NASA TECHNICAL MEMORANDUM

NASA TM X-62,333

NASA TM X-62,333

COMPUTER PROGRAMS FOR ESTIMATING AIRCRAFT TAKEOFF AND LANDING PERFORMANCE

Jeff V. Bowles and Thomas L. Galloway

Ames Research Center
Moffett Field, Calif. 94035

July 1973

ABSTRACT

A set of computer programs has been developed to estimate the takeoff and landing maneuver of a given aircraft. The program is applicable to conventional, vectored lift and powered-lift concept aircraft. Portions of the program may also be used to evaluate the static performance of these types of aircraft. The aircraft is treated as a point mass confined to motion in a vertical plane, and rotational dynamics have been neglected. The required input is described and a sample case is presented.

TABLE OF CONTENTS

COMPUTER PROGRAMS FOR ESTIMATING AIRCRAFT TAKEOFF AND LANDING PERFORMANCE

By Jeff V. Bowles and Thomas L. Galloway

Ames Research Center, Moffett Field, Calif. 94035

INTRODUCTION

A set of computer programs has been developed to estimate the takeoff and landing maneuver of a given aircraft. The program is applicable to conventional, vectored lift and powered-lift concept aircraft. Portions of the program may also be used to evaluate the static performance of these type aircraft. The aircraft is treated as a point mass confined to motion in a vertical plane, and the rotational dynamics have been neglected. The takeoff subroutines are a modification and simplification of an unpublished program by V. R. Corsiglia of Ames Research Center.

The user is required to provide two subroutines which compute the total force coefficients along and normal to the flight path, and determine various required engine characteristics.

This report describes the various subroutines and the required input, the equations used, and the computational techniques involved. Also included is a listing of the total program and an example case.

NOTATION

Symbol	Fortran Name		
a_n	S(9)	-	acceleration normal to flight path (rad/sec^2)
a_t	T(6), S(8)	-	acceleration along flight path (ft/sec^2)
C_x	CX	-	force coefficient along flight path
Cy	CY	-	force coefficient normal to flight path
ENP	ENP	-	number of engines
g		-	acceleration due to gravity (ft/sec^2)
h	HABS, S(7)	-	absolute altitude (ft)
i_w	EYEW, EYEWNG	-	incidence of wing (deg)
LF	XLF	-	load factor
q		-	dynamic pressure (lb/ft^2)
R/C	S(11), ROC, RTCL	-	rate of climb (ft/sec,fpm)
S	SW, SWING	-	reference wing area (ft^2)
T	THRUST	-	thrust, net or gross, per engine (lb)
V	T(4), S(4), VEL	-	aircraft velocity (ft/sec)
V_x	S(10)	-	horizontal component of V (ft/sec)
V_y	S(11)	-	vertical component of V (ft/sec)
V_1	V1	-	engine failure speed, EAS (knots)
V_R	VR	-	rotation speed, EAS (knots)
W	W, WG	-	aircraft weight (lb)
W_f	WF, WFUEL	-	fuel flow (lb/hr)
α	ALPHA	-	angle of attack (deg)
γ	S(5), GAMMA	-	flight path angle (rad,deg)
δ_F	DELFD	-	flap deflection (deg)
δ_s	DELSPL	-	spoiler deflection (deg)

θ	THETAF	-	pitch attitude fuselage angle (deg)
ν	ANGLE	-	vectored thrust deflection angle (deg)
ρ	RHO	-	air density (slugs/ft^3)
μ	MU	-	rolling coefficient of friction
μ_{BRK}	XMUBRK	-	braking coefficient of friction

SUBROUTINES

Subroutine TAKOFF

The subroutine TAKOFF simulates the takeoff maneuver of a given aircraft. The program is applicable to conventional, vectored lift and powered lift aircraft. The aircraft is treated as a point mass confined in motion to a vertical plane and the rotational dynamics are neglected. This simplification necessitates an estimation of all rotational rates involved. These rates are either input by the user or are approximated by a finite difference form.

The takeoff maneuver is divided into four basic segments: ground roll and rotation, liftoff and initial segment climb, acceleration to final climb speed at a constant rate of climb, and finally, the pullup maneuver to establish the final climb speed. Provisions in the program are made for gear retraction, flap retraction, changing of vectored thrust angle and changes in the power setting.

The ground roll is made at a specified power setting and flap deflection. When the rotation speed is reached, the aircraft is "rotated" by increasing the angle of attack linearly with time until liftoff occurs or the tail scrape angle is reached. If the latter occurs, the ground roll is continued with the fuselage angle equal to the tail scrape angle.

The flight path control is obtained by monitoring three dynamic variables — acceleration along flight path, load factor and fuselage angle (pitch attitude). The aircraft is not permitted to decelerate and the load factor and fuselage angle are restricted to values less than or equal to a specified value. If any of these conditions are violated, the angle of attack is reduced until all constraints are satisfied.

Once a specified altitude is attained, called the maneuvering altitude, the aircraft is pitched down by a reduction in angle of attack until a specified rate of climb is obtained. The aircraft then accelerates at this rate of climb until the desired final climb speed is reached.

When the final climb speed is attained, the pullup maneuver is executed in order to bring the aircraft to a zero rate of acceleration along the flight path. This maneuver is accomplished by increasing the angle of attack and pulling a load factor of 1.20, which will result in an increase in the rate of climb to a final value at the desired climb speed. It may be necessary to throttle the engines in order to maintain the desired constant climb speed subject to the fuselage angle restriction.

Program Inputs

The inputs to subroutine TAKOFF are through the argument list, input by NAMELIST and common blocks /UNIV/ and /AERO/.

The call to TAKOFF is as follows: CALL TAKOFF (INPC, IDCN, WGROSS, SWING, XENG, V1, VR, VEND) where

INPC - program control = 1 - input data loaded
= 2 - program executed
= 3 - data input and program executed

IDCN - print control = 1 - no print out
= 9 - print out

WGROSS - aircraft gross weight (lbs)

SWING - reference wing area (sq ft)

XENG - number of engines

V1 - critical engine failure speed (knots)

VR - rotation speed (knots)

VEND - final climb speed (knots)

All speeds are indicated air speeds.

There are two namelist inputs to TAKOFF, /NAM1/ and /NAM2/.

The namelist /NAM1/ input variables are as follows:

CDGEAR - drag increment due to gear

DFLPDT - flap retraction rate (deg/sec)

DTABS - temperature increment above 59°F (°F)

DTGR - time required to retract gear (sec)

DTPDWN - throttling down rate (percent/sec)

DTPUP - throttling up rate (percent/sec)

DTVECT - vectored thrust angle reduction rate (deg/sec)

EYEWNG - wing incidence angle (deg)

HAPT - airport altitude (ft)

HGR - altitude at which gear retraction is started (ft)

HMAN - maneuvering altitude (ft)

HMAX - takeoff termination altitude (ft)

IOUT - for engine out takeoff, set IOUT = 1

UM - rolling coefficient of friction

NPAGE - no. of lines printed per page

PMARG - pullup speed margin

RTCL - rate of climb during accelerate segment (fpm)

THTFLY - maximum allowable fuselage angle while airborne (deg)

THTSCP - tail scrape angle (deg)

XLFMAX - maximum allowable load factor

DTFAIL - time required for one engine to fail (sec)

The user may input all, some, or none of the above input variables. The default values of these input variables are listed below:

CDGEAR = 0.0, DFLPDT = 3.0 deg/sec, DTABS = 0.0°F,

DTGR = 5.0 sec, DTPDWN = 5.0 percent/sec,

DTPUP = 6.0 percent/sec, DTVECT = 10. deg/sec,

EYEWNG = 1.0 deg, HAPT = 0.0 ft, HGR = 25.0 ft,

HMAN = 1000. ft, IOUT = 0, UM = 0.02, PMARG = 0.04,

RTCL = 750. fpm, THTFLY = 15.0 deg,

THTSCP = 10. deg, XLFMAX = 1.15, DTFAIL = 3.0 sec

If the default value of CDGEAR is used, the program will calculate, based on an empirical formula, a value for the gear drag as a function of gross weight and wing area.

The second set of namelist variables, /NAM2/, constitute the flap, throttle and vectored thrust schedules. These are tables that manage the flap setting, power setting and vectored thrust angle as a function of the aircraft speed and altitude. These variables are arrays of dimension 5.

XDELFD(I) - flap deflection (deg)

XHFLAP(I) - flap retraction altitude (ft)

XVFLAP(I) - flap retraction speed (knots)

XPOWER(I) - power setting

XHPWR(I) - power setting change altitude (ft)

XVPWR(I) - power setting change speed (knots)

XNU(I) - vectored thrust angle (deg)

XHVECT(I)- vectored thrust angle change altitude (ft)

XVVECT(I)- vectored thrust angle change speed (knots)

All altitudes are absolute altitudes and all speeds are indicated air speeds.

These schedules are constructed as follows: If the speed or altitude of the aircraft is equal to, say, XVFLAP(I) or XHFLAP(I), respectively, then the flaps are retracted to the value XDELFD(I). The power setting and vectored thrust angle management work in a similar manner. The power setting may either be increased or decreased. The flap setting and vectored thrust angle setting can only be reduced with speed and altitude. The values of XDELFD(1), XNU(1) and XPOWER(1) are all for the ground run. The user is permitted four changes in flap, power, vectored thrust angle settings during the airborne portion of the takeoff.

The default values for /NAM2/ are as follows:

- 100. percent throttle throughout takeoff
- 0. degrees vectored thrust
- 15.0 degrees flaps during ground roll, retracted to 5.0 degrees at 250 ft altitude, retraction to 2.0 degrees at 200 knots, and finally, complete retraction at 210 knots

Again, the user may choose to use all, some, or none of the above schedule values. No changes to any of these settings are allowed during the pullup maneuver.

Program Output

The program output consists of a time history of several aircraft and flight path parameters. See sample listing presented in Appendix B. The variable DIST is the flight path track distance along the ground. The variable TAS is the true airspeed and EAS is the indicated airspeed. DALPH/DT and DTHTA/DT are the time rate of change of the angle of attack and fuselage angle respectively. In addition, the user may also obtain the following values through the common block /EXCHNG/:

SROLL	-	distance to liftoff (ft)
S35	-	track distance to 35 ft (ft)
V35	-	speed (EAS) at 35 ft altitude (knots)
T5J	-	ground distance covered by aircraft at time of engine failure

The variable T5J may be used in conjunction with the subroutine ROLL to calculate accelerate – stop distances.

The program will terminate normally when the end speed is reached (VEND) or when the maximum specified altitude (HMAX) is attained. Abnormal termination will occur under several conditions:

- flight path constraints cannot be met by further reduction in angle of attack
- aircraft cannot accelerate at input rate of climb (RTCL)
- aircraft altitude goes negative

- ground track distance greater than 10. nautical miles
- ground run exceeds 90 seconds
- elapsed time greater than 300 seconds

For further definitions and explanations refer to the listing of TAKOFF and supporting subroutines contained in Appendix C, and the example case presented in Appendix B.

Subroutine LANDING

The subroutine LANDING simulates the landing maneuver of a given aircraft. As in the TAKOFF subroutine, the aircraft is treated as a point mass confined to motion in a vertical plane and the rotational dynamics are neglected. The landing is divided into three distinct phases: the approach, the flare to touchdown, and the rollout.

The landing approach is made at the thrust and angle of attack required for zero acceleration along and normal to the flight path. These values of thrust and angle of attack for the steady state approach may either be input by the user or calculated in subroutine ZERO.

The flare is executed by increasing the angle of attack at a specified rate (DADT). The program iterates on the flare initiation altitude until the rate of sink at touchdown equals a specified value (SINKTD).

Flight path control during the flare is obtained by monitoring the load factor, fuselage angle and pitch rate. These three variables are all restricted to values less than or equal to values specified by the user. If any of these conditions are violated, the angle of

attack is reduced until all constraints are satisfied. Provisions are made for spoiler retraction and increases in power setting during the flare if desired.

After the touchdown, the rollout is then executed, with provisions in the program for any pilot reaction or delay time, brake application, deployment of spoilers and thrust reversing. The landing run terminates when the aircraft comes to rest.

Program Inputs

The inputs to subroutine LANDNG are through the argument list, common blocks /UNIV/ and /AERO/ and namelist /NAM3/.

The call to LANDNG is as follows: CALL LANDNG (INPC, IWRITE, WGROSS, SWING, XENG, PWRSET, MODE, IZERO, IREV, NER) where

INPC	-	program control	= 1	input data loaded
			= 2	program executed
			= 3	input data loaded and program executed
IWRITE	-	print control	= 1	no print out
			= 2	print out
			= 3	iteration on HFLARE printed out

WGROSS - aircraft gross weight (lbs)

SWING - reference wing area (sq ft)

XENG - number of engines

PWRSET - power setting

MODE - roll out option control (see below)

IZERO - approach values of thrust and angle of attack = 0 calculated by program
= 1 input by user

IREV - reverse thrust control = 0 no reverse thrust
= 1 reverse thrust used

NER - error indicator = 1 successful case

The user has two options for the ground roll calculations. An average deceleration rate and delay time may be input and the ground roll distance determined by a single formula (MODE = 2). The second option is the time step integration (MODE = 1).

The namelist /NAM3/ input variables are as follows:

ABAR - average deceleration (g's)

CDGEAR - drag increment due to gear

DADT - time rate of change of angle of attack during flare (deg/sec)

DTABS - temperature increment above 59°F (°F)

DTDTMX - maximum allowable pitch rate during flare (deg/sec)

DTPENG - throttling rate (percent/sec)

FAAFTR - field length factor

HAPP - approach screen height (ft)

HAPT - altitude of airport (ft)

HSPOIL - minimum spoiler retraction altitude (ft)

IPOWER - throttle modulation control

ISPOIL - spoiler retraction control

} see below

PWRIDL - power setting at idle thrust

PWRMAX - maximum power setting

PWRMRG - power setting margin

SINKTD - sink rate at touchdown (ft/sec)

TDELAY - delay time after touchdown before any braking action taken (sec)

THEMAX - maximum fuselage angle (deg)

TBRK - time before brake application (sec)

TFLP - time before flap retraction (sec)

TOFF - time before engine shut down (sec)

TREV - time before reverse thrust applied (sec)

TSPL - time before spoilers deployed (sec)

XDIST - desired landing distance (ft)

XMUBRK - braking coeffient of friction

XLFMAX - maximum allowable load factor

QUESS - see subroutine ZERO

HFLARE - initial quess of flare altitude (ft)

STEP - step size for iteration on HFLARE

In addition, the following are input through common block /LAND/

```
COMMON /LAND/ GAMAPP, VKAPP, ALPHMX
```

where

GAMAPP - flight path angle, measured positive below horizon (deg)

VKAPP - true airspeed at approach (knots)

ALPHMX - maximum allowable angle of attack during flare, set equal to THEMAX in subroutine ZERO (deg)

The user may choose to input the approach speed (VKAPP) or input the desired landing distance (XDIST) and the program will calculate the required approach speed. This calculated approach speed is based on a constant load factor flare (XLFMAX) and the input values of ABAR, XLFMAX, GAMAPP, SINKTD and TDELAY. To use this option, the user should set VKAPP = 0. before the call to LANDNG.

The approach values of thrust per engine and angle of attack may be input (IZERO = 1) through common block /AERO/ or calculated by subroutine ZERO (IZERO = 0).

The guess for the flare initiation altitude can be input by the user or calculated by the program (use default value of HFLARE). The variable STEP controls the step size in the iteration to find the proper flare height. For powered – lift or any other aircraft that experience negative ground effects (suck-down), it is suggested that HFLARE and STEP be input as 40.0 ft and 0.9 respectively.

For ISPOIL = 0, the spoiler deflection angle (DELSPL) remains constant throughout the flare, and for IPOWER = 0, the throttle setting (PWRSET) is held fixed during the flare. For those aircraft experiencing negative ground effects, the spoilers may be retracted to obtain direct lift control (ISPOIL = 1) and the power setting advanced during the flare (IPOWER = 1). If the load factor during the flare should decrease, usually occuring when at or near maximum allowable angle of attack, the power setting will be advanced. Spoilers may be retracted when the aircraft is at an altitude of HSPOIL or when PWRMAX - PWRSET < PWRMRG.

If the user chooses to use reverse thrust during the ground roll, he must supply a subroutine REVRSE, which returns the total force

coefficients along and normal to the ground roll velocity vector. It is invisioned that this subroutine would be similar in structure to subroutine ARODYN, using the same common blocks to transfer required variable values. When using reverse thrust, IREV = 1 must be input.

During the ground roll, the user may select to apply brakes, retract flaps, deploy spoilers, shut engines down or use reverse thrust, or any combination thereof. Following touchdown, there is a delay of TDELAY seconds during which time no braking action is taken. After this delay time interval has elapsed, the various braking techniques listed above may be employed, with additional time delays for each (TBRK, TFLP, TOFF, TREV, TSPL). Spoilers are extended at a rate of 90 degrees per second, 90 degrees being fully deployed, and flaps are retracted to zero degrees deflection at a rate of 10.0 degrees/sec. If no reverse thrust is used, the power setting is reduced to idle (PWRIDL) at the specified rate of DTPENG percent/sec. If reverse thrust is applied, the power setting is advanced to 100 percent at the specified rate of DTPENG percent/sec. If the user desires the flap setting, spoiler deflection or throttle setting to remain fixed at the value at touchdown, he should input a large value for the associated time delay interval (say, 100 seconds).

The user may input all, some, or none of the input variables discussed above. The default values of these variables are listed below.

ABAR = 0.35 g's, CDGEAR = 0.0, DADT = 5.0 deg/sec, DTABS = 0.0 °F, DTDTMX = 7.0 deg/sec, DTPENG = 3.0 percent/sec, FAAFTR = 0.60, HAPP = 35.0 ft, HAPT = 0.0 ft., THEMAX = 10.0 deg, TBRK = 0.0 sec,

TFLP = 0.0 sec, TOFF = 0.0 sec, TREV = 0.0 sec,

TSPL = 0.0 sec, XDIST = 5000 ft,

XMUBRK = 0.40, XLFMAX = 1.15, HFLARE = 0.0 ft,

STEP = 0.9

If the default value of CDGEAR is used, the program will calculate a value for the gear drag based on an empirical formula as a function of gross weight and wing area.

For further definitions and explanations refer to the listing of LANDNG and supporting subroutines contained in Appendix C, and the example case presented in Appendix B.

Subroutine ZERO

Given the aircraft configuration, its velocity and flight path angle, subroutine ZERO determines the required values of thrust per engine and angle of attack for zero acceleration along and normal to the flight path.

The mathematical problem entails driving two functions of two independent variables simultaneously to zero.

$$\text{Let} \quad f(T,\alpha) = dV/dt$$

$$g(T,\alpha) = d\gamma/dt$$

where

T - engine thrust

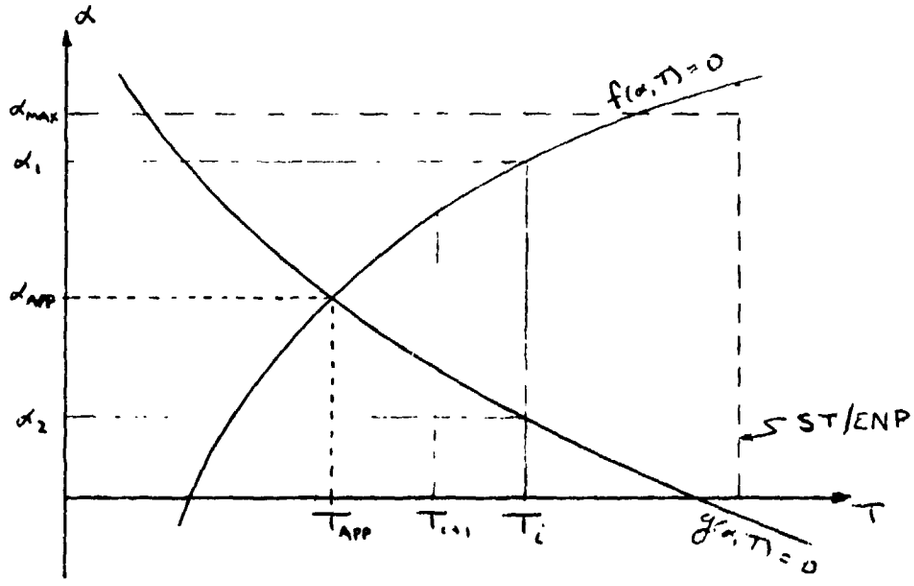
α - angle of attack

dV/dt - time rate of change of velocity along flight path

$d\gamma/dt$ - time rate of change of flight path angle

Subroutine ZERO generates the locus of $f(T,\alpha) = 0$ and $g(T,\alpha) = 0$, and then searches for their intersection.

The values of thrust and angle of attack at this point are the required value for the steady state approach.



The above figure shows a typical plot of the f and g functions. The values α_{APP} and T_{APP} are the desired values of angle of attack and thrust. The upper bound on the thrust is the static sea level thrust. The search for α_1 and α_2 is made on the interval $-\alpha_{max} \leq \alpha_i \leq \alpha_{max}$, $i = 1,2$. The error is then defined to be the difference between α_1 and α_2. The subroutine varies the value of the thrust until the error is driven to zero, or $\alpha_1 = \alpha_2$. In varying the thrust, $T_{i+1} = T_i$ times STEP, where STEP is an input parameter, usually less than 1.0. The initial guess at the value of thrust is defined to be the sea level static thrust times the parameter QUESS:

$$T_1 = T_{SLS} * QUESS$$

It is suggested that the user make this initial quess of the thrust "high", so that the program does not search in areas where there is no solution of $f(\alpha,T) = 0$. Recommended values of QUESS are:

1) conventional aircraft — QUESS = 0.50
2) powered — lift — QUESS = 0.80

The default value for QUESS is 0.80. In the main calling program, the user must specify a value for the total sea level static thrust (ST) before the call to LANDNG is made.

Subroutine ZERO may also be used to determine the static performance of an aircraft, given its velocity and flight path angle. The flight path angle is measured positive below the horizon (i.e., if the aircraft is climbing at a flight path angle of 5.0 degrees, the call to ZERO would be made with GAMAPP = -5.0).

The call to ZERO would be as follows:

```
CALL ZERO (NER, EN, ALT, DTABS, KENG, PWRSET, QUESS)
```

where

NER - error indicator = 1 successful case

EN - aircraft Mach no.

ALT - aircraft altitude (ft)

DTABS - temperature increment above 59°F (°F)

KENG = 1

PWRSET - power setting (returned)

QUESS - initial quess parameter

The following common blocks must also be used:

/UNIV/ - passes ENP, ST, W, etc.

/LAND/ - passes GAMAPP, ALPHMX

/AREO/ - passes QS, VEL, HABS

where

QS = product of dynamic pressure and wing area

VEL = aircraft speed (ft/sec)

HABS = absolute altitude of aircraft (ft)

Subroutines ENGINE and ARODYN are both called by ZERO. The call to ENGINE is made with KENG = 1 (see description of subroutine ENGINE). The user may choose to work either with gross thrust or net thrust, as long as he is consistent in the use of the variable "THRUST" in the definition of total force coefficients and power setting (PWRSET).

Subroutines ARODYN and ENGINE

The takeoff and landing subroutines described above require the user to provide two subroutines to compute total force coefficients and determine various required engine characteristics (e.g., thrust per engine and fuel flow). The format and structure of these subroutines is left to the discretion of the user.

I. Subroutine ARODYN

This subroutine computes the total force coefficients along the flight path and normal to the flight path as a function of angle of attack and thrust. A force coefficient in a particular direction is

defined to be the sum of all forces (excluding the weight component) in that particular direction divided by the dynamic pressure times the wing area.

$$C_x \equiv \frac{\vec{e}_x \cdot \Sigma \vec{F}}{qS}$$

The transfer of the various computer variables to and from subroutine ARODYN is through labeled common blocks /UNIV/ and /AERO/. Of primary concern is the common block /AERO/:

/AERO/ VEL, QS, HABS, THRUST, TVECT, ANGLE, DELFD, DELSPL, ALPHA, CX, CY, CL, CD, RHO, GRCD, IFAST

The input variables from TAKOFF, LANDNG and ZERO are:

VEL	-	aircraft velocity along flight path (ft/sec)
QS	-	dynamic pressure time wing area (lbs)
HABS	-	absolute altitude of aircraft (ft)
THRUST	-	thrust (gross or net) per engine (lbs)
TVECT	-	total vectored thrust (lbs)
ANGLE	-	angle of vectored thrust relative to aircraft center line, positive down (deg)
DELFD	-	flap deflection (deg)
DELSPL	-	spoiler deflection (deg)
ALPHA	-	angle of attack (deg)
RHO	-	air density (slugs/ft^3)
GRCD	-	drag increment due to gear

The return from ARODYN should be:

CX - total force coefficient along flight path

CY - total force coefficient normal to flight path

The output variable CL and CD are provided to the user as a means to distinguish between pure aerodynamic coefficients and total force coefficients. The output variables CL and CD are printed out in the time histories, but are not used in the actual calculations. If desired, in subroutine ARODYN, CL and CD may be directly equated to CY and CX, respectively.

There is a certain amount of redundancy among some of the input variables. The user may utilize only those variables he desires and disregard the others. Due to the wide range of velocities encountered during the takeoff and landing maneuver, there will be a correspondingly large variation in the magnitude of the force coefficients which must be accommodated in subroutine ARODYN.

II. Subroutine ENGINE

This subroutine provides the various propulsion data to subroutines TAKOFF, ZERO and LANDNG.

The inputs to subroutine ENGINE are through the argument list and labeled common blocks /AERO/ and /UNIV/.

The call to ENGINE is as follows:

```
CALL ENGINE (ALT, DTABS, EN, PWRSET, WFUEL, KENG)
```

where

ALT - aircraft altitude

DTABS - temperature increment above 59°F (°F)

EN - aircraft Mach no.

