(NASA-CR-169027) USE OF OPTIMIZATION TO PREDICT THE EFFECT OF SELECTED PARAMETERS ON COMMUTER AIRCRAFT PERFORMANCE Final Report (Stanford Univ.) 61 p HC A04/MF A01 CSCL 01C G3/05
N82-26279
Unclas
28083

**Department of AERONAUTICS and ASTRONAUTICS**
**STANFORD UNIVERSITY**

SUDAAR 534

# USE OF OPTIMIZATION TO PREDICT THE EFFECT OF SELECTED PARAMETERS ON COMMUTER AIRCRAFT PERFORMANCE

by

Valana L. Wells and Richard S. Shevell

This Research Was Supported By
NASA Langley Research Center
Hampton, Virginia 23665
Grant NAG 1-202

June 1982

# USE OF OPTIMIZATION TO PREDICT THE EFFECT OF SELECTED PARAMETERS ON COMMUTER AIRCRAFT PERFORMANCE

## Final Report for
## NASA Research Grant NAG-1-202

Valana L. Wells and Richard S. Shevell

Department of Aeronautics and Astronautics
Stanford University
Stanford, California 94305

June 1982

PRECEDING PAGE BLANK NOT FILMED

## Abstract

An optimizing computer program, developed as part of this study, determined the turboprop aircraft with lowest direct operating cost for various sets of cruise speed and field length constraints. External variables included wing area, wing aspect ratio and engine sea level static horsepower; tail sizes, climb speed and cruise altitude were varied within the function evaluation program. Direct operating cost was minimized for a 150 n. mi typical mission. Generally, DOC increased with increasing speed and decreasing field length but not by a large amount. Ride roughness, however, increased considerably as speed became higher and field length became shorter.

# TABLE OF CONTENTS

# I. Introduction

The resurgence of interest in small, propeller-driven aircraft has sparked renewed analysis of the aerodynamics, structures and propulsion systems of such planes. Along with advanced technology research, which is the bent of much of the recent concern, there remains a need for the answer to a, perhaps, more basic question—that is, for what mission should this airplane be designed? The "mission" includes not just stage length (which is determined by the actual leg distances flown by commuter airlines) but also the speed at which to climb and cruise and the field length from which the aircraft must takeoff and land.

This study, rather than seeking to prescribe a particular design or mission, discovers the relationships between field length and cruise speed and aircraft direct operating cost. To do this, a gradient optimizing computer program was developed to minimize direct operating cost (DOC) as a function of airplane geometry. In this way, one can compare the best airplane operating under one set of constraints with the best operating under another. Best, in this case, means having the minimum DOC.

To compare different airplanes, one can make use of relatively simple techniques for some parameter estimation. For example, a complete stability and control analysis for tail size determination is superfluous for preliminary design when statistical correlations of tail sizes with wing and fuselage characteristics exist for similar airplanes. Thus several such statistical correlations methods appear in the program. However, one must also use more sophisticated procedures when a high degree of accuracy is required or when the particular calculation may have a major influence on the performance index. The program, therefore, has extensive and detailed routines for drag, climb, range and other critical values.

For this study a constant 30-passenger fuselage and "rubberized" engines based on the General Electric CT-7 were used as a baseline. All aircraft had to have a 600 nautical mile maximum range and were designed to FAR part 25 structural integrity and climb gradient regulations. Direct operating cost was minimized for a typical design mission of 150 nautical miles. For purposes of $C_{L_{max}}$ calculation, all aircraft had double-slotted flaps but with no Fowler action.

## II. Procedure

### A. The Optimizer

The optimizer minimizes direct operating cost as a function of wing area, aspect ratio and engine sea-level static horsepower rating through use of a variable metric algorithm which is, in fact, a quasi-Newton's method. A true Newton's method utilizes the following strategy for size and direction of step:

$$\vec{x}_{j+1} = \vec{x}_j - H_j^{-1} \vec{g}_j$$

where $\vec{x}$ represents the vector of variables, $H_j$ is the Hessian (matrix of second derivatives) at step $j$, and $\vec{g}_j$ is the gradient vector at step $j$. In the absence of second derivative information, a numerical approximation of the Hessian using known values of the first derivatives provides an adequate substitute. The variable metric method follows exactly this procedure.

Of course, for such a complicated function as the one in this study (the "function" is a thirty page FORTRAN program), even first derivatives do not exist in closed form. Thus, the program must calculate a gradient estimate using a forward difference approximation. The differencing step size is constrained to be rather large (one percent of the variable value) since noise in the function evaluation leads to incorrect gradients for small steps.

The variable metric method solves unconstrained problems only. Thus, in order to account for the inequality constraints which must hold in order for the aircraft to meet such mission requirements as maximum takeoff distance, minimum engine-out climb gradient, etc., the program uses what is termed the "penalty function" or "soft constraint" approach. In a mathematical sense, this method changes the problem to an unconstrained one by including the constraints in the goal function. The goal function becomes,

$$\text{GOAL} = \text{DOC} + K|\text{constraint value} - \text{constraint value required}|$$

where DOC = direct operating cost
K = penalty coefficient

$$= \begin{cases} 0 & \text{if constraint is met} \\ \text{large} & \text{if constraint is not met.} \end{cases}$$

A penalty is added to the goal function for each of the five inequality constraints:

- takeoff distance
- landing distance
- available cruise power
- second segment climb gradient
- enroute climb gradient

## B. The Function Evaluation

The function evaluation program, which comprises the bulk of the calculations involved in the optimization, acts as a mathematical aircraft model. This routine determines, for a prescribed wing area, sea level static horsepower rating, and wing aspect ratio, the complete geometry, performance, and operating cost of the resulting airplane. For simplicity, it employs preliminary design methodology for estimating such parameters as zero-lift equivalent drag area, tail sizes, $C_{L_{max}}$, and airplane efficiency factor. The direct operating cost calculation is based on the 1967 ATA DOC method with corrections for inflation and commuter operation. The following outline briefly describes the function evaluation scheme.

### 1. Airplane Geometry and Drag Parameters

In order to compute the airplane geometry (wing span, wing mean aerodynamic chord, vertical and horizontal tail areas) the program assumes as constants:

| | |
|---|---|
| wing average thickness ratio | .15 |
| tail average thickness ratio | .1 |
| wing taper ratio | .4 |
| horizontal tail aspect ratio | 4.0 |
| vertical tail aspect ratio | 1.8 |
| wing sweep | 0 |
| tail surface sweep angle | $20^{\circ}$ |

To avoid a complex iteration involving weight and balance, the horizontal and vertical tail lengths are estimated as 32 ft. and 30 ft., respectively, and a center of gravity range of 25% of the wing mac is allowed. Using these estimates, the program calculates tail areas as a function of fuselage and wing sizes according to ref. 1.

Once all surface areas are known, the program computes the zero-lift equivalent drag area, f. The formula for f of a component has the form

$$f_i = C_{f_i} K_i Swet_i$$

where $C_f$ = friction coefficient; function of Reynolds number
$K$ = form factor; function of fineness ratio or thickness ratio
$Swet$ = component wetted area
$i$ refers to the ith component such as wing, fuselage, nacelle, etc.

A summation of all component drag areas, plus a 6% addition for miscellaneous components, gives the total airplane equivalent parasite drag area:

$$f = \sum_i f_i / .94$$

The zero-lift or parasite drag coefficient, $C_{D_P}$, is just: $C_{D_P} = f/Sw$, Sw = wing reference area.

The program computes airplane efficiency factor, e, from:

$$e = \frac{1}{\pi \mathit{AR} \left( \frac{1}{\pi \mathit{AR}\, us} + .43\, C_{D_P} \right)}$$

where $u$ = induced drag factor due to planform; function of $\mathit{AR}$, taper ratio, sweep.
$s$ = induced drag factor due to fuselage interferences function of wing span/fuselage diameter.

Inclusion of $C_{D_p}$ in this formula accounts for the increase in profile drag with angle of attack.

## 2. Range and Maximum Takeoff Weight

For any combination of wing area, sea level horsepower, and wing aspect ratio (other possible variables assumed constant) there exists a takeoff weight necessary to travel a given distance at a given speed. This routine determines that takeoff weight required for the airplane described by those three variables to cover a maximum range of 600 N mi. at a prescribed cruising speed. The takeoff weight depends rather heavily on two other variables - cruise altitude and climb speed. Thus, in order to include these as variables, the program performs a two dimensional grid search on altitude and climb speed and saves the combination of the two which uses the least fuel to complete the 600 N mi. mission.

Determining the maximum takeoff weight is an iterative procedure completed through the use of a one dimensional minimization routine. The minimizer employs a "linear search with parabolic inverse interpolation" with the goal function defined as the square of the difference between the actual range and the desired range.

The range calculation itself has four major parts:

(a) calculation of empty weight
(b) climb
(c) descent
(d) cruise

The weight is calculated using a statistical method based on data from large and small commercial aircraft, (ref. 1).

The time, fuel, and distance to climb are calculated according to:

$$\text{time to climb} = \int_{h_{min}}^{h_{max}} \frac{dh}{R/c}$$

$$\text{fuel to climb} = \int_{h_{min}}^{h_{max}} \frac{SHP * SFC}{3600 * R/C}\, dh$$

$$\text{distance to climb} = \int_{h_{min}}^{h_{max}} \frac{V}{R/C}\, dh$$

where R/C = rate of climb in ft/sec
h = altitude in ft
SHP = shaft power in horsepower
SFC = specific fuel consumption in lb/SHP-hr
V = true airpseed in ft/s

The climb routine numerically evaluates these integrals making the following assumptions:

- climb at constant equivalent airspeed
- climb at maximum continuous power
- SFC constant at maximum continuous power

The numerical integration uses a forward Euler technique and an altitude step size of 200 ft.

The descent method assumes:

- descent at constant equivalent airspeed
- descent at constant rate of descent
- idle (minimum) power at 10% of maximum power

The aircraft rate of descent corresponds to a 300 feet per minute cabin pressure descent where the cabin has a 6000 foot pressure altitude in cruise. The program computes fuel and distance to descend using the following integrals:

$$\text{fuel to descend} = \int_{h_{min}}^{h_{max}} \frac{SHP * SFC}{3600 * R/D}\, dh$$

$$\text{distance to descend} = \int_{hmin}^{hmax} \frac{V}{R/D}\, dh$$

where h = altitude

SFC = specific fuel consumption

R/D = rate of descent, ft/sec

V = descent true airspeed, ft/sec

SHP = shaft power

Since the airplane descends at constant equivalent airspeed and constant rate of descent, the distance integral becomes:

$$\text{distance to descend} = \frac{V_E}{R/D} \int_{hmin}^{hmax} (1 - 6.8634 \times 10^{-6} h)^{-2.1324}\, dh$$

where $V_E$ = equivalent airspeed for descent

$V = V_E\ (1 - 6.8634 \times 10^{-6} h)^{-2.1324}$

Integrating gives:

$$\text{distance to descend} = \frac{V_E}{R/D}\left(\frac{1}{6.8634 * 1.1324 * 10^{-6}}\right)$$

$$\times \left[(1 - 6.8634 \times 10^{-6}\ h)^{-1.1324}\right|_{hmin}^{hmax}$$

Letting hmin = 0,

$$\text{distance to descend} = \frac{V_E}{R/D}\left(\frac{1}{6.8634 * 1.1324 * 10^{-6}}\right)$$

$$\times \left[(1 - 6.8634 * 10^{-6}\ hmax)^{-1.1324} - 1\right]$$

Fuel to descend is numerically calculated using an explicit Euler integration and an altitude step size of 500 ft. The fuel integral begins at the end of the descent and integrates backwards until the aircraft reaches the cruising altitude. The weight at the bottom of descent is the zero fuel weight plus additional fuel weight for an appropriate reserve mission (100 n. mi. at best specific range plus 45 minutes at best endurance).

The distance covered in the cruising portion of the mission depends on the weights at the end of climb and at the top of descent. For propeller-driven aircraft,

$$R = \int_{Wf}^{Wi} 325 \frac{\eta}{SFC} \frac{dW}{D}$$

where $\eta$ = propeller efficiency in cruise
$D$ = drag
$Wi$ = weight at beginning of cruise
$Wf$ = weight at end of cruise
$SFC$ = specific fuel consumption
$R$ = range, n. miles

Letting SFC be approximately constant and equal to the average SFC during cruise, and noting that

$$D = C_{D_p} q S + \frac{W^2}{q \pi b^2 e} ,$$

and letting

$$A1 = C_{D_p} q S$$

$$A2 = \frac{1}{q \pi b^2 e} ,$$

the integral becomes,

$$R = 325 \frac{\eta}{SFC} \int_{Wf}^{Wi} \frac{dW}{A1 + A2\, W^2}$$

for constant cruise speed and cruise altitude. Integrating gives:

$$R = 325 \frac{\eta}{SFC} \frac{1}{\sqrt{A1\ A2}} \left[ \tan^{-1} \frac{W}{\sqrt{A1/A2}} \right]_{Wf}^{Wi}$$

or

$$R = 325 \frac{\eta}{SFC}\, b \sqrt{\frac{\pi e}{C_{D_p}\, S}} \left[ \tan^{-1} \frac{W}{qb\sqrt{C_{D_p}\, \pi e\, S}} \right]_{Wf}^{Wi}$$

This formula holds only for the case of constant dynamic pressure, $q$. Since the commuter cruises at constant speed and altitude, it satisfies the condition of invariant $q$.

3. Evaluation of Constraint Parameters

Five acceptability criteria constrain the aircraft design.

(a) Maximum Cruise Thrust

To fly at the prescribed cruising speed, the maximum thrust produced by the engines must equal or exceed the cruise drag. Maximum thrust depends on maximum cruise power according to the relation

$$TH = 550\, \eta \frac{SHP}{V}$$

where $\eta$ = propeller efficiency
$V$ = true airspeed, ft/sec
$TH$ = thrust, lb.

Maximum cruise shaft horsepower is determined as a function of airspeed, cruise altitude, and static sea level power rating. Power calculations are based on the General Electric CT-7 turboprop engine.

### (b) Takeoff Distance

Allowed takeoff distances range from 3500 feet to 4500 feet. FAR takeoff field lengths depend on the parameter:

$$\frac{TOW^2}{\sigma C_{L_{max}} S_w TH}$$

where $\sigma$ = $\sqrt{\rho/\rho_o}$

$S_w$ = reference wing area, $ft^2$

Takeoff distance is calculated for a hot day (ISA + 30.8° F) sea level.

### (c) Landing Distance

The allowed FAR landing distances range from 3500 feet to 4500 feet, and they depend on the square of the airplane's stalling speed. Since commuter airplanes do not usually have the ability to jettison fuel, the studied aircraft must land at their takeoff weights.

### (d) Second Segment Climb Gradient

To comply with the Federal Air Regulation, part 25, a twin-engined airplane must have a second segment climb gradient of 2.4%. The gradient is computed for hot day conditions (ISA + 30.8° F) at takeoff power and with one engine inoperative. The drag in this configuration includes that due to a feathered propeller, due to excess rudder deflection as a consequence of asymmetric thrust, and due to a 25 degree takeoff flap deflection.

