

AD-766 943

STOL TACTICAL AIRCRAFT INVESTIGATION.
VOLUME III. PERFORMANCE GROUND RULES
AND METHODS. BOOK 2, TAKEOFF AND LANDING
DIGITAL COMPUTER PROGRAM

J. Hebert, Jr., et al

General Dynamics

Prepared for:

Air Force Flight Dynamics Laboratory

May 1973

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

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

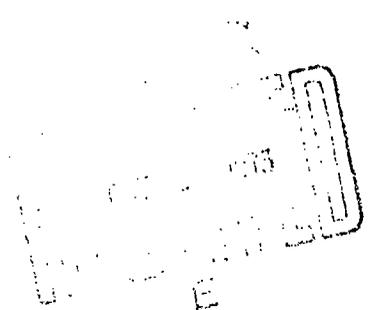
J. Hebsrt, Jr.
C. A. Whitney

Convair Aerospace Division of
General Dynamics Corporation

TECHNICAL REPORT AFFDL-TR-73-21

May 1973

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
US Department of Commerce
Springfield, VA. 22151



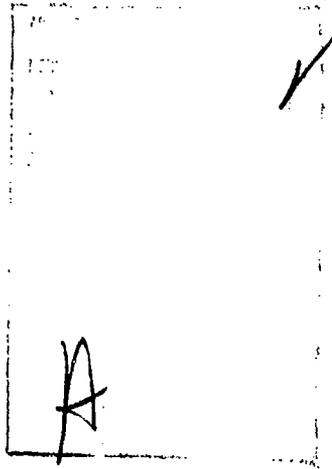
Approved for public release; distribution unlimited

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

AD 766943

NOTICE

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



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

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 May 1973		7a. TOTAL NO. OF PAGES 126 <i>133</i>	7b. NO. OF REFS 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		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned <i>(via report)</i> AFFL TR-73-21-Vol III Book 2	
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 <ol style="list-style-type: none">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 1473
1 NOV 65

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
<ol style="list-style-type: none"> 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 						

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 K₈ central memory for loading and execution. The program is compatible with both the CDC RUN and FTN compiler systems.

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
1	INTRODUCTION	1-1
2	PROGRAM DISCUSSION	2-1
2.1	EXECUTIVE PROGRAM (MILSTOL)	2-2
2.2	MANEUVER DRIVING SUBROUTINES	2-2
2.2.1	Takeoff Trajectory Subroutine (TAKEOFF)	2-2
2.2.2	Landing Trajectory Subroutine (LANDING)	2-3
2.3	PHYSICAL DATA SUBROUTINES	2-4
2.3.1	Atmospheric Properties Subroutine (ATMOS)	2-4
2.3.2	Aerodynamic Data Subroutine (AERO1)	2-5
2.3.3	Propulsion Characteristics Subroutine (THST1)	2-6
2.4	GENERAL USE SUBROUTINES	2-6
2.4.1	AEQFM	2-7
2.4.2	GILL	2-7
2.4.3	GINTG	2-8
2.4.4	HEAD Subroutine	2-8
2.4.5	NWRP2	2-8
2.4.6	SKIP	2-8
2.4.7	SMLT2	2-8
2.4.8	TRIM	2-8
2.5	DATA HANDLING SUBROUTINES	2-8
2.5.1	ALIN	2-8
2.5.2	FIND	2-8
2.5.3	INTP2	2-9
2.5.4	KABD	2-9
2.5.5	LAGRA	2-9
2.5.6	LOOK	2-9
2.5.7	SL	2-9
3	COMMON LIST VARIABLES	3-1
4	INPUT VARIABLES AND SAMPLE CASE	4-1
4.1	INPUT VARIABLES	4-1
4.2	SAMPLE CASE	4-1
5	REFERENCES	5-1
<u>Appendix</u>		
I	PROGRAM AND SUBROUTINE SOURCE LISTINGS	I-1
II	PROGRAM AND SUBROUTINE FLOW CHARTS	II-1

LIST OF FIGURES

Figure		Page
1-1	Balanced Field Takeoff Ground Rules For STAI	1-1
1-2	Landing Ground Rules For STAI	1-2
2-1	MILSTOL Functional Structure	2-1
2-2	Force And Angle System Used In The MILSTOL Program	2-7

LIST OF TABLES

Table		Page
3-1	Description Of Labeled Common Blocks	3-1
3-2	Subroutines In Which Labeled Common Blocks Are Used	3-2
3-3	Definition Of Common Block Variables	3-3
4-1	Definition Of MILSTOL Input Variables	4-2
4-2	Definition Of SKIP Title Cards	4-2
4-3	Definition Of Takeoff Input Variables	4-2
4-4	Definition Of Landing Input Variables	4-3
4-5	Definition Of AERO1 Input Variables	4-3
4-6	Definition Of THST1 Input Variables	4-4
4-7	Sample Input Data	4-4
4-8	Takeoff Calculation Output	4-6
4-9	Landing Calculation Output	4-9

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-

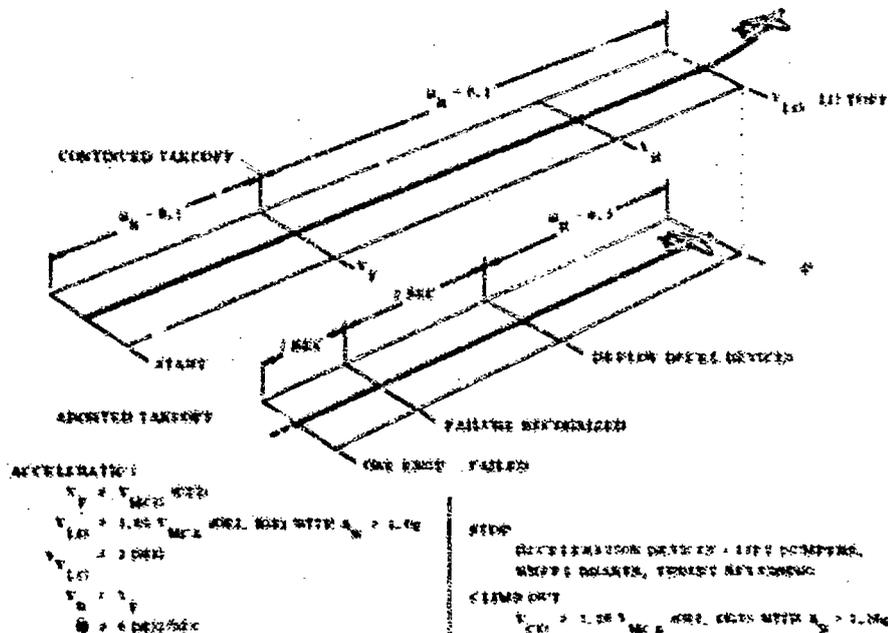


Figure 1-1. Balanced Field Takeoff Ground Rules for STOL.

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.

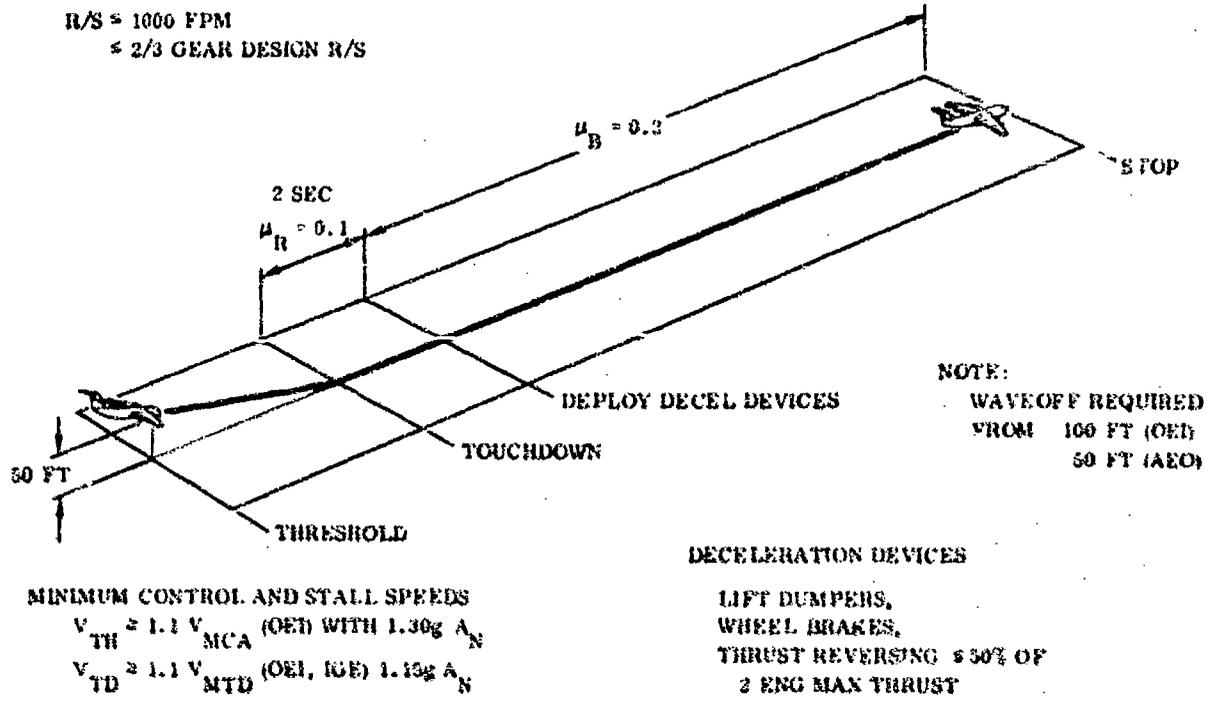


Figure 1-2. Landing Ground Rules for STAL.

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.

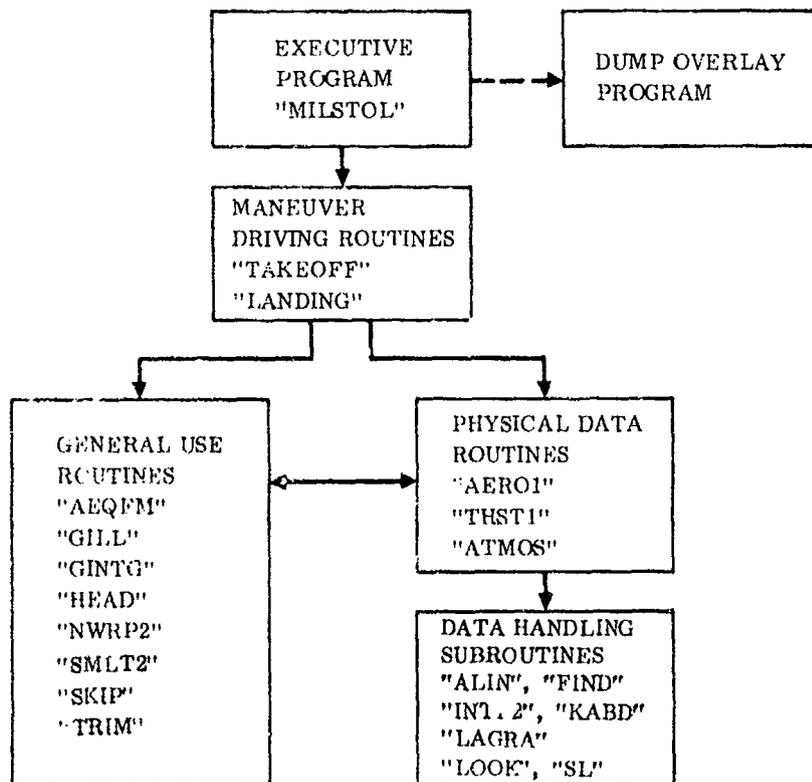


Figure 2-1. MILSTOL Functional Structure.

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:

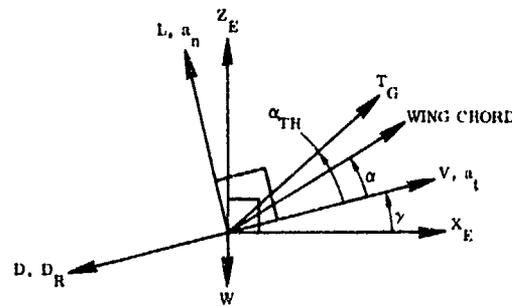
1. All internally blown jet flap configurations are assumed to have cross-ducting so that the spanwise distribution of blowing is always symmetric.
2. Engine-out degradation in the mechanical flap plus vectored thrust configurations is due to the loss of the thrust vector component, not loss of aerodynamic 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
- D_R - RAM DRAG
- L - LIFT
- α - ANGLE-OF-ATTACK
- α_{TH} - ANGLE BETWEEN T_G AND V
- W - WEIGHT
- a_n - NORMAL ACCELERATION

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

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

WHERE

$$G_t = a_t/g, \quad G_n = a_n/g + 1,$$

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

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-Whitson 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											
KABP	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 2-3. Definition of Common Block Variables.

Reproduced from
best available copy.

VARIABLE NAME	COMMON NAME	DEFINITION
ALPAP	LIST2	WING ANGLE OF ATTACK (RADIAN)
ALMAX	LIST3	MAXIMUM ALLOWABLE ANGLE OF ATTACK (RADIAN)
ALPHC	LIST3	WING ANGLE OF ATTACK FOR STALL (RADIAN)
ALTHD	LIST3	ANGLE OF ATTACK OF THE ENGINE AXIS (RADIAN)
LN	LIST2	ACCELERATION NORMAL TO THE VELOCITY VECTOR (FT/SEC SQ)
ANGLF	LG204	NOT USED
ANS	LIST0	THE VALUE OF 'A' RETURNED FROM SUBROUTINE LOOK
ANC2	LIST0	THE VALUE OF 'A' RETURNED FROM SUBROUTINE LOOK
AP	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)
AS	LIST3	WING SPAN (FT)
CADP	LIST1	(INTEGER) UNIT FOR READING CARD INPUT
CPAR	LG204	NOT USED
CD	LIST3	DRAW COEFFICIENT
CMACG	LIST3	NOT USED
CL	LIST3	LIFT COEFFICIENT
CMOVD	LIST3	NOT USED
COFF	LIST8	COEFFICIENT OF FRICTION
CLAR	LIST3	NOT USED
CLMAX	LIST3	MAXIMUM LIFT COEFFICIENT
COSAL	LG204	NOT USED
COSTH	LG204	NOT USED
CX	LIST3	NOT USED
CZ	LIST3	NOT USED
CAHT	LIST15	HEIGHT OF WING ABOVE THE GROUND (FT)
DPFX	LIST9	NOT USED
DPFY	LIST9	NOT USED
DPTZ	LIST9	NOT USED
DPFX	LIST9	NOT USED
DPFY	LIST9	NOT USED
DPFZ	LIST9	NOT USED
DELTD	LIST3	NOT USED
DPLD	LIST16	LOAD DEFLECTION (DEG)
DDAC	LIST2	AERODYNAMIC FORCE IN THE DRAG DIRECTION (LBS)
DPNO	LIST5	NOT USED
DPNO	LIST5	NOT USED
DSTWZ	LG204	NOT USED
DY	LIST2	NOT USED
DPHWT	LG204	NOT USED
DPHTH	LG204	NOT USED
DPWWS	LG204	NOT USED
DPJWF	LIST14	TIME INTERVALS INTERVAL (SEC)
DTAO	LIST4	CONVERTS DEGREES TO RADIANS
DWA	LIST9	RUN DRAG FOR ALL ENGINES ALONG THE VELOCITY VECTOR (LBS)
DWLT	LG204	NOT USED
ENGNO	LIST2	THE NUMBER OF ENGINES OPERATING
EPT	LIST4	NOT USED
ELIFT	LIST15	EXCESS FORCE IN THE LIFT DIRECTION (LBS)
FTS	LIST8	CONVERTS FEET/SECOND TO KNOTS
FTSM	LIST8	CONVERTS FEET TO NAUTICAL MILES
FLVL	LIST3	NOT USED
GALEWD	LIST14	ANGLE OF ATTACK FOR SECOND CONTACT (RADIAN)
GAWV	LIST2	FLIGHT PATH ANGLE (RADIAN)
GN	LIST2	LOAD FACTOR NORMAL TO THE VELOCITY VECTOR (G)
GT	LIST8	LOAD FACTOR PARALLEL TO THE VELOCITY VECTOR (G)
GP	LIST8	ACCELERATION DUE TO GRAVITY (FT/SEC SQ)
HOLD	LIST1	VARIABLE NAME AVAILABLE FOR OUTPUT HEADINGS
HE	LIST2	ALTITUDE (FT)
HMW	LIST2	NOT USED
ISLW	LIST16	INDEX OF WING MAXIMUM ANGLE OF ATTACK IS REACHED
ISW	LIST4	INDEX TO SELECT AIRPLANE TYPE
ISF	LIST15	NEGATIVE ACCELERATION INDEX DURING TAKEOFF
ISAK	LIST7	DATE (MM/DD/YY)
ISAC	LIST9	FLOOR INDEX
ISAC	LIST9	NOT USED
ISAP	LIST10	LOOP COUNTER IN SUBROUTINE CALL
ISAW	LIST3	NOT USED
ISW	LIST2	DIRECTS CALLER SUBROUTINE TO READ INPUTS ON EXECUTE
ISW	LIST4	OPTION SELECTOR FOR SUBROUTINE LOOK
ISW	LIST5	OUT OF FIELD INDICATOR FOR SUBROUTINE LOOK
ISW	LIST6	SELECTS VARIABLES TO INTERPRET IN SUBROUTINE GEMW
ISW	LIST7	DISPATCHES SUBROUTINE CALLS BY TEST PROGRAM
ISW	LIST11	NOT USED
ISW	CONTROL	CONTROL SECURITY INDICATOR
ISW	CONTROL	LIFT NUMBER INDICATOR
ISW	LIST2	DIRECTS OPTION IN LOOK
ISW	LG204	NOT USED
ISW	CONTROL	INDEX FOR PART OF POWERED LIFT SYSTEM

