AFFDL-TR-73-21-Vol. III
Book 2

STOL TACTICAL AIRCRAFT INVESTIGATION

VOLUME III ✦ PERFORMANCE GROUND RULES AND METHODS

Book 2 ✦ Takeoff and Landing Digital Computer Program

J. Hebert, Jr.
C. A. Whitney

Convair Aerospace Division of
General Dynamics Corporation

FOREWORD

The Takeoff and Landing Digital Computer Program was prepared by the Convair Aerospace Division of General Dynamics Corporation under USAF Contract F33615-71-C-1754, Project 643A, "STOL Tactical Aircraft Investigation." This contract was sponsored by the Prototype Division of the Air Force Flight Dynamics Laboratory. The USAF Project Engineer was G. Oates (PT) and the Convair Aerospace Program Manager was J. Hebert. C. A. Whitney was the principal contributor.

The research reported was conducted during the period from 7 June 1971 through 31 January 1973. This report was submitted by the author on 31 January 1973 under contractor report number GDCA-DHG73-001.

This report has been reviewed and is approved.

E. J. CROSS, JR.
Lt. Col. USAF
Chief, Prototype Division

ABSTRACT

The MILSTOL (MILitary STOL) takeoff and landing digital computer program was developed under USAF Contract F33615-71-C-1754, "STOL Tactical Aircraft Investigation," to compute takeoff and landing characteristics of powered-lift STOL aircraft. It calculates a point mass takeoff and/or landing for a trimmed configuration with either externally blown jet flaps, internally blown jet flaps, or mechanical flaps with vectored thrust within the constraints set forth in Reference 1. Contained in this report are:

1. Discussion of assumptions and methods used in the trajectory calculations.
2. Definition of common list variables.
3. Definition of the input variables and sample input data for the externally blown jet flap configuration.
4. Sample output for the externally blown flap configuration.
5. Program listings and flow charts.

The program is written in Fortran IV for use on CDC 6000 series digital computers and requires 37 K_8 central memory for loading and execution. The program is compatable with both the CDC RUN and FTN compiler systems.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION 1

INTRODUCTION

The military STOL takeoff and landing digital computer program (MILSTOL - Convair Aerospace Division, San Diego Operation, scientific computer program P5592) was developed to calculate the takeoff and landing performance of powered-lift STOL aircraft. The performance calculations are made using exact two-degree-of-freedom equations of motion for a point mass aircraft (i.e., no pitch dynamics). The program was developed from the takeoff portion of the Aircraft Performance Analysis System, Reference 2, and uses data handling, equations of motion, and general use subroutines from that program.

The takeoff portion of the MILSTOL program performs a constant-weight "balanced" takeoff from zero forward speed to liftoff and to stop, within the constraints shown in Figure 1-1, for a matrix of gross weights and runway altitudes. Velocity cues for engine failure, rotation, and liftoff are factors times the minimum control speed and the stall speed with power on and the critical engine failed.

The landing phase of this program performs a "no-flare" style landing approach, touchdown, and deceleration to stop for the input matrix of gross weights and runway altitudes. Velocity cues for approach and touchdown speeds are functions of the minimum control speed and power-on stall speed with the critical engine failed. The landing trajectory is calculated within the constraints shown in Figure 1-2. Aero-

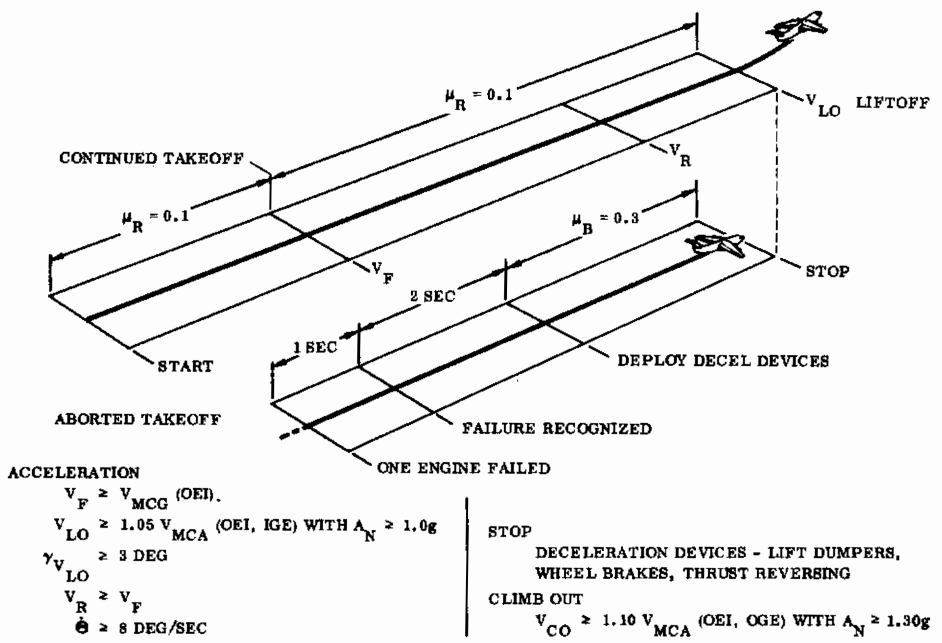


Figure 1-1. Balanced Field Takeoff Ground Rules for STAI.

dynamic and propulsion data is input in tabular form and is handled by individual modularized subroutines. The atmospheric properties subroutine is compatible with the 1962 U. S. Standard Atmosphere and the MIL-STD-210A temperature conditions.

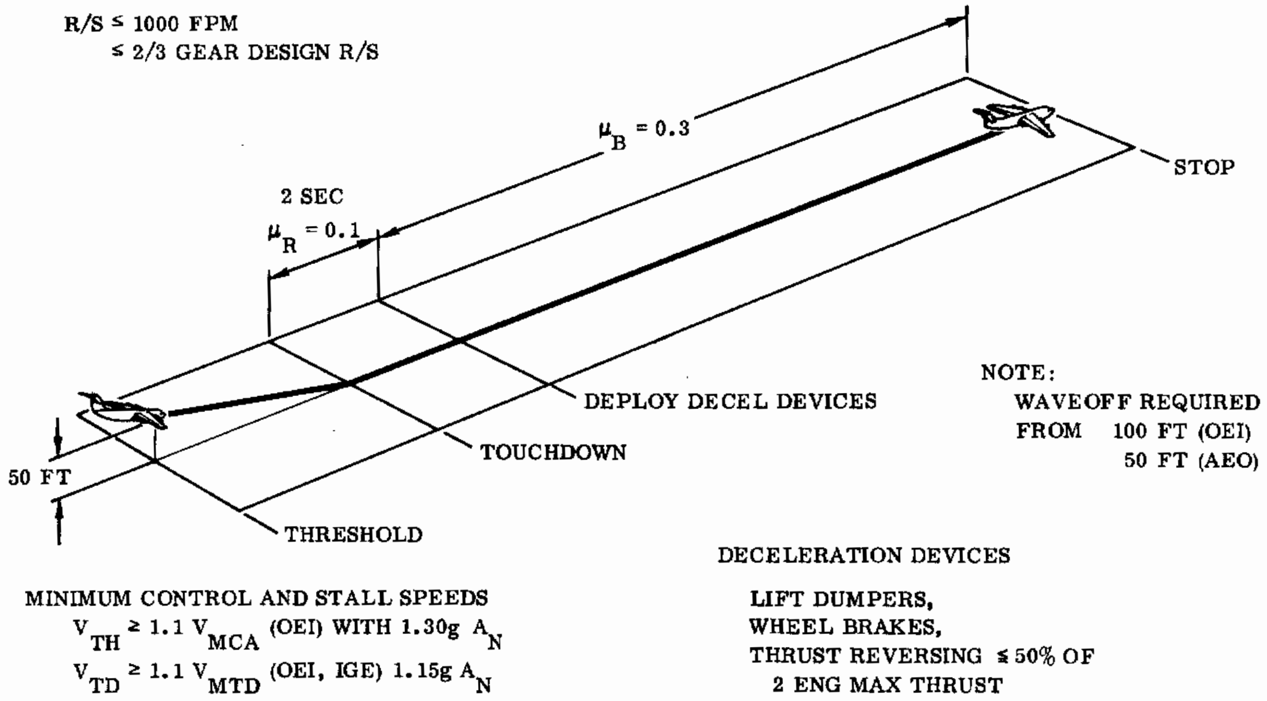


Figure 1-2. Landing Ground Rules for STAI.

SECTION 2

PROGRAM DISCUSSION

The subroutines in the MILSTOL program are classified into five functional categories.

1. Executive Program

2. Maneuver Driving Subroutines

3. Physical Data Subroutines

4. General Use Subroutines

5. Data Handling Subroutines

Program flow and structure are shown in Figure 2-1; each program subroutine is discussed by functional category in the following sections.

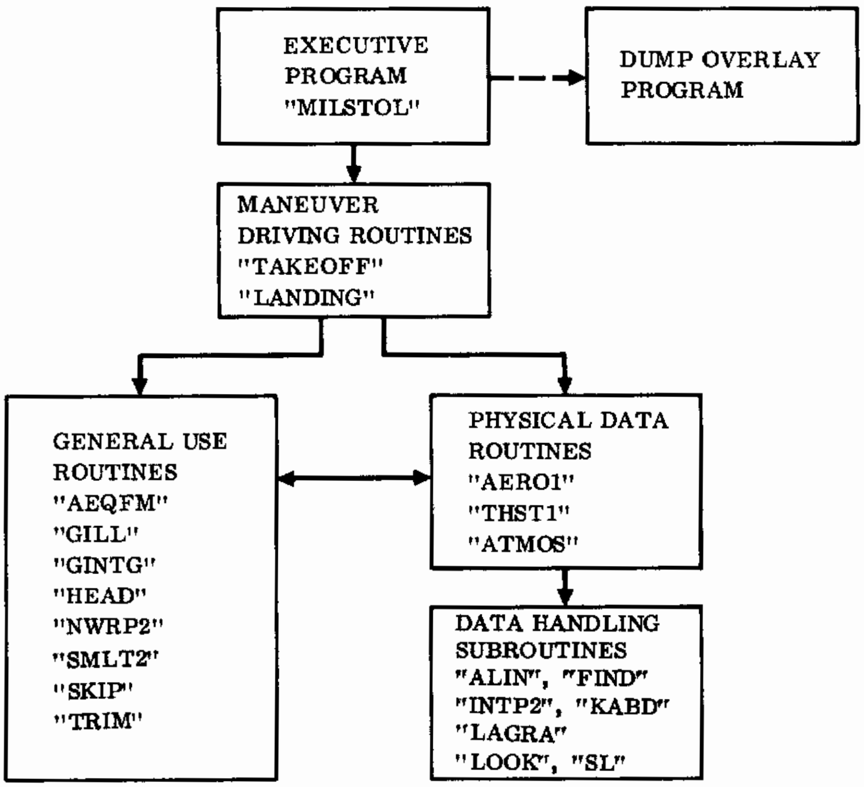


Figure 2-1. MILSTOL Functional Structure.

2.1 EXECUTIVE PROGRAM (MILSTOL)

The Executive Program controls the reading of inputs, the initializing of weights, altitudes, and temperature, and execution of the appropriate trajectory subroutine for the given weight and altitude matrix. In addition, a Dump Overlay program is included in the MILSTOL procedure. This Dump Overlay program is executed only in the event of a fatal error. At the time of the abnormal termination, the Dump Overlay program is loaded from a local file and prints a listing of all common list variables in Namelist format.

2.2 MANEUVER DRIVING SUBROUTINES

Two maneuver subroutines, TAKEOFF and LANDING, are incorporated in the MILSTOL program.

2.2.1 TAKEOFF TRAJECTORY SUBROUTINE (TAKEOFF) — The TAKEOFF subroutine is the driver for all portions of the takeoff maneuver. Ground rules and constraints for this maneuver are shown in Figure 1-1. This subroutine reads trajectory-related variables (e.g., rolling and braking coefficients of friction, time delays, minimum control speeds, etc.) executes the takeoff calculation procedure, adjusts the engine failure speed to balance the continued and aborted takeoffs, and causes the takeoff time history to be output.

Because of the balanced-type takeoff specified by the ground rules and to ensure efficient program operation, the takeoff is not calculated as a continuous function of time. Instead, the program is divided into segments. The sequence of calculations is:

1. Stall speed with power on and the critical engine failed is calculated. Liftoff speed is set using the air minimum control speed and stall speed. The initial value of engine failure speed is set equal to ground minimum control speed.

2. Angle of attack for liftoff is calculated with the critical engine failed. If the ground contact angle (the angle for the tail striking the ground during rotation) is exceeded, liftoff speed is increased by one percent of the air minimum control speed or power-on stall speed. At this point, the pertinent aircraft conditions are output. When the conditions for liftoff are established, maximum rate of climb at the liftoff speed is calculated and output.

3. The first segment of the takeoff is calculated with the critical engine failed by integrating time and tangential acceleration, along with the input rotation rate in a negative sense from liftoff speed to rotation speed to obtain velocity,

distance, and aircraft attitude. The integration is terminated when the aircraft pitch attitude is zero, and a check is made to ensure that the rotation velocity is greater than the selected engine failure speed and the ground minimum control speed. If this criterion is not satisfied, the liftoff speed is increased using an empirical relationship. The program then returns to Step 2 and continues until the rotation velocity criterion is satisfied.

4. If rotation velocity is greater than engine failure speed, time and tangential acceleration are integrated, in a negative sense from the rotation velocity to the engine failure speed, with the critical engine failed to obtain velocity and distance. This distance is the second takeoff segment and, when added to the first segment from Step 3, is the "continued takeoff" distance used in the field-balancing relationship.

5. The distance for the "aborted takeoff" used in balancing the field length is calculated by integrating, in a positive sense, the time and tangential acceleration variables from engine failure speed to stop for velocity and distance with the critical engine failed. During the integration, engines are set to idle after the reaction time and the deceleration devices (brakes, lift dumpers, and reverse thrust) are deployed at the end of an actuation time interval.

6. The "continued takeoff" distance, Steps 3 and 4, and the "aborted takeoff" distance are then used in a linear convergence procedure to adjust the engine failure speed so that these distances are equal. After the new engine-failure speed is selected, the program returns to Step 2 and calculates new "continued" and "aborted" takeoff distances. If the "aborted takeoff" distance is greater than or equal to the "continued takeoff" distance and the engine failure speed is equal to the ground minimum control speed, the takeoff is by definition balanced and the program continues.

7. After the preceding steps have balanced the takeoff distance, time and tangential acceleration are integrated from start to the engine failure speed (with all engines operating) for velocity and distance. Summations of the distances from Steps 3 and 4 with this distance and the distance from Step 5 comprises the balanced takeoff distance for this configuration.

2.2.2 LANDING TRAJECTORY SUBROUTINE (LANDING) — The LANDING subroutine is the driver for all portions of the landing maneuver. Ground rules and constraints for the landing trajectory are shown in Figure 1-2. This subroutine reads trajectory-related inputs, executes the landing calculation procedure, and outputs a landing trajectory time history for the input configuration.

The sequence of calculations is:

1. Stall speed with power on and the critical engine failed is calculated. The approach speed that conforms to the ground rules of Figure 1-2 is set using the air minimum control speed and the power-on stall speed.

2. Angle of attack at touchdown is calculated with all engines operating. If the pitch attitude exceeds the ground contact angle, the approach speed is increased by one percent of the power-on stall speed. If the rate of sink at touchdown exceeds the input maximum, the glideslope angle is reduced so that the rate of sink limit is met. If the configuration attitude is such that the nosewheel hits first, the program prints an error message and returns control to the executive routine. When all touchdown criteria are satisfied, the program outputs the conditions at touchdown and calculates the maximum rate of climb available at touchdown speed with the critical engine failed.

3. The program then calculates angle of attack at the obstacle (with all engines operating). This calculation is performed at the approach speed calculated in Step 2. Aircraft conditions at the obstacle are output along with the maximum rate of climb available with the critical engine failed. This calculation is made at the obstacle to account for ground effects.

4. Landing air distance is calculated by performing a one-step integration using the velocities from Steps 2 and 3.

5. After touchdown, the program performs a step-wise integration from touchdown to stop to calculate ground distance. During this integration, engines are set to idle power, aircraft attitude is rotated down to zero, and deceleration devices (brakes, lift dumpers, and reverse thrust) are deployed, after allowing for actuation time delay.

6. Total landing distance is the summation of distances from Steps 4 and 5.

2.3 PHYSICAL DATA SUBROUTINES

ATMOS, AEROI, and THSTI are the three physical data subroutines in the MILSTOL program. These subroutines read inputs and store, retrieve, and calculate the atmospheric, aerodynamic, and propulsion characteristics required to solve the equations of motion.

2.3.1 ATMOSPHERIC PROPERTIES SUBROUTINE (ATMOS) — The ATMOS subroutine supplies the program with ambient temperature, pressure, density ratio, and speed of sound as a function of altitude and type of day or an input temperature. The temperature/type of day options available are:

1. U.S. Standard Atmosphere, 1962.

2. MIL-STD-210A Tropic Day.

3. MIL-STD-210A Polar Day.

4. MIL-STD-210A Hot Day.

5. MIL-STD-210A Cold Day.

6. An input temperature in °F.

These options all use the standard day pressure altitude relationship in the calculation procedure. Options 1 through 5 use the appropriate temperatures from References 3 and 4.

2.3.2 AERODYNAMIC DATA SUBROUTINE (AERO1) — This subroutine was developed to store and retrieve trimmed aerodynamic data for configurations with externally blown jet flaps, internally blown jet flaps, and mechanical flaps with vectored thrust. Conventional configurations without thrust augmented lift can be used by either modifying the table lookup procedure or by entering the power-off data for four dummy thrust coefficients and using the mechanical flap plus vectored thrust option.

AERO1 has two entries: AERO1, which retrieves maximum lift characteristics as a function of flap deflection and momentum coefficient, and AERO2, which retrieves lift and drag data as a function of flap deflection, angle of attack, and momentum coefficient. The inputs to this subroutine are geometric data, configuration type, and aerodynamic data tables.

Because of the differences in methods for estimating aerodynamic data for each of the three configurations, there is a unique method of storing and retrieving the data for each of the three. The externally blown jet flap data includes all direct and indirect thrust effects. The internally blown jet flap data includes all thrust effects due to trailing edge slot blowing but none of the thrust effects due to the cruise engines. The mechanical flap plus vectored thrust sequence assumes that the aerodynamic data includes all indirect thrust effects including any supercirculation effects, but none of the direct thrust vector effects.

All additional items that degrade the aerodynamic data (e.g., lift dumpers, engine out corrections, etc.) are cued from the trajectory subroutines and are included in the final lift and drag values before returning to the calling subroutine.

Retrieval and calculation of the aerodynamic characteristics with the critical engine failed are handled in the same manner for all configurations. In this calculation scheme, it is assumed that the input aerodynamic data tables are valid for either the all-engines-operating or the one-engine-failed condition if a later correction is made to compensate for engine-out moments. The correction for trimming of engine-out moments are input increments to lift and drag. Configuration design and aerodynamic conditions that allow this assumption are:

1. All internally blown jet flap configurations are assumed to have cross-ducting so that the spanwise distribution of blowing is always symmetric.

2. Engine-out degradation in the mechanical flap plus vectored thrust configurations is due to the loss of the thrust vector component, not loss of aerodynamic lift.

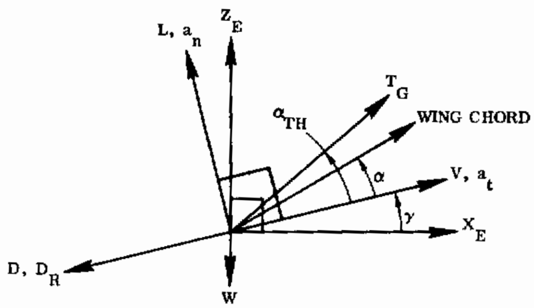
3. For configurations with externally blown jet flaps, the aerodynamic characteristics are a function of total thrust coefficient. Analysis of the data in Reference 5 shows that aerodynamic lift and drag characteristics are the same for a given total thrust coefficient, whether or not an engine is failed, if the moments due to the engine failure are not trimmed. The lift and drag increments due to trimming the engine-out rolling and yawing moments are incorporated as stated above.

2.3.3 PROPULSION CHARACTERISTICS SUBROUTINE (THST1) — The THST1 subroutine is used for storing and retrieving single-altitude propulsion characteristics for all three powered lift configurations. Inputs to this routine are in the form of single-altitude maximum gross thrust, ram drag at maximum thrust, idle gross thrust, maximum reverse thrust, windmilling drag of a single engine, and maximum gross thrust at the slot exit for internally blown jet flap configurations as a function of velocity. Procedures have been incorporated so that reverse thrust is always symmetric in the engine-out case.

2.4 GENERAL USE SUBROUTINES

There are eight general use subroutines in the MILSTOL program. These are general purpose subroutines extracted from Reference 2 and have applications outside of the MILSTOL program. These subroutines perform integrations, print page headings, handle equations of motion, and provide the logic for convergence procedures. Each of these subroutines is described in alphabetical order in the following paragraphs.

2.4.1 AEQFM — To maintain consistency between different programs and calculations, all equation-of-motion calculations are performed using the AEQFM subroutine. Figure 2-2 shows the axis systems, forces, and angles used in the two-degree-of-freedom calculations. Equations 1 and 2 are balanced for the appropriate



(X_E, Z_E) = EARTH AXIS SYSTEM
T_g = GROSS THRUST
V = AIRCRAFT VELOCITY
a_t = TANGENTIAL ACCELERATION
γ = FLIGHT PATH ANGLE
D = DRAG
DR = RAM DRAG
L = LIFT
α = ANGLE-OF-ATTACK
α_{TH} = ANGLE BETWEEN T_G AND V
W = WEIGHT
a_n = NORMAL ACCELERATION

$$ma_t = WG_t = T_G \cos(\alpha_{TH}) - (D+DR) - W \sin(\gamma) \quad (1)$$

$$ma_n = W(G_n - 1.) = T_G \sin(\alpha_{TH}) + L - W \cos(\gamma) \quad (2)$$

WHERE

$G_t = a_{t/g}$, $G_n = a_{n/g} + 1.$,

$g = 32.174049$, $m = W/g$

Figure 2-2. Force and Angle System Used In the MILSTOL Program.

flight condition. When acceleration or deceleration on the ground is required, an additional term is added to Equation 1 to account for the ground friction force. All accelerations in Equations 1 and 2 are converged simultaneously using the TRIM and SMLT2 subroutines.

2.4.2 GILL — This is an integrating subroutine that uses the method developed by S. Gill (Reference 6), to provide fourth-order accuracy while requiring a minimum number of storage registers. The subroutine requires four passes to accomplish the integrating step:

Pass One. Take derivatives at the start of the interval and predict conditions at the middle of the interval.

Pass Two. Take derivatives based on predicted conditions at the mid-interval and combine with derivatives from first pass to predict conditions at the mid-interval.

Pass Three. Take derivatives based on latest estimate of mid-interval conditions and combine with derivatives from first two passes to predict end-of-interval conditions.

Pass Four. Take derivatives based on end-of-interval conditions and combine with derivatives from other passes to calculate conditions at the end of the interval.

This process is repeated for each integration step. The calling subroutine, GINTG, keeps track of the number of passes and checks for terminations after four passes.

2.4.3 GINTG — This subroutine is the driver for trajectory integrations. It calls the equations-of-motion subroutine (AEQFM) for accelerations, then the GILL subroutine to integrate for velocities, distances, and aircraft attitude.

2.4.4 HEAD SUBROUTINE — This subroutine is used for printing columnar headings in the time history printout.

2.4.5 NWRP2 — This is a Newton-Wrapson iteration subroutine which determines the value of x that will return y equal to zero based on a linear prediction using two previously calculated points.

2.4.6 SKIP — This is an output formatting subroutine that starts a new page, prints a standard page heading, and restarts the line count.

2.4.7 SMLT2 — This subroutine performs a simultaneous equation solution, using derivatives from subroutine TRIM, to obtain increments to the independent variables that will result in the desired accelerations for subroutine AEQFM.

2.4.8 TRIM — This subroutine controls the systematic perturbation of independent variables and stores the variation of each acceleration with respect to each variable as a derivative.

2.5 DATA HANDLING SUBROUTINES

The MILSTOL program uses five subroutines and two functions whose sole purpose is data handling. These routines fit curves to data, evaluate curve fits, and perform table look-ups. These routines and functions are discussed in alphabetical order in the following paragraphs.

2.5.1 ALIN — This function makes a linear fit between two (x,y) points.

2.5.2 FIND — This subroutine performs a one-dimensional linear interpolation within a data table. For arguments greater than or less than the table, a linear extrapolation is performed.

2.5.3 INTP2 — Subroutine INTP2 fits a third-order polynomial through four (x,y) points, returns the coefficients, and returns a y-answer for an x-argument.

2.5.4 KABD — The subroutine evaluates a hyperbolic fit by returning a y-answer for an x-argument of the equation $y = K/(x-A) + B + D\,x$, where coefficients, K, A, B, and D are provided in the calling list.

2.5.5 LAGRA — Subroutine LAGRA returns a y-answer for an x-argument using a Lagrangian interpolation on four (x,y) points.

2.5.6 LOOK — Originally written to handle three-dimensional tabulated thrust and fuel flow data, this subroutine has been developed into a more general form that can handle any three-dimensional tabular data. The LOOK subroutine performs the table lookup using a non-linear technique the basis of which is the LAGRA subroutine. It has four options for locating data and it can also return derivatives with respect to three independent variables using a four-point interpolation of each independent variable.

2.5.7 SL — The SL function calculates the linear slope between two (x,y) points.

SECTION 3

COMMON LIST VARIABLES

This section contains three tables that provide the user with a key to the definition and usage of the labeled common blocks incorporated in the MILSTOL program. Table 3-1 describes each labeled common block. The blocks have been constructed by function to aid the user in future modifications or upgrading.

Table 3-1. Description of Labeled Common Blocks.

LIST 1	Contains variables used for input and output units, carriage control, and page headings.
LIST 2	Contains variables describing forces, velocities, altitudes, and weights.
LIST 3	Contains coefficients, angles, and aerodynamically significant geometry.
LIST 4	Contains variables used for transmitting propulsion characteristics.
LIST 5	Contains variables used for atmospheric properties.
LIST 6	Contains physical constants and conversion factors.
LIST 9	Used for transmitting data from the LOOK subroutine.
LIST 15	Contains integration variables and controls.
LGEOM	Contains aircraft geometry and angles.
CONTROL	Control flags used during takeoffs and landings.
LIST 99	Error index flag.

Table 3-2. Subroutines in Which Labeled Common Blocks Are Used.

	Common List Name											
	LIST 1	LIST 2	LIST 3	LIST 4	LIST 5	LIST 6	LIST 8	LIST 9	LIST 15	LGEOM	CONTROL	LIST 99
MILSTOL (Main Program)	X	X	X	X	X	X	X	X	X	X	X	X
MILSTOL (Dump Program)	X	X	X	X	X	X	X	X	X	X	X	X
AEQFM	X	X	X	X	X	X	X			X		X
AERO1	X	X	X	X	X	X	X	X	X	X	X	X
ATMOS	X	X			X							
FIND	X											
GINTG	X	X	X	X	X	X	X		X	X		
HEAD	X											
KABD	X											
LANDING	X	X	X	X	X	X	X		X	X	X	X
LOOK								X				
SKIP	X	X	X		X							
TAKEOFF	X	X	X	X	X	X	X		X	X	X	X
THSTI	X	X	X	X	X	X				X	X	

Table 3-3. Definition of Common Block Variables.

VARIABLE NAME	COMMON NAME	DEFINITION
ALFAR	LIST2	WING ANGLE OF ATTACK (RADIANS)
ALMXR	LIST3	MAXIMUM ALLOWABLE ANGLE OF ATTACK (RADIANS)
ALPHD	LIST3	WING ANGLE OF ATTACK FOR STALL (RADIANS)
ALTHR	LIST3	ANGLE OF ATTACK OF THE ENGINE AXIS (RADIANS)
AN	LIST8	ACCELERATION NORMAL TO THE VELOCITY VECTOR (FT/SEC SQ)
ANGLE	LGEOM	NOT USED
ANS	LIST9	THE VALUE OF 'B' RETURNED FROM SUBROUTINE LOOK
ANS2	LIST9	THE VALUE OF 'B2' RETURNED FROM SUBROUTINE LOOK
AR	LIST3	NOT USED
AT	LIST8	ACCELERATION PARALLEL TO THE VELOCITY VECTOR (FT/SEC SQ)
AX	LIST8	ACCELERATION PARALLEL TO THE EARTH (FT/SEC SQ)
AZ	LIST8	ACCELERATION NORMAL TO THE EARTH (FT/SEC SQ)
BW	LIST3	WING SPAN (FT)
CARD	LIST1	(INTEGER) UNIT FOR READING CARD INPUT
CBAR	LGEOM	NOT USED
CD	LIST3	DRAG COEFFICIENT
CMACG	LIST3	NOT USED
CL	LIST3	LIFT COEFFICIENT
CMDMP	LIST3	NOT USED
COFF	LIST8	COEFFICIENT OF FRICTION
CLAR	LIST3	NOT USED
CLMAX	LIST3	MAXIMUM LIFT COEFFICIENT
COSAL	LGEOM	NOT USED
COSTH	LGEOM	NOT USED
CX	LIST3	NOT USED
CZ	LIST3	NOT USED
C4HT	LIST15	HEIGHT OF WING ABOVE THE GROUND (FT)
DPDX	LIST9	NOT USED
DPDY	LIST9	NOT USED
DPDZ	LIST9	NOT USED
DP2DX	LIST9	NOT USED
DP2DY	LIST9	NOT USED
DP2DZ	LIST9	NOT USED
DFLTD	LIST2	NOT USED
DFLP	LIST15	FLAP DEFLECTION (DEG)
DRAG	LIST2	AERODYNAMIC FORCE IN THE DRAG DIRECTION (LBS)
DPHO	LIST5	NOT USED
DSPDH	LIST5	NOT USED
DSTHZ	LGEOM	NOT USED
DT	LIST2	NOT USED
DTHHZ	LGEOM	NOT USED
DTHTH	LGEOM	NOT USED
DTHWA	LGEOM	NOT USED
DTIME	LIST15	TIME INTEGRATING INTERVAL (SEC)
DTOR	LIST6	CONVERTS DEGREES TO RADIANS
DWA	LIST4	RAM DRAG FOR ALL ENGINES ALONG THE VELOCITY VECTOR (LBS)
DVLHZ	LGEOM	NOT USED
ENGNO	LIST2	THE NUMBER OF ENGINES OPERATING
FES	LIST4	NOT USED
FLIFT	LIST15	EXCESS FORCE IN THE LIFT DIRECTION (LBS)
FTOK	LIST6	CONVERTS FEET/SECOND TO KNOTS
FTONM	LIST6	CONVERTS FEET TO NAUTICAL MILES
FUEL	LIST2	NOT USED
GALMXR	LIST15	ANGLE OF ATTACK FOR GROUND CONTACT (RADIANS)
GAMR	LIST2	FLIGHT PATH ANGLE (RADIANS)
GN	LIST8	LOAD FACTOR NORMAL TO THE VELOCITY VECTOR (G)
GT	LIST8	LOAD FACTOR PARALLEL TO THE VELOCITY VECTOR (G)
GZ	LIST6	ACCELERATION DUE TO GRAVITY (FT/SEC SQ)
HD(60)	LIST1	VARIABLE NAME AVAILABLE FOR OUTPUT HEADINGS
HF	LIST2	ALTITUDE (FT)
HIND	LIST2	NOT USED
IALMX	LIST15	INDEX SET WHEN MAXIMUM ANGLE OF ATTACK IS REACHED
IATM	LIST5	INDEX TO SELECT ATMOSPHERIC TYPE
IAT	LIST15	NEGATIVE ACCELERATION INDEX DURING TAKEOFF
IDATE	LIST1	DATE (MO/DAY/YEAR)
IERR	LIST99	ERROR INDEX
IFLAG	LIST9	NOT USED
ILOOP	LIST15	LOOP COUNTER IN SUBROUTINE GILL
IYOM	LIST3	NOT USED
IN	LIST2	DIRECTS CALLED SUBROUTINE TO READ INPUTS OR EXECUTE
INDEX	LIST8	OPTION SELECTOR FOR SUBROUTINE AEQFM
INDIC	LIST9	OUT OF FIELD INDICATOR FOR SUBROUTINE LOOK
INT	LIST15	SELECTS VARIABLES TO INTEGRATE IN SUBROUTINE GINTG
INP	LIST1	DESIGNATES SUBROUTINE CALLED BY MAIN PROGRAM
IPUNCH	LIST1	NOT USED
IREV	CONTROL	REVERSE THRUST INDICATOR
ISP	CONTROL	LIFT DUMPER INDICATOR
ITRM	LIST8	DIRECTS OPTION IN AEQFM
IYY	LGEOM	NOT USED
JFIG	CONTROL	INDEX FOR TYPE OF POWERED LIFT SYSTEM

Table 3-3. **Definition of Common Block Variables.** (Contd)

JPOW	LIST4	NOT USED
KTOF	LIST6	(REAL) CONVERTS KNOTS TO FEET/SECOND
LIFT	LIST2	(REAL) AERODYNAMIC FORCE IN THE LIFT DIRECTION (LBS)
LIMIT	LIST1	LINE LIMIT PER OUTPUT PAGE
LINE	LIST1	THE CURRENT NUMBER OF LINES WRITTEN ON THAT PAGE
MACH	LIST2	(REAL) TRUE MACH NUMBER
ND	LIST9	OPTION SELECTOR FOR SUBROUTINE LOOK
NFEP(3)	LIST9	NOT USED
NENG	CONTROL	INDEX FOR REVERSE THRUST MODE
NMTOF	LIST6	(REAL) CONVERTS NAUTICAL MILES TO FEET
NU	LIST5	(REAL) KINEMATIC VISCOSITY (SQ.FT/SEC)
PAGE	LIST1	(INTEGER) CURRENT PAGE NUMBER
PAMB	LIST5	ATMOSPHERIC AMBIENT PRESSURE (LBS/SQ.FT)
PFN	LGEOM	NOT USED
PRINT	LIST1	(INTEGER) UNIT FOR WRITING PRINTED OUTPUT
PZ	LIST6	SEAL LEVEL ATMOSPHERIC PRESSURE (LB/SQ.FT)
Q	LIST3	AIRCRAFT PITCH RATE (RADIANS/SEC)
QDOT	LGEOM	NOT USED
QS	LIST3	DYNAMIC PRESSURE TIMES REFERENCE AREA (LBS)
QSC	LIST3	NOT USED
RCF	LIST2	RATE OF CLIMB (FT/SEC)
RHOZ	LIST6	SEA LEVEL STANDARD DAY AIR DENSITY (SLUGS/CUBIC FT)
RHOZ2	LIST6	RHOZ DIVIDED BY 2
RHZ	LGEOM	NOT USED
RTH	LGEOM	NOT USED
RTOD	LIST6	CONVERTS RADIANS TO DEGREES
RWA	LGEOM	NOT USED
S	LIST3	WING REFERENCE AREA (SQ.FT)
SCP	LGEOM	NOT USED
SHSW	LGEOM	NOT USED
SIG	LIST5	DENSITY RATIO
SINAL	LGEOM	NOT USED
SINTH	LGEOM	NOT USED
SOUND	LIST5	SPEED OF SOUND (FT/SEC)
TEMF	LIST5	AIR TEMPERATURE (DEG FAHRENHEIT)
TEMR	LIST5	AIR TEMPERATURE (DEG RANKINE)
TGROS	LIST4	GROSS THRUST FOR ALL ENGINES ALONG THE THRUST AXIS (LBS)
THETR	LGEOM	AIRCRAFT ATTITUDE (RADIANS)
THIR	LIST3	THRUST INCIDENCE (RADIANS)
THMOM	LIST4	GROSS THRUST AT THE SLOT EXIT FOR IBF CONFIGS (LBS)
THRFO	LIST4	NOT USED
THRST	LIST2	GROSS THRUST PER ENGINE ALONG THE THRUST AXIS (LBS)
THRTL	LIST2	THROTTLE SETTING
THV	LIST2	NOT USED
TIMS	LIST2	TIME (SEC)
TZ	LIST6	SEA LEVEL STANDARD DAY TEMPERATURE (DEG RANKINE)
U	LGEOM	NOT USED
UDOT	LGEOM	NOT USED
VSF	LIST3	STALL SPEED (FT/SEC)
VTF	LIST2	TRUE AIRSPEED (FT/SEC)
VTSQ	LGEOM	VTF SQUARED (FT SQ/SEC SQ)
VUPPER	LIST8	NOT USED
VWF	LIST2	HEADWIND COMPONENT - TRUE AIRSPEED (FT/SEC)
W	LGEOM	NOT USED
WDOT	LGEOM	NOT USED
WINCR	LIST3	WING INCIDENCE (RADIANS)
WT	LIST2	GROSS WT (LBS)
XA	LIST9	X ARGUMENT TO SUBROUTINE LOOK
XCG	LGEOM	NOT USED
XE	LIST15	X DISTANCE IN EARTH AXIS SYSTEM (FT)
XEDOT	LIST15	RATE OF CHANGE OF XE WITH TIME (FT/SEC)
XF	LIST2	NOT USED
YA	LIST9	Y ARGUMENT TO SUBROUTINE LOOK
ZA	LIST9	Z ARGUMENT TO SUBROUTINE LOOK
ZCG	LGEOM	NOT USED
ZE	LIST15	Z DISTANCE IN EARTH AXIS SYSTEM (FT)
ZEDOT	LIST15	RATE OF CHANGE OF ZE WITH TIME (FT/SEC)

SECTION 4

INPUT VARIABLES AND SAMPLE CASE

4.1 INPUT VARIABLES

Input data for the MILSTOL program is read by the executive routine, the trajectory subroutines, the physical data subroutines, and by the page heading subroutine (SKIP). Except for the page heading subroutine, all inputs are in namelist format. The SKIP subroutine reads two 80-character title cards and prints them at the top of each output page. The majority of the variables in the TAKEOFF and LANDING subroutines are input via a data statement and correspond to the ground rules specified in Figures 1-1 and 1-2. If these values are to be changed, the data statement may be overridden by inputting the appropriate variable in the namelist.

Tables 4-1 through 4-6 describe all MILSTOL input variables. The namelist MAIN is read by the executive routine MILSTOL, and its variables are described in Table 4-1. Table 4-2 is the description of the title cards for the SKIP subroutine. Inputs for the maneuver driving subroutines, TAKEOFF and LANDING, are contained in the namelists TAKEOF1 and LAND1, Tables 4-3 and 4-4 respectively. AERT1 and THT1 are the namelists for the aerodynamic and propulsion data subroutines and are described in Tables 4-5 and 4-6.

4.2 SAMPLE CASE

The sample input data presented in Table 4-7 is a representative externally blown flap configuration with an aspect ratio of 8 and a quarter chord sweep angle of 25 degrees. The aerodynamic and propulsion data is consistent with that used in Reference 7 for the configuration definition studies. The input data of Table 4-7 is set up to calculate the "balanced" takeoff distance for a 135,000-pound aircraft at a runway altitude of 2,500 feet on a MIL-STD-210A hot day. After completion of the takeoff calculation, the program will read another set of inputs and calculate a landing at the same runway and weight conditions. The resulting takeoff calculation output for this sample case is shown in Table 4-8; the landing calculation output is shown in Table 4-9.

Table 4-1. Definition of MILSTOL Input Variables.

```
C      DEFINITION OF VARIABLES IN NAMELIST 'MAIN'
C      NWT=   THE NUMBER OF WEIGHTS IN THE WEIGHT MATRIX
C      WTL=   THE LIST OF WEIGHTS (LBS.)
C      NHF=   THE NUMBER OF ALTITUDES IN THE ALTITUDE MATRIX
C      HFL=   THE LIST OF RUNWAY ALTITUDES (FT.)
C      VWK=   THE HEADWIND COMPONENT (KEAS)
C      IATM IS THE INDEX FOR SELECTING ATMOSPHERIC PROPERTIES
C      IATM= 0  STANDARD DAY TEMPERATURES (COMPATIBLE WITH US STANDARD
C               AND ICAO TEMPERATURES)
C          = 1  MIL-STD-210A TROPIC TEMPERATURES
C          = 2  MIL-STD-210A POLAR TEMPERATURES
C          = 3  MIL-STD-210A HOT TEMPERATURES
C          = 4  MIL-STD-210A COLD TEMPERATURES
C          = 5  TEMPERATURE IS SPECIFIED AS 'TEMP' (DEG F)
C      INSEQ IS THE SEQUENCE FOR DATA INPUT
C      INSEQ= 2 SUBROUTINE SKIP - TWO TITLE CARDS
C          = 3 SUBROUTINE AEROI - NAMELIST AERTI
C          = 4 SUBROUTINE THSTI - NAMELIST THTI
C          = 5 SUBROUTINE TAKEOFF - NAMELIST TAKEOFI
C          = 6 SUBROUTINE LANDING - NAMELIST LANDI
C          = 7 TERMINATES READING INPUTS AND BEGINS EXECUTION
C          = 8 STOP - END OF JOB
C      EXSEQ IS THE EXECUTION SEQUENCE (THE NUMBERING SEQUENCE IS THE
C             SAME AS INSEQ)
C      EXAMPLES - INSEQ=2,5,3,4,7,    CAUSES INPUTS TO BE READ BY
C                 SUBROUTINES SKIP,TAKEOFF,AEROI,THSTI IN ORDER AND THEN
C                 EXECUTE THE JOB
C                 EXSEQ=5,8, EXECUTES SUBROUTINE TAKEOFF AND THEN CALLS
C                 STOP
```

Table 4-2. Definition of SKIP Title Cards.

```
C      THE INPUT CONSISTS OF TWO ALPHANUMERIC TITLE CARDS
C      CARD 1 - COLUMNS 2 THROUGH 20 ARE RESERVED FOR ENGINE
C               IDENTIFICATION
C               COLUMNS 21 THROUGH 80 ARE PRINTED AS A TITLE LINE
C      CARD 2 - COLUMNS 1 THROUGH 80 ARE PRINTED AS A SECOND
C               TITLE LINE
```

Table 4-3. Definition of TAKEOFF Input Variables.

```
C      DEFINITION OF VARIABLES IN NAMELIST 'TAKEOFI'
C      VCS = RATIO OF LIFTOFF SPEED TO AIR MINIMUM CONTROL SPEED
C      DGLO=INCREMENTAL LOAD FACTOR REQUIRED AT LIFTOFF
C      DFLP=FLAP SETTING IN DEGREES
C      RCOEF = ROLLING COEFFICIENT OF FRICTION.
C      BCOEF = COEFFICIENT OF BRAKING FRICTION
C      VMCGK= MINIMUM CONTROL SPEED ON THE GROUND (KNOTS)
C      VMCAK = MINIMUM CONTROL SPEED IN THE AIR (KNOTS)
C      ROTATN   ROTATION RATE (DEGREES PER SECOND)
C      TIMR = REACTION TIMR FOR ENGINE FAILURE (SEC)
C      TIMB = BRAKING DELAY AFTER 'TIMR' (SEC)
C      ROTPT      EQUAL TO ZERO  SUPRESSES PRINTING OUTPUT FOR
C           SEGMENTS 4,6 AND 8.

C      THE FOLLOWING VALUES ARE ENTERED AT TIME OF LOADING AND ARE USED
C      UNTIL OVERRIDEN BY READING THE APPROPRIATE VARIABLES IN TAKEOFI
C         VSC=1.05
C         DGLO=0.10
C         DFLP=25.
C         RCOEF=0.10
C         BCOEF=0.30
C         TIMR=1.0
C         TIMB=2.0
C         ROTATN=8.0
C         ROTPT=1.
```

Table 4-4. Definition of LANDING Input Variables.

```
C      DEFINITION OF VARIABLES IN NAMELIST 'LANDI'
C      DFLP=     FLAP SETTING (DEG)
C      HFOB=     OBSTACLE HEIGHT (FT)
C      ROTATN=   ROTATION RATE (DEG/SEC)
C      VMCGK=    MINIMUM CONTROL SPEED ON THE GROUND (KTS)
C      VMCAK=    MINIMUM CONTROL SPEED IN THE AIR (KTS)
C      TBRK=     TIME DELAY AFTER TOUCHDOWN FOR BRAKE APPLICATION (SEC)
C      TSP=      TIME DELAY AFTER TOUCHDOWN FOR SPOILER DEPLOYMENT (SEC)
C      TREV=     TIME DELAY AFTER TOUCHDOWN FOR THRUST REVERSAL (SEC)
C      APR=      RATIO OF APPROACH SPEED TO MINIMUM CONTROL SPEED
C      DGTH=     INCREMENTAL LOAD FACTOR AVAILABLE AT THE OBSTACLE
C      DGTD=     INCREMENTAL LOAD FACTOR AVAILABLE AT TOUCHDOWN
C      RCOEF=    ROLLING COEFFICIENT OF FRICTION
C      BCOEF=    BRAKING COEFFICIENT OF FRICTION
C      GAMMA=    INITIAL FLIGHT PATH ANGLE (DEG) (NEGATIVE IS DESCENDING)
C      RS=       MAXIMUM RATE OF SINK AT TOUCHDOWN (FT/SEC)
C                (POSITIVE IS DESCENDING)
C      ROTPT=    EQUAL TO ZERO SUPRESSES PRINTING OF THE ENTIRE TIME HISTORY

C      THE FOLLOWING VARIABLES ARE ENTERED AT TIME OF LOADING AND ARE USED
C      UNTIL OVERRIDDEN BY READING THE APPROPRIATE VARIABLES IN LANDI
C         DFLP = 60.0
C         HFOB = 50.0
C         ROTATN = 8.0
C         TBRK = 2.0
C         TSP = 2.0
C         TREV = 2.0
C         APR = 1.10
C         DGTH = 0.30
C         DGTD = 0.15
C         RCOEF = 0.10
C         BCOEF = 0.30
C         GAMMA = -7.50
C         RS = 10.0
C         ROTPT = 1.0
```

Table 4-5. Definition of AERO1 Input Variables.

```
C      DEFINITION OF VARIABLES IN NAMELIST 'AERTI'
C      S = WING REFERENCE AREA.
C      BW = WING SPAN
C      C4HT=     HEIGHT OF THE QUARTER CHORD ABOVE THE GROUND (FT)
C      GALMX=    ANGLE OF ATTACK FOR GROUND CONTACT (DEG)
C      ALPMX =   MAXIMUM ALLOWABLE ANGLE OF ATTACK - USUALLY A CL LIMIT
C                (DEG)
C      ACLMD = STALL ANGLE OF ATTACK (DEG)
C      FBP = EQUIVALENT FLAT PLATE AREA OF ADDITIONAL DRAG ITEMS (SQ FT)
C      WINCD = WING INCIDENCE(DEG)
C      DCLSP = INCREMENTAL LIFT COEFFICIENT DUE TO LIFT DUMPERS
C      DCDSP = INCREMENTAL DRAG COEFFICIENT DUE TO LIFT DUMPERS
C      EODCL = INCREMENTAL LIFT COEFFICIENT DUE TO ENG OUT CONTROLS
C      EODCD = INCREMENTAL DRAG COEFFICIENT DUE TO ENG OUT CONTROLS
C      NX,NY,NZ ARE THE NUMBER OF XC,YC,ZC
C      XC = THE LIST OF ANGLES OF ATTACK (DEG)
C      YC = THE LIST OF THRUST COEFFICIENTS
C      ZC = THE LIST OF FLAP DEFLECTIONS (DEG)
C      CLA = LIFT COEFFICIENT AS A FUNCTION OF (ALPHA,CT,FLAP DEFLECTION)
C      CDA = DRAG COEFFICIENT AS A FUNCTION OF (ALPHA,CT,FLAP DEFLECTION)
C      JFIG = 1 FOR MECHANICAL FLAPS PLUS VECTORED THRUST
C             2 FOR EXTERNALLY BLOWN FLAPS
C             3 FOR INTERNALLY BLOWN FLAPS
```

Table 4-6. Definition of THST1 Input Variables.

```
C       DEFINITION OF VARIABLES IN NAMELIST *THT1*
C       N-    NUMBER OF X,Y POINTS IN EACH TABLE
C       VKS- THE VELOCITY TABLE FOR THE PROPULSION TABLES IN KTAS
C            (USED AS THE INDEPENDENT VARIABLE IN ALL TABLES)
C       THST-GROSS THRUST TABLE AT MAX POWER (IN LBS.)
C       RDRG-RAM DRAG TABLE AT MAX POWER (IN LBS.)
C       TIDL-GROSS THRUST TABLE AT IDLE POWER (IN LBS.)
C       TSLT-GROSS THRUST AT THE SLOT EXIT (IBF CONFIGS) AT MAX POWER (LBS)
C       TREV-MAX REVERSE THRUST (EXPRESSED AS A NEGATIVE VALUE - IN LBS.)
C       DWME-WINDMILLING DRAG FOR A DEAD ENGINE (IN LBS.)
C
C       THID-THRUST VECTOR INCIDENCE REF. TO A WATER LINE IN DEG.
C       ENGNO-THE NUMBER OF ENGINES
C       SCALE-SCALING FACTOR FOR THE PROPULSION DATA
C       NFNG=0  NO REVERSE THRUST
C           =1  ALL ENGINES REVERSING
C           =2  ENGINE OUT REVERSING PROCEDURE

C       THE FOLLOWING VARIABLES ARE ENTERED AT TIME OF LOADING AND ARE USED
C       UNTIL OVERRIDDEN BY READING THE APPROPRIATE VARIABLES IN THT1
C          ENGNO = 4.0
C          SCALE = 1.0
C          NENG = 1
```

Table 4-7. Sample Input Data.

```
$MAIN NWT=1, WTL=135000., NHF=1, HFL=2500., IATM=3, VWK=0.,
     INSEQ=2,5,3,4,7, EXSEQ=5,7,  $
 GE 13-F28 ENGINE                    SAMPLE CASE
           FOR AN EXTERNALLY BLOWN FLAP CONFIGURATION
$TAKEOF1 VSC=1.05, DFLP=25., RCOEF=0.1, BCOEF=0.3, VNCGK=65., VNCAK=85.,
          ROTATN=8., TIMR=1., TIMB=2., DGLO=0.1, ROTPT=0.,  $
$AERT1 S=1550., ALPMX=20., GALMX=20., ACLMD=27., JFIG=2, DCLSP=-2.0,
        DCDSP=0.25, EODCL=-0.45, EODCD=0.06,
          NX=10.,    NY=4.,    NZ=4.,
   XC= -4.,0.,4.,8.,12.,16.,20.,24.,28.,32.,
   YC= 0.0,1.0,3.0,5.0,
   ZC= 0.0,30.0,60.0,80.0,
   CLA=
         -0.25,      0.03,      0.35,      0.67,      0.99,
          1.30,      1.62,      1.94,      2.25,      2.50,
         -0.35,      0.01,      0.34,      0.74,      1.14,
          1.51,      1.91,      2.30,      2.70,      3.10,
         -0.75,     -0.25,      0.33,      0.86,      1.39,
          1.93,      2.47,      2.99,      3.52,      4.00,
         -1.00,     -0.35,      0.31,      0.96,      1.63,
          2.31,      2.98,      3.64,      4.31,      4.90,
          0.85,      1.19,      1.51,      1.83,      2.19,
          2.52,      2.82,      3.09,      3.15,      3.00,
          0.95,      1.44,      1.96,      2.49,      3.01,
          3.53,      4.05,      4.48,      4.65,      4.50,
          1.30,      1.98,      2.66,      3.35,      4.02,
          4.72,      5.39,      6.07,      6.71,      7.10,
          1.50,      2.33,      3.14,      3.93,      4.73,
          5.53,      6.28,      7.11,      7.87,      8.50,
          1.35,      1.65,      1.96,      2.28,      2.59,
          2.90,      3.21,      3.50,      3.33,      3.00,
          2.50,      2.91,      3.37,      3.81,      4.29,
          4.76,      5.18,      5.45,      5.26,      5.00,
          3.80,      4.43,      5.03,      5.66,      6.28,
          6.83,      7.30,      7.65,      7.90,      8.00,
          4.80,      5.45,      6.12,      6.80,      7.48,
          8.13,      8.87,      9.11,      9.42,      9.50,
          1.35,      1.65,      1.96,      2.28,      2.59,
          2.90,      3.21,      3.50,      3.33,      3.00,
          2.50,      2.91,      3.37,      3.81,      4.29,
          4.76,      5.18,      5.45,      5.26,      5.00,
          3.80,      4.43,      5.03,      5.66,      6.28,
          6.83,      7.30,      7.65,      7.90,      8.00,
          4.80,      5.45,      6.12,      6.80,      7.48,
          8.13,      8.87,      9.11,      9.42,      9.50,
```

Table 4-7. Sample Input Data. (Contd)

```
CDA =
        0.15,      0.15,      0.15,      0.16,      0.19,
        0.20,      0.22,      0.26,      0.33,      0.45,
       -0.90,     -0.90,     -0.88,     -0.85,     -0.82,
       -0.78,     -0.73,     -0.67,     -0.58,     -0.40,
       -2.90,     -2.90,     -2.85,     -2.83,     -2.79,
       -2.73,     -2.66,     -2.55,     -2.41,     -2.10,
       -4.90,     -4.90,     -4.85,     -4.83,     -4.77,
       -4.68,     -4.56,     -4.43,     -4.23,     -3.90,
        0.21,      0.21,      0.24,      0.30,      0.38,
        0.47,      0.58,      0.71,      0.91,      0.65,
       -0.42,     -0.42,     -0.36,     -0.26,     -0.12,
        0.08,      0.32,      0.54,      0.83,      0.55,
       -2.02,     -2.02,     -1.95,     -1.80,     -1.57,
       -1.28,     -0.93,     -0.51,      0.01,       .40,
       -3.02,     -3.02,     -2.96,     -2.81,     -2.57,
       -2.22,     -1.89,     -1.24,     -0.56,      0.10,
        0.35,      0.35,      0.39,      0.45,      0.53,
        0.63,      0.76,      0.94,      1.15,      0.65,
        0.34,      0.34,      0.46,      0.62,      0.83,
        1.08,      1.37,      1.66,      1.85,      1.25,
       -0.40,     -0.30,     -0.05,      0.26,      0.63,
        1.04,      1.47,      1.80,      2.13,      2.25,
       -0.80,     -0.70,     -0.38,      0.02,      0.49,
        1.01,      1.48,      1.88,      2.20,      2.30,
        0.35,      0.35,      0.39,      0.45,      0.53,
        0.63,      0.76,      0.94,      1.15,      0.65,
        0.34,      0.34,      0.46,      0.62,      0.83,
        1.08,      1.37,      1.66,      1.85,      1.25,
       -0.40,     -0.30,     -0.05,      0.26,      0.63,
        1.04,      1.47,      1.80,      2.13,      2.25,
       -0.80,     -0.70,     -0.38,      0.02,      0.49,
        1.01,      1.48,      1.88,      2.20,      2.30,

$THT1 THID=0., ENGNO=4., SCALE=1.,
  N=11,
  VKS=0.0,20.0,40.0,60.0,80.0,100.0,120.0,140.0,160.0,180.0,200.0,
  THST=16480.,16600.,16735.,16893.,17098.,17400.,17707.,18029.,18378.,
      18773.,19289.,
  RORG= 0.0,693.,1386.,2080.,2773.,3466.,4160.,4901.,5649.,6412.,7210.,
   TIDL=490.,370.,260.,155.,65.,-30.,-125.,-210.,-295.,-370.,-450.,
   TREV=0.0,-3000.,-6000.,-7410.,-7190.,-6970.,-6780.,-6580.,-6400.,-6210.,
      -6050.,
   $
$MAIN INSEQ=6,3,7, EXSEQ=6,8,  $
$LAND1 DFLP=50., HFOB=50., ROTATN=8., TBRK=2., TSP=2., TREV=2., APR=1.1,
      DGTH=0.3, DGTD=0.15, RCOEF=0.1, BCOEF=0.3, GAMMA=-7.5, RS=12.,
      ROTPT=0.,  $
$AERT1 ALPMX=18., GALMX=20., ACLMD=24.,  $
```

Table 4-8. Takeoff Calculation Output.

