ASD-TR-68-19

AD 675643

COMPUTER PROGRAM FOR REDUCING STATIC PROPELLER TEST DATA

GERALD T. CAFARELLI
MATTHEW H. CHOPIN

TECHNICAL REPORT ASD-TR-68-19

JUNE 1968

DEPUTY FOR ENGINEERING
AERONAUTICAL SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This document has been approved for public release and sale; its distribution is unlimited.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

100 - Sept. 1968 - CO455 - 48-1041

ASD-TR-68-19

COMPUTER PROGRAM FOR REDUCING STATIC PROPELLER TEST DATA

GERALD T. CAFARELLI
MATTHEW H. CHOPIN

FOREWORD

This report was prepared by the V/STOL Propulsion Branch, Directorate of Propulsion and Power Subsystems Engineering, Deputy for Engineering, Aeronautical Systems Division, under System 478A. Work was accomplished from July 1966 to November 1967. The authors served as project engineers.

This report was released by the authors in April 1968.

The authors wish to express their appreciation to Major Allan Gay, Directorate of Computation Services, Deputy for Engineering, for developing the computer program and applying the curve-fit technique to the data reduction as described in this report.

This technical report has been reviewed and is approved.

JAMES G. BARRETT
Technical Director
Directorate of Propulsion and
Power Subsystems Engineering

ABSTRACT

A computer program using a curve-fit technique was developed to reduce performance data obtained from static tests of aircraft propellers. The entire program is written in Fortran IV language for use on the IBM 7094 computer located at Wright-Patterson AFB, Ohio.

The program accepts static whirl rig test data (ie, raw RPM, horsepower, and thrust data) obtained at a fixed blade angle and reduces it into pertinent propeller relationships. The program first reduces the test data into various coefficients and computes the propeller tip Mach number. A curve fit technique then fits running curves through the test thrust and horsepower data points at the test tip Mach numbers. Intermediate horsepower and thrust values are determined from the fitted curves at selected Mach number increments, and all coefficients are recomputed. This results in a presentation of the reduced data in two forms, coeffcients computed from the actual test data and coefficients obtained at specific constant Mach number increments from the fitted curves. The data is presented in tabular printout form. This is a general program and is written so that the order of the curve fit, the Mach number increment, the number of test data points and the Mach number range can be varied.

The entire program deck and all nonstandard subroutines are included. Detail instructions are provided which should allow the program to be used by technicians or students who are familiar with Fortran IV language.

CONTENTS

ILLUSTRATIONS

SECTION I

INTRODUCTION

This program was developed to reduce data accumulated during an extensive series of static tests of propellers for the XC-142A V/STOL aircraft conducted at Wright-Patterson Air Force Base, Ohio. These tests are described in detail in a report being prepared entitled "Propeller Static Performance Tests for V/STOL Aircraft." The program is complete in that it contains all non-standard subroutines.

The program reduces RPM and corrected horsepower and thrust data at any given blade angle and computes Power Coefficient (C_p), Thrust Coefficient (C_t), C_t/C_p, Figure of Merit (F.M.) and thrust/horsepower (TH/HP). Propeller tip Mach number is calculated using propeller diameter, RPM and ambient temperature. A curve fit technique is then employed to fit a running selected degree polynomial using least squares to NPHP (horsepower) or NPTH (thrust) consecutive test data points (see NAMELIST) at the test tip Mach numbers. In this case, a 2nd degree polynomial was fit to 6 consecutive data points. Intermediate horsepower and thrust values are determined from the fitted curves at the selected Mach number increments. The routine proceeds until smoothed curves for horsepower (HPI) and thrust (THI), and derived coefficient values plus corresponding RPM and tip speed (TIPS), have been computed at all of the selected increments of Mach number. The procedure is essentially one of creating equal tabular entries through a smoothing technique rather than through an interpolative procedure.

The Namelist variables establish the boundaries of the program, such as the Mach number increments and range, number of data points fitted by the polynomial, and the degree of the polynomial. All variables can be changed to adapt the program to other uses.

SECTION II

COMPUTER PROGRAM

1. PROGRAM OPERATION

The computer program is designed to provide parameters for aircraft propellers at specified tip Mach number increments by fitting curves to data for horsepower and thrust versus Mach number at a set blade angle and recomputing values for the parameters at the predetermined Mach number increments. A flow chart showing the order of the program and the data input is given in Figure 1. Data is obtained from a test rig data sheet, such as is shown in Figure 2, and coded for the computer as shown in Figure 3.

For a given blade angle, the data must be presented to the computer at increasing values of RPM. After receiving the data input, the computer calculates the various parameters from the test data, including Mach number, C_t, C_p, C_t/C_p, F.M., and TH/HP, according to the appropriate equations (given in the list of program variables). Values are printed out on the output sheet, Figure 4, as Raw Data Points.

Subroutine XIPLSQ is then called to fit a 6-point, 2nd order, least squares curve through the data points for HP and TH at test tip Mach numbers. Then the computer determines values for HP and TH from the fitted curves that correspond to the selected Mach numbers, and uses these values to recompute values for all parameters at the selected Mach numbers. These values are then printed out on the output sheet as shown on Figure 4 under Fitted Curve Data for Constant Mach Number Increments.

The computer then reads in the data for a new blade angle (2nd Beta, 3rd Beta, etc.) and repeats the computations.

2. SUBROUTINE XIPLSQ

The least squares curve is fitted by means of Subroutine XIPLSQ. HPI and THI values are computed as functions of the independent variable BMACH by

