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

Progress Report for
NASA Research Grant NAG-1-202

Valana L. Wells and Richard S. Shevell

(NASA-CR-168439) USE OF OPTIMIZATION TO PREDICT THE EFFECT OF SELECTED PARAMETERS ON COMMUTER AIRCRAFT PERFORMANCE Progress Report (Stanford Univ.) 45 p HC A03/MF A01 CSCL 01C G3/05

N82-17151

Unclas 08935

Department of Aeronautics and Astronautics
Stanford University
Stanford, California 94305

February 1982

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

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 function evaluation program, which comprises the bulk of the calculations involved in the optimizing process, acts as a mathematical aircraft model. This routine determines, for prescribed wing area, aspect ratio, and engine power, the complete geometry, performance, and operating cost of the resulting airplane. It employs preliminary design methodology for estimating zero-lift equivalent parasite drag area, tail sizes, $C_{L_{max}}$, airplane efficiency factor, lift-curve slope, structural load factors, and component weights. (See references 1 and 6.) For simplicity, such parameters as thickness ratios and sweep angle of the aerodynamic surfaces are held constant since compressibility effects are essentially negligible at commuter aircraft speeds. Also, taper ratios and tail aspect ratios are treated as constants.

The program uses preliminary design methods (reference 1) to calculate takeoff and landing distances but uses analytically derived expressions for determining range, climb performance, descent performance and climb gradients. Some of the resulting integrals necessitate numerical routines for their solution. The program determines best climb speed and cruise altitude by conducting a grid search of five speeds and three altitudes and choosing the lowest cost combinations of the two.

The direct operating cost calculation is based on the 1967 ATA DOC method (reference 2) with corrections for inflation and commuter operation. It assumes that commuter pilot pay rates run 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	from ref. 4; inflated 25%
Fuel Cost	$1.50/gallon
Oil Cost	$10/lb.

Appendix I contains a complete listing of the program.

III. Results

The results of the optimization program show that the airplane with the lowest direct operating cost flies at 290 knots 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 knots 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 knots 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, is often restricted by this regulation if it is designed according to part 25 rules.

At the higher speed, in most cases, minimum cruise power to fly at 330 knots 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 knots 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 knots 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 knots airplanes have higher aspect ratios than the 290 knots 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 knots aircraft. The 330 knots 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 knots and 290 knots 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 cr ise drag, and the curve indicating a 330 knots 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 knots 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.

IV. 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.

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.