OUTPUT DEFINITIONS - TAKEOFF
SEGMENT = 1 DETERMINATION OF LIFTOFF ANGLE OF ATTACK
2 INCREASE VLO TO AVOID GROUND CONTACT
3 R/C AVAILABLE AT LIFTOFF - 1 ENGINE OUT
4 VLO TO VR INTEGRATION STEPS
5 CONDITIONS AFTER VLO TO VR INTEGRATION
6 VR TO VF INTEGRATION STEPS
7 CONDITIONS AFTER VR TO VF INTEGRATION
8 VF TO STOP INTEGRATION STEPS
9 CONDITIONS AFTER VF TO STOP INTEGRATION
10 V=0 TO VR INTEGRATION STEPS
11 CONDITIONS AFTER GROUND RUN TO VR

AIRCRAFT PERFORMANCE 12/07/72 20.38
3 ENGINE GE 13-F28 ENGINE S = 1550 SQ FT HOT TEMPERATURES
SAMPLE CASE
FOR AN EXTERNALLY BLOWN FLAP CONFIGURATION
STOL TAKEOFF

SEGMENT	ALTITUDE (FEET) / TIME (SECONDS)	FLAP (DEGREES) / DISTANCE (FEET)	WEIGHT (POUNDS) / HEIGHT (FEET)	AIRSPEED (KNOTS) / FLT. PATH (DEGREES)	R/C ACTUAL (FT/MIN) / ALPHA (DEGREES)	THETA (DEGREES) / THETA DOT (DEG/SEC)	NOR. ACCEL (F/S/S) / TAN. ACCEL (F/S/S)	CL / CD	LIFT (POUNDS) / DRAG (POUNDS)	THRUST (POUNDS) / THST ALPHA (DEGREES)
2	2500.000	25.000	135000.000	89.250	0.000	20.831	.00007	3.25714	116617.137	51693.579
	0.000	0.000	0.000	0.000	20.831	0.00000	-1.61236	1.27919	45799.441	20.831

AIRPLANE LIFTS OFF AT ALPHA = 20.831 DEG
VLO INCREASED TO 1.060 TIMES VMCAIR

SEGMENT	ALTITUDE / TIME	FLAP / DISTANCE	WEIGHT / HEIGHT	AIRSPEED / FLT. PATH	R/C ACTUAL / ALPHA	THETA / THETA DOT	NOR. ACCEL / TAN. ACCEL	CL / CD	LIFT / DRAG	THRUST / THST ALPHA
2	2500.000	25.000	135000.000	90.100	0.000	20.462	.00006	3.20414	116914.902	51732.464
	0.000	0.000	0.000	0.000	20.462	0.00000	-1.59083	1.25446	45774.527	20.462

AIRPLANE LIFTS OFF AT ALPHA = 20.462 DEG
VLO INCREASED TO 1.070 TIMES VMCAIR

SEGMENT	ALTITUDE / TIME	FLAP / DISTANCE	WEIGHT / HEIGHT	AIRSPEED / FLT. PATH	R/C ACTUAL / ALPHA	THETA / THETA DOT	NOR. ACCEL / TAN. ACCEL	CL / CD	LIFT / DRAG	THRUST / THST ALPHA
2	2500.000	25.000	135000.000	90.950	0.000	20.106	.00005	3.15228	117203.248	51771.624
	0.000	0.000	0.000	0.000	20.106	0.00000	-1.57239	1.23068	45757.143	20.106

AIRPLANE LIFTS OFF AT ALPHA = 20.106 DEG
VLO INCREASED TO 1.080 TIMES VMCAIR

SEGMENT	ALTITUDE / TIME	FLAP / DISTANCE	WEIGHT / HEIGHT	AIRSPEED / FLT. PATH	R/C ACTUAL / ALPHA	THETA / THETA DOT	NOR. ACCEL / TAN. ACCEL	CL / CD	LIFT / DRAG	THRUST / THST ALPHA
1	2500.000	25.000	135000.000	91.800	0.000	19.758	.00003	3.10163	117483.411	51811.036
	0.000	0.000	0.000	0.000	19.758	0.00000	-1.53843	1.20571	45670.538	19.758
3	2500.000	25.000	135000.000	91.800	-439.323	17.018	-.00404	3.09790	117344.446	51811.036
	0.000	0.000	0.000	-2.708	19.727	0.00000	-.00000	1.20395	45604.051	19.727
5	2500.000	25.000	135000.000	89.663	0.000	0.000	-25.62934	.75994	27461.118	51712.453
	-2.470	-380.394	0.000	0.000	0.000	0.00000	2.68420	.56378	20372.539	0.000
7	2500.000	25.000	135000.000	65.000	0.000	0.000	-27.70504	.98742	18751.650	50813.672
	-11.632	-1540.093	0.000	0.000	0.000	0.00000	4.57499	.69680	13232.575	0.000
9	2500.000	25.000	135000.000	0.000	0.000	0.000	-32.17405	-.95023	0.000	490.000
	11.150	717.122	0.000	0.000	0.000	0.00000	-9.53543	.43691	0.000	0.000

Table 4-8. Takeoff Calculation Output. (Contd)

FIELD NOT BALANCED NEW V1 SPEED = 79.55 KTS

1	2500.000	25.000	135000.000	91.800	0.000	19.758	.00003	3.10163	117485.411	51811.036
	0.000	0.000	0.000	0.000	19.758	0.00000	-1.53843	1.20571	45670.538	19.758
3	2500.000	25.000	135000.000	91.800	-439.323	17.018	-.00404	3.09790	117344.446	51811.036
	0.000	0.000	0.000	-2.708	19.727	0.00000	-.00000	1.20395	45604.051	19.727
5	2500.000	25.000	135000.000	89.663	0.000	0.000	-25.62934	.75994	27461.118	51712.453
	-2.470	-380.394	0.000	0.000	0.000	0.00000	2.68420	.56378	20372.539	0.000
7	2500.000	25.000	135000.000	79.552	0.000	0.000	-26.50594	.82429	23447.301	51277.685
	-5.549	-794.659	0.000	0.000	0.000	0.00000	3.58162	.60316	17157.322	0.000
9	2500.000	25.000	135000.000	0.000	0.000	0.000	-32.17405	-.95023	0.000	490.000
	12.368	931.045	0.000	0.000	0.000	0.00000	-9.53543	.43691	0.000	0.000

FIELD NOT BALANCED NEW V1 SPEED = 83.25 KTS

1	2500.000	25.000	135000.000	91.800	0.000	19.758	.00003	3.10163	117485.411	51811.036
	0.000	0.000	0.000	0.000	19.758	0.00000	-1.53843	1.20571	45670.538	19.758
3	2500.000	25.000	135000.000	91.800	-439.323	17.018	-.00404	3.09790	117344.446	51811.036
	0.000	0.000	0.000	-2.708	19.727	0.00000	-.00000	1.20395	45604.051	19.727
5	2500.000	25.000	135000.000	89.663	0.000	0.000	-25.62934	.75994	27461.118	51712.453
	-2.470	-380.394	0.000	0.000	0.000	0.00000	2.68420	.56378	20372.539	0.000
7	2500.000	25.000	135000.000	83.254	0.000	0.000	-26.25401	.79733	24840.841	51428.851
	-3.685	-538.326	0.000	0.000	0.000	0.00000	3.20343	.58786	18314.363	0.000
9	2500.000	25.000	135000.000	0.000	0.000	0.000	-32.17405	-.95023	0.000	490.000
	12.657	986.031	0.000	0.000	0.000	0.00000	-9.53543	.43691	0.000	0.000

FIELD NOT BALANCED NEW V1 SPEED = 82.45 KTS

1	2500.000	25.000	135000.000	91.800	0.000	19.758	.00003	3.10163	117485.411	51811.036
	0.000	0.000	0.000	0.000	19.758	0.00000	-1.53843	1.20571	45670.538	19.758
3	2500.000	25.000	135000.000	91.800	-439.323	17.018	-.00404	3.09790	117344.446	51811.036
	0.000	0.000	0.000	-2.708	19.727	0.00000	-.00000	1.20395	45604.051	19.727
5	2500.000	25.000	135000.000	89.663	0.000	0.000	-25.62934	.75994	27461.118	51712.453
	-2.470	-380.394	0.000	0.000	0.000	0.00000	2.68420	.56378	20372.539	0.000
7	2500.000	25.000	135000.000	82.453	0.000	0.000	-26.32758	.80278	24531.354	51395.024
	-4.102	-596.729	0.000	0.000	0.000	0.00000	3.26806	.59106	18061.668	0.000
9	2500.000	25.000	135000.000	0.000	0.000	0.000	-32.17405	-.95023	0.000	490.000
	12.595	973.438	0.000	0.000	0.000	0.00000	-9.53543	.43691	0.000	0.000

FIELD NOT BALANCED NEW V1 SPEED = 82.49 KTS

1	2500.000	25.000	135000.000	91.800	0.000	19.758	.00003	3.10163	117485.411	51811.036
	0.000	0.000	0.000	0.000	19.758	0.00000	-1.53843	1.20571	45670.538	19.758
3	2500.000	25.000	135000.000	91.800	-439.323	17.018	-.00404	3.09790	117344.446	51811.036
	0.000	0.000	0.000	-2.708	19.727	0.00000	-.00000	1.20395	45604.051	19.727

Table 4-8. Takeoff Calculation Output. (Contd)

5	2500.000	25.000	135000.000	89.663	0.000	0.000	-25.62934	.75994	27461.118	51712.453
	-2.470	-380.394	0.000	0.000	0.000	8.00000	2.58420	.56378	20372.539	0.000
7	2500.000	25.000	135000.000	82.495	0.000	0.000	-26.32379	.80253	24547.269	51396.769
	-4.081	-593.743	0.000	0.000	0.000	8.00000	3.26471	.59090	18074.749	0.000
9	2500.000	25.000	135000.000	0.000	0.0[illegible]	0.000	-32.17405	-.95023	0.000	490.000
	12.598	974.034	0.000	0.000	0.000	6.00000	-9.53543	.43691	0.000	0.000

FIELD BALANCED IN 4 ITERATIONS

1	2500.000	25.000	135000.000	82.495	0.000	0.000	-22.29382	1.35530	41456.723	68529.025
	21.568	1431.337	0.000	0.000	0.000	6.00000	7.10380	.58616	17929.988	0.000

ENGINE FAILURE SPEED = 82.49 KTS
ROTATION SPEED = 89.66 KTS
LIFTOFF SPEED = 91.80 KTS
STALL SPEED = 78.97 KTS (WITH ONE ENGINE OUT)

V=0 TO LIFTOFF
DISTANCE =2405.47 FT TIME = 28.22 SEC

ABORTED TAKEOFF
DISTANCE =2405.43 FT TIME = 34.27 SEC

Table 4-9. Landing Calculation Output.

```
OUTPUT DEFINITIONS - LANDING
SEGMENT = 1  DETERMINATION OF ANGLE OF ATTACK AT TOUCHDOWN
          2  R/C AVAILABLE AT TOUCHDOWN - 1 ENGINE OUT
          3  DETERMINATION OF ANGLE OF ATTACK AT THE OBSTACLE
          4  R/C AVAILABLE AT THE OBSTACLE - 1 ENGINE OUT
          5  INTEGRATION FROM OBSTACLE TO TOUCHDOWN
          6  TOUCHDOWN TO NOSEDOWN INTEGRATION
          7  CONDITIONS AT NOSEDOWN
          8  NOSEDOWN TO STOP INTEGRATION
          9  CONDITIONS AT STOP

          MAXIMUM RATE-OF-SINK EXCEEDED.  GAMMA RESET TO -5.226 DEG

AIRCRAFT PERFORMANCE                                  12/07/72          30.40
          4 ENGINE GE 13-F20 ENGINE    S =  1550 SQ FT               HOT TEMPERATURES
                          SAMPLE CASE
              FOR AN EXTERNALLY BLOWN FLAP CONFIGURATION

      STOL LANDING
```

SEGMENT	ALTITUDE (FEET) / TIME (SECONDS)	FLAP (DEGREES) / DISTANCE (FEET)	WEIGHT (POUNDS) / HEIGHT (FEET)	AIRSPEED (KNOTS) / FLT. PATH (DEGREES)	R/C ACTUAL (FT/MIN) / ALPHA (DEGREES)	THETA (DEGREES) / THETA DOT (DEG/SEC)	NOR. ACCEL (F/S/S) / TAN. ACCEL (F/S/S)	CL / CD	LIFT (POUNDS) / DRAG (POUNDS)	THRUST (POUNDS) / THST ALPHA (DEGREES)
1	2500.000	50.000	135000.000	78.047	-720.000	4.353	-.00014	4.52252	123022.833	63700.916
	0.000	0.000	0.000	-5.226	9.579	0.000	-.00002	2.37733	65009.261	9.579
2	2500.000	50.000	135000.000	78.047	-1562.396	3.257	-.00004	4.36007	119375.004	51223.530
	0.000	0.000	0.000	-11.400	14.657	0.000	-.00000	2.48818	68124.227	14.657
3	2550.000	50.000	135000.000	78.047	-720.000	4.377	-.00003	4.52404	123709.744	63033.062
	0.000	0.000	50.000	-5.226	9.603	0.000	.00004	2.38206	65121.093	9.603
4	2550.000	50.000	135000.000	78.047	-1563.918	3.278	-.00004	4.36535	119342.319	51223.530
	0.000	0.000	50.000	-11.411	14.689	0.000	-.00000	2.49256	68143.025	14.689
5	2500.000	50.000	135000.000	78.047	-720.000	4.353	-.00003	4.52253	123023.196	63709.702
	4.167	546.648	0.000	-5.226	9.579	0.000	.00004	2.37735	65009.726	9.579
7	2500.000	50.000	135000.000	76.085	0.000	0.000	-22.38502	1.57842	41070.692	329.532
	.544	70.777	0.000	0.000	0.000	-0.000	-4.18340	.32433	8439.217	0.000

```
      LIFT DUMPERS DEPLOYED

      BRAKES ON

      REVERSE THRUST ON
```

SEGMENT										
9	2500.000	50.000	135000.000	0.000	0.000	0.000	-32.17405	-.43503	0.000	900.000
	10.527	676.107	0.000	0.000	0.000	0.000	-9.41065	.56167	0.000	0.000

```
GROUND DISTANCE =   747.0 FT    GROUND TIME = 11.07 SEC    GLIDE SLOPE = -5.23 DEG

AIR DISTANCE =   546.6 FT      AIR TIME =  4.17 SEC     APPROACH SPEED = 78.05 KTS

TOTAL DISTANCE =  1293.6 FT     TOTAL TIME = 15.24 SEC      STALL SPEED = 60.45 KTS
```

SECTION 5
REFERENCES

1. Letter from Garland S. Oates, "Revision of Takeoff and Landing Ground Rules for STAI," Air Force Flight Dynamics Laboratory (AFSC), 20 July 1972.

2. D. L. Gross, Aircraft Performance Analysis System, GDC-ERR-1453, Convair Division of General Dynamics, December 1969.

3. U.S. Standard Atmosphere, 1962, National Aeronautics and Space Administration, December 1962.

4. Climatic Extremes for Military Equipment, MIL-STD-2104, June 1953.

5. M. B. Eilert "Low Speed Wind Tunnel Test of a 1/20th Scale STOL Tactical Transport Model With Wing Sweep of 25 Degrees Aspect Ratio of 8.0," General Dynamics Corporation GDLST 612-3, (Unpublished).

6. S. Gill, "A Process for the Step-by-Step Integration of Differential Equations in an Automatic Digital Computing Machine," Cambridge Philosophical Society Proceedings, Volume 47, Part 1, January 1951.

7. C. A. Whitney Configuration Definition and Performance of MST Configurations Developed for the STOL Tactical Aircraft Investigation, TN-71-STOL-008, Convair Aerospace Division of General Dynamics, San Diego Operation, November 1971.