PWRSET - power setting (see below)

WFUEL - fuel flow (lbs/hr)

KENG - engine control parameter (see below)

The variable PWRSET is defined to be:

$$\text{PWRSET} = \frac{\text{Net thrust}}{\text{Net thrust available}}$$

and is the parameter used in controlling the thrust level. It is used for power setting management during the takeoff (e.g. throttle cutback for noise abatement), thrust modulation during the landing flare, and is calculated for the steady state approach.

Used in conjunction with PWRSET is the variable KENG, the engine control parameter. For KENG = 0 or 2 (takeoff and landing respectively), the value of PWRSET is input and the thrust and fuel flow returned. In the static performance calculations of subroutine ZERO, the call to ENGINE is made with KENG = 1. Here, the thrust per engine is input and PWRSET and fuel flow are calculated.

The user may choose to work with either the gross thrust per engine or the net thrust per engine, provided he uses the variable THRUST properly in the calculation of the total force coefficients. For example, when using gross thrust per engine, the ram drag must be included in the total summation of forces. If the gross thrust vector and ram drag vector are collinear, the user may choose instead to work simply with the net thrust.

Refer to the example case presented for an illustration of subroutines AERODYN and ENGINE.

REFERENCES

1. Jim Webb, [illegible] NG, The Iowa State University Press, Ames, 1971.

2. Susan E. [illegible] "Computer Programs for Estimation of STOL Takeoff, Landing, and Static Performance", NASA TM X-62,217, December 1972.

APPENDIX A

EQUATIONS

Equations used in TAKOFF and LANDNG

1) Equation of motion along flight path

$$dV/dt = (g/W)(-C_x qS - W \sin \gamma)$$

2) Equation of motion normal to flig. path

$$d\gamma/dt = (g/WV)(C_y qS - W \cos \gamma)$$

3) Equation of motion during ground roll

$$dV/dt = (g/W)[-W\mu + qS(C_y\mu - C_x)]$$

where

g = gravity constant

W = aircraft weight

q = dynamic pressure

S = wing area

γ = flight path angle

V = aircraft velocity

C_x = total force coefficient along flight path

C_y = total force coefficient normal to flight path

4) Load factor

$$XLF = \frac{qSC_y}{W \cos \gamma}$$

5) Constant rate of climb equation

$$\text{Rate of climb} \triangleq \text{ROC} = V \sin \gamma$$

For ROC to be constant with time,

$$\frac{d\text{ROC}}{dt} \equiv 0$$

$$\therefore \quad \frac{d\text{ROC}}{dt} = \frac{d}{dt}(V \sin \gamma)$$

$$= \frac{dV}{dt} \sin \gamma + V \cos \gamma \frac{d\gamma}{dt} = 0$$

Substitute for terms dV/dt and dγ/dt from equations 1) and 2)

$$(g/W)(-C_x qS - W \sin \gamma) \sin \gamma + V \cos \gamma (g/WV)(C_y qS - W \cos \gamma) \cos \gamma = 0$$

$$-C_x qS \sin \gamma - W \sin^2 \gamma + C_y qS \cos \gamma - W \cos^2 \gamma = 0$$

or

$$qS (C_y \cos \gamma - C_x \sin \gamma) - W = 0$$

6) Rotational rate approximations by finite difference

$$\theta = \gamma + \alpha - i_w$$

where θ - pitch attitude (fuselage angle)

γ - flight path angle

α - angle of attack

i_w - incidence of wing

Differentiating with respect to time we obtain:

$$\frac{d\theta}{dt} = \frac{d\alpha}{dt} + \frac{d\gamma}{dt}$$

where $\frac{d\gamma}{dt}$ is given by equation 2).

$\frac{d\alpha}{dt}$ is approximated by the finite difference form:

$$\frac{d\alpha}{dt} = (\alpha_{now} - \alpha_{past})/\Delta t$$

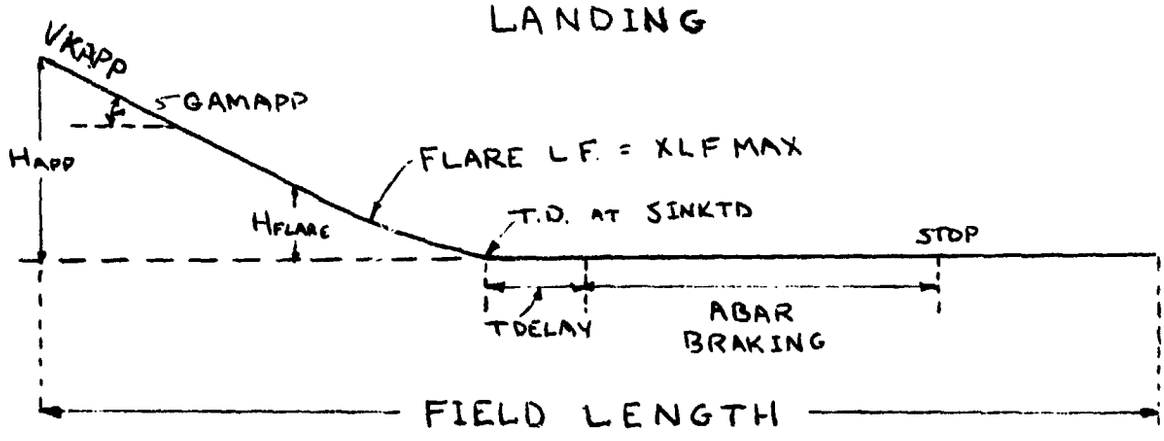
where

α_{now} = current value of angle of attack

α_{past} = previous value of angle of attack

Δt = integration time interval

7) Simplified landing equations — for constant load factor (XLF_{max}), constant speed (V_{APP}) flare



where: $$R = \frac{V_{APP}^2}{g(XLF_{max} - 1.0)}$$

$$h_{flare} = \frac{V_{APP}^2(\gamma_{APP}^2 - \gamma_{TD}^2)}{2g\ (XLF_{max} - 1.0)}$$

and

$$XDIST = \frac{35.0}{\tan \gamma_{APP}} + \frac{R\ \gamma_{APP}}{2} \left(1.0 - \frac{\gamma_{TD}}{\gamma_{APP}}\right)^2$$

$$+ \ V_{APP}\ t_{DELAY} + \frac{V_{APP}^2}{2g\ *\bar{a}}$$

where

t_{DELAY} - delay time before braking

$\bar{a}$ - average deceleration (g's)

APPENDIX B

SAMPLE CASE

Shown below is an example of the input, calling format, subroutines ARODYN and ENGINE, and the print out obtained for the takeoff and landing programs.

The main calling program TEST1 is set up to do the takeoff and landing of a Boeing 727-200. The required common blocks are shown, but others may be added if required. Note that the spoiler deflection angle DELSPL has been set equal to zero before the call to TAKOFF. Also note that before calling the landing program, the flap deflection DELFD, the spoiler deflection DELSPL and the total static sea level thrust ST must be given values externally to LANDNG.

The namelist input was as follows:

```
$NAM1 NPAGE = 48, RTCL = 530., THTFLY = 20., HMAN = 2000. $END
$NAM2 XPOWER(2) = 0.75, XHPWR(2) = 750., XPOWER(3) = 0.95
      XHPWR(3) = 1750.  $END
$NAM3 DADT = 1.0, IPOWER = 0, TFLP = 99., PWRIDL = 0.10,
      SINKTD = 3.0, TOFF = 3.0, TSPL = 2.00, XMUBRK = 0.45  $END
```

Subroutine ARODYN calculates the lift and drag coefficients of the 727-200 as a function of angle of attack, flap and spoiler deflection. The increments of lift and drag due to flaps is determined in a table look-up format. The increase in drag and loss in lift increment due to

spoilers is assumed to vary linearly with spoiler deflection. Once the lift and drag coefficients are computed, the thrust components, normalized with respect to dynamic pressure times wing area (QS), are added in to determine the total force coefficients CX and CY.

Subroutine ENGINE computes the thrust and fuel flow of the JT8D engine, based on a simplified model. The thrust lapse is assumed to be linear with Mach number, and the fuel flow assumed linear with power setting. Note the use of the parameter PWRSET. The subroutine consists basically of two sections. For KENG = 0 or 2, the power setting PWRSET is input and the thrust and fuel flow returned. For KENG = 1, the thrust is input and PWRSET and fuel flow returned. No reverse thrust was used during the landing ground roll and subroutine REVRSE is simply a dummy subroutine.

This particular run was made on the Lawrence Berkeley Laboratory CDC 7600 and required a CPU time of 2.38 seconds.

```
TEST1
                PROGRAM TEST1(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
                COMMON /UNIV/ NPC    ,NSC    ,IDC    ,H      ,ST     ,R      ,W      ,
               1WF     ,EM     ,VMO    ,EMMO   ,ALPHL0,CLALPH,SW     ,AR     ,B      ,
               2EYEW   ,ENP    ,TA     ,WG     ,WGS    ,KWRITE,DLMC4
               3,KSIZE
                COMMON /AERO/ VEL,QS,HABS,THRUST,TVECT, ANGLE,DELFD,DELSPL,ALPHA,
               9CX,CY,CL,CD,RHO,GRCD,IFAST
                COMMON/LAND/ GAMAPP,VKAPP,ALPHMX
                COMMON /EXCHNG/ SROLL,S35,V35,T5J
                WRITE(6,100)
            100 FORMAT(1H1,///)
      6         WG = 172000.
      7         SWING = 1720.
     11         ENP = 3.0
     12         V1 = 125.
     14         VR = 135.
     15         VEND = 250.
     17         DELSPL = 0.0
     20         CALL TAKOFF(3,9,WG,SWING,ENP,V1,VR,VEND)
     27         DELSPL = 0.0
     30         DELFD = 40.
     31         WG = 150000.
     33         ST = 42000.
     34         VKAPP = 130.
     36         GAMAPP = 3.0
     40         CALL LANDNG(3,2,WG,SWING,ENP,PWRSET,1,1,0,NER)
     51         END


PROGRAM LENGTH INCLUDING I/O BUFFERS

01165

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
100     - 000065

BLOCK NAMES AND LENGTHS
UNIV    - 000030/01  AERO    - 000020/02  LAND    - 000003/03  EXCHNG - 000004/04

VARIABLE ASSIGNMENTS
DELFD   - 000006/02  DELSPL - 000007/02  ENP     - 000021/01  GAMAPP - 000000/03  NER     - 000116      PWRSET - 000115
ST      - 000004/01  SWING  - 000111     VEND    - 000114     VKAPP  - 000001/03  VR      - 000113      V1     - 000112
WG      - 000023/01

START OF CONSTANTS-000054       TEMPS--000107    INDIRECTS-000111

7600 COMPILATION -- RUN76 LEVEL 5C      73/07/04.

ROUTINE COMPILES IN 044000
```

```
ARODYN
                SUBROUTINE ARODYN
                REAL NU
                COMMON /UNIV/ NPC    ,NSC    ,IDC    ,H      ,ST     ,R      ,W      ,
               1WF    ,EM     ,VMO    ,EMMO   ,ALPHL0,CLALPH,SW     ,AR     ,B      ,
               2EYEW   ,ENP    ,TA     ,WG     ,WGS    ,KWRITE,DUMC4
               3,KSIZE
                COMMON /AERO/ VEL,QS,HABS,THRUST,IVECT, ANGLE,DELFD,DELSPL,ALPHA,
               9CX,CY,CL,CD,RHO,GRCD,IFAST
                DIMENSION XDELFD(6),XDELCL(6),XDELCD(6),XVDEL6(6)
                DATA XDELFD/0.0,5.0,10.,15.,20.,25./
                DATA XDELCL/0.0,0.186,0.347,0.482,0.60,0.702/
                DATA XDELCD/0.0,.0148,.0295,.0451,.0607,.0837/
                DATA XVDEL6/1.0,.995,.990,.980,.970,.955/
                IF(IFAST .EQ. 1) GO TO 1
      3         CALL ITRLN(XDELFD,XDELCD,DELFD,DELCDF,6)
      7         CALL ITRLN(XDELFD,XDELCL,DELFD,DELCLF,6)
     13         CALL ITRLN(XDELFD,XVDEL6,DELFD,VDEL6,6)
     21         SA5 = 0.016
     22         CLALPH = 4.5
     24         ALPHL0 = -1.5
     25         SA7 = 0.0546
     27         SIGMA = 0.6
     30         DCLSPL = 0.31*(DELSPL/90.)
     32         DCDSPL = 0.12*(DELSPL/90.)
     34         CL = CLALPH*(ALPHA - ALPHL0)*.017453 + DELCLF
     40         CL = CL - DCLSPL
     42         CD = SA5 + DELCDF +(SA7/VDEL6)*(CL - SIGMA*DELCLF)**2 + GRCD
     51         CD = CD + DCDSPL
     53       1 ALPHX = ALPHA * .0174533
     55         IF(QS .EQ. 0.0)QS = 0.1
     57         CX = CD - THRUST*ENP*COS(ALPHX)/QS
     65         CY = CL + THRUST*ENP*SIN(ALPHX)/QS
     74         RETURN
     74         END


SUBPROGRAM LENGTH

00171

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1       - 000054

BLOCK NAMES AND LENGTHS
UNIV    - 000030/01  AERO    - 000020/02

VARIABLE ASSIGNMENTS
ALPHA   - 000010/02  ALPHL0 - 000013/01  ALPHX   - 000170      CD      - 000014/02  CL      - 000013/02  CLALPH - 000014/01
CX      - 000011/02  CY      - 000012/02  DCDSPL - 000167      DCLSPL - 000166      DELCDF - 000160      DELCLF - 000161
DELFD   - 000006/02  DELSPL - 000007/02  ENP     - 000021/01  GRCD    - 000016/02  IFAST  - 000017/02  NU      - 000127
QS      - 000001/02  SA5     - 000163      SA7     - 000164      SIGMA   - 000165      THRUST - 000003/02  VDEL6   - 000162
XDELCD - 000144      XDELCL - 000136      XDELFD - 000130      XVDEL6 - 000152
```

```
ENGINE
                SUBROUTINE ENGINE(ALT,DTABS,EN,PWRSET,WFUEL,KENG)
                COMMON /AERO/ VEL,QS,HABS,THRUST,TVECT, ANGLE,DELFD,DELSPL,ALPHA,
               9CX,CY,CL,CD,RHO,GROD,IFAST
                IF(KENG .EQ. 1)GO TO 10
     11         TO = 14000.
     12         THRUST = (TO - 6.0*EN*1100.)*PWRSET
     16      11 WFUEL = THRUST*0.63*PWRSET
     20         RETURN
     21      10 PWRSET = THRUST/(TO - 6.0*EN*1100.)
     25         GO TO 11
     26         END


SUBPROGRAM LENGTH

00043

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
10      - 000022     11      - 000017

BLOCK NAMES AND LENGTHS
AERO    - 000020/01

VARIABLE ASSIGNMENTS
THRUST - 000003/01  TO      - 000042

START OF CONSTANTS-000031       TEMPS--000036    INDIRECTS-000042

7600 COMPILATION -- RUN76 LEVEL 5C      73/07/04.

ROUTINE COMPILES IN 044000
```

```
REVRSE
                SUBROUTINE REVRSE
                RETURN
      2         END


SUBPROGRAM LENGTH

00010

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS

START OF CONSTANTS-000005       TEMPS--000006    INDIRECTS-000010

7600 COMPILATION -- RUN76 LEVEL 5C      73/07/04.

ROUTINE COMPILES IN 044000
```

** INPUTS TO TAKE OFF - ALTITUDE = 0.0 TEMPERATURE = 59.0 DEG. F

A/C CHARACTERISTICS
GROSS RAMP WT. = 172000 WING AREA = 1720 STATIC SEA LEVEL THRUST = 14000
WING LOADING = 100.0 THRUST/WEIGHT = .244

A/C PARAMETERS,
NO. ENGINES = 3.0 CDGEAR = .0287 EYEWNG = 1.0 TAIL SCRAPE ANGLE = 10.0

FLIGHT PATH CONTROL PARAMETERS
MAX LOAD FACTOR = 1.10 GEAR RETRACTION ALT. = 25.0 MAX FLOOR ANGLE = 20.0
MANEUVER ALT. = 2000 ACCELERATE RATE OF CLIMB = 530

PARAMETER VARIATION RATES
DADT = 1.0 DFLPDT = 3.0 DTGR = 5.0 DTPDWN = 5.0
DTDUP = 6.0 DTVECT = 10.0 DTFAIL = 3.0

POWER, VECTORED THRUST, AND FLAP SCHEDULES

THROTTLE/POWER SETTING

PWRSET	1.00	.75	.95	1.00	1.00
SPEED	0.0	0.0	0.0	999.0	999.0
ALTITUDE	0	750	1750	0	0

VECTORED THRUST ANGLE

ANGLE	0.0	0.0	0.0	0.0	0.0
SPEED	0.0	999.0	999.0	999.0	999.0
ALTITUDE	0	0	0	0	0

FLAP DEFLECTION ANGLE

DELFD	15.0	5.0	2.0	0.0	0.0
SPEED	0.0	0.0	200.0	210.0	0.0
ALTITUDE	0	250	0	0	0

ALL SPEEDS ARE INDICATED AIR SPEEDS AND ALL ALTITUDES ARE ABSOLUTE ALTITUDES

TAKEOFF (ELEVATION = 0 FT)

TIME (SEC)	DIST. (FEET)	WEIGHT (LBS)	ALT. (FEET)	TAS (KTS)	EAS (KTS)	MACH NO.	ACCEL (FPS2)	CL	CD	ALPHA (DEG)	GAMMA (DEG)	R/C (FPM)	LOAD FACT	THRUST (LBS)	FUS. ANG.	DALPH /DT	DTHTA /DT
0.0	0.0	172000	0.0	0.0	0.0	0.000	7.22	.678	.0982	1.00	0.00	0.0	0.00	42000	0.0	0.00	0.00
1.0	3.6	171993	0.0	4.3	4.3	.006	7.20	.678	.0982	1.00	0.00	0.0	0.00	41885	0.0	0.00	0.00
2.0	14.4	171985	0.0	8.5	8.5	.012	7.17	.678	.0982	1.00	0.00	0.0	0.00	41757	0.0	0.00	0.00
3.0	32.4	171978	0.0	12.8	12.8	.019	7.14	.678	.0982	1.00	0.00	0.0	0.00	41630	0.0	0.00	0.00
4.0	57.5	171971	0.0	17.0	17.0	.025	7.10	.678	.0982	1.00	0.00	0.0	0.00	41504	0.0	0.00	0.00
5.0	89.7	171964	0.0	21.2	21.2	.031	7.06	.678	.0982	1.00	0.00	0.0	0.00	41378	0.0	0.00	0.00
6.0	129.0	171956	0.0	25.3	25.3	.038	7.02	.678	.0982	1.00	0.00	0.0	0.00	41253	0.0	0.00	0.00
7.0	175.3	171949	0.0	29.5	29.5	.044	6.98	.678	.0982	1.00	0.00	0.0	0.00	41129	0.0	0.00	0.00
8.0	228.6	171942	0.0	33.6	33.6	.050	6.93	.678	.0982	1.00	0.00	0.0	0.00	41005	0.0	0.00	0.00
9.0	288.8	171935	0.0	37.7	37.7	.056	6.88	.678	.0982	1.00	0.00	0.0	0.00	40882	0.0	0.00	0.00
10.0	355.9	171928	0.0	41.8	41.8	.063	6.83	.678	.0982	1.00	0.00	0.0	0.00	40761	0.0	0.00	0.00
11.0	429.8	171920	0.0	45.8	45.8	.069	6.78	.678	.0982	1.00	0.00	0.0	0.00	40640	0.0	0.00	0.00
12.0	510.5	171913	0.0	49.8	49.8	.075	6.72	.678	.0982	1.00	0.00	0.0	0.00	40520	0.0	0.00	0.00
13.0	597.9	171906	0.0	53.7	53.7	.081	6.66	.678	.0982	1.00	0.00	0.0	0.00	40401	0.0	0.00	0.00
14.0	692.0	171899	0.0	57.7	57.7	.087	6.60	.678	.0982	1.00	0.00	0.0	0.00	40283	0.0	0.00	0.00
15.0	792.6	171892	0.0	61.5	61.5	.093	6.53	.678	.0982	1.00	0.00	0.0	0.00	40167	0.0	0.00	0.00
16.0	899.8	171885	0.0	65.4	65.4	.098	6.46	.678	.0982	1.00	0.00	0.0	0.00	40051	0.0	0.00	0.00
17.0	1013.5	171878	0.0	69.2	69.2	.104	6.40	.678	.0982	1.00	0.00	0.0	0.00	39937	0.0	0.00	0.00
18.0	1133.5	171871	0.0	73.0	73.0	.110	6.33	.678	.0982	1.00	0.00	0.0	0.00	39824	0.0	0.00	0.00
19.0	1259.9	171864	0.0	76.7	76.7	.116	6.25	.678	.0982	1.00	0.00	0.0	0.00	39712	0.0	0.00	0.00
20.0	1392.6	171857	0.0	80.4	80.4	.121	6.18	.678	.0982	1.00	0.00	0.0	0.00	39602	0.0	0.00	0.00
21.0	1531.4	171850	0.0	84.0	84.0	.127	6.10	.678	.0982	1.00	0.00	0.0	0.00	39493	0.0	0.00	0.00
22.0	1676.3	171843	0.0	87.6	87.6	.132	6.03	.678	.0982	1.00	0.00	0.0	0.00	39385	0.0	0.00	0.00
23.0	1827.3	171837	0.0	91.1	91.1	.137	5.95	.678	.0982	1.00	0.00	0.0	0.00	39278	0.0	0.00	0.00
24.0	1984.2	171830	0.0	94.6	94.6	.143	5.87	.678	.0982	1.00	0.00	0.0	0.00	39173	0.0	0.00	0.00
25.0	2146.9	171823	0.0	98.1	98.1	.148	5.79	.678	.0982	1.00	0.00	0.0	0.00	39070	0.0	0.00	0.00
26.0	2315.5	171816	0.0	101.5	101.5	.153	5.71	.678	.0982	1.00	0.00	0.0	0.00	38968	0.0	0.00	0.00
27.0	2489.8	171809	0.0	104.9	104.9	.158	5.62	.678	.0982	1.00	0.00	0.0	0.00	38867	0.0	0.00	0.00
28.0	2669.6	171802	0.0	108.2	108.2	.163	5.54	.678	.0982	1.00	0.00	0.0	0.00	38768	0.0	0.00	0.00
29.0	2855.1	171796	0.0	111.4	111.4	.168	5.46	.678	.0982	1.00	0.00	0.0	0.00	38670	0.0	0.00	0.00
30.0	3046.0	171789	0.0	114.6	114.6	.173	5.37	.678	.0982	1.00	0.00	0.0	0.00	38574	0.0	0.00	0.00
31.0	3242.2	171782	0.0	117.8	117.8	.178	5.29	.678	.0982	1.00	0.00	0.0	0.00	38479	0.0	0.00	0.00
32.0	3443.8	171775	0.0	120.9	120.9	.183	5.20	.678	.0982	1.00	0.00	0.0	0.00	38386	0.0	0.00	0.00
33.0	3650.5	171769	0.0	123.9	123.9	.187	5.11	.678	.0982	1.00	0.00	0.0	0.00	38294	0.0	0.00	0.00
34.0	3862.4	171762	0.0	126.9	126.9	.192	5.03	.678	.0982	1.00	0.00	0.0	0.00	38204	0.0	0.00	0.00
35.0	4079.3	171755	0.0	129.9	129.9	.196	4.94	.678	.0982	1.00	0.00	0.0	0.00	38115	0.0	0.00	0.00
36.0	4301.1	171749	0.0	132.8	132.8	.201	4.85	.678	.0982	1.00	0.00	0.0	0.00	38028	0.0	0.00	0.00
ROTATION (TIME = 36.8 AND TAS = 135.1 EAS = 135.1)																	
37.0	4527.8	171742	0.0	135.6	135.6	.205	4.76	.694	.0989	1.20	0.00	0.0	0.00	37943	.2	1.00	1.00
38.0	4759.2	171735	0.0	138.4	138.4	.209	4.62	.773	.1028	2.20	0.00	0.0	0.00	37859	1.2	1.00	1.00
39.0	4995.3	171729	0.0	141.1	141.1	.213	4.47	.851	.1074	3.20	0.00	0.0	0.00	37778	2.2	1.00	1.00
40.0	5235.8	171722	0.0	143.7	143.7	.217	4.29	.930	.1127	4.20	0.00	0.0	0.00	37700	3.2	1.00	1.00
41.0	5480.6	171716	0.0	146.2	146.2	.221	4.09	1.008	.1186	5.20	0.00	0.0	0.00	37625	4.2	1.00	1.00
42.0	5729.5	171709	0.0	148.5	148.5	.225	3.87	1.087	.1252	6.20	0.00	0.0	0.00	37554	5.2	1.00	1.00
43.0	5982.3	171702	0.0	150.8	150.8	.228	3.63	1.165	.1326	7.20	0.00	0.0	0.00	37487	6.2	1.00	1.00
LIFTOFF (TIME = 43.9 DIST = 6212.9 TAS = 152.6 EAS = 152.6)																	
44.0	6238.8	171696	.0	152.8	152.8	.231	3.35	1.244	.1406	8.20	.01	3.0	1.02	37424	7.2	1.00	1.11
45.0	6498.6	171689	.8	154.7	154.7	.234	2.79	1.311	.1479	9.05	.46	125.9	1.10	37369	8.5	-.00	.69
46.0	6761.3	171683	4.5	156.2	156.2	.236	2.40	1.287	.1453	8.75	1.15	317.9	1.10	37322	8.9	-.00	.69
47.0	7026.3	171676	11.4	157.5	157.5	.238	2.02	1.263	.1427	8.45	1.83	510.4	1.10	37282	9.3	-.50	.18
48.0	7293.1	171670	21.6	158.6	158.5	.240	1.63	1.248	.1410	8.25	2.51	703.9	1.10	37249	9.8	-.50	.16

