

AFFDL-TR-70-16
VOLUME II

FLOW OVER AIRFOILS IN THE TRANSONIC REGIME - COMPUTER PROGRAMS

Richard J. Magnus
William H. Gallagher

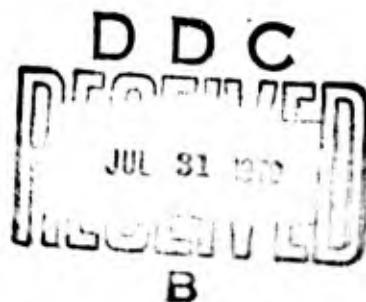
CONVAIR DIVISION OF GENERAL DYNAMICS

AD 709378

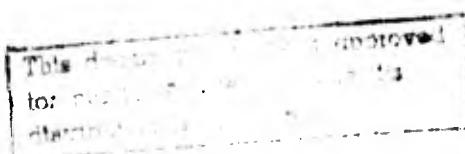
TECHNICAL REPORT AFFDL-TR-70-16, VOLUME II

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.

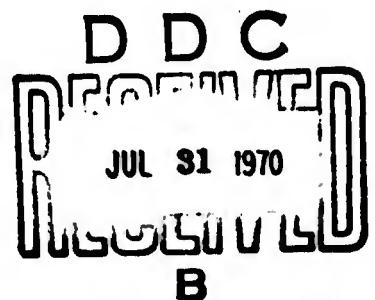


Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AFFDL-TR-70-16
VOLUME II

FLOW OVER AIRFOILS IN THE TRANSONIC REGIME-- COMPUTER PROGRAMS

Richard J. Magnus
William H. Gallagher



Distribution of this Document is Unlimited

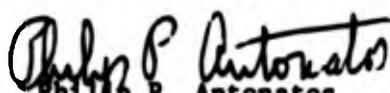
This document has been approved
for public release and authorization
distribution is unlimited

FOREWORD

The present study was undertaken by the Convair Division of General Dynamics, P. O. Box 1128, San Diego, California 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

<u>Section</u>		<u>Page</u>
I.	INTRODUCTION	1
II.	DESCRIPTION OF THE INVISCID FLOW PROBLEM	2
	1. Basic Equations	2
	2. Boundary Conditions	4
	3. Finite Difference Procedure at Field Points	4
	4. Finite Difference Procedure at Boundaries	14
	5. Allowable Time Step	18
	6. Time to Reach Steady Flow	19
III.	APPLICATION TO THE TRANSONIC FLOW OVER AN AIRFOIL	21
	1. Airfoil Description	21
	2. Spatial Field Description	21
	3. Results of Calculations	22

		<u>Description</u>	<u>Listing</u>	<u>Flow Chart</u>
IV.	SETUP PROGRAM	31	87	129
	1. Description of Program	31	-	-
	1.1 Main Routine TEHAI	31	87	130
	1.2 Subroutine SHAPE	32	114	167
	1.3 Subroutine CAXIS	33	120	176

<u>Section</u>		<u>Description</u>	<u>Listing</u>	<u>Flow Chart</u>
1.4	Subroutine PONDER	34	121	177
1.5	Subroutine BUNTE	36	123	179
1.6	Subroutine DISTW	38	124	180
1.7	Subroutine WAIF	38	127	183
1.8	Subroutine CARTW	38	128	184
2.	Input Data	39	-	-
3.	Output Data	39	-	-
4.	Operating Instructions	39	-	-
V.	FLOW FIELD PROGRAM	43	185	236
1.	Description of Program	43	-	-
1.1	Main Routine HANE	43	186	237
1.2	Subroutine REPAIR	44	-	-
1.3	Subroutine OUCH	44	192	244
1.4	Subroutine PUTOUT	44	195	247
1.5	Subroutine AIROUT	45	196	248
1.6	Subroutine YTRE	45	197	249
1.7	Subroutine OOKII	46	200	252
1.8	Subroutine SUSUMU	46	203	255
1.9	Subroutine WOP	50	209	264
1.10	Subroutine WOAP	50	211	265
1.11	Subroutine HERI	50	212	266

<u>Section</u>		<u>Description</u>	<u>Listing</u>	<u>Flow Chart</u>
	1.12 Subroutine JIKAN	52	215	270
	1.13 Subroutine ENGS	53	217	271
	1.14 Subroutine SAKAI	53	218	272
	1.15 Subroutine KAKOI	56	231	285
	1.16 Subroutine WAKU	57	232	286
	1.17 Subroutine KAKARU	58	234	288
	1.18 Subroutine TREND	58	235	289
	2. Input Data	58	-	-
	3. Output Data	61	-	-
	4. Operating Instructions	61	-	-
VI.	BOUNDARY LAYER CALCULATION PROGRAM	63	290	318
	1. Program Description	63	-	-
	1.1 Program TURBL	63	291	319
	1.2 Subroutine LAMINAR	64	298	326
	1.3 Subroutine TRANS	64	300	328
	1.4 Subroutine SLOPED	65	302	329
	1.5 Subroutine VEL	65	303	330
	1.6 Subroutine GAUSSEN	66	305	332
	1.7 Subroutine SOLVE	66	307	334
	1.8 Subroutine INTER	66	309	336
	1.9 Subroutine INTERP	66	310	337

<u>Section</u>		<u>Description</u>	<u>Listing</u>	<u>Flow Chart</u>
1.10	Subroutine PARFIT	66	311	339
1.11	Subroutine SIMP	66	312	340
1.12	Function SKINF	67	313	341
1.13	Function HFACT	67	314	343
1.14	Function XINCOM	67	315	344
1.15	Function VISCOM	67	316	345
1.16	Subroutine SAVEQ	67	317	346
2.	Program Input	68	-	-

		<u>Page</u>
APPENDIX I	DISCUSSION OF PRINCIPAL FORTRAN VARIABLES	72
APPENDIX II	SETUP PROGRAM FORTRAN LISTING	86
APPENDIX III	SETUP PROGRAM FLOW CHARTS	129
APPENDIX IV	FLOW FIELD PROGRAM LISTING	185
APPENDIX V	FLOW FIELD PROGRAM FLOW CHARTS	236
APPENDIX VI	BOUNDARY LAYER PROGRAM LISTING	290
APPENDIX VII	BOUNDARY LAYER PROGRAM FLOW CHARTS	318
REFERENCES		347

LIST OF ILLUSTRATIONS

	<u>Page</u>
1. Illustration of Application of Difference Scheme to a Simple One-Dimensional Unsteady Problem	5
2. Construction Points Used by Symmetric Difference Scheme for Advancing Solution in Cartesian Mesh	9
3. Construction Points Used by Difference Scheme for Advancing Solution in Local Rotated Mesh Tangent to Airfoil Surface	15
4. Arrangement of Mesh Regions Around NACA 64A-410 Airfoil	
(a) Outer Regions	23
(b) Inner Regions Around Nose	25
5. Pressure Distribution on NACA 64A-410 Airfoil at $M = 0.72$, $\alpha = 4^\circ$	27
6. Mach Number Contours of the Flowfield About a NACA 64A-410 Airfoil at $M = 0.72$, $\alpha = 4^\circ$	
(a) General View	28
(b) Detail Around Nose	29
7. Relation of Local Rotated Mesh to Cartesian Mesh and Distorted Mesh	35
8. Special Features of Mesh Interrelations Where the Airfoil Penetrates a Mesh Size Change	37
9. Typical Arrangement of Damping Augmentation Regions	49
10. Cell Types Used in Computer Program	51

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
R _h	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

Subscripts

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

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

Momentum

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

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

Energy

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

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

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

$$p = (\gamma - 1) \rho e \quad (6)$$

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:

$$\left(\frac{p}{p_0}\right)^\gamma = \left(\frac{\rho}{\rho_0}\right)^\gamma \quad (7)$$

where p_0 and ρ_0 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 $H_0 = \frac{1}{\gamma - 1} + \frac{1}{2} M_\infty^2$

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

$$W_t = F_x + G_y \quad (8)$$

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

$$W = \begin{bmatrix} \rho u \\ \rho v \\ \rho \end{bmatrix} \quad F = - \begin{bmatrix} \rho u^2 + p \\ \rho u v \\ \rho u \end{bmatrix} \quad G = - \begin{bmatrix} \rho u v \\ \rho v^2 + p \\ \rho v \end{bmatrix} \quad (9)$$

2. BOUNDARY CONDITIONS

For the boundary conditions far from the airfoil, the flow is required to approach a uniform flow with the conditions $u = U \cos \alpha$, $v = U \sin \alpha$, 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

$$\langle U_1^{1/2} \rangle = U_1^0 + \frac{\alpha \Delta t}{2} \left(\frac{3U_1^0 - 4U_0^0 + U_{-1}^0}{2 \Delta X} \right)$$

SECOND STEP

$$\langle U_{-1}^{1/2} \rangle = U_{-1}^0 + \frac{\alpha \Delta t}{2} \left(\frac{-U_1^0 + 4U_0^0 - 3U_{-1}^0}{2 \Delta X} \right) + \alpha \Delta t \left(\frac{\langle U_1^0 + U_{-1}^0 \rangle}{2} \right)$$

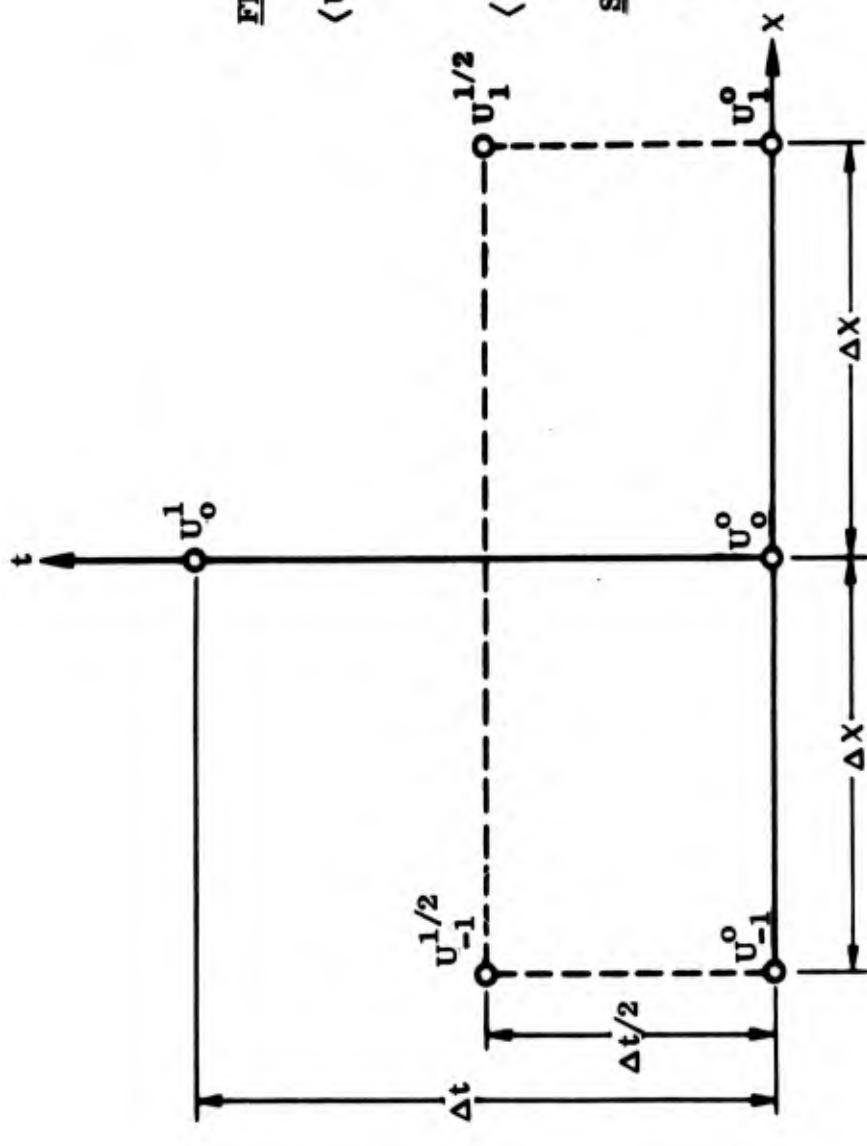


Figure 1. Illustration of application of difference scheme to a simple one-dimensional unsteady problem.

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:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial u}{\partial x} \quad (10)$$

where α is a constant. Considering the basic lattice element shown in Fig. 1, the marching procedure for determining the value of u_0^1 knowing the values u_0^0 , u_1^0 , and u_{-1}^0 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:

$$u_1^{1/2} = u_1^0 + \frac{\alpha \Delta t}{2} \left(\frac{3u_1^0 - 4u_0^0 + u_{-1}^0}{2\Delta x} \right)$$

$$u_{-1}^{1/2} = u_{-1}^0 + \frac{\alpha \Delta t}{2} \left(\frac{-u_1^0 + 4u_0^0 - 3u_{-1}^0}{2\Delta x} \right) \quad (11)$$

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_0^1 in terms of the known initial values by the Taylor expansion:

$$u_0^1 = \left[(1 - k) u_0^0 + k \left(\frac{u_1^0 + u_{-1}^0}{2} \right) \right] + \alpha \Delta t \left(\frac{u_1^{1/2} - u_{-1}^{1/2}}{2\Delta x} \right) \quad (12)$$

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 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 stability¹. 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 plane². The former modification for non-zero values of k has an essential advantage of enhancing 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 $w_{1,0}^{1/2}$, $w_{-1,0}^{1/2}$, $w_{0,1}^{1/2}$, and $w_{0,-1}^{1/2}$ would be calculated using equations:

$$w_{1,0}^{1/2} = w_{1,0}^0 + \frac{\Delta t}{2} \left[\frac{3F_{1,0}^0 - 4F_{0,0}^0 + F_{-1,0}^0}{2\Delta x} + \frac{G_{1,1}^0 - G_{1,-1}^0}{2\Delta y} \right]$$

²This 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.

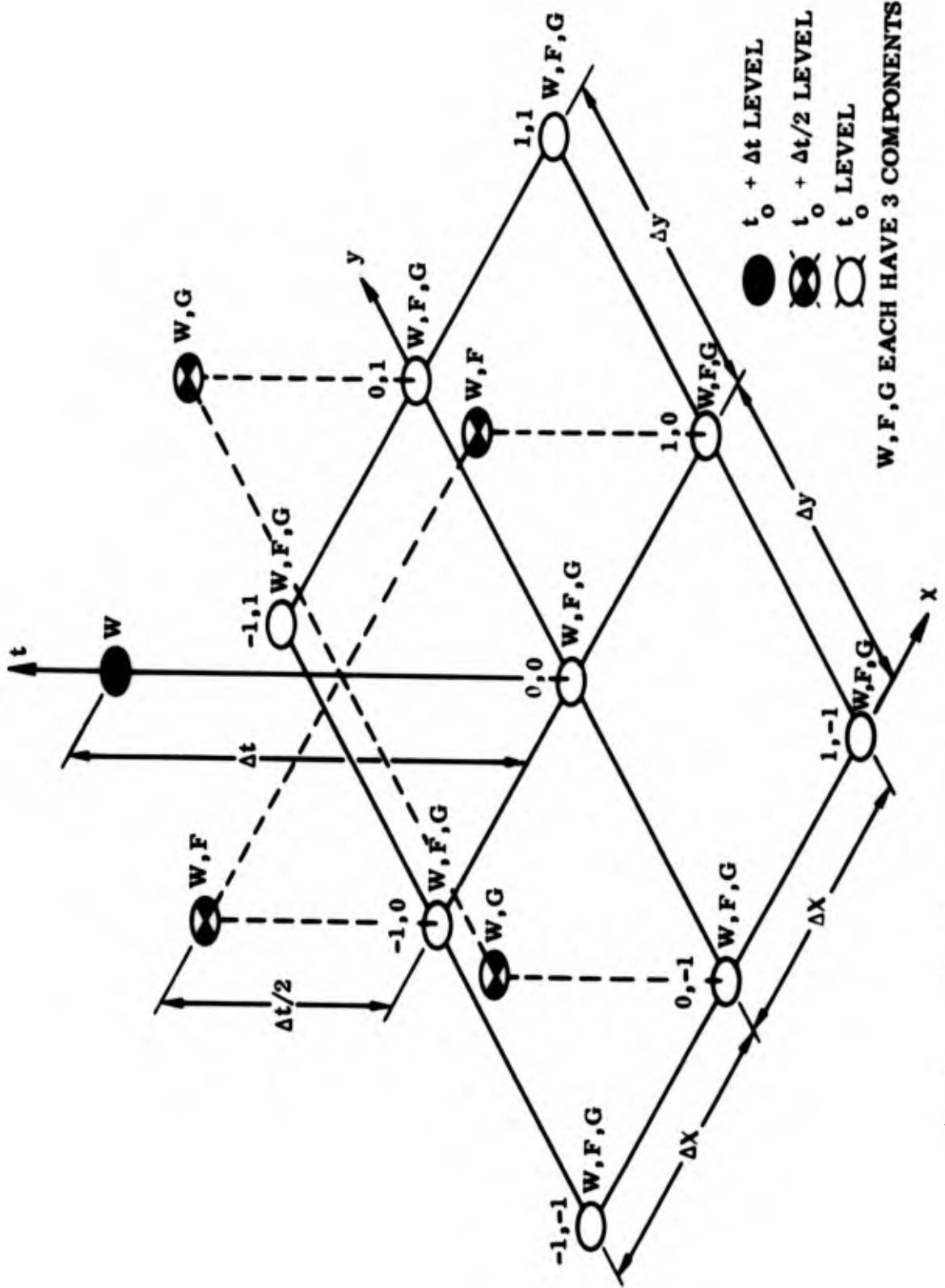


Figure 2. Construction points used by symmetric difference scheme for advancing solution in Cartesian mesh.

$$w_{-1,0}^{1/2} = w_{-1,0}^o + \frac{\Delta t}{2} \left[\frac{-3F_{-1,0}^o + 4F_{0,0}^o - F_{1,0}^o}{2\Delta x} + \frac{G_{-1,1}^o - G_{-1,-1}^o}{2\Delta y} \right]$$

$$w_{0,1}^{1/2} = w_{0,1}^o + \frac{\Delta t}{2} \left[\frac{F_{1,1}^o - F_{-1,1}^o}{2\Delta x} + \frac{3G_{0,1}^o - 4G_{0,0}^o + G_{0,-1}^o}{2\Delta y} \right]$$

$$w_{0,-1}^{1/2} = w_{0,-1}^o + \frac{\Delta t}{2} \left[\frac{F_{1,-1}^o - F_{-1,-1}^o}{2\Delta x} + \frac{-3G_{0,-1}^o + 4G_{0,0}^o - G_{0,1}^o}{2\Delta y} \right] \quad (13)$$

The components of $w_{1,0}^{1/2}$ and $w_{-1,0}^{1/2}$ are converted to $F_{1,0}^{1/2}$ and $F_{-1,0}^{1/2}$ and the $w_{0,1}^{1/2}$ and $w_{0,-1}^{1/2}$ converted to $G_{0,1}^{1/2}$ and $G_{0,-1}^{1/2}$ 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 :

$$\begin{aligned} w_{0,0}^1 &= \left\{ 1 - \frac{k}{2} \left[1 + \left(\frac{\Delta x}{\Delta y} \right)^2 \right] \right\} w_{0,0}^o + \frac{k}{4} \left[(w_{1,0}^o + w_{-1,0}^o) + \left(\frac{\Delta x}{\Delta y} \right)^2 (w_{0,1}^o + w_{0,-1}^o) \right] \\ &\quad + \Delta t \left[\frac{F_{1,0}^{1/2} - F_{-1,0}^{1/2}}{2\Delta x} + \frac{G_{0,1}^{1/2} - G_{0,-1}^{1/2}}{2\Delta y} \right] \end{aligned} \quad (14)$$

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

$$w_t = \alpha w_x + \beta w_y \quad (15)$$

where α , β are matrices whose elements are:

$\frac{\partial F_i}{\partial w_j}$ and $\frac{\partial G_i}{\partial w_j}$ respectively.

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

$$\begin{aligned} w_{0,0}^1 - w_{0,0}^0 &= \frac{k}{4} (\Delta x)^2 \left[(w_{xx} + \frac{\Delta x^2}{12} w_{xxxx} + \dots) + (w_{yy} + \frac{\Delta y^2}{12} w_{yyyy} + \dots) \right] \\ &\quad + \Delta t \left[\alpha (w_x + \frac{\Delta x^2}{6} w_{xxx} + \dots) + \beta (w_y + \frac{\Delta y^2}{6} w_{yyy} + \dots) \right] \\ &\quad + \frac{\Delta t^2}{2} \left[\alpha^2 (w_{xx} + \frac{\Delta x^2}{12} w_{xxxx} + \dots) + \right. \\ &\quad \left. + (\alpha\beta + \beta\alpha) \left(w_{xy} + \frac{\Delta x^2}{6} w_{xxx} + \frac{\Delta y^2}{6} w_{xyyy} + \dots \right) \right. \\ &\quad \left. + \beta^2 (w_{yy} + \frac{\Delta y^2}{12} w_{yyyy} + \dots) \right] \end{aligned} \quad (16)$$

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

$$\begin{aligned}
 w_{o,o}^1 - w_{o,o}^0 &= \Delta t \left[\alpha w_x + \beta w_y \right] + \frac{\Delta t^2}{2} \left[\alpha^2 w_{xx} + (\alpha\beta + \beta\alpha) w_{xy} + \beta^2 w_{yy} \right] \\
 &\quad + \frac{\Delta t^3}{6} \left[\alpha^3 w_{xxx} + (\alpha\alpha\beta + \alpha\beta\alpha + \beta\alpha\alpha) w_{xxy} + (\alpha\beta\beta + \beta\alpha\beta + \beta\beta\alpha) w_{xyy} \right. \\
 &\quad \left. + \beta^3 w_{yyy} \right] + \dots \tag{17}
 \end{aligned}$$

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:

$$\underline{\alpha w_x + \beta w_y} = 0,$$

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

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:

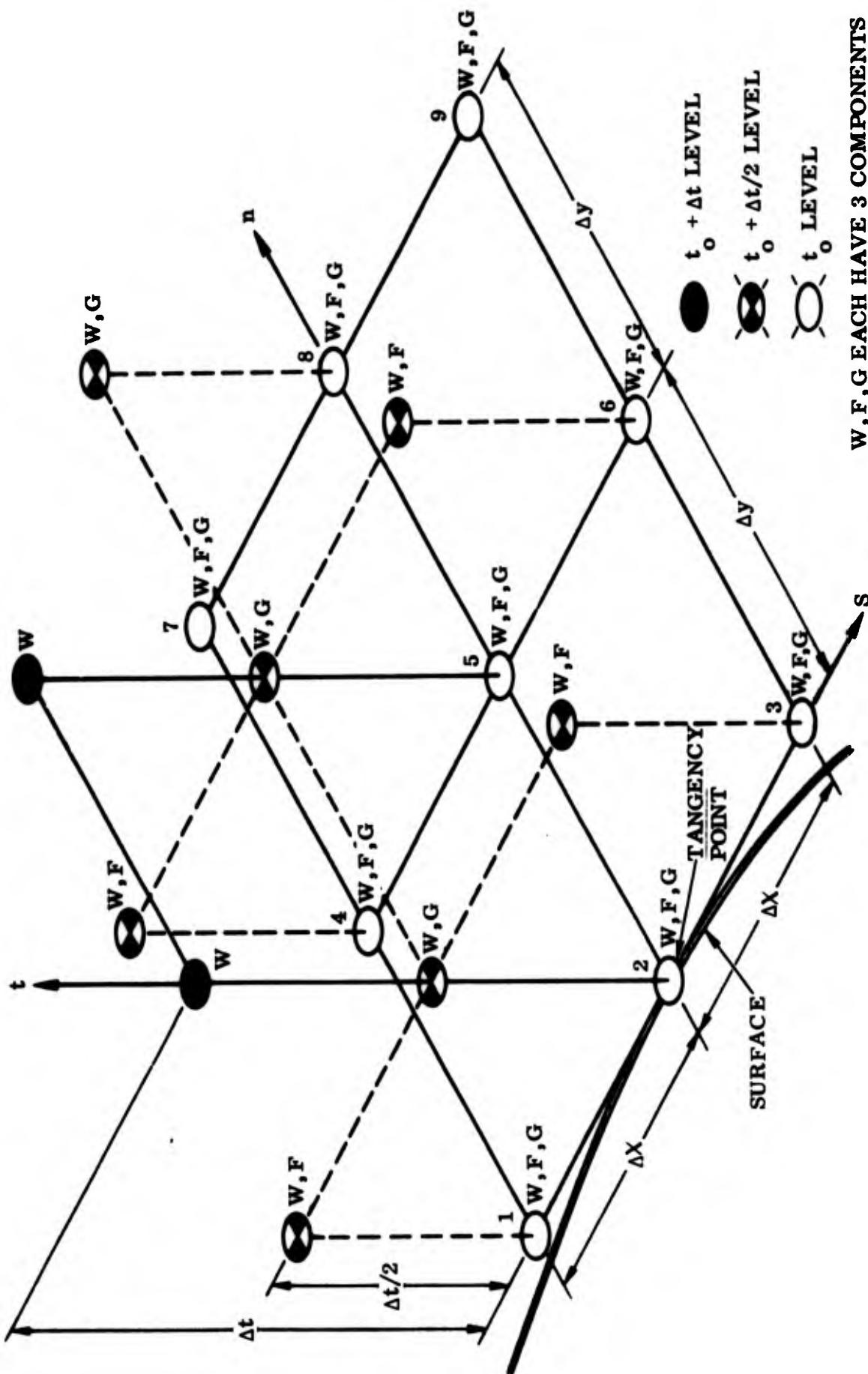


Figure 3. Construction points used by difference scheme for advancing solution in local rotated mesh tangent to airfoil surface.

$$\begin{aligned}
w_2^1 &= \left\{ 0.5 (1.0 - k) + 0.5 \left[1.0 - \left(\frac{\Delta x}{\Delta y} \right)^2 k \right] + 0.25 \left(\frac{\Delta x}{\Delta y} \right)^2 k \right\} w_2^0 \\
&\quad + 0.25 k (w_1^0 + w_3^0) + 0.5k \left(\frac{\Delta x}{\Delta y} \right)^2 w_5^0 - 0.25 k \left(\frac{\Delta x}{\Delta y} \right)^2 w_8^0 \\
&\quad + \frac{\Delta t}{2} \left\{ \frac{\frac{F_3^{1/2} - F_1^{1/2}}{2\Delta x} + \frac{-jG_2^{1/2} + 4G_5^{1/2} - G_8^{1/2}}{2\Delta y}}{} \right\} \tag{18}
\end{aligned}$$

The weights attached to the various w_k^0 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:

$$\begin{aligned}
w_{(1A)} &= \rho_A u_{tA} \\
w_{(2A)} &= \rho_A U_{nA} \\
w_{(3A)} &= \rho_A
\end{aligned} \tag{19}$$

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

$$\begin{aligned}
w_{(3B)} &= \left[\left(\frac{w_{(3A)}}{w_{(1A)}} \right)^{\frac{\gamma-1}{2}} - \frac{\gamma-1}{2} \frac{w_{(1A)}}{w_{(3A)}} \right]^{\frac{2}{\gamma-1}} \\
w_{(1B)} &= \frac{w_{(1A)}}{w_{(3A)}} w_{(3B)} \\
w_{(2B)} &= 0
\end{aligned} \tag{20}$$

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.

$$Rh = \frac{r}{r_0} = \frac{1.4}{\sqrt{x^2 + y^2}}$$

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

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).

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

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 flow-field. 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:

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

where a is the sound speed;

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

The criterion is roughly equivalent to the Courant-Freidrichs-Lowy criterion; referring to Figure 2, the backward Mach cone through point $(x_o, y_o, t_o + \Delta t)$ should intersect the t 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:

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

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

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

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:

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

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:

<u>Region</u>	<u>Identifier</u>	<u>Mesh Description</u>
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.

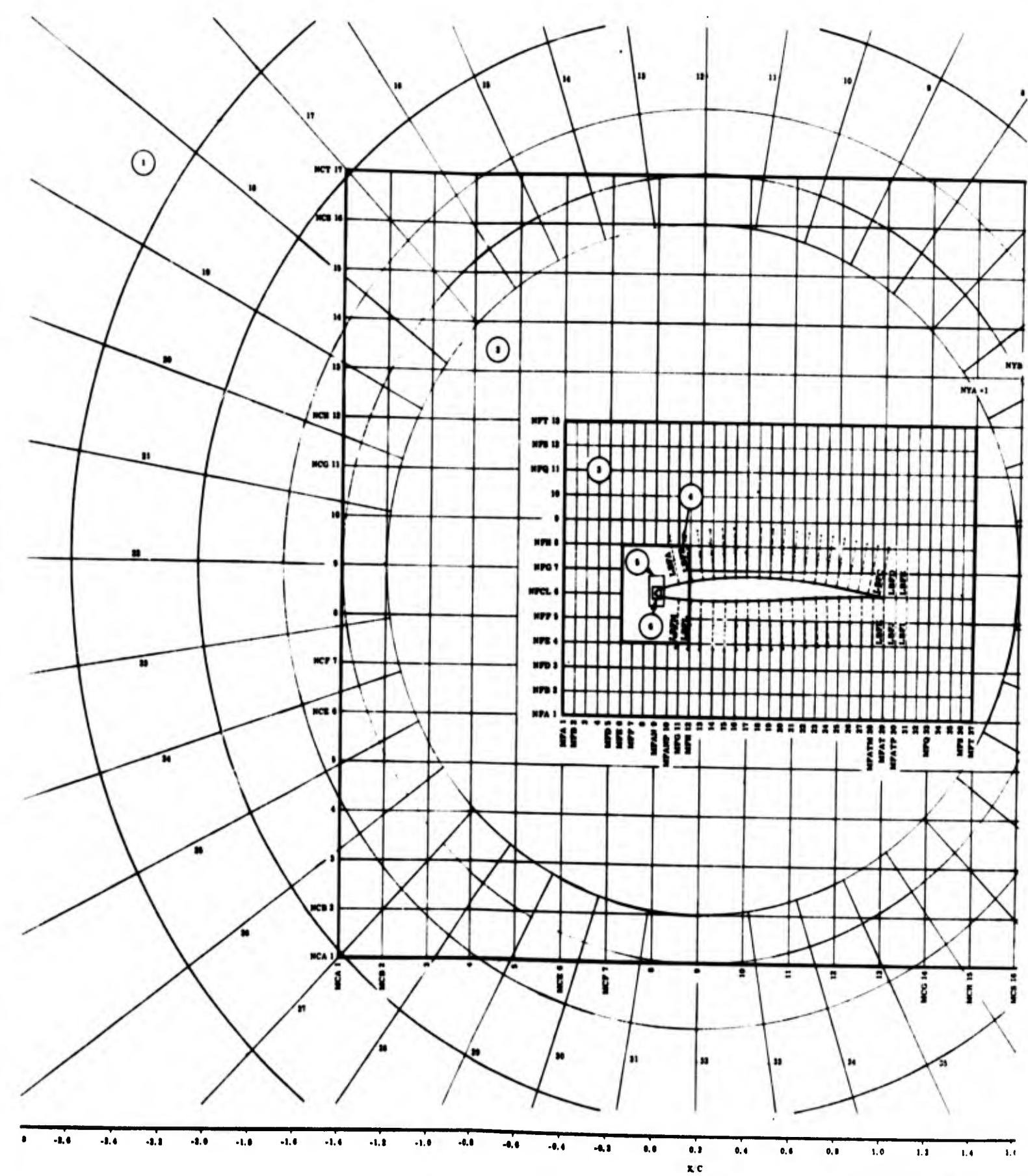


Figure 4. Arrangement o:
Airfoil. (a)

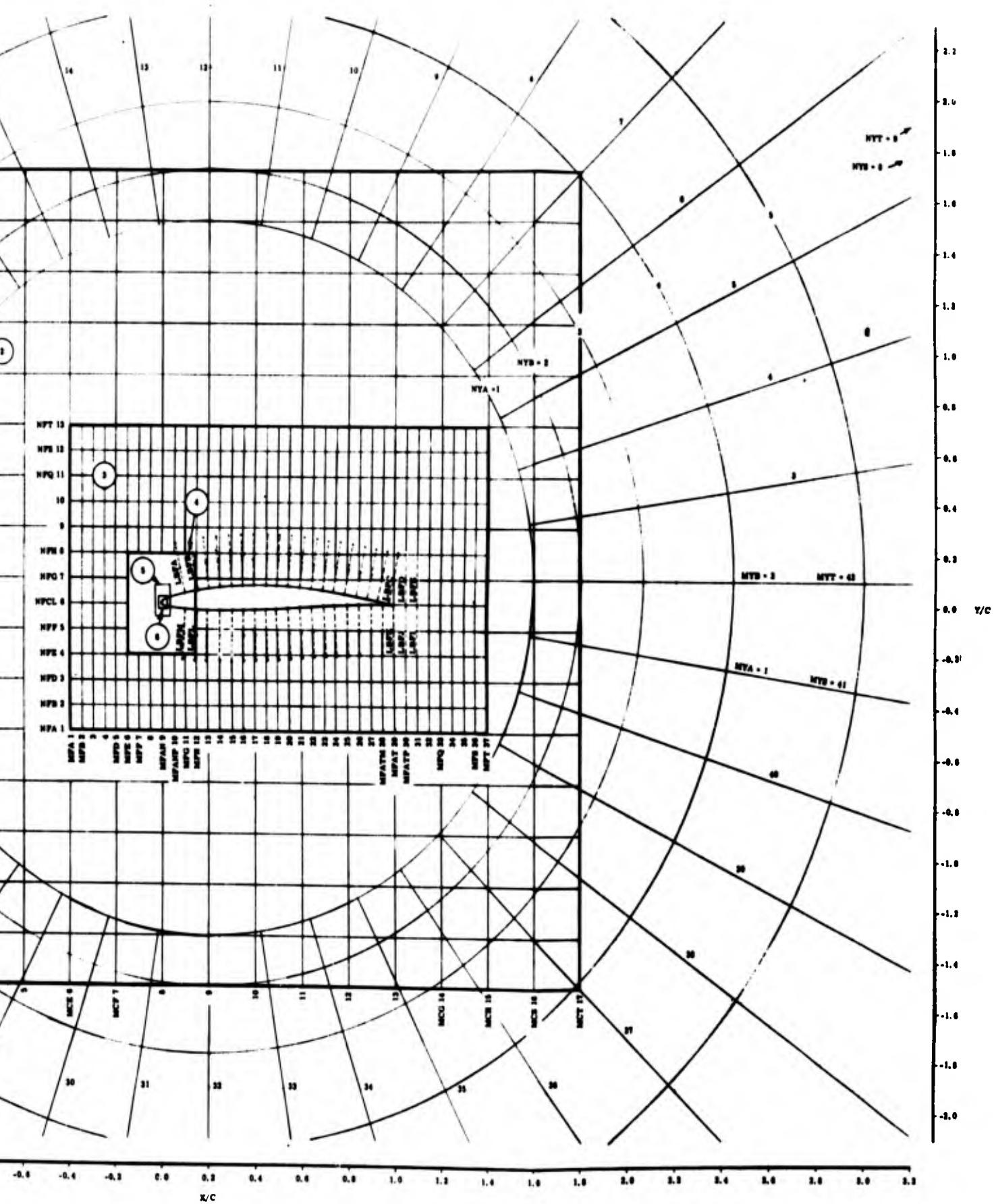


Figure 4. Arrangement of mesh regions around NACA 64A-410 Airfoil. (a) Outer regions.

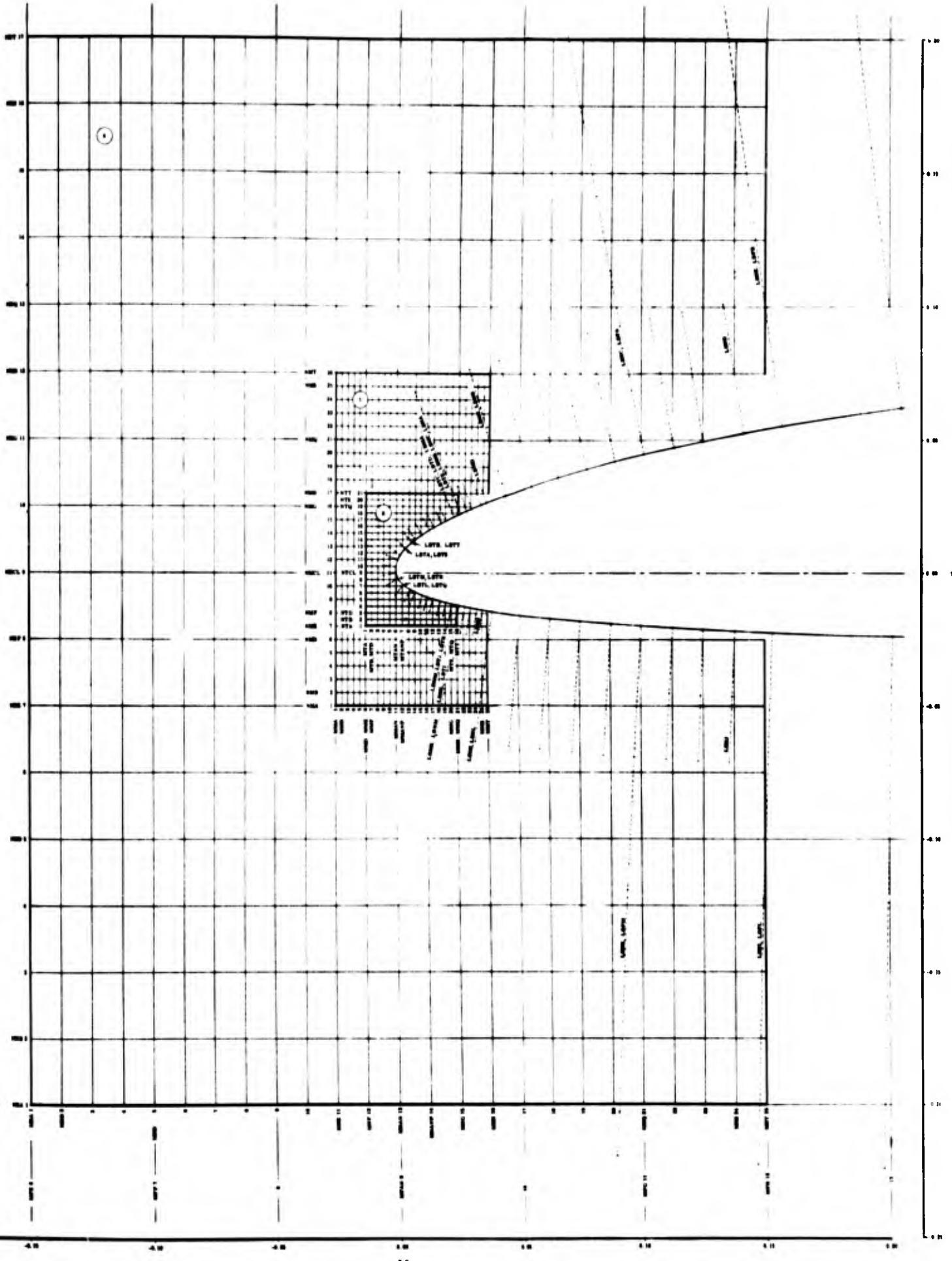


Figure 4 (concluded). Arrangement of mesh regions around NACA 64A-410 Airfoil. (b) Inner regions around nose.

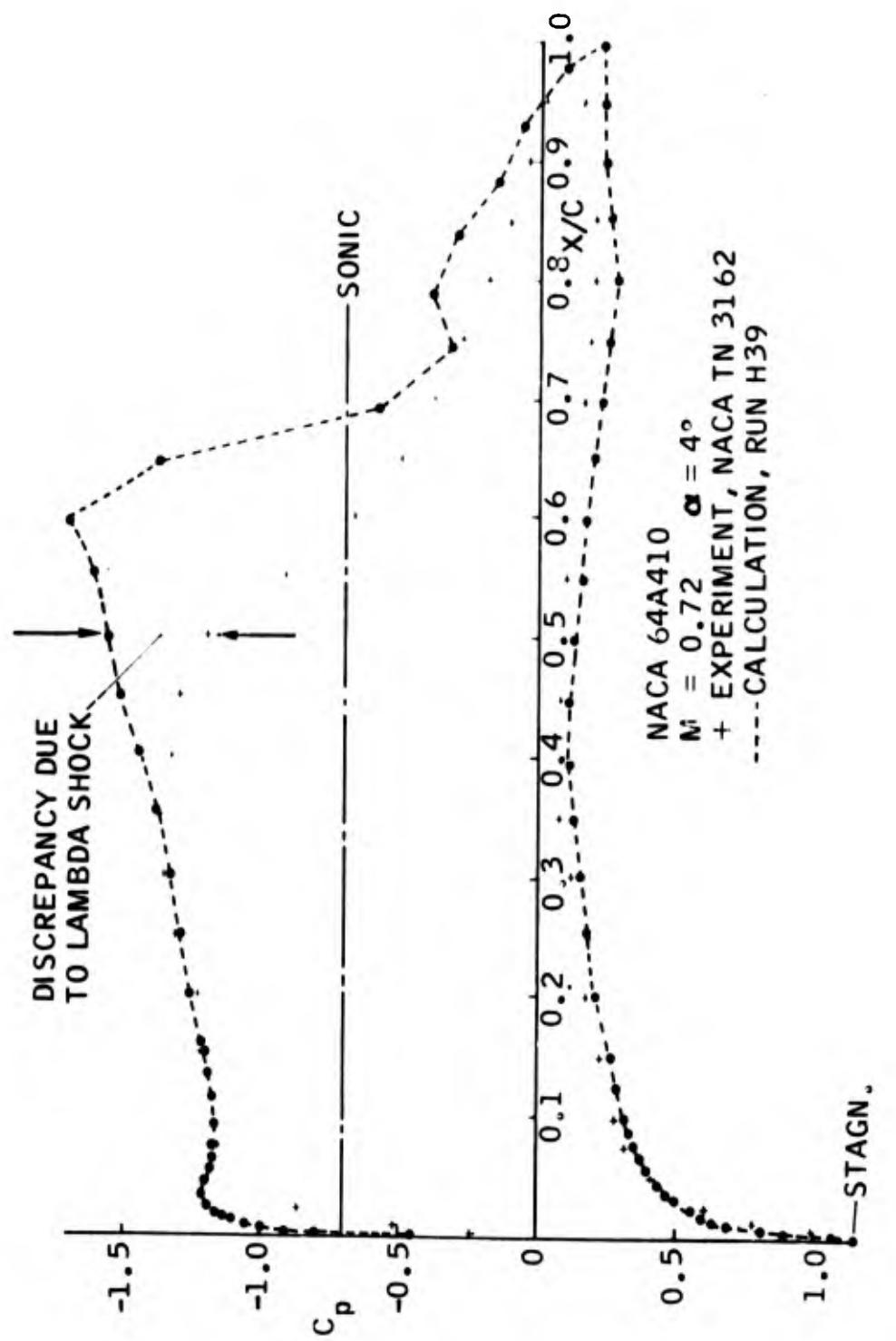


Figure 5. Pressure distribution on NACA 64A410 Airfoil at $M = 0.72$, $\alpha = 4^\circ$.

BLANK PAGE

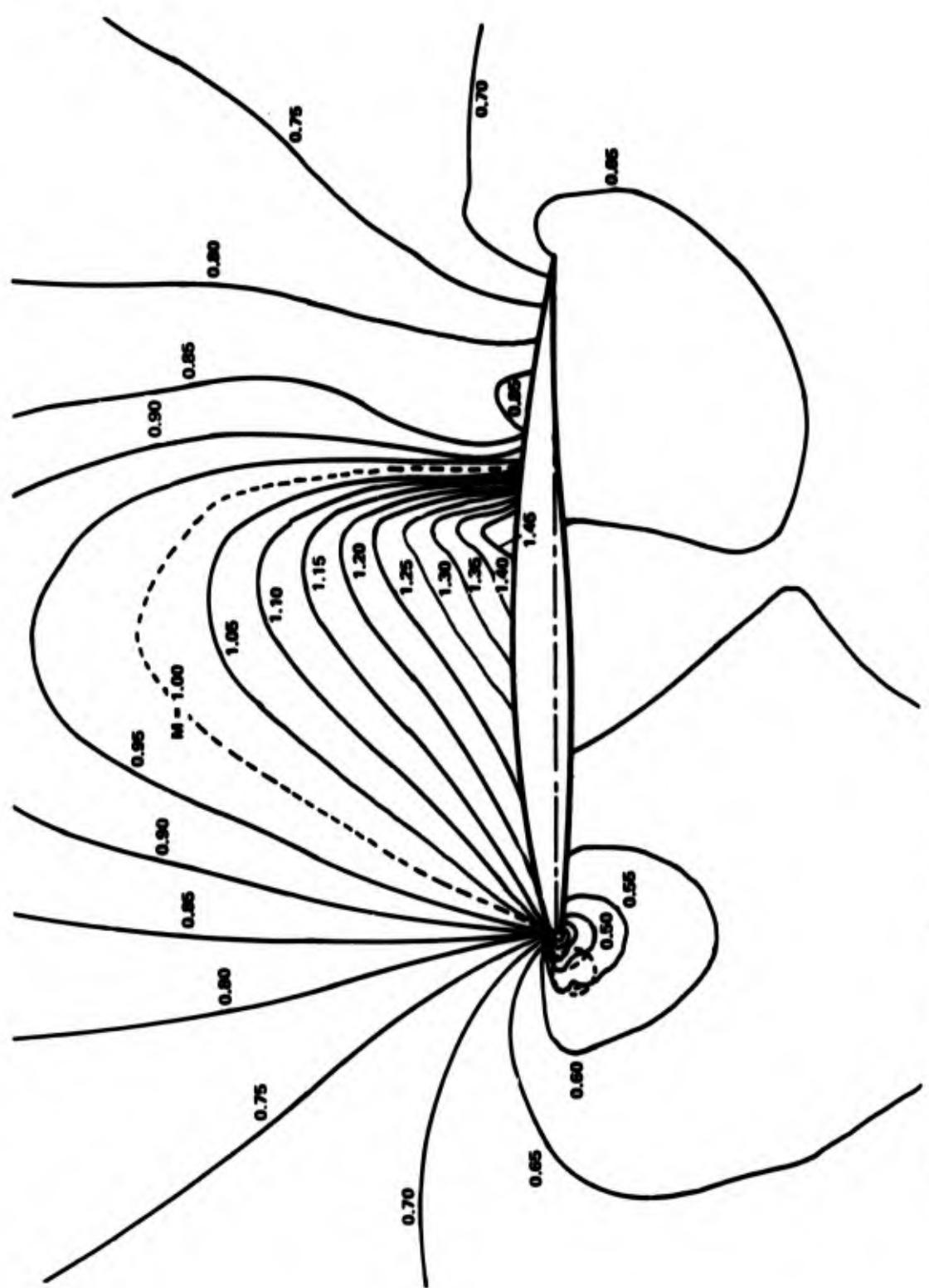
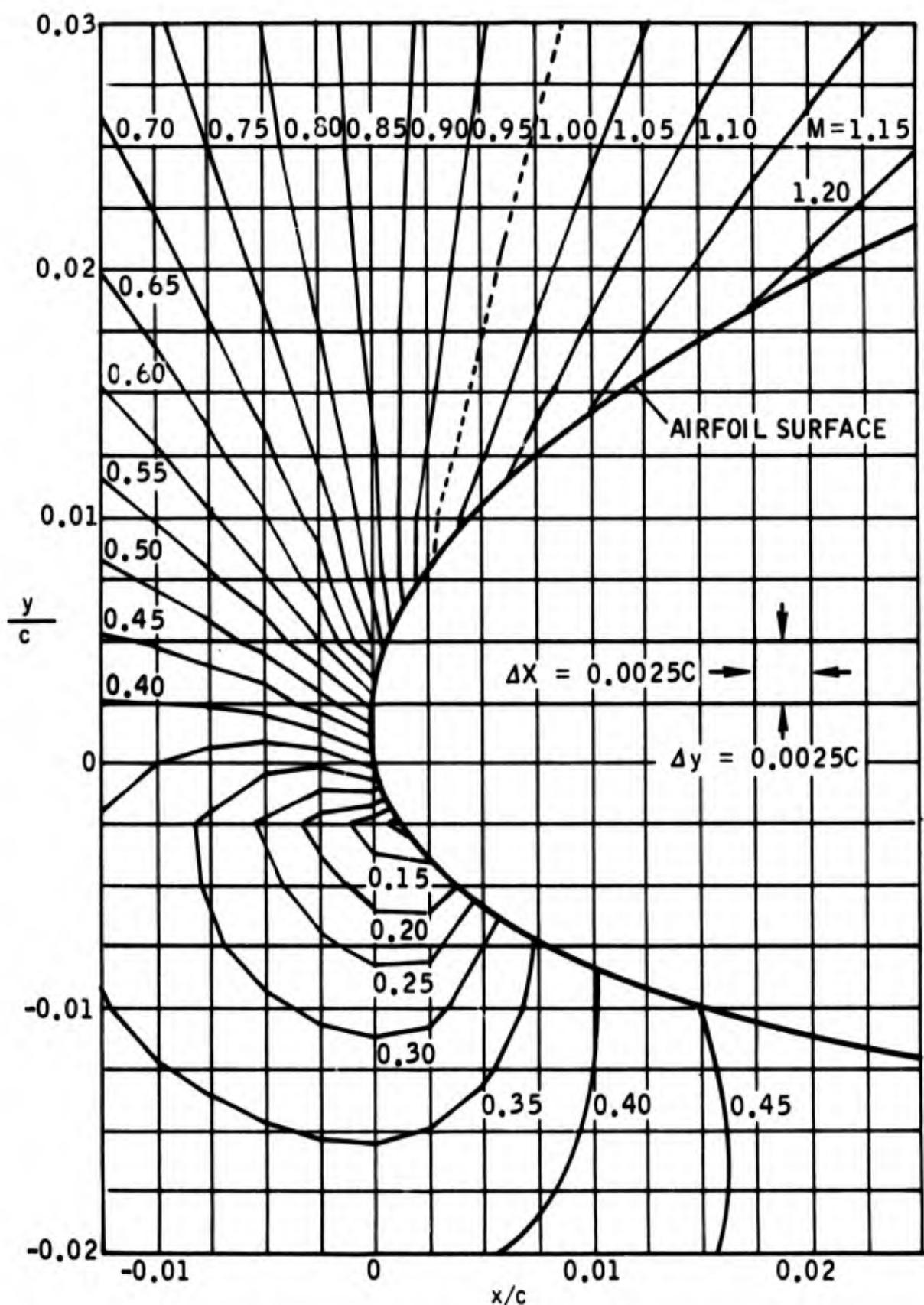


Figure 6. Mach number contours of the flowfield about a NACA 64A-410 Airfoil
at $M = 0.72$, $\alpha = 4^\circ$. (a) General view.



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 (W_b and WS_b) 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 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, DWTK,
AWT, CWT, IWT, LBX, MTT)

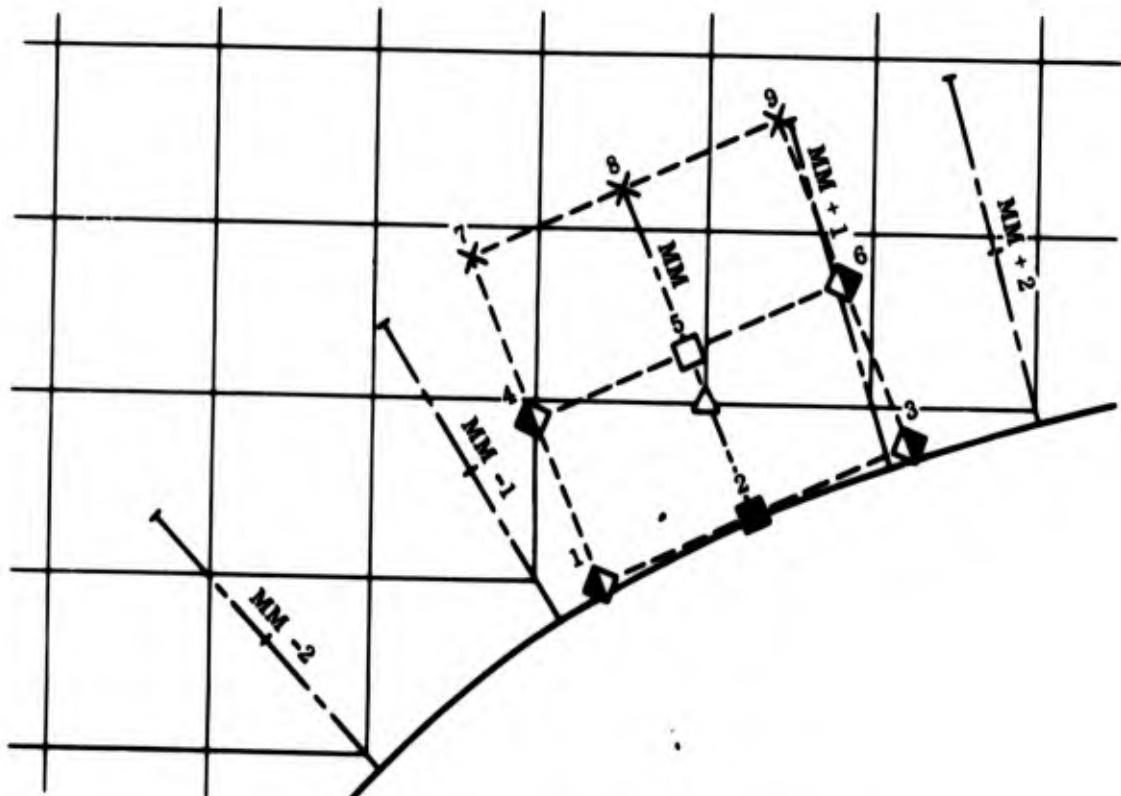
Subroutine PONDER directs the calculation of interpolation weights to be applied in transferring from the distorted mesh to adjacent local rotated mesh, also the weights for interpolating from the airfoil surface normals to those regular cartesian points adjacent to the airfoil surface and weights for 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 DWTK 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 (DWTK) 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 (DWTK) 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



- X** 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.

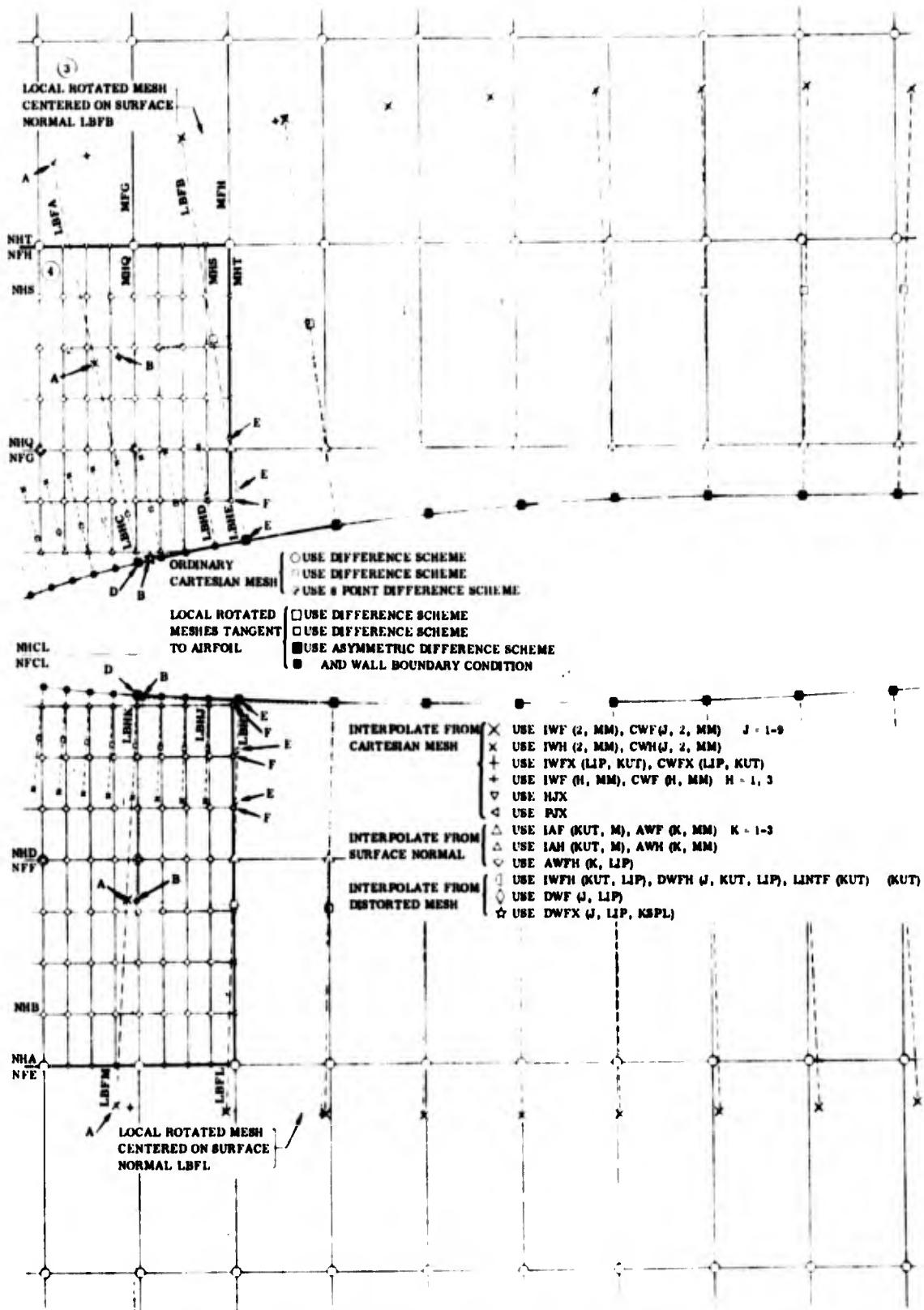


Figure 8. Special features of mesh interrelations where the airfoil penetrates a mesh size change.

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

<u>Card No.</u>	<u>Format</u>	<u>Cols.</u>	<u>Name</u>	<u>Definition</u>
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	3E1D.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 70,000₈.
- 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: $W_1 = \rho u$, $W_2 = \rho v$ and $W_3 = \rho$. 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

$\frac{\partial R_h}{\partial x}$, $\frac{\partial \theta}{\partial x}$, $\frac{\partial R_h}{\partial y}$ 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 $MCB \leq M \leq MCS$ and $NCB \leq N \leq NCS$. 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.

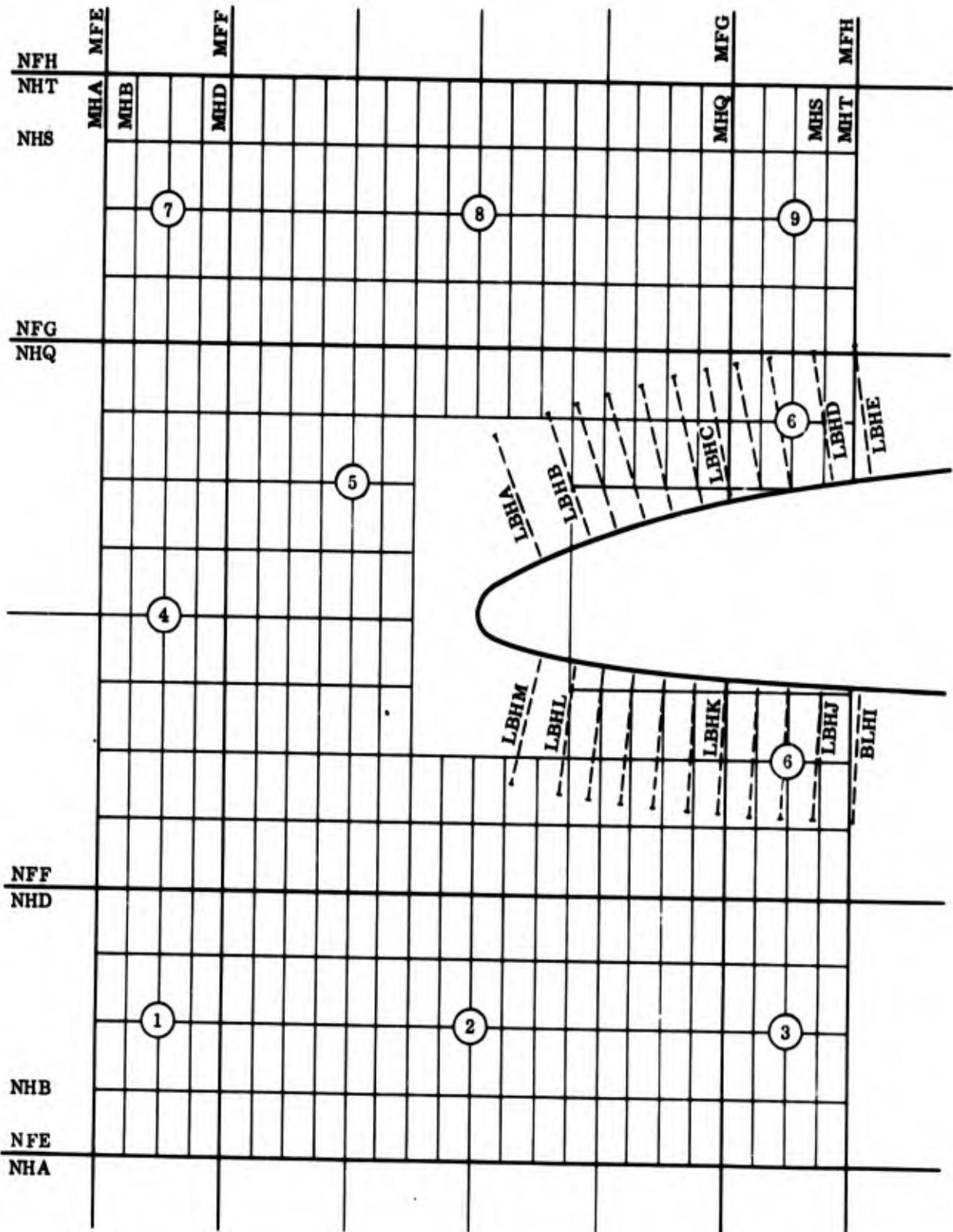
Near the left and right boundaries of the field the routine described is suitably modified to handle the problem of getting started and to assure that all the advanced data gets stored properly when the field has been completely covered.

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:

<u>I</u>	<u>JCT</u>	<u>IPX</u>	<u>IPY</u>
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

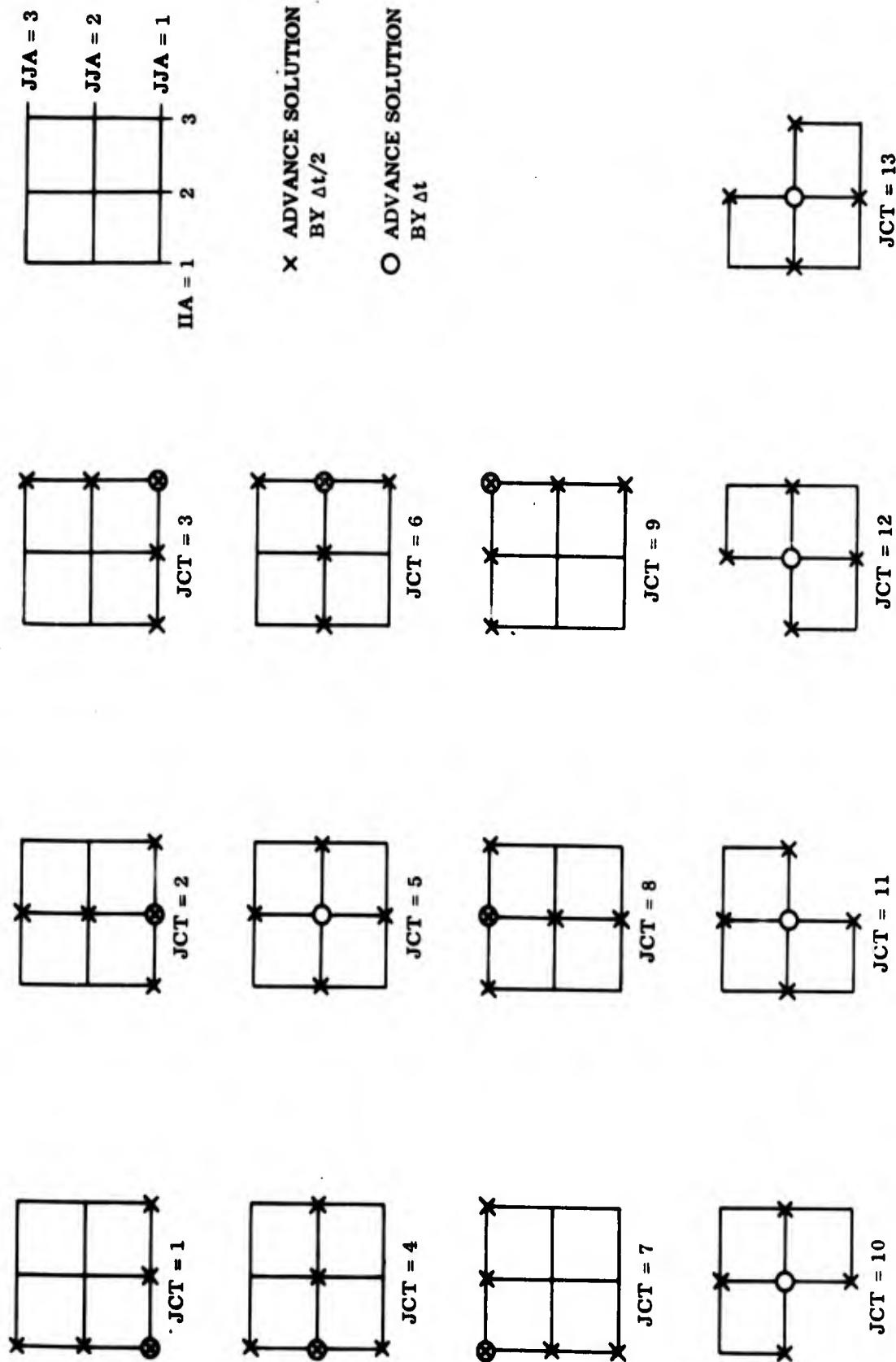


Figure 10. Cell types used in computer program.

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 $t_0 + \Delta t$ fall within the lozenge 2-6-8-4-2 in the t_0 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 number of 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 function 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 (ρ_u) 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 (ρ_v) at the cartesian mesh point representing the trailing edge has been set at zero.

1.15 SUBROUTINE KAKOI (WT,MTT,NTT,W^M,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 ρ_u , ρ_v , and ρ where u and v are cartesian mesh velocity components. In local rotated meshes the principal dependent variables are ρ_{u_t} , ρ_{u_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 $SCT(1,MM) = \sin \theta$ and $SCT(2,MM) = \cosine \theta$ 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, LEX, 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,LEX) and the orientation information is stored as SCT(2,LEX).

The interpolations from cartesian to local rotated meshes are carried out using locators IWT(3,LEX) and weights CWT(9,3,LEX). Interpolations from distorted meshes to local rotated meshes are carried out using weights DWT(9,4,LEX) 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

<u>Card No.</u>	<u>Format</u>	<u>Cols.</u>	<u>Variable</u>	<u>Definition</u>
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

<u>Card No.</u>	<u>Format</u>	<u>Cols.</u>	<u>Variable</u>	<u>Description</u>
				= 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 TEND
				= 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

<u>Card No.</u>	<u>Format</u>	<u>Cols.</u>	<u>Variable</u>	<u>Description</u>
		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 65500₈ 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 inputted 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, DELL)

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.

<u>Card</u>	<u>Variable</u>	<u>Format</u>	<u>Description</u>
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.

<u>Card</u>	<u>Variable</u>	<u>Format</u>	<u>Description</u>
	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.

<u>Card</u>	<u>Variable</u>	<u>Format</u>	<u>Description</u>
	DEL30	E10.3	Starting mass flow thickness, ft.
7	ERROR	20X, E20.3	Fractional 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 ICOMP =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, $^{\circ}$ 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.
----	---------	-------	--------------------------------

<u>Card</u>	<u>Variable</u>	<u>Format</u>	<u>Description</u>
	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, $^{\circ}$ 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:

<u>Region</u>	<u>Identifier</u>	<u>Description</u>
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.

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

<u>Identifier</u>	<u>Description</u>
<u>c</u>	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
<u>d</u>	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)
<u>e</u>	All cartesian meshes, C, F, H, M, T
<u>f</u>	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 , $\sqrt{1 + (\Delta Y / \Delta X)^2}$
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.
Ie0(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, MeT).

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.
<u>J</u> e0(J,M)	Operational parameter defining how the computing process changes at the points defined by <u>I</u> e0(J,M) (J=1,8) (M=1, MeT). Information is packed into <u>J</u> e0 as follows
	$J_e^0 = 100 n_1 + n_2$
	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.
<u>M</u> ac	Column identifier (e.g MTT is the last column in mesh T).
<u>M</u> b AN	Mesh column passing through airfoil nose (t,e.)
<u>M</u> bANP	= <u>M</u> bAN + 1
<u>M</u> bAT	Mesh column passing through airfoil tail (t.e)

<u>Mb</u> ATP	= <u>Mb</u> AT + 1
<u>Mb</u> ATM	= <u>Mb</u> AT - 1
MSL,MST	Column identifiers bounding the portion of the inner nose mesh region where the airfoil surface slope is between -l and +l.
Nac	Row identifier
<u>Nb</u> CL	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.

<u>XYb</u> (I,L)	x and y coordinates of points on the airfoil surface at the base of the surface normals for mesh <u>b</u> 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 upper 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 as 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)

$IW_{b(I,L)}$ 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 $IW_{b(I,L)}$, i.e.

$$IW_b = \underbrace{xx}_{\substack{\text{Col.} \\ \text{No.}}} \quad \underbrace{xx}_{\substack{\text{Row} \\ \text{No.}}}$$

$CW_b(9,I,L)$ Interpolation weights for the 9 point array in the cartesian mesh centered about the point indicated by $IW_b(I,L)$.

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.

$IW_{fX}(I,J)$ 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 IW_{fX}

$$IW_{fX} = \underbrace{xx}_{\substack{\text{Col.} \\ \text{No.}}} \quad \underbrace{xx}_{\substack{\text{Row} \\ \text{No.}}}$$

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 ($L=1 \text{ } LB_b \text{ MAX}$)

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.

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

$$L = 1 - LB_b \text{ MAX}$$

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).

$AWHM(3,3)$	Interpolation weights applied to the 3 values along a
$AWMT(3,3)$	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).

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

$$I = 1 \text{ upper surface}$$

$$= 2 \text{ lower surface}$$

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) Interpolation weights for points identified by IWHM()
DWMT(9,I,j) and IWMT(). I and J have same meanings.

APPENDIX II

SETUP PROGRAM FORTRAN LISTING

PROGRAM TEHAL (INPUT=OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE4, TAPE3) 000001
 C THIS VERSION FOR CAMBERED LIFTING AIRFOILS, OPERATIONAL LIMITS TO BE000002
 C UMAX = MACH NUMBER 000003
 C GAM = RATIO OF SPECIFIC HEATS 000004
 C ALPHA = ANGLE OF ATTACK, DEGREES 000005
 COMMON/LIMITS/ MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT, 000006
 C MCA, MCH, MCE, MCF, MCG, MCAL, MCS, MCI, NCAL, NCH, NCE, NCF, 000007
 C NCG, NCH, NCS, NCAL, 000008
 E MFA, MFD, MFF, MFH, MFQ, MFS, MFT, 'FAN, 000009
 E MFNP, MEAT, MEAJM, MFATP, MFA, MFD, MFE, MFF, MFG, MFT, 000010
 F NFU, NFS, NFT, NFCL, 000011
 H MH1, MH2, MH3, MH4, MH5, MH6, MH7, MH8, MH9, MH10, MH11, MH12, 000012
 H NH1, NH2, NH3, NH4, NH5, NH6, NH7, NH8, NH9, NH10, NH11, NH12, 000013
 M MM1, MM2, MM3, MM4, MM5, MM6, MM7, MM8, MM9, MM10, MM11, MM12, 000014
 M NM1, NM2, NM3, NM4, NM5, NM6, NM7, NM8, NM9, NM10, NM11, NM12, 000015
 T MT1, MT2, MT3, MT4, MT5, MT6, MT7, MT8, MT9, MT10, MT11, MT12, 000016
 T LT1, LTS, LT2, LT3, 000017
 S MS1, MST, NSL, NST, 000018
 COMMON/PLIMIT/ LHEA, LHEB, LBEC, LBED, LBFE, LPFI, LRFJ, LREK, 000019
 F LHEL, LHEM, LHEMAX, 000020
 H LH1A, LH1B, LH1C, LH1D, LH1E, LH1F, LH1G, LH1H, LH1I, LH1J, LH1K, LH1L, LH1M, 000021
 H LHMAXA, 000022
 M LH1A, LH1B, LH1C, LH1D, LH1E, LH1F, LH1G, LH1H, LH1I, LH1J, LH1K, LH1L, LH1M, 000023
 M LHMAX, 000024
 T LH1A, LH1B, LH1C, LH1D, LH1E, LH1F, LH1G, LH1H, LH1I, LH1J, LH1K, LH1L, LH1M, 000025
 T LH1W, LH1X, LH1Y, LH1Z, LH1AA, LH1AB, LH1AC, LH1AD, LH1AE, LH1AF, LH1AG, LH1AH, LH1AI, LH1AJ, LH1AK, LH1AL, LH1AM, 000026
 COMMON/ZMESH/ DX(6), DY(6), RH(9), 000027
 COMMON/SEMF/ SY(42), CY(42), QY(2,42), DCB(2,4,17), 000028
 A XY(12,42), SCE(12,42), XYH(2,22), SCH(2,22), 000029
 B XY(12,14), SCN(12,14), XYTS(2,43), SCI(2,43), 000030
 COMMON/DPLM/ IC0(4,17), JC0(4,17), JF0(8,37), JF0(8,37), 000031
 A IH1(8,25), JH0(8,25), IM0(8,26), JM0(8,26), IT0(8,16), JT0(8,16), 000032
 COMMON/DURA/ L, 000033
 COMMON/TERSE/ IYH(2,42), IC0(2,4,17), INFH(10), IOHMS(10), 000034
 A IUR(110), INF(3,42), IWH(3,22), IWM(3,14), INT(3,43), 000035
 B LINF(2), LINH(2), LINM(2), INF(2,2), IWHX(2,2), IWMX(2,2), 000036
 C IAF(2,37), IAH(2,25), IAM(2,26), IAT(2,16), IAN(2), 000037
 D INF(2,4), IWM(2,4), IWT(2,4), 000038
 COMMON/DM052/ DxF(9,4,42), DWI(9,4,22), DXV(9,4,14), 000039
 A DN1(9,4,43), AWE(3,42), AWH(3,22), AWM(3,14), AHT(3,43), 000040
 B CW(19,3,42), CWH(3,3,22), CM(9,3,14), CT(9,3,43), 000041
 C DWF(9,2,6), DWHL(9,2,6), DAXM(9,2,6), DWTX(9,2,6), 000042
 D ANFH(3,31), AWI(3,31), AWI(3,31), 000043
 E CWFM(9,2,4), JWHM(9,2,4), DWNT(9,2,4), 000044
 F CWFX(9,2,2), CWHX(9,2,2), CWNX(9,2,2), 000045
 COMMON/GAS/ GAM, HUMU, UNO2, DOGMU, ORGMU, HINF, GMU, UNOGMU, 000046
 COMMON/FLOW/ UMAX, ALPHA, RHNDY, 000047
 COMMON/SPOT/MKB, 000048
 COMMON/A16E/YUS, YLS, YPSU, YPSL, XNOSE, YPNOSE, RADCU, RADCL, RADG, 000049
 COMMON LY(3,42,9), XC(3,17,17), WF(3,37,13), WH(3,25,17), 000050
 A WM(3,26,26), WT(3,10,21), WF(3,9,42), WS(3,9,22), WSM(3,9,14), 000051
 B WS(13,9,43), WSUL(3), 000052
 DIMENSION DAF(2,37), CAF(2,37), SAF(2,37), 000053
 A DAU(2,25), CAU(2,25), SAU(2,25), 000054
 B DAU(2,26), CAU(2,26), SAU(2,26), 000055

G DAT(2,16), CAT(2,16), SAT(2,16) + 000056
 0 UAN(211), CAN(211), SAN(211) 000057
 E 022131, CSF(42), DSH(22), DSM(141), DST(63) 000058
 C NEW PROBLEM 000059
 16U PI = 3.1415926536 000060
 UTHET = PI/20.0 000061
 URHO = 0.125 000062
 UX(1) = URHO 000063
 UX(2) = 0.2 000064
 UX(3) = 0.05 000065
 UX(4) = 0.0125 000066
 UX(5) = 0.0025 000067
 UX(6) = 0.0025 000068
 LY(11) = UTHET 000069
 LY(12) = 0.2 000070
 LY(13) = 0.1 000071
 LY(14) = 0.025 000072
 LY(15) = 0.005 000073
 LY(16) = 0.0025 000074
 C QUICK POLAR COORDINATE FIELD 000075
 MYA = 1 SWYH = 2 NYS = 41 NYI = 42 000076
 LYA = 1 SWYU = 2 NYS = 8 NYI = 9 000077
 C COARSE CARTESIAN MESH 000078
 MCA = 1 SWCH = 2 MCE = 6 SMCE = 7 SMCG = 14 SMCH = 15 000079
 MCS = 16 SMCT = 17 000080
 LCA = 1 SWCH = 2 NCE = 6 SNCF = 7 SNCG = 11 SNCH = 12 000081
 NCA = 16 SNCT = 17 000082
 C EINE CARTESIAN MESH 000083
 MFA = 1 SMFH = 2 MFD = 5 SMFF = 6 SMFG = 7 SMFG = 11 000084
 MFM = 12 SWFH = 33 MFS = 36 SMEJ = 37 SMEAN = 9 SMEANP = 10 000085
 MFAT = 24 SWAJM = 28 SMEAJP = 30 000086
 MFA = 1 SWFH = 2 NFU = 3 NFE = 4 SNFF = 5 SNFG = 7 000087
 UFM = 8 SWFQ = 11 NFS = 12 SNFT = 13 SNFL = 6 000088
 C OUTER NOSE REGION 000089
 MHA = 1 SMHD = 2 MHE = 5 SMHF = 11 SMHF = 12 SMHG = 15 000090
 MMU = 16 SWHU = 21 MHS = 24 MHT = 25 SMHN = 13 SMHNP = 14 000091
 MHA = 1 SWHU = 4 SNHD = 5 SMHE = 7 SMHF = 8 SMHG = 11 000092
 MMH = 12 SWHW = 13 NMS = 16 NMHT = 17 SNHCL = 9 000093
 C INTERMEDIATE NOSE REGION 000094
 MHA = 1 SMMD = 6 SMME = 6 SMMF = 7 SMMG = 20 000095
 MMH = 21 SWMD = 21 SMMS = 25 SMMT = 26 SMMAN = 11 SMMANP = 12 000096
 UYA = 1 SWMO = 2 SMNU = 6 NMME = 7 SMNF = 8 SMNG = 16 000097
 MMH = 17 SWMD = 61 SNMS = 25 NMNT = 26 SNMCL = 11 000098
 MTA = 1 SMTH = 2 SMTU = 2 SMTG = 15 SMTS = 15 SMTT = 16 000099
 NTAN = 6 SNTAMP = 7 000100
 WTA = 1 SNTD = 2 SNTQ = 19 SNTS = 20 SNTT = 21 000101
 NTCL = 9 000102
 BHINY = 1.4 000103
 THEIA = -2.0*DTHEI 000104
 KH(11) = 1.0 000105
 LQ_101_NY = 2.0UYI 000106
 101 KH(NY) = BHINY-11 = URHO 000107
 C COARSE TO YIRE INTERPOLATION CONSTANTS 000108
 UQ_136_L = 1.0YT 000109
 SHETA = THEIA + UTHET 000110

```

-----  

    SY(L) = SIN(JHETA)..... 000111  

    LY(L) = COS(JHETA)..... 000112  

    X = BNDY*CY(L)..... 000113  

    Y = BNDY*SY(L)..... 000114  

    J = 1..... 000115  

    &JRF = -1.6..... 000116  

    102 IF(X = XTR).106,112,104..... 000117  

    104 XTR = XTR + DX(2)..... 000118  

    J = J + 1..... 000119  

    GO TO 102..... 000120  

    106 XRF = (XTR - X)/DX(2)..... 000121  

    IF(XRF = 0.5).110,108,108..... 000122  

    108 LYH(1,L) = J = 1..... 000123  

    LYH(1,L) = 1.0 - XRF..... 000124  

    GO TO 114..... 000125  

    114 LYH(1,L) = J..... 000126  

    LYH(1,L) = -XRF..... 000127  

    GO TO 116..... 000128  

    112 LYH(1,L) = J..... 000129  

    LYH(1,L) = 0.0..... 000130  

    116 J = 1..... 000131  

    YJM = -1.6..... 000132  

    122 IF(Y = YTR).126,132,124..... 000133  

    124 YJM = YTR + DY(2)..... 000134  

    J = J + 1..... 000135  

    GO TO 122..... 000136  

    126 YRF = YTR - Y/DY(2)..... 000137  

    IF(YRF = 0.5).130,128,128..... 000138  

    128 LYH(2,L) = J = 1..... 000139  

    LYH(2,L) = 1.0 - YRF..... 000140  

    GO TO 136..... 000141  

    130 LYH(2,L) = J..... 000142  

    LYH(2,L) = -YRF..... 000143  

    GO TO 136..... 000144  

    132 LYH(2,L) = J..... 000145  

    LYH(2,L) = 0.0..... 000146  

    136 CONTINUE..... 000147  

C---- YTR TO COARSE INTERPOLATION CONSTANTS..... 000148  

C---- LEFT COLUMN..... 000149  

    B = 1..... 000150  

    XSU = -1.6..... 000151  

    YSU = -1.6 - DY(2)..... 000152  

    LJP = NCI..... 000153  

    DXP = 0.0..... 000154  

    DYP = DY(2)..... 000155  

C---- COLUMN AND ROW ROUTINE..... 000156  

    140 DO 190 L = 1,LTP..... 000157  

    XSU = XSU + DXP..... 000158  

    YSU = YSU + DYP..... 000159  

    RMA = RPNTY/SQRT(XSU**2 + YSU**2)..... 000160  

    IF(XSU).142,144,142..... 000161  

    142 JHETA = ATAN2(YSU,XSU)..... 000162  

    IF(JHETA.LT.0.0) THETA = THETA + 2.0*PI..... 000163  

    GO TO 150..... 000164  

    144 IF(YSU).146,148,148..... 000165

```

140	THETA = 1.5+PI	000166
GO TO 150		000167
148	THETA = 0.5+PI	000168
C----N LOCATION		000169
150	J = 1	000170
ITR = -DTHT		000171
152	IF(LUEJA = JIRI, 156, 162, 154)	000172
154	ITR = ITK + DTHT	000173
J = J + 1		000174
GO TO 152		000175
156	TRF = (ITR - THETA)/DTHT	000176
IF(TRF = 0.5) 160, 158, 158		000177
158	ICB(1,K,L) = J = 1	000178
UCB(1,K,L) = 1.0 - TRF		000179
GO TO 170		000180
160	ICB(1,K,L) = J	000181
UCB(1,K,L) = -TRF		000182
GO TO 170		000183
162	ICB(1,K,L) = J	000184
UCB(1,K,L) = 0.0		000185
C----N LOCATION		000186
170	J = 1	000187
RTR = 1.0		000188
172	IF(RHA = RTR, 174, 182, 176)	000189
174	RTR = RTH = DRHO	000190
J = J + 1		000191
GO TO 172		000192
176	RKF = JRHF - RTR/DRHO	000193
IF(RKF = 0.5) 180, 180, 178		000194
178	ICB(2,K,L) = J = 1	000195
UCB(2,K,L) = 1.0 - RKF		000196
GO TO 190		000197
180	ICB(2,K,L) = J	000198
UCB(2,K,L) = -RKF		000199
GO TO 190		000200
182	ICB(2,K,L) = J	000201
UCB(2,K,L) = 0.0		000202
190	CONTINUE	000203
GO TO 192, 194, 196, 200, K		000204
C----RIGHT COLUMN		000205
192	K = 2	000206
ASU = 1.0		000207
YSU = -1.6 - DX(2)		000208
GO TO 140		000209
C----BOTTOM ROW		000210
194	K = 3	000211
ASU = -1.6 - DX(2)		000212
YSU = -1.6		000213
LTR = MCT		000214
DXP = DX(2)		000215
DYP = 0.0		000216
GO TO 140		000217
C----TOP ROW		000218
196	K = 4	000219
YSU = 1.6		000220

```

...XSY = -1.6 - DX(2).....000221
...GO TO 140.....000222
C----PRINT INTERPOLATION CONSTANTS.....000223
200 WRITE(6,23).....000224
...WRITE(6,25).....000225
...DO 204 MY = 1,MYI.....000226
204 WRITE(6,10) MY,IIYB(L,MY),L=1,2),(DYP(L,MY),L=1,2).....000227
...WRITE(6,27).....000228
...WRITE(6,31).....000229
...DO 206 NC = 1,NCT.....000230
206 WRITE(6,10) NC,(ICH(L,1,NC),L=1,2),(DCB(L,1,NC),L=1,2).....000231
...WRITE(6,28).....000232
...WRITE(6,31).....000233
...DO 208 NC = 1,NCT.....000234
208 WRITE(6,10) NC,(ICH(L,2,NC),L=1,2),(DCB(L,2,NC),L=1,2).....000235
...WRITE(6,29).....000236
...WRITE(6,32).....000237
...DO 210 NC = 1,NCT.....000238
210 WRITE(6,10) NC,(ICH(L,3,NC),L=1,2),(DCB(L,3,NC),L=1,2).....000239
...WRITE(6,30).....000240
...WRITE(6,32).....000241
...DO 212 NC = 1,NCT.....000242
212 WRITE(6,10) NC,(ICH(L,4,NC),L=1,2),(DCB(L,4,NC),L=1,2).....000243
C----AJKFOIL DESCRIPTION.....000244
250 CALL SHAPE(1,0,0).....000245
...WRITE(6,33).....000246
...WRITE(6,43).....000247
...WRITE(6,34).....000248
C----FINE CARTESIAN MESH.....000249
XAIR = 0.0 - DX(3).....000250
...DO 270 M = MFA1,MFAT.....000251
XAIR = XAIR + DX(3).....000252
CALL SHAPE(2,XAIR).....000253
...WRITE(6,8) XAIR,YUS,YPSU,YLS,YPSL.....000254
YSS = YLS.....000255
YPSA = YPSL.....000256
KAT = 1.....000257
N = NFF.....000258
YTEST = -CYSS.....000259
252 IF(YSS = YTEST) 256,258,254.....000260
254 JI = JI + 1.....000261
...YTEST = YTEST + DY(3).....000262
...GO TO 252.....000263
256 IAF(KAT,M) = N - 1.....000264
...IAF(KAT,M) = 1.0 - (YTEST - YSS)/DY(3).....000265
...DO 262.....000266
258 IAF(KAT,M) = N - 1.....000267
...IAF(KAT,M) = 1.0.....000268
262 CAF(KAT,M) = 1.0/SQR((1.0 + YSS)**2).....000269
...CAF(KAT,M) = YSS+CAF(KAT,M).....000270
...IF(KAT,EQ 2) GO TO 270.....000271
...YSS = YUS.....000272
...YPSA = YPSU.....000273
...YTEST = 0.0.....000274
...N = NFF.....000275

```

```

KAT = 2 ..... 000276
264 IF(YSS - YTEST) 266,267,269 ..... 000277
265 K = N + 1 ..... 000278
..... YTEST = YTEST + DY(3) ..... 000279
..... GO TO 264 ..... 000280
266 IAF(KAT,MI) = N ..... 000281
..... UAF(KAT,MI) = (YTEST - YSS)/DY(3) ..... 000282
..... GO TO 262 ..... 000283
267 IAF(KAT,MI) = N + 1 ..... 000284
..... UAF(KAT,MI) = 1.0 ..... 000285
..... GO TO 262 ..... 000286
270 CONTINUE ..... 000287
C----OUTER NOSE REGION ..... 000288
2701 KAT(6,51) ..... 000289
..... XAT(6,44) ..... 000290
..... XA1K = 0.0 - DX(4) ..... 000291
..... UV 2720 M = MHAN,MHT ..... 000292
..... XA1K = XA1K + UX(4) ..... 000293
..... CALL SHAPE(2,XA1K) ..... 000294
..... XAT(6,51) XA1K, YUS, YPSU, YLS, YPSL ..... 000295
..... YSS = YLS ..... 000296
..... YPSU = YPSL ..... 000297
..... KAT = 1 ..... 000298
..... K = NHCL - 2 ..... 000299
..... YTEST = -2.0DY(4) ..... 000300
2702 IF(YSS - YTEST) 2706,2708,2704 ..... 000301
2704 K = N + 1 ..... 000302
..... YTEST = YTEST + DY(4) ..... 000303
..... GO TO 2702 ..... 000304
2706 IAM(KAT,MI) = N = 1 ..... 000305
..... IAM(KAT,MI) = 1.0 -(YTEST - YSS)/DY(4) ..... 000306
..... GO TO 2712 ..... 000307
2708 IAM(KAT,MI) = N = 1 ..... 000308
..... IAM(KAT,MI) = 1.0 ..... 000309
2712 CAM(KAT,MI) = 1.0/SQRT(1.0 + YPSU**2) ..... 000310
..... SAM(KAT,MI) = YPSU*CAM(KAT,MI) ..... 000311
..... IF(KAT,EM,2) GO TO 2720 ..... 000312
..... YSS = YUS ..... 000313
..... YPSU = YPSU ..... 000314
..... YTEST = Q,U ..... 000315
..... K = NHCL ..... 000316
..... KAT = 2 ..... 000317
2714 IF(YSS - YTEST) 2716,2717,2715 ..... 000318
2715 K = N + 1 ..... 000319
..... YTEST = YTEST + DY(4) ..... 000320
..... GO TO 2714 ..... 000321
2716 IAM(KAT,MI) = N ..... 000322
..... IAM(KAT,MI) = (YTEST - YSS)/DY(4) ..... 000323
..... GO TO 2712 ..... 000324
2717 IAM(KAT,MI) = N + 1 ..... 000325
..... IAM(KAT,MI) = 1.0 ..... 000326
..... GO TO 2712 ..... 000327
2720 CONTINUE ..... 000328
C----INTERMEDIATE NOSE REGION ..... 000329
2801 KAT(6,52) ..... 000330

```

```

      WRITE(6,34)
      XAIR = 0.0 -DX(5)                                000331
      DO 2020 M = MMAN,MMT                            000332
      XAIR = XAIR + UX(5)                               000333
      CALL SHAPE(2,XAIR)                             000334
      WRITE (6,4) XAIR, YUS, YPSU, YLS, YPSL       000335
      YSS = YLS                                     000336
      YPSS = YPSL                                  000337
      KAT = 1                                     000338
      N = NMCL - 5                                000339
      YTEST = -5.0*DY(5)                            000340
      2002 IF(YSS - YTEST) 2806,2808,2804          000342
      2004 N = N + 1                                000343
      YTEST = YTEST + DY(5)                          000344
      GO TO 2802                                    000345
      2006 IAM(KAT,M) = N - 1                      000346
      UAM(KAT,M) = 1.0 -(YTEST - YSS)/DY(5)        000347
      GO TO 2812                                    000348
      2008 IAM(KAT,M) = N - 1                      000349
      UAM(KAT,M) = 1.0                            000350
      2012 UAM(KAT,M) = 1.0/SQR(1.0 + YPSS**2)    000351
      SAM(KAT,M) = YPSS*CAM(KAT,M)                 000352
      IF(KAT,EQ,2) GO TO 2820                      000353
      YSS = YUS                                     000354
      YPSS = YPSU                                 000355
      YTEST = 0.0                                  000356
      N = NMCL                                  000357
      KAT = 2                                     000358
      2014 IF(YSS - YTEST) 2816,2817,2815          000359
      2015 N = N + 1                                000360
      YTEST = YTEST + DY(5)                          000361
      GO TO 2814                                    000362
      2016 IAM(KAT,M) = N                      000363
      UAM(KAT,M) = (YTEST - YSS)/DY(5)            000364
      GO TO 2812                                    000365
      2017 IAM(KAT,M) = N + 1                      000366
      UAM(KAT,M) = 1.0                            000367
      GO TO 2812                                    000368
      2020 CONTINUE                                000369
      C----INNER NOSE REGION                      000370
      2401 WRITE(6,44)                                000371
      WRITE(6,34)
      XAIR = 0.0 -DX(6)                                000372
      DO 2920 M = MTAN,MTT                            000373
      XAIR = XAIR + UX(6)                               000374
      CALL SHAPE(2,XAIR)                             000375
      WRITE (6,6) XAIR, YUS, YPSU, YLS, YPSL       000376
      YSS = YLS                                     000377
      YPSS = YPSL                                  000378
      KAT = 1                                     000379
      N = NJCL - 8                                000380
      YTEST = -8.0*DY(6)                            000381
      2902 IF(YSS - YTEST) 2906,2908,2904          000382
      2904 N = N + 1                                000383
      YTEST = YTEST + DY(6)                          000384
                                              000385

```

60 TO 2942	000386
2906 IAI(KAT,M) = N - 1	000387
DAT(KAT,M) = 1.0 - (YTEST - YSS)/DY(6)	000388
GO TO 2912	000389
2908 IAI(KAT,M) = N	000390
DAT(KAT,M) = 0.0	000391
2912 CAT(KAT,M) = 1.0/SQM(11.0 + YSS**2)	000392
SAT(KAT,M) = YSS*CAT(KAT,M)	000393
IF(KAT.EQ.2) GO TO 2920	000394
YSS = YUS	000395
YSS = YPSU	000396
YTEST = U,U	000397
U = NTCL	000398
KAT = 2	000399
2914 IF(YSS - YTEST) 2916,2917,2915	000400
2915 N = N + 1	000401
YTEST = YTEST + DY(6)	000402
GO TO 2914	000403
2916 IAI(KAT,M) = N	000404
DAT(KAT,M) = (YTEST - YSS)/DY(6)	000405
GO TO 2912	000406
2917 IAI(KAT,M) = N	000407
DAT(KAT,M) = 0.0	000408
GO TO 2912	000409
2920 CONTINUE	000410
C----LOSE REGION HORIZONTAL CUTS	000411
WRITE(6,45)	000412
WRITE(6,36)	000413
KAT = 1	000414
NSL = IAI(1,M1)	000415
NST = IAI(2,M1)	000416
YAIK = FLOAT(NSL - 1 - NTCL)*DY(6)	000417
301 DO 314 NL = NSL,NST	000418
YAIK = YAIK + DY(6)	000419
CALL SHAPF(13,YAIK)	000420
WRITE(6,41,YAIK,XNOSE,YPNOSE)	000421
IF(N.EQ.NIGL) GO TO 309	000422
XSS = XNOSE	000423
M = MTAN - 9	000424
XTEST = -4.0*DX(6)	000425
302 IF(XSS - XTEST) 306,308,304	000426
304 M = M + 1	000427
XTEST = XTEST + DX(6)	000428
GO TO 302	000429
306 IAN(1) = M - 1	000430
DAN(1) = 1.0 - (XTEST - XSS)/DX(6)	000431
GO TO 312	000432
308 IAN(N) = M	000433
DAN(N) = 0.0	000434
GO TO 312	000435
309 IAN(N) = MTAN	000436
DAN(N) = 0.0	000437
312 DAN(N) = 1.0/SQRT(1.0 + YPNOSE**2)	000438
SAN(N) = YPNOSE+CAN(N)	000439
314 CONTINUE	000440

C----AIRFOIL INTERPOLATION OUTPUT.....000441
 515 WRITE(6,40).....000442
 WRITE(6,43).....000443
 WRITE(6,46).....000444
 DO 516 M = MFA,MFT.....000445
 516 WRITE(6,10) M, (IAF(K,M), K = 1:2), (DAF(K,M), K = 1:2).....000446
 WRITE(6,51).....000447
 WRITE(6,46).....000448
 DO 522 M = MHA,MHT.....000449
 522 WRITE(6,10) M, (IAH(K,M), K = 1:2), (DAH(K,M), K = 1:2).....000450
 WRITE(6,52).....000451
 WRITE(6,46).....000452
 DO 518 M = MMA,MMT.....000453
 518 WRITE(6,10) M, (IAM(K,M), K = 1:2), (DAM(K,M), K = 1:2).....000454
 WRITE(6,49).....000455
 WRITE(6,49).....000456
 DO 524 M = MTA,MTT.....000457
 524 WRITE(6,10) M, (IAT(K,M), K = 1:2), (DAT(K,M), K = 1:2).....000458
 WRITE(6,45).....000459
 WRITE(6,47).....000460
 DO 520 N = NTA,NTT.....000461
 520 WRITE(6,10) N, JAN(N), IAN(N), DAN(N).....000462
 C----SET CARTESIAN MESH OPERATIONAL LIMITS.....000463
 C----COARSE MESH, FIELD 2.....000464
 MLL = MCG.....000465
 MUL = MCF.....000466
 KL = 1.....000467
 321 DO 322 L = MLL,MUL.....000468
 JCU(1,M) = 2.....000469
 JCU(1,M) = 5.....000470
 JCU(2,M) = 17.....000471
 322 JCU(2,M) = 0.....000472
 GO TO 324.....000473
 324 BL = 2.....000474
 MLL = MCH.....000475
 MUL = MCS.....000476
 GO TO 320.....000477
 326 DO 328 M = MCF,MCG.....000478
 JCU(1,M) = 2.....000479
 JCU(1,M) = 5.....000480
 JCU(2,M) = 7.....000481
 JCU(2,M) = 0.....000482
 JCU(3,M) = 12.....000483
 JCU(3,M) = 5.....000484
 JCU(4,M) = 17.....000485
 328 JCU(4,M) = 0.....000486
 C----FINE MSH, FIELD 3. ASSUMES AIRFOIL BETWEEN Y=0.1, -0.3.....000487
 DO 330 M = MFR,MFS.....000488
 JFU(1,M) = 2.....000489
 330 JFU(2,M) = 5.....000490
 MLL = MFB.....000491
 MUL = MFQ = 1.....000492
 DO 332 M = MLL,MUL.....000493
 JFU(1,M) = 105.....000494
 JFU(2,M) = 405.....000495

IFU(3,M) = 12	000496
IFU(4,M) = 13	000497
JFU(3,ML) = 705	000498
332 JFU(4,M) = 700	000499
MLL = MFQ + 1	000500
MUL = MFS	000501
DO 334 M = MLL,MUL	000502
JFU(1,M) = 305	000503
JFU(2,M) = 605	000504
IFU(3,M) = 12	000505
IFU(4,M) = 13	000506
JFU(3,M) = 905	000507
IFU(3,M) = 12	000508
IFU(4,M) = 13	000509
334 JFU(4,M) = 900	000510
BL = 1	000511
MLL = MFD	000512
MUL = MFE	000513
330 DO 338 M = MLL,MUL	000514
JFU(1,M) = 205	000515
JFU(2,ML) = 505	000516
JFU(3,M) = 12	000517
JFU(3,M) = 805	000518
IFU(4,M) = 13	000519
338 JFU(4,M) = 800	000520
DO 10 (340,342),KL	000521
340 NL = 2	000522
MLL = MFATH	000523
MUL = MFQ	000524
DO 10 336	000525
342 DO 344 M = MFF,MEAT	000526
JFU(1,M) = 205	000527
JFU(2,M) = 505	000528
JFU(3,M) = 5	000529
JFU(3,M) = 500	000530
IFU(4,M) = 8	000531
JFU(4,M) = 505	000532
IFU(5,M) = 12	000533
JFU(5,M) = 805	000534
IFU(6,M) = 13	000535
344 JFU(6,M) = 800	000536
C----OUTER NOSE REGION, FIELD 4.	
C. ASSUMES UPPER SURFACE SLOPE POSITIVE, LOWER SLOPE NEGATIVE	
DO 350 M = MHB,MHS	000537
JHU(1,M) = 2	000538
350 JHU(2,M) = 5	000539
MLL = MHB	000540
MUL = MHD = 1	000541
DO 352 M = MLL,MUL	000542
JHU(1,M) = 105	000543
JHU(2,ML) = 405	000544
JHU(3,M) = 14	000545
JHU(3,M) = 705	000546
JHU(4,M) = 17	000547
352 JHU(4,M) = 700	000548
	000549
	000550

DO 354 M = MMQ, MHS	000551
JHQ(1,M) = 205	000552
354 HQ(2,ML) = 505	000553
DO 356 M = MMQ, MHE	000554
IHQ(3,M) = 14	000555
JHQ(3,M) = 805	000556
IHQ(4,M) = 17	000557
356 HQ(4,M) = 800	000558
DO 358 M = MMF, MMG	000559
IHQ(3,M) = 8	000560
JHQ(3,M) = 500	000561
IHQ(4,M) = 12	000562
JHQ(4,M) = 505	000563
IHQ(5,M) = 14	000564
JHQ(5,M) = 805	000565
IHQ(6,M) = 17	000566
358 HQ(6,M) = 800	000567
DO 364 M = MMH, MHS	000568
J = 3	000569
IF (IAH(1,N), EQ, IAH(1,N+1)) GO TO 360	000570
JHQ(J,ML) = IAH(1,ML-1)	000571
JHQ(J,M) = 513	000572
J = 4	000573
360 HQ(1,J,M) = IAH(1,ML)	000574
JHQ(1,ML) = 500	000575
J = J + 1	000576
IHQ(J,M) = IAH(2,M) + 1	000577
IF (IAH(2,M), EQ, IAH(2,M+1)) GO TO 362	000578
JHQ(J,M) = 511	000579
J = J + 1	000580
IHQ(J,M) = JHQ(J-1,M) + 1	000581
362 HQ(1,J,M) = 505	000582
J = J + 1	000583
IHQ(J,M) = 14	000584
JHQ(J,M) = 805	000585
J = J + 1	000586
IHQ(J,M) = 17	000587
364 HQ(1,J,M) = 800	000588
MUL = MMQ + 1	000589
MUL = MHS	000590
DO 366 M = MLL, MUL	000591
DO 366 J = 1,8	000592
366 HQ(1,J,M) = HQ(1,J,M) + 100	000593
C-----INTERMEDIATE NOSE REGION, FIELD 5	000594
C-----ASSUMES UPPER SURFACE SLOPE POSITIVE, LESS THAN 63 DEGREES,	000595
C LOWER SLOPE NEGATIVE	000596
DO 370 M = MMB, MMS	000597
IHQ(1,M) = 2	000598
JHQ(1,M) = 105	000599
IHQ(2,M) = 6	000600
370 HQ(2,M) = 405	000601
DO 372 M = MMB, MME	000602
IHQ(3,M) = 22	000603
JHQ(3,M) = 705	000604
IHQ(4,M) = 26	000605

372	JMO(4,M) = 700	000606
DU	J74 M = MMD,MMS	000607
DU	J74 J = 1,2	000608
374	JMO(J,M) = JMO(J,M) + 100	000609
DU	J76 M = MMF,MMG	000610
DU	J76 J = 3,4	000611
376	JMO(J,M) = JMO(J,M) + 100	000612
DU	J78 M = MMF,MMG	000613
	IM0(3,M) = 8	000614
	IM0(3,M) = 500	000615
	IM0(4,M) = 17	000616
	IM0(4,M) = 505	000617
	IM0(5,M) = 22	000618
	IM0(5,M) = 805	000619
	IM0(6,M) = 26	000620
378	JMO(6,M) = 800	000621
DU	J84 M = MMH,MMS	000622
	J = 3	000623
	IF(IAM(1,M),EQ,IAM(1,M)+1) GO TO 380	
	IM0(J,M) = IAM(1,M) - 1	000624
	JMO(J,M) = 513	000625
	J = 4	000626
380	IM0(J,M) = IAM(1,M)	000627
	JMO(J,M) = 500	000628
	J = J + 1	000629
	IM0(J,M) = IAM(2,M) + 1	000630
	IF(IAM(2,M),EQ,IAM(2,M)+1) GO TO 382	
	IM0(J,M) = 511	000632
	J = J + 1	000633
	IM0(J,M) = IM0(J-1,M) + 1	000634
382	JMO(J,M) = 505	000635
	J = J + 1	000636
	IM0(J,M) = 22	000637
	IM0(J,M) = 805	000638
	J = J + 1	000639
	IM0(J,M) = 26	000640
384	JMO(J,M) = 800	000641
	MLL = MMQ + 1	000642
	MUL = MMS	000643
	DU J86 M = MLL,MUL	000644
	DU J86 J = 1,8	000645
386	JMO(J,M) = JMO(J,M) + 100	000646
C----	INNER NOSE REGION, FIELD 6	000647
	MFEN = IAN(NSL)	000648
	DU J90 N = NSL,NST	000649
	IF(IAM(NL) = MFEN) 386,390,394	000650
388	MFEN = IAN(N)	000651
	MFEN = N	000652
390	CONTINUE	000653
	DU J92 M = MTB,MTS	000654
	ITQ(1,M) = 2	000655
	ITQ(2,M) = 3	000656
	JIV(1,M) = 205	000657
392	JIV(2,M) = 505	000658
	MLL = MTQ	000659
		000660

