CR-54748

MONTE CARLO
FLIGHT PERFORMANCE RESERVE
PROGRAM

GD|C-BTD65-176

January 1966

GENERAL DYNAMICS
Convair Division

MONTE CARLO
FLIGHT PERFORMANCE RESERVE
PROGRAM

GD|C-BTD65-176

January 1966

GENERAL DYNAMICS

Convair Division

FOREWORD

This report describes a computer program which was developed at the Convair Division of General Dynamics as a means of rapidly evaluating flight performance reserve (FPR) requirements for the Centaur vehicle. Major elements of the program are a linearized vehicle performance model and a Monte Carlo sampling method. Although the prime impetus for this development effort was FPR analysis, the program is generally applicable to performance evaluation of any system that may be represented by an explicit mathematical model.

Program development was conducted under the provisions of Contract NAS3-3232.

Prepared by R. A. Hayward

Prepared by for J. I. Ingber

Approved by J. A. Doggett

Approved by F. B. Anthony

E184

ACKNOWLEDGMENT

The considerable assistance of R. W. Estus in coordinating and editing the engineering and programming inputs to this report is gratefully acknowledged.

TABLE OF CONTENTS

Section		Page
1	INTRODUCTION	1
2	BASIC EQUATIONS	3
3	COMPUTER PROGRAM	7
	3.1 Input Variables	25
	3.2 Output	28
	3.3 Variables Used Within the Program	28
	3.4 Sample Case	32
	3.5 Input Card and Deck Format	41
Appendix	PROGRAM LISTING	45

LIST OF ILLUSTRATIONS

Figure	Title	Page
1	Generalized Computer Flow Chart	19
2	Detailed Computer Flow Chart	20
3	Subroutine FQPLOT	24
4	Subroutine NORAD	24
5	Sample Input	25
6	Machine Output	37
7	FPR Frequency Function	38
8	FPR Probability Function	39
9	FPR Probability Function Segment	40
10	Input Deck Setup	43

LIST OF TABLES

Table	Title	Page
1	Parameter-Contribution Equations and Method-of-Computation Numbers	8
2	Parameter-Controlled Input	32

SECTION 1
INTRODUCTION

A Monte Carlo method is essentially a sampling method for studying an artificial stochastic model of a physical or mathematical process. Systems of equations whose solutions are not readily obtainable by standard numerical techniques often may be handled by a stochastic process involving parameters that satisfy the equations. Often a judicious application of the physical model is made enabling one to circumvent the functional equations entirely.

The evaluation of Flight Performance Reserve (FPR), which is the fuel required to be held in reserve to provide for mission success under non-nominal flight operation, may be handled effectively by the above technique. The functional equations that describe the fuel reserve requirements are simply the multi-degree-of-freedom equations of motion for a powered vehicle. For more than a few parameters, the cost of such a direct approach quickly becomes prohibitive. Precision numerical solutions to such equations are generally limited to near-nominal conditions for all variables.

The physical model is, briefly, vehicle performance measured by burn-out weight at specified injection conditions, which is somewhat loosely related to a number of vehicle parameters.

Performance expressed as burn-out weight (W_{BO}) is

$$W_{BO} = P(\alpha_1, \alpha_2, \cdots, \alpha_n) \tag{1}$$

where the form of the function P is arbitrary and the α's are vehicle-related parameters that influence performance capability.

Historically, FPR has been computed by an RSS technique which assumes independence of variables. That is, given a set of parameters $p_1, p_2, \cdots, p_n$ and their associated independent changes in the vehicle's performance $\delta p_1, \delta p_2, \cdots, \delta p_n$, then

$$(3\sigma)\ \text{FPR} = \sqrt{\delta p_1^{\ 2} + \delta p_2^{\ 2} + \cdots + \delta p_n^{\ 2}} \qquad (2)$$

which ignores any covariant contribution to the calculation. Of the parameters traditionally used to determine FPR (those which contribute significantly to performance changes), many are clearly dependent. This apparent contradiction, coupled with the desire for a flexible tool to quickly evaluate the contributions of parameter variations to FPR, provided the stimulus for the creation of the Monte Carlo FPR program.

SECTION 2

BASIC EQUATIONS

The basic equations for the FPR analysis are derived from a performance function P which is configuration- and mission-independent.

Let

$$P = P(\alpha_1, \alpha_2, \cdots, \alpha_n) \tag{3}$$

represent some vehicle's performance as a function of the n variables α_i. These are arbitrary, but in total should be comprehensive in depicting any significant performance changes. Equation 3, then, is an explicit representation of performance measured as injection weight into a specified orbit.

Therefore, the change in vehicle performance, dP, is

$$dP = \frac{\partial P}{\partial \alpha_1} d\alpha_1 + \frac{\partial P}{\partial \alpha_2} d\alpha_2 + \cdots + \frac{\partial P}{\partial \alpha_n} d\alpha_n = \sum_{i=1}^{n} \frac{\partial P}{\partial \alpha_i} d\alpha_i \tag{4}$$

which holds whether or not the α_i's are independent.

Generally, Equation 4 is not evaluated directly since there may exist r relations of the form

$$\Phi(\alpha_1, \alpha_2, \cdots, \alpha_k) = 0 \tag{5}$$

correlating the variables considered.

Theoretically, it is possible to solve for the r α's in terms of the other (n-r) α's so that

$$dP = \sum_{i=1}^{n-r} \frac{\partial P^1}{\partial \alpha_i} d\alpha_i \tag{6}$$

where the function P^1 contains only independent variables. The difficulty associated with a concise formulation of the functions (Equation 5) is evident, necessitating a simplified approach.

The technique used to evaluate changes in vehicle performance, dP, corresponding to variations in the parameter values, $d\alpha$, is to use Equation 4 with the selection procedure for the $d\alpha_i$ modified to account for interdependence of the α's. (Otherwise, Equation 6 could be used directly with the selection of $d\alpha_i$ completely random.) Also, the function P is approximated by a related function f.

The analysis, then, involves the computation of the quantities

$$\sum^{\ell} \Delta P = \sum^{\ell} \sum_{i=1}^{n} \frac{\partial f}{\partial \alpha_i} d\alpha_i \tag{7}$$

and

$$\sum^{\ell} \Delta P^2 = \sum^{\ell} \left\{ \sum_{i=1}^{n} \frac{\partial f}{\partial \alpha_i} d\alpha_i \right\}^2 \tag{8}$$

where ℓ is the number of iterations required for a given confidence in the statistics.

It can be shown that the parent distribution associated with the above method will be approximately normal regardless of the individual variable distributions. Therefore, the mean, m, and standard deviation, s, of the vehicle's performance subjected to the ranges of the α variations are given directly by

$$m = \ell^{-1} \sum \Delta P \tag{9}$$

and

$$s = \sqrt{\ell^{-1} \sum \Delta P^2 - m^2} \tag{10}$$

The associated standard errors are

$$\sigma^2 (m) = s^2 \ell^{-1} \tag{11}$$

and

$$\sigma^2(s) = s^2 (2\ell)^{-1} \tag{12}$$

where the parent variance is estimated from the sample variance.

When applicable, parameter variations are selected by a random process from pre-established distribution functions. Generally, these distributions will assume a Gaussian form (often for lack of a more descriptive function). Any distribution may, nevertheless, be specified for any of the n parameters considered.

Let $R(\lambda, \nu)$ be a random variate from some distribution with parameters λ, ν. For a Gaussian distribution, λ and ν are the mean and standard deviation respectively.

Also, let σ_i be the standard deviation of the i^{th} parameter's variation. Similarly, σ_{i_j} is the standard deviation of the j^{th} variable associated with the i^{th} parameter.

Table 1 in Section 3 presents the methods presently used in determining the $d\alpha$'s.

SECTION 3
THE COMPUTER PROGRAM

This program is used to compute the overall performance of a system where each subsystem's contribution can be selected on the basis of a random number technique.

The random numbers used in this subsystem contribution simulation are normally distributed with the assurance that $|R| \leq 3.0$.

The Flight Performance Reserve study for which this program was developed has 39 subsystems or parameters, each assigned a mean and standard deviation if applicable. There are 14 separate methods used to compute the contribution of the parameters to the final results. These contributions are referred to as $d\alpha_i$'s. The method of calculating each subsystem's $d\alpha_i$ is assigned in the subroutine BLOCK DATA. Table 1 is a list of the $d\alpha_i$'s with their specific equations and method-of-computation numbers. The program presently has the capacity to study 200 parameters, each with 10 subgroups.

After the $d\alpha_i$'s are computed for each parameter, each $d\alpha_i$ is multiplied by its corresponding partial derivative. The scheme for selecting the partials is as follows: If $d\alpha_i$ is negative, the partial selected is the value of the first dependent variable; if $d\alpha_i$ is positive, the partial selected is the second dependent variable. With a minor program change, a capability can exist to linearly interpolate and extrapolate for each partial, using $d\alpha_i$ as the argument.

Figure 1 is a generalized flow chart of the computer program, and Figure 2 is a more detailed one. The definition of words used in Figure 2 can be found in Section 3.1, Input variables, and Section 3.3, Variables Used Within the Program.

There is an option to generate plots of the distributions compiled by each case considered: the frequency function, the cumulative function, and an enlargement of the cumulative function between the 95th and 100th percentiles. Figure 3 is the flow chart for the plotting package.

Figure 4 is the flow chart of the random-number generator.

The program, then, consists of four parts:

START (FORTRAN IV)

BLOCK (FORTRAN IV)

FQPLOT (FORTRAN IV)

RANDOM (MAP)

Although this program was developed with a particular study in mind, if one or more of the 14 methods of treating parameters would suit a potential users simulation, all that need be done to adapt the program for another simulation is recompile BLOCK DATA by assigning each parameter to the desired method of calculation. Further, the random numbers can easily be made to assume any distribution desired.

Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers

$d\alpha$	PARAMETER	EQUATION	METHOD NO.
		COMPONENT WEIGHT DATA (DRY)	
$d\alpha_1$	Booster	$d\alpha_1 = R(0, \sigma_1)$	1
$d\alpha_2$	Sustainer and Inter-stage Adapter	$d\alpha_{2_1} = R(0, \sigma_{2_1})$	2
		$d\alpha_{2_2} = R(0, \sigma_{2_2})$	
		$d\alpha_2 = R(0,\sigma_{2_1}) + R(0,\sigma_{2_2})$	
$d\alpha_3$	Centaur	$d\alpha_3 = R(0, \sigma_3)$	1
$d\alpha_4$	Nose Fairing	$d\alpha_4 = R(0, \sigma_4)$	1
$d\alpha_5$	Insulation Panels	$d\alpha_5 = R(0, \sigma_5)$	1
		BOOSTER FLIGHT EXPENDABLES	
$d\alpha_6$	Fuel Weight		
	Fuel Density	$\text{SUBSUM} = \sum_{i=1}^{2} R(0, \sigma_{6_{1_i}})$	1
		$d\alpha_{6_1} = \text{SUBSUM} \cdot \text{NFTV}$	

Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers (Contd)

dα	PARAMETER	EQUATION	METHOD NO.
		$AFD = NBFD + SUBSUM$	
	Probe Location	$d\alpha_{6_2} = R\ (0, \sigma_{6_2})$	
	Surface Level Variation	$d\alpha_{6_3} = R\ (0, \sigma_{6_3})$	
	Tank Pressure	$d\alpha_{6_4} = R\ (0, \alpha_{6_4})$	
	Tank Volume	$d\alpha_{6_5} = R\ (0, \sigma_{6_5})$	
	Tanking Level	$d\alpha_{6_6} = \left[\sum_{i=2}^{5} d\alpha_{6_i} + R\ (0, \sigma_{6_6}) \right] * AFD + d\alpha_{6_1}$	
	Ground Expended	$d\alpha_{6_7} = \sum_{i=1}^{3} R\ (0, \sigma_{6_{7_i}})$	
	Sustainer Thrust Decay	$d\alpha_{6_8} = R\ (0, \sigma_{6_8})$	
		Then, $d\alpha_6 = d\alpha_{6_6} + d\alpha_{6_7} + d\alpha_{6_8}$	
$d\alpha_7$	Fuel Density	$d\alpha_7 = SUBSUM$	3
$d\alpha_8$	Oxidizer Weight		
	Oxidizer Density	$SUM81 = \sum_{i=1}^{2} R\ (0, \sigma_{8_{1_i}})$	2
		$d\alpha_{8_1} = SUM81 \bullet NOTV$	
		$AOD = NLO2D + SUM81$	
	Sensor Location	$d\alpha_{8_2} = R\ (0, \sigma_{8_2})$	

Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers (Contd)

$d\alpha$	PARAMETERS	EQUATIONS	METHOD NO.
	Surface Level Variation	$d\alpha_{8_3} = R\ (0,\ \sigma_{8_3})$	
	Tank Pressure	$d\alpha_{8_4} = R\ (0,\ \sigma_{8_4})$	
	Tank Volume	$d\alpha_{8_5} = R\ (0,\ \sigma_{8_5})$	
	Tanking Level	$d\alpha_{8_6} = \left[\sum_{i=2}^{5} d\alpha_i + R\ (0,\ \sigma_{8_6})\right] * AOD + d\alpha_{8_1}$	
	Ground Expended	$d\alpha_{8_7} = \sum_{i=1}^{4} R\ (0,\ \sigma_{8_{7_i}})$	
	Thrust Decay	$d\alpha_{8_8} = R\ (0,\ \sigma_{8_8})$	
		Then, $d\alpha_8 = d\alpha_{8_6} + d\alpha_{8_7} + d\alpha_{8_8}$	
$d\alpha_9$	Oxidizer Density	$d\alpha_9 = SUM81$	3
		CENTAUR FLIGHT EXPENDABLES	
$d\alpha_{10}$	Fuel Weight		2
	Sensor Sensitivity	$d\alpha_{10_1} = R\ (0,\ \sigma_{10_1})$	
	Sensor Location	$d\alpha_{10_2} = R\ (0,\ \sigma_{10_2})$	
	Surface Variations	$d\alpha_{10_3} = R\ (0,\ \sigma_{10_3})$	
	Density	$DEND = R\ (0,\ \sigma_{10_4}) * NLH2D$ $d\alpha_{10_4} = DEND * NLH2V$	

Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers (Contd)

$d\alpha$	PARAMETERS	EQUATIONS	METHOD NO.
	Tank Volume	$d\alpha_{10_5} = R\ (0, \sigma_{10_5})\ (NLH2D + DEND)$	
	Tank Ullage	$d\alpha_{10_6} = -\left[R\ (MFUV^{\dagger}, \sigma_{10_6}) - 1.865\right] * (NLH2D + DEND)$	
		Then, $d\alpha_{10} = \sum_{i=1}^{6} d\alpha_{10_i}$	
$d\alpha_{11}$	Oxidizer Weight		2
	Sensor Sensitivity	$d\alpha_{11_1} = R\ (0, \sigma_{11_1})$	
	Sensor Location	$d\alpha_{11_2} = R\ (0, \sigma_{11_2})$	
	Surface Variations	$d\alpha_{11_3} = R\ (0, \sigma_{11_3})$	
	Density	$DENDO = R\ (0, \sigma_{11_4}) * NLO2D$	
		$d\alpha_{11_4} = DENDO * NLO2V$	
	Tank Volume	$d\alpha_{11_5} = R\ (0, \sigma_{11_5}) * (NLO2D + DENDO)$	
	Tank Ullage	$d\alpha_{11_6} = -\left[R\ (0, \sigma_{11_6}) - 1.77\right] * (NLO2D + DENDO)$	
		Then, $d\alpha_{11} = \sum_{i=1}^{6} d\alpha_{11_i}$	
$d\alpha_{12}$	<u>Booster Jettisoned Residuals</u>		2
	Trapped Fuel	$d\alpha_{12_1} = R\ (0, \sigma_{12_1})$	

† Mean Fuel Ullage Volume

Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers (Contd)

$d\alpha$	PARAMETERS	EQUATIONS	METHOD NO.
	Trapped Oxidizer	$d\alpha_{12_2} = R\ (0, \sigma_{12_2})$	
	Lube Oil	$d\alpha_{12_3} = R\ (0, \sigma_{12_3})$	
	Helium	$d\alpha_{12_4} = R\ (0, \sigma_{12_4})$	
		Then, $d\alpha_{12} = \sum_{i=1}^{4} d\alpha_{12_i}$	
$d\alpha_{13}$	Sustainer Jettisoned Residuals		2
	Trapped Fuel	$d\alpha_{13_1} = R\ (0, \sigma_{13_1})$	
	Trapped Oxidizer	$d\alpha_{13_2} = R\ (0, \sigma_{13_2})$	
	Lube Oil	$d\alpha_{13_3} = R\ (0, \sigma_{13_3})$	
	Helium	$d\alpha_{13_4} = R\ (0, \sigma_{13_4})$	
	Nitrogen	$d\alpha_{13_5} = R\ (0, \sigma_{13_5})$	
	GO_2 in Tank (Flight)	$d\alpha_{13_6} = R\ (0, \sigma_{13_6})$	
	GO_2 in Tank (Ground)	$d\alpha_{13_7} = R\ (MGO2TG^{\dagger})$	
	PU Bias	$d\alpha_{13_8} = R\ (SPUB^{\ddagger}, \sigma_{13_8})$	
		Then, $d\alpha_{13} = \sum_{i=1}^{8} d\alpha_{13_i}$	