TAKEOFF CONTINUED

TIME (SEC)	DIST. (FEET)	WEIGHT (LBS)	ALT. (FEET)	TAS (KTS)	EAS (KTS)	MACH NO.	ACCEL (FPS2)	CL	CD	ALPHA (DEG)	GAMMA (DEG)	R/C (FPM)	LOAD FACT	THRUST (LBS)	FUS. ANG.	DALPH /DT	DTHTA /DT
GEAR RETRACTION STARTED AT 48.3 SEC, COMPLETE AT 53.3 SEC																	
49.0	7561.5	171663	34.9	159.5	159.4	.241	1.34	1.236	.1363	8.10	3.18	896.5	1.10	37222	10.3	-.00	.67
DISTANCE TO 35 FT.= 7563.3 SCREEN SPEED(EAS) = 159.4 KNOTS																	
50.0	7931.0	171657	51.5	160.2	160.1	.242	1.12	1.224	.1293	7.95	3.85	1090.1	1.10	37200	10.8	-.00	.66
51.0	8101.3	171650	71.2	160.8	160.6	.243	.91	1.216	.1228	7.85	4.52	1293.4	1.10	37181	11.4	-.00	.67
52.0	8372.4	171644	94.2	161.3	161.0	.244	.70	1.208	.1162	7.75	5.18	1475.8	1.10	37166	11.9	-.00	.66
53.0	8643.8	171637	120.4	161.6	161.3	.245	.49	1.205	.1101	7.70	5.84	1667.7	1.10	37154	12.5	-.00	.67
54.0	8915.4	171631	149.[illegible]	161.8	161.5	.245	.20	1.201	.1074	7.65	6.50	1857.9	[illegible].10	37147	13.2	-.00	.67
55.0	9186.8	171624	182.3	161.9	161.5	.245	.01	1.154	.1027	7.05	7.09	2025.8	1.06	37145	13.1	-1.00	-.59
56.0	9458.1	171618	216.7	161.9	161.4	.245	.00	1.114	.0990	6.55	7.32	2089.8	1.02	37144	12.9	-.00	.12
FLAPS RETRACTED TO 5.0 DEG. IN 3.3 SEC.																	
57.0	9729.2	171611	251.7	161.9	161.3	.245	.01	1.099	.0976	6.35	7.39	2108.9	1.00	37143	12.7	-.50	-.46
58.0	10000.4	171605	286.9	161.9	161.3	.245	.13	1.096	.0923	7.35	7.41	2116.7	1.00	37142	13.8	1.00	1.03
59.0	10271.7	171598	322.3	162.0	161.3	.246	.24	1.088	.0871	8.35	7.44	2127.0	1.00	37138	14.8	1.00	1.02
60.0	10543.2	171592	357.7	162.2	161.4	.246	.38	1.070	.0822	9.35	7.42	2122.2	.99	37132	15.8	1.00	.94
61.0	10815.1	171585	393.0	162.4	161.5	.246	.22	1.117	.0862	10.35	7.45	2133.6	1.04	37125	16.8	1.00	1.22
62.0	11087.0	171579	429.5	162.5	161.5	.246	.00	1.121	.0867	10.40	7.81	2237.0	1.04	37123	17.2	-.50	-.21
63.0	11358.8	171572	467.2	162.5	161.4	.246	.01	1.093	.0837	10.05	7.98	2285.5	1.01	37122	17.0	-.00	.10
64.0	11630.5	171566	505.5	162.5	161.3	.246	.00	1.085	.0828	9.95	8.04	2302.6	1.01	37121	17.0	-.00	.04
65.0	11902.2	171559	543.9	162.5	161.2	.246	.01	1.081	.0824	9.90	8.06	2308.1	1.00	37121	17.0	-.00	.01
66.0	12173.9	171553	582.4	162.5	161.1	.246	.00	1.081	.0824	9.90	8.07	2311.1	1.00	37120	17.0	-.00	.01
67.0	12445.6	171546	620.9	162.5	161.0	.247	.00	1.081	.0824	9.90	8.07	2312.2	1.00	37119	17.0	-.00	.00
68.0	12717.3	171540	659.5	162.5	160.9	.247	.01	1.081	.0824	9.90	8.07	2311.3	1.00	37118	17.0	-.00	-.01
69.0	12989.0	171533	698.0	162.5	160.8	.247	.00	1.085	.0828	9.95	8.06	2309.8	1.00	37118	17.0	.50	.49
70.0	13260.8	171527	736.5	162.5	160.8	.247	.00	1.085	.0828	9.95	8.06	2309.5	1.00	37117	17.0	-.00	.00
RETARD THROTTLE SETTING TO 75.0 PERCENT IN 5.0 SEC.																	
71.0	13532.5	171520	774.9	162.5	160.7	.247	.01	1.038	.0779	9.35	7.96	2281.5	.95	36003	16.3	-1.00	-1.27
72.0	13804.5	171515	811.9	162.5	160.6	.247	.00	1.007	.0748	8.95	7.53	2157.0	.92	34146	15.5	-.00	-.53
73.0	14076.7	171509	846.6	162.5	160.5	.247	.00	.999	.0740	8.85	6.95	1993.9	.91	32290	14.8	-.00	-.60
74.0	14349.4	171505	878.3	162.5	160.4	.247	.00	.999	.0740	8.85	6.34	1820.0	.91	30434	14.2	-.00	-.62
75.0	14622.3	171501	907.2	162.5	160.4	.247	.00	1.003	.0744	8.90	5.71	1639.4	.91	28577	13.6	.50	-.14
76.0	14895.3	171497	933.1	162.5	160.3	.247	.03	1.050	.0791	9.50	5.17	1483.2	.95	27835	13.7	1.00	.63
77.0	15169.0	171493	957.3	162.5	160.3	.247	.01	1.089	.0832	10.00	4.98	1428.6	.98	27834	14.0	-.00	-.11
78.0	15442.5	171490	980.9	162.5	160.2	.247	.01	1.101	.0845	10.15	4.91	1410.3	.99	27834	14.1	-.00	-.05
79.0	15716.[illegible]	171486	1004.3	162.5	160.2	.247	.01	1.105	.0849	10.20	4.89	1403.9	1.00	27833	14.1	-.50	-.51
80.0	15989.5	171483	1027.7	162.5	160.1	.247	.01	1.109	.0854	10.25	4.88	1401.5	1.00	27833	14.1	-.00	-.01
81.0	16263.1	171479	1051.0	162.5	160.1	.247	.01	1.109	.0854	10.25	4.87	1399.4	1.00	27833	14.1	-.00	-.00
82.0	16536.6	171475	1074.4	162.5	160.0	.247	.00	1.113	.0858	10.30	4.87	1399.1	1.00	27832	14.2	.50	.49
83.0	16810.2	171472	1097.7	162.6	160.0	.247	.00	1.113	.0858	10.30	4.87	1398.7	1.00	27832	14.2	-.00	.00
84.0	17083.7	171468	1121.0	162.6	159.9	.247	.00	1.113	.0858	10.30	4.87	1399.7	1.00	27831	14.2	-.00	.00
85.0	17357.3	171464	1144.3	162.6	159.9	.247	.01	1.113	.0858	10.30	4.87	1399.6	1.00	27831	14.2	-.00	-.00
86.0	17630.8	171461	1167.6	162.6	159.8	.247	.01	1.113	.0858	10.30	4.87	1398.2	1.00	27831	14.2	-.00	-.01
87.0	17904.4	171457	1190.9	162.6	159.8	.247	.01	1.113	.0858	10.30	4.87	1397.6	1.00	27830	14.2	-.50	-.49
88.0	18178.0	171453	1214.2	162.6	159.7	.247	.01	1.113	.0858	10.30	4.87	1397.4	1.00	27830	14.2	-.50	-.49
89.0	18451.6	171450	1237.5	162.6	159.7	.247	.00	1.117	.0862	10.35	4.87	1397.7	1.00	27829	14.2	-.00	.01
90.0	18725.2	171446	1260.8	162.6	159.6	.247	.00	1.117	.0862	10.35	4.87	1398.6	1.00	27829	14.2	-.00	.00
91.0	18998.8	171442	1284.1	162.6	159.5	.247	.00	1.117	.0862	10.35	4.87	1398.3	1.00	27829	14.2	-.00	-.00
92.0	19272.4	171439	1307.4	162.6	159.5	.247	.01	1.117	.0862	10.35	4.86	1396.9	1.00	27828	14.2	-.00	-.01
93.0	19546.0	171435	1330.7	162.6	159.4	.247	.01	1.117	.0862	10.35	4.86	1395.9	1.00	27828	14.2	-.50	-.49
94.0	19819.6	171431	1354.0	162.6	159.4	.247	.01	1.117	.0862	10.35	4.86	1395.5	1.00	27828	14.2	-.50	-.50
95.0	20093.2	171428	1377.2	162.6	159.3	.247	.00	1.121	.0867	10.40	4.86	1395.7	1.00	27827	14.3	-.00	.00
96.0	20366.8	171424	1400.5	162.6	159.3	.247	.00	1.121	.0867	10.40	4.86	1396.4	1.00	27827	14.3	-.00	.00

TAKEOFF CONTINUED

TIME (SEC)	DIST. (FEET)	WEIGHT (LBS)	ALT. (FEET)	TAS (KTS)	EAS (KTS)	MACH NO.	ACCEL (FPS2)	CL	CD	ALPHA (DEG)	GAMMA (DEG)	R/C (FPM)	LOAD FACT	THRUST (LBS)	FUS. ANG.	DALPH /DT	DTHTA /DT
97.0	20840.5	171420	1423.7	162.6	159.2	.247	.01	1.121	.0867	10.40	4.86	1396.0	1.00	27826	14.3	-.00	-.00
98.0	20914.1	171417	1447.0	162.6	159.2	.247	.1	1.121	.0867	10.40	4.85	1394.4	1.00	27826	14.3	-.00	-.01
99.0	21187.8	171413	1470.2	162.6	159.1	.247	.00	1.125	.0871	10.45	4.85	1393.8	1.00	27826	14.3	.50	.49
100.0	21461.4	171409	1493.5	162.6	159.1	.247	.00	1.125	.0871	10.45	4.85	1393.4	1.00	27825	14.3	.50	.49
101.0	21735.1	171406	1516.7	162.6	159.0	.247	.00	1.125	.0871	10.45	4.85	1393.9	1.00	27825	14.3	-.00	.00
102.0	22008.8	171402	1539.9	162.6	159.0	.248	.00	1.125	.0871	10.45	4.85	1394.5	1.00	27825	14.3	-.00	.00
103.0	22282.5	171399	1563.2	162.6	158.9	.248	.01	1.125	.0871	10.45	4.85	1393.9	1.00	27824	14.3	-.00	-.00
104.0	22556.1	171395	1586.4	162.6	158.9	.248	.01	1.125	.0871	10.45	4.85	1392.2	1.00	27824	14.3	-.00	-.01
105.0	22829.8	171391	1609.6	162.6	158.8	.248	.01	1.125	.0871	10.45	4.85	1392.2	1.00	27823	14.3	-.50	-.49
106.0	23103.5	171388	1632.8	162.6	158.8	.248	.02	1.125	.0871	10.45	4.84	1391.6	1.00	27823	14.3	-.50	-.50
107.0	23377.3	171384	1656.0	162.6	158.7	.248	.00	1.128	.0875	10.50	4.85	1392.1	1.00	27823	14.3	-.00	.00
108.0	23651.0	171380	1679.2	162.6	158.7	.248	.00	1.128	.0875	10.50	4.85	1392.6	1.00	27822	14.3	-.00	-.00
109.0	23924.7	171377	1702.4	162.6	158.6	.248	.01	1.128	.0875	10.50	4.84	1391.9	1.00	27822	14.3	-.00	-.00
110.0	24198.4	171373	1725.6	162.7	158.6	.248	.01	1.128	.0875	10.50	4.84	1390.0	1.00	27821	14.3	-.00	-.01
111.0	24472.2	171369	1748.7	162.7	158.5	.248	.01	1.128	.0875	10.50	4.84	1390.3	1.00	21	14.3	-.50	-.49
ADVANCE THROTTLE SETTING TO 95.0 PERCENT IN 3.3 SEC.																	
112.0	24745.9	171365	1772.2	162.7	158.5	.248	.01	1.203	.0962	11.45	5.04	1449.2	1.07	29823	15.5	1.00	1.42
113.0	25019.4	171361	1797.6	162.7	158.4	.248	.01	1.231	.0995	11.80	5.60	1610.0	1.09	32049	16.4	-.00	.63
114.0	25292.7	171356	1826.0	162.7	158.4	.248	.06	1.231	.0995	11.80	6.26	1797.5	1.10	34273	17.1	-.00	.65
115.0	25565.7	171350	1857.5	162.7	158.3	.248	.00	1.199	.0957	11.40	6.88	1976.3	1.07	35236	17.3	-1.00	-.50
116.0	25838.4	171344	1891.3	162.7	158.3	.248	.00	1.152	.0902	10.80	7.18	2062.0	1.03	35236	17.0	-.00	.19
117.0	26111.1	171338	1926.0	162.7	158.2	.248	.01	1.132	.0880	10.55	7.30	2094.6	1.01	35235	16.8	-.00	.07
118.0	26383.7	171332	1961.0	162.7	158.1	.248	.01	1.125	.0871	10.45	7.34	2107.3	1.00	35234	16.8	-.00	.02
119.0	26656.3	171326	1996.2	162.7	158.0	.248	.01	1.125	.0871	10.45	7.35	2110.8	1.00	35234	16.8	-.00	.01
ACCELERATE TO CLIMB SPEED OF 250.0 KNOTS																	
120.0	26928.9	171321	2031.3	162.8	158.0	.248	.15	1.093	.0837	10.05	7.28	2091.5	.97	35232	16.3	-.50	-.67
121.0	27201.8	171315	2065.5	162.9	158.1	.248	.44	1.054	.0795	9.55	6.98	2007.3	.94	35227	15.5	-.50	-.91
122.0	27475.4	171309	2097.8	163.3	158.3	.249	.84	1.015	.0755	9.05	6.45	1860.1	.90	35217	14.5	-.50	-1.13
123.0	27750.3	171303	2127.2	164.0	158.9	.250	1.35	.975	.0717	8.55	5.71	1653.7	.87	35199	13.3	-.50	-1.33
124.0	28026.9	171297	2152.7	164.9	159.8	.251	1.95	.938	.0683	8.07	4.78	1392.6	.85	35171	11.9	-.25	-1.26
125.0	28305.9	171291	2173.5	166.3	161.0	.253	2.55	.918	.0665	7.82	3.75	1101.8	.84	35134	10.6	-.25	-1.3
126.0	28587.8	171285	2189.3	168.0	162.6	.256	3.15	.899	.0648	7.57	2.69	800..	.84	35086	9.3	-.25	-1.31
127.0	28873.1	171280	2200.1	170.0	164.6	.259	3.22	1.044	.0785	9.42	1.68	504..	1.00	35029	10.1	20.70	20.04
128.0	29161.8	171274	2208.6	171.9	166.4	.262	3.24	1.022	.0762	9.14	1.67	508.2	1.00	34973	9.8	-.33	-.33
129.0	29453.8	171268	2217.1	173.8	168.2	.265	3.25	1.001	.0742	8.87	1.65	508.1	1.00	34918	9.5	-.23	-.25
130.0	29749.0	171262	2225.5	175.7	170.1	.268	3.27	.980	.0722	8.61	1.64	508.2	1.00	34862	9.2	-.31	-.33
131.0	30047.5	171256	2234.0	177.7	171.9	.271	3.29	.960	.0703	8.35	1.62	508.2	1.00	34807	9.0	-.31	-.33
132.0	30349.3	171251	2242.5	179.6	173.8	.274	3.30	.940	.0685	8.10	1.60	508.1	1.00	34751	8.7	-.31	-.33
133.0	30654.3	171245	2250.9	181.6	175.7	.277	3.31	.921	.0667	7.86	1.58	508.1	1.00	34694	8.4	-.23	-.25
134.0	30962.7	171239	2259.4	183.5	177.5	.280	3.32	.903	.0651	7.62	1.57	508.1	1.00	34638	8.2	-.23	-.25
135.0	31274.4	171233	2267.9	185.5	179.4	.283	3.33	.885	.0636	7.40	1.55	508.0	1.00	34581	7.9	-.23	-.25
136.0	31589.5	171228	2276.3	187.5	181.3	.286	3.34	.867	.0621	7.17	1.53	508.0	1.00	34524	7.7	-.23	-.25
137.0	31907.9	171222	2284.8	189.5	183.2	.289	3.35	.850	.0607	6.95	1.52	508.0	1.00	34467	7.5	-.23	-.25
138.0	32229.6	171216	2293.3	191.5	185.1	.292	3.35	.833	.0594	6.74	1.50	508.0	1.00	34410	7.2	-.23	-.25
139.0	32554.7	171210	2301.7	193.4	187.0	.295	3.35	.817	.0581	6.54	1.48	507.9	1.00	34353	7.0	-.16	-.17
140.0	32883.1	171205	2310.2	195.4	188.9	.298	3.36	.801	.0569	6.33	1.47	508.0	1.00	34296	6.8	-.23	-.25
141.0	33214.9	171199	2318.7	197.4	190.8	.301	3.36	.786	.0558	6.14	1.46	508.3	1.00	34239	6.6	-.23	-.25
142.0	33550.1	171193	2327.1	199.4	192.7	.304	3.36	.771	.0547	5.95	1.44	508.2	1.00	34182	6.4	-.16	-.17
143.0	33888.6	171188	2335.6	201.4	194.6	.307	3.36	.757	.0536	5.76	1.43	508.3	1.00	34124	6.2	-.23	-.25
144.0	34230.5	171182	2344.1	203.4	196.5	.310	3.35	.743	.0527	5.59	1.41	508.3	1.00	34067	6.0	-.16	-.17

TAKEOFF CONTINUED

TIME (SEC)	DIST. (FEET)	WEIGHT (LBS)	ALT. (FEET)	TAS (KTS)	EAS (KTS)	MACH NO.	ACCEL (FPS2)	CL	CD	ALPHA (DEG)	GAMMA (DEG)	R/C (FPM)	LOAD FACT	THRUST (LBS)	FUS. ANG.	DALPH /DT	DTHTA /DT
145.0	34575.7	171176	2352.6	205.4	198.4	.313	3.35	.729	.0517	5.41	1.40	508.3	1.00	34010	5.8	-.23	-.25
FLAPS RETRACTED TO 2.0 DEG. IN 1.0 SEC.																	
146.0	34924.3	171171	2361.0	207.3	200.2	.316	3.36	.715	.0503	5.38	1.39	508.4	1.00	33953	5.8	1.17	1.16
147.0	35276.2	171165	2369.5	209.4	202.2	.319	3.53	.699	.0453	6.45	1.37	508.3	1.00	33894	6.8	-.16	-.17
148.0	35631.7	171159	2378.0	211.5	204.2	.322	3.53	.686	.0444	6.29	1.36	508.3	1.00	33834	6.6	-.16	-.17
149.0	35990.7	171154	2386.4	213.6	206.2	.326	3.53	.673	.0435	6.12	1.35	508.3	1.00	33774	6.5	-.16	-.17
150.0	36353.2	171148	2394.9	215.6	208.2	.329	3.53	.661	.0427	5.97	1.33	508.2	1.00	33714	6.3	-.16	-.17
FLAPS RETRACTED TO 0.0 DEG. IN .7 SEC.																	
151.0	36719.3	171143	2403.4	217.7	210.2	.332	3.53	.649	.0419	5.81	1.32	508.2	1.00	33654	6.1	-.16	-.17
152.0	37088.9	171137	2411.8	219.9	212.2	.335	3.67	.634	.0380	6.58	1.31	508.1	1.00	33592	6.9	-.16	-.17
153.0	37462.2	171131	2420.3	222.1	214.3	.339	3.67	.623	.0372	6.43	1.29	508.1	1.00	33530	6.7	-.16	-.17
154.0	37839.2	171126	2428.8	224.2	216.3	.342	3.67	.611	.0364	6.28	1.28	508.2	1.00	33467	6.6	-.16	-.17
155.0	38219.8	171120	2437.3	226.4	218.4	.345	3.67	.600	.0357	6.14	1.27	508.1	1.00	33405	6.4	-.16	-.17
156.0	38604.1	171115	2445.7	228.6	220.5	.349	3.67	.589	.0349	6.00	1.26	508.2	1.00	33342	6.3	-.16	-.17
157.0	38992.0	171109	2454.2	230.7	222.5	.352	3.67	.578	.0343	5.87	1.25	508.2	1.00	33279	6.1	-.16	-.17
158.0	39383.7	171104	2462.7	232.9	224.6	.355	3.66	.568	.0336	5.73	1.23	508.2	1.00	33217	6.0	-.16	-.17
159.0	39778.9	171098	2471.1	235.1	226.7	.359	3.66	.558	.0330	5.61	1.22	508.2	1.00	33155	5.8	-.08	-.09
160.0	40177.9	171093	2479.6	237.2	228.7	.362	3.66	.548	.0324	5.48	1.21	508.3	1.00	33092	5.7	-.16	-.17
161.0	40580.5	171087	2488.1	239.4	230.8	.365	3.65	.539	.0318	5.36	1.20	508.5	1.00	33030	5.6	-.16	-.17
162.0	40986.7	171082	2496.5	241.6	232.8	.369	3.65	.529	.0313	5.24	1.19	508.1	1.00	32968	5.4	-.16	-.17
163.0	41396.6	171076	2505.0	243.7	234.9	.372	3.64	.521	.0308	5.13	1.18	508.0	1.00	32905	5.3	-.08	-.09
164.0	41810.2	171071	2513.5	245.9	236.9	.375	3.63	.512	.0303	5.02	1.17	508.0	1.00	32843	5.2	-.08	-.09
165.0	42227.4	171065	2521.9	248.0	239.0	.378	3.62	.504	.0298	4.91	1.16	508.0	1.00	32781	5.1	-.08	-.09
EXECUTE PULLUP AT DADT = .13																	
166.0	42648.2	171060	2530.4	250.2	241.0	.382	3.59	.503	.0298	4.90	1.16	513.5	1.01	32720	5.1	.12	.17
167.0	43072.5	171054	2539.4	252.2	243.0	.385	3.43	.512	.0303	5.02	1.29	575.7	1.05	32659	5.3	.12	.33
168.0	43500.2	171049	2550.0	254.2	244.8	.38	3.19	.522	.0309	5.15	1.58	739.0	1.08	32602	5.7	.12	.48
169.0	43931.1	171044	2563.4	256.0	246.5	.391	2.87	.532	.0315	5.27	2.02	912.8	1.12	32550	6.3	.12	.63
170.0	44364.6	171038	2580.8	257.6	248.0	.393	2.47	.542	.0320	5.40	2.60	1185.1	1.16	32503	7.0	.12	.77
171.0	44800.3	171033	2603.3	258.9	249.2	.395	1.99	.552	.0326	5.52	3.33	1522.8	1.19	32464	7.9	.12	.91
172.0	45237.7	171027	2631.8	260.0	250.1	.397	1.52	.550	.0325	5.50	4.15	1904.4	1.19	32432	8.6	-.00	.45
173.0	45676.0	171022	2666.8	260.7	250.7	.398	1.05	.548	.0324	5.47	4.97	2286.8	1.20	32409	9.4	.12	.93
174.0	46114.9	171017	2708.1	261.2	251.0	.399	.60	.546	.0323	5.45	5.78	2666.7	1.20	32393	10.2	.12	.93
175.0	46553.7	171011	2755.7	261.4	251.0	.400	.15	.544	.0322	5.42	6.60	3043.1	1.19	32385	11.0	.12	.92
176.0	46991.9	171006	2809.1	261.4	250.8	.400	.01	.491	.0292	4.75	7.22	3331.1	1.08	32382	11.0	-.88	-.51
177.0	47429.8	171000	2865.5	261.4	250.6	.400	.01	.465	.0278	4.42	7.41	3417.6	1.02	32381	10.8	-.38	-.27
178.0	47867.7	170995	2922.7	261.4	250.4	.400	.01	.459	.0275	4.35	7.45	3436.9	1.01	32379	10.8	.12	.14
179.0	48305.5	170990	2980.0	261.4	250.2	.400	.01	.457	.0274	4.32	7.46	3442.1	1.00	32378	10.8	.12	.1
179.9	48699.5	170984	3031.7	261.4	250.0	.400	.01	.458	.0275	4.34	7.47	3443.3	1.00	32376	10.8	.12	.11

END OF TAKEOFF

```
** INPUTS TO LANDING - GROSS WT. =150000.0   WING AREA =1720.0   WING LOADING = 87.2
   ABAR= .35  CDGEAR= .026  DADT=1.0  DTABS= 0.0  DTDTMX= 7.0  DTPENG=3.0  FAAFTR= .60  HAPP=35.0  HAPT=   0.0
   HSPOIL=10.0  IPOWER= 0  ISPOIL= 1  PWRIDL= .10  PWRMAX= 1.00  PWRMRG= .05  SINKTD= 3.0  TDELAY=1.0
   THEMAX= 20.0  TBRK= 0.0  TFLP= 99.0  TOFF= 3.0  TREV= 0.0  TSPL=  2.0  XMUBRK= .45  XLFMAX=1.15  XDIST=5000.0
```