ORDER OF INPUT SEQUENCE

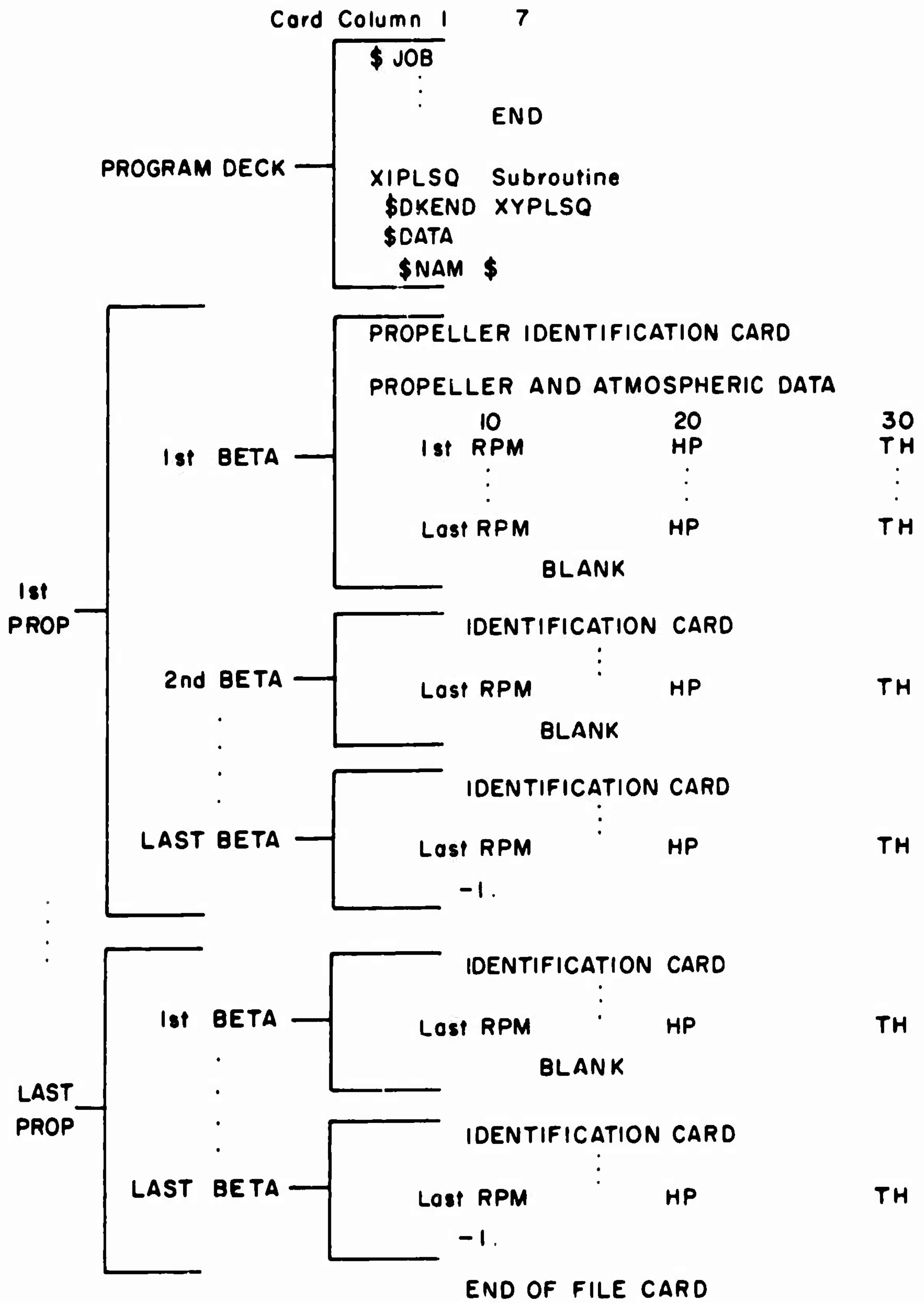

Figure 1. Sample Input Data

PROPELLER ELECTRIC MOTOR WHIRL TEST
(OBSERVED DATA - RIG NO. 4)

DATE: 7 Jan 66		MFR:	MOTOR: ☐ HIGH ☒ LOW	ENGINEER: Chopin - Schuetz
TASK NO.: 478A		HUB DWG:	SHAFT: ☒ HIGH ☐ LOW	CREW CHIEF:
BAROMETER: 29.35	TEMPERATURE: -4.0 / AIR FACTOR: 1.05	HUB SERIAL:	GEAR RATIO: 4 x 1	CONTROLS OPR: Howerton
TORQUE-MULTIPLIER: 20 x 1 (15,000 ft-lb Shaft)		BLADE DWG: 2FF16A1-4A	BRUSH TIME:	TORQUE OPR: McCabe
ROTATION: RH		BLADE SERIAL:	RIG TIME:	RIG THRUST OPR: Lee
ADAPTER: Nose				DUAL THRUST OPR: Steele
TABLE: 230	ANGLE: 12.1			BALCONY OPR:

TIME	MOTOR RPM	SHAFT RPM		TORQUE		HORSEPOWER			THRUST			CORRECTED DATA		
		IND	X	IND	FT-LBS	TOTAL	LOSS	NET	RIG	DUAL		RPM	HP	THRUST
				164		407	17	390				652	372	2762
				194		519	19	500				702	476	3257
				221		630	21	609				749	580	3714
				257		782	23	759				797	723	4314
				285		920	25	895				848	853	4790
				329		1125	27	1098				898	1046	5552
				380		1382	30	1352				955	1288	6352
				426		1625	32	1583				1003	1519	7105
				476		1905	34	1871				1052	1784	7876
				527		2200	36	2164				1096	2060	8695
				606		2624	39	2585				1148	2486	9838
				675		3070	41	3029				1194	2884	10857
				732		3430	43	3387				1230	3224	11619

Figure 2. Typical Rig Data Sheet

FORTRAN CODING FORM

Program		Punching Instructions		Page 1 of 1
Programmer	Date	Graphic / Punch ✓	Card Form #	Identification 73–80

C FOR COMMENT

FORTRAN STATEMENT

Statement Number	Cont.	Fortran Statement					
		2EF16A1-4A RUN 23A 7 JAN 66 (PRØP ID CARD)					
		(NBL) 4.	(AF) 105.0	(DIA) 15.625	(BETA) 12.	(TEMP °C) -4.0	(σ) 1.050
		(RPM) 652.	(HP) 372.	(TH) 2762.			
		702.	476.	3251.			
		749.	580.	3714.			
		797.	723.	4314.			
		848.	853.	4790.			
		898.	1046.	5552.			
		955.	1288.	6352.			
		1003.	1519.	7105.			
		1052.	1784.	7876.			
		1096.	2060.	8695.			
		1148.	2486.	9838.			
		1194.	2884.	10857.			
		1230.	3224.	11619.			

NOTE: Identifying information in ( ) is not coded.

Figure 3. Sample Coding Form
(Taken from Typical Rig Data Sheet)

```
STATIC PROP PERFORMANCE

    2FF15A1-4A 7JAN66 RUN TO 230 WALLS DOWN 15000 FT-LB

 BETA=12.1 AF= -0.  DIA=15.625 NBL=4 TEMPC= -4.0 TEMPR= 484.49 SIGMA=1.0500

***** RAW DATA POINTS *****

   RPM     HP    TH TMACH      RCT       RCP    RCT/CP      RFM    RTH/HP
  652.   372.  2762. 0.494   0.1655   0.0721   2.2964   0.7454   7.4247
  702.   476.  3257. 0.532   0.1683   0.0739   2.2786   0.7460   6.8424
  749.   530.  3714. 0.568   0.1686   0.0741   2.2752   0.7455   6.4034
  797.   723.  4314. 0.604   0.1730   0.0767   2.2559   0.7487   5.9658
  848.   853.  4790. 0.643   0.1696   0.0751   2.2589   0.7425   5.6155
  899.  1046.  5552. 0.681   0.1753   0.0775   2.2611   0.7555   5.3078
  955.  1283.  6352. 0.724   0.1774   0.0794   2.2342   0.7509   4.9317
 1003.  1517.  7105. 0.760   0.1799   0.0808   2.2255   0.7532   4.6774
 1052.  1784.  7876. 0.797   0.1812   0.0823   2.2032   0.7485   4.4149
 1096.  2060.  8695. 0.831   0.1843   0.0840   2.1945   0.7519   4.2209
 1148.  2435.  9838. 0.870   0.1901   0.0882   2.1551   0.7499   3.9574
 1194.  2884.10867. 0.905   0.1940   0.0910   2.1323   0.7494   3.7646
 1230.  3224.11619. 0.942   0.1956   0.0930   2.1028   0.7421   3.6039

***** FITTED CURVE DATA FOR CONSTANT MACH NUMBER INCREMENTS ***** (HP, 6 POINT 2ND ORDER.
                                                                   TH, 6 POINT 2ND ORDER)
   MACH      HP      TH    TIPS    RPM      CT     CP   CT/CP      FM   TH/HP
  0.725   1293.   6379.    783.    957.  0.1775 0.0793  2.239   0.753  4.934
  0.750   1453.   6892.    810.    990.  0.1793 0.0805  2.227   0.753  4.745
  0.775   1615.   7389.    837.   1023.  0.1800 0.0811  2.219   0.751  4.576
  0.800   1804.   7956.    864.   1056.  0.1819 0.0823  2.209   0.752  4.411
  0.825   2022.   8574.    891.   1088.  0.1843 0.0842  2.190   0.750  4.241
  0.850   2262.   9243.    918.   1121.  0.1872 0.0861  2.174   0.750  4.086
  0.875   2528.   9934.    944.   1154.  0.1898 0.0882  2.152   0.748  3.930
  0.900   2819.  10663.    971.   1187.  0.1926 0.0904  2.131   0.746  3.784
  0.925   3134.  11430.    998.   1220.  0.1954 0.0926  2.111   0.745  3.647
```

Figure 4. Typical Computer Printout

means of a polynomial smoothing process; this process resembles nonlinear interpolation in that it considers several points of the HP and TH versus AMACH tabular data function on either side of the desired BMACH value in determining a least squares polynomial, and this polynomial is then used to determine the smoothed HPI and THI values.

