = 26
```

```
  372 JMO(4,M) = 700
      DO 374 M = MMD,MMS
      DO 374 J = 1,2
  374 JMO(J,M) = JMO(J,M) + 100
      DO 376 M = MMD,MME
      DO 376 J = 3,4
  376 JMO(J,M) = JMO(J,M) + 100
      DO 378 M = MMF,MMG
      IMO(3,M) = 8
      JMO(3,M) = 500
      IMO(4,M) = 17
      JMO(4,M) = 505
      IMO(5,M) = 22
      JMO(5,M) = 805
      IMO(6,M) = 26
  378 JMO(6,M) = 800
      DO 384 M = MMH,MMS
      J = 3
      IF(IAM(1,M).EQ.IAM(1,M+1)) GO TO 380
      IMO(J,M) = IAM(1,M) - 1
      JMO(J,M) = 513
      J = 4
  380 IMO(J,M) = IAM(1,M)
      JMO(J,M) = 500
      J = J + 1
      IMO(J,M) = IAM(2,M) + 1
      IF(IAM(2,M).EQ.IAM(2,M+1)) GO TO 382
      JMO(J,M) = 511
      J = J + 1
      IMO(J,M) = IMO(J-1,M) + 1
  382 JMO(J,M) = 505
      J = J + 1
      IMO(J,M) = 22
      JMO(J,M) = 805
      J = J + 1
      IMO(J,M) = 26
  384 JMO(J,M) = 800
      MLL = MMG + 1
      MUL = MMS
      DO 386 M = MLL,MUL
      DO 386 J = 1,8
  386 JMO(J,M) = JMO(J,M) + 100
C-----INNER NOSE REGION, FIELD 6
      MFEN = IAN(NSL)
      DO 390 N = NSL,NST
      IF(IAN(N) - MFEN) 388,390,390
  388 MFEN = IAN(N)
      NFEN = N
  390 CONTINUE
      DO 392 M = MTB,MTS
      ITQ(1,M) = 2
      ITQ(2,M) = 3
      JTQ(1,M) = 205
  392 JTQ(2,M) = 505
      MLL = MTB
```

```
      MUL = MFFN - 1
      DO 394 M = MLL,MUL
      ITO(3,M) = 20
      JTO(3,M) = 805
      ITO(4,M) = 21
  394 JTO(4,M) = 800
C-----GRAZING PART OF NOSE, ASSUMES FORWARD CHORD OVERHANG LESS THAN DX(6)
C     NOSE CAMBER ZERO OR POSITIVE
      M = MFFN
      ITO(3,M) = NFFN - 1
      JTO(3,M) = 513
      ITO(4,M) = NFFN
      JTO(4,M) = 500
      N = NFFN
  396 IF(IAN(N+1) - IAN(N)) 398,398,400
  398 N = N + 1
      GO TO 396
  400 ITO(5,M) = N + 1
  401 JTO(5,M) = 511
      ITO(6,M) = N + 2
      JTO(6,M) = 505
      ITO(7,M) = 20
      JTO(7,M) = 805
      ITO(8,M) = 21
      JTO(8,M) = 800
      IF(M.EQ.MTAN) GO TO 420
      M = MTAN
      N = NSL
  402 IF(IAN(N) - M) 406,406,404
  404 N = N + 1
      GO TO 402
  406 ITO(3,M) = N - 1
      ITO(4,M) = N
      JTO(3,M) = 513
      JTO(4,M) = 500
      N = NTCL
  408 IF(IAN(N+1) - M) 410,410,400
  410 N = N + 1
      GO TO 408
C-----MAIN PART OF INNER NOSE REGION
  420 DO 430 M = MTANP,MTS
      J = 3
      IF(IAT(1,M).EQ.IAT(1,M+1)) GO TO 422
      ITO(J,M) = IAT(1,M+1)
      JTO(J,M) = 513
      J = 4
      ITO(J,M) = ITO(J-1,M) + 1
      GO TO 424
  422 ITO(J,M) = IAT(1,M)
  424 JTO(J,M) = 500
      J = J + 1
      IF(IAT(2,M).EQ.IAT(2,M+1)) GO TO 426
      ITO(J,M) = IAT(2,M+1)
      JTO(J,M) = 511
      J = J + 1
```

```
      ITQ(J,M) = ITQ(J-1,M) + 1
      GO TO 428
  426 ITQ(J,M) = IAT(2,M) + 1
  428 JTQ(J,M) = 505
      J = J + 1
      ITQ(J,M) = 20
      JTQ(J,M) = 805
      J = J + 1
      ITQ(J,M) = 21
  430 JTQ(J,M) = 800
C-----WRITE OPERATIONAL LIMITS
      WRITE(6,55)
      WRITE(6,60)
      DO 562 M = MCA,MCT
  562 WRITE(6,12) M, ((ICQ(K,M), JCQ(K,M)), K = 1,4)
      WRITE(6,39)
      WRITE(6,42)
      DO 568 M = MFA,MFT
  568 WRITE(6,12) M, ((IFQ(K,M), JFQ(K,M)), K = 1, 8)
      WRITE(6,51)
      WRITE(6,62)
      DO 572 M = MHA,MHT
  572 WRITE(6,12) M, ((IHQ(K,M), JHQ(K,M)), K = 1, 8)
      WRITE(6,52)
      WRITE(6,41)
      DO 564 M = MMA,MMT
  564 WRITE(6,12) M, ((IMQ(K,M), JMQ(K,M)), K = 1, 8)
      WRITE(6,38)
      WRITE(6,61)
      DO 576 M = MTA,MTT
  576 WRITE(6,12) M, ((ITQ(K,M), JTQ(K,M)), K = 1, 8)
C-----INTERPOLATION INSTRUCTIONS AT FIELD BOUNDARIES
C-----CODE 1, INERT  CODE2, 4 POINT SPREAD SYMMETRIC  CODE3, 3 POINT SPREA
C     HEAVY BELOW,   CODE 4, 3 POINT SPREAD HAAVY ABOVE
C-----FINE MESH TO OUTER NOSE REGION
      CALL BUNTE    (NEE,NFH,IFQ(1,MFH),JFQ(1,MFH),IDFH)
      CALL BUNTE    (NHE,NHH,IHQ(1,MHH),JHQ(1,MHH),IDHM)
      CALL BUNTE    (NME,NMH,IMQ(1,MMH),JMQ(1,MMH),IDMT)
      WRITE(6,51)
      WRITE(6,49)
      WRITE(6,2) (IDFH(J), J = 1,10)
      WRITE(6,52)
      WRITE(6,63)
      WRITE(6,2) (IDHM(J), J = 1,10)
      WRITE(6,35)
      WRITE(6,64)
      WRITE(6,2) (IDMT(J), J = 1,10)
C-----CONSTRUCTION OF CENTRAL AXES FOR LOCAL ROTATED BOUNDARY MESHES
C-----LOCATION OF 45 DEG. POINTS IN INNER NOSE REGION
      KAT = 1
      M = MTT
  802 IF(SAT(KAT,M) + 0.707) 806,806,804
  804 M = M - 1
      GO TO 802
  806 MSL = M
```

```
      NSL = IAT(KAT,MSL)
      KAT = 2
      M = MTT
  808 IF(SAT(KAT,M) - 0.707) 810,812,812
  810 M = M - 1
      GO TO 808
  812 MST = M
      NST = IAT(KAT,MST)
      WRITE(6,50)
      WRITE(6,2) MSL,MST,NSL,NST
C-----SURFACE NORMALS, UPPER SURFACE
      WRITE(6,65)
      WRITE(6,35)
      WRITE(6,66)
      DO 820 M = MST,MTS
      MM = M - MTAN + 1
      XA = FLOAT(M - MTAN)*DX(6)
      XTEST = XA
      N = IAT(2,M)
      YA = FLOAT(N - NTCL)*DY(6)
      XERR = -DAT(2,M)*DY(6)*SAT(2,M)*CAT(2,M)
  814 XTEST = XTEST - XERR
      CALL SHAPE(5,XTEST)
      XERR = (XTEST-XA-(YA-YUS)*YPSU)/(1.0 + YPSU**2)
      IF(ABS(XERR) - 0.01*DY(6)) 816,816,814
  816 XYT(1,MM) = XTEST
      XYT(2,MM) = YUS
      SCT(2,MM) = 1.0/SQRT(1.0 + YPSU**2)
      SCT(1,MM) = YPSU*SCT(2,MM)
      DST(MM) = SQRT((XA-XTEST)**2 + (YA-YUS)**2)/DY(6)
  820 WRITE(6,14) MM,MM,XYT(1,MM),XYT(2,MM),SCT(1,MM),SCT(2,MM),DST(MM)
C-----SURFACE NORMALS, LOWER SURFACE
      WRITE(6,66)
      DO 830 M = MSL,MTS
      MM = 2*MTT - M + 2 - MTAN
      XA = FLOAT(M - MTAN)*DX(6)
      XTEST = XA
      N = IAT(1,M)
      YA = FLOAT(N - NTCL)*DY(6)
      XERR = DAT(1,M)*DY(6)*SAT(1,M)*CAT(1,M)
  824 XTEST = XTEST - XERR
      CALL SHAPE(4,XTEST)
      XERR = (XTEST - XA + (YLS-YA)*YPSL)/(1.0 + YPSL**2)
      IF(ABS(XERR) - 0.01*DY(6)) 826,826,824
  826 XYT(1,MM) = XTEST
      XYT(2,MM) = YLS
      SCT(2,MM) = - 1.0/SQRT(1.0 + YPSL**2)
      SCT(1,MM) = YPSL*SCT(2,MM)
      DST(MM) = SQRT((XA-XTEST)**2 + (YA-YLS)**2)/DY(6)
  830 WRITE(6,14) MM,MM,XYT(1,MM),XYT(2,MM),SCT(1,MM),SCT(2,MM),DST(MM)
C-----SURFACE NORMALS, BLUNT PART OF NOSE
      WRITE(6,66)
      DO 840 N = NSL,NST
      MM = N + 2*(MTT - MTAN + 1)
      YA = FLOAT(N-NTCL)*DY(6)
```

```
      YTEST = YA
      M = IAN(N)
      XA = FLOAT(M - MTAN)*DX(6)
      YERR = DAN(N)*DX(6)*CAN(N)*SAN(N)
  834 YTEST = YTEST - YERR
      CALL SHAPE(3,YTEST)
      YERR = (YTEST-YA+(XNOSE-XA)/YPNOSE)*YPNOSE**2/(1.0+YPNOSE**2)
      IF(ABS(YERR) - 0.01*DY(6)) 836,836,834
  836 XYT(1,MM) = XNOSE
      XYT(2,MM) = YTEST
      CSF = 1.0/SQRT(1.0 + YPNOSE**2)
      SCT(2,MM) = SIGN(CSF,YPNOSE)
      SCT(1,MM) = ABS(YPNOSE*SCT(2,MM))
      DST(MM) = SQRT((YA - YTEST)**2 + (XA - XNOSE)**2)/DY(6)
  840 WRITE(6,14) MM,MM,XYT(1,MM),XYT(2,MM),SCT(1,MM),SCT(2,MM),DST(MM)
C-----INTERMEDIATE NOSE REGION
      WRITE(6,52)
      CALL CAXIS(5,MMB,MMS,MMAN,MMCL,1-MMG,2*(MMT+1)-MMG,IAM,DAM,SAM,
     A CAM,MMT,XYM,SCM,DSM,LBMMAX)
C     OUTER NOSE REGION 4
      WRITE(6,51)
      CALL CAXIS(4,MHB,MHS,MHAN,MHCL,1-MHG,2*(MHT+1)-MHG,IAH,DAH,SAH,
     A CAH,MHT,XYH,SCH,DSH,LBHMAX)
C     FINE MESH REGION 3
      WRITE(6,43)
      CALL CAXIS(3,MFB,MFAT,MFAN,MFCL,1-MFG,2*(MFAT+3)-MFG,IAF,DAF,SAF,
     A CAF,MFT,XYF,SCF,DSF,LBFMAX)
C-----TRAILING EDGE SPECIAL AUXILIARIES
      M = MFAT
      DO 882 K = 1,2
      M = M + 1
      XA = FLOAT(M - MFAN)*DX(3)
      MM = M - MFG + 1
      XYF(1,MM) = XA
      XYF(2,MM) = 0.0
      DSF(MM) = 0.0
      SCF(1,MM) = 0.0
      SCF(2,MM) = 1.0
      MM = 2*(MFAT+3) - MFG - M
      XYF(1,MM) = XA
      XYF(2,MM) = 0.0
      DSF(MM) = 0.0
      SCF(1,MM) = 0.0
  882 SCF(2,MM) = -1.0
C-----OPERATING LIMIT NAMES, BOUNDARY POINTS
C-----INNER NOSE REGION
      LC = 1 - MTAN
      LBTA = MST + LC
      LBTB = LBTA + 1
      LBTC = MTQ + LC
      LBTD = MTS + LC
      LBTE = MTT + LC
      LC = 2*(MTT + 1) - MTAN
      LBTI = LC - MTT
      LBTJ = LC - MTS
```

```
      LBTK = LC - MTQ
      LBTM = LC - MSL
      LBTL = LBTM - 1
      LC = 2*(MTI - MTAN + 1)
      LBTR = NSL + LC
      LBTQ = LBTR - 1
      LBTS = NST + LC
      LBTT = LBTS + 1
      LBTMAX = 2*(MTI-MTAN+1) + NTI
C-----INTERMEDIATE NOSE REGION
      LC = 1 - MMG
      LBMA = MMG + LC
      LBMB = MMH + LC
      LBMC = MMQ + LC
      LBMD = MMS + LC
      LBME = MMT + LC
      LC = 2*(MMT + 1) - MMG
      LBMI = LC - MMT
      LBMJ = LC - MMS
      LBMK = LC - MMQ
      LBML = LC - MMH
      LBMM = LC - MMG
      LBMMAX = 2*(MMT - MMG + 1)
C     OUTER NOSE REGION 4
      LC=1-MHG    $LBHA=MHG+LC     LBHB=MHH+LC     LBHC=MHQ+LC
      LBHD=MHS+LC    $LBHE=MHT+LC     LC=2*(MHT+1)-MHG   $LBHI=LC-MHT
      LBHJ=LC-MHS    $LBHK=LC-MHQ     LBHL=LC-MHH     $LBHM=LC-MHG
      LBHMAX=LBHM
C     FINE MESH REGION 3
      LC = 1-MFG   $LBFA=MFG+LC   $LBFB=MFH+LC   $LBFC=MFAT+LC
      LBFD=LBFC+1   $LBFE=LBFC+2    LC=2*(MFAT+3)-MFG   $LBFK=LC-MFAT
      LBFJ=LBFK-1   $LBFI=LBFK-2    LBFL=LC-MFH    $LBFM=LC-MFG
      LBFMAX=LBFM
C-----ASSIGNMENTS OF CENTRAL AXIS DATA AT OVERLAPS
C-----INNER REGION OVERLAPS AT BLUNT PART OF NOSE
      KUT = 1
      LU = LBTT
      LK = LBTB
  890 DO 892 K = 1,2
      XYT(K,LU) = XYT(K,LK)
  892 SCT(K,LU) = SCT(K,LK)
      UST(LU) = UST(LK)
      GO TO (893,894,895,896),KUT
  893 KUT = 2    LU=LBTA    $LK=LBTS    GO TO 890
  894 KUT = 3
      LU = LBTQ
      LK = LBTL
      GO TO 890
  895 KUT = 4    $ LU=LBTM    $ LK=LBTR    $GO TO 890
C-----INNER - INTERMEDIATE OVERLAP
  896 KUT = 1
      LU = LBMA
      LK = LBTC
  898 DO 900 K = 1,2
      XYM(K,LU) = XYT(K,LK)
```

```
  900 SCM(K,LU) = SCT(K,LK)
      DSM(LU) = DST(LK)*DY(6)/DY(5)
      GO TO (902,904),KUT
  902 KUT = 2
      LU = LBMM
      LK = LBTK
      GO TO 898
  904 KUT = 1
      LU = LBTE
      LK = LBMB
  906 DO 908 K = 1,2
      XYT(K,LU) = XYM(K,LK)
  908 SCT(K,LU) = SCM(K,LK)
      DST(LU) = DSM(LK)*DY(5)/DY(6)
      GO TO (910,912),KUT
  910 KUT = 2
      LU = LBTI
      LK = LBML
      GO TO 906
C     INTERMEDIATE OUTER OVERLAP
  912 KUT=1    $LU=LBOA    $LK=LBMC
  914 DO 916 K=1,2         $XYM(K,LU)=XYM(K,LK)
  916 SCM(K,LU)=SCM(K,LK)    $DSM(LU)=DSM(LK)*DY(5)/DY(4)
      GO TO (918,920),KUT
  918 KUT=2    $LU=LBOM    $LK=LBMK    $GO TO 914
  920 KUT=1    $LU=LBME    $LK=LBMB
  922 DO 924 K=1,2         $XYM(K,LU)=XYM(K,LK)
  924 SCM(K,LU)=SCM(K,LK)    $DSM(LU)=DSM(LK)*DY(4)/DY(5)
      GO TO (926,928),KUT
  926 KUT=2    $LU=LBMI    $LK=LBML    $GO TO 922
C     OUTER FINE OVERLAP
  928 KUT=1    $LU=LBFA    $LK=LBMC
  930 DO 932 K=1,2         $XYF(K,LU)=XYM(K,LK)
  932 SCF(K,LU)=SCM(K,LK)    $DSF(LU)=DSM(LK)*DY(4)/DY(3)
      GO TO (934,936),KUT
  934 KUT=2    $LU=LBFM    $LK=LBMK    $GO TO 930
  936 KUT=1    $LU=LBME    $LK=LBFM
  938 DO 940 K=1,2         $XYM(K,LU)=XYF(K,LK)
  940 SCM(K,LU)=SCF(K,LK)    $DSM(LU)=DSF(LK)*DY(3)/DY(4)
      GO TO (942,944),KUT
  942 KUT=2    $LU=LBMI    $LK=LBFL    $GO TO 938
C-----AIRFOIL SURFACE POINTS
  944 WRITE(6,65)
      WRITE(6,35)
      WRITE(6,66)
      DO 1170 MM= 1,LBTMAX
 1170 WRITE(6,14) MM,MM,XYT(1,MM),XYT(2,MM),SCT(1,MM),SCT(2,MM),DST(MM)
C-----INTERMEDIATE NOSE REGION
      WRITE(6,52)
      WRITE(6,66)
      DO 1172 MM=1,LBMMAX
 1172 WRITE(6,14)MM,MM,XYM(1,MM),XYM(2,MM),SCM(1,MM),SCM(2,MM),DSM(MM)
C     OUTER NOSE REGION 4
      WRITE(6,51)
      WRITE(6,66)
```

```
      DO 1174 MM=1,LBHMAX
 1174 WRITE(6,14)MM,MM,XYH (1,MM),XYH(2,MM),SCH(1,MM),SCH(2,MM),DSH(MM)
C     FINE MESH REGION 3
      WRITE(6,43)
      WRITE(6,66)
      DO 1176 MM=1,LBFMAX
 1176 WRITE(6,14)MM,MM,XYF(1,MM),XYF(2,MM),SCF(1,MM),SCF(2,MM),DSF(MM)
C-----END OF PRINTOUT
C-----INTERMEDIATE NOSE REGION
      KR = 6
      KUT = 1
      MLL = LBTB
      MUL = LBTD
      KSPL = 0
      XTZ = DX(6)*FLOAT(MTA-MTAN)      $YTZ = DY(6)*FLOAT(NTA-NTCL)
  950 CALL PONDER(   KSPL,MLL,MUL,XTZ,YTZ,XYT,SCT,DST,DWT,DWTX,AWT,
     A CWT,IWT,LBTMAX,MTT)
      GO TO (952,954,956),KUT
  952 KUT = 2        $MLL=LBTJ    MUL=LBTL    $GO TO 950
  954 KUT = 3        $MLL=LBTR    MUL=LBTS    $GO TO 950
  956 WRITE(6,67)
      WRITE(6,35)
      WRITE(6,68)
      DO 1178 MM= 1,LBTMAX
      DO 1178 LIP = 1,4
 1178 WRITE(6,14)MM,LIP,(DWT(K,LIP,MM),K=1,9)
C-----SPECIAL INTERPOLATION WEIGHTS AT END OF RANGE, (DWTX)
      WRITE(6,90)
      DO 1179 KUT = 1,6
      DO 1179 LIP = 1,2
 1179 WRITE(6,14)KUT,LIP,(DWTX(K,LIP,KUT),K=1,9)
C-----INTERPOLATION WEIGHTS, SURFACE NORMALS TO CARTESIAN
      WRITE(6,69)
      WRITE(6,35)
      WRITE(6,70)
      DO 1190 MM= 1,LBTMAX
      IF(MM.LE.LBTE) GO TO 1186
      IF(MM.LE.LBTM) GO TO 1188
      N = MM - 2*(MTT - MTAN + 1)
      M = IAN(N)
      GO TO 1190
 1186 M = MM + MTAN - 1
      N = IAT(2,M)
      GO TO 1190
 1188 M = 2*MTT + 2 - MTAN - MM
      N = IAT(1,M)
 1190 WRITE(6,16)MM,M,N,(AWT(K,MM),K=1,3)
C-----INTERPOLATION WEIGHTS, CARTESIOA TO ROTATED MESH
      WRITE(6,71)
      WRITE(6,35)
      WRITE(6,72)
      DO 1204 MM= 1,LBTMAX
      DO 1204 LIP = 1,3
 1204 WRITE(6,16)MM,LIP,IWT(LIP,MM),(CWT(K,LIP,MM),K=1,9)
C-----END OF PRINTOUT
```

```
      KR = 5    $KUT = 1    $MLL=LBMH    $MUL=LBMD    $KSPL=0
      XTZ = DX(5)*FLOAT(MMA-MMAN)         $YTZ = DY(5)*FLOAT(NMA-NMCL)
  960 CALL PONDER(   KSPL,MLL,MUL,XTZ,YTZ,XYM,SCM,DSM,UWM,UWMX,AWM,
     A CWM,IWM,LBMMAX,MMT)
      GO TO (962,968),KUT
  962 CALL WAIF(KR,KUT,LBMA,XTZ,YTZ,XYM,SCM,CWMX,IWMX,LBMMAX)
      KUT = 2      $MLL=LBMJ    MUL=LBML    $GO TO 960
  968 CALL WAIF(KR,KUT,LBMM,XTZ,YTZ,XYM,SCM,CWMX,IWMX,LBMMAX)
C-----INTERMEDIATE NOSE REGION
      WRITE(6,52)
      WRITE(6,68)
      DO 1180 MM=1,LBMMAX
      DO 1180 LIP=1,4
 1180 WRITE(6,14)MM,LIP,(DWM(K,LIP,MM),K=1,9)
      WRITE(6,91)
      DO 1181 KUT = 1,6
      DO 1181 LIP=1,2
 1181 WRITE(6,14)KUT,LIP,(UWMX(K,LIP,KUT),K=1,9)
C-----INTERMEDIATE NOSE REGION
      WRITE(6,52)
      WRITE(6,70)
      DO 1194 MM=1,LBMMAX
      IF(MM.LE.LBME)GO TO 1192    $M=2*(MMT+1)-MMG-MM
      N = IAM(1,M)    $GO TO 1194
 1192 M = MM - 1 + MMG      $N = IAM(2,M)
 1194 WRITE(6,16)MM,M,N,(AWM(K,MM),K=1,3)
      WRITE(6,52)
      WRITE(6,72)
      DO 1206 MM=1,LBMMAX
      DO 1206 LIP=1,3
 1206 WRITE(6,16)MM,LIP,IWM(LIP,MM),(CWM(K,LIP,MM),K=1,9)
      WRITE(6,52)
      WRITE(6,22)
      DO 1223 KUT = 1,2
      DO 1223 LIP = 1,2
 1223 WRITE(6,16)KUT,LIP,IWMX(LIP,KUT),(CWMX(K,LIP,KUT),K=1,9)
C-----END OF PRINTOUT
C     OUTER NOSE REGION 4
      KR = 4       $KUT=1       MLL=LBHH     MUL=LBHD    $KSPL=0
      XTZ = DX(4)*FLOAT(MHA-MHAN)       $YTZ = DY(4)*FLOAT(NHA-NHCL)
  970 CALL PONDER(   KSPL,MLL,MUL,XTZ,YTZ,XYH,SCH,DSH,UWH,UWHX,AWH,
     A CWH,IWH,LBHMAX,MHT)
      GO TO (972,978),KUT
  972 CALL WAIF(KR,KUT,LBHA,XTZ,YTZ,XYH,SCH,CWHX,IWHX,LBHMAX)
      KUT = 2      $MLL=LBHJ    MUL=LBHL    GO TO 970
  978 CALL WAIF(KR,KUT,LBHM,XTZ,YTZ,XYH,SCH,CWHX,IWHX,LBHMAX)
C     OUTER NOSE REGION 4
      WRITE(6,51)
      WRITE(6,68)
      DO 1182 MM=1,LBHMAX
      DO 1182 LIP=1,4
 1182 WRITE(6,14)MM,LIP,(DWH(K,LIP,MM),K=1,9)
      WRITE(6,92)
      DO 1183 KUT = 1,6
      DO 1183 LIP=1,2
```

```
 1183 WRITE(6,14)KUT,LIP,(UWHX(K,LIP,KUT),K=1,9)
C     OUTER NOSE REGION 4
      WRITE(6,51)
      WRITE(6,70)
      DO 1198 MM=1,LBHMAX
      IF(MM.LE.LBHE)GO TO 1196   $M=2*(MHT+1)-MHG-MM
      N = IAH(1,M)    $GO TO 1198
 1196 M = MM - 1 + MHG      $N = IAH(2,M)
 1198 WRITE(6,16)MM,M,N,(AWH(K,MM),K=1,3)
C     OUTER NOSE REGION 4
      WRITE(6,51)
      WRITE(6,72)
      DO 1208 MM=1,LBHMAX
      DO 1208 LIP=1,3
 1208 WRITE(6,16)MM,LIP,IWH(LIP,MM),(CWH(K,LIP,MM),K=1,9)
      WRITE(6,51)
      WRITE(6,24)
      DO 1224 KUT=1,2
      DO 1224 LIP=1,2
 1224 WRITE(6,16)KUT,LIP,IWHX(LIP,KUT),(CWHX(K,LIP,KUT),K=1,9)
C-----END OF PRINTOUT
C     FINE MESH REGION 3
      KR = 3       $KUT=1        MLL=LBFH     MUL=LBFD    $KSPL=0
      XTZ = DX(3)*FLOAT(MFA-MFAH)       $YTZ = DY(3)*FLOAT(NFA-NFCL)
  980 CALL PONDER(   KSPL,MLL,MUL,XTZ,YTZ,XYF,SCF,DSF,UWF,UWFX,AWF,
     A CWF,IWF,LBFMAX,MFT)
      GO TO (982,988),KUT
  982 CALL WAIF(KR,KUT,LBFA,XTZ,YTZ,XYF,SCF,CWFX,IWFX,LBFMAX)
      KUT = 2      $MLL=LBFJ    MUL=LBFL    $GO TO 980
  988 CALL WAIF(KR,KUT,LBFM,XTZ,YTZ,XYF,SCF,CWFX,IWFX,LBFMAX)
C     FINE MESH REGION 3
      WRITE(6,43)
      WRITE(6,68)
      DO 1184 MM=1,LBFMAX
      DO 1184 LIP=1,4
 1184 WRITE(6,14)MM,LIP,(UWF(K,LIP,MM),K=1,9)
      WRITE(6,53)
      DO 1185 KUT = 1,6
      DO 1185 LIP=1,2
 1185 WRITE(6,14)KUT,LIP,(UWFX(K,LIP,KUT),K=1,9)
C     FINE MESH REGION 3
      WRITE(6,43)
      WRITE(6,70)
      DO 1202 MM=1,LBFMAX
      IF(MM.LE.LBFE)GO TO 1200   $M=2*(MFAT+3)-MFG-MM
      N = IAF(1,M)    $GO TO 1202
 1200 M = MM - 1 + MFG      $N = IAF(2,M)
 1202 WRITE(6,16)MM,M,N,(AWF(K,MM),K=1,3)
C     FINE MESH REGION 3
      WRITE(6,43)
      WRITE(6,72)
      DO 1210 MM=1,LBFMAX
      DO 1210 LIP=1,3
 1210 WRITE(6,16)MM,LIP,IWF(LIP,MM),(CWF(K,LIP,MM),K=1,9)
      WRITE(6,43)
```

```
      WRITE(6,26)
      DO 1225 KUT=1,2
      DO 1225 LIP=1,2
 1225 WRITE(6,16)KUT,LIP,IWFX(LIP,KUT),(CWFX(K,LIP,KUT),K=1,9)
C-----END OF PRINTOUT
C-----SPECIAL INTERPOLATIONS AT FIELD BOUNDARIES
C-----ASSIGNMENTS TO FINER MESH SURFACE NORMALS
 1140 DOY = DY(6)/DY(5)
      DAY = 1.0 + DOY
      DO 1142 LIP = 1,3
      DAY = DAY - DOY
      AWMT(1,LIP)  = 0.5*DAY*(DAY + 1.0)
      AWMT(2,LIP)  = 1.0 - DAY**2
 1142 AWMT(3,LIP)  = AWMT(1,LIP) - DAY
C-----SPECIAL BOUNDARY INTERPOLATIONS
      WRITE(6,73)
      WRITE(6,70)
      DO 1212 LIP = 1,3
 1212 WRITE(6,16)LIP,LIP,LIP,(AWMT(K,LIP),K=1,3)
C-----END OF PRINTOUT
      DOY = DY(5)/DY(4)        $DAY = 1.0 + DOY
      DO 1144 LIP=1,3    $DAY=DAY-DOY    $AWHM(1,LIP)=0.5*DAY*(DAY+1.0)
      AWHM(2,LIP) = 1.0 - DAY**2
 1144 AWHM(3,LIP) = AWHM(1,LIP) - DAY
      WRITE(6,74)
      WRITE(6,70)
      DO 1214 LIP=1,3
 1214 WRITE(6,14)LIP,LIP,(AWHM(K,LIP),K=1,3)
C-----END OF PRINTOUT
      DOY = DY(4)/DY(3)        $DAY = 1.0 + DOY
      DO 1146 LIP=1,3    $DAY=DAY-DOY    $AWFH(1,LIP)=0.5*DAY*(DAY+1.0)
      AWFH(2,LIP) = 1.0 - DAY**2
 1146 AWFH(3,LIP) = AWFH(1,LIP) - DAY
      WRITE(6,75)
      WRITE(6,70)
      DO 1216 LIP=1,3
 1216 WRITE(6,14)LIP,LIP,(AWFH(K,LIP),K=1,3)
C-----END OF PRINTOUT
C-----ASSIGNMENTS TO FINER MESH CARTESIAN BOUNDARY POINTS
      LBMP = LBMP
      KR = 5
      KUT = 1
      CALL DISTW(1,0.0,0.0,0.0,XYM,SCM,LBMMAX)
      LOP = IFIX(DY(5)/DY(6) + 0.001) - 1
      XP = DX(6)*FLOAT(MTT - MTAN)
      YP = DY(5)*FLOAT(JAM(2,MMH) - NMCL)
      YPI = YP - DY(5)*DAM(2,MMH)
      MP = MTT
      NP = (JAM(2,MMH) - NMCL)*2 + NTCL
      DO 1148 LIP = 1,LOP
      YP = YP - DY(6)
      NP = NP - 1
      IWMT(KUT,LIP) = 100*MP + NP
      IF(YP.GE.YPI) LINTM(KUT) = LIP
 1148 CALL DISTW(2,XP,YP,AWMT(1,KUT,LIP),XYM,SCM,LBMMAX)    $KUT=2
```

```
      YP = DY(5)*FLOAT(IAM(1,MMH) - NMCL)
      NP = (IAM(1,MMH) - NMCL)*2 + NTCL
      YPT = YP + DY(5)*DAM(1,MMH)
      M = LBML
      CALL DISTW(1,0.0,0.0,0.0,XYM,SCM,LBMMAX)
      DO 1150 LIP = 1,LOP
      YP = YP + DY(6)
      MP = MP + 1
      IWMI(KUT,LIP) = 100*MP + NP
      IF(YP.LE.YPT) LINTM(KUT) = LIP
 1150 CALL DISTW(2,XP,YP,D,MT(1,KUT,LIP),XYM,SCM,LBMMAX)
      WRITE(6,76)
      DO 1218 KUT = 1,2
      DO 1218 LIP = 1,4
 1218 WRITE(6,16) KUT, LIP, IWMI(KUT,LIP), (DWMT(K,KUT,LIP),K=1,9)
      WRITE(6,94) LINTM(1), LINTM(2)
      M = LBHH     $KUT = 1      KR = 4
      CALL DISTW(1,0.0,0.0,0.0,XYH,SCH,LBHMAX)
      LOP = IFIX(DY(4)/DY(5)+0.001)-1   $XP=DX(5)*FLOAT(MMT-MMAN)
      YP = DY(4)*FLOAT(IAH(2,MHH)-NHCL)   ,YPT=YP-DY(4)*DAH(2,MHH)
      MP = MMT
      NP = (IAH(2,MHH) - NHCL)*5 + NMCL
      DO 1152 LIP=1,LOP   ,YP=YP-DY(5)   ,IF(YP.GE.YPT)LINTH(KUT)=LIP
      MP = MP - 1
      IWHH(KUT,LIP) = 100*MP + NP
 1152 CALL DISTW(2,XP,YP,D,HM(1,KUT,LIP),XYH,SCH,LBHMAX)    $KUT=2
      M=LBHL      YP=DY(4)*FLOAT(IAH(1,MHH)-NHCL) $YPT=YP+DY(4)*DAH(1,MHH)
      NP = (IAH(1,MHH) - NHCL)*5 + NMCL
      CALL DISTW(1,0.0,0.0,0.0,XYH,SCH,LBHMAX)
      DO 1154 LIP=1,LOP   ,YP=YP+DY(5)   ,IF(YP.LE.YPT)LINTH(KUT)=LIP
      MP = MP + 1
      IWHH(KUT,LIP) = 100*MP + NP
 1154 CALL DISTW(2,XP,YP,D,HM(1,KUT,LIP),XYH,SCH,LBHMAX)
      WRITE(6,77)
      DO 1220 KUT=1,2
      DO 1220 LIP=1,4
 1220 WRITE(6,16) KUT, LIP, IWHH(KUT,LIP), (DWHM(K,KUT,LIP),K=1,9)
      WRITE(6,95) LINTH(1), LINTH(2)
      M = LBFP      $KUT = 1     $KR = 3
      CALL DISTW(1,0.0,0.0,0.0,XYF,SCF,LBFMAX)
      LOP = IFIX(DY(3)/DY(4)+0.001)-1   $XP=DX(4)*FLOAT(MHT-MHAN)
      YP = DY(3)*FLOAT(IAF(2,MFH)-NFCL)   ,YPT=YP-DY(3)*DAF(2,MFH)
      MP = MHT
      NP = (IAF(2,MFH) - NFCL)*4 + NHCL
      DO 1156 LIP=1,LOP   ,YP=YP-DY(4)   ,IF(YP.GE.YPT)LINTF(KUT)=LIP
      MP = MP - 1
      IWFH(KUT,LIP) = 100*MP + NP
 1156 CALL DISTW(2,XP,YP,D,FH(1,KUT,LIP),XYF,SCF,LBFMAX)     $KUT=2
      M=LBFL      YP=DY(3)*FLOAT(IAF(1,MFH)-NFCL) $YPT=YP+DY(3)*DAF(1,MFH)
      NP = (IAF(1,MFH) - NFCL)*4 + NHCL
      CALL DISTW(1,0.0,0.0,0.0,XYF,SCF,LBFMAX)
      DO 1158 LIP=1,LOP   $YP=YP+DY(4)   IF(YP.LE.YPT)LINTF(KUT)=LIP
      MP = MP + 1
      IWFH(KUT,LIP) = 100*MP + NP
 1158 CALL DISTW(2,XP,YP,D,FH(1,KUT,LIP),XYF,SCF,LBFMAX)
```

```
      WRITE(6,78)
      DO 1222 KUT=1,2
      DO 1222 LIP=1,4
 1222 WRITE(6,16) KUT, LIP, IWFH(KUT,LIP), (DWFH(K,KUT,LIP),K=1,9)
      WRITE(6,96) LINTE(1), LINTE(2)
C-----WORKING VARIABLES
  600 READ(5,4) UMACH, GAM, ALPHA
      WRITE(6,48)
      WRITE(6,4) UMACH, GAM, ALPHA
      ALPHAR = ALPHA/180.0*PI
      WSU(3) = 1.0
      WSU(1) = UMACH*COS(ALPHAR)
      WSU(2) = UMACH*SIN(ALPHAR)
C-----USEFUL CONSTANTS
      HGMU = 0.5*(GAM-1.0)
      WDYN1 = 1.0/(0.7*UMACH**2)
      HINF = 1.0/(GAM - 1.0) + 0.5*UMACH**2
      GMU = GAM - 1.0
      UNOGMU = 1.0/GMU
      UNOG = 1.0/GAM
      UOGMU = 1.0/HGMU
C-----ASSIGN WORKING VARIABLES TO START NEW PROBLEM
C-----OUTER POLAR COORDINATE FIELD
  610 DO 612 M = 1,MYT
      DO 612 N = 1,NYT
      DO 612 K = 1,3
  612 WY(K,M,N) = WSU(K)
C-----COARSE CARTESIAN MESH
  620 DO 622 M = 1,MCT
      DO 622 N = 1,NCT
      DO 622 K = 1,3
  622 WC(K,M,N) = WSU(K)
C-----FINE MESH
  630 DO 632 M = 1,MFT
      DO 632 N = 1,NFT
      DO 632 K = 1,3
  632 WF(K,M,N) = WSU(K)
C-----OUTER NOSE REGION
  640 DO 642 M = 1,MHT
      DO 642 N = 1,NHT
      DO 642 K = 1,3
  642 WH(K,M,N) = WSU(K)
C-----INTERMEDIATE NOSE REGION
  650 DO 652 M = 1,MMT
      DO 652 N = 1,NMT
      DO 652 K = 1,3
  652 WM(K,M,N) = WSU(K)
C-----INNER NOSE REGION
  660 DO 662 M = 1,MIT
      DO 662 N = 1,NIT
      DO 662 K = 1,3
  662 WI(K,M,N) = WSU(K)
C-----BOUNDARY POSTAGE STAMP MESHES
      DO 660 J = 1,9
C-----FINE MESH
```

```
      DO 670 LL = 1,LBFMAX
      WSF(3,J,LL) = WSU(3)
      WSF(1,J,LL) = WSU(1)*SCF(2,LL) + WSU(2)*SCF(1,LL)
      WSF(2,J,LL) =-WSU(1)*SCF(1,LL) + WSU(2)*SCF(2,LL)
      IF((SCF(1,LL).EQ.0.0).AND.(SCF(2,LL).EQ.0.0))WSF(1,J,LL)=WSU(1)
  670 CONTINUE
C-----OUTER NOSE REGION
      DO 672 LL = 1,LBHMAX
      WSH(3,J,LL) = WSU(3)
      WSH(1,J,LL) = WSU(1)*SCH(2,LL) + WSU(2)*SCH(1,LL)
      WSH(2,J,LL) =-WSU(1)*SCH(1,LL) + WSU(2)*SCH(2,LL)
      IF((SCH(1,LL).EQ.0.0).AND.(SCH(2,LL).EQ.0.0))WSH(1,J,LL)=WSU(1)
  672 CONTINUE
C-----INTERMEDIATE NOSE REGION
      DO 674 LL = 1,LBMMAX
      WSM(3,J,LL) = WSU(3)
      WSM(1,J,LL) = WSU(1)*SCM(2,LL) + WSU(2)*SCM(1,LL)
      WSM(2,J,LL) =-WSU(1)*SCM(1,LL) + WSU(2)*SCM(2,LL)
      IF((SCM(1,LL).EQ.0.0).AND.(SCM(2,LL).EQ.0.0))WSM(1,J,LL)=WSU(1)
  674 CONTINUE
C-----INNER NOSE REGION
      DO 676 LL = 1,LBTMAX
      WST(3,J,LL) = WSU(3)
      WST(1,J,LL) = WSU(1)*SCT(2,LL) + WSU(2)*SCT(1,LL)
      WST(2,J,LL) =-WSU(1)*SCT(1,LL) + WSU(2)*SCT(2,LL)
      IF((SCT(1,LL).EQ.0.0).AND.(SCT(2,LL).EQ.0.0))WST(1,J,LL)=WSU(1)
  676 CONTINUE
  680 CONTINUE
C-----WRITE NEW STARTING TAPE DATA IN BINARY
      L = 1
      WRITE(4)        MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT,
     C MCA, MCB, MCE, MCF, MCG, MCH, MCS, MCT, NCA, NCB, NCE, NCF,
     C NCG, NCH, NCS, NCT,
     F MFA, MFB, MFD, MFE, MFF, MFG, MFH, MFQ, MFS, MFT, MFAN,
     F MFANP, MFAT, MFATM, MFATP, NFA, NFB, NFD, NFE, NFF, NFG, NFH,
     F NFQ, NFS, NFT, NFCL,
     H MHA, MHB, MHD, MHE, MHF, MHG, MHH, MHQ, MHS, MHT, MHAN, MHANP,
     H NHA, NHB, NHD, NHE, NHF, NHG, NHH, NHQ, NHS, NHT, NHCL,
     M MMA, MMB, MMD, MME, MMF, MMG, MMH, MMQ, MMS, MMT, MMAN, MMANP,
     M NMA, NMB, NMD, NME, NMF, NMG, NMH, NMQ, NMS, NMT, NMCL,
     T MTA, MTB, MTD, MTQ, MTS, MTT, MTAN, MTANP, NTA, NTB, NTD,
     T NTQ, NTS, NTT, NTCL,
     S MSL, MST, NSL, NST
      WRITE(4)           LBFA, LBFB, LBFC, LBFD, LBFE, LBFI, LBFJ, LBFK,
     F LBFL, LBFM, LBFMAX,
     H LBHA, LBHB, LBHC, LBHD, LBHE, LBHI, LBHJ, LBHK, LBHL, LBHM,
     H LBHMAX,
     M LBMA, LBMB, LBMC, LBMD, LBME, LBMI, LBMJ, LBMK, LBML, LBMM,
     M LBMMAX,
     T LBTA, LBTB, LBTC, LBTD, LBTE, LBTI, LBTJ, LBTK, LBTL, LBTM,
     T LBTQ, LBTR, LBTS, LBTT, LBTMAX
      WRITE(4)        L
      WRITE(4)     (IYB(J), J = 1,84), (ICB(J), J=1,136),
     A ((IUFH(J), IUHM(J), IUMT(J)), J=1,10), (IWF(J), J=1,126),
     B (IWH(J), J=1,66), (IWM(J), J=1,42), (IWT(J), J=1,129),
```

```
     C ((LIUTF(J), LIUTH(J), LINIM(J)), J=1,21), ((IWFX(J), IWHX(J),
     D IWMX(J)), J=1,4), (IAF(J), J=1,74), (IAH(J), J=1,50), (IAM(J),
     E J=1,52), (IAT(J), J=1,32), (IAN(J), J=1,21),
     F ((IWFH(J), IWHM(J), IWMT(J)), J=1,8)
      WRITE(4)         ((ICO(J), JCO(J)), J = 1,68), ((IFO(J), JFO(J)),
     F J=1,296), ((IHO(J),JHO(J)),J=1,200), ((IMO(J),JMO(J)),J=1,208),
     G ((ITO(J),JTO(J)), J=1,128)
      WRITE(4)         ((DX(J),DY(J)),J=1,6), (RH(J), J=1,9)
      WRITE(4)         ((CY(J),SY(J)), J=1,5 ), (DYH(J), J=1,84),
     A (RCU(J), J=1,136), ((XYF(J),SCF(J)), J=1,84),
     B ((XYH(J),SCH(J)), J=1,44), ((XYM(J),SCM(J)), J=1,28),
     C ((XYT(J),SCT(J)), J=1,86)
      WRITE(4)         (DWF(J), J=1,1512), (DWH(J), J=1,792),
     A (DWM(J), J=1,504), (DWT(J), J=1,1548), (AWF(J), J=1,126),
     B (AWH(J), J=1,66), (AWM(J), J=1,42), (AWT(J), J=1,129),
     C (CWF(J), J=1,1134), (CWH(J), J=1,594), (CWM(J), J=1,378),
     D(CWT(J),J=1,1161), ((DWFX(J),DWHX(J),DWMX(J),DWTX(J)),J=1,108),
     E ((AWFH(J), AWHM(J), AWMT(J)), J=1,9),
     F ((DWFH(J), DWHM(J), DWMT(J)), J=1,72),
     G ((CWFX(J), CWHX(J), CWMX(J)), J=1,36)
      WRITE(4)      GAM, HGMU, UNOG, DUGMU, OCNMI, HINF, GMU, UNOGMU
      WRITE(4)    UMACH, ALPHA, RBNDY
      WRITE(4)         (WY(J), J=1,1134), (WC(J), J=1,867),
     A (WF(J), J=1,1443), (WH(J), J=1,1275), (WM(J), J=1,2028),
     B (WT(J), J=1,1068), (WSF(J), J=1,1134), (WSH(J), J=1,594),
     C (WSM(J), J=1,378), (WST(J), J=1,1161), (WSU(J), J=1,3)
      CALL EXIT
      STOP
    2 FORMAT(16I5)
    4 FORMAT(8E10.3)
    8 FORMAT(6E20.8)
   10 FORMAT(I5,2I10,5X,2E20.8)
   12 FORMAT(I5,5X,2I4,3X,2I4,3X,2I4,3X,2I4,3X,2I4,3X,2I4,3X,2I4,3X,
     A 2I4,3X,2I4,3X,2I4,3X,2I4)
   14 FORMAT(2I5,12F10.5)
   16 FORMAT(I5,5X,2I5,11F10.5)
C-----TITLES AND HEADINGS
   22 FORMAT(2X,3HKUT,6X,3HLIP,2X,5HM  N,5X,7HCWMX(K))
   23 FORMAT(/20X,15HYTRE BNDY. RING)
   24 FORMAT(2X,3HKUT,6X,3HLIP,2X,5HM  N,5X,7HCWHX(K))
   25 FORMAT(3X,2HMY,8X,2HMC,8X,2HNC,17X,4HDELX,16X,4HDELY)
   26 FORMAT(2X,3HKUT,6X,3HLIP,2X,5HM  N,5X,7HCWFX(K))
   27 FORMAT(/20X,18H LEFT COARSE BNDY.)
   28 FORMAT(/20X,18HRIGHT COARSE BNDY.)
   29 FORMAT(/20X,18HBOTT. COARSE BNDY.)
   30 FORMAT(/20X,18H  TOP COARSE BNDY.)
   31 FORMAT(3X,2HNC,8X,2HMY,8X,2HNY,17X,4HDLTH,16X,7H-DELRHO)
   32 FORMAT(3X,2HMC,8X,2HMY,8X,2HNY,17X,4HDLTH,16X,7H-DELRHO)
   33 FORMAT(///10X,13HAIRFOIL SHAPE)
   34 FORMAT(/9X,1HX,18X,3HYUS,17X,4HYPSU,16X,3HYLS,17X,4HYPSL)
   35 FORMAT(//10X,17HINNER NOSE REGION)
   36 FORMAT(///9X,1HY,17X,5HXNOSE,14X,6HYPNOSE)
   38 FORMAT(//10X,36HINNER NOSE REGION OPERATIONAL LIMITS)
   39 FORMAT(//10X,30H FINE MESH  OPERATIONAL LIMITS)
   40 FORMAT(//10X,21HAIRFOIL INTERPOLATION)
```

```
   41 FORMAT(1X,4HCOL.,5X,3HIMO,2X,3HJMO)
   42 FORMAT(1X,4HCOL.,5X,3HIFO,2X,3HJFO)
   43 FORMAT(/10X,16HFINE MESH REGION)
   44 FORMAT(/10X,31HINNER NOSE REGION VERTICAL CUTS)
   45 FORMAT(/10X,31HINNER NOSE REGION HORIZON, CUTS)
   46 FORMAT(4X,1HM,9X,1HN,9X,1HN,13X,4HUELY,16X,4HDELY)
   47 FORMAT(4X,1HN,9X,1HM,9X,1HM,13X,4HUELX)
   48 FORMAT(/7X,5HUMACH,17X,3HGAM,16X,5HALPHA)
   49 FORMAT(1X,7HIDEH(1))
   50 FORMAT(2X,3HMSL,2X,3HMST,2X,3HNSL,2X,3HNST)
   51 FORMAT(//10X,17HOUTER NOSE REGION)
   52 FORMAT(//10X,24HINTERMEDIATE NOSE REGION)
   53 FORMAT(/4X,1HM,4X,1HN,7X,3HRHO,8X,2HU1,8X,2HU2)
   54 FORMAT(/3X,2HMM,2X,3HJCP,7X,3HRHO,8X,2HUS,8X,2HUN)
   55 FORMAT(1H0,10X,*COARSE MESH OPERATIONAL LIMITS*)
   60 FORMAT(1X,4HCOL.,5X,3HICO,2X,3HJCO)
   61 FORMAT(1X,4HCOL.,5X,3HITO,2X,3HJTO)
   62 FORMAT(1X,4HCOL.,5X,3HIHO,2X,3HJHO)
   63 FORMAT(1X,7HIDHM(1))
   64 FORMAT(1X,7HIDMT(1))
   65 FORMAT(/5X,39HAIRFOIL SURFACE POINT LOCI, ORIENTATION)
   66 FORMAT(3X,2HMM,3X,2HNM,9X,1HX,9X,1HY,7X,3HSIN,7X,3HCOS,4X,6HOFFSE
     AT)
   67 FORMAT(/5X,48HINTERPOLATION WEIGHTS, DISTORTED TO ROTATED MESH)
   68 FORMAT(3X,2HMM,2X,3HLIP,6X,4HW(N))
   69 FORMAT(/5X,51HINTERPOLATION WEIGHTS, SURFACE NORMALS TO CARTESIAN)
   70 FORMAT(3X,2HMM,1X,5HUN 10,4X,1HM,4X,1HN,6X,4HW(2),6X,
     A4HW(5),6X,4HW(8))
   71 FORMAT(/5X,48HINTERPOLATION WEIGHTS, CARTESION TO ROTATED MESH)
   72 FORMAT(3X,2HMM,7X,3HLIP,2X,3HM N,6X,4HW(N))
   73 FORMAT(/5X,11HAWMT(K,LIP))
   74 FORMAT(/5X,11HAWHM(K,LIP))
   75 FORMAT(/5X,11HAWFH(K,LIP))
   76 FORMAT(/2X,3HKUT,12X,3HLIP,1X,4HIWMT,2X,15HDWMT(K,KUT,LIP))
   77 FORMAT(/2X,3HKUT,12X,3HLIP,1X,4HIWHM,2X,15HDWHM(K,KUT,LIP))
   78 FORMAT(/2X, HKUT,12X,3HLIP,1X,4HIWFH,2X,15HDWFH(K,KUT,LIP))
   80 FORMAT(16I5)
   82 FORMAT(4E20.13)
   90 FORMAT(/2X,3HKUT,2X,3HLIP,5X,7HDWTX(N))
   91 FORMAT(/2X,3HKUT,2X,3HLIP,5X,7HDWMX(N))
   92 FORMAT(/2X,3HKUT,2X,3HLIP,5X,7HDWHX(N))
   93 FORMAT(/2X,3HKUT,2X,3HLIP,5X,7HDWFX(N))
   94 FORMAT(/5X,9HLINTM(1)=,I5,10X,9HLINTM(2)=,I5)
   95 FORMAT(/5X,9HLINTH(1)=,I5,10X,9HLINTH(2)=,I5)
   96 FORMAT(/5X,9HLINTF(1)=,I5,10X,9HLINTF(2)=,I5)
      END
```

```
      SUBROUTINE SHAPE(NSHP,CLINE)
C-----ARBITRARY AIRFOIL GIVEN BY TABLES OF SURFACE POINTS
C     SPLINE FIT TO POINTS IN EACH OF THREE REGIONS
C-----REGION 1  UPPER SURFACE , POINTS ARRANGED MONOTONIC IN X FROM L.E. T
C     REGION 2  LOWER SURFACE   POINTS ARRANGED MONOTONIC IN X FROM L.E. T
C     REGION3 NOSE, POINTS ARRANGED MONOTONIC IN Y FROM LWER TO UPPER SIDE
C     ALL X,Y IN ORDINARY AIRFOIL COORDINATES
C     LAST 2 POINTS OF REGION 3 TO COINCIDE WITH FIRST 2 OF REGION 1
C     FIRST POINTS 1,2 OF REGION 3 TO COINCIDE WITH POINTS 2,1 OF REGION 2
C     NTE(J) = NUMBER OF TABLE ENTRIES IN REGION J
C     NSHP = 1  SET UP SPLINE FIT, WRITE FITTED DATA
C     NSHP = 2  VERTICAL CUT, GET BOTH UPPER AND LOWER SURFACE
C     NSHP = 3  HORIZONTAL CUT IN NOSE REGION
C     NSHP = 4  VERTICAL CUT TO GET LOWER SURFACE
C     NSHP = 5  VERTICAL CUT TO GET UPPER SURFACE
C     CVRT = CONVERGENCE TOLERANCE ON X OR Y
      COMMON/AIRF/YUS,YLS,YPSU,YPSL,XNOSE,YPNOSE,RADCU,RADCL,RADC
      DIMENSION X(101,3), Y(101,3), H(101,3), DELY(101,3), H2(101,3),
     A B(101,3), DELSQY(101,3), S1(101,3), S2(101,3), S3(101,3),
     B C(101,3), S4(3), NTE(3), NTD(3), TITLE(8)
      GO TO (70,200,400,300,200),NSHP
   70 READ(5,1) (TITLE(I), I = 1,8)
      WRITE(6,3) (TITLE(I), I = 1,8)
      READ(5,2) CVRT, EPSLN
      WRITE(6,18) EPSLN
      READ(5,4) (NTE(J), J = 1,3)
      DO 82 KR = 1,2
      NT = NTE(KR)
      NTD(KR) = NTE(KR) - 1
      DO 82 I = 1,NT
   82 READ(5,2) X(I,KR), Y(I,KR)
      KR = 3
      NT = NTE(KR)
      NTD(KR) = NTE(KR) - 1
      DO 84 I = 1,NT
   84 READ(5,2) Y(I,KR), X(I,KR)
      DO 90 KR = 1,3
      NS = NTD(KR)
      DO 88 I = 1,NS
      H(I,KR) = X(I+1,KR) - X(I,KR)
   88 DELY(I,KR) = (Y(I+1,KR) - Y(I,KR))/H(I,KR)
      DO 90 I = 2,NS
      H2(I,KR) = H(I-1,KR) + H(I,KR)
      B(I,KR) = 0.5*H(I-1,KR)/H2(I,KR)
      DELSQY(I,KR) = (DELY(I,KR) - DELY(I-1,KR))/H2(I,KR)
      S2(I,KR) = 2.0*DELSQY(I,KR)
   90 C(I,KR) = 3.0*DELSQY(I,KR)
      LA = 1    $LB = 2    $LC = NTD(1)    $LD = NTE(1)
      LE = NTE(2)    $LF = NTD(2)    $LG = 2    $LH = 1
      LI = 1    $LJ = 2    $LK = NTD(3)    $LM = NTE(3)
      S2(LD,1) = 0.0
      S2(LE,2) = 0.0
      I = LB    $KR = 1
      S1(I,KR) = DELY(I,KR) - H(I,KR)*(2.0*S2(I,KR)+S2(I+1,KR))/6.0
```

```
      I = LG    $KR = 2
      S1(I,KR) = DELY(I,KR) - H(I,KR)*(2.0*S2(I,KR)+S2(I+1,KR))/6.0
      I = LJ    $KR = 3
      S1(I,KR) = DELY(I,KR) - H(I,KR)*(2.0*S2(I,KR)+S2(I+1,KR))/6.0
      I = LK    $KR = 3
      S1(I,KR) = DELY(I-1,KR)+H(I-1,KR)*(S2(I-1,KR)+2.0*S2(I,KR))/6.0
      S1(LI,3) = 1.0/S1(LG,2)
      S1(LH,2) = 1.0/S1(LJ,3)
      S1(LM,3) = 1.0/S1(LB,1)
      S1(LA,1) = 1.0/S1(LK,3)
      S2(LI,3) = S2(LG,2)*((1.0+S1(LI,3)**2)/(1.0+S1(LG,2)**2))**1.5
      S2(LH,2) = S2(LJ,3)*((1.0+S1(LH,2)**2)/(1.0+S1(LJ,3)**2))**1.5
      S2(LM,3) =-S2(LB,1)*((1.0+S1(LM,3)**2)/(1.0+S1(LB,1)**2))**1.5
      S2(LA,1) =-S2(LK,3)*((1.0+S1(LA,1)**2)/(1.0+S1(LK,3)**2))**1.5
C-----OVERRELAXATION TO FIT POLYNOMIALS TO DATA
      OMEGA = 1.0717968
  105 ETA = 0.0
      DO 115 KR = 1,3
      NS = NTD(KR)
      NT = NTE(KR)
      DO 110 I = 2,NS
      W = (C(I,KR)-B(I,KR)*S2(I-1,KR)-(0.5-B(I,KR))*S2(I+1,KR)
     A -S2(I,KR))*OMEGA
      IF(ABS(W) - ETA) 110,110,109
  109 ETA = ABS(W)
  110 S2(I,KR) = S2(I,KR) + W
  115 CONTINUE
      I = LB    $KR = 1
      S1(I,KR) = DELY(I,KR) - H(I,KR)*(2.0*S2(I,KR)+S2(I+1,KR))/6.0
      I = LG    $KR = 2
      S1(I,KR) = DELY(I,KR) - H(I,KR)*(2.0*S2(I,KR)+S2(I+1,KR))/6.0
      I = LJ    $KR = 3
      S1(I,KR) = DELY(I,KR) - H(I,KR)*(2.0*S2(I,KR)+S2(I+1,KR))/6.0
      I = LK    $KR = 3
      S1(I,KR) = DELY(I-1,KR)+H(I-1,KR)*(S2(I-1,KR)+2.0*S2(I,KR))/6.0
      S1(LI,3) = 1.0/S1(LG,2)
      S1(LH,2) = 1.0/S1(LJ,3)
      S1(LM,3) = 1.0/S1(LB,1)
      S1(LA,1) = 1.0/S1(LK,3)
      S2(LI,3) = S2(LG,2)*((1.0+S1(LI,3)**2)/(1.0+S1(LG,2)**2))**1.5
      S2(LH,2) = S2(LJ,3)*((1.0+S1(LH,2)**2)/(1.0+S1(LJ,3)**2))**1.5
      S2(LM,3) =-S2(LB,1)*((1.0+S1(LM,3)**2)/(1.0+S1(LB,1)**2))**1.5
      S2(LA,1) =-S2(LK,3)*((1.0+S1(LA,1)**2)/(1.0+S1(LK,3)**2))**1.5
      IF(ETA - EPSLN) 120,105,105
C-----COMPLETION OF DATA ON POLYNOMIALS FITTED
C-----PLUS PATCHING IN OF 4TH DEGREE POLYNOMIAL AT OVERLAPS
  120 DO 130 KR = 1,2
      WRITE(6,14) KR
      NS = NTD(KR)
      NT = NTE(KR)
      DO 122 I = 2,NS
      S3(I,KR) = (S2(I+1,KR) - S2(I,KR))/H(I,KR)
  122 S1(I,KR) = DELY(I,KR) - H(I,KR)*(2.0*S2(I,KR)+S2(I+1,KR))/6.0
      I = NT
      S1(I,KR) = DELY(I-1,KR)+H(I-1,KR)*(S2(I-1,KR)+2.0*S2(I,KR))/6.0
```

```
      S4(KR) = 6.0*(S2(2,KR)+S2(1,KR) - (S1(2,KR)-S1(1,KR))/H(1,KR))
     A /(H(1,KR)**2)
      S3(1,KR) = (S2(2,KR)-S2(1,KR))/H(1,KR) - 0.5*H(1,KR)*S4(KR)
      WRITE(6,22) S4(KR)
      WRITE(6,12)
      DO 126 I = 1,NT
      IF(S2(I,KR)) 124,123,124
  123 S2(I,KR) = 1.0E-08
  124 RADC = (1.0+S1(I,KR)**2)**1.5/S2(I,KR)
  126 WRITE(6,6) I, X(I,KR), Y(I,KR), S1(I,KR), S2(I,KR), S3(I,KR), RADC
  130 CONTINUE
      KR = 3
      WRITE(6,14) KR
      NS = NTD(KR)
      NR = NS - 1
      DO 132 I = 2,NR
      S3(I,KR) = (S2(I+1,KR) - S2(I,KR))/H(I,KR)
  132 S1(I,KR) = DELY(I,KR) - H(I,KR)*(2.0*S2(I,KR)+S2(I+1,KR))/6.0
      WRITE(6,24)
      DO 142 I = 2,NS
      RADC = (1.0+S1(I,KR)**2)**1.5/ABS(S2(I,KR))
      IF(S1(I,KR)) 136,134,138
  134 SINV1 = 1.0E+08
      SINV2 = 1.0E+08
      GO TO 140
  136 SINV1 = 1.0/S1(I,KR)
      SINV2 = (1.0+SINV1**2)**1.5/RADC
      GO TO 140
  138 SINV1 = 1.0/S1(I,KR)
      SINV2 = -(1.0+SINV1**2)**1.5/RADC
  140 WRITE(6,6) I, Y(I,KR), X(I,KR), S1(I,KR), S2(I,KR), S3(I,KR),
     A RADC, SINV1, SINV2
  142 CONTINUE
      GO TO 500
C-----X GIVEN, UPPER SURFACE WANTED
  200 XD = CLINE
      KR = 1
      IF(XD - 1.0) 202,250,250
  202 IF(XD - X(1,KR)) 210,260,260
C     X IS IN REGION GIVEN BY X=FUNCTION OF Y
  210 KR = 3
      I = NTD(KR)
  212 IF(XD - Y(I,KR)) 214,215,215
  214 I = I - 1
      IF(I - 2) 240,212,212
C-----INITIAL VALUES FOR ITERATION
  215 IF(S1(I,KR)) 2160,216,216
 2160 I = I + 1
      XINC = XD - Y(I,KR)
      YINC = XINC/S1(I,KR)
 2180 XTRY  = Y(I,KR) + YINC*(S1(I,KR) + 0.5*YINC*(S2(I,KR)
     A + YINC*S3(I-1,KR)/3.0))
      XP = S1(I,KR) + YINC*(S2(I,KR) + 0.5*YINC*S3(I-1,KR))
      XPP = S2(I,KR) + YINC*S3(I-1,KR)
      IF(ABS(XD - XTRY) - CVRT)230,230,2200
```

```
 2200 YINC = YINC + (XD - XTRY)/XP
      GO TO 2180
  216 XINC = XD - Y(I,KR)
      YINC = XINC/DELY(I,KR)
C-----ITERATE TO GET DESIRED X CUT
  218 XTRY  = Y(I,KR) + YINC*(S1(I,KR) + 0.5*YINC*(S2(I,KR)
     A + YINC*S3(I,KR)/3.0))
      XP = S1(I,KR) + YINC*(S2(I,KR) + 0.5*YINC*S3(I,KR))
      XPP = S2(I,KR) + YINC*S3(I,KR)
      IF(ABS(XD - XTRY) - CVRT)230,230,220
  220 YINC = YINC + (XD - XTRY)/XP
      GO TO 218
  230 YUS = X(I,KR) + YINC
      YPSU = 1.0/XP
      RADCU = -(1.0 + XP**2)**1.5/XPP
      GO TO (500,300,500,500,500),NSHP
C-----X VALUE DESIRED IS OUT OF RANGE AT NOSE
  240 WRITE(6,261) XD
      GO TO 500
C-----X IS IN REGION AFT OF TRAILING EDGE
  250 I = NTE(KR)
      XINC = XD - X(I,KR)
      YUS = Y(I,KR) +XINC*S1(I,KR)
      YPSU = S1(I,KR)
      RADCU = -1.0E+08
      GO TO (500,300,500,500,500),NSHP
C----- X IS IN REGION DESCRIBED BY Y = FUNCTION OF X
  260 I = 1
  262 IF(XD - X(I,KR)) 266,268,264
  264 I = I + 1
      GO TO 262
  266 I = I - 1
  268 XINC = XD - X(I,KR)
      YUS = Y(I,KR) + XINC*(S1(I,KR) + 0.5*XINC*(S2(I,KR)
     A + XINC*S3(I,KR)/3.0))
      YPSU = S1(I,KR) + XINC*(S2(I,KR) + 0.5*XINC*S3(I,KR))
      YPP = S2(I,KR) + XINC*S3(I,KR)
      IF(I.GT.1) GO TO 270
      YUS = YUS + XINC**4*S4(KR)/24.0
      YPSU = YPSU +XINC**3*S4(KR)/6.0
      YPP = YPP + 0.5*XINC**2*S4(KR)
  270 RADCU =  (1.0 + YPSU**2)**1.5/YPP
      GO TO (500,300,500,500,500),NSHP
C-----X GIVEN, LOWER SURFACE WANTED
  300 XD = CLINE
      KR = 2
      IF(XD - 1.0) 302,350,350
  302 IF(XD - X(1,KR)) 310,360,360
C     X IS IN REGION GIVEN BY X=FUNCTION OF Y
  310 KR = 3
      I = 2
  312 IF(XD - Y(I,KR)) 314,316,316
  314 I = I + 1
      IF(I - NTE(KR)) 312,312,240
C-----INITIAL VALUES FOR ITERATION
```

```
  316 XINC = XD - Y(I,KR)
      YINC = XINC/DELY(I-1,KR)
C-----ITERATE TO GET DESIRED X CUT
  318 XTRY  = Y(I,KR) + YINC*(S1(I,KR) + 0.5*YINC*(S2(I,KR)
     A + YINC*S3(I-1,KR)/3.0))
      XP = S1(I,KR) + YINC*(S2(I,KR) + 0.5*YINC*S3(I-1,KR))
      XPP = S2(I,KR) + YINC*S3(I-1,KR)
      IF(ABS(XD - XTRY) - CVRT)330,330,320
  320 YINC = YIDC + (XD - XTRY)/XP
      GO TO 318
  330 YLS = X(I,KR) + YINC
      YPSL = 1.0/XP
      RADCL =  (1.0 + XP**2)**1.5/XPP
      GO TO 500
C-----X IS IN REGION AFT OF TRAILING EDGE
  350 I = NTE(KR)
      XINC = XD - X(I,KR)
      YLS = Y(I,KR) +XINC*S1(I,KR)
      YPSL = S1(I,KR)
      RADCL =  1.0E+08
      GO TO 500
C----- X IS IN REGION DESCRIBED BY Y = FUNCTION OF X
  360 I = 1
  362 IF(XD - X(I,KR)) 366,368,364
  364 I = I + 1
      GO TO 362
  366 I = I - 1
  368 XINC = XD - X(I,KR)
      YLS = Y(I,KR) + XINC*(S1(I,KR) + 0.5*XINC*(S2(I,KR)
     A + XINC*S3(I,KR)/3.0))
      YPSL = S1(I,KR) + XINC*(S2(I,KR) + 0.5*XINC*S3(I,KR))
      YPP = S2(I,KR) + XINC*S3(I,KR)
      IF(I.GT.1) GO TO 370
      YLS = YLS + XINC**4*S4(KR)/24.0
      YPSL = YPSL +XINC**3*S4(KR)/6.0
      YPP = YPP + 0.5*XINC**2*S4(KR)
  370 RADCL = (1.0 + YPSL**2)**1.5/YPP
      GO TO 500
C-----Y GIVEN, HORIZONTAL CUT
  400 YD = CLINE
      KR = 3
      NS = NTD(KR)
      IF(YD - X(2,KR)) 440,402,402
  402 IF(YD - X(NS,KR)) 408,406,440
  406 I = 2
  408 IF(YD - X(I,KR)) 412,414,410
  410 I = I + 1
      GO TO 408
  412 I = I - 1
  414 YINC = YD - X(I,KR)
      XNOSE = Y(I,KR) + YINC*(S1(I,KR) + 0.5*YINC*(S2(I,KR)
     A + YINC*S3(I,KR)/3.0))
      XP = S1(I,KR) + YINC*(S2(I,KR) + 0.5*YINC*S3(I,KR))
      XPP = S2(I,KR) + YINC*S3(I,KR)
      RADC = (1.0 + XP**2)**1.5/XPP
```

```
      YPNOSE = 1.0/XP
      GO TO 500
C-----Y OUT OF RANGE IN WHICH FUNCTION IS WELL DEFINED
  440 WRITE(6,28)YD
  500 RETURN
    1 FORMAT(8A10)
    2 FORMAT(8E10.3)
    3 FORMAT(23X,8A10)
    4 FORMAT(16I5)
    6 FORMAT(5X,I5,8E15.3)
   12 FORMAT(/5X,5HENTRY,14X,1HX,14X,1HY,13X,2HYP,12X,3HYPP,11X,
     A 4HYPPP,11X,4HRADC)
   14 FORMAT(/5X,6HREGION,I5)
   18 FORMAT(/5X,8HEPSLN = ,E10.3)
   22 FORMAT(/5X,18HYPPPP AT ENTRY 1 =,E10.3)
   24 FORMAT(/5X,5HENTRY,14X,1HX,14X,1HY,13X,2HXP,12X,3HXPP,11X,
     A 4HXPPP,11X,4HRADC,13X,2HYP,12X,3HYPP)
   26 FORMAT(/7X,2HX=,E10.3,11HNOT ALLOWED)
   28 FORMAT(/7X,2HY=,E10.3,11HNOT ALLOWED)
      END
```

```
      SUBROUTINE CAXIS(KR,MH,MS,MAN,NCL,LCU,LCL,IA,DA,SA,CA,LA,
     A XY,SC,DS,LF)
C     DETERMINES GEOMETRY OF LOCAL SURFACE NORMALS
      COMMON/ZMESH/  DX(6), DY(6), RH(9)
      COMMON/AIRF/YUS,YLS,YPSU,YPSL,XNOSE,YPNOSE,RADCU,RADCL,RADC
      DIMENSION IA(2,LA), DA(2,LA), SA(2,LA), CA(2,LA), XY(2,LF),
     A SC(2,LF), DS(LF)
      WRITE(6,8)
      DO 20 M = MH,MS
      DO 20 KAT = 1,2
      XA = FLOAT(M - MAN)*DX(KR)
      XTEST = XA
      N = IA(KAT,M)
      YA = FLOAT(N - NCL)*DY(KR)
      XERR = DA(KAT,M)*DY(KR)*SA(KAT,M)*CA(KAT,M)
      GO TO(11,12),KAT
   11 MM = LCL - M
      KSP = 4
      GO TO 13
   12 MM = M + LCU
      KSP = 5
      XERR = -XERR
   13 XTEST = XTEST - XERR
      CALL SHAPE(KSP,XTEST)
      GO TO(14,15),KAT
   14 YS = YLS
      YPS = YPSL
      GO TO 16
   15 YS = YUS
      YPS = YPSU
   16 XERR = (XTEST-XA+(YS-YA)*YPS)/(1.0 + YPS**2)
      IF(ABS(XERR) - 0.01*DY(KR)) 17,17,13
   17 XY(1,MM) = XTEST
      XY(2,MM) = YS
      SC(2,MM) = 1.0/SQRT(1.0 + YPS**2)
      SC(1,MM) = YPS*SC(2,MM)
      DS(MM) = SQRT((YA-YS)**2 + (XA-XTEST)**2)/DY(KR)
      GO TO(18,20),KAT
   18 SC(1,MM) = -SC(1,MM)
      SC(2,MM) = -SC(2,MM)
   20 WRITE(6,4) M, MM, XY(1,MM), XY(2,MM), SC(1,MM), SC(2,MM), DS(MM)
      RETURN
    4 FORMAT(2I5,12F10.5)
    8 FORMAT(4X,1HM,3X,2HL8,5X,1HX,9X,1HY,9X,3HSIN,7X,3HCOS,6X,6HOFFSET)
      END
```

```
      SUBROUTINE PONDER(   KSPL,MLL,MUL,XTZ,YTZ,XYT,SCT,DST,DWT,DWTX,
     A AWT,CWT,IWT,LBX,MTT)
C     CALCULATES INTERPOLATION WEIGHTS TO BE APPLIED IN TRANSFERRING FROM
C     DISTORTED MESH TO ADJACENT LOCAL ROTATED MESH (DWT) AND (DWTX), FROM
C     SURFACE NORMALS TO CARTESIAN AUXILARIES,(AWT), AND FROM CARTESIAN GR
C     TO LOCAL ROTATED MESH (CWT).
      COMMON/ZMESH/  DX(6), DY(6), RH(9)
      COMMON/SPOT/M,KR
      COMMON/AIRF/YUS,YLS,YPSU,YPSL,XNOSE,YPNOSE,RADCU,RADCL,RADC
      DIMENSION XYT(2,LBX), SCT(2,LBX), DST(LBX), DWT(9,4,LBX),
     A DWTX(9,2,6), AWT(3,LBX), CWT(9,3,LBX),IWT(3,LBX)
  950 DO 980 M = MLL,MUL
      CALL DISTW(1,0.0,0.0,0.0,XYT,SCT,LBX)
      LIP = 1
      XP = XYT(1,M-1) + DX(KR)*SCT(2,M-1)
      YP = XYT(2,M-1) + DX(KR)*SCT(1,M-1)
      CALL DISTW(2,XP,YP,DWT(1,LIP,M),XYT,SCT,LBX)
      LIP = 2
      XP = XP - DY(KR)*SCT(1,M-1)
      YP = YP + DY(KR)*SCT(2,M-1)
      CALL DISTW(2,XP,YP,DWT(1,LIP,M),XYT,SCT,LBX)
      LIP = 3
      XP = XYT(1,M+1) - DX(KR)*SCT(2,M+1)
      YP = XYT(2,M+1) - DX(KR)*SCT(1,M+1)
      CALL DISTW(2,XP,YP,DWT(1,LIP,M),XYT,SCT,LBX)
      LIP = 4
      XP = XP - DY(KR)*SCT(1,M+1)
      YP = YP + DY(KR)*SCT(2,M+1)
      CALL DISTW(2,XP,YP,DWT(1,LIP,M),XYT,SCT,LBX)
C-----SPECIAL INTERPOLATION WEIGHTS AT ENDS OF OPERATING RANGES
      IF((M.NE.MLL).AND.(M.NE.MUL)) GO TO 951
      KSPL = KSPL + 1
      LIP = 1
      XP = XYT(1,M) - DX(KR)*SCT(2,M)
      YP = XYT(2,M) - DX(KR)*SCT(1,M)
      IF(M.EQ.MUL) XP = XP + 2.0*DX(KR)*SCT(2,M)
      IF(M.EQ.MUL) YP = YP + 2.0*DX(KR)*SCT(1,M)
      CALL DISTW(2,XP,YP,DWTX(1,LIP,KSPL),XYT,SCT,LBX)
      LIP = 2
      XP = XP - DY(KR)*SCT(1,M)
      YP = YP + DY(KR)*SCT(2,M)
      CALL DISTW(2,XP,YP,DWTX(1,LIP,KSPL),XYT,SCT,LBX)
  951 AWT(1,M) = 0.5*(1.0 - DST(M))*(2.0 - DST(M))
      AWT(2,M) = DST(M)*(2.0 - DST(M))
      AWT(3,M) = 0.5*DST(M)*(DST(M) - 1.0)
      LIP = 2
      XP = XYT(1,M) - 2.0*DY(KR)*SCT(1,M)
      YP = XYT(2,M) + 2.0*DY(KR)*SCT(2,M)
  952 MK = 1
      XT = XTZ
  954 IF(XT - XP) 956,958,958
  956 XT = XT + DX(KR)
      MK = MK + 1
      GO TO 954
```

```
958 AP = (XP - XT)/DX(KR)
    IF(AP + 0.5) 960,960,961
960 MK = MK - 1
    AP = 1.0 + AP
961 IF(MK+1-MTI) 963,963,962
962 MK = MK - 1
    AP = 1.0 + AP
963 NK = 1
    YT = YTZ
964 IF(YT - YP) 966,968,968
966 YT = YT + DY(KR)
    NK = NK + 1
    GO TO 964
968 BP = (YP - YT)/DY(KR)
    IF(BP + 0.5) 970,970,972
970 NK = NK - 1
    BP = 1.0 + BP
972 CALL CARTA(AP,BP,CNT(1,LIP,M))
    INT(LIP,M) = 100*MK + NK
    GO TO (976,974,980),LIP
974 LIP = 1
    XP = XP - DX(KR)*SCT(2,M)
    YP = YP - DX(KR)*SCT(1,M)
    GO TO 952
976 LIP = 3
    XP = XP + 2.0*DX(KR)*SCT(2,M)
    YP = YP + 2.0*DX(KR)*SCT(1,M)
    GO TO 952
980 CONTINUE
    RETURN
    END
```

```
      SUBROUTINE BUNTE   (NE,NH,IFO,JFO,ID)
C     INTERPOLATION INSTRUCTIONS AT FIELD BOUNDARIES
      DIMENSION ID(10), IFO(8), JFO(8)
C     CODE 1 INERT, CODE 2 SYMMETRIC 4 POINT SPREAD, CODE 3 PARABOLA WITH
C     POINTS BELOW, CODE 4 PARABOLA WITH TWO POINTS ABOVE
      J = 1
   10 JCT = MOD(JFO(J),100)
      IF(JCT) 14,14,12
   12 J = J + 1
      GO TO 10
   14 N1 = IFO(J)
      N2 = IFO(J+1) - 1
      NJ = NH - 1
      J = 0
      DO 26 N = NE, NJ
      J = J + 1
      IF(N+1-N1) 20,22,16
   16 IF(N   -N2) 18,24,20
   18 ID(J) = 1
      GO TO 26
   20 ID(J) = 2
      GO TO 26
   22 ID(J) = 3
      GO TO 26
   24 ID(J) = 4
   26 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DISTW(MD,XP,YP,TW,XYU,SCU,LDIM)
C-----CALCULATES WEIGHTS (TW) TO BE USED ON VALUES OF F AT 9 POINTS IN A
C     DISTORTED ARRAY TO FIND INTERPOLATED VALUE OF F AT CARTESIAN POINT X
C     MD = 1, BASIC PARAMETER, DETERMINED
C     MD = 2, WEIGHTS (TW) GENERATED FOR SPECIFIC XP,YP
      COMMON/SPOT/M,KR
      DIMENSION SSK(2), CSK(2), A(15),B(15),CB(9,9), DPU(9), DPPU(9),
     A DPL(9), CPPL(9), TW(9), XYU(2,LDIM), SCU(2,LDIM)
      COMMON/ZMESH/  DX(6), DY(6), RH(9)
      GO TO (100,200),MD
C-----CALCULATON OF INITIAL FITTING INFORMATION
C-----SKEWNESS OF SIDE LINES
  100 SSK(2) = SCU(1,M+1)*SCU(2,M) - SCU(2,M+1)*SCU(1,M)
      SSK(1) = SCU(1,M-1)*SCU(2,M) - SCU(2,M-1)*SCU(1,M)
      DO 102 N = 1,2
  102 CSK(N) = SQRT(1.0 - SSK(N)**2)
C-----COORDINATES OF NINE POINTS
      DO 104 N = 1,3,2
      IF(N.EQ.1) MM = M - 1
      IF(N.EQ.3) MM = M + 1
      A(N) = ((XYU(1,MM) - XYU(1,M))*SCU(2,M) + (XYU(2,MM) - XYU(2,M))
     A*SCU(1,M))/DY(KR)
  104 B(N) = (-(XYU(1,MM) - XYU(1,M))*SCU(1,M) + (XYU(2,MM) - XYU(2,M))
     A*SCU(2,M))/DY(KR) - 1.0
      A(4) = A(1) - SSK(1)
      A(7) = A(4) - SSK(1)
      A(6) = A(3) - SSK(2)
      A(9) = A(6) - SSK(2)
      DO 105 N = 2,8,3
  105 A(N) = 0.0
      B(4) = B(1) + CSK(1)
      B(7) = B(4) + CSK(1)
      B(6) = B(3) + CSK(2)
      B(9) = B(6) + CSK(2)
      B(2) = -1.0
      B(5) = 0.0
      B(8) = 1.0
C-----COORDINATES OF SIX AUXILIARIES
      DO 106 N = 1,3
      A(8+2*N) = A(1) - SSK(1)*(B(-1+3*N) - B(1))/CSK(1)
      B(8+2*N) = B(-1+3*N)
      A(9+2*N) = A(3) - SSK(2)*(B(-1+3*N) - B(3))/CSK(2)
  106 B(9+2*N) = B(-1+3*N)
      WRITE(6,18) KR, M
      WRITE(6,12)
      DO 140 J = 1,15
  140 WRITE (6,2) J, A(J), B(J)
C-----BASIC GEOMETRIC INPUT WEIGHTING FUNCTIONS
      DO 108 L = 1,9
      DO 108 K = 1,9
  108 CB(K,L) = 0.0
      CB(1,5) = 1.0
      CB(2,8) = 0.5
      CB(2,2) = -0.5
```

```
      CB(3,2) = 0.5
      CB(3,8) = 0.5
      CB(3,5) = - 1.0
      ALC = A(2) - A(10)
      ALD = A(5) - A(12)
      ALE = A(8) - A(14)
      ARC = (A(11) - A(2))/ALC
      ARD = (A(13) - A(5))/ALD
      ARE = (A(15) - A(8))/ALE
      BINCL = 1.0/(B(4) - B(1))
      SBCL = (B(10) - B(4))*BINCL
      SBDL = (B(12) - B(4))*BINCL
      SBEL = (B(14) - B(4))*BINCL
      BINCR = 1.0/(B(6) - B(3))
      SBCR = (B(11) - B(6))*BINCR
      SBDR = (B(13) - B(6))*BINCR
      SBER = (B(15) - B(6))*BINCR
      DENT = 1.0/(ALD*ARD*(ARD + 1.0))
      CB(4,5) = (ARD**2 - 1.0)*DENT
      CB(4,3) = 0.5*SBDR*(SBDR - 1.0)*DENT
      CB(4,6) = (1.0 - SBDR**2)*DENT
      CB(4,9) = 0.5*SBDR*(SBDR + 1.0)*DENT
      CMULT = -ARD**2*DENT
      CB(4,1) = CMULT*0.5*SBDL*(SBDL - 1.0)
      CB(4,4) = CMULT*(1.0 - SBDL**2)
      CB(4,7) = CMULT*0.5*SBDL*(SBDL + 1.0)
      DENT = DENT/ALD
      CB(7,5) = -(ARD + 1.0)*DENT
      CMULT = 1.0/ALD
      DO 112 K = 3,9,3
  112 CB(7,K) = CB(4,K)*CMULT
      CMULT = -CMULT/ARD
      DO 114 K = 1,7,3
  114 CB(7,K) = CB(4,K)*CMULT
      DO 116 L = 1,9
      DPU(L) = 0.0
      DPPU(L) = 0.0
      DPL(L) = 0.0
  116 DPPL(L) = 0.0
      DENT = 1.0/(ARE*(ARE + 1.0)*ALE)
      DPU(8) = (ARE**2 - 1.0)*DENT
      DPU(3) = 0.5*SBER*(SBER - 1.0)*DENT
      DPU(6) = (1.0 - SBER**2)*DENT
      DPU(9) = 0.5*SBER*(SBER + 1.0)*DENT
      CMULT = -ARE**2*DENT
      DPU(1) = CMULT*0.5*SBEL*(SBEL - 1.0)
      DPU(4) = CMULT*(1.0 - SBEL**2)
      DPU(7) = CMULT*0.5*SBEL*(SBEL + 1.0)
      DENT = DENT/ALE
      DPPU(8) = -(ARE + 1.0)*DENT
      CMULT = 1.0/ALE
      DO 118 K = 3,9,3
  118 DPPU(K) = DPU(K)*CMULT
      CMULT = - CMULT/ARE
      DO 120 K = 1,7,3
```