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

JPOW	LIST4	NOT USED
KTOP	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
NO	LIST9	OPTION SELECTOR FOR SUBROUTINE LOOK
NOFR13)	LIST9	NOT USED
NPNG	CONTROL	INDEX FOR REVERSE THRUST MODE
NNTOP	LIST6	(REAL) CONVERTS NAUTICAL MILES TO FEET
MU	LIST5	(REAL) KINEMATIC VISCOSITY (SQ.FT/SEC)
P:OF	LIST1	(INTEGER) CURRENT PAGE NUMBER
PAMP	LIST5	ATMOSPHERIC AMBIENT PRESSURE (LBS/SQ.FT)
PPA	LGE0Y	NOT USED
POINT	LIST1	(INTEGER) UNIT FOR WRITING PRINTED OUTPUT
PZ	LIST5	SEA LEVEL ATMOSPHERIC PRESSURE (LBS/SQ.FT)
Q	LIST3	AIRCRAFT PITCH RATE (RAD/SEC)
QDOT	LGE0Y	NOT USED
QS	LIST3	DYNAMIC PRESSURE TIMES REFERENCE AREA (LBS)
QXC	LIST3	NOT USED
QXF	LIST2	RATE OF CLIMB (FT/SEC)
QXOF	LIST6	SEA LEVEL STANDARD DAY AIR DENSITY (SLUGS/CUBIC FT)
QMOZ2	LIST6	QMOZ DIVIDED BY 2
QHZ	LGE0Y	NOT USED
QTM	LGE0Y	NOT USED
QTCM	LIST6	CONVERTS RADIAN TO DEGREE
QVA	LGE0Y	NOT USED
S	LIST3	WING REFERENCE AREA (SQ.FT)
SCB	LGE0Y	NOT USED
SMW	LGE0Y	NOT USED
SIG	LIST5	DENSITY RATIO
SINAL	LGE0Y	NOT USED
SIATH	LGE0Y	NOT USED
SOUND	LIST4	SPEED OF SOUND (FT/SEC)
TEMP	LIST4	AIR TEMPERATURE (DEG FAHRENHEIT)
TEMP	LIST5	AIR TEMPERATURE (DEG RANKINE)
TCOOS	LIST6	OR SS THRUST FOR ALL ENGINES ALONG THE THRUST AXIS (LBS)
THETA	LGE0Y	AIRCRAFT ATTITUDE (RADIAN)
THID	LIST3	THRUST INCIDENCE (RADIAN)
THROW	LIST4	GROSS THRUST AT THE SLOT EXIT FOR 100 COMPS (LBS)
THROW	LIST6	NOT USED
THRST	LIST2	GROSS THRUST FOR ENGINE ALONG THE THRUST AXIS (LBS)
THRTL	LIST2	THROTTLE SETTING
THW	LIST2	NOT USED
TIME	LIST2	TIME (SEC)
TZ	LIST4	SEA LEVEL STANDARD DAY TEMPERATURE (DEG RANKINE)
V	LGE0Y	NOT USED
V007	LGE0Y	NOT USED
VW	LIST3	STALL SPEED (FT/SEC)
VW	LIST2	TRUE AIRSPEED (FT/SEC)
VW	LGE0Y	VW SQUARED (FT SQ/SEC SQ)
VW009	LIST3	NOT USED
VW	LIST2	WINDWIND COMPONENT - TRUE AIRSPEED (FT/SEC)
V	LGE0Y	NOT USED
WCT	LGE0Y	NOT USED
WIND	LIST3	WIND INCIDENCE (RADIAN)
WT	LIST2	GROSS WT (LBS)
WA	LIST4	W ARGUMENT TO SUBROUTINE LOOK
W	LGE0Y	NOT USED
WT	LIST2	W DISTANCE IN EARTH AXIS SYSTEM (FT)
WDOT	LIST2	RATE OF CHANGE OF W WITH TIME (FT/SEC)
W	LIST2	NOT USED
W	LIST2	W ARGUMENT TO SUBROUTINE LOOK
W	LIST2	W ARGUMENT TO SUBROUTINE LOOK
W	LGE0Y	NOT USED
W	LIST2	W DISTANCE IN FOOT. AXIS SYSTEM (FT)
WDOT	LIST2	RATE OF CHANGE OF W 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 6 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      NMP= THE NUMBER OF ALTITUDES IN THE ALTITUDE MATRIX
C      WPL= THE LIST OF RUNWAY ALTITUDES (FT.)
C      VVK= 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 AERO1 - NAMELIST AERT1
C            4 SUBROUTINE THST1 - NAMELIST THY1
C            5 SUBROUTINE TAKEOFF - NAMELIST TAKEOFF1
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 3 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

```

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 'TAKEOFF'
C      VCS = RATIO OF LIFTOFF SPEED TO AIR MINIMUM CONTROL SPEED
C      DGL0=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 TIME FOR ENGINE FAILURE (SEC)
C      TIMB = BRAKING DELAY AFTER 'TIMR' (SEC)
C      ROTPT = EQUAL TO ZERO SUPPRESSES 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 TAKEOFF
C      VSC=1.05
C      DGL0=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      MFOB=  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      TRRK=  TIME DFLAY 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 SUPPRESSES 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      MFOB = 50.0
C      ROTATN = 8.0
C      TRRK = 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      CAHT=  HEIGHT OF THE QUARTER CHORD ABOVE THE GROUND (FT)
C      GAMBX=  ANGLE OF ATTACK FOR GROUND CONTACT (DEG)
C      ALPHX =  MAXIMUM ALLOWABLE ANGLE OF ATTACK - USUALLY A CL LIMIT
C           (DEG)
C      ACLMD =  STALL ANGLE OF ATTACK (DEG)
C      FEP =  EQUIVALENT FLAT PLATE AREA OF ADDITIONAL DRAG ITEMS (SQ FT)
C      WINCO =  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      FODCD =  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 CONFIG) AT MAX POWER (LBS)
C      TRRV=MAX REVERSE THRUST (EXPRESSED AS A NEGATIVE VALUE - IN LBS.)
C      OWNE=WINDMILLING DRAG FOR A DEAD ENGINE (IN LBS.)
C
C      THID=THRUST VECTOR INCIDENCE REF. TO A WATER LINE IN DEG.
C      ENGINO=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
C      THE FOLLOWING VARIABLES ARE ENTERED AT TIME OF LOADING AND ARE USED
C      UNTIL OVERRIDDEN BY READING THE APPROPRIATE VARIABLES IN THT1
C      ENGINO = 4.0
C      SCALE = 1.0
C      NENG = 1
    
```

Table 4-7. Sample Input Data.

```

SMAIN NWT=1, WTL=135000., NWF=1, WFL=2500., IATH=3, VKK=0.,
INSEQ=2,5,3,4,7, EXSEQ=5,7, $
SE 13-F2B ENGINE          SAMPLE CASE
FOR AN EXTERNALLY BLOWN FLAP CONFIGURATION
STAKEP1 VSC=1.05, DPLP=25., RCOEF=0.1, BCOEF=0.3, VMCCK=55., VMCCK=55.,
ROTATN=5., TIMR=1., TIMB=2., DGL0=0.1, ROTPY=0., $
SAERT1 S=1550., ALPHX=20., GALHX=20., ACLMD=27., JFIG=2, DCLSP=-2.0,
OCOSP=0.25, EOCL=-0.45, EOCD=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,
 1.91,   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,   3.39,   4.02,
 4.72,   5.39,   6.07,   6.71,   7.10,
 1.50,   2.33,   3.14,   3.93,   4.73,
 5.53,   6.26,   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, -0.48,
 -2.90, -2.90, -2.85, -2.63, -2.79,
 -2.73, -2.66, -2.55, -2.41, -2.10,
 -4.90, -4.90, -4.85, -4.83, -4.77,
 -4.66, -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,

```

```

STHT: 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.,19269.,
RDRG= 0.0,693.,1386.,2080.,2773.,3466.,4160.,4901.,5649.,6412.,7210.,
YIDL=490.,370.,260.,155.,65.,-30.,-125.,-210.,-295.,-370.,-450.,
TRFY=0.0,-3000.,-6000.,-7410.,-7190.,-6970.,-6780.,-6580.,-6400.,-6210.,
-6050.,
S
SMAIN INSEQ=6,3,7, EXSEQ=6,8, S
SLAND: DFLP=50., MFOB=50., ROYATN=8., TBRK=2., TSP=2., TREV=2., APR=1.1,
DGTN=0.3, DGTD=0.15, RCOEF=0.1, BCOEF=0.3, GAMMA=-7.5, RS=12.,
ROTPT=., S
SAERT: ALMX=18., GALMX=20., ACLMD=24., S

```

Table 4-8. Takeoff Calculation Output.

OUTPUT DEFINITIONS - TAKEOFF													
SEGMENT	1	2	3	4	5	6	7	8	9	10	11	HOT TEMPERATURES	
	3 ENGINE GE 11-228 ENGINE S = 1550 SO FT											20.38	
	SAMPLE CASE												
	FOR AN EXTERNALLY BLOWN FLAP CONFIGURATION												
STOL TAKEOFF													
	ALTI- TUDINE (FEET)	FLAP (DEGREES)	WEIGHT (POUNDS)	AIR- SPEED (KNOTS)	R/C ACTUAL (FT/MIN)	THETA (DEGREES)	WIND- COR- RECTED (F/S/S)	CL	LIFT (POUNDS)	THRUST (POUNDS)	CL	LIFT (POUNDS)	THRUST (POUNDS)
	TIME (SECONDS)	DISTANCE (FEET)	HEIGHT (FEET)	FLT. PATH (DEGREES)	ALPHA (DEGREES)	THETA DOT (DEG/SEC)	TAN- ACCEL (F/S/S)	CO	DRAG (POUNDS)	TANST ALPHA (DEGREES)	CO	DRAG (POUNDS)	TANST ALPHA (DEGREES)
2	2500.000 8.333	25.000 0.000	135000.000 0.000	89.250 8.300	0.000 20.031	20.031 0.00000	-0.0007 -1.61236	3.25714 1.27919	116617.137 45799.443	51093.979 28.031			
AIRPLANE LIFTS OFF AT ALPHA = 20.031 DEG VLO INCREASED TO 1.060 TIMES VMCAIR													
2	2503.203 0.000	25.000 0.000	135000.000 0.000	90.500 0.000	0.000 20.462	20.462 0.00000	-0.0006 -1.59083	3.20514 1.25448	115914.902 45774.927	51732.664 28.462			
AIRPLANE LIFTS OFF AT ALPHA = 20.462 DEG VLO INCREASED TO 1.070 TIMES VMCAIR													
2	2503.800 0.000	25.000 0.000	135000.000 0.000	90.950 0.000	0.000 20.106	20.106 0.00000	-0.0005 -1.57239	3.15228 1.23068	117203.248 45757.143	51771.624 28.106			
AIRPLANE LIFTS OFF AT ALPHA = 23.106 DEG VLO INCREASED TO 1.080 TIMES VMCAIR													
1	2503.100 0.000	25.000 0.000	135000.000 0.000	91.000 -2.708	0.008 19.758	19.758 0.00000	-0.0003 -1.53943	3.10163 1.20571	117402.411 45676.539	51811.036 19.758			
3	2503.303 3.303	25.000 0.000	135000.000 0.000	91.000 0.000	-439.323 19.727	17.018 0.00000	-0.00404 -0.00000	3.03790 1.20395	117344.446 45604.051	51811.036 19.727			
5	2503.000 -2.473	25.100 -120.394	135000.000 0.000	89.663 0.000	0.000 0.000	0.000 0.00000	-25.62934 2.68420	.75994 .56378	27461.118 20372.539	51722.493 0.000			
7	2503.003 -11.632	25.000 -156.675	135000.000 0.000	65.000 0.000	0.000 0.000	0.000 0.00000	-27.76504 4.57499	.90742 .69683	18751.650 13232.875	90813.672 0.000			
9	2503.003 11.150	25.000 717.122	135000.000 0.000	0.000 0.000	0.000 0.000	0.000 0.00000	-32.17405 -9.53543	-.95023 -43691	0.000 0.000	490.688 0.000			

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

1	2500.000	25.122	12000.000	65.661	5.000	0.000	-25.62916	.75994	27461.118	5172.433
	-7.472	-192.036	0.000	0.000	0.000	0.000	2.58420	.56374	20372.539	0.000
2	2500.000	25.122	17500.000	42.005	0.000	0.000	-26.72370	.80253	24547.269	51398.769
	-8.281	-583.763	0.000	0.000	0.000	0.000	3.26474	.59090	18074.749	0.000
3	2500.000	25.122	17500.000	0.103	0.000	0.000	-32.17405	-.95023	0.000	490.000
	12.398	474.436	0.000	0.000	0.000	0.000	-9.53547	.43691	0.000	0.000
FIELD CALCULATIONS										
1	2500.000	25.122	17500.000	62.607	0.000	0.000	-22.29262	1.35530	41456.723	60529.025
	21.524	1-35.717	0.000	3.501	0.000	0.000	7.10700	.58616	17929.908	0.000
ENGINE FAILURE SPEED = 82.00 KTS										
ROTATION SPEED = 89.84 KTS										
LIFTOFF SPEED = 91.00 KTS										
STALL SPEED = 70.00 KTS (WITH ONE ENGINE OUT)										
WIND TO LIFTOFF										
DISTANCE = 2500.00 FT										
TIME = 28.22 SEC										
ASBESTIC TAKEOFF										
DISTANCE = 2500.00 FT										
TIME = 24.27 SEC										

**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. Ellert "Low Speed Wind Tunnel Test of a 1/20th Scale STOL Tactical Transport Model With Wing Sweep of 25 Degrees Aspect Ratio of 6.0," 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.

