NASA TECHNICAL NOTE

NASA TN D-3667

c. 1

NASA TN D-3667

SOLID PROPELLANT ROCKET MOTOR PERFORMANCE COMPUTER PROGRAMS USING THE GROUP TRANSFORMATION METHOD

by B. J. Lee and P. B. Burchfield

Manned Spacecraft Center
Houston, Texas

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • OCTOBER 1966

NASA TN D-3667

SOLID PROPELLANT ROCKET MOTOR PERFORMANCE COMPUTER PROGRAMS USING THE GROUP TRANSFORMATION METHOD

By B. J. Lee and P. B. Burchfield

Manned Spacecraft Center
Houston, Texas

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – Price $3.00

ABSTRACT

This paper presents a method of evaluating the performance of a solid propellant rocket motor of fixed geometry and given propellant characteristics using the constant K_N (ratio of propellant burning-surface area to nozzle-throat area) process and group transformation method. This method does not require a prior knowledge of the K_N values, but requires only that the parameters to be evaluated be selected at constant regressed distances normal to the original propellant surface. Two computer programs utilizing this method are presented. The first performs the evaluation for general performance parameters over selected time intervals, and the second performs the evaluation for instantaneous performance versus time.

CONTENTS

TABLES

FIGURES

SOLID PROPELLANT ROCKET MOTOR PERFORMANCE COMPUTER PROGRAMS USING THE GROUP TRANSFORMATION METHOD

By B. J. Lee and P. B. Burchfield
Manned Spacecraft Center

SUMMARY

A solid propellant rocket motor of fixed geometry and given propellant will yield different performance with various prefire propellant temperatures. This paper presents two computer programs that transform the test data from as many as three different prefire propellant temperature groups to a like number of common prefire propellant temperatures of interest. Statistical calculations are made on the test data after they have been transformed. A detailed description of these computer programs is presented.

INTRODUCTION

The internal ballistic performance of a solid propellant rocket motor is a function of geometry, propellant temperature, and propellant composition. Therefore, a solid propellant rocket motor of fixed geometry (figs. 1(a) through 1(c)) and given propellant characteristics (fig. 2) will yield different performance with various propellant temperatures (fig. 3 and refs. 1 and 2). This phenomenon is due to the variation of propellant burning rate with propellant temperature (fig. 2) and will be referred to as the thermal sensitivity of the propellant.

By using adequate experimental test data, the variation of motor performance with propellant temperature can be mathematically defined. With proper utilization of this mathematical definition the performance test data (from motor test firings conducted at different propellant temperatures) can be transformed to any common temperature of interest, within the extremities of the test data.

There is generally a limited number of motors allotted to determine the performance variations (performance versus time and product variance) of a given motor design. The transformation of test data from different prefire propellant temperatures to given temperatures of interest is utilized in this report to obtain greater confidence in the statistical analyses performed on the test data.

This report describes two solid propellant rocket motor performance computer programs that were written at the Manned Spacecraft Center. These programs transform experimental test data from as many as three different prefire propellant temperature groups to a like number of propellant temperatures of interest. The computer program presented in appendix A performs the transformations and calculates general performance parameters over selected time intervals. The computer program presented in appendix B performs the transformation for instantaneous performance versus time.

Statistical analyses are performed on the test data after they have been transformed. The statistical analyses consist of calculating means, standard deviations, one-sided tolerance limits, and two-sided tolerance limits.

SYMBOLS

A area

a burning-rate equation coefficient, $r = aP^n$

b K_N equation coefficient, $K_N = bP^m$

e base of natural logarithm, 2.71828...

F longitudinal thrust

K tolerance factor

K_N ratio of propellant burning-surface area to nozzle-throat area

ln natural logarithm

m K_N equation exponent, $K_N = bP^m$

N sample size

n	burning-rate equation exponent, $r = aP^n$
P	chamber pressure; percentage of population
P_a	ambient pressure
r	burning rate
s	estimated standard deviation
T	prefire propellant temperature
t	motor operating time interval
u	tolerance limit
w	web thickness
X	experimentally determined values that make up a sample
γ	probability or confidence level
π_K	temperature sensitivity coefficient of chamber pressure
σ_K	temperature sensitivity coefficient of burning rate
ϕ	nozzle cant angle

Subscripts:

d	desired
e	exit
f	value of the parameter that corresponds to T_d as determined from a second-order least-squares curve fit of the parameter versus T
g	prefire propellant temperature group whose data are to be transformed

h value of the parameter that the explicit motor would be expected to experience if the prefire propellant temperature had been T_d

i value of the parameter experimentally acquired from the explicit motor whose data are to be transformed

j value of the parameter that corresponds to $\overline{T}_g$ as determined from a second-order least-squares curve fit of the parameter versus T

l lower

t throat

u upper

v vacuum

1 one-sided

2 two-sided

Operator:

— average (such as $\overline{F}$)

RELATED SOLID PROPELLANT ROCKET MOTOR THEORY

Ratio of Propellant Burning-Surface Area to Nozzle-Throat Area

Solid propellants burn in parallel layers and regress normal to the propellant surface (fig. 1(c)). Therefore, at any increasing distance normal to the original propellant surface, the exposed surface area can be predicted (fig. 4). Assuming that nozzle-throat erosion is reproducible, a geometric relationship can be determined between the ratio of propellant burning-surface area to nozzle-throat area K_N and the distance regressed normal to the original propellant surface (fig. 5).

The chamber pressure of a solid propellant rocket motor, with fixed geometry and given propellant characteristics at a known propellant temperature, is a function of K_N. The relationship between K_N, chamber pressure, and propellant temperature is generally determined experimentally and can, in some cases, be quite complex (fig. 2). However, the relationship between K_N and motor chamber pressure for limited chamber-pressure ranges can be approximated for a given propellant at a known propellant temperature by the following empirically determined relationship (fig. 6)

$$K_N = bP^m \tag{1}$$

where

$b = K_N$ equation coefficient.

$P =$ chamber pressure.

$m = K_N$ equation exponent.

Burning Rate

Burning rate r is the rate at which a solid propellant is consumed. It is measured in a direction normal to the propellant surface and expressed by in./sec. The burning rate of a specific propellant is a function of chamber pressure and propellant temperature. The complex relationship between burning rate, chamber pressure, and propellant temperature (fig. 2) can be approximated for a given propellant at a known propellant temperature for limited chamber pressure ranges by the following empirically determined relationship (fig. 6)

$$r = aP^n \tag{2}$$

where

$a =$ burning-rate equation coefficient.

$n =$ burning-rate equation exponent.

PROPELLANT THERMAL SENSITIVITY

Data obtained from test firings, conducted at different propellant temperatures, can be transformed to any common temperature of interest within the extremities of the available test data by any one of three processes. These processes are constant pressure, constant burn rate, and constant K_N. Compensations must be made for factors such as erosive burning and pressure losses along the length of the grain perforation during motor operation. The constant K_N process presented in this report is based on a constant geometric configuration at specifically regressed distances normal to the original propellant surface that compensates for the above mentioned factors. The actual regressed distance need not be known as long as the transformations are conducted at a constant regressed distance. This transformation method is also valid for the average geometric configuration during a time interval corresponding to a specific regressed distance (such as burn time, action time, and tail-off time).

Chamber-Pressure Transformation-Equation Derivation

The chamber-pressure transformation equation can be derived by first taking the natural logarithm of equation (1) giving

$$\ln K_N = \ln b + (m) \ln P \tag{3}$$

Taking the partial derivative of equation (3) with respect to the propellant temperature T at a constant K_N, and assuming m is independent of temperature, yields

$$\frac{\partial}{\partial T}\left(\ln K_N\right)_{K_N} \equiv 0 = \frac{\partial}{\partial T}(\ln b)_{K_N} + (m)\frac{\partial}{\partial T}(\ln P)_{K_N} \tag{4}$$

Therefore,

$$\frac{\partial}{\partial T}(\ln P)_{K_N} = -\frac{\frac{\partial}{\partial T}(\ln b)_{K_N}}{m} \tag{5}$$

The expression $\frac{\partial}{\partial T}(\ln P)_{K_N}$ is defined as the temperature sensitivity coefficient of chamber pressure (ref. 1) and is expressed as π_K.

Performing the indicated differentiation of equation (5) yields

$$\pi_K = \left[\frac{1}{P}\left(\frac{\partial P}{\partial T}\right)\right]_{K_N} \tag{6}$$

Integrating equation (6) yields

$$\int_{\overline{T}_g}^{T_d} \pi_K \, dT = \left[\int_{P_j}^{P_f} \frac{dP}{P}\right]_{K_N} \tag{7}$$

where

$\overline{T}_g$ = mean propellant temperature of the motors in the propellant temperature group whose data are to be transformed.

T_d = desired propellant temperature or the propellant temperature to which the experimental test data are to be transformed.

P_j = chamber pressure that corresponds to $\overline{T}_g$ as obtained from a second-order least-squares curve fit of the experimentally determined chamber pressures, at the specific regressed distance of interest normal to the original propellant surface, versus propellant temperature (fig. 7).

P_f = chamber pressure that corresponds to T_d as obtained from a second-order least-squares curve fit of the experimentally determined chamber pressures, at the specific regressed distance of interest normal to the original propellant surface, versus propellant temperature (fig. 7).

All least-squares curve fits are of the form $\ln x = a + by + cy^2$ where x is the dependent parameter, y is the independent parameter, and a, b, and c are constants.

The mean value theorem permits writing equation (7) as

$$\pi_K^* \int_{\overline{T}_g}^{T_d} dT = \left[\int_{P_j}^{P_f} \frac{dP}{P}\right]_{K_N} \tag{8}$$

where π_K^* is the average value of π_K over the path of constant K_N from $\overline{T}_g$ to T_d. This quantity π_K^* will hereafter be referred to as π_K. Performing the indicated integration of equation (8) yields

$$\pi_K\left(T_d - \overline{T}_g\right) = \left[\ln P_f - \ln P_j\right]_{K_N} \tag{9}$$

Therefore,

$$\pi_K = \left[\frac{1}{\left(T_d - \overline{T}_g\right)}\right]\left[\ln\frac{P_f}{P_j}\right]_{K_N} \tag{10}$$

By rearranging equation (10), the chamber pressure for each motor in the prefire propellant temperature group T_g (experimentally determined at the specific regressed distance for which π_K was calculated) is transformed to the chamber pressure corresponding to the desired prefire propellant temperature

$$P_h = \left[P_i \, e^{\pi_K\left(T_d - T_i\right)}\right]_{K_N} \tag{11}$$

where

P_h = expected chamber pressure (at the specific regressed distance of interest normal to the original propellant surface) of the explicit motor if the propellant temperature had been T_d.

P_i = chamber pressure experimentally determined at the specific regressed distance of interest normal to the original propellant surface of the explicit motor whose data are to be transformed.

T_i = actual propellant temperature of the explicit motor whose data are to be transformed.

Burning-Rate Transformation-Equation Derivation

Reference 1 presents a relationship between m and n, $m = 1 - n$, thus equations (1) and (2) may be combined and rearranged to give

$$P = \left(\frac{K_N}{b}\right)^{\left(\frac{1}{1-n}\right)} \tag{12}$$

Then, substituting equation (12) in equation (2) yields

$$r = a\left(\frac{K_N}{b}\right)^{\left(\frac{n}{1-n}\right)} \tag{13}$$

Taking the natural logarithm of equation (13) gives

$$\ln r = \ln a - \left(\frac{n}{1 - n}\right) \ln b + \left(\frac{n}{1 - n}\right) \ln K_N \tag{14}$$

Partial differentiation of equation (14) with respect to the propellant temperature T at a constant K_N, and assuming n is independent of temperature, yields

$$\frac{\partial}{\partial T}(\ln r)_{K_N} = \frac{\partial}{\partial T}(\ln a)_{K_N} - \frac{n}{1 - n}\frac{\partial}{\partial T}(\ln b)_{K_N} + \frac{n}{1 - n}\frac{\partial}{\partial T}\left(\ln K_N\right)_{K_N} \tag{15}$$

The expression $\frac{\partial}{\partial T}(\ln r)_{K_N}$ is defined as the temperature sensitivity coefficient of burning rate (ref. 1) and is expressed as σ_K.

Performing the indicated differentiation of equation (15) yields

$$\sigma_K = \left[\left(\frac{1}{r}\right)\frac{\partial r}{\partial T}\right]_{K_N} \tag{16}$$

Integrating equation (16) yields

$$\int_{\overline{T}_g}^{T_d} \sigma_K \, dT = \left[\int_{r_j}^{r_f} \frac{dr}{r}\right]_{K_N} \tag{17}$$

where

r_j = burning rate that corresponds to $\overline{T}_g$ as obtained from a second-order least-squares curve fit of the experimentally determined burning rates, at the specific regressed distance of interest normal to the original propellant surface, versus propellant temperature (fig. 8).

r_f = burning rate corresponding to T_d as obtained from a second-order least-squares curve fit of the experimentally determined burning rates, at the specific regressed distance of interest normal to the original propellant surface, versus propellant temperature (fig. 8).

The mean value theorem permits writing equation (17) as

$$\sigma_K^* \int_{\overline{T}_g}^{T_d} dt = \left[\int_{r_j}^{r_f} \frac{dr}{r}\right]_{K_N} \tag{18}$$

where σ_K^* is the average value of σ_K over the path of constant K_N from $\bar{T}_g$ to T_d. This quantity σ_K^* will hereafter be referred to as σ_K. Performing the indicated integration of equation (18) yields

$$\sigma_K \left[T_d - \bar{T}_g \right] = \left[\ln r_f - \ln r_j \right]_{K_N} \qquad (19)$$

Therefore,

$$\sigma_K = \left[\frac{1}{\left(T_d - \bar{T}_g \right)} \right] \left[\ln \frac{r_f}{r_j} \right]_{K_N} \qquad (20)$$

By rearranging equation (20), the burning rate for each motor in the propellant temperature group T_g (experimentally determined at the specific regressed distance for which σ_K was calculated) is transformed to the burning rate corresponding to the desired propellant temperature

$$r_h = \left[r_i \, e^{\sigma_K \left(T_d - T_i \right)} \right]_{K_N} \qquad (21)$$

where

r_h = expected burning rate (at the specific regressed distance of interest normal to the original propellant surface) of the explicit motor if the propellant temperature had been T_d.

r_i = burning rate experimentally determined at the specific regressed distance of interest normal to the original propellant surface of the explicit motor whose data are to be transformed.

Time Transformation Equation Derivation

Since burning rate is the rate at which a solid propellant is consumed,

$$\bar{r} = \frac{w}{t} \tag{22}$$

where

$\bar{r}$ = average burning rate during time interval t.

w = propellant thickness consumed (measured normal to the original propellant surface) during time interval t.

Assuming that the thermal expansion of the propellant is negligible and substituting equation (22) in equation (20)

$$\sigma_K = \left[\frac{1}{T_d - \bar{T}_g}\right]\left[\ln \frac{t_j}{t_f}\right]_{K_N} \tag{23}$$

where

t_j = estimated motor operating time interval at temperature $\bar{T}_g$ (fig. 9).

t_f = estimated motor operating time interval at temperature T_d (fig. 9).

By rearranging equation (23), the measured operating time interval for each motor in the propellant temperature group T_g is transformed to the operating time interval corresponding to the desired propellant temperature

$$t_h = \left[\frac{t_i}{e^{\sigma_K (T_d - T_i)}} \right]_{K_N} \tag{24}$$

where

t_i = measured operating time interval of interest of the explicit motor whose data are to be transformed.

t_h = estimated operating time interval of each motor at propellant temperature T_d.

Thrust Related Calculations and Transformation Equation

In order to transform longitudinal thrust in the simplest manner and also to provide the capability of handling motors that utilize as many as four nozzles (with equal or unequal expansion ratios) and with a nozzle cant angle (such as the Apollo launch escape motor), the thrust measurements are first corrected to vacuum pressure altitude

$$\overline{F}_{vi} = \overline{F}_i + P_a A_e \cos \phi \tag{25}$$

where

$\overline{F}_{vi}$ = measured average longitudinal thrust of each motor during the time interval of interest corrected to vacuum pressure altitude.

$\overline{F}_i$ = measured average longitudinal thrust of each motor during the time interval of interest.

P_a = ambient pressure experienced by each motor during the time interval of interest.

A_e = total nozzle-exit area.

ϕ = nozzle cant angle from the longitudinal centerline of the motor.

Equation (25) can also be used to correct thrust data to any pressure altitude of interest; however, it should be noted that there has been no attempt to evaluate the performance data during the time interval of nozzle flow separation.

The average longitudinal thrust for each motor in the propellant temperature group T_g (experimentally determined at the specific regressed distance for which π_K was calculated) is transformed to the average longitudinal thrust corresponding to the desired propellant temperature

$$\overline{F}_{vh} = \overline{F}_{vi}\left(\frac{P_h}{P_i}\right) \tag{26}$$

where

$\overline{F}_{vh}$ = average expected thrust of the explicit motor if the propellant temperature had been T_d.

STATISTICAL ANALYSIS

When parameters are experimentally determined, it is desirable for design, performance, and reliability evaluations to establish limits or bounds which contain a desired percentage of a specific parameters population, with a confidence or probability that the intended condition is satisfied. The bounds thus established are called tolerance limits.

To determine these tolerance limits, a frequency distribution must be assumed. This paper assumes normal distribution since experience has shown that the majority of experimental data is approximately normally distributed. Although a check for normality may be performed, moderate departure from the assumed distribution will not seriously affect the tolerance limits computed.

Given a sample of data, the mean is estimated by

$$\overline{X} = \frac{\Sigma X}{N} \qquad (27)$$

and the standard deviation is estimated by

$$s = \sqrt{\frac{N\Sigma X^2 - (\Sigma X)^2}{N(N - 1)}} \qquad (28)$$

where

N = the number of points in the sample.

X = experimentally determined values that make up the sample.

Now that estimates of the mean and standard deviation are known, tolerance limits may be determined. The tolerance limits are the bounds which with probability γ contain at least P percent of the population.

The tolerance limits are of the form

$$u = \overline{X} \pm Ks \qquad (29)$$

where

K = tolerance factor.

Note:

K_1 = tolerance factor for one-sided tolerance limits.

K_2 = tolerance factor for two-sided tolerance limits.

This equation may be applied in two different ways. It can be used to determine the two-sided tolerance limits (fig. 10), or for determining either the upper or the lower one-sided tolerance limit (fig. 11).

Two-Sided Tolerance Limits

The two-sided tolerance limits are the upper and the lower bounds that will enclose at least the desired percentage of the expected population with a preselected confidence. The tolerance factors required in equation (29) are given in table I with confidence γ for P percent of the expected population and N data points.

Example: A sample of 25 points with $\overline{X} = 10.02$ and $s = 0.13$.

Two-sided tolerance limits for 90 percent of the population with 95 percent confidence are desired. From table I, for $P = 0.90$ and $\gamma = 0.95$, $K = 2.208$.

$$u = \overline{X} \pm K_2 s = 10.02 \pm (2.208)(0.13) = 10.02 \pm 0.29$$

Therefore, with 95 percent confidence, at least 90 percent of the population lie between 9.73 and 10.31.

One-Sided Tolerance Limits

The one-sided tolerance limit is the upper or the lower bound below which or above which at least the desired percentage of the population can be expected to lie with a preselected confidence. The tolerance factors required in equation (29) are given in table II with confidence γ for P (where $P = 1 - \alpha$) percentage of the expected population and N data points.

Example: Find the one-sided tolerance limit for the preceding example. From table II, K = 1.838

$$u = \overline{X} \pm K_1 s = 10.02 \pm (1.838)(0.13) = 10.02 \pm 0.24$$

Therefore, with 95 percent confidence, at least 90 percent of the population lies either above 9.78 or below 10.26.

The computer program presented in appendix A calculates estimated standard deviation, one-sided tolerance limits, and two-sided tolerance limits for both physical data and transformed general performance parameters over selected time intervals. The computer program presented in appendix B calculates estimated standard deviation and one-sided tolerance limits or two-sided tolerance limits for transformed instantaneous performance data versus time.

GENERAL SOLID PROPELLANT ROCKET MOTOR PERFORMANCE COMPUTER PROGRAM USING THE GROUP TRANSFORMATION METHOD

General Description

This computer program was written at the Manned Spacecraft Center in Fortran IV for the IBM 7094 computer with 32K storage. The program transforms general performance parameters for selected time intervals acquired from as many as three different propellant temperature groups to a like number of propellant temperatures of interest for a solid propellant rocket motor of fixed geometry and a given propellant. The transformations are performed for the average geometric configuration during the time intervals that correspond to the specific regressed distances of interest (such as burn time, action time, and tail-off time).

The actual regressed distances need not be known as long as the transformations are conducted at the average geometric configuration for the specific regressed distances. Statistical calculations are performed on both physical data and general performance parameters after the parameters have been transformed to the specific common temperatures of interest. The input format for the program is presented in table III.

Computer Deck Setup

This program exceeds 32K core storage, and the overlay feature of Fortran IV is employed to allow the program to be run as a single input job.

Six decks constitute the program in its entirety and include a calling program and five subroutines.

The actual deck setup for the IBM 7094 is illustrated in figure 12.

1. Setup Cards

   $JOB card
   $IBJOB card

2. Program

   DRIVER
   STATS
   $ORIGIN ALPHA card
   MOTORS
   GLS1
   $ORIGIN ALPHA card
   MOT2
   MOT3

3. Data

   $DATA
   DATA deck
   7/8 card

DRIVER. - DRIVER is the executive routine required in an overlayed program. This routine never leaves the machine storage area and calls the various overlayed subroutines.

Subroutine STATS. - Subroutine STATS computes the mean and the standard deviation of an input array of given size. Due to the computer arithmetic methods, standard deviations were sometimes found to exist with a small nonzero value when all elements of the input array were identical. To avoid this difficulty, standard deviation is set to zero if its ratio to the mean is less than 0.00025. Since STATS is used by all overlayed sections of the program, it has been placed in the permanent storage area with DRIVER.

Subroutine MOTORS. - Subroutine MOTORS reads in all the data. It sorts the data for each motor into the proper propellant temperature group and does performance calculations on the input data. It calculates the transformation factors for pressures and times $(\pi_K$ and $\sigma_K)$ and performs the transformations on these data.

Subroutine GLS1. - Subroutine GLS1 is used to calculate a second-order least-squares curve fit of the natural logarithms of the chamber pressures and the operation time intervals versus propellant temperature.

Subroutine MOT2. - Subroutine MOT2 calculates transformed thrusts, impulses and specific impulses, and the tolerance limits at each desired pressure altitude.

Subroutine MOT3. - Subroutine MOT3 calculates transformed burning rate, characteristic velocity, and their tolerance limits.

Program Restrictions

This program was written to perform the desired transformations for a maximum of 15 motors per experimental prefire propellant temperature group, 3 experimental prefire propellant temperature groups, and 9 time intervals such as ignition delay time, thrust rise time, burn time, total time, and so forth.

The input restrictions are that the number of propellant temperatures (to which the data are to be transformed) must equal the number of experimental temperature groups, the first ambient pressure (EXPA (1)) must equal zero psia (vacuum), and the desired temperatures must be in ascending order.

Care must be exercised in the evaluation of the program results. For example, outputs of specific impulse, total impulse, and chamber pressure integral are meaningless if instantaneous data are inputed. The evaluation of data acquired during nozzle flow separation is beyond the scope of this program; therefore, action time, tail-off time, and total time data must be carefully handled. It should be noted that the output values for characteristic velocity (CSTAR) are valid for total time only.

Output Formats

The output formats are basically of two types: transformed data and nontransformed data. Table IV presents a typical page of nontransformed data output. The type of output (time, pressure, thrust, or other parameters) is identified in the upper left corner of the page. The nine columns represent the motor operating time intervals of interest (such as ignition delay time, thrust rise time, burn time, total time, and so forth). Nine columns will always be printed out even though fewer time intervals are inputed. The non-used columns will be printed out as zeros. At the left of the page, each motor is identified by its alphanumeric identification code, and the motors are separated into propellant temperature groups (up to three). Means and standard deviations are printed out for the motors in each temperature group. In three instances (thrust, impulse, and specific impulse) identical formats are used for different values of pressure altitude. The pressure altitude is printed out at the top of the page.

Table V presents a typical output page of transformed data. Each output page represents one time interval such as ignition delay time, thrust rise time, burn time, total time, and so forth. The time interval is printed out in numerical form (time no. 1, time no. 2, time no. 3, and up to time no. 9) in the upper left corner of the page and must be correlated with the time definitions in the input data (time no. 1 = ignition delay time, time no. 2 = thrust rise time, time no. 3 = burn time, or up to time no. 9 = total time). The type of output (time, pressure, thrust, and other outputs) is printed out at the top of the page. The three columns represent the three propellant temperatures to which the input data have been transformed. At the left of the page each motor is identified by its alphanumeric identification code, and the motors are separated into the original propellant temperature groups (up to three). Means and standard deviations are printed out for motors in each temperature group. Means, standard deviations, and tolerance limits are printed out for the three groups as a whole. In three instances (thrust, impulse, and specific impulse) identical formats are used for different values of pressure altitude. The pressure altitude is printed out under the type of output.

SOLID PROPELLANT ROCKET MOTOR PERFORMANCE VERSUS TIME COMPUTER PROGRAM USING THE GROUP TRANSFORMATION METHOD

General Description

This computer program was written at the Manned Spacecraft Center in Fortran IV for the Univac 1107 computer with 65K storage. The program effectively transforms experimentally obtained thrust-and-chamber pressure versus time data (acquired for as many as three different propellant temperature groups) to a common propellant temperature of interest for a solid propellant rocket motor of fixed geometry and a given propellant. The transformations are performed at specific regressed distances normal to the original propellant surface. The actual regressed distance need not be known, as long as the transformations are conducted at a constant regressed distance (such as percents of burn time and percents of tail-off time). See figure 13.

Statistical calculations are performed on the test data after they have been transformed to the specific common temperature of interest. The input format for the program is presented in table VI.

Computer Deck Setup

Twelve decks constitute the program in its entirety, including a main program and 12 subroutines. The program for the Univac 1107 is illustrated in figure 14.

1. Setup Card

   $JOB card

2. Program

   MAIN
   SRCH1
   LSTSQ
   CURVE
   STATS
   GLS1
   ACCEND
   NORMLZ
   FIXIT

STDEV
CF2F1
CF2F2
QUIKMV

3. Data

XQT card
DATA deck
EOF card
FIN card

MAIN program. - The MAIN program reads in all data and sorts the data for each motor into its proper prefire propellant temperature group. It then calls the CURVE FIT routines to curve fit both input pressure and input thrust versus input time. It subdivides burn time and tail-off time into the required number of subintervals (fig. 13). Finally, it calls subroutine LSTSQ to compute the transformed data.

Subroutine SRCH1. - Subroutine SRCH1 computes time increments in terms of percent web time and percent tail-off time (fig. 13). The subroutine then selects from the CURVE FIT routines the proper cubic coefficients for the particular time increment being considered. After the proper coefficients have been selected, the subroutine uses them to compute the values of the ordinate (pressure or thrust) at each time increment and returns these values to the main program.

Subroutine LSTSQ. - Subroutine LSTSQ computes the transformation factors for pressures and times (π_K and σ_K) and performs the transformations on these and thrust data at each percent web time and at each percent tail-off time and prints them out as final answers. Graphs of the final answers are also produced (see subroutine QUIKMV).

Subroutine STATS. - Subroutine STATS computes the mean and standard deviation of an input array of given size. Due to computer arithmetic methods, standard deviations were sometimes found to exist with a small nonzero value when all elements of the input array were identical. To avoid this difficulty, standard deviation is set to zero if its ratio to the mean is less than 0.00025.

Subroutine GLS1. - Subroutine GLS1 is used to calculate a second-order least-squares curve fit of the natural logarithms of chamber pressures, and motor operation times (at each experimentally determined percent web time and percent tail-off time) versus propellant temperature.

CURVE FIT routines. - The CURVE FIT routines consist of subroutines CURVE, ACCEND, NORMLZ, FIXIT, STDEV, CF2F1, and CF2F2. These subroutines fit an input ordinate array versus an input abscissa array using a piecewise cubic least-squares curve fit. Because of the piecewise nature of the fit there are NPTS-ICON +1 sets of cubic coefficients, where NPTS is the size of the input arrays, and ICON points are fit in each piece. Each coefficient set is valid over a limited interval. The coefficients may be printed out by setting input parameter IPRNT to 1.

Subroutine QUIKMV. - Subroutine QUIKMV is a MSC general plot routine for the SC-4020. It has not been included in appendix B since most facilities use plotting routines individually suited to the available plotting equipment. The call to this routine may be deleted, or a dummy routine substituted, without affecting the overall program.

Program Restrictions

This program was written to perform the desired transformation for as many as 15 motors per experimental propellant temperature group, 3 experimental propellant temperature groups, and 200 selected inputs of time with corresponding chamber pressure and thrust for each motor.

Since the program is limited to 200 inputs of time with corresponding chamber pressure and thrust for each motor, care must be exercised in choosing the input points in order that they best define the variation of chamber pressure and thrust versus time for each motor (fig. 15). Care must also be exercised in choosing the proper output points that best define the variations of chamber pressure and thrust versus time. This can be accomplished since burn time can be subdivided into as many as five subintervals, and tail-off time can be subdivided into as many as four subintervals. As many time increments (percents of web time and/or percents of tail-off time) as desired can be generated to define the variation of chamber pressure and thrust versus time for each subinterval as long as the total does not exceed 200 points (fig. 13).

Output Formats

The outputs consist of the transformed results in computer printout form, such as presented in tables VII and VIII, and SC-4020 graphs as shown in figures 16(a) and 16(b).

The transformed results present the statistics (mean and tolerance limits) of the transformed time, chamber pressure, and thrust at the desired percent of burn time and percent tail-off time. Transformed thrust statistics can be presented for as many as three desired pressure altitudes.

CONCLUDING REMARKS

The method and computer programs presented in this report were developed at the Manned Spacecraft Center specifically to evaluate the performance of the solid rockets used in the Apollo Launch Escape System. From the conception, however, the method of solution, including the machine programs, was intended to be generally applicable for any rocket motor utilizing a single propellant grain with no throttling capability.

The programs were designed for two types of evaluations. Often, in the requirements, motors are constrained to operate within a given performance regime during some portion of the firing, while delivering a specified nominal performance over the total operation. The program presented in appendix A was intended to evaluate the more stringent constraints placed upon the launch escape motor where both the maximum thrust level and the minimum impulse were specified for an initial phase of firing, and an overall minimum performance required for the abort mission.

The program presented in appendix B compares the transformed thrust-time and pressure-time relationships of all motors tested to define the expected limits of the performance at any temperature. This program has been used to describe the general thrust and pressure characteristics for documentation and product variance purposes.

It is anticipated that these programs will be valuable, particularly for contracting agencies and prime contractors, in the evaluation of solid rocket test data.

Manned Spacecraft Center
National Aeronautics and Space Administration
Houston, Texas, June 29, 1966

TABLE I. - TWO-SIDED TOLERANCE FACTORS FOR NORMAL DISTRIBUTIONS[a]

[Factors K such that the probability is γ that at least a proportion P of the distribution will be included between $\bar{X} \pm Ks$, where $\bar{X}$ and s are estimates of the mean and the standard deviation computed from a sample of N]

	γ = 0.75					γ = 0.90					γ = 0.95					γ = 0.99				
N \ P	0.75	0.90	0.95	0.99	0.999	0.75	0.90	0.95	0.99	0.999	0.75	0.90	0.95	0.99	0.999	0.75	0.90	0.95	0.99	0.999
2	4.498	6.301	7.414	9.531	11.920	11.407	15.978	18.800	24.167	30.227	22.858	32.019	37.674	48.430	60.573	114.363	160.193	188.491	242.300	303.054
3	2.501	3.538	4.187	5.431	6.844	4.132	5.847	6.919	8.974	11.309	5.922	8.380	9.916	12.861	16.208	13.378	18.930	22.401	29.055	36.616
4	2.035	2.892	3.431	4.471	5.657	2.932	4.166	4.943	6.440	8.149	3.779	5.369	6.370	8.299	10.502	6.614	9.398	11.150	14.527	18.383
5	1.825	2.590	3.088	4.033	5.117	2.454	3.494	4.152	5.423	6.879	3.002	4.275	5.079	6.634	8.415	4.643	6.612	7.855	10.260	13.015
6	1.704	2.429	2.889	3.779	4.802	2.196	3.131	3.723	4.870	6.188	2.604	3.712	4.414	5.775	7.337	3.743	5.337	6.345	8.301	10.548
7	1.624	2.318	2.757	3.611	4.593	2.034	2.902	3.452	4.521	5.750	2.361	3.369	4.007	5.248	6.676	3.233	4.613	5.488	7.187	9.142
8	1.568	2.238	2.663	3.491	4.444	1.921	2.743	3.264	4.278	5.446	2.197	3.136	3.732	4.891	6.226	2.905	4.147	4.936	6.468	8.234
9	1.525	2.178	2.593	3.400	4.330	1.839	2.626	3.125	4.098	5.220	2.078	2.967	3.532	4.631	5.899	2.677	3.822	4.550	5.966	7.600
10	1.492	2.131	2.537	3.328	4.241	1.775	2.535	3.018	3.959	5.046	1.987	2.839	3.379	4.433	5.649	2.508	3.582	4.265	5.594	7.129
11	1.465	2.093	2.493	3.271	4.169	1.724	2.463	2.933	3.849	4.906	1.916	2.737	3.259	4.277	5.452	2.378	3.397	4.045	5.308	6.766
12	1.443	2.062	2.456	3.223	4.110	1.683	2.404	2.863	3.758	4.792	1.858	2.655	3.162	4.150	5.291	2.274	3.250	3.870	5.079	6.477
13	1.425	2.036	2.424	3.183	4.059	1.648	2.355	2.805	3.682	4.697	1.810	2.587	3.081	4.044	5.158	2.190	3.130	3.727	4.893	6.240
14	1.409	2.013	2.398	3.148	4.016	1.619	2.314	2.756	3.618	4.615	1.770	2.529	3.012	3.955	5.045	2.120	3.029	3.608	4.737	6.043
15	1.395	1.994	2.375	3.118	3.979	1.594	2.278	2.713	3.562	4.545	1.735	2.480	2.954	3.878	4.949	2.060	2.945	3.507	4.605	5.876
16	1.383	1.977	2.355	3.092	3.946	1.572	2.246	2.676	3.514	4.484	1.705	2.437	2.903	3.812	4.865	2.009	2.872	3.421	4.492	5.732
17	1.372	1.962	2.337	3.069	3.917	1.552	2.219	2.643	3.471	4.430	1.679	2.400	2.858	3.754	4.791	1.965	2.808	3.345	4.393	5.607
18	1.363	1.948	2.321	3.048	3.891	1.535	2.194	2.614	3.433	4.382	1.655	2.366	2.819	3.702	4.725	1.926	2.753	3.279	4.307	5.497
19	1.355	1.936	2.307	3.030	3.867	1.520	2.172	2.588	3.399	4.339	1.635	2.337	2.784	3.656	4.667	1.891	2.703	3.221	4.230	5.399
20	1.347	1.925	2.294	3.013	3.846	1.506	2.152	2.564	3.368	4.300	1.616	2.310	2.752	3.615	4.614	1.860	2.659	3.168	4.161	5.312
21	1.340	1.915	2.282	2.998	3.827	1.493	2.135	2.543	3.340	4.264	1.599	2.286	2.723	3.577	4.567	1.833	2.620	3.121	4.100	5.234
22	1.334	1.906	2.271	2.984	3.809	1.482	2.118	2.524	3.315	4.232	1.584	2.264	2.697	3.543	4.523	1.808	2.584	3.078	4.044	5.163
23	1.328	1.898	2.261	2.971	3.793	1.471	2.103	2.506	3.292	4.203	1.570	2.244	2.673	3.512	4.484	1.785	2.551	3.040	3.993	5.098
24	1.322	1.891	2.252	2.959	3.778	1.462	2.089	2.489	3.270	4.176	1.557	2.225	2.651	3.483	4.447	1.764	2.522	3.004	3.947	5.039
25	1.317	1.883	2.244	2.948	3.764	1.453	2.077	2.474	3.251	4.151	1.545	2.208	2.631	3.457	4.413	1.745	2.494	2.972	3.904	4.985
26	1.313	1.877	2.236	2.938	3.751	1.444	2.065	2.460	3.232	4.127	1.534	2.193	2.612	3.432	4.382	1.727	2.469	2.941	3.865	4.935
27	1.309	1.871	2.229	2.929	3.740	1.437	2.054	2.447	3.215	4.106	1.523	2.178	2.595	3.409	4.353	1.711	2.446	2.914	3.828	4.888

[a]From Techniques of Statistical Analysis by Eisenhart, Hastay, and Wallis. Copyright 1947. McGraw-Hill Book Company. Used by permission.