### (e) Enroute Climb Gradient

According to FAR part 25, a twin-engined airplane must have a one engine-out enroute climb gradient of 1.1%. Since speed for best climb gradient for aircraft of this type is less than the minimum allowable speed, enroute climb gradient is computed at 1.3 times the stalling speed in the clean configuration. The obstacle clearance height used is 11,000 feet.

### 4. Typical Mission

To optimize the commuter design with respect to operating cost, one must compute DOC (direct operating cost) for a typical shorthaul mission. Using a characteristic stage length of 150 N mi., the program finds the corresponding block fuel and block time for the airplane designed to meet the restrictions outlined above. Such values as takeoff weight necessary to meet the range are calculated according to the method described in the "Range and Maximum Takeoff Weight" section.

The direct operating cost routine assumes that commuter pilot pay rates are about one-third that of trunk carrier pilots. It also uses the following cost estimates:

| | |
|---|---|
| Labor Rate | \$12/hr |
| Airframe First Cost | \$200/lb of airframe |
| Engine First Cost | Taken from Ref. 4; inflated 25% |
| Fuel Cost | \$1.50/gallon |
| Oil Cost | \$10/lb |

The cost calculation proceeds as suggested in Ref. 2. Appendix I contains a complete listing of the program.

## C. Ride Roughness

Though ride-roughness was not considered in the optimization portion of the study, a relative ride-roughness parameter was computed for each optimum airplane. This parameter, taken from the FAR gust-response/structures regulations is given by

$$\Delta n = K \frac{a \, U_{de} \, Ve}{498(W/S)}$$

where $U_{de}$ = equivalent gust velocity, ft/s

$Ve$ = equivalent speed, knots

$W/S$ = wing loading, $lb/ft^2$

$a$ = $dC_L/d\alpha$

$$K = \frac{.88\mu}{5.3 + \mu}$$

$$\mu = \frac{2(W/S)}{\rho c a g}$$

Reference 7 provides further discussion of this parameter.

## III. Results

The results of the optimization program show that the airplane with the lowest direct operating cost flies at 290 knot TAS with an allowed field length greater than or equal to 4,060 feet, Figure 1. For field lengths less than 3,650 feet, the 250 knot airplane fares best in terms of DOC as the large wings required for short landing distances cause excessive drag at the higher speeds. At greater than 3,650 foot field lengths, 290 knots is the best speed. The best 330 knot airplane, however, with a landing distance of 4,275 feet has only one percent worse direct operating cost than the best airplane overall. Direct operating cost as a function of field length and cruise speed is presented in Figure 1.

The optimization, aside from determining the effect of cruise speed and field length on DOC, produced the following crucial results:

### A. Critical Field Lengths

Although, generally, direct operating cost decreases with increasing field length (for a given speed), for each speed there exists a critical field length beyond which there is no further improvement in DOC; the field length constraint becomes non-active. Two factors contribute to this phenomenon. First, though the wing area can decrease with increased takeoff or landing distance, the aircraft must still maintain a span adequate to meet climb gradient standards. The resulting increase in aspect ratio increases the weight enough to counteract the beneficial effects of the lower wing area. Secondly, a smaller wing area forces the aircraft to an inefficient $C_L$ far from that for best L/D (which indicates best specific range for propeller-driven aircraft). A drop in cruise altitude improves the $C_L$ but increases the non-lift dependent drag so the altitude modification is not worthwhile.

## B. Active Constraints and Optimal Variable Values

A rough rule of thumb governing the selection of aircraft geometry states that the landing field length requirement determines the wing area and the other operative constraint, whichever one it is, fixes the proper combination of aspect ratio (span) and engine power. In fact, though wing area is not quite independent of cruise speed for a given field length, wing loading (takeoff weight divided by wing area) does not vary with speed. Thus the landing distance has only secondary effect on aspect ratio and horsepower required.

Table 1 presents a list of the active constraints—that is, those limiting the design—for each cruise speed and field length tested. The table includes the critical field length for each speed. At the lower speeds, the required enroute climb gradient sizes the aspect ratio and engine power Since, previously, commuter aircraft have not been designed to meet FAR part 25 regulations, they have not encountered as much difficulty with the one-engine-out enroute climb restriction. Though enroute climb rarely presents a problem for turbofan aircraft, the turboprop airplane, because its speed for best climb is lower than the minimum allowable speed (30% above the stall speed), is often restricted by this regulation if it is designed according to part 25 rules.

At the highest cruise speed, in most cases, minimum cruise power to fly at 330 knot determines both engine power and aspect ratio. Obviously, increasing the horsepower increases the maximum cruise speed, but, though not as important a factor in the power-restricted cases, increasing the aspect ratio also increases the maximum cruise speed due to the reduced induced drag. So, whether the second active constraint is minimum enroute climb gradient or power to cruise at a given cruise velocity, several combinations of aspect ratio and engine power exist to satisfy that constraint. The optimizer chooses the best, or lowest cost, combination of the two.

At a cruise speed of 330 knot and landing distance 4,275 feet or more, enroute climb gradient rather than available cruise power becomes the second operational constraint. This occurs because the wing area has decreased enough that the cruise drag (and, therefore, cruise power required) has also decreased to the extent that power to climb is greater than the power to maintain a 330 knot cruise speed.

Figure 2 shows the variations of optimal wing area, aspect ratio, and horsepower with field length and cruise velocity. As expected, wing area decreases as the field length gets longer. The aspect ratio, however, increases in an attempt to keep the same span in order to maintain the same climb gradient or induced drag. The 250 knot airplanes have higher aspect ratios than the 290 knot planes because they must meet identical climb gradients but with lower power levels. The slower airplanes have lower power ratings but higher spans than the 290 knot aircraft. The 330 knot airplanes have aspect ratios lying between those of the other two speed aircraft since the cruise speed constraint affects choice of aspect ratio differently from the enroute climb constraint.

Figure 2c provides an interesting insight into the effects of differing active constraints on optimum engine power. As wing areas decrease with increasing field length, the aspect ratios increase but, in general, not enough to maintain constant span. If enroute climb is critical, then, the engine power must increase for the airplane to meet the climb gradient for reduced span. At 250 knot and 290 knot this indeed happens. However, if meeting the required cruise velocity is critical, the smaller wing area reduces the parasite drag much more than the smaller span increases induced drag. Therefore, the aircraft requires less power to overcome the cruise drag, and the curve indicating a 330 knot aircraft follows this trend.

## C. Sensitivity Studies

1. Grid Search About an Optimal Point. Although the optimizing program chooses a lowest-cost airplane for a given set of constraint parameters, it gives little information about the effects of small changes in variable values about that optimum. Figures 3a-c show cost for values of wing area, aspect ratio, and engine power above and below those calculated as the optimum for cruise speed equal to 330 knot and a field length of 4,000 feet. Constraint barriers are included in these figures to indicate areas of impossible choices. At the smallest wing area (345 $ft^2$) no airplane can meet the 4,000 foot field length constraint whereas, at a wing area of 385 $ft^2$, all airplanes easily fall below the field length requirement.

As these figures illustrate, the optimizer chooses the lowest cost configuration which can meet all requirements. At the optimum point, the design is bounded by both cruise power and field length, and, as a consequence, it cannot move in a direction of lower cost. (See Figure 3b.)

The "kinks" in the highest power curves of Figures 3b and 3c occur because the program allows only discrete values of cruise altitude which leads to slight discontinuities in the goal function.

2. Non-Optimal Operation. The previous discussion deals with aircraft operation under the conditions for which that aircraft is designed. Possibly, however, a commuter operator would like to have the ability to fly his airplanes at a fast speed even if he normally flies much more slowly.

Figure 4 shows the cost penalty incurred for two cases of non-optimal operation. The costs for the optimum airplanes designed for cruise at 330 knots and field lengths of 3,500 and 4,000 feet, but actually flown at several lower cruise speeds over the 150 nautical mile typical stage length, are shown. Although the cost does decrease as the airplane slows down, it does not reach the economy level achieved for the optimized airplane at each speed. The difference in DOC between the optimized aircraft and the high-speed airplane flown at a lower speed reaches as high as 1.4% for airplanes meeting a 4,000 foot landing distance and as high as 5% for airplanes with 3,500 foot field lengths. The non-optimized airplanes cost more to operate at a given speed since their larger engines and higher wing areas contribute to higher weight and drag and, thus, to more fuel burned per mission.

## D. Ride Roughness

Figure 5 shows ride roughness parameter, $\Delta n$, as a function of field length for the airplanes generated by the optimizing portion of this study. The plot does not form a family of smooth curves with speed as a parameter due to the discrete altitudes allowed in the optimizing routine. In particular, the lowest field length, 330 knot airplane flies at 25,000 feet because of the extremely sub-optimal $C_L$ produced at lower altitude. As the field length increases and the wing becomes smaller, the airplanes come down to

20,000 feet since smaller engines provide the required power at the lower altitude.

Also shown in figure 5 are the roughness parameters of several comparable-mission aircraft. Notice that, according to the method employed here, three of the optimum commuters would, in fact, have more favorable gust response than the DC-9-30. If indeed true, this suggests that commuters need only increase wing loading by using a good flap system, or slow down to 250 kts, in order to solve the ride roughness problem. Because of qualitative assessments of commuter ride roughness, however, some skepticism remains as to the validity of using this particular parameter to compare these somewhat different aircraft.

The theory for response to a sharp-edged gust (see Jones, ref. 8) can be applied to the optimum aircraft and to the DC-9-30. Figure 6 shows the theoretical curves for maximum $\Delta C_L$ vs. mass ratio and those points corresponding to the optimal commuters from this study. The ordinate for this plot, maximum $\Delta C_L$, is obtained from:

$$\Delta n = \frac{\Delta L}{L} = \frac{\Delta C_L}{C_L}$$

$$\Delta C_L = \Delta n \, C_L$$

Since, however, $\Delta n$ is computed for a 30 ft/sec equivalent airspeed gust and the theory shows response to a unit gust, the result must be normalized by:

$$\text{maximum } \Delta C_L = \Delta n \, C_L \, \frac{V}{U}$$

where $U$ = true gust velocity for which $\Delta n$ is computed

$V$ = true airplane speed

The theoretical plot indicates, first of all, that the gust response method of reference 7 coincides quite well with the theory for a sharp-edged gust. Secondly, it shows that the DC-9, with comparable mass ratio and aspect ratio to the optimal commuters, should indeed have comparable gust response. Even with this evidence, however, a need remains for verification of $\Delta n$ as a useful gust response parameter, especially since the theoretical curve in figure 6 has been computed only up to mass ratios of 280 - far less than those of present-day airplanes.

Figure 7 shows a diagram of the optimal airplane for 330 knot cruise speed and a 4,000 feet field length.

## IV. Conclusions

- Increasing cruise speed (beyond 290 knot ) and decreasing allowable field length tend to increase direct operating cost, but only a six percent difference in DOC exists between the best and worst airplanes studied. This occurs because each airplane is optimized with respect to direct operating cost for its particular mission.

- One-engine-out enroute climb gradient requirements restrict the commuter aircraft with turboprops more than they do a turbofan aircraft because the commuter's speed for best climb gradient is less than its enroute minimum allowable speed.

- The major drawback to increasing speed and decreasing runway length is the increased ride roughness due to both higher velocity and lower wing loading. The worst ride roughness calculated for an optimal airplane represents a 45 percent increase in the relative parameter $\Delta n$ over the lowest value.

- Some work remains to verify the FAR value $\Delta n$, as a useful parameter for comparing airplane ride roughness.

# V. References

[1] Shevell, Richard S., "Aerospace Systems Synthesis and Analysis", Department of Aeronautics and Astronautics, Stanford University, September 1978.

[2] ________________ . "Standard Method of Estimating Comparative Direct Operating Costs of Turbine Powered Transport Airplanes", Air Transport Association of America, December 1967.

[3] Nash, J.C., Compact Numerical Methods for Computers, New York: John Wiley & Sons, 1979.

[4] Lockheed-California Company, Application of Advanced Technologies to Small Short-Haul Transport Aircraft, NASA CR 152363, June 1980.

[5] Smith, C.E., Hirschkron, R., Warren, R.E., Propulsion System Study for Small Transport Aircraft Technology (STAT), General Electric Company, NASA CR 165330, May 1981.

[6] Ebeling, A., The Determination of Airplane Maximum Lift Coefficients, Douglas Aircraft Company, Inc., July 1964.

[7] Shevell, Richard S., Bayan, Fawzi P., "A Brief Ride Roughness Analysis", Department of Aeronautics and Astronautics, Stanford University, SUDAAR No. 526, May 1981.

[8] Jones, Robert T., "The Unsteady Lift of a Wing of Finite Aspect Ratio", NACA Report No. 681, 1940.

Table 1. Active Constraints

| Cruise Speed (kts) | Field Length Constraint (ft) | Active Constraint[1] |
|---|---|---|
| 250 | 3,500 | Enroute climb gradient |
| | 3,725[2] | " |
| 290 | 3,500 | " |
| | 3,750 | " |
| | 4,000 | " |
| | 4,060[2] | " |
| 330 | 3,500 | Maximum cruise power |
| | 3,750 | " |
| | 4,000 | " |
| | 4,275[2] | enroute climb gradient |

1. This column contains the second active constraint. The first active constraint is landing distance at the field length listed in column 2.

