NASA Technical Memorandum 81857

NASA-TM-81857 19830002844

OPDOT: A COMPUTER PROGRAM FOR THE OPTIMUM PRELIMINARY DESIGN OF A TRANSPORT AIRPLANE

Steven M. Sliwa and P. Douglas Arbuckle

FEDD REMOVED PER
NASA LTR DTD 10-20-82,
S/J.G. ROSS

September 1980

~~FOR EARLY DOMESTIC DISSEMINATION~~

~~Because of its significant early commercial potential, this information, which has been developed under a U.S. Government program, is being disseminated within the United States in advance of general publication. This information may be duplicated and used by the recipient with the express limitation that it not be published. Release of this information to other domestic parties by the recipient shall be made subject to these limitations.~~

~~Foreign release may be made only with prior NASA approval and appropriate export licenses. This legend shall be marked on any reproduction of this information in whole or in part.~~

~~Review for general release September 30, 1981~~

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

OPDOT: A COMPUTER PROGRAM FOR THE OPTIMUM PRELIMINARY DESIGN OF A TRANSPORT AIRPLANE

Steven M. Sliwa and P. Douglas Arbuckle

ABSTRACT

A description of a computer program, OPDOT, for the optimal preliminary design of transport aircraft is given. OPDOT utilizes constrained parameter optimization to minimize a performance index (e.g. direct operating cost per block hour) while satisfying operating constraints. The approach in OPDOT uses geometric descriptors as independent design variables. The independent design variables are systematically iterated to find the optimum design. The technical development of the program is provided and a program listing with sample input and output are utilized to illustrate its use in preliminary design. This is not meant to be a user's guide, but rather a description of a useful design tool developed for studying the application of new technologies to transport airplanes.

INTRODUCTION

When new technologies in aircraft design, fabrication and operation are evaluated, current practice requires engineering judgment in making compromises. An engineer utilizes a combination of limited analyses, experience and intuition to combine new technologies (e.g., aerodynamics, controls or structures) to maximize the benefits. This approach is imprecise because it involves extrapolating experience from previous designs and because the improvements are usually made to augment multiple, ill-defined criteria (e.g., weight, cost, or performance). To properly evaluate any changes in design concepts, the airplane configuration should be allowed to evolve to optimize a single, well-defined performance index.

This report describes OPDOT (Optimum Preliminary Design of Transports), a computer program written to perform preliminary design and evaluation of transport aircraft using nonlinear programing techniques. A set of independent design variables is iterated upon until a minimum of a performance index which satisfies a series of constraint functions has been calculated. The design variables usually consist of geometry characteristics and mission parameters, while the constraint functions include, for example, regulatory performance requirements and handling quality design criteria. A slightly modified, constrained sequential optimizer is utilized in the program.

This program, therefore, allows the evaluation of new technologies incorporated into an aircraft design in an optimal fashion. The degree of detail in the analyses when the performance function and the constraint functions are evaluated is at the preliminary design or classical aeronautics level. That is, the precision in some phases of the calculations is expected to be as poor as 5-10 percent. Hence, whereas the predictive capabilities are expected to be marginal, the accuracy of the relative comparisons of designs is expected to be good.

This report, which includes a program listing, sample input and sample output, is a description of a useful analytical tool for analyzing the effects of new technologies on the preliminary design and sizing of transport airplanes. It describes the methods of calculation, program organization and some of the various options available, but it is not meant to be a comprehensive user's manual. The program code was written to expeditiously obtain answers for a study of the impact of active controls upon transport design. This limited the amount of effort that could be spent on developing user flexibility and on integrating into the program a high degree of self-annotation.

SYMBOLS

A	wing aspect ratio
A_t	horizontal tail aspect ratio
b	span, m
$\bar{b}$	ratio of tail span to wing span, b_t/b_w
B	Brequet range factor, $\frac{M\ L/D}{c}$
c	specific fuel consumption
$\bar{c}$	mean aerodynamic chord of wing
c_i	ith constraint value
C_C	chordwise force coefficient
C_D	drag coefficient, drag/$\bar{q}$S
C_L	lift coefficient, lift/$\bar{q}$S
C_{L_A}	approach lift coefficient
C_{L_t}	tail lift coefficient
C_{L_o}	design lift coefficient for wing airfoil section

$C_{L_{so}}$	stall lift coefficient with full flaps
$C_{L_{TO}}$	lift coefficient at takeoff
C_{L_2}	lift coefficient during second segment climb
C_m	pitching moment coefficient, pitching moment/$\bar{q}S\bar{c}$
C_N	normal force coefficient
C_T	thrust coefficient
f	unaugmented performance index
F	augmented performance index
$\bar{g}_i$	ith constraint function
h_t	vertical tail height, m
i	angle of incidence, deg
K	penalizing weight
K_θ	pitch feedback gain, $\partial\delta/\partial\theta$
$K_{\dot{\theta}}$	pitch rate feedback gain, $\partial\delta/\partial_{\dot{\theta}}$
ℓ_t	distance from wing aerodynamic center to tail aerodynamic center, m
L/D	glide ratio
m	mass (slugs)
M	Mach number
M_q	pitching moment derivative due to pitching velocity, sec^{-1}
M_w	pitching moment derivative due to unit vertical velocity, sec^{-1}
$M_{\dot{w}}$	pitching moment derivative due to vertical acceleration, sec^{-1}
M_σ	pitching moment derivative due to control deflection, sec^{-1}

n	number of engines
n_Z/α	airplane vertical gain, g's/rad
N_i	ith constraint normalizing factor, $\frac{\lvert S_{\ell_i}\rvert + \lvert S_{u_i}\rvert}{2}$
p	design parameter (constant for optimization)
$\bar{q}$	dynamic pressure
R	airplane range, kilometers
S_{ℓ_i}, S_{u_i}	lower and upper boundaries of ith constraint
S_t	tail area, m^2
S_w	wing area, m^2
T	thrust at altitude, N
T_I	installed thrust at sea level, N
TOP	Take-off parameter
U_o	reference velocity, m/sec
V_{cr}	cruise velocity, m/sec
V_{ℓ_i}, V_{u_i}	lower and upper boundaries of ith independent design variable
W	aircraft weight at altitude, N
W_{TO}	maximum aircraft weight at take-off, N
W_1	initial aircraft weight during cruise segment, N
W_2	final aircraft weight after cruise segment, N
X_i	ith independent design variable
$\bar{X}_{ac}$	% MAC from datum to c.g. in x-direction

Z_i	independent design variable transformed
Z_δ	dimensional vertical force derivative due to elevator deflection, sec^{-1}
$\bar{Z}_{ac}$	% MAC from datum to c.g. in z-direction
$\bar{Z}_t$	height of thrust vector from c.g. (% MAC)
ε_1	Oswald's efficiency factor for $0 \leq C_L \leq C_{L_o}$
ε_2	Oswald's efficiency factor for $C_L > C_{L_o}$
γ	flight path angle, rad
ω_{sp}	short period frequency, sec^{-1}
σ	Munk's interference factor
t	tail efficiency, $\frac{\bar{q}_t}{q_\infty}$

Superscript

*	optimum

Subscripts

ac	aerodynamic center
cr	cruise
q	pitch rate
t	tail
u	velocity
w	wing
α	angle of attack
$\dot{\alpha}$	rate of change in angle of attack

δ elevator deflection

PROGRAM DESCRIPTION

General

The overall flow of the program is depicted in figure 1. A set of starting values for the selected independent design variables and design constants is input and used to initialize the optimizer and the data base. Initially, the program was written with seven independent variables (wing area, wing aspect ratio, fuselage length, horizontal tail area, horizontal tail aspect ratio, aft-most center-of-gravity position, and installed thrust), but it has the inherent capability to handle more and has successfully converged with thirteen. Typical design constants include nonvarying geometries, mission parameters, economic constants, nonlinear aerodynamics data and some levels of technology. An extensive list of design constants that were used in one study is shown in Appendix VII.

Design constants are prime candidates for being changed to independent design variables. Both design constants and independent design variables are held constant for each call to the performance function evaluation routines. Independent design variables are typically altered each function call by the optimizer, while design constants are not allowed to vary for the entire optimization. A method for augmenting the set of independent design variables with design constants will be described in a later section.

The inputs (the current value of independent design variables and the design constants) are utilized by a sequence of subroutines that calculate a performance index which is selected by the user. Typically, minimum direct operating cost per block hour is chosen, but minimum direct operating cost per flight, maximum return-on-investment per year, minimum income required for a 15 percent return on investment, maximum L/D and minimum take-off gross weight are also available as criteria to be optimized. During the series of subroutine calls, data is exchanged with and stored in the data base for future use. The program has been constructed in a modular fashion to allow users to replace routines with preferred versions to allow significant configuration changes or to improve the level of accuracy.

Next a series of subroutines is called to calculate the constraint functions. Those that are calculated for cruising flight utilize data stored in the data base during the performance function evaluations. Many subroutines were written in such a fashion as to provide data in slow flight configurations as well as in cruising flight. These are called to yield take-off and landing performance data. As a byproduct, the longitudinal stability derivatives are generated. These nondimensional derivatives, for both approach and cruise, are converted to dimensional derivatives and are then used to determine the roots of a fourth order linear model of the longitudinal dynamics. These roots are used to calculate the damping and frequency in the short period and phugoid modes.

The program determines which constraint functions are violated and adds a penalty term for each violation to the performance index to create an augmented performance index. The optimizer then iterates upon the design variables to minimize the augmented function. If the weights on the penalty terms are sufficiently large, the violations will be driven to zero. A convergence of the optimizer results in the minimum unaugmented performance index that satisfies the constraint functions.

Optimization Code

The optimization is performed by a sequential simplex method (Ref. 1 and 2) which utilizes a continuous penalty function. This direct search algorithm has the advantage of not using gradient evaluations, and hence does not perform poorly near "ridges" in the performance index. Additionally, the penalty scheme is independent of the number of active constraints. Its chief disadvantage is slow convergence in large regions of small gradients of the augmented performance function with respect to the independent design variables.

The general problem is formulated as follows:

Let the unaugmented performance index, f, be a function of the independent design variables, x, and design parameters, p.

$$f = fcn(x,p) \tag{1}$$

and

$$\bar{g}_i \begin{cases} = 0 & \text{if } c_i \geq S_{\ell_i} \text{ and } c_i \leq S_{u_i} \\ = S_{\ell_i} - c_i & \text{if } c_i < S_{\ell_i} \\ = c_i - S_{u_i} & \text{if } c_i > S_{u_i} \end{cases}$$

then

$$F = f + \sum_{i=0}^{m} K(\bar{g}_i/N_i)^2 \tag{2}$$

The goal is to find the minimum of the augmented function, F, with the gains, K, large.

A variable transformation (Ref. 3) is used to automatically scale the variables and apply "side" constraints, which are inequality constraints applied directly on the design variables. This resulted in a reduction in the number of iterations required for convergence.

The form of the transformation is as follows:

$$X_i = \frac{V_{u_i} - V_{\ell_i}}{2} \sin\left(\frac{\pi}{2} Z_i\right) + \frac{V_{u_i} + V_{\ell_i}}{2} \qquad (3)$$

where V_{u_i} and V_{ℓ_i} are the ith upper and lower independent design variable boundaries. So the simplex optimizer iterates on the transformed variable Z, which spans the set of allowable values of the independent design variables with the range in Z of 1 to -1. This allows consistency in step size selection and limits the allowed values of the independent design variables.

A version of the program is listed in Appendix I. The main program SIMPACT, the subroutines NELMIN and SETUP with the function FN are used to perform the optimization. Some key variables and a description of the pertinent labelled common blocks are shown in Appendices V and VI, respectively. Prior to the optimization, a series of inputs to initialize the optimization blocks is read in and XINPUT is used to initialize the aircraft data. NELMIN, the subroutine which returns the constrained minimum, is called several times (usually two) with increasing weights and diminishing convergence criteria and initial step sizes. This is to help in obtaining a satisfactory local minimum with no constraints violated and, ideally, with the active constraints resting against their boundaries.

NELMIN calls FN which returns the augmented performance index. FN calls SETUP which performs the variable transformations, obtains the unaugmented performance index, calls the constraint evaluation routines, determines the penalty terms and then assembles the augmented performance index. The unaugmented performance index is determined by calling DOCOST and the constraint functions are calculated from CNSTRN.

Evaluation of Unaugmented Performance Index

A flow diagram showing the general procedure for evaluating the unaugmented performance index is shown in figure 2. DOCOST (Included in Appendix I) assembles various cost components by first calling GEO to calculate and store some geometry constants and then calling CGCAL to assign the center-of-gravity positions for the various phases of flight and the landing gear position (if variable). Then WEIGHT is called which is used to estimate the airplane's operating weights, the amount of fuel burned during the mission and a variety of other parameters required from the cruise portion of the flight.

In WEIGHT, an initial estimate of take-off weight and fuel fraction is made. The individual weight components are determined using statistical relationships from references 4 through 6. The primary source was reference 4, but the critical components for the intended uses of the program (i.e., wing, horizontal tail and fuselage) were limited to geometric ranges to maintain validity. To improve the capability of predicting the weights of these components (e.g., at high aspect ratios) an average of values calculated from references 4 through 6 was made. After the component weights are summed, FUELCAL is used to determine the weight of the fuel required to fly the passenger mission and the reserve mission. This fuel weight is used to estimate the weight of the fuel systems.

The sum of the individual estimated weight components is compared with the initial estimate of take-off weight; and, if the difference is greater than some convergence criterion (usually about .2 Newtons), a new estimate is made and the components are summed again. This continues until the weight loop convergence criterion is satisfied. The new estimate for the gross take-off weight is made through a weighting scheme based on the number of current iterations. The total and average number of iterations is displayed to the user to provide guidance in possible programing changes in the event of slow weight loop convergence. Usually WEIGHT averages between 3 and 5 iterations per function call during an optimization run.

FUELCAL assumes a flight profile schematically illustrated in Figure 3. A fixed percentage of the total fuel burnoff is attributed to the following tasks: taxi, take-off, initial climb, climb to cruise, descent and landing. The remainder of the flight (the cruise portion) is divided into ten equal segments. During the first segment the transport is flown at a C_L for maximum range factor, B*. The initial cruise altitude is 11000 m (36000 ft), and CRUALT is called to find the desired altitude at the end of the first segment to maintain the same C_L for a new weight, while insuring the aircraft is also cruising at the desired Mach Number. The required excess thrust to generate the calculated climb gradient is then saved for future use in the constraint functions.

Segments 2 through 5 are flown in a cruise/climb mode at M_{cr} and B*, which can be calculated from classical relationships. At segment 6, however, the climb is increased so that segments 7 through 10 can be flown at cruise Mach number, M_{cr}, and 98% L/D_{max}. The cruise is backed off L/D_{max} slightly to help provide some speed stability.

Thus, as modelled above, the independent design variables only impact the cruise portion of the flight. To simulate the complex reserve mission requirement, the transport is flown for an additional 1400 kilometers (1000 nautical miles) at 9100 meters (30,000 feet) at the speed for maximum range.

CRUFUEL calculates the amount of fuel burned during each segment as well as the time required to fly it and the altitude change to satisfy the cruise/climb assumptions. As previously described, the aircraft is flown at the speed for maximum Brequet range factor during the first five segments provided the resulting Mach number is less than or equal to the desired cruise Mach number. The solution comes from classical aeronautics, for example, reference 4.

$$L/D_{B*} = .943\ L/D \qquad (4)$$

from aeronautics and assuming parabolic drag polars

$$C_{L_{B*}} = .79\ C_{L_{L/D_{max}}} \qquad (5)$$

CRUALT returns the required altitude to fly at the specified weight, lift coefficient and Mach number at the end of each segment. CRUFUEL then estimates a rate-of-climb slightly greater than that which would maintain the maximum range factor cruise for the given altitudes. The eventual goal is to achieve a cruise at 98% of maximum L/D for the last four segments of the cruise distance at the cruise Mach number. Holding the Mach number fixed results in increasing lift coefficients as altitude increases. This is continued until the airplane attains maximum L/D.

The assumed mission profile, although patently suboptimal, varies less than 3 percent in fuel consumption from some optimal profiles (Ref. 7). Given the level of accuracy of the program and the desire to compare designs rather than predict the performance of one design, this level of precision was deemed acceptable.

XLOD is used to estimate the aerodynamic performance of the airplane. The parasite drag is obtained from CDZL. CDZL performs a drag buildup by estimating the Reynolds number, friction coefficients, and various nonlinear constants as illustrated in references 4 and 8. Increments in drag are included for "crud" drag and flap deflections. XLOD then calls STABCOD to estimate the stability and control derivatives while in the indicated flight configuration. These nondimensional derivatives are obtained from a combination of empirical and analytical relations developed from references 8 through 10 for transport airplanes. Some aeroelastic correction factors are applied to the derivatives based on observations of data in references 10 through 12.

XLOD then utilizes the stability and control data as it calls TRIM. The desired airplane lift coefficient with the specified Mach number, parasite drag, flap configuration, center-of-gravity position and phase of flight are input to TRIM.

The following classical non-linear trim equations (Ref. 13) were used in TRIM to represent the normal and chordwise forces and to solve for the required tail or wing lift coefficients:

$$C_{N_t} = \frac{S_w}{S_t}\frac{\bar{c}}{\ell_t}\frac{1}{\eta_t}\left\{C_{N_w}\bar{X}_{ac} + C_{C_w}\bar{Z}_{ac} + C_{m_{ac}}\Bigg|_{\substack{\text{wing}\\\text{fuselage}\\\text{nacelle}}}\right.$$

$$\left. + C_{C_t}\frac{S_t}{S_w}\frac{h_t}{\bar{c}}\eta_t - C_T\frac{Z_t}{\bar{c}}\right\} \tag{6}$$

$$C_{C_w} = C_{D_w}\cos(\alpha_w - i_w) - C_{L_w}\sin(\alpha_w - i_w) \tag{7}$$

$$C_{N_w} = C_{L_w}\cos(\alpha_w - i_w) + C_{D_w}\sin(\alpha_w - i_w) \tag{8}$$

An iterative scheme utilizing the above equations is used whereby a new tail lift coefficient is estimated until a convergence criterion is satisfied. Direct substitution into the vertical force, horizontal force and pitching moment equation is not possible since it has been deemed inappropriate to linearize the transcendental functions. It would also require neglecting the vertical offset of the center-of-gravity from the aerodynamic center and thrust-line and neglecting the contributions due to tail drag. Typically three or four iterations are required to satisfy the trim convergence criteria ($\Delta C_{L_t} \leq .003$).

TRIM is used in one of two fashions. First, if a desired airplane lift coefficient is input, the routine iterates to find the required lift coefficients for the tail and wing. Alternatively, if a wing lift coefficient is input, the required tail lift coefficient is output along with the resulting airplane lift coefficient. The latter mode is used to determine the maximum trimmed lift coefficient for approach or take-off configurations where stalling of the wing is a concern.

The wing compressibility drag contribution is calculated in XLOD by using the empirical relationships found in reference 14, which were derived from supercritical aerodynamics wind tunnel data. The fuselage compressibility drag term is modelled from the graphs in reference 13. It should be noted that it is assumed that the fuselage is not area ruled and hence calculated drag will be pessimistic for transonic configurations ($1.0 > M_{cr} > 0.9$).

The induced drag contribution is obtained as follows:

$$C_{D_i} = \frac{C_{L_o}^{\,2}}{\pi A\varepsilon_1} + \frac{C_{L_w}^{\,2} - C_{L_o}^{\,2}}{\pi A\varepsilon_2} + \frac{2\sigma C_{L_w} C_{L_t} S_{t_1}}{\bar{b}\pi A S_w \varepsilon_1} + \frac{S_t C_{L_t}^{\,2}}{S_w \pi A_t \varepsilon_t} \tag{9}$$

The first two terms are the wing contribution including an offset for the design lift coefficient of the highly cambered wing. The third term represents the interference drag between the lift vectors of the tail and the wing. Notice how the interference term could be negative if the tail lift were downward. The fourth term is the drag contribution of the tail lift (positive for a tail load in any direction). The interference factor, σ, is a function of the gap ratio, h_t/b_w, and the span ratio, $\bar{b}$. This term is calculated from a least squares polynomial fit (Ref. 15) of the curves in reference 16.

The total drag, calculated in XLOD, is the sum of the induced drag, the drag due to elevator deflection (C_{D_δ} estimated from Ref. 13), the compressibility drag and the parasite drag. The L/D is obviously calculated as C_L/C_D. Additionally, the lift coefficient for L/D_{max} is estimated and stored for future use. XLOD, CDZL, STABCOD and TRIM are generalized to function for both cruise and approach conditions.

CRUFUEL then calls ENGINE to determine the thrust and specific fuel consumption as a function of altitude and Mach number. The engine performance comes from a normalized model of the baseline engine from reference 7. The engine weight and size are scaled according to reference 4 based upon the installed thrust. The specific fuel consumption obtained from ENGINE and the L/D from XLOD are substituted into the classical Brequet range relationship for each cruise segment to determine the fuel consumption.

$$\frac{W_1}{W_2} = \exp\left(\frac{c\ R}{V_{cr}\ L/D}\right) \qquad (10)$$

After WEIGHT has converged upon the aircraft operating weights for the desired mission, DOCOST continues with the cost estimates. AIRCOST used the weight, some production assumptions (number of prototypes, number of production, time for development, etc.) and the statistical relationships of reference 4 to predict the purchase cost of the airplane. Some cost increases based on references 17 and 18 are arbitrarily applied to account for the inclusion of active controls.

MAINCST uses statistical relationships found in references 19 and 20 to determine the cost of airplane maintenance. A number of configuration assumptions have to be made (e.g., number of APU's, windows and IMU's) to utilize these equations (see Appendix VII). The equations for estimating the other direct operating cost terms come from references 17 and 20. Indirect operating cost is predicted using the statistical relationships from references 17 and 21. An annual rate of return on investment (ROI) is calculated and the remaining performance indices are saved in the data base for future use by the optimizer.

EVALUATION OF CONSTRAINT FUNCTIONS

The program version included herein has 52 constraint functions that can be applied to the transport design. The designer chooses an upper and lower boundary for each function as an input. The program does a test on all constraint lower boundaries; and, if -999 is input for the lower boundary of a constraint function, the constraint is not included in the penalty function even if it is a violation. The constraint functions are of two general types, design or operational constraints and handling quality constraints. The first set restricts the design to avoid infeasible geometries or to insure satisfying performance regulations and mission requirements. The second set is used in the study of tail sizing and the impact of flying qualities design criteria upon transports with relaxed static stability augmentation systems.

CNSTRN returns the values of the constraint functions to SETUP, where they are identified as violated or not violated, normalized and assembled into a penalty function. The ratio of cruise thrust available to cruise thrust required is obtained from the data base as are the cruise altitudes and the cruise wing lift coefficient. The geometry constraints include insuring that the aft center-of-gravity is far enough forward of the main landing gear to provide sufficient nose wheel steering and that there is enough floor space to seat the passengers.

The missed approach climb gradient and the second segment climb gradient are engine-out performance requirements specified by the Federal Aviation Regulations, FAR's, (Ref. 22). The required thrust to weight ratio is calculated as follows:

$$\frac{T_I}{W} = \left(\frac{N}{N - 1} \right) \left(\frac{1}{L/D} + \sin \alpha \right) \left(\frac{1}{T/T_I} \right) \tag{11}$$

The flight path angle is specified by the FAR's and the L/D is obtained by calling XLOD with the proper speed and configuration specified. The second segment climb is performed at maximum gross weight and at a lift coefficient defined by

$$C_{L_2} = C_{L_{TO}}/1.44 \tag{12}$$

The missed approach climb is performed at maximum landing weight and at a lift coefficient defined by

$$C_{L_A} = C_{L_{so}}/1.69 \tag{13}$$

$C_{L_{TO}}$ and $C_{L_{so}}$ are determined by specifying the maximum lift coefficient that the wing can support in each flap configuration and then calling XLOD, which for this case trims the airplane maintaining the wing lift coefficient. Since the tail of conventional configurations is generally carrying a download at this point, the aircraft will usually trim at an overall lift coefficient less than the one specified for the wing alone.

The landing and take-off field length are determined using empirical relationships from reference 23. The landing field length utilizes approach speed as the independent parameter. TOP, which is defined as

$$TOP = \frac{W_{TO}/S_W}{C_{L_{TO}} T_I/W_{TO}} \tag{14}$$

is used as the independent parameter for the take-off analysis.

Several of the flying quality constraints are control power requirements. One is to maintain a lift coefficient on the tail greater than -.8 during approach (Ref. 24). This is to provide adequate margin from the maximum download capable of being supported by the tail (generally $C_{L_{t_{max}}} = -1.2$) to insure a capability to rotate and trim the aircraft for landing.

The tail is also required to be able to rotate the airplane for take-off. The maximum available download the tail can produce during take-off roll is calculated using the relationships in reference 14 modified for ground effect using the geometric angle-of-attack method of reference 8. The required download at the tail is determined from statics, such as the development in reference 25. The constraint specifically requires the ratio of the available tail download to the required tail download to be greater than 1.

The flying quality analysis is initiated by trimming the airplane in approach configuration with an altitude of 150 meters by calling XLOD. The nondimensional stability derivatives for cruise and approach, which are stored in the data base, are converted to dimensional stability derivatives by DIMDER. The characteristic equation for the fourth order longitudinal set of equations (Ref. 26) is assembled by LONGRT. The four roots are determined by using RPOLY, a system routine for finding roots of polynomials on Langley Research Center's FORTRAN Math Library.

The preceding analysis is used to assign the following constraint functions for both cruise and approach: static stability, maneuver stability, dynamic stability, phugoid mode frequency and damping and the short period mode frequency and damping. The dimensional stability derivatives are used to estimate the following parameters which have been suggested as useful for flying qualities analysis: time-to-double, time-to-half, flight path stability in approach, vertical gain and ω_{sp}^2/n_Z.

The tail is configured with a trimmable stabilizer, maintaining the elevator for maneuvering. If the stabilizer "hits" a control stop in either cruise or approach, the elevator is deflected to satisfy the remaining trim requirements. The amount of this trim deflection is stored as a constraint function and is usually required to be zero. Otherwise, a control deflection would indicate a loss of control authority, and in some cases, an increase in trim drag.

Since one intended use of the program is to study unaugmented flying qualities design criteria, it is desirable to insure that the airplane is capable of being practically augmented to excellent flying qualities. A pitch-attitude-hold with pitch-rate-command autopilot was chosen as a conservative estimate of an augmentation system. The airplane is arbitrarily augmented to have: $\dfrac{\omega_{sp}^2}{n_Z/\alpha} = 1$ and $\zeta_{sp} = .7$. An extension of reference 27 is used to calculate the feedback gains K_θ and $K_{\dot{\theta}}$. In reference 27 it is assumed that M_w, $M_{\dot{w}}$ and Z_δ are negligible and hence zero. If these assumptions are removed, the following relations are derived utilizing the short period approximation to the longitudinal dynamics:

$$K_\theta = \omega_{sp}^2 \frac{\left(1 - M_{\dot{w}} U_o\right)}{M_\delta + M_w} + K_{\dot{\theta}} \frac{\left(M_{\dot{w}} M_\delta\right)}{M_\delta + M_w} \omega_{sp}^2 \tag{15}$$

$$K_{\dot{\theta}} = 2\zeta_{sp}\omega_{sp}\left(M_{\dot{w}} U_o - 1\right) - M_q - M_w U_o + M_{\dot{w}}\omega_{sp}^2 \frac{\left(1 - M_{\dot{w}} U_o\right)}{\left(M_\delta + M_\delta\right)}$$

$$+ 1 - \frac{M_{\dot{w}}^2 M_\delta \omega_{sp}}{\left(M_\delta + M_w\right)} + 2\zeta_{sp}\omega_{sp} M_w M_\delta - M_\delta - M_w M_\delta \tag{16}$$

These gains are then substituted in equations B-31 and B-38 of reference 27 for estimating the variance of the elevator position and elevator position rate in cruise and approach. The turbulence is assumed to have a characteristic length of 760 meters (2500 feet) with an RMS gust level of .9 and 2.13 m/sec (3 and 7 ft/sec) in cruise and approach, respectively. These autopilot calculations are used to assign the following quantities in cruise and approach to available constraint functions: K_θ, $K_{\dot{\theta}}$, σ_δ and $\sigma_{\dot{\delta}}$. The constraint functions are used to insure that enough aerodynamic control exists to stabilize the airplane to excellent flying qualities and that enough hydraulic capability is available to prevent control surface rate saturation in heavy turbulence.

PROGRAM USE

A listing of the computer program set up to optimize seven design variables is included as Appendix I. Appendices II and III show sample input and output, respectively, for the program. Appendix IV contains a listing of a procedure file that will execute the program on the Langley Research Center computer system. As an aid in understanding the coding, a list of key program variables by routine and descriptions of their values are presented in Appendix V. Appendix VI is a compendium describing the variables in the common blocks.

The procedure file listed in Appendix IV contains a call to PPB, a program for executing a geometry preprocessor upon the output data placed on TAPE4 by subroutine XOUTPUT. This preprocessor puts on TAPE7 a data set suitable for executing ABS2290, an airplane graphics package described in reference 28. It is useful during conceptual design trade studies to see pictures of the configurations being generated. An example of this feature is shown in Figure 4.

Typically, with a case similar to the one contained in the appendices, approximately 500 function calls, or iterations, are required for a convergence of NELMIN. A function call averages about 1 second in execution time on the Langley Research Center Cyber 175.

If the user desired to add more design variables for the optimizer to iterate upon, these can be added as assignment statements beneath the transformation in SETUP (see Appendix I). Sample statements are left for adding cruise Mach number, wing sweep angle, wing thickness ratio and fuselage diameter as design variables. Usually all that is necessary to add an independent design variable is to equate it to a variable in the system of common blocks, which should contain degrees of freedom adequate for studies at the preliminary design level.

An array in the common block GEOM named PX has been included to aid in the study of certain changes representative of technological improvements. The specifics of its use are described in Appendix VI. For example, the following parameters could be studied during a design series: engine fuel efficiency, wing drag reduction, pitching moment reduction and structural efficiency.

REFERENCES

1. Olsson, D. M.: A Sequential Simplex Program for Solving Minimization Problems. Journal of Quality Technology, Vol. 6, No. 1, January 1974, pp. 53-57.

2. Olsson, D. M.; and Nelson, Lloyd S.: The Nelder-Mead Simplex Procedure for Function Minimization. Technometrics, Vol. 17, No. 1, February 1975, pp. 45-51.

3. Park, Stephen K.: A Transformation Method for Constrained Function Minimization. NASA TN D-7983, November 1975.

4. Nicolai, Leland: Fundamentals of Aircraft Design. University of Dayton Press, Dayton, Ohio, 1975.

5. Anderson, et al.: Development of Weight and Cost Estimates for Lifting Surfaces with Active Controls. NASA CR-144937, 1976.

6. Oman, G. H.: Vehicle Design Evaluation Program, NASA CR-145070, 1977.

7. Aggarwal, et al.: An Analysis of Fuel Conserving Operational Procedures and Design Modifications for Bomber/Transport Aircraft. AFFDL TR-78-96, Vol. II, 1978.

8. Ellison, D. E., et al.: USAF Stability and Control Handbook. Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, June 1969.

9. Roskam, Jan: Methods for Estimating Stability and Control Derivatives of Conventional Subsonic Airplanes. Roskam Engineering, Lawrence, Kansas, 1979.

10. Seckel, Edward: Stability and Control of Airplanes and Helicopters. Academic Press, New York City, New York, 1964.

11. Heffley, Robert K.; and Jewell, Wayne F.: Aircraft Handling Qualities Data. NASA CR-2144, 1972.

12. Roskam, Jan: Airplane Flight Dynamics. Roskam Engineering, Lawrence, Kansas, 1979.

13. Perkins, Courtland D.; and Hage, Robert : Airplane Performance, Stability and Control. John Wiley and Sons, New York City, New York, 1949.

14. Kyser, Albert C.: An Elementary Analysis of the Effect of Sweep, Mach Number and Lift Coefficient on Wing Structure Weight. NASA TM 74072, 1977.

15. Arbuckle, P. Douglas; Mitchell, Terry B.; Sliwa, Steven M.; and Tiffany, Sherwood H.: FIT: A Program for Interactively Fitting Data of up to Two Independent Variables with Least Squares Polynomials. NASA LaRC, Active Controls Project Office Memorandum, ACPO 80-1, September 1980.

16. McLaughlin, Milton D.: Calculations and Comparisons with an Ideal Minimum of Trimmed Drag for Conventional and Canard Configurations having Various Levels of Static Stability. NASA TN D-8391, May 1977.

17. Anonymous: Assessment of the Application of Advanced Technologies to Subsonic CTOL Transport Aircraft. NASA CR-112242, 1973.