MUL = MFFN - 1	000661
DQ 394 M = MLL, MUL	000662
IJU(3,M) = 20	000663
JTU(3,M) = 805	000664
IJU(4,M) = 21	000665
394 JTU(4,M) = 800	000666
C----URAGING PART OF NOSE? ASSUMES FORWARD CHORD OVERHANG LESS THAN DX(6)	000667
C NOSE CANTER ZERO OR POSITIVE	000668
M = MFFN	000669
IJU(3,M) = NFFN - 1	000670
JTU(3,M) = 513	000671
IJU(4,M) = NFFN	000672
JTU(4,M) = 500	000673
N = NFFN	000674
396 IF(IAT(N+1) = IAT(N)) 398,398,400	000675
394 N = N + 1	000676
GO TO 396	000677
400 IJU(5,M) = N + 1	000678
401 JTU(5,M) = 511	000679
IJU(6,M) = N + 2	000680
JTU(6,M) = 505	000681
IJU(7,M) = 20	000682
JTU(7,M) = 805	000683
IJU(8,M) = 21	000684
JTU(8,M) = 800	000685
IF(N,EQ,WTAN) GO TO 420	000686
M = WTAN	000687
N = NSL	000688
402 IF(IAT(M) = M) 406,406,404	000689
404 N = N + 1	000690
GO TO 402	000691
406 ITO(3,M) = N - 1	000692
IJO(4,M) = N	000693
JTU(3,M) = 513	000694
JTU(4,M) = 500	000695
N = NTCL	000696
408 IF(IAT(N+1) = N) 410,410,400	000697
410 N = N + 1	000698
GO TO 408	000699
C----MAIN PART OF INNER NOSE REGION	000700
420 DU 430 M = WTANP,NTIS	000701
J = 3	000702
IF(IAT(1,M),EQ,IAT(1,M+1)) GO TO 422	000703
IJO(J,M) = IAT(1,M+1)	000704
JTU(J,M) = 513	000705
J = 4	000706
IJO(J,M) = ITO(J-1,M) + 1	000707
GO TO 424	000708
422 IJU(J,M) = IAT(1,M)	000709
424 JIJO(J,M) = 500	000710
J = J + 1	000711
IF(IAT(2,M),EQ,IAT(2,M+1)) GO TO 426	000712
IJO(J,M) = IAT(2,M+1)	000713
JTU(J,M) = 511	000714
J = J + 1	000715

170(J,M) = JTQ(J-1,M1 + 1) 000716
 GO TO 420 000717
 426 JTQ(J,M1) = IAT12,M1 + 1 000718
 428 JTQ(J,M) = 505 000719
 J = J + 1 000720
 JTQ(J,M) = 20 000721
 JTQ(J,M) = 805 000722
 J = J + 1 000723
 JTQ(J,M) = 21 000724
 430 JTQ(J,M) = 800 000725
 C---- WRITE OPERATIONAL LIMITS 000726
 *WRITE(6,25) 000727
 *WRITE(6,20) 000728
 DO 562 M = MCA,MCT 000729
 562 *WRITE(6,12) M, ((JCQ(K,M),JLQ(K,M)), K = 1,4) 000730
 *WRITE(6,39) 000731
 *WRITE(6,42) 000732
 DO 568 M = MFA,MFT 000733
 568 *WRITE(6,12) M, ((JFQ(K,M),JFO(K,M)), K = 1, 8) 000734
 *WRITE(6,51) 000735
 *WRITE(6,62) 000736
 DO 572 M = MHA,MHT 000737
 572 *WRITE(6,12) M, ((IHQ(K,M),JHO(K,M)), K = 1, 8) 000738
 *WRITE(6,52) 000739
 *WRITE(6,41) 000740
 DO 584 M = MMA,MHT 000741
 584 *WRITE(6,12) M, ((IMQ(K,M),JMQ(K,M)), K = 1, 8) 000742
 *WRITE(6,38) 000743
 *WRITE(6,61) 000744
 DO 576 M = MTA,MJT 000745
 576 *WRITE(6,12) M, ((JTQ(K,M),JLQ(K,M)), K = 1, 8) 000746
 C---- INTERPOLATION INSTRUCTIONS AT FIELD BOUNDARIES 000747
 C---- CODE 1: IMPRT.. CODE 2: 4 POINT SPREAD SYMMETRIC CODES, 3 POINT SPREAD 000748
 C---- HEAVY BELOW.. CODE 4: 3 POINT SPREAD HEAVY ABOVE 000749
 C---- FINE MESH IN OUTER NOSE REGION 000750
 CALL BUNJE ((NEE,NFH,IFU(1,MFH),JFQ(1,MFH)),IDFH) 000751
 CALL BUNJE ((NUU,NFH,JHU(1,MFH),JHO(1,MFH)),IDHM) 000752
 CALL BUNJE ((NUU,NFH,IMU(1,MFH),JM0(1,MFH)),IDMT) 000753
 *WRITE(6,51) 000754
 *WRITE(6,49) 000755
 *WRITE(6,2) ((DEH(j)), J = 1,10) 000756
 *WRITE(6,52) 000757
 *WRITE(6,43) 000758
 *WRITE(6,2) ((DHM(j)), J = 1,10) 000759
 *WRITE(6,35) 000760
 *WRITE(6,64) 000761
 *WRITE(6,2) ((DMT(j)), J = 1,10) 000762
 C---- CONSTRUCTION OF CENTRAL AXES FOR LOCAL ROTATED BOUNDARY MESHES 000763
 C---- LOCATION OF 45 DEG. POINTS IN INNER NOSE REGION 000764
 KAT = 1 000765
 M = MTJ 000766
 B02 IF(SAT(KAT,M) + D,707) B06,B06,B04 000767
 B04 M = M - 1 000768
 GO TO 802 000769
 B06 MSL = M 000770

```

NSL = IAT(KAT,MSL) ..... 000771
KAT = 2 ..... 000772
M = MJ ..... 000773
A08 JF(SATSKAT,M) = 0.7071 810,812,812 ..... 000774
BLU M = M - 1 ..... 000775
DO TO 808 ..... 000776
A12 MST = M ..... 000777
NST = IAT(KAT,MST) ..... 000778
WRITE(6,50) ..... 000779
WRITE(6,21) NSL,MST,NST,NST ..... 000780
C----SURFACE NORMALS, UPPER SURFACE ..... 000781
WRITE(6,45) ..... 000782
WRITE(6,45) ..... 000783
WRITE(6,66) ..... 000784
DO M20 M = MST,MTS ..... 000785
MM = 11 - NIAN + 1 ..... 000786
XA = FLOAT(M - MTAN)*DX(6) ..... 000787
ATEST = XA ..... 000788
L = IAT(2,M) ..... 000789
YA = FLOAT(N - NC1)*DY(6) ..... 000790
XEMM = -UY(2,0)*DY(6)*SAT(2,M)*CAT(2,M) ..... 000791
A14 ATEST = XTGST - XEMM ..... 000792
CALL SHAPE(5,XTEST) ..... 000793
XEMM = (XTGST-XA-(YA-YUS)*YPSU/(1.0 + YPSU**2)) ..... 000794
IF(AUS(XEMM)) = 0.01*UY(6) 816,816,814 ..... 000795
B16 AYI(1,MM) = XTEST ..... 000796
AYI(2,MM) = YUS ..... 000797
SCJ(2,MM) = 1.0/SQRT(1.0 + YPSU**2) ..... 000798
SCT(1,MM) = YPSU*SCT(2,MM) ..... 000799
UST(MM) = SQRT((XA-XTEST)**2 + (YA-YUS)**2)/DY(6) ..... 000800
B20 WRITE(6,24) MM,MM,XYT(1,MM),XYT(2,MM),SCT(1,MM),SCT(2,MM),DST(MM) ..... 000801
C----SURFACE NORMALS, LOWER SURFACE ..... 000802
WRITE(6,46) ..... 000803
DO M30 M = NSL,MTS ..... 000804
MM = 2*MIT - M + 2 - MTAN ..... 000805
XA = FLOAT(M - MTAN)*DX(6) ..... 000806
ATEST = XA ..... 000807
N = IAT(1,M) ..... 000808
YA = FLOAT(N - NC1)*DY(6) ..... 000809
XEMM = DAT(1,ML)*DY(6)*SAT(1,M)*CAT(1,M) ..... 000810
B24 XTEST = XTEST - XEMM ..... 000811
CALL SHAPE(4,XTEST) ..... 000812
XEMM = (XTEST - XA + SYLS-YA)*YPSL/(1.0 + YPSL**2) ..... 000813
IF(AUS(XEMM)) = 0.01*UY(6) 825,826,824 ..... 000814
B26 AYI(1,MM) = XTEST ..... 000815
AYI(2,MM) = YLS ..... 000816
SCJ(2,MM) = -1.0/SQRT(1.0 + YPSL**2) ..... 000817
SCJ(1,MM) = YPSL*SCT(2,MM) ..... 000818
UST(MM) = SQRT((XA-XTEST)**2 + (YA-YLS)**2)/DY(6) ..... 000819
B30 WRITE(6,24) MM,MM,XYI(1,MM),XYI(2,MM),SCT(1,MM),SCT(2,MM),DST(MM) ..... 000820
C----SURFACE NORMALS, BLUNT PART OF NOSE ..... 000821
WRITE(6,66) ..... 000822
DO M40 M = NSL,NST ..... 000823
MM = N + 2*(MIT - MTAN + 1) ..... 000824
YA = FLOAT(N - NC1)*UY(6) ..... 000825

```

```

----- YTEST = YA ..... 000826
----- M = IAN(N) ..... 000827
----- XA = FLOATIM = MTAN1+DX(6) ..... 000828
----- YERK = DANIN)*UX161+CAI(N)*SAN(N) ..... 000829
----- B34 YTST = YTEST - YERK ..... 000830
----- CALL SHAPE(3,YTEST) ..... 000831
----- YERK = (YTEST-YA+(XNOSE-XA)/YPNOSE)+YPNOSE**2/(1.0+YPNOSE**2) ..... 000832
----- IF (WHS(YERK)) = 0.01*UY(6) B36,B36,B34 ..... 000833
----- B36 AT(1,MM1) = XNOSE ..... 000834
----- AT(2,MM1) = YTEST ..... 000835
----- CSF = 1.0/SQRT(1.0 + YPNOSE**2) ..... 000836
----- SCT(2,MM1) = SIGN(CSF,YPNOSE) ..... 000837
----- SC(1,MM1) = ABS(YPNOSE*SC(2,MM)) ..... 000838
----- DST(MM) = SQRT((YA - YTST)**2 + (XA - XNOSE)**2)/UY(6) ..... 000839
----- B40 WRITE(6,14),MM,MN,XYIS1,MM1,AT(2,MM),SCT(1,MM),SCT(2,MM1),DST(MM) ..... 000840
----- C---- INTERMEDIATE NOSE REGION ..... 000841
----- WRITE(6,52) ..... 000842
----- CALL CAXIS15,MMH,MHS,MHN,MHNL,1-MHG,2*(MMJ+1)-MMG,IAM,DAM,SAM, ..... 000843
----- A.CAP,EMT,XYI4,SCM,DSM,LBMMAX) ..... 000844
----- C---- OUTER NOSE REGION 4 ..... 000845
----- WRITE(6,51) ..... 000846
----- CALL CAXIS14,MMH,MHS,MHN,MHNL,1-MHG,2*(MHT+1)-MMG,IAH,DAH,S0H, ..... 000847
----- A.CAP,EMT,XYI1,SC1,DSH,LBMMAX) ..... 000848
----- C---- FINE MESH REGION 3 ..... 000849
----- WRITE(6,43) ..... 000850
----- CALL CAXIS13,MEH,MEA1,MEA1,NECL,1-MEG,2*(MEAT+3)-MEG,IAF,DAE,SAF, ..... 000851
----- A.CAP,EMT,XYF,SCE,DSF,LBFMAX) ..... 000852
----- C---- TRAILING EDGE SPECIAL AUXILIARIES ..... 000853
----- M = NFAT ..... 000854
----- U0,B82,K = 1,2 ..... 000855
----- M = M + 1 ..... 000856
----- AA = FLOATIM = MFAN1+DX(3) ..... 000857
----- MM = M + MEG + 1 ..... 000858
----- AT(1,MM) = XA ..... 000859
----- AT(2,MM) = 0.0 ..... 000860
----- USF(1,MM) = U,0 ..... 000861
----- SCF(1,MM) = 0,0 ..... 000862
----- SCF(2,MM) = 1,0 ..... 000863
----- BM = 2*(IMEAT+3) - MEG - M ..... 000864
----- AT(1,MM) = XA ..... 000865
----- AT(2,MM) = 0,0 ..... 000866
----- USF(MM) = U,0 ..... 000867
----- SCF(1,MM) = 0,0 ..... 000868
----- B82 SCF(2,MM) = -1,0 ..... 000869
----- C---- OPERATING LIMIT NAMES, BOUNDARY POINTS ..... 000870
----- C---- INNER NOSE REGION ..... 000871
----- LC = 1 - MTAN ..... 000872
----- LBTA = MST + LC ..... 000873
----- LHTA = LBTA + 1 ..... 000874
----- LHTC = MTQ + LC ..... 000875
----- LHTU = MTS + LC ..... 000876
----- LHTL = MTL + LC ..... 000877
----- LC = 2*(MTT + 1) - MTAN ..... 000878
----- LHTL = LC - MTJ ..... 000879
----- LHTU = LC - MTS ..... 000880

```

LHTK = LC - MTQ	000881
LHTM = LC - MSL	000882
LHTL = LHTM - 1	000883
LC = 2*(MTI) - MTAN + 1)	000884
LHTK = NSL + LC	000885
LHTW = LHTM - 1	000886
LHTS = NST + LC	000887
LHTT = LHTS + 1	000888
LHMAX = 2*(MTI) - MTAN + 1) + NJI	000889
C----INTERMEDIATE NOSE REGION	
LC = 1 - MMG	000890
LBMA = MMG + LC	000891
LBMI = MMH + LC	000892
LBMC = MMC + LC	000893
LBMU = MMS + LC	000894
LBME = MMT + LC	000895
LC = 2*(MMI + 1) - MMG	000896
LBML = LC - MMT	000897
LBMJ = LC - MMS	000898
LBMK = LC - MMU	000899
LBML = LC - MMH	000900
LBMM = LC - MMG	000901
LBMAX = 2*(MMI - MMG + 1)	000902
C----OUTER NOSE REGION 4	
LC=1-MFG	000903
LBHA=MHG+LC	000904
LBHD=MHG+LC	000905
LBHJ=LC-MHG	000906
LBH=LC-MHG	000907
LBHMAX=LBHM	000908
C----FINE MESH REGION 3	
LC = 1-MFG	000909
LBFA=MFG+LC	000910
LBFD=LHFC+1	000911
LBFE=LHFC+2	000912
LC=2*(MFAT+3)-MFG	000913
LBFK=LHFK-1	000914
LBFI=LHFK-2	000915
LBFM=LC-MFG	000916
LBMAX=LBFM	000917
C----ASSIGNMENTS OF CENTRAL AXIS DATA AT OVERLAPS	
C----INNER REGION OVERLAPS AT MEDIUM PART OF NOSE	
KUT = 1	000918
LU = LBTT	000919
LK = LBTH	000920
b94. UO.892.K = 1,2	000921
XYT(K,LU) = XYT(K,LK)	000922
b92. SCI(K,LU) = SCI(K,LK)	000923
UST(LU) = UST(LK)	000924
b93. KUT = 2 LU=LUTA LK=LHTS GO TO b94	000925
b94. KUT = 3 LU=LHTA LK=LHTS GO TO b94	000926
LU = LBTH	000927
LK = LBTL	000928
b95. KUT = 4 S LU=LHTM S LK=LUTH S GO TO b90	000929
C----INITIAL - INTERMEDIATE OVERLAP	
b96. KUT = 1 LU = LBMA	000930
LK = LBTC	000931
b98. UO.900.K = 1,2 XYT(K,LU) = XYT(K,LK)	000932
	000933
	000934
	000935

900. SCM(K,LUL = SCT(18,LK)	000936
USM(LU) = DST(LK)+DY(6)/DY(5)	000937
60. TO. (902,904), KUT	000938
902. KUT = 2	000939
LU = LBMM	000940
LK = LBJK	000941
60. TO. 898	000942
904. KUT = 1	000943
LU = LHTE	000944
LK = LBMA	000945
906. DU 908. K = 1/2	000946
AYT(K,LUL = XYM(K,LK)	000947
908. SCT(K,LUL = SCM(K,LK)	000948
DST(LU) = USM(LK)+DY(5)/DY(6)	000949
60. TO. (910,912), KUT	000950
910. KUT = 2	000951
LU = LHTI	000952
LK = LBML	000953
60. TO. 906.	000954
C. INTERMEDIATE OUTER OVERLAP	000955
912. KUT=1. LU=LBHA. ILK=LBMC	000956
914. DU. 916. K=1/2. AYH(K,LUL)=XYM(K,LK)	000957
916. SCM(K,LUL=SCM(K,LK). USM(LU)=DSM(LK)+DY(5)/DY(4)	000958
60. TO. (918,920), KUT	000959
918. KUT=2. LU=LBUM. ILK=LBMK. 60. TO. 914	000960
920. KUT=1. LU=LHME. ILK=LBHM	000961
922. DU. 924. K=1/2. AYH(K,LUL)=XYM(K,LK)	000962
924. SCM(K,LUL=SCM(K,LK). USM(LU)=DSM(LK)+DY(4)/DY(5)	000963
60. TO. (926,928), KUT	000964
926. KUT=2. LU=LBMI. ILK=LBML. 60. TO. 922.	000965
C. OUTER FINE OVERLAP.	000966
928. KUT=1. LU=LBFA. ILK=LBMC	000967
930. DU. 932. K=1/2. AYF(K,LUL)=XYH(K,LK)	000968
932. SCT(K,LUL)=SCM(K,LK). USM(LU)=DSM(LK)+DY(5)/DY(3)	000969
60. TO. (934,936), KUT	000970
934. KUT=2. LU=LBFM. ILK=LBHK. 60. TO. 930	000971
936. KUT=1. LU=LBME. ILK=LBHF	000972
938. DU. 940. K=1/2. AYH(K,LUL)=XYF(K,LK)	000973
940. SCM(K,LUL=SCF(K,LK). USM(LU)=DSF(LK)+DY(3)/DY(4)	000974
60. TO. (942,944), KUT	000975
942. KUT=2. LU=LBUI. ILK=LBFL. 60. TO. 934	000976
C. ---A1MFVIL SURFACE POINTS.	000977
944. WLT(6,65).	000978
WLT(6,55)	000979
WLT(6,66).	000980
DU. 1170. MM= 1. LBIMAX.	000981
1170. WLT(6,14) MM,MM,XYT(1,MM),XYT(2,MM),SCT(1,MM),SCT(2,MM),DST(MM)	000982
C. ---INTERMEDIATE NOSE REGION.	000983
WLT(6,52).	000984
WLT(6,66).	000985
DU. 1172. MM=1. LBUMMAX.	000986
1172. WLT(6,14) MM,MM,XYM(1,MM),XYM(2,MM),SCV(1,MM),SCM(2,MM),DSM(MM)	000987
C. OUTER NOSE REGION 4	000988
WLT(6,51).	000989
WLT(6,66).	000990

DO 1174 MM=1,LBIMAX 000991
 1174 WRITE(6,14)MM,MM,XYH(1,MM),XYH(2,MM),SCH(1,MM),SCH(2,MM),DSH(MM) 000992
 C FINE MESH REGION 3 000993
 WRITE(6,43) 000994
 WRITE(6,66) 000995
 DO 1176 MM=3,LBIMAX 000996
 1176 WRITE(6,14)MM,MM,XYE13,MM1,XYE12,MM1,SCE(1,MM1),SCE(2,MM1),DSF(MM) 000997
 C---END OF PRINTOUT 000998
 C---INTERPOLATE NOSE REGION 000999
 KM = 6 001000
 KUT = 1 001001
 MLL = LP1B 001002
 MUL = LR1D 001003
 NSPL = 0 001004
 AJZ = DX(6)*FLOAT(NTA-NICL) 001005
 Y50 CALL PONRHS...KSP1=LML,MUL,ATZ,YIZ,YIT,SCJ,DST,UWT,DWIX,AWT 001006
 A CWT,IWT,LBTMAX,MTT) 001007
 DO 10 1952!Y54!95B!,KUJ. 001008
 Y52 KUT = 2 001009
 954 KUT = 3 001010
 Y50 KUT(6,67) 001011
 KUT(6,35) 001012
 KUT(6,08) 001013
 DO 1178 MM= 1,LBIMAX 001014
 DO 1178 LIP = 1,4 001015
 1178 WRITE(6,14)MM,LIP,(UWT(K,LIP,MM),K=1,9) 001016
 C---SPECIAL INTERPOLATION WEIGHTS AT END OF RANGE, (UWTX) 001017
 WRITE(6,90) 001018
 DO 1179 KUT = 1,6 001019
 DO 1179 LIP = 1,2 001020
 1179 WRITE(6,14)KUT,LIP,(UWTX(K,LIP,KUT),K=1,9) 001021
 C---INTERPOLATION WEIGHTS, SURFACE NORMALS TO CARTESIAN 001022
 WRITE(6,69) 001023
 WRITE(6,35) 001024
 MLL(6,70) 001025
 DO 1190 MM= 1,LBIMAX 001026
 IF(MM.LE.L41E) GO TO 1186 001027
 IF(MM.LE.L81M) GO TO 1188 001028
 N = MM - 2*(MTT - MTAN + 1) 001029
 N = IAN(N) 001030
 GO TO 1190 001031
 1180 M = MM + MTAN - 1 001032
 N = IAT(2,M) 001033
 GO TO 1190 001034
 1180 M = 2*MTT + 2 - MTAN - MM 001035
 N = IAT(1,M) 001036
 1190 WRITE(6,16)MM,1,N,(AN(I,K,MM),K=1,3) 001037
 C---INTERPOLATION WEIGHTS, CARTESIAN TO ROTATED MESH 001038
 WRITE(6,71) 001039
 WRITE(6,35) 001040
 WRITE(6,72) 001041
 DO 1204 MM= 1,LBIMAX 001042
 DO 1204 LIP = 1,3 001043
 1204 WRITE(6,16)MM,LIP,(WIS(LIP,MM),(CWT(K,LIP,MM),K=1,9)) 001044
 C---END OF PRINTOUT 001045

```

BK = 5 . KUT = 1 . MLL=LUMH . MUL=LRHO . SKSPL=0 . 001046
XTZ = DX(5)*FLOAT(MMA-MMAN) . SYTZ = DY(5)*FLOAT(NMA-NMCL) . 001047
960 CALL PONUERI . KSPL,MLL,MUL,XTZ,YTZ,XYM,SCM,DSM,UWM,UWDX,AWH. 001048
A LUM,LBMMAX,MMT1 . 001049
GO TO 1962,19681,KUT . 001050
962 CALL WAIF(KH,KUT,LBMA,XTZ,YTZ,XYM,SCM,CWMX,IWMX,LBMMAX) . 001051
KUL = 2 . MLL=LUMJ . MUL=LPML . GO TO 960 . 001052
964 CALL WAIF(KH,KUT,LUMM,XTZ,YTZ,XYM,SCM,CWMX,IWMX,LBMMAX) . 001053
C=====INTERMEDIATE NOSE REGION . 001054
+KLT(6,52) . 001055
-KLT(6,68) . 001056
DQ 1180 MM=3,LHMMAX . 001057
DQ 1180 LIP=1,4 . 001058
1180 KLT(6,14)MM,LIP,(D,M(K,LIP,MM),K=1,9) . 001059
+KLT(6,91) . 001060
DQ 1181 KUT = 1,6 . 001061
DQ 1181 LIP=1,2 . 001062
1181 KLT(6,14)KUT,LIP,(UWMX(K,LIP,KUT),K=1,9) . 001063
C=====INTERMEDIATE NOSE REGION . 001064
+KLT(6,52) . 001065
-KLT(6,70) . 001066
DQ 1194 MM=1,LHMMAX . 001067
LE(1MM,LE,LBME)GO TO 1192 . M=2+(MMT+1)-MMG-MM . 001068
LE = IAM(L1,MJ) . GU TO 1194 . 001069
1192 M = MUL = 1 + MUG . IN = IAM(2,MI) . 001070
1194 KLT(6,16)MM,MH,(IA,M(K,MM),K=1,3) . 001071
+KLT(6,52) . 001072
+KLT(6,72) . 001073
DQ 1206 MM=1,LHMMAX . 001074
DQ 1206 LIP=1,3 . 001075
1206 KLT(6,16)MM,LIP,IWMSLIP,MM),(CWM(K,LIP,MM),K=1,9) . 001076
+KLT(6,52) . 001077
+KLT(6,22) . 001078
DQ 1223 KUT = 1,2 . 001079
DQ 1223 LIP = 1,2 . 001080
1223 KLT(6,16)KUT,LIP,IWMX(LIP,KUT),(CWMX(K,LIP,KUT),K=1,9) . 001081
C=====END OF PRINTOUT . 001082
C=====OUTER NOSE REGION 4 . 001083
BK = 4 . SKUT=1 . MLL=LPHM . MUL=LRHO . SKSPL=0 . 001084
BTZ = DX(4)*FLOAT(MHA-MMAN) . SYTZ = DY(4)*FLOAT(NHA-NMCL) . 001085
970 CALL PONUERI . KSPL,MLL,MUL,XTZ,YTZ,XYM,SCM,DSM,UWM,UWDX,AWH. 001086
A LUM,LBMMAX,MMT1 . 001087
GO TO 1972,19781,BUT . 001088
972 CALL WAIF(KH,KUT,LBMA,XTZ,YTZ,XYM,SCM,CWMX,IWMX,LBMMAX) . 001089
KUL = 2 . MLL=LUMJ . MUL=LBML . GO TO 970 . 001090
974 CALL WAIF(KH,KUT,LUMM,XTZ,YTZ,XYM,SCM,CWMX,IWMX,LBMMAX) . 001091
C=====OUTER NOSE REGION 4 . 001092
+KLT(6,21) . 001093
+KLT(6,66) . 001094
DQ 1182 MM=1,LHMMAX . 001095
DQ 1182 LIP=1,4 . 001096
1182 KLT(6,14)MM,LIP,(UWM(K,LIP,MM),K=1,9) . 001097
+KLT(6,92) . 001098
DQ 1183 KUT = 1,6 . 001099
DQ 1183 LIP=1,2 . 001100

```

1183. WRITE(6,14)KUT,LIP,(WWMX(K,LIP,KUT),K=1,9) 001101
 C OUTCH NOSE REGION 4 001102
 WRIT(6,51). 001103
 WRIT(6,70). 001104
 DO 1198 MN=1,LHMAX 001105
 IF(MM,LF,LHME)GO TO 1196 M=2*(MHT+1)-MHG-MM 001106
 N = IAH(1,M) 560.TU 1198 001107
 1196 M = MM - 1 + MHG N = IAH(2,M) 001108
 1198 WRIT(6,16)MM,M,N,(AWH(K,MM),K=1,3) 001109
 C OUTCH NOSE REGION 4 001110
 WRIT(6,51) 001111
 WRIT(6,72). 001112
 DO 1208 MN=1,LHMAX 001113
 DO 1208 LIP=1,3 001114
 1208 WRIT(6,16)MM,LIP,IWHS(LIP,MM),(CWH(K,LIP,MM),K=1,9) 001115
 WRIT(6,51) 001116
 WRIT(6,24) 001117
 DO 1224 KUT=1,2 001118
 DO 1224 LIP=1,2 001119
 1224 WRITE(6,16)KUT,LIP,I,HX(LIP,KUT), (WWMX(K,LIP,KUT),K=1,9) 001120
 C-----LNU OF PHINOUT 001121
 C FINE MESH REGION 3 001122
 KM = 3 MLL=LHFH MUL=LHFD KSPL=0 001123
 ATZ = UX(3)*FLUAT(MFA-MFAU) YTZ = DY(3)*FLUAT(NFA-NFCU) 001124
 YB0,LAL,PONUETI KSPL,MLL,MUL,ATZ,YTZ,XYF,SCF,DSF,DWF,DWFX,AWF, 001125
 A CWF,IWF,LUFMAX,MFT) 001126
 DO 10 1982/9881,KUT 001127
 YB2 CALL WAJF(KM,KUT,LHFH,XTZ,YTZ,XYF,SCF,CWFX,IWFX,LUFMAX) 001128
 KUT = 2 MLL=LHFJ MUL=LHFJ GO TO 980 001129
 YB0 CALL WAJF(KM,KUT,LHFH,XTZ,YTZ,XYF,SCF,CWFX,IWFX,LUFMAX) 001130
 C FINE MESH REGION 3 001131
 WRITE(6,43) 001132
 WRIT(6,51) 001133
 DO 1184 MN=1,LHMAX 001134
 DO 1184 LIP=1,4 001135
 1184 WRIT(6,14)MN,LIP,(DWF(K,LIP,MM),K=1,9) 001136
 WRIT(6,51) 001137
 DO 1185 KUT = 1,6 001138
 DO 1185 LIP=1,2 001139
 1185 WRITE(6,14)KUT,LIP,(DWFX(K,LIP,KUT),K=1,9) 001140
 C FINE MESH REGION 3 001141
 WRIT(6,43) 001142
 WRIT(6,70). 001143
 DO 1202 MN=1,LHMAX 001144
 IF(MM,LF,LHFE)GO TO 1200 M=2*(MFAT+3)-MFG-MM 001145
 N = IAF(1,M) 560.TU 1202 001146
 1200 M = MM - 1 + MFG N = IAF(2,M) 001147
 1202 WRIT(6,16)MM,M,N,(AWF(K,MM),K=1,3) 001148
 C FINE MESH REGION 3 001149
 WRIT(6,43) 001150
 WRIT(6,72). 001151
 DO 1210 MN=1,LUFMAX 001152
 DO 1210 LIP=1,3 001153
 1210 WRIT(6,16)MM,LIP,(WF(LIP,MM),(CWF(K,LIP,MM),K=1,9) 001154
 WRIT(6,43). 001155

```

----- WRITE(6,26)
DO 1225 KUT=1,2
DO 1225 LIP=1,2
1225 WRITE(6,16)KUT,LIP,1,FX(LIP,KUT),(CWF(X,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
AWM(1,LIP) = 0.5*DAY*(DAY+1.0)
AWM(2,LIP) = 1.0 - DAY**2
1142 AWM(3,LIP) = AWM(1,LIP) - DAY
C----SPECIAL MESH MESH INTERPOLATIONS
WRITE(6,73)
----- WRITE(6,70)
DO 1212 LIP=1,3
1212 WRITE(6,16)LIP,LIP,LIP,(AWHM(K,LIP),K=1,3)
C----END OF PRINTOUT
DOY = DY(5)/DY(4) 3DAY = 1.0 + DOY
DO 1144 LIP=1,3 3DAY=DAY-DOY 3AWHM(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,(AWFM(K,LIP),K=1,3)
C----END OF PRINTOUT
DOY = DY(4)/DY(3) 3DAY = 1.0 + DOY
DO 1146 LIP=1,3 3DAY=DAY-DOY 3AWFH(1,LIP)=0.5*DAY*(DAY+1.0)
AWFH(2,LIP) = 1.0 - DAY**2
1146 AFJ(3,LIP) = AWFM(1,LIP) - DAY
WRITE(6,75)
----- WRITE(6,70)
DO 1216 LIP=1,3
1216 WRITE(6,14)LIP,LIP,(AWFM(K,LIP),K=1,3)
C----END OF PRINTOUT
C----ASSIGNMENTS TO FINER MESH CARTESIAN BOUNDARY POINTS
DO 1148 LIP=1,2
MP = 5
NP = 1
CALL DIST(1,0,0,0,0,0,XYM,SCM,LHMMAX)
LIP = IFIX(DY(5)/DY(6)) + 0,0,1) - 1
AP = DX(6)*FLOAT(JMTT) - MTAN)
YP = DY(5)*FLOAT(JAM(2,MHH)) - NMCL)
YPT = YP - DY(5)*DAM(2,MHH)
MP = NTJ
NP = JAM(2,MHH) - NMCL)*2 + NTCL
DO 1148 LIP = 1, LIP
YP = YP - DY(6)
NP = NP - 1
IWM(KUT,LIP) = 100*MP + NP
IF(LIP.GE.YPJ).LINTM(BUTL.E.LIP
1148 CALL DIST(2,XP,YP,0,MT(1,KUT,LIP),XYM,SCM,LHMMAX1) SKUT=2
-----
```

```

----- 001211
YP = DY(5)*FLOAT(IAM(1,MHH) - NMCL) 001212
NP = (IAM(1,MHH) - NMCL)*2 + NTCL 001213
----- 001214
YPI = YP + DY(5)*QAM(1,MHH) 001215
M = LBM1 001216
CALL DISTW(1,0,0,0,0,0,XYM,SCM,LBMMAX) 001217
----- 001218
UD 1150 LIP = 1,LOP 001219
----- 001220
YP = YP + UY(6) 001221
NP = NP + 1 001222
----- 001223
JML(1KUT,LIP) = 100*MP + NP 001224
----- 001225
JFL(YP,LE,YPT),LINHM(KUT) = LIP 001226
1150 CALL DISTW(2,XP,YP,U,NP(1,KUT,LIP),XYM,SCM,LBMMAX) 001227
----- 001228
WLT(16,76) 001229
----- 001230
UD 1218 KUT = 1,2 001231
UD 1218 LIP = 1,4 001232
----- 001233
1218 WLT(16,16),KUT=LIP,IWM(IKUT,LIP),SDWMTSK,KUT,LIP),K=1,9) 001234
----- 001235
WLT(16,44),LINHM(1),LINHM(2) 001236
M = BHM SKUT = 1 KR = 4 001237
----- 001238
CALL WISIW(1,0,0,0,0,0,XYH,SCM,LBHMAX) 001239
----- 001240
LIP = JF18(UY(4)/UY(5)+0,0U1)-1 *XP=DX(5)*FLOAT(MMT-MMAN) 001241
----- 001242
YP = DY(4)*FLOAT(IAM(2,MHH)-NFCL) ,YPT=YP-DY(4)*DAH(2,MHH) 001243
----- 001244
NP = MMT 001245
----- 001246
NP = IAM(2,MHH) - NMCL,1+9 + NMCL 001247
----- 001248
UD 1152 LIP=1,LOP ,YP=YP+UY(5) :IF(YP,GE,YPT)LINH(KUT)=LIP 001249
----- 001250
NP = NP + 1 001251
IWM(IKUT,LIP) = 100*MP + NP 001252
----- 001253
1152 CALL DISTW(2,XP,YP,U,NP(1,KUT,LIP),XYH,SCM,LBHMAX) 001254
----- 001255
WLT(16,77) 001256
----- 001257
UD 1220 KUT=1,2 001258
UD 1220 LIP=1,4 001259
----- 001260
1220 WLT(16,16),KUT=LIP,IWM(IKUT,LIP),SDWMTSK,KUT,LIP),K=1,9) 001261
----- 001262
WLT(16,44),LINHM(1),LINHM(2) 001263
M = LBFP SKUT = 1 KR = 3 001264
----- 001265
CALL DISTW(1,0,0,0,0,0,XYF,SCF,LBFMAX) 001266
----- 001267
JFL(YP,LE,YPT),LINH(KUT) = LIP 001268
NP = (IAF(1,MFH) - NFCL)*4 + NMCL 001269
----- 001270
NP = IAF(1,MFH) - NFCL 001271
----- 001272
CALL DISTW(1,0,0,0,0,0,XYF,SCF,LBFMAX) 001273
----- 001274
UD 1158 LIP=1,LOP ,YP=YP+UY(4) :IF(YP,LE,YPT)LINH(KUT)=LIP 001275
----- 001276
NP = NP + 1 001277
IWFH(IKUT,LIP) = 100*MP + NP 001278
----- 001279
1158 CALL DISTW(2,XP,YP,U,NP(1,KUT,LIP),XYF,SCF,LBFMAX) 001280
----- 001281

```

```

      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(6,46) LINTF(1),LINIF(<)
C-----WORKING VARIABLES
      DOU,UMAU(5,42),UMACH,GAM,ALPHA
      WRITE(6,48)
      WRITE(6,61),UMACH,GAM,ALPHMA
      ALPHAR = ALPHA/100.0+P1
      SU(1) = 1.0
      SU(1) = UMACH+COS(ALPHAR)
      SU(2) = UMACH+SIN(ALPHAR)
C----USEFUL CONSTANTS
      UGMU = 0.5*GAM-1.0
      UDVN1 = 1.0/(0.7*UMACH**2)
      UNIF = 1.0/(GAM - 1.0) + 0.5*UMACH**2
      UMU = GAM - 1.0
      UNUGMU = 1.0/GAM
      UNUG = 1.0/GAM
      DUGM = 1.0/HGMU
C----ASSIGN WORKING VARIABLES TO START NEW PROBLEM
C----OUTER POLAR COORDINATE FIELD
      DO 112 M = 1,NYT
      DO 112 N = 1,NYJ
      DO 112 K = 1,3
      112 W(K,M,N) = WSU(K)
C----CARTESIAN MESH
      DO 622 DO 622 M = 1,MCT
      DO 622 N = 1,NCT
      DO 622 K = 1,3
      622 W(K,M,N) = WSU(K)
C----FINE MESH
      DO 632 DO 632 M = 1,MFT
      DO 632 N = 1,NFT
      DO 632 K = 1,3
      632 W(K,M,N) = WSU(K)
C----OUTER NOSE REGION
      DO 642 DO 642 M = 1,MMT
      DO 642 N = 1,MTT
      DO 642 K = 1,3
      642 W(K,M,N) = WSU(K)
C----INTERMEDIATE NOSE REGION
      DO 652 DO 652 M = 1,MMT
      DO 652 N = 1,MTT
      DO 652 K = 1,3
      652 W(K,M,N) = WSU(K)
C----INNER NOSE REGION
      DO 662 DO 662 M = 1,MTT
      DO 662 N = 1,MTT
      DO 662 K = 1,3
      662 W(K,M,N) = WSU(K)
C----BOUNDARY PUSTAGE STAMP MESHES
      DO 669 J = 1,9
C----FINE MESH

```

DD 670 LL = 1,LBEMAX
 WSF(3,J,LL) = WSU(3) 001321
 WSF(1,J,LL) = WSU(1)+SCF(2,LL) + WSU(2)*SCF(1,LL) 001322
 WSF(2,J,LL) = WSU(1)+SCF(1,LL) + WSU(2)*SCF(2,LL) 001323
 IF((SCF(1,LL),EQ,0,0),AND,(SCF(2,LL),EQ,0,0))WSF(1,J,LL)=WSU(1) 001324
 074 CONTINUE 001325
 C----OUTER NOSE REGION 001326
 DD 672 LL = 1,LBHMAX 001327
 SM(3,J,LL) = WSU(3) 001328
 SM(1,J,LL) = WSU(1)+SCH(2,LL) + WSU(2)*SCH(1,LL) 001329
 SM(2,J,LL) = WSU(1)+SCH(1,LL) + WSU(2)*SCH(2,LL) 001330
 IF((SCH(1,LL),EQ,0,0),AND,(SCH(2,LL),EQ,0,0))WSH(1,J,LL)=WSU(1) 001331
 072 CONTINUE 001332
 C----INTERMEDIATE NOSE REGION 001333
 DD 674 LL = 1,LBMMAX 001334
 SM(3,J,LL) = WSU(3) 001335
 SM(1,J,LL) = WSU(1)+SCM(2,LL) + WSU(2)*SCM(1,LL) 001336
 SM(2,J,LL) = WSU(1)+SCM(1,LL) + WSU(2)*SCM(2,LL) 001337
 IF((SCM(1,LL),EQ,0,0),AND,(SCM(2,LL),EQ,0,0))WSM(1,J,LL)=WSU(1) 001338
 074 CONTINUE 001339
 C----INNER NOSE REGION 001340
 DD 676 LL = 1,LBJMAX 001341
 ST(3,J,LL) = WSU(3) 001342
 ST(1,J,LL) = WSU(1)+SCT(2,LL) + WSU(2)*SCT(1,LL) 001343
 ST(2,J,LL) = WSU(1)+SCT(1,LL) + WSU(2)*SCT(2,LL) 001344
 IF((SCT(1,LL),EQ,0,0),AND,(SCT(2,LL),EQ,0,0))WST(1,J,LL)=WSU(1) 001345
 076 CONTINUE 001346
 DATA CONTINUE 001347
 C----WRITE NEW STARTING TAPE DATA IN BINARY 001348
 L=1 001349
 WRITE(4) MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT,
 C MCN, MCN, MCF, MCN, MCN, MCS, NCT, NCA, MCN, MCE, NCF,
 C NCN, NCH, NCS, NCT,
 F MFAL, MFBL, MFDA, MFE, MFF, MFU, MFM, MFD, MFS, MFT, MFA,
 F MFATP, MFAT, MFATM, MFATH, MFA, NFH, NFU, NFE, NFF, NFG, NFH,
 F NFU, NFS, NFT, NFCL,
 H MMAL, MMAL, MHG, MHE, MHG, MHG, MHO, MHS, MHT, MNAN, MHANP,
 H MMAL, MMAL, MHG, MHE, MHG, MHG, MHO, MHS, MHT, MNCL,
 M MMA, MMAL, MMD, MMF, MMF, MMG, MMH, MVO, MMS, MMT, MMAN, MMANP,
 M NMAL, NMAL, NMAL, NMAL, NMAL, NMAL, NMAL, NMAL, NMAL, NMCL,
 T MTA, MTD, MTO, MTS, MTI, NTAN, NTANP, NTA, NTH, NTD,
 T NTAL, NTAL, NTI, NTCL,
 S NSL, MST, NSL, NST,
 WRITE(4) LUFA, LBFB, LBFC, LBFD, LBFE, LAFI, LAFJ, LAFK,
 E LMEL, LUFB, LUFBMAX,
 H LUMA, LUDU, LUHC, LUHU, LBHE, LBHI, LBHJ, LBHK, LHMU, LBHM,
 H LHMMAX,
 M LUMA, LUMD, LUMC, LUMD, LHME, LHMI, LHMJ, LBMK, LBML, LBMM,
 M LUMMAX,
 T LUTA, LUDU, LUTC, LUDU, LUTE, LHTI, LHTJ, LBTK, LBTL, LBTM,
 T LUTU, LUDU, LUTS, LUTU, LBTMX,
 WRITE(4) (IYH(J), J=1,84), (ICB(J), J=1,136),
 A ((IUFH(J)), IUFM(J)), IUMT(J)), J=1,10), (IWF(J), J=1,126),
 B ((IHSJ), J=1,6), (IWM(J), J=1,42), (INT(J), J=1,129),
 001350
 001351
 001352
 001353
 001354
 001355
 001356
 001357
 001358
 001359
 001360
 001361
 001362
 001363
 001364
 001365
 001366
 001367
 001368
 001369
 001370
 001371
 001372
 001373
 001374
 001375

C. ((LIUTF(J)), LINTR(J), LINIM(J)), J=1,21, ((IWEX(J), IWHX(J)),
 D. IWMA(J), J=1,4), ((IAF(J), J=1,74), ((AH(J), J=1,50), (IAM(J),
 E. J=1,52), ((IAT(J), J=1,32), (IAN(J), J=1,21)), 001377
 F. ((IFH(J)), ((WMI(J), JNMT(J)), J=1,8), 001379
 WRIE(4) ((ICO(J), JC0(J)), J=1,68), ((IFO(J), JFO(J)), 001380
 F. J=1,296), ((JH01(J), JH02(J)), J=1,200), ((JMO(J), JMO(J)), J=1,200), 001381
 G. ((IT01(J), JQ(J)), J=1,128), 001382
 WRIE(4) ((DX(J), DY(J)), J=1,6), ((RH(J), J=1,9), 001383
 WRIE(4) ((CY(J), SY(J)), J=1,4), ((DYH(J), J=1,44), 001384
 A. ((CU(J), J=1,130), ((XYF(J), SCF(J)), J=1,84), 001385
 H. ((XYH(J), SCH(J)), J=1,44), ((XYM(J), SCM(J)), J=1,28), 001386
 C. ((XYT(J), SGT(J)), J=1,86), 001387
 WRIE(4) ((WF(J), J=1,1512), ((WH(J), J=1,792), 001388
 A. ((AM(J), J=1,504), ((AT(J), J=1,1548), ((AWF(J), J=1,126), 001389
 B. ((WMI(J), J=1,6), ((WM(J), J=1,42), ((AVT(J), J=1,129), 001390
 C. ((WF(J), J=1,1134), ((CWH(J), J=1,594), ((CWM(J), J=1,370), 001391
 D((WT(J), J=1,1161), ((DWFX(J), DWHX(J), DWNX(J), DWTX(J)), J=1,108), 001392
 E. ((AEH(J), AWMT(J)), AWMT(J), J=1,9), 001393
 F. ((UWFH(J), DWMT(J)), DWMT(J), J=1,72), 001394
 G. ((WFX(J), CWMX(J), CWMX(J)), J=1,36), 001395
 WRIE(4) ((AM, HM), UNOG, OGM), GMU, HINF, GMU, UNOGMU, 001396
 WRIE(4) ((WC(J), J=1,1134), ((WC(J), J=1,467), 001397
 A. ((WF(J), J=1,1493), ((WH(J), J=1,1275), ((WM(J), J=1,2928), 001398
 B. ((WT(J), J=1,1008), ((WSF(J), J=1,1134), ((WSH(J), J=1,594), 001400
 C. ((SM(J), J=1,370), ((ST(J), J=1,1161), ((SU(J), J=1,3), 001401
 CALL EXIT, 001402
 STOP, 001403
 2 FORMAT(16I5), 001404
 4 FORMAT(1A19,31), 001405
 8 FORMAT(1A20,8), 001406
 10 FORMAT(15,210,5X,2E20.8), 001407
 12 FORMAT(15,214,3X,214,3X,214,3X,214,3X,214,3X,214,9X,214,3X), 001408
 A. 214,3X,214,3X,214,3X,214,1, 001409
 14 FORMAT(215,12E10.5), 001410
 16 FORMAT(15,9X,215,11F10.5), 001411
 C----(IJLES, AIC, HEADING5, 001412
 22 FORMAT(2A,9H01,0X,3H1P,2X,3H11,5X,7HCWMX(K)), 001413
 23 FORMAT(1/20X,15HYTHE,BNDY,,RJNG), 001414
 24 FORMAT(2A,9H01,0X,3ULP,2X,3H11,5X,7HCWHX(K)), 001415
 25 FORMAT(3A,2HMY,8X,2HnC,HX,2HnC,17X,4HDELX,16X,4HDELY), 001416
 26 FORMAT(2A,9H01,0X,3ULP,2X,3H11,5X,7HCWFX(K)), 001417
 27 FORMAT(1/20X,18U,LEFT,COARSE,BNDY,), 001418
 28 FORMAT(1/20X,18U,RIGHT,COARSE,BNDY,), 001419
 29 FORMAT(1/20X,18U,BOTL,COARSE,BNDY,), 001420
 30 FORMAT(1/20X,18U, TOP,COARSE,BNDY,), 001421
 31 FORMAT(3A,2HMC,8X,2HMY,8X,2HY,17X,4HDLTH,16X,7H-DELH0), 001422
 32 FORMAT(3X,2HMC,8X,2HMY,8X,2HY,17X,4HDLTH,16X,7H-DELH0), 001423
 33 FORMAT(1/20X,13HAI,ME011,SHAPE), 001424
 34 FORMAT(1/20X,1HX,1BX,3HYUS,17X,4HYPUS,16X,3HYLS,17X,4HPSL), 001425
 35 FORMAT(1/20X,17H11,01K,NOSE,REGION), 001426
 36 FORMAT(1/20X,1HY,17X,5HANOSE,14X,6HYPNOSE), 001427
 38 FORMAT(1/20X,3HINNER,NOSE,REGION,OPERATIONAL,LIMITS), 001428
 39 FORMAT(1/20X,3UH,FINE,MESH,OPERATIONAL,LIMITS), 001429
 40 FORMAT(1/20X,21HAI,FE01,INTERPOLATION), 001430

41	FORMAT(3X,4HCOL,,5X,3HIMO,2X,3HJMO)	001431
42	FORMAT(1X,4HCOL,,5X,3HIFO,2X,3HJFO)	001432
43	FORMAT(1/10X,16HFINE MESH REGION)	001433
44	FORMAT(1/10X,31MINNER NOSE REGION VERTICAL CUTS)	001434
45	FORMAT(1/10X,31MINNER NOSE REGION HORIZON, CUTS)	001435
46	FORMAT(4X,1HM,9X,1HN,9X,1HN,13X,4HUELY,16X,4HDELY)	001436
47	FORMAT(4X,21HN,9X,1HM,9X,1HM,13X,4HUELY)	001437
48	FORMAT(1/7X,5HUMACH,17X,3HUGAM,16X,5HALPHA)	001438
49	FORMAT(1X,7HQFH3111)	001439
50	FORMAT(2X,9HMSL,2X,3HNSL,2X,3HNSL)	001440
51	FORMAT(1/1UX,17HOUTER NOSE REGION)	001441
52	FORMAT(1/10X,24HJ1INTERMEDIATE NOSE REGION)	001442
53	FORMAT(1/4X,1HM,4X,1HM,7X,3HMH0,BX,2HUI,PX,2HUS)	001443
54	FORMAT(1/3X,2HMN,2X,3HUCP,7X,3HMH0,BX,2HUS,BX,2HUM)	001444
55	FORMAT(1/10X,4HMCSE MESH OPERATIONAL LIMITS)	001445
56	FORMAT(1X,4HCOL,,5X,3HJCQ,2X,3HJCQ)	001446
61	FORMAT(1A,4HCOL,,5A,3HJIO,2X,3HJIO)	001447
62	FORMAT(1A,4HCOL,,5X,3HJHO,2X,3HJHO)	001448
63	FORMAT(1A,1/10ISM111)	001449
64	FORMAT(1X,7HQDHT111)	001450
65	FORMAT(5X,39HJRF0IL SURFACE POINT LOC, ORIENTATION)	001451
66	FORMAT(3X,21HM,3X,2HMV,9X,1HY,9X,1HY,7X,3HSIN,7X,3HCUS,4X,6HOFFSE	001452
	ALL	001453
67	FORMAT(1/5X,48HINTERPOLATION WEIGHTS, DISTORTED TO ROTATED MESH)	001454
68	FORMAT(3X,2HMV,2X,3HLIP,6X,4HW(M))	001455
69	FORMAT(1/5X,51HINTERPOLATION WEIGHTS, SURFACE NORMALS TO CARTESIAN)	001456
70	FORMAT(3X,2HMV,1X,5HMUN,10,4X,1HM,4X,1HN,6X,4HW(2),6X,	001457
	A4Hn(51,6X,4HW(4))	001458
71	FORMAT(1/5X,48HINTERPOLATION WEIGHTS, CARTESIAN TO ROTATED MESH)	001459
72	FORMAT(3X,21HM,7X,3HLIP,2X,3HM_N,6X,4HW(N))	001460
73	FORMAT(1/5X,11HAWMJ(K,LIP))	001461
74	FORMAT(1/5X,11HAWJM(K,LIP))	001462
75	FORMAT(1/5X,11HAWFM(K,LIP))	001463
76	FORMAT(1X,3HKUT,12X,3HLIP,1X,4HIWMT,2X,15HDWMT(K,KUT,LIP))	001464
77	FORMAT(1/2X,3HKUT,12X,3HLIP,1X,4HIWMM,2X,15HDWMM(K,KUT,LIP))	001465
78	FORMAT(1/2X,3HKUT,12X,3HLIP,1X,4HIWFH,2X,15HDWFH(K,KUT,LIP))	001466
84	FORMAT(1/2X,13)	001467
82	FORMAT(4E20,13)	001468
90	FORMAT(1/2X,3HKUT,2X,3HLIP,5X,7HDWTX(N))	001469
91	FORMAT(1/2X,3HKUT,2X,3HLIP,5X,7HDWMX(N))	001470
92	FORMAT(1/2X,3HKUT,2X,3HLIP,5X,7H0WHX(N))	001471
93	FORMAT(1/2X,3HKUT,2X,3HLIP,5X,7H0WFX(N))	001472
94	FORMAT(1/5X,9HLINTM(1)=,15,10X,9HLINTM(2)=,15)	001473
95	FORMAT(1/5X,9HLINTH(1)=,15,10X,9HLINTH(2)=,15)	001474
96	FORMAT(1/5X,9HLINTF(1)=,15,10X,9HLINTF(2)=,15)	001475
	ENL	001476

SUBROUTINE SHAPE(NSHP,CLINE) 001477
 C-----MULTIPLY AIRFOIL GIVEN BY TABLES OF SURFACE POINTS 001478
 C----- SPLINE FIT JO POINTS IN EACH OF THREE REGIONS 001479
 C----- REGION 1 - UPPER SURFACE - POINTS ARRANGED MONOTONIC IN X FROM L.E. TO 001480
 C----- REGION 2 - LOWER SURFACE - POINTS ARRANGED MONOTONIC IN X FROM L.E. TO 001481
 C----- REGION 3 - NOSE - POINTS ARRANGED MONOTONIC IN Y FROM LOWER TO UPPER SIDE 001482
 C----- ALL X,Y IN ORDINARY AIRFOIL COORDINATES 001483
 C----- LAST 2 POINTS OF REGION 3 TO COINCIDE WITH FIRST 2 OF REGION 1 001484
 C----- FIRST POINTS 1-2 OF REGION 3 TO COINCIDE WITH POINTS 2-1 OF REGION 2 001485
 C----- NTE(J) = NUMBER OF TABLE ENTRIES IN REGION J 001486
 C----- NSHP = 1 SET UP SPLINE FIT, WRITE FITTED DATA 001487
 C----- NSHP = 2 VERTICAL CUT, GET BOTH UPPER AND LOWER SURFACE 001488
 C----- NSHP = 3 HORIZONTAL CUT IN NOSE REGION 001489
 C----- NSHP = 4 VERTICAL CUT TO GET LOWER SURFACE 001490
 C----- NSHP = 5 VERTICAL CUT TO GET UPPER SURFACE 001491
 C----- CVRT = CONVERGENCE TOLERANCE ON X OR Y 001492
 COMMUN/ALF/YUS,YLS,YPSU,YPSL,XNOSE,YPNOSE,RADCY,RADSL,RADC 001493
 DIMENSION X(101,3),Y(101,3),N(101,3),QELY(101,3),H2(101,3),
 A(1101,3),DEL5QY(101,3),S1(101,3),S2(101,3),S3(101,3), 001494
 B(1101,3),S4(3),NTE(3),NTU(3),TITLE(8), 001495
 UU TO 174+200,400,300,200)NSHP 001497
 ZU READ(5,1) (TITLE(I), I = 1,8) 001498
 READ(5,1) (TITLE(I), I = 1,8)
 READ(5,2) CVHT, EPSLN 001499
 READ(5,2) EPSLN 001500
 READ(5,4) NTE(J), J = 1,3 001502
 UU M2 KR = 1,2 001503
 NT = NTE(KR) 001504
 NJN(KR) = NJE(KR) - 1 001505
 UU M2 J = 1,NT 001506
 B2 READ(5,2) X(I,KR), Y(I,KR) 001507
 KR = 3 001508
 UJ = NTE(KR) 001509
 NTU(KR) = NTE(KR) - 1 001510
 UU M2 I = 1,NT 001511
 B4 READ(5,2) Y(I,KR), X(I,KR) 001512
 UU UU KR = 1,3 001513
 NS = NJD(KR) 001514
 UU M2 J = 1,NS 001515
 UU(I,KR) = X(I+1,KR) - X(I,KR) 001516
 BB UEL(Y(I,KR) = (Y(I+1,KR) - Y(I,KR))/H(I,KR) 001517
 UU UU I = 2,NS 001518
 H2(I,KR) = H(I-1,KR) + H(I,KR) 001519
 U(I,KR) = U(5+H(I-1,KR))/H2(I,KR) 001520
 UELSY(I,KR) = IDELY(I,KR) - QELY(I-1,KR)/H2(I,KR) 001521
 S2(I,KR) = 2.0*DEL5QY(I,KR) 001522
 Q0(I,J,KR) = J,0*DEL5QY(I,KR) 001523
 LA = 1 LB = 2 LC = NJU(I) LD = NTE(I)
 LF = NTE(2) LF = NTU(2) LG = 2 LH = 1 001524
 LI = 1 LU = 2 LK = NJU(3) LM = NTE(3) 001525
 S2(L0,1) = 0.0 001527
 S2(LF,2) = 0.0 001528
 I = LB SKR = 1 001529
 S1(I,KR) = IDELY(I,KR) - U(I,KR)+(2.0*S2(I,KR)+S2(I+1,KR))/6.0 001530

```

I = LG . S(KR) = 2 ..... 001531
S1(I,KR) = DELY(I,KR) - H(I,KR)*(2.0+S2(I,KR)+S2(I+1,KR))/6.0 .. 001532
I = LJ . S(KR) = 3 ..... 001533
S1(I,KR) = DELY(I,J,KR) - H(I,KR)*(2.0+S2(I,KR)+S2(I+1,KR))/6.0 .. 001534
I = LK . S(KR) = 3 ..... 001535
S1(I,KR) = DELY(I-1,KR)+H(I-1,KR)*(S2(I-1,KR)+2.0+S2(I,KR))/6.0 .. 001536
S1(LJ,3) = 1.0/S1(LG,2) ..... 001537
S1(LH,2) = 1.0/S1(LJ,3) ..... 001538
S1(LN,3) = 1.0/S1(LH,3) ..... 001539
S1(LA,3) = 1.0/S1(LK,3) ..... 001540
S2(LG,2) = S2(LG,2)*((1.0+S1(LI,3)**2)/(1.0+S1(LG,2)**2))**1.5 .. 001541
S2(LH,2) = S2(LH,2)*((1.0+S1(LI,2)**2)/(1.0+S1(LH,2)**2))**1.5 .. 001542
S2(LM,3) = S2(LM,3)*((1.0+S1(LM,3)**2)/(1.0+S1(LH,1)**2))**1.5 .. 001543
S2(LA,1) = S2(LK,3)*((1.0+S1(LA,1)**2)/(1.0+S1(LK,3)**2))**1.5 .. 001544
C----OVERRELAXATION TO FILE POLYNOMIALS TO DATA ..... 001545
OMEGA = 1.4717468 ..... 001546
105 EIA = 0.0 ..... 001547
DQ.115.KR = 1e3 ..... 001548
NS = NJR(KM) ..... 001549
NT = NTE(KM) ..... 001550
DU.110.I = 2.eNS ..... 001551
R = (S1(I,KR)-P(I,KR)+S2(I-1,KR)-(0.5-R(I,KR))+S2(I+1,KR)) ..... 001552
A = S2(I,KR))*OMEGA ..... 001553
JF(AUS(NL = EIA).110,110,109 ..... 001554
109.EIA = ABS(NL) ..... 001555
110.S2(I,KR) = S2(I,KR) + W ..... 001556
115.CONTINUE ..... 001557
I = LB . S(KR) = 1 ..... 001558
S1(I,KR) = DELY(I,KR) - H(I,KR)*(2.0+S2(I,KR)+S2(I+1,KR))/6.0 .. 001559
I = LG . S(KR) = 2 ..... 001560
S1(I,KR) = DELY(I,KR) - H(I,KR)*(2.0+S2(I,KR)+S2(I+1,KR))/6.0 .. 001561
I = LJ . S(KR) = 3 ..... 001562
S1(I,KR) = DELY(I,KR)-H(I,KR)+(S2(I-1,KR)+2.0+S2(I,KR))/6.0 .. 001563
I = LK . S(KR) = 3 ..... 001564
S1(I,KR) = DELY(I-1,KR)+H(I-1,KR)*(S2(I-1,KR)+2.0+S2(I,KR))/6.0 .. 001565
S1(LI,3) = 1.0/S1(LG,2) ..... 001566
S1(LH,2) = 1.0/S1(LJ,3) ..... 001567
S1(LM,3) = 1.0/S1(LH,1) ..... 001568
S1(LA,1) = 1.0/S1(LK,3) ..... 001569
S2(LG,2) = S2(LG,2)*((1.0+S1(LI,3)**2)/(1.0+S1(LG,2)**2))**1.5 .. 001570
S2(LH,2) = S2(LH,2)*((1.0+S1(LI,2)**2)/(1.0+S1(LH,2)**2))**1.5 .. 001571
S2(LM,3) = S2(LM,3)*((1.0+S1(LM,3)**2)/(1.0+S1(LB,1)**2))**1.5 .. 001572
S2(LA,1) = S2(LK,3)*((1.0+S1(LA,1)**2)/(1.0+S1(LK,3)**2))**1.5 .. 001573
IF(EIA = EPSLN) 120,105,105 ..... 001574
C----COMPLETION OF DATA ON POLYNOMIALS FITTED ..... 001575
C----INCLUS PATCHING IN OF 4TH DEGREE POLYNOMIAL AT OVERLAPS ..... 001576
120 NO.130.KR = 1.e2 ..... 001577
WR1(E16,14).KR ..... 001578
NS = NJR(KM) ..... 001579
NT = NTE(KM) ..... 001580
DQ.122.I = 2.eNS ..... 001581
S3(I,KR) = (S2(I+1,KR) - S2(I,KR))/H(I,KR) ..... 001582
122.S1(I,KR) = DELY(I,KR) - H(I,KR)*(2.0+S2(I,KR)+S2(I+1,KR))/6.0 .. 001583
I = NT ..... 001584
S1(I,KR) = DELY(I-1,KR)+H(I-1,KR)*(S2(I-1,KR)+2.0+S2(I,KR))/6.0 .. 001585

```

```

S4(KR) = 6.0*(S2(2,KR)+S2(1,KR) - (S1(2,KR)-S1(1,KR))/H(1,KR)) 001586
A / (H(1,KR)**2) 001587
S3(1,KR) = (S2(2,KR)-S2(1,KR))/H(1,KR) = 0.5*H(1,KR)*S4(KR) 001588
KR=16,221,54(KR) 001589
WRATE(6,12) 001590
UO_126_I = 1,NT 001591
IF(S2(1,I,KR))_124,123,124 001592
123 S2(I,KR) = 1.0E-08 001593
124 RAUC = (1.0+S1(I,KR)**2)**1.5/S2(I,KR) 001594
126 WRATE(6,6),I,J,XII,KR,Y(I,KR),S1(I,KR),S2(I,KR),RADC 001595
130 CONTINUE 001596
KR = 3 001597
WRATE(6,14),KR 001598
NS = NTD(KR) 001599
NR = NS - 1 001600
UO_132_I = 2,NR 001601
S3(I,KR) = (S2(I+1,KR) - S2(I,KR))/H(I,KR) 001602
132 S1(I,KR) = DELY(I,I,KR) = H(I,KR)*(2.0*S2(I,KR)+S2(I+1,KR))/6.0 001603
WRATE(6,24) 001604
UO_142_I = 2,NS 001605
RAUC = (1.0+S1(I,I,KR)**2)**1.5/ABS(S2(I,KR)) 001606
IF(S1(1,I,KR))_136,134,138 001607
134 SINV1 = 1.0E+08 001608
SINV2 = 1.0E+08 001609
GO TO 140 001610
136 SINV1 = 1.0/S1(I,KR) 001611
SINV2 = (1.0+SINV1**2)**1.5/RADC 001612
GO TO 140 001613
138 SINV1 = 1.0/S1(I,KR) 001614
SINV2 = -(1.0+SINV1**2)**1.5/RADC 001615
140 WRATE(6,6),I,J,XII,KR,X(I,KR),S1(I,KR),S2(I,KR),S3(I,KR), 001616
A RAUC, SINV1, SINV2 001617
142 CONTINUE 001618
GO TO 500 001619
C----X GIVEN, UPPER SURFACE WANTED 001620
200 XU = CLINE 001621
KR = 1 001622
IF(XU = 1.0) 202,250,250 001623
202 IF(XU = X(I,KR))_210,260,260 001624
C A IS IN REGION GIVEN BY X=FUNCTION OF Y 001625
210 KR = 3 001626
I = NTD(KR) 001627
212 IF(XU = Y(I,KR))_214,215,215 001628
214 I = I - 1 001629
IF(I = 21) 240,212,212 001630
C----INITIAL VALUES FOR ITERATION 001631
215 IF(S1(I,KR))_2160,210,216 001632
2160 I = I + 1 001633
XINC = XU - Y(I,KR) 001634
YINC = XINC/S1(I,KR) 001635
2180 ATRY = Y(I,KR) + YINC*(S1(I,KR)) + 0.5*YINC*(S2(I,KR)) 001636
A + YINC*S3(I-1,KR)/3.0,I 001637
AP = S1(I,KR) + YINC*(S2(I,KR)) + 0.5*YINC*S3(I-1,KR) 001638
XPR = S2(I,KR) + YINC*S3(I-1,KR) 001639
IF(ABS(XU - XTRY))_230,230,220 001640

```

```

2200 YINC = YINC + (X0 - XTRY)/XP          001641
GO TO 2180                                  001642
216 XINC = XU - Y(I,KR)                      001643
YINC = XINC/DELY(I,KR)                      001644
C----ITERATE TO GET DESIRED X CUT           001645
218 XTRY = Y(I,KR) + YINC*(S1(I,KR) + 0.5*YINC*(S2(I,KR)
A + YINC*S3(I,KR)/3.0))                   001646
XP = S1(I,KR) + YINC*(S2(I,KR) + 0.5*YINC*S3(I,KR)) 001647
YPP = S2(I,KR) + YINC*S3(I,KR)              001648
IF(ABS(XU - XTRY) - CYRT(230,230,220)     001649
220 YINC = YINC + (XD - XTRY)/XP          001650
GO TO 214                                  001651
230 YUS = X(I,KR) + YINC                  001652
YPSU = 1.0/XP                            001653
YADCU = -(1.0 + YPSU**2)**1.5/YPP        001654
GO TO 1500,300,500,500,500,500,NSHP      001655
C----X VALUE DESIRED IS OUT OF RANGE AT NOSE 001656
240 YADCU=16,261,XD                         001657
GO TO 10,500                                001658
C----X IS IN REGION AFT OF TRAILING EDGE    001659
250 I = 1,I(KR)                            001660
XINC = XU - X(I,KR)                      001661
YUS = Y(I,KR) + XINC*S1(I,KR)            001662
YPSU = S1(I,KR)                          001663
YADCU = -1.0E+0B                          001664
GO TO (500,300,500,500,500),NSHP        001665
C----X IS IN REGION DESCRIBED BY Y = FUNCTION OF X 001666
260 I = 1                                  001667
262 IF(XD - X(I,KR)) 266,268,264        001668
264 I = I + 1                            001669
GO TO 262                                  001670
266 I = I - 1                            001671
268 XINC = XD - X(I,KR)                  001672
YUS = Y(I,KR) + XINC*(S1(I,KR) + 0.5*YINC*(S2(I,KR)
A + XINC*S3(I,KR)/3.0))                 001673
YPSU = S1(I,KR) + XINC*(S2(I,KR) + 0.5*YINC*S3(I,KR)) 001674
YPP = S2(I,KR) + XINC*S3(I,KR)            001675
IF(I.GT.1) GO TO 270                     001676
YUS = YUS + XINC**4*S4(KR)/24.0          001677
YPSU = YPSU + XINC**3*S4(KR)/6.0          001678
YPP = YPP + 0.5*XINC**2*S4(KR)            001679
270 YADCU = -(1.0 + YPSU**2)**1.5/YPP    001680
GO TO 1500,300,500,500,500,500,NSHP      001681
C----X GIVEN, LOWER SURFACE WANTED        001682
300 XU = CLINE                           001683
KR = 2                                    001684
IF(XD - 1.0) 302,350,350                001685
302 IF(XU = X(I,KR)) 310,360,360        001686
C----X IS IN REGION GIVEN BY X=FUNCTION OF Y 001687
310 KR = 3                                001688
I = 2                                    001689
312 IF(XU = Y(I,KR)) 314,316,316        001690
314 I = I + 1                            001691
IF(I = NTU(KR)) 312,312,240             001692
C----INITIAL VALUES FOR ITERATION         001693

```

310 YINC = XU - Y(I,KR) 001696
 YINC = XINC/DELY(I-1,KR) 001697
 C----1TEHATE TO GET DESI(HCD,X,CUT) 001698
 310 XTRY1 = Y(I,KR) + YINC*(S1(I,KR)) + 0.5*YINC*(S2(I,KR)) 001699
 A + YINC*S3(I-1,KR)/3.0) 001700
 AP = S1(I,KR) + YINC*(S2(I,KR)) + 0.5*YINC*S3(I-1,KR) 001701
 APP = S2(I,KR) + YINC*S3(I-1,KR) 001702
 IF(AHS(XU - XTRY1) = GVRT)330,330,320 001703
 320 YINC = YINC + (XD - XTRY1)/XP 001704
 GO TO 318 001705
 330 YLS = X(I,KR) + YINC 001706
 YPSL = 1.0/XP 001707
 RADCL = (1.0 + XP*+2)**1.5/XPP 001708
 GO TO 500 001709
 C----2. LS IN REGION I, AFJ. OF TRAILING EDGE 001710
 350 L = JTE(KR) 001711
 XINC = XU - X(I,KR) 001712
 YLS = Y(I,KR) + XINC*S1(I,KR) 001713
 YPSL = S1(I,KR) 001714
 RADCL = 1.0E+08 001715
 GO TO 500 001716
 C----3. X IS IN REGION DESCRIBED BY Y = FUNCTION OF X 001717
 360 L = 1 001718
 362 IF(XU - X(I,KR))=366,368,364 001719
 364 L = 1 + 1 001720
 GO TO 362 001721
 366 L = 1 - 1 001722
 368 XINC = XU - X(I,KR) 001723
 YLS = Y(I,KR) + XINC*(S1(I,KR)) + 0.5*XINC*(S2(I,KR)) 001724
 A + XINC*S3(I,KR)/3.0) 001725
 YPSL = S1(I,KR) + XINC*(S2(I,KR)) + 0.5*XINC*S3(I,KR) 001726
 YPP = S2(I,KR) + XINC*S3(I,KR) 001727
 IT(31,GT,11,GO TO 370 001728
 YLS = YLS + XINC*+4*S4(KR)/24.0 001729
 YPSL = YPSL + XINC*+3*S4(KR)/6.0 001730
 YPP = YPP + 0.5*XINC*+2*S4(KR) 001731
 370 RADCL = (1.0 + YPSL*+2)**1.5/YPP 001732
 GO TO 500 001733
 C----4. Y GIVEN, HORIZONTAL CUT 001734
 400 YD = CLINE 001735
 KR = 3 001736
 NS = NJD(KR) 001737
 IF(YU - X(2,KR))=440,402,492 001738
 402 IF(YU - X(NS,KR))=400,406,440 001739
 406 L = 2 001740
 408 IF(YU - X(I,KR))=412,414,410 001741
 410 L = 1 + 1 001742
 GO TO 408 001743
 412 L = 1 - 1 001744
 414 YINC = YD - X(I,KR) 001745
 AINC = Y(I,KR) + YINC*(S1(I,KR)) + 0.5*YINC*(S2(I,KR)) 001746
 A + YINC*S3(I,KR)/3.0) 001747
 AP = S1(I,KR) + YINC*(S2(I,KR)) + 0.5*YINC*S3(I,KR) 001748
 APP = S2(I,KR) + YINC*S3(I,KR) 001749
 RADCL = (1.0 + XP*+2)**1.5/XPP 001750

YPOUSE = 1.0/XP	001751
GO TO 500	001752
C----Y OUT OF RANGE IN WHICH FUNCTION IS WELL DEFINED	001753
440 WRITE(5,20)YD	001754
500 RETURN	001755
1 FORMAT(8A10)	001756
2 FORMAT(8E10,3)	001757
3 FORMAT(23X,8A10)	001758
4 FORMAT(16I5)	001759
6 FORMAT(5X,1D,8E15,3)	001760
12 FORMAT(/5X,5HE10H,14X,1HX,1HY,13X,2HXP,12X,3HYP,11X,	001761
A 4HPPP,11X,4HADCI)	001762
14 FORMAT(/5X,6HREGION,15)	001763
18 FORMAT(/5X,8HEPSLN = ,E10,3)	001764
22 EOMAT1/5X,1BHYPFFFF AT ENIHY,1 E,F10,3)	001765
24 EOMAT1/5X,1MEUJRY,14X,1HX,14X,1HY,13X,2HXP,12X,3HYP,11X,	001766
A 4HPPP,11X,4HADCI,10X,2HYP,12X,3HYP)	001767
26 FORMAT(7X,2HX=,E10,3,11HNOT ALLOWED)	001768
28 FORMAT(7X,2HY=,E10,3,11HNOT ALLOWED)	001769
END	001770

```

----- SUBROUTINE CAXIS(KR,MH,MS,MAN,NCL,LCU,LCL,IA,DA,SA,CA,LA) ----- 001771
----- A XY,SC,DS,LF1 ----- 001772
C. DETERMINES GEOMETRY OF LOCAL SURFACE NORMALS ----- 001773
----- COMMON/ZMESH/, DA(6), DY(6), RH(9) ----- 001774
----- COMMON/AIRE/YUS,YLS,YPSU,YPSL,XNOSE,YPNOSE,RAQCU,RAQCL,RAQC ----- 001775
----- DIMENSION IA(2,LA), L, DA(2,LA), SA(2,LA), CA(2,LA), XY(2,LF1) ----- 001776
----- A SC(2,LF1), DS(LF1) ----- 001777
----- WRITE(16,81) ----- 001778
----- DO 20 M = MH, MS ----- 001779
----- DO 20 KAT = 1, 2 ----- 001780
----- XA = FLOAT(M - MAN)*UX(KR) ----- 001781
----- XTEST = XA ----- 001782
----- I = IA(KAT,M) ----- 001783
----- YA = FLOAT(I - NCL)*UY(KR) ----- 001784
----- XEKR = DA(KAT,M)*DY(KR)*SA(KAT,M)*CA(KAT,M) ----- 001785
----- GO TO 11, 12, KAT ----- 001786
----- 11 MM = LCL - M ----- 001787
----- KSP = 4 ----- 001788
----- GO TO 13 ----- 001789
----- 12 MM = M + LCU ----- 001790
----- KSP = 5 ----- 001791
----- XEKR = -XEKR ----- 001792
----- 13 XTEST = XTEST - XEKR ----- 001793
----- CALL SHAPE(KSP, XTEST) ----- 001794
----- GO TO (14, 15), KAT ----- 001795
----- 14 YS = YLS ----- 001796
----- YPS = YPSL ----- 001797
----- GO TO 16 ----- 001798
----- 15 YS = YUS ----- 001799
----- YPS = YPSU ----- 001800
----- 16 XEKR = (XTEST-XA+(YS-YA)*YPS)/(1.0 + YPS**2) ----- 001801
----- IF(AHS(XEKR)) = .0.D1*UY(KR)) 17, 13 ----- 001802
----- 17 XY(1,MM) = XTEST ----- 001803
----- XY(2,MM) = YS ----- 001804
----- SC(2,MM) = 1.0/SURJ(1.0 + YPS**2) ----- 001805
----- SC(1,MM) = YPS + SC(2,MM) ----- 001806
----- DS(M,M) = SURJ((YA - YS)**2 + ((XA - XTEST)**2)/DY(KR)) ----- 001807
----- GO TO (18, 20), KAT ----- 001808
----- 18 SC(1,MM) = -SC(1,MM) ----- 001809
----- SC(2,MM) = -SC(2,MM) ----- 001810
----- 20 WRITE(16,41) M, XY(1,MM), XY(2,MM), SC(1,MM), SC(2,MM), DS(M,M) ----- 001811
----- RETURN ----- 001812
----- 4 FORMAT(2I5, 12E10.5) ----- 001813
----- 5 FORMAT(14X, 1HM, 5X, 2HL0, 5X, 1HX, 9X, 1HY, 9X, 3HSIN, 7X, 3HCOS, 6X, 6HQFFSET1) ----- 001814
----- END ----- 001815

```

```

SUBROUTINE PONUER( KSPL, MLL, MUL, XTZ, YTZ, XYT, SCT, UST, DWT, DWTX, 001816
  A, WTI, CWT, INT, LUX, MTI) 001817
C CALCULATES INTERPOLATION WEIGHTS TO BE APPLIED IN TRANSFERRING FROM 001818
C DISTORTED MESH TO ADJACENT LOCAL ROTATED MESH (DWT) AND (DWTX), FROM 001819
C SURFACE NORMALS TO CARTESIAN AUXILIARIES, (AWT), AND FROM CARTESIAN GR001820
C TO LOCAL ROTATED MESH (CWT). 001821
COMMON/ZMESH/,DX(16),UY(6),RH(9) 001822
COMMON/SPUT/M,KP 001823
COMMON/AIRK/YUS,YLS,YP,SU,YPSL,XNOSL,YPNQSE,RADC1,RADC2,RADC3 001824
DIMENSION XYT(2,LBX), SCL(2,LBX), UST(LBX), DWT(9,4,LBX), 001825
A,DX(19,2,6),AWT(3,LBX),CWT(9,3,LBX),INT(3,LBX) 001826
950.00.400.M.E.MLL,MUL 001827
CALL DISTW(1,0,0,0,0,0,XYT,SCT,LBX) 001828
LIP = 1 001829
XP = XYT(1,M-1) + DX(KR)*SCT(2,M-1) 001830
YP = XYT(2,M-1) + DX(KR)*SCT(1,M-1) 001831
CALL DISTW(2,XP,YP,UWT(1,LIP,M),XYT,SCT,LBX) 001832
LIP = 2 001833
XP = XP - UY(KR)*SCT(1,M-1) 001834
YP = YP + UY(KR)*SCT(2,M-1) 001835
CALL DISTW(2,XP,YP,UWT(1,LIP,M),XYT,SCT,LBX) 001836
LIP = 3 001837
XP = XYT(1,M+1) - DX(KR)*SCT(2,M+1) 001838
YP = XYT(2,M+1) - DX(KR)*SCT(1,M+1) 001839
CALL DISTW(2,XP,YP,UWT(1,LIP,M),XYT,SCT,LBX) 001840
LIP = 4 001841
XP = XP - UY(KR)*SCT(1,M+1) 001842
YP = YP + UY(KR)*SCT(2,M+1) 001843
CALL DISTW(2,XP,YP,UWT(1,LIP,M),XYT,SCT,LBX) 001844
C-----SPECIAL INTERPOLATION WEIGHTS AT ENDS OF OPERATING RANGES 001845
IF(M.EQ.MLL) AND (M.NE.MUL) GO TO 951 001846
KSPL = KSPL + 1 001847
LIP = 1 001848
XP = XYT(1,M) - DX(KR)*SCT(2,M) 001849
YP = XYT(2,M) - DX(KR)*SCT(1,M) 001850
IF(M.EQ.MUL) XP = XP + 2.0*UX(KR)*SCT(2,M) 001851
IF(M.EQ.MUL) YP = YP + 2.0*UX(KR)*SCT(1,M) 001852
CALL DISTW(2,XP,YP,UWT(1,LIP,KSPL),XYT,SCT,LBX) 001853
LIP = 2 001854
XP = XP - UY(KR)*SCT(1,M) 001855
YP = YP + UY(KR)*SCT(2,M) 001856
CALL DISTW(2,XP,YP,UWT(1,LIP,KSPL),XYT,SCT,LBX) 001857
951 AWT(1,M) = 0.5*(1.0 - UST(M))*(2.0 - DST(M)) 001858
AWT(2,M) = UST(M)*(2.0 - DST(M)) 001859
AWT(3,M) = 0.5*DST(M)*(UST(M) - 1.0) 001860
LIP = 2 001861
XP = XYT(1,M) - 2.0*UY(KR)*SCT(1,M) 001862
YP = XYT(2,M) + 2.0*UY(KR)*SCT(2,M) 001863
952 MK = 1 001864
XT = XT2 001865
954 IF(XT - XP) 956,958,958 001866
956 AT = XT + UX(KR) 001867
MK = MK + 1 001868
GO TO 954 001869

```

958 AP = XP - XT/0x1KRI	001870
IE(AP + 0.5) 960,960,961	001871
960 MK = NK - 1	001872
AP = 1.0 + AP	001873
961 IF(MK+1-MTII) 963,963,962	001874
962 MK = MK - 1	001875
AP = 1.0 + AP	001876
963 NK = 1	001877
YJ = YJ2	001878
964 IF(YT - YH) 960,968,968	001879
964 YT = YT + UX(KR)	001880
NK = NK + 1	001881
GO TO 964	001882
965 LP = (YP - YI)/DY(KR)	001883
IF(LP + 0.5) 970,970,972	001884
970 MK = NK - 1	001885
LP = 1.0 + LP	001886
972 CALL CART4(AP,LP,CNT(1,LIP,M))	001887
INT(LIP,M) = 100*MK + NK	001888
GO TO 976,974,980,LIP	001889
974 LIP = 1	001890
AP = AP - UX(KR)*SCI(2,M)	001891
YP = YP - UX(KR)*SCI(1,M)	001892
GO TO 952	001893
976 LIP = 3	001894
AP = AP + 2.0*DX(KR)*SCI(2,M)	001895
LP = LP + 2.0*UX(KR)*SCI(1,M)	001896
GO TO 952	001897
980 CONTINUE	001898
RETURN	001899
END	001900

```

SUBROUTINE BUNTE (NE,NH,IFU,JFO,JO) 001901
C. INTERPOLATION INSTRUCTIONS AT FIELD BOUNDARIES 001902
C. DIMENSION ID(10), IF0(8), JFU(8) 001903
C. CODE 1 INERT, CODE 2 SYMMETRIC 4 POINT SPREAD, CODE 3 PARABOLA WITH 001904
C. POINTS BELOW, CODE 4 PARABOLA WITH TWO POINTS ABOVE 001905
C. J = 1 001906
10 JCT = MOD(JFO(J),100) 001907
SF(JCT).14,14,12 001908
12 J = J + 1 001909
GO TO 10 001910
14 N1 = IF0(J) 001911
N2 = IF0(J+1) - 1 001912
NJ = NH - 1 001913
J = 0 001914
00,26 N = NE, NJ 001915
J = J + 1 001916
IF(N+1=N1).20,22,16 001917
16 IF(N = N2).18,24,20 001918
18 IU(J) = 1 001919
GO,19,26 001920
20 ID(J) = 2 001921
GO TO 26 001922
22 ID(J) = 3 001923
GO,19,26 001924
24 IU(J) = 4 001925
26 CONTINUE 001926
RETURN 001927
ENU 001928

```

```

SUBROUTINE DISTW(MD,XP,YP,TW,XYU,SCU,LDIM) 001929
C----CALCULATES WEIGHTS (TW) TO BE USED ON VALUES OF F AT 9 POINTS IN A 001930
C----DISTORTED ARRAY TO FIND INTERPOLATED VALUE OF F AT CARTESIAN POINT X001931
C----MD = 1, BASIC PARAMETER, DETERMINED 001932
C----MD = 2, WEIGHTS (TW) GENERATED FOR SPECIFIC XP,YP 001933
COMMON/SPOJ/M,KB 001934
DIMENSION SSK(2), CSK(2), A(15), B(15), CR(9,9), DPW(9), DPPU(9), 001935
A(DP(9)), CPPL(9), TW(9), XYU(2,LDIM), SCU(2,LDIM) 001936
COMMON/U/ZMESM/, DX(6), DY(6), RHS(9) 001937
GO TO (100,200),MD 001938
C----CALCULATION OF INITIAL FITTING INFORMATION 001939
C----SKENESS OF SIDE LINES 001940
100 SSK(2) = SCU(1,M+1)*SCU(2,M) - SCU(2,M+1)*SCU(1,M) 001941
SSK(1) = SCU(1,M-1)*SCU(2,M) - SCU(2,M-1)*SCU(1,M) 001942
DO 102 N = 1,2 001943
102 CSK(N) = 5*HT(1,0) - SSK(N)*#2 001944
C----COORDINATES OF NINE POINTS 001945
DO 104 M = 1,3,2 001946
IF(N,EQ,1) MM = M - 1 001947
IF(N,EQ,3) MM = M + 1 001948
A(N) = ((XYU(1,MM) - XYU(1,M)) * SCU(2,M) + (XYU(2,MM) - XYU(2,M)) * 001949
A * SCU(1,M)) / DY(KR) 001950
104 L(N) = ((-XYU(1,MM) - XYU(1,M)) * SCU(1,M) + (XYU(2,MM) - XYU(2,M)) * 001951
A * SCU(2,M)) / DY(KR) - 1.0 001952
A(4) = A(1) - SSK(1) 001953
A(7) = A(4) - SSK(1) 001954
A(10) = A(3) - SSK(2) 001955
A(9) = A(6) - SSK(2) 001956
DO 105 N = 2,8,3 001957
105 A(N) = 0.0 001958
B(4) = B(1) + CSK(1) 001959
B(7) = B(4) + CSK(1) 001960
B(10) = B(3) + CSK(2) 001961
B(4) = B(6) + CSK(2) 001962
B(2) = -1.4 001963
B(5) = 0.0 001964
B(8) = 1.0 001965
C----COORDINATES OF SIX AUXILIARIES 001966
DO 106 N = 1,1 001967
A(B+2*N) = A(1) * (B(-1+3*N) - H(1)) / CSK(1) 001968
B(B+2*N) = B(-1+3*N) 001969
A(9+2*N) = A(3) - SSK(2) * (B(-1+3*N) - H(3)) / CSK(2) 001970
106 B(4+2*N) = B(-1+3*N) 001971
WRITE(16,101) KR, M 001972
WRITE(16,121) 001973
DO 140 J = 1,15 001974
140 WRITE(16,21) A(J), B(J) 001975
C----BASIC GEOMETRIC INPUT WEIGHTING FUNCTIONS 001976
DO 108 L = 1,9 001977
DO 108 K = 1,9 001978
108 CR(K,L) = 0.0 001979
CR(1,5) = 1.0 001980
CR(2,8) = 0.5 001981
CR(2,2) = -0.5 001982

```

CB(3,2) = 0,5	001983
CB(3,8) = 0,5	001984
CH(3,5) = -1,0	001985
ALC = A(2) - A(10)	001986
ALD = A(5) - A(12)	001987
ALE = A(8) - A(14)	001988
ARC = SA(18) - B(2)/ALC	001989
ARD = A(13) - A(5)/ALD	001990
ARE = SA(19) - A(8)/ALE	001991
BINCL = 1,0/(B(4) - B(1))	001992
SHCL = (B(10) - B(4))*BINCL	001993
SBUL = (B(12) - B(4))*BINCL	001994
SBEL = (B(14) - B(4))*BINCL	001995
BINCR = 1,0/(B(6) - B(3))	001996
SBGR = SB(11) - B(6)*BINCR	001997
SBUR = SB(13) - B(6)*BINCR	001998
SHER = (B(15) - B(6))*BINCR	001999
DENT = 1,0*(ARD + ARD*(ARD + 1,0))	002000
CH(4,5) = (ARD**2 - 1,0)*DENT	002001
CH(4,3) = 0,5*SBUR*(SBUR - 1,0)*DENT	002002
CH(4,6) = 1,0 - SBUR**2*DENT	002003
CH(4,9) = 0,5*SBUR*(SBUR + 1,0)*DENT	002004
CMULT = -ARD**2*DENT	002005
CH(4,1) = CMULT*0,5*SPDL*(SBUL - 1,0)	002006
CH(4,4) = CMULT*(1,0 - SBUL**2)	002007
CH(4,7) = CMULT*0,5*SPDL*(SBUL + 1,0)	002008
DENT = DENT/ALD	002009
CB(7,5) = -(ARD + 1,0)*DENT	002010
CMULT = 1,0/ALD	002011
DO_112_K = 3,9e3	002012
112 CB(4,K) = CB(4,K)*CMULT	002013
CMULT = -CMULT/ARD	002014
DO_114_K = 1,7e3	002015
114 CB(7,K) = CB(4,K)*CMULT	002016
DO_116_L = 1,9	002017
DPU(L) = 0,0	002018
DPPU(L) = 0,0	002019
DPL(L) = 0,0	002020
116 DPPL(L) = 0,0	002021
DENT = 1,0/SARE*(SARE + 1,0)*ALE	002022
DPU(6) = (ALE**2 - 1,0)*DENT	002023
DPU(3) = 0,5*SHER*(SHER - 1,0)*DENT	002024
DPU(6) = (1,0 - SHER**2)*DENT	002025
DPU(4) = 0,5*SHER*(SHER + 1,0)*DENT	002026
CMULT = -ALE**2*DENT	002027
DPU(1) = CMULT*0,5*SUFL*(SBEL - 1,0)	002028
DPU(4) = CMULT*(1,0 - SBEL**2)	002029
DPU(7) = CMULT*0,5*SUFL*(SBEL + 1,0)	002030
DENT = DENT/ALE	002031
DPPU(K) = DPU(K)*CMULT	002032
CMULT = -CMULT/ALE	002033
DO_118_K = 3,9e3	002034
118 DPPU(K) = DPU(K)*CMULT	002035
CMULT = -CMULT/ALE	002036
DO_120_K = 1,7e3	002037

```

120 DPPU(K) = DPU(K)*CMULT ..... 002038
DENT = 1.0/(ARC*(ARC + 1.0)*ALC) ..... 002039
DPL(2) = ARC**2 - 1.0)*DENT ..... 002040
DPL(3) = 0.5*SUCR*(SUCR - 1.0)*DENT ..... 002041
DPL(4) = (1.0 - SMCR**2)*DENT ..... 002042
DPL(9) = 0.5*SUCR*(SUCR + 1.0)*DENT ..... 002043
CMULT = -ARC**2*DENT ..... 002044
DPL(1) = CMULT*0.5*SMCL*(SMCL - 1.0) ..... 002045
DPL(4) = CMULT*1.0 - SMCL**2) ..... 002046
DPL(7) = CMULT*0.5*SMCL*(SMCL + 1.0) ..... 002047
DENT = DENT/ALC ..... 002048
DPL(2) = -(ARC + 1.0)*DENT ..... 002049
CMULT = 1.0/ALC ..... 002050
00 122 K = 3,9,3 ..... 002051
122 UPPL(K) = CMULT*DPL(K) ..... 002052
CMULT = -CMULT/ARC ..... 002053
00 124 K = 1,7,3 ..... 002054
124 UPPL(8) = CMULT*DPL(K) ..... 002055
00 130 K = 1,9 ..... 002056
CH(5,K) = 0.5*(DPU(K) - DPL(K)) ..... 002057
CH(8,K) = 0.5*(DPU(K) - DPL(K)) ..... 002058
CH(6,K) = 0.5*(DPU(K) + DPL(K)) - CB(4,K) ..... 002059
130 CM(9,K) = 0.5*(DPU(K) + DPL(K)) - CB(7,K) ..... 002060
..R1E(6,18).KR = M ..... 002061
..R1E(6,14). ..... 002062
00 144 J = 1,9 ..... 002063
144 R1E(6,2).J = (CB(J,K)).K = 1,9 ..... 002064
RETURN ..... 002065
C----USE OF BASIC INFORMATION TO GET WEIGHTING FACTORS FOR SPECIFIC POINTS 002066
200 AP = ((XP - XYU(1,M))*SCU(2,M) + (YP - XYU(2,M))*SCU(1,M))/DY(KR) ..... 002067
DP = ((XP - XYU(1,M))*SCU(1,M) + (YP - XYU(2,M))*SCU(2,M))/DY(KR) ..... 002068
A/DY(KR) = 1.0 ..... 002069
00 202 J = 1,9 ..... 002070
202 IK(J) = CE(1,J) + AP*(CH(4,J) + AP*CB(7,J)) ..... 002071
A + AP*(CH(2,J) + AP*(CH(5,J) + AP*CH(8,J))) ..... 002072
B + AP*(CH(3,J) + AP*(C3(6,J) + AP*CB(9,J))) ..... 002073
RETURN ..... 002074
2 FORMAT (5X,1HJ,6X,4HA(J),6X,4HB(J)) ..... 002075
12 FORMAT (/9X,1HJ,6X,15HCH(J,K),K=1,9) ..... 002076
14 FORMAT (/9X,1HJ,6X,15HCH(J,K),K=1,9) ..... 002077
16 FORMAT (/9X,1HJ,6X,5HTW(J)) ..... 002078
18 FORMAT (5X,4HRR,=,15,5X,3MM,E,15) ..... 002079
END ..... 002080

```

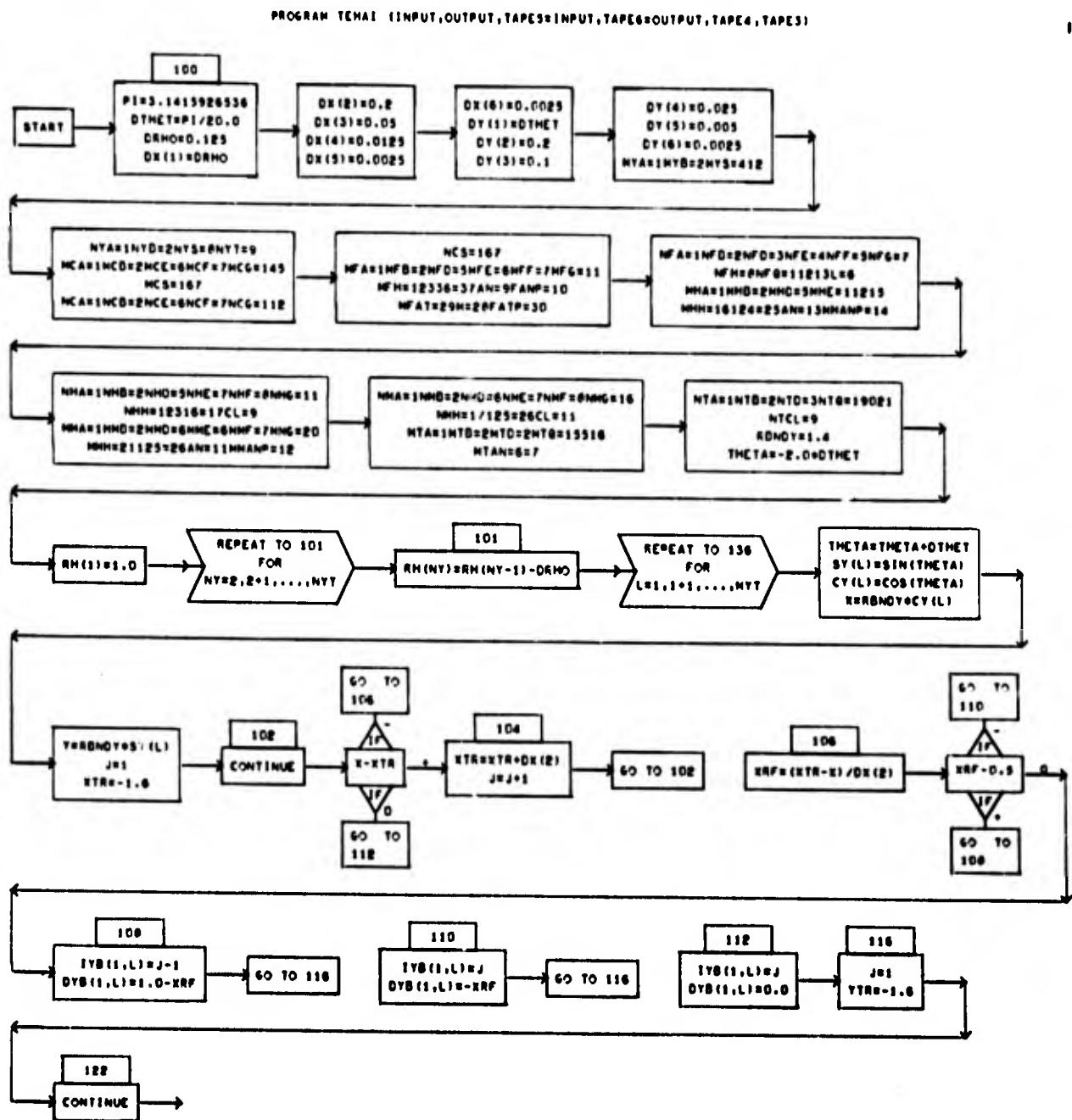
```

SUBROUTINE WAIF(KR,KUT,M,XTZ,YTZ,XYM,SCM,CWMX,IWMX,LBX) 002081
C----SPECIAL AUXILIARY INTERPOLATIONS, CARTESIAN TO SURFACE NORMALS. 002082
COMMON/ZMESH/,DX(6),DY(6),RH(9) 002083
DIMENSION XYM(2,LMX),SCM(2,LMX),CWMX(9,2,2),IWMX(2,2) 002084
9002 LIP=1 002085
XP = XYM(1,M) - DY(KR)*SCM(1,M) 002086
YP = XYM(2,M) + DY(KR)*SCM(2,M) 002087
9003 MK=1 002088
XJ = XTZ 002089
9004 IF(XT - XP) 9006,9008,9008 002090
9006 XJ = XJ + UX(KR) 002091
MK = MK + 1 002092
GO TO 9004 002093
9008 AP = (XP - XJ)/DX(KR) 002094
IF(AP + 0.5) 9010,9010,9012 002095
9010 MK = MK - 1 002096
AP = 1.0 + AP 002097
9012 MK = 1 002098
YI = YIZ 002099
9014 IF(YI - YP) 9016,9018,9018 002100
9016 YI = YI + UY(KR) 002101
IK = NK + 1 002102
GO TO 9014 002103
9018 LP = (YP - YI)/DY(KR) 002104
IF((LIP,EQ,1),AND,(KUT,EQ,1)) GO TO 9022 002105
IF((LIP,EQ,1),AND,(KUT,EQ,2)) GO TO 9020 002106
IF(AP + 0.5) 9020,9020,9022 002107
9020 IK = NK - 1 002108
UP = 1.0 + AP 002109
9022 CALL CARIX(AP,UP,CWMX(1,LIP,KUT)). 002110
IWMX(LIP,KUT) = 100*MK + NK 002111
GO TO 19024,9026),LIP. 002112
9024 LIP=2 002113
XP = XP - UY(KR)+SCM(1,M) 002114
YP = YP + UY(KR)+SCM(2,M) 002115
GO TO 9003 002116
9026 CONTINUE 002117
RETURN 002118
END 002119

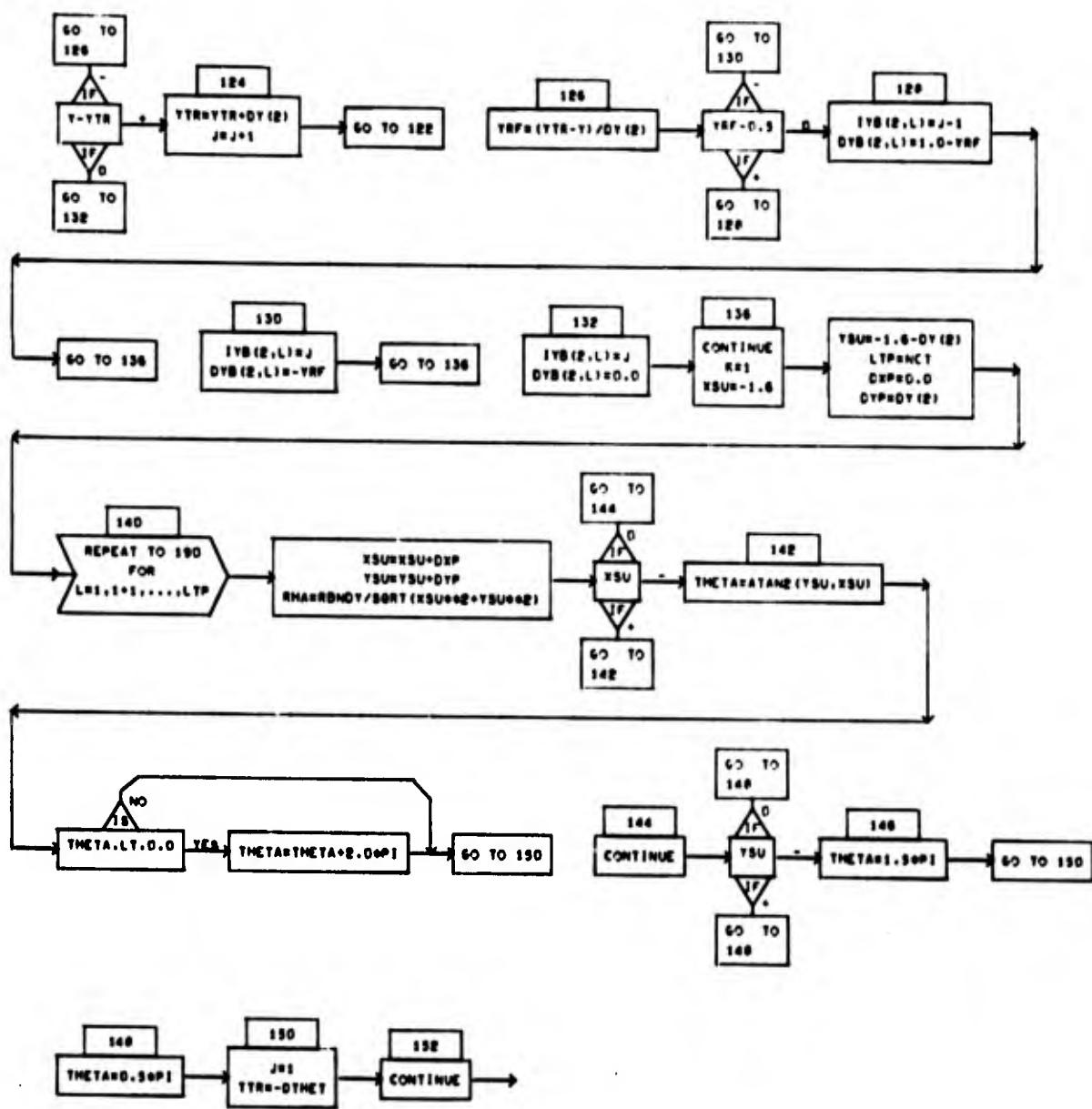
```

```
      SUBROUTINE CARTWIAP,HP,TW)          002120
C-----CALCULATES WEIGHTS TO BE USED IN INTERPOLATION FROM CARTESIAN MESH 002121
      DIMENSION TW(9)                   002122
      TW(1) = 0.25*AP*BP               002123
      TW(3) = -TW(1)                  002124
      TW(7) = TW(3)                   002125
      TW(9) = TW(1)                  002126
      TW(2) = 0.5*HP*1BP - 1.01       002127
      TW(8) = TW(2) + BP              002128
      TW(4) = 0.5*AP*(AP - 1.0)       002129
      TW(6) = TW(4) + AP              002130
      TW(5) = 1.0 - AP**2 - BP**2    002131
      RETURN                           002132
      END                             002133
```