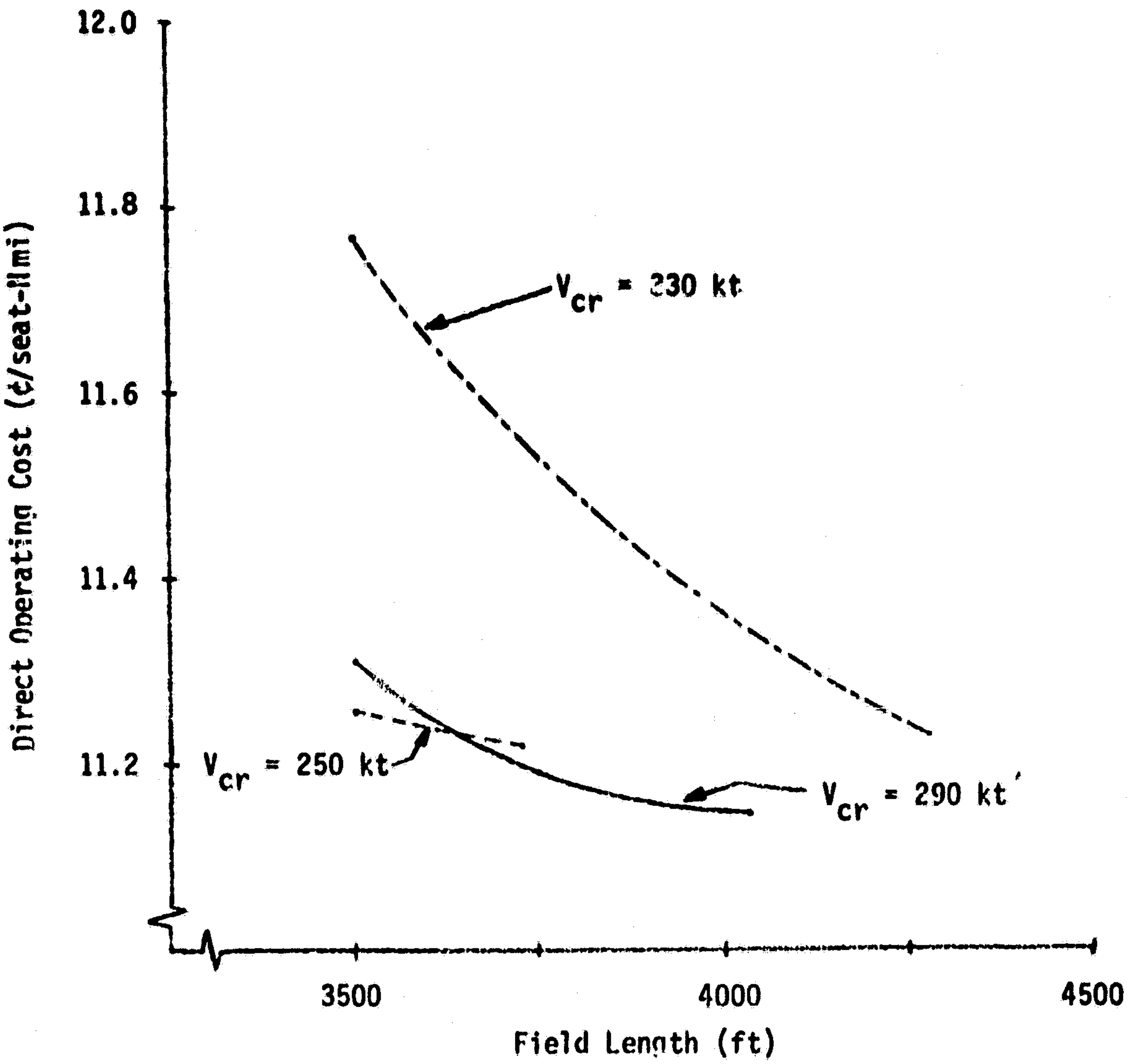

Figure 1. Direct Operating Cost vs. Field Length

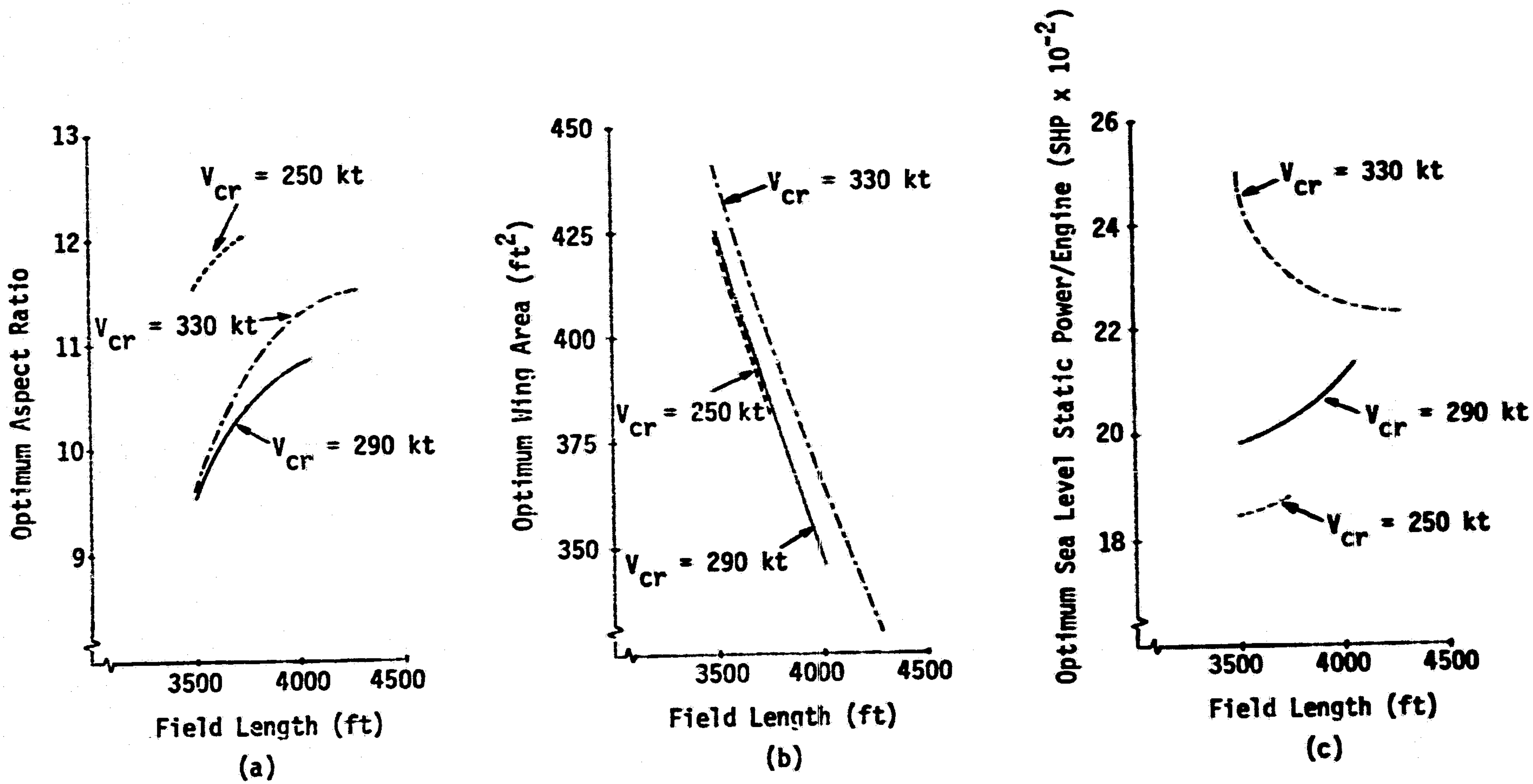

Figure 2. Optimal Variable