† Mean GO_2 in Tank - Ground

‡ Sustainer PU Bias

Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers (Contd)

$d\alpha$	PARAMETERS	EQUATIONS	METHOD NO.
$d\alpha_{14}$	Centaur Jettisoned Residuals		6
	Trapped LO_2	$d\alpha_{14_1} = R\ (0, \sigma_{14_1})$	
	Trapped LH_2	$d\alpha_{14_2} = R\ (0, \sigma_{14_2})$	
	GO_2 in Tank	$d\alpha_{14_3} = R\ (0, \sigma_{14_3})$	
	GH_2 in Tank	$d\alpha_{14_4} = R\ (0, \sigma_{14_4})$	
	H_2O_2 Weight	$d\alpha_{14_5} = R\ (0, \sigma_{14_5})$	
	Helium	$d\alpha_{14_6} = R\ (0, \sigma_{14_6})$	
	Ice and Frost	$d\alpha_{14_7} = R\ (0, \sigma_{14_7})$	
	PU	Without end effect (MPURES = 1) $d\alpha_{14_8} = R\ (0, \sigma_{14_8})$ $d\alpha_{14_8} < -MU\ 14(8)$? Yes | RES = 1 $d\alpha_{14_8} = -PUSET * \left(d\alpha_{14_8} + MU\ 14(8)\right) - MU\ 14(8)$ No | RES = 0 With end effect (MPURES = 2) $d\alpha_{14_8} = R\ (0, \sigma_{14_8})$ R2 = R (0, R2) * HBPDIS $HBP = HBP + R2 - TLH2 + d\alpha_{14_2}$ R3 = R (0, R3) * LBPDIS	

Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers (Contd)

dα	PARAMETERS	EQUATIONS	METHOD NO.
		$LBP = LBP + R3 - TLO2 + d\alpha_{14_1}$	
		$CPUB = MU14(8)$	
		$d\alpha_{14_8} < -(CPUB - SUBIAS)$?	
		Yes | $R4 = R\ (0,\ R4) \bullet PUSET \bullet \sigma_{14_8}$	
		| $\overline{\overline{MR}} = \dfrac{(\dot{\omega}_1 + \dot{\omega}_2) * VLVLAG * (PUSET\text{-}MINSET)}{(MR + 1.0) * 2.0 * HBP} + MINSET$	
		| $d\alpha_{14_8} = R4 - (CPUB\text{-}SUBIAS) * PUSET$	
		| $d\alpha_{14_8} > 0.0$?	
		| Yes | $d\alpha_{14_8} = d\alpha_{14_8} + LBP - \overline{\overline{MR}} * HBP$	
		| | $d\alpha_{14_8} \geq 0.0$?	
		| | No | $d\alpha_{14_8} = -d\alpha_{14_8} / \overline{\overline{MR}}$	
		| No | $d\alpha_{14_8} = LPB - \overline{\overline{MR}} * HBP$	
		| | $d\alpha_{14_8} > 0.0$?	
		| | No | $d\alpha_{14_8} = -d\alpha_{14_8} / \overline{\overline{MR}}$	
		No | $\overline{MR} = \dfrac{MR\ (\dot{\omega}_1 + \dot{\omega}_2)\ VLVLAG * (PUSET - MAXSET)}{2.0 \bullet LBP \bullet (MR + 1.0)} + MAXSET$	
		| $d\alpha_{14_8} = d\alpha_{14_8} + CPUB - SUBIAS$	
		| $d\alpha_{14_8} > 0.0$?	

Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers (Contd)

$d\alpha$	PARAMETER	EQUATIONS	METHOD NO.
		No | Yes | $d\alpha_{14_8} = d\alpha_{14_8} - CPUB + HBP - (LBP/\overline{MR})$ $d\alpha_{14_8} \geq 0.0$? No | $d\alpha_{14_8} = -d\alpha_{14_8} * \overline{MR}$ No | $d\alpha_{14_8} = HBP - (LBP/\overline{MR})$ $d\alpha_{14_8} \geq 0.0$? No | $d\alpha_{14_8} = -d\alpha_{14_8} * \overline{MR}$	
		$d\alpha_{14} = \sum_{i=1}^{8} d\alpha_{14_i}$	
		<u>CENTAUR VENTING</u>	
$d\alpha_{15}$	Ground and Inflight LH_2	$d\alpha_{15} = R(0, \sigma_{15})$	1
$d\alpha_{16}$	Ground and Inflight LO_2	$d\alpha_{16} = R(0, \sigma_{16})$	1
$d\alpha_{17}$	Coast	$d\alpha_{17} = R(0, \sigma_{17})$	1
		<u>BOOSTER PROPULSION</u>	
$d\alpha_{18}$	Booster Mixture Ratio	$d\alpha_{18} = R(0, \sigma_{18})$	1
$d\alpha_{19}$	Booster Thrust	$d\alpha_{19} = TB + R(0, \sigma_{19})$	7
$d\alpha_{20}$	Booster ISP	$d\alpha_{20} = IB + R(0, \sigma_{20})$	8
$d\alpha_{21}$	Sustainer Mixture Ratio	$d\alpha_{21} = R(0, \sigma_{21})$	1

Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers (Contd)

$d\alpha$	PARAMETER	EQUATIONS	METHOD NO.
$d\alpha_{22}$	Sustainer Thrust	$d\alpha_{22} = TS + R\ (0, \sigma_{22})$	9
$d\alpha_{23}$	Sustainer ISP	$d\alpha_{23} = IS + R\ (0, \sigma_{23})$	10
$d\alpha_{24}$	Vernier Mixture Ratio	$d\alpha_{24} = R\ (0, \sigma_{24})$	1
$d\alpha_{25}$	Vernier Thrust	$d\alpha_{25} = TV + R\ (0, \sigma_{25})$	11
$d\alpha_{26}$	Vernier ISP	$d\alpha_{26} = IV + R\ (0, \sigma_{26})$	12
		CENTAUR PROPULSION	
$d\alpha_{27}$	Thrust	LOXRES = MU14 (1) + $d\alpha_{14_1}$ + MU14 (3) + $d\alpha_{14_3}$ LOXVNT = MU16 (1) + $d\alpha_{16}$ RES = 0 ? No | LOXRES = LOXRES + $d\alpha_{14_8}$ LOXAVL = MU11 (1) + $d\alpha_{11}$ - LOXRES - LOXVNT LH2RES = MU14 (2) + $d\alpha_{14_2}$ + MU14 (4) + $d\alpha_{14_4}$ LH2VNT = MU15 (1) + $d\alpha_{15}$ RES = 1 ? No | LH2RES = LH2RES + $d\alpha_{14_8}$ LH2AVL = MU10 (1) + $d\alpha_{10}$ - LH2RES - LH2VNT MR = LOXAVL/LH2AVL	13

Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers (Contd)

$d\alpha$	PARAMETER	EQUATIONS	METHOD NO.
		DTE1 = R (0, σ_{27})	
		DTE2 = R (0, σ_{27})	
		DIE1 = R (0, σ_{27})	
		DIE2 = R (0, σ_{27})	
		$\dot{\omega}1$ = (THNOM+DTE1)/(ISPNOM+DIE1)	
		$\dot{\omega}2$ = (THNOM+DTE2)/(ISPNOM+DIE2)	
		$d\alpha_{27}$ = DTE1 + DTE2 + (2.0 * [THNOM - MU27(1)])	
$d\alpha_{28}$	Specific Impulse (ISP)	$d\alpha_{28}$ = [$d\alpha_{27}$ + 2 MU27(1)]/($\dot{\omega}1$ + $\dot{\omega}2$) - MU28(1)	14
$d\alpha_{29}$	Thrust (Burn 2)	Set = 0.0	4
$d\alpha_{30}$	ISP (Burn 2)		
$d\alpha_{31}$	Thrust (Burn 3)		
$d\alpha_{32}$	ISP (Burn 3)		
$d\alpha_{33}$	Attitude Control (Coast 1)		
$d\alpha_{34}$	Attitude Control (Coast 2)		
$d\alpha_{35}$	Atmosphere	$d\alpha_{35}$ = R (0, 1)	5
$d\alpha_{36}$	Launch Azimuth		2
	Null Voltage	$d\alpha_{36_1}$ = R (0, σ_{36_1})	
	Roll Gyro Torquing Rate	$d\alpha_{36_2}$ = R (0, σ_{36_2})	
	Time Uncertainties	$d\alpha_{36_3}$ = R (0, σ_{36_3})	

Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers (Contd)

$d\alpha$	PARAMETER	EQUATIONS	METHOD NO.
	Allowed Tolerance	$d\alpha_{36_4} = R\,(0, \sigma_{36_4})$	
		Then, $d\alpha_{36} = \sum_{i=1}^{4} d\alpha_{36_i}$	
$d\alpha_{37}$	Pitch Program		2
	Voltage-Time Integral	$d\alpha_{37_1} = R\,(0, \sigma_{37_1})$	
	Gryo Torquing Rate-Voltage-Time Average	$d\alpha_{37_2} = R\,(0, \sigma_{37_2})$	
	Inverter Voltage	$d\alpha_{37_3} = R\,(0, \sigma_{37_3})$	
	Inverter Frequency	$d\alpha_{37_4} = R\,(0, \sigma_{37_4})$	
		Then, $d\alpha_{37} = \sum_{i=1}^{4} d\alpha_{37_i}$	
$d\alpha_{38}$	Drag Force	$d\alpha_{38} = R\,(0, \sigma_{38})$	1
$d\alpha_{39}$	Wind Profile	$d\alpha_{39} = R\,(0, 1)$	5

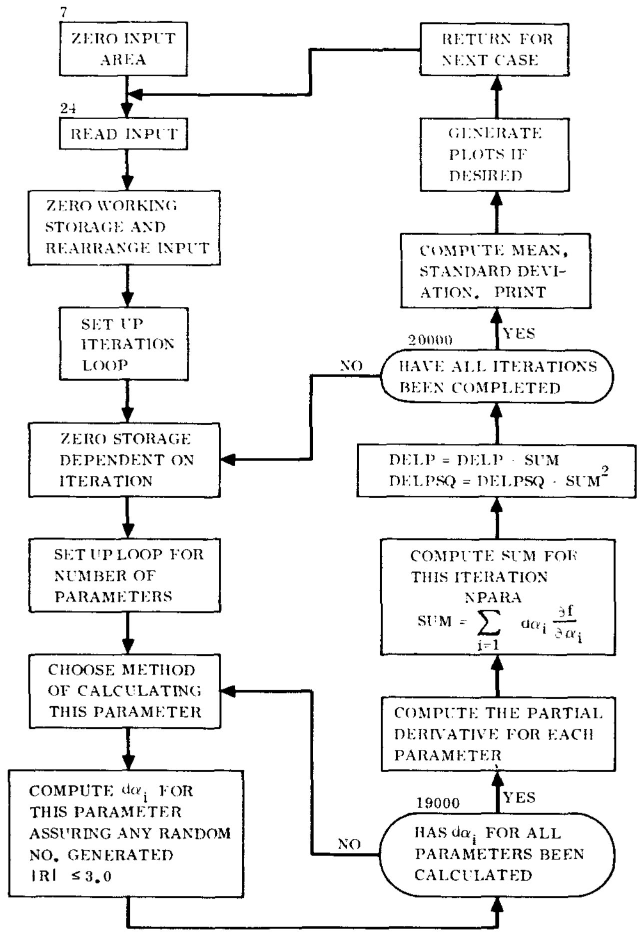


Figure 1. Generalized Computer Flow Chart

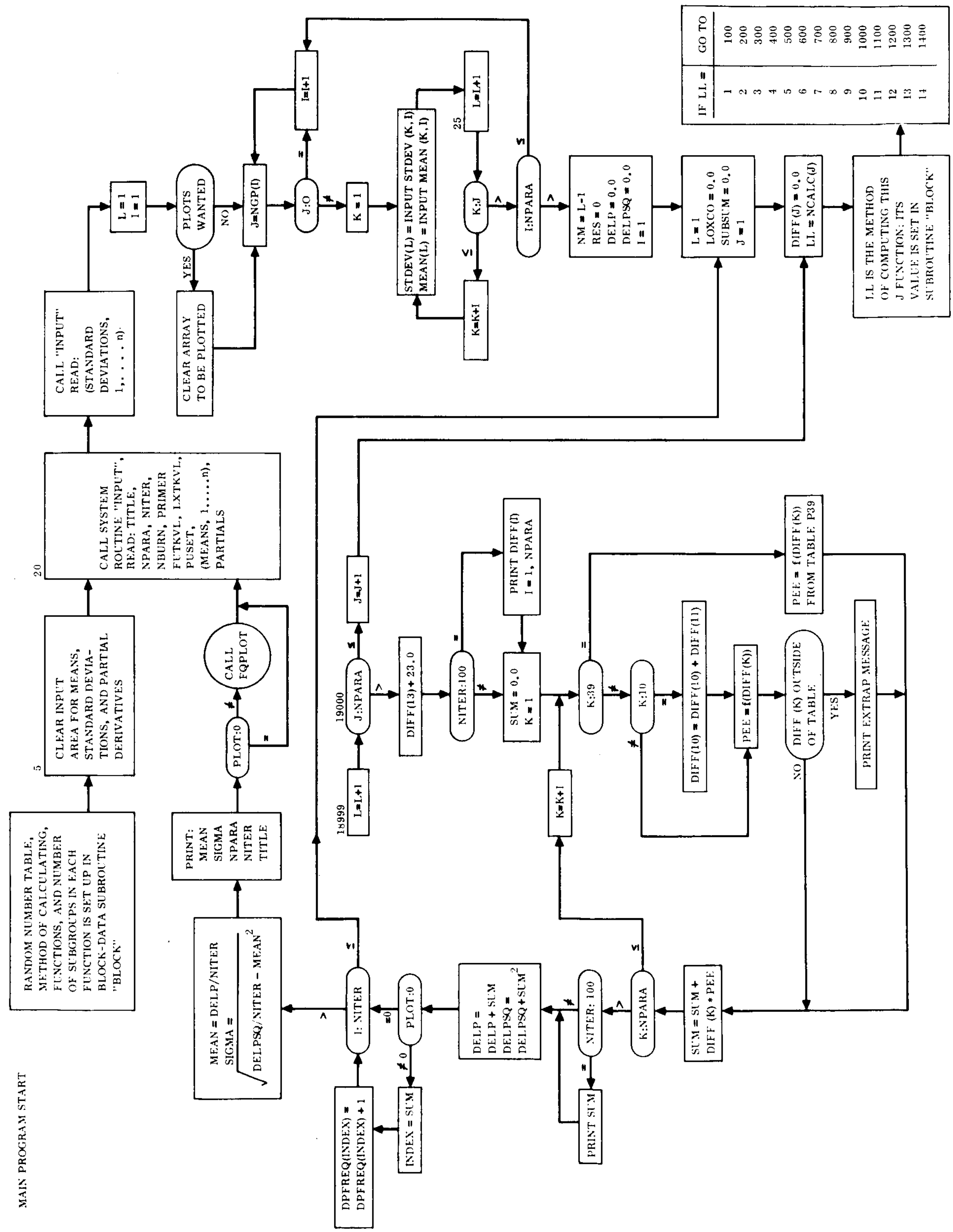
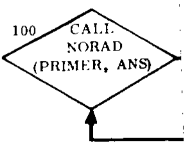
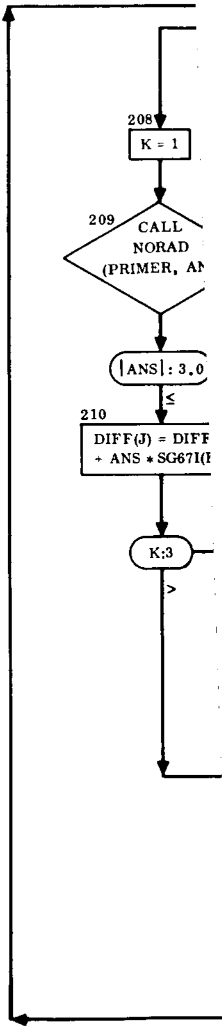