18. Sizlo, T. R.; Berg, R. A.; and Gilles, D. L.: Development of a Low-Risk Augmentation System for an Energy Efficient Transport having Relaxed Static Stability. NASA CR 159166, December, 1979.

19. Madolon, Dal V.: Estimating Airline Operating Costs. NASA TM 78694, 1978.

20. Anonymous: A New Method for Estimating Current and Future Transport Aircraft Operating Economics. NASA CR-144937, 1976.

21. Stossel, Robert F.: A Proposed Standard Method for Estimating Airplane Indirect Operating Cost. Lockheed-Georgia Company, Report No. LW70-500R, 1970.

22. Anonymous: Airworthiness Standards--Transport Category Airplanes, Federal Aviation Regulations, Part 25. Department of Transportation, Federal Aviation Agency, 1974.

23. Loftin, Lawrence K., Jr.: Subsonic Aircraft, Evolution and the Matching of Size to Performance. NASA Reference Publication 1060, 1980.

24. Anonymous: Stability and Control Design Criteria. The Boeing Company, D6-6800-1, 1972.

25. Chalk, C. R., et al.: Background Information and User Guide for MIL-F-8785B (ASG), "Military Specification-Flying Qualities for Piloted Airplanes." AFFDL-TR-69-72, August 1969.

26. McRuer, et al.: Aircraft Dynamics and Automatic Control. Princeton University Press, Princeton, New Jersey, 1973.

27. Hoffman, Geggor L., et al.: Vehicle Design Considerations for Active Control Application to Subsonic Transport Aircraft. NASA CR-2408, 1974.

28. Craidon, Charlotte B.: Description of a Digital Computer Program for Airplane Configuration Plots. NASA TM X-2074, September 1970.

