

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Trajectory Module of the NASA Ames Research Center Aircraft Synthesis Program ACSYNT

Michael E. Tauber and John A. Paterson

(NASA-TM-78497) TRAJECTORY MODULE OF THE
NASA AMES RESEARCH CENTER AIRCRAFT SYNTHESIS
PROGRAM ACSYNT (NASA) 73 p HC A04/MF A01

CSSL 01C

N78-26133

G3/05 Unclass
23352

July 1978



National Aeronautics and
Space Administration

Trajectory Module of the NASA Ames Research Center Aircraft Synthesis Program ACSYNT

Michael E. Tauber

John A. Paterson, Ames Research Center, Moffett Field, California



National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

TABLE OF CONTENTS

	Page
SUMMARY	1
INTRODUCTION	1
GENERAL PROGRAM DESCRIPTION	2
CONTROL PROGRAM-TRAJA	3
INPUT-TRAJIN	4
MISSION	7
Takeoff (TAKEOF)	8
Climb (CLIMB)	9
Cruise (CRUISE)	11
Acceleration (ACCEL)	13
Combat (COMBAT)	13
Loiter (LOITER)	15
Descent (DESCNT)	16
OUTPUT-TRAJOO	17
PLOTS-TRPLOT	19
PROGRAM LISTING	21
REFERENCES	69

ORIGINAL PAGE IS
OF POOR QUALITY

TRAJECTORY MODULE OF THE NASA AMES RESEARCH CENTER

AIRCRAFT SYNTHESIS PROGRAM ACSYNT

Michael E. Tauber and John A. Paterson

Ames Research Center

SUMMARY

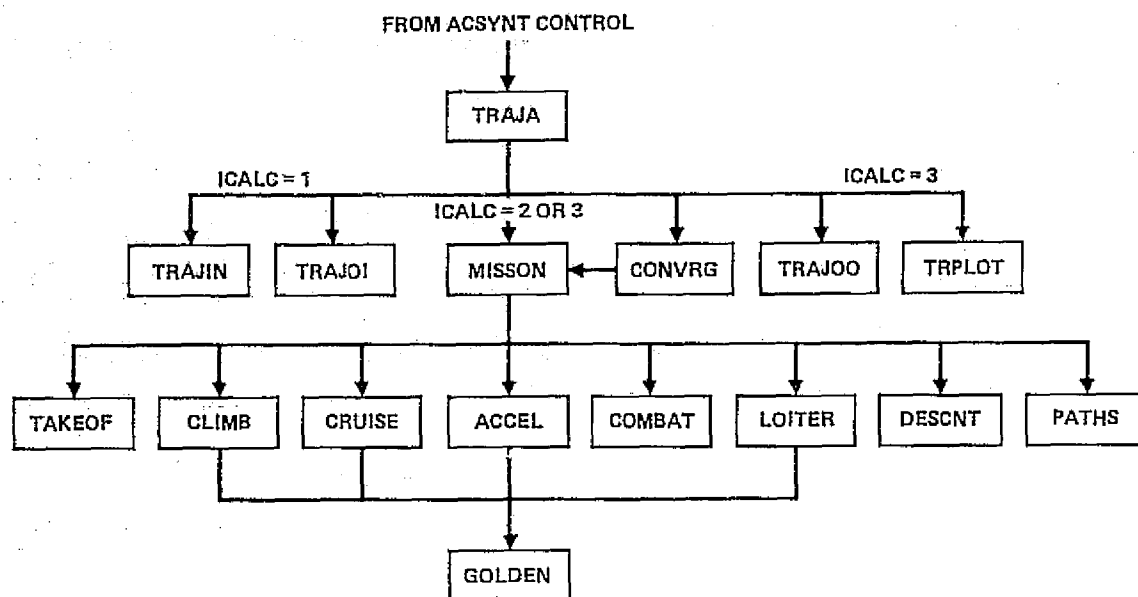
A program was developed to calculate trajectories for both military and commercial aircraft for use in the NASA Ames Research Center aircraft synthesis program, ACSYNT. The function of the trajectory module was to calculate the changes in the vehicle's flight conditions and weight, as fuel is consumed, during the flying of one or more missions. The trajectory calculations started with a takeoff, followed by up to 12 phases chosen from among the following: climb, cruise, acceleration, combat, loiter, descent, and paths. In addition, a balanced field length was computed. The emphasis was on relatively simple formulations and analytic expressions suitable for rapid computation since a prescribed trajectory had to be calculated many times in the process of converging an aircraft design, or finding an optimum configuration. The trajectory module consists of about 2500 cards and was operational, as part of the ACSYNT program, on a CDC 7600 computer.

INTRODUCTION

Computerized aircraft design at the conceptual, or early preliminary design level, has proven to be a practical engineering tool (refs. 1-3). The development of such a program requires the efficient integration of many disciplines such as the aircraft's geometric layout, weight estimation, aerodynamics, propulsion, and trajectory analysis. This report contains a description of a program which was developed to calculate trajectories for both military and commercial aircraft and was used in the NASA Ames Research Center aircraft synthesis program, ACSYNT. The function of the trajectory module was to calculate the changes in the vehicle's flight conditions and weight, as fuel is consumed, during the flying of one or more missions. The trajectory calculations started with a takeoff, followed by up to 12 phases chosen from among the following: climb, cruise, acceleration, combat, loiter, descent, and paths. In addition, a balanced field length was computed. The emphasis was on relatively simple formulations and analytic expressions suitable for rapid computation since a prescribed trajectory had to be calculated many times in the process of converging an aircraft design, or arriving at an optimum configuration. The trajectory module consists of about 2500 cards and was operational, as part of the ACSYNT program, on a CDC 7600 computer. Although it could potentially be modified, the program is presently not suitable for stand-alone computations since it depends on inputs from a propulsion module for thrust and fuel flow rate, and an aerodynamics module for lift and drag coefficients.

GENERAL PROGRAM DESCRIPTION

The function of the trajectory module is to calculate the changes in the flight conditions and weight that take place in a vehicle during the flying of one or more missions. Any number of missions may be flown during a given computer run, where each mission is divided into phases, and particular phases may be further subdivided into legs. A given mission may have up to 12 phases, selected in any order from among the following 7 types: climb, cruise, acceleration, combat, loiter, descent, and paths. The trajectory module (called TRAJ hereafter) consists of 16 different subroutines organized as shown in the block diagram below. The arrows point in the direction of called subroutines.



A brief description of the above subroutines follows:

- TRAJA - This is the master control program for TRAJ. ICALC is a control variable passed from ACSYNT control to TRAJA and informs TRAJA what tasks are to be performed.
- TRAJIN - Reads all input data for TRAJ.
- TRAJOI - Prints all input data for TRAJ.
- MISSION - Secondary control program for TRAJ. Calls all necessary subroutines needed to fly one mission. Subroutines for each phase of the mission are called in the same order as the various phases of the mission are read in by TRAJIN.

TAKEOF - Performs all calculations required during takeoff. This is always the first routine called by MISSION. It is not counted as a phase of the mission.

CLIMB - Performs all calculations of a climb phase.

CRUISE - Performs all calculations of a cruise phase.

ACCEL - Performs all calculations of an acceleration phase.

COMBAT - Performs all calculations of a combat phase.

LOITER - Performs all calculations of a loiter phase.

DESCNT - Performs all calculations of a descent phase.

PATHS - Performs all calculations of a paths phase. Input consists of up to 50 pairs of altitude vs velocity (or Mach number) data points.

GOLDEN - Performs golden section type of search for (1) Breguet altitude or (2) maximum endurance altitude or (3) optimum endurance Mach number.

CONVGR - Converges vehicle on gross weight. Process consists of taking estimated gross weight, calling on MISSION to get computed gross weight, comparing estimated and computed gross weight to get a new value of estimated gross weight, and repeating this process until vehicle weight has converged to within a specified tolerance limit.

TRAJ00 - Prints all output data for one mission.

TRPLOT - Computes and writes onto tape various parameters which are to be displayed by some graphics device.

CONTROL PROGRAM - TRAJA

This is the master control program for TRAJ. ACSYNT control communicates with TRAJ only through this program. When ACSYNT control calls TRAJA, it always sends a value for a variable called ICALC which tells TRAJA what tasks are to be performed by TRAJ.

Briefly, when ICALC = 1, TRAJ reads and prints all TRAJ input data. When ICALC = 2, TRAJ executes for mission 1 (the basic mission). When ICALC = 3, TRAJ makes one final execution of mission 1 and prints output. Further, if other missions exist (missions 2, 3, 4, . . .), then these missions are executed over and over until they have converged on gross takeoff weight, with the airframe kept fixed as it existed after mission 1 was completed. The output of these converged missions is then printed:

After mission 1 has been completed, the fixed airframe weight is computed by the equation:

$$W_{FIXED} = W_{GTO} - W_{TOT} - W_{PL}$$

where W_{GTO} is the gross takeoff weight of the converged vehicle, W_{TOT} is the total fuel weight, and W_{PL} is the payload weight.

INPUT-TRAJIN

This routine reads all the TRAJ input data. The data consist of two parts. The first part consists of general input data which is not specific to any one mission. This part is stored in core memory only. The second part is the data relevant to some particular mission. This second part is stored on scratch tape where it can be read back into core memory, mission by mission, as it is needed.

The general TRAJ input data consists of the following 40 variables:

- (1) TIMTO1 - Time during takeoff at first power setting
- (2) TIMTO2 - Time during takeoff at second power setting
- (3) FRFURE - Fuel reserve, as fraction of total fuel
- (4) DESLF - Design load factor
- (5) ULTLF - Ultimate load factor
- (6) RANGE - Aircraft range
- (7) WFUEL - Total internal fuel weight
- (8) WFEXT - External fuel weight
- (9) WFTRAP - Weight of internally trapped fuel
- (10) MENDUR - Mach number used for final endurance calculation
- (11) QMAX - Maximum dynamic pressure
- (12) XDESC - Initial guess for horizontal distance covered during descent
- (13) WKFUEL - Fuel multiplication factor
- (14) CRMACH - Cruise Mach number
- (15) NCRUSE - Number of cruise legs in mission
- (16) IPSIZE - Power setting for sizing engine
- (17) IPSTOI - Power setting during first part of takeoff

ORIGINAL PAGE IS
OF POOR QUALITY

- (18) IFSTO2 - Power setting during second part of takeoff
- (19) IBREG - Breguet altitude search indicator
- (20) IENDUR - Endurance altitude search indicator
- (21) IPRINT - Diagnostic print indicator
- (22) KERROR - Error print indicator
- (23) NLEGCL - Number of legs per climb phase
- (24) NLEGCR - Number of legs per cruise phase
- (25) NLEGLO - Number of legs per loiter phase
- (26) MILCOM - Takeoff obstacle height indicator

The following quantities are used in the landing field computation:

- (27) WKLAND - $WLAND = WGTO - WKLAND * WFTOT$
- (28) FLFAC - $FLLAND = (XAIR + XGRLAN)/FLFAC$
- (29) DECEL - Deceleration factor

The following quantities are used for the plotting routine:

- (30) IPLOT - Plotting indicator
- (31) HMINP - Minimum altitude
- (32) HMAXP - Maximum altitude
- (33) DELHP - Altitude interval to be used
- (34) SMMINP - Minimum Mach number
- (35) SMMAXP - Maximum Mach number
- (36) DELMP - Mach number interval to be used
- (37) WCOMBP - Aircraft weight used

The following quantities are used for multimission computations:

- (38) NMISS - Number of missions per job
- (39) FWGMAX - Maximum weight factor used in convergence
- (40) TOL - Convergence tolerance

The input for each mission consists of 2 variables which apply to the whole mission, and a series of 14 variables which apply to each phase of the mission.

The two general variables are:

- (1) NPHASE - The number of phases in the mission
- (2) WPL - Payload weight (not needed for mission 1, where it is supplied by the weights module)

The following 14 variables are input for each phase of the mission, although no single phase makes use of all 14 variables.

- (1) MSTART - Starting Mach number. If $MSTART > 0$, then starting Mach number is set equal to MSTART. If $MSTART < 0$, then starting Mach number is set equal to final Mach number from preceding phase.
- (2) MEND - Ending Mach number
- (3) HSTART - Starting altitude. If $HSTART \geq 0$, then starting altitude is set equal to HSTART. If $HSTART < 0$, the starting altitude is set equal to final altitude from preceding phase.
- (4) HEND - Ending altitude. In CLIMB, if $IBREG = 0$ or $HEND > 0$, then ending altitude is set equal to HEND; otherwise, ending altitude is determined by making a Breguet search.
- (5) X - Horizontal distance covered
- (6) TIM - Time
- (7) NT - Number of turns
- (8) VIND - Constant indicated airspeed (for climb)
- (9) IP - Power setting
 - = 1 maximum afterburner power
 - = 2 intermediate (commercial maximum takeoff power)
 - = 3 maximum continuous power
 - = 4 thrust = drag (cruise)
 - = 5 idle
- (10) IX - Indicator which tells whether horizontal distance covered during CLIMB, ACCEL, or DESCNT will be added, subtracted, or ignored in computing horizontal distance to be covered during CRUISE phase.
- (11) IW - Weapons drag indicator
- (12) IPRT - Print indicator

(13) IB - Bombs drop indicator

(14) IA - AMMO drop indicator

The table below indicates which inputs may possibly be needed for each of the seven types of phases.

PHASE	1 MSTART	2 MEND	3 HSTART	4 HEND	5 X	6 TIM	7 NT	8 VIND	9 IP	10 IX	11 IW	12 IPRT	13 IB	14 IA
CLIMB	✓	✓	✓	✓				✓	✓	✓	✓	✓	✓	✓
CRUISE	✓		✓		✓				✓	✓	✓	✓	✓	✓
ACCEL	✓	✓	✓						✓	✓	✓	✓	✓	✓
COMBAT	✓		✓			✓	✓		✓		✓	✓	✓	✓
LOITER	✓		✓		✓	✓					✓	✓	✓	✓
DESCNT				✓							✓	✓	✓	✓
PATHS	Mach vs h table								✓		✓	✓	✓	✓

MISSION

The primary function of this subroutine is to call all the subroutines needed to fly one mission. For example, if a given mission consists of three phases such as climb, cruise, and descent, then MISSION will first call TAKEOF (always called first for all missions), then CLIMB, then CRUISE, then DESCNT. The order of the called subroutines is determined by the order of the various phases of the mission as read by TRAJIN.

After this, MISSION computes used, reserve, total, and internal fuel weights according to the following formulas:

$$\begin{aligned}
 \text{WFUSED} &= \text{WKFUEL} * (\text{WGTO} - \text{W} - \text{f(W)}) \\
 \text{WFRES} &= \text{WFUSED} / (1. - \text{FRFURE}) - \text{WFUSED} \\
 \text{WFTOT} &= \text{WFUSED} + \text{WFRES} + \text{WFTRAP} \\
 \text{WFUEL} &= \text{WFTOT} - \text{WFEXT}
 \end{aligned}$$

where WFUSED is the weight of the fuel used, WGTO is gross takeoff weight, W is the landing weight of the vehicle, f(W) is the weight of all missiles, bombs, ammunition, and external tanks jettisoned during the mission, WKFUEL is a multiplying factor, WFRES is the reserve fuel weight, FRFURE is the fraction of fuel kept in reserve, WFTOT is the total fuel weight, WFTRAP is the weight of fuel trapped in the fuel tanks, WFUEL is the weight of internal fuel, and WFEXT is the weight of fuel stored in external tanks.

Next, the landing field length is determined by the following sequence of equations (ref. 4):

$$\begin{aligned}
 \text{WLAND} &= \text{WGTO} - \text{WKFUEL} * \text{WFUSED} \\
 \text{VSTALL} &= \text{SQRT}[2. * \text{WLAND} / (\text{SWING} * \text{CL} * \text{RHO})] \\
 \text{VSCREN} &= 1.3 * \text{VSTALL}
 \end{aligned}$$

$$\begin{aligned}
VTDOWN &= 1.15 * VSTALL \\
XAIR &= [(VSCREN^{**2} - VTDOWN) / 64.4 + 50.] * RLD \\
XGRLAN &= VTDOWN^{**2} / (64.4 * DECEL) \\
FLAND &= (XAIR + XGRLAN) / FLFAC
\end{aligned}$$

where WKLAND, DECEL, and FLFAC are input to TRAJ (see TRAJIN writeup), RHO is sea level air density, CL is lift coefficient at landing, and RLD is lift to drag ratio.

Next, the endurance altitude and endurance time are calculated based on the approximate (input) Mach number (MEENDUR). (The results of the approximate calculation for endurance altitude and time were only intended for use in the economics module of ACSYNT and are never used within TRAJ for performance or fuel calculations.) A golden section type of search is performed to find the endurance altitude corresponding to the maximum value of the endurance factor (RLD/SFC). Using this value of the endurance factor, the endurance time is calculated from $RLD/SFC[WGTO/(WGTO - WFUSED)]$.

Finally, the gross weight of the vehicle is calculated using the equation

$$WGALC = WFIXED + WPL + WFTOT$$

where WPL is the payload and WFIXED is equal to $WGTO - WFTOT - WPL$ calculated just once after mission 1 has been converged. WGALC is needed for transfer back to CONVGR.

Takeoff

The first calculation performed in subroutine TAKEOF is the determination of the total static thrust of the aircraft, and the corresponding value of SFC. The fuel used during startup and taxi is based on the static value of the SFC for a power setting which is input (IPSTOL) and for an input time (TINTOL).

The takeoff balanced field length is computed using an expression from reference 4, which is written

$$FLTO = 2.10 \left(\frac{0.01163 W}{\rho C_{L_2} S} + 0.374 TOOBHT \right) \left(\frac{1}{k_t \frac{T}{W} - 0.04} + 2.7 \right) + \frac{32.0}{\rho}$$

where FLTO is in ft, W/S in psf and ρ in slugs/ft³. The lift coefficient C_{L_2} is evaluated at $1.2 V_{stall}$, where

$$V_{stall} = \sqrt{\frac{2 WGTO}{\rho C_{L_{stall}} S}}$$

k_t is the thrust lapse factor during takeoff and the program uses the average of the value at takeoff and the value at a velocity of $1.2 V_{stall}$. TOOBHT

is the obstacle height and is 50 ft (15.24 m) for military aircraft and 35 ft (10.67 m) for commercial aircraft.

The fuel used during takeoff can be calculated by one of two methods. If the takeoff time (TIMT02) is input as a positive or zero value, the fuel used is

$$WFTO2 = 0.0167(TIMT02)(SFC)(TN)$$

where the SFC and thrust (TN) are based on conditions at the end of takeoff. Alternatively, if TIMT02 is set negative, the fuel used is based on the balanced field length and is

$$WFTO2 = \frac{(TN_{AVE})(SFC_{AVE})(FLT0)}{1800.(1.2 V_{stall})}$$

where TN_{AVE} and SFC_{AVE} are the values averaged between static and takeoff conditions.

The total fuel used is the sum of the values for startup, taxi and takeoff

$$WFTO = WFT01 + WFTO2$$

Climb

The program can calculate climb in either of two modes. When $VIND > 0$, climb is calculated using a constant indicated airspeed. When $VIND \leq 0$, climb is calculated using an approximate minimum fuel path. If the aircraft is to cruise at Breguet altitude ($IBREG > 0$), the program performs a search for the ending altitude; otherwise ending altitude must be input ($IBREG = 0$).

The forces acting on the airplane during climb vary with altitude, necessitating dividing the climb phase into a finite number of legs. The conditions changing with altitude include the temperature, pressure, and density of the ambient air, and the weight and Mach number of the vehicle. Also, since the available power decreases at an ever greater rate as the ceiling altitude of the vehicle is approached, it is desirable to decrease the altitude increments of the legs as altitude increases.

The total number of legs in a climb phase is given by the formula

$$\begin{aligned} NLEGS &= \frac{\text{total interval}}{\text{average step size}} = \int_{H_1}^{H_2} \frac{dH}{f(H)} = \int_{H_1}^{H_2} \frac{dH}{4000 - 0.05 H} \\ &= 20 \ln \left(\frac{4000 - 0.05 H_1}{4000 - 0.05 H_2} \right) \end{aligned}$$

where H_1 is the starting altitude of the phase, H_2 is the ending altitude of the phase, and $f(H)$ is an expression giving step size as a function of altitude. A further restriction on NLEGS is given by the relation $3 \leq \text{NLEGS} \leq 20$. The above formula can be overridden by giving NLEGCL a positive integer value, in which case, $\text{NLEGS} = \text{NLEGCL}$. This, however, will cause all climb phases in all missions of a run to have the same value for NLEGS.

Once the number of legs in a climb phase is determined, the distance climbed during each leg is given by the formula:

$$\text{DELH} = \ln \left[\frac{1 + (\text{LEG})(e - 1)/\text{NLEGS}}{1 + (\text{LEG} - 1)(e - 1)/\text{NLEGS}} \right] (H_2 - H_1)$$

where LEG is the current leg being executed by the program. This causes the step size to decrease logarithmically with increasing altitude.

Summing forces in the axial direction, the equations of motion give

$$\frac{\Delta V}{\Delta t} = g \left(\frac{T}{W} - \frac{D}{W} - \sin \gamma \right)$$

while in the normal direction

$$L + T \sin \alpha = W \cos \gamma$$

which gives two equations in three unknowns. The three unknowns are T, L, and D, or γ .

For the approximation to the most economical or fastest climb, the relation for γ comes from reference 5, and is

$$\sin \gamma = \frac{5}{6} \frac{T}{W} - \frac{\sqrt{\left(\frac{T}{W}\right)^2 E_m^2 + 3}}{6 E_m} - \frac{3}{2 E_m \left[\frac{T}{W} E_m + \sqrt{\left(\frac{T}{W}\right)^2 E_m^2 + 3} \right]}$$

where

$$E_m = \frac{C_L}{2 \sqrt{C_{D_0} C_{D_L}}}$$

For the constant indicated airspeed climb,

$$\sin \gamma = \frac{\Delta h}{V \Delta t}$$

where, now

$$V = \frac{V_{ind}}{\sqrt{\rho/\rho_{SL}}} = \frac{V_{ind}}{\sqrt{\sigma}}$$

and where σ is the density ratio referenced to the sea level value. By differentiating the equation for velocity with respect to time, we find that the relationship between time and altitude is

$$\Delta t = \frac{\frac{\sqrt{\sigma}}{V_{ind}} \Delta h - \frac{V_{ind}}{2g\sigma^{1.5}}}{\frac{T}{W} - \frac{\rho_{SL} V_{ind}^2 C_D S}{2W}}$$

We now have all the necessary terms for calculating flight-path angle.

The fuel used is calculated from

$$\Sigma W_F = \Sigma (sfc) T \Delta t$$

and the equivalent ground distance travelled during climb is

$$\Sigma X = \Sigma V \cos \gamma \Delta t$$

Cruise

The starting altitude for the cruise phase will be the input value (HSTART) if HSTART is positive. If HSTART = 0 and IBREG = 1, then the starting altitude will be determined by a Breguet search. The ending altitude will be the same as the starting altitude only if the input value (HEND) is set equal to -1. Otherwise, the airplane will be allowed to climb with each leg of the cruise phase as fuel is burned off.

At the beginning and at the end of the cruise phase a test is made to determine whether the fuel used to that point exceeds the weight of the fuel stored in the external tanks if these are present. If it does, then the external fuel tanks are dropped immediately.