2. Critical field length above which field length does not determine wing area, power or aspect ratio.

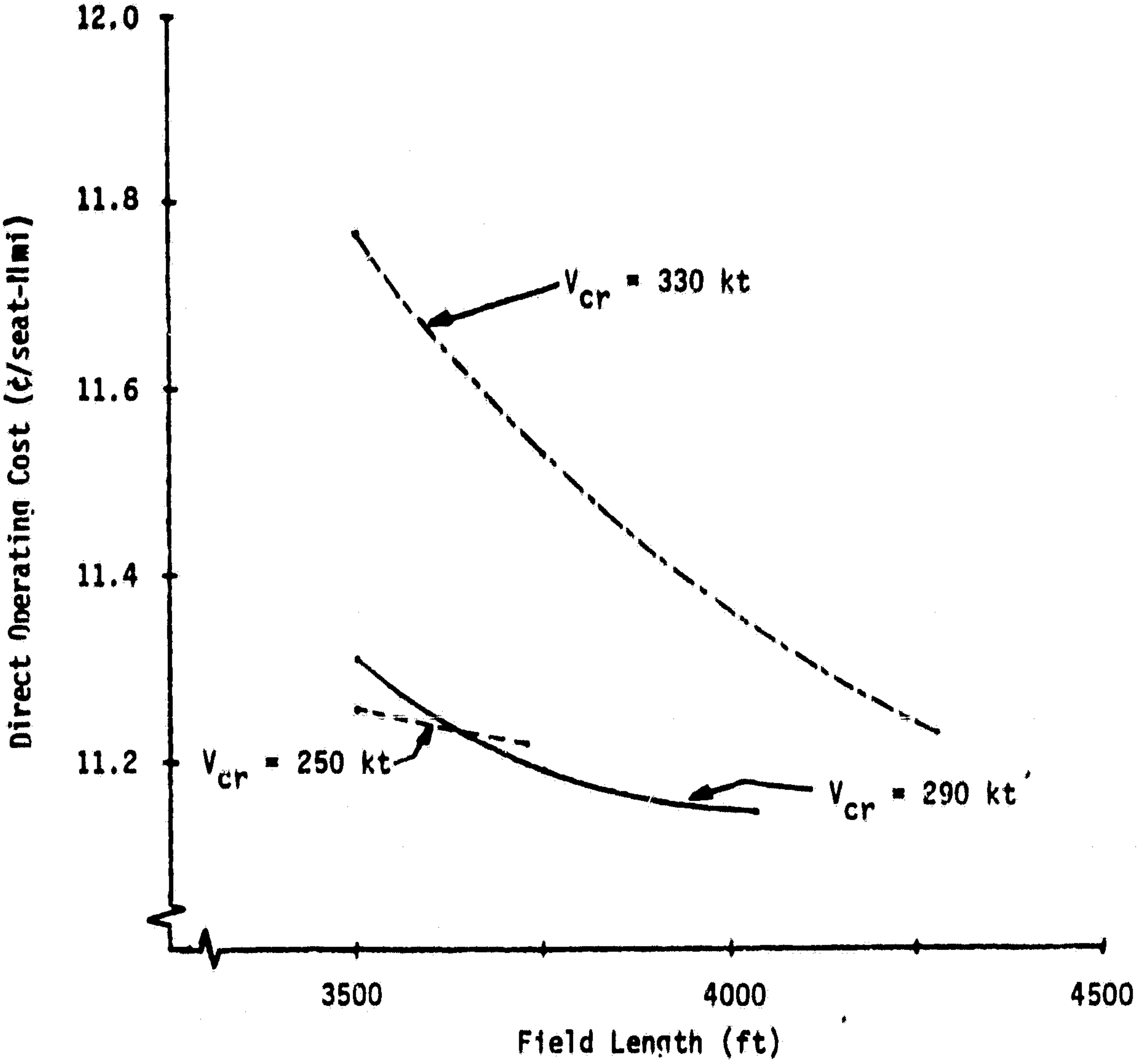


Figure 1. Direct Operating Cost vs. Field Length

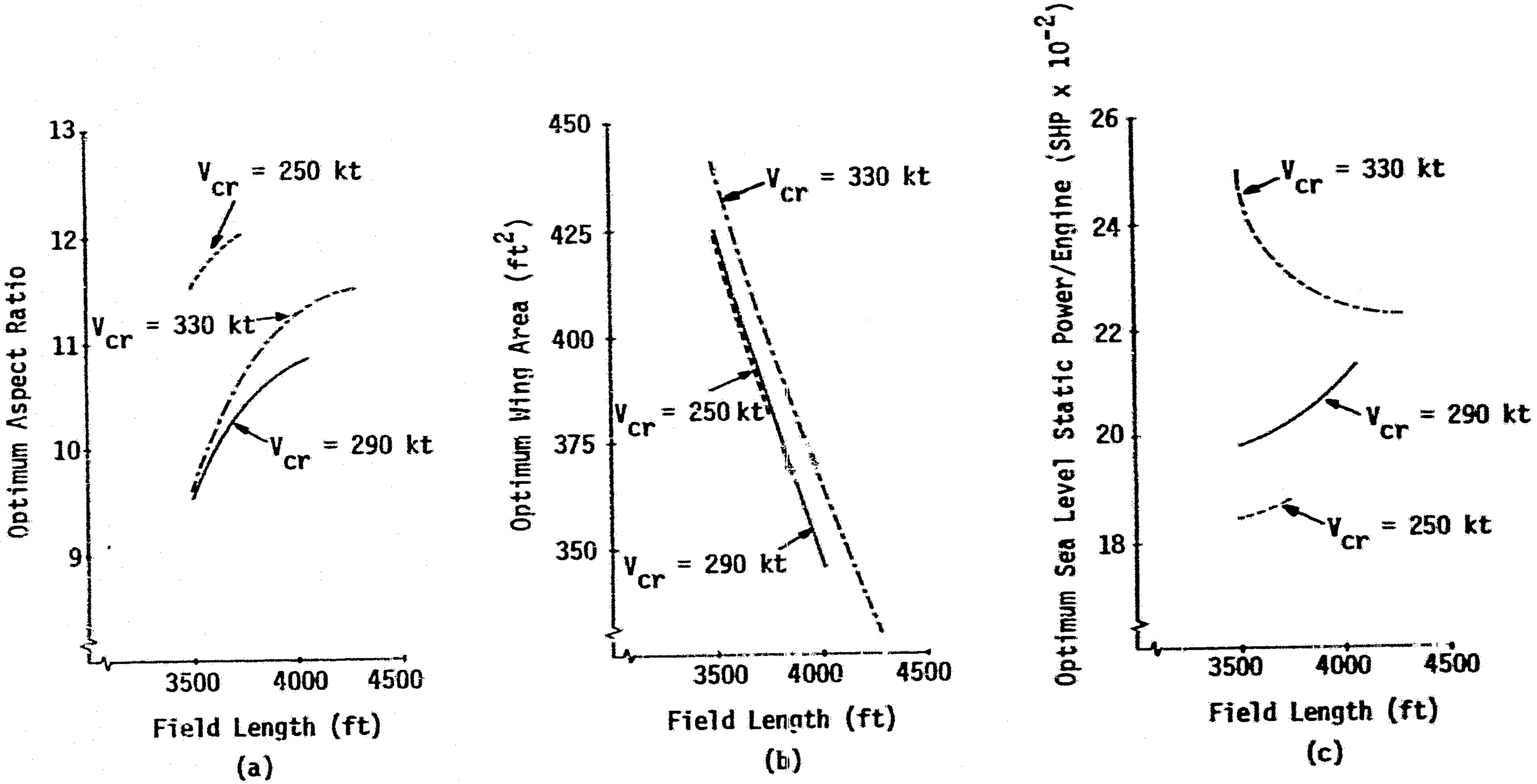


Figure 2. Optimal Variable Values vs. Field Length

ORIGINAL PAGE IS
OF POOR QUALITY

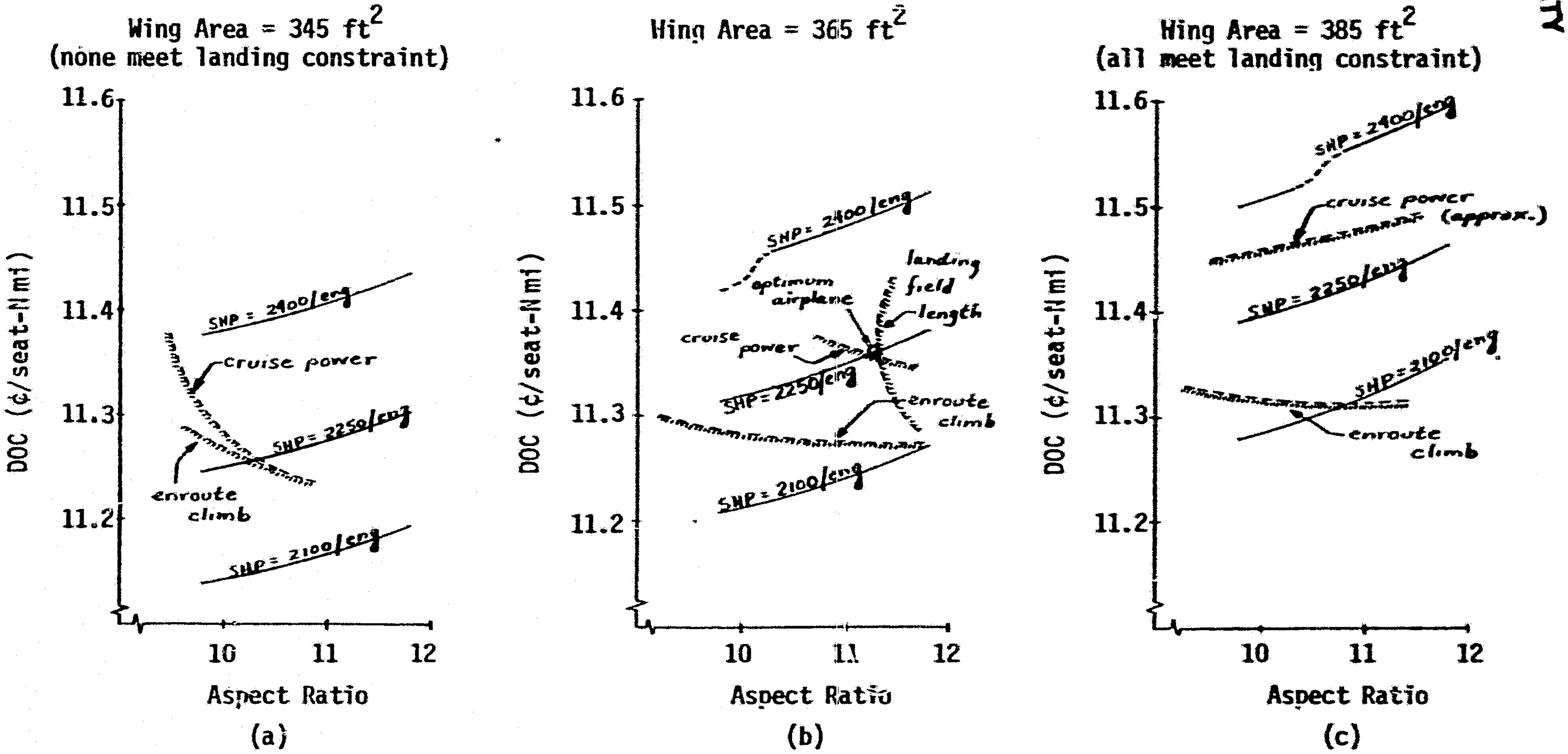


Figure 3. Optimal Point Sensitivities
Cruise Speed = 330 kt, Maximum Field Length = 4000 feet

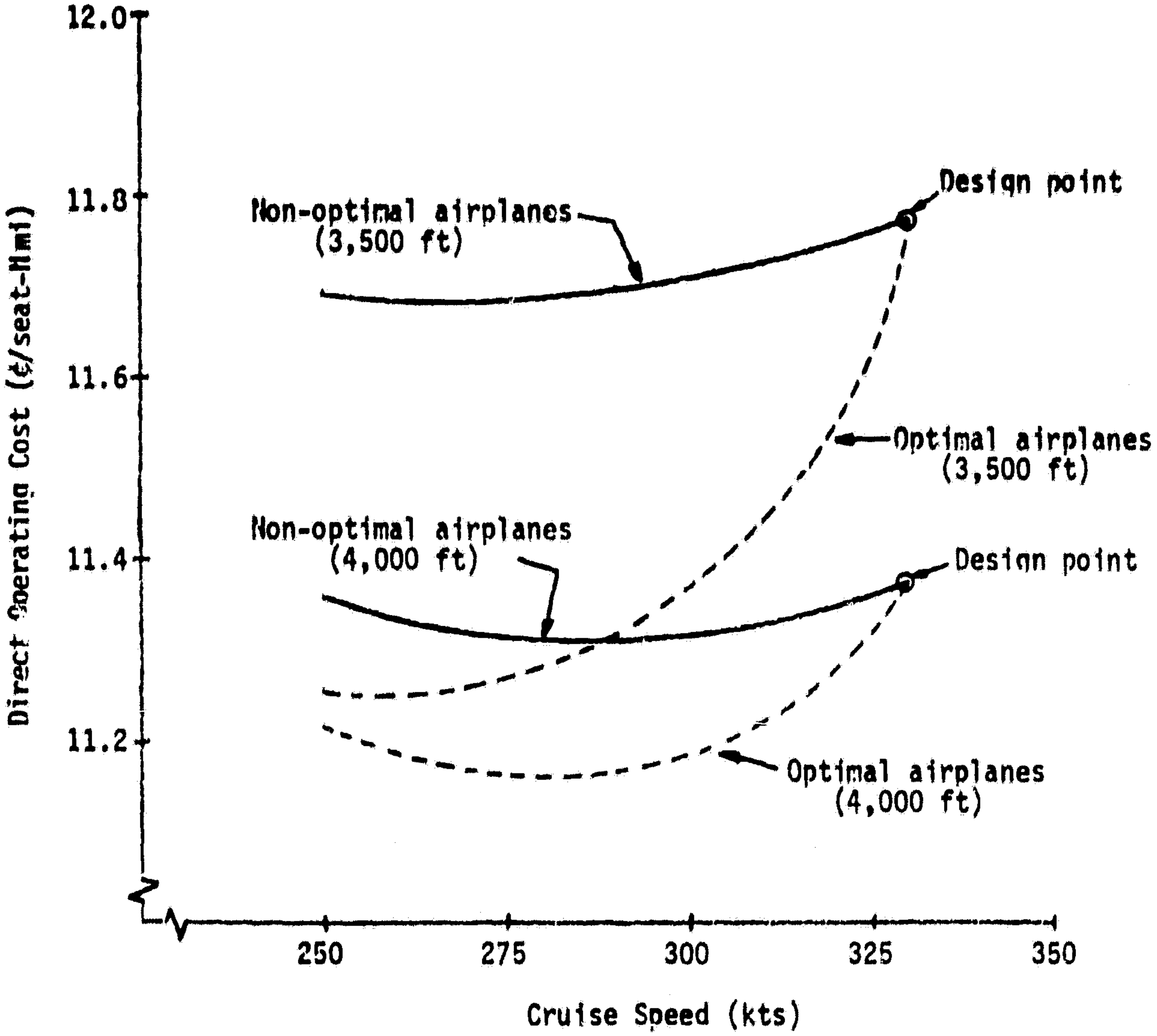


Figure 4. Cost for Non-Optimal Operation
[330 kt designs flown at lower speeds]