LANDING

VKAPP =130.0 GAMAPP = 3.00 RTSNK = 689.9 SINKTD = 3.00 XLFMAX = 1.15 ALPHMX =20.00

DELSPL = 0.00 DELFD =40.00 ALPHA = 7.33 THRUST = 5825.1 IPOWER = 0 ISPOIL = 1

APPROACH AT THROTTLE SETTING = .459

TIME (SEC)	DIST. (FEET)	WEIGHT (LBS)	ALT. (FEET)	TAS (KTS)	EAS (KTS)	MACH NO.	ACCEL (FPS2)	CL	CD	ALPHA (DEG)	GAMMA (DEG)	R/C (FPM)	LOAD FACT	THRUST (LBS)	FUS. ANG.	THETA DOT
0.0	0.0	150000	35.0	130.0	129.9	.195	0.00	1.702	.2666	7.33	-3.00	689.9	1.00	17475	3.3	0.000
.5	109.7	150000	29.3	130.0	129.9	.195	0.00	1.702	.2666	7.33	-3.00	689.9	1.00	17475	3.3	0.000
1.0	219.4	150000	23.5	130.0	129.9	.195	0.00	1.702	.2666	7.33	-3.00	689.9	1.00	17475	3.3	0.000
1.5	329.1	150000	17.8	130.0	129.9	.195	0.00	1.702	.2666	7.33	-3.00	689.9	1.00	17475	3.3	0.000
** BEGIN FLARE AT 12.9 FEET																
1.9	422.6	150000	12.9	130.0	130.0	.197	-.24	1.702	.2666	7.33	-3.00	689.8	1.00	17475	3.3	0.000
2.0	444.6	150000	11.7	130.0	130.0	.197	-.35	1.717	.2687	7.53	-2.88	662.0	1.15	17475	3.7	3.222
2.1	466.5	150000	10.6	130.0	129.9	.197	-.43	1.721	.2692	7.58	-2.76	633.4	1.15	17475	3.8	1.741
2.2	488.5	150000	9.6	129.9	129.9	.197	-.50	1.721	.2692	7.58	-2.63	604.8	1.15	17475	4.0	1.237
2.3	510.4	149999	8.6	129.9	129.9	.197	-.58	1.725	.2698	7.63	-2.51	576.1	1.15	17475	4.1	1.754
2.4	532.3	149999	7.7	129.9	129.8	.197	-.64	1.725	.2698	7.63	-2.38	547.1	1.15	17475	4.3	1.249
2.5	554.2	149999	6.8	129.8	129.8	.196	-.71	1.725	.2698	7.63	-2.26	518.4	1.15	17475	4.4	1.243
2.6	576.2	149999	6.0	129.8	129.8	.196	-.79	1.729	.2703	7.68	-2.13	489.6	1.15	17475	4.6	1.759
2.7	598.1	149999	5.2	129.7	129.7	.196	-.86	1.729	.2703	7.68	-2.01	460.5	1.15	17475	4.7	1.252
2.8	620.0	149999	4.4	129.7	129.7	.196	-.92	1.729	.2703	7.68	-1.88	431.7	1.15	17475	4.8	1.244
2.9	641.9	149999	3.7	129.6	129.6	.196	-1.00	1.733	.2708	7.73	-1.76	402.9	1.15	17475	5.0	1.759
3.0	663.7	149998	3.1	129.6	129.6	.196	-1.06	1.733	.2708	7.73	-1.63	373.9	1.15	17475	5.1	1.250
3.1	685.6	149998	2.5	129.5	129.5	.196	-1.14	1.737	.2714	7.78	-1.51	345.0	1.15	17475	5.3	1.763
3.2	707.5	149998	1.9	129.4	129.4	.196	-1.20	1.737	.2714	7.78	-1.38	315.9	1.15	17475	5.4	1.253
3.3	729.3	149998	1.4	129.4	129.3	.196	-1.28	1.741	.2719	7.83	-1.25	287.0	1.15	17475	5.6	1.765
3.4	751.2	149998	1.0	129.3	129.3	.196	-1.34	1.741	.2719	7.83	-1.13	257.9	1.15	17475	5.7	1.254
3.5	773.0	149998	.6	129.2	129.2	.196	-1.42	1.745	.2724	7.88	-1.00	229.1	1.15	17475	5.9	1.765
3.6	794.8	149998	.2	129.1	129.1	.195	-1.48	1.745	.2724	7.88	-.88	200.1	1.15	17475	6.0	1.253
** TOUCHDOWN																
3.7	816.6	149997	0.0	129.0	129.0	.195	-1.56	1.749	.2730	7.93	0.00	180.0	0.00	17475	6.9	1.763
4.7	1033.8	149996	0.0	128.2	128.2	.194	-1.03	1.356	.2283	2.93	0.00	0.0	0.00	17498	1.9	-5.000
APPLY BRAKES AND/OR SPOILERS AND/OR REVERSE THRUST - MUBRK = .450																
5.7	1248.7	149995	0.0	126.1	126.1	.191	-4.21	1.204	.2160	1.00	0.00	0.0	0.00	17525	0.0	0.000
6.7	1459.5	149993	0.0	123.5	123.5	.187	-4.46	1.204	.2160	1.00	0.00	0.0	0.00	17560	0.0	0.000
7.7	1665.1	149992	0.0	119.7	119.7	.182	-8.75	.925	.3086	1.00	0.00	0.0	0.00	17609	0.0	0.000
8.7	1862.3	149991	0.0	114.1	114.1	.174	-10.11	.894	.3195	1.00	0.00	0.0	0.00	13536	0.0	0.000
9.7	2049.3	149990	0.0	107.8	107.8	.164	-11.24	.894	.3195	1.00	0.00	0.0	0.00	8970	0.0	0.000
10.7	2225.0	149990	0.0	100.8	100.8	.154	-12.39	.894	.3195	1.00	0.00	0.0	0.00	4361	0.0	0.000
11.7	2388.3	149990	0.0	93.3	93.3	.142	-12.74	.894	.3195	1.00	0.00	0.0	0.00	3918	0.0	0.000
12.7	2539.0	149990	0.0	85.7	85.7	.131	-12.87	.894	.3195	1.00	0.00	0.0	0.00	3941	0.0	0.000
13.7	2676.7	149990	0.0	78.1	78.1	.119	-13.00	.894	.3195	1.00	0.00	0.0	0.00	3964	0.0	0.000
14.7	2801.5	149990	0.0	70.3	70.3	.108	-13.11	.894	.3195	1.00	0.00	0.0	0.00	3987	0.0	0.000
15.7	2913.1	149990	0.0	62.5	62.5	.096	-13.22	.894	.3195	1.00	0.00	0.0	0.00	4010	0.0	0.000
16.7	3011.5	149990	0.0	54.7	54.7	.084	-13.31	.894	.3195	1.00	0.00	0.0	0.00	4034	0.0	0.000
17.7	3096.6	149989	0.0	46.8	46.8	.072	-13.39	.894	.3195	1.00	0.00	0.0	0.00	4057	0.0	0.000
18.7	3168.3	149989	0.0	38.9	38.9	.060	-13.45	.894	.3195	1.00	0.00	0.0	0.00	4081	0.0	0.000
19.7	3226.5	149989	0.0	30.9	30.9	.048	-13.50	.894	.3195	1.00	0.00	0.0	0.00	4105	0.0	0.000
20.7	3271.3	149989	0.0	22.9	22.9	.036	-13.54	.894	.3195	1.00	0.00	0.0	0.00	4129	0.0	0.000
21.7	3302.5	149989	0.0	14.9	14.9	.024	-13.57	.894	.3195	1.00	0.00	0.0	0.00	4153	0.0	0.000
22.7	3320.1	149989	0.0	6.8	6.8	.012	-13.58	.894	.3195	1.00	0.00	0.0	0.00	4177	0.0	0.000
22.9	3324.4	149989	0.0	0.0	0.0	0.000	-13.58	.894	.3195	1.00	0.00	0.0	0.00	0	0.0	0.000

LANDING DISTANCE = 3324.4 FEET. FIELD LENGTH = 5540.7 FEET. ABAR = .342

```
A19NB06 .  7600 DAYFILE SUMMARY
73/07/04. ** 1.1.6  BKY6    ** 73/06/18
15.50.41.A19NBXX,P4,T50,CM70000.                        BOWLES      029
15.50.41.RUN76,S.
15.50.42.           COMPILING  TEST1
15.50.42. LCM       40000
15.50.42. RUN76 COMPILE FL    044000
15.50.42. TIME 000.065 SEC.  BKY RUN76 LEVEL 5C
15.50.42.ATTACH,A,BJEFF.
15.50.42. LCM       100000
15.50.43. LCM       140000
15.50.43. LCM       200000
15.50.43. LCM       240000
15.50.44. FILE ATTACHED.
15.50.44.ATTACH,B,SJEFF.
15.50.44. FILE ATTACHED.
15.50.44.COPY,A/RXP,1FM,B/RXB,1F,LGO.
15.50.45.MAP,OFF.
15.50.45.LINK,X.
15.50.45. SCM       34000
15.50.47.END TEST1
15.50.47. ACCOUNTING INFORMATION.
15.50.47. SCM KWS     38.98398
15.50.47. LCM KWS    123.61856
15.50.47. TRACKS             3
15.50.47. I/O REQ.         326
15.50.47. RMS CALLS         25
15.50.47. LCM BLD.           9
15.50.47. SWAPS             28
15.50.47. CPU TIME       2.393 SECONDS.
15.50.47. 73/07/04.         20 CDS.
```