```
  120 DPPU(K) = DPU(K)*CMULT
      DENT = 1.0/(ARC*(ARC + 1.0)*ALC)
      DPL(2) = (ARC**2 - 1.0)*DENT
      DPL(3) = 0.5*SBCR*(SBCR - 1.0)*DENT
      DPL(6) = (1.0 - SBCR**2)*DENT
      DPL(9) = 0.5*SBCR*(SBCR + 1.0)*DENT
      CMULT = -ARC**2*DENT
      DPL(1) = CMULT*0.5*SBCL*(SBCL - 1.0)
      DPL(4) = CMULT*(1.0 - SBCL**2)
      DPL(7) = CMULT*0.5*SBCL*(SBCL + 1.0)
      DENT = DENT/ALC
      DPPL(2) = -(ARC + 1.0)*DENT
      CMULT = 1.0/ALC
      DO 122 K = 3,9,3
  122 DPPL(K) = CMULT*DPL(K)
      CMULT = -CMULT/ARC
      DO 124 K = 1,7,3
  124 DPPL(K) = CMULT*DPL(K)
      DO 130 K = 1,9
      CB(5,K) = 0.5*(DPU(K) - DPL(K))
      CB(8,K) = 0.5*(DPPU(K) - DPPL(K))
      CB(6,K) = 0.5*(DPU(K) + DPL(K)) - CB(4,K)
  130 CB(9,K) = 0.5*(DPPU(K) + DPPL(K)) - CB(7,K)
      WRITE(6,18) KR, M
      WRITE (6,14)
      DO 144 J = 1,9
  144 WRITE(6,2) J, (CB(J,K), K = 1,9)
      RETURN
C-----USE OF  BASIC INFORMATION TO GET WEIGHTING FACTORS FOR SPECIFIC POIN
  200 AP = ((XP - XYU(1,M))*SCU(2,M) + (YP - XYU(2,M))*SCU(1,M))/DY(KR)
      BP = (-(XP - XYU(1,M))*SCU(1,M) + (YP-XYU(2,M))*SCU(2,M))
     A/DY(KR) - 1.0
      DO 202 J = 1,9
  202 TW(J) = CB(1,J) + AP*(CB(4,J) + AP*CB(7,J))
     A + BP*(CB(2,J) + AP*(CB(5,J) + AP*CB(8,J))
     B + BP*(CB(3,J) + AP*(CB(6,J) + AP*CB(9,J))))
      RETURN
    2 FORMAT (5X,I5,9F10.4)
   12 FORMAT (/9X,1HJ,6X,4HA(J),6X,4HB(J))
   14 FORMAT (/9X,1HJ,6X,15HCB(J,K),  K=1,9)
   16 FORMAT (/9X,1HJ,6X,5HTW(J))
   18 FORMAT(/5X,4HKR =,I5,5X,3HM =,I5)
      END
```

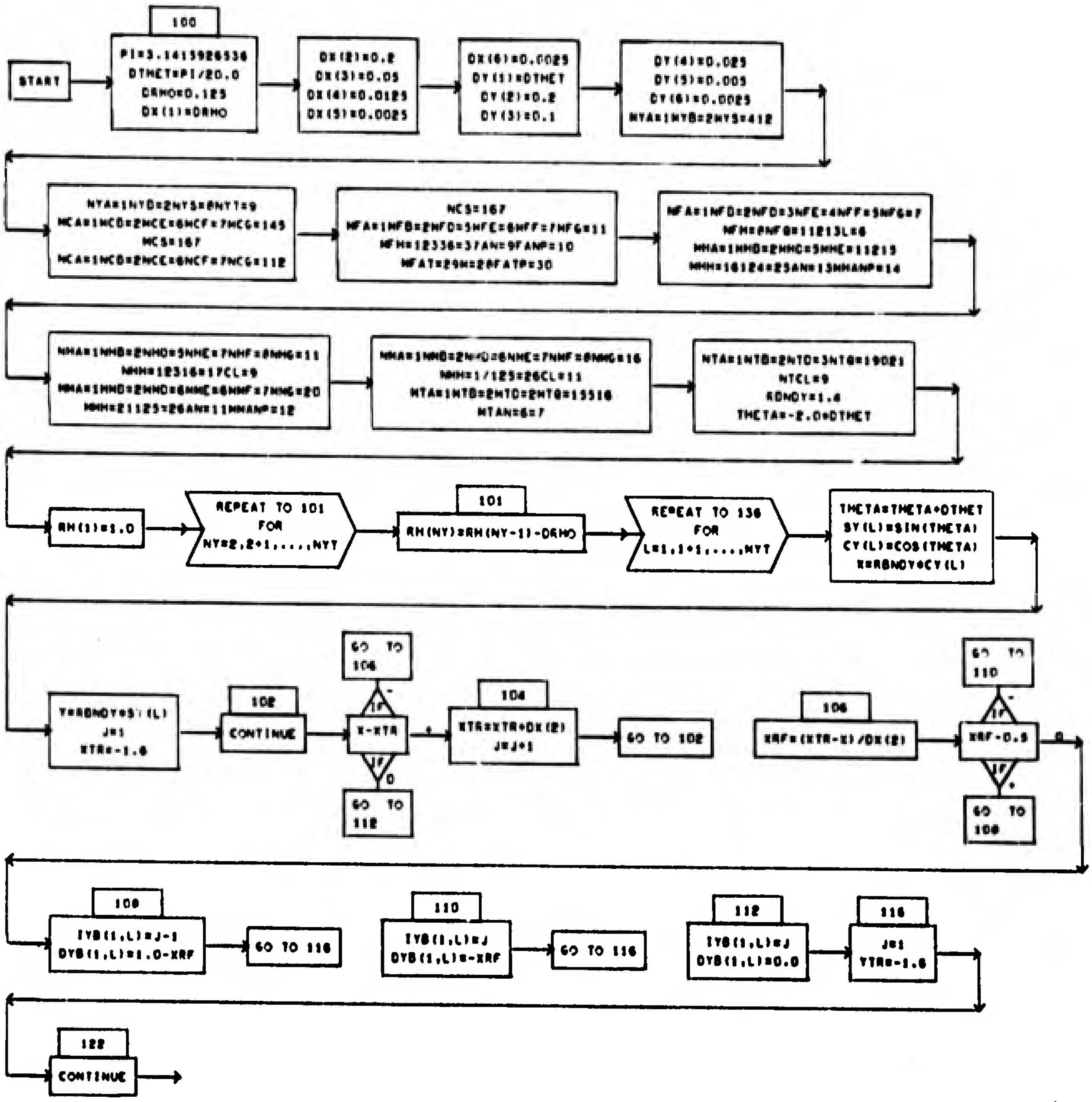
```
      SUBROUTINE WAIF(KR,KUT,M,XTZ,YTZ,XYM,SCM,CWMX,IWMX,LBX)
C-----SPECIAL AUXILIARY INTERPOLATIONS, CARTESIAN TO SURFACE NORMALS
      COMMON/ZMESH/  DX(6), DY(6), RH(9)
      DIMENSION XYM(2,LBX),SCM(2,LBX),CWMX(9,2,2),IWMX(2,2)
 9002 LIP = 1
      XP = XYM(1,M) - DY(KR)*SCM(1,M)
      YP = XYM(2,M) + DY(KR)*SCM(2,M)
 9003 MK = 1
      XT = XTZ
 9004 IF(XT - XP) 9006,9008,9008
 9006 XT = XT + DX(KR)
      MK = MK + 1
      GO TO 9004
 9008 AP = (XP - XT)/DX(KR)
      IF(AP + 0.5) 9010,9010,9012
 9010 MK = MK - 1
      AP = 1.0 + AP
 9012 NK = 1
      YT = YTZ
 9014 IF(YT - YP) 9016,9016,9018
 9016 YT = YT + DY(KR)
      NK = NK + 1
      GO TO 9014
 9018 BP = (YP - YT)/DY(KR)
      IF((LIP.EQ.1).AND.(KUT.EQ.1)) GO TO 9022
      IF((LIP.EQ.1).AND.(KUT.EQ.2)) GO TO 9020
      IF(BP + 0.5) 9020,9020,9022
 9020 NK = NK - 1
      BP = 1.0 + BP
 9022 CALL CARTW(AP,BP,CWMX(1,LIP,KUT))
      IWMX(LIP,KUT) = 100*MK + NK
      GO TO (9024,9026),LIP
 9024 LIP = 2
      XP = XP - DY(KR)*SCM(1,M)
      YP = YP + DY(KR)*SCM(2,M)
      GO TO 9003
 9026 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CARTW(AP,BP,TW)
C-----CALCULATES WEIGHTS TO BE USED IN INTERPOLATION FROM CARTESIAN MESH
      DIMENSION TW(9)
      TW(1) = 0.25*AP*BP
      TW(3) = -TW(1)
      TW(7) = TW(3)
      TW(9) = TW(1)
      TW(2) = 0.5*BP*(BP - 1.0)
      TW(8) = TW(2) + BP
      TW(4) = 0.5*AP*(AP - 1.0)
      TW(6) = TW(4) + AP
      TW(5) = 1.0 - AP**2 - BP**2
      RETURN
      END
```

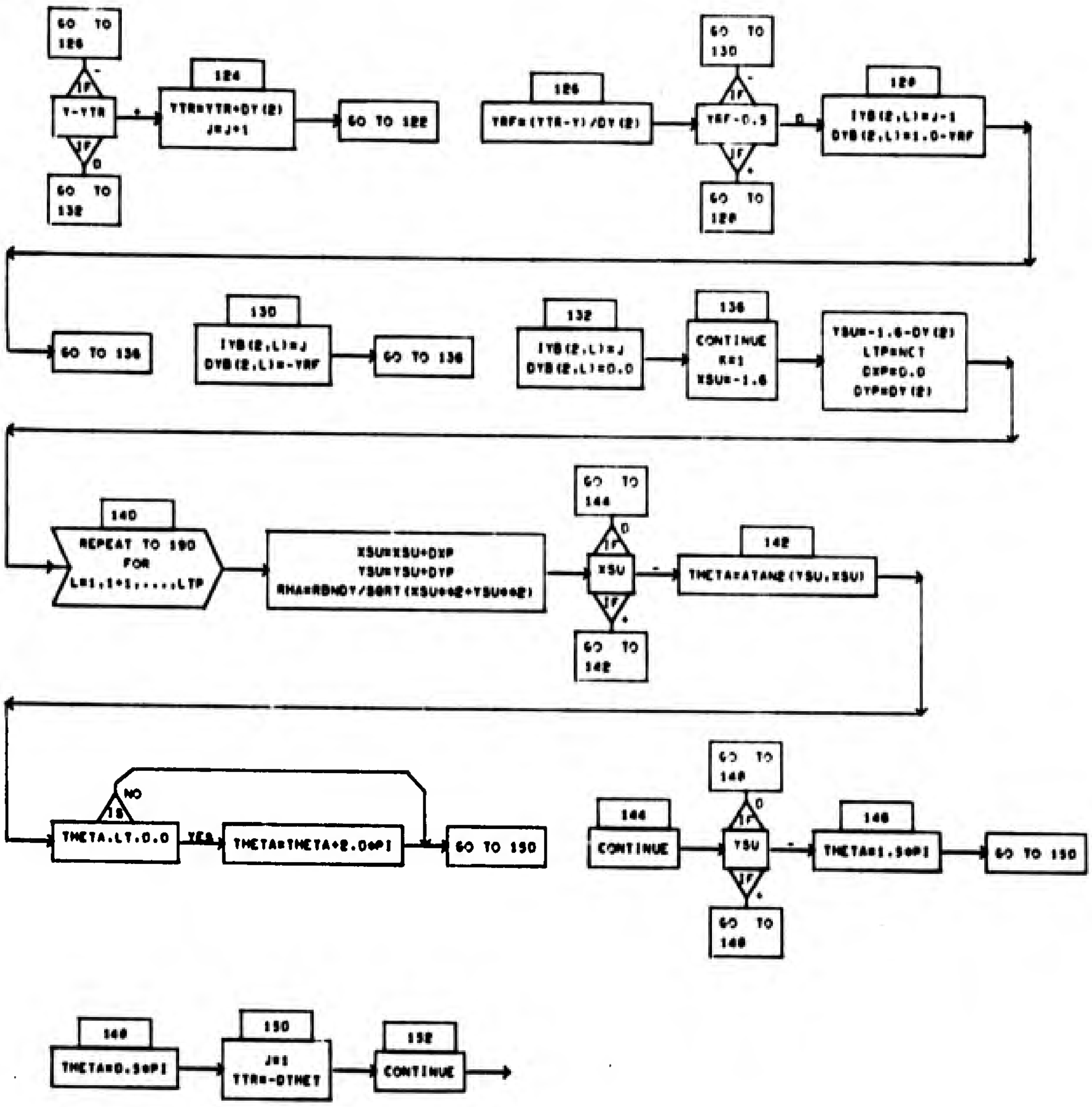
APPENDIX III

SETUP PROGRAM FLOW CHARTS

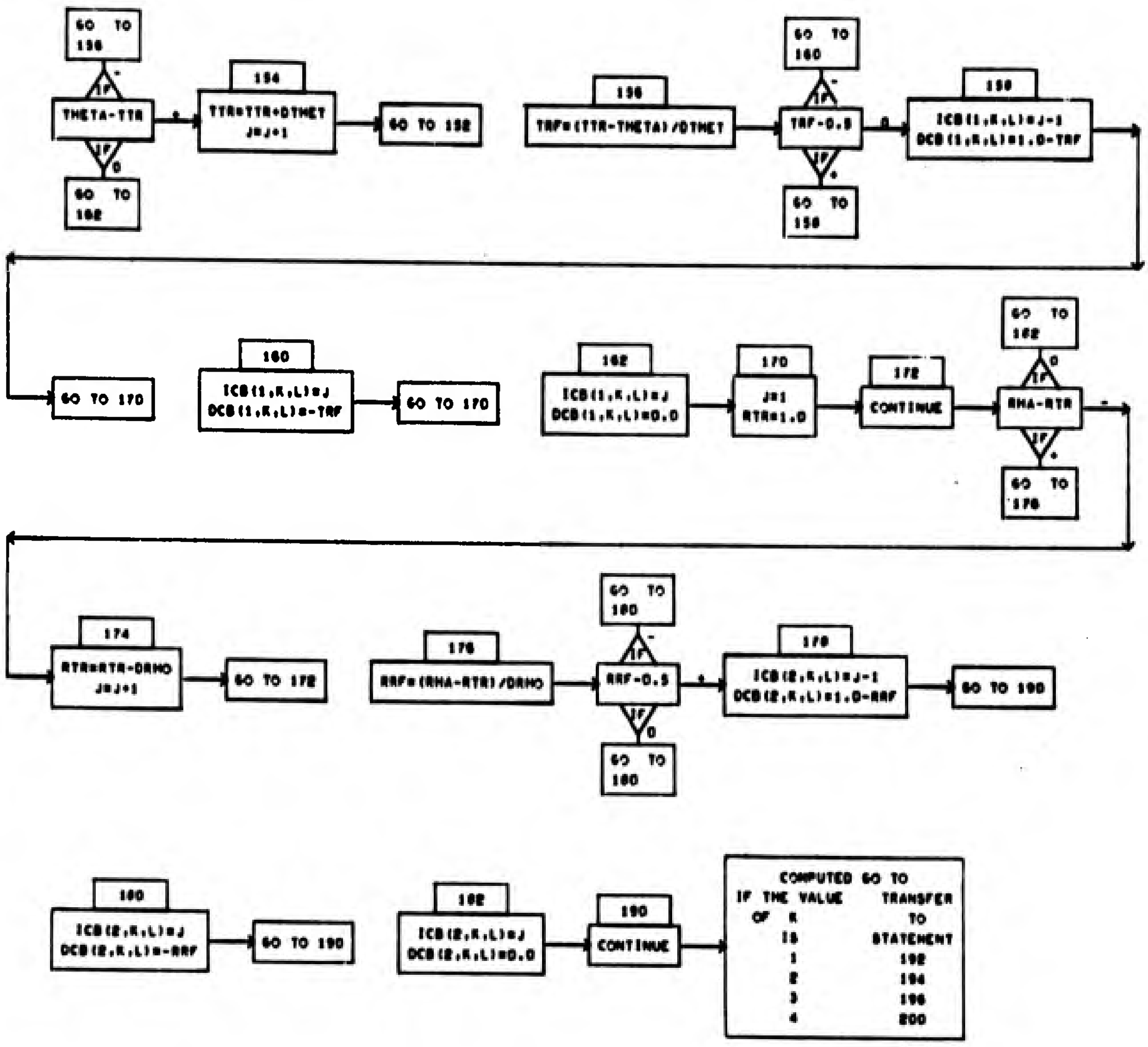
PROGRAM TEHAI (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE4,TAPE3)



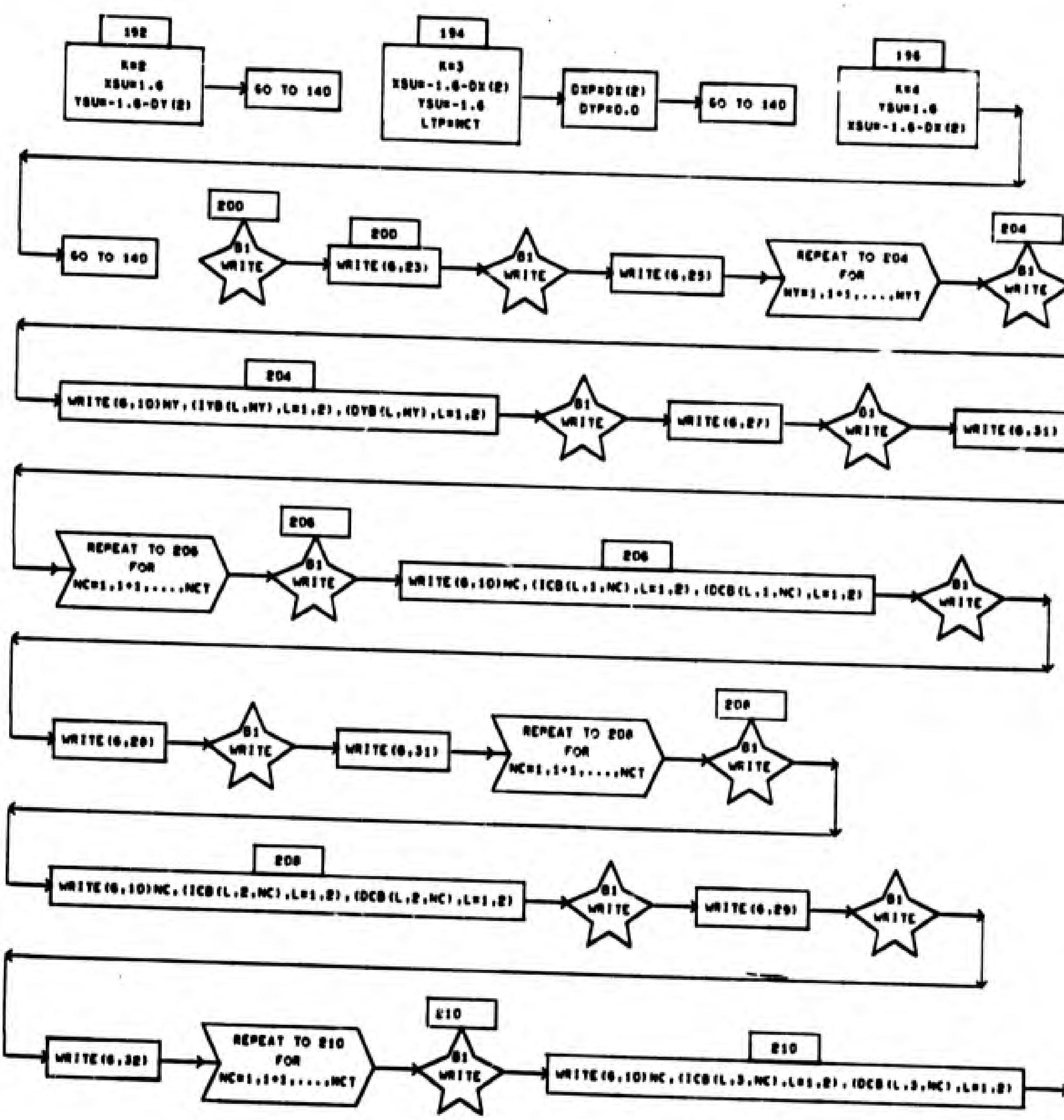
Detailed Flow Charts of Subroutine TEHAI (1 of 37)



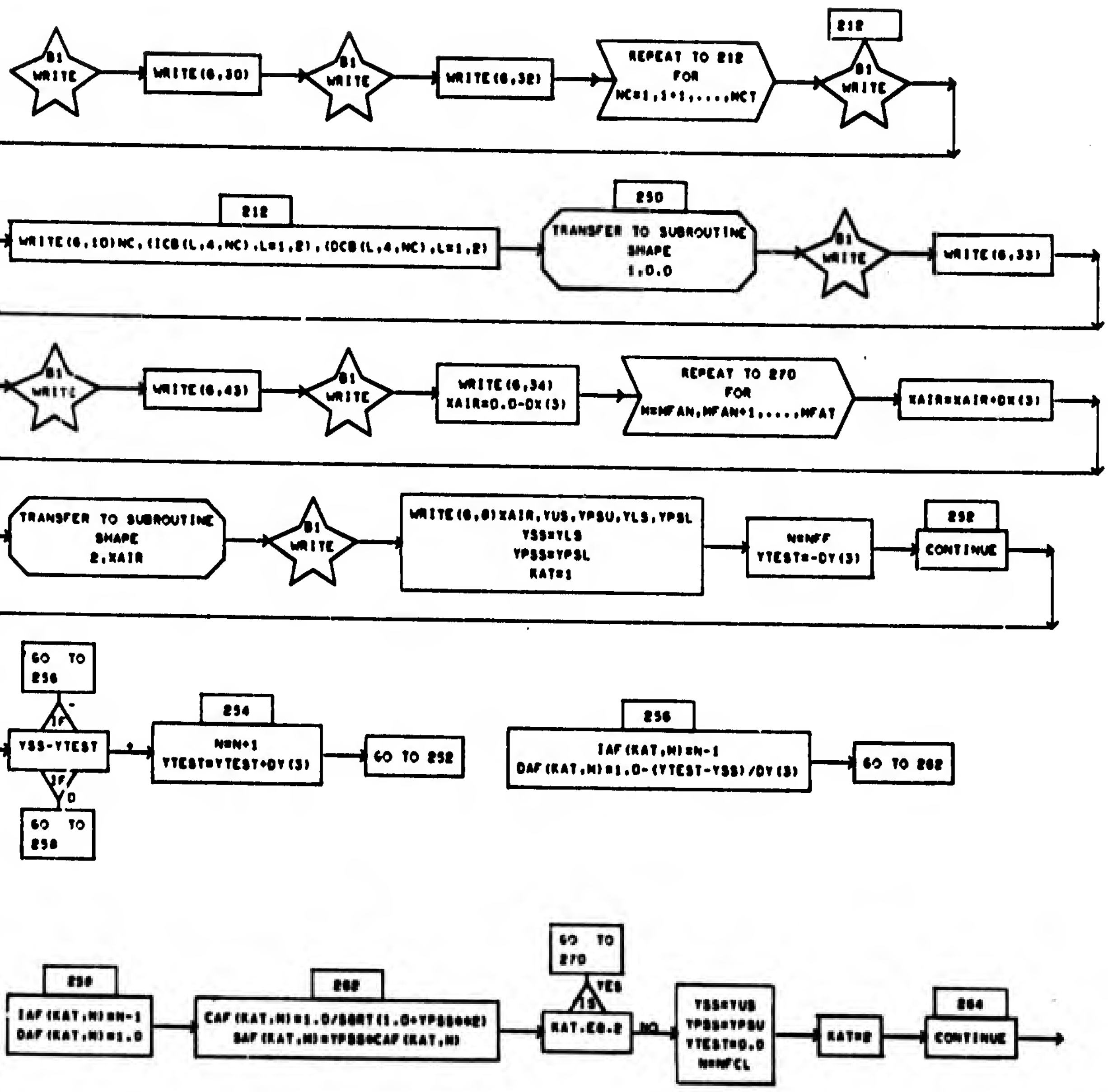
Detailed Flow Charts of Subroutine TEHA1 (2 of 37)



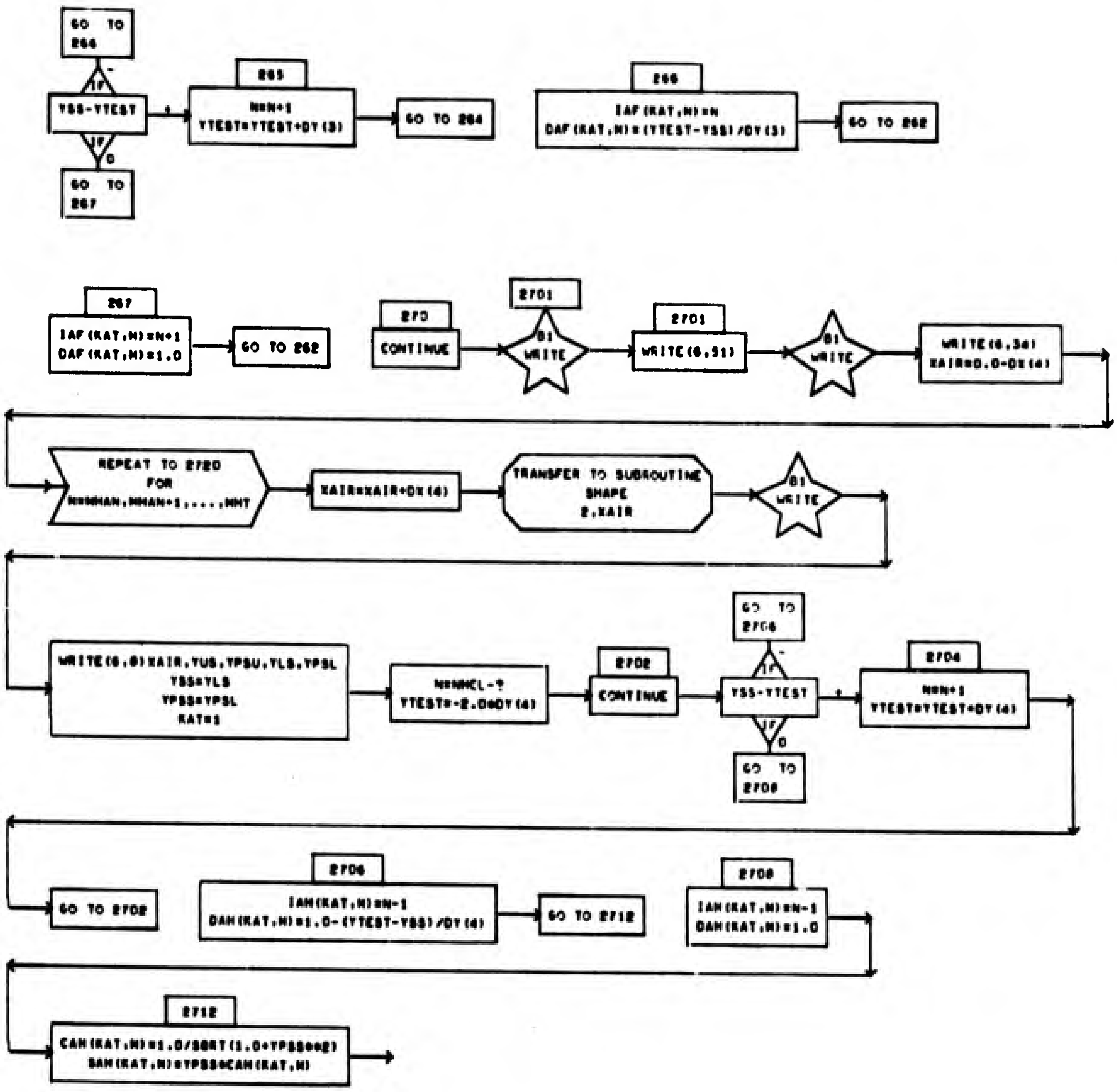
Detailed Flow Charts of Subroutine TEHAI (3 of 37)



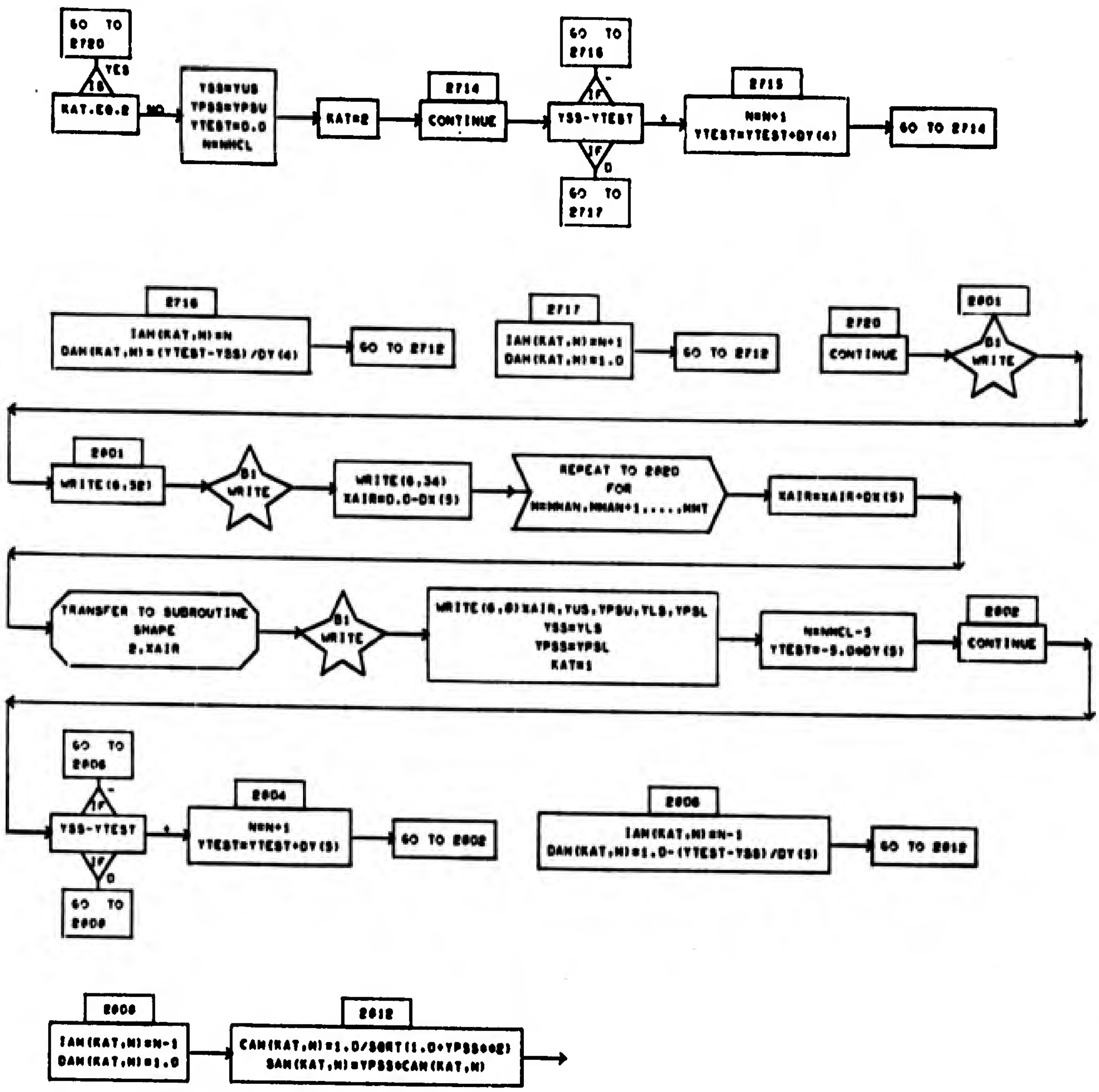
Detailed Flow Charts of Subroutine TEHAI (4 of 37)



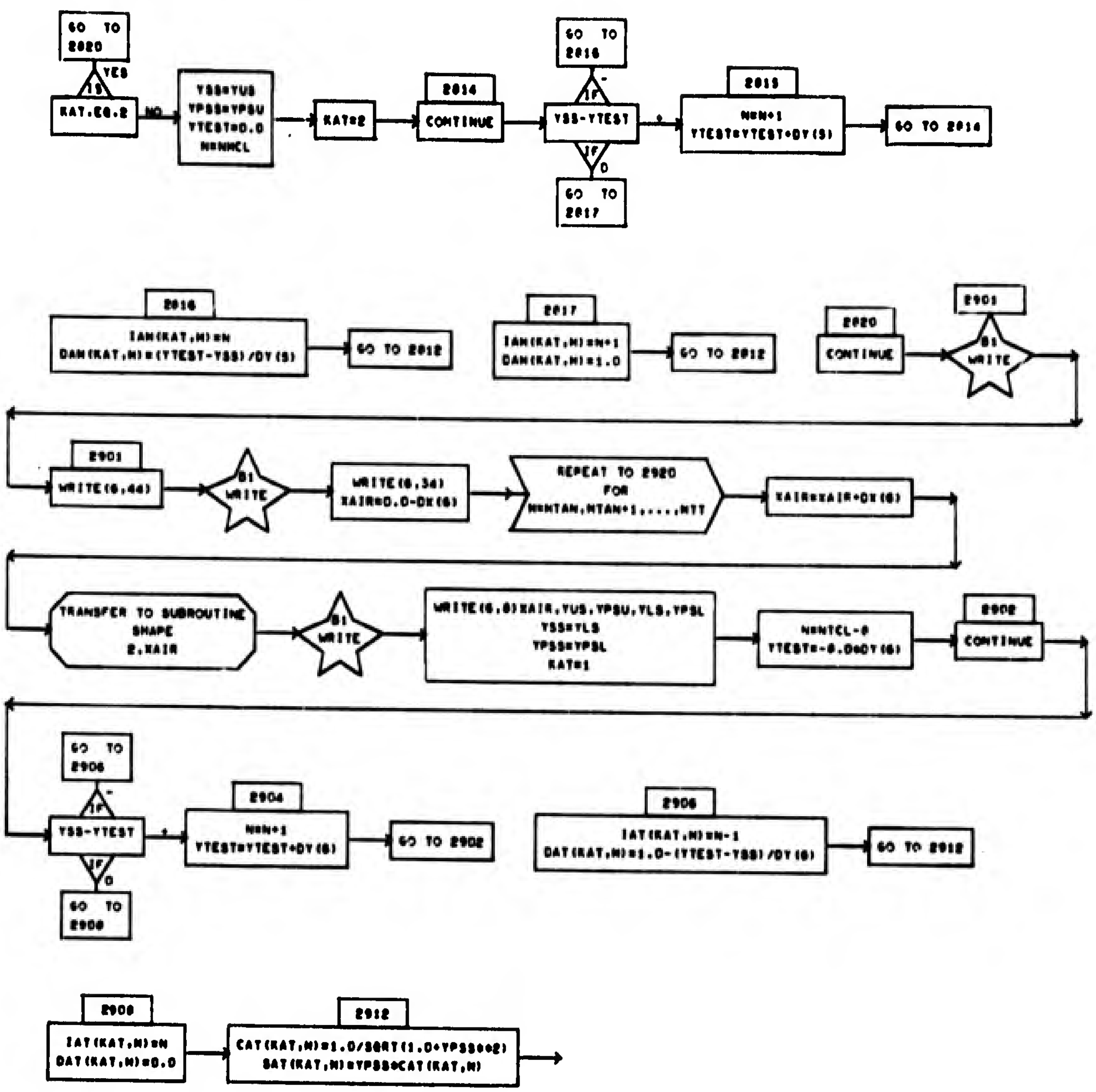
Detailed Flow Charts of Subroutine TEHAI (5 of 37)



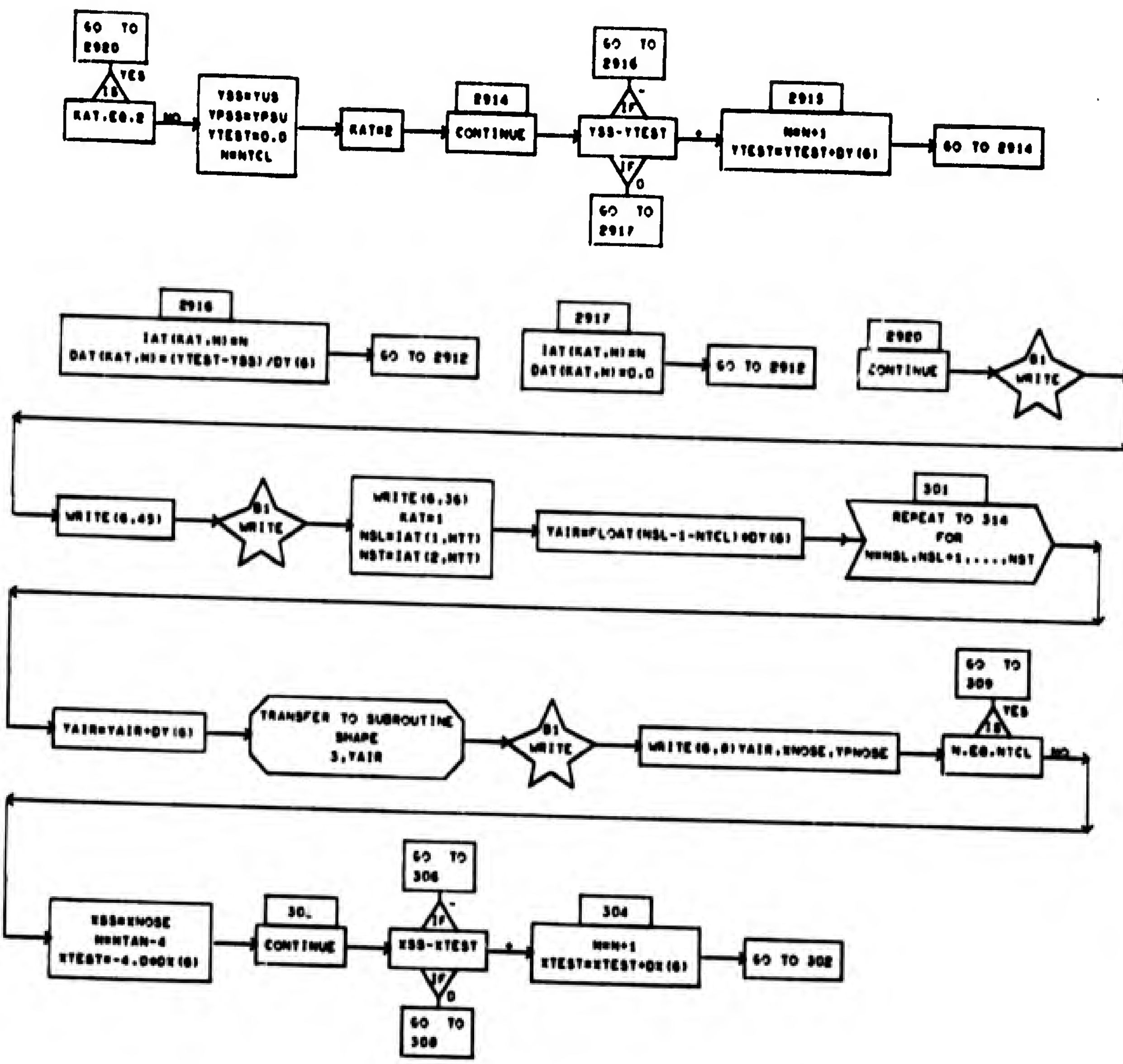
Detailed Flow Charts of Subroutine TEHAI (6 of 37)



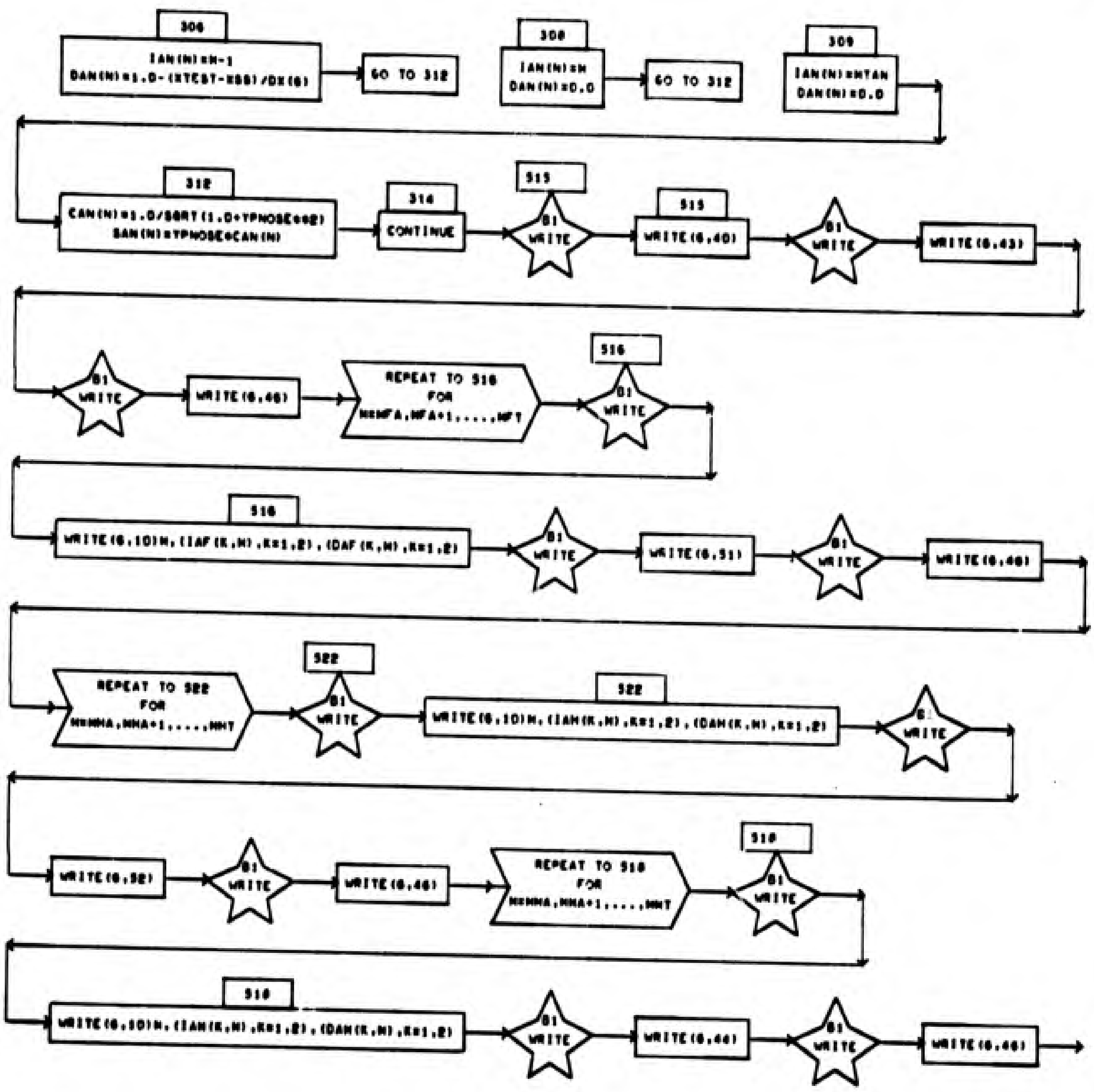
Detailed Flow Charts of Subroutine TEHAI (7 of 37)



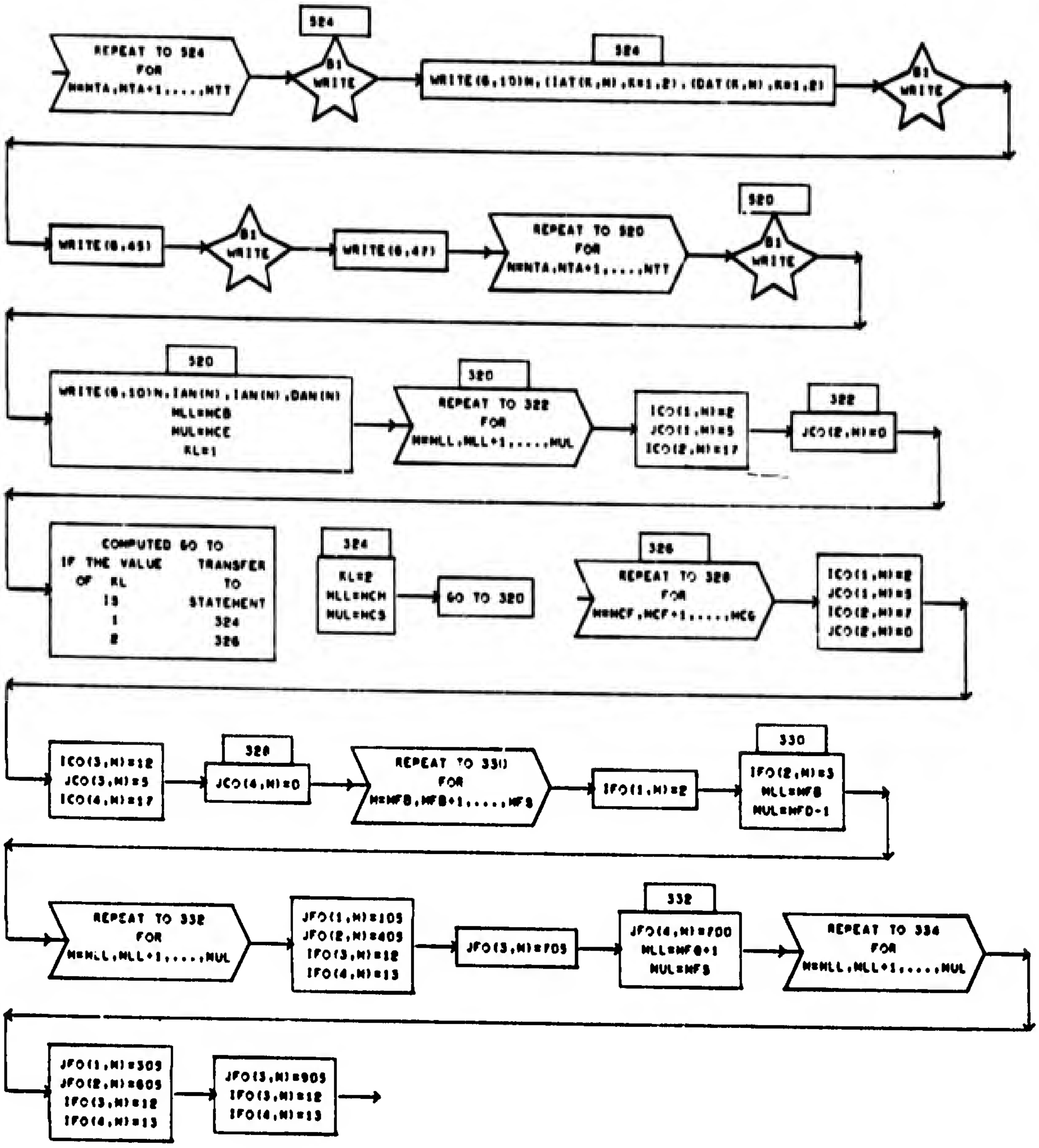
Detailed Flow Charts of Subroutine TEHAI (8 of 37)



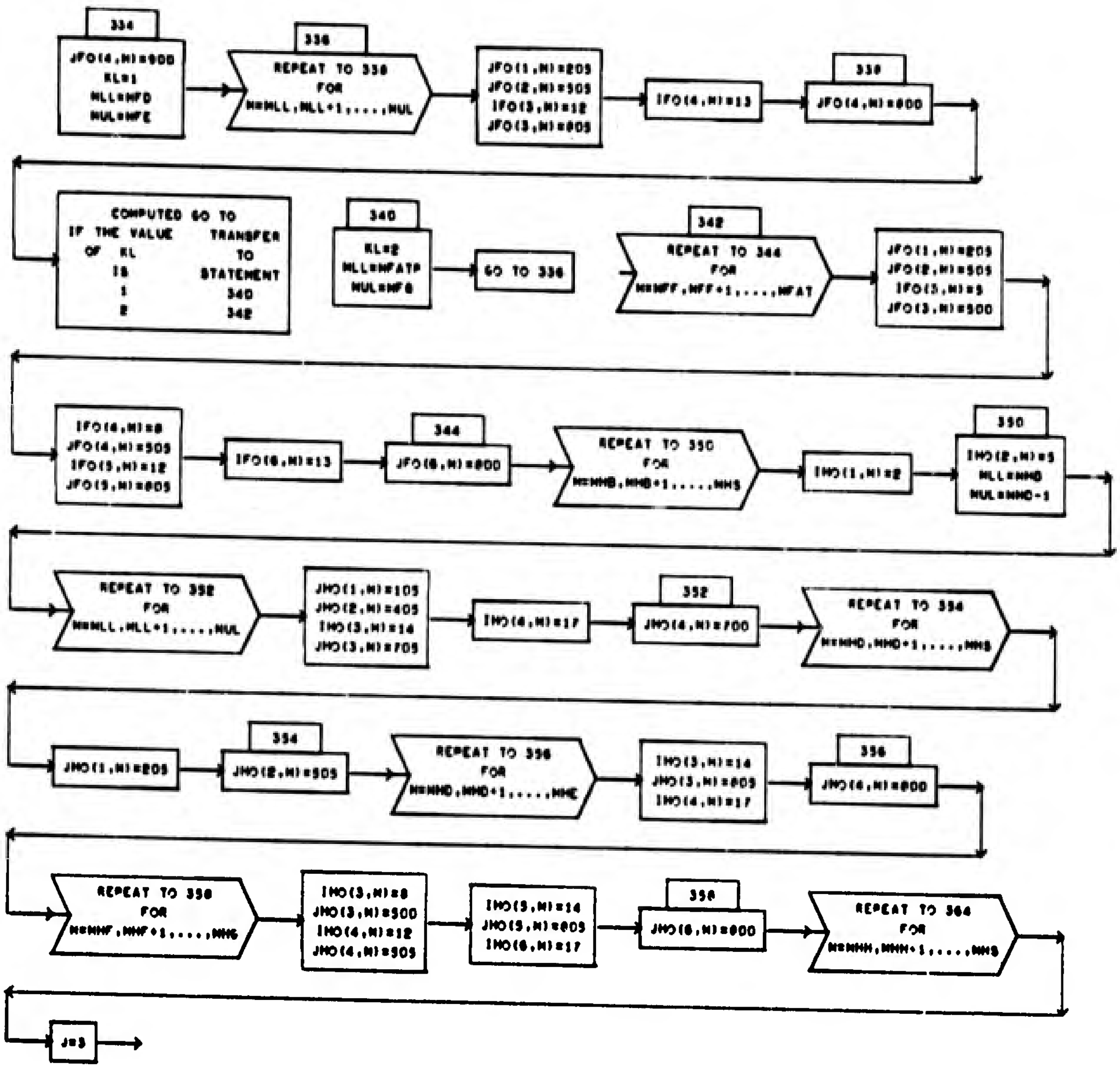
Detailed Flow Charts of Subroutine TEHAI (9 of 37)



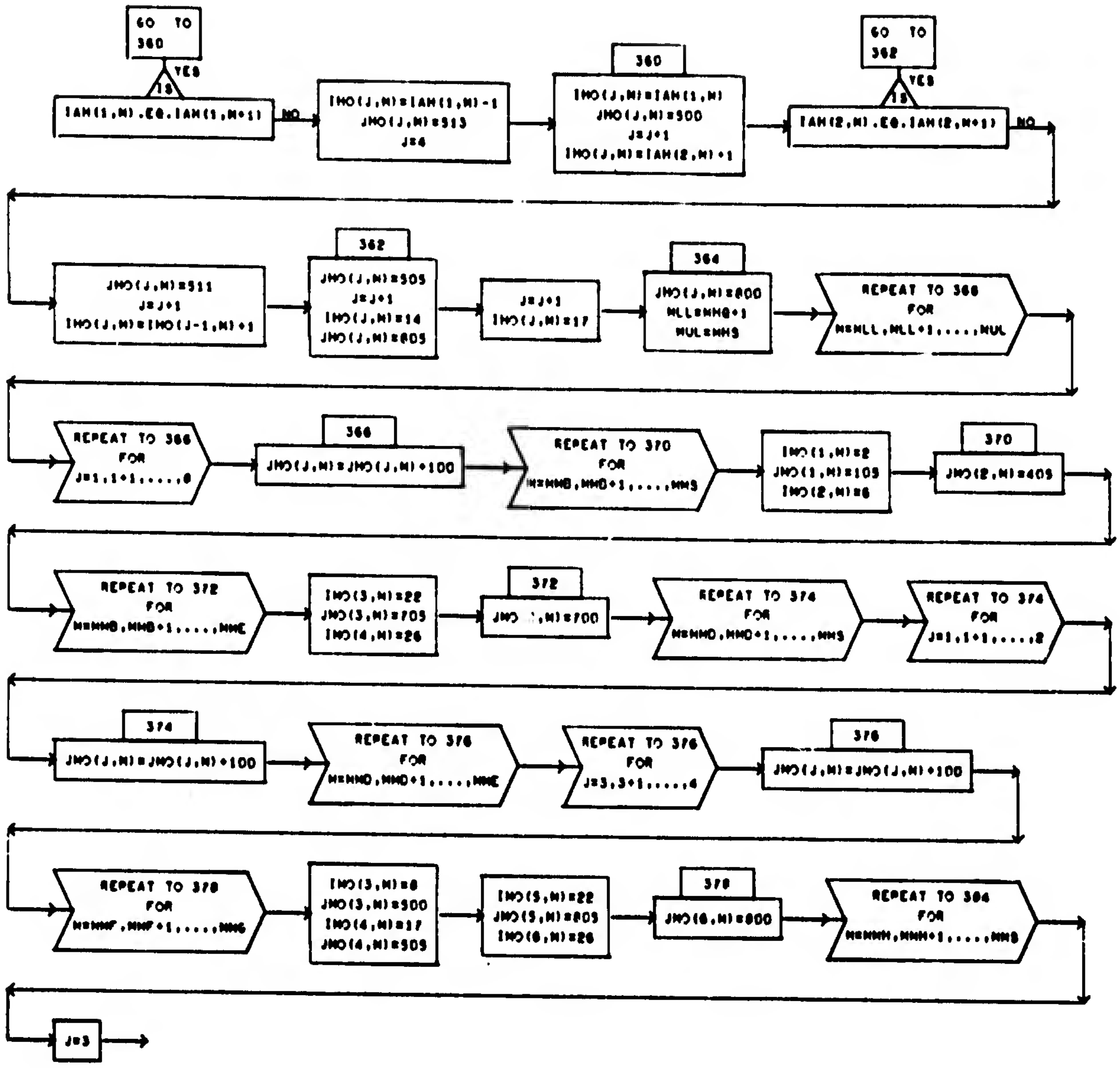
Detailed Flow Charts of Subroutine TEHAI (10 of 37)



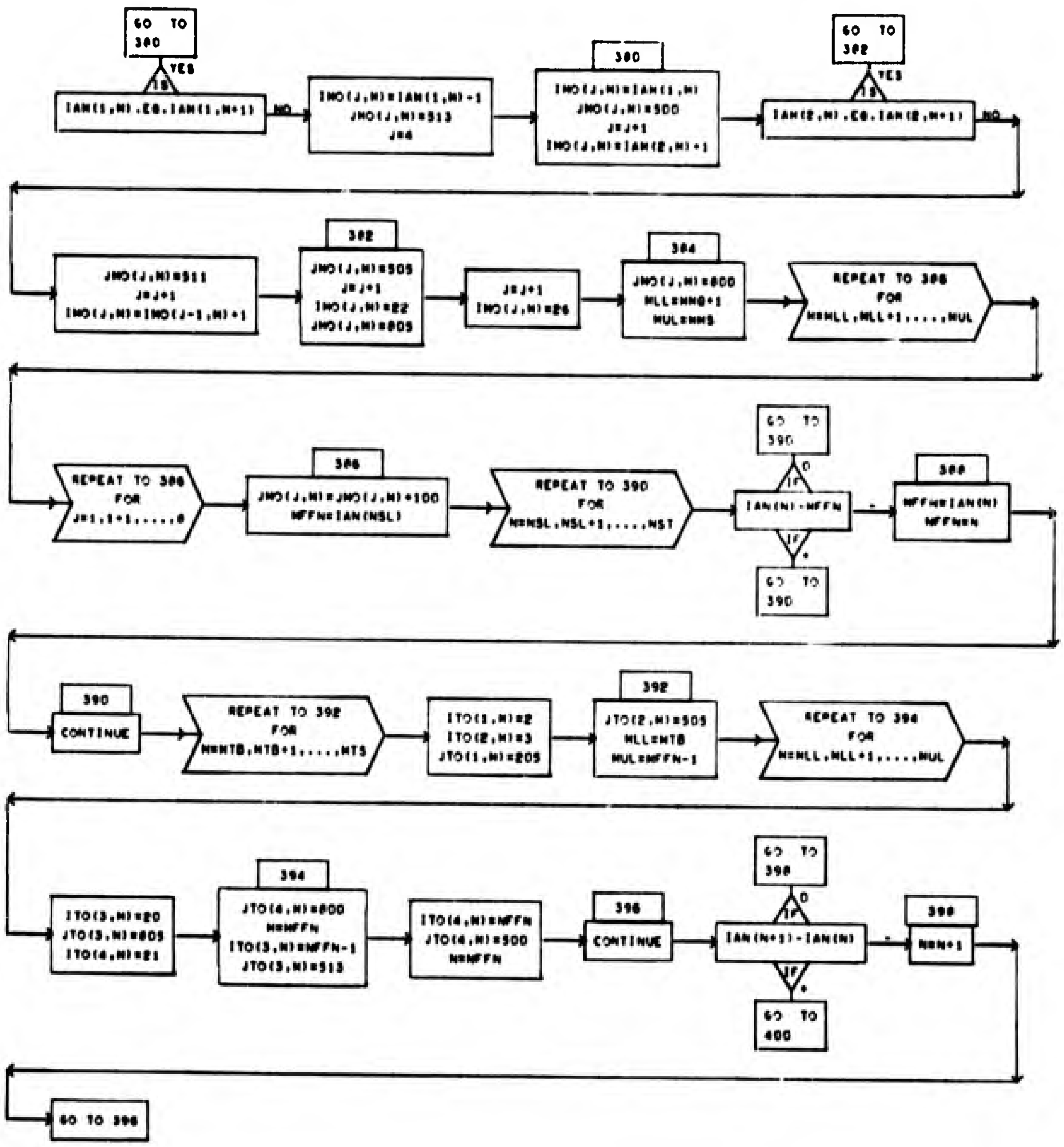
Detailed Flow Charts of Subroutine TEHAI (11 of 37)



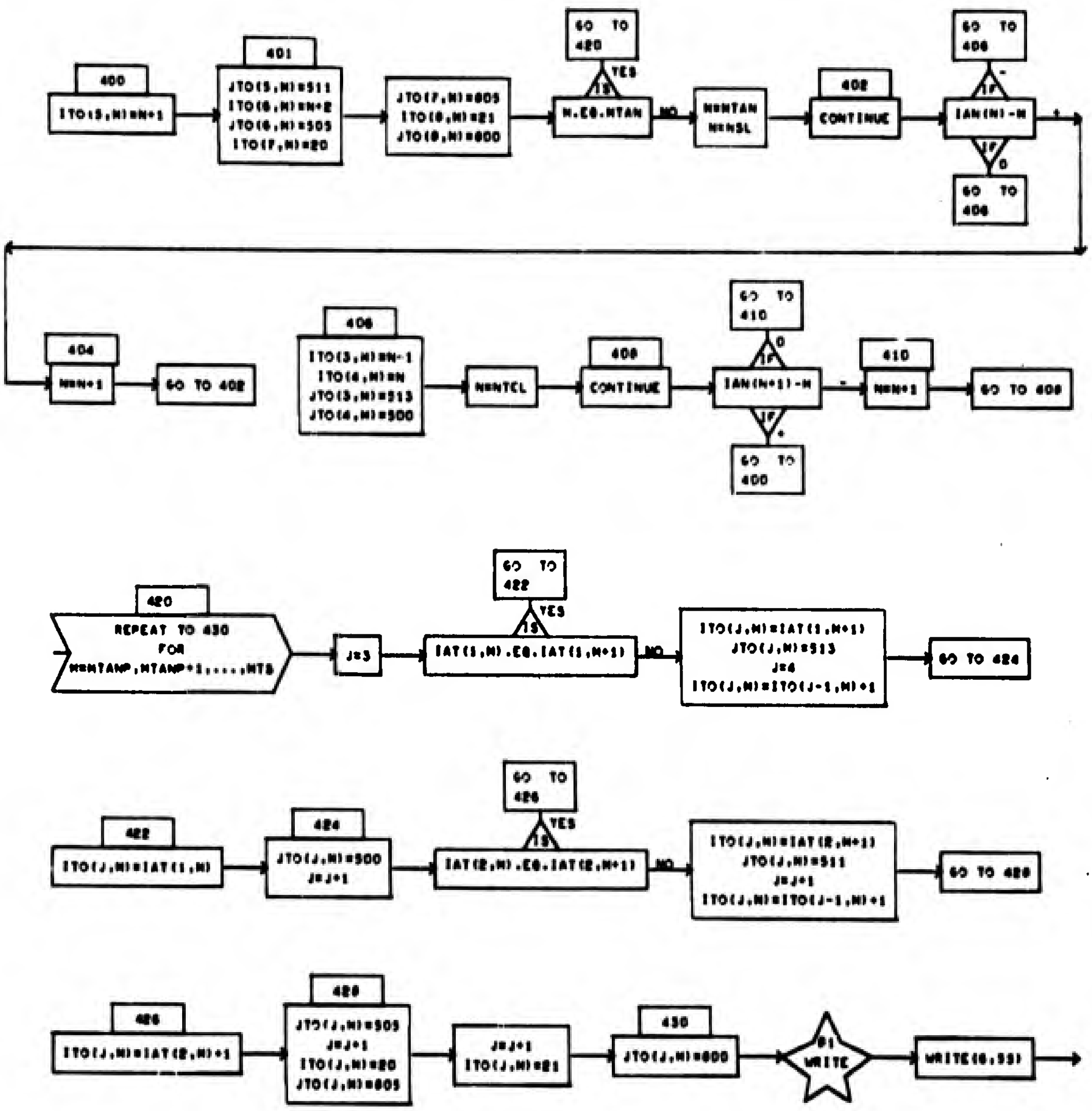
Detailed Flow Charts of Subroutine TEHAI (12 of 37)



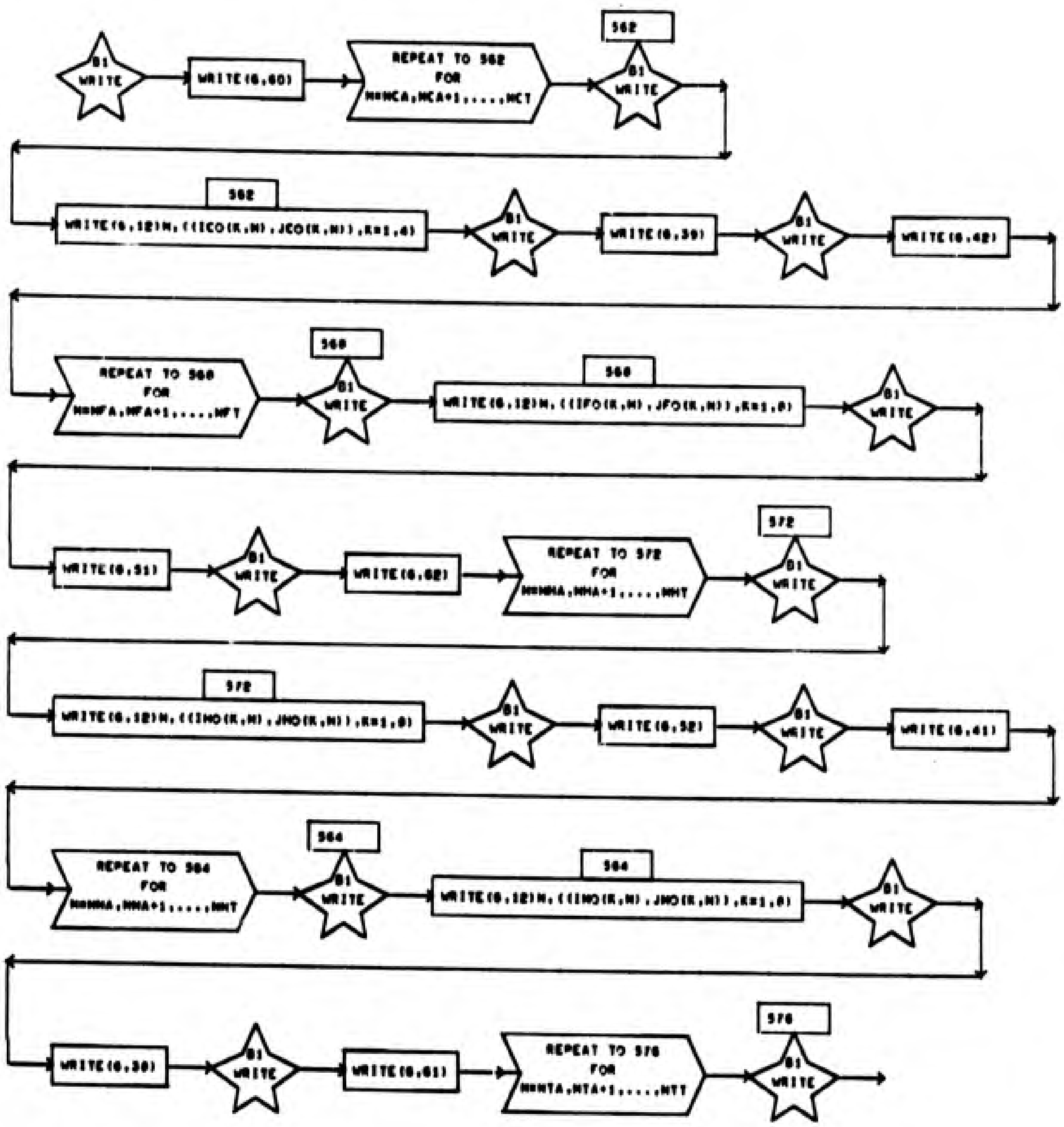
Detailed Flow Charts of Subroutine TEHAI (13 of 37)



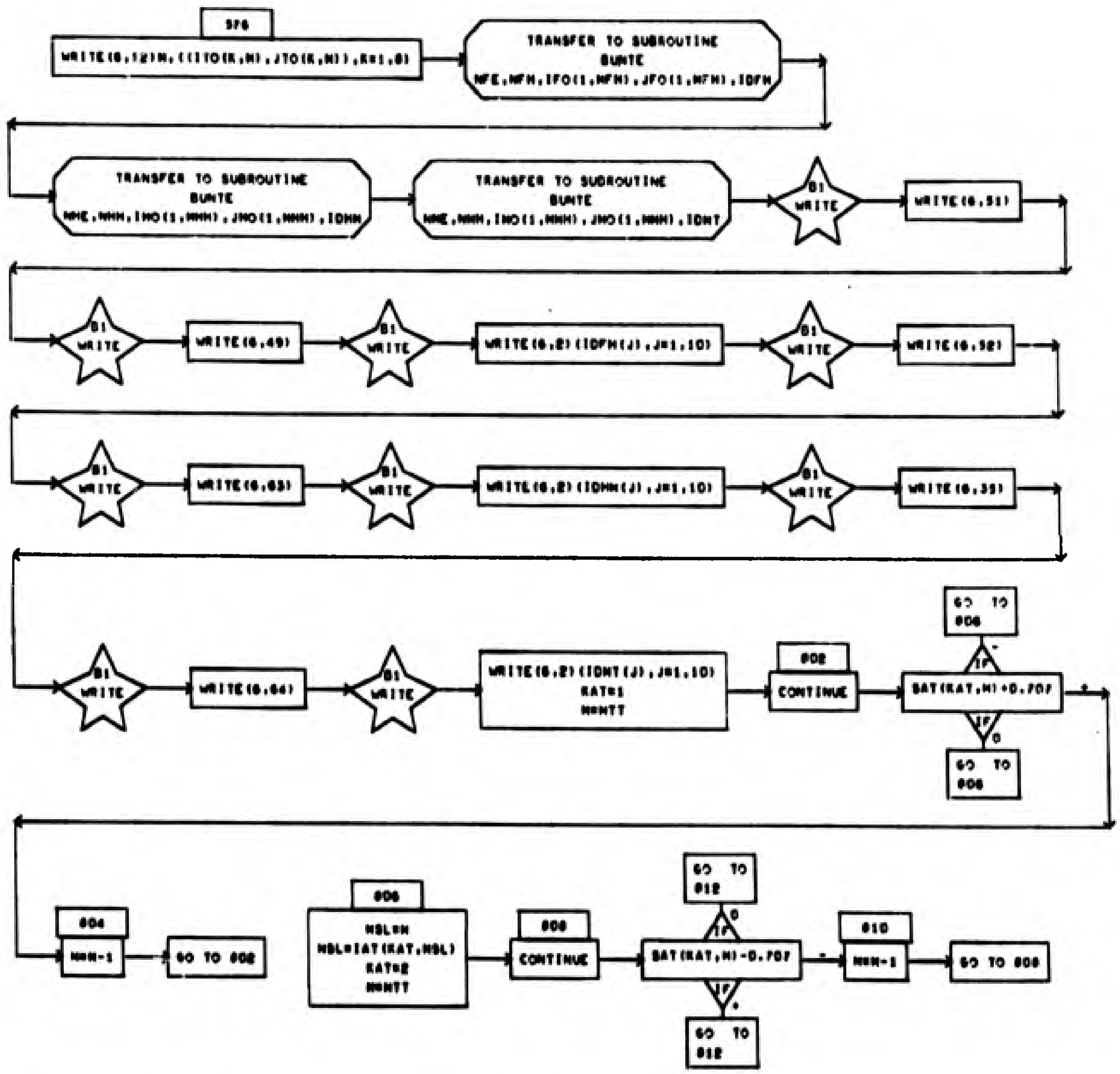
Detailed Flow Charts of Subroutine TEHAI (14 of 37)



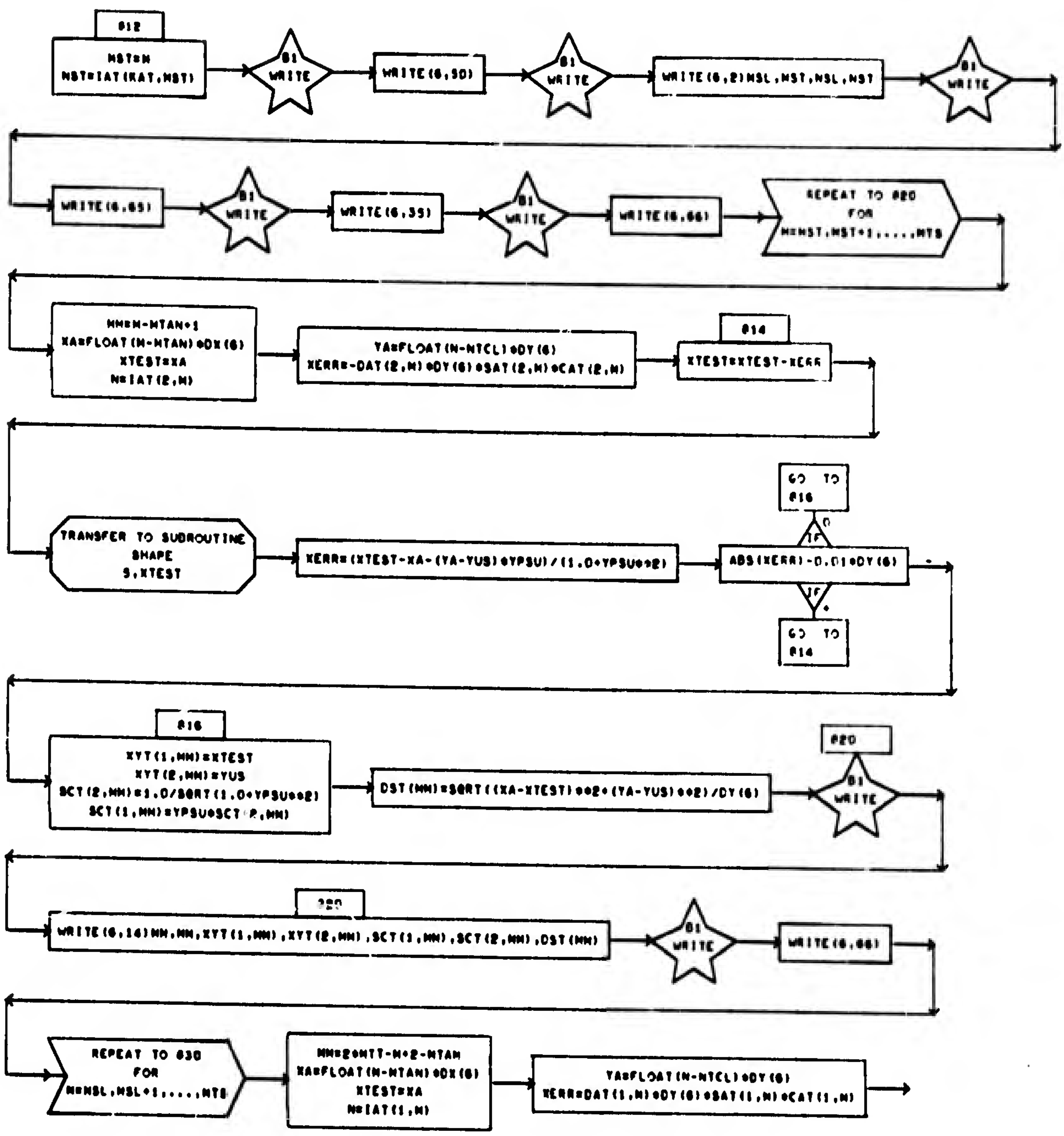
Detailed Flow Charts of Subroutine TEHAI (15 of 37)



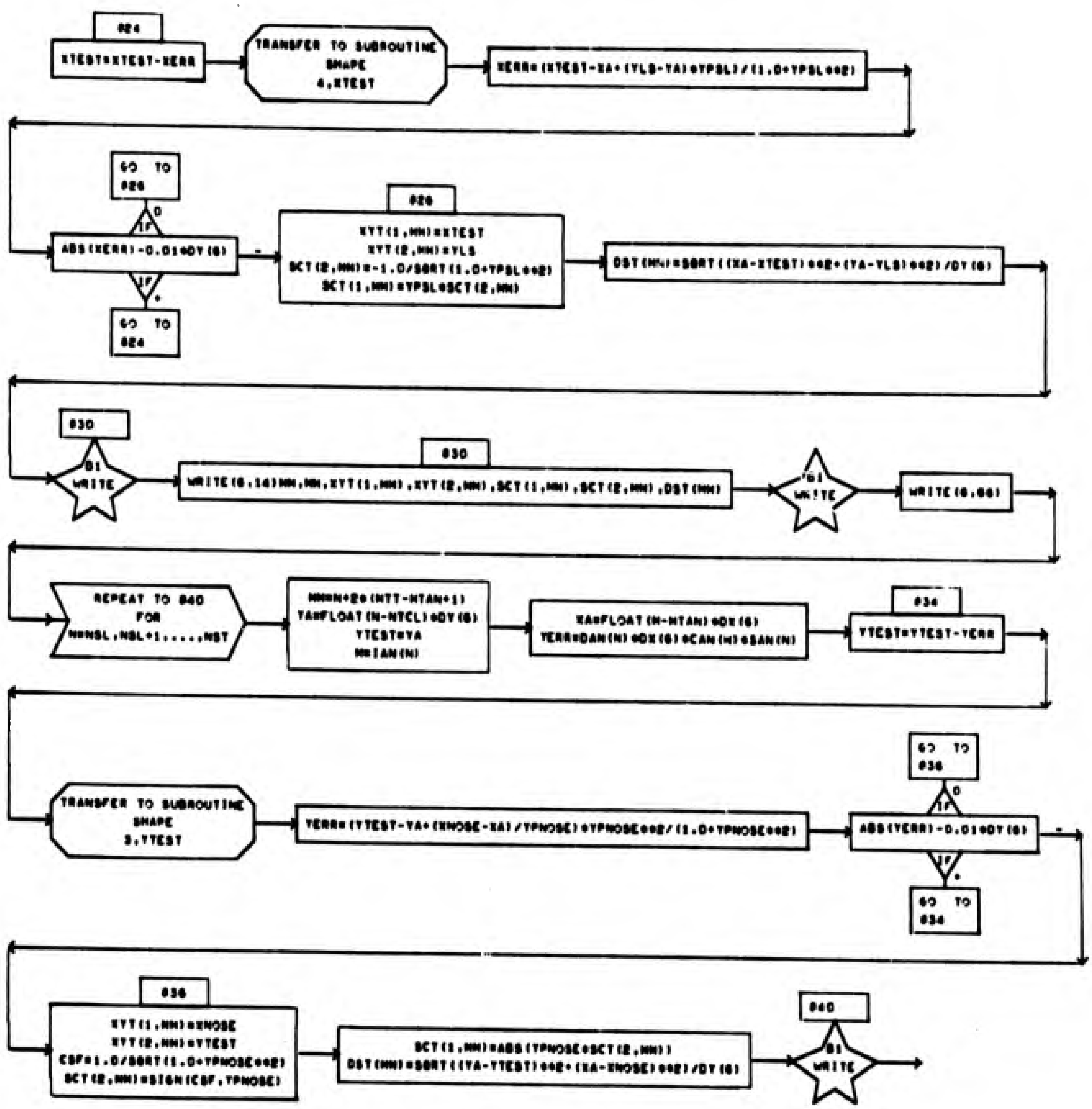
Detailed Flow Charts of Subroutine TEHAI (16 of 37)



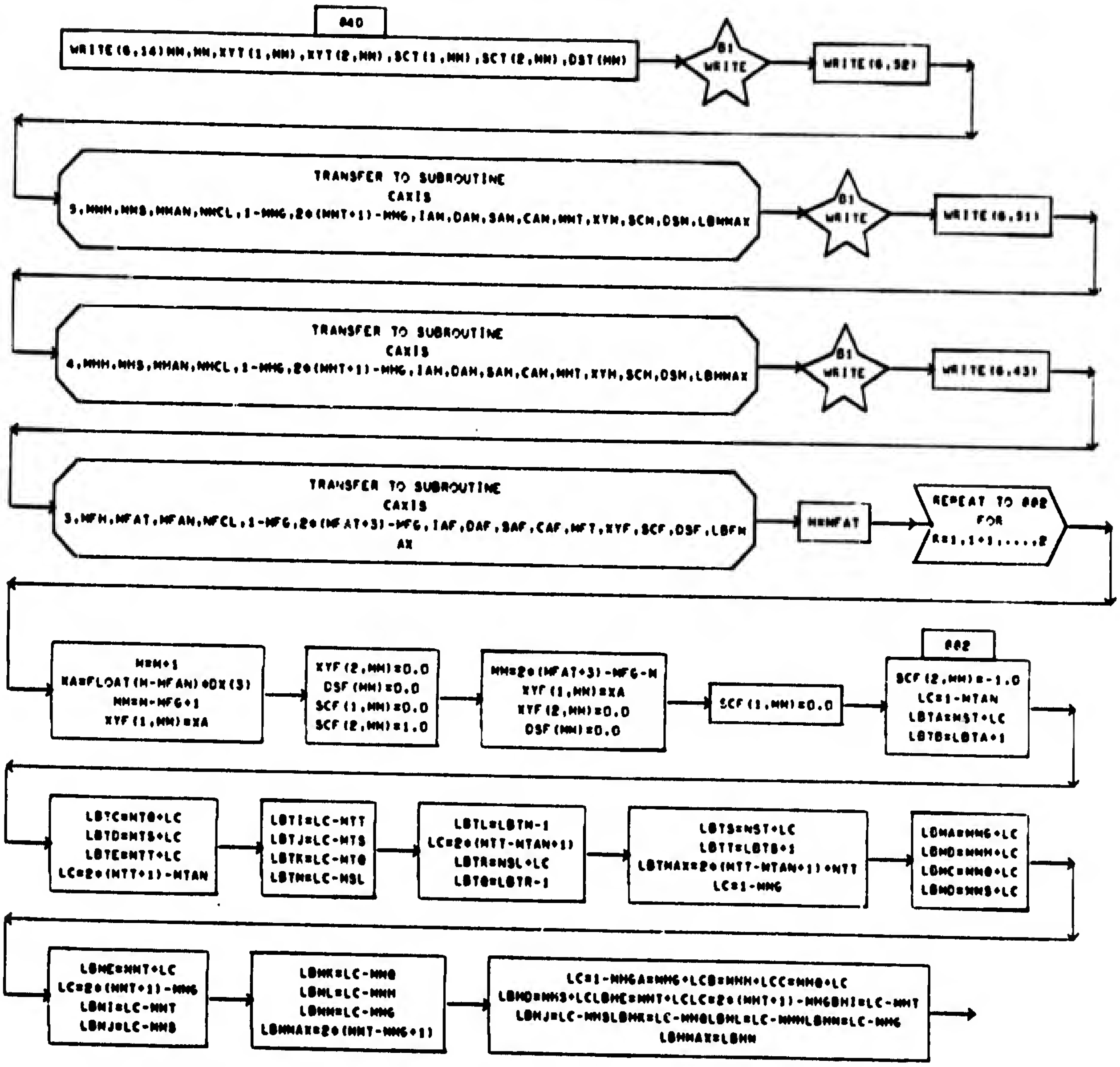
Detailed Flow Charts of Subroutine TEHAI (17 of 37)



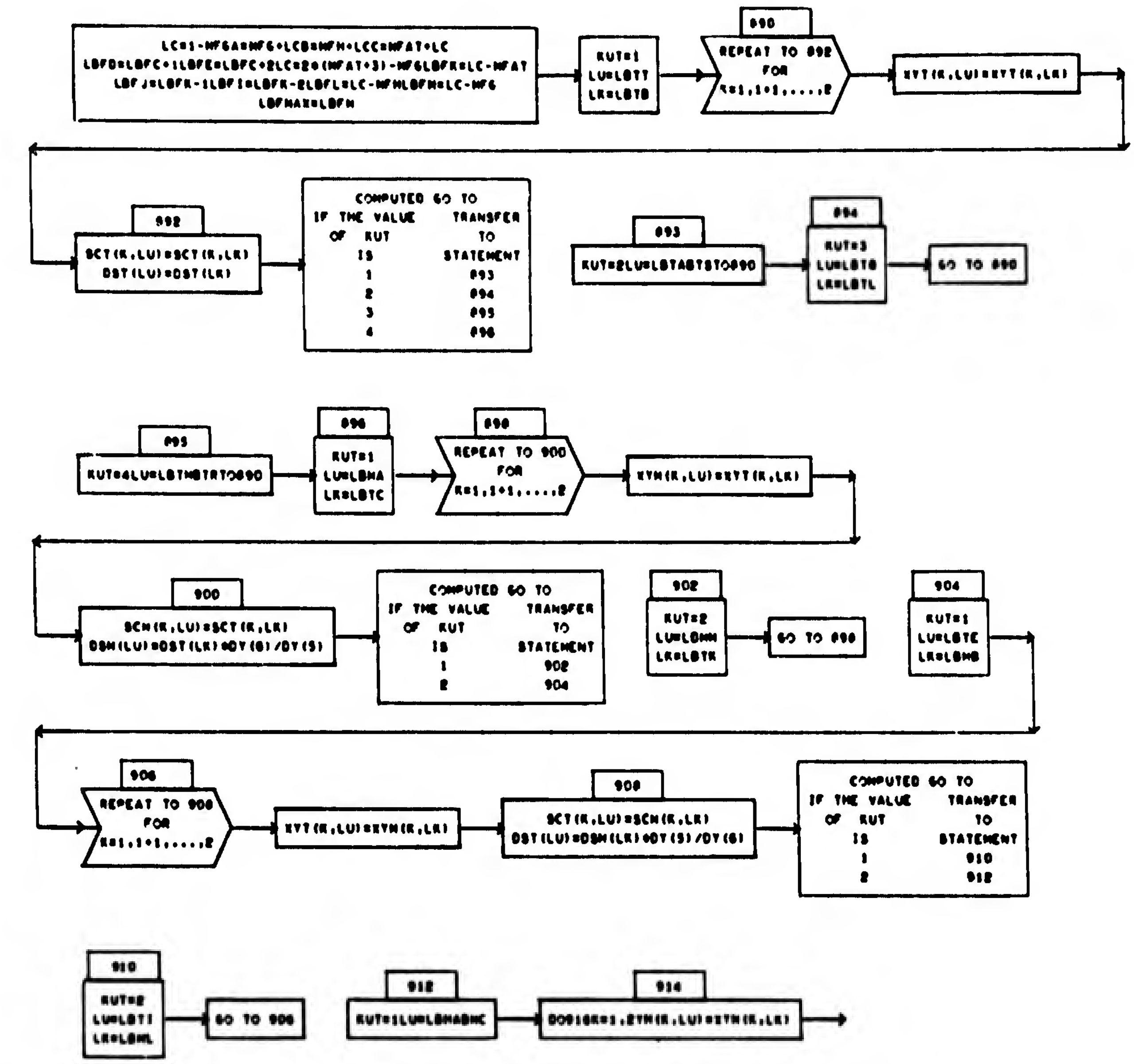
Detailed Flow Charts of Subroutine TEHAI (18 of 37)



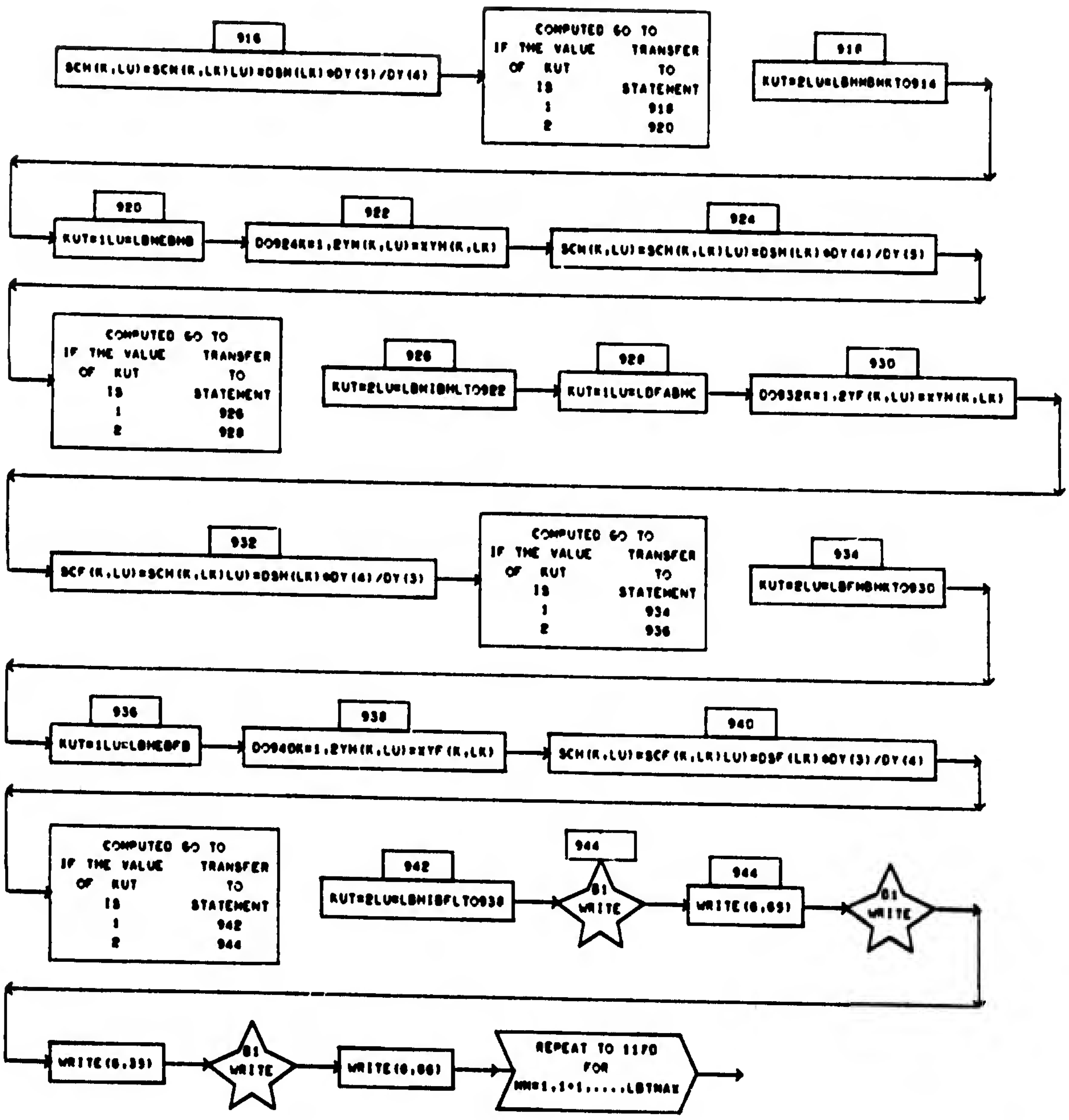
Detailed Flow Charts of Subroutine TEHAI (19 of 37)



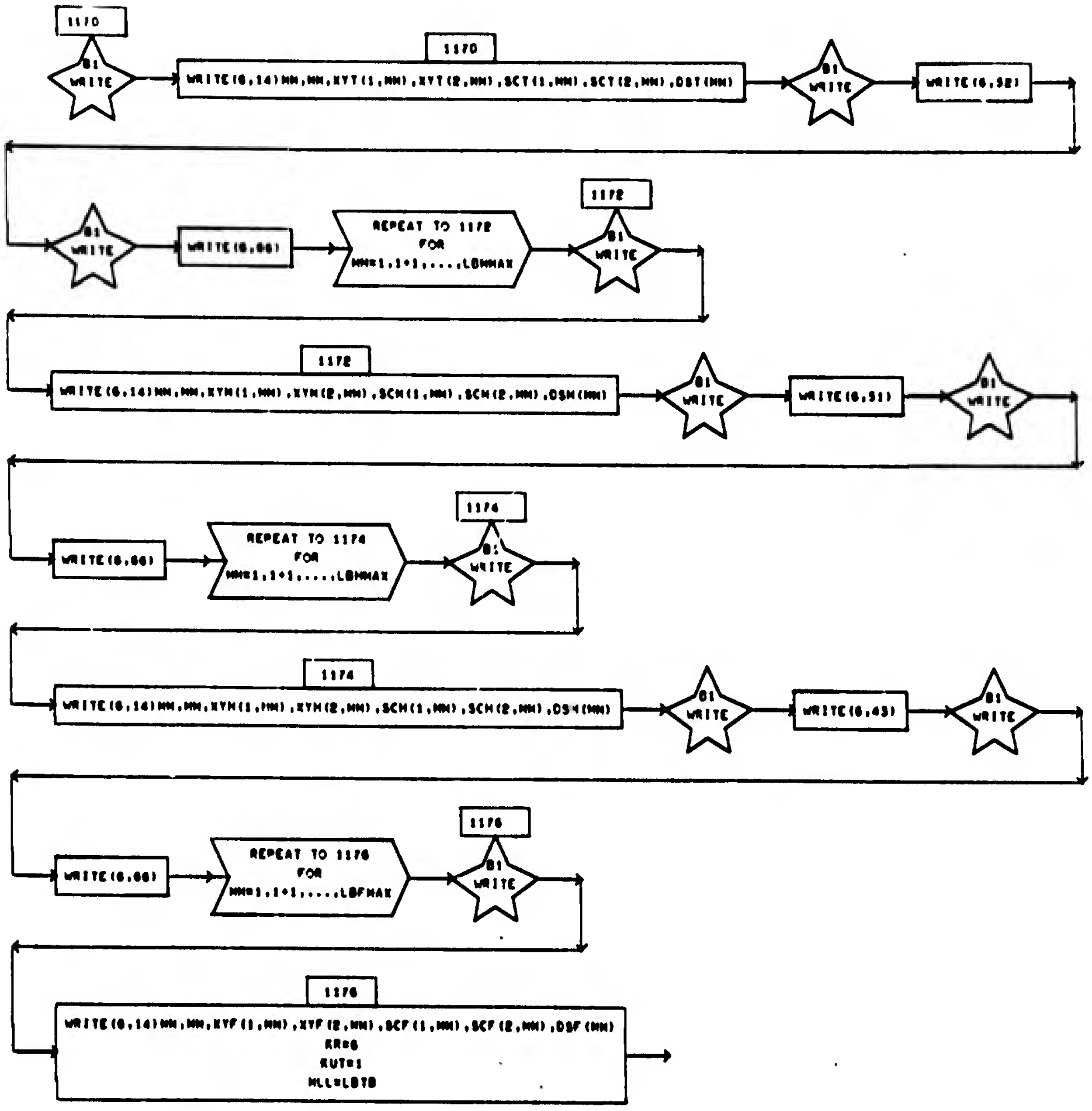
Detailed Flow Charts of Subroutine TEHAI (20 of 37)



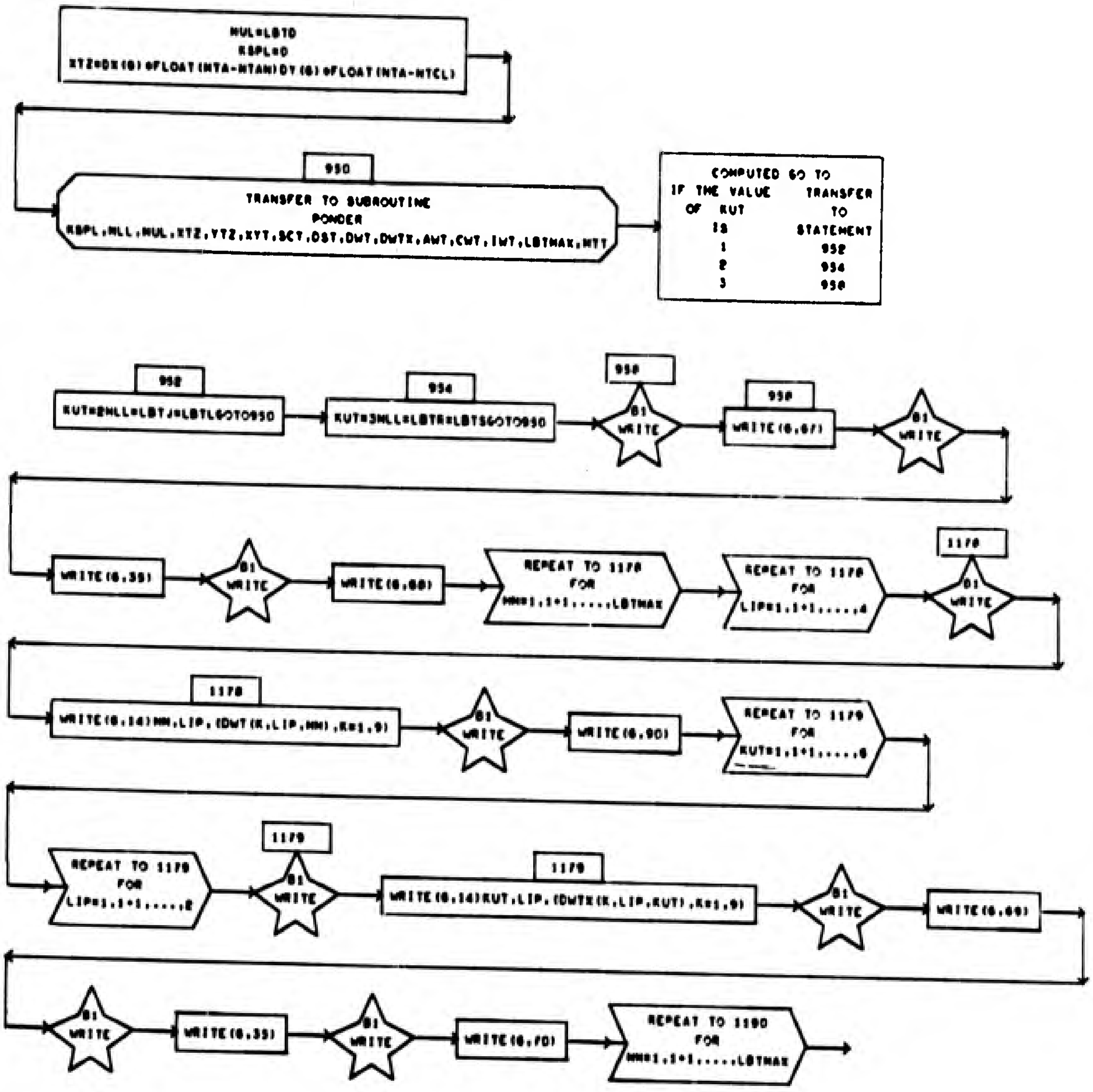
Detailed Flow Charts of Subroutine TEHAI (21 of 37)



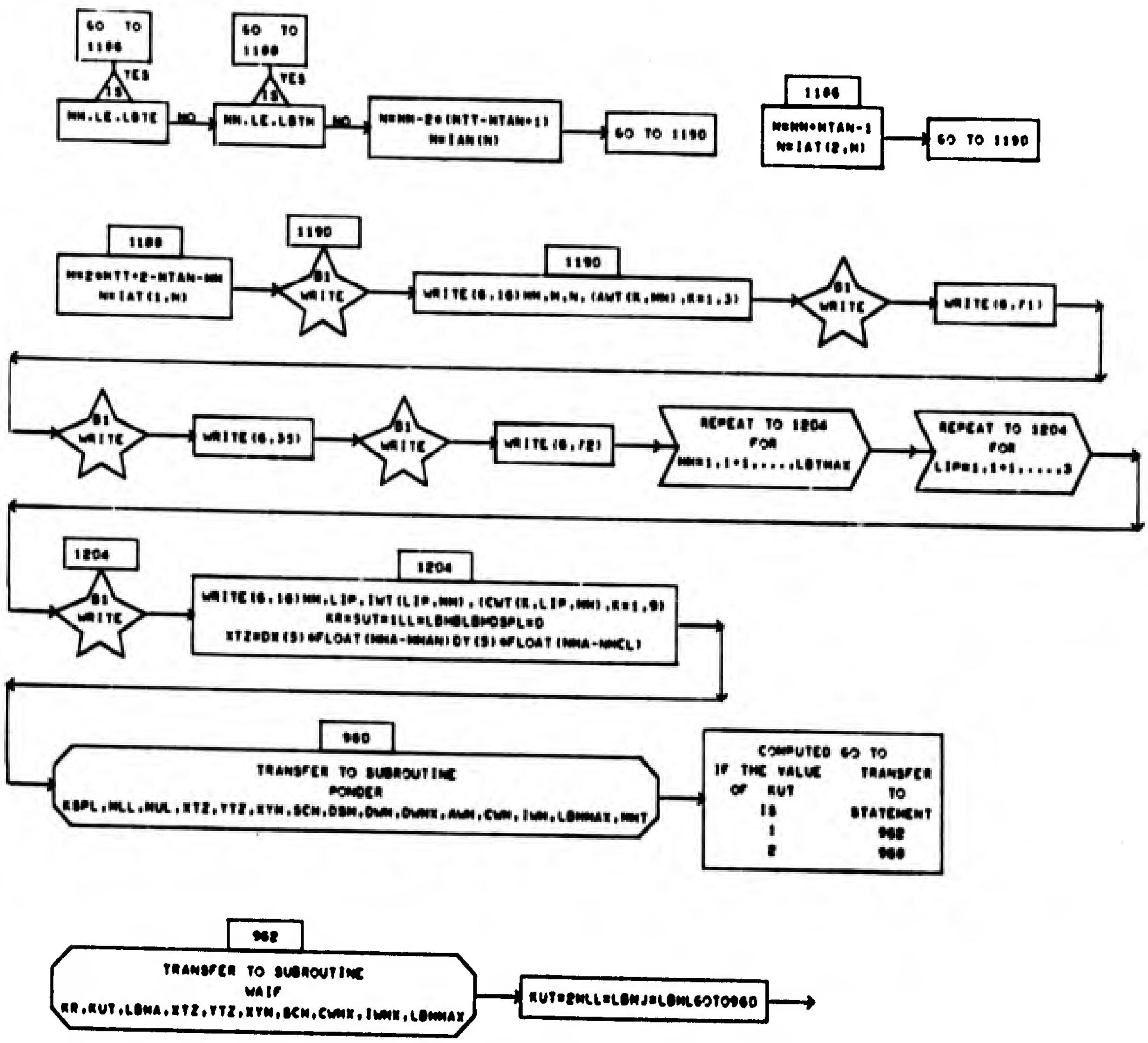
Detailed Flow Charts of Subroutine TEHAI (22 of 37)



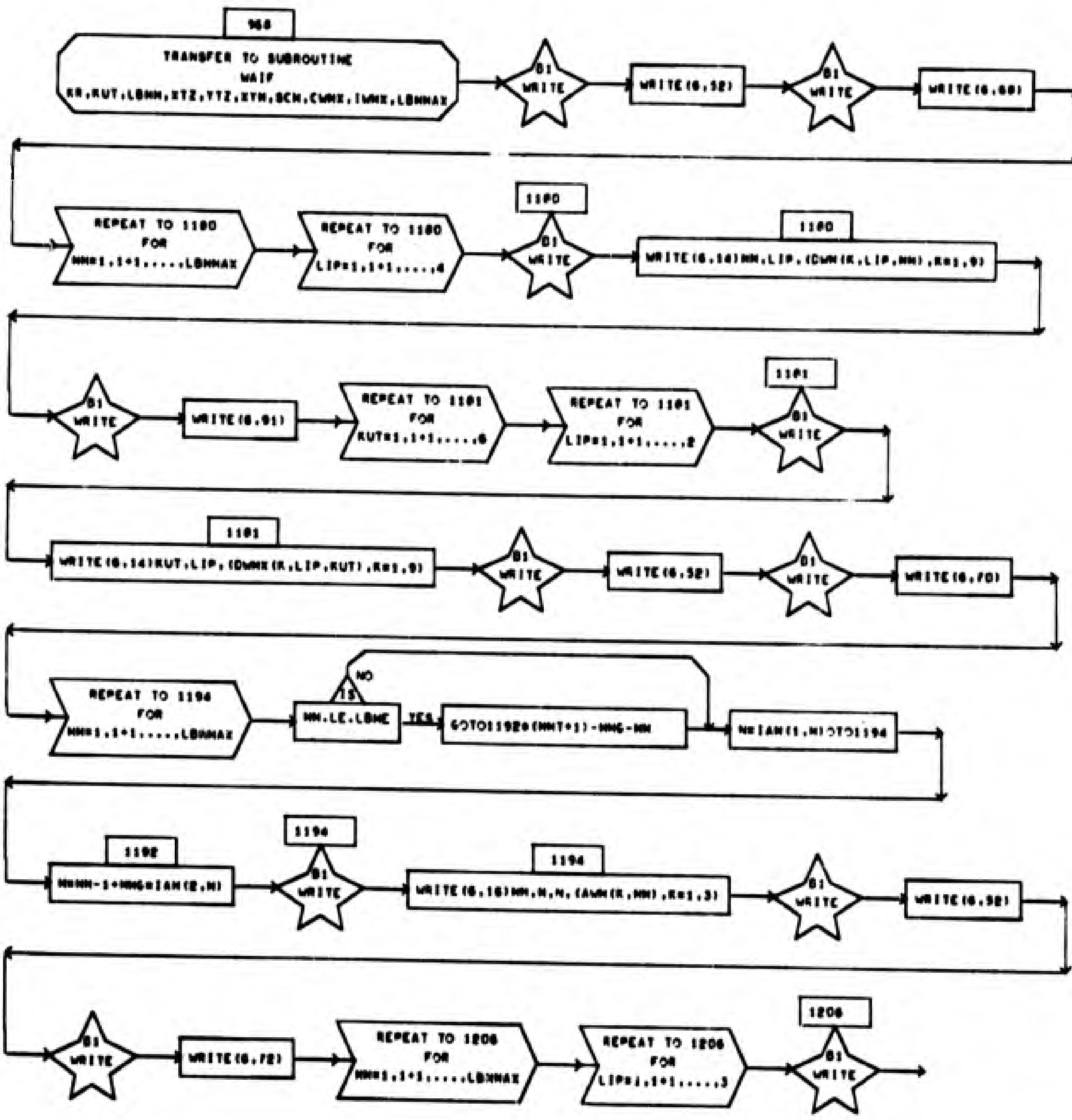
Detailed Flow Charts of Subroutine TEHAI (23 of 37)



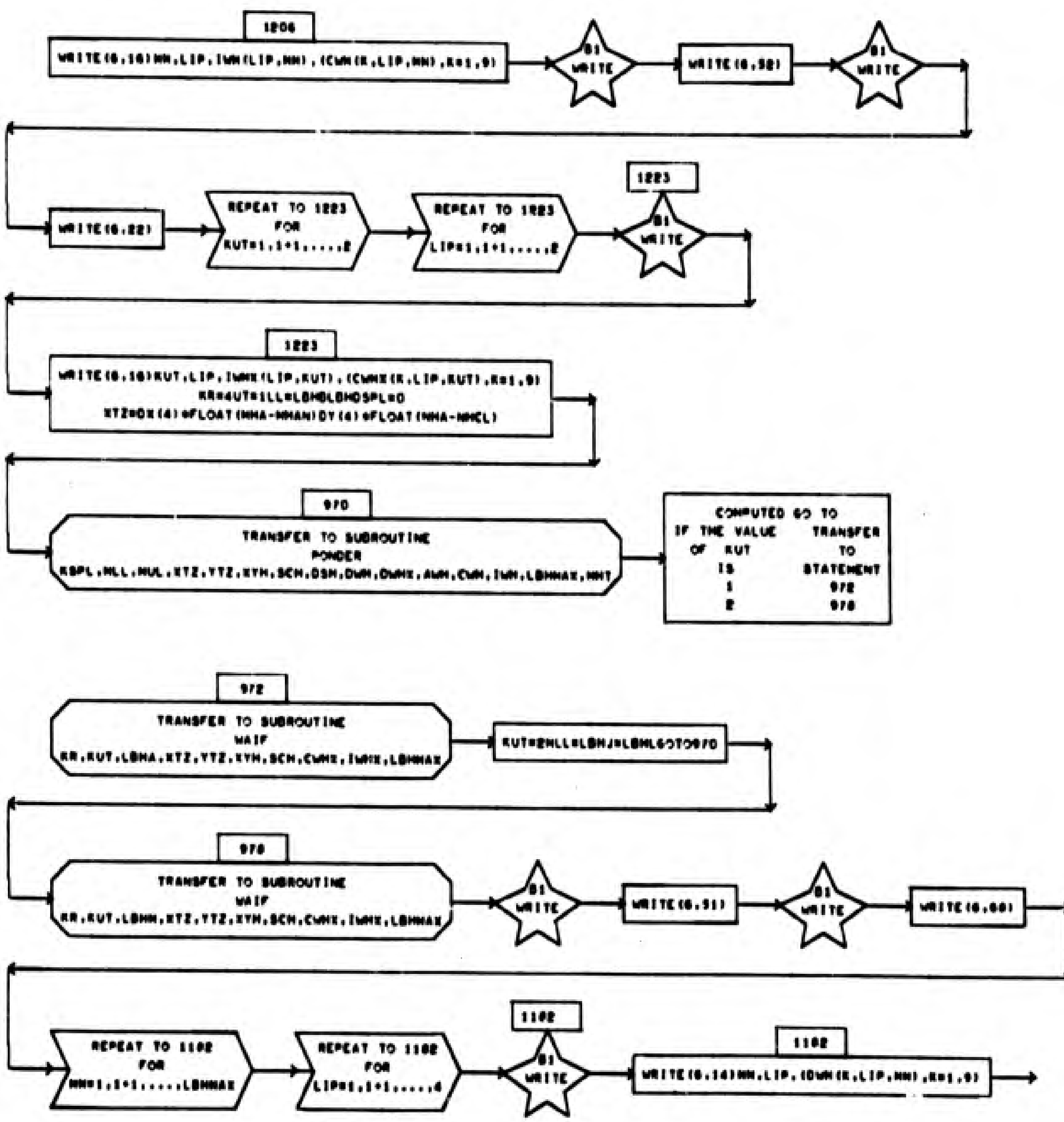
Detailed Flow Charts of Subroutine TEHAI (24 of 37)



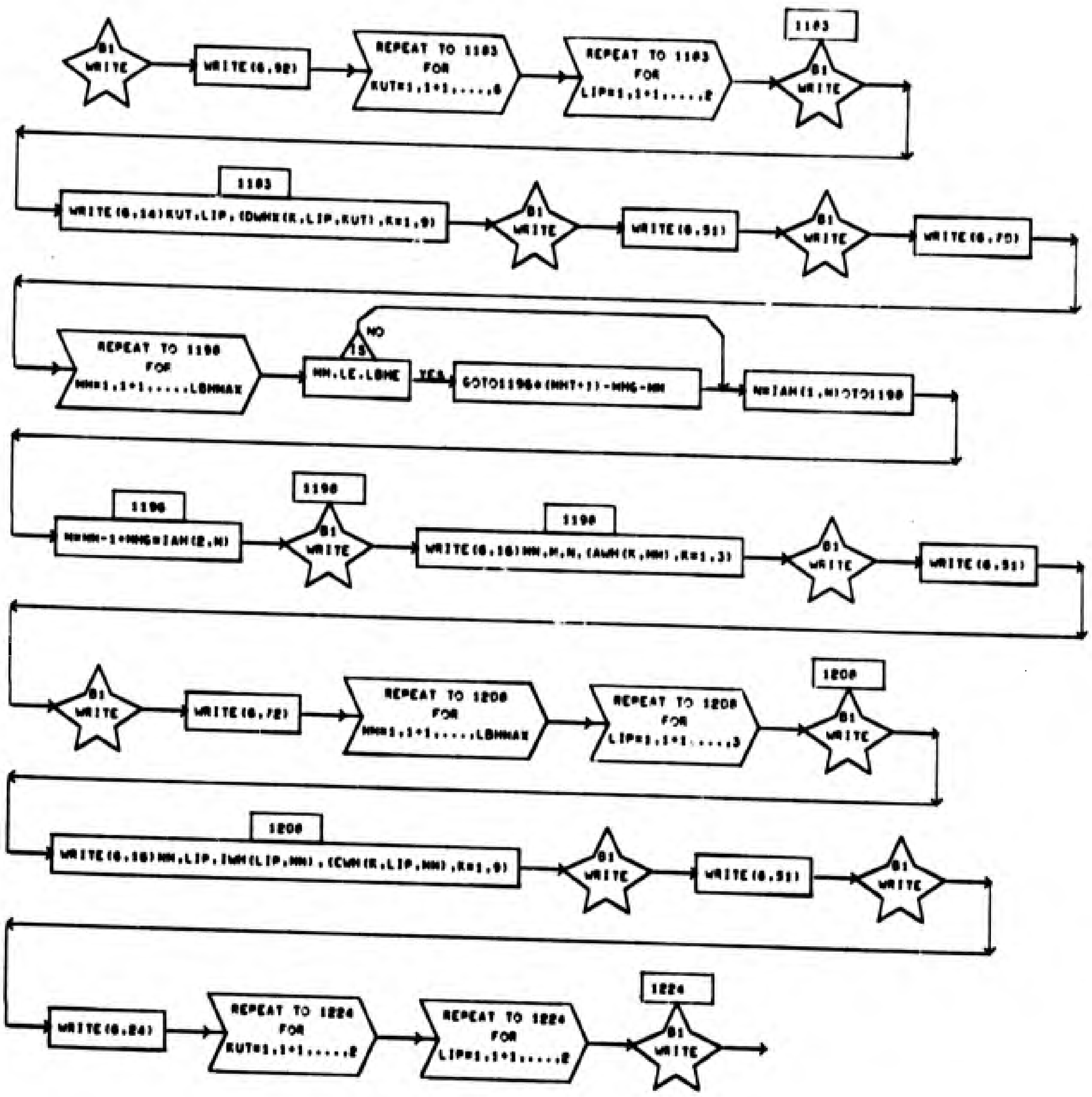
Detailed Flow Charts of Subroutine TEHAI (25 of 37)



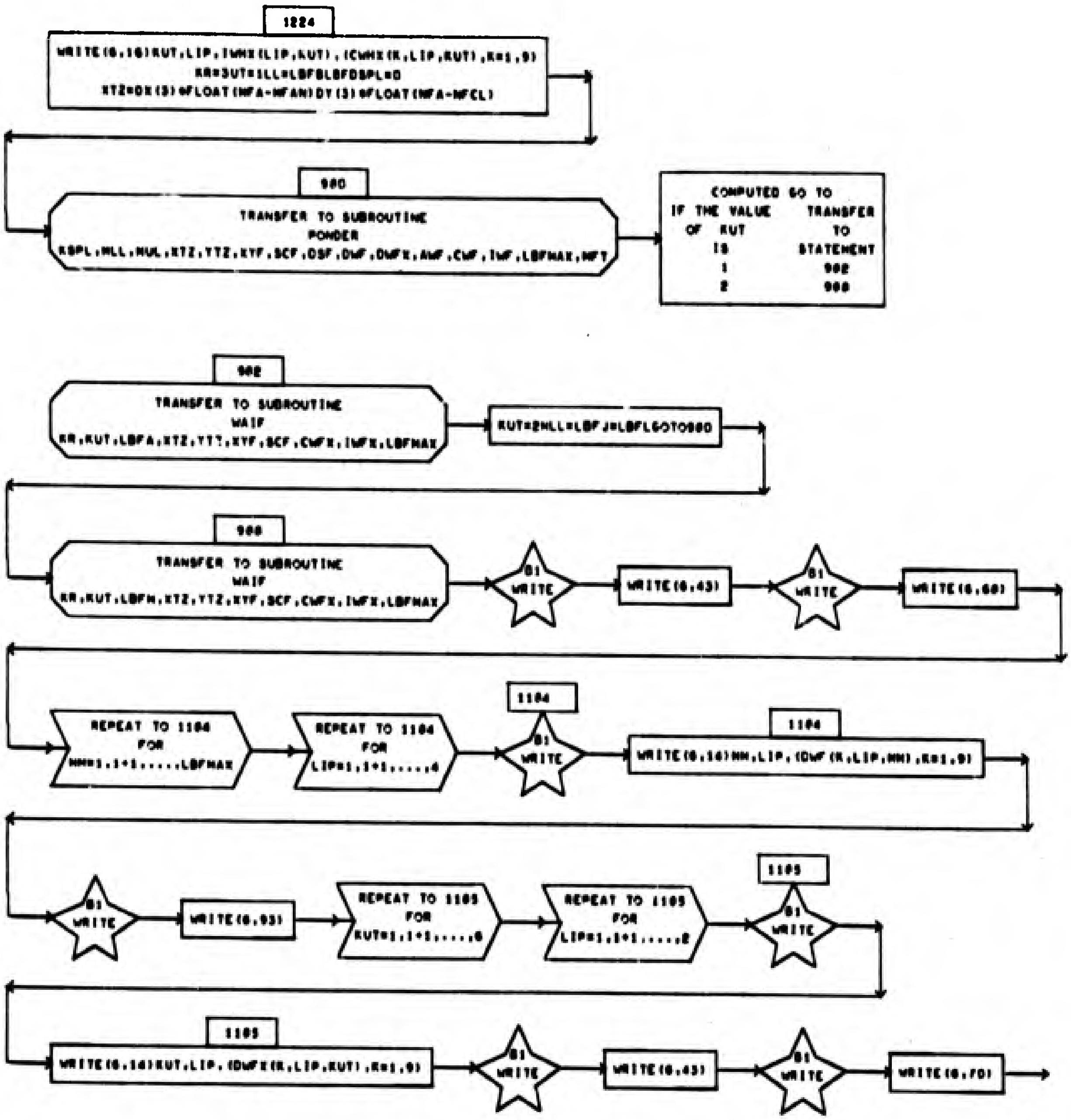
Detailed Flow Charts of Subroutine TEHAI (26 of 37)



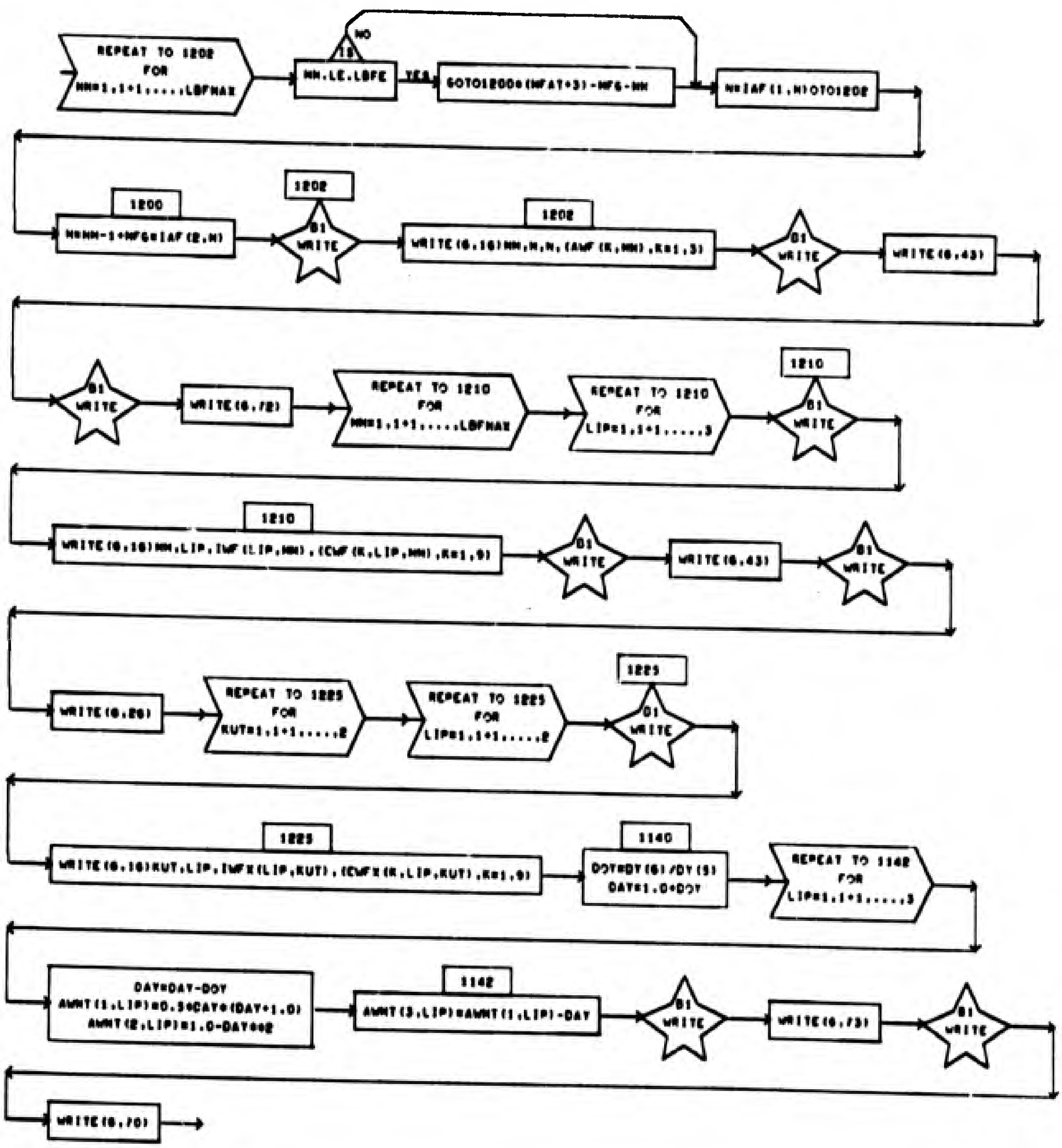
Detailed Flow Charts of Subroutine TEHAI (27 of 37)



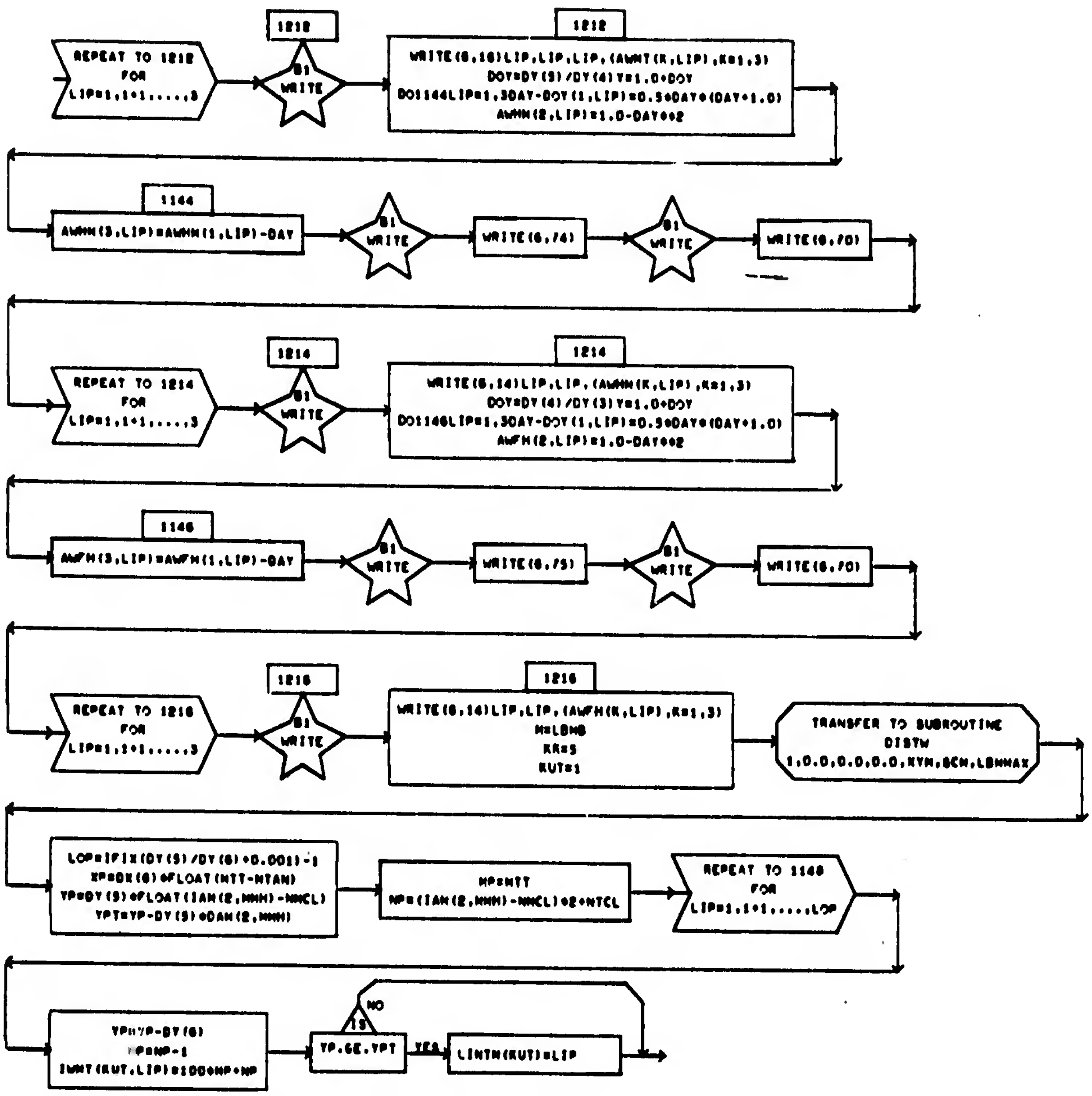
Detailed Flow Charts of Subroutine TEHAI (28 of 37)



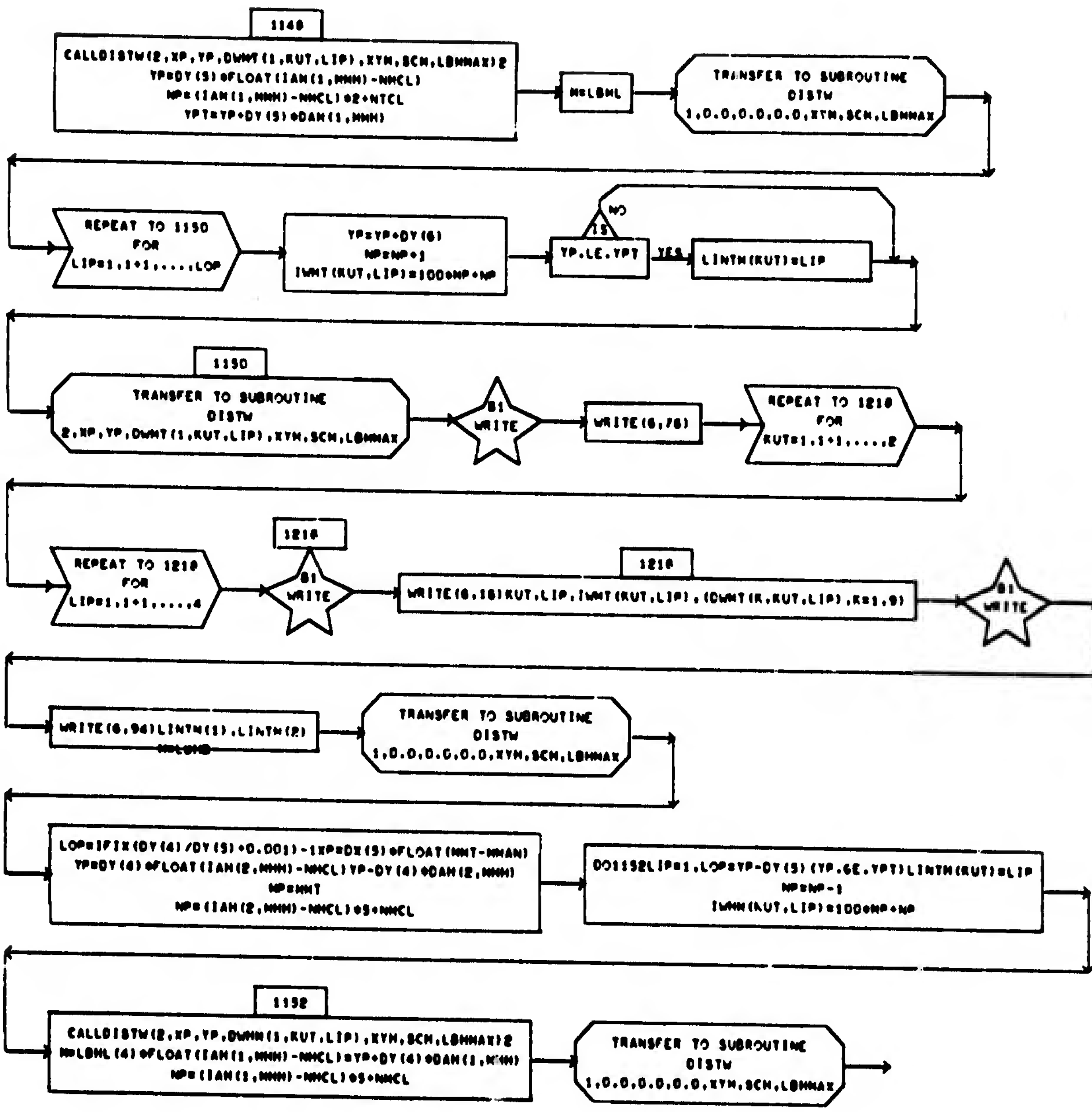
Detailed Flow Charts of Subroutine TEHAI (29 of 37)



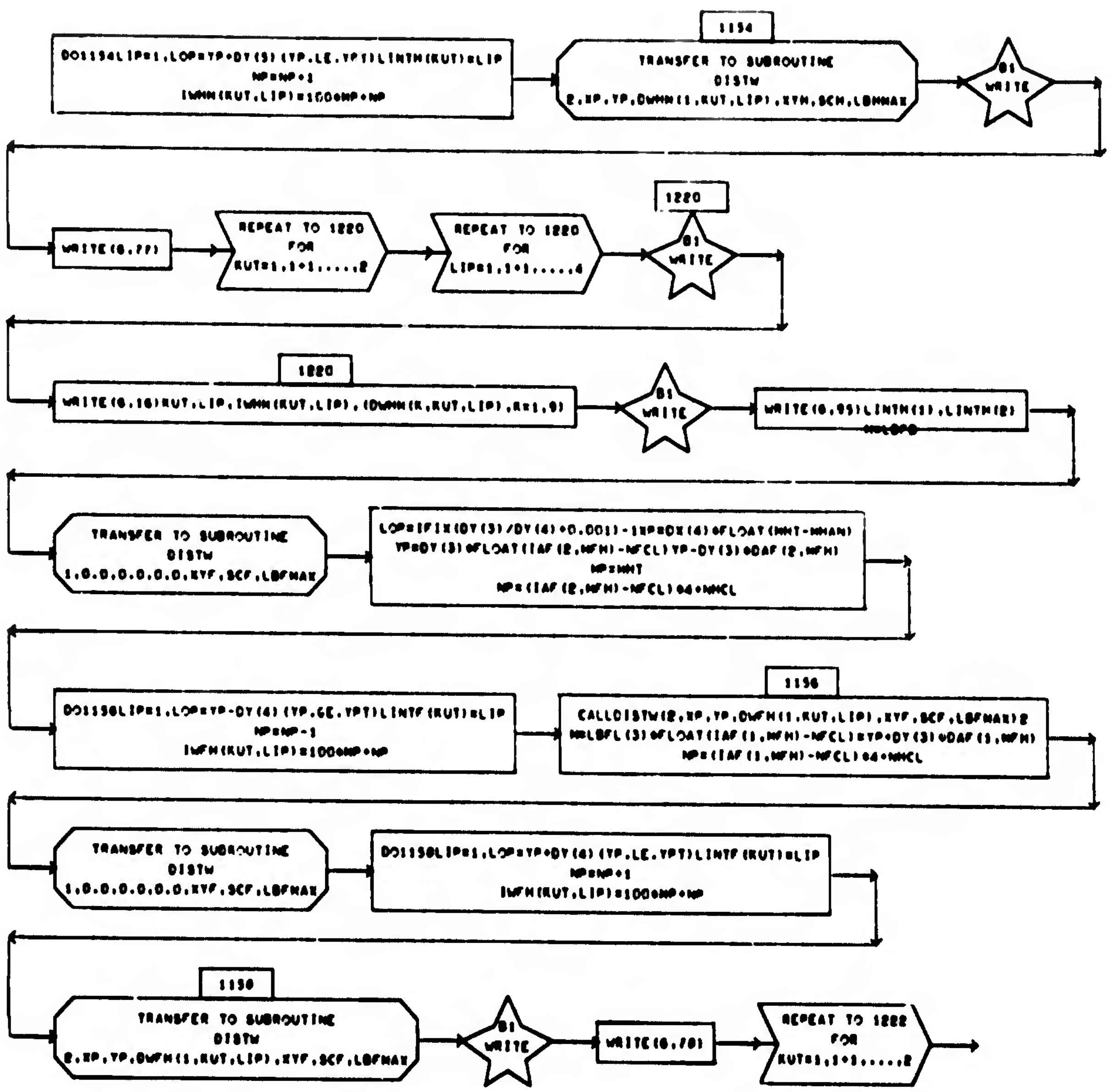
Detailed Flow Charts of Subroutine TEHAI (30 of 37)



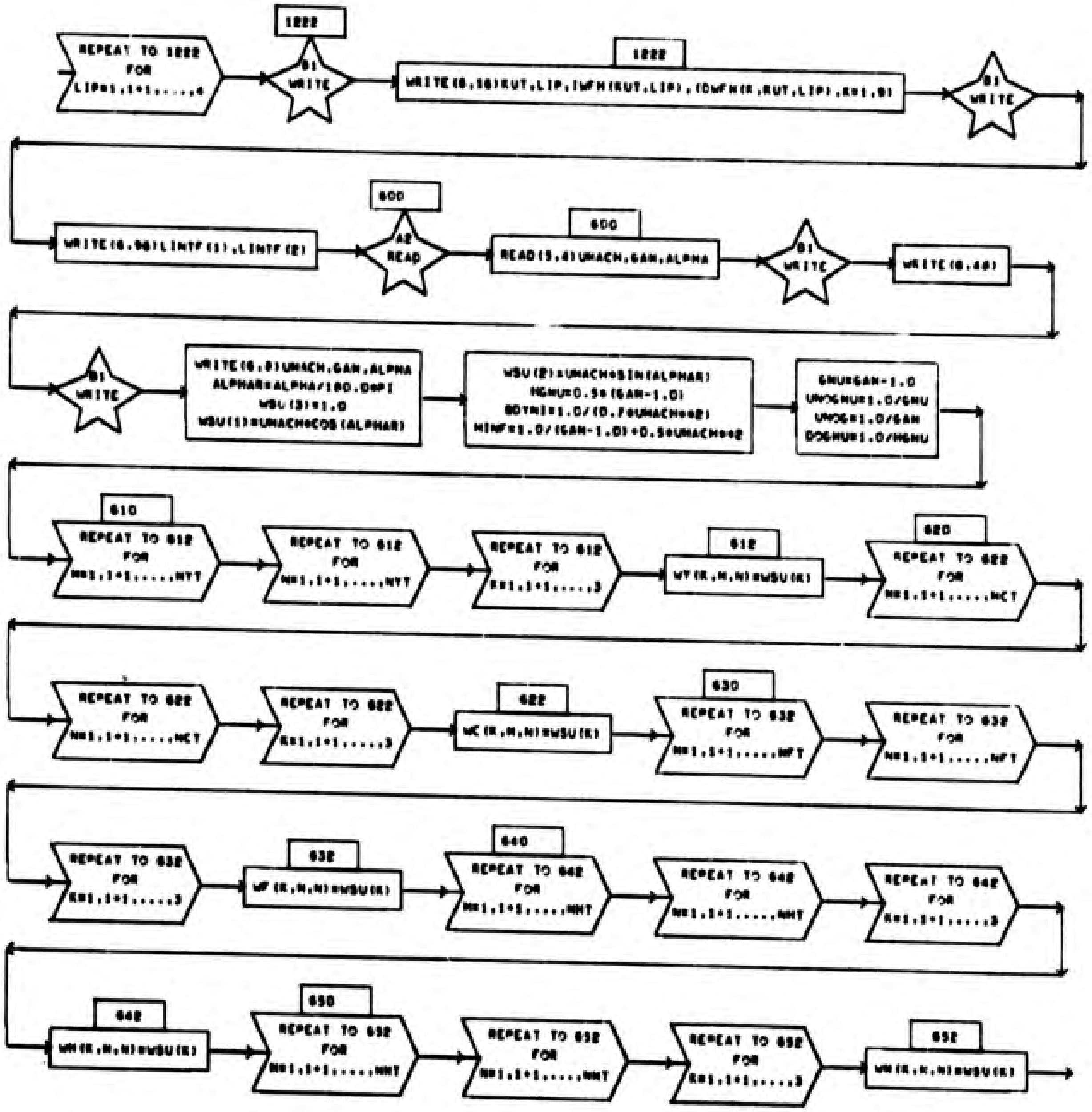
Detailed Flow Charts of Subroutine TEHAI (31 of 37)



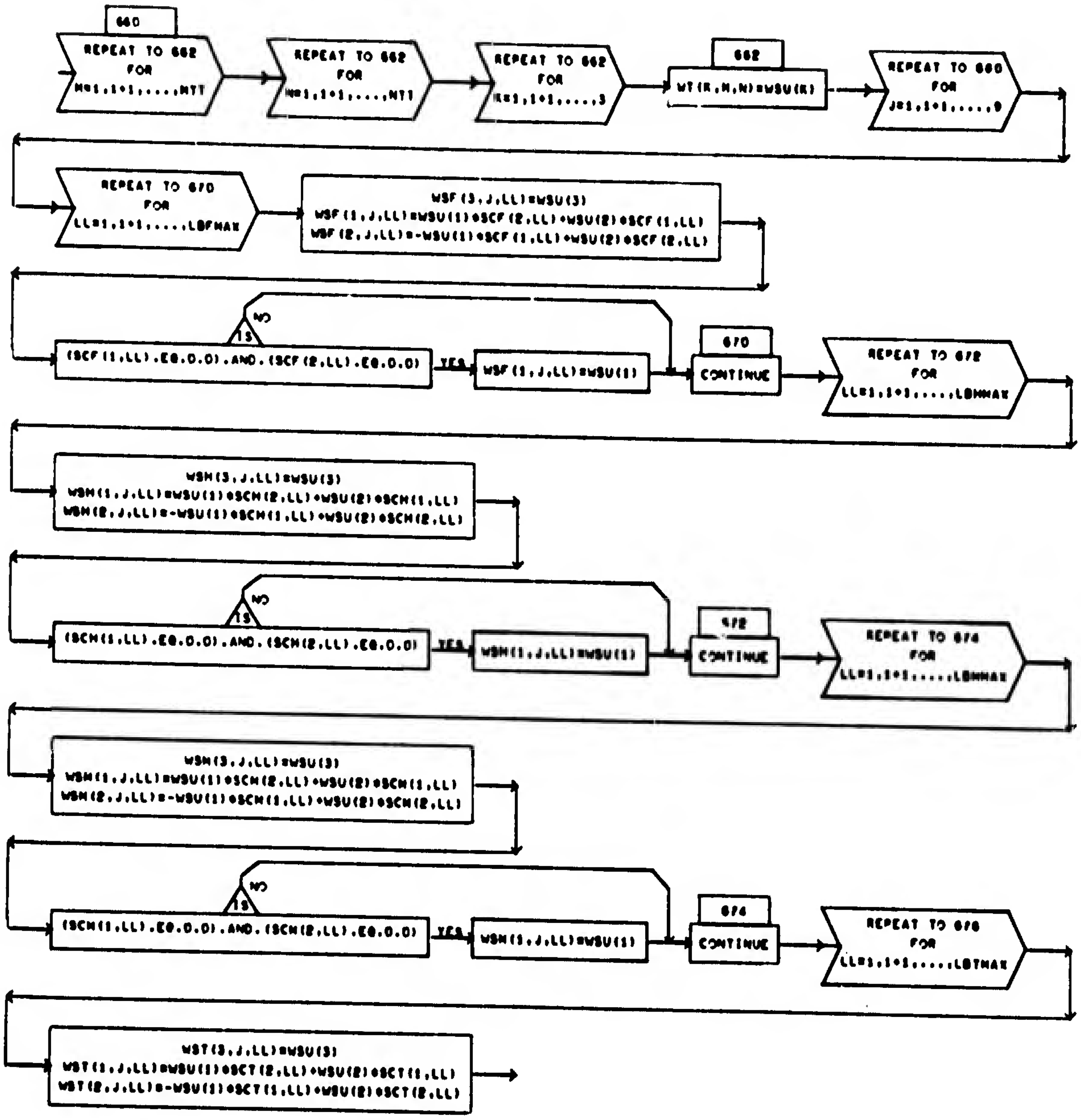
Detailed Flow Charts of Subroutine TEHAI (32 of 37)



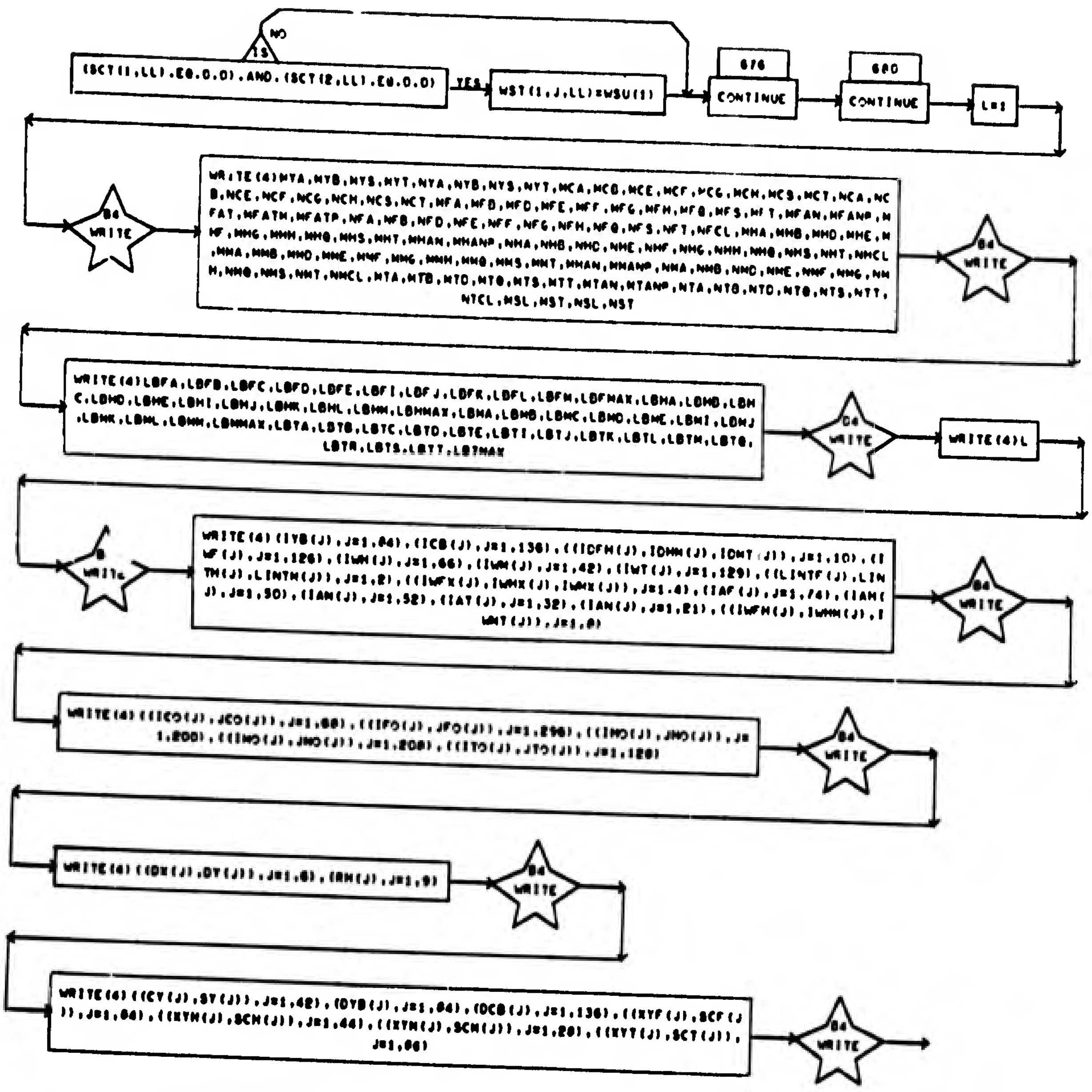
Detailed Flow Charts of Subroutine TEHAI (33 of 37)



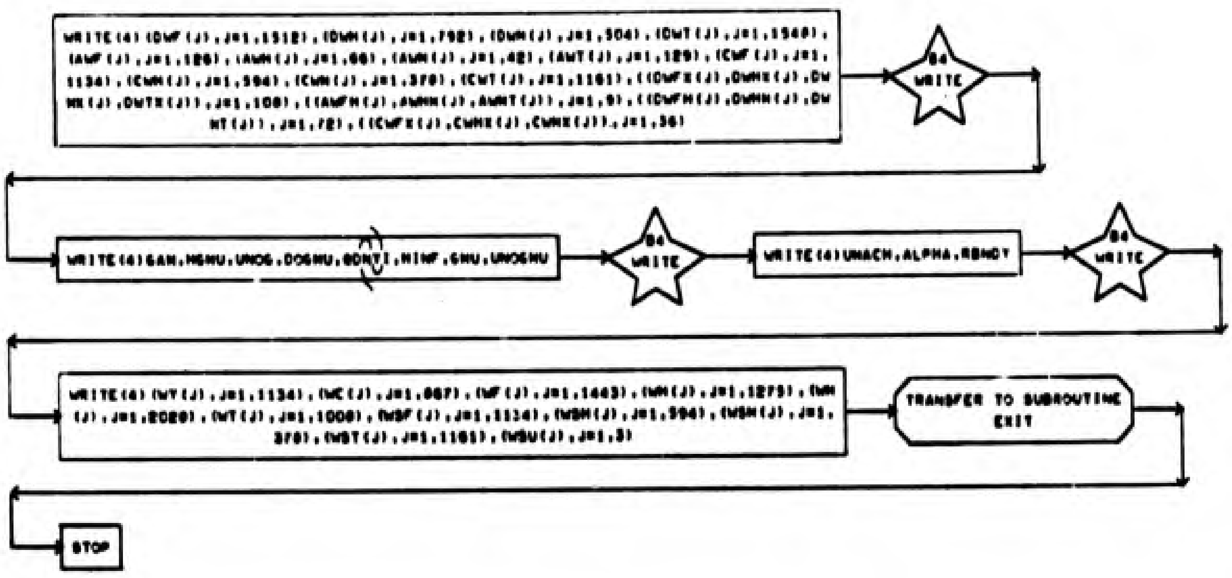
Detailed Flow Charts of Subroutine TEHAI (34 of 37)



Detailed Flow Charts of Subroutine TEHAI (35 of 37)

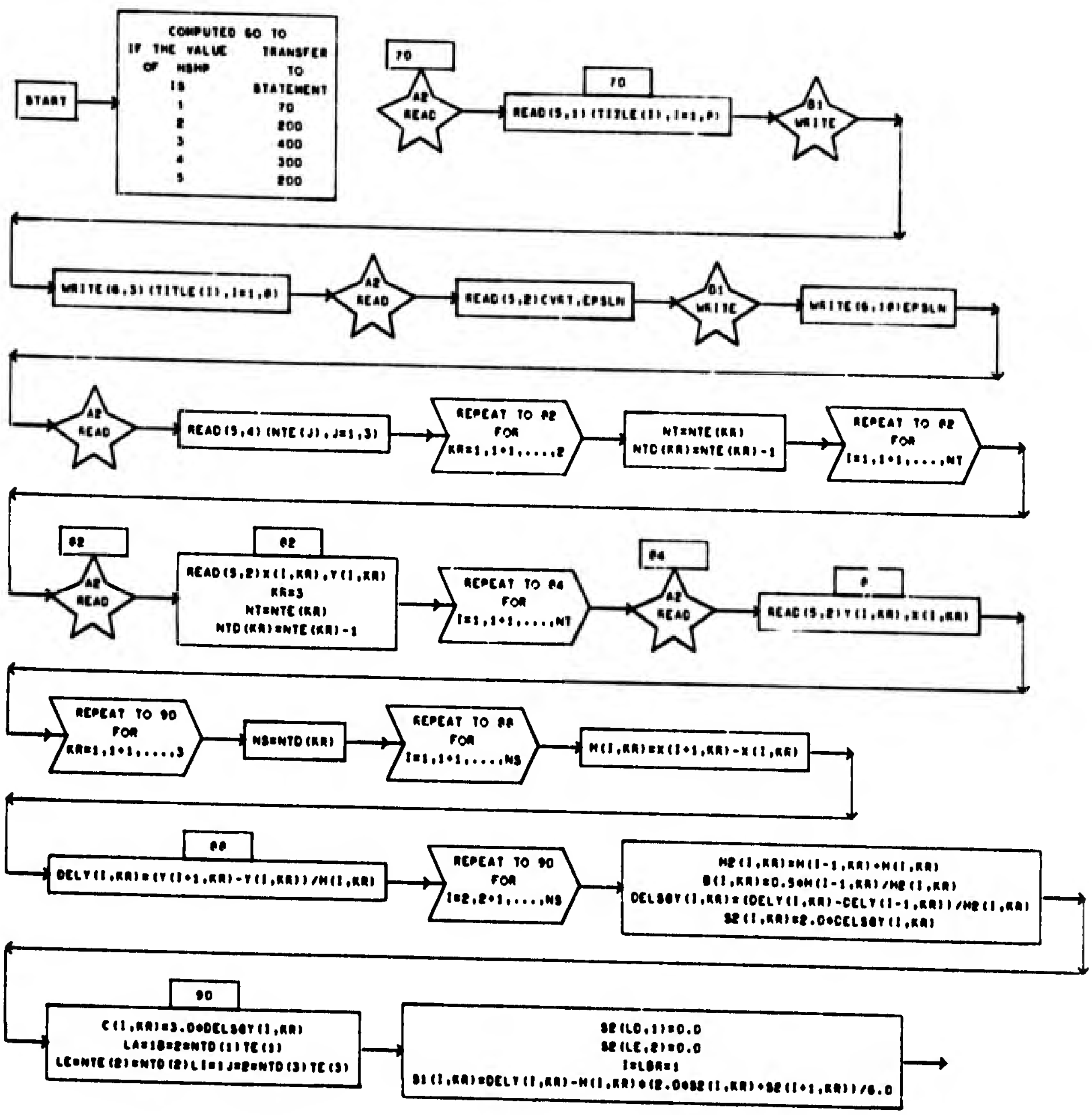


Detailed Flow Charts of Subroutine TEHA1 (36 of 37)

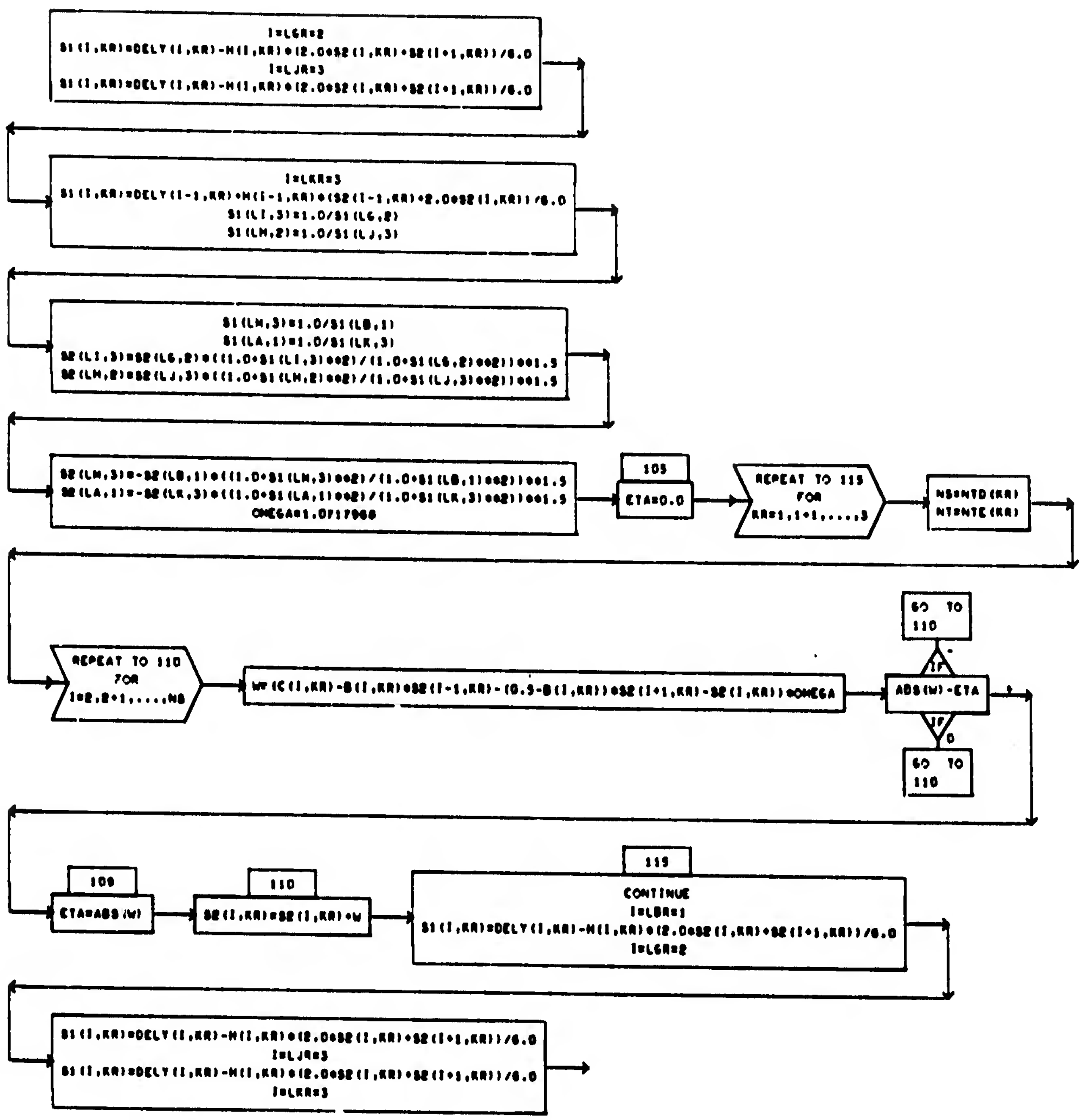


Detailed Flow Charts of Subroutine TEHAI (37 of 37)

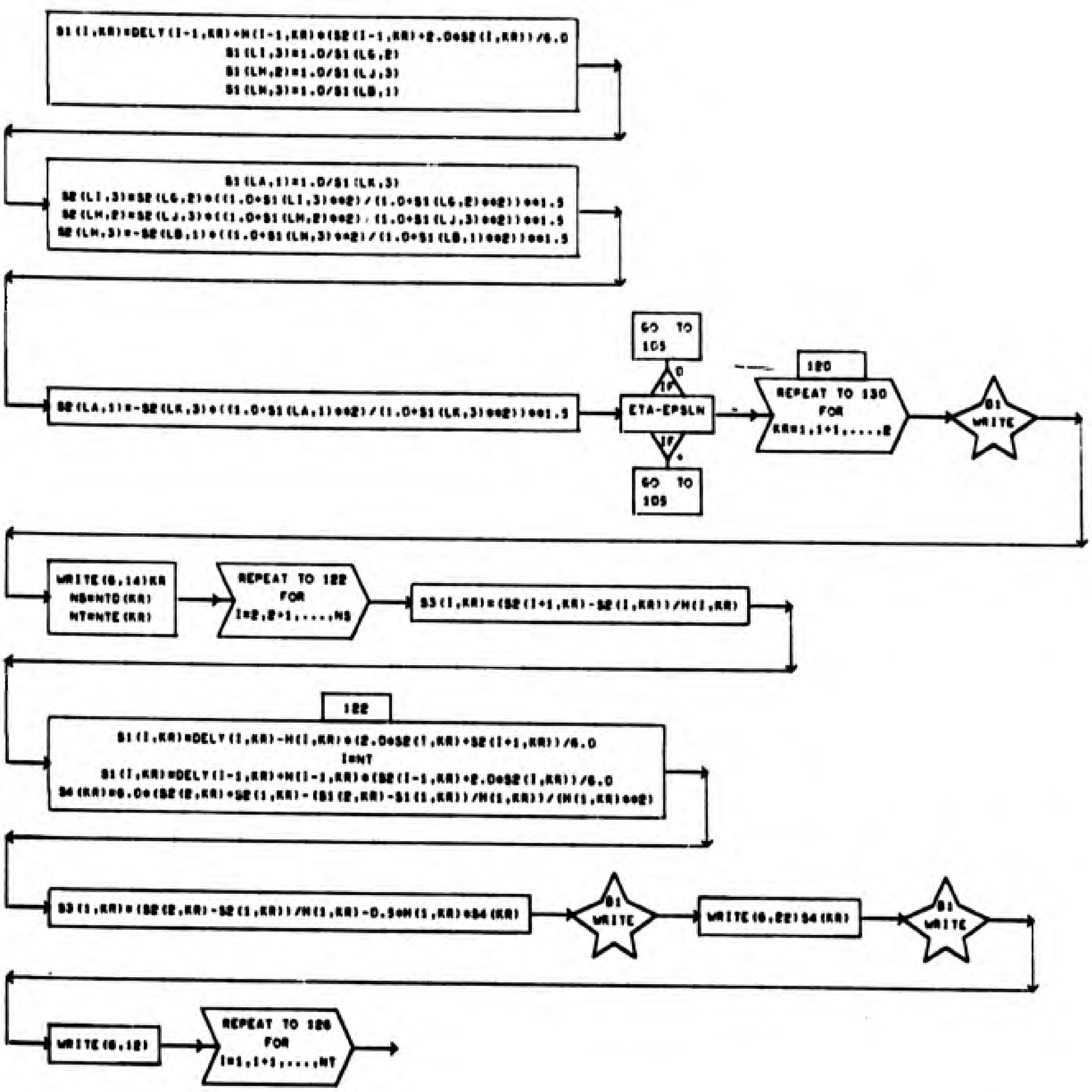
SUBROUTINE SHAPE(NSHP,CLINE)



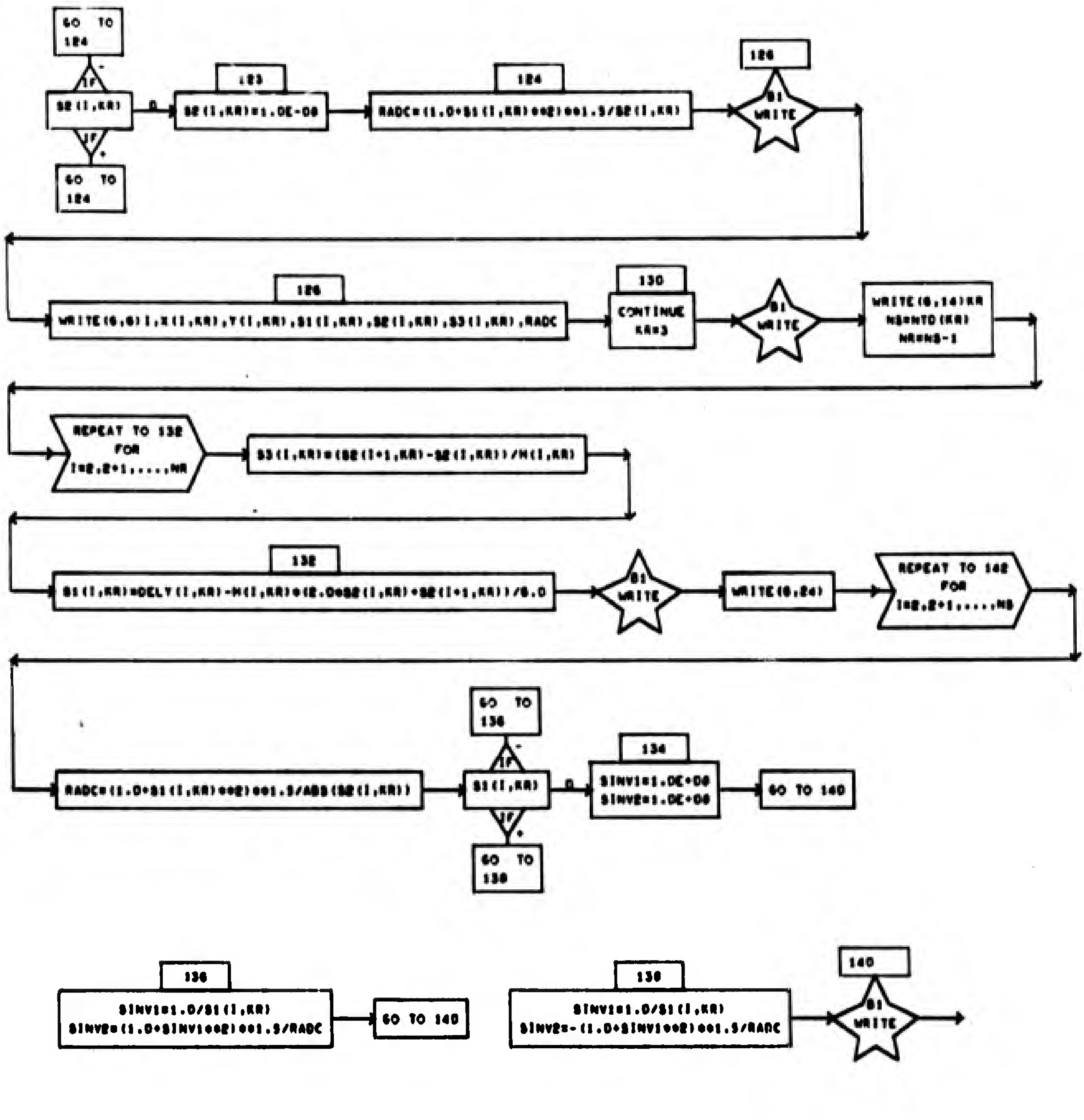
Detailed Flow Charts of Subroutine SHAPE (1 of 9)



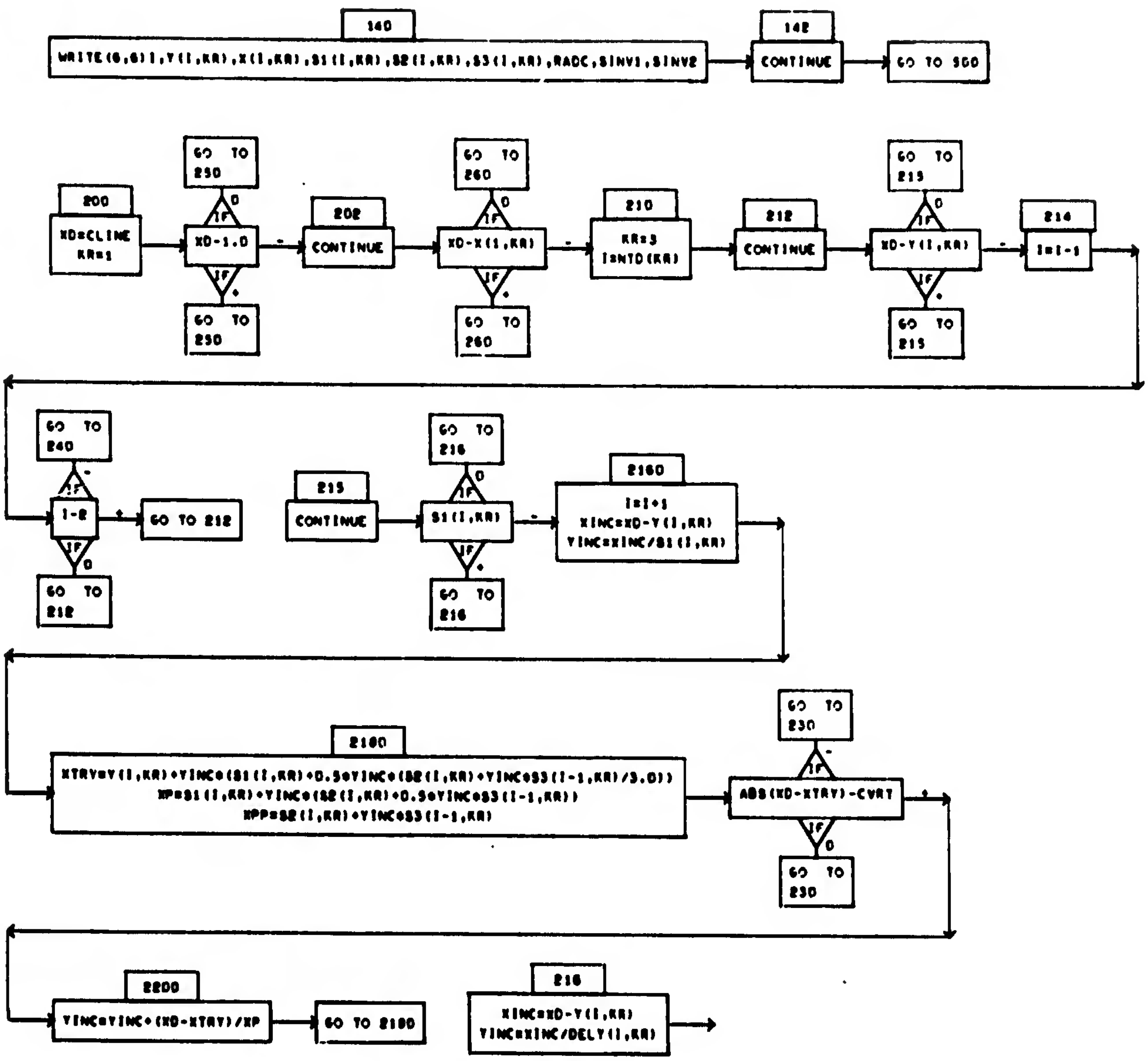
Detailed Flow Charts of Subroutine SHAPE (2 of 9)



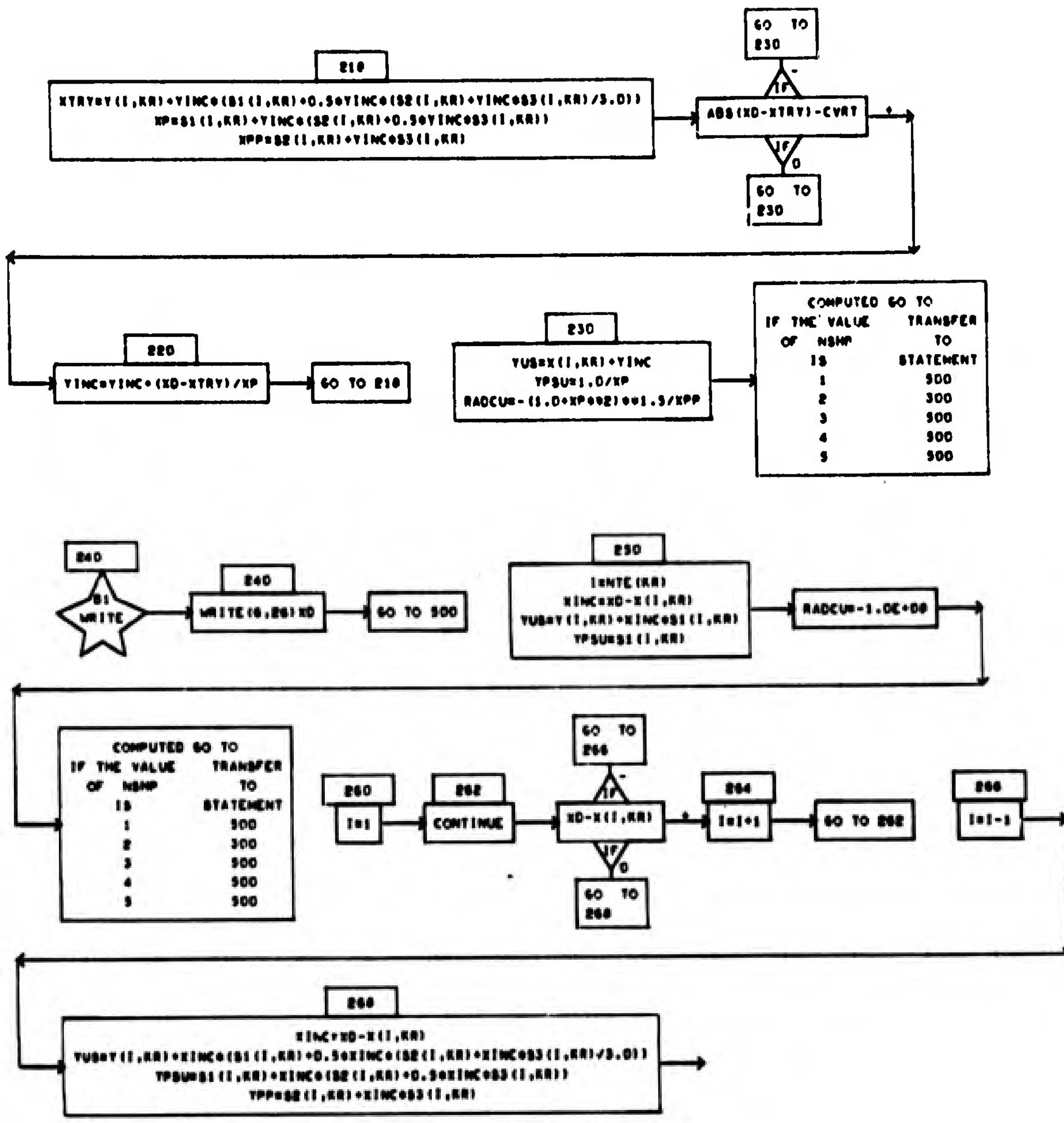
Detailed Flow Charts of Subroutine SHAPE (3 of 9)



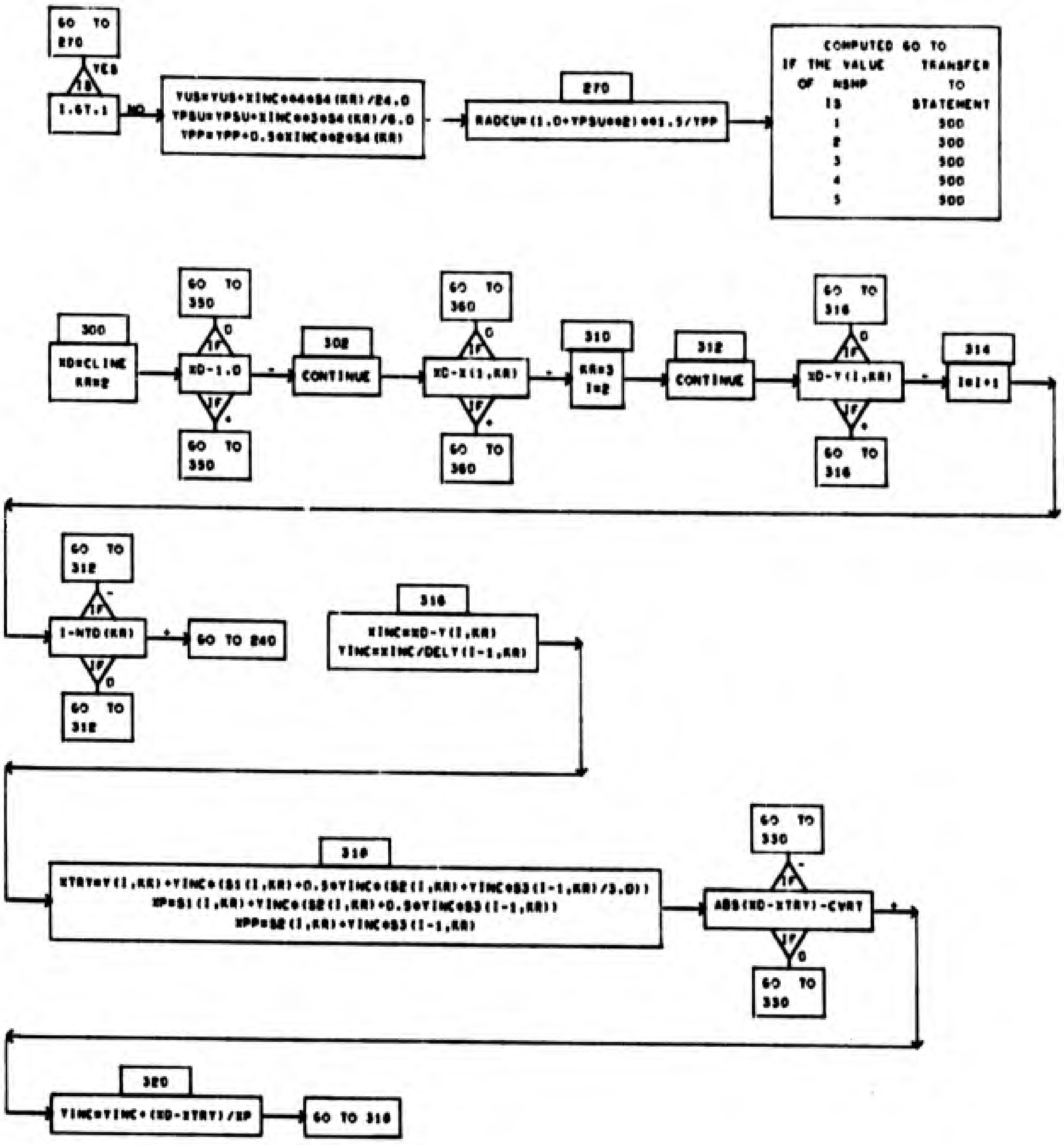
Detailed Flow Charts of Subroutine SHAPE (4 of 9)



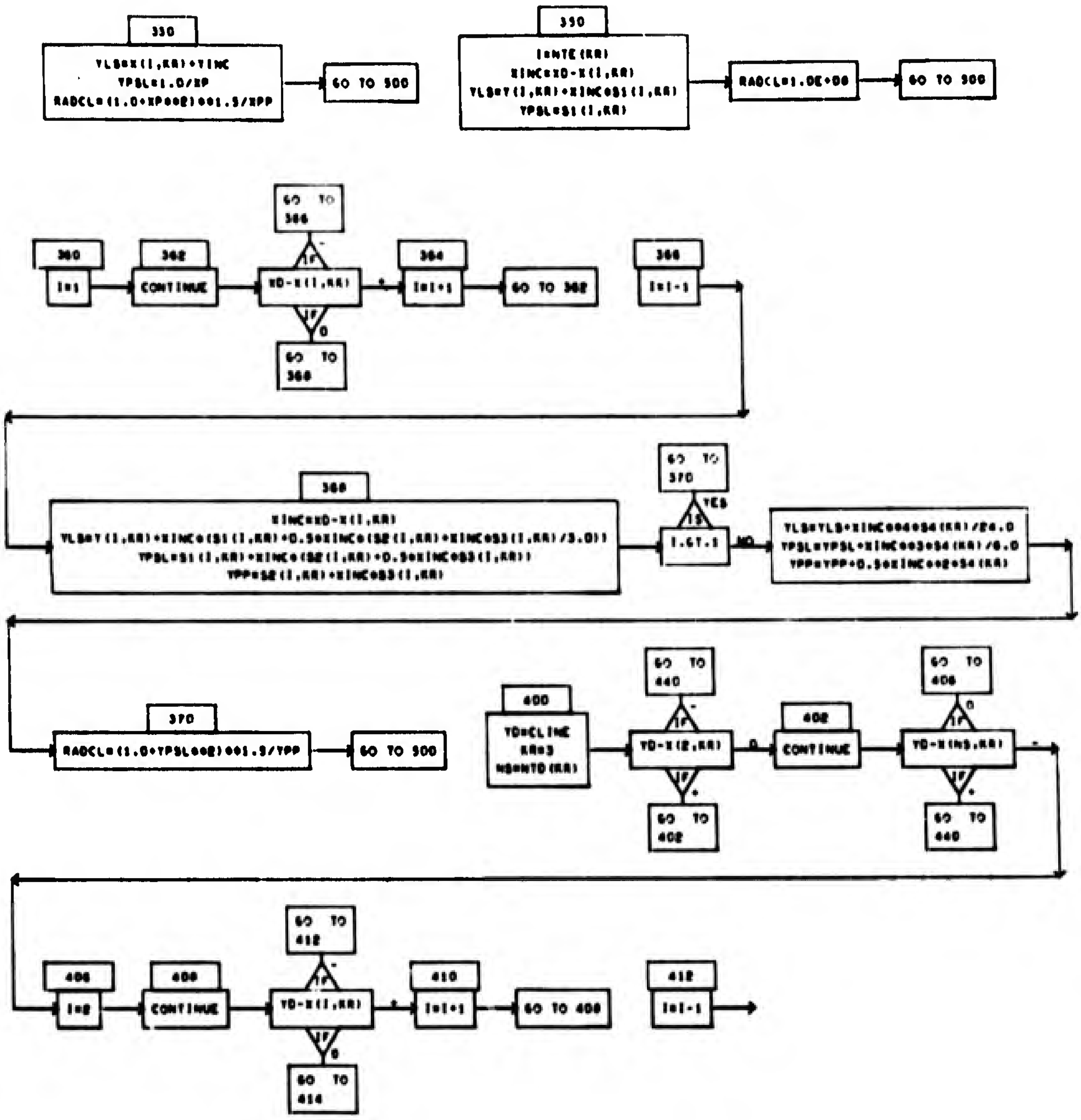
Detailed Flow Charts of Subroutine SHAPE (5 of 9)



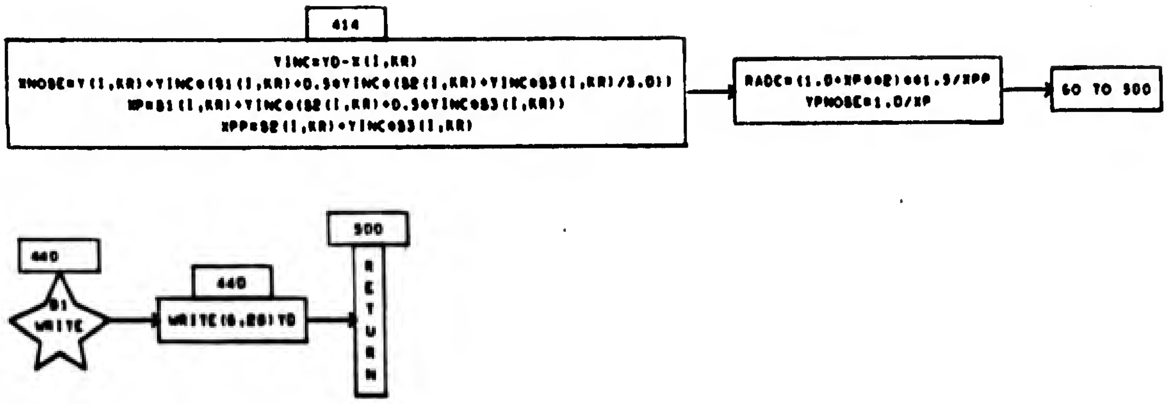
Detailed Flow Charts of Subroutine SHAPE (6 of 9)



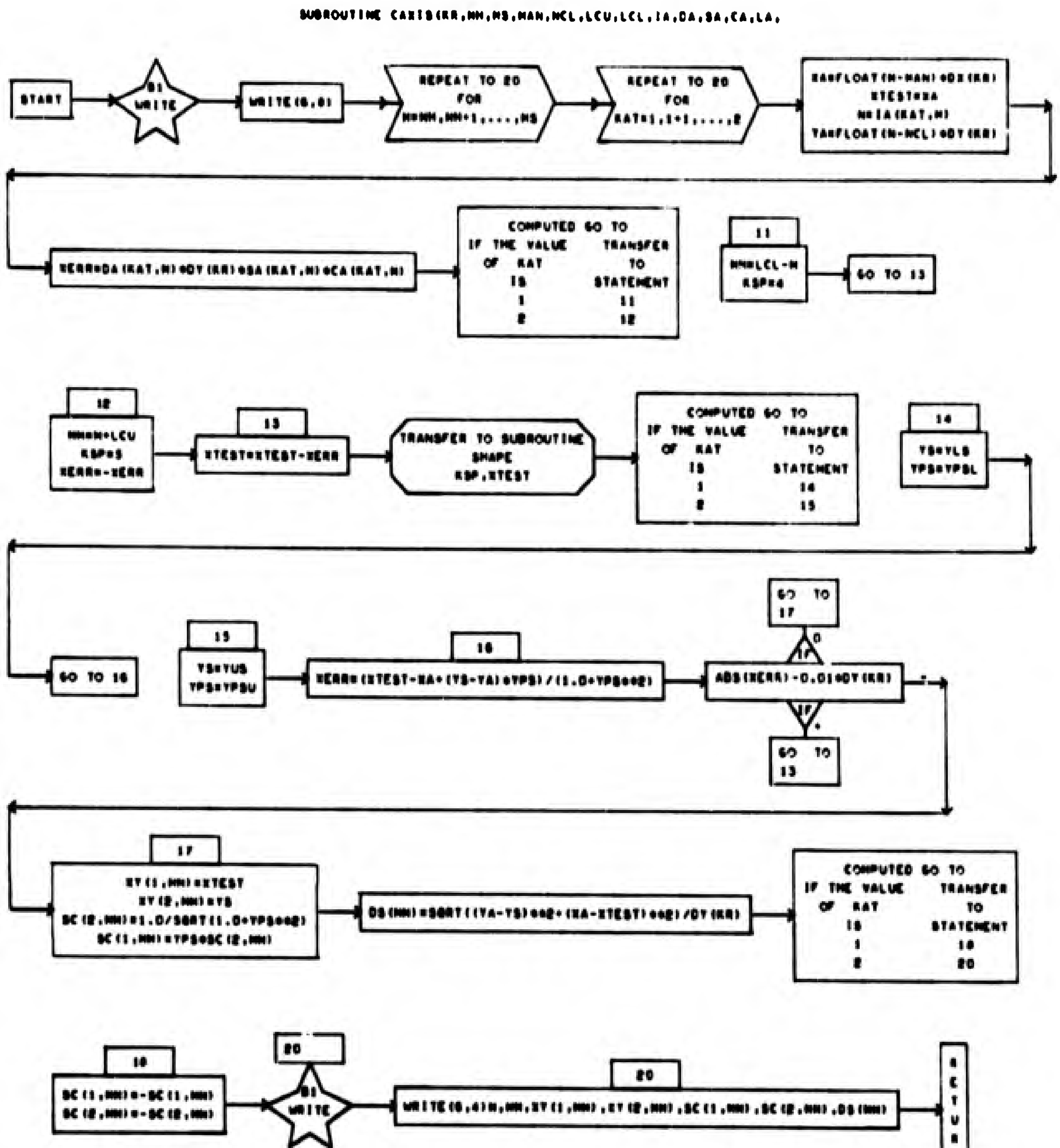
Detailed Flow Charts of Subroutine SHAPE (7 of 9)



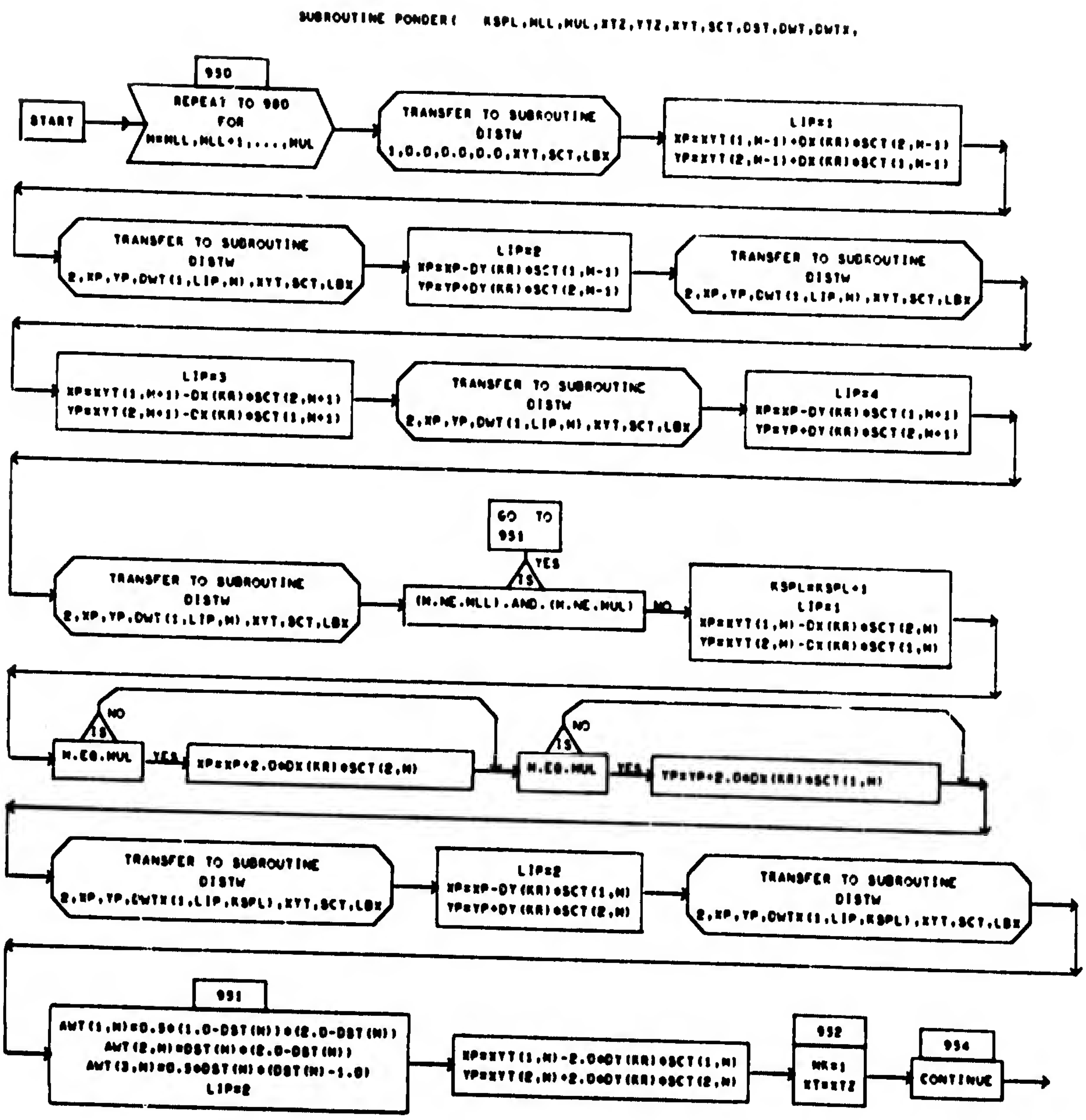
Detailed Flow Charts of Subroutine SHAPE (8 of 9)



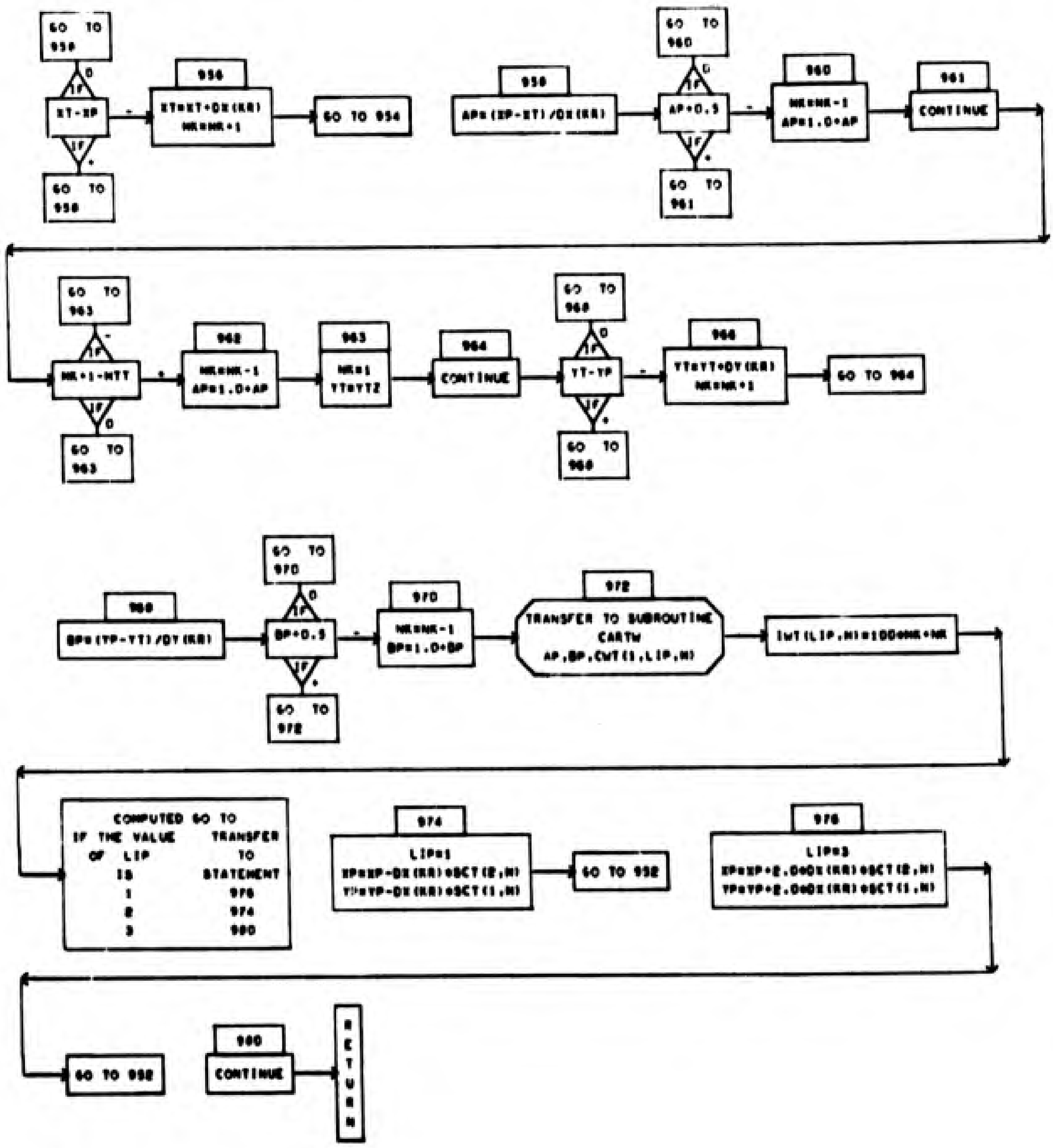
Detailed Flow Charts of Subroutine SHAPE (9 of 9)



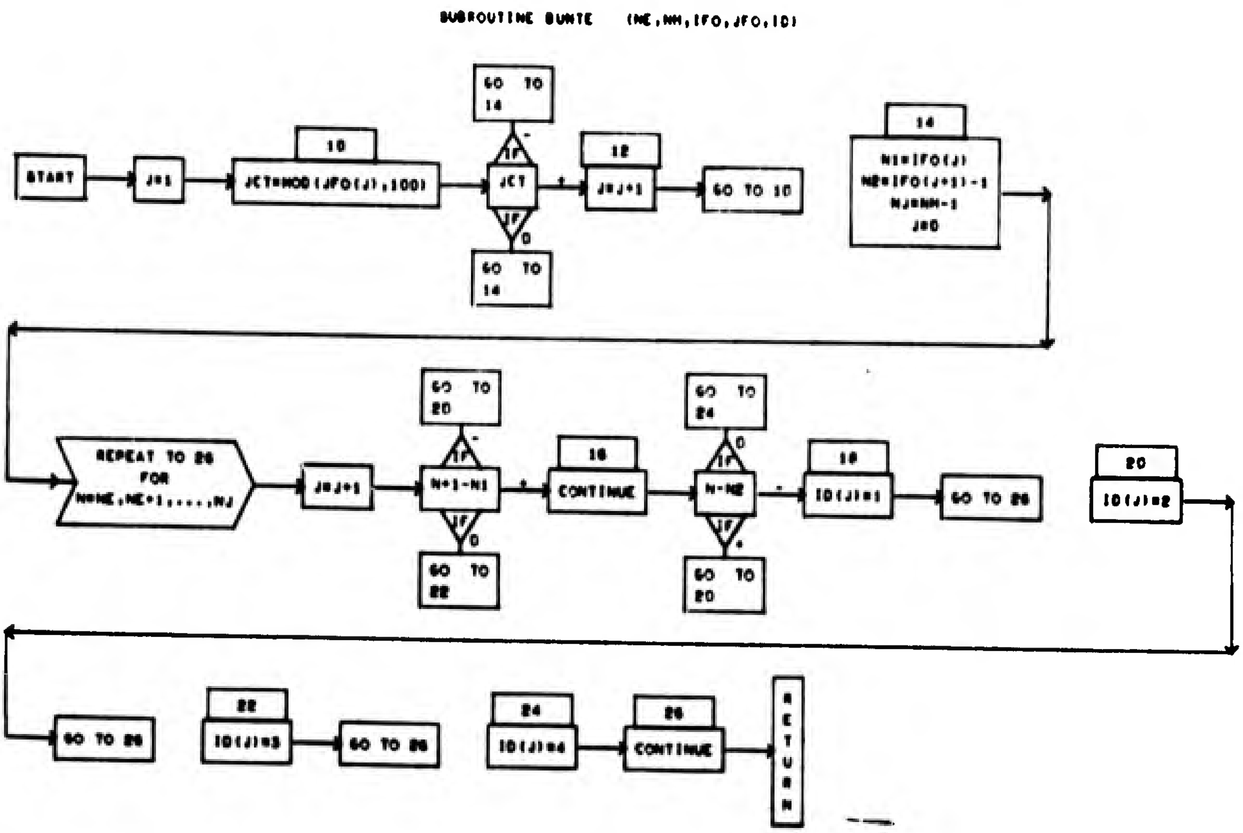
Detailed Flow Chart of Subroutine CAXIS



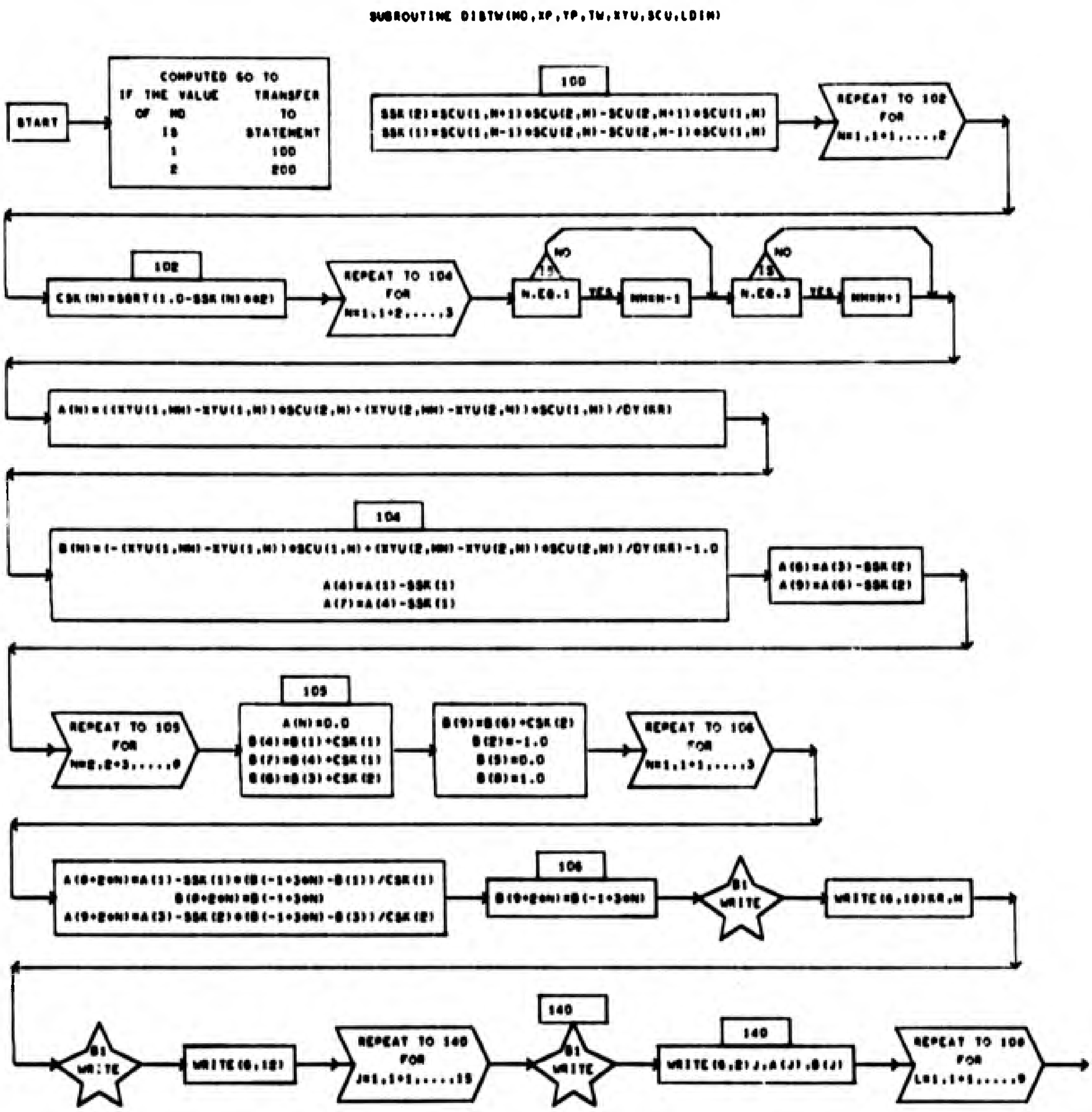
Detailed Flow charts of Subroutine PONDER (1 of 2)



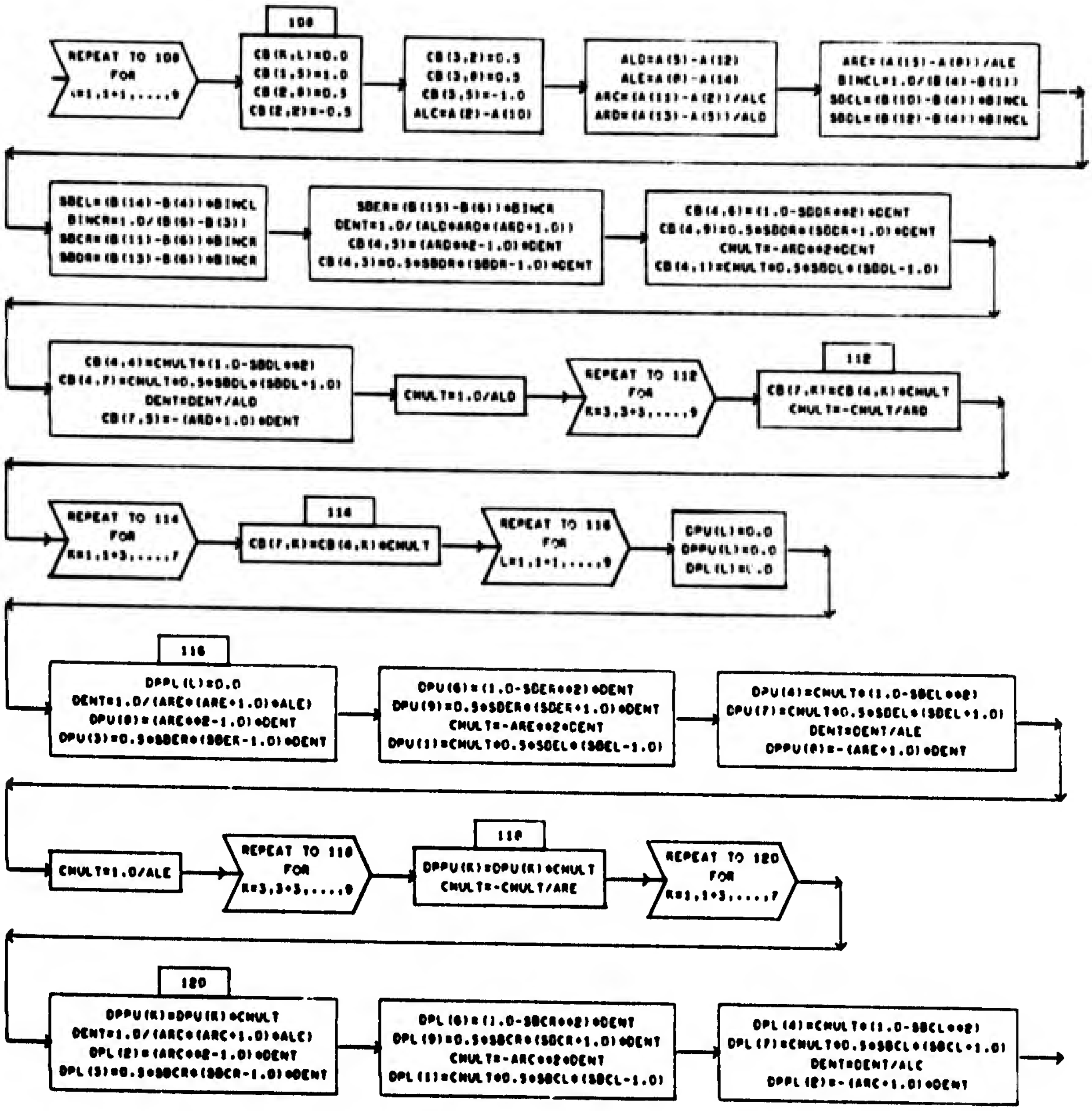
Detailed Flow Charts of Subroutine PONDER (2 of 2)



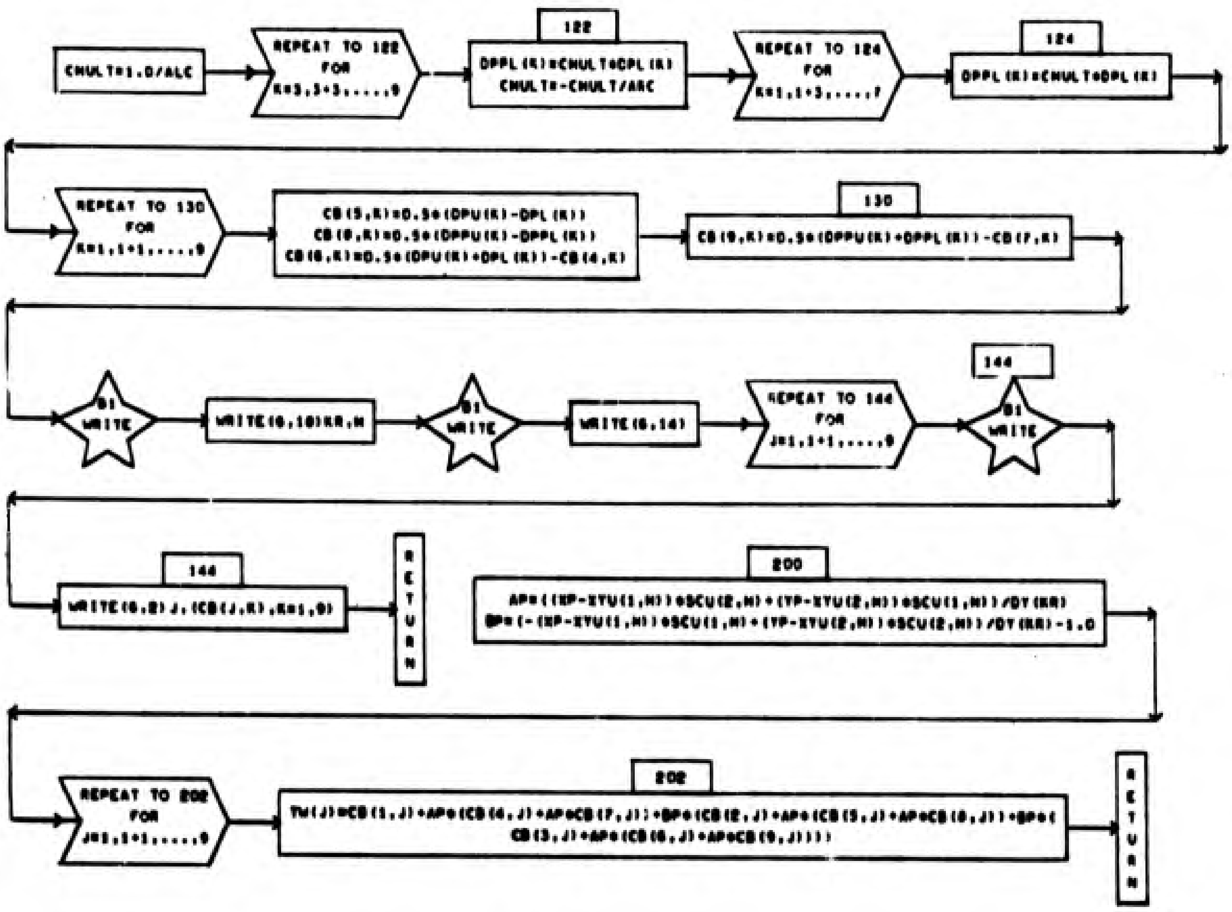
Detailed Flow Chart of Subroutine BUNTE



Detailed Flow charts of Subroutine DISTW (1 of 3)

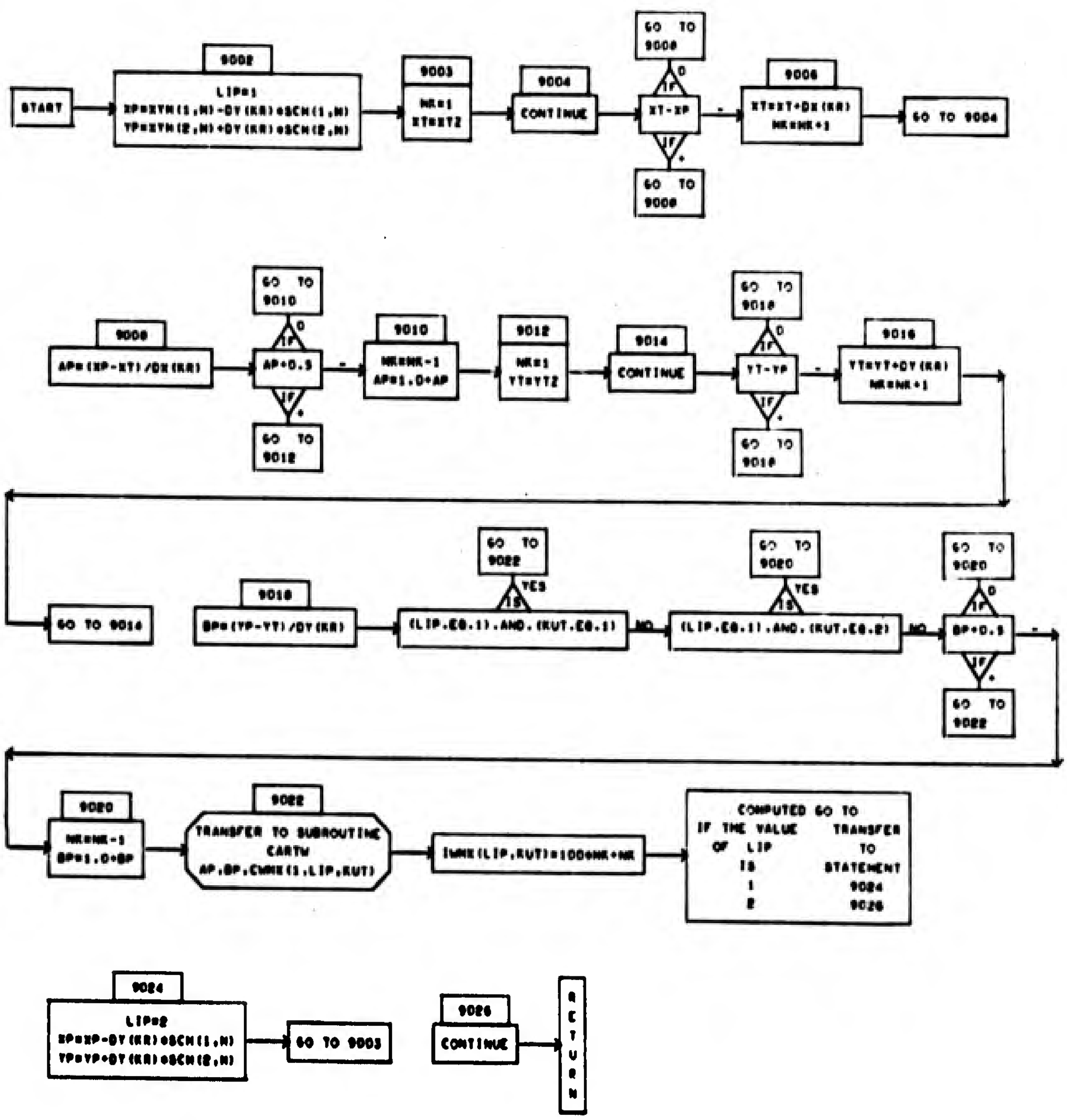


Detailed Flow charts of Subroutine DISTW (2 of 3)

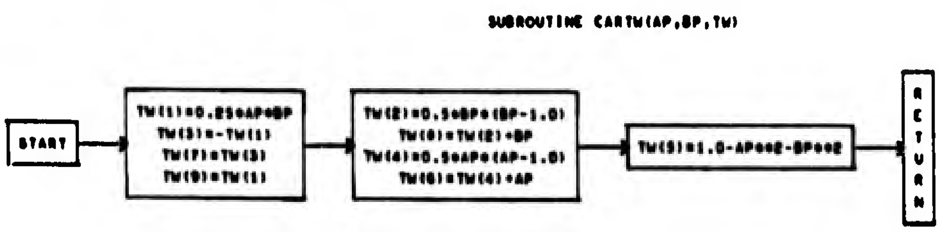


Detailed Flow charts of Subroutine DISTW (3 of 3)

SUBROUTINE WAIF(KR,KUT,N,XT2,YT2,XYN,SCH,CWMX,IWMX,LBX)



Detailed Flow Chart of Subroutine WAIF



Detailed Flow Chart of Subroutine CARTW

APPENDIX IV

FLOW FIELD PROGRAM LISTING