In general, the horizontal distance travelled in a given cruise phase is calculated using the formula

$$X_{CRUSE} = X + DXCRUS + XDESC$$

where X_{CRUSE} is the actual horizontal distance that will be traversed in the cruise phase, X is the nominal horizontal distance input for that phase, $DXCRUS$ is the algebraic sum of the horizontal distances traversed in the preceding climb and acceleration phases (but following the preceding cruise

phase if one exists), and XDESC is the horizontal distance traversed in the last descent phase of the mission. An estimated value for XDESC is inputted and used for the first iteration of the mission, and the calculated value from the previous iteration is used for all succeeding iterations of the mission. DXCRUS is set to zero at the beginning of the mission and is reset to zero at the end of every cruise phase. The calculated value of horizontal distance traversed in a climb or acceleration phase will be added or subtracted from DXCRUS or ignored depending on whether IX for that phase is +1, -1, or 0, respectively. Similarly, XDESC will be positive, negative, or zero depending upon whether IX for the cruise phase under consideration is +1, -1, or 0, respectively.

When X is given a negative value, then the formula for XCRUSE becomes

$$XCRUSE = RANGE/NCRUSE + DXCRUS + XDESC$$

where RANGE and NCRUSE are input at the beginning of the run.

The number of legs in a given cruise phase (NLEGS) is given by the following inequalities:

$$X < 100, \text{ then } NLEGS = 2$$

$$100 \leq X < 300, \text{ then } NLEGS = 3$$

$$300 \leq X < 1000, \text{ then } NLEGS = 4$$

$$1000 \leq X < 3000, \text{ then } NLEGS = 5$$

$$X \geq 3000, \text{ then } NLEGS = 6$$

The cruise consists of an approximation to the most efficient flight path which uses a stepwise cruise-climb path. There are two options for determining the initial cruising altitude: (1) the Breguet altitude, found by maximizing the value of $(V/sfc)(L/D)$ as a function of altitude and (2) altitude is specified.

The summation of forces tangential to and normal to the flight path are, respectively,

$$T \cos \alpha = D \quad \text{and} \quad L = W$$

From these relations, the weight of fuel used is

$$\Sigma W_F = \Sigma \frac{C_D W(sfc)}{\cos \alpha} \sqrt{\frac{\rho}{2C_L} \frac{S}{W} \Delta X}$$

where ΔX is the distance flown at each altitude and the cruise altitude is increased, as the aircraft burns fuel, according to the relation

$$\rho = \frac{2W}{C_L V^2 S}$$

and C_L is kept constant at the value corresponding to $(L/D)_{\max}$; otherwise, C_L is determined by the input values of Mach number and altitude.

The cruising time and the equivalent ground distance covered are, respectively,

$$t = \sum \frac{\Delta X}{V} \quad \text{and} \quad X = \sum \Delta X$$

Acceleration

The purpose of the acceleration phase (ACCEL) is to increase the Mach number of the vehicle from an initial low value to a specified higher value. The phase is divided into steps with the size of the Mach number steps being 0.02 between Mach 0.87 and 1.1 and equal to 0.05 elsewhere.

The aircraft is assumed to accelerate at a fixed altitude. The equation of motion along the flight path is

$$\frac{dV}{dt} = g \left(\frac{T}{W} \cos \alpha - \frac{D}{W} \right)$$

from which we can calculate the fuel used to be

$$W_F = W \left\{ 1 - \exp \left[\frac{1}{g} \int \frac{(sfc)T}{T \cos \alpha - D} dV \right] \right\}$$

The time required to accelerate is

$$t = \frac{1}{g} \int \frac{W dV}{T \cos \alpha - D}$$

and the equivalent ground distance covered is given by

$$X = \frac{1}{g} \int \frac{W dV}{T \cos \alpha - D}$$

Combat

In the combat phase, the acceleration and turning capabilities are calculated and also the amount of fuel used by performing combat at a fixed altitude, either for a specified time or a fixed number of turns. The airplane's potential acceleration capability is expressed as the "specific excess power," for which the expression is derived below.

The specific energy (sum of potential and kinetic energy per unit airplane weight) is

$$E_S = \frac{E}{W} = h + \frac{V^2}{2g}$$

and differentiating the specific energy with respect to time gives the specific power

$$P_S = \frac{dE_S}{dt} = \frac{dh}{dt} + \frac{V}{g} \frac{dV}{dt}$$

Summing forces along the flight path gives

$$T \cos \alpha - D - W \sin \gamma = m \frac{dV}{dt}$$

Dividing both sides by W and multiplying by V

$$\frac{V(T \cos \alpha - D)}{W} = V \sin \gamma + \frac{V}{g} \frac{dV}{dt}$$

and, since

$$\frac{dh}{dt} = V \sin \gamma$$

we get

$$P_S = \frac{dE_S}{dt} = \frac{V(T \cos \alpha - D)}{W}$$

The relations for turning radius and turning rate are, respectively (ref. 6)

$$R = \frac{V^2}{g \tan \phi} \quad \text{and} \quad \frac{d\theta}{dt} = \dot{\theta} = \frac{V}{R}$$

where the airplane bank angle, ϕ , is related to the load factor, n , through

$$\cos \phi = \frac{1}{n}$$

The load factor is written as

$$n = \frac{L + T \sin \alpha}{W}$$

and cannot exceed the structural design load factor.

Two types of turns are considered. These are: (1) sustained, or coordinated turns for which $P_s = 0$, requiring the solution of the equation $T \cos \alpha = D$ to obtain the aerodynamic parameters needed to find the load factor and (2) instantaneous turns corresponding to the maximum value of P_s and limited by either the maximum lift coefficient, $C_{L_{max}}$, or the structural design load factor of the airplane.

The time required to complete a turn is

$$t = \frac{2\pi R}{V}$$

and the fuel used during combat turns is

$$W_F = \Sigma T(sfc)Nt$$

where N is the number of turns required.

Loiter

The starting altitude for the loiter phase will be the input value (HSTART) if HSTART is positive. If HSTART = 0 and IENDUR = 1, then a search will be made to determine the most economical starting altitude.

The Mach number will change from leg to leg or remain constant throughout the loiter phase depending on the input value of MSL/RT. Ordinarily, a search for the most economical loiter Mach number will be made for each leg of the phase. However, whenever MSTART is positive, the Mach number will remain equal to MSTART throughout the phase and the altitude in each leg will increase to take advantage of the reduced fuel weight.

Each loiter phase is divided into legs, the number of which depends on the amount of time to be spent in the phase. The number of legs is equal to the time in hours rounded up to the next higher integer, but is not allowed to exceed six.

There are two options for determining the loiter flight conditions: (1) most economical loiter altitude and/or velocity is found by maximizing the value of $(L/D)/sfc$ as a function of altitude, velocity, or both, and (b) altitude and/or velocity is specified.

The fuel used is calculated from

$$\Sigma W_F = \Sigma T(sfc)\Delta t$$

where t is the loiter time. As the aircraft burns fuel, the loiter altitude is increased, or the speed decreased, to maintain the most economical flight condition according to the relation

$$\rho = \frac{2W}{C_L S V^2}$$

Since loitering generally consists of flying in circles, it may be desirable to fly in a smaller circle than that corresponding to the most economical one. To decrease the loiter circle size, more lift is needed. The equation of motion normal to the flight path is

$$L + D \tan \alpha = \frac{W}{\sqrt{1 + \left(\frac{V^2}{gR}\right)^2}}$$

where R is the loiter circle radius. The exact solution is a tedious iteration between C_L , C_D , and α , since

$$C_L - C_D \tan \alpha = \frac{2W}{\rho V^2 S} \sqrt{1 + \left(\frac{V^2}{gR}\right)^2}$$

However, for most cases of practical interest

$$C_D \tan \alpha \ll C_L$$

and, we have used the simple approximation

$$C_L \approx \frac{2W}{\rho V^2 S} \sqrt{1 + \left(\frac{V^2}{gR}\right)^2}$$

Descent

The most economical descent path is the flattest descent with engines at idle power setting. The aircraft should fly at $(L/D)_{\max}$, thus $C_{D_i} = C_{D_o}$ and for flight with thrust much less than drag, the optimum descent angle is approximately (ref. 6)

$$\sin \gamma = - \frac{2C_{D_o}}{C_L}$$

Summation of forces normal to the flight path gives

$$L = W \cos \gamma$$

resulting in the following expression for the speed

$$V = \sqrt{\frac{2W \cos \gamma}{C_L \rho S}}$$

The descent time is calculated from

$$\Sigma \Delta t = \Sigma \frac{-\Delta h}{V \sin \gamma}$$

while the fuel used is

$$\Sigma W_F = \Sigma T(\text{sfc})\Delta t$$

and the equivalent ground distance covered is

$$\Sigma \Delta X = \Sigma V \cos \gamma \Delta t$$

OUTPUT-TRAJOO

The output quantities are listed in tabular form and consist of 19 quantities representing values at the end of each phase. In addition, there are eight values giving a fuel weight. If there are combat phases, an additional 21 values of combat parameters are printed for each combat phase. Lastly, there are 12 miscellaneous quantities printed.

The following 19 quantities are printed for each phase:

- (1) Mach number
- (2) Altitude
- (3) Lift coefficient
- (4) Angle of attack
- (5) Fuel used in phase
- (6) Flight time in phase
- (7) Flight velocity
- (8) Specific fuel consumption (installed)
- (9) Specific fuel consumption (uninstalled)
- (10) Total thrust (installed)
- (11) Total thrust (uninstalled)
- (12) Drag coefficient
- (13) Lift-drag ratio
- (14) Weight at end of phase
- (15) Engine airflow rate
- (16) Dynamic pressure
- (17) Horizontal distance covered

(18) Engine installation drag coefficient

(19) Pressure recovery

The following eight quantities give a fuel weight:

- (1) Fuel weight during takeoff with first power setting
- (2) Fuel weight during takeoff with second power setting
- (3) Mission fuel weight
- (4) Reserve fuel weight
- (5) Trapped fuel weight
- (6) Internal fuel weight
- (7) External fuel weight
- (8) Total fuel weight

The following seven quantities are printed for each combat phase for each of three conditions: 1g flight, sustained, and instantaneous — making a total of 21 combat parameters in all. They are:

- (1) Specific excess power
- (2) Load factor
- (3) Turning rate
- (4) Radius
- (5) Angle of attack
- (6) Lift coefficient
- (7) Drag coefficient

The following 12 miscellaneous quantities are also printed out:

- (1) Total mission time
- (2) Total mission range
- (3) Takeoff field length (total run)
- (4) Landing field length (total run)
- (5) Landing field length (ground run)

- (6) Weight used for landing calculation
- (7) Takeoff weight
- (8) Landing weight
- (9) Endurance Mach number (input value)
- (10) Endurance altitude
- (11) Endurance time
- (12) Loiter radius

PLOTS-TRPLOT

This routine calculates various parameters which are written onto tape so that they can later be displayed graphically. The main part of this routine calculates certain parameters which may be displayed graphically on a plot of altitude vs Mach number. These parameters are:

- (1) PSIG = specific excess power at 1g
- (2) NZS = load factor during turn
- (3) TDOTS = sustained turning rate
- (4) NZI = maximum instantaneous load factor
- (5) PSI = maximum instantaneous specific excess power
- (6) TDOTI = maximum instantaneous turning rate
- (7) Q = dynamic pressure

These parameters are calculated for a certain range of values of altitude and Mach number with beginning, ending, and incremental values of each being input.

The inputs to the routine are:

- (1) IPLOT = 0, no print, no plot
 = 1, print, no plot
 = 2, no print, plot
 = 3, print, plot
- (2) HMINP = minimum altitude

- (3) HMAXP = maximum altitude
- (4) DELHP = altitude interval
- (5) SMMINP = minimum Mach number
- (6) SMMAXP = maximum Mach number
- (7) DELMP = Mach number interval
- (8) WCOMBP ≤ 0 , aircraft weight taken from first COMBAT phase
 $0 < \text{WCOMBP} \leq 1$, aircraft weight equals WCOMBP times takeoff weight
 WCOMBP > 1 , aircraft weight equals WCOMBP

The second part of this routine calculates landing field lengths at altitudes of 0, 4000 ft (1219 m), 8000 ft (2438 m), and 12,000 ft (3657 m).

PROGRAM LISTING

SUBROUTINE TRAJA

76/76 OPT=2

FTN 4.5+410

```

1      SUBROUTINE TRAJA (ICALC,NERROR,ICEO,KGPRNT,IGPLT)
      EXTERNAL MISSION
      REAL MSTART,MEND,NT,MENDUR
      COMMON /TRAJCM/ ALPHA,ARW,CD,CDD,CL,DESLF,DRAG,EN,HN,KP,
5      1 KLD,SMN,QMAX,RANGE,SFC,SWING,THRUST,TW,ULTLF,W,WAMMUN,WETANK,
      2 WFUEL,WGTOWT,WMISS,WTOT,WPLWT,MSTART(12),HSTART(12),TIM(12),
      3 PSIGT(12),TDOTST(12),NZST(12),PSIT(12),TDOIT(12),NZIT(12),
      4 WBOMBS,WFEKT,SFCU,THRSTU,CDINSP,PRTOT,WKFUEL,CRMACH,FLTQ,X(12),
10     5 FLLAND,TENDUR,BLRANG,BLTIME,TIMTO1,TIMTO2,WFTO,PNAME1(12),
      6 PNAME2(12),SMNT(12),HNT(12),WF1(12),TIMET(12),XT(12),CLT(12),
      7 CDT(12),ALPHAT(12),CLIT(12),CDIT(12),ALPHIT(12),WFUSED,WLAND,
      8 QV(12),RLDT(12),SFCT(12),TNT(12),CET(12),DY(18),
      9 IAO,IPS,ITS,IWS,IBS,NPHASE,IDY(14)
      COMMON /TRAJEX/ ALPIGT(12),CDINST(12),CDIGT(12),CLIGT(12),DECEL,
15     1 DELHP,DELMP,DXCRUS,FLFAC,FRFURE,HEND(12),HENDUR,HMAXP,HMINP,
      2 HPAHT(50),MEND(12),MENDUR,NT(12),PRTOT(12),RADIT(12),
      3 RADST(12),SF,SFCUT(12),SMMAXP,SMMINP,
      4 TNUT(12),VIND(12),VELT(12),VPATHT(50),WART(12),WCOMBP,WEELT(12),
20     5 WFRES,WFTOT,WFTO1,WFTO2,WFTAP,WKLAND,XDESC,XGRLAN,FWGMAX,TOL,
      6 WFIXED,WGCALC,WPL,WGTO,DUMY(82),IA(12),IAS,IB(12),IBREG,IENDUR,
      7 IMISS,IP(12),IPHASE,IPLT,IPRINT,IPRT(12),IPSIZE,IPSTO1,IPSTO2,
      8 IW(12),IX(12),KERROR,MILCOM,NCRUSE,NLEGCL,NLEGCR,NLEGLO,NMISS,
      9 NPATH,IDUMY(6)
      DATA PATH/4HPATH/
25     WPL=WPLWT
      WGTOW=WGTOWT
      IF (KGPRNT.EQ.0) GO TO 5
      KSTORE=KERROR
      KERROR=2
30     5 IF (ICALC.EQ.1) GO TO 10
      IF (ICALC.EQ.2) GO TO 20
      IF (ICALC.EQ.3) GO TO 30
      IF (KGPRNT.NE.0) KERROR=KSTORE
      RETURN
35     C-----
      C   ICALC = 1
      C-----
40     10 TIMTO1=5.0
      TIMTO2=1.0
      FRFURE=.05
      DESLF=2.5
      ULTLF=3.75
      RANGE=1000.
      WFUEL=10000.
45     WFEKT=0.0
      WFTAP=100.
      GMAX=700.
      XDESC=80.
      WKFUEL=1.0
50     CRMACH=.8
      WKLAND=.57
      FLFAC=.6
      DECEL=.25
      NCRUSE=2
      IPSIZE=0
55     IPSTO1=5
      IPSTO2=2

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

        IBREG=0
        IENDUR=0
60      IPRINT=0
        KERROR=2
        IPLOT=0
        HMINP=0.0
        HMAXP=40000.
65      DELHP=4000.
        SMMINP=.3
        SMMAXP=.9
        DELMP=.1
        WCOMBP=.7
70      MILCOM=0
        MENDUR=.5
        NLEGCL=0
        NLEGCK=0
        NLEGLO=0
75      NMISS=1
        FWGMAX=1.2
        TOL=.001
        CALL TRAJIN
        CALL TRAJOI
80      C-----READ MISSION 1 FROM TAPE 22.
        12 IMISS=1
        REWIND 22
        READ (22,2200) NPHASE,WPLTRA
        2200 FORMAT(I10,E15.8)
85      DO 15 IPHASE=1,NPHASE
        READ (22,2201) PNAME1(IPHASE),PNAME2(IPHASE),MSTART(IPHASE),
        1 MEND(IPHASE),HSTART(IPHASE),HEND(IPHASE),X(IPHASE),TIM(IPHASE),
        2 NT(IPHASE),VIND(IPHASE),IP(IPHASE),IX(IPHASE),IW(IPHASE),
        3 IPRT(IPHASE),IB(IPHASE),IA(IPHASE)
90      2201 FORMAT(2A4,2X,7E15.8/7E15.8,6I10)
        IF (PNAME1(IPHASE).NE.PATH) GO TO 15
        READ (22,2200) NPATH
        READ (22,2202) (HPATHT(I),I=1,NPATH)
        READ (22,2202) (VPATHT(I),I=1,NPATH)
95      2202 FORMAT(8E15.8)
        15 CONTINUE
        C-----END OF READING MISSION 1 FROM TAPE 22.
        IF (ICALC.EQ.3) RETURN
        WFIXED=0.0
100      SF=ULTLF/DESLF
        ALPHA=0.0
        CL=0.0
        DRAG=0.0
        HN=0.0
105      SHN=0.0
        W=0.0
        IF (KGPRNT.NE.0) KERROR=KSTORE
        RETURN
110      C-----
        C ICALC = 2
        C-----
        20 CALL MISSION(ICALC,KERROR,IGED,KGPRNT)
        IF (KGPRNT.NE.0) KERROR=KSTORE
        RETURN

```

115 C-----
C ICALC = 3
C-----

30 IF (NMISS.EQ.1) GO TO 40
KCALC=2

120 CALL MISSION(KCALC,NERROR,IGEO,KGPRNT)
IF (NERROR.GE.2) RETURN

40 CALL TRAJDD
IF (IMISS.GE.NMISS) GO TO 70
IF (IMISS.GE.2) GO TO 50

125 WGEST=WGTO
WGMAX=FWGMAX+WGEST
WFIXED=WGTO-WFTOT-WPL

C-----READ NEW MISSION FROM TAPE 22.
50 IMISS=IMISS+1
130 READ (22,2200) NPHASE,WPLTRA
DO 60 IPHASE=1,NPHASE
READ (22,2201) PNAME1(IPHASE),PNAME2(IPHASE),HSTART(IPHASE),
1 HEND(IPHASE),HSTART(IPHASE),HEND(IPHASE),X(IPHASE),TIM(IPHASE),
2 NT(IPHASE),VIND(IPHASE),IP(IPHASE),IX(IPHASE),IW(IPHASE),
135 3 IB(IPHASE),IA(IPHASE)
IF (PNAME1(IPHASE).NE.PATH) GO TO 60
READ (22,2200) NPATH
READ (22,2202) (HPATHT(I),I=1,NPATH)
READ (22,2202) (VPATHT(I),I=1,NPATH)

140 60 CONTINUE
C-----END OF READING NEW MISSION FROM TAPE 22.
WPL=WPLTRA
WGEST1=WGEST
IF (IPRINT.EQ.0) GO TO 65

145 WRITE (6,600) IMISS,WGEST,WGMAX,WFIXED,WFTOT,WGTO,WGCONV,WPL,
1 WGEST1,WGCALC

600 FORMAT(1H1,18H CHECK FOR MISSION,12/23H BEFORE CALLING CONVGR:/
1 9H WGEST=,E13.6,4X,7H WGMAX=,E13.6,4X,7HWFIXED=,E13.6/
2 9H WFTOT=,E13.6,4X,7H WGTO=,E13.6,4X,7HWGCONV=,E13.6/
150 3 9H WPL=,E13.6,4X,7HWGEST1=,E13.6,4X,7HWGCALC=,E13.6)

65 CALL CONVGR(MISSION,WGEST1,WGCONV,WGMAX,TOL,IPRINT)
IF (IPRINT.EQ.0) GO TO 40
WRITE (6,601) WGEST,WGMAX,WFIXED,WFTOT,WGTO,WGCONV,WPL,WGEST1,
1 WGCALC

155 601 FORMAT(22H AFTER CALLING CONVGR:/
1 9H WGEST=,E13.6,4X,7H WGMAX=,E13.6,4X,7HWFIXED=,E13.6/
2 9H WFTOT=,E13.6,4X,7H WGTO=,E13.6,4X,7HWGCONV=,E13.6/
3 9H WPL=,E13.6,4X,7HWGEST1=,E13.6,4X,7HWGCALC=,E13.6)