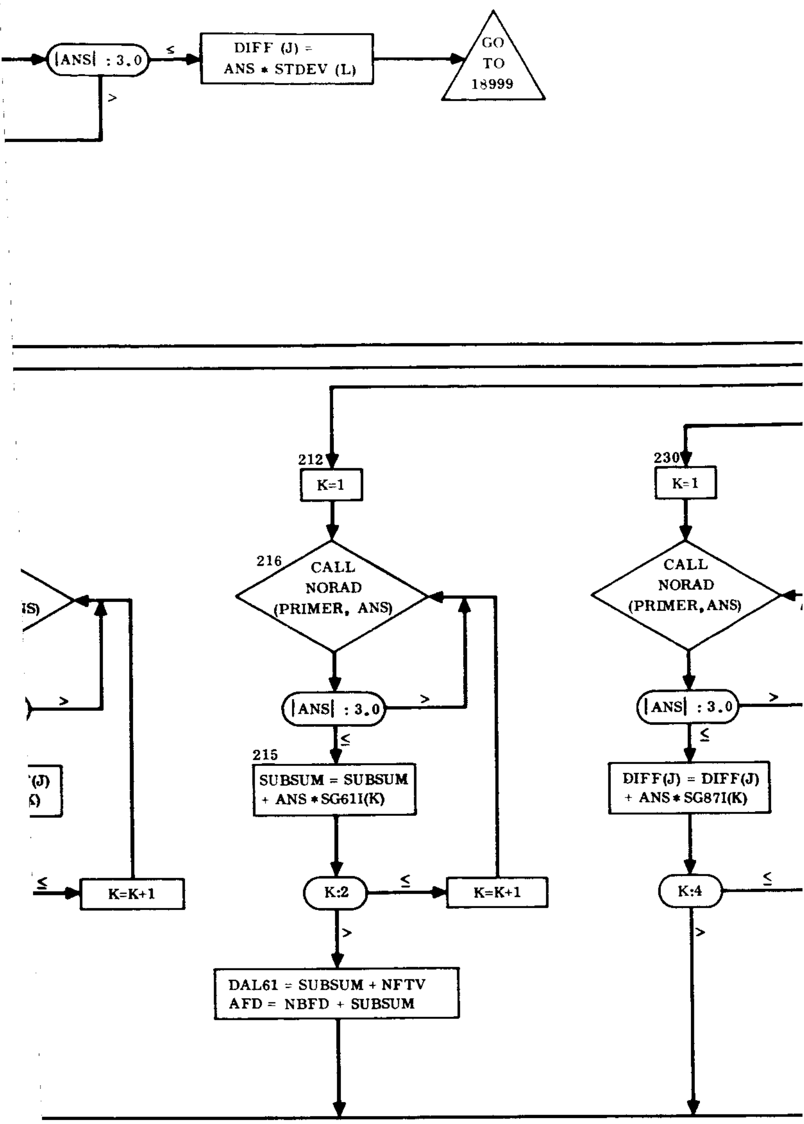
Figure 2. Detailed Flow Chart (Sheet 1 of 4)

FOR LL   1



FOR LL = 2





|ANS| : 3.0
≤
>
DIFF (J) = ANS * STDEV (L)
GO TO 18999
212
K=1
216
CALL NORAD (PRIMER, ANS)
|ANS| : 3.0
>
≤
215
SUBSUM = SUBSUM + ANS * SG61I(K)
K:2
≤
K=K+1
>
DAL61 = SUBSUM + NFTV
AFD = NBFD + SUBSUM
230
K=1
CALL NORAD (PRIMER, ANS)
|ANS| : 3.0
>
≤
DIFF(J) = DIFF(J) + ANS * SG87I(K)
K:4
≤
>
K=K+1

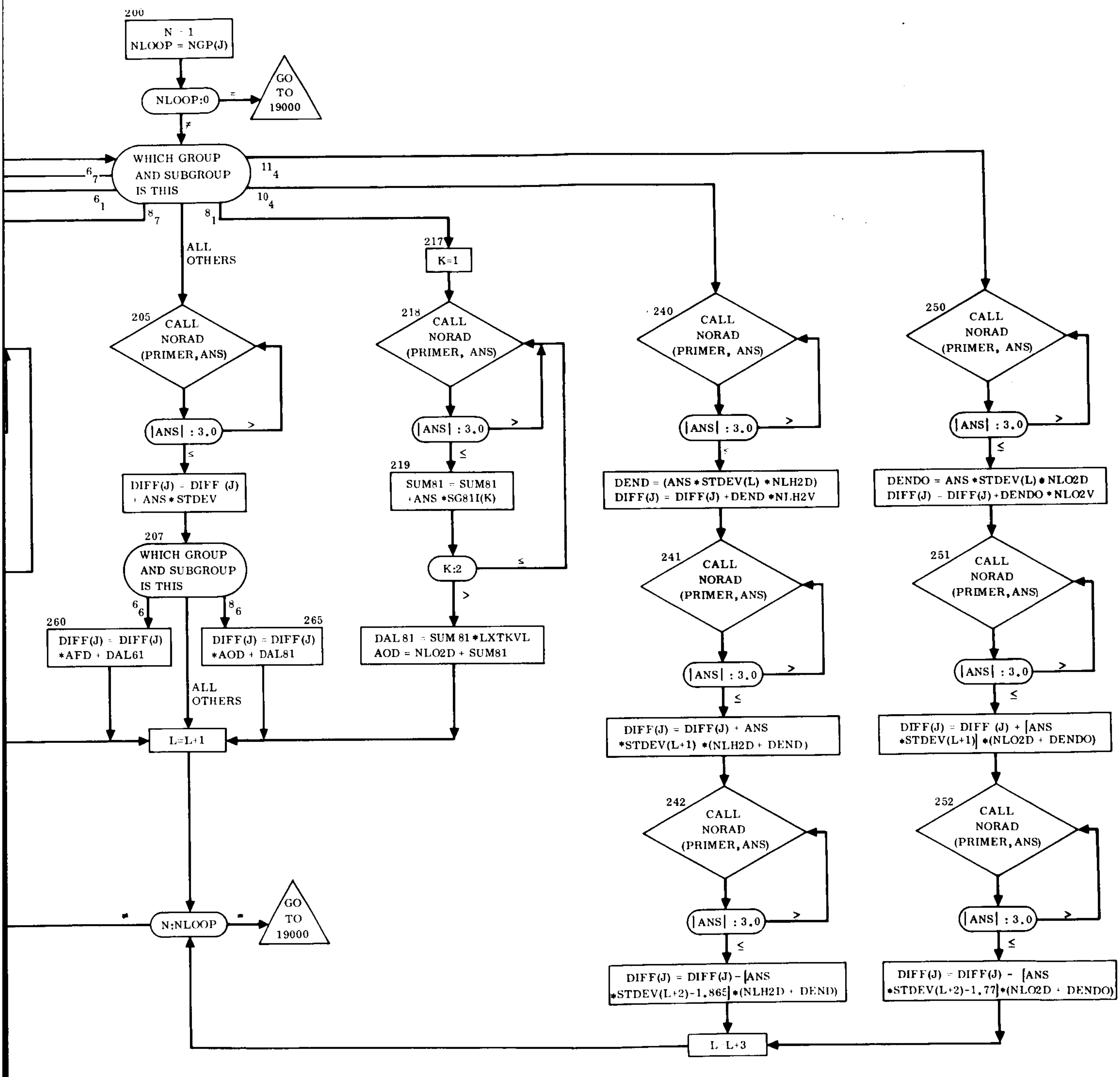
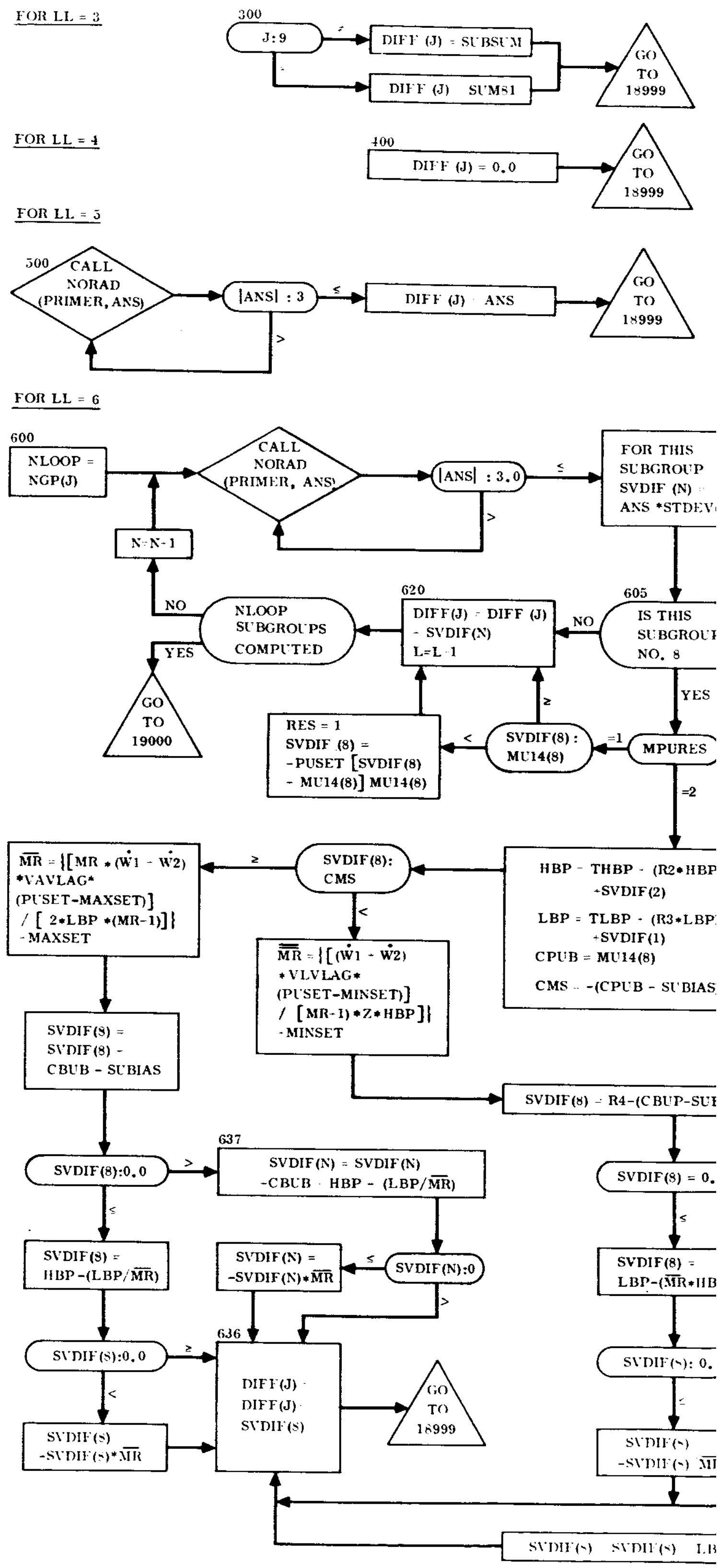
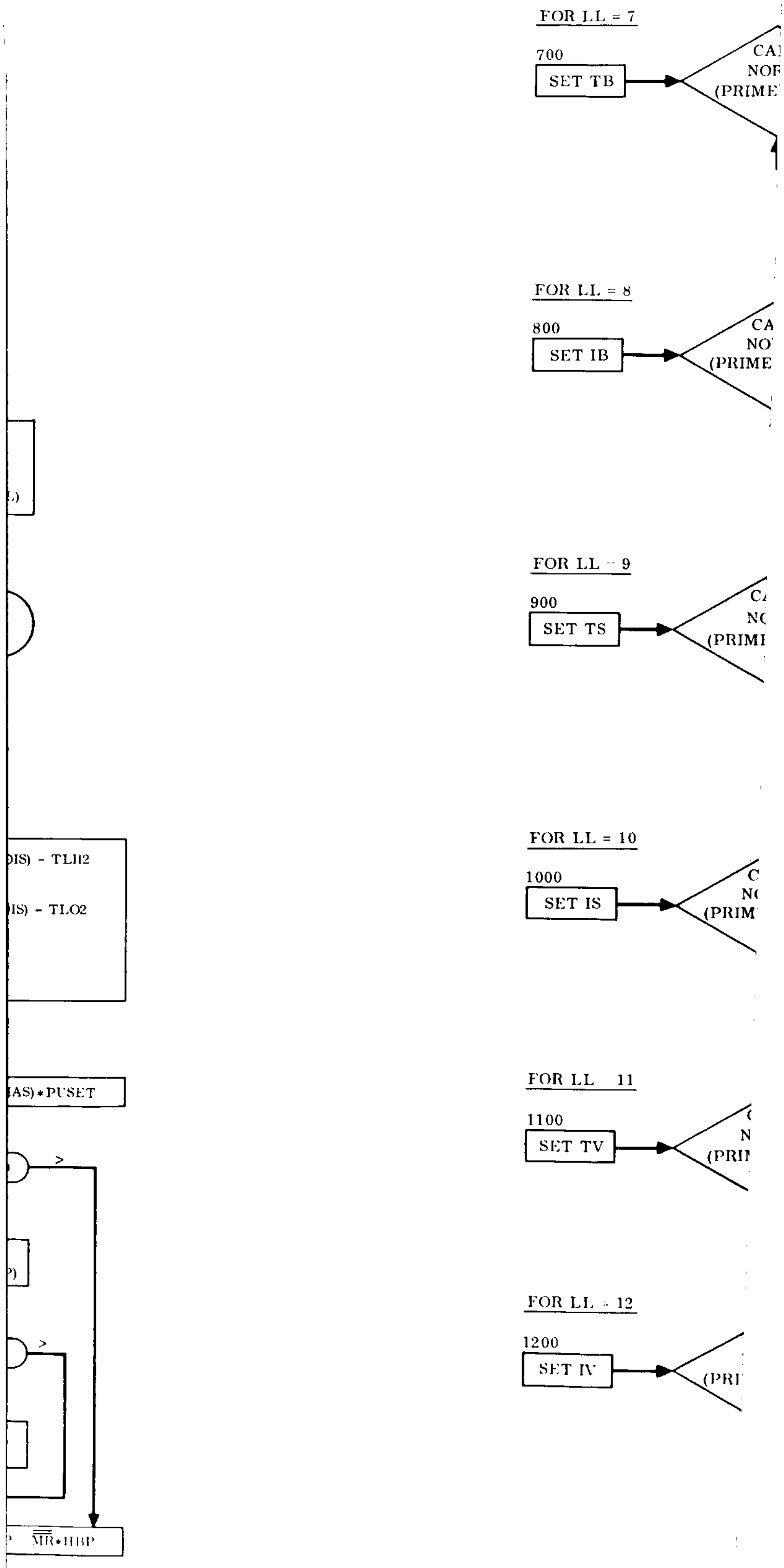


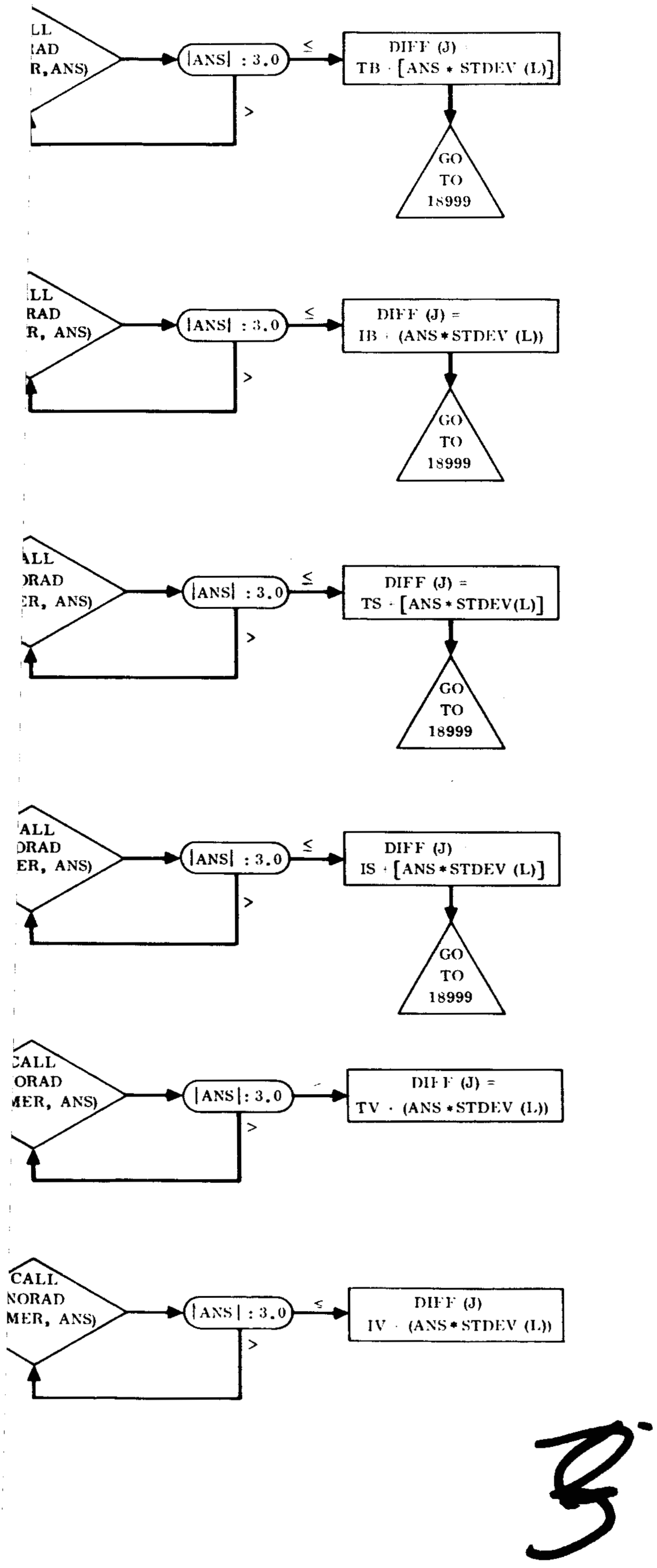
Figure 2. Detailed Flow Chart (Sheet 2 of 4)