APPENDIX I - PROGRAM LISTING

```
      PROGRAM SIMPACT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE4)
**************************************
*     OPTIMIZATION COMMON BLOCKS     *
**************************************
      COMMON /AVOID/FACT(59),GNORM(59)
      COMMON /CONSTR/SU(59),SL(59),XINEQ(59)
      COMMON /DEBUG/IDEBUG,IDEBUG2
      COMMON /FCOUNT/ICNT,JCNT,KCNT,NDAV,LCNT,IWT,NOIT
      COMMON /LABELP/ARUN(8)
      COMMON /PERF/UNAUG,SCF,NVAR,MINEQ
      COMMON /STRAIN/ CON(59)
      COMMON /VARIAB/AMP(15),AVE(15)
      COMMON /VIOL/CAYY,MC,ILINE,IOUT
********************************
*     DESIGN COMMON BLOCKS     *
********************************
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /DRAG/CDS(6),CDSAP(6)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      COMMON /GRAVITY/CG(6)
      COMMON /STAB/DERIVCR(15),DERIVAP(15),STOR(20)
      COMMON /WTSVE/WTS(20)
**********************
*     INPUT DATA     *
**********************
      REAL STEP(15),XMIN(15),XSEC(15)
      REAL XBAR(15),XBARO(15)
      REAL XL(15),XU(15)
      READ(5,18) ARUN
   18 FORMAT(8A10)
      READ(5,*) NVAR,MINEQ
      READ(5,*) (XBARO(I),I=1,NVAR)
      IDEBUG=0
      IWT=0
      IDEBUG2=0
      DO 19 J=1,NVAR
      READ(5,*) JI,XL(J),XU(J)
      AVE(J)=(XL(J)+XU(J))/2.0
      AMP(J)=(XU(J)-XL(J))/2.0
   19 CONTINUE
      DO 20 I=1,MINEQ
   20 READ(5,*) IJ,SL(I),SU(I)
      WRITE(6,927) ARUN
  927 FORMAT("1"/5X*RUN NO=*8A10/)
      WRITE(6,718) NVAR,MINEQ
  718 FORMAT(/10X*NO. OF VARIABLES=* I5/10X
     $*NO. OF CONSTRAINTS=*I5)
      WRITE(6,720) (XBARO(L),L=1,NVAR)
```

```
  720 FORMAT(10X*VARIABLES=*4(T30,5F10.4/))
      DO 24 J=1,NVAR
      WRITE(6,22) J, XL(J),XU(J),AVE(J),AMP(J)
   22 FORMAT(5X*J,XL,XU,AVE,AMP=*I5,4F15.4)
   24 CONTINUE
      DO 26 I=1,MINEQ
      WRITE(6,928) I,SL(I),SU(I)
  928 FORMAT(5X*I,SL(I),SU(I)=*I5,2F10.2)
      FACT(I)=1.
      XINEQ(I)=1.
      IF(SL(I).EQ.-999.) FACT(I)=0.
      GNORM(I)=(ABS(SU(I))+ABS(SL(I)))/2.0
   26 CONTINUE
      DO 27 I=1,NVAR
   27 XBAR(I)=XBARO(I)
      READ(5,*) NONEL
      WRITE(6,39)  NONEL
   39 FORMAT(10X*NUMBER OF REQUESTED NELMINS=*I5)
      READ(5,*) SCF,REQMIN,CAYY,STEP1,ILINE
      CALL XINPUT
      IOUT=0
      ILINER=ILINE
      ILINE=0
      JCNT=0
******************************
*     START OPTIMIZATION     *
******************************
      DO 1020 I=1,NONEL
      DO 1000 K=1,NVAR
 1000 STEP(K)=STEP1/I
      CAYY=CAYY*10.
      REQMIN=REQMIN/10
      KCNT=0
      WRITE(6,904)  (XBAR(L),L=1,NVAR)
  904 FORMAT(/10X*INITAIL XBAR*5(T35,5F12.4/))
      XSIGN=1.
      DO 915 IJ=1,NVAR
      IF(ABS(XBAR(IJ)).LT.0.8) GO TO 915
      XO=XBAR(IJ)
      XSIGN=SIGN(XSIGN,XBAR(IJ))
      DXB=ABS(XBAR(IJ))-1.
      IF(DXB.GE.1..OR.DXB.EQ.0.)XBAR(IJ)=XBAR(IJ)/(ABS(XBAR(IJ))+.25)
      IF(DXB.GT.0.0.AND.DXB.LT.1.0) XBAR(IJ)=XBAR(IJ)-XSIGN*DXB*2.
      STEP(IJ)=SIGN(STEP(IJ),-XBAR(IJ))
      WRITE(6,911) IJ,XO,XBAR(IJ),STEP(IJ),XSIGN
  911 FORMAT(3X"--- RESET (VAR,XO,XBAR,STEP,XSIGN)="I5,4F12.4)
  915 CONTINUE
      WRITE(6,906)  (STEP(L),L=1,NVAR)
  906 FORMAT(10X*INITAIL STEPS*5(T35,5F12.4/))
      WRITE(6,908) REQMIN,SCF,CAYY
  908 FORMAT(10X*REQMIN,SCF,CAYY*T35,3E12.4)
****************************************
*      START NELDER-MEAD SUBROUTINE    *
```

```
****************************************
      ICOUNT=1500
      CALL NELMIN(NVAR,XBAR,XMIN,XSEC,YNEWLO,YSEC,REQMIN,STEP,
     $ ICOUNT)
      WRITE(6,806)
  806 FORMAT(/5X*/ / / NELMIN COMPLETE/ / / */)
      WRITE(6,810)  (XMIN(L),L=1,NVAR)
  810 FORMAT(10X*XMIN=*3(T35,5F15.8/))
      WRITE(6,812) YNEWLO
  812 FORMAT(10X*YNEWLO=*T35,E15.6)
      WRITE(6,814) ICOUNT,JCNT
  814 FORMAT(10X*ICOUNT=*T35,I5/10X*TOT. FUNCTION CALLS=*T35,I5/)
      XIWT=IWT
      XJCNT=JCNT
      RATWT=XIWT/XJCNT
      WRITE(6,830) XIWT,RATWT
  830 FORMAT(/10X,22HTOT. WEIGHT ITERATIONS,T35,F12.2/
     $ 10X,28HAVE. WT. ITERATIONS PER CALL,T35,F12.3)
      DO 1010 K=1,NVAR
 1010 XBAR(K)=XMIN(K)*1.0
      IOUT=1
      ILINE=0
      IOUT=0
      ILINE=ILINER
***********************************************************************
*     THE FOLLOWING STATEMENTS CAN BE USED TO FIND THE GRADIENTS      *
*     AT THE OPTIMAL SOLUTION POINT:                                  *
*                                                                     *
*               DX=1.0E-5                                             *
*               DO 610 JI=1,NVAR                                      *
*               IDX=0                                                 *
*           603 DO 605 J=1,NVAR                                       *
*               XBARO(J)=XMIN(J)                                      *
*               IF(J.EQ.JI) XBARO(J)=XMIN(J)+DX                       *
*           605 CONTINUE                                              *
*               VAL=FN(XBARO)                                         *
*               FX(JI)=(VAL-ORIG)/DX                                  *
*               IF(FX(JI).LT.1.0E4) GO TO 610                         *
*               IF(IDX.GT.0) GO TO 610                                *
*               IDX=1                                                 *
*               DX=-DX                                                *
*               GO TO 603                                             *
*           610 CONTINUE                                              *
*               DO 620 JJI=1,NVAR                                     *
*               WRITE(6,607) JJI,FX(JJI)                              *
*           607 FORMAT(10X*DERIVATIVE WITH RESPECT TO VARIABLE NO.*   *
*              $ I5,10X,E15.5)                                        *
*           620 CONTINUE                                              *
***********************************************************************
 1020 CONTINUE
      CALL XOUTPUT(4)
      WRITE(6,810)  (XMIN(L),L=1,NVAR)
      STOP
```

```
      END
      SUBROUTINE NELMIN(N,START,XMIN,XSEC,YNEWLO,
     1YSEC,REQMIN,STEP,ICOUNT)
      REAL START(N),STEP(N),XMIN(N),
     1XSEC(N),YNEWLO,YSEC,REQMIN,P(20,21),PSTAR(20),
     2P2STAR(20),PBAR(20),Y(20),DN,Z,YLO,RCOEFF,
     3YSTAR,ECOEFF,Y2STAR,CCOEFF,FN,DABIT,DCHK,
     4COORD1,COORD2
      DATA RCOEFF/1.0/,ECOEFF/2.0/,CCOEFF/0.5/
      DATA PSTAR,P2STAR,PBAR /60*0./
      KCOUNT=ICOUNT
      ICOUNT=0
      IF( REQMIN .LE. 0.) ICOUNT=ICOUNT-1
      IF(N .LE. 0) ICOUNT=ICOUNT-10
      IF(N .GT. 20  ) ICOUNT=ICOUNT-10
      IF(ICOUNT .LT. 0) RETURN
      DABIT=2.04607E-35
      BIGNUM=1.0E38
      KONVGE=5
      XN=FLOAT(N)
      DN=DBLE(XN)
      NN=N+1
*********************************************
*     CONSTRUCTION  OF  INITIAL  SIMPLEX    *
*********************************************
      DO 1 I=1,N
    1 P(I,NN)=START(I)
      Y(NN)=FN(START)
      ICOUNT=ICOUNT+1
      DO 2 J=1,N
      DCHK=START(J)
      START(J)=DCHK+STEP(J)
      DO 3 I=1,N
    3 P(I,J)=START(I)
      Y(J)=FN(START)
      ICOUNT=ICOUNT+1
    2 START(J) =DCHK
***********************************************************************
*        SIMPLEX CONSTRUCTION COMPLETE                                *
*                                                                     *
*        FIND HIGHEST AND LOWEST Y VALUES                             *
*                                                                     *
*     YNEWLO (Y(IHI)) INDICATES THE VERTEX OF THE SIMPLEX TO BE       *
*     REPLACED.                                                       *
***********************************************************************
 1000 YLO=Y(1)
      YNEWLO=YLO
      ILO=1
      IHI=1
      DO 5 I=2,NN
      IF(Y(I) .GE. YLO) GO TO 4
      YLO=Y(I)
      ILO=I
```

```
    4 IF(Y(I) .LE. YNEWLO) GO TO 5
      YNEWLO=Y(I)
      IHI=I
    5 CONTINUE
**************************************************
*     PERFORM CONVERGENCE CHECKS ON FUNCTION     *
**************************************************
      DCHK=(YNEWLO+DABIT)/(YLO+DABIT)-1.
      IF(ABS(DCHK) .LT. REQMIN) GO TO 900
      KONVGE=KONVGE-1
      IF(KONVGE .NE. 0) GO TO 2020
      KONVGE=5
******************************************************************
*     CHECK CONVERGENCE OF COORDINATES ONLY EVERY 5 SIMPLEXES    *
******************************************************************
      DO 2015 I=1,N
      COORD1=P(I,1)
      COORD2=COORD1
      DO 2010 J=2,NN
      IF(P(I,J) .GE. COORD1) GO TO 2005
      COORD1=P(I,J)
 2005 IF(P(I,J) .LE. COORD2) GO TO 2010
      COORD2=P(I,J)
 2010 CONTINUE
      DCHK=(COORD2+DABIT)/(COORD1+DABIT)-1.
      IF(ABS(DCHK) .GT. REQMIN) GO TO 2020
 2015 CONTINUE
      GO TO 900
 2020 IF(ICOUNT .GE. KCOUNT) GO TO 900
**********************************************************************
*     CALCULATE PBAR, THE CENTROID OF THE SIMPLEX VERTICES EXCEPTING *
*     THAT WITH Y VALUE YNEWLO.                                      *
**********************************************************************
      DO 7 I=1,N
      Z=0.0
      DO 6 J=1,NN
    6 Z=Z+P(I,J)
      Z=Z-P(I,IHI)
    7 PBAR(I)=Z/DN
**************************************************
*        REFLECTION THROUGH THE CENTROID         *
**************************************************
      DO 8 I=1,N
    8 PSTAR(I)=(1.0+RCOEFF)*PBAR(I)-RCOEFF*P(I,IHI)
      YSTAR=FN(PSTAR)
      ICOUNT=ICOUNT+1
      IF(YSTAR .GE. YLO) GO TO 12
      IF(ICOUNT .GE. KCOUNT) GO TO 19
**************************************************
*        SUCCESSFUL REFLECTION, SO EXTENSION     *
**************************************************
      DO 9 I=1,N
    9 P2STAR(I)=ECOEFF*PSTAR(I)+(1.0-ECOEFF)*PBAR(I)
```

```
      Y2STAR=FN(P2STAR)
      ICOUNT=ICOUNT+1
*************************************************
*        RETAIN EXTENSION OR CONTRACTION        *
*************************************************
      IF(Y2STAR .GE. YSTAR) GO TO 19
   10 DO 11 I=1,N
   11 P(I,IHI)=P2STAR(I)
      Y(IHI)=Y2STAR
      GO TO 1000
*****************************
*        NO EXTENSION       *
*****************************
   12 L=0
      DO 13 I=1,NN
      IF(Y(I) .GT. YSTAR) L=L+1
   13 CONTINUE
      IF(L .GT. 1) GO TO 19
      IF(L .EQ. 0) GO TO 15
***************************************************************
*     CONTRACTION ON THE REFLECTION SIDE OF THE CENTROID      *
***************************************************************
      DO 14 I=1,N
   14 P(I,IHI)=PSTAR(I)
      Y(IHI)=YSTAR
***************************************************************
*        CONTRACTION ON THE Y(IHI) SIDE OF THE CENTROID       *
***************************************************************
   15 IF(ICOUNT .GE. KCOUNT) GO TO 900
      DO 16 I=1,N
   16 P2STAR(I)=CCOEFF*P(I,IHI)+(1.0-CCOEFF)*PBAR(I)
      Y2STAR=FN(P2STAR)
      ICOUNT=ICOUNT+1
      IF(Y2STAR .LT. Y(IHI)) GO TO 10
*******************************************
*        CONTRACT THE WHOLE SIMPLEX       *
*******************************************
      DO 18 J=1,NN
      DO 17 I=1,N
      P(I,J)=(P(I,J)+P(I,ILO))*0.5
   17 XMIN(I)=P(I,J)
      Y(J)=FN(XMIN)
   18 CONTINUE
      ICOUNT=ICOUNT+NN
      IF(ICOUNT .LT. KCOUNT) GO TO 1000
      GO TO 900
******************************
*        RETAIN REFLECTION   *
******************************
   19 CONTINUE
      DO 20 I=1,N
   20 P(I,IHI)=PSTAR(I)
      Y(IHI)=YSTAR
```

```
      GO TO 1000
**********************************************************************
*     SELECT THE TWO BEST FUNCTION VALUES (YNEWLO AND YSEC) AND THEIR   *
*     COORDS.(XMIN AND XSEC).                                           *
**********************************************************************
  900 DO 23 J=1,NN
      DO 22 I=1,N
   22 XMIN(I)=P(I,J)
      Y(J)=FN(XMIN)
   23 CONTINUE
      YNEWLO=BIGNUM
      DO 24 J=1,NN
      IF(Y(J) .GE. YNEWLO) GO TO 24
      YNEWLO=Y(J)
      IBEST=J
   24 CONTINUE
      Y(IBEST)=BIGNUM
      YSEC=BIGNUM
      DO 25 J=1,NN
      IF(Y(J) .GE. YSEC) GO TO 25
      YSEC=Y(J)
      ISEC=J
   25 CONTINUE
      DO 26 I=1,N
      XMIN(I)=P(I,IBEST)
      XSEC(I)=P(I,ISEC)
   26 CONTINUE
      RETURN
      END
      FUNCTION FN(XBAR)
      COMMON /VIOL/CAYY,MC,ILINE,IOUT
      COMMON /FCOUNT/ICNT,JCNT,KCNT,NDAV,LCNT,IWT,NOIT
      REAL XBAR(15)
      CALL SETUP(XBAR,IOUT,OBJ)
      FN=OBJ
      IF(ILINE.GT.0) WRITE(6,50) JCNT,FN,MC,IWT,NOIT
   50 FORMAT(10X,"CALL NUMBER=",I5,5X"OBJ="E15.8,10X
     $ ,"NO. VIOLATIONS="I5,5X
     $ ,"IWT=",I5,5X,"NOIT="I5)
      RETURN
      END
      SUBROUTINE SETUP(GAINS,IPR,OBJ)
      INTEGER OUTPUT
      REAL GAINS(15),GBAR(59)
      COMMON /STRAIN/CON(59)
      COMMON /VARIAB/AMP(15),AVE(15)
      COMMON /VIOL/CAYY,MC,ILINE,IOUT
      COMMON /CONSTR/SU(59),SL(59),XINEQ(59)
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      COMMON /PERF/UNAUG,SCF,NVAR,MINEQ
      COMMON /AVOID/FACT(59),GNORM(59)
      DO 22 I=1,NVAR
```

```
      DESIGN(I)=AMP(I)*SIN(1.5707963*GAINS(I))+AVE(I)
   22 CONTINUE
************************************************************************
*     TO DEFINE ADDITIONAL DESIGN VARIABLES, INSERT DEFINITION CARDS   *
*     HERE.  SAMPLES ARE GIVEN BELOW:                                  *
*                                                                      *
*     GX(3)=DESIGN(8)      --DESIGN(8) IS MACH NUMBER                  *
*     W(1)=DESIGN(9)       --DESIGN(9) IS WING SWEEP                   *
*     W(4)=DESIGN(10)      --DESIGN(10) IS WING THICKNESS RATIO        *
*     W(3)=DESIGN(11)      --DESIGN(11) IS WING TAPER RATIO            *
*     GX(5)=DESIGN(12)     --DESIGN(12) IS FUSELAGE DIAMETER           *
************************************************************************
*     ALSO, VARIOUS DESIGN CONSTRAINTS CAN BE ADDED.  FOR EXAMPLE,     *
*     ADDING THE STATEMENT:                                            *
*                                                                      *
*     GX(32)=W(1)                                                      *
*                 RESTRICTS THE HORIZONTAL TAIL SWEEP ANGLE TO BE      *
*     EQUAL TO THE WING SWEEP ANGLE.                                   *
************************************************************************
      IF(PX(5).GT.0) DESIGN(6)=GX(21)*ITERM(4)
      IF(IPR.GT.0) WRITE(6,441) DESIGN
  441 FORMAT(//15X*SET-UP*/20X*DESIGN=*5(T40,5F15.4/))
      OUTPUT=0
      CALL DOCOST(COST,0)
      CALL CNSTRN(0)
      UNAUG=COST*SCF
      IF(IPR.GT.0) WRITE(6,442) UNAUG
  442 FORMAT(20X*CALL TO DOC/CNSTRN COMPLETE---DOC= $*E15.5)
*********************************************
*     THIS SECTION CALCULATES GBAR ARRAY    *
*     AND THE PENALTY FOR VIOLATIONS        *
*********************************************
      MC=0
      PENT=0.
      DO 150 I=1,MINEQ
      T=CON(I)
      XINEQ(I)=0.
      GBAR(I)=AMAX1(T-SU(I),SL(I)-T)
      GBAR(I)=GBAR(I)*FACT(I)/GNORM(I)
      IF(GBAR(I).LE.0.0) GO TO 150
      XINEQ(I)=1.
      MC=MC+1
      PENT=PENT+CAYY*GBAR(I)*GBAR(I)
  150 CONTINUE
      IF(IPR.GT.0) WRITE(6,160) PENT
  160 FORMAT(20X*PENALTY TERM=*E15.5)
*******************************************
*     AUGMENTED FUNCTION IS CREATED       *
*******************************************
      OBJ=UNAUG+PENT
      RETURN
      END
      SUBROUTINE XOUTPUT(IPRNT)
```

```
      COMMON /CONSTR/SU(59),SL(59),XINEQ(59)
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /DRAG/CDS(6),CDSAP(6)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      COMMON /GRAVITY/CG(6)
      COMMON /LABELP/ARUN(8)
      COMMON /STAB/DERIVCR(15),DERIVAP(15),STOR(20)
      COMMON /WTSVE/WTS(20)
***********************************************************************
*     PRINT OUT PARTS OF THE FUNCTION EVALUATION                      *
*                                                                     *
*     0=NONE,1=DOC,2=CNSTRN,3=DOC & CNSTRN,4=DOC & CNSTRN & DUMP      *
*     5=DOC(2) & CNSTRN(2),6=COMMON DUMP                                *
***********************************************************************
      IDUMP=0
      IF(IPRNT.LT.1) GO TO 999
      IF(IPRNT.EQ.1) GO TO 10
      IF(IPRNT.EQ.2) GO TO 20
      IF(IPRNT.EQ.3) GO TO 30
      IF(IPRNT.EQ.4) GO TO 40
      IF(IPRNT.EQ.5) GO TO 50
      IF(IPRNT.EQ.6) GO TO 70
   10 IDOC=1
      ICRN=0
      GO TO 80
   20 ICRN=1
      IDOC=0
      GO TO 80
   30 IDOC=1
      ICRN=1
      GO TO 80
   40 IDOC=1
      ICRN=1
      IDUMP=1
      GO TO 80
   50 IDOC=2
      ICRN=2
      IDUMP=0
   80 CALL DOCOST(TERM,IDOC)
      CALL CNSTRN(ICRN)
      IF(IDUMP.LT.1) GO TO 999
   70 WRITE(6,1010)
 1010 FORMAT(*1*30X*COMMON DUMP*///)
      WRITE(6,1012) DESIGN
 1012 FORMAT(/15X*DESIGN=*3(T30,5F15.5/))
      WRITE(6,1014) ITERM
 1014 FORMAT(/15X*ITERM=*3(T30,5I15/))
      WRITE(6,1016) PX
 1016 FORMAT(/15X"PX=",3(T30,5F15.4/))
      WRITE(6,1015) CDS
 1015 FORMAT(15X*CDS=*T30,6F15.9/)
      WRITE(6,1020) CDSAP
 1020 FORMAT(/15X*CDSAP=*T30,6F15.9/)
```

```
      WRITE(6,1025) W
 1025 FORMAT(/15X*W=*5(T30,5F15.9/))
      WRITE(6,1030) HX
 1030 FORMAT(/15X*HX=*5(T30,5F15.9/))
      WRITE(6,1035) GX
 1035 FORMAT(/15X*GX=*8(T30,5F15.9/))
      WRITE(6,1040) CG
 1040 FORMAT(/15X*CG=*T30,6F15.5)
      WRITE(6,1045) DERIVCR
 1045 FORMAT(/15X*DERIVCR=*5(T30,5F15.9/))
      WRITE(6,1050) DERIVAP
 1050 FORMAT(/15X*DERIVAP=*5(T30,5F15.9/))
      WRITE(6,1055) STOR
 1055 FORMAT(/15X*STOR=*5(T30,5F15.9/))
      WRITE(6,1060) WTS
 1060 FORMAT(/15X*WTS=*5(T30,5F15.3/))
      WRITE(6,1070) CST
 1070 FORMAT(/15X"CST="5(T30,5F15.3/))
      WRITE(4,1200) ARUN
 1200 FORMAT(8A10)
*******************************************************************
*     WRITE AN OUTPUT TAPE FOR THE PLOTTING PREPROCESSOR          *
*******************************************************************
      WRITE(4,*) DESIGN
      WRITE(4,*) ITERM
      WRITE(4,*) W
      WRITE(4,*) HX
      WRITE(4,*) GX
  999 RETURN
      END
      SUBROUTINE XINPUT
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      COMMON /GRAVITY/CG(6)
      COMMON /WTSVE/WTS(20)
************************************
*     INPUT DESIGN CONSTANTS       *
************************************
      DATA GX/35*0./
      READ(5,*) WTS(1)
      READ(5,*)  (PX(J),J=1,8)
      READ(5,*)  (ITERM(I),I=1,10)
      READ(5,*) WTS(16),CG(5)
      READ(5,*) W(1),W(2),W(3),W(4),W(5)
      READ(5,*) W(6),W(7),W(8),W(14),W(16)
      READ(5,*) W(17),W(18),W(19),W(20)
      READ(5,*) HX(1),HX(2),HX(3),HX(10)
      READ(5,*) HX(16),HX(17),HX(18),HX(19),HX(20)
      READ(5,*) GX(3),GX(4),GX(5),GX(6),GX(7)
      READ(5,*) GX(8),GX(11),GX(12),GX(17),GX(18)
      READ(5,*) GX(19),GX(20),GX(21),GX(22)
      READ(5,*) GX(23),GX(24),GX(25),GX(26),GX(27)
      READ(5,*) GX(32),GX(34),GX(16)
```

```
**************************
*     WRITE OUT DATA     *
**************************
      WRITE(6,100)
  100 FORMAT(*1*///30X*/ / / FUNCTION INPUT/ / / *//)
      WRITE(6,110) WTS(1)
  110 FORMAT(10X*WTS(1)*T40,F15.4)
      WRITE(6,120) ITERM
  120 FORMAT(10X*ITERM*T40,10I6)
      WRITE(6,125) PX
  125 FORMAT(10X,"PX=",3(T40,5F15.4/))
      WRITE(6,130) WTS(16)
  130 FORMAT(10X*WTS(16)*T40,F15.4)
      WRITE(6,140) W(1),W(2),W(3),W(4),W(5)
  140 FORMAT(10X*W...1-5=*T40,5F15.4)
      WRITE(6,150) W(6),W(7),W(8),W(14),W(16)
  150 FORMAT(10X*W...6-8,14,16=*T40,5F15.4)
      WRITE(6,160) W(17),W(18),W(19),W(20)
  160 FORMAT(10X*W...17-20=*T40,4F15.4)
      WRITE(6,170) HX(1),HX(2),HX(3),HX(10)
  170 FORMAT(10X*HX...1-3,10=*T40,4F15.4)
      WRITE(6,175) HX(16),HX(17),HX(18),HX(19),HX(20)
  175 FORMAT(10X*HX...16-20-*T40,5F15.4)
      WRITE(6,180) GX(3),GX(4),GX(5),GX(6),GX(7)
  180 FORMAT(10X*GX...3-7=*T40,5F15.4)
      WRITE(6,190) GX(8),GX(11),GX(12),GX(17),GX(18)
  190 FORMAT(10X*GX...8,11-12,17,18=*T40,5F15.4)
      WRITE(6,200) GX(19),GX(20),GX(21),GX(22)
  200 FORMAT(10X*GX...19-22=*T40,4F15.4)
      WRITE(6,210) GX(23),GX(24),GX(25),GX(26),GX(27)
  210 FORMAT(10X*GX...23-27=*T40,5F15.4)
      WRITE(6,220) GX(32),GX(34)
  220 FORMAT(10X*GX...32,34=*T40,2F15.4)
      RETURN
      END
      SUBROUTINE DOCOST(UNAUG,OUTPUT)
      INTEGER OUTPUT
      REAL INSUR,PER(12),XIOC(12),YIOC(12),YCOST(9)
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /FCOUNT/ICNT,JCNT,KCNT,NDAV,LCNT,IWT,NOIT
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      COMMON /WTSVE/WTS(20)
*******************************************
*     INITIALIZE GEOMETRY AND C.G.        *
*******************************************
      CALL GEO
      CALL CGCAL
************************************************
*     INITIALIZE AND INCREMENT COUNTERS        *
************************************************
      DATA JCNT/0/
      JCNT=JCNT+1
      KCNT=KCNT+1
```

```
      IF(OUTPUT.GE.1) GO TO 10
*************************************************
*     COSTS PER BLOCK HOUR OF DESIGN FLIGHT     *
*************************************************
   20 IOUT=OUTPUT
      IF(OUTPUT.GT.1) IOUT=0
      CALL WEIGHT(IOUT)
      CALL AIRCOST(PRICE,IOUT)
      YRMULT=1.07**(ITERM(6)-1976)
      PRICE=YRMULT*PRICE
      FL=WTS(18)
      BLKHR=WTS(19)
***********************************
*     DIRECT OPERATING COSTS      *
***********************************
      DEPRE=0.88*PRICE/(14.0*GX(26))
      SUPPORT=0.12*PRICE/(14.0*GX(26))
      SPARES=0.06*PRICE/(14.0*GX(26))
      DELAY=YRMULT*8.40
      INSUR=0.01*PRICE/(GX(26))
      FCOST=WTS(20)*WTS(16)/6.4
      FCOST=FCOST/BLKHR
      CALL MAINCST(XMCOST,IOUT)
      WTL=WTS(5)*0.453592
      FEELAND=YRMULT*1.54*WTL/1000.0
      FEELAND=FEELAND/BLKHR
      ATT=YRMULT*GX(19)*(0.691*FL+0.00175*FL*FL)
      ATT=ATT/BLKHR
      CREW=YRMULT*174*FL+43.5+(0.452*FL+.11299)*(WTS(1)*.453592/1000.)
      CREW=CREW/BLKHR
      SERVICE=YRMULT*63.0
      CONTROL=YRMULT*82.58/BLKHR
      DOC=DEPRE+SUPPORT+SPARES+DELAY+INSUR+FCOST+FEELAND+SERVICE
     $+ATT+CREW+XMCOST+CONTROL
***************************
*     PERCENT OF TOTAL    *
***************************
      PER(1)=DEPRE/DOC
      PER(2)=SUPPORT/DOC
      PER(3)=SPARES/DOC
      PER(4)=DELAY/DOC
      PER(5)=INSUR/DOC
      PER(6)=FCOST/DOC
      PER(7)=XMCOST/DOC
      PER(8)=FEELAND/DOC
      PER(9)=CREW/DOC
      PER(10)=ATT/DOC
      PER(11)=SERVICE/DOC
      PER(12)=CONTROL/DOC
      DO 30 J=1,12
   30 PER(J)=PER(J)*100.
      TOT=100.
******************************************************************
```

```
*      INDIRECT OPERATING COSTS PER BLOCK HOUR OF DESIGN FLIGHT     *
********************************************************************
      DATA YIOC/10HMAIN BURDN, 9HFOOD COST,5HMOVIE,8HPASS INS,
     $ 9HMISC PASS,9HADVERTISE,10HCOMMISSION,5HRESER,
     $ 9HPASS HDLG,8HBAG HDLG,10HCARGO HDLG,9HSERVICING/
      XIOC(1)=XMCOST*1.05
      IFIRS=.15*GX(19)*GX(24)
      IECON=GX(19)*GX(24)-IFIRS
      XIOC(2)=(IFIRS*2.42+IECON*1.05)
      XIOC(3)=196./BLKHR
      RPM=GX(19)*GX(24)*GX(4)/1000.
      XIOC(4)=0.52*RPM/BLKHR
      XIOC(5)=GX(19)*.18/BLKHR
      REVYR=GX(25)*GX(19)*GX(24)*GX(26)*GX(4)/BLKHR
      REVHR=REVYR/GX(26)
      XIOC(6)=.023*REVHR
      XIOC(7)=2.35*RPM/BLKHR
      PASSPHR=GX(19)*GX(24)/BLKHR
      XIOC(8)=4.40*PASSPHR
      XIOC(9)=2.87*PASSPHR
      XIOC(10)=1.31*PASSPHR
      TONCAR=GX(22)/2000.
      XIOC(11)=131.08*TONCAR/BLKHR
      XIOC(12)=(0.03*9.5+0.0025)*GX(19)/BLKHR
      TOTIOC=0.
      XIOC(1)=XIOC(1)/YRMULT
      XIOC(6)=XIOC(6)/YRMULT
      DO 200 I=1,12
      XIOC(I)=XIOC(I)*YRMULT
      TOTIOC=TOTIOC+XIOC(I)
  200 CONTINUE
**********************************************
*     RETURN ON INVESTMENT CALCULATIONS      *
**********************************************
      XINVEST=0.9*PRICE
      TAXRT=0.48
      COSTHR=DOC+TOTIOC
      PROFIT=(REVHR-COSTHR)*GX(26)
      ROI=(1.-TAXRT)*PROFIT/XINVEST
      FARROI=(.26*PRICE+COSTHR*GX(26))*(BLKHR/(GX(19)*GX(24)*GX(26)
     $ *GX(4)))
*************************************
*     ASSIGN PERFORMANCE INDEX      *
*************************************
      CST(1)=DOC
      CST(2)=DOC*BLKHR
      CST(3)=ROI
      CST(4)=FARROI
      CST(7)=WTS(1)
      INCPH=COSTHR+.15*XINVEST/(GX(26)*(1.-TAXRT))
      INCPF=INCPH*BLKHR
      CST(8)=INCPF
      CST(9)=PRICE
```

```
      UNAUG=CST(ITERM(5))
      GO TO 40
****************************
*     OUTPUT SECTION       *
****************************
   10 WRITE(6,42)
      WRITE(6,44) (DESIGN(JK),JK=1,12)
      IF(OUTPUT.GT.1) GO TO 20
      WRITE(6,705)
  705 FORMAT(//5X*INPUT CONSTANTS*/)
      WRITE(6,710) (W(J),J=1,8)
  710 FORMAT(10X*WING...SWEEP,INCIDENCE,TAPER RATIO:*T55,3F12.4/
     $ 10X*THICKNESS,TWIST,E1,E2,DESIGN CL:*T55,5F12.4)
      WRITE(6,720) W(14),W(19),W(17),W(18),W(16),GX(17),GX(18)
  720 FORMAT(10X*CM(CR,APP)*T55,2F12.4/
     $ 10X*DELTA CM(10,25 DEGREES FLAP):*T55,2F12.4/
     $ 10X*ANGLE OF ZERO LIFT(0,10,45 DEGREES FLAP):*T55,3F12.4)
      WRITE(6,730) W(20),GX(11),GX(5),GX(8),GX(12)
  730 FORMAT(10X*DELTA CD (10-45 DEGREES FLAP):*T55,F12.4/
     $ 10X*TURBULENCE LENGTH/ROOT 3,FUSE. DIA.:*T55,2F12.4/
     $10X*CL-MAX(TO),CL-MAX(L):*T55,2F12.4)
      WRITE(6,740) GX(3),GX(4),GX(19),GX(22),GX(23),GX(27)
  740 FORMAT(10X*MISSION...MACH NO.,RANGE,NO. PASS:*T55,3F12.4/
     $ 10X*CARGO WEIGHT:*T55,F12.4/
     $ 10X*DELTA CG, WTL(MAX)/WTO:*T55,3F12.4)
      WRITE(6,750) GX(6),GX(7),GX(20),GX(21)
  750 FORMAT(10X*ENGINE...L,W,WT,TREF:*T55,4F12.4)
      WRITE(6,760) HX(1),HX(2),HX(3),HX(10),GX(24),GX(25),GX(26)
  760 FORMAT(10X*TAIL...TAPER RATIO,THICKNESS,ELE EFF:*T55,3F12.4/
     $ 10X*ELEVATOR TIME CONSTANT:*T55,F12.4/
     $ 10X*ECONOMICS...LOAD FACT,$/SEAT MI,BLK HR/YR:*T55,3F12.4)
      WRITE(6,770) W(9),W(10),HX(4),HX(5),HX(6),GX(32)
  770 FORMAT(/5X*SOME GEOMETRY CALCULATIONS*/
     $ 10X*WING...SPAN,CMAC:*T55,2F12.4/
     $ 10X*TAIL...SPAN,CMAC,VBAR,SWEEP*T55,4F12.4)
      WRITE(6,780) HX(16),HX(17),HX(18),HX(19),HX(20),GX(35)
  780 FORMAT(10X*VERT. TAIL...VBAR,TAPER,AR,SWEEP,SR/SV,SV:*
     $ T55,6F12.4)
      GO TO 20
   40 IF(OUTPUT.LT.1) GO TO 100
      IF(OUTPUT.GT.1) GO TO 549
   42 FORMAT(*1*//30X*AIRCRAFT SIZING PROGRAM*//)
   44 FORMAT(5X*DESIGN VARIABLES*/10X*WING AREA (FTXX2)=*
     $T40,F15.4/10X*WING ASPECT RATIO=*T40,F15.4/10X
     $*FUSELAGE LENGTH (FT)=*T40,F15.4/10X
     $*HOR. TAIL AREA (FTXX2)=*T40,F15.4/10X
     $*HOR. TAIL ASPECT RATIO=*T40,F15.4/10X
     $*TOTAL THRUST (LBS)=*T40,F15.4/10X"AFT MOST CG="T40,F15.4/
     $ 10X"CRUISE MACH NO.="T40,F15.4/10X"SWEEP="T40,F15.4/10X
     $ "WING T/C="T40,F15.4/10X"WING TAPER RATIO="T40,F15.4/
     $ 10X,"FUSE. DIA=",T40,F15.4)
      WRITE(6,52)
   52 FORMAT(*1*//30X*DIRECT OPERATING COSTS--DOLLARS/FLT. HOUR*)
```

```
      WRITE(6,54) DEPRE,PER(1),SUPPORT,PER(2),SPARES,PER(3),DELAY,
     $ PER(4),INSUR,PER(5),FCOST,PER(6),XMCOST,PER(7),FEELAND,PER(8),
     $ CREW,PER(9),ATT,PER(10),SERVICE,PER(11),CONTROL,PER(12)
   54 FORMAT(//10X*DEPREC*T40,2F10.2/10X*SUPPORT*T40,2F10.2/10X*SPARES*
     $T40,
     $2F10.2/10X*DELAY*T40,2F10.2/10X*INSURANCE*T40,2F10.2/10X*FUEL*T40,
     $2F10.2/10X*MAINTENANCE*T40,2F10.2/10X*LANDING FEE*T40,2F10.2/10X
     $*CREW*T40,2F10.2/10X*ATTENDANTS*T40,2F10.2/10X*FUEL SERVICE*
     $ T40,2F10.2/
     $10X*CONTROL*T40,2F10.2)
      WRITE(6,56) DOC,TOT
   56 FORMAT(/3X*TOTAL DIRECT OPERATING COSTS*T40*$*F9.2,F10.2)
      WRITE(6,150)
  150 FORMAT(///30X*INDIRECT OPERATING COSTS--DOLLARS/FLT. HOUR*///)
      DO 300 I=1,12
      PER(I)=100.0*XIOC(I)/TOTIOC
      WRITE(6,152) YIOC(I),XIOC(I),PER(I)
  152 FORMAT(10X,A10,T40,2F10.2)
  300 CONTINUE
      WRITE(6,154) TOTIOC,TOT
  154 FORMAT(/5X*TOTAL INDIRECT OPERATING COSTS*T40,2F10.2/)
  549 WRITE(6,550) REVHR,COSTHR,ROI
  550 FORMAT(*1*///30X*PERFORMANCE FUNCTION SUMMARY*///
     $ 10X*REVENUE PER BLOCK HOUR*T50,F12.2/
     $ 10X*TOTAL COST PER BLOCK HOUR*T50,F12.2/
     $ 10X*RETURN ON INVESTMENT*T50,F12.4///)
      DATA YCOST/6HDOC/HR,7HDOC/FLT,3HROI,4HFARE,10HSEAT-MI/GA,
     $ 8HL/D(MAX),5HMTOGW,4HFARE,5HPRICE/
      DO 570 I=1,9
      WRITE(6,560) I,YCOST(I),CST(I)
  560 FORMAT(10X,I5,2X,A10,F12.3)
  570 CONTINUE
  100 RETURN
      END
      SUBROUTINE CGCAL
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      COMMON /STAB/DERIVCR(15),DERIVAP(15),STOR(20)
      COMMON /GRAVITY/ CG(6)
*******************************
*     ASSIGN CG POSITIONS     *
*     (1) AFT-CRUISE          *
*     (2) AFT-APPROACH        *
*     (3) FWD-CRUISE          *
*     (4) FWD-APPROACH        *
*******************************
      CG(1)=DESIGN(7)
      CG(2)=CG(1)
      CG(3)=CG(1)-GX(23)/W(10)
      CG(4)=CG(2)-GX(23)/W(10)
      DATA CG(6)/.18/
C
C     ALLOW FREE GEAR LOCATION
```

```
C
      IF(ITERM(7).GT.0.) CG(5)=CG(1)-CG(6)
      RETURN
      END
      SUBROUTINE GEO
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
**********************************************
*     WING CONSTANTS OR COMMON VARIABLES     *
**********************************************
      W(9)=(DESIGN(1)*DESIGN(2))**0.5
      W(10)=4.*DESIGN(1)*(1.+W(3)+W(3)*W(3))/
     $(3.*W(9)*(1.+W(3))**2.0)
      W(11)=TAN(W(1)/GX(2))
      W(12)=.496+0.45/DESIGN(2)
      W(13)=W(12)-.9/DESIGN(2)
      W(15)=COS(W(1)/GX(2))
*******************************************************
*     HORIZONTAL TAIL CONSTANTS OR COMMON VARIABLES     *
*******************************************************
      HX(4)=(DESIGN(4)*DESIGN(5))**0.5
      HX(5)=4.*DESIGN(4)*(1.+HX(1)+HX(1)*HX(1))/
     $(3.*HX(4)*(1.+HX(1))**2.0)
      XTXLT=0.45
      IF(ITERM(4).EQ.3) XTXLT=0.43
      XLT=XTXLT*DESIGN(3)
      GX(33)=XLT
      HX(6)=DESIGN(4)*XLT/(DESIGN(1)*W(10))
      HX(7)=TAN(GX(32)/GX(2))
      T=(1.-HX(1))/(4.*(1.+HX(1)))
      HX(8)=HX(7)+4.*T/DESIGN(5)
      HX(9)=HX(8)-2.*(1.-HX(1))/(DESIGN(5)*(1+HX(1)))
**********************************************
*     MISC. CONSTANTS OR COMMON VARIABLES     *
**********************************************
      DATA GX(1),GX(2)/3.14159265,57.295779/
      DATA HX(15),GX(13),GX(31)/0.,0.,0./
      DATA GX(15)/32.174/
      GX(35)=HX(16)*W(9)*DESIGN(1)/(XLT*0.95)
      RETURN
      END
      SUBROUTINE WEIGHT(OUTPUT)
      REAL LT,F1(2),F2(2),ANS(4),DWTA(41)
      COMMON /DEBUG/IDEBUG,IDEBUG2
      COMMON /FCOUNT/ICNT,JCNT,KCNT,NDAV,LCNT,IWT,NOIT
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /DRAG/CDS(6),CDSAP(6)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      COMMON /WTSVE/WTS(20)
      INTEGER OUTPUT
      DATA WFOWTO/.26/
      DO 20 I=1,41
   20 DWTA(I)=0.
```

```
***************************************************
*     FUDGE IS A FACTOR FOR WEIGHT OVERRUNS       *
***************************************************
      FUDGE=1.05
      IF(IDEBUG.GT.0) WRITE(6,337)
  337 FORMAT(*1*/20X*WEIGHT ITERATION LOOPS*)
***************************************
*     SOME GEOMETRY DEFINITIONS       *
***************************************
      IACT=ITERM(1)
      ICGX=ITERM(2)
      AR=DESIGN(2)
      LT=DESIGN(1)*W(10)*HX(6)/DESIGN(4)
      SHT=DESIGN(4)
      TRV=HX(17)
      SVT=HX(16)*W(9)*DESIGN(1)/LT
      TENG=DESIGN(6)/ITERM(4)
      NOIT=0
      DELWTO=20.
      DIV=1.
      WTO=WTS(1)
      WTINIT=WTO
      WTFUEL=WTO*WFOWTO
      FUEL=WTFUEL/6.4
      WTENG=FUDGE*(TENG/GX(21))*GX(20)*ITERM(4)
      WTS(7)=WTS(1)/(GX(15)*0.000889*W(10)*DESIGN(1))
      WTS(9)=(WTS(1)-WTFUEL)/(GX(15)*0.002378*W(10)*DESIGN(1))
*********************************************************************
*     THIS SECTION CALCULATES WTS. INDEPENDENT OF WTO AND FUEL      *
*********************************************************************
      F1(1)=1.
      F1(2)=1.15
      WECTL=F1(IACT+1)*88.46*((190.+W(9))*4.0E-2)**0.294
      WTSRT=49.19*(4.0*WTENG*1.0E-3)**0.541
      PASS=GX(19)*170.
      WTFURN=39.51*GX(19)
      WTFOOD=214.5
      WTO2=300.7
      WTWIN=501.55
      WTBGH=144.72
      WTAC=3647.
      WTTR=1500.
      F2(1)=0.
      F2(2)=250.
      ACTCON=F2(IACT+1)
      DATA CREW/1700./
      BAGGAGE=GX(19)*35.0
      CARGO=GX(22)
*************************************************************
*     THIS SECTION COMPUTES WEIGHTS DEPENDENT UPON WTO      *
*         BEGIN WEIGHT ITERATION THIS SECTION               *
*************************************************************
   50 WTWING=2.0*FUDGE*0.00428*(DESIGN(1))**0.48*AR*(GX(3)-0.05)**0.43
```

```
     $*(WTO*1.5*(3.0-PX(8)))**0.84*W(3)**0.14/((100.0*W(4))**0.76
     $*(COS(W(1)/GX(2)))*1.54)
      FUDHT=1.
      IF(DESIGN(5).GT.5.0) FUDHT=1.+(DESIGN(5)-5.0)/10.
*****************************************************************************
*     THIS IS A LINEAR CORRRECTION FACTOR FOR TAIL ASPECT RATIOS   5;   *
*     INTENDED AS A PENALTY TERM TO TRY AND REFLECT THE RANGE OF         *
*     TAIL WEIGHT EQUATION VALIDITY                                        *
*****************************************************************************
      TTEMP=HX(5)*1.5*HX(2)
      TEMP=(WTO*4.5)**0.813*(SHT**0.584)*(HX(4)/TTEMP)**0.033
     $*(W(10)/LT)**0.28
      WTHT1=2.0*(0.0034*TEMP**0.915)*FUDGE
      WTHT2=2.0*.00563*(WTO**0.6)*(SHT**.469)*(DESIGN(5)/.75)**.539
     $ *((1.+HX(1))/HX(2))**0.692
      WTHT3=1.0*(4.566*1.0E-4*(DESIGN(4)**.48)*DESIGN(5)*
     $ (WTO*4.5/1.)**0.84)
*****************************************************************************
*     WTHT1 -- FROM NICOLAI (REF. 4)                                       *
*     WTHT2 -- FROM VDEP (REF. 6)                                          *
*     WTHT3 -- ANALOGOUS TO WING WEIGHT EQUATION                           *
*                                                                          *
*     THE THREE HORIZONTAL TAIL WEIGHT EQUATIONS ARE THEN AVERAGED        *
*****************************************************************************
      WTHT=((WTHT1+WTHT2+WTHT3)/3.0)*FUDHT
      TEMPV=1.02*(4.5*WTO)**.363*SVT**1.089*(GX(3)*0.8)**.601
     $ *LT**(-.726)*(1.+HX(20))**.217*HX(18)**.337*
     $(1.+TRV)**.36*(COS(HX(19)/GX(2)))**(-.484)
      WTVT=(0.19*TEMPV**1.014)*FUDGE
      TQ=10.43*(.000364*((GX(3)+0.04)*971.15)**2.0)**0.283
      WTFUSE1=((TQ*(DESIGN(3)/GX(5))**0.71)*(WTO*1.0E-3)**0.95)*FUDGE
      WTFUSE2=0.0796*2.1861*(WTO**0.33)*(DESIGN(3)**0.76)
     $ *(GX(5)+GX(5))**1.2
*****************************************************************
*     WTFUSE1 -- FROM NICOLAI (REF. 4)                          *
*     WTFUSE2 -- FROM VDEP (REF. 6)                             *
*                                                               *
*     THE TWO FUSELAGE WEIGHT EQUATIONS ARE THEN AVERAGED      *
*****************************************************************
      WTFUSE=(WTFUSE1+WTFUSE2)/2.
      WTLG=2.0*(62.21*(WTO*1.0E-3)**0.84)*FUDGE
      WTCTL=(56.01*(WTO*245.6*1.0E-5)**0.576)*FUDGE
      WTINST=2.0*(15.+.032*WTO*1.E-3)+4.*(4.8+.006*WTO*1.0E-3)+.15*WTO*1
     $.0E-3
      WTMISC=(0.771*WTO*1.0E-3*1.1)*FUDGE
      IF(NOIT.LT.20) GO TO 94
      IF(DELWTO.LT.20.0) GO TO 96
***********************************
*     CALCULATE FUEL WEIGHT       *
***********************************
   94 CALL FUELCAL(WTO,WTFUEL,0)
      FUEL=WTFUEL/6.4
      GO TO 98
```

```
   96 WTFUEL=WTO*WFOWTO
      FUEL=WTFUEL/6.4
*****************************************************************
*     THIS SECTION CALCULATES WTS. DEPENDENT UPON FUEL WT.     *
*****************************************************************
   98 WTF1=41.6*(FUEL*1.0E-2)**0.818
      WTF2=7.91*(FUEL*1.0E-2)**0.854
      WTF3=7.38*(FUEL*1.0E-2)**0.458
      WTF4=ICGX*28.38*(FUEL*1.0E-2)**0.442
      WTFSYS=WTF1+WTF2+WTF3+WTF4
      WTELEC=1162.66*((WTFSYS+931.3)*1.0E-3)**0.506
*****************************************************************
*     THIS SECTION CALCULATES THE EMPTY AND FIXED WEIGHTS      *
*****************************************************************
      T1=WTWING+WTHT+WTVT+WTFUSE+WTLG+WTCTL+WTINST+WTMISC+WECTL+WTSRT
     $+WTFURN+WTO2+WTWIN+WTBGH+WTAC+WTTR+ACTCON
      WTEMPTY=(T1+WTENG+WTELEC+WTFSYS)*FUDGE
      WTEMPTY=(1.0-PX(3))*WTEMPTY
      WTFIXED=WTFOOD+PASS+CREW+BAGGAGE+CARGO
*****************************************************
*     THIS SECTION COMPUTES THE TAKE-OFF WEIGHT    *
*****************************************************
      WTONEW=WTFIXED+WTFUEL+WTEMPTY
      DELWTO=WTONEW-WTO
      NOIT=NOIT+1
      IWT=IWT+1
      IF(NOIT.GT.39) GO TO 58
      DIV=1.2
      IF(NOIT.LT.3) DIV=0.85
      IF(NOIT.GT.10) DIV=1.95
      IF(IDEBUG.GT.0) WRITE(6,338) NOIT,WTO,WTONEW,DELWTO
     $ ,WTFUEL,WFOWTO
  338 FORMAT(5X*ITER. NO.=*I5,10X*WTO,WTONEW,DELWTO,WTFUEL,WFOWTO=*
     $ 3F12.2,F12.2,F12.5)
       WTO=WTONEW+DELWTO/DIV
      DWTA(NOIT)=DELWTO
***********************************************
*     ITERATE WEIGHT UNTIL WITHIN 0.05 LBS    *
***********************************************
      IF(ABS(DELWTO).GT.0.05) GO TO 50
   57 IF(OUTPUT.EQ.0) GO TO 150
      GO TO 60
   58 WRITE(6,59) WTO,WTONEW,DELWTO,WTINIT,WFOWTO,WTFUEL,NOIT
   59 FORMAT(/*  / / / WEIGHT LOOP DID NOT CONVERGE/ / / *
     $ /10X*WTO,WTONEW,DELWTO,WTINIT,WFOWTO,WTFUEL,NOIT=*4F12.2,
     $ F12.4,F12.2,I5/)
      WRITE(6,64) DWTA
   64 FORMAT(10X*DELWTO*9(T25,5F12.3/))
      WRITE(6,80)WTWING,WTHT,WTVT,WTFUSE,WTLG,WTCTL,WTINST,WTMISC,WTELEC
     $,WTSRT,WTFURN,WTO2,WTWIN,WTBGH,WTAC,WTTR,WTENG,WECTL,WTFSYS,ACTCON
      WRITE(6,82) WTEMPTY
      WRITE(6,84) PASS,CREW,BAGGAGE,WTFOOD,CARGO,WTFIXED
      WRITE(6,86) WTFUEL,WTO,NOIT
```

```
      WRITE(6,88) WTHT1,WTHT2,WTHT3,WTFUSE1,WTFUSE2,WTINIT
      CALL XOUTPUT(6)
      WRITE(6,887) ICNT,JCNT,KCNT,NDAV,LCNT,IWT,NOIT
  887 FORMAT(/5X"DUMP OF FCOUNT COMMON BLOCK"/10X
     $ "ICNT,JCNT,NDAV,KCNT,LCNT,IWT,NOIT=",10X,7I5/)
      WTO=WTONEW+DELWTO/2.0
      WTS(1)=WTO
      GO TO 57
*****************************
*     OUTPUT SECTION        *
*****************************
   60 WRITE(6,80)WTWING,WTHT,WTVT,WTFUSE,WTLG,WTCTL,WTINST,WTMISC,WTELEC
     $,WTSRT,WTFURN,WTO2,WTWIN,WTBGH,WTAC,WTTR,WTENG,WECTL,WTFSYS,ACTCON
   80 FORMAT(*1*//30X,23H***WEIGHT ESTIMATION***//10X*WING*T40,F10.1
     $/10X*HOR. TAIL*T40,F10.1/10X*VERT. TAIL*T40,F10.1/10X*FUSELAGE*T40
     $,F10.1/10X*LANDING GEAR*T40,F10.1/10X*CONTROL SYSTEM*T40,F10.1/10X
     $*INSTRUMENTS*T40,F10.1/10X*MISC INTERIOR*T40,F10.1/10X*ELECTRICAL*
     $T40,F10.1/10X*STARTERS*T40,F10.1/10X*FURNISHINGS*T40,F10.1/10X
     $*OXYGEN*T40,F10.1/10X*WINDOWS*T40,F10.1/10X*BAGGAGE HNDLING*T40,
     $F10.1/10X*AIR CONDITIONING*T40,F10.1/10X*THRUST REVERSER*T40,F10.1
     $/10X*ENGINE*T40,F10.1/10X*ENGINE CONTROLS*T40,F10.1/10X*FUEL SYS*
     $T40,F10.1/10X*ACTIVE CONTROL SYSTEMS*T40,F10.1)
      WRITE(6,82) WTEMPTY
   82 FORMAT(/5X*EMPTY WEIGHT*T40,F10.1)
      WRITE(6,84) PASS,CREW,BAGGAGE,WTFOOD,CARGO,WTFIXED
   84 FORMAT(//10X*PASSENGERS*T40,F10.1/10X*CREW*T40,F10.1/10X
     $*BAGGAGE*T40,F10.1/10X*WTFOOD*T40,F10.1/10X*CARGO*T40,F10.1
     $//5X*FIXED WEIGHT*T40,F10.1)
      WRITE(6,86) WTFUEL,WTO,NOIT
   86 FORMAT(//5X*FUEL*T40,F10.1///5X*TAKE-OFF WEIGHT*T40,F10.1///10X
     $*NO. OF ITERATIONS REQUIRED*T40,I10)
      WRITE(6,88) WTHT1,WTHT2,WTHT3,WTFUSE1,WTFUSE2,WTINIT
   88 FORMAT(/5X*WTHT(1,2,3),WTFUSE(1,2),WTINIT=*T45,5F10.2,F12.2)
      CALL FUELCAL(WTO,WTFUEL,1)
      CALL XLOD(GX(9),GX(3),WTS(11),EFF,1,OUTPUT)
*****************************************
*     ASSIGN DATA BASE VARIABLES        *
*****************************************
  150 WTS(1)=WTO
      WTS(2)=WTEMPTY
      WTS(4)=.75*WTEMPTY
      WTS(6)=WTFUEL
      IF(NOIT.LT.40) WFOWTO=WTFUEL/WTO
      AENG=0.
      IF(ITERM(4).EQ.3) AENG=WTENG*0.81*LT*LT/(GX(15)*3.)
      WTS(13)=13.45E6
      WTS(14)=13.33E6
      CALL AT62(WTS(11),ANS)
      WTS(7)=WTS(1)/(ANS(1)*W(10)*DESIGN(1)*GX(15))
      WTS(8)=WTS(13)*GX(15)/WTS(1)
      CALL AT62(500.,ANS)
      WTS(9)=GX(27)*WTS(1)/(GX(15)*DESIGN(1)*ANS(1)*W(10))
      WTS(10)=WTS(14)*GX(15)/(WTS(1)*GX(27))
```

```
      IF(IDEBUG2.GT.0) WRITE(6,990) NOIT
  990 FORMAT(T100*NOIT=*I10)
      RETURN
      END
      SUBROUTINE CRUALT(WT,CLCR,M,ALT)
*****************************************************************************
*     THIS SUBROUTINE RETURNS ALTITUDE TO SATISFY THE SPECIFIED           *
*     WEIGHT, CL, AND M.  ITERATIVE TABLE LOOK UP IS USED.                *
*****************************************************************************
      REAL M,HOLD,HNEW,ANS(4)
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      IC=0
      DATA TO,PO,ALF,R/518.14,2116.229,.00356617,53.3/
      P=2.0*WT/(1.4*DESIGN(1)*CLCR*M*M)
      HOLD=(1.0-(P/PO)**(ALF*R))*TO/ALF
   50 CALL AT62(HOLD,ANS)
      DP=P-ANS(2)
      DRDH=-1.0*ANS(1)*GX(15)
      HNEW=HOLD+DP*0.9/DRDH
      IC=IC+1
      IF(IC.GT.100) GO TO 100
      DALT=HNEW-HOLD
      IF(ABS(DALT).LT.7.5)  GO TO 150
      HOLD=HNEW
      GO TO 50
  100 WRITE(6,102) HNEW
  102 FORMAT(*1*////40X,38H***CRUISE ALTITUDE DID NOT CONVERGE***//10X
     $*LAST ALTITUDE=*F12.2)
  150 ALT=HNEW
  152 FORMAT(///10X,*ALTITUDE,NO. OF ITERATIONS=*F12.2,I10)
      RETURN
      END
      SUBROUTINE ENGINE(ALT,M,TCTM,TSFC)
C
C     EMPIRICAL ADJUSTMENTS HAVE BEEN MADE TO THIS SUBROUTINE TO
C     MATCH DATA.
C
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      REAL M,HN,ANS(4)
      CALL AT62(ALT,ANS)
      DATA TO,PO/561.2,2116.229/
      HN=ALT/40000.
C
C     TCTM FROM REF(7) -- AGGARWAL, ET. AL.
C
      TCTM=(1000./41100.)*(30.06-34.74*HN+7.25*M+12.11*HN*HN
     $ -1.12*M*HN-13.96*M*M-5.46*M*HN*HN+11.82*M*M*HN
     $ -.01*M*M*HN*HN)
      ALTK=((ALT-35332.)*.004/1000.)+1.
      IF(ALT.GT.35332) GO TO 40
      TCTM=(1000./41100.)*(34.71-32.22*HN-27.4*M
     $ +5.8*HN*HN+36.43*HN*M+2.65*M*M-11.64*M*HN*HN
```

```
     $ +3.19*M*M*HN-5.72*M*M*HN*HN)
      ALTK=1.
   40 DELTA=ANS(2)/PO
      THETA=ANS(3)/TO
      TCTM=1.2*TCTM
      TN=TCTM*41100./DELTA/20000.
      FN=1000.*(2.04+3.71*TN-4.38*M+1.69*TN*TN+5.94*M*TN
     $ +12.99*M*M)
      FNS=FN*ALTK
      F=FNS*DELTA*(THETA)**0.5
      TSFC=F/(TCTM*41100.)
      TSFC=(1.0-PX(2))*TSFC
      RETURN
      END
      SUBROUTINE CDZL(ALT,CDO,M,OUTPUT)
      REAL MU(2),ANS(4),U(2),K(2),DCD(2),M
      INTEGER OUTPUT
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /DRAG/CDS(6),CDSAP(6)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      CALL AT62(ALT,ANS)
      DATA MU/3.6878E-7,2.9652E-7/
      IALT=1
      IF(ALT.GT.10000.) IALT=2
      SHT=DESIGN(4)
      TENG=DESIGN(6)/ITERM(4)
      U(1)=0.6*ANS(4)
      U(2)=M*ANS(4)
      SREF=DESIGN(1)
***********************************
*     WING DRAG CALCULATIONS      *
***********************************
      REW=W(10)*U(IALT)*ANS(1)/MU(IALT)
      CF=0.455/(ALOG10(REW))**2.58
      CF=CF*(1.0-PX(4))
      SWET=2.05*SREF
      K(1)=1.2230
      K(2)=1.456
      CDOW=CF*K(IALT)*SWET/SREF
***********************************************
*     HORIZONTAL TAIL DRAG CALCULATIONS       *
***********************************************
      REH=HX(5)*U(IALT)*ANS(1)/MU(IALT)
      CF=0.074/REH**0.2
      K(1)=1.2560
      K(2)=1.3803
      SWET=SHT*2.05
      CDOH=CF*K(IALT)*SWET/SREF
********************************************
*     VERTICAL TAIL DRAG CALCULATIONS      *
********************************************
      TRV=HX(17)
      SV=GX(35)
```

```
      BV=(SV*HX(18))**0.5
      CVBAR=4.*SV*(1.+TRV+TRV*TRV)/(3.*BV*(1.+TRV)**2.0)
      REV=CVBAR*U(IALT)*ANS(1)/MU(IALT)
      CF=.074/REV**0.2
      K(1)=1.1413
      K(2)=1.3504
      SWET=2.05*SV
      CDOV=CF*K(IALT)*SWET/SREF
******************************************
*     FUSELAGE DRAG CALCULATIONS         *
******************************************
      REF=DESIGN(3)*U(IALT)*ANS(1)/MU(IALT)
      CF=0.0455/(ALOG10(REF))**2.58
      DIA=GX(5)
      FLOD=DESIGN(3)/DIA
      RDIA=DIA/2.0
      ANOSE=2.*GX(1)*RDIA*RDIA
      CONEL=(DESIGN(3)-RDIA)*.25
      ABOD=GX(1)*GX(5)*(DESIGN(3)-RDIA-CONEL)
      SLANT=(CONEL*CONEL+RDIA*RDIA)**0.5
      ACONE=GX(2)*RDIA*SLANT
      SFUS=ANOSE+ABOD+ACONE
      RAT=(1.+60./(FLOD*FLOD*FLOD)+.0025*FLOD)*SFUS/50.
      CDFF=CF*RAT
      CDBF=5.0112E-5/(CDFF)**0.5
      CDOF=CDFF*50.0/SREF
**********************************************
*     ENGINE NACELLE DRAG CALCULATIONS       *
**********************************************
      EL=GX(6)*(TENG/GX(21))**0.5
      EDIA=GX(7)*(TENG/GX(21))**0.5
      SWET=GX(1)*EDIA*EL
      REE=EL*U(IALT)*ANS(1)/MU(IALT)
      CF=0.074/REE**0.2
      TT=2
      IF(ITERM(4).GT.3) TT=4
      CDOE=1.0033*CF*SWET*TT/SREF
****************************************************************************
*     SUM UP DRAG COMPONENTS                                               *
*                                                                          *
*     CRUD DRAG   (IALT=2)                                                 *
*     LANDING GEAR DOWN (0.0150)  +  10 DEGREES FLAPS (0.008)              *
*                          (IALT=1)                                        *
****************************************************************************
      DCD(1)=0.023
      DCD(2)=0.0007
      CDO=(CDOW+CDOH+CDOV+CDOV+CDOF+CDBF+CDOE+DCD(IALT))*1.05
      IF(IALT.NE.1) GO TO 76
***********************************
*     DATA BASE ASSIGNMENTS       *
***********************************
      CDSAP(1)=CDO
      CDSAP(2)=CDOW
```

```
      CDSAP(3)=CDOH
      GO TO 78
   76 CDS(1)=CDO
      CDS(2)=CDOW
      CDS(3)=CDOH
      CDS(6)=CDOF
*************************
*     OUTPUT SECTION      *
*************************
   78 IF(OUTPUT.EQ.0) GO TO 100
      WRITE(6,80) CDOW,CDOH,CDOV,CDOF,CDBF,CDOE,DCD(IALT),CDO
   80 FORMAT(*1*//15X,19H***DRAG ANALYSIS***,//10X*WING*T35,F10.4/10X
     $*HORIZONTAL TAIL*T35,F10.4/10X*VERTICAL TAIL*T35,F10.4/10X*FUSELAG
     $E*T35,F10.4/10X*BASE*T35,F10.4/10X*ENGINE NACELLE*T35,F10.4/10X
     $*CRUD/FLAPS*T35,F10.4//5X
     $*AIRCRAFT DRAG*T35,F10.4///10X*INTERFERENCE FACTOR IS 5 PERCENT*)
      WRITE(6,82) REW,REH,REV,REF,REE
   82 FORMAT(////15X,23H***REYNOLD'S NUMBERS***//10X*WING*T35,F12.1/10X,
     $*HORIZONTAL TAIL*T35,F12.1/10X*VERTICAL TAIL*T35,F12.1/10X*FUSELAG
     $E*T35,F12.1/10X*ENGINE*T35,F12.1//)
      IF(IALT.EQ.1) WRITE(6,84)
   84 FORMAT(///30X,14H***APPROACH***/)
  100 RETURN
      END
      SUBROUTINE STABCOD(CL,M,ICG,OUTPUT)
      INTEGER OUTPUT
      REAL DCDM(2),XACDM(2),ANS(4)
      REAL KA,KTR,KH,M,KWB,K
      REAL LF
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      COMMON /GRAVITY/CG(6)
      COMMON /STAB/DERIVCR(15),DERIVAP(15),STOR(20)
      COMMON /WTSVE/WTS(20)
**************************************************************************
*     THIS SUBROUTINE CALCULATES STABILITY AND CONTROL DERIVATIVES       *
*     IN EITHER APPROACH OR CRUISE CONFIGURATIONS                        *
*          ICR=0  (APPROACH); ICR=1 (CRUISE)                             *
**************************************************************************
      XCG=(CG(1)+CG(3))/2.0
      IF(ICG.LT.7) XCG=CG(ICG)
      ICR=1
      ELASTK=0.875
      IF(M.LT.0.45) ELASTK=0.825
      IF(M.LT.0.46) ICR=0
*****************************************************
*     THE FOLLOWING SECTION CALCULATES CL(ALPHA)      *
*****************************************************
      KWB=1.0-0.25*GX(5)*GX(5)/(W(9)*W(9))+0.025*GX(5)/W(9)
      BETA=(1.0-M*M)**0.5
      PIAR=DESIGN(2)*GX(1)
      K=1.0
      XNUM=2.0*PIAR
```

```
      T1=DESIGN(2)*DESIGN(2)*BETA*BETA/(K*K)
      T2=1.0+W(13)*W(13)/(BETA*BETA)
      CLAW=XNUM*ELASTK/(2.0+(T1*T2+4.0)**0.5)
      T1=DESIGN(2)*DESIGN(2)/(K*K)
      T2=1.0+W(13)*W(13)
      CLAWM0=XNUM*.725/(2.0+(T1*T2+4.0)**0.5)
      KA=1./DESIGN(2)-1.0/(1.+DESIGN(2)**1.7)
      KTR=(10.-3.*W(3))/7.
      ZTL=GX(34)
      XLT=DESIGN(1)*W(10)*HX(6)/DESIGN(4)
      KH=(1.0-ZTL/W(9))/(2.0*XLT/W(9))**0.40
      DEDAM0=4.44*(KA*KTR*KH*(W(15))**2.0)**1.119
      DEDA=DEDAM0*(CLAW/CLAWM0)**0.444444
      TM1=(1.0+HX(9)*HX(9)/(BETA*BETA))
      TM2=BETA*BETA*DESIGN(5)*DESIGN(5)
      CLAH=2.0*GX(1)*ELASTK*DESIGN(5)/(2.0+(TM2*TM1+4.0)**0.5)
      CLAWB=KWB*CLAW
      ETAH=0.90
      CLA=CLAWB+CLAH*ETAH*DESIGN(4)*(1.0-DEDA)/DESIGN(1)
      EWING=W(7)
************************************************************
*     THE FOLLOWING STATEMENT CALCULATES CD(ALPHA)         *
************************************************************
      CDA=2.0*CL*CLA/(PIAR*EWING)
*********************************************************
*     THE FOLLOWING SECTION CALCULATES CM(ALPHA)        *
*********************************************************
      XACW=.25-.265*M*M/(DESIGN(2))**0.5
      DXACT=XLT/W(10)
      AW=CLAW/GX(2)
      XNRM=DESIGN(1)*W(10)*AW
      TFUS=0.5836+.1690*(1.0-DEDA)
      DXACF=-GX(5)*GX(5)*DESIGN(3)*TFUS/(XNRM*36.5)
      DMWDF=GX(5)*W(10)/(DESIGN(1)*AW*290.)
      DXACF=DXACF+DMWDF
      STOR(11)=DXACF*AW
      T=DESIGN(6)/ITERM(4)
      WF=GX(6)*(T/GX(21))**0.5
      LF=GX(7)*(T/GX(21))**0.5
      TT=2.
      IF(ITERM(4).GT.3) TT=4
      DXACE=-TT*LF*WF*WF/(15.21*XNRM)
      STOR(12)=DXACE*AW
      TA1=CLAH*ETAH*DESIGN(4)*(1.0-DEDA)/(CLAWB*DESIGN(1))
      XACWB=XACW+DXACF+DXACE
      XAC=(XACWB+TA1*DXACT)/(1+TA1)
      DCMCL=XCG-XAC
      CMA=DCMCL*CLA
*******************************************************************
*     THE FOLLOWING SECTION CALCULATES VELOCITY DERIVATIVES       *
*******************************************************************
      CLU=(1.-M)*M*M*CL/(1.0-M*M)
      DATA DCDM,XACDM/0.0202,0.1005,-3.1E-4,5.1E-4/
```

```
      Z=0.
      IF(ICR.GT.0) Z=40000.
      CALL AT62(Z,ANS)
      AM=ANS(4)
      CDU=DCDM(ICR+1)/AM+2.*CL*CLU/(PIAR*EWING*10.)
      CMU=-CL*XACDM(ICR+1)
*******************************************************************
*     THE FOLLOWING SECTION CALCULATES Q DERIVATIVES      *
*******************************************************************
      CLQH=2.0*CLAH*ETAH*HX(6)
      CLQ=CLQH*1.1
      CDQ=0
      CMQT=-2.0*CLAH*ETAH*0.9*HX(6)*XLT/W(10)
******************************************************************************
*     THE FOLLOWING SECTION CALCULATES ALPHA-DOT DERIVATIVES      *
******************************************************************************
      CMQ=CMQT*1.1
      CLADOT=-2.0*CLAH*ETAH*HX(6)*DEDA
      CDADOT=0.0
      CMADOT=-2.0*CLAH*ETAH*HX(6)*XLT*DEDA/W(10)
****************************************************************************
*     THE FOLLOWING SECTION CALCULATES CONTROL DERIVATIVES      *
****************************************************************************
      CLDEL=0.9*DESIGN(4)*CLAH*0.46/DESIGN(1)
*************************************************************************************
*         THIS IS PER DEGREE CONTROL DEFLECTION, OTHER PER RAD.         *
*************************************************************************************
      CDDEL=DESIGN(4)*1.980E-3/DESIGN(1)
      CMDEL=-CLAH*HX(6)*0.9*0.46
      IF(ICR.LT.1) GO TO 120
      XMU=215.
      IF(OUTPUT.GT.-1) XMU=WTS(7)
**************************************
*     DATA BLOCK ASSIGNMENTS      *
**************************************
      STOR(1)=CLAW
      STOR(2)=CLAH
      STOR(3)=DEDA
      STOR(4)=XAC
      STOR(5)=XAC-CMQ/(4.*XMU)
      STOR(13)=TA1
      DERIVCR(1)=CLA
      DERIVCR(2)=CDA
      DERIVCR(3)=CMA
      DERIVCR(4)=CLU
      DERIVCR(5)=CDU
      DERIVCR(6)=CMU
      DERIVCR(7)=CLQ
      DERIVCR(8)=CDQ
      DERIVCR(9)=CMQ
      DERIVCR(10)=CLADOT
      DERIVCR(11)=CDADOT
      DERIVCR(12)=CMADOT
```

```
      DERIVCR(13)=CLDEL
      DERIVCR(14)=CDDEL
      DERIVCR(15)=CMDEL
      GO TO 150
  120 DERIVAP(1)=CLA
      DERIVAP(2)=CDA
      DERIVAP(3)=CMA
      DERIVAP(4)=CLU
      DERIVAP(5)=CDU
      DERIVAP(6)=CMU
      DERIVAP(7)=CLQ
      DERIVAP(8)=CDQ
      DERIVAP(9)=CMQ
      DERIVAP(10)=CLADOT
      DERIVAP(11)=CDADOT
      DERIVAP(12)=CMADOT
      DERIVAP(13)=CLDEL
      DERIVAP(14)=CDDEL
      DERIVAP(15)=CMDEL
      XMU=70.
      IF(OUTPUT.GT.-1) XMU=WTS(9)
      STOR(6)=CLA
      STOR(7)=CLAH
      STOR(8)=DEDA
      STOR(9)=XAC
      STOR(10)=XAC-CMQ/(4.*XMU)
      STOR(14)=TA1
  150 IF(OUTPUT.LT.1) GO TO 200
*************************
*     OUTPUT SECTION    *
*************************
      WRITE(6,162)
  162 FORMAT(*1*//20X*STABILITY AND CONTROL DERIVATIVES*//)
      IF(ICR.EQ.0) WRITE(6,163)
  163 FORMAT(25X,14H***APPROACH***//)
      WRITE(6,164) CL,M,XCG
  164 FORMAT(10X*CL,M,CG. POSITION=*3F15.3//)
      WRITE(6,166)
  166 FORMAT(/28X*CL*14X*CD*12X*CM*/)
      IF(ICR.EQ.1) GO TO 180
      WRITE(6,174) DERIVAP
  174 FORMAT(10X*ALPHA*4X,3F15.5/10X*VELOCITY*3F15.4/10X*Q*8X,3F15.5/
     $10X*ALPHA-DOT*3F15.5/10X*ELEVATOR *3F15.5//)
      GO TO 190
  180 WRITE(6,174) DERIVCR
  190 WRITE(6,192) XAC,DCMCL,CMA,STOR
  192 FORMAT(/10X*NEUTRAL POINT*T35,F12.3/10X*STATIC STABILITY*
     $T35,F12.3/10X*CM(ALPHA)*T35,F12.3/10X*STOR=*/
     $2(T35,2F12.3/))
      WRITE(6,196) DXACE,XACWB,XACW,AW,XNRM,TFUS,DXACF,DMWDF,
     $ CLAWB,CLAH,DEDAM0,XLT,ZTL,CLAWM0,BETA,TA1,CLAW,DEDA
     $ ,XAC,KH,KTR,TM1,TM2,KWB,XCG,KA
  196 FORMAT(5(T30,5F12.4/))
```

```
  200 RETURN
      END
      SUBROUTINE TRIM(CL,M,CDO,CLWING,CLTAIL,CRIT,CRDE,EPS,IPHASE
     $,ICG)
      REAL LTOTOQ,LWOQ,LTLOQ,IW,M
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /DRAG/CDS(6),CDSAP(6)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      COMMON /GRAVITY/CG(6)
      COMMON /STAB/DERIVCR(15),DERIVAP(15),STOR(20)
****************************************************************************
*     IPHASE=1 (CR), =0 (10 DEG FLP,FWD CG), =-1 (45 DEG FLP),             *
*     =-2 (45 DEG FLAP, FWD CG), =-3 (45 DEG FLAP, FWD CG, FIND            *
*     CLMAX), =-4 (10 DEG FLAP, FWD CG, FIND CLMAX), =2 (CR, AFT CG)       *
****************************************************************************
      N=0
****************************************************
*     INITIALIZE VARIABLES FROM DATA BASE          *
****************************************************
      XA=.25-.265*M*M/(DESIGN(2))**0.5
      XAC=(CG(1)+CG(3))/2.-XA
      IF(ICG.LT.7) XAC=CG(ICG)-XA
      ZAC=0.08
      ZTLC=-.12
      ST=DESIGN(4)
      CMACW=-.10-.030218*W(11)-.046875*M
      CMACW=CMACW*(1.0-PX(1))
      CDT=CDS(3)
      AT=STOR(2)/GX(2)
      AW=STOR(1)/GX(2)
      DEDA=STOR(3)
      AOL=W(16)
      IF(IPHASE.GT.0) GO TO 10
      AT=STOR(7)/GX(2)
      AW=STOR(6)/GX(2)
      DEDA=STOR(8)
      CDT=CDSAP(3)
      CMACW=CMACW+W(17)
      W(19)=CMACW-W(17)
      AOL=W(16)+GX(17)
      IF(IPHASE.LT.-3) GO TO 10
      IF(IPHASE.EQ.0) GO TO 10
      CMACW=CMACW+W(18)-W(17)
      AOL=W(16)+GX(18)
      CDO=CDO+W(20)
   10 CD=CDO+CL*CL/(DESIGN(2)*GX(1)*W(7))
      CMAFUS=STOR(11)+STOR(12)
      IW=W(2)
      IF(IPHASE.LT.-2) GO TO 200
      LTOTOQ=CL*DESIGN(1)
      LTLOQ=-.1*LTOTOQ
****************************************************************************
*     THIS SECTION CALCULATES CL(WING) AND CL(TAIL) GIVEN CL(TOTAL)        *
```

```
*************************************************************************
   20 LWOQ=LTOTOQ-LTLOQ
      CLWING=LWOQ/DESIGN(1)
      IF(CLWING.GT.3.5) CLWING=3.5
      IF(CLWING.LT.0.1) CLWING=0.1
      CLTAIL=LTLOQ/ST
      IF(ABS(CLTAIL).GT.2.0) CLTAIL=2.0*CLTAIL/ABS(CLTAIL)
      CDTAIL=CDT*DESIGN(1)/ST+CLTAIL*CLTAIL/(GX(1)*DESIGN(5)*0.90)
      ALFAW=CLWING/AW+AOL
      IF(ALFAW.GT.16.0) ALFAW=16.0
      CMFUS=CMAFUS*(ALFAW-IW)
      EPS=CLWING*DEDA/AW
      CRDE=0.
      ALFAT=CLTAIL/AT-CRDE
      CTLPWR=(CMACW+CLWING*XAC+CMFUS-AT*(ALFAW-EPS-IW)*HX(6)*.9)/
     $(AT*HX(6)*0.9)
      TAO=0.46
      CRIT=CTLPWR
      CRDE=0.
      IF(CRIT.LT.5.0) GO TO 40
      CRDE=(CTLPWR-5.0)/TAO
      CRIT=5.0
      GO TO 60
   40 IF(CRIT.GT.-14.0) GO TO 60
      CRDE=(CTLPWR+14.0)/TAO
      CRIT=-14.0
   60 CNW=CLWING*COS((ALFAW-IW)/GX(2))+CD*SIN((ALFAW-IW)/
     $GX(2))
      CCW=CD*COS((ALFAW-IW)/GX(2))-CLWING*SIN((ALFAW-IW)/
     $GX(2))
      CT=CD
C
C           IF TRIJET, THE UNBALANCE IS ONLY ONE ENGINE
C
      IF(ITERM(4).EQ.3) CT=CT/3.0
      CNT=(1./(0.9*HX(6)))*(CNW*XAC+CCW*ZAC+CMFUS-CT*ZTLC+CMACW)
      CLTNEW=(CNT-CDTAIL*SIN((ALFAT-CRIT)/GX(2)))/(COS((ALFAT
     $-CRIT)/GX(2)))
      DTEST=CLTNEW-CLTAIL
      CLTAIL=CLTNEW+DTEST/(N+1)
      IF(ABS(CLTAIL).GT.2.0) CLTAIL=2.0*CLTAIL/ABS(CLTAIL)
      LTLOQ=CLTAIL*ST
      N=N+1
      IF(ABS(DTEST).LT.0.003) GO TO 100
      IF(N.GT.25) GO TO 80
      GO TO 20
   80 WRITE(6,82) CLWING,CLTAIL,CRIT,CRDE,CNW,DTEST
     $,ALFAW,EPS,CMFUS
   82 FORMAT(///20X*TRIM DID NOT CONVERGE*//3(3F15.4/)//)
  100 IF(CLWING.EQ.3.5) WRITE(6,84) CLWING
     $,CLTAIL,CRIT,CRDE,CNW,DTEST
      IF(ALFAW.EQ.16.0) WRITE(6,84) CLWING
     $,CLTAIL,CRIT,CRDE,CNW,DTEST
```

```
 84 FORMAT(///30X*TRIM HIT ALPHA OR CL LIMIT*//2(10X,3F12.4/))
    IF(ALFAW.EQ.16.0) CALL XOUTPUT(6)
    RETURN
*************************************************************************
*     THIS SECTION CALCULATES CL(TAIL) AND CL(TOTAL) GIVEN CL(WING)      *
*************************************************************************
200 CLWING=CL
    CLTAIL=-1.
    CDTAIL=CDT*DESIGN(1)/ST+CLTAIL*CLTAIL/(GX(1)*DESIGN(5)*0.90)
    ALFAW=CLWING/AW+AOL
    IF(ALFAW.GT.16.0) ALFAW=16.0
    CMFUS=CMAFUS*(ALFAW-IW)
    EPS=CLWING*DEDA/AW
    CRDE=0.
    ALFAT=CLTAIL/AT-CRDE
    CTLPWR=(CMACW+CLWING*XAC+CMFUS-AT*(ALFAW-EPS-IW)*HX(6)*.9)/
   $(AT*HX(6)*0.9)
    TAO=0.46
    CRIT=CTLPWR
    CRDE=0.
    IF(CRIT.LT.5.0) GO TO 540
    CRDE=(CTLPWR-5.0)/TAO
    CRIT=5.0
    GO TO 560
540 IF(CRIT.GT.-14.0) GO TO 560
    CRDE=(CTLPWR+14.0)/TAO
    CRIT=-14.0
560 CNW=CLWING*COS((ALFAW-IW)/GX(2))+CD*SIN((ALFAW-IW)/
   $GX(2))
    CCW=CD*COS((ALFAW-IW)/GX(2))-CLWING*SIN((ALFAW-IW)/
   $GX(2))
    CT=CD
    IF(ITERM(4).EQ.3) CT=CT/3.0
    CNT=(1./(0.9*HX(6)))*(CNW*XAC+CCW*ZAC+CMFUS-CT*ZTLC+CMACW)
    CLTNEW=(CNT-CDTAIL*SIN((ALFAT-CRIT)/GX(2)))/(COS((ALFAT
   $-CRIT)/GX(2)))
    CLTAIL=CLTNEW
    LTLOQ=CLTAIL*ST
    LWOQ=CLWING*DESIGN(1)
    LTOTOQ=LTLOQ+LWOQ
    CL=LTOTOQ/DESIGN(1)
    RETURN
    END
    SUBROUTINE FUELCAL(WTO,WTFUEL,OUTPUT)
    COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
    COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
    COMMON /WTSVE/WTS(20)
    REAL ANS1(4)
    INTEGER OUTPUT
    IF(OUTPUT.LT.1) GO TO 30
    WRITE(6,21)
 21 FORMAT(*1*///30X*/ / / CRUISE ANALYSIS/ / / *//)
    WRITE(6,23) RANGE
```

```
   23 FORMAT(10X*TOTAL MISSION RANGE=*F10.2/10X
     $ *CLIMB DISTANCE=      189.00*/10X
     $ *DESCENT DISTANCE=     113.00*/)
      WRITE(6,25)
   25 FORMAT(//2X*LEG*2X*MACH NO.*3X*CL*4X*V*7X*TIME*4X*L/D*
     $ 4X*TSFC*2X*T/T(IN)*1X*ALT(BEG)*2X*ALT(END)*2X*WT(BEG)*
     $ 3X*WT(END)*6X*DIST*4X*TCON*4X*GAMA*5X*CLM*/)
   30 WTBEG=WTO*0.97*0.965
      GX(31)=.0020
      RANGE=GX(4)
      GX(30)=5.
      TOTTIME=0.78
      ALT1=36000.
      ALT11=ALT1
      WT1=WTBEG
      RLOVER=RANGE-302.
      R=RLOVER/10.
****************************************************
*     CALCULATE CRUISE PORTION IN 10 SEGMENTS       *
****************************************************
      DO 40 I=1,10
      CALL CRUFUEL(WT1,R,WT2,ALT1,ALT2,I,TIME,OUTPUT,0)
      WT1=WT2
      ALT1=ALT2
      TOTTIME=TOTTIME+TIME
      IF(I.LT.5) GO TO 40
      IF(I.GT.5) GO TO 40
      WTMID=WT2
      WTS(15)=WTMID
      WTS(11)=ALT2
   40 CONTINUE
**************************************
*     RESERVE MISSION ASSUMPTION     *
**************************************
      WTGRD=WT2
      CALL CRUFUEL(WTGRD,1000.,WTRES,30000.,ALTR,11,TM,OUTPUT,1)
************************************
*     INSERT DATA IN DATA BASE     *
************************************
      FL=TOTTIME
      BLKHR=FL+.327
      SPEED=RANGE/FL
      BLKSPD=RANGE/BLKHR
      WTS(5)=.97*WTGRD
      WTEND=0.97*WTRES
      WTFUEL=WTO-WTEND
      WTS(17)=SPEED
      WTS(18)=FL
      WTS(19)=BLKHR
      WTS(20)=WTBEG-WTS(5)
      WTS(3)=WTS(20)/BLKHR
      WTS(3)=(WTBEG-WTS(5))/TIME
      FUEL=WTS(20)/6.4
```

```
      XMG=RANGE/FUEL
      XSMG=XMG*GX(19)
      CST(5)=XSMG
***************************
*     OUTPUT SECTION      *
***************************
      IF(OUTPUT.EQ.0) GO TO 100
      WRITE(6,52) WTO,WTBEG,WTMID,WTGRD,WTRES,WTEND,WTFUEL
   52 FORMAT(*1*//40X,26H***FUEL WEIGHT ANALYSIS***///10X*TAKE-OFF*T35,
     $F12.2/10X*START-CRUISE*T35,F12.2/10X*MID-CRUISE*T35,F12.2/
     $10X*END-CRUISE*T35,F12.2/10X*AFTER RESERVE*T35,F12.2/10X
     $*AFTER DESCENT/TAXI*T35,F12.2//5X*NET FUEL WEIGHT(LBS)*T35,F12.2)
      WRITE(6,54) ALT11,ALT2,ALTR
   54 FORMAT(////5X*CRUISE ALTITUDES*/10X*LEG 1*T35,F12.2/10X*LEG 2*
     $T35,F12.2/10X*RESERVE LEG*T35,F12.2)
      WRITE(6,64) FL,SPEED,BLKHR,BLKSPD
   64 FORMAT(///10X*FLIGHT LENGTH (HR)*T35,F12.2/
     $ 10X*AVERAGE SPEED (KTS)*T35,F12.2/
     $ 10X*BLOCK TIME (HR)*T35,F12.2/
     $ 10X*BLOCK SPEED (KTS)*T35,F12.2)
      WRITE(6,66) WTS(20),FUEL,XMG,XSMG
   66 FORMAT(10X*BLOCK FUEL (LBS)*T35,F12.2/
     $10X*BLOCK FUEL (GALS)*T35,F12.2/10X
     $*NAUT. MI/GAL*T35,F12.2/10X*NAUT. SEAT MI./GAL.*
     $T35,F12.2)
      TENG=DESIGN(6)/ITERM(4)
      TREF=GX(21)
      SCF=TENG/TREF
      WRITE(6,68) DESIGN(6),ITERM(4),TENG,TREF,SCF
   68 FORMAT(/10X*INSTALLED THRUST (LBS)*T35,F12.2/
     $ 10X*NO. OF ENGINES*T35,I12/10X
     $ *ENGINE THRUST (LBS)*T35,F12.2/10X
     $ *REFERENCE ENGINE (LBS)*T35,F12.2/10X
     $ *SCALE FACTOR*T35,F12.3)
  100 RETURN
      END
      SUBROUTINE CRUFUEL(WTBEG,RANGE,WTEND,ALTCR,ALTEND,ICOUNT,TIME
     $ ,OUTPUT,IRES)
******************************************************************************
*     CRUFUEL CALCULATES PERFORMANCE DURING CRUISE/CLIMB OR RESERVE          *
*     SEGMENT                                                                *
******************************************************************************
      REAL ANS(4),LOD,HZ,M
      INTEGER OUTPUT
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /DRAG/CDS(6),CDSAP(6)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      COMMON /WTSVE/WTS(20)
      M=GX(3)
      GAMA=0.
      CALL CDZL(ALTCR,CDO,M,0)
      CDO=CDO+GX(31)
      XT1=W(8)*W(8)*(1.0/W(6)-1.0/W(7))/(GX(1)*DESIGN(2))
```

```
      CLM=1.00*(GX(1)*DESIGN(2)*W(7)*(CDO+XT1))**0.5
      IF(IRES.LT.1) GO TO 10
      CLCR=0.79*CLM
************************************************************************
*     THE FOLLOWING STATEMENTS CALCULATE THE RESERVE MISSION WHICH     *
*     IS AT 30000 FT AT MAXIMUM RANGE.                                 *
************************************************************************
      CALL AT62(30000.,ANS)
      VCR=(WTBEG*2.0/(DESIGN(1)*ANS(1)*CLCR))**0.5
      M=VCR/ANS(4)
      CALL XLOD(CLCR,M,30000.,LOD,1,0)
      HZ=3.0
      ALTEND=30000.
      TIME=RANGE/(VCR*1.467*1.1507)
      GO TO 88
************************************************************************
*     THE FOLLOWING STATEMENTS CALCULATE THE CRUISE/CLIMB AT MAXIMUM   *
*     RANGE FACTOR.                                                    *
************************************************************************
   10 CALL AT62(ALTCR,ANS)
      VCR=M*ANS(4)
      WTT=.98*WTBEG
      Q=0.5*ANS(1)*VCR*VCR
      CLCR=WTBEG/(Q*DESIGN(1))
      IF(CLCR.GT.0.79*CLM) GO TO 15
      CLCR=0.79*CLM
      VCR=(WTBEG*2./(ANS(1)*DESIGN(1)*CLCR))**0.5
      M=VCR/ANS(4)
      CRCL=.79*CLM
      TIME=RANGE/(VCR/(1.467*1.1507))
      CALL CRUALT(WTT,CRCL,GX(3),ALTD)
      ROC=(ALTD-ALTCR)/(3600.*(TIME-.02))
      GAMA=ASIN(ROC/VCR)
      GO TO 20
*************************************************************************
*     THE FOLLOWING STATEMENTS CALCULATE THE CRUISE/CLIMB AT 90% OF     *
*     CL FOR (L/D)MAX.                                                  *
*************************************************************************
   15 IF(CLCR.LT.0.98*CLM) GO TO 18                                     *
      CRCL=0.9*CLM                                                      *
      CALL CRUALT(WTT,CRCL,GX(3),ALTD)                                  *
      TIME=RANGE/(VCR/(1.467*1.1507))                                   *
      ROC=(ALTD-ALTCR)/(3600.*(TIME-.02))                               *
      GAMA=ASIN(ROC/VCR)                                                *
      GO TO 20                                                          *
*************************************************************************
   18 IF(ICOUNT.LT.6) GO TO 20
      CRCL=.9*CLM
      CALL CRUALT(WTT,CRCL,GX(3),ALTD)
      ROC=(ALTD-ALTCR)/(3600.*(TIME+0.02))
      GAMA=ASIN(ROC/VCR)
   20 CALL XLOD(CLCR,M,ALTCR,LOD,1,0)
      HZ=ALTCR/10000.
```

```
   88 CALL ENGINE(ALTCR,M,TCTM,TSFC)
      WTEND=WTBEG/(EXP(TSFC*RANGE/(VCR*0.59239*LOD)))
      TIME=RANGE/(VCR/(1.467*1.1507))
      TOWAV=DESIGN(6)/WTBEG
      TOWRQ=(1./TCTM)*(1./LOD+SIN(GAMA))
      TCON=TOWAV/TOWRQ
      IF(TCON.LT.GX(30)) GX(30)=TCON
      IF(ICOUNT.GT.10) GO TO 92
      CALL CRUALT(WTEND,CLCR,M,ALTEND)
      ALTEND=ALTEND+SIN(GAMA)*VCR*3600.*TIME
   92 IF(ICOUNT.NE.5) GO TO 99
*********************************
*     DATA BASE ASSIGNMENTS     *
*********************************
      CDS(4)=LOD
      CDS(5)=CLCR/LOD
      GX(9)=CLCR
      CALL CRUALT(WTEND,CLM,GX(3),ALTLDM)
      WTS(12)=ALTLDM
   99 IF(OUTPUT.LT.1) GO TO 100
**************************
*     OUTPUT SECTION     *
**************************
      WRITE(6,188) ICOUNT,M,CLCR,VCR,TIME,LOD,TSFC,TCTM,
     $ ALTCR,ALTEND,WTBEG,WTEND,RANGE,TCON,GAMA,CLM
  188 FORMAT(I5,F8.2,F7.3,F8.1,F8.3,F8.2,F7.3,F7.3,5F10.1,
     $ F8.3,F8.4,F8.3)
  100 RETURN
      END
      SUBROUTINE XLOD(CL,XM,ALT,LOD,IPHASE,OUTPUT)
      REAL LOD,ANS(4),A1(5),A2(4),A3(5),B1(4)
      INTEGER OUTPUT
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /DRAG/CDS(6),CDSAP(6)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      COMMON /GRAVITY/CG(6)
      COMMON /STAB/DERIVCR(15),DERIVAP(15),STOR(20)
      COMMON /WTSVE/WTS(20)
************************************************************************
*     IPHASE=1 (CR), =0 (10 DEG FLP,FWD CG), =-1 (45 DEG FLP),        *
*     =-2 (45 DEG FLAP, FWD CG), =-3 (45 DEG FLAP, FWD CG, FIND       *
*     CLMAX), =-4 (10 DEG FLAP, FWD CG, FIND CLMAX), =2 (CR, AFT CG) , *
************************************************************************
      DATA A1/.00102,.028817,.841,5.714076,10.706253/
      DATA A2/1.00131,-.122063,.030714,-.005556/
      DATA A3/276.559019,-1306.362579,2314.351122,-1817.552645,
     $ 534.505085/
      DATA B1/-.794,-.296,.812,.111/
      IF(IPHASE.LT.1) XM=.15
      ICG=ITERM(3)
      IF(IPHASE.GT.1) ICG=1
      IF(IPHASE.GT.0) GO TO 25
      CALL AT62(ALT,ANS)
```

```
      F1=GX(27)
      IF(IPHASE.EQ.0.OR.IPHASE.LT.-3) F1=1.
      V=(2.*F1*WTS(1)/(ANS(1)*DESIGN(1)*CL))**0.5
      XM=V/ANS(4)
      ICG=4
      IF(IPHASE.EQ.-1.OR.IPHASE.EQ.0) ICG=2
      ICG=2
      IF(IPHASE.LT.-1) ICG=4
   25 CALL CDZL(ALT,CDO,XM,OUTPUT)
      CALL STABCOD(CL,XM,ICG,OUTPUT)
      CALL TRIM(CL,XM,CDO,CLWING,CLTAIL,CRIT,CRDE,EPS,IPHASE,
     $ICG)
************************************
*     COMPRESSIBILITY DRAG         *
************************************
      DFME=0.
      FMCR=0.
      DWME=0.
      WMCR=0.
      IF(XM.LT.0.65) GO TO 40
      CLWN=CLWING/(W(15)*W(15))
      WMCRN=(W(4)-(B1(4)*CLWN+B1(3)))/(B1(2)*CLWN+B1(1))
      WMCR=WMCRN/W(15)
      DM=XM-WMCR
      Z1=A1(1)
      DO 28 I=2,5
      Z1=Z1+A1(I)*DM**(I-1)
   28 CONTINUE
      DWME=Z1
      BARL=DESIGN(3)/(GX(5)*10.)
      Z1=A2(1)
      DO 30 I=2,4
      Z1=Z1+A2(I)*BARL**(I-1)
   30 CONTINUE
      FMCR=Z1
      Z1=A3(1)
      DM=XM-FMCR
      BARM=0.89+DM
      DO 32 I=2,5
      Z1=Z1+A3(I)*BARM**(I-1)
   32 CONTINUE
      DFME=(Z1-1.)*CDS(6)
   40 E1=W(6)
      E2=W(7)
      IF(IPHASE.LT.1) E2=0.7
********************************
*     WING INDUCED DRAG        *
********************************
      CLO=W(8)
      T1=CLO*CLO/(GX(1)*DESIGN(2)*E1)
      T2=(CLWING*CLWING-CLO*CLO)/(GX(1)*DESIGN(2)*E2)
      TT1=T1+T2
****************************************
```

```
*     MUNK'S INTERFERENCE TERM          *
*****************************************
      SMU=HX(4)/W(9)
      G=(.25*GX(5)+GX(34))/W(9)
      G2=G*G
      G3=G2*G
      C0=.000076+.006814*G-.088417*G2+.247037*G3
      C1=1.002161+2.242040*G-34.140971*G2+73.096667*G3
      C2=.000145-24.824801*G+211.181316*G2-442.515185*G3
      C3=-.014537+42.231817*G-375.564896*G2+803.7551111*G3
      C4=.009817-24.947988*G+220.010784*G2-472.608148*G3
      SIGMA=C0+C1*SMU+C2*SMU*SMU+C3*SMU*SMU*SMU+C4*SMU*SMU*SMU*SMU
*******************************
*     INTERFERENCE DRAG       *
*******************************
      T3=2.*SIGMA*CLWING*CLTAIL*DESIGN(4)/(SMU*GX(1)*DESIGN(2)
     $ *DESIGN(1)*E1)
*******************************
*     TAIL INDUCED DRAG       *
*******************************
      T4=DESIGN(4)*CLTAIL*CLTAIL/(DESIGN(1)*GX(1)*DESIGN(5)*.8)
********************************************
*     INDUCED DRAG FROM HAVING TAIL        *
********************************************
      DELCL2=CLWING*CLWING-CL*CL
      TDG=DELCL2/(GX(1)*DESIGN(2)*E2)+T3+T4
      CDI=TT1+T3+T4
      CDDEL=(ABS(CRDE))*DERIVCR(14)
      IF(IPHASE.LT.1) CDDEL=(ABS(CRDE)+GX(2)*GX(13))*DERIVAP(14)
      CDTOT=CDI+CDO+CDDEL+DWME+DFME
      XT1=CLO*CLO*(1./E1-1./E2)/(GX(1)*DESIGN(2))
      CDOP=CDO+TDG
***********************************
*     DATA BASE ASSIGNMENTS       *
***********************************
      GX(31)=TDG
      XLDMX=.5*((GX(1)*DESIGN(2)*E2)/(CDOP+XT1))**0.5
      CLLDMX=(GX(1)*DESIGN(2)*W(7)*(CDOP+XT1))**0.5
      IF(IPHASE.LT.1) GO TO 48
      HX(14)=CRDE
      CST(6)=XLDMX
      GX(28)=CLWING
      DTAIL=CDS(3)+TDG
      GO TO 50
   48 CDSAP(4)=CL/CDTOT
      DTAIL=CDSAP(3)+TDG
      CDSAP(5)=CDTOT
      HX(11)=CLTAIL
      GX(29)=CLWING
      HX(12)=CRIT
      HX(13)=CRDE
   50 LOD=CL/CDTOT
*************************
```

```
*     OUTPUT SECTION     *
**************************
      IF(OUTPUT.EQ.0) GO TO 100
      WRITE(6,80)
   80 FORMAT(*1*//20X*ACCURATE L/D ANALYSIS*)
      IF(IPHASE.LT.0) WRITE(6,81)
   81 FORMAT(//20X*APPROACH WITH 45 DEGREES FLAP*)
      IF(IPHASE.LT.-1) WRITE(6,87)
   87 FORMAT(20X*FORWARD C.G.*/)
      WRITE(6,82)CL,CLWING,CLTAIL,CRDE,CRIT
   82 FORMAT(//10X*CL(REQUESTED)=*T45,F15.3/10X*CL(WING)=*T45,F15.3/10X
     $*CL(TAIL)=*T45,F15.3/10X*ELEVATOR(DEGREES)=*T45,F15.2/
     $10X*STABILIZER(DEGREES)*T45,F15.2/)
      WRITE(6,83) TT1,T3,T4,TDG,DTAIL,SIGMA
   83 FORMAT(/5X*INDUCED DRAG COMPONENTS*/10X*WING=*T45,F15.4/
     $10X*INTERFERENCE=*T45,F15.4/10X*TAIL=*T45,F15.4
     $ /10X*(TRIM)=*T45,F15.4/10X*(TAIL)=*T45,F15.4/10X*SIGMA*T45,
     $ F15.4/)
      WRITE(6,84) CDI,CDO,CDDEL,DWME,WMCR,DFME,FMCR,CDTOT
   84 FORMAT(5X*DRAG COEFFICIENTS*/10X*INDUCED=*T45,F15.4/
     $ 10X*ZERO LIFT=*
     $T45,F15.4/10X*ELEVATOR=*T45,F15.4/10X
     $ *WING(MACH), M(CRIT)=*T45,F15.4,F15.2/10X
     $ *FUSE(MACH), M(CRIT)=*T45,F15.4,F15.2/10X
     $ *TOTAL=*T45,F15.4/)
      WRITE(6,86) LOD
   86 FORMAT(//15X*L/D=*T45,F15.3)
      WRITE(6,92) XLDMX,CLLDMX,CDOP
   92 FORMAT(///15X*MAX. L/D=*T45,F15.3/15X*CL-L/D MAX=*T45,F15.3/
     $ 10X*MODIFIED CDO=*T45,F15.4/)
  100 RETURN
      END
      SUBROUTINE AIRCOST(COST,OUTPUT)
      REAL CD(10),CP(10)
      REAL T1(2),T3(2),T9(2),T10(2),T32(2)
      INTEGER OUTPUT
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /WTSVE/WTS(20)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      A=WTS(4)
      DATA T1,T3,T9,T10/1.,1.15,1.,2.,1.,1.15,0.,1./
      DATA T32/1.,1.1/
*****************************************************
*     LIST OF PURCHASE PRICE ELEMENTS IN 1974 $     *
*           REF(4) -- NICOLAI                       *
*         ENGINEERING (1)                           *
*         DEVELOPMENT SUPPORT (2)                   *
*         FLIGHT TEST (3)                           *
*         TOOLING (4)                               *
*         MANUFACTURING LABOR (5)                   *
*         QUALITY CONTROL (6)                       *
*         MATERIALS (7)                             *
*         ENGINE (8)                                *
```

```
*         AVIONICS (9)                                       *
*         ACTIVE CONTROL SYSTEM (10)                         *
*************************************************************
      IACT=0
      ICGX=ITERM(2)
      CD(1)=T1(IACT+1)*10964.13*A**0.791
      CP(1)=T1(IACT+1)*26567.12*A**0.791-CD(1)
      CD(2)=1627.68*A**0.873
      CP(2)=0.0
      CD(3)=T3(IACT+1)*T32(ICGX+1)*19.16*A**1.160
      CP(3)=T3(IACT+1)*T32(ICGX+1)*0.0
      CD(4)=15716.31*A**0.764
      CP(4)=43272.95*A**0.764-CD(4)
      CD(5)=13026.56*A**0.74
      CP(5)=164219.18*A**0.74-CD(5)
      CD(6)=CD(5)*0.13
      CP(6)=CP(5)*0.13
      CD(7)=2733.64*A**0.689
      CP(7)=125960.83*A**0.689-CD(7)
      CD(8)=(ITERM(4)+1)*2.*1.31*169.0*(DESIGN(6)/ITERM(4))**0.8356
      CP(8)=ITERM(4)*250.*1.31*169.0*(DESIGN(6)/ITERM(4))**0.8356
      CD(9)=2.*300000.*T9(IACT+1)
      CP(9)=250.*300000.*T9(IACT+1)
C
C     ACTIVE CONTROLS PRICE HAS TO BE ESTIMATED
C
      CD(10)=206250.*2.*ITERM(1)
      CP(10)=206250.*250.*ITERM(1)
      TOTD=0.
      TOTP=0.
C
C     CONVERT FROM 1974 $ TO 1976 $
C
      DO 50 J=1,10
      CD(J)=CD(J)*1.23077
      CP(J)=CP(J)*1.23077
      TOTD=TOTD+CD(J)
      TOTP=TOTP+CP(J)
   50 CONTINUE
C
C     INCLUDING 10% PERCENT PROFIT
C
      TOTCOST=TOTD*1.1+TOTP*1.1
      COST=TOTCOST/250.0
      COST=COST*(1.0+PX(7)*0.05)
***************************
*     OUTPUT SECTION      *
***************************
      IF(OUTPUT.EQ.0) GO TO 100
      WRITE(6,70)
   70 FORMAT(*1*//20X*AIRCRAFT COST ESTIMATES*//43X*DEVELOPMENT*
     $9X*PRODUCTION*/)
      WRITE(6,72) CD(1),CP(1)
```

```
   72 FORMAT(10X*ENGINEERING*T40,F15.2,T60,F15.2)
      WRITE(6,74) CD(2),CP(2)
   74 FORMAT(10X*DEVELOPMENT SUPPORT*T40,F15.2,T60,F15.2)
      WRITE(6,80) CD(3),CP(3)
   80 FORMAT(10X*FLIGHT TEST*T40,F15.2,T60,F15.2)
      WRITE(6,82) CD(4),CP(4)
   82 FORMAT(10X*TOOLING*T40,F15.2,T60,F15.2)
      WRITE(6,84) CD(5),CP(5)
   84 FORMAT(10X*MANUFAC. LABOR*T40,F15.2,T60,F15.2)
      WRITE(6,86) CD(6),CP(6)
   86 FORMAT(10X*QUALITY CONTROL*T40,F15.2,T60,F15.2)
      WRITE(6,88) CD(7),CP(7)
   88 FORMAT(10X*MATERIALS*T40,F15.2,T60,F15.2)
      WRITE(6,90) CD(8),CP(8)
   90 FORMAT(10X*ENGINE*T40,F15.2,T60,F15.2)
      WRITE(6,92) CD(9),CP(9)
   92 FORMAT(10X*AVIONICS*T40,F15.2,T60,F15.2)
      WRITE(6,94) CD(10),CP(10)
   94 FORMAT(10X*ACTIVE CONTROLS SYSTEM*T40,F15.2,T60,F15.2)
      WRITE(6,96) TOTD,TOTP
   96 FORMAT(/10X*TOTAL*T40,F15.2,T60,F15.2)
      WRITE(6,78) COST
   78 FORMAT(/5X*TOTAL COST PER AIRCRAFT=  $*F12.2)
  100 RETURN
      END
      SUBROUTINE MAINCST(COST,OUTPUT)
      INTEGER OUTPUT
      REAL MCOST(27),LCOST(27),T9(2),XNM(27)
      REAL T1(2),T3(2),T8(2),T12(2),T15(2)
      REAL LCST(27),MCST(27)
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /WTSVE/WTS(20)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      DATA T1,T3,T8/1.,1.15,1.,1.2,1.,1.2/
      DATA T9,T12,T15/1.,1.3,1.,1.15,1.,1.15/
      IACT=ITERM(1)
      ICGX=ITERM(2)
      NENG=ITERM(4)
      NPASS=GX(19)
      WTO=WTS(1)*0.453592
      WTE=WTS(2)*0.453592
      WTF=WTS(6)*0.456592
      DATA XNM/4HINSP,8HAIR COND,10HAUTO PILOT,6HCOMMUN,
     $4HELEC,4HFURN,9HFIRE PROT,9HFLT CONTL,4HFUEL,
     $9HHYD POWER,3HICE,5HINSTR,9HLAND GEAR,
     $8HLIGHTING,5HNAVIG,6HOXYGEN,7HPNUEMAT,
     $9HWAT/WASTE,7HAIR APU,9HSTRUCTURE,5HDOORS,
     $8HFUSELAGE,8HNACELLES,5HWINGS,4HSTAB,
     $7HWINDOWS,6HENGINE/
*************************************************
*      MAINTENANCE COSTS--1976 DOLLARS/HOUR       *
*          REF(20) -- AMERICAN AIRLINES            *
*         (1)    INSPECTION AND MISC.             *
```

```
*          (2)    AIR CONDITIONING                  *
*          (3)    AUTO PILOT                        *
*          (4)    COMMUNICATIONS                    *
*          (5)    ELECTRICAL                        *
*          (6)    EQUIPMENT AND FURNISHINGS         *
*          (7)    FIRE PROTECTION                   *
*          (8)    FLIGHT CONTROLS                   *
*          (9)    FUEL                              *
*          (10)   HYDRAULIC POWER                   *
*          (11)   ICE AND RAIN                      *
*          (12)   INSTRUMENTS                       *
*          (13)   LANDING GEAR                      *
*          (14)   LIGHTING                          *
*          (15)   NAVIGATION                        *
*          (16)   OXYGEN                            *
*          (17)   PNUEMATICS                        *
*          (18)   WATER/WASTE                       *
*          (19)   AIRBORNE APU                      *
*          (20)   STRUCTURE                         *
*          (21)   DOORS                             *
*          (22)   FUSELAGE                          *
*          (23)   NACELLES/PYLONS                   *
*          (24)   WINGS                             *
*          (25)   STABILIZERS                       *
*          (26)   WINDOWS                           *
************************************************
      LCOST(1)=T1(IACT+1)*7.66+0.377*WTE/1000.0
      MCOST(1)=T1(IACT+1)*1.21+0.062*WTE/1000.0
      DATA LCOST(2),MCOST(2)/5.1026,4.52/
      LCOST(3)=T3(IACT+1)*11.19
      MCOST(3)=T3(IACT+1)*2.621
      LCOST(4)=.0276*NPASS
      MCOST(4)=0.0118*NPASS
      DATA LCOST(5),MCOST(5)/4.306,5.748/
      LCOST(6)=9.11+0.08496*NPASS
      MCOST(6)=2.38+0.05776*NPASS
      LCOST(7)=.213+2.29*(2.+NENG)
      MCOST(7)=0.365*(2.+NENG)
      LCOST(8)=T8(IACT+1)*6.84+0.0035*WTO/1000.
      MCOST(8)=T8(IACT+1)*3.876+0.00655*WTO/1000.
      LCOST(9)=1.114+0.0262*WTF*T9(ICGX+1)/1000.0
      MCOST(9)=0.595+0.0123*WTF*T9(ICGX+1)/1000.0
      DATA LCOST(10),MCOST(10) /3.33,3.95/
      LCOST(11)=.5089+0.0013*WTO/1000.0
      MCOST(11)=.0847+.0037*WTO/1000.0
      LCOST(12)=T12(IACT+1)*0.509+.009*WTE/1000.0
      MCOST(12)=T12(IACT+1)*0.235+.0031*WTE/1000.0
      LCOST(13)=4.58+.071*WTO/1000.0
      MCOST(13)=4.961+.181*WTO/1000.0
      LCOST(14)=1.51+0.01152*NPASS
      MCOST(14)=0.047+0.01392*NPASS
      LCOST(15)=T15(IACT+1)*10.077
      MCOST(15)=T15(IACT+1)*7.166
```

```
      LCOST(16)=.515+0.00265*NPASS
      MCOST(16)=0.00752*NPASS
      DATA AC/200./
      T=DESIGN(6)*4.448/NENG
      LCOST(17)=0.181+.0003*AC*T/10000.
      MCOST(17)=0.0019*AC*T/10000.
      LCOST(18)=.339+0.00368*NPASS
      MCOST(18)=0.00768*NPASS
      DATA LCOST(19),MCOST(19)/.315,.462/
      LCOST(20)=3.+.0099*WTE/1000.
      DATA MCOST(20)/0./
      LCOST(21)=1.147+0.006*NPASS
      MCOST(21)=.387+0.00785*NPASS
      LCOST(22)=1.5+0.046*WTE/1000.
      DATA MCOST(22)/0.5833/
      NAC=4
      IF(NENG.LT.4) NAC=2
      LCOST(23)=.3366*NAC
      MCOST(23)=.1391*NAC
      SW=DESIGN(1)/(3.281*3.281)
      DATA LCOST(24)/2.9475/
      MCOST(24)=0.126+0.00506*SW
      DATA LCOST(25),MCOST(25)/0.8321,0.3737/
      LCOST(26)=0.763+0.00043*NPASS
      MCOST(26)=0.0362*NPASS
      DO 50 K=1,26
      LCST(K)=LCOST(K)/2.5
      MCST(K)=MCOST(K)/2.5
   50 CONTINUE
      TLCOST=0.
      TMCOST=0.
      DO 75 K=1,26
      TLCOST=LCST(K)+TLCOST
      TMCOST=MCST(K)+TMCOST
   75 CONTINUE
      TENG=DESIGN(6)/ITERM(4)
*******************************************************************************
*     MAIN ENGINE COST                                                        *
*******************************************************************************
      LCOST(27)=(ITERM(4)/(4.0*2.5))*88.5*(TENG/20000.)**0.5                  *
      MCOST(27)=(ITERM(4)/(4.0*2.5))*109.0*(TENG/20000.)**0.5                 *
      COST=TLCOST+TMCOST+LCOST(27)+MCOST(27)                                  *
      COST=COST*(1.0+PX(6)*0.05)                                              *
*******************************************************************************
      IF(OUTPUT.EQ.0) GO TO 100
      WRITE(6,86)
   86 FORMAT(*1*//30X*MAINTENANCE OPERATING COSTS*//7X*NO.  SYSTEM*7X
     $10X*LABOR*7X*MATERIAL*)
      DO 90 K=1,26
      WRITE(6,88) K,XNM(K),LCOST(K),MCOST(K)
   88 FORMAT(I10,2X,A10,3X,2F15.2)
   90 CONTINUE
      WRITE(6,92)TLCOST,TMCOST,LCOST(27),MCOST(27),COST
```

```
   92 FORMAT(//5X*LABOR COST*T35,F15.2/5X*MATERIAL COST*T35,F15.2//5X
     $*ENGINE LABOR COST*T35,F15.2/5X*ENGINE MATERIAL COST*T35,F15.2
     $///30X*MAINTNENACE DOC IN 1976 DOLLARS PER HOUR*F15.2)
  100 RETURN
      END
      SUBROUTINE CNSTRN(OUTPUT)
      REAL LODMA,LOD2,ANS(4),LFL,NZOACR,NZOAA
      REAL PRAMCR(4),PRAMAP(4),ACR(5),AAP(5)
      REAL TCCR(4),TCAP(4)
      COMPLEX ROOTCR(4),ROOTAP(4)
      COMMON /STRAIN/CON(59)
      INTEGER OUTPUT
      REAL KT1,KT2,KTDOT1,KTDOT2,DERCR(15),DERAP(15)
      COMMON /CONSTR/SU(59),SL(59),XINEQ(59)
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /DRAG/CDS(6),CDSAP(6)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      COMMON /GRAVITY/CG(6)
      COMMON /STAB/DERIVCR(15),DERIVAP(15),STOR(20)
      COMMON /WTSVE/WTS(20)
*************************************************************************
*      CONSTRAINT IDENTIFICATION                                        *
*                                                                       *
*      NO.           DESCRIPTION                                        *
*   *******        *****************************************************  *
*       1 ..........CRUISE THRUST REQUIREMENT                           *
*       2 ..........SECOND SEGMENT CLIMB GRADIENT - THRUST REQUIREMENT  *
*       3 ..........MISSED APPROACH CLIMB GRADIENT - THRUST REQUIREMENT *
*       4 ..........LANDING FIELD LENGTH - WING LOADING REQUIREMENT     *
*       5 ..........TAKE-OFF FIELD LENGTH - WING LOADING REQUIREMENT    *
*       6 ..........LANDING GEAR - AFT CG LIMIT                         *
*     7,8 ..........STATIC STABILITY - CRUISE, APPROACH                 *
*     9,10..........MANUEVER MARGIN - CRUISE, APPROACH                  *
*      11 ..........TAIL LIFT - APPROACH                                *
*      12 ..........NOSE GEAR UNSTICK                                   *
*   13,14...........DYNAMIC STABILITY - CRUISE,APPROACH                 *
*   15,16...........PHUGOID FREQUENCY - CRUISE, APPROACH                *
*   17,18...........PHUGOID DAMPING - CRUISE, APPROACH                  *
*   19,20...........SHORT PERIOD FREQUENCY - CRUISE,APPROACH            *
*   21,22...........SHORT PERIOD DAMPING - CRUISE,APPROACH              *
*   23,24...........TIME-TO-DOUBLE (CRUISE, APPROACH)                   *
*   25,26...........TIME-TO-HALF (CRUISE, APPROACH)                     *
*      27 ..........FLIGHT PATH STABILITY - APPROACH                    *
*   28,29...........VERTICAL GAIN - CRUISE, APPROACH                    *
*   30,31...........TIME SUB THETA 2 - CRUISE, APPROACH                 *
*   32,33...........FREQUENCY**2/VERTICAL GAIN - CRUISE,APPROACH        *
*      34 ..........T(1) PARAMETER - APPROACH                           *
*   35,36...........RATIO OF MODE FREQUENCIES - CRUISE, APPROACH        *
*   37-39...........ELEVATOR VARIANCE - CRUISE                          *
*   40-42...........ELEVATOR VARIANCE - APPROACH                        *
*   43,44...........VARIANCE OF ELEVATOR RATE - CRUISE, APPROACH        *
*      45 ..........PASSENGER VOLUME LIMIT                              *
*   46,47...........ELEVATOR DEFLECTIONS - TRIMMED (CRUISE, APPROACH)   *
```

```
*     48 ..........CRUISE ALTITUDE                                                 *
*     49 ..........CRUISE ALTITUDE  ((L/D)MAX)                                     *
*   50,51..........WING CL - CRUISE, APPROACH                                      *
*     52 ..........TAIL ASPECT RATIO LIMIT  -  AR(TAIL/AR(WING)                    *
************************************************************************************
      IOUT=OUTPUT
      IF(OUTPUT.GT.1) IOUT=0
      CON(1)=GX(30)
**************************************************************************
*     FIND TAKE-OFF CL-MAX AND SECOND SEGMENT CLIMB GRADIENTS            *
**************************************************************************
      CLWTO=GX(8)
      CALL XLOD(CLWTO,.1,0.,EFF,-4,0)
      NENG=ITERM(4)
      CALL AT62(0.,ANS)
      TOWAV=DESIGN(6)/WTS(1)
      CL2=CLWTO/1.44
      V=(2.*WTS(1)/(DESIGN(1)*ANS(1)*CL2))**0.5
      XM=V/ANS(4)
      CALL ENGINE(0.,XM,TCTM,TSFC)
      GRAD=0.030
      CALL XLOD(CL2,.1,0.,LOD2,0,0)
      IF(ITERM(4).EQ.3) GRAD=0.027
      IF(ITERM(4).EQ.2) GRAD=0.024
      TOWRQ2=(NENG/(NENG-1.))*(1./LOD2+SIN(GRAD))*(1./TCTM)
      CON(2)=TOWAV/TOWRQ2
*****************************
*     NOSE GEAR UNSTICK     *
*****************************
      ZACT=4.0/W(10)
      XMUF=0.025
      ZACLG=2.*ZACT
      XLT=HX(6)*DESIGN(1)*W(10)/DESIGN(4)
      CALL AT62(0.,ANS)
      VSTALL=(WTS(1)*2.0/(DESIGN(1)*ANS(1)*GX(8)))**0.5
      VLO=VSTALL*0.9
      CLW=(STOR(6)*(W(2)-GX(17)-3.0))/GX(2)
      Q=0.5*ANS(1)*VLO*VLO
      XLW=Q*DESIGN(1)*CLW*1.2
      XMW=1.05*Q*DESIGN(1)*(W(19)+W(17))*W(10)
      C1=XLT+(0.25-CG(5)-XMUF*ZACLG)*W(10)
      UNTHRU=DESIGN(6)
      IF(ITERM(4).EQ.3) UNTHRU=DESIGN(6)/3.0
      XCG=CG(3)
      COIT=HX(12)
C
C     TAIL INCIDENCE SET TO WORSE CASE FOR TAKE-OFF (BOEING)
C
      IF(GX(16).NE.-99.0) COIT=GX(16)
      XLTRQ=(XMW+ZACT*W(10)*.9*UNTHRU+W(10)*(XCG-.25)*XLW
     $ -W(10)*(CG(5)-CG(3)+ZACLG*XMUF)*(WTS(1)-XLW))/C1
      DEMAX=20.
      XLTAV=STOR(7)*HX(6)*.9*((1.-STOR(8))*(W(2)-GX(17)-3.0)
```

```
     $-W(2)+COIT-HX(3)*DEMAX)*Q*DESIGN(4)/GX(2)
C
C     GROUND EFFECTS--DATCOM FUNCTION OF GEOMETRIC ALPHA
C
      TLGE=.355*Q*DESIGN(4)
      XLTAV2=XLTAV+TLGE
      CLTAV=XLTAV2/(Q*DESIGN(4))
      IF(CLTAV.GT.1.5) XLTAV2=-1.5*Q*DESIGN(4)
      CON(12)=XLTAV2/XLTRQ
      IF(OUTPUT.EQ.1) WRITE(6,548) ZACT,XMUF,ZACLG,ANS,VSTALL,CLW
     $ ,Q,XLW,XMW,C1,UNTHRU,XCG,XLTRQ,CL2,COIT,XLT
     $ ,XLTAV,CON(12),STOR(7),STOR(8),W(2),GX(17),HX(6),W(10)
     $ ,XLTAV2,TLGE,CLTAV
  548 FORMAT(*1*//20X*DEBUG OF NOSE GEAR UNSTICK*///
     $10X"ZACT,XMUF,ZACLG="3F15.4/
     $ 10X"ANS(1,2,3,4)=" 4F15.6/
     $ 10X"VSTALL,CLW,Q,XLW,XMW="5F15.4/
     $ 10X"C1,UNTHRU,XCG,XLTRQ,CL2,COIT,XLT="7F12.4/
     $ 10X"XLTAV,CON(12),STOR(7),STOR(8),W(2)="5F15.4/
     $ 10X"GX(17),HX(6),W(10),XLTAV2,TLGE,CLTAV="6F15.4/)
*************************************************************************
*     FIND APPROACH CL-MAX AND MISSED APPROACH CLIMB GRADIENTS          *
*       SET UP APPROACH WITH FORWARD CG AND 45 DEG FLAPS                *
*************************************************************************
      CLWMAX=GX(12)
      CALL XLOD(CLWMAX,.1,0.,EFF,-3,0)
      CLS=CLWMAX
      GRAD=GRAD-.003
      CLA=CLS/1.69
      V=(2.*WTS(1)*GX(27)/(DESIGN(1)*ANS(1)*CLA))**0.5
      XM=V/ANS(4)
      CALL ENGINE(0.,XM,TCTM,TSFC)
      GX(10)=CLA
      CALL XLOD(CLA,.1,0.,LODMA,-2,0)
      TOWRQ3=(NENG/(NENG-1.))*(1./LODMA+SIN(GRAD))*(1./TCTM)
      TOWAV3=DESIGN(6)/(WTS(1)*GX(27))
      CON(3)=TOWAV3/TOWRQ3
      VA2=GX(27)*WTS(1)*498.23/(DESIGN(1)*CLA)
      VA=VA2**0.5
      GX(14)=VA*1.6881
      LFL=0.29875*VA2+25.
      CON(4)=LFL
      TOWAV=DESIGN(6)/WTS(1)
      TOP=(WTS(1)/DESIGN(1))/(CLWTO*TOWAV)
      TOFL=(31.7*TOP)+910.0
      CON(5)=TOFL
      CON(6)=CG(5)-CG(6)-DESIGN(7)
      VOLPAS=GX(19)*52.
      PASSL=DESIGN(3)-1.2*GX(15)-.5*GX(33)
      VOLAV=GX(1)*GX(5)*GX(5)*PASSL/8.
      CON(45)=VOLAV/VOLPAS
      CLTAIL=HX(11)
**********************************
```

```
*     DATA BASE ASSIGNMENTS       *
***********************************
      CON(11)=CLTAIL
      CON(46)=HX(14)
      CON(47)=HX(13)
      CON(48)=WTS(11)
      CON(49)=WTS(12)
      CON(50)=GX(28)
      CON(51)=GX(29)
      CON(52)=DESIGN(5)/DESIGN(2)
*******************************************************************************
*     THIS ENDS DESIGN CONSTRAINT SECTION, AND BEGINS HANDLING QUALITY        *
*     CONSTRAINT SECTION                                                      *
*******************************************************************************
      CALL XLOD(CLA,.1,500.,EFF,-1,IOUT)
      CALL XLOD(GX(9),GX(3),WTS(11),EFF,2,0)
      CON(7)=CG(1)-STOR(4)
      CON(8)=CG(2)-STOR(9)
      CON(9)=CG(1)-STOR(5)
      CON(10)=CG(2)-STOR(10)
************************************************
*     CALCULATE DIMENSIONAL DERIVATIVES        *
************************************************
      CALL AT62(WTS(11),ANS)
      UOCR=GX(3)*ANS(4)
      CALL DIMDER(UOCR,1,DERCR)
      CALL DIMDER(GX(14),0,DERAP)
      IF(OUTPUT.EQ.1) WRITE(6,28) DERCR,DERAP
   28 FORMAT(*1*//30X*/ / / DIMENSIONAL STABILITY DERIVATIVES/ / / *
     $ //10X*DERCR=*/5(T30,3F12.4/)//10X*DERAP=*/5(T30,3F12.4/))
      CON(13)=DERCR(4)*DERCR(3)-DERCR(1)*DERCR(6)
      CON(14)=DERAP(4)*DERAP(3)-DERAP(1)*DERAP(6)
      IFULL=1
      IF(IFULL.GT.0) GO TO 30
*************************************************************************
*     THIS NEXT SECTION APPROXIMATES AIRCRAFT DYNAMIC PROPERTIES        *
*************************************************************************
      TPHUGCR=DERCR(1)*DERCR(9)-UOCR*DERCR(3)
      OMPH2=GX(15)*(DERCR(3)*DERCR(4)-DERCR(6)*DERCR(1))/TPHUGCR
      OMPH=(ABS(OMPH2))**0.5
      CON(15)=OMPH
      TPHUGAP=DERAP(1)*DERAP(9)-GX(14)*DERAP(3)
      OMPHA2=GX(15)*(DERAP(3)*DERAP(4)-DERAP(6)*DERAP(1))/TPHUGAP
      OMPHA=(ABS(OMPHA2))**0.5
      CON(16)=OMPHA
      TTT=-DERCR(5)-(DERCR(6)*(DERCR(2)*UOCR-GX(15)))/TPHUGCR
      ZETAPH=TTT/(2.0*OMPH)
      CON(17)=ZETAPH
      TTTA=-DERAP(5)-(DERAP(6)*(DERAP(2)*GX(14)-GX(15)))/TPHUGAP
      ZETAPHA=TTTA/(2.0*OMPHA)
      CON(18)=ZETAPHA
      OMSP2=DERCR(9)*DERCR(1)-DERCR(3)*UOCR
      OMSP=(ABS(OMSP2))**0.5
```

```
      CON(19)=OMSP
      OMSPA2=DERAP(9)*DERAP(1)-DERAP(3)*GX(14)
      OMSPA=(ABS(OMSPA2))**0.5
      CON(20)=OMSPA
      TXT=-(DERCR(1)+DERCR(9)+UOCR*DERCR(12))
      ZETASP=TXT/(2.0*OMSP)
      CON(21)=ZETASP
      TXTA=-(DERAP(1)+DERAP(9)+GX(14)*DERAP(12))
      ZETASPA=TXTA/(2.0*OMSPA)
      CON(22)=ZETASPA
************************************************************************
*     CALCULATE EXACT DYNAMICS FROM FOURTH ORDER MODEL                 *
************************************************************************
   30 CALL LONGRT(DERCR,UOCR,ROOTCR,PRAMCR,ACR,NOCR,TCCR)
      CALL LONGRT(DERAP,VA,ROOTAP,PRAMAP,AAP,NOAP,TCAP)
      IF(IFULL.LT.1) GO TO 45
      OMPH=PRAMCR(1)
      ZETAPH=PRAMCR(2)
      OMSP=PRAMCR(3)
      ZETASP=PRAMCR(4)
      OMPHA=PRAMAP(1)
      ZETAPHA=PRAMAP(2)
      OMSPA=PRAMAP(3)
      ZETASPA=PRAMAP(4)
   45 IF(OUTPUT.LT.1) GO TO 50
      IF(OUTPUT.GT.1) GO TO 50
      WRITE(6,31)
   31 FORMAT(*1*///30X*LONGITUDINAL DYNAMICS*//20X*/ / CRUISE/ / *)
      WRITE(6,32) ACR
   32 FORMAT(/10X*COEFFICIENTS=*5F15.6)
      WRITE(6,33) ROOTCR
   33 FORMAT(/10X*ROOTS(REAL,IMAGINARY)*/4(2F15.4/))
      WRITE(6,35) PRAMCR
   35 FORMAT(/10X*PHUGOID FREQUENCY*T35,F15.4/10X
     $*PHUGOID DAMPING*T35,F15.4/10X*SHORT PER. FREQ.*
     $T35,F15.4/10X*SHORT PER. DAMPING*T35,F15.4)
      WRITE(6,36) NOCR,TCCR
   36 FORMAT(/10X*NO. OF NON-OSCILLATORY ROOTS=*I11
     $ /10X*TIME CONSTANTS=*T35,4F15.4)
      WRITE(6,37)
   37 FORMAT(//20X*/ / APPROACH/ / *)
      WRITE(6,32) AAP
      WRITE(6,33) ROOTAP
      WRITE(6,35) PRAMAP
      WRITE(6,36) NOAP,TCAP
      WRITE(6,48) GX(8),CLWTO,CL2,GX(12),CLWMAX,CLA
   48 FORMAT(//10X*CL-MAX TO(W),CL-MAX TO (AC),CL2*T60,3F10.3/
     $ 10X*CL-MAX (W), CL-MAX (AC), CLA=*T60,3F10.3)
   50 IF(IFULL.LT.1) GO TO 100
      CON(15)=OMPH
      CON(16)=OMPHA
      CON(17)=ZETAPH
      CON(18)=ZETAPHA
```

```
      CON(19)=OMSP
      CON(20)=OMSPA
      CON(21)=ZETASP
      CON(22)=ZETASPA
      OMSP2=OMSP*OMSP
      OMSPA2=OMSPA*OMSPA
      IF(OMSP.EQ.0.) OMSP=1.0E-20
      IF(OMSPA.EQ.0.) OMSPA=1.0E-20
      IF(OMPH.EQ.0.) OMPH=1.0E-20
      IF(OMPHA.EQ.0.) OMPHA=1.0E-20
      GREAT=-999.
  100 IF(NOCR.LT.1) GO TO 210
      DO 209 K=1,NOCR
      IF(TCCR(K).LT.0.0) GO TO 209
      IF(TCCR(K).GT.GREAT) GREAT=TCCR(K)
  209 CONTINUE
  210 TDOUBCR=-0.693/(OMPH*ZETAPH)
      T1=-0.693/(OMSP*ZETASP)
      T2=0.693/GREAT
      IF(TDOUBCR.GT.T1.AND.T1.GT.0.0) TDOUBCR=T1
      IF(T2.LT.0.0) GO TO 219
      IF(TDOUBCR.LT.0.0.OR.TDOUBCR.GT.T2) TDOUBCR=T2
  219 IF(TDOUBCR.LT.0.0) TDOUBCR=99.
      CON(23)=TDOUBCR
      GREAT=-999.
      IF(NOAP.LT.1) GO TO 230
      DO 231 K=1,NOAP
      IF(TCAP(K).LT.0.0) GO TO 231
      IF(TCAP(K).GT.GREAT) GREAT=TCAP(K)
  231 CONTINUE
  230 TDOUBAP=-0.693/(OMPHA*ZETAPHA)
      T1=-0.693/(OMSPA*ZETASPA)
      T2=0.693/GREAT
      IF(TDOUBAP.GT.T1.AND.T1.GT.0.0) TDOUBAP=T1
      IF(T2.LT.0.0) GO TO 239
      IF(TDOUBAP.LT.0.0.OR.TDOUBAP.GT.T2) TDOUBAP=T2
  239 IF(TDOUBAP.LE.0.) TDOUBAP=99.
      CON(24)=TDOUBAP
      THALF=-99.
      THALFA=-99.
      IF(OMSP.EQ.1.0E-20) OMSP=-1.0E-20
      IF(OMSPA.EQ.1.0E-20) OMSPA=-1.0E-20
      IF(OMPH.EQ.1.0E-20) OMPH=-1.0E-20
      IF(OMPHA.EQ.1.0E-20) OMPHA=-1.0E-20
      GREAT=-999.
      IF(NOCR.LT.1) GO TO 245
      IF(TDOUBCR.NE.99.) GO TO 250
      DO 242 K=1,NOCR
      IF(TCCR(K).GT.0.0) GO TO 242
      IF(TCCR(K).GT.GREAT) GREAT=TCCR(K)
  242 CONTINUE
  245 THALF=0.693/(OMPH*ZETAPH)
      T1=0.693/(OMSP*ZETASP)
```

```
      T2=-0.693/GREAT
      IF(THALF.LT.T1.AND.T1.GT.0.0)  THALF=T1
      IF(THALF.LT.T2.AND.T1.GT.0.0)  THALF=T2
      IF(THALF.LE.0.) THALF=-99.
  250 CON(25)=THALF
      GREAT=-999.
      IF(NOAP.LT.1) GO TO 255
      IF(TDOUBAP.NE.99.) GO TO 260
      DO 252 K=1,NOAP
      IF(TCAP(K).GT.0.0) GO TO 252
      IF(TCAP(K).GT.GREAT) GREAT=TCAP(K)
  252 CONTINUE
  255 THALFA=0.693/(OMPHA*ZETAPHA)
      T1=0.693/(OMSPA*ZETASPA)
      T2=-0.693/GREAT
      IF(THALFA.LT.T1.AND.T1.GT.0) THALFA=T1
      IF(THALFA.LT.T2.AND.T2.GT.0) THALFA=T2
      IF(THALFA.LE.0.) THALFA=-99.
  260 CON(26)=THALFA
      TDEL=DERAP(13)/DERAP(15)
      TZT=(DERAP(6)*DERAP(4)-DERAP(6)*DERAP(1))/
     $ (-DERAP(1)+DERAP(3)*TDEL)
      TYT=(DERAP(4)-TDEL*DERAP(6))/(DERAP(1)-DERAP(3)*TDEL)
      DGDU=(DERAP(5)-(DERAP(2)-GX(15)/GX(14))*TYT-DERAP(14)*TZT
     $/DERAP(15))/GX(15)
      CON(27)=DGDU
      ZZT=UOCR*(DERCR(13)*DERCR(3)-DERCR(15)*DERCR(1))
      NZOACR=ZZT/((DERCR(15)-DERCR(13)*DERCR(9)/UOCR)*GX(15))
      CON(28)=NZOACR
      ZZTA=GX(14)*(DERAP(13)*DERAP(3)-DERAP(15)*DERAP(1))
      NZOAA=ZZTA/((DERAP(15)-DERAP(13)*DERAP(9)/GX(14))*GX(15))
      CON(29)=NZOAA
      TTH2=ZZT/(UOCR*(DERCR(15)+DERCR(13)*DERCR(12)))
      CON(30)=TTH2
      TTHA2=ZZTA/(GX(14)*(DERAP(15)+DERAP(13)*DERAP(9)))
      CON(31)=TTHA2
      CON(32)=OMSP2/NZOACR
      CON(33)=OMSPA2/NZOAA
      PIARE=1./(GX(1)*DESIGN(2)*W(7))
      TA1=GX(14)/(2.0*GX(15)*(1./CDSAP(4)-2.*PIARE*GX(10)))
      CON(34)=TA1
      IF(OMPH*OMPH.GT.0.) GO TO 80
      CON(35)=99.
      GO TO 82
   80 CON(35)=OMSP/OMPH
   82 IF(OMPHA*OMPHA.GT.0.) GO TO 84
      CON(36)=99.
      GO TO 86
   84 CON(36)=OMSPA/OMPHA
***********************************************************************
*     THE FOLLOWING SECTION CALCULATES THE RESPONSE OF A PITCH        *
*     ATTITUDE HOLD/RATE COMMAND AUTOPILOT IN TURBULENCE.             *
***********************************************************************
```

```
 86 WCR2=NZOACR
    WCR=WCR2**0.5
    DZETA=.7
    DM=DERCR(15)+DERCR(3)
    DM1=1.-DERCR(12)*DERCR(12)*DERCR(15)*WCR2/DM
    DM2=2.*DZETA*WCR*DERCR(12)*DERCR(15)-DERCR(15)-DERCR(3)*DERCR(15)
    KTDOT1=2.*DZETA*WCR*(DERCR(12)*UOCR-1.)-DERCR(9)-DERCR(3)*UOCR
   $ +DERCR(12)*WCR2*(1.-DERCR(12)*UOCR)/DM
    KTDOT1=KTDOT1/(DM1*DM2)
    KT1=WCR2*(1.-DERCR(12)*UOCR)+KTDOT1*WCR2*DERCR(12)*DERCR(15)
    KT1=KT1/DM
    XNUM=((DERCR(9)/UOCR)**2.0)*(KT1*KT1-KTDOT1*KTDOT1*(2.0*DZETA
   $*WCR+WCR*WCR*GX(11)/(UOCR)))
    TSTO=WCR2*GX(11)/UOCR
    DENOM=TSTO-(1.+2.*DZETA*TSTO/WCR)*(2.*DZETA*WCR+TSTO)
    SIG=(ABS(XNUM/DENOM))**0.5
    CON(37)=3.0*SIG
    HX(15)=3.0*SIG
C
C        ASSUME THAT CRUISE RMS IS 3 FT/SEC
C
    CON(38)=KTDOT1
    CON(39)=KT1
    WA2=NZOAA.
    WA=WA2**0.5
    DM=DERAP(15)+DERAP(3)
    DM1=1.-DERAP(12)*DERAP(12)*DERAP(15)*WA2/DM
    DM2=2.*DZETA*WA*DERAP(12)*DERAP(15)-DERAP(15)-DERAP(3)*DERAP(15)
    KTDOT2=2.*DZETA*WA*(DERAP(12)*GX(14)-1.)-DERAP(9)-DERAP(3)*GX(14)
   $ +DERAP(12)*WA2*(1.-DERAP(12)*GX(14))/DM
    KTDOT2=KTDOT1/(DM1*DM2)
    KT2=WA2*(1.-DERAP(12)*GX(14))+KTDOT1*WA2*DERAP(12)*DERAP(15)
    KT2=KT1/DM
    XNUM=((DERAP(9)/GX(14))**2.0)*(KT2*KT2-KTDOT2*KTDOT2*(2.*WA*DZETA
   $+WA2*GX(11)/(GX(14))))
    TSTO=WA2*GX(11)/GX(14)
    DENOM=TSTO-(1.+2.*DZETA*TSTO/WA)*(2.0*DZETA*WA+TSTO)
    SIGA=(ABS(XNUM/DENOM))**0.5
    CON(40)=7.0*SIGA
    GX(13)=7.0*SIGA
C
C        ASSUME THAT APPROACH RMS IS 7 FT/SEC
C
    CON(41)=KTDOT2
    CON(42)=KT2
    TA=HX(10)
    TUR=GX(11)/UOCR
    AO=TUR*TA
    A2=TA*(2.*DZETA*WCR+WCR2*TUR)+1.+2.*DZETA*WCR*TUR
    A1=TA*(1.+2.*DZETA*WCR*TUR+TUR)
    A3=WCR2*(TA+TUR)+2.*DZETA*WCR
    A4=WCR
    VV1=-A1*A4+A2*A3
```

```
      XNUM=(1./TA)*(DERCR(9)/UOCR)*KTDOT1*KTDOT1*VV1
      XNUM=XNUM-A3*TUR*(DERCR(9)/UOCR)**2.0*KT1*KT1
      DEN=AO*A3*A3+A1*(A1*A4-A2*A3)
      SIGDOT=(ABS(XNUM/DEN))**0.5
      CON(43)=3.0*SIGDOT
      TUR=GX(11)/GX(14)
      AO=TUR*TA
      A2=TA*(2.*DZETA*WA+WA2*TUR)+1.+2.*DZETA*TUR*WA
      A1=TA*(1.+2.*DZETA*WA*TUR+TUR)
      A3=WA2*(TA+TUR)+2.*DZETA*WA
      A4=WA2
      VV1=-A1*A4+A2*A3
      XNUM=XNUM-A3*TUR*(DERAP(9)/UOCR)**2.0*KT2*KT2
      DEN=AO*A3*A3+A1*(A1*A4-A2*A3)
      SIGDOTA=(ABS(XNUM/DEN))**0.5
      CON(44)=7.0*SIGDOTA
******************************
*     OUTPUT SECTION         *
******************************
      IF(OUTPUT.EQ.0) GO TO 999
      WRITE(6,102)
  102 FORMAT(*1*///20X*AIRCRAFT OPTIMIZATION CONSTRAINTS*//5X
     $*DESIGN CONSTRAINTS*/14X*ID*5X*CONSTRAINT*T49*VALUE*
     $9X*SL*10X*SU*9X*VIOLATION?*)
      WRITE(6,112) (I,CON(I),SL(I),SU(I),XINEQ(I),I=1,6)
  112 FORMAT(/10X,I5,*  CRUISE THRUST*T45,4F12.4/10X,I5,*  2ND SEGMENT C
     $LIMB*T45,4F12.4/10X,I5,*  MISSED APPROACH CLIMB*T45,4F12.4/10X,I5
     $*  LANDING*T45,4F12.4/10X,I5*  TAKE-OFF*T45,4F12.4/10X,I5
     $*  LANDING GEAR LIMIT*T45,4F12.4)
      WRITE(6,116) CON(45),SL(45),SU(45),XINEQ(45)
  116 FORMAT(13X*45  PASSENGER VOLUME*T45,4F12.4)
      WRITE(6,117) CON(48),SL(48),SU(48),XINEQ(48)
  117 FORMAT(13X*48  CRUISE ALTITUDE*T45,4F12.4)
      WRITE(6,118) CON(49),SL(49),SU(49),XINEQ(49)
  118 FORMAT(13X*49  CRUISE ALTITUDE(L/D(MAX))*T45,4F12.4)
      WRITE(6,119) CON(50),SL(50),SU(50),XINEQ(50),
     $ CON(51),SL(51),SU(51),XINEQ(51)
  119 FORMAT(13X*50  CRUISE WING CL*T45,4F12.4/
     $ 13X*51  APPROACH WING CL*T45,4F12.4)
      WRITE(6,120) CON(52),SL(52),SU(52),XINEQ(52)
  120 FORMAT(13X*52  AR(TAIL)/AR(WING)*T45,4F12.4)
      WRITE(6,122)
  122 FORMAT(//5X*HANDLING QUALITY CONSTRAINTS*)
      WRITE(6,128) (I,CON(I),SL(I),SU(I),XINEQ(I),I=7,10)
  128 FORMAT(/10X,I5*  STATIC STAB. (CR)*T45,4F12.4/10X,I5
     $*  STATIC STAB. (AP)*
     $T45,4F12.4/10X,I5*  MANEUVER MARGIN (CR)*T45,4F12.4/10X,I5
     $*  MANEUVER MARGIN (AP)*T45,4F12.4)
      WRITE(6,132) (I,CON(I),SL(I),SU(I),XINEQ(I),I=11,15)
  132 FORMAT(10X,I5*  TAIL LIFT (AP)*T45,4F12.4/10X,I5*  NOSE GEAR UNSTI
     $CK*T45,4F12.4/10X,I5*  DYN. STAB. (CR)*T45,4F12.4/10X,I5
     $*  DYN. STAB. (AP)*T45,4F12.4/10X,I5*  PHUGOID FREQ (CR)*T45,
     $4F12.4)
```

```
      WRITE(6,136)  (I,CON(I),SL(I),SU(I),XINEQ(I),I=16,20)
  136 FORMAT(10X,I5*  PHUGOID FREQ  (AP)*T45,4F12.4/10X,I5
     $*  PHUGOID DAMPING  (CR)*T45,4F12.4/10X,I5*  PHUGOID DAMPING  (AP)*
     $T45,4F12.4/10X,I5*  SHORT PER. FREQ. (CR)*T45,4F12.4/10X,I5
     $*  SHORT PER. FREQ. (AP)*T45,4F12.4)
      WRITE(6,141)  (I,CON(I),SL(I),SU(I),XINEQ(I),I=21,25)
  141 FORMAT(10X,I5*  SHORT PER. DAMP  (CR)*T45,4F12.4/10X,I5
     $*  SHORT PER. DAMP  (AP)*T45,4F12.4/10X,I5*  TIME-TO-DOUBLE  (CR)*
     $T45,4F12.4/10X,I5*  TIME-TO-DOUBLE  (AP)*T45,4F12.4/10X,I5
     $*  TIME-TO-HALF  (CR)*T45,4F12.4)
      WRITE(6,146)  (I,CON(I),SL(I),SU(I),XINEQ(I),I=26,30)
  146 FORMAT(10X,I5*  TIME-TO-HALF  (AP)*T45,4F12.4/10X,I5
     $*  FLIGHT PATH STAB.  (AP)*T45,4F12.4/10X,I5*  VERT. GAIN  (CR)*
     $T45,4F12.4/10X,I5*  VERT. GAIN  (AP)*T45,4F12.4/10X,I5
     $*  T(THETA(2))  (CR)*T45,4F12.4)
      WRITE(6,147)  (I,CON(I),SL(I),SU(I),XINEQ(I),I=31,40)
  147 FORMAT(10X,I5*  T(THETA(2))  (AP)*T45,4F12.4/10X,I5*  WW/NZA  (CR)*
     $T45,4F12.4/10X,I5*  WW/NZA  (AP)*T45,4F12.4/10X,I5*  T(1)  (AP)*
     $T45,4F12.4/10X,I5*  MODE RATIO  (CR)*T45,4F12.4/10X,I5
     $*  MODE RATIO  (AP)*T45,4F12.4/10X,I5*  ELE. VAR.  (CR)*T45,4F12.4
     $/10X,I5*  THETA-DOT GAIN  (CR)*T45,4F12.4/10X,I5*  THETA GAIN*
     $T45,4F12.4/10X,I5*  ELE. VAR.  (AP)*T45,4F12.4)
      WRITE(6,152)  (I,CON(I),SL(I),SU(I),XINEQ(I),I=41,44)
  152 FORMAT(10X,I5*  THETA-DOT GAIN  (AP)*T45,4F12.4/10X,I5
     $*  THETA GAIN  (AP)*T45,4F12.4/10X,I5*  ELE-DOT VAR.  (CR)*T45,
     $4F12.4/10X,I5*  ELE-DOT VAR.  (AP)*T45,4F12.4)
      WRITE(6,157)  (I,CON(I),SL(I),SU(I),XINEQ(I),I=46,47)
  157 FORMAT(10X,I5*  TRIM ELEVATOR  (CR)*T45,4F12.4/
     $10X,I5*  TRIM ELEVATOR  (AP)*T45,4F12.4)
      WRITE(6,167)
  167 FORMAT(1H1/)
  999 RETURN
      END
      SUBROUTINE DIMDER(U,ICR,C)
      REAL C(15),A(15),ANS(4)
      COMMON /DEVAR/DESIGN(15),ITERM(10),CST(10)
      COMMON /DRAG/CDS(6),CDSAP(6)
      COMMON /GEOM/W(20),HX(20),GX(35),PX(15)
      COMMON /STAB/DERIVCR(15),DERIVAP(15),STOR(20)
      COMMON /WTSVE/WTS(20)
*************************************************************************
*     DIMDER CONVERTS FROM NON-DIMENSIONAL TO DIMENSIONAL STABILITY     *
*     DERIVATIVES.                                                      *
*************************************************************************
      IF(ICR.EQ.0) GO TO 20
      DO 15 I=1,15
   15 A(I)=DERIVCR(I)
      XMU=WTS(7)
      CALL AT62(WTS(11),ANS)
      DRAG=CDS(5)
      XIY=WTS(13)
      CL=GX(9)
      WT=WTS(15)
```

```
      GO TO 30
   20 DO 25 I=1,15
   25 A(I)=DERIVAP(I)
      XMU=WTS(9)
      CALL AT62(500.,ANS)
      DRAG=CDSAP(5)
      XIY=WTS(14)
      CL=GX(10)
      WT=WTS(1)*GX(27)
   30 T1=ANS(1)*DESIGN(1)*U*32.174/WT
      C(1)=T1*(-A(1)-DRAG)/2.0
      C(2)=T1*(CL-A(2))/2.0
      T2=ANS(1)*DESIGN(1)*U/(2.0*XIY)
      C(3)=A(3)*W(10)*T2
      C(4)=T1*(-CL-A(4))
      C(5)=T1*(-DRAG-A(5))
      C(6)=W(10)*A(6)*2.0*T2
      C(7)=-W(10)*A(7)*2.0*T2
      C(8)=0.
      C(9)=W(10)*W(10)*A(9)*T2/2.0
      C(10)=-A(10)/(4.0*XMU)
      C(11)=0.
      C(12)=ANS(1)*DESIGN(1)*W(10)*W(10)*A(12)/(4.0*XIY)
      C(13)=-T1*U*A(13)/2.
      C(14)=-T1*U*A(14)/2.
      C(15)=T2*U*W(10)*A(15)
      RETURN
      END
      SUBROUTINE LONGRT(DIM,U,ROOT,PARAM,A,NO,TCONST)
      REAL DIM(15),PARAM(4),A(5),OM(2),ZET(2)
      REAL TCONST(4)
      COMPLEX ROOT(4),COM(4)
**********************************************************************
*     LONGRT FINDS ROOTS OF FOURTH ORDER DYNAMICS MODEL              *
**********************************************************************
      A(1)=1.
      A(2)=-DIM(9)-U*DIM(12)-DIM(1)-DIM(5)
      A(3)=DIM(1)*DIM(9)-DIM(3)*U-DIM(2)*DIM(4)+DIM(5)*(DIM(9)+U*DIM(12)
     $+DIM(1))
      A(4)=-DIM(5)*(DIM(1)*DIM(9)-U*DIM(3))+DIM(4)*(DIM(2)*DIM(9)
     $+32.174*DIM(12))-DIM(6)*(U*DIM(2)-32.174)
      A(5)=32.174*(DIM(4)*DIM(3)-DIM(6)*DIM(1))
      IDEGRE=4
      CALL RPOLY(IDEGRE,A,ROOT,IERR)
      IF(IERR.LT.0) WRITE(6,22) IDEGRE
   22 FORMAT(*1*///*ROOT SOLVER BROKE DOWN--ONLY*I5* TERMS FOUND*//)
      OM(1)=0.
      OM(2)=0.
      ZET(1)=1.
      ZET(2)=1.
      DO 25 I=1,4
      TCONST(I)=0.
      PARAM(I)=0.
```

```
   25 CONTINUE
      ICOM=0
      ICON=0
C
C     ASSUME THE ROOTS ARE GIVEN IN PAIRS
C
      DO 35 IROOT=1,4
      D=AIMAG(ROOT(IROOT))
      IF(D.EQ.0.0) GO TO 30
      ICOM=ICOM+1
      COM(ICOM)=ROOT(IROOT)
      GO TO 35
   30 ICON=ICON+1
      TCONST(ICON)=ROOT(IROOT)
   35 CONTINUE
      JCOM=0
      IF(ICOM.LT.1) GO TO 200
      DO 40 I=1,ICOM,2
      D=AIMAG(COM(I))
      E=COM(I)
      JCOM=JCOM+1
      OM(JCOM)=(D*D+E*E)**0.5
      ZET(JCOM)=-E/OM(JCOM)
   40 CONTINUE
  200 NO=ICON
********************************************
*     PHUGOID(1,2); SHORT PERIOD(3,4)      *
********************************************
      IF(OM(1).GT.OM(2)) GO TO 250
  210 PARAM(3)=OM(2)
      PARAM(4)=ZET(2)
      PARAM(1)=OM(1)
      PARAM(2)=ZET(1)
      GO TO 275
  250 IF(OM(1).LT.0.8) GO TO 210
      PARAM(3)=OM(1)
      PARAM(4)=ZET(1)
      PARAM(1)=OM(2)
      PARAM(2)=ZET(2)
  275 IF(NO.LT.1) GO TO 999
      SMALL=100.
      SMALL2=100.
      DO 300 I=1,NO
      IF(TCONST(I).LT.SMALL) SMALL=TCONST(I)
  300 CONTINUE
      DO 310 I=1,NO
      IF(TCONST(I).LT.SMALL2.AND.TCONST(I).NE.SMALL)
     $ SMALL2=TCONST(I)
  310 CONTINUE
      R12=ABS(SMALL*SMALL2)
      ZT=-(SMALL+SMALL2)/(2.0*R12**0.5)
      IF(R12.EQ.0.0) ZT=0.
      IF(PARAM(3).EQ.0.) PARAM(3)=R12**0.5
```

```
      IF(PARAM(4).EQ.1.0) PARAM(4)=ZT
  999 RETURN
      END
```