Smoothing in the vicinity of the ends of the test data array will not produce as satisfactory results as in the center because there are not sufficient data points available on both sides of the desired BMACH argument. The polynomial is always fit to the predetermined number of test data points, however, so additional data points from the other side are used to augment the least squares input data. If the desired BMACH value is out of range of the test data, however, extrapolative smoothing would be required, which would give results of even less reliability.

Note that if the fitted polynomial is of the Nth degree, N + 1 data points are required to develop the smoothing polynomial.

3. DATA INPUT

The first data input card (propeller identification card) shows which propeller data is being computed. The second card gives the basic propeller data and atmospheric conditions at the time that the test was run, including the number of blades, propeller diameter, activity factor, blade angle setting, temperature (°C), and density ratio (air factor). The first RPM value with its corresponding HP and TH values is punched on the third card, and values for increasing RPMs are punched on succeeding cards. A blank card is placed after the last RPM-HP-TH card if another blade angle run is to be computed for that propeller; if no more blade angle runs are to be made, a "-1." card (with -1. punched within the first 10 spaces) follows.

SECTION III
PROGRAM PARAMETERS

1. NAMELIST

The variables in the Namelist control various aspects of the program. They are set at specified values by the DATA statement near the beginning of the program deck.

There is one Namelist, NAM, for this program, consisting of the following variables:

VL = VL + VS, - minimum selected Mach number for curve-fitted data (VL set at 0.500)

VH - maximum selected Mach number for curve fitted data (VH set at 1.000)

VS - increment increase in selected Mach numbers (VS set at 0.025)

NPTH - maximum number of raw thrust points considered at one time for a curve fit (NPHP set at 6)

NPHP - maximum number of raw horsepower points considered at one time for a curve fit (NPHP set at 6)

NDTH - order of the polynomial used for fitting the thrust curve. (NDTH is set at 2; that is, a running polynomial of the form $C_0 + C_1 x + C_2 x^2$ is used in smoothing the thrust data unless modified by parameter control.)

NDHP - see information for NDTH, above (set at 2).

Namelist values can be modified by the simple procedure of replacing one card. For example, if the value for VH (maximum Mach number) is to be changed from 1.000 to 0.900, the following card would be removed

Card Column 2

$ NAM $ (the first card after the $ DATA control card)

and the following card inserted:

Card Column 2

$ NAM VH = .9$

2. EQUATIONS

The following equations are used in computing the parameters.

$$TR = 1.8(TC + 273.16)$$

$$AMACH = \left(\frac{\pi}{60}\right) \frac{(RPM)}{(49.04)} \frac{(DIA)}{\sqrt{TR}}$$

$$CT = CT(TH) = 0.1518 \times 10^{7} \frac{TH}{(RPM)^{2}\ (DIA)^{4}}$$

$$CTI = CT(THI) \qquad \text{(see THI)}$$

$$CP = CP(HP) = 0.5 \times 10^{11} \frac{HP}{(RPM)^{3}\ (DIA)^{5}}$$

$$CPI = CP(HPI) \qquad \text{(see HPI)}$$

$$FM = FM(CT,\ CP) = 0.798 \frac{|CT|^{1.5}}{CP}$$; if CT < 0, FM is set at minus value.

$$FMI = FM(CTI,\ CPI)$$

3. OTHER PROGRAM VARIABLES

a. Nonsubscripted Variables

NBL - number of blades per propeller (reference data)

AF - activity factor (reference data)

DIA - diameter of propeller, feet

RPM - propeller speed (revolutions per minute)

CT - coefficient of thrust

CP - coefficient of power

FM - Figure of Merit

TIPS - actual tip speed, ft/sec

b. Subscripted Variables

HEAD - propeller identification

B - BETA, blade angle, degrees (reference data)

TC - TEMPC, ambient temperature (^{o}C)

S - "sigma" = (air factor) density ratio (σ)

TR - TEMPR, ambient temperature (^{o}R)

HP - corrected horsepower; (Corrected HP = $\frac{\text{Test HP}}{\sigma}$)

TH - corrected thrust, pounds (Corrected TH = $\frac{\text{Test TH}}{\sigma}$)

AMACH - actual tip Mach number

BMACH - selected Mach number for curve fit

THI - thrust values determined from the smoothed thrust curve.

HPI - horsepower values determined from the smoothed horsepower curve.

FMI - interpolated figure of merit

CTI - interpolated coefficient of thrust

CPI - interpolated coefficient of power

c. Counters

NP - number of propellers

NB - number of blade angles

N - number of cards per blade angle run

NV=J-1 - number of selected Mach numbers per run

d. Miscellaneous Parameters

AF - activity factor = $\frac{100,000}{16} \int_{0.2}^{1.0} \frac{b}{DIA} \left(\frac{r}{R}\right)^3 d\left(\frac{r}{R}\right)$

b - blade width, feet

R - total blade radius, feet

r - radius along blade, feet

σ - density ratio; $\sigma = \frac{\rho}{\rho_o}$ (reference data)

ρ - local density, lb sec^2/ft^4

ρ_o - sea level standard density, lb sec^2/ft^4

SECTION IV

COMPARISON OF CURVES

A comparison of the computer curve-fit data with hand-faired data for thrust and horsepower is shown in Figures 5 and 6. These figures show that the test data and the computer curve-fit provide very good agreement. The curve-fit results in a substantial smoothing of the data.