3


FOR LL = 3
300
J:9
DIFF (J) = SUBSUM
DIFF (J) SUMS1
GO TO 18999
FOR LL = 4
400
DIFF (J) = 0.0
GO TO 18999
FOR LL = 5
500
CALL NORAD (PRIMER, ANS)
|ANS| : 3
DIFF (J) ANS
GO TO 18999
FOR LL = 6
600
NLOOP = NGP(J)
CALL NORAD (PRIMER, ANS)
|ANS| : 3.0
FOR THIS SUBGROUP SVDIF (N) = ANS *STDEV
N=N-1
NO
NLOOP SUBGROUPS COMPUTED
YES
GO TO 19000
620
DIFF(J) = DIFF (J) - SVDIF(N) L=L-1
NO
605
IS THIS SUBGROUP NO. 8
YES
RES = 1 SVDIF (8) = -PUSET [SVDIF(8) - MU14(8)] MU14(8)
SVDIF(8): MU14(8)
MPURES
=1
=2
HBP = THBP - (R2*HBP +SVDIF(2)
LBP = TLBP - (R3*LBP +SVDIF(1)
CPUB = MU14(8)
CMS = -(CPUB - SUBIAS
SVDIF(8): CMS
MR = {[MR * (W1 - W2) *VAVLAG* (PUSET-MAXSET)] / [2*LBP *(MR-1)]} -MAXSET
MR = {[(W1 - W2) *VLVLAG* (PUSET-MINSET)] / [MR-1)*Z*HBP]} -MINSET
SVDIF(8) = SVDIF(8) - CBUB - SUBIAS
SVDIF(8) = R4-(CBUP-SU
637
SVDIF(8):0.0
SVDIF(N) = SVDIF(N) -CBUB HBP - (LBP/MR)
SVDIF(8) = 0.
SVDIF(8) = HBP-(LBP/MR)
SVDIF(N) = -SVDIF(N)*MR
SVDIF(N):0
SVDIF(8) = LBP-(MR*HB
636
SVDIF(8):0.0
DIFF(J) = DIFF(J) SVDIF(8)
GO TO 18999
SVDIF(8): 0.
SVDIF(8) -SVDIF(8)*MR
SVDIF(8) -SVDIF(8) MR
SVDIF(8) SVDIF(8) LB


FOR LL = 7
700
SET TB
FOR LL = 8
800
SET IB
FOR LL = 9
900
SET TS
FOR LL = 10
1000
SET IS
FOR LL 11
1100
SET TV
FOR LL = 12
1200
SET IV
IS) - TLH2
IS) - TLO2
AS)*PUSET
MR*HBP

Figure



2. Detailed Flow Chart (Sheet 3 of 4)

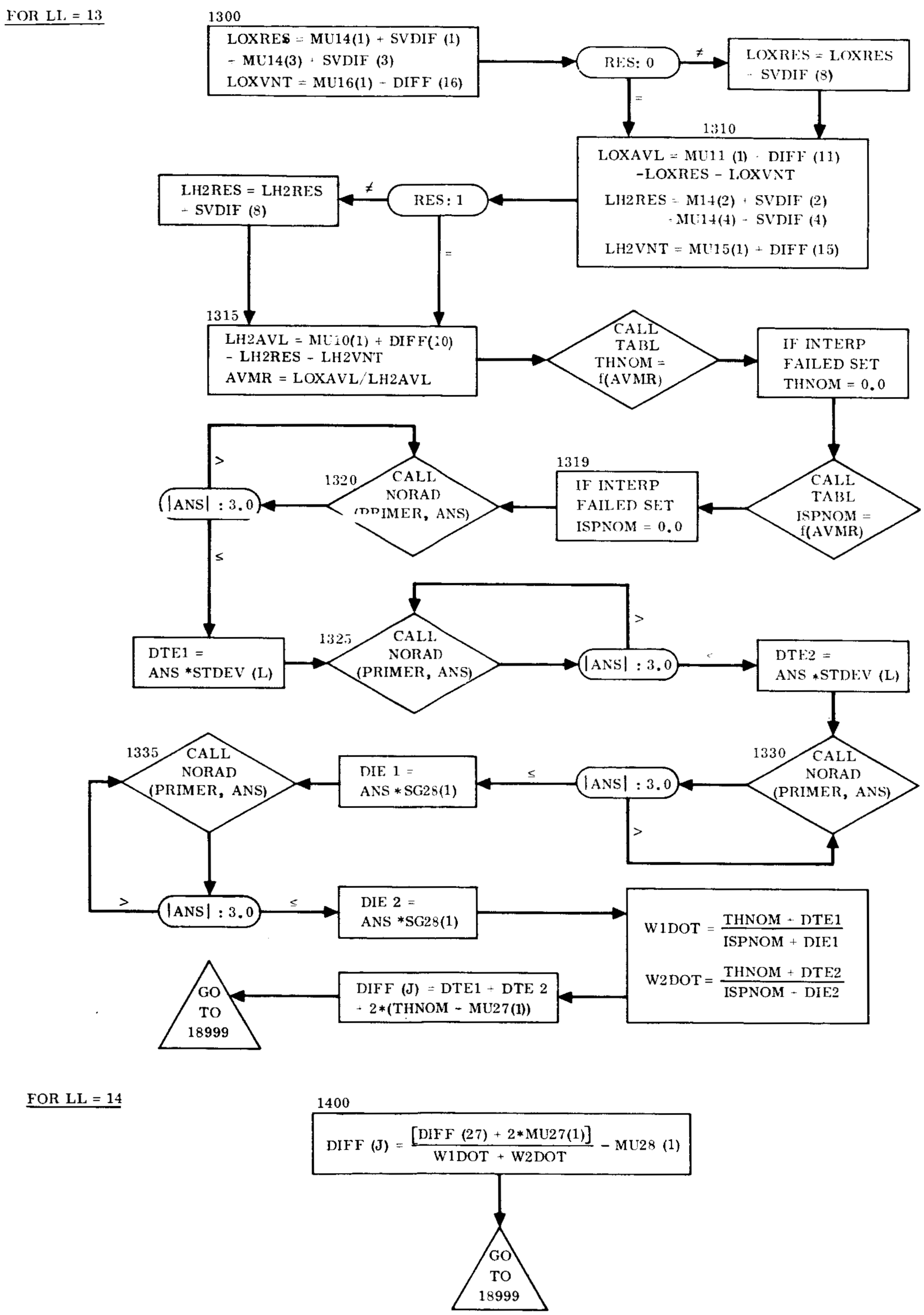


Figure 2. Detailed Flow Chart (Sheet 4 of 4)

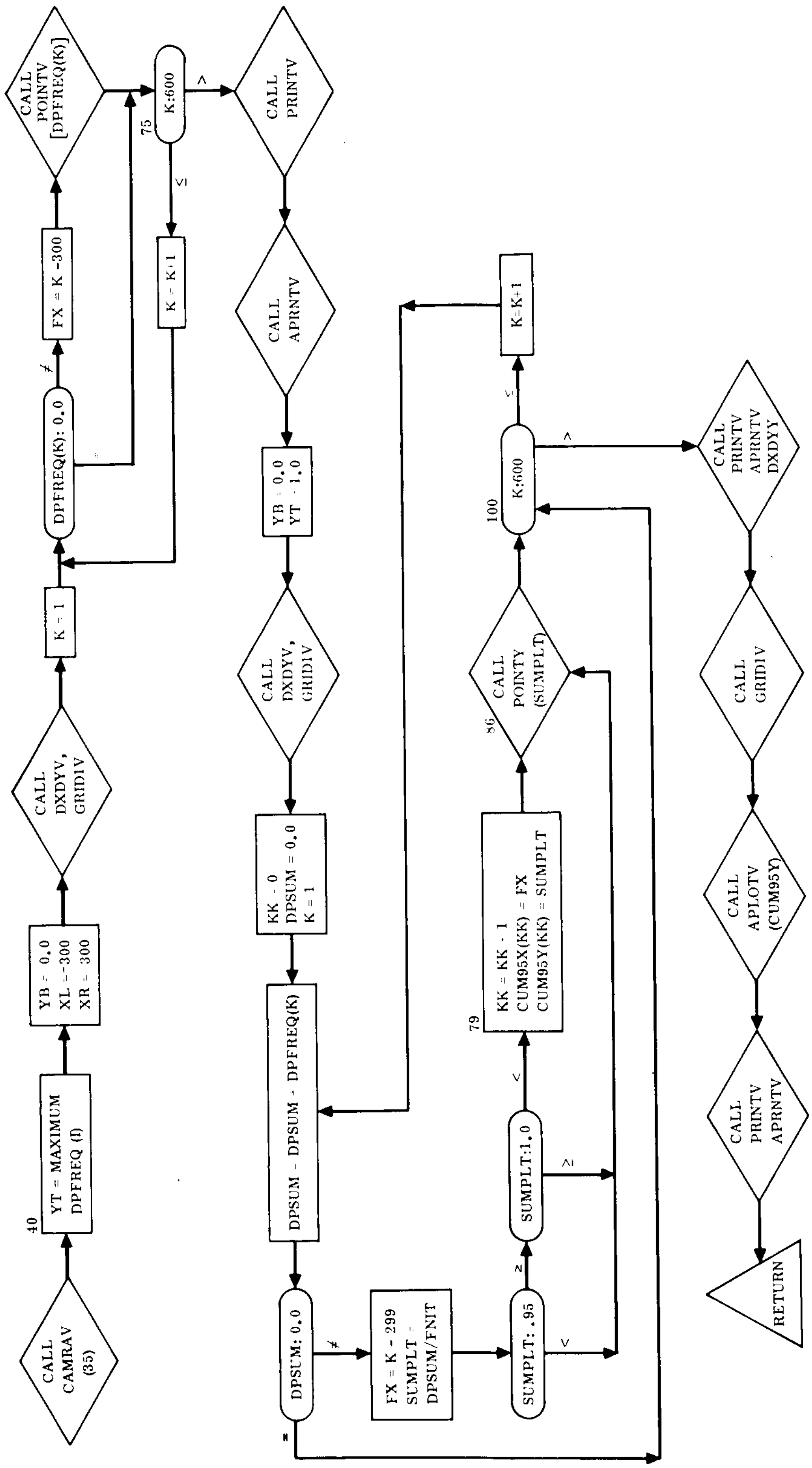


Figure 3. Subroutine FQPLOT

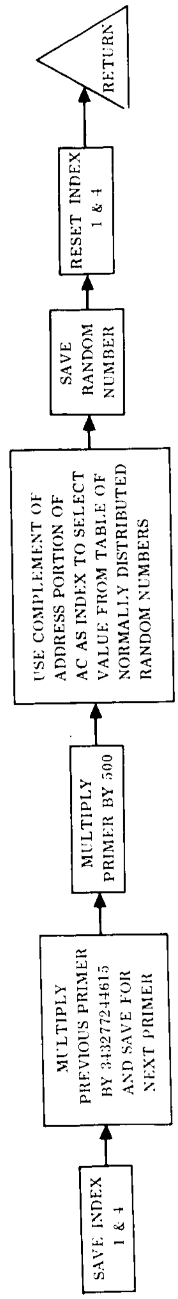


Figure 4. Subroutine NORAD

3.1 INPUT VARIABLES. The input variables, Figure 5, are defined in the following paragraph. (The names of the variables are compatable with the nomenclature used in Report GD/C-BTD65-143 where possible.)

```
TITLE=120H    FPR FREQUENCY FUNCTION ● 10000 ITERATIONS ● 9-30-65
  FPR PROBABILITY FUNCTION ● 10000 ITERATIONS * 9-30-65   .
 NPARA=39,
NITER=10000.              PRIMER = $213536715425  ,
PLOT = 0 ,
P(1)=         -30.0,      30.0,       .0741,      .0328,                1
              -69.0,      69.0,       +.0932,     +.0874,
              -59.0,      59.0,        .9385,      .9196,
              -75.0,      75.0,       .0681,      .0810,                4
              -48.0,      48.0,       .0609,      .0548,                5
              -847.0,     847.0,      -.0288,     -.0181,               6
              -.55,       .55,        -36.1342,   -16.1405,             7
              -1269.0,    1269.0,     -.0123,     -.0087,               8
              -.4635,     .4635,      -46.9105,   -36.7873,             9
              -373.0,     470.0,      -.0755,     -.0709,               10
F(45)=        -87.0,      87.0,       .0703,      .0690,                12
              -259.0,     276.0,      .1099,      .1066,                13
              -100.0,     100.0,      1.0,        1.0,                  14
F(69)=        -.023,      .023,       677.0043,   -1121.3739,           18
              -3000.0,    3000.0,     -.004,      -.004,                19
              -2.4,       2.4,        -18.8711,   -18.695,              20
F(85)=        -855.0,     855.0,      .0091,      .0092,                22
              -2.8,       2.8,        -15.421,    -15.0036,             23
P(105)=       -424.0,     424.0,      -.0099,     -.0114;               27
              -3.54,      3.54,       -25.9483,   -25.9252,             28
F(137)=       3.0,        -3.0,       7.3709,     6.3237,               35
              -2.0,       2.0,        -4.4804,    3.6507,               36
              -5.0,       5.0,        -11.2546,   -5.6079,              37
              -5.0,       5.0,        .0139,      .0978,                38
P39=          -3.0,  0.0,  3.0,  -.5767,  -11.4989,  -2.8836,           39
MU10=5056..
MUI1= 25035..
MU14(8)=9.0.
MU27=15011.86 ,
FUSET=5.0,
      NFTV=1500.. NOTV=2500.. NLH2D= 4.22, NLH2V= 1254.17, NLO2D= 68.78,
      NLO2V= 369.1, NBFD= 50.. NBOD= 68.78,
MU28 = 432.26777*
                                                  RECORD BLOCK 1
---------------------------------------------------------------------
MPURES =2,
LBPDIS=11.3333, LBP=1001.0, TLO2=68.0, HBP=261.0, TLH2=66.0,
SUBIAS= 9.00 , VLVLAG = 5.0, MAXSET = 5.55, MINSET = 4.39,HBPDIS=2.3333,
SG1= 10..           SG3=20.. SG4=25.. SG5=16..
      SG2    =16.666,16.3333,
 SG6= 0.0,.4,.6,.08666,.76666,.66666,.0,2.0,
SG7= .017, SG8= 0.0,.5333,.583333,.4,2.0833,1.4666,0.0,3.0,
SG9= .14,  SG10 =  1., 1., 1., .01   , 4.2167  ,  1.095  .
SG11= 11., 11., 5., .003333  , 1.25333  ,  .81666, SG12= 17., 23.,6.,4..
SG13= 20., 52., 2., 2., 1., 2., 44.0,41.66666,0.0,
SG14= 1., 1., 12., 5., 5., 0., 4., 8.33333,
SG15=4.8, SG16=9.0,
SG18 = .008, SG19 = 1000., SG20 = .8, SG22 = 285., SG23 = .9,
SG27=70.66666, SG28=1.66666,  SG35=1.0,
SG36 = .07, .42, .04, .51,  SG37 = .33, .83, 1.17, .75,
SG611= .15, .104,
SG811= .05, .13666,
SG671    =16.6666,28.6666,6.6666,
SG871    =66.6666,50.0,65.3333,10.0,
                                                  RECORD BLOCK 2
                                                  CASE 1
---------------------------------------------------------------------
SG38 = 1.7,  SG39 = 1. ●
MU10 = 5040.0,  MU14(8) = 25.0 *
                                                  BLOCK 1
---------------------------------------------------------------------
*
                                                  BLOCK 2
                                                  CASE 2
---------------------------------------------------------------------
MU10 = 5020.0,  MU14(8) = 45.0 *
                                                  BLOCK 1
---------------------------------------------------------------------
*
                                                  BLOCK 2
                                                  CASE 3
---------------------------------------------------------------------
```

Figure 5. Sample Input

The names of the input variables are defined as they appear in Figure 5.

Record Block 1

TITLE - One hundred and twenty columns of alphanumeric information. The first 60 columns are used as a title on the plot of the frequency function. The last 60 are used as a title on the plot of the cumulative function.

NPARA - (Integer). The number of parameters.

NITER - (Integer). The number of iterations in the Monte-Carlo simulation (NITER < 32,468).

PLOT - (Integer). If set nonzero, plots will be generated.

PRIMER - A 12-digit octal number. The first value used by the random-number generator. The first digit must be either a 0, 1, 2, or 3. The last must be a 5. On the data card it must be preceded by a dollar sign ($).

(The remaining input items in Record Block No. 1 are all floating point numbers.)

P - Partial derivatives (4 • NPARA values). These are entered as a block of numbers with four values per parameter. The index for the first independent variable for a particular parameter is

$$N = [(\text{Parameter No.} - 1) \bullet 4] + 1$$

MU1 through MU39 - The mean of each parameter. Each parameter can be composed of 2 to 10 subgroups. If one wished to assign a mean of 50 to the eighth subgroup of parameter 10, the input would be MU10(8) = 50,.