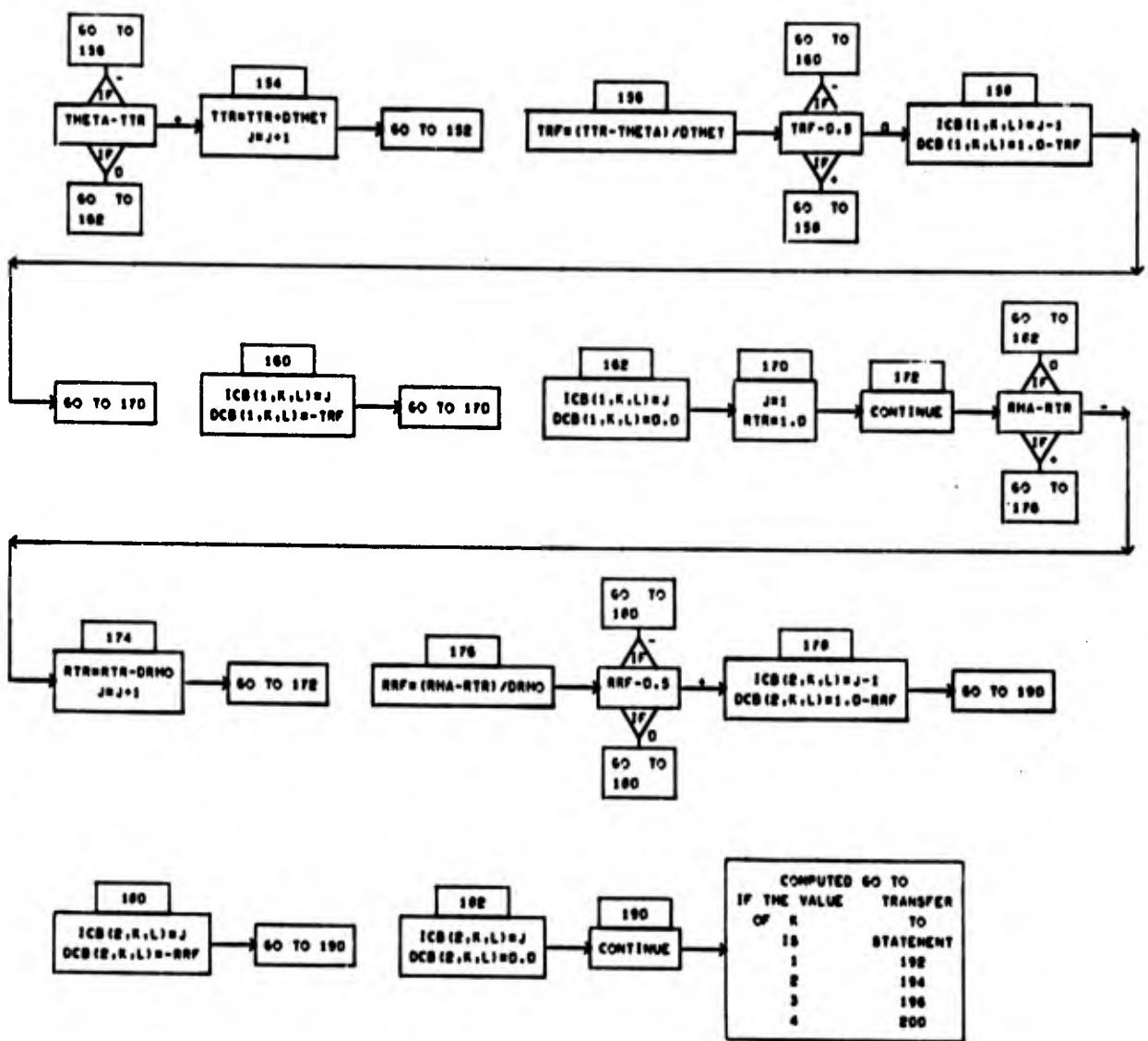
APPENDIX III
SETUP PROGRAM FLOW CHARTS



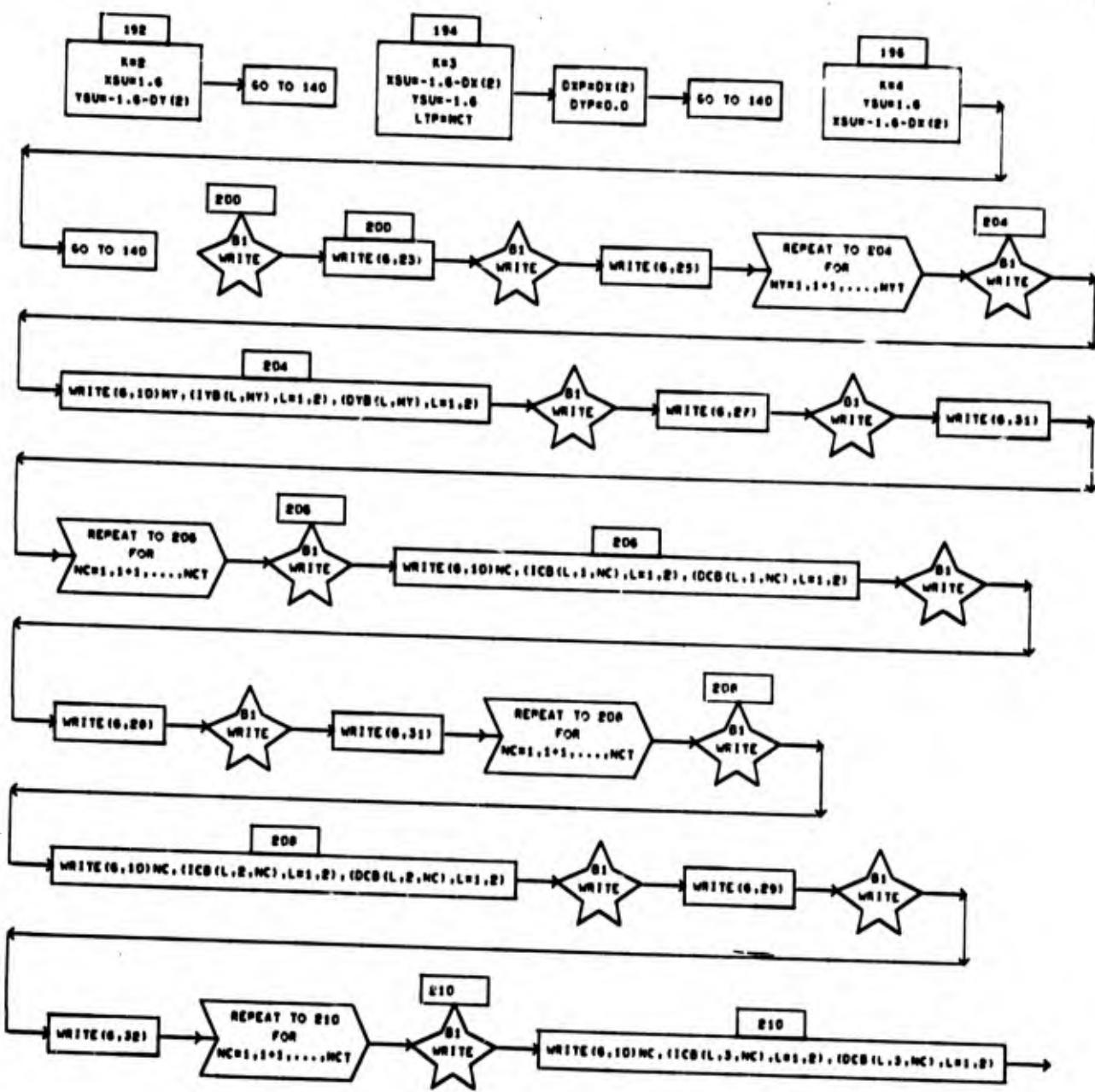
Detailed Flow Charts of Subroutine TEHAI (1 of 37)



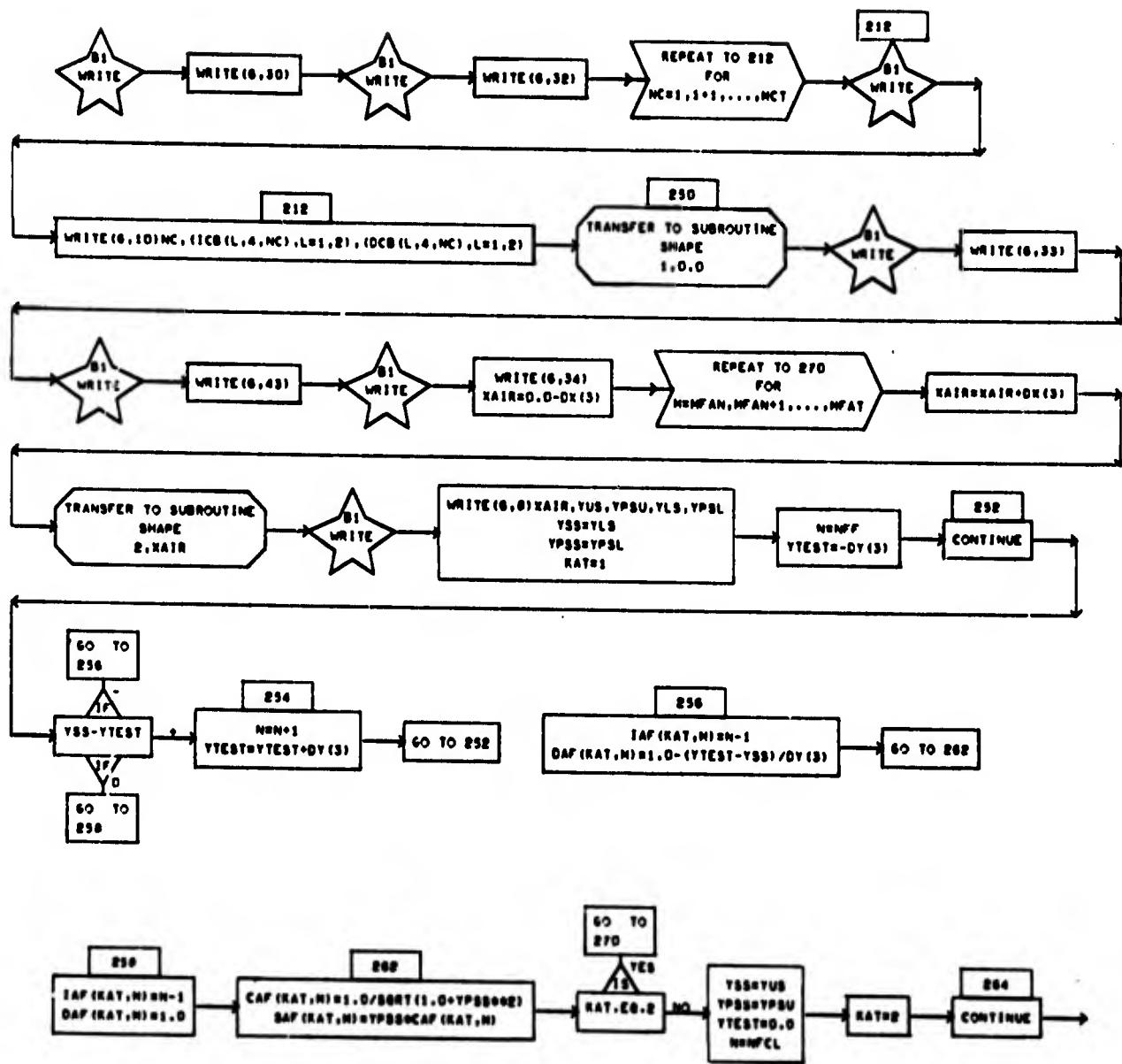
Detailed Flow Charts of Subroutine TEHAI (2 of 37)



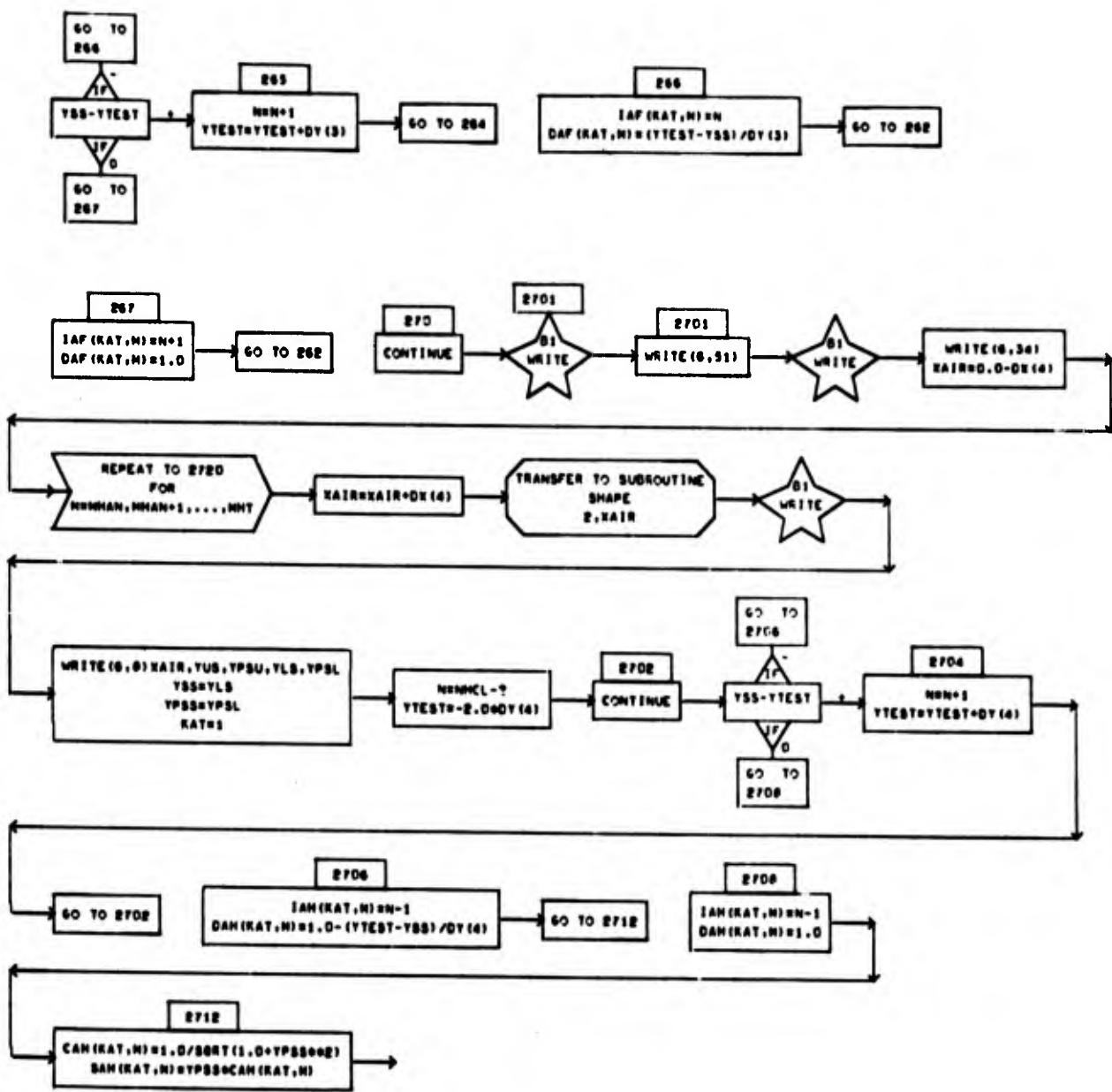
Detailed Flow Charts of Subroutine TEHAI (3 of 37)



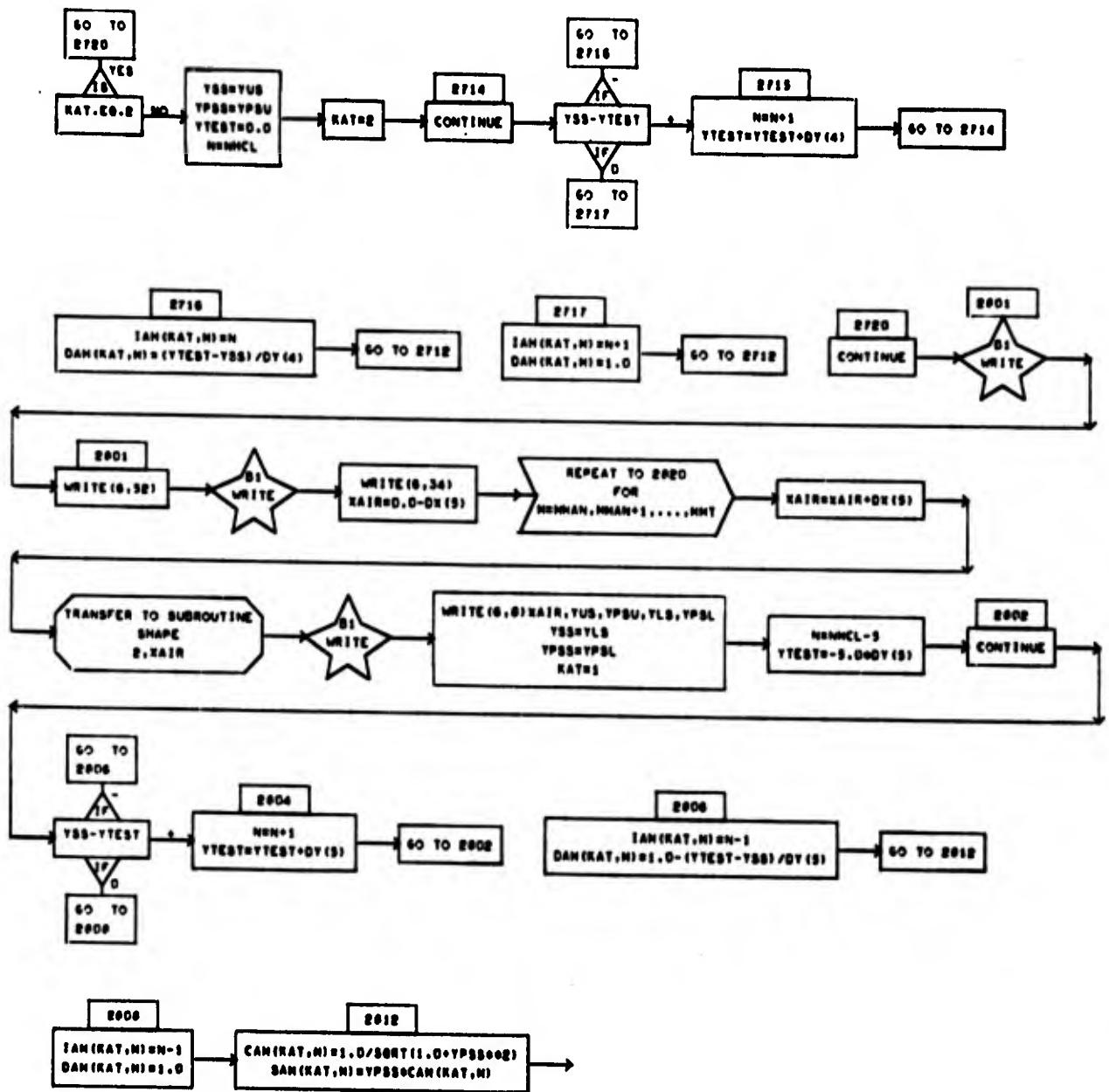
Detailed Flow Charts of Subroutine TEHAI (4 of 37)



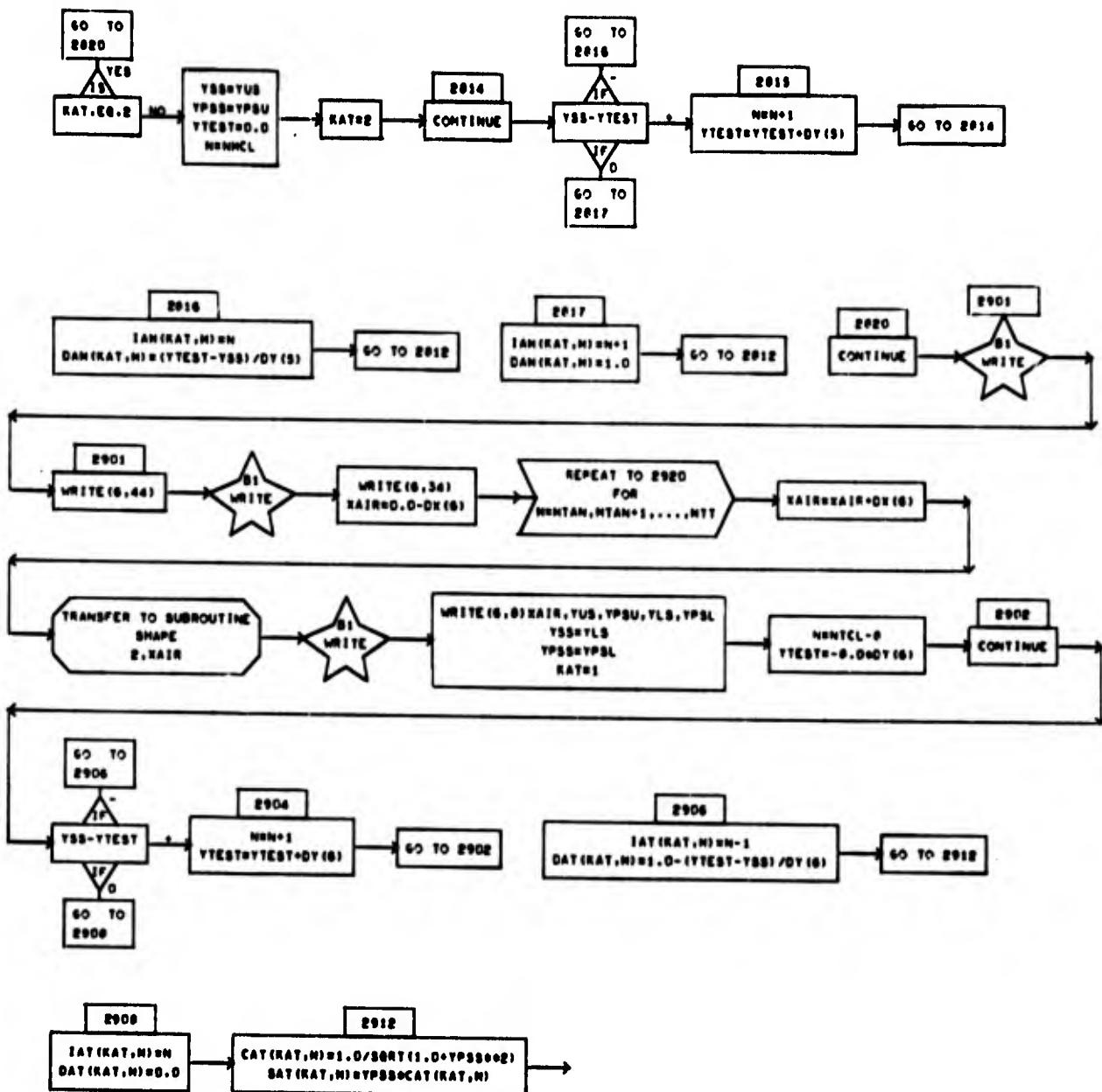
Detailed Flow Charts of Subroutine TEHAI (5 of 37)



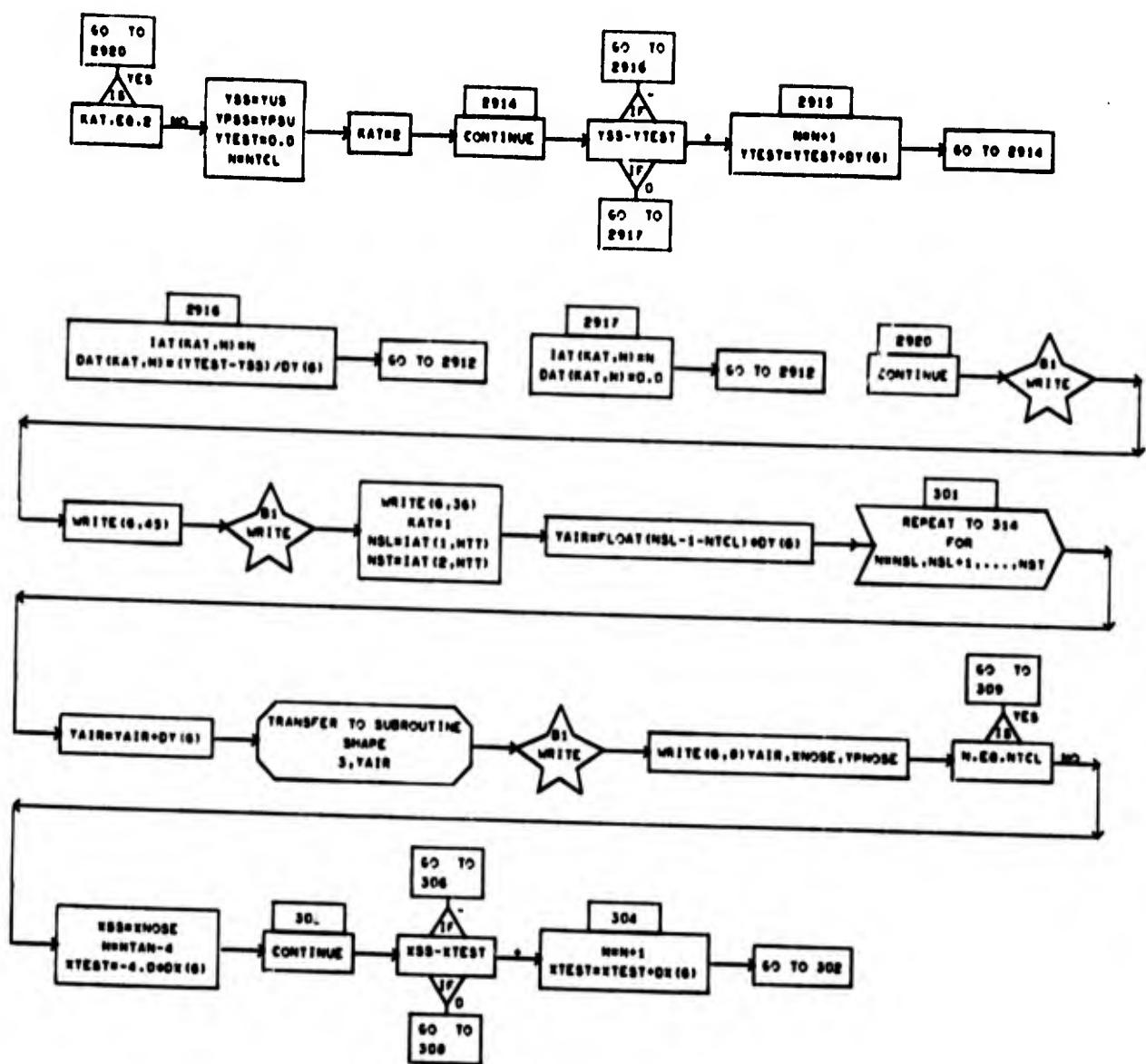
Detailed Flow Charts of Subroutine TEHAI (6 of 37)



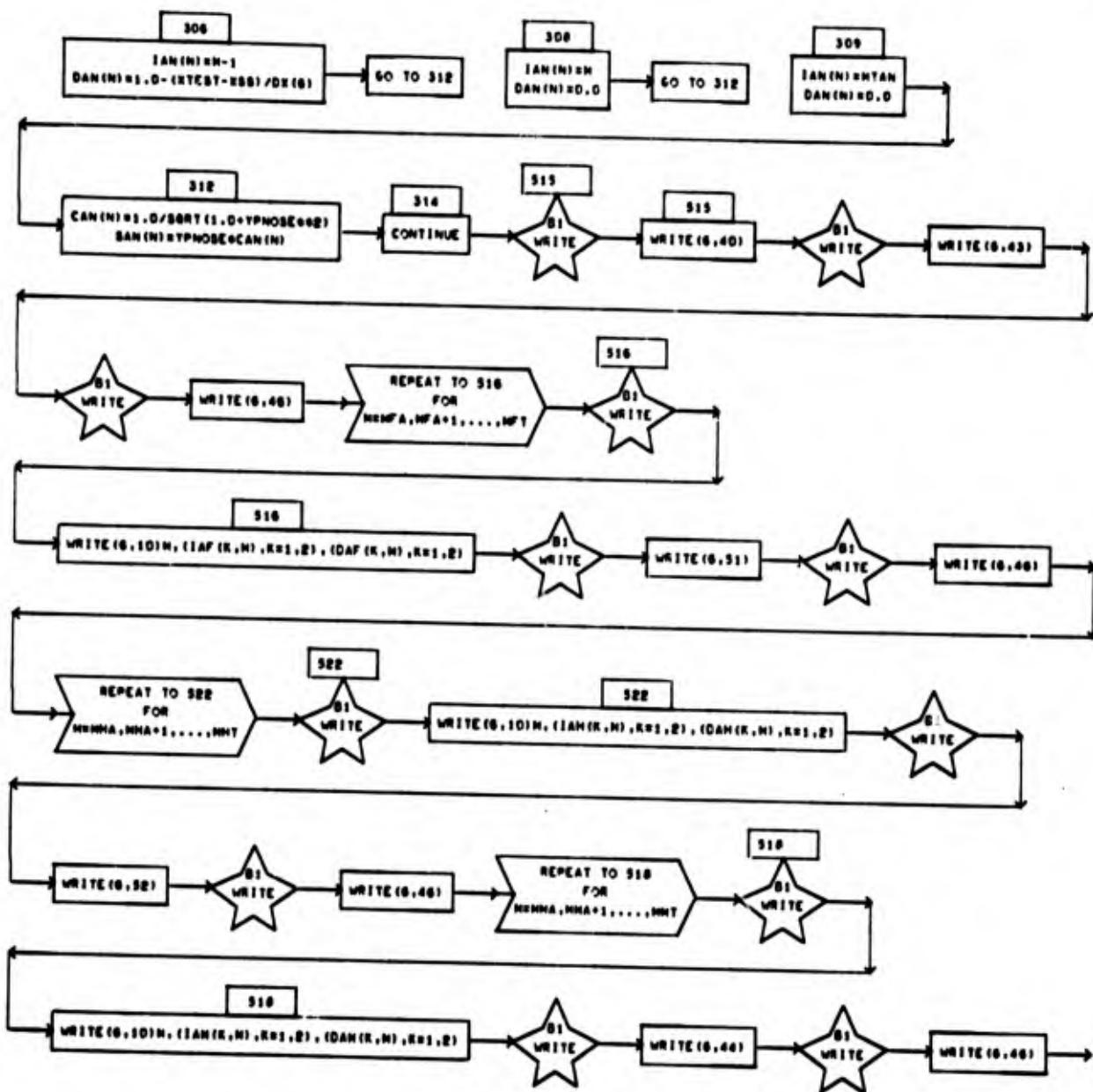
Detailed Flow Charts of Subroutine TEHAI (7 of 37)



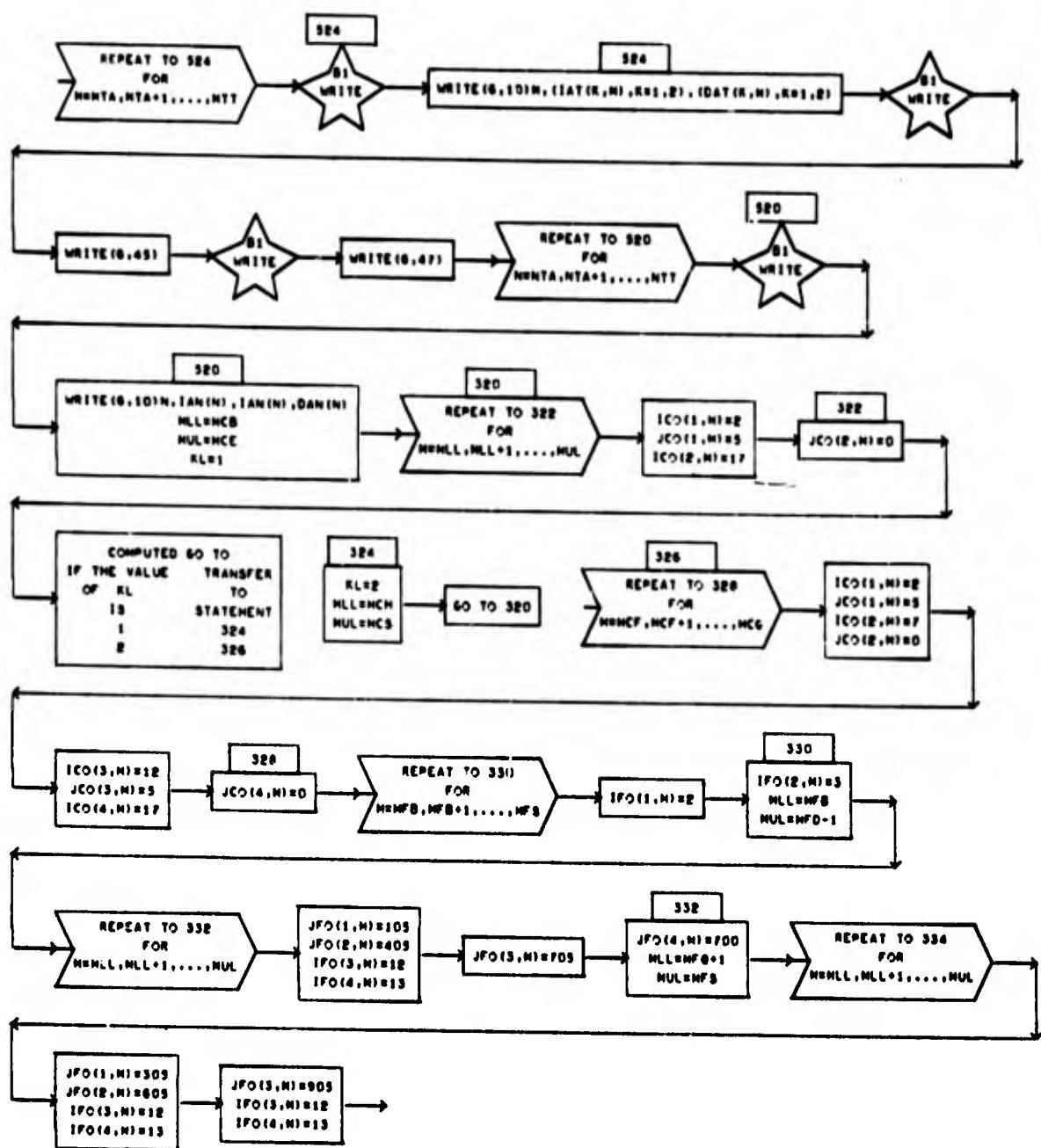
Detailed Flow Charts of Subroutine TEHAI (8 of 37)



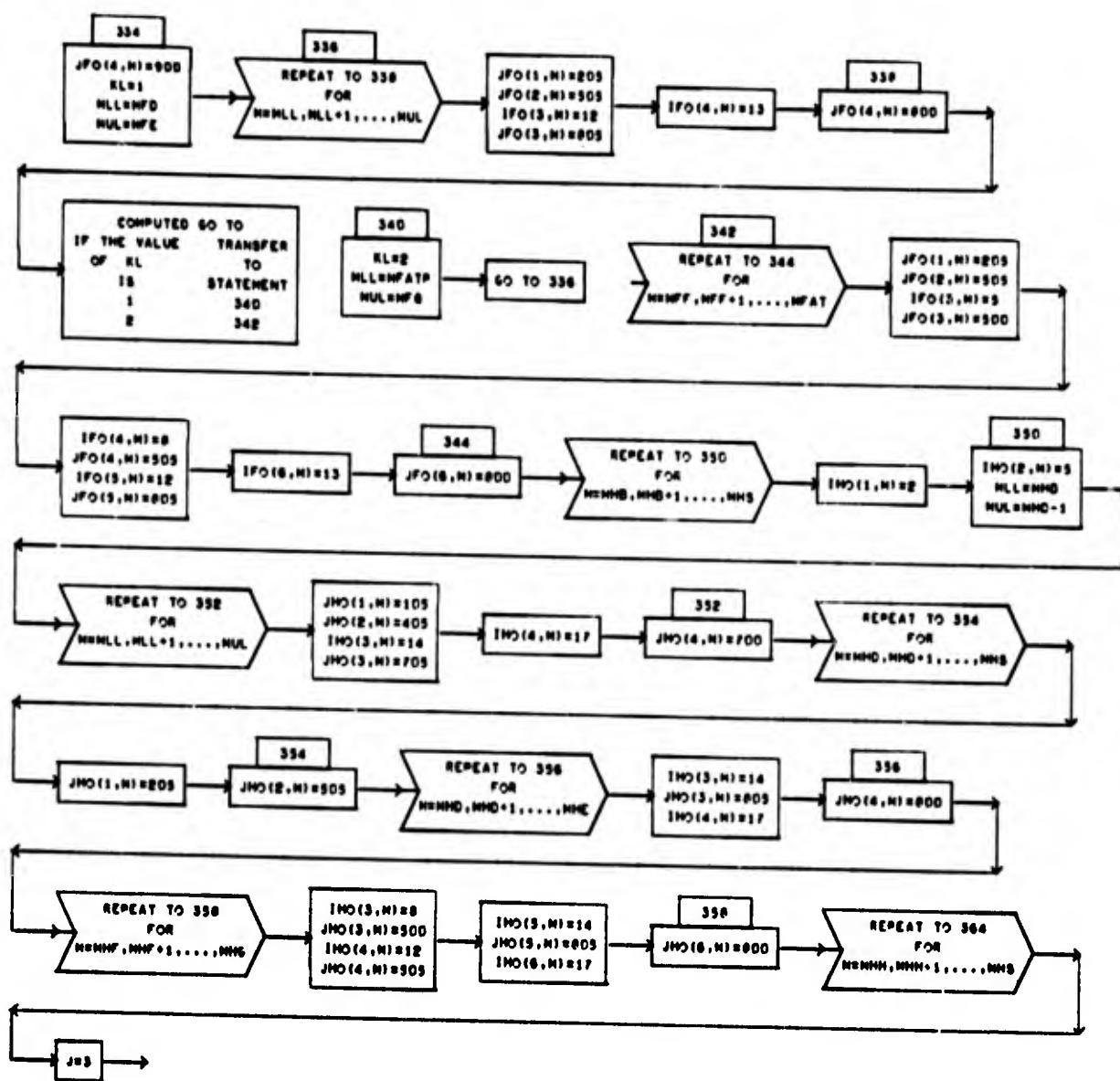
Detailed Flow Charts of Subroutine TEHAI (9 of 37)



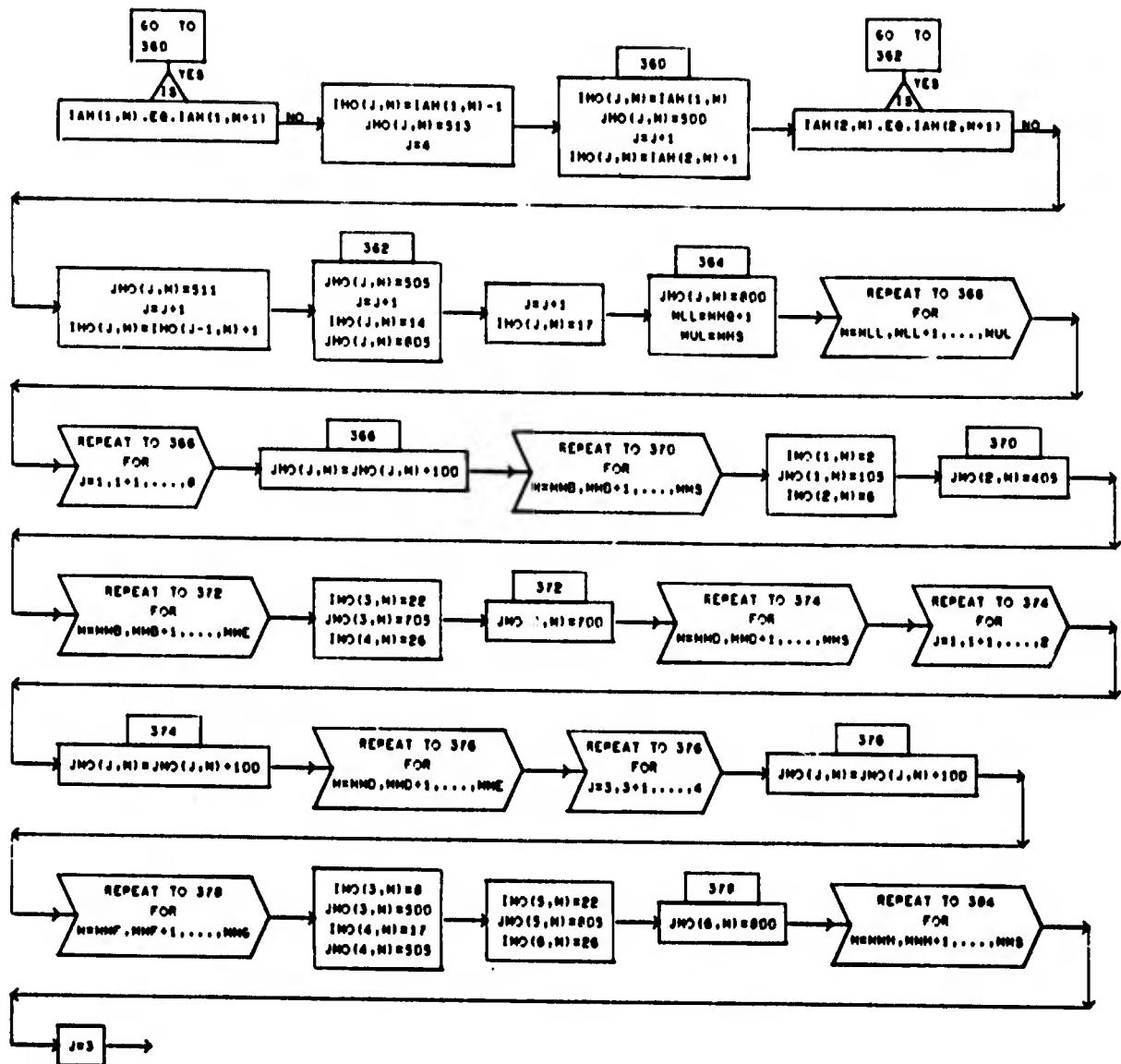
Detailed Flow Charts of Subroutine TEHAI (10 of 37)



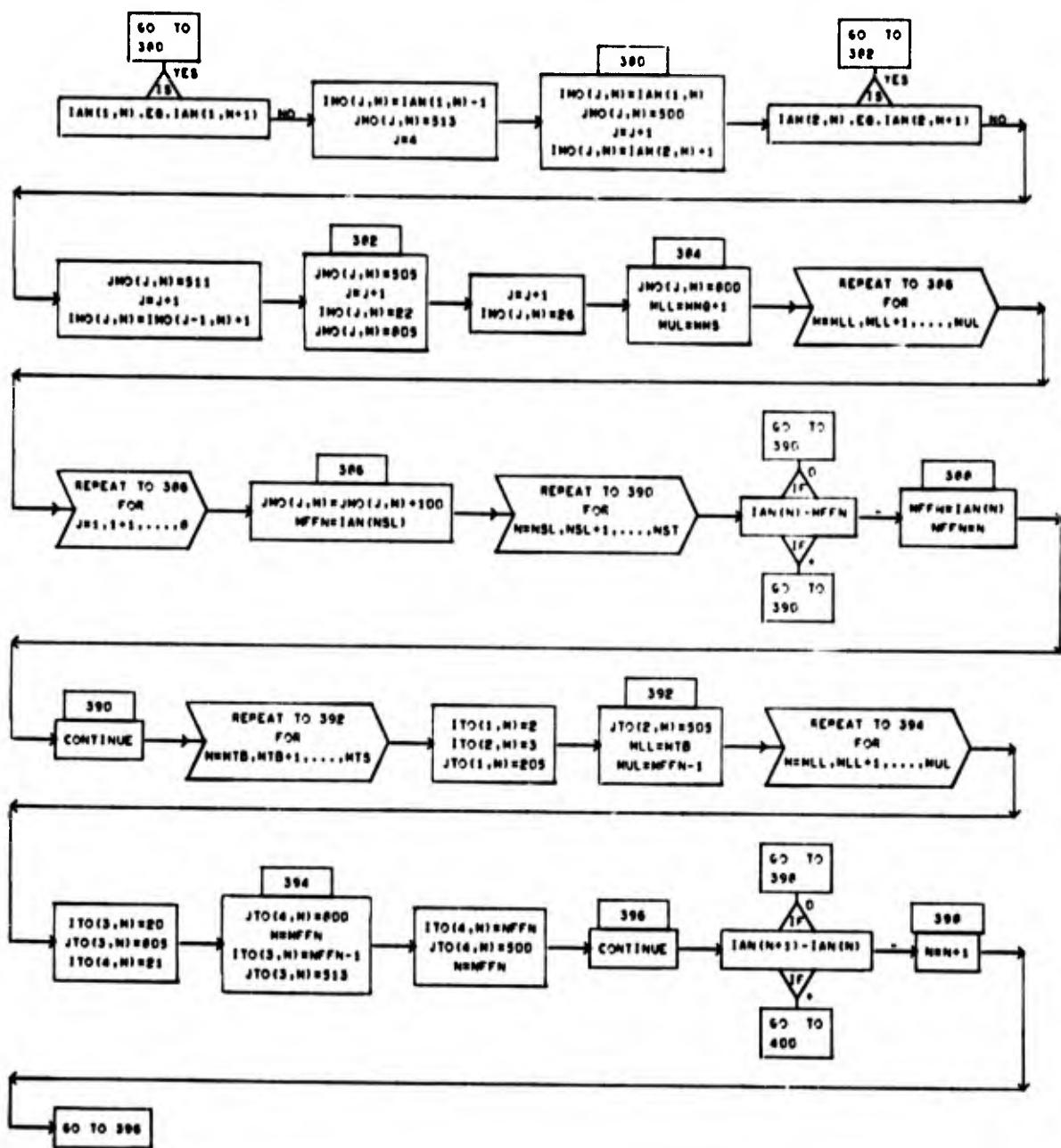
Detailed Flow Charts of Subroutine TEHAI (11 of 37)



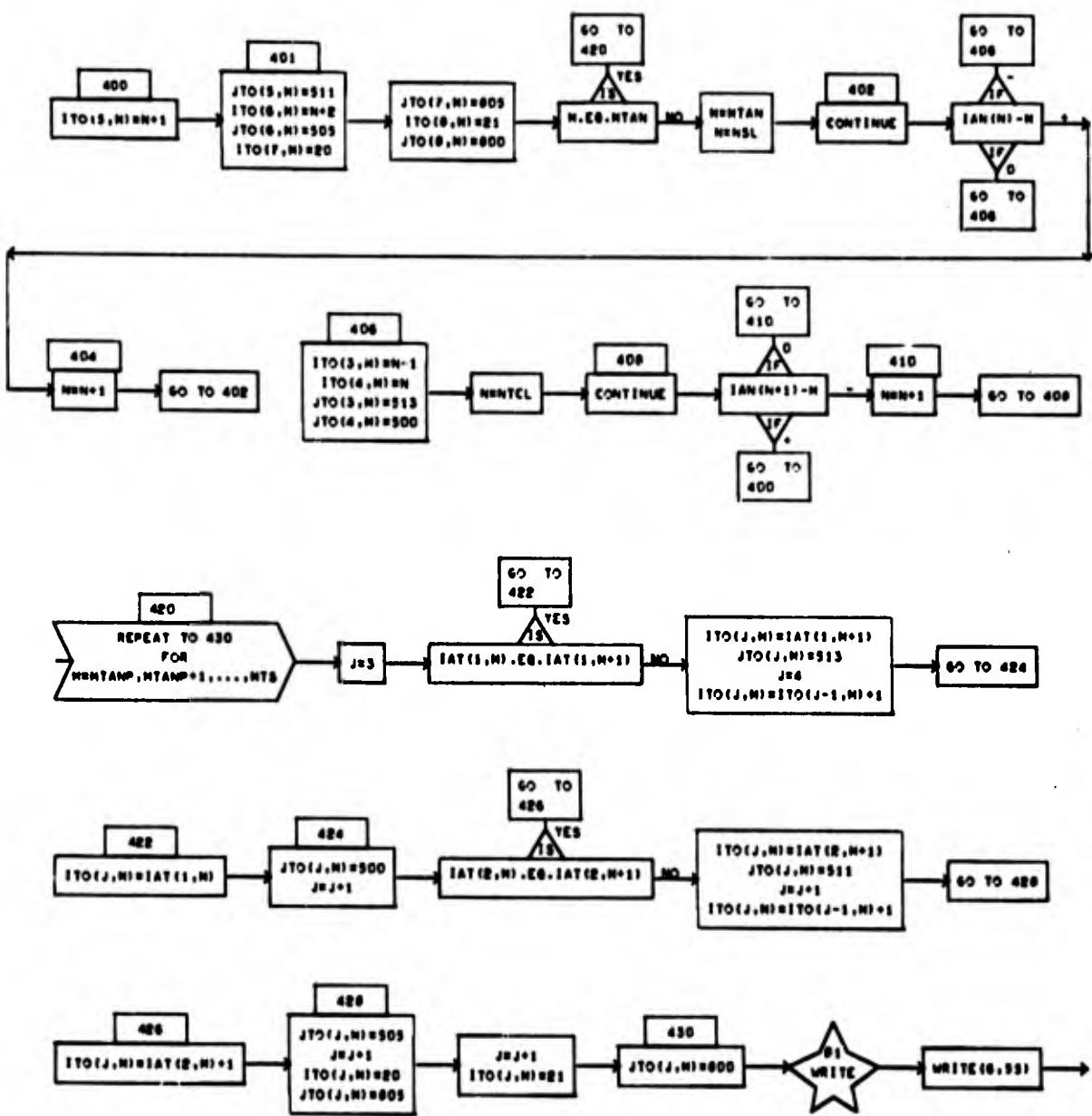
Detailed Flow Charts of Subroutine TEHAI (12 of 37)



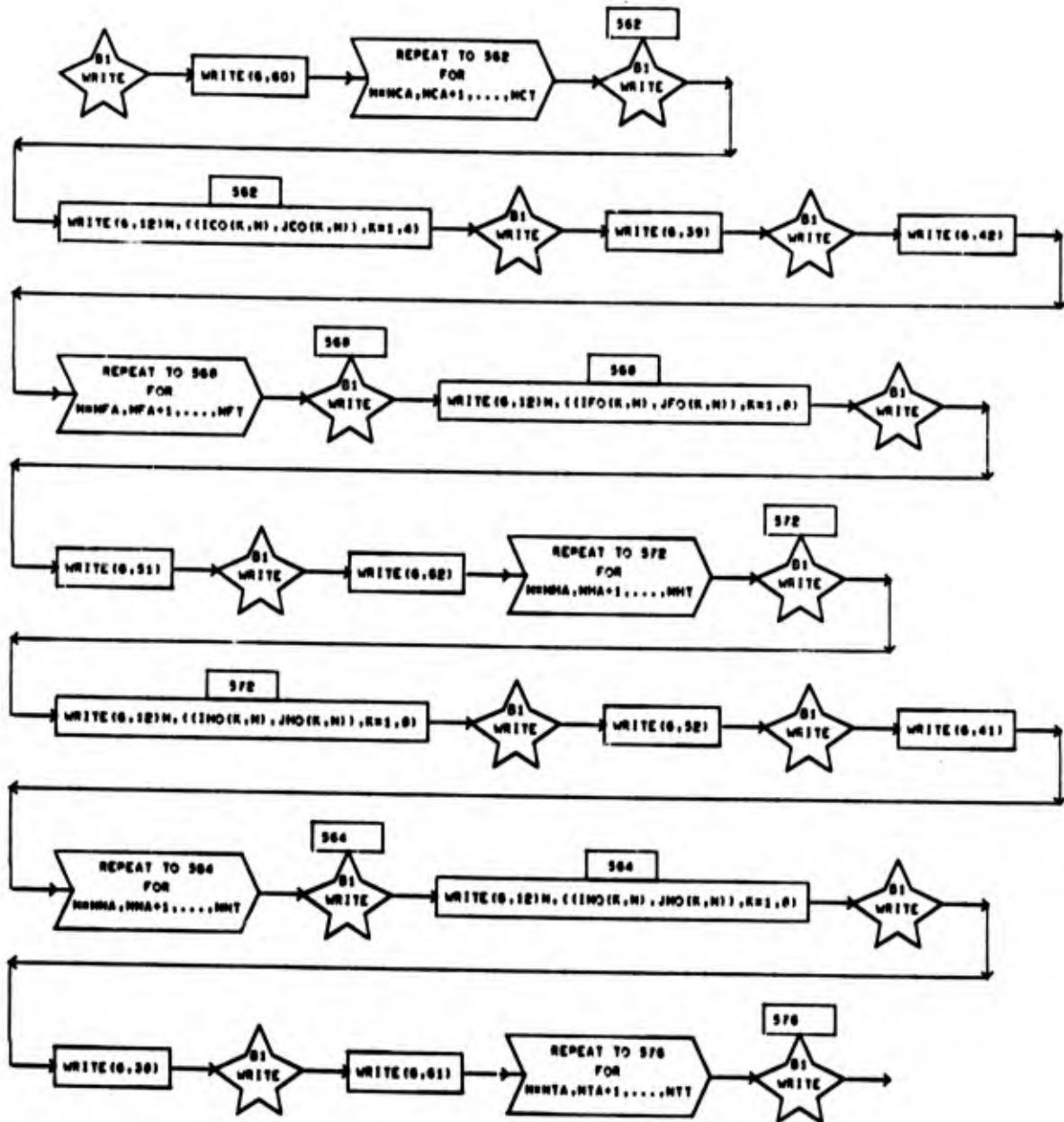
Detailed Flow Charts of Subroutine TEHAI (13 of 37)



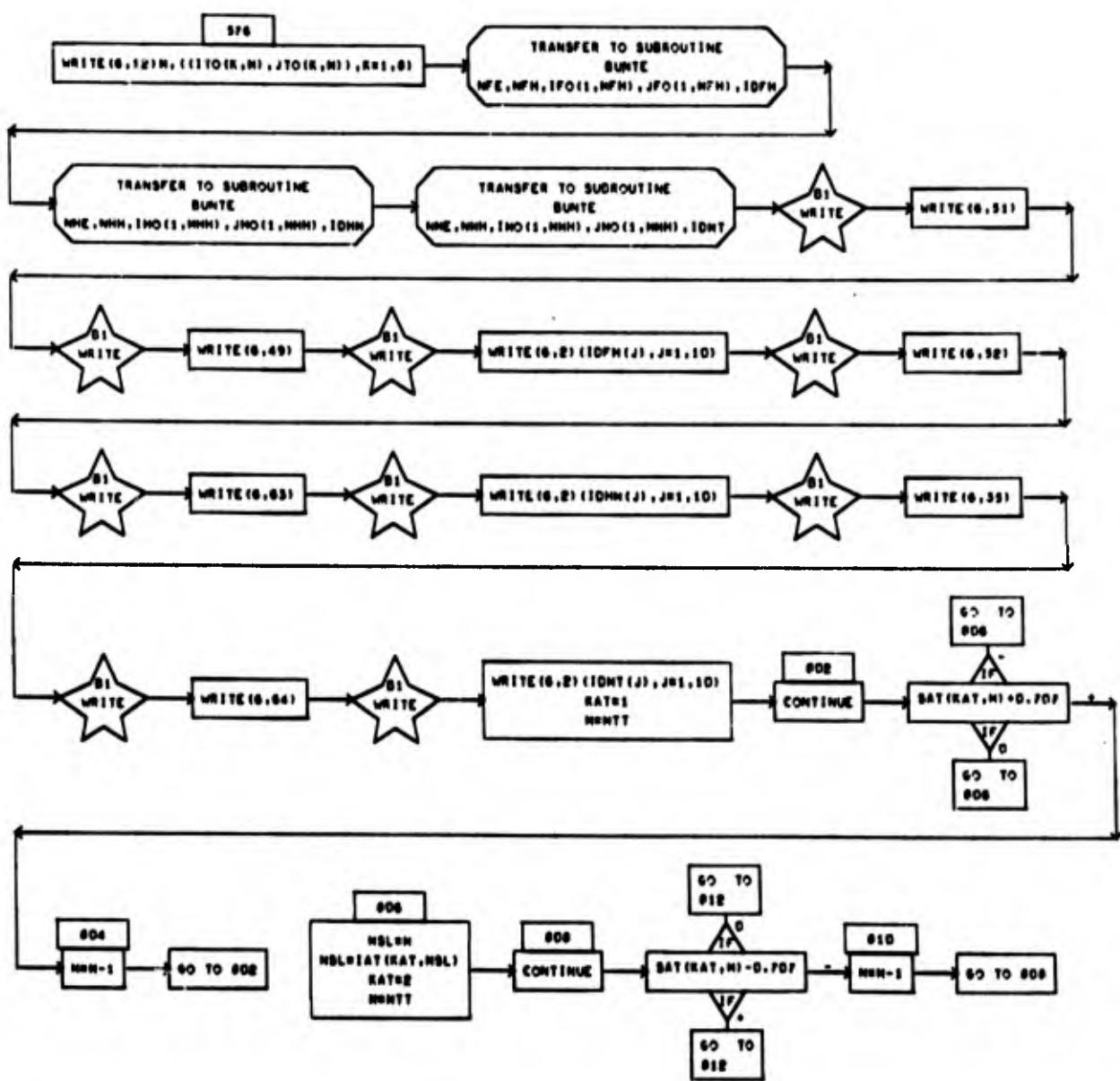
Detailed Flow Charts of Subroutine TEHAI (14 of 37)



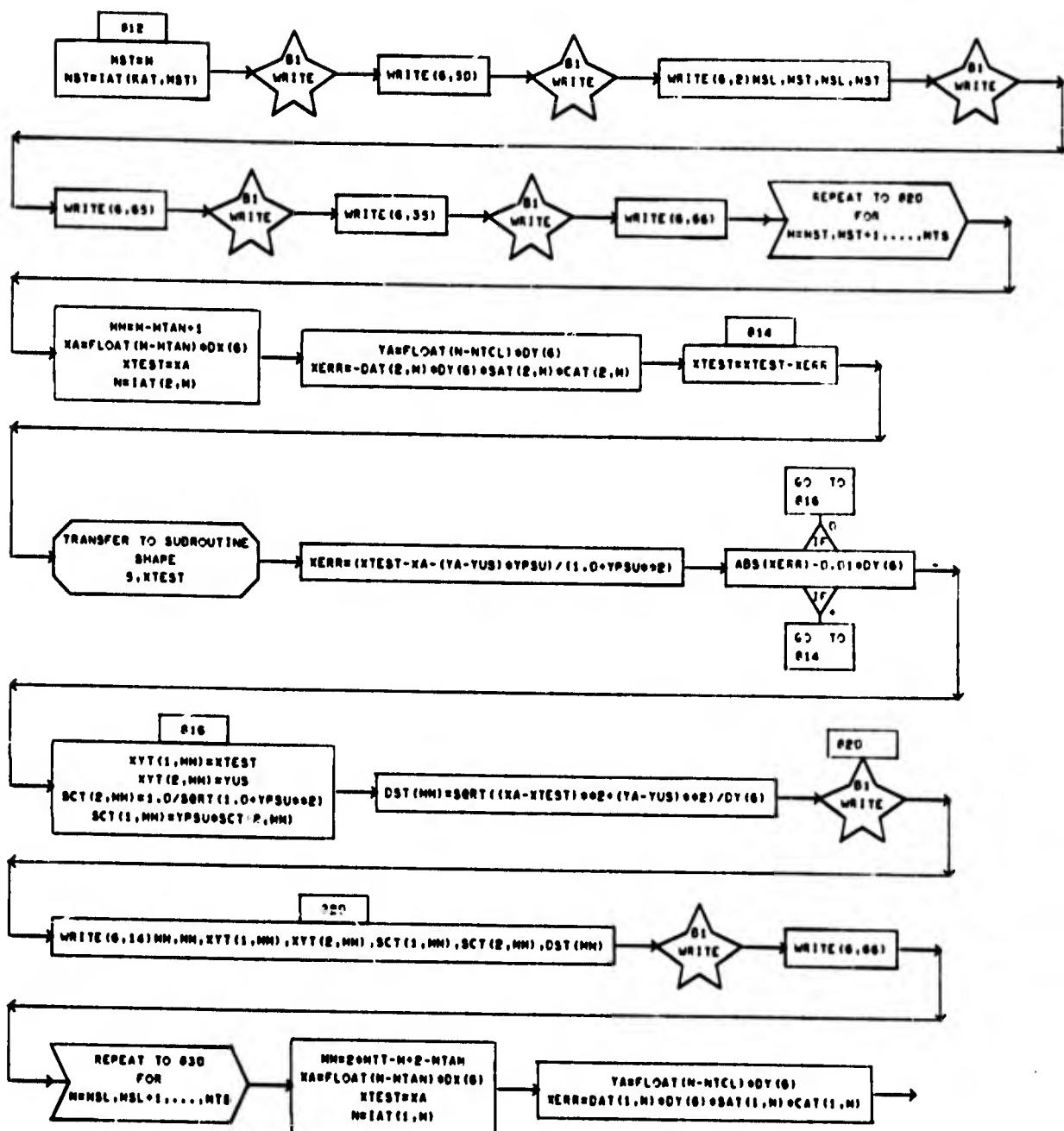
Detailed Flow Charts of Subroutine TEHAI (15 of 37)



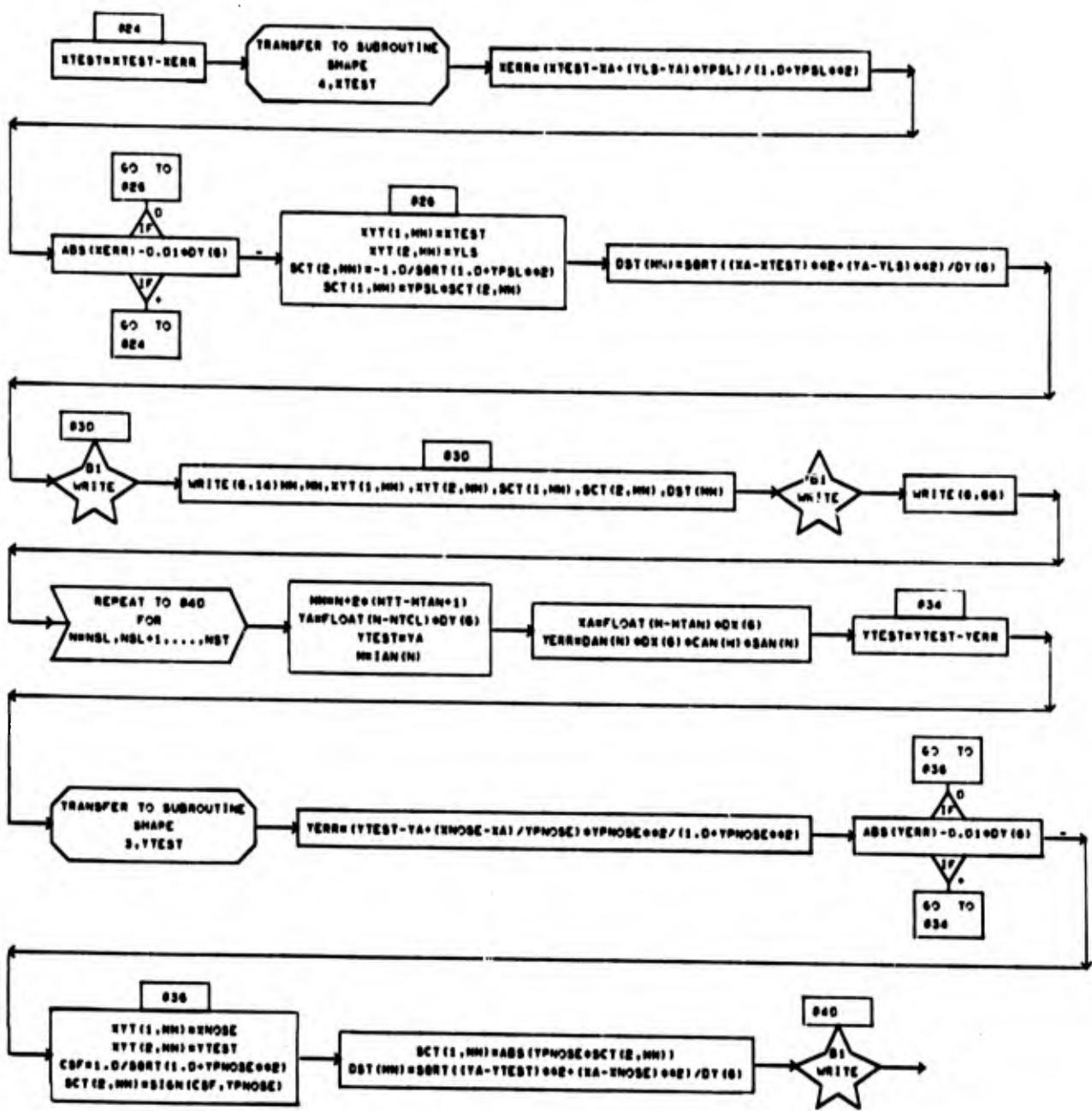
Detailed Flow Charts of Subroutine TEHAI (16 of 37)



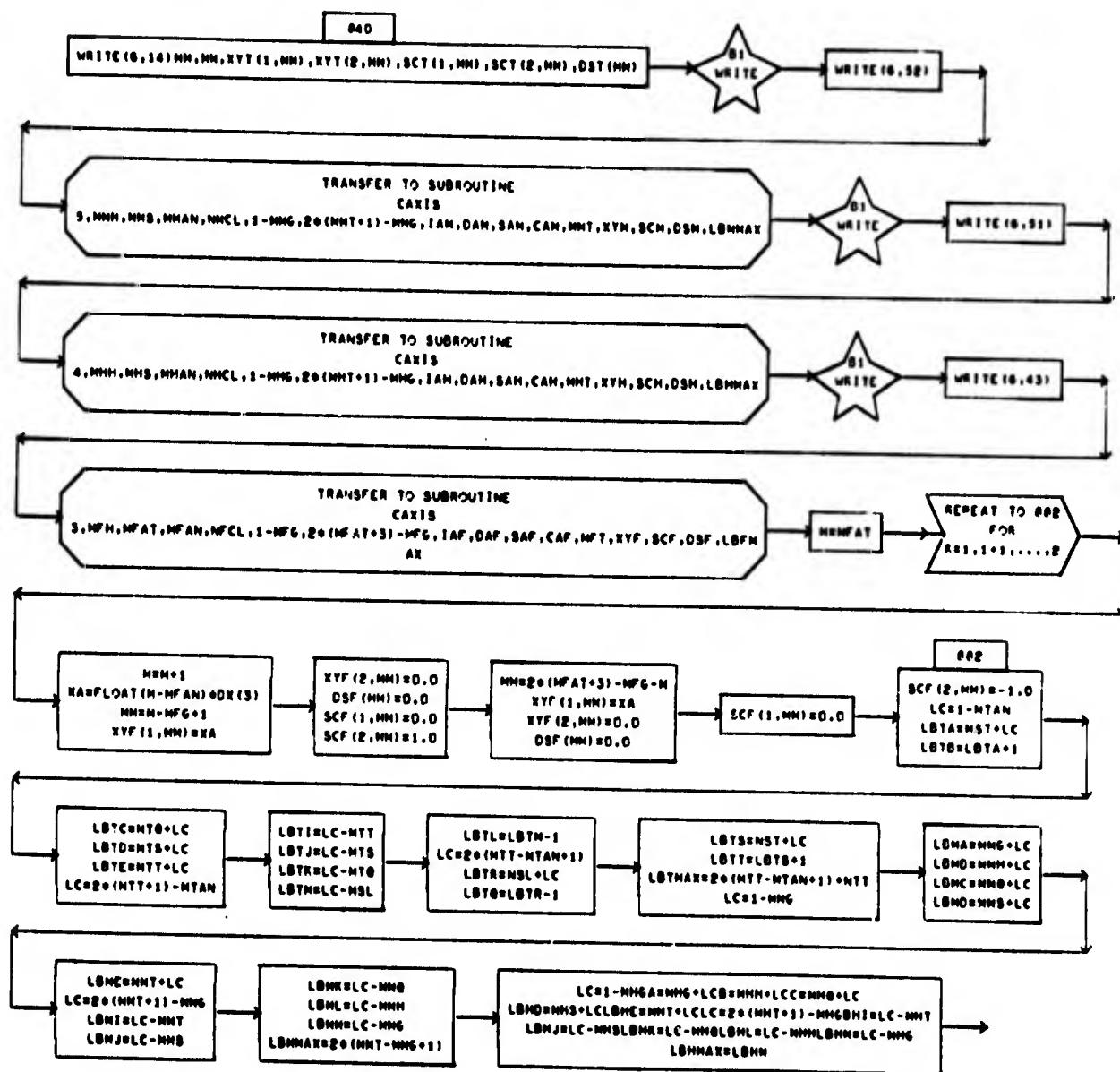
Detailed Flow Charts of Subroutine TEHAI (17 of 37)



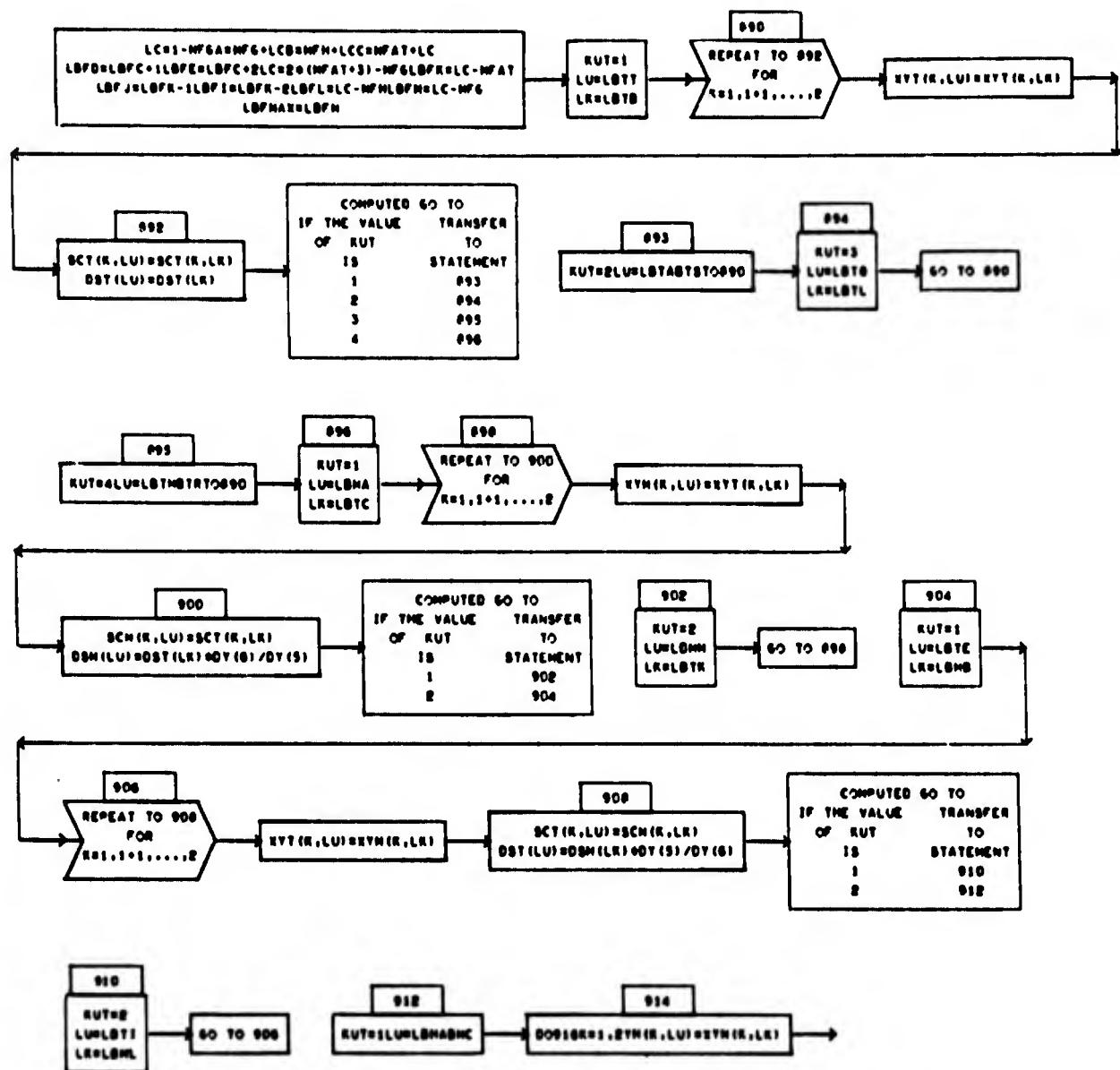
Detailed Flow Charts of Subroutine TEHAI (18 of 37)



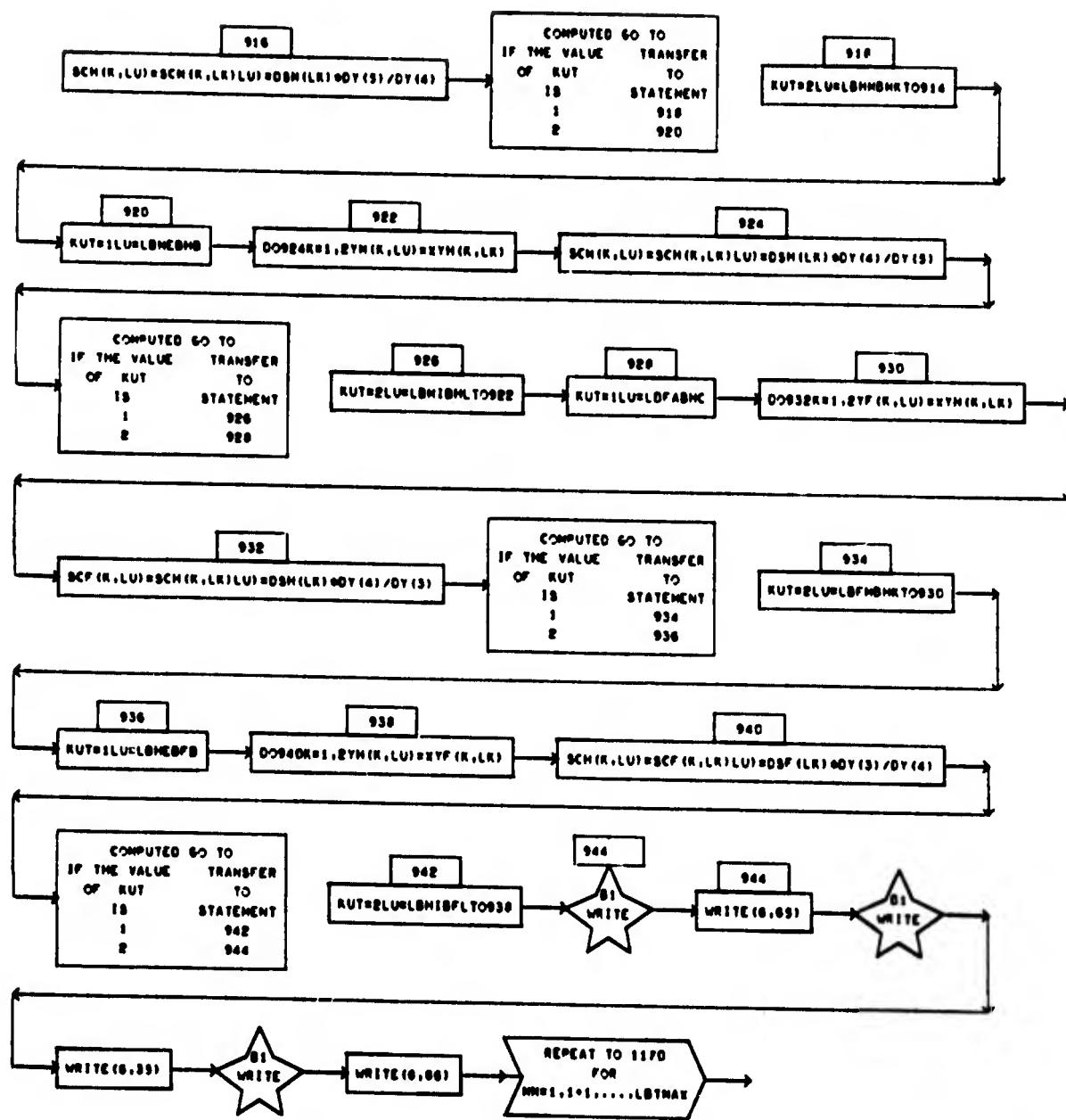
Detailed Flow Charts of Subroutine TEHAI (19 of 37)



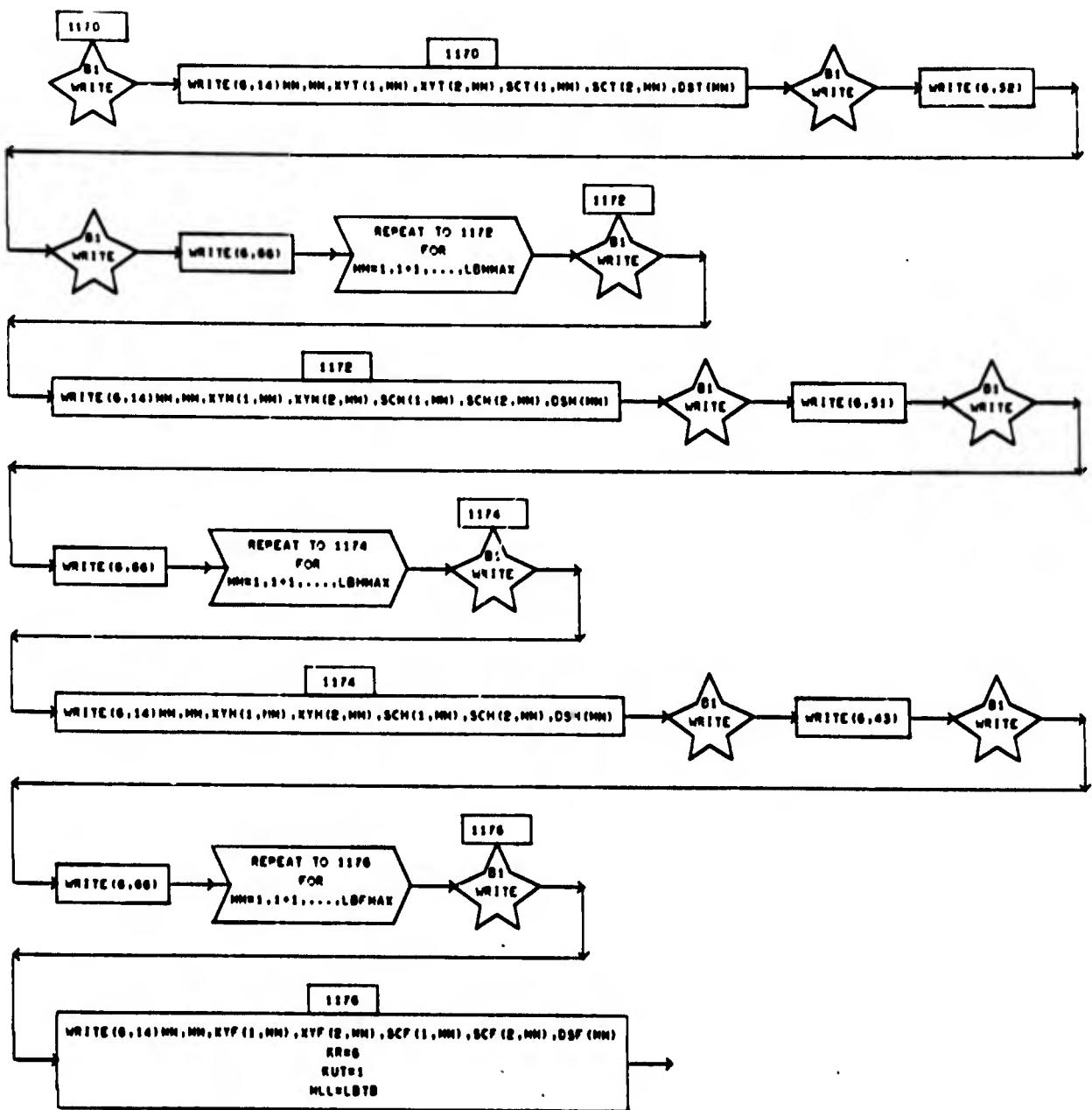
Detailed Flow Charts of Subroutine TEHAI (20 of 37)



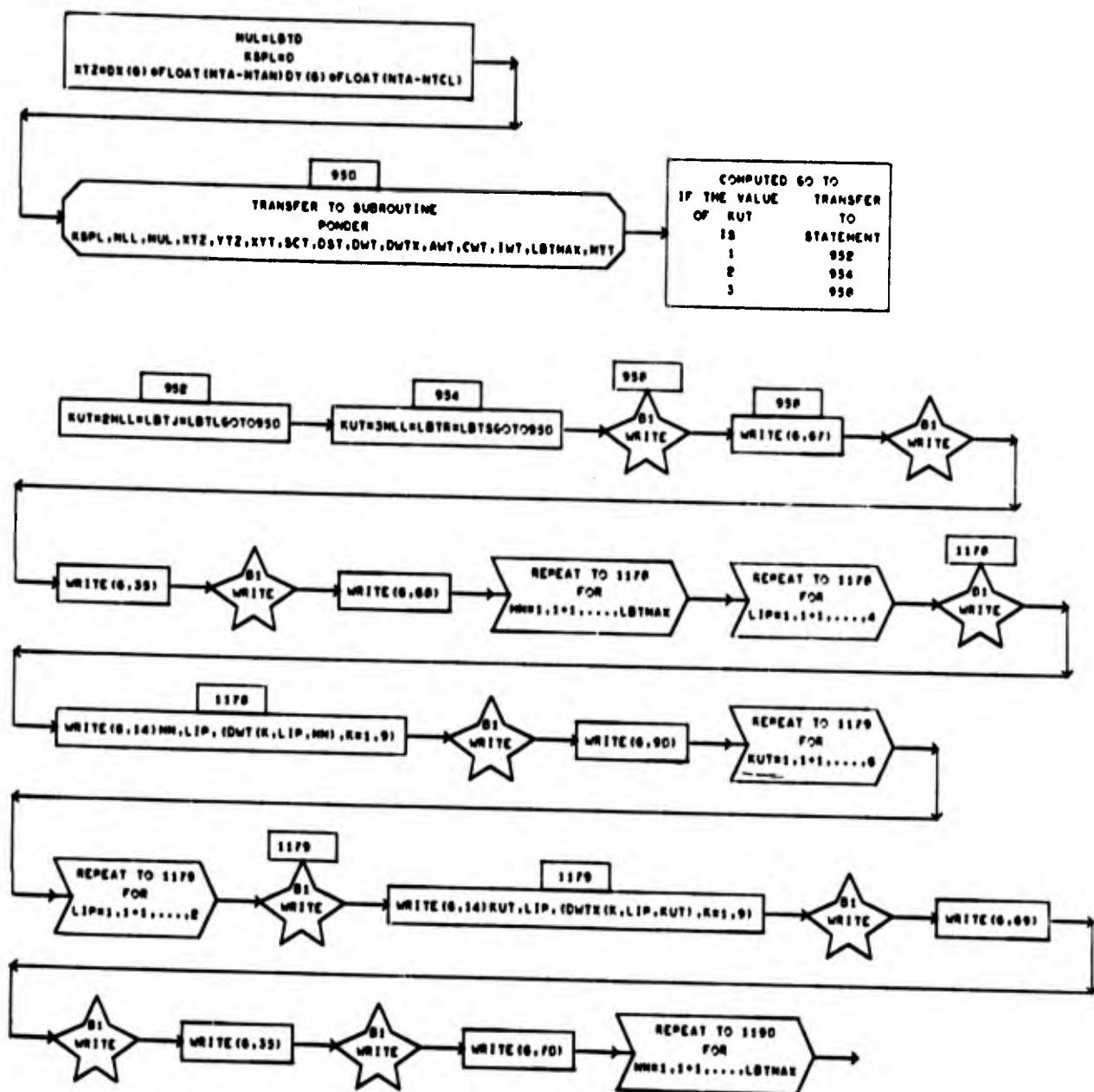
Detailed Flow Charts of Subroutine TEHAI (21 of 37)



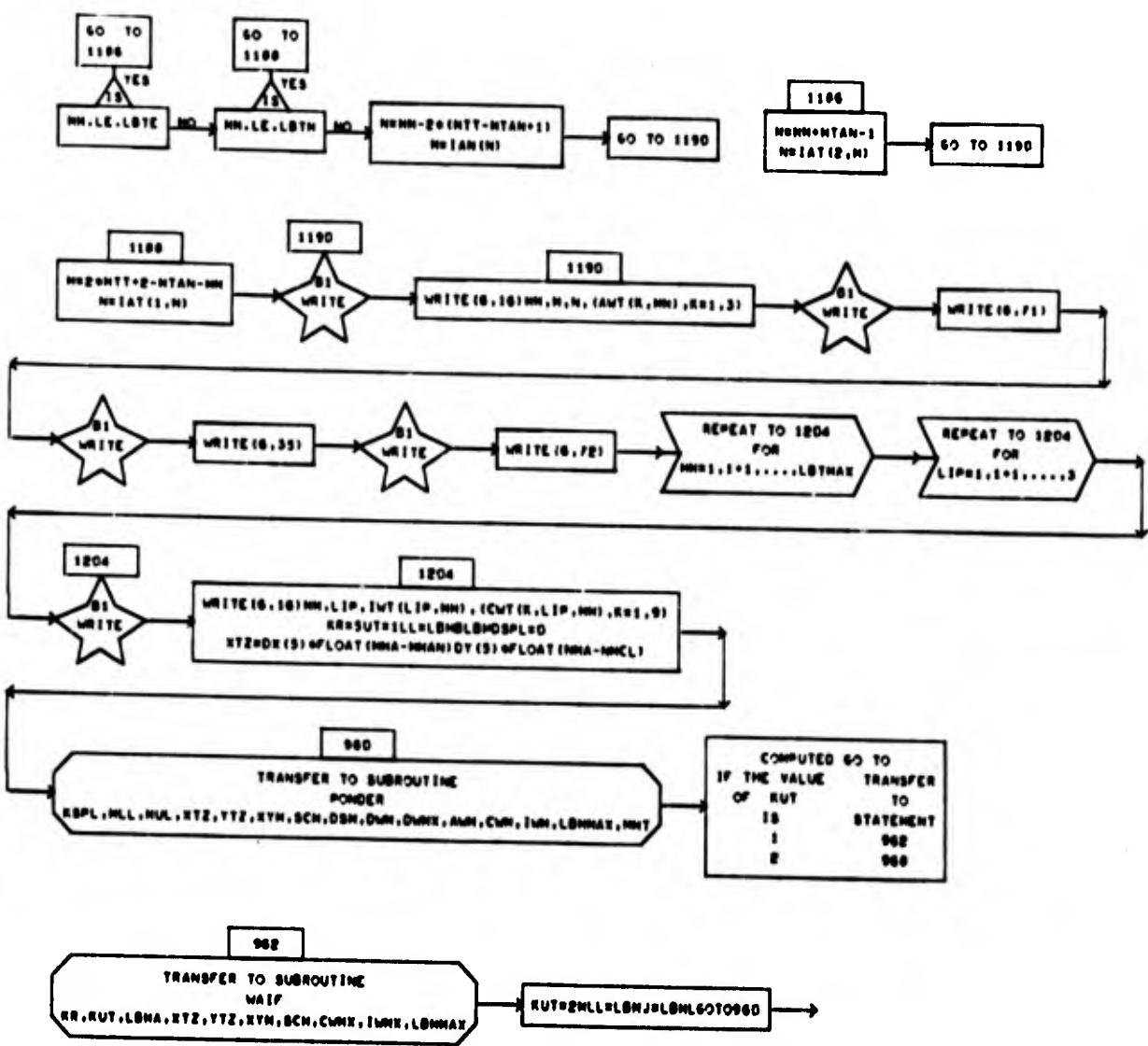
Detailed Flow Charts of Subroutine TEHAI (22 of 37)



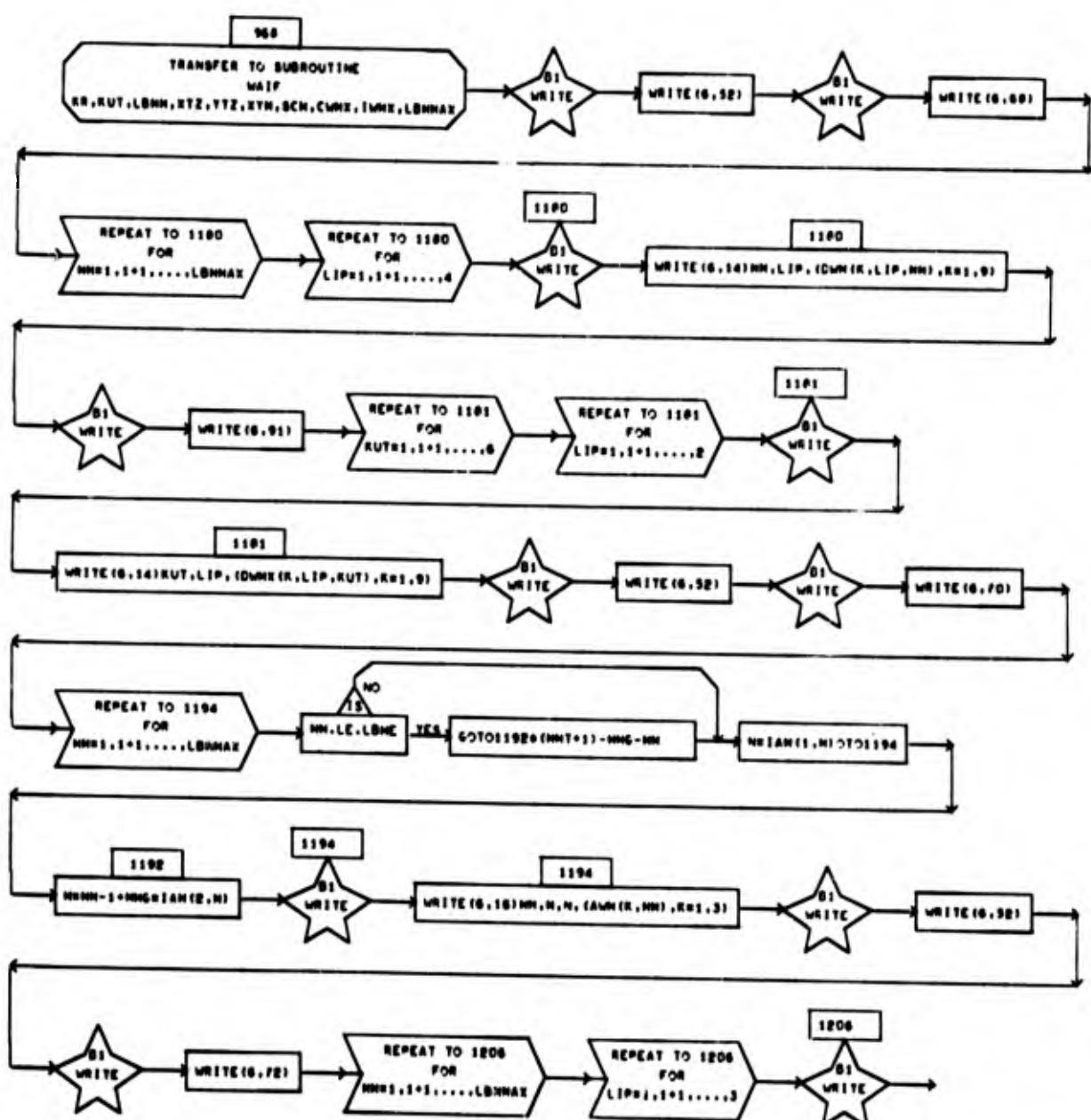
Detailed Flow Charts of Subroutine TEHAI (23 of 37)



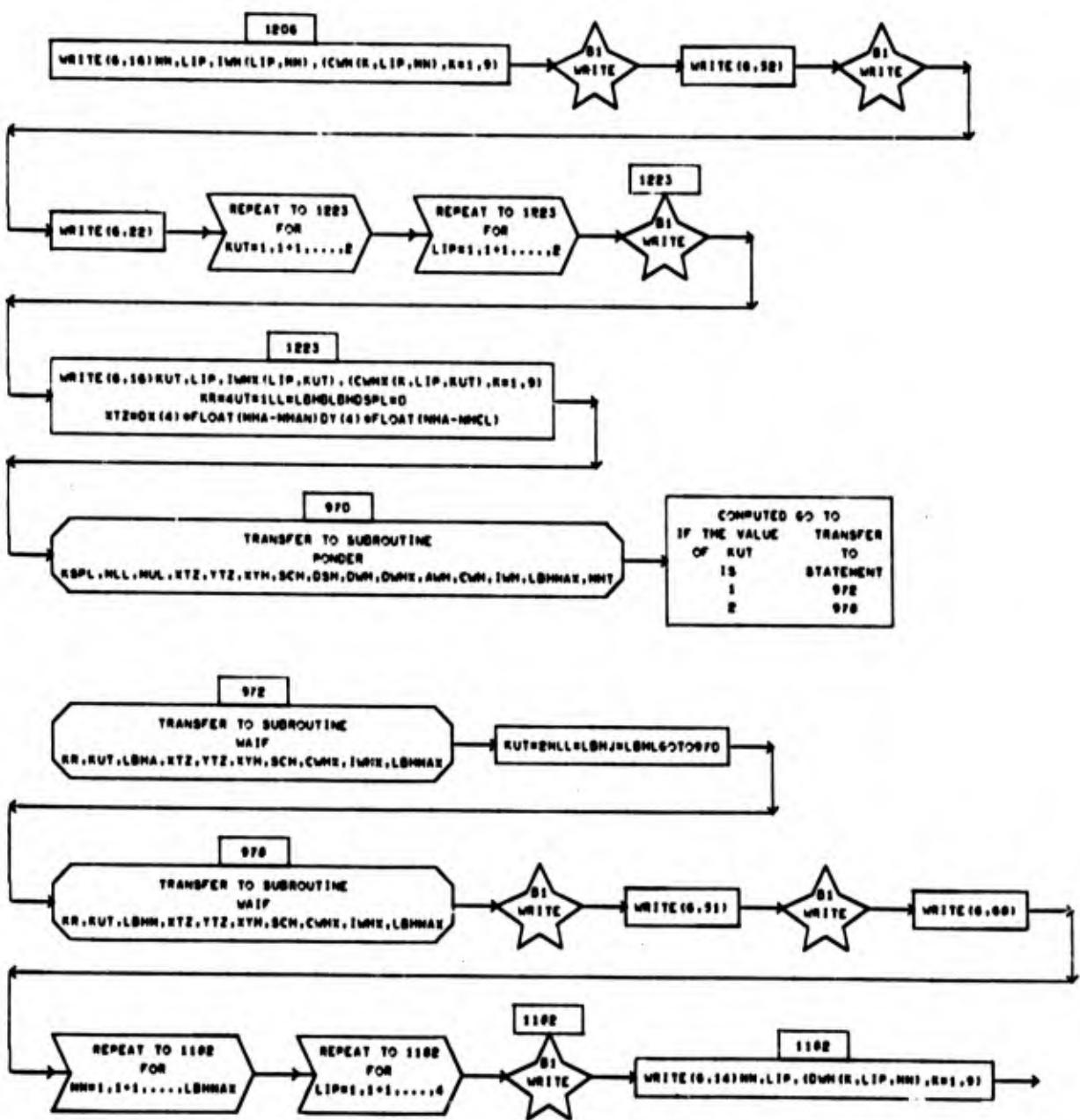
Detailed Flow Charts of Subroutine TEHAI (24 of 37)



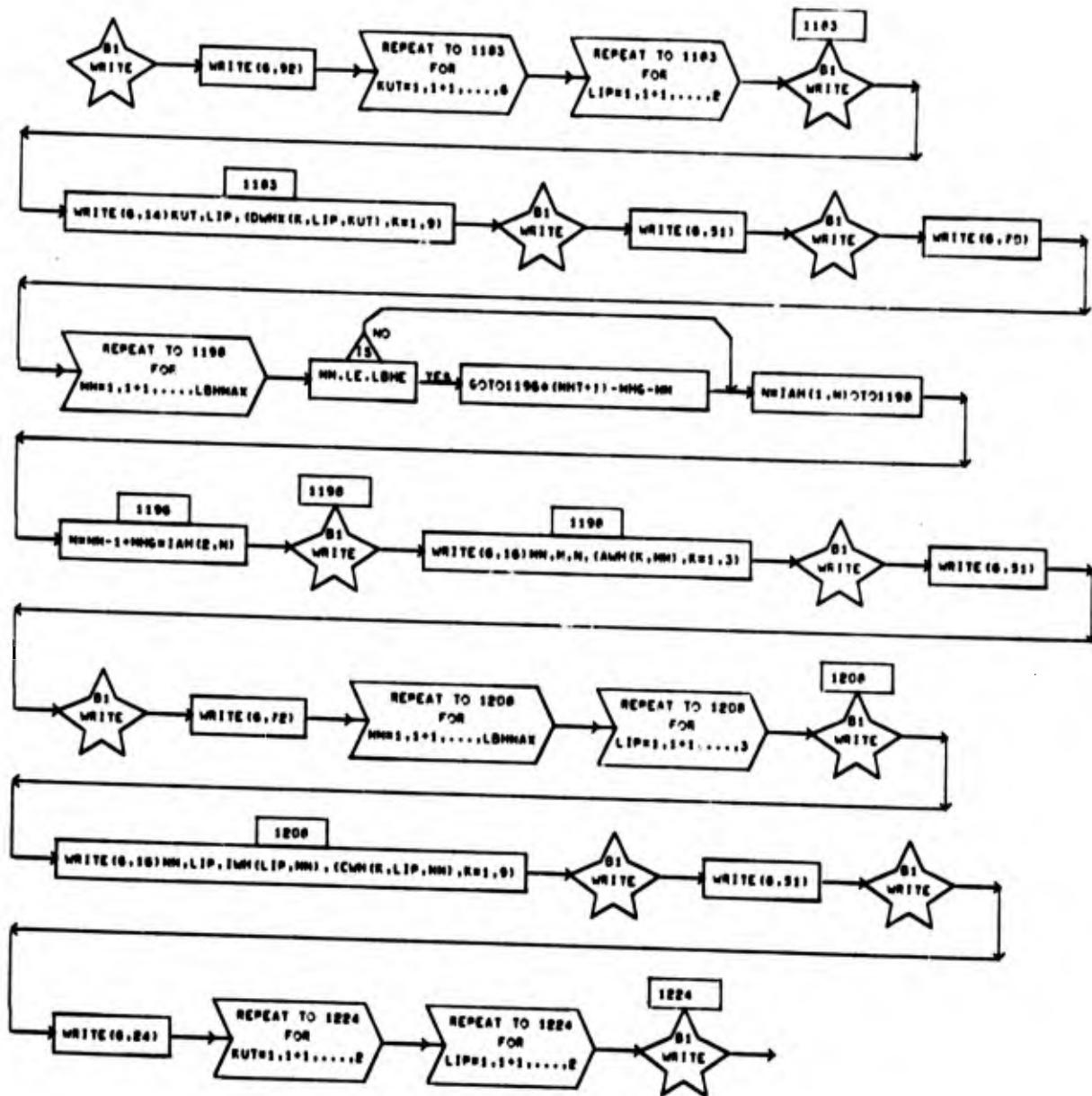
Detailed Flow Charts of Subroutine TEHAI (25 of 37)



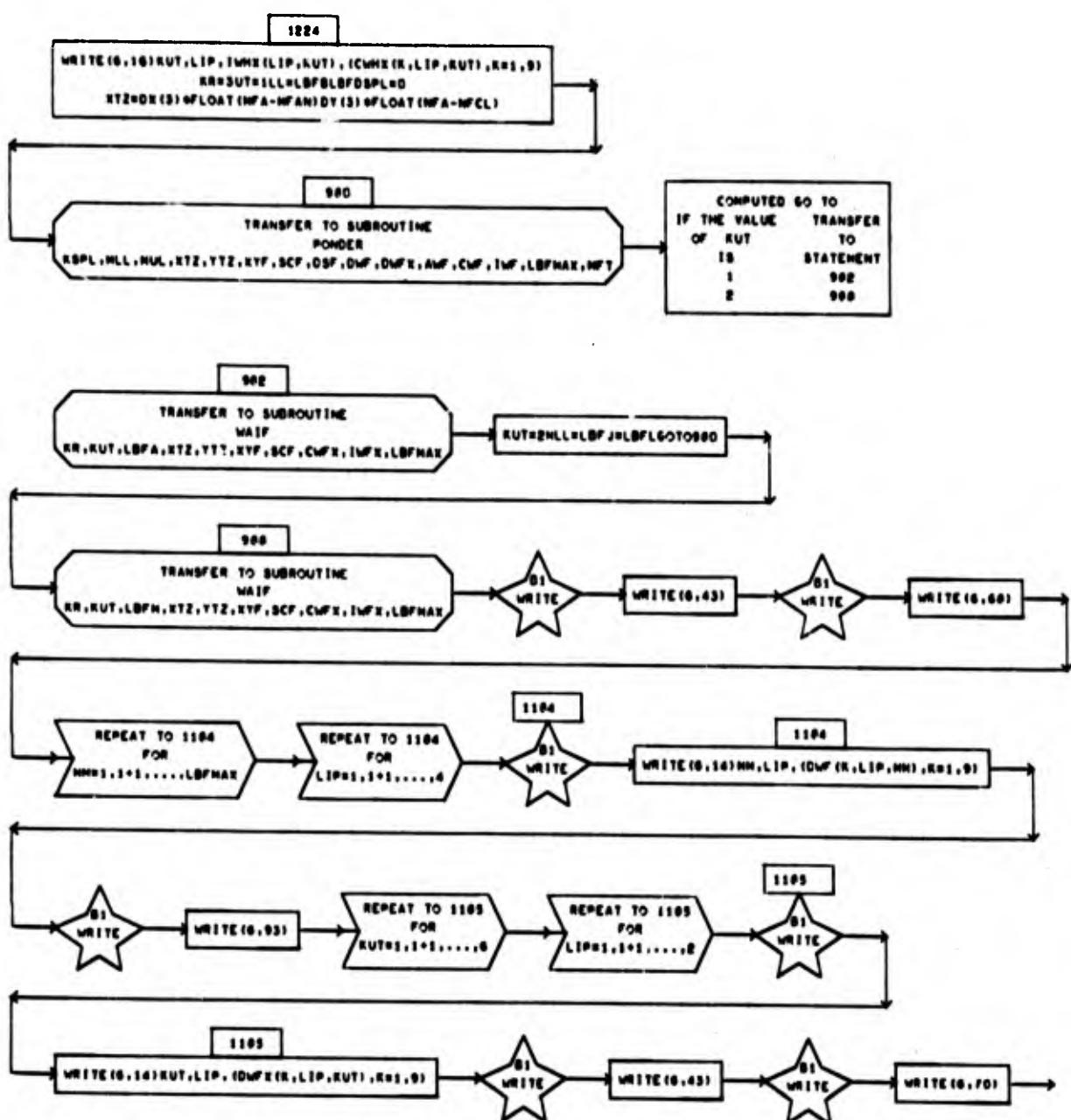
Detailed Flow Charts of Subroutine TEHAI (26 of 37)



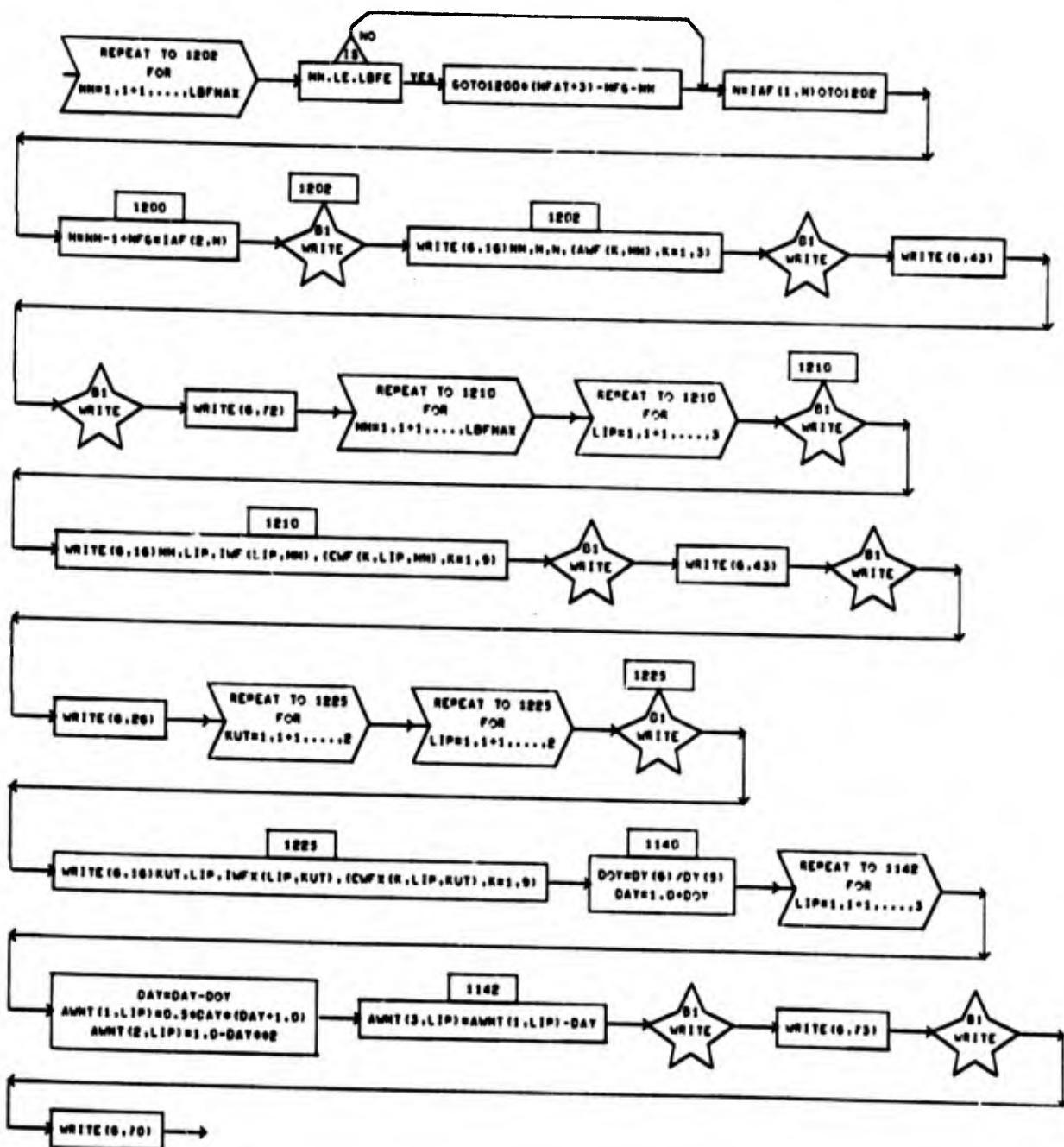
Detailed Flow Charts of Subroutine TEHAI (27 of 37)