APPENDIX I

PROGRAM AND SUBROUTINE SOURCE LISTINGS

The following source listings are contained in this appendix.

```
      PROGRAM MILSTOL(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     CDC 6400 SHORT TAKEOFF AND LANDING COMPUTER PROGRAM USING MILITARY
C     GROUND RULES
C     THIS PROGRAM REQUIRES INPUTS
C     THIS PROGRAM CALLS THE FOLLOWING ENTRIES
C         SKIP,TAKEOFF,LANDING,AERO1,THST1

      INTEGER CARD,PRINT,PAGE,EXSEQ
      PEAL LIFT,MACH,NU,KTOF,NMTOF

      DIMENSION EXSEQ(2),INSEQ(5),WTL(10),HFL(10)

      COMMON /LIST1/LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,HD(60)
      COMMON /LIST2/DELTD,ALFAR,THRTL,GAMR,HIND,VTF,WT,HF,DT,TIMS,XF,
     * FUEL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN
      COMMON /LIST3/CL,CD,S,QS,THIR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHD,CX,
     * CZ,Q,IMOM,WINCR,CMDMP,QSC,ALMXR,VSF,BW
      COMMON /LIST4/FFS,JPOW,TGROS,DWA,THMOM,THREQ
      COMMON /LIST5/SIG,SOUND,NU,TEMR,PAMB,IATM,TEMF,DSODH,DRHO
      COMMON /LIST6/RTOD,DTOR,KTOF,FTOK,NMTOF,FTONM,RHOZ,RHOZ2,GZ,PZ,TZ
      COMMON/LIST8/INDEX,COEF,GN,GT,AN,AT,AX,AZ,ITRM,VUPPER
      COMMON/LIST9/ANS,ANS2,ND,XA,YA,ZA,INDIC,IFLAG,NDER(3),DBDX,DBDY,
     * DBDZ,DB2DX,DB2DY,DB2DZ
      COMMON/LIST15/IALMX,INT,XEDOT,ZEDOT,XE,ZE,ILOOP,DTIME,   IAT,FLIFT,
     1DFLP,C4HT,GALMXR
      COMMON/LGEOM/SHSW,SCR,QDOT,RTH,DTHTH,RWA,DTHWA,DSTHZ,DWLHZ,IYY,
     1 XCG,ZCG,CBAR,PFN,SINTH,COSTH,ANGLE,SINAL,COSAL,RHZ,DTHHZ,VTSQ,
     2 UDOT,WDOT,THETR,U,W
      COMMON/CONTROL/JFIG,IREV,ISP,NENG
      COMMON/LIST99/IERR

      DATARTOD,KTOF,NMTOF,RHOZ,RHOZ2,GZ,PZ,TZ/57.2957795,1.688,6076.8,.0
     1023768924,.0011884462,32.174049,2116.2218,518.67/

C     DEFINITION OF VARIABLES IN NAMELIST 'MAIN'
C     NWT=    THE NUMBER OF WEIGHTS IN THE WEIGHT MATRIX
C     WTL=    THE LIST OF WEIGHTS (LBS.)
C     NHF=    THE NUMBER OF ALTITUDES IN THE ALTITUDE MATRIX
C     HFL=    THE LIST OF RUNWAY ALTITUDES (FT.)
C     VWK=    THE HEADWIND COMPONENT (KEAS)
C     IATM IS THE INDEX FOR SELECTING ATMOSPHERIC PROPERTIES
C     IATM= 0  STANDARD DAY TEMPERATURES (COMPATIBLE WITH US STANDARD
C              AND ICAO TEMPERATURES)
C         = 1  MIL-STD-210A TROPIC TEMPERATURES
C         = 2  MIL-STD-210A POLAR TEMPERATURES
C         = 3  MIL-STD-210A HOT TEMPERATURES
C         = 4  MIL-STD-210A COLD TEMPERATURES
C         = 5  TEMPERATURE IS SPECIFIED AS 'TEMF' (DEG F)
C     INSEQ IS THE SEQUENCE FOR DATA INPUT
C     INSEQ= 2 SUBROUTINE SKIP - TWO TITLE CARDS
C         = 3 SUBROUTINE AERO1 - NAMELIST AERT1
C         = 4 SUBROUTINE THST1 - NAMELIST THT1
C         = 5 SUBROUTINE TAKEOFF - NAMELIST TAKEOF1
C         = 6 SUBROUTINE LANDING - NAMELIST LAND1
C         = 7 TERMINATES READING INPUTS AND BEGINS EXECUTION
C         = 8 STOP - END OF JOB
```

```
C      EXSEQ IS THE EXECUTION SEQUENCE (THE NUMBERING SEQUENCE IS THE
C            SAME AS INSEQ)
C      EXAMPLES - INSEQ=2,5,3,4,7,    CAUSES INPUTS TO BE READ BY
C                 SUBROUTINES SKIP,TAKEOFF,AERO1,THST1 IN ORDER AND THEN
C                 EXECUTE THE JOB
C                 EXSEQ=5,8, EXECUTES SUBROUTINE TAKEOFF AND THEN CALLS
C                 STOP

      NAMELIST/MAIN/NWT,WTL,NHF,HFL,VWK,IATM,TEMF,INSEQ,EXSEQ

      CARD=5
      PRINT=6
      IDATE=MIORF(31B)
      LIMIT=55
      DTOR=1./RTOD
      FTOK=1./KTOF
      FTONM=1./NMTOF
      PAGE=1
    1 IN=1
      READ(CARD,MAIN)
      WRITE(PRINT,MAIN)
      I=1
    5 INP=INSEQ(I)
      GO TO(99,20,30,40,50,60,150,160),INP
   20 CALL SKIP
      GO TO 90
   30 CALL AERO1
      GO TO 90
   40 CALL THST1
      GO TO 90
   50 CALL TAKEOFF
      GO TO 90
   60 CALL LANDING
      GO TO 90
   99 WRITE(PRINT,1000)
 1000 FORMAT(1H1,10X,40HERROR IN NAMELIST 'MAIN', INSEQ OR EXSEQ)
      STOP
   90 IF(IN.GT.1)GO TO 140
      I=I+1
      GO TO 5
  150 II=1
      IN = 2
  151 IEX=EXSEQ(II)-4
      DO 152 J=1,NHF
      DO 153 K=1,NWT
      HF=HFL(J)
      CALL ATMOS
      VWF=VWK*KTOF/SQRT(SIG)
      WT=WTL(K)
      IERR=0
      INP=EXSEQ(II)
      GO TO (50,60,1,160),IEX
  140 CONTINUE
  153 CONTINUE
  152 CONTINUE
      II=II+1
      GO TO 151
  160 WRITE(PRINT,1001)
 1001 FORMAT(1H1,10X,10HEND OF JOB)
      STOP
      END
```

```
      PROGRAM MILSTOL(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)

C     CDC 6400 DUMP OVERLAY PROGRAM FOR 'MILSTOL'
C     THE DUMP IS A FLOATING POINT NAMELIST PRINTOUT

      INTEGER CARD,PRINT,PAGE,EXSEQ
      REAL LIFT,MACH,NU,KTOF,NMTOF

      COMMON /LIST1/LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,HD(60)
      COMMON /LIST2/DELTD,ALFAR,THRTL,GAMR,HIND,VTF,WT,HF,DT,TIMS,XF,
     * FUEL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN
      COMMON /LIST3/CL,CD,S,QS,THIR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHD,CX,
     * CZ,Q,IMOM,WINCR,CMDMP,QSC,ALMXR,VSF,BW
      COMMON /LIST4/FFS,JPOW,TGROS,DWA,THMOM,THREQ
      COMMON /LIST5/SIG,SOUND,NU,TEMR,PAMB,IATM,TEMF,DSODH,DRHO
      COMMON /LIST6/RTOD,DTOR,KTOF,FTOK,NMTOF,FTONM,RHOZ,RHOZ2,GZ,PZ,TZ
      COMMON/LIST8/INDEX,COEF,GN,GT,AN,AT,AX,AZ,ITRM,VUPPER
      COMMON/LIST9/ANS,ANS2,ND,XA,YA,ZA,INDIC,IFLAG,NDER(3),DBDX,DBDY,
     * DBDZ,DB2DX,DB2DY,DB2DZ
      COMMON/LIST15/IALMX,INT,XEDOT,ZEDOT,XE,ZE,ILOOP,DTIME,   IAT,FLIFT,
     1DFLP,C4HT,GALMXR
      COMMON/LGEOM/SHSW,SCR,QDOT,RTH,DTHTH,RWA,DTHWA,DSTHZ,DWLHZ,IYY,
     1 XCG,ZCG,CBAR,PFN,SINTH,COSTH,ANGLE,SINAL,COSAL,RHZ,DTHHZ,VTSQ,
     2 UDOT,WDOT,THETR,U,W
      COMMON/CONTROL/JFIG,IREV,ISP,NENG
      COMMON/LIST99/IERR

      NAMELIST/DUMP/LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,DFLP,
     1               DELTD,ALFAR,THRTL,GAMR,HIND,VTF,WT,HF,DT,TIMS,XF,
     2 FUEL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN,C4HT,GALMXR
     3              ,CL,CD,S,QS,THIR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHD,CX,
     4 CZ,Q,IMOM,WINCR,CMDMP,QSC,ALMXR,VSF,BW
     5              ,FFS,JPOW,TGROS,DWA,THMOM,THREQ
     6              ,SIG,SOUND,NU,TEMR,PAMB,IATM,TEMF,DSODH,DRHO
     7              ,RTOD,DTOR,KTOF,FTOK,NMTOF,FTONM,RHOZ,RHOZ2,GZ,PZ,TZ
     8             ,INDEX,COEF,GN,GT,AN,AT,AX,AZ,ITRM,VUPPER
     9             ,ANS,ANS2,ND,XA,YA,ZA,INDIC,IFLAG,NDER      ,DBDX,DBDY,
     1 DBDZ,DB2DX,DB2DY,DB2DZ,IALMX,INT,XEDOT,ZEDOT,XE,ZE,ILOOP,DTIME,
     2IAT,FLIFT     ,SHSW,SCR,QDOT,RTH,DTHTH,RWA,DTHWA,DSTHZ,DWLHZ,IYY,
     3 XCG,ZCG,CBAR,PFN,SINTH,COSTH,ANGLE,SINAL,COSAL,RHZ,DTHHZ,VTSQ,
     4 UDOT,WDOT,THETR,U,W,JFIG,IREV,ISP,IERR

      WRITE(6,DUMP)
      STOP
      END
```

```
      SUBROUTINE AEQFM
C     6400 SUBROUTINE FOR CALCULATING ACCELERATIONS.

C     THIS SUBROUTINE CALLS FOR THE FOLLOWING ENTRIES.
C          THST1, AERO2, TRIM

C     ALL INDEX OPTIONS ASSUME WEIGHT AND ALTITUDE ARE SPECIFIED)
C     ALL EXCEPT 3 ASSUME GAMR SPECIFIED.
C     ALL EXCEPT 6 ASSUME POWER SETTING SPECIFIED.
C     ALL EXCEPT 7 ASSUME VELOCITY SPECIFIED.
C     GN AND GT ARE  RESPECTIVELY THE NORMAL AND TANGENTIAL LOAD FACTORS
C     INDEX = 1 FOR CALCULATING GROUND ACCELERATIONS(GIVEN ALFAR, GAMR,
C     AND COEF).
C     INDEX = 2 FOR CALCULATING AT AND AN(GIVEN ALFAR, GAMR).
C     INDEX = 3 FOR CALCULATING ALFAR AND GAMR(GIVEN GN AND GT).
C     INDEX = 4 FOR CALCULATING ALFAR AND AN(GIVEN GAMR AND GT).
C     INDEX = 5 FOR CALCULATING ALFAR AND AT(GIVEN GAMR AND GN).
C     INDEX = 6 FOR CALCULATING ALFAR,THRTL,    (GIVEN GAMR,GN,GT).
C     INDEX = 7 FOR CALCULATING ALFAR,VTF(GIVEN GAMR,GN,GT).

      INTEGER CARD,PRINT,PAGE
      REAL LIFT,MACH,NU,KTOF,NMTOF,IYY

      DIMENSION SA(3),SV(6),SPV(6),ST(3),JA(3),JV(3),A(6),V(6),PV(6),T(6
     1)
      COMMON /LIST1/ LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,HD(60)

      COMMON /LIST2/ DELTD,ALFAR,THRTL,GAMR,HIND,VTF,WT,HF,DT,TIMS,XF,FU
     1EL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN
      COMMON /LIST3/ CL,CD,S,QS,THIR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHD,CX,
     1CZ,Q,IMOM,WINCR,CMDMP,QSC,ALMXR,VSF,BW
      COMMON /LIST4/ FFS,JPOW,TGROS,DWA,THMOM,THREQ
      COMMON /LIST5/ SIG,SOUND,NU,TEMR,PAMB,IATM,TEMF,DSODH,DRHO
      COMMON /LIST6/ RTOD,DTOR,KTOF,FTOK,NMTOF,FTONM,RHOZ,RHOZ2,GZ,PZ,TZ
      COMMON /LIST8/ INDEX,COEF,GN,GT,AN,AT,AXE,AZE,ITRM,VUPPER
      COMMON /LGEOM/ SHSW,SCR,QDOT,RTH,DTHTH,RWA,DTHWA,DSTHZ,DWLHZ,IYY,X
     1CG,ZCG,CBAR,PFN,SINTH,COSTH,ANGLE,SINAL,COSAL,RHZ,DTHHZ,VTSQ,UDOT,
     2WDOT,THETR,U,W
      COMMON /LIST99/ IERR

      EQUIVALENCE (SA(1),QCDOT), (SA(2),ACN), (SA(3),ACT), (SV(1),DELTD)

      DATA ST,SPV/.0003,2*.0002,.2,.0034906584,-.02,.0034906584,.2,8./

      RECPM=GZ/WT
      RHOS=RHOZ2*SIG*S
      K=-1
      NVSTP=0
      DVTF=0.025*SOUND
      NSTEP=0

C     SET SUBSCRIPTS FOR A,V,PV,T.

      I=1
      IF (IMOM.EQ.0) GO TO 10
      JA(I)=1
```

```
      JV(I)=1
      IF (ITRM.EQ.2) JV(I)=5
      IF (INDEX.LT.3) GO TO 50
      I=I+1
      GO TO 20
   10 IF (INDEX.LT.3) GO TO 100
   20 JA(I)=2
      IF (INDEX.EQ.4) JA(I)=3
      JV(I)=2
      IF (INDEX-4) 50,50,30
   30 IF (INDEX.NE.6) GO TO 40
      I=I+1
      JA(I)=3
      JV(I)=3
      JPOW=3
      CALL THST1
      THRNP=THRTL
      JPOW=5
      GO TO 50
   40 IF (INDEX.EQ.7) NVSTP=0
   50 NSOL=I

C     STORE VALUES OF INDEPENDENT VARIABLES (V), PERTURBATION
C          INCREMENTS (PV) AND TOLERANCES (T).

      DO 60 I=1,NSOL
      J=JV(I)
      V(I)=SV(J)
      PV(I)=SPV(J)
      J=JA(I)
   60 T(I)=ST(J)

C     START CONVERGENCE LOOP.

      IF (INDEX-2) 100,100,70
   70 AGT=GT
      AGN=GN-1.
      IF (INDEX.NE.6) ITRM=1
      IF (INDEX-4) 110,80,90
   80 AGN=0.0
      GO TO 110
   90 IF (INDEX.EQ.5) AGT=0.0
      GO TO 110
  100 AGT=0.0
      AGN=0.0
  110 IF (INDEX.NE.7) GO TO 120
      QS=RHOS*VTF*VTF
      QSC=QS*CBAR
      MACH=VTF/SOUND
  120 ALTHR=ALFAR-WINCR+THIR
      SINTH=SIN(ALTHR)
      COSTH=COS(ALTHR)
      CALL THST1
      CALL AERO2
      TMIND=TGROS*COSTH-DRAG-DWA
      IF (INDEX.NE.3) GO TO 130
      SNGAM=TMIND/WT-AGT
      GAMR=ASIN(SNGAM)
      GO TO 140
  130 SNGAM=SIN(GAMR)
  140 CSGAM=COS(GAMR)
```

```
      FLIFT=LIFT+TGROS*SINTH-WT*CSGAM
      AN=FLIFT*RECPM
      IF (INDEX-1) 150,150,170
  150 IF (FLIFT) 160,170,170
  160 FRICT=COEF*FLIFT
      GO TO 180
  170 FRICT=0.0
  180 FTHST=TMIND+FRICT-WT*SNGAM
      AT=FTHST*RECPM
      ACN=AN/GZ-AGN
      ACT=AT/GZ-AGT
      IF (IMOM.EQ.0) GO TO 200
      IF (ITRM.EQ.0) GO TO 190
      ANGLE=ALFAR-WINCR
      THETR=ANGLE+GAMR
      SINAL=SIN(ANGLE)
      COSAL=COS(ANGLE)
      U=VTF*COSAL
      W=VTF*SINAL
  190 UDOT=AN*SINAL+AT*COSAL-Q*W
      WDOT=AT*SINAL-AN*COSAL+Q*U
      ALPHD=(U*WDOT-W*UDOT)/VTSQ
      CMQ=-13.137
      CMALD=-11.94252*(.4008279-7.523412/(ALFAR*RTOD-25.84259)**2)
      A1=0.5*CBAR/VTF
      CMDMP=A1*(CMQ*Q+CMALD*ALPHD)
      QCDOT=(THMOM+(CMACG+CMDMP)*QSC)/IYY
  200 IF (ITRM.EQ.0) GO TO 300

C     STORE CURRENT VALUES OF ACCELERATIONS (A).

      DO 210 I=1,NSOL
      J=JA(I)
  210 A(I)=SA(J)
C     CONVERGE VALUES OF(A) BY VARYING VALUES OF (V).

      CALL TRIM(NSOL,A,V,PV,T,K)
      IF (K-NSOL) 230,230,220
  220 IF (INDEX.NE.7) 290,250
C     STORE RESET VALUES OF INDEPENDENT VARIABLES.
  230 NSTEP=NSTEP+1
      IF (NSTEP.GT.33) GO TO 280
C     RESET VALUES OF (V).
      DO 240 I=1,NSOL
      J=JV(I)
  240 SV(J)=V(I)
      GO TO 110
  250 IF (ABS(ACT).LE.ST(3)) GO TO 290
      IF (NVSTP.GT.0) GO TO 270
      IF (ACT.LT.0.0) GO TO 270
  260 VTF=VTF+DVTF
      NSTEP=0
      K=-1
      GO TO 110
  270 DVTF=-ACT*DVTF/(ACT-ASAV)
      ASAV=ACT
      NVSTP=NVSTP+1
      IF (NVSTP.GT.12) GO TO 280
      GO TO 260
  280 IERR=3
  290 ITRM=0
  300 QDOT=QCDOT
      RCF=VTF*SNGAM
      AXE=AT*CSGAM-AN*SNGAM
      AZE=AN*CSGAM+AT*SNGAM
      RETURN

      END
```

```
      SUBROUTINE AERO1
C     6400 FORTRAN IV PROGRAM TO STORE AND RETRIEVE AERODYNAMIC DATA
C     FOR USE WITH TABULATED AERODYNAMIC DATA
C     CL AND CD = F(ALPHA,THRUST COEFFICIENT,FLAP DEFLECTION)

C     THIS PROGRAM CALLS THE FOLLOWING SUBROUTINES,
C          KABD
C          LOOK

      INTEGER CARD,PRINT,PAGE
      REAL LIFT,KTOF,MACH,NU,NMTOF,IYY

      DIMENSION XC(15),YC(5),ZC(5),CLA(375),CDA(375)

      COMMON /LIST1/ LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,HD(60)
      COMMON /LIST2/ DELTD,ALFAR,THRTL,GAMR,HIND,VTF,WT,HF,DT,TIMS,XF,FU
     1EL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN
      COMMON /LIST3/ CL,CD,S,QS,THIR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHD,CX,
     1CZ,Q,IMOM,WINCR,CMDMP,QSC,ALMXR,VSF,BW
      COMMON /LIST4/ FFS,JPOW,TGROS,DWA,THMOM,THREQ
      COMMON /LIST5/ SIG,SOUND,NU,TEMR,PAMB,IATM,TEMF,DSODH,DRHO
      COMMON /LIST6/ RTOD,DTOR,KTOF,FTOK,NMTOF,FTONM,RHOZ,RHOZ2,GZ,PZ,TZ
      COMMON /LIST8/ INDEX,COEF,GN,GT,AN,AT,AX,AZ,ITRM,VUPPER
      COMMON/LIST9/ANS,ANS2,ND,X ,Y ,Z ,INDIC,IFLAG,NDER(3),DBDX,DBDY,
     * DBDZ,DB2DX,DB2DY,DB2DZ
      COMMON /LIST15/ IALMX,INT,XEDOT,ZEDOT,XE,ZE,ILOOP,DTIME,IAT,FLIFT,
     1DFLP,C4HT,GALMXR
      COMMON /LGEOM/ SHSW,SCR,QDOT,RTH,DTHTH,RWA,DTHWA,DSTHZ,DWLHZ,IYY,X
     1CG,ZCG,CBAR,PFN,SINTH,COSTH,ANGLE,SINAL,COSAL,RHZ,DTHHZ,VTSQ,UDOT,
     2WDOT,THETR,U,W
      COMMON/CONTROL/JFIG,IREV,ISP,NENG
      COMMON /LIST99/ IERR

C     DEFINITION OF VARIABLES IN NAMELIST 'AERT1'
C     S = WING REFERENCE AREA.
C     BW = WING SPAN
C     C4HT=     HEIGHT OF THE QUARTER CHORD ABOVE THE GROUND (FT)
C     GALMX=    ANGLE OF ATTACK FOR GROUND CONTACT (DEG)
C     ALPMX =   MAXIMUM ALLOWABLE ANGLE OF ATTACK - USUALLY A CL LIMIT
C               (DEG)
C     ACLMD = STALL ANGLE OF ATTACK (DEG)
C     FBP = EQUIVALENT FLAT PLATE AREA OF ADDITIONAL DRAG ITEMS (SQ FT)
C     WINCD = WING INCIDENCE(DEG)
C     DCLSP = INCREMENTAL LIFT COEFFICIENT DUE TO LIFT DUMPERS
C     DCDSP = INCREMENTAL DRAG COEFFICIENT DUE TO LIFT DUMPERS
C     EODCL = INCREMENTAL LIFT COEFFICIENT DUE TO ENG OUT CONTROLS
C     EODCD = INCREMENTAL DRAG COEFFICIENT DUE TO ENG OUT CONTROLS
C     NX,NY,NZ ARE THE NUMBER OF XC,YC,ZC
C     XC = THE LIST OF ANGLES OF ATTACK (DEG)
C     YC = THE LIST OF THRUST COEFFICIENTS
C     ZC = THE LIST OF FLAP DEFLECTIONS (DEG)
C     CLA = LIFT COEFFICIENT AS A FUNCTION OF (ALPHA,CT,FLAP DEFLECTION)
C     CDA = DRAG COEFFICIENT AS A FUNCTION OF (ALPHA,CT,FLAP DEFLECTION)
C     JFIG = 1 FOR MECHANICAL FLAPS PLUS VECTORED THRUST
C            2 FOR EXTERNALLY BLOWN FLAPS
```

```
C             3 FOR INTERNALLY BLOWN FLAPS

      NAMELIST /AERT1/ S,BW,JFIG,C4HT,GALMX,   ALPMX,ACLMD,FBP,WINCD,        PRE
     1                 NX,XC,NY,YC,NZ,ZC,CLA,CDA,DCLSP,DCDSP,EODCL,EODCD

C     READ INPUTS.

      IF (IN.NE.1) GO TO 20
      READ (CARD,AERT1)
      WRITE (PRINT,AERT1)
      WINCR=WINCD*DTOR
      GALMXR=GALMX*DTOR
      ALMXR=DTOR*ALPMX
      ALPHD = ACLMD*DTOR
      ND = 1
      II = 0
      CDBP = FBP/S                                                           PRE
      RETURN

   20 RHOS=RHOZ2*SIG*S
      IF (VTF       .LE.1.0)II=0
      QS = RHOS*VTF*VTF
      X = ACLMD
      IF (II.EQ.0) GO TO 21
      IF(JFIG.LE.2) GO TO 23
      Y=THMOM/QS
      GO TO 22
   23 Y = TGROS/QS
      GO TO 22
   21 Y = 0
   22 Z = DFLP
      CALL LOOK(NX,XC,NY,YC,NZ,ZC,CLA,CDA)
      CLMAX = ANS
      IF(NENG.EQ.2) CLMAX=CLMAX+EODCL
      IF (II.GT.0) RETURN
      II = 1
      VSF=SQRT(WT/(CLMAX*RHOS))
      RETURN

      ENTRY AERO2
   60 ALTHR = ALFAR-WINCR+THIR
      ALFAD = ALFAR*RTOD
      IF(VTF.LE.1.0.OR.IREV.GT.0) GO TO 61
      IF(JFIG.LE.2) GO TO 65
      CT = THMOM/QS
      GO TO 62
   65 CT=TGROS/QS
      GO TO 62
   61 CT = 0.00
   62 X = ALFAD
      Y = CT
      Z = DFLP
      CALL LOOK(NX,XC,NY,YC,NZ,ZC,CLA,CDA)
      IF(JFIG.EQ.2.AND.IREV.EQ.0) GO TO 72
      IF(VTF.LE.1..AND.JFIG.EQ.3.AND.IREV.EQ.0) GO TO
      CL = ANS
      CD = ANS2
      IF(ISP.GT.0) GO TO 66
      IF(NENG.NE.2) GO TO 64
      CL = CL+EODCL
      CD = CD+EODCD
```

```
      GO TO 64
   66 CL = CL+DCLSP
      CD = CD+DCDSP
   64 CD = CD+CDBP
      LIFT=CL*QS
      DRAG=CD*QS
      RETURN
   63 LIFT = THMOM*SIN(ALFAR+DFLP*DTOR)
      DRAG = THMOM*COS(ALFAR+DFLP*DTOR)
      RETURN
   72 CL = ANS-CT*SIN(ALTHR)
      IF (VTF     .LE.1.0) GO TO 73
      CD = ANS2+CT*COS(ALTHR)
      GO TO 74
   73 CD = ANS2+CT*COS(ALTHR)
   74 CD = CD+CDBP
      IF (ISP.GT.0) GO TO 75
      IF (NENG.NE.2) GO TO 80
      CL = CL+EODCL
      CD = CD+EODCD
      GO TO 80
   75 CL = CL+DCLSP
      CD = CD+DCDSP
   80 LIFT=CL*QS
      DRAG=CD*QS
      RETURN
      END
```

```
FUNCTIONALIN(I,X,Y,XA)

DIMENSION X(1),Y(1)

ALIN = Y(I)+(XA-X(I))*(Y(I+1)-Y(I))/(X(I+1)-X(I))
RETURN

END
```

```
      SUBROUTINE ATMOS
C     6400 FORTRAN ATMOSPHERIC SUBROUTINE WITH MIL STD 210A TEMPERATURE OPTIONS.

C     THIS SUBROUTINE CALLS FOR THE FOLLOWING ENTRIES.
C          KABD,DKABD
C     INDEX IATM, SET IN THE MAIN PROGRAM, SELECTS THE TEMPERATURE OPTION.
C     IATM = 0, FOR STANDARD TEMPERATURES
C          IATM = 1 MIL STD 210A TROPIC TEMPERATURES.
C          IATM = 2 MIL STD 210A POLAR TEMPERATURES.
C          IATM = 3 MIL STD 210A HOT TEMPERATURES.
C          IATM = 4 MIL STD 210A COLD TEMPERATURES.
C          IATM = 5  TEMPERATURE IS INPUT AS TEMF

      INTEGER CARD,PRINT,PAGE
      REAL LIFT,NU,MACH

      DIMENSION TROPIC(15),POLAR(20),HOT(20),COLD(35)

      COMMON /LIST1/LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,HD(60)
      COMMON /LIST2/DELTD,ALFAR,THRTL,GAMR,HIND,VTF,WT,HF,DT,TIMS,XF,
     * FUEL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN
      COMMON /LIST5/SIG,SOUND,NU,TEMR,PAMB,IATM,TEMF,DSODH,DRHO
      DATA  NTROPC,NPOLAR,NHOT,NCOLD/3,4,4,7/,(TROPIC(I),I=1,15)/
     1 -84270.12,62808.804,548.14631,-.0039108943,53595.,-215803.01,
     2 94397.625,233.43864,.0020330332,69620.,453.86956,67598.227,
     3 289.9406,0.0013433339,100000./,(POLAR(I),I=1,20)/220.31153,
     4 -1058.4386, 443.77985, 0.0031011192, 3243., 1623.9492, 15111.866,
     5 455.6985, -5.1608824E-04, 9882., 15029.377, 39599.146, 478.47166,
     6 -0.002800841, 30065., 3863.7771, 13319.676, 400.08921,
     7 -2.5384826E-04, 86092./,(HOT(I),I=1,20)/-196560.19, 55620.773,
     8 559.15407, -0.0039742099, 39400.,-109942.52, 73731.986,
     9 398.98426, 3.172946E-04, 50400.,-123.23365, 47876.697, 410.41972,
     * 1.8486342E-04, 66400., 10388998., -63820.346, 260.37596,
     1 0.0012429502, 100000./,(COLD(I),I=1,35)/-48393.349, 10737.404,
     2 395.18101, 0.012984173, 3311., 0.0, 1.0, 444.688, 0.0, 10744.,
     3 627229540., 381783.16, 2121.0792, 0.0013101285, 30715., 0.0, 1.0,
     4 374.688, 0.0, 42377.,-1609.4722, 51071.814, 597.11641,
     5 -0.0052531685, 50583., 0.0, 1.0, 334.688, 0.0, 61087.,
     6 -50710058., -31326.366, 1046.3219, -0.0026667396, 73055./

      NAMELIST/ATM1/NTROPC ,TROPIC,NPOLAR,POLAR,NHOT,HOT,NCOLD,COLD

      IF(IN-1)5,1,5
    1 WRITE(PRINT,ATM1)
      RETURN

C     STANDARD ATMOSPHERE

    5 IF(HF.GE.36089.239)GO TO 10
      PAMB = 2116.2218*(1.-.68755856E-05*HF)**5.2558761
      IF(IATM.GT.0)GO TO 15
      TEMR = 1.8*(288.15-.0019812*HF)
      GO TO 30
   10 IF(HF.GE.65616.798)GO TO 11
      PAMB = 472.68164*EXP(1.7345725-.48063428E-04*HF)
      IF(IATM.GT.0)GO TO 15
```

```
      TEMR = 389.97
      GO TO 30
   11 PAMB = 114.3457/(.9076852+.14068774E-05*HF)**34.163194
      IF(IATM.GT.0)GO TO 15
      TEMR = 1.8*(196.65+.0003048*HF)
      GO TO 30
   15 GO TO(16,1000,2000,95,26),IATM

C     TROPICAL ATMOSPHERE TEMPERATURES.

   16 IF(HF.GT.100000.)GO TO 25
      CALL KABD(NTROPC,HF,TEMR,TROPIC)
      GO TO 30
   25 TEMR = 424.27
      GO TO 30

C     POLAR ATMOSPHERE TEMPERATURES.

 1000 IF(HF.GE.86092.)GO TO 1130
      CALL KABD(NPOLAR,HF,TEMR,POLAR)
      GO TO 30
 1130 TEMR = 378.27
      GO TO 30

C     HOT ATMOSPHERE TEMPERATURES.

 2000 IF(HF.GT.100000.)GO TO 2120
      CALL KABD(NHOT,HF,TEMR,HOT)
      GO TO 30
 2120 TEMR = 448.07
      GO TO 30

C     COLD ATMOSPHERE TEMPERATURES.

   95 IF(HF.GT.100000.)GO TO 170
      CALL KABD(NCOLD,HF,TEMR,COLD)
      GO TO 30
  170 TEMR = 355.77
      GO TO 30
   26 TEMR = TEMF+459.67
   30 SIG = .24509246*PAMB/TEMR
      TEMF = TEMR-459.67
      TEMK=TEMR/1.8
      SOUND=65.770350*SQRT(TEMK)
      NU = .1281125E-04*SQRT(TEMK)/(SIG*(1.+110.4/TEMK))
      RETURN
      END
```

```
      SUBROUTINE FIND(N,X,Y,XC,YC)
C     CDC 6400 SUBROUTINE WHICH PERFORMS A LAGRANIAN INTERPOLATION
C     ON A ONE DIMENSIONAL ARRAY.  IF THE ARGUMENT IS OUTSIDE OF THE
C     RANGE, A LINEAR EXTRAPOLATION IS MADE.

C     THIS SUBROUTINE CALLS THE FOLLOWING ENTRIES
C             INTP2
C
C     N-      THE NUMBER OF X,Y POINTS IN THE ARRAY
C     X-      THE ARGUMENT FOR INTERPOLATING
C     Y-      THE SOLUTION
C     XC-     THE ARGUMENT LIST
C     YC-     THE DEPENDENT VARIABLE LIST

      DIMENSION XC(20),YC(20)

      INTEGER CARD,PRINT,PAGE

      COMMON/LIST1/LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,HD(60)

      IF(N.LT.4) GO TO 50
      I=1
      NI=N-1
      DO 10 I=1,NI
      IF(X.GE.XC(I).AND.X.LE.XC(I+1)) GO TO 15
   10 CONTINUE
      GO TO 50
   15 IF(X.NE.XC(I)) GO TO 20
      Y=YC(I)
      RETURN
   20 IF(X.NE.XC(I+1)) GO TO 25
      Y=YC(I+1)
      RETURN
   25 IF(I.GT.2) GO TO 30
      I=1
      GO TO 45
   30 IF(I.LT.N-1) GO TO 35
      I=N-3
      GO TO 45
   35 I=I-1
   45 NX=I+3
      CALL INTP2(NX,XC,YC,X,Y,A,B,C,D)
      RETURN
   50 IF(X.LT.XC(1)) I=1
      IF(X.GT.XC(N)) I=NI
      Y = ALIN(I,XC,YC,X)
      RETURN
      END
```

```
      SUBROUTINE GILL(YDOT,Y,DUMQ,ILOOP,DTIME)                          U
C                                                                       
C     THIS IS AN INTEGRATION SUBROUTINE.  TO EXECUTE AN INTEGRTATION STE U
C     REQUIRES FOUR PASSES WITH ILOOP SET TO 2 THROUGH 5. THE FIRST TIME U
C     SUBROUTINE IS USED, THE FIRST PASS IS MADE WITH ILOOP SET EQUAL TO U
C     ALL VARIABLES BEING INTEGRATED MUST BE CALLED IN EACH PASS BEFORE  U
C     GOING ON TO THE NEXT PASS.                                         U
                                                                         U
      DY=YDOT*DTIME                                                      U
      Q=DUMQ                                                             U
      GO TO (10,20,30,40,50), ILOOP                                      U
   10 Q=0.                                                               U
      C=1./SQRT(2.)                                                      U
      CMINUS=1.-C                                                        U
      CPLUS=1.+C                                                         U
   20 R=.5*DY-Q                                                          U
      Q=3.*R-.5*DY+Q                                                     U
      GO TO 60                                                           U
   30 R=CMINUS*(DY-Q)                                                    U
      Q=Q+3.*R-CMINUS*DY                                                 U
      GO TO 60                                                           U
   40 R=CPLUS*(DY-Q)                                                     U
      Q=Q+3.*R-CPLUS*DY                                                  U
      GO TO 60                                                           U
   50 R=(DY-2.*Q)/6.                                                     U
      Q=Q+3.*R-.5*DY                                                     U
   60 Y=Y+R                                                              U
      DUMQ=Q                                                             U
      RETURN                                                             U
      END                                                                U
```

```
      SUBROUTINE GINTG                                                  AI
C     6400 FORTRAN SUBROUTINE WHICH INTEGRATES ALTITUDE,R/C AND ATTITUDE AI
C               AND/OR GROUND SPEED, DISTANCE                            AI
C               OR GROUND SPEED, DISTANCE AND ATTITUDE                   AI
                                                                        AI
C     THIS PROGRAM CALLS THE FOLLOWING SUBROUTINES,                     AI
C          AEQFM,ATMOS,GILL                                             AI
                                                                        AI
      INTEGER CARD,PRINT,PAGE                                           AI
      REAL LIFT,KTOF,MACH,NU,NMTOF,IYY                                  AI

      COMMON /LIST1/ LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,HD(60)  AI
      COMMON /LIST2/ DELTD,ALFAR,THRTL,GAMR,HIND,VTF,WT,HF,DT,TIMS,XF,FU AI
     1EL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN                      AI
      COMMON /LIST3/ CL,CD,S,QS,THIR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHD,CX, AI
     1CZ,Q,IMOM,WINCR,CMDMP,QSC,ALMXR,VSF,BW                            AI
      COMMON /LIST4/ FFS,JPOW,TGROS,DWA,THMOM,THREQ                     AI
      COMMON /LIST5/ SIG,SOUND,NU,TEMR,PAMB,IATM,TEMF,DSODH,DRHO        AI
      COMMON /LIST6/ RTOD,DTOR,KTOF,FTOK,NMTOF,FTONM,RHOZ,RHOZ2,GZ,PZ,TZ AI
      COMMON /LIST8/ INDEX,COEF,GN,GT,AN,AT,AXE,AZE,ITRM,VUPPER         AI
      COMMON /LIST15/ IALMX,INT,XEDOT,ZEDOT,XE,ZE,ILOOP,DTIME,IAT,FLIFT, AI
     1DFLP,C4HT,GALMXR
      COMMON /LGEOM/ SHSW,SCR,QDOT,RTH,DTHTH,RWA,DTHWA,DSTHZ,DWLHZ,IYY,X AI
     1CG,ZCG,CBAR,PFN,SINTH,COSTH,ANGLE,SINAL,COSAL,RHZ,DTHHZ,VTSQ,UDOT, AI
     2WDOT,THETR,U,W                                                    AI
                                                                        AI
   10 IF (XEDOT) 30,20,30                                               AI
   20 GAMR=0.0                                                          AI
      GO TO 40                                                          AI
   30 GAMR=ATAN(ZEDOT/XEDOT)                                            AI
   40 VTSQ=XEDOT*XEDOT+ZEDOT*ZEDOT                                      AI
      VTF=SQRT(VTSQ)                                                    AI
      ALFAR=THETR-GAMR+WINCR                                            AI
      IF (IALMX) 60,60,50                                               AI
   50 ALFAR=ALMXR                                                       AI
      Q=(XEDOT*ZE2DT-ZEDOT*XE2DT)/VTSQ                                  AI
   60 CALL ATMOS                                                        AI
      MACH=VTF/SOUND                                                    AI
      QS=RHOZ2*SIG*S*VTSQ                                               AI
      CALL AEQFM                                                        AI
      XE2DT=AXE                                                         AI
      ZE2DT=AZE                                                         AI
      GO TO (70,70,80,70,80), ILOOP                                     AI
   70 TIMS=TIMS+.5*DTIME                                                AI
   80 XGDOT=XEDOT-VWF                                                   AI
      CALL GILL(XGDOT,XE,QXE,ILOOP,DTIME)                               AI
      CALL GILL(XE2DT,XEDOT,QXE2,ILOOP,DTIME)                           AI
      IF (INT-2) 110,100,90                                             AI
   90 CALL GILL(ZEDOT,ZE,QZE,ILOOP,DTIME)                               AI
      CALL GILL(ZE2DT,ZEDOT,QZE2,ILOOP,DTIME)                           AI
  100 CALL GILL(Q,THETR,QTH,ILOOP,DTIME)                                AI
  110 GO TO (120,130,130,130,140), ILOOP                                AI
  120 ILOOP=3                                                           AI
      GO TO 10                                                          AI
  130 ILOOP=ILOOP+1                                                     AI
      GO TO 10                                                          AI
  140 ILOOP=2                                                           AI
      RETURN                                                            AI

      END                                                               AI
```

```
      SUBROUTINE HEAD(NL)                                               AK
                                                                        AK
      INTEGER CARD,PRINT,PAGE                                           AK

      DIMENSION ID(10),HP(10),NL(8)                                     AK

      COMMON /LIST1/ LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,HD(60) AK
                                                                        AK
      DO 20 I=1,4                                                       AK
      J=2*I-1                                                           AK
      IF (MOD(I,2).NE.0) WRITE (PRINT,30)                               AK
   30 FORMAT (1H )                                                      AK
      DECODE (10,40,NL(J)) ID                                           AK
   40 FORMAT (5I2)                                                      AK
      DO 10 K=1,10                                                      AK
      KK=ID(K)                                                          AK
   10 HP(K)=HD(KK)                                                      AK
      WRITE (PRINT,50) HP                                               AK
   50 FORMAT (12X,10(2X,A10))                                           AK
   20 CONTINUE                                                          AK
      LINE=LINE+6                                                       AK
      RETURN                                                            AK
      END                                                               AK
```

```
      SUBROUTINE INTP2(NX,X,Y,XC,YC,A,B,C,D)

C     6400 FORTRAN 3RD DEGREE POLYNOMIAL INTERPOLATION SUBROUTINE.
C     ENTRY WITH SUBSCRIPTED X AND Y DATA.

      DIMENSION X(1), Y(1)

      I = NX-3
   10 AL = (Y(I)-Y(I+1))/(X(I)-X(I+1))
      BE = (AL-(Y(I)-Y(I+2))/(X(I)-X(I+2)))/(X(I+1)-X(I+2))
      A = (BE-(AL-(Y(I)-Y(I+3))/(X(I)-X(I+3)))/(X(I+1)-X(I+3)))/(X(I+2)-
     *X(I+3))
      GA = X(I)+X(I+1)
      B = BE-A*(GA+X(I+2))
      C = AL-A*(X(I)*X(I)+X(I+1)*GA)-B*GA
      D = Y(I)-((A*X(I)+B)*X(I)+C)*X(I)
      YC = ((A*XC+B)*XC+C)*XC+D
      RETURN
      END
```

```
      SUBROUTINE KABD(NFITS,XC,YC,X)                                          X
C     6400 FORTRAN KABD EVALUATION PROGRAM.                                   X
C     NFITS IS THE NUMBER OF CURVE FITS STRUNG TOGETHER FOR THE               X
C     PARTICULAR VARIABLE.                                                    X
                                                                              X
      INTEGER CARD,PRINT,PAGE                                                 X
      COMMON /LIST1/ LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,HD(60)       X
      COMMON /LIST99/ IERR                                                    X
      DIMENSION X(50)                                                         X
      IERR=0                                                                  X
      I=1                                                                     X
   10 IF (XC-X(I+4)) 20,20,30                                                 X
   20 YC=X(I)/(XC-X(I+1))+X(I+2)+X(I+3)*XC                                    X
      RETURN                                                                  X
   30 I=I+5                                                                   X
      IF (I.LE.5*NFITS) GO TO 10                                              X
      WRITE (PRINT,40) NFITS,XC,X(I-1)                                        X
      IERR=1                                                                  X
      RETURN                                                                  X
C                                                                             X
   40 FORMAT (5X,35H UPPER LIMIT OF X EXCEEDED.  NFITS=,I3,5H, XC=,E12.5      X
     1,9H, X(I+4)=,E12.5)                                                     X
      END                                                                     X
```

```
      SUBROUTINE LAGRA(X,Y,INDEX)                                             Y
C     6400 FORTRAN SUBROUTINE WHICH DOES A FOUR POINT LAGRANGIAN INTERPO      Y
                                                                              Y
      DIMENSION X(5),Y(5),DX(10)                                              Y
                                                                              Y
      IF (INDEX) 30,10,30                                                     Y
   10 INDEX=INDEX+1                                                           Y
      K=0                                                                     Y
      DO 20 I=1,4                                                             Y
      II=I+1                                                                  Y
      DO 20 J=II,5                                                            Y
      K=K+1                                                                   Y
      DX(K)=X(I)-X(J)                                                         Y
   20 CONTINUE                                                                Y
      PROD2=DX(2)/DX(5)*DX(3)/DX(6)*DX(4)/DX(7)                               Y
      PROD3=-DX(1)/DX(5)*DX(3)/DX(8)*DX(4)/DX(9)                              Y
      PROD4=DX(1)/DX(6)*DX(2)/DX(8)*DX(4)/DX(10)                              Y
      PROD5=-DX(1)/DX(7)*DX(2)/DX(9)*DX(3)/DX(10)                             Y
   30 Y(1)=Y(2)*PROD2+Y(3)*PROD3+Y(4)*PROD4+Y(5)*PROD5                        Y
      RETURN                                                                  Y
      END                                                                     Y
```

```
      SUBROUTINE LANDING
C     6400 FORTRAN IV LANDING SUBROUTINE FOR STOL AIRCRAFT
C     THE BOUNDARIES OF THIS PROGRAM ARE -
C        VTD GT OR EQ TO APV*VSTALL
C        VOBS GT OR EQ TO APV*VSTALL
C        TOUCHDOWN RATE OF SINK LT OR EQ TO RS
C        THETA AT TOUCHDOWN GT OR EQ TO ZERO
C        THETA AT TOUCHDOWN LT OR EQ TO GALMX (AERO INPUT)
C        ALPHA DURING APPROACH LT OR EQ TO ALPMX (AERO INPUT)
C     THIS PROGRAM CALL THE FOLLOWING SUBROUTINES
C           ATMOS,SKIP,HEAD,THST1,NWTP2,AERO1,GINTG
      INTEGER CARD,PRINT,PAGE
      REAL LIFT,KTOF,MACH,NU,NMTOF
      DIMENSION NL(8)
      COMMON /LIST1/ LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,HD(60)     F
      COMMON /LIST2/ DELTD,ALFAR,THRTL,GAMR,HIND,VTF,WT,HF,DT,TIMS,XF,FU
     1EL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN
      COMMON /LIST3/ CL,CD,S,QS,THIR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHD,CX,
     1CZ,Q,IMOM,WINCR,CMDMP,QSC,ALMXR,VSF,BW
      COMMON /LIST4/ FFS,JPOW,TGROS,DWA,THMOM,THREQ
      COMMON /LIST5/ SIG,SOUND,NU,TEMR,PAMB,IATM,TEMF,DSODH,DRHO
      COMMON /LIST6/ RTOD,DTOR,KTOF,FTOK,NMTOF,FTONM,RHOZ,RHOZ2,GZ,PZ,TZ
      COMMON /LIST8/ INDEX,COEF,GN,GT,AN,AT,AX,AZ,ITRM,VUPPER
      COMMON /LIST15/ IALMX,INT,XEDOT,ZEDOT,XE,ZE,ILOOP,DTIME,IAT,FLIFT,
     1DFLP,C4HT,GALMXR
      COMMON /LGEOM/ SHSW,SCR,QDOT,RTH,DTHTH,RWA,DTHWA,DSTHZ,DWLHZ,IYY,X
     1CG,ZCG,CBAR,PFN,SINTH,COSTH,ANGLE,SINAL,COSAL,RHZ,DTHHZ,VTSQ,UDOT,
     2WDOT,THETR,U,W
      COMMON/CONTROL/JFIG,IREV,ISP,NENG
      COMMON /LIST99/ IERR
      DATA DFLP,ROTPT,HFOB,ROTATN,TBRK,TSP,TREV,APR,RCOEF,BCOEF,DGTH,
     1DGTD/60.,1.,50.,8.,2.,2.,2.,1.1,0.1,0.3,0.3,0.15/
      DATA GAMMA,RS/-7.5,10./
      DATA (NL(I),I=1,8)/10H0556370226,10H3323082052,10H3938514542,10H38
     140015151,10H3610171403,10H3431091158,10H5739393838,10H5940015138/
C     DEFINITION OF VARIABLES IN NAMELIST 'LAND1'
C     DFLP=     FLAP SETTING (DEG)
C     HFOB=     OBSTACLE HEIGHT (FT)
C     ROTATN=   ROTATION RATE (DEG/SEC)
C     VMCGK=    MINIMUM CONTROL SPEED ON THE GROUND (KTS)
C     VMCAK=    MINIMUM CONTROL SPEED IN THE AIR (KTS)
C     TBRK=     TIME DELAY AFTER TOUCHDOWN FOR BRAKE APPLICATION (SEC)
C     TSP=      TIME DELAY AFTER TOUCHDOWN FOR SPOILER DEPLOYMENT (SEC)
C     TREV=     TIME DELAY AFTER TOUCHDOWN FOR THRUST REVERSAL (SEC)
C     APR=      RATIO OF APPROACH SPEED TO MINIMUM CONTROL SPEED
C     DGTH=     INCREMENTAL LOAD FACTOR AVAILABLE AT THE OBSTACLE
C     DGTD=     INCREMENTAL LOAD FACTOR AVAILABLE AT TOUCHDOWN
C     RCOEF=    ROLLING COEFFICIENT OF FRICTION
```

```
C      BCOEF=   BRAKING COEFFICIENT OF FRICTION
C      GAMMA=   INITIAL FLIGHT PATH ANGLE (DEG) (NEGATIVE IS DESCENDING)
C      RS=      MAXIMUM RATE OF SINK AT TOUCHDOWN (FT/SEC)
C               (POSITIVE IS DESCENDING)
C      ROTPT=   EQUAL TO ZERO SUPRESSES PRINTING OF THE ENTIRE TIME HISTORY

C      THE FOLLOWING VARIABLES ARE ENTERED AT TIME OF LOADING AND ARE USED
C      UNTIL OVERRIDDEN BY READING THE APPROPRIATE VARIABLES IN LAND1
C         DFLP = 60.0
C         HFOB = 50.0
C         ROTATN = 8.0
C         TBRK = 2.0
C         TSP = 2.0
C         TREV = 2.0
C         APR = 1.10
C         DGTH = 0.30
C         DGTD = 0.15
C         RCOEF = 0.10
C         BCOEF = 0.30
C         GAMMA = -7.50
C         RS = 10.0
C         ROTPT = 1.0

      NAMELIST/LAND1/DFLP,ROTPT,HFOB,ROTATN,VMCGK,TBRK,TSP,APR,RCOEF,BCO
     1EF,TREV,GAMMA,RS,VMCAK,DGTH,DGTD

      IF (IN.NE.1) GO TO 5
      READ(CARD,LAND1)
      WRITE(PRINT,LAND1)
      RETURN

    5 HFRUN=HF
      HF=HFRUN+C4HT+50.
      CALL ATMOS
      RHOS=RHOZ2*SIG*S
      VMCG = VMCGK*KTOF
      VMCA=VMCAK*KTOF
      APV=APR
      COEF = 0.00
      ISP=0
      IREV=0
      JSP = 0
      JREV = 0
      JBRK = 0
      IALMX = 0
      WRITE (PRINT,1000)
 1000 FORMAT(1H1,3X,*OUTPUT DEFINITIONS - LANDING*,/,4X,*SEGMENT = *,
     1*1  DETERMINATION OF ANGLE OF ATTACK AT TOUCHDOWN*,/,14X,
     2*2  R/C AVAILABLE AT TOUCHDOWN - 1 ENGINE OUT*,/,14X,
     3*3  DETERMINATION OF ANGLE OF ATTACK AT THE OBSTACLE*,/,14X,
     4*4  R/C AVAILABLE AT THE OBSTACLE - 1 ENGINE OUT*,/,14X,
     5*5  INTEGRATION FROM OBSTACLE TO TOUCHDOWN*,/,14X,
     6*6  TOUCHDOWN TO NOSEDOWN INTEGRATION*,/,14X,
     7*7  CONDITIONS AT NOSEDOWN*,/,14X,
     8*8  NOSEDOWN TO STOP INTEGRATION*,/,14X,
     9*9  CONDITIONS AT STOP*,/)
      LINE = LIMIT+1
      GO TO 50

C      OUTPUT BLOCK
```

```
   20 IF (ROTPT.NE.0.) GO TO 25
      IF((ISEG.EQ.6).OR.(ISEG.EQ.8)) GO TO 35
   25 IF (LINE.LT.LIMIT) GO TO 30
      CALL SKIP
      CALL HEAD(NL)
      WRITE (PRINT,1001) HD(27)
 1001 FORMAT (1H+,1X,A10)
   30 VTK=VTF*FTOK
      GAMD=GAMR*RTOD
      RCM=60.*VTF*SIN(GAMR)
      THETD=THETR*RTOD
      THDTD=Q*RTOD
      ALFAD=ALFAR*RTOD
      ALTHD=ALTHR*RTOD
      WRITE (PRINT,1002) ISEG,HF,DFLP,WT,VTK,RCM,THETD,AN,CL,LIFT,TGROS,
     1TIMS,XE,ZE,GAMD,ALFAD,THDTD,AT,CD,DRAG,ALTHD
      LINE=LINE+3
 1002 FORMAT (1H0,I11,6F12.3,2F12.5,2F12.3,/,12X,6F12.3,2F12.5,2F12.3)
      IF (THRTL.GT.1.0) GO TO 40
      IF (THRTL.LT.0.0) GO TO 45
   35 GO TO (125,150,175,200,250,260,300,330,350), ISEG
   40 IF(ISEG.NE.1.OR.ISEG.NE.3.OR.ISEG.NE.5) GO TO 35
      WRITE (PRINT,1008)
 1008 FORMAT (1H ,13X,*THROTTLE GREATER THAN MAXIMUM POWER*)
      GO TO 35
   45 IF(ISEG.NE.1.OR.ISEG.NE.3.OR.ISEG.NE.5) GO TO 35
      WRITE(PRINT,1009)
 1009 FORMAT(1H ,13X,*THROTTLE LESS THAN IDLE POWER*)
      GO TO 35

C     SET APPROACH SPEED - FACTOR TIMES STALL SPEED (POWER ON 1 ENG OUT)

   50 IN=0
      THRTL=1.0
      CALL AERO1
      ALTMR=ALPHD-WINCR+THIR
      VTF1=VSF
      ENGNO=ENGNO-1.
      NENG=2
      MACH=VTF1/SOUND
      VTF=VTF1
      NSTEP=0
   55 CALL THST1
      CALL AERO1
      IF(JFIG.EQ.2.AND.VTF.GT.1.) CLMAX=CLMAX-((TGROS*SIN(ALTMR))/(RHOS*
     1VTF*VTF))
      VTF=SQRT((WT-TGROS*SIN(ALTMR))/(RHOS*CLMAX))
      YVTF=(VTF1-VTF)/VTF
      IF (ABS(YVTF).LE.0.0001) GO TO 65
      IF (NSTEP.EQ.0) DMAC= MACH-VTF/SOUND
      CALL NWRP2(DMAC,YVTF,MACH,YSAV,XSAV)
      NSTEP=NSTEP+1
      IF (NSTEP.GT.15) GO TO 60
      VTF1=MACH*SOUND
      VTF=VTF1
      GO TO 55
   60 VTF=MACH*SOUND
   65 VSF=VTF
      ENGNO=ENGNO+1.
      NENG=0
      IN=2
```

```
      VSK=VSF*FTOK
      VAPF=VMCA*APV
      IF(VAPF.LT.VMCG*APV) VAPF=VMCG*APV
      IF(VAPF.LT.VSF*APV) VAPF=VSF*APV
      IF(VAPF.LT.VSF*SQRT(1.+DGTH)) VAPF=VSF*SQRT(1.+DGTH)
      IF(VAPF.LT.VSF*SQRT(1.+DGTD)) VAPF=VSF*SQRT(1.+DGTD)
      APV = VAPF/VSF
      IAPV=0
      IGAM = 0
   70 VTF=VAPF
      VAPK=VAPF*FTOK

C     DETERMINE ANGLE OF ATTACK AT TOUCHDOWN

  100 VTF=VAPF
      TIMS = 0.00
      XE=0.0
      ZE=0.0
      Q=0.0
      INTEG=0
      HF=HFRUN+C4HT
      IF (IGAM.EQ.0.OR.GAMR.LT.GAMMA*DTOR) GAMR=GAMMA*DTOR
      CALL ATMOS
      RHOS=RHOZ2*SIG*S
      QS=RHOS*VTF*VTF
      MACH=VTF/SOUND
      ITRM = 1
      GN=1.0
      GT=0.0
      INDEX=6
      COEF=0.0
      CALL AEQFM
      THETR=ALFAR+GAMR-WINCR
      IF (ALFAR.LE.ALMXR) GO TO 110

C     INCREASE VELOCITY TO DECREASE APPROACH ANGLE OF ATTACK

  105 IAPV=IAPV+1
      APV1=APV+FLOAT(IAPV)/100.
      VAPF=VSF*APV1
      WRITE (PRINT,1003) APV1
 1003 FORMAT(1H ,13X,16HVAP INCREASED TO,F7.3,12HTIMES VSTALL,/)
      LINE = LINE+2
      GO TO 70

C     DECREASE GAMR TO DECREASE R/S TO MAXIMUM INPUT VALUE

  110 IF (VTF*SIN(GAMR).GE.-(RS*1.01)) GO TO 115
      GAMR=ASIN(-RS/VTF)
      IGAM=IGAM+1
      IF (IGAM.GT.20) GO TO 405
      GAMD = GAMR*RTOD
      WRITE (PRINT,1004) GAMD
 1004 FORMAT(1H ,13X,*MAXIMUM RATE-OF-SINK EXCEEDED.  GAMMA RESET TO*,
     1F7.3,* DEG*)
      LINE = LINE+2
      GO TO 100

C     CHECK THETA TO AVOID 'TAIL HIT' OR 'NOSEWHEEL FIRST' LANDING

  115 IF (THETR.GT.GALMXR-WINCR)GO TO 105
```

```
      IF(THETR.GE.-0.001 ) GO TO 120
      GAMR=WINCR-ALFAR
      IGAM=IGAM+1
      IF (IGAM.GT.20) GO TO 405
      IF(GAMR.LT.0.0.AND.VTF*SIN(GAMR).GE.-RS) GO TO 100
      GO TO 400
  120 ISEG = 1
      GAMTR = GAMR
      GO TO 20

C     DETERMINE R/C AVAILABLE AT TOUCHDOWN

  125 THRTL = 1.0
      INDEX = 3
      ENGNO = ENGNO-1.0
      NENG=2
      CALL AEQFM
      ENGNO = ENGNO+1.0
      NENG=0
      THETR = ALFAR-WINCR+GAMR
      ISEG = 2
      GO TO 20

C     DETERMINE ANGLE OF ATTACK AT THE OBSTACLE

  150 HF=HFRUN+C4HT+HFOB
      ZE = HFOB
      GAMR = GAMTR
      CALL ATMOS
      RHOS=RHOZ2*SIG*S
      QS=RHOS*VTF*VTF
      MACH=VTF/SOUND
      GN=1.0
      GT=0.0
      INDEX = 6
      ITRM = 1
      COEF=0.0
      CALL AEQFM
      THETR=ALFAR+GAMR-WINCR
      IF (ALFAR.GT.ALMXR) GO TO 105
      ISEG=3
      VOBS=VTF
      GO TO 20

C     DETERMINE R/C AVAILABLE AT THE OBSTACLE

  175 THRTL = 1.0
      INDEX = 3
      ENGNO = ENGNO-1.0
      NENG=2
      CALL AEQFM
      ENGNO = ENGNO+1.0
      NENG=0
      THETR = ALFAR-WINCR+GAMR
      ISEG = 4
      GO TO 20

C     INTEGRATION FROM OBSTACLE TO TOUCHDOWN (DUE TO THE NO FLARE
C     LANDING TECHNIQUE USED IN THIS PROGRAM THE INTEGRATION IS MADE
C     IN ONE STEP)
```

```
  200 GAMR = GAMTR
      IF(GAMR.GE.0.0) GO TO 400
      IF (THETR.LT.-0.001) GO TO 405
      TIME1=HFOB/(VAPF*SIN(ABS(GAMR)))
      DIST1= TIME1*VAPF*COS(GAMR)
      TIMS=TIME1
      XE=DIST1
      ZE=0.0
      HF = HFRUN+C4HT
      CALL ATMOS
      RHOS=RHOZ2*SIG*S
      QS=RHOS*VTF*VTF
      ITRM = 1
      GN=1.0
      GT=0.0
      INDEX=6
      COEF=0.0
      CALL AEQFM
      THETR=ALFAR-WINCR+GAMR
      ISEG = 5
      GO TO 20

C     INTEGRATION FROM TOUCHDOWN TO NOSEDOWN

  250 TIMS=0.0
      THRTL=0.00
      XE=0.0
      ZF=0.0
      XEDOT=VAPF*COS(GAMR)
      ZEDOT=0.0
      GAMR=0.0
      ALFAR=THETR+WINCR
      INDEX=1
      INT=2
      Q=-ROTATN*DTOR
      COEF=RCOEF
      DTIME=0.2
      ILOOP=1
      VTF=XEDOT
  255 VTF1=VTF
      CALL GINTG
      ISEG = 6
      GO TO 20

  260 IF (ALFAR-WINCR) 265,270,255
  265 DT=(WINCR-ALFAR)/Q
      ALFAR=WINCR
      VNDF= VTF-(VTF-VTF1)/DTIME*DT
      XE=XE+(0.5*(VTF+VNDF)-VWF)*DT
      VTF=VNDF
      TIMS=TIMS+DT
  270 TIME2=TIMS
      DIST2=XE
      ISEG = 7
      INDEX=1
      QS=RHOS*VTF*VTF
      MACH=VTF/SOUND
      CALL AEQFM
      THETR=0.0
      GO TO 20
```

```
C     INTEGRATE FROM NOSEDOWN TO STOP

  300 INT=1
      DTIME=0.2
      TIMS=0.0
      XF=0.0
      ZF=0.0
      THETR=0.0
      ZEDOT=0.0
      Q=0.0
      XFDOT=VNDF
      ILOOP=1
  305 VTF1=VTF
      IF (TIMS+TIME2.GE.TSP) ISP=1
      IF (TIMS+TIME2.GE.TBRK) COEF=BCOEF
      IF (TIMS+TIME2.GE.TREV) IREV=1
      IF(ISP.EQ.1.AND.JSP.EQ.1) GO TO 315
      IF (ISP.EQ.0) GO TO 315
      JSP=1
      WRITE (PRINT,1005)
 1005 FORMAT(1H0,13X,*LIFT DUMPERS DEPLOYED*)
      LINE = LINE+2
  315 IF (COEF.EQ.BCOEF.AND.JBRK.EQ.1) GO TO 320
      IF (COEF.NE.BCOEF) GO TO 320
      JBRK=1
      WRITE (PRINT,1006)
 1006 FORMAT(1H0,13X,*BRAKES ON*)
      LINE = LINE+2
  320 IF (IREV.EQ.1.AND.JREV.EQ.1) GO TO 325
      IF(IREV.EQ.0) GO TO 325
      JREV=1
      NENG=1
      WRITE (PRINT,1007)
 1007 FORMAT(1H0,13X,*REVERSE THRUST ON*)
      LINE = LINE+2
  325 CALL GINTG
      ISEG = 8
      GO TO 20

  330 IF(XEDOT-VWF) 335,340,305
  335 DT = (VWF-XEDOT)/(VTF1-XEDOT)*DTIME
      XE = XE+(0.5*(VWF+XEDOT)-VWF)*DT
      TIMS=TIMS-DT
  340 VTF = VWF
      ISEG = 9
      INDEX=1
      QS=RHOS*VTF*VTF
      MACH=VTF/SOUND
      CALL AEQFM
      GO TO 20

C     OUTPUT SUMMARY OF LANDING

  350 GAMT=GAMTR*RTOD
      GDIST=DIST2+XE
      GTIME=TIME2+TIMS
      TDIST=GDIST+DIST1
      TTIME=GTIME+TIME1
      VAPK=VAPF*FTOK
      WRITE (PRINT,1050) GDIST,GTIME,GAMT,DIST1,TIME1,VAPK,TDIST,TTIME,V
     1SK
```

```
 1050 FORMAT(1H-,10X,17HGROUND DISTANCE =,F8.1,3H FT,4X,13HGROUND TIME =
     1,F6.2,4H SEC,4X,13HGLIDE SLOPE =,F6.2,4H DEG,//,11X,14HAIR DISTANC
     2E =,F8.1,3H FT,5X,
     3               10HAIR TIME =,F6.2,4H SEC, 5X,16HAPPROACH SPEED =,F6.
     42,4H KTS,//,11X,16HTOTAL DISTANCE =,F8.1,3H FT,5X,12HTOTAL TIME =,
     5F6.2,4H SEC, 6X,13HSTALL SPEED =,F6.2,4H KTS)
      RETURN
C     OUTPUT SECTION FOR DEFAULT NOTES
  400 WRITE(PRINT,1051)
 1051 FORMAT(1H0,9X,*GAMMA IS GT. OR EQ. TO ZERO - RETURN *)
      RETURN
  405 WRITE(PRINT,1052)
 1052 FORMAT(1H0,9X,*THETA IS LT. ZERO - CONFIGURATION LANDS NOSEWHEEL F
     1IRST - RETURN*)
      ALFAD = ALFAR*RTOD
      GAMD = GAMR*RTOD
      THETD = THETR*RTOD
      WRITE(PRINT,1053) ALFAD,GAMD,THETD
 1053 FORMAT(1H0,12X,*ALPHA =*,F7.3,* DEG*,/,13X,*GAMMA =*,F7.3,* DEG*,
     1/,13X,*THETA =* ,F7.3,* DEG*)
      RETURN
      END
```

```
      SUBROUTINE LOOK(NX,XC,NY,YC,NZ,ZC,B,B2)                          LOOK
C     6400 FORTRAN SUBROUTINE WHICH DOES FOUR POINT LAGRANGIAN         LOOK
C     INTERPOLATIONS IN THREE DIRECTIONS FOR TWO DEPENDENT VARIABLES.  LOOK
                                                                       LOOK
C     XC, YC, AND ZC ARE INCREASING LISTS OF THREE INDEPENDENT VARIABLES LOOK
C     B AND B2 ARE DEPENDENT VARIABLE ARRAYS CORRESPONDING TO ALL      LOOK
C     COMBINATIONS OF THE INDEPENDENT VARIABLES BY VARYING FIRST XC,   LOOK
C     SECOND YC AND LAST ZC.                                           LOOK
C     ND = 1, WILL RETURN A VALUE OF B AS ANS AND A VALUE OF B2 AS ANS2 LOOK
C             FOR SPECIFIED VALUES XA,YA AND ZA OF XC,YC AND ZC.       7APR2
C     ND = 2, WILL RETURN A VALUE OF B2 AS ANS2 FOR SPECIFIED VALUES   LOOK
C             ANS,XA AND ZA OF B,XC AND ZC.                            7APR2
C     ND = 3, WILL RETURN A VALUE OF B AS ANS FOR SPECIFIED VALUES     LOOK
C             XA, YA AND ZA OF XC,YC AND ZC.                           7APR2
C     ND = 4, WILL RETURN A VALUE OF YC AS ANS2 FOR SPECIFIED VALUES   LOOK
C             ANS,XA AND ZA OF B,XC AND ZC.                            7APR2
C     NDER(1) = GREATER THAN ZERO, CALCULATE DB2DX AND/OR DBDX.        LOOK
C     NDER(2) = GREATER THAN ZERO, CALCULATE DB2DY AND/OR DBDY.        LOOK
C     NDER(3) = GREATER THAN ZERO, CALCULATE DB2DZ AND/OR DBDZ.        LOOK
C     THIS SUBROUTINE CALLS FOR THE FOLLOWING ENTRIES.                 LOOK
C           LAGRA,INTP2                                                LOOK
                                                                       LOOK
      DIMENSION XL(5),XM(20),XN(10),YL(5),YM(20),YN(10),YLL(5),        LOOK
     1 YMM(20),YNN(10),B(1),B2(1),XC(1),YC(1),ZC(1)                    LOOK
      COMMON/LIST9/ANS,ANS2,ND,XA,YA,ZA,INDIC,IFLAG,NDER(3),DBDX,DBDY, LIST9
     * DBDZ,DB2DX,DB2DY,DB2DZ                                          LIST9
C     THIS EQUIVALENCE STATEMENT IS NOT USED IN THE MILSTOL PROGRAM
C     EQUIVALENCE (L,I),(K,M)                                          LOOK
      DWDX(W) = C3+W*(2.*C2+W*3.*C1)                                   7APR2
      ASSIGN 350 TO IWHIT
                                                                       LOOK
      INDIC=1                                                          LOOK
      LINX = LINY = LINZ = 0                                           LOOK
      DO 10 I=1,NX                                                     LOOK
      IF(XA-XC(I))11,20,10                                             7APR2
   10 CONTINUE                                                         LOOK
      II = NX-1                                                        LOOK
      IF = NX                                                          LOOK
      LINX = 1                                                         LOOK
      GO TO 25                                                         LOOK
   11 IF(I-2)120,12,15                                                 LOOK
  120 II = 1                                                           LOOK
      IF = 2                                                           LOOK
      LINX = 1                                                         LOOK
      GO TO 25                                                         LOOK
   12 I = 3                                                            LOOK
   15 IF(I-NX)17,16,16                                                 LOOK
   16 I = NX-1                                                         LOOK
   17 II = I-2                                                         LOOK
      IF = I+1                                                         LOOK
      GO TO 25                                                         LOOK
   20 IF(NDER(1).NE.0)GO TO 11                                         LOOK
      II = I                                                           LOOK
      IF = I                                                           LOOK
   25 GO TO(28,26,28,26),ND                                            LOOK
   26 JI = 1                                                           LOOK
```

```
      JF = NY                                                           LOOK
      GO TO 45                                                          LOOK
   28 DO 30 J=1,NY                                                      LOOK
      IF(YA-YC(J))31,40,30                                              7APR2
   30 CONTINUE                                                          LOOK
      JI = NY-1                                                         LOOK
      JF = NY                                                           LOOK
      LINY = 1                                                          LOOK
      GO TO 45                                                          LOOK
   31 IF(J-2)315,32,35                                                  LOOK
  315 JI = 1                                                            LOOK
      JF = 2                                                            LOOK
      LINY = 1                                                          LOOK
      GO TO 45                                                          LOOK
   32 J = 3                                                             LOOK
   35 IF(J-NY)37,36,36                                                  LOOK
   36 J = NY-1                                                          LOOK
   37 JI = J-2                                                          LOOK
      JF = J+1                                                          LOOK
      GO TO 45                                                          LOOK
   40 IF(NDER(2).NE.0)GO TO 31                                          LOOK
      JI = J                                                            LOOK
      JF = J                                                            LOOK
   45 DO 50 K = 1,NZ                                                    LOOK
      IF(ZA-ZC(K))51,60,50                                              7APR2
   50 CONTINUE                                                          LOOK
      KI = NZ-1                                                         LOOK
      KF = NZ                                                           LOOK
      LINZ = 1                                                          LOOK
      GO TO 65                                                          LOOK
   51 IF(K-2)515,52,55                                                  LOOK
  515 KI = 1                                                            LOOK
      KF = 2                                                            LOOK
      LINZ = 1                                                          LOOK
      GO TO 65                                                          LOOK
   52 K = 3                                                             LOOK
   55 IF(K-NZ)57,56,56                                                  LOOK
   56 K = NZ-1                                                          LOOK
   57 KI = K-2                                                          LOOK
      KF = K+1                                                          LOOK
      IF(IFLAG)65,65,575                                                LOOK
  575 IF(KF-(IFLAG+1))65,58,59                                          LOOK
   58 KI = IFLAG-3                                                      LOOK
      KF = IFLAG                                                        LOOK
      GO TO 65                                                          LOOK
   59 IF(KI-IFLAG)595,65,65                                             LOOK
  595 KI = IFLAG                                                        LOOK
      KF = IFLAG+3                                                      LOOK
      GO TO 65                                                          LOOK
   60 IF(NDER(3).NE.0)GO TO 51                                          LOOK
      KI = K                                                            LOOK
      KF = K                                                            LOOK
   65 XL(1) = XA                                                        7APR2
      XM(1) = YA                                                        7APR2
      XN(1) = ZA                                                        7APR2
      IF(NDER(3).EQ.0)GO TO 200                                         LOOK
      IPATH = 1                                                         LOOK
      GO TO 300                                                         LOOK
  200 IF(NDER(1).EQ.0)GO TO 210                                         LOOK
      IPATH = 2                                                         LOOK
      GO TO 310                                                         LOOK
```

```
210 IF(NDER(2).EQ.0)GO TO 220                                    LOOK
    IPATH = 3                                                    LOOK
    GO TO 320                                                    LOOK
220 IF(NDER(1).EQ.0.AND.NDER(2).EQ.0)GO TO 222                   LOOK
221 IF(MOD(ND,2).NE.0)RETURN                                     LOOK
222 IPATH = 4                                                    LOOK
    GO TO 320                                                    LOOK
300 IZ = 1                                                       LOOK
    DO 1100 M=KI,KF                                              LOOK
    IZ = IZ+1                                                    LOOK
310 IX = 1                                                       LOOK
    DO 1200 L=II,IF                                              LOOK
    IX = IX+1                                                    LOOK
320 IY = 1                                                       LOOK
    DO 1300 JJ = JI,JF                                           LOOK
    IY = IY+1                                                    LOOK
    IF(IPATH.LT.4)GO TO 330                                      LOOK
    IF(MOD(ND,2).NE.0)GO TO 330                                  LOOK
    J = NY+1-JJ                                                  LOOK
    GO TO 340                                                    LOOK
330 J = JJ                                                       LOOK
340 CONTINUE
    IF(IPATH.EQ.1) GO TO IWHIT,(300)
    IZ = 1                                                       LOOK
    DO 1400 K=KI,KF                                              LOOK
    IZ = IZ+1                                                    LOOK
    IF(IPATH.EQ.2) GO TO IWHIT,(300)
    IX = 1                                                       LOOK
    DO 1500 I=II,IF                                              LOOK
    IX = IX+1                                                    LOOK
350 IJK = I+NX*(J-1+NY*(K-1))                                    LOOK
    GO TO(360,370,380,380),IPATH                                 LOOK
360 YM(IY) = B(IJK)                                              LOOK
    IF(ND.LE.2)YMM(IY) = B2(IJK)                                 LOOK
    GO TO 430                                                    LOOK
370 YN(IZ) = B(IJK)                                              LOOK
    IF(ND.LE.2)YNN(IZ) = B2(IJK)                                 LOOK
    GO TO 405                                                    LOOK
380 YL(IX) = B(IJK)                                              LOOK
    IF(ND.LE.2)YLL(IX) = B2(IJK)                                 LOOK
1500 XL(IX) = XC(I)                                              LOOK
    IF(LINX.EQ.0)GO TO 388                                       LOOK
    YN(IZ) = ALIN(2,XL,YL,XA)                                    7APR2
    IF(ND.LE.2)YNN(IZ) = ALIN(2,XL,YLL,XA)                       7APR2
    GO TO 405                                                    LOOK
388 IF(II.NE.IF)GO TO 400                                        LOOK
    YN(IZ) = YL(IX)                                              LOOK
    YNN(IZ) = YLL(IX)                                            LOOK
    GO TO 405                                                    LOOK
400 IND = 0                                                      LOOK
    CALL LAGRA(XL,YL,IND)                                        LOOK
    YN(IZ) = YL(1)                                               LOOK
    IF(ND.GT.2)GO TO 405                                         LOOK
    CALL LAGRA(XL,YLL,IND)                                       LOOK
    YNN(IZ) = YLL(1)                                             LOOK
405 XN(IZ) = ZC(K)                                               LOOK
1400 CONTINUE                                                    LOOK
    IF(LINZ.EQ.0)GO TO 408                                       LOOK
    YM(IY) = ALIN(2,XN,YN,ZA)                                    7APR2
    IF(ND.LE.2)YMM(IY) = ALIN(2,XN,YNN,ZA)                       7APR2
    GO TO 430                                                    LOOK
```

```
408 IF(KI.NE.KF)GO TO 420                                                LOOK
    YM(IY) = YN(IZ)                                                      LOOK
    IF(ND.LE.2)YMM(IY) = YNN(IZ)                                         LOOK
    GO TO 430                                                            LOOK
420 IND = 0                                                              LOOK
    CALL LAGRA(XN,YN,IND)                                                LOOK
    YM(IY) = YN(1)                                                       LOOK
    IF(ND.GT.2)GO TO 430                                                 LOOK
    CALL LAGRA(XN,YNN,IND)                                               LOOK
    YMM(IY) = YNN(1)                                                     LOOK
430 XM(IY) = YC(J)                                                       LOOK
    IF(IPATH.EQ.4)GO TO(1300,100,1300,98),ND                             LOOK
1300 CONTINUE                                                            LOOK
    IF(LINY.EQ.0)GO TO 438                                               LOOK
    ANS = ALIN(2,XM,YM,YA)                                               7APR2
    IF(IPATH.EQ.3)DBDY = SL(2,XM,YM)                                     LOOK
    IF(ND.GT.2)GO TO 460                                                 LOOK
    ANS2 = ALIN(2,XM,YMM,YA)                                             7APR2
    IF(IPATH.EQ.3)DB2DY = SL(2,XM,YMM)                                   LOOK
    GO TO 460                                                            LOOK
438 IF(JI.NE.JF)GO TO 450                                                LOOK
    ANS = YM(IY)                                                         LOOK
    IF(ND.LE.2)ANS2 = YMM(IY)                                            LOOK
    GO TO 460                                                            LOOK
450 CALL INTP2(5,XM,YM,YA,ANS,C1,C2,C3,C4)                               7APR2
    IF(IPATH.EQ.3)DBDY = DWDX(YA)                                        7APR2
    IF(ND.GT.2)GO TO 460                                                 LOOK
    CALL INTP2(5,XM,YMM,YA,ANS2,C1,C2,C3,C4)                             7APR2
    IF(IPATH.EQ.3)DB2DY = DWDX(YA)                                       7APR2
460 IF(IPATH-3)490,221,480                                               LOOK
480 RETURN                                                               LOOK
490 YL(IX) = ANS                                                         LOOK
    IF(ND.LE.2)YLL(IX) = ANS2                                            LOOK
1200 XL(IX) = XC(I)                                                      LOOK
    IF(LINX.EQ.0)GO TO 498                                               LOOK
    ANS = ALIN(2,XL,YL,XA)                                               7APR2
    IF(IPATH.EQ.2)DBDX = SL(2,XL,YL)                                     LOOK
    IF(ND.GT.2)GO TO 522                                                 LOOK
    ANS2 = ALIN(2,XL,YLL,XA)                                             7APR2
    IF(IPATH.EQ.2)DB2DX = SL(2,XL,YLL)                                   LOOK
    GO TO 522                                                            LOOK
498 IF(II.NE.IF)GO TO 510                                                LOOK
    YN(IZ) = YL(IX)                                                      LOOK
    IF(ND.LE.2)YNN(IZ) = YLL(IX)                                         LOOK
    GO TO 1100                                                           LOOK
510 CALL INTP2(5,XL,YL,XA,ANS,C1,C2,C3,C4)                               7APR2
    IF(IPATH.EQ.2)DBDX = DWDX(XA)                                        7APR2
    IF(ND.GT.2)GO TO 522                                                 LOOK
    CALL INTP2(5,XL,YLL,XA,ANS2,C1,C2,C3,C4)                             7APR2
    IF(IPATH.EQ.2)DB2DX = DWDX(XA)                                       7APR2
522 IF(IPATH.EQ.2)GO TO 210                                              LOOK
    YN(IZ) = ANS                                                         LOOK
    YNN(IZ) = ANS2                                                       LOOK
1100 XN(IZ) = ZC(K)                                                      LOOK
    IF(LINZ.EQ.0)GO TO 528                                               LOOK
    ANS = ALIN(2,XM,YM,ZA)                                               7APR2
    DBDZ = SL(2,XM,YM)                                                   LOOK
    IF(ND.GT.2)GO TO 200                                                 LOOK
    ANS2 = ALIN(2,XM,YMM,ZA)                                             7APR2
    DB2DZ = SL(2,XM,YMM)                                                 LOOK
    GO TO 200                                                            LOOK
```

```
  528 IF(KI.NE.KF)GO TO 540                                          LOOK
      ANS = YN(IZ)                                                   LOOK
      ANS2 = YNN(IZ)                                                 LOOK
  540 CALL INTP2(5,XM,YM,ZA,ANS,C1,C2,C3,C4)                         7APR2
      DRDZ = DWDX(ZA)                                                7APR2
      IF(ND.GT.2)GO TO 200                                           LOOK
      CALL INTP2(5,XM,YMM,ZA,ANS2,C1,C2,C3,C4)                       7APR2
      DR2DZ = DWDX(ZA)                                               7APR2
      GO TO 200                                                      LOOK
   98 YMM(IY) = XM(IY)                                               LOOK
  100 IF(J-NY)101,106,106                                            LOOK
  101 IF(J-1)107,107,102                                             LOOK
  102 IF(JJ-3)1300,1300,104                                          LOOK
  104 IF(ANS-YM(IY-1))1300,115,110                                   LOOK
  106 IF(ANS-YM(IY))1300,116,114                                     LOOK
  107 IF(ANS-YM(IY))113,116,110                                      LOOK
  110 CALL INTP2(IY,YM,YMM,ANS,ANS2,C1,C2,C3,C4)                     LOOK
      IY = IY-4                                                      7APR2
      DO 111 I=1,4                                                   7APR2
      IY = IY+1                                                      7APR2
      J = NY+2-IY                                                    7APR2
  111 YMM(IY) = YC(J)                                                7APR2
      CALL INTP2(IY,YM,YMM,ANS,YA,C1,C2,C3,C4)                       7APR2
      RETURN                                                         LOOK
  113 INDIC=2                                                        LOOK
      GO TO 116                                                      LOOK
  114 INDIC=3                                                        LOOK
      ANS = YM(IY)                                                   LOOK
  116 ANS2 = YMM(IY)                                                 LOOK
      J = NY+2-IY                                                    7APR2
      YA = YC(J)                                                     7APR2
      RETURN                                                         LOOK
  115 IY = IY-1                                                      7APR2
      GO TO 116                                                      7APR2
      END                                                            LOOK
```

```
      SUBROUTINE NWRP2(DX,YIN,XIN,YSAV,XSAV)                         AA
C     6400 FORTRAN SUBROUTINE TO DO A NEWTON WRAPSON LINEAR CONVERGENCE. AA
                                                                     AA
      IF (DX) 10,20,10                                               AA
   10 YSAV=YIN                                                       AA
      XSAV=XIN                                                       AA
      XIN=XIN+DX                                                     AA
      DX=0.0                                                         AA
      GO TO 30                                                       AA
   20 X=XIN                                                          AA
      XIN=XIN-YIN*(XIN-XSAV)/(YIN-YSAV)                              AA
      XSAV=X                                                         AA
      YSAV=YIN                                                       AA
   30 RETURN                                                         AA
      END                                                            AA
```

```
      SUBROUTINE SKIP                                                   AL
C     6400 FORTRAN PAGE EJECT SUBROUTINE.                               AL
                                                                        AL
      INTEGER PAGE,PRINT,CARD                                           AL
      REAL LIFT,MACH,NU                                                 AL

      DIMENSION HEAD(16),NAME(10)                                       AL

      COMMON /LIST1/ LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,HD(60) AL
      COMMON /LIST2/ DELTD,ALFAR,THRTL,GAMR,HIND,VTF,WT,HF,DT,TIMS,XF,FU AL
     1EL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN                      AL
      COMMON /LIST3/ CL,CD,S,QS,THIR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHD,CX, AL
     1CZ,Q,IMOM,WINCR,CMDMP,QSC,ALMXR,VSF,BW                            AL
      COMMON /LIST5/ SIG,SOUND,NU,TEMR,PAMB,IATM,TEMF,DSODH,DRHO        AL

      DATA (NAME(I),I=1,10)/10HSTANDARD A,9HTMOSPHERE,10HTROPIC TEM,9HPE AL
     1RATURES,10H POLAR TEM,9HPERATURES,10H   HOT TEM,9HPERATURES,10H   C AL
     2OLD TEM,9HPERATURES/                                              AL
      DATA (HD(I),I=1,59)/10H          ,10H  AIRSPEED,10H     ALPHA,10H AL
     1ALT ERROR,10H  ALTITUDE,10H       AXB,10H       AZB,10H       CL, AL
     210H       CD,10H  DISTANCE,10H      DRAG,10H  ELEVATOR,10H FUEL U AL
     3SED,10H FLT. PATH,10H      FUEL,10H FUEL FLOW,10H    HEIGHT,10H  H AL
     4OOK HT.,10HHORIZONTAL,10H      LIFT,10HLOAD FACT.,10H  MACH NO.,10 AL
     5HNOR. ACCEL,10H        QS,10H     RANGE,10HR/C ACTUAL,10H   SEGMEN AL
     6T,10H      SEP,10H    SEP/WF,10H SP. RANGE,10HTAN. ACCEL,10H  THR AL
     7OTTLE,10H     THETA,10H THETA DOT,10HTH DBL DPT,10H      TIME,10H AL
     8   WEIGHT,10H (DEGREES),10H    (FEET),10H   (F/S/S),10H   (FT/LB), AL
     910H  (FT/MIN),10H  (FT/SEC),10H       (G),10H   (KNOTS),10H   (LB/ AL
     $HR),10H (MINUTES),10H (N MI/LB),10H (N MILES),10H (PERCENT),10H  ( AL
     $POUNDS),10H    THRUST,10HR/C INSTAN,10H     TEMP.,10H   (DEG F),10 AL
     $H      FLAP,10H (SECONDS),10HTHST ALPHA,10H (DEG/SEC)/            AL
                                                                        AL
C     THE INPUT CONSISTS OF TWO ALPHANUMERIC TITLE CARDS
C     CARD 1 - COLUMNS 2 THROUGH 20 ARE RESERVED FOR ENGINE
C              IDENTIFICATION
C              COLUMNS 21 THROUGH 80 ARE PRINTED AS A TITLE LINE
C     CARD 2 - COLUMNS 1 THROUGH 80 ARE PRINTED AS A SECOND
C              TITLE LINE

      IF (IN.NE.1) GO TO 10                                             AL
      READ (CARD,130) (HEAD(I),I=1,16)                                  AL
  130 FORMAT (1X,A9,7A10/8A10)                                          AL
      RETURN                                                            AL

   10 CALL SECOND(T)                                                    AL
      WRITE (PRINT,140) IDATE,T,PAGE                                    AL
  140 FORMAT (1H1,4X,20HAIRCRAFT PERFORMANCE,49X,A10,10X,F10.2,5X,5HPAGE AL
     1 ,I3)                                                             AL
      NENG=ENGNO                                                        AL
      WRITE (PRINT,150) NENG,HEAD(1),HEAD(2),S                         PRE
  150 FORMAT (1H ,25X,I2* ENGINE*A9,A10*  S = *F5.0* SQ FT *)           PRE
      IF (IATM.EQ.5) GO TO 20                                           AL
      IF (IATM.EQ.0.AND.DT.NE.0.) GO TO 30                              AL
      N=2*(IATM+1)                                                      AL
      WRITE (PRINT,160) NAME(N-1),NAME(N)                               AL
  160 FORMAT (1H+,85X,A10,A9)                                           AL
```

```
      GO TO 40                                                          AL
   20 WRITE (PRINT,170) TEMF                                            AL
  170 FORMAT (1H+,84X12HTEMPERATURE=,F5.1,5HDEG F)                      AL
      GO TO 40                                                          AL
   30 WRITE (PRINT,180) DT                                              AL
  180 FORMAT (1H+,84X5HTEMP=,F5.1,12HDEG FROM STD)                      AL
   40 WRITE (PRINT,190) (HEAD(I),I=3,16)                                AL
  190 FORMAT (35X,6A10/25X,8A10)                                        AL
      PAGE=PAGE+1                                                       AL
      IF (INP.GT.9) GO TO 50                                            AL
      GO TO (60,60,60,60,80,90, 60,60,60), INP                          AL
   50 IP=INP-15                                                         AL
      GO TO 120
   60 LINE=4                                                            AL
      RETURN                                                            AL
   80 WRITE (PRINT,210)                                                 AL
  210 FORMAT (20X,*STOL TAKEOFF*)
      GO TO 120                                                         AL
   90 WRITE (PRINT,220)                                                 AL
  220 FORMAT (20X,*STOL LANDING*)
  120 LINE=5                                                            AL
      RETURN                                                            AL

      END                                                               AL
```

```
      FUNCTION SL(I,X,Y)

      DIMENSION X(1),Y(1)

      SL = (Y(I+1)-Y(I))/(X(I+1)-X(I))
      RETURN

      END
```

```
      SUBROUTINE SMLT2(NSOL,CN)
C     GENERAL SIMULTANEOUS EQUATION SOLUTION SUBROUTINE.

C     NSOL = NO. OF SOLUTIONS(MAX 6), CN = COEFFICIENTS, EQUATION FORM
C     CN(I,1) = CN(I,2)A(1)+CN(I,3)A(2)+...CN(I,NSOL+1)A(NSOL)
C     WHERE I TAKES VALUES 1 THROUGH NSOL, AND A(I) ARE SOLUTIONS.
C     SOLUTIONS ARE RETURNED AS CN(I,1).

      DIMENSION CN(6,7)

      II=NSOL
      GO TO 20
   10 II=IL
   20 JJ=II+1
      DO 60 I=1,II
C     IF CN(I,JJ).EQ.0.0
C          IF I.GT.1 ADD EQ 1 TO EQ I.
C          IF I.EQ.1 FIND LOWEST I(=N) WHERE CN(I,JJ).NE.0.0,
C          AND ADD EQ N TO EQ 1.
      IF(CN(I,JJ).NE.0.0)GO TO 30
      DO 32 N=1,II
      IF(CN(N,JJ).NE.0.0)GO TO 36
   32 CONTINUE
      WRITE(6,1000)JJ
 1000 FORMAT(10X6HCOLUMN,2X,I2,2X8HALL ZERO)
      STOP
   36 DO 38 M=1,JJ
   38 CN(I,M) = CN(I,M)+CN(N,M)

C     DIVIDE ROW BY CN(I,JJ), LEAVING 1. IN LAST OR JJ COLUMN OF
C          CURRENT MATRIX.

   30 RCP=1./CN(I,JJ)
      DO 40 J=1,JJ
   40 CN(I,J)=CN(I,J)*RCP
   60 CONTINUE

C     STORE EQUATION FOR CALCULATING SOLUTION IN EMPTY COLUMN.
C     LAST NUMBER STORED IS FIRST SOLUTION.

      DO 65 I=1,II
   65 CN(I,JJ) = CN(1,I)
      IL=II-1
      IF (IL) 90,90,70

C     WHEN IL = 0, THE FIRST SOLUTION IS COMPLETE.

C     ASSUMING 1 IN EACH LAST COLUMN, SUBTRACTING SUCCESSIVE EQUATIONS
C          RESULTS IN ONE LESS EQUATION EACH WITH ONE LESS UNKNOWN.

   70 DO 80 I=1,IL
      I2=I+1
      DO 80 J=1,II
   80 CN(I,J)=CN(I,J)-CN(I2,J)
      GO TO 10
   90 IF (NSOL-1) 100,120,100
C     CALCULATE REMAINING SOLUTIONS, STORE IN FIRST COLUMN.

  100 DO 110 I=2,NSOL
      J1=I+1
      CN(I,1)=CN(1,J1)
      JF=I-1
      DO 110 J=1,JF
      J2=J+1
  110 CN(I,1)=CN(I,1)-CN(J2,J1)*CN(J,1)
  120 RETURN
      END
```

```
      SUBROUTINE TAKEOFF
C     6400 FORTRAN IV TAKEOFF SUBROUTINE
C     STOL TACTICAL AIRCRAFT INVESTIGATION GROUND RULES - JULY 1972

C     THE BOUNDARIES OF THIS PROGRAM ARE -
C        VR GT OR EQ TO V1 AND VMC GROUND
C        VLO GT OR EQ TO VSC * VMC AIR
C        ALPHA AT LIFTOFF LT OR EQ TO GALMX (AERO INPUT)
C        ALPHA DURING ROTATION LT OR EQ TO ALPMX (AERO INPUT)

C     THIS PROGRAM CALLS THE FOLLOWING SUBROUTINES.
C         ATMOS,SKIP,HEAD,THST1,NWRP2,AERO1,AEQFM,GINTG

      INTEGER CARD,PRINT,PAGE
      REAL LIFT,KTOF,MACH,NU,NMTOF

      DIMENSION NL(8)

      COMMON /LIST1/ LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,HD(60)     F
      COMMON /LIST2/ DELTD,ALFAR,THRTL,GAMR,HIND,VTF,WT,HF,DT,TIMS,XF,FU
     1EL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN
      COMMON /LIST3/ CL,CD,S,QS,THIR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHD,CX,
     1CZ,Q,IMOM,WINCR,CMDMP,QSC,ALMXR,VSF,BW
      COMMON /LIST4/ FFS,JPOW,TGROS,DWA,THMOM,THREQ
      COMMON /LIST5/ SIG,SOUND,NU,TEMR,PAMB,IATM,TEMF,DSODH,DRHO
      COMMON /LIST6/ RTOD,DTOR,KTOF,FTOK,NMTOF,FTONM,RHOZ,RHOZ2,GZ,PZ,TZ
      COMMON /LIST8/ INDEX,COEF,GN,GT,AN,AT,AX,AZ,ITRM,VUPPER
      COMMON /LIST15/ IALMX,INT,XEDOT,ZEDOT,XE,ZE,ILOOP,DTIME,IAT,FLIFT,
     1DFLP,C4HT,GALMXR
      COMMON /LGEOM/ SHSW,SCR,QDOT,RTH,DTHTH,RWA,DTHWA,DSTHZ,DWLHZ,IYY,X
     1CG,ZCG,CBAR,PFN,SINTH,COSTH,ANGLE,SINAL,COSAL,RHZ,DTHHZ,VTSQ,UDOT,
     2WDOT,THETR,U,W
      COMMON/CONTROL/JFIG,IREV,ISP,NENG
      COMMON /LIST99/ IERR

      DATA VSC,DFLP,RCOEF,BCOEF,ROTATN,ROTPT,TIMR,TIMB,DGLO/1.05,25.,.1,
     10.,3,8.,0.,1.,2.,0.,10/
      DATA (NL(I),I=1,8)/10H0556370226,10H3323082052,10H3938514542,10H38
     140015151,10H3610171403,10H3431091158,10H5739393838,10H5940015138/

C     DEFINITION OF VARIABLES IN NAMELIST 'TAKEOF1'
C     VCS = RATIO OF LIFTOFF SPEED TO AIR MINIMUM CONTROL SPEED
C     DGLO=INCREMENTAL LOAD FACTOR REQUIRED AT LIFTOFF
C     DFLP=FLAP SETTING IN DEGREES
C     RCOEF = ROLLING COEFFICIENT OF FRICTION.
C     BCOEF = COEFFICIENT OF BRAKING FRICTION
C     VMCGK= MINIMUM CONTROL SPEED ON THE GROUND (KNOTS)
C     VMCAK = MINIMUM CONTROL SPEED IN THE AIR (KNOTS)
C     ROTATN   ROTATION RATE (DEGREES PER SECOND)
C     TIME = REACTION TIMR FOR ENGINE FAILURE (SEC)
C     TIMB = BRAKING DELAY AFTER 'TIMR' (SEC)
C     ROTPT     EQUAL TO ZERO  SUPRESSES PRINTING OUTPUT FOR
C          SEGMENTS 4,6 AND 8.

C     THE FOLLOWING VALUES ARE ENTERED AT TIME OF LOADING AND ARE USED
C     UNTIL OVERRIDEN BY READING THE APPROPRIATE VARIABLES IN TAKEOF1
```

```
C         VSC=1.05
C         DGLO=0.10
C         DFLP=25.
C         RCOEF=0.10
C         BCOEF=0.30
C         TIMR=1.0
C         TIMB=2.0
C         ROTATN=8.0
C         ROTPT=1.

      NAMELIST/TAKEOF1/VSC,DFLP,RCOEF,BCOEF,VMCGK,VMCAK,ROTATN,
     1ROTPT,TIMR,TIMB,DGLO

      IF (IN.NE.1) GO TO 10
      READ (CARD,TAKEOF1)
      WRITE (PRINT,TAKEOF1)
      RETURN

   10 HFRUN=HF
      HF=HFRUN+C4HT
      IBAL=0
      CALL ATMOS
      RHOS=RHOZ2*SIG*S
      VMCA = VMCAK*KTOF
      VMCG = VMCGK*KTOF                                                PRE
      V1F=VMCG
      V1=VMCGK

      WRITE (PRINT,914)
  914 FORMAT(1H1,3X,*OUTPUT DEFINITIONS - TAKEOFF*,/,4X,*SEGMENT = *,
     1*1  DETERMINATION OF LIFTOFF ANGLE OF ATTACK*,/,14X,
     2*2  INCREASE VLO TO  AVOID GROUND CONTACT*,/,14X,
     3*3  R/C AVAILABLE AT LIFTOFF - 1 ENGINE OUT*,/,14X,
     4*4  VLO TO VR INTEGRATION STEPS*,/,14X,
     5*5  CONDITIONS AFTER VLO TO VR INTEGRATION*,/,14X,
     6*6  VR TO VF INTEGRATION STEPS*,/,14X,
     7*7  CONDITIONS AFTER VR TO VF INTEGRATION*,/,14X,
     8*8  VF TO STOP INTEGRATION STEPS*,/,14X,
     9*9  CONDITIONS AFTER VF TO STOP INTEGRATION*,/,14X,
     1*10 V=0 TO VR INTEGRATION STEPS*,/,14X,
     2*11 CONDITIONS AFTER GROUND RUN TO VR*)
      LINE=LIMIT+1
      GO TO 160

C     OUTPUT BLOCK.

   20 IF (ROTPT.NE.0.) GO TO 30
      IF ((ISEG.EQ.4).OR.(ISEG.EQ.6).OR.(ISEG.EQ.8).OR.(ISEG.EQ.10))
     1GO TO 60
   30 IF (LINE-LIMIT) 50,40,40
   40 CALL SKIP
      CALL HEAD(NL)
      WRITE (PRINT,913) HD(27)
  913 FORMAT (1H+,1X,A10)
   50 VTK=VTF*FTOK
      GAMD=GAMR*RTOD
      RCM=60.*VTF*SIN(GAMR)
      THETD=THETR*RTOD
      THDTD=Q*RTOD
      ALFAD=ALFAR*RTOD
      ALTHD=ALTHR*RTOD
```

```
      WRITE (PRINT,912) ISEG,HF,DFLP,WT,VTK,RCM,THETD,AN,CL,LIFT,TGROS,T
     1IMS,XE,ZE,GAMD,ALFAD,THDTD,AT,CD,DRAG,ALTHD
      LINE=LINE+3
  912 FORMAT (1H0,I11,6F12.3,2F12.5,2F12.3/12X,5F12.3,3F12.5,2F12.3)
   60 GO TO (250,210,570,610,670,680,700,730,750,820,950),ISEG

C     SET REQUIRED LIFT-OFF SPEED  (POWER ON STALL LESS 1 ENGINE)

  160 IN=0
      CALL AERO1
      ALTMR = ALPHD-WINCR+THIR
      VTF1 = VSF
      ENGNO = ENGNO-1.
      NENG=2
      THRTL=1.00
      MACH = VTF1/SOUND
      VTF = VTF1
      NSTEP = 0
  170 CALL THST1
      CALL AERO1
      IF(JFIG.EQ.2.AND.(VTF+VWF).GT.1.)CLMAX=CLMAX-((TGROS*SIN(ALTMR))/(
     1VTF*VTF*RHOS))
      VTF =SQRT((WT-TGROS*SIN(ALTMR))/(RHOS*CLMAX))
      YVTF = (VTF1-VTF)/VTF
      IF(ABS(YVTF)-0.0001) 174,174,171
  171 IF(NSTEP) 173,172,173
  172 DMAC = MACH-VTF/SOUND
  173 CALL NWRP2(DMAC,YVTF,MACH,YSAV,XSAV)
      NSTEP = NSTEP+1
      IF(NSTEP.GT.10) GO TO 161
      VTF1 = MACH*SOUND
      VTF = VTF1
      GO TO 170
  161 VTF = MACH*SOUND
  174 VSF = VTF
      IN=2
      VLOF = VSC*VMCA
      IVSC=0
      ILOF=0
      VTF=VLOF
      VSK=VSF*FTOK
      IF(VTF.GE.VSF*SQRT(1.+DGLO)) GO TO 25
      ILOF=1
      VTF=VSF*SQRT(1.+DGLO)
   25 VLOF=VTF
      VLOK=VLOF*FTOK

C     DETERMINE ANGLE OF ATTACK AT LIFTOFF.

  190 VTF=VLOF
      THRTL=1.00
      ISP = 0
      IREV = 0
      COEF = RCOEF
      TIMS=0.0
      XE=0.0
      ZE=0.0
      Q = 0.0
      IERR=0
      IAT=0
      IALMX=0
```

```
      INTEG=0
      HF=HFRUN+C4HT
      GAMR=0.
      ALFAR=WINCR
      INDEX=5
      QS=RHOS*VTF*VTF
      MACH=VTF/SOUND
      GN=1.0
      GT=0.0
      CALL AEQFM
  200 ALFLO=ALFAR
      THETR=ALFAR-WINCR
      IF (ALFLO-WINCR) 900,201,201
  201 IF(ALFAR.LE.GALMXR) ISEG=1
      IF(ALFAR.GT.GALMXR) ISEG=2
      GO TO 20

C     ADJUST VLOF TO AVOID GROUND CONTACT

  210 IVSC = IVSC+1
      IF(ILOF.GT.0) GO TO 211
      VSC1 = VSC+FLOAT(IVSC)/100.
      VTF=VMCA*VSC1
      WRITE (PRINT,910) ALFAD
      WRITE(PRINT,917) VSC1
  917 FORMAT(1H ,9X,16HVLO INCREASED TO,F7.3,2X,12HTIMES VMCAIR)
      LINE = LINE+3
      GO TO 25
  211 VTF=VSF*(SQRT(1.+DGLO)+FLOAT(IVSC)/100.)
      VSC1=VTF/VSF
      WRITE(PRINT,910) ALFAD
      WRITE(PRINT,911) VSC1
  911 FORMAT(1H ,9X,16HVLO INCREASED TO,F7.3,2X,12HTIMES VSTALL)
      LINE = LINE+3
      GO TO 25


C     DETERMINE R/C AVAILABLE AT LIFTOFF.

  250 INDEX=3
      ALFAR=WINCR
      QS=RHOS*VTF*VTF
      MACH=VTF/SOUND
      CALL AEQFM
      THETR=ALFAR-WINCR+GAMR
      ISEG=3
      GO TO 20

C     INTEGRATION FROM VLO TO VR.

  570 IF (INTEG) 590,590,580
  580 SEG2=0.0
      TIME2=0.0
      VRF=VLOF
      GO TO 660
  590 HF=HFRUN+C4HT
      CALL ATMOS
      RHOS=RHOZ2*SIG*S
      TIMS=0.0
      XF=0.0
      ZE=0.0
```

```
      IALMX=0
      IAT=0
      THETR=ALFLO-WINCR
      XEDOT=VLOF
      ZEDOT=0.0
      INDEX=1
      INT=2
      Q=ROTATN*DTOR
      DTIME=-0.2
      ILOOP=1
  600 VTF1=VTF
      CALL GINTG
      ISEG=4
      GO TO 20

C     CONDITIONS AFTER VLO TO VR INTEGRATION

  610 IF (ALFAR-WINCR) 620,650,600
  620 DT=-(ALFAR-WINCR)/Q
      ALFAR=  WINCR
      THETR = 0.00
      VRF=VTF+DT*(VTF-VTF1)/DTIME
      IF(VRF-V1F) 625,640,640
  625 DVLOF = (V1F        - VRF)
      IF(DVLOF-.1) 640,640,630

C        EMPIRICAL FACTOR ADDED TO IMPROVE ESTIMATE OF DV NEEDED            PRE

  630 VLOF=VLOF+ .75*(V1F       -VRF)
      GO TO 190
  640 XE=XE+(.5*(VTF+VRF)-VWF)*DT
      VTF=VRF
      TIMS=TIMS+DT
  650 SEG2=-XE
      TIME2=-TIMS
  660 VTK2=VLOF*FTOK
      IF(ABS(VTF-V1F).LE.1.0) ISEG=7
      IF(ABS(VTF-V1F).GT.1.0) ISEG=5
      INDEX=1
      QS=RHOS*VTF*VTF
      MACH=VTF/SOUND
      CALL AEQFM
      GO TO 20


C     INTEGRATION FROM VR TO VF

  670 SEG3=0.00
      TIME3=0.00
      TIMS=0.00
      INT=1
      DTIME=-0.2
      XF=0.00
      ZE=0.00
      THETR=0.0
      XEDOT=VRF
      ZEDOT=0.00
      ILOOP=1
      VTF=VRF
  675 VTF1=VTF
      CALL GINTG
```

```
      ISEG=6
      GO TO 20
C     CONDITIONS AFTER VR TO VF INTEGRATION
  680 IF(VTF-V1F) 685,690,675
  685 DT =  DTIME*((VTF-V1F)/(VTF-VTF1))
      TIMS = TIMS-DT
      XE = XE-((VTF+V1F)*0.5-VWF)*DT
      VTF=V1F
  690 SEG3=-XE
      TIME3=-TIMS
      V1=V1F*FTOK
      ISEG=7
      INDEX=1
      QS=RHOS*VTF*VTF
      MACH=VTF/SOUND
      CALL AEQFM
      GO TO 20
C     INTEGRATION FROM VF TO STOP
  700 TIMS=0.0
      INT=1
      SEG5=0.0
      DTIME=0.5
      TIME5=0.00
      XE=0.0
      ZE=0.0
      THETR=0.0
      ZEDOT=0.0
      Q=0.0
      XEDOT=V1F
      ILOOP=1
      VTF=V1F
  710 VTF1=VTF
      IF(TIMS+0.1*DTIME.GE.TIMR) THRTL=0.0
      IF(TIMS+0.1*DTIME.LT.TIMR+TIMB) GO TO 720
      COEF=BCOEF
      IREV=1
      NENG = 2
      ISP=1
  720 CALL GINTG
      ISEG=8
      GO TO 20
C     CONDITIONS AFTER VF TO STOP INTEGRATION
  730 IF(XEDOT-VWF)735,740,710
  735 DT = (VWF-XEDOT)/(VTF1-XEDOT)*DTIME
      XE = XE+(0.5*(VWF+XEDOT)-VWF)*DT
      TIMS = TIMS-DT
  740 VTF=VWF
      ISEG=9
      INDEX=1
      QS=RHOS*VTF*VTF
      MACH=VTF/SOUND
      SEG5=XE
      TIME5=TIMS
      CALL AEQFM
      GO TO 20
```

```
C     MODIFY V1 TO BALANCE ISEG=3 + ISEG=2 WITH ISEG=5

  750 IF (SEG2+SEG3.LE.SEG5.AND.IBAL.EQ.0) GO TO 795
      IF(ABS(SEG2+SEG3-SEG5).LE. 2.) GO TO 795
      IF(IBAL.GT.20) GO TO 795
      IF(IBAL.GT.0) GO TO 755
      VOLD=V1F
      D1=SEG2+SEG3
      D2=SEG5
      VNEW=V1F+10.
      D3=D1*(VLOF-VNEW)/(VLOF-VOLD)
      D4=D2*VNEW/VOLD
      GO TO 760
  755 D3=SEG2+SEG3
      D4=SEG5
      VNEW=V1F
  760 IBAL=IBAL+1
      V1F=VOLD+((D2-D1)*(VNEW-VOLD)/(D3+D2-D1-D4))
      V1 = V1F*FTOK
      IF(IBAL.EQ.1) GO TO 765
      VOLD=VNEW
      D1=D3
      D2=D4
  765 WRITE(PRINT,915) V1
  915 FORMAT(1H0,10X,*FIELD NOT BALANCED*,5X,*NEW V1 SPEED = *,F7.2,
     1* KTS*)
      GO TO 190
  795 WRITE(PRINT,916) IBAL
  916 FORMAT(1H0,10X,*FIELD BALANCED IN *,I2,* ITERATIONS*)

C     INTEGRATION FROM V=0 TO VF

  800 DTIME=1.0
      INDEX=1
      HF=HFRUN+C4HT
      CALL ATMOS
      RHOS=RHOZ2*SIG*S
      TIMS=0.0
      XF=0.0
      ZF=0.0
      THETR=0.0
      THRTL=1.00
      ENGNO = ENGNO+1.
      NENG=0
      COEF = RCOEF
      ISP=0
      IRFV=0
      XFDOT=VWF
      ZFDOT=0.0
      Q=0.0
      ILOOP=1
  810 VTF1=VTF
      CALL GINTG
      ISEG=10
      GO TO 20

C     CONDITIONS AFTER V=0 TO VF INTEGRATION

  820 IF(VTF-V1F) 810,840,830
  830 DT=-DTIME*(VTF-V1F)/(VTF-VTF1)
```

```
      XE=XE+((VTF+V1F)*.5-VWF)*DT
      TIMS=TIMS+DT
      VTF=V1F
  840 SEG1=XE
      TIME1=TIMS
      VTK1=V1F*FTOK
      ISEG=11
      INDEX=1
      QS=RHOS*VTF*VTF
      MACH=VTF/SOUND
      CALL AEQFM
      GO TO 20

C     OUTPUT SECTION FOR DEFAULT NOTES

  900 ALFLO = ALFLO * RTOD
      WRITE (PRINT,910) ALFLO
  910 FORMAT(1H0,9X,*AIRPLANE LIFTS OFF AT ALPHA =*F7.3,* DEG*)
      RETURN

C     OUTPUT SUMMARY OF TAKEOFF

  950 GDIST=SEG1+SEG2+SEG3
      GTIME=TIME1+TIME2+TIME3
      ADIST=SEG1+SEG5
      ATIME=TIME1+TIME5
      VLOK=VLOF*FTOK
      VRK=VRF*FTOK
      V1K=V1F*FTOK
      WRITE(PRINT,955) V1K,VRK,VLOK,VSK
  955 FORMAT(1H0,10X,*ENGINE FAILURE SPEED =*,F7.2,* KTS*,/,11X,*ROTATIO
     1N SPEED =*,F7.2,* KTS*,/,11X,*LIFTOFF SPEED =*,F7.2,* KTS*,/,11X,*
     2STALL SPEED =*,F7.2,* KTS (WITH ONE ENGINE OUT)*)
      WRITE(PRINT,956) GDIST,GTIME
  956 FORMAT(1H0,10X,*V=0 TO LIFTOFF*,/,11X,*DISTANCE =*,F7.2,* FT*,10X,
     1*TIME =*,F7.2,* SEC*)
      WRITE(PRINT,957) ADIST,ATIME
  957 FORMAT(1H0,10X,*ABORTED TAKEOFF*,/,11X,*DISTANCE =*,F7.2,* FT*,10X
     1,*TIME =*,F7.2,* SEC*)
      RETURN

      END
```

```
      SUBROUTINE THST1
C     6400 FORTRAN SUBROUTINE TO CALCULATE THRUST AS A FUNCTION OF
C     TRUE AIRSPEED IN KNOTS
C     NOTE... THIS PROGRAM USES THE COMMON LIST ELEMENT -THMOM- (LIST4) AS THE
C             GROSS THRUST FROM THE BLOWING SLOT

      INTEGER CARD,PRINT,PAGE
      REAL LIFT,MACH,KTOF,NMTOF,IYY

      COMMON /LIST1/LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,HD(60)
      COMMON /LIST2/DELTD,ALFAR,THRTL,GAMR,HIND,VTF,WT,HF,DT,TIMS,XF,
     * FUEL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN
      COMMON /LIST3/CL,CD,S,QS,THIR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHD,CX,
     * CZ,Q,IMOM,WINCR,CMDMP,QSC,ALMXR,VSF,BW
      COMMON /LIST4/FFS,JPOW,TGROS,DWA,THMOM,THREQ
      COMMON /LIST5/ SIG,SOUND,NU,TEMR,PAMB,IATM,TEMF,DSODH,DRHO
      COMMON /LIST6/RTOD,DTOR,KTOF,FTOK,NMTOF,FTONM,RHOZ,RHOZ2,GZ,PZ,TZ
      COMMON/LGEOM/SHSW,SCR,QDOT,RTH,DTHTH,RWA,DTHWA,DSTHZ,DWLHZ,IYY,
     1 XCG,ZCG,CBAR,PFN,SINTH,COSTH,ANGLE,SINAL,COSAL,RHZ,DTHHZ,VTSQ,
     2 UDOT,WDOT,THETR,U,W
      COMMON/CONTROL/JFIG,IREV,ISP,NENG

      DIMENSION VKS(20),THST(20),RDRG(20),TIDL(20),TSLT(20),TREV(20),
     1          DWME(20)

      DATA ENGNO,SCALE,NENG/4.0,1.0,1/

C     DEFINITION OF VARIABLES IN NAMELIST 'THT1'
C     N-   NUMBER OF X,Y POINTS IN EACH TABLE
C     VKS- THE VELOCITY TABLE FOR THE PROPULSION TABLES IN KTAS
C          (USED AS THE INDEPENDENT VARIABLE IN ALL TABLES)
C     THST-GROSS THRUST TABLE AT MAX POWER (IN LBS.)
C     RDRG-RAM DRAG TABLE AT MAX POWER (IN LBS.)
C     TIDL-GROSS THRUST TABLE AT IDLE POWER (IN LBS.)
C     TSLT-GROSS THRUST AT THE SLOT EXIT (IBF CONFIGS) AT MAX POWER (LBS)
C     TREV-MAX REVERSE THRUST (EXPRESSED AS A NEGATIVE VALUE - IN LBS.)
C     DWME-WINDMILLING DRAG FOR A DEAD ENGINE (IN LBS.)
C
C     THID-THRUST VECTOR INCIDENCE REF. TO A WATER LINE IN DEG.
C     ENGNO-THE NUMBER OF ENGINES
C     SCALE-SCALING FACTOR FOR THE PROPULSION DATA
C     NENG=0  NO REVERSE THRUST
C         =1  ALL ENGINES REVERSING
C         =2  ENGINE OUT REVERSING PROCEDURE

C     THE FOLLOWING VARIABLES ARE ENTERED AT TIME OF LOADING AND ARE USED
C     UNTIL OVERRIDDEN BY READING THE APPROPRIATE VARIABLES IN THT1
C        ENGNO = 4.0
C        SCALE = 1.0
C        NENG = 1

      NAMELIST/THT1/N,VKS,THST,RDRG,TIDL,TSLT,TREV,THID,ENGNO,SCALE,NENG
     1              ,DWME

      IF(IN-1)5,1,5
    1 READ(CARD,THT1)
```

```
      WRITE(PRINT,THT1)
      THIR=THID*DTOR
      ENG=ENGNO
      RETURN
    5 VTK=VTF*FTOK
      CALL FIND(N,VTK,DWA,VKS,RDRG)
      CALL FIND(N,VTK,THRST,VKS,THST)
      CALL FIND(N,VTK,WMD,VKS,DWME)
      IF(IREV.GT.0.) GO TO 50
      THMOM = 0.00
      IF(JFIG.LE.2) GO TO 10
      CALL FIND(N,VTK,THMOM,VKS,TSLT)
   10 IF(THRTL.EQ.1.0) GO TO 25
      CALL FIND(N,VTK,THIDL,VKS,TIDL)
      RATIO=(THIDL+THRTL*(THRST-THIDL))/THRST
      THRST=THRST*RATIO
      DWA=DWA*RATIO
      THMOM=THMOM*RATIO
   25 TGROS=THRST*ENGNO*SCALE
      DWA=DWA*ENGNO*SCALE
      IF(NENG.EQ.2) DWA=DWA+WMD*SCALE
      THMOM=THMOM*ENGNO*SCALE
      RETURN
   50 CALL FIND(N,VTK,THREV,VKS,TREV)
      CALL FIND(N,VTK,THIDL,VKS,TIDL)
      IF(NENG.EQ.0)GO TO 65
      IF(NENG.EQ.1)GO TO 60
      THRST=THREV*(2.0*FLOAT (IFIX(ENGNO/2.+.0001)))+THIDL
   75 TGROS=THRST*SCALE
      DWA=DWA*SCALE*ENGNO
      IF(NENG.EQ.2) DWA=DWA+WMD*SCALE
      THMOM=0.0
      RETURN
   60 THRST=THREV*(ENGNO-2.)+THIDL*2.
      GO TO 75
   65 DWA=ABS(DWA*THIDL/THRST)
      THRST=THIDL*ENGNO
      GO TO 75
      END
```