NFTV - Nominal booster fuel-tank volume.

NOTV - Nominal booster-oxidizer-tank volume.

PUSET - Nominal propellant-utilization mixture-ratio setting.

NLO2V - Nominal LO_2 volume.

NLO2D - Nominal LO_2 density.

NLH2V - Nominal LH_2 volume.

NLH2D - Nominal LH_2 density.

NBFD - Nominal booster fuel density.

NBOD - Nominal booster oxidizer density.

P39 - Partial derivative for the 39th parameter. Three independent variables, then the three dependent variables.

Record Block 2

SG1 through SG39 - The standard deviations for each parameter. Each parameter can have up to 10 subgroups.

SG61I = Standard deviation for $d\alpha_{61_i}$.

SG67I - Standard deviation for $d\alpha_{67_i}$.

SG81I - Standard deviation for $d\alpha_{81_i}$.

SG87I - Standard deviation for $d\alpha_{87_i}$.

SUBIAS - Amount of PU bias below LH_2 probe.

VLVLAG - Average time required for the PU system mixture control valve to travel from its position to minimum or maximum stops.

MAXSET - Maximum PU system mixture-ratio valve setting.

MINSET - Minimum PU system mixture-ratio valve setting.

HBP - Hydrogen below the probe.

HBPDIS - Dispersion associated with HBP.

LBP - LO_2 below the probe.

LBPDIS - Dispersion associated with LBP.

MPURES - (Integer). Method of computing PU residuals. = 1 without end effect; = 2 with end effect.

3.2 OUTPUT. Normal output consists of the mean and standard deviations for each case.

For debugging purposes, if NITER is set equal to 100, the NPARA values of the $d\alpha_i$'s are printed for each iteration, and the sum of

$$\sum_{i=1}^{NPARA} d\alpha_i \frac{\partial f}{\partial \alpha_i}$$

for each iteration is also printed.

If the option to generate plots is exercised the distributions compiled by each case considered will be plotted.

3.3 VARIABLES USED WITHIN THE PROGRAM. A brief explanation is given here of each name used in each routine and its type of dimensioning if appropriate (e.g., a doubly subscripted array would be represented as ABC (N, N)).

Input variables are not included in this section (see Section 3.1).

Main Program Start

AFD - Average fuel density (Atlas).

ANS - The random number supplied to START from subroutine RANDOM.

AOD - Average oxidizer density (Atlas).

CPUB - MU14(8).

CMS - -(CPUB-SUBIAS).

DAL61 - SUBSUM * NFTV.

DAL81 - SUM81 * NOTV.

DELP - $\sum_{i=1}^{NITER} \left(\sum_{k=1}^{NPARA} \frac{\partial f}{\partial \alpha_k} d\alpha_k \right)_i$

DELPSQ - $DELP^2$.

DEND - Fuel density dispersion (Centaur).

DENDO - LO_2 density dispersion (Centaur).

DIE1 - Dispersion and ISP for Engine No. 1.

DIE2 - Dispersion and ISP for Engine No. 2.

DIFF(I) - $d\alpha_i$, NPARA values, NITER sets.

DPFREQ(N) - Frequency of occurance of values of SUM over NITER iterations. This array is for plotting.

DTE1 - Thrust dispersion for Engine No. 1.

DTE2 - Thrust dispersion for Engine No. 2.

DUMSD (M,N) - A dummy array equivalenced to input standard deviation array for generating STDEV array.

FM - Final value computed for the mean after NITER iterations (output).

FNIT - Floating point value for NITER.

IB - Booster ISP.

INDEX - Integer value for SUM used as an index to compile DPFREQ.

IS - Sustainer ISP.

ISPNOM - Thrust/flow rate.

IV - Vernier ISP.

I - Index in iteration loop.

J - Index of NPARA loop.

K - Index used for small loops within the program.

L - Index for standard deviations.

LH2AVL - Liquid hydrogen available.

LH2VNT - Liquid hydrogen vented.

LH2RES - Liquid hydrogen residual.

LL - Value set in BLOCK DATA to determine which of the 14 methods a particular function uses.

LOXRES - LOX residuals.

LOXVNT - LOX vented.

MR - Mixture ratio.

N - Index controlling the number of subfunctions for each of the NPARA functions.

NLOOP - Number of subfunctions in each function. NLOOP = NGP(J).

PEE - Table look-up value for partial derivatives. PEE = f (DIFF).

RES - Set nonzero if $d\alpha_{14_8} < \mu_{14_8}$.

STDEV(N) - A packed array of the input standard deviations.

SIGMA - Final standard deviation (output).

SUM - $\sum_{k=1}^{NPARA} \frac{\partial f}{\partial \alpha_k} d\alpha_k$.

SUBSUM - $\sum_{i=1}^{2} d\alpha_{6_{1_i}}$.

SUM81 - $\sum_{i=1}^{2} d\alpha_{8_{1_i}}$.

SVDIF - Array of $d\alpha$ for Function No. 14.

R2 - The random number associated with HBPDIS.

R3 - The random number associated with LBPDIS.

R4 - The random number associated with LH_2 probe uncovery.

THNOM - Nominal thrust.

TB - Booster thrust.

TS - Sustainer thrust.

TV - Vernier thrust.

W1DOT - Flow rate for Engine No. 1.

W2DOT - Flow rate for Engine No. 2.

MRBAR - Mixture ratio associated with LO_2 probe uncovery.

MRDBAR - Mixture ratio associated with LH_2 probe uncovery.

Subroutine BLOCK DATA

NGP - NPARA values giving the number of subfunctions within each function.

NCALC - The method of computing each function.

TICEN - A table of nine values used in computing THNOM and ISPNOM.

TABLE - Five hundred normally-distributed random numbers.

Subroutine FQPLOT

CUM95X - X array for plot of cumulative function from 0.95 to 1.0.

CUM95Y - Y array for plot of cumulative function from 0.95 to 1.0.

DPSUM - $\sum_{i=1}^{600} DPFREQ_i$.

FX - X value for frequency and cumulative plots.

DPFREQ - Y array for frequency plot.

SUMPLT - Y value for cumulative plot.

YB - Minimum value of Y for grid.

YT - Maximum value of Y for grid.

XL - Minimum value of X for grid.

XR - Maximum value of X for grid.

3.4 SAMPLE CASE. An evaluation of the Flight Performance Reserve for the Atlas/ Centaur vehicle is presented as the sample case. The case selected has a PU accuracy of ±25 pounds of LH_2 and a PU bias of 9 pounds of LH_2.

The input for each parameter and the program name of each input quantity, where applicable, are shown in Table 2.

The output is presented in Figures 6 through 9, which include plotted data. Figure 6 is the actual machine output. The input variables are listed on the output exactly as they appear on the input cards. The computed values that appear on the machine output are the mean and standard deviations of the vehicle's performance.

Table 2. Parameter-Controlled Input

PARAMETER	PROGRAM NAME (SG = Standard Deviation MU = Mean)	INPUT QUANTITY
Booster	SG1	10.0
Sustainer and Interstage	SG2(1)	16.6666
Adapter	SG2(1)	16.3333
Centaur	SG3	20.0
Nose Fairing	SG4	25.0
Insulation Panels	SG5	16.0
Fuel Weight (Booster)		
Fuel Density	SG61I(1)	0.15
	SG61I(2)	0.104

Table 2. Parameter-Controlled Input (Contd)

PARAMETER	PROGRAM NAME (SG = Standard Deviation MU = Mean)	INPUT QUANTITY
Probe Location	SG6(2)	0.4
Surface Level Variation	SG6(3)	0.6
Tank Pressure	SG6(4)	0.08666
Tank Volume	SG6(5)	0.76666
Tanking Level	SG6(6)	0.66666
Ground Expended	SG67I(1)	16.6666
	SG67I(2)	28.6666
	SG671(3)	6.6666
Sustainer Thrust Decay	SG6(8)	2.0
Fuel Density	SG7	0.017
Oxidizer Weight (Booster)		
Oxidizer Density	SG81I(1)	0.05
	SG81I(2)	0.13666
Sensor Location	SG8(2)	0.5333
Surface Level Variation	SG8(3)	0.58333
Tank Pressure	SG8(4)	0.4
Tank Volume	SG8(5)	2.0833
Tanking Level	SG8(6)	1.4666
Ground Expended	SG87I(1)	66.6666
	SG87I(2)	50.0
	SG87I(3)	65.3333
	SG87I(4)	10.0
Thrust Decay	SG8(8)	3.0
Oxidizer Density	SG9	0.14
Fuel Weight (Centaur)		
Sensor Sensitivity	SG10(1)	1.0
	MU10	5056.0
Sensor Location	SG10(2)	1.0

Table 2. Parameter-Controlled Input (Contd)

PARAMETER	PROGRAM NAME (SG = Standard Deviation MU = Mean)	INPUT QUANTITY
Surface Variation	SG10(3)	1.0
Density	SG10(4)	0.01
Tank Volume	SG10(5)	4.2167
Tank Ullage	SG10(6)	1.095
Oxidizer Weight (Centaur)		
Sensor Sensitivity	SG11(1)	11.0
	MU11	25035.0
Sensor Location	SG11(2)	11.0
Surface Variations	SG11(3)	5.0
Density	SG11(4)	0.003333
Tank Volume	SG11(5)	1.25333
Tank Ullage	SG11(6)	0.81666
Booster Jettisoned Residuals		
Trapped Fuel	SG12(1)	17.0
Trapped Oxidizer	SG12(2)	23.0
Lube Oil	SG12(3)	6.0
Helium	SG12(4)	4.0
Sustainer Jettisoned Residuals		
Trapped Fuel	SG13(1)	20.0
Trapped Oxidizer	SG13(2)	52.0
Lube Oil	SG13(3)	2.0
Helium	SG13(4)	2.0
Nitrogen	SG13(5)	1.0
GO_2 in Tank (Flight)	SG13(6)	2.0
GO_2 in Tank (Ground)	SG13(7)	44.0
PU Bias	SG13(8)	41.6666

Table 2. Parameter-Controlled Input (Contd)

PARAMETER	PROGRAM NAME (SG = Standard Deviation MU = Mean)	INPUT QUANTITY
Centaur Jettisoned Residuals		
Trapped LO_2	SG14(1)	1.0
Trapped LH_2	SG14(2)	1.0
GO_2 in Tank	SG14(3)	12.0
GH_2 in Tank	SG14(4)	5.0
H_2O_2 Weight	SG14(5)	5.0
Helium	SG14(6)	0.0
Ice and Frost	SG14(7)	4.0
PU	SG14(8)	8.3333
	MU14(8)	9.0
Centaur Venting		
Ground and Inflight LH_2	SG15	4.8
Ground and Inflight LO_2	SG16	9.0
Booster Propulsion		
Booster Mixture Ratio	SG18	0.008
Booster Thrust	SG19	1000.0
Booster ISP	SG20	0.8
Sustainer Thrust	SG22	285.0
Sustainer ISP	SG23	0.9
Centaur Propulsion		
Thrust	SG27	70.6666
	MU27	15011.86
Specific Impulse (ISP)	SG28	1.6666
	MU28	423.26777
Atmosphere	SG35	1.0
Launch Azimuth		
Null Voltage	SG36(1)	0.07

Table 2. Parameter-Controlled Input (Contd)

PARAMETER	PROGRAM NAME (SG = Standard Deviation MU = Mean	INPUT QUANTITY
Roll-Gyro Torquing Rate	SG36(2)	0.42
Time Uncertainties	SG36(3)	0.04
Allowed Tolerance	SG36(4)	0.51
Pitch Program		
Voltage-Time Integral	SG37(1)	0.33
Gyro-Torquing-Rate Voltage Time Average	SG37(2)	0.83
Inverter Voltage	SG37(3)	1.17
Inverter Frequency	SG37(4)	0.75
Drag Force	SG38	1.7
Wind Profile	SG39	1.0

NOTE: The remaining input variables have been defined in Section 3.1. Their names remain constant in both internal and external references to the program.

```
TITLE=120H    FPR FREQUENCY FUNCTION * 10000 ITERATIONS * 9-30-65
  FPR PROBABILITY FUNCTION * 10000 ITERATIONS * 9-30-65    ,
 NPARA=39,
NITER=10000,               PRIMER = $213536715425  ,
PLOT = 0 ,
P(1)=         -30.0,       30.0,       .0741,       .0328,
              -69.0,       69.0,       +.0932,      +.0874,
              -59.0,       59.0,        .9385,       .9196,
              -75.0,       75.0,       .0681,       .0810,
              -48.0,       48.0,       .0609,       .0548,
              -847.0,      847.0,      -.0288,      -.0181,
              -.55,        .55,        -36.1342,    -16.1405,
              -1269.0,     1269.0,     -.0123,      -.0087,
              -.4635,      .4635,      -46.9105,    -36.7873,
              -373.0,      470.0,      -.0755,      -.0709,
P(45)=        -87.0,       87.0,       .0703,       .0690,
              -259.0,      276.0,      .1099,       .1066,
              -100.0,      100.0,      1.0,         1.0,
P(69)=        -.023,       .023,       677.0043,    -1121.3739,
              -3000.0,     3000.0,     -.004,       -.004,
              -2.4,        2.4,        -18.8711,    -18.695,
P(85)=        -855.0,      855.0,      .0091,       .0092,
              -2.8,        2.8,        -15.421,     -15.0036,
P(105)=       -424.0,      424.0,      -.0099,      -.0114,
              -3.54,       3.54,       -25.9483,    -25.9252,
P(137)=       3.0,         -3.0,       7.3709,      6.3237,
              -2.0,        2.0,        -4.4804,     3.6507,
              -5.0,        5.0,        -11.2546,    -5.6079,
              -5.0,        5.0,        .0139,       .0978,
P39=          -3.0,  0.0,  3.0,  -.5767,  -11.4989,  -2.8836,
MU10=5056.,
MU11= 25035.,
MU14(8)=9.0,
MU27=15011.86 ,
PUSET=5.0,
      NFTV=1500., NOTV=2500., NLH2D= 4.22, NLH2V= 1254.17, NLO2D= 58.78,
      NLO2V= 369.1, NBFD= 50., NBOD= 68.78,
MU28 = 432.26777*
MPURES =1,
SG1= 10.,            SG3=20., SG4=25., SG5=16.,
      SG2    =16.666,16.3333,
 SG6= 0.0,.4,.6,.08666,.76666,.66666,.0,2.0,
SG7= .017, SG8= 0.0,.5333,.583333,.4,2.0833,1.4666,0.0,3.0,
SG9= .14,  SG10 =  1., 1., 1., .01    , 4.2167  ,  1.095   ,
SG11= 11., 11., 5., .003333  , 1.25333  ,  .81666, SG12= 17., 23.,6.,4.,
SG13= 20., 52., 2., 2., 1., 2., 44.0,41.66666,0.0,
SG14= 1., 1., 12., 5., 5., 0., 4., 8.33333,
SG15=4.8, SG16=9.0,
SG18 = .008, SG19 = 1000., SG20 = .8, SG22 = 285., SG23 = .9,
SG27=70.66666, SG28=1.66666,  SG35=1.0,
SG36 = .07, .42, .04, .51,  SG37 = .33, .83, 1.17, .75,
SG61I= .15, .104,
SG81I= .05, .13666,
SG67I    =16.6666,28.6666,6.6666,
SG87I    =66.6666,50.0,65.3333,10.0,
SG38 = 1.7,  SG39 = 1. *

               NUMBER OF PARAMETERS =   39                     NUMBER OF ITERATIONS =  10000


               MEAN OF FPR = -1.55287E 01                      STANDARD DEVIATION =  5.27740E 01
```