GO TO 40

160 70 IF (IPLOT.GT.0) CALL TRPLOT(NERROR,IGEO,KGPRNT)
IF (KGPRNT.NE.0) KERROR=KSTORE
GO TO 12
END

ORIGINAL PAGE IS
OF POOR QUALITY

```

1      SUBROUTINE TRAJIN
      REAL MSTART,MEND,NT,MENDUR
      COMMON /TRAJCM/ ALPHA,ARW,CD,CDL,CDQ,CL,DESLF,DRAG,EN,HN,KP,
2      RLD,SMN,QMAX,RANGE,SFC,SWING,THRUST,TW,ULTLF,W,WAMMUN,WETANK,
5      WFUEL,WGTOWT,WMISS,WTOT,WPLVT,MSTART(12),HSTART(12),TIM(12),
3      PSIGT(12),TOOTST(12),NZST(12),PSIT(12),TOOTIT(12),NZIT(12),
4      WBOMBS,WFEKT,SFCU,THRSTU,CDINSP,PRTOT,WKFUEL,CRMACH,FLTD,X(12),
5      FLLAND,TENDUR,BLRANG,BLTIME,TIMTQ1,TIMTQ2,WFTQ,PNAME1(12),
6      PNAME2(12),SMNT(12),HNT(12),WFT(12),TIMET(12),XT(12),CLT(12),
10     CDT(12),ALPHAT(12),CLIT(12),CDIT(12),ALPHIT(12),WFUSED,WLAND,
8     QT(12),RLDT(12),SFCT(12),TNT(12),CET(12),DY(18),
9     IAD,IPS,ITS,IWS,IBS,NPHASE,IDY(14)
      COMMON /TRAJEX/ ALPGT(12),CDINST(12),CD1GT(12),CL1GT(12),DECEL,
15     DELHP,DELMP,DXCRUS,FLFAC,FRFURE,HEND(12),HENDUR,HMAXP,HMINP,
2     HPATH(50),MEND(12),MENDUR,NT(12),PRTOT(12),RADIT(12),
3     RADST(12),SF,SFCUT(12),SMHXP,SMHINP,
4     TNUT(12),VIND(12),VELT(12),VPATH(50),WART(12),WCOMBP,WEEIT(12),
5     WFRES,WFTUT,WFTQ1,WFTQ2,WFTQ3,WKLAND,XDESC,XGRLAN,FMGMAX,TOL,
6     WFIXED,WGCALC,WPL,WGTQ,DUMY(82),IA(12),IAS,IB(12),IBREG,IENDUR,
20     IMISS,IP(12),IPHASE,IPLT,IPRINT,IPRT(12),IPSIZE,IPSTQ1,IPSTQ2,
8     IW(12),IX(12),KERROR,MILCOM,NCRUSE,NLEGCL,NLEGCR,NLEGLO,NMISS,
9     NPATH,IDUMY(6)
      DATA PATH/4HPATH/
      NAMELIST/TRAJDATA/TIMTQ1,TIMTQ2,FRFURE,DESLF,ULTLF,RANGE,WFUEL,
25     WFEKT,WFTQ3,QMAX,XDESC,WKFUEL,CRMACH,WKLAND,FLFAC,DECEL,NCRUSE,
2     IPSIZE,IPSTQ1,IPSTQ2,IBREG,IENDUR,IPRINT,KERROR,IPLT,HMINP,
3     HMAXP,DELHP,SMHINP,SMHXP,DELMP,WCOMBP,MILCOM,MENDUR,NLEGCL,
4     NLEGCR,NLEGLO,NMISS,FMGMAX,TOL
      READ (5,TRAJDATA)
30     REWIND 22
      DO 10 I=1,NMISS
      READ (5,500) NPHASE,WPLTRA
      WRITE (22,2200) NPHASE,WPLTRA
500     FORMAT(I10,E10.0)
35     2200    FORMAT(I10,E15.8)
      C READ IN THE FOLLOWING WORDS FOR EACH PHASE
      C PHASE(2 PARTS), STARTING MACH, ENDING MACH, STARTING ALTITUDE,
      C ENDING ALTITUDE, HORIZONTAL DISTANCE, TIME, NUMBER OF TURNS,
      C CONSTANT INDICATED AIRSPEED CLIMB, POWER SETTING, RANGE INCREMENT
40     C INDICATOR, WEAPONS RELEASE INDICATOR, BOMBS RELEASE INDICATOR,
      C AMMO RELEASE INDICATOR.
      DO 10 IPHASE=1,NPHASE
      READ (5,501) PNAME1(IPHASE),PNAME2(IPHASE),MSTART(IPHASE),
45     1 MEND(IPHASE),HSTART(IPHASE),HEND(IPHASE),X(IPHASE),TIM(IPHASE),
2     NT(IPHASE),VIND(IPHASE),IP(IPHASE),IX(IPHASE),IW(IPHASE),
3     IPRT(IPHASE),IB(IPHASE),IA(IPHASE)
501     FORMAT(2A4,2X,7E10.0/E10.0,6I10)
      WRITE (22,2201) PNAME1(IPHASE),PNAME2(IPHASE),MSTART(IPHASE),
50     1 MEND(IPHASE),HSTART(IPHASE),HEND(IPHASE),X(IPHASE),TIM(IPHASE),
2     NT(IPHASE),VIND(IPHASE),IP(IPHASE),IX(IPHASE),IW(IPHASE),
3     IPRT(IPHASE),IB(IPHASE),IA(IPHASE)
2201    FORMAT(2A4,2X,7E15.8/E15.8,6I10)
      IF (PNAME1(IPHASE).NE.PATH) GO TO 10
      READ (5,500) NPATH
55     READ (5,502) (HPATH(I),I=1,NPATH)
      READ (5,502) (VPATH(I),I=1,NPATH)
502     FORMAT(8E10.0)

```

SUBROUTINE TRAJIN

76/76 OPT=2

FTN 4.5+410

6C

```
2202 WRITE (22,2200) NPATH
10    WRITE (22,2202) (HPATHT(I),I=1,NPATH)
      WRITE (22,2202) (VPATHT(I),I=1,NPATH)
      FORMAT(8E15.8)
      CONTINUE
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```

1      SUBROUTINE TRAJOI .
C      PRINTOUT OF TRAJECTORY INPUT
      REAL MSTART,MEND,NT,MENDUR
      COMMON /TRAJCM/ ALPHA,ARW,CD,CDL,CDD,CL,DESLF,DRAG,EN,HN,KP,
5      1 RLD,SMN,QMAX,RANGE,SFC,SWING,THRUST,TW,ULTLF,W,WAMMUN,WETANK,
      2 WFUEL,WGTOWT,WMISS,WTOT,WPLWT,MSTART(12),HSTART(12),TIM(12),
      3 PSIGT(12),TDOTST(12),NZST(12),PSIT(12),TDOTIT(12),NZIT(12),
      4 WBOMBS,WFEKT,SFCU,THRSTU,CDINSP,PRTOT,WKFUEL,CRMACH,FLTO,X(12),
10     5 FLLAND,TENDUR,BLRANG,BLTIME,TIMTO1,TIMTO2,WFTO,PNAME1(12),
      6 PNAME2(12),SMNT(12),HNT(12),WFT(12),TIMET(12),XT(12),CLT(12),
      7 CDT(12),ALPHAT(12),CLIT(12),CDIT(12),ALPHIT(12),WFUSED,WLAND,
      8 QT(12),RLDT(12),SFCT(12),TNT(12),CET(12),DY(18),
      9 IAO,IPS,ITS,IWS,IBS,NPHASE,IDY(14)
      COMMON /TRAJEX/ ALPGT(12),CDINST(12),CD1GT(12),CL1GT(12),DECEL,
15     1 DELHP,DELMP,DXCRUS,FLFAC,FRFURE,HEND(12),HENDUR,HMAXP,HMINP,
      2 HPATHT(50),MEND(12),MENDUR,NT(12),PRTOTT(12),RADIT(12),
      3 RADST(12),SF,SFCUT(12),SMMAXP,SMMINP,
      4 TNUT(12),VIND(12),VELT(12),VPATHT(50),WART(12),WCOMBP,WEELT(12),
20     5 WFRES,WFTOT,WFTO1,WFTO2,WFTRAP,WKLAND,XDESC,XGRLAN,FWGMAX,TOL,
      6 WFIXED,WGALC,WPL,WGTO,DUMY(82),IA(12),IAS,IB(12),IBREG,IENDUR,
      7 IMISS,IP(12),IPHASE,IPLT,IPRINT,IPRT(12),IPSIZE,IPSTO1,IPSTO2,
      8 IW(12),IX(12),KERROR,MILCOM,NCRUSE,NLEGCL,NLEGCR,NLEGLO,NMISS,
      9 NPATH,IDUMY(6)
      DATA PATH/4HPATH/
25     WRITE (6,600) TIMTO1,MENDUR,NCRUSE,IPLT,TIMTO2,QMAX,IPSIZE,HMINP,
      1 FRFURE,XDESC,IPSTO1,HMAXP,DESLF,WKFUEL,IPSTO2,DELHP,ULTLF,
      2 CRMACH,IBREG,SMMINP,RANGE,WKLAND,IENDUR,SMMAXP,WFUEL,FLFAC,
      3 IPRINT,DELMP,WFEKT,DECEL,KERROR,WCOMBP,WFTRAP,NLEGCL,NLEGCR,
30     4 NLEGLO,FWGMAX,TOL,MILCOM,NMISS
600    FORMAT(1H1,25X,16HTRAJECTORY INPUT//
      1 1X,8HTIMTO1 =,F4.1, 7X,8HMENDUR =,F6.0, 4X,8HNCRUSE =,I2,
      2 3X,8H IPLT =,I2 / 1X,8HTIMTO2 =,F4.1, 7X,8H QMAX =,F7.0,
      3 3X,8H IPSIZE =,I2 , 3X,8H HMINP =,F7.0/ 1X,8HFRFURE =,F5.2,
      4 6X,8H XDESC =,F6.1, 4X,8HIPSTO1 =,I2 , 3X,8H HMAXP =,F7.0/
35     5 1X,8H DESLF =,F5.2, 6X,8HWKFUEL =,F6.3, 4X,8HIPSTO2 =,I2,
      6 3X,8H DELHP =,F7.0/ 1X,8H ULTLF =,F5.2, 6X,8HCRMACH =,F6.3,
      7 4X,8H IBREG =,I2 , 3X,8HSMMINP =,F6.3/ 1X,8H RANGE =,F7.0,
      8 4X,8HWKLAND =,F6.3, 4X,8HIENDUR =,I2 , 3X,8HSMMAXP =,F6.3/
      9 1X,8H WFUEL =,F8.0, 3X,8H FLFAC =,F6.3, 4X,8HIPRINT =,I2,
40     1 3X,8H DELMP =,F6.3/ 1X,8H WFEKT =,F8.0, 3X,8H DECEL =,F6.3,
      2 4X,8HKERROR =,I2 , 3X,8HWCOMBP =,F9.2/ 1X,8HWFTRAP =,F8.0,
      3 3X,8HNLEGCL =,I2 , 8X,8HNLEGCR =,I2 , 3X,8HNLEGLO =,I2/
      4 1X,8HFWGMAX =,F6.3, 5X,8H TOL =,F6.3, 4X,8HMILCOM =,I2,
      5 3X,8H NMISS =,I2)
45     REWIND 22
      DO 10 IMISS=1,NMISS
      READ (22,2200) NPHASE,WPLTRA
2200    FORMAT(I10,E15.8)
      IF (IMISS.EQ.1) WRITE (6,601) IMISS
50     601    FORMAT(/// 24X,7HMISSION,I2//
      1 9X,66HMACH MACH ALT ALT HORIZ NO. IP
      2 1X IW/75H PHASE START END START END DIST TIME TURNS
      3 VIND IPRT IB IA/)
      IF (IMISS.GT.1) WRITE (6,602) IMISS,WPLTRA
55     602    FORMAT(/// 24X,7HMISSION,I2,12H (PAYLOAD =,F8.0,4H LB)//
      1 9X,66HMACH MACH ALT ALT HORIZ NO. IP
      2 1X IW/75H PHASE START END START END DIST TIME TURNS

```

```
      3 VIND IPRT IB IA/)
      DO 1C IPHASE=1,NPHASE
60      READ (22,2201) PNAME1(IPHASE),PNAME2(IPHASE),MSTART(IPHASE),
      1 MEND(IPHASE),HSTART(IPHASE),HEND(IPHASE),X(IPHASE),TIM(IPHASE),
      2 NT(IPHASE),VIND(IPHASE),IP(IPHASE),IX(IPHASE),IW(IPHASE),
      3 IPRT(IPHASE),IB(IPHASE),IA(IPHASE)
      2201 FORMAT(2A4,2X,7E15.8/E15.8,6I10)
65      WRITE (6,603) PNAME1(IPHASE),PNAME2(IPHASE),MSTART(IPHASE),
      1 MEND(IPHASE),HSTART(IPHASE),HEND(IPHASE),X(IPHASE),TIM(IPHASE),
      2 NT(IPHASE),VIND(IPHASE),IP(IPHASE),IX(IPHASE),IW(IPHASE),
      3 IPRT(IPHASE),IB(IPHASE),IA(IPHASE)
      603 FORMAT(1X,A4,A3,F5.2,F6.2,2F8.0,F7.1,F6.1,F6.1,F7.1,I4,I5,I4/
70      1 I65,I5,I4)
      IF (PNAME1(IPHASE).NE.PATH) GO TO 10
      READ (22,2200) NPATH
      READ (22,2202) (HPATHT(I),I=1,NPATH)
      READ (22,2202) (VPATHT(I),I=1,NPATH)
75      2202 FORMAT(8E15.8)
      WRITE (6,604) NPATH
      WRITE (6,605) (HPATHT(I),I=1,NPATH)
      WRITE (6,606) (VPATHT(I),I=1,NPATH)
      WRITE (6,607)
80      604 FORMAT(9X,8H NPATH =,I3)
      605 FORMAT(9X,8HHPATHT =,7F8.0/(17X,7F8.0))
      606 FORMAT(9X,8HVPATHT =,7F8.0/(17X,7F8.0))
      607 FORMAT(/)
      10 CONTINUE
85      RETURN
      END
```

ORIGINAL PAGE :
OF POOR QUALITY

```

1      SUBROUTINE MISSION(ICALC,NERROR,IGED,KGPRNT)
      REAL LOIT,MENDUR
      COMMON /TRAJCH/ ALPHA,ARW,CD,CDL,COO,CL,DESLF,DRAG,EN,HN,KP,
5      1 RLD,SMN,QMAX,RANGE,SFC,SWING,THRUST,TW,ULTLF,W,WAMMUN,WETANK,
      2 WFUEL,WGTGWT,WMISS,WTOT,WPLWT,MSTART(12),HSTART(12),TIM(12),
      3 PSIGT(12),TDOYST(12),NZST(12),PSIT(12),TDOIT(12),NZIT(12),
      4 WBOMBBS,WFEET,SFCU,THRSTU,CDINSP,PRTOT,WKFUEL,CRMACH,FLTO,X(12),
      5 FLLAND,TENDUR,BLRANG,BLTIME,TIMTO1,TIMTO2,WFTO,PNAME1(12),
10     6 PNAME2(12),SMNT(12),HNT(12),WFT(12),TIMET(12),XT(12),CLT(12),
      7 CDT(12),ALPHAT(12),CLIT(12),CDIT(12),ALPHIT(12),WFUSED,WLAND,
      8 QT(12),RLDT(12),SFCT(12),TNT(12),CET(12),DY(18),
      9 IAO,IPS,ITS,IWS,IBS,NPHASE,IDY(14)
      COMMON /TRAJEX/ ALPIGT(12),CDINST(12),CD1GT(12),CL1GT(12),DECEL,
15     1 DELHP,DELMF,DXCRUS,FLFAC,FRFURE,HEND(12),HENDUR,HMAXP,HMINP,
      2 HPATHT(50),HEND(12),MENDUR,NT(12),PRTOTT(12),RADIT(12),
      3 RADST(12),SF,SFCUT(12),SMMAXP,SMMINP,
      4 TNUT(12),VIND(12),VELT(12),VPATHT(50),WART(12),WCOMBP,WEELT(12),
      5 WFRES,WFTOT,WFTO1,WFTO2,WFTRAP,WKLAND,XDESC,XGRLAN,FWGMAX,TOL,
      6 WFIXED,WGCALC,WPL,WGYO,DUMY(82),IA(12),IAS,IB(12),IBREG,IENDUR,
20     7 IMISS,IP(12),IPHASE,IPLT,IPRINT,IPRT(12),IPSIZE,IPSTO1,IPSTO2,
      8 IW(12),IX(12),KERROR,MILCOM,NCRUSE,NLEGCL,NLEGCR,NLEGLO,NMISS,
      9 NPATH,IDUMY(6)
      DATA CLIM/4HCLIM/
      DATA CRUI/4HCRUI/
25     DATA ACCE/4HACCE/
      DATA COMB/4HCOMB/
      DATA LOIT/4HLOIT/
      DATA DESC/4HDESC/
      DATA PATH/4HPATH/
30     C SET ULTLF SO AS TO MAINTAIN RATIO BETWEEN ULTLF AND DESLF
20     ULTLF=SF*DESLF
      C SET WEAPONS, BOMBS, AMMO, AND TANKS DROP INDICATORS TO 1.
      C THIS WILL CAUSE AERO TO CALCULATE DRAG INCREMENTS
      C FOR WEAPONS, BOMBS, AND TANKS UNTIL SOME TRAJECTORY PHASE
35     C SETS THEM TO ZERO. IF THE AMMO INDICATOR IS CHANGED
      C TO ZERO BY SOME TRAJECTORY PHASE, THIS WILL AFFECT
      C THE CALCULATION OF WFUSED.
      IWS=1
      IBS=1
40     IAS=1
      ITS=1
      C SET SUM OF HORIZONTAL DISTANCES COVERED DURING CLIMB AND
      C ACCELERATION PHASES TO ZERO AT BEGINNING OF MISSION. NEW VALUE
      C WILL BE CALCULATED IN EACH CLIMB AND ACCELERATION PHASE AND
45     C WILL BE SET BACK TO ZERO AT END OF EACH CRUISE PHASE.
      DXCRUS=0.0
      C SET BLOCK TIME AND BLOCK RANGE TO ZERO SO THAT THEY CAN BE
      C ACCUMULATED IN EACH PHASE.
      BLTIME=0.0
50     BLRANG=0.0
      CALL TAKEOF(ICALC,NERROR,IGED,KGPRNT)
      IF (NERROR.GE.2) GO TO 160
      IF (W.LT.WPL) GO TO 130
      C-----PERFORM ALL PHASES OF MISSION IN THE ORDER THEY WERE READ IN.
55     DO 120 IPHASE=1,NPHASE
      IF (PNAME1(IPHASE).EQ.CLIM) GO TO 40
      IF (PNAME1(IPHASE).EQ.CRUI) GO TO 50

```



```

        IF (PNAME1(IPHASE).EQ.ACCE) GO TO 60
        IF (PNAME1(IPHASE).EQ.COMB) GO TO 70
60      IF (PNAME1(IPHASE).EQ.LOIT) GO TO 80
        IF (PNAME1(IPHASE).EQ.DESC) GO TO 90
        IF (PNAME1(IPHASE).EQ.PATH) GO TO 100
        WRITE (6,601) IPHASE
601     FORMAT(/47H TRAJECTORY INPUT USES UNDEFINED NAME FOR PHASE,I3/)
        GO TO 160
65      40  CALL CLIMB(ICALC,NERROR,IGED,KGPRNT)
        GO TO 110
        50  CALL CRUISE(ICALC,NERROR,IGED,KGPRNT)
        GO TO 110
70      60  CALL ACCEL(ICALC,NERROR,IGED,KGPRNT)
        GO TO 110
        70  CALL COMBAT(ICALC,NERROR,IGED,KGPRNT)
        GO TO 110
75      80  CALL LOITER(ICALC,NERROR,IGED,KGPRNT)
        GO TO 110
        90  CALL DESCNT(ICALC,NERROR,IGED,KGPRNT)
        GO TO 110
        100 CALL PATHS(ICALC,NERROR,IGED,KGPRNT)
        110 IF (NERROR.GE.2) GO TO 160
        IF (W.LT.WPL) GO TO 130
80      120 CONTINUE
        C-----COMPUTE USED, RESERVE, TOTAL, AND INTERNAL FUEL WEIGHT.
        130 WFUSED=WGTO-W
        IF (IWS.EQ.0) WFUSED=WFUSED-WMISS
85      IF (IBS.EQ.0) WFUSED=WFUSED-WBOMBS
        IF (IAS.EQ.0) WFUSED=WFUSED-WAMMUN
        IF (ITS.EQ.C) WFUSED=WFUSED-WETANK
        WFUSED=WKFUEL*WFUSED
        WFRES=WFUSED/(1.-FRFURE)-WFUSED
90      WFTLT=WFUSED+WFRES+WFTAP
        WFUEL=WFTOT-WFEXT
        IF (W.GE.WPL) GO TO 140
        IF (KERROR.EQ.2) WRITE (6,602)
602     FORMAT(/37H FUEL WEIGHT EXCEEDS AVAILABLE WEIGHT/)
95      WFUEL=WGTO-WPL
        GO TO 160
        C-----CALCULATE LANDING FIELD LENGTH.
        140 RHO=.00238
        IAG=12
100     CALL TRAJ01(2)
        CALL AERO(ICALC,NERROR,IGED,KGPRNT)
        IF (NERROR.GE.2) GO TO 160
        CALL TRAJC1(1)
        WLAND=WGTO-WKLAND*WFUSED
105     VSTALL=SQRT(2.*WLAND/(SWING*CL*RHO))
        VSCREN=1.3*VSTALL
        VTDOWN=1.15*VSTALL
        XAIR=((VSCREN**2-VTDOWN**2)/(64.4+50.))*RLD
        XGRLAN=VTDOWN**2/(64.4*DECEL)
110     FLLAND=(XAIR+XGRLAN)/FLFAC
        C-----CALCULATE ENDURANCE ALTITUDE AND TIME.
        SMN=MENDUR
        CALL GOLDEN(2000.,55000.,2,12,XX,Y,NERROR,IGED,KGPRNT)
        IF (NERROR.GE.2) GO TO 160

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
115      HENDUR=XX
      TENDUR=Y*ALOG(WGTO/(WGTO-WFUSED))
C-----COMPUTE GROSS WEIGHT.
      WGCALC=WFIXED+WPL+WFTOT
      IF (IPRINT.EQ.0) GO TO 160
120      WRITE (6,603) WGTO,WFUSED,WFRES,WFTOT,WFUEL,WPL,W,
1      WLAND,XGRLAN,FLLAND,HENDUR,TENDUR,WGCALC
603      FORMAT(///7X,7HLANDING///3X,6HWGTO =E14.7,1X,8HWFUSED =E14.7,
1 2X,7HWFRES =E14.7,2X,7HWFOT =E14.7,2X,7HWFUEL =E14.7/
2 4X,5HWPL =E14.7,6X,3HW =E14.7,2X,7HWLAND =E14.7,
125 3 1X,8HXGRLAN =E14.7,1X,8HFLLAND =E14.7/1X,8HHENDUR =E14.7,
4 1X,8HTENDUR =E14.7,1X,8HWGCALC =E14.7/)
160      RETURN
      END
```

SUBROUTINE TAKEOF

76/76 DPT=2

FTN 4.5+410

05

```

1      SUBROUTINE TAKEOF(ICALC,NERROR,IGEO,KGPRNT)
      COMMON /TRAJCM/ ALPHA,ARW,CD,CDL,CDO,CL,DESLF,DRAG,EN,HN,KP,
2      RLD,SMN,QMAX,RANGE,SFC,SWING,THRUST,TW,ULTLF,W,WAMMUN,WETANK,
5      WFUEL,WGTDWT,WMISS,WTOT,WPLWT,HSTART(12),HSTART(12),TIM(12),
      PS1GT(12),TDOTST(12),NZST(12),PSIT(12),TDOTIT(12),NZIT(12),
      WBOMBS,WFEET,SFCU,THRSTU,CDINSP,PRTOT,WKFUEL,CRMACH,FLTO,X(12),
      FLLAND,TENDUR,BLRANG,BLTIME,TIMT01,TIMT02,WFTD,PNAME1(12),
      PNAME2(12),SMNT(12),HNT(12),WFT(12),TIMET(12),XT(12),CLT(12),
      CDT(12),ALPHAT(12),CLIT(12),CDIT(12),ALPHIT(12),WFUSED,WLAND,
10     QT(12),RLDT(12),SFCT(12),TNT(12),CET(12),DY(18),
      IAD,IPS,ITS,IWS,IBS,NPHASE,IDY(14)
      COMMON /TRAJEX/ ALP1GT(12),CDINST(12),CD1GT(12),CL1GT(12),DECEL,
      DELHP,DELMP,DXCRUS,FLFAC,FRFURE,HEND(12),HENDUR,HMAXP,HMINP,
      HPAHT(50),MEND(12),MENDUR,NT(12),PRTOTT(12),RADIT(12),
15     RADST(12),SF,SFCUT(12),SMHXP,SMHINP,
      TNUT(12),VIND(12),VELT(12),VPATHT(50),WARY(12),WCOMBP,WEELT(12),
      WFRES,WFTOT,WFT01,WFT02,WFTAP,WKLAND,XDESC,XGRLAN,FWGMAX,TOL,
      WFIXED,WGCALC,WPL,WGTD,DUMY(82),IA(12),IAS,IB(12),IBREG,IENDUR,
      IMISS,IP(12),IPHASE,IPLOT,IPRINT,IPRT(12),IPSIZE,IPST01,IPST02,
20     IW(12),IX(12),KERROR,MILCOM,NCRUSE,NLEGCL,NLEGCR,NLEGLO,NMISS,
      NPATH,IDUMY(6)
      DIMENSION ANS(4)
      C-----CALL PROP TO SIZE ENGINE.
      IPS=IPSIZE
25     HN=0.
      SMN=0.
      CALL TRAJ01(2)
      CALL PROP(ICALC,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
30     C-----CALCULATE STARTUP AND TAXI FUEL USED.
      IPS=IPST01
      CALL TRAJ01(2)
      CALL PROP(ICALC,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
      TN=THRUST*EN
      WFT01=.0167*TIMT01*SFC*TN
40     C-----BEGIN FIELD LENGTH CALCULATION.
      HNTD=HSTART(1)
      HN=HNTD
      CALL AT62(HN,ANS)
      RHO=ANS(1)
      A=ANS(4)
45     C-----GUESS STALLING MACH NUMBER.
      SMN=.25
      IAD=14
      CALL TRAJ01(2)
      CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
50     CALL TRAJ01(1)
      CLS=CL
      C-----CALCULATE STALLING SPEED.
      VS=SQRT(2.*WGTD/(RHO*CLS*SWING))
      V2=1.2*VS
55     SMN2=V2/A
      SMN=SMN2

