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.

Produced by the NASA Center for Aerospace Information (CASI)
Trajectory Module of the NASA Ames Research Center Aircraft Synthesis Program ACSYNT

Michael E. Tauber and John A. Paterson

July 1978
Trajectory Module of the NASA Ames Research Center Aircraft Synthesis Program ACSYNT

Michael E. Tauber
John A. Paterson, Ames Research Center, Moffett Field, California
# TABLE OF CONTENTS

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

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.

FROM ACSYNT CONTROL

\[ \text{TRAJA} \]

ICALC = 1

\[ \text{TRAJIN} \]

\[ \text{TRAJOI} \]

ICALC = 2 OR 3

\[ \text{Misson} \]

ICALC = 3

\[ \text{CONVRG} \]

\[ \text{TRAJOO} \]

\[ \text{TRPLOT} \]

TAKEOFF

CLIMB

CRUISE

ACCEL

COMBAT

LOITER

DESCNT

PATHS

GOLDEN

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.

Misson - 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.

DESCENT - 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.

TRAJO - 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:
WFIXED = WGT0 - WFTOT - WPL

where WGT0 is the gross takeoff weight of the converged vehicle, WFTOT is the total fuel weight, and WPL 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) TMTTO1 - Time during takeoff at first power setting
(2) TMTTO2 - Time during takeoff at second power setting
(3) FRFURE - Fuel reserve, as fraction of total fuel
(4) DESLFL - Design load factor
(5) ULTLFL - Ultimate load factor
(6) RANGE - Aircraft range
(7) WFTUEL - Total internal fuel weight
(8) WFTEXT - 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) IPST01 - Power setting during first part of takeoff
(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** - \( \text{WLAND} = \text{WGTO} \times \text{WKLAND} \times \text{WFTOT} \)

(28) **FLFAC** - \( \text{FLFAC} = \text{FLLAND} = \frac{(XAIR + XGRLAN)}{FLFAC} \)

(29) **DECEL** - Deceleration factor

The following quantities are used for the plotting routine:

(30) **IPLT** - 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) **DELMMP** - Mach number interval to be used

(37) **WCOMBP** - Aircraft weight used

The following quantities are used for multimission computations:

(38) **NMIS** - 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 ≥ 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
The table below indicates which inputs may possibly be needed for each of the seven types of phases.

<table>
<thead>
<tr>
<th>PHASE</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLIMB</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CRUISE</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ACCEL</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>COMBAT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>LOITER</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>DESCENT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>PATHS</td>
<td>Mach vs h table</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</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 TAKEOFF (always called first for all missions), then CLIMB, then CRUISE, then DESCENT. 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:

\[
WFUSED = WFUEL \times (WGTO - W - f(W))
\]
\[
WFRES = WFUSED/(1. - FRFURE) - WFUSED
\]
\[
WFTOT = WFUSED + WFRES + WFTRAP
\]
\[
WFUEL = WFTOT - WSEX
\]

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, WFUEL 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 WSEX is the weight of fuel stored in external tanks.

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

\[
WLAND = WGTO - WKLAND*WFUSED
\]
\[
VSTALL = SQRT[2.*WLAND/(SWING*CL*RHO)]
\]
\[
VSCREEN = 1.3*VSTALL
\]
VTDOWN = 1.15*VSTALL
XAIR = [(VSCREN**2 - VTDOWN)/64.4 + 50.]**RLD
XRGLAN = VTDOWN**2/(64.4*DECEL)
FLLAND = (XAIR + XRGLAN)/FLFAC

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 (MENDUR). (The results of the approximate calculation for endurance altitude and time were only intended for use in the economics module of ACSYN/T 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 - WFTOT)].

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

WGCALC = 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. WGCALC 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 (IPST01) and for an input time (TINT01).

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

\[ FLTO = 2.10 \left( \frac{0.01163 \ W}{\rho C_L_S} + 0.374 \ \text{TOOBHT} \right) \left( \frac{1}{k_t \ \frac{F}{W} - 0.04} + 2.7 \right) + \frac{32.0}{\rho} \]

where FLTO is in ft, W/S in psf and \( \rho \) in slugs/ft\(^3\). The lift coefficient \( C_L \) is evaluated at 1.2 \( V_{\text{stall}} \) where

\[ V_{\text{stall}} = \sqrt{\frac{2 \ \text{WGTO}}{\rho C_{L_{\text{stall}}} S}} \]

\( k_t \) 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 \( V_{\text{stall}} \).
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 (TMT02) is input as a positive or zero value, the fuel used is

\[ WFT02 = 0.0167(TMT02)(SFC)(TN) \]

where the SFC and thrust (TN) are based on conditions at the end of takeoff. Alternatively, if TMT02 is set negative, the fuel used is based on the balanced field length and is

\[ WFT02 = \frac{(TN_{AVE})(SFC_{AVE})(FLTO)}{1800(1.2V_{stall})} \]

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

\[ WFT0 = WFT01 + WFT02 \]

**Climb**

The program can calculate climb in either of two modes. When VIND > 0, climb is calculated using a constant indicated airspeed. When VIND < 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

\[ NLEGS = \frac{\text{total interval}}{\text{average step size}} = \int_{H_1}^{H_2} \frac{dH}{f(H)} = \int_{H_1}^{H_2} \frac{dH}{4000 - 0.05H} \]

\[ = 20 \ln \left( \frac{4000 - 0.05H_1}{4000 - 0.05H_2} \right) \]
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 $3 \leq NLEGS \leq 20$. The above formula can be overridden by giving NLEGCL a positive integer value, in which case, $NLEGS = NLEGCL$. 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:

$$\Delta H = \min\left(\frac{1 + (\text{LEG})(e-1)/NLEGS}{1 + (\text{LEG} - 1)(e-1)/NLEGS}\right) (H_2 - H_1)$$

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 $\gamma$.

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

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

where

$$E_m = \frac{C_L}{2 \sqrt{C_D o C_D L}}$$

For the constant indicated airspeed climb,

$$\sin \gamma = \frac{\Delta h}{Vht}$$
where, now

\[ V = \frac{V_{\text{ind}}}{\sqrt{\rho / \rho_{\text{SL}}}} = \frac{V_{\text{ind}}}{\sqrt{\alpha}} \]

and where \( \alpha \) 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

\[ \Delta t = \frac{\sqrt{\alpha}}{V_{\text{ind}}} \Delta h - \frac{V_{\text{ind}}}{2g^{1.5}} \]

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

\[ X_{\text{CRUSE}} = X + DX_{\text{CRUS}} + X_{\text{DESC}} \]

where \( X_{\text{CRUSE}} \) is the actual horizontal distance that will be traversed in the cruise phase, \( X \) is the nominal horizontal distance input for that phase, \( DX_{\text{CRUS}} \) 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

\[ XCRUSE = \frac{RANGE}{NCRUSE} + DXCRUS + XDESC \]

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:

- \( X < 100 \), then NLEGS = 2
- \( 100 \leq X < 300 \), then NLEGS = 3
- \( 300 \leq X < 1000 \), then NLEGS = 4
- \( 1000 \leq X < 3000 \), then NLEGS = 5
- \( X \geq 3000 \), 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 \((\frac{V}{sfc})(\frac{L}{D})\) 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

\[ \sum W_F = \sum \frac{C_D W(sfc)}{\cos \alpha} \sqrt{\frac{D}{2 \gamma \frac{S}{W} \Delta X}} \]

where \( \Delta X \) 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 \((L/D)_{\text{max}}\); 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,

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

**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

\[ t = \frac{1}{g} \int \frac{WdV}{T \cos \alpha - D} \]

and the equivalent ground distance covered is given by

\[ X = \frac{1}{g} \int \frac{WdV}{T \cos \alpha - D} \]

**Combat**

In the combat phase, the acceleration and turning capabilities are calculated and the amount of fuel used by performing combat at a fixed altitude, \( \alpha \), 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)

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

where the airplane bank angle, \( \phi \), 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, $C_{l,\text{max}}$, 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

$$W_F = ET(sfc)Nt$$

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 MS1.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

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

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

$$\rho = \frac{2W}{C_LSV^2}$$
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

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

where \( R \) is the loiter circle radius. The exact solution is a tedious iteration between \( C_L \), \( C_D \), and \( \alpha \), 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 \((L/D)_{\text{max}}\); thus \( C_{Dl} = C_{Do} \) and for flight with thrust much less than drag, the optimum descent angle is approximately (ref. 6)

\[ \sin \gamma = -\frac{2C_{Do}}{C_L} \]

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

\[ \Sigma W_p = \Sigma T(sfc)At \]

and the equivalent ground distance covered is

\[ \Sigma AX = EV \cos \gamma At \]

OUTPUT-TRAJOC

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: Ig 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)
 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 lg
(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) $\text{HMAXP} = \text{maximum altitude}$

(4) $\text{DELHP} = \text{altitude interval}$

(5) $\text{SMMINP} = \text{minimum Mach number}$

(6) $\text{SMMAXP} = \text{maximum Mach number}$

(7) $\text{DELMXP} = \text{Mach number interval}$

(8) $\text{WCOMBP} \leq 0$, aircraft weight taken from first COMBAT phase

\[ 0 < \text{WCOMBP} \leq 1, \text{ aircraft weight equals } \text{WCOMBP times takeoff weight} \]

\[ \text{WCOMBP} > 1, \text{ aircraft weight equals } \text{WCOMBP} \]

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).
SUBROUTINE TRAJA

SUBROUTINE TRAJA (ICALC, NERROR, ICED, KGPRNT, IGPLT)
EXTERNAL MISSION
REAL MSTART, MEND, MENDUR
COMMON /TRAJCM/ ALPHA, AR, CD, CDL, CDQ, CL, DESL, DRAG, ENS, HN, KP,
  KLD, SMN, QMAX, RANGE, SFC, SWING, THRUST, TW, UTLF, W, WAMMUN, WEATANK,
  WFOE, WGTOK, WMISE, WMT, WPL, WSTART(12), HSTART(12), TIM(12),
  PS1GT(12), TOOTST(12), NZST(12), PSIT(12), TOOTIT(12), NZIT(12),
  WBOBMS, WFLXT, SFCU, THRST, COINSP, PRD, WFWO, CRMACH, FLT3, X(12),
  FLAND, TENDUR, BLRANG, BLTIME, TMT01, TMT02, WFTP, PNAMES(12),
  PNAMES(12), SMNT(12), HNT(12), WF1(12), TIMET(12), XT(12), CLT(12),
  COT(12), ALPHAT(12), CLIT(12), COIT(12), ALPHIT(12), WFWU, WLAND,
  WFT(12), RLT(12), SFLT(12), TNP(12), CET(12), DT(18),
  IAO, IPS, IIS, IBS, MPHASE, IDY(14)
COMMON /TRAJEX/ ALPHGT(12), CDINST(12), CDGT(12), CLGT(12), DECCL,
  DELHP, DELMP, DXCRUS, FLFAC, FRFURE, HEND(12), HENDUR, HMXP, HMIN,
  HPATHT(50), HEND(12), MENDUR, NT(12), PRDT(12), RADIT(12),
  RADIT(12), SF, SFCUT(12), SMMAPO, SMMIN,
  INUT(12), WINO(12), VELT(12), VPATHT(50), WART(12), WCOMB, WEELT(12),
  WFT, WFTOK, WFTO2, WFTAP, WKLAND, XDESC, XGLAN, WGMMA, WTLS,
  WAT, WEM, WECALC, WPL, WGTOK, WOUM, OMU(82), IA(12), IAS, IB(12), IBREG, IENDUR,
  IMISS, IP(12), IPHASE, IPRT(12), IPSIZE, IPSTO1, IPSTO2,
  IUT(12), IX(12), KERROR, MILCOM, NCRUSE, NLEGCL, NLEGLO, NMISS,
  WPATH, IDUMY(6)
DATA PATH/4HPATH/
  WPL=WPLMT
  WGTOK=WGTOKT
  IF (KGPRNT.EQ.0) GO TO 5
  KSTORE=KERROR
  KERROR=2
  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

C--
C ICALC = 1
C--
10  TIMTO1=5.0
    TIMT02=1.0
40  FKFURE=.05
    DESL=.5
    UTLF=.75
    RANGE=1000.
    WFWO=1000.
    WFT=0.0
    WFTAO=100.
    CMAX=700.
    XDESC=80.
    WFTO=1.0
    CMECH=.8
    WLAND=.57
    FLFAC=.6
    DECEL=.25
    NCRUSE=.2
    IPSIZE=0
35  IPSTO1=5
    IPSTO2=2
SUBROUTINE TRAJA  76/76  OPT=2  FTN 4.5+410  C

IBREG=0
IENDUR=0
IPRINT=0
KERROR=2
IPLLOT=0
HMINE=0.0
MAXP=40000.
65  DELHP=4000.
SMINP=3
SMAXP=9
DELHP=1
WCOMBP=7
IBREG=0
IENDUR=0
6c  IPRINT=0
KERROR=2
IPLOT=0

DE LHP=40: X1.
SMMINP=.3
SMMAXP=.9
DELMP=.1
WCGMBP=.7
70  MILCOM=0
MENDUR=.5
NELGCL=0
NELGCK=0
NELGLG=0
75  NMISS=1
FWGHAX=1.2
TOL=.01
CALL TRAJIN
CALL TRAJOI

C-----READ MISSION 1 FROM TAPE 22.
12  IMISS=1
REIND 22
READ (22,2200) NPHASE, WPLTRA
2200  FORMAT(110, E15.8)
85  DG 15 IPHASE=1, NPHASE
READ (22,2201) PNAME(IPHASE), PNAME2(IPHASE), HSTART(IPHASE),
1  MENU(IPHASE), HSTART(IPHASE), HEND(IPHASE), X(IPHASE), TIM(IPHASE),
2  NT(IPHASE), VIND(IPHASE), IP(IPHASE), IX(IPHASE), IA(IPHASE),
3  IRT(IPHASE), IB(IPHASE), IA(IPHASE)
9C  2201  FORMAT(2A4, 2X, 7E15.8, 6E10)
IF (PNAME1(IPHASE).NE.PATH) GO TO 15
READ (22,2200) NPATH
READ (22,2202) (HPATH1(I), I=1, NPATH)
READ (22,2202) (VPATH1(I), I=1, NPATH)
95  2202  FORMAT(8F15.8)
15  CONTINUE
C-----END OF READING MISSION 1 FROM TAPE 22.
IF (ICALC.EQ.3) RETURN
WFIXED=0.0
10C  SF=ULTF/OESLF
ALPHA=0.0
CL=0.0
DRAG=0.0
HN=0.0
105  SN=0.0
w=0.0
IF (KGPRNT.NE.0) KERROR=KSTORE
RETURN
C-----------
11C  ICALC = 2
C-----------
20  CALL MISSON(ICALC, KERROR, IGE0, KGPRNT)
IF (KGPRNT.NE.0) KERROR=KSTORE
RETURN
SUBROUTINE TRAJA