Figure 6. Machine Output

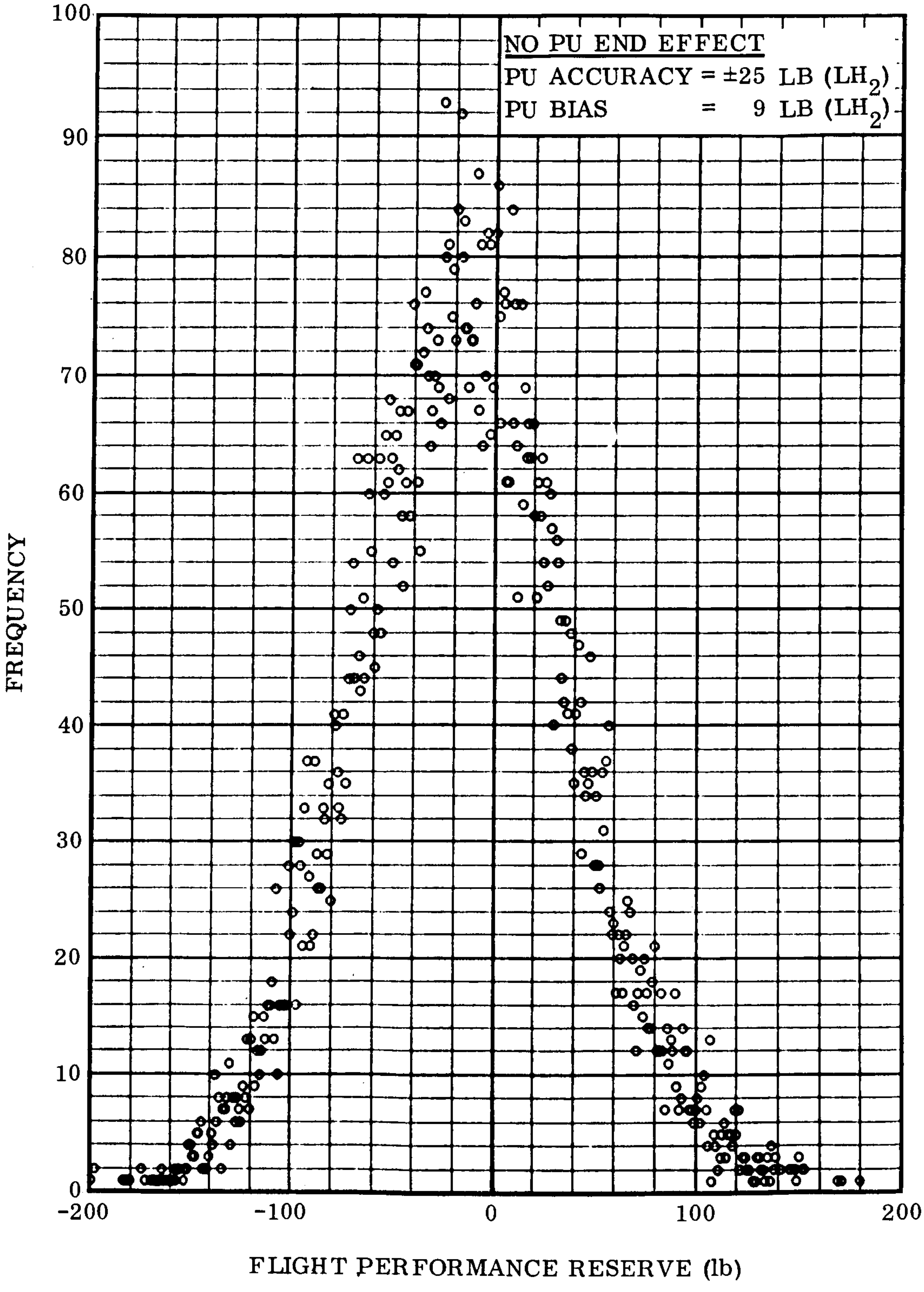


Figure 7. FPR Frequency Function

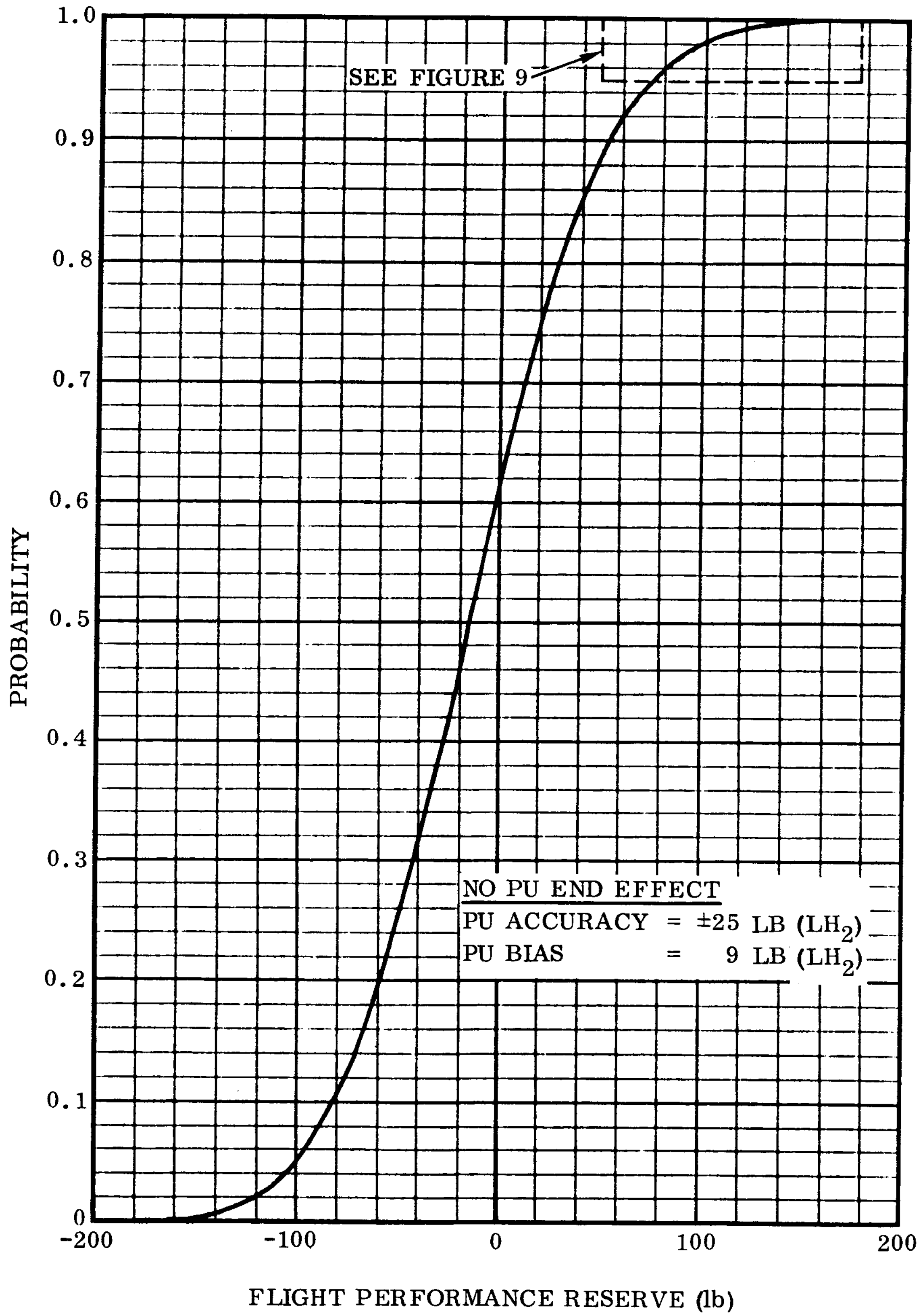


Figure 8. FPR Probability Function

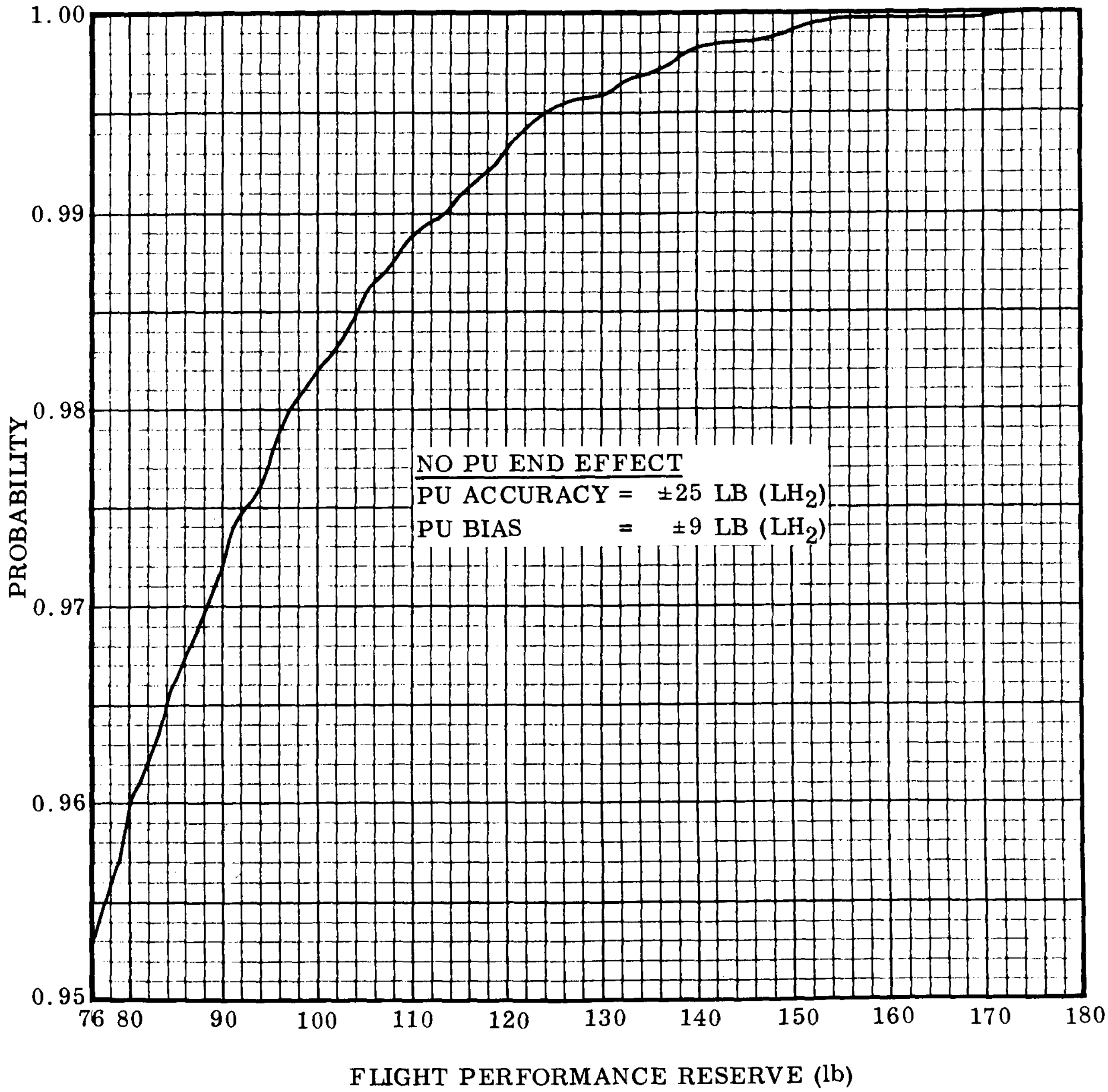


Figure 9. FPR Probability Function Segment

3.5 INPUT CARD AND DECK FORMAT. This program uses a systems routine "input" that greatly simplifies the handling of the input data. There is no rigid format for the data. No specific order is required in entering variables within each block. Card columns 1 to 72 are used, and more than one variable can appear on a single data card.

Because of the amount of data used by this program, it is separated into two blocks or records (these records are defined in the explanation of the input variables, Section 3.1).

3.5.1 Data Card Setup Rules. It is not necessary that variable names on the data cards appear in the same order as those in the calling sequence. The routine will search the list for the name and its core location.

Individual data items are separated by commas.

An equal sign or a comma separates the name of a variable and its first data item.

A comma separates the end of a data set and the next variable name.

A data input record is terminated by an asterisk (*).

It is not necessary to input a data set for each name in the calling sequence.

Elements of an array may be skipped by writing consecutive commas (i.e., no data between the commas) or by singly subscripting the array name. Double subscripting is illegal. Thus, if it is desired to input data into a three-element vector V, one could write

V = 2.79,,1.32.

No data would be entered into V(2). What was originally there remains there. Alternatively, it could be written

V(1) = 2.79, V(3) = 1.32

3.5.2 Additional Feature. The card image is normally written on the system output unit prior to being processed by the routine. If an N is punched in column 73, the card will not be listed. If column 73 contains a C, the card is treated as a comment only; i.e., it is not scanned for data. If the card contains CE in columns 73 - 74, the card will be treated as a comment card and a page will be ejected.

3.5.3 Multiple Cases. When running multiple cases, only those variables that change need be entered. All others remain unchanged.

3.5.4 Restrictions. The following errors will be detected by the subroutine, and a diagnostic message and the card in error will be printed on the system output unit:

a. Name on data card exceeds six characters.

b. Name on data card does not appear in the calling sequence.

c. Punctuation errors.

d. Octal field errors.

e. Decimal or octal data out of range.

3.5.5 Input Deck Setup. This program has a main deck START and three subroutines FQPLOT, BLOCK and RANDOM (see Figure 10). If the input value for PLOT is nonzero, a $SETUP card must be placed in front of the deck to ensure the mounting of a tape for generating S-C 4020 data. Its form is

Column 1: $SETUP

Column 8: LB4

Column 16: DISK, PLOT, SAVE

Also, a save-tape tag must accompany the run request (three for each case).

TAPE ACTION REQUEST — LOCATION

☐ SAVE ☒ PLOT ☐ PRINT ☐ PUNCH

PROGRAMMER	RUN	UNIT	EXT	DATE
Hayward	XXX	LB4	2191	7-29-65

SPECIAL INSTRUCTIONS — SECRET CONF. — BCD BIN

A3482 (7-64) 4020 PROCESSING 200, (556,) 800

CONT.	PROJ.	JOB	WAP	GROUP	COPIES	FRAMES
			K20006713	966	1	

☒ COPY-FLO ☐ THERMOFAX ☐ MICROMATE

The data card (placed between the last card of the program and the first input card) can be either 7-8 in column 1 or $DATA in column 1.

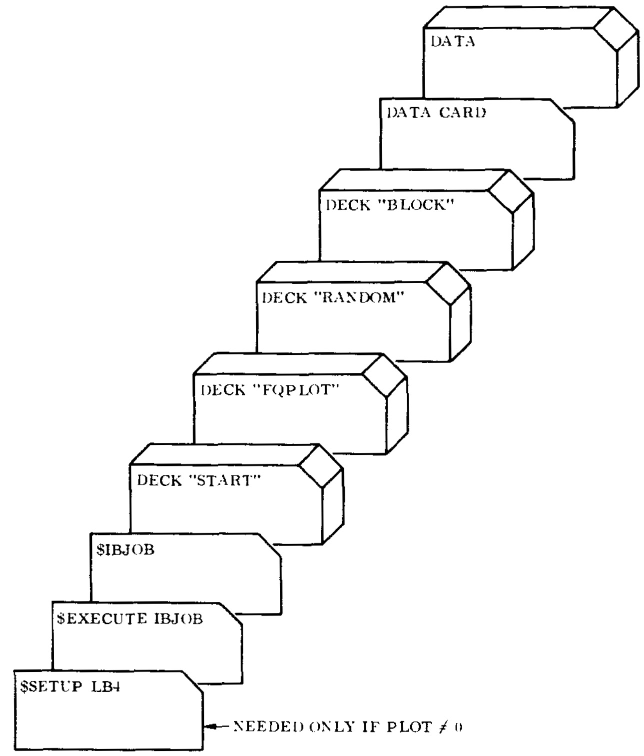


Figure 10. Input Deck Setup

3.5.6 Multiple Case Capability. As many cases as desired can be run on one pass through the computer. From the second case on, only the data that have changed from the previous run need be included in the input (see Figure 5).

Since there are two calls to the input routine, each case must have two asterisks. Even if there is no change in one of the sections, a card with an asterisk must be included.

3.5.7 Time and Line Estimate. The time required is proportional to the number of iterations. To be safe, allow five minutes for 10,000 iterations. If two or more cases are stacked together on one run, allow five minutes for the first case and four minutes for each additional case.

For normal output, allow 300+50*(number of cases) for the line estimate. If IRITE $\neq 0$, raise this estimate by 10,000.