APPENDIX II - SAMPLE INPUT DECK

```
  MIL 8785B LEVEL III (SIMPACT)                     (Title)
7   52                       (Number of design variables, Number of constraints)
-.2170  .3383  -.1576  -.1123  -.3620  .0358  .1895            (Initial XBAR)
1  1000.  4000.              (Design variable number, lower bound, upper bound)
2  3.  15.
3  120.  260.
4  100.  1600.
5  2.  15.
6  10000.  120000.
7  -0.5  1.
1  1.  2.                    (Constraint number, lower bound, upper bound)
2  1.  5.
3  1.  5.
4  1000.  8000.
5  2000.  10000.
6  0.  1.0
7  -999.  999.
8  -1.0   -0.10
9  -999.  999.
10  -999.  999.
11  -.8  .8
12  1.  3.00
13  -999.  999.
14  -999.  999.
15  -999.  999.
16 -999.   999.
17  -999.  999.
18  -999.0  999.0
19  -999.  999.
20  -999.  999.
21  -999.0  999.0
22  -999.0  999.0
23  -999.  999.
24  -999.  999.
25  -999.  999.
26  -999.  999.
27  -3.  .24
28  -999.  999.
29  -999.  999.
30  -999.  999.
31  -999.  999.
32  -999.0  999.0
33  -999.0  999.0
34  -999.  999.
35  -999.  999.
36  -999.  999.
37  -999.  999.
38  -999.  999.
```

```
39  -999.  999.
40  -999.  999.
41  -999.  999.
42  -999.  999.
43  -999.  999.
44  -999.  999.
45  1.  2.5
46  -1.  1.
47  -1.  1.0
48  30000.  46000.
49  30000.  52000.
50  .1  .75
51  1.  2.7
52  0.  1.
2                                              (Number of calls to NELMIN)
1.  1.0E-6  1.0E10  .2  0                  (SCF, REQMIN, CAYY, STEP, ILINE)
190000.                                                      ( WTS(1) )
0.  0.  0.  0.  0.  0.  0.  0.                            ( PX(1-- 8) )
1  0  7  2  1  1979  0  0  0  0                      ( ITERM(1-- 10) )
.75  .65                                              ( WTS(16), CG(5) )
21.2  2.0  .38  .14  5.                                    ( W(1-- 5) )
.98  .850  0.3  -.15  -1.                       ( W(6-- 8, 14, 16) )
-.15  -.25  -.12  .025                                    ( W(17-- 20) )
.4  .1  .46  .2                                     ( HX(1-- 3, 10) )
.08  .36  1.8  35.  .3                                  ( HX(16-- 20) )
.8  3000.  16.667  22.58  8.33                              ( GX(3-- 7) )
2.2    1443.38  3.15  -5.25  -12.725     ( GX(8, 11, 12, 17, 18) )
200.  8870.  41100.  7500.                              ( GX(19-- 22) )
4.0  .55  .09  3200.  .70                               ( GX(23-- 27) )
30.  0.  -99.                                         ( GX(32, 34, 16) )
```