C--------
C ICALC = 3
C--------
30 IF (NMISS.EQ.1) GOTO 40
KCALC=2
C------READ NEW MISSION FROM TAPE 22.
130 READ (22,2201) NPHASE, WPLTRA
DO 60 IPHASE=IsNPHASE,
READ (22,2201) PNAME1(IPHASE), PNAME2(IPHASE), MSTART(IPHASE),
1 MEND(IPHASE), X(IPHASE), TIM(IPHASE),
2 NTIIPHASE), VNDIPHASE), IP(IPHASE), IX(IPHASE), IW(IPHASE),
3 IB1IPHASE), IAIIPHASE)
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)
60 CONTINUE
C------END OF READING NEW MISSION FROM TAPE 22.
WPL=WPLTRA
WGEST=WGEST
IF (IPRINT.EQ.0) GOTO 65
WRITE (6x,600) IMISS, WGEST, WGAME, WFIXED, WFTOT, WGTO, WGCONV, WPL,
1 WGEST1, WGCALC
600 FORMAT(1H3,H18H CHECK FOR MISSION PRIOR TO CALLING CONVGR/)
1 9H WGEST=*E13.6,4X,7H WGAME=*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,7HWGCONV=*E13.6,7H WGCALC=*E13.6/)
65 CALL CONVGR(MISSON, WGESTl, WGCALC, WGAME, TOL, IPRINT)
IF (IPRINT.EQ.0) GOTO 40
WRITE (6x,601) WGEST, WGAME, WFIXED, WFTOT, WGTO, WGCONV, WPL, WGEST1,
1 WGCALC
601 FORMAT(1H3,H18H AFTER CALLING CONVGR/
1 9H WGEST=*E13.6,4X,7H WGAME=*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,7HWGCONV=*E13.6,7H WGCALC=*E13.6/)
SUBROUTINE TRAJIN
REAL MSTART, MEND, NT, MENDUR
COMMON /TRAJCM/ ALPHA1AR, CD, CDG, CL, DESLF, DRAG, EN, HN, KP
1 RLD, SMN, QMAX, RANG, SFC, SWING, THRUST, TW, UULF, W, WAMNUN, WETANK,
2 WFUEL, WGTOUT, WMISS, WDTF, WPLUT, MSTART(12), MSTART(12), NTIM(12),
3 PS1GT(12), PS1TO(12), NT(12), PS1T(12), PS1TO(12), NTIM(12),
4 WBOMBS, WFE, SFCU, THR, TUP, CDINS, PRTO, WKFUEL, CRM, HFG, PTO(12),
5 FLLAND, TFENDUR, LIRANG, UBTIM, TIMTO1, TIMTO2, WFTD, PNAME1(12),
6 PNAME2(12), SMNT(12), NT(12), WFT(12), TIMET(12), XT(12), CLT(12),
7 CDT(12), ALPHAT1(12), CLIT(12), CDIT(12), ALPHIT(12), WFUSED, WLAND,
8 QT(12), RLOT(12), SFCIT(12), NT(12), CET(12), DMT(18),
9 PID0, I5S, IBS, NPHASE, IDY(14),
10 CMMON /TRAJEX/, ALPIGT(12), SDINST(12), SDLGT(12), CLGT(12), DECEL,
15 DT, DLNP, DLNP, DCRUS, FLFAC, FFRE, HEND(12), HENDUR, HMA, HMIN, NP,
2 HPATHT(10), MEND(12), MENDUR, NT(12), PRTO(12), PRDI(12), RADDI(12),
3 RADT(12), SF, SFCUT(12), MMAX, SHMIN, NP,
4 TW(12), VINDT(12), VELT(12), WPATH(50), WFTD, WCP, WEE, T(12),
5 WFTD, WPATH(50), WFTD, WTRAP, WKLAND, XDESC, XGRLAN, WFGMAX, TTL,
6 WFIXED, WGCALC, WPLT, WGTOUT, UMTY(18), IA(12), IAS, IB(12), IC, E, ENC, NENDUR,
20 IMISS, IP(12), IPHASE, IPRINT, IPRT(12), IPSIZE, IPSTO1, IPSTO2,
6 W1(I12), IX(I12), KERROR, HICOM, NCURSE, NEC, NFT, NLEGC, NLEGD, NMISS,
9 NPATH, IMUTY(14),
DATA PATH/4PATH/, NPATH/4PATH/, NTIMTO1/TIMTO1, TIMTO2/TIMTO2,
25 FFRE/DESFL, UTLF, RANG, WFUEL, WPFXE, WTRAP, QMAX, XDESC, WKFUEL, CRM, HMAX, HMIN, NP,
1 WPTO, WMISS, WDTF, WPLUT, MSTART(12), MENDUR, NTMOM, CS, PFORMAT(S5),
6 IF (PNAME1(IPHASE).NE.PATH) CALL READ(5)
30 IF (NPATH) READ(5, S501)
DO 10 IPHASE=1, NSPHASE
READ(5, S501) NPHASE, WPLTRA
WRITE(22, S2000) NPHASE, WPLTRA
500 FORMAT(S10, S15.6)
35 FORMAT(S10, S15.8)
2200 FORMAT(S10, S15.8/S15.6/S11O)
C READ IN THE FOLLOWING WORDS FOR EACH PHASE
C PHASE(2 PART), STARTING MACH, ENDING MACH, STARTING ALTITUDE,
C ENDING ALTITUDE, HORIZONTAL DISTANCE, TIME, NUMBER OF TURNS,
C CCSTANT INDICATED AIRSPEED CLIMB, POWER SETTING, RANGE INCREMENT
40 C INDICATOR, WEAPONS RELEASE INDICATOR, GOMBS RELEASE INDICATOR,
C AMMO RELEASE INDICATOR.
DO 10 IPHASE=1, NPHASE
READ (5, S501) PNAME1(IPHASE), PNAME2(IPHASE), MSTART(IPHASE),
1 MEND(IPHASE), NTSTART(IPHASE), NTEND(IPHASE), TIMET(IPHASE),
2 NT(IPHASE), VIND(IPHASE), IIP(IPHASE), IX(IPHASE), IW(IPHASE),
3 IPRT(IPHASE), IB(IPHASE), IA(IPHASE)
501 FORMAT(S24, 2X, 7E10.0/6E10.0/6E10.0)
WRITE (22, S24G1) NAME1(IPHASE), NAME2(IPHASE), MSTART(IPHASE),
1 MEND(IPHASE), NTSTART(IPHASE), NTEND(IPHASE), TIMET(IPHASE),
2 NT(IPHASE), VIND(IPHASE), IIP(IPHASE), IX(IPHASE), IW(IPHASE),
3 IPRT(IPHASE), IB(IPHASE), IA(IPHASE)
22G1 FORMAT(S24, 2X, 7E10.0/6E10.0)
IF (PNAME1(IPHASE).NE.PATH) GO TO 10
READ (5, S501) NPATH
5F READ (5, S502) (HPATH(TI), I=1, NPATH)
README (5, S502) (VPATH(TI), I=1, NPATH)
5G2 FORMAT(S8E10.6)
SUBROUTINE TRAJIN  76/76  OPT=2

WRITE (22,2200) NPATH
WRITE (22,2202) (HPATH(I),I=1,NPATH)
WRITE (22,2202) (VPATH(I),I=1,NPATH)

6C 2202 FORMAT(8E15.8)
10 CONTINUE
RETURN
END

ORIGINAL PAGE IS OF POOR QUALITY
SUBROUTINE TRAJ

C PRINTOUT OF TRAJECTORY INPUT

REAL MSTART,MEND,NT,MENDUR

COMMON /TRAJCM/ ALPHA1,AR,W,C,CDL,CDO,CL,DESFL,DRAG,EN,HN,KP,
R,LD,SM,GMAX,RANGE,SP,SWING,THUST,TU,ULTF,W,WMHUM,WTANK,
WFUEL,WGTOWN,WMISS,WHT,WPLAT,WSTART(12),HSTART(12),TIM(12),
PS1T(12),TDOUT1(12),NZST(12),PS1T(12),TDOUT1(12),NZIT(12),
WBOBD,WFEXT,SPC,THRE,GRID,PRTOT,WKFC,MACH,FLT0,XT(12),
FLAND,ENDUR,BLANGE,BLITME,TIMTO1,TIMTO2,WFD1,PNAME1(12),
PNAME2(12),SMNT(12),HT(12),WFT(12),TIMET(12),XT(12),CLT(12),
C1T(12),ALPHAT(12),CLIT(12),C1T(12),ALPHIT(12),WGUW,WTLAND,
V(12),RLD1(T12),SFCT(12),TNT(12),CET(12),DY(18),
IA0,IPS,ITS,IB0,IBS,NPHASE,ITYD(14)

COMMON /TRAJEX/ AP1T(12),CDINST(12),CD1T(12),CL1T(12),DECEL,
RADST(12),SF,SFCUT(12),SMAXP,SMINP,
TNT1(12),VIND(12),VELT(12),VPATH(50),WART(12),WCOMBP,WEELT(12),
WRES,WFTOT,WFT0,WFTRAP,WLAND,XDESC,XGRLAN,WGMAKXP,TOL,
HFEXD,NGCALC,WPLY,WGTO,DUMY(82),IA(12),IAS(12),IBREG,EINDUR
IMISS,IP1,IP2,IP3,IP4,IP5,IP6,IP7,IP8,IP9,IP10,IP11,IP12,
KERROR,MILCOM,MCRUSE,NLEGCL,NLEGCR,NLGEDC,NLGEDN,NMISS,
NPATH,NDUMY(61)

DATA PATH/4HPATH/

WRITE (6,600) TIMTO1,MENDUR,NCRUSE,IPLOT,TIMTO2,GMAX,IPSIZE,HMINP
FRFURE,XDESC,IPSTO1,HMAXP,DELMP,ULTF,
SFC,THRE,GRID,PRTOT,WKFC,MACH,FLT0,XT(12),
IPRINT,DELMP,WFEXT,DECEL,KERROR,WCOMBP,WFRAG,MILCOM,MCRUSE,
NLEGCL,NLEGCR,NLGEDC,NLGEDN,NMISS

600 FORMAT(1H1,25X,16HTRAJECTORY INPUT///

1 1X,BHT1M岗 *,F6.1, 7X,OHMENDUR *,F6.0, 4X,OBNRUSE *,I2,
2 3X,BIHPSIZE *,I2, 3X,OH HMINP *,F7.0, 1X,OHBFRE *,F5.2,
3 6X,BH XDESC *,F6.1, 4X,BHIPST01 *,I2, 3X,OH HAXP *,F7.0/
4 5X,BH DEFL *,F5.2, 6X,BHWFUEL *,F6.3, 4X,BHIPST02 *,I2,
5 3X,BH DELP *,H7.0/ 1X,BH ULLF *,F5.2, 6X,BHCRMACH *,F6.3,
6 4X,BH IBREG *,I2, 3X,BHSMINP *,F6.3, 1X,BH RANGE *,F7.0/
7 3X,BH WFL *,F6.3, 4X,BHIENDUR *,I2, 3X,BHSMAXP *,F6.3/
1X,BH WFL *,F6.0, 3X,BH FLFAC *,F6.3, 4X,BHPRINT *,I2,
2 3X,BH IBREF *,I2, 3X,BHGCMB *,F9.2/ 1X,BHTRAP *,F8.0/
3 3X,BHNLGCL *,I2, 8X,BHNLGCR *,I2, 3X,BHNLGCL *,I2/
4 3X,BHWFMAX *,F6.3, 5X,BH TOL *,F6.3, 4X,BHMLCOM *,I2,
5 3X,BH NMISS *,I2)

45 REWIND 22

DD 10 IMISS=1,NMISS
READ (22,2200) NPHASE,WPLTRA

2200 FORMAT(10E15.8)

IF (IMISS.EQ.1) WRITE (6,601) IMISS

601 FORMAT/// 24X,7MISSION=I2//

1 9X,66HMACH MACH ALT ALT HORIZ NO. IP
2 1X,7575 PHASE START END START END DIST TIME TURNS
3 VINE IPRT 1B IA/

IF (IMISS.GT.1) WRITE (6,602) IMISS,WPLTRA

602 FORMAT/// 24X,7MISSION=I2,12H (PAYLOAD =F8.0,4H LB)///

1 9X,66HMACH MACH ALT ALT HORIZ NO. IP
2 1X,7575 PHASE START END START END DIST TIME TURNS

26
SUBROUTINE TRAJOI
76/76 UPT=2 FTN 4.5.410

3 VIND IPRT 18 IA/
DO 1C IPHASE=1,NPHASE
   READ (22,2201) PNAME1(IPHASE),PNAME2(IPHASE),MSTART(IPHASE),
   HEND(IPHASE),X(IPHASE),TIM(IPHASE),
2 NT(IPHASE),VIND(IPHASE),IP(IPHASE),IX(IPHASE),IW(IPHASE),
   IPRT(IPHASE),IB(IPHASE),IA(IPHASE)
2201 FORMAT(2A4,2X,7E15.8/E15.8,6I10)
   WRITE (6,603) PNAME1(IPHASE),PNAME2(IPHASE),MSTART(IPHASE),
   HEND(IPHASE),X(IPHASE),TIM(IPHASE),
   NT(IPHASE),VIND(IPHASE),IP(IPHASE),IX(IPHASE),IW(IPHASE),
   IPRT(IPHASE),IB(IPHASE),IA(IPHASE)
   1 I65,I5,I4)
   IF (PNAME1(IPHASE).NE.PATH) GO TO 10
   READ (22,2202) NPATH
   READ (22,2202) (HPATHT(I),I=1,NPATH)
   READ (22,2202) (VPATHT(I),I=1,NPATH)
2202 FORMAT(8E15.8)
   WRITE (6,604) NPATH
   WRITE (6,605) (HPATHT(I),I=1,NPATH)
   WRITE (6,606) (VPATHT(I),I=1,NPATH)
8C 604 FORMAT(9X,8H NPATH =,I3)
   6C5 FORMAT(9X,8HPATHT =,7F8.0/(17X,7F8.0))
   6C6 FORMAT(9X,8HPATHT =,7F8.0/(17X,7F8.0))
   6C7 FORMAT(/)
10 CONTINUE
RETURN
END
SUBROUTINE MISSON

IMPLICIT REAL*8 (A-H, I-O)
COMMON /TRAJCH/ ALPHAPARIIjPCDjCDLPCL)OPCLsDESLFPDRArPENPHNjPKPs
KLDPSMN,QMAXsRANGEsSFCPSWIHGYTHRUS
COMMON /TRAJEX/ ALPIGT(12)PCDINST(12)PC01GT(12)sCL1GT(12)PDECELs
1 DELHPsDELMPsDXCRUSsFLFACsFRFUREsHEND(12)PHENDURPHMAXPsHMINPP
2 HPATHT(50)sMEND(12)PMENDURPNT(12)sRADIT(12)s
3 RADST1121sSFsSFCUT(12)DSMMAXPsSMMINPs
4 TNUT(12)PVIND(12)DWELT(12)PVPATH(50)PkART(12)PWCOMBPPWEELT(12)s
5 WFRESsWFTOTPWFTOIPWFTO2PWFTRAPPWKLAND.PXDESCRXGRLANsFWGMAXsTOLs
6 b kFIXEDsWGCALC
7 IMISSPIP(12)PIPHASEPIPLOTPIPRINTPIRT(12)sIPSIZEIPSTOIPIPST02s
8 IW(12)

DATA CLIMI4HCLIM/
DATA CRU114HCRUI/
DATA ACCE14HACCEI
DATA COMB14HCOMB/
DATA LOIT/4HLOIT/
DATA DESC/4HDESC/
DATA PATH/4HPATH/

C SET ULTLF SO AS TO MAINTAIN RATIO BETWEEN ULTLF AND DESLF
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
C SET THEM TO ZERO IF THE AMMO INDICATOR IS CHANGED
C TO ZERO BY SOME TRAJECTORY PHASE, THIS WILL AFFECT
C THE CALCULATION OF WFL^SED*

IWS=1
IBS=1
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
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
BLRANG=0.0
CALL TAKEOF(ICALC,MERROR,IGEO,KGPRNT)
IF (MERROR*GE.2) GO TO 160
IF (WLT*WPL) GO TO 130
C PERFORM ALL PHASES OF MISSION IN THE ORDER THEY WERE READ IN.
DO 120 IPHASE=1,NPHASE
IF (PNAMEI(IPHASE)*EQ.CLIM) GO TO 40
IF (PNAMEI(IPHASE)*EQ.CRUI) GO TO 50
120 CONTINUE

120 CONTINUE
SUBROUTINE MISSON

IF (PNAME1(IPHASE) .EQ. ACCE) GO TO 60
IF (PNAME1(IPHASE) .EQ. COMB) GO TO 70
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 (/97H TRAJECTORY_INPUT USES UNDEFINED NAME FOR PHASE, I3/) GO TO 160
40 CALL CLIMB(ICALC,NERROR,IGEO,KGPRNT) GO TO 110
50 CALL CRUISE(ICALC,NERROR,IGEO,KGPRNT) GO TO 110
70 CALL ACCEL(ICALC,NERROR,IGEO,KGPRNT) GO TO 110
60 CALL COMBAT(ICALC,NERROR,IGEO,KGPRNT) GO TO 110
75 CALL LOITER(ICALC,NERROR,IGEO,KGPRNT) GO TO 110
90 CALL DESERT(ICALC,NERROR,IGEO,KGPRNT) GO TO 110
100 CALL PATHS(ICALC,NERROR,IGEO,KGPRNT) GO TO 110
110 IF (NERROR.GE.2) GO TO 160
120 CALL TRAJ01(2) IF (NERROR.GE.2) GO TO 160

C-----COMPUTE USED, RESERVE, TOTAL, AND INTERNAL FUEL WEIGHT.
130 WFUSED=WGTO-W
85 IF (IWS.EQ.0) WFUSED=WFUSED-WMISS
IF (IBS.EQ.0) WFUSED=WFUSED-WBOMBS
IF (IAS.EQ.0) WFUSED=WFUSED-WAMMUN
IF (ITS.EQ.C) WFUSED=WFUSED-WETANK
WFUSED=WFUEL*WFUSED
WFRES=WFUSED/(1.-PRFURE)-WFUSED
WFTLT=WFUSED+WFRES+WFTRAP
WFUEL=WFTOT-WFEXT
IF (WG.EQ.WPL) GO TO 140
IF (KERRU.EQ.2) WRITE (6,602)

602 FORMAT (/37H FUEL WEIGHT EXCEEDS AVAILABLE WEIGHT/) GO TO 160

C-----CALCULATE LANDING FIELD LENGTH.
140 RHOD=.00238
14C IF (RHOD.GE.12) CALL TRAJ01(2)
10C CALL AERO(ICALC,NERROR,IGEO,KGPRNT)
105 CALL TRAJ01(1)
110 IF (NERROR.GE.2) GO TO 160