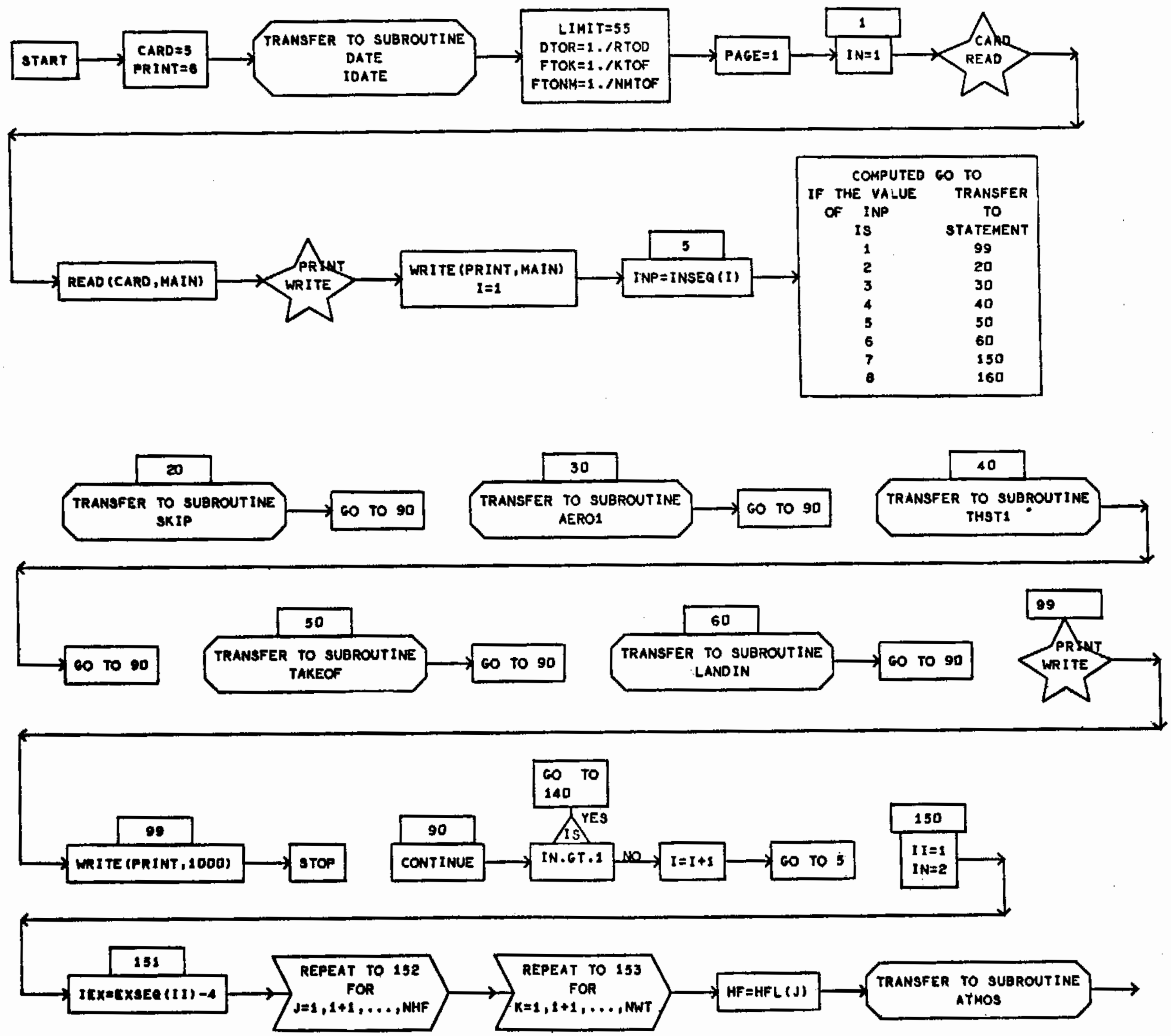
```
      SUBROUTINE TRIM(N,A,V,PV,T,K)                                    K
C     FORTRAN SUBROUTINE WHICH TRIMS N VALUES OF ACCELERATION(A) TO ZERO K
C          BY VARYING N INDEPENDENT VARIABLES (V).                     K
                                                                       K
C     PV = IS A SET OF PERTURBATION INCREMENTS FOR (V) TO USE FOR      K
C          ESTABLISHING 'DERIVATIVES'.                                 K
C     T IS A SET OF TOLERANCES TO THE  VALUES OF (A) WHICH MUST        K
C          BE SATISIFIED.                                              K
C     THIS SUBROUTINE CALLS SUBROUTINE SMLT2.                          K
                                                                       K
      DIMENSION A(N),V(N),PV(N),T(N),D(6,7),DV(6)                      K

      EQUIVALENCE (D(1,1),DV(1))                                       K

      IF (K) 10,10,60                                                  K
   10 DO 20 I=1,N                                                      K
      IF (ABS(A(I))-T(I)) 20,20,30                                     K
   20 CONTINUE                                                         K
      K=N+1                                                            K
      RETURN                                                           K
   30 K=1                                                              K
      DO 40 I=1,N                                                      K
   40 D(I,1)=-A(I)                                                     K
   50 V(K)=V(K)+PV(K)                                                  K
      RETURN                                                           K
   60 DO 70 I=1,N                                                      K
   70 D(I,K+1)=(A(I)+D(I,1))/PV(K)                                     K
      V(K)=V(K)-PV(K)                                                  K
      IF (K-N) 80,90,90                                                K
   80 K=K+1                                                            K
      GO TO 50                                                         K
   90 CALL SMLT2(N,D)                                                  K
      DO 100 I=1,N                                                     K
  100 V(I)=V(I)+DV(I)                                                  K
      K=0                                                              K
      RETURN                                                           K
      END                                                              K
```

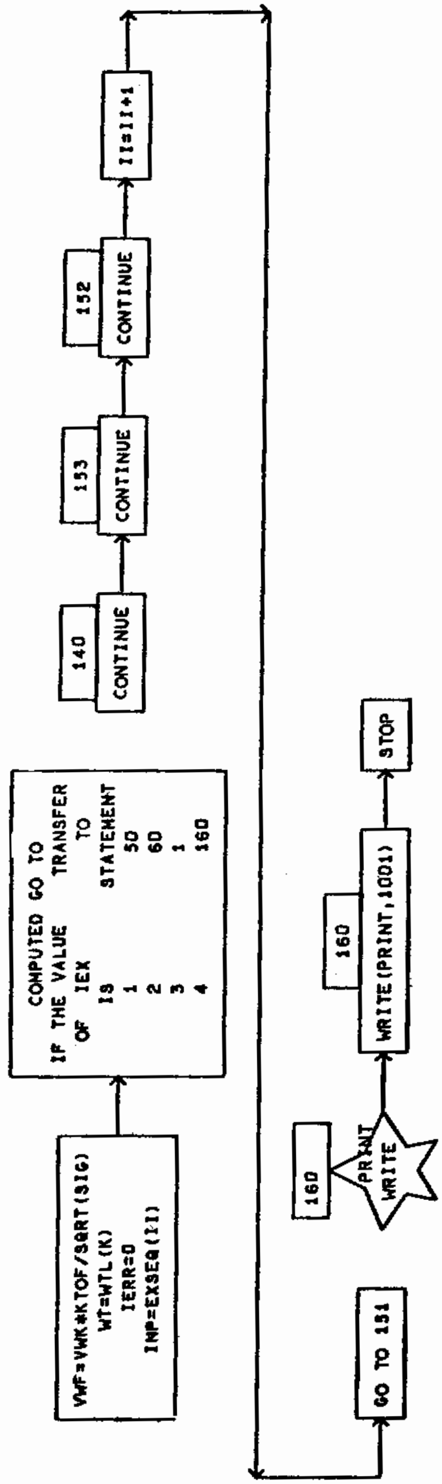
APPENDIX II

PROGRAM AND SUBROUTINE FLOW CHARTS

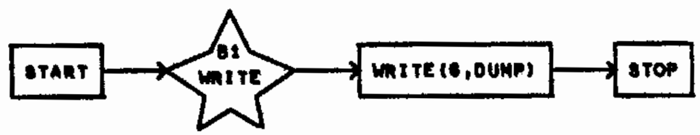
The following flow charts are contained in this appendix.

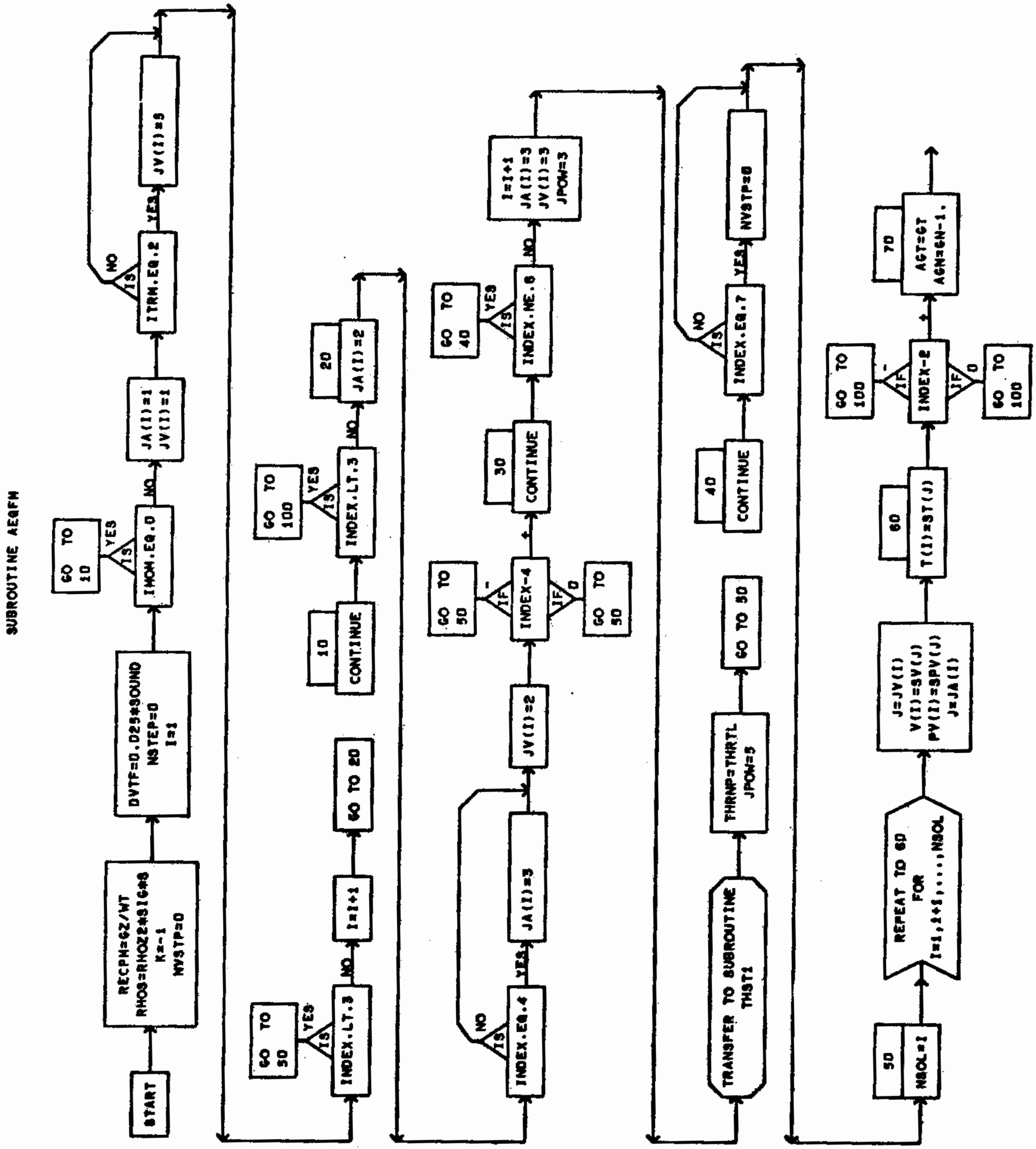
PROGRAM MILSTOL(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)



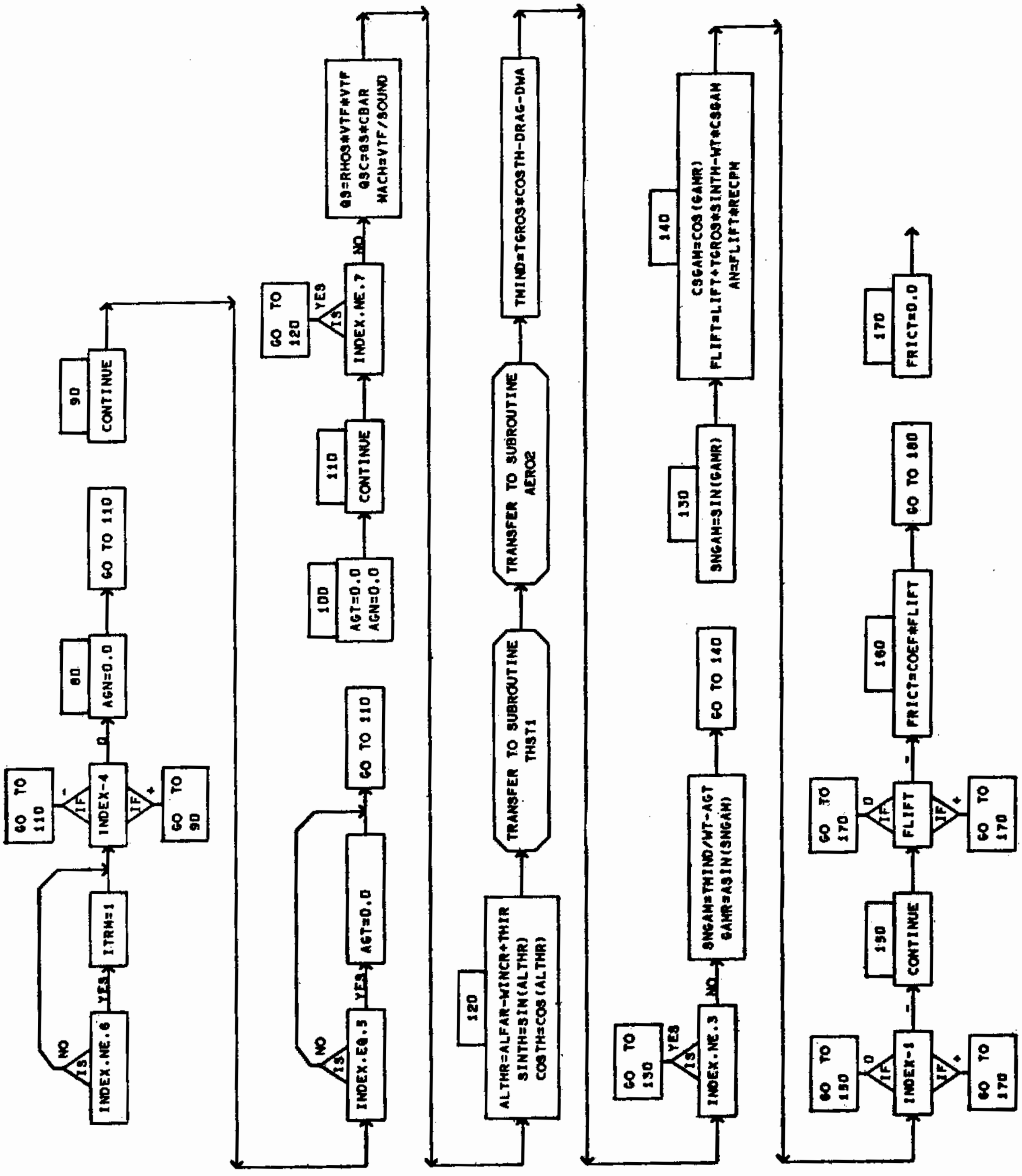

VWF=VWK*KTOF/SQRT(SIG)
WT=WTL(K)
IERR=0
INP=EXSEQ(II)
COMPUTED GO TO
IF THE VALUE OF IEX IS 1 2 3 4
TRANSFER TO STATEMENT 50 60 1 160
140
CONTINUE
153
CONTINUE
152
CONTINUE
II=II+1
GO TO 151
160
PRINT
WRITE
160
WRITE(PRINT,1001)
STOP