APPENDIX III - SAMPLE OUTPUT CORRESPONDING TO SAMPLE INPUT

```
RUN NO=  MIL 8785B LEVEL III (SIMPACT)

     NO. OF VARIABLES=        7
     NO. OF CONSTRAINTS=     52
     VARIABLES=                 -.2170       .3383     -.1576      -.1123      -.3620
                                 .0358       .1895
J,XL,XU,AVE,AMP=      1       1000.0000     4000.0000     2500.0000     1500.0000
J,XL,XU,AVE,AMP=      2          3.0000       15.0000        9.0000        6.0000
J,XL,XU,AVE,AMP=      3        120.0000      260.0000      190.0000       70.0000
J,XL,XU,AVE,AMP=      4        100.0000     1600.0000      850.0000      750.0000
J,XL,XU,AVE,AMP=      5          2.0000       15.0000        8.5000        6.5000
J,XL,XU,AVE,AMP=      6      10000.0000   120000.0000    65000.0000    55000.0000
J,XL,XU,AVE,AMP=      7         -.5000        1.0000         .2500         .7500
I,SL(I),SU(I)=      1       1.00       2.00
I,SL(I),SU(I)=      2       1.00       5.00
I,SL(I),SU(I)=      3       1.00       5.00
I,SL(I),SU(I)=      4    1000.00    8000.00
I,SL(I),SU(I)=      5    2000.00   10000.00
I,SL(I),SU(I)=      6       0.00       1.00
I,SL(I),SU(I)=      7    -999.00     999.00
I,SL(I),SU(I)=      8      -1.00       -.10
I,SL(I),SU(I)=      9    -999.00     999.00
I,SL(I),SU(I)=     10    -999.00     999.00
I,SL(I),SU(I)=     11       -.80        .00
I,SL(I),SU(I)=     12       1.00       3.00
I,SL(I),SU(I)=     13    -999.00     999.00
I,SL(I),SU(I)=     14    -999.00     999.00
I,SL(I),SU(I)=     15    -999.00     999.00
I,SL(I),SU(I)=     16    -999.00     999.00
I,SL(I),SU(I)=     17    -999.00     999.00
I,SL(I),SU(I)=     18    -999.00     999.00
I,SL(I),SU(I)=     19    -999.00     999.00
I,SL(I),SU(I)=     20    -999.00     999.00
I,SL(I),SU(I)=     21    -999.00     999.00
I,SL(I),SU(I)=     22    -999.00     999.00
I,SL(I),SU(I)=     23    -999.00     999.00
I,SL(I),SU(I)=     24    -999.00     999.00
I,SL(I),SU(I)=     25    -999.00     999.00
I,SL(I),SU(I)=     26    -999.00     999.00
I,SL(I),SU(I)=     27      -3.00        .24
I,SL(I),SU(I)=     28    -999.00     999.00
I,SL(I),SU(I)=     29    -999.00     999.00
I,SL(I),SU(I)=     30    -999.00     999.00
I,SL(I),SU(I)=     31    -999.00     999.00
I,SL(I),SU(I)=     32    -999.00     999.00
I,SL(I),SU(I)=     33    -999.00     999.00
I,SL(I),SU(I)=     34    -999.00     999.00

I,SL(I),SU(I)=     35    -999.00     999.00
I,SL(I),SU(I)=     36    -999.00     999.00
I,SL(I),SU(I)=     37    -999.00     999.00
I,SL(I),SU(I)=     38    -999.00     999.00
I,SL(I),SU(I)=     39    -999.00     999.00
I,SL(I),SU(I)=     40    -999.00     999.00
I,SL(I),SU(I)=     41    -999.00     999.00
I,SL(I),SU(I)=     42    -999.00     999.00
I,SL(I),SU(I)=     43    -999.00     999.00
I,SL(I),SU(I)=     44    -999.00     999.00
I,SL(I),SU(I)=     45       1.00       2.50
I,SL(I),SU(I)=     46      -1.00       1.00
I,SL(I),SU(I)=     47      -1.00       1.00
I,SL(I),SU(I)=     48   30000.00   46000.00
I,SL(I),SU(I)=     49   30000.00   52000.00
I,SL(I),SU(I)=     50        .10        .75
I,SL(I),SU(I)=     51       1.00       2.70
I,SL(I),SU(I)=     52       0.00       1.00
     NUMBER OF REQUESTED NELMINS=      2
```

```
                    /\/\/\FUNCTION INPUT/\/\/\

WTS(1)                       190000.0000
ITEPM                          1     0     7     2     1  1979     0     0     0     0
PX=                                0.0000        0.0000        0.0000        0.0000        0.0000
                                   0.0000        0.0000        0.0000             I             I
                                        I             I             I             I             I

WTS(16)                             .7500
W...1-5=                          21.2000        2.0000         .3800         .1400        5.0000
W...6-8,14,16=                      .9800         .8500         .3000        -.1500       -1.0000
W...17-20=                        -.1500        -.2500        -.1200         .0250
HX...1-3,10=                        .4000         .1000         .4600         .2000
HX...16-20-                         .0800         .3600        1.8000       35.0000         .3000
GX...3-7=                           .8000     3000.0000       16.6670       22.5800        8.3300
GX...8,11-12,17,18=                2.2000     1443.3800        3.1500       -5.2500      -12.7250
GX...19-22=                      200.0000     8870.0000    41100.0000     7500.0000
GX...23-27=                        4.0000         .5500         .0900     3200.0000         .7000
GX...32,34=                       30.0000        0.0000

INITAIL XBAR                 -.2170      .3383      -.1576      -.1123      -.3620
                              .0358      .1895
INITAIL STEPS                 .2000      .2000       .2000       .2000       .2000
                              .2000      .2000
REOMIN,SCF,CAYY           .1000E-06  .1000E+01   .1000E+12

/\/\/\NELMIN COMPLETE/\/\/\

XMIN=                        -.15129477      .18686605     -.15769764      .05231575     -.48770473
                              .20364349      .16912692
YNEWLO=                      .200533E+04
ICOUNT=                   817
TOT. FUNCTION CALLS=      825


TOT. WEIGHT ITERATIONS       2973.00
AVE. WT. ITEPATIONS PER         3.604

INITAIL XBAR                 -.1513      .1869      -.1577       .0523      -.4877
                              .2036      .1691
INITAIL STEPS                 .1000      .1000       .1000       .1000       .1000
                              .1000      .1000
REOMIN,SCF,CAYY           .1000E-07  .1000E+01   .1000E+13

/\/\/\NELMIN COMPLETE/\/\/\


XMIN=                        -.15147188      .18662137     -.15771096      .05225537     -.48798586
                              .20313425      .16913646
YNEWLO=                      .200507E+04
ICOUNT=                   484
TOT. FUNCTION CALLS=     1317


TOT. WEIGHT ITERATIONS       4421.00
AVE. WT. ITERATIONS PEP         3.357
```