C-----CALCULATE ENDURANCE ALTITUDE AND TIME.
29
SUBROUTINE MISSGN  76/76  OPT*2  FTN 4.5410  C

115  HENDUR=XX
120  TENDUR=Y*ALOG(WGTO/(WGTO-WFUSED))

C-----COMPUTE GROSS WEIGHT.
WGCALC=FIXED+WPL+WFPTOT
IF (IPRINT.EQ.0) GO TO 160

WRITE (6,603) WGTO,WFUSED,WFRES,WFTOT,WFUEL,WPL,WLAND,
1  WLAND,XGRLAN,XFLAND,HENDUR,TENDUR,WGCALC


160  RETURN
END
SUBROUTINE TAKEOFF

COMMON /TRAJCH/, ALPH, ARN, CD, COL, CD0, CL, DESLF, DRAG, EN, HN, KP,
1 RLD, SHN, QMAX, RANGE, SFC, SWING, THRUST, TML, UTLF, W, HAMMUN, WETANK,
2 WFUEL, WFT0DL, WMIN, WOT, WPLT, WSTART, WSTOP, WTOP, X(12),
3 PS16T(12), TSTART(12), ZSTART(12), PSIT(12), TDOSTT(12), NZIT(12),
4 PBOMBS, WEXT, SFCU, THRUSTU, C1NSP, PRG, WFUEL, CRMACH, FLT0, X(12),
5 FLAND, TENDUR, BRLANG, BTIME, TIMTO, TIMT2, WFT0L, WNAME1(12),
6 WNAME2(12), SMN(12), HNT(12), WFT(12), TIMET(12), X1(12), CLT(12),
7 CDT(12), ALPHAT(12), CLIT(12), CDIT(12), ALEPH(12), WFUSED, WLAND,
8 OT(12), RLDT(12), SFC(12), TN(12), CET(12), DY(18),
9 IA, IPS, ITS, IWS, IB, NPHASE, IDY(14)
COMMON /TRAJEX/, ALPIG(12), D1NST(12), C1G(12), CLG(12), DECEL,
1 DELH, DELM, DCRUS, FLFAC, PPFUTE, HEND(12), HENDUR, HMAXP, HMINP,
2 HPATH(50), HEND(12), HENDUR, NT(12), PRTOTT(12), RADT(12),
3 RADST(12), SF, SFCUT(12), SHMAXP, SMINP,
4 TNt(12), VIND(12), VELT(12), VPATHT(50), WARD(12), WCOMBP, WELT(12),
5 WFE, WPWT0FL, WPWT0L, WPWTRAP, WKLAND, XDESC, XGRLAN, WFGMAX, TOL,
6 IFIXE, WSGAF, WPT0L, WFT0DL, DUMY(12), IA(12), I, IS, IB(12), IBREG, HENDUR,
7 IMISS, IP(12), IPHASE, IPLOT, IPRT(12), IPSIZE, IPST01, IPST02,
8 IN(12), X1(12), XERROR, MILCOM, NCRUSE, NLEGCL, NLEGCR, NLEGDL, NMISS,
9 NPATH, IDUMY(6)

C----CALL PROP TO SIZE ENGINE.
  IPS=IPSIZE
  HN=0,
  SMN=0.
  CALL TRAJ0J(2)
  CALL PROP(ICALC, NERROR, IGEQ, KGPRNT)
  IF (NERROR.GE.2) RETURN
  CALL TRAJ0J(1)

C----CALCULATE STARTUP AND TAXI FUEL USED.
  IPS=IPST01
  CALL TRAJ0J(2)
  CALL PROP(ICALC, NERROR, IGEQ, KGPRNT)
  IF (NERROR.GE.2) RETURN ...
  CALL TRAJ0J(1)
  TN=THUST*EN
  WFT0L=0.167*TIMTO1*SFC*TN

C----BEGIN FIELD LENGTH CALCULATION.
  HNTO=HSTART(1)
  HN=HNTO
  CALL AT62(HN, ANS)
  RHO=ANS(1)
  A=ANS(4)

C----GUESS STALLING MACH NUMBER.
  SMN=.25
  IAD=14
  CALL TRAJ0J(2)
  CALL AKO(IKCALC, NERROR, IGEQ, KGPRNT)
  IF (NERROR.GE.2) RETURN
  CALL TRAJ0J(1)
  CLS=CL

C----CALCULATE STALLING SPEED.
  WS=SQR(2.*WGT0L/(RHO*CLS*SWING))
  V2=1.2*WS
  SM2=V2/A
  SMN=SM2

C----CALL PROP TO SIZE ENGINE.
  IPS=IPSIZE
  HN=0,
  SMN=0.
  CALL TRAJ0J(2)
  CALL PROP(ICALC, NERROR, IGEQ, KGPRNT)
  IF (NERROR.GE.2) RETURN
  CALL TRAJ0J(1)
  TN=THUST*EN
  WFT0L=0.167*TIMTO1*SFC*TN

C----BEGIN FIELD LENGTH CALCULATION.
  HNTO=HSTART(1)
  HN=HNTO
  CALL AT62(HN, ANS)
  RHO=ANS(1)
  A=ANS(4)

C----GUESS STALLING MACH NUMBER.
  SMN=.25
  IAD=14
  CALL TRAJ0J(2)
  CALL AKO(IKCALC, NERROR, IGEQ, KGPRNT)
  IF (NERROR.GE.2) RETURN
  CALL TRAJ0J(1)
  CLS=CL

C----CALCULATE STALLING SPEED.
  WS=SQR(2.*WGT0L/(RHO*CLS*SWING))
  V2=1.2*WS
  SM2=V2/A
  SMN=SM2

C----CALL PROP TO SIZE ENGINE.
  IPS=IPSIZE
  HN=0,
  SMN=0.
  CALL TRAJ0J(2)
  CALL PROP(ICALC, NERROR, IGEQ, KGPRNT)
  IF (NERROR.GE.2) RETURN
  CALL TRAJ0J(1)
  TN=THUST*EN
  WFT0L=0.167*TIMTO1*SFC*TN

C----BEGIN FIELD LENGTH CALCULATION.
  HNTO=HSTART(1)
  HN=HNTO
  CALL AT62(HN, ANS)
  RHO=ANS(1)
  A=ANS(4)

C----GUESS STALLING MACH NUMBER.
  SMN=.25
  IAD=14
  CALL TRAJ0J(2)
  CALL AKO(IKCALC, NERROR, IGEQ, KGPRNT)
  IF (NERROR.GE.2) RETURN
  CALL TRAJ0J(1)
  CLS=CL

C----CALCULATE STALLING SPEED.
  WS=SQR(2.*WGT0L/(RHO*CLS*SWING))
  V2=1.2*WS
  SM2=V2/A
  SMN=SM2

C----CALL PROP TO SIZE ENGINE.
  IPS=IPSIZE
  HN=0,
  SMN=0.
  CALL TRAJ0J(2)
  CALL PROP(ICALC, NERROR, IGEQ, KGPRNT)
  IF (NERROR.GE.2) RETURN
  CALL TRAJ0J(1)
  TN=THUST*EN
  WFT0L=0.167*TIMTO1*SFC*TN

C----BEGIN FIELD LENGTH CALCULATION.
  HNTO=HSTART(1)
  HN=HNTO
  CALL AT62(HN, ANS)
  RHO=ANS(1)
  A=ANS(4)
CALL TRAJO1(2)
CALL AERO(ICL, NERROR, IGE0, KGPRNT)
IF (NERROR.GE.2) RETURN
CALL TRAJO1(1)
CL2=CL

C-----FIND THRUST AT END OF TAKEOFF.

IF (NERROR.GE.2) RETURN
CALL TRAJO1(2)
CALL PROP(ICL, NERROR, IGE0, KGPRNT)
IF (NERROR.GE.2) RETURN
CALL TRAJO1(1)
TN=THRUST*EN
TN2=TN
SFC2=SFC

C-----FIND STATIC THRUST.

SMN=0.
CALL TRAJO1(2)
CALL PROP(ICL, NERROR, IGE0, KGPRNT)
IF (NERROR.GE.2) RETURN
CALL TRAJO1(1)
TN=THRUST*EN
TN0=TN
SFC0=SFC

C-----CALCULATE AVERAGE THRUST AND SFC DURING TAKEOFF.

TNAVE=.5*(TN0+TN2)
SFCAVE=.5*(SFC0+SFC2)

C-----CALCULATE TAKEOFF FIELD LENGTH REQUIRED.

IF (MILCOM.EQ.0) TOOBHT=50.0
IF (MILCOM.NE.0) TOOBHT=35.0
FLTO=2.1*9.374*TOOBHT/0.0163*WGT0/(RHO*CL2*SWING)+
     1 (2.7+1.0/(TNAVE/WGT0-0.4)+32.0/5RHO)

C-----CALCULATE TAKEOFF FUEL USED.

IF (TIMTO2.LT.0.0) GO TO 10
IF (TIMTO2.EQ.0.0) GO TO 20
HN=0.
IPS=IPST02
CALL TRAJO1(2)
CALL PROP(ICL, NERROR, IGE0, KGPRNT)
IF (NERROR.GE.2) RETURN
CALL TRAJO1(1)
TN=THRUST*EN
WFT02=.0167*TIMTO2*SFC*TN

10 WFT02=TNAVE*SFC*FLTO/(1800.*V2)
GO TO 30

20 WFT02=0.0

C-----CALCULATE TOTAL FUEL USED FOR STARTUP, TAXI, AND TAKEOFF.

30 WFTO=WFT01+WFT02
W=WGT0-WFTO
IF (IPRINT.EQ.0) RETURN
WRITE (6,600) WGT0, WFT01, WFT02, WFTO, W, HNTO, CLS, VS, V2, SMN2, CL2,
     1 TN2, SFC2, TNO, SFC0, TNAVE, SFCAVE, FLTO

600 FORMAT(1H1, 6X, 7HTAKEOFF/ /3X, 6HWGT0 =E14.7, 2X, 7HWFTO1 =E14.7,
     1 2X, 7HWFTO2 =E14.7, 3X, 6HWFTO =E14.7, 6X, 3HW =E14.7/,
     2 3X, 6HWHTO =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/,
     5 2X, 7HTNAVE =E14.7, 1X, 9HSFCAVE =E14.7, 3X, 6HFLTO =E14.7/)
RETURN
END
SUBROUTINE CLIMB ICALC, NERROR, IGE0, KGPRNT
REAL MSTART,MEND
COMMON /TRAJCH/ ALPHA, ALPHAT, CD, CDL, CD0, CLF, DESL, DRAG, EN, HN, KP,
1 RLD, QMAX, RANGE, SFC, SWING, THRUST, TUE, ULTLF, WAMMUN, WETANK,
2 WFUEL, WGTOW, WFTO, WFTO2, WFTO3, WFRAP, WKLAND, XDESC, XGLAN,
3 WLAND, WMAX, WMAX2, WMAX3,
4 WFLUX, XDRAG, XDF, XDR, XDR2, XD, XE, XT, YT,
5 ZT, ZT2, ZT3, ZT4, ZT5, ZT6, ZT7, ZT8, ZT9, ZT10, ZT11, ZT12,
6 ZT13, ZT14, ZT15, ZT16, ZT17, ZT18, ZT19, ZT20, ZT21, ZT22,
7 ZT23, ZT24, ZT25, ZT26, ZT27, ZT28, ZT29, ZT30, ZT31, ZT32,
8 ZT33, ZT34, ZT35, ZT36, ZT37, ZT38, ZT39, ZT40, ZT41, ZT42,
9 ZT43, ZT44, ZT45, ZT46, ZT47, ZT48, ZT49, ZT50, ZT51, ZT52,
10 ZT53, ZT54, ZT55, ZT56, ZT57, ZT58, ZT59, ZT60, ZT61, ZT62,
11 ZT63, ZT64, ZT65, ZT66, ZT67, ZT68, ZT69, ZT70, ZT71, ZT72,
12 ZT73, ZT74, ZT75, ZT76, ZT77, ZT78, ZT79, ZT80, ZT81, ZT82,
13 ZT83, ZT84, ZT85, ZT86, ZT87, ZT88, ZT89, ZT90, ZT91, ZT92,
14 ZT93, ZT94, ZT95, ZT96, ZT97, ZT98, ZT99, ZT100, ZT101, ZT102,
15 ZT103, ZT104, ZT105, ZT106, ZT107, ZT108, ZT109, ZT110, ZT111, ZT112,
16 HNFIN=HEND
IF ( (IFLAG.EQ.0) .OR. (HEND(IFLIPE) .GT. 0)) GO TO 40
IF (IFLAG .GT. 2) RETURN
HNFIN=HH
40 SMN=MSTART(IFPAS)
IF (MSTART(IFPAS) .LT. 0) SMN=SMNT(IFPAS-1)
IF (HNFIN .LT. 80000.) GO TO 43
NLEGS=20
GO TO 45
43 NLEGS=20+ALOG((400.0-0.05*HNFIN)/(400.0-0.05*HNFIN))+.5
IF (NLEGS .LT. 9) NLEGS=3
IF (NLEGS .GT. 20) NLEGS=20
45 NLEGS=9
NLEGS1=NLEGS+1
B1=HNFIN-HNFIN
B2=1.718282*FLOAT(NLEGS)
IF (B2 .LT. 0) GO TO 50
SMNT(IFPAS)=SMN
IF (SHNT(IFPAS) .GT. HNFIN) HNFIN=SHNT(IFPAS)
46 B1=B1+1
50 IF (B1 .GT. 10) GO TO 50
MH=(HNFIN-BL)/FLOAT(NLEGS)
IF (MH .LT. 0) GO TO 50
51 NLEGS=9
NLEGS1=NLEGS+1
B1=HNFIN-HNFIN
B2=1.718282*FLOAT(NLEGS)
IF (B2 .LT. 0) GO TO 50
SMNT(IFPAS)=SMN
IF (SHNT(IFPAS) .GT. HNFIN) HNFIN=SHNT(IFPAS)
52 C-----END OF SUBROUTINE CLIMB
SUBROUTINE CLIMB

RLDT(IPHASE)=0.0
WEELT(IPHASE)=W
WART(IPHASE)=0.0
RETURN

50  HN=HNINT
TIME=0.0
WF=0.0
XCLIMB=0.0
VINDFS=0.0
IPS=IP(IPHASE)
IAO=8

60  IF (VIND(IPHASE).LE.0.0) GO TO 60

C-----INITIALIZE CONDITIONS FOR CONSTANT INDICATED AIRSPEED CLIMB.
601  IF (IPRT(IPHASE).NE.0) WRITE (6,601)

70  CALL AT62(HN,ANS)
RHOP=ANS(1)
CL=2.*W/((.002378*VINDFS**2*SWMG)
CALL TRAJ01(2)
CALL AERD(ICALC,NERROR,IGEO,KPRNT)

80  CALL TRAJ01(1)
CALL TRAJ01(2)
CALL PROPC(ICALC,NERROR,IGEO,KPRNT)
IF (NERROR.GE.2) RETURN

85  CALL TRAJ01(1)
TN=THRUST*Eh
THU=THRUST*EN

90  IF (IPRT(IPHASE).EQ.0) GO TO 70
WRITE (6,602) W,HNINT,HFINN,SMN,HN

95  DO 140 LEG=1,NLEGSI

100  IF (LEG.LT.NLEGSI) GO TO 80

105  SMN=MIN1(SMN,MEND(IPHASE))
CALL GOLD(HN,HFINN,HIGH,W,FIL,XY,SMN,NERROR,IGEO,KPRNT)
SMN=SMNSAV
HFIN=XX
DELH=HFIN-HN1

110  HN=HFIN
CL=CL1
ALPHA=ALPHA1
SFC=SFC1
TN=TN1

140  CONTINUE
SUBROUTINE CLIMB

115
CD = CD1
WTOT = WTOT1
IF (NERROR .LT. 2 .AND. DELH .GT. 0.005*HN) GO TO 81
NERROR = 0
GO TO 140

12C
80 DELH = 60*ALOG1(1.0+FLOAT(LEG)*B2)/(1.0+FLOAT(LEG-1)*B2))
81 IF (VINDFS .GT. 0.0) GO TO 120
CALL AT62(HN, ANS)
RHO = ANS(1)
A = ANS(4)

IF (LEG .EQ. 1) VEL = SMN*A
IF (LEG .LT. 1) VEL = VEL/A
IF (SMN .LT. 0.0) NERROR = 2
IF (SMN .LT. 0.0) WRITE (6) IPHASE, LEG
610 FORMAT(1/2H ****NEGATIVE MACH NUMBER. // // )
IF (SMN .LT. 0.0) RETURN
Q* = 0.5*RHO*VEL**2
QS = Q*SING
CALL TRAJ31(2)

13C
CALL PROPICALC,NERROR, IGE, KGPRNT)
IF (NERROR .GT. 2) RETURN
CALL TRAJ01(1)
TN = THRUST*EN
TNU = THRUST*EN
IF (TN/W .LT. 1.5) GO TO 90

14C
CALL AEROICALC,NERROR, IGE, KGPRNT)
IF (NERROR .GT. 2) RETURN
CALL TRAJ01(1)
CD = CD0
GO TO 100

15C
CALL TRAJ01(1)
CALL AEROICALC,NERROR, IGE, KGPRNT)
IF (NERROR .GT. 2) RETURN
CALL TRAJ01(1)
CL = (w/QS)*SQRT((1.0-4.444*(TN/W)**2)
IAO = 0

16C
IF (SING .LE. 0.0) GO TO 110
COSG = SQRT(1.0-SING**2)

17C
DELT = DELH/(VEL*SING)
IF (DELT .LT. 0.3) GO TO 110
NERROR = 2
SUBROUTINE CLIMB

WRITE (6,603) IPHASE,Leg
603 FORMAT(/1111/134H *****FATAL ERROR IN CLIMB. PHASE=I2,7H Leg=I2/
1 26H *****NEGATIVE DELTA TIME////////
RETURN

110 DVD T=32.2*(IN-CD*QS-W*SING)/W
IF (DVDT .LT. 0.) SING=AMAX1(0.,(IN-CD*QS)/W)
IF (SING .LE. 0.) DVD T=32.2*(IN-CD*QS)/W
DRAG=CD*QS

180 IF (SING .LE. 0.) DELT=60.0
DELV=DVDT*DELT
IF (SING .LE. 0.) COSG=1.0
IF (SING .LE. 0.) DELH=0.0
DELX=VEL*DEL T*CD*QS/6080.