```
REQUEST(TAPE3,HY,ID=14713)
REQUEST(TAPE4,HY,R,SV,ID=RANLBIN)
REWIND(TAPE4)
REWIND(TAPE3)
SET(0)
RUN(S,,,,,,,,,1)
LGO(LC,100000)
UNLOAD(TAPE3)
UNLOAD(TAPE4)
EXIT.
UNLOAD(TAPE3)
UNLOAD(TAPE4)
RETURN(TAPE4)
R
      PROGRAM HANE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE3,TAPE4)
C
C*****BLUNTED LIFTING TRANSONIC AIRFOIL
C
      COMMON/LIMITS/ MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT,
     C MCA, MCB, MCE, MCF, MCG, MCH, MCS, MCT, NCA, NCB, NCE, NCF,
     C NCG, NCH, NCS, NCT,
     F MFA, MFB, MFD, MFE, MFF, MFG, MFH, MFQ, MFS, MFT, MFAN,
     F MFANP, MFAT, MFATM, MFATP, NFA, NFB, NFD, NFE, NFF, NFG, NFH,
     F NFQ, NFS, NFT, NFCL,
     H MHA, MHB, MHD, MHE, MHF, MHG, MHH, MHQ, MHS, MHT, MHAN, MHANP,
     H NHA, NHB, NHD, NHE, NHF, NHG, NHH, NHQ, NHS, NHT, NHCL,
     M MMA, MMB, MMD, MME, MMF, MMG, MMH, MMQ, MMS, MMT, MMAN, MMANP,
     M NMA, NMB, NMD, NME, NMF, NMG, NMH, NMQ, NMS, NMT, NMCL,
     T MTA, MTB, MTD, MTQ, MTS, MTT, MTAN, MTANP, NTA, NTB, NTD,
     T NTQ, NTS, NTT, NTCL,
     S MSL, MST, NSL, NST
      COMMON/BLIMIT/  LBFA, LBFB, LBFC, LBFD, LBFE, LBFI, LBFJ, LBFK,
     F LBFL, LBFM, LBFMAX,
     H LBHA, LBHB, LBHC, LBHD, LBHE, LBHI, LBHJ, LBHK, LBHL, LBHM,
     H LBHMAX,
     M LBMA, LBMB, LBMC, LBMD, LBME, LBMI, LBMJ, LBMK, LBML, LBMM,
     M LBMMAX,
     T LBTA, LBTB, LBTC, LBTD, LBTE, LBTI, LBTJ, LBTK, LBTL, LBTM,
     T LBTQ, LBTR, LBTS, LBTT, LBTMAX
      COMMON/ZMESH/  DX(6), DY(6), RH(9)
      COMMON/ GEONF/ SY(42), CY(42), DYB(2,42), DCB(2,4,17),
     A XYF(2,42), SCF(2,42), XYH(2,22), SCH(2,22),
     B XYM(2,14), SCM(2,14), XYT(2,43), SCT(2,43)
      COMMON/OPLM/   ICO(4,17), JCO(4,17), IFO(8,37), JFO(8,37),
     A IHO(8,25), JHO(8,25), IMO(8,26), JMO(8,26), ITO(8,16), JTO(8,16)
      COMMON/DURA/   L
      COMMON/TERSE/  IYB(2,42), ICB(2,4,17), IDFH(10), IDHM(10),
     A IDMT(10), IWF(3,42), IWH(3,22), IWM(3,14), IWT(3,43),
     B LINTF(2), LINTH(2), LINTM(2), IWFX(2,2), IWHX(2,2), IWMX(2,2),
     C IAF(2,37), IAH(2,25), IAM(2,26), IAT(2,16), IAN(21),
     D IWFH(2,4), IWHM(2,4), IWMT(2,4)
      COMMON/OMOSA/  DWF(9,4,42), DWH(9,4,22), DWM(9,4,14),
     A DWT(9,4,43), AWF(3,42), AWH(3,22), AWM(3,14), AWT(3,43),
     B CWF(9,3,42), CWH(9,3,22), CWM(9,3,14), CWT(9,3,43),
     C DWFX(9,2,6), DWHX(9,2,6), DWMX(9,2,6), DWTX(9,2,6),
     D AWFH(3,3), AWHM(3,3), AWMT(3,3),
     E DWFH(9,2,4), DWHM(9,2,4), DWMT(9,2,4),
```