TABLE I. - TWO-SIDED TOLERANCE FACTORS

FOR NORMAL DISTRIBUTIONS[a] - Concluded

[Factors K such that the probability is γ that at least a proportion P of the distribution will be included between $\bar{X} \pm Ks$, where $\bar{X}$ and s are estimates of the mean and the standard deviation computed from a sample of N]

	$\gamma = 0.75$					$\gamma = 0.90$					$\gamma = 0.95$					$\gamma = 0.99$				
N \ P	0.75	0.90	0.95	0.99	0.999	0.75	0.90	0.95	0.99	0.999	0.75	0.90	0.95	0.99	0.999	0.75	0.90	0.95	0.99	0.999
28	1.305	1.865	2.222	2.920	3.728	1.430	2.044	2.435	3.199	4.085	1.514	2.164	2.579	3.388	4.326	1.695	2.424	2.888	3.794	4.845
29	1.301	1.860	2.216	2.911	3.718	1.423	2.034	2.424	3.184	4.066	1.505	2.152	2.564	3.368	4.301	1.681	2.404	2.864	3.763	4.805
30	1.297	1.855	2.210	2.904	3.708	1.417	2.025	2.413	3.170	4.049	1.497	2.140	2.549	3.350	4.278	1.668	2.385	2.841	3.733	4.768
31	1.294	1.850	2.204	2.896	3.699	1.411	2.017	2.403	3.157	4.032	1.489	2.129	2.536	3.332	4.256	1.656	2.367	2.820	3.706	4.732
32	1.291	1.846	2.199	2.890	3.690	1.405	2.009	2.393	3.145	4.016	1.481	2.118	2.524	3.316	4.235	1.644	2.351	2.801	3.680	4.699
33	1.288	1.842	2.194	2.883	3.682	1.400	2.001	2.385	3.133	4.001	1.475	2.108	2.512	3.300	4.215	1.633	2.335	2.782	3.655	4.668
34	1.285	1.838	2.189	2.877	3.674	1.395	1.994	2.376	3.122	3.987	1.468	2.099	2.501	3.286	4.197	1.623	2.320	2.764	3.632	4.639
35	1.283	1.834	2.185	2.871	3.667	1.390	1.988	2.368	3.112	3.974	1.462	2.090	2.490	3.272	4.179	1.613	2.306	2.748	3.611	4.611
36	1.280	1.830	2.181	2.866	3.660	1.386	1.981	2.361	3.102	3.961	1.455	2.081	2.479	3.258	4.161	1.604	2.203	2.732	3.590	4.585
37	1.278	1.827	2.177	2.860	3.653	1.381	1.975	2.353	3.092	3.949	1.450	2.073	2.470	3.246	4.146	1.595	2.281	2.717	3.571	4.560
38	1.275	1.824	2.173	2.855	3.647	1.377	1.969	2.346	3.083	3.938	1.446	2.068	2.464	3.237	4.134	1.587	2.269	2.703	3.552	4.537
39	1.273	1.821	2.169	2.850	3.641	1.374	1.964	2.340	3.075	3.927	1.441	2.060	2.455	3.226	4.120	1.579	2.257	2.690	3.534	4.514
40	1.271	1.818	2.166	2.846	3.635	1.370	1.959	2.334	3.066	3.917	1.435	2.052	2.445	3.213	4.104	1.571	2.247	2.677	3.518	4.493
41	1.269	1.815	2.162	2.841	3.629	1.366	1.954	2.328	3.059	3.907	1.430	2.045	2.437	3.202	4.090	1.564	2.236	2.665	3.502	4.472
42	1.267	1.812	2.159	2.837	3.624	1.363	1.949	2.322	3.051	3.897	1.426	2.039	2.429	3.192	4.077	1.557	2.227	2.653	3.486	4.453
43	1.266	1.810	2.156	2.833	3.619	1.360	1.944	2.316	3.044	3.888	1.422	2.033	2.422	3.183	4.065	1.551	2.217	2.642	3.472	4.434
44	1.264	1.807	2.153	2.829	3.614	1.357	1.940	2.311	3.037	3.879	1.418	2.027	2.415	3.173	4.053	1.545	2.208	2.631	3.458	4.416
45	1.262	1.805	2.150	2.826	3.609	1.354	1.935	2.306	3.030	3.871	1.414	2.021	2.408	3.165	4.042	1.539	2.200	2.621	3.444	4.399
46	1.261	1.802	2.148	2.822	3.605	1.351	1.931	2.301	3.024	3.863	1.410	2.016	2.402	3.156	4.031	1.533	2.192	2.611	3.431	4.383
47	1.259	1.800	2.145	2.819	3.600	1.348	1.927	2.297	3.018	3.855	1.406	2.011	2.396	3.148	4.021	1.527	2.184	2.602	3.419	4.367
48	1.258	1.798	2.143	2.815	3.596	1.345	1.924	2.292	3.012	3.847	1.403	2.006	2.390	3.140	4.011	1.522	2.176	2.593	3.407	4.352
49	1.256	1.796	2.140	2.812	3.592	1.343	1.920	2.288	3.006	3.840	1.399	2.001	2.384	3.133	4.002	1.517	2.169	2.584	3.396	4.337
50	1.255	1.794	2.138	2.809	3.588	1.340	1.916	2.284	3.001	3.833	1.396	1.996	2.379	3.126	3.993	1.512	2.162	2.576	3.385	4.323
51	1.253	1.792	2.135	2.806	3.584	1.338	1.913	2.279	2.995	3.826	1.393	1.992	2.373	3.119	3.984	1.507	2.155	2.568	3.374	4.310
52	1.252	1.790	2.133	2.803	3.581	1.336	1.910	2.276	2.990	3.820	1.390	1.988	2.368	3.112	3.975	1.503	2.148	2.560	3.364	4.297
53	1.251	1.789	2.131	2.801	3.577	1.334	1.907	2.272	2.985	3.813	1.387	1.984	2.363	3.106	3.967	1.498	2.142	2.552	3.354	4.284
54	1.250	1.787	2.129	2.798	3.574	1.331	1.904	2.268	2.981	3.807	1.384	1.980	2.359	3.100	3.959	1.494	2.136	2.545	3.344	4.272
55	1.249	1.785	2.127	2.795	3.571	1.329	1.901	2.265	2.976	3.801	1.382	1.976	2.354	3.094	3.951	1.490	2.130	2.538	3.335	4.260
56	1.247	1.784	2.125	2.793	3.567	1.327	1.898	2.261	2.972	3.796	1.379	1.972	2.350	3.088	3.944	1.486	2.124	2.531	3.326	4.249
57	1.246	1.782	2.123	2.790	3.564	1.325	1.895	2.258	2.967	3.790	1.377	1.968	2.345	3.082	3.937	1.482	2.119	2.524	3.318	4.238
58	1.245	1.781	2.122	2.788	3.561	1.323	1.892	2.255	2.963	3.785	1.374	1.965	2.341	3.076	3.930	1.478	2.113	2.518	3.309	4.227
59	1.244	1.779	2.120	2.786	3.558	1.322	1.890	2.252	2.959	3.779	1.372	1.961	2.337	3.071	3.923	1.474	2.108	2.512	3.301	4.216
60	1.243	1.778	2.118	2.784	3.556	1.320	1.887	2.248	2.955	3.774	1.369	1.958	2.333	3.066	3.916	1.471	2.103	2.506	3.293	4.206
61	1.242	1.776	2.117	2.781	3.553	1.318	1.885	2.245	2.951	3.769	1.367	1.955	2.329	3.061	3.909	1.467	2.098	2.500	3.285	4.196
62	1.241	1.775	2.115	2.779	3.550	1.316	1.882	2.243	2.947	3.765	1.365	1.951	2.325	3.056	3.903	1.464	2.093	2.494	3.278	4.187
63	1.240	1.774	2.113	2.777	3.548	1.315	1.880	2.240	2.944	3.760	1.363	1.948	2.322	3.051	3.897	1.461	2.089	2.489	3.271	4.178
64	1.240	1.772	2.112	2.775	3.545	1.313	1.878	2.237	2.940	3.755	1.361	1.945	2.318	3.046	3.891	1.458	2.084	2.483	3.264	4.169

[a]From Techniques of Statistical Analysis by Eisenhart, Hastay, and Wallis.

TABLE II. - ONE-SIDED TOLERANCE FACTORS FOR NORMAL DISTRIBUTIONS[a]

[Factors K such that the probability is γ that at least a proportion $1 - \alpha$ of the distribution will be less than $\overline{X} \pm Ks$ (or greater than $\overline{X}$ - Ks), where $\overline{X}$ and s are estimates of the mean and standard deviation computed from a sample of size N]

α	γ = 0.75					γ = 0.90					γ = 0.95					γ = 0.99				
N	0.25	0.10	0.05	0.01	0.001	0.25	0.10	0.05	0.01	0.001	0.25	0.10	0.05	0.01	0.001	0.25	0.10	0.05	0.01	0.001
3	1.464	2.501	3.152	4.396	5.805	2.602	4.258	5.310	7.340	9.651	3.804	6.158	7.655	10.552	13.857					
4	1.256	2.134	2.680	3.726	4.910	1.972	3.187	3.957	5.437	7.128	2.619	4.163	5.145	7.042	9.215					
5	1.152	1.961	2.463	3.421	4.507	1.698	2.742	3.400	4.666	6.112	2.149	3.407	4.202	5.741	7.501					
6	1.087	1.860	2.336	3.243	4.273	1.540	2.494	3.091	4.242	5.556	1.895	3.006	3.707	5.062	6.612	2.849	4.408	5.409	7.334	9.540
7	1.043	1.791	2.250	3.126	4.118	1.435	2.333	2.894	3.972	5.201	1.732	2.755	3.399	4.641	6.061	2.490	3.856	4.730	6.411	8.348
8	1.010	1.740	2.190	3.042	4.008	1.360	2.219	2.755	3.783	4.955	1.617	2.582	3.188	4.353	5.686	2.252	3.496	4.287	5.811	7.566
9	0.984	1.702	2.141	2.977	3.924	1.302	2.133	2.649	3.641	4.772	1.532	2.454	3.031	4.143	5.414	2.085	3.242	3.971	5.389	7.014
10	0.964	1.671	2.103	2.927	3.858	1.257	2.065	2.568	3.532	4.629	1.465	2.355	2.911	3.981	5.203	1.954	3.048	3.739	5.075	6.603
11	0.947	1.646	2.073	2.885	3.804	1.219	2.012	2.503	3.444	4.515	1.411	2.275	2.815	3.852	5.036	1.854	2.897	3.557	4.828	6.284
12	0.933	1.624	2.048	2.851	3.760	1.188	1.966	2.448	3.371	4.420	1.366	2.210	2.736	3.747	4.900	1.771	2.773	3.410	4.633	6.032
13	0.919	1.606	2.026	2.822	3.722	1.162	1.928	2.403	3.310	4.341	1.329	2.155	2.670	3.659	4.787	1.702	2.677	3.290	4.472	5.826
14	0.909	1.591	2.007	2.796	3.690	1.139	1.895	2.363	3.257	4.274	1.296	2.108	2.614	3.585	4.690	1.645	2.592	3.189	4.336	5.651
15	0.899	1.577	1.991	2.776	3.661	1.119	1.866	2.329	3.212	4.215	1.268	2.068	2.566	3.520	4.607	1.596	2.521	3.102	4.224	5.507
16	0.891	1.566	1.977	2.756	3.637	1.101	1.842	2.299	3.172	4.164	1.242	2.032	2.523	3.463	4.534	1.553	2.458	3.028	4.124	5.374
17	0.883	1.554	1.964	2.739	3.615	1.085	1.820	2.272	3.136	4.118	1.220	2.001	2.486	3.415	4.471	1.514	2.405	2.962	4.038	5.268
18	0.876	1.544	1.951	2.723	3.595	1.071	1.800	2.249	3.106	4.078	1.200	1.974	2.453	3.370	4.415	1.481	2.357	2.906	3.961	5.167
19	0.870	1.536	1.942	2.710	3.577	1.058	1.781	2.228	3.078	4.041	1.183	1.949	2.423	3.331	4.364	1.450	2.315	2.855	3.893	5.078
20	0.865	1.528	1.933	2.697	3.561	1.046	1.765	2.208	3.052	4.009	1.167	1.926	2.396	3.295	4.319	1.424	2.275	2.807	3.832	5.003
21	0.859	1.520	1.923	2.686	3.545	1.035	1.750	2.190	3.028	3.979	1.152	1.905	2.371	3.262	4.276	1.397	2.241	2.768	3.776	4.932
22	0.854	1.514	1.916	2.675	3.532	1.025	1.736	2.174	3.007	3.952	1.138	1.887	2.350	3.233	4.238	1.376	2.208	2.729	3.727	4.866
23	0.849	1.508	1.907	2.665	3.520	1.016	1.724	2.159	2.987	3.927	1.126	1.869	2.329	3.206	4.204	1.355	2.179	2.693	3.680	4.806
24	0.845	1.502	1.901	2.656	3.509	1.007	1.712	2.145	2.969	3.904	1.114	1.853	2.309	3.181	4.171	1.336	2.154	2.663	3.638	4.755
25	0.842	1.496	1.895	2.647	3.497	0.999	1.702	2.132	2.952	3.882	1.103	1.838	2.292	3.158	4.143	1.319	2.129	2.632	3.601	4.706
30	0.825	1.475	1.869	2.613	3.454	0.966	1.657	2.080	2.884	3.794	1.059	1.778	2.220	3.064	4.022	1.249	2.029	2.516	3.446	4.508
35	0.812	1.458	1.849	2.588	3.421	0.942	1.623	2.041	2.833	3.730	1.025	1.732	2.166	2.994	3.934	1.195	1.957	2.431	3.334	4.364
40	0.803	1.445	1.834	2.568	3.395	0.923	1.598	2.010	2.793	3.679	0.999	1.697	2.126	2.941	3.866	1.154	1.902	2.365	3.250	4.255
45	0.795	1.435	1.821	2.552	3.375	0.908	1.577	1.986	2.762	3.638	0.978	1.669	2.092	2.897	3.811	1.122	1.857	2.313	3.181	4.168
50	0.788	1.426	1.811	2.538	3.358	0.894	1.560	1.965	2.735	3.604	0.961	1.646	2.065	2.863	3.766	1.096	1.821	2.296	3.124	4.096

[a]From Engineering Statistics by Albert H. Bowker and Gerald J. Lieberman. Copyright 1959. Prentice-Hall, Inc. Used by permission.

TABLE III. - INPUT FORMAT FOR THE GENERAL SOLID PROPELLANT ROCKET MOTOR PERFORMANCE COMPUTER PROGRAM

Card number (a)	Variable name	Format	Columns	Description	Units
1	CASE	A6	1-6	Case number being evaluated	-
2	HEADING	80H	2-80	Designation of the motor being evaluated	-
3	NTIMES	I2	11-12	Integer designating the number of time intervals being evaluated (maximum of 9)	-
4	Description of time interval no. 1	80H	2-80	Definition of regressed distance no. 1 (such as ignition delay time)	-
5	Description of time interval no. 2	80H	2-80	Definition of regressed distance no. 2 (such as thrust rise time)	-
6	Description of time interval no. 3	80H	2-80	Definition of regressed distance no. 3 (such as ignition time)	-
7	Description of time interval no. 4	80H	2-80	Definition of regressed distance no. 4 (such as burn time)	-
8	Description of time interval no. 5	80H	2-80	Definition of regressed distance no. 5 (such as action time)	-
9	Description of time interval no. 6	80H	2-80	Definition of regressed distance no. 6 (such as tail-off time)	-
10	Description of time interval no. 7	80H	2-80	Definition of regressed distance no. 7 (such as total time)	-
11	Description of time interval no. 8	80H	2-80	Definition of regressed distance no. 8	-
12	Description of time interval no. 9	80H	2-80	Definition of regressed distance no. 9	-

[a]Cards 4 through 12 are shown; however, only NTIMES cards are required.

TABLE III. - INPUT FORMAT FOR THE GENERAL SOLID PROPELLANT ROCKET MOTOR PERFORMANCE COMPUTER PROGRAM - Continued

Card number	Variable name	Format	Columns	Description	Units
13	NP	I12	12	Integer designating the number of pressure altitudes at which the performance data are to be reported (up to 4)	-
	EXPA(1)[b]	F12.4	13-24	First pressure altitude at which the performance data are to be reported	psia
	EXPA(2)	F12.4	25-36	Continue for each individual pressure altitude at which the performance data are to be reported (up to 4)	psia
	EXPA(3)	F12.4	37-48		psia
	EXPA(4)	F12.4	49-60		psia
14	NT[c]	I12	12	Integer designating the number of prefire propellant temperatures to which the performance data are to be transformed (up to 3)	-
	EXTEM(1)[d]	F12.4	13-24	First prefire propellant temperature to which the performance data are to be transformed	°F
	EXTEM(2)[d]	F12.4	25-36	Continue for each individual prefire propellant temperature to which the performance data are to be transformed (up to 3)	°F
	EXTEM(3)[d]	F12.4	37-48		°F

[b]The first pressure altitude (at which the performance data are to be reported) must be zero psia (vacuum).

[c]The number of prefire propellant temperatures (to which the data are to be transformed) must equal the number of experimental temperature groups; therefore, NT on card 14 must equal NT on card 15.

[d]The value of EXTEM(3) must be greater than EXTEM(2) and EXTEM(2) must be greater than EXTEM(1).

TABLE III. - INPUT FORMAT FOR THE GENERAL SOLID PROPELLANT ROCKET MOTOR PERFORMANCE COMPUTER PROGRAM - Continued

Card number	Variable name	Format	Columns	Description	Units
15	NT[c]	I12	12	Integer designating the number of prefire propellant temperature group (up to 3)	-
	TGRP(1)[e]	F12.4	13-24	First prefire propellant temperature about which the motors are grouped (prefire propellant temperature group no. 1)	°F
	TGRP(2)[e]	F12.4	25-36	Continue for each individual prefire propellant temperature about which the motors are grouped (up to 3)	°F
	TGRP(3)[e]	F12.4	37-48		°F
16	G	F12.5	1-12	Acceleration of gravity	ft/sec^2
	PHI	F12.5	13-24	Nozzle cant (from the longitudinal center line of the motor)	rad
	NTOT	I2	25-26	Integer designating the time interval that is total time	-
	NBURN	I2	27-28	Integer designating the time interval that is burn time	-
	NFIRST[f]	I2	29-30	Integer designating the first time interval that all data are to be evaluated (see note)	-

[c]The number of prefire propellant temperatures (to which the data are to be transformed) must equal the number of experimental temperature groups; therefore, NT on card 14 must equal NT on card 15.

[e]The value of TGRP(3) must be greater than TGRP(2), and TGRP(2) must be greater than TGRP(1).

[f]In some cases (as in ignition transient data, for example) it is desirable to transform only the time data. This is accomplished in the program by inputing "time only" data in the first few time intervals, and setting the value of NFIRST to indicate the first time interval that all data are to be evaluated.

TABLE III. - INPUT FORMAT FOR THE GENERAL SOLID PROPELLANT ROCKET MOTOR PERFORMANCE COMPUTER PROGRAM - Continued

Card number (g)	Variable name	Format	Columns	Description	Units
17	CT1	F12.4	1-12	One-sided tolerance factor for transformed data	-
	CT2	F12.4	13-24	Two-sided tolerance factor for transformed data	-
	CP1	F12.4	25-36	One-sided tolerance factor for nontransformed data	-
	CP2	F12.4	36-48	Two-sided tolerance factor for nontransformed data	-
18	XMOT	A6	1-6	Motor number	-
	FIRETP	F12.4	7-18	Prefire propellant temperature	°F
	ENDMOT	I2	19-2	Integer designating that the last motor to be processed has been reached (1 for last motor, 0 for all other motors)	-
19	AT1	E12.5	1-12	Throat area of nozzle no. 1	$in.^2$
	AT2	E12.5	13-24	Continue for each individual nozzle (up to 4)	$in.^2$
	AT3	E12.5	25-36		$in.^2$
	AT4	E12.5	37-48		$in.^2$

[g]Cards 18 through 26 are required for each individual motor.

TABLE III. - INPUT FORMAT FOR THE GENERAL SOLID PROPELLANT ROCKET MOTOR PERFORMANCE COMPUTER PROGRAM - Continued.

Card number (g)	Variable name	Format	Columns	Description	Units
20	AE1	E12.5	1-12	Exit area of nozzle no. 1	$in.^2$
	AE2	E12.5	13-24	Continue for each individual nozzle (up to 4)	$in.^2$
	AE3	E12.5	25-36		$in.^2$
	AE4	E12.5	37-48		$in.^2$
21	WP	E12.5	1-12	Total propellant weight	lbf
	WEB	E12.5	13-24	Web thickness	in.
	DENS	E12.5	25-36	Propellant density	$lb/in.^3$
	PAF	E12.5	37-48	Ambient pressure at the time of motor firing	psia
22	Time 1	F9.4	1-9	Time interval required to achieve regress distance no. 1	sec
	Time 2	F9.4	10-18	Continue for each individual time interval being evaluated (up to NTIMES)	sec
	Time 3	F9.4	19-27		sec
	Time 4	F9.4	28-36		sec
	Time 5	F9.4	37-45		sec
	Time 6	F9.4	46-54		sec
	Time 7	F9.4	55-63		sec
	Time 8	F9.4	64-72		sec
	Time 9	F8.4	73-80		sec

[g]Cards 18 through 26 are required for each individual motor.

TABLE III. - INPUT FORMAT FOR THE GENERAL SOLID PROPELLANT ROCKET MOTOR PERFORMANCE COMPUTER PROGRAM - Continued

Card number (g)	Variable name (h)	Format	Columns	Description	Units
23	INTP1	F9.4	1-9	Measured chamber pressure integral during Time interval 1	psia-sec
	INTP2	F9.4	10-18	Continue for each individual time interval being evaluated (up to NTIMES)	psia-sec
	INTP3	F9.4	19-27		psia-sec
	INTP4	F9.4	28-36		psia-sec
	INTP5	F9.4	37-45		psia-sec
	INTP6	F9.4	46-54		psia-sec
	INTP7	F9.4	55-63		psia-sec
	INTP8	F9.4	64-72		psia-sec
	INTP9	F8.4	73-80		psia-sec
24	P1	F9.4	1-9	Measured average chamber pressure during Time interval 1	psia
	P2	F9.4	10-18	Continue for each individual time interval being evaluated (up to NTIMES)	psia

[g]Cards 18 through 26 are required for each individual motor.

[h]Average or integral data (but not both) are required for a particular time interval; however, cards 23, 24, 25, and 26 are required because average and integral data may be intermingled with the various time intervals.

TABLE III. - INPUT FORMAT FOR THE GENERAL SOLID PROPELLANT

ROCKET MOTOR PERFORMANCE COMPUTER PROGRAM - Continued

Card number (g)	Variable name (h)	Format	Columns	Description	Units
24	P3	F9.4	19-27	Continue for each individual time interval being evaluated (up to NTIMES)	psia
	P4	F9.4	28-36		psia
	P5	F9.4	37-45		psia
	P6	F9.4	46-54		psia
	P7	F9.4	55-63		psia
	P8	F9.4	64-72		psia
	P9	F8.4	73-80		psia
25	INTF1	F9.4	1-9	Measured impulse during Time interval 1	lbf-sec
	INTF2	F9.4	10-18	Continue for each individual time interval being evaluated (up to NTIMES)	lbf-sec
	INTF3	F9.4	19-27		lbf-sec
	INTF4	F9.4	28-36		lbf-sec
	INTF5	F9.4	37-45		lbf-sec
	INTF6	F9.4	46-54		lbf-sec
	INTF7	F9.4	55-63		lbf-sec

[g]Cards 18 through 26 are required for each individual motor.

[h]Average or integral data (but not both) are required for a particular time interval; however, cards 23, 24, 25, and 26 are required because average and integral data may be intermingled with the various time intervals.

TABLE III. - INPUT FORMAT FOR THE GENERAL SOLID PROPELLANT ROCKET MOTOR PERFORMANCE COMPUTER PROGRAM - Concluded

Card number (g)	Variable name (h)	Format	Columns	Description	Units
25	INTF8	F9.4	64-72	Continue for each individual time interval being evaluated (up to NTIMES)	lbf-sec
	INTF9	F8.4	73-80		lbf-sec
26	F1	F9.4	1-9	Measured average thrust during Time interval 1	lbf
	F2	F9.4	10-18	Continue for each individual time interval being evaluated (up to (NTIMES)	lbf
	F3	F9.4	19-27		lbf
	F4	F9.4	28-36		lbf
	F5	F9.4	37-45		lbf
	F6	F9.4	46-54		lbf
	F7	F9.4	55-63		lbf
	F8	F9.4	64-72		lbf
	F9	F8.4	73-80		lbf

[g] Cards 18 through 26 are required for each individual motor.

[h] Average or integral data (but not both) are required for a particular time interval; however, cards 23, 24, 25, and 26 are required because average and integral data may be intermingled with the various time intervals.

TABLE IV. - TYPICAL NONTRANSFORMED DATA OUTPUT FORMAT FOR THE GENERAL SOLID PROPELLANT ROCKET MOTOR PERFORMANCE COMPUTER PROGRAM

PA = -.000

IMPULSE MOTOR NO.	IT1	IT2	IT3	IT4	IT5	IT6	IT7	IT8	IT9
TEMP GROUP 1									
37A	.00	.00	479.00	1510.07	1755.88	1781.61	271.11	.00	.00
70	.00	.00	730.06	1515.86	1728.77	1752.70	236.41	.00	.00
44A	.00	.00	630.96	1498.65	1757.28	1780.02	280.95	.00	.00
45A	.00	.00	544.38	1535.37	1772.00	1795.93	260.17	.00	.00
43A	.00	.00	549.86	1525.38	1771.66	1798.37	272.60	.00	.00
74A	.00	.00	688.55	1556.49	1768.54	1794.93	238.10	.00	.00
MEAN	.00	.00	603.80	1523.64	1759.02	1783.93	259.89	.00	.00
STAND. DEV.	.00	.00	95.79	20.44	16.41	17.15	18.75	.00	.00
TEMP GROUP 2									
77	.00	.00	552.44	1496.92	1754.25	1779.85	282.59	.00	.00
50A	.00	.00	780.61	1532.28	1771.82	1797.06	264.50	.00	.00
38A	.00	.00	742.15	1470.46	1723.10	1751.85	281.05	.00	.00
59A	.00	.00	708.81	1544.61	1765.84	1787.62	242.64	.00	.00
MEAN	.00	.00	696.00	1511.07	1753.75	1779.09	267.69	.00	.00
STAND. DEV.	.00	.00	100.11	33.78	21.70	19.48	18.60	.00	.00
TEMP GROUP 3									
69	.00	.00	848.42	1485.66	1755.91	1778.68	292.68	.00	.00
64A	.00	.00	620.51	1537.45	1797.06	1820.64	282.93	.00	.00
58A	.00	.00	860.31	1508.21	1801.22	1823.41	314.78	.00	.00
65A	.00	.00	705.46	1521.27	1795.95	1811.99	290.27	.00	.00
40A	.00	.00	[illegible]92.60	1505.43	1774.35	1800.10	294.36	.00	.00
78	.00	.00	682.98	1513.31	1747.85	1772.93	259.16	.00	.00
60A	.00	.00	796.89	1503.47	1768.28	1794.46	290.66	.00	.00
MEAN	.00	.00	729.60	1510.68	1777.23	1800.32	289.26	.00	.00
STAND. DEV.	.00	.00	107.36	16.06	21.31	19.73	16.52	.00	.00

TABLE V.- TYPICAL TRANSFORMED DATA OUTPUT FORMAT FOR THE GENERAL SOLID PROPELLANT ROCKET MOTOR PERFORMANCE COMPUTER PROGRAM

[Transformed impulse]

TIME NO. 5

PA = -.00

MOTOR NO.	TEMP1	TEMP2	TEMP3	TEMP4
TEMP GROUP 1				
37A	1717.24	1808.10	1997.85	.00
70	1598.79	1683.38	1860.04	.00
44A	1687.36	1776.63	1963.08	.00
45A	1708.40	1798.20	1986.00	.00
43A	1691.79	1780.72	1966.69	.00
74A	1690.42	1779.28	1965.10	.00
MEAN	1682.33	1771.05	1956.46	.00
STANDARD DEV.	42.56	44.69	49.20	.00
TEMP GROUP 2				
77	1617.59	1704.09	1884.33	.00
50A	1725.21	1816.26	2006.52	.00
38A	1660.79	1748.16	1930.84	.00
59A	1707.16	1796.08	1982.41	.00
MEAN	1677.69	1766.15	1951.03	.00
STANDARD DEV.	48.39	50.28	54.53	.00
TEMP GROUP 3				
69	1636.31	1723.54	1905.41	.00
64A	1672.84	1762.01	1947.95	.00
58A	1746.90	1840.02	2034.19	.00
65A	1676.67	1765.46	1950.87	.00
40A	1681.40	1769.57	1954.06	.00
78	1633.62	1718.44	1896.29	.00
60A	1678.60	1765.76	1948.50	.00
MEAN	1675.19	1763.54	1948.18	.00
STANDARD DEV.	37.51	39.86	44.65	.00
TOTAL MEAN	1678.30	1766.81	1951.77	.00
TOTAL STANDARD DEV.	39.28	41.30	45.56	.00
CONFIDENCE ON NORMAL DISTRIBUTION				
(ONE MIN	1537.47	1618.76	1788.44	.00
SIDED MAX	1819.13	1914.86	2115.11	.00
(TWO MIN	1522.94	1603.48	1771.58	.00
SIDED MAX	1833.66	1930.14	2131.97	.00

TABLE VI- INPUT FORMAT FOR THE SOLID PROPELLANT ROCKET MOTOR

PERFORMANCE VERSUS TIME COMPUTER PROGRAM

Card number (a)	Variable name	Format	Columns	Description	Units
1	NXNW	I6	1-6	Integer designating the number of subintervals into which web time is divided (up to 5)	-
	NXNT	I6	7-12	Integer designating the number of subintervals into which tail-off time is divided (up to 4)	-
	TD	F12.4	13-24	Prefire propellant temperature to which the performance data are to be transformed	°F
	PHI	F12.4	25-36	Nozzle cant angle (from the longitudinal center line of the motor)	rad
	CT2	F12.4	37-48	Tolerance factor (one- or two-sided)	-
	NSW[b]	I6	49-54	Integer designating the method by which T0 is to be determined (0 if T0 is inputed, 1 if T0 is to be determined by the program as a function of chamber pressure)	-
1A	PSIA[b]	F10.0	1-10	Value of chamber pressure at which TO is determined (card not included if NSW = 0)	psia
2	PCTW1	F10.4	1-10	Upper limit of first web-time subinterval	percent
	NW1	I6	11-16	Integer designating the number of equally divided percent-web-times in the first subinterval that time, chamber pressure, and thrust transformations are to be performed	-
	PCTW2	F10.4	17-26	Continue for each subinterval up to NXNW (last subinterval must equal 100 percent web time)	percent

[a]Cards 1 through 10 apply for all motors and are read in only once; however, cards 11 through 13 + $\frac{NPTS}{2}$ are required for each motor and must be entered in sequence.

[b]Card 1A can be omitted if TO is an input value (NSW = 0).

TABLE VI. - INPUT FORMAT FOR THE SOLID PROPELLANT ROCKET MOTOR PERFORMANCE VERSUS TIME COMPUTER PROGRAM - Continued

Card number (a)	Variable name	Format	Columns	Description	Units
2	NW2	I6	27-32	Continue for each subinterval up to NXNW (last subinterval must equal 100 percent web time)	-
	PCTW3	F10.4	33-42		percent
	NW3	I6	43-48		-
	PCTW4	F10.4	49-58		percent
	NW4	I6	59-64		-
	PCTW5	F10.4	65-74		percent
	NW5	I6	75-80		-
3	PCTT1	F10.4	1-10	Upper limit of first tail-off time subinterval	percent
	NT1	I6	11-16	Integer designating the number of equally divided percent tail-off times in the first subinterval that time, chamber pressure, and thrust transformations are to be performed	-
	PCTT2	F10.4	17-26	Continue for each subinterval up to NXNT (last subinterval must equal 100 percent tail-off time)	percent
	NT2	I6	27-32		-
	PCTT3	F10.4	33-42		percent
	NT3	I6	43-48		-

[a]Cards 1 through 10 apply for all motors and are read in only once; however, cards 11 through 13 + $\frac{NPTS}{2}$ are required for each motor and must be entered in sequence.

TABLE VI. - INPUT FORMAT FOR THE SOLID PROPELLANT ROCKET MOTOR

PERFORMANCE VERSUS TIME COMPUTER PROGRAM - Continued

Card number (a)	Variable name	Format	Columns	Description	Units
3	PCTT4	F10.4	49-58	Continue for each subinterval up to NXNT (last subinterval must equal 100 percent tail-off time)	percent
	NT4	I6	59-64		-
4	NA	I12	1-12	Integer designating the number of prefire propellant temperatures about which the motors are grouped (up to 3)	-
	TGRP1[c]	F12.4	13-24	First prefire propellant temperature about which the motors are grouped (prefire propellant temperature group no. 1)	°F
	TGRP2[c]	F12.4	25-36	Continue for each prefire propellant temperature about which the motors are grouped (up to 3)	°F
	TGRP3[c]	F12.4	37-48		°F
5	NP	I12	1-12	Integer designating the number of pressure altitudes at which thrust is to be reported (up to 3)	-
	EXPA1	F12.4	13-24	The first pressure altitude at which thrust is to be reported	psia
	EXPA2	F12.4	25-36	Continue for each pressure altitude at which thrust is to be reported (up to 3)	psia
	EXPA3	F12.4	37-48		psia
6	HEADX	12A6	1-72	Heading for the abscissa of all graphs (time axis)	sec
7	HEAD1	12A6	1-72	Ordinate heading on plot of chamber pressure versus time	psia

[a]Cards 1 through 10 apply for all motors and are read in only once; however, cards 11 through $13 + \frac{NPTS}{2}$ are required for each motor and must be entered in sequence.

[c]The value of TGRP(3) must be greater than TGRP(2), and TGRP(2) must be greater than TGRP(1).

TABLE VI. - INPUT FORMAT FOR THE SOLID PROPELLANT ROCKET MOTOR PERFORMANCE VERSUS TIME COMPUTER PROGRAM - Continued

Card number (a),(d)	Variable name	Format	Columns	Description	Units
8	HEAD2	12A6	1-72	Ordinate heading on plot of thrust at pressure altitude no. 1 (EXPA 1) versus time	lbf
9	HEAD3	12A6	1-72	Ordinate heading on plot of thrust at pressure altitude no. 2 (EXPA 2) versus time	lbf
10	HEAD4	12A6	1-72	Ordinate heading on plot of thrust at pressure altitude no. 3 (EXPA 3) versus time	lbf
11	XMOT	A6	1-6	Motor number	-
	FIRETP	F12.4	7-18	Prefire propellant temperature	°F
	PAX	F12.4	19-30	Ambient pressure at the time of motor firing	psia
	TB	F12.4	31-42	Burn-time for the motor	sec
	TT	F12.4	43-54	Total time for the motor	sec
	TO	F12.4	55-66	Time offset (time that is zero percent web time)	sec
	PCP	F12.4	67-78	Prefire chamber pressure (used to convert input chamber pressures from psig to psia)	psia
	NDMOT	I1	80	Integer designating that the last motor to be processed has been reached (1 for last motor, 0 for all other motors)	-
12	AE1	E12.5	1-12	Exit area of nozzle no. 1	$in.^2$
	AE2	E12.5	13-24	Continue for each nozzle (up to 4)	$in.^2$

[a]Cards 1 through 10 apply for all motors and are read in only once; however, cards 11 through $13 + \frac{NPTS}{2}$ are required for each motor and must be entered in sequence.

[d]If fewer than three ambient pressures are used, cards 9 and 10 must be present, even as blank cards.

TABLE VI. - INPUT FORMAT FOR THE SOLID PROPELLANT ROCKET MOTOR PERFORMANCE VERSUS TIME COMPUTER PROGRAM - Concluded

Card number (a)	Variable name	Format	Columns	Description	Units
12	AE3	E12. 5	25-36	Continue for each nozzle (up to 4)	$in.^2$
	AE4	E12. 5	37-48		$in.^2$
13	NPTS	I6	1-6	Integer designating number of points to be read in to define motor performance versus time (up to 200)	-
	ICON	I6	7-12	Integer designating the number of input points in piecewise curve fit (6 is a good choice)	-
	IPRNT	I6	13-18	Integer designating that the curve fit coefficients are to be printed out (0 for no coefficients, 1 for all coefficients)	-
14	TREP1	E16. 8	1-16	First time input in the performance input array	sec
	PC1	E12. 5	17-28	Chamber pressure that corresponds to the time input in the performance input array	psig
	F1	E12. 5	29-40	Thrust that corresponds to time input in the performance input array	lbf
	TREP2	E16. 5	41-56	Continue for each chamber pressure and thrust at the proper time (up to NPTS) at 2 times, pressures and thrusts inputs per card	sec
	PC2	E12. 5	57-68		psig
	F2	E12. 5	69-80		lbf

[a]Cards 1 through 10 apply for all motors and are read in only once; however, cards 11 through 13 + $\frac{NPTS}{2}$ are required for each motor and must be entered in sequence.