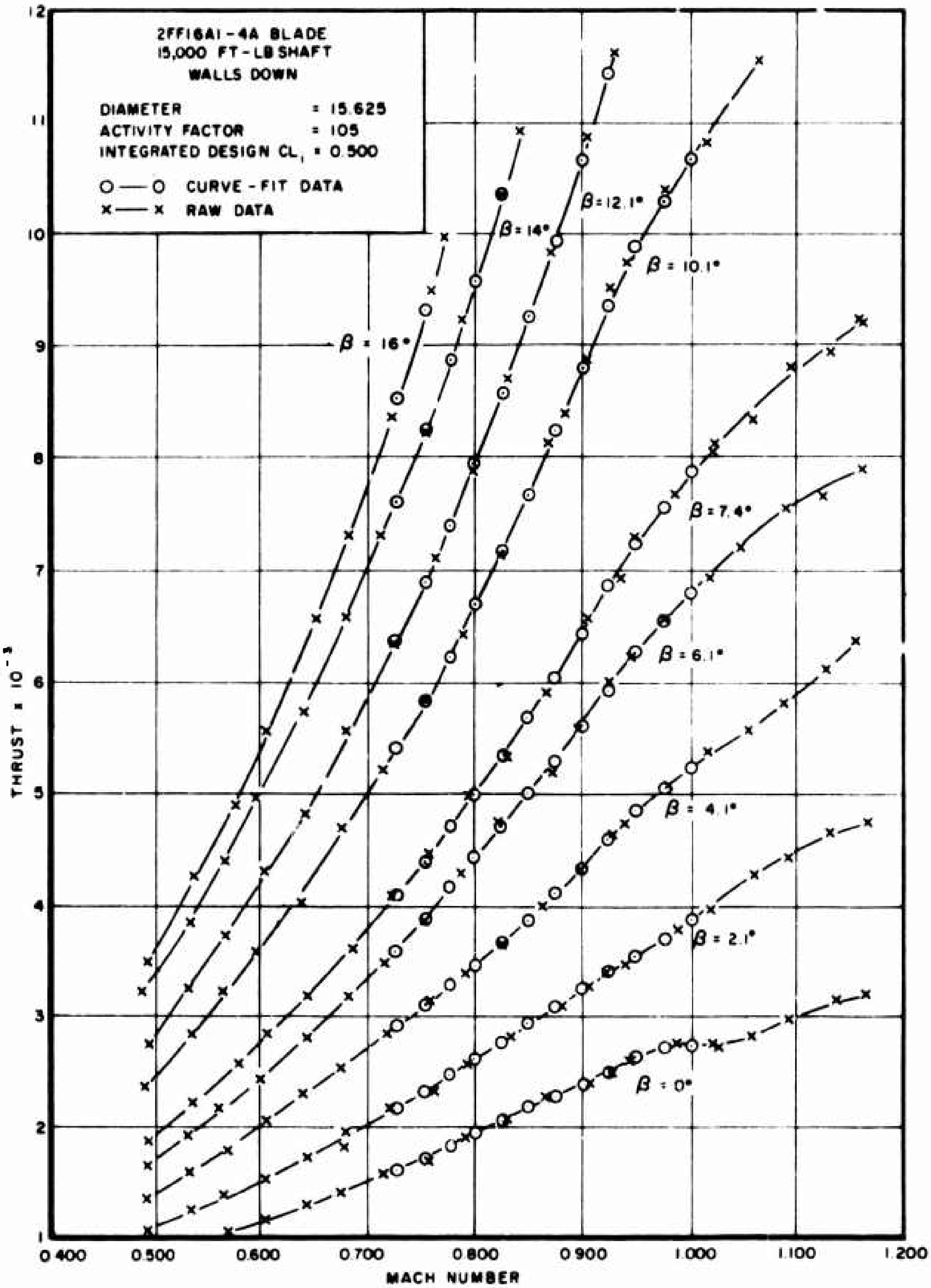

Figure 5. Comparison of Thrust Curve-Fit Data to Raw Data

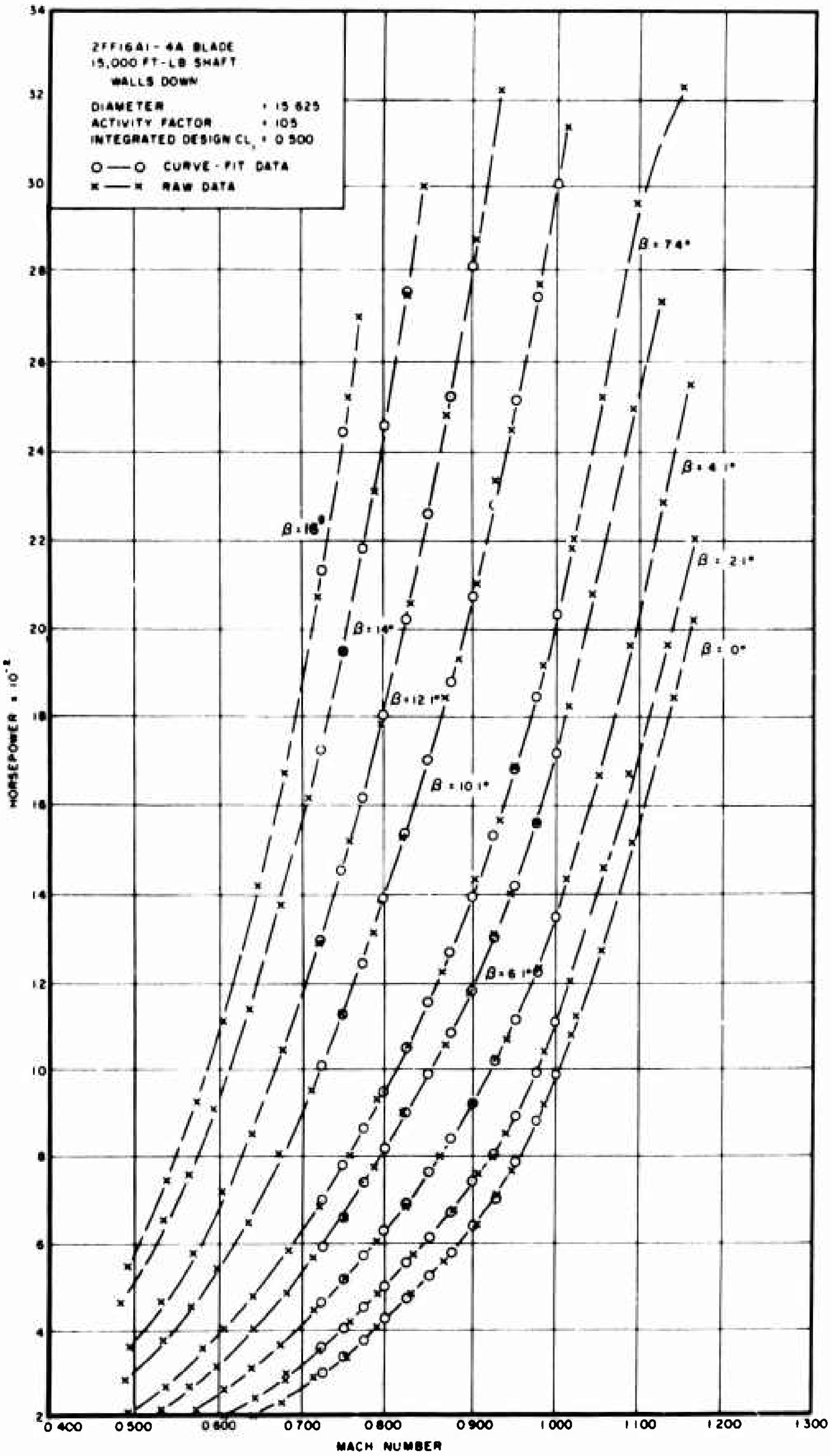

Figure 6. Comparison of Horsepower Curve-Fit Data to Raw Data

APPENDIX

COMPUTER PROGRAM

This appendix presents the basic computer program and the necessary subroutines, followed by an explanation of the subroutines.

```
$IBJOB PROP     MAP
$IBFTC CHOPN    DECK
      DIMENSION HEAD(13),B(30),TC(30),S(30),TR(30),HP(30),TH(30),
     1 AMACH(30),BMACH(60,60),THI(60),HPI(60),FMI(60,60),
     2 CTI(60,60),CPI(60,60)
      DATA ZC/273.16/,COFT/49.04/,COFCT/.1518E7/,COFCP/.5E11/,
     1 COFFMA.798/,VL/.5/,VH/1./,VS/.025/,NPTH/6/,NPHP/6/,
     2 NDTH/2/,NDHP/2/,LIST/0/
      NAMELIST/NAM/VL,VH,VS,NPTH,NPHP,NDTH,NDHP,LIST
C   NAMELIST NAM PARAMETERS,
C    VL(VS)VH = LOW, STEP, HIGH, MACH NUMBER TABULAR INTERVAL (SMOTHED).
C    NPTH, NPHP = NUMBER OF POINTS USED IN SMOTHING THRUST , HORSEPOWER.
C    NDTH, NDHP = DEGREE OF POLYNOMIAL USED IN SMOTHING TH AND HP.
C    LIST = 0,1 LEAST SQUARES ANALYSIS NOT USED (0), USED (1).
      PIO60=4.*DATAN(1.D0)/60.
   10 READ(5,NAM)
      NP=0
   15 NP=NP+1
      DO 20 K=1,30
      DO 20 L=1,30
      CTI(K,L)=0.
   20 CPI(K,L)=0.
      NB=0
   25 NB=NB+1
      READ(5,30)(HEAD(J),J=1,13)
   30 FORMAT(13A6)
      READ(5,35)NBL,AF,DIA,B(NB),TC(NB),S(NB)
   35 FORMAT(I10,5F10.5)
      TR(NB)=1.8*(TC(NB)+ZC)
      WRITE(6,40)(HEAD(J),J=1,13),B(NB),AF,DIA,NBL,TC(NB),TR(NB),S(NB)
   40 FORMAT(24H1STATIC PROP PERFORMANCE/////1H0,13A6/7H0 BETA=,F4.1,
     14H AF=,
     2 F5.1,5H DIA=,F6.3,5H NBL=,I1,7H TEMPC=,F5.1,7H TEMPR=,F7.2,
     3 7H SIGMA=,F6.4)
      WRITE(6,45)
   45 FORMAT(28H0***** RAW DATA POINTS *****/
     1 1H0,7X,3HRPM,4X,2HHP,4X,2HTH,1X,5HTMACH,7X,3HRCT,7X,3HRCP,4X,
     2 6HRCT/CP,7X,3HRFM,4X,6HRTH/HP)
      N=0
   50 N=N+1
      READ(5,55)RPM,HP(N),TH(N)
   55 FORMAT(3F10.1)
      TEST=RPM
      IF(TEST.LE.0.)  GO TO 75
      AMACH(N)=(PIO60*RPM*DIA)/(COFT*SQRT(TR(NB)))
      CT=COFCT*TH(N)/(RPM**2*DIA**4)
```

```
      CP=COFCP*HP(N)/(RPM**3*DIA**5)
      CTOCP=CT/CP
      FM=COFFM*SIGN(ABS(CT)**1.5,CT)/CP
      THOHP=TH(N)/HP(N)
      WRITE(6,70)RPM,HP(N),TH(N),AMACH(N),CT,CP,CTOCP,FM,THOHP
 70   FORMAT(5X,3F6.0,F6.3,5F10.4)
      GO TO 50
 75   N=N-1
      WRITE(6,80)NPHP,NDHP,NPTH,NDTH
 80   FORMAT(       66H0***** FITTED CURVE DATA FOR CONSTANT MACH NUMBER
     1 INCREMENTS *****,5H (HP,,I2,6H POINT,I2,14HND ORDER.   TH,,I2,
     2 6H POINT,I2,9HND ORDER)/
     3                      1H0,6X,4HMACH,5X,2HHP,5X,2HTH,3X,4HTIPS,4X,
     4 3HRPM,6X,2HCT,5X,2HCP,2X,5HCT/CP,5X,2HFM,2X,5HTH/HP)
      J=0
      STEP=0.
 82   J=J+1
 83   STEP=STEP+1.
      BMACH(J,NB)=VL+STEP*VS
      IF(BMACH(J,NB).LT.AMACH(1))   GO TO 83
      IF(BMACH(J,NB).GT.AMACH(N).OR.BMACH(J,NB).GT.VH)   GO TO 90
      CALL XIPLSQ(AMACH,HP,BMACH(J,NB),HPI(J),
     1 N,MINO(N,NPHP),NDHP,LIST,LE)
      CALL XIPLSQ(AMACH,TH,BMACH(J,NB),THI(J),
     1 N,MINO(N,NPTH),NDTH,LIST,LE2)
      TIP=BMACH(J,NB)*COFT*SQRT(TR(NB))
      RPM=TIP/(PIO60*DIA)
      CTI(J,NB)=COFCT*THI(J)/(RPM**2*DIA**4)
      CPI(J,NB)=COFCP*HPI(J)/(RPM**3*DIA**5)
      CTOCP=CTI(J,NB)/CPI(J,NB)
      FMI(NB,J)=COFFM*SIGN(ABS(CTI(J,NB))**1.5,CTI(J,NB))/CPI(J,NB)
      THOHP=THI(J)/HPI(J)
      WRITE(6,85)BMACH(J,NB),HPI(J),THI(J),TIP,RPM,CTI(J,NB),
     1 CPI(J,NB),CTOCP,FMI(NB,J),THOHP
 85   FORMAT(5X,F6.3,4F7.0,F8.4,F7.4,3F7.3)
      GO TO 82
 90   NV=J-1
 95   IF(TEST.EQ.0.)   GO TO 25
 110  IF(TEST.EQ.-1.)   GO TO 15
      GO TO 10
      END
```

```
$IBFTC XYPLSQ  DECK
      SUBROUTINE XIPLSQ (X,Y,XI,YI,N,NP,LD,LIST,LE)
      DIMENSION X(1),Y(1),C(11)
      LE=0
      LD1=LD+1
      IF (N) 1,1,2
 1    LE=1
      RETURN
 2    IF (NP) 3,3,4
 3    LE=2
      RETURN
 4    IF (N-NP) 5,6,6
 5    LE=3
      RETURN
 6    IF (X(1)-X(N)) 7,7,8
 7    DO 11 I=1,N
      K1=I
      IF (X(I)-XI) 11,12,10
 11   CONTINUE
      GO TO 14
 8    DO 13 I=1,N
      K1=N
      IF (X(I)-XI) 10,12,13
 13   CONTINUE
 14   K1=N+1-NP
      GO TO 17
 10   K1=K1-1
 12   K1=K1-(NP-1)/2
      IF (K1) 9,9,15
 9    K1=1
      GO TO 17
 15   IF (K1+NP-1-N) 17,17,16
 16   K1=N+1-NP
 17   CALL PLSQ (X(K1),Y(K1),NP,LD,C,LIST,EMAX,ERMS,EMEQ)
      YI=C(1)
      DO 18 I=2,LD1
 18   YI=YI*XI+C(I)
      RETURN
      END
```

```
$IBFTC MTXEQ.  DECK
      SUBROUTINE MTXEQ(A,X,B,N,K)                                       MTXEQ001
C                                                                       MTXEQ002
C     MATRIX EQUATION SOLVER     (7094 FORTRAN IV)                      MTXEQ003
C                                                                       MTXEQ004
C     USAGE...                                                          MTXEQ005
C                                                                       MTXEQ006
C     TO SOLVE THE LINEAR SYSTEM      AX=B                              MTXEQ007
C                                                                       MTXEQ008
C     CALL MTXEQ(A,X,B,N,K)                                             MTXEQ009
C                                                                       MTXEQ010
C        WHERE A MUST BE DIMENSIONED N X N                              MTXEQ011
C              X MUST BE DIMENSIONED N X K                              MTXEQ012
C              B MUST BE DIMENSIONED N X K                              MTXEQ013
C              N IS THE NO. OF EQUATIONS (ROWS IN A,X,B)                MTXEQ014
C              K IS THE NO. OF SOLUTION VECTORS (COLS. IN X,B)          MTXEQ015
C                                                                       MTXEQ016
C     664 CELLS OF BLANK COMMON ARE USED.                               MTXEQ017
C                                                                       MTXEQ018
C     NOTE...   TO CHANGE DIMENSIONS OF ARRAYS C AND PIV, ALSO          MTXEQ019
C            CHANGE VALUES OF NMAX AND NKMAX IN DATA STATEMENT.         MTXEQ020
C                                                                       MTXEQ021
C                                                                       MTXEQ022
      DIMENSION   A(N,N), B(N,K), X(N,K)                                MTXEQ023
      COMMON   ATPE, I, IFROM, IP1, IPIV, ITO,                          MTXEQ024
     1          J, KP, L, NP, NP1, NPJ, NPK, RM                         MTXEQ025
      COMMON     PIV(26), C(24,26)                                      MTXEQ026
      DATA      NMAX, NKMAX/ 24, 26/                                    MTXEQ027
C                                                                       MTXEQ028
C     TEST N AND K FOR CORRECT RANGE                                    MTXEQ029
C                                                                       MTXEQ030
      IF ( N .LE. 0  .OR.  N .GT. NMAX )      GO TO 190                 MTXEQ031
      IF ( K .LE. 0  .OR.  (N+K) .GT. NKMAX )     GO TO 190             MTXEQ032
C                                                                       MTXEQ033
C     GET ARGUMENTS N AND K                                             MTXEQ034