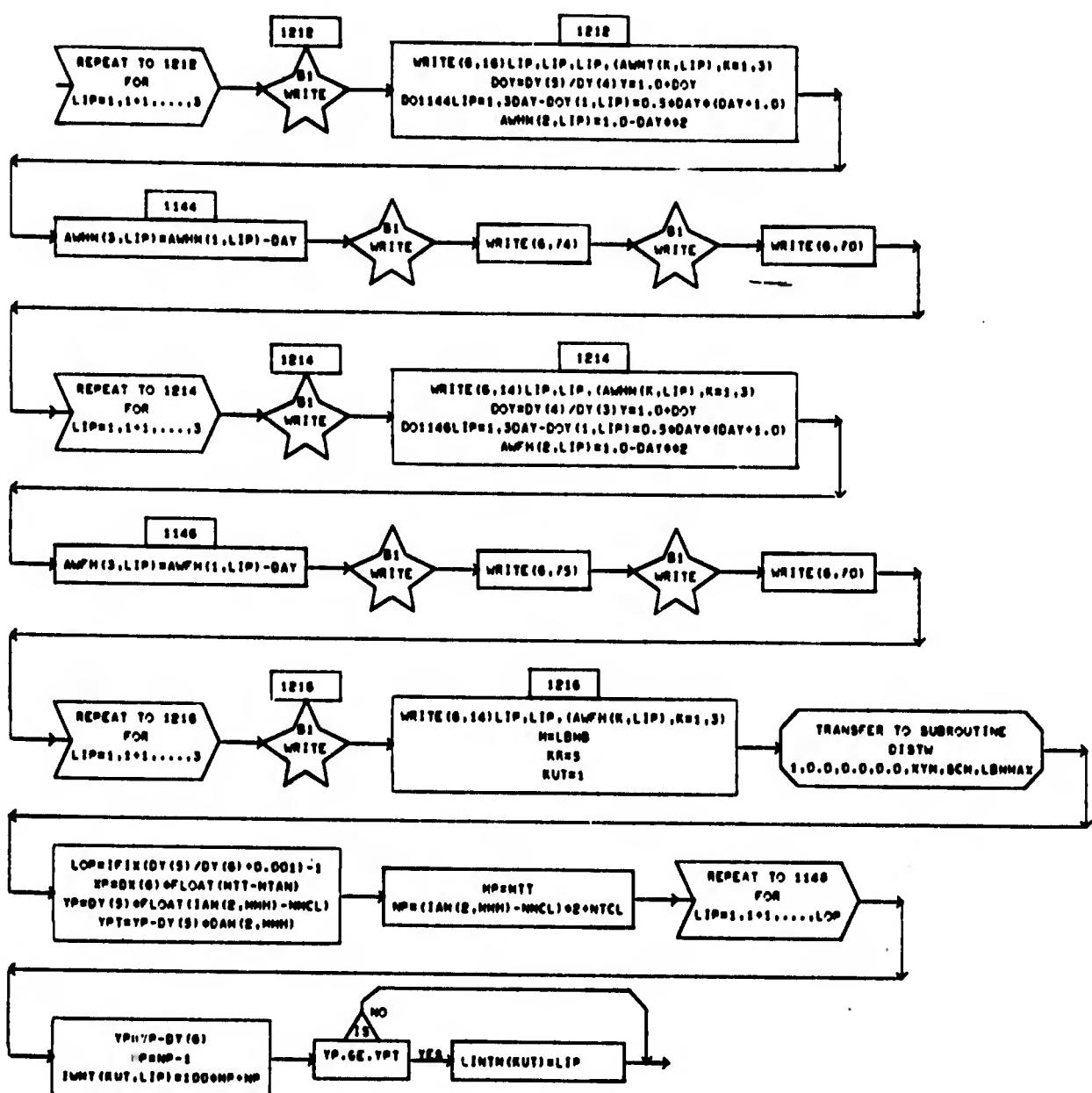
Detailed Flow Charts of Subroutine TEHAI (28 of 37)



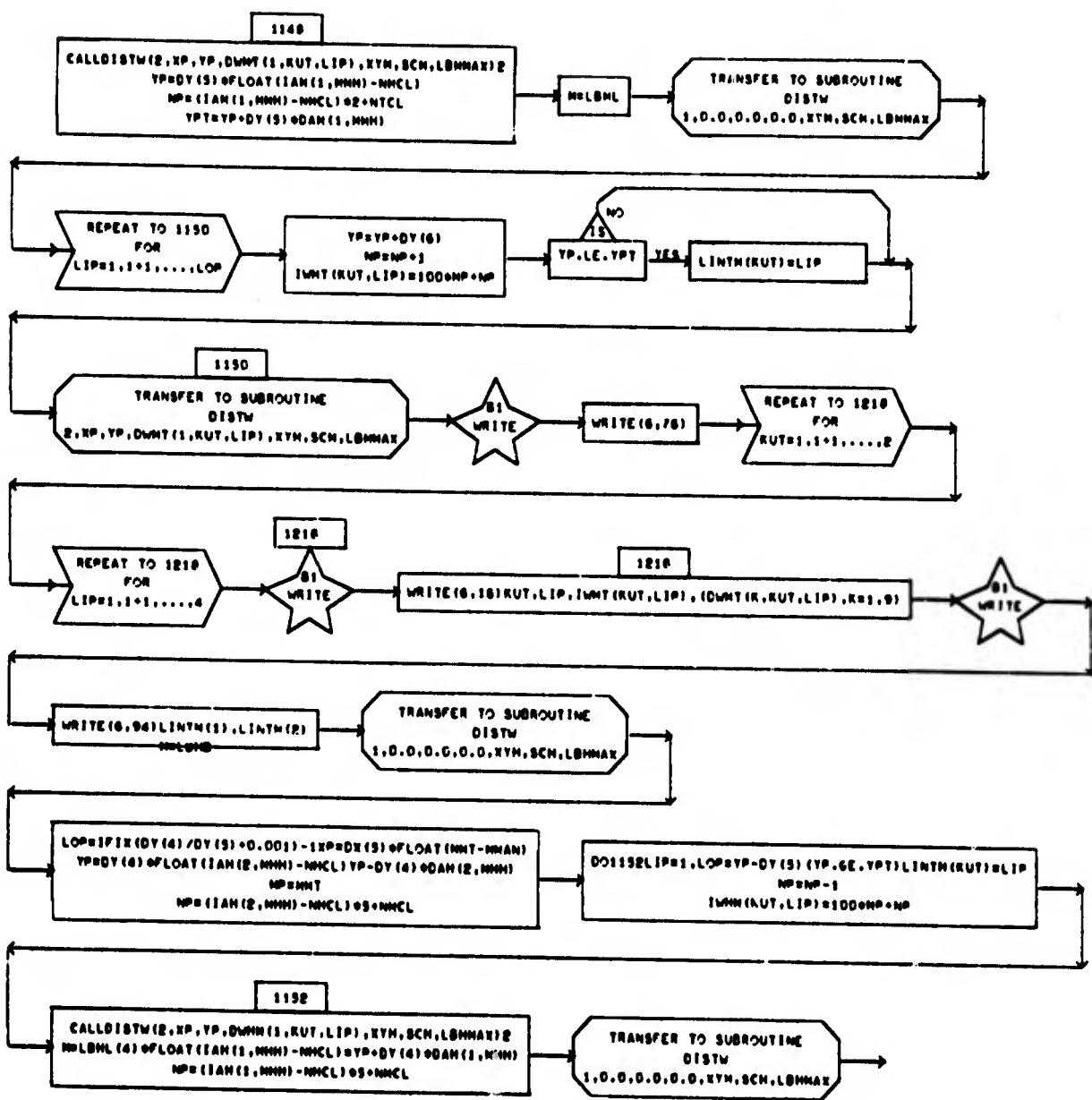
Detailed Flow Charts of Subroutine TEHAI (29 of 37)



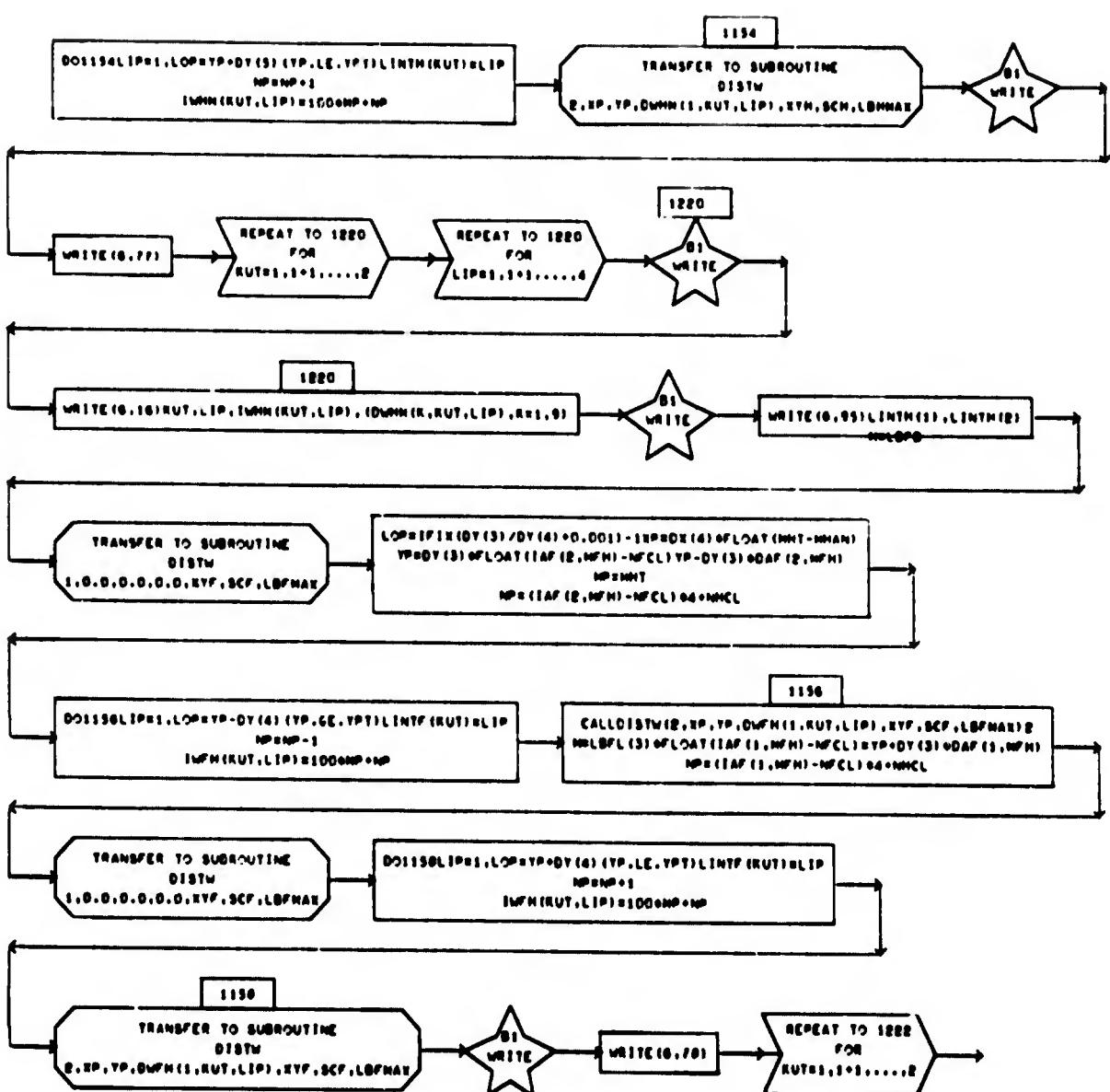
Detailed Flow Charts of Subroutine TEHAI (30 of 37)



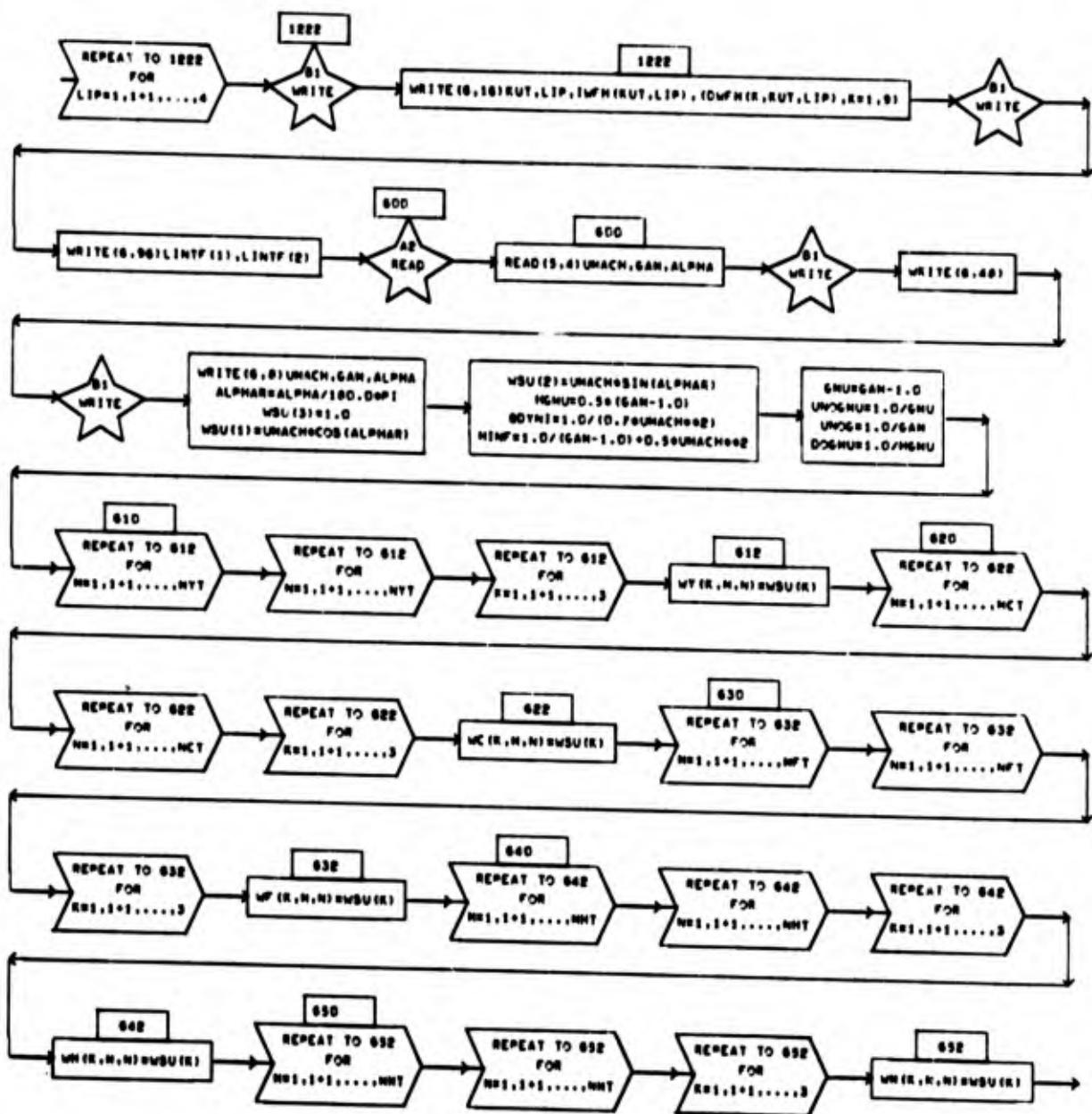
Detailed Flow Charts of Subroutine TEHAI (31 of 37)



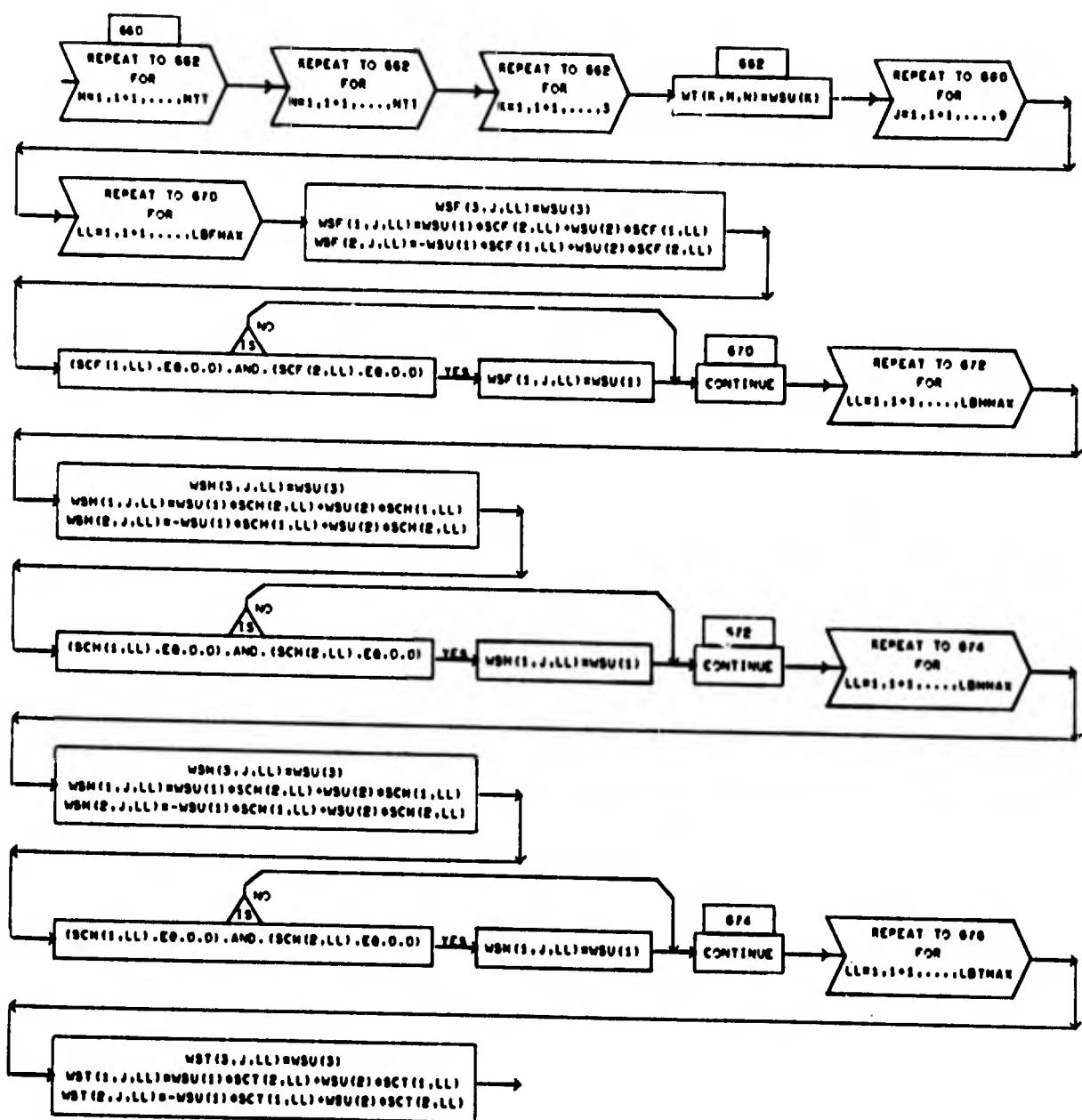
Detailed Flow Charts of Subroutine TEHAI (32 of 37)



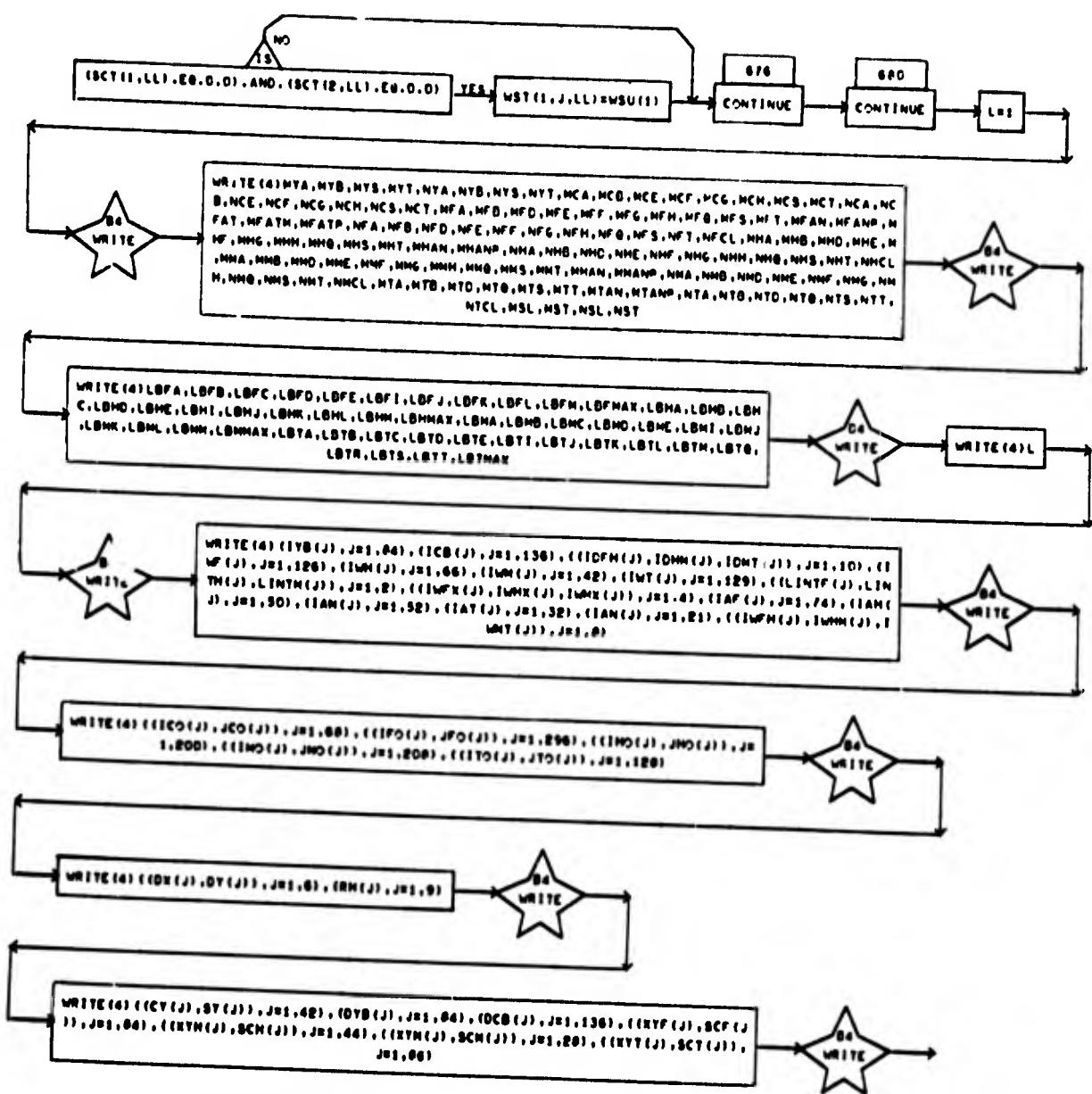
Detailed Flow Charts of Subroutine TEHAI (33 of 37)



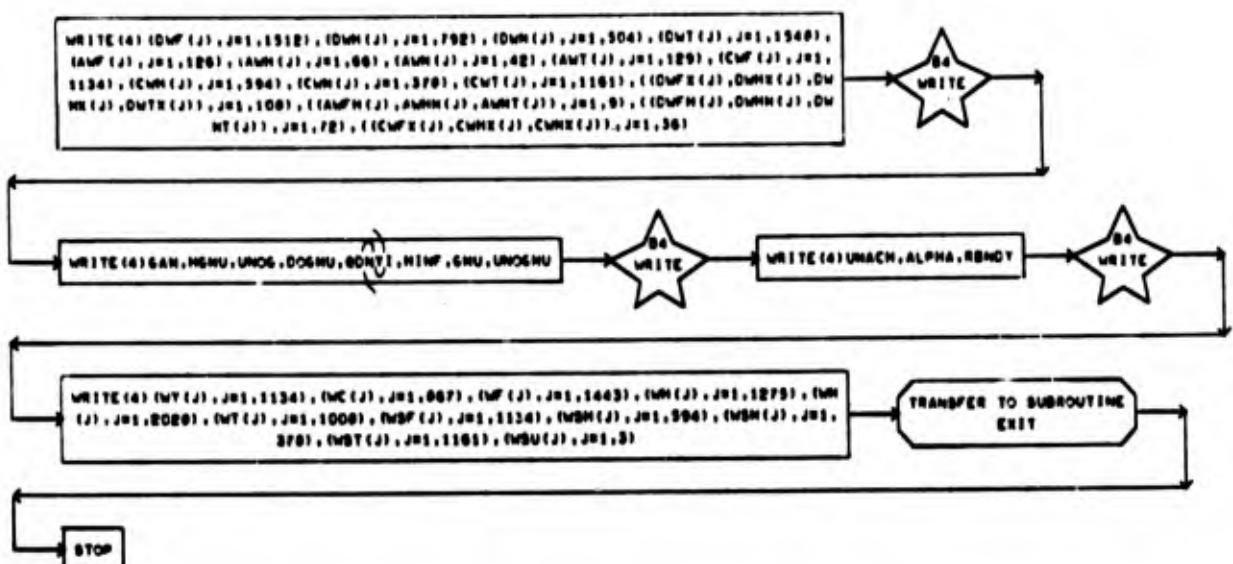
Detailed Flow Charts of Subroutine TEHAI (34 of 37)



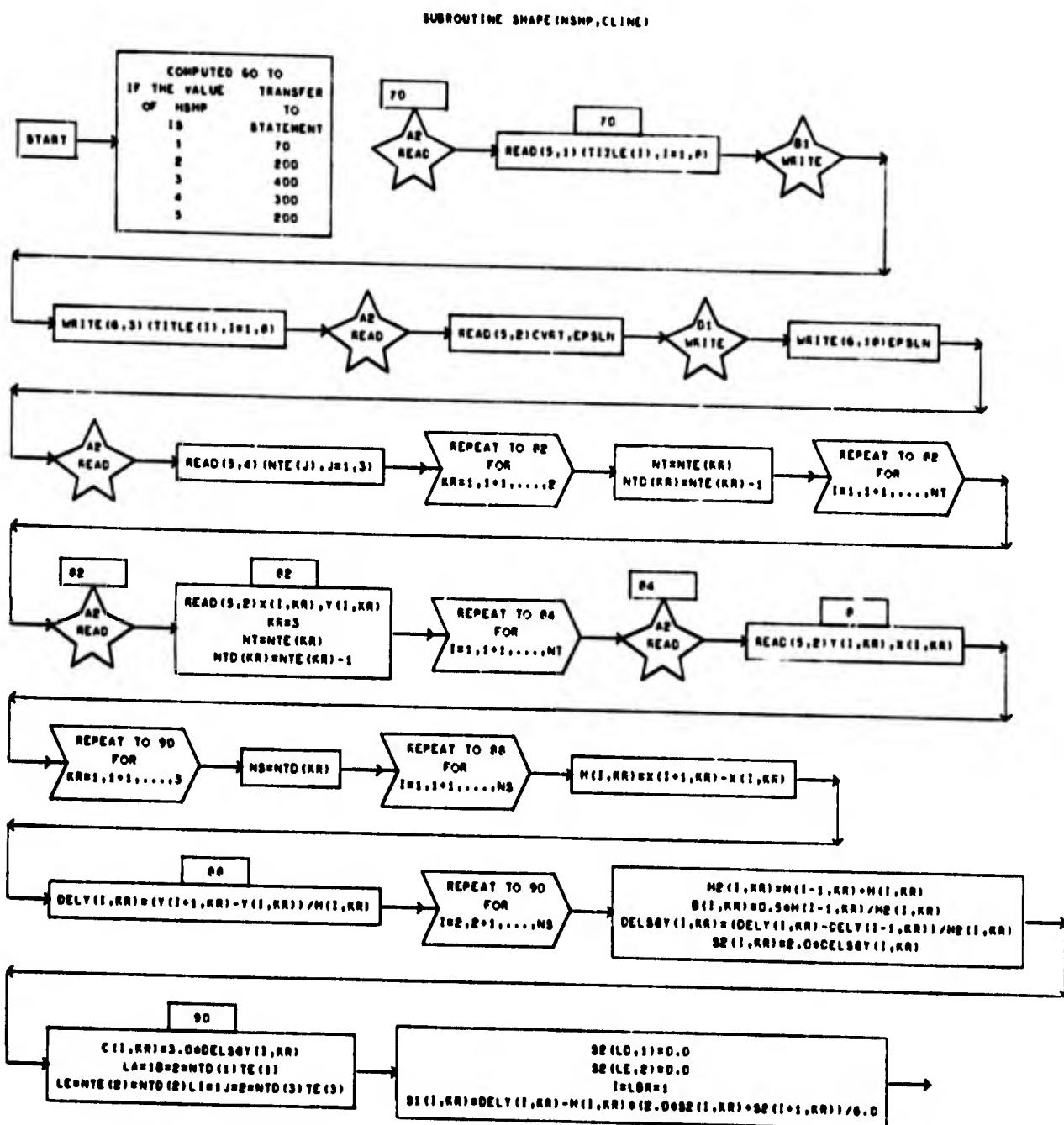
Detailed Flow Charts of Subroutine TEHAI (35 of 37)



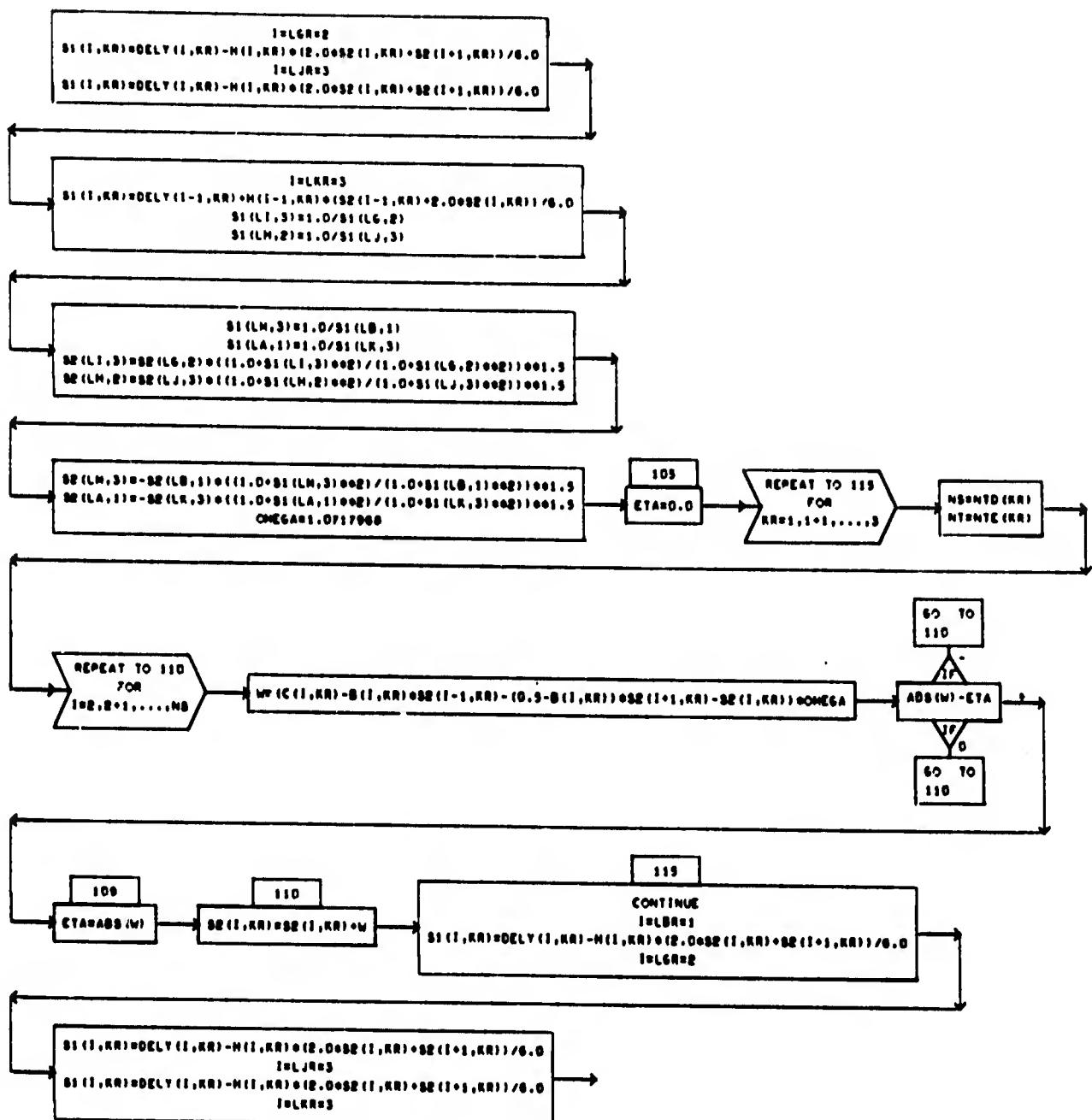
Detailed Flow Charts of Subroutine TEHAL (36 of 37)



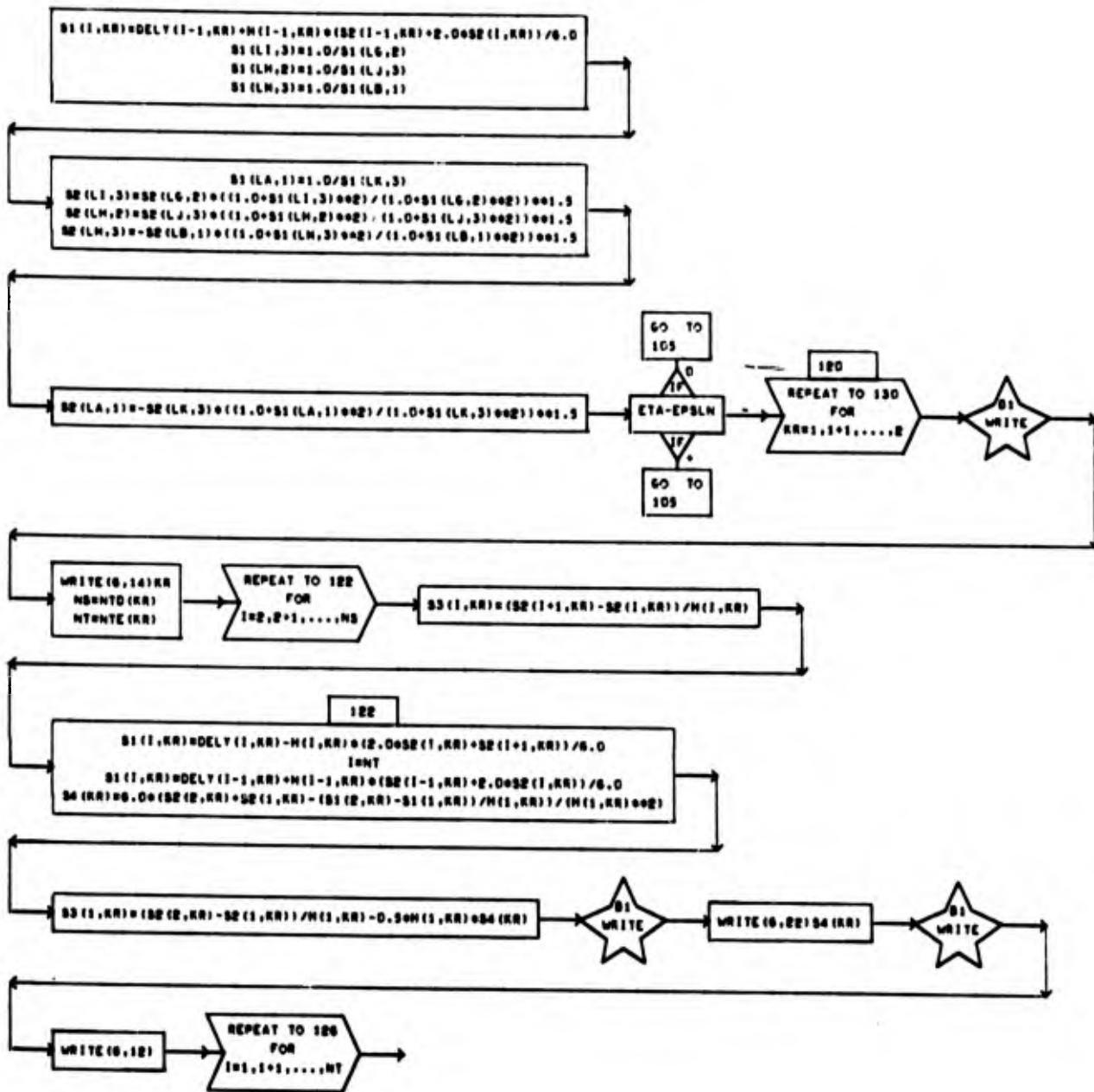
Detailed Flow Charts of Subroutine TEHAI (37 of 37)



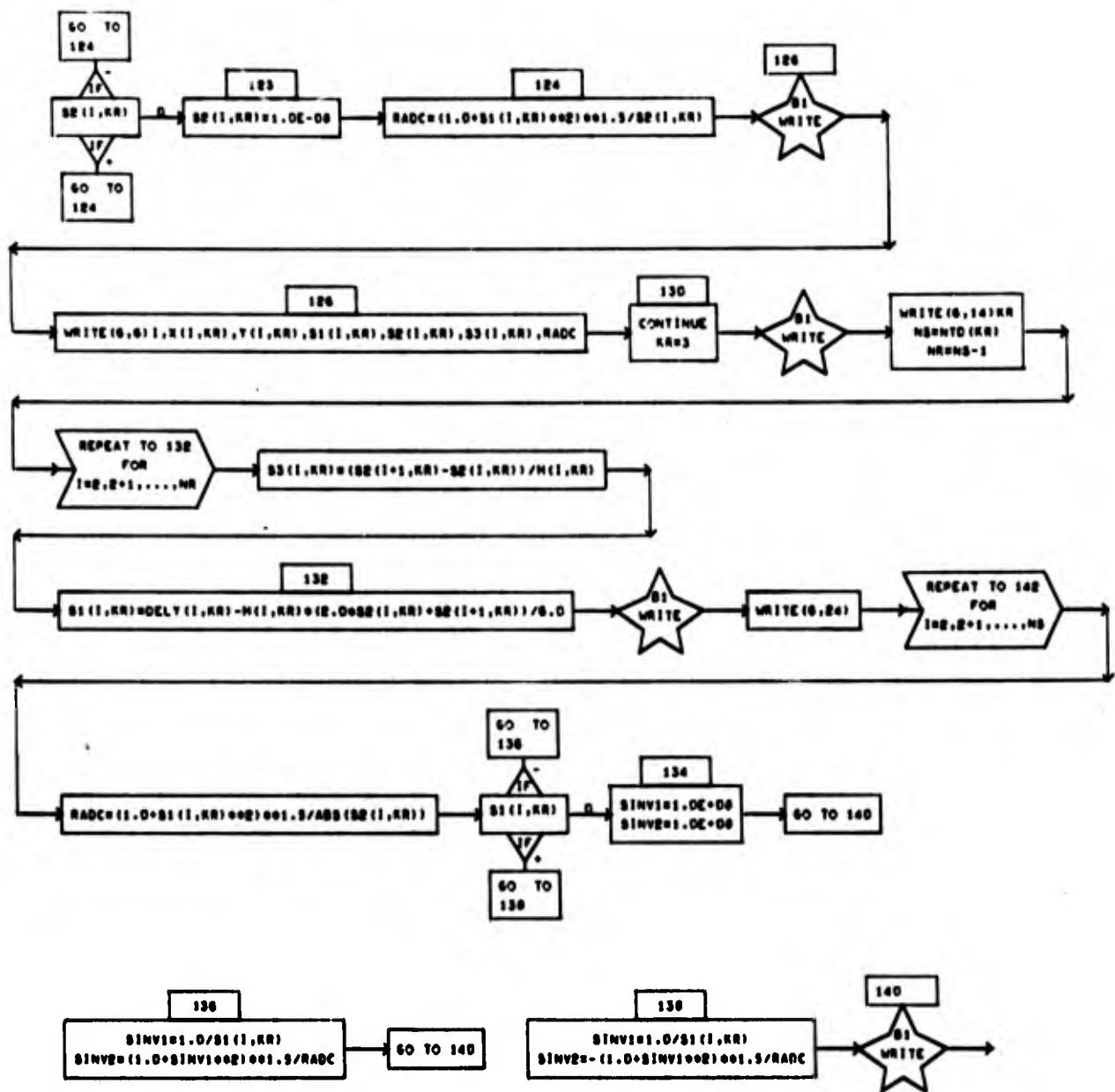
Detailed Flow Charts of Subroutine SHAPE (1 of 9)



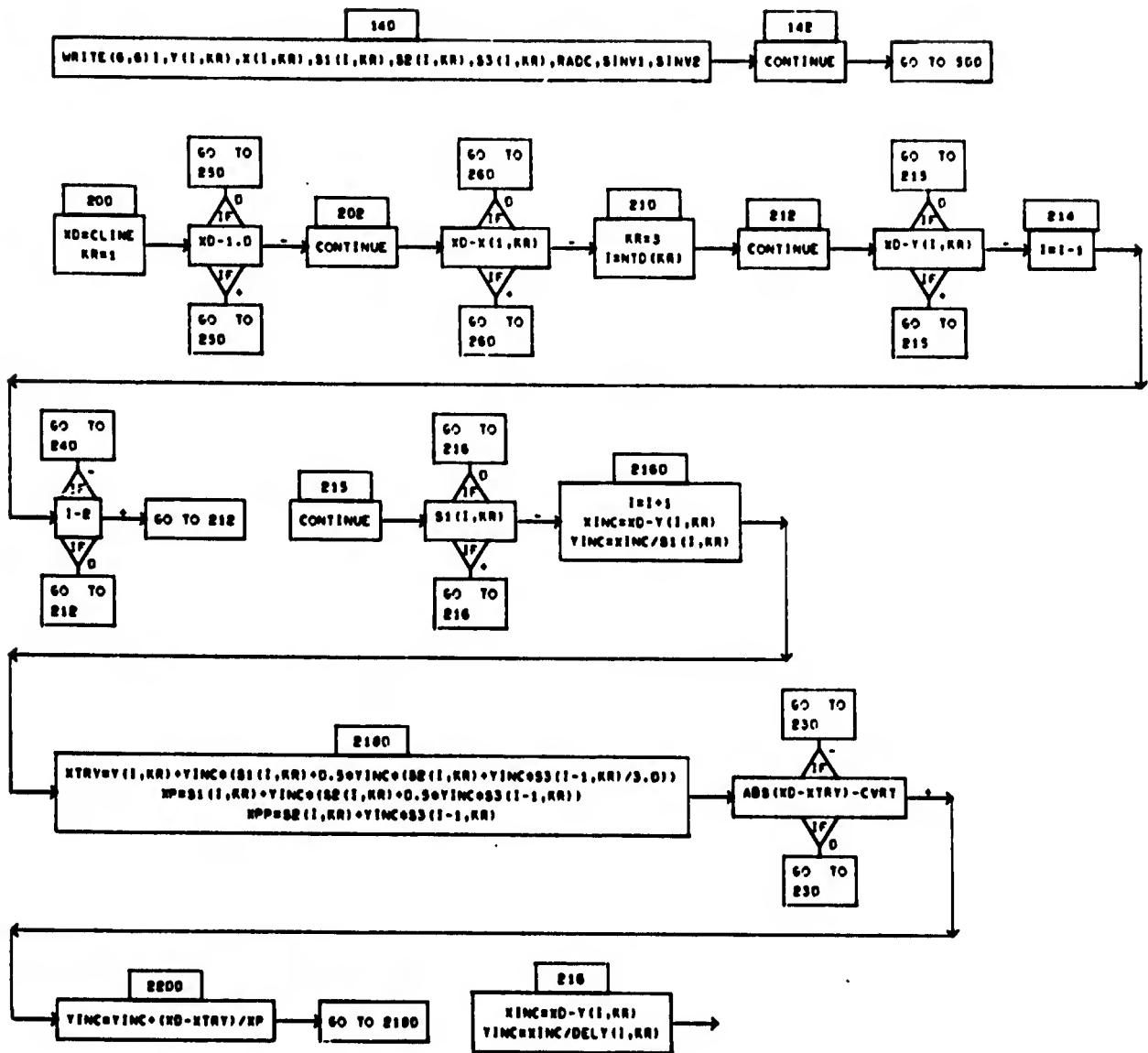
Detailed Flow Charts of Subroutine SHAPE (2 of 9)



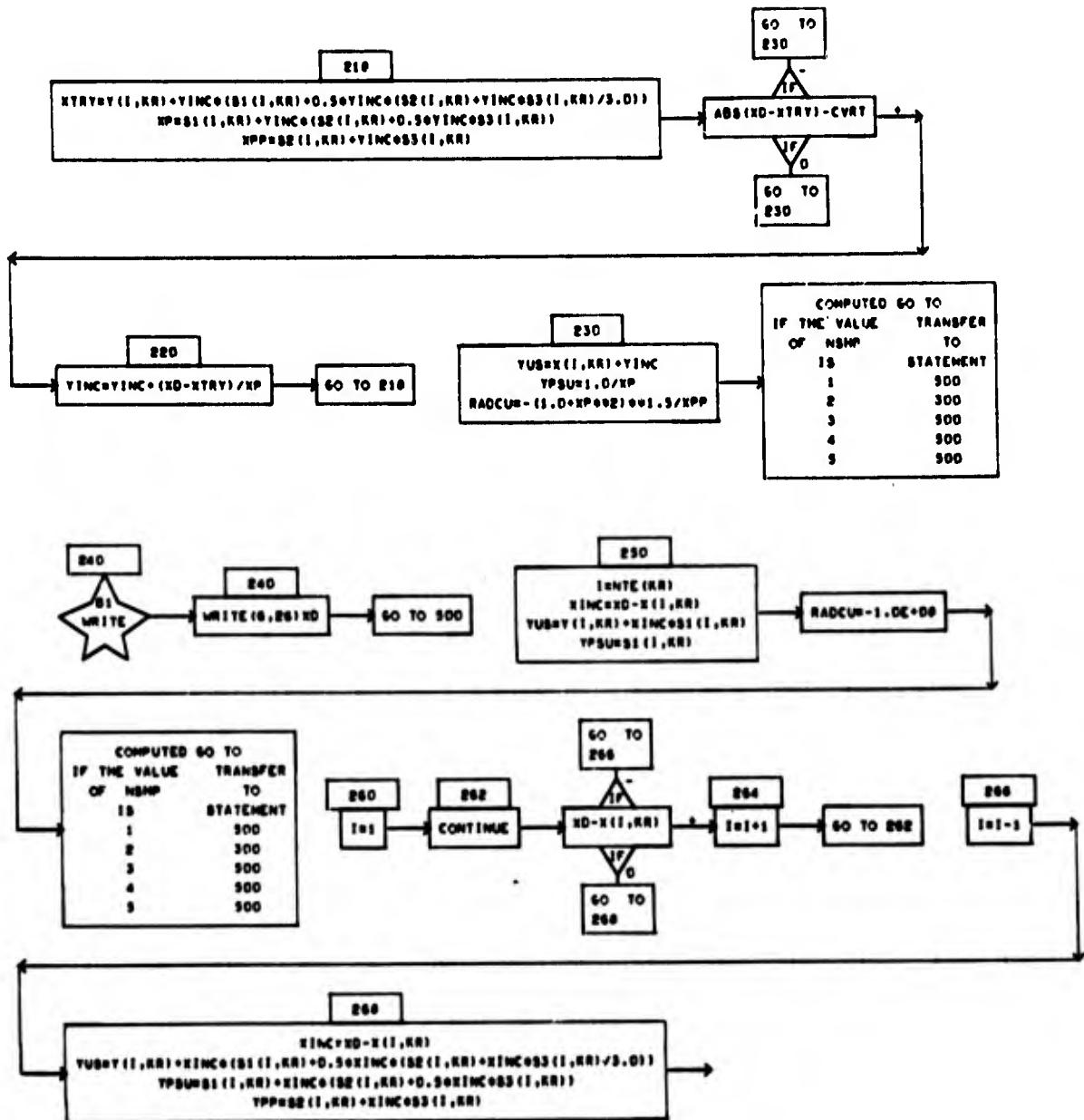
Detailed Flow Charts of Subroutine SHAPE (3 of 9)



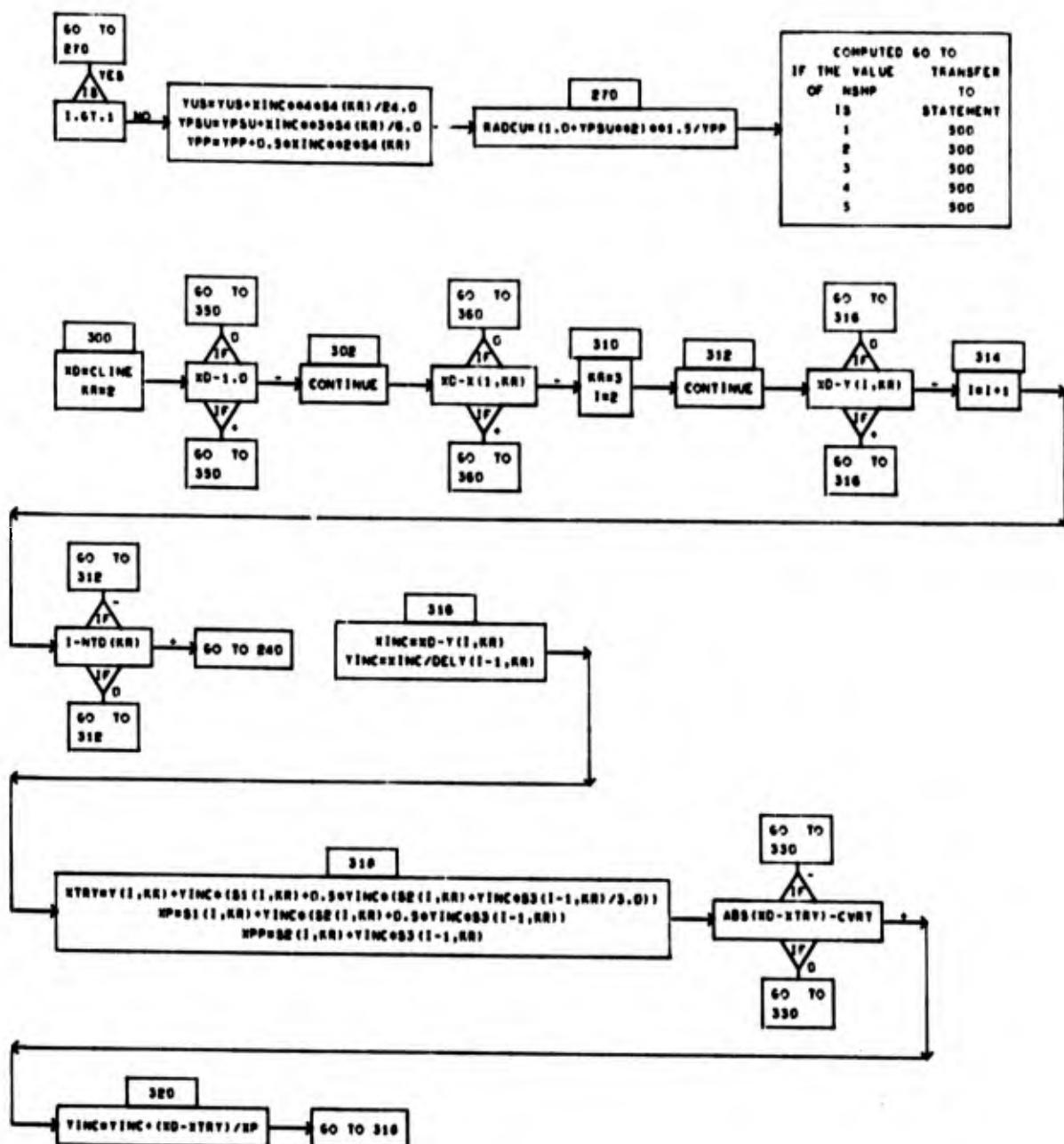
Detailed Flow Charts of Subroutine SHAPE (4 of 9)



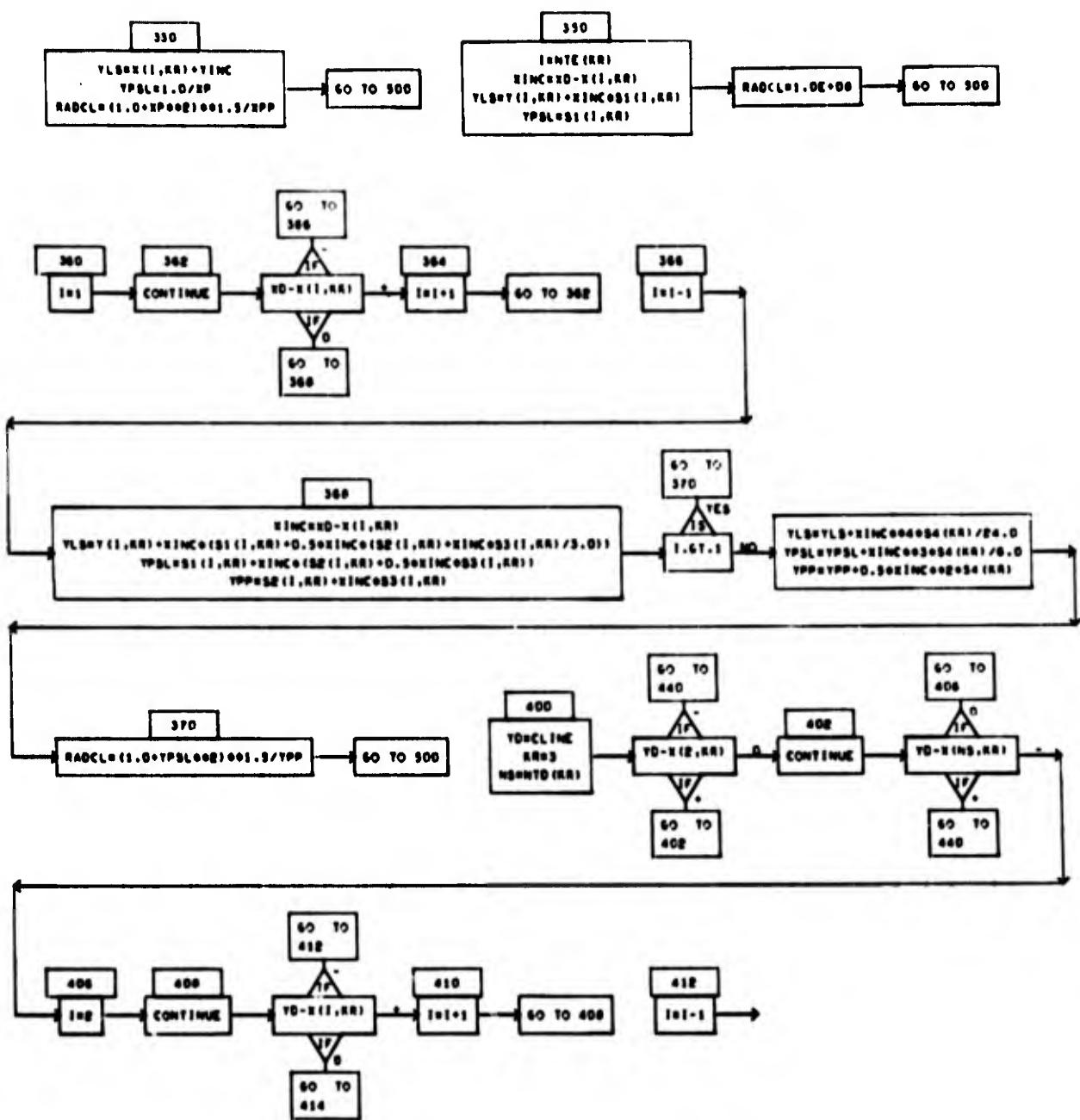
Detailed Flow Charts of Subroutine SHAPE (5 of 9)



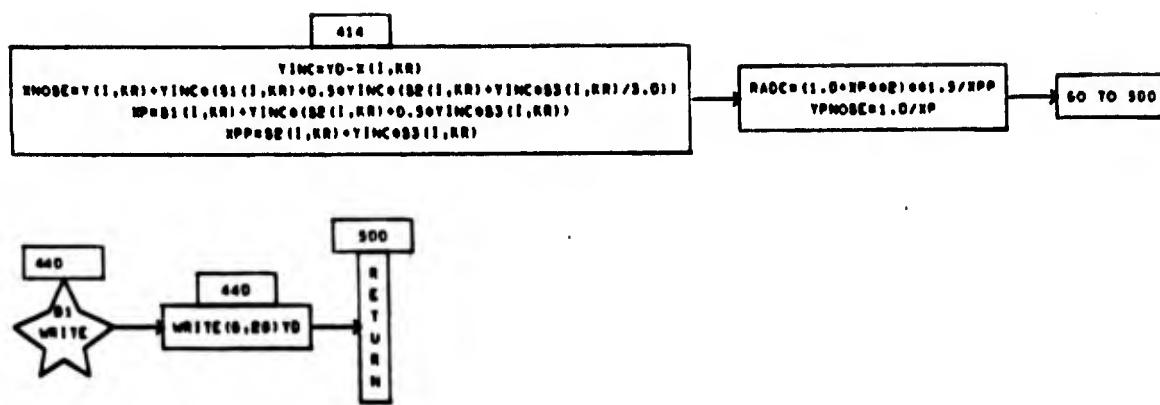
Detailed Flow Charts of Subroutine SHAPE (6 of 9)



Detailed Flow Charts of Subroutine SHAPE (7 of 9)

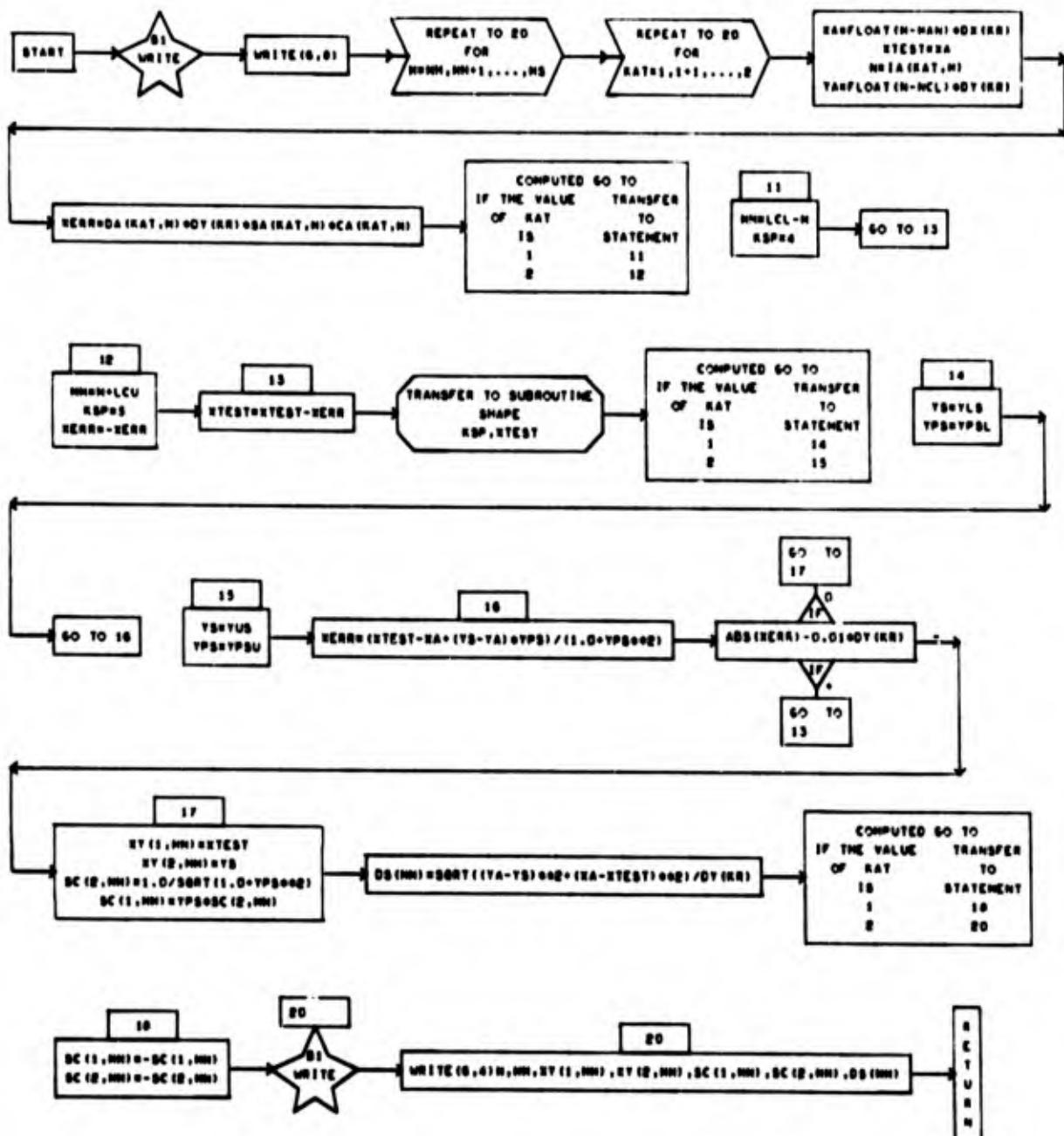


Detailed Flow Charts of Subroutine SHAPE (8 of 9)

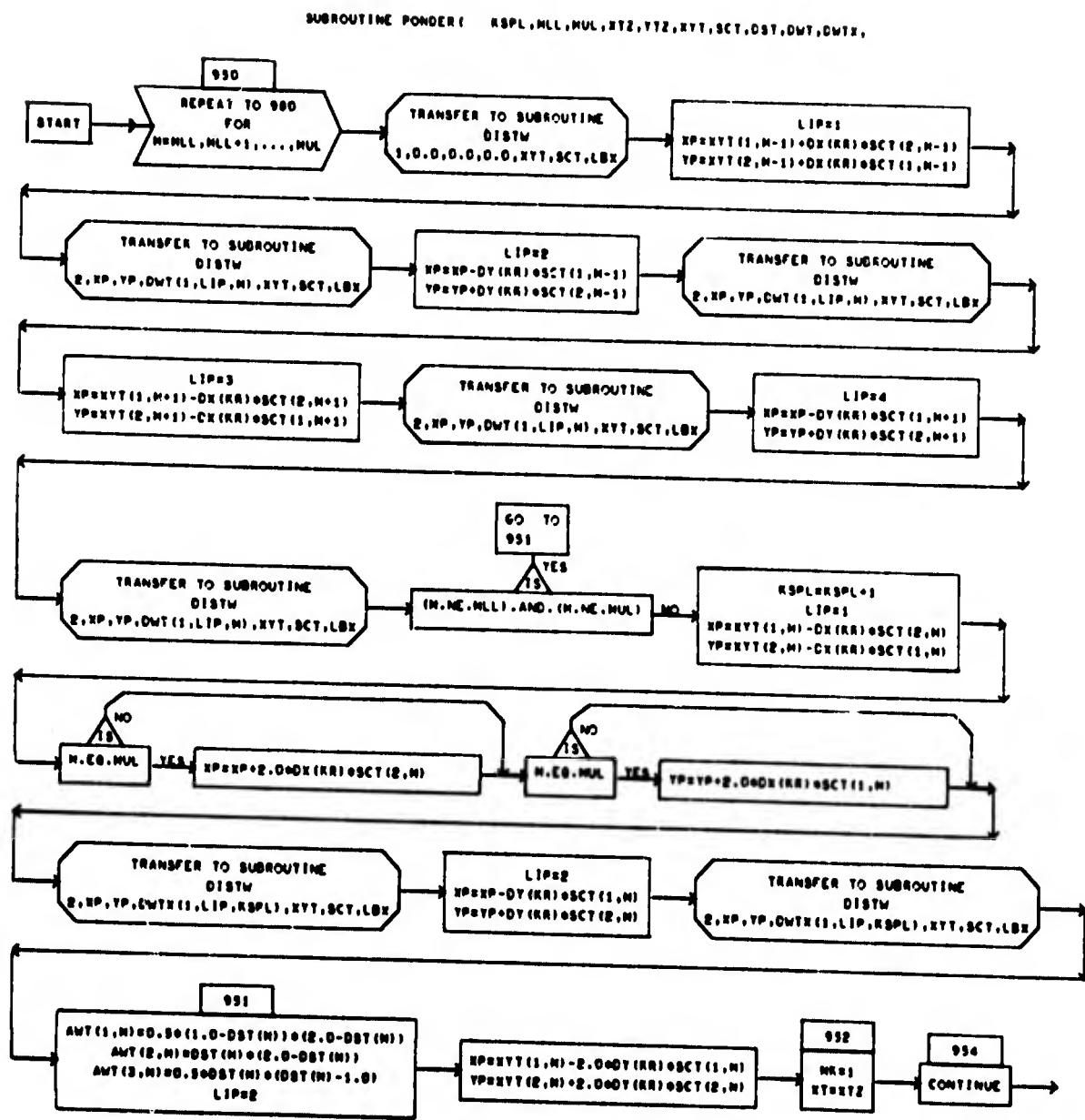


Detailed Flow Charts of Subroutine SHAPE (9 of 9)

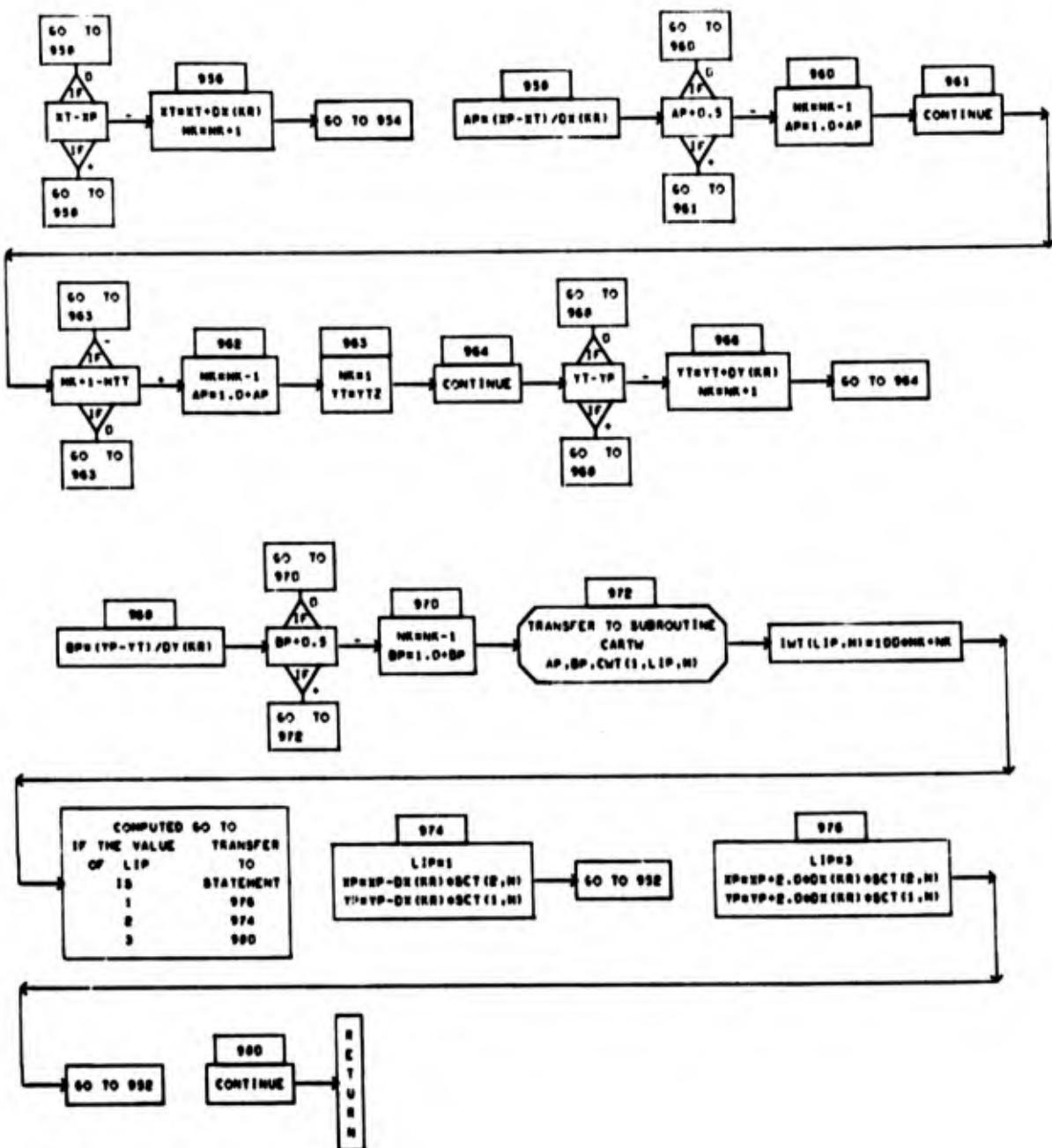
SUBROUTINE CANTERR,MN,MS,MAN,NCL,LCU,LCL,IA,DA,SA,CA,LA,



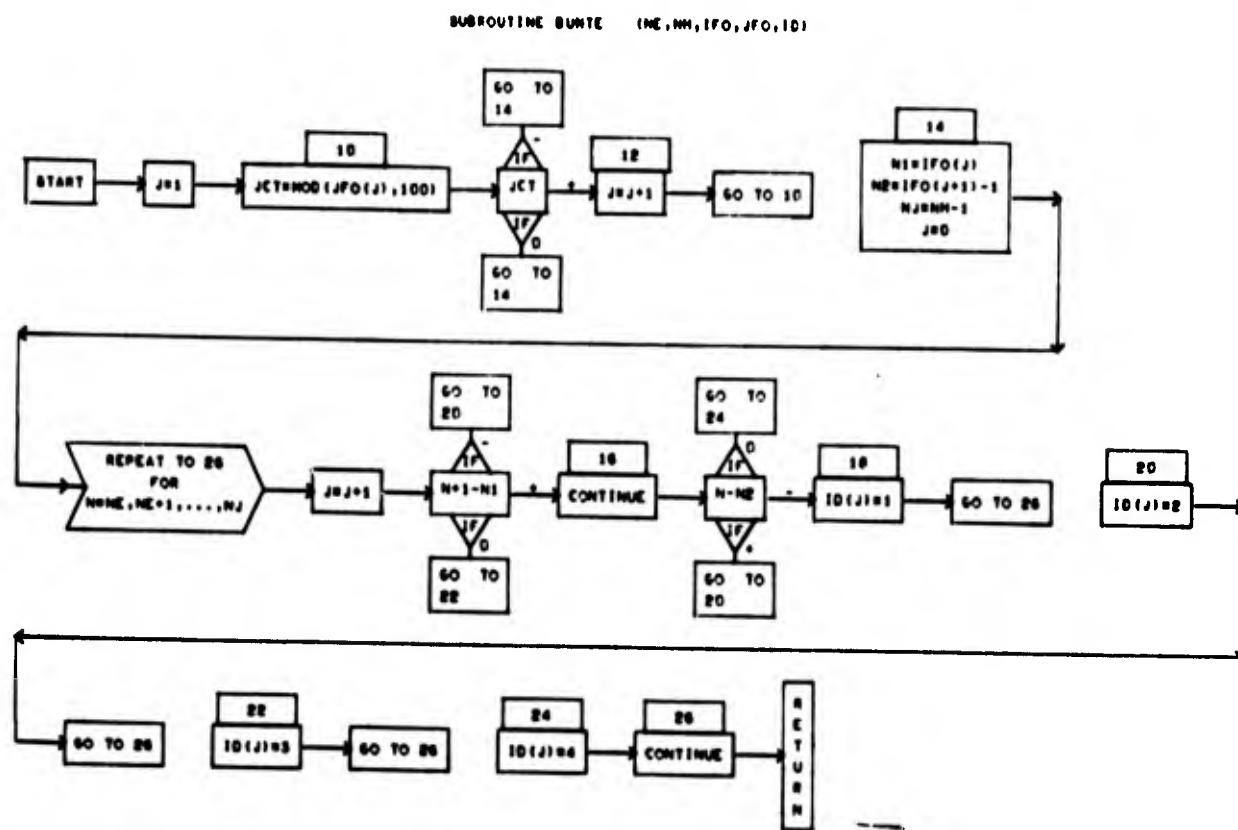
Detailed Flow Chart of Subroutine CAXIS



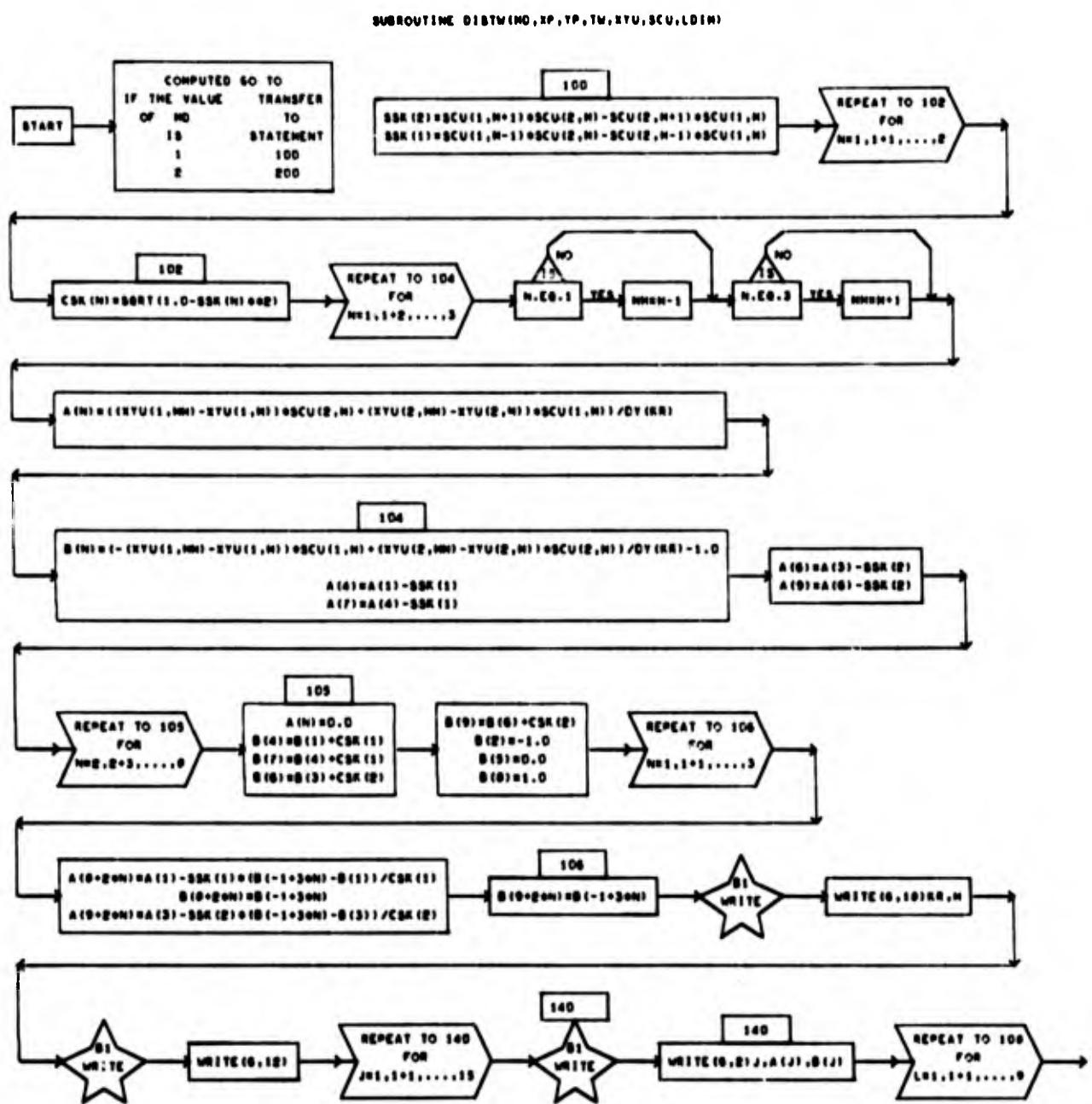
Detailed Flow charts of Subroutine PONDER (1 of 2)



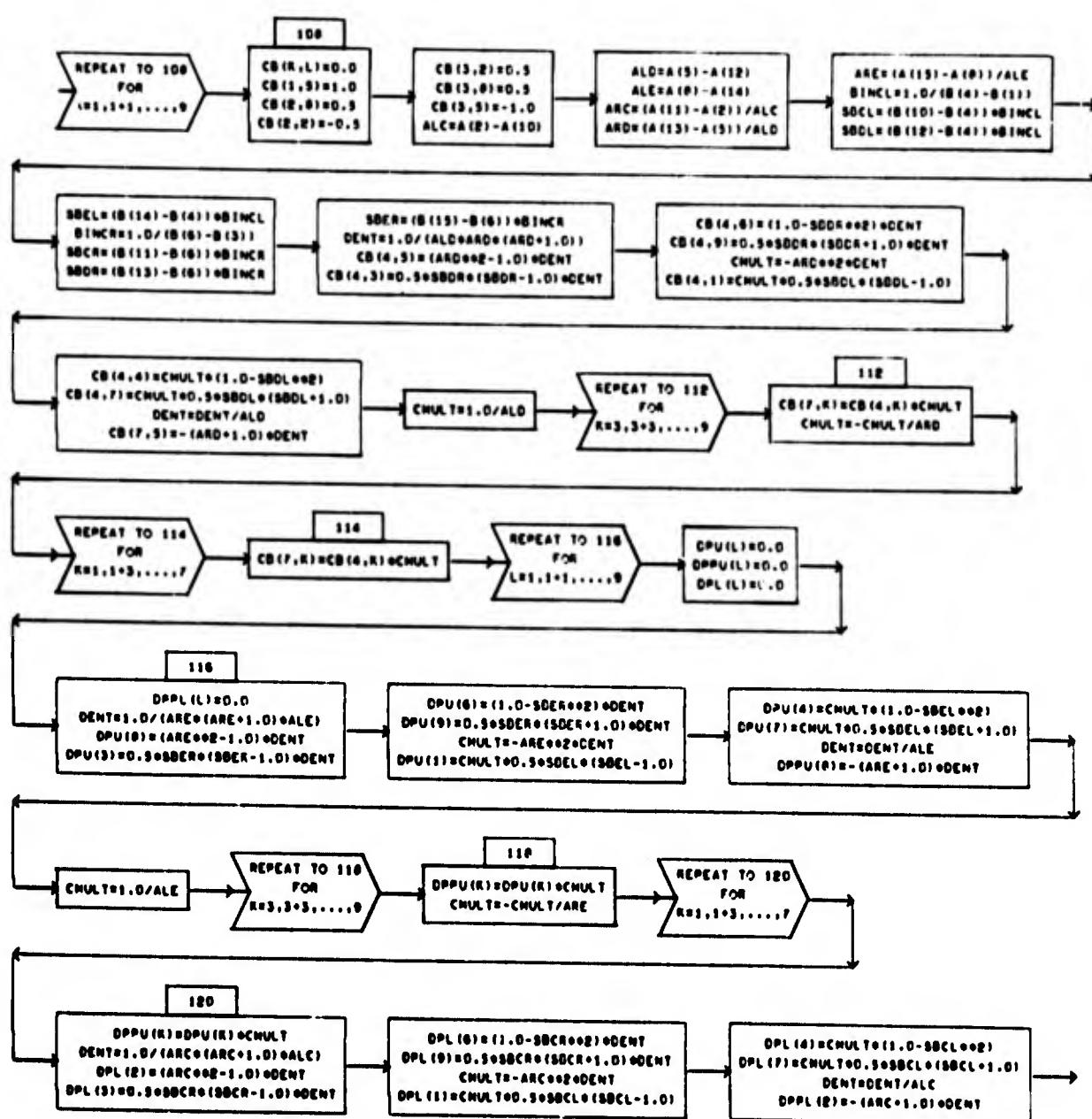
Detailed Flow Charts of Subroutine PCNDER (2 of 2)



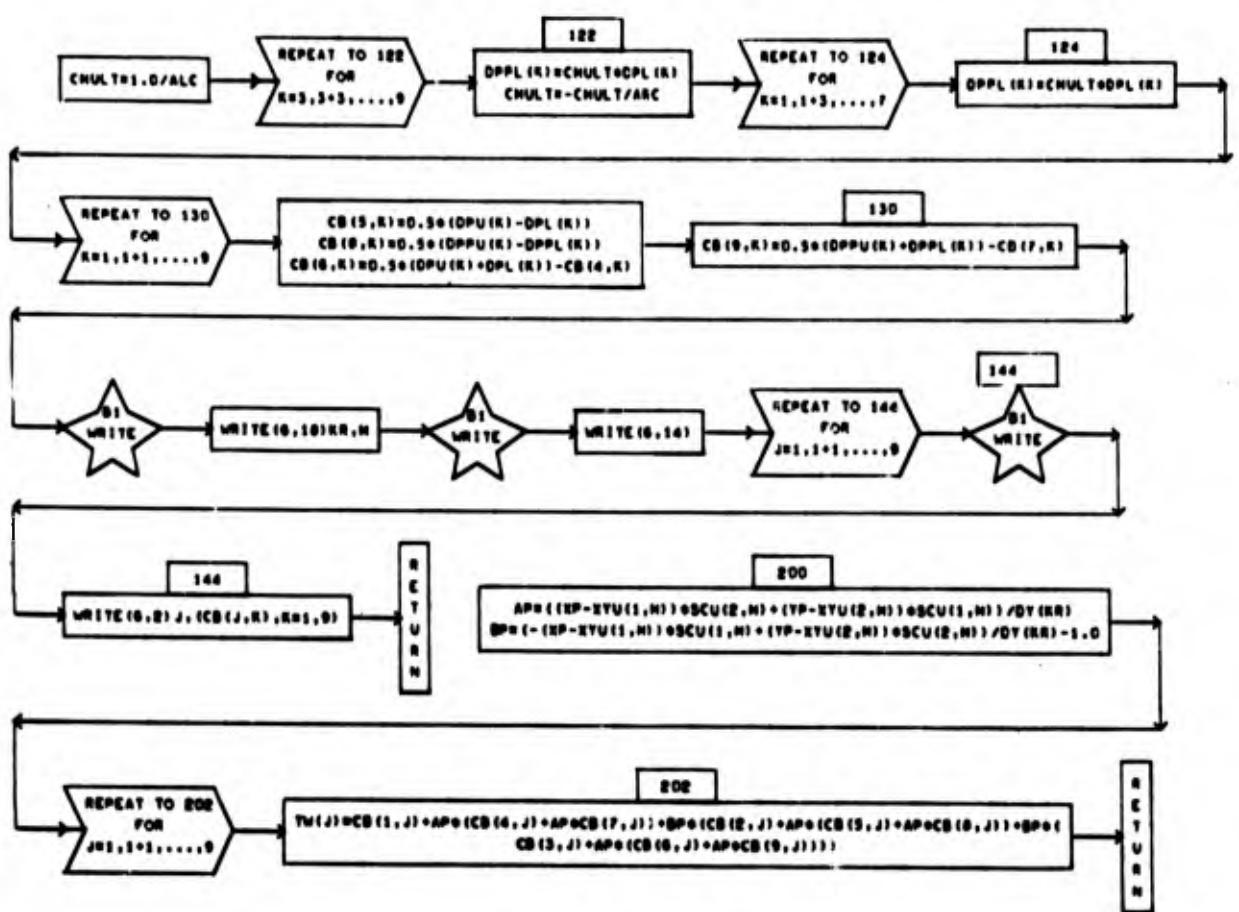
Detailed Flow Chart of Subroutine BUNTE



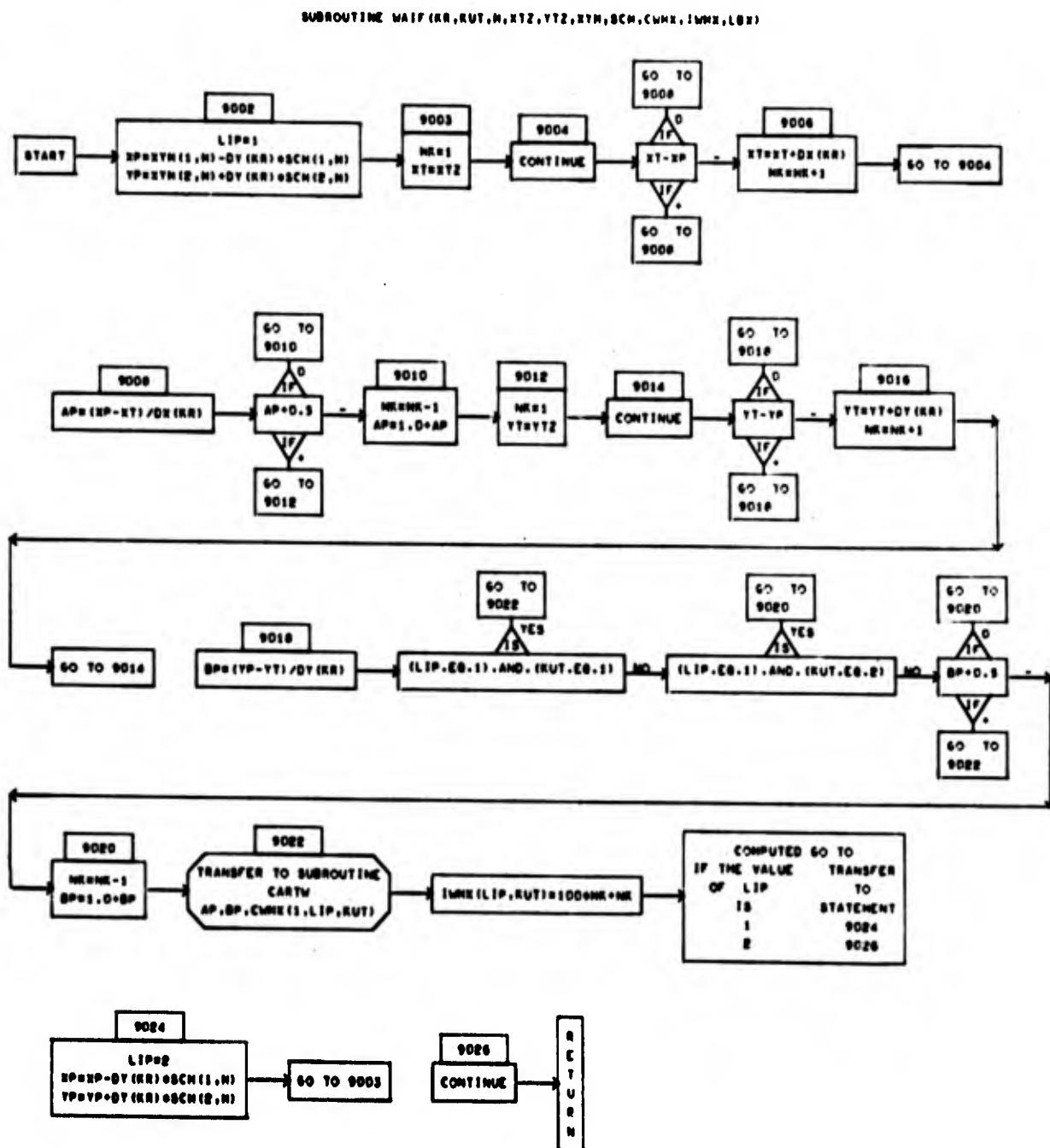
Detailed Flow charts of Subroutine DISTW (1 of 3)



Detailed Flow charts of Subroutine DISTW (2 of 3)



Detailed Flow charts of Subroutine DISTW (3 of 3)



Detailed Flow Chart of Subroutine WAIF

SUBROUTINE CARTW(AP,BP,TW)



Detailed Flow Chart of Subroutine CARTW

APPENDIX IV
FLOW FIELD PROGRAM LISTING

REQUEST(TAPE3,TYPE1=14713)	000000
REQUEST(TAPE4,RY,R,SV,ICERANLBIN)	000001
RFWIND(TAPE4)	000002
REWIND(TAPE3)	000003
SET(0)	000004
RUN(S,.....,1)	000005
LGO(LC,1UUU000)	000006
UNLOAD(TAPE3)	000007
UNLOAD(TAPE4)	000008
EXIT.	000009
UNLOAD(TAPE3)	000010
UNLOAD(TAPE4)	000011
RETURN(TAPE4)	000012
R	000013
PROGRAM NAME? INPUT,OUTPUT,TAPES=INPUT,TYPE6=OUTPUT,TAPES,TAPE4)	000014
C	000015
C****LLUNTED LIFTING TRANSONIC AIRFOIL	000016
C	000017
COMMON/LIMITS/ MYA, MYB, MYS, MYT, NYA, NYH, NYS, NYT,	000018
C MCN, MCU, NCE, NCF, MCG, MCH, MCS, MCT, NCA, NCB, NCE, NCF,	000019
C NCG, NCN, NCS, NCT,	000020
F MFA, MFD, MFF, MFG, MFH, MFO, MFS, MFT, MFAN,	000021
F MFAT, MFATP, MFA, MFB, MFD, MFE, MFF, MFG, MFH,	000022
F NFU, NFS, NFT, NFCL,	000023
H MHM, MHD, MHE, MHF, MHG, MHF, MHO, MHS, MHT, MTAN, MHANP,	000024
H NHM, NHU, NHD, NHE, NHF, NHG, NHH, NHU, NHS, NHT, NHCL,	000025
M MMH, MMU, MMD, MMF, MMG, MMH, MHO, MHS, MNT, MMAN, MMANP,	000026
M NMH, NMU, NMD, NMF, NMF, NMH, NMU, NMS, NMT, NMCL,	000027
T MTB, MTD, MTQ, MTS, MTT, MTAN, MTANP, NTA, NTB, NYT,	000028
T NTU, NTS, NTT, NTCL,	000029
S MSL, MST, NSL, NST	000030
COMMON/FLIMIT/ LBFA, LBFL, LBFC, LBFO, LBFE, LBFI, LBFI, LBFK,	000031
F LBFL, LBFM, LBFMAX,	000032
H LBHA, LBHD, LBHG, LBHD, LBHE, LBHI, LBHQ, LBHK, LBHL, LBHM,	000033
H LBHMAX,	000034
M LBVA, LBVB, LBVC, LBMD, LBME, LBVI, LBVJ, LBVK, LBVL, LBMM,	000035
M LBMMAX,	000036
T LBVA, LBVB, LBVC, LBMD, LBME, LBVI, LBVJ, LBVK, LBVL, LBTM,	000037
T LBVA, LBVB, LBVC, LBMD, LBME, LBVI, LBVJ, LBVK, LBVL, LBTMAX	000038
COMMON/ZMESH/ DX(6), DY(6), RH(9)	000039
COMMON/GEOMF/ SY(42), CY(42), DY(2,42), DCB(2,4,17),	000040
A XYF(2,42), SCF(2,42), XYH(2,22), SCH(2,22),	000041
H XY(2,14), SC(2,14), XYT(2,43), SCT(2,43)	000042
COMMON/OPLM/ ICO(4,17), JCU(4,17), IF(8,37), JF(8,37),	000043
A JHU(8,25), JHU(8,25), IMO(8,26), IT(8,16), JT(8,16)	000044
COMMON/DURA/ L	000045
COMMON/TERSE/ IYB(2,42), ICN(2,4,17), IDFH(10), IDHM(10),	000046
A IDP(10), IWF(3,42), IWH(3,22), IW(3,14), IWT(3,43),	000047
B LINF(2), LINH(2), LINTM(2), IWFX(2,2), IWGX(2,2), IWGX(2,2),	000048
C IAF(2,37), IAH(2,25), IAM(2,26), IAT(2,16), IAN(2),	000049
D INF(2,4), IWIM(2,4), IWMT(2,4)	000050
COMMON/OMOSA/ DWF(9,4,42), DWH(9,4,22), DWM(9,4,14),	000051
A DWI(9,4,43), AWF(3,42), AWH(3,22), AWM(3,14), AWY(3,43),	000052
B CWF(9,3,42), CWH(9,3,22), CM(9,3,14), CW(9,3,43),	000053
C DWF(9,2,6), DWFX(9,2,6), DWMX(9,2,6), DWYX(9,2,6),	000054
D AWFI(3,37), AWIM(3,37), AWMT(3,37),	000055
E DWF(9,2,4), DWHM(9,2,4), DWM(9,2,4),	000056

F CWFX(9,2,2)	CWDX(9,2,2)	CWDX(9,2,2)	000057
COMMON/GAS7	GAM, HGMU, UNOG, DOGMR, GMV, HINF, GMD, UNOGMU	000058	
COMMON/FLOW	UMACH, ALPHA, RBNDY	000059	
COMMON/PETID	CJX(3,3), DCX(3,6), DCY(3,6), DHU(3,3), DMDX(5,6),	000060	
A DMRY(5,6), LBIX(6), LBYT(6)	000061		
COMMON/FLAGS	IBUG, JBCT, JOFB	000062	
COMMON/ENTHP	JIND, TEND, TENB	000063	
COMMON/TOC/PRMSH	ATSFY, HGPU, HGPV, HGPW, ATSF(6), TSN(6), PRMSH(6),	000064	
1 PRMSH(6), PRHYI, CFTT	000065		
COMMON	WY(3,42,9), WCT(3,17,17), WF(3,37,13), WH(3,25,17),	000066	
A WM(3,26,26), WT(3,16,21), WSF(3,9,42), WSH(3,9,22), WSM(3,9,14),	000067		
B WST(3,9,43), WSU(3)	000068		
C DIMENSION DAMP(8), SAFE(8), TITLE(8)	000069		
C----WEIGHTING FACTORS FOR DERIVATIVES	000070		
DATA CJX/-1.5,-0.5,0.5,2.0,1.0,-2.0,-0.5,0.5,1.5/	000071		
C	000072		
C----IBUG = 1 NORMAL PRINTOUT	000073		
C----IBUG = 2 CALLS FOR PRINTOUT OF EACH STEP IN THE FINE REGION	000074		
C----INIT = 1 DO NOT SAVE TAPE OF LAST LINE	000075		
C----INIT = 2 SAVE LAST LINE ON TAPE	000076		
C----JIND = 1, NO FORCING OF FIELD TOWARD STEADY FREE-STREAM TOTAL ENTH	000077		
C----IF JIND = 2 - KICK BOUNDARY POINTS AND ADJACENT REGULAR MESH POINTS	000078		
C-----LY TEND	000079		
C----JIND = 3, KICK ALL FIELD POINTS ALSO BY TEND FOR ALL L	000080		
C----JIND = 4, KICK ALL FIELD POINTS ALSO BY TEND FOR THOSE L BEING PRINT	000081		
C----JBCT USFD TO SPACE CALLING OF ENTHALPY TRENDS	000082		
READ(5,1)(TITLE(1),I=1,8)	000083		
WRITE(6,3)(TITLE(1),I=1,8)	000084		
READ(5,2)L2,LMAX,LPRINT,JOUT,JIND,IBUG,INIT,IFIZ,JBCT,JOFB	000085		
WRITE(6,6)L2,LMAX,LPRINT,JOUT,JIND,IBUG,INIT,IFIZ,JBCT,JOFB	000086		
READ(5,4)(SAFE(K),K=1,8)	000087		
READ(5,4)(DAMP(K),K=1,8)	000088		
DO 7 K = 1,8	000089		
7 WRITE(6,8) K,SAFE(K),DAMP(K)	000090		
READ(5,4) TEND, TENB	000091		
WHITE(6,12) TEHD, TEHB	000092		
C READ DATA FROM TEHAI FROM TAPE 3	000093		
C READ STARTING TAPE DATA IN BINARY	000094		
READ (3) NYA, NYB, NYS, NYT, NYA, NYB, NYS, NYT,	000095		
C MCA, MCB, MCE, MCF, MCG, MCH, MCS, MCT, NCA, NCB, NCE, NCF,	000096		
C NCCT, NCH, NCS, NCT,	000097		
F MFA, MFU, MFD, MFE, MFF, MFG, MFR, MFO, MFS, MFT, MFAF,	000098		
F MFANP, MFAT, MFATM, MFATP, MFAF, NFU, NFU, NFE, NFF, NFG, NFH,	000099		
F NFU, NFS, NFT, NFCL,	000100		
H NHAD, NHB, NHG, MHE, MHF, MHG, MHK, MHQ, MHS, MHT, MHN, MHNP,	000101		
H NHAD, NHB, NHG, NHE, NHF, NHG, NHQ, NHG, NHT, NHCL,	000102		
M MMAD, MMF, MMD, MME, MMF, MMG, MMH, MMQ, MMS, MMU, MMV, MMNP,	000103		
M MMAD, MMF, MMD, MME, MMF, MMG, MMH, MMQ, MMS, MMU, MMV, MMNP,	000104		
M NMAD, NMD, NME, NMF, NMG, NMH, NMQ, NMS, NMU, NMV, NMCL,	000105		
T NTAD, MTB, MTD, MTO, MTS, MTT, MTAN, MTANP, NTA, NTR, NTD,	000106		
T NTG,	000107		
S MSL, MST, NSL, NST	000108		
READ (3) LBPA, LBFB, LBFC, LBFD, LBFE, LBFI, LBFI, LBFK, LBFK,	000109		
F LBFL, LBFM, LBFBMAX,	000110		
H LBHA, LBHG, LBHD, LBHE, LBHI, LBHQ, LBHK, LBAL, LBAM,	000111		
H LBHMAX,	000112		
M LBMA, LBMB, LBMC, LBMO, LBME, LBMI, LBQJ, LBQK, LBML, LBMM,	000113		

```

M LBMMAX, 000114
T LHTA, LBTB, LHTC, LBTD, LBTE, LBTI, LHTJ, LBTK, LBTL, LBTM, 000115
T LBIG, LBTR, LBTS, LBTT, LBTMX 000116
READ (3) L 000117
READ (3) ((IYB(J)), J=1,84), (ICR(J), J=1,136), 000118
A ((IDFI(J), IDFM(J), IDFT(J)), J=1,10), (IWFT(J), J=1,128), 000119
B (IWH(J), J=1,66), (IWM(J), J=1,42), (IWT(J), J=1,129), 000120
C ((LINTF(J), LINTH(J), LINTM(J)), J=1,2), ((IWFX(J), IWX(J), 000121
D IWGX(J)), J=1,4), (IAFT(J), JE1,74), (IAH(J), J=1,50), (IAM(J), 000122
E JE1,52), (IAT(J), JE1,32), (IAN(J), JE1,21), 000123
F ((IWFH(J), IWHM(J), IWMT(J)), JE1,8) 000124
READ (3) ((ICO(J), JCO(J)), J = 1,68), ((IFO(J), JFO(J)), 000125
F JE1,296), ((IHO(J), JHO(J)), JE1,200), ((IMO(J), JMO(J)), JE1,208), 000126
G ((ITO(J), JTO(J)), JE1,128) 000127
READ (3) ((DX(J),DY(J)),J=1,6), (RH(J), J=1,9) 000128
READ (3) ((CY(J),SY(J)), J=1,42), (DYB(J), J=1,84), 000129
A (UCD(J), J=1,136), (XYF(J),SCF(J)), J=1,84), 000130
H ((XYH(J),SCH(J)), J=1,44), (XYM(J),SCN(J)), J=1,28), 000131
C ((XYT(J),SCT(J)), JE1,86) 000132
READ (3) (DWF(J), J=1,1512), (DWII(J), J=1,792), 000133
A (DWW(J), J=1,504), (DWT(J), J=1,1548), (AWF(J), J=1,126), 000134
B (AWII(J), J=1,66), (AWM(J), JE1,42), (AWT(J), J=1,129), 000135
C (CWJ(J), J=1,1134), (CWH(J), J=1,594), (CWN(J), J=1,378), 000136
D (CWII(J), J=1,1161), ((DWFX(J),DWHX(J),DWVX(J),DWTX(J)), J=1,108), 000137
E ((AWFH(J), AWHM(J), AWMT(J)), J=1,9), 000138
F ((DWFH(J), DWHM(J), DWMT(J)), JE1,72), 000139
G ((CAFX(J), CWHX(J), CWVX(J)), JE1,36) 000140
READ (3) GAM, HGMU, UNOG, DOGMU, UNNNY, HINF, GMU, UNOGMU 000141
READ (3) UMACH, ALPHA, RBNDY 000142
READ (3) (WY(J), J=1,1134), (WC(J), J=1,867), 000143
A (WF(J), J=1,1443), (WH(J), J=1,1275), (WM(J), J=1,2028), 000144
B (WI(J), J=1,1000), (WSF(J), J=1,1134), (WSH(J), J=1,594), 000145
C (WSM(J), J=1,378), (WST(J), J=1,1161), (WSU(J), J=1,3) 000146
WRITE(6,10) UMACH, GAM, ALPHA 000147
C----USEFUL CONSTANTS 000148
HGPU = 0.5*(GAM + 1.0) 000149
HGPV = 0.5*(HGPU - 1.0) 000150
HGPW = (HGPU - 2.0)/3.0 000151
C----SPECIAL REPAIRS 000152
IF(IFIZ,LG,2) CALL REPAIR 000153
WRITE(6,10) UMACH, GAM, ALPHA 000154
C****TIME STEP CONSTANTS 000155
C----CARTESIAN REGIONS 000156
DO 204 K = 1,6 000157
PRMSH(K) = DX(K)/DY(K) 000158
PRMSH(K) = DY(K)/DX(K) 000159
ATSF(K) = SQRT(1.0 + PRMSH(K)**2) 000160
204 TSN(K) = SAFE(K)*0.5 000161
C----OUTER REGION 000162
DRHO = DX(1) 000163
DTHET = DY(1) 000164
PRMSHY = DRHO / DTHET 000165
ATSFY = SQRT(1.0 + PRMSHY**2) 000166
TSN(1) = 0.5*SAFE(1)*DRHO*RBNDY/DTHET 000167
PRMY1 = 1.0/PRMSHY 000168
C----WEIGHTING FACTORS FOR DIFFUSIVE DAMPING 000169
CDP = DAMPT8 000170