<u>Title</u>	<u>Description</u>	<u>Page</u>
MILSTOL	Main Program	I-1
MILSTOL	Overlay Dump Program	I-3
AEQFM	Equations of Motion Subroutine	I-4
AERO1	Aerodynamic Data Subroutine	I-7
ALIN	Linear Equation Function	I-9
ATMOS	Atmospheric Properties Subroutine	I-10
FIND	1-Dimensional Table Lookup Subroutine	I-12
GILL	Integration Subroutine	I-13
GINTG	Integration Driver Subroutine	I-14
HEAD	Page Heading Subroutine	I-15
INTP2	Curve Fitting Subroutine	I-15
KABD	Hyperbolic Curve Fit Solution Subroutine	I-16
LAGRA	Lagranlan Interpolation Subroutine	I-16
LANDING	Landing Trajectory Driver Subroutine	I-17
LOOK	3-Dimensional Table Lookup Subroutine	I-25
NWRP2	Newton-Wrapson Iteration Subroutine	I-25
SKIP	Page Eject Subroutine	I-30
SL	Linear Slope Function	I-31
SMLT2	Simultaneous Equation Solution Subroutine	I-32
TAKEOFF	Takeoff Trajectory Driver Subroutine	I-33
THST1	Propulsion Data Subroutine	I-41
TRIM	Aircraft Trimming Subroutine	I-43

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
 REAL LIFT, MACH, NU, KTOF, NMTOF

DIMENSION EXSEQ(2), INSEQ(3), 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, OS, THIR, LTHR, CLMAX, AR, CLAR, CMACG, ALPHO, CX,
 * CZ, O, IMOM, WINCR, CMDMP, OSC, 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, GK, GT, AN, AT, AX, AZ, ITRM, VUPPER
 COMMON /LIST9/ ANS, ANSE, ND, XA, YA, ZA, INDIC, IFLAG, NDER(3), DBOX, DSDY,
 * DRDZ, DR2DX, DR2DY, DR2DZ
 COMMON /LIST15/ IALMX, INT, XEDOT, ZEDOT, XE, ZE, ILOOP, DTIME, IAT, FLIFT,
 * OFLP, C4HT, GALMXR
 COMMON /LGEOM/ SHSW, SCR, ODOT, RTH, DTHTH, RWA, DTHWA, DSTHZ, DDLHZ, IYY,
 1 XCG, ZCG, CBAR, PFN, SINTH, COSTH, ANGLE, SINL, COSAL, RHZ, DTHZ, VTSO,
 2 LINT, WDOT, THETR, U, W
 COMMON /CONTROL/ JFIG, IREV, ISP, NENG
 COMMON /LIST99/ IERR

DATARTO, KTOF, NMTOF, RHOZ, RHOZ2, GZ, PZ, TZ/57.2957795, 1.600, 6076.6, .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 NMF = THE NUMBER OF ALTITUDES IN THE ALTITUDE MATRIX
 C HFL = THE LIST OF RUNWAY ALTITUDES (FT.)
 C VWK = THE HEADWIND COMPONENT (KIAS)
 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 AERO1 - NAMELIST AERT1
 C * 4 SUBROUTINE THST1 - NAMELIST THY1
 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

NAMLIST/MAIN/NWT,WTL,NHF,FL,VWK,IATM,TEMP,INSEQ,EXSEQ

```
CARD=5
PRINT=6
IDATE=MIDRF(318)
LIMIT=55
DTOR=1./RTOD
FTOK=1./KTOF
FTONM=1./NMTOF
PAGE=1
1 IN=1
  OFAQ(CARD,MAIN)
  WRITE(PRINT,MAIN)
  I=1
5 INP=INSEQ(1)
  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(1H,10X,ACHENSOB IN NAMLIST 'MAIN', INSEQ OR EXSEQ)
  STOP
90 IF(IIN.GT.1)GO TO 140
  I=I+1
  GO TO 5
150 II=1
  IN = 2
151 IF(EXSEQ(II)-A
  DO 152 J=1,OFQ
  DO 152 K=1,NWT
  WFL=FL(J)
  CALL ATMOS
  VFK=VWK*(KTOP/SCOT(SIG))
  WTL=TL(K)
  IF(OFQ)
  INP=EXSEQ(II)
  GO TO(150,60,11,160).IEX
140 CONTINUE
152 CONTINUE
152 CONTINUE
  II=II+1
  GO TO 151
160 WRITE(PRINT,1001)
1001 FORMAT(1H,10X,10NEEDS 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,DATE,HD(60)
COMMON /LIST2/DELTD,ALFAR,THRSL,GAMR,HIND,VTF,WT,MF,DT,TIMS,XF,
* FUEL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN
COMMON /LIST3/CL,CD,S,OS,THIR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHO,CX,
* CZ,O,INOM,WINCR,CMDMP,OSC,ALMXR,VSF,BW
COMMON /LIST4/FFS,JPOW,TGROS,DWA,THMOM,THREQ
COMMON /LIST5/SIG,SOUND,NU,TEMP,PAMB,IATM,TEMP,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,IYRM,VUPPER
COMMON /LIST9/ANS,ANS2,ND,XA,YA,ZA,INDIC,IFLAG,NDER(3),DBDX,DBDY,
* DBZ,DBZX,DBZY,DBZ2
COMMON /LIST15/IALMX,INT,XEDOT,ZEDOT,XE,ZE,ILOOP,DTIME, IAT,FLIFT,
IDFLP,CANT,GALMXR
COMMON /LGEOM/SMSW,SCR,ODOT,RTH,DTHTH,RWA,DTMWA,OSTHZ,DWLHZ,IYY,
1 XCG,ZCG,CBAR,PFN,SINTH,COSTH,ANGLE,SINAL,COSAL,RHZ,DTMHZ,VTSG,
2 UDOT,WDOT,THETD,U,W
COMMON /CONTROL/JFIG,IREV,ISP,NEVG
COMMON /LIST99/IERR

NAMELIST/DUMP/LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,DFLP,
1 DELTD,ALFAR,THRSL,GAMR,HIND,VTF,WT,MF,DT,TIMS,XF,
2 FUEL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN,CANT,GALMXR
3 CL,CD,S,OS,THIR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHO,CX,
4 CZ,O,INOM,WINCR,CMDMP,OSC,ALMXR,VSF,BW
5 FFS,JPOW,TGROS,DWA,THMOM,THREQ
6 SIG,SOUND,NU,TEMP,PAMB,IATM,TEMP,DSODH,DRHO
7 RTOD,DTOR,KTOF,FTOK,NMTOF,FTONM,RHOZ,RHOZ2,GZ,PZ,TZ
8 INDEX,COEF,GN,GT,AN,AT,AX,AZ,IYRM,VUPPER
9 ANS,ANS2,ND,XA,YA,ZA,INDIC,IFLAG,NDER DBDX,DBDY,
1 DBZ,DBZX,DBZY,DBZ2,IALMX,INT,XEDOT,ZEDOT,XE,ZE,ILOOP,DTIME,
2 IAT,FLIFT SMSW,SCR,ODOT,RTH,DTHTH,RWA,DTMWA,OSTHZ,DWLHZ,IYY,
3 XCG,ZCG,CBAR,PFN,SINTH,COSTH,ANGLE,SINAL,COSAL,RHZ,DTMHZ,VTSG,
4 UDOT,WDOT,THETD,U,W,JFIG,IREV,ISP,IERR

WRITE(6,DUMP)
STOP
END

```

SUBROUTINE AEOFM
C 6400 SUBROUTINE FOR CALCULATING ACCELERATIONS.
C
C THIS SUBROUTINE CALLS FOR THE FOLLOWING ENTRIES.
C THST1, AERO2, TRIM
C
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).
C
C INTEGER CARD,PRINT,PAGE
C RFAL,LIFT,MACH,NU,KTOF,NMTOF,IYY
C
C DIMENSION SA(3),SV(6),SPV(6),ST(3),JA(3),JV(3),A(6),V(6),PV(6),T(6
C )
C COMMON /LIST1/ LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,DATE,ND(60)
C
C COMMON /LIST2/ DELTD,ALFAR,THRTL,GAMR,HIND,VTF,WT,NF,DT,TIMS,XF,PU
C IFL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN
C COMMON /LIST3/ CL,CD,S,QS,THR,ALTHR,CLMAX,AN,CLAR,CMACC,ALPHO,CX,
C ICZ,Q,IMOM,WINCR,CMDMP,QSC,ALMXR,VSF,BW
C COMMON /LIST4/ FFS,JPOW,TGROS,DWA,THNOM,THREQ
C COMMON /LIST5/ SIG,SOUND,NU,TEMP,RAMB,IATM,TEMP,DSOON,DRNO
C COMMON /LIST6/ RTOD,DTOR,KTOF,FTOK,NMTOF,FTONM,RHOZ,RHOZ2,GZ,PZ,YZ
C COMMON /LIST8/ INDEX,COEF,GN,GT,AN,AT,AXE,AZE,I,TRM,VUPPER
C COMMON /LIST9/ QMSW,SCR,QDOT,RYH,DTHM,RWA,DTHWA,DSH2,DWLHZ,IYY,X
C ICG,ZCG,CBAN,MFN,SINTH,COSTH,ANGLE,SINAL,COSAL,RHZ,DTHHZ,VTSO,QDOT,
C ZWOT,THEYR,U,W
C COMMON /LIST99/ IERR
C
C EQUIVALENCE (SA(1),QCOOT), (SA(2),ACN), (SA(3),ACT), (SV(1),DELTD)
C
C DATA ST,SPV/.0003,2*.0002,.2,.0034906584,-.02,.0034906584,.2,B./
C
C RFCM=GZ/WT
C RND5=RHOZ2*SIG*5
C K=-1
C AVTF=0
C EVTF=0.025*SOUND
C NSTEP=0
C
C 8FT SUBSCRIPTS FOR A,V,PV,T.
C
C 1=1
C IF (IMOM.EQ.0) GO TO 10
C J=1:1

```

```

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)/VTSO
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)/I/Y
200 IF (ITRM.EQ.0) GO TO 300

```

C STORE CURRENT VALUES OF ACCELERATIONS (A).

```

DO 210 I=1,NSOL
J=JA(I)

```

C 210 A(I)=SA(J)
CONVERGE VALUES OF(A) BY VARYING VALUES OF (V).

```

CALL TRI(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=C
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

```

SURROUTINE 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,THRSL,GAMR,HIND,VTF,WT,HF,DT,TIMS,XF,FU
IEL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN
COMMON /LIST3/ CL,CD,S,OS,THR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHD,CX,
ICZ,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,
IDFLP,C4HT,GALMXR
COMMON /LGEOM/ SHSW,SCR,QDOT,RTH,DTHH,RWA,DTHWA,DSTHZ,DWLHZ,IYY,X
ICG,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 FRP = 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

NAMLIST /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
 CDRP = FBP/S
 RETURN

PRE

20 RHOS=RHOZ2*SIG*S
 IF (VTF .LE.1.0) II=0
 QS = RHOS*VTF*VTF
 X = ACLMD
 IF (II,EG.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,EG.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,EG.2,AND,IREV,EG.0) GO TO 72
 IF(VTF,LE.1.,AND,JFIG,EG.3,AND,IREV,EG.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(1)+(XA-X(1))*(Y(I+1)-Y(1))/(X(I+1)-X(1))
RETURN
```

```
END
```

```

SURROUTINE 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 TEMP

```

```

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,THRRL,GAMR,HIND,VTF,WT,HF,DT,TIMS,XF
* FUEL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN
COMMON /LIST5/SIG,SOUND,NU,TEMP,PAMB,IATM,TEMP,DSODH,DRHC
DATA NTROPIC,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,.002033033,69620.,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., 3663.7771, 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/NTROPIC ,TROPIC,NPOLAR,POLAR,NHOT,HOT,NCOLD,COLD

```

```

IF((N-1)/5).EQ.4
1 WRITE(PRINT,ATM1)
RETURN

```

```

C STANDARD ATMOSPHERE

```

```

5 IF(HF.GE.36089.239)GO TO 10
PAMB = 2116.2218*(1.-.687958*6E-05*HF)**5.2558761
IF(IATM.GT.0)GO TO 15
TEMP = 1.8*(288.15-.0019812*HF)
GO TO 30
10 IF(HF.GE.65616.798)GO TO 11
PAMB = 472.68164*EXP(1.7345725-.4806347*HF-.24*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(INTROP,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(INPOLAR,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(INHOT,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(INCOLD,HF,TEMR,COLD)
      GO TO 30
170  TEMR = 359.77
      GO TO 30
26  TEMR = TEMR+659.67
30  SIG = .24509246*PAMB/TEMR
      TEMF = TEMR-459.67
      TEMK = TEMR/1.8
      SOLNG = 65.770350*SORT(TEMK)
      NU = .126129E-04*SORT(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/IST1/LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,HD(60)

IF(N.LT.4) GO TO 50
I=1
N=N-1
DO 10 I=1,N
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-1
GO TO 45
35 I=I-1
45 NX=I+3
CALL INTP2(INX,XC,YC,X,Y,A,B,C,D)
RETURN
50 IF(X.LE.XC(1)) I=1
IF(X.GT.XC(N)) I=N
Y = ALIN(I,XC,YC,X)
RETURN
END

```



```

SUBROUTINE GINTG
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
C THIS PROGRAM CALLS THE FOLLOWING SUBROUTINES, AI
C AEOFM,ATMOS,GILL AI
C
C INTEGR CARD,PRINT,PAGE AI
C REAL LIFT,KTOF,MACH,NU,NMTOF,IYY AI
C
COMMON /LIST1/ LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,HD(60) AI
COMMON /LIST2/ DELTD,ALFAR,THRTL,GAMR,HIND,VTF,WT,WF,DT,TIMS,XF,FU AI
IFL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN AI
COMMON /LIST3/ CL,CO,S,OS,THIR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHD,CX, AI
IC 4CM,WINLR,CORRP,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,COFF,GN,GT,AN,AT,AXE,AZE,ITRM,VUPPER AI
COMMON /LIST15/ IALMX,INT,XEDOT,ZEDOT,XE,ZE,ILOOP,OTIME,IAT,FLIFT, AI
IDFLP,CANT,GALMXR
COMMON /LGEOM/ SHSW,SCR,UDOT,RTH,OTHYH,RWA,OTHWA,DSTHZ,DWLHZ,IYY,X AI
ICG,ZCG,CBAR,PFN,SINTH,COSTH,ANGLE,SINAL,COSAL,RHZ,OTHZ,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+WINCQ AI
IF (IALMX) 60,60,50 AI
50 ALFAR=ALMXR AI
O=(XEDOT*ZEDOT-ZEDOT*XEDOT)/VTSQ AI
60 CALL ATMOS AI
MACH=VTF/SOUND AI
OS=RHOZ2*SIG*S*VTSQ AI
CALL AEOFM AI
XF2OT=AXE AI
T*2OT=AZE AI
GO TO (70,70,80,70,80), ILOOP AI
70 TIMS=TIMS+.5*OTIME AI
80 XEDOT=XEDOT-VWF AI
CALL GILL(XEDOT,XF,OXE,ILOOP,OTIME) AI
CALL GILL(XEDOT,XEDOT,OXE2,ILOOP,OTIME) AI
IF (INT=2) 110,100,90 AI
90 CALL GILL(ZEDOT,ZE,OZE,ILOOP,OTIME) AI
CALL GILL(ZEDOT,ZEDOT,OZE2,ILOOP,OTIME) AI
100 CALL GILL(O,THETR,OTH,ILOOP,OTIME) AI
110 GO TO (120,130,130,130,140), ILOOP AI
120 ILOOP=J 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)
INTEGER CARD,PRINT,PAGE
DIMENSION ID(10),MP(10),NL(8)
COMMON /LIST1/ LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,MD(60)
DO 20 I=1,4
  J=2*I-1
  IF (MOD(I,2).NE.0) WRITE (PRINT,30)
30 FORMAT (1H )
  DECODE (10,40,NL(J)) ID
40 FORMAT (5I2)
  DO 10 K=1,10
  KK=ID(K)
10 MP(K)=MD(KK)
  WRITE (PRINT,50) MP
50 FORMAT (12X,10(2X,A10))
20 CONTINUE
  LINE=LINE+6
  RETURN
END

```

AK
AK
AK
AK
AK
AK
AK
AK
AK
AK
AK
AK
AK
AK
AK
AK
AK
AK
AK
AK

```

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

```

C 6400 FORTRAN 900 DEGREE POLYNOMIAL INTERPOLATION SUBROUTINE.
C ENTRY WITH SUBSCRIPTED X AND Y DATA.

```

DIMENSION X(1), Y(1)

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

```

```

SUBROUTINE K&B(NFITS,XC,YC,X)
C 6400 FORTRAN K&B EVALUATION PROGRAM.
C NFITS IS THE NUMBER OF CURVE FITS STRUNG TOGETHER FOR THE
C PARTICULAR VARIABLE.
INTEGER CARD,PRINT,PAGE
COMMON /LIST1/ LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IOATE,HD(60)
COMMON /LIST99/ IERR
DIMENSION X(50)
IERR=0
I=1
10 IF (XC-X(I+4)) 20,20,30
20 YC=X(I)/(XC-X(I+1))+X(I+2)+X(I+3)*XC
  RETURN
30 I=I+5
  IF (I.LE.5*NFITS) GO TO 10
  WRITE (PRINT,40) NFITS,XC,X(I-1)
  IERR=1
  RETURN
C 40 FORMAT (5X,35H UPPER LIMIT OF X EXCEEDED. NFITS=.I3.5H, XC=.E12.5
1.9H, X(I+4)=".E12.5)
  END

```

```

SUBROUTINE LAGRA(X,Y,INDEX)
C 6400 FORTRAN SUBROUTINE WHICH DOES A FOUR POINT LAGRANGIAN INTERPO
DIMENSION X(5),Y(5),DX(10)
IF (INDEX) 30,10,30
10 IN=1+(INDEX*)
  K=0
  DO 20 I=1,4
    I=I+1
  DO 20 J=1,4
    J=J+1
    DX(K)=X(I)-X(J)
  20 CONTINUE
  P0002=DX(2)/DX(5)+DX(3)/DX(6)+DX(4)/DX(7)
  P0003=-DX(1)/DX(5)+DX(3)/DX(6)+DX(4)/DX(9)
  P0004=DX(1)/DX(6)+DX(2)/DX(8)+DX(4)/DX(10)
  P0005=-DX(1)/DX(7)+DX(2)/DX(9)+DX(3)/DX(10)
30 Y(I)=Y(2)+P0002*Y(5)+P0003*Y(6)+P0004*Y(8)+P0005*Y(10)
  RETURN
  END

```

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 VOB5 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,KTOP,MACH,NU,NMTOF

DIMENSION NL(6)