C                                                                       MTXEQ035
      NP=N                                                              MTXEQ036
      KP=K                                                              MTXEQ037
C                                                                       MTXEQ038
C     MOVE ARRAYS A(I,J) AND B(I,J) INTO C(I,J)                         MTXEQ039
C                                                                       MTXEQ040
      DO 10 J=1,NP                                                      MTXEQ041
      DO 10 I=1,NP                                                      MTXEQ042
   10 C(I,J)=A(I,J)                                                     MTXEQ043
      DO 20 J=1,KP                                                      MTXEQ044
      NPJ=NP+J                                                          MTXEQ045
```

```
      DO 20 I=1,NP                                                      MTXEQ046
   20 C(I,NPJ)=B(I,J)                                                   MTXEQ047
C                                                                       MTXEQ048
C     SET TO PERFORM N ELIMINATION SWEEPS (I=1,N)                       MTXEQ049
C                                                                       MTXEQ050
      NP1=NP+1                                                          MTXEQ051
      NPK=NP+KP                                                         MTXEQ052
      DO 120 I=1,NP                                                     MTXEQ053
      IP1=I+1                                                           MTXEQ054
C                                                                       MTXEQ055
C     SEARCH FOR NEXT PIVOT ROW (I-TH PIVOT IS IN COL. I)               MTXEQ056
C                                                                       MTXEQ057
      ATPE=0.                                                           MTXEQ058
      DO 40 J=I,NP                                                      MTXEQ059
      IF (ABS(C(J,I))-ATPE) 40,30,30                                    MTXEQ060
   30 ATPE=ABS(C(J,I))                                                  MTXEQ061
      IPIV=J                                                            MTXEQ062
   40 CONTINUE                                                          MTXEQ063
C                                                                       MTXEQ064
C     OPERATE ON THE PIVOT ROW                                          MTXEQ065
C                                                                       MTXEQ066
      IF (ATPE) 210,210,50                                              MTXEQ067
   50 DO 60 J=IP1,NPK                                                   MTXEQ068
   60 PIV(J)=C(IPIV,J)/C(IPIV,I)                                        MTXEQ069
C                                                                       MTXEQ070
C     PERFORM ELIMINATIONS BELOW THE DIAGONAL (COL. I)                  MTXEQ071
C                                                                       MTXEQ072
      IFROM=NP                                                          MTXEQ073
      ITO=NP                                                            MTXEQ074
   70 IF (IFROM-IPIV) 80,100,80                                         MTXEQ075
   80 RM=-C(IFROM,I)                                                    MTXEQ076
      DO 90 J=IP1,NPK                                                   MTXEQ077
   90 C(ITO,J)=C(IFROM,J)+RM*PIV(J)                                     MTXEQ078
      ITO=ITO-1                                                         MTXEQ079
  100 IFROM=IFROM-1                                                     MTXEQ080
      IF (IFROM-I) 110,70,70                                            MTXEQ081
C                                                                       MTXEQ082
C     PUT THE I-TH PIVOT ROW IN THE VACATED ROW I                       MTXEQ083
C                                                                       MTXEQ084
  110 DO 120 J=IP1,NPK                                                  MTXEQ085
  120 C(I,J)=PIV(J)                                                     MTXEQ086
C                                                                       MTXEQ087
C     NOW DO THE BACK SOLUTION                                          MTXEQ088
C                                                                       MTXEQ089
      I=NP                                                              MTXEQ090
  130 IP1=I                                                             MTXEQ091
```

```
      I=I-1                                                             MTXEQ092
      IF (I) 160,160,140                                                MTXEQ093
  140 DO 150 J=NP1,NPK                                                  MTXEQ094
      DO 150 L=IP1,NP                                                   MTXEQ095
  150 C(I,J)=C(I,J)-C(I,L)*C(L,J)                                       MTXEQ096
      GO TO 130                                                         MTXEQ097
C                                                                       MTXEQ098
C     MOVE THE SOLUTION TO ARRAY X(I,J)                                 MTXEQ099
C                                                                       MTXEQ100
  160 DO 170 J=1,KP                                                     MTXEQ101
      NPJ=NP+J                                                          MTXEQ102
      DO 170 I=1,NP                                                     MTXEQ103
  170 X(I,J)=C(I,NPJ)                                                   MTXEQ104
  180 RETURN                                                            MTXEQ105
C                                                                       MTXEQ106
  190 WRITE (6,1000) NP,KP                                              MTXEQ107
      RETURN
  210 WRITE (6,1001)                                                    MTXEQ109
      RETURN                                                            MTXEQ111
 1000 FORMAT(3HON=I12,5H   K=I12,35H ARE INCORRECT FOR SUBROUTINE MTXEQ)MTXEQ112
 1001 FORMAT (37HODET(A)=0 IN CALL TO SUBROUTINE MTXEQ)                 MTXEQ113
      END                                                               MTXEQ114
```

```
$IBFTC PLSQ.    DECK
      SUBROUTINE PLSQ(X,Y,N,K,C,LIST,EMAX,ERMS,EMEQ)                    PLSQ0001
C                                                                       PLSQ0002
C     PLSQ      POLYNOMIAL LEAST SQUARE CURVE FIT                       PLSQ0003
C                                                                       PLSQ0004
C                                                                       PLSQ0005
C         PLSQ WILL FIT A GIVEN SET OF DATA TO A                        PLSQ0006
C     POLYNOMIAL OF DEGREE K OF THE FORM...                             PLSQ0007
C                                                                       PLSQ0008