```
     F CWFX(9,2,2), CWHX(9,2,2), CWMX(9,2,2)
      COMMON/GAS/   GAM, HGMU, UNOG, DOGMU, ODNYI, HINF, GMO, UNOGMU
      COMMON/FLOW/ UMACH, ALPHA, RBNDY
      COMMON/PEND/CJX(3,3), DCX(3,6), DCY(3,6), DHD(3,3), DMBX(5,6),
     A DMBY(5,6), LBIX(6), LBIY(6)
      COMMON/FLAGS/ IBUG, JBCT, JOFB
      COMMON/ENTHP/ JTND, TEND, TENB
      COMMON/TOC/PRMSHY,ATSFY,HGPU,HGPV,HGPW,ATSF(6),TSN(6),PRMSH(6),
     1 PRMSHI(6), PRMYI, CFTT
      COMMON        WY(3,42,9), WC(3,17,17), WF(3,37,13), WH(3,25,17),
     A WM(3,26,26), WT(3,16,21), WSF(3,9,42), WSH(3,9,22), WSM(3,9,14),
     B WST(3,9,43), WSU(3)
C
      DIMENSION DAMP(8), SAFE(8), TITLE(8)
C-----WEIGHTING FACTORS FOR DERIVATIVES
      DATA CJX/-1.5,-0.5,0.5,2.0,0.0,-2.0,-0.5,0.5,1.5/
C
C-----IBUG = 1 NORMAL PRINTOUT
C-----IBUG = 2 CALLS FOR PRINTOUT OF EACH STEP IN THE FINE REGION
C-----INIT = 1 DO NOT SAVE TAPE OF LAST LINE
C-----INIT = 2 SAVE LAST LINE ON TAPE
C-----JTND = 1, NO FORCING OF  FIELD TOWARD STEADY FREE-STREAM TOTAL ENTH
C-----IF JTND = 2 - KICK BOUNDARY POINTS AND ADJACENT REGULAR MESH POINTS
C       BY TENB
C-----JTND = 3, KICK ALL FIELD POINTS ALSO BY TEND FOR ALL L
C-----JTND = 4, KICK ALL FIEL POINTS ALSO BY TEND FOR THOSE L BEING PRINT
C-----JBCT USED TO SPACE CALLING OF ENTHALPY TREND
      READ(5,1)(TITLE(I),I=1,8)
      WRITE(6,3)(TITLE(I),I=1,8)
      READ(5,2) L2,LMAX,LPRINT,JOUT,JTND,IBUG,INIT,IFIZ,JBCT,JOFB
      WRITE(6,6)L2,LMAX,LPRINT,JOUT,JTND,IBUG,INIT,IFIZ,JBCT,JOFB
      READ(5,4)   (SAFE(K),K=1,8)
      READ(5,4)   (DAMP(K),K=1,8)
      DO 7 K = 1,8
    7 WRITE(6,8) K,         SAFE(K),DAMP(K)
      READ (5,4)  TEND, TENB
      WRITE(6,12) TEND, TENB
C-----READ DATA FROM TEHAI FROM TAPE 3
C-----READ STARTING TAPE  DATA IN BINARY
      READ (3)          MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT,
     C MCA, MCB, MCE, MCF, MCG, MCH, MCS, MCT, NCA, NCB, NCE, NCF,
     C NCG, NCH, NCS, NCT,
     F MFA, MFB, MFD, MFE, MFF, MFG, MFH, MFQ, MFS, MFT, MFAN,
     F MFANP, MFAT, MFATM, MFATP, NFA, NFB, NFD, NFE, NFF, NFG, NFH,
     F NFQ, NFS, NFT, NFCL,
     H MHA, MHB, MHD, MHE, MHF, MHG, MHH, MHQ, MHS, MHT, MHAN, MHANP,
     H NHA, NHB, NHD, NHE, NHF, NHG, NHH, NHQ, NHS, NHT, NHCL,
     M MMA, MMB, MMD, MME, MMF, MMG, MMH, MMQ, MMS, MMT, MMAN, MMANP,
     M NMA, NMB, NMD, NME, NMF, NMG, NMH, NMQ, NMS, NMT, NMCL,
     T MTA, MTB, MTD, MTQ, MTS, MTT, MTAN, MTANP, NTA, NTB, NTD,
     T NTQ, NTS, NTT, NTCL,
     S MSL, MST, NSL, NST
       READ (3)          LBFA, LBFB, LBFC, LBFD, LBFE, LBFI, LBFJ, LBFK,
     F LBFL, LBFM, LBFMAX,
     H LBHA, LBHB, LBHC, LBHD, LBHE, LBHI, LBHJ, LBHK, LBHL, LBHM,
     H LBHMAX,
     M LBMA, LBMB, LBMC, LBMD, LBME, LBMI, LBMJ, LBMK, LBML, LBMM,
```

```
     M LBMMAX,
     T LBTA, LBTB, LBTC, LBTD, LBTE, LBTI, LBTJ, LBTK, LBTL, LBTM,
     T LBTQ, LBTR, LBTS, LBTT, LBTMAX
      READ (3)        L
      READ (3)     (IYB(J), J = 1,84), (ICB(J), J=1,136),
     A ((IDFH(J), IDHM(J), IDMT(J)), J=1,10), (IWF(J), J=1,126),
     B (IWH(J), J=1,66), (IWM(J), J=1,42), (IWT(J), J=1,129),
     C ((LINTF(J), LINTH(J), LINTM(J)), J=1,2), ((IWFX(J), IWHX(J),
     D IWMX(J)), J=1,4), (IAF(J), J=1,74), (IAH(J), J=1,50), (IAM(J),
     E J=1,52), (IAT(J), J=1,32), (IAN(J), J=1,21),
     F ((IWFH(J), IWHM(J), IWMT(J)), J=1,8)
      READ (3)          ((ICO(J), JCO(J)), J = 1,68), ((IFO(J), JFO(J)),
     F J=1,296), ((IHO(J),JHO(J)),J=1,200), ((IMO(J),JMO(J)),J=1,208),
     G ((ITO(J),JTO(J)), J=1,128)
      READ (3)          ((DX(J),DY(J)),J=1,6), (RH(J), J=1,9)
      READ (3)          ((CY(J),SY(J)), J=1,42), (DYB(J), J=1,84),
     A (DCB(J), J=1,136), ((XYF(J),SCF(J)), J=1,84),
     B ((XYH(J),SCH(J)), J=1,44), ((XYM(J),SCM(J)), J=1,28),
     C ((XYT(J),SCT(J)), J=1,86)
      READ (3)          (DWF(J), J=1,1512), (DWH(J), J=1,792),
     A (DWM(J), J=1,504), (DWT(J), J=1,1548), (AWF(J), J=1,126),
     B (AWH(J), J=1,66), (AWM(J), J=1,42), (AWT(J), J=1,129),
     C (CWF(J), J=1,1134), (CWH(J), J=1,594), (CWM(J), J=1,378),
     D(CWT(J),J=1,1161), ((DWFX(J),DWHX(J),DWMX(J),DWTX(J)),J=1,108),
     E ((AWFH(J), AWHM(J), AWMT(J)), J=1,9),
     F ((DWFH(J), DWHM(J), DWMT(J)), J=1,72),
     G ((CWFX(J), CWHX(J), CWMX(J)), J=1,36)
      READ (3)          GAM, HGMU, UNOG, DOGMU, QDNYI, HINF, GMU, UNOGMU
      READ (3)     UMACH, ALPHA, RBNDY
      READ (3)          (WY(J), J=1,1134), (WC(J), J=1,867),
     A (WF(J), J=1,1443), (WH(J), J=1,1275), (WM(J), J=1,2028),
     B (WT(J), J=1,1008), (WSF(J), J=1,1134), (WSH(J), J=1,594),
     C (WSM(J), J=1,378), (WST(J), J=1,1161), (WSU(J), J=1,3)
      WRITE(6,10) UMACH, GAM, ALPHA
C-----USEFUL CONSTANTS
      HGPU    = 0.5*(GAM + 1.0)
      HGPV    = 0.5*(HGPU - 1.0)
      HGPW    = (HGPU - 2.0)/3.0
C-----SPECIAL REPAIRS
      IF(IFIZ.EQ.2) CALL REPAIR
      WRITE(6,10) UMACH, GAM, ALPHA
C*****TIME STEP CONSTANTS
C-----CARTESIAN REGIONS
      DO 204 K = 1,6
      PRMSHI(K) = DX(K)/DY(K)
      PRMSH(K) = DY(K)/DX(K)
      ATSF(K) = SQRT(1.0 + PRMSH(K)**2)
  204 TSN(K) = SAFE(K)*0.5
C-----OUTER REGION
      DRHO = DX(1)
      DTHET = DY(1)
      PRMSHY  = DRHO / DTHET
      ATSFY   = SQRT(1.0 + PRMSHY**2)
      TSN(1) = 0.5*SAFE(1)*DRHO*RBNDY/DTHET
      PRMYI   = 1.0/PRMSHY
C-----WEIGHTING FACTORS FOR DIFFUSIVE DAMPING
      CDP = DAMP(6)
```

```
      HCDP = 0.5*CDP
      DHD(1,1) = 0.5*(1.0 - HCDP)
      DHD(1,2) = 0.5*CDP
      DHD(1,3) = -0.5*HCDP
      DHD(2,1) = 0.5*HCDP
      DHD(2,2) = 0.5*(1.0 - CDP)
      DHD(2,3) = 0.5*HCDP
      DHD(3,1) = -0.5*HCDP
      DHD(3,2) = 0.5*CDP
      DHD(3,3) = 0.5*(1.0 - HCDP)
      DO 210 KR = 1,6
      DCX(1,KR) = 0.5*(1.0 - DAMP(KR))
      DCX(2,KR) = 0.25*DAMP(KR)
      DCY(1,KR) = 0.5*(1.0 - PRMSHI(KR)**2*DAMP(KR))
  210 DCY(2,KR) = DCX(2,KR)*PRMSHI(KR)**2
      DO 220 KR = 3,6
      BMRTX = DX(KR-1)/DX(KR)
      BMRTY = DY(KR-1)/DY(KR)
      LBIX(KR) = IFIX(BMRTX + 0.0001)
      LBIY(KR) = IFIX(BMRTY + 0.0001)
      RDRX = DCX(2,KR-1)/DCX(2,KR) - 1.0
      RDRY = DCY(2,KR-1)/DCY(2,KR) - 1.0
      DMBX(1,KR) = 1.0
      DMBY(1,KR) = 1.0
      LL = LBIX(KR)
      IF(LL.LT.2) GO TO 215
      DO 214 JB = 2,LL
      EN = JB
      DMBX(JB,KR) = EN**2*(1.0 + EN*RDRX/BMRTX)
      DUCX = 0.5 - DMBX(JB,KR)*(0.5 - DCX(1,KR))
      IF(DUCX.GE.0.0) GO TO 214
      DMBX(JB,KR) = 0.5/(0.5 - DCX(1,KR))
  214 CONTINUE
  215 LL = LBIY(KR)
      IF(LL.LT.2) GO TO 220
      DO 216 JB = 2,LL
      EN = JB
      DMBY(JB,KR) = EN**2*(1.0 + EN*RDRY/BMRTY)
      DUCY = 0.5 - DMBY(JB,KR)*(0.5 - DCY(1,KR))
      IF(DUCY.GE.0.0) GO TO 216
      DMBY(JB,KR) = 0.5/(0.5 - DCY(1,KR))
  216 CONTINUE
  220 CONTINUE
C-----CONFIRMATION OF DAMPING CONSTANTS
      WRITE(6,14)
      DO 224 J = 1,3
  224 WRITE(6,4)  (DHD(J,I), I = 1,3)
      WRITE(6,16)
      DO 226 KR = 1,6
  226 WRITE(6,18) KR, (DCX(J,KR),J=1,2), (DCY(J,KR),J=1,2)
      DO 230 KR = 3,6
      WRITE(6,20)KR
      WRITE(6,22) LBIX(KR), LBIY(KR)
      WRITE(6,24)
      DO 230 JB = 1,5
  230 WRITE(6,4) DMBX(JB,KR), DMBY(JB,KR)
C*****MAIN ROUTINE
```

```
  130 LZ        = LZ + 1
C-----MAIN LOOP
      DO 1310 L=LZ,LMAX
      GO TO (140,150,160),JOFB
  140 CALL YTRE
  150 CALL OOKII
  160 CALL SUSUMU(3,MFA,MFT,NFA,NFT,WF,IFO,JFO,1)
      CALL HERI(3,LBFMAX,WSF,1)
      CALL SUSUMU(4,MHA,MHT,NHA,NHT,WH,IHO,JHO,1)
      CALL HERI(4,LBHMAX,WSH,1)
      CALL SUSUMU(5,MMA,MMT,NMA,NMT,WM,IMO,JMO,1)
      CALL HERI(5,LBMMAX,WSM,1)
      CALL JIKAN
      CALL SUSUMU(6,MTA,MTT,NTA,NTT,WT,ITO,JTO,2)
      CALL HERI(6,LBTMAX,WST,2)
      CALL SAKAI
C-----OUTPUT AND TERMINATION
      IF(JIND.EQ.3) CALL TREND
      IF((JIND.EQ.4).AND.(MOD(L,JBCT).EQ.0)) CALL TREND
      IF(MOD(L,LPRINT).NE.0) GO TO 1310
      CALL OUCH(JOUT)
 1310 CONTINUE
C-----BOTTOM OF TREATMENT OF ONE TIME PLANE
      IF((MOD(LMAX,LPRINT).NE.0).OR.(JOUT.NE.2)) CALL OUCH(2)
C
      IF(INIT.EQ.1) GO TO 1500
C-----WRITE RESULTS ON TAPE FOR FUTURE USE
C-----WRITE NEW STARTING TAPE DATA IN BINARY
      WRITE(4)        MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT,
     C MCA, MCB, MCE, MCF, MCG, MCH, MCS, MCT, NCA, NCB, NCE, NCF,
     C NCG, NCH, NCS, NCT,
     F MFA, MFB, MFD, MFE, MFF, MFG, MFH, MFQ, MFS, MFT, MFAN,
     F MFANP, MFAT, MFATM, MFATP, NFA, NFB, NFD, NFE, NFF, NFG, NFH,
     F NFQ, NFS, NFT, NFCL,
     H MHA, MHB, MHD, MHE, MHF, MHG, MHH, MHQ, MHS, MHT, MHAN, MHANP,
     H NHA, NHB, NHD, NHE, NHF, NHG, NHH, NHQ, NHS, NHT, NHCL,
     M MMA, MMB, MMD, MME, MMF, MMG, MMH, MMQ, MMS, MMT, MMAN, MMANP,
     M NMA, NMB, NMD, NME, NMF, NMG, NMH, NMQ, NMS, NMT, NMCL,
     T MTA, MTB, MTD, MTQ, MTS, MTT, MTAN, MTANP, NTA, NTB, NTD,
     T NTG, NTS, NTT, NTCL,
     S MSL, MST, NSL, NST
      WRITE(4)          LBFA, LBFB, LBFC, LBFD, LBFE, LBFI, LBFJ, LBFK,
     F LBFL, LBFM, LBFMAX,
     H LBHA, LBHB, LBHC, LBHD, LBHE, LBHI, LBHJ, LBHK, LBHL, LBHM,
     H LBHMAX,
     M LBMA, LBMB, LBMC, LBMD, LBME, LBMI, LBMJ, LBMK, LBML, LBMM,
     M LBMMAX,
     T LBTA, LBTB, LBTC, LBTD, LBTE, LBTI, LBTJ, LBTK, LBTL, LBTM,
     T LBTQ, LBTR, LBTS, LBTT, LBTMAX
      WRITE(4)       L
      WRITE(4)       (IYB(J), J = 1,84), (ICB(J), J=1,136),
     A ((IDFH(J), IDHM(J), IDMT(J)), J=1,10), (IWF(J), J=1,126),
     B (IWH(J), J=1,66), (IWM(J), J=1,42), (IWT(J), J=1,129),
     C ((LINTF(J), LINTH(J), LINTM(J)), J=1,2), ((IWFX(J), IWHX(J),
     D IWMX(J)), J=1,4), (IAF(J), J=1,74), (IAH(J), J=1,50), (IAM(J),
     E J=1,52), (IAT(J), J=1,32), (IAN(J), J=1,21),
     F ((IWFH(J), IWHM(J), IWMT(J)), J=1,8)
```

```
      WRITE(4)         ((ICO(J), JCO(J)), J = 1,68), ((IFO(J), JFO(J)),
     F J=1,296), ((IHO(J),JHO(J)),J=1,200), ((IMO(J),JMO(J)),J=1,208),
     G ((ITO(J),JTO(J)), J=1,128)
      WRITE(4)         ((DX(J),DY(J)),J=1,6), (FR(J), J=1,9)
      WRITE(4)         ((CY(J),SY(J)), J=1,42), (DYB(J), J=1,84),
     A (DCB(J), J=1,136), ((XYF(J),SCF(J)), J=1,84),
     B ((XYH(J),SCH(J)), J=1,44), ((XYM(J),SCM(J)), J=1,28),
     C ((XYT(J),SCT(J)), J=1,86)
      WRITE(4)         (DWF(J), J=1,1512), (DWH(J), J=1,792),
     A (DWM(J), J=1,504), (DWT(J), J=1,1548), (AWF(J), J=1,126),
     B (AWH(J), J=1,66), (AWM(J), J=1,42), (AWT(J), J=1,129),
     C (CWF(J), J=1,1134), (CWH(J), J=1,594), (CWM(J), J=1,378),
     D(CWT(J),J=1,1161), ((DWFX(J),DWHX(J),DWMX(J),DWTX(J)),J=1,108),
     E ((AWFH(J), AWHM(J), AWMT(J)), J=1,9),
     F ((DWFH(J), DWHM(J), DWMT(J)), J=1,72),
     G ((CWFX(J), CWHX(J), CWMX(J)), J=1,36)
      WRITE(4)        GAM, HGMU, UNOG, DOGMU, ORNYI, HINF, GMU, UNOGMU
      WRITE(4)      UMACH, ALPHA, RBNDY
      WRITE(4)         (WY(J), J=1,1134), (WC(J), J=1,867),
     A (WF(J), J=1,1443), (WH(J), J=1,1275), (WM(J), J=1,2028),
     B (WT(J), J=1,1008), (WSF(J), J=1,1134), (WSH(J), J=1,594),
     C (WSM(J), J=1,378), (WST(J), J=1,1161), (WSU(J), J=1,3)
 1500 CALL EXIT
      STOP
C
    1 FORMAT(8A10)
    2 FORMAT(16I5)
    3 FORMAT(23X,8A10)
    4 FORMAT(8E10.3)
    6 FORMAT(//6X,2HLZ,6X,4HLMAX,4X,6HLPRINT,6X,4HJOUT,6X,4HJTND,
     1 6X,4HIPUG,6X,4HINIT,6X,4HIFI2,6X*JHCT*6X*JOFB*/I6I10)
    8 FORMAT(//9X,1HK,            16X,4HSAFE,16X,4HDAMP,/I10,      2E20.5)
   10 FORMAT(//7X,5HUMACH,15X,3HGAM,15X,5HALPHA,/3E20.5)
   12 FORMAT(/7X,14HENTHALPY PUSH=,F7.4)
   14 FORMAT(/4X,40HDAMPING FACTORS FOR ODD CELLS DHD(FP,MP))
   16 FORMAT(/4X,33HDIFFUSIVE DAMPING FACTORS,DCX,DCY)
   18 FORMAT(/4X,I5,10X,2E10.3,10X,2E10.3)
   20 FORMAT(//4X,18HDAMPING AT REGION ,I5,11H BOUNDARIES)
   22 FORMAT(/4X,16HCOLUMN OVERLAP =,I5,10X,13HROW OVERLAP =,I5)
   24 FORMAT(/4X,32HDAMPING MULTIPLIERS  DMBX,  DMBY)
   80 FORMAT(16I5)
   82 FORMAT(4E20.13)
      END
```

```
      SUBROUTINE OUCH(JO)
C
      COMMON/LIMITS/ MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT,
     C MCA, MCB, MCE, MCF, MCG, MCH, MCS, MCT, NCA, NCB, NCE, NCF,
     C NCG, NCH, NCS, NCT,
     F MFA, MFB, MFD, MFE, MFF, MFG, MFH, MFQ, MFS, MFT, MFAN,
     F MFANP, MFAT, MFATM, MFATP, NFA, NFB, NFD, NFE, NFF, NFG, NFH,
     F NFQ, NFS, NFT, NFCL,
     H MHA, MHB, MHD, MHE, MHF, MHG, MHH, MHQ, MHS, MHT, MHAN, MHANP,
     H NHA, NHB, NHD, NHE, NHF, NHG, NHH, NHQ, NHS, NHT, NHCL,
     M MMA, MMB, MMD, MME, MMF, MMG, MMH, MMQ, MMS, MMT, MMAN, MMANP,
     M NMA, NMB, NMD, NME, NMF, NMG, NMH, NMQ, NMS, NMT, NMCL,
     T MTA, MTB, MTD, MTQ, MTS, MTT, MTAN, MTANP, NTA, NTB, NTD,
     T NTQ, NTS, NTT, NTCL,
     S MSL, MST, NSL, NST
      COMMON/PLIMIT/  LBFA, LBFB, LBFC, LBFD, LBFE, LBFI, LBFJ, LBFK,
     F LBFL, LBFM, LBFMAX,
     H LBHA, LBHB, LBHC, LBHD, LBHE, LBHI, LBHJ, LBHK, LBHL, LBHM,
     H LBHMAX,
     M LBMA, LBMB, LBMC, LBMD, LBME, LBMI, LBMJ, LBMK, LBML, LBMM,
     M LBMMAX,
     T LBTA, LBTB, LBTC, LBTD, LBTE, LBTI, LBTJ, LBTK, LBTL, LBTM,
     T LBTQ, LBTR, LBTS, LBTT, LBTMAX
      COMMON/ GEOMF/ SY(42), CY(42), DYB(2,42), DCB(2,4,17),
     A XYF(2,42), SCF(2,42), XYH(2,22), SCH(2,22),
     B XYM(2,14), SCM(2,14), XYT(2,43), SCT(2,43)
      COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, GDYNI, HINF, GMU, UNOGMU
      COMMON/DURA/    L
      COMMON          WY(3,42,9), WC(3,17,17), WF(3,37,13), WH(3,25,17),
     A WM(3,26,26), WT(3,16,21), WSF(3,9,42), WSH(3,9,22), WSM(3,9,14),
     B WST(3,9,43), WSU(3)
C
      WRITE(6,51) L
      GO TO (100,130,140,160),JO
C-----CONSERVATION VARIABLES
C-----OUTER FIELD
  100 WRITE(6,52)
      WRITE(6,53)
      WRITE(6,51) L
      CALL PUTOUT(1,WY,MYT,NYT)
C-----COARSE MESH
      WRITE(6,55)
      WRITE(6,51) L
      CALL PUTOUT(1,WC,MCT,NCT)
C-----FINE MESH
      WRITE(6,57)
      WRITE(6,51) L
      CALL PUTOUT(1,WF,MFT,NFT)
C-----OUTER NOSE REGION
      WRITE(6,69)
      WRITE(6,51) L
      CALL PUTOUT(1,WH,MHT,NHT)
C-----INTERMEDIATE NOSE REGION
      WRITE(6,70)
      WRITE(6,51) L
      CALL PUTOUT(1,WM,MMT,NMT)
```

```
C-----INNER NOSE REGION
      WRITE (6,59)
      WRITE(6,51) L
      CALL PUTOUT(1,WT,MTT,NTT)
C-----PHYSICAL VARIABLES
  130 WRITE (6,61)
C-----OUTER FIELD
      WRITE (6,53)
      WRITE(6,51) L
      CALL PUTOUT(2,WY,MYT,NYT)
C-----COARSE MESH
      WRITE(6,55)
      WRITE(6,51) L
      CALL PUTOUT(2,WC,MCT,NCT)
C-----FINE MESH
  140 WRITE(6,57)
      WRITE(6,51) L
      CALL PUTOUT(2,WF,MFT,NFT)
C-----OUTER NOSE REGION
      WRITE(6,69)
      WRITE(6,51) L
      CALL PUTOUT(2,WH,MHT,NHT)
C-----INTERMEDIATE NOSE REGION
      WRITE(6,70)
      WRITE(6,51) L
      CALL PUTOUT(2,WM,MMT,NMT)
C-----INNER NOSE REGION
      WRITE(6,59)
      WRITE(6,51) L
      CALL PUTOUT(2,WT,MTT,NTT)
C-----AIRFOIL SURFACE
  160 WRITE(6,63)
C-----NOSE REGION HORIZONTAL CUTS
      WRITE(6,59)
      WRITE(6,51) L
      WRITE (6,68)
      WRITE(6,73)
      CALL AIROUT(LBTI,LBTM,WST,LBTMAX,XYT)
      WRITE(6,73)
      CALL AIROUT(LBTQ,LBTT,WST,LBTMAX,XYT)
      WRITE(6,67)
      WRITE(6,73)
      CALL AIPOUT(LBTA,LBTE,WST,LBTMAX,XYT)
C-----INTERMEDIATE NOSE REGION VERTICAL CUTS
      WRITE(6,70)
      WRITE(6,51) L
      WRITE(6,67)
      WRITE(6,73)
      CALL AIROUT(LBMA,LBME,WSM,LBMMAX,XYM)
      WRITE (6,68)
      WRITE(6,73)
      CALL AIROUT(LBMI,LBMM,WSM,LBMMAX,XYM)
C-----OUTER NOSE REGION VERTICAL CUTS
      WRITE(6,69)
      WRITE(6,51) L
      WRITE(6,67)
      WRITE(6,73)
```

```
      CALL AIROUT(LBHA,LBHE,WSH,LBHMAX,XYH)
      WRITE (6,68)
      WRITE(6,73)
      CALL AIROUT(LBHI,LBHM,WSH,LBHMAX,XYH)
C-----FINE MESH VERTICAL CUTS
      WRITE(6,57)
      WRITE(6,51) L
      WRITE(6,67)
      WRITE(6,73)
      CALL AIROUT(LBFA,LBFE,WSF,LBFMAX,XYF)
      WRITE (6,68)
      WRITE(6,73)
      CALL AIROUT(LBFI,LBFM,WSF,LBFMAX,XYF)
      RETURN
C
   11 FORMAT(4I5,5E10.3)
   12 FORMAT(I5,I10,6E20.6)
   51 FORMAT(/4X,2HL=,I5)
   52 FORMAT(//4X,22HCONSERVATION VARIABLES)
   53 FORMAT(//4X,11HOUTER FIELD)
   54 FORMAT(/3X,2HM ,8X,2HN ,17X,4HW(1),16X,4HW(2),16X,4HW(3))
   55 FORMAT(//4X,21HCOARSE CARTESIAN MESH)
   57 FORMAT(//4X,21H  FINE CARTESIAN MESH)
   59 FORMAT(//4X,17HINNER NOSE REGION)
   61 FORMAT(//4X,18HPHYSICAL VARIABLES)
   62 FORMAT(/3X,2HM ,8X,2HN ,12X,8HPRESSURE,13X,7HDENSITY,18X,2HUX,18X,
     12HUY,9X,11HMACH NUMBER,9X,12HTOT.ENTHALPY)
   63 FORMAT(//4X,16HAIRFOIL SURFACES)
   64 FORMAT(//4X,36HHORIZONTAL CUTS IN INNER NOSE REGION)
   65 FORMAT(//4X,34HVERTICAL CUTS IN INNER NOSE REGION)
   66 FORMAT(//4X,33HVERTICAL CUTS IN FINE MESH REGION)
   67 FORMAT(/4X,13HUPPER SURFACE)
   68 FORMAT(/4X,13HLOWER SURFACE)
   69 FORMAT(//4X,17HOUTER NOSE REGION  )
   70 FORMAT(//10X,24HINTERMEDIATE NOSE REGION)
   71 FORMAT(1H0,4X,*VERTICAL CUTS*)
   72 FORMAT(/3X,2HM ,8X,2HN ,12X,8HPRESSURE,13X,7HDENSITY,18X,2HUT,18X,
     12HUN,9X,11HMACH NUMBER,9X,12HTOT.ENTHALPY)
   73 FORMAT(/3X,2HMM,5X,1HX,9X,1HY,9X,1HP,7X,3HRHO,5X,5HU TAN,5X,
    A 5HU NOP,5X,8HMACH NO.,2X,10HTOT.ENTHL.)
      END
```

```
      SUBROUTINE PUTOUT(J,W,MAX,NAX)
      COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, GDYNI, HINF, GMU, UNOGMU
      DIMENSION W(3,MAX,NAX)
      N2 = NAX
      M2 = MAX
      GO TO (1,20),J
    1 DO 10 N = 1,N2
      WRITE(6,54)
      DO 10 M = 1,M2
   10 WRITE(6,12) M,N,(W(K,M,N),K=1,3)
   54 FORMAT(73X,1HM,9X,1HN,18X,4HW(1),16X,4HW(2),16X,4HW(3)   )
   12 FORMAT(I5,I10,6E20.6)
      RETURN
   20 DO 30 N = 1,N2
      WRITE(6,62)
      DO 30 M=1,M2
      RHO = W(3,M,N)
      IF(RHO) 21,21,22
   21 WRITE(6,12) M,N,(W(K,M,N), K = 1,3)
      RHO = 0.010101
   22 P = RHO**GAM
      R = 1.0/RHO
      UX = W(1,M,N)*R
      UY = W(2,M,N)*R
      USQ = UX**2 + UY**2
      AM = SQRT(USQ)*R**HGMU
      HTOT = 2.5*P*R + 0.5*USQ
   30 WRITE(6,12) M,N,P,RHO,UX,UY,AM,HTOT
   62 FORMAT(73X,2HM ,8X,2HN ,12X,8HPRESSURE,13X,7HDENSITY,18X,2HUX,18X,
     1 2HUY,9X,11HMACH NUMBER,9X,12HTOT.ENTHALPY)
      RETURN
      END
```

```
      SUBROUTINE AIROUT (M1, M2, W, MAX, XY)
      COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, QOYNT, HINF, GMU, UNOGMU
      DIMENSION W(3,9,MAX), XY(2,MAX)
      K = 2
      DO 15 M=M1,M2
      RHO = W(3,K,M)
      IF(RHO) 1,1,2
    1 WRITE(6,12) M, (W(J,K,M), J = 1,3)
      RHO = 0.010101
    2 P = RHO**GAM
      R = 1.0 / RHO
      UX = W(1,K,M)*R
      UY = W(2,K,M)*R
C-----UN POSITIVE IS OUT FROM THE SURFACE, UT POSITIVE IN CLOCKWISE DIREC
    3 USQ = UX**2 + UY**2
      AM = SQRT(USQ)*R**HGMU
      HTOT = 2.5*P*R + 0.5*USQ
   15 WRITE(6,12) M, XY(1,M), XY(2,M), P, RHO, UX, UY, AM, HTOT
   12 FORMAT (I5,8F10.5)
      RETURN
      END
```

```
      SUBROUTINE YTRE
C
      COMMON/LIMITS/ MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT,
     C MCA, MCB, MCE, MCF, MCG, MCH, MCS, MCT, NCA, NCB, NCE, NCF,
     C NCG, NCH, NCS, NCT,
     F MFA, MFB, MFD, MFE, MFF, MFG, MFH, MFQ, MFS, MFT, MFAN,
     F MFANP, MFAT, MFATM, MFATP, NFA, NFB, NFD, NFE, NFF, NFG, NFH,
     F NFQ, NFS, NFT, NFCL,
     H MHA, MHB, MHD, MHE, MHF, MHG, MHH, MHQ, MHS, MHT, MHAN, MHANP,
     H NHA, NHB, NHD, NHE, NHF, NHG, NHH, NHQ, NHS, NHT, NHCL,
     M MMA, MMB, MMD, MME, MMF, MMG, MMH, MMQ, MMS, MMT, MMAN, MMANP,
     M NMA, NMB, NMD, NME, NMF, NMG, NMH, NMQ, NMS, NMT, NMCL,
     T MTA, MTB, MTD, MTQ, MTS, MTT, MTAN, MTANP, NTA, NTB, NTD,
     T NTQ, NTS, NTT, NTCL,
     S MSL, MST, NSL, NST
      COMMON/ZMESH/  DX(6), DY(6), RH(9)
      COMMON/ GEOMF/ SY(42), CY(42), DYB(2,42), DCB(2,4,17),
     A XYF(2,42), SCF(2,42), XYH(2,22), SCH(2,22),
     B XYM(2,14), SCM(2,14), XYT(2,43), SCT(2,43)
      COMMON/CON1/JR, MR, NR, DTDX, DTDY, F(3,5,26), G(3,5,26),
     1 FP(3,3,3), GP(3,3,3), WAD(3,3,26)
      COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, QDYNI, HINF, GMU, UNOGMU
      COMMON/TOC/PRMSHY,ATSFY,HGPU,HGPV,HGPW,ATSF(6),TSN(6),PRMSH(6),
     1 PRMSHI(6), PRMYI, CFTT
      COMMON/PEND/CJX(3,3), DCX(3,6), DCY(3,6), DHD(3,3), DMBX(5,6),
     A DMBY(5,6), LBIX(6), LBIY(6)
      COMMON          WY(3,42,9), WC(3,17,17), WF(3,37,13), WH(3,25,17),
     A WM(3,26,26), WT(3,16,21), WSF(3,9,42), WSH(3,9,22), WSM(3,9,14),
     B WST(3,9,43), WSU(3)
C                  .
      DIMENSION WW(3)
C-----YTRE FIELD
      KR = 1
      M = 1
      J = 1
C-----F AND G CONVERSION
  402 DO 404 N = 1,NYT
      P = UNOG*WY(3,M,N)**GAM
      UXP = WY(1,M,N)/WY(3,M,N)
      UYP = WY(2,M,N)/WY(3,M,N)
      F(1,J,N) = -WY(1,M,N)*UXP - P
      F(2,J,N) = -WY(1,M,N)*UYP
      F(3,J,N) = -WY(1,M,N)
      G(1,J,N) = F(2,J,N)
      G(2,J,N) = -WY(2,M,N)*UYP - P
  404 G(3,J,N) = -WY(2,M,N)
      GO TO (406,408,410),J
  406 M = 2
      J = 2
      GO TO 402
  408 M = M + 1
      J = 3
      GO TO 402
  410 M = M - 1
      MM2 = M - 2
      DO 600 N = 2,NYS
```

```
C-----ALLOWABLE STEP
      RUX      = ABS(-WY(1,M,N)*SY(M) + WY(2,M,N)*CY(M))*PRMSHY
      RUY      = ABS(WY(1,M,N)*CY(M) +  WY(2,M,N)*SY(M))
      RMU      = WY(3,M,N) - 1.0
      RA       = ATSFY*(1.0 + RMU*HGPU*(1.0 + RMU*HGPV*(1.0 + RMU*HGPW)))
      CFTT = TSN(1)*WY(3,M,N)/(RUX + RUY + RA)
C-----ADVANCED SOLUTION AT A POINT
      NM2 = N - 2
C-----HALF STEPS
      KTS = 1
      IA = 2
      JA = 1
  602 MM = MM2 + IA
      NN = NM2 + JA
      DO 640 K = 1,3
      IF (NN - NYS) 603,603,632
  603 SGR = SGT = SFR = SFT = 0.0
      DO 624 JL = 1,3
      SGT = SGT + CJX(IA,JL)*G(K,JL,NN)
      SFT = SFT + CJX(IA,JL)*F(K,JL,NN)
      JN = NM2 + JL
      SGR = SGR - CJX(JA,JL)*G(K,IA,JN)
  624 SFR = SFR - CJX(JA,JL)*F(K,IA,JN)
C-----W VALUE ON HALF STEP LEVEL
      WW(K) = WY(K,MM,NN) + CFTT*(-RH(N)*PRMYI*(SFR*CY(M) + SGR*SY(M))
     A -SFT*SY(M) + SGT*CY(M))
      GO TO 640
C     ASSIGNMENT OF POINT AT INFINITY
  632 WW(K) = WY(K,MM,NN)
  640 CONTINUE
      PA = UNOG*WW(3)**GAM
      R = 1.0/WW(3)
      U1A = WW(1)*R
      U2A = WW(2)*R
      FP(1,IA,JA) = -WW(1)*U1A - PA
      FP(2,IA,JA) = -WW(1)*U2A
      FP(3,IA,JA) = -WW(1)
      GP(1,IA,JA) = FP(2,IA,JA)
      GP(2,IA,JA) = -WW(2)*U2A - PA
      GP(3,IA,JA) = -WW(2)
C-----OTHER AUXILIARY POINTS
      GO TO (642,644,646,650), KTS
  642 KTS = 2
      JA = 3
      GO TO 602
  644 KTS = 3
      JA = 2
      IA = 1
      GO TO 602
  646 KTS = 4
      IA = 3
      GO TO 602
C-----SECOND STEP
  650 DO 656 K = 1,3
  656 WAU(K,2,N) = 2.0*DCX(1,KR)*WY(K,M,N)
     A +DCX(2,KR)*(WY(K,M+1,N)+WY(K,M-1,N)+WY(K,M,N+1)+WY(K,M,N-1))
     B +CFTT*(-RH(N)*PRMYI*((FP(K,2,1) - FP(K,2,3))*CY(M)
```

```
     C +(GP(K,2,1) - GP(K,2,3))*SY(M)) -(FP(K,3,2) - FP(K,1,2))*SY(M)
     D +(GP(K,3,2) - GP(K,1,2))*CY(M))
  660 CONTINUE
C-----COLUMN SHIFTING
      IF (M - MYS) 422,426,426
  422 DO 424 J = 1,2
      DO 424 N = 1,NYT
      DO 424 K = 1,3
      F(K,J,N) = F(K,J+1,N)
  424 G(K,J,N) = G(K,J+1,N)
  426 IF (M - 2) 432,432,428
  428 DO 430 N = 2,NYS
      DO 430 K = 1,3
  430 WY(K,M-1,N) = WAD(K,1,N)
  432 DO 434 N = 2,NYS
      DO 434 K = 1,3
  434 WAD(K,1,N) = WAD(K,2,N)
      IF(M - MYS) 440,436,436
  436 DO 438 N = 2,NYS
      DO 438 K = 1,3
  438 WY(K,M,N) = WAD(K,2,N)
      RETURN
  440 M = M + 1
      GO TO 408
      END
```

```
      SUBROUTINE OOKII
C-----COARSE CARTESIAN MESH
C
      COMMON/LIMITS/ MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT,
     C MCA, MCB, MCE, MCF, MCG, MCH, MCS, MCT, NCA, NCB, NCE, NCF,
     C NCG, NCH, NCS, NCT,
     F MFA, MFB, MFD, MFE, MFF, MFG, MFH, MFQ, MFS, MFT, MFAN,
     F MFANP, MFAT, MFATM, MFATP, NFA, NFB, NFD, NFE, NFF, NFG, NFH,
     F NFQ, NFS, NFT, NFCL,
     H MHA, MHB, MHD, MHE, MHF, MHG, MHH, MHQ, MHS, MHT, MHAN, MHANP,
     H NHA, NHB, NHD, NHE, NHF, NHG, NHH, NHQ, NHS, NHT, NHCL,
     M MMA, MMB, MMD, MME, MMF, MMG, MMH, MMQ, MMS, MMT, MMAN, MMANP,
     M NMA, NMB, NMD, NME, NMF, NMG, NMH, NMQ, NMS, NMT, NMCL,
     T MTA, MTB, MTD, MTQ, MTS, MTT, MTAN, MTANP, NTA, NTB, NTD,
     T NTQ, NTS, NTT, NTCL,
     S MSL, MST, NSL, NST
      COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, QDYNI, HINF, GMU, UNOGMU
      COMMON/OPLM/   ICO(4,17), JCO(4,17), IFO(8,37), JFO(8,37),
     A IHO(8,25), JHO(8,25), IMO(8,26), JMO(8,26), ITO(8,16), JTO(8,16)
      COMMON/TUC/PRMSHY,ATSFY,HGPU,HGPV,HGPW,ATSF(6),TSN(6),PRMSH(6),
     1 PRMSHI(6), PRMYI, CFTT
      COMMON/CON1/JR, MR, NR, DTDX, DTDY, F(3,5,26), G(3,5,26),
     1 FP(3,3,3), GP(3,3,3), WAD(3,3,26)
      COMMON/PLTD/CJX(3,3), DCX(3,6), DCY(3,6), DHD(3,3), DMBX(5,6),
     A DMBY(5,6), LBIX(6), LBIY(6)
      COMMON          WY(3,42,9), WC(3,17,17), WF(3,37,13), WH(3,25,17),
     A WM(3,26,26), WT(3,16,21), WSF(3,9,42), WSH(3,9,22), WSM(3,9,14),
     B WST(3,9,43), WSU(3)
      DIMENSION WW(3)
C
      KR = 2
      DCP = DCX(1,KR) + DCY(1,KR)
      M = MCA
      J = 1
C-----F AND G CONVERSIONS
  502 DO 504 N = NCA,NCT
      P = UNOG*WC(3,M,N)**GAM
      RI = 1.0/WC(3,M,N)
      F(1,J,N) = -WC(1,M,N)**2*RI - P
      F(2,J,N) = -WC(1,M,N)*WC(2,M,N)*RI
      F(3,J,N) = -WC(1,M,N)
      G(1,J,N) = F(2,J,N)
      G(2,J,N) = -WC(2,M,N)**2*RI - P
  504 G(3,J,N) = -WC(2,M,N)
      GO TO (506, 508, 510), J
  506 M = MCB
      J = 2
      GO TO 502
  508 M = M + 1
      J = 3
      GO TO 502
  510 M = M - 1
      MM2 = M - 2
      KK =1
      KOT = 1
  512 NL1 = ICO(KK,M)
```

```
      NL2 = ICO(KK+1,M) - 1
      JCT = JCO(KK,M)
      KK = KK + 1
      IF(NL2 - NCS) 516,514,514
  514 NL2 = NCS
      KOT = 2
  516 IF(JCT)520,520,518
  518 DO 440 N = NL1,NL2
C-----ALLOWABLE STEP
      RMU = WC(3,M,N) - 1.0
      RA = (1.0 + RMU*HGPU*(1.0 + RMU*HGPV*(1.0 + RMU*HGPW)))*ATSF(KR)
      DTDY = TSN(KR)*WC(3,M,N)/(ABS(WC(1,M,N))*PRMSH(KR)
     A + ABS(WC(2,M,N)) + RA)
      DTDX = PRMSH(KR)*DTDY
      NM2 = N - 2
C-----HALF STEPS
      JA = 1
      MM = M
      NN = N - 1
  402 DO 404 K = 1,3
  404 WW(K) = WC(K,MM,NN)+ DTDX*CJX(2,3)*(F(K,3,NN) - F(K,1,NN))
     A + DTDY*(CJX(JA,1)*G(K,2,N-1) + CJX(JA,2)*G(K,2,N)
     B + CJX(JA,3)*G(K,2,N+1))
      RI = 1.0/WW(3)
      GP(1,2,JA) = -WW(1)*WW(2)*RI
      GP(2,2,JA) = -WW(2)**2*RI - UNOG*WW(3)**GAM
      GP(3,2,JA) = -WW(2)
      IF(JA - 1) 406,406,410
  406 JA = 3
      NN = N + 1
      GO TO 402
  410 NN = N
      MM = M - 1
      IA = 1
  412 DO 414 K = 1,3
  414 WW(K) = WC(K,MM,NN)+ DTDX*(CJX(IA,1)*F(K,1,NN)
     A + CJX(IA,2)*F(K,2,NN) + CJX(IA,3)*F(K,3,NN))
     B + DTDY*CJX(2,3)*(G(K,2,N+1) - G(K,2,N-1))
      RI = 1.0/WW(3)
      FP(1,IA,2) = -WW(1)**2*RI - UNOG*WW(3)**GAM
      FP(2,IA,2) = -WW(1)*WW(2)*RI
      FP(3,IA,2) = -WW(1)
      IF(IA - 1) 416,416,430
  416 IA = 3
      MM = M + 1
      GO TO 412
C-----SECOND STEP
  430 DO 432 K = 1,3
  432 WAD(K,2,N) = DCP*WC(K,M,N) + DCX(2,KR)*(WC(K,M+1,N)+WC(K,M-1,N))
     A +DCY(2,KR)*(WC(K,M,N+1)+WC(K,M,N-1)) + DTDX*(FP(K,3,2)-FP(K,1,2))
     B +DTDY*(GP(K,2,3) - GP(K,2,1))
  440 CONTINUE
      GO TO(512,530),KOT
  520 DO 522 N = NL1,NL2
      DO 522 K = 1,3
  522 WAD(K,2,N) = WC(K,M,N)
      GO TO (512,530),KOT
```

```
C-----COLUMN SHIFTING
  530 IF(M - MCS) 532,538,538
  532 DO 534 N = NCA,NCT
      DO 534 K = 1,3
      DO 534 JL = 1,2
      G(K,JL,N) = G(K,JL+1,N)
  534 F(K,JL,N) = F(K,JL+1,N)
  536 IF(M - MCH) 542,542,538
  538 DO 540 N = NCB,NCS
      DO 540 K = 1,3
  540 WC(K,M-1,N) = WAD(K,1,N)
  542 DO 544 N = NCB,NCS
      DO 544 K = 1,3
  544 WAD(K,1,N) = WAD(K,2,N)
      IF(M-MCS)546,548,548
  546 M = M + 1
      GO TO 508
  548 DO 550 N = NCB,NCS
      DO 550 K = 1,3
  550 WC(K,M,N) = WAD(K,2,N)
      RETURN
      END
```

```
      SUBROUTINE SUSUMU(KR,MRA,MRT,NRA,NRT,W,IRO,JRO,ITSMD)
C-----ADVANCES SOLUTION IN RECTANGULAR CARTESIAN FIELD BY
C-----LAX TYPE SCHEME, CELL TYPES DETERMINED BY IRO, JRO ARRAYS.
C     LIGHT DAMPING ON ALL CENTERED DIFFERENCES, RUN L33 FF
C
      COMMON/TOC/PRMSHY,ATSFY,HGPU,HGPV,HGPW,ATSF(6),TSN(6),PRMSH(6),
     1 PRMSHI(6), PRMYI, CFTT
      COMMON/OPLM/   ICO(4,17), JCO(4,17), IFO(8,37), JFO(8,37),
     A IHO(8,25), JHO(8,25), IMO(8,26), JMO(8,26), ITO(8,16), JTO(8,16)
      COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, GDYNI, HINF, GMU, UNOGMU
      COMMON/CON1/JR, MR, NR, DTDX, DTDY, F(3,5,26), G(3,5,26),
     1 FP(3,3,3), GP(3,3,3), WAD(3,3,26)
      COMMON/PEND/CJX(3,3), DCX(3,6), DCY(3,6), DHD(3,3), DMBX(5,6),
     A DMBY(5,6), LBIX(6), LBIY(6)
C
      DIMENSION W(3,MRT,NRT), IRO( 8,MRT),JRO( 8,MRT),WW(3)
C
      MX = MRT
      NX = NRT
      MRB = MRA + 1
      MRS = MRT - 1
      NRB = NRA + 1
      NRS = NRT - 1
C-----FIRST COLUMN
      M = MRA
      J = 2
      NLL = NRA
      NUL = NRT
C-----F AND G CONVERSIONS
  610 DO 614 N = NLL,NUL
      P = UNOG*W(3,M,N)**GAM
      RI = 1.0/W(3,M,N)
      F(1,J,N) = -W(1,M,N)**2*RI - P
      F(2,J,N) = -W(1,M,N)*W(2,M,N)*RI
      F(3,J,N) = -W(1,M,N)
      G(1,J,N) = F(2,J,N)
      G(2,J,N) = -W(2,M,N)**2*RI - P
  614 G(3,J,N) = -W(2,M,N)
      GO TO (669,620,630,632,668), J
C-----SECOND COLUMN OF F
  620 M = MRB
      J = 3
      GO TO 610
C-----SUBSEQUENT COLUMNS
  630 M = M + 1
      J = 4
      NLL = NRA
      NUL = NRT
      GO TO 610
  632 M = M - 1
  639 KK = 1
      KOT = 1
  640 NL1 = IRO(KK,M)
      NL2 = IRO(KK+1,M) - 1
      JHC = JRO(KK;M)
      JDR = JHC/100
```

```
      JCT = JRC - 100*JDR
      KK = KK + 1
      IF(NL2 - NRS)644,642,642
  642 NL2 = NRS
      KOT = 2
  644 IF(JCT) 680,680,645
  645 IF(JCT-9) 646,646,657
  646 GO TO (650,651,652,656,657,658,662,663,664),JCT
C-----TYPE 1 CELL
  650 J = 5
      M = M + 2
      ASSIGN 110 TO JSWCH
      GO TO 654
C-----TYPE 2 CELL
  651 ASSIGN 120 TO JSWCH
      GO TO 100
C-----TYPE 3 CELL
  652 J = 1
      M = M - 2
      ASSIGN 130 TO JSWCH
  654 NLL = NL1
      NUL = NL2 + 2
      GO TO 610
C-----TYPE 4 CELL
  656 J = 5
      M = M + 2
      ASSIGN 140 TO JSWCH
      GO TO 660
C-----TYPE 5 CELL
  657 ASSIGN 150 TO JSWCH
      GO TO 100
C-----TYPE 6 CELL
  658 J = 1
      M = M - 2
      ASSIGN 160 TO JSWCH
  660 NLL = NL1 - 1
      NUL = NL2 + 1
      GO TO 610
C-----TYPE 7 CELL
  662 J = 5
      M = M + 2
      ASSIGN 170 TO JSWCH
      GO TO 666
C-----TYPE 8 CELL
  663 ASSIGN 180 TO JSWCH
      GO TO 100
C-----TYPE 9 CELL
  664 J =1
      M = M - 2
      ASSIGN 190 TO JSWCH
  666 NLL = NL1 - 2
      NUL = NL2
      GO TO 610
  668 M = M - 2
      GO TO 100
  669 M = M + 2
C-----ADVANCE SOLUTION OF POINTS IN COLUMN HAVING SIMILAR TYPE CELLS
```

```
C      THESE VALUES FOR MR,NR,JR WORK ONLY FOR TYPES 5,10,11,12,13 CELLS
  100 MR = M- 2
      JR = 1
C-----NEED FOR EXTRA DAMPING NEAR FIELD OUTER BOUNDARIES
      IF(JCT-9) 300,300,301
  300 GO TO (105,301,105,301,301,301,105,301,105),JCT
  301 IF(JDR) 309,309,302
  302 GO TO(304,308,306,304,309,306,304,308,306),JDR
C-----LEFT BOUNDARY COLUMNS
  304 JDMX = LBIX(KR) + MRA - M + 1
  305 DMBXT = DMBX(JDMX,KR)
      GO TO 105
C-----RIGHT BOUNDARY COLUMNS
  306 JDMX = M - MRT + LBIX(KR) + 1
      GO TO 305
C-----INTERMEDIATE COLUMNS
  308 DMBXT = 1.0
      GO TO 105
C-----CENTRAL FIELD
  309 DDCX = DCX(1,KR)
      DDCY = DCY(1,KR)
      DDXE = DCX(2,KR)
      DDYE = DCY(2,KR)
      JDR = 5
  105 DO 200 N = NL1,NL2
C-----ALLOWABLE STEP
      GO TO (318,319),ITSMD
  318 RMU = W(3,M,N) - 1.0
      RA = (1.0 + RMU*HGPU*(1.0 + RMU*HGPV*(1.0 + RMU*HGPW)))*ATSF(KR)
      DTDY = TSN(KR)*W(3,M,N)/(ABS(W(1,M,N))*PRMSH(KR) + ABS(W(2,M,N))
     A + RA)
      DTDX = PRMSH(KR)*DTDY
C      THESE VALUES FOR MR,NR,JR WORK ONLY FOR TYPES 5,10,11,12,13 CELLS
  319 NR = N - 2
      GO TO (109,320,109,320,320,320,109,320,109,320,320,320,320),JCT
  320 GO TO(321,321,321,324,109,324,327,327,327),JDR
C-----BOTTOM BOUNDARY ROWS
  321 JDNY = LBIY(KR) + NRA - N + 1
  322 DMBYT = DMBY(JDNY,KR)
      GO TO 330
C-----TOP BOUNDARY ROWS
  327 JDNY = N - NRT + LBIY(KR) + 1
      GO TO 322
C-----INTERMEDIATE ROWS
  324 DMBYT = 1.0
C-----DAMPING ALTERATIONS
  330 DMULT = AMAX1(DMBXT,DMBYT)
      DDCX = 0.5 - DMULT*(0.5 - DCX(1,KR))
  333 DDCY = 0.5 - DMULT*(0.5 - DCY(1,KR))
  335 DDXE = DMULT*DCX(2,KR)
      DDYE = DMULT*DCY(2,KR)
  109 GO TO JSWCH
C-----TYPE 1 CELL, POINT IS IN LOWER LEFT CORNER OF ARRAY
  110 CALL WOAP(1,1,W,MX,NX)
      CALL WOAP(1,3,W,MX,NX)
      CALL WOAP(3,1,W,MX,NX)
      DO 114 K = 1,3
```

```
  114 WAD(K,3,N) = 2.0*DHD(1,1)*W(K,M,N)
     A + DHD(1,2)*(W(K,M+1,N) + W(K,M,N+1))
     B + DHD(1,3)*(W(K,M+2,N) + W(K,M,N+2))
     C + DTDX*(FP(K,3,1) - FP(K,1,1))
     D + DTDY*(GP(K,1,3) - GP(K,1,1))
      GO TO 200
C-----TYPE 2 CELL, POINT IS AT BOTTOM CENTER OF ARRAY
  120 CALL WOAP(1,1,W,MX,NX)
      CALL WOAP(2,1,W,MX,NX)
      CALL WOAP(2,3,W,MX,NX)
      CALL WOAP(3,1,W,MX,NX)
      DO 124 K = 1,3
  124 WAD(K,3,N) = (DHD(1,1) + DUCX     )*W(K,M,N)
     A + DUXE     *(W(K,M+1,N) + W(K,M-1,N))
     B + DHD(1,2)*W(K,M,N+1) + DHD(1,3)*W(K,M,N+2)
     C + DTDX*(FP(K,3,1) - FP(K,1,1))
     D + DTDY*(GP(K,2,3) - GP(K,2,1))
      GO TO 200
C-----TYPE 3 CELL, POINT IS AT LOWER RIGHT CORNER OF ARRAY
  130 CALL WOAP(1,1,W,MX,NX)
      CALL WOAP(3,1,W,MX,NX)
      CALL WOAP(3,3,W,MX,NX)
      DO 134 K = 1,3
  134 WAD(K,3,N) = 2.0*DHD(1,1)*W(K,M,N)
     A + DHD(1,2)*(W(K,M-1,N) + W(K,M,N+1))
     B + DHD(1,3)*(W(K,M-2,N) + W(K,M,N+2))
     C + DTDX*(FP(K,3,1) - FP(K,1,1))
     D + DTDY*(GP(K,3,3) - GP(K,3,1))
      GO TO 200
C-----TYPE 4 CELL, POINT IS AT LEFT CENTER OF ARRAY
  140 CALL WOAP(1,1,W,MX,NX)
      CALL WOAP(1,2,W,MX,NX)
      CALL WOAP(1,3,W,MX,NX)
      CALL WOAP(3,2,W,MX,NX)
      DO 144 K = 1,3
  144 WAD(K,3,N) = (DHD(1,1) + DUCY     )*W(K,M,N)
     A + DUYE     *(W(K,M,N+1) + W(K,M,N-1))
     B + DHD(1,2)*W(K,M+1,N) + DHD(1,3)*W(K,M+2,N)
     C + DTDX*(FP(K,3,2) - FP(K,1,2))
     D + DTDY*(GP(K,1,3) - GP(K,1,1))
      GO TO 200
C-----TYPE 6 CELL, POINT IS AT RIGHT CENTER OF ARRAY
  160 CALL WOAP(3,1,W,MX,NX)
      CALL WOAP(1,2,W,MX,NX)
      CALL WOAP(3,2,W,MX,NX)
      CALL WOAP(3,3,W,MX,NX)
      DO 164 K = 1,3
  164 WAD(K,3,N) = (DHD(1,1) + DUCY     )*W(K,M,N)
     A + DUYE     *(W(K,M,N+1) + W(K,M,N-1))
     B + DHD(1,2)*W(K,M-1,N) + DHD(1,3)*W(K,M-2,N)
     C + DTDX*(FP(K,3,2) - FP(K,1,2))
     D + DTDY*(GP(K,3,3) - GP(K,3,1))
      GO TO 200
C-----TYPE 7 CELL, POINT IS AT UPPER LEFT CORNER OF ARRAY
  170 CALL WOAP(1,1,W,MX,NX)
      CALL WOAP(1,3,W,MX,NX)
      CALL WOAP(3,3,W,MX,NX)
```

```
      DO 174 K = 1,3
  174 WAD(K,3,N) = 2.0*DHD(1,1)*W(K,M,N)
     A + DHD(1,2)*(W(K,M+1,N) + W(K,M,N-1))
     B + DHD(1,3)*(W(K,M+2,N) + W(K,M,N-2))
     C + DTDX*(FP(K,3,3) - FP(K,1,3))
     D + DTDY*(GP(K,1,3) - GP(K,1,1))
      GO TO 200
C-----TYPE 8 CELL, POINT IS AT TOP CENTER OF ARRAY
  180 CALL WOAP(1,3,W,MX,NX)
      CALL WOAP(2,1,W,MX,NX)
      CALL WOAP(2,3,W,MX,NX)
      CALL WOAP(3,3,W,MX,NX)
      DO 184 K = 1,3
  184 WAD(K,3,N) = (DHD(1,1) + DUCX      )*W(K,M,N)
     A + DUXE      *(W(K,M+1,N) + W(K,M-1,N))
     B + DHD(1,2)*W(K,M,N-1) + DHD(1,3)*W(K,M,N-2)
     C + DTDX*(FP(K,3,3) - FP(K,1,3))
     D + DTDY*(GP(K,2,3) - GP(K,2,1))
      GO TO 200
C-----TYPE 9 CELL, POINT IS AT UPPER RIGHT CORNER OF ARRAY
  190 CALL WOAP(1,3,W,MX,NX)
      CALL WOAP(3,1,W,MX,NX)
      CALL WOAP(3,3,W,MX,NX)
      DO 194 K = 1,3
  194 WAD(K,3,N) = 2.0*DHD(1,1)*W(K,M,N)
     A + DHD(1,2)*(W(K,M-1,N) + W(K,M,N-1))
     B + DHD(1,3)*(W(K,M-2,N) + W(K,M,N-2))
     C + DTDX*(FP(K,3,3) - FP(K,1,3))
     D + DTDY*(GP(K,3,3) - GP(K,3,1))
      GO TO 200
C-----TYPE 5 CELL, POINT IS AT CENTER OF ARRAY
  150 IF(JCT-9) 400,400,380
C-----TYPES 10,11,12,13
  380 CALL WOAP(1,2,W,MX,NX)
      CALL WOAP(2,1,W,MX,NX)
      CALL WOAP(2,3,W,MX,NX)
      CALL WOAP(3,2,W,MX,NX)
      JCTP = JCT - 9
      CALL WOP (JCTP,W,MX,NX)
      GO TO 430
C-----HALF STEPS
  400 JA = 1
      MM = M
      NN = N - 1
  402 DO 404 K = 1,3
  404 WW(K) = W(K,MM,NN) + DTDX*CJX(2,3)*(F(K,4,NN) - F(K,2,NN))
     A + DTDY*(CJX(JA,1)*G(K,3,N-1) + CJX(JA,2)*G(K,3,N)
     B + CJX(JA,3)*G(K,3,N+1))
      RI = 1.0/WW(3)
      GP(1,2,JA) = -WW(1)*WW(2)*RI
      GP(2,2,JA) = -WW(2)**2*RI - UNOG*WW(3)**GAM
      GP(3,2,JA) = -WW(2)
      IF(JA - 1) 406,406,410
  406 JA = 3
      NN = N + 1
      GO TO 402
  410 NN = N
```

```
      MM = M - 1
      IA = 1
  412 DO 414 K = 1,3
  414 WW(K) = W(K,MM,NN) + DTDX*(CJX(IA,1)*F(K,2,NN)
     A + CJX(IA,2)*F(K,3,NN) + CJX(IA,3)*F(K,4,NN))
     B + DTDY*CJX(2,3)*(G(K,3,N+1) - G(K,3,N-1))
      RI = 1.0/WW(3)
      FP(1,IA,2) = -WW(1)**2*RI - UNOG*WW(3)**GAM
      FP(2,IA,2) = -WW(1)*WW(2)*RI
      FP(3,IA,2) = -WW(1)
      IF(IA - 1) 416,416,430
  416 IA = 3
      MM = M + 1
      GO TO 412
C-----SECOND STEP
  430 DO 154 K = 1,3
  154 WAD(K,3,N) = (DUCX+DUCY)*W(K,M,N) + DUXE*(W(K,M+1,N)+W(K,M-1,N))
     A + DUYE*(W(K,M,N+1)+W(K,M,N-1))
     B +DTDX*(FP(K,3,2)-FP(K,1,2)) + DTDY*(GP(K,2,3)-GP(K,2,1))
  200 CONTINUE
      GO TO (640,700),KOT
C-----INERT CELLS
  680 DO 682 N = NL1,NL2
      DO 682 K = 1,3
  682 WAD(K,3,N) = W(K,M,N)
      GO TO (640,700),KOT
C-----COLUMN SHIFTING
  700 IF(M - MRS)702,706,706
  702 DO 704 N = NRA,NRT
      DO 704 K = 1,3
      DO 704 JL = 2,3
      F(K,JL,N) = F(K,JL+1,N)
  704 G(K,JL,N) = G(K,JL+1,N)
      IF(M -MPD-1) 710,710,706
  706 DO 708N=NRB,NRS
      DO 708 K = 1,3
  708 W(K,M-2,N) = WAD(K,1,N)
  710 DO 712 N = NRB,NRS
      DO 712 K = 1,3
      DO 712 JL = 1,2
  712 WAD(K,JL,N) = WAD(K,JL+1,N)
      IF(M-MRS)714,716,716
  714 M =  M + 1
      GO TO 630
  716 DO 718 N = NRB,NRS
      DO 718 K = 1,3
      W(K,M-1,N) = WAD(K,1,N)
  718 W(K,M,N) = WAD(K,2,N)
      RETURN
      END
```

```
      SUBROUTINE WOP(I,W,M,N)
C     I            JCT          IPX          IPY
C     1            10           1            1
C     2            11           2            3
C     3            12           3            2
C     4            13           4            4
      COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, QDYNI, HINF, GMU, UNOGMU
      COMMON/CONI/JR, MR, NR, DTDX, DTDY, F(3,5,26), G(3,5,26),
     1 FP(3,3,3), GP(3,3,3), WAD(3,3,26)
      COMMON/PEND/CJX(3,3), DCX(3,6), DCY(3,6), DHD(3,3), DMBX(5,6),
     A DMBY(5,6), LBIX(6), LBIY(6)
      DIMENSION WW(3),W(3,M,N),ZCX(4,3,3)
      DATA ZCX/0.0,-1.0,0.5,-0.5,-1.0,1.0,-1.0,1.0,1.0,0.0,0.5,-0.5,-1.0
     1,1.0,-1.0,1.0,2.0,-2.0,2.0,-2.0,-1.0,1.0,-1.0,1.0,0.5,-0.5,0.0,
     2-1.0,-1.0,1.0,-1.0,1.0,0.5,-0.5,1.0,0.0/
      EQUIVALENCE (W1,WW(1)),(W2,WW(2)),(W3,WW(3))
      IPX = I
      IA = 2
      JA = 1
      II = 3
      IF(I.GE.3) JA = 3
      MM = MR + IA
      NN = NR + JA
      DO 16 KA = 1,3
      SF = SG = 0.0
      DO 10 JL = 1,3
      JN = NR + JL
      SG = SG + CJX(JA,JL)*G(KA,II,JN)
      DO 10 JJ = 1,3
      JI = JR + JJ
   10 SF = SF + ZCX(IPX,JJ,JL)*F(KA,JI,JN)
   16 WW(KA) = W(KA,MM,NN) + DTDX*SF + DTDY*SG
      IFLAG = 1
   17 PA = UNOG*W3**GAM
      R = 1.0 / W3
      U1A = W1*R
      U2A = W2*R
      FP(1,IA,JA) = - W1*U1A - PA
      FP(2,IA,JA) = - W1*U2A
      FP(3,IA,JA) = - W1
      GP(1,IA,JA) = FP(2,IA,JA)
      GP(2,IA,JA) = - W2*U2A - PA
      GP(3,IA,JA) = - W2
      IF(IFLAG.GT.1) RETURN
      JA = 2
      IA = 1
      IF(IPX.EQ.2.OR.IPX.EQ.4) IA = 3
      IPY = IPX
      IF(IPX.EQ.2) IPY = 3
      IF(IPX .EQ.3) IPY = 2
      MM = MR + IA
      NN = NR + JA
      DO 26 KA = 1,3
      SF = SG = 0.0
      DO 20 JL = 1,3
      JN = NR + JL
```

```
      JI = JR + JL
      SF = SF + CJX(IA,JL) * F(KA,JI,NN)
      DO 20 JJ = 1,3
      II = JJ + JR
   20 SG = SG + ZCX(IPY,JL,JJ)*G(KA,II,NN)
   26 WW(KA) = W(KA,MM,NN) + DTDX*SF + DTDY*SG
      IFLAG = 2
      GO TO 17
      END
```

```
      SUBROUTINE WOAP(IIA,JJA,W,M,N)
      COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, ODYNI, HINF, GMU, UNOGMU
      COMMON/CONI/JR, MR, NR, DTDX, DTDY, F(3,5,26), G(3,5,26),
     1 FP(3,3,3), GP(3,3,3), WAD(3,3,26)
      COMMON/PLND/CJX(3,3), DCX(3,6), DCY(3,6), DHD(3,3), DMBX(5,6),
     A DMBY(5,6), LBIX(6), LBIY(6)
      DIMENSION WW(3),W(3,M,N)
      EQUIVALENCE (W1,WW(1)),(W2,WW(2)),(W3,WW(3))
      IA = IIA
      JA = JJA
      II = IA + JR
      MM = MR + IA
      NN = NR + JA
      DO 16 KA=1,3
      SF = SG = 0.0
      DO 10 JL=1,3
      JN = NR + JL
      JJ = JR + JL
      SF = SF + CJX(IA,JL)*F(KA,JJ,NN)
   10 SG = SG + CJX(JA,JL)*G(KA,II,JN)
   16 WW(KA) = W(KA,MM,NN) + DTDX*SF + DTDY * SG
      PA = UNOG*W3**GAM
      R  = 1.0/W3
      U1A = W1*R
      U2A = W2*R
      FP(1,IA,JA) = - W1*U1A - PA
      FP(2,IA,JA) = - W1*U2A
      FP(3,IA,JA) = - W1
      GP(1,IA,JA) = FP(2,IA,JA)
      GP(2,IA,JA) = - W2*U2A - PA
      GP(3,IA,JA) = - W2
      RETURN
      END
```

```
      SUBROUTINE HERI(KR,LBX,W,ITSMD)
C-----ADVANCES SOLUTION AT AIRFOIL SURFACE POINTS AND AT ONE ROW OF OUTPO
C-----MESH POINTS USING LOCAL CARTESIAN MESH TANGENT    SURFACE
C-----ADVANCES SOLUTION AT 2 POINTS IN LOCAL CARTESIAN MESH TANGENT TO SU
      COMMON/PLIMIT/  LBFA, LBFB, LBFC, LBFD, LBFE, LPFI, LBFJ, LBFK,
     F LBFL, LBFM, LBFMAX,
     H LBHA, LBHB, LBHC, LBHD, LBHE, LBHI, LBHJ, LBHK, LBHL, LBHM,
     H LBHMAX,
     M LBMA, LBMB, LBMC, LBMD, LBME, LBMI, LBMJ, LBMK, LBML, LBMM,
     M LBMMAX,
     T LBTA, LBTB, LBTC, LBTD, LBTE, LBTI, LBTJ, LBTK, LBTL, LBTM,
     T LBTQ, LBTR, LBTS, LBTT, LBTMAX
      COMMON/GAS/   GAM, HGMU, UNOG, DOGMU, GMMYI, HINF, GMU, UNOGMU
      COMMON/CON1/JR, MR, NR, DTDX, DTDY, F(3,5,26), G(3,5,26),
     1 FP(3,3,3), GP(3,3,3), WAD(3,3,26)
      COMMON/ENTHP/ JTND, TEND, TENB
      COMMON/TOC/PRMSHY,ATSFY,HGPU,HGPV,HGPW,ATSF(6),TSN(6),PRMSH(6),
     1 PRMSHI(6), PRMYI, CFTT
      COMMON/PEND/CJX(3,3), DCX(3,6), DCY(3,6), DHD(3,3), DMBX(5,6),
     A DMBY(5,6), LBIX(6), LBIY(6)
      DIMENSION W(3,9,LBX), FA(3,3,3), GA(3,3,3), WA(3),
     A FO(3,3,3), GO(3,3,3)
      KUT = 1
      GO TO (150,150,30,40,50,60),KR
C-----FINE MESH REGION 3
   30 MLL = LPFB
      MUL = LPFC
      MDSW = LBFC + 1
      MDFW = LBFC + 1
      GO TO 100
   32 MLL = LBFK
      MUL = LBFL
      MDSW = LBFL + 1
      MDFW = LBFL + 1
      GO TO 100
C-----OUTER NOSE REGION 4
   40 MLL = LBHB
      MUL = LBHC
      MDSW = LBHC + 1
      MDFW = LBHD + 1
      GO TO 92
   42 MLL = LBHJ
      MUL = LBHL
      MDSW = LBHJ
      MDFW = LBHK
      GO TO 96
C-----INTERMEDIATE NOSE REGION 5
   50 MLL = LBMB
      MUL = LBMD
      MDSW = LBMC + 1
      MDFW = LBMD + 1
      GO TO 92
   52 MLL = LBMJ
      MUL = LBML
      MDSW = LBMJ
      MDFW = LBMK
```

```
      GO TO 96
C-----INNER NOSE REGION 6
   60 MLL = LBTB
      MUL = LBTC
      MDSW = LBTC + 1
      MDFW = LBTD + 1
      GO TO 92
   62 MLL = LBTJ
      MUL = LBTL
      MDSW = LBTJ
      MDFW = LBTK
      GO TO 96
   64 MLL = LBTR
      MUL = LBTS
      MDSW = LBTS + 1
      MDFW = LBTS + 1
      GO TO 100
   92 MODOC = 1
      JDMX = 1
      GO TO 100
   96 MODOC = 2
      JDMX = LBIX(KR) + 1
  100 DO 140 M = MLL,MUL
C-----F,G CONVERSION
      DO 102 K = 1,9
      I = K - 3*((K-1)/3)
      J = 1 + (K - 1)/3
      P = UNOG*W(3,K,M)**GAM
      R = 1.0/W(3,K,M)
      FO(1,I,J) = -W(1,K,M)**2*R - P
      FO(2,I,J) = -W(1,K,M)*W(2,K,M)*R
      FO(3,I,J) = -W(1,K,M)
      GO(2,I,J) = -W(2,K,M)**2*R - P
      GO(1,I,J) = FO(2,I,J)
  102 GO(3,I,J) = - W(2,K,M)
      GO TO (104,105),ITSMD
C-----ALLOWABLE STEP, USE WALL POINT AS CRITICAL
  104 J = 2
      RMU = W(3,J,M) - 1.0
      RA = (1.0+RMU*HGPU*(1.0+RMU*HGPV*(1.0+RMU*HGPW)))*ATSF(KR)
      DTDY = TSN(KR)*W(3,J,M)/(ABS(W(1,J,M))*PRMSH(KR) + ABS(W(2,J,M))
     A + RA)
      DTDX = PRMSH(KR)*DTDY
C-----HALF STEPS,  F,G CONVERSIONS
  105 DO 120 K = 1,9
      GO TO(106,106,106,106,106,106,120,106,120),K
  106 IR = K - 3*((K - 1)/3)
      JR = 1 + (K - 1)/3
      DO 108 KA = 1,3
  108 WA(KA) = W(KA,K,M) + DTDX*(CJX(IR,1)*FO(KA,1,JR)
     A + CJX(IR,2)*FO(KA,2,JR)+ CJX(IR,3)*FO(KA,3,JR))
     B + DTDY*(CJX(JR,1)*GO(KA,IR,1)  + CJX(JR,2)*GO(KA,IR,2)
     C + CJX(JR,3)*GO(KA,IR,3))
      P = UNOG*WA(3)**GAM
      GO TO(110,114,110,110,114,110,120,114,120),K
  110 FA(3,IR,JR) = -WA(1)
      FA(2,IR,JR) = -WA(1)*WA(2)/WA(3)
```