COMMON /LIST1/ LINE,PAGE,LIMIT,CARD,PRINT,INP,IPUNCH,IDATE,ND(60) F
 COMMON /LIST2/ DELTA,ALFAR,THRSL,GAMR,HIND,VTF,WT,HF,DT,TIMS,XF,FU
 IFL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGRD,THV,IN
 COMMON /LIST3/ CL,CD,S,OS,THIR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHD,CX,
 ICZ,O,IMOM,WINCR,CMOMP,OSC,ALMXR,VSF,BW
 COMMON /LIST4/ FFS,JPOW,TGRS,DWA,TKMOM,THREO
 COMMON /LIST5/ SIG,SOUND,NU,TEMP,PAMD,IATM,TEMP,OSODM,DRMO
 COMMON /LIST6/ RTOD,DTOR,KTOP,FTOK,NMTOF,FTONM,RMOZ,RMOZ2,GZ,PZ,TZ
 COMMON /LIST8/ INDEX,COFF,GN,JA,AN,AT,AX,AZ,ITRM,VUPPER
 COMMON /LIST15/ IALMX,INT,XE00T,ZE00T,XE,ZE,ILOOP,DTIME,IAT,PLIFT,
 ICFLE,CANT,GALMXR
 COMMON /LGFORM/ SHSW,SCR,QDOT,RTM,DTATH,RVA,DTMVA,OSTHZ,DWLHZ,IYY,X
 ICG,ZCG,CBAR,PFN,SINTM,COSTM,ANGLE,SINAL,COSAL,RMZ,DTMZZ,VTSQ,UOOT,
 ZNDOT,THETR,U,
 COMMON/CONTROL/JFIG,IREV,ISP,MENG
 COMMON /LIST99/ IERR

DATA DFLP,QOTPY,XFOB,QOTAYN,YBRX,YSP,YREV,APR,RCOEF,BCOEF,OGTM,
 ICGTQ/60.,1.,30.,8.,2.,2.,2.,1.,10.,1.,0.3,0.3,0.15/
 DATA GAMMA,RS/-7.,9.,10./
 DATA INL(1),(-1.,0)/10M596770226,10M323082052,10M3938514542,10M38
 140015151,10M3610171403,10M3431091198,10M5739243838,10M5940015138/

C DEFINITION OF VARIABLES IN NAMELIST 'LANDI':
 C DFLP# FLAP SETTING (DEG)
 C MFOB# OBSTACLE HEIGHT (FT)
 C QOTAYN# ROTATION RATE (DEG/SEC)
 C VMOG# MINIMUM CONTROL SPEED ON THE GROUND (KTS)
 C VMOAK# MINIMUM CONTROL SPEED IN THE AIR (KTS)
 C YBRK# TIME DELAY AFTER TOUCHDOWN FOR BRAKE APPLICATION (SEC)
 C YSD# TIME DELAY AFTER TOUCHDOWN FOR SPOILER DEPLOYMENT (SEC)
 C YSPV# TIME DELAY AFTER TOUCHDOWN FOR THRUST REVERSAL (SEC)
 C SPD# RATIO OF APPROACH SPEED TO MINIMUM CONTROL SPEED
 C OGTM# INCREMENTAL LOAD FACTOR AVAILABLE AT THE OBSTACLE
 C OGTQ# INCREMENTAL LOAD FACTOR AVAILABLE AT TOUCHDOWN
 C BCOEF# 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 SUPPRESSES 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     DGM = 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
IFF,TREV,GAMMA,RS,VMCAK,DGM,DGTD

```

```

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

```

```

5 MFRUN=MF
MF=MFRUN*CAHT*50.
CALL ATMOS
RHO5=RHOZ2*SIG*5
VMCG = VMCGK*KTOP
VMCA = VMCAK*KTOP
APV=APR
COFF = 0.00
ISP=0
IDPV=0
JED = 0
JDFV = 0
JBRK = 0
IALMX = 0
WRITE (PRINT,1000)

```

```

1000 FORMAT(1H1,3X,40OUTPUT DEFINITIONS - LANDING,/,10X,0SEGMENT = *,
101 DETERMINATION OF ANGLE OF ATTACK AT TOUCHDOWN,/,10X,
202 R/C AVAILABLE AT TOUCHDOWN - 1 ENGINE OUT,/,10X,
303 DETERMINATION OF ANGLE OF ATTACK AT THE OBSTACLE,/,10X,
404 R/C AVAILABLE AT THE OBSTACLE - 1 ENGINE OUT,/,10X,
505 INTEGRATION FROM OBSTACLE TO TOUCHDOWN,/,10X,
606 TOUCHDOWN TO NOSEDOWN INTEGRATION,/,10X,
707 CONDITIONS AT NOSEDOWN,/,10X,
808 NOSEDOWN TO STOP INTEGRATION,/,10X,
909 CONDITIONS AT STOP,/,
LINE = LINIT,
GO TO 50

```

```

C     OUTPUT BLOCK

```

```

20 IF (ROTP,NE,0.) GO TO 23
   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
   PCM=60.*VTF*SIN(GAMR)
   THETD=THETR*RTOD
   THDTD=G*RTOD
   ALFAD=ALFAR*RTOD
   ALTHD=ALTHR*RTOD
   WRITE (PRINT,1002) ISEG,HF,DFLP,WT,VTK,PCM,THETD,AN,CL,LIFT,TGROS,
1TIMS,XE,ZE,GAMD,ALFAD,THDTD,AT,CD,DRAG,ALTHD
   LINE=LINE+3
1002 FORMAT (1H0,111,6F12.3,2F12.5,2F12.3,/,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.
   NFNG=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,YS AV,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.
   NFNG=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
ZF=0.0
Q=0.0
INTEG=0
MF=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.(.AND.VTF*SIN(GAMR).GE.-RS) GO TO 100
GO TO 400
```

```
120 ISEF = 1
    GAMTR = GAMR
    GO TO 20
```

C DETERMINE R/C AVAILABLE AT TOUCHDOWN

```
125 THR TL = 1.0
    INDEX = 3
    ENGNO = ENGNO-1.0
    NENG=2
    CALL AEGFM
    ENGNO = ENGNO+1.0
    NENG=0
    THETR = ALFAR-WINCR+GAMR
    ISEF = 2
    GO TO 20
```

C DETERMINE ANGLE OF ATTACK AT THE OBSTACLE

```
150 HF=HFRUN+C*HT+HFOB
    ZE = HFOB
    GAMR = GAMTR
    CALL ATMOS
    RHOS=RHOZ2*SIG*S
    OS=RHOS*VTF*VTF
    MACH=VTF/SOUND
    GN=1.0
    GT=0.0
    INDEX = 6
    ITRM = 1
    COFF=0.0
    CALL AEGFM
    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 THR TL = 1.0
    INDEX = 3
    FNGNO = ENGNO-1.0
    NFNG=2
    CALL AEGFM
    ENGNO = ENGNO+1.0
    NENG=0
    THETR = ALFAR-WINCR+GAMR
    ISEF = 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)
   TMS=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 AEGFM
   THETR=ALFAR-WINCR+GAMR
   ISEG = 5
   GO TO 20

```

C INTEGRATION FROM TOUCHDOWN TO NOSEDOWN

```

250 TMS=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
   TMS=TMS+DT
270 TIME2=TMS
   DIST2=XE
   ISEG = 7
   INDEX=1
   QS=RHOS*VTF*VTF
   MACH=VTF/SOUND
   CALL AEGFM
   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
    XEDOT=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
    ISFG = 8
    GO TO 20

330 IF (XEDOT-VWF) 335,340,305
335 DT = (VWF-XEDOT)/(VTF1-XEDOT)*DTIME
    XF = 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 AEOFM
    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
15K

```

```
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,14HHAIR 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 SUPROUTINE 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, INTF2	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
	* DRDZ, DB2DX, DB2DY, DB2DZ	LIST9
C	THIS EQUIVALENCE STATEMENT IS NOT USED IN THE MILSTOL PROGRAM	
C	EQUIVALENCE (L, I), (K, M)	LOOK
	DVDX(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	LOOK
10	CONTINUE	7APR2
	I1 = NX-1	LOOK
	IF = NX	LOOK
	LINX = 1	LOOK
	GO TO 25	LOOK
11	IF(I-2)120, 12, 15	LOOK
120	I1 = 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	I1 = I-2	LOOK
	IF = I+1	LOOK
	GO TO 25	LOOK
20	IF(NDER(1), NE, 0)GO TO 11	LOOK
	I1 = 1	LOOK
	IF = 1	LOOK
25	GO TO(28, 26, 28, 26), ND	LOOK
26	J1 = 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
J1 = NY-1	LOOK
JF = NY	LOOK
L1NY = 1	LOOK
GO TO 45	LOOK
31 IF(J-2)315,32,35	LOOK
315 J1 = 1	LOOK
JF = 2	LOOK
L1NY = 1	LOOK
GO TO 45	LOOK
32 J = 3	LOOK
35 IF(J-NY)37,36,36	LOOK
36 J = NY-1	LOOK
37 J1 = J-2	LOOK
JF = J+1	LOOK
GO TO 45	LOOK
40 IF(INDR(2),NE,0)GO TO 31	LOOK
J1 = J	LOOK
JF = J	LOOK
45 DO 50 K = 1,NZ	LOOK
IF(ZA-ZC(K))51,60,50	7APR2
50 CONTINUE	LOOK
K1 = NZ-1	LOOK
KF = NZ	LOOK
L1NZ = 1	LOOK
GO TO 65	LOOK
51 IF(K-2)515,52,55	LOOK
515 K1 = 1	LOOK
KF = 2	LOOK
L1NZ = 1	LOOK
GO TO 65	LOOK
52 K = 3	LOOK
55 IF(K-NZ)57,56,56	LOOK
56 K = NZ-1	LOOK
57 K1 = K-2	LOOK
KF = K+1	LOOK
IF(1FLAG)65,65,575	LOOK
575 IF(KF-(1FLAG+1))65,58,59	LOOK
58 K1 = 1FLAG-3	LOOK
KF = 1FLAG	LOOK
GO TO 65	LOOK
59 IF(K1-1FLAG)595,65,65	LOOK
595 K1 = 1FLAG	LOOK
KF = 1FLAG+3	LOOK
GO TO 65	LOOK
60 IF(INDR(3),NE,0)GO TO 51	LOOK
K1 = K	LOOK
KF = K	LOOK
65 XL(1) = XA	7APR2
XN(1) = YA	7APR2
XN(2) = ZA	7APR2
IF(INDR(3),EQ,0)GO TO 200	LOOK
IPATH = 1	LOOK
GO TO 300	LOOK
200 IF(INDR(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	I7 = 1	LOOK
	DO 1100 M=K1,KF	LOOK
	I2 = I2+1	LOOK
310	I1 = 1	LOOK
	DO 1200 L=11,IF	LOOK
	I1 = I1+1	LOOK
320	IY = 1	LOOK
	DO 1300 JJ = J1,JF	LOOK
	IY = IY+1	LOOK
	IF(IPATH.LT.4)GO TO 330	LOOK
	IF(MOD(ND.2).NE.C)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)	
	I2 = 1	LOOK
	DO 1400 K=K1,KF	LOOK
	I2 = I2+1	LOOK
	IF(IPATH.EQ.7) GO TO IWHIT.(300)	
	I1 = 1	LOOK
	DO 1500 I=11,IF	LOOK
	I1 = I1+1	LOOK
350	IJK = I+NX*(J-1)+NY*(K-1)	LOOK
	GO TO(360,370,380,380).IPATH	LOOK
360	YNI(IY) = B(IJK)	LOOK
	IF(ND.LE.2)YNI(IY) = B2(IJK)	LOOK
	GO TO 430	LOOK
370	YNI(I2) = B(IJK)	LOOK
	IF(ND.LE.2)YNI(I2) = B2(IJK)	LOOK
	GO TO 405	LOOK
380	YI(I1) = B(IJK)	LOOK
	IF(ND.LE.2)YI(I1) = B2(IJK)	LOOK
1500	XL(I1) = XC(I1)	LOOK
	IF(LINX.EQ.0)GO TO 388	LOOK
	YNI(I2) = ALIN(2,XL,YL,XA)	7APR2
	IF(ND.LE.2)YNI(I2) = ALIN(2,XL,YLL,XA)	7APR2
	GO TO 405	LOOK
388	IF(11.NE.IF)GO TO 400	LOOK
	YNI(I2) = YL(I1)	LOOK
	YNI(I2) = YLL(I1)	LOOK
	GO TO 405	LOOK
400	IND = 0	LOOK
	CALL LAGRA(XL,YL,IND)	LOOK
	YNI(I2) = YL(I1)	LOOK
	IF(ND.GT.2)GO TO 405	LOOK
	CALL LAGRA(XL,YLL,IND)	LOOK
	YNI(I2) = YLL(I1)	LOOK
405	XNI(I2) = ZC(K)	LOOK
1400	CONTINUE	LOOK
	IF(LINZ.EQ.0)GO TO 408	LOOK
	YNI(IY) = ALIN(2,XN,YN,ZA)	7APR2
	IF(ND.LE.2)YNI(IY) = ALIN(2,XN,YNI,ZA)	7APR2
	GO TO 430	LOOK

408	IF(KI,NE,KF)GO TO 420	LOOK
	YM(IY) = YN(IZ)	LOOK
	IF(IND.LE.2)YMM(IY) = YNW(IZ)	LOOK
	GO TO 430	LOOK
420	IND = 0	LOOK
	CALL LAGRA(XN,YN,IND)	LOOK
	YM(IY) = YN(I)	LOOK
	IF(IND.GT.2)GO TO 430	LOOK
	CALL LAGRA(XN,YN,IND)	LOOK
	YMM(IY) = YNW(I)	LOOK
430	XN(IY) = YC(J)	LOOK
	IF(IIPATH.EQ.4)GO TO(1300,100,1300,98).ND	LOOK
1300	CONTINUE	LOOK
	IF(LINV.EQ.0)GO TO 438	LOOK
	ANS = ALIN(2,XM,YM,YA)	7APR2
	IF(IIPATH.EQ.3)DBDY = SL(2,XM,YM)	LOOK
	IF(IND.GT.2)GO TO 460	LOOK
	ANS2 = ALIN(2,XM,YMM,YA)	7APR2
	IF(IIPATH.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(IND.LE.2)ANS2 = YMM(IY)	LOOK
	GO TO 460	LOOK
450	CALL INTP2(5,XM,YM,YA,ANS,C1,C2,C3,C4)	7APR2
	IF(IIPATH.EQ.3)DBDY = DWDX(YA)	7APR2
	IF(IND.GT.2)GO TO 460	LOOK
	CALL INTP2(5,XM,YMM,YA,ANS2,C1,C2,C3,C4)	7APR2
	IF(IIPATH.EQ.3)DB2DY = DWDX(YA)	7APR2
460	IF(IIPATH-3)490,221,480	LOOK
480	RETURN	LOOK
490	YL(IX) = ANS	LOOK
	IF(IND.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(IIPATH.EQ.2)DBDX = SL(2,XL,YL)	LOOK
	IF(IND.GT.2)GO TO 522	LOOK
	ANS2 = ALIN(2,XL,YLL,XA)	7APR2
	IF(IIPATH.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(IND.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(IIPATH.EQ.2)DBDX = DWDX(XA)	7APR2
	IF(IND.GT.2)GO TO 522	LOOK
	CALL INTP2(5,XL,YLL,XA,ANS2,C1,C2,C3,C4)	7APR2
	IF(IIPATH.EQ.2)DB2DX = DWDX(XA)	7APR2
522	IF(IIPATH.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(IND.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
	DPDZ = DWDX(ZA)	LOOK
	IF(ND.GT.2)GO TO 200	LOOK
	CALL INTP2(5,XM,YMM,ZA,ANS2,C1,C2,C3,C4)	7APR2
	DPDZ = 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
118	IY = IY-1	7APR2
	GO TO 116	7APR2
	END	LOOK