C         Y=C(K+1)+C(K)*X+C(K-1)*X**2+...+C(2)*X**(K-1)+C(1)*X**K       PLSQ0009
C                                                                       PLSQ0010
C     PLSQ THEN COMPUTES THE MAXIMUM ERROR AND ROOT                     PLSQ0011
C     MEAN SQUARE ERROR OBTAINED BY USING THE C                         PLSQ0012
C     COEFFICIENTS TO RE-COMPUTE Y FROM X                               PLSQ0013
C                                                                       PLSQ0014
C     USAGE...                                                          PLSQ0015
C                                                                       PLSQ0016
C     DIMENSION X(N), Y(N), C(L)                                        PLSQ0017
C         WHERE L IS K+1                                                PLSQ0018
C     CALL PLSQ(X,Y,N,K,C,LIST,EMAX,ERMS,EMEQ)                          PLSQ0019
C                                                                       PLSQ0020
C        WHERE,                                                         PLSQ0021
C                                                                       PLSQ0022
C        X  IS THE ARRAY OF N INDEPENDENT VARIABLES                     PLSQ0023
C                                                                       PLSQ0024
C        Y  IS THE ARRAY OF N DEPENDENT VARIABLES                       PLSQ0025
C                                                                       PLSQ0026
C        N  IS THE NUMBER OF INDEPENDENT(DEPENDENT)                     PLSQ0027
C           VARIABLES                                                   PLSQ0028
C                                                                       PLSQ0029
C        K  IS THE DEGREE OF THE LEAST SQUARES POLYNOMIAL               PLSQ0030
C                                                                       PLSQ0031
C        C  IS THE ARRAY OF THE COEFFICIENTS,HIGH ORDER                 PLSQ0032
C           TO LOW ORDER, OF THE LEAST SQUARES POLYNOMIAL               PLSQ0033
C                                                                       PLSQ0034
C        LIST  =0   SUPPRESSES THE ERROR ANALYSIS OUTPUT                PLSQ0035
C              =1   GIVES THE ERROR ANALYSIS OUTPUT                     PLSQ0036
C                                                                       PLSQ0037
C        EMAX  IS THE MAXIMUM ABSOLUTE ERROR OBTAINED                   PLSQ0038
C              BY USING THE COMPUTED C COEFFICIENTS TO                  PLSQ0039
C              APPROXIMATE THE DEPENDENT VARIABLE                       PLSQ0040
C                                                                       PLSQ0041
C        ERMS  IS THE ROOT MEAN SQUARE ERROR OBTAINED                   PLSQ0042
C              BY USING THE COMPUTED C COEFFICIENTS TO                  PLSQ0043
C              APPROXIMATE THE DEPENDENT VARIABLE                       PLSQ0044
C                                                                       PLSQ0045
```

```
C        EMEQ  IS THE MAXIMUM DEVIATION FROM UNITY                    PLSQ0046
C              IN THE LINEAR SYSTEM CHECK SOLUTION                    PLSQ0047
C                                                                     PLSQ0048
C                                                                     PLSQ0049
C     PLSQ CALLS SUBROUTINE MTXEQ
C                                                                     PLSQ0051
C     PLSQ USES 1309 CELLS OF BLANK COMMON                            PLSQ0052
C                                                                     PLSQ0053
C                                                                     PLSQ0054
      COMMON    MTXEQT(664), CF, DIF, I, J, JC, JK,                   PLSQ0055
     *          L, LL, LU, M, SUM, XI, XM(576),                       PLSQ0056
     *          XMAX, XMIN, XP, YC, YM(48)                            PLSQ0057
      DIMENSION    X(N), Y(N), C(24),                                 PLSQ0058
     *             XDP(48), XYDP(24)                                  PLSQ0059
      EQUIVALENCE    (MTXEQT(1),XDP(1)), (MTXEQT(97),XYDP(1))         PLSQ0060
      LOGICAL    LIST                                                 PLSQ0061
      DOUBLE PRECISION    CF, XDP, XI, XMAX, XMIN, XP, XYDP           PLSQ0062
      DATA    KMAX/ 23/                                               PLSQ0063
C                                                                     PLSQ0064
C     CHECK K AND N FOR PROPER RANGE                                  PLSQ0065
C                                                                     PLSQ0066
      IF (K .GT. KMAX .OR. N .LE. K .OR. K .LE. 0)    GO TO 200       PLSQ0067
      L=K+1                                                           PLSQ0068
C                                                                     PLSQ0069
C     FIND MINIMUM AND MAXIMUM VALUES FOR X                           PLSQ0070
C                                                                     PLSQ0071
      XMIN=X(1)                                                       PLSQ0072
      XMAX=X(1)                                                       PLSQ0073
      DO 10 I=2,N                                                     PLSQ0074
      XMIN=AMIN1(XMIN,X(I))                                           PLSQ0075
   10 XMAX=AMAX1(XMAX,X(I))                                           PLSQ0076
C                                                                     PLSQ0077
C     ZERO DOUBLE PRECISION ARRAYS FOR SUMMING                        PLSQ0078
C                                                                     PLSQ0079
      M=2*K+1                                                         PLSQ0080
      DO 20 I=1,M                                                     PLSQ0081
   20 XDP(I)=0.0D+00                                                  PLSQ0082
      DO 25 I=1,L                                                     PLSQ0083
   25 XYDP(I)=0.0D+00                                                 PLSQ0084
C                                                                     PLSQ0085
C     TRANSFORM RANGE OF X TO (-1,+1) AND                             PLSQ0086
C     COMPUTE SUMS OF POWERS OF X AND SUMS                            PLSQ0087
C     OF Y TIMES POWERS OF X                                          PLSQ0088
C                                                                     PLSQ0089
      LL=K+2                                                          PLSQ0090
      LU=2*K+1                                                        PLSQ0091
```

```
      DO 40 I=1,N                                                       PLSQ0092
      XP=1.0D+00                                                        PLSQ0093
      XI=2.0D+00*(X(I)-XMIN)/(XMAX-XMIN)-1.0D+00                        PLSQ0094
      DO 30 J=1,L                                                       PLSQ0095
      XDP(J)=XDP(J)+XP                                                  PLSQ0096