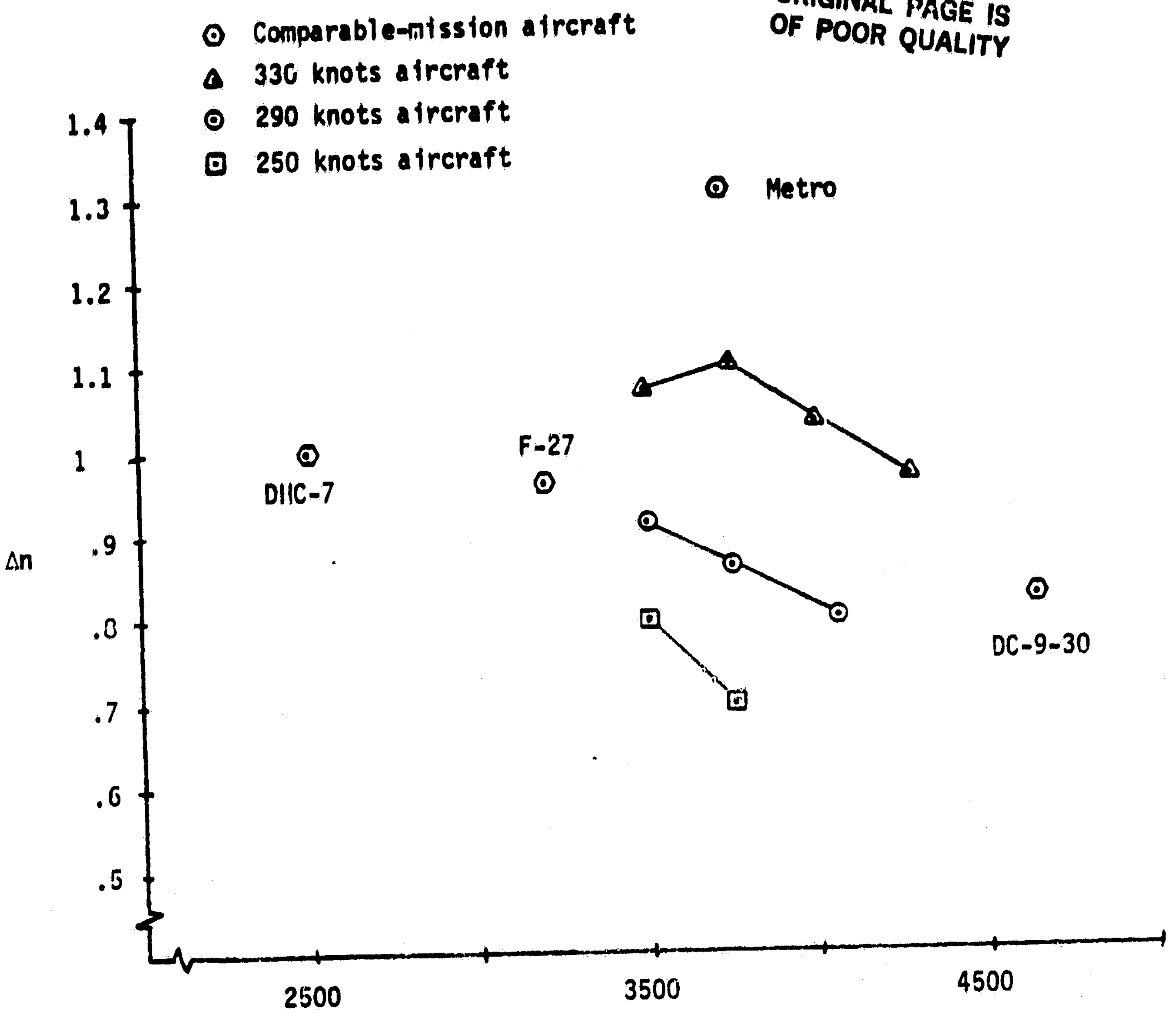


Figure 5. Ride Roughness vs. Field Length

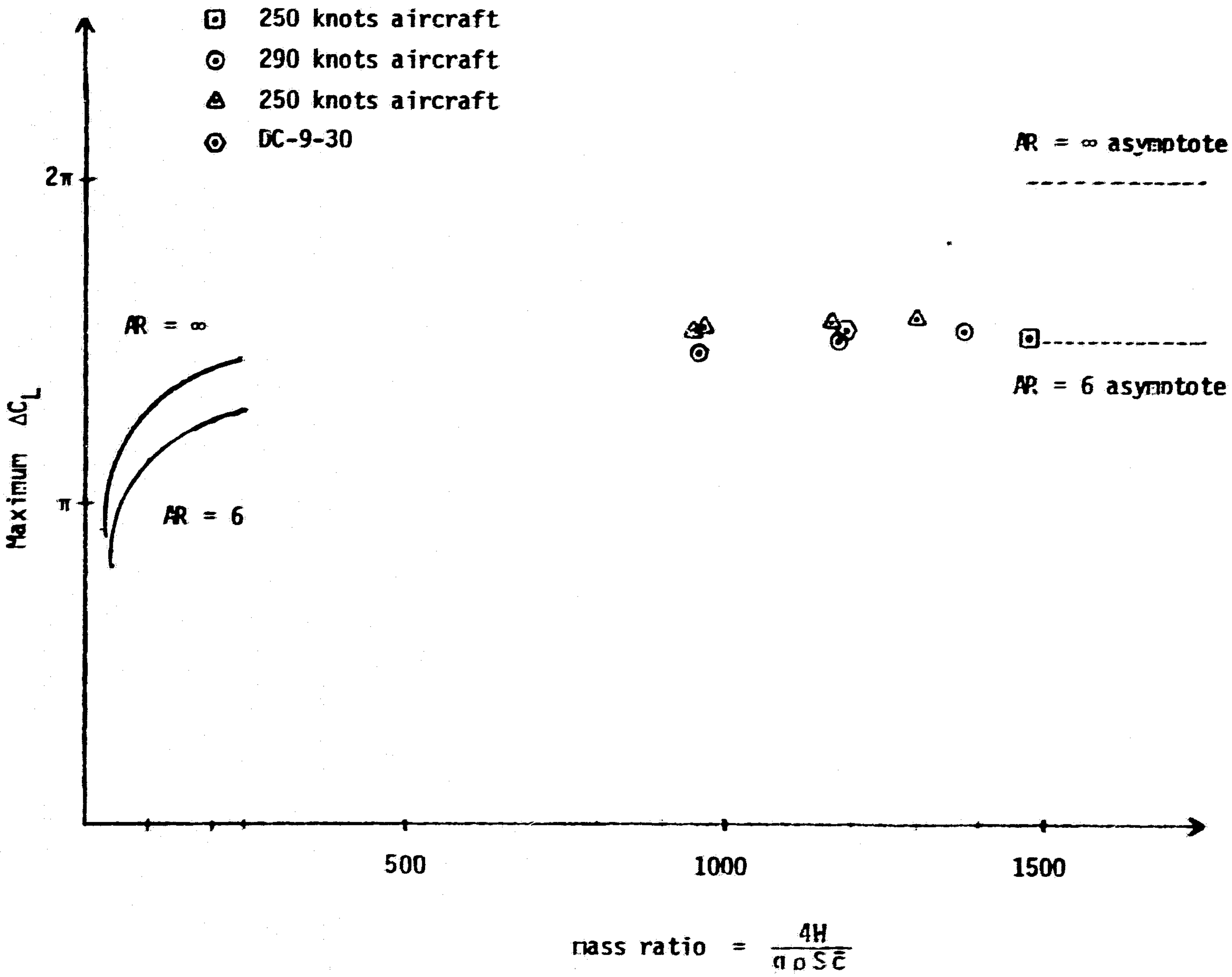
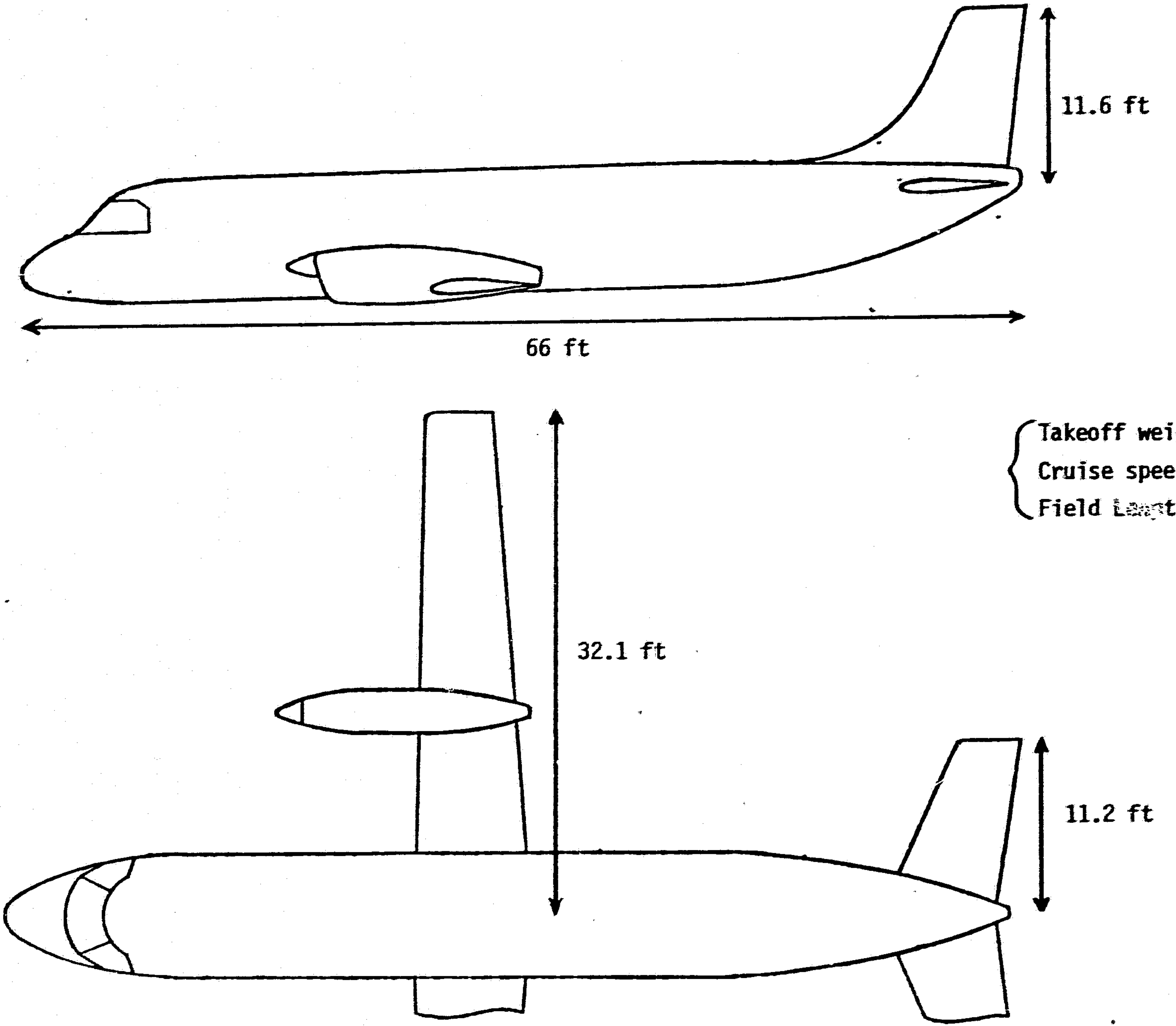


Figure 6. Theoretical Gust Response



Takeoff weight: 25,370 lbs
Cruise speed : 330 kts
Field Length : 4,000 ft

Figure 7. Optimized Commuter Aircraft

# Appendix I

# Program Listing

ORIGINAL PAGE IS
OF POOR QUALITY

```
      COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                         SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                    C  PROGRAM TO MINIMIZE A FUNCTION OF N VARIABLES BY A `VARIABLE
                    C  METRIC ALGORITHM,' WHICH, IN FACT, IS A SECOND ORDER GRADIENT
                    C  METHOD USING AN ESTIMATED GRADIENT COMPUTATION (SUBROUTINE
                    C  GRDENT).
                    C
  ISN 0002                COMMON/CONST/PI,G
  ISN 0003                COMMON/BLOCK1/BLOCKT,BLOCKF,VCRUS,STAGE
  ISN 0004                REAL K
  ISN 0005                INTEGER COUNT
  ISN 0006                DIMENSION XB(15),GR(15),C(15),X(15),T(15),BI(15,15)
  ISN 0007                N = 3
  ISN 0008                IG = 0
  ISN 0009                IFN = 0
  ISN 0010                H = .2
  ISN 0011                TOL = .0001
                    C
  ISN 0012                PI = 3.1415927
  ISN 0013                G = 32.18
  ISN 0014                DO 500 LENGTH = 3500,4250,250
  ISN 0015                FLEN = FLOAT(LENGTH)
  ISN 0016                VCRUS = 422.25
  ISN 0017                IFN = 0
  ISN 0018                IG = 0
                    C
                    C  FILL IN THE INITIAL VALUES FOR THE STATE VECTOR.
                    C
  ISN 0019                DO 1 I = 1,N
  ISN 0020                XB(I) = 1.
  ISN 0021           1    CONTINUE
                    C
                    C  CALCULATE FUNCTION VALUE AT INITIAL CONDITIONS.
                    C
  ISN 0022                CALL EVAL(XB,N,P0,FLEN)
  ISN 0023                CALL INFO(FLEN)
                    C
                    C  SET THE FUNCTION EVALUATION COUNTER.
                    C
  ISN 0024                IFN = IFN +1
                    C
                    C  COMPUTE THE GRADIENT OF THE FUNCTION AT THE INITIAL POINT.
                    C
  ISN 0025                CALL GRDENT(P0,XB,N,GR,FLEN)
                    C     GR(1) = 2.*(XB1) - 1.) - 400.*XB1)*(XB2) - XB1)**2)
                    C     GR(2) = 200.*(XB2) - XB1)**2)
                    C
                    C  SET THE GRADIENT EVALUATION COUNTER.
                    C
  ISN 0026                IG = IG + 1
                    C
                    C  SET UP THE MATRIX BI AS THE IDENTITY MATRIX.
                    C
  ISN 0027           3    DO 20 I = 1,N
  ISN 0028                DO 21 J = 1,N
  ISN 0029                BI(I,J) = 0.
```

```
ISN 0030     21       CONTINUE
ISN 0031              B1(I,I) = 1.
ISN 0032     20       CONTINUE
            C
            C  SET COUNTER
            C
ISN 0033              ILAST = IG
            C
            C  START MAIN ITERATION LOOP.
            C
ISN 0034     4        WRITE(6,100)IG,IFN,P0
ISN 0035     100      FORMAT(/,1X,'#GRADS,#FUNCS,FUNC VALUE = ',2I5,F16.8)
ISN 0036              WRITE(6,79)(GR(I),I=1,N),(XB(I),I=1,N)
ISN 0037     79       FORMAT(1X,'GRADS = ',3F16.8,/,' VARIABLES = ',3F16.8)
            C
ISN 0038              DO 22 I = 1,N
ISN 0039              X(I) = XB(I)
ISN 0040              C(I) = GR(I)
ISN 0041     22       CONTINUE
            C
            C  COMPUTE THE STEP VECTOR, T, AND THE DIRECTION INDICATOR, D1.
            C
ISN 0042              D1 = 0.
ISN 0043              DO 23 I = 1,N
ISN 0044              SS = 0.
ISN 0045              DO 24 J = 1,N
ISN 0046              SS = SS - B1(I,J)*GR(J)
ISN 0047     24       CONTINUE
ISN 0048              T(I) = SS
ISN 0049              D1 = D1 - SS*GR(I)
ISN 0050     23       CONTINUE
            C
            C  NOW, DETERMINE WHICH DIRECTION WE ARE TRAVELLING.
            C
ISN 0051              IF (D1 .LE. 0.) GO TO 60
            C
            C  IF WE ARE GOING DOWN, CONTINUE THE SEARCH.
            C
ISN 0053              K = 1
ISN 0054     8        COUNT = 0
            C
            C  FIND THE NEXT STATE VECTOR.
            C
ISN 0055              DO 25 I = 1,N
ISN 0056              STEP = K*T(I)
ISN 0057              IF (ABS (STEP) .GT. .1) STEP = .1*STEP/ABS(STEP)
ISN 0059              XB(I) = X(I) + STEP
ISN 0060              IF (XB(I) .EQ. X(I)) COUNT = COUNT + 1
ISN 0062     25       CONTINUE
            C
            C  DETERMINE CONVERGENCE
            C
ISN 0063              IF (COUNT .GE. N) GO TO 60
ISN 0065              CALL EVAL(XB,N,P,FLEN)
ISN 0066              IFN = IFN + 1
            C
```

```
                C  TRY ACCEPTANCE TEST.  IF THE NEW POINT IS NOT WITHIN THE TOLERANCE
                C  REQUIRED, CONTINUE THE ITERATION. IF SO, TRY A NEW TOLERANCE.
                C
ISN 0067                PACC =  P0 - D1*K*TOL
ISN 0068                IF (P .GE. PACC) GO TO 70
ISN 0070                P0 = P
                C
                C  COMPUTE THE GRADIENT AT THE NEW POINT.
                C
ISN 0071                CALL GRDENT(P0,XB,N,GR,FLEN)
                C       GR(1) = 2.*(XB(1) - 1.) - 400.*XB(1)*(XB(2) - XB(1)**2)
                C       GR(2) = 200.*(XB(2) - XB(1)**2)
ISN 0072                IG = IG + 1
                C
                C  COMPUTE GRADIENT DIFFERENCE DIRECTION.
                C
ISN 0073                D1 = 0.
ISN 0074                DO 26 I = 1,N
ISN 0075                T(I) = K*T(I)
ISN 0076                C(I) = GR(I) - C(I)
ISN 0077                D1 = D1 + T(I)*C(I)
ISN 0078         26     CONTINUE
                C
ISN 0079                IF (D1 .LE. 0.) GO TO 3
                C
                C  NOW DO SOME STUFF THAT I DON'T COMPLETELY UNDERSTAND.
                C  HAS TO DO WITH COMPUTING THE RATE OF CHANGE OF THE GRADIENT.
                C
ISN 0081                D2 = 0.
ISN 0082                DO 27 I = 1,N
ISN 0083                SS = 0.
ISN 0084                DO 28 J = 1,N
ISN 0085                SS = SS + B1(I,J)*C(J)
ISN 0086         28     CONTINUE
ISN 0087                X(I) = SS
ISN 0088                D2 = D2 + SS*C(I)
ISN 0089         27     CONTINUE
                C
ISN 0090                D2 = 1. + D2/D1
ISN 0091                DO 29 I = 1,N
ISN 0092                DO 29 J = 1,N
ISN 0093                B1(I,J) = B1(I,J)-(T(I)*X(J)+X(I)*T(J)-D2*T(I)*T(J))/D1
ISN 0094         29     CONTINUE
                C
ISN 0095                GO TO 4
                C
ISN 0096         60     IF (ILAST .NE. IG) GO TO 3
ISN 0098                GO TO 18
                C
ISN 0099         70     K = W*K
ISN 0100                GO TO 8
                C
ISN 0101         18     WRITE(6,102)(XB(I),I=1,N)
ISN 0102        102     FORMAT(/,1X,'VARIABLES = ',3F16.8)
ISN 0103                WRITE(6,100)IG,IFN,P0
ISN 0104                CALL EVAL(XB,N,P0,FLEN)
```

```
ISN 0105          CALL INFO(FLEN)
            C
ISN 0106     500  CONTINUE
ISN 0107      99  STOP
ISN 0108          END
```