195 VEL=VEL+DEL V
DELW=TN*SFC*DELT/3600.

C-----TIME TO CLIMB
TIME=TIME+DELT/60.
C------FUEL USED DURING CLIMB
WF=WF+DEL W
C------GROUND DISTANCE COVERED DURING CLIMB
XCLIMB=XCLIMB+DELX

IF (W .GT. WPL) GO TO 115
IF (ERROR .NE. 2) 605
ELSE 607
RETURN

115 IF (IPRT(IPHASE) .NE. 0) GO TO 140
WRITE (6,605) IPHASE,Leg
605 FORMAT(/1111/134H *****FATAL ERROR IN CLIMB. PHASE=I2,7H Leg=I2/
RETURN

120 CALL AT62(HN,ANS)
121 RHO=ANS(1)
A=ANS(4)
C1=5*(RHO+RHD)
C2=SQR T(C1)
C3=20.51*C2
C4=C1*C2
C5=32.2*C3*DELT/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
WRITE (6,607) IPHASE,Leg,DELT,VINDFS,RHO,RHOP,TH,W,CD,SWING,DEL T,
1 C1,C2,C3,C4,C5,C6
607 FORMAT(/1111/134H *****FATAL ERROR IN CLIMB. PHASE=I2,7H Leg=I2/

SUBROUTINE CLIMB

1 40H ******NEGATIVE DELTA TIME IN VIND CLIMB, 5X, 5HDELH=E12,5,2X,
2 7VINDFS=E12,5,2X,4HRH0=E12,5,5X,5HRHO=E12,5,2X,3HTN=E12,5,
3 2X2HW=E12,5,5X,3HCD=E12,5,2X,6HSWING=E12,5,2X,5HDEL=E12,5/4

RETURN

125 TIME=TIME+DELT/60
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

SING=DELH/(VEL*DELT)
COSG=SQRT(1.-SING)**2
DELT=VEL*DELT*COSG/600.
XCLIMB=XCLIMB+DELT

CALL TRAJO1(2)
CALL PROP(ICALC,NERROR,IGEO,KPRNT)
IF (NERROR+GE.2) RETURN

TN=THRUST*EN
TNU=THRUST*EN
DELTW=TN/SFC/DELT/3600.
WF=WF+DELTW

IF (WN+LT.WPL) GO TO 130
IF (KERROR.EQ.2) WRITE (6,605) IPHASE,LEG
RETURN

130 C7=1.-((DELT/(VINDFS*DELT*C3)**2)
IF (C7+LT.0.0) C7=0.0
CL=2.*WSQRT(C7)/(002378*VINDFS**2*SWING)
IAO=8
CALL TRAJO1(2)
CALL AERO(ICALC,NERROR,IGEO,KPRNT)
IF (NERROR+GE.2) RETURN

CALL TRAJO1(1)
RHOP=RHO
IF (IPRT(IPHASE),EQ.0) GO TO 140
WRITE (6,608) LEG,VINDFS,SMN,HN,DELT,DELX,C6,VEL,DELWF,W,WN,CL,CD

C END OF CONSTANT INDICATED AIRSPEED SECTION

140 CONTINUE
CALL AT62(HN,ANS)
A=ANS(4)
SMN=VEL/A
RH0=ANS(1)
Q=.5*RHO*VEL**2
DXCRUS=DXCRUS+XCLIMB*FLOAT(IX(IPHASE))
BLTIME=BLTIME+TIME/60.
BLRANG=BLRANG+XCLIMB

C TEST FOR WEAPONS DROP

C----TEST FOR WEAPONS DROP
SUBROUTINE CLIMB

IF (I0(IPHASE).EQ.0) GO TO 150
WS=WS-WMISS

C-----TEST FOR BOMBS DROP

190 IF (IB(IPHASE).EQ.0) GO TO 160
WS=WS-WBOMB

C-----TEST FOR AMMUN DROP

160 IF (IA(IPHASE).EQ.0) GO TO 170
WS=WS-WAMMUN

170 IF (IPRT(IPHASE).EQ.0) GO TO 180
WRITE (6,609) SMN,HN,VEL,XCLIMB,DXCRUS,TIME,WF,W

609 FORMAT(/4X,5SHMN =E14.7,5X,4HHN =E14.7,4X,5HVEL =E14.7,
1 1X,8HXCCLIMB =E14.7,2X,8HDXCRU5 =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
TIME(IPHASE)=TIME
VELT(IPHASE)=VEL
SFC(IPHASE)=SFC

315 TMT(IPHASE)=TN
CDT(IPHASE)=CD
RLDT(IPHASE)=RlD
WEELT(IPHASE)=W
WATT(IPHASE)=WTOT

315 QT(IPHASE)=Q
SFCUT(IPHASE)=SFCU
TNUT(IPHASE)=TNU
CDINST(IPHASE)=CINS
PRTOTT(IPHASE)=PRTOT

320 XT(IPHASE)=XCLIMB
RETURN
END
SUBROUTINE CRUISE (ICALC, NERRR, IGED, KGPRNT)
REAL MSTART
COMMON /TRAJC/ ALPHA, ARW, CDL, CDG, CL, DESLF, DRAG, EN, HN, KP,
1 RLD, SMN, QMAX, RANGE, SFC, SWING, THRUST, TW, UTLF, W, WAMMUN, WETANK,
2 WFCUL, WGTOT, WMINS, WPLT, MSTART(12), HSTART(12), TIM(12),
3 PS1G(12), TDTST(12), NZST(12), PSIT(12), TDSTT(12), NSTT(12),
4 WBMBS, WFXET, SFCU, THRSTU, CDINS, PRTOT, WKFUEL, CRMACH, FLT0, X(12),
5 FLDN, TENDUR, BLRANG, BLTIME, TIM01, TIM02, WFT0, PNAME1(12),
6 PNAME2(12), SHTN(12), HNT(12), WFT(12), TIMET(12), XT(12), CLT(12),
7 CD(12), ALPHAT(12), CLIT(12), CDIT(12), ALPHIT(12), WFUSED, \$LAND,
8 QT(12), RLDT(12), SFTC(12), TNT(12), CET(12), DIT(18),
9 IAD, IPS, ITS, IW, IBS, NPHASE, IDY(14)
COMMON /TRAJE/ ALPIGT(12), CDINST(12), CD1GT(12), CL1GT(12), OCEL,
1 DELPH, DELMP, DCRUS, FLAC, FRFRE, HEND(12), HENDUR, HMAXP, HMINP,
2 HPATH(50), MEND(12), MENDUR, NT(12), PRTOTT, RADIT(12),
3 RADST(12), SFC, SFCUT(12), SMMAXP, SMMINP,
4 TNU(12), VIND(12), VELT(12), VPATH0(10), WART(12), WCOMBP, WELT(12),
5 WRES, WFTOT, WFTO1, WFTO2, WFTP, WKLAND, XDESC, XGRLAN, XGWMAX, TOL,
6 WFIXED, WCAC, WGT0, DUHY(82), IA(12), IAS, ID(12), IBREG, IENDUR,
7 IMISS, IP(12), IPHASE, IPTLAT, IPRT(12), IPSIZE, IPST01, IPST02,
8 IN(12), IX(12), KERRR, MILCDM, NRCREASE, NLEGCL, NLEGCR, NLELO, NMISS,
9 NPATH, IDUUM(6)
DIMENSION ANS(4), HNTAB(10), SHTAB(10)
DATA HNTAB/0.0, 5000.0, 10000.0, 15000.0, 20000.0, 25000.0, 30000.0,
1 35000.0, 37000.0, 100000.0/
DATA SHTAB/3450.0, 3290.0, 3140.0, 3050.0, 2940.0, 2820.0, 2710.0,
2 2590.0, 2090.0, 2090.0/
DATA DMON/0.0/
IF (IPRT(IPHASE).NE.0) WRITE (6, 600)
C-----TEST FOR TANKS DROP------
600 FORMAT('///19H CRUISE PHASE//')!
        ORIGINAL PAGE IS
OF POOR QUALITY.
C-----TEST FOR TANKS DROP------
WFMUSD=WGST-W
IF (IWS.EQ.0) WFUSED=WFMUSD-WMSS
35 IF (IBS.EQ.0) WFMUSD=WFMUSD-WBMBS
35 IF (IAS.EQ.0) WFMUSD=WFMUSD-WAMMUN
35 IF (ITS.EQ.1 .AND. WFMUSD.GE.WFXET) W=W-WETANK
35 IF (ITS.EQ.1 .AND. WFMUSD.GE.WFXET) ITS=0
35 IPS=0(IPHASE)
35 SNW=MSTART(IPHASE)
35 HN=HSTART(IPHASE)
40 IF (SNW.GT.0.0) SNW=SNW-1
40 IF (HN.GT.0.0) HN=HN-1
40 IF (IBREG.EQ.0 .OR. HSTART(IPHASE).NE.0.0) GO TO 10
40 CALL GOLDEN(2000, 50000, 1, 12, XX, Y, NERRR, IGED, KGPRNT)
45 IF (NERRR.GE.2) RETURN
45 HN=XX
45 CALL AT62(HN, ANS)
45 A=ANS(1)
45 VE=SNW*A
40 CL=2.*W/(RHO*SWING*VEL**2)
40 IAD=0
40 CALL TRAJO12)
40 CALL AERO(ICALC, NERRR, IGED, KGPRNT)
45 IF (NERRR.GE.2) RETURN
45 CALL TRAJO1(1)
45 COSA=COS(ALPHA/57.3)
SUBROUTINE CRUISE

TIME=0.0
WF=0.0

IF (X(IPHASE).GE.0.) XCRUSE=X(IPHASE)+XDESC*FLOAT(IX(IPHASE))
IF (X(IPHASE).LT.0.) XCRUSE=Range/FLOAT(NCRUSE)+DXCRUS+
1  XDESC*FLOAT(IX(IPHASE))
IF (XCRUSE .LE. 0.) 6XCRUS*0.0
IF (XCRUSE .LE. J.) GO TO 90

IF ((X(IPHASE).GE.100.).AND.(X(IPHASE).LT.3000.)) NLEGS=3
IF ((X(IPHASE).GE.1000.).AND.(X(IPHASE).LT.10000.)) NLEGS=4
IF ((X(IPHASE).GE.3000.).AND.(X(IPHASE).LT.100000.)) NLEGS=5
IF ((X(IPHASE).GE.30000.).AND.(X(IPHASE).LT.1000000.)) NLEGS=6

IF (NLEGCR.GT.0) NLEGS=NLEGCR

DELX=XCRUSE/FLOAT(NLEGS)

E601 FORMAT(16X3,601HSMN=,5HSMN=,5X4HSMN)

IF (IPRT(IPHASE).EQ.0) GO TO 20

WRITE (6,601) W,SMNSHNYRWSVELOASLPHA,TIMESWF,XCRUSE,DXCRUS,

1 DELX,IAO,IPS

E601

DO 50 50 LEG=1,NLEGS

IF (HEND(IPHASE).EQ.-1.) GO TO 21

RH01=RH0

RHO2=CL*SWING*VEL**2

CALL TAIN(HN,HANS) = HN-HN+2*SH*(RHO1—RHO2)/(RHO1+RHO2)

CALL AT62(HN,ANS)

A=ANS(4)

VEL=SMN*A

RH0=2.2*W/(CL*SWING*VEL**2)

CALL AT62(HN,ANS)

A=ANS(4)

WH0=ANS(1)

VEL=SMN*A

EQ=5*RH0*VEL**2

Q0=Q*SWING

GO TO 22

21 HN=HSTART(IPHASE)

CALL AT62(HN,ANS)

A=ANS(4)

WH0=ANS(1)

VEL=SMN*A

EQ=5*RH0*VEL**2

Q0=Q*SWING

CALL AERO(1CALC,NERROR,IGEO,KGPRNT)

IF (NERROR.GE.2) RETURN

CALL TRAJ01(1)

DRAG=CD*QS

CALL TRAJ01(2)

CALL PROFIICALC,NERROR,IGEO,KGPRNT

IF (NERROR.GE.2) RETURN

CALL TRAJ01(1)

TN=THRUST*EN

TNU=THRUST*EN

IF (ABS(DRAG/TN-1.0).LE.0.03) GO TO 30

IF (KERROR.EQ.2) WRITE (6,602) IPHASE,LEG,DRAG,TN

6C2 FORMAT(17H IN CRUISE, PHASE,12,5H, LEG,12, 46H, THRUST DOES NOT

1 MATCH DRAG WITHIN 3 PERCENT, 7H DRAG =,E12.5,4X,4HTN =,E12.5/)
SUBROUTINE CRUISE 76/76 OPT=2 FTN 4.5+410

15
30 DELWF=.8444*W*SFC*CD*DELX*SQR(2*RHO*SWING/(CL*W))/CDSA

C----FUEL USED DURING CRUISE

W=W+DELWF

DELT=6080.*VEL/DELX

C----CRUISING TIME TO TARGET

TIME=TIME+DELT/60.

W=W-DELWF

C----TEST FOR TANKS DROP

WFUSED=WGT=W

IF (IWS.EQ.0) WFUSED=WFUSED-WMISS

IF (IBS.EQ.0) WFUSED=WFUSED-WBOMBS

IF (IBS.EQ.0) WFUSED=WFUSED-WAMMUN

IF (ITSD.EQ.1) AND (WFUSED.GE.WEXT) W=W-WETANK

IF (ITSD.EQ.1) AND (WFUSED.GE.WFEXT) ITS=0

IF (IPRINT(IPHASE).EQ.0) GO TO 40

WRITE (6,603) LEG, RHO1, RHO2, SH, HN, A, VEL, RHO, Q, QS, CD, DRAG, TN, SFC,

603 FORMAT(/4X,5HLEG -12/3X,6HRHO1 =E14.7,3X,6HRHO2 =E14.7,5X,4HSH =

1.14.7,7X,6X,5X,4HNN =E14.7,7X,5X,4HVEL =E14.7,7X,5X,4HRHD =


3 1.14.7,7X,5X,4HTDELT =E14.7,7X,5X,4HTIME =E14.7,6X,3HW =E14.7/)

4 IF (WG.T.EQU.0) GO TO 50

IF (KERROR.EQ.0) WRITE (6,604) IPHASE, LEG

604 FORMAT(/127H W.LE.WPL IN CRUISE.: PHASE: I2,TH LEG=,I2/)

50 CONTINUE

BLTIME=BLTIME+TIME/60.

BLRANG=BLRANG+XCRUSE

DXCRUS=0.0

C----TEST FOR WEAPONS DROP

IF (IW(IPHASE).EQ.0) GO TO 60

W=W-WMISS

IWS=0

C----TEST FOR AMMO DROP

IF (IA(IPHASE).EQ.0) GO TO 70

W=W-WAMMUN

IAS=0

80 IF (IPRINT.EQ.0) GO TO 90

WRITE (6,605) W

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

SFCY(IPHASE)=SFC

TNT(IPHASE)=TN

CDT(IPHASE)=CD

RLDT(IPHASE)=RLD

WEELT(IPHASE)=W

ORIGINAL PAGE IS OF POOR QUALITY
SUBROUTINE CRUISE  76/76  OPT=2  FTN 4.5410

\begin{verbatim}
  \texttt{\textbf{\texttt{\texttt{\texttt{\texttt{\texttt{\texttt{
WART(IPHASE) = WTOT
QT(IPHASE) = 0
SFCUT(IPHASE) = SFCU
THUT(IPHASE) = TNU
CDINST(IPHASE) = CDINSP
PRTOTT(IPHASE) = PRTO\nXH(IPHASE) = XCRUSE
RETURN
END
\end{verbatim}}}}}}
\end{verbatim}
SUBROUTINE ACCEL
REAL MSTART, MEND
COMMON /TRAJCM/ ALPHAP, ARW, CD, CDL, CDO, CL, DESLF, DRAG, EN, HN, KP,
1 RL0, SMH, QMAX, RANGE, SFC, SWING, THRUST, TW, ULTLF, W, WANGRUN, WETANK,
2 WFUEL, WGTOWT, WMISS, WDT, WLPW, MSTART(12), HSTART(12), TIM(12),
3 PS1GT(12), TDTST1(12), NIST(12), PSIT(12), TDT1T1(12), NIST(12),
4 BOMBS, WFEX, SFIC, THRSTU, CDINS, PRTDT, WKFUEL, CRMACH, FLT0, X1(12),
5 FLAND, TENDUR, BLRANG, BLTIME, TMT0, TMT02, WFTD, PNAME1(12),
6 PNAME2(12), SMNT(12), HNT(12), WFT1(12), TIMET1(12), X1(12), CLT(12),
7 CDIT(12), ALPHAT(12), CLIT(12), CDIT(12), ALPHIT(12), WFUSED, WLAND,
8 QT(12), RLD1T(12), SFT(12), TMT(12), CET(12), DY(18),
9 IA0, IPS, ITS, IWS, IBS, NPHASE, IPRT(14)
COMMON /TRAJEX/ ALP1GT(12), CDNST(12), CL1GT(12), DEC1,
1 DELH, DELM, DXCRUS, FLFAC, FRFURE, HEND(12), HENDUR, HMAXP, HMINP,
2 HPATH(50), MEND(12), MENDUR, NT(12), PRTDTT(12), LADT(12),
3 RADST(12), SFBSTCUT(12), SMAXP, SMINP,
4 TNI(12), VIN(12), VELT(12), VPATHT(50), WART(12), WCOMB, WELLT(12),
5 WFRS, WFTOT, WFTOL, WFT02, WFT2, WKLAND, XDESC, XGRAN, FWGMA, TOL,
6 WFIXED, WGCALC, WPC, WGT0, WFT0, WFT, WFT1, WFT11, WFT11, WFT11,
7 IMISS(12), IPHASE, IPRT(12), IPST01, IPST02,
8 IMDb(12), IX(12), IP(12), IAO, IP(IH), ICRC1, ICRC2, IENDUR, IDUMY(6),
9 NPATH, IDUMY(6)
DIMENSION ANS(4)
IF (IPRT(IPHASE).NE.0) WRITE (6,600)
FORMAT //25H ___ ACCELERATION PHASE/1
IF (HSTART(IPHASE).LT.0.0) SMN=SMNT(IPHASE-1)
IF (HSTART(IPHASE).GT.0.0) SMN=MSTART(IPHASE)
IF (HSTART(IPHASE).LT.0.0) HN=HNT(IPHASE-1)
IF (HSTART(IPHASE).GT.0.0) HN=HSTART(IPHASE)
30 CALL AT62(HNRANS)
55 IF (LEG.GT.10) GO TO 85
C-----COMPUTE INTEGRALS SUMWF AND SUMDT BY TRAPEZIODAL RULE
SUMWF=0.0
SUMDT=0.0
XACCEL=0.0
TIMECR=0.0
FUELCR=0.0
WF=0.0
TIME=0.0
40 LEG=0
IF (IPHASE.LT.0) GO TO 10
WRITE (6,601) LEG, HN, ARW, RHO, TEMP, SUMWF, SUMDT, SMN, W, IAO, IPS
601 FORMAT(4X,5HLEG =12,5X,4HHN =E14.7,6X,3HARW =E14.7,4X,
1 5HRHO =E14.7,3X,6HTEMP =E14.7,2X,7HSUMWF =E14.7/2X,7HSUMDT =,
2 E14.7,4X,5HSMN =E14.7,6X,3HHIAD =I2,16X,5HIPS =I2/)
10 IF (HSTART(IPHASE).GT.0.0) THEN MEND(IPHASE) GO TO 120
IF (SMNT(IPHASE-1).GT.0.0) THEN MEND(IPHASE) GO TO 120
C-----START OF LCDS
20 C**************************************************************
LE4=LE4+1
IF (LE4.GT.50) GO TO 80
CALL TRAJ21(2)
CALL PROP(ICALC, NERROR, IGE0, KGPRNT)
43
SUBROUTINE ACCEL 76/76  OPT=2  FTN 4.5+410 05

IF (NERRR.GE.2) RETURN
CALL TRAJ01(1)

TH=THRUST*EN
INU=THRUST*EN
VEL=SMN*A

CL=2.*W/(RHO*SWING*VEL**2)
CALL TRAJ01(2)

CALL AERDICALC, NERRR, IGE0, KGPRNT)

IF (NERRR.GE.2) RETURN
CALL TRAJ01(1)

DENOM=TN*COS(ALPHA/57.3)-.5*CD*RHO*SWING*VEL**2

YWP=SFC*TN/DENOM

YDT=(1./DENOM)

IF (YWP.GT.0.0) AND (YDT.GT.0.0) GO TO 50

DTIME=60.