```

HCDP = 0.5*COP	000171
UHD(1,1) = 0.5*(1.0 - HCDP)	000172
UHD(1,2) = 0.5*COP	000173
UHD(1,3) = -0.5*HCDP	000174
UHD(2,1) = 0.5*HCDP	000175
UHD(2,2) = 0.5*(1.0 - COP)	000176
UHD(2,3) = 0.5*HCDP	000177
UHD(3,1) = -0.5*HCDP	000178
UHD(3,2) = 0.5*COP	000179
UHD(3,3) = 0.5*(1.0 - HCDP)	000180
DO 210 KR = 1.6	000181
DCX(1,KR) = 0.5*(1.0 - DAMP(KR))	000182
DCX(2,KR) = 0.25*DAMP(KR)	000183
DCY(1,KR) = 0.5*(1.0 - PRMSHI(KR)**2*DAMP(KR))	000184
210 DCY(2,KR) = DCX(2,KR)*PRMSAI(KR)**2	000185
DO 220 KR = 3.6	000186
LMRTX = DX(KR-1)/DX(KR)	000187
LMRTY = DY(KR-1)/DY(KR)	000188
LBIX(KR) = IFIX(LMRTX + 0.0001)	000189
LBYY(KR) = IFIX(LMRTY + 0.0001)	000190
KDRX = DCX(2,KR-1)/DCX(2,KR) - 1.0	000191
KDRY = DCY(2,KR-1)/DCY(2,KR) - 1.0	000192
DMUX(1,KR) = 1.0	000193
DMUY(1,KR) = 1.0	000194
LL = LBIX(KR)	000195
IF(LL,LT,2) GO TO 215	
DO 214 JB = 2,LL	000196
LN = JB	000197
DMUX(JB,KR) = EN**2*(1.0 + EN*RDRX/BMRTX)	000198
DMUX = 0.5 - DMUX(JB,KR)*(0.5 - DCX(1,KR))	000199
IF(DMUX,GE,0.0) GO TO 214	000200
DMUX(JB,KR) = 0.5/(0.5 - DCX(1,KR))	000201
214 CONTINUE	000202
215 LL = LBYY(KR)	000203
IF(LL,LT,2) GO TO 220	000204
DO 216 JB = 2,LL	000205
LN = JB	000206
DMUY(JB,KR) = EN**2*(1.0 + EN*RDRY/BMRTY)	000207
DMUY = 0.5 - DARRY(JB,KR)*(0.5 - DCYT1,KR))	000208
IF(DMUY,GE,0.0) GO TO 216	000209
DMUY(JB,KR) = 0.5/(0.5 - DCYT1,KR))	000210
216 CONTINUE	000211
220 CONTINUE	000212
C-----CONFIRMATION OF DAMPING CONSTANTS	000213
WRITE(6,14)	000214
DO 224 J = 1,3	000215
224 WRITE(6,4) TDHD(J,I), I = 1,3	000216
WRITE(6,18)	000217
DO 226 KR = 1.6	000218
226 WRITE(6,18) KR, (DCX(J,KR),J=1,2), (DCY(J,KR),J=1,2)	000219
DO 230 KR = 3.6	000220
WRITE(6,20)KR	000221
WRITE(6,22) LBIX(KR), LBYY(KR)	000222
WRITE(6,24)	000223
DO 230 JB = 1,5	000224
230 WRITE(6,4) DMUX(JB,KR), DMUY(JB,KR)	000225
C****MAIN ROUTINE	000226
	000227

```

130 LZ = LZ + 1 000228
C----MAIN LOOP 000229
DO 1310 L=LZ,LMAX 000230
GO TO (140,150,160),JOFB 000231
140 CALL YTRE 000232
150 CALL OOKII 000233
160 CALL SUSOMU(3,MFA,MFT,MFA,NFT,WF,IFO,JFO,1) 000234
CALL HERI(3,LBFMAX,WSF,1) 000235
CALL SUSOMU(4,MHA,MHT,NHA,NHT,WH,IHO,JHO,1) 000236
CALL HERI(4,LHMMAX,WSH,1) 000237
CALL SUSOMU(5,MMA,MMT,NMA,NMT,WM,IMO,JMO,1) 000238
CALL HERI(5,LHMMAX,WSM,1) 000239
CALL JIKAN 000240
CALL SUSOMU(6,MTA,MTT,NTA,NTT,WT,ITO,JTO,2) 000241
CALL HLR(6,LBTMAX,WST,2) 000242
CALL SAKAI 000243
C----OUTPUT AND TERMINATION 000244
IF(JIND,EQ.3) CALL TREND 000245
IF((JIND,EQ.4).AND.(MOD(L,JBLT),EQ.0)) CALL TREND 000246
IF((JOUT,LPRINT),NE.0) GO TO 1310 000247
CALL OUCH(JOUT) 000248
1310 CONTINUE 000249
C----DO ITUM OF TREATMENT OF ONE TIME PLANE 000250
IF((MOD(LMAX,LPRINT),NE.0).OR.(JOUT,NE.2)) CALL OUCH(2) 000251
C 000252
IF(INIT,EO.1) GO TO 1500 000253
C----RITE RESULTS ON TAPE FOR FUTURE USE 000254
C----WRITE NEW STARTING TAPE DATA IN BINARY 000255
  WRITE(4) MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT, 000256
  C NCA, MCN, MCE, MCF, MCG, MCII, MCS, MCT, NCA, NCB, NCE, NCF, 000257
  C NCII, NCH, NCS, NCT, 000258
  F NFH, NFD, NFE, MFF, MFG, MFH, MFO, MFS, MFT, MFAN, 000259
  F NFH, NFD, NFE, NFATP, NFA, NFB, NFU, NFE, NFF, NFG, NFH, 000260
  F NFU, NFS, NFT, NFCL, 000261
  H MHF, MHG, MHH, MHQ, MHS, MHT, MHN, MHNP, 000262
  H NHF, NHG, NHH, NHQ, NHS, NHT, NHCL, 000263
  M MMF, MMG, MMH, MMQ, MMS, MMT, MMAN, MMNP, 000264
  M NMF, NMG, NMH, NMQ, NMS, NMT, NMCL, 000265
  T MTB, MTD, MTQ, MTS, MTI, MTAN, MTAP, NTA, NTB, NTU, 000266
  T NTG, NTS, NTI, NTCL, 000267
  S MSL, MST, NSL, NST, 000268
  WRITE(4) LBFA, LBFB, LBFC, LBFD, LBFE, LBFI, LBFI, LBFK, 000269
  F LBFL, LBFM, LBMAX, 000270
  H LBH, LBHB, LBHC, LBHD, LBHE, LBHI, LBHQ, LBHK, LBHL, LBHM, 000271
  H LBMAX, 000272
  M LBMA, LBMB, LBMC, LBMD, LBME, LAMI, LBNU, LBPK, LBML, LBMM, 000273
  M LBMAX, 000274
  T LBTA, LBTB, LBTC, LBTD, LBTE, LBTT, LBTR, LBTK, LBTC, LBTM, 000275
  T LBTL, LBTR, LBTS, LBTT, LBMAX 000276
  WRITE(4) L 000277
  WRITE(4) (IYR(J), J=1,84), (ICB(J), J=1,136), 000278
  A ((IUFH(J), IDHM(J), IDMT(J)), J=1,10), (INF(J), J=1,126), 000279
  H (I,HI(J), J=1,66), (IWM(J), J=1,42), (IWT(J), J=1,129), 000280
  C ((LINTF(J), LINTH(J), LINTM(J)), J=1,2), ((IWFX(J), IWX(J), 000281
  D IWXA(J), J=1,4), (IAF(J), J=1,74), (IAH(J), J=1,50), (IAM(J), 000282
  E J=1,52), (IAT(J), J=1,32), (IAN(J), J=1,21), 000283
  F ((I,FH(J), IHM(J), IWFT(J)), J=1,8) 000284

```

WRITE(4)	((ICO(J), JOI(J)), J=1,68), ((IFO(J), JFO(J)),	000285
F J=1,296), ((IHO(J), JHO(J)), J=1,200), ((IMO(J), JMO(J)), J=1,208),	000286	
G ((ITO(J), JTO(J)), J=1,128)	000287	
WRITL(4)	((UX(J), DY(J)), J=1,6), (FH(J), J=1,9)	000288
WRITE(4)	((CY(J), SY(J)), J=1,42), (DYB(J), J=1,84),	000289
A ((UCB(J), J=1,136), ((XYF(J), SCF(J)), J=1,84),	000290	
B ((XYH(J), SCH(J)), J=1,44), ((XYM(J), SCM(J)), J=1,28),	000291	
C ((XYT(J), SCT(J)), J=1,86)	000292	
WRITL(4)	(DWF(J), J=1,1512), (DWH(J), J=1,792),	000293
A (DWL(J), J=1,504), (DWY(J), J=1,1548), (AWF(J), J=1,126),	000294	
B (AWH(J), J=1,66), (AWM(J), J=1,42), (AWT(J), J=1,129),	000295	
C (CWF(J), J=1,1134), (CWH(J), J=1,594), (CWN(J), J=1,378),	000296	
D(CWT(J), J=1,1161), ((DWF(X(J), DWH(X(J), DWY(X(J), DWY(T(J)), J=1,108),	000297	
E ((AWFH(J), AWIM(J), AWMT(J)), J=1,9),	000298	
F ((DWFI(J), DWIM(J), DWMT(J)), J=1,72),	000299	
G ((CWFX(J), CWHX(J), CAXX(J)), J=1,36)	000300	
ARITL(4)	GAM, HNU, UNOG, DOGMU, GRNYI, HINF, GMU, UNOGMU	000301
WRITL(4)	UMACH, ALPHA, RHNDY	000302
WRITL(4)	((Y(J), J=1,1134), (WC(J), J=1,867),	000303
A (WF(J), J=1,1443), (WH(J), J=1,1275), (WM(J), J=1,2026),	000304	
B (WI(J), J=1,1008), (WSF(J), J=1,1134), (WSH(J), J=1,594),	000305	
C (WSW(J), J=1,378), (WST(J), J=1,1161), (WSU(J), J=1,3)	000306	
1500 CALL EXIT	000307	
STOP	000308	
C		
1 FORMAT(8A10)	000309	
2 FORMAT(16I5)	000310	
3 FORMAT(20X,8A10)	000311	
4 FORMAT(8E10.3)	000312	
5 FORMAT(/16X,2HLZ,6X,4HMAX,4X,6HLPRINT,6X,4HJOUT,6X,4HJTND,	000313	
1 6X,4HINIT,6X,4HTIZ,6X,JRCT*6X*JOFR*7I0I10)	000314	
8 FORMAT(/16X,1HK, 16X,4HSAFE,16X,4HNAMP,/I10, 2E20.5)	000315	
10 FORMAT(/17X,5HUMACH,15X,3HGA..,15X,5HALPHA,/3E20.5)	000316	
12 FORMAT(/14HE17HALPY PUSH#,F7.4)	000317	
14 FORMAT(/4X,4UHDAMPING FACTORS FOR ODD CFLLS DHD(FP,MP))	000318	
16 FORMAT(/4X,33HDTIFFUSIVE DAMPING FACTORS,DCX,DCY)	000319	
18 FORMAT(/4X,15,10X,2E10.3,10X,2E10.3)	000320	
20 FORMAT(/4X,18HDAMPING AT REGION ,15,11H BOUNDARIES)	000321	
22 FORMAT(/4X,16HCOLUMN OVERLAP =,15,10X,13HROW OVERLAP =,15)	000322	
24 FORMAT(/4X,32HDAMPING MULTIPLIERS DMAX, DMAY)	000323	
8U FORMAT(16I5)	000324	
B2 FORMAT(4E20.13)	000325	
END	000326	
	000327	

```

SUBROUTINE OUCHIJOS
C
COMMON/LIMITS/ MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT,
C NCA, MCN, NCE, NCF, MCG, MCH, MCS, MCT, NCA, NCN, NCE, NCF,
C NCG, NCH, NCS, NCT,          000328
C
F MFA, MFU, MFD, MFE, MFF, MFG, MFH, MFO, MFS, MFT, MFA,          000329
F MFANP, MFAT, MFATM, MFATH, MFA, MFB, MFU, MFE, MFF, MFG, MFH,          000330
F MFANP,          000331
F MFAT,          000332
F MFATH,          000333
F MFU, MFS, MFT, NFCL,          000334
H MMA, MMH, MHD, MHE, MHG, MHH, MHQ, MHS, MHT, MHN, MHTNP,          000335
H MMA, MMH, MHD, MHE, MHG, MHH, MHO, MHS, MHT, MHL,          000336
M MMA, MMH, MHD, MHE, MHG, MHH, MHO, MMS, MMT, MMAN, MMANP,          000337
M MMA, MMH, MHD, MHE, MHG, MHH, MHO, NMH, NMU, NMS, NMT, NMCL,          000338
T MTA, MTB, MTD, MTQ, MTS, MTT, MTA, MTA, MTA, MTA, MTD,          000339
T NTQ, NTS, NTI, NTCL,          000340
S MSL, MST, NSL, NST,          000341
COMMON/PLIMIT/ LBFA, LBFB, LBFC, LBFD, LBFE, LBFI, LBFJ, LBFK,          000342
F LBFL, LBFM, LBEMAX,          000343
H LIMA, LBMB, LBMC, LBMD, LBME, LBMI, LBHQ, LBHK, LBHL, LBHM,          000344
H LBMAX,          000345
M LBMA, LBMB, LBMC, LBMD, LBME, LBMI, LBHQ, LBHK, LBML, LBMM,          000346
M LBMAX,          000347
T LBIA, LBIB, LBIC, LBID, LBIE, LBII, LBII, LBTK, LBTL, LBTM,          000348
T LBIC, LBIR, LBIS, LBIT, LBIMAX,          000349
COMMON/GEOM/F SY(42), CY(42), DYB(2,42), DCB(2,4,17),          000350
A XYF(2,42), SCF(2,42), XYH(2,22), SCH(2,22),          000351
B XYI(2,14), SCM(2,14), XYT(2,43), SCT(2,43),          000352
COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, GDYMI, HINF, GMU, UNOGMU,          000353
COMMON/DURA/ L,          000354
COMMON WY(3,42,9), WC(3,17,17), WF(3,37,13), WI(3,25,17),          000355
A WM(3,26,26), WT(3,16,21), WSF(3,9,42), WSH(3,9,22), WSM(3,9,14),          000356
B WSI(3,9,43), WSU(3),          000357
C
      WRITE(6,51) L,          000358
      GO TO (100-130,140,160),JO,          000359
C----CONSERVATION VARIABLES,          000360
C----OUTER FIELD,          000361
100 WRITE(6,52),          000362
      WRITE(6,53),          000363
      WRITE(6,51) L,          000364
      CALL PUTOUT(1,WY,MYT,NYT),          000365
C----COARSE MESH,          000366
      WRITE(6,55),          000367
      WRITE(6,51) L,          000368
      CALL PUTOUT(1,WC,MCT,NCT),          000369
C----FINE MESH,          000370
      WRITE(6,57),          000371
      WRITE(6,51) L,          000372
      CALL PUTOUT(1,WF,MFT,NFT),          000373
C----OUTER NOSE REGION,          000374
      WRITE(6,69),          000375
      WRITE(6,51) L,          000376
      CALL PUTOUT(1,WH,MHT,NHT),          000377
C----INTERMEDIATE NOSE REGION,          000378
      WRITE(6,70),          000379
      WRITE(6,51) L,          000380
      CALL PUTOUT(1,WM,MMT,NMT),          000381
      WRITE(6,51) L,          000382
      CALL PUTOUT(1,WM,MMT,NMT),          000383

```

C---- INNER NOSE REGION	000384
WRITE(6,59)	000385
WRITE(6,51) L	000386
CALL PUTOUT(1,WT,MYT,NYT)	000387
C---- PHYSICAL VARIABLES	000388
13U WRITE(6,61)	000389
C---- OUTER FIELD	000390
WRITE(6,53)	000391
WRITE(6,51) L	000392
CALL PUTOUT(2,WY,MYT,NYT)	000393
C---- COARSE MESH	000394
WRITE(6,55)	000395
WRITE(6,51) L	000396
CALL PUTOUT(2,WC,MCT,NCT)	000397
C---- FINE MESH	000398
14U WRITE(6,57)	000399
WRITE(6,51) L	000400
CALL PUTOUT(2,WF,MFT,NFT)	000401
C---- OUTER NOSE REGION	000402
WRITE(6,69)	000403
WRITE(6,51) L	000404
CALL PUTOUT(2,WH,MHT,NHT)	000405
C---- INTERMEDIATE NOSE REGION	000406
WRITE(6,70)	000407
WRITE(6,51) L	000408
CALL PUTOUT(2,WM,MMT,NMT)	000409
C---- INNER NOSE REGION	000410
WRITE(6,59)	000411
WRITE(6,51) L	000412
CALL PUTOUT(2,WT,MYT,NYT)	000413
C---- AIRFOIL SURFACE	000414
16U WRITE(6,63)	000415
C---- NOSE REGION HORIZONTAL CUTS	000416
WRITE(6,59)	000417
WRITE(6,51) L	000418
WRITE(6,68)	000419
WRITE(6,73)	000420
CALL AIROUT(LBT1,LBTM,WST,LBTMAX,XYT)	000421
WRITE(6,73)	000422
CALL AIROUT(LBTQ,LBT1,WST,LBTMAX,XYT)	000423
WRITE(6,67)	000424
WRITE(6,73)	000425
CALL AIROUT(LBTA,LBTE,WST,LBTMAX,XYT)	000426
C---- INTERMEDIATE NOSE REGION VERTICAL CUTS	000427
WRITE(6,70)	000428
WRITE(6,51) L	000429
WRITE(6,67)	000430
WRITE(6,73)	000431
CALL AIROUT(LBMA,LBME,WSM,LBMMAX,XYM)	000432
WRITE(6,68)	000433
WRITE(6,73)	000434
CALL AIROUT(LBMI,LHMM,WSM,LBMMAX,XYM)	000435
C---- OUTER NOSE REGION VERTICAL CUTS	000436
WRITE(6,69)	000437
WRITE(6,51) L	000438
WRITE(6,67)	000439
WRITE(6,73)	000440

```

CALL AIFROUT(LBHA,LBHE,WSH,LBHIMAX,XYH) 000441
WRITE(6,68)                                000442
WRITE(6,73)                                000443
CALL AIFOUT(LHMI,LBHMI,WSH,LBHIMAX,XYH) 000444
C----FINE MESH VERTICAL CUTS                000445
WRITE(6,57)                                000446
WRITE(6,51) L                               000447
WRITE(6,67)                                000448
WRITE(6,73)                                000449
CALL AIFOUT(LBFA,LBF,WSF,LBFMAX,XYF) 000450
WRITE(6,68)                                000451
WRITE(6,73)                                000452
CALL AIFOUT(LHFI,LBFM,WSF,LBFMAX,XYF) 000453
RETURN                                     000454
C                                         000455
11 FORMAT(415,5E10.3)                       000456
12 FORMAT(15,I10,6E20.6)                     000457
51 FORMAT(/4X,2HLE,15)                       000458
52 FORMAT(/4X,2HCOMSERVATION VARIABLES)    000459
53 FORMAT(/4X,1IOUTER FIELD)                 000460
54 FORMAT(/3X,2HM,18X,2HM,17X,4HW(1),16X,4HW(2),16X,4HW(3)) 000461
55 FORMAT(/4X,2I(CARSE CARTESIAN MESH)      000462
57 FORMAT(/4X,2I(FINE CARTESIAN MESH)        000463
59 FORMAT(/4X,1I(INNER NOSE REGION)          000464
61 FORMAT(/4X,18I(PHYSICAL VARIABLES)        000465
62 FORMAT(/3X,2HM,18X,2HM,12X,6HPRESSURE,13X,7HDENSITY,18X,2HUX,18X, 000466
12HUY,9X,11H(MACH NUMBER,9X,12HTOT.ENTHALPY) 000467
63 FORMAT(/4X,16I(AIRFOIL SURFACES)         000468
64 FORMAT(/4X,36IHORIZONTAL CUTS IN INNER NOSE REGION) 000469
65 FORMAT(/4X,34HVERTICAL CUTS IN INNER NOSE REGION) 000470
66 I FORMAT(/4X,33HVERTICAL CUTS IN FINE MESH REGION) 000471
67 FORMAT(/4X,13HUPPER SURFACE)               000472
68 FORMAT(/4X,13HLOWER SURFACE)               000473
69 FORMAT(/4X,17HOUTER NOSE REGION)           000474
70 FORMAT(/10X,24HINTERMEDIATE NOSE REGION) 000475
71 FORMAT(1H0,4X,*VERTICAL CUTS*)            000476
72 FORMAT(/3X,2HM,18X,2HM,12X,6HPRESSURE,13X,7HDENSITY,18X,2HUT,18X, 000477
12HUT,9X,11H(MACH NUMBER,9X,12HTOT.ENTHALPY) 000478
73 FORMAT(/3X,2HM,5X,1HX,9X,1HY,9X,1HP,7X,3HRHO,5X,SHU TAN,5X, 000479
A SHU HOP,5X,8HMACH NO.,2X,10HTOT.ENTHL.) 000480
END                                         000481

```

```

SUBROUTINE PUTOUT(J,W,MAX,NAX)          000482
COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, GDMU, RINF, GMU, UNOGMU 000483
DIMENSION W(3,MAX,NAX)                 000484
N2 = NAX                               000485
M2 = MAX                               000486
GO TO 1,20,J
1 DO 10 N = I,N2                      000487
  WRITE(6,54)                           000488
  10 DO 11 M = I,M2                   000489
    11 WRITL(6,12) M,I,(W(K,M,N),K=1,3) 000490
    54 FORMAT(13X,1HM,9X,1HII,18X,4HW(1),16X,4HW(2),16X,4HW(3)) 000491
    12 FORMAT(15,I10,6E20.6)              000492
    RETURN                               000493
20 DO 30 N = I,N2                      000494
  WRITL(6,62)                          000495
  30 DO 31 M=1,M2                     000496
    31 NH0 = W(3,M,N)                  000497
    IF(IH0) 21,21,22                  000498
    21 WRITL(6,12) M,I,(W(K,M,N), K = 1,3) T 000499
    RH0 = 0.010101                     T 000500
    22 R = RH0*GAM                     T 000501
    K = 1.0/RHO                       000502
    UX = W(1,M,I)*R                  000503
    UY = W(2,M,I)*R                  000504
    USQ = UX**2 + UY**2               000505
    AM = SQR(T050)*R**3*HGMU        000506
    HTOT = 2.5*R**K + 0.5*USQ       000507
    30 WRITL(6,12) M,I,P,RH0,OX,UY,AM,HTOT 000508
    02 FORMAT(13X,2HM,8X,2HM,12X,BPRESSURE,13X,7HDENSITY,18X,2HUUX,18X, 000509
    12HUY,9X,11H'ACH NUMBERP,9X,12HTOT,ENTHALPY) 000510
    RETURN                               000511
  END                                   000512
  END                                   000513

```

```

SUBROUTINE AIROUT (R1, M2, W, MAX, XY)          000514
COMMON/GAS/ GAM, HGMO, UNOG, DOGMU, QDYN1, HINF, GMU, UNOGMU   000515
DIMENSION W(3,4,MAX), XY(2,MAX)                 000516
K = 2                                              000517
DO 15 M=M1,M2                                     000518
RHO = W(3,K,M)
IF(RHO) 1,1,2                                     000519
1 WRITE(6,12) M, (W(J,K,M), J = 1,3)           T 000520
RHO = U_010101                                     T 000521
P = RHO**GAM                                      T 000522
K = 1.0 / RHO                                      T 000523
UX = W(1,K,M)*K                                    000524
UY = W(2,K,M)*K                                    000525
C---UT POSITIVE IS OUT FROM THE SURFACE, UT POSITIVE IN CLOCKWISE DIREC 000526
3 USU = UX**2 + UY**2
AM = SQRT(USU)*R**HGMO                           000528
HTOT = 2.5*P*R + 0.5*USU                         000529
15 WRITE(6,12) M, XY(1,M), XY(2,M), P, RHO, UX, UY, AM, HTOT      000530
1Z FORMAT (15.8F10.5)                             000531
RETURN                                         000532
END                                           000533
                                          000534

```

SUBROUTINE YTRE	
C	000535
COMMON/LIMITS/ MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT,	000536
C NCA, MCB, MCE, MCF, MCG, MCH, MCS, MCT, NCA, NCB, NCE, NCF,	000537
C NCG, NCH, NCS, NCT,	000538
F MFA, MFB, MFD, MFE, MFF, MFG, MFH, M_Q, MFS, MFT, MFAN,	000539
F M_FNP, M_FAT, M_FATM, M_FATP, MFA, MFB, MFU, MFE, MFF, MFG, MFH,	000540
F M_FU, MFS, MFT, MFCI,	000541
H MH1, MH2, MH3, MH4, MH5, MH6, MH7, MH8, MH9, MH10, MH11P,	000542
H NH1, NH2, NH3, NH4, NH5, NH6, NH7, NH8, NH9, NH10, NH11, NH12,	000543
M MM1, MM2, MM3, MM4, MM5, MM6, MM7, MM8, MM9, MM10, MM11P,	000544
M NM1, NM2, NM3, NM4, NM5, NM6, NM7, NM8, NM9, NM10, NM11P,	000545
M NM11, NM12, NM13, NM14, NM15, NM16, NM17, NM18, NM19, NM20,	000546
T MT1, MT2, MT3, MT4, MT5, MT6, MT7, MT8, MT9, MT10, MT11P, NT1,	000547
T NT1, NT2, NT3, NT4, NT5, NT6, NT7, NT8, NT9, NT10, NT11P, NT12,	000548
S MSL, MST, NSL, NST	000549
COMMON/ZMESII/ DX(6), DY(6), RH(9)	000550
COMMON/ GEO-F/ SY(42), CY(42), DYC(2,4,22), DCB(2,4,22),	000551
A XY(2,42), SCF(2,42), XYH(2,22), SCH(2,22),	000552
B XY(2,14), SC(2,14), XY(2,43), SCT(2,43)	000553
COMMON/CUN1/JR, MK, NR, UDX, DTDY, F(3,5,26), G(3,5,26),	000554
I FP(3,3,3), GP(3,3,3), WAD(3,3,26)	000555
COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, QDYN1, HINF, GMU, UNOGMU	000556
COMMON/TOC/PRMSHY, ATSFY, HGPU, HGPV, HGPW, ATSF(6), TSN(6), PRMSH(6),	000557
I PRMSH(6), PR-YI, CFTT	000558
COMMON/PET/DCJX(3,3), DCX(3,6), DCY(3,6), DHU(3,3), DN8X(5,6),	000559
A DMRY(5,6), LHIX(6), L3TY(6)	000560
COMMON WY(3,42,9), WC(3,17,17), WF(3,37,13), WH(3,25,17),	000561
A WM(3,26,26), WT(3,16,21), WSF(3,9,42), WSH(3,9,22), WSM(3,9,14),	000562
B WS(3,9,43), WSU(3)	000563
C DIMENSION WW(3)	000564
C-----YTRE FIELD	000565
RR = 1	000566
M = 1	000567
J = 1	000568
C-----F AND G CONVERSION	000569
404 UC 404 N = 1, NYT	000570
P = UT, OC*WY(3,M,N)**GAM	000571
UXP = WY(1,M,N)/WY(3,M,N)	000572
UYP = WY(2,M,N)/WY(3,M,N)	000573
F(I,J,N) = -WY(1,M,N)*UXP - P	000574
F(2,J,N) = -WY(1,M,N)*UYP	000575
F(3,J,N) = -WY(1,M,N)	000576
G(1,J,N) = F(2,J,N)	000577
G(2,J,N) = -WY(2,M,N)*UYP - P	000578
404 G(3,J,N) = -WY(2,M,N)	000579
GO TO (406,408,410), J	000580
406 M = 6	000581
J = 2	000582
GO TO 402	000583
406 M = M + 1	000584
J = 3	000585
GO TO 402	000586
410 M = M - 1	000587
MM2 = M - 2	000588
DO 660 M = 2, NYS	000589
	000590

```

C----ALLOWABLE STEP
KUX = AHS(-WY(I,M,N)+SY(M) + WY(2,M,N)*CY(M))*PRMSHY 000591
KUY = ABS(WY(I,M,N)*CY(M) + WY(2,M,N)*SY(M)) 000592
RNU = WY(3,M,N) - 1.0 000593
RA = ATSFY*(1.0 + RNU*HGPU*(1.0 + RMU*HGPV*(1.0 + RMU*HGPW))) 000594
CFIT = TSN(1)*WY(3,M,N)/(RUX + RUY + RA) 000595
000596
C---ADVANCED SOLUTION AT A POINT
NM2 = N - 2 000597
000598
C----HALF STEPS
KTS = 1 000599
IA = 2 000600
JA = 1 000601
000602
000603
000604
000605
000606
000607
000608
SGH = SGT = SFK = SFT = 0.0 000609
000610
JL = 1,3 000611
SGT = SGT + CJX(IA,JL)*G(K,JL,NN) 000612
SFT = SFT + CJX(IA,JL)*F(K,JL,NN) 000613
JN = NM2 + JL 000614
SGH = SGH - CJX(JA,JL)*G(K,IA,JN) 000615
SFK = SFK - CJX(JA,JL)*F(K,IA,JN) 000616
C--" VALUE OF HALF STEP LEVEL
WW(K) = WY(K,MM,NN) + CFIT*(-RH(N)*PRMY)*(SFR*CY(M) + SGN*SY(M)) 000617
A = SFT*SY(M) + SGT*CY(M) 000618
GO TO 640 000619
C--ASSIGNMENT OF POINT AT INFINITY
WW(K) = WY(K,MM,NN) 000620
000621
640 CONTINUE
PA = UNOG*WW(3)**GAM 000622
K = 1.0/WW(3) 000623
U1A = WW(1)*R 000624
U2A = WW(2)*R 000625
FP(1,IA,JA) = -WW(1)*U1A - PA 000626
FP(2,IA,JA) = -WW(1)*U2A 000627
FP(3,IA,JA) = -WW(1) 000628
GP(1,IA,JA) = FP(2,IA,JA) 000629
GP(2,IA,JA) = -WW(2)*U2A - PA 000630
GP(3,IA,JA) = -WW(2) 000631
C--OTHER AUXILIARY POINTS
GO TO (642,644,646,650), KTS 000632
000633
642 KTS = 2 000634
JA = 3 000635
GO TO 602 000636
000637
000638
000639
000640
000641
000642
000643
000644
000645
000646
000647
C---SECOND STEP
000648
000649
000650
000651
000652
000653
000654
000655
000656
000657
000658
000659
000660
000661
000662
000663
000664
000665
000666
000667
000668
000669
000670
000671
000672
000673
000674
000675
000676
000677
000678
000679
000680
000681
000682
000683
000684
000685
000686
000687
000688
000689
000690
000691
000692
000693
000694
000695
000696
000697
000698
000699
000700
000701
000702
000703
000704
000705
000706
000707
000708
000709
000710
000711
000712
000713
000714
000715
000716
000717
000718
000719
000720
000721
000722
000723
000724
000725
000726
000727
000728
000729
000730
000731
000732
000733
000734
000735
000736
000737
000738
000739
000740
000741
000742
000743
000744
000745
000746
000747
000748
000749
000750
000751
000752
000753
000754
000755
000756
000757
000758
000759
000760
000761
000762
000763
000764
000765
000766
000767
000768
000769
000770
000771
000772
000773
000774
000775
000776
000777
000778
000779
000780
000781
000782
000783
000784
000785
000786
000787
000788
000789
000790
000791
000792
000793
000794
000795
000796
000797
000798
000799
000800
000801
000802
000803
000804
000805
000806
000807
000808
000809
000810
000811
000812
000813
000814
000815
000816
000817
000818
000819
000820
000821
000822
000823
000824
000825
000826
000827
000828
000829
000830
000831
000832
000833
000834
000835
000836
000837
000838
000839
000840
000841
000842
000843
000844
000845
000846
000847
000848
000849
000850
000851
000852
000853
000854
000855
000856
000857
000858
000859
000860
000861
000862
000863
000864
000865
000866
000867
000868
000869
000870
000871
000872
000873
000874
000875
000876
000877
000878
000879
000880
000881
000882
000883
000884
000885
000886
000887
000888
000889
000890
000891
000892
000893
000894
000895
000896
000897
000898
000899
000900
000901
000902
000903
000904
000905
000906
000907
000908
000909
000910
000911
000912
000913
000914
000915
000916
000917
000918
000919
000920
000921
000922
000923
000924
000925
000926
000927
000928
000929
000930
000931
000932
000933
000934
000935
000936
000937
000938
000939
000940
000941
000942
000943
000944
000945
000946
000947
000948
000949
000950
000951
000952
000953
000954
000955
000956
000957
000958
000959
000960
000961
000962
000963
000964
000965
000966
000967
000968
000969
000970
000971
000972
000973
000974
000975
000976
000977
000978
000979
000980
000981
000982
000983
000984
000985
000986
000987
000988
000989
000990
000991
000992
000993
000994
000995
000996
000997
000998
000999
000999

```

$C + (GP(K,2,1) - GP(K,2+3)) * SY(M))$	$-(FP(K,3,2) - FP(K+1,2)) * SY(M)$	000648
$D + (GP(K,3,2) - GP(K+1,2)) * CY(M))$		000649
660 CONTINUE		000650
C-----COLUMN SHIFTING		000651
IF ($M = NYS$) 422,426,426		000652
422 DO 424 J = 1,2		000653
DO 424 N = 1,NYT		000654
DO 424 K = 1,3		000655
$F(K,J,N) = F(K,J+1,N)$		000656
424 G(K,J,N) = G(K,J+1,N)		000657
426 IF ($M = 2$) 432,432,428		000658
428 DO 430 N = 2,NYS		000659
DO 430 K = 1,3		000660
430 WY(K,M-1,N) = WA0(K,1,N)		000661
432 DO 434 N = 2,NYS		000662
DO 434 K = 1,3		000663
434 WAU(K,1,N) = WAD(K,2,N)		000664
IF ($M = NYS$) 440,436,436		000665
436 DO 438 N = 2,NYS		000666
DO 438 K = 1,3		000667
438 WY(K,M,N) = WAU(K,2,N)		000668
RETURN		000669
440 M = M + 1		000670
GO TO 408		000671
END		000672

SUBROUTINE DOKII
 C-----COARSE CARTESIAN MESH
 C
 COMMON/LIMITS/ MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT,
 C MCA, MCN, NCE, NCF, MCN, MCH, MCS, MCT, NCA, NCB, NCE, NCF,
 C NCG, NCH, NCS, NCT,
 F MFA, MFU, MFD, MFE, MFF, MFG, MFO, MFS, MFT, MFAN,
 F MFANP, MFAT, MFATM, MFATH, MFA, NFH, NFD, NFE, NFF, NFG, NPH,
 F NFU, NFS, NFT, NFCL,
 H NHX, NHU, NHD, NHE, NHF, NHG, NHH, NHQ, NHS, NHT, MHAN,
 H NHX, NHU, NHD, NHE, NHF, NHG, NHH, NHQ, NHS, NHT, NHCL,
 M MMX, MMU, MMH, MME, MMF, MMG, MMH, MMO, MMS, MMF, MMAN, MMANP,
 M NMX, NMU, NMH, NME, NMF, NMG, NMH, NMO, NMS, NMT, NMCL,
 I MTX, MTU, MTD, MTG, MTS, MTT, MTAN, MTANP, MTA, NTA, NTD,
 T NTU, NTS, NTT, NTCL,
 S MSL, MST, NSL, NST
 COMMON/GHS/ GAH, HGHU, UNOG, DOGMU, GDYMI, HINF, GMU, UNOGMU
 COMMON/OHLM/ ICO(4,17), JCU(4,17), IFO(8,37), JFU(8,37),
 A IHU(8,25), JIO(8,25), IMO(8,26), IT0(8,16), JT0(8,16)
 COMMON/TUC/PRNSHY, ATSFY, HGPU, HGPV, HGPW, ATSF(6), TSN(6), PAMSH(6),
 1 PRMSH(6), PRMYI, CFIT
 COMMON/CUN/JR, MR, NR, DTDX, DTOY, F(3,5,26), G(3,5,26),
 1 FP(3,3,3), GP(3,3,3), WAD(3,3,26)
 COMMON/PLT/DCX(3,3), DCY(3,6), DHU(3,3), DMBX(5,6),
 A DMBY(5,6), LHIX(6), LUUY(6)
 COMMON KY(3,42,9), KC(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 WS1(3,9,43), WSU(3)
 DIMENSION WH(3)

C
 KR = 2
 UCP = DCX(1,KR) + DCY(1,KR)
 M = MCA
 J = 1
 C-----F AND G CONVERSIONS
 5J2 DO 504 N = NCA,NLT
 F = 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 TLI = ICO(KK,M)

```

NL2 = IC01(KK+1,M) - 1 000729
JCT = JC01(KK,M) 000730
KK = KK + 1 000731
IF(NL2 = NCS) 516,514,514 000732
514 NL2 = NCS 000733
KOT = 2 000734
516 IF(JCT)520,520,518 000735
518 DO 440 N = NL1,NL2 000736
C----ALLOWABLE STEP 000737
RNU = WC(3,M,N) - 1.0 000738
RA = (1.0 + RNU*HGPU*(1.0 + RNU*HGPV*(1.0 + RNU*HGPW)))*ATSF(KR) 000739
DTDY = TSM(KR)*WC(3,M,N)/(AB5(WC(1,M,N))*PRMSH(KR)) 000740
A + AB5(WC(2,M,1)) + RA 000741
DTUX = PRMSH(KR)*DTDY 000742
NM2 = N - 2 000743
C----HALF STEPS
JA = 1 000744
MM = M 000745
NN = N - 1 000746
402 DO 404 K = 1,3 000747
404 w(k) = wC(K,MM,NN) + DTDX*CJX(2,3)*(F(K,3,NN) - F(K,1,NN)) 000748
A + DTDX*(CJX(JA,1)+G(K,2,N-1) + CJX(JA,2)*G(K,2,N)) 000749
B + CJX(JA,3)*G(K,2,N+1)) 000750
RI = 1.0/w(3) 000751
GP(1,2,JA) = -wW(1)*wW(2)*RI 000752
GP(2,2,JA) = -wW(2)**2*RI - UNOG*wW(3)**GAM 000753
GP(3,2,JA) = -wW(2) 000754
IF(JA = 1) 406,406,410 000755
406 JA = 3 000756
NN = N + 1 000757
GO TO 402 000758
410 NN = N 000759
MM = M - 1 000760
IA = 1 000761
412 DO 414 K = 1,3 000762
414 w(k) = wC(K,M,NN) + DTDX*(CJX(IA,1)+F(K,1,NN)) 000763
A + CJX(IA,2)*F(K,2,NN) + CJX(IA,3)*F(K,3,NN)) 000764
B + DTDX*CJX(2,3)*(G(K,2,N+1) - G(K,2,N-1)) 000765
RI = 1.0/w(3) 000766
FP(1,IA,2) = -wW(1)**2*RI - UNOG*wW(3)**GAM 000767
FP(2,IA,2) = -wW(1)*wW(2)*RI 000768
FP(3,IA,2) = -wW(1) 000769
IF(IA = 1) 416,416,430 000770
416 IA = 3 000771
MM = M + 1 000772
GO TO 412 000773
C----SECOND STEP
430 DO 432 K = 1,3 000774
432 wAU(K,2,N) = DCP + wC(K,M,N) + DCX(2,KR)*(WC(K,M+1,N) + wC(K,M-1,N)) 000775
A + DCY(2,KR)*(wC(K,M,N+1) + wC(K,M,N-1)) + DTDX*(FP(K,3,2) - FP(K,1,2)) 000777
B + DTDX*(GPTK,2,3) - GP(K,2,1)) 000778
440 CONTINUE 000779
GO TO (512,530),KOT 000780
520 DO 522 N = NL1,NL2 000781
DO 522 K = 1,3 000782
522 wAU(K,2,N) = WC(K,M,N) 000783
GO TO (512,530),KOT 000784

```

C----COLUMN SHIFTING		000786
530	IF(M = MCS)	000787
532	DO 534 N = NCA,NCT	000788
	DO 534 K = 1,3	000789
	DO 534 JL = 1,2	000790
	G(K,JL,N) = G(K,JL+1,N)	000791
	534 F(K,JL,N) = F(K,JL+1,N)	000792
	536 IF(M = MCH)	000793
	538 DO 540 N = NCB,NCS	000794
	DO 540 K = 1,3	000795
	540 WC(K,M-1,I,) = WAD(K,1,N)	000796
	542 DO 544 N = NCB,NCS	000797
	DO 544 K = 1,3	000798
	544 WAD(K,1,N) = WAD(K,2,N)	000799
	1F(M-MCS)546,548,548	000800
	546 M = M + 1	000801
	GO TO 508	000802
	548 DO 550 N = NCH,NCS	000803
	DO 550 K = 1,3	000804
	550 WC(K,M,N) = WAD(K,2,N)	000805
	RETURN	000806
	END	000807

```

SUBROUTINE SUSUMU(RR,MRA,MRT,NRA,NRT,W,IRO,JRO,ITSMDJ          000906
C---ADVANCES SOLUTION IN RECTANGULAR "CARTESIAN FIELD BY           000907
C---LAX TYPE SCHEME, CELL TYPES DETERMINED BY IRO, JRO ARRAYS.       000908
C   LIGHT DAMPING ON ALL CENTERED DIFFERENCES, RUN L33 FF             000909
C
COMMON/TOC/PRMSHY,ATSFY,HGPU,HGPV,HGPW,ATSF(6),TSN(6),PRMSH(6),    000910
1 PRMSH(6), PRMYI, CFTT                                           000911
COMMON/OPLM/ ICO(4,17), JCO(4,17), IF0(8,37), JF0(8,37),            000912
A IH0(8,25), JH0(8,25), IM0(8,26), JM0(8,26), IT0(8,16), JT0(8,16)  000913
COMMON/GAS/ GAM, HGMU, UNUG, DOGMU, QDYN, HINF, GMU, UNOGMU        000914
COMMON/COM/ JR, MR, NR, DTDX, DTDY, F(3,5,26), G(3,5,26),            000915
1 FP(3,3,3), GP(3,3,3), WAD(3,3,26)                                000916
COMMON/PEN/ CU(3,3), DCX(3,6), DCY(3,6), DHD(3,3), DMBX(5,6),      000917
A DMHSY(5,6), LBIX(6), LHLY(6)                                     000918
C
DIMENSION W(3,MRT,NRT), IRO( 8,MRT), JRO( 8,MRT),WW(3)              000919
C
MX = MRT
NX = NRT
MRB = MPA + 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
010 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)
U(1,J,N) = F(2,J,N)
U(2,J,N) = -W(2,M,N)*2*RI - P
014 U(3,J,N) = -W(2,M,N)
GO TO (609,620,630,632,668), J
C---SECOND COLUMN OF F
620 M = MRB
J = 3
GO TO 610
C---SUBSEQUENT COLUMNS
030 M = M + 1
J = 4
NLL = NPA
NUL = NRT
GO TO 610
032 M = M - 1
033 KK = 1
KOT = 1
640 NL1 = IRO(KK,M)
NL2 = IPU(KK+1,M) - 1
JHC = JPO(KK,M)
JDR = JHC/100

```

JCT = JHC - 100*JDR	000962
KK = KK + 1	000963
IF(NL2 = NRS) 644,642,642	000964
642 NL2 = NRS	000965
NOT = 2	000966
644 IF(JCT) 680,680,645	000967
645 IF(JCT-9) 646,646,657	000968
646 GO TO 1650,651,652,656,657,658,662,663,664),JCT	000969
C----TYPE 1 CELL	000970
654 J = 5	000971
M = M + 2	000972
ASSIGN 110 TO JSWCH	000973
GO TO 654	000974
C----TYPE 2 CELL	000975
651 ASSIGN 120 TO JSWCH	000976
GO TO 100	000977
C----TYPE 3 CELL	000978
652 J = 1	000979
M = M - 2	000980
ASSIGN 130 TO JSWCH	000981
654 NLL = NL1	000982
NUL = NL2 + 2	000983
GO TO 610	000984
C----TYPE 4 CELL	000985
656 J = 5	000986
M = M + 2	000987
ASSIGN 140 TO JSWCH	000988
GO TO 660	000989
C----TYPE 5 CELL	000990
657 ASSIGN 150 TO JSWCH	000991
GO TO 100	000992
C----TYPE 6 CELL	000993
658 J = 1	000994
M = M - 2	000995
ASSIGN 160 TO JSWCH	000996
660 NLL = NL1 - 1	000997
NUL = NL2 + 1	000998
GO TO 610	000999
C----TYPE 7 CELL	001000
662 J = 5	001001
M = M + 2	001002
ASSIGN 170 TO JSWCH	001003
GO TO 660	001004
C----TYPE 8 CELL	001005
663 ASSIGN 180 TO JSWCH	001006
GO TO 100	001007
C----TYPE 9 CELL	001008
664 J = 1	001009
M = M - 2	001010
ASSIGN 190 TO JSWCH	001011
666 NLL = NL1 - 2	001012
NUL = NL2	001013
GO TO 610	001014
668 M = M - 2	001015
GO TO 100	001016
669 M = M + 2	001017
C----ADVANCE SOLUTION OF POINTS IN COLUMN HAVING SIMILAR TYPE CELLS	001018

```

C THESE VALUES FOR MR,NR,JR WORK ONLY FOR TYPES 5,10,11,12,13 CELLS 001019
100 MR = M - 2 001020
JR = 1 001021
C----NEED FOR EXTRA DAMPING NEAR FIELD OUTER BOUNDARIES 001022
1F(JCT-9) 300,300,301 001023
300 GO TO (105,301,105,301,301,301,105,301,105),JCT 001024
301 1F(JDR) 309,309,302 001025
302 GO TO (304,308,306,304,309,306,304,308,306),JDR 001026
C----LEFT BOUNDARY COLUMNS 001027
304 JDMX = LDIX(KR) + NRA - M + 1 001028
305 UMBXT = DMBX(JDMX,KR) 001029
GO TO 105 001030
C----RIGHT BOUNDARY COLUMNS 001031
306 JDMX = M - NRT + LUIX(KR) + 1 001032
GO TO 305 001033
C----INTERMEDIATE COLUMNS 001034
308 UMBXT = 1.0 001035
GO TO 105 001036
C----CENTRAL FIELD 001037
309 DCX = DCX(1,KR) 001038
DCY = DCY(1,KR) 001039
DUAE = DCX(2,KR) 001040
DUYE = DCY(2,KR) 001041
JDR = 5 001042
105 DO 200 N = NL1,NL2 001043
C----ALLOWABLE STEP 001044
GO TO (318,319),ITSMD 001045
318 RMU = W(3,M,N) - 1.0 001046
RA = (1.0 + RMU*HGPU*(1.0 + RMU*HGPV*(1.0 + RMU*HGPW)))*ATSF(KR) 001047
DTUY = TSII(KR)*W(3,M,N)/(ABS(W(1,M,N))+PRMSH(KR)+ABS(W(2,M,N))) 001048
A + RA 001049
DTUX = PRMSH(KR)*DTUY 001050
C THESE VALUES FOR MR,NP,JR WORK ONLY FOR TYPES 5,10,11,12,13 CELLS 001051
319 NK = N - 2 001052
GO TO (109,320,109,320,320,320,109,320,109,320,320,320,320,320),JCT 001053
320 GO TO (321,321,321,324,109,324,327,327,327,327),JDR 001054
C----LOTUM POUNDARY ROWS 001055
321 JUNY = LBIY(KR) + NRA - N + 1 001056
322 UMBYT = UMBY(JUNY,KR) 001057
GO TO 330 001058
C----TOP BOUNDARY ROWS 001059
327 JDNY = N - NRT + LBIY(KR) + 1 001060
GO TO 322 001061
C----INTERMEDIATE ROWS 001062
324 UMBYT = 1.0 001063
C----DAMPING ALTERATIONS 001064
330 DMULI = AMAX1(UMBXT,UMBYT) 001065
DCXA = 0.5 - DMULI*(0.5 - DCX(1,KR)) 001066
333 DCY = 0.5 - DMULI*(0.5 - DCY(1,KR)) 001067
335 DUAE = DMULI*DCX(2,KR) 001068
DUYE = DMULI*DCY(2,KR) 001069
103 GO TO JSWCH 001070
C----TYPE I CELL, POINT IS IN LOWER LEFT CORNER OF ARRAY 001071
110 CALL WOAP(1,1,W,MX,NX) 001072
CALL WOAP(1,3,W,MX,NX) 001073
CALL WOAP(3,1,W,MX,NX) 001074
DO 114 K = 1,3 001075

```

```

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 + DTDX*(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 + DTDX*(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(1,2,w,MX,NX)
CALL WOAP(1,3,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 + DTDX*(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 + DTDX*(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(1,3,w,MX,NX)
CALL WOAP(3,2,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 + DTDX*(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          001133
174 W(AU(K,3,N)) = 2.0*DHD(1,1)*W(K,M,N)          001134
A + DHD(1,2)*(W(K,M+1,N) + W(K,M,N-1))          001135
B + DHD(1,3)*(W(K,M+2,N) + W(K,M,N-2))          001136
C + DTDX*(FP(K,3,3) - FP(K,1,3))                001137
D + DTOY*(GP(K,1,3) - GP(K,1,1))                001138
GO TO 200          001139
C----TYPE 6 CELL, POINT IS AT TOP CENTER OF ARRAY    001140
180 CALL WOAP(1,3,W,MX,NX)          001141
CALL WOAP(2,1,W,MX,NX)          001142
CALL WOAP(2,3,W,MX,NX)          001143
CALL WOAP(3,3,W,MX,NX)          001144
DO 184 K = 1,3          001145
184 W(AU(K,3,N)) = (DHD(1,1) + DUX) * W(K,M,N)      001146
A + DUX*(W(K,M+1,N) + W(K,M-1,N))                001147
B + DHD(1,2)*(W(K,M+2,N) + DHD(1,3)*W(K,M,N-2))  001148
C + DTDX*(FP(K,3,3) - FP(K,1,3))                001149
D + DTOY*(GP(K,2,3) - GP(K,2,1))                001150
GO TO 200          001151
C----TYPE 9 CELL, POINT IS AT UPPER RIGHT CORNER OF ARRAY 001152
190 CALL WOAP(1,3,W,MX,NX)          001153
CALL WOAP(3,1,W,MX,NX)          001154
CALL WOAP(3,3,W,MX,NX)          001155
DO 194 K = 1,3          001156
194 W(AU(K,3,N)) = 2.0*DHD(1,1)*W(K,M,N)          001157
A + DHD(1,2)*(W(K,M-1,N) + W(K,M,N-1))          001158
B + DHD(1,3)*(W(K,M-2,N) + W(K,M,N-2))          001159
C + DTDX*(FP(K,3,3) - FP(K,1,3))                001160
D + DTOY*(GP(K,3,3) - GP(K,3,1))                001161
GO TO 200          001162
C----TYPE 5 CELL, POINT IS AT CENTER OF ARRAY        001163
150 IF(JCT=9) 400,400,380          001164
C----TYPES 10,11,12,13          001165
380 CALL WOAP(1,2,W,MX,NX)          001166
CALL WOAP(2,1,W,MX,NX)          001167
CALL WOAP(2,3,W,MX,NX)          001168
CALL WOAP(3,2,W,MX,NX)          001169
JCTP = JCT - 9          001170
CALL WOP(JCTP,W,MX,NX)          001171
GO TO 430          001172
C----HALF STEPS          001173
400 JA = 1          001174
MM = M          001175
NN = N - 1          001176
402 DO 404 K = 1,3          001177
404 W(K) = W(K,MM,NN) + DTDX*CJX(2,3)*(F(K,4,NN) - F(K,2,NN))  001178
A + DTOY*(CJX(JA,1)*G(K,3,N-1) + CJX(JA,2)*G(K,3,N))          001179
B + CJX(JA,3)*G(K,3,N+1)          001180
R1 = 1.0/RW(3)          001181
GP(1,2,JA) = -WW(1)*WW(2)*R1          001182
UP(2,2,JA) = -WW(2)**2*R1 - UNOG*WW(3)**GAM          001183
GP(3,2,JA) = -WW(2)          001184
IF(JA = 1) 406,406,410          001185
406 JA = 3          001186
NN = N + 1          001187
GO TO 402          001188
410 NN = N          001189

```

```

MM = M - 1 001190
IA = 1 001191
412 UO 414 K = 1,3 001192
414 "W(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))
H + DTDX*CJX(2,3)*(G(K,3,N+1) - G(K,3,N-1))
RI = 1.0/W(3) 001193
FP(1,IA,2) = -W(1)*2*RI - UNOG*WW(3)*GAM 001194
FP(2,IA,2) = -W(1)*WW(2)*RI 001195
FP(3,IA,2) = -W(1) 001196
1F(IA - 1) 416,416,430 001197
430 IA = 3 001198
MM = M + 1 001199
GO TO 412 001200
C----SECOND STEP 001201
430 UO 154 K = 1,3 001202
154 "AU(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))
H + DTDX*(FP(K,3,2)-FP(K,1,2)) + DTDX*(GP(K,2,3)-GP(K,2,1)) 001203
200 CONTINUE 001204
GO TO 1640,700,KOT 001205
C----INERT CELLS 001206
680 UO 082 N = NL1,NL2 001207
UO 082 K = 1,3 001208
DHZ "AU(K,3,N) = W(K,M,N) 001209
GO TO 1640,700,KOT 001210
C----COLUMN SHIFTING 001211
700 1F(1 - MRS)702,706,706 001212
702 UO 704 N = 1,RA,NR1 001213
UO 704 K = 1,3 001214
UO 704 JL = 2,3 001215
F(K,JL,N) = F(K,JL+1,N) 001216
704 G(K,JL,N) = G(K,JL+1,N) 001217
1F(1 - MFD-1) 710,710,706 001218
706 UO 708 N = 1,RB,NRS 001219
UO 708 K = 1,3 001220
708 W(K,M-2,N) = WAD(K,1,N) 001221
710 UO 712 N = 1,RB,NRS 001222
UO 712 K = 1,3 001223
UO 712 JL = 1,2 001224
712 "AU(K,JL,N) = WAD(K,JL+1,N) 001225
1F(K-MRS)714,716,716 001226
714 M = M + 1 001227
GO TO 630 001228
716 UO 718 N = 1,RA,NRS 001229
UO 718 K = 1,3 001230
W(K,M-1,N) = WAD(K,1,N) 001231
720 "W(K,M,N) = WAD(K,2,N) 001232
RETURN 001233
END 001234

```

SUBROUTINE WOP(I,W,M,N)		000808
C 1 JCT IPX IPY		000809
C 1 10 1 1		000810
C 2 11 2 3		000811
C 3 12 3 2		000812
C 4 13 4 4		000813
COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, UNYNN, HINF, GMU, UNOGMU		000814
COMMON/CUR, JR, MR, NR, DTOX, DTDY, F(3,5,26), G(3,5,26),		000815
1 FP(3,3,3), GP(3,3,3), WADT(3,3,26)		000816
COMMON/PELUD/CJX(3,3), JCX(3,6), DCY(3,6), DHD(3,3), DMGX(5,6),		000817
A LMMY(5,6), LBIX(6), LABY(6)		000818
DIMENSION W(3), W(3,MM,NN), ZCX(4,3,3)		000819
DATA ZCX/0.0,-1.0,0.5,-0.5,-1.0,-1.0,-1.0,1.0,1.0,0.0,0.0,0.5,-0.5,-1.0		000820
1,1.0,-1.0,1.0,2.0,-2.0,2.0,-2.0,-1.0,1.0,-1.0,1.0,1.0,0.5,-0.5,0.0,		000821
2-1.0,-1.0,1.0,-1.0,1.0,0.5,-0.5,1.0,0.0/		000822
EQUIVALENCE (W1,WN(1)),(W2,WN(2)),(W3,WN(3))		000823
IPX = 1		000824
IA = 2		000825
JA = 1		000826
II = 3		000827
IF (I,GE,3) JI = 3		000828
MR = MR + IA		000829
NN = NR + JA		000830
DO 10 KA = 1,3		000831
SF = SG = 0.0		000832
DO 10 JL = 1,3		000833
JN = NR + JL		000834
SG = SG + CJX(JA,JL)*G(KA,II,JN)		000835
DO 10 JJ = 1,3		000836
JI = JR + JJ		000837
10 SF = SF + ZCX(IPX,JI,JL)*F(KA,JI,JN)		000838
10 WN(KA) = W(RA,MM,NN) + DTOX*SF + DTDY*SG		000839
IFLAG = 1		000840
PA = UNOG+W3*GAM		000841
K = 1.0 / W3		000842
U1A = W1*K		000843
U2A = W2*K		000844
FP(1,IA,JA) = - W1*U1A - PA		000845
FP(2,IA,JA) = - W1*U2A		000846
FP(3,IA,JA) = - W1		000847
GP(1,IA,JA) = FP(2,IA,JA)		000848
GP(2,IA,JA) = - W2*U2A - PA		000849
GP(3,IA,JA) = - W2		000850
IF (IFLAG,GT,1) RETURN		000851
JA = 2		000852
IA = 1		000853
IF (IPX,FO,2.0R,IPX,EG,4) IA = 3		000854
IPY = IPX		000855
IF (IPX,EG,2) IPY = 3		000856
IF (IPX,EG,3) IPY = 2		000857
MR = MR + IA		000858
NN = NR + JA		000859
DO 26 KA = 1,3		000860
SF = SG = 0.0		000861
DO 20 JL = 1,3		000862
JN = NR + JL		000863

JI = JR + JL 000864
SF = SF + CJX(IA,JL) * F(KA,JI,NN) 000865
DO 20 JJ = 1,3 000866
II = JJ + JK 000867
20 SG = SG + ZCX(IPY,JL,JJ) * G(KA,II,NN) 000868
20 W(KA) = W(KA,MM,NN) + DTDX*SF : STDY*SG 000869
IFLAG = 2 000870
GO TO 19 000871
END 000872

```

SUBROUTINE WOAP(IIA, JJA, W, M, N)          000873
COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, QDYNII, HINF, GMU, UNOGMU   000874
COMMON/CONI/JR, MR, IR, DTDX, DTDY, F(3,5,26), G(3,5,26),      000875
1 FP(3,3,3), GP(3,3,3), WAD(3,3,26)           000876
COMMON/PLRD/CJX(3,3), DCX(3,3), DCY(3,3), DHD(3,3), DMBX(5,6), 000877
A DMIY(5,6), LEIX(6), LBIY(6)                 000878
DIMENSION W(3,M,N)                           000879
EQUIVALENCE (W1,W(1)),(W2,W(2)),(W3,W(3))       000880
IA = IIA                                     000881
JA = JJA                                     000882
II = IA + JR                                000883
MM = MR + IA                                000884
NN = NR + JA                                000885
DO 16 KA=1,3                                000886
SF = SG = 0.0                                 000887
DO 10 JL=1,3                                000888
JN = NR + JL                                000889
JJ = JR + JL                                000890
SF = SF + CJX(IA,JL)+F(KA,JJ,NN)            000891
10 SG = SG + CJX(JA,JL)+G(KA,II,JN)          000892
16 ..(KA) = W(KA,..M,NN) + DTDX*SF + DTDY * SG  000893
PA = UNOG*W3*GAM                            000894
R = 1.0/W3                                    000895
U1A = W1*R                                    000896
U2A = W2*R                                    000897
FP(1,IA,JA) = - W1*U1A - PA                000898
FP(2,IA,JA) = - W1*U2A                      000899
FP(3,IA,JA) = - W1                         000900
GP(1,IA,JA) = FP(2,IA,JA)                   000901
GP(2,IA,JA) = - W2*U2A - PA                000902
GP(3,IA,JA) = - W2                         000903
RETURN
END

```

```

----- SUBROUTINE MERI(KR,LUX,W,ITSMO) ----- 001240
C---- ADVANCES SOLUTION AT AIRFOIL SURFACE POINTS AND AT ONE ROW OF OUTPUT 001241
C---- MESH POINTS USING LOCAL CARTESIAN MESH TANGENT SURFACE 001242
C---- ADVANCES SOLUTION AT 2 POINTS IN LOCAL CARTESIAN MESH TANGENT TO SU 001243
COMMON/PLIMIT/ LBFA, LBFR, LBFC, LBFD, LBFE, LBFI, LBFI, LBFK, 001244
F LBFL, LBFM, LBMAX,
H LBHA, LBHD, LBHC, LBHD, LBHE, LBHI, LBHJ, LBHK, LBHL, LBHM, 001245
H LBIMAX, 001246
M LBKA, LBKB, LBMC, LBMD, LBME, LBNI, LBNU, LBPK, LBVL, LBWM, 001247
M LBMAX, 001248
T LBTA, LBTB, LBTC, LBTD, LBTE, LBTI, LBTL, LBTK, LBTL, LBTM, 001249
T LBTU, LBTR, LBTS, LBTT, LBMAX 001250
COMMON/GAS/ GAM, HUMU, UNOU, DOGMU, GRMT, HINF, CMU, UNOGMU 001251
COMMON/CUN1/JR, MR, NR, DDX, DTY, F(3,5,26), G(3,5,26), 001253
1 FPI(3,3,3), GP(3,3,3), WAD(3,3,26) 001254
COMMON/EI.THP/ JTNO, TEND, TEHB 001255
COMMON/TUC/PRMSHY,ATSFY,HGP0,HGPV,IGPW,ATSF(6),TSN(6),PRMSH(6), 001256
1 PRMSH(6), PRMYI, CFTT 001257
COMMON/PELU/CJX(3,3), DCX(3,6), DCY(3,6), DHD(3,3), DMBX(5,6), 001258
A CMISY(5,6), LRIX(6), LBIX(6) 001259
DIMENSION W(3,9,LBX), FA(3,3,3), GA(3,3,3), WA(3), 001260
A FU(3,3,3), GO(3,3,3) 001261
KUT = 1 001262
GO TO 150,150,30,40,50,60,T,KR 001263
C---- FINE MESH REGION 3 001264
30 MLL = LPFK 001265
MUL = LPFC 001266
MDSW = LBFC + 1 001267
MDFW = LBFC + 1 001268
GO TO 100 001269
32 MLL = LPFK 001270
MUL = LPFL 001271
MDSW = LBFL + 1 001272
MDFW = LBFL + 1 001273
GO TO 100 001274
C---- OUTER NOSE REGION 4 001275
40 MLL = LPHB 001276
MUL = LBHC 001277
MDSW = LBHC + 1 001278
MDFW = LBHU + 1 001279
GO TO 92 001280
42 MLL = LPHU 001281
MUL = LBHL 001282
MDSW = LBHJ 001283
MDFW = LBHK 001284
GO TO 96 001285
C---- INTERMEDIATE NOSE REGION 5 001286
50 MLL = LPMB 001287
MUL = LBMD 001288
MDSW = LBMC + 1 001289
MDFW = LBMD + 1 001290
GO TO 92 001291
52 MLL = LPMJ 001292
MUL = LBML 001293
MDSW = LBMJ 001294
MDFW = LBMK 001295
-----
```

```

GO TO 96
C---- INNER NOSE REGION 6
60 MLL = LBTB
    MUL = LBTC
    MDSW = LBTC + 1
    MDFW = LBTD + 1
    GO TO 92
62 MLL = LATJ
    MUL = LPIL
    MDSW = LBTP
    MDFW = LBTK
    GO TO 96
64 MLL = LPTK
    MUL = LRTS
    MDSW = LRTS + 1
    MDFW = LRTS + 1
    GO TO 100
68 MODOC = 1
    JDMX = 1
    GO TO 100
90 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 = I + (K-1)/3
        P = U(10G+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(1,I,J) = -w(2,K,M)*2*R - P
        GO(2,I,J) = FO(2,I,J)
102 GO(3,I,J) = -w(2,K,M)
    GO TO (104,105), ITSMU
C---- ALLOWABLE STEP. USE WALL POINT AS CRITICAL
104 J = 2
    RMU = W(3,J,M) - 1.0
    WA = (1.0+RMU+IGPU*(1.0+RMU*IGPW)*(1.0+RMU*IGPW)) * ATSF(KR)
    UTUY = TSF(KR)*w(3,J,M)/(ABS(w(1,J,M))*PRNSH(KR) + ABS(w(2,J,M)))
    A + RA)
    UTUX = PRNSH(KR)*DTDY
C---- HALF STEPS, F,G CONVERSIONS
105 DO 120 K = 1,9
    GO TO (104,106,106,106,106,106,120,106,120), K
106 IR = K - 3*((K-1)/3)
    JR = I + (K-1)/3
    WO 108 KA = 1,3
    108 WA(KA) = w(KA,K,M) + DTDX*(CJX(IR,1)*FO(KA,1,IR))
    A + CJX(IR,2)*FO(KA,2,IR) + CJX(IR,3)*FO(KA,3,IR))
    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 = U(10G+WA(3)**GAM
    GO TO (110,114,110,110,114,110,120,114,120), K
110 FA(3,IR,IR) = -WA(1)
    FA(2,IR,IR) = -WA(1)*WA(2)/WA(3)

```

```

-----  

FA(1,IR,IR) = -WA(1)*Z, WA(3) = .  

GO TO 120  

-----  

114 GA(3,IR,IR) = -WA(1)  

GA(2,IR,IR) = -WAT2)*Z/WA(3) = F  

GA(1,IR,IR) = -WA(1)*WA(1)/WA(3)  

-----  

120 CONTINUE  

IF(M = MUSW) 129,121,121  

121 IF(M = MUFW) 122,129,129  

122 GO TO (123,124), MODUC  

123 JDMX = JUMX + 1  

GO TO 125  

124 JDMX = JUMX - 1  

125 DMULT = UFBX(JUMX,KR)  

DUCX = 0.5 - DMULT*(0.5 - DCX(1,KR))  

1252 DUCY = 0.5 - DMULT*(0.5 - DCY(1,KR))  

126 LUKE = DMULT*DCX(2,KR)  

DUYE = DMULT*DCY(2,KR)  

DO 127 KA = 1,3  

WA(KA) = (DUCX+DUCY+UYE)*W(KA,2,M) + DUXE*(W(KA,1,M)  

A + W(KA,3,M)) + 2.0*UYE*W(KA,5,M) - DUYE*W(KA,8,M)  

B + DTDX*(FA(KA,3,1) - FA(KA,1,1)) + 2.0*UTDY*(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)) + UTDY*(GA(KA,2,3) - GA(KA,2,1))  

GO TO 132  

C----SECOND STEP  

128 DO 130 KA = 1,3  

WA(KA) = (DCX(1,KR) + DCY(1,KR) + DCY(2,KR))*W(KA,2,M)  

A + DUCX(2,KR)*(W(KA,1,M)+W(KA,3,M)) + 2.0*DCY(2,KR)*W(KA,5,M)  

B - DUCY(2,KR)*W(KA,8,M) + DTDX*(FA(KA,3,1)-FA(KA,1,1))  

C + 2.0*DTDX*(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,M)*(W(KA,2,M) + W(KA,8,M)) + DTDX*(FA(KA,3,2)  

C - FA(KA,1,2)) + UTDY*(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)/HGMU  

W(1,2,M) = UTB*W(3,2,M)  

W(2,2,M) = 0.0  

C----ENTHALPY COERCION  

IF(JTID.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 ----- 001405
C---- DETERMINES TIME STEP ALLOWABLE IN REGION 6 001406
----- COMMON/LIMITS/ MYA, MYB, MYS, MYI, NYA, NYB, NYS, NYI, 001407
C YCA, MCB, NCE, MCF, MCG, MCII, MCS, MCT, NCA, NCB, NCE, NCF, 001408
C NCG, NCH, NCS, NCT, 001409
F MFAD, MFBD, MFED, MFE, MFF, MFG, MFH, MFS, MFT, MFAN, 001410
F MFATD, MFATB, MFATP, MFA, MFB, MFD, MFE, MFF, MFG, MFH, 001411
F NFU, NFS, NFT, NFCL, 001412
H NHAD, NHBD, NHCD, NHED, NHFD, NHGD, NHHD, NHID, NHSD, NHTD, NHCL, 001413
H NHAD, NHBD, NHCD, NHED, NHFD, NHGD, NHHD, NHID, NHSD, NHCL, 001414
M NMAD, NMBD, NMCD, NMED, NMFD, NMGD, NMHD, NMID, NMSD, NMCL, 001415
M NMAD, NMBD, NMCD, NMED, NMFD, NMGD, NMHD, NMID, NMSD, NMCL, 001416
T NTAD, NTBD, NTCD, NTED, NTFD, NTGD, NTHD, NTID, NTSD, NTCL, 001417
T NTG, NTS, NTT, NTCL, 001418
S NSL, MST, NSL, NST 001419
COMMON/LIMIT/ LBFA, LBFB, LBFC, LBFD, LBFE, LBFI, LBFI, LBFK, 001420
F LBFL, LBFM, LBMAX, 001421
H LBIA, LBIB, LBIC, LBID, LBIE, LBIF, LBIF, LBHK, LBHL, LBHM, 001422
H LBIMAX, 001423
M LBIA, LBIB, LBIC, LBID, LBIE, LBIF, LBIM, LBIM, LBML, LBMM, 001424
N LBIMAX, 001425
T LBIA, LBIB, LBIC, LBID, LBIE, LBIF, LBIF, LBTL, LBTM, 001426
T LBIA, LBIB, LBIC, LBID, LBIE, LBIF, LBIF, LBTL, LBTM, 001427
COMMON/GMS/ GAM, GMU, UNOG, DOGMU, QMVI, RINF, GMU, UNOGMU 001428
COMMON/TUC/PRASHY, ATSFY, HGPU, HGPV, HGPW, ATSF(6), TSN(6), PRMSH(6), 001429
1 PRMSH(6), PRMYI, CFTT 001430
COMMON/COM1/JR, MR, NR, DTDX, DTDY, F(3,5,26), G(3,5,26), 001431
1 FPC(3,3), GP(3,3,3), WAIT(3,3,26) 001432
COMMON WY(3,42,9), WC(3,17,17), WF(3,37,13), WH(3,25,17), 001433
A WM(3,26,26), WT(3,10,21), WSF(3,9,42), WSH(3,9,22), WSM(3,9,14), 001434
H WST(3,9,43), WSU(3) 001435
CT = 10.0 001436
KR = 6 001437
KUT = 1 001438
MLL = LBTR 001439
MUL = LPTC 001440
100 DO 110 M = MLL, MUL 001441
DO 110 JC = 2,5,3 001442
RMU = WST(3,JC,M) - 1.0 001443
RA = (1.0 + RMU*HGPU*(1.0 + RMU*HGPV*(1.0 + RMU*HGPW))) * ATSF(KR) 001444
CTE = WST(3,JC,M) / (ABS(WST(1,JC,M)) + PRNSH(KR) + ABS(WST(2,JC,M)) + RA) 001445
1F(CTE - CT) 102,110,110 001446
102 CT = CTE 001447
110 CONTINUE 001448
GO TO (112,114,120), KUT 001449
112 KUT = 2 001450
MLL = LPTJ 001451
MUL = LPTL 001452
GO TO 100 001453
114 KUT = 3 001454
MLL = LPTK 001455
MUL = LBTS 001456
GO TO 100 001457
120 DO 130 M = MTH, MTS 001458
DO 130 N = NTB, NTS 001459
RMU = WT(3,M,N) - 1.0 001460

```

RA = (I.U + RMU+HGPU*(I.U + RMU+HGPW)) * ATSF(KR) 001461
CTE = WT(3,M,N) / (ADS(WT(1,M,N)) * PRMSH(KR) * ABS(WT(2,M,N)) * RA) 001462
IF(CTE - CT) 122,130,130 001463
122 CT = CTE 001464
130 CONTINUE 001465
DTDX = TSN(KR) * CT 001466
DTUX = PRMSH(KR) * DTDY 001467
RETURN 001468
END 001469

```

SUBROUTINE ENGS(W) 001470
C THIS VERSION HOLDS THE PRESSURE AND CHANGES THE VELOCITY MAGNITUDE 001471
C IN ORDER TO TEND TOWARD STEADY FREE-STREAM TOTAL ENTHALPY 001472
C PRESERVES MACH NUMBER IF STATIC ENTHALPY EXCEEDS DESIRED TOTAL 001473
COMMON/GAS/ GAM, HGMU, UNOG, DOGMU, GDYN, HINF, GMU, UNOGMU 001474
COMMON/ENTH/ P, JTND, TEIN, TENS
DIMENSION W(3) 001475
K = 1.0/W(3) 001476
UX = W(1)*R 001477
UY = W(2)*R 001478
L = W(3)**GMU 001479
LSKI = 0.5*(UX**2 + UY**2) 001480
HSTAT = 2.5*E 001481
HTOT = HSTAT + ESKI 001482
001483
IF (HSTAT = HINF) 5,10,10 001484
5 UFAC = SQRT(1.0 + TEIN*(HINF-HTOT)/ESKI) 001485
W(1) = UFAC*W(1) 001486
W(2) = UFAC*W(2) 001487
RETURN 001488
10 EFAC = 1.0 - TEIN*HINF/HTOT 001489
W(3) = (E+EFAC)**UNOGMU 001490
UFAC = W(3)*SQRT(EFAC) 001491
W(1) = UFAC*UX 001492
W(2) = UFAC*UY 001493
RETURN 001494
END 001495

```

SUBROUTINE SAKAI
 C---SUBROUTINE INTERCHANGES INFORMATION BETWEEN VARIOUS REGIONS, 001672
 C APPLIES KUTTA CONDITION 001673
 C JTND = 2, AIRFOIL POINTS FORCED TO AND STEADY FREE-STREAM TOTAL ENT 001674
 C---SWITCHING CODE FOR IUMT, ETC. 001675
 C 1--NO INTERPOLATION, 2--USE 4 POINT SPREAD, 001676
 C 3--USE 0 POINT SPREAD WITH ONLY ONE ABOVE, 4--USE 3 POINT SPREAD 001677
 C WITH ONLY ONE BELOW 001678
 C 001679
 C JOFB = 1 FREESTREAM CONDITION MAINTAINED AT INFINITY 001680
 C JOFB = 2 FREESTREAM CONDITION MAINTAINED OUTSIDE REGION 2 001681
 C JOFB = 3 FREESTREAM CONDITION MAINTAINED OUTSIDE REGION 3 001682
 C 001683
 COMMON/GAS/ GAM, HGMU, UNOG, DOGMR, OODYN, HINF, GMU, UNOGMU 001684
 COMMON/FLAGS/ IBUG, JBCT, JOFB 001685
 COMMON/ENTHP/ JTND, TEND, TENB 001686
 COMMON/LIMITS/ MYA, MYB, MYS, MYT, NYA, NYB, NYS, NYT, 001687
 C NCA, MCN, NCE, MCF, MCG, MCM, MCS, MCT, NCA, NCB, NCE, NCF, 001688
 C NCG, NCH, NCS, NCT, 001689
 F MFA, MFB, MFD, MFE, MFF, MFG, MFO, MFS, MFT, MFAN, 001690
 F MFANP, MFAT, MFATP, MPA, NFB, NFD, NFE, NFF, NFG, NFH, 001691
 F NFO, NFS, NFT, NFCL, 001692
 H NHAA, NHBA, NHBD, NHDE, NHF, NHG, NHJ, NHU, NHS, NHV, NHCL, 001693
 H NHAA, NHBD, NHDE, NHF, NHG, NHJ, NHU, NHS, NHV, NHCL, 001694
 M MMA, MMBA, MMBD, MMDE, MMF, MMG, MMH, MMJ, MMU, MMV, MMCL, 001695
 M NMHA, NMHD, NMID, NMDE, NMF, NMG, NMH, NMJ, NMV, NMCL, 001696
 T MTAA, MTBA, MTD, MTO, MTS, MTI, MTAN, MTANP, NTA, NTB, NTD, 001697
 T NTU, NTS, NTT, NTCL, 001698
 S NSL, NST, NSL, NST 001699
 COMMON/RLIMIT/ LHFA, LBFB, LBFC, LHF, LBFE, LRFI, LBFI, LBFK, 001700
 F LBFL, LBFM, LBMAX, 001701
 H LBHA, LBHB, LBHC, LBHD, LBHE, LBHI, LBHJ, LBHK, LBHL, LBHM, 001702
 H LBHMAX, 001703
 M LBMA, LBMB, LBMC, LBMD, LBME, LBM1, LBQJ, LBK, LBML, LHM, 001704
 M LBMMAX, 001705
 T LUTA, LBTA, LBTC, LBTD, LHTE, LBTI, LHTJ, LBTK, LHTL, LBTM, 001706
 T LBTU, LUTR, LBTS, LBTT, LBTHMAX, 001707
 COMMON/PEND/CJX(3,3), DCX(3,6), DCY(3,-1), DHD(3,3), DMGX(5,6), 001708
 A UMBY(5,6), LBIX(6), LBYY(6) 001709
 COMMON/ZMESH/ DX(6), DY(6), RH(9) 001710
 T JMN0N/ GEOMF/ SY(42), CY(42), DYR(2,42), DCBTZ(4,17), 001711
 A XYF(2,42), SCF(2,42), XYH(2,22), SCH(2,22), 001712
 B XYM(2,14), SCM(2,14), XYT(2,43), SCT(2,43) 001713
 COMMON/TERSE/ ITH(2,42), TCB(2,4,17), IDHM(10), 001714
 A IUMT(10), IWFI(3,42), IWH(3,22), IWM(3,14), IWT(3,43), 001715
 B LINTF(2), LINTH(2), LINTM(2), IWFX(2,2), IWHX(2,2), IWMX(2,2), 001716
 C IAF(2,37), IAH(2,25), IAM(2,26), IAT(2,16), IAT(2,17), 001717
 D INFIT(2,4), IWH(2,4), IWM(2,4) 001718
 COMMON/OMOSA/ DWF(9,4,42), DWH(9,4,22), DWA(9,4,14), 001719
 A DWF(9,4,43), AWF(3,42), AWI(3,22), AWI(3,14), AWI(3,43), 001720
 B CWF(9,3,42), CWF(9,3,22), CWM(9,3,14), CWT(9,3,43), 001721
 C DWF(9,2,6), DWF(9,2,6), DWM(9,2,6), DWXT(9,2,6), 001722
 D AWFH(3,3), AWHM(3,3), AWMT(3,3), 001723
 E DWFH(9,2,4), UWHM(9,2,4), UWMT(9,2,4), 001724
 F CWFX(9,2,2), CWHX(9,2,2), CWM(9,2,2) 001725
 COMMON/TOPL/MUL,MUL,KR,LIP,KUT,LC,JC,MLBR,NLBR,XSPL 001726
 001727

```

COMMON      WY(3,42,9), WC(3,17,17), WT(3,37,13), WH(3,25,17),     001728
A WM(3,26,26), WT(3,16,21), WSF(3,9,42), WSH(3,9,22), WSM(3,9,14),  001729
B WST(3,9,43), WSU(3)                                001730
DIMENSION GJX(4,4), HJX(4,3), PJX(3,3), UJX(3,6), RJX(4,9),     001731
A DELW(3), SJX(4,7), TJX(3,7), RHOB(2), UTHB(2)      001732
C                                     001733
C-----EIGHTING FACTORS FOR INTERPOLATIONS        001734
C
DATA GJX/-0.75294117, .945882350, .134117650, -.004705882,      001735
1-.083076923, .729230770, .390769230, -.036923077, -.036923077,   001736
2.390769230, .729230770, -.083076923, -.004705882, .134117650,   001737
3.945882350, -.075294117                                001738
C                                     001739
C-----DATA HJX/-0.084375, .909375, .184375, -.009375, -.0625, .5625, .5625, -.06    001740
1<5, -.009375, .184375, .909375, -.084375/          001741
C                                     001742
C-----DATA PJX/-0.09375, .9375, .15625, -.125, .75, .375, -.09375, .4375, .65625/  001743
C                                     001744
C-----DATA UJX/-0.08, .90, .12, -.12, .84, .28, -.12, .64, .48, -.08, .36, .72,      B 001746
1 0.0, 0.0, 1.0, .12, -.44, 1.32/                     H 001747
DATA RJX/-0.044451219, .98353658, .056646341, -.000548780, -.075294117  001748
1, .945882350, .134117650, -.004705882, -.088706497, .861120690, .2438793  001749
210, -.016203103, -.083076923, .729230770, .390769230, -.036923077, -.062  001750
35, .5625, .5625, -.0625, -.036923077, .390769230, .729230770, -.083076923 001751
4, -.016293103, .243879310, .861120690, -.088706497, -.004705882, .134117  001752
5050, .945882350, -.075294117, -.000548780, .056646341, .068353658, -.044  001753
6451<19/                                001754
DATA SJX/-0.05359375, .98109375, .07359375, -.00109375,      001755
A -.004375, .909375, .184375, -.009375, -.08616728, .76631434,      001756
B .350873107, -.03102022, -.0625, .5625, .5625, -.0625,      001757
C -.03102022, .35087316, .76631434, -.08616728, -.009375,      001758
D .184375, .909375, -.084375, -.00109375, .07359375, .98109375,      001759
E -.05359375/                                001760
DATA TJX/-0.0546875, .984375, .0703125, -.09375, .9375, .15625,      001761
A -.1171875, .659375, .2578125, -.125, .75, .375, -.1171875,      001762
B .609375, .5078125, -.09375, .4375, .65625, -.0546875,      001763
C .234375, .8203125/                          001764
C-----THE LOADING POINTS FROM COARSE MESH          001765
560 DO 562 N = MYB,MYS                                001766
MA = IYB(1,M)                                001767
NA = IYB(2,M)                                001768
AX = DYB(1,M)                                001769
AY = DYB(2,M)                                001770
CCP = 0.125*AX+AY     . 001771
CZP = 1.0 - AX**2 - AY**2     001772
CXP = 0.5*AX*(AX + 1.0)     001773
CXM = CXP - AX     001774
CYP = 0.5*AY*(AY + 1.0)     001775
CYM = CYP - AY     001776
DC 562 K = 1,3                                001777
562 wY(K,MA,1) = WC(K,MA,NA)*CZP + WC(K,MA+1,NA)*CXP + WC(K,MA-1,NA)  001778
1 *CXM + WC(K,MA,NA+1)*CYP + WC(K,MA,NA-1)*CYM + CCP + (WC(K,MA+1, 001779
2 NA+1) + WC(K,MA-1,NA-1) - WC(K,MA+1,NA-1) - WC(K,MA-1,NA+1))      001780
DO 564 N = 1,NYS                                001781
DO 564 K = 1,3                                001782
wY(K,MYA,N) = wY(K,MYS,N)                    001783
564 wY(K,MYT,N) = wY(K,MYB,N)                  001784

```

```

C----COARSE MESH BOUNDARY POINTS FROM TIME          001785
C----LEFT COLUMN                                     001786
J = 1                                                 001787
M = MCA                                              001788
N = 0                                                 001789
MD = 0                                               001790
ND = 1                                               001791
LTP = NCT                                            001792
LBT = I                                              001793
001794
001795
001796
001797
001798
001799
001800
001801
001802
001803
001804
001805
001806
001807
001808
001809
001810
001811
001812
001813
001814
001815
001816
001817
001818
001819
001820
001821
001822
001823
001824
001825
001826
001827
001828
001829
001830
001831
001832
001833
001834
001835
001836
001837
001838
001839
001840
001841

C----COLUMN AND ROW ROUTINE
570 DO 572 LA = LBT,LTP
      M = M + MD
      N = N + NE
      MA = ICBT1,J,LA)
      NA = ICBT2,J,LA)
      AX = DCR(1,J,LA)
      AY = DCR(2,J,LA)
      CCP = U,1?5*AX*AY
      LXP = 1.U - AX**2 - AY**2
      LXM = 0.5*AX*(AX + 1.0)
      CXM = CXP - AX
      LYP = 0.5*AY*(AY + 1.0)
      CYM = CYP - AY
      DO 572 K = 1,3
      572 WC(K,M,N) = WY(K,MA,NA)*CZP + 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 = U
      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 = NTA
      DO 324 JN = NMF,NMG

```

$N = N + 1$	001842
$NN = NN + 2$	001843
$DO 324 K = 1,3$	001844
$324 _W(K,M,N) = WT(K,MM,NN)$	001845
C----OUTER NOSE REGION -- INTERMEDIATE NOSE REGION INTERCHANGE	001846
C----ASSIGNMENT TO OUTER NOSE REGION	001847
$580 M = MHE$	001848
$NN = NMA$	001849
$DO 582 JM = MHF,MHG$	001850
$M = M + 1$	001851
$MM = MM + 5$	001852
$N = NHE$	001853
$NN = NMA$	001854
$DO 582 JN = NHF,NHG$	001855
$N = N + 1$	001856
$NN = NN + 5$	001857
$DO 582 K = 1,3$	001858
$582 _W(K,M,N) = WN(K,MM,NN)$	001859
C----FINE MESH -- OUTER NOSE REGION INTERCHANGE	001860
C----ASSIGNMENT TO FINE MESH REGION	001861
$612 M = MFE$	001862
$NN = NHA$	001863
$DO 612 JM = MFF,MFG$	001864
$M = M + 1$	001865
$MM = MM + 4$	001866
$N = NFE$	001867
$NN = NHA$	001868
$NLL = NFF$	001869
$NUL = NFG$	001870
$612 DO 612 JN = NLL,NUL$	001871
$N = N + 1$	001872
$NN = NN + 4$	001873
$DO 612 K = 1,3$	001874
$612 _W(K,M,N) = WH(K,MM,NN)$	001875
C----COARSE REGION --- FINE REGION INTERCHANGE	001876
C----ASSIGNMENT TO COARSE MESH	001877
$690 M = MCE$	001878
$MM = MFA$	001879
$DO 700 JM = MCF,MCG$	001880
$M = M + 1$	001881
$MM = MM + 4$	001882
$N = NCE$	001883
$NN = NFA$	001884
$JNL = 1$	001885
$JNU = NCG - NCE$	001886
$692 DO 700 JN = JNL,JNU$	001887
$N = N + 1$	001888
$NN = NN + 2$	001889
$DO 700 K = 1,3$	001890
$700 _W(K,M,N) = WF(K,MM,NN)$	001891
C----EXCHANGES OF INFORMATION BETWEEN MAIN CARTESIAN MESHES AND BOUNDARY	001892
C----LOCAL SURFACE NORMALS	001893
C----INNER NOSE REGION 6	001894
$NR = 6$	001895
$LIP = 2$	001896
$JC = 8$	001897
$KUT = 1$	001898

LC = 1 - TAN	001899
MLL = LB10	001900
MUL = LBTC	001901
MLBR = MME	001902
NLBR = NME	001903
1U10 CALL KAKO1(IWT,MTT,INT,NM,MMT,NHT,WST,IWT,CWT,AWT,SCT,LBTMAX, A IAI,MTT)	001904
GO TO (1U12,1014),KUT	001905
1U12 KUT = 2	001906
LC = 2*(MTT+1) - MTAN	001907
MLL = LB10	001908
MUL = LBTL	001909
GO TU 1010	001910
1U14 KUT = 3	001911
LC = 2*(MTT - MTAN + 1)	001912
MLL = LPTK	001913
MUL = LPTS	001914
CALL KAKO1(IWT,MTT,INT,NM,MMT,NMT,WST,IWT,CWT,AWT,SCT,LBTMAX, A IAN,NTT)	001915
RR = 5	001916
KUT = 1	001917
LC = 1 - MMG	001918
MLL = LEMB	001919
MUL = LEFD	001920
MLBK = MHE	001921
NLBK = NHE	001922
1U20 CALL KAKO1(IWM,INT,NM1,WH,MHT,NHT,WSM,IWM,CWM,AWM,SCW,LBMMAX, A IAM,MHM)	001923
GO TO (1U22,1026),KUT	001924
1U22 KUT = 2	001925
LC = 2*(MMT+1) - MMG	001926
MLL = LEMJ	001927
MUL = LPML	001928
GO TU 1020	001929
1U26 RR = 4	001930
KUT = 1	001931
LC = 1 - MHG	001932
MLL = LBHG	001933
MUL = LBHD	001934
MLBK = MFE	001935
NLBK = NFE	001936
1U30 CALL KAKO1(IWH,MHT,NHT,WF,MFT,NFT,WCF,IWH,CWF,AWF,SCF,LBHMAX, A IAM,MHT)	001937
GO TO (1U32,1036),KUT	001938
1U32 KUT = 2	001939
LC = 2*(MHT+1) - MHG	001940
MLL = LPHJ	001941
MUL = LBHL	001942
GO TO 1030	001943
1U36 RR = 3	001944
KUT = 1	001945
LC = 1 - MFG	001946
MLL = LBFB	001947
MUL = LBFC	001948
MLBK = MCE	001949
NLBK = NCE	001950
1U40 CALL KAKO1(WF,MFT,NFT,WCF,NCT,NCT,WSF,IWF,CWF,AWF,SCF,LBFMAX, A IAM,MFT)	001951
	001952
	001953
	001954
	001955

A (IAF,IFT)
 DO TO (1042,1046),KUT 001956
 1042 KUT = 2 001957
 LC = 2*(MFAT+3) - MFG 001958
 MLL = LBFD 001959
 MUL = LPFL 001960
 DO TO 1040 001961
 1040 CONTINUE 001962
 C----SPECIAL ASSIGNMENTS NEAR TRAILING EDGE 001963
 DO 1160 KA = 1,3 001964
 .SF(KA,7,LBFD) = WF(KA,MFAT,NFCL+2) 001965
 DO 1160 JA = 1,3 001966
 ..SF(KA,-1+3+JA,LBFD) = WF(KA,MFATP,NFCL-1+JA) 001967
 ..SF(KA,3+JA,LBFD) = WF(KA,MFATP+1,NFCL-1+JA) 001968
 DO 1160 LA = 1,3 001969
 1160 ..SF(KA,-3+LA+3+JA,LBFE) = WF(KA,MFAT+LA,NFCL-1+JA) 001970
 RFC = -1.0 001971
 DO 1162 KA = 1,3 001972
 IF(KA,EO,3) RFC = 1.0 001973
 ..SF(KA,9,LBFJ) = RFC*WF(KA,MFAT,NFCL-2) 001974
 DO 1162 JA = 1,3 001975
 ..SF(KA,-1+3+JA,LBFJ) = RFC*WF(KA,MFATP,NFCL+1-JA) 001976
 ..SF(KA,-2+3+JA,LBFJ) = RFC*WF(KA,MFATP+1,NFCL+1-JA) 001977
 DO 1162 LA = 1,3 001978
 1162 ..SF(KA,-3+LA+3+JA,LBFJ) = RFC*WF(KA,MFAT+4-LA,NFCL+I-JA) 001979
 C----SPECIAL ASSIGNMENTS AT OVERLAPS 001980
 C----INNER NOSE REGION 6 001981
 KUT = 1 001982
 LU = LBTT 001983
 LK = LBTD 001984
 1160 DO 1168 KA = 1,3 001985
 DO 1168 JC = 2,8,3 001986
 1160 ..ST(KA,JC,LUF) = WST(KA,JC,LK) 001987
 DO TO (1170,1172,1174,1178),KUT 001988
 1170 KUT = 2 001989
 LU = LBTA 001990
 LK = LBTS 001991
 DO TO 1166 001992
 1172 KUT = 3 001993
 LU = LBTL 001994
 LK = LBTR 001995
 DO TO 1166 001996
 1174 KUT = 4 001997
 LU = LBTM 001998
 LK = LBTR 001999
 DO TO 1166 002000
 C----REGION ABUTMENTS (CODE D) POINTS 002001
 1178 DO 1180 KA = 1,3 002002
 ..SM(KA,2,LBMA) = WST(KA,2,LBTM) 002003
 ..SM(KA,2,LBMM) = WST(KA,2,LBTR) 002004
 ..SM(KA,2,LBHA) = ..SM(KA,2,LBMC) 002005
 ..SM(KA,2,LBHM) = ..SM(KA,2,LBMK) 002006
 ..SF(KA,2,LBFM) = WSH(KA,2,LBIC) 002007
 1180 ..SF(KA,2,LBFM) = WSH(KA,2,LBIR) 002008
 C----REGION ABUTMENTS, (CODE E) POINTS 002009
 KUT = 1 002010
 LU = LBTE 002011
 DO TO 1180 002012

LK = LBMB 002013
 1182 DO 1184 KA = 1,3 002014
 DO 1184 LIP = 1,3 002015
 JC = -1 + 3*LIP 002016
 "ST(KA,JC,LU) = 0.0 002017
 DO 1184 JL = 1,3 002018
 LUP = -1 + 3*JL 002019
 1184 "ST(KA,JC,LU) = WST(KA,JC,LU) + AWMT(JL,LIP)*WSM(KA,LUP,LK) 002020
 DO TU (1186,1188), KUT 002021
 1186 KUT = 2 002022
 LU = LBTL 002023
 LK = LBML 002024
 DO TU 1182 002025
 1188 KUT = 1 002026
 LU = LBME 002027
 LK = LUHB 002028
 1190 DO 1192 KA = 1,3 002029
 DO 1192 LIP = 1,3 002030
 JC = -1 + 3*LIP 002031
 "SM(KA,JC,LU) = 0.0 002032
 DO 1192 JL = 1,3 002033
 LUP = -1 + 3*JL 002034
 1192 "SM(KA,JC,LU) = WSM(KA,JC,LU) + AWHM(JL,LIP)*WSH(KA,LUP,LK) 002035
 DO TU (1194,1198), KUT 002036
 1194 KUT = 2 002037
 LU = LBMI 002038
 LK = LBHL 002039
 DO TU 1190 002040
 1198 KUT = 1 002041
 LU = LBRE 002042
 LK = LBFB 002043
 1200 DO 1202 KA = 1,3 002044
 DO 1202 LIP = 1,3 002045
 JC = -1 + 3*LIP 002046
 "SH(KA,JC,LU) = 0.0 002047
 DO 1202 JL = 1,3 002048
 LUP = -1 + 3*JL 002049
 1202 "SH(KA,JC,LU) = WSH(KA,JC,LU) + WFF(KA,LIP)*WSF(KA,LUP,LK) 002050
 DO TU (1204, 329), KUT 002051
 1204 KUT = 2 002052
 LU = LBHI 002053
 LK = LBFL 002054
 DO TU 1200 002055
 C-----INTERPOLATIONS AT CARTESIAN BOUNDARY FROM COARSER TO FINER MESH 002056
 C-----INTERPOLATION INTO INNER NOSE REGION 002057
 C-----LEFT COLUMN 002058
 329 KTS = 1 002059
 M = PME 002060
 MN = NTA 002061
 N = NNE - 1 002062
 LN = NTA - 2 002063
 DO TU 340 002064
 330 "T(K,MM,NN+1) = HJX(1,2)*WM(K,M,N-1) + HJX(2,2)*WM(K,M,N) 002065
 A + HJX(3,2)*WM(K,M,N+1) + HJX(4,2)*WM(K,M,N+2) 002066
 DO TU (342,348), KTS 002067
 334 "T(K,MM,NN+1) = PUX(1,2)*WM(K,M,N-1) + PUX(2,2)*WM(K,M,N) 002068
 A + PUX(3,2)*WM(K,M,N+1) 002069

GO TO 348
 330 $wT(K, MM, NN+1) = PJX(1,2) * WM(K, M, N+2) + PJX(2,2) * wN(K, M, N+1)$
 $A + PJX(3,2) * WM(K, M, N)$
 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
 $KTS = 2$
 $M = MMH$
 $MM = MFT$
 $N = NME - 1$
 $NN = NTA - 2$
 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 (346, 350), 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 $wT(K, MM, NK) = WM(K, M, N)$
 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

002070
002071
002072
002073
002074
002075
002076
002077
002078
002079
002080
002081
002082
002083
002084
002085
002086
002087
002088
002089
002090
002091
002092
002093
002094
002095
002096
002097
002098
002099
002100
002101
002102
002103
002104
002105
002106
002107
002108
002109
002110
002111
002112
002113
002114
002115
002116
002117
002118
002119
002120
002121
002122
002123
002124
002125
002126

592 $w(K, M, N+JL) = GJX(1, JL) * WH(K, M, N-1) + GJX(2, JL) * WH(K, M, N)$ 002127
 $A + GJX(3, JL) * WH(K, M, N+1) + GJX(4, JL) * WH(K, M, N+2)$ 002128
 GO TO (602, 604), KTS 002129

593 DO 594 JL = 1,4 002130
 $w(K, M, N+L-JL) = GJX(1, JL) * WH(K, M, N+1) + GJX(2, JL) * WH(K, M, N+1)$ 002131
 $A + GJX(3, JL) * WH(K, M, N)$ 002132
 GO TO 604 002133

595 DO 596 JL = 1,4 002134
 $w(K, M, N+JL) = GJX(1, JL) * WH(K, M, N-1) + GJX(2, JL) * WH(K, M, N)$ 002135
 $A + GJX(3, JL) * WH(K, M, N+1)$ 002136
 GO TO 604 002137

600 JNL = 1 002138
 $JNU = NH - NHE$ 002139
 DO 602 JN = JNL, JNU 002140
 $N = N + 1$ 002141
 $NN = NN + 5$ 002142
 DO 602 K = 1,3 002143
 $w(K, MM, NN) = wH(K, M, N)$ 002144
 GO TU 590 002145

602 CONTINUE 002146
 DO 603 K = 1,3 002147
 $w(K, MM, NN) = wH(K, M, N)$ 002148
 C---RIGHT COLUMN 002149
 KTS = 2 002150
 $M = NH$ 002151
 $MM = MM$ 002152
 $N = NHE - 1$ 002153
 $NN = NMA - 5$ 002154
 DO 604 JN = JNL, JNU 002155
 $N = N + 1$ 002156
 $NN = NN + 5$ 002157
 $ISWCH = ICNM(JN)$ 002158
 DO 604 K = 1,3 002159
 $w(K, MM, NN) = wH(K, M, N)$ 002160
 GO TU (604, 590, 595, 593), ISWCH 002161

604 CONTINUE 002162
 DO 6040 K = 1,3 002163
 $w(K, MM, NN) = wH(K, M, N)$ 002164
 C----LOWEST ROW 002165
 $N = NHE$ 002166
 $NN = NMA$ 002167
 $KTS = 1$ 002168
 $JML = 1$ 002169
 $JMU = MH - MHE$ 002170
 DO 605 M = MHE - 1 002171
 $MM = MMA - 5$ 002172
 DO 608 JM = JML, JMU 002173
 $M = M + 1$ 002174
 $MM = MM + 5$ 002175
 DO 606 K = 1,3 002176
 $w(K, MM, NN) = wH(K, M, N)$ 002177
 DO 606 JL = 1,4 002178
 $w(K, MM+JL, NN) = GJX(1, JL) * WH(K, M-1, N) + GJX(2, JL) * WH(K, M, N)$ 002179
 $A + GJX(3, JL) * WH(K, M+1, N) + GJX(4, JL) * WH(K, M+2, N)$ 002180

606 CONTINUE 002181
 GO TO (6080, 629), KTS 002182
 C----TOP ROW 002183

6080 KTS = 2	002184
N = NHH	002185
NN = NMJ	002186
60 TU 605	002187
C----INTLKPOLATION INTO OUTER NOSE REGION	002188
C----LEFT COLUMN	002189
623 KTS = 1	002190
M = MFE	002191
MM = MMA	002192
N = NFE - 1	002193
NN = NHA - 4	002194
JNL = NFE	002195
JNU = NFG	002196
60 10 640	002197
630 60 632 JL = 1,3	002198
632 ..H(K,M,N,N+JL) = HJX(1,JL)*WF(K,M,N-1) + HJX(2,JL)*WF(K,M,N)	002199
A + HJX(3,JL)*WF(K,M,N+1) + HJX(4,JL)*WF(K,M,N+2)	002200
60 TU (642+644),KTS	002201
633 60 634 JL = 1,3	002202
634 ..H(K,M,N,N+JL) = PJX(1,JL)*WF(K,M,N+2) + PJX(2,JL)*WF(K,M,N+1)	002203
A + PJX(3,JL)*WF(K,M,N)	002204
60 TU 644	002205
635 60 636 JL = 1,3	002206
636 ..H(K,M,N,N+JL) = PJX(1,JL)*WF(K,M,N-1) + PJX(2,JL)*WF(K,M,N)	002207
A + PJX(3,JL)*WF(K,M,N+1)	002208
..H(K,M,N,N+4) = WF(K,M,N+1)	002209
60 TU 644	002210
640 60 642 JN = JNL,JNU	002211
N = N + 1	002212
NN = NN + 4	002213
60 642 K = 1,3	002214
..H(K,M,N,N) = WF(K,M,N)	002215
60 TU 630	002216
642 CONTINUE	002217
6430 60 6431 K = 1,3	002218
6431 ..H(K,M,N,N+1) = WF(K,M,NFH)	002219
C----RIGHT COLUMN	002220
KTS = 2	002221
MM = MMJ	002222
M = MFH	002223
N = NFE - 1	002224
NN = NHA - 4	002225
JNL = 1	002226
JNU = NFH - NFE	002227
6432 60 644 JN = JNL,JNU	002228
N = N + 1	002229
NN = NN + 4	002230
ISWCH = IDFH(JN)	002231
60 644 K = 1,3	002232
..H(K,M,N,N) = WF(K,M,N)	002233
6434 60 TU (644+630+635+637),ISWCH	002234
644 CONTINUE	002235
6444 N = NFE	002236
NN = NHA	002237
KTS = 1	002238
JML = 1	002239
JMU = MFH - MFE	002240

6445 M = MFE	002241
MM = MHA	002242
DO 645 K = 1,3	002243
645 wH(K,MM,NN) = wF(K,M,N)	002244
DO 648 JM = JML,JMU	002245
DO 646 K = 1,3	002246
DO 646 JL = 1,3	002247
646 wH(K,M+JL,NN) = HJX(1,JL)*WF(K,M-1,N) + HJX(2,JL)*WF(K,M,N)	002248
A + HJX(3,JL)*WF(K,M+1,N) + HJX(4,JL)*WF(K,M+2,N)	002249
M = M + 1	002250
MM = MM + 4	002251
DO 648 K = 1,3	002252
648 wH(K,MM,NN) = wF(K,M,N)	002253
GO TO (6446,709), KTS	002254
6446 KTS = 2	002255
N = NFH	002256
NN = NHT	002257
DO 10 6445	002258
C---INTERPOLATION INTO FINE MESH REGION	002259
C---LEFT COLUMN	002260
70, KTS = 1	002261
M = MCE	002262
MM = MFA	002263
/10 N = NCE - 1	002264
NN = NFA - 2	002265
JNL = 1	002266
JNU = NCII = NCE	002267
712 DO 714 JN = JNL,JNU	002268
N = N + 1	002269
NN = NN + 2	002270
DO 714 K = 1,3	002271
wF(K,MM,NN + 1) = HJX(1,2)*WC(K,N,N-1) + HJX(2,2)*WC(K,M,N)	002272
* + HJX(3,2)*WC(K,M,N+1) + HJX(4,2)*WC(K,M,N+2)	002273
714 wF(K,MM,NN) = WC(K,M,N)	002274
715U DO 7152 K = 1,3	002275
7152 wF(K,MM,NFT) = WC(K,M,NCH)	002276
GO TO (716,720),KTS	002277
C---RIGHT COLUMN	002278
716 KTS = 2	002279
M = MCH	002280
MM = MFT	002281
GO TO 710	002282
C---BOTTOM ROW	002283
720 KTS = 1	002284
N = NCE	002285
NN = NFA	002286
722 M = MCE - 1	002287
MM = MFA - 4	002288
DO 720 JM=MCE,MCG	002289
M = M + 1	002290
MM = MM + 4	002291
DO 726 K = 1,3	002292
wF(K,MM,NN) = WC(K,M,N)	002293
DO 726 JL = 1,3	002294
726 wF(K,MM+JL,NN) = HJX(1,JL)*WC(K,M-1,N) + HJX(2,JL)*WC(K,M,N)	002295
A + HJX(3,JL)*WC(K,M+1,N) + HJX(4,JL)*WC(K,M+2,N)	002296
DO 728 K = 1,3	002297

128	IF(K,MFT,NN) = WC(K,MCH,NN)	002298
GO TO (729,740),KTS		002299
729 KTS = 2		002300
N = NCH		002301
NN = NFT		002302
GO TO 722		002303
740 CONTINUE		002304
C----REGION ADJACENCIES, OVERHANGING (CODE A) POINTS OBTAINED BY INTERPOLA	002305	
C----FROM CARTESIAN MESH	002306	
KUT = 1		002307
LU = LBMA		002308
1200 CALL KAKA(U,KUT,LU,IWMX,CWMX,WM,MFT,NMT,WSM,SCM,LBMMAX)	002309	
GO TO (1210,1214), KUT		002310
1210 KUT = 2		002311
LU = LBMM		002312
GO TO 1208		002313
1214 KUT = 1		002314
LU = LBHA		002315
1216 CALL KAKA(U,KUT,LU,IWHX,CWHX,WH,MFT,IHT,WSH,SCH,LBHMAX)	002316	
GO TO (1218,1220), KUT		002317
1218 KUT = 2		002318
LU = LBHM		002319
GO TO 1216		002320
1220 KUT = 1		002321
LU = LBFA		002322
1222 CALL KAKA(U,KUT,LU,IWFX,CWFX,WF,MFT,NFT,WSF,SCF,LBFMAX)	002323	
GO TO (1224,1240), KUT		002324
1224 KUT = 2		002325
LU = LBFM		002326
GO TO 1222		002327
C----OBTAINING AUXILIARIES FOR LOCAL ROTATED MESHES BY INTERPOLATIO, FRO	002328	
C----CARTESIAN MESHES AND DISTORTED MESHES	002329	
1240 KR = 6		002330
KUT = 1		002331
MLL = LPTR		002332
MUL = LBTC		002333
KSPL = 0		002334
1242 CALL WAKU(IWT,LWT,WST,SCT,DWT,DWTX,LBTMAX,WT,MFT,NFT,	002335	
A WT,MFT,IWT,DWT,IWMT,LINTM)	002336	
GO TO (1244,1246,1250), KUT		002337
1244 KUT = 2		002338
MLL = LPTR		002339
MUL = LFTL		002340
GO TO 1242		002341
1246 KUT = 3		002342
MLL = LPTR		002343
MUL = LPTS		002344
GO TO 1242		002345
1250 KR = 5		002346
KUT = 1		002347
MLL = LBMR		002348
MUL = LPMC		002349
KSPL = 0		002350
1252 CALL WAKU(IWM,CWM,WSM,SCM,DWM,DWMX,LUMMAX,WM,MFT,NMT,	002351	
A WT,MFT,IWT,DWT,IWMT,LINTM)	002352	
GO TO (1254,1260), KUT		002353
1254 KUT = 2		002354