APPENDIX

PROGRAM LISTING

```
$IBFTC START    FULIST,REF
      COMMON /BD/NGP,NCALC,P39,TICEN,TABLE
      DIMENSION NGP(300),NCALC(300),P39(6),TICEN(9),TABLE(500)
      COMMON DELP,DELPSQ,FM,SIGMA,FNIT ,NITER
      COMMON NM,  MR,THNOM,ISPNOM,FUTKVL,LXTKVL,PUSET,DPFREQ(600)
      COMMON STDEV(300),MEAN(300),P(600),DIFF(300),SVDIF(10),TITLE(20)
      DIMENSION DUMSD(10,200)
      EQUIVALENCE (DUMSD(1),SG1(1))
      COMMON MU1(10),MU2(10),MU3(10),MU4(10),MU5(10),MU6(10),MU7(10),
     1MU8(10),MU9(10),MU10(10),MU11(10),MU12(10),MU13(10),MU14(10),
     2MU15(10),MU16(10),MU17(10),MU18(10),MU19(10),MU20(10),MU21(10),
     3MU22(10),MU23(10),MU24(10),MU25(10),MU26(10),MU27(10),MU28(10),
     4MU29(10),MU30(10),MU31(10),MU32(10),MU33(10),MU34(10),MU35(10),
     5MU36(10),MU37(10),MU38(10),MU39(10)
      COMMON SG1(10),SG2(10),SG3(10),SG4(10),SG5(10),SG6(10),SG7(10),
     1SG8(10),SG9(10),SG10(10),SG11(10),SG12(10),SG13(10),SG14(10),
     2SG15(10),SG16(10),SG17(10),SG18(10),SG19(10),SG20(10),SG21(10),
     3SG22(10),SG23(10),         SG24(10),SG25(10),SG26(10),SG27(10),
     4SG28(10),SG29(10),SG30(10),SG31(10),SG32(10),SG33(10),SG34(10),
     5SG35(10),SG36(10),SG37(10),SG38(10),SG39(10),SG61I(10),SG67I(10),
     5SG87I(10),SG81I(10)
      REAL MAXSET,MINSET,LBPDIS,LBP,MR,MRBAR,MRDBAR,
     *     IB,IS,IV,LOXRES,LOXAVL,LH2AVL,LH2RES,ISPNOM,MU1,MU2,MU3,
     1MU4,MU5,MU6,MU7,MU8,MU9,MU10,MU11,MU12,MU13,MU14,MU15,MU16,MU17,
     2MU18,MU19,MU20,MU21,MU22,MU23,MU24,MU25,MU26,MU27,MU28,MU29,MU30,
     3MU31,MU32,MU33,MU35,MU36,MU37,MU38,MU39,
     4NOTV,LXTKVL,LOXCO,LOXVNT,LH2VNT,LH2VOL,NLH2V,LH2DEN,NLH2D,LO2VOL,
     5NLO2V,NBOD,NFUDEN,NBFD,NFTV,LO2DEN
      INTEGER RES,PLOT,EXTRAP
      EXTRAP=0
      IRITE=0
      PLOT=0
      DO6I=1,200
    7 P(I)=0.0
      DO5J=1,10
    5 DUMSD(J,I)=0.0
    6 CONTINUE
   20 WRITE(6,21)
   21 FORMAT(1H1)
      CALL INPUT(5HTITLE,TITLE,5HNPARA,NPARA,5HNITER,NITER
     1,6HPRIMER,PRIMER,1HP,P,3HMU1,MU1,3HMU2,MU2,3HMU3,MU3,3HMU4,MU4,
     23HMU5,MU5,3HMU6,MU6,3HMU7,MU7,3HMU8,MU8,3HMU9,MU9,4HMU10,MU10,
     34HMU11,MU11,4HMU12,MU12,4HMU13,MU13,4HMU14,MU14,4HMU15,MU15,
     44HMU16,MU16,4HPLOT,PLOT,4HMU17,MU17,4HMU18,MU18,4HMU19,MU19,
     54HMU20,MU20, 6HEXTRAP,EXTRAP,4HMU21,MU21,4HMU22,MU22,4HMU23,MU23,
     64HMU24,MU24,4HMU25,MU25,4HMU26,MU26,4HMU27,MU27,4HMU28,MU28,
     74HMU29,MU29,4HMU30,MU30,4HMU31,MU31,4HMU32,MU32,4HMU33,MU33,
     84HMU34,MU34,4HMU35,MU35,4HMU36,MU36,4HMU37,MU37,4HMU38,MU38,
     94HMU39,MU39,4HNFTV,NFTV,  4HNOTV,NOTV,   5HPUSET,PUSET,
     *5HNLH2V,NLH2V, 5HNLH2D,NLH2D, 5HNLO2V,NLO2V, 5HNLO2D,NLO2D,
     *4HNBFD,NBFD,   5HIRITE,IRITE,3HP39,P39,4HNBOD,NBOD)
      CALL INPUT
     1(3HSG1,SG1,3HSG2,SG2,3HSG4,SG4,3HSG5,SG5,3HSG6,SG6,3HSG7,SG7,
     13HSG8,SG8,3HSG9,SG9,4HSG10,SG10,4HSG11,SG11,4HSG12,SG12,
     24HSG13,SG13,4HSG14,SG14,4HSG15,SG15,4HSG16,SG16,4HSG17,SG17,
```

```
     34HSG18,SG18,4HSG19,SG19,4HSG20,SG20,4HSG21,SG21,4HSG22,SG22,
     44HSG23,SG23,4HSG24,SG24,4HSG25,SG25,4HSG26,SG26,4HSG27,SG27,
     54HSG28,SG28,4HSG29,SG29,4HSG30,SG30,4HSG31,SG31,4HSG32,SG32,
     64HSG33,SG33,4HSG34,SG34,4HSG35,SG35,4HSG36,SG36,4HSG37,SG37,
     74HSG38,SG38,4HSG39,SG39,3HSG3,SG3,5HSG61I,SG61I,5HSG67I,SG67I,
     85HSG87I,SG87I,5HSG81I,SG81I,
     *4HGLH2,GLH2,6HSUBIAS,SUBIAS,6HMPURES,MPURES,6HVLVLAG,VLVLAG,
     *6HMAXSET,MAXSET,6HMINSET,MINSET,6HHBPDIS,HBPDIS,6HLBPDIS,LBPDIS,
     *3HLBP,LBP,4HTLO2,TLO2,4HGLO2,GLO2,3HHBP,HBP,
     *4HTLH2,TLH2)
      L=1
      IF(PLOT.EQ.0)GOTO24
      DO23I=1,600
   23 DPFREQ(I)=0.0
   24 DO30I=1,NPARA
      J=NGP(I)
      IF(J.EQ.0)GOTO030
      DO25K=1,J
      STDEV(L)=DUMSD(K,I)
   25 L=L+1
   30 CONTINUE
      RES=0
      DELP=0.0
      DELPSQ=0.0
      DO20000I=1,NITER
      L=1
      DAL61=0.0
      DAL81=0.0
      SUM81=0.0
      SUBSUM=0.0
      DO19000J=1,NPARA
      DIFF(J)=0.0
      LL=NCALC(J)
      GOTO(100,200,300,400,500,600,700,800,900,1000,1100,1200,1300,1400)
     1,LL
  100 CALL NORAD(PRIMER,ANS)
      IF(SQRT(ANS**2).GT.3.0)GOTO0100
      DIFF(J)=ANS*STDEV(L)
  110 GOTO18999
  200 NLOOP=NGP(J)
      IF(NLOOP.EQ.0)GOTO19000
      DO225N=1,NLOOP
      IF(N.EQ.7.AND.J.EQ.6)GOTO208
      IF(N.EQ.1.AND.J.EQ.6)GOTO212
      IF(N.EQ.1.AND.J.EQ.8)GOTO217
      IF(N.EQ.7.AND.J.EQ.8)GOTO230
      IF(N.EQ.4.AND.J.EQ.10)GOTO240
      IF(N.EQ.4.AND.J.EQ.11)GOTO250
  205 CALL NORAD(PRIMER,ANS)
      IF(SQRT(ANS**2).GT.3.0)GOTO0205
      DIFF(J)=DIFF(J)+ANS*STDEV(L)
  207 IF(N.EQ.6.AND.J.EQ.6)GOTO260
      IF(N.EQ.6.AND.J.EQ.8)GOTO265
      GOTO220
  212 DO215K=1,2
```

```
216 CALL NORAD(PRIMER,ANS)
    IF(SQRT(ANS**2).GT.3.0)GOTO216
215 SUBSUM=SUBSUM+ANS*SG61I(K)
    DAL61=SUBSUM*NFTV
    AFD=NBFD+SUBSUM
    GOTO220
208 DO210K=1,3
209 CALL NORAD(PRIMER,ANS)
    IF(SQRT(ANS**2).GT.3.0)GOTO209
210 DIFF(J)=DIFF(J)+ANS*SG67I(K)
    GOTO220
230 DO235K=1,4
232 CALL NORAD(PRIMER,ANS)
    IF(SQRT(ANS**2).GT.3.0)GOTO232
235 DIFF(J)=DIFF(J)+ANS*SG87I(K)
    GOTO220
217 DO219K=1,2
218 CALL NORAD(PRIMER,ANS)
    IF(SQRT(ANS**2).GT.3.0)GOTO218
219 SUM81=SUM81+ANS*SG81I(K)
    DAL81=SUM81*NOTV
    AOD=NBOD+SUM81
    GOTO220
240 CALL NORAD(PRIMER,ANS)
    IF(SQRT(ANS**2).GT.3.0)GOTO240
    DEND=ANS*STDEV(L)*NLH2D
    DIFF(J)=DIFF(J)+DEND*NLH2V
241 CALL NORAD(PRIMER,ANS)
    IF(SQRT(ANS**2).GT.3.0)GOTO241
    DIFF(J)=DIFF(J)+ANS*STDEV(L+1)*(NLH2D+DEND)
242 CALL NORAD(PRIMER,ANS)
    IF(SQRT(ANS**2).GT.3.0)GOTO242
    DIFF(J)=DIFF(J)-(ANS*STDEV(L+2)-1.865)*(NLH2D+DEND)
243 L=L+3
    GOTO19000
250 CALL NORAD(PRIMER,ANS)
    IF(SQRT(ANS**2).GT.3.0)GOTO250
    DENDO=ANS*STDEV(L)*NLO2D
    DIFF(J)=DIFF(J)+DENDO*NLO2V
251 CALL NORAD(PRIMER,ANS)
    IF(SQRT(ANS**2).GT.3.0)GOTO251
    DIFF(J)=DIFF(J)+(ANS*STDEV(L+1))*(NLO2D+DENDO)
252 CALL NORAD(PRIMER,ANS)
    IF(SQRT(ANS**2).GT.3.0)GOTO252
    DIFF(J)=DIFF(J)-(ANS*STDEV(L+2)-1.77)*(NLO2D+DENDO)
    GOTO243
260 DIFF(J)=DIFF(J)*AFD+DAL61
    GOTO220
265 DIFF(J)=DIFF(J)*AOD+DAL81
220 L=L+1
225 CONTINUE
    GOTO19000
300 IF(J.EQ.9)GOTO310
    DIFF(J)=SUBSUM
    GOTO18999
```

```
310  DIFF(J)=SUM81
     GOTO18999
400  DIFF(J)=0.0
     GOTO18999
500  CALL NORAD(PRIMER,ANS)
     IF(SQRT(ANS**2).GT.3.0)GOTO500
     DIFF(J)=ANS
     GOTO18999
600  NLOOP=NGP(J)
     DO625N=1,NLOOP
603  CALL NORAD(PRIMER,ANS)
     IF(SQRT(ANS**2).GT.3.0)GOTO603
     SVDIF(N)=ANS*STDEV(L)
605  IF(N.NE.8)GOTO620
     IF(MPURES.EQ.2)GOTO630
     IF(SVDIF(N).GE.-MU14(8))GOTO620
     RES=1
     SVDIF(N)=-PUSET*(SVDIF(N)+MU14(8))-MU14(8)
620  DIFF(J)=DIFF(J)+SVDIF(N)
624  L=L+1
625  CONTINUE
     GOTO19000
630  GOTO1301
631  CALL NORAD(PRIMER,R2)
     IF(ABS(R2).GT.3.0)GOTO631
     R2=R2*HBPDIS
     HBP=HBP+R2-TLH2+SVDIF(2)
635  CALL NORAD(PRIMER,R3)
     IF(ABS(R3).GT.3.0)GOTO635
     R3=R3*LBPDIS
     LBP=LBP+R3-TLO2+SVDIF(1)
     CPUB=MU14(8)
     CMS=-(CPUB-SUBIAS)
     IF(SVDIF(N).LT.CMS)GOTO680
     MRBAR=((MR*(W1DOT+W2DOT)*VLVLAG*(PUSET-MAXSET))/(2.0*LBP*(MR+1.0))
    1)+MAXSET
     SVDIF(N)=SVDIF(N)+CPUB-SUBIAS
     IF(SVDIF(N).GT.0.0)GOTO637
     SVDIF(N)=HBP-(LBP/MRBAR)
     IF(SVDIF(N).GE.0.0)GO TO 636
     SVDIF(N)=-SVDIF(N)*MRBAR
636  DIFF(J)=DIFF(J)+SVDIF(N)
     GOTO18999
637  SVDIF(N)=SVDIF(N)-CPUB+HBP-(LBP/MRBAR)
     IF(SVDIF(N).GE.0.0)GO TO 636
     SVDIF(N)=-SVDIF(N)*MRBAR
     GOTO636
680  CALL NORAD(PRIMER,R4)
     IF(ABS(R4).GT.3.0)GOTO680
     R4=R4*PUSET*STDEV(L)
     MRDBAR=(((W1DOT+W2DOT)*VLVLAG*(PUSET-MINSET))/((MR+1.0)*2.0*HBP))+
    1MINSET
     SVDIF(N)=R4-(CPUB-SUBIAS)*PUSET
     IF(SVDIF(N).GT.0.0)GOTO681
     SVDIF(N)=LBP-MRDBAR*HBP
```

```
      IF(SVDIF(N).GE.0.0) GO TO 636
      SVDIF(N)=-SVDIF(N)/MRDBAR
      GOTO636
  681 SVDIF(N)=SVDIF(N)+LBP-MRDBAR*HBP
      IF(SVDIF(N).GE.0.0)GO TO 636
      SVDIF(N)=-SVDIF(N)/MRDBAR
      GOTO636
  700 TB=0.0
  705 CALL NORAD(PRIMER,ANS)
      IF(SQRT(ANS**2).GT.3.0)GOTO0705
      DIFF(J)=TB+ANS*STDEV(L)
      GOTO18999
  800 IB=0.0
  805 CALL NORAD(PRIMER,ANS)
      IF(SQRT(ANS**2).GT.3.0)GOTO0805
      DIFF(J)=IB+ANS*STDEV(L)
      GOTO18999
  900 TS=0.0
  905 CALL NORAD(PRIMER,ANS)
      IF(SQRT(ANS**2).GT.3.0)GOTO0905
      DIFF(J)=TS+ANS*STDEV(L)
      GOTO18999
 1000 IS=0.0
 1005 CALL NORAD(PRIMER,ANS)
      IF(SQRT(ANS**2).GT.3.0)GOTO1005
      DIFF(J)=IS+ANS*STDEV(L)
      GOTO18999
 1100 TV=0.0
 1105 CALL NORAD(PRIMER,ANS)
      IF(SQRT(ANS**2).GT.3.0)GOTO1105
      DIFF(J)=TV+ANS*STDEV(L)
      GOTO18999
 1200 IV=0.0
 1205 CALL NORAD(PRIMER,ANS)
      IF(SQRT(ANS**2).GT.3.0)GOTO1205
      DIFF(J)=IV+ANS*STDEV(L)
      GOTO18999
 1300 IF(MPURES.EQ.2)GOTO1370
 1301 LOXRES=MU14(1)+SVDIF(1)+MU14(3)+SVDIF(3)
      LOXVNT=MU16(1)+DIFF(16)
      IF(RES.EQ.0)GOTO1310
      LOXRES=LOXRES+SVDIF(8)
 1310 LOXAVL=MU11(1)+DIFF(11)-LOXRES-LOXVNT
      LH2RES=MU14(2)+SVDIF(2)+MU14(4)+SVDIF(4)
      LH2VNT=MU15(1)+DIFF(15)
      IF(RES.EQ.1)GOTO1315
      LH2RES=LH2RES+SVDIF(8)
 1315 LH2AVL=MU10(1)+DIFF(10)-LH2RES-LH2VNT
      MR=LOXAVL/LH2AVL
C
      CALL TABL(MR,THNOM,TICEN(1),TICEN(4),1,1,1,3,IFBAD)
      GOTO(1318,1317),IFBAD
 1317 THNOM=0.0
 1318 CALL TABL(MR,ISPNOM,TICEN(1),TICEN(7),1,1,1,3,IFBAD)
      GOTO(1320,1319),IFBAD
```

```
 1319 ISPNOM=0.0
 1320 CALL NORAD(PRIMER,ANS)
      IF(SQRT(ANS**2).GT.3.0)GOTO1320
      DTE1=ANS*STDEV(L)
 1325 CALL NORAD(PRIMER,ANS)
      IF(SQRT(ANS**2).GT.3.0)GOTO1325
      DTE2=ANS*STDEV(L)
 1330 CALL NORAD(PRIMER,ANS)
      IF(SQRT(ANS**2).GT.3.0)GOTO1330
      DIE1=ANS*SG28(1)
 1335 CALL NORAD(PRIMER,ANS)
      IF(SQRT(ANS**2).GT.3.0)GOTO1335
      DIE2=ANS*SG28(1)
      W1DOT=(THNOM+DTE1)/(ISPNOM+DIE1)
      W2DOT=(THNOM+DTE2)/(ISPNOM+DIE2)
      IF(MPURES.EQ.2)GOTO0631
 1370 DIFF(J)=DTE1+DTE2+2.0*(THNOM-MU27(1))
      GOTO18999
 1400 DIFF(J)=((DIFF(27)+2.0*MU27(1))/(W1DOT+W2DOT))-MU28(1)
      GOTO18999
18999 L=L+1
19000 CONTINUE
      DIFF(13)=DIFF(13)+23.0
      IF(NITER.NE.100)GOTO19001
      WRITE(6,20024)(DIFF(N),N=1,NPARA)
19001 SUM=0.0
      DO 19050K=1,NPARA
      IF(K.EQ.39)GOTO19030
      IF(K.NE.10)GOTO19010
      DIFF(K)=DIFF(K)+DIFF(K+1)
19010 M=4*(K-1)+1
      IF(P(M).EQ.0.0)GOTO19050
      IF(DIFF(K).GT.0.0)GOTO19015
      PEE=P(M+2)
      GOTO19033
19015 PEE=P(M+3)
      GOTO19033
19030 CALL TABL(DIFF(39),PEE,P39(1),P39(4),1,1,1,3,IFBAD)
      GOTO(19033,19031),IFBAD
19031 PEE=0.0
19033 SUM=SUM+DIFF(K)*PEE
19049 IF(K.NE.10)GOTO19050
      K=K+1
19050 CONTINUE
      IF(NITER.NE.100)GOTO19075
      WRITE(6,20024)SUM
19075 DELP=DELP+SUM
      DELPSQ=DELPSQ+SUM**2
      IF(PLOT.EQ.0)GOTO20000
      INDEX=SUM+300.0
      IF(INDEX.LE.0)GOTO20000
      IF(INDEX.GT.600)GOTO20000
      DPFREQ(INDEX)=DPFREQ(INDEX)+1.0