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                 C
                 C
                 C
                 C  SUBPROGRAM TO COMPUTE THE GRADIENT OF THE FUNCTION AT A POINT.
                 C
      ISN 0002            SUBROUTINE GRDENT(S,B,N,G,FLEN)
                 C
      ISN 0003            DIMENSION B(15),G(15),D(15)
                 C
      ISN 0004            DO 10 I = 1,N
      ISN 0005            DO 20 J = 1,N
      ISN 0006            D(J) = B(J)
      ISN 0007     20     CONTINUE
      ISN 0008            H = ABS(B(I) + 1.E-2)*1.E-2
      ISN 0009            D(I) = B(I) + H
      ISN 0010            CALL EVAL(D,N,SP,FLEN)
      ISN 0011            G(I) = (SP - S)/H
      ISN 0012     10     CONTINUE
                 C
      ISN 0013            RETURN
      ISN 0014            END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                    C
                    C
                    C
  ISN 0002                SUBROUTINE EVAL(X,N,GOAL1,FL)
                    C
                    C  DEFINES GOAL FUNCTION FOR MINIM PROGRAM.
                    C
  ISN 0003                COMMON/VARIAB/TOW,S,SLSHP,AR,VCLIMB,HCRUS
  ISN 0004                COMMON/BLOCK1/BLOCKT,BLOCKF,VCRUS,STAGE
  ISN 0005                COMMON/CONSTR/DIST,DIFF,TOFL,XLFL,GAMSS,GAMER,CENT
                    C
                    C  CHANGE VARIABLES IN THE X MATRIX, WHICH HAVE BEEN NORMALIZED TO
                    C  ONE, TO THEIR VALUES.
                    C
  ISN 0006                DIMENSION X(15)
  ISN 0007                S = X(1) * 400.
  ISN 0008                SLSHP = X(2) * 4000.
  ISN 0009                AR = X(3) * 10.
                    C
                    C
                    C
  ISN 0010                XNM = 0.
  ISN 0011                XNT = 0.
  ISN 0012                XNL = 0.
  ISN 0013                XNS = 0.
  ISN 0014                XNE = 0.
                    C
  ISN 0015                CALL PLANE
  ISN 0016                CALL RANGE(DIST,WC,VCRUS)
  ISN 0017                HMAX = HCRUS
  ISN 0018                CALL MAXTHR(WC,VCRUS,HMAX,DIFF)
  ISN 0019                CALL TAKOFF(TOFL)
  ISN 0020                CALL LANDNG(XLFL)
  ISN 0021                CALL SECSEG(GAMSS)
  ISN 0022                CALL ENRCLM(GAMER)
  ISN 0023                CALL BLOCK(VCRUS,BLOCKT,BLOCKF,STAGE)
  ISN 0024                CALL DOC(STAGE,BLOCKT,BLOCKF,CENT)
                    C
                    C  COMPUTE THE GOAL FUNCTION.  IF THE CONSTRAINTS ARE NOT MET, THE
                    C  FUNCTION WILL BE VERY LARGE DUE TO THE FOLLOWING DEFINITIONS OF
                    C  THE COEFFICIENTS.
                    C
  ISN 0025                IF (DIFF .LT. 0.) XNM = 10.
  ISN 0027                IF (TOFL .GT. FL) XNT = 1.
  ISN 0029                IF (XLFL .GT. FL) XNL = 1.
  ISN 0031                IF (GAMSS .LT. .024) XNS = 10000.
  ISN 0033                IF (GAMER .LT. .011) XNE = 10000.
                    C
  ISN 0035                GOAL1 = CENT + XNM*ABS(DIFF) + XNT*
                         1 ABS(TOFL-FL) + XNL*ABS(XLFL-FL) + XNS*ABS(GAMSS-.024) +
                         2 XNE*ABS(GAMER-.011)
                    C
  ISN 0036                RETURN
  ISN 0037                END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                   C
                   C
                   C
                   C
  ISN 0002               SUBROUTINE INFO(FL)
                   C
                   C  ROUTINE TO PRINT OUT THE VALUE OF ALL CONSTRAINTS AT THE TIME IT
                   C  IS CALLED.
                   C
  ISN 0003                 COMMON/CONSTR/DIST,DIFF,TOFL,XLFL,GAMSS,GAMER,CENT
  ISN 0004                 COMMON/BLOCK1/BLOCKT,BLOCKF,VCRUS,STAGE
  ISN 0005                 COMMON/CHARAC/CDP,E,B
  ISN 0006                 COMMON/WEIGHT/OWEPL,AFWT
  ISN 0007                 COMMON/GEOM/XMAC,SHE,SHG,BH,SVE,SVG,BV
  ISN 0008                 COMMON/VARIAB/TOW,S,SLSHP,AR,VCLIMB,HCRUS
                   C
  ISN 0009                 WRITE(6,200)VCRUS,FL
  ISN 0010          200    FORMAT(1X,'CRUISE SPEED = ',F6.2,2X,'FIELD LENGTH = ',F7.1)
  ISN 0011                 WRITE(6,201)TOW,S,SLSHP,AR,VCLIMB,HCRUS
  ISN 0012          201    FORMAT(//,1X,'TOW,S,SLSHP,AR = ',4F10.3,/,' VCLIMB,HCRUS = '
                       1   ,2F9.2)
  ISN 0013                 WRITE(6,210)OWEPL,AFWT
  ISN 0014          210    FORMAT(//,1X,'ZFW, AIRFRAME WEIGHT = ',2F10.2)
  ISN 0015                 WRITE(6,202)XMAC,SHG,SVE
  ISN 0016          202    FORMAT(//,1X,'MAC = ',F5.3,/,' GROSS HORIZONTAL AREA = ',
                       1   F7.3,/,' EXPOSED VERTICAL AREA = ',F7.3)
  ISN 0017                 WRITE(6,203)CDP,E
  ISN 0018          203    FORMAT(//,' CDP,E = ',2F12.8)
  ISN 0019                 WRITE(6,100)DIST,DIFF,TOFL
  ISN 0020          100    FORMAT(//,1X,'RANGE = ',F14.7,/,1X,'DIFF = ',F14.7,/,1X,
                       1   'TOFL = ',F14.7)
                   C
  ISN 0021                 WRITE(6,101)XLFL,GAMSS,GAMER
  ISN 0022          101    FORMAT(/,1X,'LFL = ',F14.7,/,1X,'2ND SEGMENT CLIMB GRADIENT = '
                       1   ,F14.7,/,1X,'ENROUTE CLIMB GRADIENT = ',F14.7)
                   C
  ISN 0023                 WRITE(6,204)STAGE,BLOCKT,BLOCKF
  ISN 0024          204    FORMAT(//,1X,'STAGE LENGTH = ',F5.1,/,' BLOCK TIME = ',
                       1   F5.3,'HR',/,' BLOCK FUEL = ',F8.2,'LBS')
                   C
  ISN 0025                 WRITE(6,102)CENT
  ISN 0026          102    FORMAT(/,1X,'CENTS PER SEAT NAUTICAL MILE = ',F14.7,//)
                   C
  ISN 0027                 RETURN
  ISN 0028                 END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST:NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                     C
                     C
                     C
 ISN 0002                  SUBROUTINE RANGE(DIST,WCT,VCRUS)
                     C
 ISN 0003                    REAL MAXRNG
 ISN 0004                  COMMON/VARIAB/TOW,S,SLSHP,AR,VCLIMB,HCRUS
 ISN 0005                  COMMON/WEIGHT/OWEPL,AFWT
 ISN 0006                  COMMON/CONST/PI,G
 ISN 0007                  COMMON/CHARAC/CDP,E,B
 ISN 0008                    COMMON/FLAG/KANT
 ISN 0009                    COMMON/AERO/VEQW,CLALPH
                     C
                     C  COMPUTES THE TOW TO MEET THE MAXIMUM RANGE MISSION USING A FITTED
                     C  PARABOLA MINIMIZATION PROCEDURE.  USES A GRID SEARCH TO FIND THE
                     C  BEST CLIMB SPEED AND CRUISE ALTITUDE FOR THE GIVEN AIRPLANE.
                     C
 ISN 0010                    TOW = 60000.
 ISN 0011                    TOWH = 50000.
 ISN 0012                    ICOUNT = 0
                     C
                     C  BEGIN SEARCH LOOP ON CLIMB SPEED.
                     C
 ISN 0013                    DO 200 I = 250,300,25
 ISN 0014                    CLMSPD = FLOAT(I)
                     C
                     C  BEGIN SEARCH LOOP ON CRUISE ALTITUDE.
                     C
 ISN 0015                    DO 100 J = 20000,30000,5000
 ISN 0016                    HMAX = FLOAT(J)
                     C       WRITE(6,300)HMAX,CLMSPD
 ISN 0017             300    FORMAT(//,' HCRUS,VCLM = ',2F14.4)
 ISN 0018                    ICOUNT = ICOUNT + 1
 ISN 0019                    INTNUM = 1
 ISN 0020                    TEMP = 518.69 - .00356*HMAX
 ISN 0021                    XMACH = VCRUS/SQRT(1.4*1718.*TEMP)
 ISN 0022                    BETA2 = 1. - (1.07*XMACH)**2
 ISN 0023                    CLALPH = (2.*PI*AR)/(2.+SQRT(AR**2*BETA2*1.0628+4.))
                     C
                     C  CALCULATE DESIGN EQUIVALENT STRUCTURAL SPEED AT 10000 FT.
                     C
 ISN 0024                    VEQW = 1.07*VCRUS*SQRT(.0017553/.0023769)
                     C
 ISN 0025                 RHO = 2.3769E-3*(1. - 6.8634E-6*HMAX)**(4.2648)
 ISN 0026                 Q = .5 * RHO * VCRUS**2
                     C
                     C  GUESS TOW AND FIND EMPTY WEIGHT.
                     C
 ISN 0027                 X2 = 25000.
 ISN 0028                 STEP = X2/50.
 ISN 0029                 MAXRNG = 600.
 ISN 0030                 CONV = 1.
 ISN 0031                   H = HMAX
 ISN 0032                   VCL = CLMSPD
```

```
              C
              C  NOW CALL THE FUNCTION EVALUATION ROUTINE TO FIND THE TOW FOR THE
              C  GIVEN RANGE.
              C
              C  MINIMIZATION ROUTINE
              C
              C
  ISN 0033          CALL DISTR(X2,F2,H,VCL,WD,Q,MAXRNG,WC,FI,VCRUS,ZFW,WTAF)
              C
              C  TEST FOR CLIMB VIOLATION.  KANT IS 1 IF CLIMB GRADIENT < 0.
              C
  ISN 0034            IF (KANT .EQ. 0) GO TO 10
  ISN 0036            IF (ICOUNT .EQ. 1) GO TO 30
  ISN 0038            IF (ICOUNT .NE. 1) GO TO 100
              C
  ISN 0040     10   X3 = X2 + STEP
  ISN 0041          X1 = X2 - STEP
  ISN 0042          CALL DISTR(X1,F1,H,VCL,WD,Q,MAXRNG,WC,FI,VCRUS,ZFW,WTAF)
  ISN 0043          CALL DISTR(X3,F3,H,VCL,WD,Q,MAXRNG,WC,FI,VCRUS,ZFW,WTAF)
  ISN 0044     3      INTNUM = INTNUM + 1
  ISN 0045            IF (INTNUM .GT. 50) GO TO 89
  ISN 0047            IF (F1 .LE. F2) GO TO 1
  ISN 0049            IF (F3 .LE. F2) GO TO 2
              C
              C  WHERE F2 IS LESS THAN F1 AND F3, FIT A PARABOLA, FIND MINIMUM AND
              C  USE THAT AS THE NEXT X2
              C
  ISN 0051          X2 = X2 + .5*STEP*(F1-F3)/(F3-2.*F2*F1)
  ISN 0052          CALL DISTR(X2,F2,H,VCL,WD,Q,MAXRNG,WC,FI,VCRUS,ZFW,WTAF)
  ISN 0053          IF (F2 .LE. CONV) GO TO 99
  ISN 0055          STEP = STEP/3.
  ISN 0056          GO TO 10
              C
              C  WHERE F2 IS BETWEEN OR GREATER THAN F1 AND F3, MAKE X2 THE VALUE
              C  FOR A MINIMUM
              C
  ISN 0057     1    IF (F3 .LT. F1) GO TO 2
  ISN 0059            X3 = X2
  ISN 0060            F3 = F2
  ISN 0061            X2 = X1
  ISN 0062            F2 = F1
  ISN 0063            X1 = X1 - STEP
  ISN 0064            CALL DISTR(X1,F1,H,VCL,WD,Q,MAXRNG,WC,FI,VCRUS,ZFW,WTAF)
  ISN 0065            GO TO 3
  ISN 0066     2      X1 = X2
  ISN 0067            F1 = F2
  ISN 0068            X2 = X3
  ISN 0069            F2 = F3
  ISN 0070            X3 = X3 + STEP
  ISN 0071            CALL DISTR(X3,F3,H,VCL,WD,Q,MAXRNG,WC,FI,VCRUS,ZFW,WTAF)
  ISN 0072            GO TO 3
              C
              C  NOT CONVERGING IF INTNUM > 50.  PRINT ERROR MESSAGE.
              C
  ISN 0073     89     WRITE(6,302)
  ISN 0074     302    FORMAT(' INTNUM = 50, CONTINUING ...')
```