TABLE VII. - TRANSFORMED TIME AND CHAMBER-PRESSURE STATISTICAL DATA OUTPUT FORMAT FOR THE SOLID PROPELLANT ROCKET MOTOR PERFORMANCE VERSUS TIME COMPUTER PROGRAM

PCT. WEB TIME	TRANSFORMED TIMES MEAN	MIN.	MAX.	TRANSFORMED CHAMBER PRESSURE MEAN	MIN.	MAX.
	MEANS WITH TWO SIDED TOLERANCE LIMITS					
.00	.0000	-.0000	.0000	100.0727	96.2117	103.9337
.81	.0050	.0045	.0054	186.2759	47.4130	325.1388
1.61	.0099	.0091	.0108	288.4526	76.4223	500.4828
2.42	.0149	.0136	.0162	400.0431	154.3467	645.7395
3.23	.0199	.0182	.0215	514.5706	243.7581	785.3831
4.03	.0248	.0227	.0269	624.0311	338.0396	910.0225
4.84	.0298	.0273	.0323	731.4412	443.1322	1019.7502
5.65	.0348	.0318	.0377	834.9045	548.8521	1120.9568
6.45	.0397	.0364	.0431	928.8627	649.3971	1208.3282
7.26	.0447	.0409	.0485	1011.5578	735.0849	1288.0307
8.06	.0497	.0455	.0538	1084.9299	807.9524	1361.9073
8.87	.0546	.0500	.0592	1150.4757	871.0561	1429.8953
9.68	.0596	.0545	.0646	1208.8554	926.5884	1491.1225
10.48	.0645	.0591	.0700	1261.3881	974.6762	1548.0998
11.29	.0695	.0636	.0754	1308.2346	1017.1144	1599.3549
12.10	.0745	.0682	.0808	1350.0422	1056.0642	1644.0203
12.90	.0794	.0727	.0862	1387.5762	1091.8444	1683.3080
13.71	.0844	.0773	.0915	1421.3233	1125.6662	1716.9805
14.52	.0894	.0818	.0969	1451.6023	1157.9969	1745.2076
15.32	.0943	.0864	.1023	1478.7972	1188.1652	1769.4291
16.13	.0993	.0909	.1077	1503.5798	1216.8781	1790.2816
16.94	.1043	.0955	.1131	1526.5542	1244.7761	1808.3324
17.74	.1092	.1000	.1185	1546.9323	1270.6313	1823.2334
18.55	.1142	.1045	.1238	1564.9614	1294.1378	1835.7851
19.35	.1192	.1091	.1292	1581.5182	1315.8552	1847.1812
20.16	.1241	.1136	.1346	1596.2879	1335.8879	1856.6880
20.97	.1291	.1182	.1400	1609.3506	1354.5360	1864.1652
21.77	.1341	.1227	.1454	1620.4328	1371.0981	1869.7676
22.58	.1390	.1273	.1508	1630.3001	1386.5419	1874.0583
23.39	.1440	.1318	.1562	1639.5942	1401.9310	1877.2574
24.19	.1490	.1364	.1615	1647.5151	1414.6645	1880.3656
25.00	.1539	.1409	.1669	1654.6235	1426.0915	1883.1555
25.81	.1589	.1455	.1723	1661.2788	1437.3437	1885.2139
26.61	.1638	.1500	.1777	1666.9022	1447.2263	1886.5780
27.42	.1688	.1545	.1831	1672.2225	1456.2725	1888.1725
28.23	.1738	.1591	.1885	1676.4756	1464.7989	1888.1524
29.03	.1787	.1636	.1938	1680.2541	1471.3614	1889.1469
29.84	.1837	.1682	.1992	1683.6993	1476.9883	1890.4102
30.65	.1887	.1727	.2046	1686.7656	1482.4679	1891.0632
31.45	.1936	.1773	.2100	1689.6254	1485.9445	1893.3062
32.26	.1986	.1818	.2154	1691.9939	1489.3389	1894.6489
33.06	.2036	.1864	.2208	1694.2330	1492.0268	1896.4392
33.87	.2085	.1909	.2261	1696.0794	1493.4425	1898.7162
34.68	.2135	.1955	.2315	1697.3031	1495.5069	1899.0993
35.48	.2185	.2000	.2369	1698.3392	1497.4662	1899.2123
36.29	.2234	.2045	.2423	1698.8275	1498.6623	1898.9928
37.10	.2284	.2091	.2477	1699.3327	1499.6715	1898.9940
37.90	.2334	.2136	.2531	1699.4336	1500.4301	1898.4371
38.71	.2383	.2182	.2585	1699.4032	1501.1038	1897.7027
39.52	.2433	.2227	.2638	1699.6042	1501.0358	1898.1726
40.32	.2483	.2273	.2692	1699.4354	1500.4852	1898.3857
41.13	.2532	.2318	.2746	1698.9325	1498.8431	1899.0219
41.94	.2582	.2364	.2800	1698.1861	1497.3153	1899.0568
42.74	.2631	.2409	.2854	1697.0608	1495.6663	1898.4554

TABLE VII. - TRANSFORMED TIME AND CHAMBER-PRESSURE STATISTICAL DATA OUTPUT FORMAT FOR THE SOLID PROPELLANT ROCKET MOTOR PERFORMANCE VERSUS TIME COMPUTER PROGRAM - Continued

43.55	.2681	.2455	.2908	1695.5937	1493.6824	1897.5050
44.35	.2731	.2500	.2961	1694.0745	1491.7364	1896.4126
45.16	.2780	.2545	.3015	1692.4855	1489.7363	1895.2347
45.97	.2830	.2591	.3069	1690.9648	1487.6661	1894.2635
46.77	.2880	.2636	.3123	1689.2429	1486.1593	1892.3265
47.58	.2929	.2682	.3177	1687.9059	1484.8948	1890.9170
48.39	.2979	.2727	.3231	1686.0139	1482.4461	1889.5818
49.19	.3029	.2773	.3285	1683.7669	1480.0048	1887.5290
50.00	.3078	.2818	.3338	1681.6899	1477.6165	1885.7634
50.81	.3128	.2864	.3392	1679.0422	1474.1835	1883.9009
51.61	.3178	.2909	.3446	1676.3945	1470.2831	1882.5059
52.42	.3227	.2955	.3500	1673.4878	1466.3678	1880.6078
53.23	.3277	.3000	.3554	1670.3387	1463.3387	1877.3388
54.03	.3327	.3045	.3608	1667.2517	1460.9975	1873.5059
54.84	.3376	.3091	.3661	1663.6708	1457.1346	1870.2070
55.65	.3426	.3136	.3715	1659.7595	1453.2724	1866.2467
56.45	.3476	.3182	.3769	1655.1865	1449.5914	1860.7816
57.26	.3525	.3227	.3823	1650.3546	1444.5683	1856.1410
58.06	.3575	.3273	.3877	1645.4045	1439.3457	1851.4635
58.87	.3624	.3318	.3931	1639.4879	1433.0053	1845.9706
59.68	.3674	.3364	.3985	1633.2585	1426.7380	1839.7791
60.48	.3724	.3409	.4038	1626.4598	1420.3194	1832.6002
61.29	.3773	.3455	.4092	1619.3300	1412.7360	1825.9241
62.10	.3823	.3500	.4146	1612.2121	1405.1977	1819.2267
62.90	.3873	.3545	.4200	1604.6806	1399.5986	1809.7626
63.71	.3922	.3591	.4254	1596.7055	1392.7685	1800.6425
64.52	.3972	.3636	.4308	1588.9195	1386.0671	1791.7719
65.32	.4022	.3682	.4361	1581.3076	1377.7398	1784.8755
66.13	.4071	.3727	.4415	1573.1360	1369.0930	1777.1790
66.94	.4121	.3773	.4469	1564.8889	1361.5914	1768.1865
67.74	.4171	.3818	.4523	1556.3409	1351.5890	1761.0929
68.55	.4220	.3864	.4577	1547.9214	1344.4722	1751.3705
69.35	.4270	.3909	.4631	1538.9636	1336.8781	1741.0492
70.16	.4320	.3955	.4685	1529.6991	1328.6521	1730.7462
70.97	.4369	.4000	.4738	1520.4931	1320.7040	1720.2822
71.77	.4419	.4046	.4792	1511.7381	1313.5326	1709.9436
72.58	.4469	.4091	.4846	1503.2595	1306.7792	1699.7398
73.39	.4518	.4136	.4900	1494.4600	1299.1186	1689.8014
74.19	.4568	.4182	.4954	1485.9684	1292.4371	1679.4997
75.00	.4617	.4227	.5008	1477.3739	1285.7598	1668.9881
75.81	.4667	.4273	.5061	1467.8084	1277.2995	1658.3173
76.61	.4717	.4318	.5115	1457.9984	1268.0385	1647.9583
77.42	.4766	.4364	.5169	1447.9451	1258.8627	1637.0274
78.23	.4816	.4409	.5223	1438.8756	1252.0673	1625.6840
79.03	.4866	.4455	.5277	1430.2524	1245.2208	1615.2841
79.84	.4915	.4500	.5331	1421.2985	1237.7059	1604.8912
80.65	.4965	.4546	.5384	1411.9651	1229.8191	1594.1112
81.45	.5015	.4591	.5438	1402.8683	1222.0734	1583.6631
82.26	.5064	.4636	.5492	1393.4044	1213.5523	1573.2565
83.06	.5114	.4682	.5546	1384.1218	1205.4579	1562.7856
83.87	.5164	.4727	.5600	1374.9182	1197.3819	1552.4545
84.68	.5213	.4773	.5654	1365.6267	1189.2108	1542.0426
85.48	.5263	.4818	.5708	1356.5140	1181.2270	1531.8011
86.29	.5313	.4864	.5761	1347.2283	1172.4927	1521.9639
87.10	.5362	.4909	.5815	1337.7622	1163.4899	1512.0345
87.90	.5412	.4955	.5869	1328.4356	1154.5029	1502.3684
88.71	.5462	.5000	.5923	1318.8686	1145.1311	1492.6062
89.52	.5511	.5046	.5977	1308.9783	1136.0403	1481.9163

TABLE VII. - TRANSFORMED TIME AND CHAMBER-PRESSURE STATISTICAL DATA OUTPUT FORMAT FOR THE SOLID PROPELLANT ROCKET MOTOR PERFORMANCE VERSUS TIME COMPUTER PROGRAM - Continued

90.32	.5561	.5091	.6031	1299.0389	1126.5444	1471.5334
91.13	.5610	.5136	.6084	1288.9000	1116.8996	1460.9004
91.94	.5660	.5182	.6138	1278.3531	1106.6177	1450.0885
92.74	.5710	.5227	.6192	1267.3961	1095.2660	1439.5262
93.55	.5759	.5273	.6246	1256.3310	1084.2116	1428.4503
94.35	.5809	.5318	.6300	1245.2392	1072.4862	1417.9923
95.16	.5859	.5364	.6354	1234.1475	1060.2894	1408.0054
95.97	.5908	.5409	.6408	1223.1327	1048.0887	1398.1768
96.77	.5958	.5455	.6461	1212.0997	1035.5067	1388.6929
97.58	.6008	.5500	.6515	1200.4740	1021.8971	1379.0508
98.39	.6057	.5546	.6569	1185.8283	1003.6930	1367.9636
99.19	.6107	.5591	.6623	1168.4571	981.6796	1355.2347
100.00	.6157	.5636	.6677	1145.7223	952.4611	1338.9837
PCT. TAILOFF TIME						
1.33	.6206	.5686	.6725	1118.7697	924.0121	1313.5273
2.67	.6254	.5736	.6773	1086.7690	892.7339	1280.8042
4.00	.6303	.5785	.6821	1050.3857	856.7386	1244.0328
5.33	.6352	.5835	.6870	1011.7241	821.6552	1201.7930
6.67	.6401	.5883	.6919	971.5473	786.4144	1156.6802
8.00	.6450	.5932	.6968	931.6782	752.8625	1110.4938
9.33	.6499	.5980	.7017	892.6076	721.8643	1063.3509
10.67	.6548	.6029	.7067	855.3248	693.0366	1017.6131
12.00	.6597	.6077	.7117	819.4540	665.4108	973.4971
13.33	.6646	.6124	.7167	785.5828	639.5456	931.6200
14.67	.6694	.6171	.7217	753.2597	615.1995	891.3198
16.00	.6743	.6219	.7268	722.6703	592.2510	853.0896
17.33	.6792	.6265	.7319	693.6676	570.4025	816.9326
18.67	.6841	.6312	.7370	666.1639	549.3537	782.9741
20.00	.6890	.6359	.7422	640.3914	529.4356	751.3472
21.33	.6939	.6405	.7473	615.8021	510.9543	720.6499
22.67	.6988	.6451	.7525	592.5075	493.9920	691.0230
24.00	.7037	.6496	.7577	570.0886	477.3924	662.7849
25.33	.7086	.6542	.7629	548.8185	461.4724	636.1645
26.67	.7134	.6587	.7682	528.5498	446.1409	610.9587
28.00	.7183	.6632	.7735	509.2877	431.4305	587.1449
29.33	.7232	.6677	.7788	491.0974	417.5962	564.5985
30.67	.7281	.6721	.7841	473.4347	403.6292	543.2402
32.00	.7330	.6766	.7894	456.6112	390.2160	523.0063
33.33	.7379	.6810	.7948	440.9169	377.3766	504.4572
34.67	.7428	.6854	.8002	425.7773	365.3657	486.1889
36.00	.7477	.6898	.8056	411.5136	354.3595	468.6677
37.33	.7526	.6942	.8110	398.0980	343.7363	452.4596
38.67	.7575	.6985	.8164	384.9482	333.3091	436.5872
40.00	.7623	.7028	.8219	372.6211	323.3540	421.8882
41.33	.7672	.7071	.8273	360.9189	314.0585	407.7792
42.67	.7721	.7114	.8328	349.3216	304.5711	394.0722
44.00	.7770	.7157	.8383	338.4454	295.3758	381.5149
45.33	.7819	.7200	.8438	327.8011	286.6873	368.9149
46.67	.7868	.7242	.8494	317.3519	278.4996	356.2041
48.00	.7917	.7285	.8549	307.4221	269.9771	344.8672
49.33	.7966	.7327	.8605	297.9488	261.7621	334.1354
50.67	.8015	.7369	.8660	288.7212	254.5062	322.9362
52.00	.8063	.7411	.8716	279.9151	247.4768	312.3534
53.33	.8112	.7453	.8772	271.7372	240.9243	302.5501
54.67	.8161	.7494	.8828	263.6080	234.3644	292.8517
56.00	.8210	.7536	.8884	255.9788	228.0174	283.9402

TABLE VII. - TRANSFORMED TIME AND CHAMBER-PRESSURE STATISTICAL DATA OUTPUT FORMAT FOR THE SOLID PROPELLANT ROCKET MOTOR PERFORMANCE VERSUS TIME COMPUTER PROGRAM - Concluded

57.33	.8259	.7577	.8941	248.3934	221.6398	275.1469
58.67	.8308	.7619	.8997	241.0648	215.5795	266.5502
60.00	.8357	.7660	.9054	234.3061	209.6224	258.9898
61.33	.8406	.7701	.9110	227.6651	203.7331	251.5970
62.67	.8455	.7742	.9167	221.2060	198.5070	243.9050
64.00	.8503	.7783	.9224	214.8457	193.2729	236.4185
65.33	.8552	.7824	.9281	208.5310	187.7487	229.3132
66.67	.8601	.7865	.9338	202.1990	182.8354	221.5626
68.00	.8650	.7905	.9395	196.4787	178.2372	214.7202
69.33	.8699	.7946	.9452	191.1092	173.6381	208.5804
70.67	.8748	.7986	.9510	185.6251	169.1642	202.0860
72.00	.8797	.8027	.9567	180.2799	164.7641	195.7957
73.33	.8846	.8067	.9625	175.4276	160.8767	189.9785
74.67	.8895	.8107	.9682	170.6171	156.4750	184.7591
76.00	.8944	.8147	.9740	165.6604	152.0464	179.2745
77.33	.8992	.8187	.9797	160.6167	147.5117	173.7218
78.67	.9041	.8227	.9855	156.2894	143.6445	168.9342
80.00	.9090	.8267	.9913	152.1756	139.9957	164.3556
81.33	.9139	.8307	.9971	147.9813	136.9321	159.0305
82.67	.9188	.8347	1.0029	144.0087	133.8443	154.1731
84.00	.9237	.8387	1.0087	139.9193	130.7487	149.0900
85.33	.9286	.8427	1.0145	136.0162	128.7390	143.2933
86.67	.9335	.8466	1.0203	132.1325	125.1921	139.0730
88.00	.9384	.8506	1.0261	128.4227	121.7876	135.0579
89.33	.9432	.8545	1.0320	124.8258	118.3964	131.2551
90.67	.9481	.8585	1.0378	121.6686	115.0492	128.2881
92.00	.9530	.8624	1.0436	118.5244	111.5678	125.4810
93.33	.9579	.8664	1.0495	115.4463	108.7479	122.1447
94.67	.9628	.8703	1.0553	112.2210	106.0667	118.3753
96.00	.9677	.8742	1.0611	108.9699	103.3730	114.5669
97.33	.9726	.8782	1.0670	106.0028	100.9064	111.0992
98.67	.9775	.8821	1.0728	103.1398	98.1133	108.1662
100.00	.9824	.8860	1.0787	100.3185	95.4782	105.1589

TABLE VIII. - TRANSFORMED TIME-AND-THRUST STATISTICAL DATA OUTPUT FORMAT FOR THE SOLID PROPELLANT ROCKET MOTOR PERFORMANCE VERSUS TIME COMPUTER PROGRAM

	TRANSFORMED TIMES			TRANSFORMED THRUST AT PA = .00		
	MEANS WITH TWO SIDED TOLERANCE LIMITS					
PCT. WEB TIME	MEAN	MIN.	MAX.	MEAN	MIN.	MAX.
.00	.0000	-.0000	.0000	29.9100	-85.4260	145.2459
.81	.0050	.0045	.0054	128.6946	-81.6653	339.0546
1.61	.0099	.0091	.0108	281.9090	1.7245	562.0935
2.42	.0149	.0136	.0162	471.5607	140.4345	802.6869
3.23	.0199	.0182	.0215	694.4515	321.5112	1067.3919
4.03	.0248	.0227	.0269	924.5868	547.2773	1301.8962
4.84	.0298	.0273	.0323	1137.4147	732.9413	1541.8882
5.65	.0348	.0318	.0377	1333.2352	896.6773	1769.7933
6.45	.0397	.0364	.0431	1509.5691	1058.4309	1960.7075
7.26	.0447	.0409	.0485	1663.7580	1204.7103	2122.8056
8.06	.0497	.0455	.0538	1801.8282	1336.0550	2267.6016
8.87	.0546	.0500	.0592	1920.1355	1451.1768	2389.0942
9.68	.0596	.0545	.0646	2020.9526	1542.3207	2499.5845
10.48	.0645	.0591	.0700	2111.9834	1620.7030	2603.2638
11.29	.0695	.0636	.0754	2194.1727	1700.4281	2687.9172
12.10	.0745	.0682	.0808	2267.3524	1766.0181	2768.6867
12.90	.0794	.0727	.0862	2332.6759	1817.9741	2847.3777
13.71	.0844	.0773	.0915	2391.3051	1873.9083	2908.7020
14.52	.0894	.0818	.0969	2445.1623	1932.8522	2958.2724
15.32	.0943	.0864	.1023	2495.1170	1983.4714	3006.7626
16.13	.0993	.0909	.1077	2539.3426	2032.2972	3046.3880
16.94	.1043	.0955	.1131	2577.5675	2078.8729	3076.2620
17.74	.1092	.1000	.1185	2613.2377	2122.4005	3104.0750
18.55	.1142	.1045	.1238	2645.6056	2165.5149	3125.6963
19.35	.1192	.1091	.1292	2673.7763	2205.3074	3142.2453
20.16	.1241	.1136	.1346	2699.8305	2236.7407	3162.9203
20.97	.1291	.1182	.1400	2722.2712	2265.7028	3178.8398
21.77	.1341	.1227	.1454	2740.8897	2299.9138	3181.8657
22.58	.1390	.1273	.1508	2758.4704	2328.6655	3188.2754
23.39	.1440	.1318	.1562	2774.7448	2352.4293	3197.0602
24.19	.1490	.1364	.1615	2788.7643	2373.2424	3204.2862
25.00	.1539	.1409	.1669	2800.5063	2394.6269	3206.3857
25.81	.1589	.1455	.1723	2811.7792	2415.0017	3208.5566
26.61	.1638	.1500	.1777	2821.9678	2429.0447	3214.8910
27.42	.1688	.1545	.1831	2830.4479	2444.2039	3216.6918
28.23	.1738	.1591	.1885	2837.2719	2456.9333	3217.6106
29.03	.1787	.1636	.1938	2842.8841	2466.1893	3219.5789
29.84	.1837	.1682	.1992	2849.0198	2475.7554	3222.2841
30.65	.1887	.1727	.2046	2854.2417	2486.5305	3221.9529
31.45	.1936	.1773	.2100	2858.9887	2494.0985	3223.8790
32.26	.1986	.1818	.2154	2863.4712	2497.6341	3229.3084
33.06	.2036	.1864	.2208	2866.8565	2503.7991	3229.9140
33.87	.2085	.1909	.2261	2869.7965	2512.1626	3227.4305
34.68	.2135	.1955	.2315	2871.5731	2518.4520	3224.6942
35.48	.2185	.2000	.2369	2873.1184	2522.2716	3223.9653
36.29	.2234	.2045	.2423	2873.8475	2524.1828	3223.5122
37.10	.2284	.2091	.2477	2874.6448	2525.6334	3223.6563
37.90	.2334	.2136	.2531	2875.0168	2526.4722	3223.5613
38.71	.2383	.2182	.2585	2874.4876	2524.0527	3224.9225
39.52	.2433	.2227	.2638	2874.7769	2524.4575	3225.0962
40.32	.2483	.2273	.2692	2874.8362	2524.5221	3225.1503
41.13	.2532	.2318	.2746	2874.4308	2526.1292	3222.7324
41.94	.2582	.2364	.2800	2873.1526	2526.2619	3220.0433
42.74	.2631	.2409	.2854	2871.0146	2525.0269	3217.0023

TABLE VIII. - TRANSFORMED TIME-AND-THRUST STATISTICAL DATA OUTPUT FORMAT FOR THE SOLID PROPELLANT ROCKET MOTOR PERFORMANCE VERSUS TIME COMPUTER PROGRAM - Continued

43.55	.2681	.2455	.2908	2868.3248	2523.1066	3213.5431
44.35	.2731	.2500	.2961	2865.2769	2520.5601	3209.9938
45.16	.2780	.2545	.3015	2862.4629	2517.4233	3207.5026
45.97	.2830	.2591	.3069	2859.7900	2513.0028	3206.5773
46.77	.2880	.2636	.3123	2856.6862	2510.6798	3202.6925
47.58	.2929	.2682	.3177	2854.2220	2508.1865	3200.2576
48.39	.2979	.2727	.3231	2851.2232	2504.6908	3197.7557
49.19	.3029	.2773	.3285	2848.1022	2502.6069	3193.5976
50.00	.3078	.2818	.3338	2844.7198	2498.3733	3191.0663
50.81	.3128	.2864	.3392	2839.4213	2494.3550	3184.4877
51.61	.3178	.2909	.3446	2834.2778	2491.1850	3177.3707
52.42	.3227	.2955	.3500	2829.4647	2487.6159	3171.3136
53.23	.3277	.3000	.3554	2824.1272	2482.0980	3166.1563
54.03	.3327	.3045	.3608	2818.4720	2475.7414	3161.2025
54.84	.3376	.3091	.3661	2812.7021	2470.3959	3155.0082
55.65	.3426	.3136	.3715	2807.0796	2464.7331	3149.4260
56.45	.3476	.3182	.3769	2799.8079	2456.7768	3142.8389
57.26	.3525	.3227	.3823	2791.6342	2444.7939	3138.4745
58.06	.3575	.3273	.3877	2783.1130	2433.6220	3132.6041
58.87	.3624	.3318	.3931	2773.3599	2425.3489	3121.3707
59.68	.3674	.3364	.3985	2762.9403	2413.6415	3112.2390
60.48	.3724	.3409	.4038	2751.2329	2399.6761	3102.7897
61.29	.3773	.3455	.4092	2740.2078	2391.2279	3089.1877
62.10	.3823	.3500	.4146	2728.8394	2382.7666	3074.9121
62.90	.3873	.3545	.4200	2715.7695	2372.2278	3059.3112
63.71	.3922	.3591	.4254	2702.5398	2357.6389	3047.4407
64.52	.3972	.3636	.4308	2688.8404	2343.0470	3034.6339
65.32	.4022	.3682	.4361	2675.3877	2330.1481	3020.6273
66.13	.4071	.3727	.4415	2662.1713	2319.3293	3005.0132
66.94	.4121	.3773	.4469	2649.3055	2310.2878	2988.3231
67.74	.4171	.3818	.4523	2637.0740	2309.5849	2964.5630
68.55	.4220	.3864	.4577	2623.4088	2303.8965	2942.9211
69.35	.4270	.3909	.4631	2608.4831	2289.6410	2927.3251
70.16	.4320	.3955	.4685	2593.8872	2280.3134	2907.4610
70.97	.4369	.4000	.4738	2577.2821	2259.7080	2894.8563
71.77	.4419	.4046	.4792	2561.1299	2240.5366	2881.7232
72.58	.4469	.4091	.4846	2545.1835	2220.8183	2869.5489
73.39	.4518	.4136	.4900	2529.2288	2200.8713	2857.5862
74.19	.4568	.4182	.4954	2514.2934	2190.9509	2837.6359
75.00	.4617	.4227	.5008	2499.9293	2180.8189	2819.0397
75.81	.4667	.4273	.5061	2483.7794	2163.4445	2804.1142
76.61	.4717	.4318	.5115	2467.0346	2144.9338	2789.1355
77.42	.4766	.4364	.5169	2450.3193	2127.9415	2772.6971
78.23	.4816	.4409	.5223	2435.9662	2118.0287	2753.9037
79.03	.4866	.4455	.5277	2421.6738	2106.7662	2736.5813
79.84	.4915	.4500	.5331	2406.6572	2094.8505	2718.4640
80.65	.4965	.4546	.5384	2391.4948	2084.5418	2698.4479
81.45	.5015	.4591	.5438	2376.3506	2072.0121	2680.6892
82.26	.5064	.4636	.5492	2359.9619	2058.2972	2661.6267
83.06	.5114	.4682	.5546	2343.4240	2046.6704	2640.1778
83.87	.5164	.4727	.5600	2328.0088	2034.0621	2621.9555
84.68	.5213	.4773	.5654	2312.4452	2020.0016	2604.8888
85.48	.5263	.4818	.5708	2296.8338	2006.6184	2587.0491
86.29	.5313	.4864	.5761	2280.9606	1994.8274	2567.0937
87.10	.5362	.4909	.5815	2265.0679	1981.7576	2548.3783
87.90	.5412	.4955	.5869	2249.1552	1965.7215	2532.5889
88.71	.5462	.5000	.5923	2232.3401	1949.2994	2515.3807
89.52	.5511	.5046	.5977	2215.3879	1934.6026	2496.1732

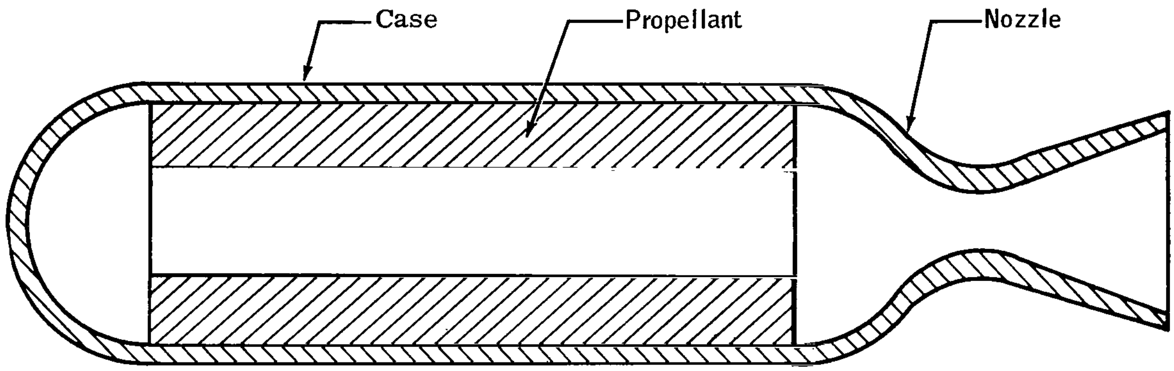
TABLE VIII. - TRANSFORMED TIME-AND-THRUST STATISTICAL DATA OUTPUT FORMAT FOR THE SOLID PROPELLANT ROCKET MOTOR PERFORMANCE VERSUS TIME COMPUTER PROGRAM - Continued

90.32	.5561	.5091	.6031	2198.1009	1919.2804	2476.9214
91.13	.5610	.5136	.6084	2181.0201	1901.8137	2460.2263
91.94	.5660	.5182	.6138	2163.9528	1883.1708	2444.7348
92.74	.5710	.5227	.6192	2145.8561	1865.0740	2426.6382
93.55	.5759	.5273	.6246	2127.6139	1847.1024	2408.1255
94.35	.5809	.5318	.6300	2108.6294	1827.6688	2389.5900
95.16	.5859	.5364	.6354	2089.5475	1807.7857	2371.3093
95.97	.5908	.5409	.6408	2070.9862	1787.6872	2354.2851
96.77	.5958	.5455	.6461	2052.5068	1769.6109	2335.4027
97.58	.6008	.5500	.6515	2033.8807	1750.0885	2317.6729
98.39	.6057	.5546	.6569	2010.3788	1718.7040	2302.0537
99.19	.6107	.5591	.6623	1983.3887	1682.3773	2284.3999
100.00	.6157	.5636	.6677	1947.5539	1638.6585	2256.4492
PCT. TAILOFF TIME						
1.33	.6206	.5686	.6725	1904.0982	1589.7141	2218.4823
2.67	.6254	.5736	.6773	1852.2381	1536.7176	2167.7586
4.00	.6303	.5785	.6821	1793.0064	1474.8251	2111.1877
5.33	.6352	.5835	.6870	1727.7396	1415.1121	2040.3671
6.67	.6401	.5883	.6919	1659.3158	1359.2429	1959.3887
8.00	.6450	.5932	.6968	1591.0564	1300.3271	1881.7856
9.33	.6499	.5980	.7017	1523.3241	1242.0927	1804.5556
10.67	.6548	.6029	.7067	1459.6777	1187.9408	1731.4146
12.00	.6597	.6077	.7117	1399.1327	1141.0040	1657.2614
13.33	.6646	.6124	.7167	1341.8144	1097.6074	1586.0214
14.67	.6694	.6171	.7217	1288.3407	1049.1997	1527.4816
16.00	.6743	.6219	.7268	1236.0442	1005.2416	1466.8468
17.33	.6792	.6265	.7319	1186.3722	964.8985	1407.8460
18.67	.6841	.6312	.7370	1139.1027	927.2142	1350.9913
20.00	.6890	.6359	.7422	1095.0792	892.5298	1297.6286
21.33	.6939	.6405	.7473	1053.0400	856.7447	1249.3352
22.67	.6988	.6451	.7525	1013.0901	824.4401	1201.7401
24.00	.7037	.6496	.7577	974.8540	796.1760	1153.5320
25.33	.7086	.6542	.7629	938.6829	768.1775	1109.1883
26.67	.7134	.6587	.7682	904.5105	739.2651	1069.7559
28.00	.7183	.6632	.7735	872.0610	711.4493	1032.6727
29.33	.7232	.6677	.7788	840.9072	686.0063	995.8081
30.67	.7281	.6721	.7841	810.0386	659.9553	960.1218
32.00	.7330	.6766	.7894	780.9014	634.7568	927.0461
33.33	.7379	.6810	.7948	753.9982	611.2459	896.7505
34.67	.7428	.6854	.8002	728.2765	589.3771	867.1760
36.00	.7477	.6898	.8056	704.2560	570.6066	837.9053
37.33	.7526	.6942	.8110	681.1219	550.2386	812.0052
38.67	.7575	.6985	.8164	658.7540	528.6794	788.8285
40.00	.7623	.7028	.8219	637.8936	510.2441	765.5431
41.33	.7672	.7071	.8273	617.7568	493.5347	741.9790
42.67	.7721	.7114	.8328	598.1981	476.5672	719.8290
44.00	.7770	.7157	.8383	580.0054	460.5784	699.4325
45.33	.7819	.7200	.8438	561.9808	444.6524	679.3093
46.67	.7868	.7242	.8494	544.4940	430.1807	658.8073
48.00	.7917	.7285	.8549	527.5411	416.6257	638.4565
49.33	.7966	.7327	.8605	511.1165	402.8734	619.3596
50.67	.8015	.7369	.8660	495.4716	387.9626	602.9807
52.00	.8063	.7411	.8716	480.7791	375.7061	585.8521
53.33	.8112	.7453	.8772	465.7836	363.2112	568.3561
54.67	.8161	.7494	.8828	451.2083	348.3974	554.0193
56.00	.8210	.7536	.8884	438.0007	336.8681	539.1334

TABLE VIII. - TRANSFORMED TIME-AND-THRUST STATISTICAL DATA OUTPUT FORMAT FOR THE SOLID PROPELLANT ROCKET MOTOR PERFORMANCE VERSUS TIME COMPUTER PROGRAM - Concluded

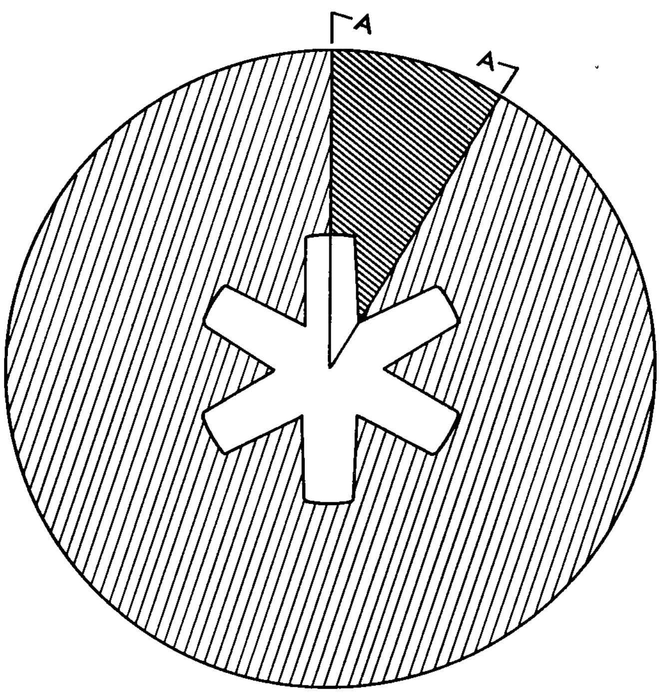
57.33	.8259	.7577	.8941	424.4465	324.1052	524.7877
58.67	.8308	.7619	.8997	412.4682	314.0708	510.8655
60.00	.8357	.7660	.9054	401.7756	307.3071	496.2441
61.33	.8406	.7701	.9110	390.5586	298.3997	482.7176
62.67	.8455	.7742	.9167	380.7095	290.9875	470.4315
64.00	.8503	.7783	.9224	370.5793	279.2780	461.8806
65.33	.8552	.7824	.9281	359.7040	265.4061	454.0018
66.67	.8601	.7865	.9338	348.3283	253.0975	443.5590
68.00	.8650	.7905	.9395	337.3324	242.0264	432.6385
69.33	.8699	.7946	.9452	328.3641	234.3927	422.3355
70.67	.8748	.7986	.9510	319.0463	227.0833	411.0094
72.00	.8797	.8027	.9567	309.4565	219.1186	399.7944
73.33	.8846	.8067	.9625	300.9247	211.2617	390.5877
74.67	.8895	.8107	.9682	292.8332	204.5929	381.0734
76.00	.8944	.8147	.9740	284.5126	196.5410	372.4843
77.33	.8992	.8187	.9797	276.4818	188.9881	363.9755
78.67	.9041	.8227	.9855	269.1266	182.8596	355.3936
80.00	.9090	.8267	.9913	262.2100	177.3512	347.0687
81.33	.9139	.8307	.9971	255.5590	173.4801	337.6380
82.67	.9188	.8347	1.0029	248.8581	166.6553	331.0609
84.00	.9237	.8387	1.0087	241.7264	159.6450	323.8078
85.33	.9286	.8427	1.0145	235.0206	155.6245	314.4167
86.67	.9335	.8466	1.0203	228.3924	150.5706	306.2142
88.00	.9384	.8506	1.0261	222.2223	146.2521	298.1924
89.33	.9432	.8545	1.0320	216.5944	141.0506	292.1381
90.67	.9481	.8585	1.0378	211.6711	136.8468	286.4954
92.00	.9530	.8624	1.0436	206.2429	131.1978	281.2881
93.33	.9579	.8664	1.0495	200.4277	123.9730	276.8824
94.67	.9628	.8703	1.0553	195.1405	117.3585	272.9225
96.00	.9677	.8742	1.0611	190.1864	113.0641	267.3087
97.33	.9726	.8782	1.0670	185.5576	110.6363	260.4790
98.67	.9775	.8821	1.0728	180.3647	108.3940	252.3354
100.00	.9824	.8860	1.0787	175.2670	105.3755	245.1585

NASA-S-66-2023 FEB 24



(a) Longitudinal cross section.

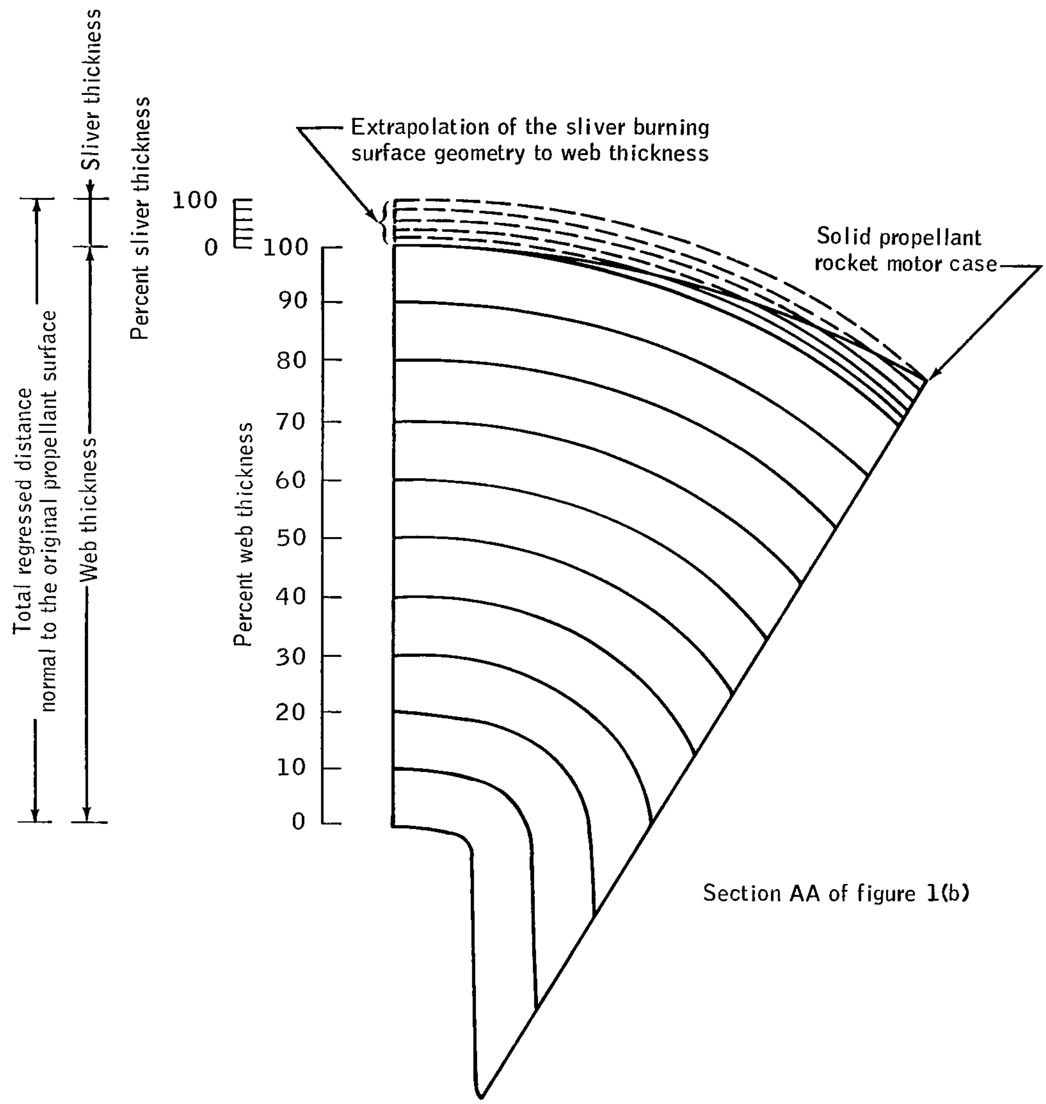
NASA-S-66-2025 FEB 24



(b) Typical internal-burning six-point star propellant grain configuration.

Figure 1. - Typical solid propellant rocket motor.

NASA-S-66-2022 FEB 24



(c) **Burning surface geometry of a typical internal-burning six-point star propellant grain configuration versus regressed distances normal to the original propellant surface.**

Figure 1. - Concluded.