AIRCRAFT SIZING PROGRAM

```
DESIGN VARIABLES
     WING AREA (FTXX2)=                  2146.4606
     WING ASPECT RATIO=                    10.7338
     FUSELAGE LENGTH (FT)=                172.8356
     HOR. TAIL AREA (FTXX2)=              911.4927
     HOR. TAIL ASPECT RATIO=                3.9914
     TOTAL THRUST (LBS)=                82253.2621
     AFT MOST CG=                            .4469
     CRUISE MACH NO.=                            I
     SWEEP=                                      I
     WING T/C=                                   I
     WING TAPER RATIO=                           I
     FUSE. DIA=                                  I


INPUT CONSTANTS

     WING...SWEEP,INCIDENCE,TAPER RATIO:              21.2000     2.0000     .3800
     THICKNESS,TWIST,E1,E2,DESIGN CL:                   .1400     5.0000     .9800      .8500      .3000
     CM(CR,APP)                                        -.1500     -.1207
     DELTA CM(10,25 DEGREES FLAP):                     -.1500     -.2500
     ANGLE OF ZERO LIFT(0,10,45 DEGREES FLAP):        -1.0000    -5.2500   -12.7250
     DELTA CD (10-45 DEGREES FLAP):                     .0250
     TURBULENCE LENGTH/ROOT 3,FUSE. DIA.:           1443.3800    16.6670
     CL-MAX(TO),CL-MAX(L):                             2.2000     3.1500
     MISSION...MACH NO.,RANGE,NO. PASS:                 .8000  3000.0000   200.0000
     CARGO WEIGHT:                                  7500.0000
     DELTA CG, WTL(MAX)/WTO:                           4.0000      .7000
     ENGINE...L,W,WT,TREF:                            22.5800     8.3300  8870.0000  41100.0000
     TAIL...TAPER RATIO,THICKNESS,ELE EFF:              .4600      .1000     .4600
     ELEVATOR TIME CONSTANT:                            .2000
     ECONOMICS...LOAD FACT,$/SEAT MI,BLK HR/YR:         .5500      .0900  3200.0000

SOME GEOMETRY CALCULATIONS
     WING...SPAN,CMAC:                                151.7881    15.0926
     TAIL...SPAN,CMAC,VBAR,SWEEP                       60.3166    16.0370    2.1883    30.0000
     VERT. TAIL...VBAR,TAPER,AR,SWEEP,SR/SV,SV:          .0800      .3600    1.8000    35.0000      .3000    352.7617

                              ***WEIGHT ESTIMATION***

     WING                             37077.8
     HOR. TAIL                         7028.3
     VERT. TAIL                        1088.5
     FUSELAGE                         47372.1
     LANDING GEAR                     14912.5
     CONTROL SYSTEM                    2541.3
     INSTRUMENTS                        116.2
     MISC INTERIOR                      250.6
     ELECTRICAL                        2192.3
     STARTERS                           506.9
     FURNISHINGS                       7902.0
     OXYGEN                             300.7
     WINDOWS                            501.6
     BAGGAGE HNDLING                    144.7
     AIR CONDITIONING                  3547.0
     THRUST REVERSER                   1500.0
     ENGINE                           18639.1
     ENGINE CONTROLS                    219.5
     FUEL SYS                          2571.1
     ACTIVE CONTROL SYSTEMS             250.0

 EMPTY WEIGHT                        156200.2


     PASSENGERS                       34000.0
     CREW                              1700.0
     BAGGAGE                           7000.0
     WTFOOD                             214.5
     CARGO                             7500.0

 FIXED WEIGHT                         50414.5


 FUEL                                 74815.1


 TAKE-OFF WEIGHT                     281429.7


     NO. OF ITERATIONS REQUIRED            1

WTHT(1,2,3),WTFUSE(1,2),WTINIT=          6835.45    7829.81    6419.59   57851.38   36892.78    281429.74
```