```
ISN 0075             GO TO 100
               C
ISN 0076        99   IF (X2 .GE. TOWH) GO TO 100
ISN 0078             CALL MAXTHR(WC,VCRUS,H,DIFF)
ISN 0079             IF (DIFF .LT. 0. .AND. HMAX .NE. 20000.) GO TO 100
ISN 0081             WCH = WC
ISN 0082             AFWT1 = WTAF
ISN 0083             OWEPL1 = ZFW
ISN 0084             DIST1 = FI
ISN 0085             TOWH = X2
ISN 0086             HCRUS = HMAX
ISN 0087       100   CONTINUE
               C
ISN 0088             IF (TOWH .GE. TOW) GO TO 200
ISN 0090             TOW = TOWH
ISN 0091             VCLIMB = CLMSPD
ISN 0092             WCT = WCH
ISN 0093             AFWT = AFWT1
ISN 0094             OWEPL = OWEPL1
ISN 0095             DIST = DIST1
ISN 0096       200   CONTINUE
               C
ISN 0097           RETURN
               C
ISN 0098        30   CALL INFO
ISN 0099             STOP
               C
ISN 0100           END
```

```
LEVEL 21.8 ( JUN 74 )                    OS/360  FORTRAN H                         DATE  82.016/13.51.58

       COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                         SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                 C
                 C
                 C
  ISN 0002                 SUBROUTINE TAKOFF(TOFL)
                 C
  ISN 0003                 COMMON/VARIAB/TOW,S,SLSHP,AR,VCLIMB,HCRUS
                 C
                 C  COMPUTES FAR TAKEOFF DISTANCE BASED ON PROF SHEVELL'S CURVE
                 C  FOR TWIN ENGINE AIRCRAFT IN AA 241 NOTES.  CURVE WAS FITTED
                 C  TO A QUADRATIC -- 952.8 + 26.672A + .0255A¢2 -- WHERE A
                 C  IS THE PARAMETER W¢2/SIGMA*CLMAX*S*TH.  TAKEOFF POWER VS.
                 C  SPEED FOR SEA LEVEL AND ISA + 30.8 DEGREES F WAS FITTED TO
                 C  A CUBIC.  POWER RATIO WAS BASED ON THE GENERAL ELECTRIC CT-7
                 C  ENGINE.  ETA OF .65 IS ASSUMED FOR TAKEOFF.
                 C
  ISN 0004                 ETA = .65
  ISN 0005                 CLMAX = 2.25
  ISN 0006                 V = .84 * SQRT((2.*TOW)/(.002244*CLMAX*S))
                 C
  ISN 0007                 SHP = SLSHP*(.89181-4.057E-4*V+3.2768E-6*V**2-5.2103E-9*V**3)
  ISN 0008                 TH = 550.*ETA*SHP/V
                 C
  ISN 0009                 A = TOW**2/(.9441*CLMAX*S*TH)
                 C
  ISN 0010                 TOFL = 952.8 + 26.672*A + .0255*A**2
                 C
  ISN 0011                 RETURN
  ISN 0012                 END
```