- END OF SAMPLE CASE -

APPENDIX C

Program Listing

```
TAKOFF
            SUBROUTINE TAKOFF(INPC,IDCN,WGROSS,SWING,XENG,V1,VR,VEND)
      C
      C   SUBROUTINE TAKOFF COMPUTES THE TAKEOFF MANEUVER OF A GIVEN AIRCRAFT,
      C   INCLUDING GROUND ROLL AND CLIMBOUT.
      C
      C   PROVISIONS ARE MADE FOR CHANGES IN FLAP,VECTORED THRUST ANGLE AND POWER
      C   SETTINGS AS FUNCTIONS OF SPEED AND ALTITUDE.
      C
      C   THE ENGINE-OUT TAKEOFF IS DONE AS FOLLOWS - ACCELERATION TO SPEED V1,
      C   A LINEAR DECREASE WITH TIME IN THE NUMBER OF ENGINES FROM A VALUE OF
      C   ENP TO A VALUE OF ENP-1,CONTINUE GROUND ROLL TO VR,ROTATE,LIFTOFF AND
      C   CLIMBOUT TO 250 FT.
      C
      C   FOLLOWING SOME COMMENT CARDS WILL BE TWO NUMBERS IN PARENTHESIS WHICH
      C   WILL GIVE THE APPROXIMATE STATEMENT-NUMBER RANGE OF THE FUNCTION
      C   DESCRIBED IN THAT COMMENT.
      C
            EXTERNAL DERIV1
            EXTERNAL DERIV3
            REAL MU
            COMMON /UNIV/ NPC    ,NSC    ,IDC    ,M      ,ST     ,R      ,W      ,
           1WF    ,EM     ,VMO    ,EMMO   ,ALPHL0,CLALPH,SW     ,AR     ,B      ,
           2EYEW   ,ENP    ,TA     ,WG     ,WGS    ,KWRITE,DLMC4
           3,KSIZE
            COMMON /AERO/ DEL,QS,HABS,THRUST,TVECT, ANGLE,DELFD,DELSPL,ALPHA,
           9CX,CY,CL,CD,RMU,GRCD,IFAST
            COMMON /EXCHNG/ SROLL,S35,V35,T5J
            COMMON /XROLL/T(30),NEQ,MU,NREV
            COMMON /XFLATE/ S(52),HOL,THEMAX,THETAF,XLF,XLFMAX,K
            DIMENSION ANS(8)
            DIMENSION XDELFD(5),XMFLAP(5),XVFLAP(5)
            DIMENSION XHPWR(5),XVPWR(5),XPOWER(5)
            DIMENSION XNU(5),XHVECT(5),XVVECT(5)
            YYY(XXX,Y1,Y2,X1,X2) = Y1 + (Y2-Y1)*(XXX-X1)/(X2-X1)
            DATA XDELFD/15.0,5.0,2.0,0.0,0.0/, XMFLAP/0.0,250.,0,0,0,0,0.0/,
           9XVFLAP/0.0,0.0,200.,210.,0.0/, XPOWER/1.0,1.0,1.0,1.0,1.0/, XHPWR/
           90.0,0.0,0.0,0.0,0.0,0.0/, XVPWR/0.0,999.,999.,999.,999./, XNU/0.0,0.0,
           90.0,0.0,0.0,0.0/, XHVECT/0.0,0.0,0.0,0.0,0.0,0.0/, XVVECT/0.0,999.,999.,
           9999.,999./
            DATA COGEAR,DFLPDT,DTABS,DTGR,DTPDWN,DTPUP,DTVECT,EYEWNG,HAPT,
           9HGR,HMAN,IOUT,UM,PMARG,RTCL,TMTFLY,TMTSCP/0.0,3.0,0.0,0.0,5.0,5.0,6.0,
           910.,1.0,0.0,25.0,1000.,0.0,0.02,0.04,750.,15.,10.0/
            DATA DTFAIL/3.0/
            NAMELIST/NAM1/COGEAR,DADT,DFLPDT,DTABS,DTGR,DTPDWN,DTPUP,DTVECT,
           9EYEWNG,HAPT,HGR,HMAN,HMAX,IOUT,UM,NPAGE,PMARG,RTCL,TMTFLY,
           9TMTSCP,XLFMAX,DTFAIL
            NAMELIST/NAM2/XDELFD,XMFLAP,XVFLAP,XPOWER,XHPWR,XVPWR,XNU,XHVECT,
           9XVVECT
            GO TO(5,0,5),INPC
   37     5 NPAGE = 33
   40       DADT = 1.0
   41       HMAX = 5000.
   43       XLFMAX = 1.10
   45       READ(5,NAM1)
      C
      C   CALL INPUT TO TAKOFF THRU NAMELIST /NAM1/ AND /NAM2/
```

```
TAKOFF
      C
     47         READ(5,*AM2)
     55         IF(INPC .EQ. 1)RETURN
     63       6 ENP = XENG
     64         SW = SWING
     65         W = WGROSS
     66         EYEW = EYEWNG
     67         MU = UM
     71         S(7) = HAPT
     73         DO 7 I = 2,4
     74         IF(XMP*R(I) .NE. 0.0)XVP*P(I) = 0.0
    100         IF(XMVECT(I) .NE. 0.0)XVVECT(I) = 0.0
    104       7 CONTINUE
      C
      C    SET UP LOGIC CONTROL VARIABLES
      C
    106         NEG = 2
    107         IPAGE = 0
    110         KENG = 0
    110         JJ1 = 1
    111         JJ2 = 1
    112         IUP = 1
    113         IDOWN = 1
    113         IFLY = 1
    114         IFLAP = 2
    115         MFLAP = 1
    116         JROUTE = 1
    116         IPOWER = 1
    117         MPOWER = 2
    120         MVECT = 1
    120         IVECT = 2
    121         I35 = 1
    122         IG = 0
    123         NOUT = 0
    123         ENPJ = ENP
    125         ENPOUT = ENP - 1.0
      C
      C    FIND GEAR DRAG IF DEFAULT OF CDGEAR=0.0 USED
      C
    126         IF(CDGEAR .EQ. 0.0)CDGEAR = (0.0032/SW)*0.**0.80
    136         GMCD = CDGEAR
    137         IF(IDEN .NE. 9)GO TO 299
    141         TEMP = 59.0 + DTABS
    143         WG = WGROSS
    144         WOS = W/SW
      C
      C    FIND STATIC THRUST/WEIGHT
      C
    146         CALL ENGINE(0.0,0.0,0.0,0.0,1.0,WF,KENG)
    152         TOA = ENP * THRUST / W
      C
      C    WRITE OUT PROGRAM INPUTS
      C
    155         WRITE(6,201)HAPT,TEMP
        201   FORMAT(//, 3OH ** INPUTS TO TAKE OFF - ALTITUDE = ,F6.1,2X,14HTEMP
           9ERATURE = ,F5.1,7H DEG. F)
```

```
TAKOFF
    174       WRITE(6,202)WG,SW,THRUST
          202 FORMAT(/,6X,19HA/C CHARACTERISTICS,/,9X,17HGROSS RAMP WT. = ,F8.0,
             93X,12HWING AREA = ,F6.0,3X,26HSTATIC SEA LEVEL THRUST = ,F6.0)
    211       WRITE(6,203)WOS,TOW
          203 FORMAT(9X,15HWING LOADING = ,F5.1,3X,16HTHRUST/WEIGHT = ,F4.3)
    224       WRITE(6,204)ENP,CDGEAR,EYEW,THTSCP
          204 FORMAT(/,6X, 15HA/C PARAMETERS,/,9X,14HNO. ENGINES = ,F3.1,3X, 9HC
             9DGEAR = ,F6.4,3X,9HEYEWNG = ,F4.1,3X,20HTAIL SCRAPE ANGLE = ,F4.1)
    243       WRITE(6,205)XLFMAX,HGR,THTFLY,HMAN,RTCL
          205 FORMAT(/,6X,30HFLIGHT PATH CONTROL PARAMETERS,/,9X,18HMAX LOAD FAC
             9TOR = ,F4.2,3X,23HGEAR RETRACTION ALT. = ,F5.1,3X,18HMAX FLOOR ANG
             9LE = ,F4.1,/,9X,16HMANEUVER ALT. = ,F5.0,5X,27HACCELERATE RATE OF
             9CLIMB = ,F5.0)
    264       WRITE(6,206)DADT,DFLPDT,DTGR,DTPDWN,DTPUP,DTVECT,DTFAIL
          206 FORMAT(/,6X,25HPARAMETER VARIATION RATES,/,9X, 7HDADT = ,F4.1,3X
             9,9HDFLPDT = ,F4.1,3X,7HDTGR = ,F4.1,3X,9HDTPDWN = ,F4.1,/,9X,8HDTO
             9UP = ,F4.1,3X,9HDTVECT = ,F4.1,3X,9HDTFAIL = ,F4.1)
    311       WRITE(6,207)
          207 FORMAT(//,6X,42HPOWER, VECTORED THRUST, AND FLAP SCHEDULES)
    320       WRITE(6,208)(XPOWER(I),I = 1,5)
          208 FORMAT(/,9X,22HTHROTTLE/POWER SETTING,/,12X,8HPWRSET  ,5F9.2)
    331       WRITE(6,209)(XVPWR(I),I= 1,5)
          209 FORMAT(12X,8HSPEED    ,5F9.1)
    342       WRITE(6,210)(XHPWR(I),I = 1,5)
          210 FORMAT(12X,8HALTITUDE,5F9.0)
    353       WRITE(6,211)(XNU(I),I = 1,5)
          211 FORMAT(/,9X,21HVECTORED THRUST ANGLE,/,12X,8HANGLE    ,5F9.1)
    364       WRITE(6,209)(XVVECT(I),I = 1,5)
    371       WRITE(6,210)(XHVECT(I),I = 1,5)
    402       WRITE(6,212)(XDELFD(I),I = 1,5)
          212 FORMAT(/,9X,21HFLAP DEFLECTION ANGLE,/,12X,8HDELFD    ,5F9.1)
    417       WRITE(6,209)(XVFLAP(I),I = 1,5)
    424       WRITE(6,210)(XHFLAP(I),I = 1,5)
    435       WRITE(6,213)
          213 FORMAT(/,9X,76HALL SPEEDS ARE INDICATED AIR SPEEDS AND ALL ALTITUD
             9ES ARE ABSOLUTE ALTITUDES)
    450   299 CONTINUE
    450       IF(IDCN .EQ. 9)
             +WRITE(6,999)HAPT
          999 FORMAT(1H1,//21H TAKEOFF (ELEVATION =,F6.0,4H FT),//)
    463       IF(IDCN .EQ. 9)
             +WRITE(6,1000)
          C
          C   SET FLAPS, VECTORED THRUST ANGLE AND POWER SETTING FOR GROUND RUN
          C
    474       PWRSET = XPOWER(1)
    475       ANGLE = XNU(1)
    477       DELFD = XDELFD(1)
    500       THEMAX = THTSCP
    502       HABS = 0.0
    502       VEL = 0.
    503       ZERO = 0.
    504       THETAF = 0.
    504       QS = 0.1
    506       S(7) = HAPT
    507       EM = 0.
```

```
TAKOFF
      C    GROUND ROLL INTEGRATION VARIABLES
      C      T(1) = NUMBER OF EQUATIONS
      C      T(2) = TIME (SEC.)
      C      T(3) = TIME INTERVAL, STEP SIZE (SEC.)
      C      T(4) = VELOCITY (FT./SEC.)
      C      T(5) = DISTANCE (FT.)
      C      T(6) = ACCELERATION (FT./SEC.**2)
      C
  510         T(1) = 2
  511         T(2) = 0.0
  512         T(3) = 0.1
  513         T(4) = 0.
  513         T(5) = 0.
      C
      C
      C    OBTAIN ATMOSPHERIC VARIABLES
      C
  515         CALL ATMOS(HAPT,DTABS,ANS)
  517         SA = ANS(4)*1.68781
  521         RHO = .0023769*ANS(7)
  523         NCOUNT = 0
  523         ALPHA = EYE*
      C
      C    FIND INITIAL CONDITIONS OF THRUST AND FORCE COEFF. FOR GROUND ROLL
      C
  525         CALL ENGINE(S(7),DTABS,EM,FWRSET,WF,KENG)
  533         IFAST = 0
  534         CALL AERODYN
  535         CALL INTS(T,2,2,1.,1.,1.,1.,1.,1.,1., DERIV1)
  547         TA = THRUST * EAP
  551         IF(IDCN .EQ. 9)
             +WRITE(6,1002)T(2),T(5),W,ZERO,ZERO,ZERO,ZERO,T(6),CL,CD,ALPHA,
             4ZERO,ZERO,ZERO,TA,ZERO,ZERO,ZERO
      C
      C    MAIN GROUND ROLL INTEGRATION LOOP (I=120)
      C
  635     1 NCOUNT = NCOUNT + 1
  637         ALPHAJ = ALPHA
  640         THETAJ = THETAF
      C
      C    CHECK IF LIFT = WEIGHT - LIFTOFF
      C
  641         IF(QS*CY .GE. W)GO TO 120
      C
      C    CHECK SPEED FOR V1
      C
  645         IF(T(4)*.592087*SQRT(ANS(7)) .GE. V1 .AND. JJ1.EQ.1)GO TO 105
      C
      C    CHECK SPEED FOR VR
      C
  665   106 IF(T(4)*.592087*SQRT(ANS(7)) .GE. VR)GO TO 101
  700         GO TO 110
      C
      C    ENGINE FAILURE (105-101)
      C
  701   105 ALPHJ = ALPHA
```

```
TAKOFF
    702         T2J = T(2)
    704         T4J = T(4)
    705         T5J = T(5)
    707         WJ = W
    710         JJ1 = 2
    711         IF(IOUT .EQ. 1)HOUT = 1
    715         IF(IOUT .EQ. 1)HMAX = 250.
    720         VOUT = T(4) * 0.592087
    722         IF(IOUT.EQ. 1 .AND. IDCN .EQ. 9)WRITE(6,1020)VOUT,V1
   1020 FORMAT(5X,23HENGINE FAILURE AT TAS =,F7.1,13H KNOTS  EAS =,F7.1,
       9&H KNOTS)
    744         GO TO 106
              C
              C   BEGIN ROTATION
              C
    745   101 IF(JJ2 .EQ. 2)GO TO 103
    747         JJ2 = 2
    750         VKTS = T(4) * 0.592087
    752         EAS = VKTS*SQRT(ANS(7))
    756         IF(IDCN .EQ. 9)WRITE(6,1009)T(2),VKTS,EAS
   1009 FORMAT(1X,18HROTATION  (TIME = ,F6.1,3X,10HAND TAS = ,F7.1,2X,
       9&HEAS = ,F7.1,1H))
              C
              C   INCREASE ALPHA AT RATE DADT AND RESTRICT TO VALUE .LE. TAIL SCRAPE
              C   ANGLE (103-102)
              C
   1002   103 ALPHA = ALPHA + T(5)*DADT
   1005         IF((ALPHA - EYEW) .LE. THEMAX)GO TO 102
   1010         ALPHA = THEMAX + EYEW
   1011         GO TO 110
   1011   102 THETAF = ALPHA - EYEW
   1013   110 EM = T(4)/SA
   1015         QS = 0.5*RHO*SW*T(4)*T(4)
   1021         CALL ENGINE(S(7),DTABS,EM,FWRSET,WF,KENG)
   1026         IFAST = 0
   1027         VEL = T(4)
              C
              C   ENGINE FAILURE CONTROL ( =111)
              C
   1030         IF(HOUT .EQ. 0)GO TO 111
   1035         IF(DTFAIL .EQ. 0.0)GO TO 112
   1036         ENP = ENPJ - (T(2) - T2J)/DTFAIL
   1042         IF(ENP .LT. ENPOUT)ENP = ENPOUT
   1045         GO TO 111
   1046   112 ENP = ENPOUT
   1050   111 CONTINUE
   1050         TA = ENP*THRUST
   1052         CALL INTM(T,2,2,1.,1.,1.,1.,1.,1.,1., DERIV1)
              C
              C   MAKE INTEGRATION STEP
              C
   1064         W = W - WF*ENP*T(3)/3600.
   1070         IF(T(2) .LE. 90.)GO TO 113
   1077         WRITE(6,1057)
   1057 FORMAT(/,35H TIME LIMIT FOR GROUND RUN EXCEEDED)
   1106         RETURN
```

```
TAKOFF
   1106    115 CONTINUE
C
C   COMPUTE ALPHA DOT AND THETA DOT
C
   1106        XDADT = (ALPHA - ALPHAJ)/T(3)
   1111        DTHTDT = (THETAF - THETAJ)/T(3)
   1113        IF(NCOUNT .LT. 10)GO TO 1
   1116        NCOUNT = 0
   1116        VKTS = T(4) * 0.592087
   1121        EAS = VKTS * SQRT(ANS(7))
   1125        IF(IDCN .NE.9)GO TO 1
   1133        IF(IDCN .EQ. 9)
              +WRITE(6,1002)T(2),T(5),W,ZERO,VKTS,EAS,EM,T(6),CL,CD,ALPHA,ZERO,
              9ZERO,ZERO,TA,THETAF,XDADT,DTHTDT
   1212        IPAGE = IPAGE +1
   1214        IF(IPAGE .LT. NPAGE)GO TO 1
   1216        IPAGE = 0
   1216        IF(IDCN .EQ. 9)
              +WRITE(6,998)
    998 FORMAT(1H1,/,19H TAKEOFF CONTINUED ,/)
   1227        IF(IDCN .EQ. 9)
              +WRITE(6,1000)
   1240        GO TO 1
   1241    120 VKTO = T(4) * 0.592087
   1243        EASTO = VKTO * SQRT(ANS(7))
   1247        SROLL = T(5)
   1251        IF(IOUT .EQ. 1)ENP = ENPOUT
   1260        IF(IDCN .EQ. 9)WRITE(6,1010)T(2),T(5),VKTO,EASTO
   1010 FORMAT(1X,17HLIFTOFF  (TIME = ,F6.1,2X,7HDIST = ,F8.1,2X
              9,6HTAS = ,F7.1,1X,6HEAS = ,F7.1,1H))
   1000 FORMAT(131H  TIME     DIST.    WEIGHT    ALT.     TAS    EAS    MACH  A
              9CCEL     CL      CD    ALPHA  GAMMA     R/C    LOAD    THRUST  FUS.  DAL
              9PH    DTHTA,/,130H (SEC)    ( FET)     (LBS)    (FEET)  (KTS) (KTS)
              9NO. (FPS2)                        (DEG)  (DEG)    (FPM)  FACT     (LBS)  ANG
              9.    /DT      /DT ,/)
   1002 FORMAT(1X,F5.1,F9.1,F9.1,F9.0,F9.1,F7.1,F6.1,F6.3,F7.2,F7.3,F7.4,2F7.2,
              9F8.1,F6.2,F9.0,F6.1,F7.2,F8.2)
C
C END OF GROUND ROLL - BEGIN AIRBORNE PORTION OF TAKEOFF
C
   1303        VMARG = PHANG*VEND
   1305        THEMAX = THIFLY
C
C   FLAP, ANGLE AND POWER SCHEDULES SET UP FOR AIRBORNE PORTION
C
   1306        NFLAP = XNFLAP(2)
   1310        VFLAP = XVFLAP(2)
   1311        NPOWER = XNPWR(2)
   1313        VPOWER = XVPWR(2)
   1314        NVECT = XNVECT(2)
   1316        VVECT = XVVECT(2)
   1317        XLF = 1.0
C
C   AIRBORNE INTERGRATION VARIABLES
C     S(1) = NUMBER OF EQUATIONS
C     S(2) = TIME (SEC.)
```

```
TAKOFF
C       S(3) = TIME INTERVAL,STEP SIZE (SEC.)
C       S(4) = VELOCITY ALONG FLIGHT PATH (FT./SEC.)
C       S(5) = FLIGHT PATH ANGLE (RAD.)
C       S(6) = DISTANCE ALONG FLIGHT PATH (FT.)
C       S(7) = ALTITUDE (FT.)
C       S(8) = ACCELERATION ALONG FLIGHT PATH (FT./SEC.**2)
C       S(9) = TIME RATE OF CHANGE OF FLIGHT PATH ANGLE (RAD./SEC.)
C       S(10) = HORIZONTAL SPEED (FT./SEC.)
C       S(11) = RATE OF CLIMB (FT./SEC.)
C
1321        S(1) = 4
1322        S(4) = T(4)
1324        S(2) = T(2)
1325        S(6) = T(5)
1327        S(5) = 0.
1327        S(3) = 0.1
1331        CALL INTS(S,4,2,1.,1.,1.,1.,1., DERIV3)
1343        GO TO 300
C
C   MAIN AIRBORNE INTEGRATION LOOP (2=8)
C
1347    2 NCOUNT = NCOUNT + 1
1351        ALPHAJ = ALPHA
1352        THETAJ = THETAP
1353        IF((S(7) - HAPT) .GE. HMAX)RETURN
C
C   CHECK PROGRAM PROTECTION LIMITS
C
1360        IF(T(5) .GT. 60760.)GO TO 9
1364        IF(S(7) .LT. -0.1)GO TO 9
1366        IF(S(2) .GT. 300.)GO TO 9
C
C   OBTAIN ATMOSPHERIC VARIABLES
C
1372  300 CALL ATMOS(S(7),DTAMS,ANS)
1376        SA = ANS(4)*1.68781
1400        RHO = .0023769*ANS(7)
1402        EM = S(4)/SA
1403        HAMS = S(7) - HAPT
C
C   CALL ENGINE WITH PWRSET (KENG=0)
C
1406        CALL ENGINE(S(7),DTAMS,EM,PWRSET,WF,KENG)
1413        W = W - WF*ENP*S(3)/3600.
C
C   BEGIN AERODYNAMIC CONTROL
C
1417        K = 1
C
C   CHECK FOR START OF PULLUP MANEUVER
C
1420        IF ((VEAS -S(4)*SQRT(ANS(7))*.592087).LT. VMARG)GO TO 27
1440        IF((S(7) - HAPT) .LT. HMAN)GO TO 4
1444        IF(IFLY .EQ. 2)GO TO 28
1446        IFLY = 2
1446        IF(RTCL .GT. ROC)GO TO 9
```

```
TAKOFF
   1452         IF(IDCN .EQ. 9)
              C
              C    ACCELERATION TO VEND AT CONSTANT RATE OF CLIMB (28-27)
              C
                   WRITE(6,1040)VEND
             1040 FORMAT(1X,29HACCELERATE TO CLIMB SPEED OF ,F6.1,6H KNOTS)
   1465      28 IF(S(4)*SIN(S(5))*60.0 .LE. RTCL + 10.)GO TO 29
              C
              C    REDUCE ALPHA TO START ACCELERATION PHASE
              C
   1501         ALPHA = ALPHA - DADT*S(3)*0.5
   1504         IF(XLF .LT. 0.85)ALPHA = ALPHA + DADT*S(3)*0.25
   1512         GO TO 26
   1513      29 K = 9
   1514         ROC = RTCL
   1516         GO TO 26
   1516      27 IF(JROUTE .EQ. 2)GO TO 41
   1520         ALPMAX = 1.2
              C
              C    PULLUP MANEUVER - FIND REQUIRED DADT (27-44)
              C
   1522         CALL PULLUP(DADT,KODE,PWRSET,KENG,VEND,HAPT,DTABS)
   1531         IF(KODE .EQ. 9)GO TO 9
   1537         IF(IDCN .EQ. 9)WRITE(6,1056)DADT
             1056 FORMAT(24H EXECUTE PULLUP AT DADT = ,F4.2)
   1551         JROUTE = 2
   1552         GO TO 4
   1553      41 GO TO(42,43,44),KODE
   1562      42 IF(S(4)*SQRT(ANS(7))*0.592087 .LT. VEND .AND. S(8) .LT. .02)GO TO
                96
   1603         GO TO 4
   1603      43 IF(S(4)*SQRT(ANS(7))*0.592087.GE. VEND )GO TO 8
   1616         GO TO 8
   1617      44 IF(S(8) .LT. 0.02)GO TO 8
              C
              C    INCREASE ANGLE OF ATTACK EACH TIME BEFORE INTEGRATION STEP TAKEN
              C    (EXCEPT FOR CONSTANT RATE OF CLIMB PORTION).  FOR S(9) NEGATIVE(XLF
              C    .LT. 1.0), ADDITIONAL INCREASE IN ALPHA.  IF THE INCREASE IN ALPHA
              C    RESULTS IN ANY VIOLATION OF FLIGHT PATH CONSTRAINTS(ACCEL,,XLF,THETAF),
              C    ALPHA WILL BE REDUCED ACCORDINGLY IN DERIV2.
              C
   1622       4 ALPHA = ALPHA + DADT*S(3)
   1625         IF(XLF .LT. 0.9)ALPHA = ALPHA + DADT*S(3)
   1631         IF(XLF .LT. 0.8)ALPHA = ALPHA + DADT*S(3)
   1636      26 CONTINUE
              C
              C    MAKE INTEGRATION STEP
              C
   1638       5 CALL INTM(S,4,2,1.,1.,1.,1.,1.,1., DERIV3)
   1650         IF(K .EQ. 99)GO TO 9
   1656         T(5) = T(5) + S(3)*S(10)
   1660         IF(I35 .EQ. 2)GO TO 22
              C
              C    SAVE VALUES FOR 35 FT. INTERPOLATION
              C
   1662         IF((S(7) - HAPT) .GE. 35.)GO TO 21
```

```
TAKOFF
   1666          VJ = S(4)
   1667          TJS = T(5)
   1670          HJ = S(7)
   1672          GO TO 22
                 C
                 C    FIND VALUES AT 35 FT.(21-1015)
                 C
   1672       21 I35 = 2
   1673          S35 = YYY(35.,TJS,T(5),HJ,S(7))
   1703          V35 = YYY(35.,VJ,S(4),HJ,S(7))*SQRT(ANS(7))*0.592087
   1722          IF(IDCN .EQ. 9)
                 *WRITE(6,1015 S35,V35
                1015 FORMAT(20H DISTANCE TO 35 FT.=,F7.1,21H  SCREEN SPEED(EAS) =,F6.1,
                 9H KNOTS)
   1743       22 IF(IG .EQ. 2)GO TO 25
   1745          IF(IG .EQ. 1)GO TO 23
                 C
                 C    GEAR RETRACTION (22-23)
                 C
   1747          IF((S(7) - HAPT) .LT. HGR)GO TO 25
   1752          TG = S(2)
   1753          TGU = S(2) + DTGR
   1755          IF(IDCN .EQ. 9)
                 *WRITE(6,1025)TG,TGU
                1025 FORMAT(1X,27HGEAR RETRACTION STARTED AT ,F6.1,17H SEC,COMPLETE AT
                 9,F6.1,4H SEC)
   1772          IG = 1
                 C
                 C    GEAR DRAG INCREMENT REDUCED LINEARLY WITH TIME IN DTGR SECONDS
                 C
   1773       23 GRCD =CDGEAR*(1.0 - (S(2) - TG)/DTGR)
   1777          IF(GRCD .GE. 0.0)GO TO 25
   2001          GRCD = 0.0
   2002          IG = 2
   2003       25 CONTINUE
   2003          IF(JROUTE .EQ. 2)GO TO 39
                 C
                 C    FLAP RETRACTION (25-16)
                 C
   2005          IF(DELFD .EQ. 0.0)GO TO 16
   2006          GO TO(10,15),MFLAP
   2014       10 IF((S(7)- HAPT) .LT. HFLAP .OR. S(4)*SQRT(ANS(7))*.592087 .LT.
                 9 VFLAP)GO TO 16
   2036          MFLAP = 2
   2036          TIME =(DFLFD - XDELFD(IFLAP))/DFLPDT
   2042          IF(IDCN .EQ. 9)
                 *WRITE(6,1030)XDELFD(IFLAP),TIME
                1030 FORMAT(1X,19HFLAPS RETRACTED TO ,F4.1,9H DEG. IN ,F4.1,5H SEC.)
   2061       15 DELFD = DELFD - DFLPDT*S(3)
   2064          IF(DELFD .GT. XDELFD(IFLAP))GO TO 16
   2070          DELFD = XDELFD(IFLAP)
   2072          MFLAP = 1
   2072          IFLAP = IFLAP + 1
   2074          HFLAP = XHFLAP(IFLAP)
   2076          VFLAP = AVFLAP(IFLAP)
                 C
```

```
TAKOFF
C     VECTORED THRUST ANGLE REDUCTION (16-56)
C
2101     16 CONTINUE
2101        IF(ANGLE .EQ. 0.0)GO TO 56
2102        GO TO(50,55),MVECT
2110     50 IF((S(7)-HAPT) .LT. HVECT .OR. S(4)*SQRT(ANS(7))*0.592087 .LT.
           9 VVECT)GO TO 56
2132        MVECT = 2
2132        TIME = (ANGLE - XNU(IVECT))/DTVECT
2136        IF(IDCN .EQ. 9)
           +WRITE(6,1055)XNU(IVECT),TIME
 1055 FORMAT(1X,33HVECTORED THRUST ANGLE REDUCED TO ,F4.1,9H DEG. IN ,
           9F4.1,5H SEC.)
2155     55 ANGLE = ANGLE - DTVECT*S(3)
2160        IF(ANGLE .GT. XNU(IVECT))GO TO 56
2164        ANGLE = XNU(IVECT)
2166        MVECT = 1
2166        IVECT = IVECT + 1
2170        HVECT = AMVECT(IVECT)
2172        VVECT = VVECT(IVECT)
C
C     THROTTLE SETTING MANAGEMENT (56-39)
C
2175     56 CONTINUE
2175        GO TO(31,32,33),IPOWER
2204     31 IF((S(7) - HAPT) .LT. HPOWER .OR. S(4)*SQRT(ANS(7))*.592087 .LT.
           9 VPOWER)GO TO 39
C
C     DETERMINE POWER INCREASE OR DECEASE
C
2220        IF(PWRSET - XPOWER(MPOWER))  ,39,35
C
C     ADVANCE THROTTLE SETTING LOOP (34-35)
C
2233     34 IPOWER = 2
2233        TIME = 100.*(XPOWER(MPOWER) - PWRSET)/DTPUP
2237        SET = XPOWER(MPOWER)*100.
2240        IF(IDCN .EQ. 9)
           +WRITE(6,1050)SET,TIME
 1050 FORMAT(1X,28HADVANCE THROTTLE SETTING TO ,F6.1,12H PERCENT IN ,
           9F4.1,5H SEC.)
2250     32 PWRSET = PWRSET + (DTPUP/100.)*S(3)
2262        IF(PWRSET .LT. XPOWER(MPOWER))GO TO 39
2265        PWRSET = XPOWER(MPOWER)
2267        MPOWER = MPOWER + 1
2271        HPOWER = XHPWR(MPOWER)
2273        VPOWER = XVPWR(MPOWER)
2275        IPOWER = 1
2276        GO TO 39
C
C     RETARD THROTTLE SETTING LOOP (35-39)
C
2301     35 IPOWER = 3
2301        TIME = 100.*(PWRSET - XPOWER(MPOWER))/DTPDWN
2305        SET = XPOWER(MPOWER)*100.
2306        IF(IDCN .EQ. 9)
```

```
TAKOFF
            +WRITE(6,1051)SET,TIME
         1051 FORMAT(1X,'THETARO THROTTLE SETTING TO ',F5.1,12H PERCENT IN ,F4.1
            9,5H SEC.)
    2324     33 PWRSET = PWRSET - (DTPOWR/100.)*S(3)
    2330        IF(PWRSET .GT. XPOWER(MPOWER))GO TO 39
    2334        PWRSET = XPOWER(MPOWER)
    2336        MPOWER = MPOWER + 1
    2337        MPOWER = XMPWR(MPOWER)
    2341        VPOWER = XVPWR(MPOWER)
    2344        IPOWER = 1
    2345     39 CONTINUE
    2345        IF(NCOUNT .LT.10)GO TO 2
    2350        NCOUNT = 0
    2350        XDADT = (ALPHA - ALPHAJ)/S(3)
    2353        DTHTDT = (THETAF - THETAJ)/S(3)
    2355        TA = ENP * THRUST
    2357        VKTS = S(4)*0.592087
    2362        EAS = VKTS*SQRT(ANS(7))
    2366        GAMMA = S(5)*57.295
    2370        HCC = S(4)*SIN(S(5))*60.
    2376        IF(IDCN .EQ. 9)
               +WRITE(6,1002)S(2),T(5),M,S(7),VKTS,EAS,EM,S(8),CL,CD,ALPHA,GAMMA
               9,HCC,XLF,TA,THETAF,XDADT,DTHTDT
    2403        IPAGE = IPAGE +1
    2405        IF(IPAGE .LT. NPAGE)GO TO 2
    2407        IPAGE = 0
    2407        IF(IDCN .EQ. 9)
               +WRITE(6,998)
    2500        IF(IDCN .EQ. 9)
               +WRITE(6,1000)
          C
          C   END OF MAIN AIRBORNE INTEGRATION LOOP.
          C
    2511        GO TO 2
          C
          C   CONVERSIONS AND PRINT OUT.
          C
    2512     6 TA = ENP * THRUST
    2514        VKTS = S(4)*0.592087
    2516        EAS = VKTS*SQRT(ANS(7))
    2522        IF(KODE .NE. 3)EAS = VEND
    2532        GAMMA = S(5)*57.295
    2534        HCC = S(4)*SIN(S(5))*60.
    2541        IF(IDCN .EQ. 9)
               +WRITE(6,1002)S(2),T(5),M,S(7),VKTS,EAS,EM,S(8),CL,CD,ALPHA,GAMMA
               9,HCC,XLF,TA,THETAF,XDADT,DTHTDT
    2626        WRITE(6,1052)
    2632        IF(KODE .EQ. 2)WRITE(6,1053)
    2647        IF(KO   .EQ. 3)WRITE(6,1054)
         1052 FORMAT (,1X,14HEND OF TAKEOFF)
         1053 FORMAT(1X,45HTHROTTLING REQUIRED TO MAINTAIN CONSTANT SPEED)
         1054 FORMAT(1X,54HDESIRED SPEED AND SPEED NOT ATTAINABLE AT SPECIFIED POWER
             SETTING)
    2661        RETURN
    2662     9 CONTINUE
    2662        IF(HTFL .GT. HCC)WRITE(6,996)
```

```
TAKOFF
          996 FORMAT(/,2X,56HCANNOT ACCEP, AT INPUT R/C (RTCL), TRY VALUE ,LT. L
             9AST R/C PRINTED)
  2674        WRITE(6,997)
          997 FORMAT(/,1X,38H*** ABNORMAL TERMINATION OF TAKOFF ***)
  2704        RETURN
  2704        END


SUBPROGRAM LENGTH

04066

FUNCTION ASSIGNMENTS
YYY     = 000021

STATEMENT ASSIGNMENTS
1       = 000636     2       = 001350     3       = 001637     4       = 001623     5       = 000040     6       = 000064
7       = 000105     8       = 002513     9       = 002663     10      = 002015     15      = 002062     16      = 002102
21      = 001673     22      = 001744     23      = 001774     25      = 002004     26      = 001637     27      = 001517
28      = 001466     29      = 001514     31      = 002205     32      = 002257     33      = 002325     34      = 002232
35      = 002300     39      = 002346     41      = 001554     42      = 001563     43      = 001604     44      = 001620
50      = 002111     55      = 002156     56      = 002176     101     = 000746     102     = 001012     103     = 001003
105     = 000702     106     = 000666     110     = 001018     111     = 001051     112     = 001047     113     = 001107
120     = 001242     201     = 003064     202     = 003076     203     = 003114     204     = 003123     205     = 003141
206     = 003167     207     = 003212     208     = 003221     209     = 003230     210     = 003234     211     = 003240
212     = 003247     213     = 003256     299     = 000451     300     = 001373     996     = 003617     997     = 003630
998     = 003343     999     = 003270     1000    = 003364     1002    = 003421     1009    = 003317     1010    = 003351
1015    = 003476     1020    = 003305     1025    = 003507     1030    = 003522     1040    = 003446     1050    = 003545
1051    = 003557     1052    = 003574     1053    = 003600     1054    = 003607     1055    = 003532     1056    = 003463
1057    = 003334

BLOCK NAMES AND LENGTHS
UNIV    = 000030/01  AERO    = 000020/02  EXCHNG  = 000004/03  XROLL   = 000041/04  XFLATE  = 000072/05

VARIABLE ASSIGNMENTS
ALPHA   = 000010/02  ALPHAJ  = 004031     ALPHJ   = 004033     ANGLE   = 000005/02  ANS     = 003664     CD      = 000014/02
CDGEAR  = 003751     CL      = 000013/02  CY      = 000012/02  DADT    = 003773     DELFD   = 000006/02  DFLPDT  = 003752
DTABS   = 003753     DTFAIL  = 003772     DTGR    = 003754     DTHTDT  = 004043     DTPDWN  = 003755     DTPUP   = 003756
DTVECT  = 003757     EAS     = 004041     EASTO   = 004045     EM      = 000010/01  ENP     = 000021/01  ENPJ    = 004020
ENPOUT  = 004021     EYE     = 000020/01  EYEWNG  = 003760     GAMMA   = 004065     GRCD    = 000016/02  HABS    = 000002/02
HAPT    = 003761     HFLAP   = 004047     HGR     = 003762     HJ      = 004060     HMAN    = 003763     HMAX    = 003774
HPOWER  = 004051     HVECT   = 004053     I       = 003776     IDOWN   = 004004     IFAST   = 000017/02  IFLAP   = 004006
IFLY    = 004005     IG      = 004016     IOUT    = 003764     IPAGE   = 003777     IPOWER  = 004011     IUP     = 004007
IVECT   = 004014     I35     = 004015     JJ1     = 004001     JJ2     = 004002     JROUTE  = 004010     K       = 000071/05
KENG    = 004000     KODE    = 004055     MFLAP   = 004007     MOUT    = 004017     MPOWER  = 004012     MU      = 000037/04
MVECT   = 004013     NCOUNT  = 004030     NEG     = 000036/04     NPAGE   = 003775     PMARG   = 003766     PWRSET  = 004025
QS      = 000001/02  RHO     = 000015/02  ROC     = 000064/05  RTCL    = 003767     S       = 000000/05     SA      = 004027
SET     = 004064     SROLL   = 000000/05  SW      = 000015/01  S35     = 000001/03  T       = 000000/04  TA      = 000022/01
TEMP    = 004022     TG      = 004061     TGU     = 004062     THEMAX  = 000065/05  THETAF  = 000066/05  THETAJ  = 004032
THRUST  = 000003/02  THTPLY  = 003770     THTSCP  = 003771     TIME    = 004063     TJS     = 004057     TOW     = 004024
T2J     = 004034     T4J     = 004035     T5J     = 000003/03  UM      = 003765     VEL     = 000000/02  VEND    = 000001
VFLAP   = 004050     VJ      = 004056     VKTO    = 004044     VKTS    = 004040     VMARG   = 004046     VOUT    = 004037
VPOWER  = 004052     VR      = 000000     VVECT   = 004054     V35     = 000002/03  W       = 000006/01  WF      = 000007/01
WG      = 000023/01  WJ      = 004036     WOS     = 004023     XDADT   = 004042     XDELFD  = 003674     XHFLAP  = 003701
XHPWR   = 003713     XHVECT  = 003737     XLF     = 000067/05  XLFMAX  = 000070/05  XNU     = 003732     XPOWER  = 003725
```

```
DERIV1
                SUBROUTINE DERIV1
          C
          C     SUBROUTINE DERIV1 COMPUTES THE ACCELERATION T(6) FOR THE GROUND ROLL
          C
                REAL MU
                COMMON /UNIV/ NPC    ,NSC    ,IDC    ,H      ,ST     ,R      ,W      ,
               1WF    ,EM     ,VMO    ,EMMO   ,ALPHLO,CLALPH,SW     ,AR     ,B      ,
               2EYEW  ,ENP    ,TA     ,WG     ,WGS    ,KWRITE,DLMCG
               3,KSIZE
                COMMON /AERO/ VEL,QS,HABS,THRUST,TVECT, ANGLE,DELFD,DELSPL,ALPHA,
               9CX,CY,CL,CD,RHO,GRCD,IFAST
                COMMON /XROLL/T(30),NEQ,MU,NREV
                QS = 0.5*RHO*SW*T(4)*T(4)
          4     IF(QS .EQ. 0.)QS = 0.1
          7     IFAST = 1
         10     CALL AERODYN
         11     T(6) = (32.2/W)*(-W*MU + QS*(CY*MU - CX))
         20     T(7) = T(4)
         21     RETURN
         22     END
```

-51-

SUBPROGRAM LENGTH

00041

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

BLOCK NAMES AND LENGTHS
UNIV = 000030/01 AERO = 000020/02 XROLL = 000041/03

VARIABLE ASSIGNMENTS
CX = 000011/02 CY = 000012/02 IFAST = 000017/02 MU = 000037/03 QS = 000001/02 RHO = 000015/02
SW = 000015/01 T = 000000/03 W = 000006/01