MLL = LPHJ	002355
MUL = LPML	002356
GO TO 1252	002357
1260 KR = 4	002358
KUT = 1	002359
MLL = LBHB	002360
MUL = LPHC	002361
KSPL = 0	002362
1262 CALL WAKU(IWH,CWH,WSH,SCH,DWT,DWAX,LBHMAX,WH,MHT,NHT, A WM,MHT,NHT,DWHM,IWHM,LINTH)	002363
GO TO (1264,1270), KUT	002364
1264 KUT = 2	002365
MLL = LBHJ	002366
MUL = LPML	002367
GO TO 1262	002368
1270 KR = 3	002369
KUT = 1	002370
MLL = LPFF	002371
MUL = LPFD	002372
KSPL = 0	002373
1272 CALL WAKU(IWF,CWF,WSF,SCF,DWF,DWFX,LIFMAX,WF,MFT,NFT, A WH,MHT,NHT,DWFH,IWFH,LINTF)	002374
GO TO (1274,1280), KUT	002375
1274 KUT = 2	002376
MLL = LPFJ	002377
MUL = LPFL	002378
GO TO 1272	002379
1280 CONTINUE	002380
C---KUTTA CONDITION	002381
1420 WF(3,7,FAT,NFC1) = 0.5*(WSF(3,1,LBF0) + WSF(3,3,LBFJ))	002382
DO 1430 KA = 1,2	002383
1430 WF(KA,MFAT,NFC1) = 0.5*(WSF(KA,1,LBF0) - WSF(KA,3,LBFJ))	002384
RETURN	002385
END	002386
	002387
	002388

```

SUBROUTINE KAKUI(WT,MYT,NTY,M,MMT,NMT,WST,IWT,CWT,AWT,SCT,      001496
A LBX,IAT,MNX)                                              001497
C----INTERPOLATES FROM CARTESIAN ONTO TOP (3RD) POINT OF LOCAL SURFACE 001498
C NORMAL AND FROM LOCAL SURFACE NORMAL TO CARTESIAN POINT NEAREST SUR 001499
COMMON/TOPL/MLL,MUL,KP,LIP,KUT,LC,JC,VLBR,NLBR,KSPL           001500
COMMON/PLID/CJX(3,3),DCX(3,6),DCH(3,3),DNBX(5,6),             001501
A DMHY(5,6),LBIX(6),LBIY(6)                                 001502
DIMENSION WT(3,NTT,NTY),RM(3,MMT,NMT),WST(3,6,LBX),            001503
A INT(3,LBX),CWT(9,3,LBX),AWT(3,LBX),SCT(2,LBX),IAT(2,MNX),    001504
B WONT(3)                                         001505
DO 20 M = MLL,MUL                                     001506
C----DETERMINE CENTER OF CARTESIAN REFERENCE ARRAY          001507
MK = IWT(LIP,M)/100                                    001508
NK = IWT(LIP,M) - 100*MK - 2                         001509
MK = MK - 2                                         001510
C----INTERPOLATE FROM CARTESIAN MESH TO TOP POINT OF LOCAL SURFACE NORMA 001511
DO 4 KA = 1,3                                         001512
WONT(KA) = 0.0                                         001513
DO 4 LW = 1,9                                         001514
JB = (LW+2)/3                                         001515
JA = LW - 3*(JB-1)                                     001516
4 WONT(KA) = WONT(KA) + CWT(LW,LIP,M)*WT(KA,MK+JA,NK+JB)   001517
WST(3,JC,N) = WONT(3)                                 001518
WST(1,JC,N) = WONT(1)*SCT(2,M) + WONT(2)*SCT(1,M)        001519
WST(2,JC,N) = -WONT(1)*SCT(1,M) + WONT(2)*SCT(2,M)       001520
GO TO (6,8+10),KUT                                     001521
U MK = M - LC                                         001522
NK = IAT(2*MK)                                         001523
GO TO 12                                         001524
U MK = LC - M                                         001525
NK = IAT(1*MK)                                         001526
GO TO 12                                         001527
U NK = M - LC                                         001528
MK = IAT(NK)                                         001529
C----INTERPOLATE FROM SURFACE NORMAL TO CARTESIAN POINT NEAREST THE SURF 001530
12 DO 14 KA = 1,3                                     001531
14 WONT(KA) = AWT(1,M)*WST(KA,2,M) + AWT(2,M)*WST(KA,5,M)   001532
A + AWT(3,M)*WST(KA,8,M)                           001533
WT(3,MK,NK) = WONT(3)                                001534
WT(1,MK,NK) = WONT(1)*SCT(2,M) - WONT(2)*SCT(1,M)        001535
WT(2,MK,NK) = WONT(1)*SCT(1,M) + WONT(2)*SCT(2,M)       001536
1COIN = MK - 1 - LUIA(KR)*(MK-1)/LBIX(KR)           001537
1F(1COIN) 20,16,20                                     001538
10 JCWIN = NK - 1 - LBIY(KR)*(NK - 1)/LBIX(KR)        001539
1F(JCWIN) 20,17,20                                     001540
17 ML = (MK-1)/LBIX(KR) + MLBR                      001541
NL = (NK-1)/LBIY(KR) + NLBR                      001542
DO 18 KA = 1,3                                         001543
18 WM(KA,ML,NL) = WT(KA,MK,NK)                      001544
20 CONTINUE                                         001545
RETURN                                              001546
END                                                 001547

```

```

----- SUBROUTINE WAKO1WT,CWT,WST,SCT,DWT,DWTX,LBX,W1,MTT,NTT, -----
A MU,MUT,NUT,DW1U,IWTU,LINTU      001548
----- C---- INTERPOLATES FROM CARTESIAN TO POINTS 7,9 OF LOCAL ROTATED MESH. 001550
C---- FROM DISTORTED MESH INTO POINTS 3,6 OF ADJACENT (TO LEFT) MESH AND 001551
C---- POINTS 1,4 OF ADJACENT (TO RIGHT) MESH. ALSO PICKS UP POINTS 1,4 001552
C---- UN 3D FOR MESHES AT ENDS OF RANGE AND PUTS IN BORDER CARTESIAN POI 001553
C---- FOR FINER MESH (CODE F)          001554
COMMON/T0PL/MLL,MUL,KR,LIP,KUT,LC,JC,MUTR,NLBR,KSPL 001555
DIMENSION IWT(3,LBX), CWT(9,3,LBX), WST(3,9,LBX), SCT(2,LBX), 001556
A WT(3,4,LBX), DWTX(4,2,6), WT(3,NTT,NTT), WU(3,MUT,NUT), 001557
B DWU(4,2,4), IWTU(2,4), LINTU(2), WONT(3), WINT(3,9) 001558
DO 50 MM = MLL,MUL               001559
----- C---- UPPER ROW OBTAINED FROM CARTESIAN MESH             001560
DO 16 LIP = 1,3,2                 001561
JC = 6 + LIP                      001562
NK = IWT(LIP,MM)/100              001563
IK = IWT(LIP,MM) - 100*MK - 2    001564
MK = NK - 2                        001565
DO 14 KA = 1,3
WONT(KA) = 0.0                     001566
DO 14 LW = 1,9                     001567
JU = (LK+2)/3                     001568
JA = LW - 3*(JU-1)                001569
----- 14 WONT(KA) = WONT(KA) + CWT(LK+LIP,MM)*WT(KA,MK+JA,NK+JA) 001571
WST(3,JC,LM) = WONT(3)            001572
WST(1,JC,LM) = WONT(1)*SCT(2,MM) + WONT(2)*SCT(1,MM) 001573
16 WST(2,JC,LM) = -WONT(1)*SCT(1,MM) + WONT(2)*SCT(2,MM) 001574
----- C---- UNIFORM w CONTENT FOR DISTORTED MESH             001575
DO 20 JA = 1,3                     001576
MU = MM - 2 + JA                  001577
DO 20 JU = 1,3                     001578
JC = 3*(JU-1) + JA                001579
JU = -1 + 3*JB                     001580
WINT(3,JC) = WST(3,JU,MU)        001581
WINT(1,JC) = WST(1,JU,MU)*SCT(2,MU) - WST(2,JU,MU)*SCT(1,MU) 001582
20 WINT(2,JC) = WST(1,JU,MU)*SCT(1,MU) + WST(2,JU,MU)*SCT(2,MU) 001583
----- C---- OTHER COORDINATES OF ADJACENT ROTATED MESHES OBTAINED FROM DISTORTED ME 001584
DO 24 LIP = 1,4                   001585
MU = MM - 3 + 2*((LIP+1)/2)       001586
JC = 8 - 2*((LIP+1)/2) - 3*(LIP-2*(LIP/2)) 001587
DO 22 KA = 1,3                   001588
WONT(KA) = 0.0                     001589
DO 22 LW = 1,9                   001590
22 WONT(KA) = WONT(KA) + DW1U(LW+LIP,MM)*WINT(KA,LW) 001591
WST(3,JC,MU) = WONT(3)           001592
WST(1,JC,MU) = WONT(1)*SCT(2,MU) + WONT(2)*SCT(1,MU) 001593
24 WST(2,JC,MU) = -WONT(1)*SCT(1,MU) + WONT(2)*SCT(2,MU) 001594
----- C---- AUXILIARIES AT ENDS OF OPERATING RANGES           001595
1F(MM-MLL) 28,28,26               001596
26 IF(MM=MUL) 50,50,50           001597
28 JU = -2                         001598
GO TO 32                           001599
30 JU = 0                           001600
32 KSPL = KSPL + 1                 001601
MU = MM                           001602
DO 36 LIP = 1,2                   001603

```

JC = JU + 3*LIP	001604
DO 34 KA = 1,3	001605
WONT(KA) = 0.0	001606
DO 34 LW = 1,9	001607
34 WONT(KA) = WONT(KA) + DWTX(LW,LIP,KSPL)*WINT(KA,LW)	001608
AST(3,JC,MU) = WONT(3)	001609
AST(1,JC,MU) = WONT(1)*SCT(2,MU) + WONT(2)*SCT(1,MU)	001610
36 AST(2,JC,MU) = -WONT(1)*SCT(1,MU) + WONT(2)*SCT(2,MU)	001611
40 IF(KK=6) 42,50,50	001612
C-----FINER CARTESIAN MESH AUXILIARIES ALONG BORDER (CODE F)	001613
42 DO 10 (40,50,50,40,50,50),KSPL	001614
40 LUP = LINTU(KUT)	001615
MU = INTU(KUT,1)/100	001616
DO 48 LIP = 1,LUP	001617
MU = INTU(KUT,LIP) - 100*MU	001618
DO 48 KA = 1,3	001619
WU(KA,MU,MU) = 0.0	001620
DO 40 LW = 1,9	001621
48 WU(KA,MU,LW) = WU(KA,MU,NMU) + DWTU(LW,KUT,LIP)*WINT(KA,LW)	001622
50 CONTINUE	001623
RETURN	001624
END	001625

```

----- SUBROUTINE KAKARU(KUT,LU,IWMX,CWMX,WM,NMT,WSM,SCM,CBXT ----- 001626
C----- INTERPOLATES FROM CARTESIAN MESH TO TOP 2 POINTS OF SURFACE NORMAL 001627
C----- EDGE OF RANGE (CODE A) 001628
DIMENSION IWMX(2,2), CWMX(2,2), WM(3,NMY,NMT), WSM(3,9,LU), 001629
A SCM(2,LU), WONT(3) 001630
DO 10 LIP = 1,2 001631
LUP = 2 + 3*LIP 001632
MK = IWMX(LIP,KUT)/100 001633
NK = IWMX(LIP,KUT) - 100*MK - 2 001634
MK = MK - 2 001635
DO 14 KA = 1,3 001636
WONT(KA) = 0.0 001637
DO 14 LW = 1,9 001638
JH = (LW+2)/3 001639
JA = LW - 3*(JH-1) 001640
14 WONT(KA) = WONT(KA) + CWMX(LW,LIP,KUT)*WM(KA,MK+JA,NK+JH) 001641
WSM(3,LUP,LU) = WONT(3) 001642
15 WSM(1,LUP,LU) = WONT(1)*SCM(2,LU) + WONT(2)*SCM(1,LU) 001643
16 WSM(2,LUP,LU) = -WONT(1)*SCM(1,LU) + WONT(2)*SCM(2,LU) 001644
      RETURN 001645
      END 001646
-----
```

```

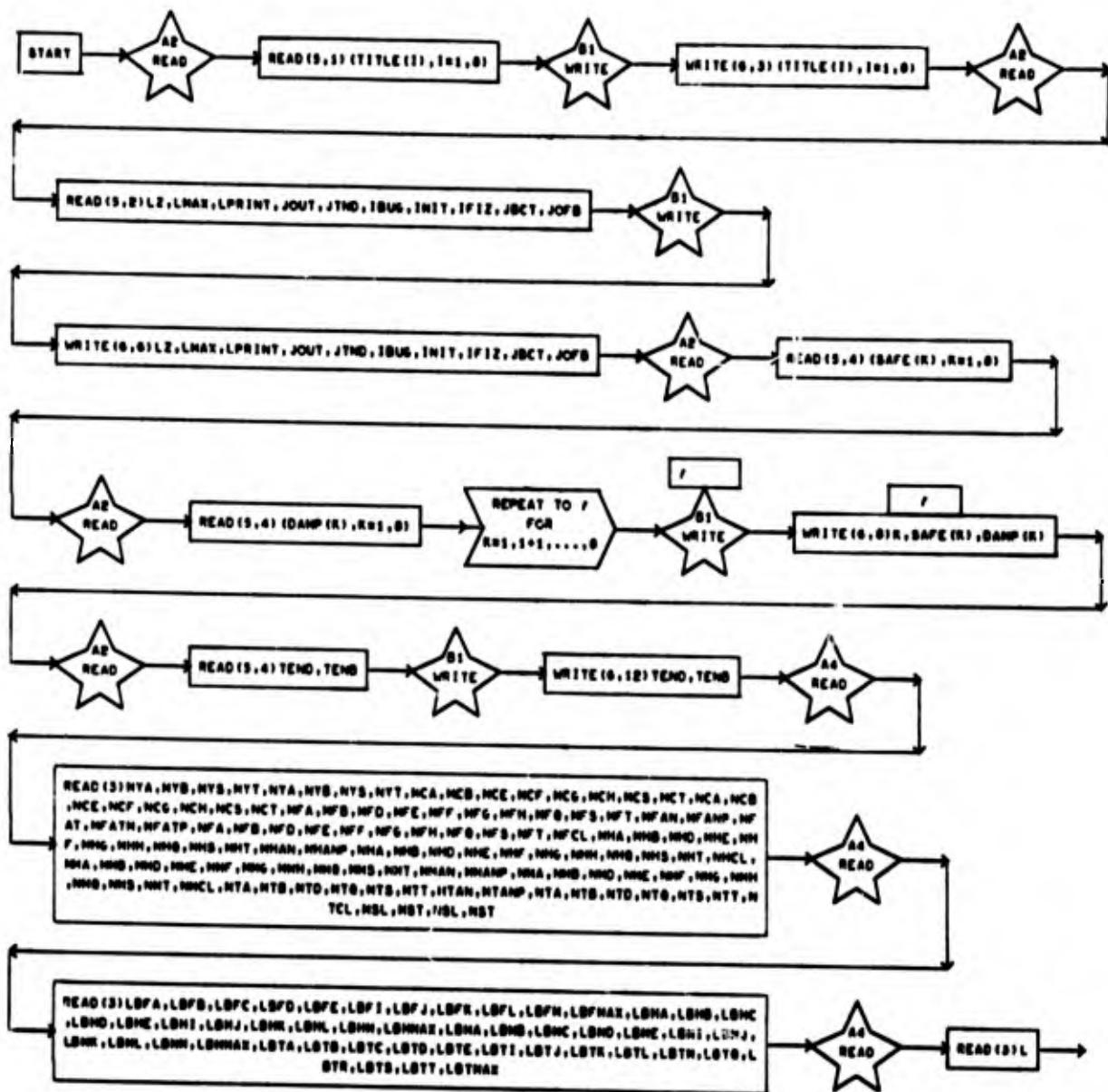
SUBROUTINE TREND          001647
C---- THIS SUBROUTINE TRIES TO PUSH CONDITIONS AT ALL MESH POINTS TOWARD 001648
C STEADY STATE FREE-STREAM TOTAL ENTHALPY. DISCREPANCIES ARE REDUCED 001649
C (1.0 - TEND) OF THEIR VALUE AT ALL REGULAR MESH AND BOUNDARY POINTS 001650
C
C COMMUN/GAS/ GMU, HGMU, UNOG, UOGMU, GDGMU, HINF, GMU, UNOGMU      001651
C COMMUN/ENTHP/ JTND, TEND, TENB                                         001652
C COMMUN   WY(3,42,9), WC(3,17,17), WF(3,37,13), WH(3,25,17),      001653
C A WM(3,26,26), WT(3,16,21), WSF(3,9,42), WSM(3,9,22), WSM(3,9,14), 001654
C B WS1(3,9,43), WSU(3)                                              001655
C
C DO 10 LM = 1, 3674                                                 001656
C LK = 3*LM                                                       001657
C R = 1.0/WY(LK)                                                 001658
C UX = WY(LK-2)*R                                              001659
C UY = WY(LK-1)*R                                              001660
C E = WY(LK)**GMU                                             001661
C ITUT = 2.5*E + 0.5*(UX**2 + UY**2)                                001662
C EFAC = 1.0 - TEND + TEND*HINF/HTOT                               001663
C WY(LK) = (E*EFAC)**UNOGMU                                         001664
C UFAC = WY(LK)*SQRT(EFAC)                                         001665
C WY(LK-2) = UFAC*UX                                              001666
C WY(LK-1) = UFAC*UY                                              001667
C RETURN
C END

```

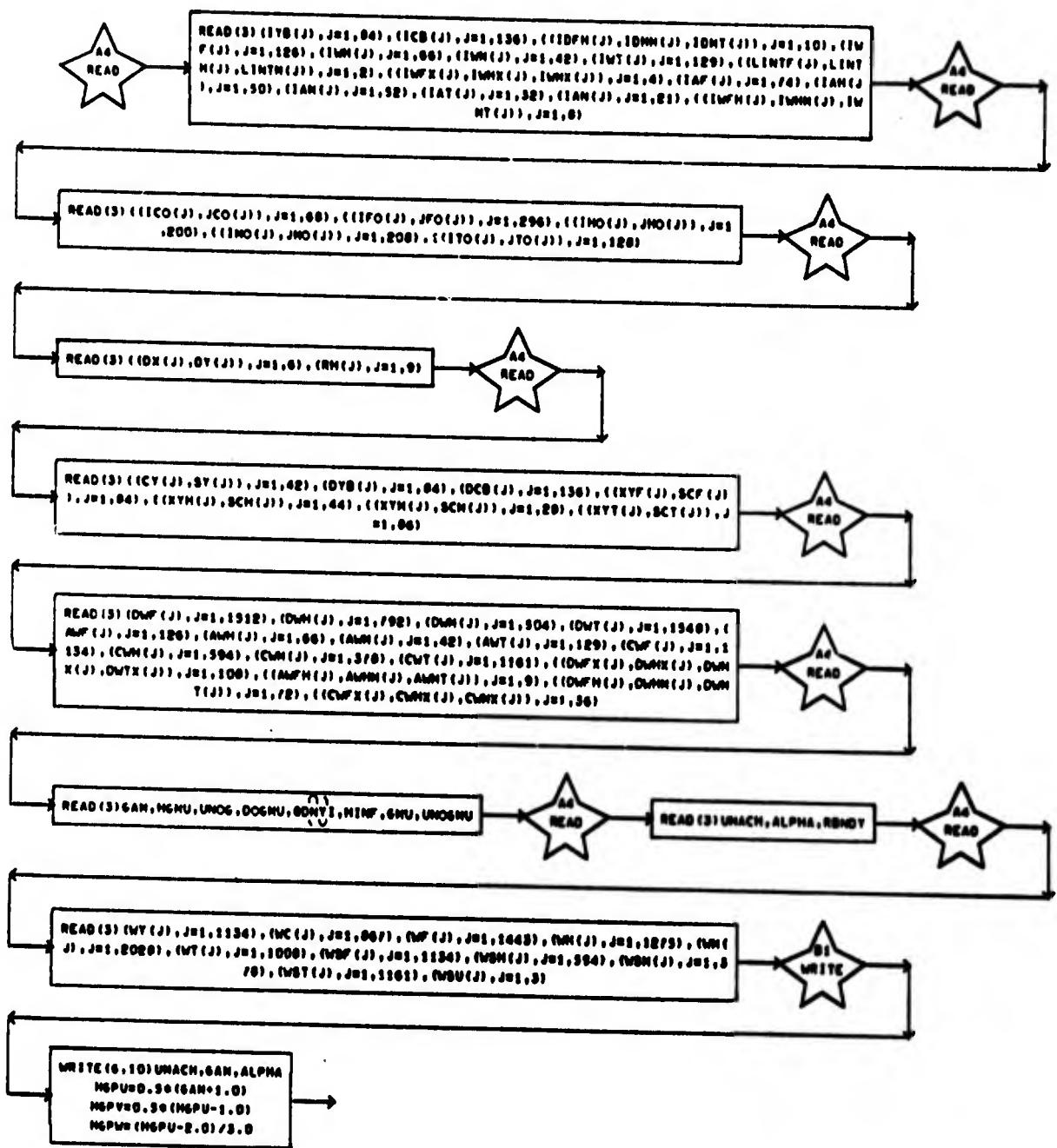
APPENDIX V

FLOW FIELD PROGRAM FLOW CHARTS

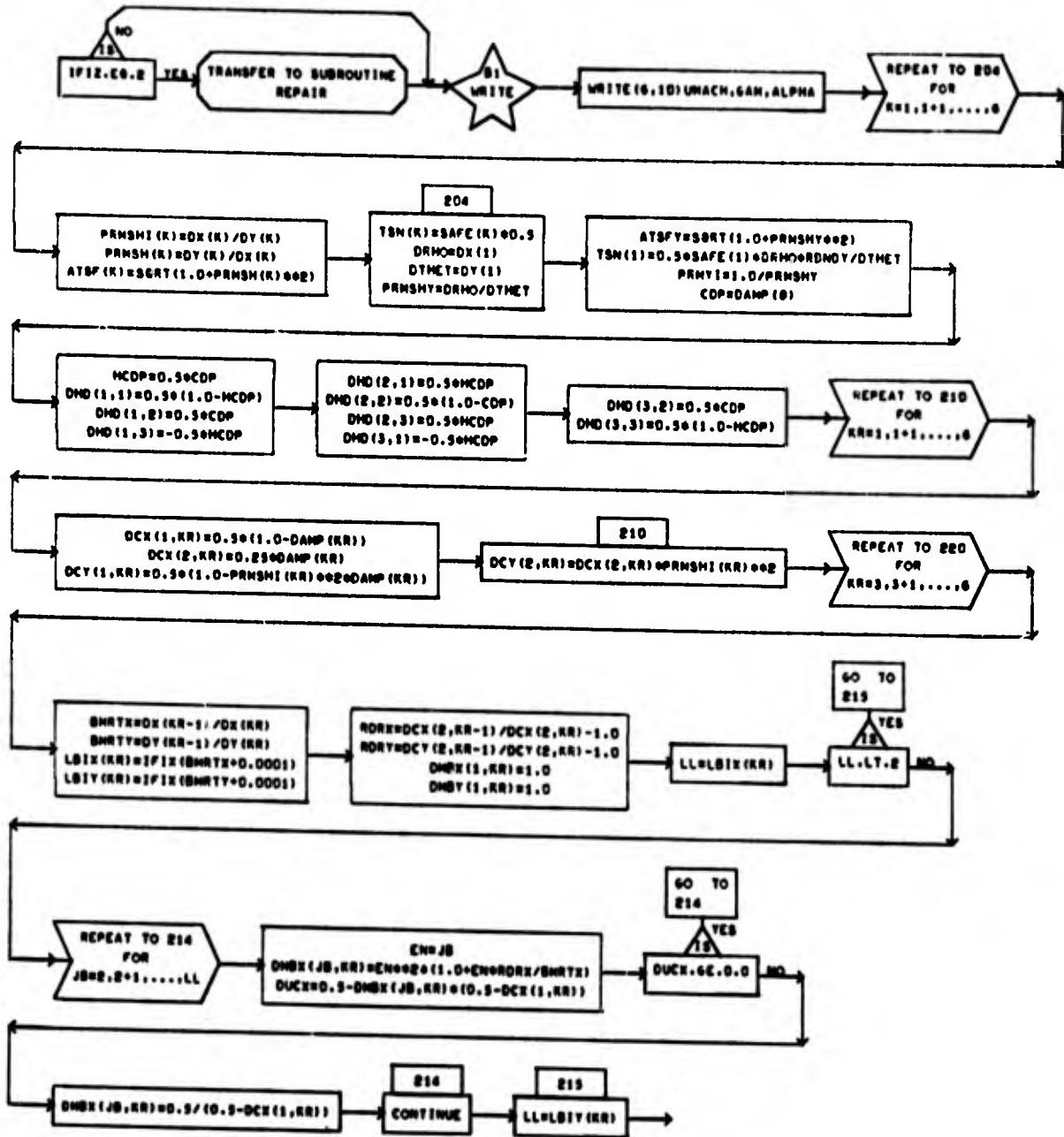
PROGRAM NAME (INPUT,OUTPUT,TAPES=INPUT,TAPE6=OUTPUT,TAPES3,TAPES4)



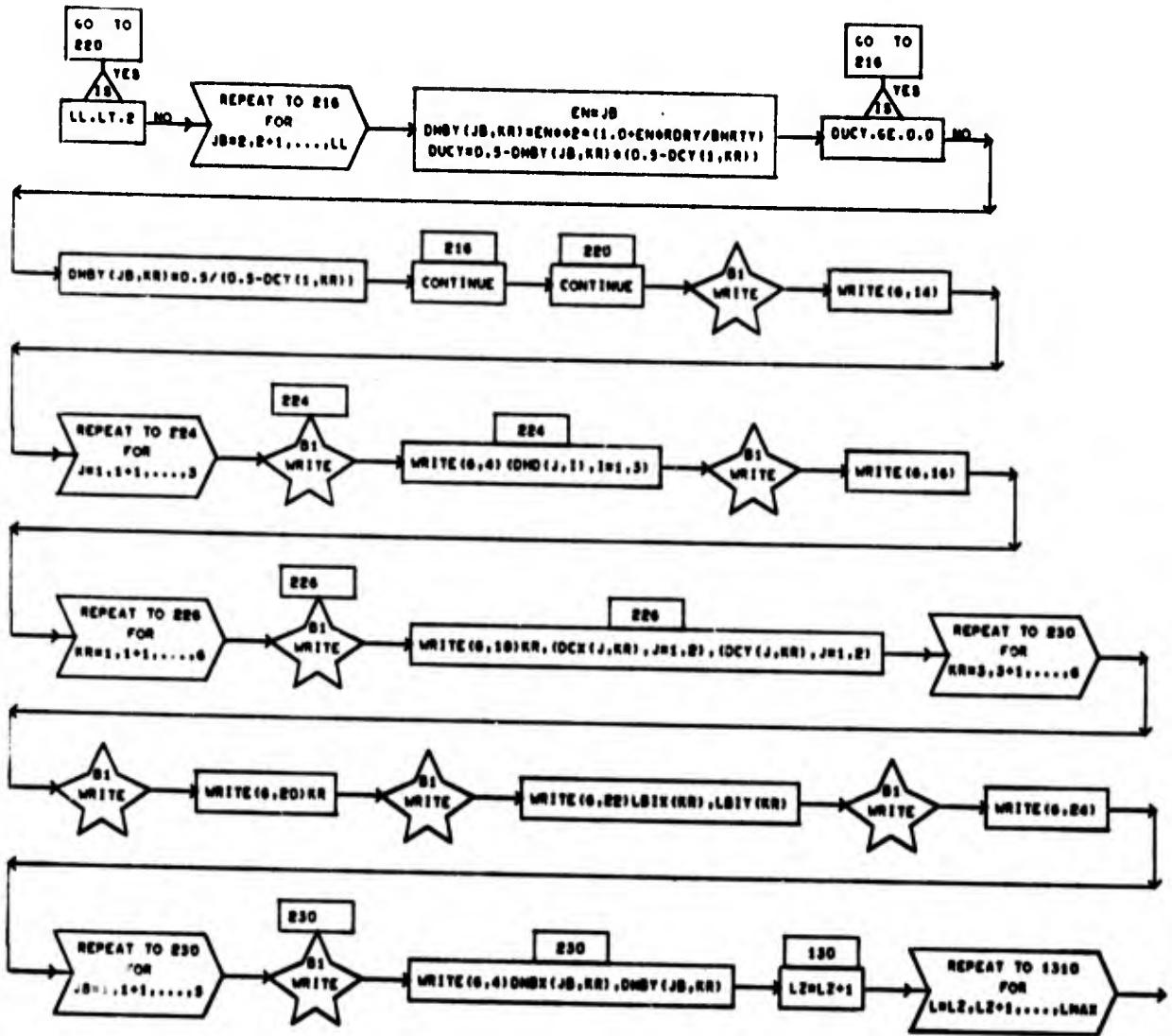
Detailed Flow Charts of Subroutine HANE (1 of 7)



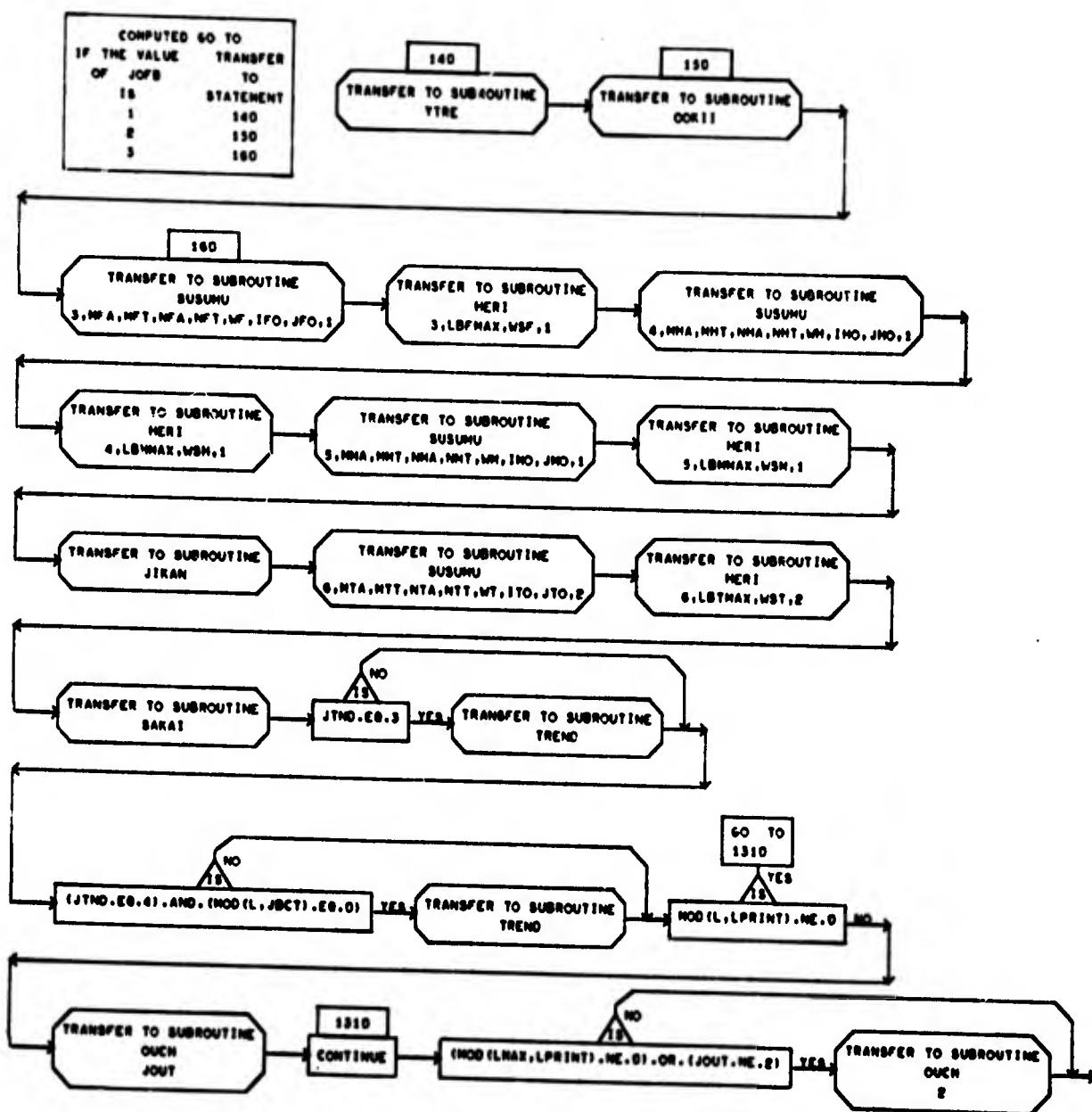
Detailed Flow Charts of Subroutine HANE (2 of 7)



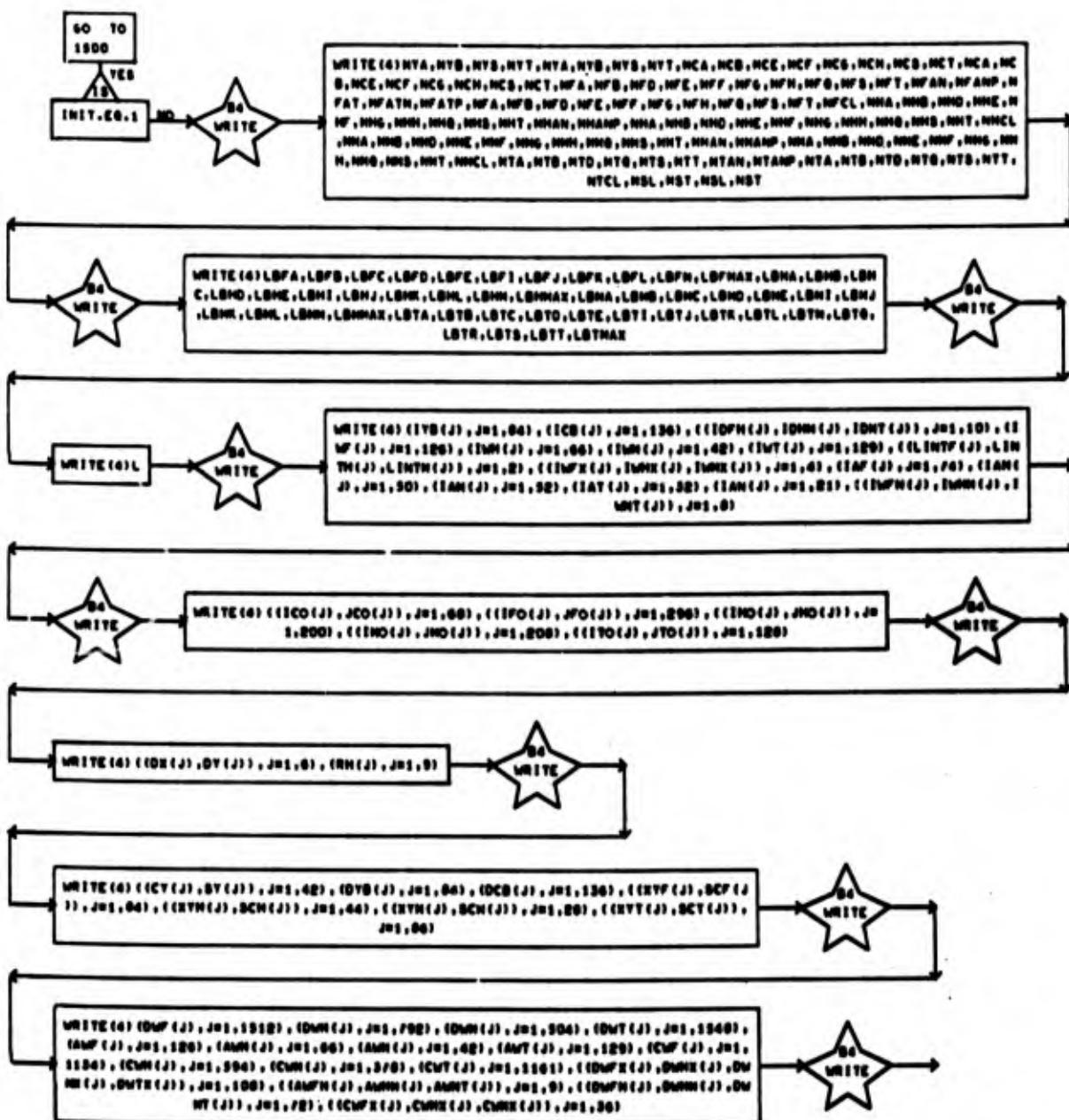
Detailed Flow Charts of Subroutine HANE (3 of 7)



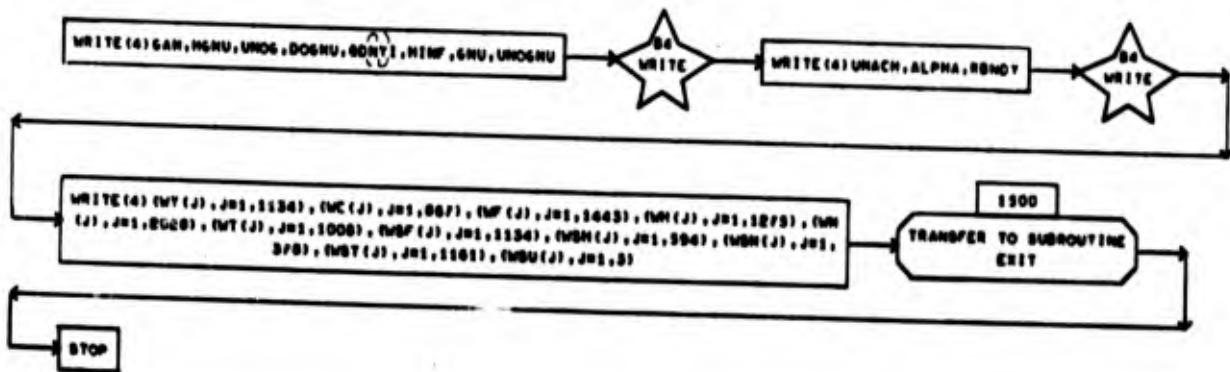
Detailed Flow Charts of Subroutine HANE (4 of 7)



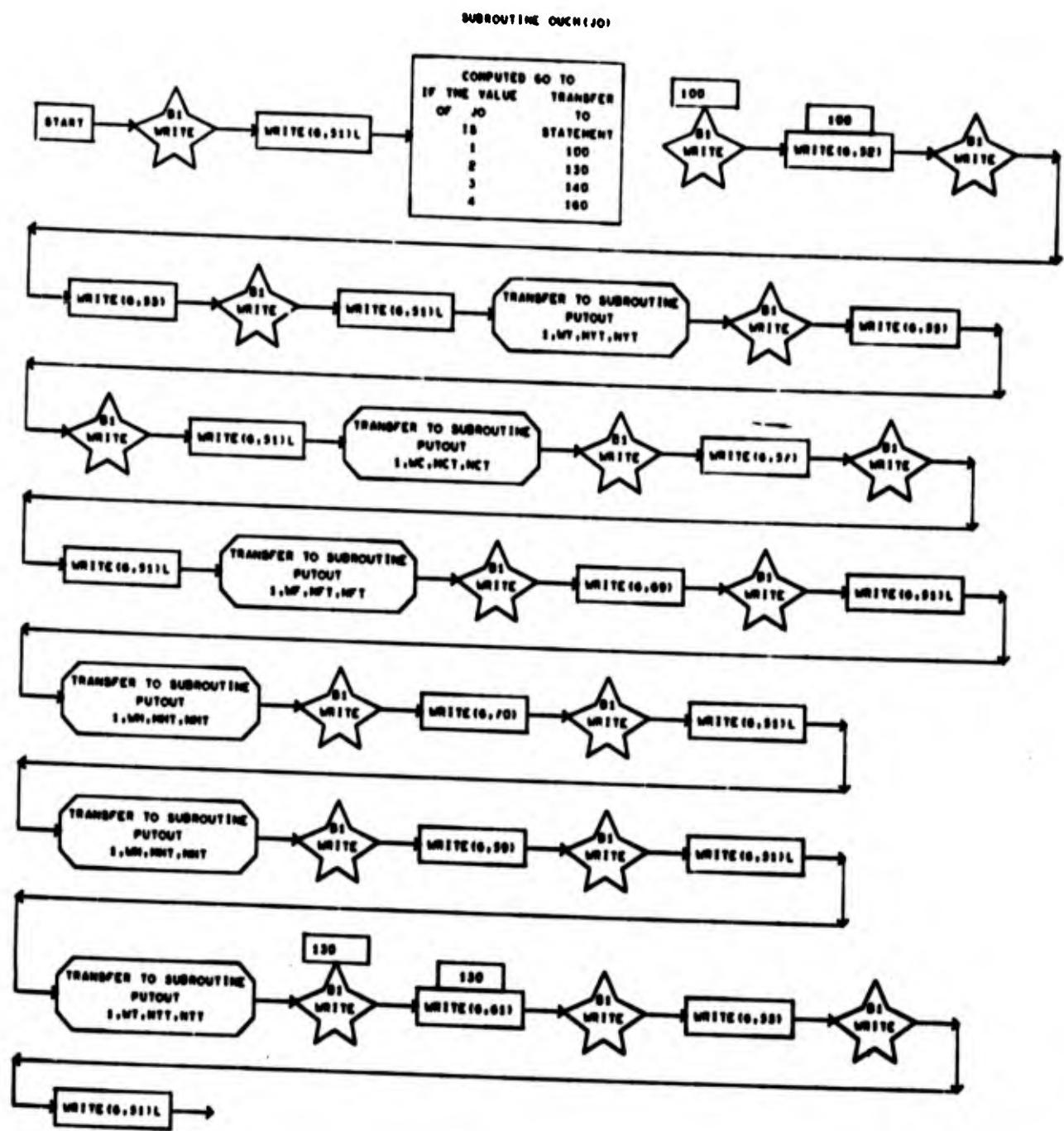
Detailed Flow Charts of Subroutine HANE (5 of 7)



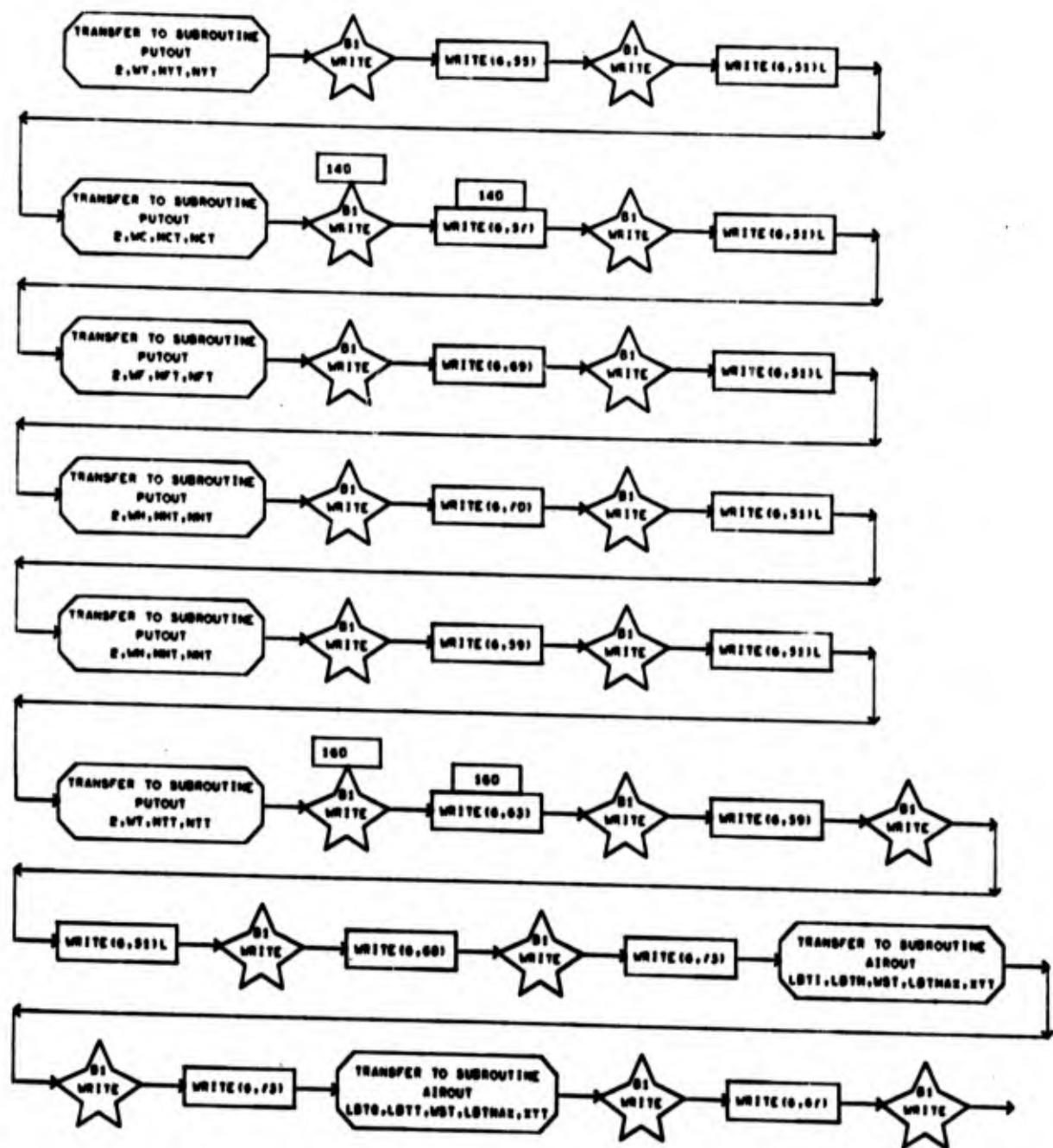
Detailed Flow Charts of Subroutine HANE (6 of 7)



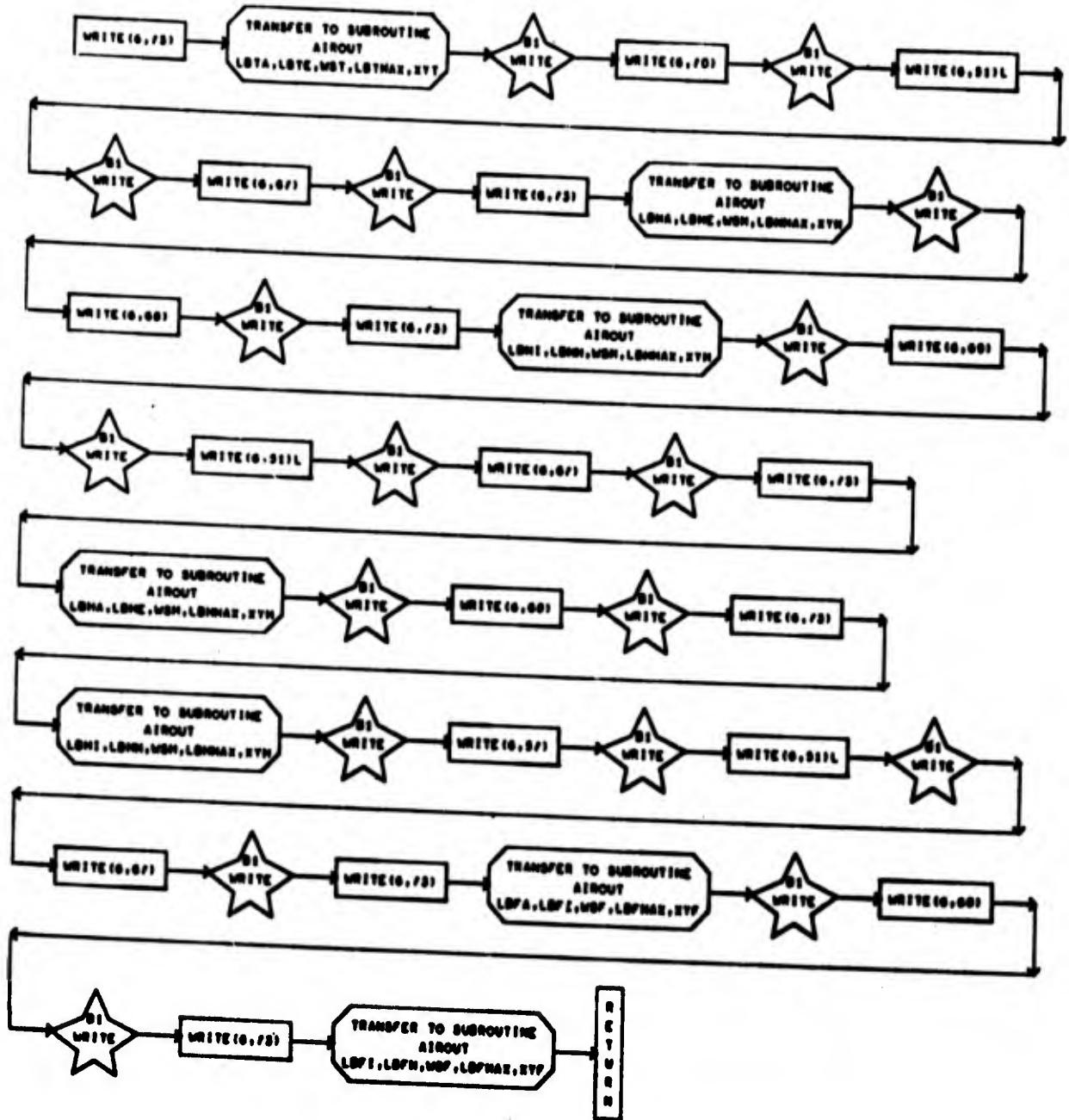
Detailed Flow Charts of Subroutine HANE (7 of 7)



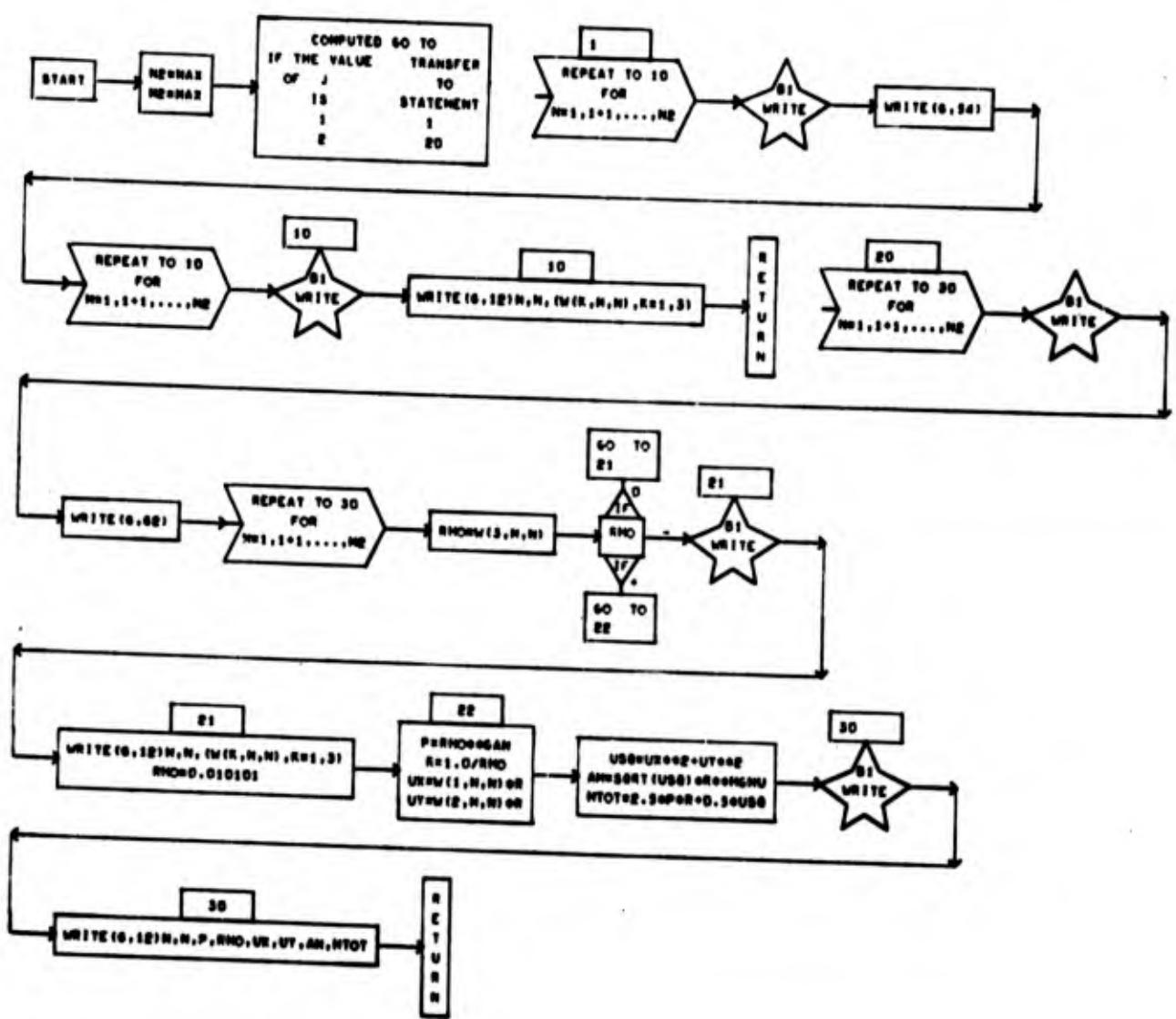
Detailed Flow Charts of Subroutine OUCH (1 of 3)



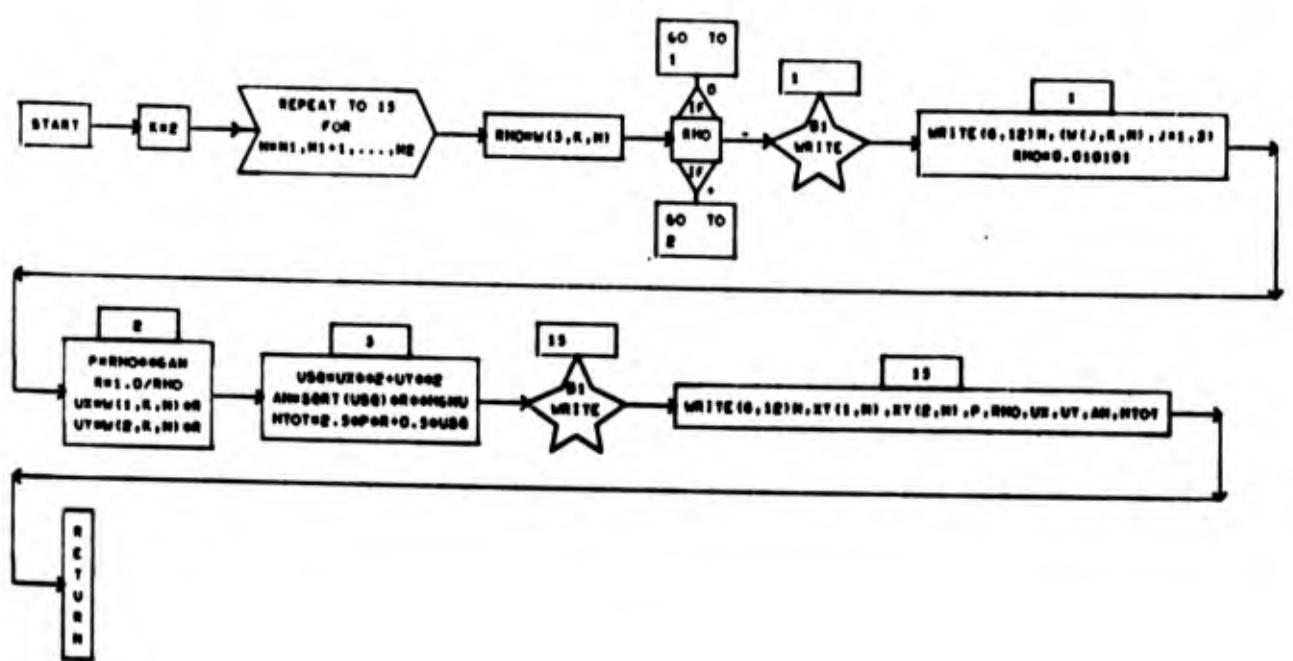
Detailed Flow Charts of Subroutine OUCH (2 of 3)



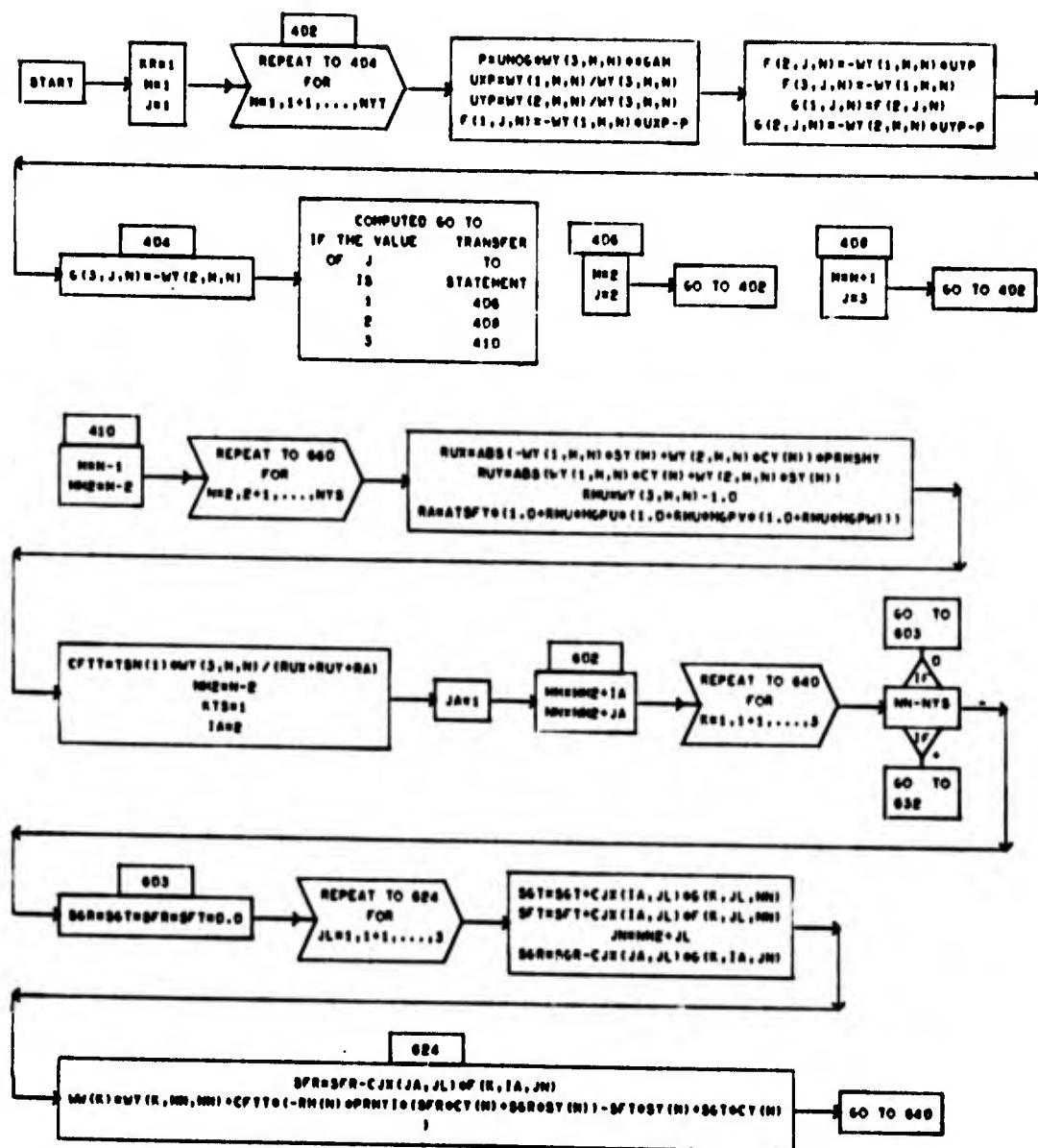
Detailed Flow Charts of Subroutine OUCH (3 of 3)

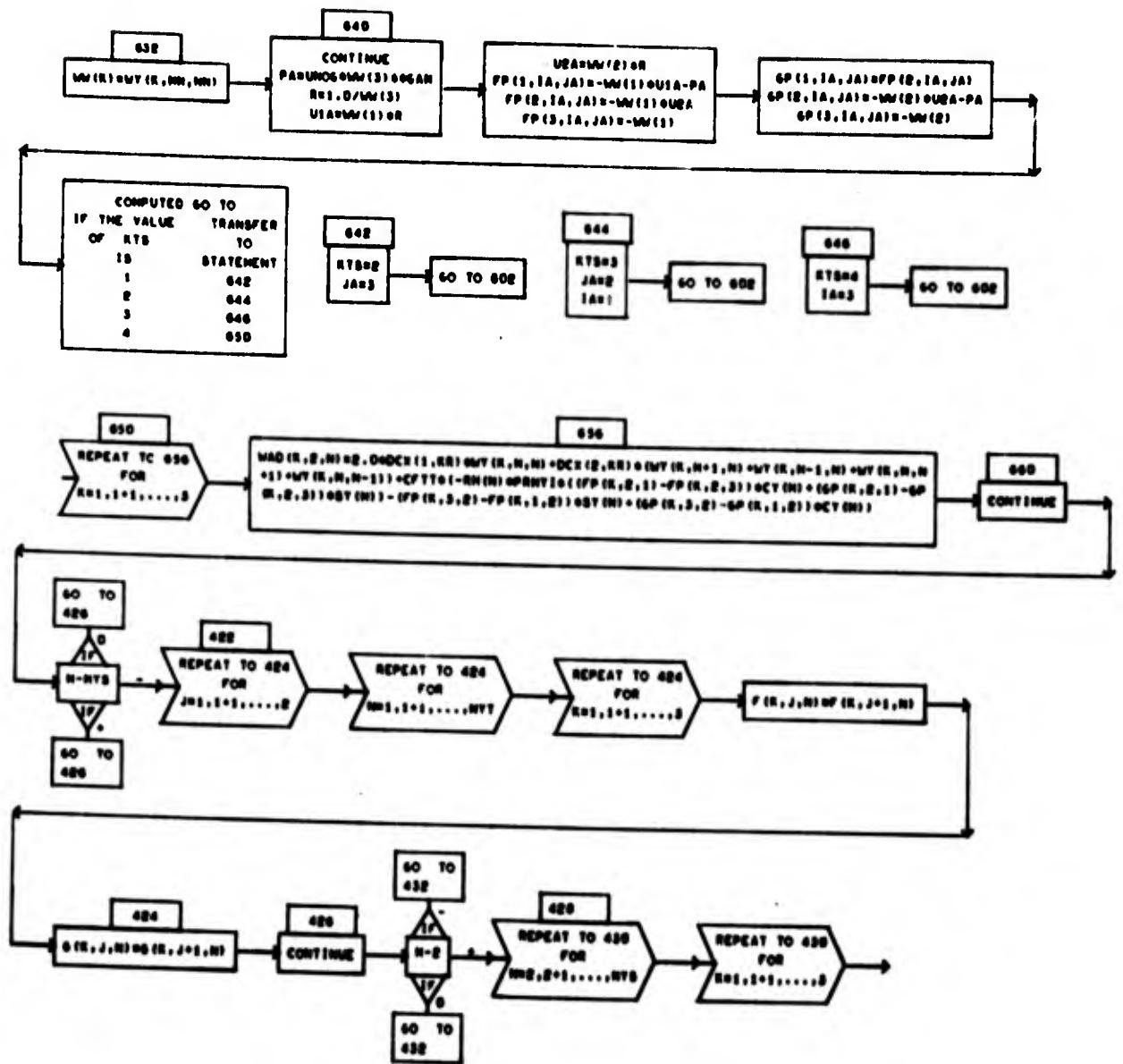


Detailed Flow Chart of Subroutine PUTOUT

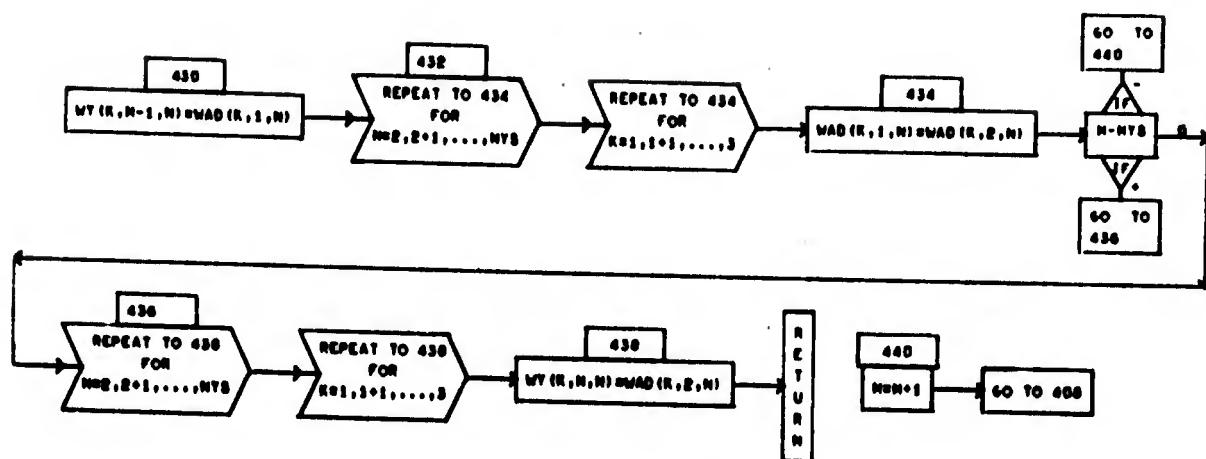


Detailed Flow Charts of Subroutine AIROUT

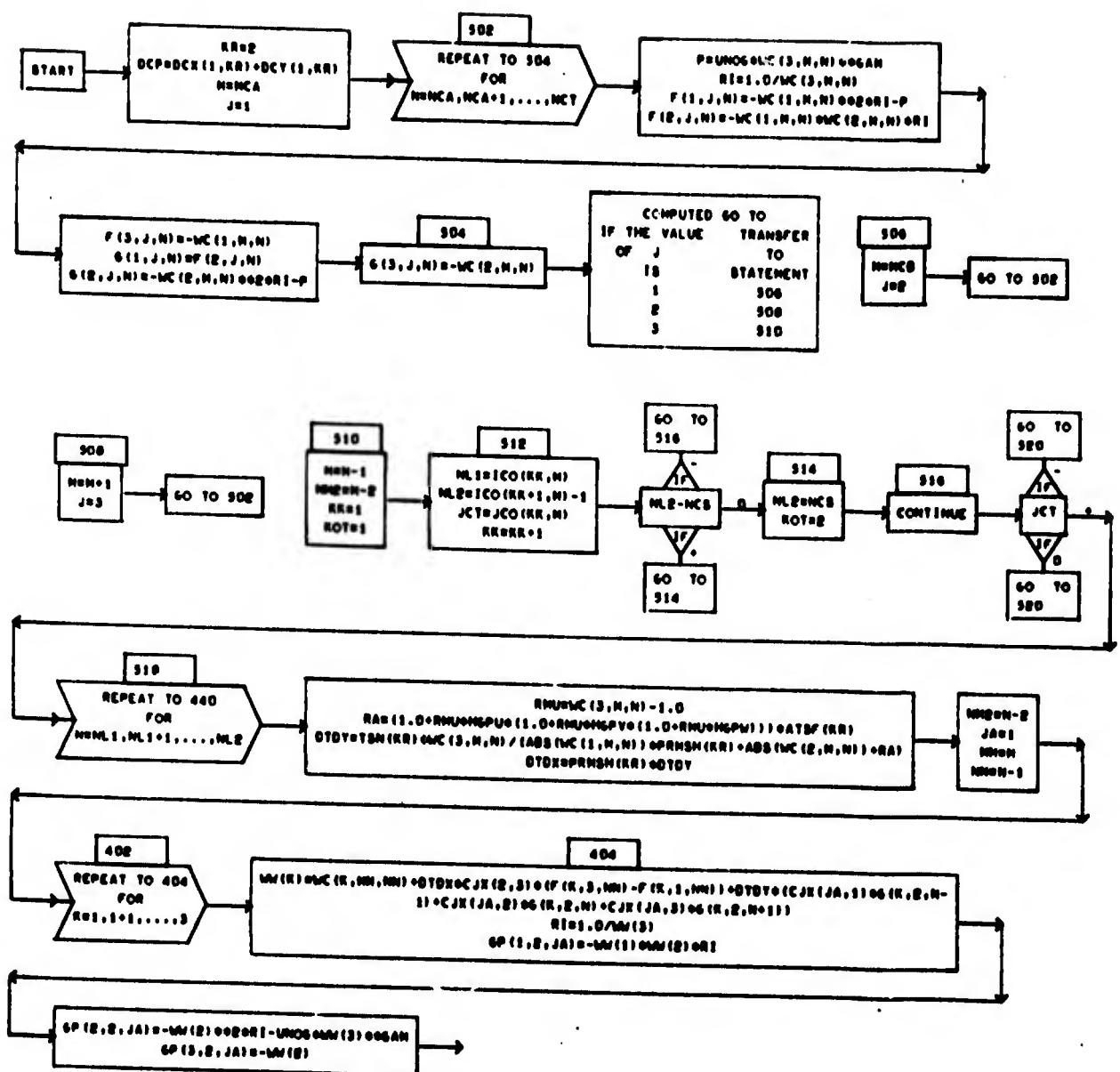




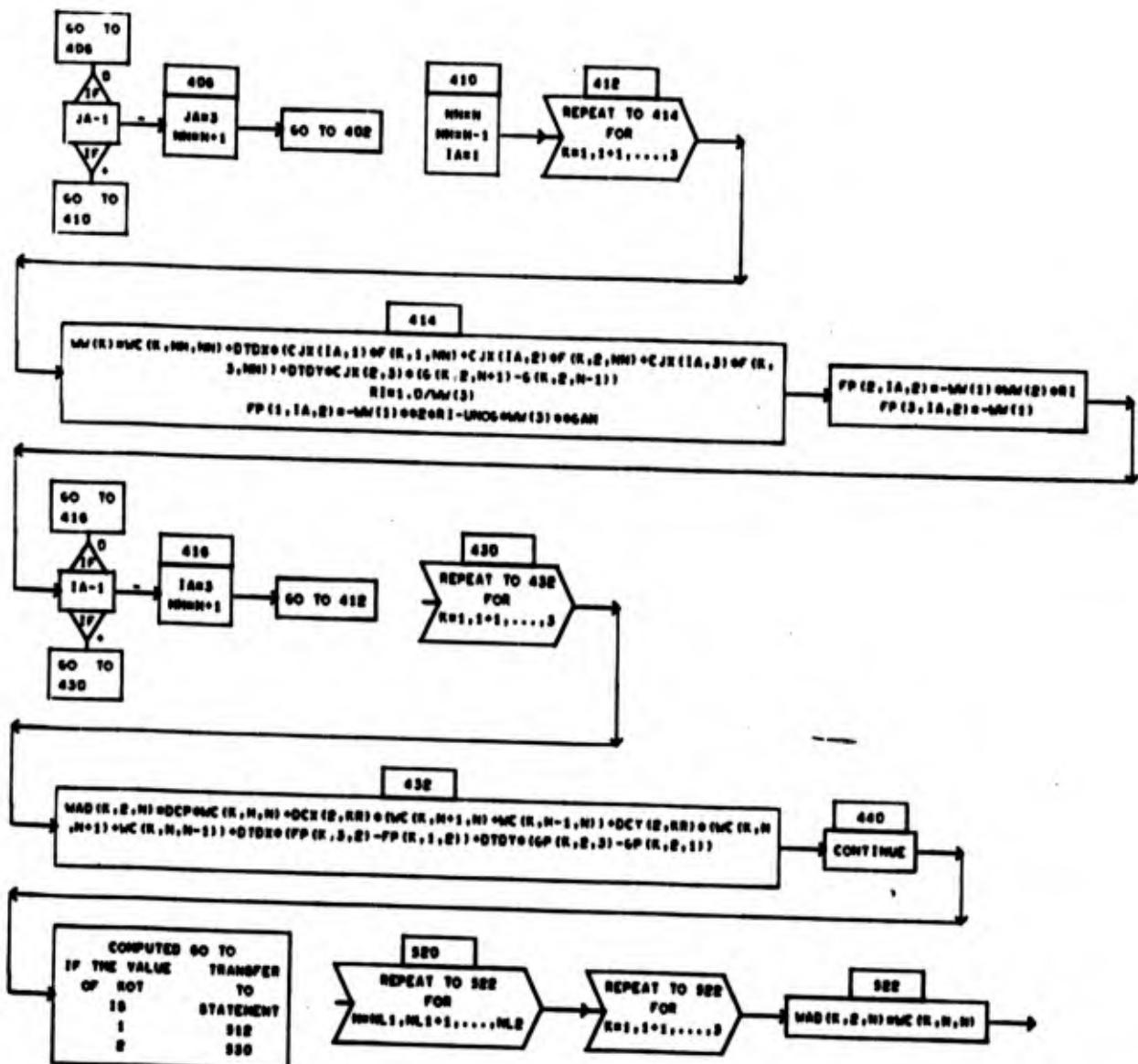
Detailed Flow Charts of Subroutine YTRE (2 of 3)



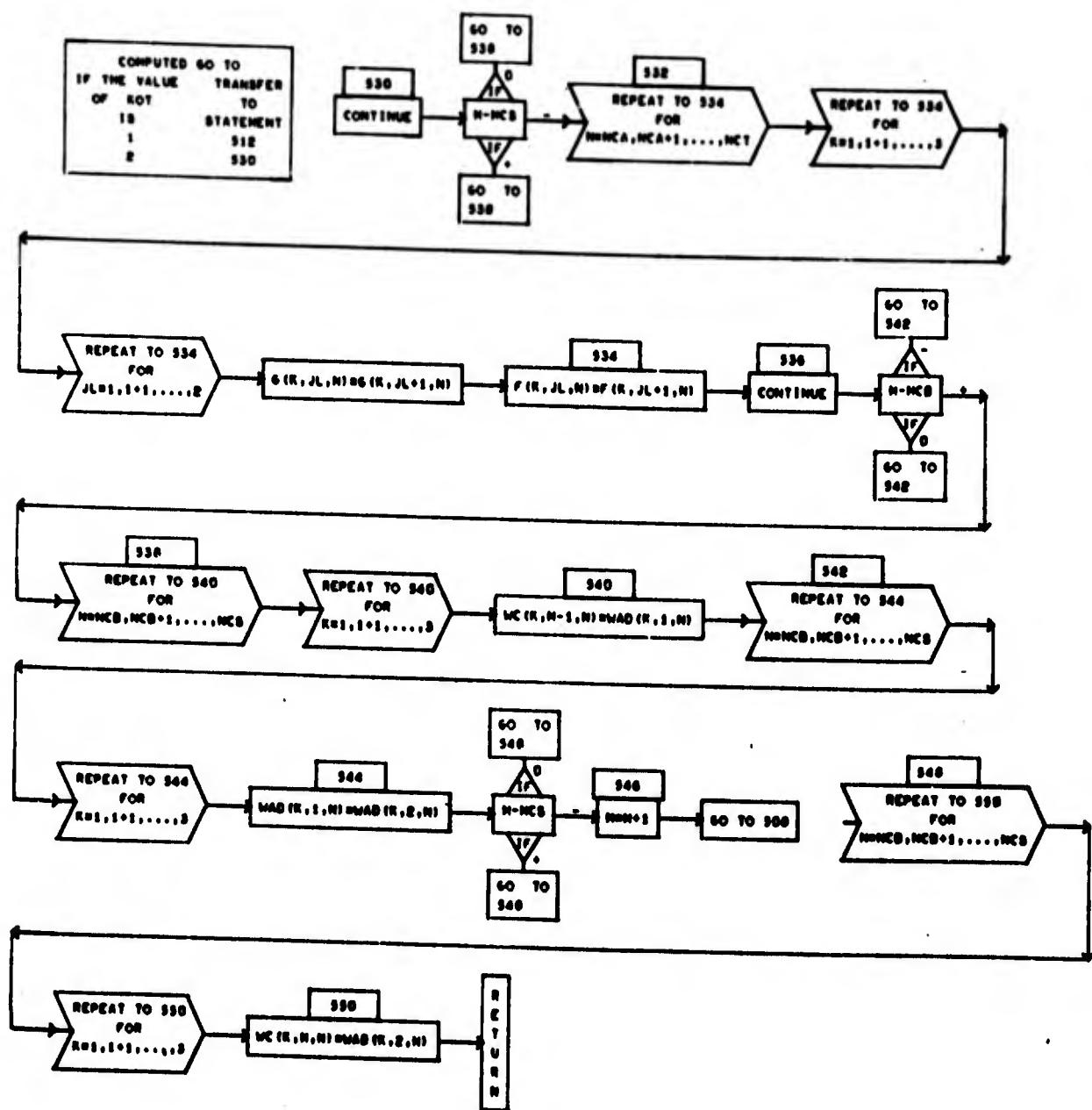
Detailed Flow Charts of Subroutine YTRE (3 of 3)



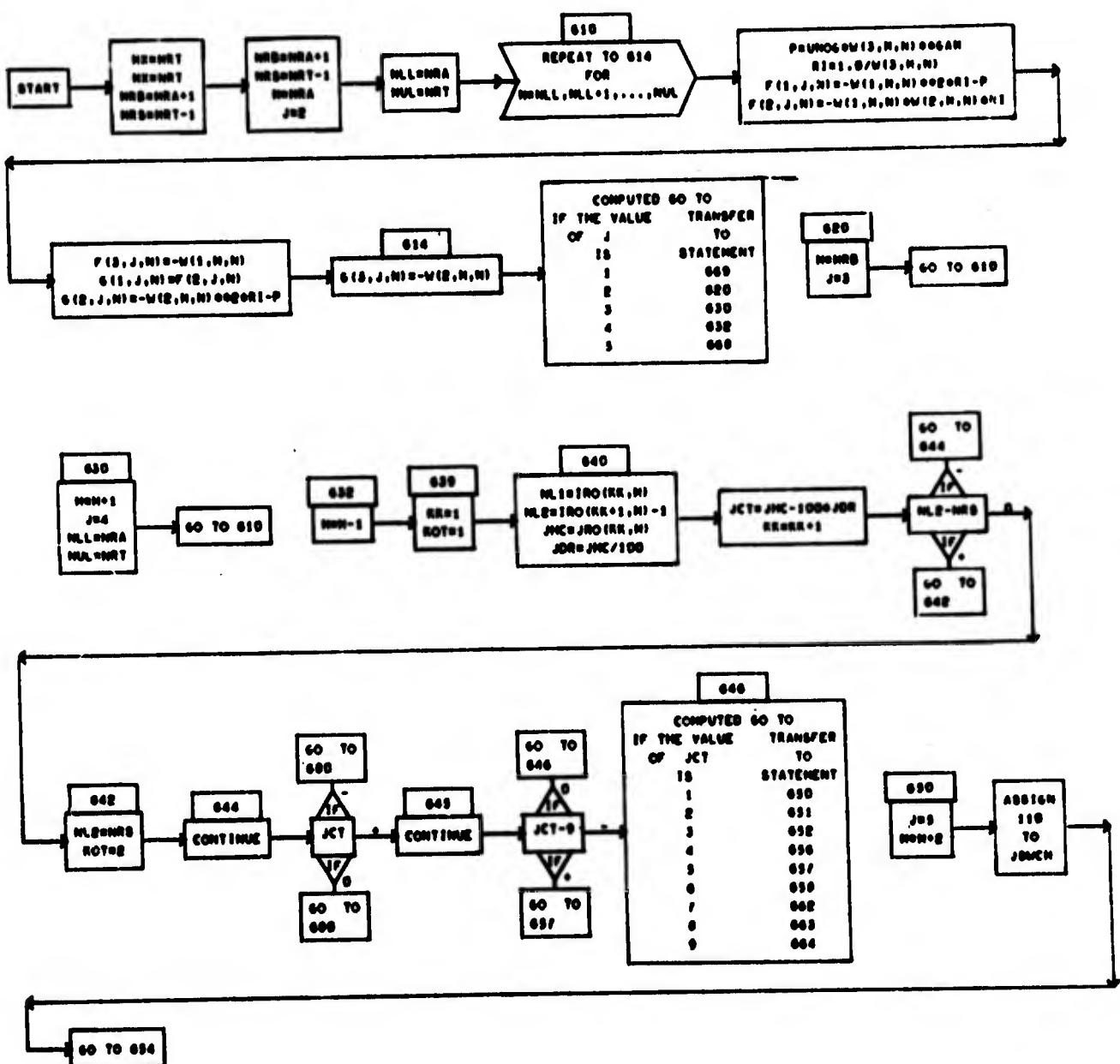
Detailed Flow Charts of Subroutine 00KII (1 of 3)



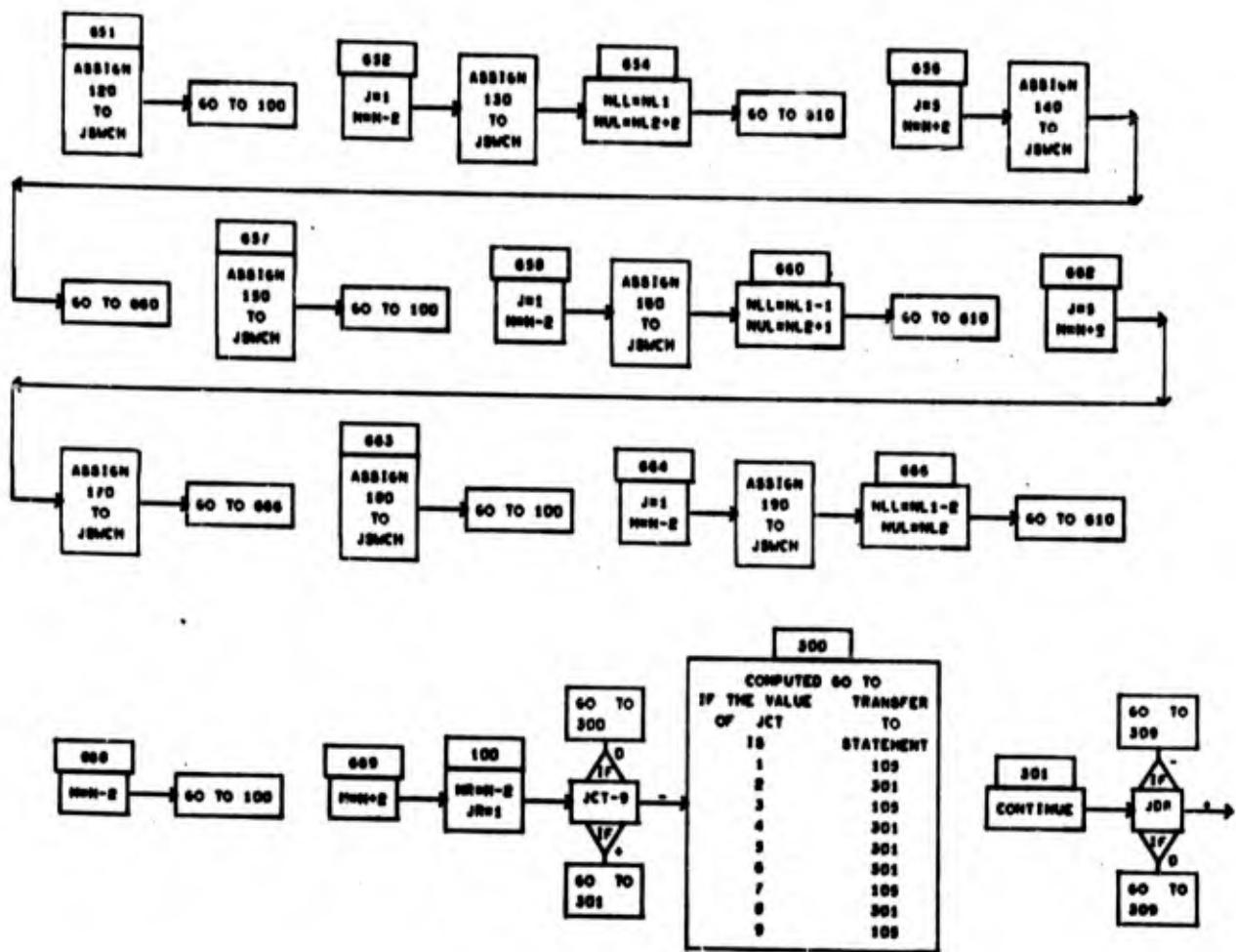
Detailed Flow Charts of Subroutine 00KII (2 of 3)



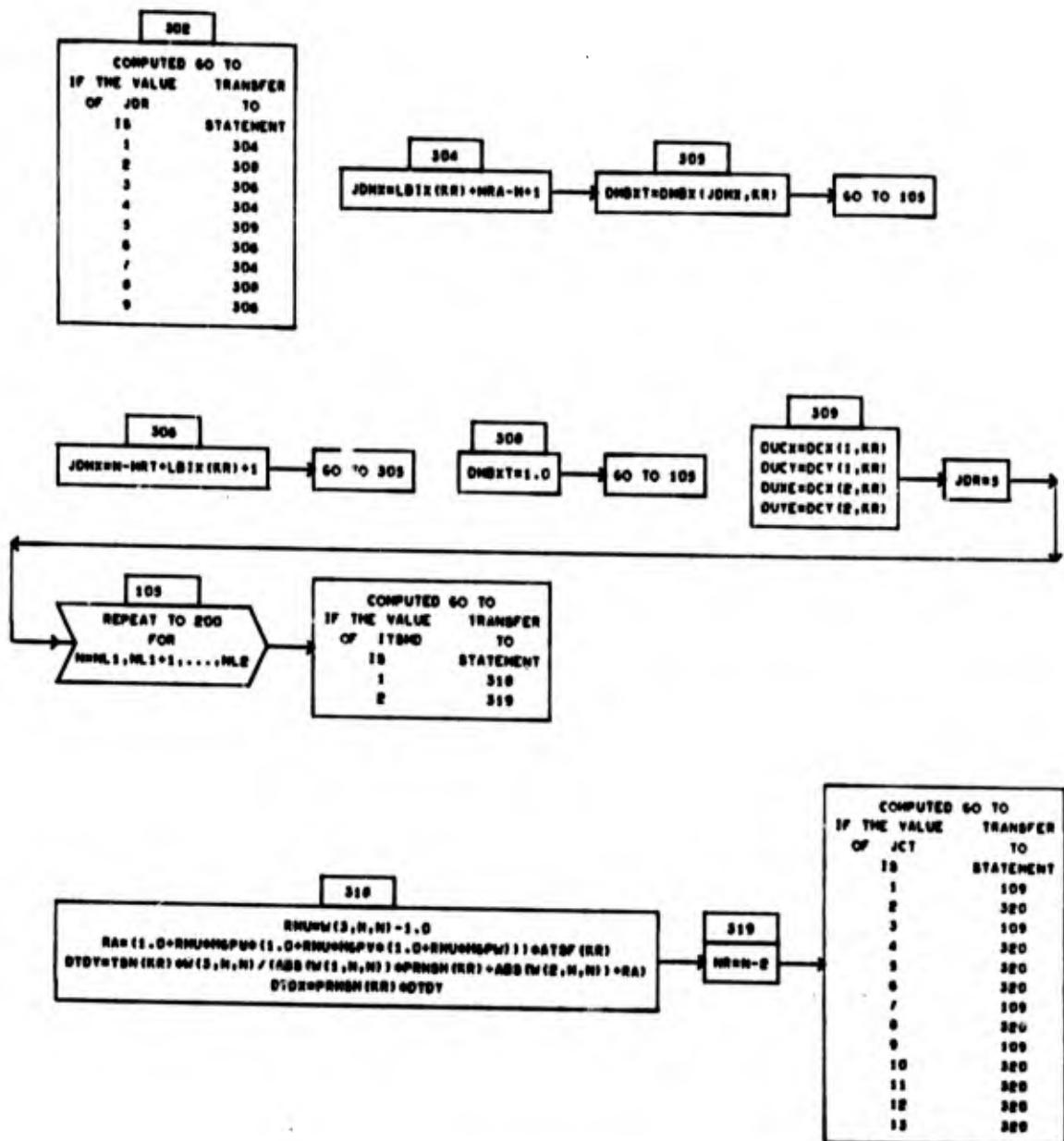
Detailed Flow Charts of Subroutine 00KII (3 of 3)



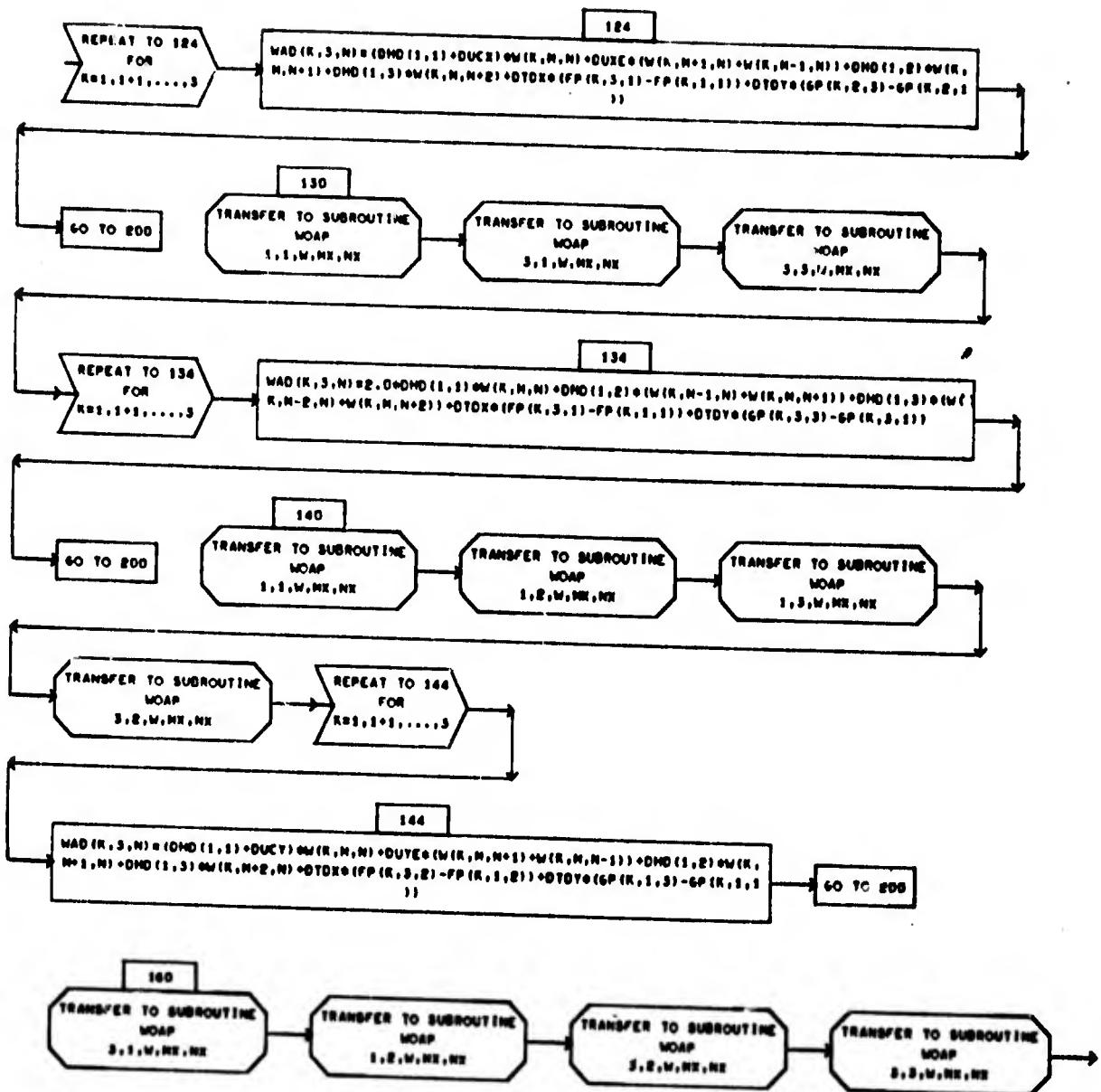
Detailed Flow Chart of Subroutine SUSUMU (1 of 9)

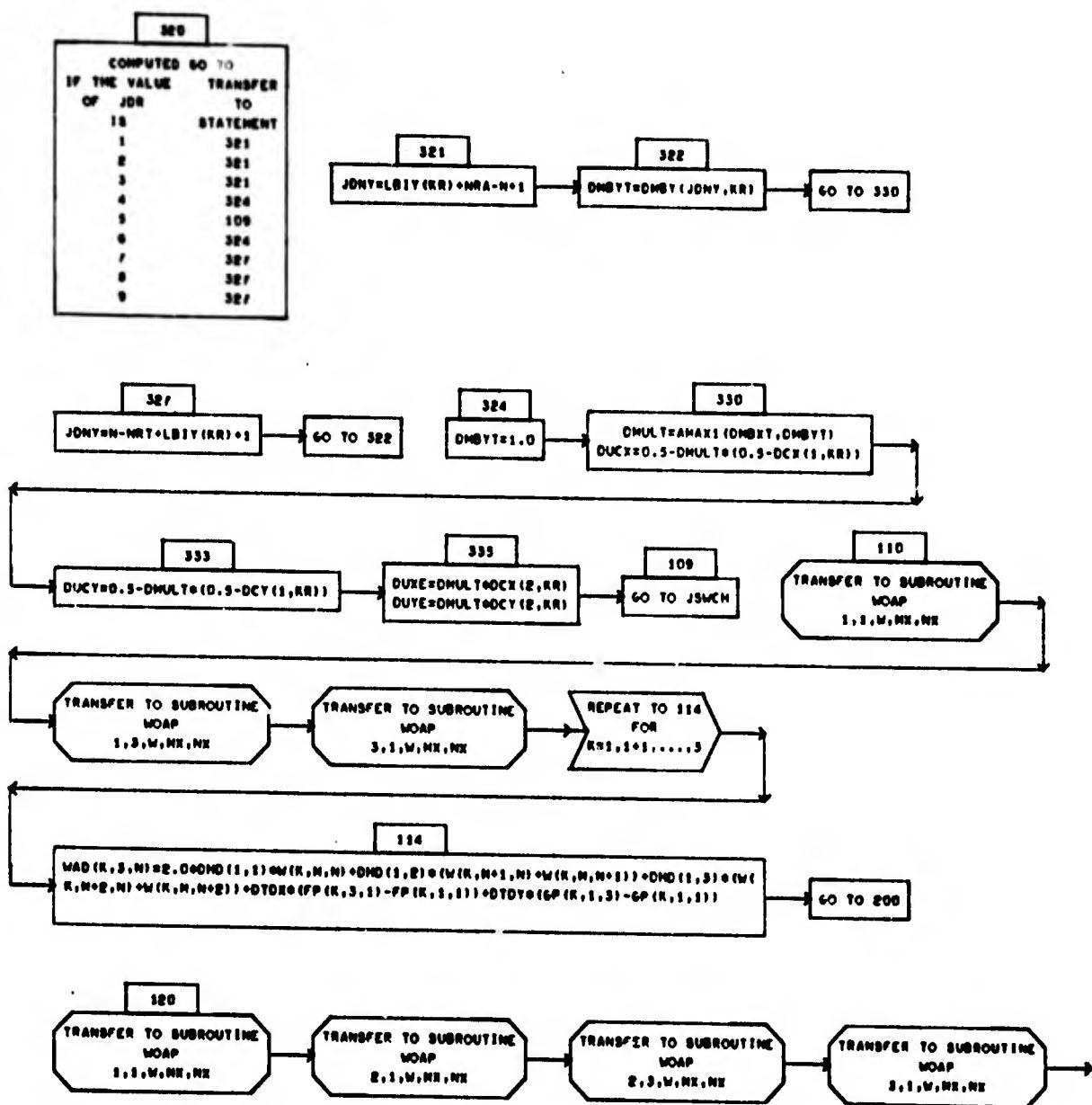


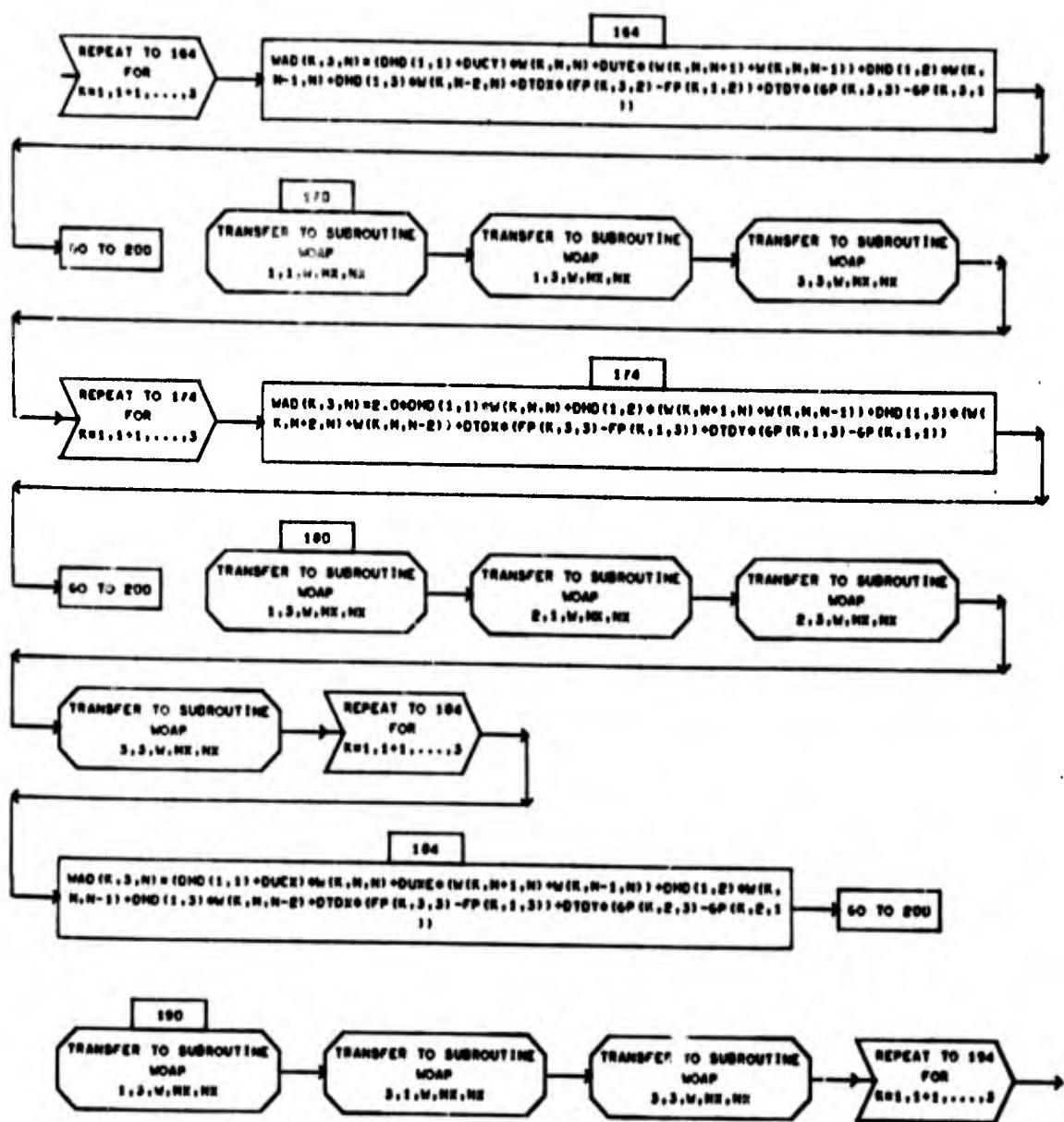
Detailed Flow Chart of Subroutine SUSUMU (2 of 9)



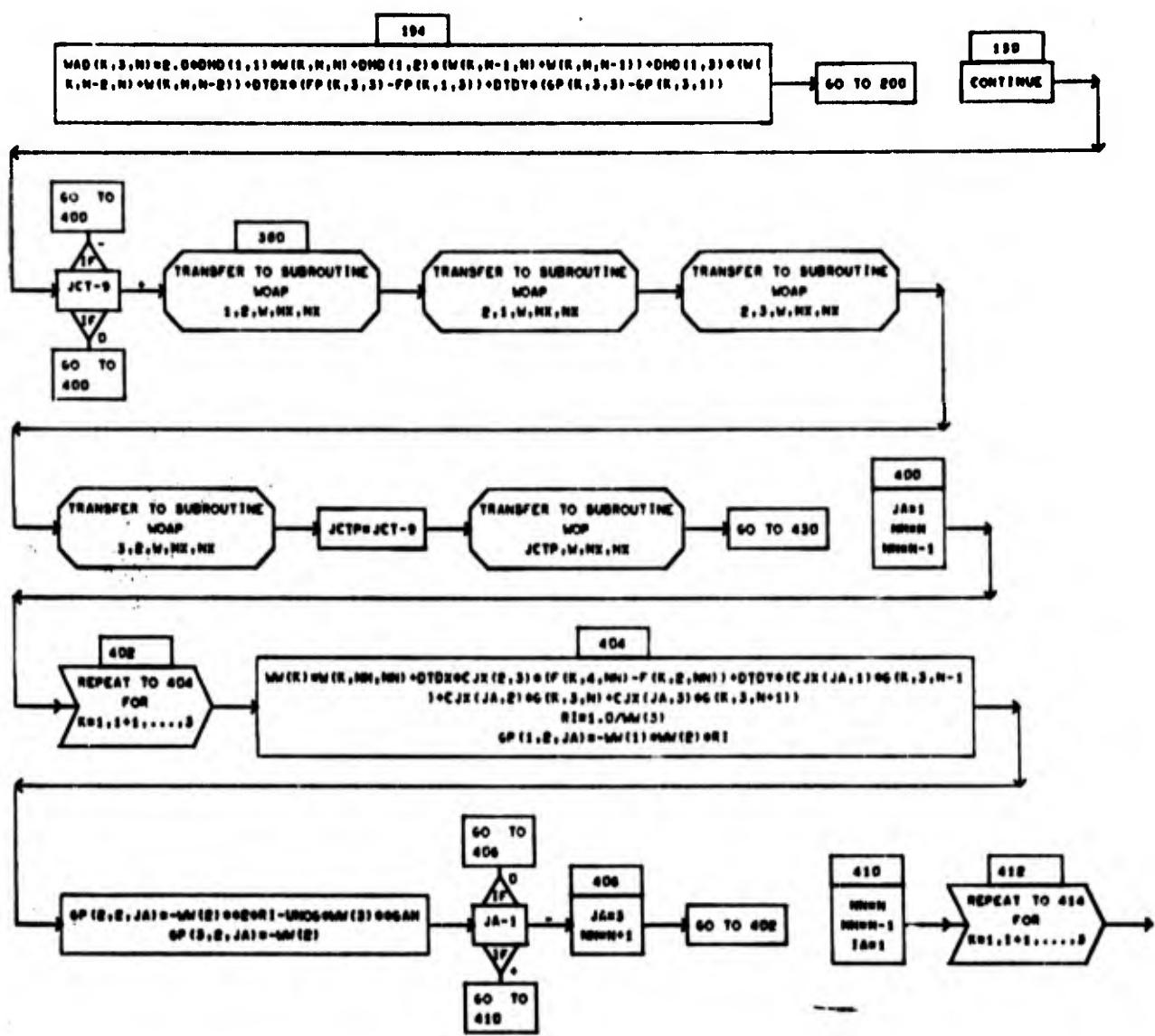
Detailed Flow Chart of Subroutine SUSUMU (3 of 9)



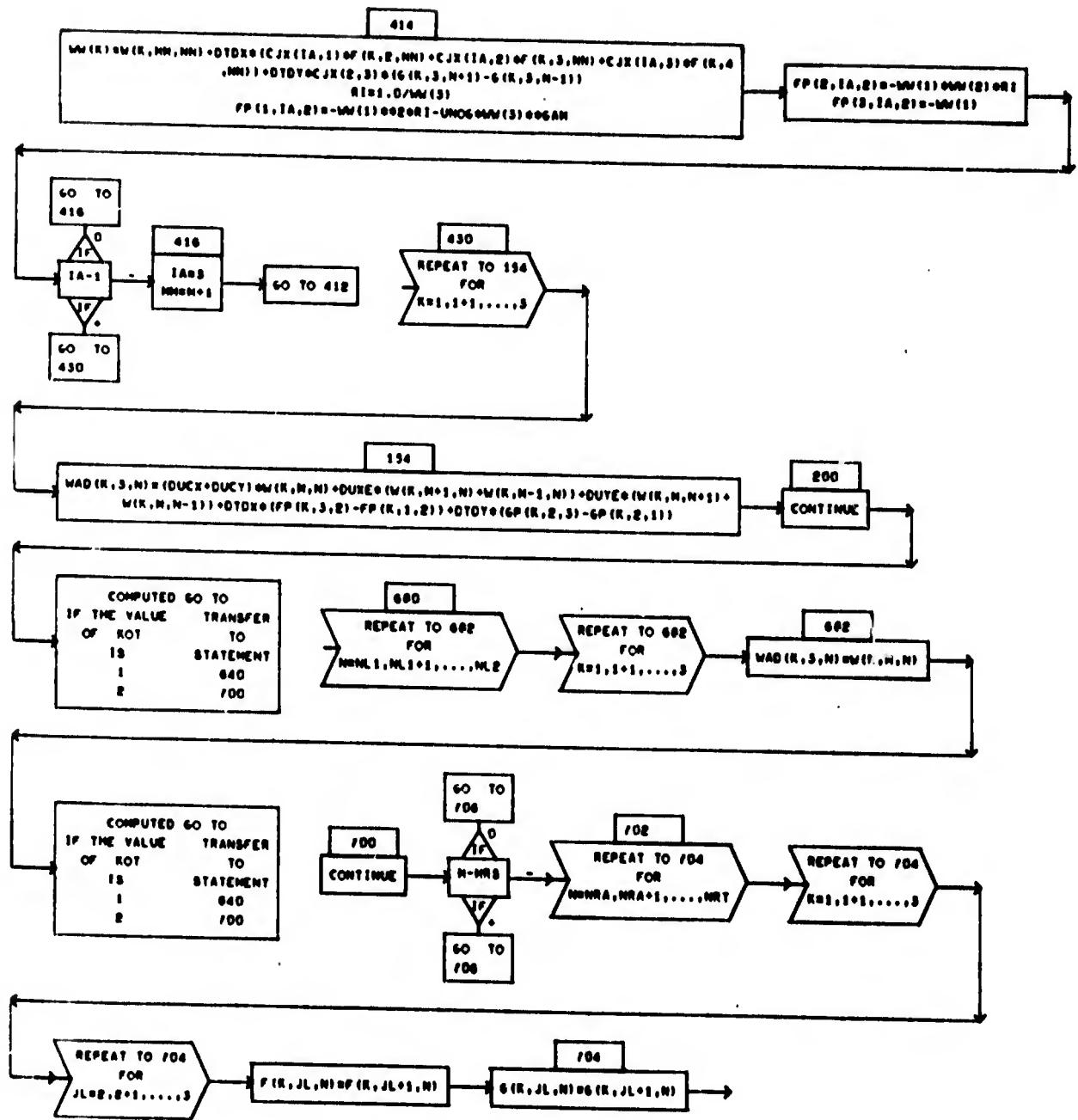




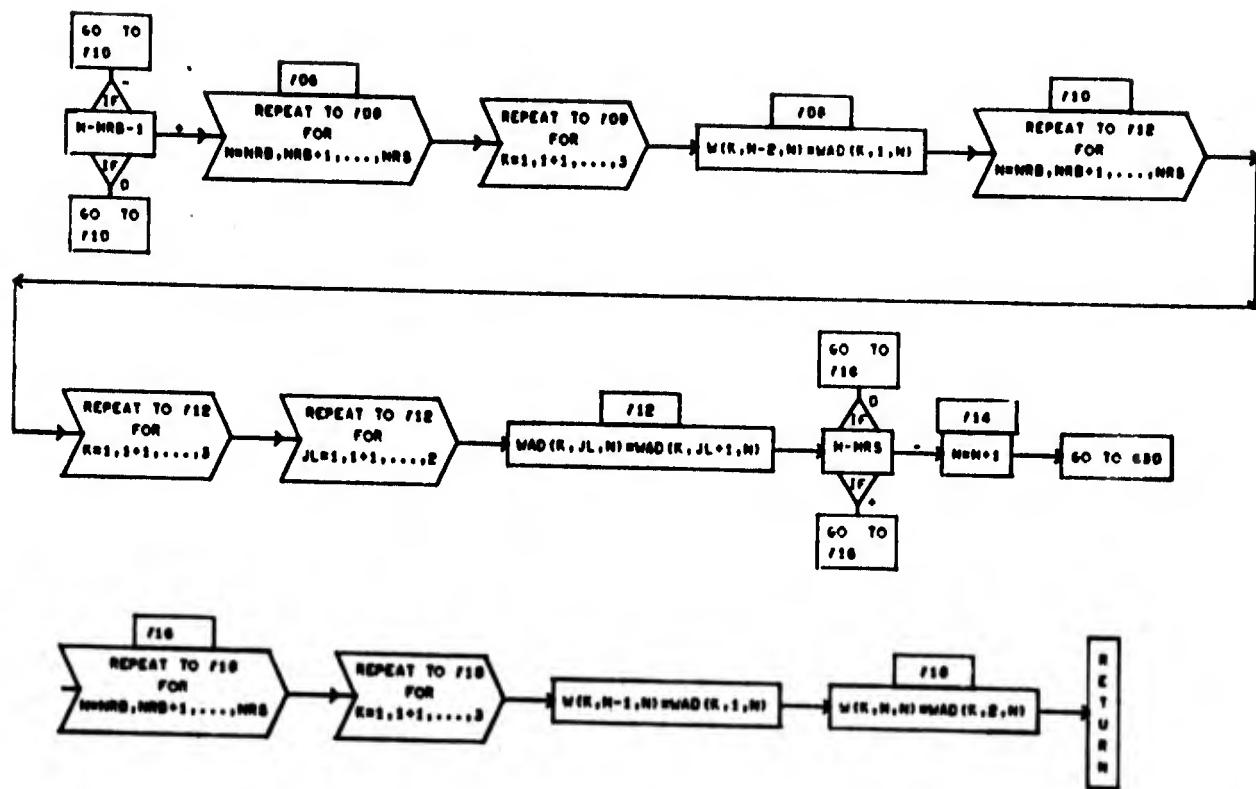
Detailed Flow Chart of Subroutine SUSUMU (6 of 9)