NASA-S-66-2033 FEB 24

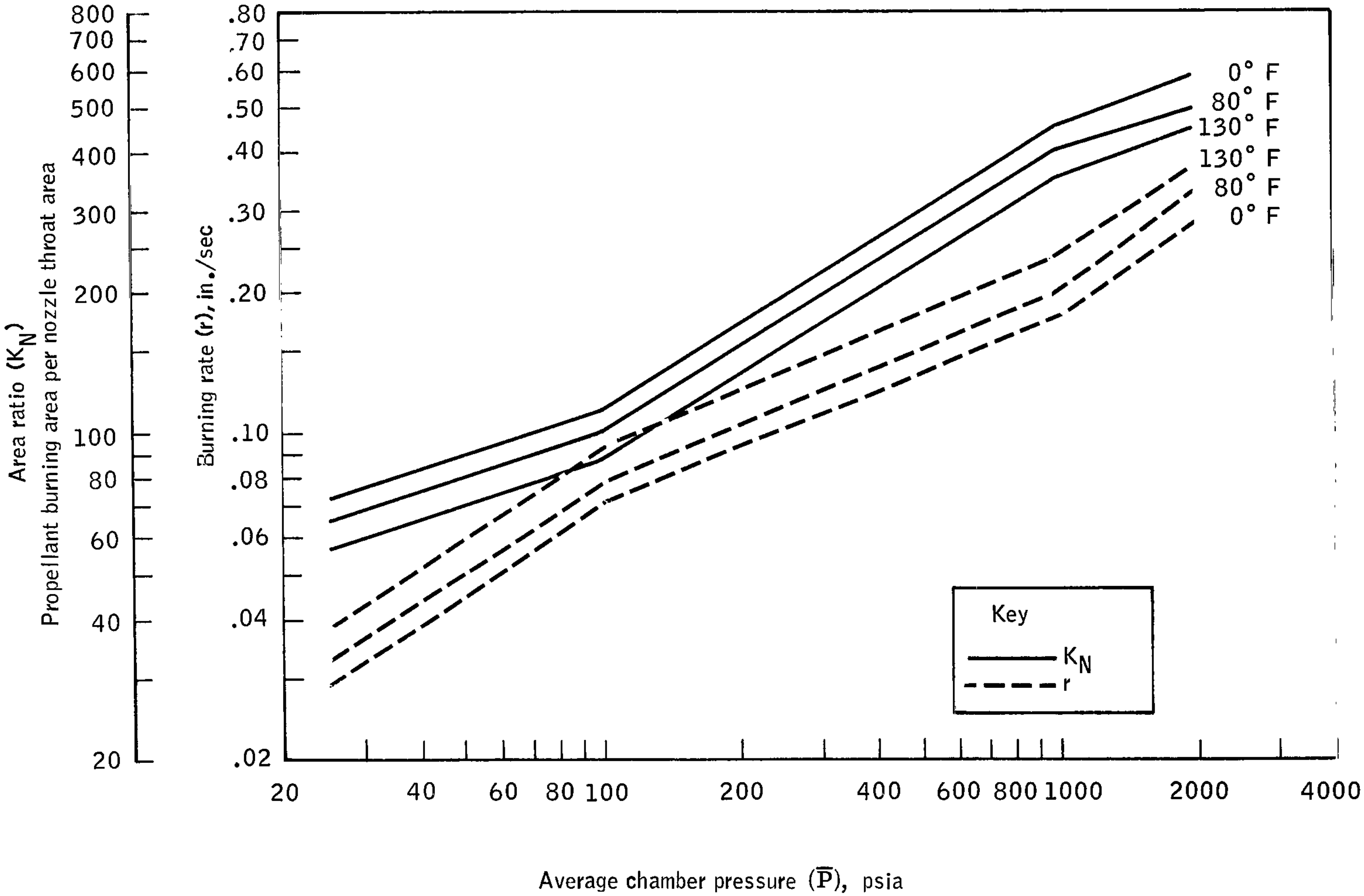


Figure 2. - Typical variation of propellant performance characteristics over extended burning rate and chamber pressure ranges.

NASA-S-66-2032 FEB 24

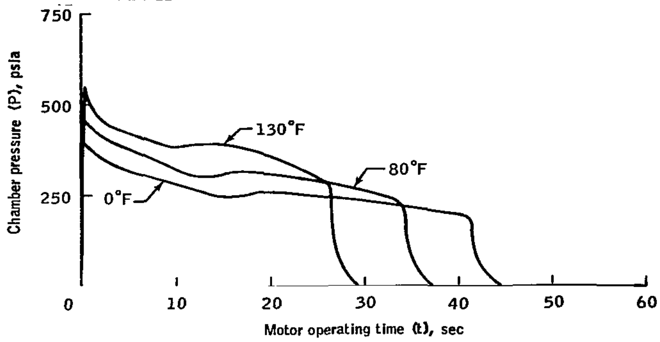


Figure 3. - Typical performance variation of a solid propellant rocket motor of fixed geometry and given propellant that is tested at differing prefire propellant temperatures.

NASA-S-66-2036 FEB 24

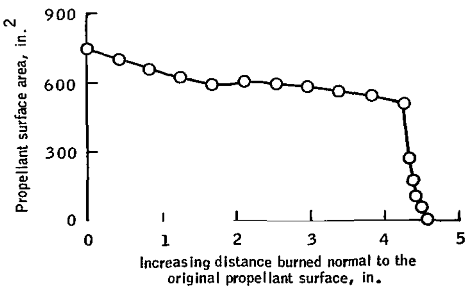


Figure 4. - Propellant burning surface area versus increasing distance burned normal to the original propellant surface for the six-point star propellant grain configuration shown in figures 1(b) and 1(c).

NASA-S-66-2037 FEB 24

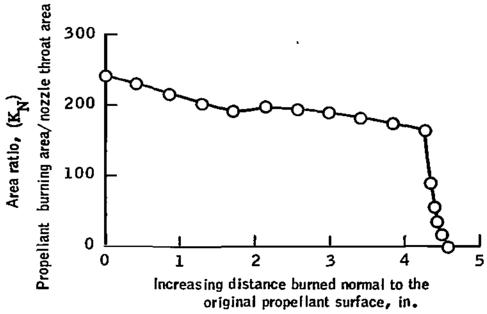


Figure 5. - Area ratio of propellant burning surface area to nozzle throat area versus increasing distance burned normal to the original propellant surface for the solid propellant rocket motor shown in figure 1.

NASA-S-66-2027 FEB 24

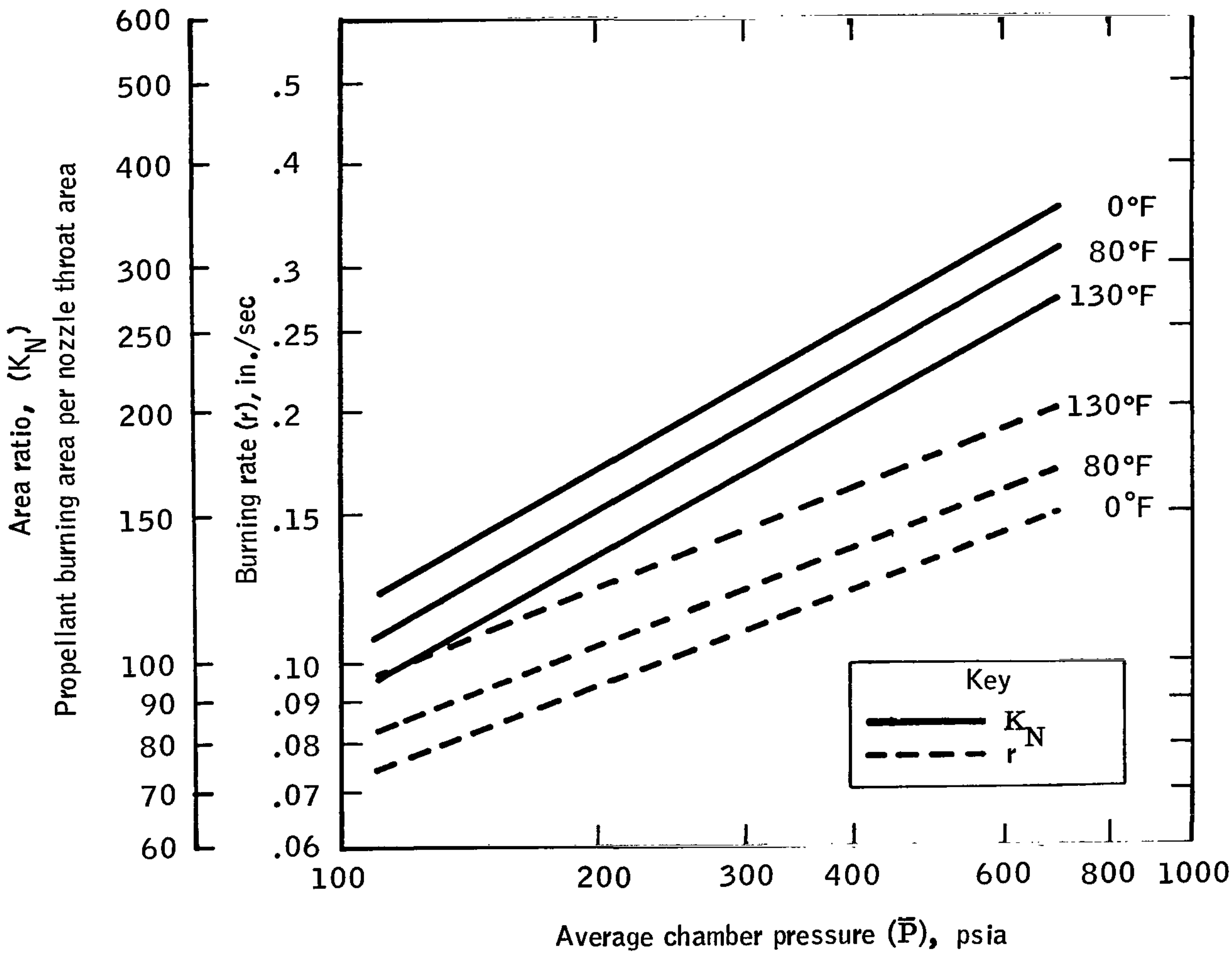


Figure 6. - Typical variation of propellant performance characteristics over limited burning rate and chamber pressure ranges.

NASA-S-66-2028 FEB 24

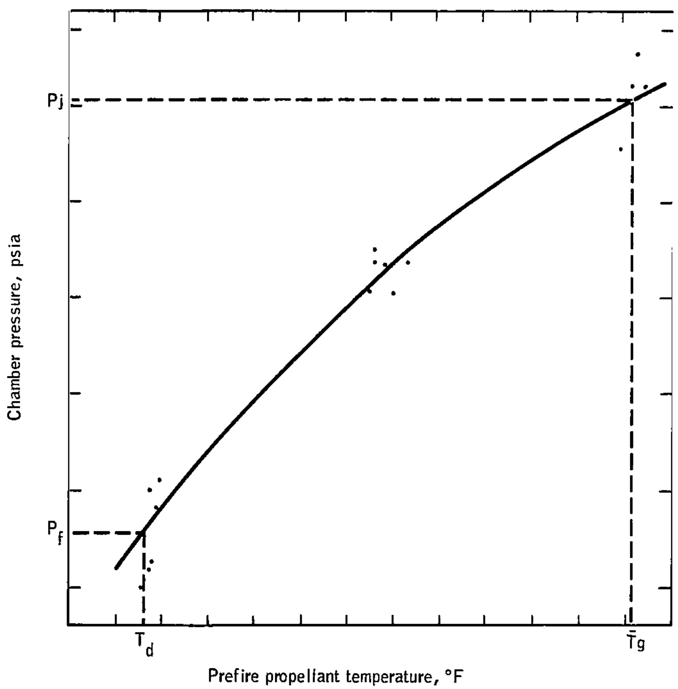


Figure 7. - Second-order, least-squares curve fit of experimentally determined chamber pressures at the individual regressed distance of interest normal to the original propellant surface versus prefire propellant temperature.

NASA-S-66-2031 FEB 24

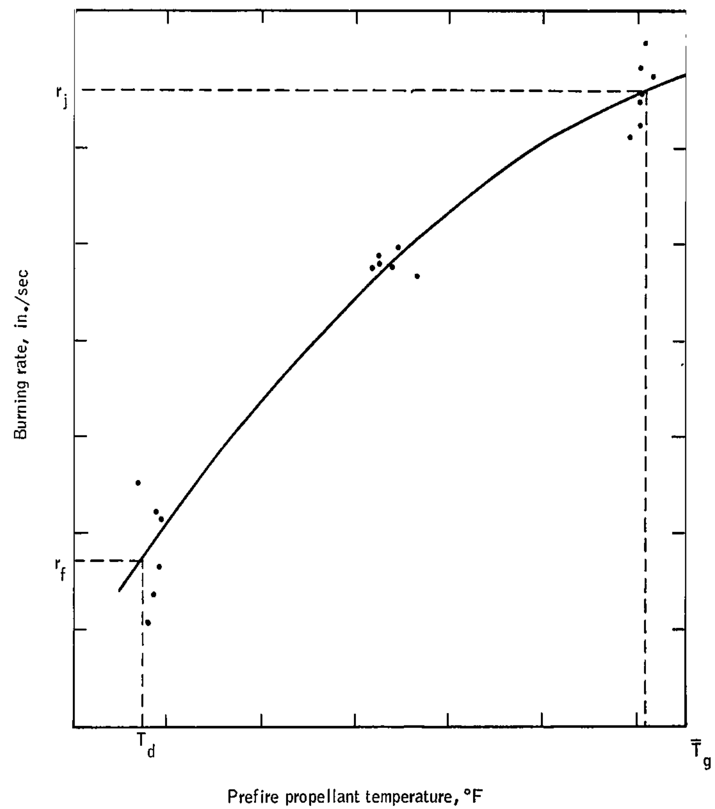


Figure 8. - Second-order, least-squares curve fit of experimentally determined burning rates at the individual regressed distance of interest normal to the original propellant surface versus prefire propellant temperature.

NASA-S-66-2030 FEB 24

NASA-S-66-2026 FEB 24

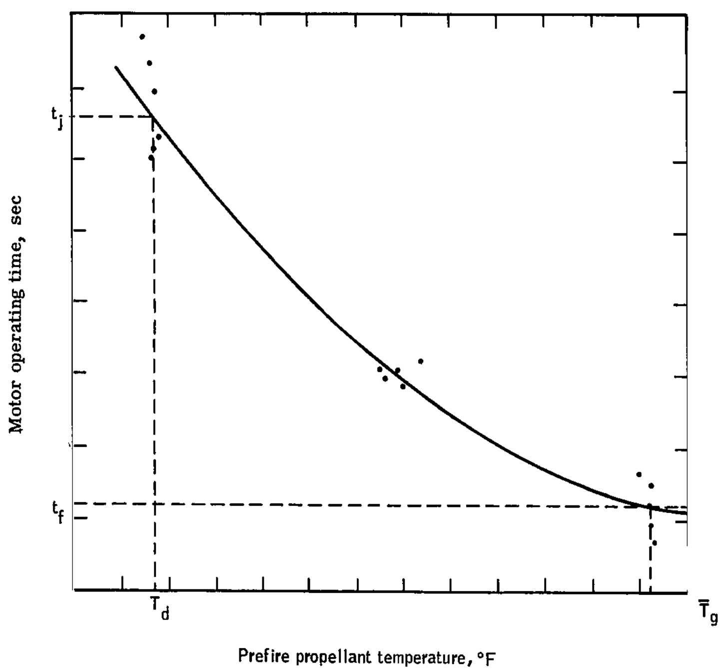


Figure 9. - Second-order, least-squares curve fit of experimentally determined motor operation times to the individual regressed distance of interest normal to the original propellant surface versus prefire propellant temperature.

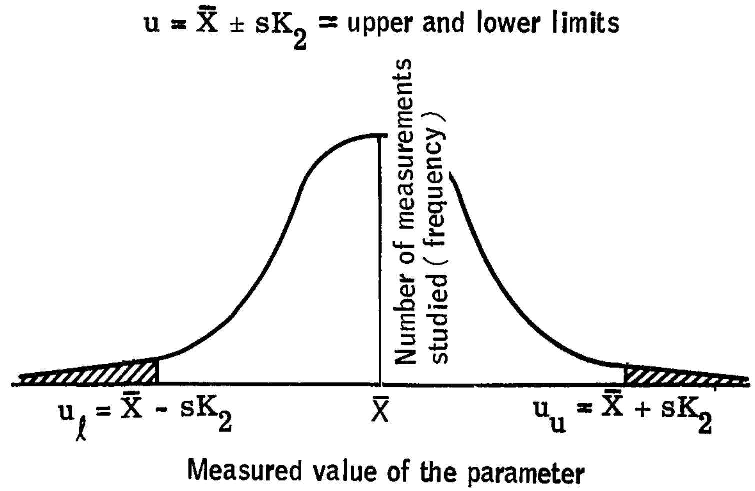
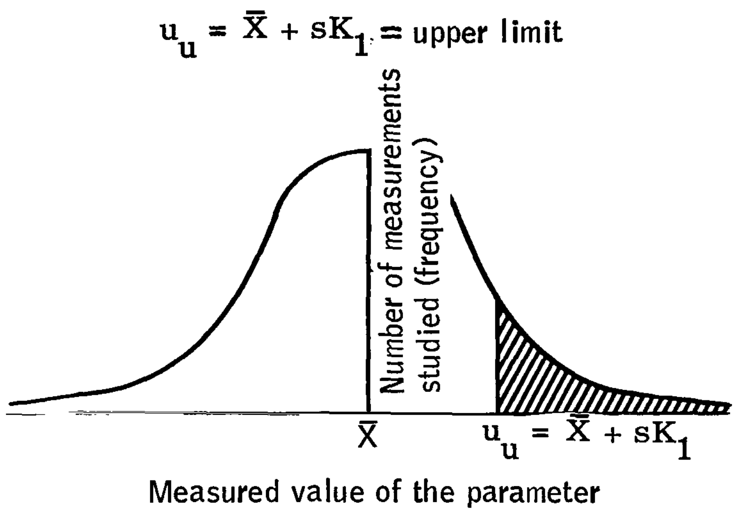
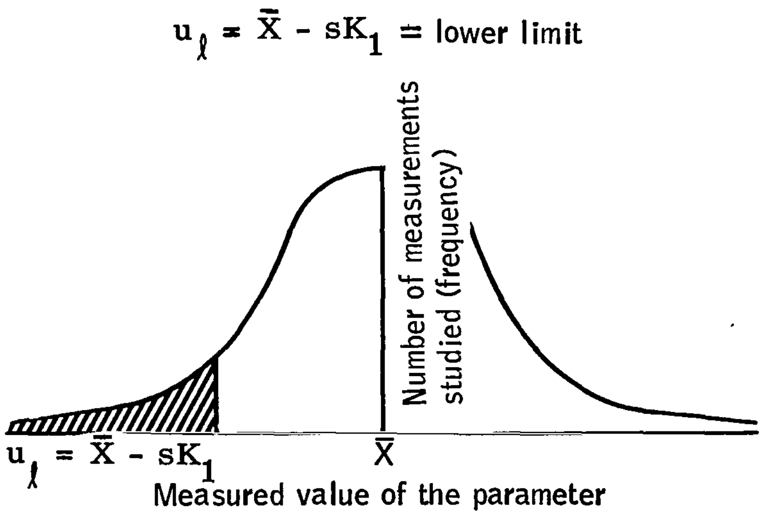


Figure 10. - Example of two-sided tolerance limits for a normal distribution.

NASA-S-66-2029 FEB 24



(a) Upper tolerance limit



(b) Lower tolerance limit

Figure 11. - Example of one-sided tolerance limits for a normal distribution.

NASA-S-66-2469 MAR 16

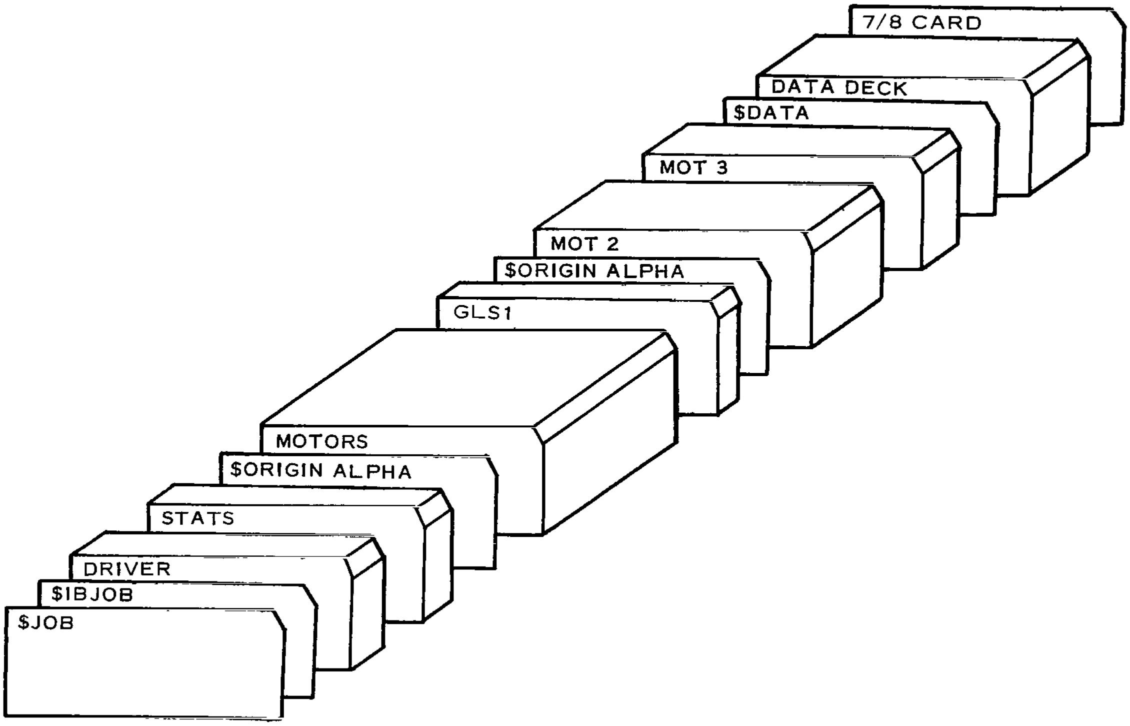


Figure 12. - IBM 7094 deck setup for the general solid propellant rocket motor performance computer program using the group transformation method.

NASA-S-66-2035 FEB 24

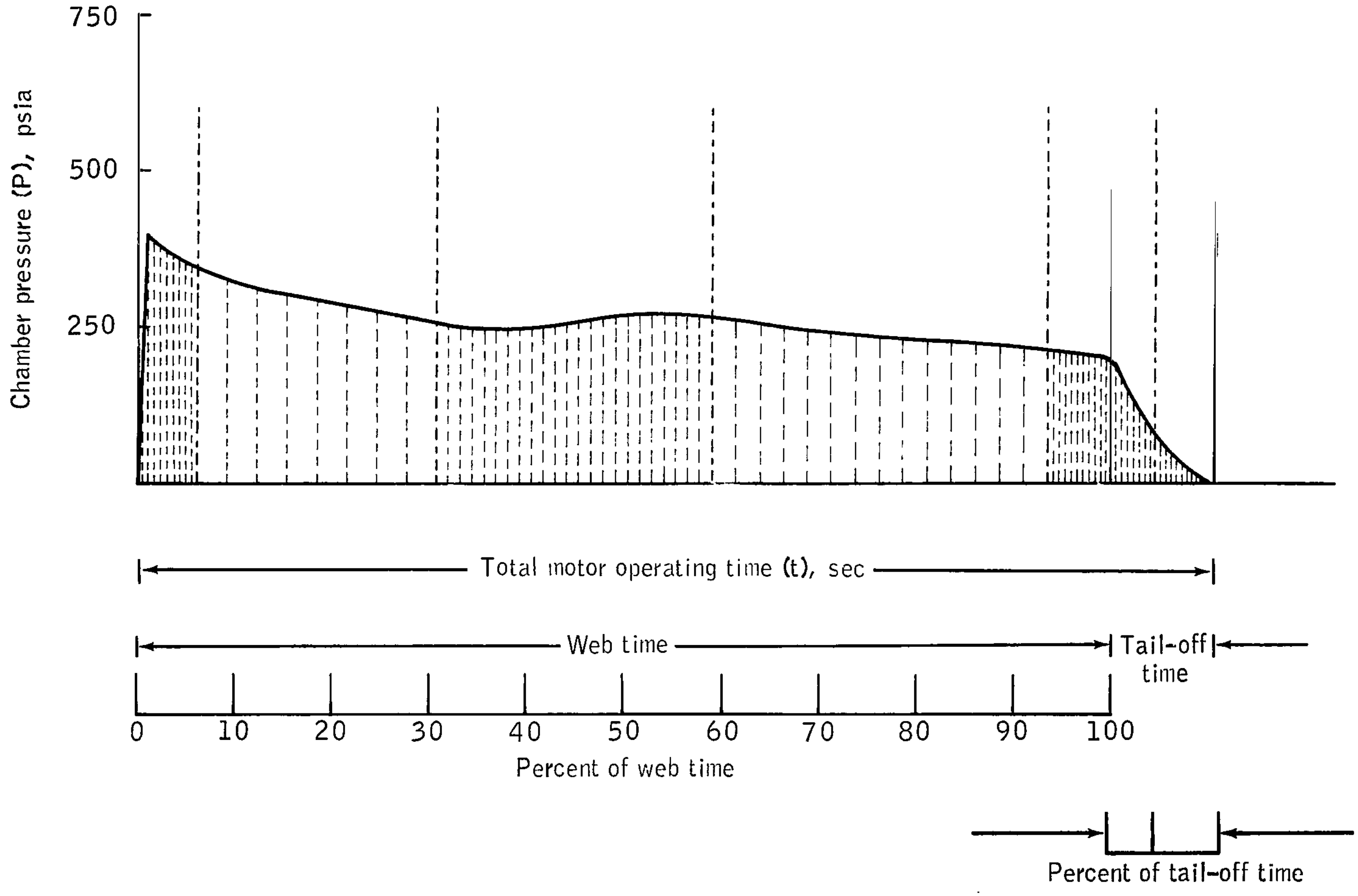


Figure 13. - Example of choosing the output points that will best define the variation of chamber pressure and thrust versus time.

NASA-S-66-2470 MAR 16

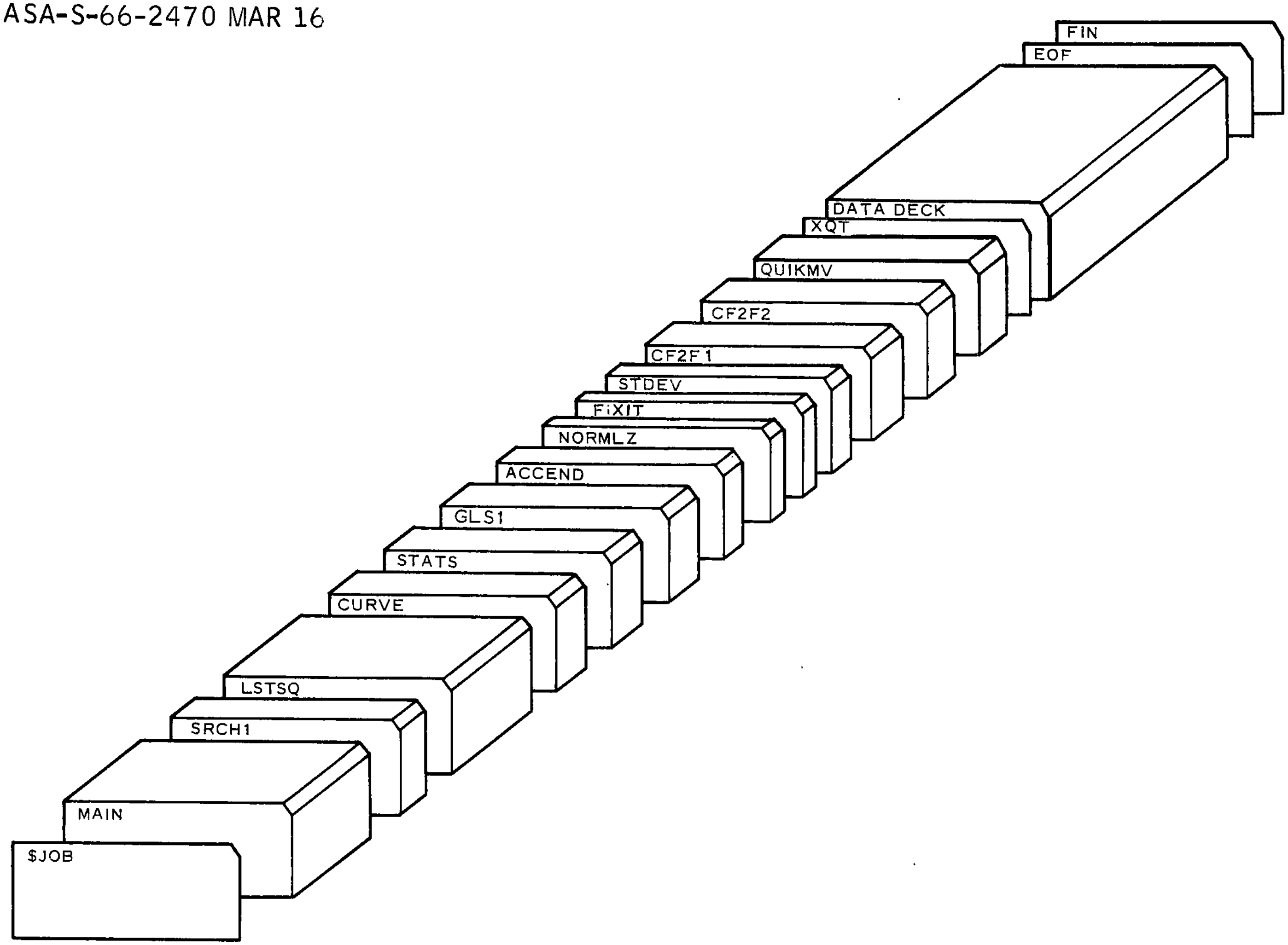


Figure 14. - Univac 1107 deck setup for the solid propellant rocket motor performance versus time computer program using the group transformation method.

NASA-S-66-2021 FEB 24

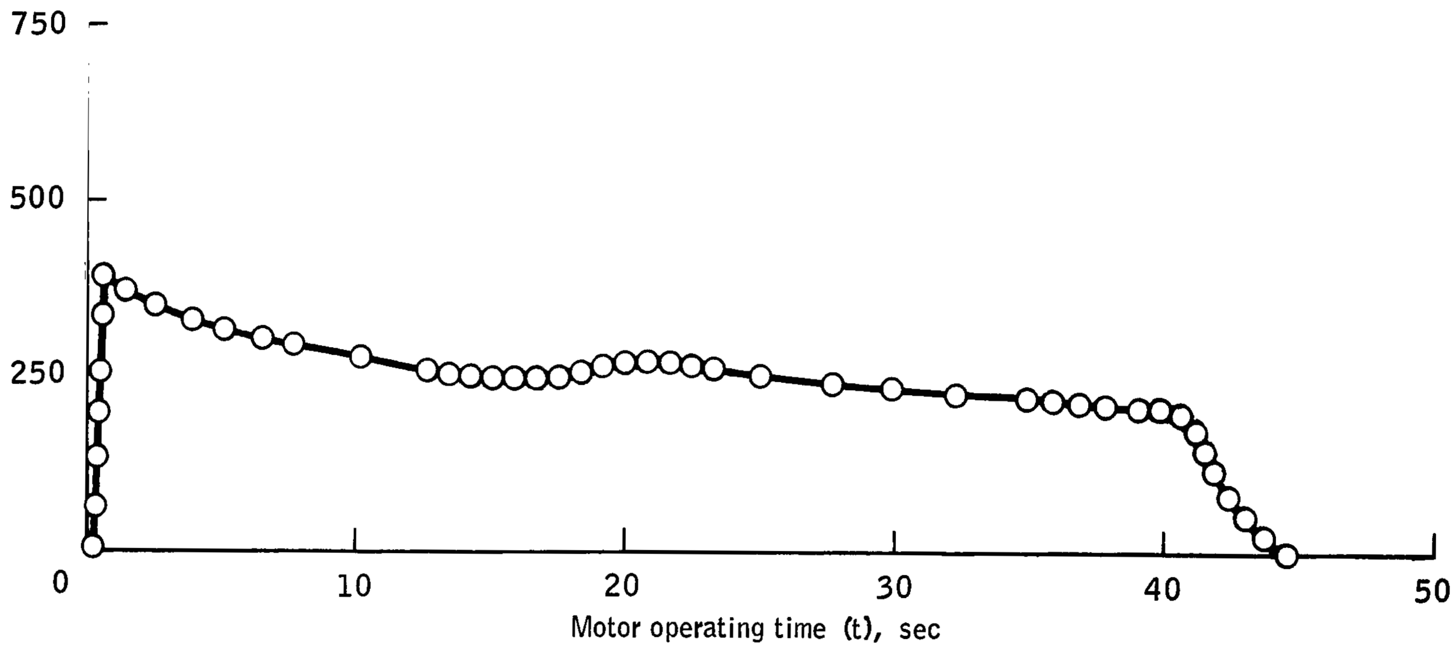
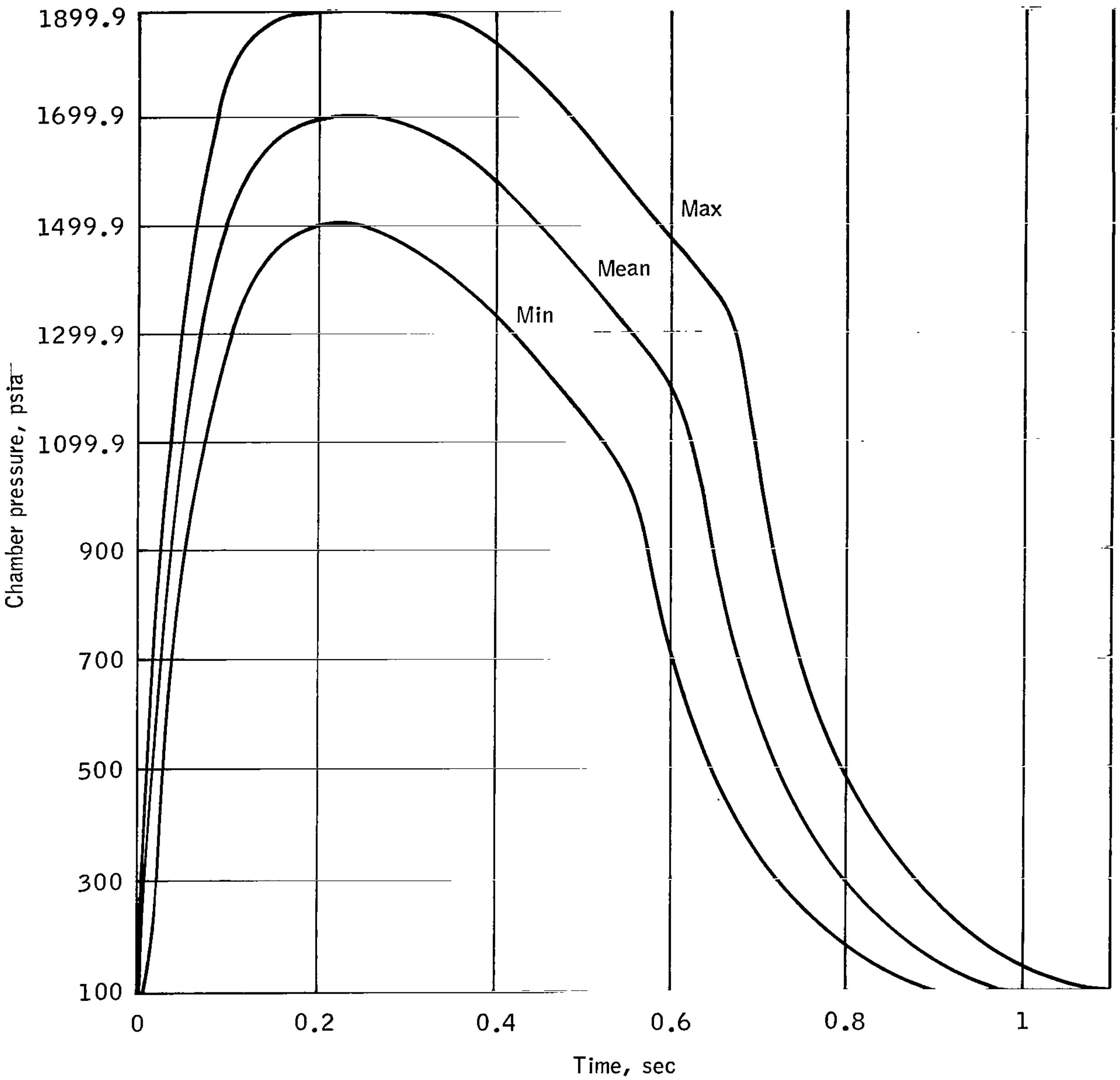
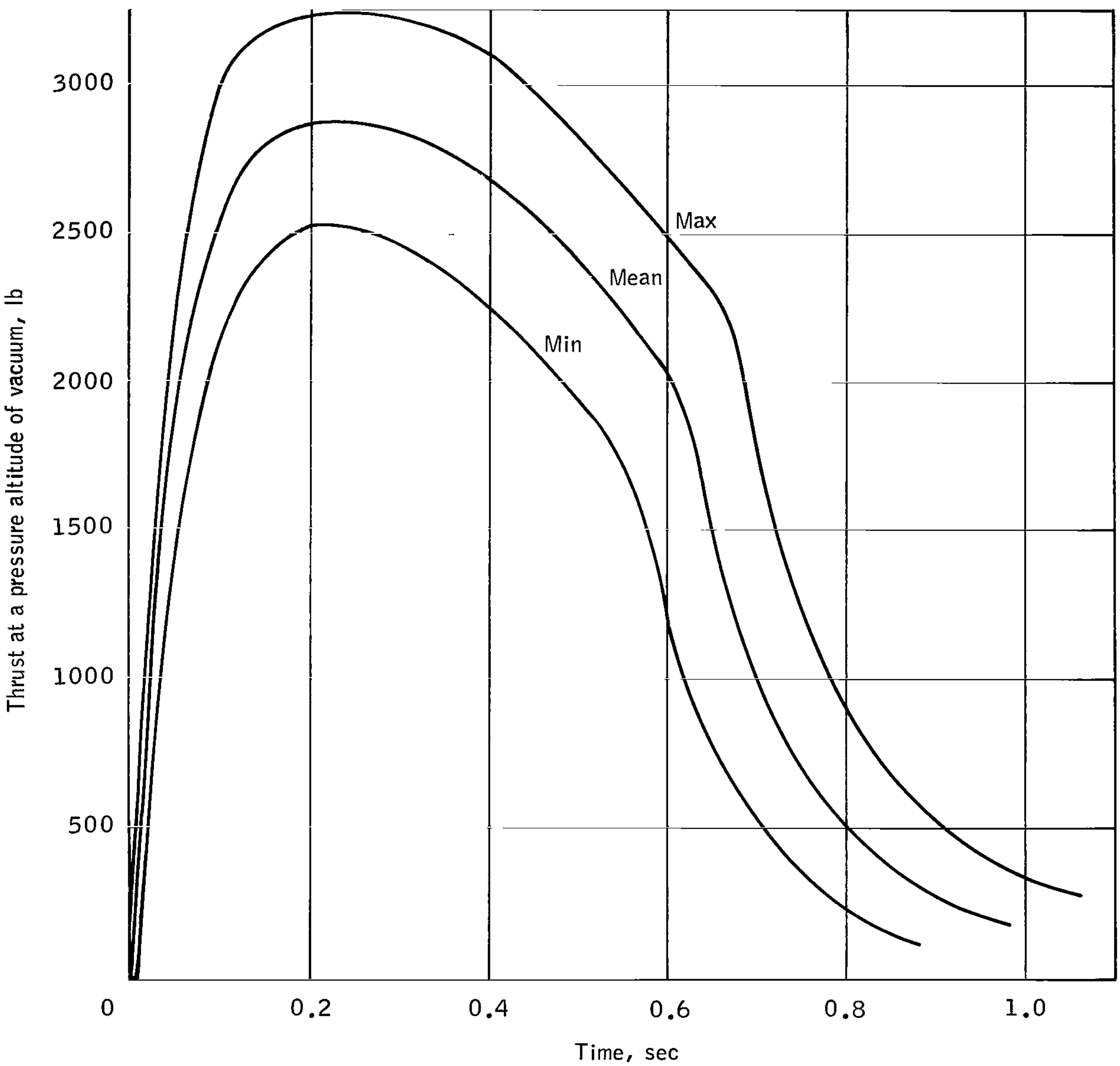


Figure 15. - Example of choosing the input points that will best define the variation of chamber pressure and thrust versus time.