/\/\/\CRUISE ANALYSIS/\/\/\

TOTAL MISSION RANGE= 3000.00
CLIMB DISTANCE= 189.00
DESCENT DISTANCE= 113.00

LEG	MACH NO.	CL	V	TIME	L/D	TSFC	T/T(IN)	ALT(BEG)	ALT(END)	WT(BEG)	WT(END)	DIST	TCON	GAMA	CLM
1	.79	.593	763.4	.597	18.78	.471	.233	36000.0	37398.1	263432.3	259519.3	269.8	1.348	.0007	.751
2	.80	.606	774.5	.588	18.58	.478	.217	37398.1	37714.4	259519.3	255617.9	269.8	1.279	0.0000	.733
3	.80	.606	774.5	.588	18.55	.479	.214	37714.4	38031.6	255617.9	251765.1	269.8	1.279	0.0000	.734
4	.80	.606	774.5	.588	18.53	.480	.211	38031.6	38349.6	251765.1	247960.6	269.8	1.278	0.0000	.735
5	.80	.606	774.5	.588	18.50	.480	.208	38349.6	38668.4	247960.6	244203.8	269.8	1.276	0.0000	.735
6	.80	.606	774.5	.588	18.48	.481	.205	38668.4	41197.6	244203.8	240494.3	269.8	1.244	.0013	.736
7	.80	.674	774.5	.588	18.31	.486	.182	41197.6	41785.7	240494.3	236771.9	269.8	1.134	.0002	.743
8	.80	.682	774.5	.588	18.26	.487	.176	41795.7	42272.8	236771.9	233087.9	269.8	1.117	.0001	.749
9	.80	.687	774.5	.588	18.21	.488	.172	42272.8	42657.4	233087.9	229445.8	269.8	1.106	.0000	.751
10	.80	.689	774.5	.588	18.18	.488	.169	42657.4	43026.6	229445.8	225849.4	269.8	1.101	.0000	.752
11	.65	.569	644.4	2.620	19.00	.425	.300	30000.0	30000.0	225849.4	213004.8	1000.0	2.078	0.0000	.720

FUEL WEIGHT ANALYSIS

TAKE-OFF	281429.74
START-CRUISE	263432.31
MID-CRUISE	244203.84
END-CRUISE	225849.37
AFTER RESERVE	213004.83
AFTER DESCENT/TAXI	206614.68
NET FUEL WEIGHT(LBS)	74815.05

CRUISE ALTITUDES

LEG 1	36000.00
LEG 2	43026.59
RESERVE LEG	30000.00

FLIGHT LENGTH (HR)	6.67
AVERAGE SPEED (KTS)	449.82
BLOCK TIME (HR)	7.00
BLOCK SPEED (KTS)	428.80
BLOCK FUEL (LBS)	44358.42
BLOCK FUEL (GALS)	6931.00
NAUT. MI/GAL	.43
NAUT. SEAT MI./GAL.	86.57

INSTALLED THRUST (LBS)	82253.26
NO. OF ENGINES	2
ENGINE THRUST (LBS)	41126.63
REFERENCE ENGINE (LBS)	41100.00
SCALE FACTOR	1.001

```
          ***DRAG ANALYSIS***

     WING                                .0078
     HORIZONTAL TAIL                     .0029
     VERTICAL TAIL                       .0011
     FUSELAGE                            .0025
     BASE                                .0002
     ENGINE NACELLE                      .0013
     CRUD/FLAPS                          .0007

AIRCRAFT DRAG                            .0184

     INTERFERENCE FACTOR IS 5 PERCENT


          ***REYNOLD'S NUMBERS***

     WING                          24674227.1
     HORIZONTAL TAIL               26218180.1
     VERTICAL TAIL                 24576159.6
     FUSELAGE                     282560989.7
     ENGINE                        36926961.6


               STABILITY AND CONTROL DERIVATIVES

     CL,M,CG. POSITION=           .606           .800           .314


                      CL             CD             CM

     ALPHA         6.69958         .28332       -1.51107
     VELOCITY       .2155          .0010         -.0003
     Q            16.85623        0.00000       -78.17792
     ALPHA-DOT    -5.19019        0.00000       -26.74635
     ELEVATOR       .68393         .00004        -3.52448


     NEUTRAL POINT                 .540
     STATIC STABILITY             -.226
     CM(ALPHA)                   -1.511
     STOR=
                                  5.718          3.890
                                   .339           .540

                                   .604          4.865
                                  3.051           .290

                                   .547           .737
                                  -.028          -.017

                                   .172           .205
                                      I              I

                                      I              I
                                      I              I

                        -.1729       -.2535         .1982        .0998    3232.9597
                         .6954       -.2789         .0040       5.7163       3.8903
                         .2724      77.7760        0.0000       3.5033        .6000
                         .1720       5.7179         .3387        .5400        .9902
                        1.2657       1.6135        3.7351        .9997        .3144

                         .0758
```

ACCURATE L/D ANALYSIS

CL(REQUESTED)=	.606	
CL(WING)=	.655	
CL(TAIL)=	-.118	
ELEVATOR(DEGREES)=	0.00	
STABILIZER(DEGREES)	-3.09	
INDUCED DRAG COMPONENTS		
WING=	.0146	
INTERFERENCE=	-.0019	
TAIL=	.C006	
(TRIM)=	.0009	
(TAIL)=	.0038	
SIGMA	.3715	
DRAG COEFFICIENTS		
INDUCED=	.0133	
ZERO LIFT=	.0184	
ELEVATOR=	0.0000	
WING(MACH), M(CRIT)=	.0011	.80
FUSE(MACH), M(CRIT)=	-.0000	.90
TOTAL=	.0328	
L/D=	18.477	
MAX. L/D=	19.466	
CL-L/D MAX=	.736	
MODIFIED CDO=	.0193	

AIRCRAFT COST ESTIMATES

	DEVELOPMENT	PRODUCTION
ENGINEERING	137888052.30	196227689.86
DEVELOPMENT SUPPORT	53303827.90	0.00
FLIGHT TEST	17877799.44	0.00
TOOLING	144227140.03	252884765.95
MANUFAC. LABOR	90339216.98	1048521091.06
QUALITY CONTROL	11744098.21	136307741.84
MATERIALS	10453988.44	471245525.87
ENGINE	11723234.28	976936190.35
AVIONICS	738462.00	92307750.00
ACTIVE CONTROLS SYSTEM	507692.63	63461578.13
TOTAL	478803512.20	3237892333.06

TOTAL COST PER AIRCRAFT= $ 16353461.72

MAINTENANCE OPEPATING COSTS

NO.	SYSTEM	LABOR	MATERIAL
1	INSP	35.52	5.78
2	AIR COND	5.10	4.52
3	AUTO PILOT	13.43	3.15
4	COMMUN	5.52	2.36
5	ELEC	4.31	5.75
6	FURN	26.10	13.93
7	FIRE PROT	9.37	1.46
8	FLT CONTL	8.65	5.49
9	FUEL	2.01	1.02
10	HYD POWER	3.33	3.95
11	ICE	.67	.56
12	INSTR	1.22	.49
13	LAND GEAR	13.64	28.07
14	LIGHTING	3.81	2.83
15	NAVIG	11.59	8.24
16	OXYGEN	1.05	1.50
17	PNUEMAT	1.28	6.95
18	WAT/WASTE	1.08	1.54
19	AIR APU	.32	.46
20	STRUCTURE	3.70	0.00
21	DOOPS	2.35	1.96
22	FUSELAGE	4.76	.58
23	NACELLES	.67	.28
24	WINGS	2.95	1.13
25	STAB	.83	.37
26	WINDOWS	.85	7.24

LABOR COST	65.64
MATERIAL COST	43.84
ENGINE LABOR COST	25.38
ENGINE MATERIAL COST	31.26

MAINTNENACE DOC IN 1976 DOLLARS PER HOUR 166.13

DIRECT OPERATING COSTS--DOLLARS/FLT. HOUR

DEPREC	393.52	19.63
SUPPORT	53.66	2.68
SPARES	26.83	1.34
DELAY	10.29	.51
INSURANCE	62.61	3.12
FUEL	743.00	37.06
MAINTENANCE	166.13	8.29
LANDING FEE	26.80	1.34
CREW	266.48	13.29
ATTENDANTS	164.11	8.18
FUEL SERVICE	77.18	3.85
CONTROL	14.46	.72
TOTAL DIRECT OPERATING COSTS	$ 2005.07	100.00

INDIRECT OPERATING COSTS--DOLLARS/FLT. HOUR

MAIN BURDN	174.44	19.21
FOOD COST	168.35	18.53
MOVIE	34.32	3.78
PASS INS	30.05	3.31
MISC PASS	6.30	.69
ADVERTISE	97.64	10.75
COMMISSION	135.79	14.95
RESER	84.75	9.33
PASS HDLG	55.28	6.09
BAG HDLG	25.23	2.78
CARGO HDLG	86.07	9.48
SERVICING	10.07	1.11
TOTAL INDIRECT OPERATING COSTS	908.28	100.00

PERFORMANCE FUNCTION SUMMARY

```
REVENUE PER BLOCK HOUR                    4245.12
TOTAL COST PER BLOCK HOUR                 2913.34
RETURN ON INVESTMENT                        .1229


   1  DOC/HR         2005.065
   2  DOC/FLT       14027.968
   3  ROI                .123
   4  FARE               .096
   5  SEAT-MI/GA       86.568
   6  L/D(MAX)         19.466
   7  MTOGW        281429.737
   8  FARE          31749.000
   9  PRICE          4538.000
```

DEBUG OF NOSE GEAR UNSTICK

```
ZACT,XMUF,ZACLG=          .2650          .0250          .5301
ANS(1,2,3,4)=        .002377    2116.238417    288.150000    1116.449901
VSTALL,CLW,O,XLW,XMW=       223.9352          .3642         48.2736      45279.0297     -449306.5408
C1,UNTHRU,XCG,XLTRO,CL2,COIT,XLT=     71.5390  82253.2621         .1819 -26773.6962        1.3723      -12.3103       77.7760
XLTAV,CON(12),STOR(7),STOR(8),W(2)=    -95242.4919          2.9739          3.0719          .2915          2.0000
GX(17),HX(6),W(10),XLTAV2,TLGE,CLTAV=        -5.2500          2.1883         15.0926     -79622.1355      15620.3564           -1.8096
```

DRAG ANALYSIS

WING	.0057
HORIZONTAL TAIL	.0022
VERTICAL TAIL	.0008
FUSELAGE	.0022
BASE	.0002
ENGINE NACELLE	.0010
CRUD/FLAPS	.0230
AIRCRAFT DRAG	.0376

INTERFERENCE FACTOR IS 5 PERCENT

REYNOLD'S NUMBERS

WING	64103940.0
HORIZONTAL TAIL	60115140.6
VERTICAL TAIL	63049159.4
FUSELAGE	734096064.3
ENGINE	95936692.3

APPROACH

STABILITY AND CONTROL DERIVATIVES

APPROACH

CL,M,CG, POSITION= 1.713 .192 .447

	CL	CD	CM
ALPHA	4.86481	.58147	-.40650
VELOCITY	.0529	.0007	.0005
Q	13.21915	0.00000	-61.30943
ALPHA-DOT	-3.48736	0.00000	-17.97125
ELEVATOR	.53636	.00004	-2.76400

NEUTRAL POINT	.547	
STATIC STABILITY	-.100	
CM(ALPHA)	-.407	
STOR=		
	5.718	3.090
	.339	.540
	.604	4.865
	3.051	.290
	.547	.737
	-.028	-.017
	.172	.205
	I	I
	I	I
	I	I

-.2448	-.3974	.2470	.0705	2263.2770
.7036	-.3996	.0057	4.0372	3.0509
.2724	77.7760	0.0000	3.5033	.9014
.2050	4.0383	.2902	.5469	.9902
1.2657	1.2293	15.3442	.9997	.4469

.0758

ACCURATE L/D ANALYSIS

APPROACH WITH 45 DEGREES FLAP

CL(REQUESTED)=	1.713
CL(WING)=	1.761
CL(TAIL)=	-.114
ELEVATOR(DEGREES)=	0.00
STABILIZER(DEGREES)	-1.35

INDUCED DRAG COMPONENTS

WING=	.1303
INTERFERENCE=	-.0048
TAIL=	.0005
(TRIM)=	.0028
(TAIL)=	.0050
SIGMA	.3715

DRAG COEFFICIENTS

INDUCED=	.1260	
ZERO LIFT=	.0626	
ELEVATOR=	.0005	
WING(MACH), M(CRIT)=	0.0000	0.00
FUSE(MACH), M(CRIT)=	0.0000	0.00
TOTAL=	.1891	

L/D=	9.057

MAX. L/D=	9.574
CL-L/D MAX=	1.358
MODIFIED CDO=	.0655

/\/\/\DIMENSIONAL STABILITY DERIVATIVES/\/\/\

DFRCR=

-.4615	.0221	-.0004
-.1126	-.0046	-.0000
-.0197	0.0000	-.3444
.0030	0.0000	-.0002
-36.3072	-.0446	-1.5935

DERAP=

-.5723	.1281	-.0004
-.3999	-.0430	.0000
-.0208	0.0000	-.3632
.0108	0.0000	-.0004
-16.7513	-.0263	-.5984

LONGITUDINAL DYNAMICS

/\/\CRUISE/\/\

COEFFICIENTS=	1.000000	.926777	.443088	.003445	.001313

ROOTS(REAL,IMAGINARY)

-.0008	.0544
-.0008	-.0544
-.4634	.4779
-.4634	-.4779

PHUGOID FREQUENCY	.0544
PHUGOID DAMPING	.0145
SHORT PER. FREQ.	.6656
SHORT PER. DAMPING	.6961

NO. OF NON-OSCILLATORY ROOTS=	0			
TIME CONSTANTS=	0.0000	0.0000	0.0000	0.0000

/\/\APPROACH/\/\

COEFFICIENTS=	1.000000	.977315	.284314	.035203	.004930

ROOTS(REAL,IMAGINARY)

-.0321	.1277
-.0321	-.1277
-.4887	.2134
-.4887	-.2134

PHUGOID FREQUENCY	.1317
PHUGOID DAMPING	.2439
SHORT PER. FREQ.	.5332
SHORT PER. DAMPING	.9164

NO. OF NON-OSCILLATORY ROOTS=	0			
TIME CONSTANTS=	0.0000	0.0000	0.0000	0.0000

CL-MAX TO(W),CL-MAX TO (AC),CL2	2.200	1.976	1.372
CL-MAX (W), CL-MAX (AC), CLA=	3.150	2.895	1.713

AIRCRAFT OPTIMIZATION CONSTRAINTS

DESIGN CONSTRAINTS

ID	CONSTRAINT	VALUE	SL	SU	VIOLATION?
1	CRUISE THRUST	1.1009	1.0000	2.0000	0.0000
2	2ND SEGMENT CLIMB	1.0000	1.0000	5.0000	0.0000
3	MISSED APPROACH CLIMB	1.3067	1.0000	5.0000	0.0000
4	LANDING	7999.9572	1000.0000	8000.0000	0.0000
5	TAKE-OFF	8106.5719	2000.0000	10000.0000	0.0000
6	LANDING GEAR LIMIT	.0231	0.0000	1.0000	0.0000
45	PASSENGER VOLUME	1.0000	1.0000	2.5000	0.0000
48	CRUISE ALTITUDE	38668.4002	30000.0000	46000.0000	0.0000
49	CRUISE ALTITUDE(L/D(MAX))	42707.2395	30000.0000	52000.0000	0.0000
50	CRUISE WING CL	.6552	.1000	.7500	0.0000
51	APPROACH WING CL	1.6799	1.0000	2.7000	0.0000
52	AR(TAIL)/AR(WING)	.3719	0.0000	1.0000	0.0000

HANDLING QUALITY CONSTRAINTS

ID	CONSTRAINT	VALUE	SL	SU	VIOLATION?
7	STATIC STAB. (CR)	-.0930	-999.0000	999.0000	0.0000
8	STATIC STAB. (AP)	-.1000	-1.0000	-.1000	0.0000
9	MANEUVER MARGIN (CR)	-.1383	-999.0000	999.0000	0.0000
10	MANEUVER MARGIN (AP)	-.2900	-999.0000	999.0000	0.0000
11	TAIL LIFT (AP)	-.3897	-.8000	.8000	0.0000
12	NOSE GEAR UNSTICK	2.9739	1.0000	3.0000	0.0000
13	DYN. STAB. (CR)	.0000	-999.0000	999.0000	0.0000
14	DYN. STAB. (AP)	.0002	-999.0000	999.0000	0.0000
15	PHUGOID FREQ (CR)	.0544	-999.0000	999.0000	0.0000
16	PHUGOID FREQ (AP)	.1317	-999.0000	999.0000	0.0000
17	PHUGOID DAMPING (CR)	.0145	-999.0000	999.0000	0.0000
18	PHUGOID DAMPING (AP)	.2439	-999.0000	999.0000	0.0000
19	SHORT PER. FREQ. (CR)	.6656	-999.0000	999.0000	0.0000
20	SHORT PER. FREQ. (AP)	.5332	-999.0000	999.0000	0.0000
21	SHORT PER. DAMP (CR)	.6961	-999.0000	999.0000	0.0000
22	SHORT PER. DAMP (AP)	.9164	-999.0000	999.0000	0.0000
23	TIME-TO-DOUBLE (CR)	99.0000	-999.0000	999.0000	0.0000
24	TIME-TO-DOUBLE (AP)	99.0000	-999.0000	999.0000	0.0000
25	TIME-TO-HALF (CR)	879.9306	-999.0000	999.0000	0.0000
26	TIME-TO-HALF (AP)	21.5825	-999.0000	999.0000	0.0000
27	FLIGHT PATH STAB. (AP)	-.0016	-3.0000	.2400	0.0000
28	VERT. GAIN (CR)	10.7991	-999.0000	999.0000	0.0000
29	VERT. GAIN (AP)	4.6431	-999.0000	999.0000	0.0000
30	T(THETA(2)) (CR)	.4548	-999.0000	999.0000	0.0000
31	T(THETA(2)) (AP)	-.0613	-999.0000	999.0000	0.0000
32	WW/NZA (CR)	.0410	-999.0000	999.0000	0.0000
33	WW/NZA (AP)	.0612	-999.0000	999.0000	0.0000
34	T(1) (AP)	-470.5489	-999.0000	999.0000	0.0000
35	MODE RATIO (CR)	12.2270	-999.0000	999.0000	0.0000
36	MODE RATIO (AP)	4.0494	-999.0000	999.0000	0.0000
37	ELE. VAR. (CR)	.0011	-999.0000	999.0000	0.0000
38	THETA-DOT GAIN (CR)	-2.8326	-999.0000	999.0000	0.0000
39	THETA GAIN	-7.5689	-999.0000	999.0000	0.0000
40	ELE. VAR. (AP)	.0094	-999.0000	999.0000	0.0000
41	THETA-DOT GAIN (AP)	-4.7296	-999.0000	999.0000	0.0000
42	THETA GAIN (AP)	12.6395	-999.0000	999.0000	0.0000
43	ELE-DOT VAR. (CR)	.3184	-999.0000	999.0000	0.0000
44	ELE-DOT VAR. (AP)	.4264	-999.0000	999.0000	0.0000
46	TRIM ELEVATOR (CR)	0.0000	-1.0000	1.0000	0.0000
47	TRIM ELEVATOR (AP)	0.0000	-1.0000	1.0000	0.0000

COMMON DUMP

```
DESIGN=          2146.46060      10.73378     172.03561     911.49267       3.99136
                82253.26214        .44692             I             I             I
                          I             I             I             I             I

ITERM=                    1             0             7             2             1
                       1979             0             0             0             0

PX=                  0.0000        0.0000        0.0000        0.0000        0.0000
                     0.0000        0.0000        0.0000             I             I
                          I             I             I             I             I

CDS=             .018435285    .007789039    .002919233  10.502198924    .032756961    .002502042

CDSAP=           .037644705    .005683280    .002194607   9.056559926    .189143434             I

W=             21.200000000   2.000000000    .380000000    .140000000   5.000000000
                 .980000000    .050000000    .300000000 151.788142905  15.092617897
                 .387874441    .537923713    .454076207   -.150000000    .932323000
               -1.000000000   -.150000000   -.250000000   -.120716547    .025000000

HX=              .400000000    .100000000    .460000000  60.316604382  16.037015975
                2.188322980    .577350275    .684725140    .469975411    .200000000
                -.113905980  -1.349290339   0.000000000   0.000000000    .001076020
                 .080000000    .360000000   1.800000000  35.000000000    .200000000

GX=             3.141592650  57.295779000    .800000000 3000.000000000 16.667000000
               22.580000000   8.330000000   2.200000000    .606075810   1.712980861
             1443.380000000   3.150000000    .009396652 275.808973653  32.174000000
              -99.000000000  -5.250000000 -12.725000000 200.000000000 8870.000000000
            41100.000000000 7500.000000000  6.000000000    .530000000    .090000000
             3200.000000000    .700000000    .636912543   1.761201849   1.100904500
                 .000464779  30.600000000  77.776023746   0.000000000 352.761609100

CG=                  .44692        .44692        .18189        .18189        .65000        .18000

DERIVCR=        6.699575550    .283323942   -.623274330    .215493621    .001025132
                -.000309099  16.856229399   0.000000000 -78.177918252  -5.190192106

                0.000000000 -26.746354637    .683934363    .000840805  -3.524484329

DERIVAP=        4.864806036    .581471571   -.486502225    .052930270    .000650740
                 .000531027  13.219153412   0.000000000 -61.309434652  -3.487362363
                0.000000000 -17.971267917    .536361547    .000840805  -2.764004004

STOR=           5.717885050   3.890306526    .336700386    .539955222    .505263796
                4.864806036   3.050093386    .290192456    .546927754    .736870592
                -.027830200   -.017255523    .172003128    .205004590             I
                          I             I             I             I             I

WTS=             281429.737    156200.104     75429.489    117150.138    219073.804
                  74815.053       431.364      1537.650        80.692      2177.046
                  38668.400     42707.240   13450000.000  13330000.000   244203.035
                       .750       449.025         6.669         6.996     44358.421

CST=               2005.065     14027.968          .123          .096        86.568
                     19.690    281429.737     31749.000      4538.000             I

XMIN=                  -.15147188      .18662137     -.15771096      .05229537     -.48790586
                        .20313425      .16913646
```

APPENDIX IV - Procedure File used to Execute OPDOT on the Langley Research Center Computer System

```
OPDOT,T7770,CM70000.                      RM 1174   ARBUCKLE/SLIWA
USER(820235N)
CHARGE,101264,LRC.
GET,OPDOT1.
GET,INPUT=SM10.               (SM10 is static margin 10% case -- APPENDIX II)
FTN(I=OPDOT1,OPT=2,R=0)
ATTACH(FTNMLIB/UN=LIBRARY,NA)
LDSET(PRESET=ZERO,LIB=FTNMLIB)    (All variables set to zero since program is
LGO.                                                already operational)
REWIND(TAPE4)
REWIND(LGO)
GET,PPB.                                   (PPB is the binary code for the
PPB.                                       OPDOT plotting preprocessor)
REWIND(TAPE7)
ATTACH(LRCGOSF/UN=LIBRARY)
GET,ABS2290/UN=181500N.                                      (Reference 28)
ABS2290,TAPE7.
PLOT.CALPOST,11
CONT.//BLANK PAPER, LEROY .3 PEN,
CONT.  BLACK INK, MULTIPLE PLOT MODE//
EXIT.
```

APPENDIX V - KEY PROGRAM VARIABLES

SUBROUTINE	VARIABLE	DESCRIPTION [UNITS, IF APPLICABLE]
SIMPACT (Main Program)	AMP	Amplitude of sinusoid transformation Z to X domain
	AVE	Ave. of sinusoid transformation Z to X domain
	CAYY	Penalizing weight for constraint violation
	FACT	= 1 if constraint is to be considered, = 0 otherwise
	GNORM	Constraint normalization--ave. of boundaries
	MC	Number of constraint violations
	MINEQ	Number of constraint functions
	NONEL	Number of NELMIN optimizations
	NVAR	Number of independent design variables
	REQMIN	Convergence criteria
	SCF	Scale factor to multiply performance index
	SL	Lower constraint boundary
	STEP	Initial optimizer step size
	SU	Upper constraint boundary
	XBAR	Independent design variable in Z domain
	XBARO	Initial value for independent design
	XINEQ	= 1 if constraint is violated, = 0 otherwise
	XL	Lowest allowable value of independent design variable in X domain
	XMIN	Vector of optimum independent design variable in Z domain
	XU	Upper value of independent design variable in X domain
	YNEWLO	Optimum performance index from NELMIN
NELMIN	IHO	Vertex with highest performance index
	ILO	Vertex with lowest performance index
	P	Coordinates of simplex
	PBAR	Centroid of simplex
	START	Initial independent design variables
FN	ILINE	= 1 output; = 0 no output
SETUP	COST	Performance index
	IPR	= 1 output; = 0 no output
	OBJ	Augmented performance index
	PENT	Penalty contribution to OBJ
	UNAUG	Unaugmented performance index
DOCOST	ATT	Attendant's cost [$/hr]
	BLKHR	Block hours of design mission [hr]
	CONTROL	Cost of logistics control [$/hr]
	COSTHR	Total cost per hour [$/hr]
	CREW	Crew cost [$/hr]
	DELAY	Delay cost [$/hr]
	DEPRE	Depreciation cost [$/hr]

SUBROUTINE	VARIABLE	DESCRIPTION [UNITS, IF APPLICABLE]
	FARROI	Fare for 15% ROI [$/pass.-mile]
	FCOST	Fuel cost [$/hr]
	FEELAND	Landing fees [$]
	FL	Flight length (lift-off to touchdown)
	IECON	Number of economy seats
	IFIRS	Number of first class seats
	INSUR	Insurance cost [$/hr]
	PASSPHR	Passengers per hour
	PER	Percent of total cost [%]
	PRICE	Purchase price of airplane [$]
	PROFIT	Profit of operations [$]
	REVHR	Revenue hours
	REVYR	Revenue years
	ROI	Return on investment [year^{-1}]
	RPM	Revenue passenger miles
	SERVICE	Servicing cost [$]
	SPARES	Spares cost [$]
	SUPPORT	Support cost [$]
	TAXRT	Tax rate
	TONCAR	Tons of cargo [tons]
	TOT	Total operating cost [$/hr]
	XINVEST	Investment cost [$]
	XIOC	Indirect operating cost [$/hr]
	YRMULT	Year inflation factor
WEIGHT	ACTCON	Weight of active control system [lbs]
	BAGGAGE	Weight of passenger's baggage [lbs]
	CARGO	Cargo weight [lbs]
	CREW	Crew weight [lbs]
	DELWTO	Difference between last gross weight and new gross weight [lbs]
	FUDGE	Weight overrun fudge factor
	FUEL	Mission fuel [gallons]
	LT	Length of tail [ft]
	NOIT	Number of weight iterations
	PASS	Passenger weight [lbs]
	SHT	Horizontal tail volume coefficient
	SVT	Vertical tail volume coefficient
	TRV	Vertical tail taper ratio
	WECTL	Weight of electrical system [lbs]
	WTAC	Air conditioning weight [lbs]
	WTENG	Weight of engine [lbs]
	WTFOOD	Weight of food [lbs]
	WTFSYS	Fuel system weight [lbs]
	WTFUEL	Fuel weight [lbs]
	WTFURN	Weight of furnishings [lbs]