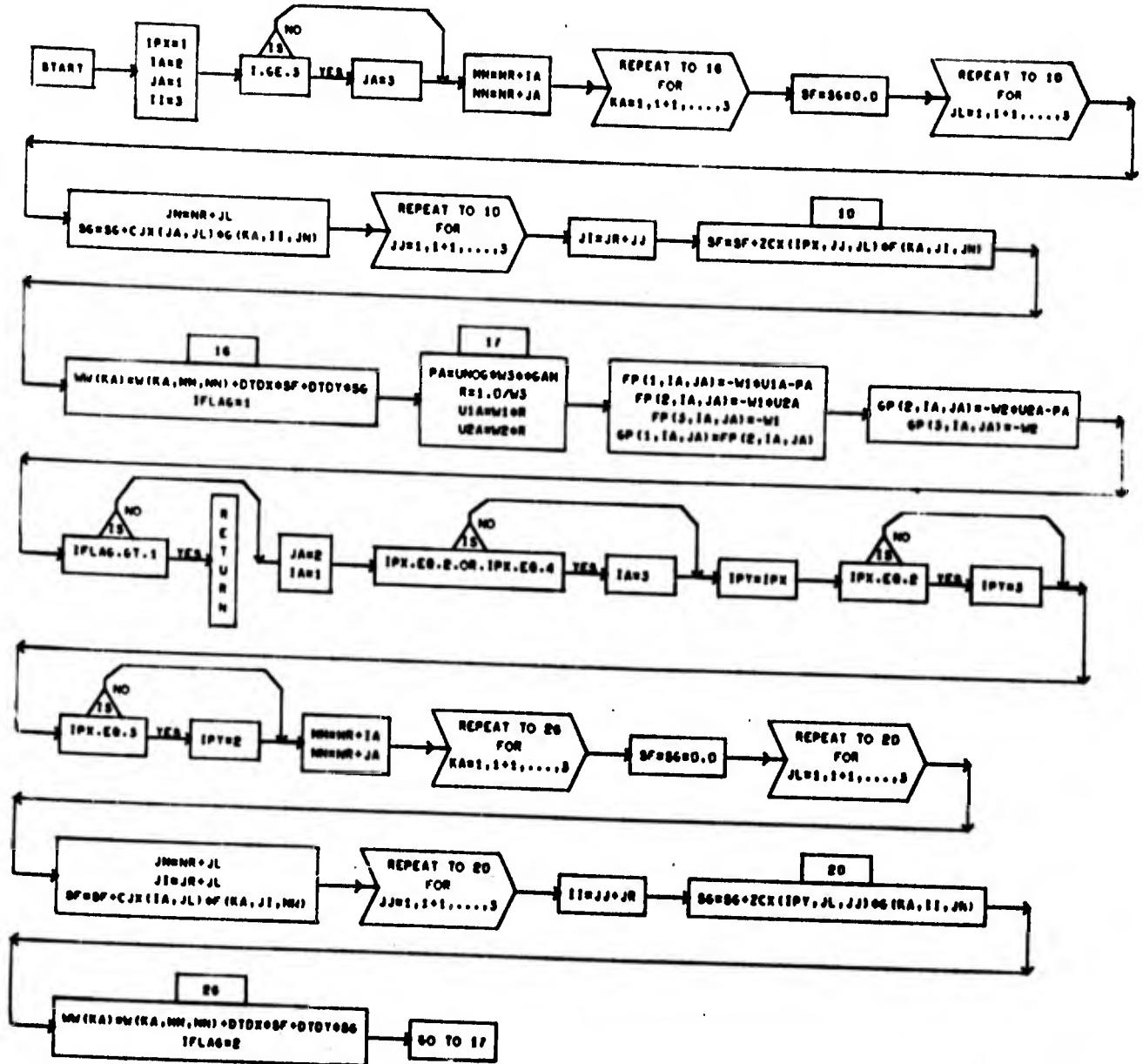
Detailed Flow Chart of Subroutine SUSUMU (7 of 9)



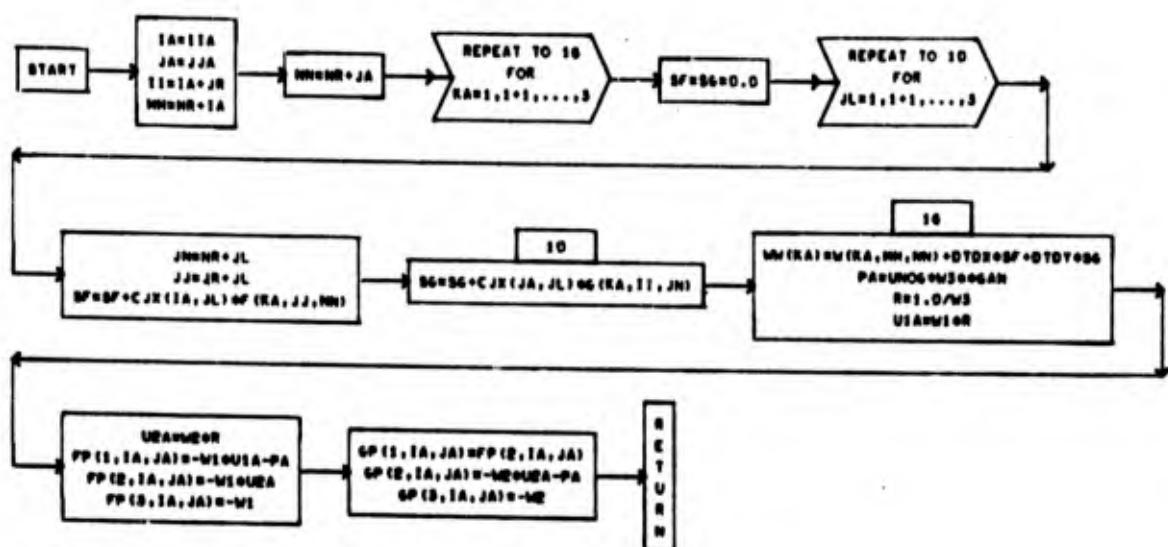
Detailed Flow Chart of Subroutine SUSUMU (8 of 9)



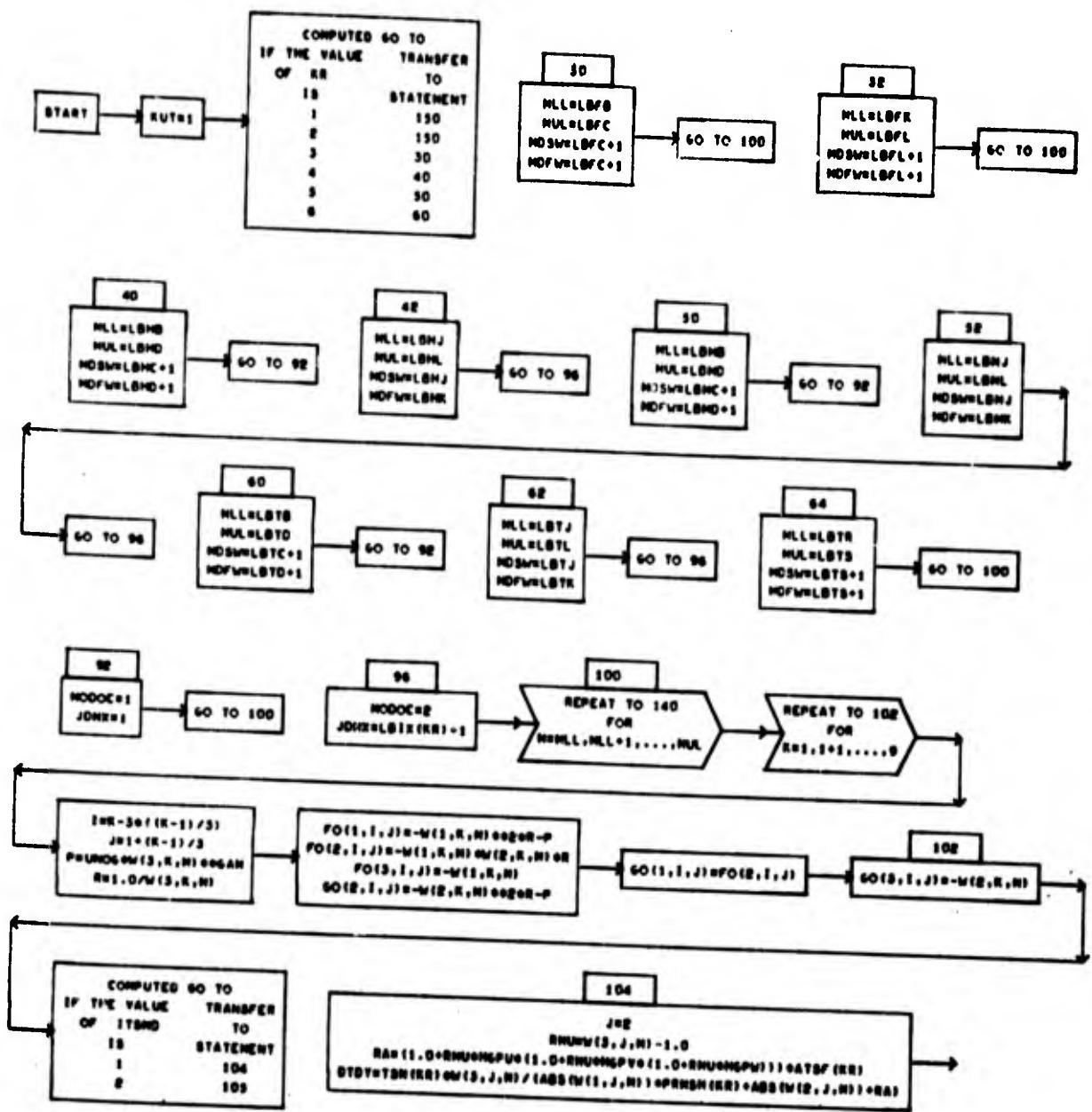
Detailed Flow Chart of Subroutine SUSUMU (9 of 9)



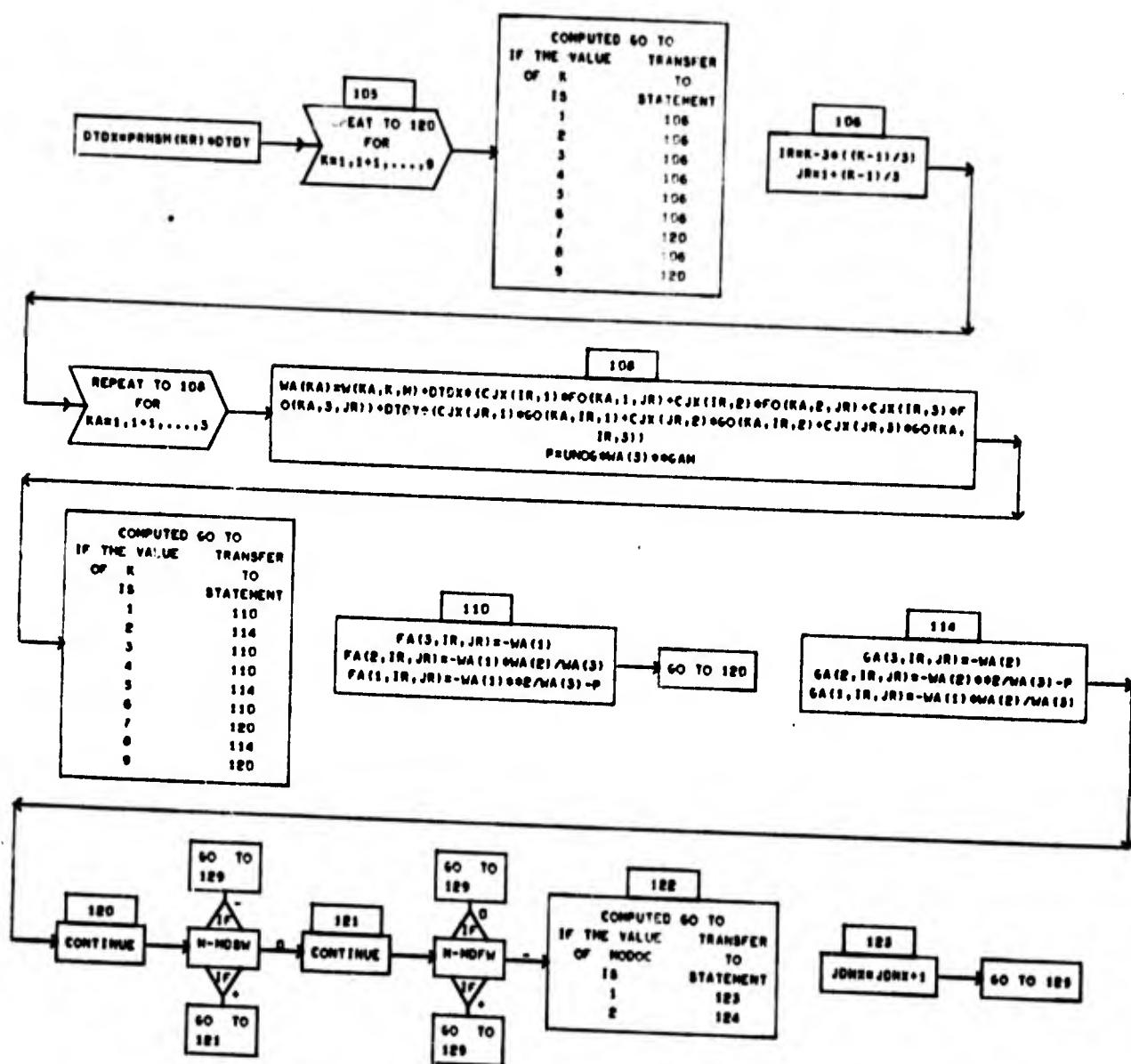
Detailed Flow Chart of Subroutine WOP



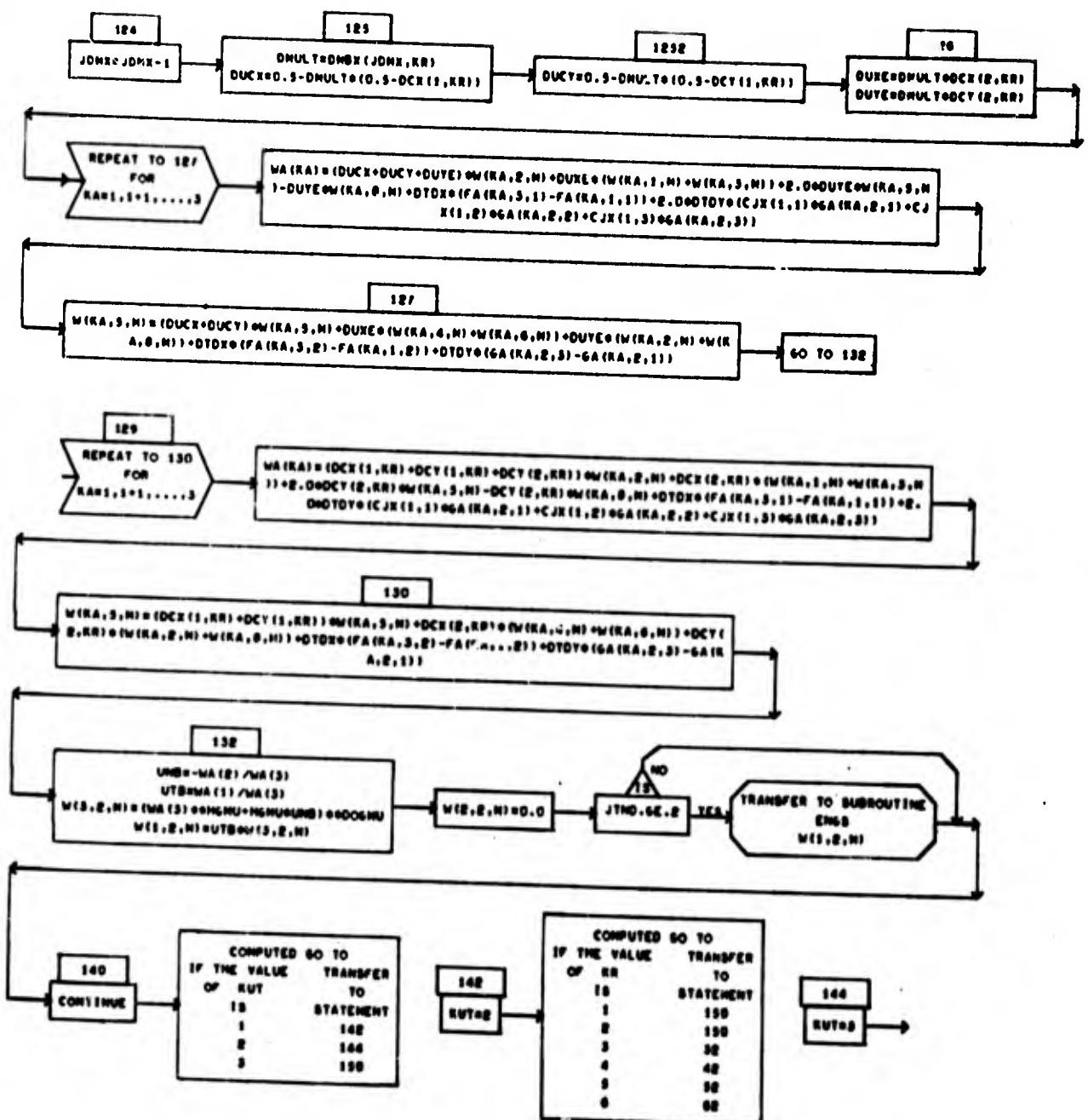
Detailed Flow Chart of Subroutine WOAP



Detailed Flow Charts of Subroutine HERI (1 of 4)



Detailed Flow Charts of Subroutine HERI (2 of 4)



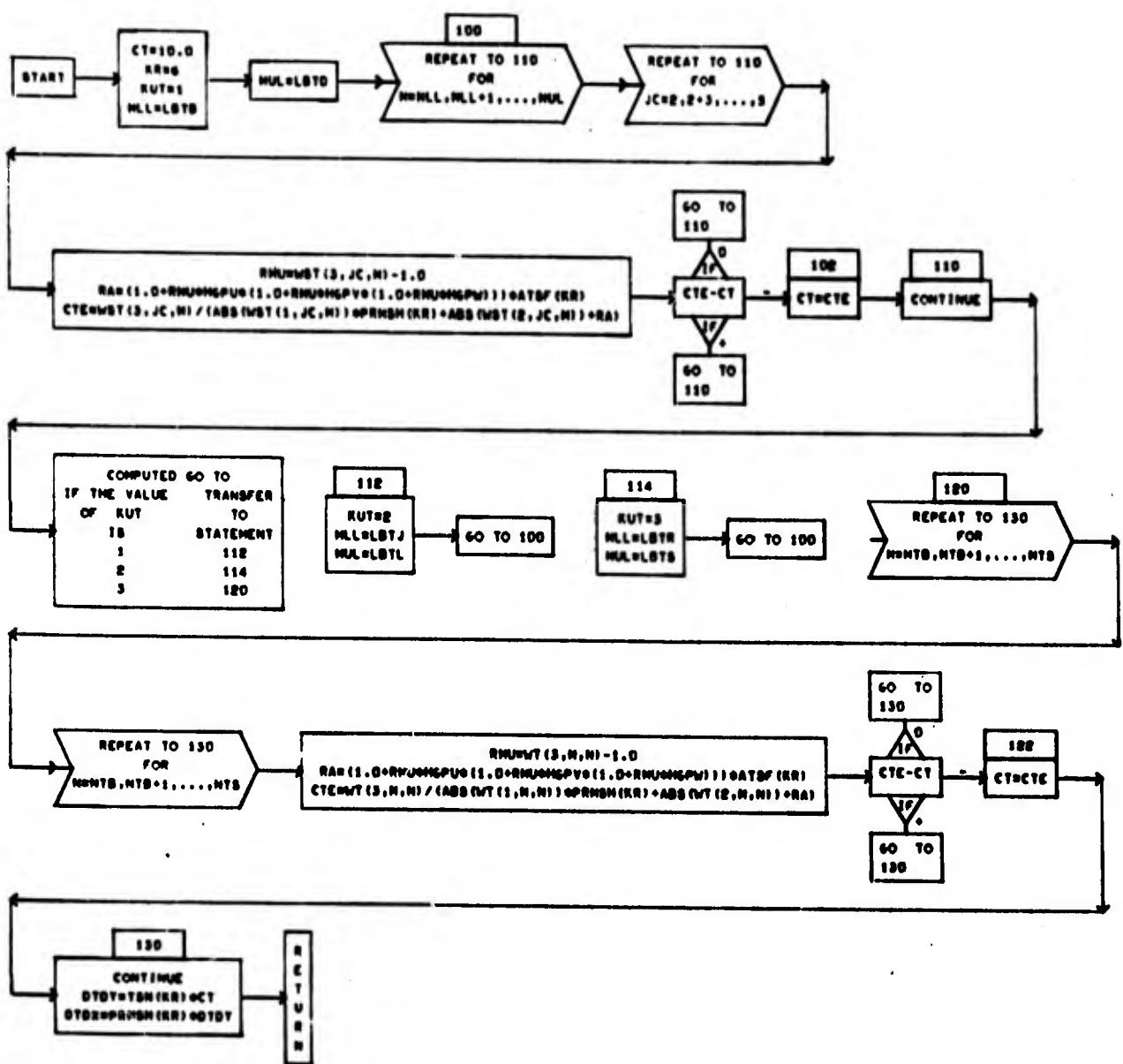
Detailed Flow Charts of Subroutine HERI (3 of 4)

COMPUTED SO TO
IF THE VALUE TRANSFER
OF IR TO
IS STATEMENT

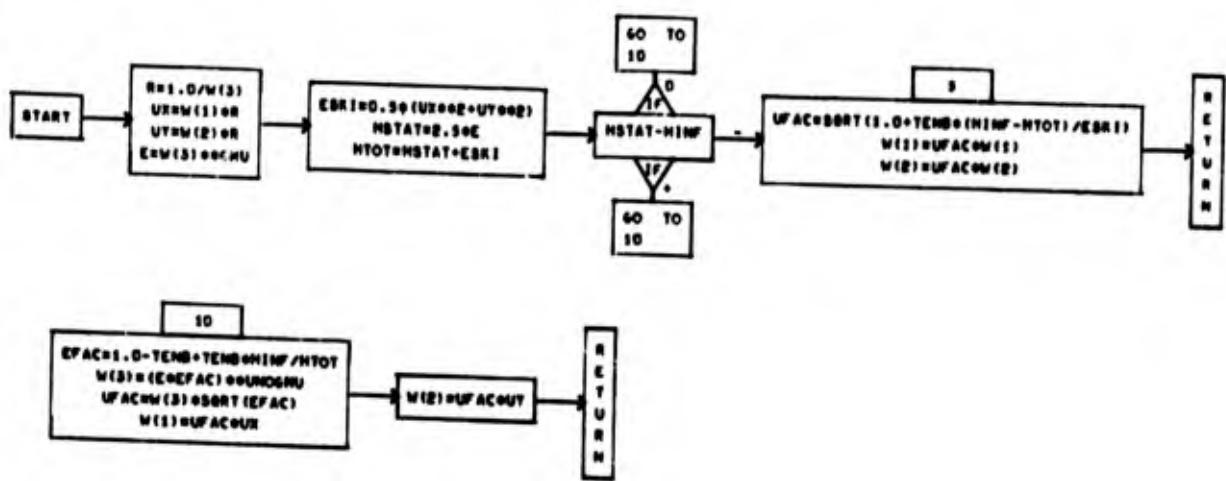
1	150
2	150
3	150
4	150
5	150
6	64

150
R
E
T
U
R
N

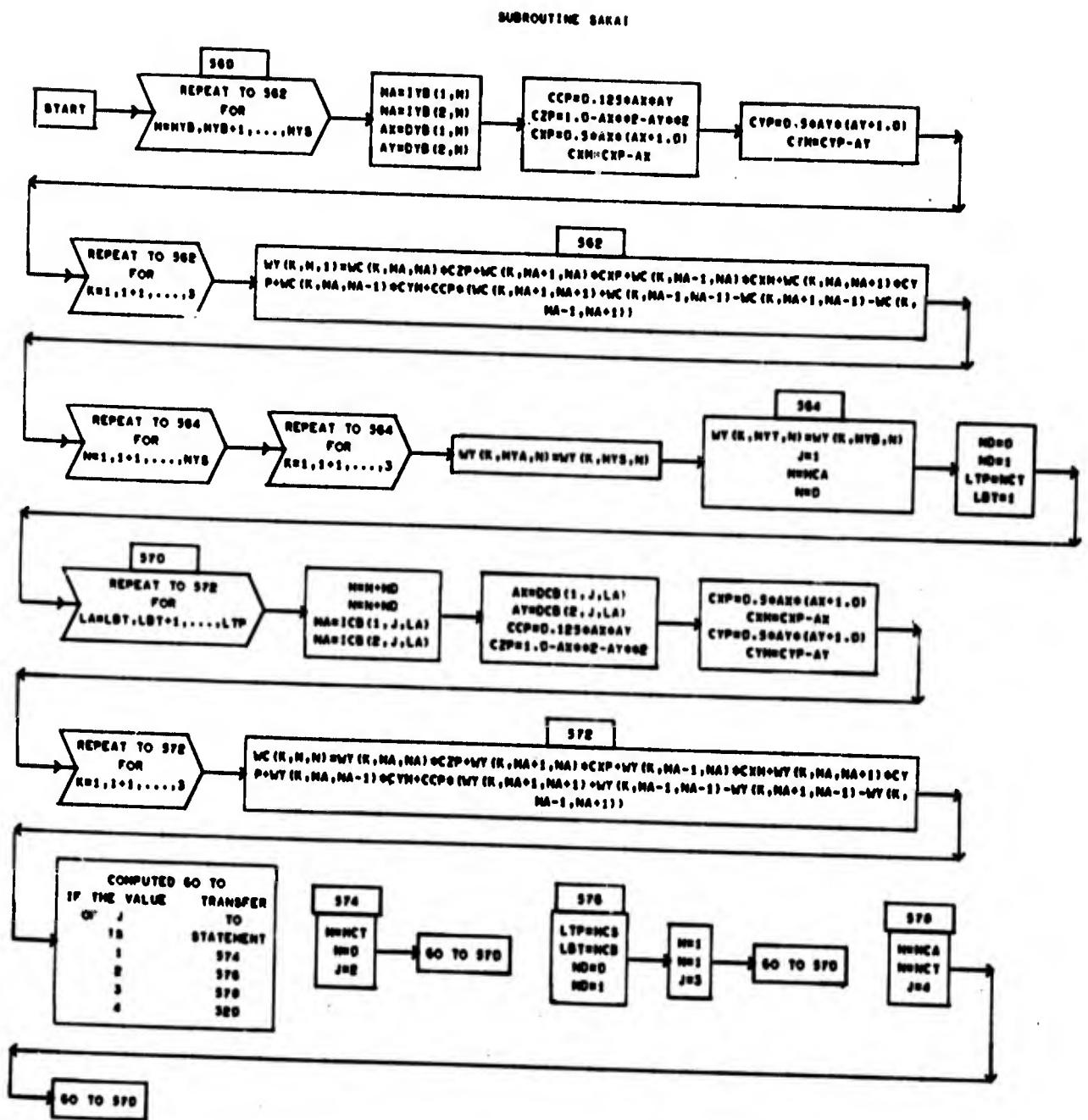
Detailed Flow Charts of Subroutine HERI (4 of 4)



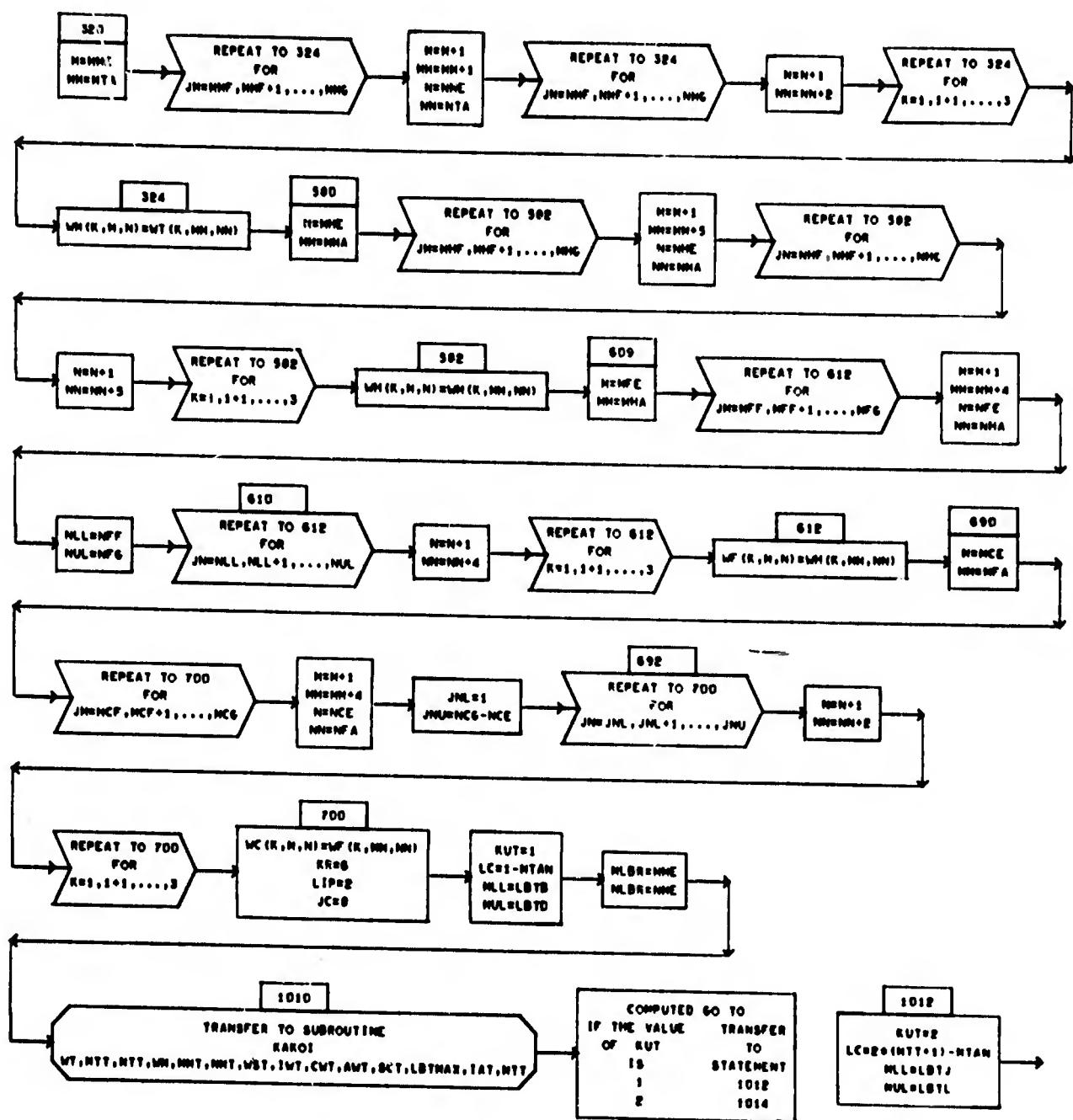
Detailed Flow Chart of Subroutine JIKAN



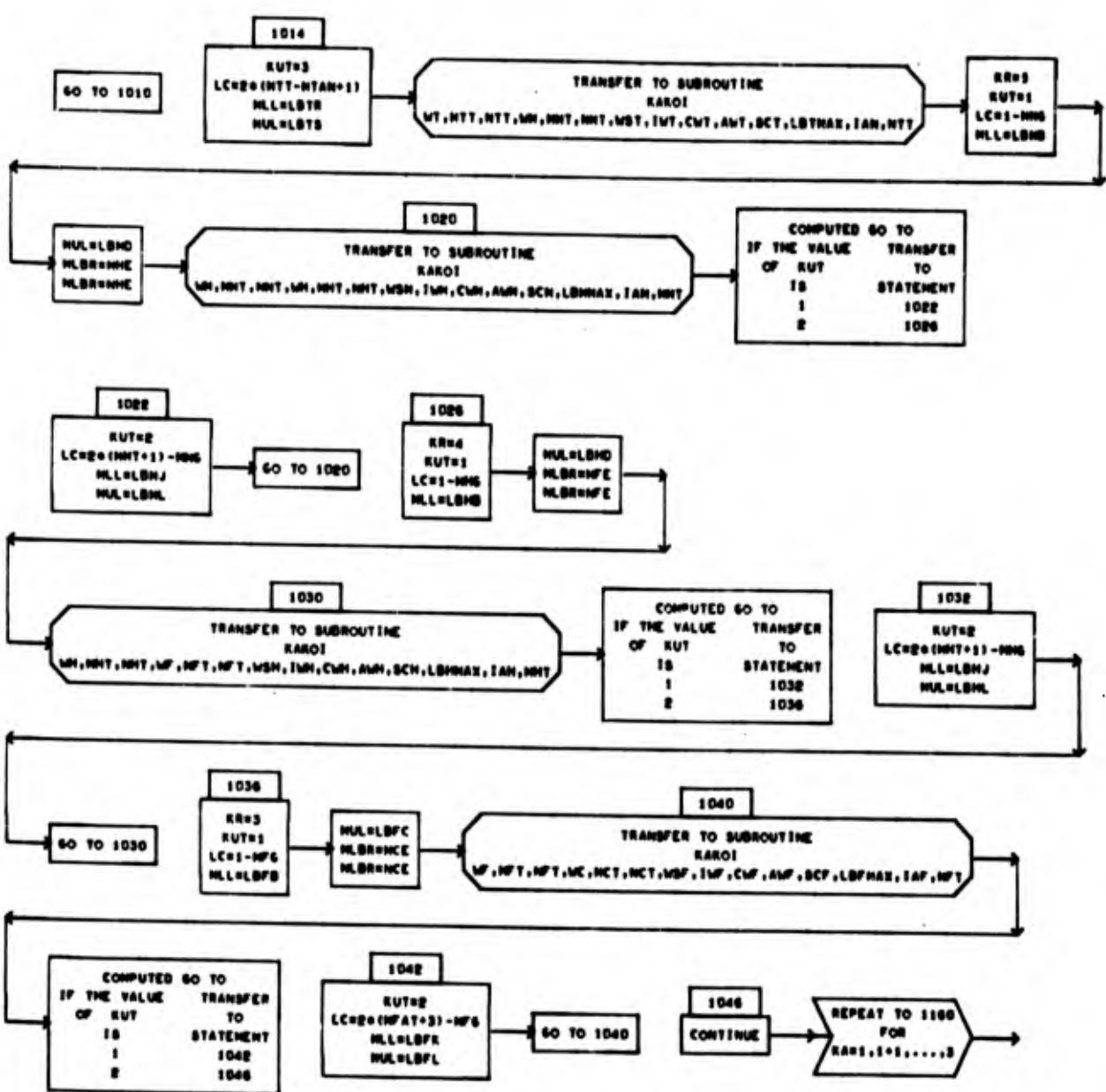
Detailed Flow Chart of Subroutine ENGS



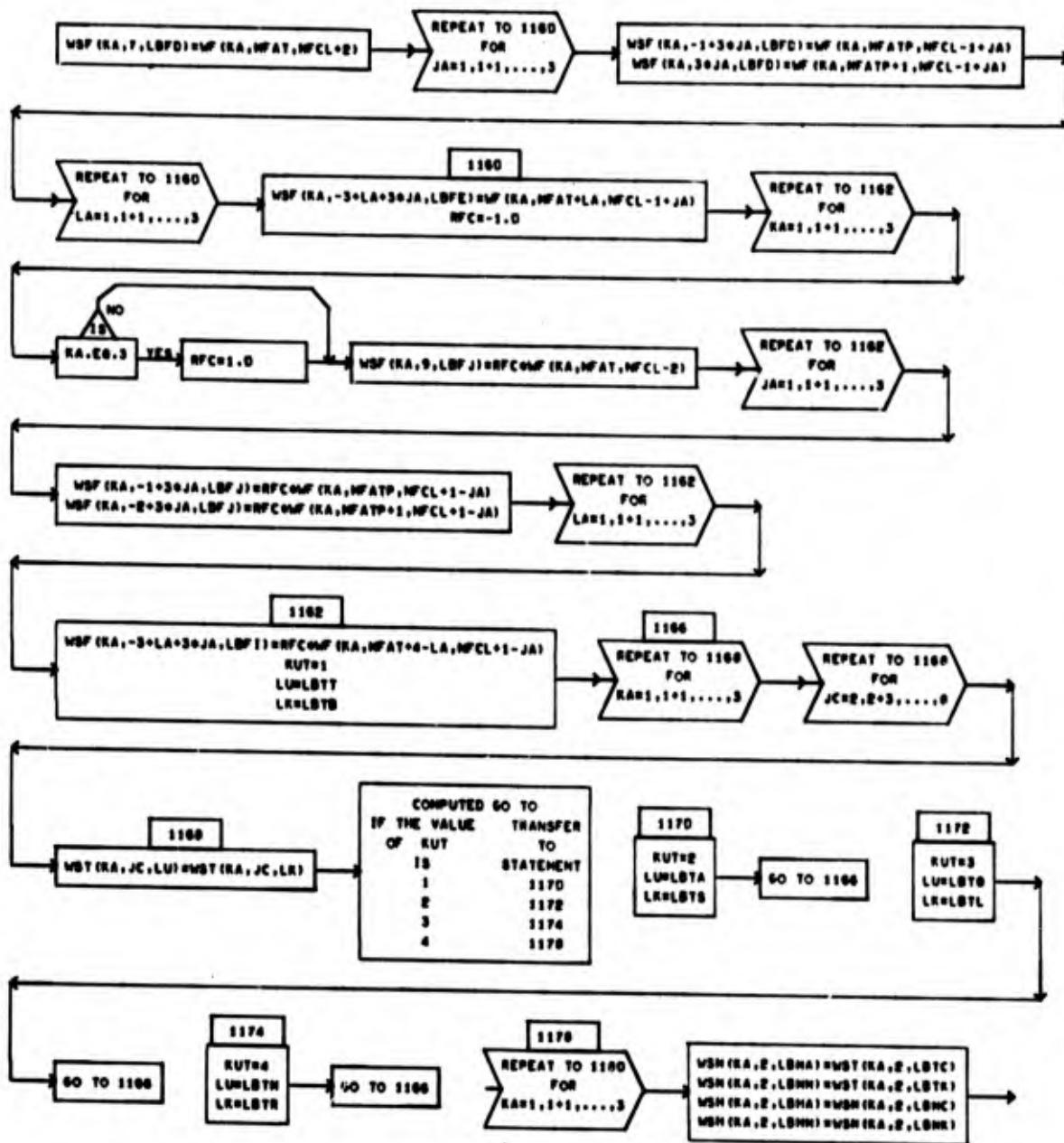
Detailed Flow Charts of Subroutine SAKAI (1 of 13)



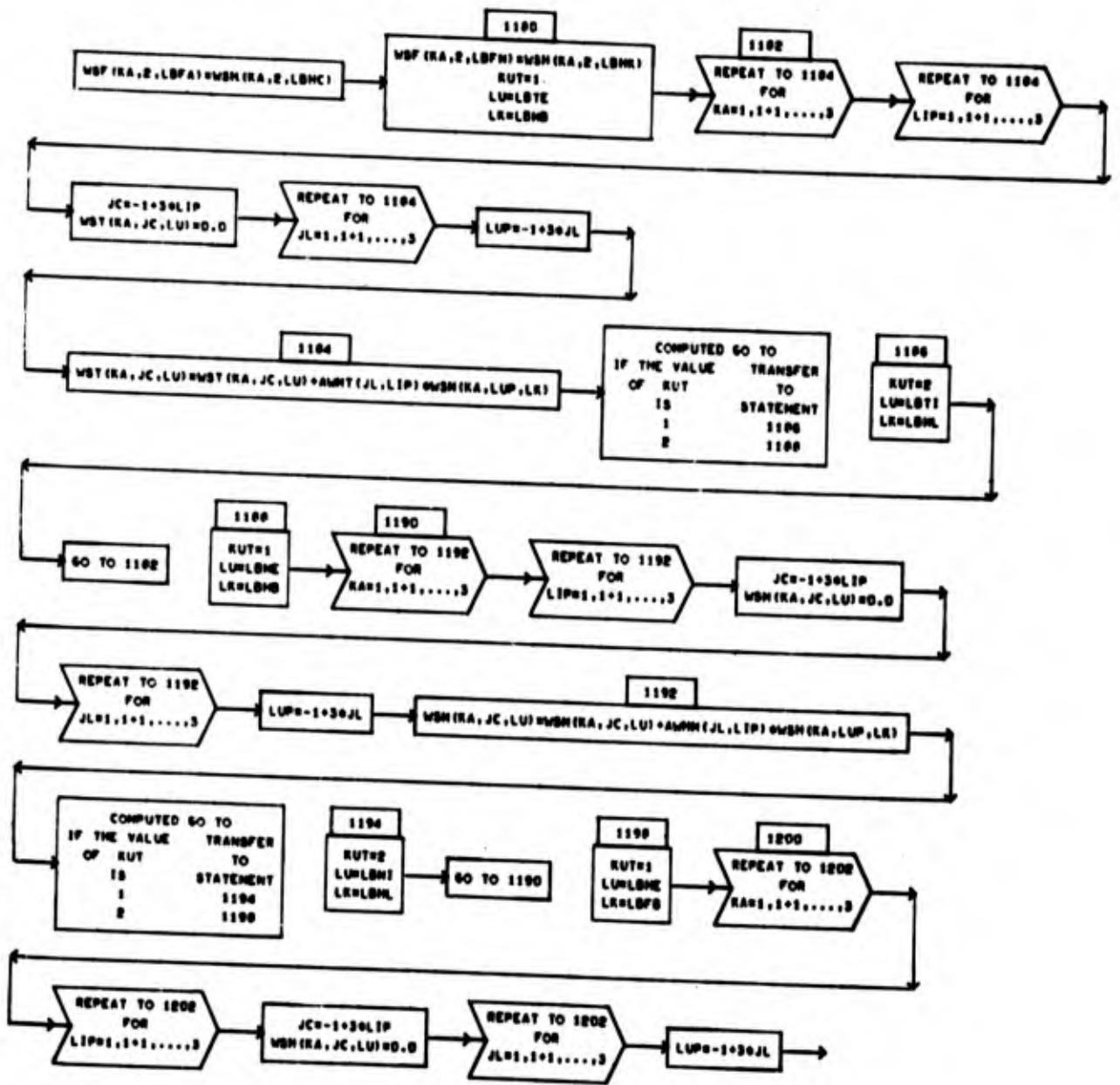
Detailed Flow Charts of Subroutine SAKAI (2 of 13)



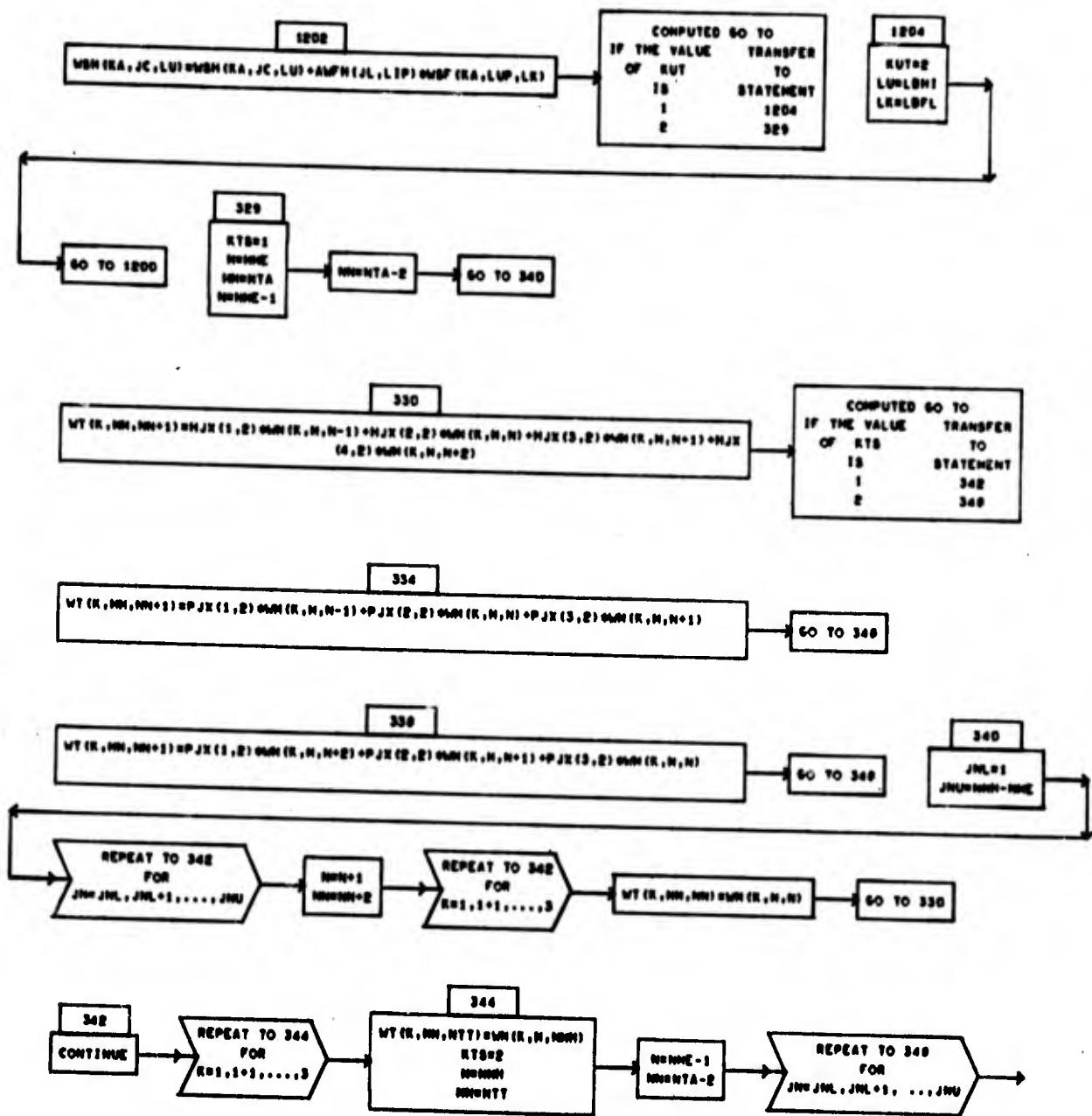
Detailed Flow Charts of Subroutine SAKAI (3 of 13)



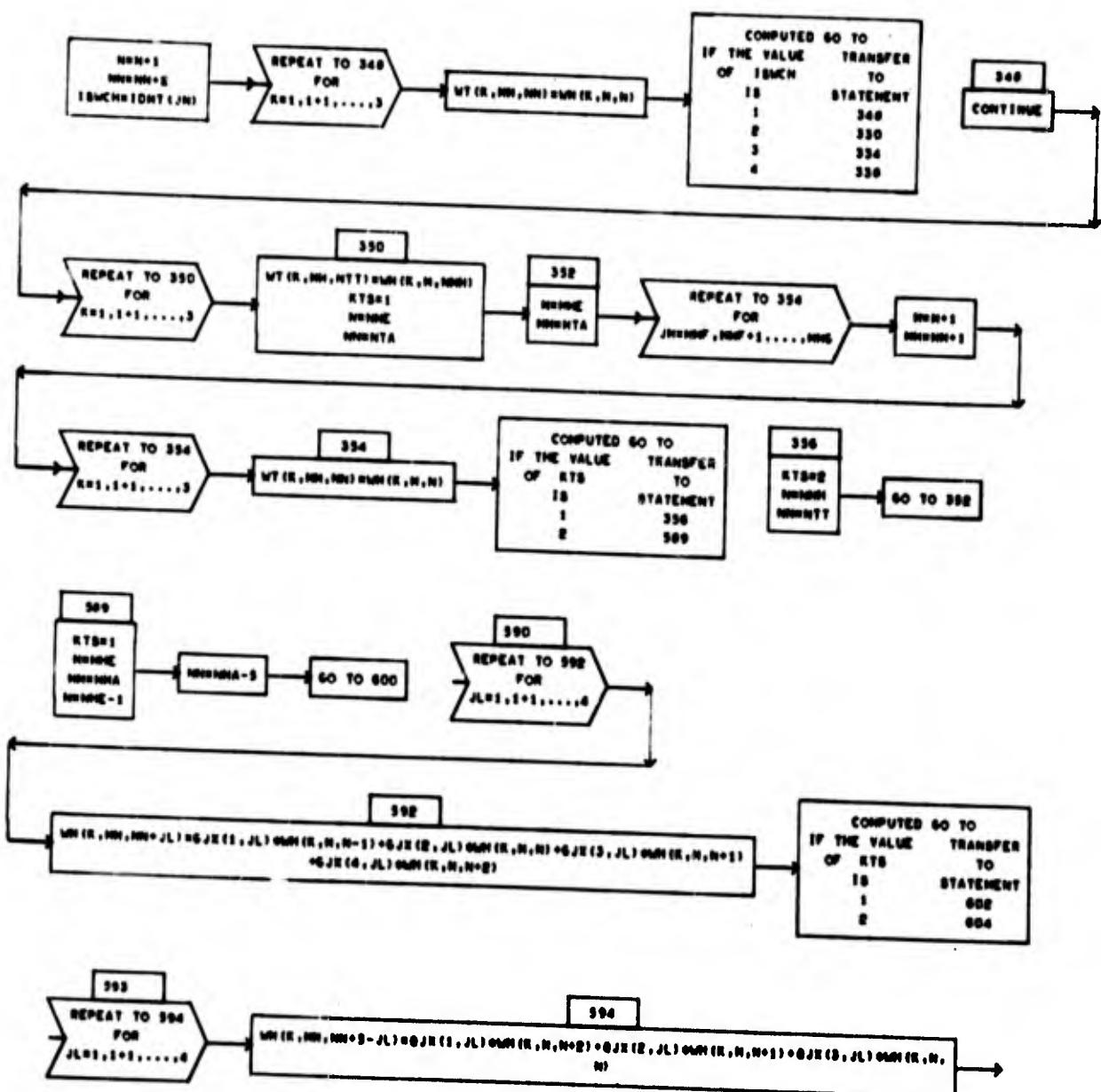
Detailed Flow Charts of Subroutine SAKAI (4 of 13)



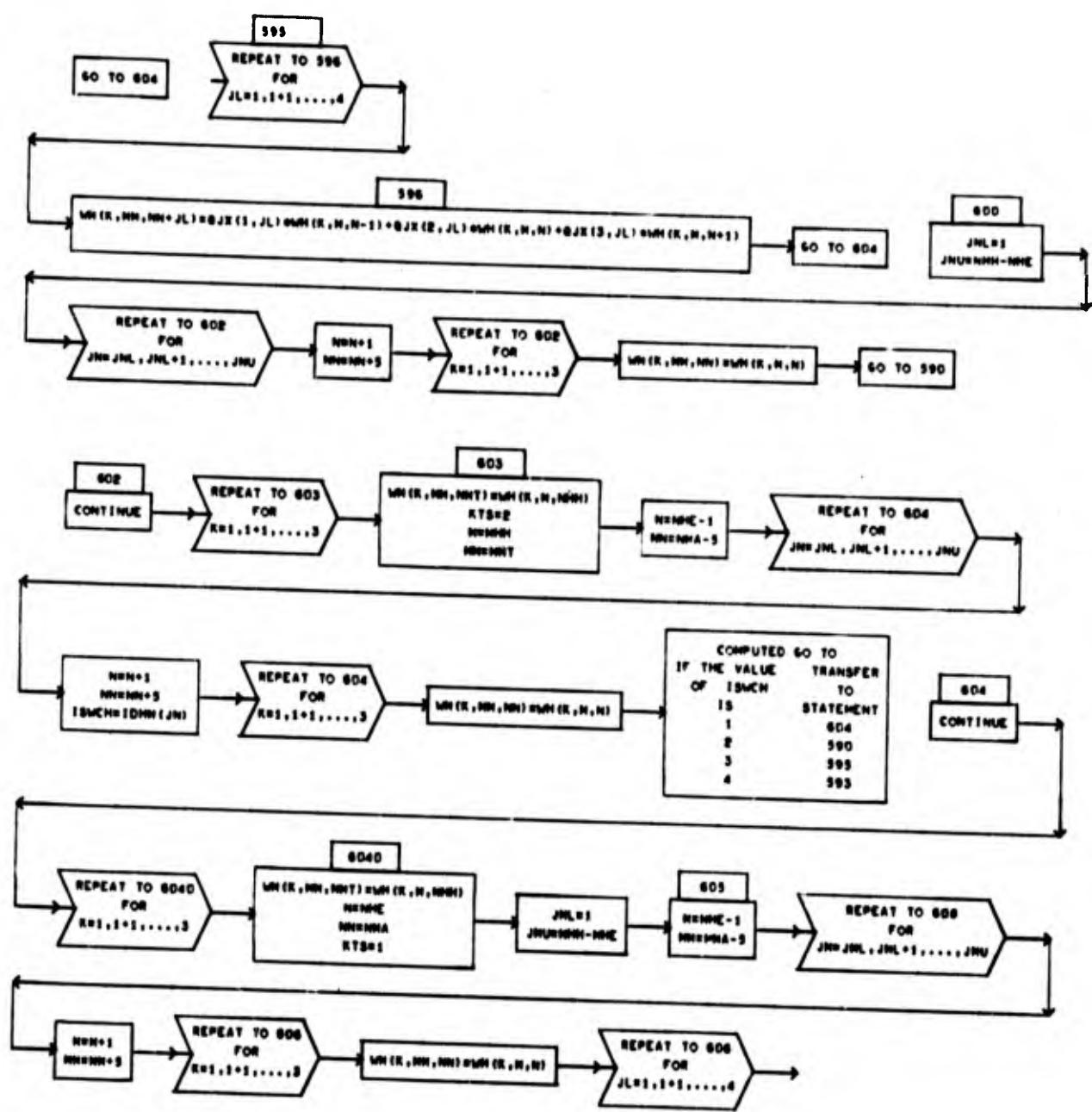
Detailed Flow Charts of Subroutine SAKAI (5 of 13)



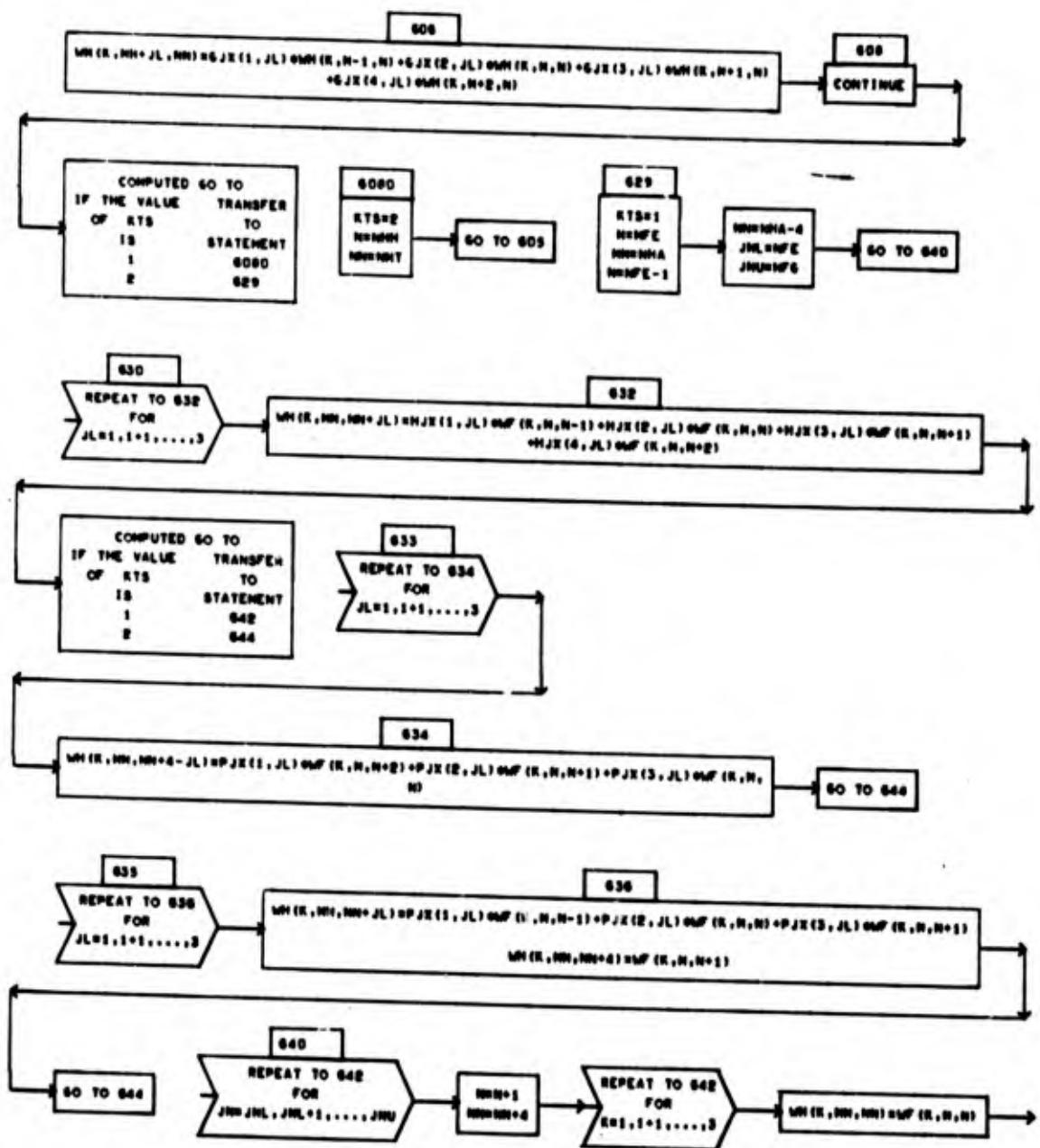
Detailed Flow Charts of Subroutine SAKAI (6 of 13)



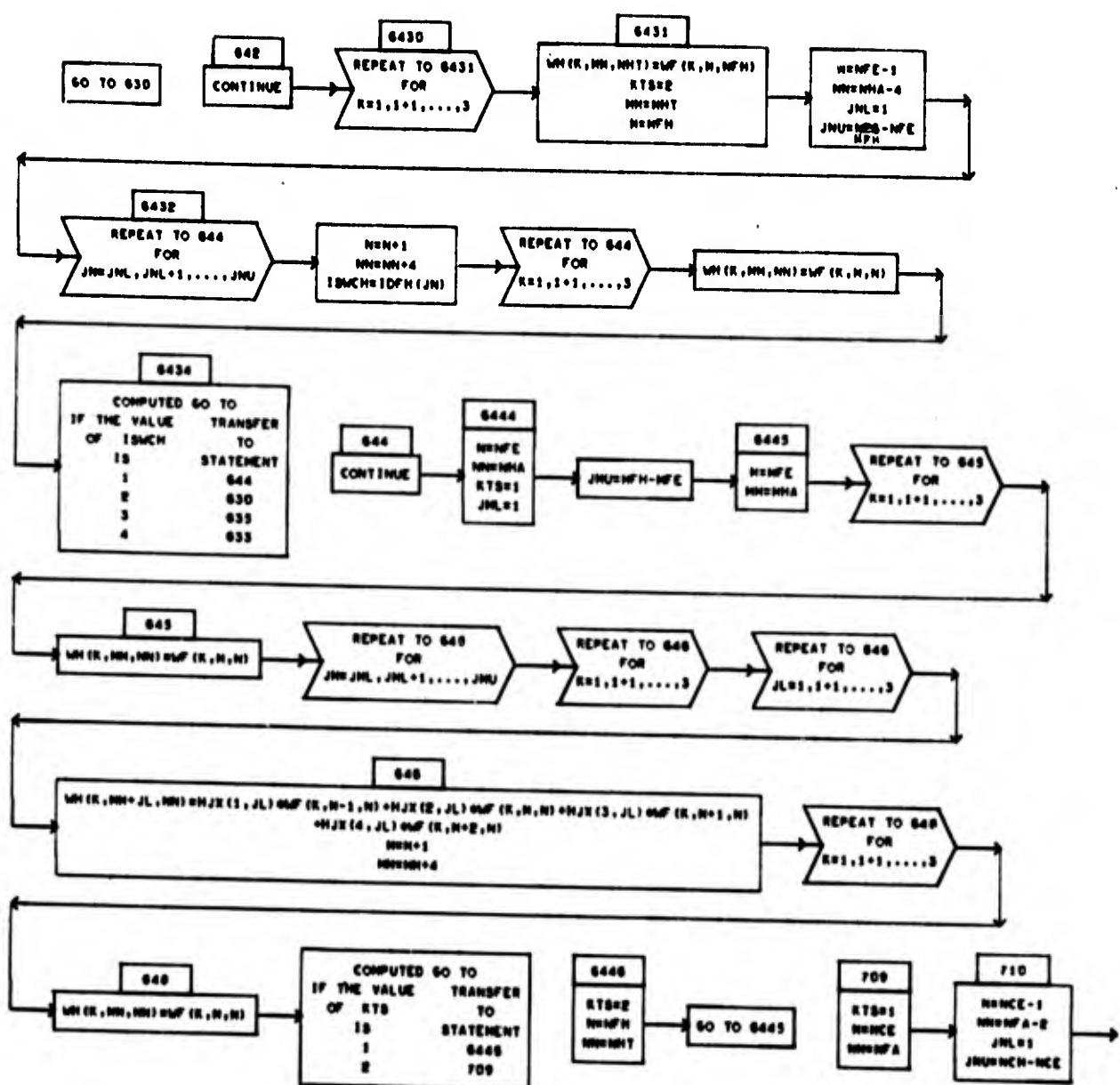
Detailed Flow Charts of Subroutine SAKAI (7 of 13)



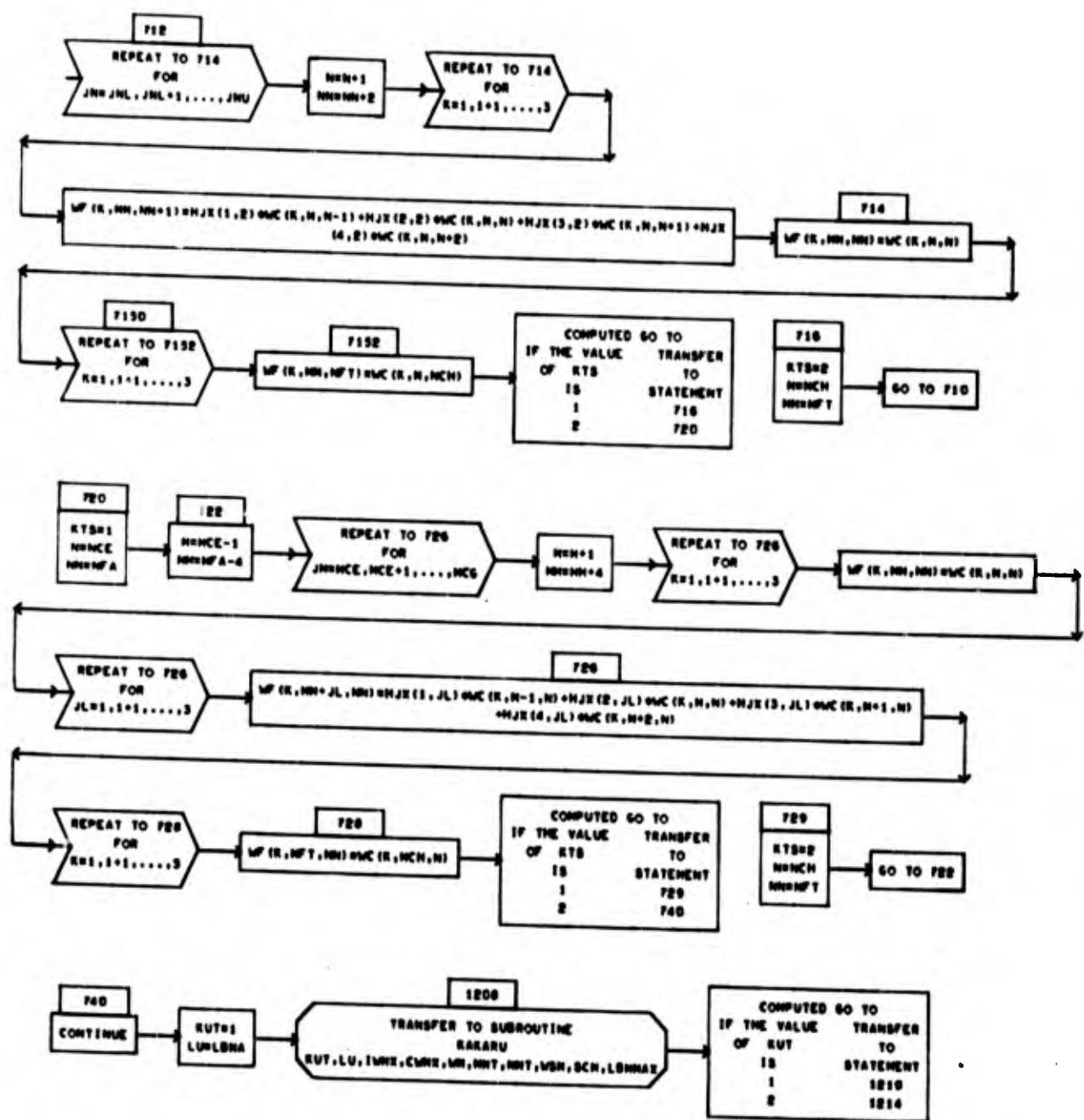
Detailed Flow Charts of Subroutine SAKAI (8 of 13)



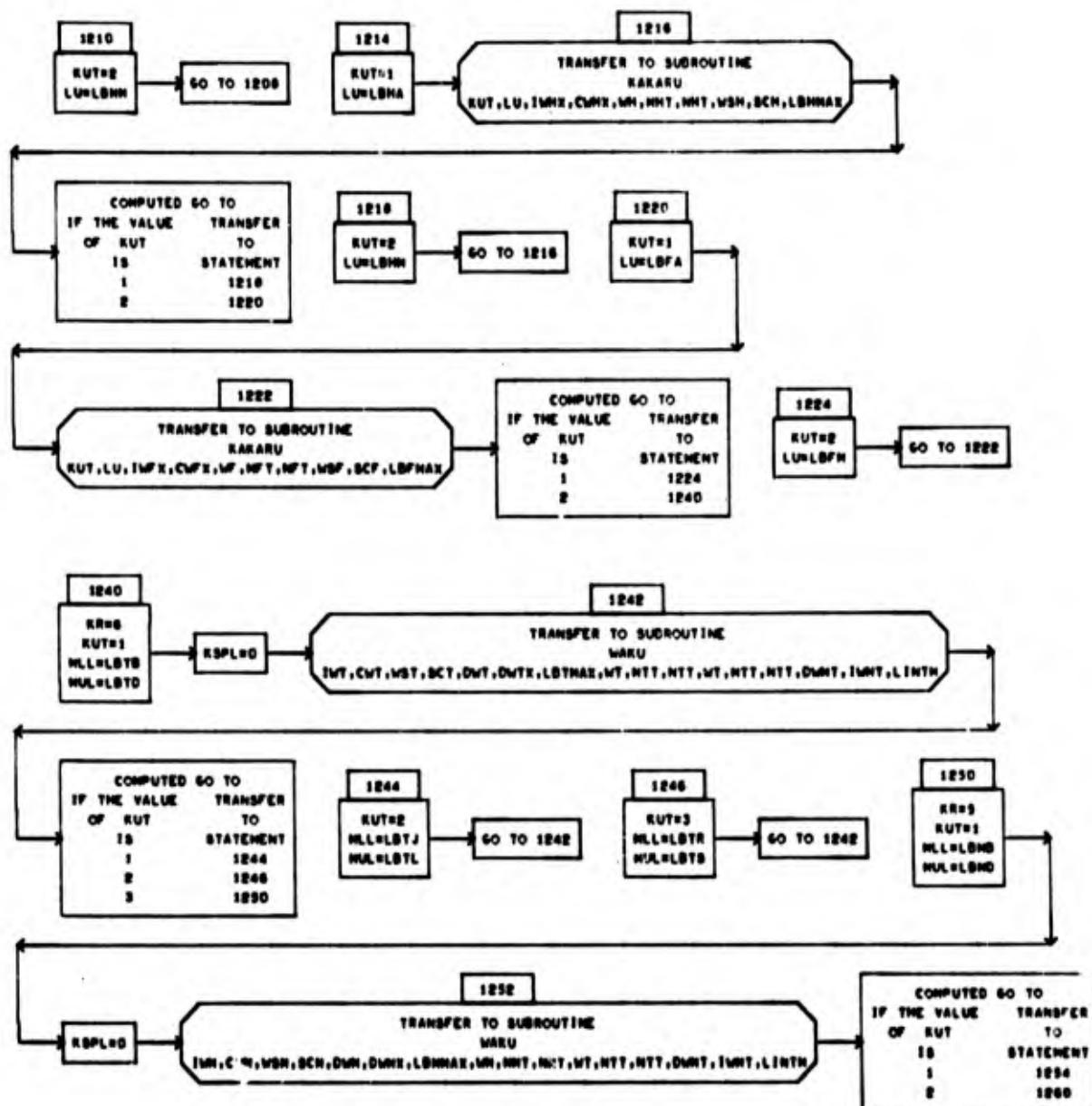
Detailed Flow Charts of Subroutine SAKAI (9 of 13)



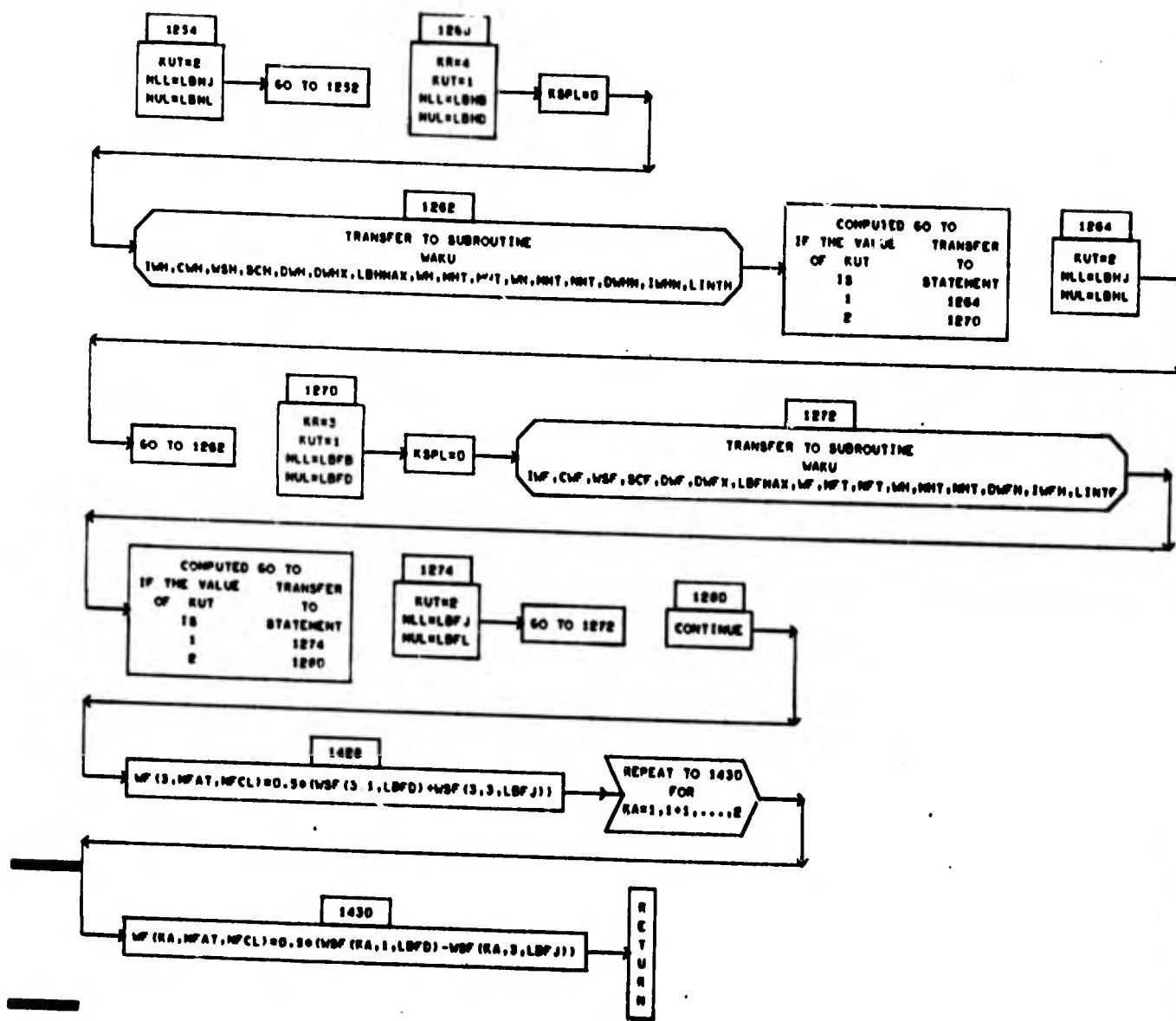
Detailed Flow Charts of Subroutine SAKAI (10 of 13)



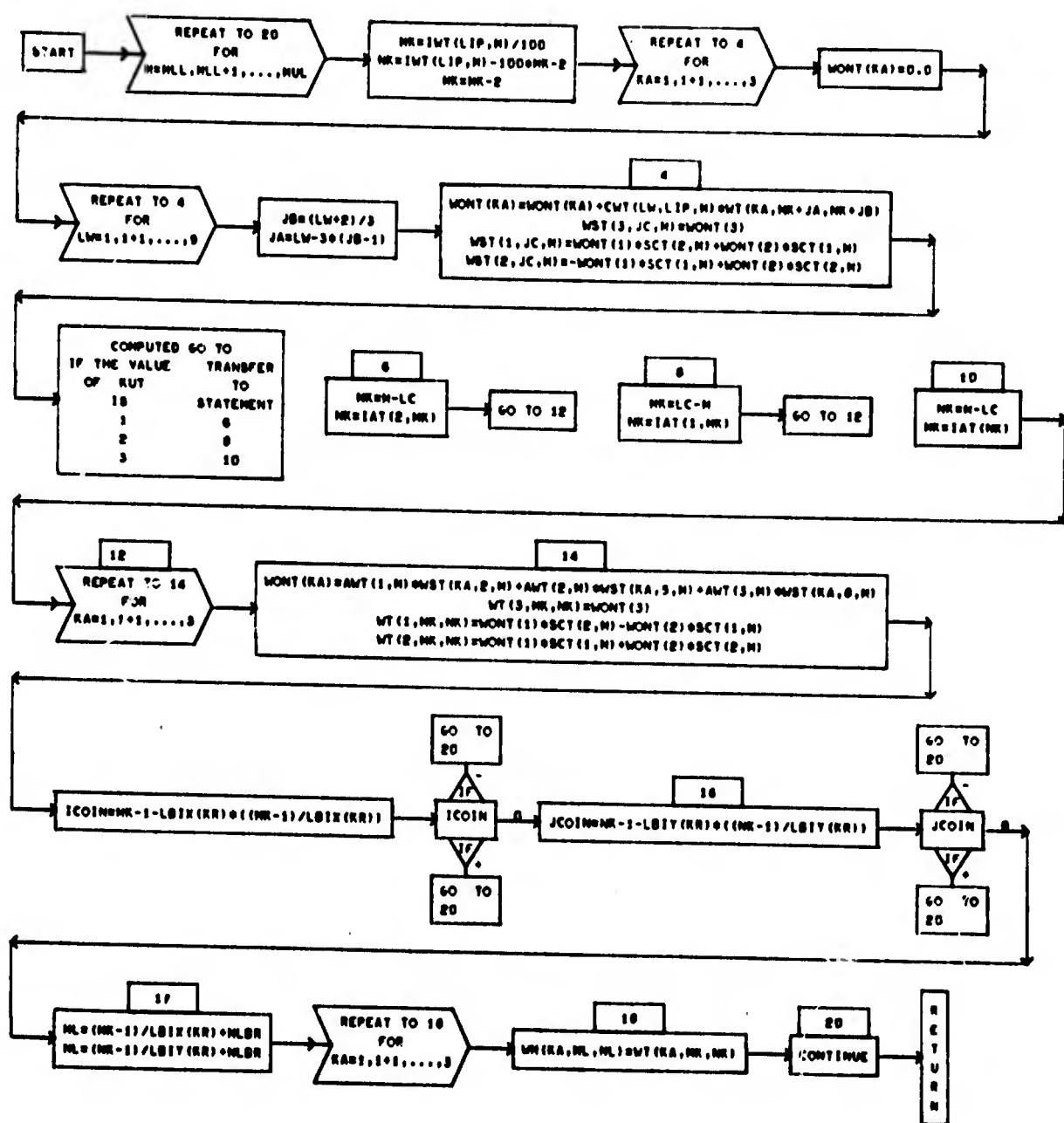
Detailed Flow Charts of Subroutine SAKAI (11 of 13)



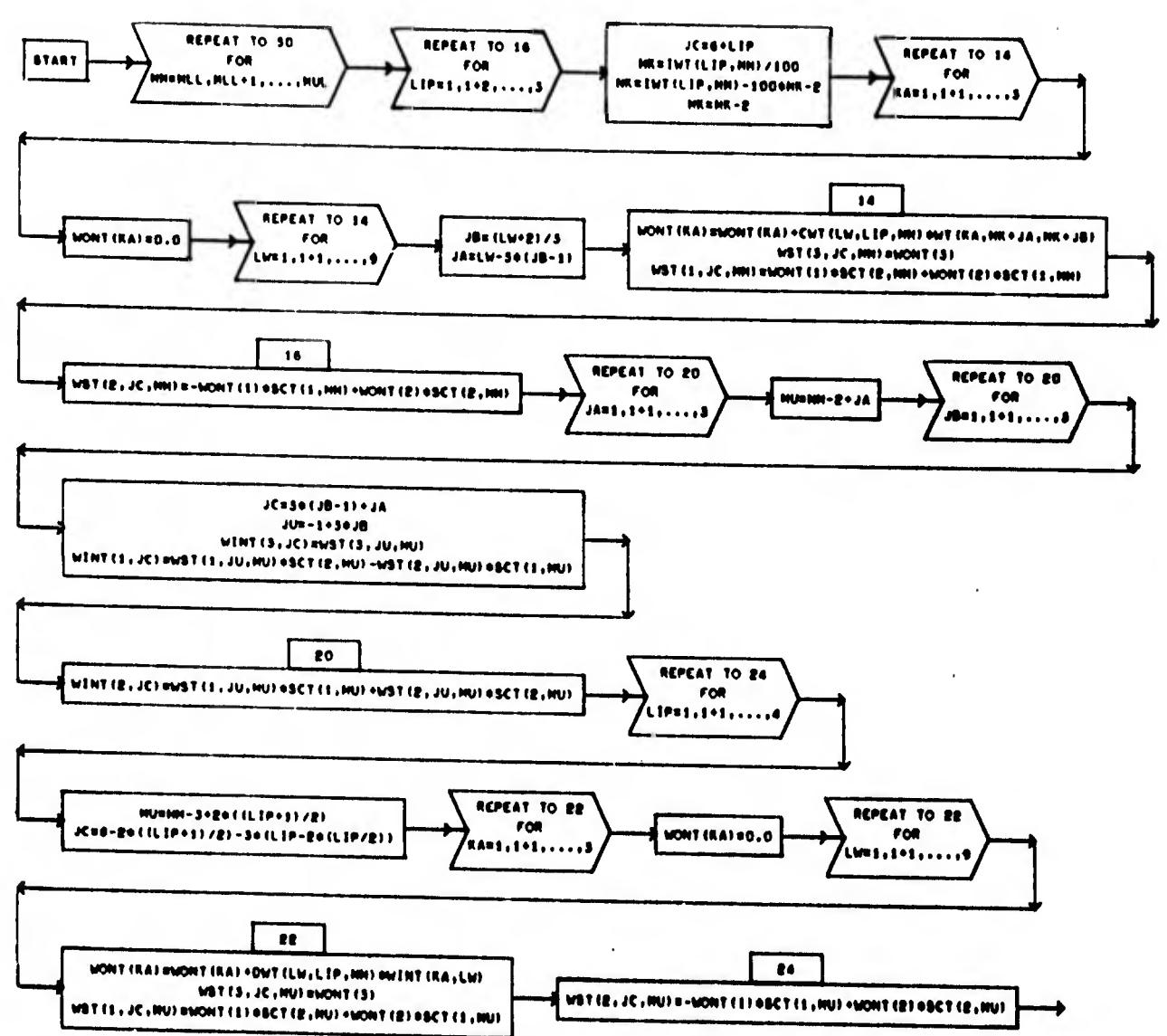
Detailed Flow Charts of Subroutine SAKAI (12 of 13)



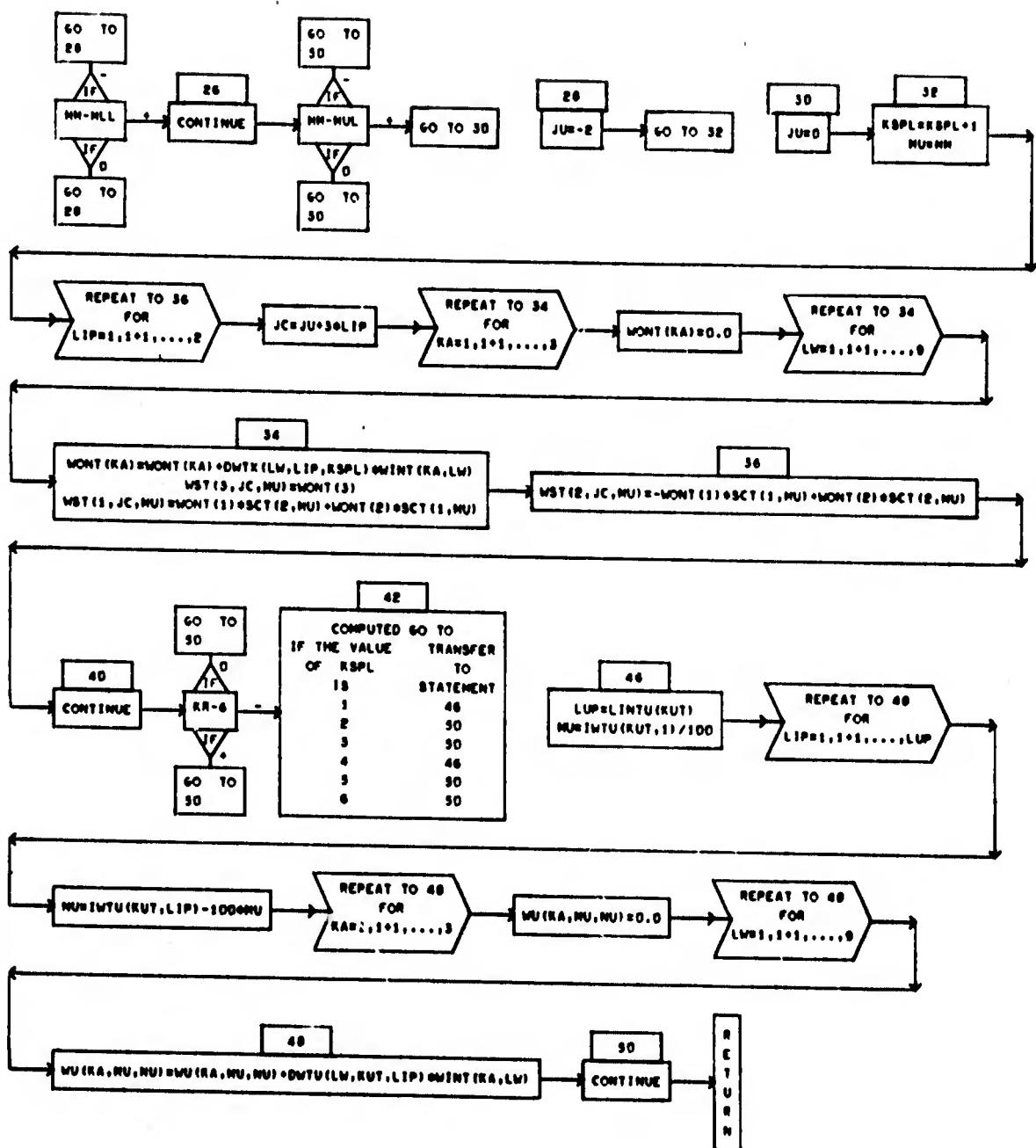
Detailed Flow Charts of Subroutine SAKAI (13 of 13)



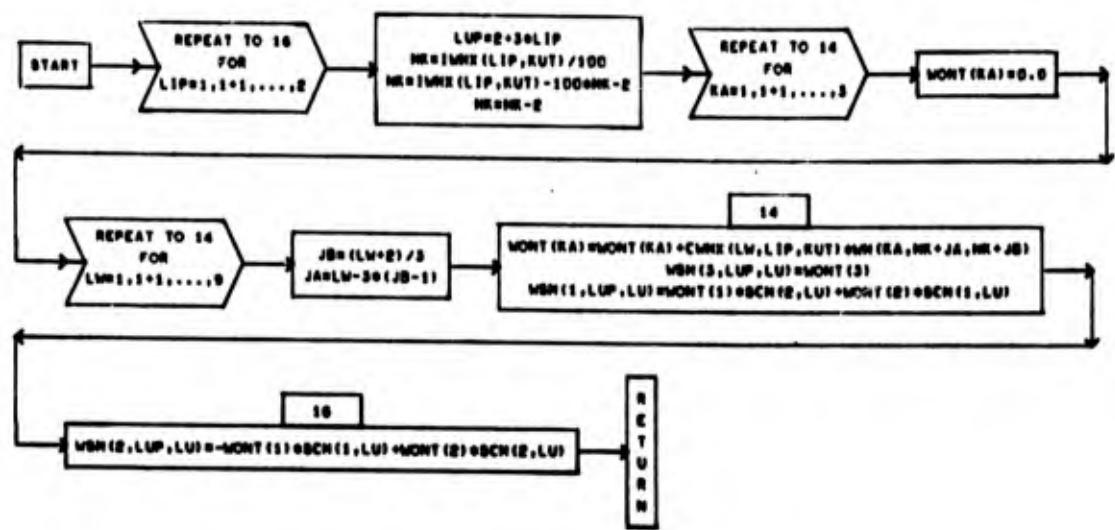
Detailed Flow Chart of Subroutine KAKOI



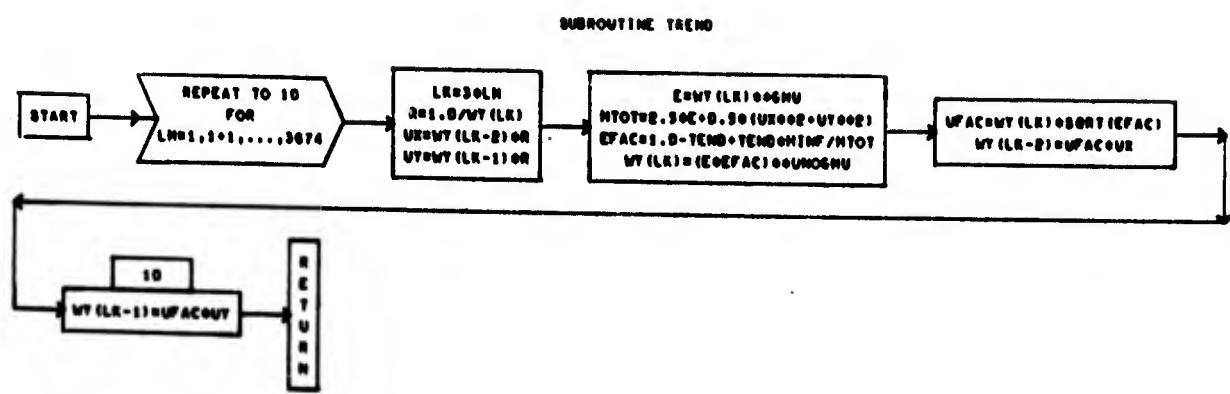
Detailed Flow Charts of Subroutine WAKU (1 of 2)



Detailed Flow Charts of Subroutine WAKU (2 of 2)



Detailed Flow Chart of Subroutine KAKARU



Detailed Flow Chart of Subroutine TREND

APPENDIX VI
BOUNDARY LAYER PROGRAM LISTING

```

PROGRAM TURBL(INPUT,OUTPUT,TAPE6=OUTPUT)
COMMON/VI/COLF/ Z(20),NO,ISIGN,IKIND,T0,PO,IPAR,
COMMON/XSKIN/ ISKIN,DEFL,LF,IL,CD,FOOTL,IFF,ICE,HSEP,HREAT
COMMON/XYZ/ UE,UEDX, PE, TE, ME, RHOE, V0, IM, VM, RHOM, SINR
COMMON/V/CLV/LAN12,INCR,GGAM1,GA,IA,ICOMP,IU,IDE
COMMON/XUP/ DISX(75),TIME(75),PIN(75),MINE(75),TIME(75),BETA(75)
COMMON/FREE/ XIIF,XMIF,QIF,PIF,AIF,UIF,AIF2,UIF2,GXGAM1,CON37,
1 XSTART, DELX, XEND
COMMON/PLQ/ ILAM, MSTART, XVIS, C
COMMON/SAVEIT/ QUAB(17,100)
COMMON/XI1(5),UEI1(5),
REAL PE, IP, ISTARP, MINF
DIMENSION X(3), UEL1(3), UEL2(3), UEL3(3), H12(3), H32(3)
*, LP(75), CFF(3), WUD(10), ETO(3), AFACT(3)
*, LAPI(3), CAUX(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 PRINOUT AT EACH ITERATION
C----IPRF = FLA = READ IN CP DISTRIBUTION AND CALCULATE U DISTRIBUTION
C----ICOMP = U INCOMPRESSIBLE
C----ICOMP = 1. COMPRESSIBLE
C----IDE = 1 USE INCOMPRESSIBLE DIFFERENTIAL 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 YUSH-S CURVE FIT TO SEDDON-S H = H(H1)
C----ISEP = ISKIN = 3 SEPARATED FLOW
C----ISEP = ISKIN = 4 OUR SWF 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 LUDWIG-TILLMANN
C----ICF = 2 USE ALPHA A FUNCTION OF HSEP
C----ITER = LIMIT OF ITERATIONS TO BE ATTEMPTED
C
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----IKIN = 2, PROGRAM SET UP TO HANDLE THIS CASE ONLY
C----ISIGN = 1, NEGATIVE EXPONENTS
C---- = 2 POSITIVE EXPONENTS
C
C
C+++++FLOW TOTAL AND FREESTREAM CONDITIONS
C
C----PO = TOTAL PRESSURE IN PSIA
C----T0 = TOTAL TEMPERATURE IN DEGREES K
C

```

```

C+++++QUANTITIES OUTSIDE THE BOUNDARY LAYER
C
C-----XINF(1) = VELOCITY OUTSIDE THE BOUNDARY LAYER IN FPS
C-----MINF(1) = MACH NUMBER OUTSIDE THE BOUNDARY LAYER
C-----PINF(1) = PRESSURE OUTSIDE THE BOUNDARY LAYER IN PSIA
C-----SET PINF(1) = 0.0 IF ONLY THE VELOCITY DISTRIBUTION IS KNOWN
C-----TINF(1) = 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-----THETA0 = MOMENTUM THICKNESS IN FEET
C-----DEL30 = MASS FLOW THICKNESS IN FEET
C-----ERROR = LIMIT FOR THE ITERATION SCHEME
C-----VISX(I) = LOCATION ALONG SURFACE OF DATA POINT IN FEET
C-----CP(I) = PRESSURE COEFFICIENT
C-----LTA(I) = LOCAL MACH ANGLE OUTSIDE THE BOUNDARY LAYER IN RADIANS
C-----GAM = RATIO OF SPECIFIC HEATS
C-----DELI = DISPLACEMENT THICKNESS
C-----DEL2 = MOMENTUM THICKNESS
C-----DELS = BOUNDARY LAYER THICKNESS MINUS DEL1
C-----XMF = FREE STREAM MACH NUMBER
C-----XTIF = FREE STREAM TEMPERATURE IN R
C-----XIF = FREE STREAM DYNAMIC PRESSURE IN PSI
C-----DELK = COMPRESSIBLE DISPLACEMENT THICKNESS
C-----HSTART = INCOMPRESSIBLE H
C-----ISEP = SEPARATION INCOMPRESSIBLE SHAPE FACTOR
C-----IREAT = REATTACHMENT INCOMPRESSIBLE SHAPE FACTOR
C-----AVIS = LINEAR VISCOSITY KATIV CONSTANT
C-----C = CHORAL IN FEET
C
1 CONTINUE
INDEX = 0
READ 13, (WORD(I), I=1,8)
READ 22, NU, IKIND, SIGN, IPAR
READ 12, ILAM, XVIS, C
16 FORMAT(1B, 5X, 7E10.3)
READ 23, ISEP, ISKIN, IEF, ICF, HSFP, HREAT
READ 2, INCR, ISUG, GAM, XSTART, XEND, DELX, DELK, HSTART, THETA0, DEL30
READ 7, POUT, ERROR, ITER, IPHYF, ICOMP
* 1D
READ 50, XMF, XTIF, PIF
C
C-----PRINT 16, (WORD(I), I=1,8)
C-----IF(ILAM, EQ, 1) PRINT 17
17 FORMAT(//, *IF WILL START WITH A LAMINAR BOUNDARY LAYER*)
C-----PRINT 4, INCR, ISUG, GAM, XSTART, XEND, DELX, DELK, HSTART, THETA0, DEL30

```

```

PRINT 18, XVIS, C
18 FORMAT(1XVIS,E15.5, E10.5)
IF(NO.NE.0) GO TO 32
PRINT 31
90 TO 36
32 PRINT 33, NO
34 PRINT 37, 1SEP, ISKIN, IEE
  GAM12 = 0.5*(GAM - 1.0)
  UGAM1 = GAM/(GAM - 1.0)
  UIF = PIR*U.5+GAM*XMF**2
  PQ = PIR*(1.0 + GAM12*XMF**2)**GGAM1
  T0 = XMF*(1.0 + GAM12*XMF**2)
  AIF = .49.0*SQRT(XTIE)
  UIF = XMIF*AIF
  AIF2 = AIF**2/GAM12
  UIF2 = UIF**2
  GXGAM1 = 1.0/GGAM1
  CON37 = U.5*GAM*XMF**2
  IF(IPROF.NE.1) GO TO 11
  READ 51, IU, (DISX(I), CPL(I), UETIA(I), I=1, IU)
  502 I=1, IU
  54 UINF(I) = SQRT(UIF2 - AIF2*((1.0 + CON37*CPL(I))**GXGAM1 - 1.0))
  GO TO 53
C
C
  11 CONTINUE
  READ 3, IU, (DISX(I), UINF(I), PINF(I), MINF(I), TINF(I),
  1 BETA(I), I=1, IU)
  53 CONTINUE
  AT = .49.0*SQRT(T0)
  IF(.9.GT.1) CALL GAUSSLN(IU, NO, IKIND, ISIGN, DISY, UINF, WORD, Z1)
  PRINT 8, PU, TO, EMRK, ITEM, IPROF
  IF(WORD.EQ.0) GO TO 200
  RHOO = PU*144.0/(55.3*T0)
  VO = VISCOS(T0)/RHOO
  IF(PINF(I).GT.0.0) GO TO 6
  DO 9 I=1,IU
  UAI = UINF(I)/AT
  UAT2 = UAI**2
  IEMP = 1.0 - GAM12*UAT2
  TINF(I) = T0+IEMP
  PINF(I) = PU*T1EMP**GGAM1
  MINF(I) = SQRT(UAT2/IEMP)
  6 CONTINUE
  PRINT 5, (DISX(I), UINF(I), PINF(I), MINF(I), TINF(I), BETA(I), I=1, IU)
  IF(ILAM.EQ.1) GO TO 1999
  IF(ILDE.EQ.2) GO TO 8000
  7000 CALL TRANSWELX, THETA0, DEL30, HSTART, XSTART)
C----STARTING WITH A LAMINAR BOUNDARY LAYER
  CALL LAM1HAK(T1EAU, XSTART, DEL30)
C----RETURNS COMPRESSIBLE BOUNDARY LAYER THICKNESSES
  8000 CALL VEL(XSTART)
  X(1) = XSTART
  X(3) = XSTART
  XP(1) = XP(3) = XSTART

```

```

RUM = RHOF*UE*SINB
UO 10 I = 1,3
UEL1(1) = UELK
UEL2(1) = JHETAO
UEL3(1) = UEL30
H12(1) = HSIART
CAPY(1) = [JHETAO + RUM + UE
CAPX(1) = RUM + UEL30
10 H32(1) = UEL30/DEL2(1)
LFF(1) = SKINF(ISEP,M12(1),M22(1), DEL2(1),DEL1(1))
KEUEL2 = UE+UEL2(3)/VO
CALL SAVE0(INDX,X,X,DEL1,DEL2,DEL3,M12,M32,CF)
IF(I1BUG.GT.0) PRINT 102,X(3),DEL1(3),DEL2(3),DEL3(3),H12(3),
+ H32(3),UE,IE,ME,CF,CO,REDEL2
+ H,LFF(3),A,DUEDX

15. CONTINUE
ILEAG = H
ICOUNT = 0
X(2) = X(1) + H*5*DELX
X(3) = X(1) + DELX
X1 = X(1) + X3 = X(3)
UEL1 = DELX
XP1(3) = X(3)
IF(I1UE,FE,2) GO TO 8001
CALL VELIX(2)
UEL1 = (IE/I0)+4*DELX
XP(3) = XINCOM(X(3))
X1 = XP(1) + X3 = XP(3)
C---SECOND ORDER RUNGE-KUTTA OR TRAPEZOIDAL FORMULA
8001 CONTINUE
CALL SLOPEUX1,M12(1),H32(1),DEL1(1),DEL2(1),DEL3(1),D2DX,D3DX
* 1
20. CONTINUE
CALL SLOPEUX3,M12(3),H32(3),DEL1(3),DEL2(3),DEL3(3),D2DX,D3DX
TEMP2 = CAPY(1) + 0.5*UEL1*(D2DX + D2DX1)
TEMP3 = CAPX(1) + 0.5*UEL1*(D3DX + D3DX1)
TEMP4 = TEMP3/TEMP2 + UE
20. LAPPX(3) = TEMP2
CAPX(3) = TEMP3
H32(3) = TEMP4
H12(3) = 1.5FACT(ISEP,M32(3),HSSEP1)
RUM = RHOF*UE*SINH
TEMP5 = CAPX(3) - RUM
IF(AUS(ITEMP5 - DEL3(3))/DEL3(3) .LT. ERROR) IFLAG = 1
UEL2(3) = CAPY(3)/RUM /UE
UEL3(3) = CAPX(3)/RUM
UEL1(3) = H12(3)*DEL2(3)
IF(I1UE,FE,2) DEL1(3) = DEL2(3) + H
ICOUNT = ICOUNT + 1
KEUEL2 = UE+DEL2(3)/VO
LFF(3) = SKINF(ISEP,H12(3),H32(3),DEL2(3),DEL1(3))
H = FODEL
IF(I1BUG.GT.0) PRINT 102,X(3),DEL1(3),DEL2(3),DEL3(3),H12(3),
+ H32(3),UE,IE,ME,CF,CO,REDEL2
+ H,LFF(3),A,DUEDX

```

```

-- IF(1FLAG.EQ.1) GO TO 100
-- IF(1COUNT.GT.ITER) GO TO 200
GO TO 20
100 IF(1DE.EQ.2) GO TO 8002
REAL = DE*X(3) / VO
TE10 = TE/10
TOTE = T0/10
UCL = UCL+SUM(I(TE10))
REAL = UCL*X(3) / VO
D2 = TOTE**3+DLL2(3)
D3 = TOTE**3+DEL3(3)
D1 = H*D2
CFC = TE10*CFF(3)
REDEL = REDEL2*TOTE**2*SQR(TOTE)
PRINT 339
PRINT 338,X(3),D1,D2,D3,H1H32(3),UCL,TE,ME,CFC,CD,REDEL
C----NOW TEST FOR CHANGE IN REGIONS
8002 CONTINUE
GO TO 10 (1000,2000,3000,4000),1SEP
109 DO 110 I = 1,2
  DEL2(I) = DEL2(3)
  DEL1(I) = DEL1(3)
  DEL3(I) = DEL3(3)
  CAPY(I) = CAPY(3)
  LAPX(I) = CAPX(3)
  X(I) = X(3)
  AP(I) = XF(3)
  LFF(I) = CFF(3)
  H12(I) = H12(3)
  H32(I) = H32(3)
  CALL SAVE,(INDEX,X,DEL1,DEL2,DEL3,H12,H32,CF)
  IF(X(3).LT.XEND) GO TO 15
  IF(1INDEX.GT.100) INDEX = 100
  PRINT 111,((1COUNT(J,I),J=1,7),I=1,INDEX)
  111 FORMAT(7E18.0)
  GO TO 1
C----TEST POWER LAW PROFILE RESULTS
1000 IF(H32(3).GT.9.0) GO TO 109
  1SEP = 2 *ISKIN = 2
  PRINT 1001
  1001 FORMAT(10W SWITCH FROM POWER LAW TO YOSH-S FIT OF SEDDON*)
  GO TO 109
C----TEST OF SANBORUS ONE PARAMETER PROFILE
  2000 IF(H12(3).LT.HSEPI) GO TO 109
C----NOW SWITCH TO SEPARATE FLOW EQUATIONS
  1SEP = 3
  ISKIN = 3
  XSEP = X(3)
  PRINT 337,XSEP
  XSEPT = H32(3)
  U = (H12(3) + 1.0)ALO/TE - 1.0
  DELS(3) = H32(3)*DEL2(3)
  PRINT 338,XSEP,DEL1(3),DEL2(3),DEL3(3),H12(3),H32(3),IE,TE,ME,
  CFF(3),CD,REDEL2
  GO TO 109