(a) Typical transformed time and chamber pressure graphed outputs.

Figure 16. - Example of graphical format from the solid propellant rocket motor performance versus time computer program.



(b) Typical transformed time and thrust graphed outputs.

Figure 16. - Concluded.

REFERENCES

1. Huggett, Clayton; Bartley, C. E.; and Mills, Mark M.: Solid Propellant Rockets. Princeton Univ. Press Co., 1960.

2. Barrère, Marcel; Jaumotte, André; De Veubeke, Baudouin Fraeijs; and Vandenkerckhove, Jean: Rocket Propulsion. Elsevier Publishing Co., N. Y., 1960.

3. Crow, Edwin L.; Davis, Frances A.; and Maxfield, Margaret W.: Statistics Manual. Dover Publications, Inc., New York, 1955.

4. "Design Data for Pressurized Gas Systems (U)." Stanford Research Institute, Menlo Park, California. Nov. 1963.

5. Eisenhart, Churchill; Hastay, Millard W.; and Wallis, W. Allen: Techniques of Statistical Analysis. McGraw-Hill Book Co., Inc., 1947.

6. Bowker, Albert H.; and Lieberman, Gerald J.: Engineering Statistics. Prentice-Hall, Inc., 1959.

APPENDIX A

LISTING OF THE GENERAL SOLID PROPELLANT ROCKET MOTOR PERFORMANCE COMPUTER PROGRAM USING THE GROUP TRANSFORMATION METHOD

```
$IBJOB DRIVER  GO,SOURCE
$IBFTC DRIVER
C
C                 DRIVER PROGRAM FOR MOTORS
C
      DIMENSION XX(4860)
      DIMENSION EXPA(4), COUNT(4), WP  (4,15), WEB(4,15), ATTOT(4,15),
     1 AETOT(4,15), P(9,4,15), MOTNO(4,15), FVAC1(9,4,15)
      COMMON G, CPHI, NT, EXPA, COUNT, WP, WEB, ATTOT, AETOT,
     1        P, MOTNO, FVAC1, NTIMES, XX, NTOT, NBURN, NFIRST, NP
      COMMON CP1, CP2, CT1, CT2, VEC1, VEC2, VEC3
    1 READ (5,10) CASE
      WRITE (6,11) CASE
      WRITE (6,21)
      READ (5,80)
      WRITE (6,80)
      READ (5,79) NTIMES
      WRITE (6,14) NTIMES
      DO 2    I = 1, NTIMES
      READ (5,80)
    2 WRITE (6,80)
   10 FORMAT (A6)
   11 FORMAT (1H1, A6/)
   14 FORMAT (1X //I2, 33H TIMES WERE USED IN THIS ANALYSIS )
   21 FORMAT ( 46HOGENERAL SOLID-FUEL ROCKET PERFORMANCE PROGRAM /
     1        120H ROCKET TEST FIRINGS CONDUCTED AT SEVERAL TEMPERATURES
     2 ARE EFFECTIVELY TRANSFORMED TO SPECIFIED TEMPERATURES OF INTEREST
     3 /        89H STATISTICAL ANALYSES ARE PERFORMED ON THE TEST FIRING
     4 DATA AFTER IT HAS BEEN TRANSFORMED //)
   79 FORMAT (10X, I2)
   80 FORMAT (80H
     1                             )
      CALL MOTORS
      CALL MOT2
      CALL MOT3
      GO TO 1
      END

$IBFTC STATS
      SUBROUTINE STATS (X, N, XM, SD)
      DIMENSION X(100)
      SUM1 = 0.
      SUM2 = 0.
      XNO = N
      XNO1 = XNO - 1.
      DENOM = XNO*XNO1
      DO 10 I = 1,N
      SUM1 = SUM1 + X(I)
   10 SUM2 = SUM2 + X(I)*X(I)
      XM = SUM1 / XNO
      XNUM = (XNO*SUM2 - SUM1*SUM1)
      IF (XNUM .LE. 0.) GO TO 14
      SD = SQRT(XNUM/DENOM)
      IF ((SD/XM) .LT. .00025) GO TO 14
      GO TO 15
   14 SD = 0.
   15 RETURN
      END
```

```
$ORIGIN      ALPHA
$IBFTC MOTORS
      SUBROUTINE MOTORS
C
C                 SOLID PROPELLANT ROCKET MOTOR TEST PROGRAM - FIRST HALF
C
      DIMENSION ZZ(648), XX(4860)
      COMMON G, CPHI, NT, EXPA, COUNT, WP, WEB, ATTOT, AETOT,
     1       P, MOTNO, FVAC1, NTIMES, XX, NTOT, NBURN, NFIRST, NP
      COMMON CP1, CP2, CT1, CT2, VEC1, VEC2, VEC3
      DIMENSION                          EXTEM( 4), EXPA( 4),  EPS( 4),
     1           COUNT( 4),  SET1(45),  SET2(45), SET3(45), SET4(45),
     2            SET5(45),  SET6(45),  SET7(45), TGRP(3),
     3           TENTRY(60), TENT2(60),  ARRAY(46,4), ENTRY(9,60),
     4          COEF1(9,4), COEF2(9,4), IFLAG1(9), IFLAG2(9)
C
      DIMENSION   X(4), TIME(9,4,15),   INTP(9,4,15),     P(9,4,15),
     1   INTF(9,4,15),       F(9,4,15),     P1(9,4,15),     P2(9,4,15),
     2     P3(9,4,15),      T1(9,4,15),     T2(9,4,15),     T3(9,4,15),
     3 INTPT1(9,4,15), INTPT2(9,4,15), INTPT3(9,4,15)
C
      DIMENSION  AT1(4,15), AT2(4,15), AT3(4,15), AT4(4,15), AE1(4,15),
     1           AE2(4,15), AE3(4,15), AE4(4,15),  WP(4,15), WEB(4,15),
     2          DENS(4,15), PAF(4,15), LNP(9,4), LNT(9,4),
     3         MOTNO(4,15),TEMP(4,15), RU (4,15), KN (4,15)
C
      DIMENSION          MP(9,4),     SDP(9,4),     MF(9,4),    SDF(9,4),
     1  MTIME(9,4), SDTIME(9,4),    MPT1(9,4),  SDPT1(9,4),    MPT2(9,4),
     2  SDPT2(9,4),    MPT3(9,4),  SDPT3(9,4),    MTT1(9,4),   SDTT1(9,4),
     3    MTT2(9,4),  SDTT2(9,4),    MTT3(9,4),  SDTT3(9,4),    MPTT(9,4),
     4  SDPTT(9,4),    MTTT(9,4),  SDTTT(9,4),  MINT1(9,4), SDINT1(9,4),
     5  MINT2(9,4), SDINT2(9,4),  MINT3(9,4), SDINT3(9,4),  MINTT(9,4),
     6 SDINTT(9,4),  MTEMP( 15), SDTEMP( 15)
C
C
      DIMENSION    R1   (4,15), R2   (4,15), R3   (4,15), AFTOT(4,15),
     1 ATTOT(4,15),EA   (4,15), EB   (4,15), EC   (4,15), ED   (4,15),
     2 ETOT (4,15)
     3 ,C1MIN(4), C1MAX(4), C2MIN(4), C2MAX(4), C1TMN(4), C1TMX(4),
     4 C2TMN(4), C2TMX(4)
      DIMENSION C1MN(5), C1MX(5), C2MN(5), C2MX(5)
      DIMENSION
     1 FVAC1(9,4,15), WPU  (9,4,15), ISP   (9,4,15), IVAC1(9,4,15),
     2 ISP1 (9,4,15), I12  (9,4,15), F12   (9,4,15),ISPU12(9,4,15),
     3 I13  (9,4,15), F13  (9,4,15),ISPU13 (9,4,15), I14  (9,4,15),
     4 F14  (9,4,15),ISPU14(9,4,15), CSTU  (9,4,15)
      DIMENSION   MAT(5), SDAT(5), MAE(5), SDAE(5), ME(5), SDE(5)
      DIMENSION       MINTP(9,4), SDINTP(9,4),  MINTF(9,4), SDINTF(9,4),
     1    MWPU(9,4),  SDWPU(9,4),    MISP(9,4),  SDISP(9,4),  MCSTU(9,4),
     2 SDCSTU(9,4),     MRU(  4),    SDRU(  4),     MKN(  4),    SDKN(  4),
     3   MIVC1(9,4), SDIVC1(9,4),  MFVC1(9,4), SDFVC1(9,4),  MIU11(9,4),
     4 SDIU11(9,4),    MI12(9,4),   SDI12(9,4),    MF12(9,4),  SDF12(9,4),
     5   MIU12(9,4), SDIU12(9,4),    MI13(9,4),   SDI13(9,4),    MF13(9,4),
     6   SDF13(9,4),  MIU13(9,4), SDIU13(9,4),    MI14(9,4),   SDI14(9,4),
     7    MF14(9,4),  SDF14(9,4),  MIU14(9,4), SDIU14(9,4), IL(4)
      DIMENSION  SIGP12(9), SIGP22(9), SIGP32(9), SIGP11(9),
     1 SIGP21(9), SIGP31(9), SIGP13(9), SIGP23(9), SIGP33(9),
     2 PIK11(9), PIK21(9), PIK31(9), PIK12(9), PIK22(9), PIK32(9),
     3  PIK13(9), PIK23(9), PIK33(9)
C
      EQUIVALENCE
```

```
      1  (XX(   1),           I12(1), I13(1), I14(1),          P1(1)),
      2  (XX( 541),                                  WPU (1), P2(1)),
      3  (XX(1081),                                  ISP (1), P3(1)),
      4  (XX(1621),           F12(1), F13(1), F14(1),          T1(1)),
      5  (XX(2161),                                  INTF(1), T2(1)),
      6  (XX(2701),                                  INTP(1), T3(1)),
      7  (XX(3241), ISP1(1), ISPU12(1), ISPU13(1), ISPU14(1), INTPT1(1)),
      8  (XX(3781),                                 CSTU(1), INTPT2(1)),
      9  (XX(4321),                                          INTPT3(1))
      EQUIVALENCE
      1  (ZZ(   1), MIVC1(1), MI12(1), MI13(1), MI14(1), MPT1(1)),
      2  (ZZ(  37),                                      MPT2(1)),
      3  (ZZ(  73),                                      MPT3(1)),
      4  (ZZ( 109), MFVC1(1), MF12(1), MF13(1), MF14(1), MTT1(1)),
      5  (ZZ( 145),                                      MTT2(1)),
      6  (ZZ( 181),                                      MTT3(1)),
      7  (ZZ( 217), MIU11(1),MIU12(1),MIU13(1),MIU14(1),MINT1(1)),
      8  (ZZ( 253),                                     MINT2(1)),
      9  (ZZ( 289),                                     MINT3(1))
      EQUIVALENCE
      1  (ZZ( 325), SDIVC1(1), SDI12(1), SDI13(1), SDI14(1), SDPT1(1)),
      2  (ZZ( 361),                                          SDPT2(1)),
      3  (ZZ( 397),                                          SDPT3(1)),
      4  (ZZ( 433), SDFVC1(1), SDF12(1), SDF13(1), SDF14(1), SDTT1(1)),
      5  (ZZ( 469),                                          SDTT2(1)),
      6  (ZZ( 505),                                          SDTT3(1)),
      7  (ZZ( 541), SDIU11(1),SDIU12(1),SDIU13(1),SDIU14(1), SDPTT(1)),
      8  (ZZ( 577),                                          SDTTT(1)),
      9  (ZZ( 613),                                         SDINT3(1))
      INTEGER  COUNT, ENDMOT, TEMSEL
      REAL     INTP   , INTF   , INTPT1, INTPT2, INTPT3, INTPTT,
     1 MTEMP  , MTIME  , SDTIME, MP     , MF     , MPT1   , MPT2   , MPT3   ,
     2 MPTT   , MTT1   , MTT2   , MTT3   , MTTT   , MINT1  , MINT2  , MINT3  ,
     3 MINTT  , LNT    , LNP    , MOTNO  , ISP    , IVAC1  , ISP1   , I12    ,
     4 ISPU12, I13    , ISPU13, I14    , ISPU14, KN     , MINTP  , MINTF  ,
     5 MWPU   , MISP   , MCSTU  , MRU    , MKN    , MIVC1  , MFVC1  , MIU11  ,
     6 MI12   , MF12   , MI13   , MF13   , MIU13  , MI14   , MF14   , MIU14
      REAL  MAT, MAE, ME
C
C                INPUT FORMATS
C
   81 FORMAT (I12, 4F12.4)
   82 FORMAT (2F12.5, 3I2)
   83 FORMAT (4F12.4)
   84 FORMAT (A6, F12.4, I2)
   85 FORMAT (4E12.5)
   86 FORMAT (8F9.4, F8.4)
C
C                OUTPUT FORMATS
C
   16 FORMAT (10H0THERE ARE I2, 46H AMBIENT PRESSURES BEING CONSIDERED,
     1 THEY ARE / 4F12.4)
   18 FORMAT (10H0THERE ARE I2, 46H EXACT TEMPERATURES BEING CONSIDERED,
     1 THEY ARE / 4F12.4)
   20 FORMAT (41H G AND PHI ARE CONSTANT FOR A RUN AND ARE / 2F12.5)
 2001 FORMAT ( 62H1THROAT AREA      ATA        ATB        ATC        ATD
     1AT TOTAL / 10H MOTOR NO.//)
 2002 FORMAT (11H0TEMP GROUP I2 // (1X, A6,5X, 5F10.4))
 2003 FORMAT ( 62H1EXIT AREA        AEA        AEB        AEC        AED
     1AE TOTAL / 10H MOTOR NO.//)
 2004 FORMAT ( 62H1EXPANSION RATIO  EA         EB         EC         ED
     1 E TOTAL / 10H MOTOR NO.//)
```

```
2005 FORMAT (101H1TIMES              T1         T2         T3         T4
   1    T5         T6         T7         T8         T9 / 10H MOTOR NO.)
2006 FORMAT (11H0TEMP GROUP I2 // (1X, A6,5X, 9F10.4))
2007 FORMAT (101H1PRESSURES          PT1        PT2        PT3        PT4
   1    PT5        PT6        PT7        PT8        PT9 / 10H MOTOR NO.)
2008 FORMAT (119H1INTEGRATED       INTPT1       INTPT2       INTPT3       I
    1NTPT4       INTPT5       INTPT6       INTPT7       INTPT8       INTPT9/
    2  9H PRESSURE / 10H MOTOR NO.)
2009 FORMAT (102H1                CSTAR T1  CSTAR T2  CSTAR T3  CSTAR T4
    1CSTAR T5  CSTAR T6  CSTAR T7  CSTAR T8  CSTAR T9 /10H MOTOR NO.)
2010 FORMAT ( 52H1                 AVERAGE R BASED      AVERAGE KN BASED /
    1         48H                  ON BURN TIME         ON BURN TIME /
    2         10H MOTOR NO.)
2012 FORMAT (101H1PROPELLANT WT.  WPT1       WPT2       WPT3       WPT4
    1   WPT5       WPT6       WPT7       WPT8       WPT9 / 10H MOTOR NO.)
2014 FORMAT (1H1, 30X, 13H PA = AMBIENT //)
2015 FORMAT (119H THRUST             FT1          FT2          FT3
    1FT4          FT5          FT6          FT7          FT8          FT9  /
    2 10H MOTOR NO.)
2016 FORMAT (119H TOTAL IMPULSE     IT1          IT2          IT3
    'IT4          IT5          IT6          IT7          IT8          IT9  /
    2 10H MOTOR NO.)
2017 FORMAT (119H SPEC. IMPULSE   ISPT1        ISPT2        ISPT3        I
    1SPT4        ISPT5        ISPT6        ISPT7        ISPT8        ISPT9 /
    2 10H MOTOR NO.)
2024 FORMAT (1H1, 30X,  5H PA = F 9.3)
2030 FORMAT (1H1, 30X, 18H TRANSFORMED TIME / 9H TIME NO.I2)
2031 FORMAT (1H1, 30X, 22H TRANSFORMED PRESSURES / 9H TIME NO.I2)
2040 FORMAT (11H0TEMP GROUP I2 //(1X, A6,18X, 4F10.4))
2041 FORMAT (11H0TEMP GROUP I2 // (1X, A6, 18X, 4F12.2, 10X, 4F12.2))
2042 FORMAT (      30X, 18H PRESSURE INTEGRAL 40X, 17H AVERAGE PRESSURE)
2044 FORMAT (11H0TEMP GROUP I2 // (1X,A6,7X, F10.4, 12X, F10.4))
2045 FORMAT (11H0TEMP GROUP I2 // (1X, A6, 5X, 9F12.2))
2046 FORMAT (11H0TEMP GROUP I2 // (1X, A6,5X, 9F10.2))
2047 FORMAT ( 27X,36H TEMP1     TEMP2     TEMP3     TEMP4    /
    1          10H MOTOR NO.)
2048 FORMAT ( 27X,43H  TEMP1       TEMP2       TEMP3       TEMP4 ,17X,
    1          43H  TEMP1       TEMP2       TEMP3       TEMP4 /
    2          10H MOTOR NO.)
2050 FORMAT (1H1, 30X, 23H TRANSFORMATION FACTORS )
2051 FORMAT ( 7H  PIK11 9F10.6 / 7H  PIK21 9F10.6 / 7H  PIK31 9F10.6 /
    1        7H  PIK12 9F10.6 / 7H  PIK22 9F10.6 / 7H  PIK32 9F10.6 /
    2        7H  PIK13 9F10.6 / 7H  PIK23 9F10.6 / 7H  PIK33 9F10.6//)
2052 FORMAT ( 7H SIGP11 9F10.6 / 7H SIGP21 9F10.6 / 7H SIGP31 9F10.6 /
    1        7H SIGP12 9F10.6 / 7H SIGP22 9F10.6 / 7H SIGP32 9F10.6 /
    2        7H SIGP13 9F10.6 / 7H SIGP23 9F10.6 / 7H SIGP33 9F10.6 )
2055 FORMAT (12H0MEAN          9F10.4 / 12H STAND. DEV    9F10.4)
2060 FORMAT (12H0MEAN          9F10.2 / 12H STAND. DEV.   9F10.2)
2062 FORMAT (34H0CONFIDENCE ON NORMAL DISTRIBUTION /
    1       25H (ONE   MIN              4F10.4 /
    2       25H SIDED  MAX              4F10.4 //
    3       25H (TWO   MIN              4F10.4 /
    4       25H SIDED  MAX              4F10.4 )
2063 FORMAT (14H0MEAN            F10.4, 12X, F10.4 /
    1       14H STANDARD DEV.    F10.4, 12X, F10.4 )
2064 FORMAT (25H0MEAN                   4F10.4 /
    1       25H STANDARD DEV.           4F10.4)
2065 FORMAT (25H0TOTAL MEAN             4F10.4 /
    1       25H TOTAL STANDARD DEV.     4F10.4 )
2066 FORMAT (25H0MEAN                   4F12.2, 10X, 4F12.2 /
    1       25H STANDARD DEV.           4F12.2, 10X, 4F12.2 )
2067 FORMAT (25H0TOTAL MEAN             4F12.2, 10X, 4F12.2 /
```

```
      1          25H TOTAL STANDARD DEV.      4F12.2, 10X, 4F12.2 )
 2068 FORMAT (34H0CONFIDENCE ON NORMAL DISTRIBUTION /
      1          25H (ONE    MIN                  4F12.2,10X,4F12.2 /
      2          25H SIDED) MAX                   4F12.2,10X,4F12.2 //
      3          25H (TWO    MIN                  4F12.2,10X,4F12.2 /
      4          25H SIDED) MAX                   4F12.2,10X,4F12.2 )
 2070 FORMAT (12H0TOTAL MEAN     5F10.4 / 12H TOTAL S.D.     5F10.4)
 2071 FORMAT (34H0CONFIDENCE ON NORMAL DISTRIBUTION /
      1          12H (ONE    MIN     5F10.4 / 12H SIDED) MAX     5F10.4//
      2          12H (TWO    MIN     5F10.4 / 12H SIDED) MAX     5F10.4)
 2080 FORMAT (12H0MEAN          9F12.2 / 12H STAND. DEV. 9F12.2)
C
      COUNT(1) = 0
      COUNT(2) = 0
      COUNT(3) = 0
      BLANK = 0.
C
C                READ DATA AND GROUP IT INTO TEMPERATURE GROUPS
C
    5 READ (5,81) NP, (EXPA(I), I = 1,NP)
      WRITE (6,16) NP, (EXPA(I), I = 1,NP)
      READ (5,81) NT, (EXTEM(I), I=1,NT)
      READ (5,81) NT, (TGRP (I), I=1,NT)
      TGRP (NT + 1) = TGRP (NT) * 2.
      WRITE (6,18) NT, (EXTEM(I), I = 1,NT)
      READ (5,82) G, PHI, NTOT, NBURN, NFIRST
      WRITE (6,20) G, PHI
      READ (5,83) CT1, CT2, CP1, CP2
C
C                ASSIGN DATA FOR A MOTOR INTO PROPER TEMPERATURE GROUP
C
    9 READ (5,84) XMOT , FIRETP, ENDMOT
      DO 19  J = 1,3
      EPS(J) = (TGRP (J+1) - TGRP (J)) *.5
      IF ((TGRP (J) + EPS(J)) .GT. FIRETP) GO TO 25
   19 CONTINUE
   25 COUNT(J) = COUNT(J) + 1
      K = COUNT(J)
   45 TEMP (J,K) = FIRETP
      MOTNO(J,K) = XMOT
C
C                READ REMAINDER OF DATA FOR A MOTOR
C
      READ (5,85) AT1(J,K), AT2(J,K), AT3(J,K), AT4(J,K),
     1              AE1(J,K), AE2(J,K), AE3(J,K), AE4(J,K),
     2               WP(J,K), WEB(J,K),DENS(J,K), PAF(J,K)
      READ (5,86) (TIME(I,J,K), I = 1,NTIMES)
      READ (5,86) (INTP(I,J,K), I = 1,NTIMES)
      READ (5,86) (   P(I,J,K), I = 1,NTIMES)
      READ (5,86) (INTF(I,J,K), I = 1,NTIMES)
      READ (5,86) (   F(I,J,K), I = 1,NTIMES)
      IF (ENDMOT .LT. 1) GO TO 9
C
C                PROCEED WITH MAIN BODY OF PROGRAM
C
      J1= COUNT(1)
      J2 = COUNT(2)
      J3 = COUNT(3)
      TEMSEL = 1
      CPHI = COS (PHI)
C
C           COMPUTE P(I,J,K), F(I,J,K) AND STATISTICS ON P,F AND TIME
```

```
C
      DO 115  I = 1,NTIMES
      ISW = 1
      IF (P(I,J,K) .GT. 0.) ISW = ISW + 1
      IF (F(I,J,K) .GT. 0.) ISW = ISW + 2
      DO 115  J = 1,NT
      L = 0
      JJ = COUNT(J)
      DO 108  K = 1,JJ
      GO TO (104, 105, 106, 107), ISW
  104 P(I,J,K) = INTP(I,J,K) / TIME(I,J,K)
  105 F(I,J,K) = INTF(I,J,K) / TIME(I,J,K)
      GO TO 107
  106 P(I,J,K) = INTP(I,J,K) / TIME(I,J,K)
  107 L = L + 1
      SET1(L) = P(I,J,K)
      SET2(L) = F(I,J,K)
  108 SET3(L) = TIME(I,J,K)
      CALL STATS (SET1, L, MP    (I,J), SDP   (I,J))
      CALL STATS (SET2, L, MF    (I,J), SDF   (I,J))
  115 CALL STATS (SET3, L, MTIME(I,J), SDTIME(I,J))
      WRITE (6,2005)
      DO 55   J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2006) J, (MOTNO(J,K), (TIME(I,J,K),I=1,9), K = 1,JJ)
   55 WRITE (6,2055) (MTIME(I,J), I=1,9), (SDTIME(I,J),I=1,9)
  116 WRITE (6,2007)
      DO 118  J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2046) J, (MOTNO(J,K), (P(I,J,K), I = 1,9), K = 1,JJ)
  118 WRITE (6,2060) (MP(I,J), I=1,9), (SDP(I,J), I=1,9)
      WRITE (6,2014)
  216 WRITE (6,2015)
      DO 218  J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (F(I,J,K), I = 1,9), K = 1,JJ)
  218 WRITE (6,2080) (MF(I,J), I=1,9), (SDF(I,J), I=1,9)
C
C              COMPUTE STATISTICS FOR TEMPERATURE
C
      DO 270  J = 1,NT
      L = 0
      JJ = COUNT(J)
      DO 260  K = 1,JJ
      L = L + 1
  260 SET1(L) = TEMP(J,K)
  270 CALL STATS (SET1, L, MTEMP(J), SDTEMP(J))
C
C              COMPUTE AETOTAL, ATTOTAL AND AREA RATIOS FOR FUTURE USE
C
      L = 0
      DO 704  J = 1,NT
      JJ = COUNT(J)
      DO 704  K = 1,JJ
      L = L + 1
      AETOT(J,K)= AE1(J,K) + AE2(J,K) + AE3(J,K) + AE4(J,K)
      ATTOT(J,K) = AT1(J,K) + AT2(J,K) + AT3(J,K) + AT4(J,K)
      EA(J,K) = AE1(J,K) / AT1(J,K)
      EB(J,K) = AE2(J,K) / AT2(J,K)
      EC(J,K) = AE3(J,K) / AT3(J,K)
      ED(J,K) = AE4(J,K) / AT4(J,K)
      ETOT(J,K) = AETOT(J,K) / ATTOT(J,K)
```

```
      SET1(L) = AT1(J,K)
      SET2(L) = AT2(J,K)
      SET3(L) = AT3(J,K)
      SET4(L) = AT4(J,K)
      SET5(L) = ATTOT(J,K)
      SET6(L) = AE1(J,K)
      SET7(L) = AE2(J,K)
  704 TENT2(L)= AE3(J,K)
      CALL STATS (SET1, L, MAT(1), SDAT(1))
      CALL STATS (SET2, L, MAT(2), SDAT(2))
      CALL STATS (SET3, L, MAT(3), SDAT(3))
      CALL STATS (SET4, L, MAT(4), SDAT(4))
      CALL STATS (SET5, L, MAT(5), SDAT(5))
      CALL STATS (SET6, L, MAE(1), SDAE(1))
      CALL STATS (SET7, L, MAE(2), SDAE(2))
      CALL STATS (TENT2,L, MAE(3), SDAE(3))
      L = 0
      DO 1704 J = 1,NT
      JJ = COUNT(J)
      DO 1704 K = 1,JJ
      L = L + 1
      SET1(L) = EA(J,K)
      SET2(L) = EB(J,K)
      SET3(L) = EC(J,K)
      SET4(L) = ED(J,K)
      SET5(L) = ETOT(J,K)
      SET6(L) = AE4(J,K)
 1704 SET7(L) = AETOT(J,K)
      CALL STATS (SET1, L, ME(1), SDE(1))
      CALL STATS (SET2, L, ME(2), SDE(2))
      CALL STATS (SET3, L, ME(3), SDE(3))
      CALL STATS (SET4, L, ME(4), SDE(4))
      CALL STATS (SET5, L, ME(5), SDE(5))
      CALL STATS (SET6, L, MAE(4), SDAE(4))
      CALL STATS (SET7, L, MAE(5), SDAE(5))
      DO 1706 K = 1,5
      C1MN(K) = MAT(K) - CP1*SDAT(K)
      C1MX(K) = MAT(K) + CP1*SDAT(K)
      C2MN(K) = MAT(K) - CP2*SDAT(K)
 1706 C2MX(K) = MAT(K) + CP2*SDAT(K)
C
C               PERFORM CALCULATIONS ON NON-TRANSFORMED DATA
C
      DO 1100 I = NFIRST,NTIMES
      DO 1100 J = 1,NT
      L = 0
      JJ = COUNT(J)
      DO 1076 K = 1,JJ
      L = L + 1
      IF (INTP(I,J,K) .NE. 0.) GO TO 1070
      INTP(I,J,K) = P(I,J,K) * TIME(I,J,K)
 1070 IF (INTF(I,J,K) .NE. 0.) GO TO 1075
      INTF(I,J,K) = F(I,J,K) * TIME(I,J,K)
 1075 WPU(I,J,K) = WP(J,K)*INTP(I,J,K) / INTP(NTOT,J,K)
      CSTU(I,J,K) = INTP(I,J,K)*G*ATTOT(J,K) / WPU(I,J,K)
      ISP(I,J,K)  = INTF(I,J,K) / WPU(I,J,K)
      FVAC1(I,J,K)=  F(I,J,K) + PAF(J,K)*AETOT(J,K)*CPHI
      IVAC1(I,J,K)= FVAC1(I,J,K) * TIME(I,J,K)
      ISP1(I,J,K) = IVAC1(I,J,K) / WPU(I,J,K)
      TENT2(L)= CSTU(I,J,K)
      SET1(L) = IVAC1(I,J,K)
      SET2(L) = INTP(I,J,K)
```

```
      SET3(L) = INTF(I,J,K)
      SET4(L) = FVAC1(I,J,K)
      SET5(L) = ISP(I,J,K)
      SET6(L) = WPU(I,J,K)
 1076 SET7(L) = ISP1 (I,J,K)
      CALL STATS (TENT2,L, MCSTU(I,J), SDCSTU(I,J))
      CALL STATS (SET1, L, MIVC1(I,J), SDIVC1(I,J))
      CALL STATS (SET2, L, MINTP(I,J), SDINTP(I,J))
      CALL STATS (SET3, L, MINTF(I,J), SDINTF(I,J))
      CALL STATS (SET4, L, MFVC1(I,J), SDFVC1(I,J))
      CALL STATS (SET5, L, MISP (I,J), SDISP (I,J))
      CALL STATS (SET6, L, MWPU (I,J), SDWPU (I,J))
      CALL STATS (SET7, L, MIU11(I,J), SDIU11(I,J))
 1100 CONTINUE
C
      WRITE (6,2008)
      DO 1080 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (INTP(I,J,K), I = 1,9), K = 1,JJ)
 1080 WRITE (6,2080) (MINTP(I,J), I=1,9), (SDINTP(I,J), I=1,9)
      WRITE (6,2014)
      WRITE (6,2016)
      DO 1085 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (INTF(I,J,K),I = 1,9), K = 1,JJ)
 1085 WRITE (6,2080) (MINTF(I,J), I=1,9), (SDINTF(I,J), I=1,9)
      WRITE (6,2014)
      WRITE (6,2017)
      DO 1090 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (ISP(I,J,K), I = 1,9), K = 1,JJ)
 1090 WRITE (6,2080) (MISP (I,J), I=1,9), (SDISP (I,J), I=1,9)
      WRITE (6,2024) EXPA(1)
      WRITE (6,2015)
      DO 1061 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (FVAC1(I,J,K), I=1,9), K=1,JJ)
 1061 WRITE (6,2080) (MFVC1(I,J), I=1,9), (SDFVC1(I,J), I=1,9)
      WRITE (6,2024) EXPA(1)
      WRITE (6,2016)
      DO 1062 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (IVAC1(I,J,K), I=1,9), K=1,JJ)
 1062 WRITE (6,2080) (MIVC1(I,J), I=1,9), (SDIVC1(I,J), I=1,9)
      WRITE (6,2024) EXPA(1)
      WRITE (6,2017)
      DO 1063 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (ISP1 (I,J,K),I=1,9), K=1,JJ)
 1063 WRITE (6,2080) (MIU11(I,J), I=1,9), (SDIU11(I,J), I=1,9)
C
      IF (NP .LE. 1) GO TO 1099
      DO 1101 I = NFIRST,NTIMES
      DO 1101 J = 1,NT
      L = 0
      JJ = COUNT(J)
      DO 1160 K = 1,JJ
      L = L + 1
      I12(I,J,K)  = IVAC1(I,J,K) - (EXPA(2)*AETOT(J,K)*CPHI)*TIME(I,J,K)
      F12(I,J,K)  = I12(I,J,K) / TIME(I,J,K)
      ISPU12(I,J,K) = I12(I,J,K) / WPU(I,J,K)
      SET1(L) = I12(I,J,K)
```

```
      SET4(L) = F12(I,J,K)
 1160 SET7(L) = ISPU12(I,J,K)
      CALL STATS (SET1, L, MI12(I,J), SDI12(I,J))
      CALL STATS (SET4, L, MF12(I,J), SDF12(I,J))
 1101 CALL STATS (SET7, L, MIU12(I,J), SDIU12(I,J))
C
      WRITE (6,2024) EXPA(2)
      WRITE (6,2015)
      DO 1071 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (F12(I,J,K), I=1,9), K=1,JJ)
 1071 WRITE (6,2080) (MF12(I,J), I=1,9), (SDF12(I,J), I=1,9)
      WRITE (6,2024) EXPA(2)
      WRITE (6,2016)