```

```

        CALL TRAJ01(2)
        CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
60      IF (NERROR.GE.2) RETURN
        CALL TRAJ01(1)
        CL2=CL
C-----FIND THRUST AT END OF TAKEOFF.
        IPS=IPST02
65      CALL TRAJ01(2)
        CALL PROP(ICALC,NERROR,IGEO,KGPRNT)
        IF (NERROR.GE.2) RETURN
        CALL TRAJ01(1)
        TN=THRUST*EN
70      TN2=TN
        SFC2=SFC
C-----FIND STATIC THRUST.
        SMN=0.
        CALL TRAJ01(2)
75      CALL PROP(ICALC,NERROR,IGEO,KGPRNT)
        IF (NERROR.GE.2) RETURN
        CALL TRAJ01(1)
        TN=THRUST*EN
        TNO=TN
80      SFC0=SFC
C-----CALCULATE AVERAGE THRUST AND SFC DURING TAKEOFF.
        TNAVE=.5*(TNO+TN2)
        SFCAVE=.5*(SFC0+SFC2)
C-----CALCULATE TAKEOFF FIELD LENGTH REQUIRED.
85      IF (MILCOM.EQ.0) TDOBHT=50.0
        IF (MILCOM.NE.0) TDOBHT=35.0
        FLTD=2.10*(.374*TDOBHT+.01163*WGTO/(RHO*CL2*SWING))*
        1 (2.7+1.0/(TNAVE/WGTO-.04))+32.0/SQRT(RHO)
C-----CALCULATE TAKEOFF FUEL USED.
90      IF (TIMT02.LT.0.0) GO TO 10
        IF (TIMT02.EQ.0.0) GO TO 20
        HN=0.
        IPS=IPST02
        CALL TRAJ01(2)
95      CALL PROP(ICALC,NERROR,IGEO,KGPRNT)
        IF (NERROR.GE.2) RETURN
        CALL TRAJ01(1)
        TN=THRUST*EN
        WFT02=.0167*TIMT02*SFC*TN
100     GO TO 30
10      WFT02=TNAVE*SFC*FLTD/(1800.*V2)
        GO TO 30
20      WFT02=0.0
C-----CALCULATE TOTAL FUEL USED FOR STARTUP, TAXI, AND TAKEOFF.
105     WFT0=WFT01+WFT02
        W=WGTO-WFT0
        IF (IPRINT.EQ.0) RETURN
        WRITE (6,600) WGTO,WFT01,WFT02,WFT0,W,HNT0,CLS,VS,V2,SMN2,CL2,
110     600  1 TN2,SFC2,TNO,SFC0,TNAVE,SFCAVE,FLTD
        FORMAT(1H1,6X,7HTAKEOFF///3X,6HWGTO =E14.7,2X,7HWFT01 =E14.7,
        1 2X,7HWFT02 =E14.7,3X,6HWFT0 =E14.7,6X,3HW =E14.7/
        2 3X,6HHNT0 =E14.7,4X,5HCLS =E14.7,5X,4HVS =E14.7,5X,4HV2 =E14.7,
        3 3X,6HSMN2 =E14.7,4X,5HCL2 =E14.7,4X,5HTN2 =E14.7,
        4 3X,6HSFC2 =E14.7,4X,5HTNO =E14.7,3X,6HSFC0 =E14.7/
115     5 2X,7HTNAVE =E14.7,1X,8HSFCAVE =E14.7,3X,6HFLTD =E14.7/)
        RETURN
        END

```

SUBROUTINE CLIMB

76/76 OPT=2

FTN 4.5+410

C

```

1      SUBROUTINE CLIMB(ICALC,NERROR,IGEO,KGPRNT)
      REAL MSTART,MEND
      COMMON /TRAJCH/ ALPHA,ARW,CD,CDL,CDD,CL,COESLF,DRAG,EN,HN,KP,
5      1 RLD,SMN,QMAX,RANGE,SFC,SWING,THRUST,TW,ULTLF,W,WAMHUN,WETANK,
      2 WFUEL,WGTOWT,WMISS,WTDI,WPLWT,MSTART(12),HSTART(12),TIM(12),
      3 PSIGT(12),TDOST(12),NZST(12),PSIT(12),TDOIT(12),NZIT(12),
      4 WBOMBS,WFEXT,SFCU,THRSTU,CDINSP,PRTOT,WKFUEL,CRMACH,FLTO,X(12),
      5 FLLAND,TENDUR,BLRANG,BLTIME,TIMTO1,TIMTO2,WFTO,PNAME1(12),
      6 PNAME2(12),SMNT(12),HNT(12),WFT(12),TIMET(12),XT(12),CLT(12),
10     7 CDT(12),ALPHAT(12),CLIT(12),CDIT(12),ALPHIT(12),WFUSED,WLAND,
      8 QT(12),RLDT(12),SFCT(12),TNT(12),CET(12),DY(18),
      9 IAO,IPS,ITS,IWS,IBS,NPHASE,IDY(14)
      COMMON /TRAJEX/ ALPIGT(12),CDINST(12),CDIGT(12),CLIGT(12),DECEL,
15     1 DELHP,DELMP,DXCRUS,FLFAC,FRFURE,HEND(12),HENDUR,HMAXP,HMINP,
      2 HPATH(50),MEND(12),MENDUR,NT(12),PRTOT(12),RADIT(12),
      3 RADST(12),SF,SFCUT(12),SMHXP,SMHNP,
      4 TNUT(12),VIND(12),VELT(12),VPATH(50),WART(12),WCOMBP,WEELT(12),
      5 WFRES,WFTOT,WFTO1,WFTO2,WFTAP,WKLAND,XDESC,XGRLAN,FWGMAX,TOL,
      6 WFIXED,WGCALC,WPL,WGTO,DUMY(82),IA(12),IAS,IB(12),IBREG,IENDUR,
20     7 IMISS,IP(12),IPHASE,IPLDT,IPRINT,IPRT(12),IPSIZE,IPSTO1,IPSTO2,
      8 IW(12),IX(12),KERROR,MILCOM,NCRUSE,NLEGCL,NLEGCR,NLEGLO,NMISS,
      9 NPATH,IDUMY(6)
      DIMENSION ANS(4)
      IF (IPRT(IPHASE).NE.0) WRITE (6,600)
25     600 FORMAT(///18H CLIMB PHASE//)
      C-----FIX STARTING ALTITUDE
      IF (HSTART(IPHASE).GE.0.0) HNINT=HSTART(IPHASE)
      IF (HSTART(IPHASE).LT.0.0) HNINT=HNT(IPHASE-1)
      C-----FIX ENDING ALTITUDE
      HNFN=HEND(IPHASE)
      IF ((IBREG.EQ.0).OR.(HEND(IPHASE).GT.0.0)) GO TO 40
      SMN=HEND(IPHASE)
      CALL GOLDEN(2000.,60000.,1,12,XX,Y,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
35     HNFN=XX
      40 SMN=MSTART(IPHASE)
      IF (MSTART(IPHASE).LT.0.0) SMN=SMNT(IPHASE-1)
      IF (HNFN.LT.80000.) GO TO 43
      NLEGS=20
40     GO TO 45
      43 NLEGS=20.*ALOG((4000.-.05*HNINT)/(4000.-.05*HNFN))+.5
      IF (NLEGS.LT.3) NLEGS=3
      IF (NLEGS.GT.20) NLEGS=20
45     IF (NLEGCL.GT.0) NLEGS=NLEGCL
45     NLEGS1=NLEGS+1
      B1=HNFN-HNINT
      B2=1.718282/FLOAT(NLEGS)
      IF (B1.GT.0.0) GO TO 50
      SMNT(IPHASE)=SMN
50     HNT(IPHASE)=HNFN
      CLT(IPHASE)=0.0
      ALPHAT(IPHASE)=0.0
      WFT(IPHASE)=0.0
      TIMET(IPHASE)=0.0
55     SFCT(IPHASE)=0.0
      TNT(IPHASE)=0.0
      CDT(IPHASE)=0.0

```

```

        RLDT(IPHASE)=0.0
        WEELT(IPHASE)=W
60      WART(IPHASE)=0.0
        RETURN
50      HN=HNINT
        TIME=0.0
        WF=0.0
65      XCLIMB=0.0
        VINDFS=0.0
        IPS=IP(IPHASE)
        IAO=8
        IF (VIND(IPHASE).LE.0.) GO TO 60
70      C----- INITIALIZE CONDITIONS FOR CONSTANT INDICATED AIRSPEED CLIMB.
        IF (IPRT(IPHASE).NE.0) WRITE (6,601)
601     FORMAT(/34H CONSTANT INDICATED AIRSPEED CLIMB//)
        IF (MSTART(IPHASE).GE.0.) VINDFS=1.689*VIND(IPHASE)
        IF (MSTART(IPHASE).LT.0.) VINDFS=VELT(IPHASE-1)
75      CALL AT62(HN,ANS)
        RHOP=ANS(1)
        CL=2.*W/(.002378*VINDFS**2*SWING)
        CALL TRAJ01(2)
        CALL AERD(ICALC,NERROR,IGEO,KGPRNT)
80      IF (NERROR.GE.2) RETURN
        CALL TRAJ01(1)
        CALL TRAJ01(2)
        CALL PROP(ICALC,NERROR,IGEO,KGPRNT)
        IF (NERROR.GE.2) RETURN
85      CALL TRAJ01(1)
        TN=THRUST*EN
        TNU=THRSTU*EN
60      IF (IPRT(IPHASE).EQ.0) GO TO 70
        WRITE (6,602) W,HNINT,HNFIN,SMN,HN
90      602     FORMAT(/6X,3HW =E14.7,2X,7HHNINT =E14.7,2X,7HHNFIN =E14.7,
70      1 4X,5HSMN =E14.7,5X,4HHN =E14.7/)
        DO 140 LEG=1,NLEGS1
        IF (LEG.LT.NLEGS1) GO TO 80
        IF (IBREG.EQ.0.OR.HEND(IPHASE).GT.0.) GO TO 140
95      HN1=HN
        CL1=CL
        ALPHA1=ALPHA
        SFC1=SFC
        TN1=TN
100     CD1=CD
        WTOT1=WTOT
        HLOW=HN1
        HHIGH=HN1+3000.
        SMNSAV=SMN
105     SMN=AMIN1(SMN,HEND(IPHASE))
        CALL GOLDEN(HLOW,HHIGH,1,8,XX,Y,NERROR,IGEO,KGPRNT)
        SMN=SMNSAV
        HNFIN=XX
        DELH=HNFIN-HN1
110     HN=HN1
        CL=CL1
        ALPHA=ALPHA1
        SFC=SFC1
        TN=TN1

```

SUBROUTINE CLIMB

76/76 OPT=2

FTN 4.5+410

0

```

115      CD=CD1
          WTOT=WTOT1
          IF (NERROR.LT.2.AND.DELH.GT.0.005*HN) GO TO 81
          NERROR=0
          GO TO 140
120      80  DELH=B1*ALOG((1.+FLOAT(LEG)*B2)/(1.+FLOAT(LEG-1)*B2))
          81  IF (VINDFS.GT.0.) GO TO 120
          CALL AT62(HN,ANS)
          RHO=ANS(1)
          A=ANS(4)
125      IF (LEG.EQ.1) VEL=SMN*A
          IF (LEG.GT.1) SMN=VEL/A
          IF (SMN.LT.0.0) NERROR=2
          IF (SMN.LT.0.0) WRITE (6,610) IPHASE,LEG
610      FORMAT(/////34H *****FATAL ERROR IN CLIMB. PHASE=,I2,7H    LEG=,I2/
130      1 27H *****NEGATIVE MACH NUMBER.//////)
          IF (SMN.LT.0.0) RETURN
          Q=.5*RHO*VEL**2
          QS=Q*SWING
          CALL TRAJ01(2)
135      CALL PROP(ICALC,NERROR,IGEO,KGPRNT)
          IF (NERROR.GE.2) RETURN
          CALL TRAJ01(1)
          TN=THRUST*EN
          TNU=THRSTU*EN
140      IF (TN/W.LT.1.5) GO TO 90
          C-----EXECUTE FOLLOWING SECTION ONLY IF TN/W .GE. 1.5
          SING=1.0
          IAO=0
          CALL TRAJ01(2)
145      CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
          IF (NERROR.GE.2) RETURN
          CALL TRAJ01(1)
          CD=CDO
          GO TO 100
150      C-----EXECUTE FOLLOWING SECTION ONLY IF TN/W .LT. 1.5
          90  CL=(W/QS)*SQRT(1.-.444*(TN/W)**2)
          IAO=8
          CALL TRAJ01(2)
          CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
155      IF (NERROR.GE.2) RETURN
          CALL TRAJ01(1)
          CL=CL-TN*SIN(ALPHA/57.3)/QS
          CALL TRAJ01(2)
          CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
160      IF (NERROR.GE.2) RETURN
          CALL TRAJ01(1)
          EMAX=.5*SQRT(CL**2/(CDO*(CD-CDO)))
          A1=TN*EMAX/W
          A2=SQRT(A1**2+3.)
165      SING=.8933*TN/W-.167*A2/EMAX-1.5/(EMAX*(A1+A2))
          IF(SING.LE.0.0) SING=0.0
          IF(SING.LE.0.0) GO TO 110
          COSG=SQRT(1.0-SING**2)
          DELT=DELH/(VEL*SING)
170      100 DELT=DELH/(VEL*SING)
          IF (DELT.GT.0.0) GO TO 110
          NERROR=2

```

```

        WRITE (6,603) IPHASE,LEG
603  FORMAT(//////34H *****FATAL ERROR IN CLIMB. PHASE=,I2,7H  LEG=,I2/
1 26H *****NEGATIVE DELTA TIME.//////)
175  RETURN
110  DVDT=32.2*(TN-CD*QS-W*SING)/W
      IF (DVDT.LT.0.) SING=AMAX1(0.,(TN-CD*QS)/W)
      IF (SING.LE.0.) DVDT=32.2*(TN-CD*QS)/W
      DRAG=CD*QS
180  IF (SING.LE.0.0) DELT=60.0
      DELV=DVDT*DELT
      IF (SING.LE.0.0) COSG=1.0
      IF (SING.LE.0.0) DELH=0.0
      DELX=VEL*DELT*COSG/6080.
185  VEL=VEL+DELV
      DELWF=TN*SFC*DELT/3600.
C-----TIME TO CLIMB
      TIME=TIME+DELT/60.
C-----FUEL USED DURING CLIMB
190  WF=WF+DELWF
C-----GROUND DISTANCE COVERED DURING CLIMB
      XCLIMB=XCLIMB+DELX
      HN=HN+DELH
      W=W-DELWF
195  IF (W.GT.WPL) GO TO 115
      IF (KERROR.EQ.2) WRITE (6,605) IPHASE,LEG
605  FORMAT(//26H W.LE.WPL IN CLIMB. PHASE=,I2,7H  LEG=,I2/)
      RETURN
115  IF (IPRT(IPHASE).EQ.0) GO TO 140
      WRITE (6,606) LEG,SMN,HN,RHO,A,VEL,SFC,TN,ALPHA,CL,CDD,CD,EMAX,A1,
1  A2,SING,COSG,DELT,DVDT,DELX,DELWF,W,WF,TIME,XCLIMB
606  FORMAT(/4X,5HLEG =I2/4X,5HSMN =E14.7,5X,4HHN =E14.7,4X,
1 5HRHO =E14.7,6X,3HA =E14.7,4X,5HVEL =E14.7/4X,5HSFC =E14.7,5X,
2 4HTN =E14.7,2X,7HALPHA =E14.7,5X,4HCL =E14.7,4X,5HCDD =E14.7/5X,
3 4HCD =E14.7,3X,6HEMAX =E14.7,5X,4HA1 =E14.7,5X,4HA2 =E14.7,3X,
4 6HSING =E14.7/3X,6HCOSG =E14.7,3X,6HDELT =E14.7,3X,6HDVDT =E14.7,
5 3X,6HDELV =E14.7,3X,6HDELX =E14.7/2X,7HDELWF =E14.7,6X,3HW =,
6  E14.7,5X,4HWF =E14.7,3X,6HTIME =E14.7,2X,7HXCLIMB=E14.7/)
      GO TO 140
110  C-----
C  START OF CONSTANT INDICATED AIRSPEED SECTION
C-----
120  HN=HN+DELH
      CALL AT62(HN,ANS)
115  RHO=ANS(1)
      A=ANS(4)
      C1=.5*(RHO+RHOP)
      C2=SQRT(C1)
      C3=20.51*C2
120  C4=C1*C2
      C5=32.2*C3*DELH/VINDFS-.02438*VINDFS*(RHO-RHOP)/C4
      C6=32.2*(TN/W-.001189*VINDFS**2*CD*SWING/W)
      DELT=C5/C6
      IF (DELT.GT.0.0) GO TO 125
125  KERROR=2
      WRITE (6,607) IPHASE,LEG,DELH,VINDFS,RHO,RHOP,TN,W,CD,SWING,DELT,
1  C1,C2,C3,C4,C5,C6
607  FORMAT(//////34H *****FATAL ERROR IN CLIMB. PHASE=,I2,7H  LEG=,I2/

```



```

230      1 40H *****NEGATIVE DELTA TIME IN VIND CLIMB, /5X, 5HDELH=, E12.5, 2X,
      2 7HVINDFS=, E12.5, 2X, 4HRRHQ=, E12.5/5X, 5HRRHOP=, E12.5, 2X, 3HTN=, E12.5,
      3 2X, 2HW=, E12.5/5X, 3HCD=, E12.5, 2X, 6Hswing=, E12.5, 2X, 5HDELT=, E12.5/
      4 5X, 5HC1-C6/E13.5/E13.5/E13.5/E13.5/E13.5/E13.5/////
      RETURN
125      TIME=TIME+DELT/60.
235      VEL=VINDFS/SQRT(RHO/.002378)
      SMN=VEL/A
      IF (SMN.LT.0.0) NERROR=2
      IF (SMN.LT.0.0) WRITE (6,613) IPHASE,LEG
      IF (SMN.LT.0.0) RETURN
240      SING=DELH/(VEL*DELT)
      IF (SING.GT.1.) SING=1.0
      COSG=SQRT(1.-SING**2)
      DELX=VEL*DELT*COSG/6080.
      XCLIMB=XCLIMB+DELX
245      CALL TRAJ01(2)
      CALL PROP(ICALC, NERROR, IGEO, KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
      TN=THRUST*EN
250      TNU=THRSTU*EN
      DELWF=TN*SFC*DELT/3600.
      WF=WF+DELWF
      W=W-DELWF
      IF (W.GT.WPL) GO TO 130
      IF (NERROR.EQ.2) WRITE (6,605) IPHASE,LEG
      RETURN
130      C7=1.-(DELH/(VINDFS*DELT*C3))**2
      IF (C7.LT.0.) C7=0.0
      CL=2.*W*SQRT(C7)/(0.002378*VINDFS**2*SWING)
260      IAQ=8
      CALL TRAJ01(2)
      CALL AERO(ICALC, NERROR, IGEO, KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
      RHOP=RHO
265      IF (IPRT(IPHASE).EQ.0) GO TO 140
      WRITE (6,608) LEG, VINDFS, SMN, HN, DELT, DELX, C6, VEL, DELWF, W, TN, CL, CD
608      FORMAT(/4X, 5HLEG = I2/
270      1 1X, 8HVINDFS =E14.7, 4X, 5HSMN =E14.7, 5X, 4HHN =E14.7,
      2 3X, 6HDELT =E14.7, 3X, 6HDELX =E14.7/5X, 4HC6 =E14.7,
      3 4X, 5HVEL =E14.7, 2X, 7HDELWF =E14.7, 6X, 3HW =E14.7,
      4 5X, 4HTN =E14.7/5X, 4HCL =E14.7, 5X, 4HCD =E14.7/)
      C-----
      C      END OF CONSTANT INDICATED AIRSPEED SECTION
      C-----
275      140  CONTINUE
      CALL AT62(HN,ANS)
      A=ANS(4)
      SMN=VEL/A
280      RHO=ANS(1)
      Q=.5*RHO*VEL**2
      DXCRUS=DXCRUS+XCLIMB*FLOAT(IX(IPHASE))
      BLTIME=BLTIME+TIME/60.
      BLRANG=BLRANG+XCLIMB
285      C-----TEST FOR WEAPONS DROP

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

      IF (IW(IPHASE).EQ.0) GO TO 150
      W=W-WMISS
      IWS=0
      C-----TEST FOR BOMBS DROP
:90    150  IF (IB(IPHASE).EQ.0) GO TO 160
      W=W-WBOMBS
      IBS=0
      C-----TEST FOR AMMO DROP
295    160  IF (IA(IPHASE).EQ.0) GO TO 170
      W=W-WAMMUN
      IAS=0
      170  IF (IPRT(IPHASE).EQ.0) GO TO 180
      WRITE (6,609) SMN,HN,VEL,XCLIMB,DXCRUS,TIME,WF,W
      609  FORMAT(/4X,5HSMN =E14.7,5X,4HHN =E14.7,4X,5HVEL =E14.7,
300    1 1X,8HXCLIMB =E14.7,1X,8HDXCRUS =E14.7/3X,6HTIME =E14.7,
      2 5X,4HWF =E14.7,6X,3HW =E14.7/)
      180  SMNT(IPHASE)=SMN
      HNT(IPHASE)=HN
      CLT(IPHASE)=CL
305    ALPHAT(IPHASE)=ALPHA
      WFT(IPHASE)=WF
      TIMET(IPHASE)=TIME
      VELT(IPHASE)=VEL
      SFCT(IPHASE)=SFC
310    TNT(IPHASE)=TN
      CDT(IPHASE)=CD
      RLDI(IPHASE)=RLD
      WEELT(IPHASE)=W
      WART(IPHASE)=WTOT
315    QT(IPHASE)=Q
      SFCUT(IPHASE)=SFCU
      TNUT(IPHASE)=TNU
      CDINST(IPHASE)=CDINSP
      PRTOTT(IPHASE)=PRTOT
320    XT(IPHASE)=XCLIMB
      RETURN
      END