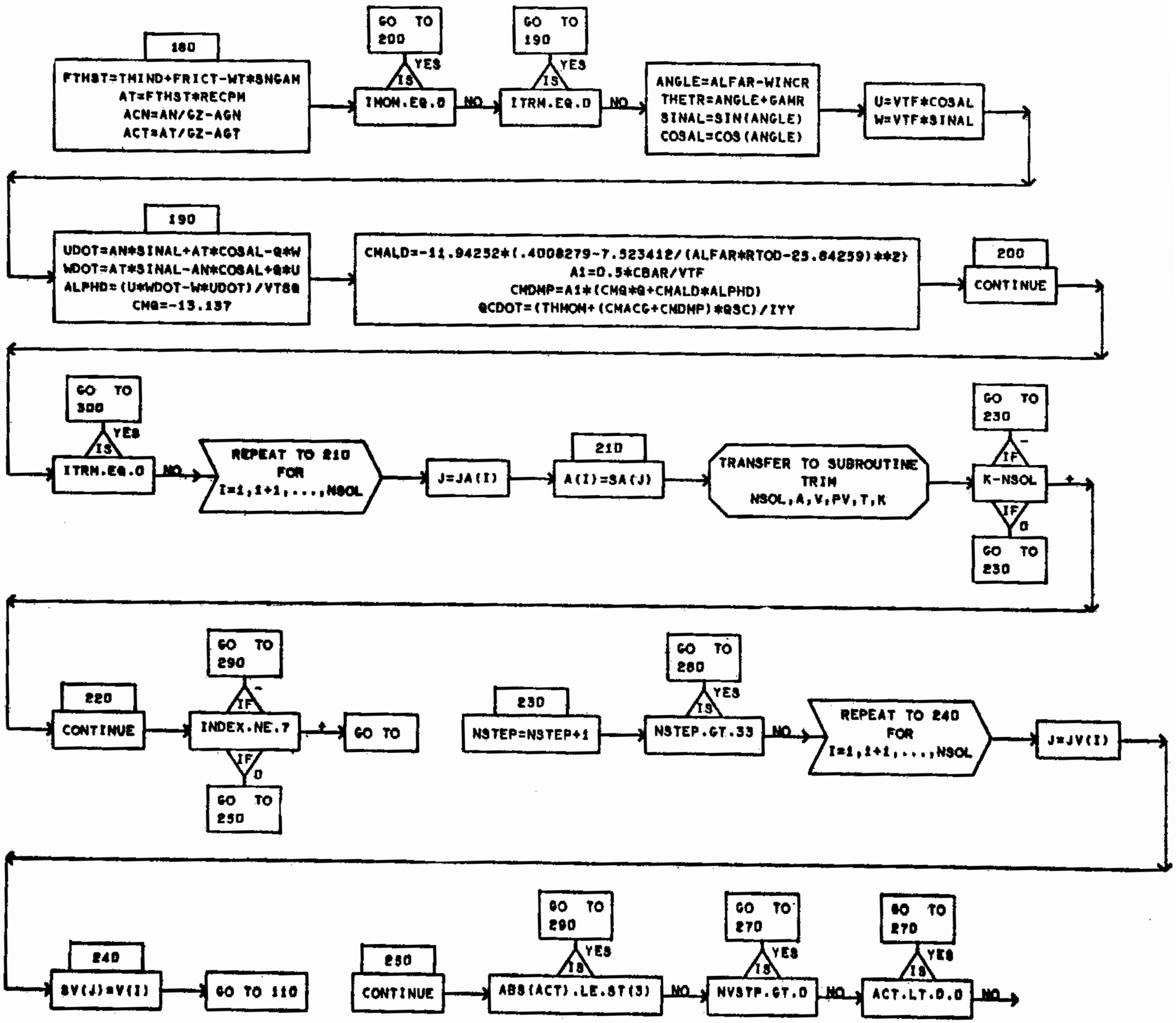
PROGRAM NILSTOL(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)

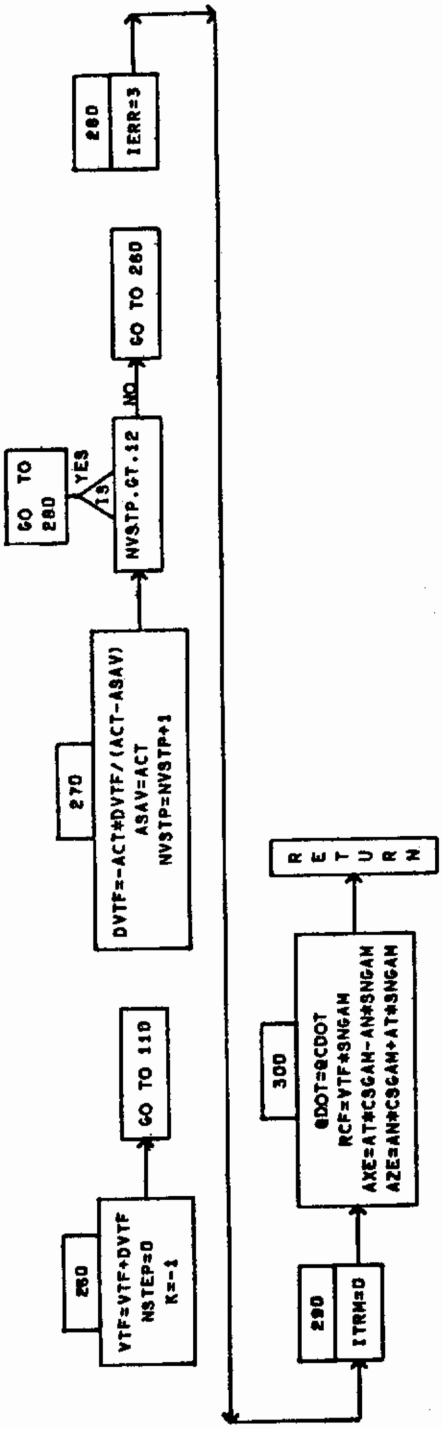




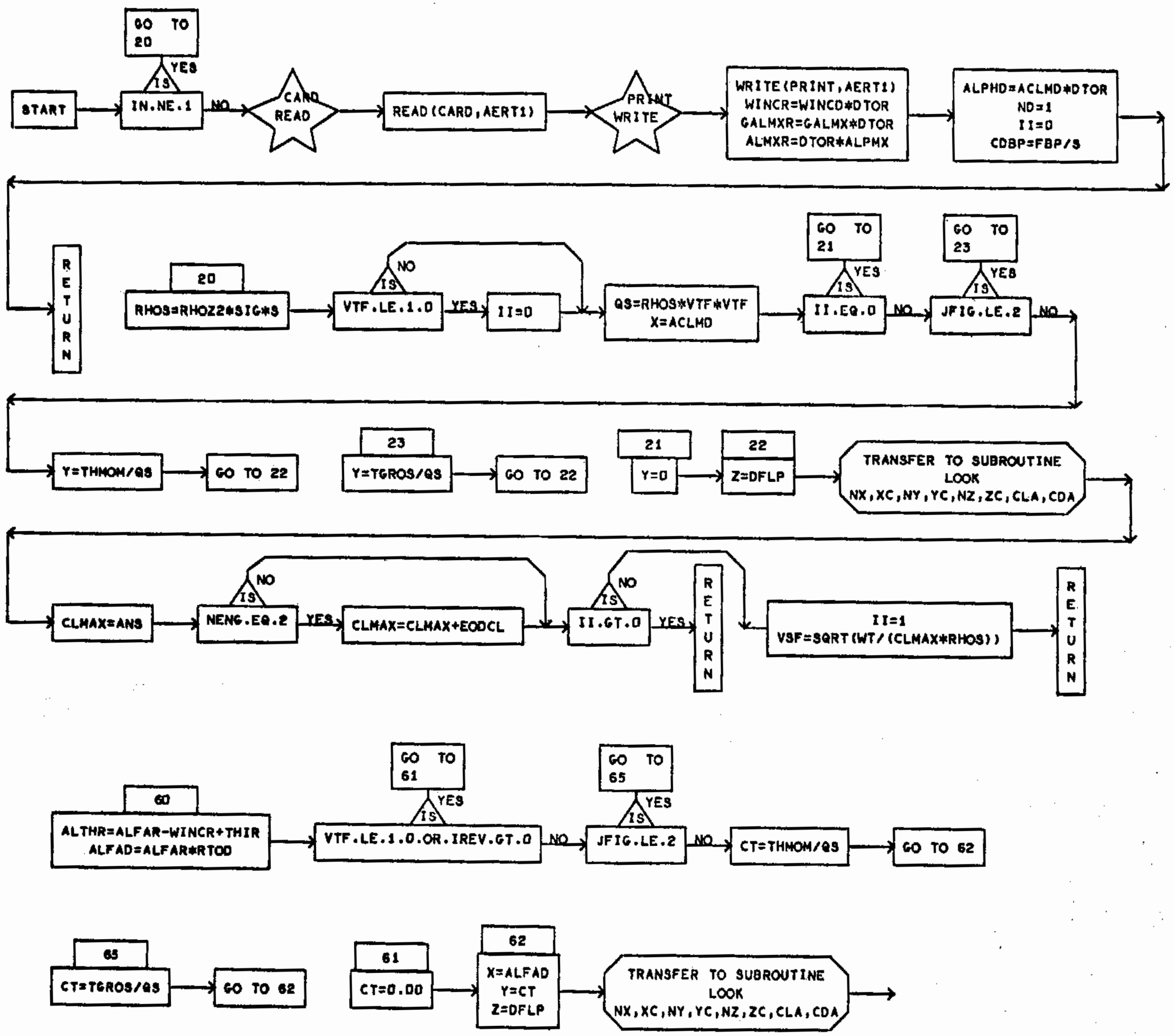
SUBROUTINE AEQFN
START
RECPH=GZ/WT
RHOS=RHOZ2*SIG*S
K=-1
NVSTP=0
DVTF=0.025*SOUND
NSTEP=0
I=1
IS INOM.EQ.0
YES
GO TO 10
NO
JA(I)=1
JV(I)=1
IS ITRM.EQ.2
YES
JV(I)=5
NO
IS INDEX.LT.3
YES
GO TO 50
NO
I=I+1
GO TO 20
10
CONTINUE
IS INDEX.LT.3
YES
GO TO 100
NO
20
JA(I)=2
IS INDEX.EQ.4
YES
JA(I)=5
NO
JV(I)=2
IF INDEX-4
-
GO TO 50
0
GO TO 50
+
30
CONTINUE
IS INDEX.NE.6
YES
GO TO 40
NO
I=I+1
JA(I)=3
JV(I)=3
JPOW=3
TRANSFER TO SUBROUTINE THST1
THRNP=THRTL
JPOW=5
GO TO 50
40
CONTINUE
IS INDEX.EQ.7
YES
NVSTP=8
NO
50
NSOL=1
REPEAT TO 60 FOR I=1,I+1,...,NSOL
J=JV(I)
V(I)=SV(J)
PV(I)=SPV(J)
J=JA(I)
60
T(I)=ST(J)
IF INDEX-2
-
GO TO 100
0
GO TO 100
+
70
AGT=GT
AGN=GN-1.

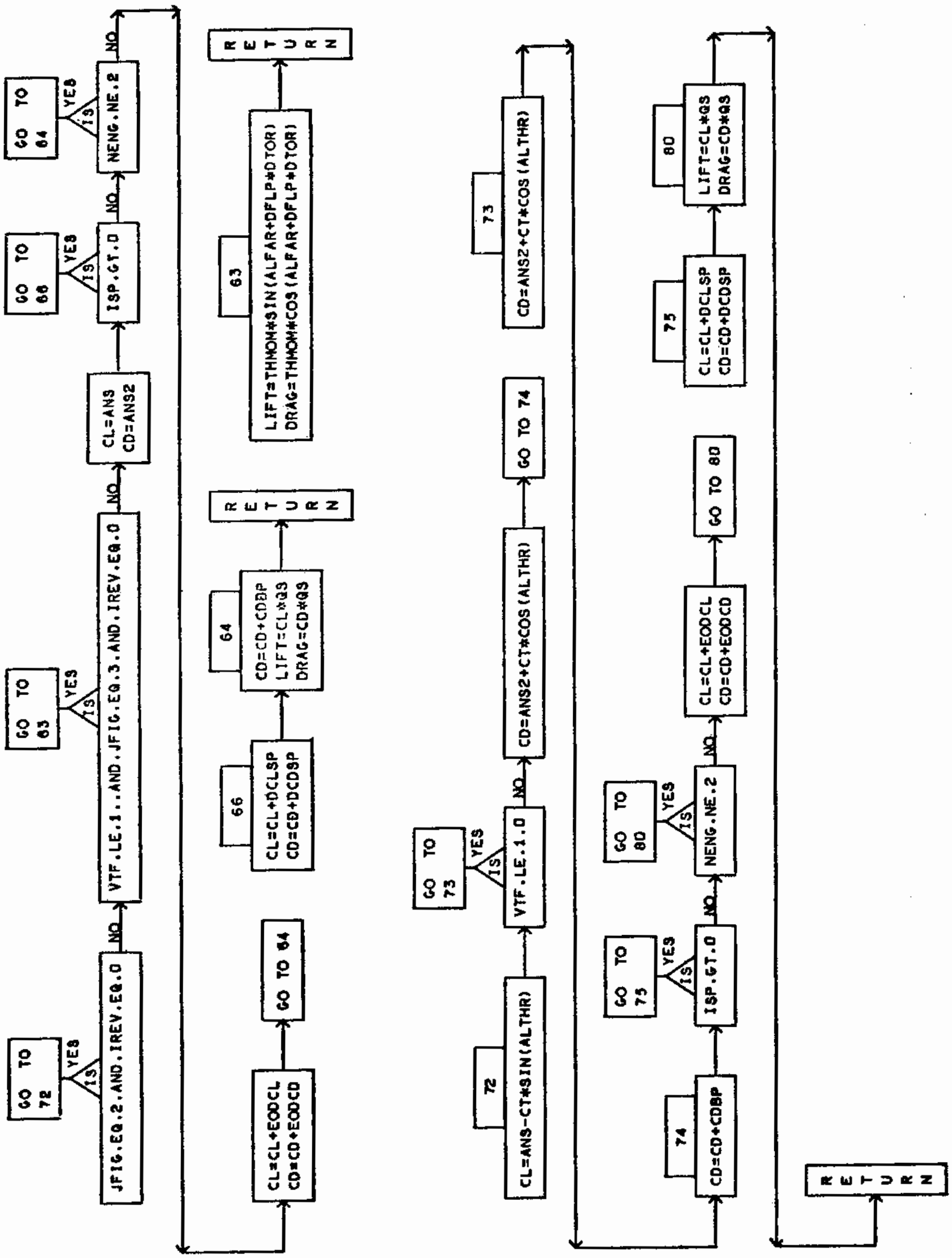

INDEX.NE.6
IS
NO
YES
ITRM=1
INDEX-4
IF
GO TO 110
GO TO 90
80
AGN=0.0
GO TO 110
90
CONTINUE
INDEX.EQ.5
AGT=0.0
GO TO 110
100
AGT=0.0
AGN=0.0
110
CONTINUE
INDEX.NE.7
GO TO 120
QS=RHOS*VTF*VTF
QSC=QS*CBAR
MACH=VTF/SOUND
120
ALTHR=ALFAR-WINCR+THIR
SINTH=SIN(ALTHR)
COSTH=COS(ALTHR)
TRANSFER TO SUBROUTINE THST1
TRANSFER TO SUBROUTINE AERO2
THIND=TGROS*COSTH-DRAG-DWA
INDEX.NE.3
GO TO 130
SNGAM=THIND/WT-AGT
GAMR=ASIN(SNGAM)
GO TO 140
130
SNGAM=SIN(GAMR)
140
CSGAM=COS(GAMR)
FLIFT=LIFT+TGROS*SINTH-WT*CSGAM
AN=FLIFT*RECPM
INDEX-1
GO TO 150
GO TO 170
150
CONTINUE
FLIFT
GO TO 170
GO TO 170
160
FRICT=COEF*FLIFT
GO TO 180
170
FRICT=0.0


180
FTHST=THIND+FRICT-WT*SNGAM
AT=FTHST*RECPM
ACN=AN/GZ-AGN
ACT=AT/GZ-AGT
IS
INOM.EQ.0
YES
GO TO 200
NO
IS
ITRM.EQ.0
YES
GO TO 190
NO
ANGLE=ALFAR-WINCR
THETR=ANGLE+GAMR
SINAL=SIN(ANGLE)
COSAL=COS(ANGLE)
U=VTF*COSAL
W=VTF*SINAL
190
UDOT=AN*SINAL+AT*COSAL-Q*W
WDOT=AT*SINAL-AN*COSAL+Q*U
ALPHD=(U*WDOT-W*UDOT)/VTSQ
CMQ=-13.137
CMALD=-11.94252*(.4008279-7.523412/(ALFAR*RTOD-25.84259)**2)
A1=0.5*CBAR/VTF
CMDMP=A1*(CMQ*Q+CMALD*ALPHD)
QCDOT=(THMOM+(CMACG+CMDMP)*QSC)/IYY
200
CONTINUE
IS
ITRM.EQ.0
YES
GO TO 300
NO
REPEAT TO 210
FOR
I=1,1+1,...,NSOL
J=JA(I)
210
A(I)=SA(J)
TRANSFER TO SUBROUTINE
TRIM
NSOL,A,V,PV,T,K
IF
K-NSOL
-
GO TO 230
0
GO TO 230
+
220
CONTINUE
IF
INDEX.NE.7
-
GO TO 290
0
GO TO 250
+
GO TO
230
NSTEP=NSTEP+1
IS
NSTEP.GT.33
YES
GO TO 280
NO
REPEAT TO 240
FOR
I=1,1+1,...,NSOL
J=JV(I)
240
SV(J)=V(I)
GO TO 110
250
CONTINUE
IS
ABS(ACT).LE.ST(3)
YES
GO TO 290
NO
IS
NVSTP.GT.0
YES
GO TO 270
NO
IS
ACT.LT.0.0
YES
GO TO 270
NO

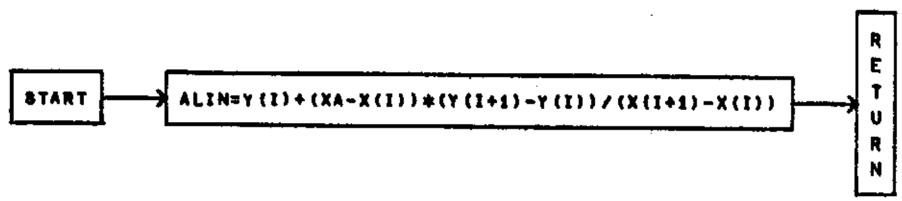

280
IERR=3
GO TO 260
NVSTP.GT.12
IS
YES
NO
GO TO 280
270
DVTF=-ACT*DVTF/(ACT-ASAV)
ASAV=ACT
NVSTP=NVSTP+1
260
VTF=VTF+DVTF
NSTEP=0
K=-1
GO TO 110
290
ITRM=0
300
QDOT=QCDOT
RCF=VTF*SNGAM
AXE=AT*CSGAM-AN*SNGAM
AZE=AN*CSGAM+AT*SNGAM
RETURN

SUBROUTINE AERO1

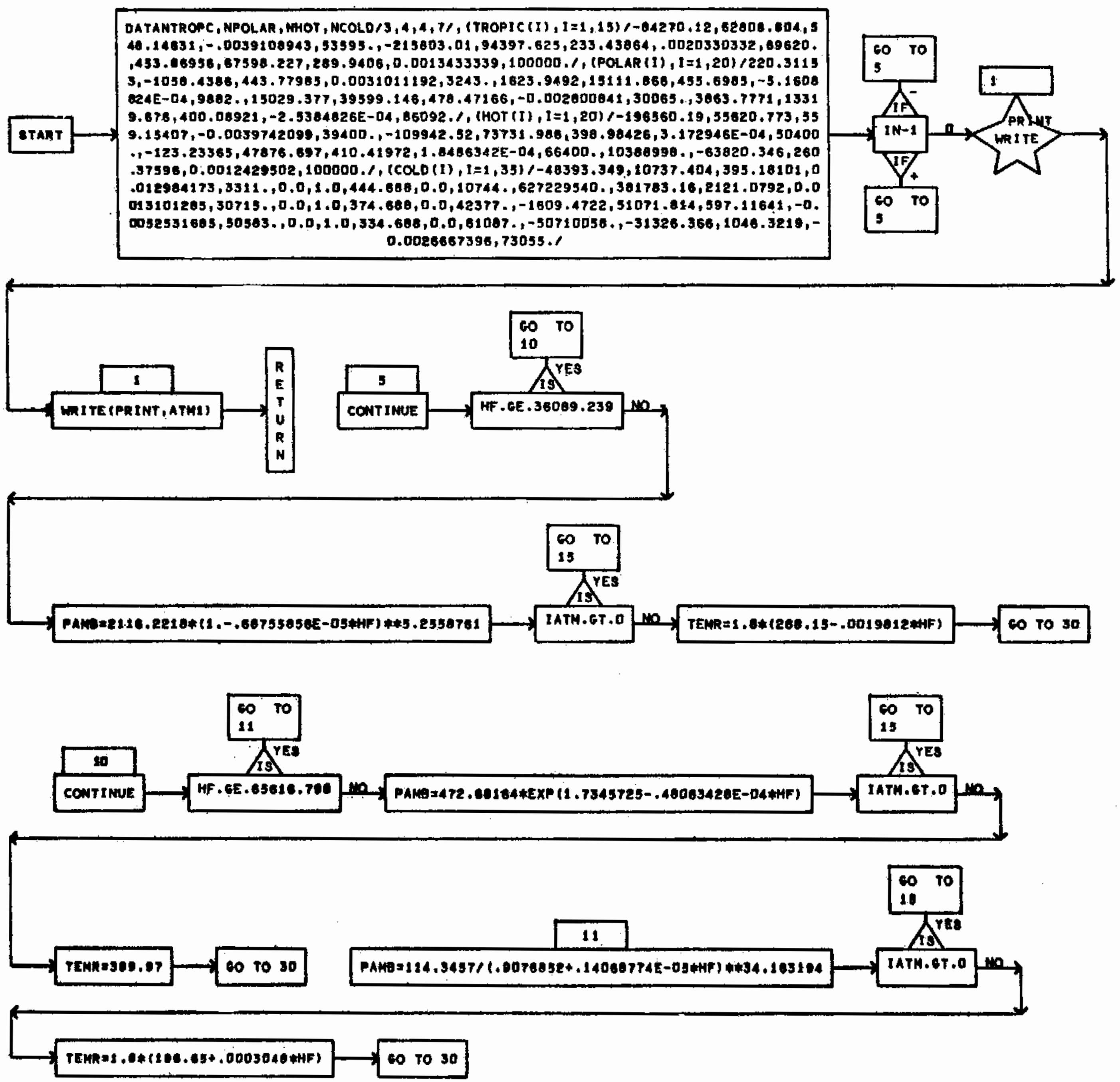


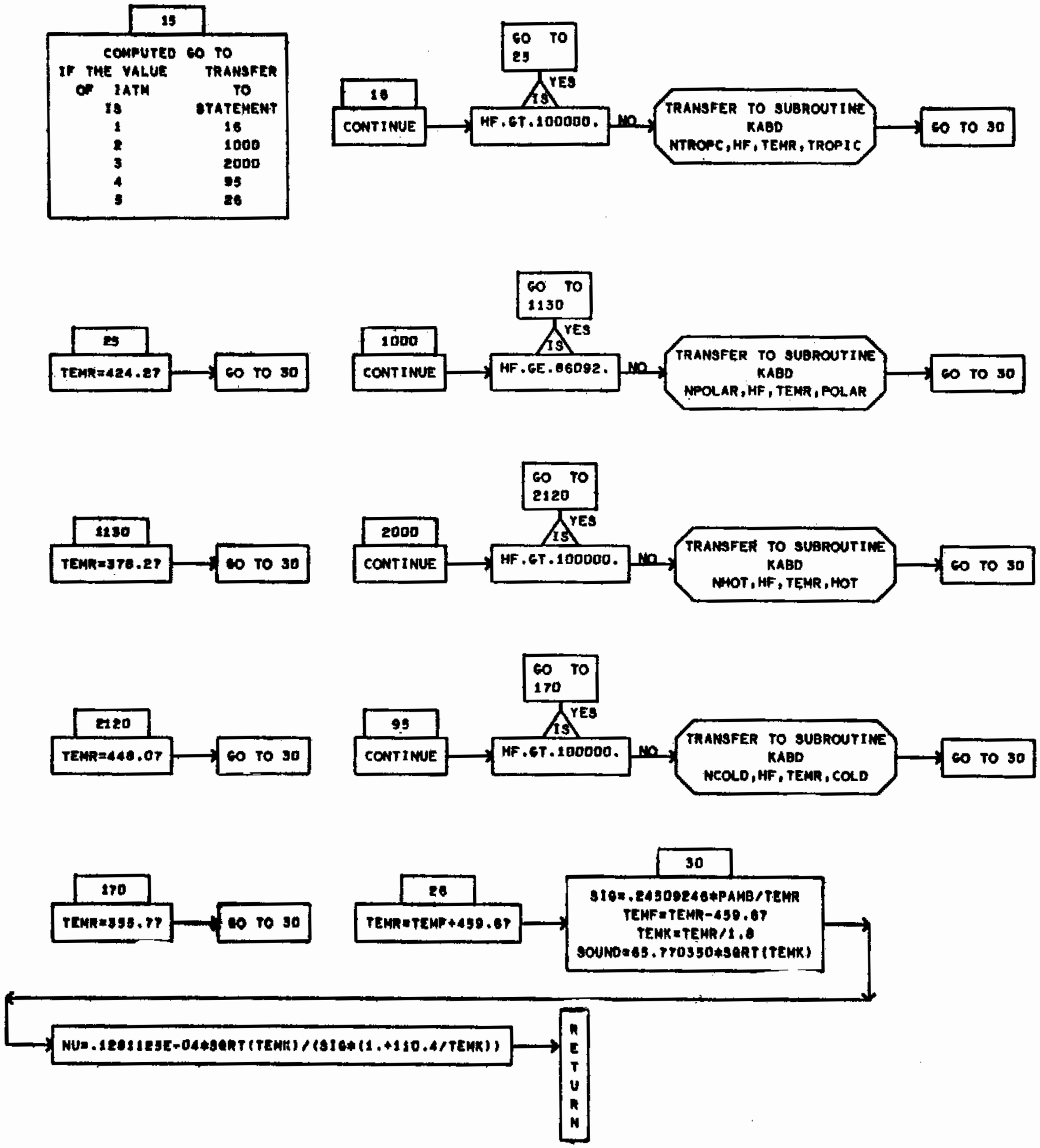

JFIG.EQ.2.AND.IREV.EQ.0
IS
YES
GO TO 72
NO
VTF.LE.1..AND.JFIG.EQ.3.AND.IREV.EQ.0
GO TO 63
CL=ANS
CD=ANS2
ISP.GT.0
GO TO 66
NENG.NE.2
GO TO 64
CL=CL+EODCL
CD=CD+EODCD
GO TO 64
66
CL=CL+DCLSP
CD=CD+DCDSP
64
CD=CD+CDBP
LIFT=CL*QS
DRAG=CD*QS
RETURN
63
LIFT=THMOM*SIN(ALFAR+DFLP*DTOR)
DRAG=THMOM*COS(ALFAR+DFLP*DTOR)
RETURN
72
CL=ANS-CT*SIN(ALTHR)
VTF.LE.1.0
GO TO 73
CD=ANS2+CT*COS(ALTHR)
GO TO 74
73
CD=ANS2+CT*COS(ALTHR)
74
CD=CD+CDBP
ISP.GT.0
GO TO 75
NENG.NE.2
GO TO 80
CL=CL+EODCL
CD=CD+EODCD
GO TO 80
75
CL=CL+DCLSP
CD=CD+DCDSP
80
LIFT=CL*QS
DRAG=CD*QS
RETURN

FUNCTIONALIN(I,X,Y,XA)

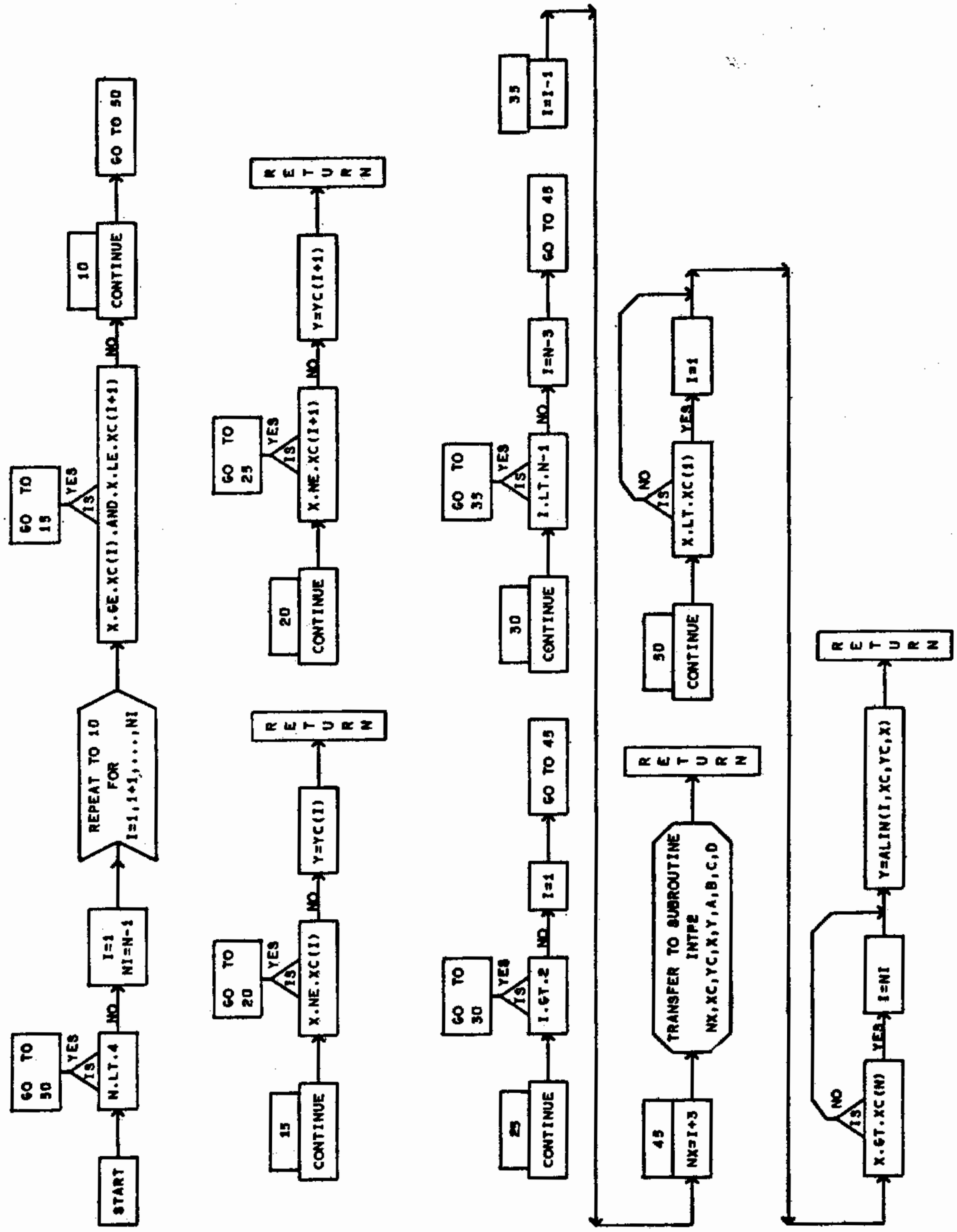


SUBROUTINE ATMOS

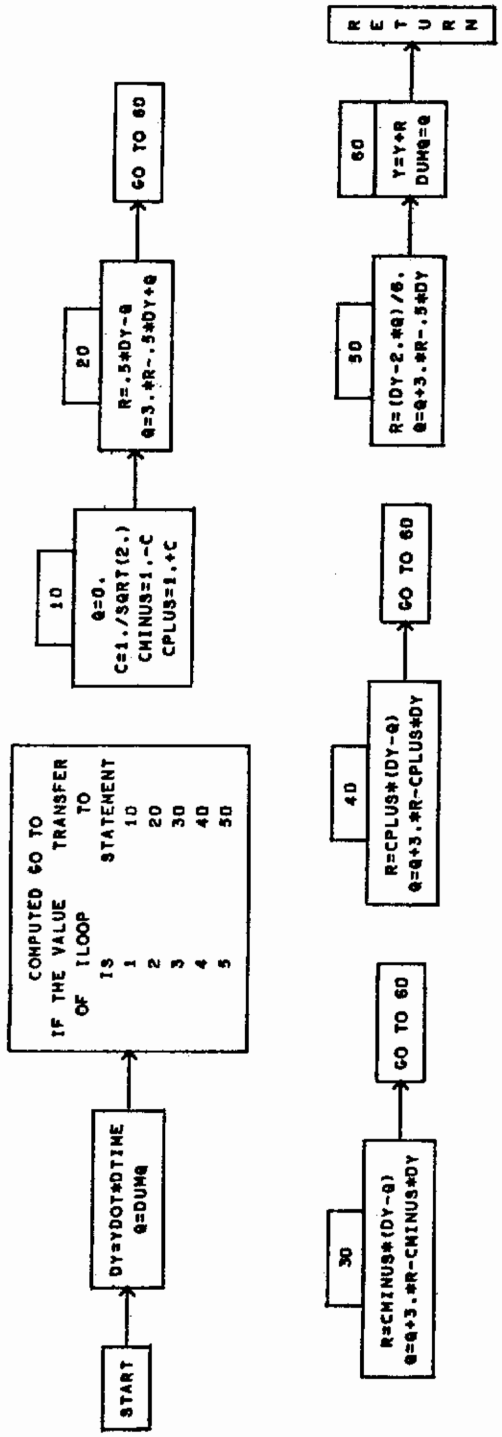



15
COMPUTED GO TO
IF THE VALUE OF IATM IS | TRANSFER TO STATEMENT
1 | 16
2 | 1000
3 | 2000
4 | 95
5 | 26
16
CONTINUE
HF.GT.100000.
IS
YES
GO TO 25
NO
TRANSFER TO SUBROUTINE KABD NTROPC,HF,TEMR,TROPIC
GO TO 30
25
TEMR=424.27
GO TO 30
1000
CONTINUE
HF.GE.86092.
GO TO 1130
NO
TRANSFER TO SUBROUTINE KABD NPOLAR,HF,TEMR,POLAR
GO TO 30
1130
TEMR=376.27
GO TO 30
2000
CONTINUE
HF.GT.100000.
GO TO 2120
NO
TRANSFER TO SUBROUTINE KABD NHOT,HF,TEMR,HOT
GO TO 30
2120
TEMR=448.07
GO TO 30
95
CONTINUE
HF.GT.100000.
GO TO 170
NO
TRANSFER TO SUBROUTINE KABD NCOLD,HF,TEMR,COLD
GO TO 30
170
TEMR=355.77
GO TO 30
26
TEMR=TEMF+459.67
30
SIG=.24509246*PAMB/TEMR
TEMF=TEMR-459.67
TEMK=TEMR/1.8
SOUND=65.770350*SQRT(TEMK)
NU=.1281125E-04*SQRT(TEMK)/(SIG*(1.+110.4/TEMK))
RETURN

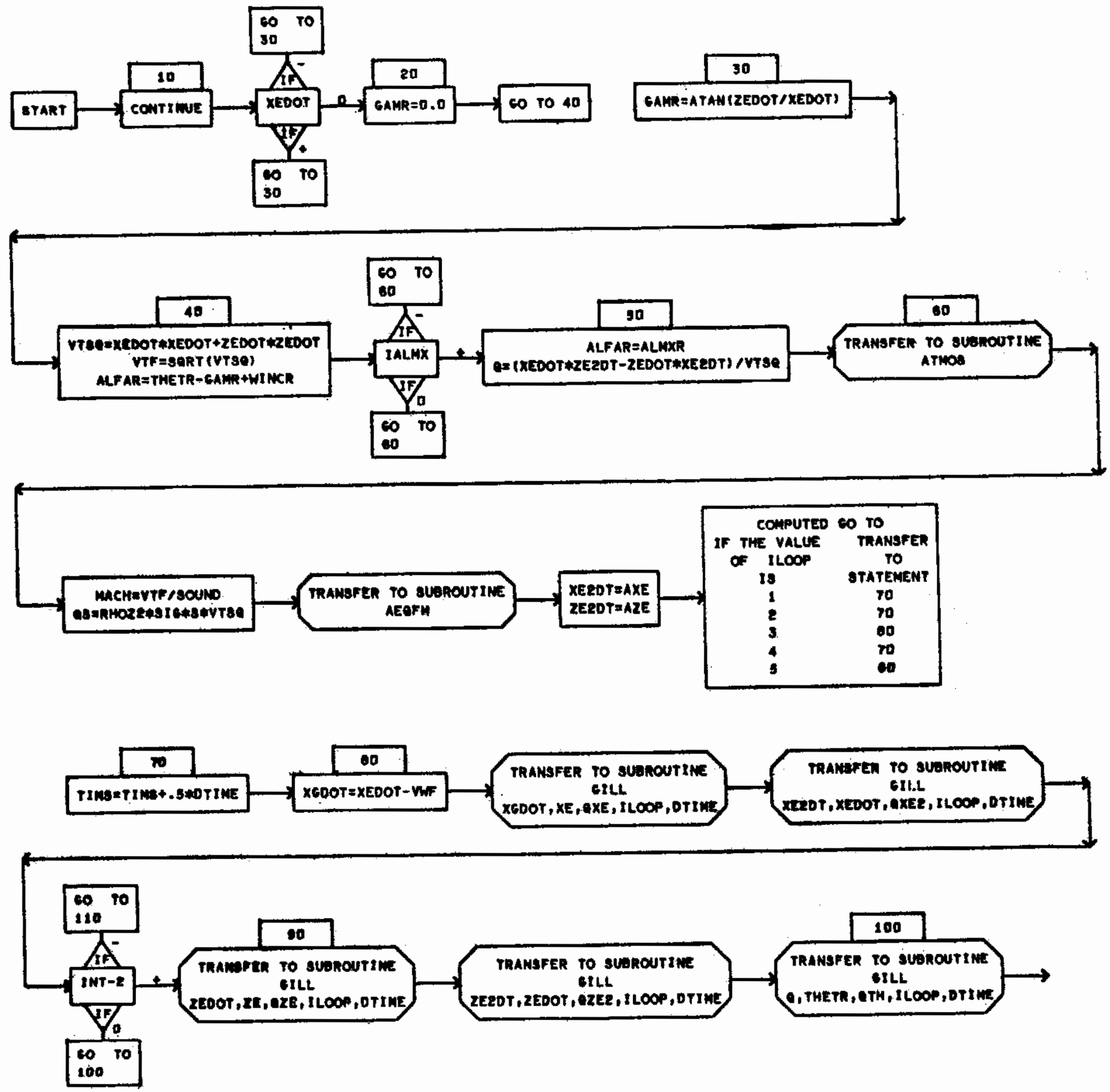
SUBROUTINE FIND(N,X,Y,XC,YC)

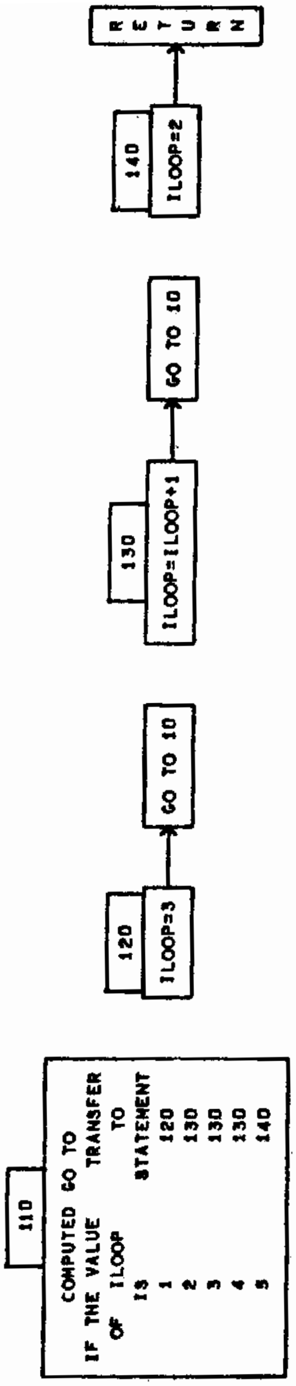


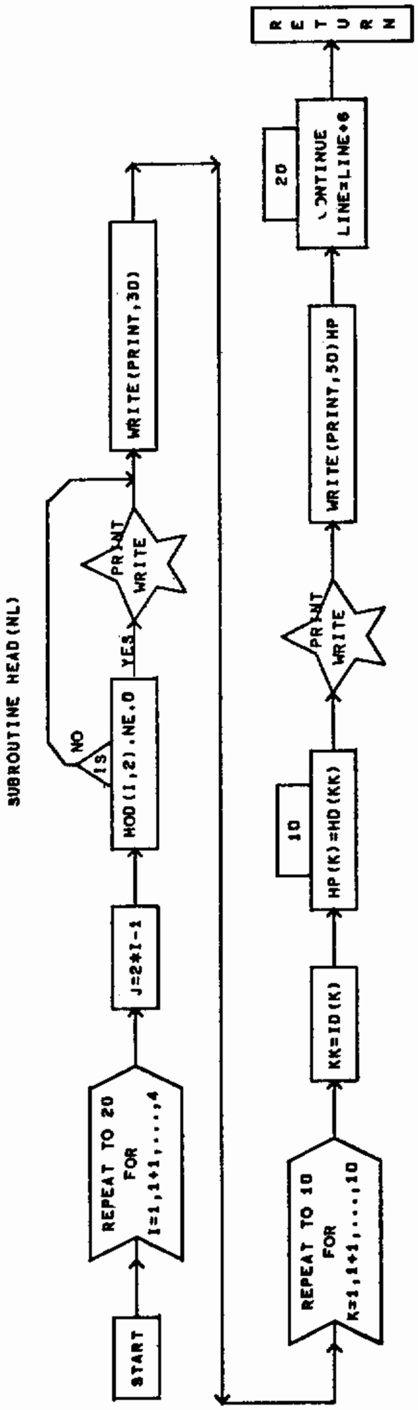
SUBROUTINE GILL(YDOT,Y,DUMQ,ILOOP,DTIME)



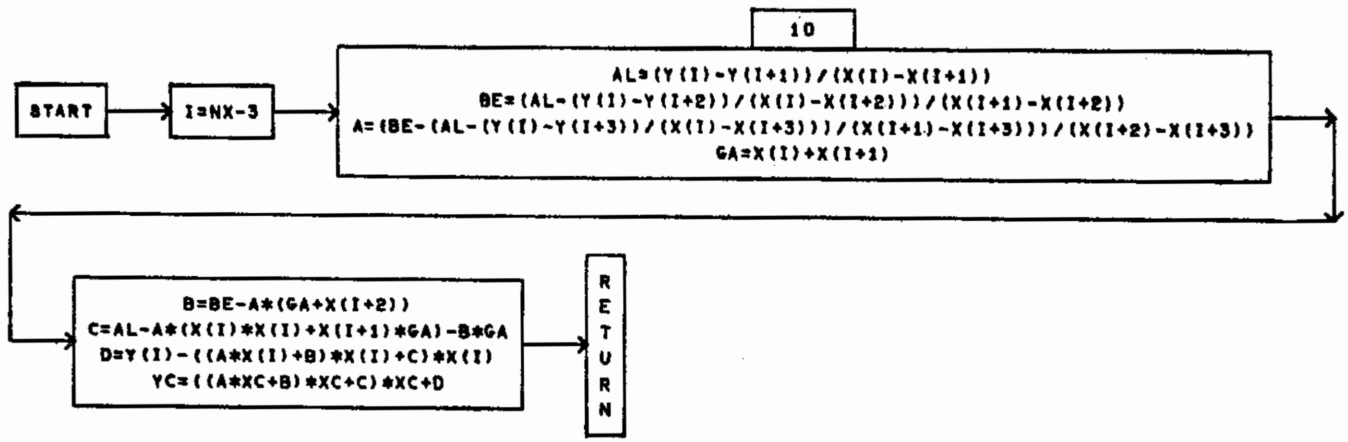
SUBROUTINE GINTG



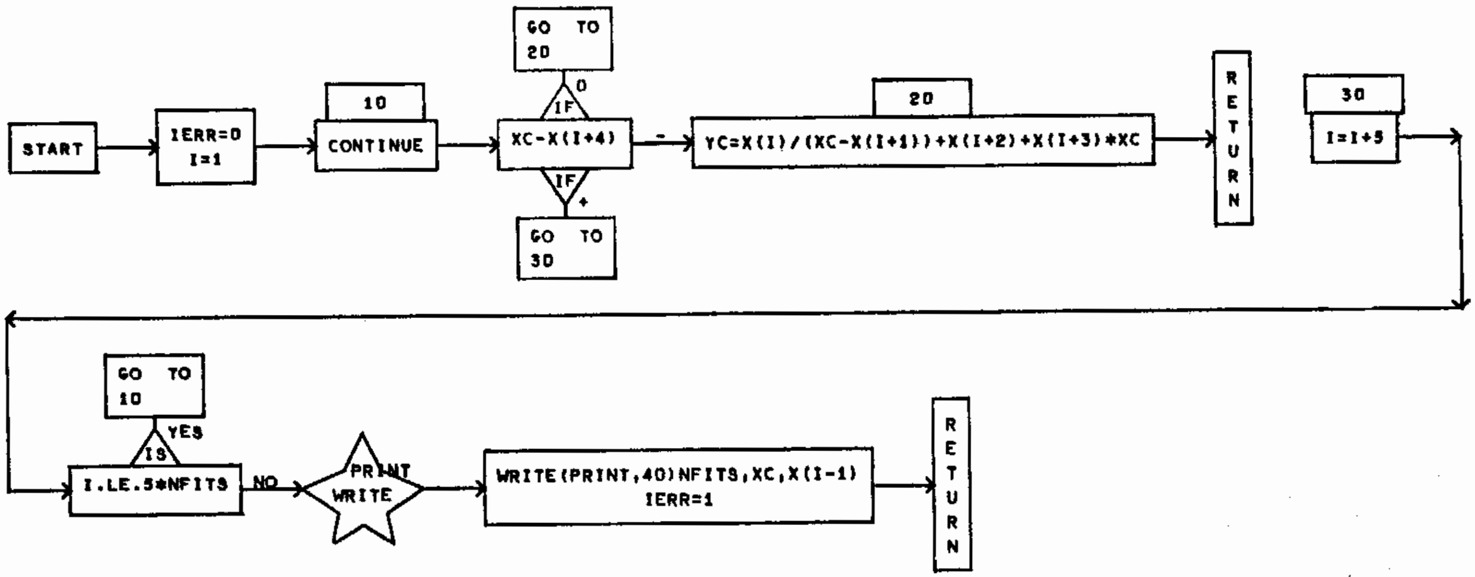

110
COMPUTED GO TO
IF THE VALUE OF ILOOP IS
1
2
3
4
5
TRANSFER TO STATEMENT
120
130
130
130
140
120
ILOOP=3
GO TO 10
130
ILOOP=ILOOP+1
GO TO 10
140
ILOOP=2
RETURN


SUBROUTINE HEAD(NL)
START
REPEAT TO 20
FOR
I=1,I+1,...,4
J=2*I-1
IS
MOD(I,2).NE.0
NO
YES
PRINT
WRITE
WRITE(PRINT,30)
REPEAT TO 10
FOR
K=1,1+1,...,10
KK=ID(K)
10
HP(K)=HD(KK)
PRINT
WRITE
WRITE(PRINT,50)HP
20
CONTINUE
LINE=LINE+6
RETURN

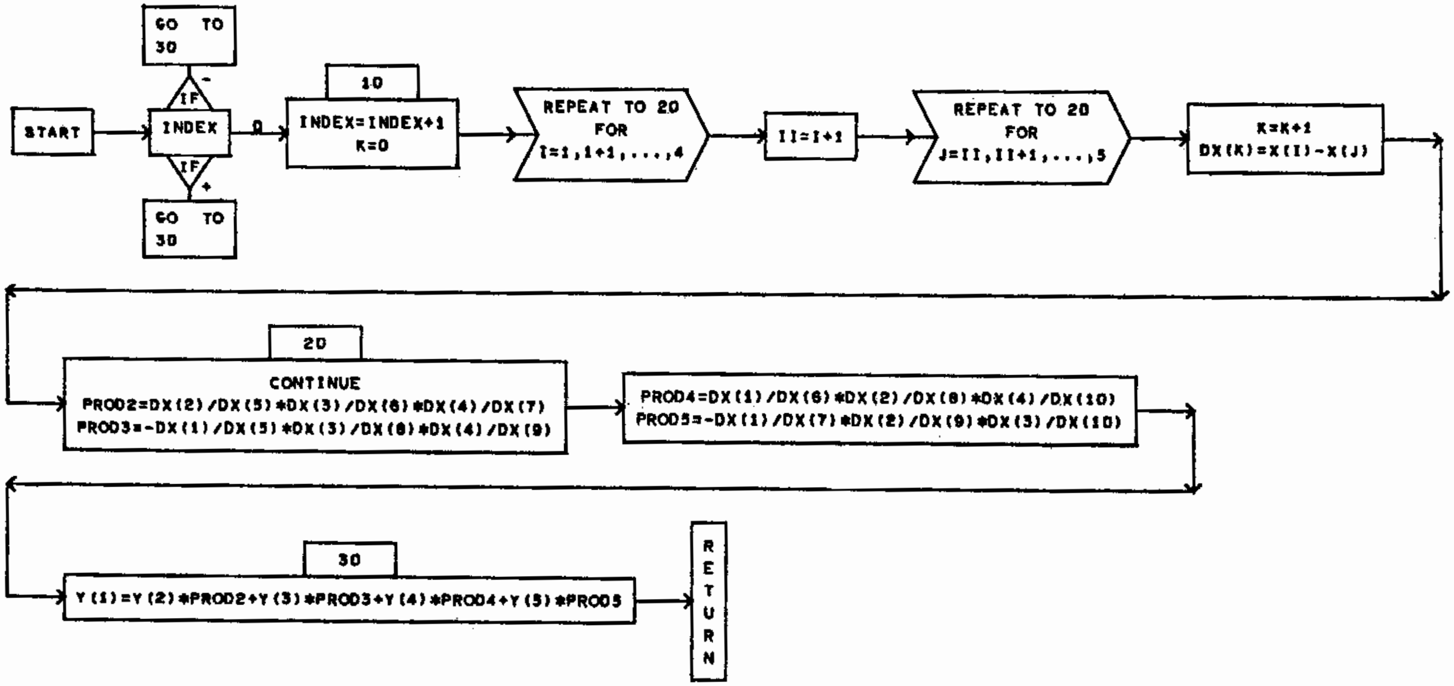
SUBROUTINE INTP2(NX,X,Y,XC,YC,A,B,C,D)