START OF CONSTANTS=000025 TEMPS=000031 INDIRECTS=000041

7600 COMPILATION -- RUN76 LEVEL 5C 73/07/05.

ROUTINE COMPILES IN 044000

```
DERIV3
            SUBROUTINE DERIV3
      C     SUBROUTINE DERIV3 COMPUTES THE TIME DERIVATIVES FOR THE AIRBORNE
      C     PORTION OF THE TAKEOFF AND MANAGES THE FLIGHT PATH CONTROL.
      C
            COMMON /UNIV/ NPC    ,NSC    ,IDC    ,M      ,ST     ,R      ,W      ,
           1WF    ,EM     ,VMO    ,EMMO   ,ALPHLO,CLALPH,SW     ,AR     ,B      ,
           2EYEW   ,ENP    ,TA     ,WG     ,WGS    ,KWRITE,DUMC4
           3,KSIZE
            COMMON /AERO/ VEL,QS,HABS,THRUST,TVECT, ANGLE,DELFD,DELSPL,ALPHA,
           9CX,CY,CL,CD,RHO,GRCD,IFAST
            COMMON /XFLATE/ S(52),ROC,THEMAX,THETAF,XLF,XLFMAX,K
            NER = 1
     2      IFAST = 1
     3      QS = 0.5*RHO*SW*S(4)*S(4)
     6      VKTS = S(4)*0.592087
      C
      C     CONSTANT RATE OF CLIMB PORTION
      C
    10        IF(K .EQ. 9)CALL CLIMB(ROC,S(5),VKTS,NER)
    16        IF(NER .NE. 1)WRITE(6,666)ALPHA,ROC
   666  FORMAT(1X,30H**ERROR IN CLIMB - ALPHA,ROC =,2F10.2)
    30    26 IFAST = 0
    31        CALL AERODYN
      C
      C     CHECK FUSELAGE ANGLE, IF THETAF .GT. THEMAX, REDUCE ALPHA (261-262)
      C
    32    261 THETAF = S(5)*57.295 + ALPHA - EYEW
    36        IF(THETAF .LE. THEMAX)GO TO 262
    40        ALPHA = THEMAX + EYEW - S(5)*57.295
    43        GO TO 26
      C
      C     CHECK LOAD FACTOR, IF XLF .GT. XLFMAX, REDUCE ALPHA (262-263)
      C
    43    262 XLF = (QS*CY)/(W*COS(S(5)))
    50        IF(XLF .LE. XLFMA )GO TO 263
    53        ALPHA = ALPHA - 0.05
    55        GO TO 26
      C
      C     CHECK ACCELERATION, IF S(6) .LT. 0.0, REDUCE ALPHA (263-30)
      C
    55    263 S(6) = (32.2/W)*(-CX*QS - W*SIN(S(5)))
    64        IF(S(6) .GE. 0.0)GO TO 30
    66        ALPHA = ALPHA - 0.05
    67        IF(ALPHA .LT. -2.0)GO TO 99
    72        GO TO 26
    72     30 CONTINUE
      C
      C     COMPUTE S(9),S(10),S(11) (30-99)
      C
    72        S(9) = (32.2/(W*S(4)))*(CY*QS - W*COS(S(5)))
   102        IF(K .EQ. 2 .OR. K .EQ. 4)S(8) = 0.0
   113        IF(K .EQ. 3 .OR. K .EQ. 4)S(9) = 0.0
   123        S(10) = S(4) * COS(S(5))
   126        S(11) = S(4) * SIN(S(5))
   132        RETURN
   132     99 WRITE(6,66)S(6)
```

```
DERIV3
          66 FORMAT(1X,46H*** UNABLE TO MAINTAIN ACCEL. .GE. 0.0, S(8) =,F9.5)
    141      K = 99
    142      RETURN
    143      END


SUBPROGRAM LENGTH

00223

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
26     = 000031    30     = 000073    66     = 000164    99     = 000133    261    = 000033    262    = 000044
263    = 000056    666    = 000152

BLOCK NAMES AND LENGTHS
UNIV   = 000030/01 AERO   = 000020/02 XFLATE = 000072/03

VARIABLE ASSIGNMENTS
ALPHA  = 000010/02 CX     = 000011/02 CY     = 000012/02 EYEW   = 000020/01 IFAST  = 000017/02 K      = 000071/03
NER    = 000221    QS     = 000001/02 RHO    = 000015/02 ROC    = 000064/03 S      = 000000/03 SW     = 000015/01
THEMAX = 000065/03 THETAF = 000066/03 VKTS   = 000222    W      = 000006/01 XLF    = 000067/03 XLFMAX = 000070/03

START OF CONSTANTS-000146       TEMPS--000174    INDIRECTS-000216

7600 COMPILATION -- RUN76 LEVEL 5C      73/07/05.

ROUTINE COMPILES IN 044100
```

```
PULLUP
            SUBROUTINE PLLLUP(DADT,KODE,PWRSET,KENG,VEND,HAPT,DTABS)
      C
      C   SUBROUTINE PLLLUP DTERMINES THE TIME RATE OF CHANGE OF THE ANGLE OF
      C   ATTACK DADT REQUIRED TO BRING THE AIRCRAFT FROM THE CONSTANT RATE OF
      C   CLIMB TO THE FINAL CLIMB SPEED(I.E, REDUCE ACCELERATION ALONG FLIGHT
      C   PATH TO ZERO AND THUS INCREASING FLIGHT PATH ANGLE).  THE PULLUP
      C   MANEUVER STARTS AT SPEED OF VEND - PMARG*VEND(E.G. VEND=250., PMARG=0.04,
      C   - START OF PULLUP AT 240. KNOTS).  THE SUBROUTINE DOES THE VERY SAME
      C   INTEGRATION LOOP AS TAKOFF , WITHOUT THE PRINT OUT, AND VARIES THE
      C   VALUE OF DADT UNTIL PROPER VALUE FOUND.  THE SEARCH FOR THE REQUIRED
      C   DADT IS FOUND BY A BISECTION TECHNIQUE.
      C
      C   IF PULLUP FAILS WITH DADT=4.0 (MAX. VALUE ALLOWED), USER SHOULD INPUT
      C   LARGER VALUE FOR PMARG.  THE PROGRAM WILL GENERALLY OVER-SHOOT THE
      C   END SPEED BY A KNOT OR SO.
      C
            EXTERNAL DERIV3
            COMMON /UNIV/ NPC    ,NSC    ,IDC    ,M      ,ST     ,R      ,W      ,
           1WF    ,EM     ,VMO    ,EMMO   ,ALPHLO,CLALPH,SW     ,AR     ,B      ,
           2EYEW  ,ENP    ,TA     ,WG     ,WGS    ,KWRITE,DLMCW
           3,KSIZE
            COMMON /AERO/ VEL,QS,HABS,THRUST,TVECT, ANGLE,DELFD,DELSPL,ALPHA,
           9CX,CY,CL,CD,RHO,GRCD,IFAST
            COMMON /XFLATE/ S(52),ROC,THEMAX,THETAF,XLF,XLFMAX,K
            DIMENSION ANS(8)
            S2J = S(2)
      12    S4J = S(4)
      13    S5J = S(5)
      15    S6J = S(6)
      16    S7J = S(7)
      20    WJ = W
      21    ALPHAJ = ALPHA
      23    KODE = 1
      24    ILOOP = 0
      24    DADT = 4.0
      26    DADTLO = 0.0
      26    ITIME = 1
      30 50 ILOOP = ILOOP + 1
      32    IF(ILOOP .GT. 15) GO TO 95
      35    ISTART = 1
      36 51 S(2) = S2J
      37    S(4) = S4J
      41    S(5) = S5J
      42    S(6) = S6J
      44    S(7) = S7J
      45    W = WJ
      47    ALPHA = ALPHAJ
      51 300 CALL ATMOS(S(7),DTABS,ANS)
      62    SA = ANS(4)*1.68781
      63    RHO = .0023769*ANS(7)
      65    EM = S(4)/SA
      67    HABS = S(7) - HAPT
      71    CALL ENGINE(S(7),DTABS,EM,PWRSET,WF,KENG)
     100    IF(ISTART .NE. 2)
           9CALL INTS(S,4,2,1.,1.,1.,1.,1.,1.,1., DERIV3)
     117    IF(ISTART .EQ. 3)RETURN
```

```
PULLUP
    126         ISTART = 2
    127         W = W - WF*EMP*S(3)/3600.
    133         K = 1
    134         ALPHA = ALPHA + DADT*S(3)
    136         IF(XLF .LT. 0.9)ALPHA = ALPHA + DADT*S(3)
    142         IF(XLF .LT. 0.8)ALPHA = ALPHA + DADT*S(3)
    147         CALL INTM(S,4,2,1.,1.,1.,1.,1.,1., DERIV3)
    161         EAS = S(4)*SQRT(ANS(7))*0.592087
              C
              C   TEST FOR VARIOUS END CONDITIONS ( =110)
              C
    166         IF(S(8) .LT. 0.02 )GO TO 100
    175         IF(EAS .GE. VEND + 0.5 .AND. ITIME .EQ. 1)GO TO 100
    206         GO TO 300
    206   100   IF(ITIME .EQ. 2)GO TO 101
    210         IF(S(8) .GT. 0.02 .AND.EAS .GE. VEND + 0.5)GO TO 150
    223         ITIME = 2
    224   101   IF(DADT .LT. 0.03)GO TO 160
    227         IF(EAS .GT. VEND .AND. EAS .LT. VEND + 1.0)GO TO 200
    240         IF(EAS .LT. VEND)GO TO 110
    241         IF(DADT .EQ. 4.0)GO TO 99
    243         DADTLO = DADT
    244         GO TO 115
    244   110   DADTUP = DADT
    245   115   DADT = 0.5*(DADTUP + DADTLO)
    250         GO TO 50
              C
              C   MODE =2- END SPEED REACHED, BUT TOO MUCH THRUST AVAILABLE - THROTTLING
              C   WILL BE REQUIRED.
              C   MODE = 3 - CANNOT REACH DESIRED END SPEED AT SPECIFIED POWER SETTING -
              C   PULLUP DONE TO ZERO ACCELERATION
              C   MODE = 9 - PROGRAM FAILS
              C
    250   150   MODE = 2
    251         GO TO 200
    252   160   MODE = 3
    253   200   ISTART = 3
    254         GO TO 51
    255    99   WRITE(6,98)
           98   FORMAT(1X,44H*** FAILED IN PULLUP - TRY INPUT PMARG = 0.1)
    265         MODE = 9
    266         RETURN
    266    95   WRITE(6,96)
           96   FORMAT(1X,29H     EXCESSIVE LOOPS IN PULLUP)
    276         MODE = 9
    277         RETURN
    277         END


SUBPROGRAM LENGTH

00372

FUNCTION ASSIGNMENTS
```

CLIMB

```
            SUBROUTINE CLIMB(ROC,GAMMA,VKTS,NER)
      C
      C   SUBROUTINE CLIMB FINDS THE REQUIRED ALPHA TO FLY AT THE CONSTANT
      C   RATE OF CLIMB RTCL, GIVEN THE THRUST AND VELOCITY.  CHANGES IN FLIGHT
      C   PATH ANGLE ARE FAIRLY INSENSITIVE TO VARIATIONS IN ANGLE OF ATTACK.
      C   AS A RESULT, THE COMPUTED RATE OF CLIMB WILL DIFFER SOME (USUALLY LOWER)
      C   THAN THE DESIRED VALUE RTCL.
      C   SUBROUTINE ZERJVB IS A ZERO-FINDER.
      C
            REAL NO
            COMMON /UNIV/ NPC    ,NSC    ,IDC    ,H      ,ST     ,R      ,W      ,
           1WF    ,EM     ,VMO    ,EMMO   ,ALPHLO,CLALPH,SW     ,AR     ,B      ,
           2EPSR  ,ENP    ,TA     ,HG     ,HGS    ,KWRITE,DLMCH
           3,NSIZE
            COMMON /AERO/ VEL,QS,HABS,THRUST,TVECT, ANGLE,DELED,DELSPL,ALPHA,
           9CX,CY,CL,CD,RHO,GMCD,IFAST
            NER = 1
            TOL = 0.01
            STEP = 1.0
            JX = 0
            JC = 0
            QS = 0.5*RHO*VKTS*VKTS*2.8561*SW
            ERROR = 999.
         40 IF(JX .EQ. 0)ERRM1 = ERROR
            IFAST = 0
            CALL AERODYN
            ALPHX = ALPHA*.017453
            ERROR = QS*(CY*COS(GAMMA) - CX*SIN(GAMMA)) - W
            IF(ABS(ERROR) .LT. 0.0025)GO TO 60
            IF(JX .EQ. 2)GO TO 60
            CALL ZERJVB(ERROR,ERRM1,ALPHA,STEP,TOL,JC,JX)
            IF(JC .GT. 25)GO TO 65
            GO TO 40
         60 RETURN
         65 NER = 9
            RETURN
            END
```