```

```

1      SUBROUTINE CRUISE(ICALC,NERROR,IGEO,KGPRNT)
      REAL MSTART
      COMMON /TRAJCM/ ALPHA,ARW,CD,CDL,CDD,CL,DESLF,DRAG,EN,HN,KP,
1      RLD,SMN,QMAX,RANGE,SFC,SWING,THRUST,TW,ULTLF,W,WAMMUN,WETANK,
5      WFUEL,WGTOWT,WMISS,WTOT,WPLWT,MSTART(12),HSTART(12),TIM(12),
      PS1GT(12),TDOTST(12),NZST(12),PSIF(12),TDOTIT(12),NZIT(12),
      WBOMBS,WFEXT,SFCU,THRSTU,CDINSP,PRTOT,WKFUEL,CRMACH,FLTD,X(12),
      FLLAND,TENDUR,BLRANG,BLTIME,TIMTO1,TIMTO2,WFTO,PNAME1(12),
      PNAME2(12),SMNT(12),HNT(12),WFT(12),TIMET(12),XT(12),CLT(12),
10     CDT(12),ALPHAT(12),CLIT(12),CDIT(12),ALPHIT(12),WFUSED,WLAND,
      QT(12),RLDT(12),SFC(12),TNT(12),CET(12),DY(18),
      IAD,IPS,ITS,IWS,IBS,NPHASE,IDY(14)
      COMMON /TRAJEX/ ALPIGT(12),CDINST(12),CD1GT(12),CL1GT(12),DECEL,
1      DELHP,DELMP,DXCRUS,FLFAC,FRFURE,HEND(12),HENDUR,HMAXP,HMINP,
15     HPATH(50),HEND(12),HENDUR,NT(12),PRTOTT(12),RADIT(12),
      RADST(12),SF,SFCUT(12),SHMAXP,SHMINP,
      TNUT(12),VIND(12),VELT(12),VPATH(50),WART(12),WCOMBP,WEELT(12),
      WFRES,WFTOT,WFTO1,WFTO2,WFTAP,WKLAND,XDESC,XGRLAN,FWGMAX,TOL,
      WFIXED,WGCALC,WPL,WGTO,DUMY(82),IA(12),IAS,IB(12),IBREG,IENDUR,
20     IMISS,IP(12),IPHASE,IPLDT,IPRINT,IPRT(12),IPSIZE,IPSTO1,IPSTO2,
      IW(12),IX(12),KERROR,MILCOM,NCRUSE,NLEGCL,NLEGCR,NLEGLO,NMISS,
      NPATH,IDUMY(6)
      DIMENSION ANS(4),HNTAB(10),SHTAB(10)
      DATA HNTAB/0.,5000.,10000.,15000.,20000.,25000.,30000.,
25     35000.,37000.,100000./
      DATA SHTAB/34500.,32900.,31400.,30500.,29400.,28200.,27110.,
      25950.,20900.,20900./
      DATA DMON/0.0/
      IF (IPRT(IPHASE).NE.0) WRITE (6,600)
30     600    FORMAT(///19H          CRUISE PHASE//)
      C-----TEST FOR TANKS DROP
      WFUSED=WGTO-W
      IF (IWS.EQ.0) WFUSED=WFUSED-WMISS
      IF (IBS.EQ.0) WFUSED=WFUSED-WBOMBS
35     IF (IAS.EQ.0) WFUSED=WFUSED-WAMMUN
      IF (ITS.EQ.1.AND.WFUSED.GE.WFEXT) W=W-WETANK
      IF (ITS.EQ.1.AND.WFUSED.GE.WFEXT) ITS=0
      IPS=IP(IPHASE)
      SMN=MSTART(IPHASE)
      HN=HSTART(IPHASE)
40     IF (MSTART(IPHASE).LT.0.0) SMN=SMNT(IPHASE-1)
      IF (HSTART(IPHASE).LT.0.0) HN=HNT(IPHASE-1)
      IF (IBREG.EQ.0.OR.HSTART(IPHASE).NE.0.0) GO TO 10
      CALL GOLDEN(2000.,60000.,1,12,XX,Y,NERROR,IGEO,KGPRNT)
45     IF (NERROR.GE.2) RETURN
      HN=XX
10     CALL AT62(HN,ANS)
      RHD=ANS(1)
      A=ANS(4)
50     VEL=SMN*A
      CL=2.*W/(RHD*SWING*VEL**2)
      IAD=8
      CALL TRAJ01(2)
      CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
55     IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
      COSA=COS(ALPHA/57.3)

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

        TIME=G.0
        WF=0.0
60      IF(X(IPHASE).GE.0.)XCRUSE=X(IPHASE)+DXCRUS+XDESC*FLOAT(IX(IPHASE))
        IF(X(IPHASE).LT.0.)XCRUSE=RANGE/FLOAT(NCRUSE)+DXCRUS+
1      XDESC*FLOAT(IX(IPHASE))
        IF (XCRUSE .LE. 0.) DXCRUS=0.0
        IF (XCRUSE .LE. 0.) GO TO 90
65      IF (X(IPHASE).LT.100.) NLEGS=2
        IF ((X(IPHASE).GE.100.) .AND. (X(IPHASE).LT.300.)) NLEGS=3
        IF ((X(IPHASE).GE.300.) .AND. (X(IPHASE).LT.1000.)) NLEGS=4
        IF ((X(IPHASE).GE.1000.) .AND. (X(IPHASE).LT.3000.)) NLEGS=5
        IF (X(IPHASE).GE.3000.) NLEGS=6
70      IF (NLEGCR.GT.0) NLEGS=NLEGCR
        DELX=XCRUSE/FLOAT(NLEGS)
        IF (IPRT(IPHASE).EQ.0) GO TO 20
        WRITE (6,601) W,SMN,HN,RHO,A,VEL,CL,ALPHA,TIME,WF,XCRUSE,DXCRUS,
1      DELX,IAD,IPS
75      601  FORMAT(/6X,3HW =E14.7,4X,5HSMN =E14.7,5X,4HHN =E14.7,4X,
1      E14.7,6X,3HA =E14.7/4X,5HVEL =E14.7,5X,4HCL =E14.7,2X,7
2      E14.7,3X,6HTIME =E14.7,5X,4HWF =E14.7/1X,8HXCRUSE =E14.7,1X,
3      8HDXCRUS =E14.7,3X,6HDELX =E14.7,4X,5HIAD =I2,16X,5HIPS =I2/)
20      DO 50 LEG=1,NLEGS
80      IF (HEND(IPHASE).EQ.-1.) GO TO 21
        RHO1=RHO
        RHO2=2.*W/(CL*SWING*VEL**2)
        CALL TAIN(THNTAB,SHTAB,HN,SH,10,1,NERR,DMON)
        HN=HN+2.*SH*(RHO1-RHO2)/(RHO1+RHO2)
85      CALL AT62(HN,ANS)
        A=ANS(4)
        VEL=SMN*A
        RHO=2.*W/(CL*SWING*VEL**2)
        Q=.5*RHO*VEL**2
90      QS=Q*SWING
        GO TO 22
21      HN=HSTART(IPHASE)
        CALL AT62(HN,ANS)
        A=ANS(4)
95      RHO=ANS(1)
        VEL=SMN*A
        Q=.5*RHO*VEL**2
        QS=Q*SWING
        CL=W/QS
100     22  CALL TRAJ01(2)
        CALL AERO(ICALC,NERRJR,IGED,KGPRNT)
        IF (NERROR.GE.2) RETURN
        CALL TRAJ01(1)
        DRAG=CD*QS
105     CALL TRAJ01(2)
        CALL PROP(ICALC,NERROR,IGED,KGPRNT)
        IF (NERROR.GE.2) RETURN
        CALL TRAJ01(1)
        TN=THRUST*EN
110     TNU=THRSTU*EN
        IF (ABS(DRAG/TN-1.0).LE.0.03) GO TO 30
        IF (KERROR.EQ.2) WRITE (6,602) IPHASE,LEG,DRAG,TN
602     FORMAT(/ 17H IN CRUISE, PHASE,I2,5H, LEG,I2, 46H, THRUST DOES NOT
1      MATCH DRAG WITHIN 3 PERCENT./7H DRAG =,E12.5,4X,4HTN =,E12.5/)

```

```

15      30      DELWF=.8444*W*SFC*CD*DELX*SQRT(2.*RHO*SWING/(CL*W))/CDSA
C-----FUEL USED DURING CRUISE
        WF=WF+DELWF
        DELT=6080.*DELX/VEL
20      C-----CRUISING TIME TO TARGET
        TIME=TIME+DELT/60.
        W=W-DELWF
C-----TEST FOR TANKS DROP
        WFUSED=WGTO-W
        IF (IWS.EQ.0) WFUSED=WFUSED-WMISS
25      IF (IBS.EQ.0) WFUSED=WFUSED-WBOMBS
        IF (IAS.EQ.0) WFUSED=WFUSED-WAMMUN
        IF (ITS.EQ.1.AND.WFUSED.GE.WFEXT) W=W-WETANK
        IF (ITS.EQ.1.AND.WFUSED.GE.WFEXT) ITS=0
        IF (IPRT(IPHASE).EQ.0) GO TO 40
30      WRITE (6,603) LEG,RHO1,RHO2,SH,HN,A,VEL,RHO,Q,QS,CD,DRAG,TN,SFC,
1 DELWF,WF,DELT,TIME,W
603      FORMAT(/4X,5HLEG =I2/3X,6HRHO1 =E14.7,3X,6HRHO2 =E14.7,5X,4HSH =
1 E14.7,5X,4HHN =E14.7,6X,3HA =E14.7/4X,5HVEL =E14.7,4X,5HRHO =
2 E14.7,6X,3HQ =E14.7,5X,4HQS =E14.7,5X,4HCD =E14.7/3X,6HDRAG =
35 3 E14.7,5X,4HTN =E14.7,4X,5HSFC =E14.7,2X,7HDELWF =E14.7,5X,4HWF =
4 E14.7/3X,6HDELT =E14.7,3X,6HTIME =E14.7,6X,3HW =E14.7/)
40      IF (W.GT.WPL) GO TO 50
        IF (KERROR.EQ.2) WRITE (6,604) IPHASE,LEG
604      FORMAT(/72H W,LE,WPL IN CRUISE. PHASE=,I2,7H   LEG=,I2/)
40      RETURN
50      CONTINUE
        BLTIME=BLTIME+TIME/60.
        BLRANG=BLRANG+XCRUSE
        DXCRUS=0.0
45      C-----TEST FOR WEAPONS DROP
        IF (IW(IPHASE).EQ.0) GO TO 60
        W=W-WMISS
        IWS=0
C-----TEST FOR BOMBS DROP
50      60      IF (IB(IPHASE).EQ.0) GO TO 70
        W=W-WBOMBS
        IBS=0
C-----TEST FOR AMMO DROP
155      70      IF (IA(IPHASE).EQ.0) GO TO 80
        W=W-WAMMUN
        IAS=0
80      IF (IPRINT.EQ.0) GO TO 90
        WRITE (6,605) W
160      605      FORMAT(/6X,3HW =E14.7/)
90      SMNT(IPHASE)=SMN
        HNT(IPHASE)=HN
        CLT(IPHASE)=CL
        ALPHAT(IPHASE)=ALPHA
        WFT(IPHASE)=WF
165      TIMET(IPHASE)=TIME
        VELT(IPHASE)=VEL
        SFCT(IPHASE)=SFC
        INT(IPHASE)=TN
        CDT(IPHASE)=CD
170      RLDT(IPHASE)=RLD
        WEELT(IPHASE)=W

```

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE CRUISE 76/76 QPT=2

FTN 4.5+410

175 WART(IPHASE)=WTOT
QT(IPHASE)=Q
SFCUT(IPHASE)=SFCU
TNUT(IPHASE)=TNU
CDINST(IPHASE)=CDINSP
PRTOTT(IPHASE)=PRTOT
XT(IPHASE)=XCRUSE
180 RETURN
END

```

1      SUBROUTINE ACCEL(ICALC,NERROR,IGEO,KGPRNT)
      REAL MSTART,MEND
      COMMON /TRAJCM/ ALPHA,ARW,CD,CDL,CDO,CL,DESLF,DRAG,EN,HN,KP,
5      1 RLD,SMN,QMAX,RANGE,SFC,SWING,THRUST,TW,ULTLF,W,WAMMUN,WETANK,
      2 WFUEL,WGTOWT,WMISS,WTOT,WPLWT,MSTART(12),HSTART(12),TIM(12),
      3 PSIGT(12),TDDTST(12),NZST(12),PSIT(12),TDDTIT(12),NZIT(12),
      4 KBOMBS,WTEXT,SFCU,THRSTU,CDINSP,PRTOT,WKFUEL,CRMACH,FLTO,X(12),
10     5 FLLAND,TENDUR,BLRANG,BLTIME,TIMT01,TIMT02,WFTO,PNAME1(12),
      6 PNAME2(12),SMNT(12),HNT(12),WFT(12),TIMET(12),XT(12),CLT(12),
      7 CDT(12),ALPHAT(12),CLIT(12),CDIT(12),ALPHIT(12),WFUSED,WLAND,
      8 QT(12),RLDT(12),SFCT(12),TNT(12),CET(12),DY(18),
      9 IAD,IPS,ITS,IWS,IBS,NPHASE,IDY(14)
      COMMON /TRAJEX/ ALPIGT(12),CDINST(12),CD1GT(12),CL1GT(12),DECEL,
15     1 DELHP,DELMP,DXCRUS,FLFAC,FRFURE,HEND(12),HENDUR,HMAXP,HMINP,
      2 HPATH(50),MEND(12),MENDUR,NT(12),PRTOTT(12),RADIT(12),
      3 RADST(12),SF,SFCUT(12),SMMAXP,SMMINP,
      4 TNUT(12),VIND(12),VELT(12),VPATH(50),WART(12),WCOMBP,WEELT(12),
      5 WFRES,WFTOT,WFTO1,WFTO2,WFTRAP,WKLAND,XDESC,XGRLAN,FWGMAX,TOL,
20     6 WFIXED,WGCALC,WPL,WGTO,DUMY(82),IA(12),IAS,IB(12),IBREG,IENDUR,
      7 IMISS,IP(12),IPHASE,IPLT,IPRINT,IPRT(12),IPSIZE,IPST01,IPST02,
      8 IW(12),IX(12),KERROR,HILCOM,NCRUSE,NLEGCL,NLEGCR,NLEGLO,NMISS,
      9 NPATH,IDUMY(6)
      DIMENSION ANS(4)
      IF (IPRT(IPHASE).NE.0) WRITE (6,600)
25     600  FORMAT(///25H _____ ACCELERATION PHASE//)
      IF (MSTART(IPHASE).LT.0.0) SMN=SMNT(IPHASE-1)
      IF (MSTART(IPHASE).GE.0.0) SMN=MSTART(IPHASE)
      IF (HSTART(IPHASE).LT.0.0) HN=HNT(IPHASE-1)
      IF (HSTART(IPHASE).GE.0.0) HN=HSTART(IPHASE)
30     CALL AT62(HN,ANS)
      A=ANS(4)
      RHO=ANS(1)
      TEMP=1.8*ANS(3)
      SQTEMP=SQRT(TEMP)
35     IPS=IP(IPHASE)
      IAD=8
      C-----COMPUTE INTEGRALS SUMWF AND SUMDT BY TRAPEZOIDAL RULE
      SUMWF=0.0
      SUMDT=0.0
40     XACCEL=0.0
      TIMECR=0.0
      FUELCR=0.0
      WF=0.0
      TIME=0.0
45     LEG=0
      IF (IPRT(IPHASE).EQ.0) GO TO 10
      WRITE (6,601) LEG,HN,A,RHO,TEMP,SUMWF,SUMDT,SMN,W,IAD,IPS
601    FORMAT(//4X,5HLEG =I2/5X,4HHN =E14.7,6X,3HA =E14.7,4X,
1      1 5HRHO =E14.7,3X,6HTEMP =E14.7,2X,7HSUMWF =E14.7/2X,7HSUMDT =,
50     2 E14.7,4X,5HSMN =E14.7,6X,3HW =E14.7,4X,5HIAD =I2,16X,5HIPS =I2/)
      10  IF (MSTART(IPHASE).GE.MEND(IPHASE)) GO TO 120
      IF (SMNT(IPHASE-1).GE.MEND(IPHASE)) GO TO 120
      C-----START OF LDDP.
20     LEG=LEG+1
55     IF (LEG.GT.50) GO TO 80
      CALL TRAJ01(2)
      CALL PROP(ICALC,NERROR,IGEO,KGPRNT)

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

        IF (NERROR.GE.2) RETURN
        CALL TRAJ01(1)
60      TN=THRUST*EN
        TNU=THRSTU*EN
        VEL=SMN*A
        30      CL=2.*W/(RHO*SWING*VEL**2)
        CALL TRAJ01(2)
65      CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
        IF (NERROR.GE.2) RETURN
        CALL TRAJ01(1)
        DENOM=TN*CD*(ALPHA/57.3)-.5*CD*RHO*SWING*VEL**2
        YWF=SFC*TN/DENOM
70      YDT=1./DENOM
        IF ((YWF.GT.0.0).AND.(YDT.GT.0.0)) GO TO 50
        DTIME=60.
        DFUEL=TN*SFC*DTIME/3600.
        W=W-DFUEL
75      IF (W.GT.WPL) GO TO 40
        IF (KERROR.EQ.2) WRITE (6,602) IPHASE,LEG
        602      FORMAT(/26H W.LE.WPL IN ACCEL. PHASE=,I2,7H   LEG=,I2/)
        RETURN
        40      DXCR=VEL*DTIME/6080.
80      TIMECR=(TIMECR+DTIME)/60.
        FUELCR=FUELCR+DFUEL
        XACCEL=XACCEL+DXCR
        GO TO 30
        50      IF (LEG.EQ.1) GO TO 60
85      SUMWF=SUMWF+.5*(YWF+YWFP)*(SMN-SMNP)
        SUMDT=SUMDT+.5*(YDT+YDTP)*(SMN-SMNP)
        DELX=.7525*(YDT+YDTP)*(SMN-SMNP)*VEL*W*SQTEMP/6080.
        XACCEL=XACCEL+DELX
        WF=(EXP(1.525*SUMWF*SQTEMP/3600.)-1.)*W
90      DELWF=WF-WFP
        W=W-DELWF
        TIME=1.525*SUMDT*SQTEMP*W/60.
        DELT=TIME-TIMEP
        60      IF (IPRT(IPHASE).EQ.0) GO TO 70
95      WRITE (6,603) LEG,TN,SFC,VEL,SMN,ALPHA,CL,CD,DENOM,YWF,YDT,SUMWF,
        1 DELWF,SUMDT,DELT,DELX,XACCEL
        603      FORMAT(/4X,5HLEG =I2/5X,4HTN =E14.7,4X,5HSFC =E14.7,4X,
        1 5HVEL =E14.7,4X,5HSMN =E14.7,2X,7HALPHA =E14.7/5X,4HCL =E14.7,5X,
        2 4HCD =E14.7,2X,7HDENOM =E14.7,4X,5HYWF =E14.7,4X,5HYDT =E14.7/
100      3 2X,7HSUMWF =E14.7,2X,7HDELWF =E14.7,2X,7HSUMDT =E14.7,
        4 3X,6HDELT =E14.7,3X,6HDELX =E14.7/1X,8HXACCEL =E14.7/)
        C-----MAKE TEST TO SEE IF YOU ARE DONE. IF SO, JUMP OUT OF LOOP.
        70      IF (SMN.GE.MEND(IPHASE)) GO TO 80
105      YWFP=YWF
        YDTP=YDT
        SMNP=SMN
        WFP=WF
        TIMEP=TIME
        IF (SMN.LE..87.OR.SMN.GE.1.1) DELSMN=.05
110      IF (SMN.GT..87.AND.SMN.LT.1.1) DELSMN=.02
        SMN=SMN+DELSMN
        IF (SMN.GE.MEND(IPHASE)) SMN=MEND(IPHASE)
        C-----END OF LOOP. GO BACK TO START OF LOOP.
        GO TO 20

```


115 C-----FUEL USED DURING ACCELERATION RUN
80 WF=WF+FUELCR
C-----TIME ELAPSED DURING ACCELERATION RUN
TIME=TIME+TIMECR
BLTIME=BLTIME+TIME/60.
120 BLRANG=BLRANG+XACCEL
IF (W.GT.WPL) GO TO 90
IF (KERROR.EQ.2) WRITE (6,604) IPHASE
604 FORMAT(/26H W.LE.WPL IN ACCEL. PHASE=,I2/)
RETURN
125 90 VEL=SMN*A
Q=.5*RHO*VEL**2
DXCRUS=DXCRUS+XACCEL*FLOAT(IX(IPHASE))
C-----TEST FOR WEAPONS DROP
IF (IW(IPHASE).EQ.0) GO TO 100
130 W=W-WMISS
IWS=0
C-----TEST FOR BOMBS DROP
100 IF (IB(IPHASE).EQ.0) GO TO 110
W=W-WBOMBS
135 IBS=0
C-----TEST FOR AMMO DROP
110 IF (IA(IPHASE).EQ.0) GO TO 120
W=W-WAMMUN
IAS=0
140 120 IF (IPRT(IPHASE).EQ.0) GO TO 130
WRITE (6,605) SUMWF,SUMDT,TIME,VEL,Q,XACCEL,DXCRUS,WF,W
605 FORMAT(/2X,7HSUMWF =E14.7,2X,7HSUMDT =E14.7,3X,6HTIME =
1 E14.7,4X,5HVEL =E14.7,6X,3HQ =E14.7/1X,8HXACCEL =E14.7,1X,
2 BHDXCRUS =E14.7,5X,4HWF =E14.7,6X,3HW =E14.7/)
145 130 SMNT(IPHASE)=SMN
HNT(IPHASE)=HN
CLT(IPHASE)=CL
ALPHAT(IPHASE)=ALPHA
WFT(IPHASE)=WF
150 TIMET(IPHASE)=TIME
VELT(IPHASE)=VEL
SFCT(IPHASE)=SFC
TNT(IPHASE)=TN
CDT(IPHASE)=CD
155 RLDT(IPHASE)=RLD
WEELT(IPHASE)=W
WART(IPHASE)=WTOT
QT(IPHASE)=Q
SFCUT(IPHASE)=SFCU
160 TNUT(IPHASE)=TNU
CDINST(IPHASE)=CDINSP
PRTOTT(IPHASE)=PRTOT
XT(IPHASE)=XACCEL
RETURN
165 END