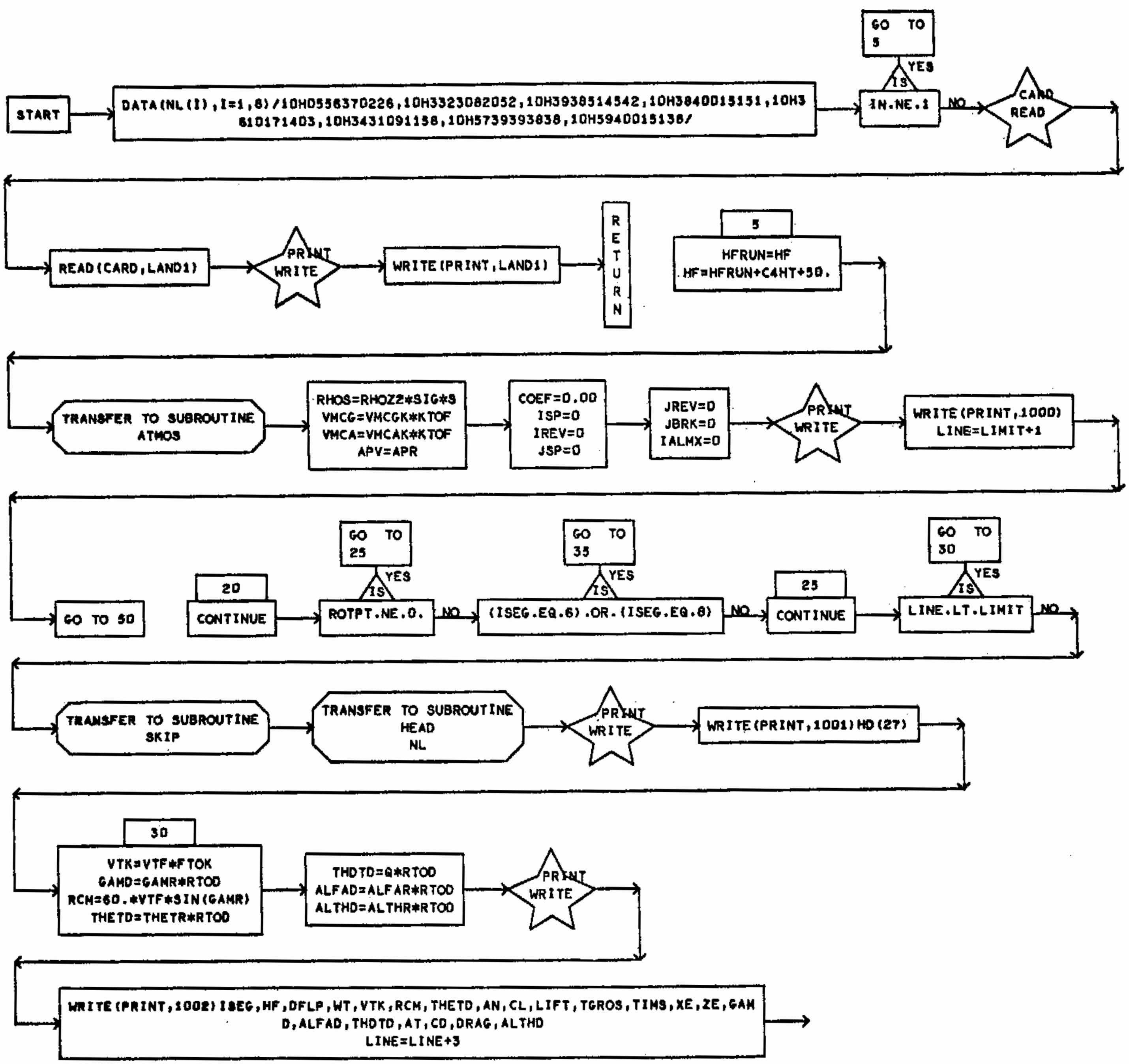
SUBROUTINE KABD(NFITS,XC,YC,X)

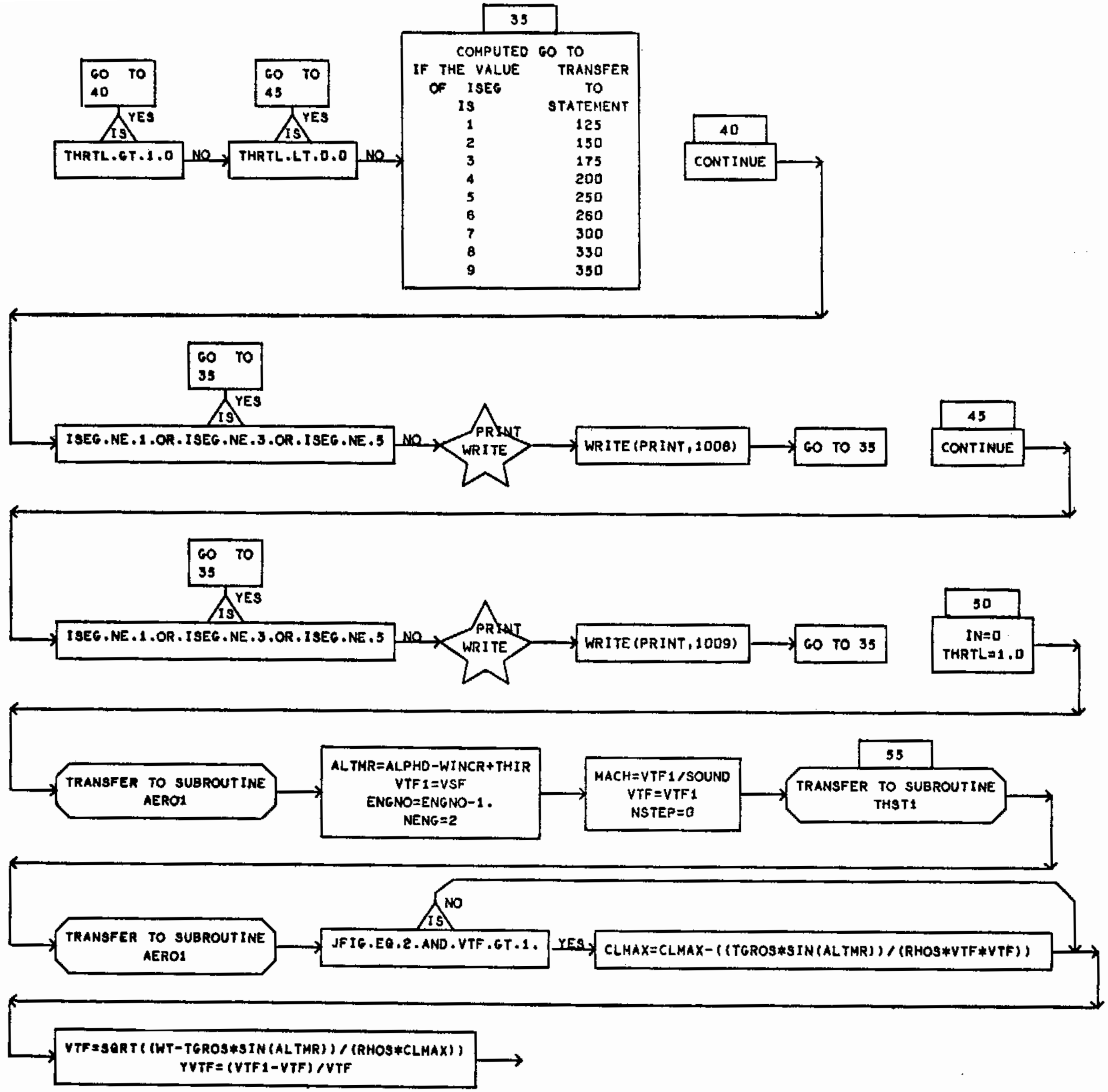


SUBROUTINE LAGRA(X,Y,INDEX)

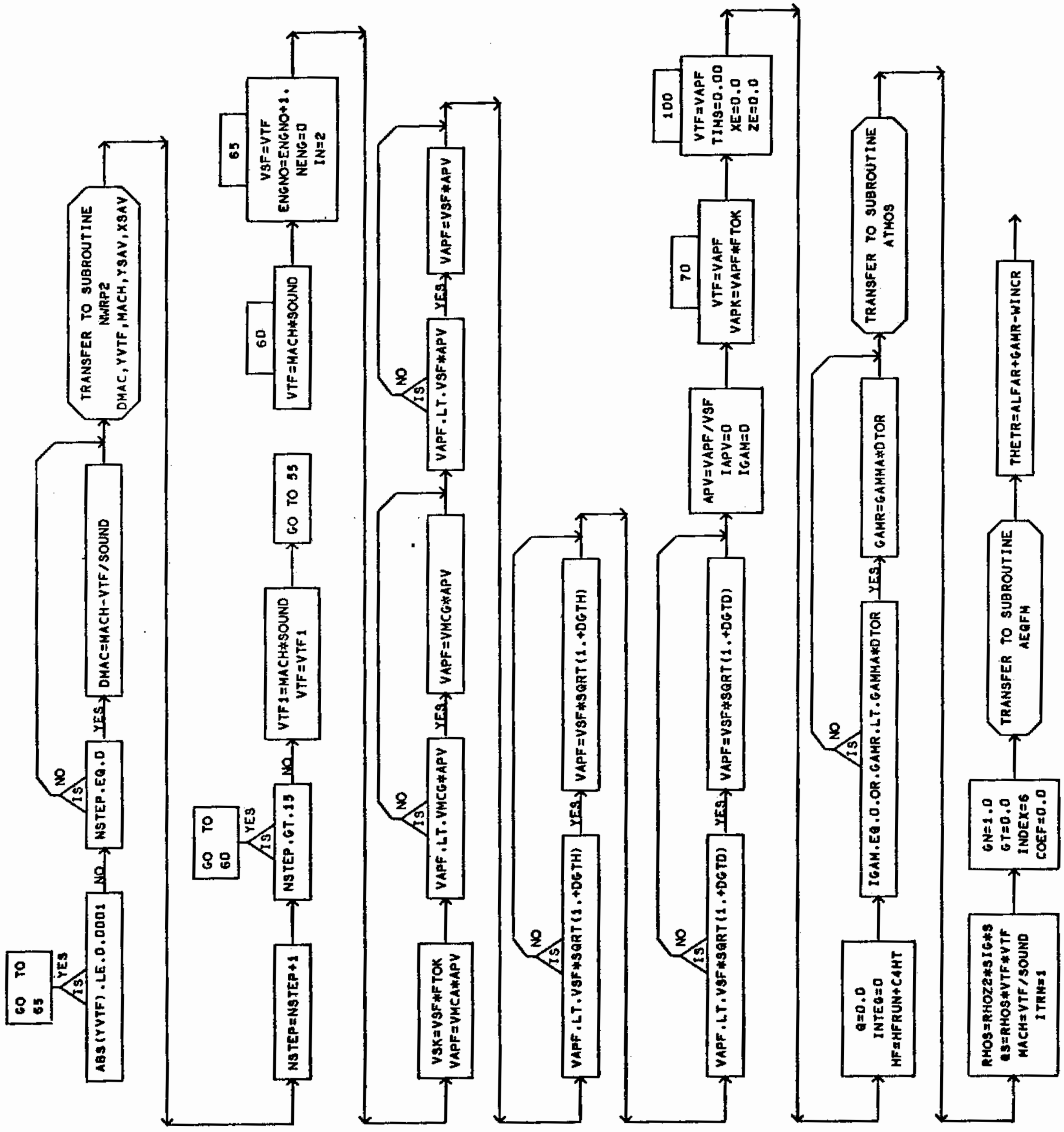


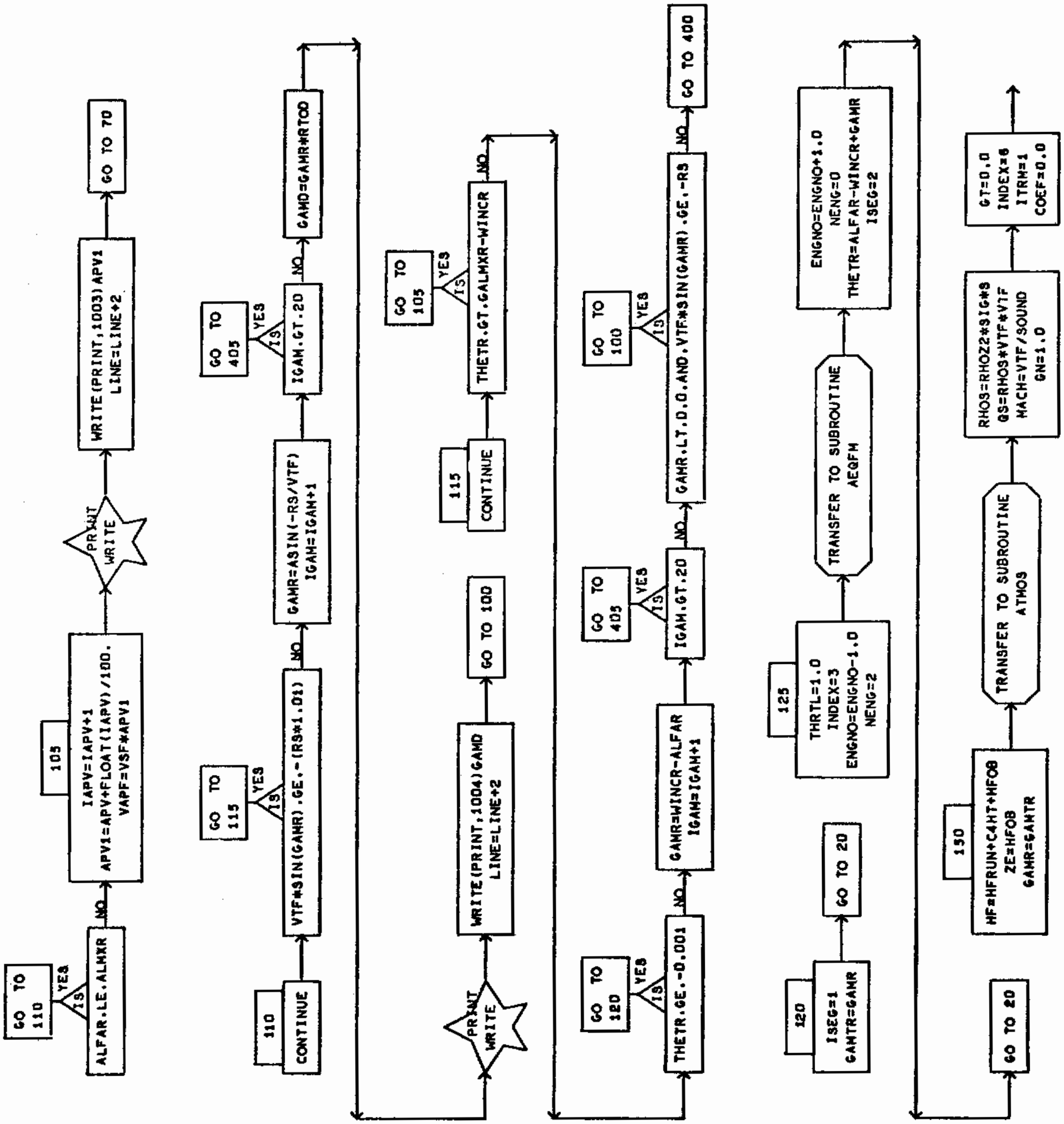
SUBROUTINE LANDING



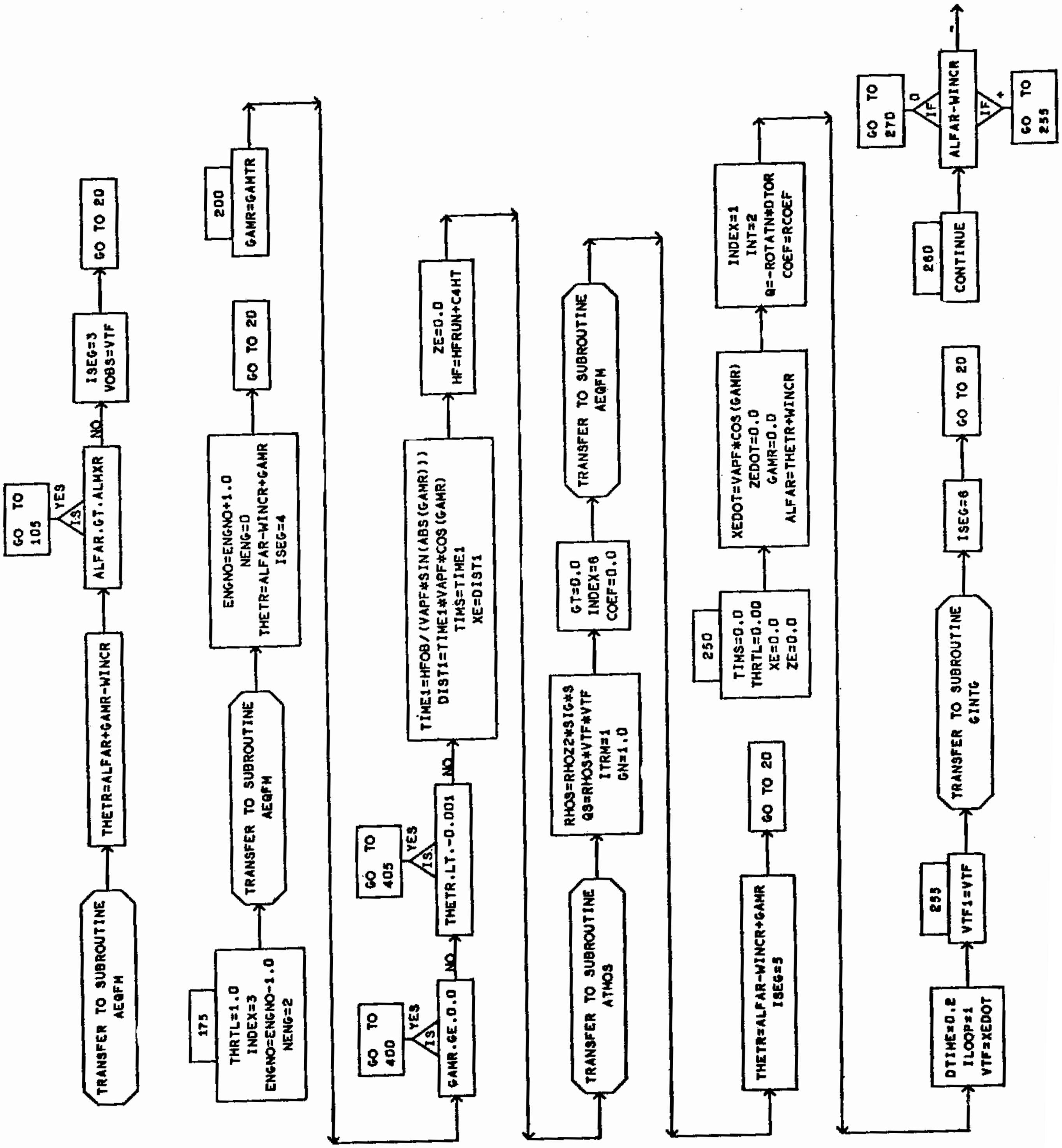


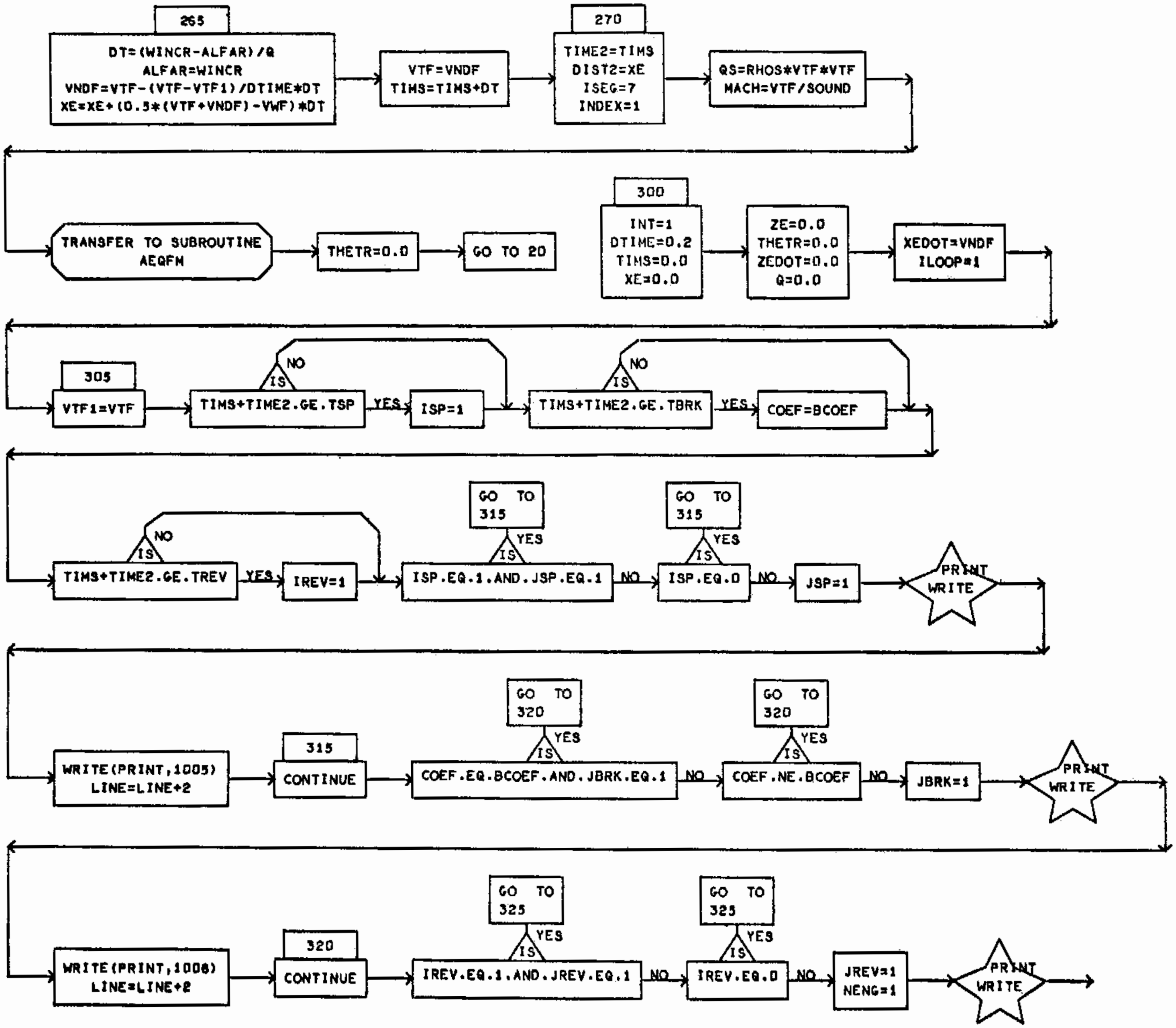
GO TO 40
YES
IS
THRTL.GT.1.0
NO
GO TO 45
YES
IS
THRTL.LT.0.0
NO
35
COMPUTED GO TO
IF THE VALUE OF ISEG IS 1 2 3 4 5 6 7 8 9
TRANSFER TO STATEMENT 125 150 175 200 250 260 300 330 350
40
CONTINUE
GO TO 35
YES
IS
ISEG.NE.1.OR.ISEG.NE.3.OR.ISEG.NE.5
NO
PRINT
WRITE
WRITE(PRINT,1008)
GO TO 35
45
CONTINUE
GO TO 35
YES
IS
ISEG.NE.1.OR.ISEG.NE.3.OR.ISEG.NE.5
NO
PRINT
WRITE
WRITE(PRINT,1009)
GO TO 35
50
IN=0
THRTL=1.0
TRANSFER TO SUBROUTINE AERO1
ALTMR=ALPHD-WINCR+THIR
VTF1=VSF
ENGNO=ENGNO-1.
NENG=2
MACH=VTF1/SOUND
VTF=VTF1
NSTEP=0
55
TRANSFER TO SUBROUTINE THST1
TRANSFER TO SUBROUTINE AERO1
NO
IS
JFIG.EQ.2.AND.VTF.GT.1.
YES
CLMAX=CLMAX-((TGROS*SIN(ALTMR))/(RHOS*VTF*VTF))
VTF=SQRT((WT-TGROS*SIN(ALTMR))/(RHOS*CLMAX))
YVTF=(VTF1-VTF)/VTF

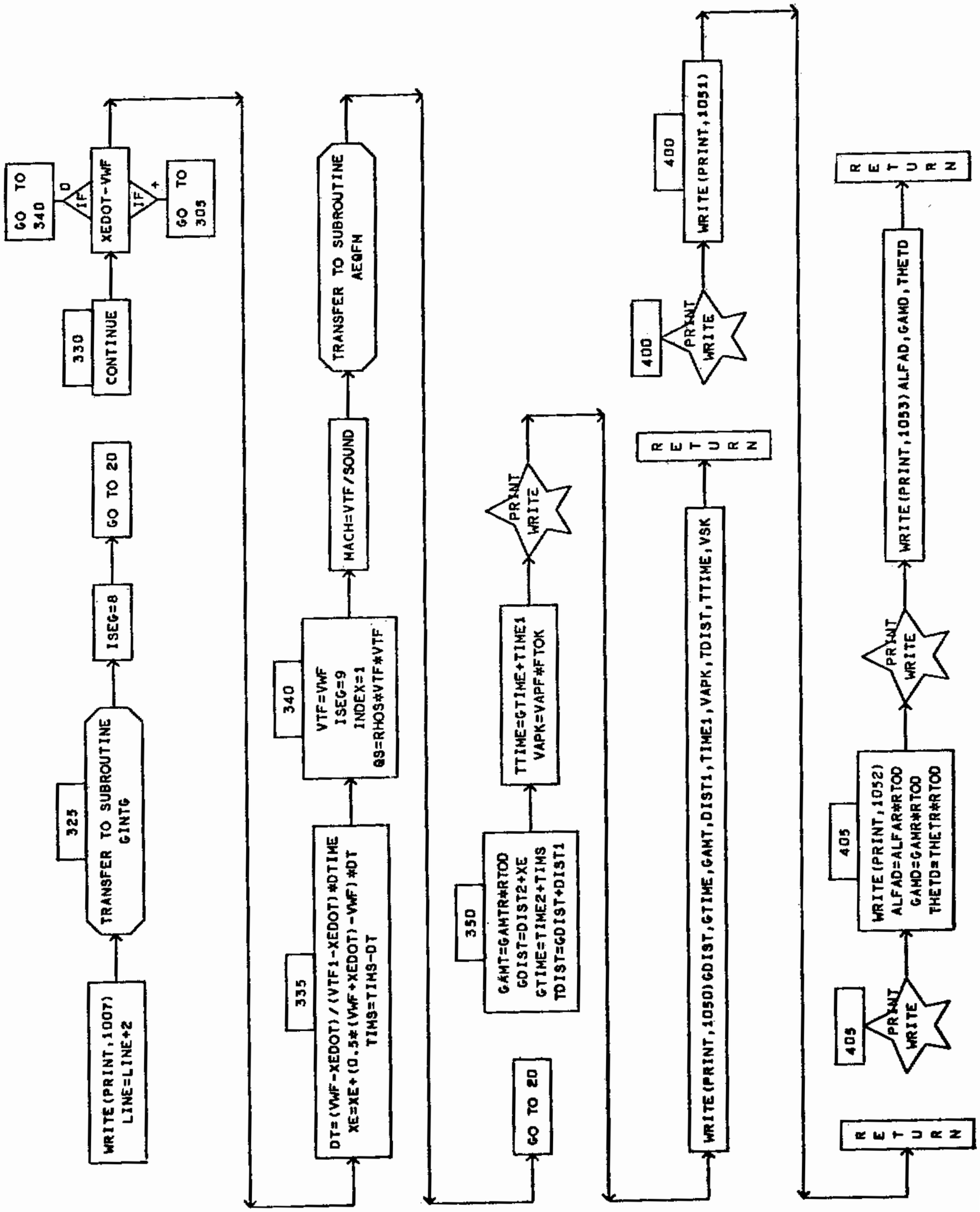

GO TO 65
YES
IS
ABS(YVTF).LE.0.0001
NO
IS
NSTEP.EQ.0
YES
NO
DMAC=MACH-VTF/SOUND
TRANSFER TO SUBROUTINE NWRP2
DMAC,YVTF,MACH,YSAV,XSAV
NSTEP=NSTEP+1
IS
NSTEP.GT.15
YES
GO TO 60
NO
VTF1=MACH*SOUND
VTF=VTF1
GO TO 55
60
VTF=MACH*SOUND
65
VSF=VTF
ENGNO=ENGNO+1.
NENG=0
IN=2
VSK=VSF*FTOK
VAPF=VMCA*APV
IS
VAPF.LT.VMCG*APV
NO
YES
VAPF=VMCG*APV
IS
VAPF.LT.VSF*APV
NO
YES
VAPF=VSF*APV
IS
VAPF.LT.VSF*SQRT(1.+DGTH)
NO
YES
VAPF=VSF*SQRT(1.+DGTH)
IS
VAPF.LT.VSF*SQRT(1.+DGTD)
NO
YES
VAPF=VSF*SQRT(1.+DGTD)
APV=VAPF/VSF
IAPV=0
IGAM=0
70
VTF=VAPF
VAPK=VAPF*FTOK
100
VTF=VAPF
TIMS=0.00
XE=0.0
ZE=0.0
Q=0.0
INTEG=0
HF=HFRUN+C4HT
IS
IGAM.EQ.0.OR.GAMR.LT.GAMMA*DTOR
NO
YES
GAMR=GAMMA*DTOR
TRANSFER TO SUBROUTINE
ATMOS
RHOS=RHOZ2*SIG*S
QS=RHOS*VTF*VTF
MACH=VTF/SOUND
ITRN=1
GN=1.0
GT=0.0
INDEX=6
COEF=0.0
TRANSFER TO SUBROUTINE
AEQFM
THETR=ALFAR+GAMR-WINCR

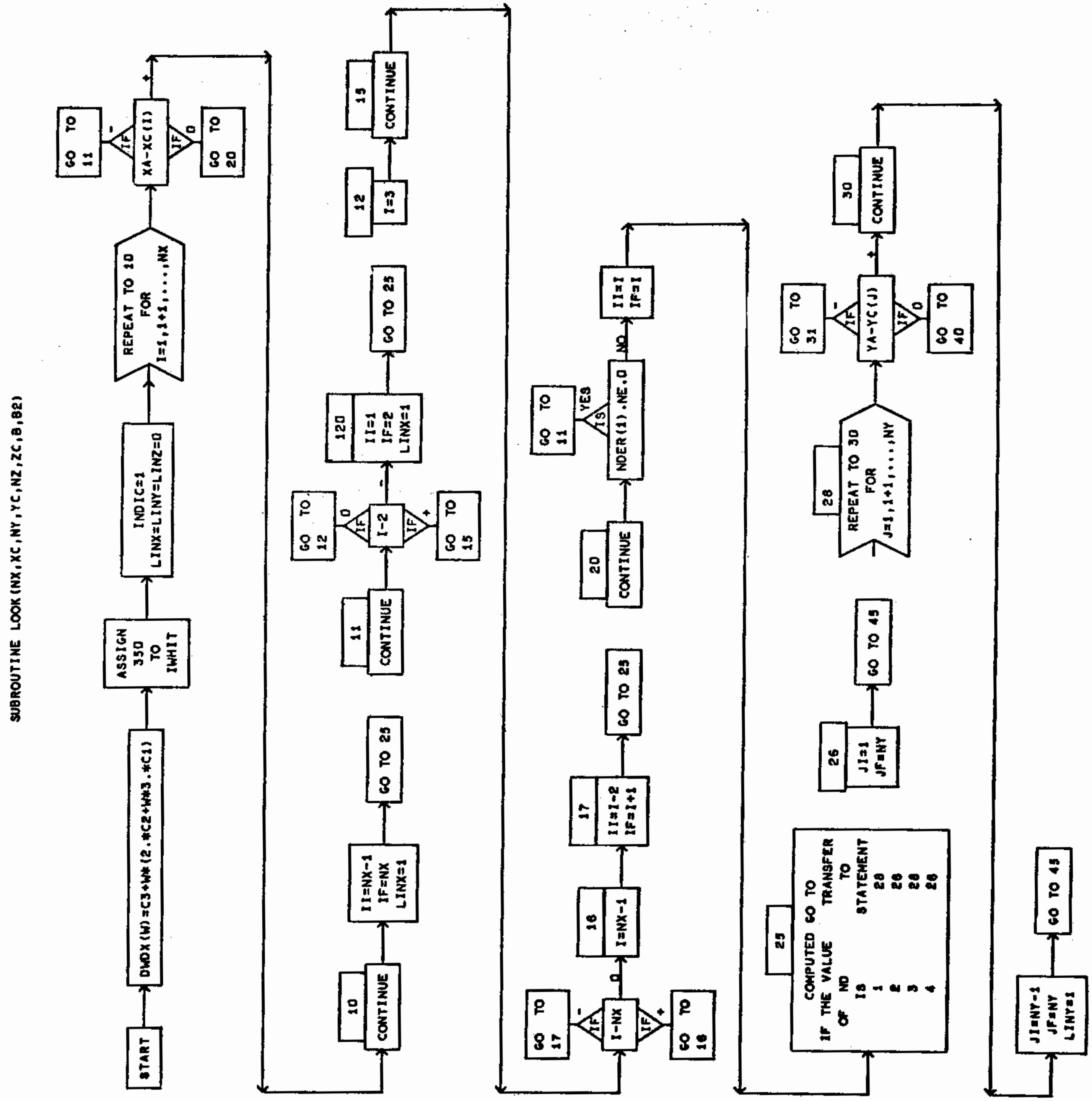


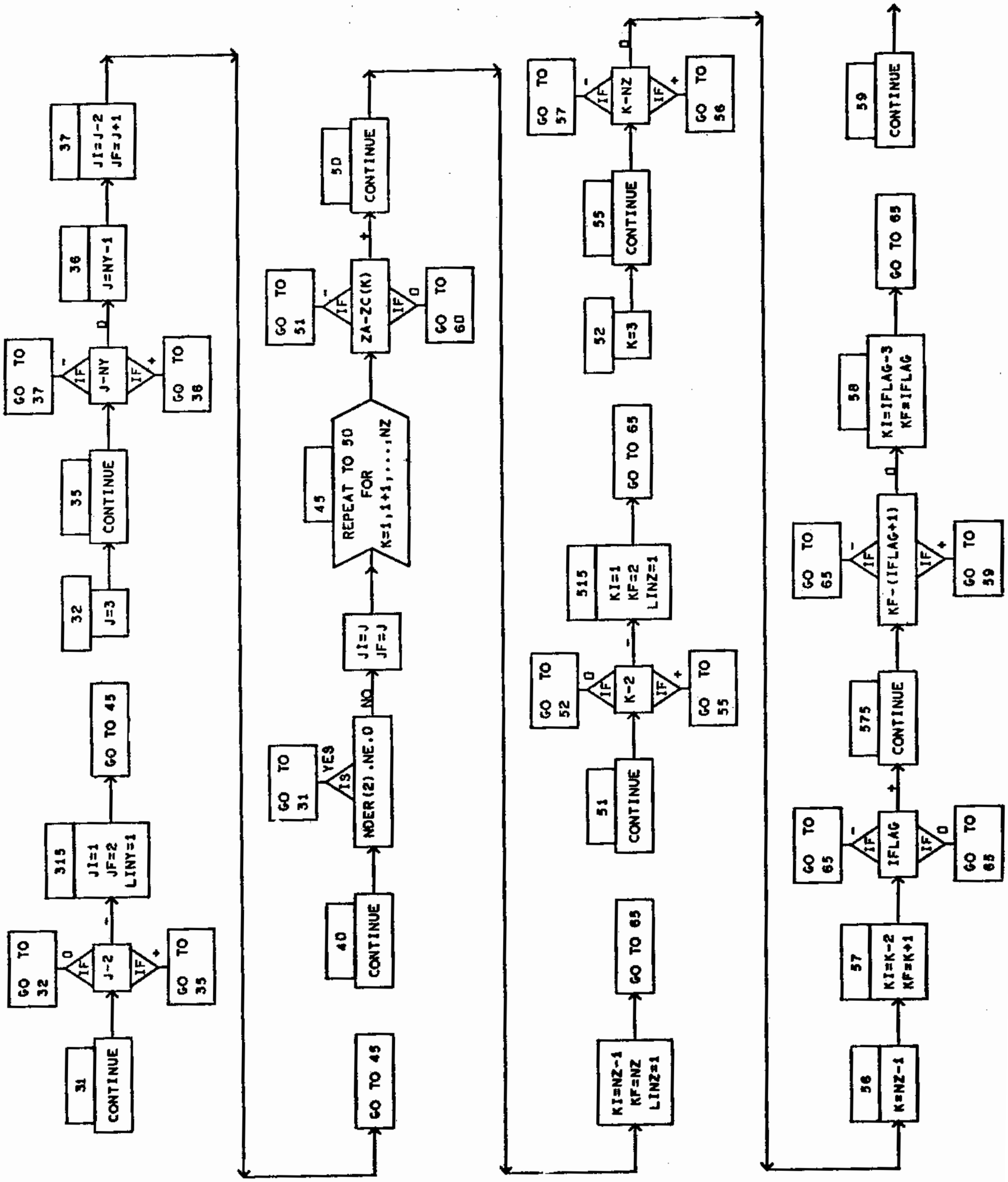
IS ALFAR.LE.ALMXR
YES
GO TO 110
NO
105
IAPV=IAPV+1
APV1=APV+FLOAT(IAPV)/100.
VAPF=VSF*APV1
PRINT WRITE
WRITE(PRINT,1003)APV1
LINE=LINE+2
GO TO 70
110
CONTINUE
IS VTF*SIN(GAMR).GE.-(RS*1.01)
YES
GO TO 115
NO
GAMR=ASIN(-RS/VTF)
IGAM=IGAM+1
IS IGAM.GT.20
YES
GO TO 405
NO
GAMD=GAMR*RTOD
PRINT WRITE
WRITE(PRINT,1004)GAMD
LINE=LINE+2
GO TO 100
115
CONTINUE
IS THETR.GT.GALMXR-WINCR
YES
GO TO 105
NO
IS THETR.GE.-0.001
YES
GO TO 120
NO
GAMR=WINCR-ALFAR
IGAM=IGAM+1
IS IGAM.GT.20
YES
GO TO 405
NO
IS GAMR.LT.0.0.AND.VTF*SIN(GAMR).GE.-RS
YES
GO TO 100
NO
GO TO 400
120
ISEG=1
GAMTR=GAMR
GO TO 20
125
THRTL=1.0
INDEX=3
ENGNO=ENGNO-1.0
NENG=2
TRANSFER TO SUBROUTINE AEQFM
ENGNO=ENGNO+1.0
NENG=0
THETR=ALFAR-WINCR+GAMR
ISEG=2
GO TO 20
130
HF=HFRUN+C4HT+HFOB
ZE=HFOB
GAMR=GAMTR
TRANSFER TO SUBROUTINE ATMOS
RHOS=RHOZ2*SIG*S
QS=RHOS*VTF*VTF
MACH=VTF/SOUND
GN=1.0
GT=0.0
INDEX=6
ITRM=1
COEF=0.0


TRANSFER TO SUBROUTINE AEQFM
THETR=ALFAR+GAMR-WINCR
IS ALFAR.GT.ALMXR
YES
GO TO 105
NO
ISEG=3 VOBS=VTF
GO TO 20
175
THRTL=1.0 INDEX=3 ENGNO=ENGNO-1.0 NENG=2
TRANSFER TO SUBROUTINE AEQFM
ENGNO=ENGNO+1.0 NENG=0 THETR=ALFAR-WINCR+GAMR ISEG=4
GO TO 20
200
GAMR=GAMTR
IS GAMR.GE.0.0
YES
GO TO 400
NO
IS THETR.LT.-0.001
YES
GO TO 405
NO
TIME1=HFOB/(VAPF*SIN(ABS(GAMR))) DIST1=TIME1*VAPF*COS(GAMR) TIMS=TIME1 XE=DIST1
ZE=0.0 HF=HFRUN+C4HT
TRANSFER TO SUBROUTINE ATMOS
RHOS=RHOZ2*SIG*S QS=RHOS*VTF*VTF ITRM=1 GN=1.0
GT=0.0 INDEX=6 COEF=0.0
TRANSFER TO SUBROUTINE AEQFM
THETR=ALFAR-WINCR+GAMR ISEG=5
GO TO 20
250
TIMS=0.0 THRTL=0.00 XE=0.0 ZE=0.0
XEDOT=VAPF*COS(GAMR) ZEDOT=0.0 GAMR=0.0 ALFAR=THETR+WINCR
INDEX=1 INT=2 Q=-ROTATN*DTOR COEF=RCOEF
DTIME=0.2 ILOOP=1 VTF=XEDOT
255
VTF1=VTF
TRANSFER TO SUBROUTINE GINTG
ISEG=6
GO TO 20
260
CONTINUE
IF ALFAR-WINCR
0
GO TO 270
+
GO TO 255
-

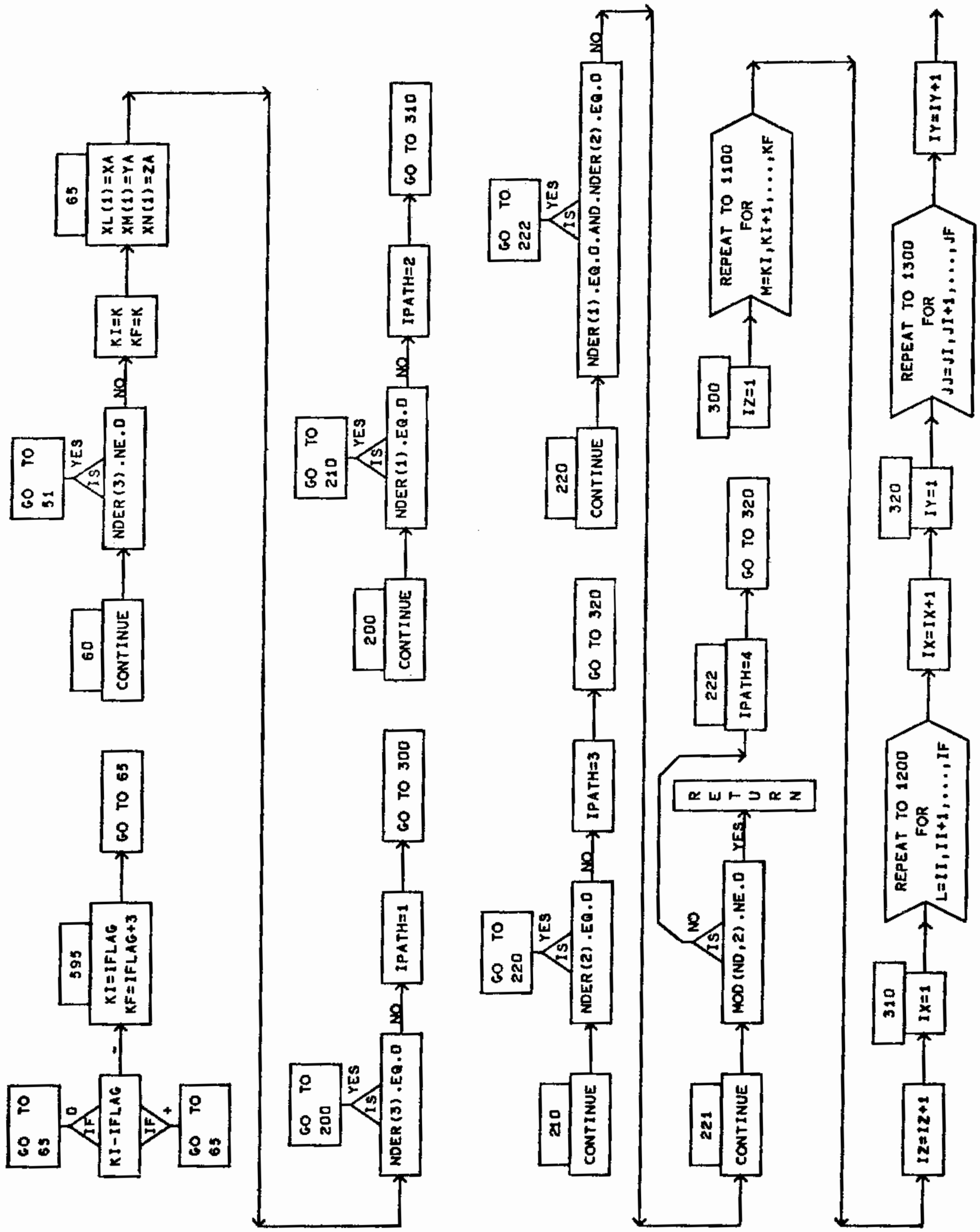

265
DT=(WINCR-ALFAR)/Q
ALFAR=WINCR
VNDF=VTF-(VTF-VTF1)/DTIME*DT
XE=XE+(0.5*(VTF+VNDF)-VWF)*DT
VTF=VNDF
TIMS=TIMS+DT
270
TIME2=TIMS
DIST2=XE
ISEG=7
INDEX=1
QS=RHOS*VTF*VTF
MACH=VTF/SOUND
TRANSFER TO SUBROUTINE
AEQFN
THETR=0.0
GO TO 20
300
INT=1
DTIME=0.2
TIMS=0.0
XE=0.0
ZE=0.0
THETR=0.0
ZEDOT=0.0
Q=0.0
XEDOT=VNDF
ILOOP=1
305
VTF1=VTF
IS
NO
TIMS+TIME2.GE.TSP
YES
ISP=1
IS
NO
TIMS+TIME2.GE.TBRK
YES
COEF=BCOEF
IS
NO
TIMS+TIME2.GE.TREV
YES
IREV=1
GO TO
315
IS
YES
ISP.EQ.1.AND.JSP.EQ.1
NO
GO TO
315
IS
YES
ISP.EQ.0
NO
JSP=1
PRINT
WRITE
WRITE(PRINT,1005)
LINE=LINE+2
315
CONTINUE
GO TO
320
IS
YES
COEF.EQ.BCOEF.AND.JBRK.EQ.1
NO
GO TO
320
IS
YES
COEF.NE.BCOEF
NO
JBRK=1
PRINT
WRITE
WRITE(PRINT,1006)
LINE=LINE+2
320
CONTINUE
GO TO
325
IS
YES
IREV.EQ.1.AND.JREV.EQ.1
NO
GO TO
325
IS
YES
IREV.EQ.0
NO
JREV=1
NENG=1
PRINT
WRITE

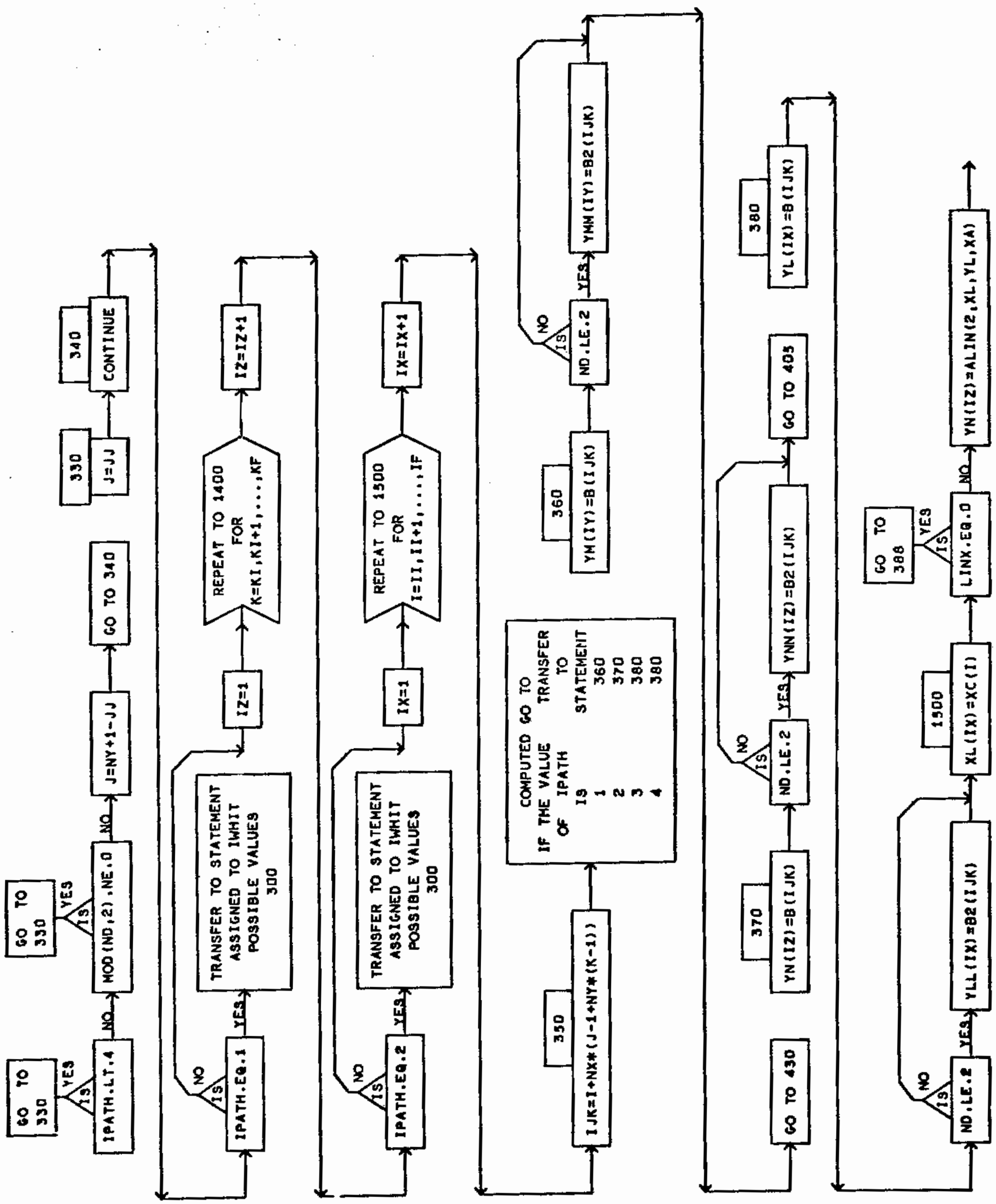

WRITE (PRINT,1007)
LINE=LINE+2
325
TRANSFER TO SUBROUTINE
GINTG
ISEG=8
GO TO 20
330
CONTINUE
XEDOT-VWF
IF
0
GO TO
340
IF
+
GO TO
305
335
DT=(VWF-XEDOT)/(VTF1-XEDOT)*DTIME
XE=XE+(0.5*(VWF+XEDOT)-VWF)*DT
TIMS=TIMS-DT
340
VTF=VWF
ISEG=9
INDEX=1
QS=RHOS*VTF*VTF
MACH=VTF/SOUND
TRANSFER TO SUBROUTINE
AEQFN
GO TO 20
350
GAMT=GAMTR*RTOD
GDIST=DIST2+XE
GTIME=TIME2+TIMS
TDIST=GDIST+DIST1
TTIME=GTIME+TIME1
VAPK=VAPF*FTOK
PRINT
WRITE
WRITE (PRINT,1050) GDIST,GTIME,GAMT,DIST1,TIME1,VAPK,TDIST,TTIME,VSK
RETURN
400
PRINT
WRITE
400
WRITE (PRINT,1051)
RETURN
405
PRINT
WRITE
405
WRITE (PRINT,1052)
ALFAD=ALFAR*RTOD
GAMD=GAMR*RTOD
THETD=THETR*RTOD
PRINT
WRITE
WRITE (PRINT,1053) ALFAD,GAMD,THETD
RETURN