SUBPROGRAM LENGTH

00132

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

40	= 000021	60	= 000073	65	= 000074

BLOCK NAMES AND LENGTHS

UNIV	= 000050/01	AERO	= 000020/02

VARIABLE ASSIGNMENTS

ALPHA	= 000010/02	ALPHX	= 000131	CX	= 000011/02	CY	= 000012/02	ERRM1	= 000130	ERROR	= 000127
IFAST	= 000017/02	JC	= 000126	JX	= 000125	NO	= 000122	QS	= 000001/02	RHO	= 000015/02
STEP	= 000124	SW	= 000015/01	TOL	= 000123	W	= 000006/01				

```
LANDNG
                SUBROUTINE LANDNG(INPC,IWRITE,WGROSS,SWING,XENG,PWRSET,MODE,
               9IZERO,IREV,NER)
      C
      C    SUBROUTINE LANDNG MANAGES THE COMPUTATION OF THE LANDING MANEUVER
      C    BY CALLING INPUT, FINDING APPROACH SPEED, IF DESIRED(CALL LAND1 LOOP),
      C    FINDING REQUIRED VALUES OF THRUST PER ENGINE AND ANGLE OF ATTACK
      C    FOR STEADY STATE APPROACH(CALL ZERO), ITERATING ON FLARE INITIATION
      C    ALTITUDE(CALL FLARE LOOP),EXECUTING FLARE(CALL FLARE), AND EXECUTING
      C    THE GROUND ROLL(CALL ROLL).
      C
      C    SUBROUTINE ITRJVB IS A ZERO-FINDING ROUTINE, WHICH VARIES THE
      C    INDEPENDENT VARIABLE BASED ON THE SIZE AND SIGN OF THE ERROR.
      C    THE LOOP CONTAINING ITRJVB IS EXITED SUCESSFULLY WHEN JX = 2,
      C    OR THE MAGNITUDE OF THE ERROR IS LESS THAN SOME ACCEPTABLE VALUE.
      C    VARIABLE JC COUNTS THE NUMBER OF TRIES AND  LIMITS THAT NUMBER TO 25
      C
      C    FOLLOWING CERTAIN COMMENTS WILL BE TWO NUMBERS IN PARENTHESIS
      C    WHICH WILL GIVE THE APPROXIMATE STATEMENT-NUMBER RANGE OF THE
      C    FUNCTION DESCRIBED IN THE PRECEDING COMMENT.
      C
                COMMON /UNIV/ NPC    ,NSC    ,IDC    ,H      ,ST     ,R      ,W      ,
               1WF    ,EM     ,VMO    ,EMMO   ,ALPHL0,CLALPH,SW     ,AR     ,B      ,
               2EYEW  ,ENP    ,TA     ,WG     ,WGS    ,KWRITE,DLMC4
               3,KSIZE
                COMMON /AERO/ VEL,QS,HABS,THRUST,TVECT, ANGLE,DELFD,DELSPL,ALPHA,
               9CX,CY,CL,CD,RHO,GRCD,IFAST
                COMMON /XFLATE/ S(52),ROC,THEMAX,THETAF,XLF,XLFMAX,K
                COMMON /XROLL/ T(30),NEQ,MU,NREV
                COMMON /XLAND/ DTPENG,PWRIDL,TBRK,TSPL,TFLP,TREV,TOFF
                COMMON/LAND/ GAMAPP,VKAPP,ALPHMX
                COMMON/PAST/ ALPAST,XLFPST,DTDT,DTDTMX,DADT
                COMMON/MARGIN/XLFMRG,PWRMRG,PWRMAX
                COMMON/TCHDWN/RSTD,VKTD,STD
                COMMON /ATMOSX/ HAPT,DTABS
                DIMENSION ANS(8)
                YYY(XNOW,XPAST,YNOW,YPAST,YANS)=XNOW-(YNOW-YANS)/((YPAST-YNOW)/
               *(XPAST-XNOW))
                DATA ABAR,CDGEAR,DADT,DTABS,DTDTMX,DTPENG,FAAFTR,HAPP,HAPT,
               9HSPOIL,IPOWER,ISPOIL,PWRIDL,PWRMAX,PWRMRG,SINKTD,TDELAY,
               8THEMAX,TBRK,TFLP,TOFF,TREV,TSPL,XDIST,XMUBRK,XLFMAX/0.35,0.0,5.0,
               70.0,7.0,3.0,0.60,35.0,0.0,10.0,1,1,0.15,1.00,0.05,10.0,
               61.0,15.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,5000.,0.40,1.15/
                DATA XLFMRG,STEP/0.003,0.90/
                NAMELIST /NAM3/ABAR,CDGEAR,DADT,DTABS,DTDTMX,DTPENG,FAAFTR,
               5HAPP,HAPT,HSPOIL,IPOWER,ISPOIL,PWRIDL,PWRMAX,PWRMRG,SINKTD,
               4TDELAY,THEMAX,TBRK,TFLP,TOFF,TREV,TSPL,XDIST,XMUBRK,XLFMAX,
               9XLFMRG,QUESS,HFLARE,STEP
                GO TO(5,6,5),INPC
   41         5 QUESS = 0.80
   42           ABAR = 0.35
   44           XLFMAX = 1.15
   45           HFLARE = 0.0
      C
      C    INPUT DATA LOADED INTO PROGRAM BY NAMELIST /NAM3/
      C
   46           READ(5,NAM3)
```

```
LANDNG
     51        IF(INPC .EQ. 1)RETURN
     57      6 ENP = XENG
     60        NER = 1
     61        SW = SWING
     62        WG = WGROSS
     63        WOS = WG/SW
     64        ALPHMX = THEMAX
               C
               C   FIND GEAR DRAG IF DEFAULT OF CDGEAR = 0.0 USED
               C
     66        IF(CDGEAR .EQ. 0.0)CDGEAR = (0.0032/SW)*WG**0.80
     75        GRCD = CDGEAR
     76        IF(IWRITE .EQ. 1)GO TO 11
               C
               C   WRITE OUT PROGRAM INPUTS
               C
    100        WRITE(6,7)WG,SW,WOS
              7 FORMAT(1H1,1X,34H** INPUTS TO LANDING - GROSS WT. =,F8.1,14H   WIN
               9G AREA =,F6.1,17H   WING LOADING =,F5.1)
    116        WRITE(6,8)ABAR,CDGEAR,DADT,DTABS,DTDTMX,DTPENG,FAAFTR,HAPP,HAPT
              8 FORMAT(5X,5HABAR=,F4.2,2X,7HCDGEAR=,F5.3,2X,5HDADT=,F3.1,2X,
               96HDTABS=,F4.1,2X,7HDTDTMX=,F3.1,2X,7HDTPENG=,F3.1,2X,7HFAAFTR=,
               9F4.2,2X,5HHAPP=,F4.1,2X,5HHAPT=,F6.1)
    147        WRITE(6,9)NSPOIL,IPOWER,ISPOIL,PWRIDL,PWRMAX,PWRMRG,SINKTD,TDELAY
              9 FORMAT(5X,7HNSPOIL=,F4.1,2X,7HIPOWER=,I2,2X,7HISPOIL=,I2,2X,
               97HPWRIDL=,F4.2,2X,7HPWRMAX=,F5.2,2X,7HPWRMRG=,F4.2,2X,7HSINKTD=,
               9F4.1,2X,7HTDELAY=,F3.1)
    176        WRITE(6,10)THEMAX,TBRK,TFLP,TOFF,TREV,TSPL,XMUBRK,XLFMAX,XDIST
             10 FORMAT(5X,7HTHEMAX=,F5.1,2X,5HTBRK=,F4.1,2X,5HTFLP=,F5.1,2X,
               95HTOFF=,F4.1,2X,5HTREV=,F4.1,2X,5HTSPL=,F5.1,2X,7HXMUBRK=,F4.2,2X,
               97HXLFMAX=,F4.2,2X,6HXDIST=,F6.1)
    227       11 CONTINUE
               C
               C   IF REQUESTED, FIND THE APPROACH SPEED VKAPP REQUIRED TO LAND IN THE
               C   DISTANCE XDIST FROM 35 FT.(11 - 100)
               C   VARIABLE JC COUNTS THE NUMBER OF TRIES TO FIND VKAPP (LIMIT OF 25)
               C
    227        IF(VKAPP .NE. 0.0)GO TO 100
    230        F = 1.1
    231        TOL = 0.25
    233        JC = 0
    233        JX = 0
    234        VKAPP = 100.
    236        FLD = XDIST
    237        ERROR = 1.0
    241       50 IF(JX .EQ. 0)ERRM1 = ERROR
    244        CALL LAND1(VKAPP,GAMAPP,XLFMAX,SINKTD,HAPP,ABAR,TDELAY,DIST)
    254        ERROR = (FLD - DIST)/FLD
    256        IF(JX .EQ. 2)GO TO 100
    264        IF(ABS(ERROR) .LT. 0.005) GO TO 100
    267        CALL ITRJVB(ERROR,ERRM1,VKAPP,F,TOL,JC,JX)
    270        IF(JC .GT. 25) GO TO 97
    305        GO TO 50
    305       97 WRITE(6,96)VKAPP,GAMAPP,XLFMAX,SINKTD,HAPP,ABAR,TDELAY,
               9ERROR,ERRM1,JC,JX
              96 FORMAT(5X,24H*** FAILED TO FIND VKAPP,/,10X,45HVKAPP,GAMAPP,XLFMAX
```

```
LANDNG
          9,SINKTD,HAPP,ABAR,TDELAY =,7F10.2,/,10X,19HERROR,ERRM1,JC,JX =,
          92F10.4,213)
  343       NER = 99
  344       RETURN
  345   100 CONTINUE
        C
        C   FIND THRUST AND ALPHA FOR STEADY STATE APPROACH, IF REQUESTED(100-200)
        C
  345       HABS = 35.0
  346       IF(IZERO .EQ. 0)GO TO 200
  350   150 ALT = HAPT + HABS
  352       CALL ATMOS(ALT,DTABS,ANS)
  355       KENG = 1
  356       EM = VKAPP/ANS(4)
  360       VEL = VKAPP * 1.69
  361       QS = 0.5*ANS(3)*SW*VEL*VEL
  365       CALL ZERO(NER,EM,ALT,DTABS,KENG,PWRSET,QUESS)
  377       IF(NER .EQ. 1)GO TO 200
  405       WRITE(6,156)VKAPP,GAMAPP,ALT,DTABS,PWRSET,QUESS,ALPHMX,ENP,ST,W
        156 FORMAT(5X,23H*** FAILED IN ZERO-LAND,/,10X,31HVKAPP,GAMAPP,ALT,DTA
           9B5,PWRSET =,5F10.2,/,10X,23HQUESS,ALPHMX,ENP,ST,W =,5F10.2)
  441       NER = 98
  442       RETURN
  443   200 CONTINUE
        C
        C   ITERATION TO FIND PROPER FLARE INITIATION ALTITUDE(200-61)
        C   VARIABLE JC COUNTS THE NUMBER OF TRIES TO FIND HFLARE (LIMIT OF 25)
        C
  443       NWRITE = 1
  444       IF(IWRITE .EQ. 3)NWRITE = 2
  447       TS = 0.0
  450       TIME = 0.0
  451       RTSNK = 101.4*VKAPP * SIN(GAMAPP*.0174533)
  462       TJ = THRUST
  463       DJ = DELSPL
  465       AJ = ALPHA
  466       PJ = PWRSET
  467       F = 1.25
  471       IF(HFLARE .NE. 0.0)F = STEP
  474       IF(HFLARE .EQ. 0.0)
           *HFLARE = ((VKAPP*1.69)**2) * ((GAMAPP/57.3)**2 - (SINKTD /(VKAPP
           1*1.69))**2) / (64.4 * (XLFMAX - 1.0))
  506       IF(HFLARE .GT. 100.0 .OR. HFLARE .LT. 0.0)HFLARE = 25.0
  517       TOL = 0.1
  520       BU = 100.
  521       JC = 0
  522       JX = 0
  523       ERROR = 1.0
        C
        C   LOOPING WITH HFLARE(40-60)
        C
  525    40 IF (JX .EQ. 0) ERRM1 = ERROR
  530       THRUST = TJ
  531       DELSPL = DJ
  533       ALPHA = AJ
  534       PWRSET = PJ
```

```
LANDNG
      536         W = WG
      537         IF(NWRITE .EQ. 2.WRITE(6,201)HFLARE,STEP,PWRSET,HSPOIL,SINKTD
          201 FORMAT(1H1,/,3X,14HHFLARE,STEP,PWRSET,HSPOIL,SINKTD =,5F8.2)
      564         IF(NWRITE .EQ. 2.WRITE(6,1000)
      576         CALL FLARE (HFLARE,PWRSET HSPOIL,T5,TIME,NWRITE,ISPOIL,IPOWER)
                C
                C    CHECK ACTUAL TOUCHDOWN RATE OF SINK AGAINST THAT DESIRED
                C
      606         ERROR = (RSTD - SINKTD) / SINKTD
      610         IF (JX .EQ. 2) GO TO 60
      616         IF (ABS(ERROR) .LT. 0.025) GO TO 60
      621         XPAST = HFLARE
      622         YPAST = RSTD
                C
                C    GUESS NEW HFLARE BASED ON ERROR(CALL ITRJVB)
                C    IF HFLARE IS LESS THAN 0.0 OR GREATER THAN 100.0 FT, EXIT PROGRAM
                C
      624         CALL ITRJVB(ERROR,ERRM1,HFLARE,F,TOL,JC,JX)
      633         IF (JC .GT. 25) GO TO 299
      642         IF (HFLARE .GT. 80 .OR. HFLARE .LE. 0.)  GO TO 299
      652         GO TO 40
      652     299 WRITE (6,298) HFLARE, RSTD, SINKTD, PWRSET, HSPOIL, IPOWER, ISPOIL
                 *, ERROR, ERRM1, JC, JX
              298 FORMAT (5X, 24H*** FAILED IN FLARE-LAND,/,10X, 48HHFLARE,RSTD,SINK
                 1TD,PWRSET,HSPOIL,IPOWER,ISPOIL = , 5F9.2, 2I3/ 10X, 19HERROR,ERRM1
                 1,JC,JX = ,2F7.4,2I3)
      711         NER = 97
      712         RETURN
      713      60 THRUST = TJ
      714         ALPHA = AJ
      716         DELSPL = DJ
      717         PWRSET = PJ
      721         IF (JC .LT. 1) GO TO 61
      723         HFLARE = YYY(HFLARE,XPAST,RSTD,YPAST,SINKTD)
      734      61 IF (IWRITE .GT. 1) WRITE (6,1001) VKAPP, GAMAPP, RTSNK, SINKTD,
                 *XLFMAX, ALPHMX, DELSPL, DELFD, ALPHA, THRUST, IPOWER, ISPOIL ,
                 9PWRSET
             1001 FORMAT(1H1, 2X, 7HLANDING / 5X, 7HVKAPP =, F5.1, 3X, 8HGAMAPP =,
                 1F5.2, 3X, 7HRTSNK =, F6.1, 3X, 8HSINKTD =, F5.2, 3X, 8HXLFMAX =,
                 9F5.2,3X,8HALPHMX =,F5.2,/,5X,8HDELSPL =,F5.2,3X,7HDELFD =,F5.2,
                 93X,7HALPHA =,F5.2,3X,8HTHRUST =,F8.1,3X,8HIPOWER =,I3,3X,
                 98HISPOIL =,I3,/,5X,30HAPPROACH AT THROTTLE SETTING =,F5.3,/)
      1001       IF(IWRITE .GT. 1)WRITE(6,1000)
                C
                C    SET UP FOR PRINT OUT(61-)60)
                C
      1013        IF(HFLARE .GE. HAPP)GO TO 161
      1016        GO TO 162
      1016     161 TIME = 0.0
      1017        SAPP = 0.0
      1017        OVER = HFLARE
      1021        GO TO 160
      1022     162 TIME = -0.5
      1023        OVER = 0.0
      1024        S(4) = VKAPP*1.69
      1026        VRTS = VKAPP
```

```
LANDNG
   1027         EAS = VKTS * SQRT(ANS(7))
   1034         GAMMA = -GAMAPP
   1035         S(8) = 0.0
   1035         XLF = 1.0
   1037         DTDT = 0.0
   1040         CONST = (RTSNK/120.)/TAN(GAMAPP*.0174533)
   1046         S(7) = HAPP + RTSNK/120.
   1050         TS = -CONST
   1055     163 S(7) = -RTSNK/120. + S(7)
   1060         IF(S(7) .GT. HFLARE)GO TO 164
   1063         GO TO 165
   1063     164 HABS = S(7)
   1065         CALL AERODYN
   1066         SXX7 = S(7)
   1067         TS = TS + CONST
   1071         TIME = TIME + 0.5
   1073         IF(IWRITE .EQ. 1)GO TO 163
   1101         S(8) = 0.0
   1101         XLF = 1.0
   1103         THRIST = ENP * THRUST
   1105         THETAF = ALPHA - EYEW + GAMMA
   1110         WRITE(6,1002)TIME,TS,WG,S(7),VKTS,EAS,EM,S(8),CL,CD,ALPHA,GAMMA,
               9RTSNK,XLF,THRIST,THETAF,DTDT
   1160         GO TO 163
   1164     165 XPART = (SXX7 - HFLARE)/(RTSNK/120.)
   1170         TIME = TIME + XPART*0.5
   1172         TS   = TS + CONST*XPART
   1175     160 IF(IWRITE .GT. 1)WRITE(6,899)HFLARE
            899 FORMAT(5X,18H** BEGIN FLARE AT ,F4.1,5H FEET)
   1211         W = WG
                C
                C   CALL TO FLARE WITH FINAL HFLARE VALUE
                C
   1213         CALL FLARE(HFLARE,PWRSET,HSPOIL,TS,TIME,IWRITE,ISPOIL,IPOWER)
   1223         IF(IWRITE .GT. 1)WRITE(6,996)
            996 FORMAT(5X,12H** TOUCHDOWN)
   1240         VKTS = VKTD
   1241         EAS = VKTS*SQRT(ANS(7))
   1247         TS = STD
   1250         S(7) = 0.0
   1250         GAMMA = 0.0
   1251         XLF = 0.0
   1252         THRIST = ENP * THRUST
   1254         THETAF = ALPHA - EYEW
   1256         RTSNK = SINKTD * 60.
   1260         IF(IWRITE .GT. 1)WRITE(6,1002)S(2),IS,W,S(7),VKTS,EAS,EM,S(8),CL,
               9CD,ALPHA,GAMMA,RTSNK,XLF,THRIST,THETAF,DTDT
            1002 FORMAT(1X,F5.1,F9.1,F9.0,F9.1,F7.1,F6.1,F6.3,F7.2,F7.3,F7.4,2F7.2,
               9F8.1,F6.2,F9.0,F6.1,F8.3)
                C
                C   CALL TO ROLL FOR GROUND ROLL OUT
                C
   1342         CALL ROLL(MODE,PWRSET,VKTD,STD,TIME,ABAR,TDELAY,XMUBRK,TS,IWRITE)
   1354         IF(OVER .GT. 0.0)TS = TS - (HFLARE - HAPP)/TAN(GAMAPP*.0174533)
   1371         DLAND = TS/FAAFTR
   1373         IF(IWRITE .GT. 1)WRITE(6,898)TS,DLAND,ABAR
```

```
LANDNG
          898 FORMAT(/,3X,18HLANDING DISTANCE =,F7.1,23H FEET.   FIELD LENGTH =,
            9F7.1,15H FEET.   ABAR =,F6.3)
         1000 FORMAT(131H  TIME    DIST.   WEIGHT    ALT.    TAS   EAS   MACH  A
            9CCEL     CL      CD   ALPHA  GAMMA     R/C   LOAD   THRUST  FUS.   TH
            9ETA        ,/,130H (SEC)    (FEET)     (LBS)   (FEET)  (KTS) (KTS)
            9NO. (FPS2)                  (DEG)  (DEG)    (FPM)  FACT     (LBS)  ANG
            9.    DOT         ,/)
  1416       RETURN
  1417       END


SUBPROGRAM LENGTH

02271

FUNCTION ASSIGNMENTS
YYY     - 000023

STATEMENT ASSIGNMENTS
5       - 000042     6       - 000060     7       - 001565     8       - 001601     9       - 001623     10      - 001644
11      - 000230     40      - 000526     50      - 000242     60      - 000714     61      - 000735     96      - 001672
97      - 000305     100     - 000346     150     - 000351     156     - 001714     160     - 001176     161     - 001017
162     - 001023     163     - 001056     164     - 001064     165     - 001165     200     - 000444     201     - 001744
298     - 001754     299     - 000653     898     - 002071     899     - 002043     996     - 002051     1000    - 002104
1001    - 001774     1002    - 002056

BLOCK NAMES AND LENGTHS
UNIV    - 000030/01  AERO    - 000020/02  XFLATE  - 000072/03  XROLL   - 000041/04  XLAND   - 000007/05  LAND    - 000003/06
PARM    - 000005/07  MARGIN  - 000003/08  TCHDWN  - 000003/11  ATMOSX  - 000002/12

VARIABLE ASSIGNMENTS
ABAR    - 002213     AJ      - 002252     ALPHA   - 000010/02  ALPHMX  - 000002/06  ALT     - 002242     ANS     - 002203
BU      - 002254     CD      - 000014/02  CDGEAR  - 002214     CL      - 000013/02  CONST   - 002264     DADT    - 000004/07
DELFD   - 000006/02  DELSPL  - 000007/02  DIST    - 002241     DJ      - 002251     DLAND   - 002270     DTABS   - 000001/12
DTDT    - 000002/07  DTDTMX  - 000003/07  DTPENG  - 000000/05  EAS     - 002262     EM      - 000010/01  ENP     - 000021/01
ERRM1   - 002240     ERROR   - 002237     EYEW    - 000020/01  F       - 002232     FAAFTR  - 002215     FLD     - 002236
GAMAPP  - 000000/06  GAMMA   - 002263     GRCD    - 000016/02  HABS    - 000002/02  HAPP    - 002216     HAPT    - 000000/12
HFLARE  - 002230     ISPOIL  - 002217     IPOWER  - 002220     IREV    - 000002     ISPOIL  - 002221     IZERO   - 000001
JC      - 002234     JX      - 002235     KENG    - 002243     MODE    - 000000     NER     - 000003     NWRITE  - 002244
OVER    - 002260     PJ      - 002253     PWRIDL  - 000001/05  PWRMAX  - 000002/08  PWRMRG  - 000001/08  QS      - 000001/02
QUESS   - 002227     RSTD    - 000000/11  RTSNK   - 002247     S       - 000000/03  SAPP    - 002257     SINKTD  - 002222
ST      - 000004/01  STD     - 000002/11  STEP    - 002226     SW      - 000015/01  SXAT    - 002265     T       - 000000/04
TBRK    - 000002/05  TDELAY  - 002223     TFLP    - 000004/05  THEMAX  - 000065/03  THETAF  - 000066/03  THRIST  - 002266
THRUST  - 000003/02  TIME    - 002246     TJ      - 002250     TOFF    - 000006/05  TOL     - 002233     TREV    - 000005/05
TSPL    - 000003/05  TS      - 002245     VEL     - 000000/02  VKAPP   - 000001/06  VKTD    - 000001/11  VKTS    - 002260
W       - 000006/01  WG      - 000023/01  WOS     - 002231     XDIST   - 002224     XLF     - 000067/03  XLFMAX  - 000070/03
XLFMRG  - 000000/08  XNUBRK  - 002225     XPART   - 002267     XPAST   - 002255     YPAST   - 002256

START OF CONSTANTS-001556       TEMPS--002141    INDIRECTS-002202

7600 COMPILATION -- RUN76 LEVEL 5C      73/07/04.

ROUTINE COMPILES IN 051500
```

```
FLARE
                SUBROUTINE FLARE(HFLARE,PWRSET,HSPOIL,TS,TIME,IWRITE,ISPOIL,
               9IPOWER)
      C
      C   SUBROUTINE FLARE COMPUTES THE FLARE MANEUVER BY RUNGE-KUTTA NUMERICAL
      C   INTEGRATION(CALL INTK).  THE INTEGRATION STEP SIZE IS 0.1 SECONDS.
      C   THE INPUT VARIABLE HFLARE IS THE HEIGHT AT WHICH THE FLARE IS STARTED.
      C
      C   THE VARIABLE DAGODA IS PROVIDED TO THE USER AS A MEANS TO ACCOUNT FOR
      C   A LOSS IN THE PITCH RATE CAPABILITY DUE TO NEGATIVE GROUND EFFECTS.
      C   DAGODA = 1.0 WOULD BE THE VALUE FOR OUT OF GROUND EFFECT, AND LESS THAN
      C   1.0 FOR IN GROUND EFFECT.  DAGODA MAY BE A FUNCTION OF WING HEIGHT/SPAN,
      C   TOTAL LIFT COEFFICIENT, CIRCULATION LIFT COEFF., ETC, AND SHOULD BE
      C   COMPUTED IN SUBROUTINE AERODYN AND PASSED THRU COMMON /EBFGRD/.
      C   THE USER MAY SIMPLY IGNORE DAGODA IF HE SO DESIRES.
      C
            EXTERNAL DERIV2
            DIMENSION ANS(8)
            COMMON /UNIV/ NPC    ,NSC    ,IDC    ,H      ,ST     ,R      ,W      ,
           1WF     ,EM     ,VMO    ,EMMO   ,ALPHL9,CLALPH,SW     ,AR     ,B      ,
           2EYEW   ,ENP    ,TA     ,WG     ,WGS    ,KWRITE,DLMC4
           3,KSIZE
            COMMON /AERO/ VEL,QS,HABS,THRUST,TVECT, ANGLE,DELFD,DELSPL,ALPHA,
           9CX,CY,CL,CD,RHO,GRCD,IFAST
            COMMON /XFLATE/ S(52),ROC,THEMAX,THETAF,XLF,XLFMAX,K
            COMMON/LAND/ GAMAPP,VKAPP,ALPHMX
            COMMON/PAST/ ALPAST,XLFPST,DTDT,DTDTMX,DADT
            COMMON /EBFGRD/ DELJR,ETATRN,DAGODA
            COMMON/MARGIN/XLFMRG,PWRMRG,PWRMAX
            COMMON/TCHDWN/RSTD,VKTD,STD
            COMMON /ATMOSX/ HAPT,DTABS
            YYY(XNOW,XPAST,YNOW,YPAST,YANS)=XNOW-(YNOW-YANS)/((YPAST-YNOW)/
           *(XPAST-XNOW))
      C
      C   INTEGRATION VARIABLES
      C     S(1) = NUMBER OF EQUATIONS
      C     S(2) = TIME (SEC.)
      C     S(3) = TIME INTERVAL,STEP SIZE (SEC.)
      C     S(4) = VELOCITY ALONG FLIGHT PATH (FT./SEC.)
      C     S(5) = FLIGHT PATH ANGLE (RAD.)
      C     S(6) = DISTANCE ALONG FLIGHT PATH (FT.)
      C     S(7) = ALTITUDE (FT.)
      C     S(8) = ACCELERATION ALONG FLIGHT PATH (FT./SEC.**2)
      C     S(9) = TIME RATE OF CHANGE OF FLIGHT PATH ANGLE (RAD./SEC.)
      C     S(10) = HORIZONTAL SPEED (FT./SEC.)
      C     S(11) = RATE OF CLIMB (FT./SEC.)
            N = 4
   31       INODE = 1
   32       JNODE = 1
   33       MROUTE = 1
   33       DAGODA = 1.0
   35       W = WG
      C
      C   INITIALIZATION FOR INTEGRATION LOOP(109-108)
      C
   36       S(1) = 4.
   40       S(2) = TIME
```

```
FLARE
      41          S(3) = 0.1
      42          S(4) = VKAPP * 1.69
      44          S(5) =-GAMAPP / 57.3
      46          S(6) = 0.
      47          S(7) = HFLARE
      51          VEL = S(4)
      52          ALPAST = ALPHA
      54          GO TO 108
      54      109 MROUTE = 2
      55          CALL INTS(S,N,2,1.,1.,1.,1.,1.,1., DERIV2)
      67          HABS = S(7)
      70          VKTS = VKAPP
      72          EAS = VKTS * SQRT(ANS(7))
      77          EM = VKAPP / ANS(4)
     100          GAMMA =-GAMAPP
     102          RTSNK =-101.4*VKAPP*SIN(S(5))
     110          XLF = 0.999
     111          XLFPST = 0.999
     111          THETAF = ALPHA - EYEW - GAMAPP
     114          DTDT = 0.
     115          THRIST = ENP * THRUST
     117          IF(IWRITE .GT. 1)
                 *WRITE(6,1002)S(2),T5,W,S(7),VKTS,EAS,EM,S(8),CL,CD,ALPHA,GAMMA,
                 *RTSNK,XLF,THRIST,THETAF,DTDT
                C
                C   MAIN INTEGRATION LOOP(108-99)
                C
     205      108 HALT = S(7) + HAPT
                C
                C   OBTAIN ATMOSPHERIC VARIABLES
                C
     207          CALL ATMOS(HALT,DTABS,ANS)
     212          EM = S(4)/(ANS(4) * 1.69)
     214          RHO = ANS(3)
     216          HABS = S(7)
     217          HALT = S(7) + HAPT
     221          VEL = S(4)
     222          VKTS = S(4)/1.69
     224          IF(MROUTE .EQ. 1)GO TO 109
     232       70 ALPAST = ALPHA
     233          XLFPST = XLF
                C
                C   INCREASE ANGLE OF ATTACK AT A RATE OF DADT*DAGODA
                C
     235          ALPHA = ALPHA + DADT * S(3) * DAGODA
     240          IF(XLF .LT. 1.00)ALPHA = ALPHA + DADT*S(3)*DAGODA
                C
                C   MAKE INTEGRATION STEP(CALL INTM)
                C
     245          CALL INTM(S,N,2,1.,1.,1.,1.,1.,1., DERIV2)
     263          W = W - WF*ENP*S(3)/3600.
     267          T5 = T5 + S(3) * S(10)
     271          HABS = S(7)
     272          IF(S(11) .GE.0.0) GO TO 301
     274          IF(S(7) .LE. 0.00)GO TO 300
     276          S7PAST = S(7)
```

```
FLARE
    276         SLIPS/ = S(11)
    277         IF(IWRITE .EQ. 1) GO TO 107
    301         IF(XLF .GT. XLFMAX)XLF = XLFMAX
    304         VKTS = S(4)/1.69
    306         EAS = VKTS * SQRT(ANS(7))
    313         GAMMA = S(5) * 57.3
    314         RTSNK =-60. * S(11)
    316         THETAF = ALPHA - EYEW + S(5)*57.3
    322         THRIST = ENP * THRUST
    324         IF(S(7) .LT. 0.00)S(7) = 0.10
    332         WRITE(6,1002)S(2),T5,W,S(7),VKTS,EAS,EM,S(8),CL,CD,ALPHA,GAMMA,
               *RTSNK,XLF,THRIST,THETAF,DTDT
         1002 FORMAT(1X,F5.1,F9.1,F9.0,F9.1,F7.1,F6.1,F6.3,F7.2,F7.3,F7.4,2F7.2,
              9F8.1,F6.2,F9.0,F5.1,F8.2)
    412    107 CONTINUE
C
C     SPOILER RETRACTION(107-89)
C
    412         IF(ISPOIL .EQ. 0 .OR. DELSPL .EQ. 0.0)GO TO 89
    420         IF((PWRMAX - PWRSET) .LE. PWRMRG)GO TO 80
    423         IF(HABS .GT. HSPOIL)GO TO 89
    427         IF(ABS(XLFMAX - XLF) .LE. XLFMRG)GO TO 89
    433     80 DELSPL = DELSPL - 2.5
    435         IF(JNODE .EQ. 1 .AND. IWRITE .GT. 1)WRITE(6,86)HABS
             86 FORMAT(5X,22HSPOILERS RETRACTED AT ,F4.1,14H FEET ALTITUDE)
    456         JNODE = 2
    457         IF(DELSPL .LT. 0.0)DELSPL = 0.0
    461         CALL DERIV2
    462         IF(DELSPL .EQ. 0.0)GO TO 89
    467         IF(ABS(XLFMAX - XLF) .LE. XLFMRG)GO TO 89
    473         GO TO 80
    474     89 CONTINUE
C
C     THRUST MODULATION(89-99)
C
    474         IF(IPOWER .EQ. 0)GO TO 99
    476     90 IF(XLF .GE. XLFPST)GO TO 99
    501         PWRSET = PWRSET + 0.01
    502         IF(INODE .EQ. 1 .AND. IWRITE .GT. 1)WRITE(6,91)
             91 FORMAT(5X,50HTHRUST MODULATION REQUIRED TO MAINTAIN LOAD FACTOR )
    521         INODE = 2
    522         IF(PWRSET .GT. PWRMAX)PWRSET = PWRMAX
C
C     CALL ENGINE IF POWER SETTING CHANGED (KENG = 2) AND TRY TO MAINTAIN
C     LOAD FACTOR.
C
    525         KENG = 2
    526         CALL ENGINE(HALT,DTABS,EM,PWRSET,WF,KENG)
    533         IF(PWRSET .EQ. PWRMAX)GO TO 99
    541         CALL DERIV2
    542         IF(XLF .GE. XLFMAX)GO TO 99
    550         GO TO 90
    551     99 CONTINUE
    551         GO TO 108
C
C     FIND RATE OF SINK AT TOUCHDOWN.
```

```
FLARE
      C
    552     300 RSTD =-YYY(S(11),S11PST,S(7),S7PAST,0.0)
    563         VKTD = S(4)/1.69
    565         STD = T5
    566         TIME = S(2)
    567         RETURN
    570     301 RSTD = 0.
    571         RETURN
    572         END


SUBPROGRAM LENGTH

00720

FUNCTION ASSIGNMENTS