SUBROUTINE	VARIABLE	DESCRIPTION [UNITS, IF APPLICABLE]
	WTFUSE	Fuselage weight [lbs]
	WTHT	Horizontal tail weight [lbs]
	WTINST	Instrument weight [lbs]
	WTLG	Landing gear weight [lbs]
	WTMISC	Miscellaneous weight [lbs]
	WTOZ	Weight of oxygen system [lbs]
	WTSRT	Weight of engine starters [lbs]
	WTVT	Weight of vertical tail [lbs]
	WTWIN	Weight of windows [lbs]
	WTWING	Weight of wing [lbs]
CRUALT	ALF	Standard atmospheric property α, reference 14
	ALT	Cruise altitude [ft]
	ANS	Vector of atmosphere as a function of altitude
	CLCR	Cruise lift coefficient
	DALT	Difference between old altitude guess and new [ft]
	DP	Δ pressure [lbs/ft^2]
	DRDH	$\partial\rho/\partial H$, density gradient [slug/ft^2]
	HNEW	New altitude guess [ft]
	HOLD	Old altitude guess [ft]
	IC	Number of iterations in table look up
	M	Mach number
	P	Pressure [lbs/ft^2]
	PO	Sea level pressure [lbs/ft^2]
	R	Gas constant
	TO	Sea level temperature [^{o}R]
	WT	Cruise weight [lbs]
ENGINE	ALT	Operating altitude [ft]
	DELTA	Pressure ratio, P/P_o
	F	Fuel flow rate [lbs/hr]
	FN	Normalized fuel flow
	HN	Normalized altitude
	TCTM	Cruise thrust over installed thrust
	THETA	Temperature ratio, T/T_o
	TN	Normalized thrust
	TSFC	Thrust specific fuel consumption
CDZL	ABOD	Surface area of fuselage body [ft^2]
	ACONE	Surface area of tail cone [ft^2]
	ANOSE	Surface area of nose cone [ft^2]
	BV	Span of vertical tail [ft]
	CDBF	Fuselage bluff body drag coefficient
	CDFF	Fuselage friction drag coefficient
	CDOE	Engine drag coefficient
	CDOF	Fuselage drag coefficient

SUBROUTINE	VARIABLE	DESCRIPTION [UNITS, IF APPLICABLE]
	CDOH	Horizontal tail drag coefficient
	CDOV	Vertical tail drag coefficient
	CDOW	Wing drag coefficient
	CF	Friction coefficient
	CONEL	Length of tail cone [ft]
	CVBAR	Vertical tail mean aerodynamic chord [ft]
	DIA	Fuselage diameter [ft]
	EDIA	Engine diameter [ft]
	EL	Engine length [ft]
	FLOD	Ratio of fuselage length to diameter
	MU	Viscosity [sec^{-2}]
	SREF	Reference area [ft^2]
STABCOD	AW	Wing lift curve slope [deg^{-1}]
	BETA	Mach number correction factor
	CLAWB	C_{L_α} of wing body
	DCDM	$\partial C_D/\partial M$
	DCMDCL	$C_m/\partial C_L$, static stability
	DEDA	Downwash gradient, $\partial\varepsilon/\partial\alpha$
	DEDAMO	Incompressible downwash gradient, $\left.\frac{\partial\varepsilon}{\partial\alpha}\right\|_{M=0}$
	DXACE	Shift in aerodynamic center due to engines
	ELASTK	Elasticity correction factor
	ETAH	Tail efficiency, η_t
	XAC	Aerodynamic center
	XACDM	Shift in aerodynamic center due to compressibility
	XACW	Wing aerodynamic center
TRIM	ALFAT	Angle-of-attack of tail [deg]
	ALFAW	Angle-of-attack of wing [deg]
	AOL	Angle-of-zero-lift [deg]
	AT	Tail lift curve slope [deg^{-1}]
	AW	Wing lift curve slope [deg^{-1}]
	CLTAIL	Tail lift coefficient
	CLWING	Wing lift coefficient
	CMACW	Wing pitching moment coefficient
	CMFUS	Fuselage pitching moment coefficient
	CNT	Normal tail force coefficient
	CNW	Normal wing force coefficient
	CRDE	Elevator deflection [deg]
	CRIT	Stabilizer deflection [deg]
	CTLPWR	Control power coefficient
	DEDA	Downwash gradient
	EPS	Downwash angle [deg]
	LTLOQ	Tail lift normalized by dynamic pressure
	LTOTOQ	Total lift normalized by dynamic pressure
	LWOQ	Wing lift normalized by dynamic pressure

SUBROUTINE	VARIABLE	DESCRIPTION [UNITS, IF APPLICABLE]
	TAO	Elevator control effectiveness parameter
FUELCAL	BLKSPD	Block speed for mission [knots]
	RANGE	Range of mission [naut-mi]
	WTMID	Mid-point weight [lbs]
	WTRES	Weight after reserve mission [lbs]
	XMG	Miles per gallon of fuel
	XSMG	Seat miles per gallon of fuel
CRUFUEL	ALTCR	Cruise altitude [ft]
	ALTD	Desired altitude [ft]
	ALTEND	Altitude at end of leg [ft]
	ALTLDM	Altitude for maximum glide ratio [ft]
	CLCR	Cruise lift coefficient
	CLM	Lift coefficient for maximum glide ratio
	GAMA	Climb gradient
	IRES	= 1, reserve leg, = 0 otherwise
	LOD	Glide ratio
	ROC	Rate-of-climb, [ft/hr]
	TOWAV	T/W available
	TOWRQ	T/W required
XLOD	CLLDMX	Lift coefficient for maximum glide ratio
	CLO	Design lift coefficient of wing section
	CLWN	Wing lift coefficient normal to leading edge
	EPS	Wing downwash angle at tail [deg]
	SIGMA	Wing-tail interference factor
	TDG	Tail drag contribution
	WMCR	Critical Mach number of wing
	XLDMX	Maximum glide ratio
MAINCST	LCOST(I)	Labor cost of Ith component [$/hr]
	MCOST(I)	Material cost of Ith component [$/hr]
CNSTRN	AAP	Coefficients of longitudinal characteristic polynomial in approach
	ACR	Coefficients of longitudinal characteristic polynomial in cruise
	CLA	Approach lift coefficient
	CLS	Stall lift coefficient with full flaps
	CLTAIL	Required tail lift coefficient
	CLTAV	Available tail lift coefficient for nose gear unstick
	CL2	Second segment climb lift coefficient
	DERAP	Dimensional stability derivatives in approach
	DERCR	Dimensional stability derivatives in cruise
	DGDU	$\partial\gamma/\partial u$

APPENDIX V - cont.

SUBROUTINE	VARIABLE	DESCRIPTION [UNITS, IF APPLICABLE]
	DZETA	Desired short period damping ratio
	ROOTAP	Longitudinal roots in approach
	ROOTCR	Longitudinal roots in cruise
	UNTHRU	Unbalanced thrust component during take-off roll

APPENDIX VI - MAP OF COMMON BLOCKS USED WITHIN DESIGN SECTION OF PROGRAM

COMMON BLOCK	ARRAY	NO.	DESCRIPTION [UNITS, IF APPLICABLE]
DEVAR	CST	1	Direct operating cost per block hour [$/hr]
		2	Direct operating cost per flight [$/flight]
		3	Return on investiment [$year^{-1}$]
		4	Fare for a ROI of .15 [$/seat-naut. mi]
		5	Fuel efficiency [seat-naut. mi/gal]
		6	Maximum glide ratio in cruise
		7	Maximum take-off gross wt. [lbs]
	DESIGN	1	Wing area [ft^2]
		2	Wing aspect ratio
		3	Fuselage length [ft]
		4	Horizontal tail area [ft^2]
		5	Horizontal tail aspect ratio
		6	Installed thrust [lbs]
		7	Aft most center-of-gravity in cruise [% MAC]
	ITERM	1	= 1 if active controls,=0 otherwise
		2	= 1 if center-of-gravity control,=0 otherwise
		3	Cruise C.G. Selector: = 7 if midway between FWD & AFT limits
		4	Number of engines
		5	Element of CST used in optimizer
		6	Year of evaluation
		7	= 1 if landing gear is movable,=0 otherwise
DRAG	CDS	1	Cruise Total parasite drag coefficient
		2	Cruise Wing drag coefficient
		3	Cruise Horizontal tail drag coefficient
		4	Cruise Glide ratio
		5	Cruise Total drag coefficient
		6	Cruise Fuselage drag coefficient

COMMON BLOCK	ARRAY	NO.	DESCRIPTION [UNITS, IF APPLICABLE]
	CDSAP	1	Approach Total parasite drag coefficient with take-off flaps
		2	Approach Wing drag coefficient
		3	Approach Horizontal tail drag coefficient
		4	Approach Glide ratio with landing flaps
		5	Approach Total drag coefficient
		6	Approach Fuselage drag coefficient
GEOM	GX	1	PI (π) = 3.1415927
		2	Radian to degree conversion factor = 57.295779
		3	Cruise Mach number
		4	Design range [naut. miles]
		5	Fuselage diameter [ft]
		6	Reference engine length [ft]
		7	Reference engine diameter [ft]
		8	Maximum wing lift coefficient with flaps in take-off position
		9	Trimmed lift coefficient in cruise
		10	Trimmed lift coefficient
		11	Characteristic turbulence length divided by $\sqrt{3}$ [ft]
		12	Maximum wing lift coefficient with full flaps
		13	Variance of elevator deflection in approach [rad]
		14	Velocity in approach [ft/sec]
		15	Acceleration constant, g [ft/sec^2]
		16	Take-off stabilizer position, [deg], if -99 then use trim position for climbout
		17	Angle of zero lift in take-off [deg]
		18	Angle of zero lift with landing flaps [deg]
		19	Number of passenger seats
		20	Reference engine weight [lbs]
		21	Thrust of reference engine [lbs]
		22	Weight of cargo [lbs]
		23	Allowable C.G. range [% MAC]
		24	Passenger load factor
		25	Fare [$/seat-mile]
		26	Airplane utilization [hrs/yr]
		27	Ratio of maximum landing weight to gross weight
		28	Wing lift coefficient in cruise
		29	Wing lift coefficient in approach
		30	Minimum ratio of thrust available to thrust required in cruise
		31	Trim drag correction for estimating L/D_{max}
		32	Sweep of horizontal tail [deg]
		33	Distance between horizontal tail and wing [ft]

COMMON BLOCK	ARRAY	NO.	DESCRIPTION [UNITS, IF APPLICABLE]
		34	Height of horizontal tail above fuselage centerline [ft]
		35	Vertical tail area [ft^2]
	HX	1	Horizontal tail taper ratio
		2	Horizontal tail thickness ratio
		3	Elevator control effectiveness factor
		4	Horizontal tail span [ft]
		5	Horizontal tail mean aerodynamic chord [ft]
		6	Horizontal tail volume coefficient
		7	Tan (tail quarter chord sweep angle)
		8	Tan (tail leading edge sweep angle)
		9	Tan (tail half chord sweep angle)
		10	Elevator servo time constant [sec]
		11	Approach tail lift coefficient
		12	Tail incidence in approach [deg]
		13	Trimmed elevator position in approach [deg]
		14	Trimmed elevator position in cruise [deg]
		15	Variance of elevator in cruise [rad]
		16	Vertical tail volume coefficient
		17	Vertical tail taper ratio
		18	Vertical tail aspect ratio
		19	Vertical tail sweep at quarter chord [deg]
		20	Ratio of rudder area to vertical tail area
	W	1	Wing sweep at quarter chord [deg]
		2	Incidence of wing [deg]
		3	Wing taper ratio
		4	Wing thickness ratio
		5	Wing geometric twist [deg]
		6	Oswald's efficiency factor in drag bucket
		7	Osward's efficiency factor out of drag bucket
		8	Design lift coefficient for wing section
		9	Wing span [ft]
		10	Wing mean aerodynamic chord, MAC [ft]
		11	Tan (wing quarter chord sweep angle)
		12	Tan (wing leading edge sweep angle)
		13	Tan (wing half chord sweep angle)
		14	Wing pitching moment coefficient at cruise
		15	Cos (wing quarter chord sweep angle)
		16	Wing angle-of-zero-lift with no flaps [deg]
		17	Increase in wing pitching moment coefficient at take-off
		18	Increase in wing pitching moment coefficient at landing

COMMON BLOCK	ARRAY	NO.	DESCRIPTION [UNITS, IF APPLICABLE]
		19	Wing pitching moment coefficient at approach
		20	Increase in drag due to flap deflection
	PX	1	Pitching moment reduction coefficient
		2	TSFC reduction coefficient
		3	Weight reduction coefficient
		4	Drag reduction coefficient
		5	Disconnect rubber engine (1.0 = yes; 0.0 = no)
		6	Maintenance cost boost (% Boost = PX(6) * 5%)
		7	Purchase price boost (% Boost = PX(7) * 5%)
		8	Maneuver load alleviation (Reduction in design limit load in g's)
GRAVITY	CGS	1	Aft cruise C.G. position [% MAC]
		2	Aft approach C.G. position [% MAC]
		3	Forward cruise C.G. position [% MAC]
		4	Forward approach C.G. powition [%MAC]
		5	Landing gear position [% MAC]
		6	Required distance between C.G. and landing gear [% MAC]
STAB	DERIVAP	1	Approach C_{L_α}
		2	C_{D_α}
		3	C_{M_α}
		4	C_{L_u}
		5	C_{D_u}
		6	C_{M_u}
		7	C_{L_q}
		8	C_{D_q}
		9	C_{M_q}
		10	$C_{L_{\dot\alpha}}$
		11	$C_{D_{\dot\alpha}}$
		12	$C_{M_{\dot\alpha}}$

COMMON BLOCK	ARRAY	NO.	DESCRIPTION [UNITS, IF APPLICABLE]	
		13		C_{L_δ}
		14		C_{D_δ}
		15		C_{M_δ}
	DERIVCR	1	Cruise	C_{L_α}
		2		C_{D_α}
		3		C_{M_α}
		4		C_{L_u}
		5		C_{D_u}
		6		C_{M_u}
		7		C_{L_q}
		8		C_{D_q}
		9		C_{M_q}
		10		$C_{L_{\dot\alpha}}$
		11		$C_{D_{\dot\alpha}}$
		12		$C_{M_{\dot\alpha}}$
		13		C_{L_δ}
		14		C_{D_δ}
		15		C_{M_δ}
	STOR	1	Cruise	C_{L_α} wing
		2		C_{L_α} horizontal tail
		3		$\partial\varepsilon/\partial\alpha$
		4		Stick fixed neutral point [% MAC]
		5		Stick fixed maneuver point [% MAC]

COMMON BLOCK	ARRAY	NO.	DESCRIPTION [UNITS, IF APPLICABLE]
		6	Approach C_{L_α} wing
		7	C_{L_α} horizontal tail
		8	$\partial\varepsilon/\partial\alpha$
		9	Stick fixed neutral point [% MAC]
		10	Stick fixed maneuver point [% MAC]
		11	C_{mac} fuselage
		12	C_{mac} engine
WTSVE	WTS	1	Maximum gross weight at take-off [lbs]
		2	Empty weight [lbs]
		3	Fuel flow rate [lbs/block hour]
		4	Manufacturers airframe weight [lbs]
		5	Landing weight after mission [lbs]
		6	Fuel weight including reserves [lbs]
		7	Aircraft specific density (cruise), $\frac{m}{S\bar{c}}$
		8	Radius of gyration squared (cruise), Iy/m [ft]
		9	Aircraft specific density (approach), $\frac{m}{S\bar{c}}$
		10	Radius of gyration squares (approach), Iy/m [ft]
		11	Altitude at mid cruise [ft]
		12	L/D_{max} at cruise altitude
		13	Pitch moment of inertia (cruise), [slug-ft]
		14	Pitch moment of inertia (approach), [slug-ft]
		15	Weight at mid cruise [lbs]
		16	Fuel cost [$/gal]
		17	Cruise velocity [knots]
		18	Flight time for mission [hrs]
		19	Block time [hrs]
		20	Weight of fuel to fly economic mission [lbs]

APPENDIX VII - ASSUMPTIONS USED IN CALCULATING TRANSPORT DESIGN FACTORS

MISSION:	
Cruise Mach Number	.80
Divergence Mach Number	.84
Design Range	6500 km
Number of Seats	200
Cargo	33400 N
Maximum Lift Coefficient	3.15
Landing Field Requirement	2440 m
Take-Off Field Requirement	3050 m
GEOMETRY:	
Wing Sweep Angle	26.4 deg
Wing Thickness Ratio	.12
Wing Taper Ratio	.38
Wing Incidence Angle	2 deg
Wing Geometric Twist	5 deg
Tail Thickness Ratio	.10
Tail Sweep Angle	30 deg
Tail Taper Ratio	.4
Vertical Tail Sweep	35 deg
Ratio of Rudder Area to Vertical Tail Area	.30
Ratio of Elevator Chord to Horizontal Tail Chord	.25
Ratio of Flap Span to Wing Span	.6
Maximum Flap Deflection	45 deg
Fuselage Diameter	5.08 m
Height of Aerodynamic Center Above c.g.	.08 MAC
Height of Thrust Vector Above c.g.	-.12 MAC
Height of Horizontal Tail Above c.g.	0
Number of Engines	2
ECONOMICS:	
Fuel Cost	20¢/liter
Load Factor	.55
Fare	9¢/seat-naut. mi
Utilization Rate	3200 hr/yr
Depreciation Period	14 yr
Residual Value	12 percent
Tax Rate	.48
Year of Study	1979
Assumed Annual Inflation Rate	.07
Number of Prototype Aircraft	2

Aircraft Fleet Size	250
Initial Production Rate	.5/month
Full Production Rate	5/month
Engineering Rate	19.55 '74 $/hr
Tooling Rate	14.00 '74 $/hr
Labor Rate	10.90 '74 $/hr
Engines for Test Aircraft	3
Ratio of Manufacturer's Airframe Weight to Take-Off Wt.	.75

MISCELLANEOUS:

Maximum Dynamic Pressure	5.13 N/m^2
Pressurized Volume	178.2 m^3
Number of Pilots	3
Number of Attendants	8
Air Conditioning Flow Rate	200 kg/min
Autopilot Channels (w/MUX)	5
General Capacity	750 kilovolt-amperes
Maintenance Complexity Factor	1.6
Hydraulics Volume Flow Rate	300 liters/min
Number of Inertial Platform Systems	1
Ratio of APU-on Time to Engine on Time	.1
Curved Windshield	
Ratio of First Class to Economy Seating	.15
Maximum Speed	483 knots
Supercritical Airfoil Technology	
Airfoil Design Lift Coefficient	.5
Some Nonlinear Aerodynamics Terms	
Baseline Engine	CF-6
Elevator Servo Time Constant	.1 sec

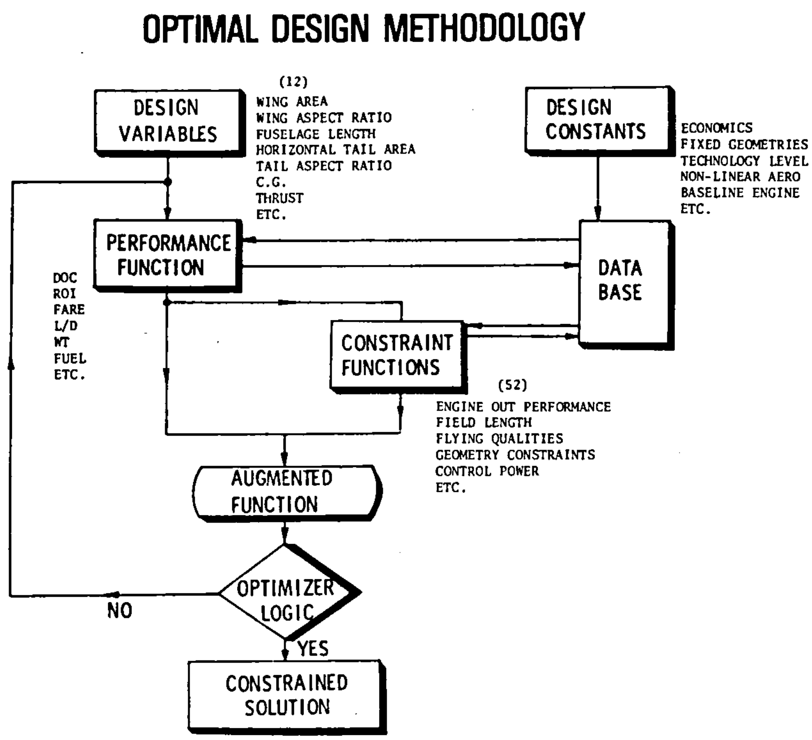

Figure 1.- Generalized flow diagram for OPDOT.

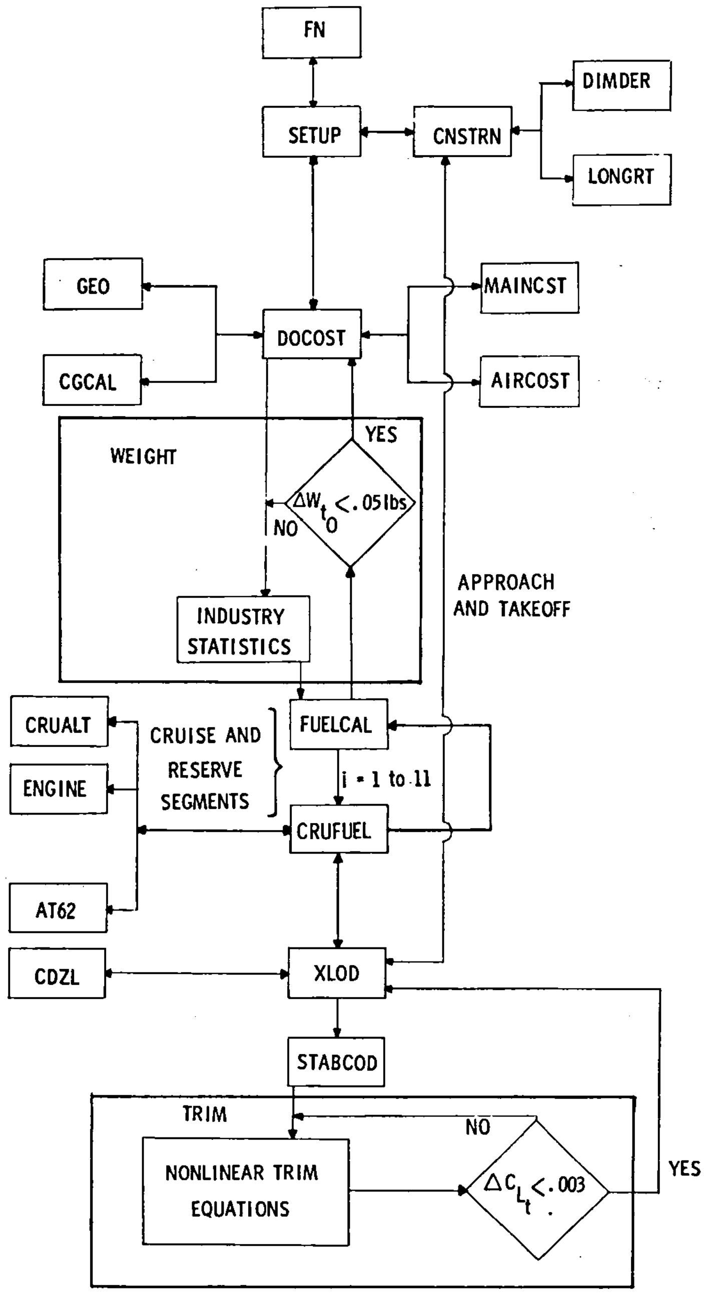

Figure 2.- Schematic showing primary calling sequence of subroutines used to evaluate the performance index and constraint functions.

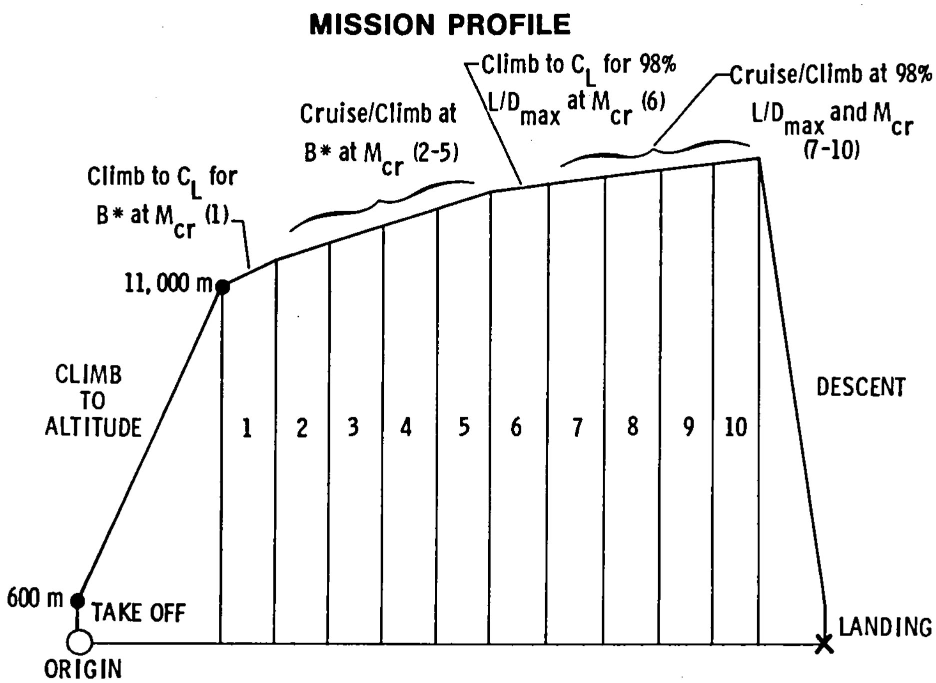

Figure 3.- Mission profile used in OPDOT.

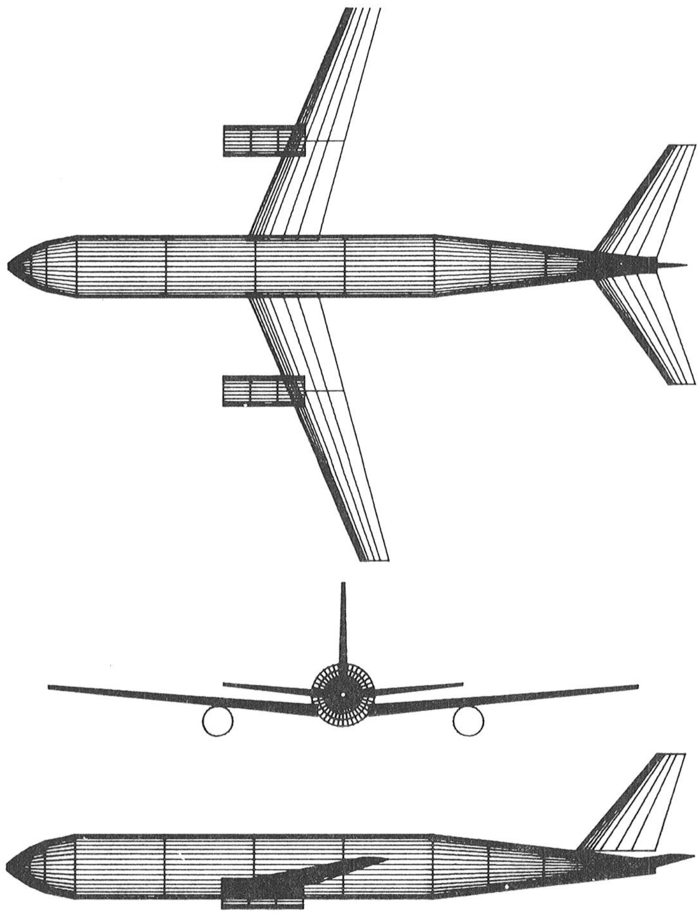

Figure 4.- Aircraft "picture" as drawn by the method of reference 28.
Aircraft pictured was optimized from data in Appendix II.

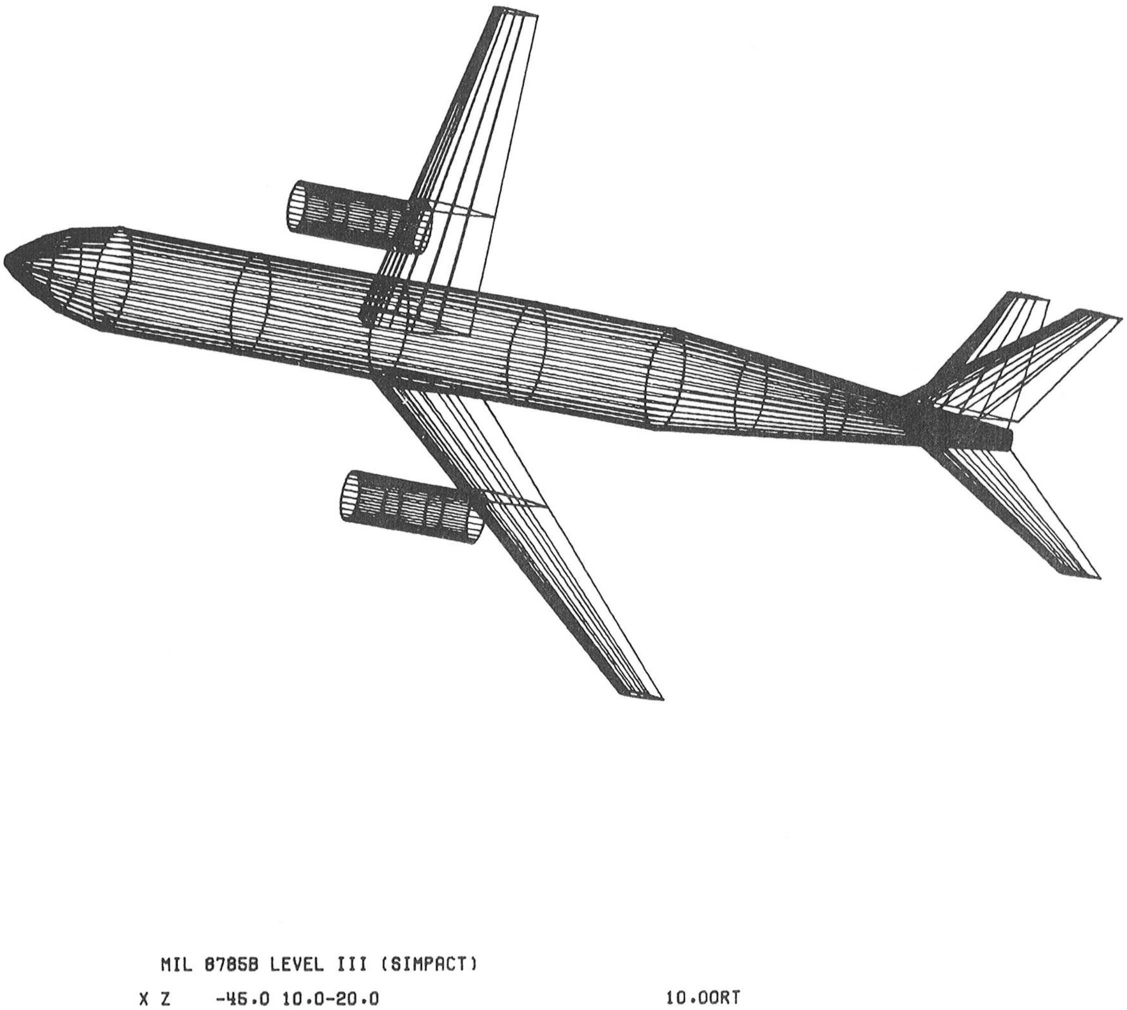

Figure 4.- Concluded.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-81857		

4. Title and Subtitle	5. Report Date
OPDOT: A Computer Program for the Optimum Preliminary Design of a Transport Airplane	September 1980
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Steven M. Sliwa and P. Douglas Arbuckle	
	10. Work Unit No.
	505-34-33-05
9. Performing Organization Name and Address	11. Contract or Grant No.
NASA Langley Research Center Hampton, VA 23665	
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A description of a computer program, OPDOT, for the optimal preliminary design of transport aircraft is given. OPDOT utilizes constrained parameter optimization to minimize a performance index (e.g. direct operating cost per block hour) while satisfying operating constraints. The approach in OPDOT uses geometric descriptors as independent design variables. The independent design variables are systematically iterated to find the optimum design. The technical development of the program is provided and a program listing with sample input and output are utilized to illustrate its use in preliminary design. This is not meant to be a user's guide, but rather a description of a useful design tool developed for studying the application of new technologies to transport airplanes.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
airplane design airplane performance non-linear programing constrained parameter optimization	[redacted] Subject Category 05

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	106	

N-305 *Available: NASA's Industrial Applications Centers