Values vs. Field Length

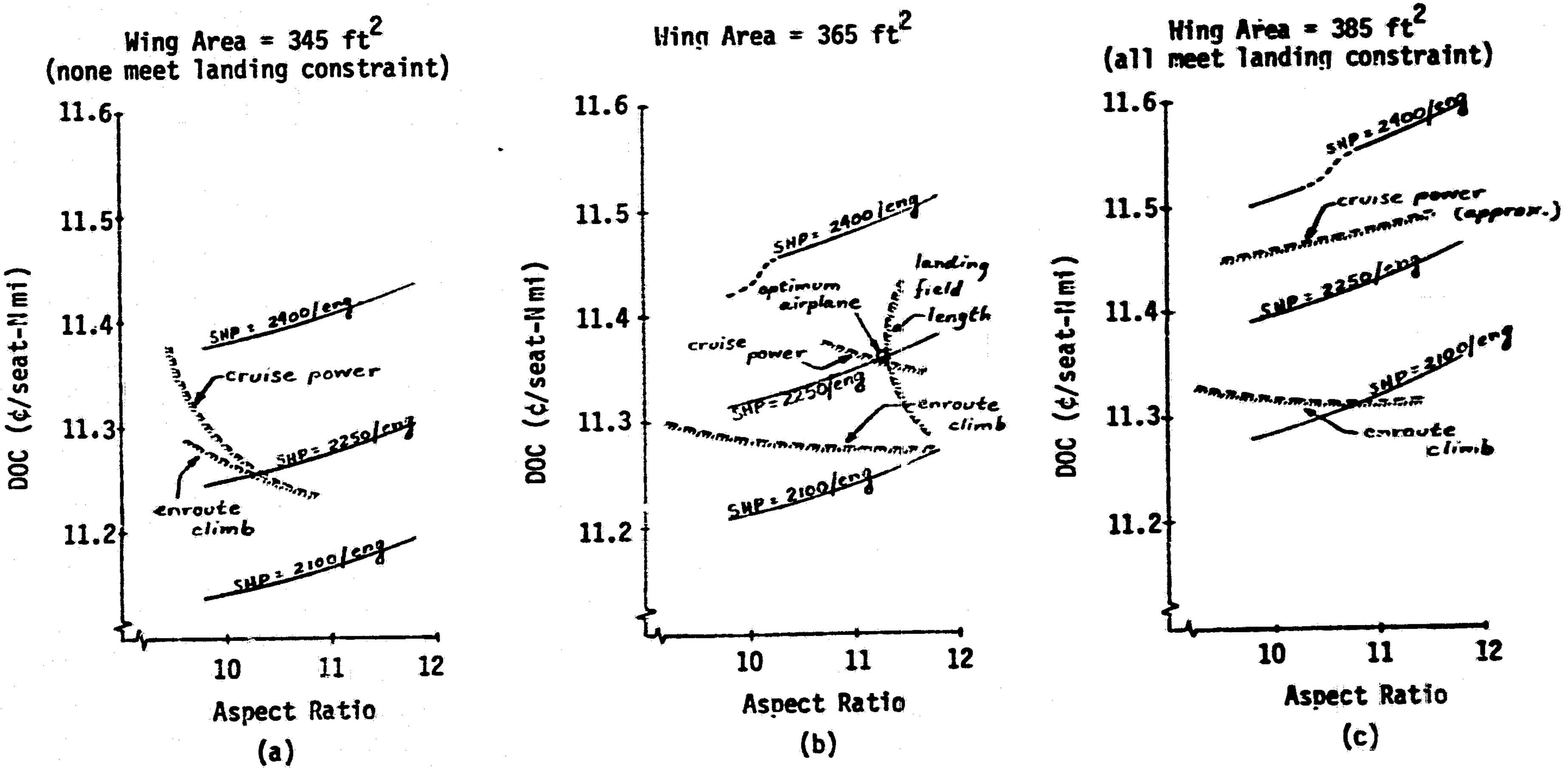

Figure 3. Optimal Point Sensitivities
Cruise Speed = 330 kt, Maximum Field Length = 4000 feet