      XYDP(J)=XYDP(J)+XP*Y(I)                                           PLSQ0097
   30 XP=XP*XI                                                          PLSQ0098
      DO 40 J=LL,LU                                                     PLSQ0099
      XDP(J)=XDP(J)+XP                                                  PLSQ0100
   40 XP=XP*XI                                                          PLSQ0101
C                                                                       PLSQ0102
C     STORE ABOVE COMPUTED SUMS IN ARRAY XM                             PLSQ0103
C     AND COMPUTE ROW SUMS FOR CHECK SOLUTION                           PLSQ0104
C                                                                       PLSQ0105
      DO 50 I=1,L                                                       PLSQ0106
      LL=I+L                                                            PLSQ0107
      YM(LL)=0.0                                                        PLSQ0108
      LU=(I-1)*L                                                        PLSQ0109
      JK=I-1                                                            PLSQ0110
      DO 50 J=1,L                                                       PLSQ0111
      JK=JK+1                                                           PLSQ0112
      JC=LU+J                                                           PLSQ0113
      XM(JC)=XDP(JK)                                                    PLSQ0114
   50 YM(LL)=YM(LL)+XM(JC)                                              PLSQ0115
      DO 60 I=1,L                                                       PLSQ0116
   60 YM(I)=XYDP(I)                                                     PLSQ0117
C                                                                       PLSQ0118
C     SOLVE THE SYSTEM XM*C=YM                                          PLSQ0119
C                                                                       PLSQ0120
      CALL MTXEQ(XM,YM,YM,L,2)                                          PLSQ0121
C                                                                       PLSQ0122
C     REORDER AND MOVE SOLUTION TO C AND FIND                           PLSQ0123
C     MAXIMUM ERROR IN CHECK SOLUTION                                   PLSQ0124
C                                                                       PLSQ0125
      EMEQ=0.0                                                          PLSQ0126
      DO 70 I=1,L                                                       PLSQ0127
      JK=K-I+2                                                          PLSQ0128
      C(JK)=YM(I)                                                       PLSQ0129
      JC=I+L                                                            PLSQ0130
   70 EMEQ=AMAX1(EMEQ,ABS(YM(JC)-1.0))                                  PLSQ0131
C                                                                       PLSQ0132
C     ADJUST COEFFICIENTS FOR ORIGINAL RANGE                            PLSQ0133
C     OF X                                                              PLSQ0134
C                                                                       PLSQ0135
      CF=(XMAX-XMIN)/2.0D+00                                            PLSQ0136
      DO 80 I=1,K                                                       PLSQ0137
```

```
      DO 80 J=1,I                                                       PLSQ0138
   80 C(J)=C(J)/CF                                                      PLSQ0139
      CF=(XMAX+XMIN)/2.0D+00                                            PLSQ0140
      DO 90 I=1,K                                                       PLSQ0141
      M=L-I+1                                                           PLSQ0142
      DO 90 J=2,M                                                       PLSQ0143
   90 C(J)=-CF*C(J-1)+C(J)                                              PLSQ0144
C                                                                       PLSQ0145
C     INITIATE PRINT OF ERROR ANALYSIS IF LIST .NE. 0                   PLSQ0146
C                                                                       PLSQ0147
      IF (LIST)    WRITE (6,1001)                                       PLSQ0148
C                                                                       PLSQ0149
C     COMPUTE MAXIMUM AND ROOT MEAN SQUARE ERRORS                       PLSQ0150
C     AND OUTPUT ERROR ANALYSIS IF LIST .NE. 0                          PLSQ0151
C                                                                       PLSQ0152
      EMAX=0.0                                                          PLSQ0153
      SUM=0.0                                                           PLSQ0154
      DO 130 I=1,N                                                      PLSQ0155
      YC=C(1)                                                           PLSQ0156
      DO 100 J=1,K                                                      PLSQ0157
  100 YC=YC*X(I)+C(J+1)                                                 PLSQ0158
      DIF=YC-Y(I)                                                       PLSQ0159
      IF (.NOT. LIST)    GO TO 120                                      PLSQ0160
      IF (I .GT. L)    GO TO 110                                        PLSQ0161
      WRITE (6,1002) I, X(I), Y(I), YC, DIF, C(I)                       PLSQ0162
      GO TO 120                                                         PLSQ0163
  110 WRITE (6,1002) I, X(I), Y(I), YC, DIF                             PLSQ0164
  120 EMAX=AMAX1(EMAX,ABS(DIF))                                         PLSQ0165
  130 SUM=SUM+DIF**2                                                    PLSQ0166
      ERMS=SQRT(SUM/FLOAT(N))                                           PLSQ0167
      IF (LIST)    WRITE (6,1003) EMAX, ERMS, EMEQ                      PLSQ0168
      RETURN                                                            PLSQ0169
C                                                                       PLSQ0170
C     GIVE ERROR MESSAGE AND RETURNS
C                                                                       PLSQ0173
  200 WRITE (6,1000) N,K                                                PLSQ0174
      RETURN                                                            PLSQ0176
 1000 FORMAT (3HON=,I12,3H K=,I12,29HINCORRECT FOR SUBROUTINE PLSQ)     PLSQ0177
 1001 FORMAT (1H1,20X,32HPLSQ    POLYNOMIAL LEAST SQUARE                PLSQ0178
     *24HCURVE FIT ERROR ANALYSIS//                                     PLSQ0179
     *6H0    I,11X,9HX - GIVEN,11X,9HY - GIVEN,11X,                     PLSQ0180
     *10HY - FITTED,12X,5HERROR,16X,4HB(I)//)                           PLSQ0181
 1002 FORMAT (1X,I5,8X,5(1PE14.6,6X))                                   PLSQ0182
 1003 FORMAT (1H0,9X,5HEMAX=,1PE15.6,9X,5HERMS=,E15.6,                  PLSQ0183
     *9X,5HEMEQ=,E15.6)                                                 PLSQ0184
      END                                                               PLSQ0185

$DATA
 $NAM $
$EOF
```