C	SUBROUTINE NRP2(OX,YIN,XIN,YSAV,XSAV)	AA
	6400 FORTRAN SUBROUTINE TO DO A NEWTON RAPHSON LINEAR CONVERGENCE.	AA
	IF (OX) 10,20,10	AA
10	YSAV=YIN	AA
	XSAV=XIN	AA
	XIN=XIN-OX	AA
	OX=0.0	AA
	GO TO 30	AA
20	Z=XIN	AA
	XIN=XIN-YIN*(XIN-XSAV)/(YIN-YSAV)	AA
	XSAV=X	AA
	YSAV=YIN	AA
30	RETURN	AA
	END	AA

```

SUBROUTINE SKIP
6400 FORTRAN PAGE EJECT SUBROUTINE.

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

DIMENSION HEAD(16),NAME(10)

COMMON /LIST1/ LINE,PAGE,LIMIT,CARD,PRINT,INP,IFUNCH,DATE,HD(60)
COMMON /LIST2/ DELTO,ALFAR,THRSL,GAMP,HIND,VTF,WT,HF,DT,TIMS,XF,FU
IEL,MACH,VWF,LIFT,DRAG,THRST,RCF,ENGNO,THV,IN
COMMON /LIST3/ CL,CD,S,OS,THR,ALTHR,CLMAX,AR,CLAR,CMACG,ALPHD,CX,
ICZ,O,IMOM,WINCR,CMOMP,OSC,ALMXR,VSF,BW
COMMON /LIST5/ SIG,SOUND,NU,TEMP,PAMB,IATM,TEMP,DSODH,DRHO

DATA (NAME(I),I=1,10)/1CHSTANDARD A,9HTMOSPHERE,1CHTROPIC TEM,9HPE
1RATURES,10M POLAR TEM,9HPERATURES,10M HOT TEM,9HPERATURES,10M C
2OLD TEM,9HPERATURES/
DATA (HD(I),I=1,59)/10M AIRSPEED,10M ALPHA,10M
1ALT ERROR,10M ALTITUDE,10M AXB,10M AZB,10M CL,
210M CD,10M DISTANCE,10M DRAG,10M ELEVATOR,10M FUEL U
3SPD,10M FLY. PATH,10M FUEL,10M FUEL FLOW,10M HEIGHT,10M H
400K HT.,10MHORIZONTAL,10M LIFT,10MLOAD FACT.,10M MACH NO.,10
5MNO9. ACCEL,10M OS,10M RANGE,10HR/C ACTUAL,10M SEGME
6T,10M SEP,10M SEP/WF,10M SP. RANGE,10MTAN. ACCEL,10M THR
70TTL,10M THETA,10M THETA DOT,10MTH DEL OPT,10M TIME,10M
8 WEIGHT,10M (DEGREES),10M (FEET),10M (F/S/S),10M (FT/LB),
910M (FT/MIN),10M (FT/SEC),10M (G),10M (KNOTS),10M (LB/
SHR),10M (MINUTES),10M (IN MI/LB),10M (IN MILES),10M (PERCENT),10M (
SPOUNDS),10M THRUST,10HR/C INSTAN,10M TEMP.,10M (DEG F),10
SH FLAP,10M (SECONDS),10MTHST ALPHA,10M (DEG/SEC)/

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
READ (CARD,130) (HEAD(I),I=1,16)
130 FORMAT (1X,A9,7A10/8A10)
RETURN

10 CALL SECOND(T)
WRITE (PRINT,140) (DATE,Y,PAGE
140 FORMAT (1H,4X,20M)AIRCRAFT PERFORMANCE,4X,A10,10X,P10,2,5X,5M)PAGE
1,13)
NENG=ENGNO
WRITE (PRINT,150) NENG,HEAD(1),HEAD(2),S
150 FORMAT (1H,25X,12* ENGINE#A9,A10* S * #PS,0* 50 FT *)
IF (IATM.EQ.0) GO TO 20
IF (IATM.EQ.0,AND,DT.NE.0.) GO TO 30
N=2*(IATM+1)
WRITE (PRINT,160) NAME(N-1),NAME(N)
160 FORMAT (1H,.85X,A10,49)

```

GO TO 40	AL
20 WRITE (PRINT,170) TEMP	AL
170 FORMAT (1H+.84X12HTEMPURE*.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(1),1=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	AL
60 LINE=4	AL
RETURN	AL
80 WRITE (PRINT,210)	AL
210 FORMAT (20X.*STOL TAKEOFF*)	AL
GO TO 120	AL
90 WRITE (PRINT,220)	AL
220 FORMAT (20X.*STOL LANDING*)	AL
120 LINE=5	AL
RETURN	AL
END	AL

FUNCTION SL(I,X,Y)

DIMENSION X(I),Y(I)

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

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)

    I1=NSOL
    GO TO 20
10  I1=I1
20  JJ=I1+1
    DO 60 I=1,I1
C   IF CN(I,JJ).EQ.0.0
C       IF I.GT.1 ADD EQ I TO EQ 1.
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,I1
        IF(CN(N,JJ).NE.0.0)GO TO 36
32  CONTINUE
        WRITE(6,1000)JJ
1000 FORMAT(10X#COLUMN,2X,12,2XB#ALL 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.

70  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,I1
65  CN(I,JJ) = CN(I,1)
    IL=I1-1
    IF (IL) GO,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,I1
80  CN(I,J)=CN(I,J)-CN(I2,J)
    GO TO 10
90  IF (NSOL-I) 100,120,100
C   CALCULATE REMAINING SOLUTIONS. STORE IN FIRST COLUMN.

100 DO 110 I=2,NSOL
    J1=I+1
    CN(I,1)=CN(I,J1)
    J2=I-1
    DO 110 J=1,J2
110  CN(I,1)=CN(I,1)-CN(I2,J1)+CN(I,J1)
120  GO TO 50
    END

```

SUBROUTINE TAKEOFF

C 6000 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,THRTHL,GAMR,HIND,VTF,WT,HF,DT,TIMS,XF,FU
 1FL,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.,1C/
 DATA (NL(I),I=1,8)/10H0556370226,10H3323082052,10H3938514542,10H38
 140015151,10H3610171403,10H3431091158,10H5739392838,10H5940015138/

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 SUPPRESSES 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      DGLQ=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,1TIMR,TIMB,DGLQ

```

```

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

```

```

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

```

PRE

```

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 150

```

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
PCM=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,THETO,AN,CL,LIFT,TGROS,T
1IMS,XE,ZE,GAMD,ALFAD,THOTO,AT,CD,DRAG,ALTHD
LINE=L:NE+3
912 FORMAT (1H0,111,6F12.3,2F12.5,2F12.3/12X,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
ENGN0 = ENGN0-1.
NFNG=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
XF=0.0
ZF=0.0
Q = 0.0
IFRR=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 AEOFM
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 AEOFM
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*5
TJMS=0.0
XF=0.0
ZF=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 XF=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 AEGFM
GO TO 20

```

C INTEGRATION FROM VR TO VF

```

670 SFG3=0.00
TIME3=0.00
TIMS=0.00
INT=1
DTIME=-0.2
XF=0.00
ZF=0.00
THETR=0.0
XFROT=VRF
ZFDOT=0.00
ILOOP=1
VTF=VRF
675 VTF1=VTF
CALL GINTG

```

ISFG=6
GO TO 20

C CONDITIONS AFTER VR TO VF INTEGRATION

680 IF(VTF-VIF) 685,690,675
685 DT = DTIME*((VTF-VIF)/(VTF-VTF1))
TIMS = TIMS-DT
XF = XE-((VTF+VIF)*0.5-VWF)*DT
VTF=VIF
690 SEG3=-XE
TIME3=-TIMS
V1=VIF*FTOK
ISEG=7
INDEX=1
QS=RHOS*VTF*VTF
MACH=VTF/SOUND
CALL AFOFM
GO TO 20

C INTEGRATION FROM VF TO STOP

700 TIMS=0.0
INT=1
SFG5=0.0
DTIME=0.5
TIMES=0.00
XF=0.0
ZF=0.0
THFTR=0.0
ZFNOT=0.0
O=0.0
XEDOT=VIF
ILOOP=1
VTF=VIF
710 VTF1=VTF
IF(TIMES+0.1*DTIME.GE.TIMR) THRTL=0.0
IF(TIMES+0.1*DTIME.LT.TIMR+TIMB) GO TO 720
COFF=BCOEF
IRFV=1
NENG = 2
ISP=1
720 CALL GINTG
ISFG=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
XF = XE+(0.5*(VWF+XEDOT)-VWF)*DT
TIMS = TIMS-DT
740 VTF=VWF
ISFG=9
INDEX=1
QS=RHOS*VTF*VTF
MACH=VTF/SOUND
SFG5=XE
TIMES=TIMS
CALL AEOFM
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 (IHO,10X, *FIELD NOT BALANCED*,5X, *NEW V1 SPEED = *,F7.2,
    * KTS*)
    GO TO 190
795 WRITE (PRINT,916) IBAL
916 FORMAT (IHO,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
    TMS=0.0
    XF=0.0
    ZF=0.0
    THFTR=0.0
    THRTL=1.0C
    ENGNO = ENGNO+1.
    NENG=0
    COEF = RCOEF
    ISP=0
    IRFV=0
    XFNOT=VWF
    ZFNOT=0.0
    O=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=-DT/ME*(VTF-V1F)/(VTF-VTF1)
```

```

      XE=XE+((VTF+VIF)*.5-VWF)*DT
      TMS=TMS+DT
      VTF=VIF
840  SEG1=XE
      TIME1=TIME
      VTK1=VIF*FTOK
      ISFG=11
      INDEX=1
      QS=RHOS*VTF*VTF
      MACH=VTF/SOUND
      CALL AEOFM
      GO TO 20

```

C OUTPUT SECTION FOR DEFAULT NOTES

```

900  ALFLO = ALFLO * RTOD
      WRITE (PRINT,910) ALFLO
910  FORMAT(1H0,10X,*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
      VIK=VIF*FTOK
      WRITE(PRINT,955) VIK,VRK,VLOK,VSK
955  FORMAT(1H0,10X,*ENGINE FAILURE SPEED **F7.2,* KTS*,/,11X,*ROTATIO
IN 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 (IF 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
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((N-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(!REV.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 (FIX(ENGNO/2.+0.0001)))+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

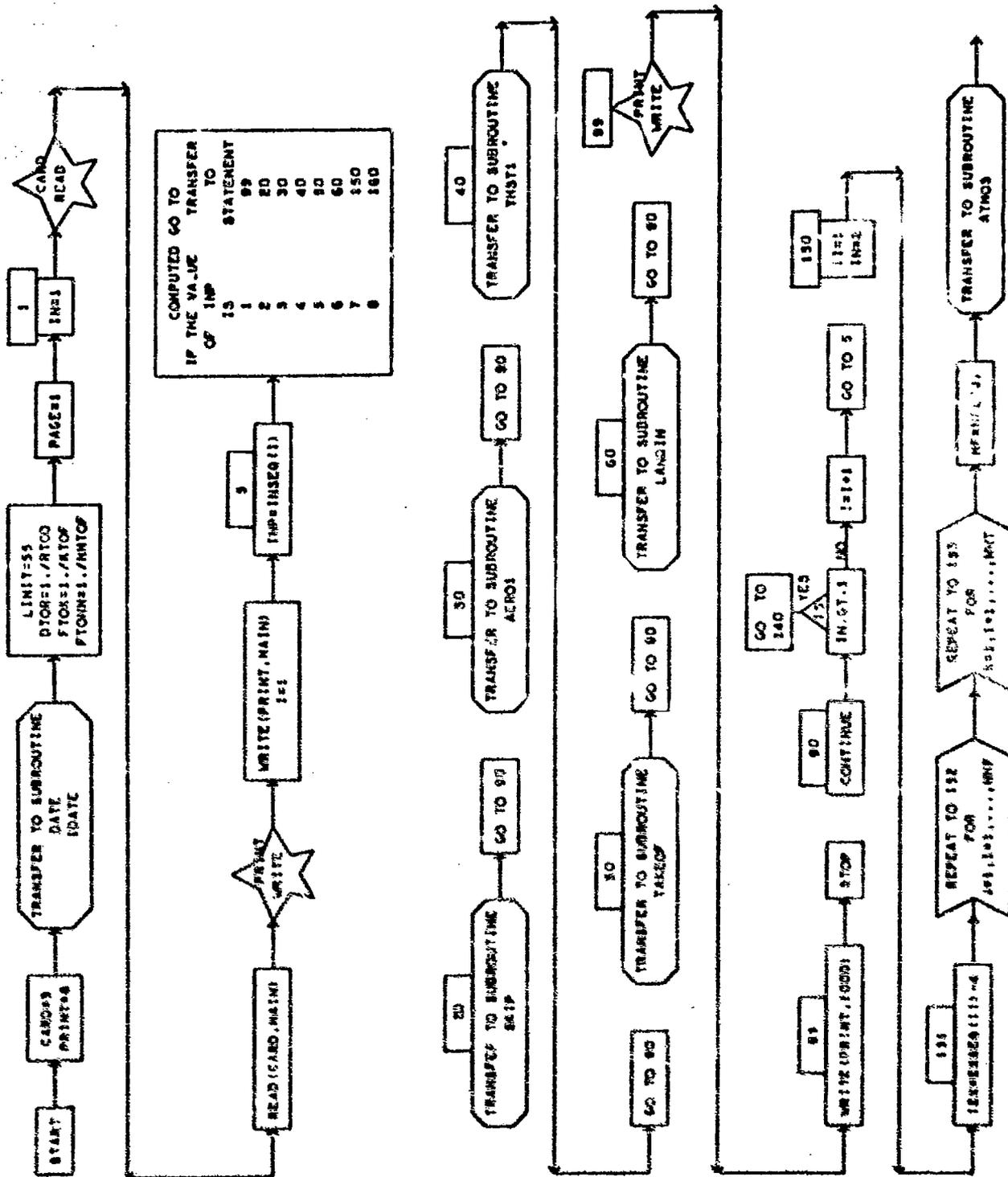
```


APPENDIX II
PROGRAM AND SUBROUTINE FLOW CHARTS

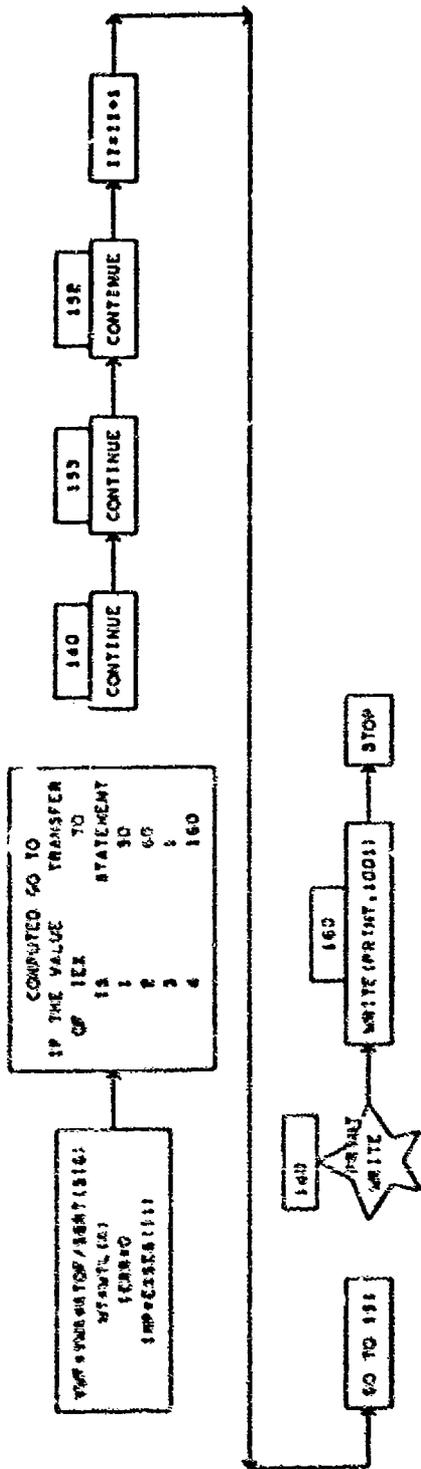
The following flow charts are contained in this appendix.

<u>Title</u>	<u>Description</u>	<u>Page</u>
MILSTOL	Main Program	II-1
MILSTOL	Overlay Dump Program	II-3
AEQFM	Equations of Motion Subroutine	II-4
AERO1	Aerodynamic Data Subroutine	II-8
ALIN	Linear Equation Function	II-10
ATMOS	Atmospheric Properties Subroutine	II-11
FIND	1-Dimensional Table Lookup Subroutine	II-13
GILL	Integration Subroutine	II-14
GINTG	Integration Driver Subroutine	II-15
HEAD	Page Heading Subroutine	II-17
INTP2	Curve Fitting Subroutine	II-18
KABD	Hyperbolic Curve Fit Solution Subroutine	II-19
LAGRA	Lagranian Interpolation Subroutine	II-20
LANDING	Landing Trajectory Driver Subroutine	II-21
LOOK	3-Dimensional Table Lookup Subroutine	II-23
NWRP2	Newton-Wrapson Iteration Subroutine	II-37
SKIP	Page Eject Subroutine	II-38
SL	Linear Slope Function	II-40
SMLT2	Simultaneous Equation Solution Subroutine	II-41
TAKEOFF	Takeoff Trajectory Driver Subroutine	II-43
THST1	Propulsion Data Subroutine	II-51
TRIM	Aircraft Trimming Subroutine	II-53

PROGRAM HILSTOL (INPUT, OUTPUT, TAPES=INPUT, TAPES=OUTPUT)