      DO 1072 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (I12(I,J,K), I=1,9), K=1,JJ)
 1072 WRITE (6,2080) (MI12(I,J), I=1,9), (SDI12(I,J), I=1,9)
      WRITE (6,2024) EXPA(2)
      WRITE (6,2017)
      DO 1073 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (ISPU12(I,J,K), I=1,9), K=1,JJ)
 1073 WRITE (6,2080) (MIU12(I,J), I=1,9), (SDIU12(I,J), I=1,9)
      IF (NP .LE. 2) GO TO 1099
C
      DO 1102 I = NFIRST,NTIMES
      DO 1102 J = 1,NT
      L = 0
      JJ = COUNT(J)
      DO 1180 K = 1,JJ
      L = L + 1
      I13(I,J,K)  = IVAC1(I,J,K) - (EXPA(3)*AETOT(J,K)*CPHI)*TIME(I,J,K)
      F13(I,J,K)  = I13(I,J,K) / TIME(I,J,K)
      ISPU13(I,J,K) = I13(I,J,K) / WPU(I,J,K)
      SET1(L) = I13(I,J,K)
      SET4(L) = F13(I,J,K)
 1180 SET7(L) = ISPU13(I,J,K)
      CALL STATS (SET1, L, MI13(I,J), SDI13(I,J))
      CALL STATS (SET4, L, MF13(I,J), SDF13(I,J))
 1102 CALL STATS (SET7, L, MIU13(I,J), SDIU13(I,J))
C
      WRITE (6,2024) EXPA(3)
      WRITE (6,2015)
      DO 1081 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (F13(I,J,K), I=1,9), K=1,JJ)
 1081 WRITE (6,2080) (MF13(I,J), I=1,9), (SDF13(I,J), I=1,9)
      WRITE (6,2024) EXPA(3)
      WRITE (6,2016)
      DO 1082 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (I13(I,J,K), I=1,9), K=1,JJ)
 1082 WRITE (6,2080) (MI13(I,J), I=1,9), (SDI13(I,J), I=1,9)
      WRITE (6,2024) EXPA(3)
      WRITE (6,2017)
      DO 1083 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (ISPU13(I,J,K), I=1,9), K=1,JJ)
 1083 WRITE (6,2080) (MIU13(I,J),I=1,9), (SDIU13(I,J),I=1,9)
      IF (NP .LE. 3) GO TO 1099
C
```

```
      DO 1103 I = NFIRST,NTIMES
      DO 1103 J = 1,NT
      L = 0
      JJ = COUNT(J)
      DO 1200 K = 1,JJ
      L = L + 1
      I14(I,J,K)  = IVAC1(I,J,K) - (EXPA(4)*AETOT(J,K)*CPHI)*TIME(I,J,K)
      F14(I,J,K)  = I14(I,J,K) / TIME(I,J,K)
      ISPU14(I,J,K) = I14(I,J,K) / WPU(I,J,K)
      SET1(L) = I14(I,J,K)
      SET4(L) = F14(I,J,K)
 1200 SET7(L) = ISPU14(I,J,K)
      CALL STATS (SET1, L, MI14(I,J), SDI14(I,J))
      CALL STATS (SET4, L, MF14(I,J), SDF14(I,J))
 1103 CALL STATS (SET7, L, MIU14(I,J), SDIU14(I,J))
      WRITE (6,2024) EXPA(4)
      WRITE (6,2015)
      DO 1091 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (F14(I,J,K), I=1,9), K=1,JJ)
 1091 WRITE (6,2080) (MF14(I,J), I=1,9), (SDF14(I,J), I=1,9)
      WRITE (6,2024) EXPA(4)
      WRITE (6,2016)
      DO 1092 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (I14(I,J,K), I=1,9), K=1,JJ)
 1092 WRITE (6,2080) (MI14(I,J), I=1,9), (SDI14(I,J), I=1,9)
      WRITE (6,2024) EXPA(4)
      WRITE (6,2017)
      DO 1093 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2045) J, (MOTNO(J,K), (ISPU14(I,J,K), I=1,9), K=1,JJ)
 1093 WRITE (6,2080) (MIU14(I,J),I=1,9), (SDIU14(I,J),I=1,9)
C
C              COMPUTE RU, KN AND THEIR STATISTICS
C
 1099 DO 1104 J = 1,NT
      L = 0
      JJ = COUNT(J)
      DO 1120 K = 1,JJ
      L = L + 1
      RU(J,K) = WEB(J,K) / TIME(NBURN,J,K)
      KN(J,K) = G * INTP(NBURN,J,K)/ (DENS(J,K)*CSTU(NBURN,J,K)*RU(J,K))
      SET6(L) = RU   (J,K)
      SET7(L) = KN   (J,K)
 1120 CONTINUE
      CALL STATS (SET6, L, MRU   (   J), SDRU   (   J))
      CALL STATS (SET7, L, MKN   (   J), SDKN   (   J))
 1104 CONTINUE
      WRITE (6,2009)
      DO 1095 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2046) J, (MOTNO(J,K), (CSTU(I,J,K), I = 1,9), K = 1,JJ)
 1095 WRITE (6,2060) (MCSTU(I,J),I=1,9), (SDCSTU(I,J),I=1,9)
      WRITE (6,2012)
      DO 1077 J = 1,NT
      JJ = COUNT(J)
      WRITE (6,2046) J,(MOTNO(J,K), (WPU(I,J,K), I=1,9), K = 1,JJ)
 1077 WRITE (6,2060) (MWPU(I,J), I=1,9), (SDWPU(I,J), I=1,9)
      WRITE (6,2010)
      DO 1098 J = 1,NT
      JJ = COUNT(J)
```

```
      WRITE (6,2044) J, (MOTNO(J,K), RU(J,K), KN(J,K), K=1,JJ)
 1098 WRITE (6,2063) MRU(J), MKN(J), SDRU(J), SDKN(J)
C
C                 PRINTOUT OF THROAT AREAS, EXIT AREAS AND AREA RATIOS
C
      WRITE (6,2001)
      DO 701  J = 1,NT
      JJ = COUNT(J)
  701 WRITE (6,2002) J, (MOTNO(J,K),   AT1(J,K), AT2(J,K), AT3(J,K),
     1                       AT4(J,K), ATTOT(J,K), K = 1,JJ)
      WRITE (6,2070) MAT, SDAT
      WRITE (6,2071) C1MN, C1MX, C2MN, C2MX
      DO 1707 K = 1,5
      C1MN(K) = MAE(K) - CP1*SDAE(K)
      C1MX(K) = MAE(K) + CP1*SDAE(K)
      C2MN(K) = MAE(K) - CP2*SDAE(K)
 1707 C2MX(K) = MAE(K) + CP2*SDAE(K)
      WRITE (6,2003)
      DO 702  J = 1,NT
      JJ = COUNT(J)
  702 WRITE (6,2002) J, (MOTNO(J,K),   AE1(J,K), AE2(J,K), AE3(J,K),
     1                       AE4(J,K), AETOT(J,K), K = 1,JJ)
      WRITE (6,2070) MAE, SDAE
      WRITE (6,2071) C1MN, C1MX, C2MN, C2MX
      DO 1708 K = 1,5
      C1MN(K) = ME  (K) - CP1*SDE  (K)
      C1MX(K) = ME  (K) + CP1*SDE  (K)
      C2MN(K) = ME  (K) - CP2*SDE  (K)
 1708 C2MX(K) = ME  (K) + CP2*SDE  (K)
      WRITE (6,2004)
      DO 703  J = 1,NT
      JJ = COUNT(J)
  703 WRITE (6,2002) J, (MOTNO(J,K),    EA(J,K),  EB(J,K),  EC(J,K),
     1                        ED(J,K),  ETOT(J,K), K = 1,JJ)
      WRITE (6,2070) ME , SDE
      WRITE (6,2071) C1MN, C1MX, C2MN, C2MX
C
C                 COMPUTE CONSTITUENTS OF LNP USING LEAST SQUARES
C
      LL = 0
      DO 282  J = 1,NT
      JJ = COUNT(J)
      DO 282  K = 1,JJ
      LL = LL + 1
      TENTRY(LL) = TEMP(J,K)
  282 TENT2 (LL) = TENTRY(LL)**2
C
  300 DO 311  I = 1,NTIMES
      IFLAG1(I) = 0
      IFLAG2(I) = 0
      IF (P(I,1,1)) 308, 308, 309
  308 IFLAG1(I) = 1
      GO TO 311
  309 LL = 0
      DO 310  J = 1,NT
      JJ = COUNT(J)
      DO 310  K = 1,JJ
      LL = LL + 1
      ENTRY(I,LL) = ALOG (P(I,J,K))
  310 CONTINUE
  311 CONTINUE
      NM = COUNT(1) + COUNT(2) + COUNT(3)
```

```
C
      DO 320  I = 1,NTIMES
      IF(IFLAG1(I) - 1) 312,320,312
  312 DO 315  LL = 1,NM
      ARRAY(LL,1) = 1.
      ARRAY(LL,2) = TENTRY(LL)
      ARRAY(LL,3) = TENT2(LL)
  315 ARRAY(LL,4) = ENTRY(I,LL)
      CALL GLS1 (ARRAY, X, IL, NM, 3, ALPHA, 0., 0.)
      COEF1 (I,1) = X(1)
      COEF1 (I,2) = X(2)
      COEF1 (I,3) = X(3)
  320 CONTINUE
C
C                 COMPUTE CONSTITUENTS OF LNT USING LEAST SQUARES
C
  400 DO 411  I = 1,NTIMES
      IF (TIME(I,1,1)) 408, 408, 409
  408 IFLAG2(I) = 1
      GO TO 411
  409 LL = 0
      DO 410  J = 1,NT
      JJ = COUNT(J)
      DO 410  K = 1,JJ
      LL = LL + 1
      ENTRY (I,LL) = ALOG (TIME(I,J,K))
  410 CONTINUE
  411 CONTINUE
C
      DO 420  I = 1,NTIMES
      IF (IFLAG2(I) - 1) 412,420,412
  412 DO 415 LL = 1,NM
      ARRAY(LL,1) = 1.
      ARRAY(LL,2) = TENTRY(LL)
      ARRAY(LL,3) = TENT2(LL)
  415 ARRAY(LL,4) = ENTRY(I,LL)
      CALL GLS1 (ARRAY, X, IL, NM, 3, ALPHA, 0., 0.)
      COEF2 (I,1) = X(1)
      COEF2 (I,2) = X(2)
      COEF2 (I,3) = X(3)
  420 CONTINUE
C
C                 COMPUTE ACTUAL LNP AND LNT
C
  450 DO 460  I = 1,NTIMES
      DO 460  J = 1,NT
      LNP(I,J) = COEF1(I,1)+ COEF1(I,2)*EXTEM(J)+ COEF1(I,3)*EXTEM(J)**2
  460 LNT(I,J) = COEF2(I,1)+ COEF2(I,2)*EXTEM(J)+ COEF2(I,3)*EXTEM(J)**2
C
      DO 462  I = 1,NTIMES
      SET1(1) = 0.
      IF (MP(I,1) .GT. 0.) SET1(1)= COEF1(I,1) + COEF1(I,2)*MTEMP(1) +
     1                              COEF1(I,3)*MTEMP(1)**2
      SET1(2) = 0.
      IF (MP(I,2) .GT. 0.) SET1(2)= COEF1(I,1) + COEF1(I,2)*MTEMP(2) +
     1                              COEF1(I,3)*MTEMP(2)**2
      SET1(3) = 0.
      IF (MP(I,3) .GT. 0.) SET1(3)= COEF1(I,1) + COEF1(I,2)*MTEMP(3) +
     1                              COEF1(I,3)*MTEMP(3)**2
      PIK11(I) = (LNP(I,1) - SET1(1)) / (EXTEM(1) - MTEMP(1))
      PIK12(I) = (LNP(I,2) - SET1(1)) / (EXTEM(2) - MTEMP(1))
      PIK13(I) = (LNP(I,3) - SET1(1)) / (EXTEM(3) - MTEMP(1))
```

```
      PIK21(I) = (LNP(I,1) - SET1(2)) / (EXTEM(1) - MTEMP(2))
      PIK22(I) = (LNP(I,2) - SET1(2)) / (EXTEM(2) - MTEMP(2))
      PIK23(I) = (LNP(I,3) - SET1(2)) / (EXTEM(3) - MTEMP(2))
      PIK31(I) = (LNP(I,1) - SET1(3)) / (EXTEM(1) - MTEMP(3))
      PIK32(I) = (LNP(I,2) - SET1(3)) / (EXTEM(2) - MTEMP(3))
      PIK33(I) = (LNP(I,3) - SET1(3)) / (EXTEM(3) - MTEMP(3))
      SET2(1) = 0.
      IF (MTIME(I,1) .GT. 0.) SET2(1)= COEF2(I,1) +COEF2(I,2)*MTEMP(1) +
     1                                COEF2(I,3)*MTEMP(1)**2
      SET2(2) = 0.
      IF (MTIME(I,2) .GT. 0.) SET2(2)= COEF2(I,1) +COEF2(I,2)*MTEMP(2) +
     1                                COEF2(I,3)*MTEMP(2)**2
      SET2(3) = 0.
      IF (MTIME(I,3) .GT. 0.) SET2(3)= COEF2(I,1) +COEF2(I,2)*MTEMP(3) +
     1                                COEF2(I,3)*MTEMP(3)**2
      SIGP11(I) = (SET2(1) - LNT(I,1)) / (EXTEM(1) - MTEMP(1))
      SIGP12(I) = (SET2(1) - LNT(I,2)) / (EXTEM(2) - MTEMP(1))
      SIGP13(I) = (SET2(1) - LNT(I,3)) / (EXTEM(3) - MTEMP(1))
      SIGP21(I) = (SET2(2) - LNT(I,1)) / (EXTEM(1) - MTEMP(2))
      SIGP22(I) = (SET2(2) - LNT(I,2)) / (EXTEM(2) - MTEMP(2))
      SIGP23(I) = (SET2(2) - LNT(I,3)) / (EXTEM(3) - MTEMP(2))
      SIGP31(I) = (SET2(3) - LNT(I,1)) / (EXTEM(1) - MTEMP(3))
      SIGP32(I) = (SET2(3) - LNT(I,2)) / (EXTEM(2) - MTEMP(3))
  462 SIGP33(I) = (SET2(3) - LNT(I,3)) / (EXTEM(3) - MTEMP(3))
      WRITE (6,2050)
      WRITE (6,2051) PIK11, PIK21, PIK31, PIK12, PIK22, PIK32, PIK13,
     1               PIK23, PIK33
      WRITE (6,2052)SIGP11,SIGP21,SIGP31,SIGP12,SIGP22,SIGP32,SIGP13,
     1              SIGP23,SIGP33
C
C             COMPUTE P TT AND TIME TT TRANSFORMATIONS - SELECTOR
C             CHOOSES WHCIH TEMPERATURE DATA IS TO BE TRANSFORMED TO
C
      I1 = 1
      I2 = 2
      I3 = 3
  480 GO TO (500, 520, 540),TEMSEL
C
  500 DO 510  I = 1,NTIMES
      JJ = COUNT(1)
      DO 502  K =1,J1
      IF (IFLAG1(I) .EQ. 1) GO TO 501
      P1(I,1,K) = P(I,1,K) * EXP (PIK11(I)*(EXTEM(1)-TEMP(1,K)))
  501 IF (IFLAG2(I) .EQ. 1) GO TO 502
      T1(I,1,K) = TIME(I,1,K) / (EXP (SIGP11(I)*(EXTEM(1)- TEMP(1,K))))
  502 CONTINUE
      DO 504  K =1,J2
      IF (IFLAG1(I) .EQ. 1) GO TO 503
      P2(I,1,K) = P(I,2,K) * EXP (PIK21(I)*(EXTEM(1)-TEMP(2,K)))
  503 IF (IFLAG2(I) .EQ. 1) GO TO 504
      T2(I,1,K) = TIME(I,2,K) / (EXP (SIGP21(I)*(EXTEM(1)- TEMP(2,K))))
  504 CONTINUE
      DO 510  K =1,J3
      IF (IFLAG1(I) .EQ. 1) GO TO 508
      P3(I,1,K) = P(I,3,K) * EXP (PIK31(I)*(EXTEM(1)-TEMP(3,K)))
  508 IF (IFLAG2(I) .EQ. 1) GO TO 510
      T3(I,1,K) = TIME(I,3,K) / (EXP (SIGP31(I)*(EXTEM(1)- TEMP(3,K))))
  510 CONTINUE
      JSAVE = 1
      GO TO 580
C
  520 DO 530  I = 1,NTIMES
```

```
      JJ = COUNT(2)
      DO 522  K = 1,J1
      IF (IFLAG1(I) .EQ. 1) GO TO 521
      P1(I,2,K) = P(I,1,K) * EXP (PIK12(I)*(EXTEM(2)-TEMP(1,K)))
  521 IF (IFLAG2(I) .EQ. 1) GO TO 522
      T1(I,2,K) = TIME(I,1,K) / (EXP (SIGP12(I)*(EXTEM(2)- TEMP(1,K))))
  522 CONTINUE
      DO 524  K = 1,J2
      IF (IFLAG1(I) .EQ. 1) GO TO 523
      P2(I,2,K) = P(I,2,K) * EXP (PIK22(I)*(EXTEM(2)-TEMP(2,K)))
  523 IF (IFLAG2(I) .EQ. 1) GO TO 524
      T2(I,2,K) = TIME(I,2,K) / (EXP (SIGP22(I)*(EXTEM(2)- TEMP(2,K))))
  524 CONTINUE
      DO 530  K =1,J3
      IF (IFLAG1(I) .EQ. 1) GO TO 528
      P3(I,2,K) = P(I,3,K) * EXP (PIK32(I)*(EXTEM(2)-TEMP(3,K)))
  528 IF (IFLAG2(I) .EQ. 1) GO TO 530
      T3(I,2,K) = TIME(I,3,K) / (EXP (SIGP32(I)*(EXTEM(2)- TEMP(3,K))))
  530 CONTINUE
      JSAVE = 2
      GO TO 580
C
  540 DO 550  I = 1,NTIMES
      JJ □ COUNT(3)
      DO 542  K = 1,J1
      IF (IFLAG1(I) .EQ. 1) GO TO 541
      P1(I,3,K) = P(I,1,K) * EXP (PIK13(I)*(EXTEM(3)-TEMP(1,K)))
  541 IF (IFLAG2(I) .EQ. 1) GO TO 542
      T1(I,3,K) = TIME(I,1,K) / (EXP (SIGP13(I)*(EXTEM(3)- TEMP(1,K))))
  542 CONTINUE
      DO 544  K =1,J2
      IF (IFLAG1(I) .EQ. 1) GO TO 543
      P2(I,3,K) = P(I,2,K) * EXP (PIK23(I)*(EXTEM(3)-TEMP(2,K)))
  543 IF (IFLAG2(I) .EQ. 1) GO TO 544
      T2(I,3,K) = TIME(I,2,K) / (EXP (SIGP23(I)*(EXTEM(3)- TEMP(2,K))))
  544 CONTINUE
      DO 550  K =1,J3
      IF (IFLAG1(I) .EQ. 1) GO TO 548
      P3(I,3,K) = P(I,3,K) * EXP (PIK33(I)*(EXTEM(3)-TEMP(3,K)))
  548 IF (IFLAG2(I) .EQ. 1) GO TO 550
      T3(I,3,K) = TIME(I,3,K) / (EXP (SIGP33(I)*(EXTEM(3)- TEMP(3,K))))
  550 CONTINUE
      JSAVE = 3
      GO TO 580
C
C                COMPUTE STATISTICS FOR PTT AND TIME TT
C
  580 J = JSAVE
      DO 640  I = 1,NTIMES
      L = 0
      M = 0
      DO 620  K = 1,J1
      L = L + 1
      M = M + 1
      TENTRY(L)= P1(I,J,K)
      TENT2(L) = T1(I,J,K)
      SET1(M ) = P1(I,J,K)
  620 SET4(M)  = T1(I,J,K)
      M = 0
      DO 625  K = 1,J2
      L = L + 1
      M = M + 1
```

```
      TENTRY(L)= P2(I,J,K)
      TENT2(L) = T2(I,J,K)
      SET2(M)  = P2(I,J,K)
  625 SET5(M) .= T2(I,J,K)
      M = 0
      DO 630  K = 1,J3
      L = L + 1
      M = M + 1
      TENTRY(L)= P3(I,J,K)
      TENT2(L) = T3(I,J,K)
      SET3(M)  = P3(I,J,K)
  630 SET6(M)  = T3(I,J,K)
      CALL STATS (SET1, J1, MPT1(I,J), SDPT1(I,J))
      CALL STATS (SET2, J2, MPT2(I,J), SDPT2(I,J))
      CALL STATS (SET3, J3, MPT3(I,J), SDPT3(I,J))
      CALL STATS (SET4, J1, MTT1(I,J), SDTT1(I,J))
      CALL STATS (SET5, J2, MTT2(I,J), SDTT2(I,J))
      CALL STATS (SET6, J3, MTT3(I,J), SDTT3(I,J))
      CALL STATS (TENTRY,L, MPTT(I,J), SDPTT(I,J))
  640 CALL STATS (TENT2, L, MTTT(I,J), SDTTT(I,J))
      IF (TEMSEL .GE. NT) GO TO 655
      TEMSEL = TEMSEL + 1
      GO TO 480
C
C                 COMPUTE INTEGRALS OF P TRANSFORMED
C
  655 DO 700 I = 1,NTIMES
      DO 700 J = 1,NT
      L = 0
      M = 0
      DO 660  K = 1,J1
      INTPT1 (I,J,K) = P1(I,J,K) * T1(I,J,K)
      L = L + 1
      M = M + 1
      TENTRY(L) = INTPT1(I,J,K)
 .660 SET1(M) = INTPT1(I,J,K)
      M = 0
      DO 670  K = 1,J2
      L = L + 1
      M = M + 1
      INTPT2 (I,J,K) = P2(I,J,K) * T2(I,J,K)
      TENTRY(L)= INTPT2(I,J,K)
  670 SET2(M)  = INTPT2(I,J,K)
      M = 0
      DO 680  K = 1,J3
      L = L + 1
      M = M + 1
      INTPT3 (I,J,K) = P3(I,J,K) * T3(I,J,K)
      TENTRY(L)= INTPT3(I,J,K)
  680 SET3(M)  = INTPT3(I,J,K)
C
C                 COMPUTE STATS ON INTEGRALS OF P TRANSFORMED
C
      CALL STATS (SET1, J1, MINT1(I,J), SDINT1(I,J))
      CALL STATS (SET2, J2, MINT2(I,J), SDINT2(I,J))
      CALL STATS (SET3, J3, MINT3(I,J), SDINT3(I,J))
  700 CALL STATS (TENTRY,L, MINTT(I,J), SDINTT(I,J))
  575 DO 565  I = 1,NTIMES
      WRITE (6,2030) I
      WRITE (6,2047)
      WRITE (6,2040)I1, (MOTNO(1,K), T1(1,1,K), T1(I,2,K), T1(I,3,K),
     1                        BLANK      , K = 1,J1)
```

```
      WRITE (6,2064) (MTT1(I,J), J=1,4), (SDTT1(I,J), J=1,4)
      IF (NT .LE. 1) GO TO 561
      WRITE (6,2040)I2, (MOTNO(2,K), T2(I,1,K), T2(I,2,K), T2(I,3,K),
     1                  BLANK       , K = 1,J2)
      WRITE (6,2064) (MTT2(I,J), J=1,4), (SDTT2(I,J), J=1,4)
      IF (NT .LE. 2) GO TO 561
      WRITE (6,2040)I3, (MOTNO(3,K), T3(I,1,K), T3(I,2,K), T3(I,3,K),
     1                  BLANK       , K = 1,J3)
      WRITE (6,2064) (MTT3(I,J), J=1,4), (SDTT3(I,J), J=1,4)
  561 WRITE (6,2065) (MTTT(I,J), J=1,4), (SDTTT(I,J), J=1,4)
      DO 562  J = 1,NT
      C1MIN(J) = MTTT(I,J) - CT1*SDTTT(I,J)
      C1MAX(J) = MTTT(I,J) + CT1*SDTTT(I,J)
      C2MIN(J) = MTTT(I,J) - CT2*SDTTT(I,J)
  562 C2MAX(J) = MTTT(I,J) + CT2*SDTTT(I,J)
      WRITE (6,2062) C1MIN, C1MAX, C2MIN, C2MAX
  565 CONTINUE
      DO 570  I = 1,NTIMES
      WRITE (6,2031) I
      WRITE (6,2042)
      WRITE (6,2048)
      WRITE (6,2041)I1, (MOTNO(1,K), INTPT1(I,1,K), INTPT1(I,2,K),
     1 INTPT1(I,3,K), BLANK, P1(I,1,K), P1(I,2,K), P1(I,3,K),
     2 BLANK, K = 1,J1)
      WRITE (6,2066) (MINT1(I,J,, J =1,4), (MPT1(I,J), J=1,4),
     1               (SDINT1(I,J), J =1,4), (SDPT1(I,J), J=1,4)
      IF (NT .LE. 1) GO TO 564
      WRITE (6,2041)I2, (MOTNO(2,K), INTPT2(I,1,K), INTPT2(I,2,K),
     1 INTPT2(I,3,K), BLANK, P2(I,1,K), P2(I,2,K), P2(I,3,K),
     2 BLANK, K = 1,J2)
      WRITE (6,2066) (MINT2(I,J), J =1,4), (MPT2(I,J), J=1,4),
     1               (SDINT2(I,J), J =1,4), (SDPT2(I,J), J=1,4)
      IF (NT .LE. 2) GO TO 564
      WRITE (6,2041)I3, (MOTNO(3,K), INTPT3(I,1,K), INTPT3(I,2,K),
     1 INTPT3(I,3,K), BLANK, P3(I,1,K), P3(I,2,K), P3(I,3,K),
     2 BLANK, K = 1,J3)
      WRITE (6,2066) (MINT3(I,J), J =1,4), (MPT3(I,J), J=1,4),
     1               (SDINT3(I,J), J =1,4),(SDPT3(I,J), J=1,4)
  564 WRITE (6,2067) (MINTT(I,J), J =1,4), (MPTT(I,J), J=1,4),
     1               (SDINTT(I,J), J =1,4),(SDPTT(I,J), J=1,4)
      DO 566  J=1,NT
      C1MIN(J) = MINTT(I,J) - CT1*SDINTT(I,J)
      C1MAX(J) = MINTT(I,J) + CT1*SDINTT(I,J)
      C2MIN(J) = MINTT(I,J) - CT2*SDINTT(I,J)
      C2MAX(J) = MINTT(I,J) + CT2*SDINTT(I,J)
      C1TMN(J) = MPTT (I,J) - CT1*SDPTT (I,J)
      C1TMX(J) = MPTT (I,J) + CT1*SDPTT (I,J)
      C2TMN(J) = MPTT (I,J) - CT2*SDPTT (I,J)
  566 C2TMX(J) = MPTT (I,J) + CT2*SDPTT (I,J)
      WRITE (6,2068) C1MIN, C1TMN, C1MAX, C1TMX,
     1                C2MIN, C2TMN, C2MAX, C2TMX
  570 CONTINUE
      RETURN
      END
```

```
$IBFTC GLS1
      SUBROUTINE GLS1(A,X,IL,N,M,ALPHA,E1,E2)
      DIMENSION A(46,4), X(4), IL(4)
      MM=M+1
      LL=1
      DO 60J=1,MM
   60 IL(J)=0
      I=1
      DO 3K=1,MM
      II=I+1
      DO 4J=II,N
      IF (ABS(A(J,K))-E1)4,4,6
    6 T1=SQRT((A(J,K))**2+(A(I,K))**2)
      S=A(J,K)/T1
      C=A(I,K)/T1
      DO 5L=K,MM
      T2=C*A(I,L)+S*A(J,L)
      A(J,L)=-S*A(I,L)+C*A(J,L)
    5 A(I,L)=T2
      LL=LL+1
    4 CONTINUE
      IF (ABS(A(I,K))-E2)3,3,8
    8 IL(K)=I
      I=I+1
    3 CONTINUE
      X(MM)=-1.0
      II=M
      DO 35I=1,M
   35 X(I)=0.
      DO 30J=1,M
      IF (IL(II))30,30,31
   31 S=0.
      LL=II+1
      I=IL(II)
      DO 32K=LL,MM
   32 S=S+A(I,K)*X(K)
      X(II)=-S/A(I,II)
   30 II=II-1
      IF (IL(MM))50,51,50
   51 ALPHA=0.
      GO TO 52
   50 I=IL(MM)
      ALPHA=A(I,MM)
   52 RETURN
      END
```

```
$ORIGIN       ALPHA
$IBFTC MOT2
      SUBROUTINE MOT2
C
C                SECOND LINK OF MOTORS PROGRAM
C
      DIMENSION XX(4860), ZZ(2124)
      COMMON G, CPHI, NT, EXPA, COUNT, WP, WEB, ATTOT, AETOT,
     1       P, MOTNO, FVAC1, NTIMES, XX, NTOT, NBURN, NFIRST, NP
      COMMON CP1, CP2, CT1, CT2, VEC1, VEC2, VEC3
      COMMON  SET1, SET2, SET3, SET4, SET5, SET6, SET7, C1MIN, C1MAX,
     1        C2MIN, C2MAX, C1TMN, C1TMX, C2TMN, C2TMX
      DIMENSION SAVE1(9,4,15), SAVE2(9,4,15), SAVE3(9,4,15)
      DIMENSION     MFT11 (9,4), SDFT11(9,4), MIT11 (9,4), SDIT11(9,4),
     1 MFT21 (9,4), SDFT21(9,4), MIT21 (9,4), SDIT21(9,4), MFT31 (9,4),
     2 SDFT31(9,4), MIT31 (9,4), SDIT31(9,4), MFTT1 (9,4), SDFTT1(9,4),
     3 MITT1 (9,4), SDITT1(9,4), MFT12 (9,4), SDFT12(9,4), MIT12 (9,4),
     4 SDIT12(9,4), MFT22 (9,4), SDFT22(9,4), MIT22 (9,4), SDIT22(9,4),
     5 MFT32 (9,4), SDFT32(9,4), MIT32 (9,4), SDIT32(9,4), MFTT2 (9,4),
     6 SDFTT2(9,4), MITT2 (9,4), SDITT2(9,4)
      DIMENSION     MIS11 (9,4), SDIS11(9,4), MIS21 (8,4), SDIS21(9,4),
     4 MIS31 (9,4), SDIS31(9,4), MIST1 (9,4), SDIST1(9,4), MIS12 (9,4),
     5 SDIS12(9,4), MIS22 (9,4), SDIS22(9,4), MIS32 (9,4), SDIS32(9,4),
     6 MIST2 (9,4), SDIST2(9,4)
C
      DIMENSION
     1 FT11 (9,4,15), FT21 (9,4,15), FT31 (9,4,15), FT12 (9,4,15),
     2 FT22 (9,4,15), FT32 (9,4,15),
     4 IT11 (9,4,15), IT21 (9,4,15), IT31 (9,4,15), IT12 (9,4,15),
     5 IT22 (9,4, 5), IT32 (9,4,15),
     7 ISP11(9,4,15), ISP21(9,4,15), ISP31(9,4,15), ISP12(9,4,15),
     8 ISP22(9,4,15), ISP32(9,4,15)
      DIMENSION RATIO1(9,4,15), RATIO2(9,4,15), RATIO3(9,4,15)
      DIMENSION        WP1 (9,4,15), WP2 (9,4,15), WP3 (9,4,15)
C
C                VARIABLES OBTAINED FROM CHAIN 1
C
      DIMENSION     P1     (9,4,15), P2     (9,4,15), P3     (9,4,15),
     1 EXPA(4),     P      (9,4,15), FVAC1 (9,4,15),
     2 AETOT(4,15),                  INTPT1(9,4,15), INTPT2(9,4,15),
     3ATTOT(4,15),INTPT3(9,4,15), T1     (9,4,15), T2     (9,4,15),
     4 T3     (9,4,15), WP    (4,15), VEC1(4,15), VEC2 (4,15), VEC3(4,15)
      DIMENSION C1MIN(4), C1MAX(4), C2MIN(4), C2MAX(4)
      DIMENSION    SET1 (45), SET2 (45), SET3 (45), SET4 (45), SET5 (45),
     1 SET6 (45 ), TENT2(60), MWP1(9,4), MWP2(9,4), MWP3(9,4),
     2 SDWP1(9,4), SDWP2(9,4), SDWP3(9,4), TENTRY(60),
     3 MOTNO(4,15), COUNT(4), MWPT(9,4), SDWPT(9,4), WEB(4,15)
C
      REAL     MFT11 , MIT11 , MFT21 , MIT21 , MFT31 , MIT31 , MFTT1 ,
     1 MITT1 , MFT12 , MIT12 , MFT22 , MIT22 , MFT32 , MIT32 , MFTT2 ,
     2  IT11 , MOTNO , MWP1  , MWP2  , MWP3  , MWPT  ,
     4 MITT2 , MIS11 , MIS21 , MIS31 , MIST1 , MIS12 , MIS22 , MIS32 ,
     7 MIST2 , IT21  , IT31  , IT12  , IT22  , IT32  , ISP11 ,
     9 ISP21 , ISP31 , ISP12 , ISP22 , ISP32 , INTPT1, INTPT2, INTPT3
      INTEGER COUNT
C
      EQUIVALENCE
     1 (XX(   1), P1(1), WP1(1)),
     2 (XX( 541), P2(1), WP2(1)),
     3 (XX(1081), P3(1), WP3(1)),
     4 (XX(1621), T1(1), IT11(1), IT12(1), ISP11(1), ISP12(1)),
```

```
     5 (XX(2161), T2(1), IT21(1), IT22(1), ISP21(1), ISP22(1)),
     6 (XX(2701), T3(1), IT31(1), IT32(1), ISP31(1), ISP32(1)),
     7 (XX(3241), INTPT1(1)),
     8 (XX(3781), INTPT2(1)),
     9 (XX(4321), INTPT3(1))
      EQUIVALENCE
     1  (ZZ(   1), MFT11(1), MFT12(1), MFT13(1), MFT14(1), MWP1 (1)),
     2  (ZZ(  37), MFT21(1), MFT22(1), MFT23(1), MFT24(1), MWP2 (1)),
     3  (ZZ(  73), MFT31(1), MFT32(1), MFT33(1), MFT34(1), MWP3 (1)),
     4 (ZZ( 109), MIT11(1), MIT12(1), MIS11(1), MIS12(1)),
     5 (ZZ( 145), MIT21(1), MIT22(1), MIS21(1), MIS22(1)),
     6 (ZZ( 181), MIT31(1), MIT32(1), MIS31(1), MIS32(1))
      EQUIVALENCE
     1  (ZZ( 217), SDFT11(1),SDFT12(1),SDFT13(1),SDFT14(1), SDWP1 (1)),
     2  (ZZ( 253), SDFT21(1),SDFT22(1),SDFT23(1),SDFT24(1), SDWP2 (1)),
     3  (ZZ( 289), SDFT31(1),SDFT32(1),SDFT33(1),SDFT34(1), SDWP3 (1)),
     4 (ZZ( 325), SDIT11(1), SDIT12(1), SDIS11(1), SDIS12(1)),
     5 (ZZ( 361), SDIT21(1), SDIT22(1), SDIS21(1), SDIS22(1)),
     6 (ZZ( 397), SDIT31(1), SDIT32(1), SDIS31(1), SDIS32(1))
      EQUIVALENCE
     1  (ZZ( 433), MFTT1(1), MFTT2(1), MFTT3(1), MFTT4(1)),
     2  (ZZ( 469),SDFTT1(1),SDFTT2(1),SDFTT3(1),SDFTT4(1)),
     3 (ZZ( 505),       FT11(1), FT12(1), FT13(1), FT14(1)),
     4 (ZZ(1045),       FT21(1), FT22(1), FT23(1), FT24(1)),
     5 (ZZ(1585),       FT31(1), FT32(1), FT33(1), FT34(1))
C
C               OUTPUT FORMATS
C
 2014 FORMAT ( 34X, 5H PA = F5.2)
 2032 FORMAT (1H1, 30X, 20H TRANSFORMED THRUST / 9H TIME NO.I2)
 2033 FORMAT (1H1, 30X, 20H TRANSFORMED IMPULSE/ 9H TIME NO.I2)
 2034 FORMAT (1H1, 30X, 30H TRANSFORMED SPECIFIC IMPULSE /9H TIME NO.I2)
 2037 FORMAT (1H1, 30X, 28H TRANSFORMED PROPELLANT WT. / 9H TIME NO.I2)