DFUEL=TN*SFC*DTIME/3600.

W=W-DFUEL

IF (W.GT.WPL) GO TO 40

IF (KERROR.EQ.2) WRITE (6)602)

RETURN

GO TO 30

50 IF (LEG.EQ.1) GO TO 60

SUMWF=SUMWF+5*(YWF+YWP)*((SMN-SMN)

SUMDT=SUMDT+5*(YDT+YDTP)*((SMN-SMN)

DELX=7625*(YDT+YDTP)*((SMN-SMN)*VEL+W*SQTEMP/6080.

XACCEL=XACCEL+DELX

WF=EXP(1.525*SUMWF*SQTEMP/3600.)-1.)*W

DELWF=WF-WFP

W=W-DELWF

TIME=1.525*SUMDT*SQTEMP+W/60.

DELT=TIME-TIMEP

IF (IPRT(IPHASE).EQ.0) GO TO 70

WRITE (6,603) LEG, TN, SFC, VEL, SMN, ALPHA, CL, CD, DENOM, YWF, YDT, SUMWF,

1 DELWF, SUMDT, DELT, DELX, XACCEL

603 FORMAT(4X,LEGL,12/5X,4HTN =E14.7,4X,5HSFC =E14.7,4X,

1 5HVEL =E14.7,4X,5HSMN =E14.7,2X,5HALPHA =E14.7,5X,4HCL =E14.7,5X,

2 4HCD =E14.7,2X,7HDENOM =E14.7,4X,5HYWF =E14.7,4X,5HYDT =E14.7/2

3 2X,7HSUMWF =E14.7,2X,7HDELWF =E14.7,2X,7HSUMDT =E14.7,

4 3X,6HDELX =E14.7,3X,6HACCEL =E14.7/1

C---- MAKE TEST TO SEE IF YOU ARE DONE. IF SO, JUMP OUT OF LOOP.

70 IF (SMN.GE.MEND(IPHASE)) GO TO 80

YWP=YWF

YDTP=YDT

SUMF=SMN

WFP=WF

TIME=TIMEP

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
SUBROUTINE ACCEL

C-----FULL FUELED DURING ACCELERATION RUN
50  WF=WF+FUELCR

C-----TIME ELAPSED DURING ACCELERATION RUN
120  TIME=TIME+TIMECR
     BLTIME=BLTIME+TIME/60
     BLRANG=BLRANG+ACCEL
     IF (WGT*WPL) GO TO 90
     IF (KERROR.EQ.2) WRITE (6,604) IPHASE
       FORMAT(126H W*LE.*WPL IN ACCEL. PHASE=*12/
     RETURN

125  90  VEL=SMN*A
     W=W-FUEL
     DXCRUS=DXCRUS+ACCEL*FLOAT(I(IPHASE))

C-----TEST FOR WEAPONS DROP
130  IF (IW(IPHASE).EQ.0) GO TO 109
     W=W-WMISS
     IWS=0

C-----TEST FOR BOMBS DROP
100  IF (IB(IPHASE).EQ.0) GO TO 110
     W=W-WBOMBS
     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, W
       605  FORMAT(7H SUMWF -E14.7, 7H SUMDT -E14.7, 6H TIME -E14.7,
                         6H VEL -E14.7, 6H Q -E14.7, 8H XACCEL -E14.7,
                         6H DXCRUS -E14.7, 6H W -E14.7)

145  130  SMNT(IPHASE)=SMN
       HNT(IPHASE)=HN
       CLT(IPHASE)=CL
       ALPHAT(IPHASE)=ALPHA
       WFT(IPHASE)=WF

150  TMT(IPHASE)=TIME
       VLT(IPHASE)=VEL
       SFC(IPHASE)=SFC
       TNT(IPHASE)=TN
       CDT(IPHASE)=CD

155  RLOT(IPHASE)=RLO
       WELT(IPHASE)=W
       WART(IPHASE)=WTOT
       QT(IPHASE)=Q
       SFCUT(IPHASE)=SFCU
       TNUT(IPHASE)=TNU
       CDINST(IPHASE)=COINSP
       PRTOTT(IPHASE)=PRTOT
       XT(IPHASE)=XACCEL

160  RETURN

END
SUBROUTINE COMBAT(ICALC, NERROR, IGE, KGPRNT)

REAL MSTART, NTURNS, LIFT, NZS, NZT, NZIT
COMMON /TRAJCM/ ALPHAs, ARW, CD, COL, CDQ, CL, DESLF, DRAG, EN, HN, KP,
 1 REX, SMN, DEL, RANG, SFC, SWING, THRUST, TWF, UTLF, W, WMMUN, WETANK,
 2 WFLUE, WGTOWT, WMISS, WDT, WPLWT, MSTART(12), NT(12), TIM(12),
 3 PSI(MT(12), TODIST(T(12), NZS, T(12)), PSIT(12), TODIST(T(12)), NZIT(12),
 4 WOBMS, DEFT, SFCU, THRSTU, COINS, PRTOT, WKFUE, CRMACH, FLT0, X(12),
 5 FLAND, TENDUR, BLRANG, BTIME, TIMO, TIMO2, WFTO, PNAME1(12),
 6 PNAME2(12), SMNT(12), HNT(12), WFS(12), TIMET(12), XT(12), CLT(12),
 7 CDT(12), ALPHAT(12), CLIT(12), CDT(12), ALPHIT(12), WFXED, WLAND,
 8 Q(F1), RLD(12), SFC, T(12), NT(12), CET(12), DNY(12),
 9 IAO, IPS, ITS, IWS, IBS, NPHASE, IDY(14),
 COMMON /TRAJEX/ AL1GT(12), COINS(12), CD1GT(12), CL1GT(12), DECCL,
 1 DELP, DELMP, BXRUS, FLFAC, PRFUSE, HEND(12), HENDUR, HMAXP, HMINP,
 2 HPATH(10), MEND(12), MENDU, NT(12), PRTOT(12), RADIT(12),
 3 RADST(12), SF, SFCUT(12), SMAXP, SMINP,
 4 TPATH(12), VIND(12), VMIN(12), VPATH(10), WAPT(12), WCMBP, WELT(12),
 5 WRES, WFTOT, WFT01, WFTO2, WFRAP, WKLST, XDESC, XSRAN, FGMMAX, TDL1,
 6 WFXED, WGCALC, WPL, WGT0, DUMY(62), IA(12), IAS, IB12(12), IREG, ENDUR,
 7 IMISS, IP(12), IFHASE, IPLOT, IPRT(12), IPSIZE, IPTST1, IPTST2,
 8 IN(12), IX(12), KERROR, MILCOM, MCRUSE, NLEG, NLEGCR, NLLEG, NMISS,
 9 NPATH, DUMY(6),
 DIMENSION ANS(4)
 HN=MSTART(IPHASE)
 25 IF (MSTART(IPHASE).LT.0.0) HN=HNT(IPHASE-1)
 CALL AT62(HN, ANS)
 RHO=AN5(1)
 A=AN5(4)
 SMN=MSTART(IPHASE)
 30 IF (MSTART(IPHASE).LT.0.0) SMN=SMNT(IPHASE-1)
 VEL=SMN*A
 Q=Q*RHO*VEL*2
 QS=Q*SWING
 XCOMB=0.0
 35 IF (IPRT(IPHASE).NE.0) WRITE (6,600)
 600 FORMAT (1/19H COMBAT PHASE
         1 6X,3H=1E14.7,5X,4H=H=1E14.7,6X,3HA=1E14.7,
 2 4X,3H=1E14.7,4X,5H=VEL=1E14.7,6X,3H=1E14.7,
 3 2X,7H=CMCOM=1E14.7)
 C---------------
 C ONE G FLIGHT
 C---------------
 CL=W/OS
 IAO=8
 45 CALL TRAJ01(2)
 CALL AERO(ICALC, NERROR, IGE, KGPRNT)
 IF (NERROR.GE.2) RETURN
 CALL TRAJ01(1)
 DRAG=CO*OS
 50 IPS=IP(IPHASE)
 CALL TRAJ01(2)
 CALL PROP(ICALC, NERROR, IGE, KGPRNT)
 IF (NERROR.GE.2) RETURN
 CALL TRAJ01(1)
 TN=THRUST*EN
 TNU=THRUST*EN
 PS1G=(TN-DRAG)*VEL/W
SUBROUTINE COMBAT 76/76 OPT=2

PS1G(IPHASE)=PS1G
ALPHA(IPHASE)=ALPHA
CL(IPHASE)=CL
CD(IPHASE)=CD

IF (IPRT(IPHASE).EQ.0) GO TO 10
WRITE (6,601) IAO, CL, CD, ALPHA, DRAG, IPS, TN, SFC, PS1G


C SUSTAINED TURNS

10 NZS=1.001
   IF (PS1G.LE.0.) GO TO 40
   IAO=4
   CALL TRAJ01(2)
   CALL AER(ICALC, NERROR, IGEO, KGPRNT)
   IF (NERROR.GE.2) RETURN
   CALL TRAJ01(1)
   LIFT=CL*Q8
   NZS=(LIFT+TN*SIN(ALPHA/57.2958)/W)
   IF (IPRT(IPHASE).EQ.0) GO TO 30
   WRITE (6,603) IAO, CL, CD, ALPHA, DESLF, NZS


30 IF (NZS.GT.DESLF) GO TO 50
   NZS=DESFLF
   IF (IPRT(IPHASE).EQ.0) GO TO 30
   WRITE (6,605) NZS

605 FORMAT(1X, I20, 12FE14.7)

50 PHI=ARCOS(I.GINZS)

C TURN RADIUS

105 RADIUS=VEL*2/(32.2*TAN(PHI))

C TURNING TIME OR NUMBER OF turns
TIME=TIM(IPHASE)
NTURNS=NTIM(IPHASE)
   IF (TIME.LE.0.0) TIME=NTURNS*RADIUS/(9.55*VEL)

110 IF (NTURNS.LE.0.0) NTURNS=9.55*TIME/RADIUS
   XCOMB=VEL*TIME/60./6080.
   BLTIME=BLTIME+TIME/60.
   BLRANG=BLRANG+XCOMB

C FULL USED DURING TURNS

47
SUBROUTINE COMBAT 7c/7e  OUt=2

115  **W** = SFC * TN * TIME / 60.

120  C COMBAT ENERGY

CE = 3600. * **W** / PS1G / (SFC * TN)
TDOTS = 57.3 * VEL / RADIUS
RADS = RADIUS

125  IF (IPRT(IPHASE),EQ.0) GO TO 60
WRITE (6,616) PHIS, RADIUS, TIME, NTURNS, **W**, TDOTS, XCMB

616  FORMAT (2x, 16x, PHIS, E14.7, 1x, RADIUS, E14.7, 1x, TIME, E14.7,
1 1x, NTURNS, E14.7, 5x, 4**W**, E14.7/2x, 7TDOTS, E14.7,
2 2x, 7XCMBO, E14.7/)

130  SMNT(IPHASE) = SMN
HNT(IPHASE) = HN
CLT(IPHASE) = CL
ALPHAT(IPHASE) = ALPHA
XCMBO = **W**

135  VELT(IPHASE) = VEL
TIMET(IPHASE) = TIME
SFT(IPHASE) = SFC
NTN(T) = TN
CDT(IPHASE) = CD

RDLT(IPHASE) = RLD
WRT(IPHASE) = WTD
QT(IPHASE) = Q

140  SFCUT(IPHASE) = SFCU
TNUT(IPHASE) = TNU
COUT(IPHASE) = COINSP
PRIO(T/IPHASE) = PRTOT
XT(IPHASE) = XCMB
TDOTS(IPHASE) = TDOTS
RADST(IPHASE) = RADS

145  NZST(IPHASE) = NZS
CEQ(IPHASE) = CE

-----------------------------------
C MAXIMUM INSTANTANEOUS CONDITIONS
-----------------------------------

150  IA0 = 5
CALL TRAJ01(2)
CALL AEROICALC(NERROR, IGE0, KGPRNT)
IF (NERROR.GE.2) RETURN
CALL TRAJ01(1)

155  CLMAX = CL
LIFT = CLMAX + OS

C LOAD FACTOR
NZI = (LIFT + TN * SIN(ALPHA/57.2957)) / W
IF (IPRT(IPHASE),EQ.0) GO TO 70
WRITE (6,617) IA0, CL, CD, ALPHA, LIFT, NZI

617  FORM(1X, 19HINSTANTANEOUS TURNS/ 4X, 5HIAD = I2.17, 4HCL = E14.7,
1 5X, 4HOC = E14.7, 2X, 7HALPHA = E14.7, 3X, 6HLIFT = E14.7/
2 4X, 5HNI = E14.7/)

70  IF (NZI.LE.DESLF) GO TO 80
IA0 = 6
CALL TRAJ01(2)
CALL AEROICALC(NERROR, IGE0, KGPRNT)
IF (NERROR.GE.2) RETURN
CALL TRAJ01(1)
NZI = DESLF
IF (IPRT(IPHASE),EQ.0) GO TO 80

48
SUBROUTINE COMBAT

WRITE (6,608) IAUP,CL,CD,ALPHA,DESLF,NZI

608 FORMAT(4X,5HIAO=12,17X,4HCL=E14.7,5X,4HCD=E14.7,1
2X,7HALPHA=E14.7,2X,7HDESLF=E14.7/4X,5HNZI=E14.7/1)

175 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=,12,5X,1
4HNZI=1PE11.4).