```
LEVEL 21.8 ( JUN 74 )                  OS/360  FORTRAN H                             DATE  82.016/13.52.08

          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                C                                                                    531.
                C                                                                    532.
                C                                                                    533.
   ISN 0002           SUBROUTINE LANDNG(XLFL)                                        534.
                C                                                                    535.
   ISN 0003           COMMON/VARIAB/TOW,S,SLSHP,AR,VCLIMB,HCRUS                      536.
                C                                                                    537.
                C  COMPUTES LANDING DISTANCE USING LINEAR FITTED CURVE OF LANDING    538.
                C  DISTANCE VS STALL SPEED SQUARED.  ASSUMES DOUBLE SLOTTED FLAPS BUT 539.
                C  NO SLATS.  MAX LANDING WEIGHT EQUALS TOW.                         540.
                C                                                                    541.
   ISN 0004             VS2 = 2.*TOW/(.002244*2.67*S*1.689**2)                       542.
   ISN 0005             XLFL = .4*VS2 + 750.                                         543.
                C                                                                    544.
   ISN 0006           RETURN                                                         545.
   ISN 0007           END                                                            546.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
         COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                            SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                      C
                      C
                      C
  ISN 0002                  SUBROUTINE SECSEG(GAMSS)
                      C
  ISN 0003                  COMMON/VARIAB/TOW,S,SLSHP,AR,VCLIMB,HCRUS
  ISN 0004                  COMMON/CONST/PI,G
  ISN 0005                  COMMON/CHARAC/CDP,E,B
                      C
                      C  COMPUTES SECOND SEGMENT CLIMB GRADIENT (MUST BE GREATER THAN
                      C  OR EQUAL TO 2.4% FOR THIN ENGINE AIRPLANES TO MEET PART 25).
                      C  ETA IS CONSIDERED TO BE .75 FOR 2ND SEGMENT CLIMB.  CALCULATED
                      C  FOR HOT DAY (ISA + 30.8 DEGREES F) BUT MAY BE CHANGED BY
                      C  CHANGING DENSITY.
                      C
  ISN 0006                  H = 400.
  ISN 0007                  CLMAX = 2.25
  ISN 0008                  V = 1.2 * SQRT((2*TOW)/(.002244*S*CLMAX) )
  ISN 0009                  HP = SLSHP/2.
  ISN 0010                  CDPO = CDP + .0008*49.*PI/S + .0027 + .0155
  ISN 0011                  ETA = .7
  ISN 0012                  SHP = HP*(.89181-4.057E-4*V+3.2760E-6*V**2-5.2103E-9*V**3)
  ISN 0013                  TH = 550.*ETA*SHP/V
  ISN 0014                    D = DRAG(H,TOW,V,CDPO,S)
  ISN 0015                    DVDH = 1.4636E-5*V*(1.-6.8634E-6*H)**(-3.1324)
  ISN 0016                    GAMSS = (TH - D)/(TOW*(1.-(V/G)*DVDH))
                      C
  ISN 0017                  RETURN
  ISN 0018                  END
```

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                     C
                     C
                     C
  ISN 0002                SUBROUTINE ENRCLM(GAMER)
                     C
  ISN 0003                COMMON/VARIAB/TOW,S,SLSHP,AR,VCLIMB,HCRUS
  ISN 0004                COMMON/CHARAC/COP,E,B
  ISN 0005                COMMON/CONST/PI,G
                     C
                     C  COMPUTES ENROUTE CLIMB GRADIENT AT V = 1.2 VS IN THE CRUISE
                     C  CONFIGURATION.  ETA IS .75.  GRADIENT MUST BE 1.1% TO MEET
                     C  PART 25 REGULATIONS.
                     C
  ISN 0006                H = 11000.
  ISN 0007                VEQ = VCLIMB
  ISN 0008                ETA = .8
  ISN 0009                CDPO = CDP
  ISN 0010                CALL CLIMB(S,SLSHP,TOW,H,VEQ,CDP,ETA,FC,DC,TC,GAMMA,WC)
                     C
                     C  NOW WE HAVE THE HEIGHT AT 11000 FT (OBSTACLE CLEARANCE HEIGHT).
                     C  COMPUTE THE CLIMB GRADIENT AT THAT HEIGHT.
                     C
  ISN 0011                RHO = 2.3769E-3*(1. - 6.8634E-6*H)**(4.2648)
  ISN 0012                CLMAX = 1.58
  ISN 0013                V = 1.3 * SQRT((2.*WC)/(RHO*S*CLMAX))
  ISN 0014                HP = SLSHP/2.
  ISN 0015                ETA = .75
  ISN 0016                SHP = POWER(HP,V,H)
  ISN 0017                  TH = 550.*ETA*SHP/V
  ISN 0018                CDPO = CDP + .0008*49.*PI/S + .0027
  ISN 0019                D = DRAG(H,WC,V,CDPO,S)
                     C
  ISN 0020                GAMER = (TH - D)/WC
                     C
  ISN 0021                RETURN
  ISN 0022                END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                C
                C
                C
  ISN 0002            SUBROUTINE PLANE
                C
  ISN 0003              REAL K
  ISN 0004            COMMON/VARIAB/TOW,S,SLSHP,AR,VCLIMB,HCRUS
  ISN 0005            COMMON/CHARAC/COP,E,B
  ISN 0006            COMMON/CONST/PI,G
  ISN 0007            COMMON/GEOM/XMAC,SHE,SHG,BH,SVE,SVG,BV
                C
                C  THIS SUBROUTINE DEFINES:
                C       1)  THE AIRPLANE GEOMETRY
                C       2)  DRAG PARAMETERS
                C
                C  AIRPLANE GEOMETRY
                C  B IS THE WING SPAN.  XMAC IS THE MEAN AERODYNAMIC CHORD CAL-
                C  CULATED BY ASSUMING A TAPER RATIO OF .4.  THE TAIL SURFACES
                C  ARE COMPUTED BY FITTING THE TAIL VOLUME CURVES USED IN AA241
                C  AND USING A TAIL LENGTH OF 30 FT FOR VERTICAL TAIL AND 32 FEET
                C  FOR THE HORIZONTAL TAIL.  THE HORIZONTAL ASPECT RATIO IS 4 AND THE
                C  VERTICAL TAIL ASPECT RATIO IS 1.8.  THE CG RANGE IS ASSUMED
                C  TO BE 25% OF THE MAC.
                C
  ISN 0008            B = SQRT(AR*S)
  ISN 0009            XMAC = 1.0612*S/B
                C
                C  HORIZONTAL TAIL.
                C
  ISN 0010            A1 = 4347.38/(S*XMAC)
  ISN 0011            SHE = (.25*(1.8667*A1+2.4)*S*XMAC)/32.
  ISN 0012            SHG = 1.2 * SHE
  ISN 0013            BH = SQRT(4.*SHG)
                C
                C  VERTICAL TAIL
                C
  ISN 0014            A2 = 4347.38/(S*B)
  ISN 0015            SVE = (S*B*(.3333*A2+.034))/30.
  ISN 0016            SVG = 1.05*SVE
  ISN 0017            BV = SQRT(1.8*SVG)
                C
                C  DRAG PARAMETERS
                C  FIND THE WING WETTED AREA BY FIRST COMPUTING THE ROOT CHORD AND
                C  THE ROOT CHORD AT THE FUSELAGE.
                C
  ISN 0018              SWEEPH = .15
  ISN 0019              SWEEPV = .15
  ISN 0020              SWEEP = 0.
  ISN 0021              TCW = .15
  ISN 0022              TCH = .1
  ISN 0023              TCV = .1
  ISN 0024              CR = 2.*(S/B)/(1.+.4)
  ISN 0025              CRF = CR - (1.-.4)*(CR/B)*8.116
                C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
              C  EXPOSED AREA IS THE TOTAL AREA MINUS THAT AREA COVERED BY THE
              C  FUSELAGE.
              C
ISN 0026            SE = ((B - 8.116)/2.)*(CRF + .4*CR)
ISN 0027            SWET = 2.04*SE
              C
              C  FIND THE EXPOSED MAC AND THE REYNOLDS NUMBER ASSOCIATED WITH THAT
              C  LENGTH AT A CRUISE SPEED OF 290 KT AT 25,000 FT.
              C
ISN 0028            XMACE = .6667*(CRF+.4*CR-.4*CRF*CR/(CRF+.4*CR))
ISN 0029            RE = 1.867E6*XMACE
              C
              C  FRICTION COEFFICIENT IS A FUNCTION OF THE LOG(BASE 10) OF THE
              C  REYNOLD NUMBER.
              C
ISN 0030            RELOG = ALOG10(RE)
ISN 0031            CF = (78.868-26.463*(RELOG)+3.1025*(RELOG)**2-.12417*(RELOG)
                  1 **3)*1.E-3
              C
              C  FIND THE FORM FACTOR USING THE FORMULA GIVEN IN THE AA241 NOTES.
              C
ISN 0032            Z = 1.75*COS(SWEEP)/SQRT(1.-.25*(COS(SWEEP))**2)
ISN 0033            K = 1. + Z*TCH + 100.*TCH**4
                    WRITE(6,*)SWET,CF,K,TCH,Z
              C
              C
              C  NOW FIND THE F OF THE WING.
              C
ISN 0034            FWING = CF*K*SWET
              C
              C  DO THE SAME THING FOR THE HORIZONTAL TAIL.
              C
ISN 0035            SWET = 2.04*SHE
ISN 0036            RE = 1.867E6*SHE/BH
ISN 0037            RELOG = ALOG10(RE)
ISN 0038            CF = (78.868-26.463*(RELOG)+3.1025*(RELOG)**2-.12417*(RELOG)
                  1 **3)*1.E-3
ISN 0039            Z = 1.75*COS(SWEEPH)/SQRT(1.-.25*(COS(SWEEPH))**2)
ISN 0040            K = 1. + Z*TCH + 100.*TCH**4
ISN 0041            FHORIZ = CF*K*SWET
              C
              C  NOW DO THE SAME FOR THE VERTICAL TAIL.
              C
ISN 0042            SWET = 2.04*SVE
ISN 0043            RE = 1.867E6*SVE/BV
ISN 0044            RELOG = ALOG10(RE)
ISN 0045            CF = (78.868-26.463*(RELOG)+3.1025*(RELOG)**2-.12417*(RELOG)
                  1 **3)*1.E-3
ISN 0046            Z = 1.75*COS(SWEEPV)/SQRT(1.-.25*(COS(SWEEPV))**2)
ISN 0047            K = 1. + Z*TCV + 100.*TCV**4
ISN 0048            FVERT = CF*K*SWET
              C
              C  FIND THE GAP DRAG USING THE METHOD OF AA241.
              C
ISN 0049            FGAP = .0042*(BH*(COS(SWEEPH))**2+BV*(COS(SWEEPV))**2+B/4.)
              C
              C  THE CONSTANT F'S HAVE BEEN DETERMINED AS:
```

```
                 C
ISN 0050               FFUSE = 3.799
ISN 0051               FNACPY = 1.15
                 C
                 C  NOW FIND THE TOTAL F USING MISCELLANEOUS DRAG TO EQUAL 6%.
                 C
ISN 0052             ' FTOT = (FWING+FHORIZ+FVERT+FFUSE+FNACPY+FGAP)/.94
                 C
                 C  TO FIND THE PARASITE DRAG COEFFICIENT, DIVIDE BY THE WING AREA.
                 C
ISN 0053               CDP = FTOT/S
                 C
                 C  EFFICIENCY FACTOR IS CALCULATED USING INDUCED DRAG FACTORS
                 C  FOUND IN SHEVELL'S NOTES.
                 C
ISN 0054              SS = 1. - .0745*(8.116/B) - 1.6338*(8.116/B)**2
ISN 0055              U = .99867+4.33864E-4*AR-9.59822E-5*AR**2+2.02546E-6*AR**3
ISN 0056              E = 1./(PI*AR*(1./(PI*AR*U*SS)+.4368*CDP))
                 C
ISN 0057              RETURN
ISN 0058              END
```

```
                C
                C
                C
ISN 0002            SUBROUTINE POUNDS(TOW,S,SLSHP,ZFW,WTAF)
                C
                C  TRANSLATION OF BASIC WEIGHT PROGRAM USED IN AA241 -- SHEVELL'S
                C  WEIGHT METHOD.  CONSTANT THIRTY PASSENGER SIZE FUSELAGE OF
                C  DIAMETER 8.116 FT AND LENGTH 66 FT.  PAYLOAD OF 6270 LBS.
                C  PROGRAM MUST KNOW TOW, S AND SLSHP AS WELL AS VALUES IN
                C  COMMON/GEOM.  MUST ALSO BE GIVEN MAX GUST VELOCITY (50 FPS).
                C
ISN 0003            COMMON/CONST/PI,G
ISN 0004            COMMON/CHARAC/CDP,E,B
ISN 0005            COMMON/GEOM/XMAC,SHE,SHG,BH,SVE,SVG,BV
ISN 0006              COMMON/AERO/VEQ,CLALPH
                C
                C
                C  DESIGN STRUCTURAL SPEED DETERMINED AT 10000 FT.
                C
ISN 0007              RHO = .0017553
                C
ISN 0008            T7 = 9601.15
ISN 0009            U9 = 50.
ISN 0010            F9 = 3500./TOW
                C
                C  LANDING GEAR WEIGHT
                C
ISN 0011            W7 = .04 * TOW
ISN 0012            Z2 = TOW * (1. - F9)
                C
ISN 0013         10   Z1 = Z2
ISN 0014            V1 = 2.*Z1*B/(RHO*G*S**2*CLALPH)
ISN 0015            XK1 = .88*V1/(5.3 + V1)
ISN 0016            V2 = 2.*TOW*B/(RHO*G*S**2*CLALPH)
ISN 0017            XK2 = .88*V2/(5.3 + V2)
ISN 0018            F1 = 1. + (XK1*CLALPH*U9*VEQ)/(841.12*(Z1/S))
ISN 0019            F3 = 1. + (XK2*CLALPH*U9*VEQ)/(841.12*(TOW/S))
ISN 0020            F5 = 2.1 + (24000./(TOW + 10000.))
ISN 0021              IF (F5 .GT. 3.8) F5 = 3.8
ISN 0023              IF (F5 .LT. 2.5) F5 = 2.5
ISN 0025            IF (F3 .LT. F5) F3 = F5
ISN 0027            F2 = 1.5 * F3
                C
                C  WING WEIGHT;  AVERAGE THICKNESS = .15
                C
ISN 0028            XI1=(F2*B**3*SQRT(TOW*Z1)*1.8E-6)/(.15*S**2*1.4)
ISN 0029            W1 = (1.642*XI1 + 4.22)*S
                C      WRITE(6,*)XI1,W1
                C
                C  HORIZONTAL TAIL WEIGHT;  AVERAGE THICKNESS = .12
                C
ISN 0030            XI2= (F2*BH**3*TOW*XMAC*SQRT(SHE)*1E-6)/(6.31125*SHG**1.5*SHE)
ISN 0031            W2 = (1.5*XI2 + 5.25)*SHE
                C
```

```
                 C  VERTICAL TAIL AND RUDDER WEIGHT; VERTICAL AVG T/C = .12
                 C
ISN 0032               XI3=(F2*BV**3*(8.+.44*TOW/S)*1E-3)/(.12*.96*.75*SVE)
ISN 0033               W3 = (.0145*XI3 + 3.51)*.75*SVE
ISN 0034               W4 = 1.6*W3/3.
                 C
                 C  SURFACE CONTROLS WEIGHT
                 C
ISN 0035               W5 = 1.7*(SHG+SVG)
                 C
ISN 0036               IF (F1 .LT. 2.5) F1 = F3
                 C
                 C  FUSELAGE WEIGHT
                 C
ISN 0038               SLSHPH = SLSHP/2.
ISN 0039               WENG = 2*(-16.56 + 12.58*SQRT(SLSHPH) + .0618*SLSHPH)
ISN 0040               W9 = 6528. + WENG
ISN 0041               W0 = 1948. + WENG
                 C
ISN 0042               T1 = .6*F1*(Z1-W1-W0)*66./(PI*8.116**2)
ISN 0043               T2 = T1 - T7
ISN 0044               IF (T2 .LT. 0.) T2 = 0.
ISN 0046               XI6 = (T7 + (T2**2/(2.*T1)))*1E-3
ISN 0047               W6 = (.102*XI6 + 1.051)*1472.47
                 C
                 C  DETERMINE ZFW.  COMPARE WITH ESTIMATED ZFW.  IF NOT SAME,
                 C  ITERATE.
                 C
ISN 0048               Z2 = W1+W2+W3+W4+W5+W6+W7+.65*S+W9+6270.
ISN 0049               IF (ABS((Z2 - Z1)/Z1) .GT. .0001) GO TO 10
ISN 0051               ZFW = Z2
ISN 0052               WTAF = Z2 - 7270. - WENG
                 C
ISN 0053               RETURN
ISN 0054               END
```

```
       COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                          SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                 C                                                                  832.
                 C                                                                  833.
 ISN 0002             REAL FUNCTION DRAG(H,W,V,CDPO,S)                              834.
                 C                                                                  835.
 ISN 0003             COMMON/CHARAC/CDP,E,B                                         836.
 ISN 0004             COMMON/CONST/PI,G                                             837.
                 C                                                                  838.
                 C  COMPUTES DRAG FOR A GIVEN FLIGHT CONDITION                      839.
                 C                                                                  840.
 ISN 0005             RHO = 2.3769E-3*(1 - 6.8634E-6*H)**(4.2648)                   841.
 ISN 0006             Q = .5*RHO*V**2                                               842.
                 C                                                                  843.
 ISN 0007             DRAG = CDPO*Q*S + (W**2)/(Q*PI*(B**2)*E)                      844.
                 C                                                                  845.
 ISN 0008             RETURN                                                        846.
 ISN 0009             END                                                           847.
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                  C
                  C
                  C
    ISN 0002            SUBROUTINE CLIMB(S,SLSHP,TOW,HMAX,VEQ,CDPO,ETA,FC,TC,DC,
                       XGAMMAC,W)
                  C
    ISN 0003            COMMON/CHARAC/CDP,E,B
    ISN 0004            COMMON/CONST/PI,G
    ISN 0005              COMMON/FLAG/KANT
                  C
                  C  COMPUTES TIME, FUEL AND DISTANCE TO CLIMB TO HMAX.  VEQ IS
                  C  CONSTANT EQUIVALENT AIRSPEED FOR CLIMB.  RETURNS CLIMB
                  C  GRADIENT, GAMMAC, AND WEIGHT AT END OF CLIMB.  CDPO IS THE
                  C  BASIC CDP PLUS WHATEVER ADDITIONAL DRAG IS INCURRED DUE TO
                  C  CONFIGURATION (IE. FLAP DRAG, SPOILER DRAG.)
                  C  TC,FC AND DC ARE TIME, FUEL AND DISTANCE TO CLIMB, RESPEC-
                  C  TIVELY.  THE PROGRAM INTEGRATES IN ALTITUDE STEPS -- SIMPLE
                  C  BACKWARD EULER INTEGRATION.
                  C
    ISN 0006              KANT = 0
    ISN 0007            DH = 200.
    ISN 0008            W = .995*TOW
    ISN 0009            H = 0.
    ISN 0010            TC = 0.
    ISN 0011            FC = 0.
    ISN 0012            DC = 0.
    ISN 0013              SFC = .47
                  C
                  C  BEGIN INTEGRATION.
                  C
    ISN 0014       10   V = VEQ*(1-6.8634E-6*H)**(-2.1324)
    ISN 0015            DVDH = 1.4636E-5*VEQ*(1-6.8634E-6*H)**(-3.1324)
    ISN 0016            SHP = POWER(SLSHP,V,H)
    ISN 0017              TH = 550.*ETA*SHP/V
                  C
                  C  FIRST, ESTIMATE DRAG USING LIFT = WEIGHT.  THEN USE LIFT =
                  C  WEIGHT * THE COSINE OF THE ESTIMATED GAMMA FOR A BETTER
                  C  VALUE OF CLIMB GRADIENT.
                  C
    ISN 0018            D = DRAG(H,W,V,CDPO,S)
    ISN 0019            GAMMA1 = (TH - D)/(W*(1+(V/G)*DVDH))
    ISN 0020            XLIFT = W * COS(GAMMA1)
    ISN 0021            D = DRAG(H,XLIFT,V,CDPO,S)
                  C
                  C  NOW FIND THE CORRECT CLIMB GRADIENT
                  C
    ISN 0022            GAMMAC = (TH - D)/(W*(1+(V/G)*DVDH))
                  C
                  C  IF GAMMA IS NEGATIVE THE AIRPLANE CAN'T CLIMB -- GIVE AN
                  C  ERROR MESSAGE.
                  C
    ISN 0023            IF (GAMMAC .LT. 0.) GO TO 50
                  C
                  C CALCULATE RATE OF CLIMB
```

```
                 C
ISN 0025              RC = V*GAMMAC
                 C
                 C  CALCULATE FUEL TO CLIMB
                 C
ISN 0026              IF (H .EQ. 0.) GO TO 20
ISN 0028              DELTAF = (SHP*SFC)/(3600.*RC)*DH
ISN 0029              FC = FC + DELTAF
ISN 0030              W = W - DELTAF
                 C
                 C  CALCULATE DISTANCE TO CLIMB
                 C
ISN 0031              DELTAD = V*DH/RC
ISN 0032              DC = DC + DELTAD
                 C
                 C  CALCULATE TIME TO CLIMB
                 C
ISN 0033              DELTAT = DH/RC
ISN 0034              TC = TC + DELTAT
                 C
                 C  TAKE A STEP
                 C
ISN 0035          20  H = H + DH
ISN 0036              IF (H .LE. HMAX) GO TO 10
                 C
ISN 0038          60  CONTINUE
ISN 0039                GO TO 70
                 C
                 C  ERROR MESSAGE
                 C
ISN 0040          50  WRITE(6,100)
ISN 0041         100  FORMAT(1X,'AIRPLANE CANNOT CLIMB -- INADEQUATE FUEL OR HP.')
ISN 0042                KANT = 1
ISN 0043          70  RETURN
ISN 0044              END
```

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                C
                C
                C
 ISN 0002             SUBROUTINE DESENT(SLSHP,S,HMAX,ETA,FD,TD,DD,W,OWEPL)
                C
 ISN 0003             COMMON/CONST/PI,G
 ISN 0004             COMMON/CHARAC/CDP,E,B
                C
                C  DESENT COMPUTES TIME, FUEL AND DISTANCE TO DESCEND FROM HMAX.
                C  IT REQUIRES A CONSTANT EQUIVALENT AIRSPEED FOR DESCENT, VEQ,
                C  AND A CONSTANT RATE OF DESCENT, RD, IN FEET PER SEC.  IT
                C  USES A SIMPLE BACKWARD EULER INTEGRATION TO WORK BACKWARDS
                C  FROM THE GROUND TO HMAX.
                C
 ISN 0005             RD = HMAX/(20.*60.)
 ISN 0006               VEQ = 300.
 ISN 0007             H = 0.
 ISN 0008             W = OWEPL/.977
 ISN 0009             DH = 500.
 ISN 0010             FD = 0.
 ISN 0011             DD = 0.
 ISN 0012             TD = 0.
                C
                C  BEGIN INTEGRATION.  DEPENDING ON THE RATE OF DESCENT, MAY
                C  NEED SOME SPOILERS.  SINCE IDLE POWER SEEMS TO BE AROUND .1 OF
                C  MAX POWER, USE SPOILERS IF THE REQUIRED POWER TURNS OUT TO BE LESS
                C  THAN .1 OF MAX.
                C
 ISN 0013             CDPO = CDP
                C
 ISN 0014        10   V = VEQ * (1 - 6.8634E-6*H)**(-2.1324)
 ISN 0015             DVDH = (1.4636E-5*VEQ*(1-6.8634E-6*H)**(-3.1324))
                C
                C  FIND DESCENT GRADIENT, GAMMAD, AND THE LIFT PRODUCED FOR THIS CONDI-
                C  TION.  THEN DETERMINE THE DRAG AND THRUST REQUIRED.
                C
 ISN 0016             GAMMAD = RD/V
 ISN 0017             XLIFT = W * COS(GAMMAD)
 ISN 0018             D = DRAG(H,XLIFT,V,CDPO,S)
 ISN 0019             TH = D - W*GAMMAD*(1-(V/G)*DVDH)
 ISN 0020             SHP = (TH * V)/(550 * ETA)
                C
                C  FIND THE SFC FOR THIS POWER SETTING.  Z IS RATIO OF SHP TO SHPMAX.
                C
 ISN 0021               SHPMAX = POWER(SLSHP,V,H)
 ISN 0022               Z = SHP/SHPMAX
 ISN 0023               IF (Z .LT. .1) Z = .1
 ISN 0025               SFC = .43 + 2.07E-2/Z + 2.02E-2/Z**2 - 1.04E-3/Z**3
                C
                C  CALCULATE FUEL TO DESCEND
                C
 ISN 0026             DELTAF = (Z*SHPMAX * SFC)/(3600. * RD) * DH
 ISN 0027             FD = FD + DELTAF
 ISN 0028             W = W + DELTAF
```