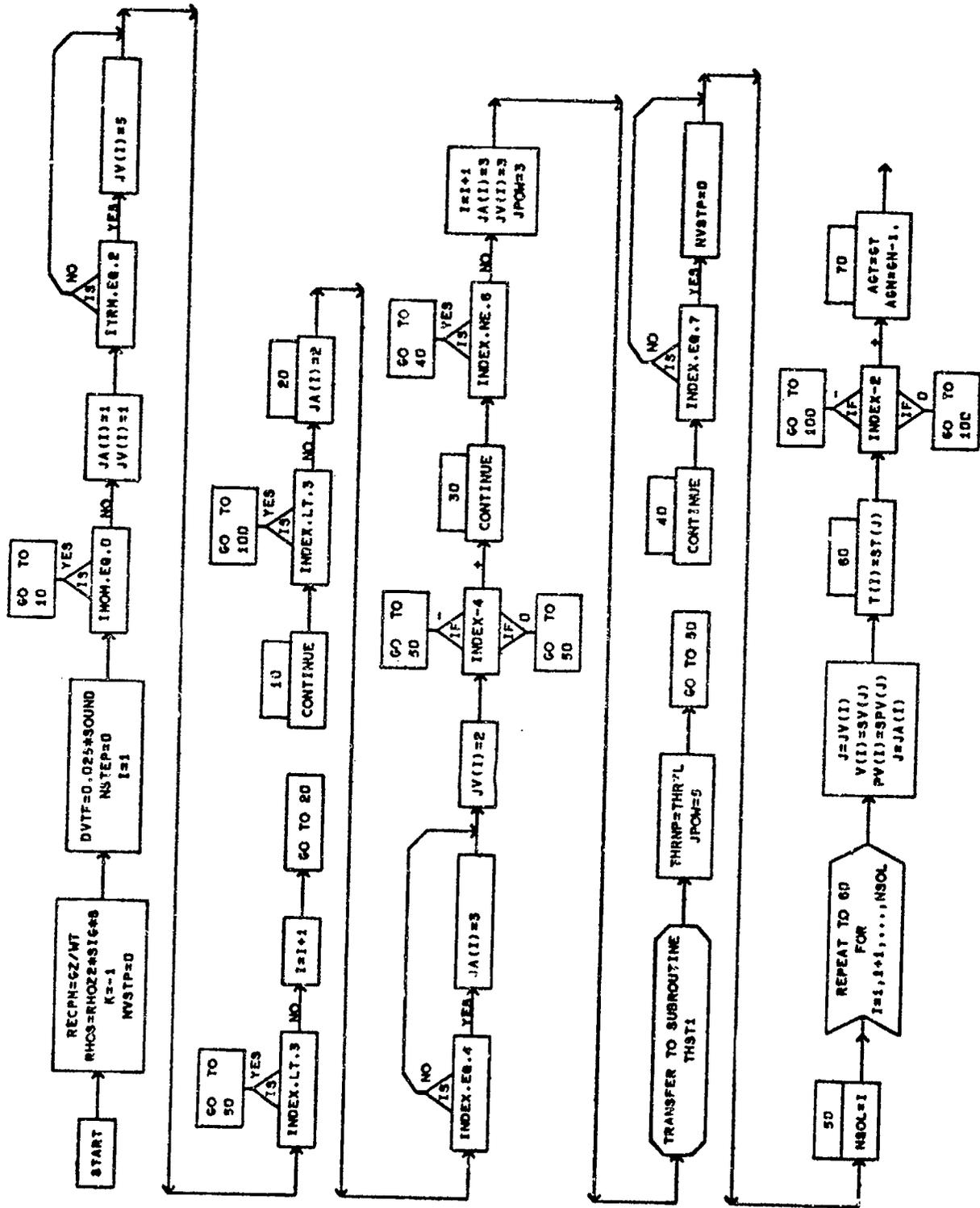
COMPUTED GO TO IF THE VALUE OF IMP	TRANSFER TO STATEMENT
13	99
1	20
2	30
3	40
4	80
5	60
6	50
7	100
8	100

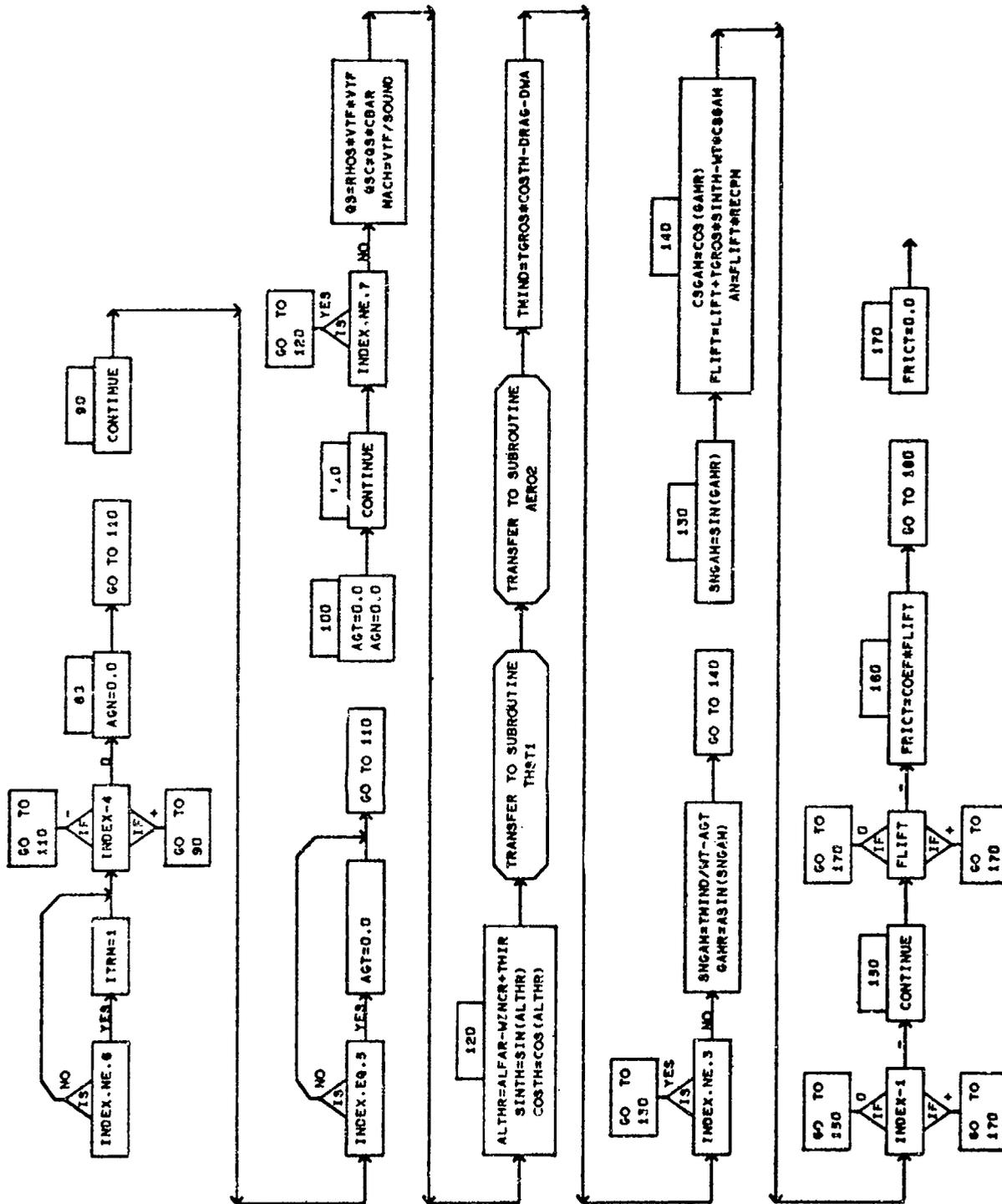


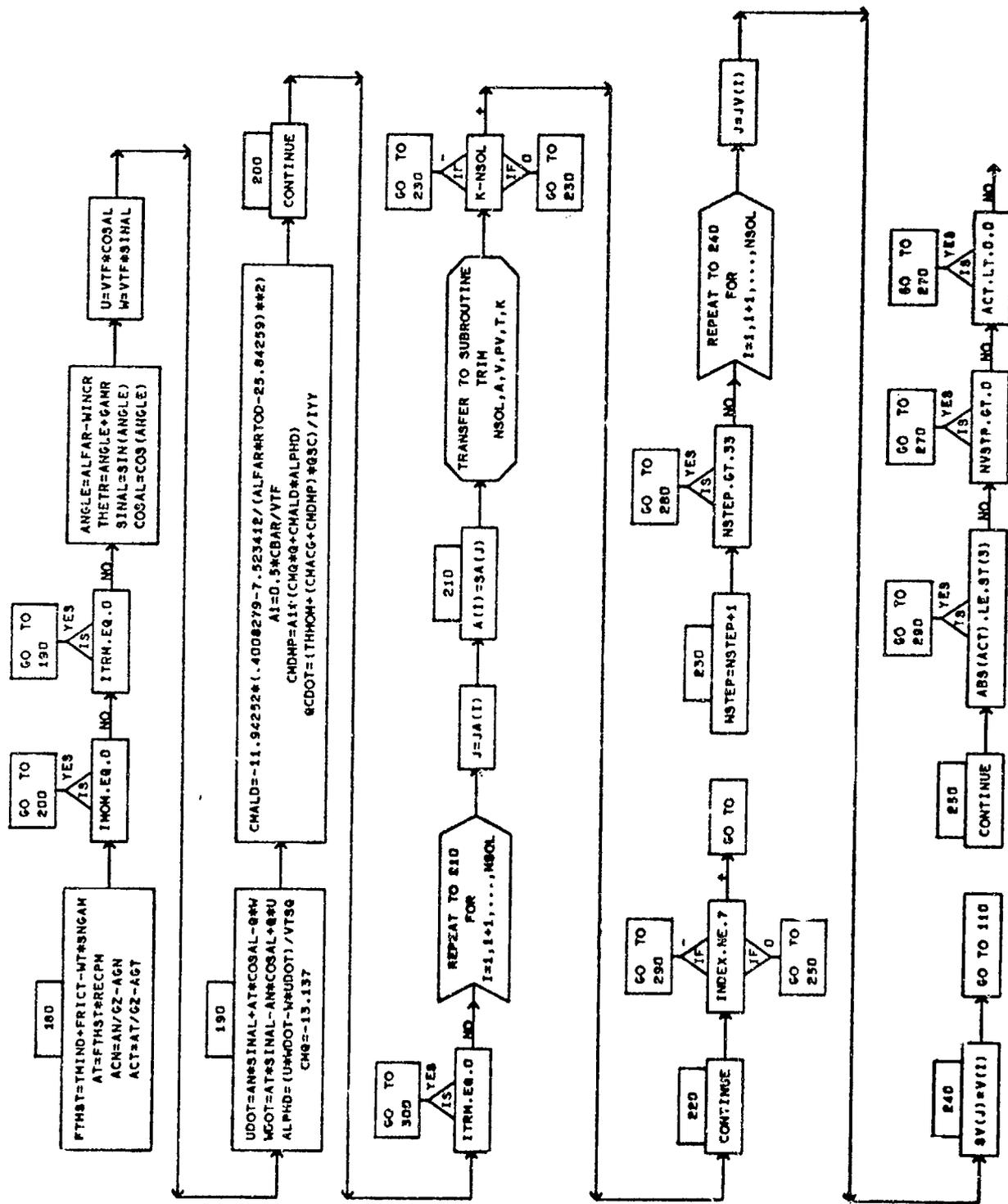
PROGRAM MILSTOL (INPUT, OUTPUT, TAPES=INPUT, TAPES=OUTPUT)