```
      FA(1,IR,JR) = -WA(1)**2/WA(3) - .
      GO TO 120
  114 GA(3,IR,JR) = -  WA(2)
      GA(2,IR,JR) = -WA(2)**2/WA(3) - P
      GA(1,IR,JR) = -WA(1)*WA(2)/WA(3)
  120 CONTINUE
      IF(M - MOSW) 129,121,121
  121 IF(M - MOFW) 122,129,129
  122 GO TO (123,124), MODUC
  123 JDMX = JDMX + 1
      GO TO 125
  124 JDMX = JDMX - 1
  125 DMULT = DFBX(JDMX,KR)
      DUCX = 0.5 - DMULT*(0.5 - DCX(1,KR))
 1252 DUCY = 0.5 - DMULT*(0.5 - DCY(1,KR))
  126 DUXE = DMULT*DCX(2,KR)
      DUYE = DMULT*DCY(2,KR)
      DO 127 KA = 1,3
      WA(KA) = (DUCX+DUCY+DUYE)*W(KA,2,M) + DUXE*(W(KA,1,M)
     A + W(KA,3,M)) + 2.0*DUYE*W(KA,5,M) - DUYE*W(KA,8,M)
     B + DTDX*(FA(KA,3,1) - FA(KA,1,1)) + 2.0*DTDY*(CJX(1,1)*
     C GA(KA,2,1) + CJX(1,2)*GA(KA,2,2) + CJX(1,3)*GA(KA,2,3))
  127 W(KA,5,M) = (DUCX+DUCY)*W(KA,5,M) + DUXE*(W(KA,4,M) + W(KA,6,M))
     A + DUYE*(W(KA,2,M) + W(KA,8,M)) + DTDX*(FA(KA,3,2)
     B - FA(KA,1,2)) +DTDY*(GA(KA,2,3) - GA(KA,2,1))
      GO TO 132
C-----SECOND STEP
  129 DO 130 KA = 1,3
      WA(KA) =   (DCX(1,KR) + DCY(1,KR) + DCY(2,KR))*W(KA,2,M)
     A + DCX(2,KR)*(W(KA,1,M)+W(KA,3,M)) + 2.0*DCY(2,KR)*W(KA,5,M)
     B -DCY(2,KR)*W(KA,8,M) + DTDX*(FA(KA,3,1)-FA(KA,1,1))
     C + 2.0*DTDY*(CJX(1,1)*GA(KA,2,1) + CJX(1,2)*GA(KA,2,2)
     D + CJX(1,3)*GA(KA,2,3))
  130 W(KA,5,M) = (DCX(1,KR)+DCY(1,KR))*W(KA,5,M) + DCX(2,KR)*(W(KA,4,M)
     A +W(KA,6,M)) + DCY(2,KR)*(W(KA,2,M)+W(KA,8,M)) + DTDX*(FA(KA,3,2)
     C -FA(KA,1,2)) + DTDY*(GA(KA,2,3)-GA(KA,2,1))
C-----BOUNDARY CONDITION
  132 UNB = -WA(2)/WA(3)
      UTB = WA(1)/WA(3)
      W(3,2,M) = (WA(3)**HGMU + HGMU*UNB)**DDGMU
      W(1,2,M) = UTB*W(3,2,M)
      W(2,2,M) = 0.0
C-----ENTHALPY COERCION
      IF(JTND.GE.2) CALL ENGS(W(1,2,M))
  140 CONTINUE
      GO TO (142,144,150), KUT
  142 KUT = 2
      GO TO (150,150,32,42,52,62), KR
  144 KUT = 3
      GO TO (150,150,150,150,150,64), KR
  150 RETURN
      END
```

```
      SUBROUTINE JIKAN
C-----DETERMINES TIME STEP ALLOWABLE IN REGION 6
      COMMON/LIMITS/ MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT,
     C MCA, MCB, MCE, MCF, MCG, MCH, MCS, MCT, NCA, NCB, NCE, NCF,
     C NCG, NCH, NCS, NCT,
     F MFA, MFB, MFD, MFE, MFF, MFG, MFH, MFQ, MFS, MFT, MFAN,
     F MFANP, MFAT, MFATM, MFATP, NFA, NFB, NFD, NFE, NFF, NFG, NFH,
     F NFQ, NFS, NFT, NFCL,
     H MHA, MHB, MHD, MHE, MHF, MHG, MHH, MHQ, MHS, MHT, MHAN, MHANP,
     H NHA, NHB, NHD, NHE, NHF, NHG, NHH, NHQ, NHS, NHT, NHCL,
     M MMA, MMB, MMD, MME, MMF, MMG, MMH, MMQ, MMS, MMT, MMAN, MMANP,
     M NMA, NMB, NMD, NME, NMF, NMG, NMH, NMQ, NMS, NMT, NMCL,
     T MTA, MTB, MTD, MTQ, MTS, MTT, MTAN, MTANP, NTA, NTB, NTD,
     T NTQ, NTS, NTT, NTCL,
     S MSL, MST, NSL, NST
      COMMON/BLIMIT/  LBFA, LBFB, LBFC, LBFD, LBFE, LBFI, LBFJ, LBFK,
     F LBFL, LBFM, LBFMAX,
     H LBHA, LBHB, LBHC, LBHD, LBHE, LBHI, LBHJ, LBHK, LBHL, LBHM,
     H LBHMAX,
     M LBMA, LBMB, LBMC, LBMD, LBME, LBMI, LBMJ, LBMK, LBML, LBMM,
     M LBMMAX,
     T LBTA, LBTB, LBTC, LBTD, LBTE, LBTI, LBTJ, LBTK, LBTL, LBTM,
     T LBTQ, LBTR, LBTS, LBTT, LBTMAX
      COMMON/GAS/    GAM, HGMU, UNOG, DOGMU, QCONYI, HINF, GMU, UNOGMU
      COMMON/TOC/PRMSHY,ATSFY,HGPU,HGPV,HGPW,ATSF(6),TSN(6),PRMSH(6),
     1 PRMSHI(6), PRMYI, CFTT
      COMMON/CON1/JR, MR, NR, DTDX, DTDY, F(3,5,26), G(3,5,26),
     1 FP(3,3,3), GP(3,3,3), WAIT(3,3,26)
      COMMON          WY(3,42,9), WC(3,17,17), WF(3,37,13), WH(3,25,17),
     A WM(3,26,26), WT(3,16,21), WSF(3,9,42), WSH(3,9,22), WSM(3,9,14),
     B WST(3,9,43), WSU(3)
      CT = 10.0
      KR = 6
      KUT = 1
      MLL = LBTB
      MUL = LBTC
  100 DO 110 M = MLL,MUL
      DO 110 JC = 2,5,3
      RMU = WST(3,JC,M) - 1.0
      RA = (1.0 + RMU*HGPU*(1.0 + RMU*HGPV*(1.0 + RMU*HGPW)))*ATSF(KR)
      CTE = WST(3,JC,M)/(ABS(WST(1,JC,M))*PRMSH(KR)+ABS(WST(2,JC,M))+RA)
      IF(CTE - CT) 102,110,110
  102 CT = CTE
  110 CONTINUE
      GO TO (112,114,120),KUT
  112 KUT = 2
      MLL = LBTJ
      MUL = LBTL
      GO TO 100
  114 KUT = 3
      MLL = LBTR
      MUL = LBTS
      GO TO 100
  120 DO 130 M = MTB,MTS
      DO 130 N = NTB,NTS
      RMU = WT(3,M,N) - 1.0
```

```
      RA = (1.0 + RMU*HGPU*(1.0 + RMU*HGPV*(1.0 + RMU*HGPW)))*ATSF(KR)
      CTE = WT(3,M,N)/(ABS(WT(1,M,N))*PRMSH(KR)+ABS(WT(2,M,N))+RA)
      IF(CTE - CT) 122,130,130
  122 CT = CTE
  130 CONTINUE
      DTDY = TSN(KR)*CT
      DTDX = PRMSH(KR)*DTDY
      RETURN
      END
```

```
      SUBROUTINE ENGS(W)
C     THIS VERSION HOLDS THE PRESSURE AND CHANGES THE VELOCITY MAGNITUDE
C     IN ORDER TO TEND TOWARD STEADY FREE-STREAM TOTAL ENTHALPY
C     PRESERVES MACH NUMBER IF STATIC ENTHALPY EXCEEDS DESIRED TOTAL
      COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, ODYNI, HINF, GMU, UNOGMU
      COMMON/ENTHP/ JTND, TEND, TENB
      DIMENSION W(3)
      R = 1.0/W(3)
      UX = W(1)*R
      UY = W(2)*R
      E = W(3)**GMU
      ESKI = 0.5*(UX**2 + UY**2)
      HSTAT = 2.5*E
      HTOT = HSTAT + ESKI
      IF (HSTAT - HINF) 5,10,10
    5 UFAC = SQRT(1.0 + TENB*(HINF-HTOT)/ESKI)
      W(1) = UFAC*W(1)
      W(2) = UFAC*W(2)
      RETURN
   10 EFAC = 1.0 - TEND + TEND*HINF/HTOT
      W(3) = (E*EFAC)**UNOGMU
      UFAC = W(3)*SQRT(EFAC)
      W(1) = UFAC*UX
      W(2) = UFAC*UY
      RETURN
      END
```

```
      SUBROUTINE SAKAI
C-----SUBROUTINE INTERCHANGES INFORMATION BETWEEN VARIOUS REGIONS,
C     APPLIES KUTTA CONDITION
C-----JTND = 2, AIRFOIL POINTS FORCED TOWARD STEADY FREE-STREAM TOTAL ENT
C-----SWITCHING CODE FOR IUMT, ETC.
C     1--NO INTERPOLATION,  2--USE 4 POINT SPREAD,
C     3--USE 0 POINT SPREAD WITH ONLY ONE ABOVE, 4--USE 3 POINT SPREAD
C     WITH ONLY ONE BELOW
C
C-----JOFB = 1 FREESTREAM CONDITION MAINTAINED AT INFINITY
C-----JOFB = 2 FREESTREAM CONDITION MAINTAINED OUTSIDE REGION 2
C-----JOFB = 3 FREESTREAM CONDITION MAINTAINED OUTSIDE REGION 3
C
      COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, QDYNI, HINF, GMO, UNOGMU
      COMMON/FLAGS/ IBUG, JBCT, JOFB
      COMMON/ENTHP/ JTND, TEND, TENB
      COMMON/LIMITS/ MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT,
     C MCA, MCB, MCE, MCF, MCG, MCH, MCS, MCT, NCA, NCB, NCE, NCF,
     C NCG, NCH, NCS, NCT,
     F MFA, MFB, MFD, MFE, MFF, MFG, MFH, MFQ, MFS, MFT, MFAN,
     F MFANP, MFAT, MFATM, MFATP, NFA, NFB, NFD, NFE, NFF, NFG, NFH,
     F NFQ, NFS, NFT, NFCL,
     H MHA, MHB, MHD, MHE, MHF, MHG, MHH, MHQ, MHS, MHT, MHAN, MHANP,
     H NHA, NHB, NHD, NHE, NHF, NHG, NHH, NHQ, NHS, NHT, NHCL,
     M MMA, MMB, MMD, MME, MMF, MMG, MMH, MMQ, MMS, MMT, MMAN, MMANP,
     M NMA, NMB, NMD, NME, NMF, NMG, NMH, NMQ, NMS, NMT, NMCL,
     T MTA, MTB, MTD, MTQ, MTS, MTT, MTAN, MTANP, NTA, NTB, NTD,
     T NTQ, NTS, NTT, NTCL,
     S MSL, MST, NSL, NST
      COMMON/BLIMIT/  LBFA, LBFB, LBFC, LBFD, LBFE, LBFI, LBFJ, LBFK,
     F LBFL, LBFM, LBFMAX,
     H LBHA, LBHB, LBHC, LBHD, LBHE, LBHI, LBHJ, LBHK, LBHL, LBHM,
     H LBHMAX,
     M LBMA, LBMB, LBMC, LBMD, LBME, LBMI, LBMJ, LBMK, LBML, LBMM,
     M LBMMAX,
     T LBTA, LBTB, LBTC, LBTD, LBTE, LBTI, LBTJ, LBTK, LBTL, LBTM,
     T LBTQ, LBTR, LBTS, LBTT, LBTMAX
      COMMON/PEND/CJX(3,3), DCX(3,6), DCY(3,6), DHD(3,3), DMBX(5,6),
     A DMBY(5,6), LBIX(6), LBIY(6)
      COMMON/ZMESH/  DX(6), DY(6), RH(9)
      COMMON/ GEOMF/ SY(42), CY(42), DYB(2,42), DCB(2,4,17),
     A XYF(2,42), SCF(2,42), XYH(2,22), SCH(2,22),
     B XYM(2,14), SCM(2,14), XYT(2,43), SCT(2,43)
      COMMON/TERSE/  IYB(2,42), ICB(2,4,17), IDFH(10), IDHM(10),
     A IUMT(10), IWF(3,42), IWH(3,22), IWM(3,14), IWT(3,43),
     B LINTF(2), LINTH(2), LINTM(2), IWFX(2,2), IWHX(2,2), IWMX(2,2),
     C IAF(2,37), IAH(2,25), IAM(2,26), IAT(2,16), IAB(21),
     D IWFH(2,4), IWHM(2,4), IWMT(2,4)
      COMMON/ORDSA/  DWF(9,4,42), DWH(9,4,22), DWM(9,4,14),
     A DWT(9,4,43), AWF(3,42), AWH(3,22), AWM(3,14), AWT(3,43),
     B CWF(9,3,42), CWH(9,3,22), CWM(9,3,14), CWT(9,3,43),
     C DWFX(9,2,6), DWHX(9,2,6), DWMX(9,2,6), DWTX(9,2,6),
     D AWFH(3,3), AWHM(3,3), AWMT(3,3),
     E DWFH(9,2,4), DWHM(9,2,4), DWMT(9,2,4),
     F CWFX(9,2,2), CWHX(9,2,2), CWMX(9,2,2)
      COMMON/TOPL/MLL,MUL,KR,LIP,KUT,LC,JC,MLBR,NLBR,KSPL
```

```
      COMMON       WY(3,42,9), WC(3,17,17), WF(3,37,13), WH(3,25,17),
     A WM(3,26,26), WT(3,16,21), WSF(3,9,42), WSH(3,9,22), WSM(3,9,14),
     B WST(3,9,43), WSU(3)
      DIMENSION GJX(4,4), HJX(4,3), PJX(3,3), QJX(3,6), RJX(4,9),
     A DELW(3), SJX(4,7), TJX(3,7), RHOB(2), UTBB(2)
C
C-----WEIGHTING FACTORS FOR INTERPOLATIONS
C
      DATA GJX/-.075294117,.945882350,.134117650,-.004705882,
     1-.083076923,.729230770,.390769230,-.036923077,-.036923077,
     2.390769230,.729230770,-.083076923,-.004705882,.134117650,
     3.945882350,-.075294117/
C
      DATA HJX/-.084375,.909375,.184375,-.009375,-.0625,.5625,.5625,-.06
     125,-.009375,.184375,.909375,-.084375/
C
      DATA PJX/-.09375,.9375,.15625,-.125,.75,.375,-.09375,.4375,.65625/
C
      DATA QJX/-.08,.96,.12,-.12,.84,.28,-.12,.64,.48,-.08,.36,.72,
     1 0.0,0.0,1.0,.12,-.44,1.32/
      DATA RJX/-.044451219,.988353658,.056646341,-.000548780,-.075294117
     1,.945882350,.134117650,-.004705882,-.088706897,.861120690,.2438793
     210,-.016293103,-.083076923,.729230770,.390769230,-.036923077,-.062
     35,.5625,.5625,-.0625,-.036923077,.390769230,.729230770,-.083076923
     4,-.016293103,.243879310,.861120690,-.088706897,-.004705882,.134117
     5650,.945882350,-.075294117,-.000548780,.056646341,.988353658,-.044
     6451219/
      DATA SJX/-.05359375, .98109375, .07359375, -.00109375,
     A -.084375, .909375, .184375, -.009375,  -.08616728, .76631434,
     B .35087316, -.03102022, -.0625, .5625, .5625, -.0625,
     C -.03102022, .35087316, .76631434, -.08616728,  -.009375,
     D .184375, .909375, -.084375,  -.00109375, .07359375, .98109375,
     E -.05359375/
      DATA TJX/-.0546875, .984375, .0703125,  -.09375, .9375, .15625,
     A -.1171875, .859375, .2578125,  -.125, .75, .375,  -.1171875,
     B .609375, .5078125,  -.09375, .4375, .65625,  -.0546875,
     C .234375, .8203125/
C-----YTRE BOUNDARY POINTS FROM COARSE MESH
  560 DO 562 M = MYB,MYS
      MA = IYB(1,M)
      NA = IYB(2,M)
      AX = DYB(1,M)
      AY = DYB(2,M)
      CCP = 0.125*AX*AY
      CZP = 1.0 - AX**2 - AY**2
      CXP = 0.5*AX*(AX + 1.0)
      CXM = CXP - AX
      CYP = 0.5*AY*(AY + 1.0)
      CYM = CYP - AY
      DO 562 K = 1,3
  562 WY(K,M,1) = WC(K,MA,NA)*CZP + WC(K,MA+1,NA)*CXP + WC(K,MA-1,NA)
     1 *CXM + WC(K,MA,NA+1)*CYP + WC(K,MA,NA-1)*CYM + CCP*(WC(K,MA+1,
     2 NA+1) + WC(K,MA-1,NA-1) - WC(K,MA+1,NA-1) - WC(K,MA-1,NA+1))
      DO 564 N = 1,NYS
      DO 564 K = 1,3
      WY(K,MYA,N) = WY(K,MYS,N)
  564 WY(K,MYT,N) = WY(K,MYB,N)
```

```
C-----COARSE MESH BOUNDARY POINTS FROM TIRE
C-----LEFT COLUMN
      J = 1
      M = MCA
      N = 0
      MD = 0
      ND = 1
      LTP = NCT
      LBT = 1
C-----COLUMN AND ROW ROUTINE
  570 DO 572 LA = LBT,LTP
      M = M + MD
      N = N + ND
      MA = ICB(1,J,LA)
      NA = ICB(2,J,LA)
      AX = DCR(1,J,LA)
      AY = DCR(2,J,LA)
      CCP = 0.125*AX*AY
      C2P = 1.0 - AX**2 - AY**2
      CXP = 0.5*AX*(AX + 1.0)
      CXM = CXP - AX
      CYP = 0.5*AY*(AY + 1.0)
      CYM = CYP - AY
      DO 572 K = 1,3
  572 WC(K,M,N) = WY(K,MA,NA)*C2P + WY(K,MA+1,NA)*CXP + WY(K,MA-1,NA)
     1 *CXM + WY(K,MA,NA+1)*CYP + WY(K,MA,NA-1)*CYM + CCP*(WY(K,MA+1,
     2 NA+1) + WY(K,MA-1,NA-1) - WY(K,MA+1,NA-1) - WY(K,MA-1,NA+1))
      GO TO (574,576,578,320),J
C-----RIGHT COLUMN
  574 M = MCT
      N = 0
      J = 2
      GO TO 570
C-----BOTTOM ROW
  576 LTP = MCS
      LBT = MCB
      ND = 0
      MD = 1
      M = 1
      N = 1
      J = 3
      GO TO 570
C-----TOP ROW
  578 M = MCA
      N = NCT
      J = 4
      GO TO 570
C-----INNER NOSE REGION---INTERMEDIATE NOSE REGION INTERCHANGE
C-----ASSIGNMENT TO INTERMEDIATE NOSE REGION
  320 M = MME
      MM = MYA
      DO 324 JM = MMF,MMG
      M = M + 1
      MM = MM + 1
      N = NME
      NN = NYA
      DO 324 JN = NMF,NMG
```

```
      N = N + 1
      NN = NN + 2
      DO 324 K = 1,3
  324 WM(K,M,N) = WT(K,MM,NN)
C-----OUTER NOSE REGION -- INTERMEDIATE NOSE REGION INTERCHANGE
C-----ASSIGNMENT TO OUTER NOSE REGION
  580 M = MHE
      MM = MMA
      DO 582 JM = MHF,MHG
      M = M + 1
      MM = MM + 5
      N = NHE
      NN = NMA
      DO 582 JN = NHF,NHG
      N = N + 1
      NN = NN + 5
      DO 582 K = 1,3
  582 WH(K,M,N) = WM(K,MM,NN)
C-----FINE MESH -- OUTER NOSE REGION INTERCHANGE
C-----ASSIGNMENT TO FINE MESH REGION
  600 M = MFE
      MM = MHA
      DO 612 JM = MFF,MFG
      M = M + 1
      MM = MM + 4
      N = NFE
      NN = NHA
      NLL = NFF
      NUL = NFG
  610 DO 612 JN = NLL,NUL
      N = N + 1
      NN = NN + 4
      DO 612 K = 1,3
  612 WF(K,M,N) = WH(K,MM,NN)
C-----COARSE REGION --- FINE REGION INTERCHANGE
C-----ASSIGNMENT TO COARSE MESH
  690 M = MCE
      MM = MFA
      DO 700 JM = MCF,MCG
      M =  M + 1
      MM = MM + 4
      N = NCE
      NN = NFA
      JNL = 1
      JNU = NCG - NCE
  692 DO 700 JN = JNL,JNU
      N = N + 1
      NN = NN  + 2
      DO 700 K = 1,3
  700 WC(K,M,N) = WF(K,MM,NN)
C-----EXCHANGES OF INFORMATION BETWEEN MAIN CARTESIAN MESHES AND BOUNDARY
C-----LOCAL SURFACE NORMALS
C-----INNER NOSE REGION 6
      KR = 6
      LIP = 2
      JC = 8
      KUT = 1
```

```
      LC = 1 - MTAN
      MLL = LBTB
      MUL = LBTC
      MLBR = MME
      NLBR = NME
 1010 CALL KAKOI(WT,MTT,NTT,WM,MMT,NMT,WST,IWT,CWT,AWT,SCT,LBTMAX,
     A IAT,MTT)
      GO TO (1012,1014),KUT
 1012 KUT = 2
      LC = 2*(MTT+1) - MTAN
      MLL = LBTJ
      MUL = LBTL
      GO TO 1010
 1014 KUT = 3
      LC = 2*(MTT - MTAN + 1)
      MLL = LPTR
      MUL = LPTS
      CALL KAKOI(WT,MTT,NTT,WM,MMT,NMT,WST,IWT,CWT,AWT,SCT,LBTMAX,
     A IAN,NTT)
      KR = 5
      KUT = 1
      LC = 1 - MMG
      MLL = LBMB
      MUL = LBMD
      MLBR = MHE
      NLBR = NHE
 1020 CALL KAKOI(WM,MMT,NMT,WH,MHT,NHT,WSM,IWM,CWM,AWM,SCM,LBMMAX,
     A IAM,MMT)
      GO TO (1022,1026),KUT
 1022 KUT = 2
      LC = 2*(MMT+1) - MMG
      MLL = LBMJ
      MUL = LBML
      GO TO 1020
 1026 KR = 4
      KUT = 1
      LC = 1 - MHG
      MLL = LBHB
      MUL = LBHD
      MLBR = MFE
      NLBR = NFE
 1030 CALL KAKOI(WH,MHT,NHT,WF,MFT,NFT,WSH,IWH,CWH,AWH,SCH,LBHMAX,
     A IAH,MHT)
      GO TO (1032,1036),KUT
 1032 KUT = 2
      LC = 2*(MHT+1) - MHG
      MLL = LPHJ
      MUL = LBHL
      GO TO 1030
 1036 KR = 3
      KUT = 1
      LC = 1 - MFG
      MLL = LBFB
      MUL = LBFC
      MLBR = MCE
      NLBR = NCE
 1040 CALL KAKOI(WF,MFT,NFT,WC,MCT,NCT,WSF,IWF,CWF,AWF,SCF,LBFMAX,
```

```
     A IAF,MFT)
      GO TO (1042,1046),KUT
 1042 KUT = 2
      LC = 2*(MFAT+3) - MFG
      MLL = LBFK
      MUL = LPFL
      GO TO 1040
 1046 CONTINUE
C-----SPECIAL ASSIGNMENTS NEAR TRAILING EDGE
      DO 1160 KA = 1,3
      WSF(KA,7,LBFD) = WF(KA,MFAT,NFCL+2)
      DO 1160 JA = 1,3
      WSF(KA,-1+3*JA,LBFD) = WF(KA,MFATP,NFCL-1+JA)
      WSF(KA,3*JA,LBFD) = WF(KA,MFATP+1,NFCL-1+JA)
      DO 1160 LA = 1,3
 1160 WSF(KA,-3+LA+3*JA,LBFE) = WF(KA,MFAT+LA,NFCL-1+JA)
      RFC = -1.0
      DO 1162 KA = 1,3
      IF(KA.EQ.3) RFC = 1.0
      WSF(KA,9,LBFJ) = RFC*WF(KA,MFAT,NFCL-2)
      DO 1162 JA = 1,3
      WSF(KA,-1+3*JA,LBFJ) = RFC*WF(KA,MFATP,NFCL+1-JA)
      WSF(KA,-2+3*JA,LBFJ) = RFC*WF(KA,MFATP+1,NFCL+1-JA)
      DO 1162 LA = 1,3
 1162 WSF(KA,-3+LA+3*JA,LBFI) = RFC*WF(KA,MFAT+4-LA,NFCL+1-JA)
C-----SPECIAL ASSIGNMENTS AT OVERLAPS
C-----INNER NOSE REGION 6
      KUT = 1
      LU = LBTT
      LK = LBTB
 1166 DO 1168 KA = 1,3
      DO 1168 JC = 2,8,3
 1168 WST(KA,JC,LU) = WST(KA,JC,LK)
      GO TO (1170,1172,1174,1178),KUT
 1170 KUT = 2
      LU = LBTA
      LK = LBTS
      GO TO 1166
 1172 KUT = 3
      LU = LBTG
      LK = LBTL
      GO TO 1166
 1174 KUT = 4
      LU = LBTM
      LK = LBTR
      GO TO 1166
C-----REGION ABUTMENTS (CODE D) POINTS
 1178 DO 1180 KA = 1,3
      WSM(KA,2,LBMA) = WST(KA,2,LBTC)
      WSM(KA,2,LBMM) = WST(KA,2,LBTK)
      WSH(KA,2,LBHA) = WSM(KA,2,LBMC)
      WSH(KA,2,LBHM) = WSM(KA,2,LBMK)
      WSF(KA,2,LBFA) = WSH(KA,2,LBHC)
 1180 WSF(KA,2,LBFM) = WSH(KA,2,LBHK)
C-----REGION ABUTMENTS, (CODE E) POINTS
      KUT = 1
      LU = LBTE
```

```
      LK = LBMB
 1182 DO 1184 KA = 1,3
      DO 1184 LIP = 1,3
      JC = -1 + 3*LIP
      WST(KA,JC,LU) = 0.0
      DO 1184 JL = 1,3
      LUP = -1 + 3*JL
 1184 WST(KA,JC,LU) = WST(KA,JC,LU) + AWMT(JL,LIP)*WSM(KA,LUP,LK)
      GO TO (1186,1188), KUT
 1186 KUT = 2
      LU = LBTI
      LK = LBML
      GO TO 1182
 1188 KUT = 1
      LU = LBME
      LK = LBHB
 1190 DO 1192 KA = 1,3
      DO 1192 LIP = 1,3
      JC = -1 + 3*LIP
      WSM(KA,JC,LU) = 0.0
      DO 1192 JL = 1,3
      LUP = -1 + 3*JL
 1192 WSM(KA,JC,LU) = WSM(KA,JC,LU) + AWHM(JL,LIP)*WSH(KA,LUP,LK)
      GO TO (1194,1198), KUT
 1194 KUT = 2
      LU = LBMI
      LK = LBHL
      GO TO 1190
 1198 KUT = 1
      LU = LBHE
      LK = LBFB
 1200 DO 1202 KA = 1,3
      DO 1202 LIP = 1,3
      JC = -1 + 3*LIP
      WSH(KA,JC,LU) = 0.0
      DO 1202 JL = 1,3
      LUP = -1 + 3*JL
 1202 WSH(KA,JC,LU) = WSH(KA,JC,LU) + AWFH(JL,LIP)*WSF(KA,LUP,LK)
      GO TO (1204, 329), KUT
 1204 KUT = 2
      LU = LBHI
      LK = LBFL
      GO TO 1200
C-----INTERPOLATIONS AT CARTESIAN BOUNDARY FROM COARSER TO FINER MESH
C-----INTERPOLATION INTO INNER NOSE REGION
C-----LEFT COLUMN
  329 KTS = 1
      M = MME
      MM = MTA
      N = NME - 1
      NN = NTA - 2
      GO TO 340
  330 WT(K,MM,NN+1) = HJX(1,2)*WM(K,M,N-1) + HJX(2,2)*WM(K,M,N)
     A + HJX(3,2)*WM(K,M,N+1) + HJX(4,2)*WM(K,M,N+2)
      GO TO (342,348), KTS
  334 WT(K,MM,NN+1) = PJX(1,2)*WM(K,M,N-1) + PJX(2,2)*WM(K,M,N)
     A + PJX(3,2)*WM(K,M,N+1)
```

```
      GO TO 348
  336 WT(K,MM,NN+1) = PJX(1,2)*WM(K,M,N+2) + PJX(2,2)*WM(K,M,N+1)
     A + PJX(3,2)*WM(K,M,N)
      GO TO 348
  340 JNL = 1
      JNU = NMH - NME
      DO 342 JN = JNL,JNU
      N = N + 1
      NN = NN + 2
      DO 342 K = 1,3
      WT(K,MM,NN) = WM(K,M,N)
      GO TO 330
  342 CONTINUE
      DO 344 K = 1,3
  344 WT(K,MM,NTT) = WM(K,M,NMH)
C----RIGHT COLUMN
      KTS = 2
      M = MMH
      MM = MTT
      N = NME - 1
      NN = NTA - 2
      DO 348 JN = JNL,JNU
      N = N + 1
      NN = NN + 2
      ISWCH = ICMT(JN)
      DO 348 K = 1,3
      WT(K,MM,NN) = WM(K,M,N)
      GO TO (348,330,334,338), ISWCH
  348 CONTINUE
      DO 350 K = 1,3
  350 WT(K,MM,NTT) = WM(K,M,NMH)
C-----BOTTOM ROW
      KTS = 1
      N = NME
      NN = NTA
  352 M = MME
      MM = MTA
      DO 354 JM = MMF,MMG
      M = M + 1
      MM = MM + 1
      DO 354 K = 1,3
  354 WT(K,MM,NN) = WM(K,M,N)
      GO TO (356,589), KTS
C-----TOP ROW
  356 KTS = 2
      N = NMH
      NN = NTT
      GO TO 352
C-----INTERPOLATION INTO INTERMEDIATE NOSE REGION
C-----LEFT COLUMN
  589 KTS = 1
      M = MHE
      MM = MMA
      N = NHE - 1
      NN = NMA - 5
      GO TO 600
  590 DO 592 JL = 1,4
```

```
  592 WM(K,MM,NN+JL) = GJX(1,JL)*WH(K,M,N-1) + GJX(2,JL)*WH(K,M,N)
     A + GJX(3,JL)*WH(K,M,N+1) + GJX(4,JL)*WH(K,M,N+2)
      GO TO (602,604),KTS
  593 DO 594 JL = 1,4
  594 WM(K,MM,NN+5-JL) = GJX(1,JL)*WH(K,M,N+2) + GJX(2,JL)*WH(K,M,N+1)
     A + GJX(3,JL)*WH(K,M,N)
      GO TO 604
  595 DO 596 JL = 1,4
  596 WM(K,MM,NN+JL) = GJX(1,JL)*WH(K,M,N-1) + GJX(2,JL)*WH(K,M,N)
     A + GJX(3,JL)*WH(K,M,N+1)
      GO TO 604
  600 JNL = 1
      JNU = NHH - NHE
      DO 602 JN = JNL,JNU
      N = N + 1
      NN = NN + 5
      DO 602 K = 1,3
      WM(K,MM,NN) = WH(K,M,N)
      GO TO 590
  602 CONTINUE
      DO 603 K = 1,3
  603 WM(K,MM,NMT) = WH(K,M,NHH)
C----RIGHT COLUMN
      KTS = 2
      M = MHH
      MM = MMT
      N = NHE - 1
      NN = NMA - 5
      DO 604 JN = JNL,JNU
      N = N + 1
      NN = NN + 5
      ISWCH = ICHM(JN)
      DO 604 K = 1,3
      WM(K,MM,NN) = WH(K,M,N)
      GO TO (604,590,595,593),ISWCH
  604 CONTINUE
      DO 6040 K = 1,3
 6040 WM(K,MM,NMT) = WH(K,M,NHH)
C-----BOTTOM ROW
      N = NHE
      NN = NMA
      KTS = 1
      JML = 1
      JMU = MHH - MHE
  605 M = MHE - 1
      MM = MMA - 5
      DO 608 JM = JML,JMU
      M = M + 1
      MM = MM + 5
      DO 606 K = 1,3
      WM(K,MM,NN) = WH(K,M,N)
      DO 606 JL = 1,4
  606 WM(K,MM+JL,NN) = GJX(1,JL)*WH(K,M-1,N) + GJX(2,JL)*WH(K,M,N)
     A + GJX(3,JL)*WH(K,M+1,N) + GJX(4,JL)*WH(K,M+2,N)
  608 CONTINUE
      GO TO (6080,629), KTS
C-----TOP ROW
```

```
 6080 KTS = 2
      N = NHH
      NN = NMT
      GO TO 605
C-----INTERPOLATION INTO OUTER NOSE REGION
C-----LEFT COLUMN
  629 KTS = 1
      M = MFE
      MM = MHA
      N = NFE - 1
      NN = NHA - 4
      JNL = NFE
      JNU = NFG
      GO TO 640
  630 DO 632 JL = 1,3
  632 WH(K,MM,NN+JL) = HJX(1,JL)*WF(K,M,N-1) + HJX(2,JL)*WF(K,M,N)
     A + HJX(3,JL)*WF(K,M,N+1) + HJX(4,JL)*WF(K,M,N+2)
      GO TO (642,644),KTS
  633 DO 634 JL = 1,3
  634 WH(K,MM,NN+4 JL) = PJX(1,JL)*WF(K,M,N+2) + PJX(2,JL)*WF(K,M,N+1)
     A + PJX(3,JL)*WF(K,M,N)
      GO TO 644
  635 DO 636 JL = 1,3
  636 WH(K,MM,NN+JL) = PJX(1,JL)*WF(K,M,N-1) + PJX(2,JL)*WF(K,M,N)
     A + PJX(3,JL)*WF(K,M,N+1)
      WH(K,MM,NN+4) = WF(K,M,N+1)
      GO TO 644
  640 DO 642 JN = JNL,JNU
      N = N + 1
      NN = NN + 4
      DO 642 K = 1,3
      WH(K,MM,NN) = WF(K,M,N)
      GO TO 630
  642 CONTINUE
 6430 DO 6431 K = 1,3
 6431 WH(K,MM,NHT) = WF(K,M,NFH)
C----RIGHT COLUMN
      KTS = 2
      MM = MHT
      M = MFH
      N = NFE - 1
      NN = NHA - 4
      JNL = 1
      JNU = NFH - NFE
 6432 DO 644 JN = JNL,JNU
      N = N + 1
      NN = NN + 4
      ISWCH = IDFH(JN)
      DO 644 K = 1,3
      WH(K,MM,NN) = WF(K,M,N)
 6434 GO TO (644,630,635,633),ISWCH
  644 CONTINUE
 6444 N = NFE
      NN = NHA
      KTS = 1
      JML = 1
      JMU = MFH - MFE
```

```
 6445 M = MFE
      MM = MHA
      DO 645 K = 1,3
  645 WH(K,MM,NN) = WF(K,M,N)
      DO 648 JM = JML,JMU
      DO 646 K = 1,3
      DO 646 JL = 1,3
  646 WH(K,MM+JL,NN) = HJX(1,JL)*WF(K,M-1,N) + HJX(2,JL)*WF(K,M,N)
     A + HJX(3,JL)*WF(K,M+1,N) + HJX(4,JL)*WF(K,M+2,N)
      M = M + 1
      MM = MM + 4
      DO 648 K = 1,3
  648 WH(K,MM,NN) = WF(K,M,N)
      GO TO (6446,709), KTS
 6446 KTS = 2
      N = NFH
      NN = NHT
      GO TO 6445
C-----INTERPOLATION INTO FINE MESH REGION
C-----LEFT COLUMN
  709 KTS = 1
      M = MCE
      MM = MFA
  710 N = NCE - 1
      NN = NFA - 2
      JNL = 1
      JNU = NCH - NCE
  712 DO 714 JN = JNL,JNU
      N = N + 1
      NN = NN + 2
      DO 714 K = 1,3
      WF(K,MM,NN + 1) = HJX(1,2) *WC(K,M,N-1) + HJX(2,2)*WC(K,M,N)
     * + HJX(3,2)*WC(K,M,N+1) + HJX(4,2)*WC(K,M,N+2)
  714 WF(K,MM,NN) = WC(K,M,N)
 7150 DO 7152 K = 1,3
 7152 WF(K,MM,NFT) = WC(K,M,NCH)
      GO TO (716,720),KTS
C----RIGHT COLUMN
  716 KTS = 2
      M = MCH
      MM = MFT
      GO TO 710
C-----BOTTOM ROW
  720 KTS = 1
      N = NCE
      NN = NFA
  722 M = MCE - 1
      MM = MFA - 4
      DO 726 JM=MCE,MCG
      M = M + 1
      MM = MM + 4
      DO 726 K = 1,3
      WF(K,MM,NN) = WC(K,M,N)
      DO 726 JL = 1,3
  726 WF(K,MM+JL,NN) = HJX(1,JL)*WC(K,M-1,N) + HJX(2,JL)*WC(K,M,N)
     A + HJX(3,JL)*WC(K,M+1,N) + HJX(4,JL)*WC(K,M+2,N)
      DO 728 K = 1,3
```

```
  728 WF(K,MFT,NN) = WC(K,MCH,N)
      GO TO (729,740),KTS
  729 KTS = 2
      N = NCH
      NN = NFT
      GO TO 722
  740 CONTINUE
C-----REGION ABUTMENTS, OVERHANGING (CODE A) POINTS OBTAINED BY INTERPOLA
C     FROM CARTESIAN MESH
      KUT = 1
      LU = LBMA
 1208 CALL KAKARU(KUT,LU,IWMX,CWMX,WM,MMT,NMT,WSM,SCM,LBMMAX)
      GO TO (1210,1214), KUT
 1210 KUT = 2
      LU = LBMM
      GO TO 1208
 1214 KUT = 1
      LU = LBHA
 1216 CALL KAKARU(KUT,LU,IWHX,CWHX,WH,MHT,NHT,WSH,SCH,LBHMAX)
      GO TO (1218,1220), KUT
 1218 KUT = 2
      LU = LBHM
      GO TO 1216
 1220 KUT = 1
      LU = LBFA
 1222 CALL KAKARU(KUT,LU,IWFX,CWFX,WF,MFT,NFT,WSF,SCF,LBFMAX)
      GO TO (1224,1240), KUT
 1224 KUT = 2
      LU = LBFM
      GO TO 1222
C-----OBTAINING AUXILIARIES FOR LOCAL ROTATED MESHES BY INTERPOLATION FRO
C     CARTESIAN MESHES AND DISTORTED MESHES
 1240 KR = 6
      KUT = 1
      MLL = LPTF
      MUL = LBTC
      KSPL = 0
 1242 CALL WAKU(IWT,CWT,WST,SCT,DWT,DWTX,LBTMAX,WT,MTT,NTT,
     A WT,MTT,NTT,DWMT,IWMT,LINTM)
      GO TO (1244,1246,1250), KUT
 1244 KUT = 2
      MLL = LPTJ
      MUL = LPTL
      GO TO 1242
 1246 KUT = 3
      MLL = LPTR
      MUL = LPTS
      GO TO 1242
 1250 KR = 5
      KUT = 1
      MLL = LPMB
      MUL = LPMC
      KSPL = 0
 1252 CALL WAKU(IWM,CWM,WSM,SCM,DWM,DWMX,LBMMAX,WM,MMT,NMT,
     A WT,MTT,NTT,DWMT,IWMT,LINTM)
      GO TO (1254,1260), KUT
 1254 KUT = 2
```

```
      MLL = LBMJ
      MUL = LBML
      GO TO 1252
 1260 KR = 4
      KUT = 1
      MLL = LBHB
      MUL = LBMC
      KSPL = 0
 1262 CALL WAKU(IWH,CWH,WSH,SCH,DWH,DWHX,LBHMAX,WH,MHT,NHT,
     A WM,MMT,NMT,DWHM,IWHM,LINTH)
      GO TO (1264,1270), KUT
 1264 KUT = 2
      MLL = LBHJ
      MUL = LBML
      GO TO 1262
 1270 KR = 3
      KUT = 1
      MLL = LPFB
      MUL = LPFD
      KSPL = 0
 1272 CALL WAKU(IWF,CWF,WSF,SCF,DWF,DWFX,LBFMAX,WF,MFT,NFT,
     A WH,MHT,NHT,DWFH,IWFH,LINTF)
      GO TO (1274,1280), KUT
 1274 KUT = 2
      MLL = LPFJ
      MUL = LPFL
      GO TO 1272
 1280 CONTINUE
C-----KUTTA CONDITION
 1428 WF(3,MFAT,NFCL) = 0.5*(WSF(3,1,LBFD) + WSF(3,3,LBFJ))
      DO 1430 KA = 1,2
 1430 WF(KA,MFAT,NFCL) = 0.5*(WSF(KA,1,LBFD) - WSF(KA,3,LBFJ))
      RETURN
      END
```