RETURN

18C C PS FOR MAXIMUM INSTANTANEOUS TURN RATE

90 DRAG=CD*05

PSI=(TN*COS(ALPHA/57.3)-DRAG)*VEL/W

ALPHI=ALPHA

CLI=CL

CDI=CD

PHI=ARCSIN(1.0/NZI)

C MAXIMUM INSTANTANEOUS TURN RATE AND RADIUS

RADIUS=VEL**2/(32.2*TAN(PHI))

TDOTI=57.3*VEL/RADIUS

19G RADI=RADIUS

W=W-DF

IF (W.GT.WPL) GO TO 100

IF (KERROR.EQ.2) WRITE (6,610) IPHASE

610 FORMAT(/27H WEPL IN COMBAT. PHASE=s12).

195 RETURN

C TEST FOR WEAPONS DROP

1C0 IF (IA(IPHASE),EQ.0) GO TO 110

W=W-WMISS

IWS=0

20C C TEST FOR BOMBS DROP

110 IF (IB(IPHASE),EQ.0) GO TO 120

W=W-WBOMBS

IBS=0

C TEST FOR AMMO DROP

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,TDOTI,W

21C 611 FORMAT(4X,6HDRAG=E14.7,4X,5HPSI=E14.7,4X,5HPHI=E14.7,1
2X,6HRADIUS=E14.7,2X,7HTDOTI=E14.7/6X,3HW=E14.7/1)

140 WEELT(IPHASE)=W

PSIT(IPHASE)=PSI

TDOTIT(IPHASE)=TDOTI

RADIT(IPHASE)=RADI

NZT(IPHASE)=NZI

ALPHIT(IPHASE)=ALPHI

CLIT(IPHASE)=CLI

CDIT(IPHASE)=CDI

22C RETURN

END
SUBROUTINE LOITER

REAL MSTART(KP, 4), SNLOIT

COMMON /TRAJCM/ ALPHA, ARM, CD, CDL, CDQ, CL, DESLF, DRAG, EN, HN, KP,
1 RIO, SH, QMAX, RANG, SFC, SWING, THRUST, TW, ULTLF, WAMMUN, WTANK,
2 WFUEL, WGTOM, WMISS, WTOP, WPLT, MSTART(12), HSTART(12), TITM(12),
3 PSIGT(12), TOTGT(12), NEST(12), PSIT(12), TOTIT(12), NITM(12),
4 WOBMS, WFEXT, SFCUT, SDINS, PRSTOT, WKFUEL, CRMACH, FLTO, X(12),
5 FLTLO, TENDRK, BLRANG, BLTIME, TNTM(12), NTM(12), PNAME1(12),
6 PNAME2(12), SMNT(12), HNT(12), WFT(12), TIMT(12), XMT(12), CLT(12),
7 CDT(12), ALPHAT(12), CLIT(12), CDT(12), ALPHIT(12), WFUSED, HLAND,
8 GT(12), RLT(12), SFCU(12), THT(12), CET1(12), OY1(18),
9 IA0, IPS, ITS, IWS, SMS, NPHASE, IDY(14)

COMMON /TRAJEX/ ALPHT1(12), CD1G1T(12), CL1GT(12), DE1C1L,
1 DELHP, DELPX, DXCRUS, FLAC, FFRE, HEND1(12), HENDUR, HMAXC, HMINP,
2 HPATHT(50), HEND1(12), RENDUR, NIT1(12), PRSTOTT(12), RAD1T(12),
3 RADST(12), SD, SFCUTL(12), SHMAXP, SHM1P,
4 TNUM(12), WIND(12), VELT(12), WPAHTH(50), WART(12), WCMB, WEE1T(12),
5 WFRED, WFOT, WFT01, WFT02, WFT03, WFMAT, WLAND, XDESC, XGRLA, XGMLA,
6 WFP1, WFCALC, WFL, WFT04, DUMY2(12), DUMY1(12), DBREG, EENDUR,
7 IMSS, IPI(12), IPIH1E, IPILO, ITRT(12), IPSIZE, IPTSTO1, IPTSTO2,
8 IWT(12), IX(12), KERROR, MILCOM, MCSELF, NLEG, NLEG1, NLEG2, NMISS,
9 NPATH, IDUMY(6)

DIMENSION ANS(4), HNTAB(10), SHTAB(10)

DATA HNTAB/0., 5000., 10000., 15000., 20000., 25000., 30000.,
1 35000., 37000., 100000. /

DATA SHTAB/34500., 32900., 32050., 30300., 28400., 26200., 27110.,
1 25900., 20900., 20900. /

DATA DMON/0.01

IF (IPRT(IPHASE).NE.0) WRITE (6, 600)

FORMAT(/19H LOITER PHASE//)

600 SMNIN=MSTART(IPHASE) .-

SNIN=SMNT(IPHASE-1)

SMN=SMIN

HN=HSTART(IPHASE)

IF (HSTART(IPHASE).LE.0.0) HN=HNT(IPHASE-1)

IF (IENDUR.EQ.0.0.OR.HSTART(IPHASE).NE.0.0) GO TO 10

CALL GOLDEN(ZSSOC, 65000., 2, XX, Y, NERROR, IGE0, KGPRNT)

IF (NERROR.GE.2) RETURN

CALL AT62(H1, HN, ANS)

RHG=ANS(1)

A=ANS(4)

VEL=SMN*A

IF (SMN.GT.C.0) CL=2.*W/(RHO*SWING*VEL**2)

TIME=TIM(IPHASE)

NLEG=TIME/60.+1.0

IF (NLEG.LT.1.0) NLEG=1

IF (NLEG.GT.6.0) NLEG=6

IF (NLEG.EQ.0.0) NLEG=NLEG0

DELT=TIME/FLOAT(NLEG)

W=0.0

XLOIT=0.0

IP = IMSS(IPHASE), EQ=0.0) GO TO 20

WRITE (6, 601) SAMIN, SMNS, HN, RHOG, A, VEL, CL, TIME, NLEG, DELT, WF, XLOIT

601 FORMAT(2X, 7HSMIN =E14.7, 4X, 5HSMN =E14.7, 5X, 5HVEL =E14.7, 5X, 5HCL =E14.7)
SUBROUTINE LOITER  76/76  OPT=2

2  14.7,3X,6HTIME =E14.7,2X,7HNLGS =12,15X,6HDELT =E14.7/5X,
3  4HWF =E14.7,2X,7HXLUIT =E14.7/

60   DO 90 LEG=1,NLGS
   IF (SMINGT.0.0) GO TO 30
   C-----PERFORM OPTIMUM LOITER MACH NO. SEARCH.
   CALL GOLDE(1.9,9,38,X,Y,NERRO,IGEO,KGPRNT)
   IF (NERRO.GE.2) RETURN
   SMN=XX
   VEL=SMN*A
   IF (LEG.EQ.1) CLOUM = 2.0/(RHO*SWING*VEL**2)
   CL=CLOUM
   GO TO 40

7C  C-----OPTION TO INCREASE ALTITUDE AS A/C WEIGHT DECREASES, MACH KEPT CONSt

30   RH01=RHO
   RH02=2.0/(CL*SWING*VEL**2)
   CALL TAINI(HN,TABPSHTAB,HNYSFIP,LPN,NERROR,DMON)
   HN=HN+2.0*SH*(RH01-RH02)/(RH01+RH02)
   CALL AT62(HN,ANS)
   A=ANS(4)
   VEL=SMN*A
   RHO=2.0/(CL*SWING*VEL**2)
   C-----END OF INCREASE ALTITUDE OPTION

8C  40   IAD=6
   CALL TRAJO1(2)
   CALL AERO(ICALC,NERRO,IGEO,KGPRNT)
   IF (NERRO.GE.2) RETURN
   CALL TRAJO1(1)
   DRAG=5.0*RHO*SWING*CD*VEL**2
   CALL TRAJO1(2)
   CALL PROP(ICALC,NERRO,IGEO,KGPRNT)
   IF (NERRO.GE.2) RETURN
   CALL TRAJO1(1)
   TN=THRST*EN
   TNU=THRSTU*EN
   C-----CHECK LOITER RADIUS.
   RALOIT=X(IOPHASE)
   IF (RALOIT.LE.0.0) GO TO 60
   NZLOIT=(5.0*CL*RHO*SWING*VEL**2+TN*SN(ALPHA/57.3))/W
   IF (NZLOIT.LT.1.00001) CZSHI = 1.0/NZLOIT
   C-----RADIUS FOR OPTIMUM LOITER.
   RALOIT=(5.11E-6*VEL**2+COSH/2)/SQR(1.-COSH)**2
   IF (RALOIT.LE.RALOIT) GO TO 60
   DO 50 J=1,3
   DRAG=DRAG
   CL=(W/(SWING*5*RHO*VEL**2))*SQR(1.+(5.11E-6*VEL**2/RALOIT)**2)
   IAD=8

50   CALL TRAJO1(2)
   CALL AERO(ICALC,NERRO,IGEO,KGPRNT)
   IF (NERRO.GE.2) RETURN
   CALL TRAJO1(1)
   DRAG=5.0*RHO*SWING*CD*VEL**2
   IF (ABS(DRAG/DRAGP-1.0).LE.0.03) GO TO 60
   IF (JaE0.3) WRITE (6,6021 IPHASE,LEG
   602 FORMAT(155H INSUFFICIENT THRUST FOR REQUIRED LOITER RADIUS, PHASE
   1*12,7H  LEG*12/1
   CALL TRAJO1(2)

   C-----END OF CODE
CALL PROPCALC,NE,ROR,IGL,KGPRNT)
IF (NERROR.GE.2) RETURN
CALL TRAJ01(L)
TH=THRST*EN
TH=THRST*EN
50 CONTINUE
60 TEST=DRAG/TH
IF (TAU(T,S,T-1.0).LE.0.05) AND (SMNIN.LE.3.0) GO TO 70
IF (TEST.GT.1.0) TH=DRAG
CD=2.*TH/(KH*SWING*VEL**2)
25 DFRAG=5*CD*SWING*CD*VEL**2
IF (KERROR.EQ.0.2) WRITE (6,603) IPHASE,LEG,TEST
IF (KERROR.EQ.0.2) WRITE (6,603) IPHASE,LEG,TEST
603 FORMAT(1/17H IN LOITER, PHASE=I2, JH, LEG=I2, JH, DRAG = F6.3,
1 7H OF TN/28H CD CHANGED TO MATCH THRUST. /)
70 DELWF=SFC*TN*DELT/60.
30 W=W+DELWF
DELF+DELWF
W=W+DELWF
DEL+DELWF
IF (IPRT(IPHASE).EQ.0) GO TO 80
604 FORMAT(4X,9HLEG =I2,2X,7HSMNIN =E14.7,4X,3HSMN =E14.7,4X,3HSMNIN =E14.7,4X,3HSMN =E14.7,
1 1FHO*VCLxDRAGxTN,DELWF,F,F,DELxXLOIT,DELxXLOIT,DELxXLOIT,DELxXLOIT
IF (IT(IIPHASE).EQ.0) GO TO 80
1 2X,7HALPHA =E14.7,5X,4HCL =E14.7,4X,5HCOORD =E14.7,5X,4HCL =E14.7,5X,4HCL =E14.7,
2 5X,4HKP =E14.7,3X,6HROH1 =E14.7,3X,6HROH2 =E14.7,5X,4HSH =E14.7/14x
3 5X,6HCONT =E14.7,6X,3HETA =E14.7,4X,5HSHR =E14.7,6X,3HETA =E14.7,4X,5HVEL =E14.7,
4 3X,5HBDAG =E14.7,5X,4HTH =E14.7,2X,7HDELWF =E14.7,5X,4HSHF =E14.7,
5 6X,3HWM =E14.7,5X,6HDELX =E14.7,5X,6HDELX =E14.7,5X,6HDELX =E14.7,
6 5X,5HSM =E14.7,7X,8HGM =E14.7,7X,8HGM =E14.7,7X,8HGM =E14.7,
605 FORMAT(1/17H IN LOITER, PHASE=I2,7H LEG=I2,7H LEG=I2,7H)
1 9HLEG =I2,2X,7HSMNIN =E14.7,4X,3HSMN =E14.7,4X,3HSMNIN =E14.7,4X,3HSMN =E14.7,
1 1 2X,7HALPHA =E14.7,5X,4HCL =E14.7,4X,5HCOORD =E14.7,5X,4HCL =E14.7,5X,4HCL =E14.7,
1 2 5X,4HKP =E14.7,3X,6HROH1 =E14.7,3X,6HROH2 =E14.7,5X,4HSH =E14.7/14x
1 3 5X,6HCONT =E14.7,6X,3HETA =E14.7,4X,5HSHR =E14.7,6X,3HETA =E14.7,4X,5HVEL =E14.7,
1 4 3X,5HBDAG =E14.7,5X,4HTH =E14.7,2X,7HDELWF =E14.7,5X,4HSHF =E14.7,
1 5 6X,3HWM =E14.7,5X,6HDELX =E14.7,5X,6HDELX =E14.7,5X,6HDELX =E14.7,
1 6 5X,5HSM =E14.7,7X,8HGM =E14.7,7X,8HGM =E14.7,7X,8HGM =E14.7,
1 80 IF (W.GT.WPL) GO TO 90
.45 IF (KERROR.EQ.2) WRITE (6,605) IPHASE,LEG
150 CONTINUE
90 FORMAT(1/17H IN LOITER, PHASE=I2,7H LEG=I2,7H)
RETURN
155 IBS=0
155 C-----TEST FOR WEAPONS DROP
160 IF (IB(IPHASE).EQ.0) GO TO 110
155 IBS=0
155 C-----TEST FOR BOMBS DROP
160 IF (IB(IPHASE).EQ.0) GO TO 110
155 IBS=0
160 C TEST FOR ATMO DROP
110 IF (IA(IPHASE).EQ.0) GO TO 120
155 IBS=0
120 SMN(IPHASE)=SMN
165 HNT(IPHASE)=HN
165 CLT(IPHASE)=CL
165 ALPHAT(IPHASE)=ALPHA
165 WFT(IPHASE)=WF
165 TIME(IPHASE)=TIME
170 VEL(IPHASE)=VEL
170 SFCT(IPHASE)=SFC
52
SUBROUTINE LOITER 76/76 OPT=2 FTN 4.5+410

TNI(IPHASE)=TN
CDT(IPHASE)=CD
RLDT(IPHASE)=RLD
WEELT(IPHASE)=W
WART(IPHASE)=WTOT
QT(IPHASE)=Q
SFCUT(IPHASE)=SFCU
TNUT(IPHASE)=TNU
175

18C
CDINST(IPHASE)=CDINS
PRTOTT(IPHASE)=PRTOT
XT(IPHASE)=XLOIT
RETURN
END
SUBROUTINE DESCNT(ICALC, NERROR, IEGO, KGPRNT)
REAL KP
COMMON /TRAJCM/ ALPHA, ARW, CD, COL, CDD, CL, DESLF, DRAG, EN, HN, KP,
1 RLD, SMN, SMAX, RANGE, SFC, SWING, THRUST, TW, ULTLF, W, WAMMUN, WETANK,
2 WFUEL, WGTOW, WMISS, WTOT, WPLWT, WSTART(12), WSTOP(12), TIM(12),
3 PSIGN(12), TSTOP(12), NZST(12), PSIT(12), TSTOP(12), NSTIT(12),
4 KBOMBS, KFEXT, SFCU, THRSTU, CIDINS, PMTOT, WKFUEL, CRMAC, FLD, X(12),
5 FLAND, TENDUR, BLRANG, BLTIME, TIMT01, TIMT02, WFTD, PNAME1(12),
6 PNAME2(12), SMNT(12), HNT(12), WFT(12), TIMET(12), X(T(12)), CLT(12),
1 C
7 CD1(12), ALPHAT(12), CLIT(12), CDIT(12), ALPHIT(12), WAMUSED, WLAND,
8 OT(12), RLD1(12), SFC1(12), T1NT(12), C1ET(12), D1Y(18),
9 IAG, IPST, IMS, IBS, NPHASE, IDY(14)
COMMON /TRAJE/ ALP1G1(12), CD1N1ST(12), CD1G1ST(12), CL1GST(12), DEC1L,
1 DELH, DELP, DXCRUS, FLFAC, FRURE, HEND1(12), HENDUR, HMAX, HMINP,
2 HPATH(50), HEND, MENDUR, NT1(12), PRYT1(12), RADT(12),
3 RAPDST(12), SF, S2CTU(12), SMHAXP, SMHINP,
4 TUN1T(12), VIND(12), VEL1T(12), WPATH(50), WART(12), WCMBP, WEELT(12),
5 WRES, WFT1T, WFTO1, WFTO2, WFRAP, WLAND, WDESC, XGRLAN, FWMGAX, TOL,
6 WFIXED, WGCALC, WPL, GNTU, DUMY(82), IA(12), IAS, IGB(12), IREG, TENDUR,
2 C
7 IM1SS, IP1(12), IPHASE, IPT1, IPT1N, IPRT1(12), IPSIZE, IPST01, IPST02,
8 IW(12), IX(12), KERR1R, MILCOM, NCRUSE, NLEGCL, NLEGCR, NLEGLO, NMISS,
9 NPATH, DUMY(1)
DIMENSION ANS(4)
IF (IPRT(IPHASE).NE.0) WRITE (6, 600)
600 FORMAT(///2OH DESCNT PHASE///)
HNINT=HNT(IPHASE-1)
HNF1N=HEND(IPHASE)
IPS=5
VEL=VEL(IPHASE-1)
NLEGS=5
DN1H=(HNINT-HNF1N)/FLOAT(NLEGS)
HN=HNINT
TIME=0.0
WF=0.0
35 XDESC=0.0
IF (IPRT(IPHASE).EQ.0) GO TO 10
WRITE (6, 661) HNINT, HNF1N, CL, RLD, VEL, HN, DELT, TIME, XDESC, WF
661 FORMAT(2X/, 7E14.7, 2X, 7E14.7, 1X, 7E14.7, 5X, 1E14.7, 5X, 1E14.7, 5X, 1E14.7)
1 E14.7, 4X, SHRLD = E14.7, 4X, 5YV = E14.7, 5X, 4HHN = E14.7, 3X, 6HDELT =,
4C 2 E14.7, 3X, 6HTIME = E14.7, 2X, 7HXXDESC = E14.7, 5X, 4HWF = E14.7)
10 DO 30 L, EGS, ILEGS
CALL AT62(HN, ANS)
K=ANS(1)
A=ANS(4)
45 SMN=VEL/A
CALL TRAJ1(2)
CALL PROPI(ICALC, NERROR, IEGO, KGPRNT)
IF (NERROR.GE.2) RETURN
CALL TRAJ01(1)
5C TH=THRUST*EN
TN=THRSTU*EN
IAG=9
CALL TRAJ01(2)
CALL AERO(ICALC, NERROR, IEGO, KGPRNT)
55 IF (NERROR.GE.2) RETURN
CALL TRAJ01(1)
CL=SORT(CDU/KP)
54
SUBROUTINE DESCNT

SING = 2.*COS/CL
COSG = SORT(1., -SING)**2
VEL = SORT(2.*W*COSG/(CL*SWING*RHO))
DELT = -DELT/(VEL*SING)
TIME = TIME + DELT/60.
DELF = SFC*TN*DELT/3600.
WF = WF + DELW