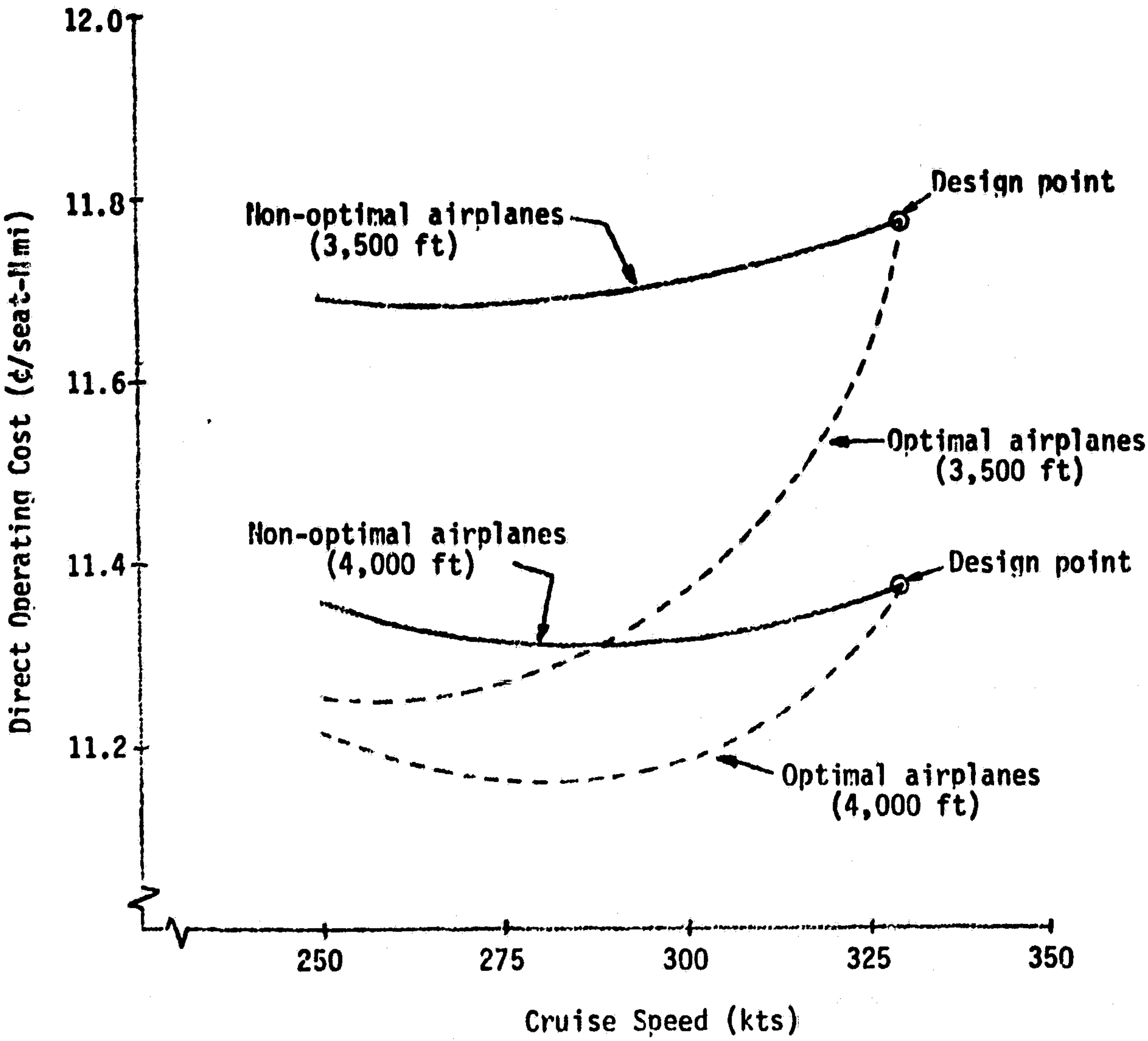

Figure 4. Cost for Non-Optimal Operation
[330 kt designs flown at lower speeds]

Appendix I

Program Listing

```
          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),B1(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 B1 AS THE IDENTITY MATRIX.
                C
ISN 0027         3      DO 20 I = 1,N
ISN 0028                DO 21 J = 1,N
ISN 0029                B1(I,J) = 0.
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
ISN 0030     21      CONTINUE
ISN 0031             BI(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,F0
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 - BI(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)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
ISN 0105          CALL INFO(FLEN)                166.
            C                                    167.
ISN 0106     500  CONTINUE                       168.
ISN 0107      99  STOP                           169.
ISN 0108          END                            170.
```

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
               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/ONEPL,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          ONEPL1 = 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          ONEPL = ONEPL1
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
```

```
        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/VARIAD/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 HAS 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
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
LEVEL 21.8 ( JUN 74 )                    OS/360  FORTRAN H                         DATE  82.016/13.52.00

        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 LANDNG(XLFL)
              C
 ISN 0003           COMMON/VARIAD/TOW,S,SLSHP,AR,VCLIMB,HCRUS
              C
              C COMPUTES LANDING DISTANCE USING LINEAR FITTED CURVE OF LANDING
              C DISTANCE VS STALL SPEED SQUARED.  ASSUMES DOUBLE SLOTTED FLAPS BUT
              C NO SLATS.  MAX LANDING WEIGHT EQUALS TOW.
              C
 ISN 0004             VS2 = 2.*TOW/(.002244*2.67*S*1.689**2)
 ISN 0005             XLFL = .4*VS2 + 750.
              C
 ISN 0006           RETURN
 ISN 0007           END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
        COMPILER OPTIONS - NAME=  MAIN,OPT=02,LINECNT=50,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 TWIN 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
```

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 ENRCLM(GAMER)