ORIGINAL PAGE IS
OF POOR QUALITY

```

1      SUBROUTINE COMBAT(ICALC,NERROR,IGEO,KGPRNT)
      REAL MSTART,NT,NTURNS,LIFT,NZS,NZI,NZST,NZIT
      COMMON /TRAJCH/ ALPHA,ARW,CD,CDL,CDQ,CL,DESLF,DRAG,EN,HN,KP,
5      1 RLD,SMN,QMAX,RANGE,SFC,SWING,THRUST,TW,ULTLF,W,WAMMUN,WETANK,
      2 WFUEL,WGTOWT,WMISS,WTOT,WPLWT,MSTART(12),HSTART(12),TIM(12),
      3 PS1GT(12),TOOTST(12),NZST(12),PSIT(12),TOOTIT(12),NZIT(12),
      4 WBOMBS,WFEET,SFCU,THRSTU,CDINSP,PRTOT,WKFUEL,CRMACH,FLTD,X(12),
      5 FLLAND,TENDUR,BLRANG,BLTIME,TIMTO1,TIMTO2,WFTD,PNAME1(12),
      6 PNAME2(12),SMNT(12),HNT(12),WFT(12),TIMET(12),XT(12),CLT(12),
10     7 CDT(12),ALPHAT(12),CLIT(12),CDIT(12),ALPHIT(12),WFUSED,WLAND,
      8 QT(12),RLDT(12),SFCT(12),TNT(12),CET(12),DY(18),
      9 IAO,IPS,ITS,IWS,IBS,NPHASE,IDY(14)
      COMMON /TRAJEX/ ALP1GT(12),CDINST(12),CD1GT(12),CL1GT(12),DECEL,
15     1 DELHP,DELMP,DXCRUS,FLFAC,FRFURE,HEND(12),HENDUR,HMAXP,HMINP,
      2 HPATH(12),MEND(12),MENDUR,NT(12),PRTOT(12),RADIT(12),
      3 RADST(12),SF,SFCUT(12),SMMAXP,SMMINP,
      4 TNUT(12),VIND(12),VELT(12),VPATH(12),WAPT(12),WCOMBP,WEELT(12),
      5 WFRES,WFTOT,WFTO1,WFTO2,WFTAP,WKLAND,XDESC,XGRLAN,FWGMAX,TOL,
      6 WFIXED,WGCALC,WPL,WGTO,DUMY(82),IA(12),IAS,IB(12),IBREG,IENDUR,
20     7 IMISS,IP(12),IPHASE,IPLDT,IPRINT,IPRT(12),IPSIZE,IPSTO1,IPSTO2,
      8 IW(12),IX(12),KERROR,MILCOM,NCRUSE,NLEGCL,NLEGCR,NLEGLO,NMISS,
      9 NPATH,IDUMY(6)
      DIMENSION ANS(4)
      HN=HSTART(IPHASE)
25     IF (HSTART(IPHASE).LT.0.0) HN=HNT(IPHASE-1)
      CALL AT62(HN,ANS)
      RHO=ANS(1)
      A=ANS(4)
      SMN=MSTART(IPHASE)
30     IF (MSTART(IPHASE).LT.0.0) SMN=SMNT(IPHASE-1)
      VEL=SMN*A
      Q=.5*RHO*VEL**2
      QS=Q*SWING
      XCOMB=0.0
35     IF (IPRT(IPHASE).NE.0) WRITE (6,600) W,HN,RHO,A,SMN,VEL,Q,QS,XCOMB
600    FORMAT(///19H          COMBAT PHASE///
      1 6X,3HW =E14.7,5X,4HHN =E14.7,4X,5HRHO =E14.7,6X,3HA =E14.7,
      2 4X,5HSMN =E14.7,4X,5HVEL =E14.7,6X,3HQ =E14.7,5X,4HQS =E14.7,
      3 2X,7HXCOMB =E14.7//)
40     C-----
      C ONE G FLIGHT
      C-----
      CL=W/QS
      IAO=8
45     CALL TRAJ01(2)
      CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
      DRAG=CD*QS
50     IPS=IP(IPHASE)
      CALL TRAJ01(2)
      CALL PROP(ICALC,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
      TN=THRUST*EN
55     TNU=THRSTU*EN
      PSIG=(TN-DRAG)*VEL/W

```

SUBROUTINE COMBAT

76/76 OPT=2

FTN 4.5+410

05

```

        PS1GT(IPHASE)=PS1G
        ALP1GT(IPHASE)=ALPHA
60      CL1GT(IPHASE)=CL
        CD1GT(IPHASE)=CD
        IF (IPRT(IPHASE).EQ.0) GO TO 10
        WRITE (6,601) IAO,CL,CD,ALPHA,DRAG,IPS,TN,SFC,PS1G
601     FORMAT(/1X,12HONE G FLIGHT// 4X,5HIAO =I2,17X,4HCL =E14.7,
65       1 5X,4HCD =E14.7,2X,7HALPHA =E14.7,3X,6HDRAG =E14.7/
        2 4X,5HIPS =I2,17X,4HTN =E14.7,4X,5HSFC =E14.7,3X,6HPS1G =E14.7/)
C-----
C  SUSTAINED TURNS
C-----
70     10  NZS=1.001
        IF(PS1G.LE.0.) GO TO 40
        IAO=4
        CALL TRAJ01(2)
        CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
75     IF (NERROR.GE.2) RETURN
        CALL TRAJ01(1)
        LIFT=CL*QS
C  LOAD FACTOR
        NZS=(LIFT+TN*SIN(ALPHA/57.2958))/W
80     IF (IPRT(IPHASE).EQ.0) GO TO 20
        WRITE (6,602) IAO,CL,CD,ALPHA,LIFT,NZS
602     FORMAT(/1X,15HSUSTAINED TURNS// 4X,5HIAO =I2,17X,4HCL =E14.7,
        1 5X,4HCD =E14.7,2X,7HALPHA =E14.7,3X,6HLIFT =E14.7/
        2 4X,5HNZS =E14.7/)
85     20  IF (NZS.LE.DESLF) GO TO 30
        IAO=6
        CALL TRAJ01(2)
        CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
90     IF (NERROR.GE.2) RETURN
        CALL TRAJ01(1)
        NZS=DESLF
        IF (IPRT(IPHASE).EQ.0) GO TO 30
        WRITE (6,603) IAO,CL,CD,ALPHA,DESLF,NZS
95     603  FORMAT(/4X,5HIAO =I2,17X,4HCL =E14.7,5X,4HCD =E14.7,
        1 2X,7HALPHA =E14.7,2X,7HDESLF =E14.7/4X,5HNZS =E14.7/)
        30  IF (NZS.GT.1.0) GO TO 50
        40  IF (NERROR.EQ.2) WRITE (6,604) IPHASE,NZS
604     FORMAT(/36H INSUFFICIENT LIFT IN COMBAT. PHASE=,I2,5X,
        1 4HNZS=,1PE11.4)
100     NZS=1.001
        WRITE (6,605) NZS
605     FORMAT(/4X,5HNZS =E14.7/)
        50  PHI=ARCOS(1.0/NZS)
C  TURN RADIUS
105     RADIUS=VEL**2/(32.2*TAN(PHI))
C  TURNING TIME OR NUMBER OF TURNS
        TIME=TIM(IPHASE)
        NURNS=NT(IPHASE)
        IF (TIME.LE.0.0) TIME=NURNS*RADIUS/(9.55*VEL)
        IF (NURNS.LE.0.0) NURNS=9.55*TIME*VEL/RADIUS
110     XCOMB=VEL*TIME*60./6080.
        BLTIME=BLTIME+TIME/60.
        BLRANG=BLRANG+XCOMB
C  FULL USED DURING TURNS

```

```

115      WF=SFC*TN*TIME/60.
      C COMBAT ENERGY
      CE=3600.*WF*PS1G/(SFC*TN)
      TDOTS=57.3*VEL/RADIUS
      RADS=RADIUS
120      IF (IPRT(IPHASE).EQ.0) GO TO 60
      WRITE (6,606) PHI,RADIUS,TIME,NURNS,WF,TDOTS,XCOMB
606      FORMAT(/4X,5PHI =E14.7,1X,8HRADIUS =E14.7,3X,6HTIME =E14.7,
1 1X,8HNURNS =E14.7,5X,4HWF =E14.7/2X,7HTDOTS =E14.7,
2 2X,7HXCOMB =E14.7/)
125      60 SMNT(IPHASE)=SMN
      HNT(IPHASE)=HN
      CLT(IPHASE)=CL
      ALPHAT(IPHASE)=ALPHA
      WFT(IPHASE)=WF
130      VELT(IPHASE)=VEL
      TIMET(IPHASE)=TIME
      SFCT(IPHASE)=SFC
      TNT(IPHASE)=TN
      CDT(IPHASE)=CD
135      RLDT(IPHASE)=RLD
      WART(IPHASE)=WTOY
      QT(IPHASE)=Q
      SFCUT(IPHASE)=SFCU
      TNUT(IPHASE)=TNU
      COINST(IPHASE)=COINSP
140      PRTOT(IPHASE)=PRTOT
      XT(IPHASE)=XCOMB
      TDOTST(IPHASE)=TDOTS
      RADST(IPHASE)=RADS
145      NZST(IPHASE)=NZS
      CET(IPHASE)=CE
      C-----
      C MAXIMUM INSTANTANEOUS CONDITIONS
      C-----
150      IAO=5
      CALL TRAJ01(2)
      CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
155      CLMAX=CL
      LIFT=CLMAX*QS
      C LOAD FACTOR
      NZI=(LIFT+TN*SIN(ALPHA/57.2958))/W
      IF (IPRT(IPHASE).EQ.0) GO TO 70
160      WRITE (6,607) IAO,CL,CD,ALPHA,LIFT,NZI
607      FORMAT(/1X,19HINSTANTANEOUS TURNS// 4X,5HIAO =I2,17X,4HCL =E14.7,
1 5X,4HCD =E14.7,2X,7HALPHA =E14.7,3X,6HLIFT =E14.7/
2 4X,5HNZI =E14.7/)
170      IF (NZI.LE.DESLF) GO TO 80
165      IAO=6
      CALL TRAJ01(2)
      CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
170      NZI=DESLF
      IF (IPRT(IPHASE).EQ.0) GO TO 80

```

```

        WRITE (6,608) IAH,CL,CD,ALPHA,DESLF,NZI
608    FORMAT(/4X,5HIAH =I2,17X,4HCL =E14.7,5X,4HCD =E14.7,
175    1 2X,7HALPHA =E14.7,2X,7HDESLF =E14.7/4X,5HNZI =E14.7/)
80    IF (NZI.GT.1.0) GO TO 90
        IF (KERROR.EQ.2) WRITE (6,609) IPHASE,NZI
609    FORMAT(/36H INSUFFICIENT LIFT IN COMBAT. PHASE=,I2,5X,
1 4HNZI=,1PE11.4)
        RETURN
180    C PS FOR MAXIMUM INSTANTANEOUS TURN RATE
90    DRAG=CD*QS
        PSI=(TN*CDOS(ALPHA/57.3)-DRAG)*VEL/W
        ALPHI=ALPHA
        CLI=CL
185    CDI=CD
        PHI=ARCOS(1.0/NZI)
        C MAXIMUM INSTANTANEOUS TURN RATE AND RADIUS
        RADIUS=VEL**2/(32.2*TAN(PHI))
        TDOI=57.3*VEL/RADIUS
190    RADI=RADIUS
        W=W-WF
        IF (W.GT.WPL) GO TO 100
        IF (KERROR.EQ.2) WRITE (6,610) IPHASE
610    FORMAT(/27H W.LE.WPL IN COMBAT. PHASE=,I2)
195    RETURN
        C TEST FOR WEAPONS DROP
100    IF (IW(IPHASE).EQ.0) GO TO 110
        W=W-WMISS
        IWS=0
200    C TEST FOR BOMBS DROP
110    IF (IB(IPHASE).EQ.0) GO TO 120
        W=W-WBOMBS
        IBS=0
        C TEST FOR AMMO DROP
205    120    IF (IA(IPHASE).EQ.0) GO TO 130
        W=W-WAMMUN
        IAS=0
130    IF (IPRT(IPHASE).EQ.0) GO TO 140
        WRITE (6,611) DRAG,PSI,PHI,RADIUS,TDOI,W
210    611    FORMAT(/3X,6HDRAG =E14.7,4X,5HPSI =E14.7,4X,5HPHI =E14.7,
1 1X,8HRADIUS =E14.7,2X,7HTDOI =E14.7/6X,3HW =E14.7/)
140    WEELT(IPHASE)=W
        PSIT(IPHASE)=PSI
        TDOIIT(IPHASE)=TDOI
215    RADIT(IPHASE)=RADI
        NZIT(IPHASE)=NZI
        ALPHIT(IPHASE)=ALPHI
        CLIT(IPHASE)=CLI
        CDIT(IPHASE)=CDI
220    RETURN
        END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

1      SUBROUTINE LOITER(ICALC,NERROR,IGEO,KGPRNT)
      REAL MSTART,KP,NZLOIT
      COMMON /TRAJCM/ ALPHA,ARW,CD,CDL,CDD,CL,DESLF,DRAG,EN,HN,KP,
1      PLO,SMN,QMAX,RANGE,SFC,SWING,THRUST,TW,ULTLF,W,WHAMMUN,WETANK,
5      WFUEL,WGTOWT,WMISS,WTOT,WPLWT,MSTART(12),HSTART(12),TIM(12),
      PSIGT(12),TDOST(12),NZST(12),PSIT(12),TDOIT(12),NZIT(12),
      WBOBBS,WFEET,SFCU,THRSTU,CDINSP,PRTOT,WKFUEL,CRMACH,FLTO,X(12),
      FLAND,TENDUR,BLRANG,BLTIME,TIMTJ1,TIMTJ2,WFTO,PNAME1(12),
      PNAME2(12),SMNT(12),HNT(12),WFT(12),TIMET(12),XT(12),CLT(12),
10     CDT(12),ALPHAT(12),CLIT(12),CDIT(12),ALPHIT(12),WFUSED,WLAND,
      GT(12),RLDT(12),SFCT(12),TNT(12),CET(12),DY(18),
      IAO,IPS,ITS,IWS,IBS,NPHASE,IDY(14)
      COMMON /TRAJEX/ ALPLGT(12),CDINST(12),CDIGT(12),CLIGT(12),DECEL,
1      DELHP,DELMF,DXCRUS,FLFAC,FRFURE,HEND(12),HENDUR,HMAXP,HMINP,
15     HPATH(50),HMEND(12),MENDUR,NT(12),PRTOTT(12),RADIT(12),
      RADST(12),SF,SFCUT(12),SMHXP,SMHNP,
      TNU(12),VIND(12),VELT(12),VPATH(50),WART(12),WCOMBP,WHEELT(12),
      WFRES,WFTOT,WFTO1,WFTO2,WFTAP,WKLAND,XDESC,XGRLAN,FWGHAX,TOL,
      WFIXED,WGALC,WPL,WGTO,DUMY(82),IA(12),IAS,IB(12),IBREG,IENDUR,
20     IMISS,IP(12),IPHASE,IPLOT,IPRINT,IPRT(12),IPSIZE,IPSTO1,IPSTO2,
      IW(12),IX(12),KERROR,MILCOM,NCRUSE,NLEGCL,NLEGCR,NLEGLO,NMISS,
      NPATH,IDUMY(6)
      DIMENSION ANS(4),HNTAB(10),SHTAB(10)
      DATA HNTAB/0.,5000.,10000.,15000.,20000.,25000.,30000.,
25     35000.,37000.,100000./
      DATA SHTAB/34500.,32900.,31400.,30500.,29400.,28200.,27110.,
      25950.,20900.,20900./
      DATA DMON/0.0/
      IF (IPRT(IPHASE).NE.0) WRITE (6,600)
30     FORMAT(///19H          LOITER PHASE//)
      IPS=4
      SMNIN=MSTART(IPHASE)
      IF (MSTART(IPHASE).LT.0.0) SMNIN=SMNT(IPHASE-1)
      SMN=SMNIN
35     HN=MSTART(IPHASE)
      IF (HSTART(IPHASE).LT.0.0) HN=HNT(IPHASE-1)
      IF (IENDUR.EQ.0.0R.HSTART(IPHASE).NE.0.0) GO TO 10
      CALL GOLDEN(2000.,65000.,2,12,XX,Y,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
40     HN=XX
      10    CALL AT62(HN,ANS)
      RHO=ANS(1)
      A=ANS(4)
      VEL=SMN*A
45     IF (SMN.GT.0.0) CL=2.*W/(RHO*SWING*VEL**2)
      TIME=TIM(IPHASE)
      NLEGS=TIME/60.+1.0
      IF (NLEGS.LT.1) NLEGS=1
      IF (NLEGS.GT.6) NLEGS=6
50     IF (NLEGLO.GT.0) NLEGS=NLEGLO
      DELT=TIME/FLOAT(NLEGS)
      WF=0.0
      XLOIT=0.0
      IF (IPRT(IPHASE).EQ.0) GO TO 20
55     WRITE (6,601) SMNIN,SMN,HN,RHO,A,VEL,CL,TIME,NLEGS,DELT,WF,XLOIT
      601   FORMAT(/2X,7HSMNIN =E14.7,4X,5HSMN =E14.7,5X,4HHN =
1      E14.7,4X,5HRHO =E14.7,6X,3HA =E14.7/4X,5HVEL =E14.7,5X,4HCL =,

```

```

      2 E14.7,3X,6HTIME =E14.7,2X,7HNLEGS =12,15X,6HDELT =E14.7/5X,
      3 4HWF =E14.7,2X,7HXLUIT =E14.7/)
60      DO 90 LEG=1,NLEGS
      IF (SMNIN.GT.0.0) GO TO 30
C-----PERFORM OPTIMUM LOITER MACH NO. SEARCH.
      CALL GOLDEN(.3,.9,3,8,XX,Y,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
65      SMN=XX
      VEL=SMN*A
      IF (LEG.EQ.1) CLDUM=2.*W/(RHO*SWING*VEL**2)
      CL=CLDUM
      GO TO 40
70      C---OPTION TO INCREASE ALTITUDE AS A/C WEIGHT DECREASES, MACH KEPT CONST
      30      RHO1=RHO
      RHO2=2.*W/(CL*SWING*VEL**2)
      CALL TAIN(THTAB,SHTAB,HN,SH,10,1,NERR,DMON)
      HN=HN+2.*SH*(RHO1-RHO2)/(RHO1+RHO2)
75      CALL AT62(HN,ANS)
      A=ANS(4)
      VEL=SMN*A
      RHO=2.*W/(CL*SWING*VEL**2)
C---END OF INCREASE ALTITUDE OPTION
80      40      IAO=8
      CALL TRAJ01(2)
      CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
85      DRAG=.5*RHO*SWING*CD*VEL**2
      CALL TRAJ01(2)
      CALL PROP(ICALC,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
90      TN=THRUST*EN
      TNU=THRSTU*EN
C-----CHECK LOITER RADIUS.
      RALOIT=X(IPHASE)
      IF (RALOIT.LE.0.0) GO TO 60
95      NZLOIT=(.5*CL*RHO*SWING*VEL**2+TN*SIN(ALPHA/57.3))/W
      IF (NZLOIT.LT.1.00001) NZLOIT=1.00001
      COSPHI=1./NZLOIT
C-----RADIUS FOR OPTIMUM LOITER.
      RLOIT=(5.11E-6*VEL**2*COSPHI)/SQRT(1.-COSPHI**2)
      IF (RLOIT.LE.RALOIT) GO TO 60
      DO 50 J=1,3
      DRAGP=DRAG
      CL=(W/(SWING*.5*RHO*VEL**2))*SQRT(1.+(5.11E-6*VEL**2/RALOIT)**2)
      IAO=8
05      CALL TRAJ01(2)
      CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
      DRAG=.5*RHO*SWING*CD*VEL**2
10      IF (ABS(DRAG/DRAGP-1.0).LE.0.03) GO TO 60
      IF (J.EQ.3) WRITE (6,602) IPHASE,LEG
602      FORMAT(/55H INSUFFICIENT THRUST FOR REQUIRED LOITER RADIUS. PHASE
      1*,I2,7H LEG=,I2/)
      CALL TRAJ01(2)

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

15      CALL PROP(ICALC,NEKRROR,IGLO,KGPRNT)
      IF (NEKRROR.GE.2) RETURN
      CALL TRAJ01(1)
      TN=THRUST*EN
      TNU=THRSTU*EN
20      50 CONTINUE
      60 TEST=DRAG/TN
      IF ((ABS(TEST-1.0).LE.0.05).AND.(SMNIN.LE.3.0)) GO TO 70
      IF (TEST.GT.1.0) TN=DRAG
      CD=2.*TN/(KHO*SWING*VEL**2)
      DFAG=.5*RHO*SWING*CD*VEL**2
25      IF (KERRUR.EQ.2) WRITE (6,603) IPHASE,LEG,TEST
      603 FORMAT(//17H IN LOITER, PHASE,I2,3H, LEG,I2,8H, DRAG =,F6.3,
1 7H OF TN./28H CD CHANGED TO MATCH THRUST./)
      70 DELWF=SFC*TN*DELT/60.
      30 WF=WF+DELWF
      W=W-DELWF
      DELX=VEL*DELT*60./6380.
      XLOIT=XLOIT+DELX
      IF (IPRT(IPHASE).EQ.0) GO TO 80
      35 WRITE (6,604) LEG,SMNIN,SMN,ALPHA,CL,CDO,CD,KP,RHO1,RHO2,SH,HN,A,
1 RHO,VEL,DRAG,TN,DELWF,W,F,W,DELX,XLOIT,SFC,NZLOIT,COSPHI,RLOIT
      604 FORMAT(/4X,5HLEG =I2/2X,7HSMNIN =E14.7,4X,5HSMN =E14.7,
1 2X,7HALPHA =E14.7,5X,4HCL =E14.7,4X,5HCDO =E14.7/5X,4HCD =E14.7,
2 5X,4HKP =E14.7,3X,6HRHO1 =E14.7,3X,6HRHO2 =E14.7,5X,4HSH =E14.7/
40 3 5X,4HHN =E14.7,6X,3HA =E14.7,4X,5HRHO =E14.7,4X,5HVEL =E14.7,
4 3X,6HDRAG =E14.7/5X,4HTN =E14.7,2X,7HDELWF =E14.7,5X,4HWF =E14.7,
5 6X,3HW =E14.7,3X,6HDELX =E14.7/2X,7HXLOIT =E14.7,4X,5HSFC =E14.7,
6 1X,8HNZLOIT =E14.7,1X,8HCOSPHI =E14.7,2X,7HRLQIT =E14.7/)
      80 IF (W.GT.WPL) GO TO 90
      45 IF (KERRUR.EQ.2) WRITE (6,605) IPHASE,LEG
      605 FORMAT(//27H W.LE.WPL IN LOITER. PHASE=,I2,7H LEG=,I2/)
      RETURN
      90 CONTINUE
      Q=.5*RHO*VEL**2
      BLTIME=BLTIME+TIME/60.
      BLRANG=BLRANG+XLOIT
      C-----TEST FOR WEAPONS DROP
      IF (IA(IPHASE).EQ.0) GO TO 100
      W=W-WMISS
      IAS=0
155 C-----TEST FOR BOMBS DROP
      100 IF (IB(IPHASE).EQ.0) GO TO 110
      W=W-WBOMBS
      IBS=0
160 C TEST FOR AMMO DROP
      110 IF (IA(IPHASE).EQ.0) GO TO 120
      W=W-WAMMUN
      IAS=0
      120 SMNT(IPHASE)=SMN
165 HNT(IPHASE)=HN
      CLT(IPHASE)=CL
      ALPHAT(IPHASE)=ALPHA
      WFT(IPHASE)=WF
      TIMET(IPHASE)=TIME
170 VELT(IPHASE)=VEL
      SFCT(IPHASE)=SFC