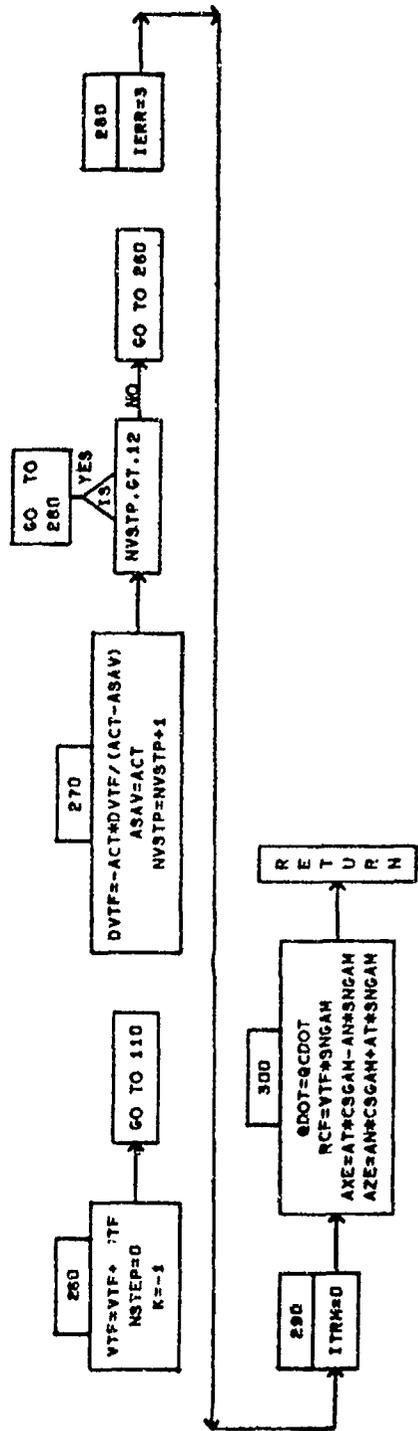


SUBROUTINE AERFN

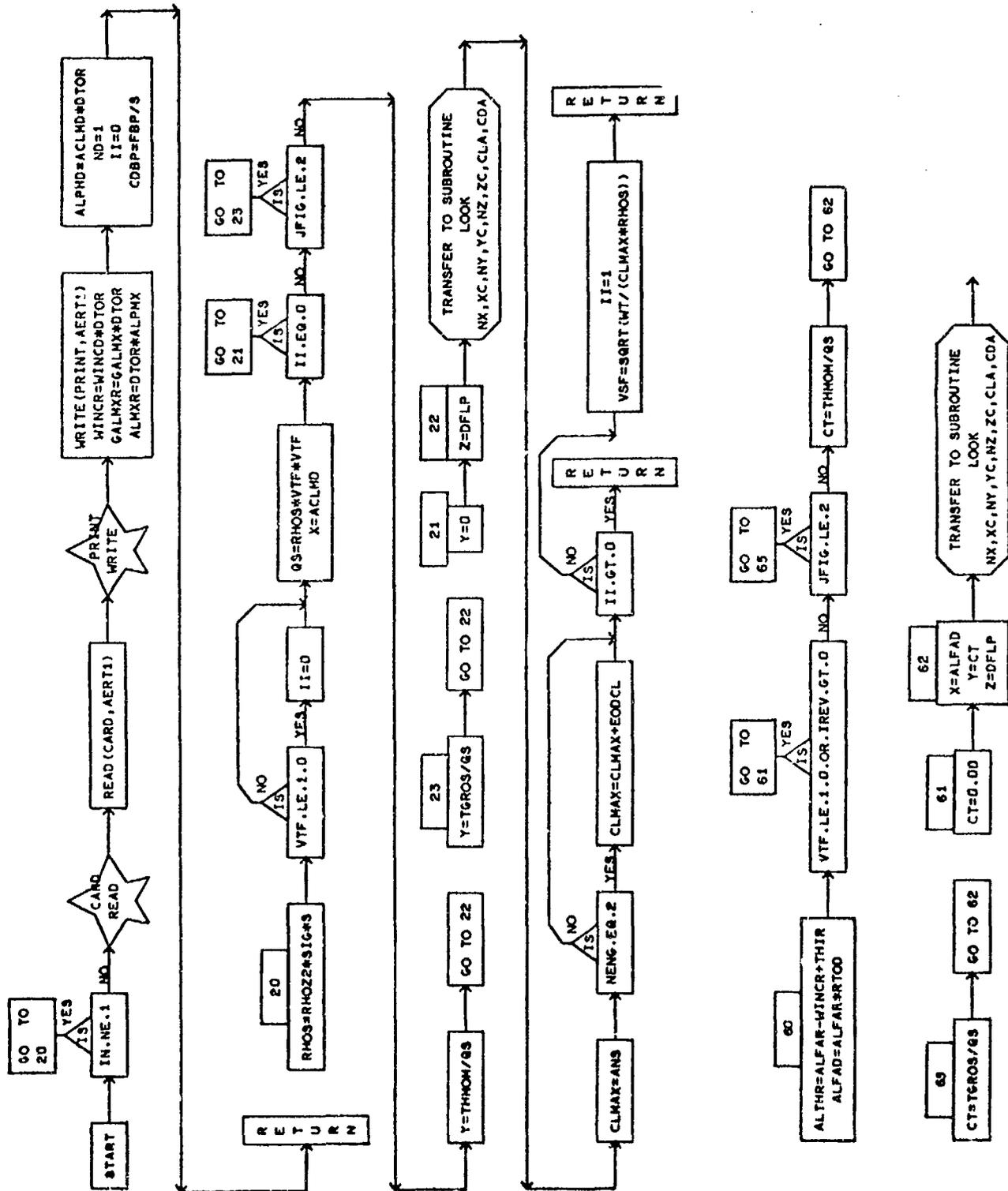


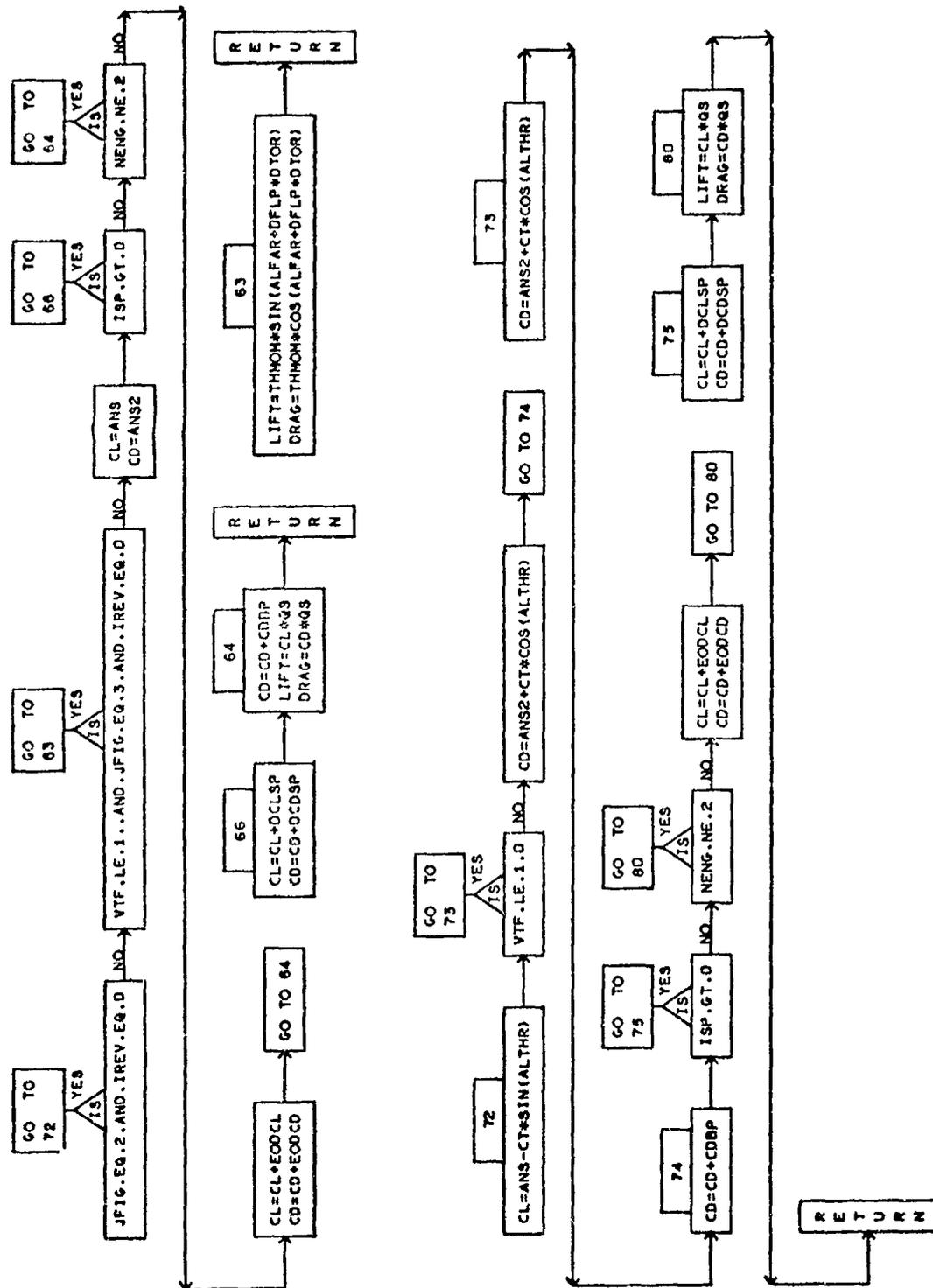




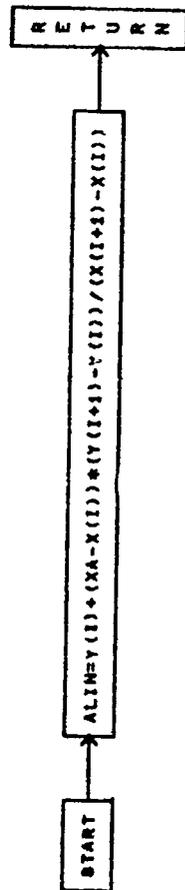


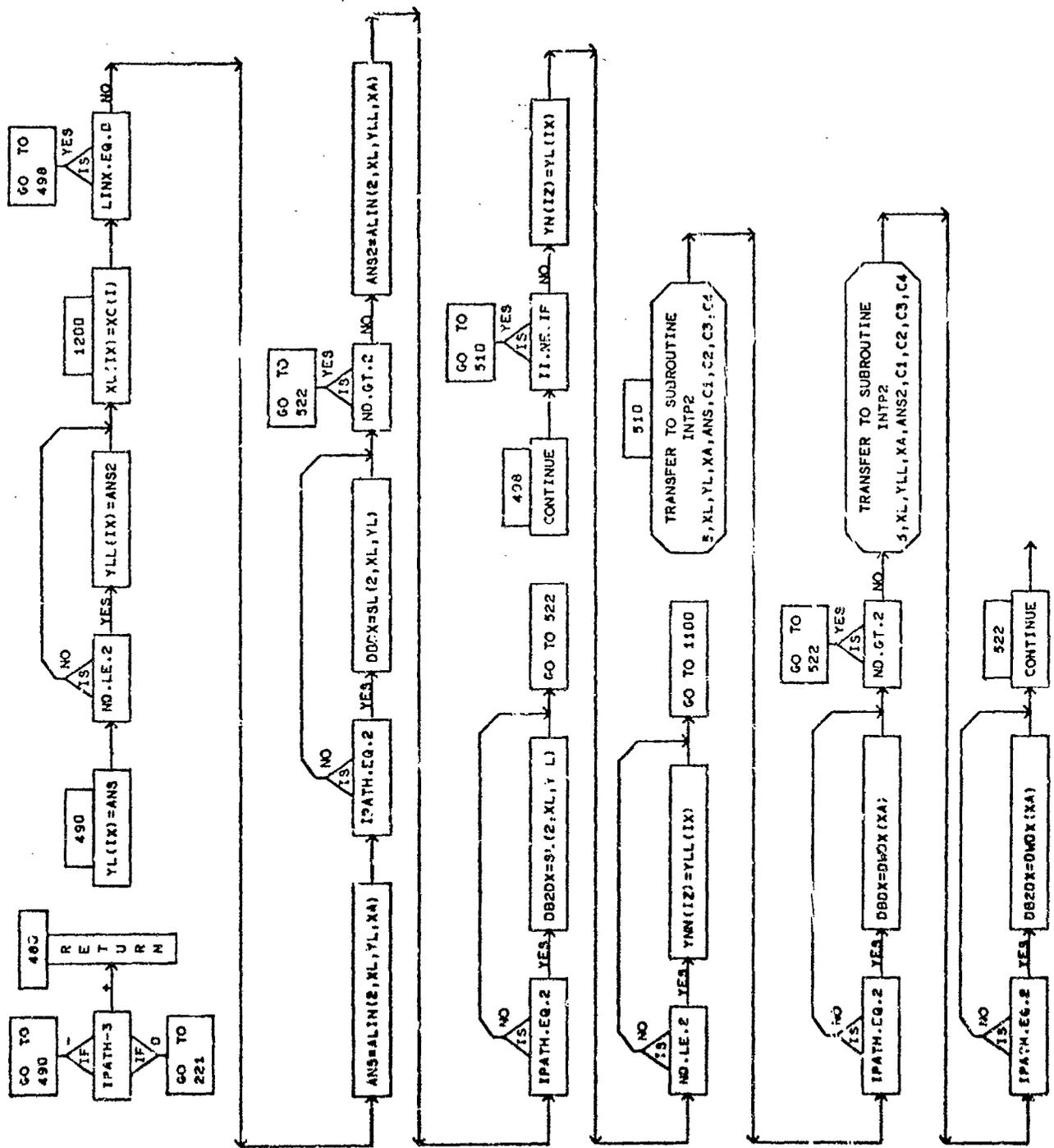
ROUTINE AERO1

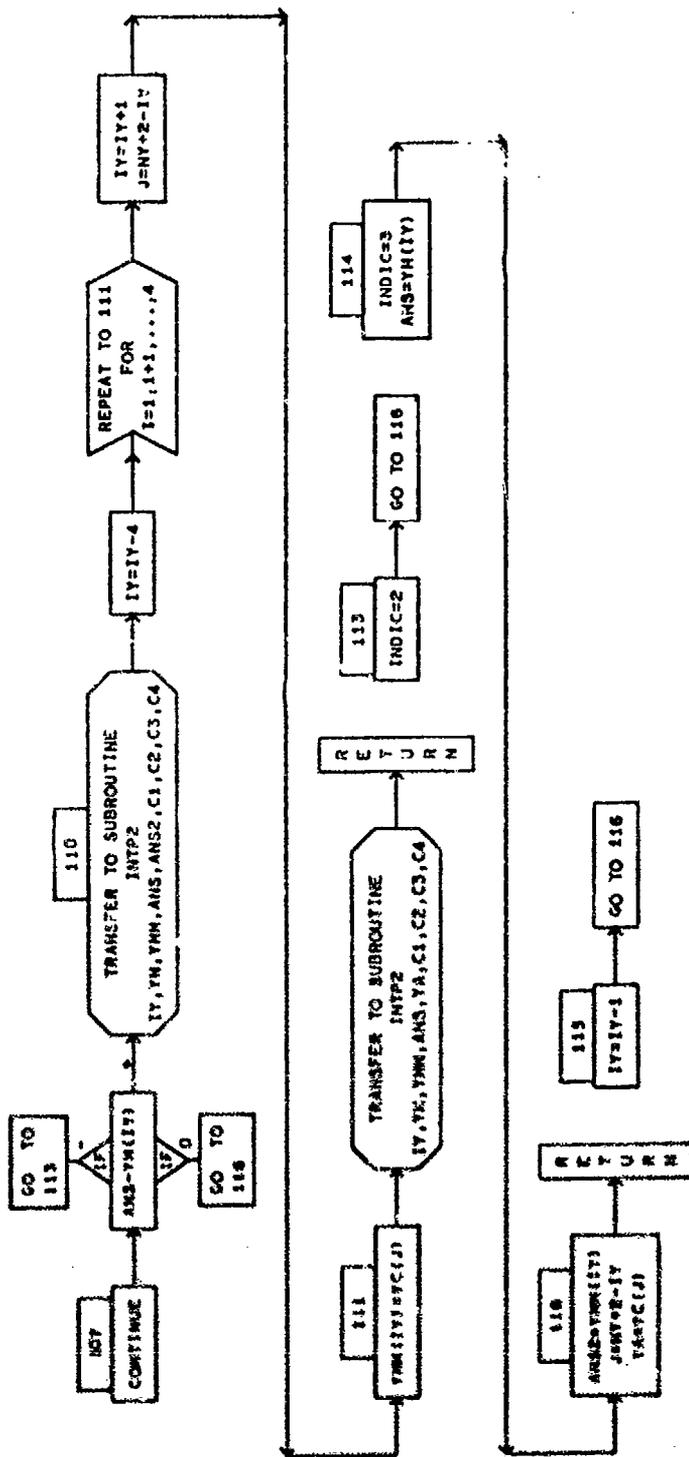




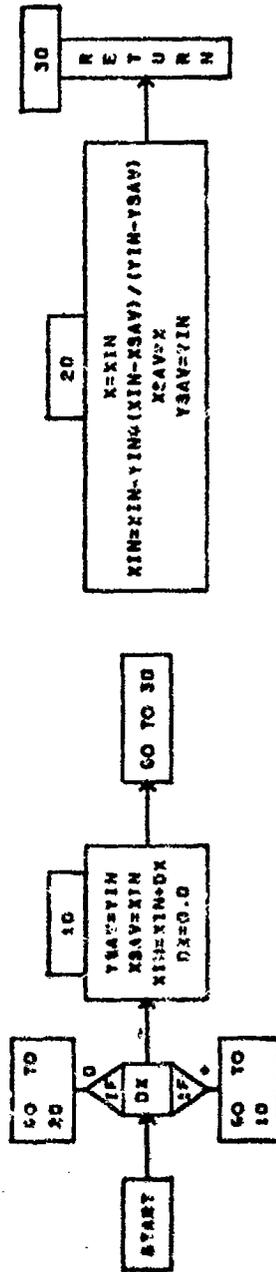
FUNCTIONALIN(I, X, Y, XA)

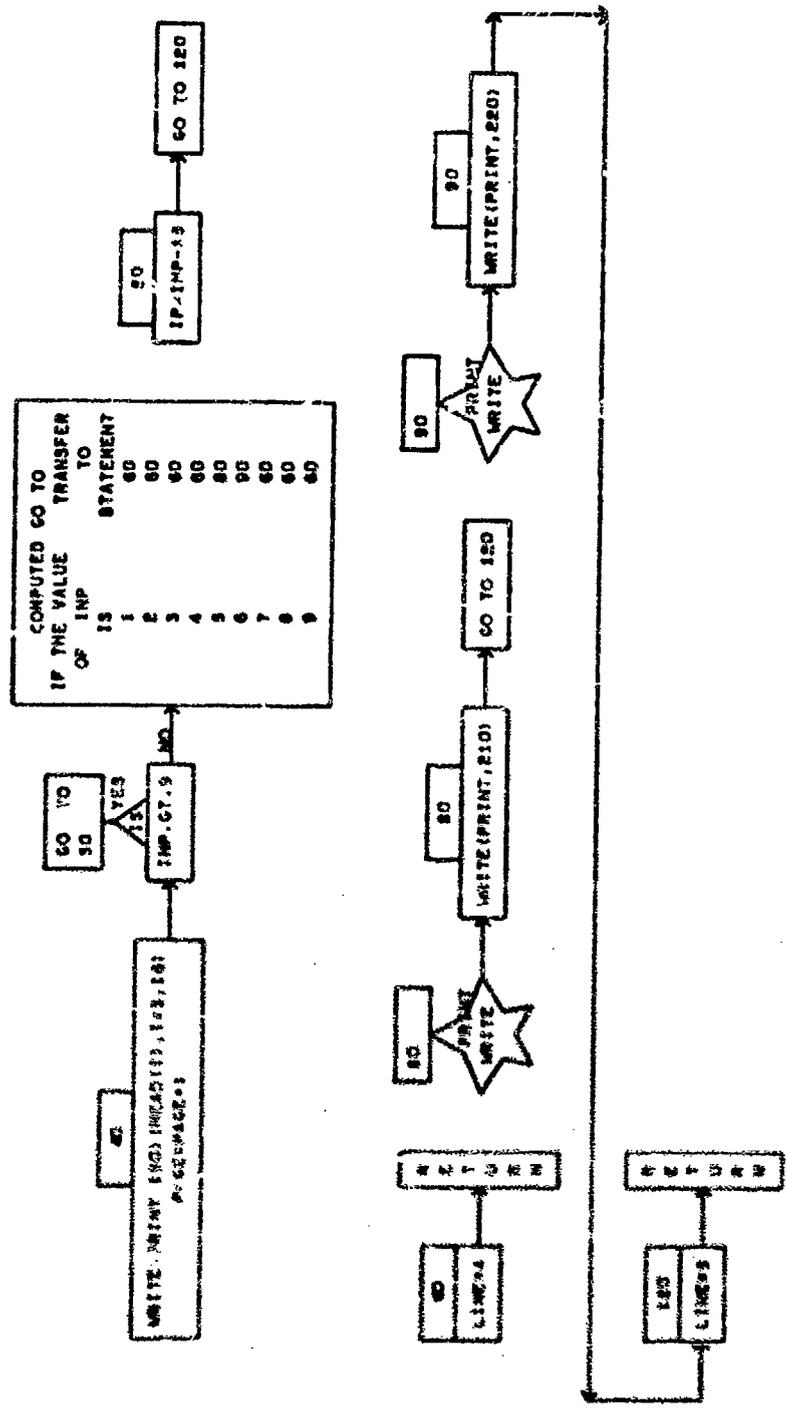




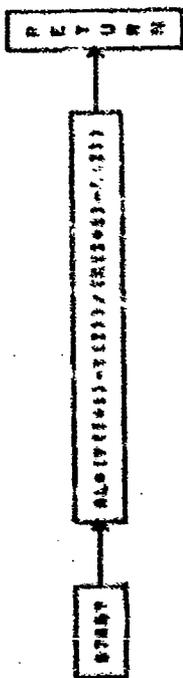


SUBROUTINE MVRPE (DX, YIN, XIN, YSAV, XSAV)