 2041 FORMAT (11H0TEMP GROUP I2 // (1X,A6, 19X, 4F12.2))
 2047 FORMAT ( 30X, 43H  TEMP1        TEMP2        TEMP3        TEMP4    /
     1            10H MOTOR NO.)
 2062 FORMAT (34H0CONFIDENCE ON NORMAL DISTRIBUTION /
     1          26H (ONE    MIN                        4F12.2 /
     2          26H SIDED   MAX                        4F12.2 //
     3          26H (TWO    MIN                        4F12.2 /
     4          26H SIDED   MAX                        4F12.2 )
 2064 FORMAT (26H0MEAN                                 4F12.2 /
     1          26H STANDARD DEV.                      4F12.2 )
 2065 FORMAT (26H0TOTAL MEAN                           4F12.2 /
     1          26H TOTAL STANDARD DEV.                4F12.2 )
C
      I1 = 1
      I2 = 2
      I3 = 3
      BLANK = 0.
      J1 = COUNT(1)
      J2 = COUNT(2)
      J3 = COUNT(3)
      DO 696  I = 1,504
  696 ZZ(I) = 0.
      C1MIN(4) = 0.
      C1MAX(4) = 0.
      C2MIN (4) = 0.
      C2MAX(4) = 0.
      DO 701  J = 1,NT
      DO 697  K = 1,J1
  697 VEC1(J,K) = T1(NBURN,J,K)
```

```
      DO 698  K = 1,J2
  698 VEC2(J,K) = T2(NBURN,J,K)
      DO 699  K = 1,J3
  699 VEC3(J,K) = T3(NBURN,J,K)
  701 CONTINUE
C
C
C           COMPUTE THRUST, IMPULSE AND THEIR STATISTICS FOR P1
C
      DO 752 I = 1,NTIMES
      DO 752  J = 1,NT
      L = 0
      M = 0
      DO 705  K = 1,J1
      L = L + 1
      M = M + 1
      RATIO1(I,J,K) = P1(I,J,K) / P (I,1,K)
      FT11(I,J,K) = RATIO1(I,J,K)*FVAC1(I,1,K)-EXPA(1)*AETOT(1,K)*CPHI
      IT11(I,J,K) = FT11(I,J,K) * T1(I,J,K)
      TENTRY(L)= FT11(I,J,K)
      TENT2(L) = IT11(I,J,K)
      SET1 (M) = FT11(I,J,K)
  705 SET4 (M) = IT11(I,J,K)
      M = 0
      DO 708  K = 1,J2
      L = L + 1
      M = M + 1
      RATIO2(I,J,K) = P2(I,J,K) / P (I,2,K)
      FT21(I,J,K) = RATIO2(I,J,K)*FVAC1(I,2,K)-EXPA(1)*AETOT(2,K)*CPHI
      IT21(I,J,K) = FT21(I,J,K) * T2(I,J,K)
      TENTRY(L)= FT21(I,J,K)
      TENT2(L) = IT21(I,J,K)
      SET2 (M) = FT21(I,J,K)
  708 SET5 (M) = IT21(I,J,K)
      M = 0
      DO 710  K = 1,J3
      L = L + 1
      M = M + 1
      RATIO3(I,J,K) = P3(I,J,K) / P (I,3,K)
      FT31(I,J,K) = RATIO3(I,J,K)*FVAC1(I,3,K)-EXPA(1)*AETOT(3,K)*CPHI
      IT31(I,J,K) = FT31(I,J,K) * T3(I,J,K)
      TENTRY(L)= FT31(I,J,K)
      TENT2(L) = IT31(I,J,K)
      SET3 (M) = FT31(I,J,K)
  710 SET6 (M) = IT31(I,J,K)
      CALL STATS (SET1, J1, MFT11(I,J), SDFT11(I,J))
      CALL STATS (SET4, J1, MIT11(I,J), SDIT11(I,J))
      CALL STATS (SET2, J2, MFT21(I,J), SDFT21(I,J))
      CALL STATS (SET5, J2, MIT21(I,J), SDIT21(I,J))
      CALL STATS (SET3, J3, MFT31(I,J), SDFT31(I,J))
      CALL STATS (SET6, J3, MIT31(I,J), SDIT31(I,J))
      CALL STATS (TENTRY,L, MFTT1(I,J), SDFTT1(I,J))
      CALL STATS (TENT2 ,L, MITT1(I,J), SDITT1(I,J))
  752 CONTINUE
C
C               PRINT THRUST, IMPULSE AND STATS FOR P1
C
      DO 1710 I = 1,NTIMES
      WRITE (6,2032) I
      WRITE (6,2014) EXPA(1)
      WRITE (6,2047)
      WRITE (6,2041) I1, (MOTNO(I1,K), FT11(I,1,K), FT11(I,2,K),
```

```
     1                                  FT11(I,3,K), BLANK, K=1,J1)
      WRITE (6,2064) (MFT11(I,J), J = 1,4), (SDFT11(I,J), J = 1,4)
      IF (NT .LE. 1) GO TO 1708
      WRITE (6,2041) I2, (MOTNO(I2,K), FT21(I,1,K), FT21(I,2,K),
     1                                  FT21(I,3,K), BLANK, K=1,J2)
      WRITE (6,2064) (MFT21(I,J), J = 1,4), (SDFT21(I,J), J = 1,4)
      IF (NT .LE. 2) GO TO 1708
      WRITE (6,2041) I3, (MOTNO(I3,K), FT31(I,1,K), FT31(I,2,K),
     1                                  FT31(I,3,K), BLANK, K=1,J3)
      WRITE (6,2064) (MFT31(I,J), J = 1,4), (SDFT31(I,J), J = 1,4)
 1708 WRITE (6,2065) (MFTT1(I,J), J = 1,4), (SDFTT1(I,J), J = 1,4)
      DO 1709 J = 1,NT
      C1MIN(J) = MFTT1(I,J) - CT1*SDFTT1(I,J)
      C1MAX(J) = MFTT1(I,J) + CT1*SDFTT1(I,J)
      C2MIN(J) = MFTT1(I,J) - CT2*SDFTT1(I,J)
 1709 C2MAX(J) = MFTT1(I,J) + CT2*SDFTT1(I,J)
      WRITE (6,2062) C1MIN, C1MAX, C2MIN, C2MAX
 1710 CONTINUE
      DO 1712 I = 1,NTIMES
      WRITE (6,2033) I
      WRITE (6,2014) EXPA(1)
      WRITE (6,2047)
      WRITE (6,2041) I1, (MOTNO(I1,K), IT11(I,1,K), IT11(I,2,K),
     1                                  IT11(I,3,K), BLANK, K=1,J1)
      WRITE (6,2064) (MIT11(I,J), J = 1,4), (SDIT11(I,J), J = 1,4)
      IF (NT .LE. 1) GO TO 1707
      WRITE (6,2041) I2, (MOTNO(I2,K), IT21(I,1,K), IT21(I,2,K),
     1                                  IT21(I,3,K), BLANK, K=1,J2)
      WRITE (6,2064) (MIT21(I,J), J = 1,4), (SDIT21(I,J), J = 1,4)
      IF (NT .LE. 2) GO TO 1707
      WRITE (6,2041) I3, (MOTNO(I3,K), IT31(I,1,K), IT31(I,2,K),
     1                                  IT31(I,3,K), BLANK, K=1,J3)
      WRITE (6,2064) (MIT31(I,J), J = 1,4), (SDIT31(I,J), J = 1,4)
 1707 WRITE (6,2065) (MITT1(I,J), J = 1,4), (SDITT1(I,J), J = 1,4)
      DO 1711 J = 1,NT
      C1MIN(J) = MITT1(I,J) - CT1*SDITT1(I,J)
      C1MAX(J) = MITT1(I,J) + CT1*SDITT1(I,J)
      C2MIN(J) = MITT1(I,J) - CT2*SDITT1(I,J)
 1711 C2MAX(J) = MITT1(I,J) + CT2*SDITT1(I,J)
      WRITE (6,2062) C1MIN, C1MAX, C2MIN, C2MAX
 1712 CONTINUE
C
C              COMPUTE TRANSFORMED PROPELLANT WEIGHT, SPECIFIC IMPULSE
C              FOR P1 AND THEIR STATISTICS
C
      DO 882  I = 1,NTIMES
      DO 882 J = 1,NT
      L = 0
      M = 0
      DO 804  K = 1,J1
      L = L + 1
      M = M + 1
      WP1(I,J,K) = WP(1,K) * INTPT1(I,J,K) / INTPT1(NTOT,J,K)
      ISP11(I,J,K) = IT11(I,J,K) / WP1(I,J,K)
      SAVE1(I,J,K) = ISP11(I,J,K)
      TENT2(L) = ISP11(I,J,K)
      TENTRY(L)=WP1(I,J,K)
      SET4(M) = WP1(I,J,K)
  804 SET1 (M) = ISP11(I,J,K)
      M = 0
      DO 806  K = 1,J2
      L = L + 1
```

```
      M = M + 1
      WP2(I,J,K) = WP(2,K) * INTPT2(I,J,K) / INTPT2(NTOT,J,K)
      ISP21(I,J,K) = IT21(I,J,K) / WP2(I,J,K)
      SAVE2(I,J,K) = ISP21(I,J,K)
      TENT2(L) = ISP21(I,J,K)
      TENTRY(L)=WP2(I,J,K)
      SET5(M) = WP2(I,J,K)
  806 SET2 (M) = ISP21(I,J,K)
      M = 0
      DO 808  K = 1,J3
      L = L + 1
      M = M + 1
      WP3(I,J,K) = WP(3,K) * INTPT3(I,J,K) / INTPT3(NTOT,J,K)
      ISP31(I,J,K) = IT31(I,J,K) / WP3(I,J,K)
      SAVE3(I,J,K) = ISP31(I,J,K)
      TENT2(L) = ISP31(I,J,K)
      TENTRY(L)=WP3(I,J,K)
      SET6(M) = WP3(I,J,K)
  808 SET3 (M) = ISP31(I,J,K)
      CALL STATS (SET1, J1, MIS11(I,J), SDIS11(I,J))
      CALL STATS (SET2, J2, MIS21(I,J), SDIS21(I,J))
      CALL STATS (SET3, J3, MIS31(I,J), SDIS31(I,J))
      CALL STATS (SET4, J1, MWP1(I,J), SDWP1(I,J))
      CALL STATS (SET5, J2, MWP2(I,J), SDWP2(I,J))
      CALL STATS (SET6, J3, MWP3(I,J), SDWP3(I,J))
      CALL STATS (TENT2, L, MIST1(I,J), SDIST1(I,J))
      CALL STATS (TENTRY,L, MWPT(I,J), SDWPT(I,J))
  882 CONTINUE
C
C               PRINT PROPELLANT WT. AND SPECIFIC IMPULSE FOR P1
C
      DO 812  I = 1,NTIMES
      WRITE (6,2037) I
      WRITE (6,2047)
      WRITE (6,2041) I1, (MOTNO(I1,K), WP1(I,1,K), WP1(I,2,K),
     1                                 WP1(I,3,K), BLANK, K = 1,J1)
      WRITE (6,2064) (MWP1(I,J), J=1,4), (SDWP1(I,J), J=1,4)
      IF (NT .LE. 1) GO TO 810
      WRITE (6,2041) I2, (MOTNO(I2,K), WP2(I,1,K), WP2(I,2,K),
     1                                 WP2(I,3,K), BLANK, K = 1,J2)
      WRITE (6,2064) (MWP2(I,J), J=1,4), (SDWP2(I,J), J=1,4)
      IF (NT .LE. 2) GO TO 810
      WRITE (6,2041) I3, (MOTNO(I3,K), WP3(I,1,K), WP3(I,2,K),
     1                                 WP3(I,3,K), BLANK, K = 1,J3)
      WRITE (6,2064) (MWP3(I,J), J=1,4), (SDWP3(I,J), J=1,4)
  810 WRITE (6,2065) (MWPT(I,J), J=1,4), (SDWPT(I,J), J=1,4)
      DO  811 J = 1,NT
      C1MIN(J) = MWPT(I,J) - CT1*SDWPT(I,J)
      C1MAX(J) = MWPT(I,J) + CT1*SDWPT(I,J)
      C2MIN(J) = MWPT(I,J) - CT2*SDWPT(I,J)
  811 C2MAX(J) = MWPT(I,J) + CT2*SDWPT(I,J)
      WRITE (6,2062) C1MIN, C1MAX, C2MIN, C2MAX
  812 CONTINUE
      DO 1775 I = 1,NTIMES
      WRITE (6,2034) I
       WRITE (6,2014) EXPA(1)
      WRITE (6,2047)
      WRITE (6,2041)I1,(MOTNO(I1,K), ISP11(I,1,K), ISP11(I,2,K),
     1                               ISP11(I,3,K), BLANK, K=1,J1)
      WRITE (6,2064) (MIS11(I,J), J=1,4), (SDIS11(I,J), J=1,4)
      IF (NT .LE. 1) GO TO 1773
      WRITE (6,2041)I2,(MOTNO(I2,K), ISP21(I,1,K), ISP21(I,2,K),
```

```
     1                                   ISP21(I,3,K), BLANK, K = 1,J2)
      WRITE (6,2064) (MIS21(I,J), J=1,4), (SDIS21(I,J), J=1,4)
      IF (NT .LE. 2) GO TO 1773
      WRITE (6,2041)I3,(MOTNO(I3,K), ISP31(I,1,K), ISP31(I,2,K),
     1                                   ISP31(I,3,K), BLANK, K = 1,J3)
      WRITE (6,2064) (MIS31(I,J), J=1,4), (SDIS31(I,J), J=1,4)
 1773 WRITE (6,2065) (MIST1(I,J), J=1,4), (SDIST1(I,J), J=1,4)
      DO 1774 J = 1,NT
      C1MIN(J) = MIST1(I,J) - CT1*SDIST1(I,J)
      C1MAX(J) = MIST1(I,J) + CT1*SDIST1(I,J)
      C2MIN(J) = MIST1(I,J) - CT2*SDIST1(I,J)
 1774 C2MAX(J) = MIST1(I,J) + CT2*SDIST1(I,J)
      WRITE (6,2062) C1MIN, C1MAX, C2MIN, C2MAX
 1775 CONTINUE
      IF (NP .LE. 1) GO TO 835
C
C           COMPUTE THRUST AND IMPULSE FOR P2 AND THEIR STATISTICS
C
      DO 1785 N = 2, NP
      DO 767 I = 1,NTIMES
      DO 767 J = 1,NT
      L = 0
      M = 0
      DO 712  K = 1,J1
      L = L + 1
      M = M + 1
      SAVE = FT11(I,J,K)
      FT12(I,J,K) = RATIO1(I,J,K)*FVAC1(I,1,K)-FXPA(N)*AETOT(1,K)*CPHI
      IT12(I,J,K) = SAVE1(I,J,K)*WP1(I,J,K)*FT12(I,J,K) / SAVE
      TENTRY(L)= FT12(I,J,K)
      TENT2(L) = IT12(I,J,K)
      SET1 (M) = FT12(I,J,K)
  712 SET4 (M) = IT12(I,J,K)
      M = 0
      DO 715  K = 1,J2
      L = L + 1
      M = M + 1
      SAVE = FT21(I,J,K)
      FT22(I,J,K) = RATIO2(I,J,K)*FVAC1(I,2,K)-EXPA(N)*AETOT(2,K)*CPHI
      IT22(I,J,K) = SAVE2(I,J,K)*WP2(I,J,K)*FT22(I,J,K) / SAVE
      TENTRY(L)= FT22(I,J,K)
      TENT2(L) = IT22(I,J,K)
      SET2 (M) = FT22(I,J,K)
  715 SET5 (M) = IT22(I,J,K)
      M = 0
      DO 718  K = 1,J3
      L = L + 1
      M = M + 1
      SAVE = FT31(I,J,K)
      FT32(I,J,K) = RATIO3(I,J,K)*FVAC1(I,3,K)-EXPA(N)*AETOT(3,K)*CPHI
      IT32(I,J,K) = SAVE3(I,J,K)*WP3(I,J,K)*FT32(I,J,K) / SAVE
      TENTRY(L)= FT32(I,J,K)
      TENT2(L) = IT32(I,J,K)
      SET3 (M) = FT32(I,J,K)
  718 SET6 (M) = IT32(I,J,K)
      CALL STATS (SET1, J1, MFT12(I,J), SDFT12(I,J))
      CALL STATS (SET4, J1, MIT12(I,J), SDIT12(I,J))
      CALL STATS (SET2, J2, MFT22(I,J), SDFT22(I,J))
      CALL STATS (SET5, J2, MIT22(I,J), SDIT22(I,J))
      CALL STATS (SET3, J3, MFT32(I,J), SDFT32(I,J))
      CALL STATS (SET6, J3, MIT32(I,J), SDIT32(I,J))
      CALL STATS (TENTRY,L, MFTT2(I,J), SDFTT2(I,J))
```

```
      CALL STATS (TENT2 ,L, MITT2(I,J), SDITT2(I,J))
  767 CONTINUE
C
C                 PRINT THRUST, IMPULSE AND STATS FOR P2
C
      DO 1718 I = 1,NTIMES
      WRITE (6,2032) I
      WRITE (6,2014) EXPA(N)
      WRITE (6,2047)
      WRITE (6,2041) I1, (MOTNO(I1,K), FT12(I,1,K), FT12(I,2,K),
     1                                 FT12(I,3,K), BLANK, K=1,J1)
      WRITE (6,2064) (MFT12(I,J), J = 1,4), (SDFT12(I,J), J = 1,4)
      IF (NT .LE. 1) GO TO 1716
      WRITE (6,2041) I2, (MOTNO(I2,K), FT22(I,1,K), FT22(I,2,K),
     1                                 FT22(I,3,K), BLANK, K=1,J2)
      WRITE (6,2064) (MFT22(I,J), J = 1,4), (SDFT22(I,J), J = 1,4)
      IF (NT .LE. 2) GO TO 1716
      WRITE (6,2041) I3, (MOTNO(I3,K), FT32(I,1,K), FT32(I,2,K),
     1                                 FT32(I,3,K), BLANK, K=1,J3)
      WRITE (6,2064) (MFT32(I,J), J = 1,4), (SDFT32(I,J), J = 1,4)
 1716 WRITE (6,2065) (MFTT2(I,J), J = 1,4), (SDFTT2(I,J), J = 1,4)
      DO 1717 J = 1,NT
      C1MIN(J) = MFTT2(I,J) - CT1*SDFTT2(I,J)
      C1MAX(J) = MFTT2(I,J) + CT1*SDFTT2(I,J)
      C2MIN(J) = MFTT2(I,J) - CT2*SDFTT2(I,J)
 1717 C2MAX(J) = MFTT2(I,J) + CT2*SDFTT2(I,J)
      WRITE (6,2062) C1MIN, C1MAX, C2MIN, C2MAX
 1718 CONTINUE
      DO 1720 I = 1,NTIMES
      WRITE (6,2033) I
      WRITE (6,2014) EXPA(N)
      WRITE (6,2047)
      WRITE (6,2041) I1, (MOTNO(I1,K), IT12(I,1,K), IT12(I,2,K),
     1                                 IT12(I,3,K), BLANK, K=1,J1)
      WRITE (6,2064) (MIT12(I,J), J = 1,4), (SDIT12(I,J), J = 1,4)
      IF (NT .LE. 1) GO TO 1715
      WRITE (6,2041) I2, (MOTNO(I2,K), IT22(I,1,K), IT22(I,2,K),
     1                                 IT22(I,3,K), BLANK, K=1,J2)
      WRITE (6,2064) (MIT22(I,J), J = 1,4), (SDIT22(I,J), J = 1,4)
      IF (NT .LE. 2) GO TO 1715
      WRITE (6,2041) I3, (MOTNO(I3,K), IT32(I,1,K), IT32(I,2,K),
     1                                 IT32(I,3,K), BLANK, K=1,J3)
      WRITE (6,2064) (MIT32(I,J), J = 1,4), (SDIT32(I,J), J = 1,4)
 1715 WRITE (6,2065) (MITT2(I,J), J = 1,4), (SDITT2(I,J), J = 1,4)
      DO 1719 J = 1,NT
      C1MIN(J) = MITT2(I,J) - CT1*SDITT2(I,J)
      C1MAX(J) = MITT2(I,J) + CT1*SDITT2(I,J)
      C2MIN(J) = MITT2(I,J) - CT2*SDITT2(I,J)
 1719 C2MAX(J) = MITT2(I,J) + CT2*SDITT2(I,J)
      WRITE (6,2062) C1MIN, C1MAX, C2MIN, C2MAX
 1720 CONTINUE
C
C             COMPUTE SPECIFIC IMPULSE AND ITS STATISTICS FOR P2
C
      DO 897  I = 1,NTIMES
      DO 897  J = 1,NT
      L = 0
      M = 0
      DO 1810 K = 1,J1
      L = L + 1
      M = M + 1
      ISP12(I,J,K) = IT12(I,J,K) / WP1(I,J,K)
```

```
      SAVE1(I,J,K) = ISP12(I,J,K)
      TENT2(L) = ISP12(I,J,K)
 1810 SET1 (M) = ISP12(I,J,K)
      M = 0
      DO 813  K = 1,J2
      L = L + 1
      M = M + 1
      ISP22(I,J,K) = IT22(I,J,K) / WP2(I,J,K)
      SAVE2(I,J,K) = ISP22(I,J,K)
      TENT2(L) = ISP22(I,J,K)
  813 SET2 (M) = ISP22(I,J,K)
      M = 0
      DO 814  K = 1,J3
      L = L + 1
      M = M + 1
      ISP32(I,J,K) = IT32(I,J,K) / WP3(I,J,K)
      SAVE3(I,J,K) = ISP32(I,J,K)
      TENT2(L) = ISP32(I,J,K)
  814 SET3 (M) = ISP32(I,J,K)
      CALL STATS (SET1, J1, MIS12(I,J), SDIS12(I,J))
      CALL STATS (SET2, J2, MIS22(I,J), SDIS22(I,J))
      CALL STATS (SET3, J3, MIS32(I,J), SDIS32(I,J))
      CALL STATS (TENT2, L, MIST2(I,J), SDIST2(I,J))
  897 CONTINUE
C
C                 PRINT SPECIFIC IMPULSE FOR P2
C
      DO 1780 I = 1,NTIMES
      WRITE (6,2034) I
      WRITE (6,2014) EXPA(N)
      WRITE (6,2047)
      WRITE (6,2041)I1,(MOTNO(I1,K), ISP12(I,1,K), ISP12(I,2,K),
     1                              ISP12(I,3,K), BLANK, K = 1,J1)
      WRITE (6,2064) (MIS12(I,J), J=1,4), (SDIS12(I,J), J=1,4)
      IF (NT .LE. 1) GO TO 1778
      WRITE (6,2041)I2,(MOTNO(I2,K), ISP22(I,1,K), ISP22(I,2,K),
     1                              ISP22(I,3,K), BLANK, K = 1,J2)
      WRITE (6,2064) (MIS22(I,J), J=1,4), (SDIS22(I,J), J=1,4)
      IF (NT .LE. 2) GO TO 1778
      WRITE (6,2041)I3,(MOTNO(I3,K), ISP32(I,1,K), ISP32(I,2,K),
     1                              ISP32(I,3,K), BLANK, K = 1,J3)
      WRITE (6,2064) (MIS32(I,J), J=1,4), (SDIS32(I,J), J=1,4)
 1778 WRITE (6,2065) (MIST2(I,J), J=1,4), (SDIST2(I,J), J=1,4)
      DO 1779 J = 1,NT
      C1MIN(J) = MIST2(I,J) - CT1*SDIST2(I,J)
      C1MAX(J) = MIST2(I,J) + CT1*SDIST2(I,J)
      C2MIN(J) = MIST2(I,J) - CT2*SDIST2(I,J)
 1779 C2MAX(J) = MIST2(I,J) + CT2*SDIST2(I,J)
      WRITE (6,2062) C1MIN, C1MAX, C2MIN, C2MAX
 1780 CONTINUE
      IF (NP .LE. 2) GO TO 835
 1785 CONTINUE
  835 CONTINUE
      RETURN
      END
```

```
$IBFTC MOT3
      SUBROUTINE MOT3
      DIMENSION XX(4860)
      COMMON G, CPHI, NT, EXPA, COUNT, WP, WEB, ATTOT, AETOT,
     1       P, MOTNO, FVAC1, NTIMES, XX, NTOT, NBURN, NFIRST, NP
      COMMON CP1, CP2, CT1, CT2, VEC1, VEC2, VEC3
      DIMENSION        WP1   (9,4,15), WP2   (9,4,15), WP3   (9,4,15)
      DIMENSION        CST1  (9,4,15), CST2  (9,4,15), CST3  (9,4,15),
     1 FVAC1 (9,4,15), INTPT1(9,4,15), INTPT2(9,4,15), INTPT3(9,4,15),
     2 P     (9,4,15), ATTOT   (4,15), AETOT   (4,15), WP      (4,15),
     3 WEB     (4,15), VEC1    (4,15), VEC2    (4,15), VEC3    (4,15),
     4 R1      (4,15), R2      (4,15), R3      (4,15), MOTNO   (4,15)
      DIMENSION    EXPA  ( 4), C1MIN(4), C1MAX(4), C2MIN( 4), C2MAX( 4),
     1 C1TMN ( 4), C1TMX ( 4), C2TMN(4), C2TMX(4), SET1 (15), SET2 (15),
     2 SET3  (15), TENTRY(60), TENT2(60),MR1 ( 4), MR2  ( 4), MR3  ( 4),
     3 MRT   ( 4), SDR1  ( 4), SDR2 ( 4), SDR3( 4), SDRT ( 4), COUNT(4)
      DIMENSION    MCST1 (9,4), MCST2(9,4), MCST3(9,4), MCSTT(9,4),
     1             SDCST1 (9,4),SDCST2(9,4),SDCST3(9,4),SDCSTT(9,4)
      EQUIVALENCE
     1 (XX(   1), WP1(1), CST1(1)),
     2 (XX( 541), WP2(1), CST2(1)),
     3 (XX(1081), WP3(1), CST3(1)),
     4 (XX(3241), INTPT1(1)),
     5 (XX(3781), INTPT2(1)),
     6 (XX(4321), INTPT3(1))
      REAL
     1 MRT   , MCST1 , MCST2 , MCST3 , MCSTT , MR1   , MR2   , MR3   ,
     2 MOTNO, INTPT1, INTPT2, INTPT3
      INTEGER COUNT
C             OUTPUT FORMATS
 2035 FORMAT (1H1, 30X, 36H TRANSFORMED CHARACTERISTIC VELOCITY /
    1                   9H TIME NO.I2)
 2036 FORMAT (1H1, 30X, 36H TRANSFORMED CHARACTERISTIC VELOCITY, 14X,
    1                  33H TRANSFORMED AVERAGE BURNING RATE /
    2                   9H TIME NO.I2, 70X, 18H BASED ON WEB TIME )
 2040 FORMAT (11H0TEMP GROUP I2 // (1X,A6, 19X, 4F10.2, 10X, 4F10.4))
 2041 FORMAT (11H0TEMP GROUP I2 // (1X,A6, 19X, 4F10.2))
 2047 FORMAT ( 29X, 36H TEMP1     TEMP2     TEMP3     TEMP4   /
    1             10H MOTOR NO.)
 2048 FORMAT ( 29X, 36H TEMP1     TEMP2     TEMP3     TEMP4   ,17X,
    1              36H TEMP1     TEMP2     TEMP3     TEMP4 /
    2             10H MOTOR NO.)
 2062 FORMAT (34H0CONFIDENCE ON NORMAL DISTRIBUTION /
    1     26H (ONE   MIN                      4F10.2/
    2     26H SIDED  MAX                      4F10.2 //
    3     26H (TWO   MIN                      4F10.2 /
    4     26H SIDED  MAX                      4F10.2 )
 2064 FORMAT (26H0MEAN                            4F10.2 /
    1     26H STANDARD DEV.                   4F10.2)
 2065 FORMAT (26H0TOTAL MEAN                      4F10.2 /
    1     26H TOTAL STANDARD DEV.             4F10.2 )
 2066 FORMAT (26H0MEAN                            4F10.2, 10X, 4F10.4 /
    1     26H STANDARD DEV.                   4F10.2, 10X, 4F10.4 )
 2067 FORMAT (26H0TOTAL MEAN                      4F10.2, 10X, 4F10.4 /
    1     26H TOTAL STANDARD DEV.             4F10.2, 10X, 4F10.4 )
 2068 FORMAT (34H0CONFIDENCE ON NORMAL DISTRIBUTION /
    1     26H (ONE  MIN                       4F10.2,10X,4F10.4/
    2     26H SIDED) MAX                      4F10.2,10X,4F10.4 //
    3     26H (TWO   MIN                      4F10.2,10X,4F10.4 /
    4     26H SIDED) MAX                      4F10.2,10X,4F10.4 )
C
```

```
C
      I1 = 1
      I2 = 2
      I3 = 3
      J1 = COUNT(1)
      J2 = COUNT(2)
      J3 = COUNT(3)
      SDR1(4) = 0.
      MCSTT(5,4) = 0.
      BLANK = 0.
C
C                COMPUTE R TRANSFORMED AND ITS STATISTICS
C
      DO 1055 J = 1,NT
      L = 0
      M = 0
      DO 1010 K = 1,J1
      L = L + 1
      M = M + 1
      R1(J,K) = WEB(1,K) / VEC1(J,K)
      TENT2(L) = R1(J,K)
 1010 SET1 (M) = R1(J,K)
      M = 0
      DO 1015 K = 1,J2
      L = L + 1
      M = M + 1
      R2(J,K) = WEB(2,K) / VEC2(J,K)
      TENT2(L) = R2(J,K)
 1015 SET2 (M) = R2(J,K)
      M = 0
      DO 1020 K = 1,J3
      L = L + 1
      M = M + 1
      R3(J,K) = WEB(3,K) / VEC3(J,K)
      TENT2(L) = R3(J,K)
 1020 SET3 (M) = R3(J,K)
      CALL STATS (SET1, J1, MR1(J), SDR1(J))
      CALL STATS (SET2, J2, MR2(J), SDR2(J))
      CALL STATS (SET3, J3, MR3(J), SDR3(J))
      CALL STATS (TENT2, L, MRT(J), SDRT(J))
 1055 CONTINUE
C
C                COMPUTE C-STAR AND ITS STATISTICS
C
      DO 1000 I = NBURN, NTOT
      DO 1000 J = 1,NT
      L = 0
      M = 0
      DO 950  K = 1,J1
      L = L + 1
      M = M + 1
      CST1(I,J,K) = INTPT1(NTOT,J,K) * G * ATTOT(1,K) / WP (1,K)
      TENT2(L) = CST1(I,J,K)
  950 SET1 (M) = CST1(I,J,K)
      M = 0
      DO 955  K = 1,J2
      L = L + 1
      M = M + 1
      CST2(I,J,K) = INTPT2(NTOT,J,K) * G* ATTOT(2,K) / WP (2,K)
      TENT2(L) = CST2(I,J,K)
  955 SET2 (M) = CST2(I,J,K)
      M = 0
```

```
      DO 960  K = 1,J3
      L = L + 1
      M = M + 1
      CST3(I,J,K) = INTPT3(NTOT,J,K) * G * ATTOT(3,K) / WP(3,K)
      TENT2(L) = CST3(I,J,K)
  960 SET3 (M) = CST3(I,J,K)
      CALL STATS (SET1, J1, MCST1(I,J), SDCST1(I,J))
      CALL STATS (SET2, J2, MCST2(I,J), SDCST2(I,J))
      CALL STATS (SET3, J3, MCST3(I,J), SDCST3(I,J))
      CALL STATS (TENT2, L, MCSTT(I,J), SDCSTT(I,J))
 1000 CONTINUE
      DO 1795 I = NBURN, NTOT
      DO 1785 J = 1,NT
      C1MIN(J) = MCSTT(I,J) - CT1*SDCSTT(I,J)
      C1MAX(J) = MCSTT(I,J) + CT1*SDCSTT(I,J)
      C2MIN(J) = MCSTT(I,J) - CT2*SDCSTT(I,J)
 1785 C2MAX(J) = MCSTT(I,J) + CT2*SDCSTT(I,J)
      IF (I .EQ. NBURN) GO TO 1786
      WRITE (6,2035) I
      WRITE (6,2047)
      GO TO 1788
 1786 WRITE (6,2036) I
      WRITE (6,2048)
 1788 IF (I .EQ. NBURN) GO TO 1791
      WRITE (6,2041)I1,(MOTNO(I1,K), CST1(I,1,K), CST1(I,2,K),
     1 CST1(I,3,K), BLANK, K = 1,J1)
      WRITE (6,2064) (MCST1(I,J), J=1,4), (SDCST1(I,J), J=1,4)
      WRITE (6,2041)I2,(MOTNO(I2,K), CST2(I,1,K), CST2(I,2,K),
     1 CST2(I,3,K), BLANK, K = 1,J2)
      WRITE (6,2064) (MCST2(I,J), J=1,4), (SDCST2(I,J), J=1,4)
      WRITE (6,2041)I3,(MOTNO(I3,K), CST3(I,1,K), CST3(I,2,K),
     1 CST3(I,3,K), BLANK, K = 1,J3)
      WRITE (6,2064) (MCST3(I,J), J=1,4), (SDCST3(I,J), J=1,4)
      WRITE (6,2065) (MCSTT(I,J), J=1,4), (SDCSTT(I,J), J=1,4)
      WRITE (6,2062) C1MIN, C1MAX, C2MIN, C2MAX
      GO TO 1795
 1791 WRITE (6,2040)I1,(MOTNO(I1,K), CST1(I,1,K), CST1(I,2,K), CST1(I,3,
     1 K), BLANK, R1(1,K), R1(2,K), R1(3,K), BLANK, K = 1,J1)
      WRITE (6,2066) (MCST1(I,J), J=1,4), MR1, (SDCST1(I,J), J=1,4),SDR1
      WRITE (6,2040)I2,(MOTNO(I2,K), CST2(I,1,K), CST2(I,2,K), CST2(I,3,
     1 K), BLANK, R2(1,K), R2(2,K), R2(3,K), BLANK, K = 1,J2)
      WRITE (6,2066) (MCST2(I,J), J=1,4), MR2, (SDCST2(I,J), J=1,4),SDR2
      WRITE (6,2040)I3,(MOTNO(I3,K), CST3(I,1,K), CST3(I,2,K), CST3(I,3,
     1 K), BLANK, R3(1,K), R3(2,K), R3(3,K), BLANK, K = 1,J3)
      WRITE (6,2066) (MCST3(I,J), J=1,4), MR3, (SDCST3(I,J), J=1,4),SDR3
      WRITE (6,2067) (MCSTT(I,J), J=1,4), MRT, (SDCSTT(I,J), J=1,4),SDRT
      DO 1794 J = 1,NT
      C1TMN(J) = MRT(J) - CT1*SDRT(J)
      C1TMX(J) = MRT(J) + CT1*SDRT(J)
      C2TMN(J) = MRT(J) - CT2*SDRT(J)
 1794 C2TMX(J) = MRT(J) + CT2*SDRT(J)
      WRITE (6,2068) C1MIN, C1TMN, C1MAX, C1TMX,
     1                C2MIN, C2TMN, C2MAX, C2TMX
 1795 CONTINUE
      RETURN
      END
```