20000 CONTINUE
      FNIT=NITER
```

```
      FM=DELP/FNIT
      SIGMA=SQRT((DELPSQ/FNIT)-FM**2)
      WRITE(6,20020)NPARA,NITER
20020 FORMAT(1H0,22X,23HNUMBER OF PARAMETERS = ,I4,22X,23HNUMBER OF ITER
     1ATIONS = ,I6)
20024 FORMAT(8(4X,1PE12.5))
      WRITE(6,20026)FM,SIGMA
20026 FORMAT(1H0,20X,14HMEAN OF FPR = ,1PE12.5,30X,21HSTANDARD DEVIATION
     1 = ,1PE12.5)
      IF(PLOT.EQ.0)GOTO20028
      CALL FQPLOT
20028 GOTO20
      END
```

```
$IBFTC BLOCK    FULIST,REF
       BLOCK DATA
       COMMON /BD/NGP,NCALC,P39,TICEN,TABLE
       DIMENSION NGP(300),NCALC(300),P39(6),TICEN(9),TABLE(500)
       DATA (NCALC(I),I=1,39)
      1      /1,2,1,1,1,2,3,2,3,2,2,2,2,6,1,1,1,1,7,8,1,9,10,1,11,12,13,
      214,4,4,4,4,4,4,5,2,2,1,5/
       DATA(NGP(I),I=1,39)/ 1, 2, 1, 1, 1, 8, 1, 8, 1, 6, 6, 4, 9, 8, 1,
      1 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 4, 4,
      2 1, 1/
       DATA (TICEN(I),I=1,9)/4.4,5.0,5.6,14735.0,15010.0,15320.0,436.3,
      1432.27,427.2/
       DATA(TABLE(I),I=1,60)
      */-3.090229,-2.7477777,-2.5758266,-2.4572651,-2.3656207,-2.2903708,
      *-2.2262151,-2.1700940,-2.1200716,-2.0748537,-2.0335239,-1.9953936,
      *-1.9599627,-1.9268407,-1.8956994,-1.8662989,-1.8384273,-1.8119147,
      *-1.7866170,-1.7624146,-1.7391994,-1.7168899,-1.6953981,-1.6746651,
      *-1.6546282,-1.6352339,-1.6164394,-1.5981967,-1.5804675,-1.5632271,
      *-1.5464361,-1.5300669,-1.5141038,-1.4985160,-1.4832834,-1.4683869,
      *-1.4538088,-1.4395321,-1.4255449,-1.4118333,-1.3983796,-1.3851714,
      *-1.3722072,-1.3594636,-1.3469419,-1.3346227,-1.3225085,-1.3105801,
      *-1.2988388,-1.2872739,-1.2758758,-1.2646415,-1.2535680,-1.2426446,
      *-1.2318667,-1.2212294,-1.2107284,-1.2003592,-1.1901178,-1.1799999/
       DATA(TABLE(I),I=61,120)
      */-1.170002,-1.1601197,-1.1503498,-1.1406886,-1.1311328,-1.1216791,
      *-1.1123243,-1.1030653,-1.0938992,-1.0848231,-1.0758388,-1.0669403,
      *-1.0581242,-1.0493880,-1.0407331,-1.0321567,-1.0236534,-1.0152221,
      *-1.0068668,-0.9985784,-0.9903568,-0.9822053,-0.9741153,-0.9660900,
      *-0.9581269,-0.9502211,-0.9423789,-0.9345899,-0.9268609,-0.9191840,
      *-0.9115629,-0.9039929,-0.8964754,-0.8890072,-0.8815894,-0.8742183,
      *-0.8668965,-0.8596179,-0.8523882,-0.8451987,-0.8380569,-0.8309547,
      *-0.8238952,-0.8168769,-0.8098959,-0.8029585,-0.7960566,-0.7891928,
      *-0.7823674,-0.7755758,-0.7688219,-0.7621027,-0.7554156,-0.7487647,
      *-0.7421463,-0.7355583,-0.7290041,-0.7224812,-0.7159873,-0.7095236/
       DATA(TABLE(I),I=121,180)
      */-.7030915,-0.6966868,-0.6903093,-0.6839620,-0.6776420,-0.6713478,
      *-0.6650790,-0.6588391,-0.6526240,-0.6464330,-0.6402654,-0.6341246,
      *-0.6280055,-0.6219124,-0.6158399,-0.6097922,-0.6037646,-0.5977609,
      *-0.5917767,-0.5858155,-0.5798731,-0.5739532,-0.5680512,-0.5621710,
      *-0.5563084,-0.5504664,-0.5446419,-0.5388366,-0.5330490,-0.5272791,
      *-0.5215273,-0.5157915,-0.5100742,-0.5043721,-0.4986875,-0.4930183,
      *-0.4873648,-0.4817275,-0.4761043,-0.4704976,-0.4649047,-0.4593264,
      *-0.4537629,-0.4482122,-0.4426767,-0.4371540,-0.4316443,-0.4261486,
      *-0.4206649,-0.4151942,-0.4097361,-0.4042893,-0.3988555,-0.3934331,
      *-0.3880215,-0.3826226,-0.3772341,-0.3718560,-0.3664898,-0.3611335/
       DATA(TABLE(I),I=181,240)
      */-.3557871,-0.3504519,-0.3451260,-0.3398095,-0.3345035,-0.3292064,
      *-0.3239181,-0.3186398,-0.3133699,-0.3081081,-0.3028559,-0.2976116,
      *-0.2923750,-0.2871470,-0.2819268,-0.2767139,-0.2715086,-0.2663111,
      *-0.2611203,-0.2559363,-0.2507599,-0.2455899,-0.2404262,-0.2352692,
      *-0.2301185,-0.2249736,-0.2198346,-0.2147019,-0.2095746,-0.2044525,
      *-0.1993361,-0.1942250,-0.1891187,-0.1840171,-0.1789209,-0.1738291,
      *-0.1687416,-0.1636585,-0.1585800,-0.1535053,-0.1484344,-0.1433676,
      *-0.1383044,-0.1332447,-0.1281883,-0.1231354,-0.1180856,-0.1130387,
      *-0.1079945,-0.1029535,-0.0979149,-0.0928787,-0.0878448,-0.0828134,
```

```
*-0.0777840,-0.0727564,-0.0677307,-0.0627069,-0.0576845,-0.0526636/
 DATA(TABLE(I),I=241,300)
*/-.0476440,-0.0426256,-0.0376083,-0.0325920,-0.0275764,-0.0225616,
*-0.0175473,-0.0125334,-0.0075199,-.0025066,.0025066,.0075199,
*.0125334,.0175473,.0225616, .0275764,.0325920,.0376083,
* 0.0426256, 0.0476440, 0.0526636, 0.0576845, 0.0627069, 0.0677307,
* 0.0727564, 0.0777840, 0.0828134, 0.0878448, 0.0928787, 0.0979149,
* 0.1029535, 0.1079945, 0.1130387, 0.1180856, 0.1231354, 0.1281883,
* 0.1332447, 0.1383044, 0.1433676, 0.1484344, 0.1535053, 0.1585800,
* 0.1636585, 0.1687416, 0.1738291, 0.1789209, 0.1840171, 0.1891187,
* 0.1942250, 0.1993361, 0.2044525, 0.2095746, 0.2147019, 0.2198346,
* 0.2249736, 0.2301185, 0.2352692, 0.2404262, 0.2455899, 0.2507599/
 DATA(TABLE(I),I=301,360)
*/0.2559363, 0.2611203, 0.2663111, 0.2715086, 0.2767139, 0.2819268,
* 0.2871470, 0.2923750, 0.2976116, 0.3028559, 0.3081081, 0.3133699,
* 0.3186398, 0.3239181, 0.3292064, 0.3345035, 0.3398095, 0.3451260,
* 0.3504519, 0.3557871, 0.3611335, 0.3664898, 0.3718560, 0.3772341,
* 0.3826226, 0.3880215, 0.3934331, 0.3988555, 0.4042893, 0.4097361,
* 0.4151942, 0.4206649, 0.4261486, 0.4316443, 0.4371540, 0.4426767,
* 0.4482122, 0.4537629, 0.4593264, 0.4649047, 0.4704976, 0.4761043,
* 0.4817275, 0.4873648, 0.4930183, 0.4986875, 0.5043721, 0.5100742,
* 0.5157915, 0.5215273, 0.5272791, 0.5330490, 0.5388366, 0.5446419,
* 0.5504664, 0.5563084, 0.5621710, 0.5680512, 0.5739532, 0.5798731/
 DATA(TABLE(I),I=361,420)
*/0.5858155, 0.5917767, 0.5977609, 0.6037646, 0.6097922, 0.6158399,
* 0.6219124, 0.6280055, 0.6341246, 0.6402654, 0.6464330, 0.6526240,
* 0.6588391, 0.6650790, 0.6713478, 0.6776420, 0.6839620, 0.6903093,
* 0.6966868, 0.7030915, 0.7095236, 0.7159873, 0.7224812, 0.7290041,
* 0.7355583, 0.7421463, 0.7487647, 0.7554156, 0.7621027, 0.7688219,
* 0.7755758, 0.7823674, 0.7891928, 0.7960566, 0.8029585, 0.8098959,
* 0.8168769, 0.8238952, 0.8309547, 0.8380569, 0.8451987, 0.8523882,
* 0.8596179, 0.8668965, 0.8742183, 0.8815894, 0.8890072, 0.8964754,
* 0.9039929, 0.9115629, 0.9191840, 0.9268609, 0.9345899, 0.9423789,
* 0.9502211, 0.9581269, 0.9660900, 0.9741153, 0.9822053, 0.9903568/
 DATA(TABLE(I),I=421,480)
*/0.9985784, 1.0068668, 1.0152221, 1.0236534, 1.0321567, 1.0407331,
* 1.0493880, 1.0581242, 1.0669403, 1.0758388, 1.0848231, 1.0938992,
* 1.1030653, 1.1123243, 1.1216791, 1.1311328, 1.1406886, 1.1503498,
* 1.1601197, 1.1700019, 1.1799999, 1.1901178, 1.2003592, 1.2107284,
* 1.2212294, 1.2318667, 1.2426446, 1.2535680, 1.2646415, 1.2758758,
* 1.2872739, 1.2988388, 1.3105801, 1.3225085, 1.3346227, 1.3469419,
* 1.3594636, 1.3722072, 1.3851714, 1.3983796, 1.4118333, 1.4255449,
* 1.4395321, 1.4538088, 1.4683869, 1.4832834, 1.4985160, 1.5141038,
* 1.5300669, 1.5464361, 1.5632271, 1.5804675, 1.5981967, 1.6164394,
* 1.6352339, 1.6546282, 1.6746651, 1.6953981, 1.7168899, 1.7391994/
 DATA(TABLE(I),I=481,500)
*/1.7624146, 1.7866170, 1.8119147, 1.8384273, 1.8662989, 1.8956994,
* 1.9268407, 1.9599627, 1.9953936, 2.0335239, 2.0748537, 2.1200716,
* 2.1700940, 2.2262151, 2.2903708, 2.3656207, 2.4572651, 2.5758266,
*2.74778       , 3.09023 /
 END
```

```
$IBFTC FQPLOT  FULIST,REF
      SUBROUTINE FQPLOT
      COMMON /BD/NGP,NCALC,P39,TICEN,TABLE
      DIMENSION NGP(300),NCALC(300),P39(6),TICEN(9),TABLE(500)
      COMMON DELP,DELPSQ,FM,SIGMA,FNIT ,NITER
      COMMON NM,AVMR,THNOM,ISPNOM,FUTKVL,LXTKVL,PUSET,DPFREQ(600)
      COMMON STDEV(300),MEAN(300),P(600),DIFF(300),SVDIF(10),TITLE(20)
      DIMENSION CUM95X(300),CUM95Y(300)
      CALL CAMRAV(35)
      YB=0.0
      YT=0.0
      DO50I=1,600
   40 IF(DPFREQ(I).LT.YT)GOTO50
      YT=DPFREQ(I)
   50 CONTINUE
      YT=YT+YT*.1
      XL=-300.
      XR=300.
      CALL DXDYV(1,XL,XR,DX,N,I,NX,15.0,IERR)
      CALL DXDYV(2,YB,YT,DY,M,J,NY,15.0,IERR)
      CALL GRIDIV(1,XL,XR,YB,YT,DX,DY,N,M,-I,-J,NX,NY)
      DO75K=1,600
      IF(DPFREQ(K).EQ.0.0)GOTO75
      FX=K-300
      CALL POINTV(FX,DPFREQ(K),-0)
   75 CONTINUE
      CALL PRINTV(-33,33HFLIGHT PERFORMANCE RESERVE ** LBS,380,20)
      CALL PRINTV(60,TITLE(1),272,0)
      CALL APRNTV(0,-14,-9,9HFREQUENCY,0,568)
      YB=0.0
      YT=1.0
      CALL DXDYV(2,YB,YT,DY,M,J,NY,15.0,IERR)
      CALL GRIDIV(1,XL,XR,YB,YT,DX,DY,N,M,-I,-J,NX,NY)
      KK=0
      DPSUM=0.0
      DO100K=1,600
      DPSUM=DPSUM+DPFREQ(K)
      IF(DPSUM.EQ.0.0)GOTO100
      FX=K-299
      SUMPLT=DPSUM/FNIT
      IF(SUMPLT.LT..95)GOTO86
      IF(SUMPLT.GE.1.0)GOTO86
   79 KK=KK+1
      CUM95X(KK)=FX
      CUM95Y(KK)=SUMPLT
   86 CALL POINTV(FX,SUMPLT,0)
  100 CONTINUE
      CALL PRINTV(-33,33HFLIGHT PERFORMANCE RESERVE ** LBS,380,20)
      CALL PRINTV(60,TITLE(11),272,0)
      CALL APRNTV(0,-14,-11,11HPROBABILITY,0,582)
      CALL DXDYV(1,CUM95X(1),CUM95X(KK),DX,N,I,NX,15.0,IERR)
      CALL DXDYV(2,.95,1.0,DY,M,J,NY,15.0,IERR)
      CALL GRIDIV(1,CUM95X(1),CUM95X(KK),.95,1.0,DX,DY,N,M,-I,-J,NX,NY)
      CALL APLOTV(KK,CUM95X    ,CUM95Y    ,1,1,1,42,IERR)
      CALL PRINTV(-33,33HFLIGHT PERFORMANCE RESERVE ** LBS,380,20)
```

```
CALL PRINTV(60,TITLE(11),272,0)
CALL APRNTV(0,-14,-11,11HPROBABILITY,0,582)
RETURN
END
```

```
$IBMAP RANDOM
        ENTRY   UNRAD
*       CALL    UNRAD (ARG1,ARG2)
*       ARG1    PRIMER TO BE USED
*       ARG2    A RANDOM NUMBER IN FLOATING POINT
*               THIS NO IS GREATER OR = ZERO AND LESS THAN ONE
 UNRAD  SAVE    4               SAVE INDEX 4
        LDQ*    3,4             LOAD WITH PREVIOUS RANDOM NO.
        MPY     =0343277244615
        STQ*    3,4             STORE RANDOM NUMBER
        CLA     =0200           FLOAT
        LRS     8               AND
        XCA
        FAD     =0              NORMALIZE NUMBER
        STO*    4,4             STORE ANS IN ARG2
        RETURN  UNRAD           RETURN TO CALLER
        ENTRY   NORAD
*       CALL    NORAD (ARG1,ARG2)
*       ARG1    PRIMER TO BE USED
*       ARG2    A RANDOM NO. OF VARIANCES
 NORAD  SAVE    1,4             SAVE INDEX 1 AND INDEX 4
        LDQ*    3,4             LOAD WITH PREVIOUS RANDOM NO.
        MPY     =0343277244615 5**15
        STQ*    3,4             STORE RANDOM NUMBER
        MPY     =500            CONVERT TO NUMBER
        PAC     ,1              SET INDEX 1 TO CORRECT ENTRY
        CLA     TABLE,1         PICK VALUE FROM TABLE
        STO*    4,4             STORE ANS IN ARG2
        RETURN  NORAD           RETURN TO CALLER
BD      CONTRL  NGP,ENDBD
NGP     BSS     300
NCALC   BSS     300
P39     BSS     6
TICEN   BSS     9
TABLE   BSS     500
ENDBD   NULL
        END
```