            C
ISN 0003          COMMON/VARIAB/TOW,S,SLSHP,AR,VCLIMB,HCRUS
ISN 0004          COMMON/CHARAC/CDP,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 WEIGHT 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 + .0000*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
```

```
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/TON,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*(1.0667*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
```

```
         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*TCW + 100.*TCW**4
         C        WRITE(6,*)SWET,CF,K,TCW,Z
         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:
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          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
```

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 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
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
               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
               C
ISN 0002             REAL FUNCTION DRAG(H,W,V,CDPO,S)
               C
ISN 0003             COMMON/CHARAC/CDP,E,B
ISN 0004             COMMON/CONST/PI,G
               C
               C  COMPUTES DRAG FOR A GIVEN FLIGHT CONDITION
               C
ISN 0005             RHO = 2.3769E-3*(1 - 6.8634E-6*H)**(4.2648)
ISN 0006             Q = .5*RHO*V**2
               C
ISN 0007             DRAG = CDPO*Q*S + (W**2)/(Q*PI*(B**2)*E)
               C
ISN 0008             RETURN
ISN 0009             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 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 = .993*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 = ONEPL/.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
```

```
        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/ONEPL,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)**(4.2648)
  ISN 0012          Q = .5 * RHO * VCRUS**2
              C
              C  CALCULATE TIME, FUEL, AND DISTANCE TO DESCEND ASSUMING WEIGHT
              C  AT END OF DESCENT IS ONE + PAYLOAD + RESERVES.
              C
  ISN 0013          CALL DESENT(SLSHP,S,HMAX,ETA,FD,TD,DD,WD,ONEPL)
              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
```

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 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,ZFW,WTAF)
                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,ZFW,WTAF)
  ISN 0013            CALL DESENT(SLSHP,S,HMAX,ETA,FD,TD,DD,WD,ZFW)
  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
```