YYY     - 000021

STATEMENT ASSIGNMENTS
70      - 000233      80      - 000434      86      - 000624      89      - 000475      90      - 000477      91      - 000634
99      - 000552      107     - 000413      108     - 000206      109     - 000055      300     - 000553      301     - 000571
1002    - 000611

BLOCK NAMES AND LENGTHS
UNIV    - 000030/01   AERO    - 000020/02   XFLATE  - 000072/03   LAND    - 000003/04   PAST    - 000005/05   EBFGRD  - 000003/06
MARGIN  - 000003/07   TCHDWN  - 000003/08   ATMOSX  - 000002/11

VARIABLE ASSIGNMENTS
ALPAST  - 000000/05   ALPHA   - 000010/02   ANS     - 000673      CD      - 000014/02   CL      - 000013/02   DADT    - 000004/05
DAGODA  - 000002/06   DELSPL  - 000007/02   DTABS   - 000001/11   DTDT    - 000002/05   EAS     - 000710      EM      - 000010/01
ENP     - 000021/01   EYEW    - 000020/01   GAMAPP  - 000000/04   GAMMA   - 000711      HABS    - 000002/02   HALT    - 000714
HAPT    - 000000/11   INODE   - 000704      IPOWER  - 000001      ISPOIL  - 000000      JNODE   - 000705      KENG    - 000717
MROUTE  - 000706      N       - 000703      PWRMAX  - 000002/07   PWRMRG  - 000001/07   RHO     - 000015/02   RSTD    - 000000/08
RTSNK   - 000712      S       - 000000/03   STD     - 000002/08   S11PST  - 000716      S7PAST  - 000715      THETAF  - 000066/03
THRIST  - 000713      THRUST  - 000003/02   VEL     - 000000/02   VKAPP   - 000001/04   VKTD    - 000001/08   VKTS    - 000707
W       - 000006/01   WF      - 000007/01   WG      - 000023/01   XLF     - 000067/03   XLFMAX  - 000070/03   XLFMRG  - 000000/07
XLFPST  - 000001/05

START OF CONSTANTS-000575       TEMPS--000647    INDIRECTS-000671

7600 COMPILATION -- RUN76 LEVEL 5C      73/07/04.

ROUTINE COMPILES IN 045700
```

```
DERIV2
                SUBROUTINE DERIV2
                COMMON /UNIV/ NPC    ,NSC    ,IDC    ,H      ,ST     ,R      ,W      ,
               1WF    ,EM     ,VM0    ,EMM0   ,ALPHL0,CLALPH,SW     ,AR     ,B      ,
               2EYEW  ,ENP    ,TA     ,WG     ,WGS    ,KWRITE,DLMC4
               3,KSIZE
                COMMON /AERO/ VEL,QS,HABS,THRUST,IVECT, ANGLE,DELFD,DELSPL,ALPHA,
               9CX,CY,CL,CD,RHO,GRCD,IFAST
                COMMON /XTRATE/ S(52),ROC,THEMAX,THETAF,XLF,XLFMAX,K
                COMMON/LAND/ GAMAPP,VKAPP,ALPHMX
                COMMON/FAST/ ALPAST,XLFPST,DTDT,DTDTMX,DADT
                K = 1
     2          QS = 0.5*RHO*SW*S(4)*S(4)
     5          VEL = S(4)
          C
          C   LIMIT ALPHA TO .LE. ALPHMX
          C
     7        1 IF(ALPHA .GT. ALPHMX) ALPHA = ALPHMX
          C
          C   CALL AERODYN TO FIND CX AND CY
          C
    12          CALL AERODYN
          C
          C   CHECK LOAD FACTOR.  IF XLF .GT. XLFMAX, REDUCE ALPHA (2-3).
          C
    13        2 XLF = (QS*CY)/(W*COS(S(5)))
    20          IF(XLF .LE. XLFMAX)GO TO 3
    23          ALPHA = ALPHA - 0.05
    24          NERPT = 1
    25          IF(ALPHA .LT. -ALPHMX      ) GO TO 99
    31          GO TO 1
          C
          C   COMPUTE TIME RATE OF CHANGE OF FLIGHT PATH ANGLE S(9) AND THE PITCH
          C   RATE DTDT.  IF DTDT .GT. MAXIMUM ALLOWED, DTDTMX, REDUCE ALPHA (3-4)
          C
    31        3 S(9) = (32.2/(W*S(4)))*(CY*QS - W*COS(S(5)))
    41          DTDT = S(9)*57.3 + (ALPHA - ALPAST)/S(3)
    45          IF(DTDT .LE. DTDTMX)GO TO 4
    50          ALPHA = ALPHA - 0.05
    52          NERPT = 2
    53          IF(ALPHA .LT. -ALPHMX)GO TO 99
    56          GO TO 1
          C
          C   COMPUTE ACCELERATION ALONG FLIGHT PATH AND VELOCITY COMPONENTS
          C
    56        4 S(8) = (32.2/W)*(-CX*QS - W*SIN(S(5)))
    65           S(10) = S(4)*COS(S(5))
    70          S(11) = S(4)*SIN(S(5))
    74          RETURN
    74       99 WRITE(6,66)NERPT
             66 FORMAT(1X,32H*** ERROR IN DERIV2 WITH NERPT =,I2)
   102          K = 9
   103          RETURN
   104          END
```

```
ROLL
            SUBROUTINE ROLL(MODE,PWRSET,VKTD,STD,TMTD,ABAR,TDELAY,XMUBRK,TS,
           9IWRITE)
      C
      C   SUBROUTINE ROLL COMPUTES THE GROUND ROLL OF THE LANDING MANEUVER.
      C   VKTD = INITIAL SPEED, STD = INITIAL DISTANCE, TMTD = INITIAL TIME.
      C
            EXTERNAL DERIV4
            REAL MU
            COMMON /UNIV/ NPC    ,NSC    ,IDC    ,H      ,ST     ,R      ,W      ,
           1WF     ,EM     ,VMO    ,EMMO   ,ALPHLO,CLALPH,SW     ,AR     ,B      ,
           2EYEW   ,ENP    ,TA     ,WG     ,WGS    ,KWRITE,DLMC4
           3,KSIZE
            COMMON /AERO/ VEL,QS,HABS,THRUST,TVECT, ANGLE,DELFD,DELSPL,ALPHA,
           9CX,CY,CL,CD,RHO,GRCD,IFAST
            COMMON /XROLL/ T(30),NEQ,MU,NREV
            COMMON /XLAND/ DTPENG,PWRIDL,TBRK,TSPL,TFLP,TREV,TOFF
            COMMON /ATMOSX/ HAPT,DTABS
            DIMENSION ANS(8)
            SXX7 = 0.0
      C
      C   ENGINE THROTTLING SET UP
      C
   14       DTPUP = (1.0 - PWRSET)/DTPENG
   17       DTPDWN = (PWRSET - PWRIDL)/DTPENG
   21       PJ = PWRSET
   21       TJDWN = TDELAY + TOFF
   24       TJREV = TDELAY + TREV
   25       NREV = 0
   26       IROUTE = 1
   27       IROUT1 = 1
   30       IROUT2 = 1
   31       IROUT3 = 1
   31       IROUT4 = 1
   32       IROUTJ = 1
      C
      C   MODE = 1 - RUNGE-KUTTA INTEGRATION OF GROUND ROLL.
      C   MODE = 2 - LANDING DISTANCE CALCULATED ASSUMING AN AVERAGE DECELERATION.
      C
   33       GO TO (100,200),MODE
   40   200 STOP = (1.0/(64.4*ABAR))*(VKTD*1.69)**2 + 1.69*VKTD*TDELAY
   46       TS = STD + STOP
   50       RETURN
      C
      C   INTEGRATION VARIABLES
      C     T(1) = NUMBER OF EQUATIONS
      C     T(2) = TIME (SEC.)
      C     T(3) = TIME INTERVAL, STEP SIZE (SEC.)
      C     T(4) = VELOCITY (FT./SEC.)
      C     T(5) = DISTANCE (FT.)
      C     T(6) = ACCELERATION (FT./SEC.**2)
      C
   51   100 NEQ = 2
   52       T(1) = 2
   53       T(2) = 0.0
   54       T(3) = 0.1
   56       T(4) = VKTD*1.69
```

```
ROLL
      57          T(5) = 0.0
      60          T5 = STD
      61          ZERO = 0.0
      62          MU = 0.02
      64          CALL ATMOS(HAPT,DTABS,ANS)
      67          RHO = ANS(3)
      70          HABS = 0.0
      71          CALL INTS(T,NEQ,2,1.,1.,1.,1.,1.,1., DERIV4)
     103          ACCEL = T(6)
     104          NCOUNT = 0
                C
                C   MAIN INTEGRATION LOOP (101-190)
                C
     111      101 NCOUNT = NCOUNT + 1
     113          EM = T(4)*0.592087/ANS(4)
     115          IF(IROUTE .EQ. 2)GO TO 105
                C
                C   REDUCE FUSELAGE ANGLE AT A RATE OF 5.0 DEGREES/SEC. UNTIL 0.0
                C
     120          DTDT = -5.0
     121          ALPHA = ALPHA -5.0*T(3)
     124          IF(ALPHA .GT. EYEW)GO TO 105
     127          ALPHA = EYEW
     127          DTDT = 0.0
     130          IROUTE = 2
     131      105 IF(T(2) .LT. TDELAY)GO TO 150
                C
                C   BRAKE APPLICATION (105-106)
                C
     134          IF(IROUT1 .EQ. 2)GO TO 120
     136          IF(T(2) .LT. (TDELAY + TBRK))GO TO 120
     141          IF(IWRITE .GT. 1 .AND. IROUTJ .EQ. 1)WRITE(6,106)XMUBRK
              106 FORMAT(5X,60HAPPLY BRAKES AND/OR SPOILERS AND/OR REVERSE THRUST -
                 9MUBRK =,F6.3)
     162          IROUTJ = 2
     163          MU = XMUBRK
     164          IROUT1 = 2
                C
                C   SPOILERS DEPLOYED (120-130)
                C
     166      120 IF(IROUT2 .EQ. 2)GO TO 130
     170          IF(T(2) .LT. (TDELAY + TSPL))GO TO 130
     174          DELSPL = DELSPL + 9.0
     176          IF(DELSPL .LT. 90.)GO TO 130
     200          DELSPL = 90.
     201          IROUT2 = 2
                C
                C   FLAPS RETRACTED (130-140)
                C
     202      130 IF(IROUT3 .EQ. 2)GO TO 140
     204          IF(T(2) .LT. (TDELAY + TFLP))GO TO 140
     210          DELFD = DELFD - 1.0
     212          IF(DELFD .GT. 0.0)GO TO 140
     214          DELFD = 0.0
     214          IROUT3 = 2
                C
```

```
ROLL
      C   CHANGES TO THROTTLE SETTING (140-150)
      C
  215   140 IF(IROUT4 .EQ. 2)GO TO 150
  217       IF(IREV .EQ. 1)GO TO 145
      C
      C   THROTTLE DOWN TO IDLE (140-145)
      C
  221         IF(T(2) .LT. (TDELAY + TOFF))GO TO 150
  225         PWRSET = PJ - DTPDWN*(T(2) - TJDWN)
  230         IF(PWRSET .GT. PWRIDL)GO TO 150
  234         PWRSET = PWRIDL
  234         IROUT4 = 2
  235         GO TO 150
      C
      C   ADVANCE THROTTLE FOR REVERSE THRUST (145-150)
      C
  236   145 IF(T(2) .LT. (TDELAY + TREV))GO TO 150
  242         NREV = 1
  243         PWRSET = PJ + DTPUP*(T(2) - TJREV)
  247         IF(PWRSET .LT. 1.0)GO TO 150
  251         PWRSET = 1.0
  252         IROUT4 = 2
  253   150 CALL INTM(T,NEQ,2,1.,1.,1.,1.,1., DERIV4)
  265         TS = TS + T(3)*T(4)
  270         ACCEL = ACCEL + T(6)
      C
      C   CALL ENGINE WITH PWRSET(KENG=2) AND GET THRUST AND FUEL FLOW.
      C
  272         KENG = 2
  273         CALL ENGINE(MAPT,DTABS,EM,PWRSET,WF,KENG)
  303         W = W - WF*ENP*T(3)/3600.
      C
      C   LIMIT ON GROUND ROLL OF 60. SECONDS.
      C
  307         IF(T(2) .GT. 60.0) GO TO 199
  317         IF(T(4) .LT. 1.0)GO TO 190
  321         IF(IWRITE .EQ. 1)GO TO 101
  323         IF(NCOUNT .LT. 10)GO TO 101
  326         NCOUNT = 0
  326         TIME = TMTO + T(2)
  327         VKTS = T(4)/1.69
  332         EAS = VKTS * SQRT(ANS(7))
  337         THETAF = ALPHA - EYEW
  340         THRIST = THRUST * ENP
  343         WRITE(6,1002)TIME,TS,W,ZERO,VKTS,EAS,EM,T(6),CL,CD,ALPHA,ZERO
             9,ZERO,ZERO,THRIST,THETAF,DTDT
 1002 FORMAT(1X,F5.1,F9.1,F9.0,F9.1,F7.1,F6.1,F6.3,F7.2,F7.3,F7.4,2F7.2,
             9F8.1,F6.2,F9.0,F6.1,F8.3)
  421         GO TO 101
  421   190 STD = TS
  422         TIME = TIME + 0.2
  424         WRITE(6,1002)TIME,TS,W,ZERO,ZERO,ZERO,ZERO,T(6),CL,CD,ALPHA,ZERO,
             9ZERO,ZERO,ZERO,ZERO,ZERO
      C
      C   COMPUTE AVERAGE DECELERATION ABAR FOR MODE = 1
      C
```

```
ROLL
    473         ABAR = -(ACCEL/(T(2)*10.0))/32.2
    502         RETURN
    503     199 WRITE(6,198)
            198 FORMAT(5X,23H*** FAILED IN ROLL-LAND)
    513         RETURN
    513         END


SUBPROGRAM LENGTH

00644

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
100    - 000052    101    - 000112    105    - 000132    106    - 000531    120    - 000167    130    - 000203
140    - 000216    145    - 000237    150    - 000254    190    - 000422    198    - 000565    199    - 000504
200    - 000041    1002   - 000550

BLOCK NAMES AND LENGTHS
UNIV   - 000030/01 AERO   - 000020/02 XROLL  - 000041/03 XLAND  - 000007/04 ATMOSX - 000002/05

VARIABLE ASSIGNMENTS
ACCEL  - 000632    ALPHA  - 000010/02 ANS    - 000604    CD     - 000014/02 CL     - 000013/02 DELFD  - 000006/02
DELSPL - 000007/02 DTABS  - 000001/05 DTDT   - 000634    DTPDWN - 000616    DTPENG - 000000/04 DTPUP  - 000615
EAS    - 000641    EM     - 000010/01 ENP    - 000021/01 EYEW   - 000020/01 HABS   - 000002/02 HAPT   - 000000/05
IREV   - 000635    IROUTE - 000622    IROUTJ - 000627    IROUT1 - 000623    IROUT2 - 000624    IROUT3 - 000625
IROUT4 - 000626    IWRITE - 000003    KENG   - 000636    MU     - 000037/03 NCOUNT - 000633    NEO    - 000036/03
NREV   - 000040/03 PJ     - 000617    PWRIDL - 000001/04 RHO    - 000015/02 STOP   - 000630    SXX7   - 000614
T      - 000000/03 TBRK   - 000002/04 TDELAY - 000000    TFLP   - 000004/04 THETAF - 000642    THRIST - 000643
THRUST - 000003/02 TIME   - 000637    TJDWN  - 000620    TJREV  - 000621    TOFF   - 000006/04 TREV   - 000005/04
TSPL   - 000003/04 T5     - 000002    VKTS   - 000640    W      - 000006/01 WF     - 000007/01 XMUBRK - 000001
ZERO   - 000631

START OF CONSTANTS-000516       TEMPS--000572    INDIRECTS-000603

7600 COMPILATION -- RUN76 LEVEL 5C      73/07/04.

ROUTINE COMPILES IN 045400
```

```
DERIV4
            SUBROUTINE DERIV4
          C
          C   SUBROUTINE DERIV4 COMPUTES THE DERIVATIVES FOR INTEGRATION IN ROLL.
          C
            REAL MU
            COMMON /UNIV/ NPC    ,NSC    ,IDC    ,H      ,ST     ,R      ,W      ,
           1WF     ,EM     ,VM0    ,EMM0   ,ALPHL0,CLALPH,SW     ,AR     ,B      ,
           2EYEW   ,ENP    ,TA     ,WG     ,WGS    ,KWRITE,DLMC4
           3,KSIZE
            COMMON /AERO/ VEL,QS,HABS,THRUST,TVECT, ANGLE,DELFD,DELSPL,ALPHA,
           9CX,CY,CL,CD,RHO,GRCO,IFAST
            COMMON /XROLL/ T(30),NEQ,MU,NREV
            QS = 0.5*RHO*SW*T(4)*T(4)
      4     IF(QS .EQ. 0.)QS = 0.1
          C
          C   IF USING REVERSE THRUST (NREV .EQ. 1) CALL SUBROUTINE REVRSE TO
          C   OBTAIN CX AND CY IN THE REVERSE THRUST MODE.
          C
      7     IF(NREV .EQ. 0)CALL ARODYN
     11     IF(NREV .EQ. 1)CALL REVRSE
     14     T(6) = (32.2/W)*(-W*MU + QS*(CY*MU - CX))
     23     T(7) = T(4)
     24     RETURN
     25     END
```



```
SUBPROGRAM LENGTH

00044

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

BLOCK NAMES AND LENGTHS
UNIV   - 000030/01  AERO   - 000020/02  XROLL  - 000041/03

VARIABLE ASSIGNMENTS
CX     - 000011/02  CY     - 000012/02  MU     - 000037/03  NREV   - 000040/03  QS     - 000001/02  RHO    - 000015/02
SW     - 000015/01  T      - 000000/03  W      - 000006/01

START OF CONSTANTS-000030       TEMPS--000034    INDIRECTS-000044

7600 COMPILATION -- RUN76 LEVEL 5C      73/07/04.

ROUTINE COMPILES IN 044000
```

```
LAND1
                SUBROUTINE LAND1(VKAPP,GAMAPP,XLFMAX,SINKTD,HAPP,ABAR,TDELAY,DIST)
          C
          C   SUBROUTINE LAND1 IS USED IN FINDING THE REQUIRED APPRAOCH SPEED IN
          C   SUBROUTINE LANDNG.  IT CALCULATES THE LANDING DISTANCE ASSUMING A
          C   CONSTANT SPEED AND LOAD FACTOR FLARE.
          C   VKAPP = SPEED(KNOTS)              GAMAPP = FLIGHT PATH ANGLE(DEG.)
          C   XLFMAX = MAX. LOAD FACTOR         SINKTD = SINK RATE AT TD. (FT./SEC.)
          C   ABAR = AVERAGE DECELERATION(G'S) TDELAY = TIME DELAY (SEC.)
          C
                GAMX = GAMAPP/57.3
       15       VAPP = VKAPP * 1.69
       16       GAMTD = SINKTD/VAPP
       17       R = (VAPP*VAPP)/(32.2*(XLFMAX - 1.0))
       22       HFLAR = VAPP*VAPP*(GAMX*GAMX - GAMTD*GAMTD)/(64.4*(XLFMAX-1.0))
       27       IF(HFLAR .LT. HAPP)GO TO 20
       32       DLGL = 0.0
       32       DLTR =   R*(SQRT((2.*HAPP/R)+ GAMTD*GAMTD) - GAMTD)
       46       GO TO 25
       47    20 DLGL = HAPP/TAN(GAMX)
       56       DLTR = ((R*GAMX)/2.)*((1.0 - GAMTD/GAMX)**2)
       64    25 DDELAY = VAPP*TDELAY
       66       DGRND = (1.0/(64.4 * ABAR))*VAPP*VAPP
       71       DIST = DLGL + DLTR + DDELAY + DGRND
       75       RETURN
       76       END
```



```
SUBPROGRAM LENGTH

00144

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
20     - 000050     25     - 000065

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
DDELAY - 000142     DGRND  - 000143     DIST   - 000001     DLGL   - 000140     DLTR   - 000141     GAMTD  - 000135
GAMX   - 000133     HFLAR  - 000137     R      - 000136     TDELAY - 000000     VAPP   - 000134

START OF CONSTANTS-000101       TEMPS--000110    INDIRECTS-000133

7600 COMPILATION -- RUN76 LEVEL 5C      73/07/04.

ROUTINE COMPILES IN 044000
```

```
ZERO
            SUBROUTINE ZERO(NER,EN,ALT,DTABS,KENG,PWRSET,QUESS)
      C
      C   SUBROUTINE ZERO COMPUTES THE REQUIRED VALUES OF THRUST PER ENGINE AND
      C   ANGLE OF ATTACK FOR ZERO ACCELERATION ALONG AND NORMAL TO THE FLIGHT
      C   PATH, DV/DT AND DG/DT, RESPECTIVELY.   THE SUBROUTINE CALLS ENGINE
      C   WITH KENG = 1 WITH A VALUE OF THRUST TO OBTAIN THE POWER SETTING.
      C   THE OUTER LOOP VARIES THE THRUST, WITH TWO INNER LOOPS VARYING ALPHA
      C   FROM ALPHMX TO -ALPHMX.  FOR A FIXED VALUE OF THRUST (OUTER LOOP),
      C   SUBROUTINES DVDT(ACCELERATION ALONG FLIGHT PATH), AND DGDT(ACCELERATION
      C   NORMAL TO FLIGHT PATH), ARE CALLED WITH THE VALUES OF ALPHA (INNER
      C   LOOP).  THESE TWO FUNCTION SUBROUTINES CALL AERODYN.  SUBROUTINES
      C   ITRJVB AND ZEROX ARE BOTH ZERO-FINDERS.
      C
            EXTERNAL DVDT,DGDT
            COMMON /UNIV/ NPC    ,NSC    ,IDC    ,M      ,ST     ,R      ,W      ,
           1WF    ,EM     ,VMO    ,EMMO   ,ALPHL0,CLALPH,SW     ,AR     ,B      ,
           2EYEW   ,ENP    ,TA     ,WG     ,WGS    ,KWRITE,DLMC4
           3,KSIZE
            COMMON /AERO/ VEL,QS,HABS,THRUST,TVECT, ANGLE,DELFD,DELSPL,ALPHA,
           9CX,CY,CL,CD,RHO,GRCD,IFAST
            COMMON/LAND/ GAMAPP,VKAPP,ALPHMX
            EM = EN
     11     NER = 1
     12     EPS = 0.001
     13     ALPHMN = - ALPHMX
      C
      C   FIRST GUESS AT THRUST
      C
     15     TFIRST = (ST/ENP) * QUESS
     20   1 THRUST = TFIRST
     21     F = 0.95
     23     TOL = ST/(ENP*1000.)
     25     IF(TOL .LT. 5.0)TOL = 5.0
     31     BU = W/ENP
     33     BL = W / 100.
     34     JC = 0
     35     JX = 0
     35     ERROR = 999.
      C
      C   OUTER LOOP WITH THRUST (2-10)
      C
     37   2 IF (JX .EQ. 0) ERRM1 = ERROR
      C
      C     CALL ENGINE
     42     CALL ENGINE(ALT,DTABS,EM,PWRSET,WF,KENG)
      C
      C
      C   FIND VALUE OF THRUST SUCH THAT DV/DT IS NEGATIVE AT ALPHA = ALPHMX,
      C   WITH A REDUCTION IN ALPHA RESULTING IN LESS DRAG, DV/DT WILL CHANGE
      C   SIGN(I.E. A BOUNDED INTERVAL IN WHICH DV/DT = 0.0 WILL BE FOUND).
      C
     46     ALPXUP = ALPHMX
     47     XUP = DVDT(ALPXUP)
     51     IF (XUP .LT. 0.0) GO TO 3
     57   7 TFIRST = TFIRST/1.05
     61     IF (TFIRST .LT. BL) GO TO 99
```

```
ZERO
     63         GO TO 1
                C
                C   SEARCH FOR DV/DT = 0.0 FROM ALPHMX TO -ALPHMX (3-4)
                C
     64        3 ALPXLO = ALPXUP - 2.0
     66          IF(ALPXLO .LT. ALPHMN)GO TO 98
     71          XLO = DVDT(ALPXLO)
                C
                C   CHECK TO SEE IF DV/DT = 0.0 IS BOUNDED.
                C
     72          IF (XLO * XUP .LE. 0.0) GO TO 4
    100          ALPXUP = ALPXLO
    101          XUP = XLO
    103          GO TO 3
                C
                C   ONCE ALP.. INTERVAL FOR DV/DT = 0.0 IS BOUNDED, CALL ZEROX TO FIND
                C   ALPHA1, THE VALUE OF ALPHA FOR DV/DT = 0.0.
                C
    103        4 ALPHA1 = ZEROX(ALPXLO,ALPXUP,DVDT,EPS)
                C
                C   SEARCH FOR DG/DT SAME AS ABOVE SEARCH FOR DV/DT = 0.0.  IF DG/DT = 0.0
                C   AT ALPHA = ALPHMX, PROGRAM FAILS.  ALPHA2 IS REQUIRED VALUE OF ALPHA
                C   FOR DG/DT = 0.0.
                C
    106          ALPXUP = ALPHMX
    110          YUP = DGDT(ALPXUP)
    112          IF (YUP .LT. 0.0) GO TO 99
    120        5 ALPXLO = ALPXUP - 2.0
    122          IF(ALPXLO .LT. ALPHMN)GO TO 7
    125          YLO = DGDT(ALPXLO)
    126          IF (YLO * YUP .LE. 0.0) GO TO 6
    134          ALPXUP = ALPXLO
    135          YUP = YLO
    137          GO TO 5
    137        6 ALPHA2 = ZEROX(ALPXLO,ALPXUP,DGDT,EPS)
                C
                C   FIND ERROR BETWEEN ALPHA1 AND ALPHA2.
                C
    142          ERROR = (ALPHA1 - ALPHA2) / ALPHMX
    145          IF (ABS(ERROR) .LT. 0.005) GO TO 10
    154          IF (ABS(ERROR) .LT. 0.01  .AND. JC .GT. 25) GO TO 10
    165          IF (JX .EQ. 2) GO TO 10
                C
                C   MAKE NEW GUESS AT THRUST AND REPEAT DV/DT AND DG/DT LOOPS UNTIL
                C   ALPHA1 = ALPHA2 (WITHIN TOLERANCES).
                C
    167          CALL ITRJVB(ERROR,ERRM1,THRUST,F,TOL,JC,JX)
    175          IF (JC .GT. 30) GO TO 99
    204          GO TO 2
    204       10 ALPHA = ALPHA1
    206          RETURN
    206       98 NER = 9
    207          WRITE(6,101)THRUST,ALPHA,ERROR,ERRM1,JC
              101 FORMAT(/,2X,18H*** FAILED IN ZERO,/,5X,51HNO SOLUTION FOR DVDT =TH
                 9RUST,ALPHA,ERROR,ERRM1,JC =,2F9.2,2F7.4,I3)
    231          RETURN
```

```
ZERO
    231      99 NER = 9
    232         WRITE (6,100) TFIRST, YUP, ERROR, ERRM1, ALPHA1, ALPHA2, JC
             100 FORMAT (/ 2X, 18H*** FAILED IN ZERO,/,5X, 42H TFIRST,YUP,ERROR,ERR
                M1,ALPHA1,ALPHA2,JC =  , 6F12.4, I4)
    260         RETURN
    260         END


SUBPROGRAM LENGTH

00365/01

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1       - 000021     2       - 000040     3       - 000065     4       - 000104     5       - 000121     6       - 000140
7       - 000060     10      - 000205     98      - 000207     99      - 000232     100     - 000313     101     - 000277

BLOCK NAMES AND LENGTHS
UNIV    - 000030/01  AERO    - 000020/02  LAND    - 000003/03

VARIABLE ASSIGNMENTS
ALPHA   - 000010/02  ALPHA1  - 000361     ALPHA2  - 000364     ALPHMN  - 000343     ALPHMX  - 000002/03  ALPXLO  - 000357
ALPXUP  - 000355     BL      - 000350     BU      - 000347     EM      - 000010/01  ENP     - 000021/01  EPS     - 000342
ERRM1   - 000354     ERROR   - 000353     F       - 000345     JC      - 000351     JX      - 000352     QUESS   - 000000
ST      - 000004/01  TFIRST  - 000344     THRUST  - 000003/02  TOL     - 000346     W       - 000006/01  WF      - 000007/01
XLO     - 000360     XUP     - 000356     YLO     - 000363     YUP     - 000362

START OF CONSTANTS-000263       TEMPS--000326    INDIRECTS-000342

7600 COMPILATION -- RUN76 LEVEL 5C      73/07/04.

ROUTINE COMPILES IN 044600
```

```
DVDT
                FUNCTION DVDT(ALPHAX)
          C
          C   FUNCTION SUBROUTINE DVDT COMPUTES DV/DT AS A FUNCTION OF ALPHA.
          C   REQUIRES VARIABLES W,QS,CX,GAMAPP
          C
                COMMON /UNIV/ NPC    ,NSC    ,IDC    ,M      ,NS     ,R      ,W      ,
               1WF     ,EM     ,VM0    ,EMM0   ,ALPHL0,CLAL M,...      ,AR     ,B      ,
               2EYEW   ,ENP    ,TA     ,WG     ,WGS    ,KWP...4
               3,XSIZE
                COMMON /AERO/ VEL,QS,HABS,THRUST,TVECT, ...LE,DELFD,DELSPL,ALPHA,
               9CX,CY,CL,CD,RHO,GRCD,IFAST
                COMMON/LAND/ GAMAPP,VKAPP,ALPHMX
                GAMX = -GAMAPP * .0174533
        4       ALPHA = ALPHAX
        5       CALL AROUYN
        6       DVDT = (32.2 / W) * (-CX * QS - W*SIN(GAMX))
       15       RETURN
       16       END


SUBPROGRAM LENGTH

00036

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

BLOCK NAMES AND LENGTHS
(NIV    - 000030/01  AERO    - 000020/02  LAND    - 000003/03

VARIABLE ASSIGNMENTS
ALPHA   - 000010/02  CX      - 000011/02  DVDT    - 000034     GAMAPP - 000000/03  GAMX    - 000035     QS      - 000001/02
W       - 000006/01

START OF CONSTANTS-000021       TEMPS--000024    INDIRECTS-000034

7600 COMPILATION -- RUN76 LEVEL 5C      73/07/04.

ROUTINE COMPILES IN 044000
```

```
DGDT
                FUNCTION DGDT(ALPHAY)
          C
          C    FUNCTION SUBROUTINE DGDT COMPUTES DG/DT AS A FUNCTION OF ALPHA.
          C    REQUIRES VARIABLES W,QS,VEL,CY,GAMAPP
          C
                COMMON /UNIV/ NPC    ,NSC    ,IDC    ,H      ,ST     ,R      ,W      ,
               1WF    ,EM     ,VMO    ,EMMO   ,ALPHLO,CLALPH,SW     ,AR     ,B      ,
               2EYEW  ,ENP    ,TA     ,WG     ,WGS    ,KWRITE,DLMC4
               3,KSIZE
                COMMON /AERO/ VEL,QS,HABS,THRUST,TVECT, ANGLE,DELFD,DELSPL,ALPHA,
               9CX,CY,CL,CD,RHO,GRCD,IFAST
                COMMON/LAND/ GAMAPP,VKAPP,ALPHMX
                GAMX = -GAMAPP * .0174533
            4   ALPHA = ALPHAY
            5   CALL AERODYN
            6   DGDT = (32.2 / (W*VEL))* (CY * QS - W * COS(GAMX))
           16   RETURN
           17   END


SUBPROGRAM LENGTH

00041

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

BLOCK NAMES AND LENGTHS
UNIV   - 000030/01  AERO   - 000020/02  LAND   - 000003/03

VARIABLE ASSIGNMENTS
ALPHA  - 000010/02  CY     - 000012/02  DGDT   - 000037     GAMAPP - 000000/03  GAMX   - 000040     QS     - 000001/02
VEL    - 000000/02  W      - 000006/01

START OF CONSTANTS-000022       TEMPS--000025    INDIRECTS-000037

7600 COMPILATION -- RUN76 LEVEL 5C      73/07/04.

ROUTINE COMPILES IN 044000
```