APPENDIX B

LISTING OF THE SOLID PROPELLANT ROCKET MOTOR PERFORMANCE VERSUS TIME COMPUTER PROGRAM USING THE GROUP TRANSFORMATION METHOD

```
•P FOR MAIN
C          GENERAL SOLID-PROPELLANT ROCKET MOTOR PERFORMANCE VS. TIME
C
      DIMENSION  XX(1700)
      DIMENSION  XLNP (3,15,200), FSAVE(3,15,200), PSAVE(3,15,200)
      DIMENSION  CC(200,4),              PCO(200),   FO(200),    Y(200),
     1           WPRNT(200),  TREP(200),  PC (200),    T(200),    F(200)
      DIMENSION  AE1(3,15),  AE2(3,15), AE3(3,15), AE4(3,15), WEB(3,15),
     1         XMOTNO(3,15), TEMP(3,15), PAF(3,15),TAIL(3,15),AETOT(3,15)
      DIMENSION   TGRP( 4),  EXTEM( 3),  EXPA( 3), KOUNT( 4),   EPS( 4)
      DIMENSION BCDX(12), BCD1(12), BCD2(12), BCD3(12), BCD4(12)
      DIMENSION  PCTW(10), NW(10), BREAKW(10), WINC(10)
      DIMENSION  PCTT( 5), NT(5), BREAKT( 5), TINC( 5)
      DIMENSION  NGO(10), NSTOP(10), XNW(10), XNT( 5), SET1(10)
      COMMON  XX
      COMMON XLNP, FSAVE, PSAVE
      COMMON  BCDX, BCD1, BCD2, BCD3, BCD4
      EQUIVALENCE
     1  (XX(    1),   NSV2),                       (XX(    3),    ICON),
     2  (XX(    4),     TAV), (XX(    5),     N1), (XX(    6), NPRIME),
     3  (XX(    7),    NPTS), (XX(    8),     TB), (XX(    9),     TT),
     4  (XX(   10),    WPCT), (XX(   11),   TPCT),
     5  (XX(   13),     CT2), (XX(   14),   TREP), (XX( 214),       T),
     6  (XX( 414),       CC), (XX(1214),    NSV1), (XX(1215),    NXNT),
     7  (XX(1216),       NA), (XX(1217),  KOUNT), (XX(1221),     TEMP),
     8  (XX(1266),    IPRNT), (XX(1267),      N2), (XX(1268),      TO),
     9  (XX(1269),       NP), (XX(1270),    EXPA), (XX(1273),   AETOT)
      EQUIVALENCE
     1  (XX(1318),      PAF), (XX(1363),    CPHI), (XX(1364),      TD),
     2  (XX(1365),    WPRNT), (XX(1565), XMOTNO), (XX(1601),     NXNW)
      EQUIVALENCE
     1  (XX(1602), BREAKW), (XX(1612), BREAKT), (XX(1617),     WINC),
     2  (XX(1627),     TINC), (XX(1632),     NGO), (XX(1642),    NSTOP),
     3  (XX(1652),     PCTW), (XX(1662),    PCTT)
C
C               INPUT FORMATS
C
   81 FORMAT (I12, 4F12.4)
   82 FORMAT (A6, 6F12.4, 1X, I1)
   83 FORMAT (2I6, 3F12.4,I6)
   84 FORMAT ( 5(F10.4, I6))
   85 FORMAT (4E12.5)
  100 FORMAT (3I6)
  101 FORMAT (2(E16.8, E12.5, E12.5))
  102 FORMAT (12A6)
C
C               OUTPUT FORMATS
C
  109 FORMAT (38H1THE FOLLOWING OUTPUT IS FOR MOTOR NO.  A6)
  110 FORMAT(54H0  PCT WEB TIME      TIME            PC          THRUST/)
  111 FORMAT (F12.4,3F14.4)
  112 FORMAT(54H0 PCT TAILOFF TIME   TIME            PC          THRUST/)
C
    1 READ (5, 83) NXNW, NXNT, TD, PHI, CT2,NSW
      IF(NSW .GT. 0)  READ (5,84)  PSIA
      WRITE(6, 83) NXNW, NXNT, TD, PHI, CT2
      READ (5, 84) (PCTW(I), NW(I), I = 1,NXNW)
      WRITE(6, 84) (PCTW(I), NW(I), I = 1,NXNW)
      READ (5, 84) (PCTT(I), NT(I), I = 1,NXNT)
      WRITE(6, 84) (PCTT(I), NT(I), I = 1,NXNT)
      READ (5, 81) NA, (TGRP (I), I = 1,NA)
```

```
      WRITE(6, 81) NA, (TGRP (I), I = 1,NA)
      READ (5, 81) NP, (EXPA (I), I = 1,NP)
      WRITE(6, 81) NP, (EXPA (I), I = 1,NP)
      READ (5,102) (BCDX(I),I=1,12)
      WRITE(6,102) (BCDX(I),I=1,12)
      READ (5,102) (BCD1(I),I=1,12)
      WRITE(6,102) (BCD1(I),I=1,12)
      READ (5,102) (BCD2(I),I=1,12)
      WRITE(6,102) (BCD2(I),I=1,12)
      READ (5,102) (BCD3(I),I=1,12)
      WRITE(6,102) (BCD3(I),I=1,12)
      READ (5,102) (BCD4(I),I=1,12)
      WRITE(6,102) (BCD4(I),I=1,12)
C
      KOUNT(1) = 0
      KOUNT(2) = 0
      KOUNT(3) = 0
      NGO(1)     = 1
      DO 4  I = 1,NXNW
      XNW(I)     = NW(I)
      WINC(I)    = 1. / XNW(I)
      NGO(I+1)   = NGO(I) + NW(I) + 1/I
      NSTOP(I)   = NGO(I+1) - 1
    4 CONTINUE
      DO 5  I = 1,NXNT
      XNT(I)     = NT(I)
      TINC(I)    = 1. / XNT(I)
      ISUB = (NXNW + I + 1)
      NGO(ISUB) = NGO(NXNW + I) + NT(I)
      NSTOP(NXNW+I) = NGO(ISUB) - 1
    5 CONTINUE
C
      IPASS = 0
      CPHI = COS (PHI)
      NSUM = NXNW + NXNT
      NSV0 = NSTOP(NXNW)
      N1   = NSV0 - 1
      NSV1 = NSV0 + 1
      NSV2 = NSTOP(NSUM)
      DO 7 I = 1,NXNW
    7 SET1(I) = PCTW(I) * .01
      DO 8 I = 1,NXNW
    8 PCTW(I+1) = SET1(I)
      PCTW(1) = 0.
      DO 9 I = 1,NXNT
    9 SET1(I) = PCTT(I) * .01
      DO 10 I = 1,NXNT
   10 PCTT(I+1) = SET1(I)
      PCTT(1) = 0.
C
C                 GROUP MOTORS INTO TEMPERATURE GROUPS
C
      TGRP(NA+1) = TGRP(NA) * 2.
   15 READ (5, 82) XMOT, FIRETP, PAX, TB, IT, TO,PCP, NDMOT
      WRITE(6, 82) XMOT, FIRETP, PAX, TB, IT, TO,PCP, NDMOT
      DO 19 J = 1, NA
      EPS(J) = (TGRP(J+1) - TGRP(J)) *.5
      IF ( (TGRP(J) + EPS(J)) - FIRETP) 19, 19, 25
   19 CONTINUE
   25 KOUNT(J) = KOUNT(J) + 1
      K = KOUNT(J)
      READ (5, 85) AE1(J,K), AE2(J,K), AE3(J,K), AE4(J,K)
```

```
      WRITE(6, 85) AE1(J,K), AE2(J,K), AE3(J,K), AE4(J,K)
      TEMP   (J,K) = FIRETP
      XMOTNO(J,K) = XMOT
      PAF    (J,K) = PAX
      WEB    (J,K) = TB
      TAIL   (J,K) = TT
      AETOT (J,K) = AE1(J,K) + AE2(J,K) + AE3(J,K) + AE4(J,K)
C
      READ (5,100) NPTS, ICON, IPRNT
      WRITE(6,100) NPTS, ICON, IPRNT
      READ (5,101) (TREP(I), PC(I), F(I), I=1,NPTS)
C
      DO 11    I=1,NPTS
   11 PC(I) = PC(I) + PCP
      PRES1 = 0.0
      PRES2 = 0.0
      IF(NSW .GT. 0)  GO TO 5005
5001  BREAKW(1) = TO
      DO 26  I = 1,NXNW
   26 BREAKW(I+1) = TO + PCTW(I+1)*TB
      BREAKT(1) = TB + TO
      DO 28  I = 1,NXNT
   28 BREAKT(I+1) = TB + PCTT(I+1)*(TT-TB) + TO
C
      IF( PRES1 .NE. 0.0)  GO TO 5002
5005  CALL CURVE (TREP, PC, NPTS)
      IF(NSW .LE. 0)  GO TO 5002
      TONEW = TREP(1) - .00001 - TAV
      DO 5003  I=1,100000
      TONEW = TONEW + .00001
      PRES1 = ((CC(1,4)*TONEW + CC(1,3)) *TONEW + CC(1,2))*TONEW+CC(1,1)
      PRES2 = CC(1,1)+ CC(1,2)*TONEW + CC(1,3)*TONEW**2
     1  + CC(1,4)*TONEW**3
      IF(PRES1 .GE. PSIA .OR. PRES2 .GE. PSIA)  GO TO 5004
5003  CONTINUE
5004  TO = TONEW + TAV
      WRITE(6,1002) TO,PSIA,PRES1,PRES2
1002  FORMAT (1H1/// 3HTO= F10.4,6X,5HPSIA= F10.3,6X,6HPRES1= F20.4,
     1  6X,6HPRES2= F20.4/////////)
      GO TO 5001
5002  CALL SRCH1(PCO)
      DO 30 L = 1, NSV2
      ARG = PCO(L)
      PSAVE(J,K,L) = ARG
   30 XLNP(J,K,L) = ALOG (ARG)
C
      CALL CURVE (TREP, F , NPTS)
      CALL SRCH1 (FO)
      DO 35 L = 1, NSV2
   35 FSAVE(J,K,L) = FO(L)
C
      WRITE (6, 109)  XMOTNO(J,K)
      WRITE (6, 110)
      DO 56  IJ = 1, NXNW
      AI = 0.
      NG = NGO(IJ)
      NSP= NSTOP(IJ)
      DELPW = PCTW(IJ+1) - PCTW(IJ)
      DO 55    I = NG, NSP
      IF (IJ .NE. 1) GO TO 53
      IF (I  .EQ. 1) GO TO 54
   53 AI = AI + 1
```

```
54 IF (IPASS .EQ. 1) GO TO 55
   WPRNT(I) =(PCTW(IJ) + WINC(IJ)*AI*DELPW) * 100.
55 WRITE (6, 111) WPRNT(I), T(I), PCO(I), FO(I)
56 CONTINUE
   WRITE (6, 112)
   DO 60  IJ = 1, NXNT
   AI = 0.
   NG = NGO   (IJ + NXNW)
   NSP = NSTOP(IJ + NXNW)
   DELPT = PCTT(IJ+1) - PCTT(IJ)
   DO 58    I = NG, NSP
   IF (IPASS .EQ. 1) GO TO 58
   AI = AI + 1.
   WPRNT(I) =(PCTT(IJ) + TINC(IJ)*AI*DELPT) * 100.
58 WRITE (6, 111) WPRNT(I), T(I), PCO(I), FO(I)
60 CONTINUE
   IPASS = 1
   IF (NDMOT) 65, 15, 65
65 CALL LSTSQ (    1, NSV0, WEB , 1)
   CALL LSTSQ (NSV1, NSV2, TAIL, 2)
   GO TO 1
   END
```

```
'IP FOR SRCH1
      SUBROUTINE SRCH1 (Y)
      DIMENSION XX(1700)
      DIMENSION  T(200), CC(200,4), TREP(200), Y(200)
      DIMENSION BREAKW(10), BREAKT(5),WINC(10),TINC(5),NGO(10),NSTOP(10)
      COMMON  XX
      EQUIVALENCE
     1  (XX(    4),    TAV), (XX(    8),     TB), (XX(   10),   WPCT),
     2  (XX(   14),   TREP), (XX( 214),       T), (XX( 414),      CC),
     3  (XX(1268),      TO), (XX(1215),    NXNT), (XX(    5),      N1),
     4  (XX(1601),    NXNW), (XX(    7),    NPTS), (XX(    9),      TT),
     5  (XX(   11),   TPCT), (XX(    3),    ICON)
      EQUIVALENCE
     1  (XX(1602), BREAKW), (XX(1612), BREAKT), (XX(1617),    WINC),
     2  (XX(1627),    TINC), (XX(1632),     NGO), (XX(1642),  NSTOP)
      NI = ICON - 1
      XI = -1.
      NSUM = NXNW + NXNT
      DO 45  IJ = 1, NSUM
      XI = 0.
      NG = NGO(IJ)
      NSP= NSTOP(IJ)
      DELW  = BREAKW(IJ+1) - BREAKW(IJ)
      IK    = IJ - NXNW
      DELT  = BREAKT(IK+1) - BREAKT(IK)
      DO 40 I = NG, NSP
      IF (IJ .GT. NXNW) GO TO 15
      IF (IJ .NE. 1) GO TO 8
      IF ( I .EQ. 1) GO TO 9
    8 XI = XI + 1.
    9 T(I)  = BREAKW(IJ) + XI*WINC(IJ)*DELW
      GO TO 21
   15 XI = XI + 1.
      T(I)  = BREAKT(IK) + XI*TINC(IK)*DELT
   21 DO 28  J = 1,NPTS
      IF (TREP(J) - T(I)) 28, 28, 22
   22 JS = J - 3
      IF (JS) 23, 23, 25
   23 JS = 1
      GO TO 32
   25 IF (JS - NPTS + NI) 32, 32, 26
   26 JS = NPTS - NI
      GO TO 32
   28 CONTINUE
   32 TQ = T(I) - TAV
      Y(I) = ((CC(JS,4)*TQ + CC(JS,3))*TQ + CC(JS,2))*TQ + CC(JS,1)
   40 CONTINUE
   45 CONTINUE
      RETURN
      END
```

```
'IP FOR LSTSQ
      SUBROUTINE LSTSQ (NGO, NSTOP, TIME, IPASS)
      DIMENSION  XX(1700)
      DIMENSION           XLNP (3,15,200), FSAVE(3,15,200), FD1(3,15,200),
     1 PSAVE(3,15,200),  FD2 (3,15,200),   FD3(3,15,200),   PD(3,15,200)
      DIMENSION             C2MAX(200,3),    C2MIN(200,3),    XMFD(200,3),
     1     PIKD(3,200),     XMLNP(3, 200),   XMOTNO( 3,15),   AETOT( 3,15),
     2       PAF(3, 15),      TEMP(3,  15),      TIME(3, 15),   ARRAY(46, 4),
     3       XLNPD(200),    C1MIN    (200),   C1MAX  (200),     WPRNT(200),
     4        XMTD(200),    C3MIN    (200),     C3MAX  (200),     XMPTD(200),
     5        SET1( 45),      SET2    ( 45),   XMTEMP   (  3),     DENOM(  3),
     6        EXPA(  3),      SIGP    (  3),          X  (  4),     KOUNT(  4),
     7          IL(  4),       LGO    ( 10),       PCTW  ( 10),       PCTT(  5),
     8        WINC( 10),      TINC    (  5),      SET3(45) , TEMX(45)
      DIMENSION BCDX(12), BCD1(12), BCD2(12), BCD3(12), BCD4(12)
      COMMON  XX
      COMMON XLNP, FSAVE, PSAVE
      COMMON  BCDX, BCD1, BCD2, BCD3, BCD4
      EQUIVALENCE
     1  (XX(    5),      N1), (XX(   11),   TPCT ), (XX(   10),     WPCT),
     2  (XX(1216),      NT), (XX(1217),   KOUNT), (XX(1221),      TEMP),
     3  (XX(1269),      NP), (XX(1270),     EXPA), (XX(1273),    AETOT),
     4  (XX(1318),     PAF), (XX(1363),     CPHI), (XX(1364),        TD),
     5  (XX(1365),   WPRNT), (XX(1565), XMOTNO), (XX(   13),       CT2),
     6  (XX(    1),    NSV2), (XX(1214),     NSV1), (XX(1601),     NXNW),
     7  (XX(1632),     LGO), (XX(1652),     PCTW), (XX(1662),     PCTT),
     8  (XX(1617),    WINC), (XX(1627),    TINC)
      EQUIVALENCE (XLNP(1), FD1(1), FD2(1), FD3(1), PD(1))
C
C          OUTPUT FORMATS
C
  200 FORMAT (1H1, 13X, 56H TRANSFORMED TIMES          TRANSFORMED CHAMB
     1ER PRESSURE )
  201 FORMAT (7H    PCT., 16X, 38H MEANS WITH TWO SIDED TOLERANCE LIMITS/
     1 36H    WEB       MEAN        MIN.         MAX. ,7X, 30H MEAN          MIN.
     2          MAX. / 7H    TIME   )
  202 FORMAT (1H1, 13X, 54H TRANSFORMED TIMES          TRANSFORMED THRUS
     1T AT PA =F8.2)
  205 FORMAT (1X, F6.2, 3F10.4, 3F12.4)
 1201 FORMAT ( 5H PCT. / 8H TAILOFF / 5H TIME )
C
      NSV0 = N1 + 1
      IF (IPASS - 2) 66, 72, 66
   66 NM = KOUNT(1) + KOUNT(2) + KOUNT(3)
C
C               COMPUTE STATISTICS FOR TEMPERATURE
C
      DO 70  J = 1,NT
      LL = 0
      JJ = KOUNT(J)
      DO 68  K = 1,JJ
      LL = LL + 1
   68 SET1(LL) = TEMP(J,K)
      CALL STATS (SET1, LL, XMTEMP(J), ARG)
      DENOM(J) = TD - XMTEMP(J)
   70 CONTINUE
C
C        LEAST SQUARE FIT LOG OF BURN TIME VS. TEMP.
C
   72 LL = 0
      DO 210  J = 1,NT
```

```
      JJ = KOUNT(J)
      DO 210  K = 1,JJ
      LL = LL + 1
      ARRAY(LL,1) = 1.
      ARRAY(LL,2) = TEMP(J,K)
      ARRAY(LL,3) = TEMP(J,K)**2
      ARG = TIME(J,K)
      ARRAY(LL,4) = ALOG(ARG)
  210 CONTINUE
      CALL GLS1 (ARRAY, X, IL, NM, 3, ALPHA, 0., 0.)
      XLNPD(1) = (X(3)*TD + X(2))*TD + X(1)
C
C               COMPUTE LOG OF MEANS OF BURN TIMES AND SIGP'S
C
      DO  230 J = 1,NT
      XMLNP(J,1) = (X(3)*XMTEMP(J) + X(2))*XMTEMP(J) + X(1)
  230 SIGP(J) = (XMLNP(J,1) - XLNPD(1)) / DENOM(J)
C
C          COMPUTE TRANSFORMED TIMES AND IN THE SAME LOOP LEAST SQUARE
C          FIT LOG OF PRESSURE VS. TEMP
C
      IJ = 0
      DO 240 L = NGO, NSTOP
      IQ = IJ + 1 + (IPASS-1)*NXNW
      IF (L .NE. LGO(IQ)) GO TO 231
      IJ = IJ + 1
      AL = 0.
      INCR = 1 / IJ
      XINC = INCR
  231 AL = AL + 1
      LL = 0
      DO  235 J = 1,NT
      JJ = KOUNT(J)
      DO  235 K = 1,JJ
      LL = LL + 1
      ARRAY(LL,4) = XLNP(J,K,L)
      ARG = SIGP(J)*(TD - TEMP(J,K))
      TEM = TIME(J,K) / (EXP (ARG))
      IF (IPASS - 1) 232, 232, 233
  232 TEMX(LL) = TEM
      DELW = PCTW(IJ+1) - PCTW(IJ)
      SET1(LL) = TEM * (PCTW(IJ) + (AL-XINC)*WINC(IJ)*DELW)
      GO TO 235
  233 DELT = PCTT(IJ+1) - PCTT(IJ)
      TAM = TEM - TEMX(LL)
      SET1(LL) = TEMX(LL)    + TAM * (PCTT(IJ) + AL*TINC(IJ)*DELT)
  235 CONTINUE
      CALL STATS (SET1, LL, XMTD(L), SDTD)
      C1MIN(L) = XMTD(L) - CT2*SDTD
      C1MAX(L) = XMTD(L) + CT2*SDTD
      CALL GLS1 (ARRAY, X, IL, NM, 3, ALPHA, 0., 0.)
      XLNPD(L) = (X(3)*TD + X(2))*TD + X(1)
C
C               COMPUTE XMLNP FROM CURVE FIT AND PIK'S
C
      DO 240  J = 1, NT
      XMLNP(J,L) = (X(3)*XMTEMP(J) + X(2))*XMTEMP(J) + X(1)
      PIKD(J,L) = (XLNPD(L) - XMLNP(J,L)) / DENOM(J)
  240 CONTINUE
C
C          COMPUTE AND PRINT OUT TRANSFORMED PRESSURES
C
```

```
      DO 100 L = NGO, NSTOP
      LL = 0
      DO 90 J = 1,NT
      JJ = KOUNT(J)
      DO 90 K = 1,JJ
      LL = LL + 1
      ARG = PIKD(J,L) * (TD - TEMP(J,K))
      PD(J,K,L) = EXP (XLNP(J,K,L) + ARG)
      SET3(LL) = PD(J,K,L)
   90 CONTINUE
      CALL STATS (SET3     , NM, XMPTD(L), SDPTD)
      C3MIN(L) = XMPTD(L) - CT2*SDPTD
      C3MAX(L) = XMPTD(L) + CT2*SDPTD
  100 CONTINUE
C
      IF (IPASS - 2) 182, 101, 182
  101 WRITE (6,200)
      WRITE (6, 201)
      DO 105  L = 1, NSV2
      IF (L - NSV1) 105, 104, 105
  104 WRITE (6, 1201)
  105 WRITE (6, 205) WPRNT(L), XMTD(L), C1MIN(L), C1MAX(L),
     1 XMPTD(L), C3MIN(L), C3MAX(L)
      CALL QUIKMV (-1, 46 , BCDX, BCD1, -NSV2,C1MAX, C3MAX)
      CALL QUIKMV ( 0, 67 , BCDX, BCD1, -NSV2, XMTD, XMPTD)
      CALL QUIKMV ( 0, 77 , BCDX, BCD1, -NSV2,C1MIN, C3MIN)
C
C                 COMPUTE TRANSFORMED THRUSTS
C
  182 DO 198  L = NGO, NSTOP
      LL = 0
      DO 190 J = 1,NT
      JJ = KOUNT(J)
      DO 190 K = 1,JJ
      LL = LL + 1
      RATIO     = PD(J,K,L) / PSAVE(J,K,L)
      FVAC  = FSAVE(J,K,L) + PAF(J,K)*AETOT(J,K)*CPHI
      FD1(J,K,L)= RATIO*FVAC - EXPA(1)*AETOT(J,K)*CPHI
      SET1(LL) = FD1(J,K,L)
      IF (NP - 1) 190, 190, 184
  184 FD2(J,K,L)= RATIO*FVAC - EXPA(2)*AETOT(J,K)*CPHI
      SET2(LL) = FD2(J,K,L)
      IF (NP - 2) 190, 190, 186
  186 FD3(J,K,L)= RATIO*FVAC - EXPA(3)*AETOT(J,K)*CPHI
      SET3(LL) = FD3(J,K,L)
  190 CONTINUE
      CALL STATS (SET1, NM, XMFD(L,1), SDFD)
      C2MIN(L,1) = XMFD(L,1) - CT2*SDFD
      C2MAX(L,1) = XMFD(L,1) + CT2*SDFD
      IF (NP - 1) 198, 198, 194
  194 CALL STATS (SET2, NM, XMFD(L,2), SDFD)
      C2MIN(L,2) = XMFD(L,2) - CT2*SDFD
      C2MAX(L,2) = XMFD(L,2) + CT2*SDFD
      IF (NP - 2) 198, 198, 196
  196 CALL STATS (SET3      , NM, XMFD(L,3), SDFD)
      C2MIN(L,3) = XMFD(L,3) - CT2*SDFD
      C2MAX(L,3) = XMFD(L,3) + CT2*SDFD
  198 CONTINUE
C
      IF (IPASS - 2) 209, 301, 209
  301 M = 0
  204 M = M + 1
```

```
      WRITE (6, 202) EXPA(M)
      WRITE (6, 201)
      DO 206  L = 1, NSV2
      IF (L - NSV1) 206, 207, 206
  207 WRITE (6, 1201)
  206 WRITE (6, 205) WPRNT(L), XMTD(L), C1MIN(L), C1MAX(L),
     1 XMFD(L,M), C2MIN(L,M), C2MAX(L,M)
      IF (M - NP) 204, 218, 218
  218 CALL QUIKMV (-1, 46 , BCDX, BCD2, -NSV2,C1MAX, C2MAX(1,1))
      CALL QUIKMV ( 0, 67 , BCDX, BCD2, -NSV2, XMTD,  XMFD(1,1))
      CALL QUIKMV ( 0, 77 , BCDX, BCD2, -NSV2,C1MIN, C2MIN(1,1))
      IF (NP - 1) 209, 209, 220
  220 CALL QUIKMV (-1, 46 , BCDX, BCD3, -NSV2,C1MAX, C2MAX(1,2))
      CALL QUIKMV ( 0, 67 , BCDX, BCD3, -NSV2, XMTD,  XMFD(1,2))
      CALL QUIKMV ( 0, 77 , BCDX, BCD3, -NSV2,C1MIN, C2MIN(1,2))
      IF (NP - 2) 209, 209, 222
  222 CALL QUIKMV (-1, 46 , BCDX, BCD4, -NSV2,C1MAX, C2MAX(1,3))
      CALL QUIKMV ( 0, 67 , BCDX, BCD4, -NSV2, XMTD,  XMFD(1,3))
      CALL QUIKMV ( 0, 77 , BCDX, BCD4, -NSV2,C1MIN, C2MIN(1,3))
  209 RETURN
      END
```

```
'IP FOR CURVF
      SUBROUTINE CURVF (T, S, N)
C     CURVE FIT
      DIMENSION  XX(1700)
      DIMENSION  ALD(400), DAL(400), ALC(400), ALW(1000),  TC(200),
     1               T(200),    S(200),  A3(280),  SC( 200), DP(200),
     2           DER(200),  CC(200,4)
      COMMON  XX
      EQUIVALENCE
     1  (XX(   3),  ICON), (XX(   4),    TAV), (XX( 414),    CC),
     2  (XX(1266), IPRNT)
      WRITE (6,111)
  110 FORMAT (2F15.5)
  111 FORMAT (1H1,12HINPUT ARRAYS)
      WRITE (6,110) (T(I),S(I), I=1,N)
C   ***THE ELEMENTS OF THE T AND S ARRAYS ARE ARRANGED IN ASCENDING ORDER
C       ,AND ARE THEN NORMALIZED.
      CALL ACCEND(T ,S ,N)
      CALL NORMLZ(T,TC,N,TAV)
      CALL NORMLZ(S,SC,N,SAV)
      DO 1 I=1,N
      ALD(2*I)=TC(I)
      ALD(2*I-1)=SC(I)
      DAL(2*I)=SC(I)
    1 DAL(2*I-1)=TC(I)
C     HERE BEGINS PIECEWISE FITS WITH CUBICS, EMBRACING M POINTS AT TIME
      K = 1
      M = ICON
      IF(M-2*(M/2))23,24,23
   23 IS=0
      GO TO 25
   24 IS=1
   25 MH=M/2
      MK = M
      MM=N+1-M
      IF (IPRNT) 62, 64, 62
   62 WRITE (6, 63) TAV, M
   63 FORMAT(78H1FOLLOWING ARE COEFFICIENTS OF SLIGHTLY DISCONTINUOUS CU
  631BICS IN THE QTY (TIME-F12.7,25H), FOR TIME RANGES SHOWN./86HOTIME
  632RANGE            COEFF. OF Q**0   COEFF. OF Q**1   COEFF. OF Q**2   C
  633OEFF. OF Q**3,I4,22H POINTS LINKED PER SET//)
C   ***COMPUTING THE COEFFICIENTS OF THE CUBICS AND STORING THEM IN THE
C       CC ARRAY
   64 DO 52 I=1,MM
      CALL CF2F1(0,ALC,0,A3,ALD(2*I-1),3,M)
      CALL CF2F2(0.0,ALC,3,DER,3,3)
      CC( I,1) = DER(1) + SAV
      B=1.
      DO 52 J=1,3
      B = B * FLOAT (J)
   52 CC( I, J+1) = DER(2*J+1) / B
      CALL CF2F1(0,ALC,0,A3,ALD,3,M)
C   ***COMPUTING THE DISCREPANCIES FOR THE PIECE-WISE FITS.
      DO 53 I=1,N
      J=I-(MH-IS)
      IF(J-1)57,57,56
   56 IF(J-MM)58,58,57
   58 CALL CF2F1(0,ALC,0,A3,ALD(2*J-1),3,M)
   57 CALL CF2F2(TC(I),ALC,3,DER,0,3)
   53 DP(I) = SC(I) - DER(1)
      E=T(1)
```

```
      IZ=MH-IS
      H=T(IZ+1)
      DO 166 I = 1,MM
      IF (IPRNT) 51, 55, 51
   51 WRITE (6, 54) B, H, (CC(I,J), J=1,4)
   54 FORMAT(F7.2,3H TOF8.2,4E17.8)
   55 B=H
      J=I+MH+1-IS
      IF(I+1-MM)164,165,164
  165 J=N
      H = T(J)
      GO TO 166
  164 H=T(J)
  166 CONTINUE
      WRITE (6,124) MK
  124 FORMAT(17H PIECEWISE CUBICS / 21H NO. OF POINTS LINKED ,I2)
      WRITE (6,127)
  127 FORMAT (1H0,3(30X,6HDELTAS))
      DO 151 J=1,N,1
  151 WRITE (6,153) T(J),S(J),DP(J)
  153 FORMAT (F10.4,2X,F10.3,F15.6)
      CALL STDEV (DP,N,S2)
      WRITE (6, 11) S2
   11 FORMAT (19H0         STD. DEV. F11.6)
      RETURN
      END


'IP FOR STATS
      SUBROUTINE STATS (X, N, XM, SD)
      DIMENSION X(100)
      SUM1 = 0.
      SUM2 = 0.
      XNO = N
      XNO1 = XNO - 1.
      DENOM = XNO*XNO1
      DO 10 I = 1,N
      SUM1 = SUM1 + X(I)
   10 SUM2 = SUM2 + X(I)*X(I)
      XM = SUM1 / XNO
      XNUM = (XNO*SUM2 - SUM1*SUM1)
      IF (XNUM .LE. 0.) GO TO 14
      SD = SQRT(XNUM/DENOM)
      IF ((SD/XM) .LT. .00025) GO TO 14
      GO TO 15
   14 SD = 0.
   15 RETURN
      END
```

```
'IP FOR GLS1-
      SUBROUTINE GLS1(B,X,IL,N,M,ALPHA,E1,E2)                           GLSQ0001
      DIMENSION A(46,4), B(46,4), X(4), IL(4)
      MM=M+1                                                            GLSQ0003
      DO 2  J = 1,N
      DO 2  K = 1,MM
    2 A(J,K) = B(J,K)
      LL=1                                                              GLSQ0004
      DO 60J=1,MM                                                       GLSQ0005
   60 IL(J)=0                                                           GLSQ0006
      I=1                                                               GLSQ0007
      DO 3K=1,MM                                                        GLSQ0008
      II=I+1                                                            GLSQ0009
      DO 4J=II,N                                                        GLSQ0010
      IF (ABS(A(J,K))-E1)4,4,6                                          GLSQ0011
    6 T1=SQRT((A(J,K))**2+(A(I,K))**2)                                  GLSQ0012
      S=A(J,K)/T1                                                       GLSQ0013
      C=A(I,K)/T1                                                       GLSQ0014
      DO 5L=K,MM                                                        GLSQ0015
      T2=C*A(I,L)+S*A(J,L)                                              GLSQ0016
      A(J,L)=-S*A(I,L)+C*A(J,L)                                         GLSQ0017
    5 A(I,L)=T2                                                         GLSQ0018
      LL=LL+1                                                           GLSQ0019
    4 CONTINUE                                                          GLSQ0020
      IF (ABS(A(I,K))-E2)3,3,8                                          GLSQ0021
    8 IL(K)=I                                                           GLSQ0022
      I=I+1                                                             GLSQ0023
    3 CONTINUE                                                          GLSQ0024
      X(MM)=-1.0                                                        GLSQ0025
      II=M                                                              GLSQ0026
      DO 35I=1,M                                                        GLSQ0027
   35 X(I)=0.                                                           GLSQ0028
      DO 30J=1,M                                                        GLSQ0029
      IF (IL(II))30,30,31                                               GLSQ0030
   31 S=0.                                                              GLSQ0031
      LL=II+1                                                           GLSQ0032
      I=IL(II)                                                          GLSQ0033
      DO 32K=LL,MM                                                      GLSQ0034
   32 S=S+A(I,K)*X(K)                                                   GLSQ0035
      X(II)=-S/A(I,II)                                                  GLSQ0036
   30 II=II-1                                                           GLSQ0037
      IF (IL(MM))50,51,50                                               GLSQ0038
   51 ALPHA=0.                                                          GLSQ0039
      GO TO 52                                                          GLSQ0040
   50 I=IL(MM)                                                          GLSQ0041
      ALPHA=A(I,MM)                                                     GLSQ0042
   52 RETURN                                                            GLSQ0043
      END                                                               GLSQ0044
```

```
'IP FOR ACCEND
      SUBROUTINE ACCEND(X,Y,N)                                            ASCN0010
C       THIS SR SORTS (X,Y) POINTS INTO A SEQUENCE OF ASCENDING X VALUES.ASCN0020
C       N IS THE NO. OF POINTS IN THE SEQUENCE  WHILE X AND Y ARE ASSOCI-ASCN0030
C       ATED ARRAYS.  THE ARRAYS OCCUPY THE SAME STORAGE AFTER SORTING ASASCN0040
C       THEY DID BEFORE SORTING.  N IS THE NO. OF POINTS.                 ASCN0050
      DIMENSION X(1),Y(1)                                                 ASCN0060
C       DIMENSIONS OF ABOVE VARIABLES ARE ACTUALLY EFFECTED BY THE        ASCN0070
C       HIGHER (CALLING) PROGRAM OR SR.                                   ASCN0080
      EQUIVALENCE(I,T)                                                    ASCN0090
      J=1                                                                 ASCN0100
C       J IS THE INDEX OF THE NEXT MEMBER OF THE SET OF POINTS WHICH WILLASCN0110
C       BE ORDERED BY OPERATIONS IN THE INNER LOOP  DO 8 ON I.            ASCN0120
      GO TO 3                                                             ASCN0130
C       THE ABOVE TRANSFER AVOIDS MIS-OPERATION IF N=1 OR LESS, SEE CARD ASCN0140
C       NO 305.  NORMALLY, PROGRESS TO STMT. 4.                           ASCN0150
    4 K=J                                                                 ASCN0160
C       K IS THE TENTATIVE INDEX OF THE SMALLEST UN-ORDERED X VALUE.      ASCN0170
      I=J+1                                                               ASCN0180
      GO TO 6                                                             ASCN0190
    5 I=I+1                                                               ASCN0200
    6 IF(X(I)-X(K))1,8,8                                                  ASCN0210
    1 K=I                                                                 ASCN0220
    8 IF(I-N)5,7,7                                                        ASCN0230
C       K IS NO LONGER TENTATIVE.  IT IS INDEED THE INDEX OF SMALLEST X, ASCN0240
C       SO FAR UNORDERED.                                                 ASCN0250
    7 IF(K-J)2,9,2                                                        ASCN0260
    2 T=X(K)                                                              ASCN0270
      X(K)=X(J)                                                           ASCN0280
      X(J)=T                                                              ASCN0290
      T=Y(K)                                                              ASCN0300
      Y(K)=Y(J)                                                           ASCN0310
      Y(J)=T                                                              ASCN0320
C       BOTH X AND Y HAVE BEEN SWAPPED, USING T AS A TEMPORARY STORAGE.   ASCN0330
    9 J=J+1                                                               ASCN0340
    3 IF(J-N)4,10,10                                                      ASCN0350
   10 RETURN                                                              ASCN0360
      END                                                                 ASCN0370
```

```
'IP FOR NORMLZ
      SUBROUTINE NORMLZ(X,Y,N,AV
C     THIS ROUTINE AVERAGES N ELEMENTS IN X ARRAY GETTING AV AS THE
C     RESULT. IT THEN PRODUCES ARRAY Y WHICH IS SAME AS X,MINUS AV,
C     TERM BY TERM
      DIMENSION X(1),Y(1)
      AV=X(1)
      DO 1 I=2,N
    1 AV=AV+X(I)
      AV = AV / FLOAT(N)
      DO 2 I=1,N
    2 Y(I)=X(I)-AV
      RETURN
      END
```

```
'IP FOR FIXIT
      SUBROUTINE FIXIT(A,B,N,N2,N7)
      DIMENSION A(10,1),B(5,1),D(50),KK(4),S1(15),S2(15)
      DO 200 J=N2,N7
      DO 100 I=1,N
  100 D(I)=A(J,I)
  200 CALL STDEV(D,N,S1(J))
      KK(1)=8
      KK(2)=10
      KK(3)=12
      KK(4)=14
      DO 400 J=1,4
      DO 300 I=1,N
  300 D(I)=B(J,I)
  400 CALL STDEV(D,N,S2(J))
      WRITE (6, 11) (S1(J), J=N2,N7), (S2(J), J=1,4)
   11 FORMAT(13H0  STD. DEV. ,6F11.6,2X,4F11.6)
      RETURN
      END
```

```
'IP FOR STDEV
      SUBROUTINE STDEV(X,N,S)
      DIMENSION X(1)
      SUMX=0.
      XSQR=0.
      EN=N
      DO 100 I=1,N
      SUMX=SUMX+X(I)
  100 XSQR=XSQR+(X(I))**2
      SP=XSQR-((SUMX**2)/EN)
      S = SQRT (SP / (EN - 1.))
      RETURN
      END
```

```
•IP FOR CF2F1
      SUBROUTINE CF2F1(J,ALC,IT,ALW,ALD,K,N)                            CF2
      DIMENSION TMP(2),ALW(1),ALD(1),ALC(1)                             CF2
      K6=6*K+4                                                          CF2
      FK=K                                                              CF2
      NT = 2                                                            CF2
      IF(IT)11,1,11                                                     CF2
   11 NT = 3                                                            CF2
    1 IF(J)33,3,33                                                      CF2
   33 TMP(2)=0.0                                                        CF2
      TMP(1)=0.0                                                        CF2
      GO TO 2                                                           CF2
    3 ALC(K6+4)=0.0                                                     CF2
      ALC(K6+3)=0.0                                                     CF2
      FN=N                                                              CF2
      NN=0                                                              CF2
      DO 5 I=1,N                                                        CF2
      NN = NN + NT                                                      CF2
      ALC(K6+4)=ALC(K6+4)+ALD(NN)                                       CF2
    5 ALC(K6+3)=ALC(K6+3)+ALD(NN-1)                                     CF2
      TMP(1)=ALC(K6+3)/FN                                               CF2
      ALC(K6+3)=TMP(1)                                                  CF2
      TMP(2)=ALC(K6+4)/FN                                               CF2
      ALC(K6+4)=TMP(2)                                                  CF2
    2 NN=0                                                              CF2
      NM=-2                                                             CF2
      DO 6 I=1,N                                                        CF2
      NM=NM+5                                                           CF2
      NN = NN + NT                                                      CF2
      ALW(NM)=ALD(NN)-TMP(2)                                            CF2
    6 ALW(NM-1)=ALD(NN-1)-TMP(1)                                        CF2
      NM=-2                                                             CF2
      NN=-4                                                             CF2
      DO 8 I=1,N                                                        CF2
      NN=NN+5                                                           CF2
      IF(IT)77,7,77                                                     CF2
   77 NM = NM + NT                                                      CF2
      ALW(NN)=ALD(NM)                                                   CF2
      GO TO 8                                                           CF2
    7 ALW(NN)=1.0                                                       CF2
    8 CONTINUE                                                          CF2
      NN=0                                                              CF2
      DO 9 I=1,N                                                        CF2
      NN=NN+5                                                           CF2
    9 ALW(NN)=1.0                                                       CF2
      FJJ=0.0                                                           CF2
      ALC(K6+5)=J                                                       CF2
      IF(J)17,15,17                                                     CF2
   17 FJJ=FJJ+1.0                                                       CF2
      NN=0                                                              CF2
      DO 18 I=1,N                                                       CF2
      NN=NN+5                                                           CF2
   18 ALW(NN)=ALW(NN)*ALW(NN-2)                                         CF2
      IF(FJJ-ALC(K6+5))17,15,15
   15 ALC(K6+1)=0.0                                                     CF2
      ALC(K6+2)=0.0                                                     CF2
      ALC(K6-2)=0.0                                                     CF2
      NN=-4                                                             CF2
      DO 19 I=1,N                                                       CF2
      NN=NN+5                                                           CF2
      TMP(1)=ALW(NN)*ALW(NN+4)                                          CF2
```

```
      ALC(K6+2)=TMP(1)*ALW(NN+1)+ALC(K6+2)                              CF2
      TMP(1)=TMP(1)*ALW(NN+4)                                           CF2
      ALC(K6+1)=TMP(1)+ALC(K6+1)                                        CF2
   19 ALC(K6-2)=TMP(1)*ALW(NN+2)+ALC(K6-2)                              CF2
      ALC(K6-2)=ALC(K6-2)/ALC(K6+1)                                     CF2
  111 ALC(K6+2)=ALC(K6+2)/ALC(K6+1)                                     CF2
  112 ALC(K6-3)=0.0                                                     CF2
  242 L3=0                                                              CF2
      L1=-1                                                             CF2
      L4=0                                                              CF2
  113 K6=K6-6                                                           CF2
  114 IF(FJJ-FK)115,122,122                                             CF2
  115 FJJ=FJJ+1.0                                                       CF2
      ALC(K6+2)=0.0                                                     CF2
      ALC(K6+1)=0.0                                                     CF2
      ALC(K6-2)=0.0                                                     CF2
      NN=-4                                                             CF2
      DO 119 I=1,N                                                      CF2
      NN=NN+5                                                           CF2
      L2=NN+L1                                                          CF2
      L5=NN+L4                                                          CF2
      TMP(1)=ALC(K6+3)*ALW(L2+4)                                        CF2
      ALW(L2+4)=(ALW(NN+2)-ALC(K6+4))*ALW(L5+4)-TMP(1)                  CF2
      TMP(1) = ALW(L2+4) * ALW(NN)                                      CF2
      ALC(K6+2) = TMP(1) * ALW(NN+1) + ALC(K6+2)                        CF2
      TMP(1)=TMP(1)*ALW(L2+4)                                           CF2
      ALC(K6+1)=TMP(1)+ALC(K6+1)                                        CF2
  119 ALC(K6-2) = TMP(1) * ALW(NN+3) + ALC(K6-2)                        CF2
      ALC(K6-2)=ALC(K6-2)/ALC(K6+1)                                     CF2
      ALC(K6+2)=ALC(K6+2)/ALC(K6+1)                                     CF2
      ALC(K6-3)=ALC(K6+1)/ALC(K6+7)                                     CF2
      IF(L3)242,120,242                                                 CF2
  120 L3=1                                                              CF2
      L1=0                                                              CF2
      L4=-1                                                             CF2
      GO TO 113                                                         CF2
  122 RETURN                                                            CF2
      END                                                               CF2
```

```
'IP FOR CF2F2
      SUBROUTINE CF2F2(ARG,ALC,KB,DER,IQ,K)                             CF2
      DIMENSION TMP(6),ALC(1),DER(1)                                    CF2
      XBAR=ARG                                                          CF2
      FKB=KB                                                            CF2
      K6=6*K+4                                                          CF2
      IF(ALC(K6+5))22,222,22                                            CF2
  222 XBAR=XBAR-ALC(K6+4)                                               CF2
   22 FJJ=ALC(K6+5)                                                     CF2
      NN=K6+6                                                           CF2
      NQ=IQ+1                                                           CF2
      DO 23 I=1,NQ                                                      CF2
      NN=NN-6                                                           CF2
   23 ALC(NN)=0.0                                                       CF2
      TMP(1)=0.0                                                        CF2
   24 TMP(2)=1.0                                                        CF2
   25 IF(ALC(K6+5)-TMP(1))2311,231,2311                                 CF2
 2311 TMP(1)=TMP(1)+1.0                                                 CF2
      TMP(2)=TMP(2)*XBAR                                                CF2
      IF(TMP(2))25,24,25                                                CF2
  231 ALC(K6)=TMP(2)                                                    CF2
      IF(TMP(1))2322,232,2322                                           CF2
 2322 KK6=K6                                                            CF2
  236 KK6=KK6-6                                                         CF2
      IF(XBAR)234,2344,234                                              CF2
 2344 TMP(2)=ALC(KK6+6)                                                 CF2
      ALC(KK6+6)=0.0                                                    CF2
      GO TO 235                                                         CF2
  234 TMP(2)=ALC(KK6+6)/XBAR                                            CF2
  235 ALC(KK6)=TMP(2)*TMP(1)                                            CF2
      TMP(1)=TMP(1)-1.0                                                 CF2
      IF(TMP(1))236,232,236                                             CF2
  232 NN=-1                                                             CF2
      TMP(1)=ALC(K6+2)                                                  CF2
      TMP(2)=ALC(K6+1)                                                  CF2
      KK6S=K6                                                           CF2
      KK6=K6+6                                                          CF2
      DO 241 I=1,NQ                                                     CF2
      KK6=KK6-6                                                         CF2
      NN=NN+2                                                           CF2
      DER(NN)=TMP(1)*ALC(KK6)                                           CF2
  241 DER(NN+1)=(ALC(KK6)**2)/TMP(2)                                    CF2
  242 L3=0                                                              CF2
      L7=0                                                              CF2
      L1=-1                                                             CF2
      L4=6                                                              CF2
   26 IF(FJJ-FKB)2111,211,211                                           CF2
 2111 FJJ=FJJ+1.0                                                       CF2
      KK6=KK6S                                                          CF2
      NN=-1                                                             CF2
      TMP(1)=ALC(KK6-2)                                                 CF2
      TMP(2)=ALC(KK6-3)                                                 CF2
      TMP(3)=ALC(KK6-4)                                                 CF2
      TMP(4)=ALC(KK6-5)                                                 CF2
      KK6S=KK6-6                                                        CF2
      TMP(6)=0.0                                                        CF2
      KK6=K6+6                                                          CF2
      DO 29 I=1,NQ                                                      CF2
      KK6=KK6-6                                                         CF2
      NN=NN+2                                                           CF2
      L6=KK6+L7                                                         CF2
```

```
      L2=KK6+L1                                                          CF2
      TMP(5)=TMP(2)*ALC(L2)                                              CF2
      TMP(5)=(XBAR-TMP(1))*ALC(L6)-TMP(5)                                CF2
      L5=KK6+L4                                                          CF2
      ALC(L2)=TMP(6)*ALC(L5)+TMP(5)                                      CF2
      DER(NN)=ALC(L2)*TMP(3)+DER(NN)                                     CF2
      DER(NN+1)=((ALC(L2)**2)/TMP(4))+DER(NN+1)                          CF2
   29 TMP(6)=TMP(6)+1.0                                                  CF2
      IF(L3)242,30,242                                                   CF2
   30 L3=1                                                               CF2
      L1=0                                                               CF2
      L4=5                                                               CF2
      L7=-1                                                              CF2
      GO TO 26                                                           CF2
  211 IF(ALC(K6+5))212,2122,212                                          CF2
 2122 DER(1)=DER(1)+ALC(K6+3)                                            CF2
  212 RETURN                                                             CF2
      END
* XQT MAIN
```

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Technical information generated in connection with a NASA contract or grant and released under NASA auspices.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

TECHNICAL REPRINTS: Information derived from NASA activities and initially published in the form of journal articles.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities but not necessarily reporting the results of individual NASA-programmed scientific efforts. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546