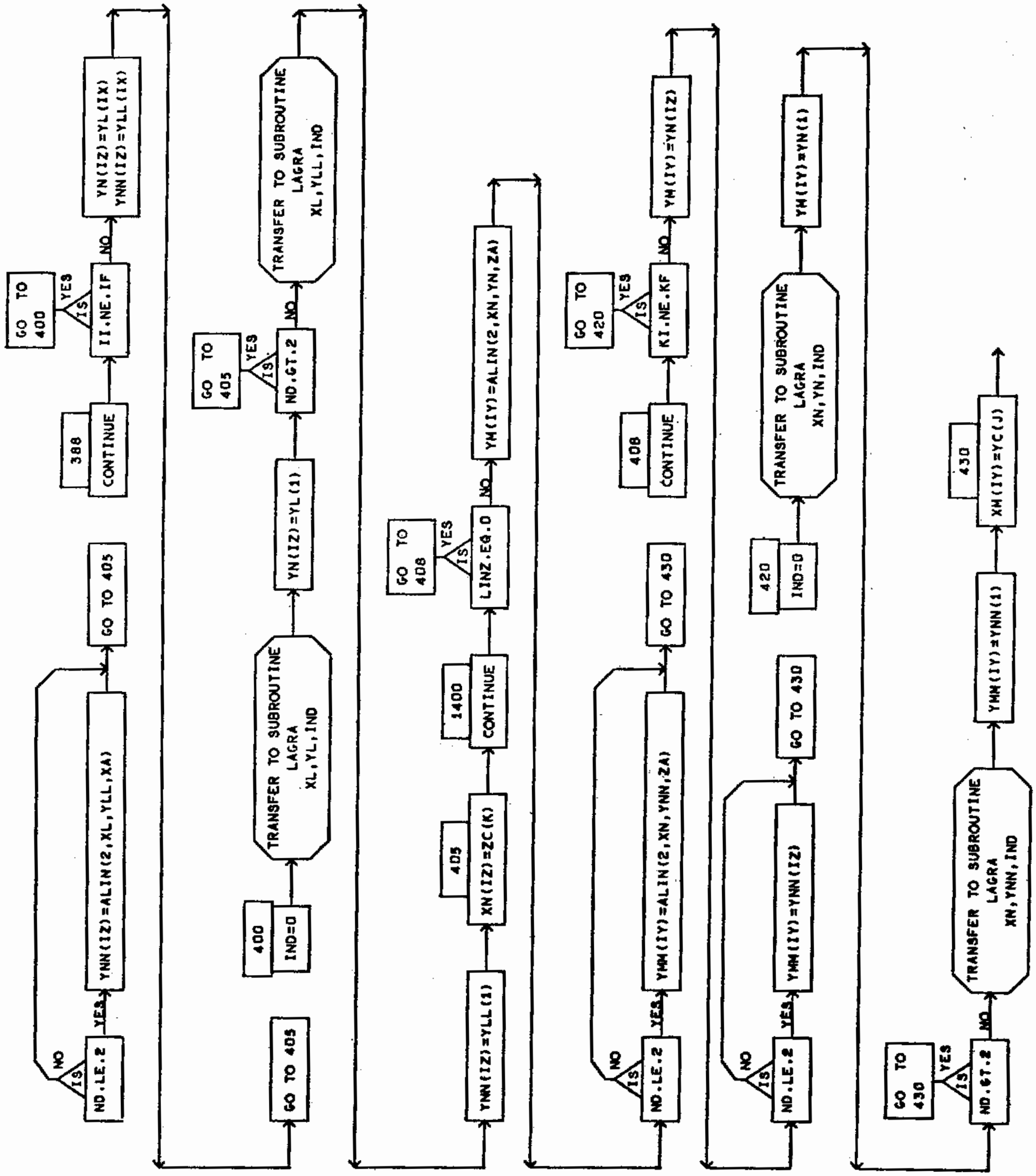

SUBROUTINE LOOK(NX,XC,NY,YC,NZ,ZC,B,B2)
START
DWDX(W)=C3+W*(2.*C2+W*3.*C1)
ASSIGN 350 TO IWHIT
INDIC=1
LINX=LINY=LINZ=0
REPEAT TO 10 FOR I=1,1+1,...,NX
IF XA-XC(I)
GO TO 11
GO TO 20
10 CONTINUE
II=NX-1
IF=NX
LINX=1
GO TO 25
11 CONTINUE
IF I-2
GO TO 12
GO TO 15
120 II=1
IF=2
LINX=1
GO TO 25
12 I=3
15 CONTINUE
IF I-NX
GO TO 17
GO TO 16
16 I=NX-1
17 II=I-2
IF=I+1
GO TO 25
20 CONTINUE
IS NDER(1).NE.0
YES
NO
GO TO 11
II=I
IF=I
25 COMPUTED GO TO
IF THE VALUE OF ND IS
1
2
3
4
TRANSFER TO STATEMENT
28
26
28
26
26 JI=1
JF=NY
GO TO 45
28 REPEAT TO 30 FOR J=1,1+1,...,NY
IF YA-YC(J)
GO TO 31
GO TO 40
30 CONTINUE
JI=NY-1
JF=NY
LINY=1
GO TO 45



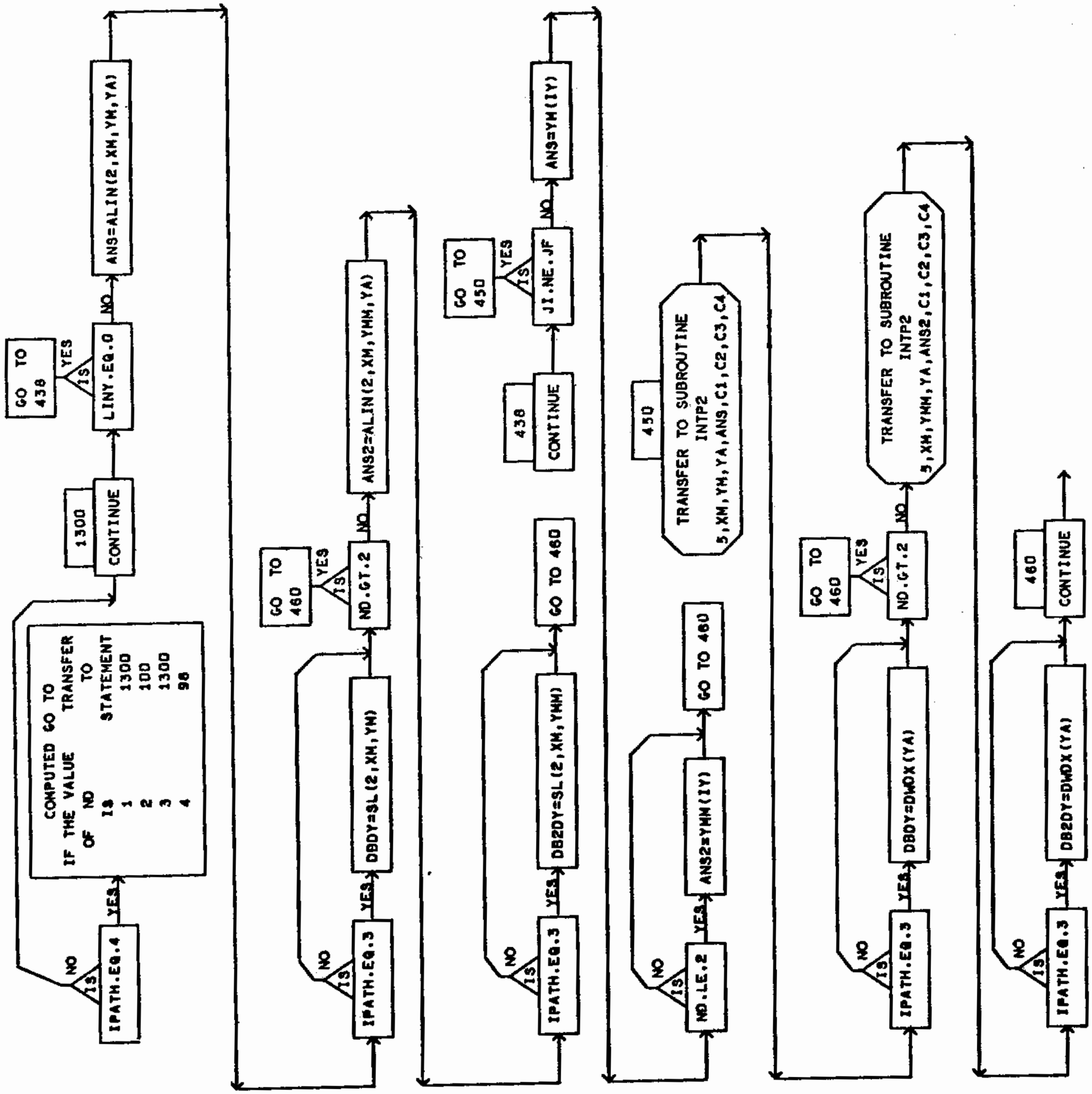
31 CONTINUE
IF J-2
GO TO 32
GO TO 35
315 JI=1 JF=2 LINY=1
GO TO 45
32 J=3
35 CONTINUE
IF J-NY
GO TO 37
GO TO 36
36 J=NY-1
37 JI=J-2 JF=J+1
GO TO 45
40 CONTINUE
IS NDER(2).NE.0
YES
NO
GO TO 31
JI=J JF=J
45 REPEAT TO 50 FOR K=1,1+1,...,NZ
IF ZA-ZC(K)
GO TO 51
GO TO 60
50 CONTINUE
KI=NZ-1 KF=NZ LINZ=1
GO TO 65
51 CONTINUE
IF K-2
GO TO 52
GO TO 55
515 KI=1 KF=2 LINZ=1
GO TO 65
52 K=3
55 CONTINUE
IF K-NZ
GO TO 57
GO TO 56
56 K=NZ-1
57 KI=K-2 KF=K+1
IF IFLAG
GO TO 65
575 CONTINUE
IF KF-(IFLAG+1)
GO TO 59
58 KI=IFLAG-3 KF=IFLAG
GO TO 65
59 CONTINUE

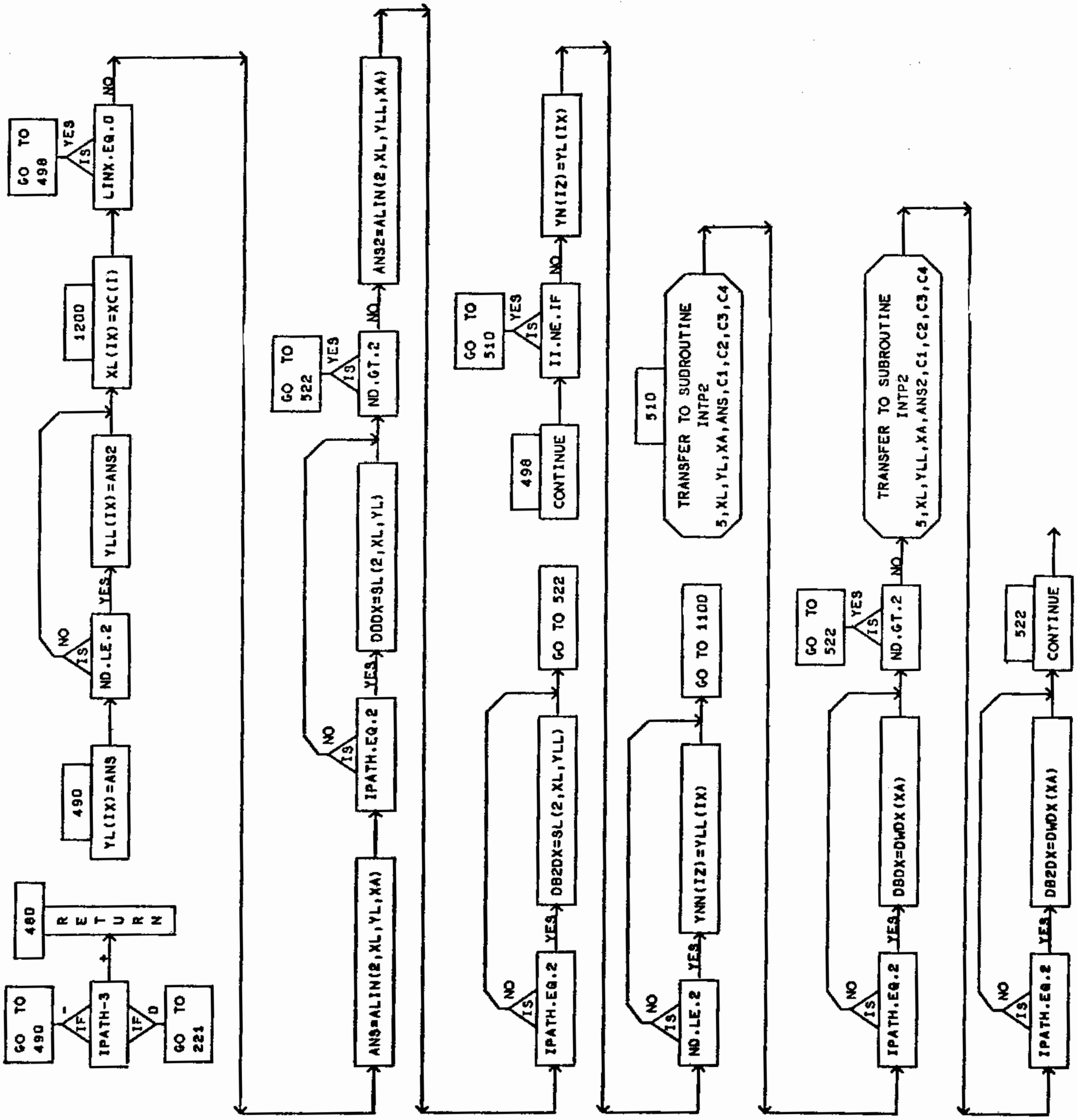

GO TO 65
IF KI-IFLAG
0
-
+
GO TO 65
595
KI=IFLAG
KF=IFLAG+3
GO TO 65
60
CONTINUE
IS NDER(3).NE.0
YES
GO TO 51
NO
KI=K
KF=K
65
XL(1)=XA
XM(1)=YA
XN(1)=ZA
IS NDER(3).EQ.0
YES
GO TO 200
NO
IPATH=1
GO TO 300
200
CONTINUE
IS NDER(1).EQ.0
YES
GO TO 210
NO
IPATH=2
GO TO 310
210
CONTINUE
IS NDER(2).EQ.0
YES
GO TO 220
NO
IPATH=3
GO TO 320
220
CONTINUE
IS NDER(1).EQ.0.AND.NDER(2).EQ.0
YES
GO TO 222
NO
221
CONTINUE
IS MOD(ND,2).NE.0
YES
RETURN
NO
222
IPATH=4
GO TO 320
300
IZ=1
REPEAT TO 1100 FOR M=KI,KI+1,...,KF
IZ=IZ+1
310
IX=1
REPEAT TO 1200 FOR L=II,II+1,...,IF
IX=IX+1
320
IY=1
REPEAT TO 1300 FOR JJ=JI,JI+1,...,JF
IY=IY+1

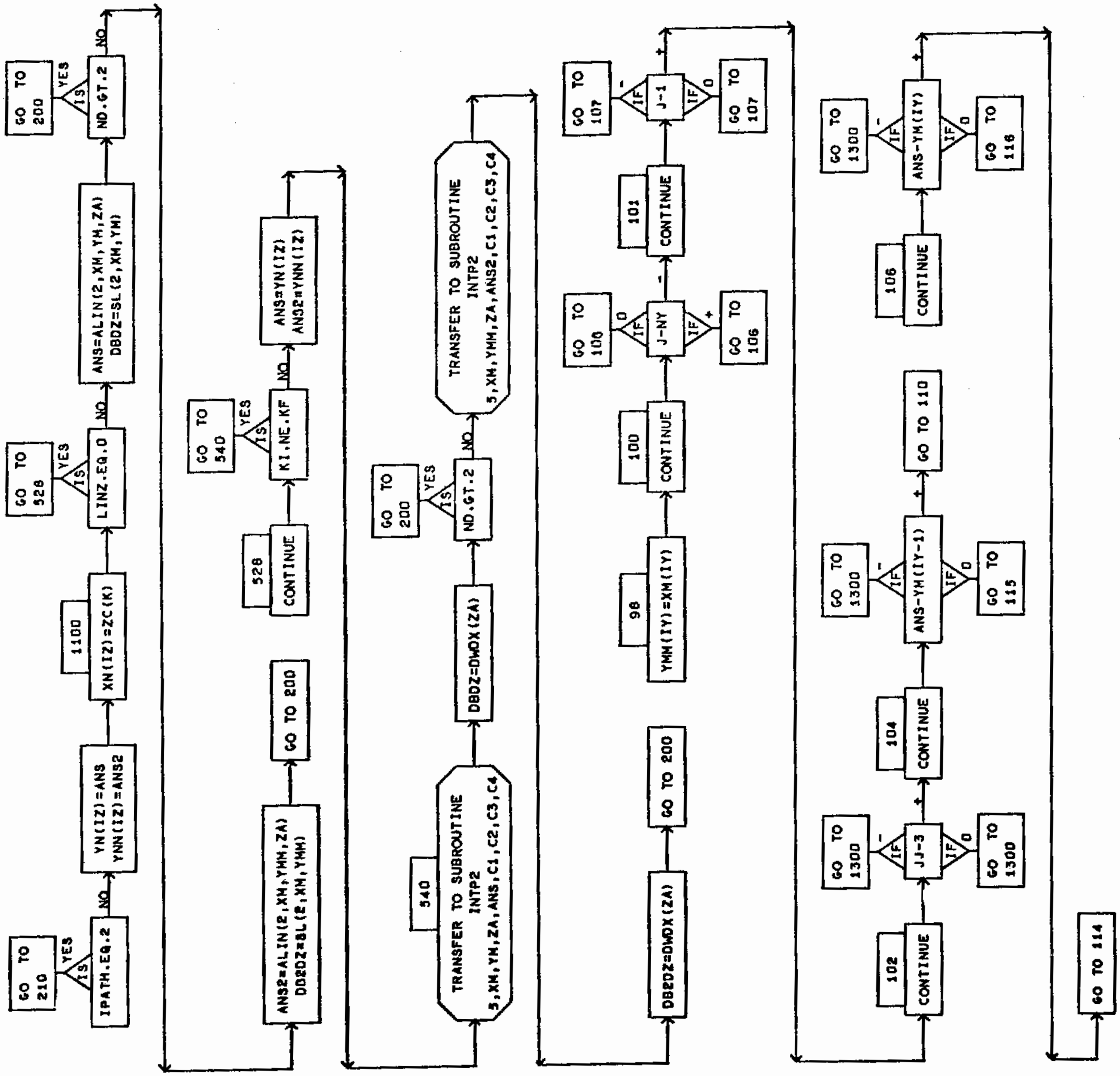

IPATH.LT.4
IS
YES
GO TO 330
NO
MOD(ND,2).NE.0
IS
YES
GO TO 330
NO
J=NY+1-JJ
GO TO 340
330
J=JJ
340
CONTINUE
IPATH.EQ.1
IS
NO
YES
TRANSFER TO STATEMENT ASSIGNED TO IWHIT POSSIBLE VALUES 300
IZ=1
REPEAT TO 1400 FOR K=KI,KI+1,...,KF
IZ=IZ+1
IPATH.EQ.2
IS
NO
YES
TRANSFER TO STATEMENT ASSIGNED TO IWHIT POSSIBLE VALUES 300
IX=1
REPEAT TO 1500 FOR I=II,II+1,...,IF
IX=IX+1
350
IJK=I+NX*(J-1+NY*(K-1))
COMPUTED GO TO
IF THE VALUE OF IPATH IS
TRANSFER TO STATEMENT
1 360
2 370
3 380
4 380
360
YM(IY)=B(IJK)
ND.LE.2
IS
NO
YES
YMM(IY)=B2(IJK)
GO TO 430
370
YN(IZ)=B(IJK)
ND.LE.2
IS
NO
YES
YNN(IZ)=B2(IJK)
GO TO 405
380
YL(IX)=B(IJK)
ND.LE.2
IS
NO
YES
YLL(IX)=B2(IJK)
1500
XL(IX)=XC(I)
LINX.EQ.0
IS
YES
GO TO 388
NO
YN(IZ)=ALIN(2,XL,YL,XA)



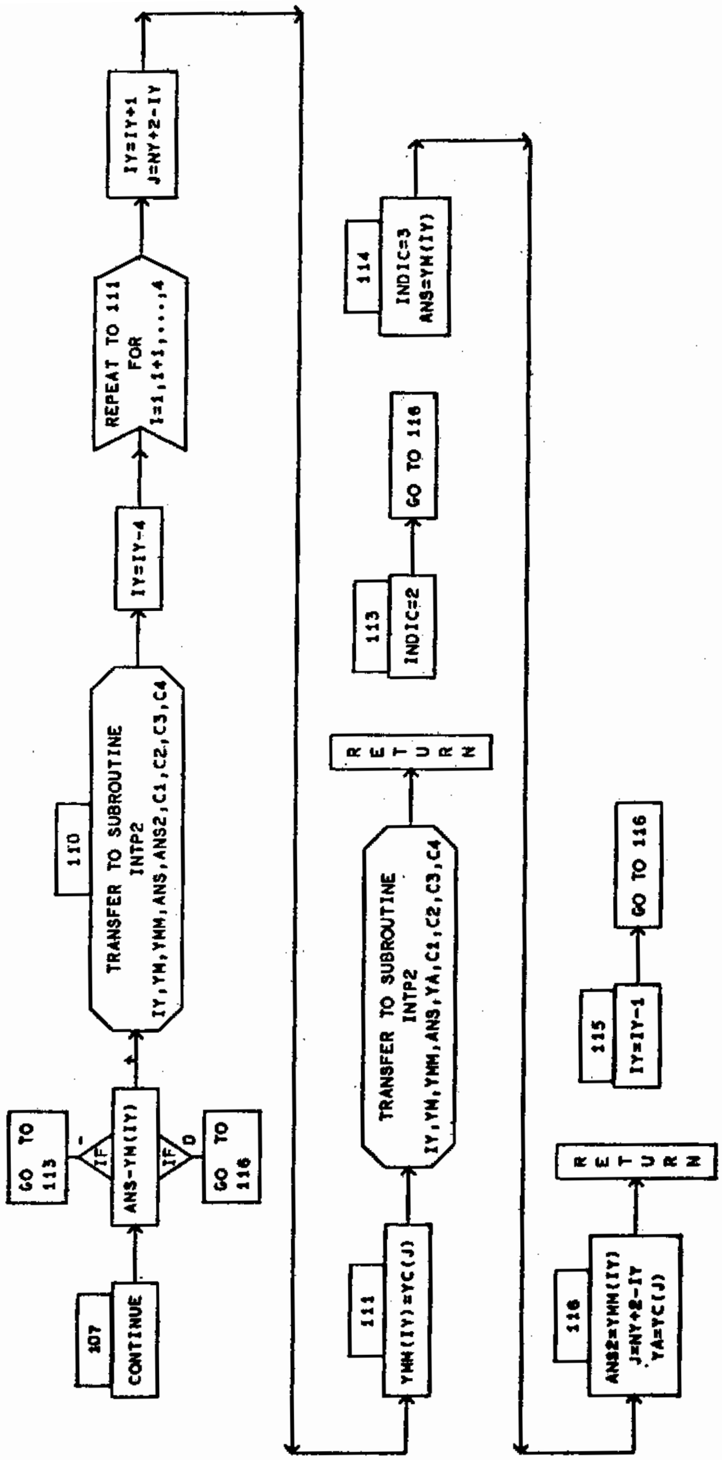
ND.LE.2
IS
NO
YES
YNN(IZ)=ALIN(2,XL,YLL,XA)
GO TO 405
388
CONTINUE
II.NE.IF
IS
YES
GO TO 400
NO
YN(IZ)=YL(IX)
YNN(IZ)=YLL(IX)
GO TO 405
400
IND=0
TRANSFER TO SUBROUTINE
LAGRA
XL,YL,IND
YN(IZ)=YL(1)
ND.GT.2
IS
YES
GO TO 405
NO
TRANSFER TO SUBROUTINE
LAGRA
XL,YLL,IND
YNN(IZ)=YLL(1)
405
XN(IZ)=ZC(K)
1400
CONTINUE
LINZ.EQ.0
IS
YES
GO TO 408
NO
YM(IY)=ALIN(2,XN,YN,ZA)
ND.LE.2
IS
NO
YES
YMM(IY)=ALIN(2,XN,YNN,ZA)
GO TO 430
408
CONTINUE
KI.NE.KF
IS
YES
GO TO 420
NO
YM(IY)=YN(IZ)
ND.LE.2
IS
NO
YES
YMM(IY)=YNN(IZ)
GO TO 430
420
IND=0
TRANSFER TO SUBROUTINE
LAGRA
XN,YN,IND
YM(IY)=YN(1)
ND.GT.2
IS
YES
GO TO 430
NO
TRANSFER TO SUBROUTINE
LAGRA
XN,YNN,IND
YMM(IY)=YNN(1)
430
XM(IY)=YC(J)


IPATH.EQ.4
IS
YES
NO
COMPUTED GO TO
IF THE VALUE OF ND IS
1
2
3
4
TRANSFER TO STATEMENT
1300
100
1300
98
1300
CONTINUE
LINY.EQ.0
GO TO 438
ANS=ALIN(2,XM,YM,YA)
IPATH.EQ.3
DBDY=SL(2,XM,YM)
ND.GT.2
GO TO 460
ANS2=ALIN(2,XM,YMM,YA)
IPATH.EQ.3
DB2DY=SL(2,XM,YMM)
GO TO 460
438
CONTINUE
JI.NE.JF
GO TO 450
ANS=YM(IY)
ND.LE.2
ANS2=YMM(IY)
GO TO 460
450
TRANSFER TO SUBROUTINE INTP2
5,XM,YM,YA,ANS,C1,C2,C3,C4
IPATH.EQ.3
DBDY=DWDX(YA)
ND.GT.2
GO TO 460
TRANSFER TO SUBROUTINE INTP2
5,XM,YMM,YA,ANS2,C1,C2,C3,C4
IPATH.EQ.3
DB2DY=DWDX(YA)
460
CONTINUE

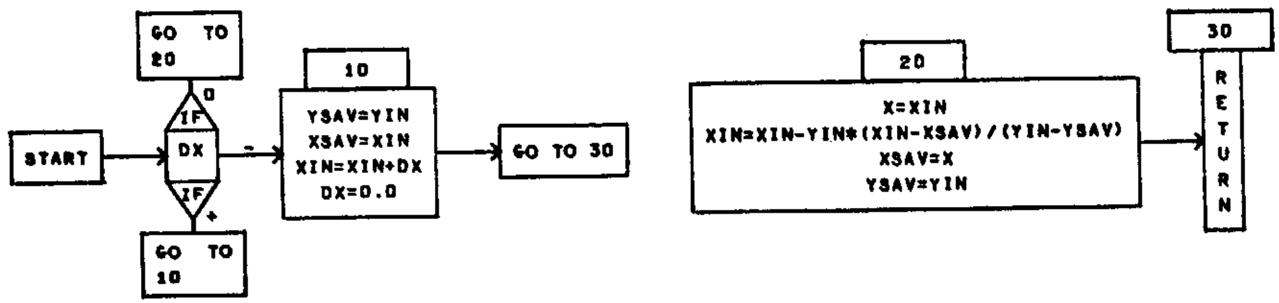

GO TO 490
IF
-
IPATH-3
+
IF
0
GO TO 221
480
RETURN
490
YL(IX)=ANS
IS
ND.LE.2
NO
YES
YLL(IX)=ANS2
1200
XL(IX)=XC(I)
GO TO 498
YES
IS
LINX.EQ.0
NO
ANS=ALIN(2,XL,YL,XA)
IS
IPATH.EQ.2
NO
YES
DDDX=SL(2,XL,YL)
GO TO 522
YES
IS
ND.GT.2
NO
ANS2=ALIN(2,XL,YLL,XA)
IS
IPATH.EQ.2
NO
YES
DB2DX=SL(2,XL,YLL)
GO TO 522
498
CONTINUE
GO TO 510
YES
IS
II.NE.IF
NO
YN(IZ)=YL(IX)
IS
ND.LE.2
NO
YES
YNN(IZ)=YLL(IX)
GO TO 1100
510
TRANSFER TO SUBROUTINE
INTP2
5,XL,YL,XA,ANS,C1,C2,C3,C4
IS
IPATH.EQ.2
NO
YES
DBDX=DWDX(XA)
GO TO 522
YES
IS
ND.GT.2
NO
TRANSFER TO SUBROUTINE
INTP2
5,XL,YLL,XA,ANS2,C1,C2,C3,C4
IS
IPATH.EQ.2
NO
YES
DB2DX=DWDX(XA)
522
CONTINUE



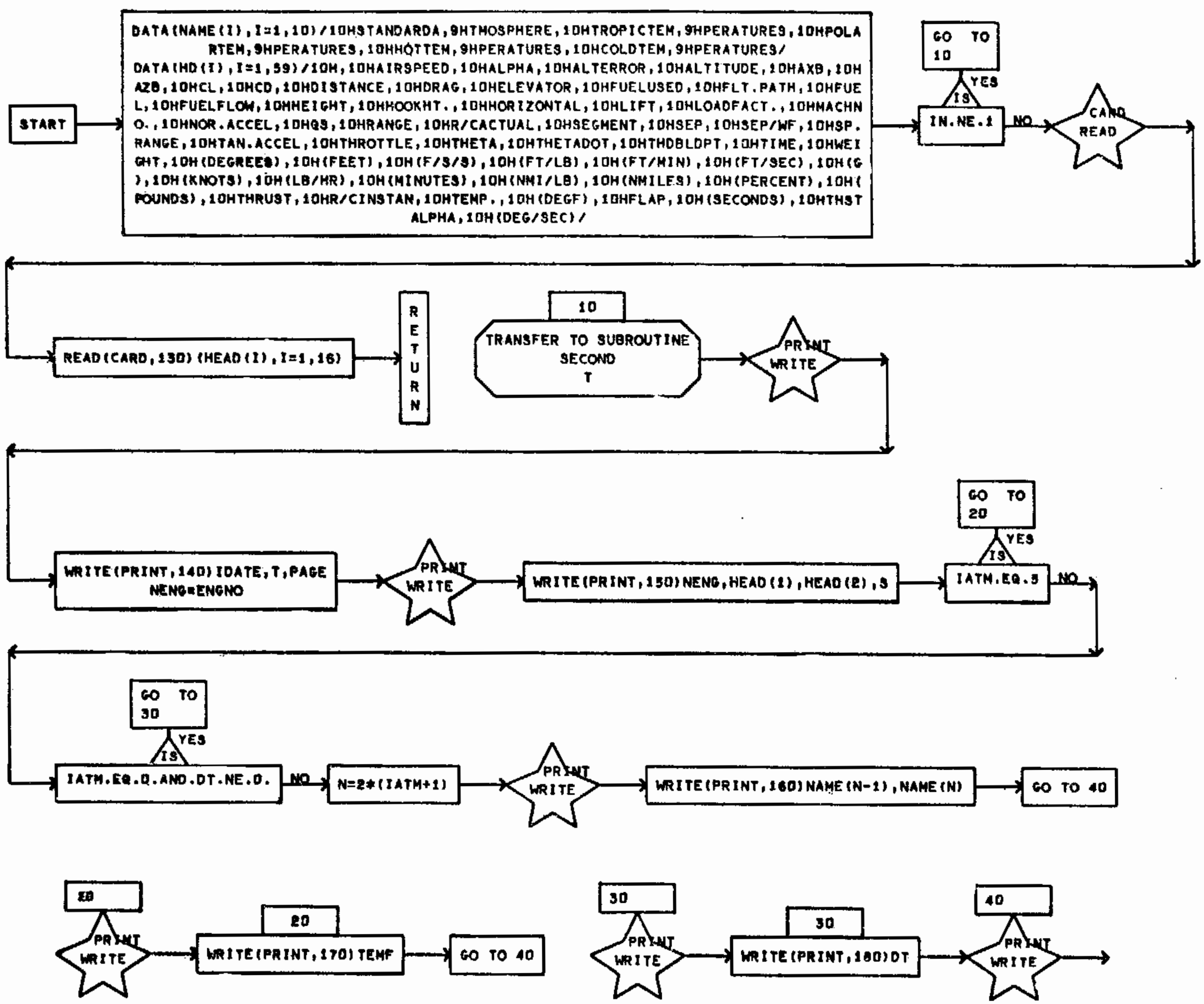
IPATH.EQ.2
GO TO 210
YN(IZ)=ANS
YNN(IZ)=ANS2
1100
XN(IZ)=ZC(K)
LINZ.EQ.0
GO TO 528
ANS=ALIN(2,XM,YM,ZA)
DBDZ=SL(2,XM,YM)
ND.GT.2
GO TO 200
ANS2=ALIN(2,XM,YMM,ZA)
DB2DZ=SL(2,XM,YMM)
GO TO 200
528
CONTINUE
KI.NE.KF
GO TO 540
ANS=YN(IZ)
ANS2=YNN(IZ)
540
TRANSFER TO SUBROUTINE INTP2
5,XM,YM,ZA,ANS,C1,C2,C3,C4
DBDZ=DWDX(ZA)
ND.GT.2
GO TO 200
TRANSFER TO SUBROUTINE INTP2
5,XM,YMM,ZA,ANS2,C1,C2,C3,C4
DB2DZ=DWDX(ZA)
GO TO 200
98
YMM(IY)=XM(IY)
100
CONTINUE
J-NY
GO TO 100
GO TO 106
101
CONTINUE
J-1
GO TO 107
GO TO 107
102
CONTINUE
JJ-3
GO TO 1300
GO TO 1300
104
CONTINUE
ANS-YM(IY-1)
GO TO 1300
GO TO 115
GO TO 110
106
CONTINUE
ANS-YM(IY)
GO TO 1300
GO TO 116
GO TO 114

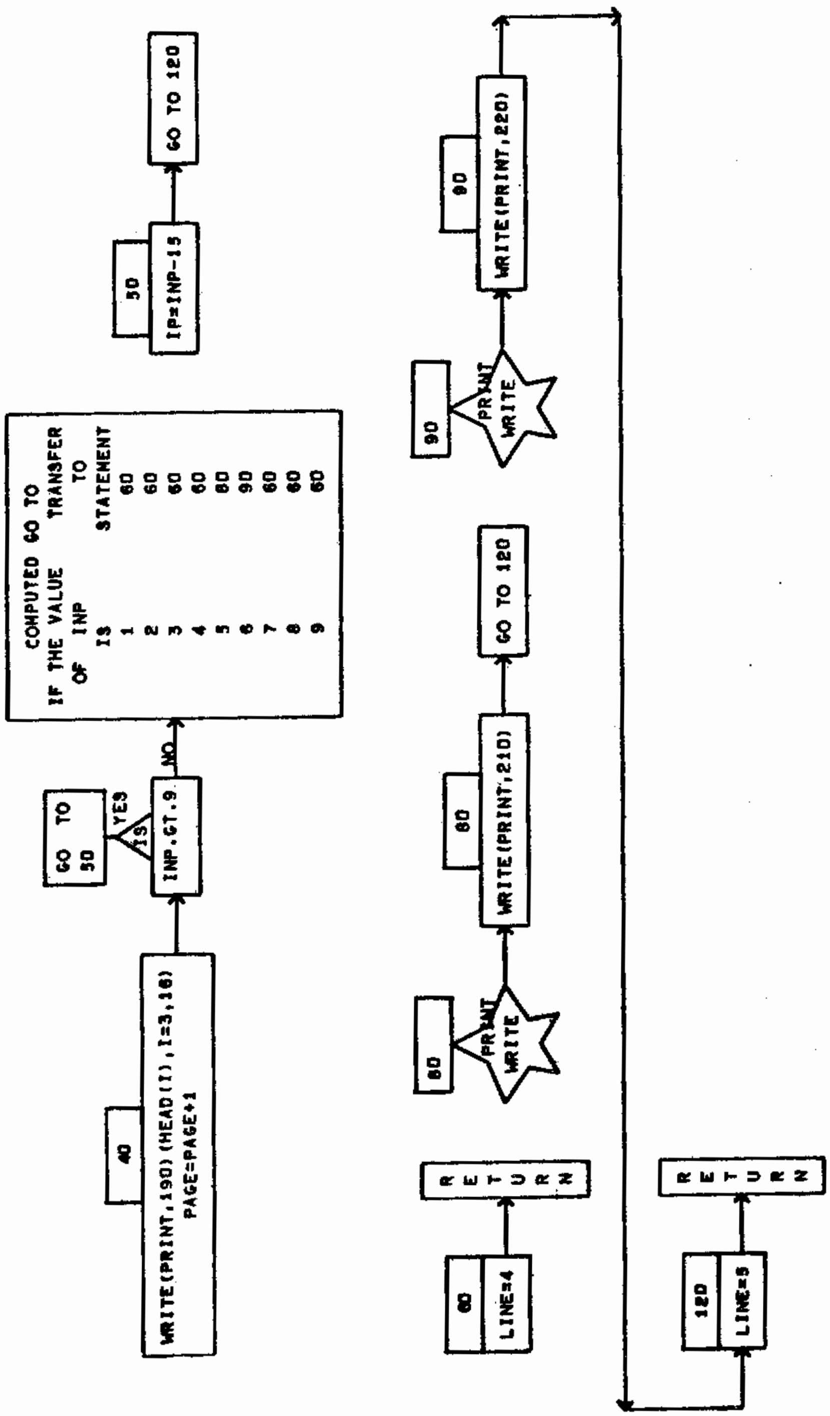

107
CONTINUE
ANS-YM(IY)
IF
-
GO TO 113
IF
0
GO TO 116
+
110
TRANSFER TO SUBROUTINE INTP2
IY,YM,YMM,ANS,ANS2,C1,C2,C3,C4
IY=IY-4
REPEAT TO 111 FOR I=1,I+1,...,4
IY=IY+1
J=NY+2-IY
111
YMM(IY)=YC(J)
TRANSFER TO SUBROUTINE INTP2
IY,YM,YMM,ANS,YA,C1,C2,C3,C4
RETURN
113
INDIC=2
GO TO 116
114
INDIC=3
ANS=YM(IY)
116
ANS2=YMM(IY)
J=NY+2-IY
YA=YC(J)
RETURN
115
IY=IY-1
GO TO 116

SUBROUTINE NWRP2(DX,YIN,XIN,YSAV,XSAV)

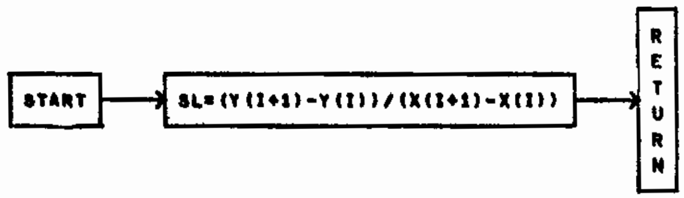


SUBROUTINE SKIP

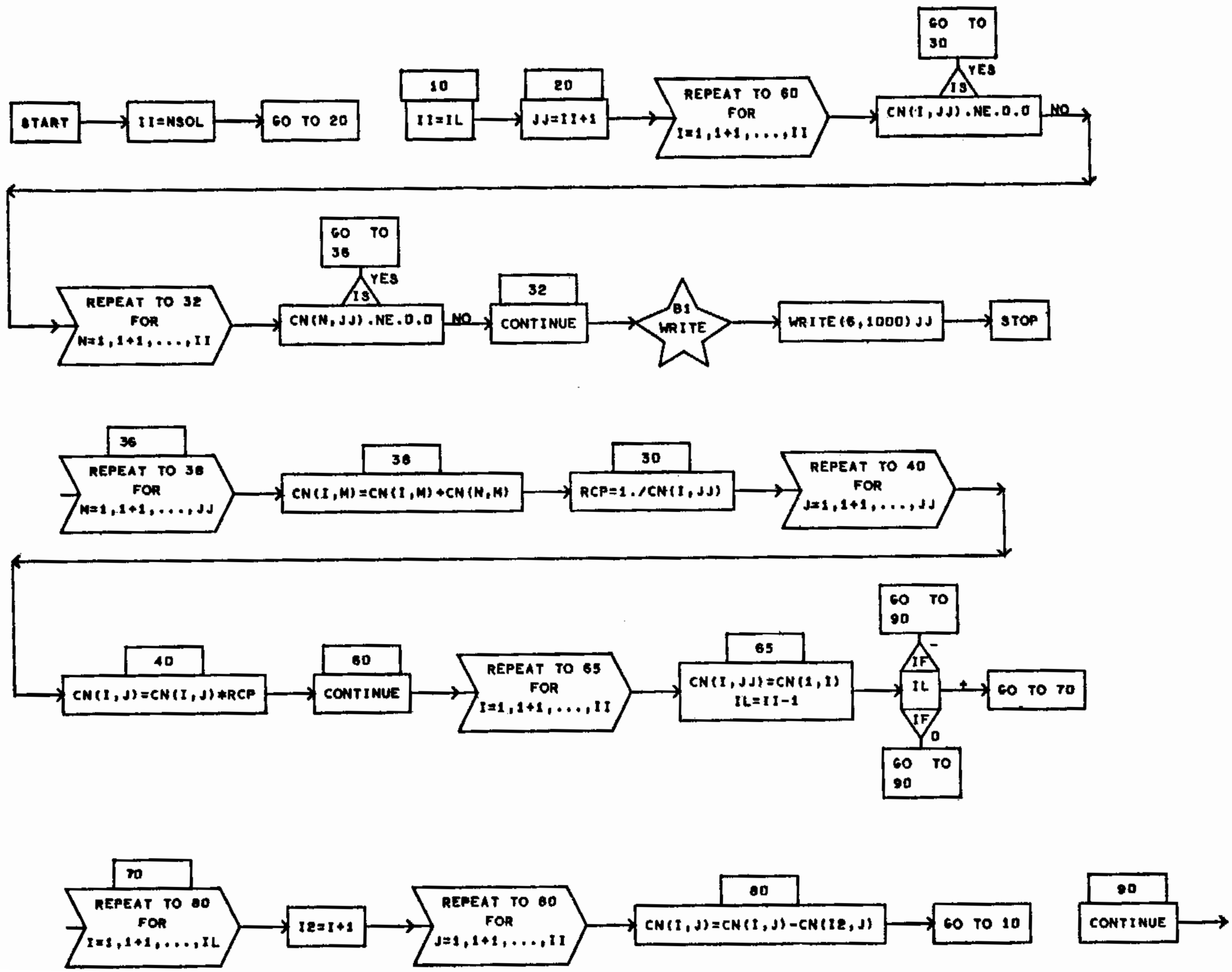


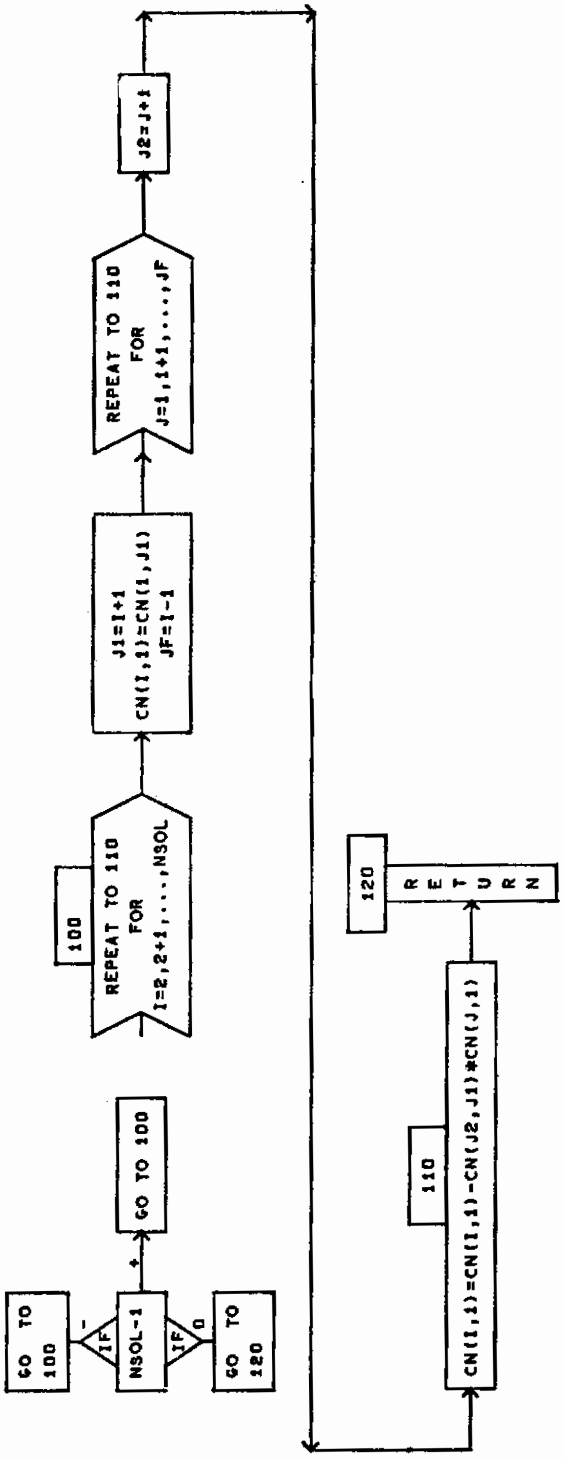

40
WRITE(PRINT,190)(HEAD(I),I=3,18)
PAGE=PAGE+1
IS
INP.GT.9
YES
GO TO 50
NO
COMPUTED GO TO
IF THE VALUE OF INP IS
TRANSFER TO STATEMENT
1 60
2 60
3 60
4 60
5 80
6 90
7 60
8 60
9 60
50
IP=INP-15
GO TO 120
80
PRINT
WRITE
80
WRITE(PRINT,210)
GO TO 120
90
PRINT
WRITE
90
WRITE(PRINT,220)
60
LINE=4
RETURN
120
LINE=5
RETURN

FUNCTION SL(I,X,Y)

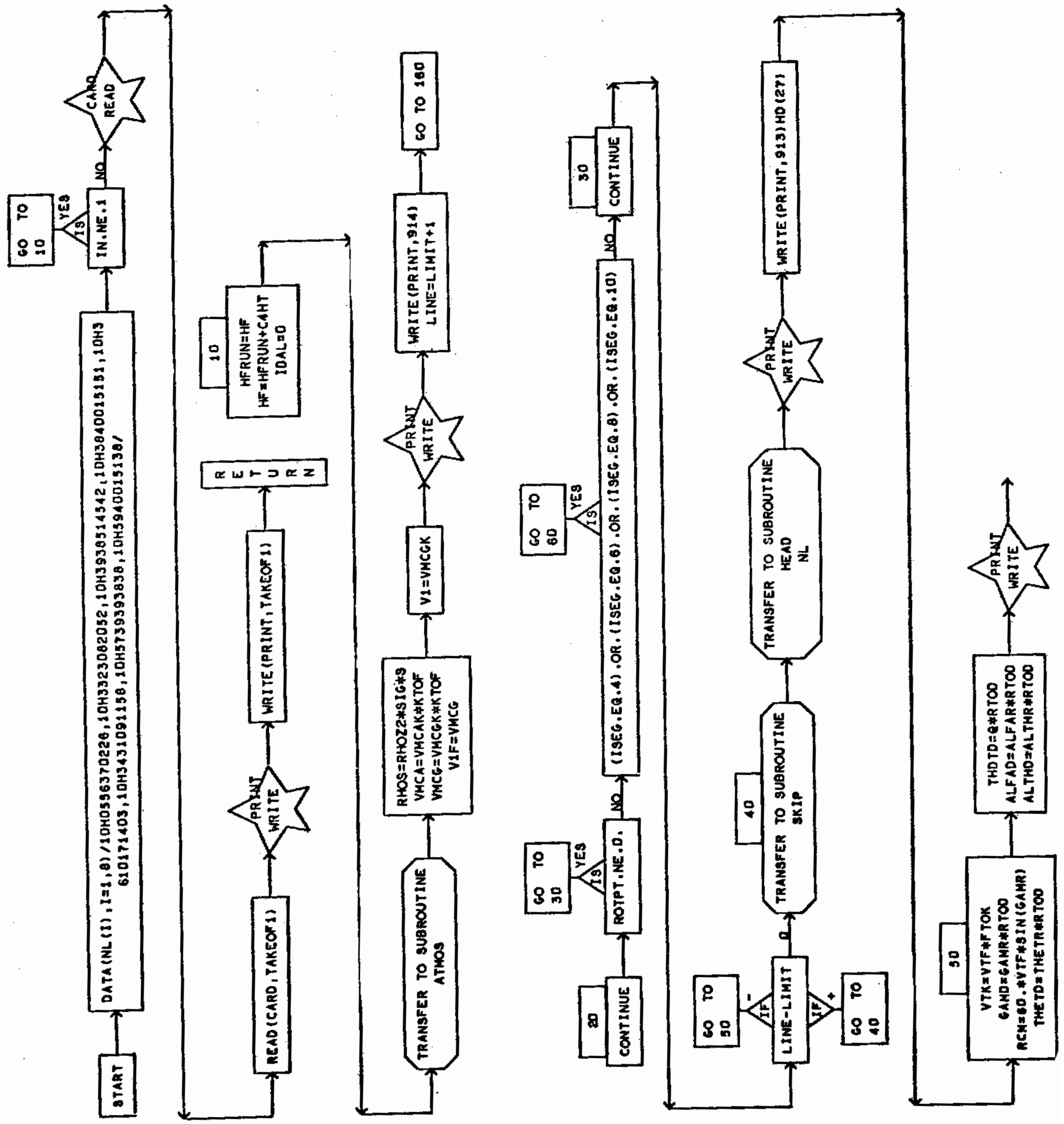


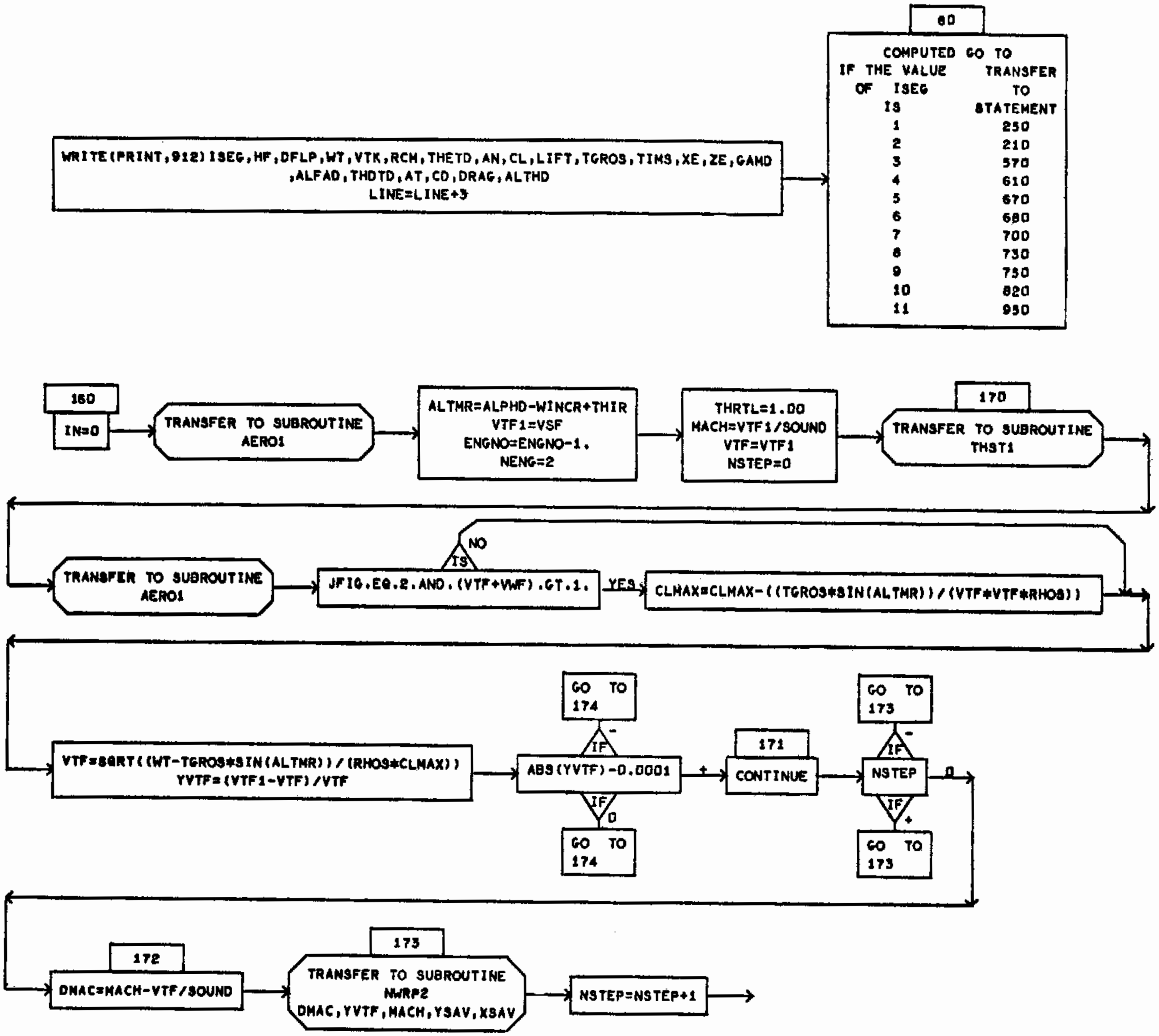
SUBROUTINE SMLT2(NSOL,CN)