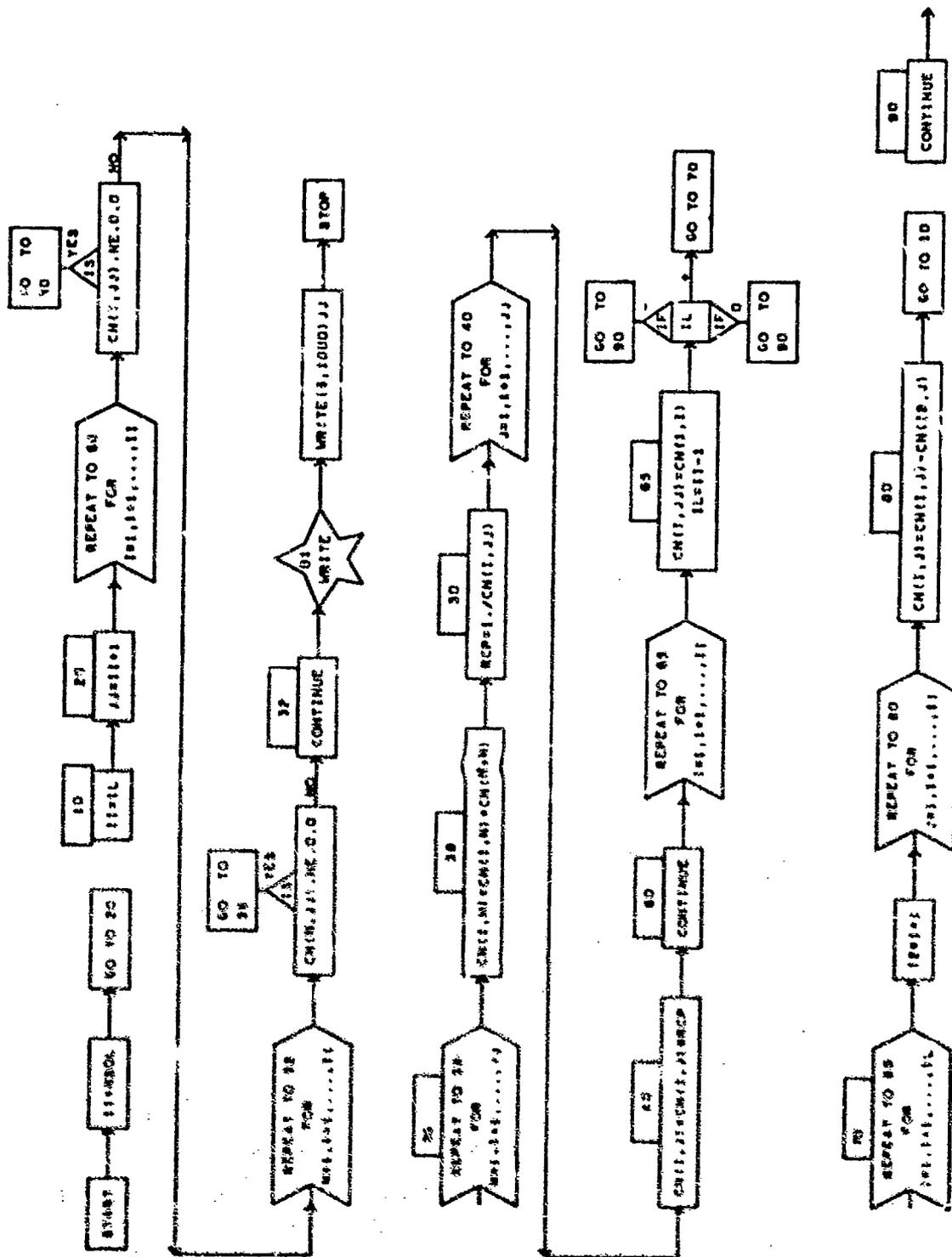




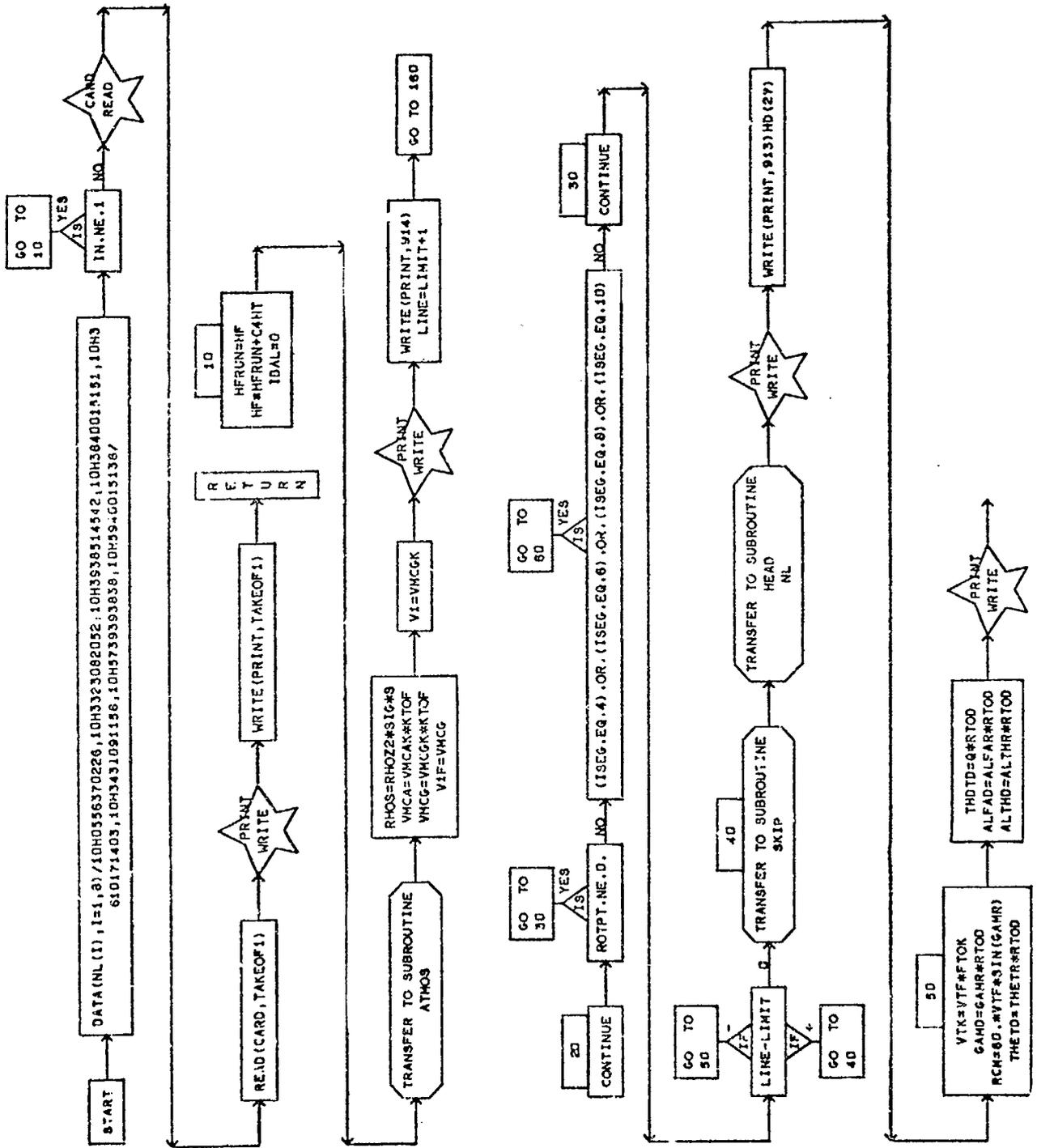
FUNCTION S(L1,N,Y)

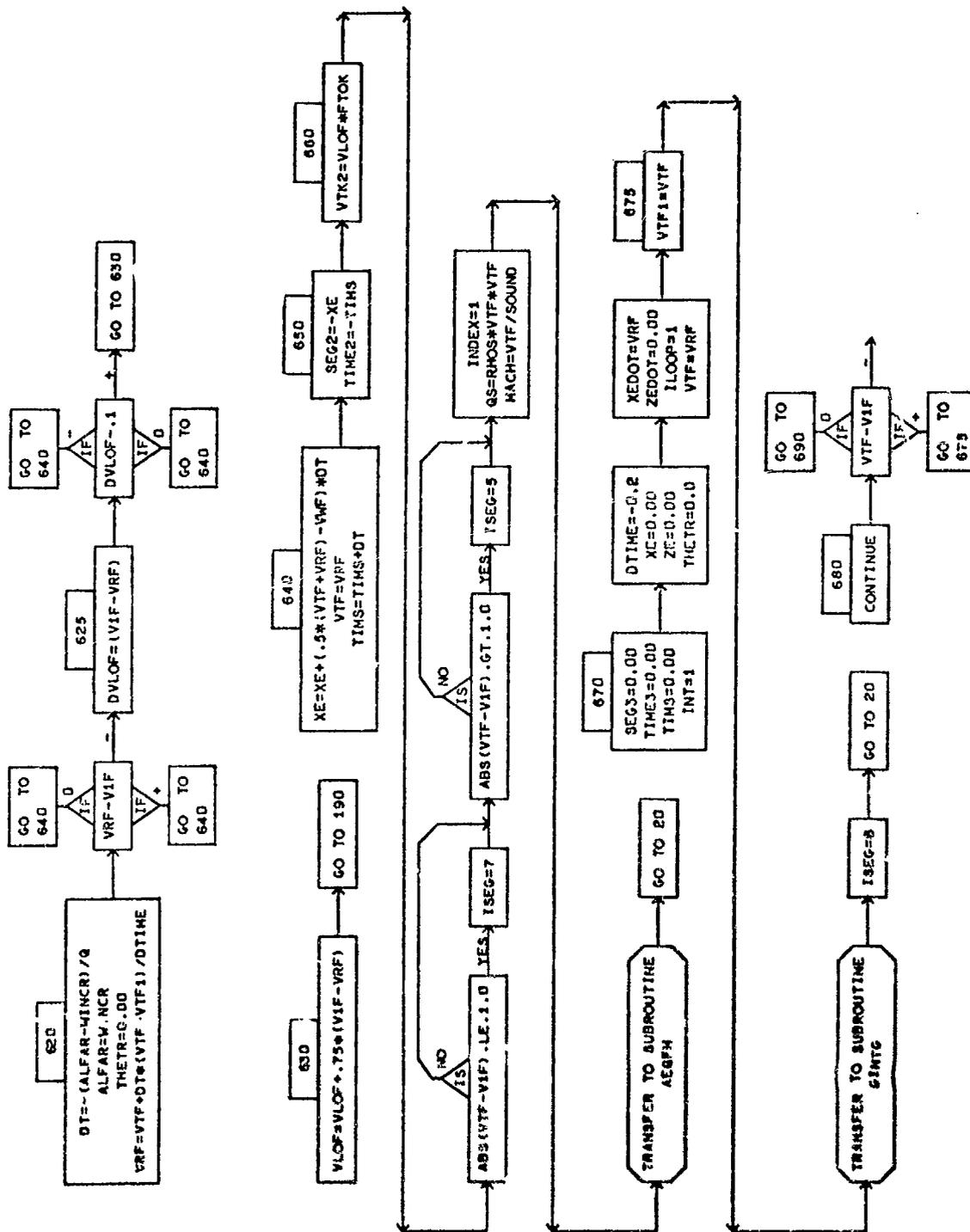


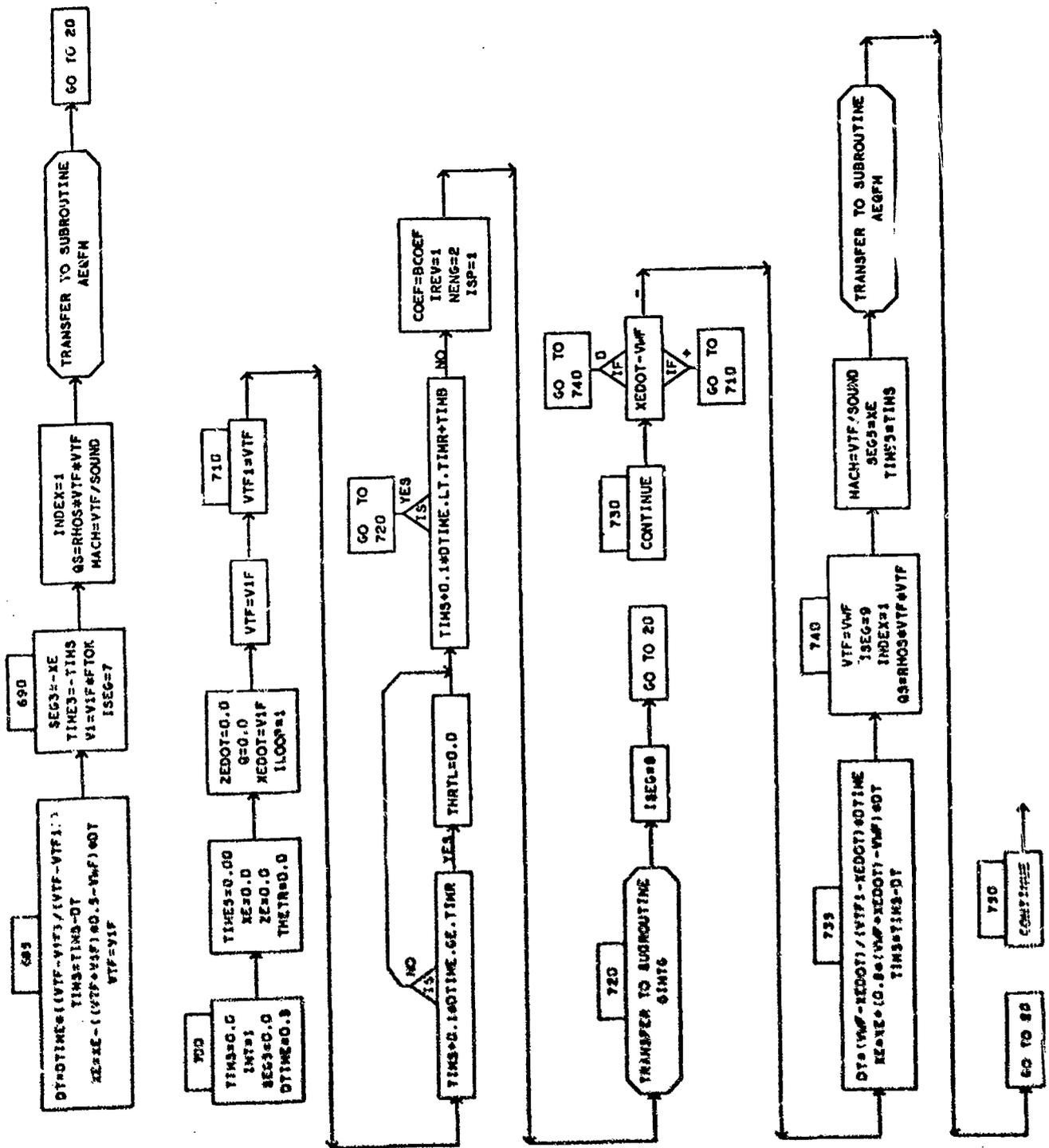
SUBROUTINE SKLTP (NSOL, CH)

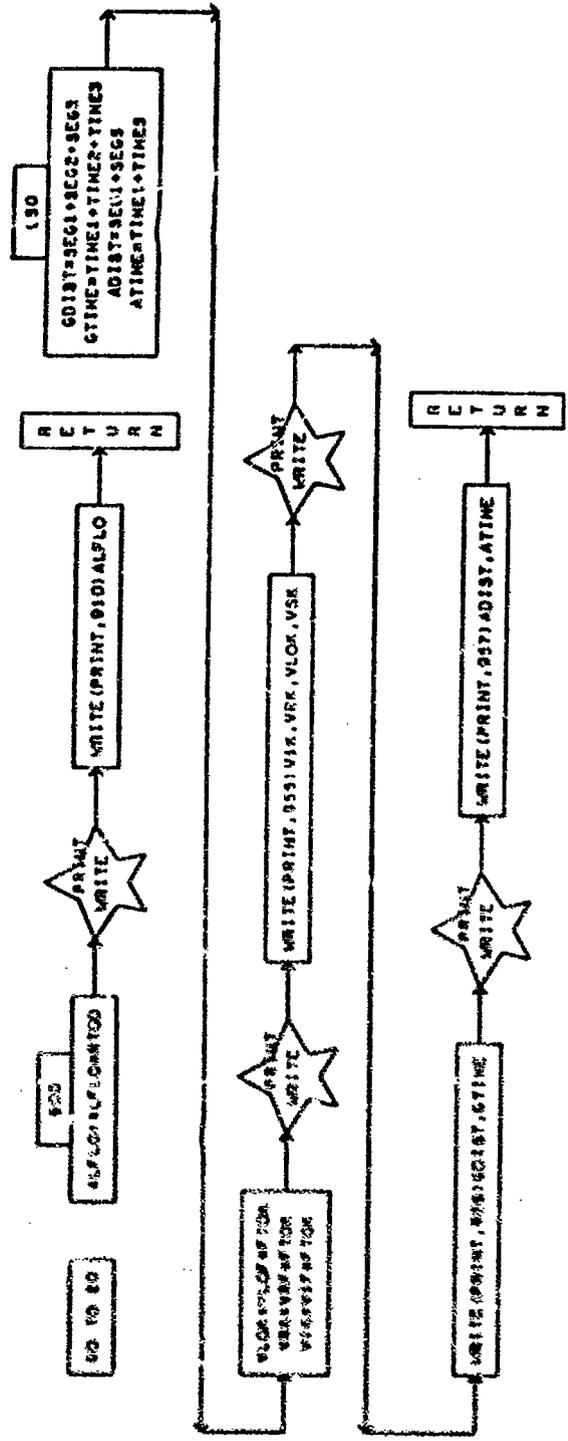


SUBROUTINE TAKEOFF









P. ROUTINE TRIN(A, V, PV, T, R)