DELX = VEL*DELT*COSG/6080.
XDESC = XDESC + DELX
HN = HN + DELH
WF = WF + DELWF

IF (IPRT(IPHASE), EQ, 0) GO TO 20
WRITE (6, 602) LEG, RHO, SMN, SFC, DELV, VEL, COSG, SING, DELT, TIME,
1 DELX, XDESC, HN, CL, CD, KP, DELWF, WF
602 FORMAT(/4X, 5HLEG - E12/4X, 5HRHO - E14.7, G1, 3HA - E14.7, 4X,
1 5HSMN = E14.7, 5X, 5HRHO = E14.7, 5X, 5HRHO = E14.7, 5X, 5HRHO = E14.7, 5X,
2 5HVEL = E14.7, 5X, 5HVEL = E14.7, 5X, 5HVEL = E14.7, 5X, 5HVEL = E14.7, 5X,
3 5X, 5HTIME = E14.7, 5X, 5HTIME = E14.7, 5X, 5HTIME = E14.7, 5X, 5HTIME = E14.7, 5X,
4 5HCO = E14.7, 5X, 5HCO = E14.7, 5X, 5HCO = E14.7, 5X, 5HCO = E14.7, 5X,
5 5HDELT = E14.7, 5X, 5HDELT = E14.7, 5X, 5HDELT = E14.7, 5X, 5HDELT = E14.7,
603 FORMAT(/4X, 5HLEG = I12/4X, 7HLEG = I12)
20 IF (W > WPL) GO TO 30
IF (KERRDR, EQ, 2) WRITE (6, 603) IPHASE, LEG
RETURN
30 CONTINUE
CALL AT62(HN, ANS)
A = ANS(4)
SMN = VEL/A
RHO = ANS(1)
Q = 5*RHO*VEL**2
BLTIME = BLTIME + TIME/60.
BLRANG = BLRANG + XDESC
90 C---- TEST FOR WEAPONS DROP
IF (IWP(IPHASE), EQ, 0) GO TO 40
W = W - WMISS
IWS = 0
95 C---- TEST FOR BOMBS DROP
4C 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
W = W - WAMMUN
IAS = 0
60 IF (IPRT(IPHASE), EQ, 0) GO TO 70
WRITE (6, 604) HN, A, SMN, RHO, Q, DXCRUS, W
604 FORMAT(/5X, 5HSMN = E14.7, 5X, 3HA = E14.7, 4X, 5HSMN = E14.7, 5X,
1 4X, 5HRHO = E14.7, 5X, 3HA = E14.7, 4X, 5HRHO = E14.7, 4X,
70 SMNT(IPHASE) = SMN
HNT(IPHASE) = HN
CLT(IPHASE) = CL
ALPHA(IPHASE) = ALPHA
WFT(IPHASE) = WF
TIME(IPHASE) = TIME
VEL(IPHASE) = VEL
SFTC(IPHASE) = SFC
TNT(IPHASE) = TN
SUBROUTINE DESCNT

115  CDT(IPHASE)=CD
     RLDT(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
SUBROUTINE PATHS (ICALC, ERROR, GEO, GPRNT)
REAL K
COMMON / TRAJCM / ALPHA, AR, CD, CDL, CDQ, CL, DESLF, DRAG, EN, HN, KP,
1 RLD, SMN, QMAX, RANGE, SFC, SWING, THRUST, TW, UTLF, W, WAMMUN, WETANK,
2 WFUEL, WGT, WMIS, WT, WPLT, MSTART(12), HSTART(12), TIM(12),
3 PS1G(12), TD0ST(12), N2ST(12), PST(12), TD0ST(12), N2ST(12),
4 WBOMBS, WEXT, SFCU, THRUST, CDINS, PRDT, WKFUEL, CRMACH, FLT0, X(12),
5 FLAND, TENDR, BLRANG, BLTIME, TMT01, TMT02, WFT0, PNAME(12),
6 PNAME(12), SNT(12), HNT(12), WFT1(12), TIMT(12), XT(12), CLT(12),
7 CDT(12), ALPHAT(12), CLT(12), CDT(12), ALPHAT(12), WFUZED, WLAND,
8 GT(12), RLDT(12), SFCT(12), TMT(12), CET(12), DYT(18),
9 IAD, IPS, ITS, SWS, IPS, PNAME, IDY(14),
COMMON / TRAJEX / ALPT(12), COINS(12), CQG1G(12), CLG1G(12), DECEL,
1 DELHP, DELMP, OXCRUS, FLFAC, FRFALL, EEND(12), HENDUR, HMAXP, HMINP,
2 HPATH(90), MEND(12), MNDFD, NT(12), PRDT(12), RADIT(12),
3 RADST(12), SFF, SFCUT(12), SHMAXP, SMMINP,
4 TNUT(12), VIND(12), VEL(12), VPATH(90), WART(12), WCOMP, WELT(12),
5 WRES, WFT, WFT0, WFT, WFT, WLAND, XDESC, XGRLAN, FWMAX, TOL,
6 WFIXED, WCALC, WPL, WGT, DUMY(82), IA(12), IAS, IB(12), IREG, IENDUR,
7 IMISS, IP(12), IPhase, IPT, IMPRINT, IPRT(12), IPSIZE, IPST01, IPST02,
8 IP(12), IX(12), KERROR, MilCom, MCRUSE, NLEGL, NLEGCL, NLEGL, NMISS,
9 NPATH, DUMY(6),
DIMENSION ANS(4)
IPS = IP(IPHASE)
TIME = IP(IPHASE)
25 XPATH = 0.0
WF = 0.0
IF (IPS(IFHASE) .NE. 3) WRITE (6, 600) IPS, TIME, XPATH, WF
FORMAT(11H PATH PHASE:1/4X SIPS - IZ, 15X TIME - E14.7, 3H 1,
3X XPATH = E14.7, 5X WF - E14.7, 6X, 3H = EE14.7/)
600 DC b0 I = 2, NPATH
LEG = I - 1
35 H1 = HPATH(I - 1)
H2 = HPATH(I)
35 HN = 5*(H1 + H2)
35 DILH = H2 - H1
35 CALL AT62I(HN, ANS)
35 RH0 = ANS(1)
A = ANS(4)
45 VM1 = VPATH(I - 1)
VM2 = VPATH(I)
VM = 5*(VM1 + VM2)
45 VM = VM1
C------ MAKE TEST TO SEE WHETHER VM IS MACH OR VELOCITY.
IF (VM .LT. 1.0) GO TO 10
45 S01 = VM1
S02 = VM2
S0 = VM
55 VEL1 = S01*A
VEL2 = S02*A
55 VEL = S0*A
GO TO 20
15 VEL1 = VM1
VEL2 = VM2
VEL = VM
55 S01 = VEL1/A
S02 = VEL2/A
S0 = VEL/A
ORIGINAL PAGE IS OF POOR QUALITY
SUBROUTINE PATHS

20 DELV=VEL2-VEL1
C------CALL PROPE TO GET TN, SFC.
   CALL TRAJ01(2)
   CALL PROPEICALC, NERROR, IGEO, KGPRNT)
   IF (NERROR.GE.2) RETURN
   CALL TRAJ01(1)
   TN=THrust*EN
   TNU=THRUST*EN
   C------CALL AERO TO GET CDO, KP.
   IAO=1
   ALPHA=5.0
   CALL TRAJ01(2)
   CALL AEROICALC, NERROR, IGEO, KGPRNT)
   IF (NERROR.GE.2) RETURN
   CALL TRAJ01(1)
   G=5.0RHO*VEL**2
   QS=G*SWING
   C1=QS/6
   C2=VEL*DELV/(32.2*DELH)+1.0
   C3=TN/6-C1*DOD
   C4=(C2/KP)**2
   C5=SQRT(0.25*C4**2—C4*C1*KP+C4/C1**2)
   C------CALCULATE FIRST APPROXIMATION TO CL.
   CL1=SQRT(-1.0+C4+C3/C1*KP+C5)
   C------CALL AERO TO GET CDO, ALPHA.
   CL=CL1
   IAO=1
   CALL TRAJ01(2)
   CALL AEROICALC, NERROR, IGEO, KGPRNT)
   IF (NERROR.GE.2) RETURN
   CALL TRAJ01(1)
   C6=((TN/6)*COS(ALPHA/57.3)-C1*DOD)/C2
   C------CALCULATE SECOND APPROXIMATION TO CL.
   CL2=SQRT(1.0+C6**2)/C1
   C------CALL AERO TO GET CDO, ALPHA.
   CL=CL2
   CALL TRAJ01(2)
   CALL AEROICALC, NERROR, IGEO, KGPRNT)
   IF (NERROR.GE.2) RETURN
   CALL TRAJ01(1)
   C------CALCULATE FLIGHT PATH ANGLE.
   SING=(TN/6)*COS(ALPHA/57.3)-C1*DOD)/C2
   IF (SING.GE.0.1 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, //////)
   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).LE.0.2) GO TO 40
   NERROR=2
   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
SUBROUTINE PATHS

* 15 40 DELT=DELH/(VEL*SING)
COSG=SORT(1-SING*2)
DELX=VEL*DELT*COSG/60.0*
DELF=TN*SFC*DELT/3600.0*
TIME=TIME+DELT/60.*
25 XPATH=XPATH+DELX
WF=WF+DELF
W=W-DELF
IF (IPRT/IPHASE).EQ.0) GO TO 50
WRITE (6,603) LEGs HIP H2sHNPDELHtSHN1pSMN29SMNsVEL12VEL2sVELsDELVs
IF (KERRR.EQ.2) WRITE (6,604) IPHASE9LEG
CONTINUE
HN-H2
SMN=SMN2
VEL=VEL2
BLTIME=BLTIME+TIME/60.*
BLRANG=BLRANG+XPATH
C-----TEST FOR WEAPONS DROP
IF (IW/IPHASE).EQ.0) GO TO 70
WM=WMISS
IWS=0
C-----TEST FOR BOMBS DROP
7C IF (IB/IPHASE).EQ.0) GO TO 80
W=W-BOMBS
IBS=0
C-----TEST FOR AMMO DROP
8C IF (IA/IPHASE).EQ.0) GO TO 90
W=W-AMMUN
IAS=0
9C IF (IPRT/IPHASE).EQ.0) GO TO 100
WRITE (6,605) HW,H2,SMN,VEL
100 SPNT/IPHASE=SMN
HN=IPHASEsHN
CLT/IPHASE=CL
ALPHA/IPHASE=ALPHA
WF/IPHASE=WF
TME=4/IPHASE=TIME
VELT/IPHASE=VEL
SFC/IPHASE=SFC
RETURN
603 FORMAT(/1Xs5HLEG -I215Xp4HH1 -E14.7s5Xp4HH2 -E14.7s5Xp4HH3s5Xp4HH4s5Xp4HH5,5Xp4HMN =
2 E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =
3 E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =
4 E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =
5 E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =
6 E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =
7 E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =
8 E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =
9 E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =
10 E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =E14.7s3Xp6HVEI =
SUBROUTINE PATHS  7b/7b  OPT=2

175
TNT(IPHASE)=TN
CDT(IPHASE)=CD
RLDT(IPHASE)=KLD
HEEET(IPHASE)=W
WART(IPHASE)=WTDT
QT(IPHASE)=Q
SFCUT(IPHASE)=SFCU
THUT(IPHASE)=THU

180
CDINST(IPHASE)=CDINSP
PRTTTT(IPHASE)=PRTDT
XT(IPHASE)=XPATH
RETURN
END
SUBROUTINE TRAJO 

C OUTPUT OF RESULTS FROM TRAJECTORY ROUTINE 

REAL NZST,NZIT,NZL0,MENDUR,LOIT 

COMMON /TRAJCM/ ALPHA,AWC,CD0,CDL,CDU,CDL,DESFL,DRAG,EN,HN,KN 

1 RD,SMJMAX,RANGE,SPC,SWING,THRUST,T,WUFL,WMAMUN,WETANK, 

2 WFUEL,WTOT,IMP,MTOT,MLSTART(12),HSTART(12),TIM1(12), 

3 PSIGT(12),TDOTT(12),NZST(12),PSIT(12),TDOTT(12),NZL0(12), 

4 WGBOMBS,WFEXT,SFU,THRUST,CDINSP,PRTO,I,WFUEL,CRMACH,FLTD,X(12), 

5 FLAND,TENDUR,BLANG,BLTIME,TTMT1,TTMT2,WFOTO,PNAME(12), 

6 PNAME(12),SMNT(12),HLT(12),WTFT(12),TIMET(12),XT(12),CLT(12), 

7 CD1(12),ALPHAT(12),CLIT(12),CD1IT(12),HPW,MPWSED,WLAND, 

8 QT(12),RLDT(12),SFCT(12),NT(12),CT(12),DY(12), 

9 IAO,IPS,ITS,ISW,IPS,YN,PHASE,IDX(12), 

COMMON /TRAJE/ ALPHAT(12),CD1NST(12),CD1GT(12),CL1GT(12),DEC00, 

1 DELH,BFELP,DFC,HRFAC,FRFUSE,HEND(12),HENDUR,HMAX,HPMNP, 

1 S HPATH(50),NEND(12),MENDUR,NT(12),PRTOS(12),RADIT(12), 

3 RAD1(12),SF,SMCUT(12),SMHAXP,SMHNP, 

4 TNU1(12),VIND(12),VELT(12),VPATH(50),WART(12),WCOMB,WEELT(12), 

5 WRES,WFOTO,WFOTO,WFTRAP,WKLAND,XDEC,XRGLAN,FWGMAX,TOL, 

6 WFIXED,GCALC,WPL,WGTO,DUMY(82),IA(12),IAS,IB(12),IBREG,ITENDUR, 

7 IMISS,IP(12),IPHAS,IPLOT,IPRINT,IPRT(12),IPSIZE,IPST01,IPST02, 

8 IPW(12),IPX(12),IPERROR,IMCGM,INCRUSE,NLEGCL,NLEGCO,NMISS, 

9 NPATH,DUMY(4), 

DATA C06B/14HCOMB/, 

DATA LOIT14HL0IT/ 

WRITE (6) 601 CC,, 

601 FORMAT(1H1,24X,17HTRAJECTORY OUTPUT// 

1 24X,7HMISSION=I2,12H (PAYLOAD =F8.3,4H L81// 

2 67H PHASE M H CL ALPHA WFUEL TIME 

3 VEL/9X 57SHSC(1) THRUST(1) CD L/D W WA 

4 SQX 57SHSC(1) THRUST(1) CDINST PR 

5X) 

do 10 IPHASE=1,NPHASE 

1 WRITE (6,601) PNAME(1PHASE),PNAME2(1PHASE),SMNT(1PHASE), 

2 MNTH(1PHASE),CLT(1PHASE),ALPHAT(1PHASE),WF1(1PHASE),TIMI(1PHASE), 

3 VELT(1PHASE),SFC(1PHASE),TNT(1PHASE),CDT(1PHASE),RELT(1PHASE), 

4 WEL(1PHASE),WART(1PHASE),GTH(1PHASE),CSF(1PHASE),CSN(1PHASE), 

4 CDINST(1PHASE),PRTO(1PHASE),XTT(1PHASE), 


2 F18.2,F18.2,F18.2,F18.2,F18.0/ 

1 CONTINUE 

WRITE (6,602) WFOTO,WFOTO,WFOTO,WFOTO,WFUSED,WFUEL,WRES,WFTRAP 

1 FORMAT(1H1,20X,12H FUEL SUMMARY//14H TAKEOFF FUEL// 

5 9X 7HWFT01 =F8.0,5X 12H TOTAL FUEL =F8.0/ 

2 9X 7HWFT02 =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 

1 ZS=1.0 

2 TDOT1G=0.0 

3 RADITG=0.0 

4 PSS=0.0 

DO 30 IPHASE=1,NPHASE 

30 IF (PNAME(IPHASE).EQ.00M8) GO TO 30 

IF (J.NE.0) GO TO 20 

WRITE (6,603)
SUBROUTINE TRAJ

603 FORMAT(//19X,28HADDITIONAL COMBAT PARAMETERS//13X,
1 58HCONDITIONS PS NZ TDOT RADIUS ALPHA CL CD)
J=1

60 WRITE (6,604) SMNT(IPHASE),PSNGT(IPHASE),PS1G(IPHASE)
1 ALP1G(IPHASE),C11G(IPHASE),CD1G(IPHASE),HNT(IPHASE),PS1S,
2 NZST(IPHASE),TD0ST(IPHASE),RADST(IPHASE),ALPH1T(IPHASE),
3 CLT(IPHASE),CDT(IPHASE),PS1T(IPHASE),NZIT(IPHASE),TD0IT(IPHASE),
4 RADIT(IPHASE),ALPHIT(IPHASE),CLIT(IPHASE),CDIT(IPHASE),
5 C1I1T(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,
3 13X,15HCOMBAT ENERGY.*E13.6)
30 CONTINUE

WRITE (6,605) BLTIME,BLTIME,FLTD,FLTD,FLRNL,LAND,LAND,
1 MENDUR,HENDUR, TENDUR

605 FORMAT(/12X,23H BLOCK TIME *F7.3,7H HOURS/
1 35H BLOCK RANGE *F7.1,7H N. M. /
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)

83 DO 40 IPHASE =1,NPHASE
IF (PNAN1(IPHASE).NE.LG1T) GO TO 40
WRITE (6,606) X(IPHASE)

606 FORMAT(20X,15HLA1TERR RADIUS *=F6.2,8H N. M.)
40 CONTINUE
90 RETURN
END
SUBROUTINE TRPLOT 76/76 OPT=2 FTN 4.5+410

SUBROUTINE TRPLOT(NERROR,NGET,KPRINT)