EXPLANATION OF SUBROUTINE MTXEQ

MTXEQ, Matrix Equation Solver, solves the system of linear equations expressed by the matrix equation AX = B, using single precision floating arithmetic.

Control:

To solve the matrix equation,

$$AX = B$$

the following calling sequence is used:

```
DIMENSION  A(N,N), X(N,K), B(N,K)
CALL MTXEQ(A,X,B,N,K)
```

where

A - is the N by N coefficient matrix. The A matrix is not destroyed.

X - will be the N by K solution matrix.

B - is the N by K right-hand-side matrix. The B matrix is not destroyed.

N - is the number of equations, i.e., rows in A, X and B. $(1 \leq N \leq 24)$

K - is the number of solution, i.e., columns in X and B. $(K > 0$ and $(N+K) \leq 26)$

Other Programming Information:

1. Other subprograms used - None.

2. Error conditions -

a. The following ranges of N and K are allowed:

$(1 \leq N \leq 24)$

$K > 0$ and $(N+K) \leq 26$

When violations of the above ranges are detected by MTXEQ, a message is written and FXEM is called. The upper limits on N and (N+K) may be altered by changing a DIMENSION and a DATA statement.

b. If the A matrix has DET(A) = 0, a message is written and FXEM is called.

c. Common usage -

MTXEQ uses 664 cells of blank common.

Method:

The matrix $C = \left[A,B\right]$ is formed in blank common. Subsequent operations are performed on the C matrix, thus preserving the A and B matrices. Elementary row operations are performed to eliminate elements below the main diagonal of the augmented matrix C. To reduce the propagation of round-off error, the i-th pivot element is chosen to be an element having maximum magnitude in the i-th sub-column. Finally, a back solution gives the desired matrix, X. The number of floating point arithmetic operations required using this method is of the order of $2/3\ (N^3 + 3KN^2)$.

EXPLANATION OF SUBROUTINE PLSQ

PLSQ, Polynomial Least Square Curve Fit, will fit a given set of data to a polynomial of degree K of the form:

$$Y = C_1X^k + C_2X^{k-1} + \ldots + C_kX + C_{k+1}$$

An error analysis of the fit is optionally given.

Control:

DIMENSION X(N), Y(N), C(L)

where L is K+1

CALL PLSQ(X,Y,N,K,C,LIST,EMAX,ERMS,EMEQ)

where

X is the array of N independent variables.

Y is the array of dependent variables.

N is the number of independent (dependent) variables.

K is the degree of the least squares polynomial.

C is the array of the coefficients, high order to low order, of the least squares polynomial.

LIST = 0 suppresses the error analysis output.

= 1 gives the error analysis output.

EMAX is the maximum absolute error obtained by using the computed C coefficients to approximate the dependent variable.

ERMS is the root mean square error obtained by using the computed C coefficients to approximate the dependent variable.

EMEQ is the maximum deviation from unity in the linear system check solution.

Other Programming Information:

1. Other subroutines used:

   PLSQ calls MTXEQ to solve the normal equations and FXEM for return to the system on error conditions.

2. Error conditions:
   PLSQ checks for the following errors:

   a. $K > 23$

   b. $N \leq K$

   c. $K \leq O$

   Upon detecting an error a pertinent message is given and FXEM is called for error tracing and return to the system.

3. Common usage:

   PLSQ uses 1309 cells of blank common.

Method:

PLSQ fits a set of observed data, $\left[x_i, y_i\right]$, with a polynomial of the form:

$$Y = C_1 X^k + C_2 X^{k-1} + \ldots + C_k X + C_{k+1}$$

by solving the normal equations for the C_i's. A transformation of the range of X to (-1,1) gives a partial decoupling of the normal equations which improves the accuracy of the solution. Double precision arithmetic is performed at critical points in the computation.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Deputy for Engineering Aeronautical Systems Division Wright-Patterson Air Force Base, Ohio	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

COMPUTER PROGRAM FOR REDUCING STATIC PROPELLER TEST DATA

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

5. AUTHOR(S) *(First name, middle initial, last name)*

Cafarelli, Gerald T.

Chopin, Matthew H.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1968	41	0

8a. CONTRACT OR GRANT NO. b. PROJECT NO. 478A c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) ASD-TR-68-19 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Deputy for Engineering Wright-Patterson Air Force Base, Ohio 45433

13. ABSTRACT

A computer program using a curve-fit technique was developed to reduce performance data obtained from static tests of aircraft propellers. The entire program is written in Fortran IV language for use on the IBM 7094 computer located at Wright-Patterson AFB, Ohio.

The program accepts static whirl rig test data (ie, raw RPM, horsepower, and thrust data) obtained at a fixed blade angle and reduces it into pertinent propeller relationships. The program first reduces the test data into various coefficients and computes the propeller tip Mach number. A curve fit technique then fits running curves through the test thrust and horsepower data points at the test tip Mach numbers. Intermediate horsepower and thrust values are determined from the fitted curves at selected Mach number increments, and all coefficients are recomputed. This results in a presentation of the reduced data in two forms, coeffcients computed from the actual test data and coefficients obtained at specific constant Mach number increments from the fitted curves. The data is presented in tabular printout form. This is a general program and is written so that the order of the curve fit, the Mach number increment, the number of test data points and the Mach number range can be varied.

The entire program deck and all nonstandard subroutines are included. Detail instructions are provided which should allow the program to be used by technicians or students who are familiar with Fortran IV language.

DD FORM 1 NOV 65 1473

UNCLASSIFIED
Security Classification

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computer Program						
Aircraft Propeller						
Static Tests						
Data Reduction						

UNCLASSIFIED

Security Classification