```
      SUBROUTINE KAKOI(WT,MTT,NTT,WM,MMT,NMT,WST,IWT,CWT,AWT,SCT,
     A LBX,IAT,MNX)
C-----INTERPOLATES FROM CARTESIAN ONTO TOP (3RRD) POINT OF LOCAL SURFACE
C     NORMAL AND FROM LOCAL SURFACE NORMAL TO CARTESIAN POINT NEAREST SUR
      COMMON/TOPL/MLL,MUL,KP,LIP,KUT,LC,JC,MLBR,NLBR,KSPL
      COMMON/PLND/CJX(3,3), DCX(3,6), DCY(3,6), DHD(3,3), DMBX(5,6),
     A DMBY(5,6), LBIX(6), LBIY(6)
      DIMENSION WT(3,MTT,NTT), WM(3,MMT,NMT), WST(3,9,LBX),
     A IWT(3,LBX), CWT(9,3,LBX), AWT(3,LBX), SCT(2,LBX), IAT(2,MNX),
     B WONT(3)
      DO 20 M = MLL,MUL
C-----DETERMINE CENTER OF CARTESIAN REFERENCE ARRAY
      MK = IWT(LIP,M)/100
      NK = IWT(LIP,M) - 100*MK - 2
      MK = MK - 2
C-----INTERPOLATE FROM CARTESIAN MESH TO TOP POINT OF LOCAL SURFACE NORMA
      DO 4 KA = 1,3
      WONT(KA) = 0.0
      DO 4 LW = 1,9
      JB = (LW+2)/3
      JA = LW - 3*(JB-1)
    4 WONT(KA) = WONT(KA) + CWT(LW,LIP,M)*WT(KA,MK+JA,NK+JB)
      WST(3,JC,M) = WONT(3)
      WST(1,JC,M) = WONT(1)*SCT(2,M) + WONT(2)*SCT(1,M)
      WST(2,JC,M) = -WONT(1)*SCT(1,M) + WONT(2)*SCT(2,M)
      GO TO (6,8,10), KUT
    6 MK = M - LC
      NK = IAT(2,MK)
      GO TO 12
    8 MK = LC - M
      NK = IAT(1,MK)
      GO TO 12
   10 NK = M - LC
      MK = IAT(NK)
C-----INTERPOLATE FROM SURFACE NORMAL TO CARTESIAN POINT NEAREST THE SURF
   12 DO 14 KA = 1,3
   14 WONT(KA) = AWT(1,M)*WST(KA,2,M) + AWT(2,M)*WST(KA,5,M)
     A + AWT(3,M)*WST(KA,8,M)
      WT(3,MK,NK) = WONT(3)
      WT(1,MK,NK) = WONT(1)*SCT(2,M) - WONT(2)*SCT(1,M)
      WT(2,MK,NK) = WONT(1)*SCT(1,M) + WONT(2)*SCT(2,M)
      ICOIN = MK - 1 - LBIX(KR)*((MK-1)/LBIX(KR))
      IF(ICOIN) 20,16,20
   16 JCOIN = NK - 1 - LBIY(KR)*((NK - 1)/LBIY(KR))
      IF(JCOIN) 20,17,20
   17 ML = (MK-1)/LBIX(KR) + MLBR
      NL = (NK-1)/LBIY(KR) + NLBR
      DO 18 KA = 1,3
   18 WM(KA,ML,NL) = WT(KA,MK,NK)
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE WAKO(IWT,CWT,WST,SCT,DWT,DWTX,LBX,WT,MTT,NTT,
     A WU,MUT,NUT,DWTU,IWTU,LINTU)
C-----INTERPOLATES FROM CARTESIAN TO POINTS 7,9 OF LOCAL ROTATED MESH,
C     FROM DISTORTED MESH ONTO POINTS 3,6 OF ADJACENT (TO LEFT) MESH AND
C     POINTS 1,4 OF ADJACENT (TO RIGHT) MESH.  ALSO PICKS UP POINTS 1,4
C     OR 3,6 FOR MESHES AT ENDS OF RANGE AND PUTS IN BORDER CARTESIAN POI
C     FOR FINER MESH (CODE F)
      COMMON/TOPL/MLL,MUL,KR,LIP,KUT,LC,JC,MLBR,NLBR,KSPL
      DIMENSION IWT(3,LBX), CWT(9,3,LBX), WST(3,9,LBX), SCT(2,LBX),
     A DWT(9,4,LBX), DWTX(9,2,6), WT(3,MTT,NTT), WU(3,MUT,NUT),
     B DWTU(9,2,4), IWTU(2,4), LINTU(2), WONT(3), WINT(3,9)
      DO 50 MM = MLL,MUL
C-----UPPER ROW OBTAINED FROM CARTESIAN MESH
      DO 16 LIP = 1,3,2
      JC = 6 + LIP
      MK = IWT(LIP,MM)/100
      NK = IWT(LIP,MM) - 100*MK - 2
      MK = MK - 2
      DO 14 KA = 1,3
      WONT(KA) = 0.0
      DO 14 LW = 1,9
      JB = (LW+2)/3
      JA = LW - 3*(JB-1)
   14 WONT(KA) = WONT(KA) + CWT(LW,LIP,MM)*WT(KA,MK+JA,NK+JB)
      WST(3,JC,MM) = WONT(3)
      WST(1,JC,MM) = WONT(1)*SCT(2,MM) + WONT(2)*SCT(1,MM)
   16 WST(2,JC,MM) = -WONT(1)*SCT(1,MM) + WONT(2)*SCT(2,MM)
C-----UNIFORM W CONTENT FOR DISTORTED MESH
      DO 20 JA = 1,3
      MU = MM - 2 + JA
      DO 20 JB = 1,3
      JC = 3*(JB-1) + JA
      JU = -1 + 3*JB
      WINT(3,JC) = WST(3,JU,MU)
      WINT(1,JC) = WST(1,JU,MU)*SCT(2,MU) - WST(2,JU,MU)*SCT(1,MU)
   20 WINT(2,JC) = WST(1,JU,MU)*SCT(1,MU) + WST(2,JU,MU)*SCT(2,MU)
C-----OUTER COLUMNS OF ADJACENT ROTATED MESHES OBTAINED FROM DISTORTED ME
      DO 24 LIP = 1,4
      MU = MM - 3 + 2*((LIP+1)/2)
      JC = 8 - 2*((LIP+1)/2) - 3*(LIP-2*(LIP/2))
      DO 22 KA = 1,3
      WONT(KA) = 0.0
      DO 22 LW = 1,9
   22 WONT(KA) = WONT(KA) + DWT(LW,LIP,MM)*WINT(KA,LW)
      WST(3,JC,MU) = WONT(3)
      WST(1,JC,MU) = WONT(1)*SCT(2,MU) + WONT(2)*SCT(1,MU)
   24 WST(2,JC,MU) = -WONT(1)*SCT(1,MU) + WONT(2)*SCT(2,MU)
C-----AUXILIARIES AT ENDS OF OPERATING RANGES
      IF(MM-MLL) 28,28,26
   26 IF(MM-MUL) 50,30,30
   28 JU = -2
      GO TO 32
   30 JU = 0
   32 KSPL = KSPL + 1
      MU = MM
      DO 36 LIP = 1,2
```

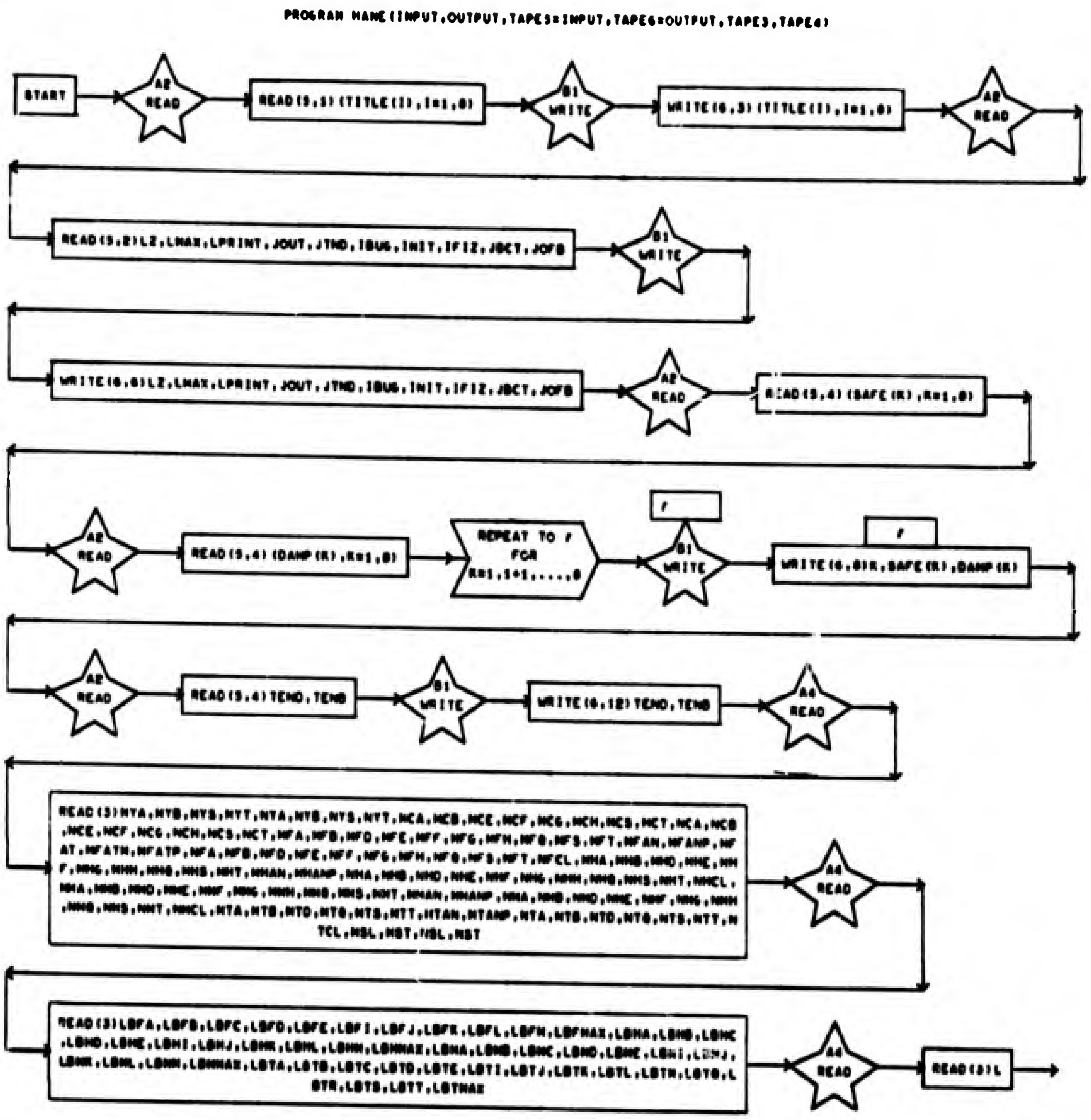
```
      JC = JU + 3*LIP
      DO 34 KA = 1,3
      WONT(KA) = 0.0
      DO 34 LW = 1,9
   34 WONT(KA) = WONT(KA) + DWTX(LW,LIP,KSPL)*WINT(KA,LW)
      WST(3,JC,MU) = WONT(3)
      WST(1,JC,MU) = WONT(1)*SCT(2,MU) + WONT(2)*SCT(1,MU)
   36 WST(2,JC,MU) = -WONT(1)*SCT(1,MU) + WONT(2)*SCT(2,MU)
   40 IF(KK-6) 42,50,50
C-----FINER CARTESIAN MESH AUXILIARIES ALONG BORDER (CODE F)
   42 GO TO (46,50,50,46,50,50),KSPL
   46 LUP = LINTU(KUT)
      MU = IWTU(KUT,1)/100
      DO 48 LIP = 1,LUP
      NU = IWTU(KUT,LIP) - 100*MU
      DO 48 KA = 1,3
      WU(KA,MU,NU) = 0.0
      DO 48 LW = 1,9
   48 WU(KA,MU,NU) = WU(KA,MU,NU) + DWTU(LW,KUT,LIP)*WINT(KA,LW)
   50 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE KAKARU(KUT,LU,IWMX,CWMX,WM,MMT,NMT,WSM,SCM,LBX)
C-----INTERPOLATES FROM CARTESIAN MESH TO TOP 2 POINTS OF SURFACE NORMAL
C     EDGE OF RANGE (CODE A)
      DIMENSION IWMX(2,2), CWMX(9,2,2), WM(3,MMT,NMT), WSM(3,9,LBX),
     A SCM(2,LBX), WONT(3)
      DO 16 LIP = 1,2
      LUP = 2 + 3*LIP
      MK = IWMX(LIP,KUT)/100
      NK = IWMX(LIP,KUT) - 100*MK - 2
      MK = MK - 2
      DO 14 KA = 1,3
      WONT(KA) = 0.0
      DO 14 LW = 1,9
      JB = (LW+2)/3
      JA = LW - 3*(JB-1)
   14 WONT(KA) = WONT(KA) + CWMX(LW,LIP,KUT)*WM(KA,MK+JA,NK+JB)
      WSM(3,LUP,LU) = WONT(3)
      WSM(1,LUP,LU) = WONT(1)*SCM(2,LU) + WONT(2)*SCM(1,LU)
   16 WSM(2,LUP,LU) = -WONT(1)*SCM(1,LU) + WONT(2)*SCM(2,LU)
      RETURN
      END
```

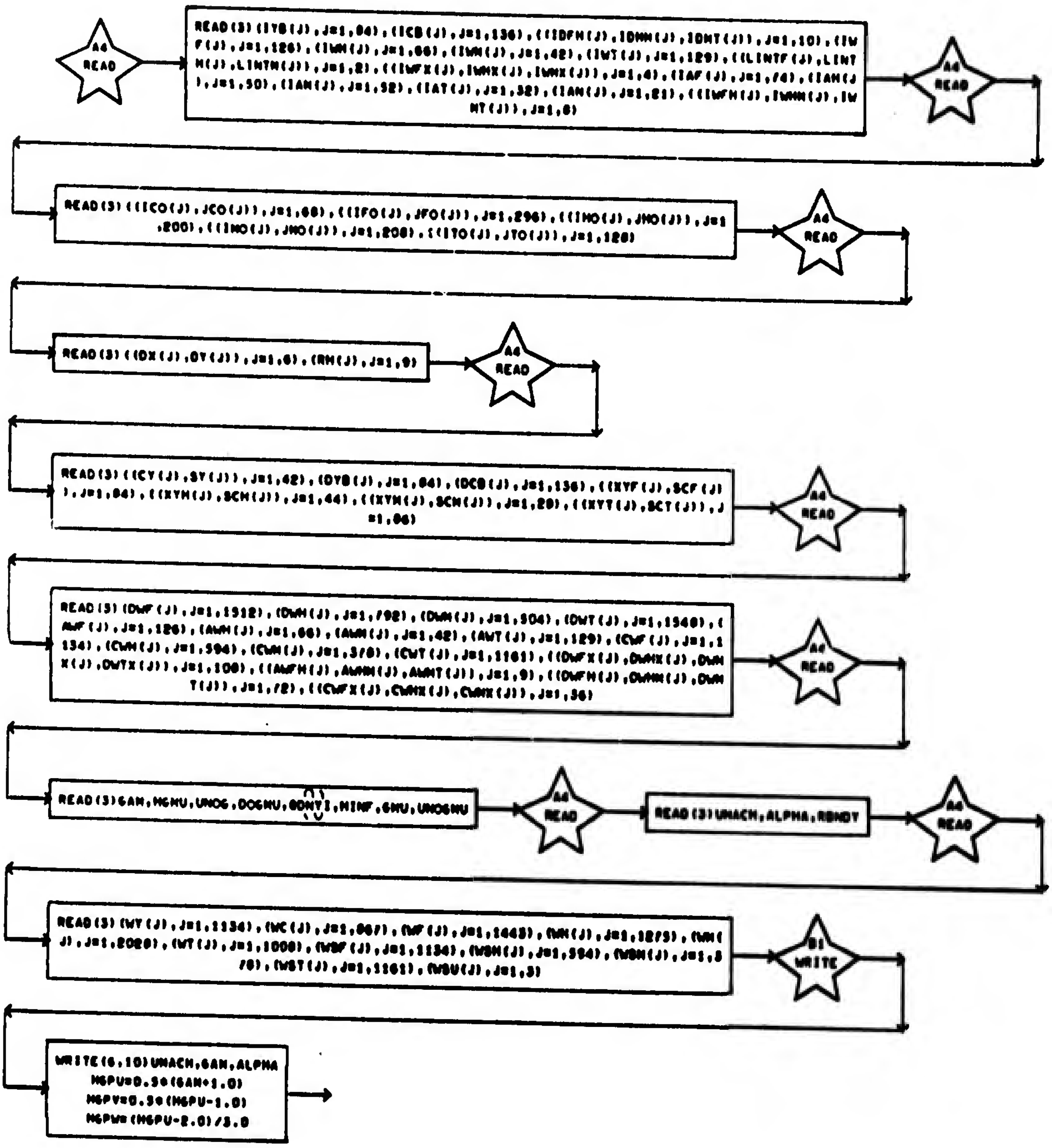
```
      SUBROUTINE TREND
C-----THIS SUBROUTINE TRIES TO PUSH CONDITIONS AT ALL MESH POINTS TOWARD
C     STEADY STATE FREE-STREAM TOTAL ENTHALPY, DISCREPANCIES ARE REDUCED
C     (1.0 - TEND) OF THEIR VALUE AT ALL REGULAR MESH AND BOUNDARY POINTS
C
      COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, GDYNI, HINF, GMU, UNOGMU
      COMMON/ENTHP/ JTND, TEND, TENB
      COMMON          WY(3,42,9), WC(3,17,17), WF(3,37,13), WH(3,25,17),
     A WM(3,26,26), WT(3,10,21), WSF(3,9,42), WSH(3,9,22), WSM(3,9,14),
     B WST(3,9,43), WSU(3)
C
      DO 10 LM = 1, 3674
      LK = 3*LM
      R = 1.0/WY(LK)
      UX = WY(LK-2)*R
      UY = WY(LK-1)*R
      E = WY(LK)**GMU
      HTOT = 2.5*E + 0.5*(UX**2 + UY**2)
      EFAC = 1.0 - TEND + TEND*HINF/HTOT
      WY(LK) = (E*EFAC)**UNOGMU
      UFAC = WY(LK)*SQRT(EFAC)
      WY(LK-2) = UFAC*UX
   10 WY(LK-1) = UFAC*UY
      RETURN
      END
```

APPENDIX V

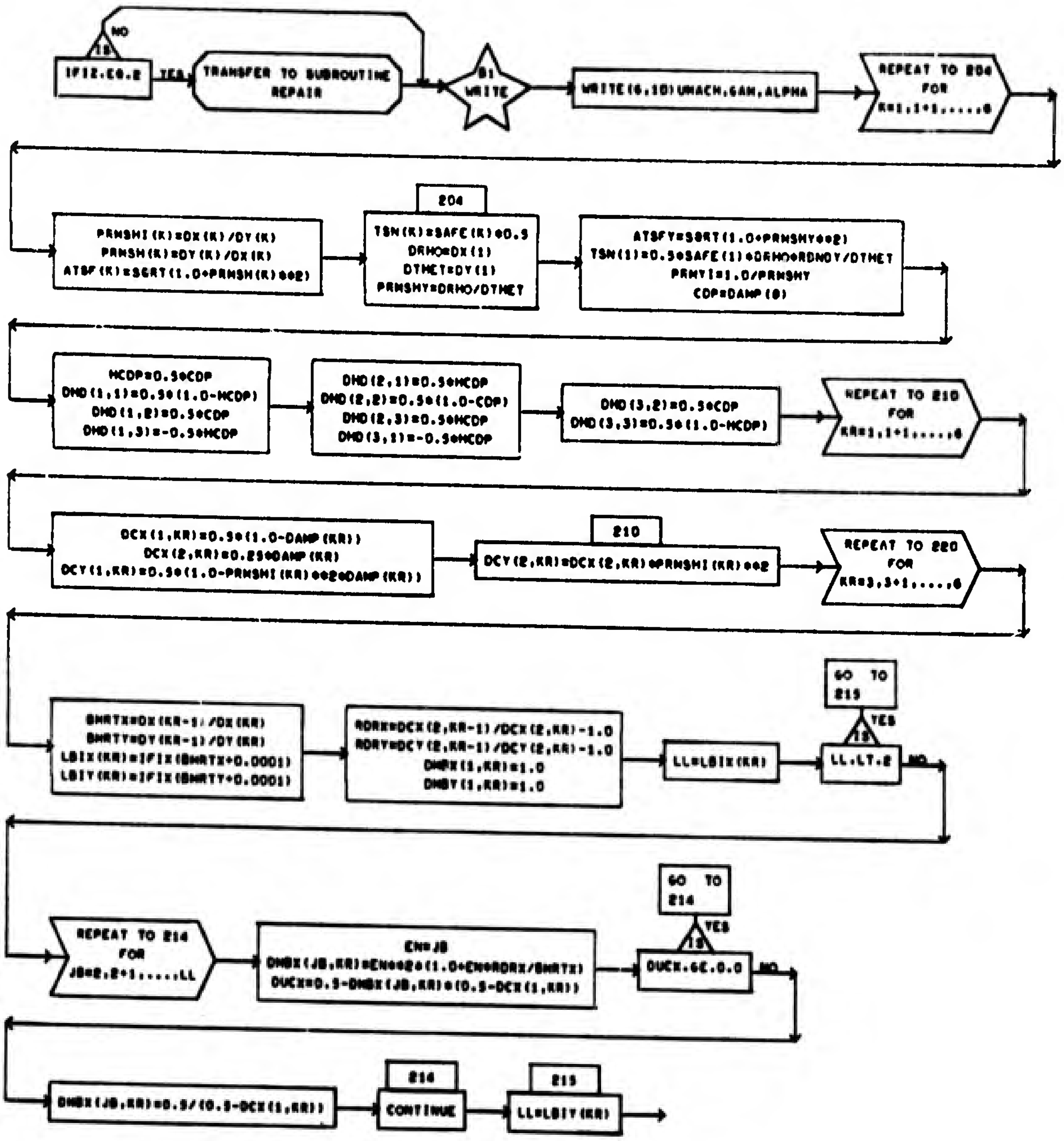
FLOW FIELD PROGRAM FLOW CHARTS



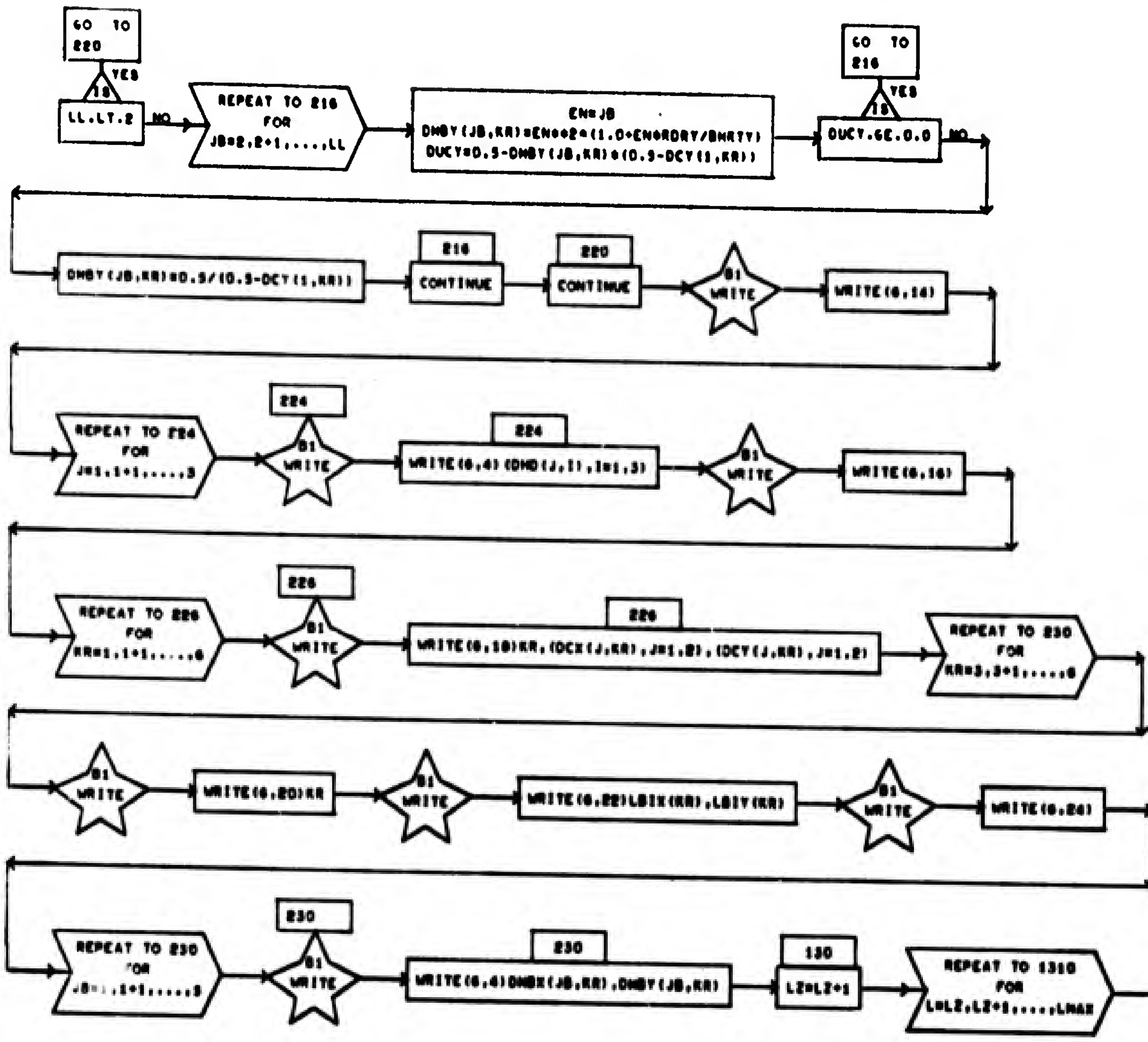
Detailed Flow Charts of Subroutine HANE (1 of 7)



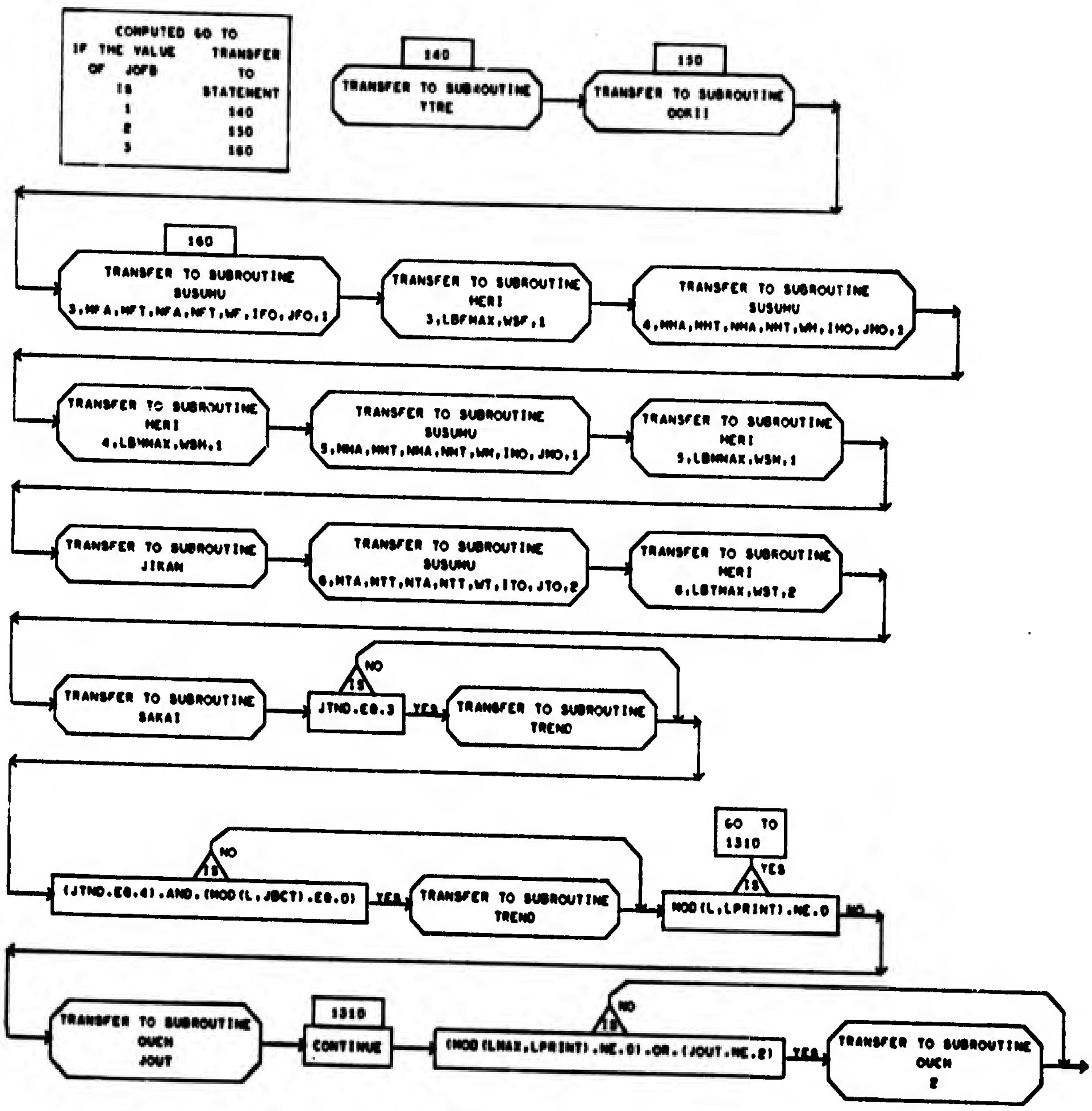
Detailed Flow Charts of Subroutine HANE (2 of 7)



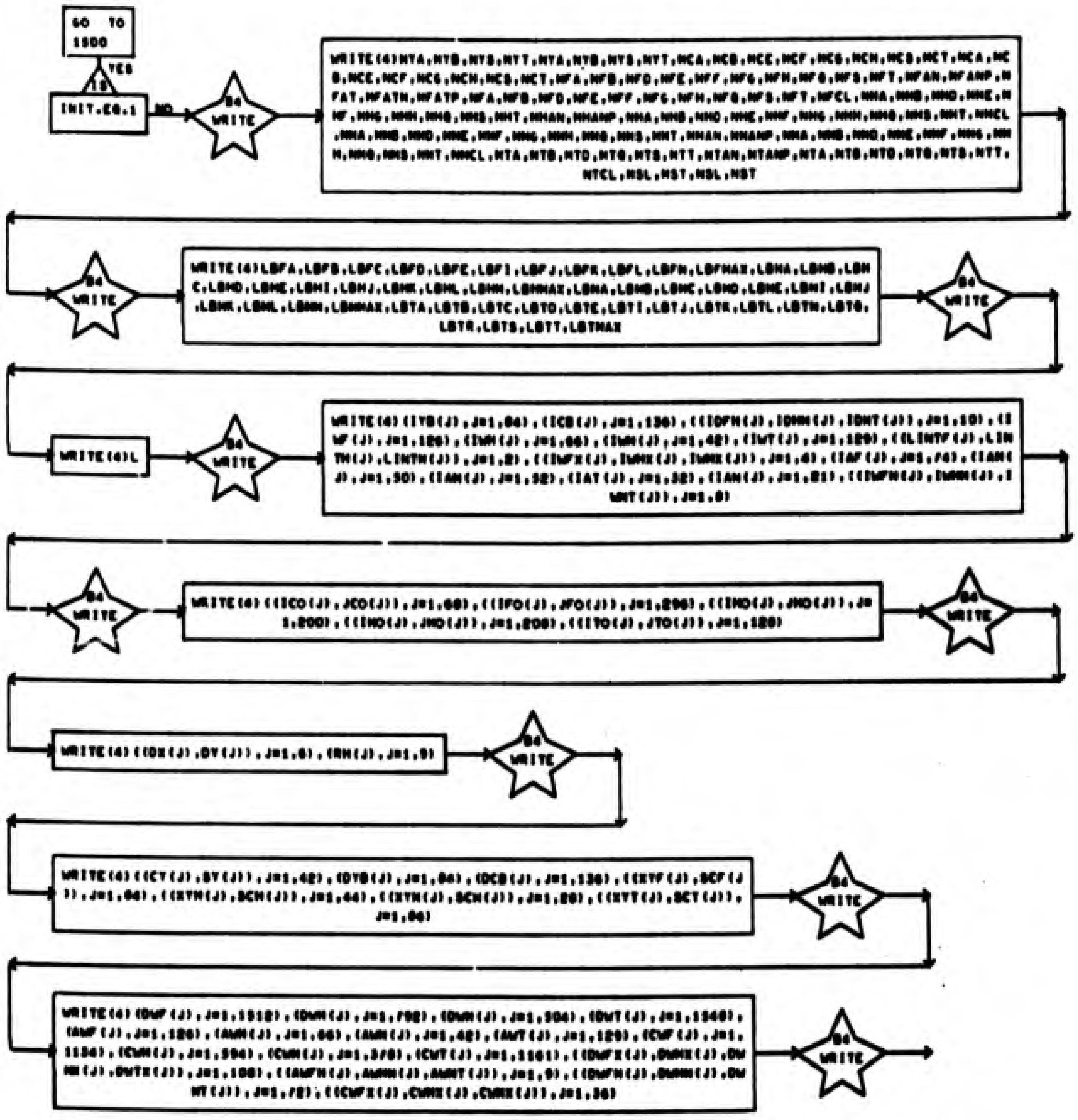
Detailed Flow Charts of Subroutine HANE (3 of 7)



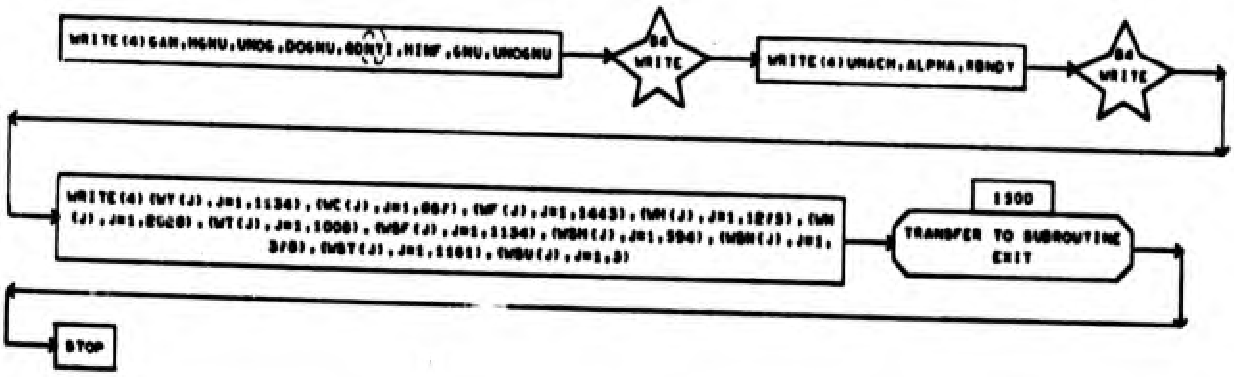
Detailed Flow Charts of Subroutine HANE (4 of 7)



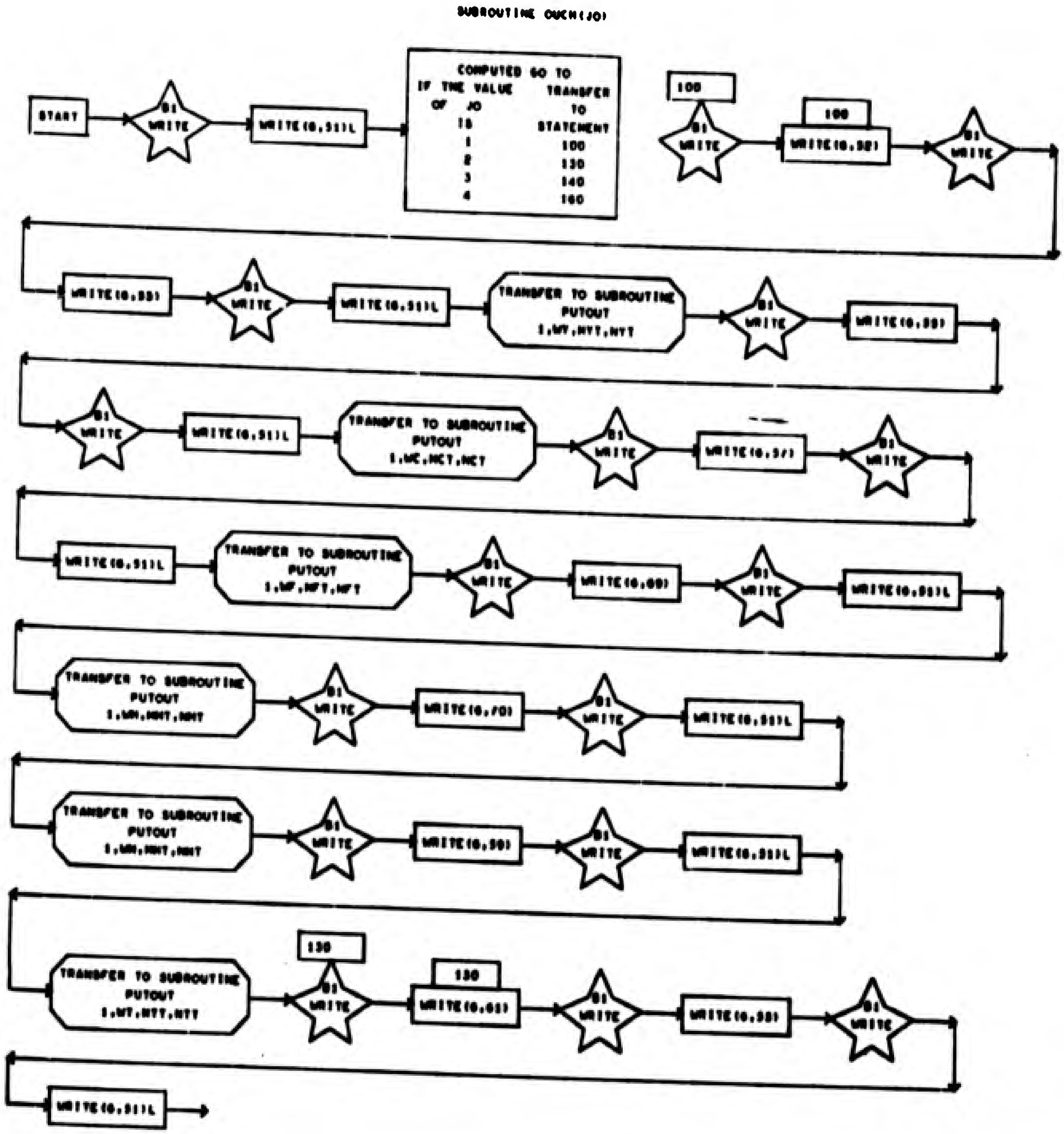
Detailed Flow Charts of Subroutine HANE (5 of 7)



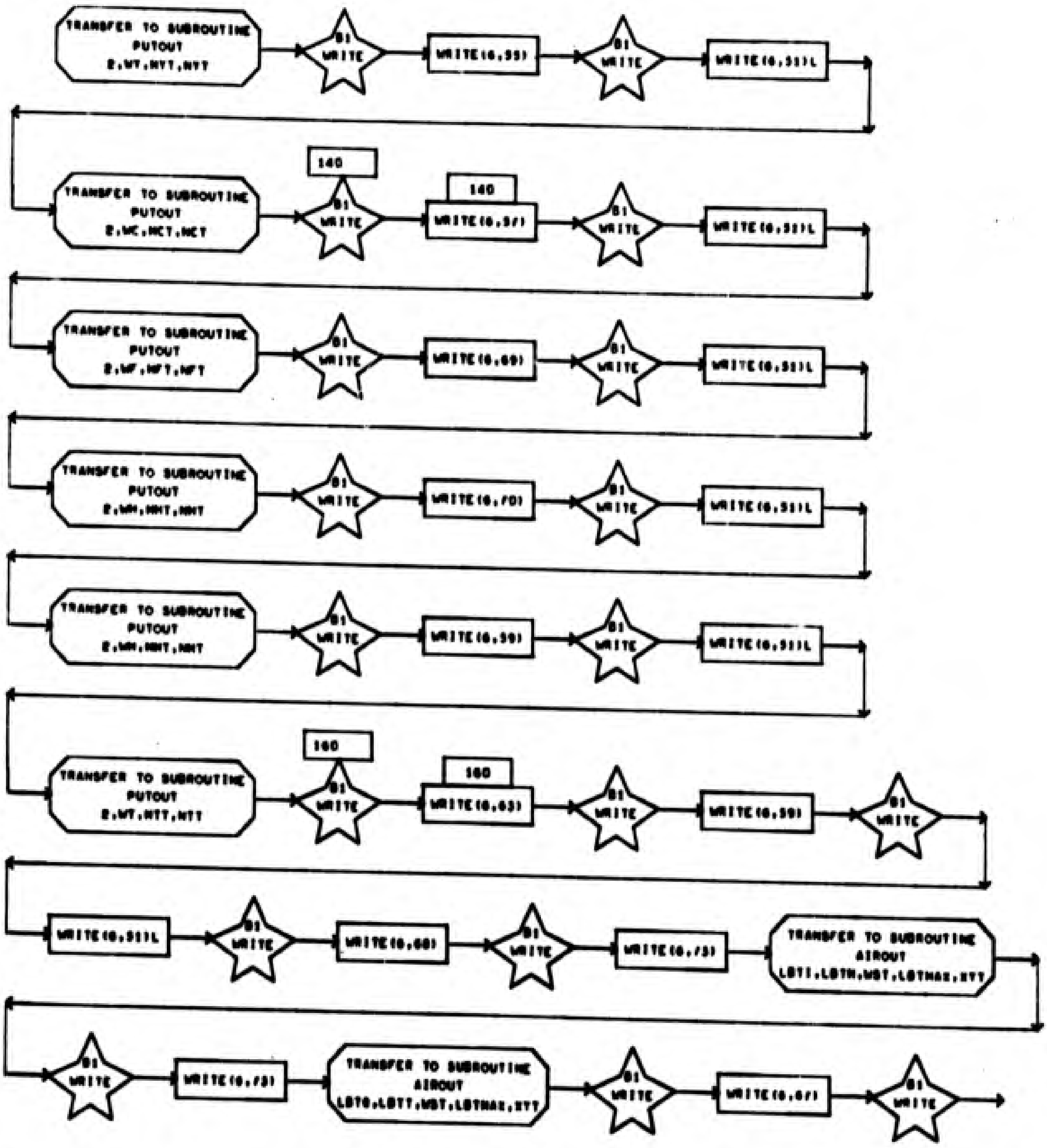
Detailed Flow Charts of Subroutine HANE (6 of 7)



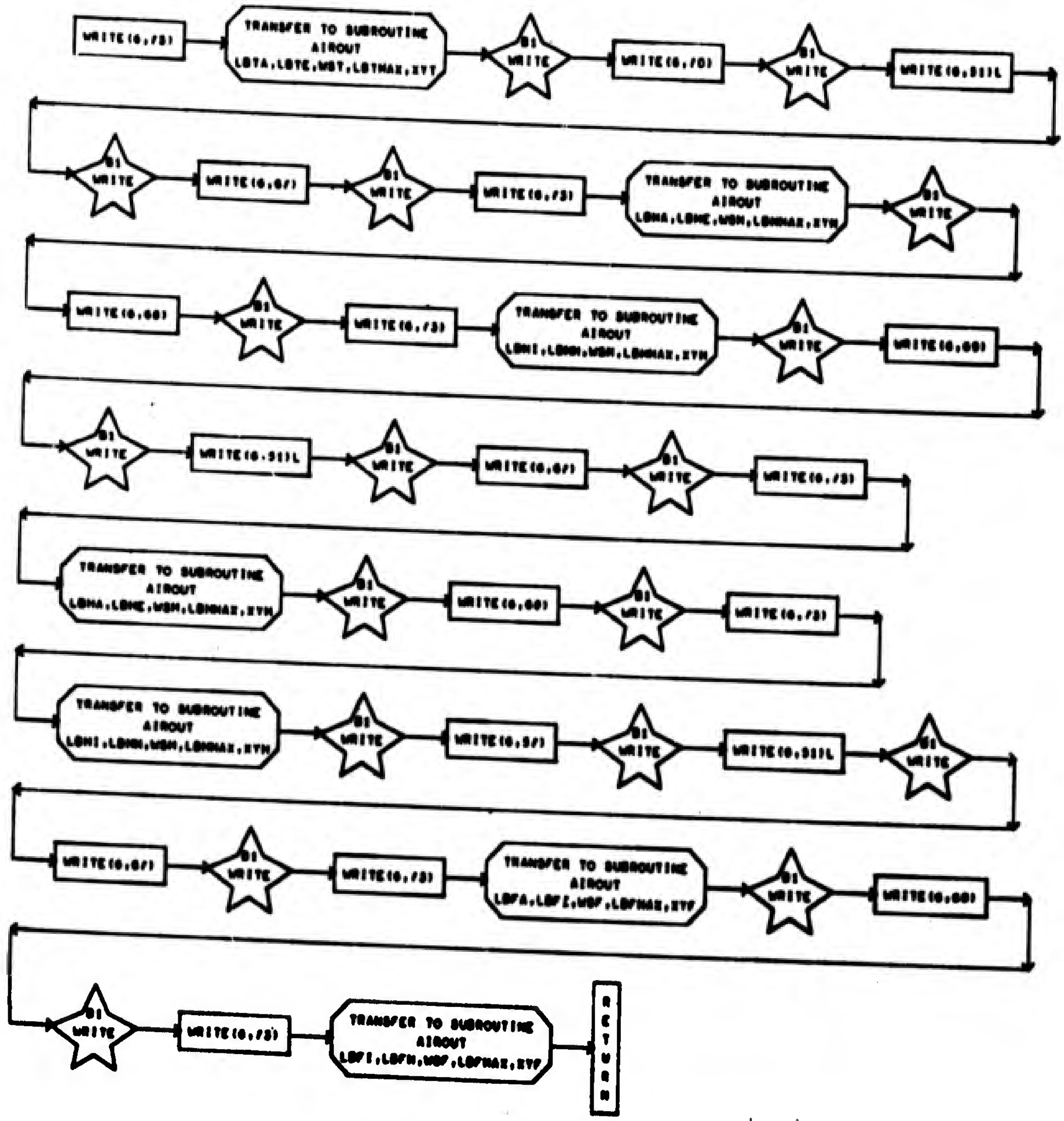
Detailed Flow Charts of Subroutine HANE (7 of 7)



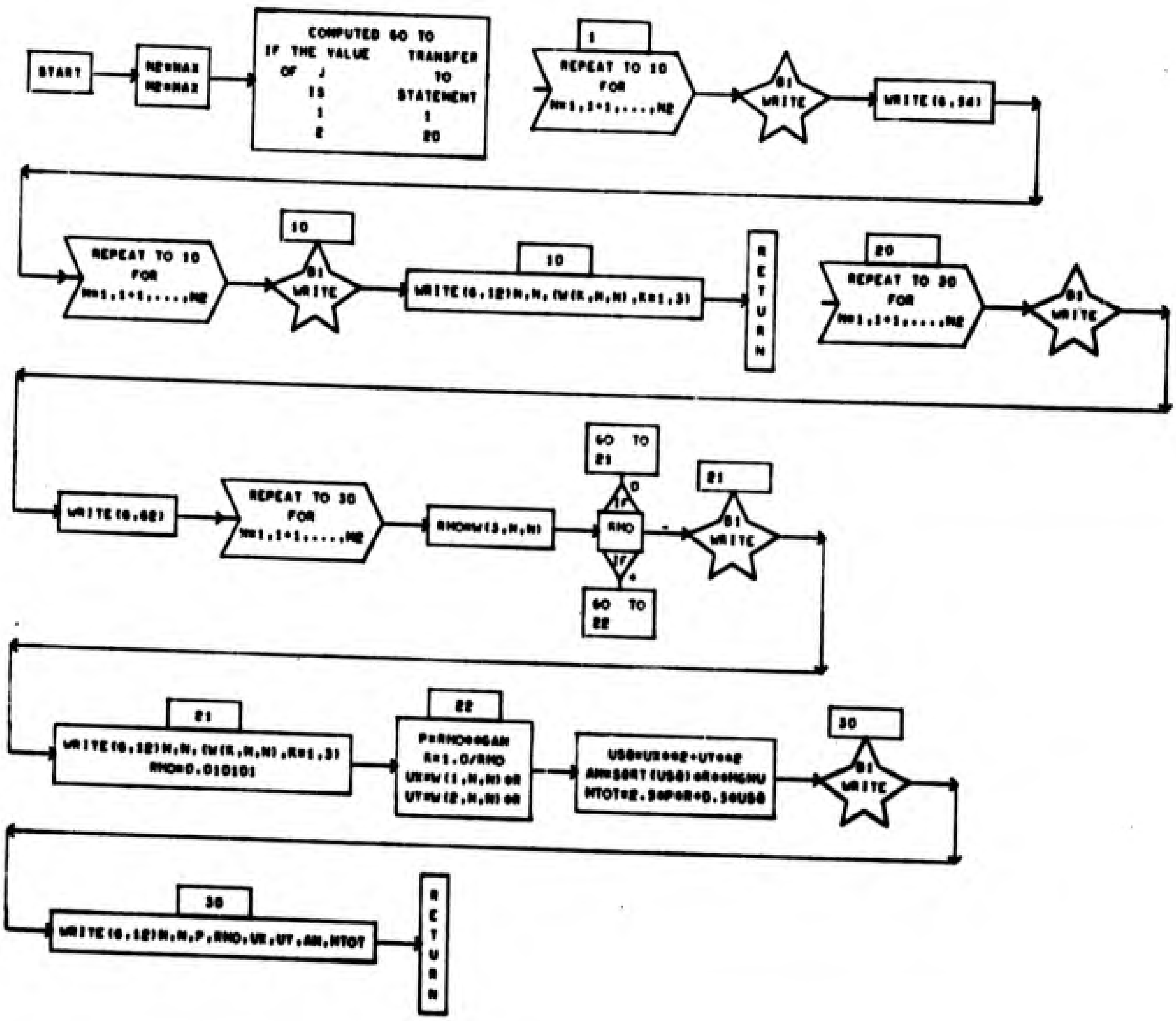
Detailed Flow Charts of Subroutine OUCH (1 of 3)



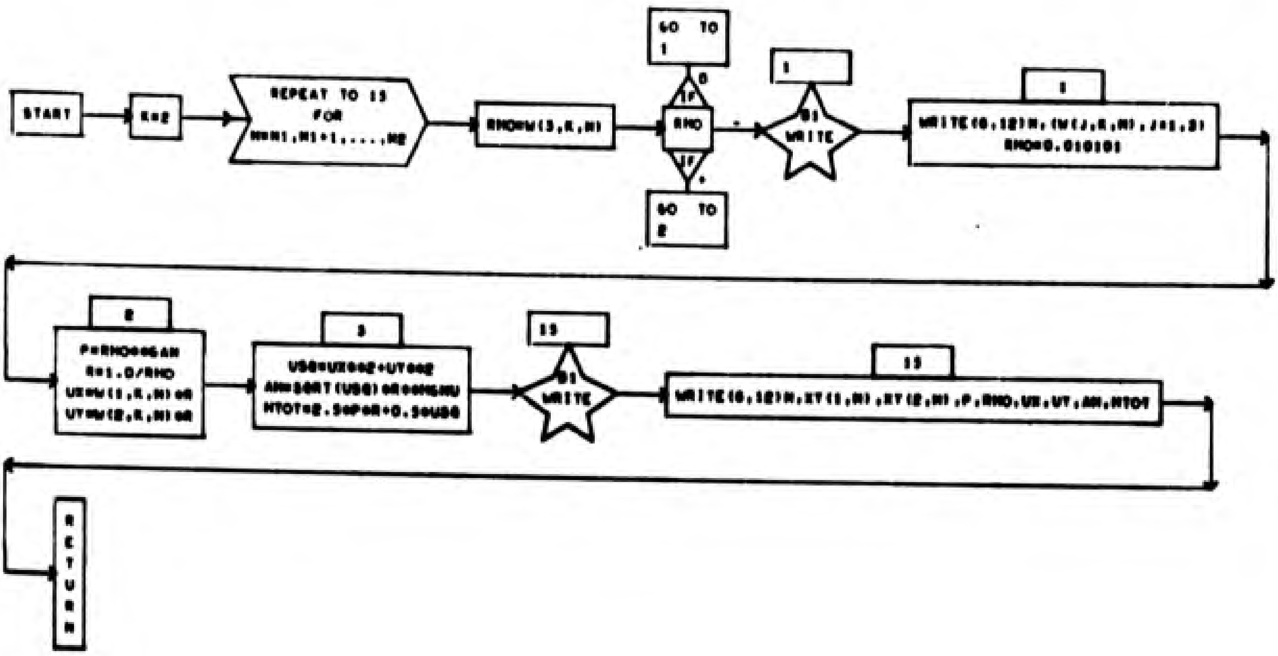
Detailed Flow Charts of Subroutine OUCH (2 of 3)



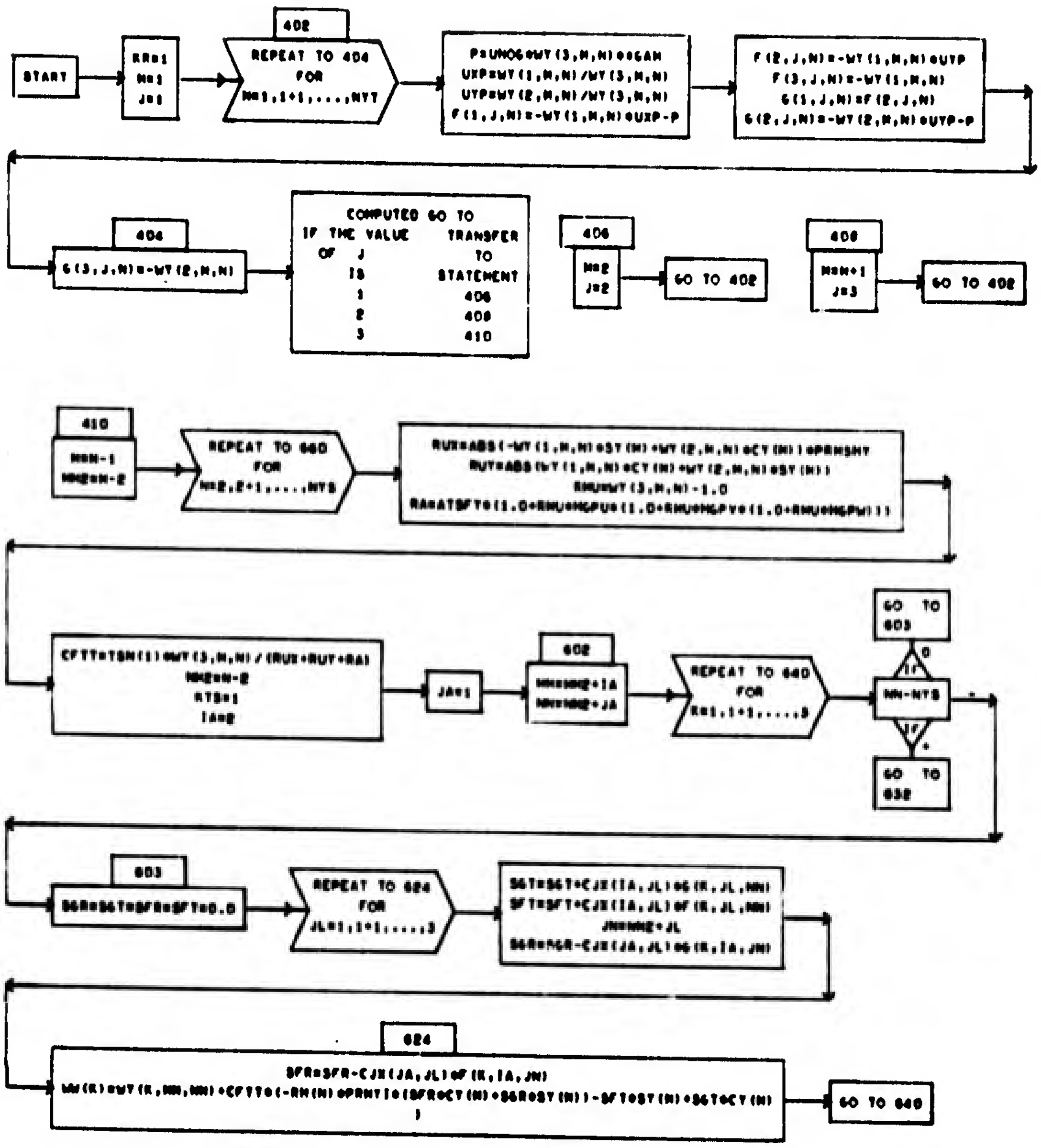
Detailed Flow Charts of Subroutine OUCH (3 of 3)



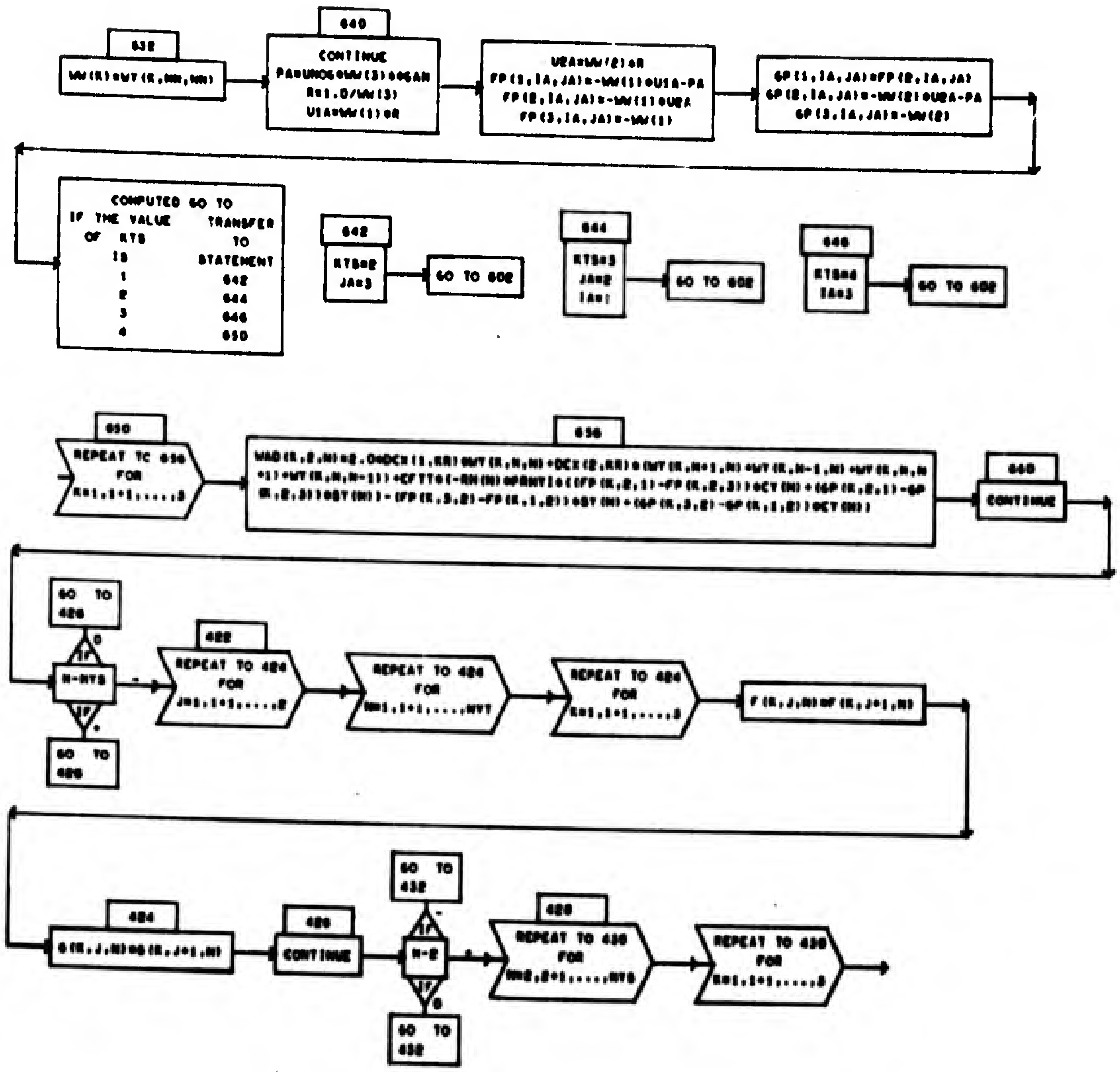
Detailed Flow Chart of Subroutine PUTOUT



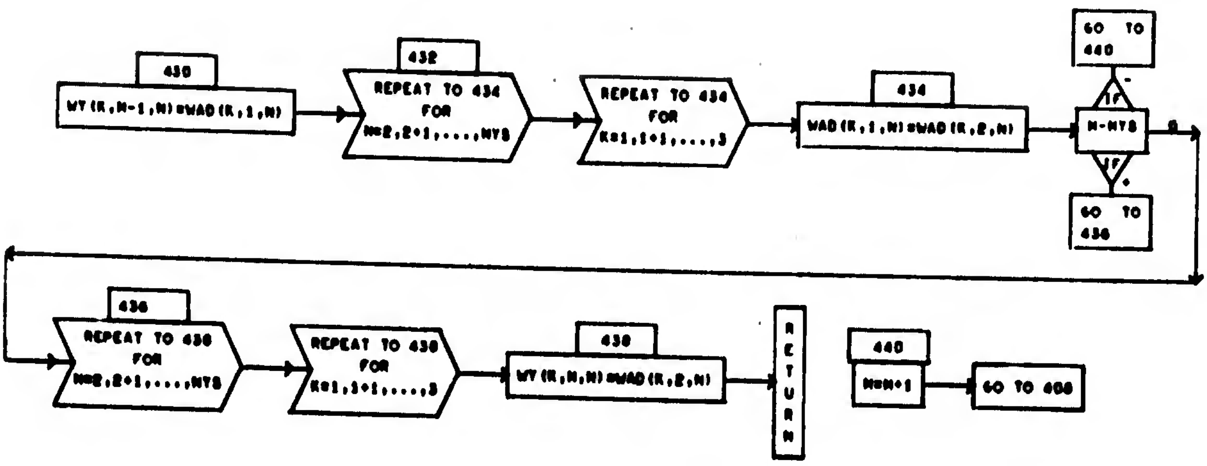
Detailed Flow Charts of Subroutine AIROUT



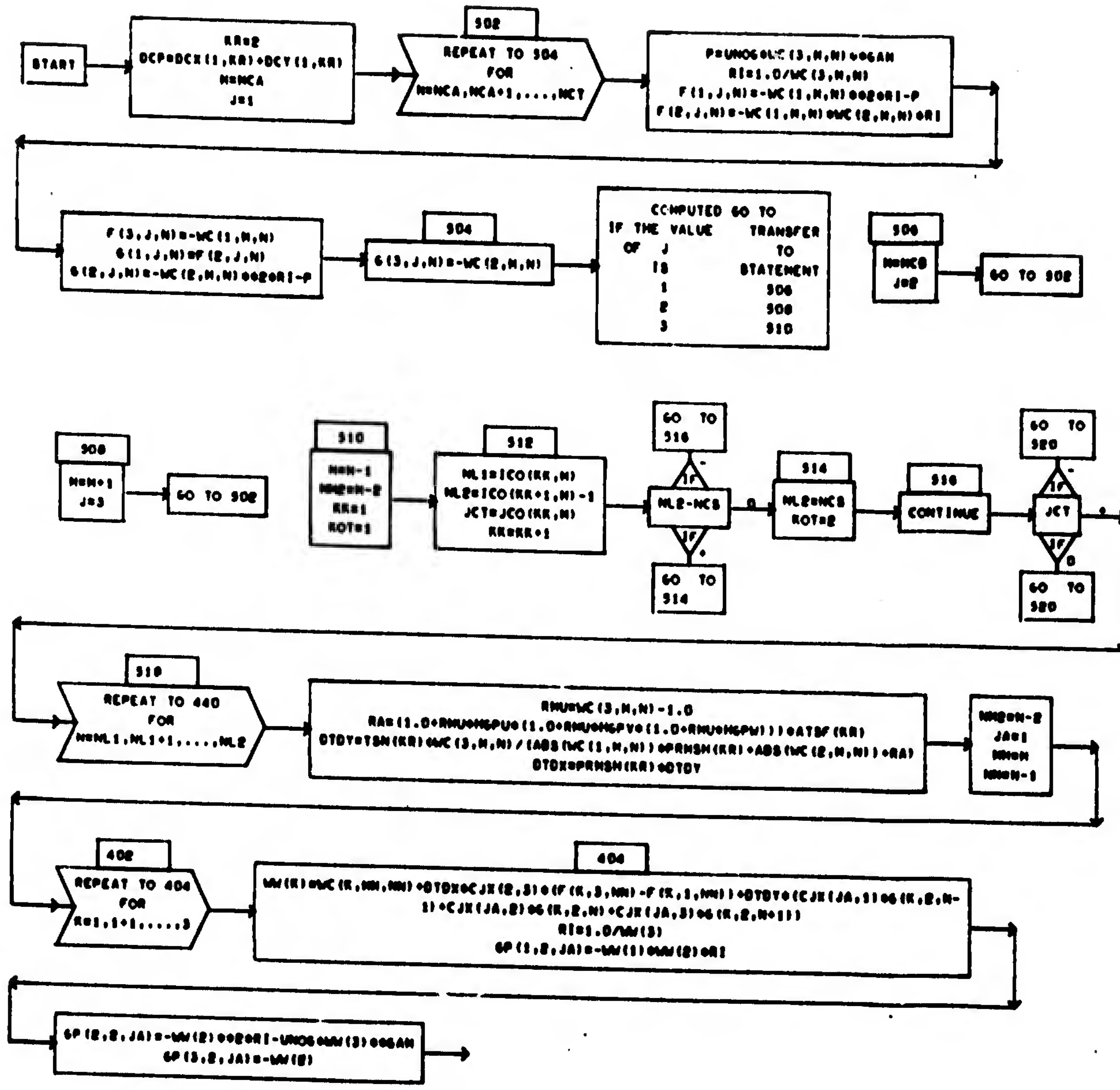
Detailed Flow Charts of Subroutine YTRF (1 of 3)



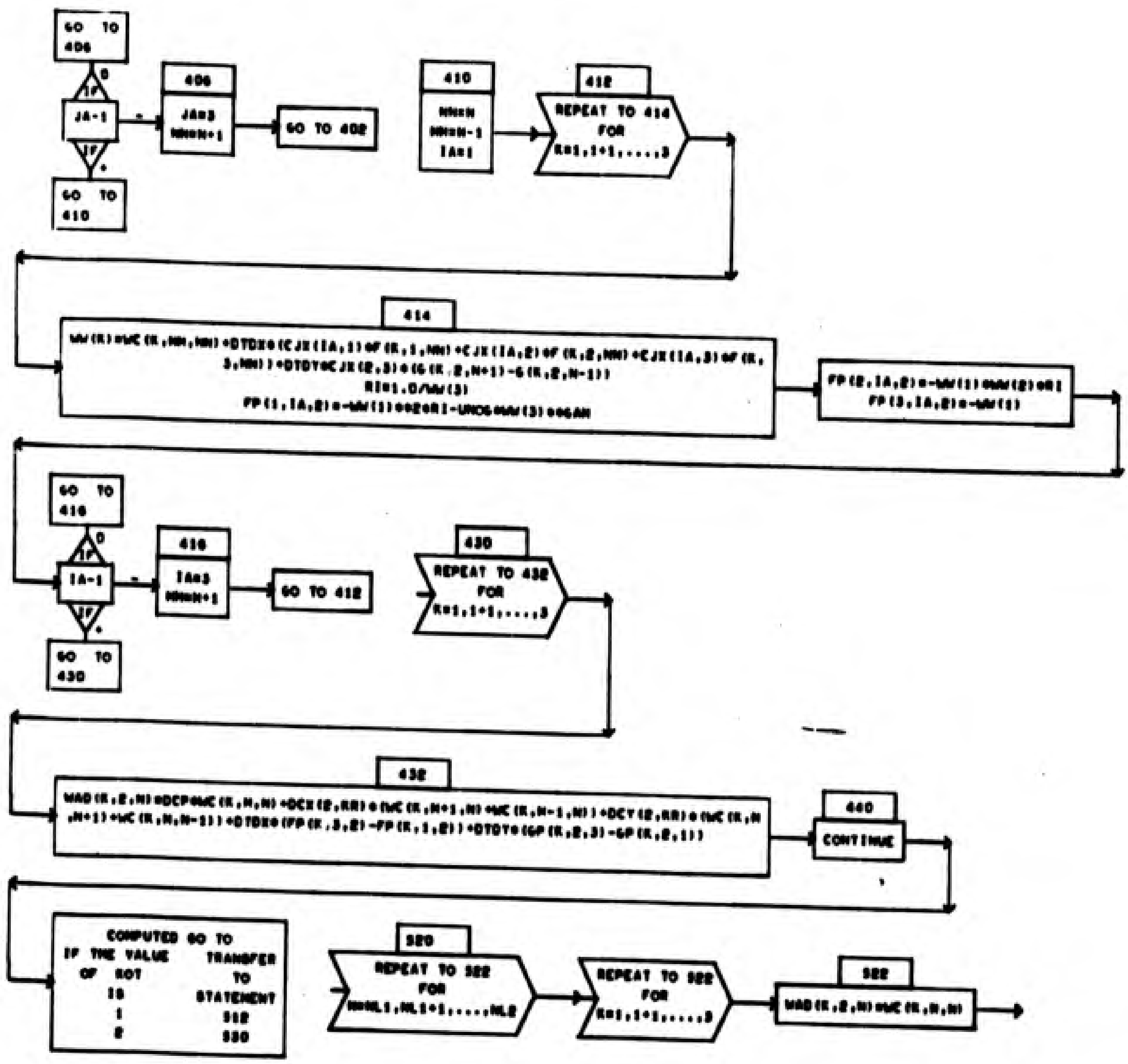
Detailed Flow Charts of Subroutine YTRE (2 of 3)



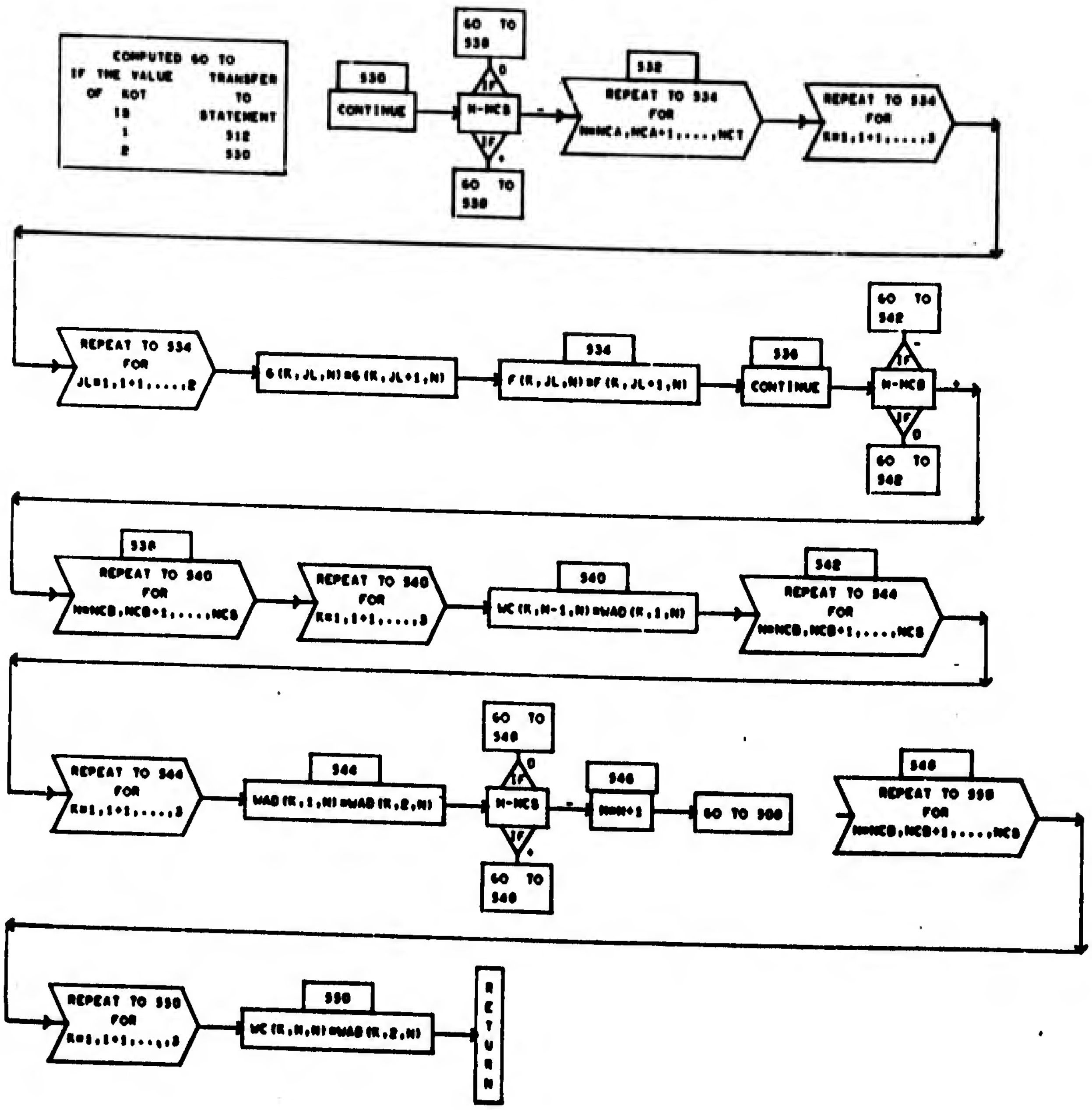
Detailed Flow Charts of Subroutine YTRE (3 of 3)



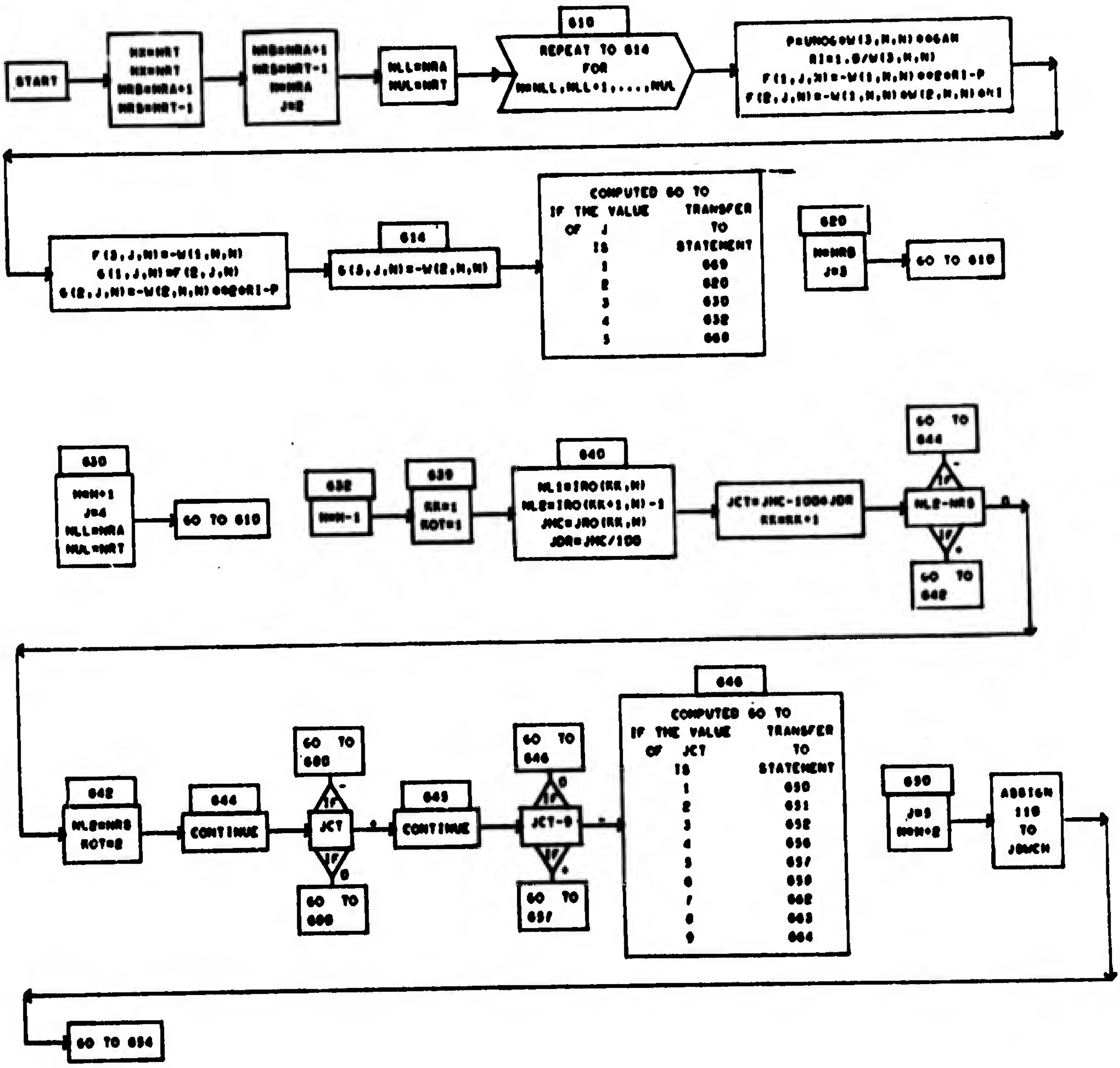
Detailed Flow Charts of Subroutine OOKII (1 of 3)



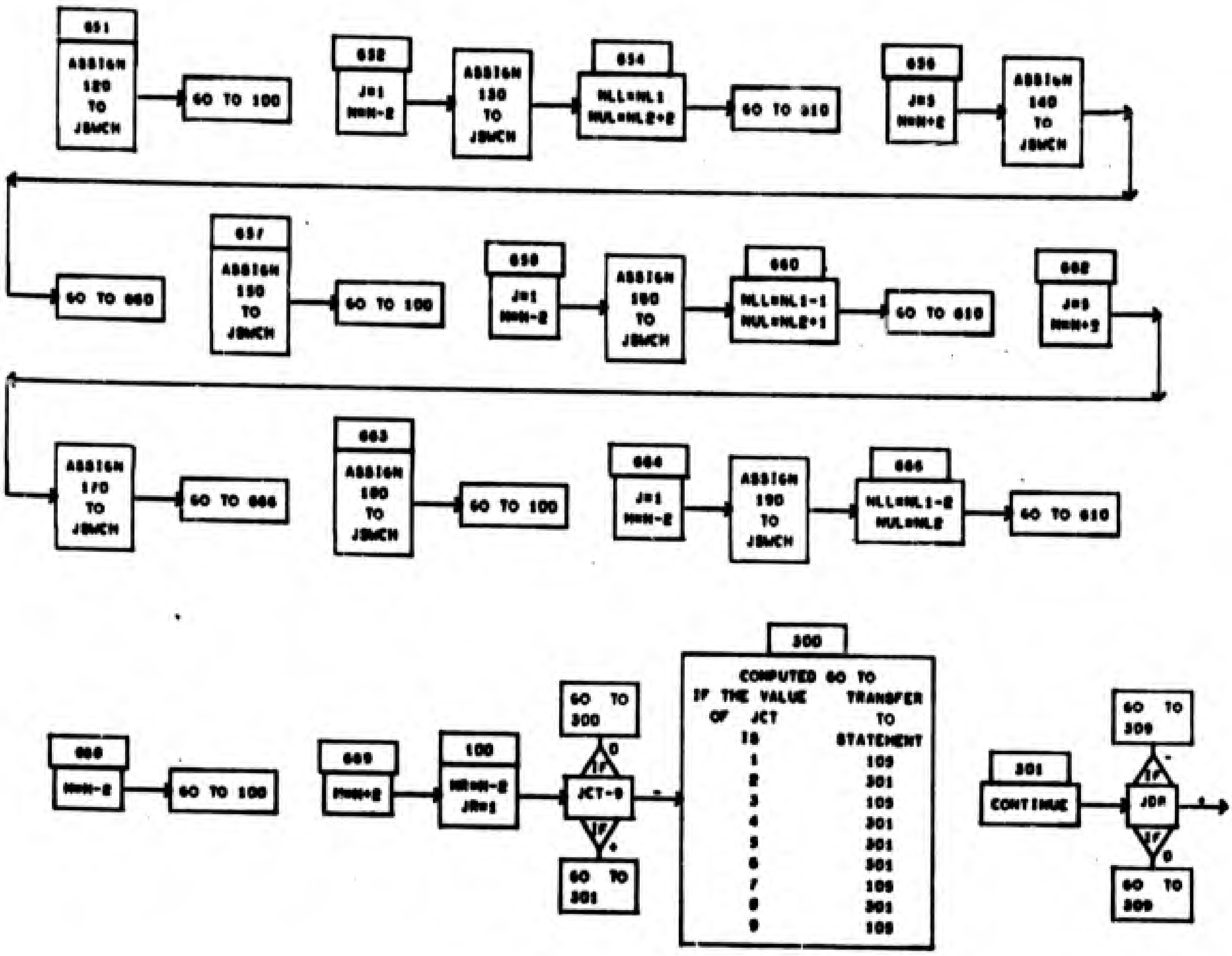
Detailed Flow Charts of Subroutine OOKII (2 of 3)



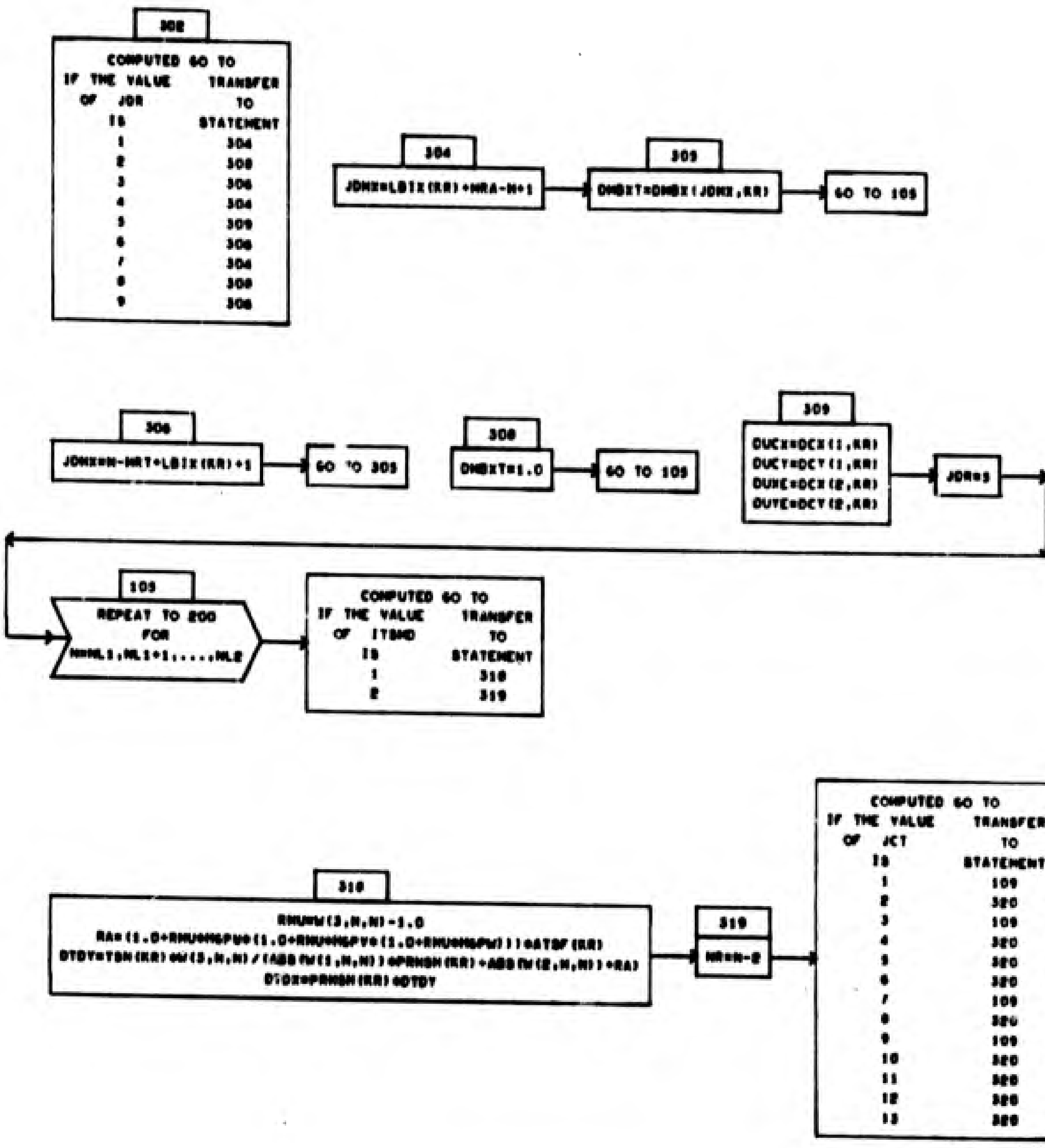
Detailed Flow Charts of Subroutine OOKII (3 of 3)



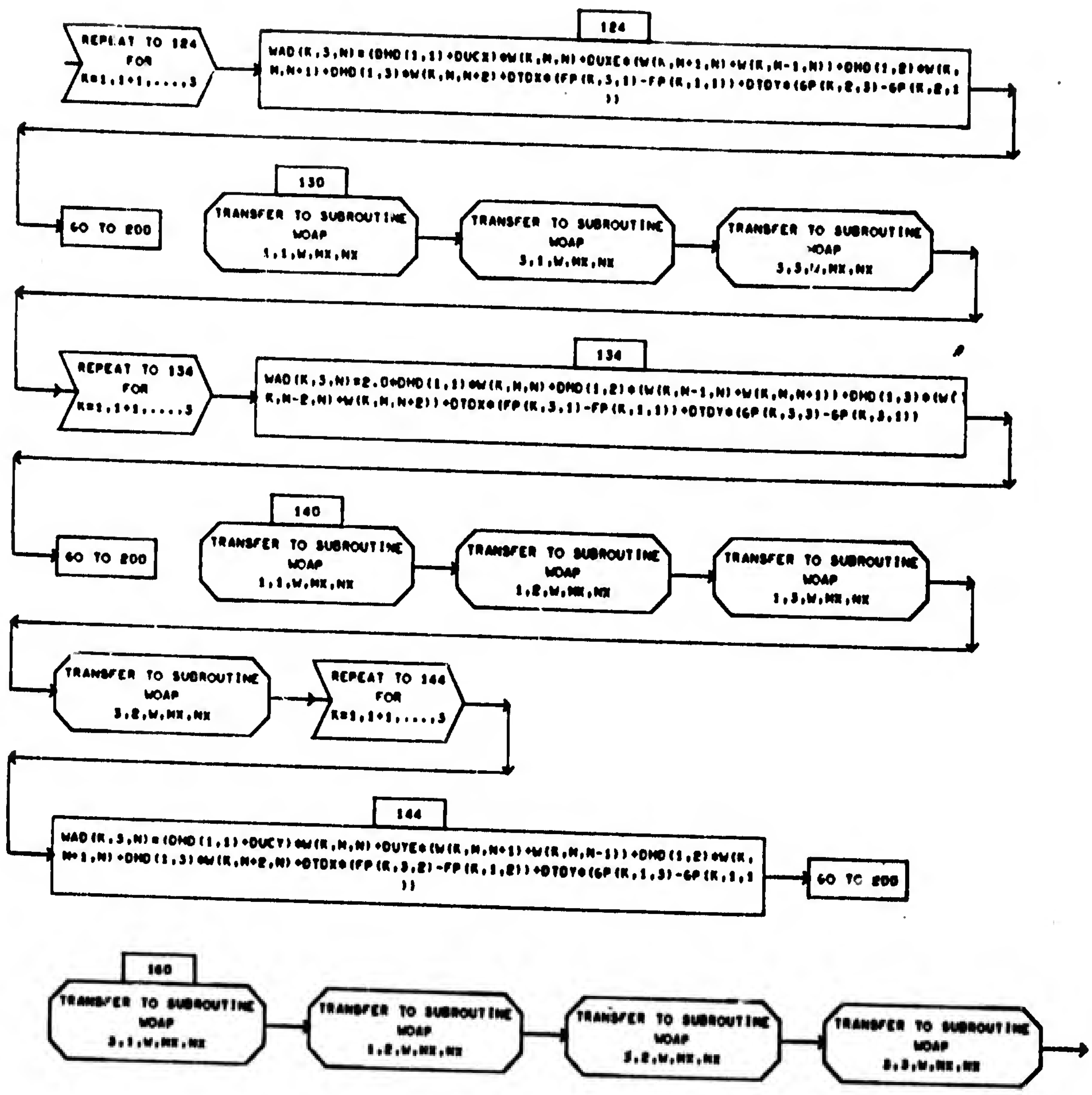
Detailed Flow Chart of Subroutine SUSUMU (1 of 9)



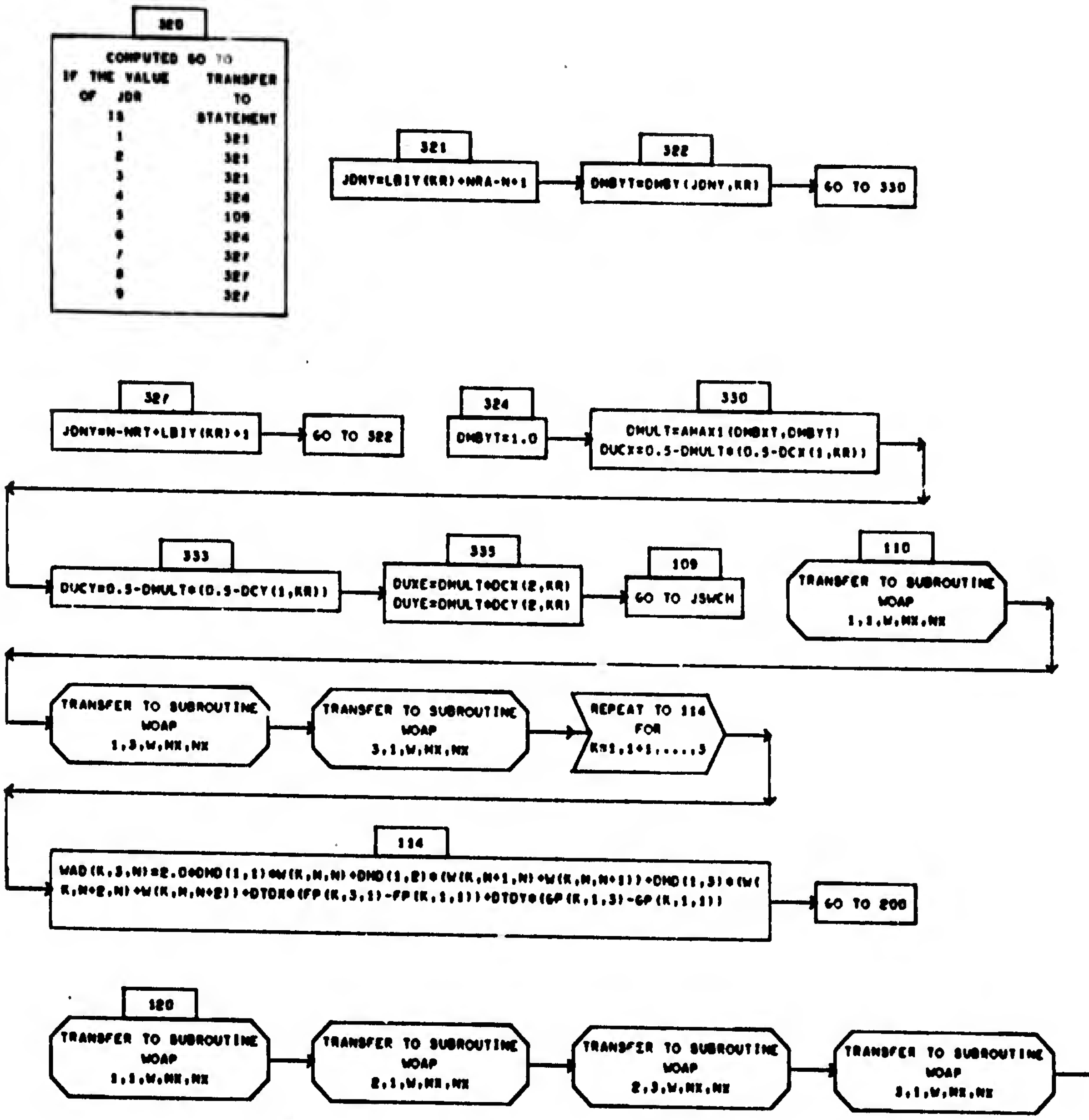
Detailed Flow Chart of Subroutine SUSUMU (2 of 9)



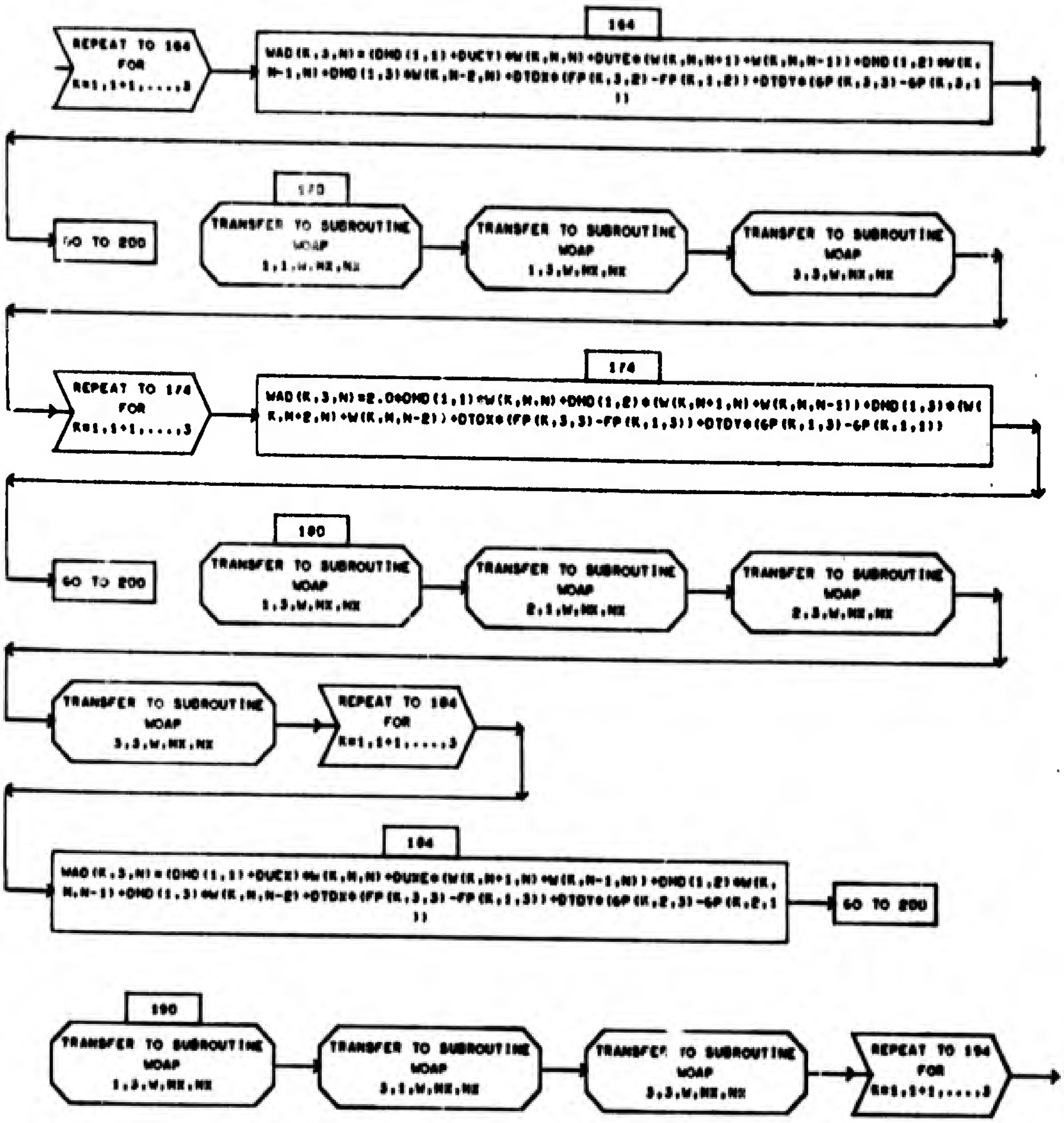
Detailed Flow Chart of Subroutine SUSUMU (3 of 9)



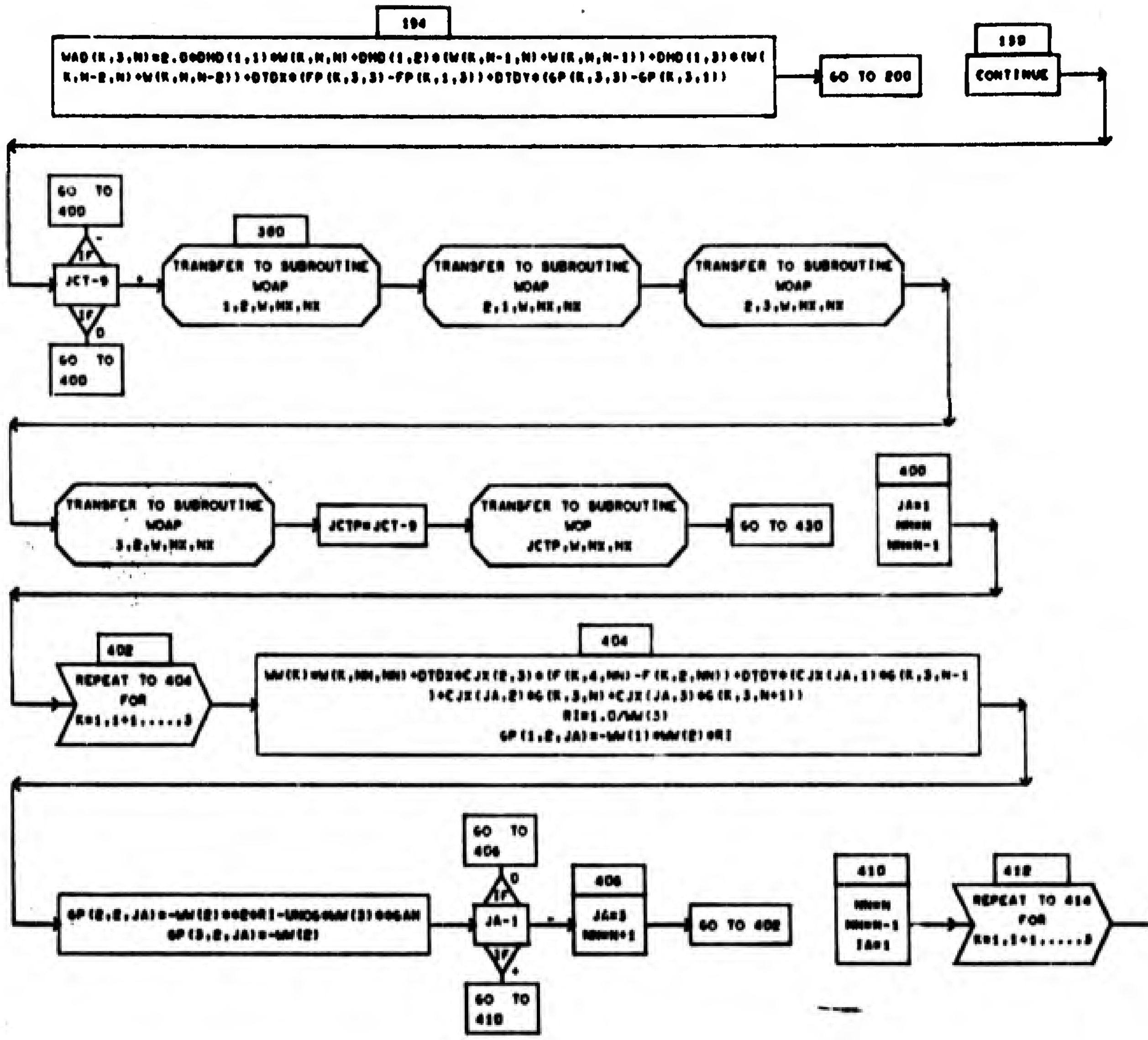
Detailed Flow Chart of Subroutine SUSUMU (4 of 9)



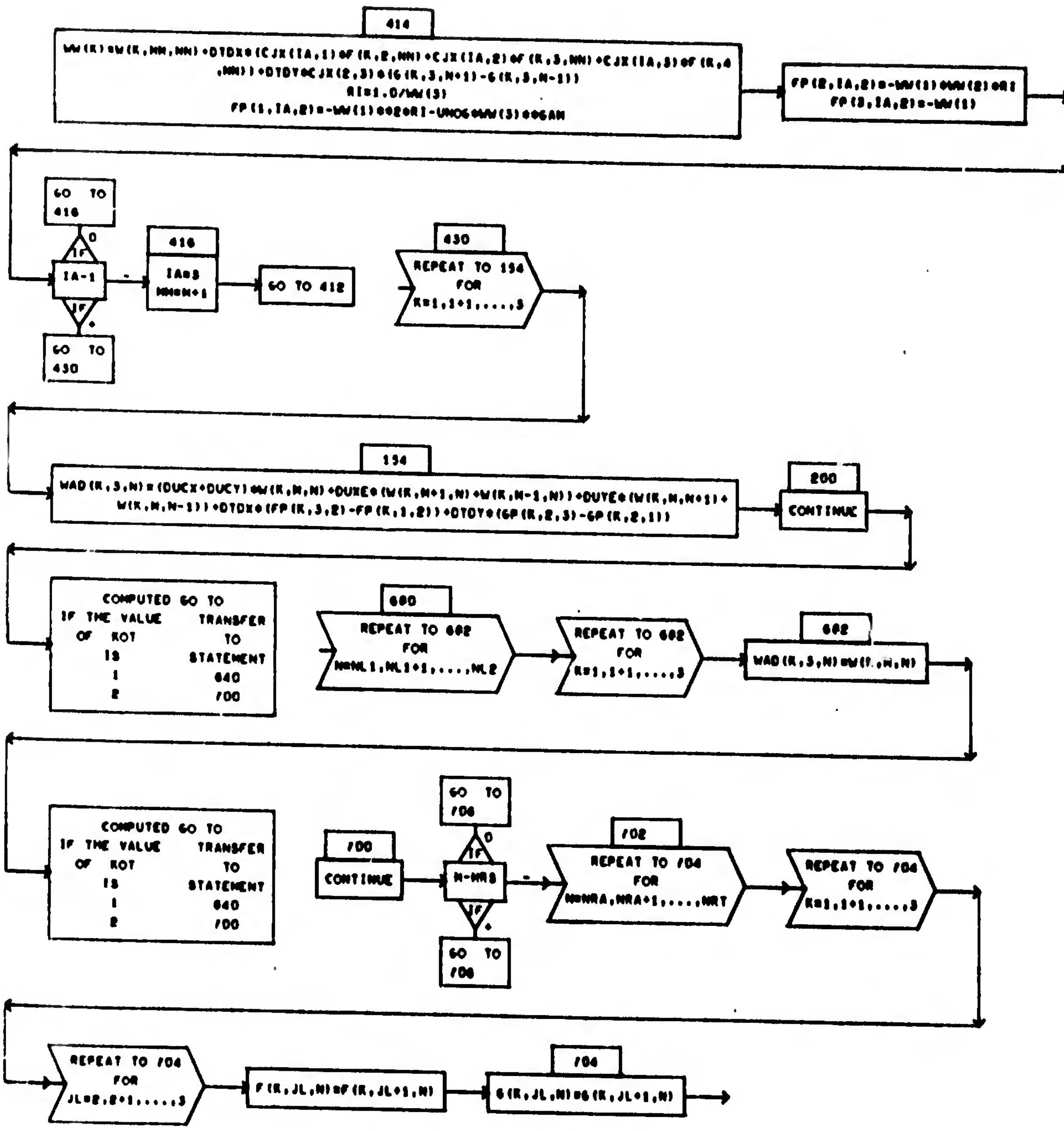
Detailed Flow Chart of Subroutine SUSUMU (5 of 9)



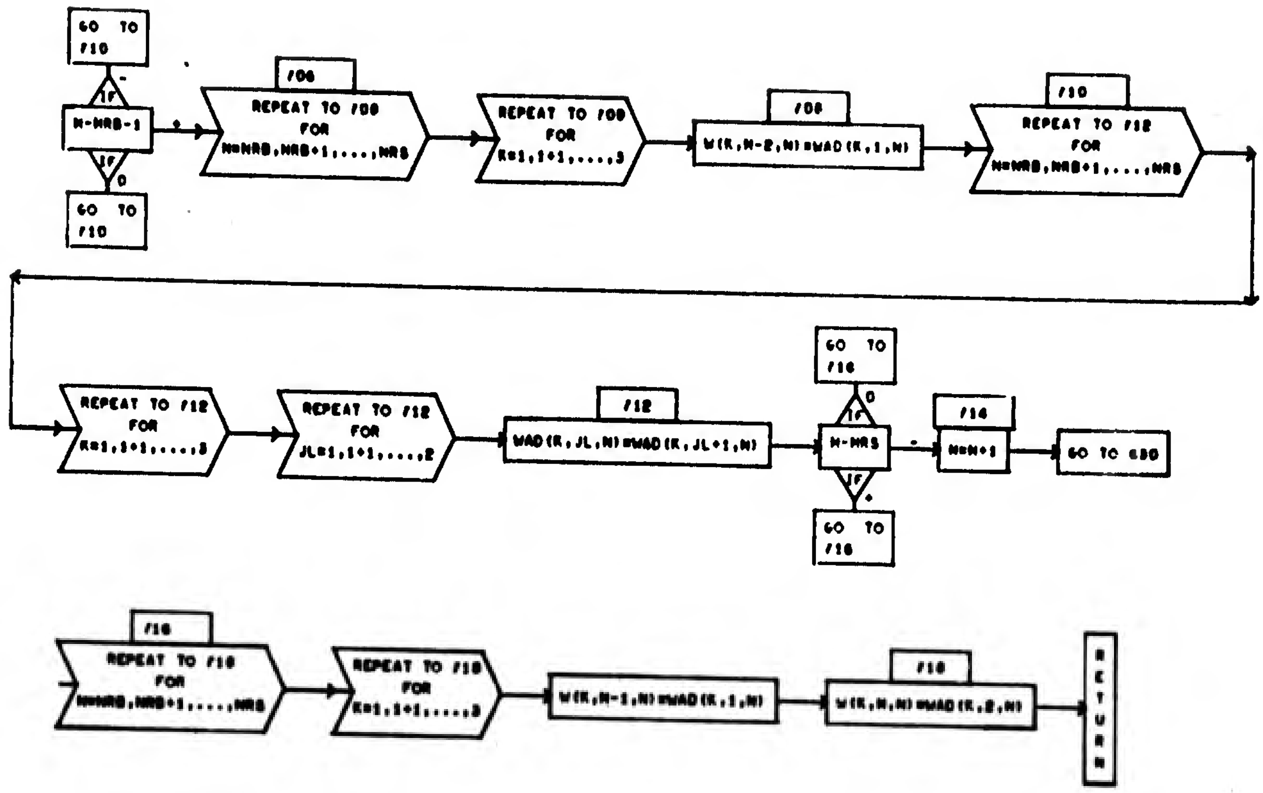
Detailed Flow Chart of Subroutine SUSUMU (6 of 9)



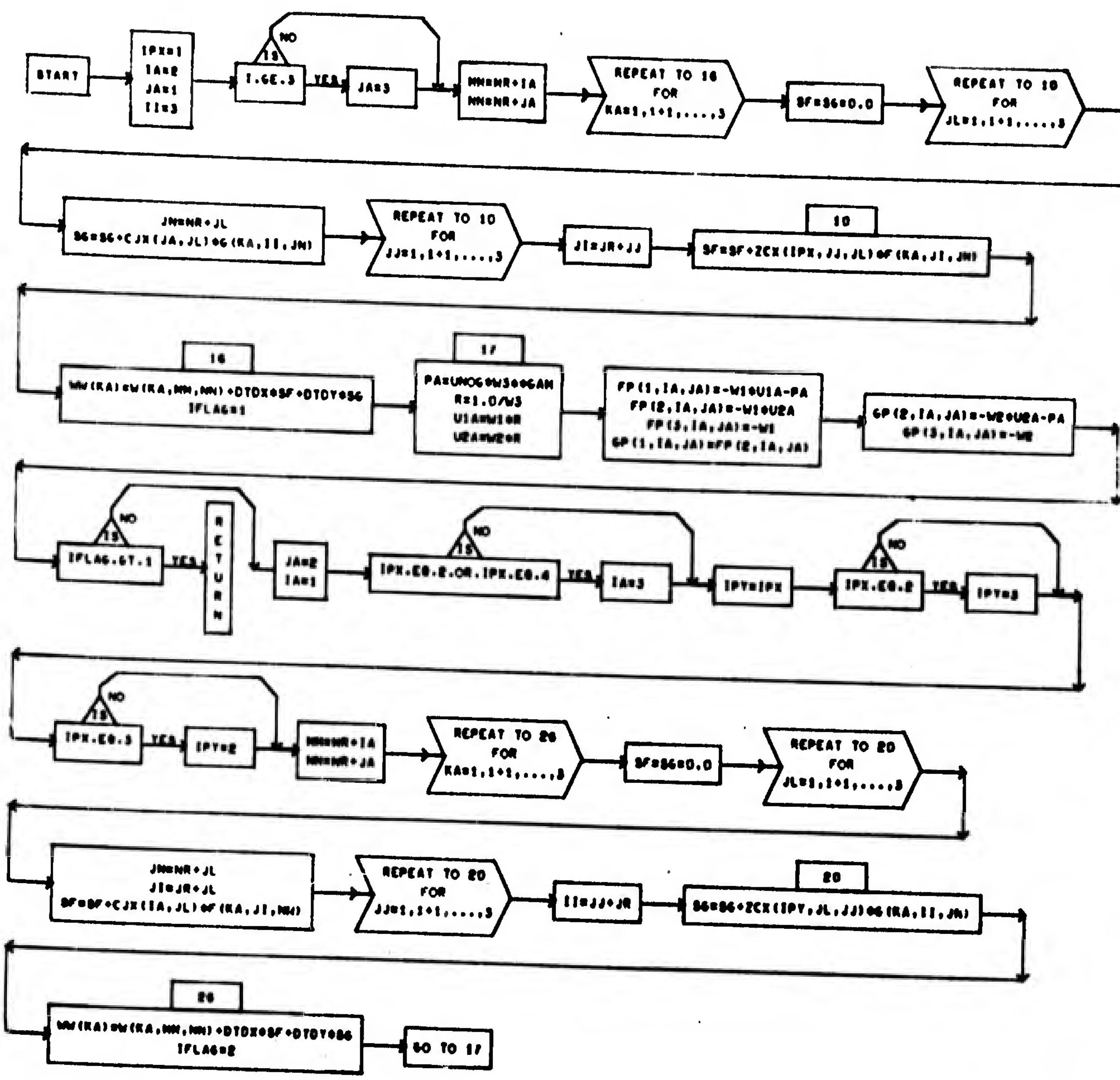
Detailed Flow Chart of Subroutine SUSUMU (7 of 9)



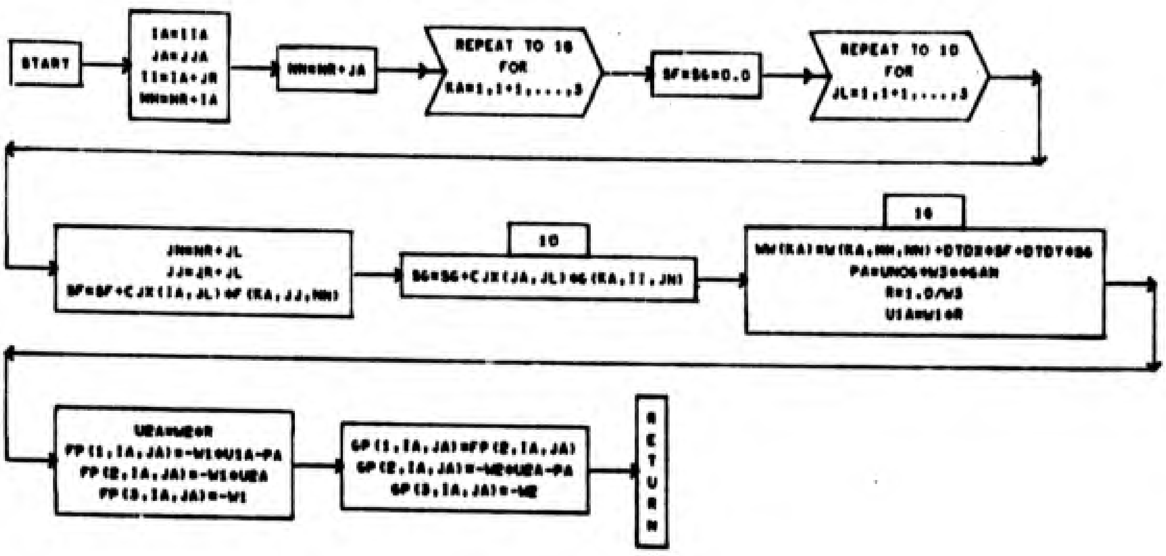
Detailed Flow Chart of Subroutine SUSUMU (8 of 9)



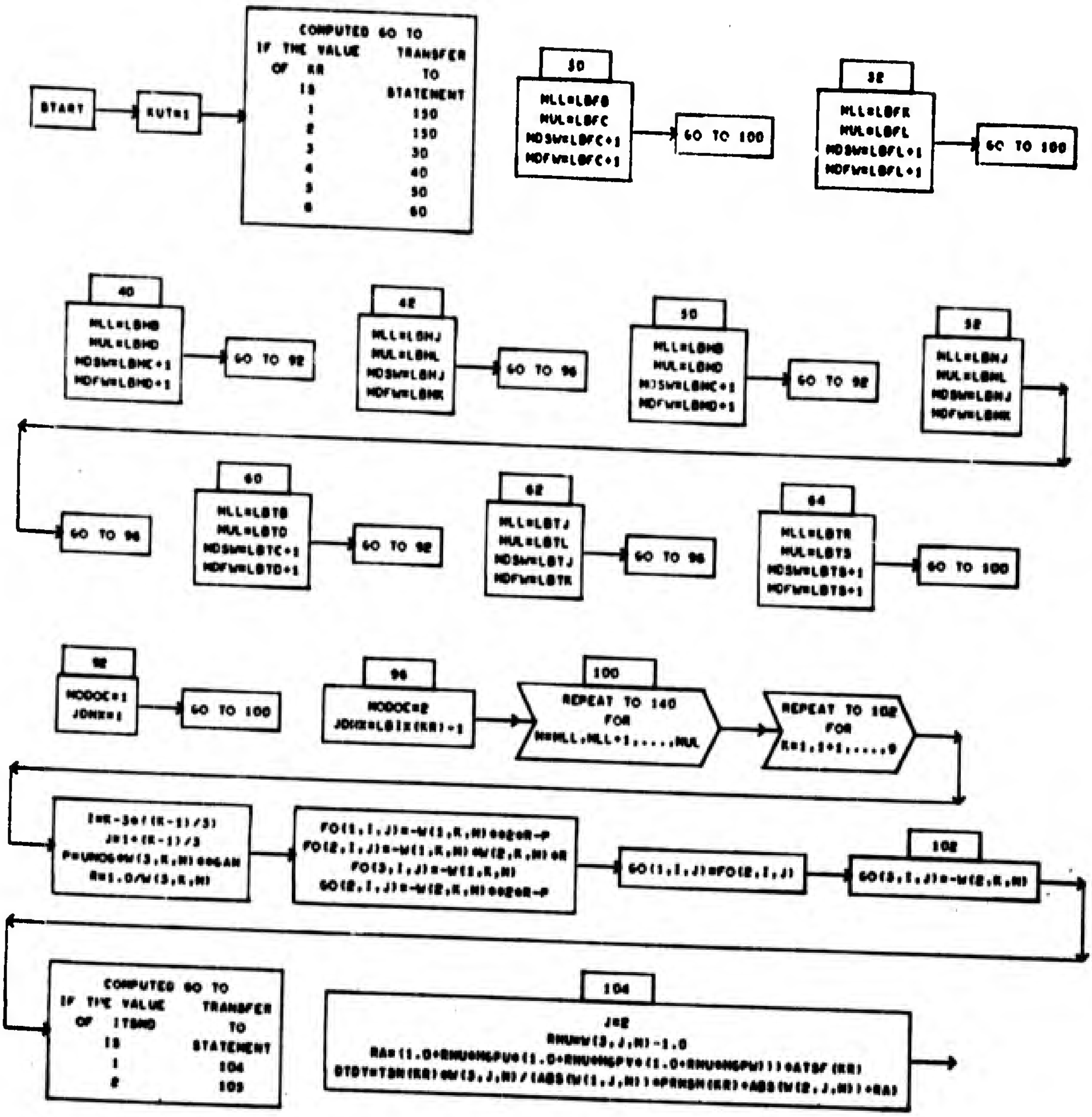
Detailed Flow Chart of Subroutine SUSUMU (9 of 9)



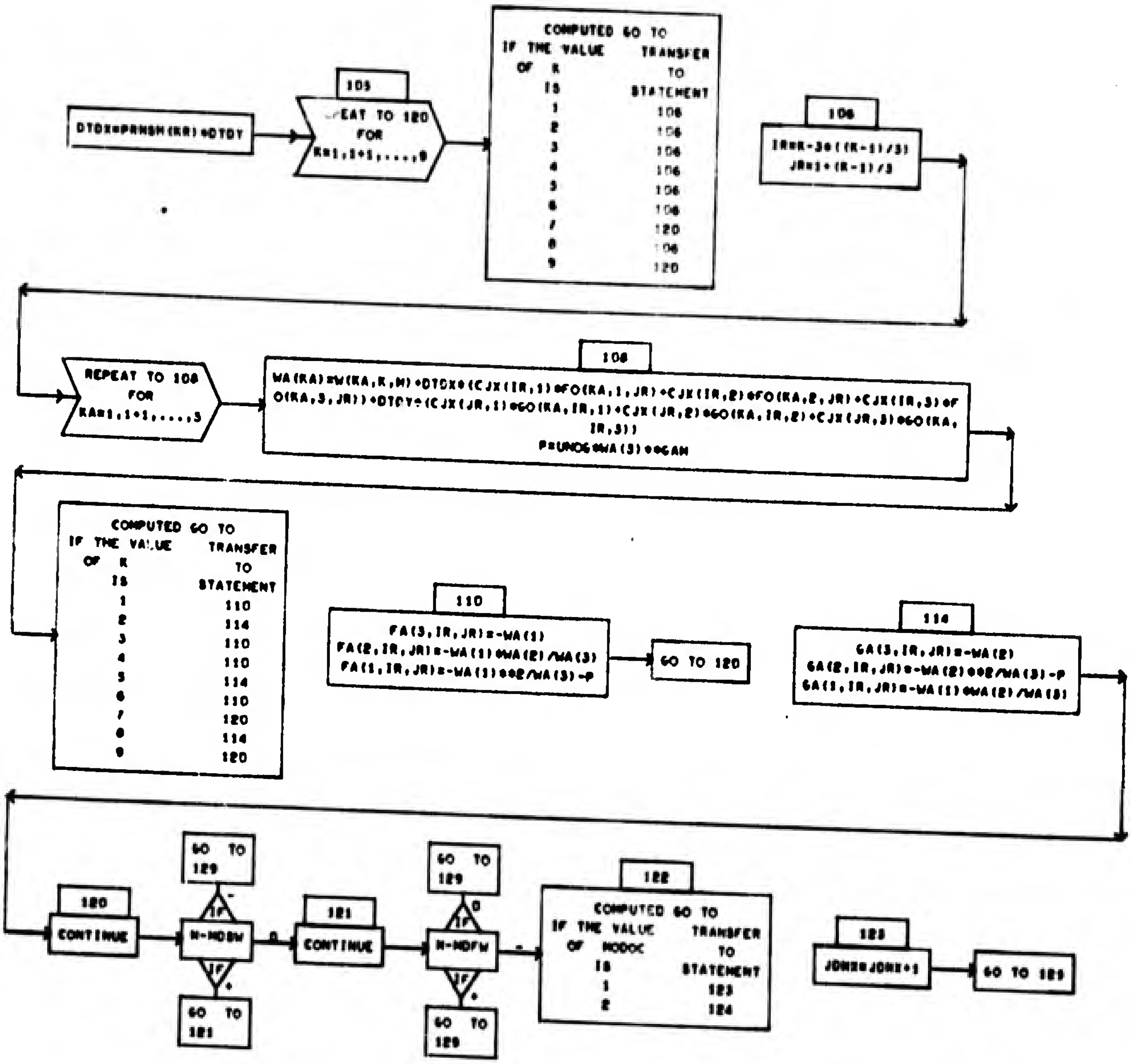
Detailed Flow Chart of Subroutine WOP



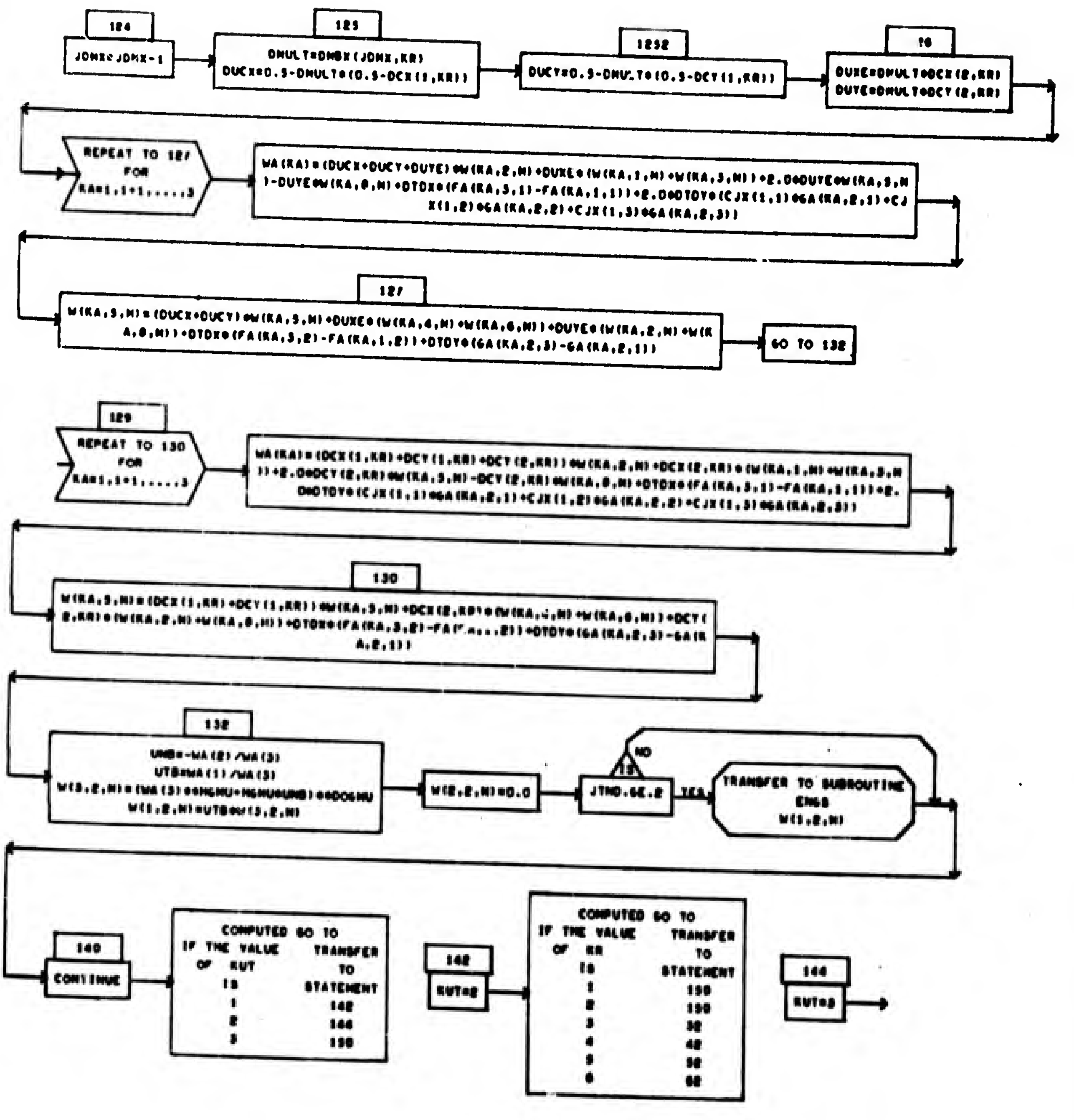
Detailed Flow Chart of Subroutine WOAP



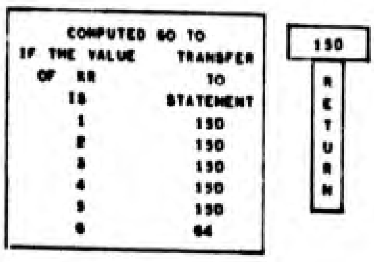
Detailed Flow Charts of Subroutine HERI (1 of 4)



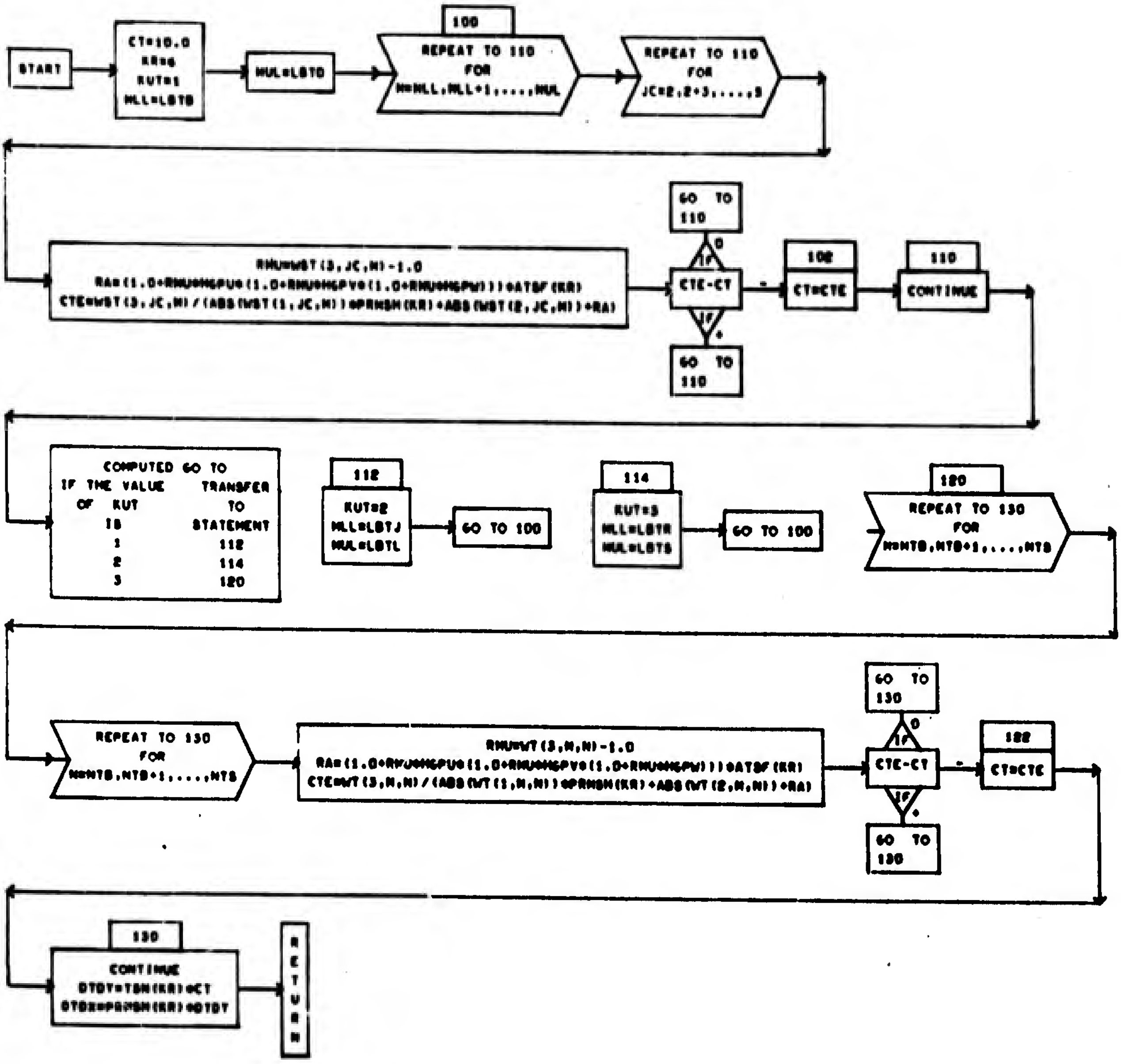
Detailed Flow Charts of Subroutine HERI (2 of 4)



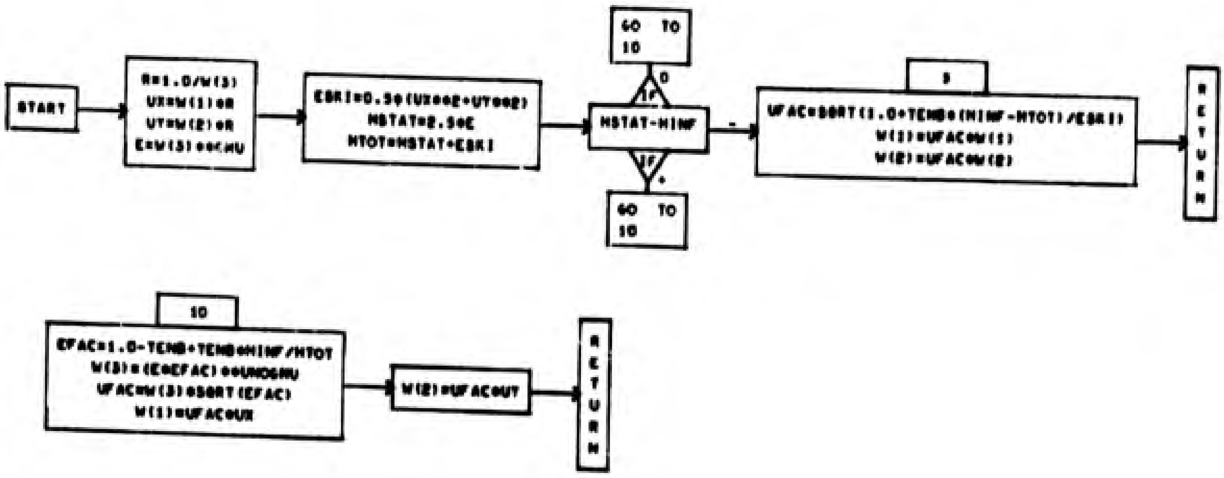
Detailed Flow Charts of Subroutine HERI (3 of 4)



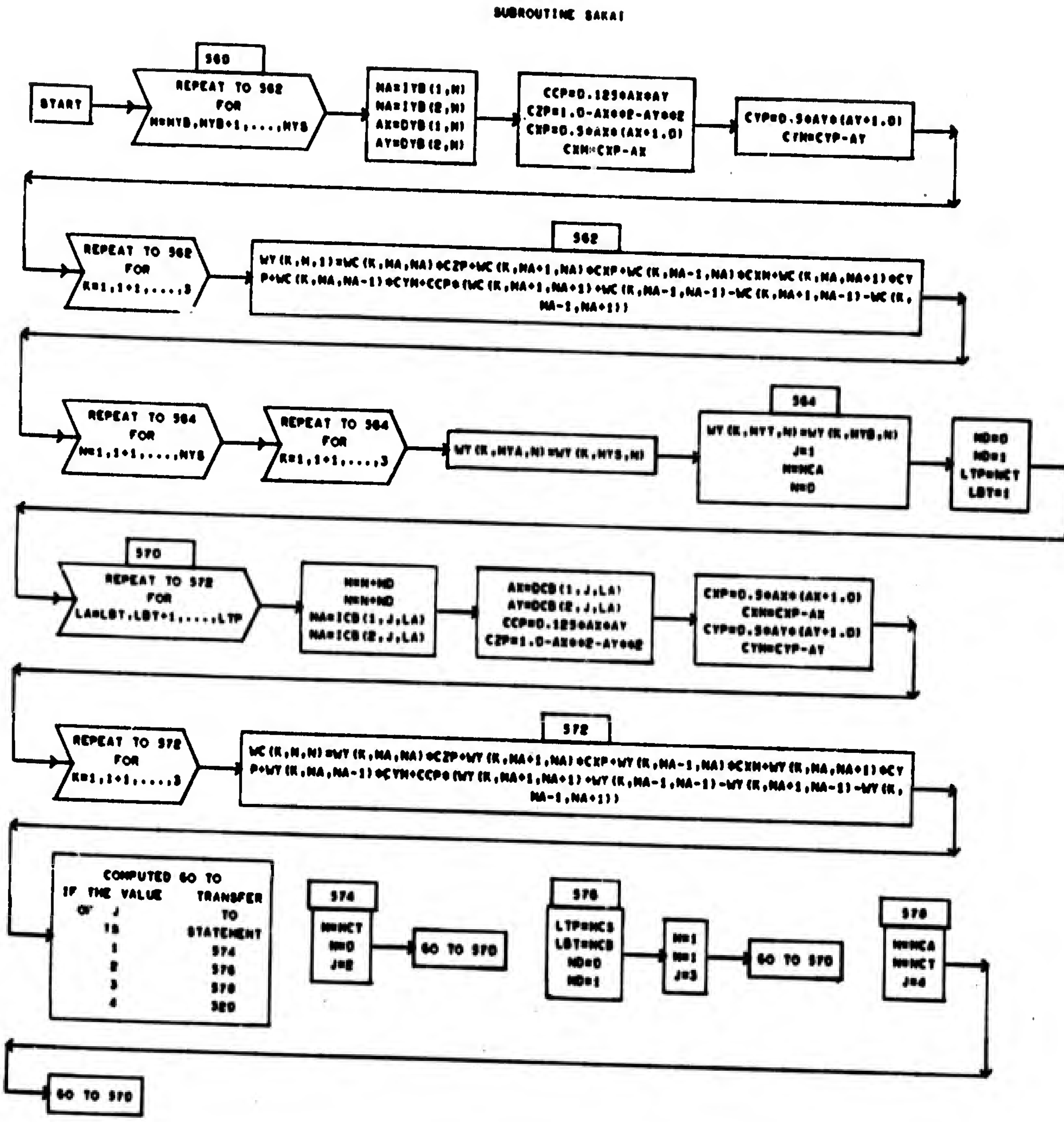
Detailed Flow Charts of Subroutine HERI (4 of 4)



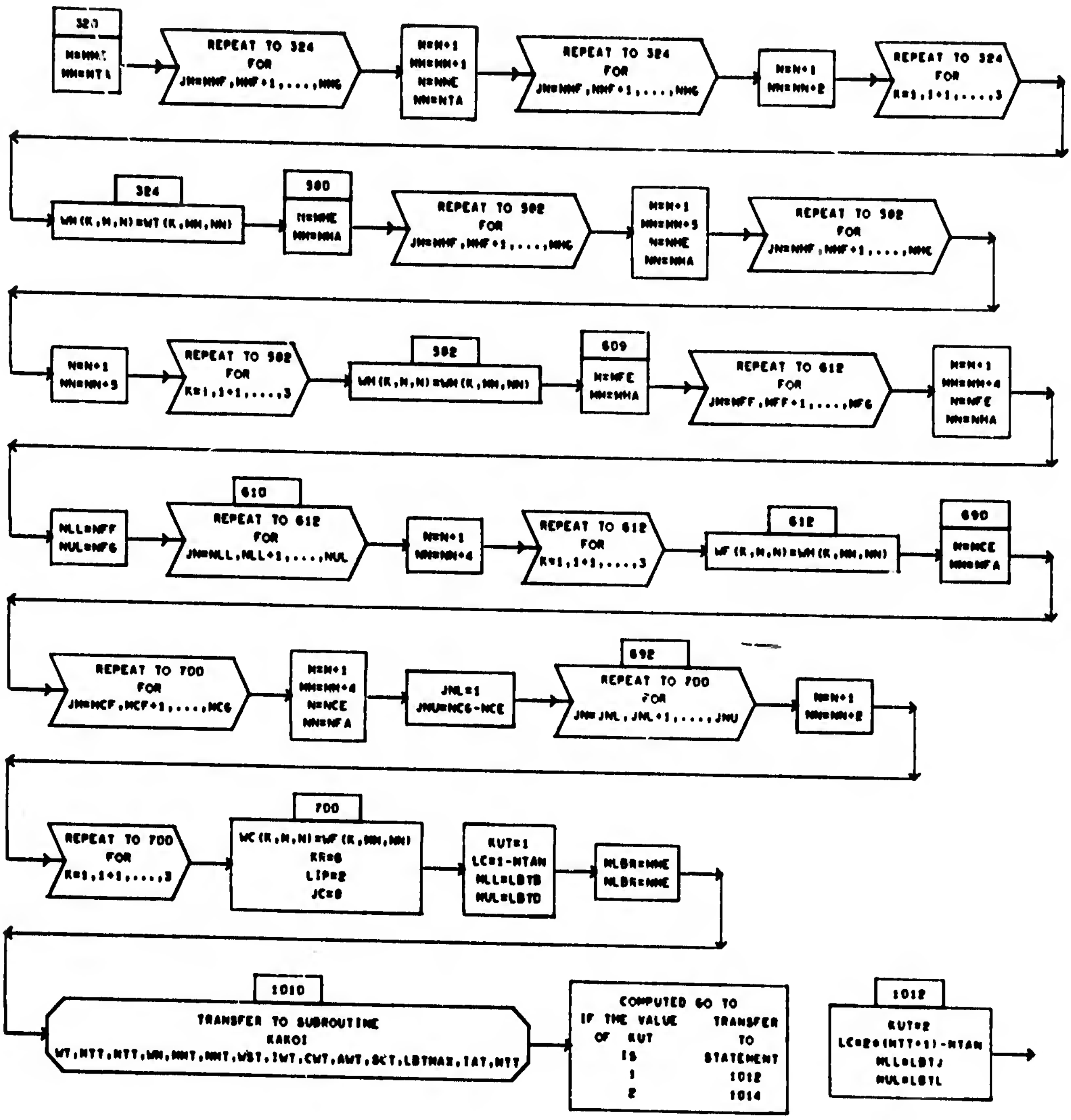
Detailed Flow Chart of Subroutine JIKAN



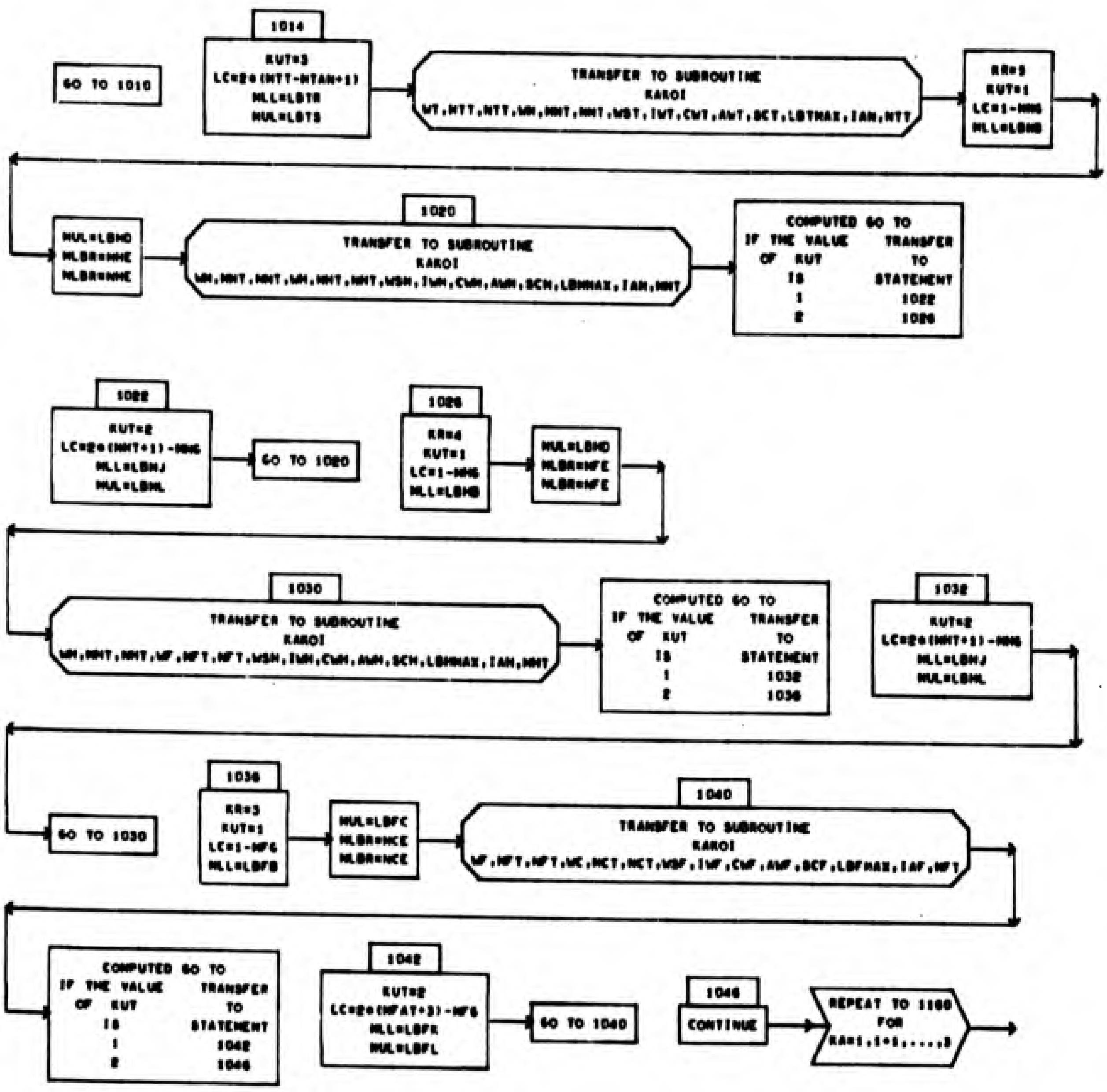
Detailed Flow Chart of Subroutine ENGS



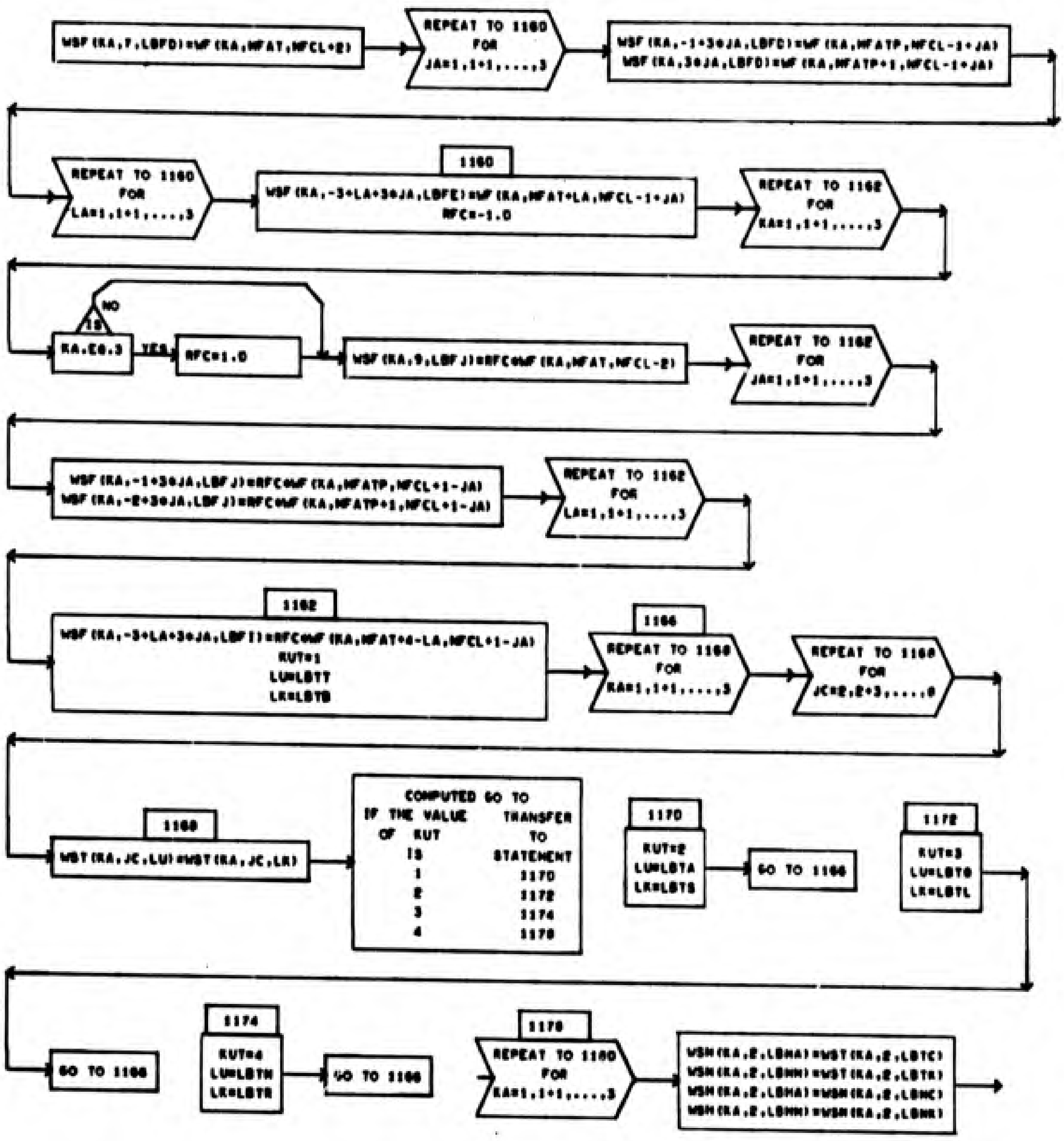
Detailed Flow Charts of Subroutine SAKAI (1 of 13)



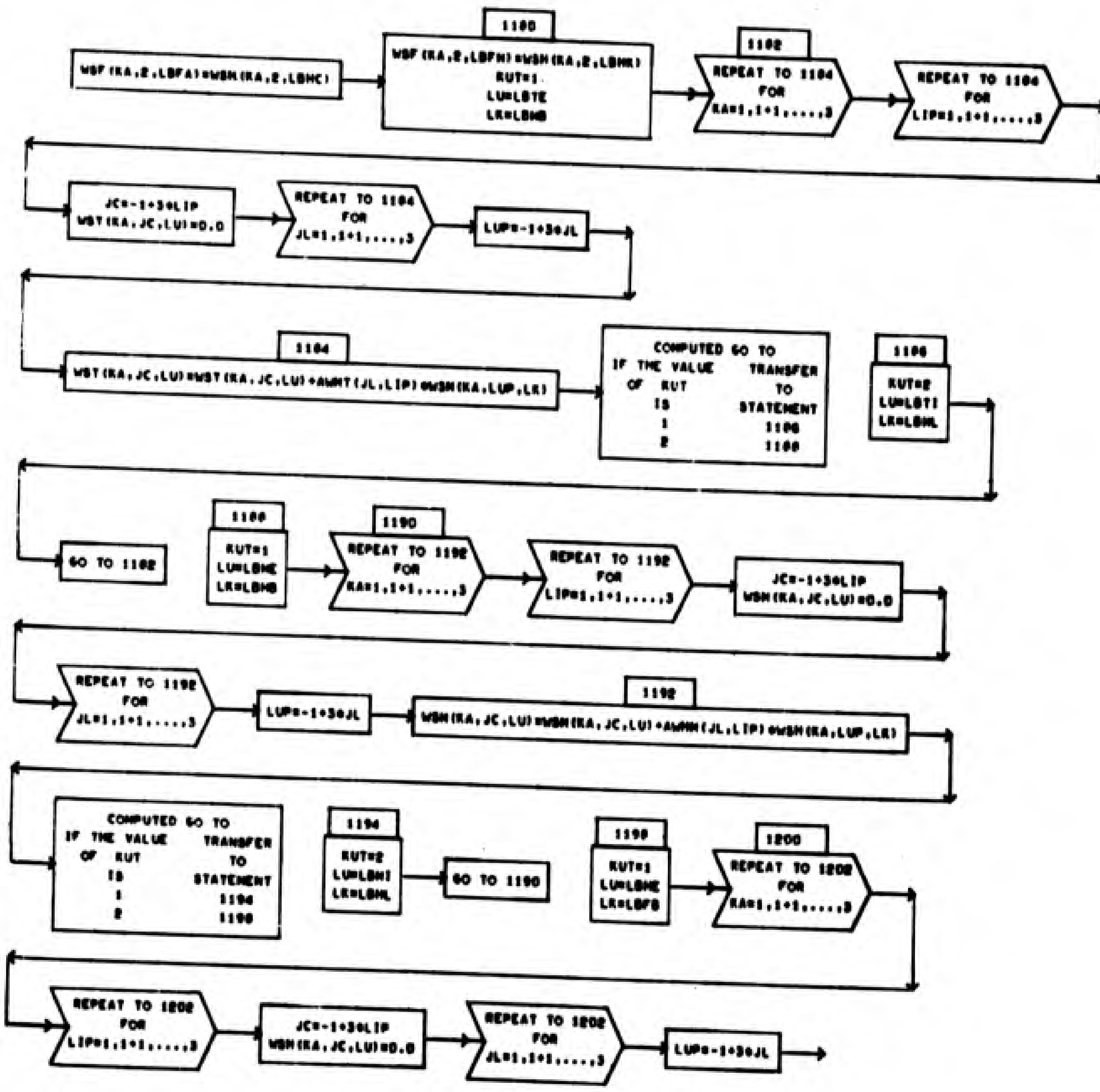
Detailed Flow Charts of Subroutine SAKAI (2 of 13)



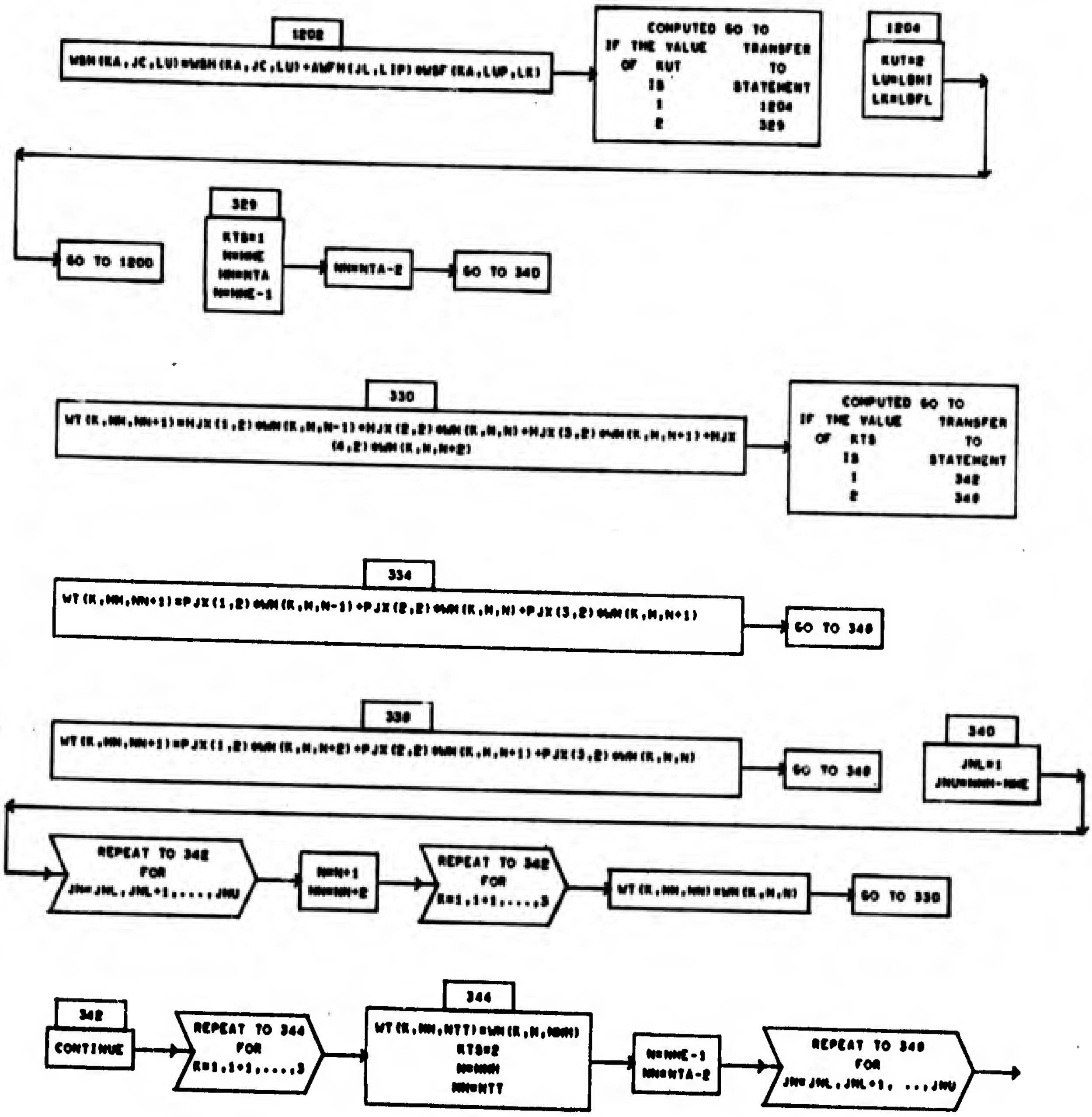
Detailed Flow Charts of Subroutine SAKAI (3 of 13)



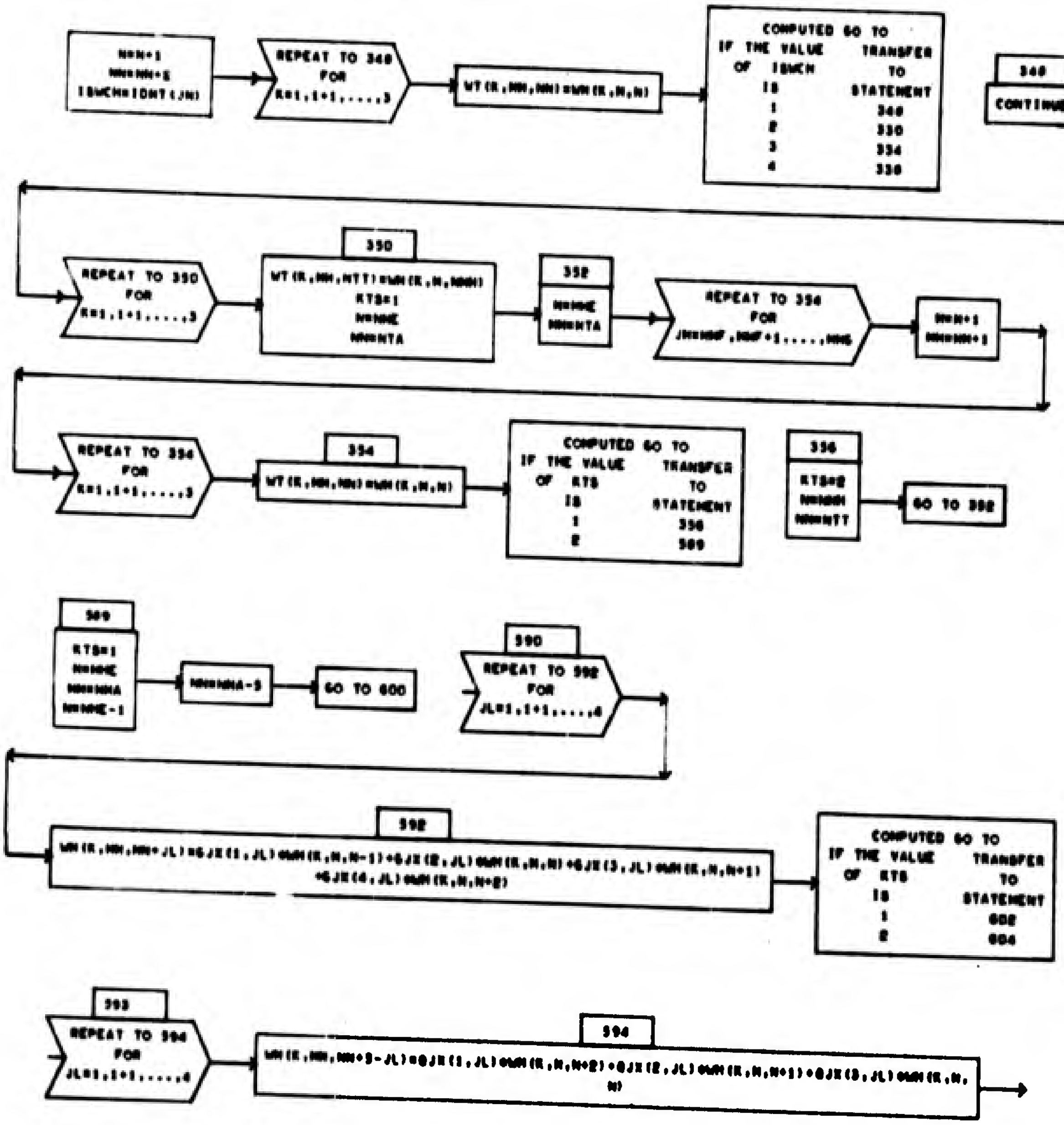
Detailed Flow Charts of Subroutine SAKAI (4 of 13)



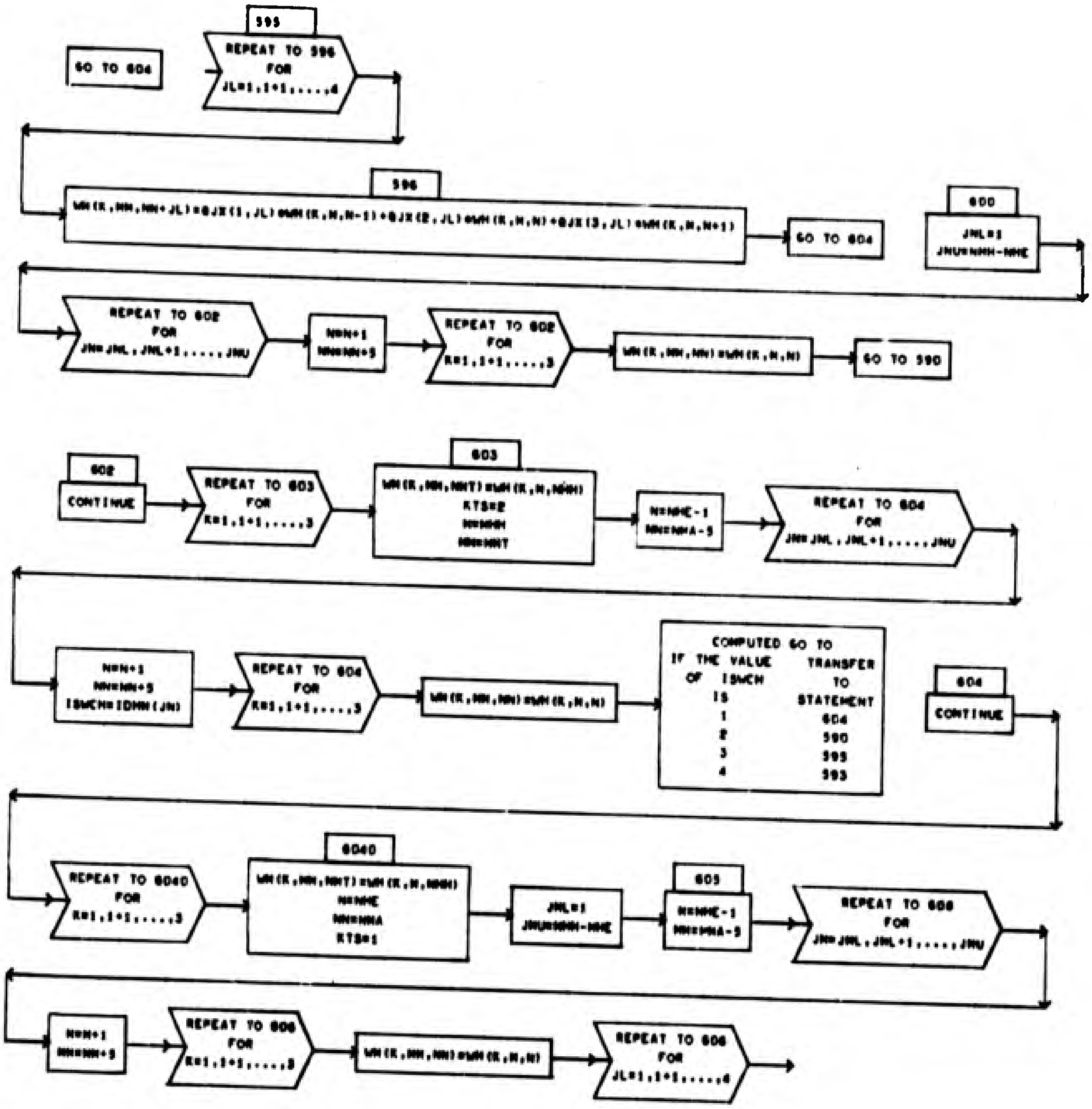
Detailed Flow Charts of Subroutine SAKAI (5 of 13)



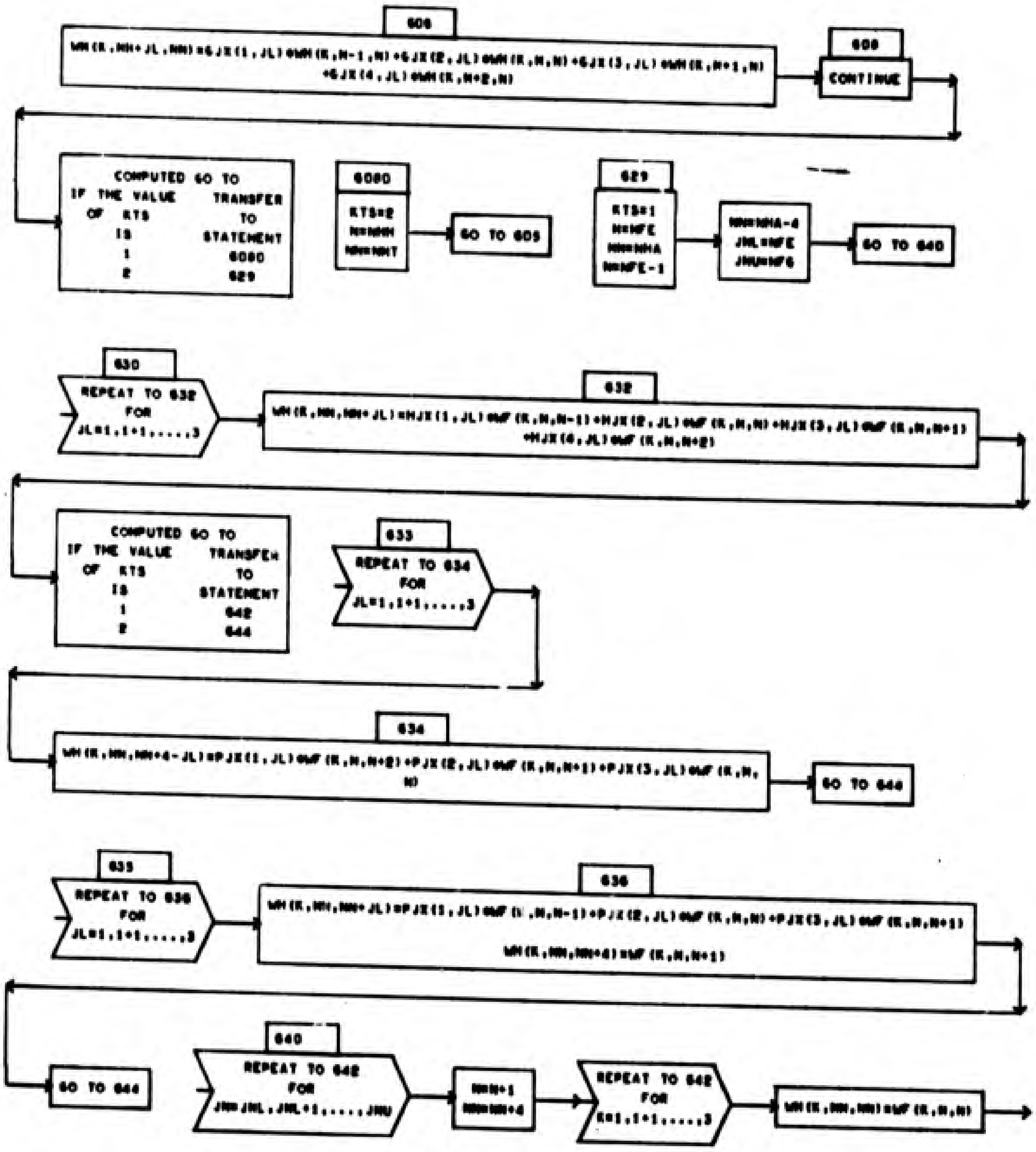
Detailed Flow Charts of Subroutine SAKAI (6 of 13)



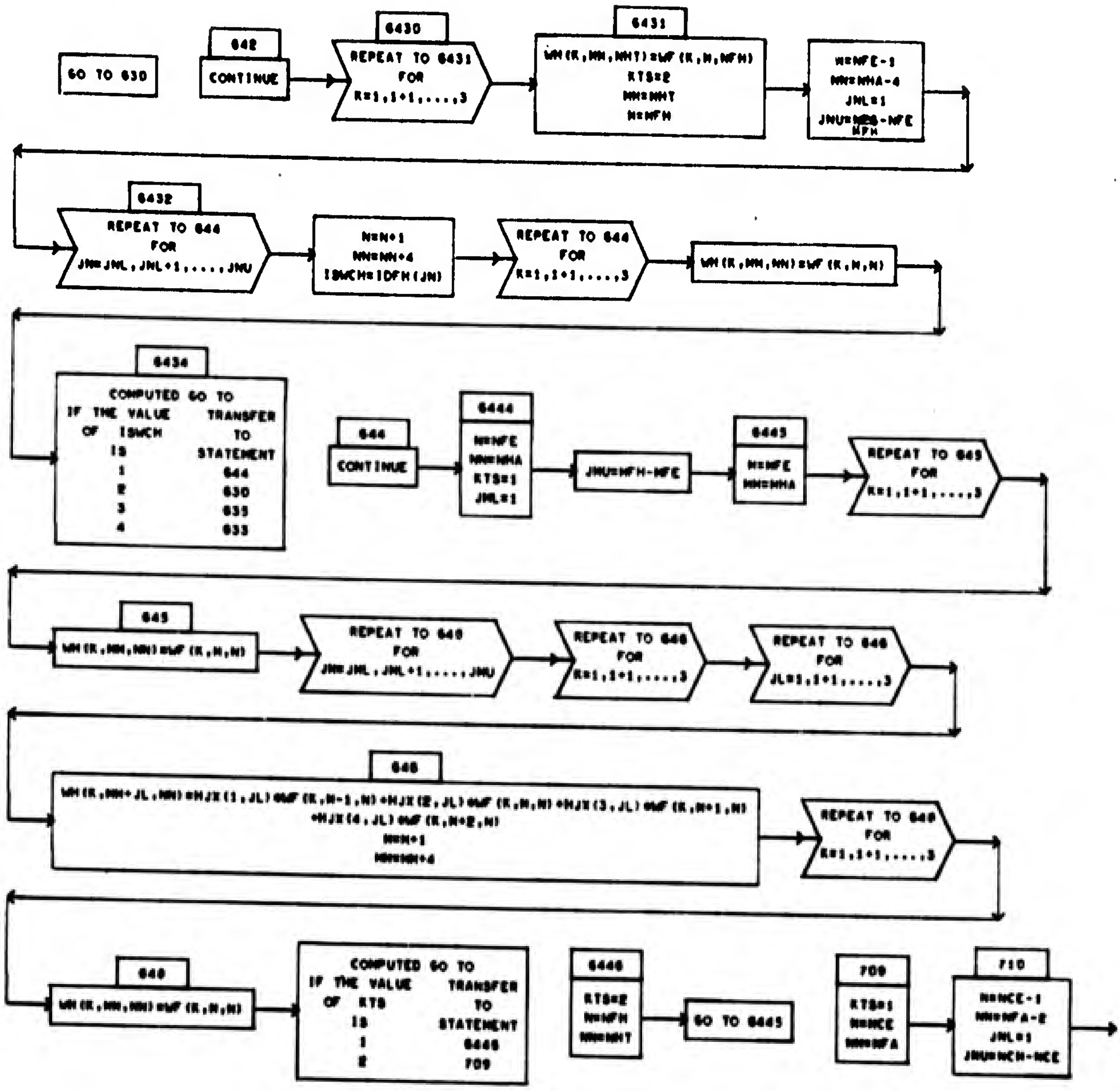
Detailed Flow Charts of Subroutine SAKAI (7 of 13)



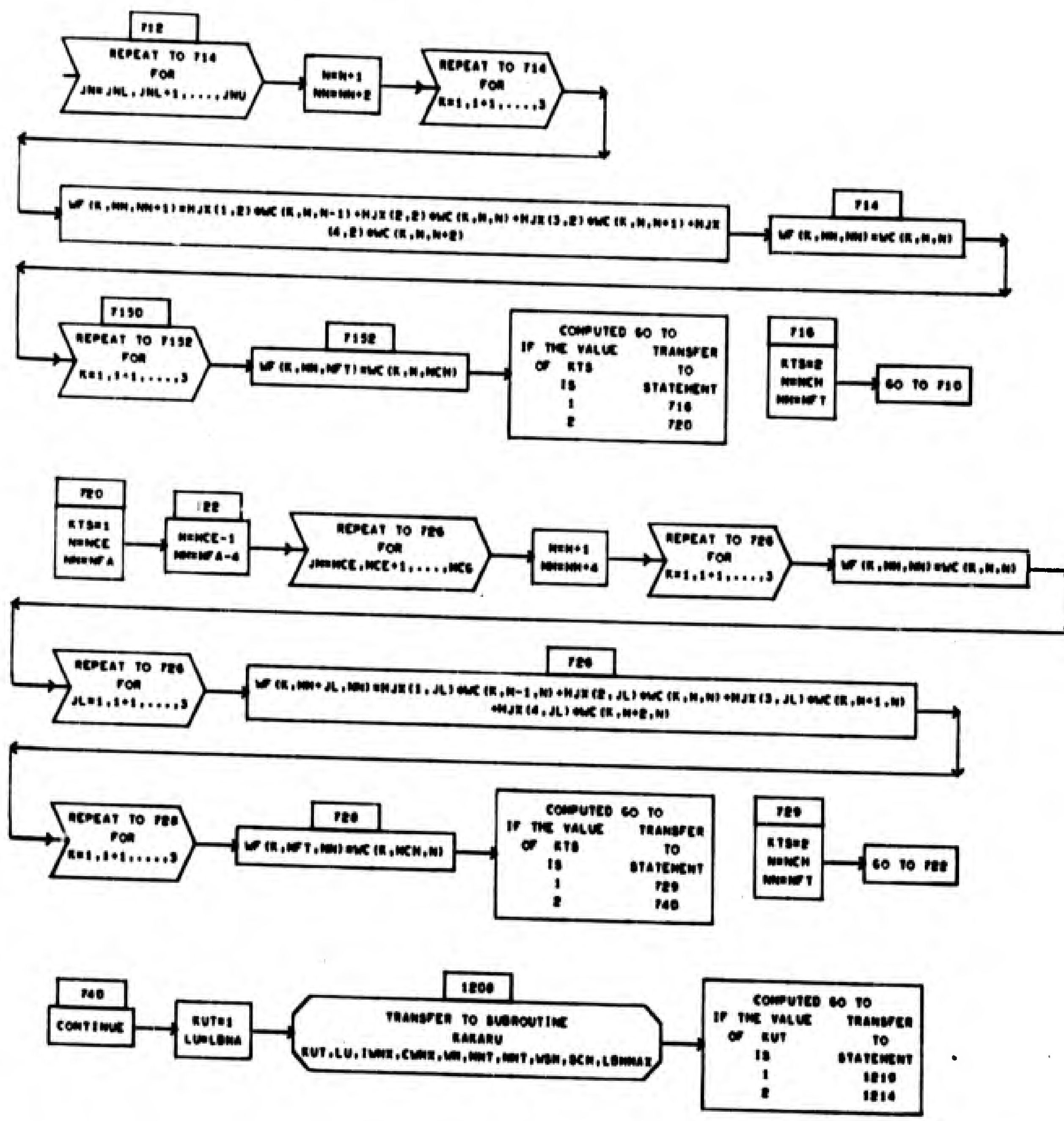
Detailed Flow Charts of Subroutine SAKAI (8 of 13)



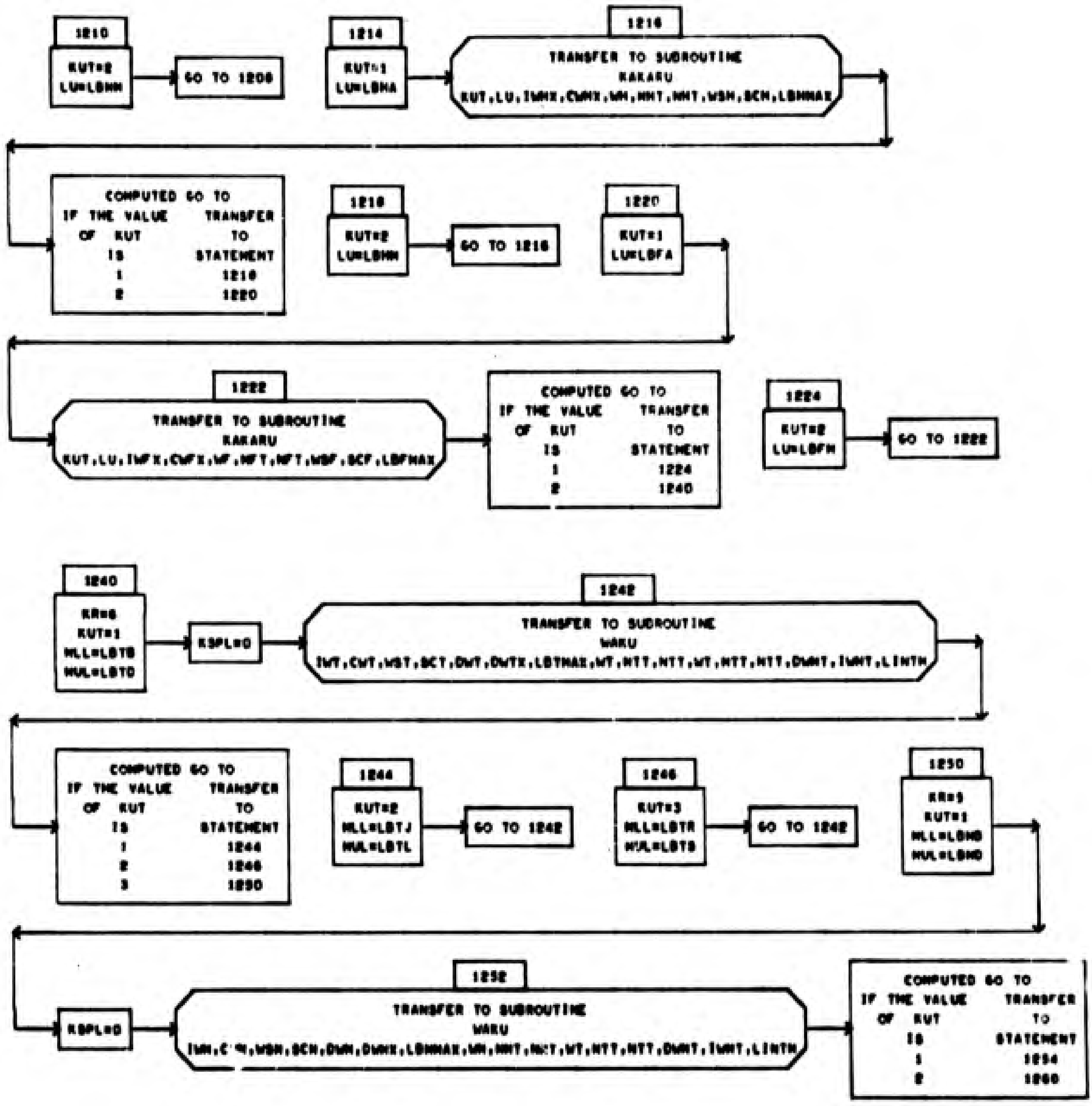
Detailed Flow Charts of Subroutine SAKAI (9 of 13)



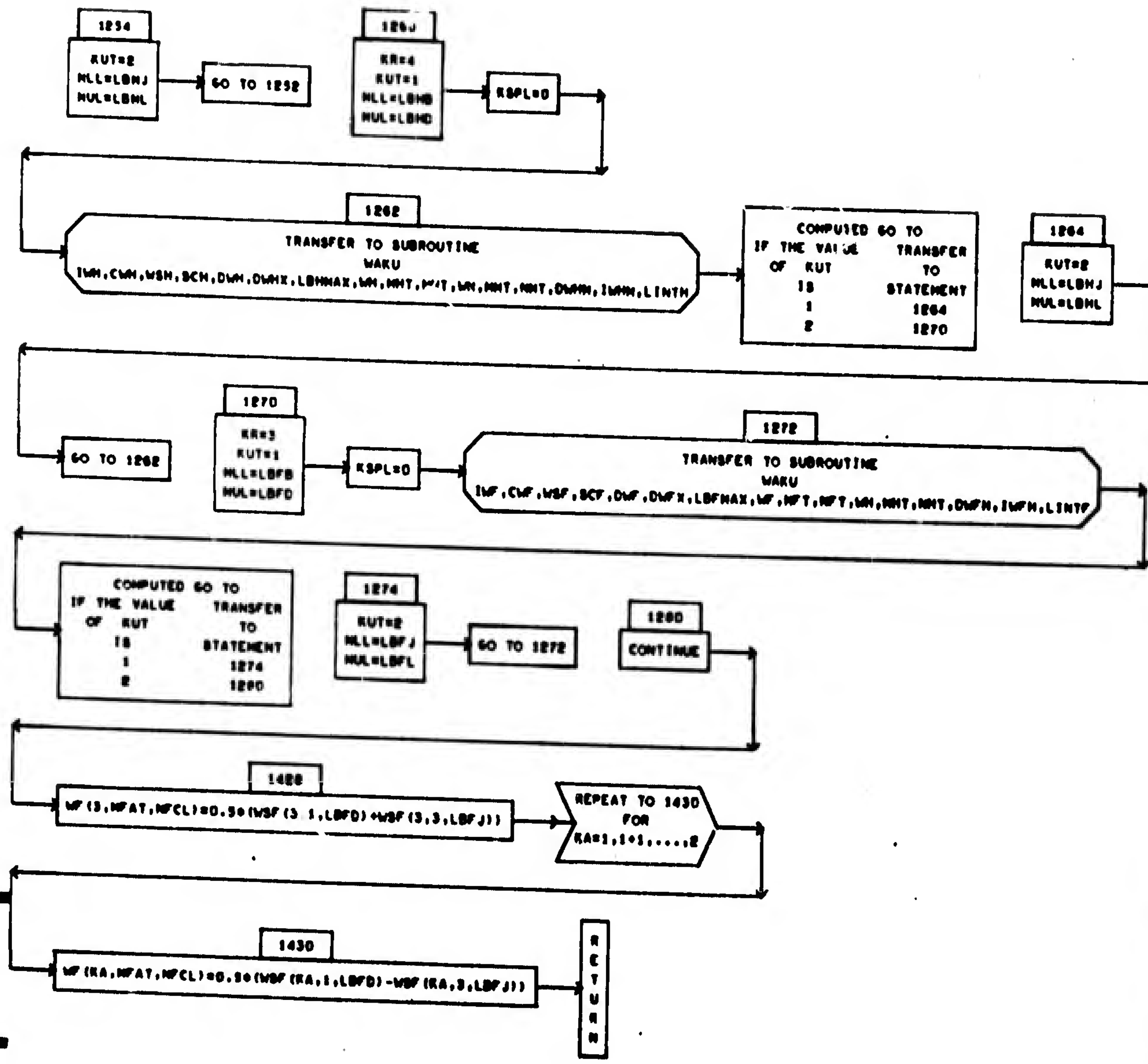
Detailed Flow Charts of Subroutine SAKAI (10 of 13)



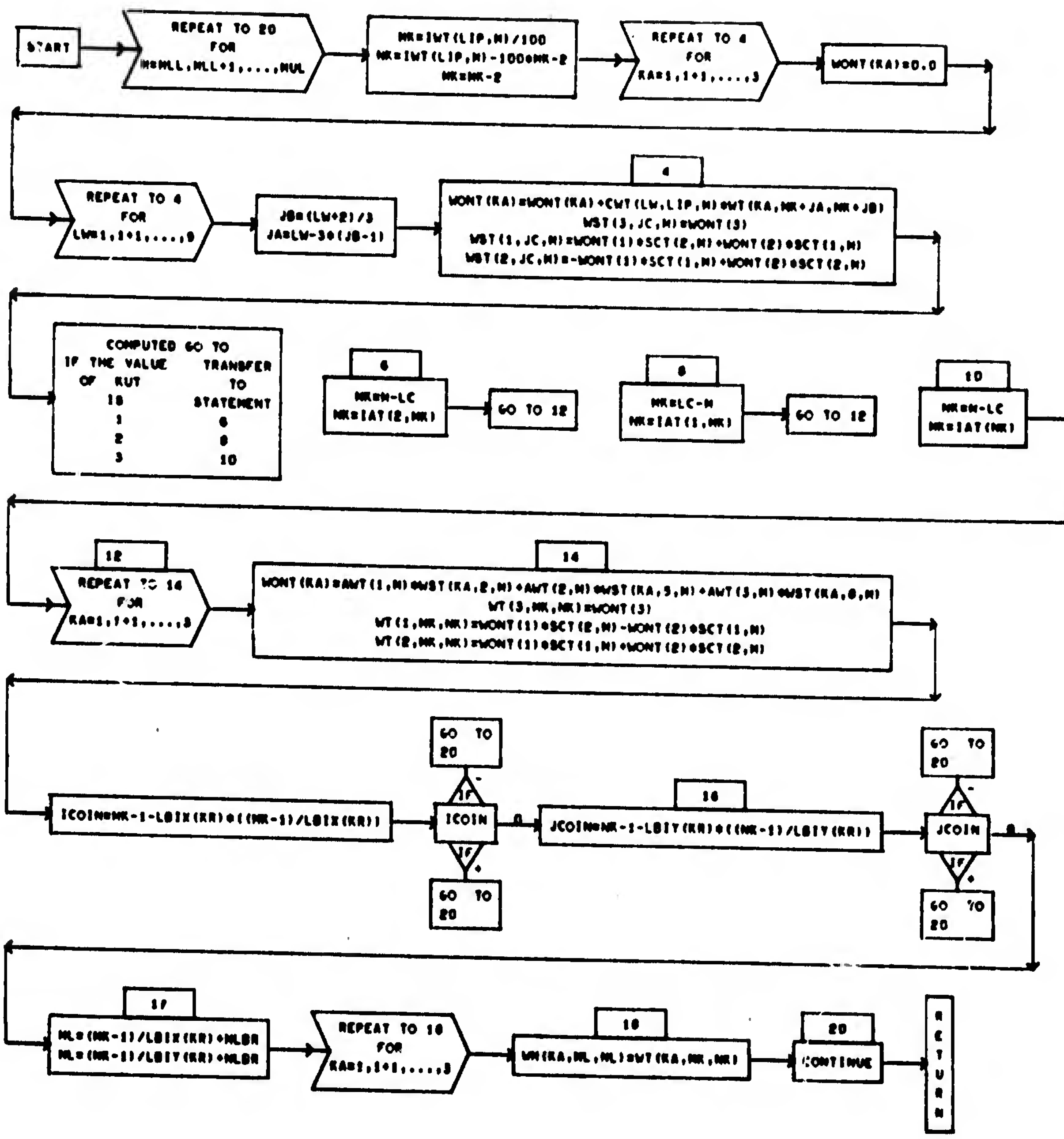
Detailed Flow Charts of Subroutine SAKAI (11 of 13)



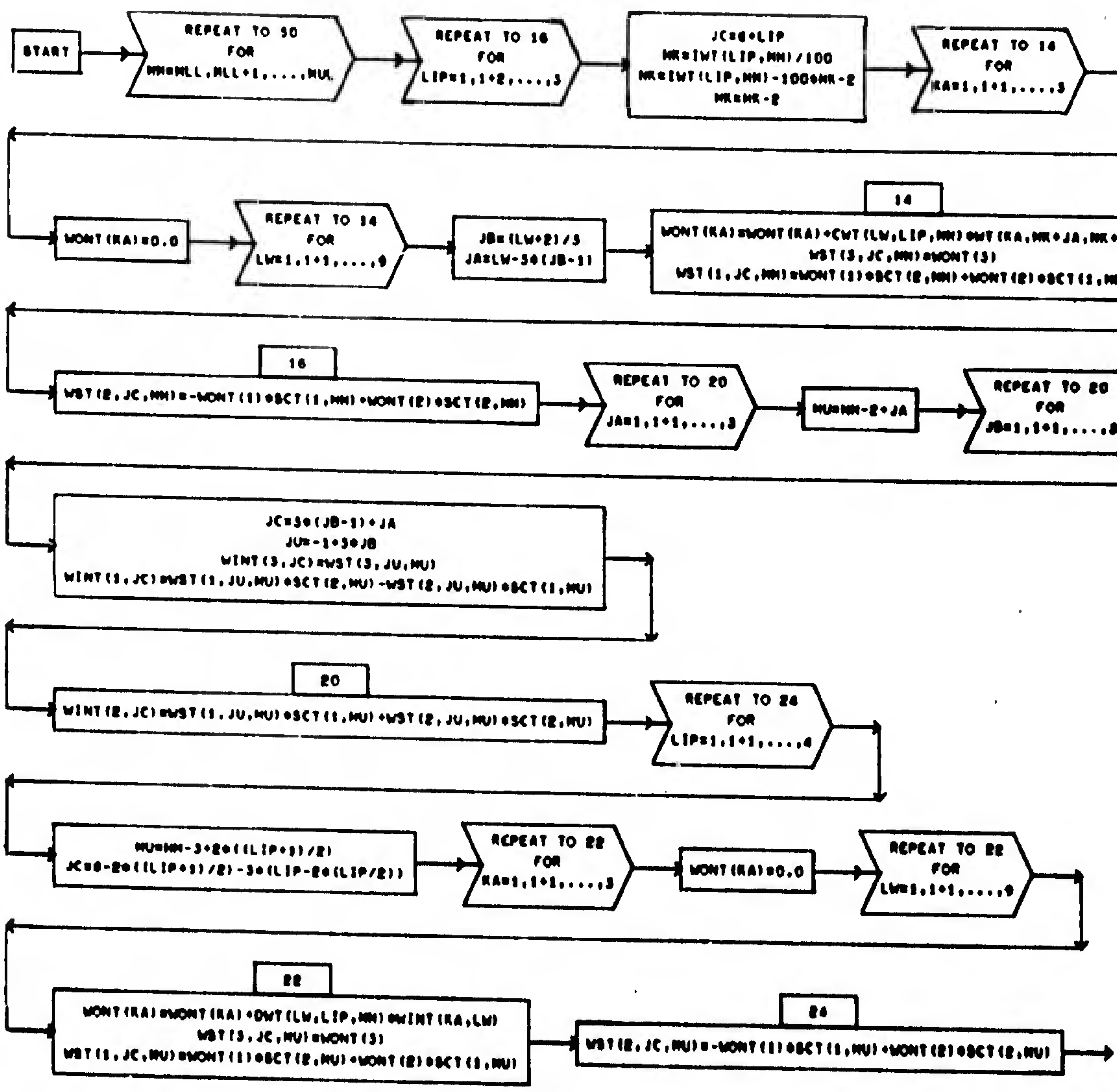
Detailed Flow Charts of Subroutine SAKAI (12 of 13)



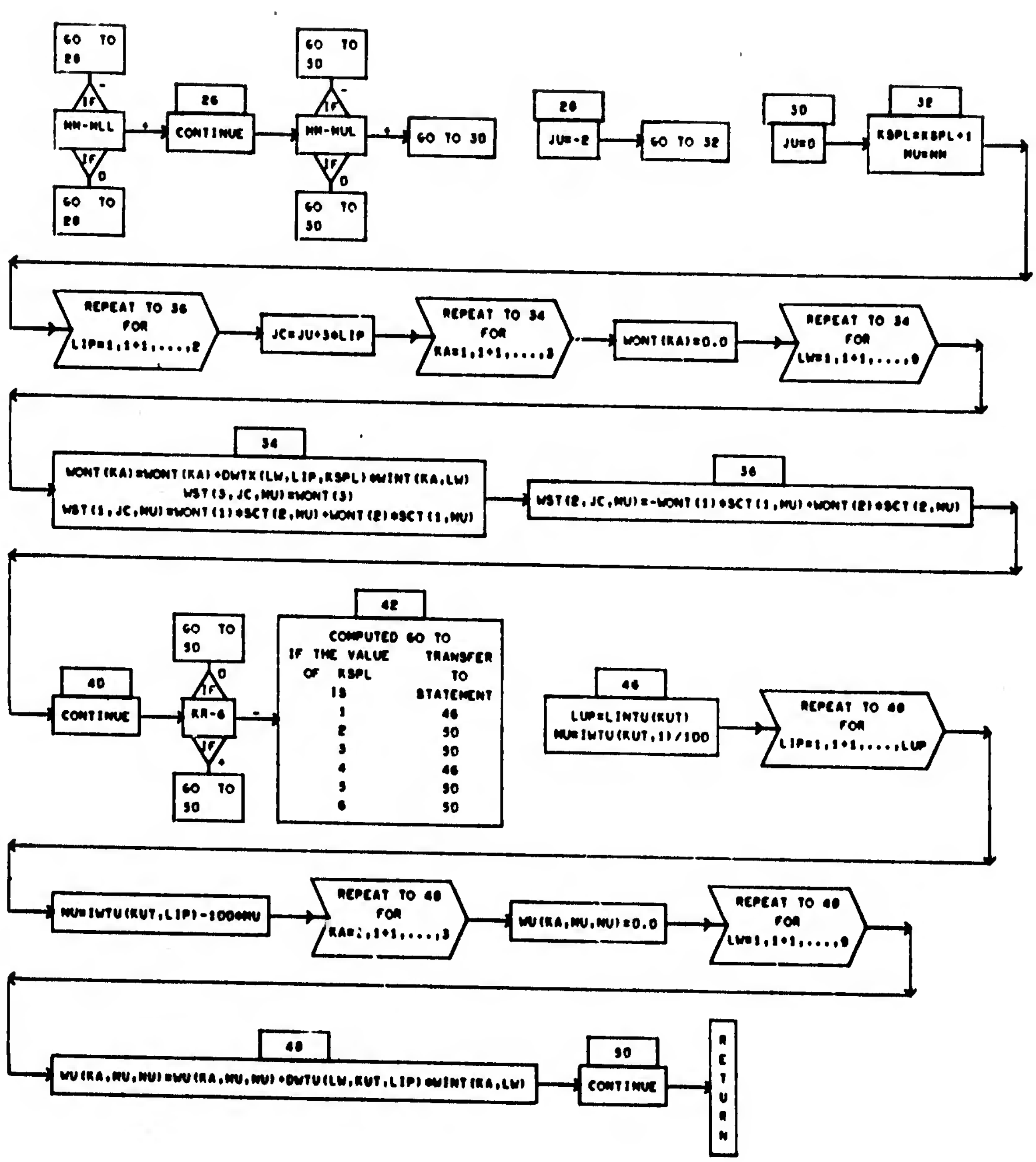
Detailed Flow Charts of Subroutine SAKAI (13 of 13)



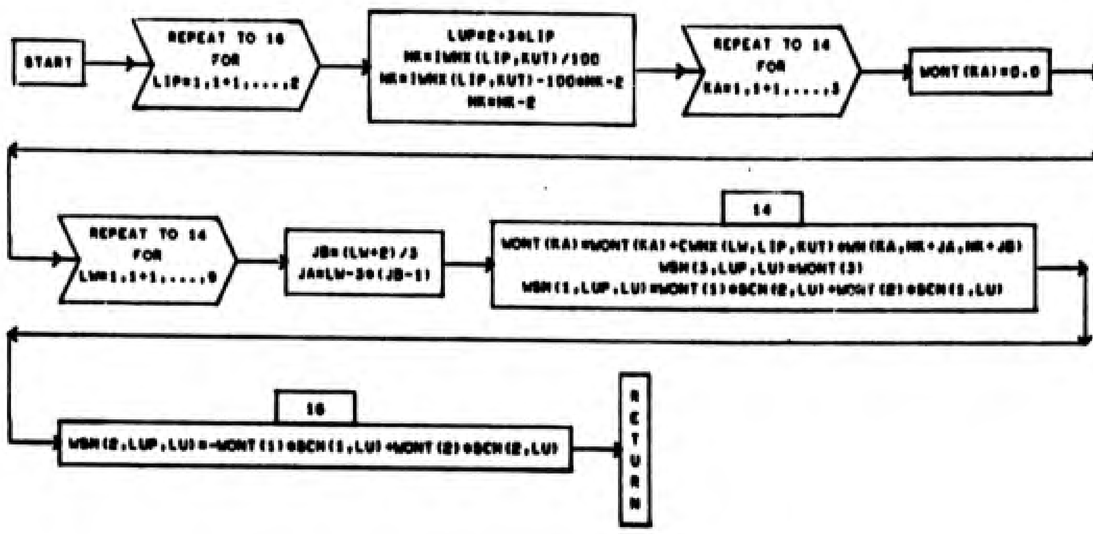
Detailed Flow Chart of Subroutine KAKOI



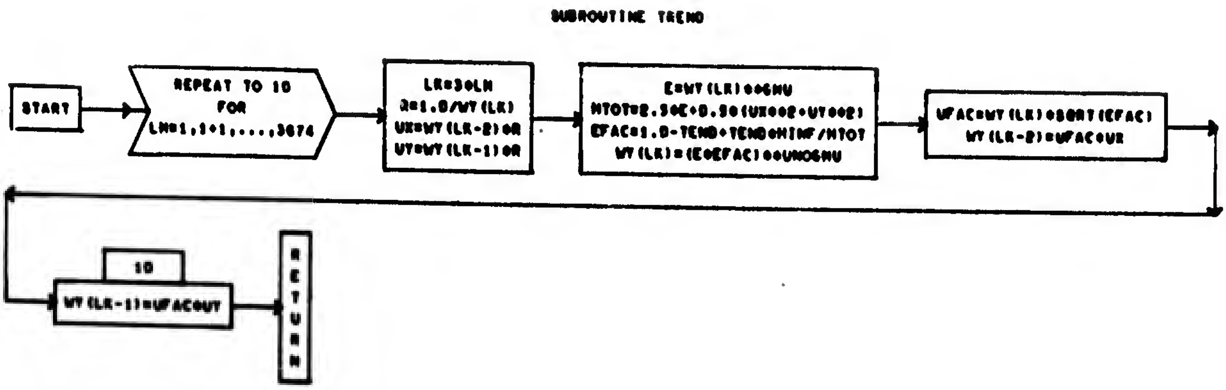
Detailed Flow Charts of Subroutine WAKU (1 of 2)



Detailed Flow Charts of Subroutine WAKU (2 of 2)



Detailed Flow Chart of Subroutine KAKARU



Detailed Flow Chart of Subroutine TREND

APPENDIV VI

BOUNDARY LAYER PROGRAM LISTING