```
LEVEL 21.8 ( JUN 74 )                    OS/360  FORTRAN H                                  DATE  82.016/13.52.28

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                   C
                   C
                   C
  ISN 0002               REAL FUNCTION POWER(SLSHP,V,H)
                   C
                   C  CALCULATES AVAILABLE MAX CRUISE AND CLIMB HORSEPOWER AS A
                   C  FUNCTION OF ALTITUDE AND SPEED (IN FT/SEC).  CURVES ARE FIT
                   C  FOR GENERAL ELECTRIC CT-7 ENGINE.  CALCULATES RATIO OF AVAIL-
                   C  ABLE POWER TO SEA LEVEL STATIC POWER RATING.  MAX POWER IS
                   C  FOUND BY MULTIPLYING RATIO BY SEA LEVEL SHP.
                   C
  ISN 0003                C1 = .8461 - 1.802E-5*H
  ISN 0004                C2= .5921*(-1.156E-4-1.443E-8*H+3.260E-13*H**2+5.133E-18*H**3)
  ISN 0005                C3 = .3505*(1.904E-6+1.758E-10*H-5.875E-15*H**2)
  ISN 0006                C4 = .2075*(-3.315E-9-2.26E-13*H+9.301E-18*H**2)
                   C
  ISN 0007                SHPR = C1 + C2*V + C3*V**2 + C4*V**3
  ISN 0008                POWER = SLSHP * SHPR
                   C
  ISN 0009                RETURN
  ISN 0010                END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                              SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                C
                C
                C
  ISN 0002            SUBROUTINE BLOCK(VCRUS,BLOCKT,BLOCKF,STAGE)
                C
  ISN 0003            COMMON/VARIAB/TOW,S,SLSHP,AR,VCLIMB,HCRUS
  ISN 0004            COMMON/WEIGHT/OWEPL,AFWT
  ISN 0005            COMMON/CONST/PI,G
  ISN 0006            COMMON/CHARAC/CDP,E,B
                C
                C  CALCULATES BLOCK TIME AND BLOCK FUEL FOR THE STAGE LENGTH TO BE
                C  OPTIMIZED.  USES AN ITERATIVE PROCEDURE TO DETERMINE THE TOW
                C  FOR THE GIVEN RANGE.  BLOCK TIME AND FUEL ARE NECESSARY FOR
                C  FINDING THE DIRECT OPERATING COSTS.
                C
  ISN 0007            STAGE = 150.
  ISN 0008            HMAX = 15000.
  ISN 0009            VEQ = VCLIMB
  ISN 0010            ETA = .8
  ISN 0011            RHO = 2.3769E-3*(1. - 6.8634E-6*HMAX)**(4.2640)
  ISN 0012            Q = .5 * RHO * VCRUS**2
                C
                C  CALCULATE TIME, FUEL, AND DISTANCE TO DESCEND ASSUMING WEIGHT
                C  AT END OF DESCENT IS OWE + PAYLOAD + RESERVES.
                C
  ISN 0013            CALL DESENT(SLSHP,S,HMAX,ETA,FD,TD,DD,WD,OWEPL)
                C
                C  ASSUME A TAKEOFF WEIGHT.  CALCULATE CLIMB AND CRUISE UNTIL THE
                C  WEIGHT IS RIGHT.
                C
  ISN 0014            X2 = WD + 1000.
  ISN 0015            STEP = X2/100.
  ISN 0016            CONV = .1
  ISN 0017            STAGEM = STAGE - DD/6072.
  ISN 0018              INTNUM = 0
                C
                C
                C  MINIMIZATION ROUTINE
                C
  ISN 0019            CALL DISTN(X2,F2,HMAX,VCLIMB,WD,Q,STAGEM,WC,FI,TC,FC,VCRUS)
  ISN 0020         10  X3 = X2 + STEP
  ISN 0021            X1 = X2 - STEP
  ISN 0022            CALL DISTN(X1,F1,HMAX,VCLIMB,WD,Q,STAGEM,WC,FI,TC,FC,VCRUS)
  ISN 0023            CALL DISTN(X3,F3,HMAX,VCLIMB,WD,Q,STAGEM,WC,FI,TC,FC,VCRUS)
                C
  ISN 0024         3    INTNUM = INTNUM + 1
  ISN 0025              IF (INTNUM .GT. 50) GO TO 89
                C
  ISN 0027            IF (F1 .LT. F2) GO TO 1
  ISN 0029            IF (F3 .LT. F2) GO TO 2
                C
                C  WHERE F2 IS LESS THAN F1 AND F3, FIT A PARABOLA, FIND MINIMUM AND
                C  USE THAT AS THE NEXT X2
                C
```

```
ISN 0031            X2 = X2 + .5*STEP*(F1-F3)/(F3-2.*F2+F1)
ISN 0032            CALL DISTN(X2,F2,HMAX,VCLIMB,WD,Q,STAGEM,WC,FI,TC,FC,VCRUS)
ISN 0033            IF (F2 .LE. CONV) GO TO 99
ISN 0035            STEP = STEP/4.
ISN 0036            GO TO 10
                 C
                 C  WHERE F2 IS BETWEEN OR GREATER THAN F1 AND F3, MAKE X2 THE VALUE
                 C  FOR A MINIMUM
                 C
ISN 0037          1     IF (F3 .LT. F1) GO TO 2
ISN 0039                  X2 = X1
ISN 0040                  F2 = F1
ISN 0041                  X1 = X1 - STEP
ISN 0042                  CALL DISTN(X1,F1,HMAX,VCLIMB,WD,Q,STAGEM,WC,FI,TC,FC,VCRUS)
ISN 0043                  GO TO 3
                 C
ISN 0044          2       X1 = X2
ISN 0045                  F1 = F2
ISN 0046                  X2 = X3
ISN 0047                  F2 = F3
ISN 0048                  X3 = X3 + STEP
ISN 0049                  CALL DISTN(X3,F3,HMAX,VCLIMB,WD,Q,STAGEM,WC,FI,TC,FC,VCRUS)
ISN 0050                  GO TO 3
                 C
ISN 0051          89      WRITE(6,302)
ISN 0052          302     FORMAT(' INTNUM = 50, CONTINUING ...')
                 C
                 C
ISN 0053          99    CONTINUE
ISN 0054                BLOCKF = FD + FC + (WC - WD) + .002*TOW
ISN 0055                BLOCKT = TD/60. + TC/3600. + FI*6072./(VCRUS*3600.) + .25
                 C
ISN 0056                RETURN
ISN 0057                END
```

```
LEVEL 21.8 ( JUN 74 )                    OS/360  FORTRAN H                                          DATE  82.016/13.52.25

        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                           SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
              C
              C
              C
  ISN 0002          SUBROUTINE DOC(STAGE,BLOCKT,BLOCKF,CENT)
              C
  ISN 0003          COMMON/VARIAB/TOW,S,SLSHP,AR,VCLIMB,HCRUS
  ISN 0004          COMMON/WEIGHT/OWEPL,AFWT
              C
              C  COMPUTES THE DIRECT OPERATING COSTS FOR A TURBOPROP FOR THE
              C  STAGE LENGTH PRESCRIBED IN THE SUBROUTINE BLOCK .  USES THE
              C  1967 ATA METHOD.  CONSTANTS INCLUDE NO OF ENGINES (2), NO OF
              C  CREW (2) AND NO OF PASSENGERS (30).
              C
              C  THE ENGINE COST IS DETERMINED BY FITTING THE LOCKHEED SHP
              C  VS COST PER SHP CURVE FOUND IN THEIR 1980 COMMUTER STUDY.
              C  THE COSTS ARE INFLATED BY 25% TO ACCOUNT FOR 1979 DOLLARS.
              C  AIRFRAME WEIGHT IS ASSUMED TO BE 200 DOLLARS FOR EACH POUND
              C  OF AIRFRAME.
              C
              C  DEFINE STAGE LENGTH IN STATUTE MILES, PRICE OF FUEL
              C
  ISN 0005          DOLGAL = 1.5
  ISN 0006          SLSHH = SLSHP/2.
  ISN 0007          STAGES = 1.15 * STAGE
              C
              C  BLOCK SPEED
              C
  ISN 0008          BLOCKS = STAGES/BLOCKT
  ISN 0009          T1 = BLOCKT - .25
              C
              C  ENGINE AND AIRFRAME ACQUISITION COSTS
              C
  ISN 0010          COSTEN=2.5*(61.747+1.65592E5/SLSHH- 8.38354E7/SLSHH**2)*SLSHH
  ISN 0011          COSTAF = 200. * AFWT
              C
              C  CREW COST
              C
  ISN 0012          CRWCST = (.05*(TOW/1000.)+63.)/BLOCKS
              C
              C  FUEL AND OIL COST
              C
  ISN 0013          FULCST = 1.02*(BLOCKF*DOLGAL/6.7 + 2.*.135*10.*BLOCKT)/STAGES
              C
              C  INSURANCE COST
              C
  ISN 0014          XINCST = .02*(COSTAF+COSTEN)/(2800.*BLOCKS)
              C
              C  MAINTENANCE COST
              C  AIRFRAME LABOR
              C
  ISN 0015          XLABAF = .05*AFWT/1000. + 6. - (630./(AFWT/1000.+120.))
  ISN 0016          XLABAH = .59 * XLABAF
  ISN 0017          AFLAB = (XLABAH*T1+XLABAF)/STAGES * 12.
              C
```

```
               C  AIRFRAME MATERIAL
               C
ISN 0018             AFMAT = ((3.08*COSTAF*T1)+(6.24*COSTAF))/(1E6*STAGES)
               C
               C  ENGINE LABOR
               C
ISN 0019             XLABEF = (.65+(.03*SLSHH)/1000.)*2.
ISN 0020             XLABEH = (.3+(.03*SLSHH)/1000.)*2.
ISN 0021             ENGLAB = (XLABEF*T1+XLABEH)/STAGES * 12.
               C
               C  ENGINE MATERIALS
               C
ISN 0022             ENGMAT = ((2.5*COSTEN)*T1 + 2.*COSTEN)/(1E5*STAGES)
               C
               C  MAINTENANCE BURDEN
               C
ISN 0023             BURDEN = 1.8 * (AFLAB + ENGLAB)
               C
               C  TOTAL MAINTENANCE COST
               C
ISN 0024             TOTMAI = AFLAB + AFMAT + ENGLAB + ENGMAT + BURDEN
               C
               C  DEPRECIATION COST
               C
ISN 0025             DEPR= ((COSTAF+COSTEN)+.1*COSTAF+.4*COSTEN)/(BLOCKS*15.*2800.)
               C
               C  CENTS PER SEAT STATUTE MILE
               C
ISN 0026             CENTS = (CRWCST+FULCST+XINCST+TOTMAI+DEPR)*100./30.
               C
               C  CENTS PER SEAT NAUTICAL MILE
               C
ISN 0027             CENT = CENTS * 1.15
               C
ISN 0028             RETURN
ISN 0029             END
```

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                                  SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                      C
                      C
                      C
   ISN 0002                 SUBROUTINE DISTN(X,F,HMAX,VEQ,WD,Q,STAGEM,WC,F1,TC,FC,VCRUS)
                      C
   ISN 0003                 COMMON/CHARAC/CDP,E,B
   ISN 0004                 COMMON/CONST/PI,G
   ISN 0005                 COMMON/VARIAB/TOW,S,SLSHP,AR,VCLIMB,HCRUS
                      C
                      C  SUBROUTINE TO COMPUTE THE GOAL FUNCTION (OR THAT TO BE MINIMIZED)
                      C  FOR THE BLOCK TIME AND FUEL DETERMINIZATION.
                      C
   ISN 0006                 ETA = .8
   ISN 0007                 CDPO = CDP
   ISN 0008                 CALL CLIMB(S,SLSHP,X,HMAX,VEQ,CDPO,ETA,FC,TC,DC,GC,WC)
                      C
   ISN 0009                   DAV = CDP*Q*S + (WD**2+WC**2)/(2.*PI*Q*B**2*E)
   ISN 0010                   SHPAV = DAV*VCRUS/(550.*.85)
   ISN 0011                   SHPMAX = POWER(SLSHP,VCRUS,HMAX)
   ISN 0012                   Z = SHPAV/SHPMAX
   ISN 0013                   SFC = .43 + 2.07E-2/Z + 2.02E-2/Z**2 - 1.04E-2/Z**3
                      C
   ISN 0014                 A1 = (325.*.85*SQRT(PI*E)*B)/(SQRT(CDP*S)*SFC)
   ISN 0015                 A2 = SQRT(CDP*S*PI*E)*Q*B
                      C
   ISN 0016                 F1 = A1*(ATAN(WC/A2) - ATAN(WD/A2))
   ISN 0017                 F = (DC/6072. + F1 - STAGEM)**2
                      C
   ISN 0018                 RETURN
   ISN 0019                 END
```

```
          COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=58,SIZE=0000K,
                             SOURCE,EBCDIC,NOLIST,NODECK,LOAD,MAP,NOEDIT,ID,NOXREF
                 C
                 C
                 C
   ISN 0002            SUBROUTINE DISTR(X,F,HMAX,EAS,WD,Q,MAXRNG,WC,F1,VCRUS,ZFH,HTAF)
                 C
   ISN 0003              REAL MAXRNG
   ISN 0004            COMMON/CONST/PI,G
   ISN 0005            COMMON/CHARAC/CDP,E,B
   ISN 0006            COMMON/VARIAB/TOW,S,SLSHP,AR,VCLIMB,HCRUS
   ISN 0007            COMMON/GEOM/XMAC,SHE,SHG,BH,SVE,SVG,BV
   ISN 0008            COMMON/WEIGHT/OWEPL,AFWT
   ISN 0009              COMMON/FLAG/KANT
                 C
                 C  FUNCTION EVALUATION FOR FINDING THE TOW FOR THE REQUIRED RANGE.
                 C
   ISN 0010            ETA = .8
   ISN 0011            CDPO = CDP
   ISN 0012            CALL POUNDS(X,S,SLSHP,ZFH,HTAF)
   ISN 0013            CALL DESENT(SLSHP,S,HMAX,ETA,FD,TD,DD,WD,ZFH)
   ISN 0014            CALL CLIMB(S,SLSHP,X,HMAX,EAS,CDPO,ETA,FC,TC,DC,GC,WC)
   ISN 0015              IF (WD .GT. WC) KANT = 1
   ISN 0017              IF (KANT .NE. 0) GO TO 10
                 C
                 C
   ISN 0019              DAV = CDP*Q*S + (WC**2+WD**2)/(2.*Q*PI*B**2*E)
   ISN 0020              SHPAV = DAV*VCRUS/(550.*.85)
   ISN 0021              SHPMAX = POWER(SLSHP,VCRUS,HMAX)
   ISN 0022              Z = SHPAV/SHPMAX
   ISN 0023              SFC = .43 + 2.07E-2/Z + 2.02E-2/Z**2 - 1.04E-3/Z**3
                 C
   ISN 0024            A1 = 325.*(.85/SFC)*B*SQRT(PI*E/(CDP*S))
   ISN 0025            A2 = Q*B*SQRT(CDP*PI*E*S)
                 C
   ISN 0026            F1 = A1*(ATAN(WC/A2) - ATAN(WD/A2)) + (DD + DC)/6072.
   ISN 0027              GO TO 20
                 C
                 C  IF AIRPLANE CANNOT CLIMB, LET THE RANGE BE EQUAL TO GAMMAC.
                 C
   ISN 0028       10     F1 = GC
                 C
   ISN 0029       20     F = (F1 - MAXRNG)**2
                 C
   ISN 0030              RETURN
   ISN 0031              END
```