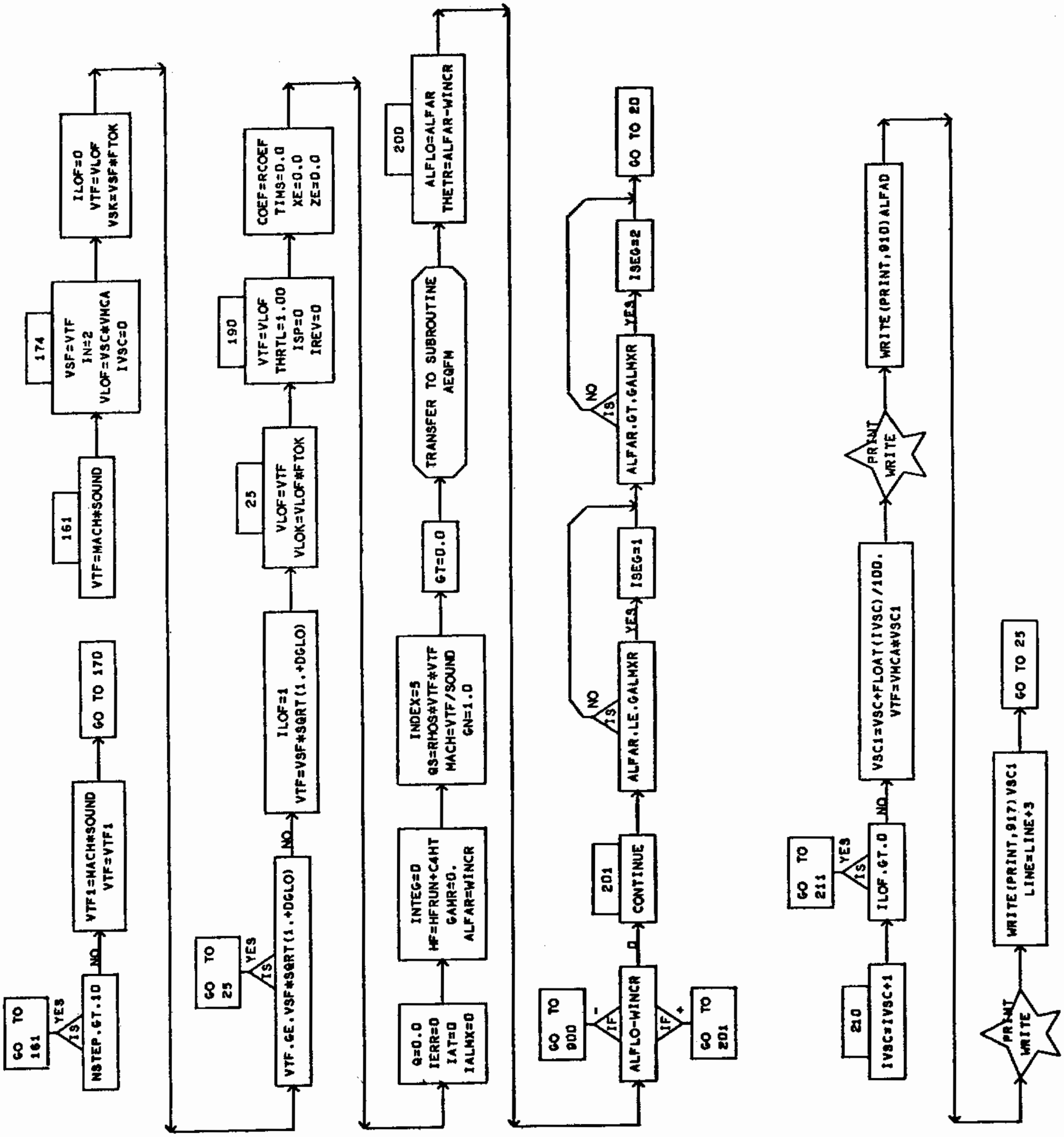


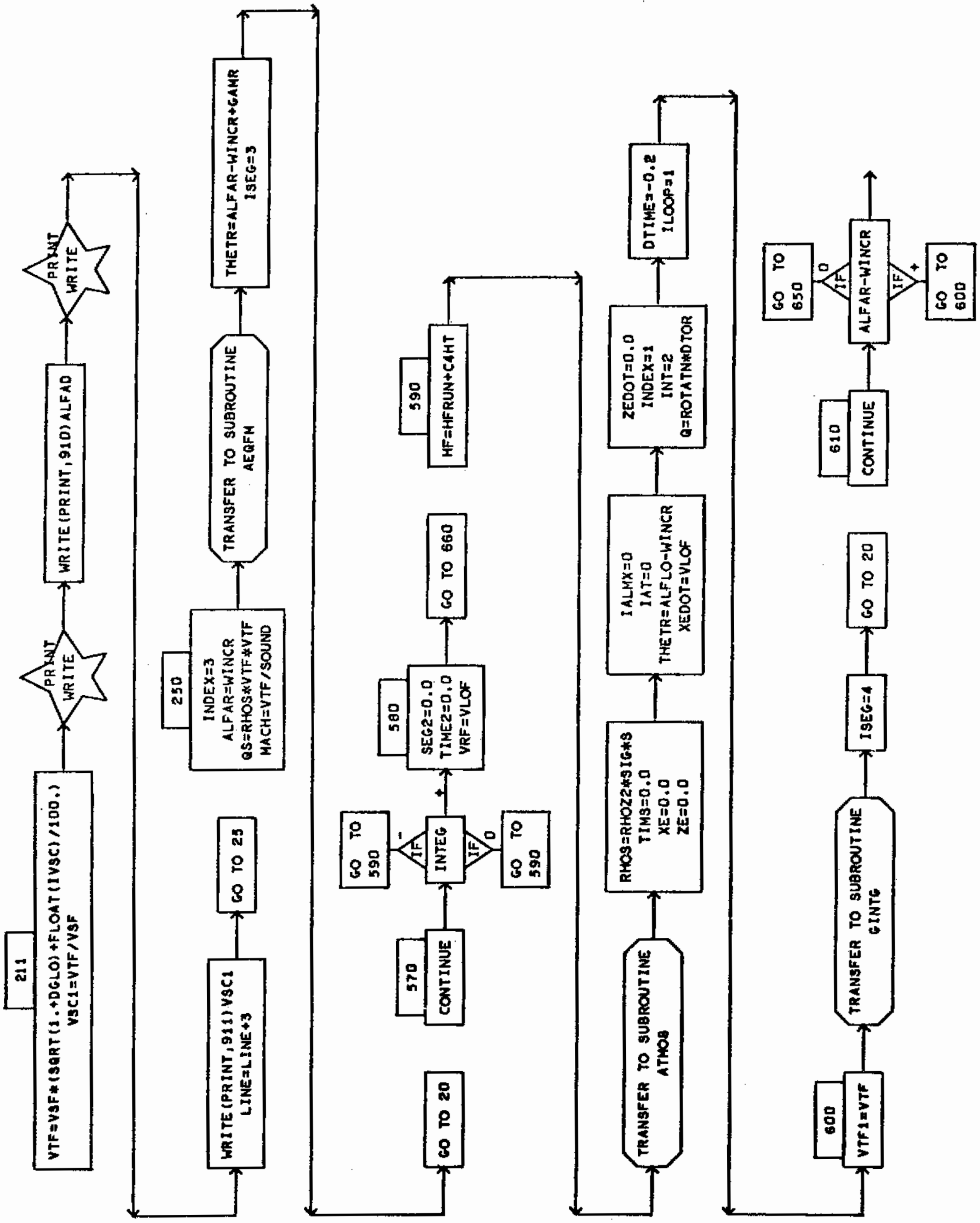

GO TO 100
IF
-
NSOL-1
IF
0
GO TO 120
+
GO TO 100
100
REPEAT TO 110
FOR
I=2,2+1,...,NSOL
J1=I+1
CN(I,1)=CN(I,J1)
JF=I-1
REPEAT TO 110
FOR
J=1,1+1,...,JF
J2=J+1
110
CN(I,1)=CN(I,1)-CN(J2,J1)*CN(J,1)
120
RETURN

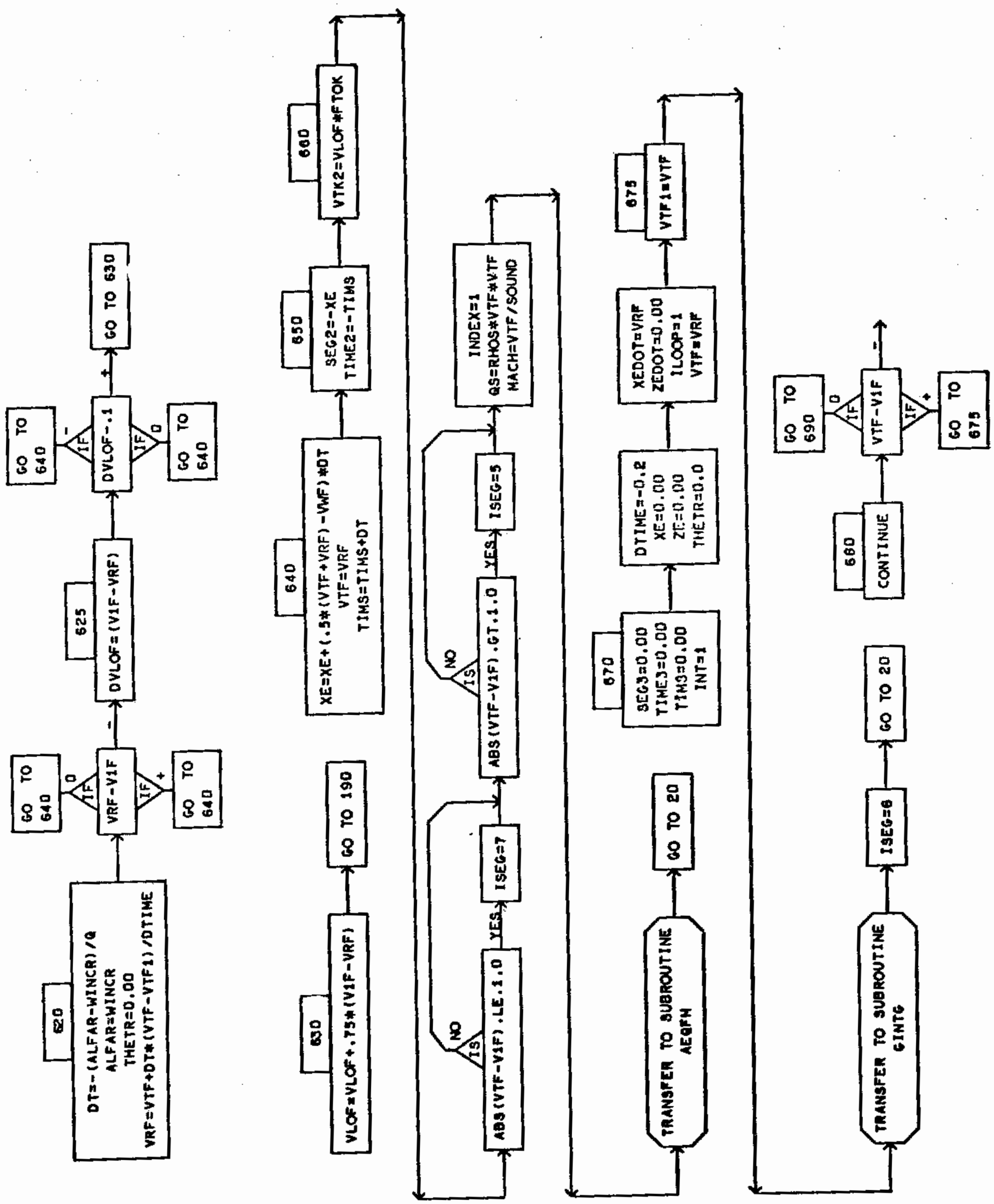
SUBROUTINE TAKEOFF

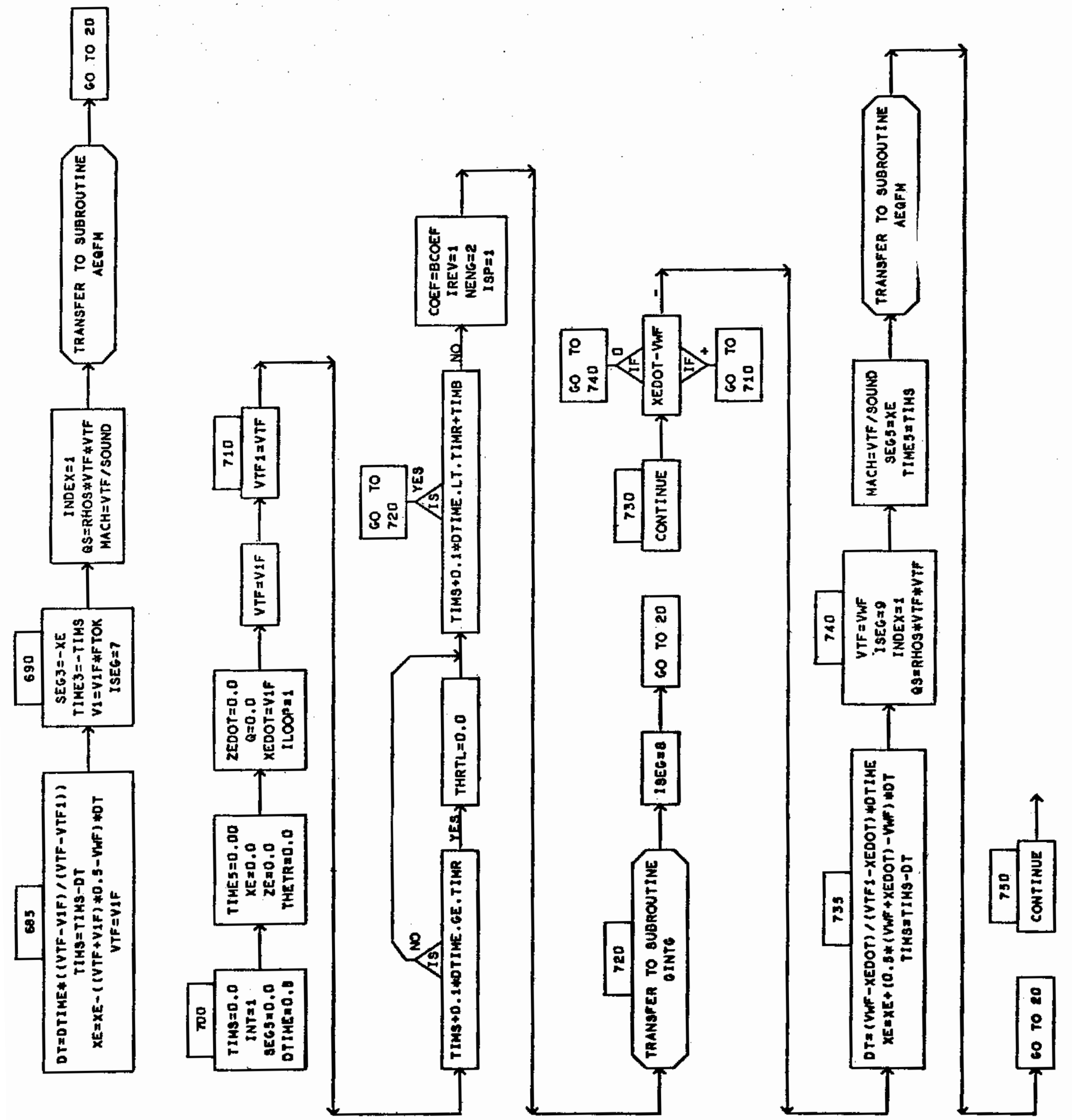


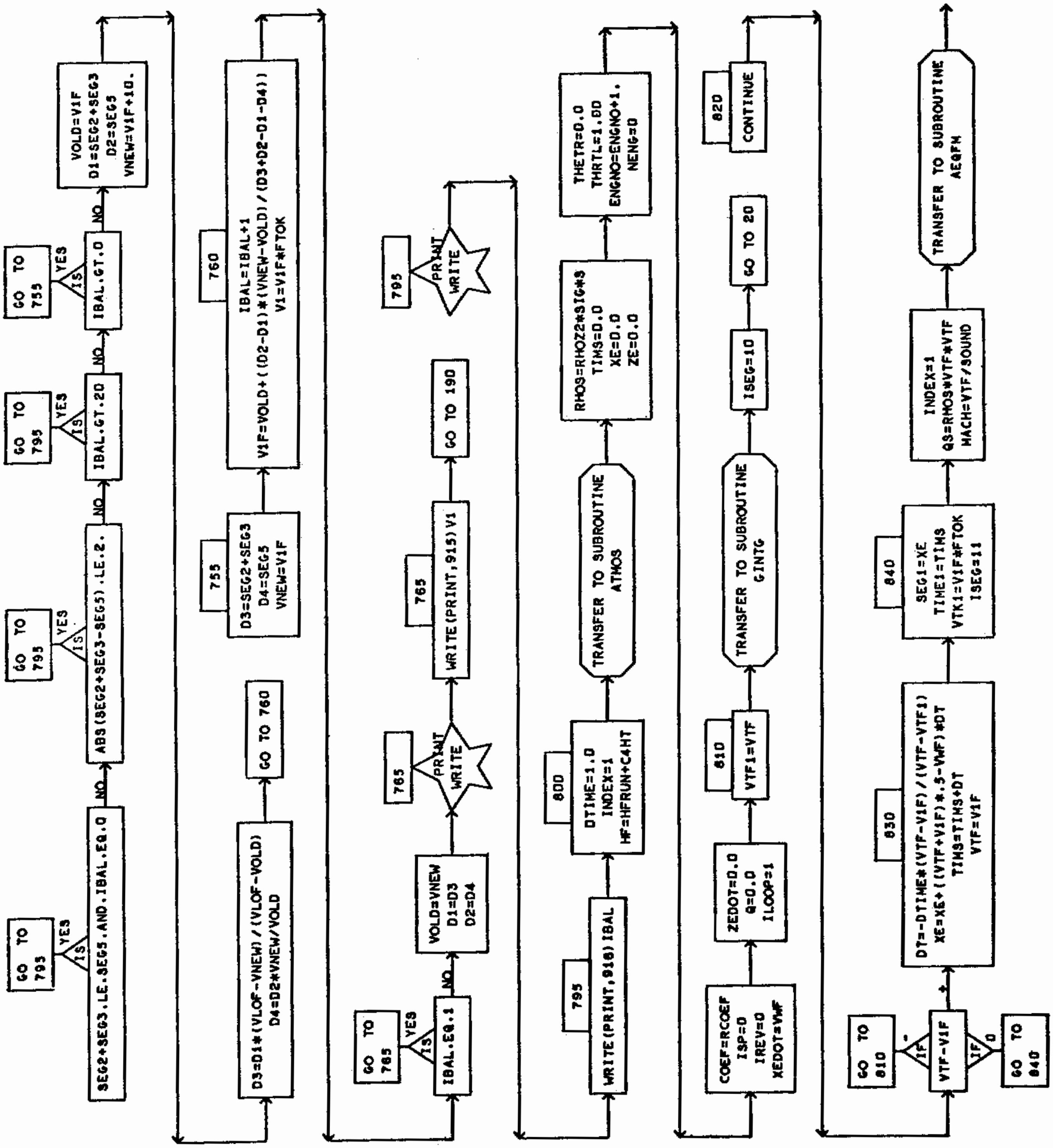

WRITE(PRINT,912)ISEG,HF,DFLP,WT,VTK,RCH,THETD,AN,CL,LIFT,TGROS,TIMS,XE,ZE,GAMD
,ALFAD,THDTD,AT,CD,DRAG,ALTHD
LINE=LINE+3
60
COMPUTED GO TO
IF THE VALUE OF ISEG IS
TRANSFER TO STATEMENT
1 250
2 210
3 570
4 610
5 670
6 680
7 700
8 730
9 750
10 820
11 950
160
IN=0
TRANSFER TO SUBROUTINE
AERO1
ALTMR=ALPHD-WINCR+THIR
VTF1=VSF
ENGNO=ENGNO-1.
NENG=2
THRTL=1.00
MACH=VTF1/SOUND
VTF=VTF1
NSTEP=0
170
TRANSFER TO SUBROUTINE
THST1
TRANSFER TO SUBROUTINE
AERO1
JFIG.EQ.2.AND.(VTF+VWF).GT.1.
IS
NO
YES
CLMAX=CLMAX-((TGROS*SIN(ALTMR))/(VTF*VTF*RHOS))
VTF=SQRT((WT-TGROS*SIN(ALTMR))/(RHOS*CLMAX))
YVTF=(VTF1-VTF)/VTF
ABS(YVTF)-0.0001
IF
-
0
+
GO TO
174
GO TO
174
171
CONTINUE
NSTEP
GO TO
173
GO TO
173
172
DMAC=MACH-VTF/SOUND
173
TRANSFER TO SUBROUTINE
NWRP2
DMAC,YVTF,MACH,YSAV,XSAV
NSTEP=NSTEP+1

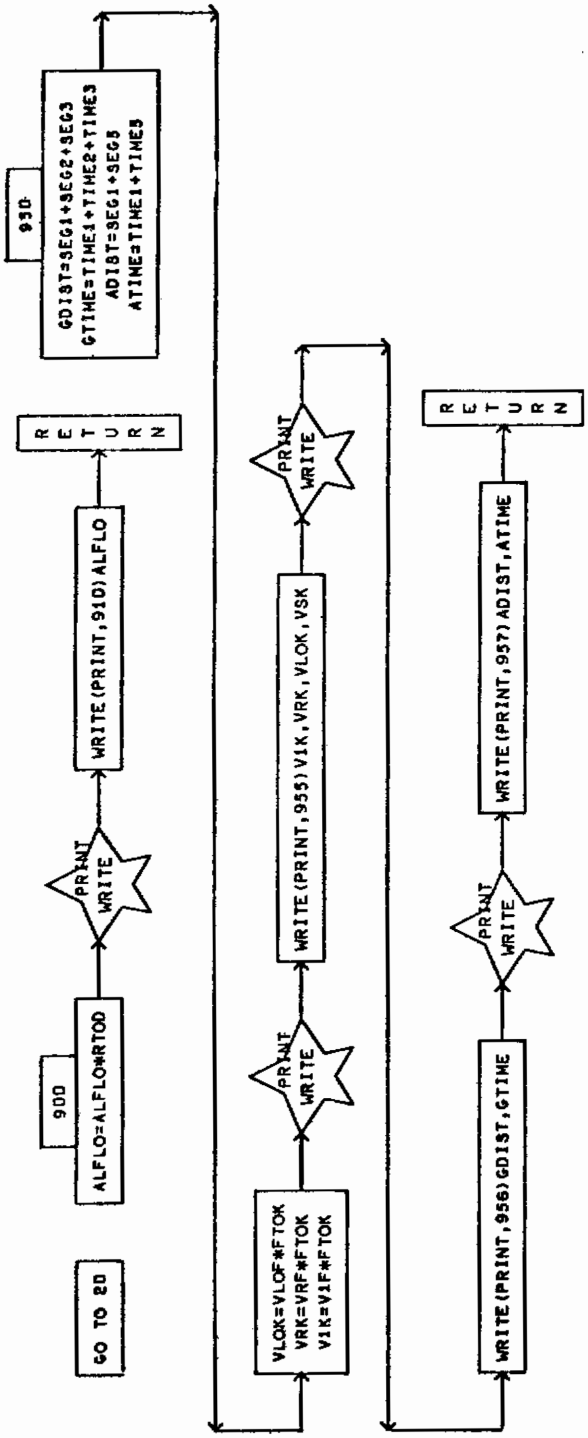

GO TO 161
IS
YES
NO
NSTEP.GT.1D
VTF1=MACH*SOUND
VTF=VTF1
GO TO 17D
161
VTF=MACH*SOUND
174
VSF=VTF
IN=2
VLOF=VSC*VMCA
IVSC=0
ILOF=0
VTF=VLOF
VSK=VSF*FTOK
GO TO 25
VTF.GE.VSF*SQRT(1.+DGLO)
ILOF=1
VTF=VSF*SQRT(1.+DGLO)
25
VLOF=VTF
VLOK=VLOF*FTOK
190
VTF=VLOF
THRTL=1.00
ISP=0
IREV=0
COEF=RCOEF
TIMS=0.0
XE=0.0
ZE=0.0
Q=0.0
IERR=0
IAT=0
IALMX=0
INTEG=0
HF=HFRUN+C4HT
GAHR=0.
ALFAR=WINCR
INDEX=5
QS=RHOS*VTF*VTF
MACH=VTF/SOUND
GN=1.0
GT=0.0
TRANSFER TO SUBROUTINE
AEQFM
200
ALFLO=ALFAR
THETR=ALFAR-WINCR
GO TO 900
GO TO 201
IF
ALFLO-WINCR
201
CONTINUE
ALFAR.LE.GALMXR
ISEG=1
ALFAR.GT.GALMXR
ISEG=2
GO TO 20
210
IVSC=IVSC+1
GO TO 211
ILOF.GT.0
VSC1=VSC+FLOAT(IVSC)/100.
VTF=VMCA*VSC1
PRINT
WRITE
WRITE(PRINT,910)ALFAD
WRITE(PRINT,917)VSC1
LINE=LINE+3
GO TO 25

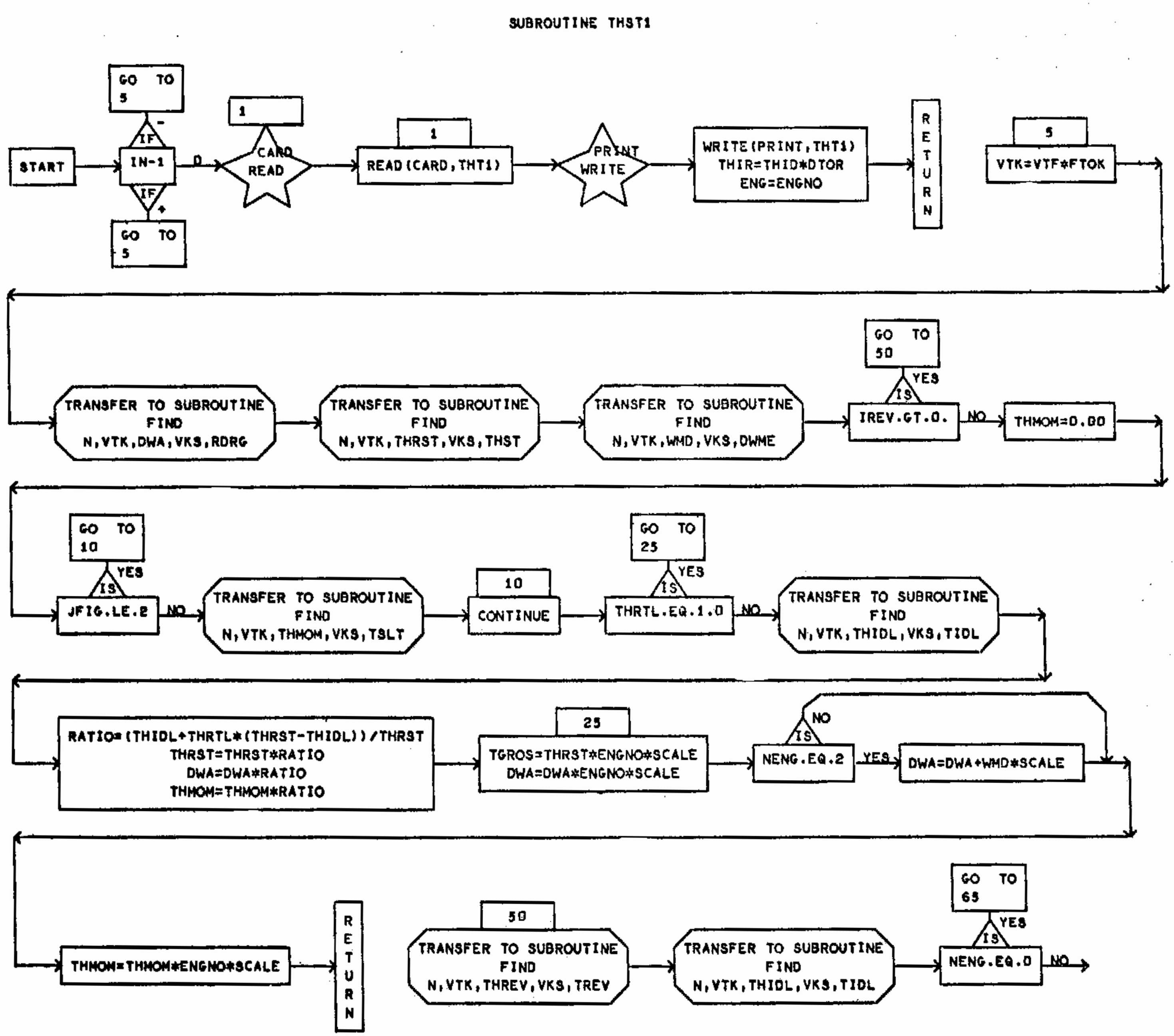

211
VTF=VSF*(SQRT(1.+DGLO)+FLOAT(IVSC)/100.)
VSC1=VTF/VSF
PRINT WRITE
WRITE (PRINT,910) ALFAD
PRINT WRITE
WRITE (PRINT,911) VSC1
LINE=LINE+3
GO TO 25
250
INDEX=3
ALFAR=WINCR
QS=RHOS*VTF*VTF
MACH=VTF/SOUND
TRANSFER TO SUBROUTINE AEQFM
THETR=ALFAR-WINCR+GAMR
ISEG=3
GO TO 20
570
CONTINUE
GO TO 590
-
IF
INTEG
IF
0
GO TO 590
+
580
SEG2=0.0
TIME2=0.0
VRF=VLOF
GO TO 660
590
HF=HFRUN+C4HT
TRANSFER TO SUBROUTINE ATMOS
RHOS=RHOZ2*SIG*S
TIMS=0.0
XE=0.0
ZE=0.0
IALMX=0
IAT=0
THETR=ALFLO-WINCR
XEDOT=VLOF
ZEDOT=0.0
INDEX=1
INT=2
Q=ROTATN*DTOR
DTIME=-0.2
ILOOP=1
600
VTF1=VTF
TRANSFER TO SUBROUTINE GINTG
ISEG=4
GO TO 20
610
CONTINUE
GO TO 650
0
IF
ALFAR-WINCR
IF
+
GO TO 600

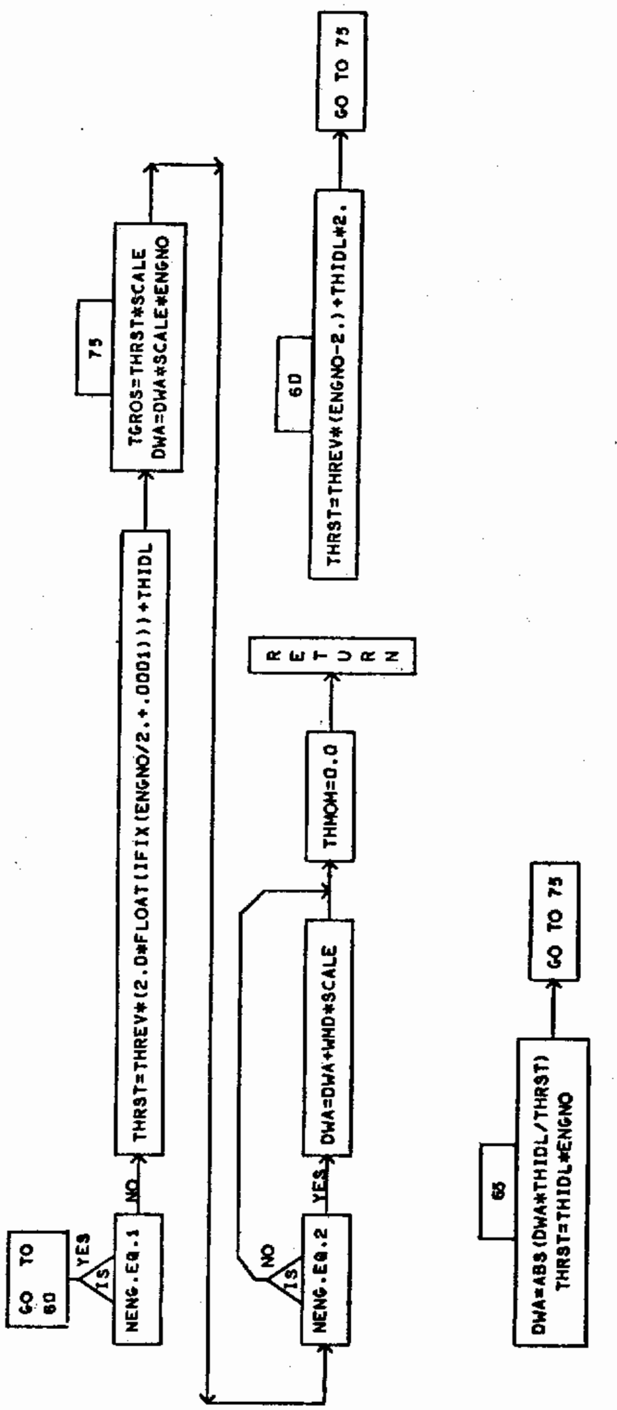

620
DT=-(ALFAR-WINCR)/Q
ALFAR=WINCR
THETR=0.00
VRF=VTF+DT*(VTF-VTF1)/DTIME
VRF-V1F
IF
GO TO 640
625
DVLOF=(V1F-VRF)
DVLOF-.1
GO TO 630
630
VLOF=VLOF+.75*(V1F-VRF)
GO TO 190
640
XE=XE+(.5*(VTF+VRF)-VWF)*DT
VTF=VRF
TIMS=TIMS+DT
650
SEG2=-XE
TIME2=-TIMS
660
VTK2=VLOF*FTOK
ABS(VTF-V1F).LE.1.0
IS
NO
YES
ISEG=7
ABS(VTF-V1F).GT.1.0
ISEG=5
INDEX=1
QS=RHOS*VTF*VTF
MACH=VTF/SOUND
TRANSFER TO SUBROUTINE AEQFN
GO TO 20
670
SEG3=0.00
TIME3=0.00
TIM3=0.00
INT=1
DTIME=-0.2
XE=0.00
ZE=0.00
THETR=0.0
XEDOT=VRF
ZEDOT=0.00
ILOOP=1
VTF=VRF
675
VTF1=VTF
TRANSFER TO SUBROUTINE GINTG
ISEG=6
GO TO 20
680
CONTINUE
VTF-V1F
GO TO 690
GO TO 675


685
DT=DTIME*((VTF-V1F)/(VTF-VTF1))
TIMS=TIMS-DT
XE=XE-((VTF+V1F)*0.5-VWF)*DT
VTF=V1F
690
SEG3=-XE
TIME3=-TIMS
V1=V1F*FTOK
ISEG=7
INDEX=1
QS=RHOS*VTF*VTF
MACH=VTF/SOUND
TRANSFER TO SUBROUTINE
AEQFM
GO TO 20
700
TIMS=0.0
INT=1
SEG5=0.0
DTIME=0.8
TIME5=0.00
XE=0.0
ZE=0.0
THETR=0.0
ZEDOT=0.0
Q=0.0
XEDOT=V1F
ILOOP=1
VTF=V1F
710
VTF1=VTF
IS
TIMS+0.1*DTIME.GE.TIMR
NO
YES
THRTL=0.0
TIMS+0.1*DTIME.LT.TIMR+TIMB
GO TO
720
NO
COEF=BCOEF
IREV=1
NENG=2
ISP=1
720
TRANSFER TO SUBROUTINE
GINTG
ISEG=8
GO TO 20
730
CONTINUE
IF
XEDOT-VWF
0
GO TO
740
+
GO TO
710
-
735
DT=(VWF-XEDOT)/(VTF1-XEDOT)*DTIME
XE=XE+(0.5*(VWF+XEDOT)-VWF)*DT
TIMS=TIMS-DT
740
VTF=VWF
ISEG=9
INDEX=1
QS=RHOS*VTF*VTF
MACH=VTF/SOUND
SEG5=XE
TIME5=TIMS
TRANSFER TO SUBROUTINE
AEQFM
GO TO 20
750
CONTINUE

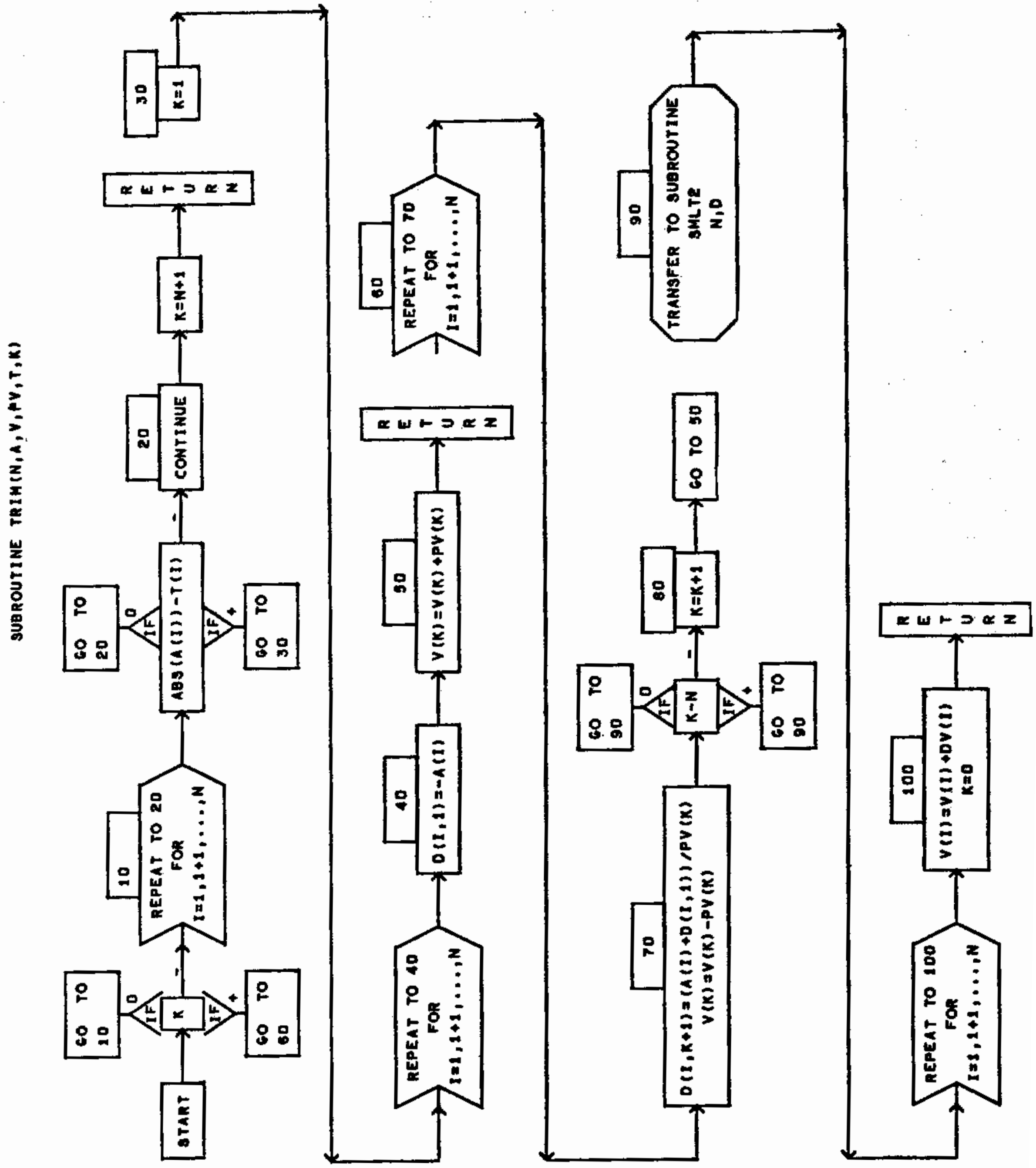

SEG2+SEG3.LE.SEG5.AND.IBAL.EQ.0
IS
YES
GO TO 795
NO
ABS(SEG2+SEG3-SEG5).LE.2.
YES
GO TO 795
NO
IBAL.GT.20
YES
GO TO 795
NO
IBAL.GT.0
YES
GO TO 755
NO
VOLD=V1F
D1=SEG2+SEG3
D2=SEG5
VNEW=V1F+10.
D3=D1*(VLOF-VNEW)/(VLOF-VOLD)
D4=D2*VNEW/VOLD
GO TO 760
755
D3=SEG2+SEG3
D4=SEG5
VNEW=V1F
760
IBAL=IBAL+1
V1F=VOLD+((D2-D1)*(VNEW-VOLD)/(D3+D2-D1-D4))
V1=V1F*FTOK
IBAL.EQ.1
YES
GO TO 765
NO
VOLD=VNEW
D1=D3
D2=D4
765
PRINT
WRITE
765
WRITE(PRINT,915)V1
GO TO 190
795
PRINT
WRITE
795
WRITE(PRINT,916)IBAL
800
DTIME=1.0
INDEX=1
HF=HFRUN+C4HT
TRANSFER TO SUBROUTINE
ATMOS
RHOS=RHOZ2*SIG*S
TIMS=0.0
XE=0.0
ZE=0.0
THETR=0.0
THRTL=1.00
ENGNO=ENGNO+1.
NENG=0
COEF=RCOEF
ISP=0
IREV=0
XEDOT=VWF
ZEDOT=0.0
Q=0.0
ILOOP=1
810
VTF1=VTF
TRANSFER TO SUBROUTINE
GINTG
ISEG=10
GO TO 20
820
CONTINUE
IF
VTF-V1F
-
GO TO 810
0
GO TO 840
+
830
DT=-DTIME*(VTF-V1F)/(VTF-VTF1)
XE=XE+((VTF+V1F)*.5-VWF)*DT
TIMS=TIMS+DT
VTF=V1F
840
SEG1=XE
TIME1=TIMS
VTK1=V1F*FTOK
ISEG=11
INDEX=1
QS=RHOS*VTF*VTF
MACH=VTF/SOUND
TRANSFER TO SUBROUTINE
AEQFN


GO TO 20
900
ALFLO=ALFLO*RTOD
PRINT
WRITE
WRITE(PRINT,910) ALFLO
RETURN
930
GDIST=SEG1+SEG2+SEG3
GTIME=TIME1+TIME2+TIME3
ADIST=SEG1+SEG5
ATIME=TIME1+TIME5
VLOK=VLOF*FTOK
VRK=VRF*FTOK
V1K=V1F*FTOK
PRINT
WRITE
WRITE(PRINT,955) V1K,VRK,VLOK,VSK
PRINT
WRITE
WRITE(PRINT,956)GDIST,GTIME
PRINT
WRITE
WRITE(PRINT,957) ADIST,ATIME
RETURN


SUBROUTINE THST1
START
IF
IN-1
IF
GO TO 5
GO TO 5
-
+
0
1
CARD READ
1
READ(CARD,THT1)
PRINT WRITE
WRITE(PRINT,THT1)
THIR=THID*DTOR
ENG=ENGNO
RETURN
5
VTK=VTF*FTOK
TRANSFER TO SUBROUTINE FIND N,VTK,DWA,VKS,RDRG
TRANSFER TO SUBROUTINE FIND N,VTK,THRST,VKS,THST
TRANSFER TO SUBROUTINE FIND N,VTK,WMD,VKS,DWME
IS
IREV.GT.0.
YES
GO TO 50
NO
THMOM=0.00
IS
JFIG.LE.2
YES
GO TO 10
NO
TRANSFER TO SUBROUTINE FIND N,VTK,THMOM,VKS,TSLT
10
CONTINUE
IS
THRTL.EQ.1.0
YES
GO TO 25
NO
TRANSFER TO SUBROUTINE FIND N,VTK,THIDL,VKS,TIDL
RATIO=(THIDL+THRTL*(THRST-THIDL))/THRST
THRST=THRST*RATIO
DWA=DWA*RATIO
THMOM=THMOM*RATIO
25
TGROS=THRST*ENGNO*SCALE
DWA=DWA*ENGNO*SCALE
IS
NENG.EQ.2
NO
YES
DWA=DWA+WMD*SCALE
THMOM=THMOM*ENGNO*SCALE
RETURN
50
TRANSFER TO SUBROUTINE FIND N,VTK,THREV,VKS,TREV
TRANSFER TO SUBROUTINE FIND N,VTK,THIDL,VKS,TIDL
IS
NENG.EQ.0
YES
GO TO 65
NO



NENG.EQ.1
IS
YES
NO
GO TO 60
THRST=THREV*(2.0*FLOAT(IFIX(ENGNO/2.+.0001)))+THIDL
75
TGROS=THRST*SCALE
DWA=DWA*SCALE*ENGNO
NENG.EQ.2
IS
YES
NO
DWA=DWA+WND*SCALE
THMOM=0.0
RETURN
60
THRST=THREV*(ENGNO-2.)+THIDL*2.
GO TO 75
65
DWA=ABS(DWA*THIDL/THRST)
THRST=THIDL*ENGNO
GO TO 75


SUBROUTINE TRIM(N,A,V,PV,T,K)
START
K
IF
0
-
+
GO TO 10
GO TO 60
10
REPEAT TO 20 FOR I=1,1+1,...,N
ABS(A(I))-T(I)
GO TO 20
GO TO 30
20
CONTINUE
K=N+1
RETURN
30
K=1
REPEAT TO 40 FOR I=1,1+1,...,N
40
D(I,1)=-A(I)
50
V(K)=V(K)+PV(K)
RETURN
60
REPEAT TO 70 FOR I=1,1+1,...,N
70
D(I,K+1)=(A(I)+D(I,1))/PV(K)
V(K)=V(K)-PV(K)
K-N
GO TO 90
GO TO 90
80
K=K+1
GO TO 50
90
TRANSFER TO SUBROUTINE SMLT2 N,D
REPEAT TO 100 FOR I=1,1+1,...,N
100
V(I)=V(I)+DV(I)
K=0
RETURN

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 Aerospace Division General Dynamics Corporation San Diego, California 92138	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE
STOL Tactical Aircraft Investigation
Volume III - Performance Ground Rules and Methods
Book 2 - Takeoff and Landing Digital Computer Program

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final Report (7 June 1971 to 31 January 1973)

5. AUTHOR(S) (First name, middle initial, last name)
J. Hebert, Jr., C. A. Whitney

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
May 1973	126	7

8a. CONTRACT OR GRANT NO. F33615-71-C-1754	9a. ORIGINATOR'S REPORT NUMBER(S) GDCA-DHG73-001
b. PROJECT NO. 643A - Task 0001	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFFDL TR-73-21-Vol III Book 2
d.	

10. DISTRIBUTION STATEMENT
Approved for Public Release
Distribution Unlimited

11. SUPPLEMENTARY NOTES ---	12. SPONSORING MILITARY ACTIVITY Air Force Flight Dynamics Laboratory (PTA) Wright Patterson AFB, Ohio 45433

13. ABSTRACT The MILSTOL (MILitary STOL) takeoff and landing digital computer program was developed under USAF Contract F33615-71-C-1754, "STOL Tactical Aircraft Investigation," to compute takeoff and landing characteristics of powered-lift STOL aircraft. It calculates a point mass takeoff and/or landing for a trimmed configuration with either externally blown jet flaps, internally blown jet flaps, or mechanical flaps with vectored thrust. Contained in this report are

1. Discussion of assumptions and methods used in the trajectory calculations.
2. Definition of common list variables.
3. Definition of the input variables and sample input data for the externally blown jet flap configuration.
4. Sample output for the externally blown flap configuration.
5. Program listings and flow charts.

The program is written in Fortran IV for use on CDC 6000 series digital computers and requires 37 K_8 central memory for loading and execution.

DD FORM 1 NOV 65 1473

Unclassified
Security Classification

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
1. Externally Blown Flaps						
2. Internally Blown Flaps						
3. Lift/Propulsion Concepts						
4. Mechanical Flaps/Vectored Thrust						
5. STOL Transport Aircraft						
6. Takeoff and Landing Computer Program						

☆U.S.Government Printing Office: 1973 — 758-426/89

Security Classification