```
      PROGRAM TURBL(INPUT,OUTPUT,TAPE6=OUTPUT)
      COMMON/COEF/  Z(20),NO,ISIGN,IKIND,TO,PO,IPAR
      COMMON/XSKIN/ ISKIN,DELK,CF,H ,CD ,FODEL,IFE,ICF,HSEP,HREAT
      COMMON/XYZ/ UE, DUEDX, PE, TE, ME, RHOE, VO,IN,VM,RHOM ,SINB
      COMMON/COD/GAM12,INCR,GGAM1,GAM,AI,ICOMP,IU  ,IDE
      COMMON/XOPT/ DISX(75),UINF(75),PINF(75),MINF(75),TINF(75),BETA(75)
      COMMON/FREE/ XIIF,XMIF,QIF,PIF,AIF,UIF,AIF2,UIF2,GXGAM1,CON37,
     1  XSTART, DELX, XEND
      COMMON/PUO/ ILAM, HSTART, XVIS, C
      COMMON/SAVEIT/ QUABI(7,100)
      COMMON XI(75)  ,UEI(75)
      REAL ME, TP, ISTARP, MINF
      DIMENSION X(3), DEL1(3), DEL2(3), DEL3(3), H12(3), H32(3)
     *, CP(75) , CFF(3) ,WORD(10), ETO(3), AFACT(3)
     * , CAPY(3), CAPX(3)  ,XP(3)
C
C
C+++++FLAGS FOR PROGRAM
C
C-----ILAM = 1 CALL FOR A LAMINAR BOUNDARY LAYER STARTING FROM THE
C      STAGNATION POINT.
C-----INCR = NOT USED IN PRESENT VERSION
C-----IBUG = CALL FOR PRINTOUT AT EACH ITERATION
C-----IPROF .EQ.1, READ IN CP DISTRIBUTION AND CALCULATE U DISTRIBUTION
C-----ICOMP = 0 INCOMPRESSIBLE
C-----ICOMP = 1  COMPRESSIBLE
C-----IDE = 1 USE INCOMPRESSIBLE DIFFERENTIOL EQUATIONS
C-----IDE = 2 USE COMPRESSIBLE EQUATIONS, MAKE SURE  TO USE COMPRESSIBLE
C-----          VARIABLES
C-----ISEP AND ISKIN = 1  POWER LAW VELOCITY PROFILE
C-----ISEP AND ISKIN = 2 YOSH-S CURVE FIT TO SEDDON-S H = H(H1)
C-----ISEP = ISKIN = 3 SEPARATED FLOW
C-----ISEP = ISKIN = 4 OUR SHIFT OF GREEN-S REATTACHMENT PROFILE
C-----IFE = 1 USE HEAD S INCOMPRESSIBLE EXPRESSION FOR ENTRAINMENT FUNCTION
C-----IFE = 2 USE GREEN S COMPRESSIBLE EXPRESSION
C-----ICF = 1 USE LUDWIEG-TILLMANN
C-----ICF = 2 USE ALPHA A FUNCTION OF HSEP
C-----ITER = LIMIT OF ITERATIONS TO BE ATTEMPTED
C
C+++++TYPE OF CURVE FIT TO DATA FOR DERIVATIVES AND INTERPOLATED VALUES
C
C-----IPAR = 1, USE A PARABOLIC FIT OF INPUT POINTS FOR THE THREE
C          POINTS NEAREST THE DESIRED VALUE OF X.   ***BE SURE TO SET NO=0*****
C-----NO =  DEGREE OF POLYNOMIAL TO BE FITTED. THIS FITS THE ENTIRE
C          VELOCITY DISTRIBUTION SO BE CAREFUL WITH COMPLEX CURVES.
C          MAX. VALUE OF NO = 19
C-----IKIND  = 2, PROGRAM SET UP TO HANDLE THIS CASE  ONLY
C-----ISIGN = 1, NEGATIVE EXPONENTS
C              = 2 POSITIVE EXPONENTS
C
C+++++FLOW TOTAL AND FREESTREAM CONDITIONS
C
C-----PO = TOTAL PRESSURE IN PSIA
C-----TO = TOTAL TEMPERATURE IN DEGREES R
C
```

```
C*****QUANITIES OUTSIDE THE BOUNDARY LAYER
C
C-----UINF(I) = VELOCITY OUTSIDE THE BOUNDARY LAYER IN FPS
C-----MINF(I) = MACH NUMBER OUTSIDE THE BOUNDARY LAYER
C-----PINF(I) = PRESSURE OUTSIDE THE BOUNDARY LAYER IN PSIA
C*****SET PINF(I) = 0.0     IF ONLY THE VELOCITY DISTRIBUTION IS KNOWN
C-----TINF(I) = TEMPERATURE OUTSIDE THE BOUNDARY LAYER IN DEGREES R
C
C*****STARTING CONDITIONS
C
C-----XSTART = STARTING VALUE OF X IN FEET
C-----XEND = FINAL VALUE OF X IN FEET
C***** BOUNDARY LAYER THICKNESSES - FOR STARTING VALUES, USE VALUES
C      CORRESPONDING TO FLAG IDE
C      USE COMPRESSIBLE VALUES OF BOUNDARY LAYER THICKNESSES, DISTANCES
C      AND VELOCITIES
C-----DELX = DESIRED STEP SIZE IN X DIRECTION
C-----DELK = DISPLACEMENT THICKNESS IN FEET
C-----THETAO = MOMENTUM THICKNESS IN FEET
C-----DEL30 = MASS FLOW THICKNESS IN FEET
C-----ERROR = LIMIT FOR THE ITERATION SCHEME
C-----DISX(I) = LOCATION ALONG SURFACE OF DATA POINT IN FEET
C-----CP(I) = PRESSURE COEFFICIENT
C-----BETA(I) = LOCAL MACH ANGLE OUTSIDE THE BOUNDARY LAYER IN RADIANS
C-----GAM = RATIO OF SPECIFIC HEATS
C-----DEL1 = DISPLACEMENT THICKNESS
C-----DEL2 = MOMENTUM THICKNESS
C-----DEL3 = BOUNDARY LAYER THICKNESS MINUS DEL1
C-----XMIF = FREE STREAM MACH NUMBER
C-----XTIF = FREE STREAM TEMPERATURE IN R
C-----QIF = FREE STREAM DYNAMIC PRESSURE IN PSI
C-----DELK = COMPRESSIBLE DISPLACEMENT THICKNESS
C-----HSTART = INCOMPRESSIBLE H
C-----HSEP = SEPARATION INCOMPRESSIBLE SHAPE FACTOR
C-----HREAT = REATTACHMENT INCOMPRESSIBLE SHAPE FACTOR
C-----XVIS = LINEAR VISCOSITY RATIO CONSTANT
C-----C = CHORD, IN FEET
C
    1 CONTINUE
      INDEX = 0
      READ 13,  (WORD(I) I=1,8)
      READ 22, NO,IKIND, SIGN,IPAR
      READ 12, ILAM,XVIS, C
   12 FORMAT(I5,5X,7E10.3)
      READ 23, ISEP,ISKIN, IFF, ICF, HSEP, HREAT
      READ 2, INCR,IBUG,GAM,XSTART,XEND,DELX,DELK    ,HSTART,THETAO,DEL30
      READ 7, PQ,TO,ERROR,ITER,IPROF,ICOMP
     * ,IDE
      READ 50,XMIF,XTIF,PIF
C
C
      PRINT 16,(WORD(I) ,I=1,8)
      IF(ILAM.EQ.1) PRINT 17
   17 FORMAT(//* WE WILL START WITH A LAMINAR BOUNDARY LAYER*)
      PRINT 4, INCR,IBUG,GAM,XSTART,XEND,DELX,DELK    ,HSTART,THETAO,DEL30
```

```
      PRINT 18, XVIS, C
   18 FORMAT(* XVIS = * F13.5 * C = * F10.5)
      IF(NO.NE.0) GO TO 32
      PRINT 31
      GO TO 36
   32 PRINT 33, NO
   36 PRINT 37, ISEP,ISKIN,IFE
      GAM12 =  0.5*(GAM - 1.0)
      GGAM1 = GAM/(GAM - 1.0)
      QIF = PIF*0.5*GAM*XMIF**2
      PO = PIF*(1.0 + GAM12*XMIF**2)**GGAM1
      TO =XTIF*(1.0 + GAM12*XMIF**2)
      AIF = 49.0*SQRT(XTIF)
      UIF = XMIF*AIF
      AIF2 = AIF**2/GAM12
      UIF2 = UIF**2
      GXGAM1 = 1.0/GGAM1
      CON37 = 0.5*GAM*XMIF**2
      IF(IPROF.NE.1) GO TO 11
      READ 51,IO,(DISX(I),CP(I),BETA(I),I=1,IO)
      DO52 I=1,IO
   52 UINF(I) = SQRT(UIF2 - AIF2*((1.0 + CON37*CP(I))**GXGAM1 - 1.0))
      GO TO 53
C
C
   11 CONTINUE
      READ 3, IO,(DISX(I),UINF(I),PINF(I),MINF(I),TINF(I),
     1 BETA(I),I=1,IO)
   53 CONTINUE
      AT = 49.0*SQRT(TO)
      IF(NO.GT.1) CALL GAUSSEN(IO,NO,IKIND,ISIGN,DISX,UINF,WORD,Z)
      PRINT 8, PO,TO ,ERROR,ITER,IPROF
      IF(DELX.LE.0.0) GO TO 200
      RHOO = PO*144.0/(53.3*TO)
      VO = VISCOS(TO)/RHOO
      IF(PINF(1) .GT.0.0) GO TO 6
      DO 9 I = 1,IO
      UAT = UINF(I) /AT
      UAT2 = UAT**2
      TEMP = 1.0 - GAM12*UAT2
      TINF(I) = TO*TEMP
      PINF(I) = PO*TEMP**GGAM1
    9 MINF(I) = SQRT(UAT2/TEMP)
    6 CONTINUE
      PRINT 5,(DISX(I),UINF(I),PINF(I),MINF(I),TINF(I),BETA(I),I=1,IO)
      IF(ILAM.EQ.1) GO TO 7999
      IF(IDE.EQ.2) GO TO 8000
 7999 CALL TRANS(DELX,      THETAO,DEL30,HSTART,XSTART)
C-----STARTING WITH A LAMINAR BOUNDARY LAYER
      CALL LAMINAR(THETAO,XSTART,DEL30)
C-----RETURNS COMPRESSIBLE BOUNDARY LAYER THICKNESSES
 8000 CALL VEL(XSTART)
      X(1) = XSTART
      X(3) = XSTART
      XP(1) = XP(3) = XSTART
```

```
      RUM = RHOE*UE*SINB
      DO 10 I = 1,3
      DEL1(I) = DELK
      DEL2(I) = THETAO
      DEL3(I) = DEL30
      H12(I) = HSTART
      CAPY(I) = THETAO * RUM *UE
      CAPX(I) = RUM * DEL30
   10 H32(I) = DEL30/DEL2(I)
      CFF(1) = SKINF(ISEP,H12(1),H32(1),   DEL2(1),DEL1(1))
      REDEL2 = UE*DEL2(3)/VO
      CALL SAVEO(INDEX,X,DEL1,DEL2,DEL3,H12,H32,CF)
      IF(IBUG.GT.0) PRINT 102, X(3), DEL1(3), DEL2(3), DEL3(3),H12(3),
     * H32(3), UE, TE, ME, CF, CD, REDEL2
     *,H,CFF(3),A,DUEDX
   15 CONTINUE
      IFLAG = 0
      ICOUNT = 0
      X(2) = X(1) + 0.5*DELX
      X(3) = X(1) + DELX
      X1 = X(1)     $ X3 = X(3)
      DEL1 = DELX
      XP(3) = X(3)
      IF(IDE.EQ.2) GO TO 8001
      CALL VEL(X(2))
      DEL1 = (TE/TO)**4*DELX
      XP(3) = XINCOM(X(3))
      X1 = XP(1) $X3 = XP(3)
C-----SECOND ORDER RUNGE-KUTTA OR TRAPEZOIDAL FORMULA
 8001 CONTINUE
      CALL SLOPED(X1  ,H12(1),H32(1),DEL1(1),DEL2(1),DEL3(1),DD2DX,DD3DX
     * )
   20 CONTINUE
      CALL SLOPED(X3  ,H12(3),H32(3),DEL1(3),DEL2(3),DEL3(3),D2DX,D3DX)
      TEMP2 = CAPY(1) + 0.5*DEL1*(DD2DX + D2DX)
      TEMP3 = CAPX(1) + 0.5*DEL1*(DD3DX + D3DX)
      TEMP4 = TEMP3/TEMP2 *UE
   70 CAPY(3) = TEMP2
      CAPX(3) = TEMP3
      H32(3) = TEMP4
      H12(3) = HFACT(ISEP,H32(3),HSEP)
      RUM = RHOE*UE*SINB
      TEMP5 = CAPX(3) / RUM
      IF(ABS((TEMP5 - DEL3(3))/DEL3(3)).LT.ERROR) IFLAG = 1
      DEL2(3) = CAPY(3)/RUM  /UE
      DEL3(3) = CAPX(3)/RUM
      DEL1(3) = H12(3)*DEL2(3)
      IF(IDE.EQ.2) DEL1(3) = DEL2(3) + H
      ICOUNT = ICOUNT + 1
      REDEL2 = UE*DEL2(3)/VO
      CFF(3) = SKINF(ISKIN,H12(3),H32(3),DEL2(3),DEL1(3) )
      B = FODEL
      IF(IBUG.GT.0) PRINT 102,X1(3), DEL1(3), DEL2(3), DEL3(3),H12(3),
     * H32(3), UE, TE, ME, CF, CD, REDEL2
     *,H,CFF(3),B,DUEDX
```

```
      IF(IFLAG.EQ.1) GO TO 100
      IF(ICOUNT.GT.ITER) GO TO 200
      GO TO 20
  100 IF(IDE.EQ.2) GO TO 8002
      REXI = UE*XI(3) / VO
      TETO = TE/TO
      TOTE = TO/TE
      UEC = UE*SQRT(TETO)
      REX = UEC*X(3) / VO
      D2 = TOTE**3*DEL2(3)
      D3 = TOTE**3*DEL3(3)
      D1 = H*D2
      CFC = TETO*CFF(3)
      REDELC = REDEL2*TOTE**2*SQRT(TOTE)
      PRINT 339
      PRINT 338,X(3),D1,D2,D3,H,H32(3),UEC,TE,ME,CFC,CD,REDELC
C-----NOW TEST FOR CHANGE IN REGIONS
 8002 CONTINUE
      GO TO (1000,2000,3000,4000),ISEP
  109 DO 110 I = 1,2
      DEL2(I) = DEL2(3)
      DEL1(I) = DEL1(3)
      DEL3(I) = DEL3(3)
      CAPY(I) = CAPY(3)
      CAPX(I) = CAPX(3)
      X(I) = X(3)
      XP(I) = XP(3)
      CFF(I) = CFF(3)
      H12(I) = H12(3)
  110 H32(I) = H32(3)
      CALL SAVEG(INDEX,X,DEL1,DEL2,DEL3,H12,H32,CF)
      IF(X(3).LT.XEND) GO TO 15
      IF(INDEX.GT.100) INDEX = 100
      PRINT 111,((QUANT(J,I),J=1,7),I=1,INDEX)
  111 FORMAT(7E18.6)
      GO TO 1
C-----TEST POWER LAW PROFILE RESULTS
 1000 IF(H32(3).GT.9.0) GO TO 109
      ISEP = 2  $ISKIN = 2
      PRINT 1001
 1001 FORMAT(* NOW SWITCH FROM POWER LAW TO YOSH-S FIT OF SEDDON*)
      GO TO 109
C-----TEST OF SANBORNS ONE PARAMETER PROFILE
 2000 IF(H12(3).LT.HSEP) GO TO 109
C-----NOW SWITCH TO SEPARATED FLOW EQUATIONS
      ISEP = 3
      ISKIN = 3
      XSEP = X(3)
      PRINT 337, XSEP
      HSEPT = H32(3)
      H = (H12(3) + 1.0)*(TO/TE) - 1.0
      DEL3(3) = H32(3)*DEL2(3)
      PRINT 338,XSEP,DEL1(3),DEL2(3),DEL3(3),H12(3),H32(3),UE,TE,ME,
     *CFF(3), CD, REDEL2
      GO TO 109
```

```
C-----TEST SEPARATED SOLUTION FOR REATTACHMENT
 3000 IF(H12(3).GT.HSEP) GO TO 109
      ISEP = 4  $ ISKIN = 4
      PRINT 3001, X(3)
 3001 FORMAT(*  REATTACHMENT X = * E10.3)
      GO TO 109
C-----TEST SOLUTION
 4000 IF(H32(3).LT.9.0) GO TO 109
      XREAT = X(3)
C-----NO. SETUP TO GO BACK TO HEADS  TURBULENT METHOD
      ISKIN = 1
      ISEP = 1
      H12(3) = HFACT(ISEP,H32(3),HSEP)
      DEL2(3) = DEL1(3)/H12(3)
      DEL2(3) = DEL1(3) / H
      DEL3(3) = H32(3) *DEL2(3)
      CALL VEL(XREAT)
      CFF(3) = SKINF(ISKIN,H12(3),H32(3),   DEL2(3),DEL1(3))
      PRINT 338,XSEP,DEL1(3),DEL2(3),DEL3(3),H12(3),H32(3),UE,TE,ME,
     *CFF(3), CD, REDEL2
      GO TO 109
C
C
  200 CALL EXIT
C
C
    2 FORMAT(2I5/8E10.3)
    3 FORMAT(I5 /(6E10.3))
    4 FORMAT(    * INCR =*I5* IBUG =*I5/* GAM =*E10.5* XSTART =*E10.5
     1 * XEND =*F10.5 , DELX =*F10.5* DELK     =*F10.5/
     2 * HSTART =*F10.5 * THETA0 =* F10.5 * DEL30 =*F10.5)
    5 FORMAT(16X*DIS*16X*UINF*16X*PINF*16X*MINF*16X*TINF*16X*BETA*/
     *    (6E20.6))
    7 FORMAT(3E10.3,4I5)
    8 FORMAT(*  PO (PSIA) =*F10.2* TO (R) =*F10.2* ERROR =*E10.3
     1 * ITER =*I5* IPROF =*I5)
   13 FORMAT(8A10)
   16 FORMAT(1H1,18X,8A10)
   2  FORMAT(5I5)
   23 FORMAT(4I5, 2E10.3)
   31 FORMAT(// *  NO CURVE- FIT OF THE ENTIRE INPUT VELOCITY DISTRIBUTI
     *ON*  /*  WE WILL USE A PARABOLIC FIT OF THREE POINTS*)
   36 FORMAT(/*. CURVE-FIT ENTIRE VELOCITY DISTRIBUTION*/
     1   * DEGREE OF POLYNOMIAL*5X,I5)
   37 FORMAT(* ISEP =*I5/ * ISKIN =*I5/* IFF =*I5)
   50 FORMAT(3E10.3)
   51 FORMAT(I5/(3E10.3))
  102 FORMAT(/16X*X-FT* 13X*DEL1-FT* 13X*DEL2-FT* 13X*DEL3-FT*
     1 17X*H12* 17X*H32*/6E20.6/ 14X*UE-FPS* 16X*TE-R* 18X*ME* 18X*CF*
     218X*CD* 14X*REDEL2*/6E20.6/ 19X*H*14X*CFF(3)*19X*A*
     315X*DUEDX* 17X *ETA*/5E20.6)
  103 FORMAT(17X*REX*18X*V0*14X*REXBAR*15X *REBAR*/
     *4E20.6/10X*ICOUNT =*I5)
  334 FORMAT(* CHECK SEPARATION EXTRAPOLATION*)
  337 FORMAT(* SEPARATION HAS OCCURRED AT X =*E10.3)
```

```
  338 FORMAT(/16X*X-FT* 13X*DEL1-FT* 13X*DEL2-FT* 13X*DEL3-FT*
     1 17X*H12* 17X*H32*/6E20.6/ 14X*UE-FPS* 16X*TE-R* 18X*ME* 18X*CF*
     218X*CD* 14X*REDEL2*/6E20.6)
  339 FORMAT(*  TRANSFORMED COMPRESSIBLE VARIABLES*)
  342 FORMAT(*REATTACHMENT OCCURRED AT X = *E10.3)
  730 FORMAT(/16X*X-FT* 13X*DEL1-FT* 13X*DEL2-FT* 13X*DEL3-FT*
     1 17X*H12* 17X*H32*/6E20.6/ 14X*UE-FPS* 16X*TE-R* 18X*ME* 17X*CD1*
     218X*CD2*14X*REDEL2*/6E20.6/
     3 17X*EEX* 17X*ETA* 15X *DUEDX* 18X *F1*
     4 18X *F2* 18X *F3*/6E20.6/
     5 18X *F4*/E20.6)
      END
```

```
      SUBROUTINE LAMINAR(THETO,XT,DEL30)
      COMMON/COEF/  Z(20),NO,ISIGN,IKIND,TO,PO,IPAR
      COMMON/XSKIN/ ISKIN,DELK,CE,H ,CD ,EODEL,IFE,ICE,HSEP,HREAT
      COMMON/XYZ/ UE, DUEDX, PE, TE, ME, RHOE, VQ,IM,VM,RHOM ,SINB
      COMMON/COM/GAM12,INCR,GGAM1,GAM,AI,ICOMP,IU  ,IDE
      COMMON/XOPI/ DISX(75),UINF(75),PINF(75),MINF(75),TINF(75),BETA(75)
      DIMENSION XC(50), DEL2(50), D2(50), UEX(61), DEL1(50), D1(50)
      COMMON / PDQ/ ILAM, H12, CVIS, C
      COMMON/FREE/ XTIF,XMIF,QIF,PIF,AIF,UIF,AIF2,UIF2,GXGAM1,CON37,
     1  XSTART, DELX, XEND
      COMMON S(75), UEI(75)
      DATA MMAX /60/
      REAL MINF , ME
      THETAO = THETO/C
      XFINAL = XEND
      G1 = 1.0 /GGAM1
      RC = UIF*C*144.0*PIF/(53.3*XTIF*VISCOS(XTIF))
      PRINT 8, RC
    8 FORMAT (/* REYNOLDS NUMBER BASED UPON CHORD LENGTH = *E20.5)
C-----DELX IS INPUT VALUE OF STEP SIZE ** COMPRESSIBLE
      DXC = DELX
      XC(1) = XSTART
      DEL2(1) = THETAO
      MAX = 0
      I = 1
      LAMSEP = 0
      LAMTRAN = 0
      D2(1) = THETAO*(XTIF/TINF(1))**3
   10 I = I + 1
      XC(I) = XC(I-1) + DXC
      MAX = MAX + 2
      IF(MAX.GT.MMAX) MAX = MMAX
      MM  = MAX + 1
      DELDXC = XC(I)/FLOAT(MAX)
      XL = - DELDXC
      ITER = 1
      DO 20 J = 1,MM
      XL = XL + DELDXC
      CALL INTERP(XL,S,UEI,IU,UE,DUEDX,ITER)
      UE = UE/UIF
   20 UEX(J) = UE**5
      DUEDX = DUEDX/UIF
      SUB = 0.0
      CALL SIMP(MAX,DELDXC,SUB,UEX,UEX5)
C-----UE = LAST POINT OF INTEGRATION
C-----DEL2(I) = THETA / C
      DEL2(I) = SQRT(0.45/( RC*UE**6)*UEX5)
      XLAM = DEL2(I)**2*RC*DUEDX
      IF(XLAM.GT.-0.090) GO TO 224
      LAMSEP = 1
      XLAM = - 0.090
  224 IF(XLAM.LT.-0.04) GO TO 225
      H12 = 2.61 - 3.87333*XLAM + 5.7333*XLAM**2
      CFX = 0.220 + 1.57*XLAM - 1.80*XLAM**2
      GO TO 226
```

```
C-----PARABOLIC FITS OF H AND L
  225 H12 = 2.61 + 0.68889*XLAM + 142.22*XLAM**2
      CFX = 0.220 +  0.46111*XLAM - 22.037*XLAM**2
  226 DEL1(I) = H12*DEL2(I)
      CALL INTERP(XC(I),S,DISX,JU,XCOMP,DUM,1)
      CALL INTERP(XC(I),S,TINF,IU,TE,DUM,1)
      CF    = 2.0*CFX/(RC*DEL2(I))
      H = TO/TE*(H12 + 1.0) - 1.0
      D2(I) =    DEL2(I)*(XTIE/TE)**3 * CVIS
      D1(I) = D2(I) * H
      RX = RC*XC(I)
      RTHETA = RC*DEL2(I)
      RTRANS = 1.718*RX**0.435
      IF(RTHETA.GE.RTRANS) LAMTRAN = 1
      PRINT 150,I,UE,XC(I),XCOMP,DEL2(I),D2(I),DEL1(I),D1(I),CF    ,
     1 RX, RTHETA, RTRANS, XLAM, LAMSEP, LAMTRAN
      IF(LAMSEP.EQ.1) PRINT 151
  151 FORMAT(* LAMINAR SEPARATION HAS OCCURRED *)
      IF(LAMTRAN.EQ.0) GO TO 55
      PRINT 152
  152 FORMAT(* LAMINAR TRANSITION TO TURBULENT FLOW HAS OCCURRED  *)
      DELH = 0.657 + 0.101*ALOG10(RTRANS)
      HTURB = H12 - DELH
      H = TO/TE*(H12 + 1.0) - 1.0
      H1 = 2.0*H12/(H12 - 1.0)
      H12 = HTURB
      HSTART = H12
      XT = XC(I)
      DELK = D2(I)*H  * C
      DEL30 = D2(I)*H1 * C
      THETO = D2(I) * C
      PRINT 153, HTURB, H, H1
  153 FORMAT(/* STARTING INCOMPRESSIBLE VALUES OF H * F10.3/
     1 * STARTING COMPRESSIBLE H * F10.3/* STARTING VALUE OF H1*F10.3)
      RETURN
   55 IF(XC(I).LT.XFINAL) GO TO 10
      RETURN
  150 FORMAT(/19X*I*13X*UE/UINF*18X*XC*6X*X-COMPRESSIBLE*14X*DEL2/C*
     1  11X*DEL2-COMP*/ I20,5E20.5//
     2 30X *DEL-STAR/C* 6X*DEL-STAR-COMP* 11X*CF-INCOMP*16X*RE-X*
     3 12X*RE-THETA*/20X,5E20.5/
     4 32X*RE-TRANS* 16X*XLAM* 10X*SEPARATION* 10X*TRANSITION*/
     5 20X, 2E20.5, 2I20)
      END
```

```
      SUBROUTINE TRANS(DELX,     THETAO,DEL30,HSTART,XSTART)
      COMMON/FREE/ XTIF,XMIF,UIF,PIF,AIF,UIE,AIF2,UIF2,GXGAM1,CON37,
     1  XSTART, DELX, XEND
      COMMON/POU/ ILAM, H12, CVIS, C
      COMMON/XSKIN/ ISKIN,DELK,CF,H,CQ,FODEL,IFE,ICF,HSEP,HREAT
      COMMON/COEF/  Z(20),HO,ISIGN,IKIND,TO,PO,IPAR
      COMMON/XYZ/ UE, DUEDX, PE, TE, ME, RHOE, VO,IV,VM,RHOM ,SINB
      COMMON/CON/GAM12,INCR,GGAM1,GAM,AT,ICOMP,IU
     * ,IUE
      COMMON/XOPI/ DISX(75),UINF(75),PINF(75),MINF(75),TINF(75),BETA(75)
      REAL MINE,ME
      COMMON XI(75)  ,UEI(75)
      DIMENSION TETO4(100), COFF(3)
      XI(1) = DISX(1)
      UEI(1) = UINF(1)/SQRT(TINF(1)/XTIF)
      DO 50 I=2,IU
      I1 = I - 1
      IF(I.LT.IU) GO TO 21
      I1 = I - 2
   21 X = DISX(I) - DISX(I-1)
      N = 10
      CALL PARFIT(DISX(I1),TINF(I1),COFF,X,DUM,DUM)
      DX = X/FLOAT(N)
      XC = DISX(I-1) - DX
      NMAX = N + 1
      DO 33 IN = 1,NMAX
      XC = XC + DX
      IF(I.EQ.IU.AND.IN.EQ.NMAX) XC = DISX(IU)
      TE = COFF(1) + XC*COFF(2) +XC**2*COFF(3)
      TETI = TE/XTIF
   33 TETO4(IN) = TETI**4
      TEMP = 0.0
      DO 34 IN =2,NMAX
   34 TEMP = TEMP + 0.5*(TETO4(IN) + TETO4(IN-1))
C-----CVIS = VISCOSITY RATIO CONSTANT
      XI(I) = XI(I-1) + DX*TEMP*CVIS
      UEI(I) = UINF(I) / SQRT(TINF(I)/XTIF)
   50 CONTINUE
      PRINT 75,(XI(I), UEI(I), I = 1,IU)
   75 FORMAT(*  INCOMPRESSIBLE PROFILE*/ 19X*X* 16X*UINF*/(2E20.6))
C-----TRANSFORM INITIAL VALUES TO INCOMPRESSILE FORM
      CALL VEL(XSTART)
      TETI = TE/XTIE
      TETO = TE/TO
      T3 = TETI**3
      THETAO = THETAO*T3
      DEL30 = DEL30*T3
      DELK = DELK*T3
      H = HSTART
      HBAR = TETO*(H + 1.0) - 1.0
      H12 = HBAR
      PRINT 79
   79 FORMAT(//* INCOMPRESSIBLE STARTING CONDITIONS  *)
      PRINT 80,DELK,THETAO,DEL30,HBAR
   80 FORMAT(10X*DELK  =*F15.7/8X*THETAO =*E15.7/9X*DEL30 =*F15.7/
```

```
      *BX*BSTART =*E15.7)
       RETURN
       END
```

```
      SUBROUTINE SLOPED(X,H12,H32,DEL1,DEL2,DEL3,DD2DX,DD3DX)
      COMMON/XSFIN/ ISKIN,DELK,CF,H ,CD ,FODEL,IFE,ICF,HSEP,HREAT
      COMMON/XYZ/ UE, DUEDX, PE, TE, ME, RHOE, VO,IM,VM,RHOM ,SINB
      COMMON/CON/GAM12,INCR,GGAM1,GAM,AT,ICOMP,IU  ,IDE
      REAL ME, IP, ISTARP, MINF
      CALL VEL(X)
      CFF= SKINF(ISKIN,H12,H32,   DEL2,DEL1)
      GO TO(10,20),IDE
   10 CONTINUE
      DD2DX = 0.5*CFF - DUEDX/UE *DEL2 *(2.0 + H12)
      DD3DX = CD -      DUEDX/UE*DEL3
      RETURN
   20 REUE = RHOE*UE
      REUEM = REUE* SINB
      DD3DX = REUE * CD
      REUE = REUE*UE
      DD2DX = 0.5*CF*REUE -   DEL1*REUEM * DUEDX
      RETURN
      END
```

```
      SUBROUTINE VEL(X)
      COMMON/CON/GAM12,INCR,GGAM1,GAM,AT,ICOMP,IU  ,IDE
      COMMON/COEF/  Z(20),NO,ISIGN,IKINC,TO,PO,IPAR
      COMMON/XUPI/ DISX(75),UINF(75),PINF(75),MINF(75),TINF(75),BETA(75)
      COMMON/XYZ/ UE, DUEDX, PE, TE, ME, RHOE, VO,IM,VM,RHOW ,SINB
      REAL ME,MINF
      DIMENSION COEF(3)
C
      IF(NO.GT.1) GO TO 90
      CALL INTER(X,DISX,LIND,XOW,IU,LI)
      I2 = LI + 1
      I1 = LI - 1
      IF(IPAR.EQ.1) GO TO 100
      IF(LIND.GT.IU)GO TO 60
      UE = UINF(LI)
      PE = PINF(LI)
      TE = TINF(LI)
      ME = MINF(LI)
      SINB = SIN(BETA(LI))
      IF(LI.GT.1)GO TO 30
      I1 = LI
   30 DUEDX = (UINF(I2) - UINF(I1))/(DISX(I2) - DISX(I1))
   31 RHOE = PE*144.0/(53.3*TE)
      RETURN
   60 CON1 = 1.0 - XOW
      UE = UINF(LI)*CON1 + XOW*UINF(I2)
      PE = PINF(LI)*CON1 + XOW*PINF(I2)
      TE = TINF(LI)*CON1 + XOW*TINF(I2)
      ME = MINF(LI)*CON1 + XOW*MINF(I2)
      TB = BETA(LI)*CON1 + XOW*BETA(I2)
      SINB = SIN(TB)
      I1 = LI
      GO TO 30
C------USE FIT OF ENTIRE VELOCITY PROFILE
   90 XX = X
      NO1 = NO + 1
      P = 1.0
      UE = 0.0
      DUEDX = 0.0
      DO 92 J = 1,NO1
      UE = UE + P*Z(J)
   92 P = P*XX
      P = 1.0
      DO 94 J=2,NO1
      DUEDX = DUEDX + P*Z(J) *FLOAT(J-1)
   94 P = P*XX
   95 UAT = UE/AT
      UAT2 = UAT**2
      IF(IDE.EQ.2) GO TO 995
      TEMP = 1.0 + GAM12*UAT2
      TE = TO/TEMP
      PE = PO/TEMP**GGAM1
      ME = UAT
      GO TO 31
  995 TEMP = 1.0 - GAM12*UAT2
```

```
      TE = TO*TEMP
      PE = PO*TEMP**GGAM1
      WE = SQRT(UAT2/TEMP)
      GO TO 31
  100 IF(LI.EQ.1) GO TO 101
      IF(XOW.GT.0.5) I1 = LI
      IF(I1.GT.IU-2) I1 = IU - 2
      GO TO 102
  101 I1 = 1
  102 CONTINUE
      XX = X
      CALL PARFIT(DISX(I1),UINF(I1),COFF,XX,UE,DUEDX)
      CALL PARFIT(DISX(I1),BETA(I1),COFF,XX,TB,DUM)
      SINB = SIN(TB)
      GO TO 95
      END
```

```
      SUBROUTINE GAUSSEN(NPTS,NO,IKIND,ISIGN,X,Y,WORD,Z)
C-----NPTS    =  NUMBER OF DATA POINTS.
C-----NO      =  DEGREE OF POLYNOMIAL TO BE FITTED.
C-----IKIND   =  1  EQUATION TO BE FITTED IS,
C-----              LN(Y) = A0 + A1*X + A2*X**2 + . . . . . + AN*X**N
C-----IKIND   =  2  EQUATION TO BE FITTED IS,
C-----              Y = A0 + A1*X + A2*X**2 + . . . . . . + AN*X**N
C-----ISIGN   =  1  NEGATIVE EXPONENTS.
C-----ISIGN   =  2  POSITIVE EXPONENTS.
      DIMENSION  A(20,21), X(100), Y(100), Z(20), C(20), WORD(10)
      PRINT 15, (WORD(I), I=1,8)
      PRINT 17, (X(I), Y(I), I=1,NPTS)
      IFLAG = 0
      DO 5 J=1,NPTS
      IF(Y(J)) 3,3,5
    3 IFLAG = 1
      GO TO 19
    5 CONTINUE
   19 IF(IFLAG-IKIND) 8,6,8
    6 PRINT 7
      RETURN
    8 NO1 = NO+1
      NO2 = NO+2
      DO 9 I=1,NO1
      DO 9 J=1,NO2
    9 A(I,J) = 0.0
      DO 20 I=1,NPTS
      R = Y(I)
      GO TO (10,12), IKIND
   10 R = LOGF(R)
   12 S = X(I)
      GO TO (14,16), ISIGN
   14 S = 1.0/S
   16 Q = 1.0
      A(1,NO2) = A(1,NO2) + R
      DO 18 J=2,NO1
      Q = Q*S
      A(1,J) = A(1,J) + Q
   18 A(J,NO2) = A(J,NO2) + Q*R
      DO 20 K=2,NO1
      Q = Q*S
   20 A(K,NO1) = A(K,NO1) + Q
      DO 30 I=2,NO1
      DO 30 J=1,NO
   30 A(I,J) = A(I-1,J+1)
      A(1,1) = NPTS
      CALL SOLVE (A, Z, C, NO1)
      SUM = 0.0
      DO 50 I=1,NPTS
      Q = X(I)
   99 ZY = 0.0
      P = 1.0
      DO 52 J=1,NO1
      ZY = ZY + P*Z(J)
   52 P = P*Q
```