REAL LIFTYNZSPNZI
COMMON /TRAJCM/ ALPHA,AR,W,CDL,CDO,CL,DESL,DRAG,EN,HN,KP,
1 RLD,SMN,MAX, RANGE,SFC,SWING,THRUST,TH,ULTF,W,WMAM,WETANK,
2 WFEU,WGTDTW,WMISS,WTOT,WPLOT,WSTART,WSTART,WTIM,WTIM,
3 PSI1T12,TDOTT12,NZT12,PSIT12,TDOTT12,NZIT12,
4 WBOENDS,WFEAT,SCUR,THRST,CDINS,PRTO,WKFUEL,CRMACH,FTL0,XT12,
5 FLAND,TENDUR,BLANGE,BLTIME,TIMTO,TIMTO,WFTO,PNAME12,
6 PNAME212,SMNT12,HNT12,WFT12,TIMET12,XT12,C12,
7 CDT12,ALPHAT12,CLIT12,CD12,ALPHIT12,WFUSED,WLAND,
8 Q12,RLDT12,SFC12,TNT12,CE12,DI12,
9 IAAG,IPS,ITS,ISWS,ISB,NPHASE,DUY114

COMMON /TRAJEX,ALP12,C12,C12,C12,C12,C12,
1 DEL,W,DEL,DCRUS,FLAC,FRFUR,END12,HEND12,HMAX12,HMIN12,
2 DFNHNT12,END12,HEND12,NT12,PRTDU,T12,RAD12,
3 RADS12,SCUT12,SMMAX12,SMHNP12,
4 TNUT12,VIND12,VELT12,VTAT12,WAR12,WCOMP,WELT12,
5 WFRE5,WFTO12,WFTO2,WFTO3,WFTRAP,WKLAN,XTESC,XRGLAN,FMGMAX,TOD12,
6 WFIXED,WCALC,WPWT,GDT,NDUM912,IA12,IAS,IB12,IBREG,IEND12,
7 IMI,IP12,IPHASE,IPLOT,IPRINT,IPRT112,IPST12,IPST22,
8 IT12,IX12,KERR12,HC12,HC12,KERROR,PICAL,PUR12,
9 IPST12,IPST2,IPST2,IPST2,IPST2,IPST2,IPST2,
0 IAAG,IPS,ITS,ISWS,ISB,NPHASE,DUY114

DATA COMB/HCOMB/

C-----IPLOT = 0, NO PRINT, NO PLOT
C-----IPLOT = 1, PRINT, NO PLOT
C-----IPLOT = 2, NO PRINT, PLOT
C-----IPLOT = 3, PRINT, PLOT
C-----IPLOT = 4, PLOT/PRT12
C-----IPLOT = 5, PRINT, PLOT
3C WCOMP = LE, 0 = W FROM LAST COMBAT PHASE
C-----WCOMP BETWEEN 0.0 AND 1.0, W = WCOMP * WGT0
C-----WCOMP, GT, 1 = W = WCOMP
C-----CALCULATE COMBAT PARAMETERS.

J=0
DO 10 IPHASE=1,NPHASE

I=0
DO 35 IPHASE=(1,NPHASE)

J=0
DO 10 IPHASE=1,NPHASE

30 CONTINUE

10 CONTINUE

IF (J.EQ.0) IPS=2
IF (J.GT.0) IPS=J+1
J=I

50 CONTINUE

IF (JM1.EQ.0) JM1=1
IF ((WCOMP .LE. 0.0) .AND. (J.EQ.0)) W=7*WGT0
IF ((WCOMP .LE. 0.0) .AND. (J.GT.0)) W=WELT12
IF (WCOMP .GT. J) W=WCOMP

45 IF (WCOMP .LE. 1.0) W=WCOMP

40 IF ((IPL0.EQ.0) .OR. (IPL0.EQ.3)) WRITE (6,600) HMNP,HMAX12,

5C 60C FORMAT(H11,4H9H,HMNP=7.0/I9H,HMNP=F7.0/I9H,HMAX12=F7.0/I9H,

10H SMNIP=F6.2/I10H SMINP=F6.2/I10H SMAX12=F6.2/I10H SMAX12=F6.2/I10H,

IOH WCOMP=F6.2/I10H WCOMP=F6.2/I10H IPL0=F12/I10H QMAX=F7.0/I10H,

30H H=FLG.1/I9H H=FLG.1/I9H H=FLG.1/I9H H=FLG.1/I9H H=FLG.1/I9H,

50H TDT:4

NIZI, PSI, TOTI, 0)

5C 601 IF (IPL0.GE.2) WRITE (42,601) NH,NM,DELHP,DELMP,QMAX

601 FORMAT(18,3F8.2)

DO 40 IH=1,NH

9C 40 CONTINUE

62C 10 CONTINUE

63
SUBROUTINE TRPLOT

60
Hn=FLOAT(HN-1)*DELHP
CALL ATO2(HN,ANS)
RHO=ANS(1)
A=ANS(4)
ZU=4G*IN*1, NM
SMN=FLOAT(IN-1)*DELMP+SMMP
VEL=SMN+A
Q=5*RHO*VEL**2
QS=Q*S*WING
CL=W/OS
IAO=8
CALL TRAJOI(2)
CALL AERO(2,NERROR,IGEO,KGPRINT)
IF (NERROR.GE.2) RETURN
CALL TRAJOI(1)
DRAG=CD*QS
CALL TRAJOI(2)
CALL PROP(2,NERROR,IGEO,KGPRINT)
IF (NERROR.GE.2) RETURN
CALL TRAJOI(1)
TN=THUST*EN
PS1G=(TN-DRAG)*VEL/W
N2S=1.001
IF (PS1G.LE.0.0) GO TO 20
IAO=4
CALL TRAJOI(2)
CALL AERO(2,NERROR,IGEO,KGPRINT)
IF (NERROR.GE.2) RETURN
CALL TRAJOI(1)
LIFT=CL*QS
N2S=(LIFT+IN*SIN(ALPHA/57.3))/W
IF (N2S.LE.DESELF) GO TO 20
IAO=6
CALL TRAJOI(2)
CALL AERO(2,NERROR,IGEO,KGPRINT)
IF (NERROR.GE.2) RETURN
CALL TRAJOI(1)
N2S=DESELF
IF (N2S.LE.1.0) N2S=1.001
PHI=ARCOS(1.0/N2S)
RADIUS=VEL**2/(132.2*TAN(PHI))
TDOTS=57.3*VEL/RADIUS
IAO=5
CALL TRAJOI(2)
CALL AERO(2,NERROR,IGEO,KGPRINT)
IF (NERROR.GE.2) RETURN
CALL TRAJOI(1)
LIFT=CL*QS
N2I=(LIFT+IN*SIN(ALPHA/57.3))/W
IF (N2I.LE.DESELF) GO TO 30
IAO=6
CALL TRAJOI(2)
CALL AERO(2,NERROR,IGEO,KGPRINT)
IF (NERROR.GE.2) RETURN
CALL TRAJOI(1)
N2I=DESELF
IF (N2I.LE.1.0) N2I=1.001

SUBROUTINE TRPLOT 76/76 OPT=2  FTN 4.5+410 05

15   DRAG=CD*OS
   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 ((IPLAT.EQ.1).OR.(IPLAT.EQ.3)) WRITE (6,602) HN,SMN,
     1 PSI,NZ,NZI,PSI,TDOTi,Q
   IF (IPLAT. GE.2) WRITE (42,602) HN,SMN,PSI,NZ,NZI,PSI,
     1 TDOTi,Q

602   FORMAT(7F0,6F2,1P7E11.3)

25   CONTINUE

30   C-----LANDING FIELD LENGTHS AT ALTITUDES OF 0, 4000, 8000, AND 12000 FT.
     IF ((IPLAT.EQ.1).OR.(IPLAT.EQ.3)) WRITE (6,603)

603   FORMAT(1'H1s8X#33HLANDING FIELD LENGTH CALCULATIONS//
     1 'ALT'  W.t1lN. _. 	 FLLAN(WKIN) 	 WGT
    1 FLLAN(WKIN)**2//)
     WHIN=WGT-WFTOT-WPL
     ID=12
     DO 50 I=1,4
     HN=(I-1)*4000
     CALL AT62(HN,ANS)
     RH0=ANS(1)
     CALL TRAJ01(2)
     CALL AERD(2,NERD,R,IGEO,KGPRNT)
     IF (NERD.RG.E.2) RETURN
     CALL TRAJ01(1)
     VSTALL=SQR(2.*WMIN/(SWING*CL*RHO))
     VSTO=n=1.15*VSTALL
     VTDOWN=1.15*VSTALL
     XAIR=((VSTO-n**2-VTDOWN**2)/(64.4*50.))*RLD
     XGRLAN=VTDOWN**2/(64.4*DECEL).
     FLLAN1=(XAIR+XGRLAN)/FLFAC
     VSTALL=SQR(2.*WGT0/(SWING*CL*RHO))
     VScren=1.3*VSTALL
     VTDOWN=1.15*VSTALL
     XAIR=((VSTO-n**2-VTDOWN**2)/(64.4*50.))*RLD
     XGRLAN=VTDOWN**2/(64.4*DECEL).
     FLLAN2=(XAIR+XGRLAN)/FLFAC
     IF ((IPLAT.EQ.1).OR.(IPLAT.EQ.3)) WRITE (6,604) HN,WMIN,FLLAN1,
     1 WGT0,FLLAN2

604   FORMAT(7F0,4EL5.6)
     IF (IPLAT. GE.2) WRITE (42,604) HN,WMIN,FLLAN1,WGT0,FLLAN2

50   CONTINUE
     RETURN
END
SUBROUTINE GOLDEN

C-----THIS ROUTINE USES THE GOLDEN SECTION METHOD TO FIND
C THE X CORRESPONDING TO THE MAXIMUM Y IN THE INTERVAL
C BETWEEN XM/N AND XMAX.

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/ ALPHAS, SARW, CDL, CD, CL, DESL, DRAG, EN, HN, KP,
      RLD, SMN, QMAX, RANGE, SFC, SWING, THRUST, TV, ULTLF, WAMM, W TANK,
      WFUEL, WGTOT, WMISS, WPTOT, WPLT, HSART(12), HSTART(12), TIM(12),
      PS1GT(12), TDOTST(12), NZST(12), PSIT(12), TDOTIT(12), NZIT(12),
      WBOMHS, WFEXT, SFCU, THRUST, CODISP, PRTOT, WKFUEL, CRMACH, FLT0, X(12),
      FLAND, TENDUR, BLANG, BTIME, TIMT01, TIMT02, WFT0, PNAME1(12),
      PNAME2(12), SNNT(12), HNT(12), WFT(12), TIMET(12), XT(12), CLT(12),
      CDT(12), ALPHAT(12), CLIT(12), CUIT(12), ALPHIT(12), WFUSED, NLAND,
      QL(12), RLOT(12), SFC(12), TNT(12), CET(12), DY(18),
      IA0, IPS, IT5, IMS, IBS, NP, NASE, IDY(14),

COMMON /TRAJEX/ ALPIGT(12), CDINST(12), CDIGT(12), CLIGT(12), DECEL,
      HPATH(50), HEND(12), HENDUR, NT(12), PRTOTT(12), RADIT(12),
      KADST(12), SF, SFCUT(12), SMMAX, SMMIN,
      TNUT(12), VIND(12), VELT(12), VPATHHT(50), WSART(12), WCOMBP, WELT(12),
      WRES, WFTOT, WFT01, WFT02, WFTAP, WKLAND, XDESC, XGRANL, XFGMAX, TOL,
      W FIXED, WGCALC, WPL, WGT0, DUHY(82), IA(12), IAS, IB(12), IBREG, IENDUR,
      IMISS, IP(12), IPHASE, IPL, IPRINT, IPT(12), IPRINT, IPT(12), IPSIZE, IPSTO1, IPSTO2,
      IP(12), IX(12), IPERROR, IMILC, NCURSE, NLEGCL, NLEGCR, NLEGLO, NMIS,
      NPATH, IDUMY(6),
      DIMENSION ANS(4)

IPSAVE = IPS
KCALC = 2
IA0 = 0
IPS = 4
IF ((IPRINT.GE.2).AND.(!IXY.EQ.1)) WRITE (6, 600)
IF ((IPRINT.GE.2).AND.(!IXY.EQ.2)) WRITE (6, 600)
IF ((IPRINT.GE.2).AND.(!IXY.EQ.3)) WRITE (6, 600)

600 FORMAT(12H SEARCH FOR BREG, ALT./23H ALT. BREG. FACT.,
1 5X, 4H MACH, 8X, 2H CL, 8X, 2H CD, 7X, 6H THRUST, 3X, 3H SFC)
601 FORMAT(12H SEARCH FOR ENDUR. ALT./23H ALT. ENDUR. FACT.,
1 5X, 4H MACH, 8X, 2H CL, 8X, 2H CD, 7X, 6H THRUST, 3X, 3H SFC)
602 FORMAT(12H SEARCH FOR ENDUR. MACH/23H MACH ENDUR. FACT.,
1 5X, 4H MACH, 8X, 2H CL, 8X, 2H CD, 7X, 6H THRUST, 3X, 3H SFC)
X1 = XM/N
X2 = XMAX
XX = 1
N = 0
XBLST = XX
YBEST = -1.
IF (IXY.LE.2) HN = XX
IF (IXY = 4) SMH = XX
IRETN = 1

C-----START OF FUNCTION EVALUATION

CALL AT62(HN, ANS)
RHO = ANS(1)
A = ANS(4)

66
SUBROUTINE GOLLEN 76/76 OPT=2  
FTN 4.5+410 0!

VEL*SMN*A
CL=2.*W/(RHO*SWING*VEL**2)
60 CALL TRAJ01(2)
CALL AERO(KCALC, NERROR, IGE0, KGPRNT)
IF (NERROR.GE.2) RETURN
CALL TRAJ01(1)
DPAG6.5*RHO*SWING*CD*VEL**2
65 CALL TRAJ01(2)
CALL PROP(KCALC, NERR0R, IGE0, KGPRNT)
IF (NERROR.GE.2) RETURN
CALL TRAJ01(1)
IF (IXY.EQ.1) Y=3600.*VEL*RLD/SFC
70 IF (IXY.GE.2) Y=RLD/SFC
IF (NERROR.EQ.1) Y=-1.0
CALL TRAJ01(2)
CALL PRJP(KCALC, NERROR, IGE0, KGPRNT)
IF (NERROR.EQ.2) RETURN
CALL TRAJ01(1)
IF (IXY.EQ.1) Y=3600.*VEL*RLD/SFC
75 IF (IXY.EQ.3) Y=SMN*CL*CD*THRACT/SFC
IF (Y.GT.YBEST) XBEST=XX
SUBROUTINE GOLDEN 76/76 OPT=2  
FTN 4.5+410 0!

VEL*SMN*A
CL=2.*W/(RHO*SWING*VEL**2)
60 CALL TRAJ01(2)
CALL AERO(KCALC, NERROR, IGE0, KGPRNT)
IF (NERROR.GE.2) RETURN
CALL TRAJ01(1)
DPAG6.5*RHO*SWING*CD*VEL**2
65 CALL TRAJ01(2)
CALL PROP(KCALC, NERR0R, IGE0, KGPRNT)
IF (NERROR.GE.2) RETURN
CALL TRAJ01(1)
IF (IXY.EQ.1) Y=3600.*VEL*RLD/SFC
70 IF (IXY.GE.2) Y=RLD/SFC
IF (NERROR.EQ.1) Y=-1.0
CALL TRAJ01(2)
CALL PRJP(KCALC, NERROR, IGE0, KGPRNT)
IF (NERROR.EQ.2) RETURN
CALL TRAJ01(1)
IF (IXY.EQ.1) Y=3600.*VEL*RLD/SFC
75 IF (IXY.EQ.3) Y=SMN*CL*CD*THRACT/SFC
IF (Y.GT.YBEST) XBEST=XX
ORIGINAL PAGE IS OF POOR QUALITY

GO TO (20, 30, 40, 60), IRETN
80 20 Y1=Y
XX2
IF (IXY.LE.2) MnXX
IF (IXY.EQ.3) SMNXX
IXY.EQ.3

GO TO 10
85 3C Y2=Y
3C TAU=(SQRT(5.))-1.
XX3=TAU*(X2-X1)

GO TO 10
90 4C Y3=Y
50 X12=.5*(X1+X2)
DELX=ABS(X12-XX3)
XX4=X12+DELX
IF (X3.GT.X12) X4=X12-DELX

GO TO 10
.0C Y4=Y
6C 05 IF (Y3.GE.Y4) GO TO 90
IF (X4.LT.X3) GO TO 70
X1=X3
Y1=Y3
GO TO 80

1C 7C X2=X3
Y2=Y3
8C X3=X4
Y3=Y4
GO TO 110
SUBROUTINE GELDEN 76/76 UPT=2 FTN 4.5+410 0

115  9c  IF (X3.LT.X4) GO TO 100
     X1=X4  
     Y1=Y4
     GO TO 110

120  Y2=Y4

110  IF (N.LE.NMAX) GO TO 50
     X2=X3
     Y=YBEST
     IF (Y.EQ.-1.0) NERROR=2
     IF (NERROR.EQ.2) WRITE (6,604) XX, Y, IXY
     FORMAT(35H **FATAL ERROR IN GOLDEN ROUTINE. /
            1 5X,3HX = E12.5,2X,3HY = E12.5,2X,5HIXY = I9)
     IF ((IPRINT.GE.2) OR (KERROR.EQ.2)) WRITE (6,605) XX, Y
     FORMAT(35H MACH OR ALTITUDE SEARCH CONCLUDED. /
            1 5X,3HX = E12.5,2X,3HY = E12.5)
     IPS=IPSAVE
     RETURN
     END
REFERENCES