```


SUBROUTINE LOITER

76/76 OPT=2

FTN 4.5+410

175 TNT(IPHASE)=TN
CDT(IPHASE)=CD
RLDT(IPHASE)=RLD
WEELT(IPHASE)=W
WART(IPHASE)=WTOT
QT(IPHASE)=Q
SFCUT(IPHASE)=SFCU
18C TNUT(IPHASE)=TNU
CDINST(IPHASE)=CDINSP
PRTOTT(IPHASE)=PRTOT
XT(IPHASE)=XLOIT
RETURN
END

ORIGINAL PAGE IS
OF POOR QUALITY

```

1      SUBROUTINE DESCNT(ICALC,NERROR,IGEO,KGPRNT)
      REAL KP
      COMMON /TRAJCH/ ALPHA,ARW,CD,CDL,CDD,CL,DESLF,DRAG,EN,HN,KP,
5      1 RLD,SMN,QMAX,RANGE,SFC,SWING,THRUST,TW,ULTLF,W,WAMMUN,WETANK,
      2 WFUEL,WGTOWT,WMISS,WTOT,WPLWT,HSTART(12),HSTART(12),TIM(12),
      3 PSIGT(12),TDOST(12),NZST(12),PSIT(12),TDOIT(12),NZIT(12),
      4 WBOMBS,WFEET,SFCU,THRSTU,CDINSP,PRTOT,WKFUEL,CRMACH,FLTO,X(12),
      5 FLLAND,TENDUR,BLRANG,BLTIME,TIMTO1,TIMTO2,WFTO,PNAME1(12),
      6 PNAME2(12),SMNT(12),HNT(12),WFT(12),TIMET(12),XT(12),CLT(12),
10     7 CDT(12),ALPHAT(12),CLIT(12),CDIT(12),ALPHIT(12),WFUSED,WLAND,
      8 QT(12),RLDT(12),SFCT(12),TNT(12),CET(12),DY(18),
      9 IAG,IPS,ITS,IWS,IBS,NPHASE,IDY(14)
      COMMON /TRAJEX/ ALPIGT(12),CDINST(12),CDLGT(12),CLLGT(12),DECEL,
15     1 DELHP,DELMF,DXCRUS,FLFAC,FRFURE,HEND(12),HENDUR,HMAXP,HMINP,
      2 HPAHT(50),HEND(12),HENDUR,NT(12),PRYOTT(12),RADIT(12),
      3 RADST(12),SF,SFCUT(12),SMMAXP,SMMINP,
      4 TNU(12),VIND(12),VELT(12),VPATHT(50),WART(12),WCOMBP,WEELT(12),
      5 WFRES,WFTOT,WFTO1,WFTO2,WFTAP,WKLAND,XDESC,XGRLAN,FNGMAX,TOL,
20     6 WFIXED,WGCALC,WPL,WGTO,DUMY(82),IA(12),IAS,IB(12),IBREG,IENDUR,
      7 IMISS,IP(12),IPHASE,IPLDT,IPRINT,IPRT(12),IPSIZE,IPSTO1,IPSTO2,
      8 IW(12),IX(12),KERROR,MILCOM,NCRUSE,NLEGCL,NLEGCR,NLEGLO,NMISS,
      9 NPATH,IDUMY(6)
      DIMENSION ANS(4)
      IF (IPRT(IPHASE).NE.0) WRITE (6,600)
25     600 FORMAT(///20H      DESCNT PHASE//)
      HNINT=HNT(IPHASE-1)
      HNFIN=HEND(IPHASE)
      IPS=5
      VEL=VELT(IPHASE-1)
30     NLEGS=5
      DELH=(HNFIN-HNINT)/FLOAT(NLEGS)
      HN=HNINT
      TIME=0.0
      WF=0.0
35     XDESC=0.0
      IF (IPRT(IPHASE).EQ.0) GO TO 10
      WRITE (6,6C1) HNINT,HNFIN,CL,RLD,VEL,HN,DEL,TIME,XDESC,WF
6C1    FORMAT(/2X,7HHNINT =E14.7,2X,7HHNFIN =E14.7,5X,4HCL =
40     1 E14.7,4X,5HRLD =E14.7,4X,5HVEL =E14.7/5X,4HHN =E14.7,3X,6HDEL =,
      2 E14.7,3X,6HTIME =E14.7,2X,7HXDESC =E14.7,5X,4HWF =E14.7/)
10     DO 30 LEG=1,NLEGS
      CALL AT62(HN,ANS)
      RHO=ANS(1)
      A=ANS(4)
45     SMN=VEL/A
      CALL TRAJJ1(2)
      CALL PROP(ICALC,NERROR,IGEO,KGPRNT)
      IF (NERROR..2) RETURN
      CALL TRAJO1(1)
50     TN=THRUST*EN
      TNU=THRSTU*EN
      IAO=9
      CALL TRAJO1(2)
      CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
55     IF (NERROR.GE.2) RETURN
      CALL TRAJO1(1)
      CL=SQRT(CDD/KP)

```

SING=2.*COD/CL
 COSG=SQRT(1.-SING**2)
 60 VEL=SQRT(2.*W*COSG/(CL*SWING*RHD))
 DELT=-DELH/(VEL*SING)
 TIME=TIME+DELT/60.
 DELWF=SFC*TN*DELT/3600.
 WF=WF+DELWF
 65 DELX=VEL*DELT*COSG/6080.
 XDESC=XDESC+DELX
 HN=HN+DELH
 W=W-DELWF
 IF (IPRT(IPHASE).EQ.0) GO TO 20
 70 WRITE (6,602) LEG,RHD,A,SMN,TN,SFC,DELV,VEL,COSG,SING,DELT,TIME,
 1 DELX,XDESC,HN,CL,CD,KP,DELWF,W,F,W
 602 FORMAT(/4X,5HLEG =I2/4X,5HRHD =E14.7,6X,3HA =E14.7,4X,
 1 5HSMN =E14.7,5X,4HTN =E14.7,4X,5HSFC =E14.7/3X,6HDELV =E14.7,4X,
 2 5HVEL =E14.7,3X,6HCOSG =E14.7,3X,6HSING =E14.7,3X,6HDELT =E14.7/
 75 3 3X,6HTIME =E14.7,3X,6HDELX =E14.7,2X,7HXDESC =E14.7,5X,4HHN =,
 4 E14.7,5X,4HCL =E14.7/5X,4HCD =E14.7,5X,4HKP =E14.7,2X,7HDELWF =,
 5 E14.7,5X,4HWF =E14.7,6X,3HW =E14.7/)
 20 IF (W.GT.WPL) GO TO 30
 IF (KERROR.EQ.2) WRITE (6,603) IPHASE,LEG
 80 603 FORMAT(/28H W.LE.WPL IN DESCENT. PHASE=,I2,7H LEG=,I2/)
 RETURN
 30 CONTINUE
 CALL AT62(HN,ANS)
 A=ANS(4)
 85 SMN=VEL/A
 RHD=ANS(1)
 Q=.5*RHD*VEL**2
 BLTIME=BLTIME+TIME/60.
 BLRANG=BLRANG+XDESC
 90 C-----TEST FOR WEAPONS DROP
 IF (IW(IPHASE).EQ.0) GO TO 40
 W=W-WMISS
 IWS=0
 C-----TEST FOR BOMBS DROP
 95 40 IF (IB(IPHASE).EQ.0) GO TO 50
 W=W-WBOMBS
 IBS=0
 C-----TEST FOR AMMO DROP
 50 IF (IA(IPHASE).EQ.0) GO TO 60
 100 W=W-WAMMUN
 IAS=0
 60 IF (IPRT(IPHASE).EQ.0) GO TO 70
 WRITE (6,604) HN,A,SMN,RHD,Q,DXCRUS,W
 604 FORMAT(/5X,4HHN =E14.7,6X,3HA =E14.7,4X,5HSMN =E14.7,
 105 1 4X,5HRHD =E14.7,6X,3HQ =E14.7/1X,6HDXCRUS =E14.7,6X,3HW =E14.7/)
 70 SMNT(IPHASE)=SMN
 HNT(IPHASE)=HN
 CLT(IPHASE)=CL
 ALPHAT(IPHASE)=ALPHA
 110 WFT(IPHASE)=WF
 TIMET(IPHASE)=TIME
 VELT(IPHASE)=VEL
 SFCT(IPHASE)=SFC
 TNT(IPHASE)=TN

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE DESCNT 76/76 OPT=2

FTN 4.5+41D

```
115      CDT(IPHASE)=CD  
        RLD(IPHASE)=RLD  
        WEELT(IPHASE)=W  
        WART(IPHASE)=WTOT  
        QT(IPHASE)=Q  
120      SFCUT(IPHASE)=SFCU  
        TNUT(IPHASE)=TNU  
        CDINST(IPHASE)=CDINSP  
        PRTOTT(IPHASE)=PRTOT  
        XT(IPHASE)=XDESC  
125      RETURN  
        END
```

```

1      SUBROUTINE PATHS(ICALC,NERROR,IGEO,KGPRNT)
      REAL KP
      COMMON /TRAJCM/ ALPHA,ARW,CD,CDL,CDO,CL,DESLF,DRAG,EN,HN,KP,
5      1 RLD,SMN,QMAX,RANGE,SFC,SWING,THRUST,TW,ULTLF,W,WAMMUN,WETANK,
      2 WFUEL,WGTOWT,WMISS,WTOT,WPLWT,HSTART(12),HSTART(12),TIM(12),
      3 PS1GT(12),TDOTST(12),NZST(12),PSIT(12),TDOTIT(12),NZIT(12),
      4 WBOMBS,WFEXT,SFCU,THRSTU,CDINSP,PRTOT,WKFUEL,CRMACH,FLTO,X(12),
      5 FLLAND,TENDUR,BLRANG,BLTIME,TIMT01,TIMID2,WFTO,PNAME1(12),
10     6 PNAME2(12),SMNT(12),HNT(12),WFT(12),TIMET(12),XT(12),CLT(12),
      7 CDT(12),ALPHAT(12),CLIT(12),CDIT(12),ALPHIT(12),WFUSED,WLAND,
      8 QT(12),RLDT(12),SFCT(12),TNT(12),CET(12),DY(18),
      9 IAO,IPS,ITS,IWS,IBS,NPHASE,IDY(14)
      COMMON /TRAJEX/ ALP1GT(12),CDINST(12),CD1GT(12),CL1GT(12),DECEL,
15     1 DELHP,DELMP,DXCRUS,FLFAC,FRFURE,HEND(12),HENDUR,HMAXP,HMINP,
      2 HPATHT(50),MEND(12),MENDUR,NT(12),PRTOTT(12),RADIT(12),
      3 RADST(12),SF,SFCUT(12),SMMAXP,SMMINP,
      4 TNUT(12),VIND(12),VELT(12),VPATHT(50),WART(12),WCOMBP,WEELT(12),
      5 WFRES,WFTOT,WFT01,WFT02,WFTRAP,WKLAND,XDESC,XGRIAN,FWGMAX,TOL,
20     6 WFIXED,WGCALC,WPL,WGTQ,DUMY(82),IA(12),IAS,IB(12),IBREG,IENDUR,
      7 IMISS,IP(12),IPHASE,IPLT,IPRINT,IPRT(12),IPSIZE,IPST01,IPST02,
      8 IW(12),IX(12),KERROR,MILCOM,NCRUSE,NLEGCL,NLEGCR,NLEGLO,NMISS,
      9 NPATH,IDUMY(6)
      DIMENSION ANS(4)
      IPS=IP(IPHASE)
25     TIME=0.0
      XPATH=0.0
      WF=0.0
      IF (IPRT(IPHASE).NE.0) WRITE (6,600) IPS,TIME,XPATH,WF,W
600    FORMAT(///17H          PATH PHASE///4X,5HIPS =12,15X,6HTIME =E14.7,
30     1 2X,7HXPATH =E14.7,5X,4HWF =E14.7,6X,3HW =E14.7/)
      DO 60 I=2,NPATH
      LEG=I-1
      H1=HPATHT(I-1)
      H2=HPATHT(I)
35     HN=.5*(H1+H2)
      DELH=H2-H1
      CALL AT62(HN,ANS)
      RHO=ANS(1)
      A=ANS(4)
40     VM1=VPATHT(I-1)
      VM2=VPATHT(I)
      VM=.5*(VM1+VM2)
      C-----MAKE TEST TO SEE WHETHER VM IS MACH OR VELOCITY.
      IF (VM.GT.10.0) GO TO 10
45     SMN1=VM1
      SMN2=VM2
      SMN=VM
      VEL1=SMN1*A
      VEL2=SMN2*A
50     VEL=SMN*A
      GC TO 20
10     VEL1=VM1
      VEL2=VM2
      VEL=VM
55     SMN1=VEL1/A
      SMN2=VEL2/A
      SMN=VEL/A

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

20   DELV=VEL2-VEL1
C-----CALL PROP TO GET TN, SFC.
60   CALL TRAJ01(2)
      CALL PROP(ICALC,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
      TN=THRUST*EN
65   TNU=THRSTU*EN
C-----CALL AERO TO GET CDO, KP.
      IAO=1
      ALPHA=5.0
      CALL TRAJ01(2)
70   CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
      Q=.5*RHO*VEL**2
      QS=Q*SWING
75   C1=QS/W
      C2=VEL*DELV/(32.2*DELH)+1.0
      C3=TN/W-C1*CDO
      C4=(C2/KP)**2
      C5=SQRT(.25*C4**2-C4*C3/(C1*KP)+C4/C1**2)
80   C-----CALCULATE FIRST APPROXIMATION TO CL.
      CL1=SQRT(-.5*C4+C3/(C1*KP)+C5)
C-----CALL AERO TO GET CD, ALPHA.
      CL=CL1
      IAO=8
85   CALL TRAJ01(2)
      CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
      C6=((TN/W)*COS(ALPHA/57.3)-C1*CD)/C2
90   C-----CALCULATE SECOND APPROXIMATION TO CL.
      CL2=SQRT(1.-C6**2)/C1
C-----CALL AERO TO GET CD, ALPHA.
      CL=CL2
      CALL TRAJ01(2)
95   CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
C-----CALCULATE FLIGHT PATH ANGLE.
      SING=((TN/W)*COS(ALPHA/57.3)-C1*CD)/C2
100  IF ((SING.GE.0.0).OR.(DELH.LE.0.0)) GO TO 30
      NERROR=2
      WRITE (6,601) IPHASE,LEG
601  FORMAT(/////33H *****FATAL ERROR IN PATH. PHASE=,I2,3X,4HLEG=,I2/
1 41H *****INSUFFICIENT THRUST TO FOLLOW PATH.//////)
105  RETURN
30   IF (SING.GT.1.0) SING=1.0
C-----CHECK CAPABILITY TO MATCH PATH ALTITUDE.
      DELHC=VEL*DELV*SING/(32.2*(SING*C2-SING))
      IF (ABS((DELHC-DELH)/DELH).LE.0.2) GO TO 40
      NERROR=2
110  WRITE (6,602) IPHASE,LEG
602  FORMAT(/////33H *****FATAL ERROR IN PATH. PHASE=,I2,3X,4HLEG=,I2/
1 44H *****PATH ALTITUDE DEVIATION EXCEEDS LIMIT.//////)
      RETURN

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

15      40      DELT=DELH/(VEL*SING)
           COSG=SQRT(1.-SING**2)
           DELX=VEL*DELT*COSG/6080.
           DELWF=TN*SFC*DELT/3600.
           TIME=TIME+DELT/60.
20      XPATH=XPATH+DELX
           WF=WF+DELWF
           W=W-DELWF
           IF (IPRT(IPHASE).EQ.0) GO TO 50
           WRITE (6,603) LEG,H1,H2,HN,DELH,SMN1,SMN2,SMN,VEL1,VEL2,VEL,DELV,
25      1 RHO,A,TN,SFC,ALPHA,CDO,KP,Q,C1,C2,C3,C4,C5,CL1,C6,CL2,CL,CD,
           2 SING,COSG,DELHC,DELT,DELX,DELWF,TIME,XPATH,W,F,W
603      FORMAT(/1X,5HLEG =I2/5X,4HH1 =E14.7,5X,4HH2 =E14.7,5X,4HHN =E14.7,
           1 3X,6HDELH =E14.7,3X,6HSMN1 =E14.7/3X,6HSMN2 =E14.7,4X,5HSMN =
           2 E14.7,3X,6HVEL1 =E14.7,3X,6HVEL2 =E14.7,4X,5HVEL =E14.7/
30      3 3X,6HDELV =E14.7,4X,5HRHO =E14.7,6X,3HA =E14.7,5X,4HTN =E14.7,
           4 4X,5HSFC =E14.7/2X,7HALPHA =E14.7,4X,5HCDU =E14.7,5X,4HKP =E14.7,
           5 6X,3HQ =E14.7,5X,4HC1 =E14.7/5X,4HC2 =E14.7,5X,4HC3 =E14.7,
           6 5X,4HC4 =E14.7,5X,4HC5 =E14.7,4X,5HCL1 =E14.7/5X,4HC6 =E14.7,
           7 4X,5HCL2 =E14.7,5X,4HCL =E14.7,5X,4HCD =E14.7,3X,6HSING =E14.7/
35      8 3X,6HCOSG =E14.7,2X,7HDELHC =E14.7,3X,6HDELT =E14.7,3X,6HDELX =
           9 E14.7,2X,7HDELWF =E14.7/3X,6HTIME =E14.7,2X,7HXPATH =E14.7,
           1 5X,4HWF =E14.7,6X,3HW =E14.7/)
50      IF (W.GT.WPL) GO TO 60
           IF (KERROR.EQ.2) WRITE (6,604) IPHASE,LEG
40      604      FORMAT(/125H W.LE.WPL IN PATH. PHASE=,I2,7H   LEG=I2/)
           RETURN
60      CONTINUE
           HN=H2
           SMN=SMN2
45      VEL=VEL2
           BLTIME=BLTIME+TIME/60.
           BLRANG=BLRANG+XPATH
C-----TEST FOR WEAPONS DROP
           IF (IW(IPHASE).EQ.0) GO TO 70
50      W=W-WMISS
           IWS=0
C-----TEST FOR BOMBS DROP
70      IF (IB(IPHASE).EQ.0) GO TO 80
           W=W-WBOMBS
55      IBS=0
C-----TEST FOR AMMO DROP
80      IF (IA(IPHASE).EQ.0) GO TO 90
           W=W-WAMMUN
           IAS=0
60      90      IF (IPRT(IPHASE).EQ.0) GO TO 100
           WRITE (6,605) W,HN,SMN,VEL
605      FORMAT(/6X,3HW =E14.7,5X,4HHN =E14.7,4X,5HSMN =E14.7,
           1 4X,5HVEL =E14.7/)
65      100      SMNT(IPHASE)=SMN
           HNT(IPHASE)=HN
           CLT(IPHASE)=CL
           ALPHAT(IPHASE)=ALPHA
           WFT(IPHASE)=WF
           TIMET(IPHASE)=TIME
           VELT(IPHASE)=VEL
70      SFC(IPHASE)=SFC

```

SUBROUTINE PATHS

76/76 OPT=2

FTN 4.5+410

0

```
175      TNT(IPHASE)=TN  
        CDT(IPHASE)=CD  
        RLD(IPHASE)=RLD  
        WEELT(IPHASE)=W  
        WART(IPHASE)=WTOT  
        QT(IPHASE)=Q  
        SFCUT(IPHASE)=SFCU  
180      TNU(IPHASE)=TNU  
        CDINST(IPHASE)=CDINSP  
        PRTOT(IPHASE)=PRTOT  
        XT(IPHASE)=XPATH  
        RETURN  
        END
```


ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE TRAJ00

75/76 DPT=2

FTN 4.5+410

01

```
1      SUBROUTINE TRAJ00
C      OUTPUT OF RESULTS FROM TRAJECTORY ROUTINE
      REAL NZST,NZIT,NZLG,MENDUR,LOIT
      COMMON /TRAJCM/ ALPHA,ARW,CD,CDL,CDU,CL,DESLF,DRAG,EN,HN,KP,
5      1 RLD,SMN,QMAX,RANGE,SFC,SWING,THRUST,TW,ULTLF,W,WAMMUN,WETANK,
      2 WFUEL,WGTOWT,WMISS,WTOT,WPLWT,MSTART(12),HSTART(12),TIM(12),
      3 PSIGT(12),TDOTST(12),NZST(12),PSIT(12),TDOTIT(12),NZIT(12),
      4 WBOMBS,WTEXT,SFCU,THRSTU,CDINSP,PRTOT,WKFUEL,CRMACH,FLTO,X(12),
10     5 FLLAND,TENDUR,BLRANG,BLTIME,TIMTQ1,TIMTQ2,WFTQ,PNAME1(12),
      6 PNAME2(12),SMNT(12),HNT(12),WFT(12),TIMET(12),XT(12),CLT(12),
      7 CDT(12),ALPHAT(12),CLIT(12),CDIT(12),ALPHIT(12),WFUSED,WLAND,
      8 QT(12),RLDT(12),SFC(12),TNT(12),CET(12),DY(18),
      9 IAO,IPS,ITS,IWS,IBS,NPHASE,IDY(14)
      COMMON /TRAJEX/ ALPIGT(12),CDINST(12),CDIGT(12),CLIGT(12),DECEL,
15     1 DELHP,DELMP,DXCRUS,FLFAC,FRFURE,HEND(12),HENDUR,HMAXP,HMINP,
      2 HPATH(50),MEND(12),MENDUR,NT(12),PRTOTT(12),RADIT(12),
      3 RADST(12),SF,SFCUT(12),SMMAXP,SMMINP,
      4 TNUT(12),VIND(12),VELT(12),VPATH(50),WART(12),WCOMBP,WEELT(12),
20     5 WFRES,WFTOT,WFTQ1,WFTQ2,WFTAP,WKLAND,XDESC,XGRLAN,FWGMAX,TOL,
      6 WFIXED,WGCALC,WPL,WGTO,DUMY(82),IA(12),IAS,IB(12),IBREG,IENDUR,
      7 IMISS,IP(12),IPHASE,IPLT,IPRINT,IPRT(12),IPSIZE,IPSTQ1,IPSTQ2,
      8 IW(12),IX(12),KERROR,MILCOM,NCRUSE,NLEGCL,NLEGCR,MLEGLO,NMISS,
      9 NPATH,IDUMY(6)
      DATA CUMB/4HCOMB/
25     DATA LOIT/4HLOIT/
      WRITE (6,600) IMISS,WPL
600    FORMAT(1H1,24X,17HTRAJECTORY OUTPUT//
      1 24X,7HMISSIOND,I2,12H (PAYLOAD =,F8.3,4H LB) //
      2 67H PHASE      M      H      CL      ALPHA      WFUEL      TIME
30     3 VEL/9X,57HSFC(I) THRUST(I)  CD      L/D      W      WA
      4 0/9X,57HSFC(U) THRUST(U)  CDINST      PR
      5X)
      DO 10 IPHASE=1,NPHASE
      WRITE (6,601) PNAME1(IPHASE),PNAME2(IPHASE),SMNT(IPHASE),
35     1 HNT(IPHASE),CLT(IPHASE),ALPHAT(IPHASE),WFT(IPHASE),TIMET(IPHASE),
      2 VELT(IPHASE),SFC(IPHASE),TNT(IPHASE),CDT(IPHASE),RLDT(IPHASE),
      3 WEELT(IPHASE),WART(IPHASE),QT(IPHASE),SFCUT(IPHASE),TNUT(IPHASE),
      4 CDINST(IPHASE),PRTOTT(IPHASE),XT(IPHASE)
601    FORMAT(1X,2A4,F4.2,F11.0,F10.4,F8.2,F10.1,F8.2,F8.0/
40     1 F13.2,F11.0,F10.4,F8.2,F10.1,F8.2,F8.0/
      2 F13.2,F11.0,F10.4,18X,F8.2,F8.0)
10     CONTINUE
      WRITE (6,602) WFTQ1,WFTOT,WFTQ2,WTEXT,WFUSED,WFUEL,WFRES,WFTAP
602    FORMAT(1H1,20X,12HFUEL SUMMARY//14H TAKEOFF FUEL/
45     1 9X,7HWFTQ1 =,F8.0,5X,12HTOTAL FUEL =,F8.0/
      2 9X,7HWFTQ2 =,F8.0,5X,12H EXTERNAL =,F8.0/
      3 16H MISSION FUEL =,F8.0,5X,12H INTERNAL =,F8.0/
      4 16H RESERVE FUEL =,F8.0/16H TRAPPED FUEL =,F8.0)
      J=0
50     NZLG=1.0
      TDOTLG=0.0
      RADLG=0.0
      PSS=0.0
      DO 30 IPHASE=1,NPHASE
55     IF (PNAME1(IPHASE).NE.COMB) GO TO 30
      IF (J.NE.0) GO TO 20
      WRITE (6,603)
```

```

603  FORMAT(///19X,28HADDITIONAL COMBAT PARAMETERS//13X,
      1 58HCONDITIGNS PS NZ TDOT RADIUS ALPHA CL CD)
60  J=1
20  WRITE (6,604) SMNT(IPHASE),PS1GT(IPHASE),NZ1G,TDOT1G,RAD1G,
      1 ALP1GT(IPHASE),CL1GT(IPHASE),CD1GT(IPHASE),HNT(IPHASE),PSS,
      2 NZST(IPHASE),TDOTST(IPHASE),RADST(IPHASE),ALPHAT(IPHASE),
      3 CLT(IPHASE),CDT(IPHASE),PSIT(IPHASE),NZIT(IPHASE),TDOTIT(IPHASE),
65  4 RADIT(IPHASE),ALPHIT(IPHASE),CLIT(IPHASE),CDIT(IPHASE),
      5 CET(IPHASE)
604  FORMAT(/3H M=,F5.2,15H 1 G FLIGHT,F8.1,F6.2,F7.2,F8.0,F7.2,
      1 F7.3,F8.4/3H H=,F6.0,13H SUSTAINED,F9.1,F6.2,F7.2,F8.0,F7.2,
      2 F7.3,F8.4/13X,10HMAX. INST.,F8.1,F6.2,F7.2,F8.0,F7.2,F7.3,F8.4/
70  3 13X,15HCOMBAT ENERGY=,E13.6)
      CONTINUE
30  WRITE (6,605) BLTIME,BLRANG,FLTO,FLLAND,XGRLAN,WLAND,WGTO,W,
      1 MENDUR,HENDUR,TENDUR
605  FORMAT(///12X,23H BLOCK TIME =,F7.3,7H HOURS/
      1 35H BLOCK RANGE =,F7.1,7H N. M./
75  2 35H TAKEOFF FIELD LENGTH(TOTAL RUN) =,F7.0,6H FEET/
      3 35H LANDING FIELD LENGTH(TOTAL RUN) =,F7.0,6H FEET/
      4 35H LANDING FIELD LENGTH(GROUND RUN) =,F7.0,6H FEET/
      5 35H WEIGHT FOR LANDING CALCULATION =,F8.0,7H POUNDS/
80  6 35H TAKEOFF WEIGHT =,F8.0,7H POUNDS/
      7 35H LANDING WEIGHT =,F8.0,7H POUNDS/
      8 35H ENDURANCE MACH NO. =,F5.1/
      9 35H ENDURANCE ALTITUDE =,F7.0,6H FEET/
      1 35H ENDURANCE TIME =,F7.3,7H HOURS)
85  DO 40 IPHASE =1,NPHASE
      IF (PNAME1(IPHASE).NE.LOIT) GO TO 40
      WRITE (6,606) X(IPHASE)
606  FORMAT(20X,15HLOITER RADIUS =,F6.2,8H N. M.)
40  CONTINUE
      RETURN
      END
90

```

```

1      SUBROUTINE TRPLOT(NERROR,IGEO,KGPRNT)
      REAL LIFT,NZS,NZI
      COMMON /TRAJCM/ ALPHA,ARW,CD,CDL,CDO,CL,DESLF,DRAG,EN,HN,KP,
5      1 RLD,SMN,QMAX,RANGE,SFC,SWING,THRUST,TW,ULTLF,W,WAMMUN,WETANK,
      2 WFUEL,WGTOWT,WMISS,WTOT,WPLWT,MSTART(12),HSTART(12),TIM(12),
      3 PSIGT(12),TDOTST(12),NZST(12),PSIT(12),TDOTIT(12),NZIT(12),
      4 WBOMBS,WFEXT,SECU,THRSTU,CDINSP,PRTOT,WKFUEL,CRMACH,FLTO,X(12),
      5 FLLAND,TENDUR,BLRANG,BLTIME,TIMTO1,TIMTO2,WFTO,PNAME1(12),
10     6 PNAME2(12),SMNT(12),HNT(12),WFT(12),TIMET(12),XT(12),CLT(12),
      7 CDT(12),ALPHAT(12),CLIT(12),CDIT(12),ALPHIT(12),WFUSED,WLAND,
20     8 QT(12),RLDT(12),SFCT(12),TNT(12),CET(12),DY(18),
      9 IAO,IPS,ITS,IWS,IBS,NPHASE,IDY(14)
      COMMON /TRAJEX/ ALPGT(12),CDINST(12),CDIGT(12),CLIGT(12),DECEL,
15     1 DELHP,DELMP,DXCRUS,FLFAC,FRFURE,HEND(12),HENDUR,HMAXP,HMINP,
      2 HPAHT(50),HEND(12),HENDUR,NT(12),PRTOT(12),RADIT(12),
      3 RADST(12),SF,SFCUT(12),SMMAXP,SMHINP,
      4 TNUT(12),VIND(12),VELT(12),VPATHT(50),WART(12),WCOMBP,WEELT(12),
      5 WFRES,WFTOT,WFTOL,WFTO2,WFTAP,WKLAND,XDESC,XGRAN,FWGMAX,TOL,
      6 WFIXED,WGALC,WPL,WGTO,DUMY(82),IA(12),IAS,IB(12),IBREG,IENDUR,
20     7 IMISS,IP(12),IPHAZE,IPLT,IPRINT,IPRT(12),IPSIZE,IPSTO1,IPSTO2,
      8 IW(12),IX(12),KERROR,HILCOM,NCRUSE,NLEGCL,NLEGCR,NLEGLO,NMISS,
      9 NPATH,IDUMY(6)
      DIMENSION ANS(4)
      DATA COMB/4HCOMB/
25     C-----IPLT = 0, NO PRINT, NO PLOT
      C-----IPLT = 1, PRINT, NO PLOT
      C-----IPLT = 2, NO PRINT, PLOT
      C-----IPLT = 3, PRINT, PLOT
      C-----WCOMBP .LE. 0.0, W = W FROM LAST COMBAT PHASE
30     C-----WCOMBP BETWEEN 0.0 AND 1.0, W = WCOMBP*WGTO
      C-----WCOMBP .GT. 1.0, W = WCOMBP
      C-----CALCULATE COMBAT PARAMETERS.
      J=0
      DO 10 IPHAZE=1,NPHASE
35     IF (PNAME1(IPHAZE).NE.COMB) GO TO 10
      J=IPHAZE
10     CONTINUE
      IF (J.EQ.0) IPS=2
      IF (J.GT.0) IPS=IP(J)
40     JMI=J-1
      IF (JMI.EQ.0) JMI=1
      IF ((WCOMBP.LE.0.0).AND.(J.EQ.0)) W=.7*WGTO
      IF ((WCOMBP.LE.0.0).AND.(J.GT.0)) W=WEELT(JMI)
      IF ((WCOMBP.GT.0.0).AND.(WCOMBP.LE.1.0)) W=WCOMBP*WGTO
45     IF (WCOMBP.GT.1.0) W=WCOMBP
      NH=(HMAXP-HMINP)/DELHP+1.0
      NM=(SMMAXP-SMHINP)/DELHP+1.0
      IF ((IPLT.EQ.1).OR.(IPLT.EQ.3)) WRITE (6,600) HMINP,HMAXP,
50     1 DELHP,SMHINP,SMMAXP,DELHP,WCOMBP,IPLT,QMAX,W
600    FORMAT(1H1,9H HMINP =F7.0/10H HMAXP =F7.0/10H DELHP =F7.0/
      1 10H SMHINP =F6.2/10H SMMAXP =F6.2/10H DELHP =F6.2/
      2 10H WCOMBP =F9.2/10H IPLT =I2/10H QMAX =F7.0/
      3 10H W =F8.1/105H H H PSIG NZS TDOT:
55     4 NZI PSI TDOTI Q/)
      IF (IPLT.GE.2) WRITE (42,601) NH,NM,DELHP,DELMP,QMAX
601    FORMAT(2I8,3F8.2)
      DO 40 IH=1,NH

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

      HN=FLOAT(IH-1)*DELHP
      CALL AT62(HN,ANS)
60      RHO=ANS(1)
      A=ANS(4)
      DO 40 IM=1,NM
      SMN=FLOAT(IM-1)*DELMF+SMHNP
      VEL=SMN*A
65      Q=.5*RHO*VEL**2
      QS=Q*SWING
      CL=W/QS
      IAO=8
      CALL TRAJ01(2)
      CALL AERO(2,NERROR,IGED,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
      DRAG=CD*QS
      CALL TRAJ01(2)
75      CALL PROP(2,NERROR,IGED,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
      TN=THRUST*EN
      PSIG=(TN-DRAG)*VEL/W
80      NZS=1.001
      IF (PSIG.LE.0.) GO TO 20
      IAO=4
      CALL TRAJ01(2)
      CALL AERO(2,NERROR,IGED,KGPRNT)
85      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
      LIFT=CL*QS
      NZS=(LIFT+TN*SIN(ALPHA/57.3))/W
      IF (NZS.LE.DESLF) GO TO 20
90      IAO=6
      CALL TRAJ01(2)
      CALL AERO(2,NERROR,IGED,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
95      NZS=DESLF
      IF (NZS.LE.1.0) NZS=1.001
      PHI=ARCOS(1.0/NZS)
      RADIUS=VEL**2/(32.2*TAN(PHI))
      TDOTS=57.3*VEL/RADIUS
100      IAO=5
      CALL TRAJ01(2)
      CALL AERO(2,NERROR,IGED,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
105      LIFT=CL*QS
      NZI=(LIFT+TN*SIN(ALPHA/57.3))/W
      IF (NZI.LE.DESLF) GO TO 30
      IAO=6
      CALL TRAJ01(2)
110      CALL AERO(2,NERROR,IGED,KGPRNT)
      IF (NERROR.GE.2) RETURN
      CALL TRAJ01(1)
      NZI=DESLF
30      IF (NZI.LE.1.0) NZI=1.001

```

```

15      DRAG=CD*QS
        PSI=(TN*COS(ALPHA/57.3)-DRAG)*VEL/W
        PHI=ARCOS(1.0/NZI)
        RADIUS=VEL**2/(32.2*TAN(PHI))
        TDOTI=57.3*VEL/RADIUS
20      IF ((IPLT.EQ.1).OR.(IPLT.EQ.3)) WRITE (6,602) HN,SMN,
        1 PSI, NZS, TDOTS, NZI, PSI, TDOTI, Q
        IF (IPLT.GE.2) WRITE (42,602) HN,SMN,PSI, NZS, TDOTS, NZI, PSI,
        1 TDOTI, Q
602     FORMAT(F7.0,F6.2,1P7E11.3)
25      40 CONTINUE
C-----LANDING FIELD LENGTHS AT ALTITUDES OF 0, 4000, 8000, AND 12000 FT.
        IF ((IPLT.EQ.1).OR.(IPLT.EQ.3)) WRITE (6,603)
603     FORMAT(1H1,8X,33HLANDING FIELD LENGTH CALCULATIONS//
30         1 6cH ALT      WMIN      FLLAN(WMIN)      WGTO      FLLAN(W
        2GTO)//)
        WMIN=WGTO-WFTOT-WPL
        IAD=12
        DO 50 I=1,4
        HN=(I-1)*4000
35         CALL AT62(HN,ANS)
        RHO=ANS(1)
        CALL TRAJ01(2)
        CALL AERO(2,NERROR,IGED,KGPRNT)
        IF (NERROR.GE.2) RETURN
40         CALL TRAJ01(1)
        VSTALL=SQRT(2.*WMIN/(SWING*CL*RHO))
        VSCREN=1.3*VSTALL
        VTDOWN=1.15*VSTALL
        XAIR=((VSCREN**2-VTDOWN**2)/64.4+50.)*RLD
45         XGRLAN=VTDOWN**2/(64.4*DECEL)
        FLLAN1=(XAIR+XGRLAN)/FLFAC
        VSTALL=SQRT(2.*WGTO/(SWING*CL*RHO))
        VSCREN=1.3*VSTALL
        VTDOWN=1.15*VSTALL
50         XAIR=((VSCREN**2-VTDOWN**2)/64.4+50.)*RLD
        XGRLAN=VTDOWN**2/(64.4*DECEL)
        FLLAN2=(XAIR+XGRLAN)/FLFAC
        IF ((IPLT.EQ.1).OR.(IPLT.EQ.3)) WRITE (6,604) HN,WMIN,FLLAN1,
        1 WGTO,FLLAN2
55         604 FORMAT(F7.0,4E15.6)
        IF (IPLT.GE.2) WRITE (42,604) HN,WMIN,FLLAN1,WGTO,FLLAN2
        50 CONTINUE
        RETURN
        END

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

1      SUBROUTINE GOLDEN(XMIN,XMAX,IXY,NMAX,XX,Y,NEERROR,IGEO,KGPRNT)
C-----THIS ROUTINE USES THE GOLDEN SECTION METHOD TO FIND
C      THE X CORRESPONDING TO THE MAXIMUM Y IN THE INTERVAL
C      BETWEEN XMIN AND XMAX.
5      C-----IF IXY = 1, THEN X = ALT. AND Y = BREG. FACTOR
C-----IF IXY = 2, THEN X = ALT. AND Y = ENDUR. FACTOR
C-----IF IXY = 3, THEN X = MACH AND Y = ENDUR. FACTOR
      COMMON /TRAJCM/ ALPHA,ARW,CD,CDL,CDU,CL,DESLF,DRAG,EN,HN,KP,
1      RLD,SMN,QMAX,RANGE,SFC,SWING,THRUST,TW,ULTLF,W,WAHMUN,WETANK,
10     2 WFUEL,WGTOWT,WMISS,WTOT,WPLWT,HSTART(12),HSTART(12),TIM(12),
      3 PS1GT(12),TDOTST(12),NZST(12),PSIT(12),TDOTIT(12),NZIT(12),
      4 WBOHBS,WFEET,SFCU,THRSTU,CDINSP,PRTOT,WKFUEL,CRMACH,FLTO,X(12),
      5 FLLAND,TENDUR,BLRANG,BLTIME,TIMTOL,TIMTO2,WFTO,PNAME1(12),
15     6 PNAME2(12),SMNT(12),HNT(12),WFT(12),TIMET(12),XT(12),CLT(12),
      7 CDT(12),ALPHAT(12),CLIT(12),CUIT(12),ALPHIT(12),WFUSED,WLAND,
      8 QT(12),RLDT(12),SFCT(12),TNT(12),CET(12),DY(18),
      9 IAO,IPS,ITS,IWS,IBS,NPHASE,IOY(14)
      COMMON /TRAJEX/ ALP1GT(12),CDINST(12),CD1GT(12),CL1GT(12),DECEL,
20     1 DELHP,DELMP,DXCRUS,FLFAC,FRFURE,HEND(12),HENDUR,HMAXP,HMINP,
      2 HPAHT(50),HEND(12),HENDUR,NT(12),PRTOT(12),RADIT(12),
      3 KADST(12),SF,SFCUT(12),SMHXP,SMHNP,
      4 TNUT(12),VIND(12),VELT(12),VPATHT(50),WART(12),WCOMBP,WEELT(12),
25     5 WFRES,WFTOT,WFTOL,WFTO2,WFTRAP,WKLAND,XDESC,XGRLAN,FWGMAX,TOL,
      6 WFIXED,WGCALC,WPL,WGTO,DUMY(82),IA(12),IAS,IB(12),IBREG,IENDUR,
      7 IMISS,IP(12),IPHASE,IPLOT,IPRINT,IPRT(12),IPSIZE,IPSTOL,IPSTO2,
25     8 Iw(12),Ix(12),KERROR,MILCOM,NCRUSE,NLEGCL,NLEGCR,NLEGLO,NMISS,
      9 NPATH,IDUMY(6)
      DIMENSION ANS(4)
      IPSAVE=IPS
30     KCALC=2
      IAO=8
      IPS=4
      IF ((IPRINT.GE.2).AND.(IXY.EQ.1)) WRITE (6,600)
      IF ((IPRINT.GE.2).AND.(IXY.EQ.2)) WRITE (6,601)
35     IF ((IPRINT.GE.2).AND.(IXY.EQ.3)) WRITE (6,602)
600    FORMAT(/23H SEARCH FOR BREG. ALT./23H ALT.      BREG. FACT.,
1      1 5X,4HMACH,8X,2HCL,8X,2HCD,7X,6HTHRUST,3X,3HSFC)
601    FORMAT(/23H SEARCH FOR ENDUR. ALT./23H ALT.      ENDUR. FACT.,
1      1 5X,4HMACH,8X,2HCL,8X,2HCD,7X,6HTHRUST,3X,3HSFC)
40     602    FORMAT(/23H SEARCH FOR ENDUR. MACH/23H MACH  ENDUR. FACT.,
1      1 5X,4HMACH,8X,2HCL,8X,2HCD,7X,6HTHRUST,3X,3HSFC)
      X1=XMIN
      X2=XMAX
      XX=X1
45     N=0
      XBEST=XX
      YBEST=-1.
      IF (IXY.LE.2) HN=XX
      IF (IXY.EQ.3) SMN=XX
50     IRETRN=1
C-----
C      START OF FUNCTION EVALUATION
C-----
10     N=N+1
55     CALL AT62(HN,ANS)
      RHO=ANS(1)
      A=ANS(4)

```

```

        VEL=SMN*A
        CL=2.*W/(RHO*SWING*VEL**2)
60      CALL TRAJ01(2)
        CALL AERO(KCALC,NERROR,IGED,KGPRNT)
        IF (NERROR.GE.2) RETURN
        CALL TRAJ01(1)
        DRAG=.5*RHO*SWING*CD*VEL**2
65      CALL TRAJ01(2)
        CALL PRJP(KCALC,NERROR,IGED,KGPRNT)
        IF (NERROR.GE.2) RETURN
        CALL TRAJ01(1)
        IF (IXY.EQ.1) Y=3600.*VEL*RLD/SFC
        IF (IXY.GE.2) Y=RLD/SFC
        IF (NERROR.EQ.1) Y=-1.0
        IF (IPRINT.GE.2) WRITE (6,603) XX,Y,SMN,CL,CD,THRUST,SFC
603      FORMAT(1X,G9.3,G13.4,G14.3,2G12.4,G13.6,G12.3)
        IF (Y.GT.YBEST) XBEST=XX
75      IF (Y.GT.YBEST) YBEST=Y
        C-----
        C      END OF FUNCTION EVALUATION
        C-----
        GO TO (20,30,40,60),IRETRN
80      20      Y1=Y
        XX=X2
        IF (IXY.LE.2) HN=XX
        IF (IXY.EQ.3) SMN=XX
        IRETRN=2
85      GO TO 10
        30      Y2=Y
        TAU=.5*(SQRT(5.)-1.)
        X3=X2-TAU*(X2-X1)
        XX=X3
90      IF (IXY.LE.2) HN=XX
        IF (IXY.EQ.3) SMN=XX
        IRETRN=3
        GO TO 10
        40      Y3=Y
95      50      X12=.5*(X1+X2)
        DELX=ABS(X12-X3)
        X4=X12+DELX
        IF (X3.GT.X12) X4=X12-DELX
        XX=X4
        IF (IXY.LE.2) HN=XX
        IF (IXY.EQ.3) SMN=XX
        IRETRN=4
        GO TO 10
        60      Y4=Y
105      IF (Y3.GE.Y4) GO TO 90
        IF (X4.LT.X3) GO TO 70
        X1=X3
        Y1=Y3
        GO TO 80
110      70      X2=X3
        Y2=Y3
        80      X3=X4
        Y3=Y4
        GO TO 110

```

ORIGINAL PAGE IS
OF POOR QUALITY

```
115      90      IF (X3.LT.X4) GO TO 100
          X1=X4
          Y1=Y4
          GO TO 110
120      100      X2=X4
          Y2=Y4
110      110      IF (N.LE.NMAX) GO TO 50
          XX=XBEST
          Y=YBEST
          IF (Y.EQ.-1.0) NERROR=2
125      125      IF (NERROR.EQ.2) WRITE (6,604) XX,Y,IXY
          604      FORMAT(35H ***FATAL ERROR IN GOLDEN ROUTINE./
1 5X,3HX =,E12.5,2X,3HY =,E12.5,2X,5HIXY =,I5)
          IF ((IPRINT.GE.2).OR.(NERROR.EQ.2)) WRITE (6,605) XX,Y
130      605      FORMAT(/35H MACH OR ALTITUDE SEARCH CONCLUDED./
1 5X,3HX =,E12.5,2X,3HY =,E12.5)
          IPS=IPSAVE
          RETURN
          END
```


REFERENCES

1. Levin, A. D.; Castellano, C. R.; and Hague, D. S.: High Performance Dash on Warning Air Mobile Missile System. NASA TM X-62,479, Sept. 1975.
2. Nelms, W. P., Jr.; Murphy, R.; and Barlow, A.: Preliminary Analysis of Long-Range Aircraft Designs for Future Heavy Airlift Missions. NASA TM X-73,131, June 1976.
3. Tauber, M. E.; and Saunders, R. C. III: Conceptual Design of Multimission Aircraft Armed with Advanced Weapon Systems. NASA TM X-73,165, Aug. 1976 (Secret).
4. Torenbeek, E.: An Analytic Expression for the Balanced Field Length. Aircraft Performance-Prediction Methods and Optimization. Williams, J., ed., AGARD Lecture Series no. 56, April 1972.
5. Miele, A.: Flight Mechanics: Theory of Flight Paths. Vol. 1. Addison-Wesley Publishing Co., Inc., 1962.
6. Perkins, C. D.; and Hage, R. E.: Airplane Performance Stability and Control, John Wiley & Sons, Inc., 1949.

ORIGINAL PAGE IS
OF POOR QUALITY