```

```

C----TEST SEPARATED SOLUTION FOR REATTACHMENT
3000 1E1H12(3),GT,HSEP) GU TO 109
 1SEP = 4  & ISKIN = 4
PRINT 301, X(3)
301 FORMAT( REATTACHMENT X = + E10.3)
GO TO 109

C----LESJ SOLUTION
4000 1E1H32(3),LI,9,0) GO TO 109
XREAT = X(3)

C----NO. SETUP TO GO BACK TO HEADS TURBULENT METHOD
ISKIN = 1
1SEP = 1
H12(3) = 1.1FACT(1SEP,H32(3),HSEP)
UEL2(3) = UEL1(3)/H12(3)
DEL2(3) = UEL1(3) / H
DEL3(3) = H32(3) * DEL2(3)
CALL VEL(XREAT)
(CFF(3) = SKIN,FLISKIN,H12(3),H32(3), DFL2(3),DFL1(3))
PRINT 338,1SEP,DEL1(3),UEL2(3),DEL3(3),H12(3),U32(3),UE,TE,ME,
* CFF(3), CD, REUEL2
GO TO 109

C
C
200 CALL EXIT
C
C
2 FORMAT(215/8E10.3)
3 FORMAT(15. /6E10.3)
4 FORMAT(* . INCH =+15* IBUG =+15/* GAM =+E10.5* XSTART =+E10.5
1 * XEND =+F10.5, DELX =+F10.5* DFLK =+F10.5/
2 * HSTART =+F10.5 * (HETAU =+ F10.5 + DEL30 =+F10.5)
5 FORMAT(10X*DISX*10X*UINF*16X*PINF*16X*MINF*16X*TINF*16X*HETA*/.
* (0E20.6))
7 FORMAT(3E10.3*4I5)
8 FORMAT(* MU_(PSIA) =+F10.2* TQ_(R) =+F10.2* ERROR =+E10.3
1 * ITFR =+15* IPHF =+15)
13 FORMAT(8A10)
14 FORMAT(1H1*18X,8A10)
2 FORMAT(5I5)
23 FORMAT(4I5, 2E10.3)
31 FORMAT(/ * NO CURVE - FIT OF THE ENTIRE INPUT VELOCITY DISTRIBUTION
* ON * /, IF WILL USE A PARAHULIC FIT OF THREE POINTS*)
33 FORMAT(* CURVE-FIT ENTIRE VELOCITY DISTRIBUTION*/
1 * DEGREE OF POLYNOMIAL =5X,I5)
37 FORMAT(* 1SEP =+15/ * ISKIN =+15/* IFF =+15)
54 FORMAT(3E10.3)
51 FORMAT(I5/(3E10.3))
102 FORMAT(/10X*FT* 13X*DFL1-FT* 13X*DFL2-FT* 13X*DFL3-FT*
1 * 17X*H12* 17X*I32* 6E20.6/ 14X*UE-FPS* 16X*TE-P* 1HX*ME* 18X*CF*
218X*LD* 14X*REUEL2* 6E20.6/ 19X*H*14X*CFF(3)* 19X*AA*
315X*DUEPA* 17X*TA* 5E20.6)
103 FORMAT(17X*REXA*18X*V0*14X*REXBAR*15X.*RFBAR*/.
* 4E20.6/1UX*ICOUNT =+15)
334 FORMAT(* CHECK SEPARATION EXTRAPOLATION*)
337 FORMAT(* SEPARATION HAS OCCURRED AT X =+E10.3)

```

530 FORMAT(1/16X+X-F14.13X+DEL1-F1+ 13X+DEL2-F1+ 13X+DEL3-F1+
1. 17X+H12+ 17X+I32+/6E20.6/ -14X+UE-FPS+ 16X+TE-P+ 18X+ME+ 18X+CF+
218X+LD+ 14X+RDEL2+/6E20.6)
533 FORMAT(*,1TRANSFORMED COMPRESSIBLE VARIABLES*)
542 FORMAT(*1EATTACHMENTS OCCURRED AT X = +E10.3)
/30 FORMAT(1/16X+X-F14.13X+DEL1-F1+ 13X+DEL2-F1+ 13X+DEL3-F1+
1. 17X+H12+ 17X+I32+/6E20.6/ -14X+UE-FPS+ 16X+TE-P+ 18X+ME+ 17X+CU1A
218X+LD2+14X+RDEL2+/6E20.6/
3 17X+RFX+ 17X+LT4+ 15Y +DUDDX+ 18X +F1+
4 18X +F2+ 18X +F3+/6E20.6/
5 18X +F4+/6E20.6)
1NU

```

SUBROUTINE LAMINAR(IHETO,XI,UEL30)
COMMON/COFL/ 2(20),N0,ISIGN,IKIND,T0,P0,IPAR
COMMON/XS11/ ISINH,DELK,CE,IL,CL,FODEL,IEE,ICE,HSEP,NREAT
COMMON/XYZ/  UE, UEDDX, PE, TC, ME, RI0E, VQ, JM, VM, RHQM, SINB
COMMON/CO/ GAM12, INCR, GGAM1, GAM, AT, ICOMP, IU, IDE
COMMON/XUPL/ DISX(75),TINF(75),PIN(75),MINF(75),TINF(75),BETA(75)
DIMENSION XC(50), DEL2(50), U2(50), UEX(61), DEL1(50), D1(50)
COMMON /P0Q/ ILAM, H12, CV12, C
COMMON/UFEE/ XTF, XMIF, UTE, PTF, ATE, UFE, AFE2, UFE2, GXGAM1, CON37
1. XSTART, DELX, XEND
COMMON S175, UEL175
DATA MMAX /60/
REAL MINF, ME
IHETAO = IHETO/C
AFLAL = XEND
G1 = 1.0/GGAM1
MC = UFE*C+144.0*PI/(53.3*XTF*VISCOS(XTF))
PHIL, H, MC
C- FORWARD 1/4 REYNOLDS NUMBER BASED UPON CHORD LENGTH = 4E20.51
C--- UELX IS INPUT VALUE OF STEP SIZE FOR COMPRESSIBLE
UXC = DELX
XC(1) = XSTART
DEL2(1) = IHETAO
MAX = 0
I = 1
LAMSEP = 0
LAMTHIN = 0
U2(1) = IHETAO*(XTF/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
DELUXC = XC(I)/FLOAT(MAX)
AL = - DELUXC
ITER = 1
UO_20, J = 1, MM
AL = XL + UEDDX
CALL THTHP(XL,S,UE,I, TU, UU, UEDDX, ITER)
UE = UU/UE
20. UEX(J) = UU**5
UEUX = UEDDX/UTE
SUB = 0.0
CALL SIMPIMAX, UEDDX, SUB, UEX, UEX5)
C--- UE = LAST POINT OF INTEGRATION
C--- UEL2(I) = THETA / C
UEL2(I) = SQRT(0.45/L)*C*UE**61*UEX5)
NLAM = DEL2(I), C, NC, UEDDX
IF(XLAM.GT.=0.094), GO TO 224
LAMSEP = 1
NLAM = - 0.090
224 IF(XLAM.LT.-0.04), GO TO 225
N12 = 2.01 - 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
(CX = 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,JU,TE,DUM,1)
CF = 2.4*CFX/(BC*DEL2(I))
H1 = IO/TE*(H12 + 1.0) - 1.0
D2(I) = DEL2(I)*(TINF/TE)**3 + CVIS
D1(I) = D2(I) + H1
RX = RC*XC(I)
RTHETA = PC*DEL2(I)
KIRANS = 1.718*RX + 0.435
IF (RTHETA.GE.KIRANS) LAMTRAN = 1
PRINT 150,I,UE,YC(I),XCOMP,DEL2(I),D2(I),DEL1(I),D1(I),CF
1 RX, RTHETA, KIRANS, XLAM, LAMSEP, LAMTRAN
IF (LAMSEP.EQ.1) PRINT 151
151 FORMAT(* LAMINAR SEPARATION HAS OCCURRED *)
IF (LAMTPA1.EQ.0) GO TO 55
PRINT 152
152 FORMAT(* LAMINAR TRANSITION TO TURBULENT FLOW HAS OCCURRED *)
DEMH = 0.657 + 0.101*ALUG1/RKTRANS
HTURB = H12 - DEMH
H = IO/TE*(H12 + 1.0) - 1.0
H1 = 2.0*H12/(H12 - 1.0)
H12 = HTURB
ISTART = H12
XT = XC(I)
DELK = D2(I)*H - C
DEL30 = D2(I)*H1 + C
THE1 = 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
154 FORMAT(1/19X+1+13X+UE/DINF*18X+XC*6X+X-COMPRESSIBLE*14X+DEL2/C*
1 11X+DEL2-COMP*/12U+5E20,5/
2 3UX+DEL-STAB/C+6X+DEL-STAB-COMP*11X+CF-INCOMP*16X+RE-X*
3 12X+RE-THERM/2UX+5E20,5/
4 32X+RE-THERM/16X+XLAM*10X+SEPARATION*10X+TRANSITION*/
5 2UX+2E20,5+2I20)
ENW

```

```

SUBROUTINE TRAIS(DELX, THETA0, DEL30, HSTART, XSTART)
COMMON/XFILE/ XIF,XMF,UIF,PIF,AIF,UIE,AIF,,UIE2,GXGAM1,CON3Z,
  XSTART, DELY, AEND
COMMON/PUG/ ILAM, H12, CVJS, C
COMMON/XSKIN/ ISKIN,DELK,LFL,CD,FOQEL,IFE,ICF,HSEP,HREAT
COMMON/COLF/ /ICOL,HO,ISIGN,IKIND,TO,PO,IPAR
COMMON/XYZ/ UE, UEDA, PE, TE, ME, RHOE, VO, VM, KHOM, SINR
COMMON/CUR/GAM12, INCR, GGAM1, VAM, AT, ICOMP, IU
  ,IUE
COMMON/XUPIL/DISX(75),UINF(75),PIFL751,MINE1751,TINE1751,BETA(75)
REAL MINE,ME
COMMON XI(75),UEI(75)
DIMENSION TETO4(100), COFF(3)
XI(1) = DISX(1)
UE(1) = UINF(1)/SQR(TINF(1)/XTIF)
DO 50 I=2,IU
  11 = I - 1
  IF(I,LT,1U) GO TO 21
  11 = I - 2
  21 X = DISX(11) - DISX(I-1)
  N = 10
  CALL PAPFL(X=DISX(11),TINE(11),COFF=X,DUM,DUM)
  DX = X/FLOAT(N)
  XC = DISX(1-1) - DX
  JMAX = N + 1
  DO 31 I=1,IU-1,NMAX
    XC = XC + DX
    IF(I,LT,1U,AND,I1,EQ,NMAX) XC = DISX(IU)
    IE = COFF(1) + XC*COFF(2) + XC**2*COFF(3)
    TETI = TE/XTIF
    33 IE = UINF(1) - TETI**4
    ITEMP = D*0
    DO 34 IN=2,NMAX
      34 ITEMP = ITEMP + 0.5*(TETO4(IN) + TETO4(IN-1))
      CVIS = VISCOSITY_HA10_CONSTANT
      XI(IN) = XI(I-1) + DX*ITEMP*CVIS
      UEI(IN) = UINF(IN)/SQR(TINF(IN)/XTIF)
  50 CONTINUE
  PRINT 75,(XI(I), UEI(I), I = 1,IU)
  75 FORMAT(*,1X,INCOMPRESSIBLE PROFILE*/.19X*X*.16X*UINF*/12E20.6*)
C-----TRANSFORM INITIAL VALUES TO INCOMPRESSILE FORM
  CALL VEL(XSTART)
  JETI = TE/XTIF
  JETO = TE/JD
  T3 = JETI**3
  JHETA0 = JHETA0+T3
  DEL30 = DEL30+T3
  DELK = DELK+T3
  HSTART = HSTART
  HRAK = TEJO*(H + 1.0) - 1.0
  HIC = HRAK
  PRINT 79
  79 FORMAT(*,1X,INCOMPRESSIBLE STARTING CONDITIONS *)
  PRINT 80,TEK,JHETA0,DEL30,HRAK
  80 FORMAT(1X*DX*DELK */F15.11BX*THETA0 */E15.7/9X*DEL30 */F15.7/

```

3BX+BSTART F4E15-71
RETURN!
ENU

```

SUBROUTINE SLOPE(U,H12,H32,UEL1,UEL2,DFL3,RD2DX,DU3DX)
COMMON/XSF/UN/ ISKIN,UELK,CF,M ,CD ,FODEL,IFE,ICF,HSEF,HREAT
COMMON/XY/ UE, DUEUL, PEL, IE, NL, RUE, VO, TM, VH, HMOV ,SIND
COMMON/CUT/ LAV12, INCH, GGM1,VAM,AT,ICOMP,IUL ,IDE
REAL ME, IP, ISTAMP, MINF
CALL VEL(X)
CFF= SKIN(ISKIN,H12,H32,UEL2,DEL1)
DO IU(10,20), IDE
10 CONTINUE
  UELA = U.5*CFF - DUEDX/UE + UEL2 +(2.0 + H12)
  UDSUA = CFF - DUEDX/UE + QELS
  RETURN
20 RUEU = PUE*UE
  REUEM = RUEU* SINH
  UDSUA = RUEU * CU
  RUEU = PUE*UE
  UDSUA = U.5*CF*REUE - DEL1*REUEM + DUEDX
  RETURN
END

```

```

SUBROUTINE VEL(IX)
COMMON/ONCUM/GAM12,INCH,GGAM1,6AM,8T,ICOMP,IU,IDE
COMMON/COFF/ 1/201,0,ISIGN,IKINC,T0,P0,IPAR
COMMON/XUP3/,DISX(75),UINF(75),PINF(75),MINE(75),TIME(75),BETA(75)
COMMON/UVXYZ/ UE, UUEDX, PE, IE, ME, RHOE, VQ, JM, VM, RHOM, SINB
REAL ME, MINE
DIMENSION UOFF(3)

C
IF(IG.GT.1) GO TO 40
CALL INTERIX,DISX,LIND,XOW,IU,L1
L2 = L1 + 1
L1 = L1 - 1
IF(IPAR.EQ.1) GO TO 100
IF(LIND.GT.IU) GO TO 60
UE = UINF(L1)
PE = PINF(L1)
IE = TIME(L1)
ME = MINE(L1)
SINB = SIN(BETA(L1))
IF(L1.GT.1) GO TO 30
L1 = L1
30 UUEDX = (UINF(L2) - UINF(L1))/(DISX(L2) - DISX(L1))
31 RHOE = PE*194.0/(153.3*IE)
RETURN
40 L0N1 = 1.0 - XOW
UE = UINF(L1)*L0N1 + XOW*MINE(L2)
PE = PINF(L1)*CON1 + XOW*PINF(L2)
IE = TIME(L1)*CON1 + XOW*TIME(L2)
ME = MINE(L1)*CON1 + XOW*MINE(L2)
TH = META(L1)*CON1 + XOW*META(L2)
SINB = SIN(BETA(L1))
L1 = L1
GO TO 30
C----USE FIT OF ENTIRE VELOCITY PROFILE.
50 AX = X
NO1 = NO + 1
P = 1.0
UE = 0.0
UUEDX = 0.0
DO 92 J = 1,NO1
UE = UE + P*Z(J)
92 P = P*XX
P = 1.0
DO 94 J=2,NO1
UUEDX = UUEDX + P*Z(J) *FLUAI(J-1)
94 P = P*XX
95 UAI = UE/AI
UAT2 = UAT**2
IF(IE.EQ.2) GO TO 945
TEMP = 1.0 + GAM12*UAT2
IE = T0/TEMP
PE = P0/TEMP + GGAM1
ME = UAT
GO TO 31
945 TEMP = 1.0 - GAM12*UAT2

```

```
IE = IQ+TEMP  
PE = PQ+TEMP+GGAM1  
WF = SORT(UAT2/TEMP)  
GO TO 31  
100 IF(L1.EQ.1) GO TO 101  
IF(XOW.GT.0.5) II = LI  
IF(II.GT.IU-2) II = IU-2  
GO TO 102  
101 II = 1  
102 LQNIINUE  
XX = X  
CALL PAPF1(WISX(11),WINF(11),COFF,XX,WF,DUDX)  
CALL PARFIT(WISX(11),BETA(11),COFF,XX,IR,DUM)  
SINH = SIN(TH)  
GO TO 95  
END
```

```

----- SUBROUTINE GAUSS(NPTS, N, IKIND, JSIGN, X, Y, WORD, Z) -----
C-- NPTS = NUMBER OF DATA POINTS.
C-- N    = DEGREE OF POLYNOMIAL TO BE FITTED.
C-- IKIND = 1 EQUATION TO BE FITTED IS.
C--          Y1=Y1+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-- JSIGN = 1 NEGATIVE EXPONENTS.
C-- JSIGN = 2 POSITIVE EXPONENTS.
----- DIMENSION A(20,21), X(100), Y(100), Z(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
1  IF(IFLAG-IKIND).NE.0
  PRINT 7
  RETURN
4  N01 = N0+1
  N02 = N0+2
  DO 4 J=1,N01
  DO 4 J=1,N02
    A(1,J) = 0.0
  DO 40 I=1,NPTS
    K = Y(I)
    DO 40 JQ=(10,12),IKIND
      R = LOGE(K)
      S = X(I)
      DO 40 IO=(14,16),JSIGN
        L = 1.0/S
        U = -1.0
        A(1,N02) = A(1,N02) + R
        DO 40 J=2,N01
          U = U*S
          A(1,J) = A(1,J) + U
        A(1,N02) = A(1,N02) + U*R
        DO 40 K=2,N01
          U = Q*S
        A(K,N01) = A(K,N01) + Q
        DO 40 J=2,N01
          U = U*S
        A(1,J) = A(1,J+1)
      A(1,11) = NPTS
      CALL SOLVE (A, Z, C, N01)
      SUM = 0.0
      DO 50 I=1,NPTS
        K = X(I)
        P = 1.0
        Q = 1.0
        DO 52 J=1,N01
          P = P*Z(J)
        Z = Z + P*Q
      52 P = P*Q

```

```
ERR = 100.* (Y(1)-ZY)
YERR = ERR*ERR
ERR = ERR/Y(1)
SUM = SUM + YERR
PRINT 54, X(1), Y(1), ZY, ERR, YERR
54 FORMAT (5X,3E15.7,F12.5,F15.7)
55 CONTINUE
PRINT 56, SUM
56 FORMAT (1H0,62X,F15.7)
RETURN
7 FORMAT (/72H SOME OF THE Y VALUES ARE ZERO OR NEGATIVE, LN(Y) = EL
1A1 CANNOT BE USED.)
12 FORMAT (BA10)
15 FORMAT (1H1//12H INPUT DATA, 18X,BA10//1)
17 FORMAT (5X,2E15.7)
ENU
```

```

----- SUBROUTINE SOLVE (A, X, C, NO) -----
C---- SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS BY
C---- GAUSS'S ELIMINATION SCHEME (LU(LITTLE'S) METHOD).
----- DIMENSION A(20,21), B(20,21), X(20), C(20), LOC(20), ROW(20)
----- NO = NO+1
----- DO 150 I=1,NO
----- 150 DO 150 J=1,KNO
----- 150 A(I,J) = A(I,J)
----- 154 PRINT 15
----- 15 FORMAT (1H1//17H MATRIX TO SOLVE.)
----- DO 5 I=1,10
----- 5 PRINT 107, (A(I,J), J=1,NO)
----- 107 FORMAT 1/(5X,E14.6)
----- PRINT 109
----- 109 FORMAT (//)
----- PRINT 107, (A(I,NO+1), I=1,NO)
----- PRINT 13
----- 13 FORMAT (1H1//10UH      I      X(I)      I      X(I))
----- 1      I      A(I,I)      I      -X(I)      I      X(I)
----- 254 DO 14 M=1,NO
----- LOC(M) = 0
----- 14 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
----- 2 IF (AMAX = ABS(A(K,I))) 3,2,2
----- C--- IS ME MAX IN NO PREVIOUSLY USED AS PIVOT.
----- 3 IF (NO*(K) = 4,4,2
----- 4 L0(L) = K
----- AMAX = ABS(A(K,I))
----- 2 CONTINUE
----- 1F (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
----- ELEMENT.
----- DO 50 J=1,NO
----- 50 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(I,NO+1)/A(L,I)
----- 254 PRINT 103, (I, X(I), J=1,NO)
----- 103 FORMAT 14(1B,2X,E15.8)
----- PRINT 109
----- DO 120 I=1,NO
----- SUM = 0.0

```

```
100 M=1,100
110 SUM = SUM + U(L,M)*X(M)
120 L(L) = SUM
130 PRINT 122, (C111,L11,N0+11, I=1,N0)
140 FORMAT (//,(5X,2E15,a))
150 RETURN
94 PRINT 104
104 FORMAT (5X,27H NO UNIQUE SOLUTION EXISTS.)
114 RETURN
124 ENU
```

```

----- SUBROUTINE INTER (WNU, WOMEGL, JIND, WNW, NW, JI)
----- DIMENSION WOMEGL(JI)
----- JI = 1
----- JUU = NW
----- J0 = NW/2
----- 1 IF(WNU-WOMEGL(J0))18,523,25
----- 0 JUU = J0
----- 10 = (J0 + JIU)/2
----- GO TO 26
----- 25 JI = J0
----- 10 = (J0 + JIU)/2
----- 26 IF(JUU-JI-1) 23,27,9
----- 27 J0 = JI
----- 1F(WNU - WOMEGL(J0))20,23,29.
----- 9N WRITE(6,42) "N"
----- 92 FORMAT(22H ERROR IN INTER.WNU = ,E20.5)
----- CALL EXIT
----- 323 JI = J0
----- JUU = JI + 1
----- 23 JINL = J0
----- GO TO 60
----- 24 IF(WNU = WOMEGL(JUU))101,62,90
----- 62 JINL = JUU
----- GO TO 60
----- C---SET FLAG TO INDICATE INTERPOLATION IS NECESSARY
----- 61 JINL = 1000
----- NW = (WNU - WOMEGL(J1))/ (WOMEGL(JUU) - WOMEGL(J1))
----- 60 RETURN
----- END

```

```

SUBROUTINE INTERP(WNU,WOMEG,ULAM,NW,ANS,DERV,ITER)
DIMENSIONI WOMEGL(1),ULAM(1),COFF(3)

C
DEKV = 0.0
JI = 1
JUU = NW
JO = NW/2
1 IF (WNU-WOMEGL(JI))>5.23*25
2 JUU = JO
3 JO = JO + JIU/2
GO TO 26
25 JI = JO
4 JO = (JO + JUU)/2
26 IF (JUU-JI)<2.27*9
27 JO = JI
5 IF (WNU = WOMEGL(JO))>23.29
9U PRINT 92,WNU
92 FORMAT(2/M ERROR IN INTER WNU = ,F20.5)
CALL EXIT
523 JI = JO
530 JIU = JI + 1
23 JINU = JO
GO TO 60
24 IF (WNU = WOMEGL(JIU))>61.62*90
62 JINU = JUU
60 GO TO 60
C---SET FLAG TO INDICATE INTERPOLATION IS NECESSARY
61 JINU = JUCL
6U ANS = (WNU - WOMEGL(JI))/ (WOMEGL(JUU) - WOMEGL(JI))
GO TO (7U,100),ITER
104 II = JI - 1
1IF (II.EQ.1) GO TO 101
1IF (XNU.GT.U.5) II = JI
1IF (II.GT.(NW-2)) II = JUW - 2
101 II = 1
102 CONTINUE
CALL PARFIT(WOMEGL(II),ULAM(II),COFF,WNU,ANS,DERV)
RETURN
7U ANS = WNU*(ULAM(JUU) - ULAM(JI)) + ULAM(JI)
DERV = (ULAM(JUU) - ULAM(JI))/ (WOMEGL(JUU) - WOMEGL(JI))
RETURN
ENU

```

```

-----  

SUBROUTINE PARFIT(Y,D,CUEF,YY,ANS,DANSY)  

-----  

DIMENSION Y(3),D(3),COEF(3)  

C---GIVES COEFFICIENTS FOR A PARAHOLIC FIT  

A1 = Y(1)  

A2 = Y(2)  

A3 = Y(3)  

D1 = D(1)  

D2 = D(2)  

D3 = D(3)  

FACTOR = X2*X3*X3 - X2*X2*X3 - X1*X3*X3 + X1*X2*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  

L1 = X2*X2 - X3*X3  

L2 = X3*X3 - X1*X1  

L3 = X1*X1 - X2*X2  

C1 = X3 - X2  

C2 = X1 - X3  

C3 = X2 - X1  

COEF(1) = (D1*A1 + D2*A2 + D3*A3) / FACTOR  

COEF(2) = (D1*L1 + D2*L2 + D3*L3) / FACTOR  

COEF(3) = (D1*C1 + D2*C2 + D3*C3) / FACTOR  

ANS = COEF(1) + COEF(2)*YY + COEF(3)*YY**2  

DANSY = CUEF(2) + 2.0*COEF(3)*YY  

RETURN  

END

```



```

FUNCTION SKINE( IS, H12, H32, DEL2, DEL1)
  LOB, U11, XYZ1, UE, DUEDX, PEL, TE, ME, RHOM, VO, IM, VM, RHUM, SINB
  COMMON/YSKIN/ ISKIN, UELK, CE, U1, CD, FODEL, IFE, ICF, HSFP, HREAT
  COMMON/COL/GAM12, LUC, GGM1, GAM, AI, ICOMP, IU, IDE
  COMMON/COE/ L201, NO, ISIGN, IKIND, TO, PO, IPAR
  REAL ME, IP, ISTAHP, MINF
  IM = 0.5*(I0 + TE) + 0.22*(IU - IE)
  GO TO (I0, 10, 30, 10) IIS
10 CONTINUE
  ALPHA = U_246*EXP(L=1.501*H12)
  IF(IFCF.EQ.11) GO TO 11
  TEMP = HSFP/H12
  IF(IIS.EQ.41) TEMP = HREAT/H12
  IF(IFTEMP.LE.1.0) GO TO 30
  ALPHA = U_054*(1.95+ ALOG10(TEMP)) + 1.705
11 RDEL2 = UE+DEL2 / VO
  IF(IUE.EQ.21) RDEL2 = UERUDE*DEL2/VISCOS(TM)
  CON1 = PFRCL2*0.268
  CE = ALPHA/CON1
20 IF(IFC.EQ.11) GO TO 21
  LONK = (H12 + 1.0)/((3.0*H12 - 1.0)*(2.0*H12 - 1.0))
  HK = H12 + (1.0 + (0.2*ME + 2.0*LONK*(H12 - 1.0)*2/H12))/(
  1 - 1.0 + 0.2*ME + 2.0*L1*U - H12*COL21)
  H1A = 3.4 + 1.47/LUK - 0.5)*3.8
  LD = U_0306/(H1K - 3.0)**U_653
  GO TO 22
21 CD = 0.0306/(H12 - 3.0)**U_653
22 SKINE = CE
  FQUEL = CE/(DEL2*HS2)
  CF = CF*IE/TO
  IF(IUE.EQ.11) RETURN
  CF = SKINE*IE/TM
  RETURN
30 LF = 0.0
  GO TO 20
END

```

```

FUNCTION HFACLT(ISEP,HS,HSEPI)
COMMON/XYZ1/ UE, DUDDX, PE, IC, ME, RHOE, VO, IM, VM, KHOM, SINB
COMMON/YSKIN/ ISATII, UFLK, LFM, CD, FOFI, IFF, ICF, HSEPI, HREAT
COMMON/COEF/ Z1201, NO, ISIGN, IKIND, IQ, PO, IPAR
REAL ME
DATA XMC,HO/1.44,1.205/
GO TO (-1,5,5,50), ISEP
C----NORMAL TURBULENT FLOW POWER LAW PROFILE
1 HEALT = HS/HS - 2.0
10 TEMP = (TO/TE)
11 E = (HFACLT + 1.0)*TEMP - 1.0
HSEPI = 3.0
RETURN
C----YOSH-S FIT TO SEDDON-S DATA
3 CONTINUE
HEALT = 1.0 + 2.0*(115.797/(HS + 9.297))**2
GO TO 10
C----REATACHMENT
50 HEACLT = 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/CUL/GAM12,INCGAGM1,GAM,AT,ICOMP,IU
      IUE
      COMMON/XUP1/,DIS(175),UNIF(175),PIN(175),MIN(175),TINF(175),BETA(175)
      COMMON X1(175),UF1(175)
      DIMENSION LOFF(3)
      CALL INTPK(X,X1,LIND,XOW,IU,L1)
      IF(L1.EQ.1) GO TO 101
      IF(XOW.GT.0.5) L1 = LI
      IF(L1.GT.IU-2) L1 = IU - 2
      GO TO 102
101  L1 = 1
102  CONTINUE
      XX = X
      CALL PAPFT1(DISX(L1),X1(L1),LOFF,XX,TEMP,DUM)
      IINCOM = TEMP
      RETURN
      END
```

FUNCTION VISCOSE

C----VISCOSITY OF AIR IN UNITS OF LB/FT-SEC

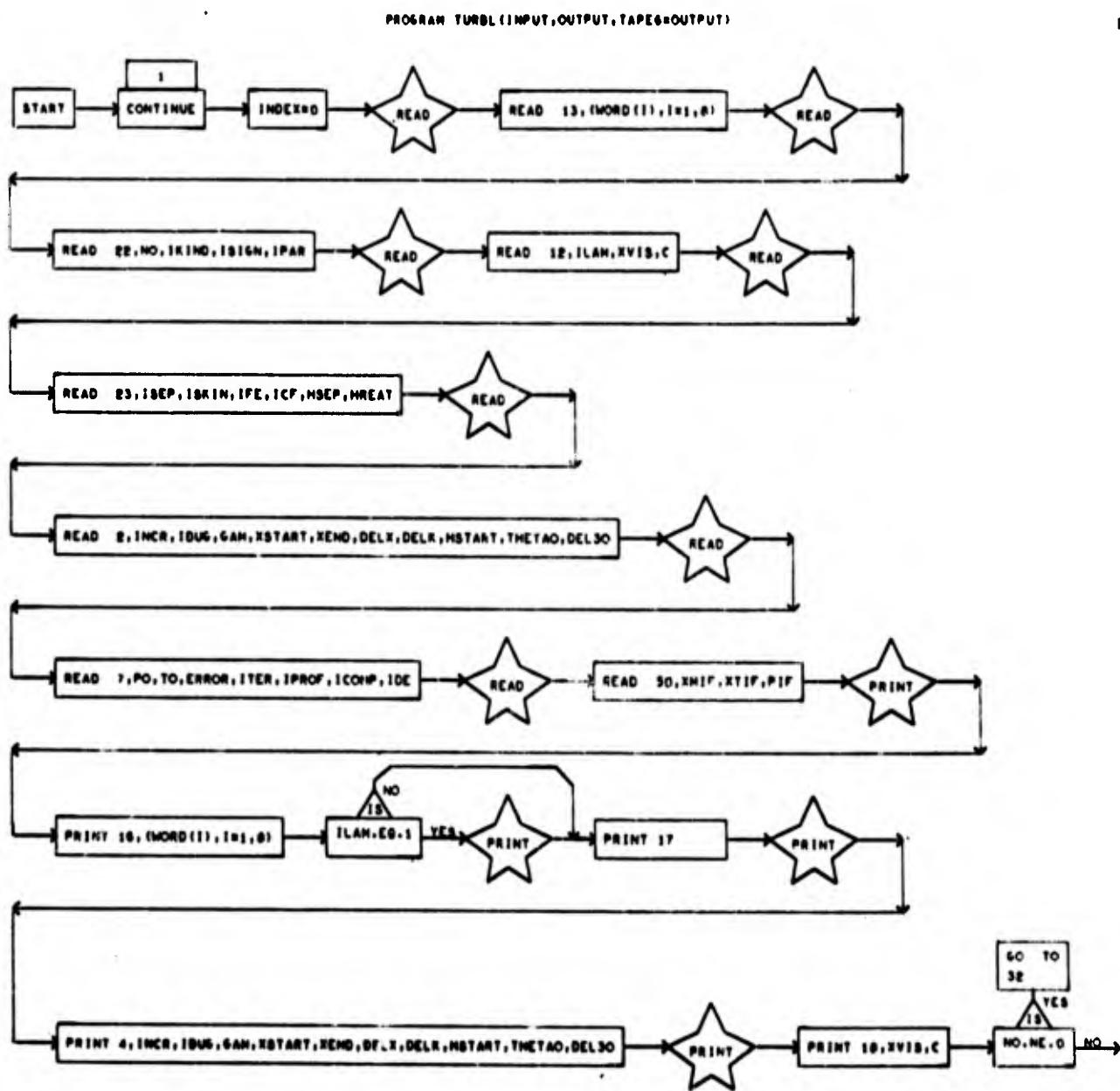
$$VISCOSE = 71.484E-8 \cdot T + 1.5 / (T + 180.0)$$

RE LUMN

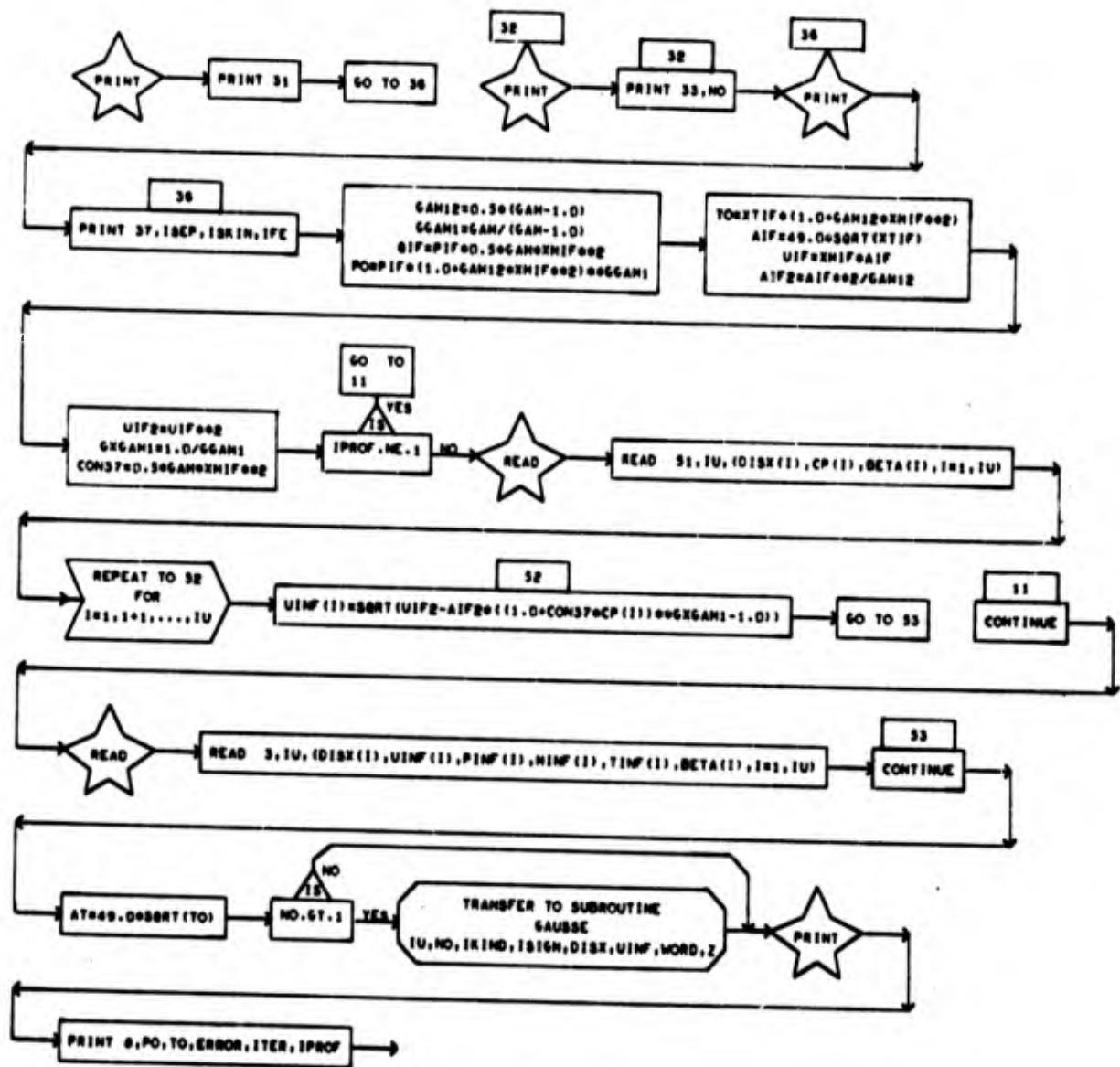
ENW

```
-- SUBROUTINE SAVEQ(INDEX,X,J1,L,U2,D3,H12,H32,CF)
-- COMMON/SAVEIT/,QUANT(7,100)
INDEX = INDEX + 1
IF(INDEX.GT.100) RETURN
L = INDEX
QUANT(1,1) = X
QUANT(2,1) = D1
QUANT(3,1) = D2
QUANT(4,1) = D3
QUANT(5,1) = H12
QUANT(6,1) = H32
QUANT(7,1) = CF
RETURN
END
```

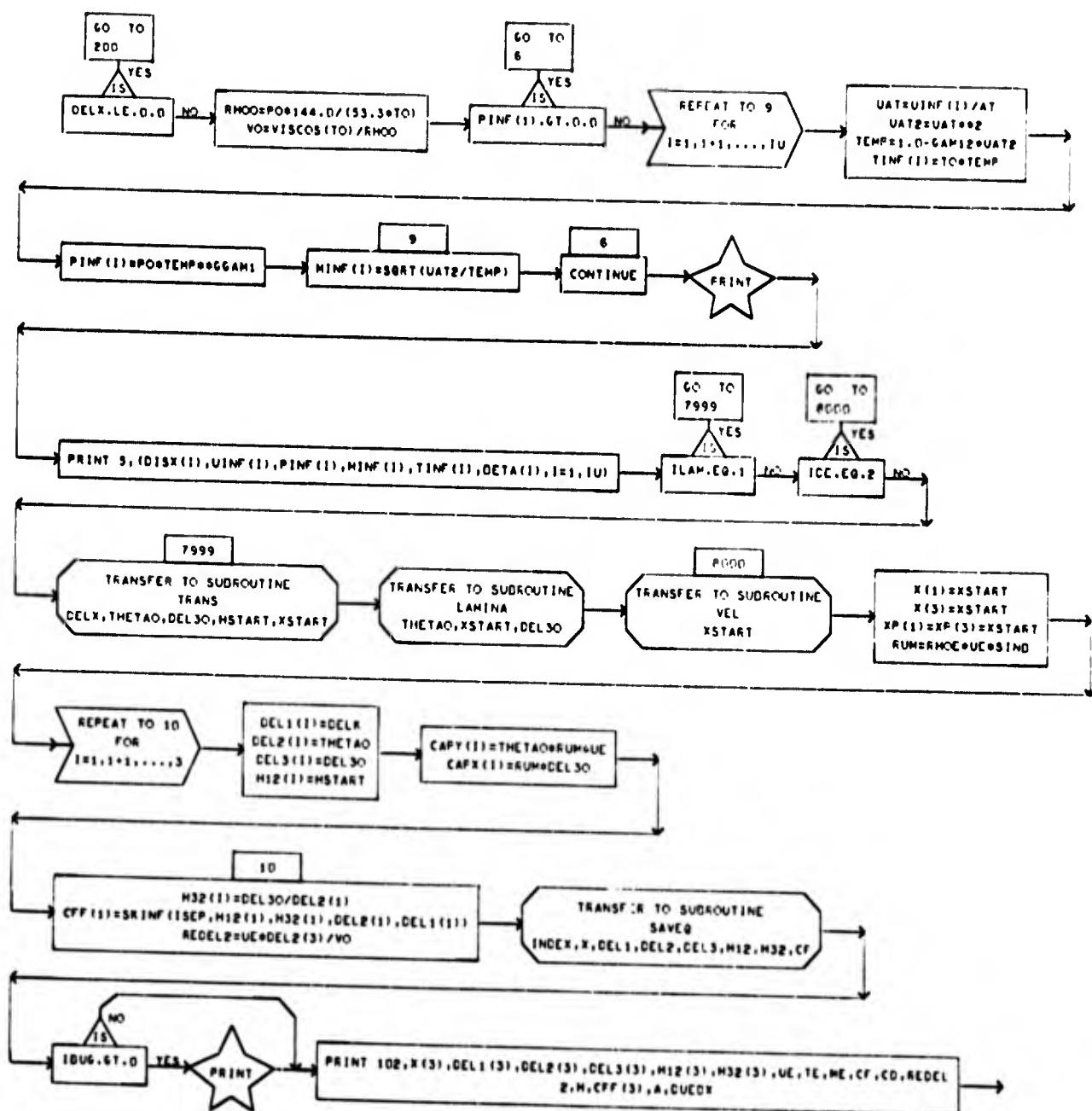
APPENDIX VII
BOUNDARY LAYER PROGRAM FLOW CHARTS



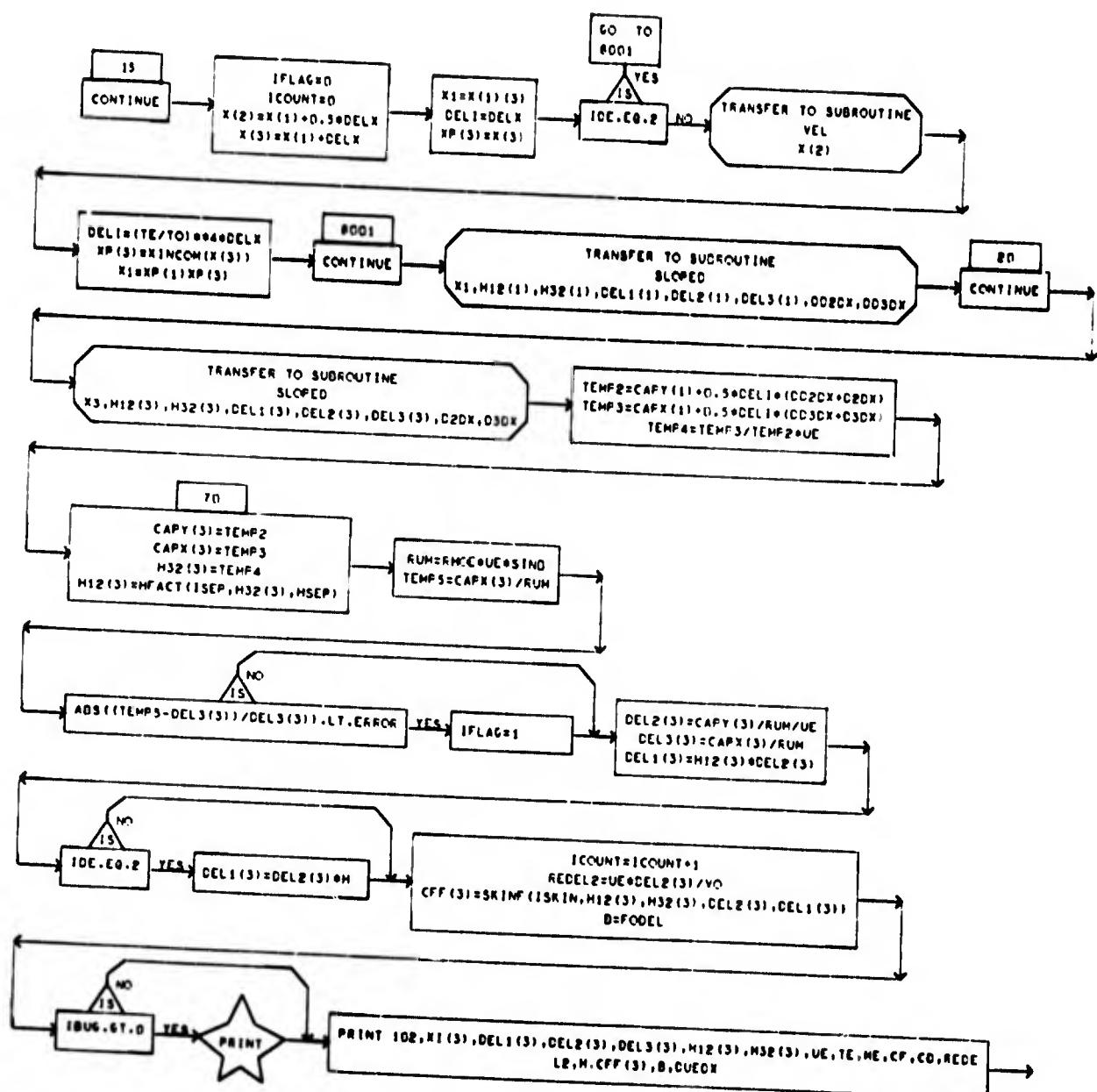
Detailed Flowcharts of Subroutine TURBL (1 of 7)



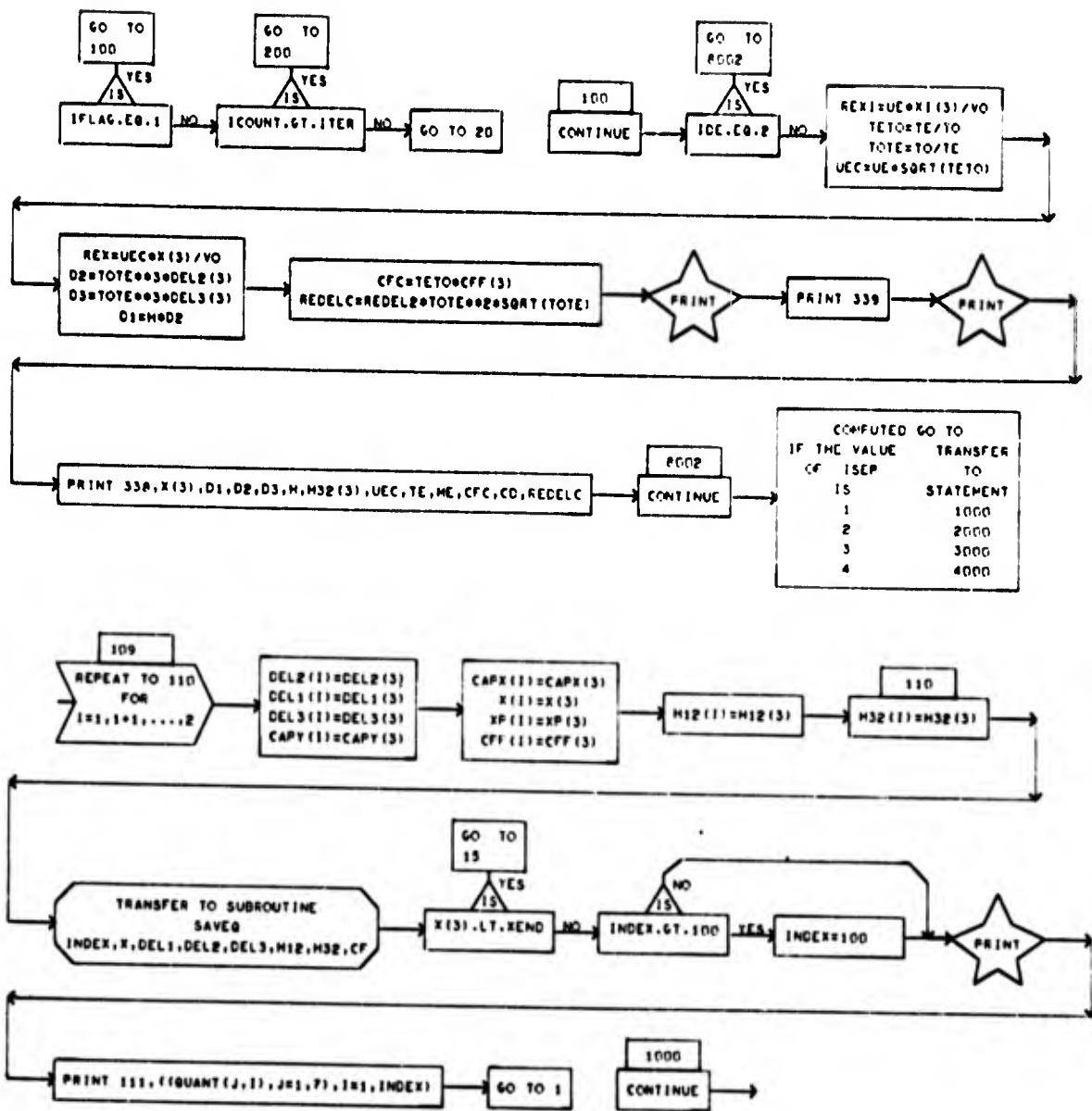
Detailed Flow Charts of Subroutine TURBL (2 of 7)



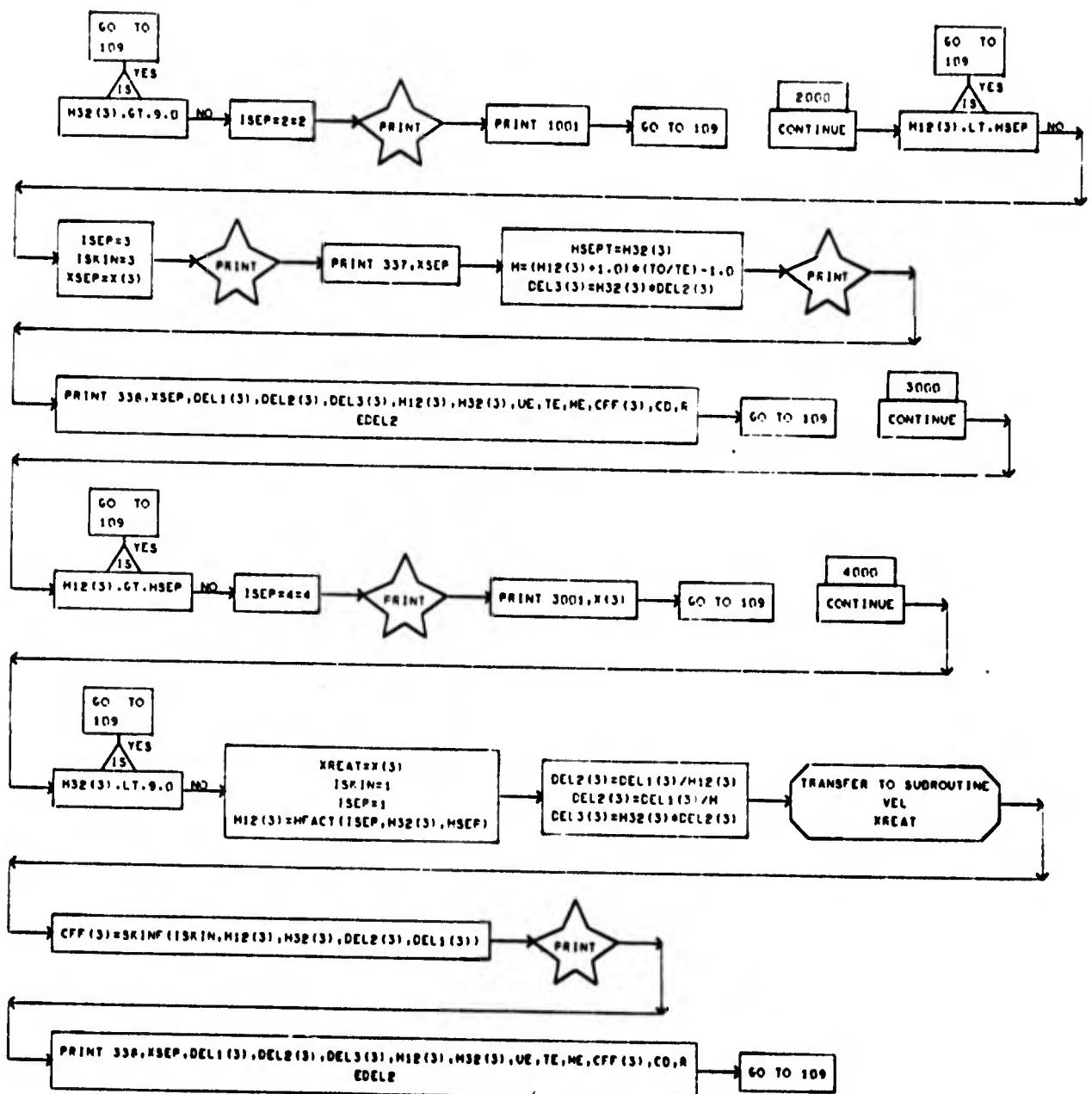
Detailed Flow Charts of Subroutine TURBL (3 of 7)



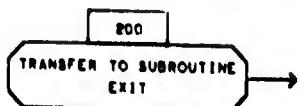
Detailed Flow Charts of Subroutine TURBL (4 of 7)



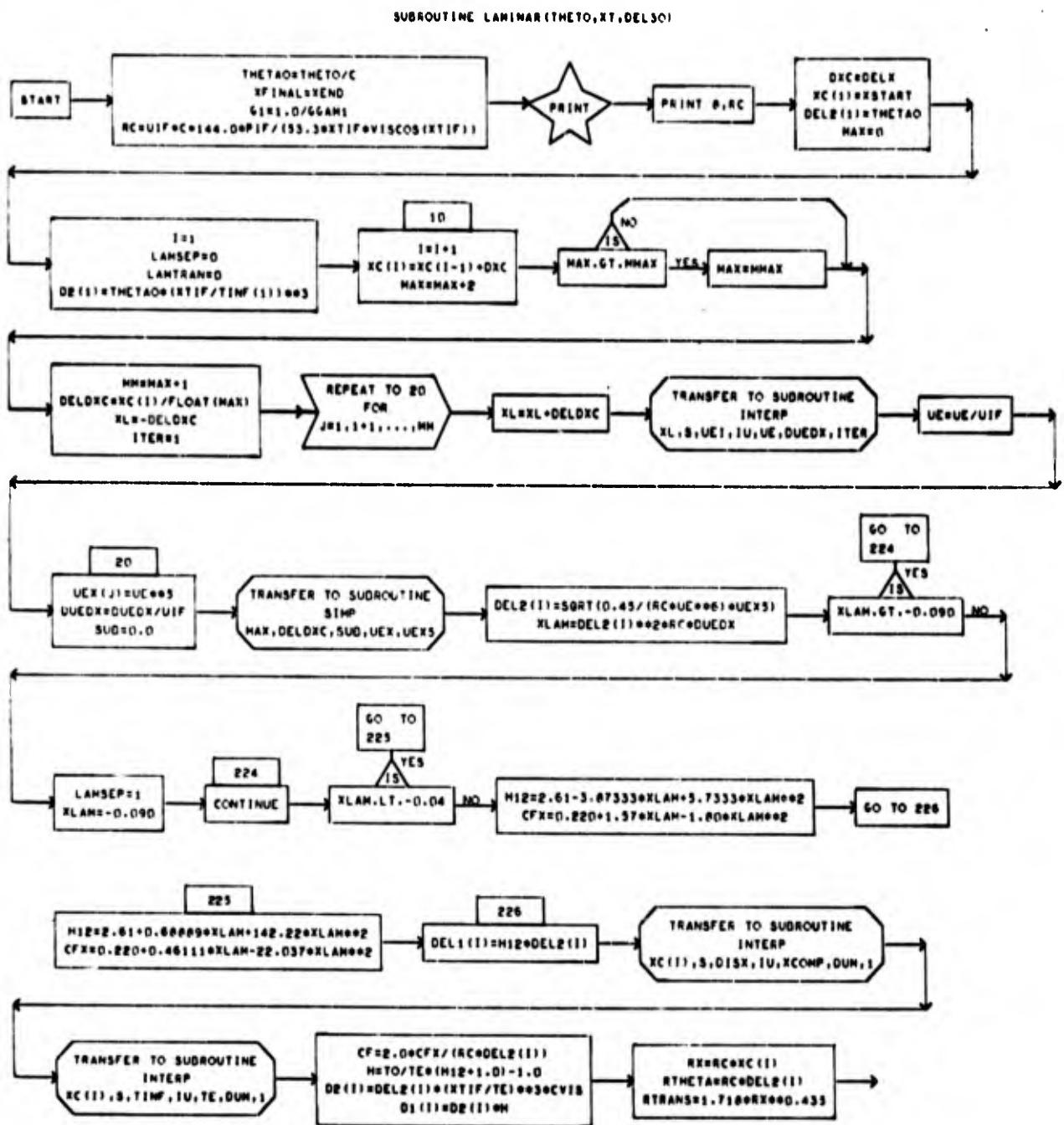
Detailed Flow Charts of Subroutine TURBL (5 of 7)



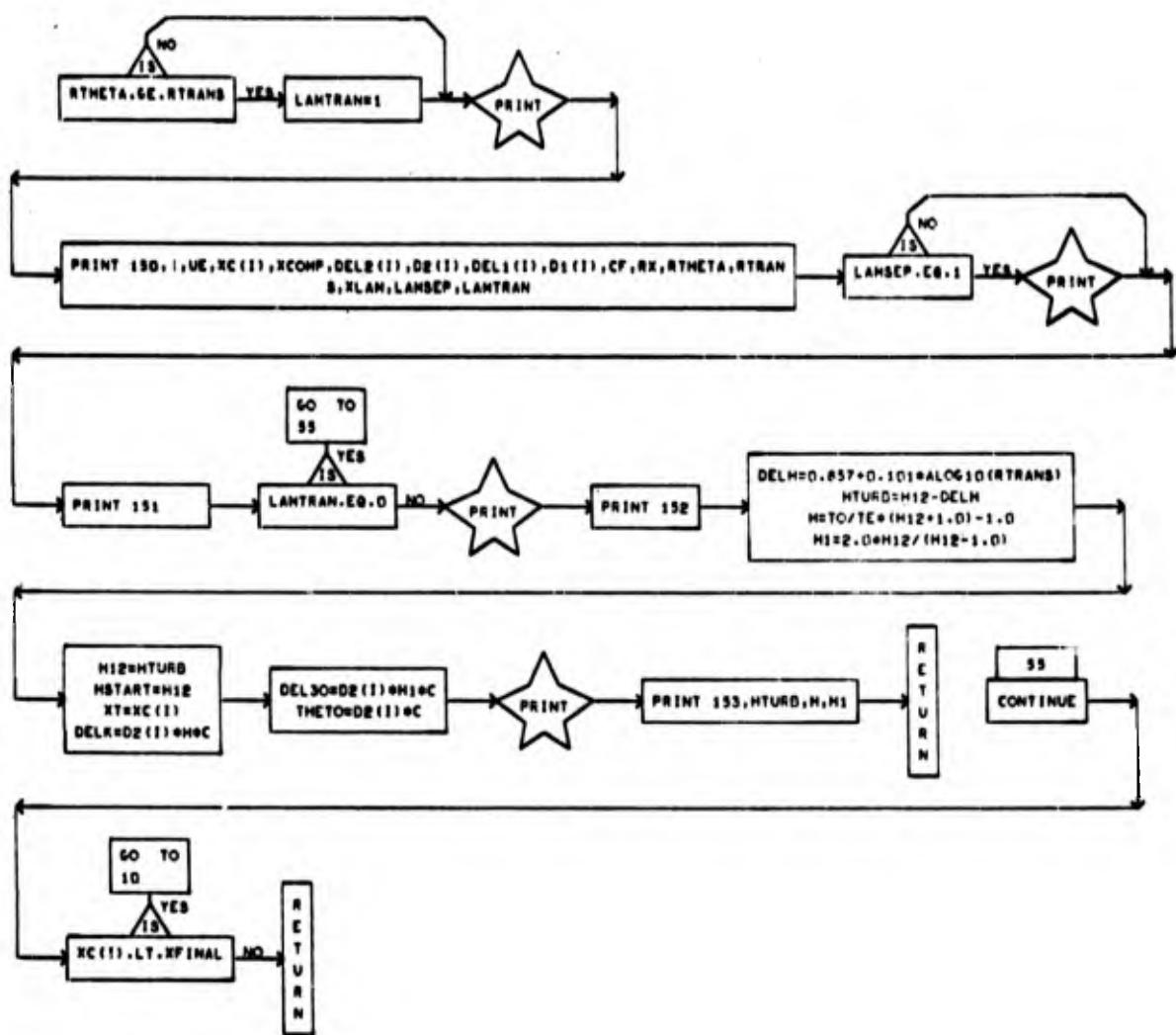
Detailed Flow Charts of Subroutine TURBL (6 of 7)



Detailed Flow Charts of Subroutine TURBL (7 of 7)

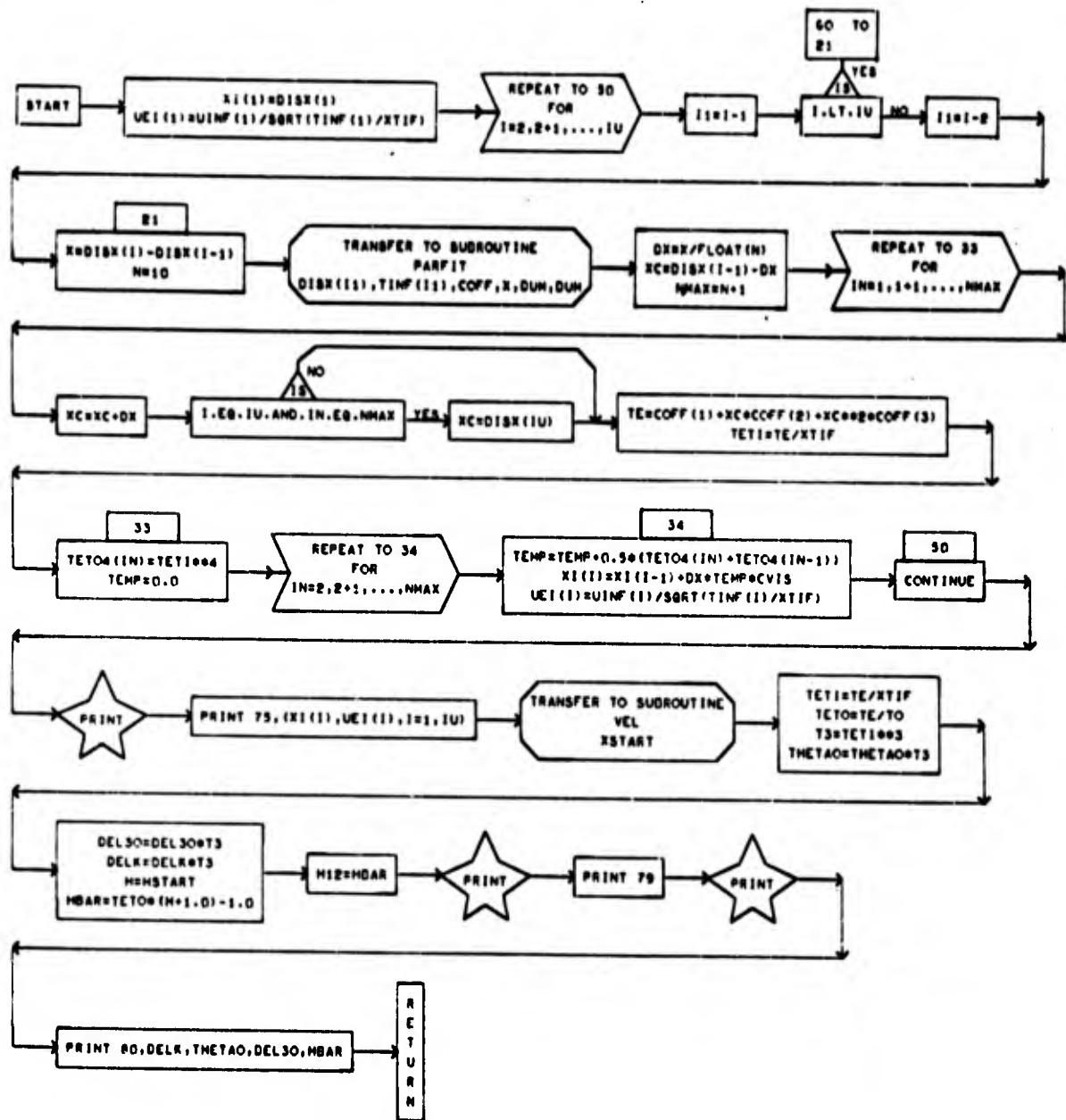


Detailed Flow Charts of Subroutine LAMINAR (1 of 2)



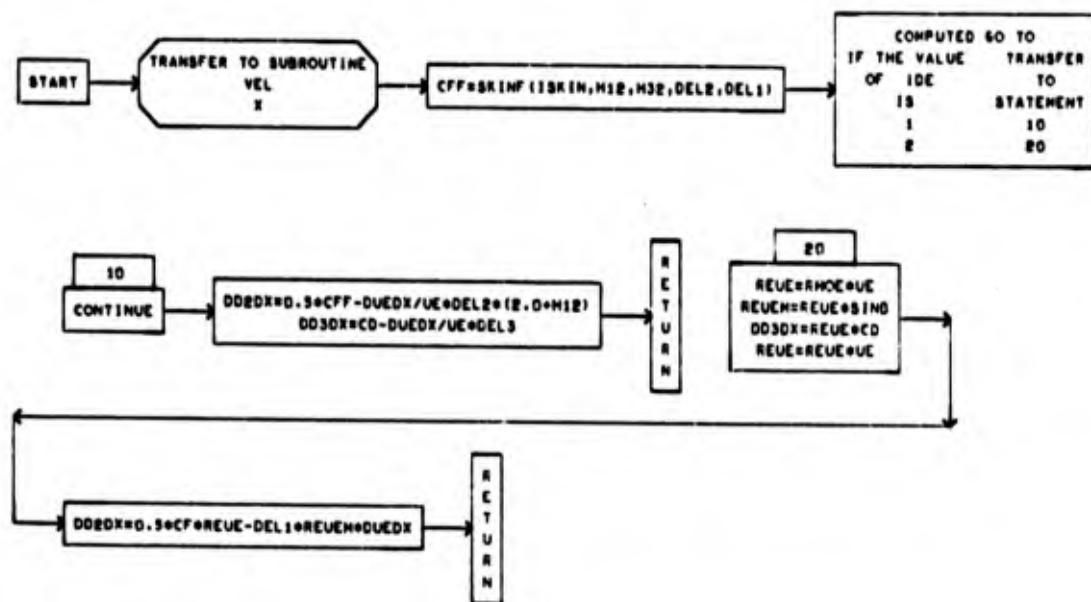
Detailed Flow Charts of Subroutine LAMINAR (2 of 2)

SUBROUTINE TRANS(DELX, THETA0, DEL30, MSTART, ISTART)

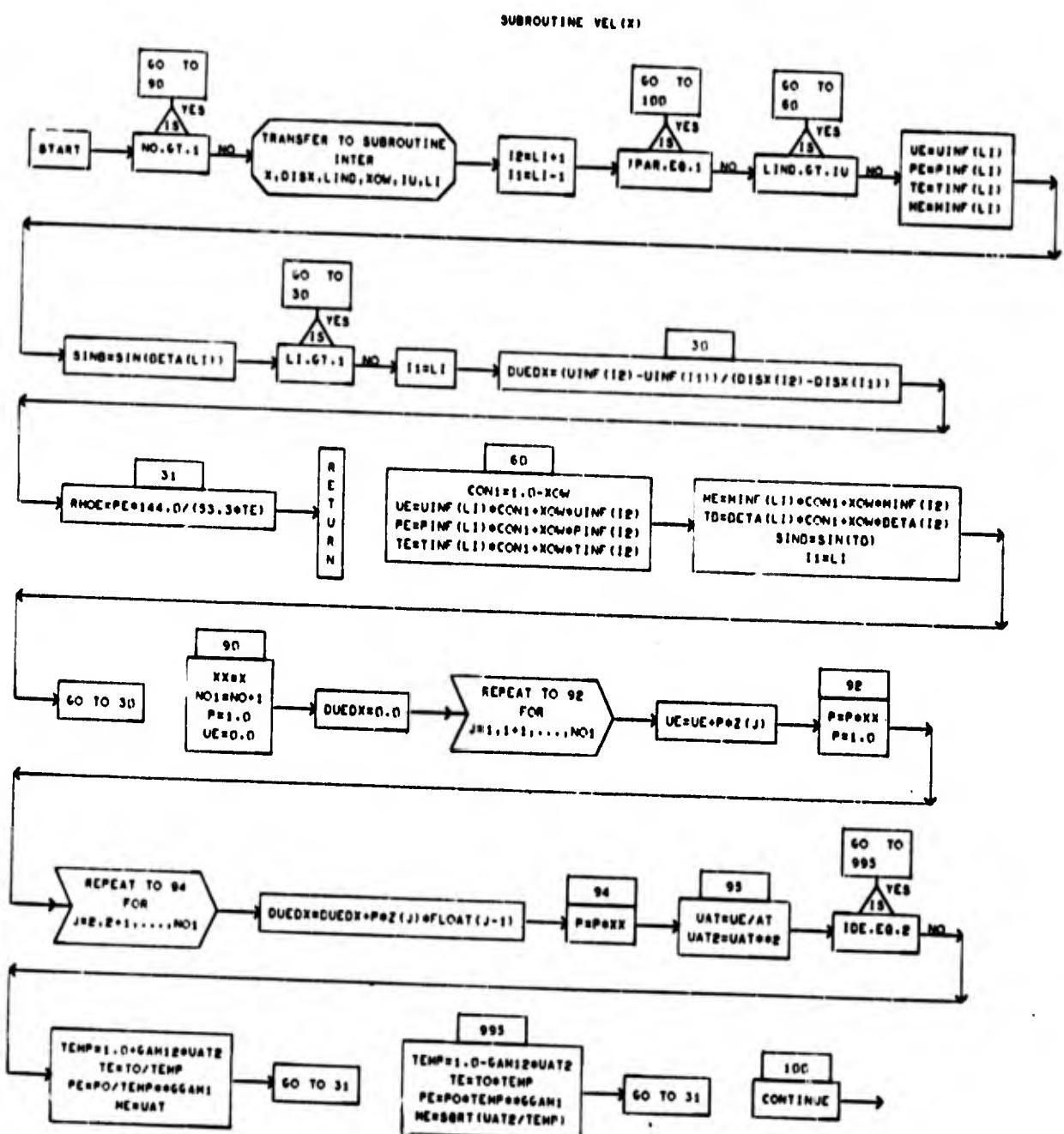


Detailed Flow Chart of Subroutine TRANS

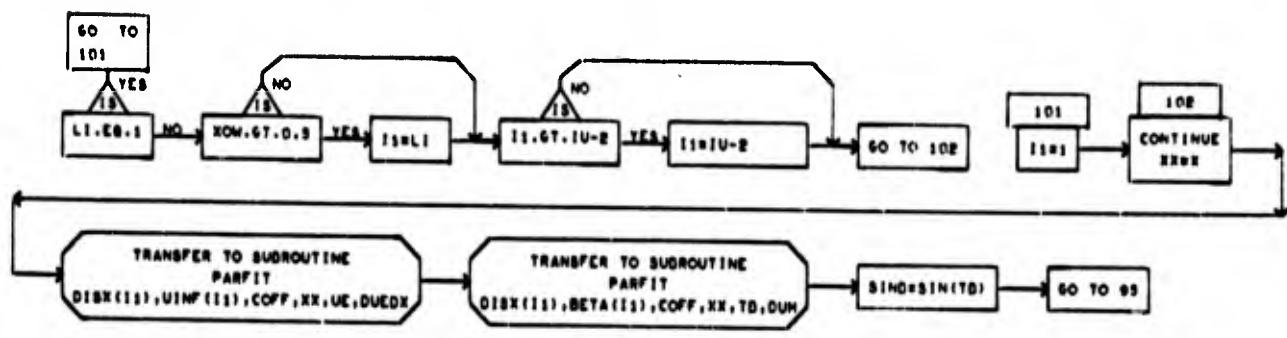
SUBROUTINE SLOPED(X,M12,M32,DEL1,DEL2,DEL3,DD2DX,DD3DX)



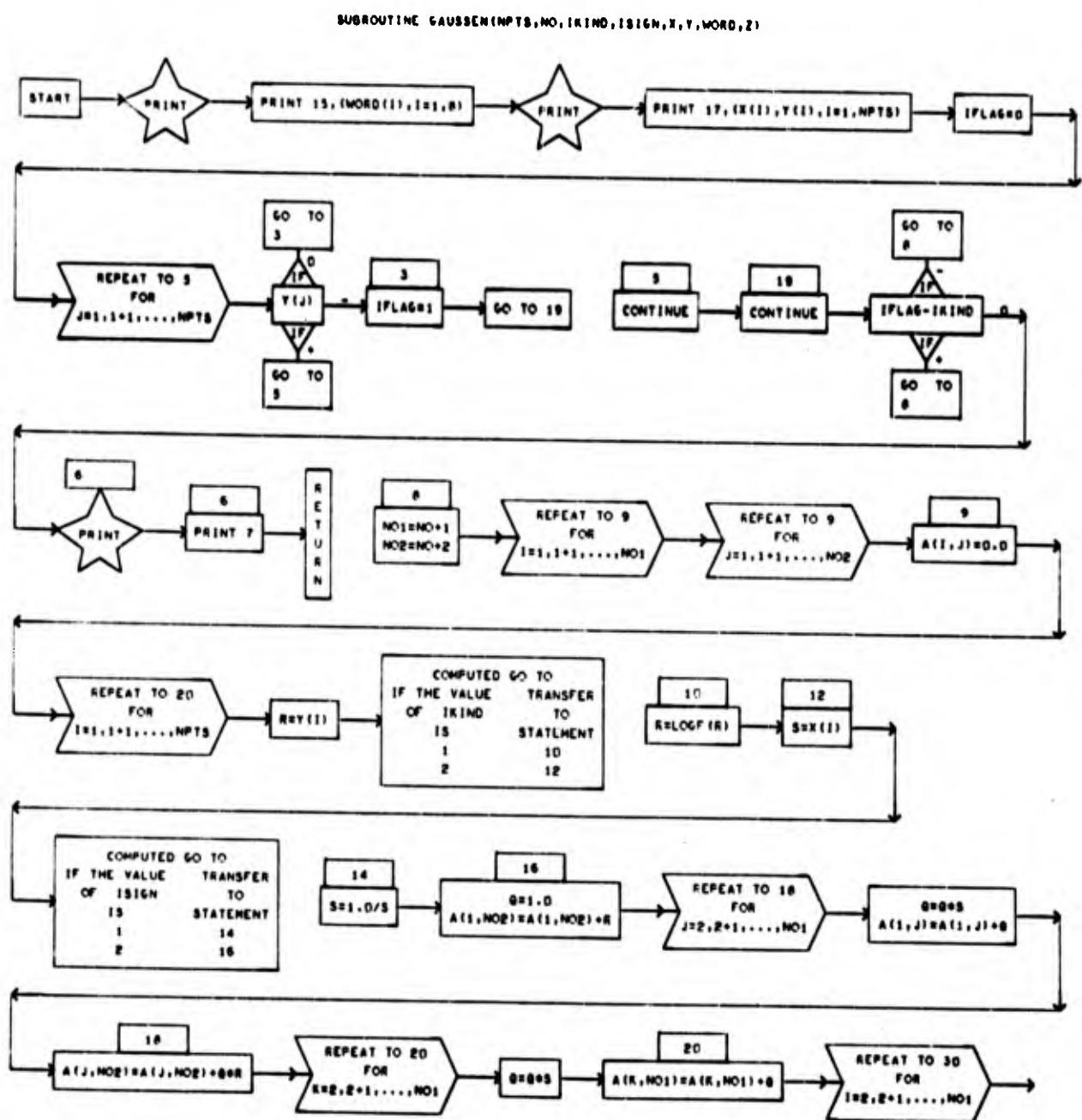
Detailed Flow Chart of Subroutine SLOPED



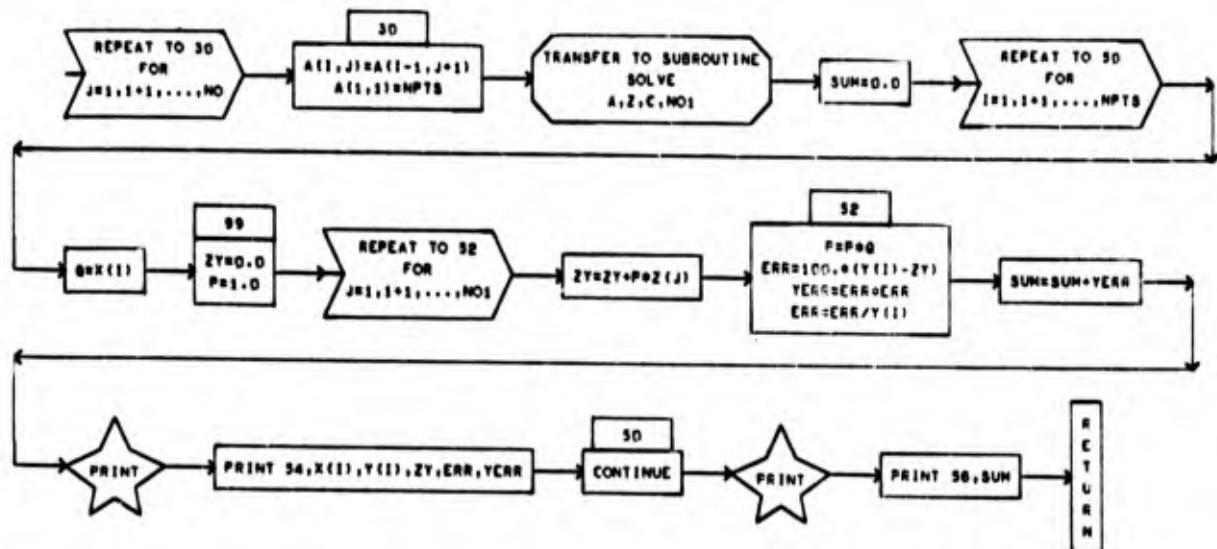
Detailed Flow Charts of Subroutine VEL (1 of 2)



Detailed Flow Charts of Subroutine VEL (2 of 2)

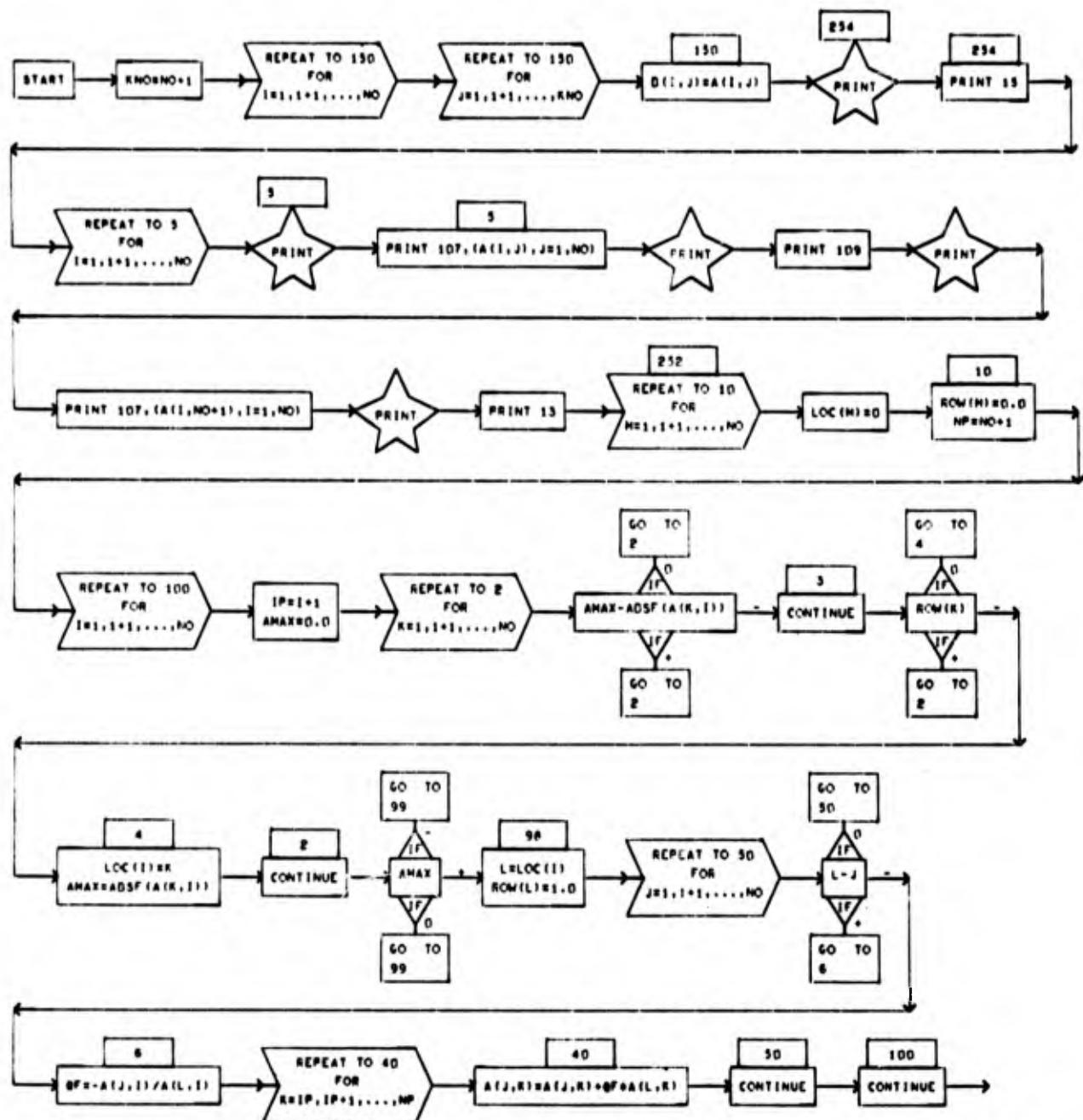


Detailed Flow Charts of Subroutine GAUSSEN (1 of 2)

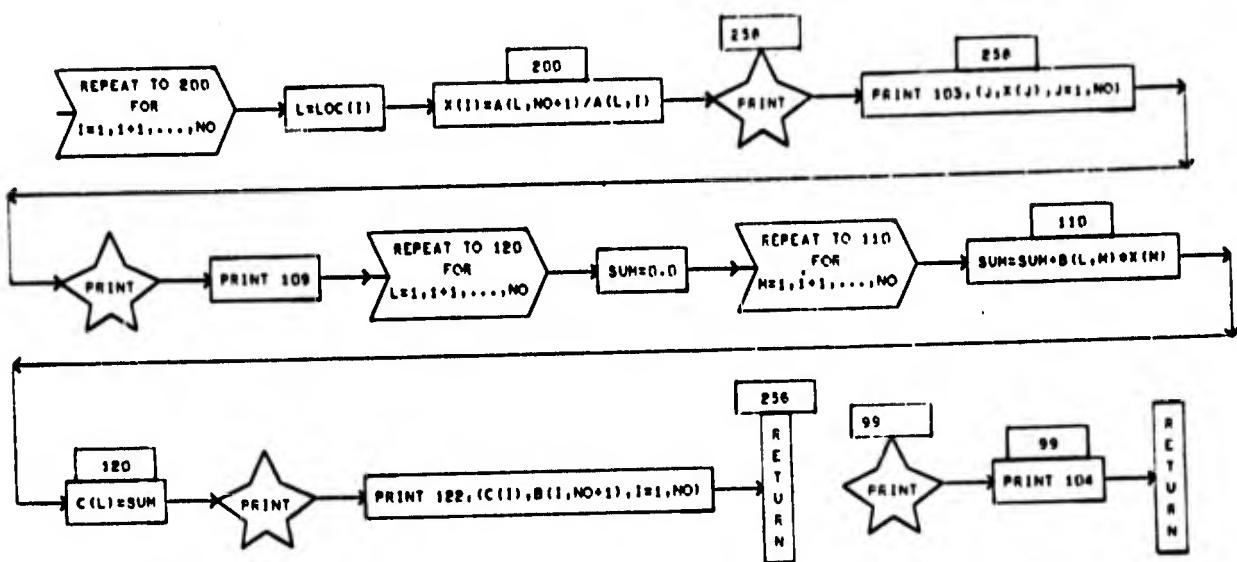


Detailed Flow Charts of Subroutine GAUSSEN (2 of 2)

SUBROUTINE SOLVE (A, X, C, NO)

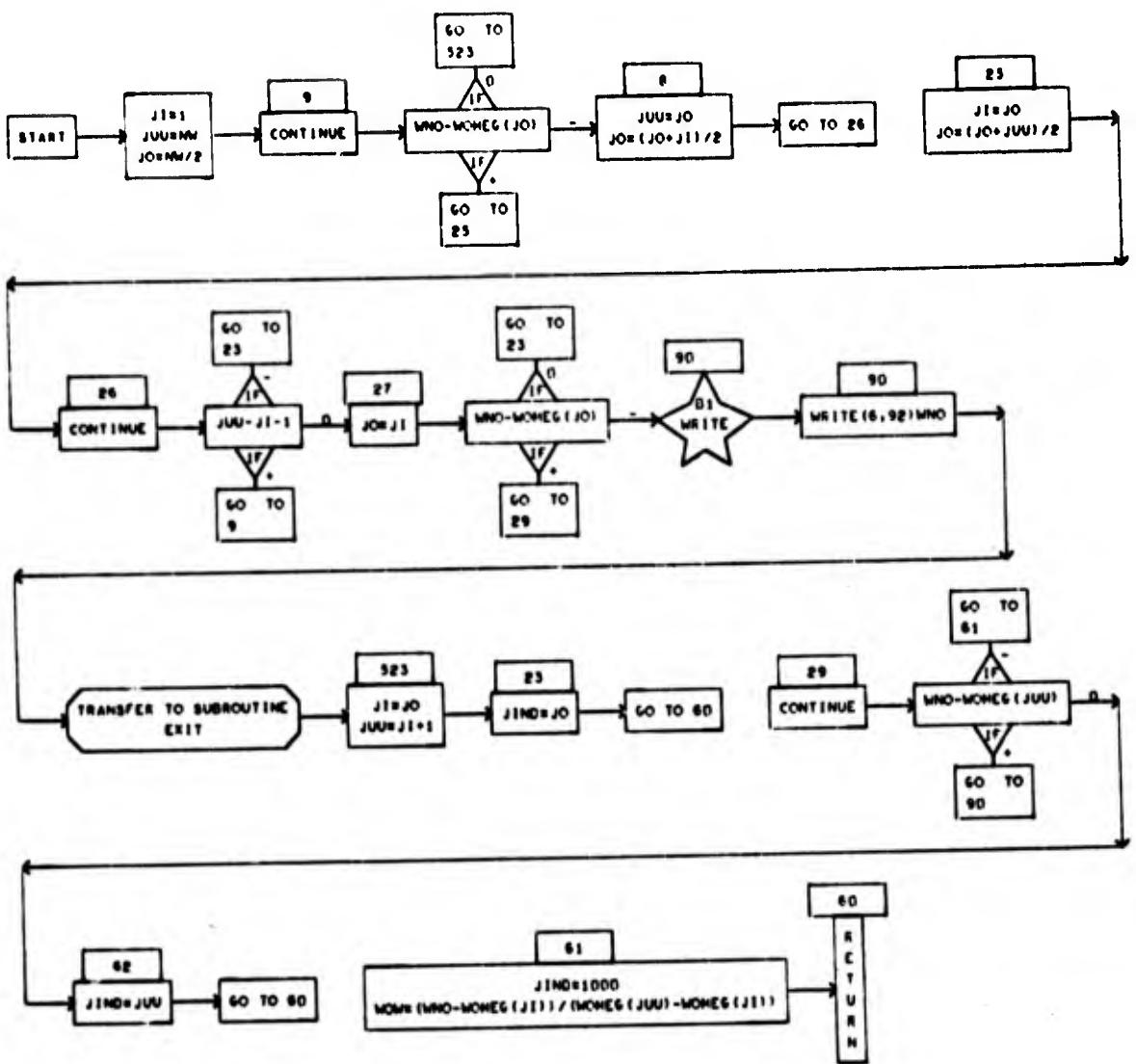


Detailed Flow Charts of Subroutine SOLVE (1 of 2)

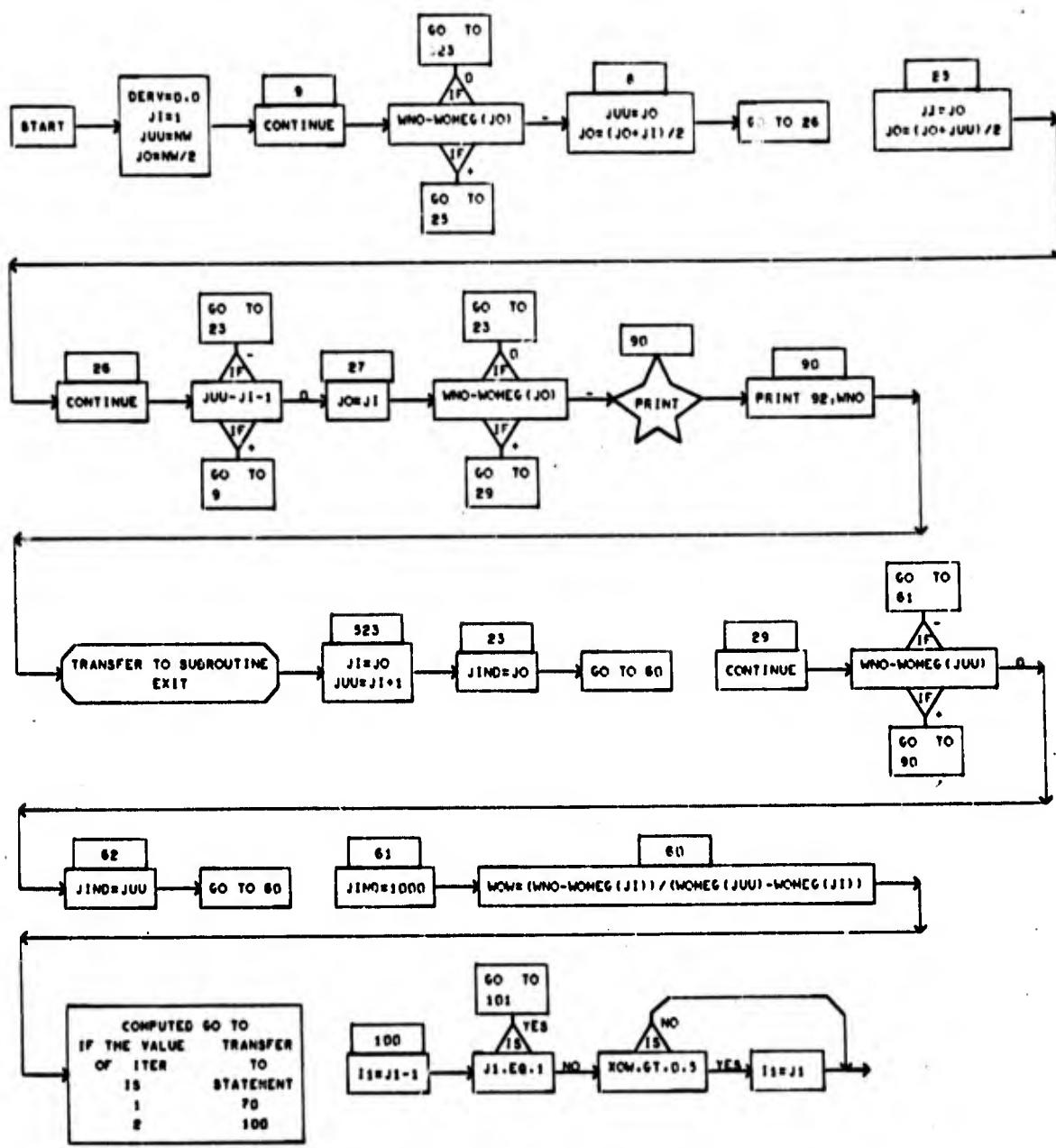


Detailed Flow Charts of Subroutine SOLVE (2 of 2)

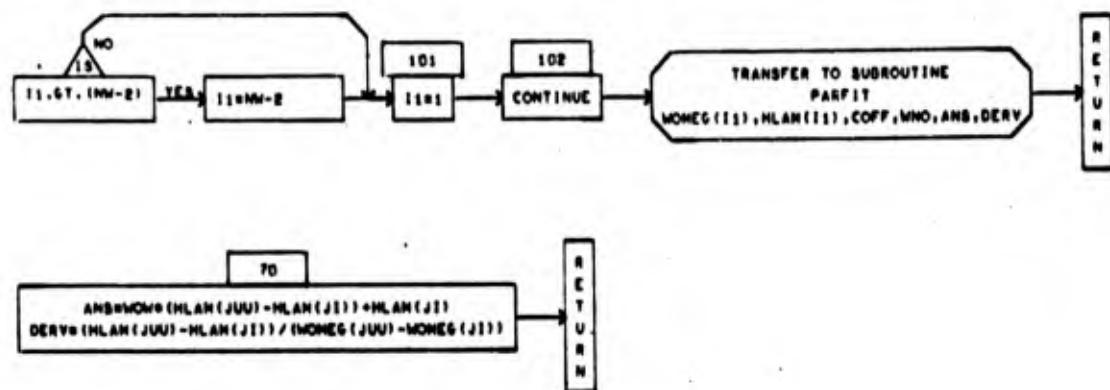
SUBROUTINE INTER (MNO,WMEG,JIND,WV,NW,J1)



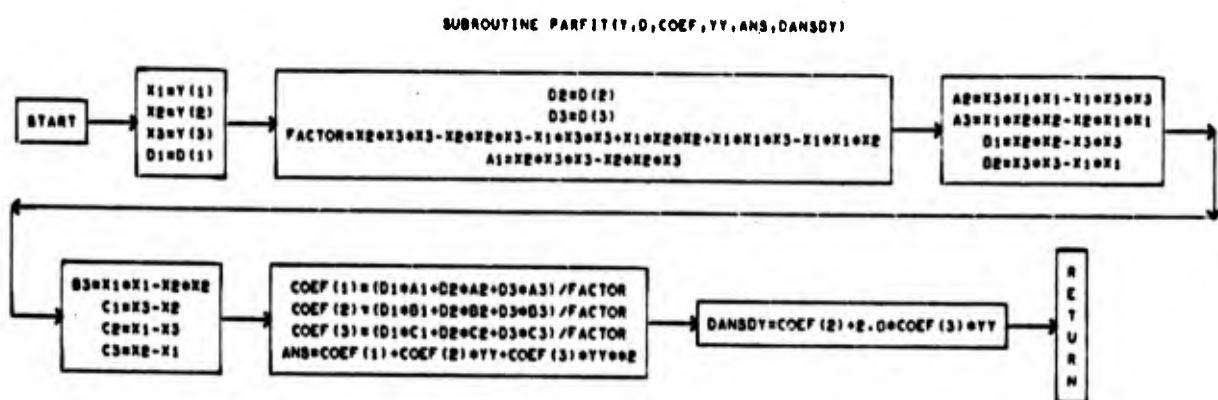
Detailed Flow Charts of Subroutine INTER



Detailed Flow Charts of Subroutine INTERP (1 of 2)

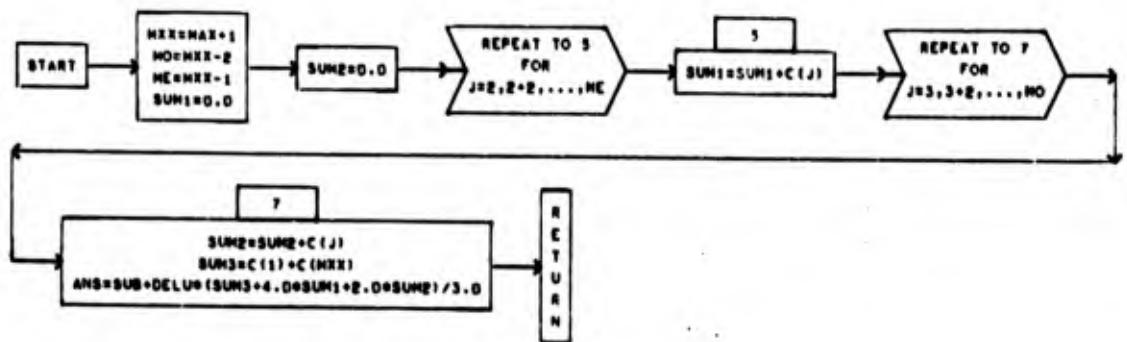


Detailed Flow Charts of Subroutine INTERP (2 of 2)

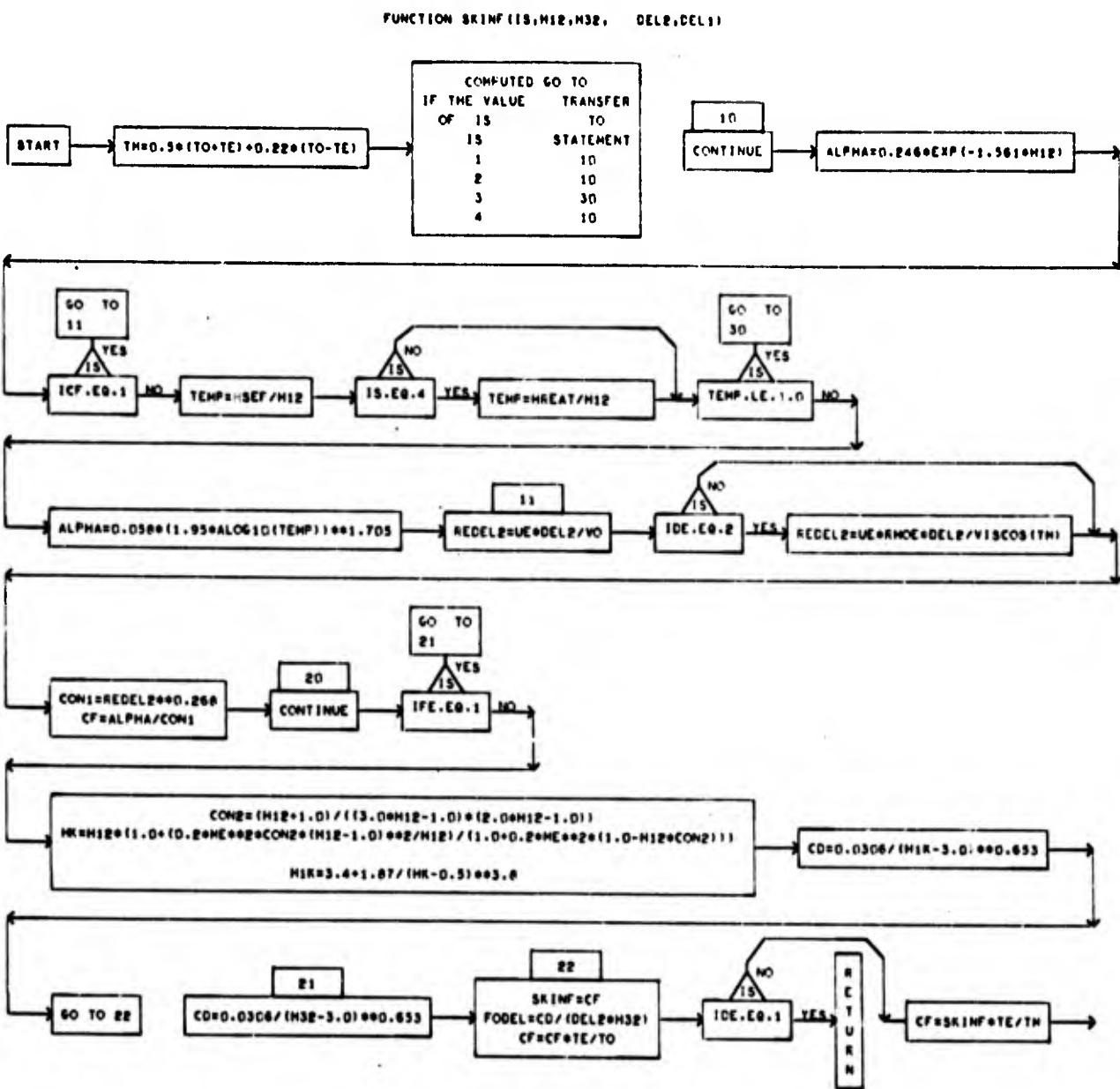


Detailed Flow Chart of Subroutine PARFIT

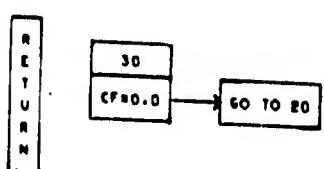
SUBROUTINE SIMP(MAX,DELU,SUB,C,ANS)



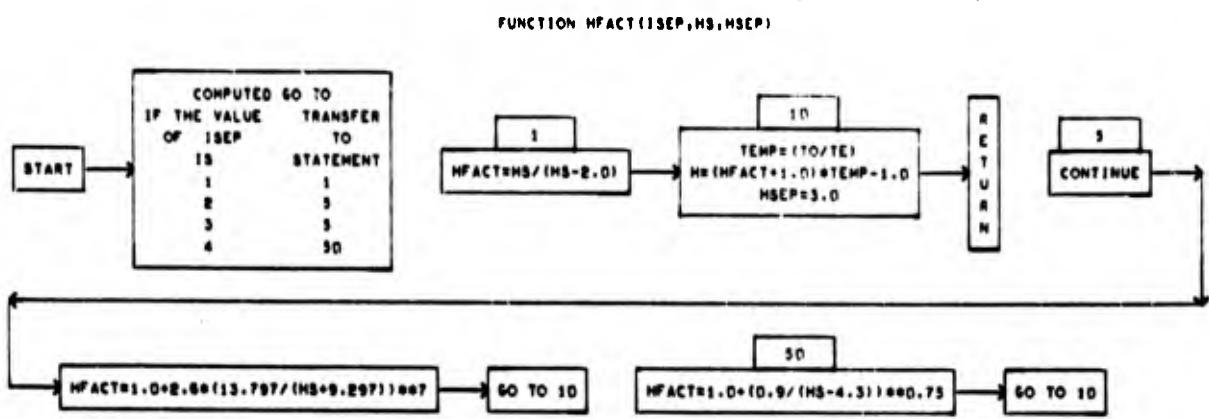
Detailed Flow Chart of Subroutine SIMP



Detailed Flow Charts of Function SKINF (1 of 2)

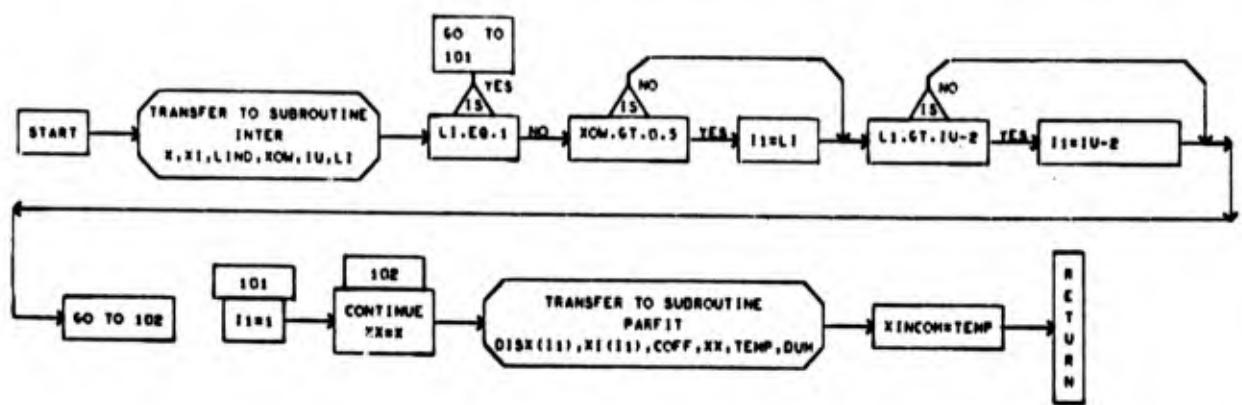


Detailed Flow Charts of Function SKINF (2 of 2)

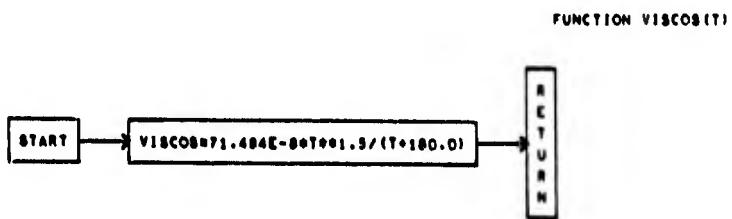


Detailed Flow Charts of Function HFACT

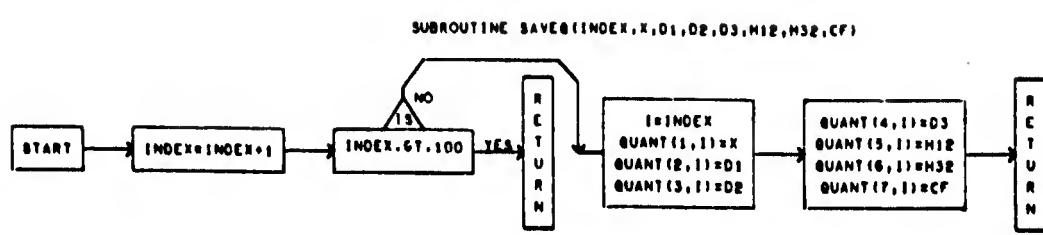
FUNCTION XINCOM(X)



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, McGraw 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 über 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 March 1970	7a. TOTAL NO. OF PAGES 349	7b. NO. OF REFS 11
8a. CONTRACT OR GRANT NO F33615-69-C-1180	9a. ORIGINATOR'S REPORT NUMBER(S) AFFDL-TR-70-16 Volume II	
b. PROJECT NO 1366	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
c.		
d.		
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		

Security Classification

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