```
      ERR = 100.*(Y(I)-ZY)
      YERR = ERR*ERR
      ERR = ERR/Y(I)
      SUM = SUM + YERR
      PRINT 54,  X(I), Y(I), ZY, ERR, YERR
   54 FORMAT (5X,3E15.7,F12.5,E15.7)
   50 CONTINUE
      PRINT 56,  SUM
   56 FORMAT (1H0,62X,E15.7)
      RETURN
    7 FORMAT (/72H SOME OF THE Y VALUES ARE ZERO OR NEGATIVE, LN(Y) = F(
     1X) CANNOT BE USED.)
   13 FORMAT (8A10)
   15 FORMAT (1H1///12H INPUT DATA.,18X,8A10///)
   17 FORMAT (5X,2E15.7)
      END
```

```
      SUBROUTINE SOLVE (A, X, C, NO)
C-----S1002  SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS BY
C-----GAUSS,S ELIMINATION SCHEME (DOOLITTLE,S METHOD).
      DIMENSION  A(20,21), B(20,21), X(20), C(20), LOC(20), ROW(20)
      KNO = NO+1
      DO 150 I=1,NO
      DO 150 J=1,KNO
  150 B(I,J) = A(I,J)
  254 PRINT 15
   15 FORMAT (1H1,///17H MATRIX TO SOLVE.)
      DO 5 I=1,NO
    5 PRINT 107,  (A(I,J), J=1,NO)
  107 FORMAT (/(5X,8E14.6))
      PRINT 109
  109 FORMAT (///)
      PRINT 107,  (A(I,NO+1), I=1,NO)
      PRINT 13
   13 FORMAT (1H0///100H          I           X(I)              I          X(I)
     1          I         X(I)              I          X(I)       /)
  252 DO 10 M=1,NO
      LOC(M) = 0
   10 ROW(M) = 0.0
      NP = NO+1
      DO 100 I=1,NO
      IP = I+1
C-----FIND MAX ELEMENT IN I-TH COL.
      AMAX = 0.0
      DO 2 K=1,NO
      IF(AMAX - ABSF(A(K,I))) 3,2,2
C-----IS NEW MAX IN ROW PREVIOUSLY USED AS PIVOT.
    3 IF(ROW(K)) 4,4,2
    4 LOC(I) = K
      AMAX = ABSF(A(K,I))
    2 CONTINUE
      IF(AMAX) 99,99,98
C-----MAX ELEMENT IN I-TH COL IS A(L,I)
   98 L = LOC(I)
      ROW(L) = 1.0
C-----PERFORM ELIMINATION, L IS PIVOT ROW, A(L,I) IS PIVOT
C----- ELEMENT.
      DO 50 J=1,NO
      IF(L-J) 6,50,6
    6 QF = -A(J,I)/A(L,I)
      DO 40 K=IP,NP
   40 A(J,K) = A(J,K) + QF*A(L,K)
   50 CONTINUE
  100 CONTINUE
      DO 200 I=1,NO
      L = LOC(I)
  200 X(I) = A(L,NO+1)/A(L,I)
  250 PRINT 103,  (J, X(J), J=1,NO)
  103 FORMAT (4(I8,2X,E15.8))
      PRINT 109
      DO 120 I=1,NO
      SUM = 0.0
```

```
      DO 110 M=1,NO
  110 SUM = SUM + B(L,M)*X(M)
  120 C(L) = SUM
      PRINT 122,  (C(I), B(I,NO+1), I=1,NO)
  122 FORMAT (///(5X,2E15.8))
  250 RETURN
   91 PRINT 104
  104 FORMAT (5X,27H NO UNIQUE SOLUTION EXISTS.)
      RETURN
      END
```

```
      SUBROUTINE INTER (WNO,WOMEG,JIND,WOW,NW,JI)
      DIMENSION WOMEG(1)
      JI = 1
      JUU = NW
      JO = NW/2
    9 IF(WNO-WOMEG(JO))8,523,25
    8 JUU = JO
      JO = (JO + JI)/2
      GO TO 26
   25 JI = JO
      JO = (JO + JUU)/2
   26 IF(JUU-JI-1) 23,27,9
   27 JO = JI
      IF(WNO - WOMEG(JO))90,23,29
   90 WRITE(6,92) WNO
   92 FORMAT(22H ERROR IN INTER WNO =  ,E20.5)
      CALL EXIT
  523 JI = JO
      JUU = JI + 1
   23 JIND = JO
      GO TO 60
   29 IF(WNO - WOMEG(JUU))61,62,90
   62 JIND = JUU
      GO TO 60
C-----SET FLAG TO INDICATE INTERPOLATION IS NECESSARY
   61 JIND = 1000
      WOW = (WNO - WOMEG(JI))/(WOMEG(JUU) - WOMEG(JI))
   60 RETURN
      END
```

```
      SUBROUTINE INTERP(WNO,WOMEG,HLAM,NW,ANS,DERV,ITER)
      DIMENSION WOMEG(1),HLAM(1), COFE(3)
C
      DERV = 0.0
      JI = 1
      JUU = NW
      JO = NW/2
    4 IF(WNO-WOMEG(JO))8,523,25
    8 JUU = JO
      JO = (JO + JI)/2
      GO TO 26
   25 JI = JO
      JO = (JO + JUU)/2
   26 IF(JUU-JI-1) 23,27,4
   27 JO = JI
      IF(WNO - WOMEG(JO))90,23,29
   90 PRINT 92,WNO
   92 FORMAT(22H ERROR IN INTER WNO =  ,E20.5)
      CALL EXIT
  523 JI = JO
      JUU = JI + 1
   23 JIND = JO
      GO TO 60
   29 IF(WNO - WOMEG(JUU))61,62,90
   62 JIND = JUU
      GO TO 60
C-----SET FLAG TO INDICATE INTERPOLATION IS NECESSARY
   61 JIND = 1000
   60 XOW = (WNO - WOMEG(JI))/(WOMEG(JUU) - WOMEG(JI))
      GO TO (70,100),ITER
  100 I1 = J1 - 1
      IF(J1.EQ.1) GO TO 101
      IF(XOW.GT.0.5) I1 = J1
      IF(I1.GT.(NW-2)) I1 = NW - 2
  101 I1 = 1
  102 CONTINUE
      CALL PARFIT(WOMEG(I1),HLAM(I1),COFE,WNO,ANS,DERV)
      RETURN
   70 ANS = XOW*(HLAM(JUU) - HLAM(JI)) + HLAM(JI)
      DERV = (HLAM(JUU) - HLAM(JI))/( WOMEG(JUU) - WOMEG(JI))
      RETURN
      END
```

```
      SUBROUTINE PARFIT(Y,D,COEF,YY,ANS,DANSDY)
      DIMENSION Y(3), D(3), COEF(3)
C-----GIVES COEFFICIENTS FOR A PARABOLIC FIT
      X1 = Y(1)
      X2 = Y(2)
      X3 = Y(3)
      D1 = D(1)
      D2 = D(2)
      D3 = D(3)
      FACTOR  = X2*X3*X3 - X2*X2*X3 - X1*X3*X3 + X1*Y2*X2 + X1*X1*X3
     *        - X1*X1*X2
      A1      = X2*X3*X3 - X2*X2*X3
      A2      = X3*X1*X1 - X1*X3*X3
      A3      = X1*X2*X2 - X2*X1*X1
      B1      = X2*X2 - X3*X3
      B2      = X3*X3 - X1*X1
      B3      = X1*X1 - X2*Y2
      C1      = X3 - X2
      C2      = X1 - X3
      C3      = X2 - X1
      COEF(1) = (D1*A1 + D2*A2 + D3*A3)/ FACTOR
      COEF(2) = (D1*B1 + D2*B2 + D3*B3) / FACTOR
      COEF(3) = (D1*C1 + D2*C2 + D3*C3) / FACTOR
      ANS = COEF(1) + COEF(2)*YY + COEF(3)*YY**2
      DANSDY = COEF(2) + 2.0*COEF(3)*YY
      RETURN
      END
```

```
      SUBROUTINE SIMP(MAX,DELU,SUB,C,ANS)
      DIMENSION C(1)
C
      MXX = MAX + 1
      MO = MXX - 2
      ME = MXX - 1
      SUM1 = 0.0
      SUM2 = 0.0
      DO 5 J = 2,ME,2
    5 SUM1 = SUM1 + C(J)
      DO 7 J=3,MO,2
    7 SUM2 = SUM2 + C(J)
      SUM3 = C(1) + C(MXX)
      ANS = SUB + DELU*(SUM3 + 4.0*SUM1 + 2.0*SUM2)/3.0
      RETURN
      END
```

```
      FUNCTION SKINF(IS,H12,H32,   DEL2,DEL1)
      COMMON/XYZ/ UE, DUEDX, PE, TE, ME, RHOE, VO,TM,VM,RHOM ,SINB
      COMMON/XSKIN/ ISKIN,DELK,CF,H ,CD ,FODEL,IFE,ICF,HSEP,HREAT
      COMMON/COM/GAM12,INCR,GGAM1,GAM,AI,ICOMP,IU  ,IDE
      COMMON/COEF/  Z(20),NQ,ISIGN,IKIND,TO,PO,IPAR
      REAL ME, TP, ISTARP, MINF
      TM = 0.5*(TO + TE) + 0.22*(TU - TE)
      GO TO (10,10,30,10),IS
   10 CONTINUE
      ALPHA = 0.246*EXP(-1.561*H12)
      IF(ICF.EQ.1) GO TO 11
      TEMP = HSEP/H12
      IF(IS.EQ.4) TEMP = HREAT/H12
      IF(TEMP.LE.1.0) GO TO 30
      ALPHA = 0.058*(1.95*ALOG10(TEMP   ))**1.705
   11 REDEL2 = UE*DEL2   /VO
      IF(IDE.EQ.2) REDEL2 = UE*RHOE*DEL2/VISCOS(TM)
      CON1 = REDEL2**0.268
      CF =     ALPHA/CON1
   20 IF(IFE.EQ.1) GO TO 21
      CON2 = (H12 + 1.0)/((3.0*H12 - 1.0)*(2.0*H12 - 1.0))
      HK = H12 *(1.0 + (0.2*ME**2*CON2*( H12 - 1.0)**2/H12)/
     1      ( 1.0 + 0.2*ME**2*(1.0 - H12*CON2)))
      H1K = 3.4 + 1.47/(HK - 0.5)**3.8
      CD = 0.0306/(H1K - 3.0)**0.653
      GO TO 22
   21 CD = 0.0306/(H32 - 3.0)**0.653
   22 SKINF = CF
      FODEL = CD/(DEL2*H32)
      CF = CF*TE/TO
      IF(IDE.EQ.1) RETURN
      CF = SKINF*TE/TM
      RETURN
   30 CF = 0.0
      GO TO 20
      END
```

```
      FUNCTION HFACT(ISEP,HS,HSEP)
      COMMON/XYZ/ UE, DUEDX, PE, TE, ME, RHOE, VO,IM,VM,RHOM ,SINB
      COMMON/XSKIN/ ISKIP,DELK,CF,M ,CD ,FODEL,IFF,ICF,HSEP,HREAT
      COMMON/COEF/  Z(20),HO,ISIGN,IKIND,TO,PO,IPAR
      REAL ME
      DATA XMO,HO/1.94,1.285/
      GO TO ( 1,5,5,50), ISEP
C-----NORMAL TURBULENT FLOW POWER LAW PROFILE
    1 HFACT = HS/(HS - 2.0)
   10 TEMP = (TO/TE)
      H = (HFACT + 1.0)*TEMP - 1.0
      HSEP = 3.0
      RETURN
C-----YOSH-S FIT TO SEDDON-S DATA
    5 CONTINUE
      HFACT = 1.0 + 2.0*(13.797/(HS + 9.297))**7
      GO TO 10
C-----REATTACHMENT
   50 HFACT = 1.0 + (0.9/(HS - 4.3))**0.75
      GO TO 10
      END
```

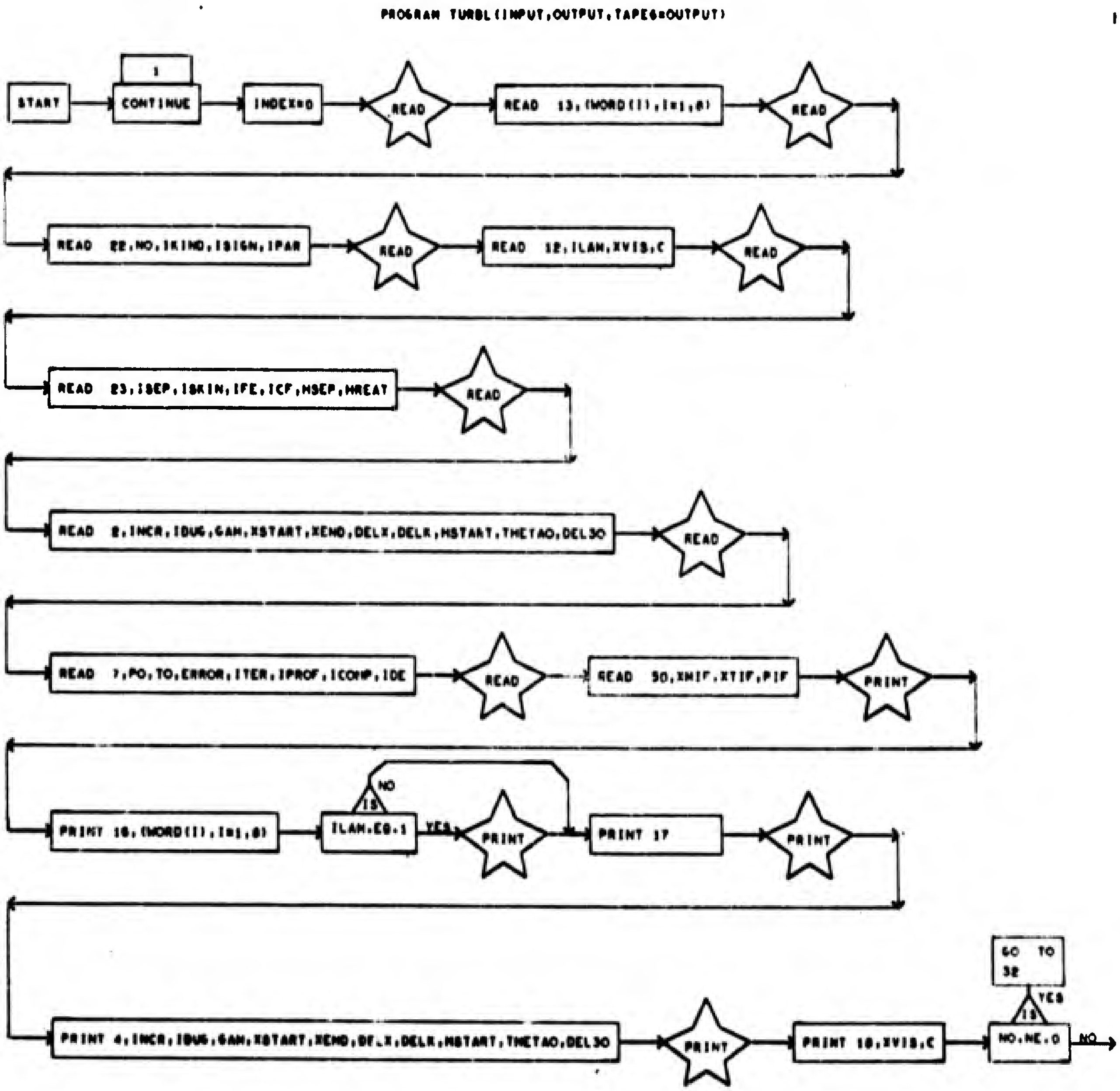
```
      FUNCTION XINCOM(X)
C-----X REPRESENTS COMPRESSIBLE VALUES OF X
C-----XINCOM IS EQUIVALENT INCOMPRESSIBLE VALUES
      COMMON/COE/GAM12,INCR,GGAM1,GAM,AT,ICOMP,IU
     * ,IUE
      COMMON/XOPT/ DISX(75),UINF(75),PINF(75),MINF(75),TINF(75),BETA(75)
      COMMON XI(75) , UEI(75)
      DIMENSION COFF(3)
      CALL INTER(X,XI,LIND,XOW,IU,LI)
      IF(LI.EQ.1) GO TO 101
      IF(XOW.GT.0.5) I1 = LI
      IF(LI.GT.IU - 2) I1 = IU - 2
      GO TO 102
  101 I1 = 1
  102 CONTINUE
      XX = X
      CALL PARFIT(DISX(I1),XI(I1),COEF,XX,TEMP,DUM)
      XINCOM = TEMP
      RETURN
      END
```

```
      FUNCTION VISCOS(T)
C-----VISCOSITY OF AIR IN UNITS OF LB/FT-SEC
      VISCOS = 71.484E-8*T**1.5/(T + 180.0)
      RETURN
      END
```

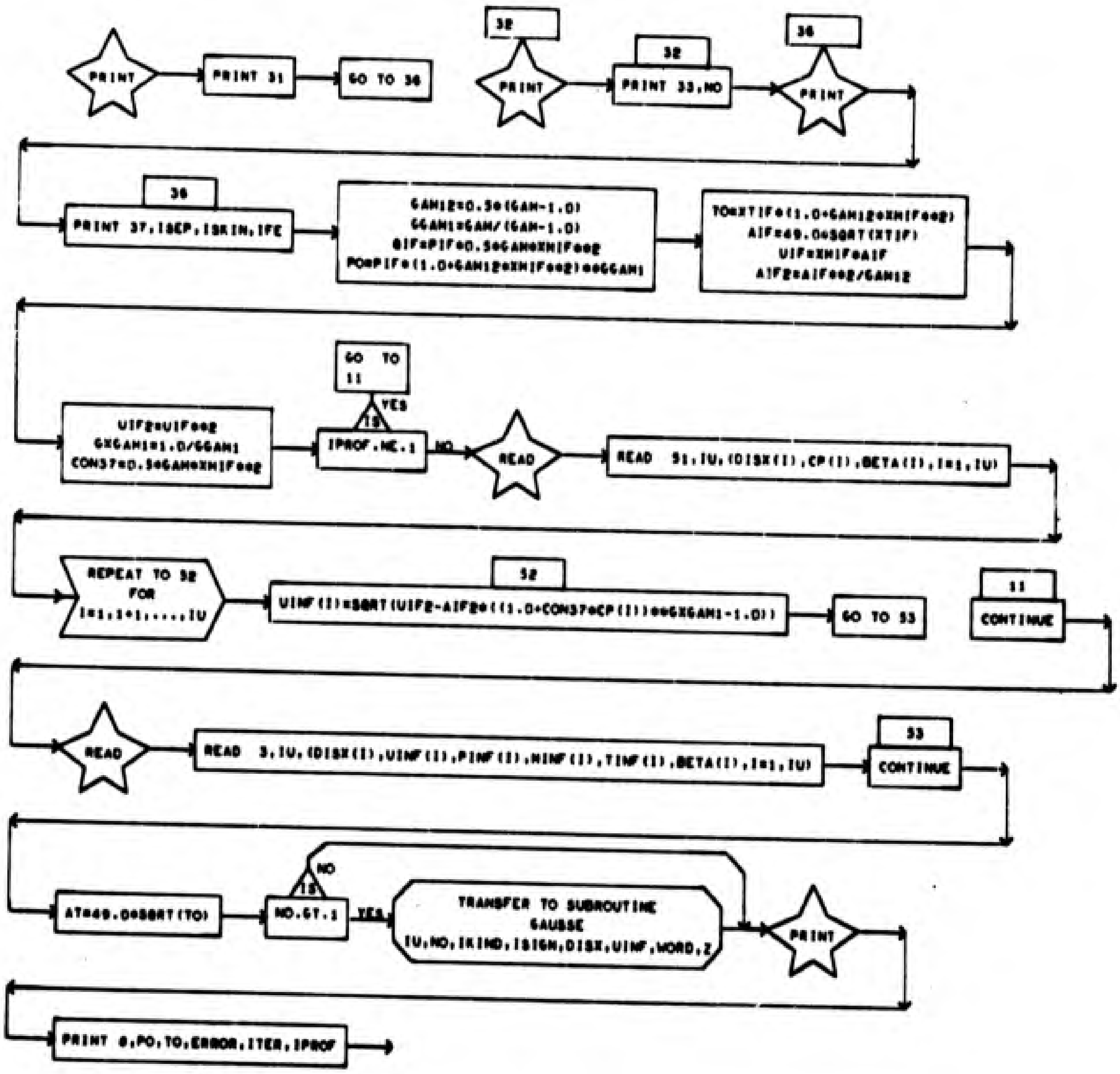
```
      SUBROUTINE SAVEQ(INDEX,X,D1,D2,D3,H12,H32,CF)
      COMMON/SAVEIT/ QUANT(7,100)
      INDEX = INDEX + 1
      IF(INDEX.GT.100) RETURN
      I = INDEX
      QUANT(1,I) = X
      QUANT(2,I) = D1
      QUANT(3,I) = D2
      QUANT(4,I) = D3
      QUANT(5,I) = H12
      QUANT(6,I) = H32
      QUANT(7,I) = CF
      RETURN
      END
```

APPENDIX VII

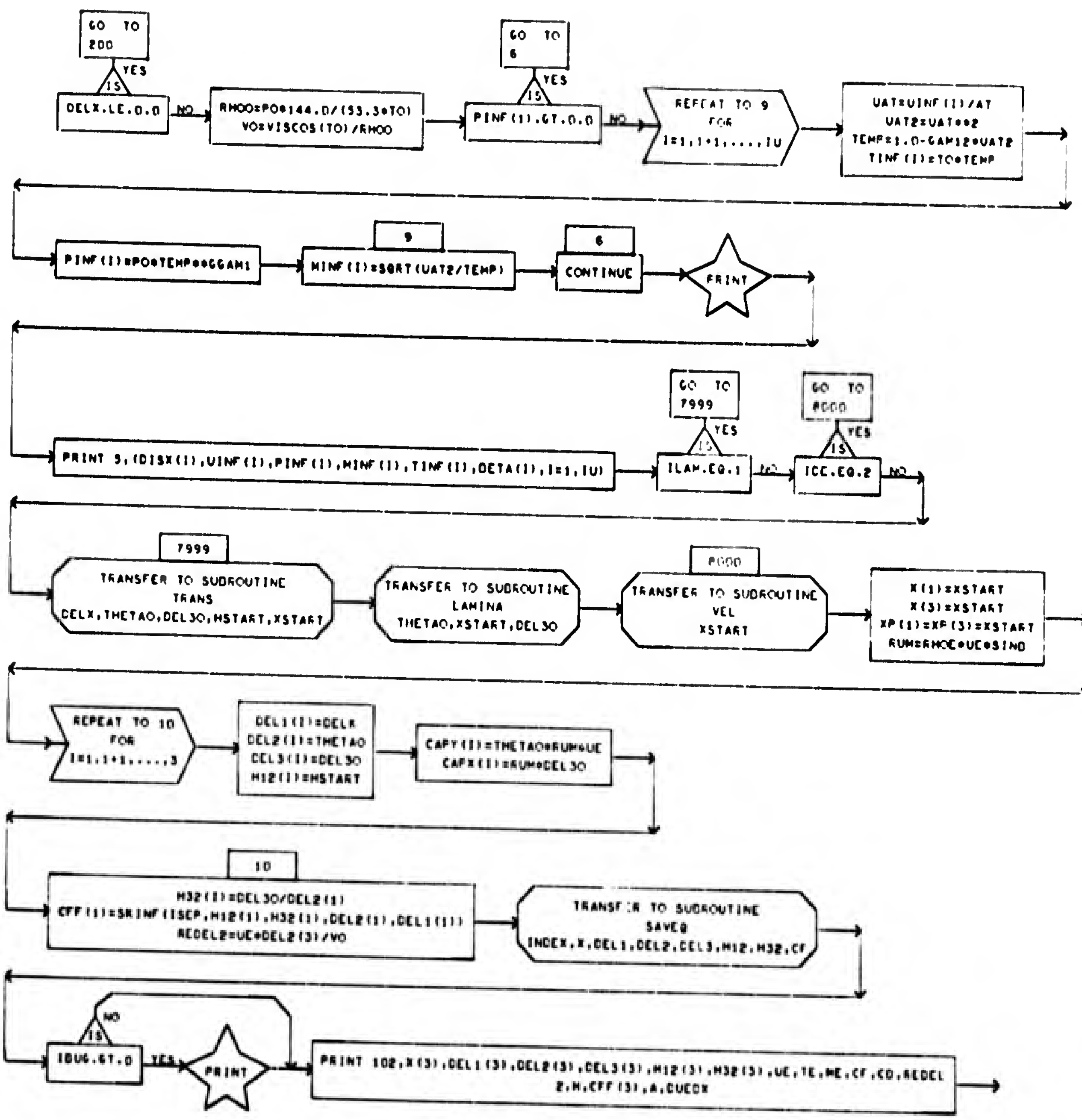
BOUNDARY LAYER PROGRAM FLOW CHARTS



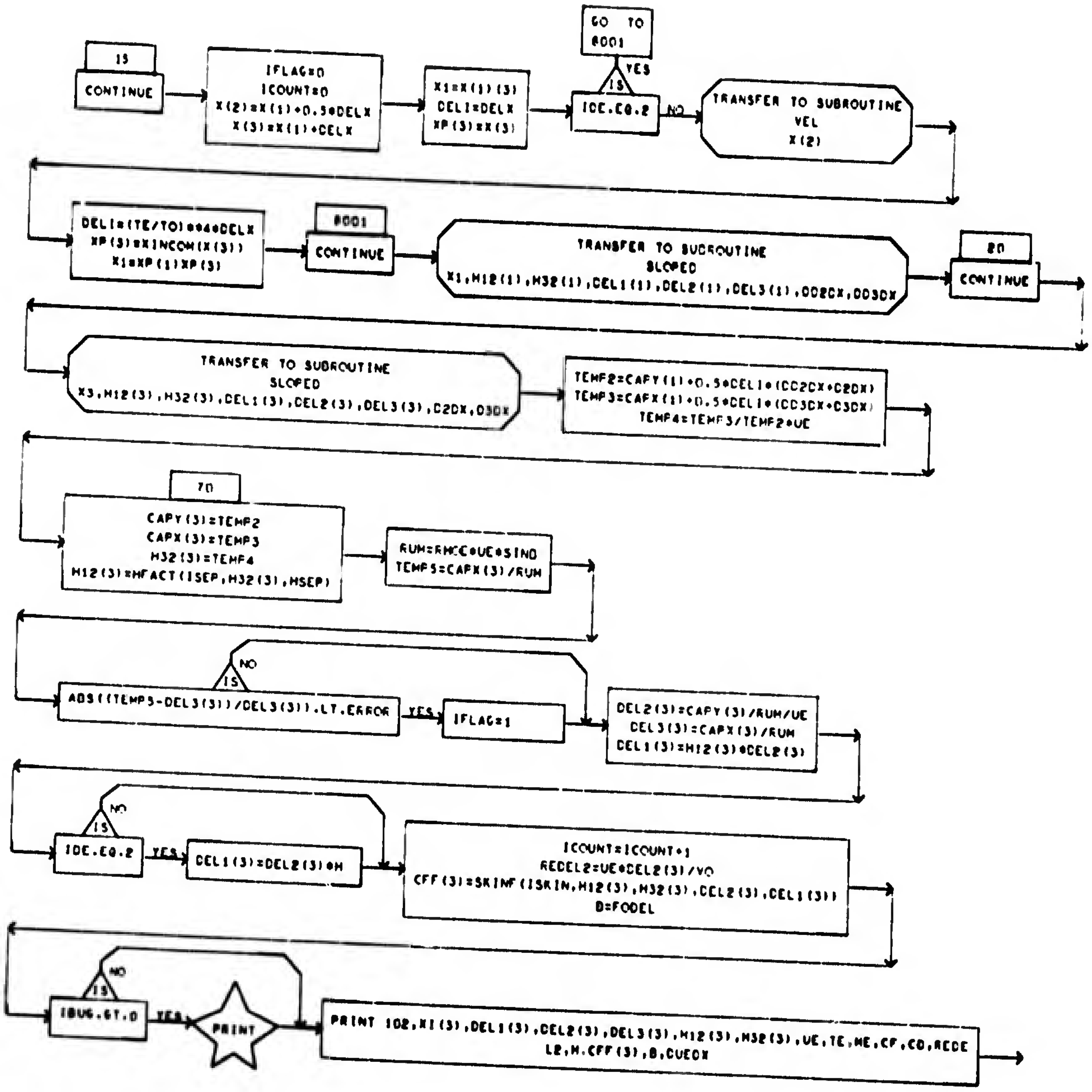
Detailed Flow Charts of Subroutine TURBL (1 of 7)



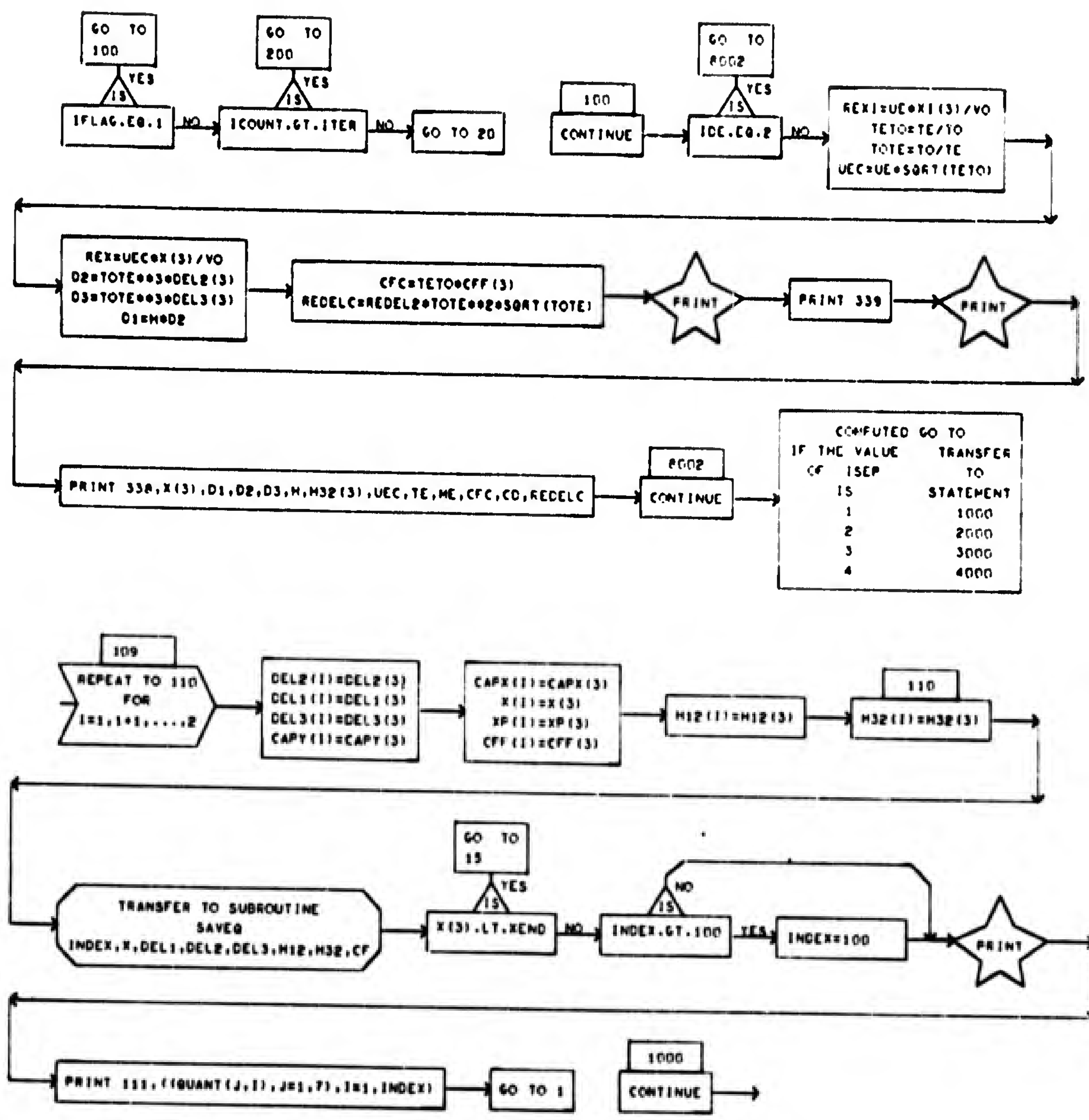
Detailed Flow Charts of Subroutine TURBL (2 of 7)



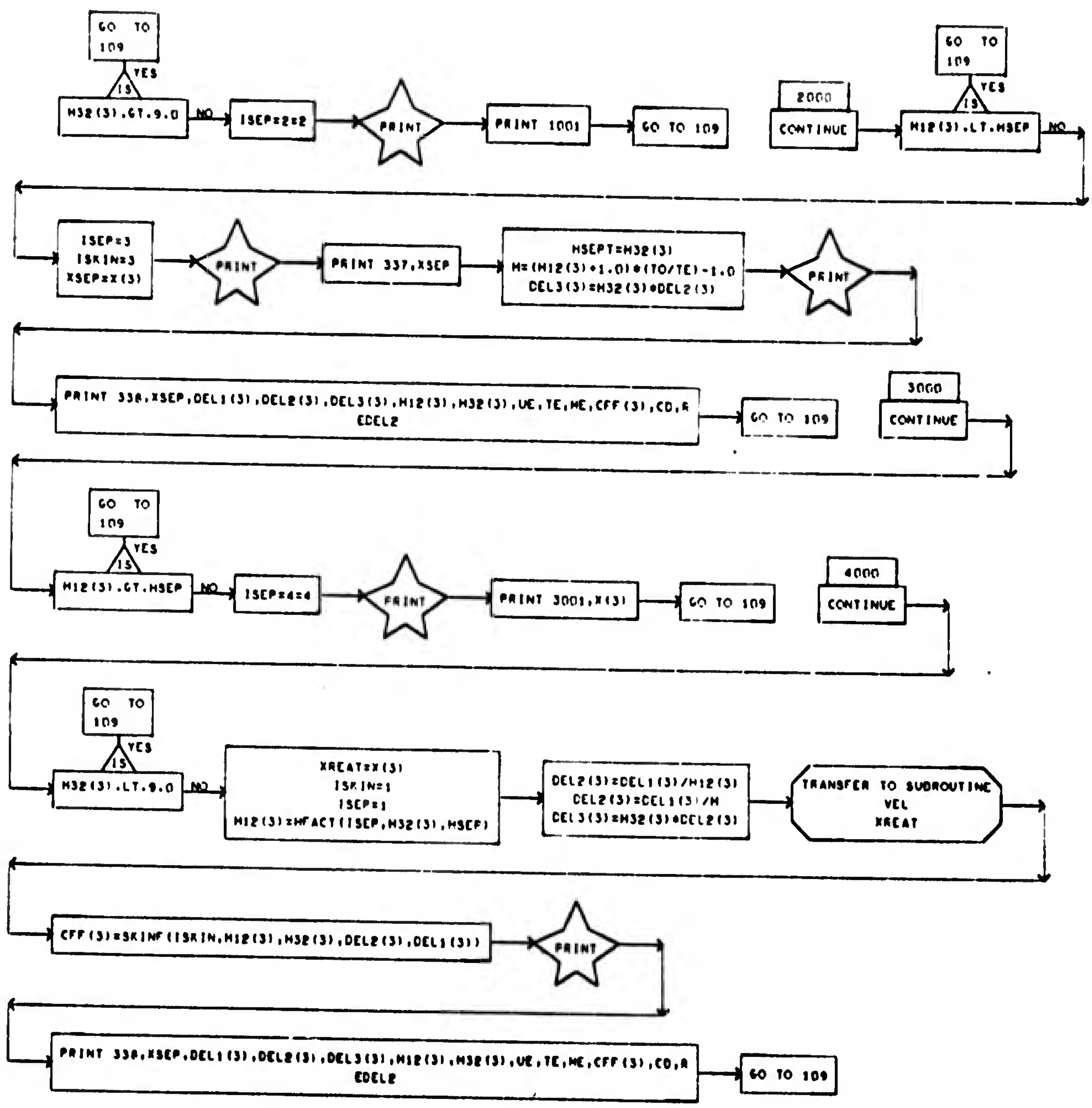
Detailed Flow Charts of Subroutine TURBL (3 of 7)



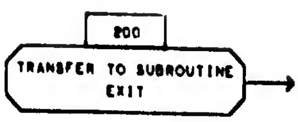
Detailed Flow Charts of Subroutine TURBL (4 of 7)



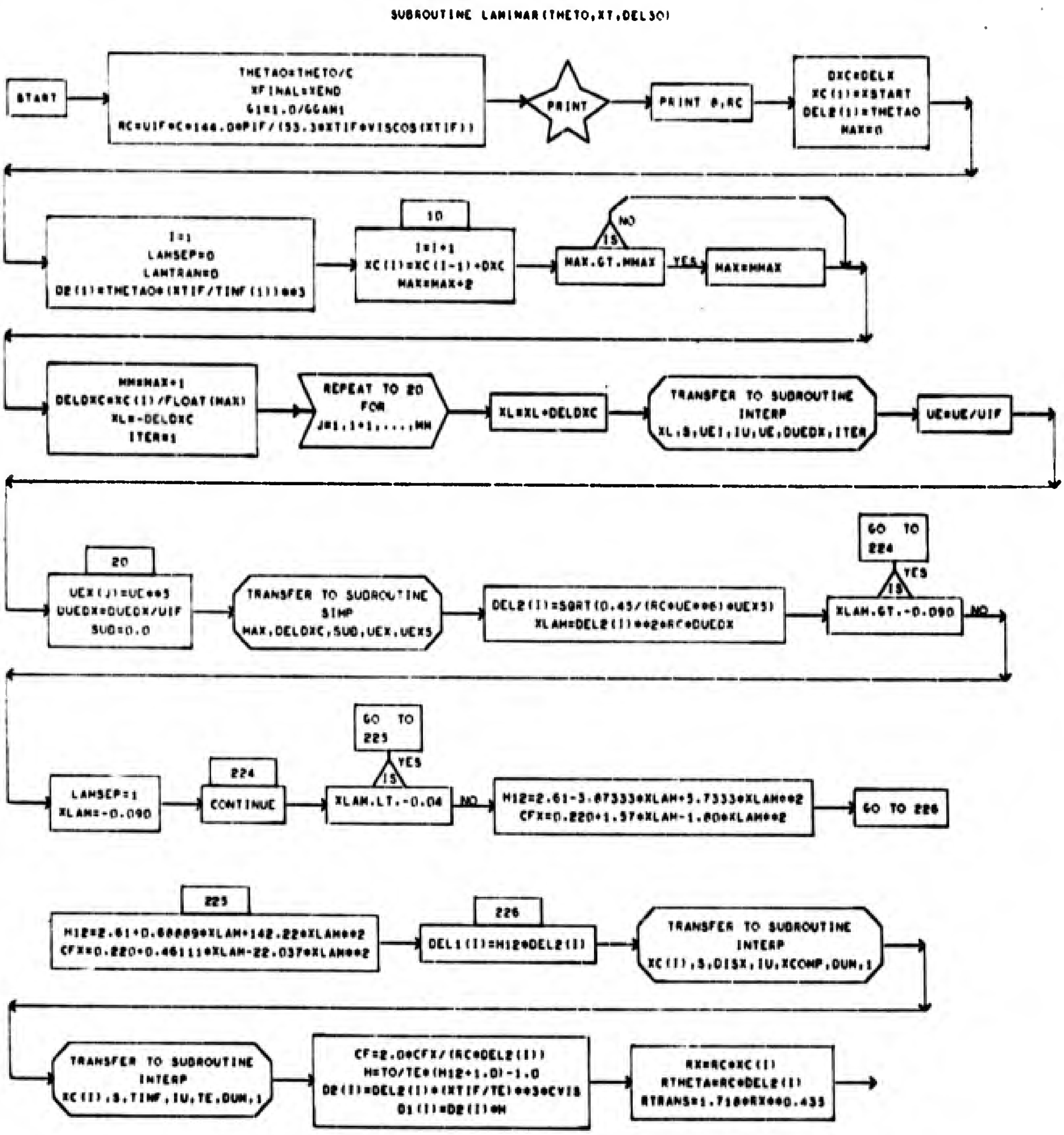
Detailed Flow Charts of Subroutine TURBL (5 of 7)



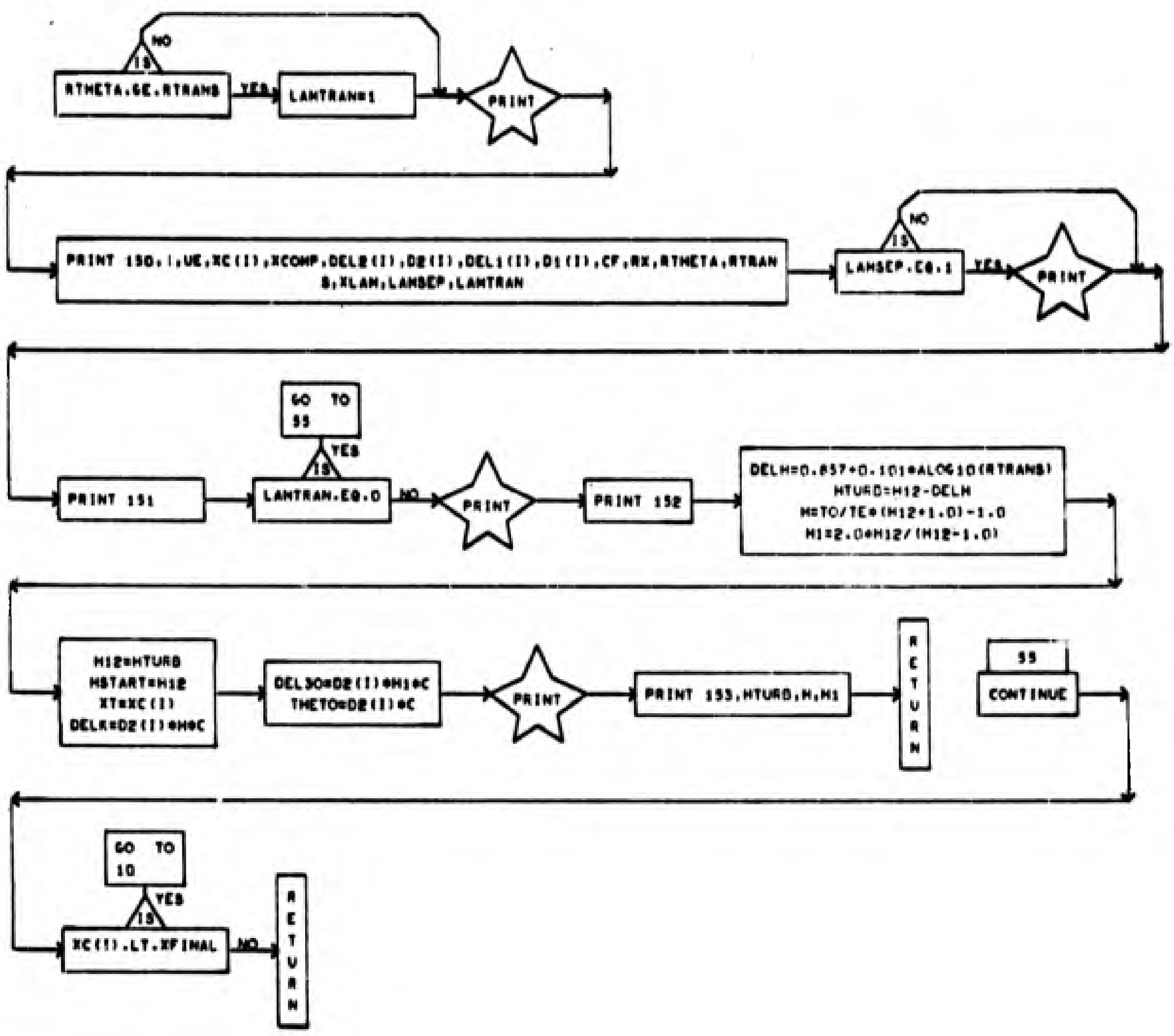
Detailed Flow Charts of Subroutine TURBL (6 of 7)



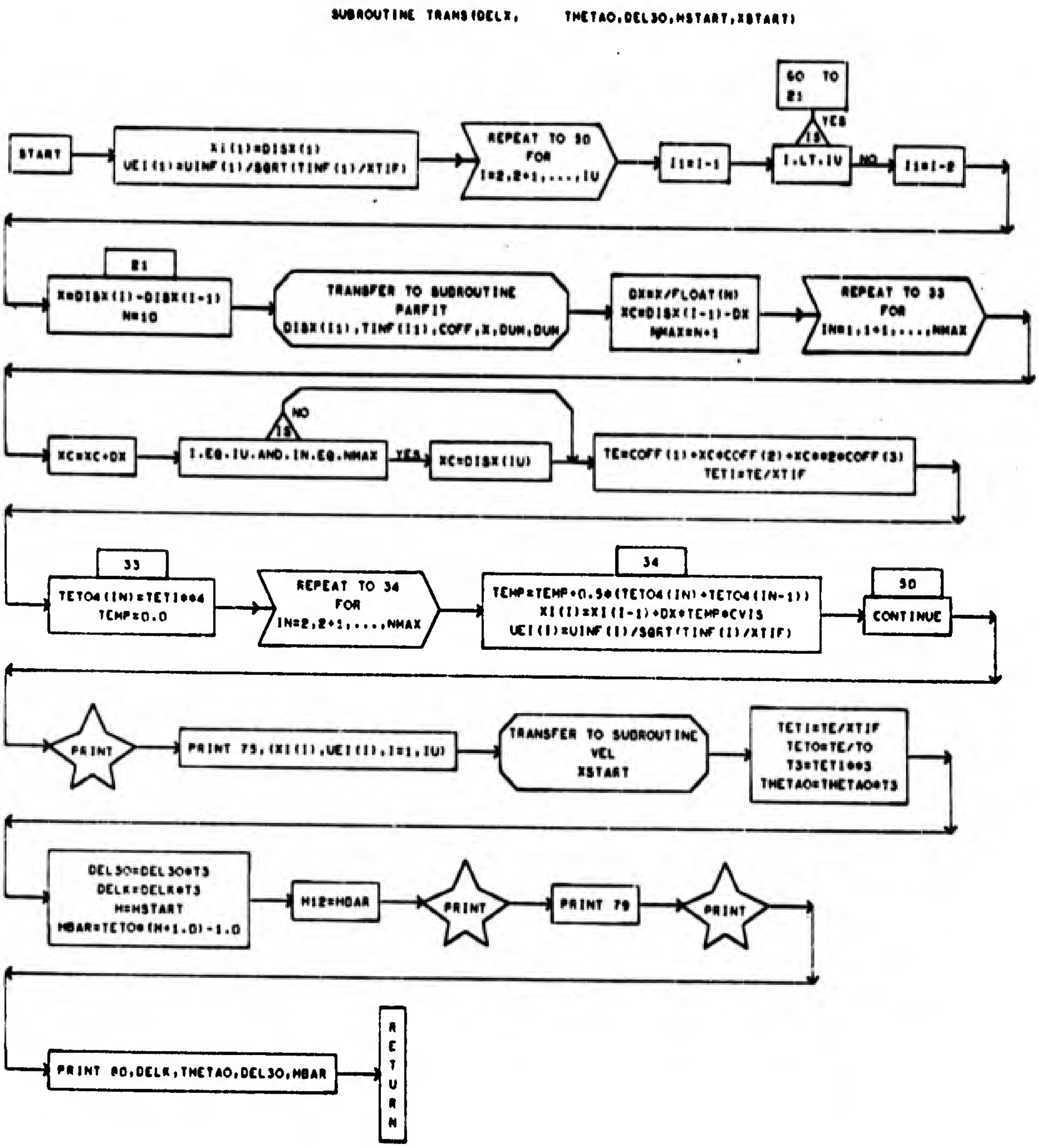
Detailed Flow Charts of Subroutine TURBL (7 of 7)



Detailed Flow Charts of Subroutine LAMINAR (1 of 2)

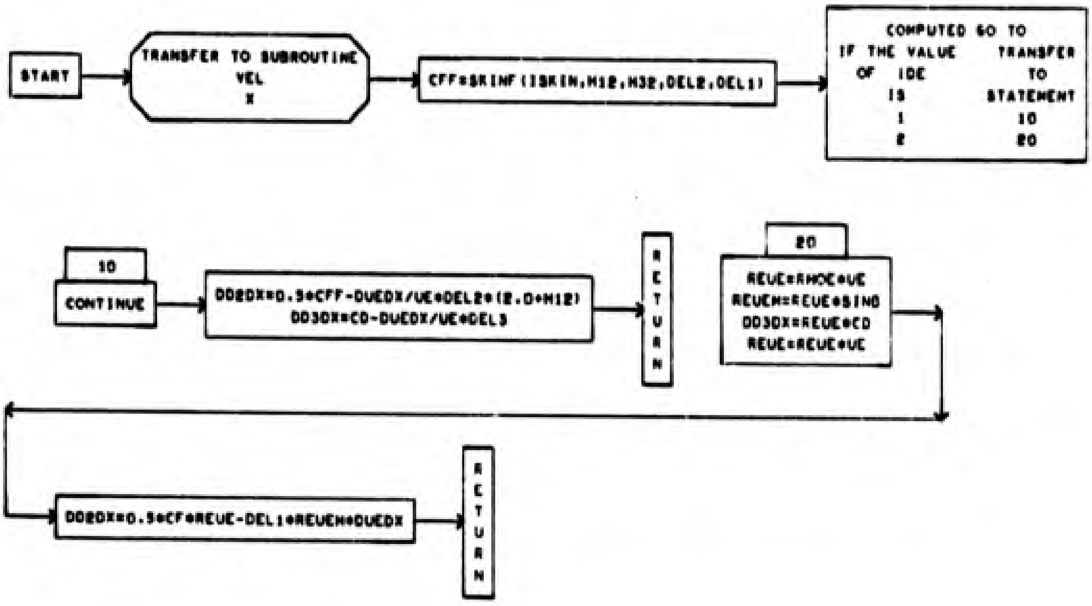


Detailed Flow Charts of Subroutine LAMINAR (2 of 2)

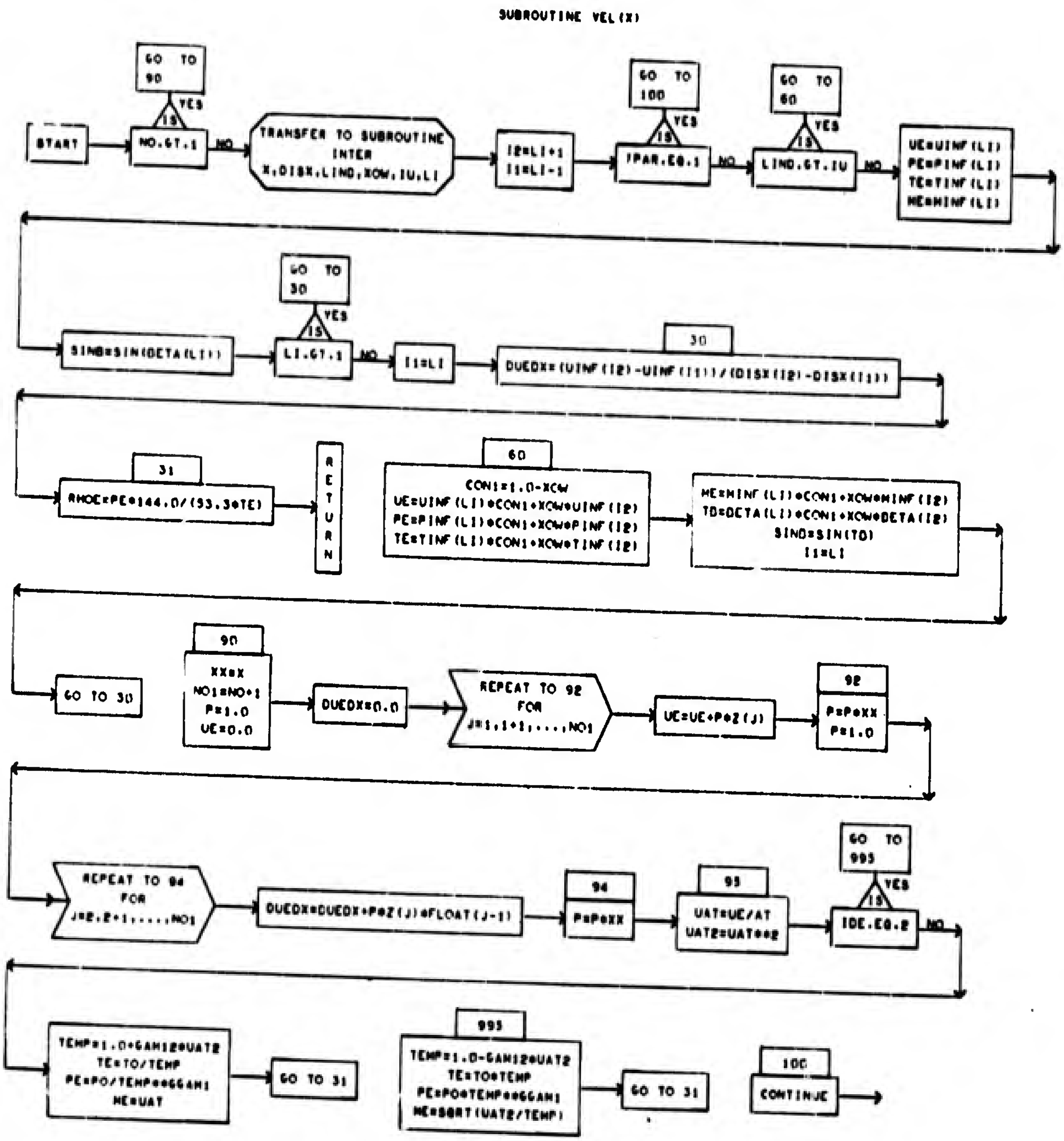


Detailed Flow Chart of Subroutine TRANS

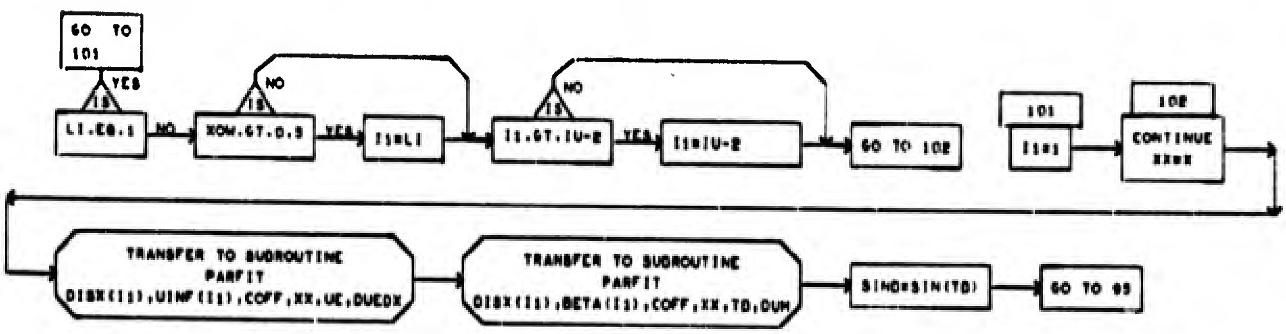
SUBROUTINE SLOPED(X,H12,H32,DEL1,DEL2,DEL3,DD2DX,DD3DX)



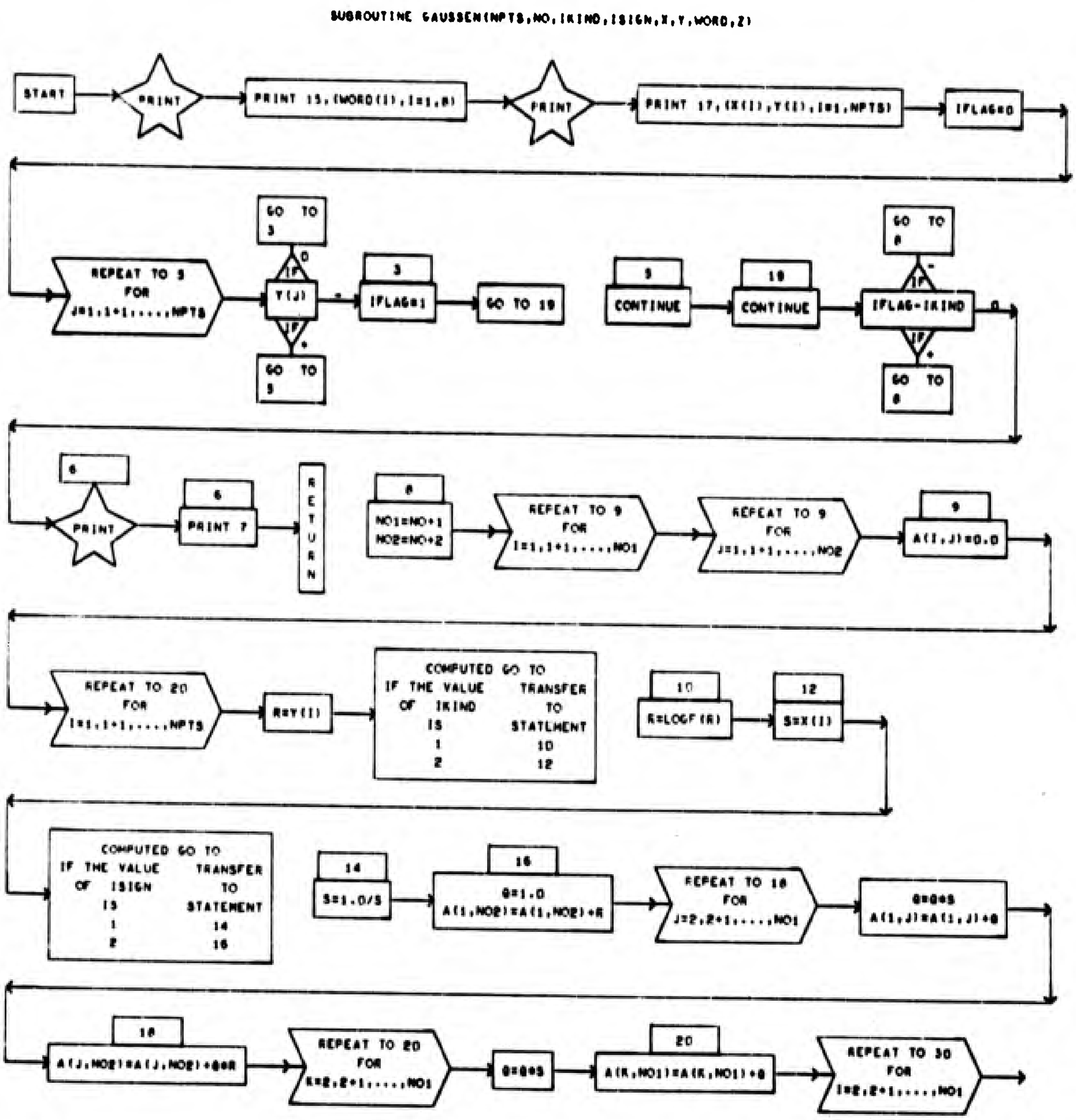
Detailed Flow Chart of Subroutine SLOPED



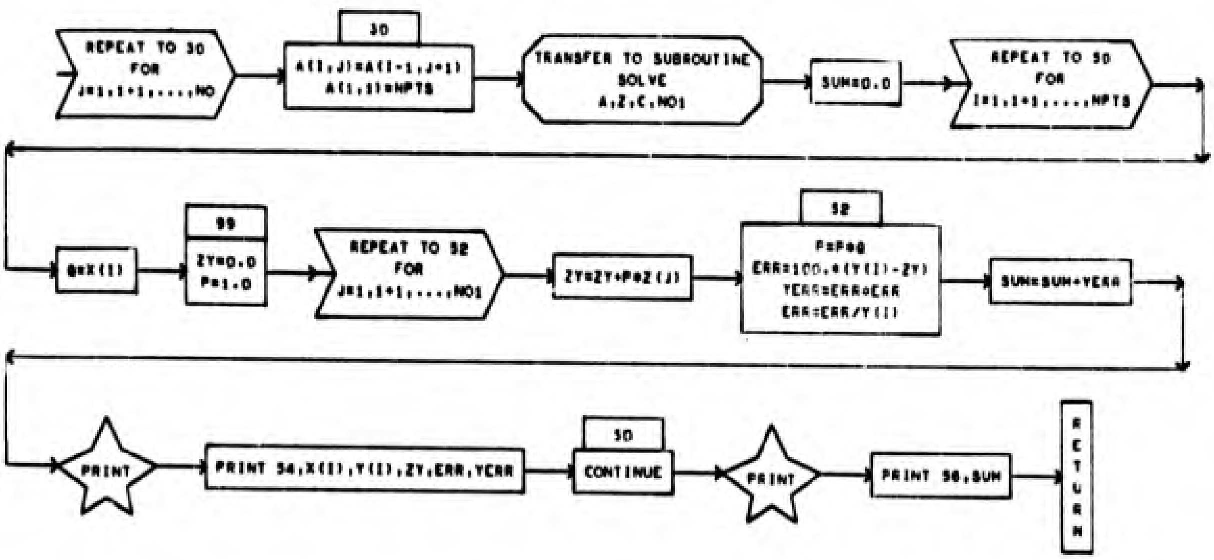
Detailed Flow Charts of Subroutine VEL (1 of 2)



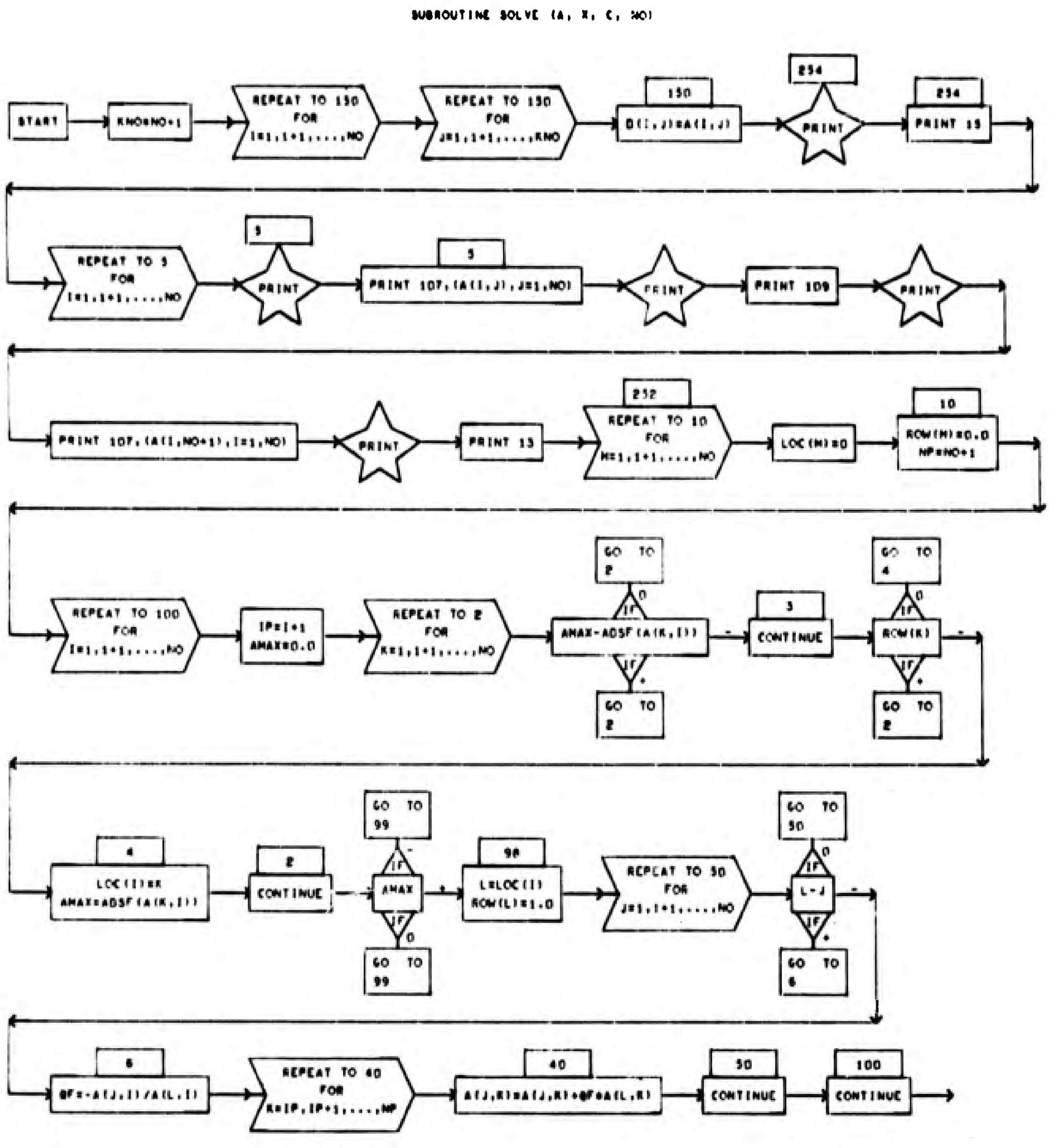
Detailed Flow Charts of Subroutine VEL (2 of 2)



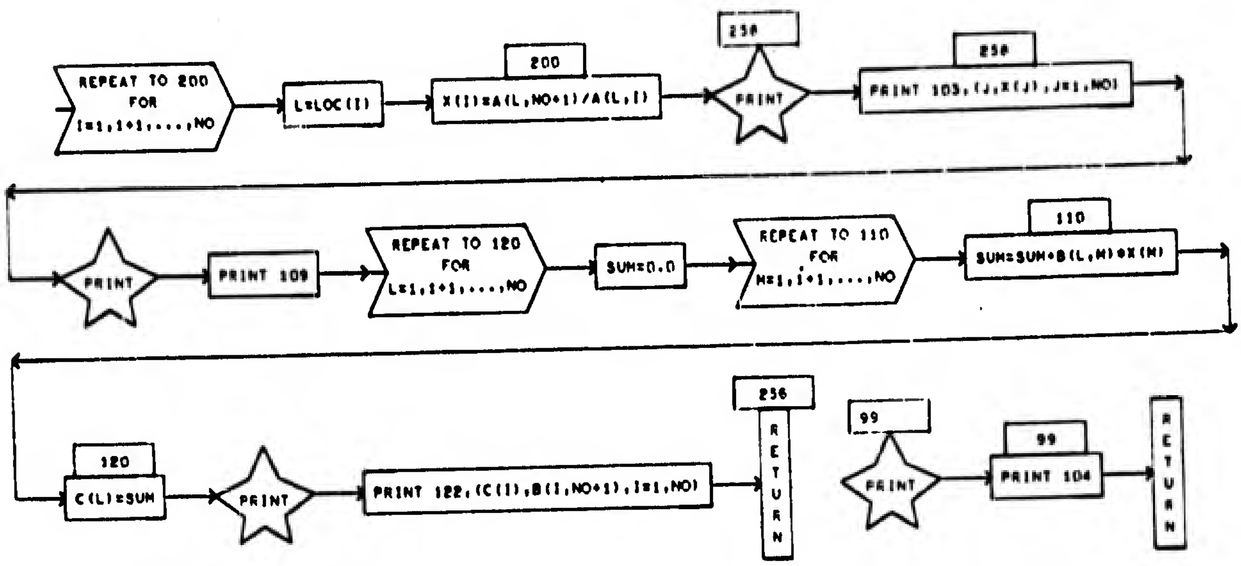
Detailed Flow Charts of Subroutine GAUSSEN (1 of 2)



Detailed Flow Charts of Subroutine GAUSSEN (2 of 2)

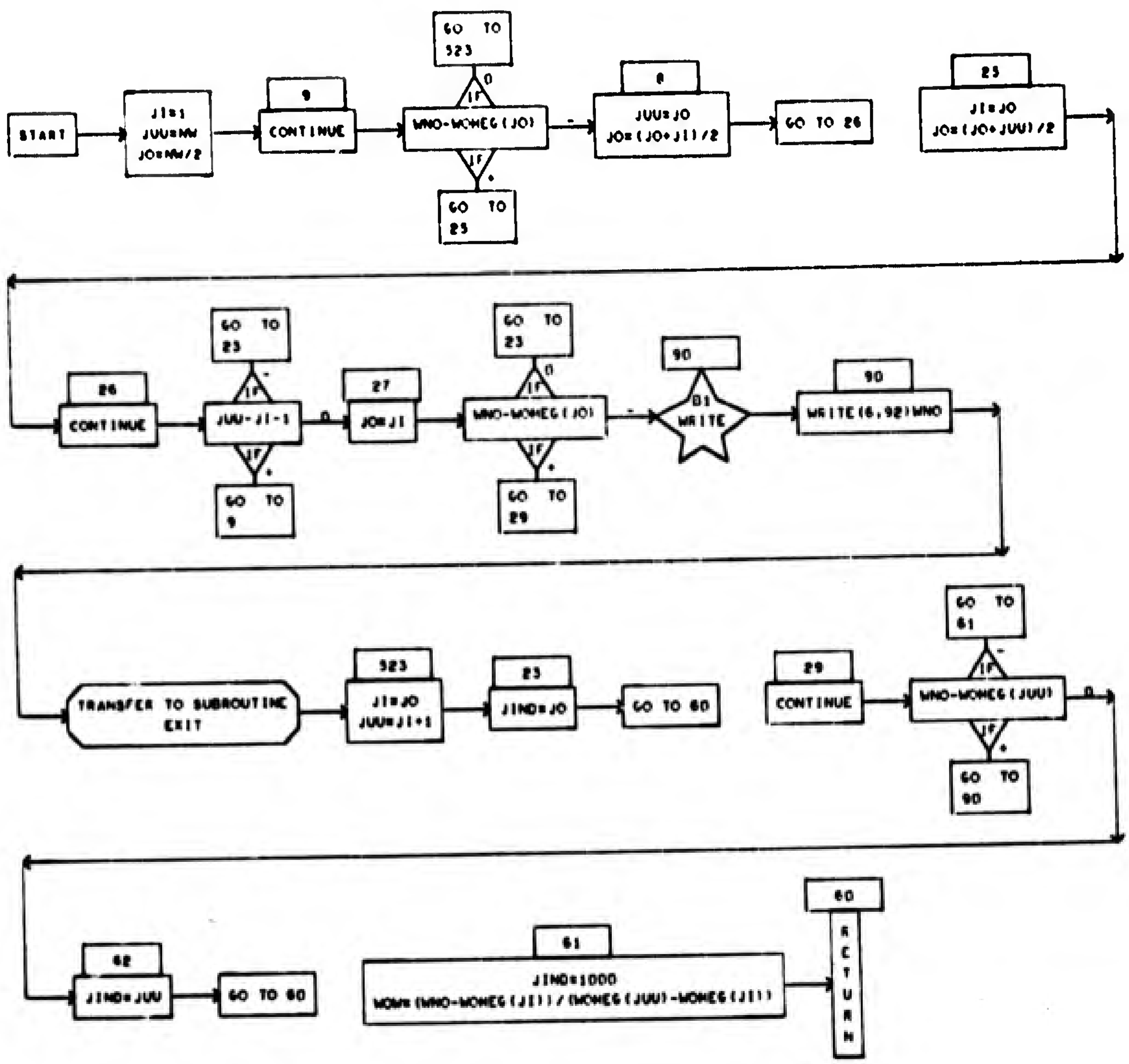


Detailed Flow Charts of Subroutine SOLVE (1 of 2)

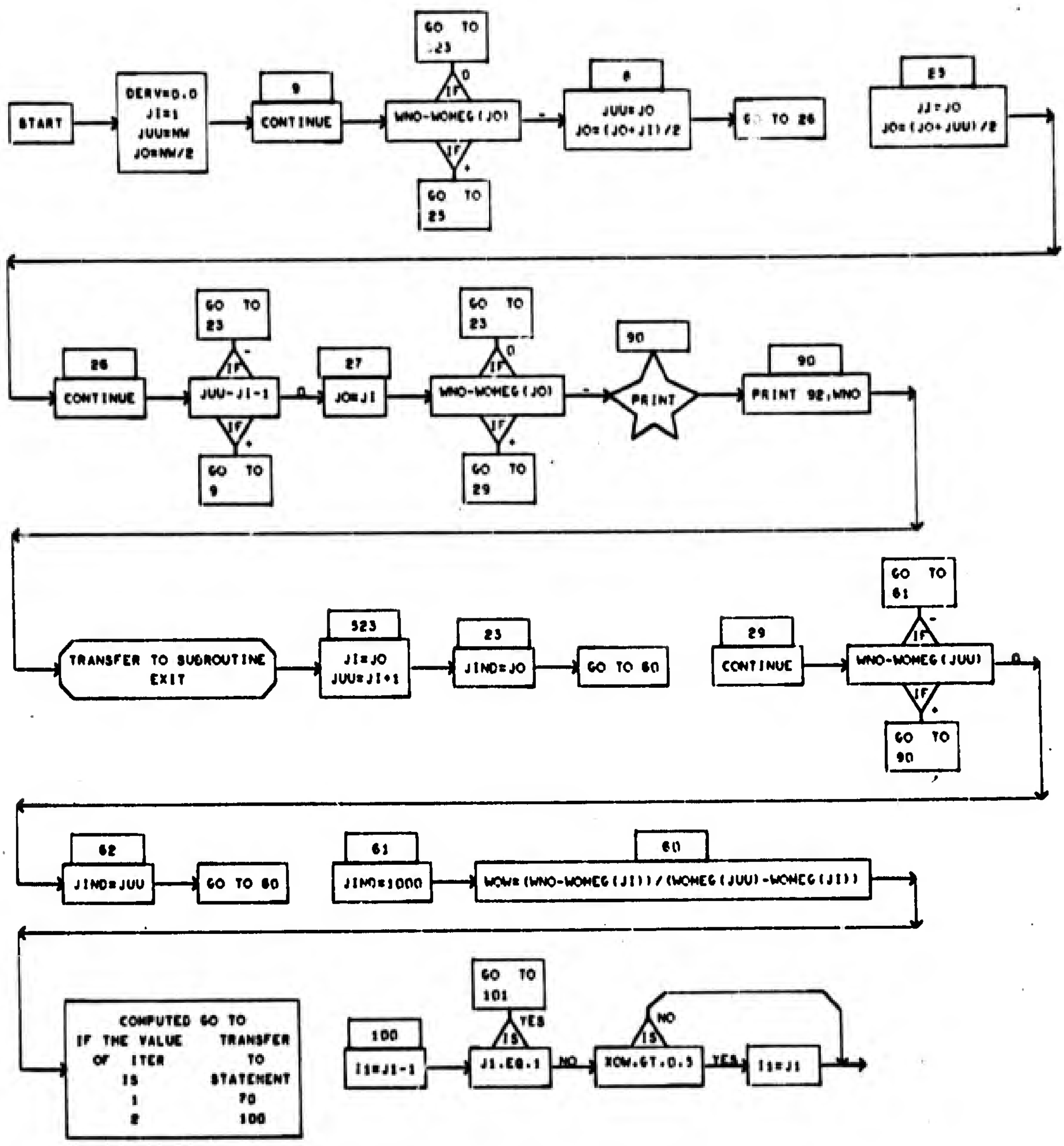


Detailed Flow Charts of Subroutine SOLVE (2 of 2)

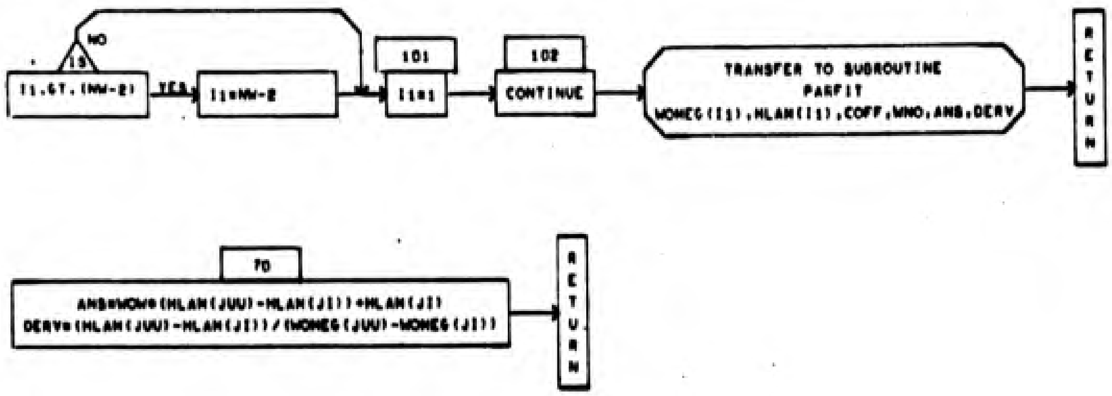
SUBROUTINE INTER (WNO,WONEG,JINO,WOW,NW,JI)



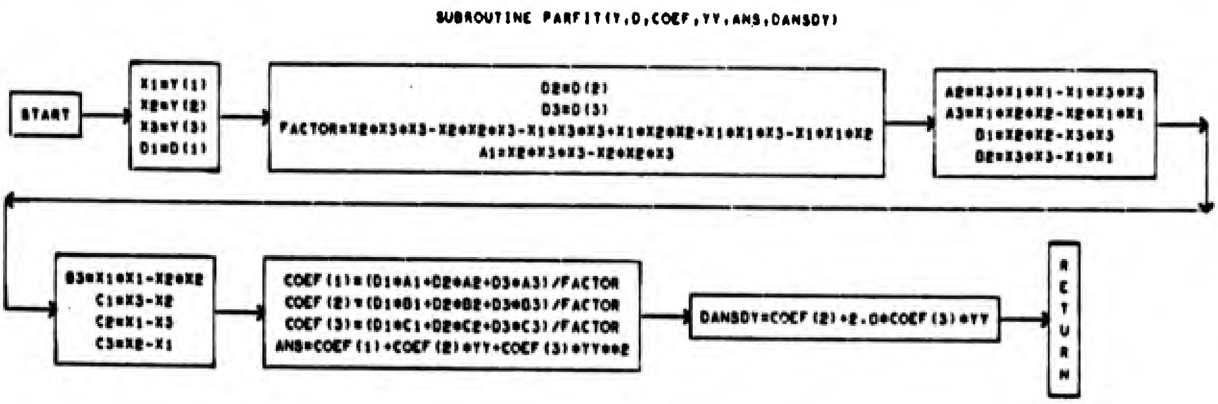
Detailed Flow Charts of Subroutine INTER



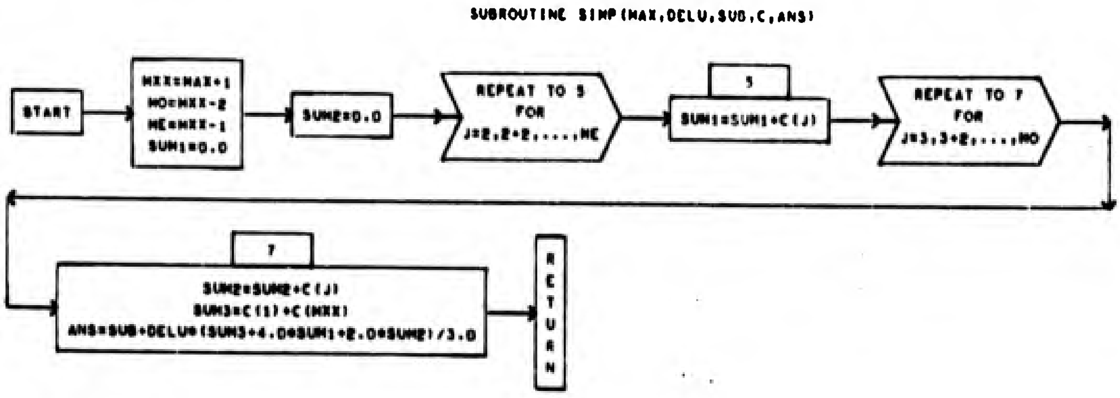
Detailed Flow Charts of Subroutine INTERP (1 of 2)



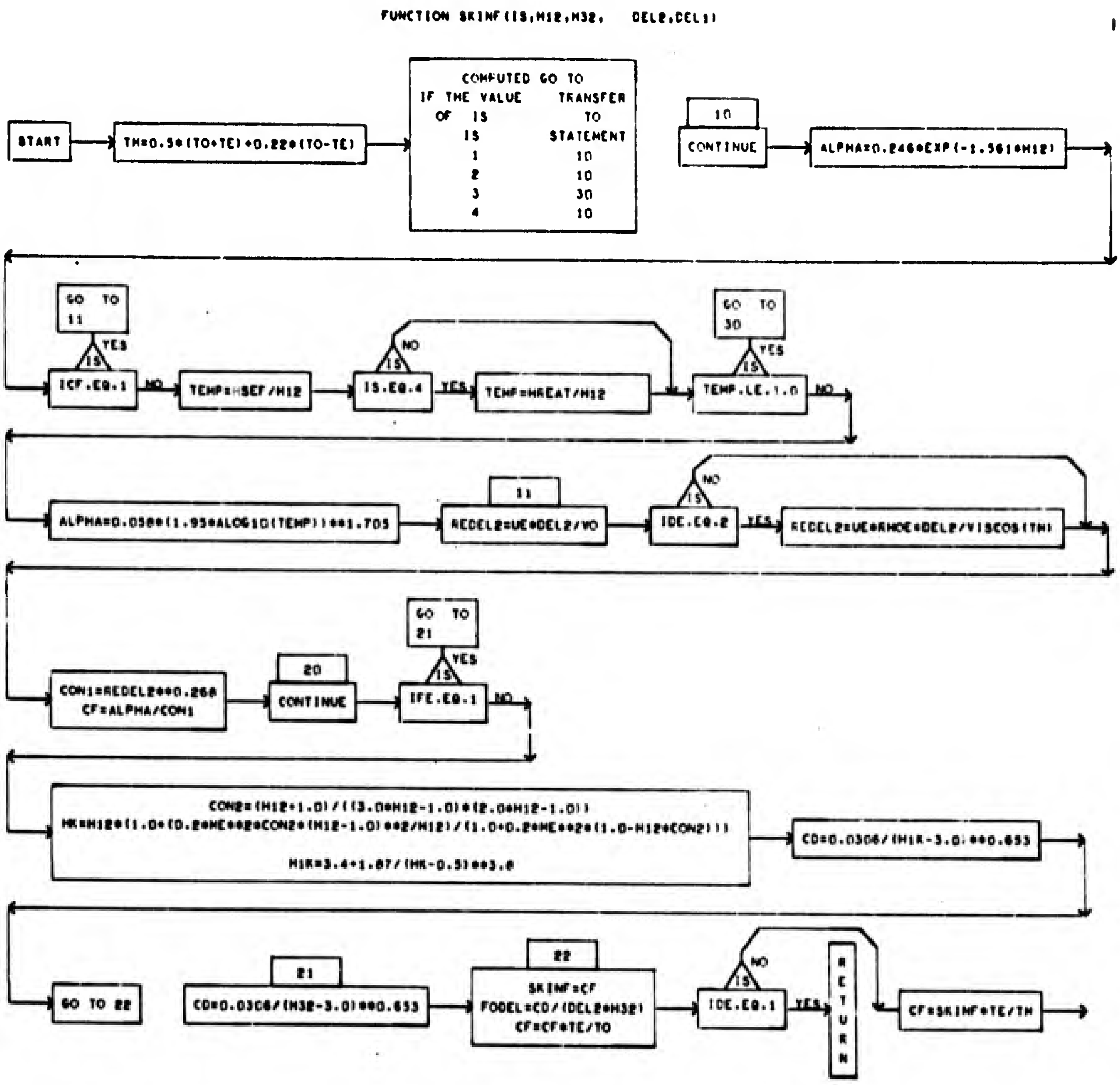
Detailed Flow Charts of Subroutine INTERP (2 of 2)



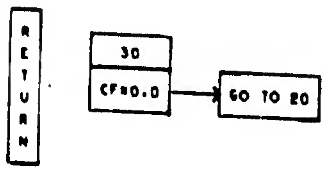
Detailed Flow Chart of Subroutine PARFIT



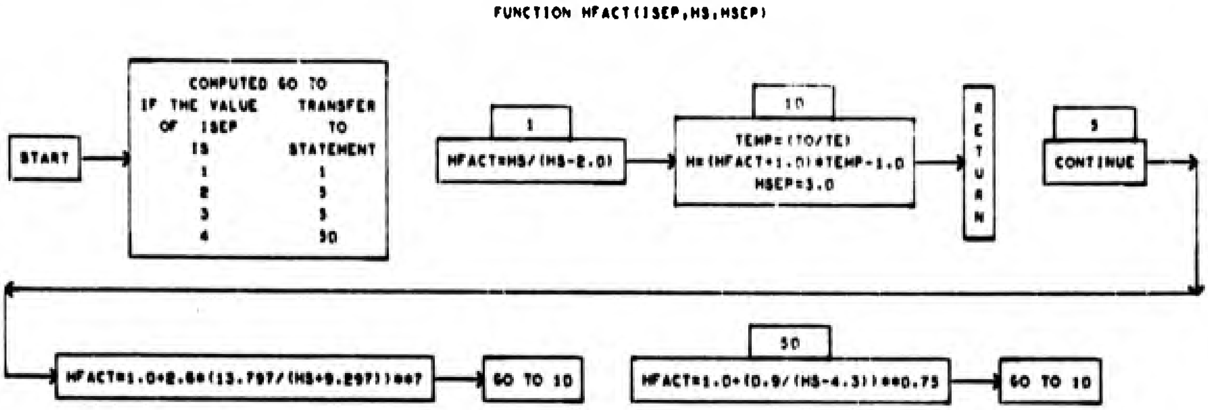
Detailed Flow Chart of Subroutine SIMP



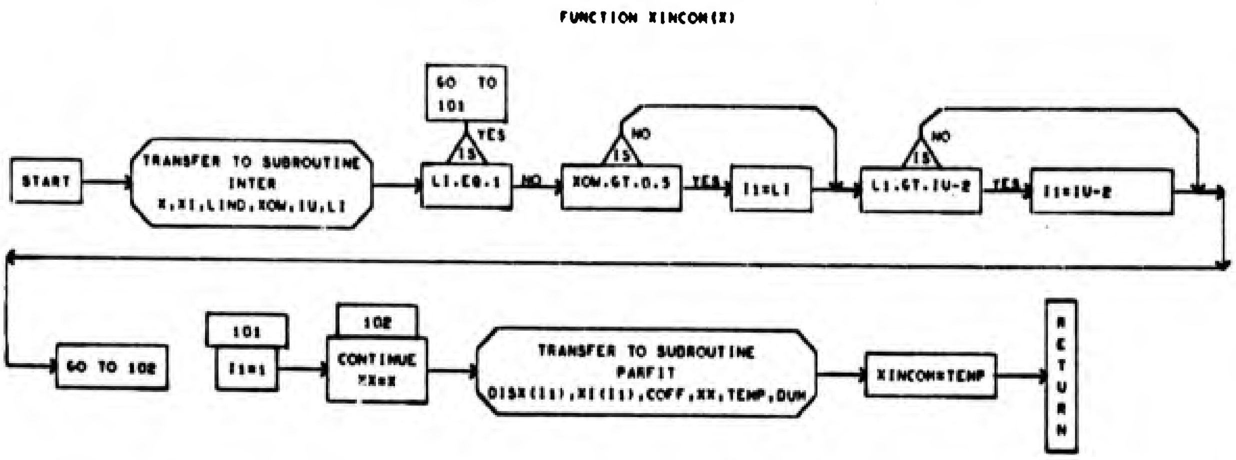
Detailed Flow Charts of Function SKINF (1 of 2)



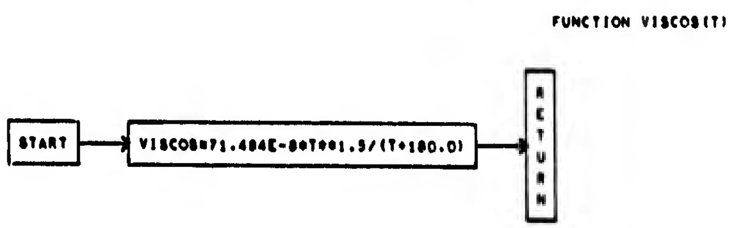
Detailed Flow Charts of Function SKINF (2 of 2)



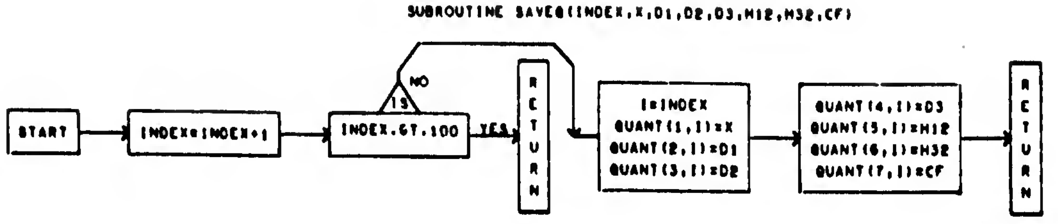
Detailed Flow Charts of Function HFACT



Detailed Flow Charts of Function XINCOM



Detailed Flow Charts of Function VISCOS



Detailed Flow Charts of Subroutine SAVEQ

REFERENCES

1. Lipmann, H. W. and Roshko, A., Elements of Gasdynamics, Wiley, N. Y., 1957.

2. Hirt, C. W., "Heuristic Stability Theory for Finite Difference Equations," J. of Comp. Physics, Vol. 2, No. 4, pp 335-355, June 1968.

3. Abbott, Ira H. and Von Doenhoff, Albert E., Theory of Wing Sections, McGraww Hill, N. Y., 1949.

4. Stivers, L., Effects of Subsonic Mach Number on the Forces and Pressure Distribution of Four NACA 64A-Series Airfoil Sections, NACA TN 3162, 1954.

5. Thwaites, B., Incompressible Aerodynamics, Oxford University Press, London, 1960.

6. Schlichting, H., Boundary-Layer Theory, McGraw Hill, New York, 1960, Page 580.

7. Ludwieg, H., Tillmann, W., "Untersuchungen uber die Wandschubspannung in Turbulenten Reibungsschichten," Ing. - Arch. 17, 1949, pp. 288-299.

8. Felsch, K. O., Contributions to the Calculation of Turbulent Boundary Layers in Two-Dimensional Incompressible Flow, Karlsruhe, T. H. Dissertation 1965.

9. Head, M. R., Entrainment in the Turbulent Boundary Layers, ARC R&M No. 3151, 1960.

10. Green, J. E., "The Prediction of Turbulent Boundary Layer Development in Compressible Flow," J. of Fluid Mech., Vol. 31, Part 4, 1968.

11. Seddon, J., Flow Produced by Interaction of a Turbulent Boundary Layer with a Normal Shock Wave, RAE TM Aero 667, 1960.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) CONVAIR DIVISION, GENERAL DYNAMICS P.O. Box 1128 San Diego, Calif. 92112	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP N/A

3. REPORT TITLE
FLOW OVER AIRFOILS IN THE TRANSONIC REGIME
Volume II - Computer Programs

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final Report, (November 1968 through December 1969)

5. AUTHOR(S) (First name, middle initial, last name)
Magnus, Richard J.
Gallaher, William H.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
March 1970	349	11

8a. CONTRACT OR GRANT NO. F33615-69-C-1180 b. PROJECT NO. 1366 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) AFFDL-TR-70-16 Volume II 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Distribution of this Document is Unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	AIR FORCE FLIGHT DYNAMICS LABORATORY WRIGHT-PATTERSON AFB, OHIO 45433

13. ABSTRACT

Digital computer programs for solving two problems are described. The first is for calculating planar inviscid transonic flow over an airfoil by a finite difference method and the second is for calculating the boundary layer on an airfoil for an arbitrary surface velocity distribution by an integral method. The physical problems are discussed in Volume I; Volume II contains descriptions of the programs and subroutines, program listings and flow charts, and discussion of some of the program features.

Key Words

Inviscid Flow
Boundary Layer
Computer Program
Finite Difference
Planar Flow
Transonic Flow
Airfoil

DD FORM 1 NOV 65 1473

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT