MONTE CARLO
FLIGHT PERFORMANCE RESERVE
PROGRAM

GD|C-BTD65-176
January 1966

GPO PRICE $ 
CFSTI PRICE(S) $

Hard copy (HC) 3.00
Microfiche (MF) 75

GENERAL DYNAMICS
Convair Division
MONTE CARLO
FLIGHT PERFORMANCE RESERVE
PROGRAM

GD\ C-BTD65-176
January 1966
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.
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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2 BASIC EQUATIONS</td>
<td>3</td>
</tr>
<tr>
<td>3 COMPUTER PROGRAM</td>
<td>7</td>
</tr>
<tr>
<td>3.1 Input Variables</td>
<td>25</td>
</tr>
<tr>
<td>3.2 Output</td>
<td>28</td>
</tr>
<tr>
<td>3.3 Variables Used Within the Program</td>
<td>28</td>
</tr>
<tr>
<td>3.4 Sample Case</td>
<td>32</td>
</tr>
<tr>
<td>3.5 Input Card and Deck Format</td>
<td>41</td>
</tr>
<tr>
<td>Appendix PROGRAM LISTING</td>
<td>45</td>
</tr>
</tbody>
</table>
### LIST OF ILLUSTRATIONS

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

### LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Parameter- Contribution Equations and Method-of-Computation Numbers</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>Parameter- Controlled Input</td>
<td>32</td>
</tr>
</tbody>
</table>
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

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

(1)

where the form of the function $P$ is arbitrary and the $\alpha$'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
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.
The basic equations for the FPR analysis are derived from a performance function $P$ which is configuration- and mission-independent.

Let

$$P = P(\alpha_1, \alpha_2, \cdots, \alpha_n)$$

represent some vehicle's performance as a function of the $n$ variables $\alpha_i$. 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

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

which holds whether or not the $\alpha_i$'s are independent.

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

$$\Phi(\alpha_1, \alpha_2, \cdots, \alpha_k) = 0$$

correlating the variables considered.

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

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

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

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

and

$$
\sum_{i=1}^{l} \sum_{i=1}^{n} \left( \frac{\partial f}{\partial \alpha_i} d\alpha_i \right)^2
$$

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 $\alpha$ variations are given directly by

$$
m = \frac{1}{l} \sum \Delta P
$$

and

$$
s = \sqrt{\frac{1}{l} \sum \Delta P^2 - m^2}
$$

The associated standard errors are

$$
\sigma^2 (m) = s^2 \frac{1}{l}
$$
\[
\sigma^2(s) = \frac{s^2}{(2\ell)^{-1}}
\]

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 \( \lambda, \nu \). For a Gaussian distribution, \( \lambda \) and \( \nu \) are the mean and standard deviation respectively.

Also, let \( \sigma_i \) be the standard deviation of the \( i \)th parameter's variation. Similarly, \( \sigma_{ij} \) 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)
FQPOLT (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 user's 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

<table>
<thead>
<tr>
<th>( d\alpha )</th>
<th>PARAMETER</th>
<th>EQUATION</th>
<th>METHOD NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d\alpha_1 )</td>
<td>Booster</td>
<td>( d\alpha_1 = R(0, \sigma_1) )</td>
<td>1</td>
</tr>
<tr>
<td>( d\alpha_2 )</td>
<td>Sustainer and Inter-stage Adapter</td>
<td>( d\alpha_{21} = R(0, \sigma_{21}) )</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( d\alpha_{22} = R(0, \sigma_{22}) )</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( d\alpha_2 = R(0, \sigma_{21}) + R(0, \sigma_{22}) )</td>
<td></td>
</tr>
<tr>
<td>( d\alpha_3 )</td>
<td>Centaur</td>
<td>( d\alpha_3 = R(0, \sigma_3) )</td>
<td>1</td>
</tr>
<tr>
<td>( d\alpha_4 )</td>
<td>Nose Fairing</td>
<td>( d\alpha_4 = R(0, \sigma_4) )</td>
<td>1</td>
</tr>
<tr>
<td>( d\alpha_5 )</td>
<td>Insulation Panels</td>
<td>( d\alpha_5 = R(0, \sigma_5) )</td>
<td>1</td>
</tr>
<tr>
<td>( d\alpha_6 )</td>
<td>Booster Flight Expendables</td>
<td>( \text{SUBSUM} = \sum_{i=1}^{2} R(0, \sigma_6_{ii}) )</td>
<td>1</td>
</tr>
<tr>
<td>( )</td>
<td></td>
<td>( d\alpha_{61} = \text{SUBSUM} \cdot \text{NFTV} )</td>
<td></td>
</tr>
</tbody>
</table>
Table 1. Parameter-Contributions Equations and Method-of-Computation Numbers (Contd)

<table>
<thead>
<tr>
<th>$d\alpha$</th>
<th>PARAMETER</th>
<th>EQUATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probe Location</td>
<td>$d\alpha_{62} = R(0, \sigma_{62})$</td>
<td>AFD = NBFD + SUBSUM</td>
</tr>
<tr>
<td>Surface Level Variation</td>
<td>$d\alpha_{63} = R(0, \sigma_{63})$</td>
<td></td>
</tr>
<tr>
<td>Tank Pressure</td>
<td>$d\alpha_{64} = R(0, \alpha_{64})$</td>
<td></td>
</tr>
<tr>
<td>Tank Volume</td>
<td>$d\alpha_{65} = R(0, \sigma_{65})$</td>
<td></td>
</tr>
<tr>
<td>Tanking Level</td>
<td>$d\alpha_{66} = \left[ \sum_{i=2}^{5} d\alpha_{6i} + R(0, \sigma_{66}) \right] \cdot AFD + d\alpha_{61}$</td>
<td></td>
</tr>
<tr>
<td>Ground Expended</td>
<td>$d\alpha_{67} = \sum_{i=1}^{3} R(0, \sigma_{67i})$</td>
<td></td>
</tr>
<tr>
<td>Sustainer Thrust Decay</td>
<td>$d\alpha_{68} = R(0, \sigma_{68})$</td>
<td></td>
</tr>
<tr>
<td>Then,</td>
<td>$d\alpha_{6} = d\alpha_{66} + d\alpha_{67} + d\alpha_{68}$</td>
<td></td>
</tr>
<tr>
<td>Fuel Density</td>
<td>$d\alpha_{7} = \text{SUBSUM}$</td>
<td>3</td>
</tr>
<tr>
<td>Oxidizer Weight</td>
<td>Oxidizer Density</td>
<td>SUM81 = $\sum_{i=1}^{2} R(0, \sigma_{81i})$</td>
</tr>
<tr>
<td></td>
<td>$\sigma_{81}$</td>
<td>$d\alpha_{81} = \text{SUM81} \cdot \text{NOTV}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AOD = NLO2D + SUM81</td>
</tr>
<tr>
<td>Sensor Location</td>
<td>$d\alpha_{82} = R(0, \sigma_{82})$</td>
<td></td>
</tr>
</tbody>
</table>
Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers (Contd)

<table>
<thead>
<tr>
<th>$d\alpha$</th>
<th>PARAMETERS</th>
<th>EQUATIONS</th>
<th>METHOD NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Surface Level</td>
<td>$d\alpha_3 = R(0, \sigma_3)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Variation</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tank Pressure</td>
<td>$d\alpha_4 = R(0, \sigma_4)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tank Volume</td>
<td>$d\alpha_5 = R(0, \sigma_5)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tanking Level</td>
<td>$d\alpha_6 = \left[ \sum_{i=2}^{5} d\alpha_i + R(0, \sigma_6) \right] \cdot \text{AOD}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$+ d\alpha_{81}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ground Expended</td>
<td>$d\alpha_7 = \sum_{i=1}^{4} R(0, \sigma_{7i})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Thrust Decay</td>
<td>$d\alpha_8 = R(0, \sigma_8)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\text{Then,}$ $d\alpha_8 = d\alpha_{6} + d\alpha_{7} + d\alpha_{8}$</td>
<td></td>
</tr>
<tr>
<td>$d\alpha_9$</td>
<td>Oxidizer Density</td>
<td>$d\alpha_9 = \text{SUM81}$</td>
<td>3</td>
</tr>
<tr>
<td>$d\alpha_{10}$</td>
<td>Fuel Weight</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Sensor Sensitivity</td>
<td>$d\alpha_{101} = R(0, \sigma_{101})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sensor Location</td>
<td>$d\alpha_{102} = R(0, \sigma_{102})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Surface Variations</td>
<td>$d\alpha_{103} = R(0, \sigma_{103})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Density</td>
<td>$\text{DEND} = R(0, \sigma_{104}) \cdot \text{NLH2D}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d\alpha_{104} = \text{DEND} \cdot \text{NLH2V}$</td>
<td></td>
</tr>
</tbody>
</table>
Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers (Contd)

<table>
<thead>
<tr>
<th>$d\alpha$</th>
<th>PARAMETERS</th>
<th>EQUATIONS</th>
<th>METHOD NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d\alpha_{10}$</td>
<td>Tank Volume</td>
<td>$d\alpha_{10} = R(0, \sigma_{10}) (NLH2D + DEND)$</td>
<td></td>
</tr>
<tr>
<td>$d\alpha_{10}$</td>
<td>Tank Ullage</td>
<td>$d\alpha_{10} = -\left[R(MFUV^{\dagger}, \sigma_{10}) - 1.865\right] \cdot (NLH2D + DEND)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Then, $d\alpha_{10} = \sum_{i=1}^{6} d\alpha_{10i}$</td>
<td>2</td>
</tr>
<tr>
<td>$d\alpha_{11}$</td>
<td>Oxidizer Weight</td>
<td>$d\alpha_{11} = R(0, \sigma_{11})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sensor Sensitivity</td>
<td>$d\alpha_{11} = R(0, \sigma_{11})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sensor Location</td>
<td>$d\alpha_{11} = R(0, \sigma_{11})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Surface Variations</td>
<td>$d\alpha_{11} = R(0, \sigma_{11})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Density</td>
<td>DENDO = $R(0, \sigma_{11}) \cdot NLO2D$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$d\alpha_{11} = DENDO \cdot NLO2V$</td>
<td></td>
</tr>
<tr>
<td>$d\alpha_{11}$</td>
<td>Tank Volume</td>
<td>$d\alpha_{11} = R(0, \sigma_{11}) \cdot (NLO2D + DENDO)$</td>
<td></td>
</tr>
<tr>
<td>$d\alpha_{11}$</td>
<td>Tank Ullage</td>
<td>$d\alpha_{11} = -\left[R(0, \sigma_{11}) - 1.77\right] \cdot (NLO2D + DENDO)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Then, $d\alpha_{11} = \sum_{i=1}^{6} d\alpha_{11i}$</td>
<td>2</td>
</tr>
<tr>
<td>$d\alpha_{12}$</td>
<td>Booster Jettisoned Residuals</td>
<td>$d\alpha_{12} = R(0, \sigma_{12})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Trapped Fuel</td>
<td>$d\alpha_{12} = R(0, \sigma_{12})$</td>
<td></td>
</tr>
</tbody>
</table>

$^{\dagger}$ Mean Fuel Ullage Volume
Table 1. Parameter- Contribution Equations and Method-of-Computation Numbers (Contd)

<table>
<thead>
<tr>
<th>$d\alpha$</th>
<th>PARAMETERS</th>
<th>EQUATIONS</th>
<th>METHOD NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trapped Oxidizer</td>
<td>$d\alpha_{12} = R(0, \sigma_{12})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lube Oil</td>
<td>$d\alpha_{13} = R(0, \sigma_{13})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Helium</td>
<td>$d\alpha_{14} = R(0, \sigma_{14})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Trapped Fuel</td>
<td>$d\alpha_{15} = R(0, \sigma_{15})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sustainer Jettisoned Residuals</td>
<td>$d\alpha_{16} = R(0, \sigma_{16})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GO$_2$ in Tank (Flight)</td>
<td>$d\alpha_{17} = R(MGO2TG^t)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GO$_2$ in Tank (Ground)</td>
<td>$d\alpha_{18} = R(SPUB^{*}, \sigma_{18})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PU Bias</td>
<td>$d\alpha_{19} = R(SPUB^{*}, \sigma_{18})$</td>
<td></td>
</tr>
</tbody>
</table>

Then,

$$d\alpha = \sum_{i=1}^{n} d\alpha_i$$

† Mean GO$_2$ in Tank - Ground

‡ Sustainer PU Bias
Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers (Contd)

<table>
<thead>
<tr>
<th>$d\alpha$</th>
<th>PARAMETERS</th>
<th>EQUATIONS</th>
<th>METHOD NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d\alpha_{14}$</td>
<td>Centaur Jettisoned Residuals</td>
<td>$d\alpha_{14} = R (0, \sigma_{14})$</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Trapped LO$_2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Trapped LH$_2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GO$_2$ in Tank</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GH$_2$ in Tank</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>H$_2$O$_2$ Weight</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Helium</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ice and Frost</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>PU</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Without end effect (MPURES = 1)</td>
<td>$d\alpha_{14} = R (0, \sigma_{14})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$d\alpha_{14}$ &lt; -MU 14(8) ?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>RES = 1</td>
<td>$d\alpha_{14} = -\text{PUSET} \cdot (d\alpha_{14} + \text{MU 14(8)}) - \text{MU 14(8)}$</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>RES = 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>With end effect (MPURES = 2)</td>
<td>$d\alpha_{14} = R (0, \sigma_{14})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R2 = R (0, R2) * HBPDIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HBP = HBP + R2 - TLH2 + $d\alpha_{14}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>R3 = R (0, R3) * LBPDIS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 1. Parameter-Contribuition Equations and Method-of-Computation Numbers (Contd)

<table>
<thead>
<tr>
<th>dα</th>
<th>PARAMETERS</th>
<th>EQUATIONS</th>
<th>METHOD NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>LBP = LBP + R3 - TLO2 + dα₁₄₈</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CPUB = MU₁₄(8)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>dα₁₄₈ &lt; -(CPUB - SUBIAS) ?</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td></td>
<td>R4 = R (0, R4) • PUSET • σ₁₄₈</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MR = ( \frac{(\omega_1 + \omega_2) \cdot VLVLAG \cdot (PUSET-MINSET)}{(MR + 1.0) \cdot 2.0 \cdot HBP} + MINSET )</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>dα₁₄₈ = R4 - (CPUB-SUBIAS) • PUSET</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>dα₁₄₈ &gt; 0.0 ?</td>
<td></td>
</tr>
<tr>
<td>Yes</td>
<td></td>
<td>dα₁₄₈ = dα₁₄₈ + LBP - ( \frac{MR}{MR} ) HBP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>dα₁₄₈ ≥ 0.0 ?</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td></td>
<td>dα₁₄₈ = -dα₁₄₈ /MR</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td></td>
<td>dα₁₄₈ = LPB - ( \frac{MR}{MR} ) HBP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>dα₁₄₈ &gt; 0.0 ?</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td></td>
<td>dα₁₄₈ = -dα₁₄₈ /MR</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td></td>
<td>MR = ( \frac{(\omega_1 + \omega_2) \cdot VLVLAG \cdot (PUSET-MAXSET)}{2.0 \cdot LBP \cdot (MR + 1.0)} + MAXSET )</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>dα₁₄₈ = dα₁₄₈ + CPUB - SUBIAS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>dα₁₄₈ &gt; 0.0 ?</td>
<td></td>
</tr>
</tbody>
</table>
Table 1. Parameter–Contribution Equations and Method–of–Computation Numbers (Contd)

<table>
<thead>
<tr>
<th>$d\alpha$</th>
<th>PARAMETER</th>
<th>EQUATIONS</th>
<th>METHOD NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>No</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>No</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>No</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$d\alpha_{15}$</td>
<td>Ground and Inflight $\text{LH}_2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$d\alpha_{16}$</td>
<td>Ground and Inflight $\text{LO}_2$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$d\alpha_{17}$</td>
<td>Coast</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$d\alpha_{18}$</td>
<td>Booster Mixture Ratio</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$d\alpha_{19}$</td>
<td>Booster Thrust</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$d\alpha_{20}$</td>
<td>Booster ISP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$d\alpha_{21}$</td>
<td>Sustainer Mixture Ratio</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CENTAUR VENTING

- $d\alpha_{15} = R(0, \sigma_{15})$ 1
- $d\alpha_{16} = R(0, \sigma_{16})$ 1
- $d\alpha_{17} = R(0, \sigma_{17})$ 1

BOOSTER PROPULSION

- $d\alpha_{18} = R(0, \sigma_{18})$ 1
- $d\alpha_{19} = \text{TB} + R(0, \sigma_{19})$ 7
- $d\alpha_{20} = \text{IB} + R(0, \sigma_{20})$ 8
- $d\alpha_{21} = R(0, \sigma_{21})$ 1

15
Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers (Contd)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Equations</th>
<th>Method No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_{22}$ Sustainer Thrust</td>
<td>$\alpha_{22} = TS + R (0, \sigma_{22})$</td>
<td>9</td>
</tr>
<tr>
<td>$\alpha_{23}$ Sustainer ISP</td>
<td>$\alpha_{23} = IS + R (0, \sigma_{23})$</td>
<td>10</td>
</tr>
<tr>
<td>$\alpha_{24}$ Vernier Mixture Ratio</td>
<td>$\alpha_{24} = R (0, \sigma_{24})$</td>
<td>1</td>
</tr>
<tr>
<td>$\alpha_{25}$ Vernier Thrust</td>
<td>$\alpha_{25} = TV + R (0, \sigma_{25})$</td>
<td>11</td>
</tr>
<tr>
<td>$\alpha_{26}$ Vernier ISP</td>
<td>$\alpha_{26} = IV + R (0, \sigma_{26})$</td>
<td>12</td>
</tr>
</tbody>
</table>

**CENTAUR PROPULSION**

- LOXRES = MU14 (1) + $\alpha_{14}$ + MU14 (3) + $\alpha_{14}$
- LOXVNT = MU16 (1) + $\alpha_{16}$
- RES = 0 ?
- No | LOXRES = LOXRES + $\alpha_{14}$
- LOXAVL = MU11 (1) + $\alpha_{11}$ - LOXRES - LOXVNT
- LH2RES = MU14 (2) + $\alpha_{14}$ + MU14 (4) + $\alpha_{14}$
- LH2VNT = MU15 (1) + $\alpha_{15}$
- RES = 1 ?
- No | LH2RES = LH2RES + $\alpha_{14}$
- LH2AVL = MU10 (1) + $\alpha_{10}$ - LH2RES - LH2VNT
- MR = LOXAVL/LH2AVL
Table 1. Parameter-Contribution Equations and Method-of-Computation Numbers (Contd)

<table>
<thead>
<tr>
<th>$d\alpha$</th>
<th>PARAMETER</th>
<th>EQUATIONS</th>
<th>METHOD NO.</th>
</tr>
</thead>
</table>
| $d\alpha_{28}$ | Specific Impulse (ISP) | $DTE_1 = R (0, \sigma_{DTE_1})$<br>$DTE_2 = R (0, \sigma_{DTE_2})$<br>$DIE_1 = R (0, \sigma_{DIE_1})$<br>$DIE_2 = R (0, \sigma_{DIE_2})$
$\omega_1 = (THNOM+DTE_1)/(ISP_{NOM}+DIE_1)$<br>$\omega_2 = (THNOM+DTE_2)/(ISP_{NOM}+DIE_2)$
$da_{27} = DTE_1 + DTE_2 + (2.0 \times (THNOM - MU_{27}(1)))$
$da_{28} = \left[da_{27} + 2 \times MU_{27}(1)\right]/(\omega_1 + \omega_2)$ | 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) | $d\alpha_{33} = R (0, 1)$ | 5 |
| $d\alpha_{34}$ | Attitude Control (Coast 2) | | 2 |
| $d\alpha_{35}$ | Atmosphere | $d\alpha_{361} = R (0, \sigma_{361})$
$\text{Null Voltage}$ | |
| $d\alpha_{36}$ | Launch Azimuth | $d\alpha_{362} = R (0, \sigma_{362})$
Roll Gyro Torquing Rate | |
| $d\alpha_{37}$ | Time Uncertainties | $d\alpha_{363} = R (0, \sigma_{363})$ | |
Table 1. Parameter- Contribution Equations and Method-of-Computation Numbers (Contd)

<table>
<thead>
<tr>
<th>$d\alpha$</th>
<th>PARAMETER</th>
<th>EQUATIONS</th>
<th>METHOD NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d\alpha_{37}$</td>
<td>Pitch Program</td>
<td>$d\alpha_{36}^4 = R(0, \sigma_{36}^4)$</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Then, $d\alpha_{36} = \sum_{i=1}^{4} d\alpha_{36}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Voltage-Time Integral</td>
<td>$d\alpha_{37}^1 = R(0, \sigma_{37}^1)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gryo Torquing Rate-Voltage-Time Average</td>
<td>$d\alpha_{37}^2 = R(0, \sigma_{37}^2)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Inverter Voltage</td>
<td>$d\alpha_{37}^3 = R(0, \sigma_{37}^3)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Inverter Frequency</td>
<td>$d\alpha_{37}^4 = R(0, \sigma_{37}^4)$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Drag Force</td>
<td>$d\alpha_{38} = R(0, \sigma_{38})$</td>
<td>1</td>
</tr>
<tr>
<td>$d\alpha_{39}$</td>
<td>Wind Profile</td>
<td>$d\alpha_{39} = R(0, 1)$</td>
<td>5</td>
</tr>
</tbody>
</table>
Figure 1. Generalized Computer Flow Chart
Figure 2. Detailed Flow Chart (Sheet 1 of 4)
Figure 2. Detailed Flow Chart (Sheet 2 of 4)
FOR LL = 7
700
SET TB
CA
N0
(PRIME)

FOR LL = 8
800
SET IB
CA
N0
(PRIME)

FOR LL = 9
900
SET TS
CA
N1
(PRIME)

FOR LL = 10
1000
SET IS
CA
N1
(PRIME)

FOR LL = 11
1100
SET TV
CA
N1
(PRIME)

FOR LL = 12
1200
SET IV
CA
N1
(PRIME)

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 compatible with the nomenclature used in Report GD/C-BTD65-143 where possible.)

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

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

**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 LO₂ volume.

**NLO2D** - Nominal LO₂ density.

**NLH2V** - Nominal LH₂ volume.
NLH2D - Nominal LH2 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 dα61i.

SG67I = Standard deviation for dα67i.

SG81I = Standard deviation for dα81i.

SG87I = Standard deviation for dα87i.

SUBIAS - Amount of PU bias below LH2 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 - LO2 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 - DLP².

DEND - Fuel density dispersion (Centaur).

DENDO - LO₂ density dispersion (Centaur).

DIE1 - Dispersion and ISP for Engine No. 1.

DIE2 - Dispersion and ISP for Engine No. 2.

DIFF(I) - dα₄, NPARA values, NITER sets.

DPFREQ(N) - Frequency of occurrence 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 $d\alpha_{148} < \mu_{148}$.

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

SIGMA - Final standard deviation (output).

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

$$\text{SUBSUM} = \sum_{i=1}^{2} d\alpha_{81}$$

$$\text{SUM81} = \sum_{i=1}^{2} d\alpha_{81}$$

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 LH₂ 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 LO₂ probe uncovery.

MRDBAR - Mixture ratio associated with LH₂ 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 LH₂ and a PU bias of 9 pounds of LH₂.

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

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>PROGRAM NAME (SG = Standard Deviation)</th>
<th>INPUT QUANTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(MU = Mean)</td>
<td></td>
</tr>
<tr>
<td>Booster</td>
<td>SG1</td>
<td>10.0</td>
</tr>
<tr>
<td>Sustainer and Interstage</td>
<td>SG2(1)</td>
<td>16.6666</td>
</tr>
<tr>
<td>Adapter</td>
<td>SG2(1)</td>
<td>16.3333</td>
</tr>
<tr>
<td>Centaur</td>
<td>SG3</td>
<td>20.0</td>
</tr>
<tr>
<td>Nose Fairing</td>
<td>SG4</td>
<td>25.0</td>
</tr>
<tr>
<td>Insulation Panels</td>
<td>SG5</td>
<td>16.0</td>
</tr>
<tr>
<td>Fuel Weight (Booster)</td>
<td>SG61I(1)</td>
<td>0.15</td>
</tr>
<tr>
<td>Fuel Density</td>
<td>SG61I(2)</td>
<td>0.104</td>
</tr>
</tbody>
</table>
Table 2. Parameter-Controlled Input (Contd)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>PROGRAM NAME</th>
<th>INPUT QUANTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probe Location</td>
<td>SG6(2)</td>
<td>0.4</td>
</tr>
<tr>
<td>Surface Level Variation</td>
<td>SG6(3)</td>
<td>0.6</td>
</tr>
<tr>
<td>Tank Pressure</td>
<td>SG6(4)</td>
<td>0.086666</td>
</tr>
<tr>
<td>Tank Volume</td>
<td>SG6(5)</td>
<td>0.76666</td>
</tr>
<tr>
<td>Tanking Level</td>
<td>SG6(6)</td>
<td>0.66666</td>
</tr>
<tr>
<td>Ground Expended</td>
<td>SG67I(1)</td>
<td>16.6666</td>
</tr>
<tr>
<td></td>
<td>SG67I(2)</td>
<td>28.6666</td>
</tr>
<tr>
<td></td>
<td>SG67I(3)</td>
<td>6.6666</td>
</tr>
<tr>
<td>Sustainer Thrust Decay</td>
<td>SG6(8)</td>
<td>2.0</td>
</tr>
<tr>
<td>Fuel Density</td>
<td>SG7</td>
<td>0.017</td>
</tr>
<tr>
<td>Oxidizer Weight (Booster)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oxidizer Density</td>
<td>SG8I(1)</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>SG8I(2)</td>
<td>0.13666</td>
</tr>
<tr>
<td>Sensor Location</td>
<td>SG8(2)</td>
<td>0.5333</td>
</tr>
<tr>
<td>Surface Level Variation</td>
<td>SG8(3)</td>
<td>0.58333</td>
</tr>
<tr>
<td>Tank Pressure</td>
<td>SG8(4)</td>
<td>0.4</td>
</tr>
<tr>
<td>Tank Volume</td>
<td>SG8(5)</td>
<td>2.0833</td>
</tr>
<tr>
<td>Tanking Level</td>
<td>SG8(6)</td>
<td>1.4666</td>
</tr>
<tr>
<td>Ground Expended</td>
<td>SG87I(1)</td>
<td>66.6666</td>
</tr>
<tr>
<td></td>
<td>SG87I(2)</td>
<td>50.0</td>
</tr>
<tr>
<td></td>
<td>SG87I(3)</td>
<td>65.3333</td>
</tr>
<tr>
<td></td>
<td>SG87I(4)</td>
<td>10.0</td>
</tr>
<tr>
<td>Thrust Decay</td>
<td>SG8(8)</td>
<td>3.0</td>
</tr>
<tr>
<td>Oxidizer Density</td>
<td>SG9</td>
<td>0.14</td>
</tr>
<tr>
<td>Fuel Weight (Centaur)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensor Sensitivity</td>
<td>SG10(1)</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>MU10</td>
<td>5056.0</td>
</tr>
<tr>
<td>Sensor Location</td>
<td>SG10(2)</td>
<td>1.0</td>
</tr>
</tbody>
</table>
Table 2. Parameter-Controlled Input (Contd)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>PROGRAM NAME</th>
<th>INPUT QUANTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surface Variation</td>
<td>SG10(3)</td>
<td>1.0</td>
</tr>
<tr>
<td>Density</td>
<td>SG10(4)</td>
<td>0.01</td>
</tr>
<tr>
<td>Tank Volume</td>
<td>SG10(5)</td>
<td>4.2167</td>
</tr>
<tr>
<td>Tank Ullage</td>
<td>SG10(6)</td>
<td>1.095</td>
</tr>
<tr>
<td><strong>Oxidizer Weight (Centaur)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensor Sensitivity</td>
<td>SG11(1)</td>
<td>11.0</td>
</tr>
<tr>
<td>Sensor Location</td>
<td>SG11(2)</td>
<td>11.0</td>
</tr>
<tr>
<td>Surface Variations</td>
<td>SG11(3)</td>
<td>5.0</td>
</tr>
<tr>
<td>Density</td>
<td>SG11(4)</td>
<td>0.003333</td>
</tr>
<tr>
<td>Tank Volume</td>
<td>SG11(5)</td>
<td>1.25333</td>
</tr>
<tr>
<td>Tank Ullage</td>
<td>SG11(6)</td>
<td>0.81666</td>
</tr>
<tr>
<td><strong>Booster Jettisoned Residuals</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trapped Fuel</td>
<td>SG12(1)</td>
<td>17.0</td>
</tr>
<tr>
<td>Trapped Oxidizer</td>
<td>SG12(2)</td>
<td>23.0</td>
</tr>
<tr>
<td>Lube Oil</td>
<td>SG12(3)</td>
<td>6.0</td>
</tr>
<tr>
<td>Helium</td>
<td>SG12(4)</td>
<td>4.0</td>
</tr>
<tr>
<td><strong>Sustainer Jettisoned Residuals</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trapped Fuel</td>
<td>SG13(1)</td>
<td>20.0</td>
</tr>
<tr>
<td>Trapped Oxidizer</td>
<td>SG13(2)</td>
<td>52.0</td>
</tr>
<tr>
<td>Lube Oil</td>
<td>SG13(3)</td>
<td>2.0</td>
</tr>
<tr>
<td>Helium</td>
<td>SG13(4)</td>
<td>2.0</td>
</tr>
<tr>
<td>Nitrogen</td>
<td>SG13(5)</td>
<td>1.0</td>
</tr>
<tr>
<td>$\text{GO}_2$ in Tank (Flight)</td>
<td>SG13(6)</td>
<td>2.0</td>
</tr>
<tr>
<td>$\text{GO}_2$ in Tank (Ground)</td>
<td>SG13(7)</td>
<td>44.0</td>
</tr>
<tr>
<td>PU Bias</td>
<td>SG13(8)</td>
<td>41.6666</td>
</tr>
</tbody>
</table>
Table 2. Parameter-Controlled Input (Contd)

<table>
<thead>
<tr>
<th>PROGRAM NAME</th>
<th>PARAMETER</th>
<th>INPUT QUANTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centaur Jettisoned Residuals</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trapped LO₂</td>
<td>SG14(1)</td>
<td>1.0</td>
</tr>
<tr>
<td>Trapped LH₂</td>
<td>SG14(2)</td>
<td>1.0</td>
</tr>
<tr>
<td>GO₂ in Tank</td>
<td>SG14(3)</td>
<td>12.0</td>
</tr>
<tr>
<td>GH₂ in Tank</td>
<td>SG14(4)</td>
<td>5.0</td>
</tr>
<tr>
<td>H₂O₂ Weight</td>
<td>SG14(5)</td>
<td>5.0</td>
</tr>
<tr>
<td>Helium</td>
<td>SG14(6)</td>
<td>0.0</td>
</tr>
<tr>
<td>Ice and Frost</td>
<td>SG14(7)</td>
<td>4.0</td>
</tr>
<tr>
<td>PU</td>
<td>SG14(8)</td>
<td>8.3333</td>
</tr>
<tr>
<td></td>
<td>MU14(8)</td>
<td>9.0</td>
</tr>
<tr>
<td>Centaur Venting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ground and Inflight LH₂</td>
<td>SG15</td>
<td>4.8</td>
</tr>
<tr>
<td>Ground and Inflight LO₂</td>
<td>SG16</td>
<td>9.0</td>
</tr>
<tr>
<td>Booster Propulsion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Booster Mixture Ratio</td>
<td>SG18</td>
<td>0.008</td>
</tr>
<tr>
<td>Booster Thrust</td>
<td>SG19</td>
<td>1000.0</td>
</tr>
<tr>
<td>Booster ISP</td>
<td>SG20</td>
<td>0.8</td>
</tr>
<tr>
<td>Sustainer Thrust</td>
<td>SG22</td>
<td>285.0</td>
</tr>
<tr>
<td>Sustainer ISP</td>
<td>SG23</td>
<td>0.9</td>
</tr>
<tr>
<td>Centaur Propulsion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thrust</td>
<td>SG27</td>
<td>70.6666</td>
</tr>
<tr>
<td></td>
<td>MU27</td>
<td>15011.86</td>
</tr>
<tr>
<td>Specific Impulse (ISP)</td>
<td>SG28</td>
<td>1.6666</td>
</tr>
<tr>
<td></td>
<td>MU28</td>
<td>423.26777</td>
</tr>
<tr>
<td>Atmosphere</td>
<td>SG35</td>
<td>1.0</td>
</tr>
<tr>
<td>Launch Azimuth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Null Voltage</td>
<td>SG36(1)</td>
<td>0.07</td>
</tr>
</tbody>
</table>
Table 2. Parameter-Controlled Input (Contd)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>PROGRAM NAME</th>
<th>INPUT QUANTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roll-Gyro Torquing Rate</td>
<td>SG36(2)</td>
<td>0.42</td>
</tr>
<tr>
<td>Time Uncertainties</td>
<td>SG36(3)</td>
<td>0.04</td>
</tr>
<tr>
<td>Allowed Tolerance</td>
<td>SG36(4)</td>
<td>0.51</td>
</tr>
<tr>
<td>Pitch Program</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voltage-Time Integral</td>
<td>SG37(1)</td>
<td>0.33</td>
</tr>
<tr>
<td>Gyro-Torquing-Rate Voltage Time Average</td>
<td>SG37(2)</td>
<td>0.83</td>
</tr>
<tr>
<td>Inverter Voltage</td>
<td>SG37(3)</td>
<td>1.17</td>
</tr>
<tr>
<td>Inverter Frequency</td>
<td>SG37(4)</td>
<td>0.75</td>
</tr>
<tr>
<td>Drag Force</td>
<td>SG38</td>
<td>1.7</td>
</tr>
<tr>
<td>Wind Profile</td>
<td>SG39</td>
<td>1.0</td>
</tr>
</tbody>
</table>

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.
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

\[
V(1) = 2.79, \quad V(3) = 1.32
\]

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).
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.

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 $300+50\times\text{(number of cases)}$ for the line estimate. If IRITE $\neq 0$, raise this estimate by 10,000.
APPENDIX
PROGRAM LISTING
COMMON /MD/NGP,NCALC,P39,TICEN,TABLE
DIMENSION NGP(300),NCALC(300),P39(6),TICFN(9),TABLE(500)
COMMON DELP,DELP50,FMSIGMA,FNIT,NITER
COMMON NM,MR,THNOM,ISPNOM,FUTKVL,LXTKVL,PUSER,DPFREQ(600)
COMMON STDDEV(300),MEAN(300),P(600),DIFF(300),SVDF(10),TITLE(20)
DIMENSION DUMSD(10,200)
EQUIVALENCE (DUMSD(1),SG1(1))
COMM TR MUI(10),MU2(10),MU3(10),MU4(10),MU5(10),MU6(10),MU7(10),
1 MU8(10),MU9(10),MU10(10),MU11(10),MU12(10),MU13(10),MU14(10),
2 MU15(10),MU16(10),MU17(10),MU18(10),MU19(10),MU20(10),MU21(10),
3 MU22(10),MU23(10),MU24(10),MU25(10),MU26(10),MU27(10),MU28(10),
4 MU29(10),MU30(10),MU31(10),MU32(10),MU33(10),MU34(10),MU35(10),
5 MU36(10),MU37(10),MU38(10),MU39(10)
COMMON SG1(1),SG2(1),SG3(1),SG4(1),SG5(1),SG6(1),SG7(1),
1 SG8(1),SG9(1),SG10(1),SG11(1),SG12(1),SG13(1),SG14(1),
2 SG15(1),SG16(1),SG17(1),SG18(1),SG19(1),SG20(1),SG21(1),
3 SG22(1),SG23(1),
4 SG24(1),SG25(1),SG26(1),SG27(1),SG28(1),SG29(1),SG30(1),SG31(1),
5 SG32(1),SG33(1),SG34(1),
7 SG43(1),SG44(1),SG45(1),SG46(1),SG47(1),SG48(1),SG49(1),SG50(1),
8 SG51(1),SG52(1),SG53(1),SG54(1),SG55(1),SG56(1),SG57(1),SG58(1),
9 SG59(1),SG60(1)
REAL MAXSET,MINSET,LBPDIS,LBP,MR,MRBAR,MRDBAR,
* IB,IS,IS,LOXRES,LOXAVL,LM2AVL,LM2RES,ISPNOM,MU1,MU2,MU3,
1 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
2 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
3 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
4 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
5 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
6 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
7 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
8 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
9 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
10 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
11 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
12 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
13 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
14 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
15 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
16 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
17 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
18 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
19 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
20 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
21 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
22 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
23 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
24 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
25 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
26 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
27 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
28 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
29 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
30 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
31 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
32 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
33 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
34 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
35 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
36 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
37 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
38 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
39 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
40 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
41 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
42 MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,MUS,
GDI
C-BTD65-176

34HSG18, SG18, 4HSG19, SG19, 4HSG20, SG20, 4HSG21, SG21, 4HSG22, SG22,
4HSG23, 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, 5HSG61, SG61, 5HSG67, SG67,
85HSG87, SG87, 5HSG81, SG81,
*4HGLH2, GLH2, 6HSSUBIAS, SUBIAS, 6HHMPURES, MPURES, 6HVULVLAG, VLVLAG,
*6HHMAXSET, MAXSET, 6HMINSET, MINSET, 6HHBPDIS, HBPDIS, 6HLBPDIS, LBPDIS,
*3HLBP, LBP, 4HTL02, TL02, 4HLG02, GL02, 3HHBP, HBP,
*4HTLH2, TLH2)

L = 1
IF (PLT > EQ 0) GOTO24
D0091 = 1,600
23 DFPFREQ(I) = 0, 0
24 NDO04I = 1, NPARA
J = NGP(I)
IF (J > EQ 0) GOTO30
D009K = 1, J
STDEV(L) = NUMSN(K, 1)
28 L = L + 1
30 CONTINUE
RES = 0
DFLP = 0, 0
DFLPsq = 0, 0
D200000I = 1, NITER
L = 1
D2AL61 = 0, 0
D2AL91 = 0, 0
SUMs = 0, 0
SUMs2 = 0, 0
D209000J = 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 NORD(A PRIME, ANS)
IF (SORT(ANS**2) > GT .3, 0) GOTO100
DIFF(J) = ANS*STDEV(L)
110 GOTO18999
200 NLOOP = NGP(J)
IF (NLOOP > EQ 0) GOTO19000
D0225N = 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 NORD(A PRIME, ANS)
IF (SORT(ANS**2) > GT .3, 0) GOTO205
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 D0225K = 1, 2

47
216 CALL NORAD(PRIMER,ANS)
    IF (SQR(T(ANS**2)) GT 3.0) GOTO 216
215 SUMSUM=SUBSUM+ANS*SG611(K)
    DAL61=SUBSUM*NFTV
    AFD=NREF+SUBSUM
    GOTO 220
208 DO210K=1.3
209 CALL NORAD(PRIMER,ANS)
    IF (SQR(T(ANS**2)) GT 3.0) GOTO 209
210 DIFF(J)=DIFF(J)+ANS*SG671(K)
    GOTO 220
211 DO210K=1.3
212 CALL NORAD(PRIMER,ANS)
    IF (SQR(T(ANS**2)) GT 3.0) GOTO 232
235 DIFF(J)=DIFF(J)+ANS*SG871(K)
    GOTO 220
217 DO210K=1.3
218 CALL NORAD(PRIMER,ANS)
    IF (SQR(T(ANS**2)) GT 3.0) GOTO 218
219 SUMMA1=SUMMA1+ANS*SGA11(K)
    DALA1=SUMMA1*NFTV
    AOD=NMON+SUMMA1
    GOTO 220
240 CALL NORAD(PRIMER,ANS)
    IF (SQR(T(ANS**2)) GT 3.0) GOTO 240
    DEND=ANS*STDEV(L)*NLH2D
    DIFF(J)=DIFF(J)+DEND*NLH2V
241 CALL NORAD(PRIMER,ANS)
    IF (SQR(T(ANS**2)) GT 3.0) GOTO 241
    DIFF(J)=DIFF(J)+ANS*STDEV(L+1)*(NLH2D+DEND)
242 CALL NORAD(PRIMER,ANS)
    IF (SQR(T(ANS**2)) GT 3.0) GOTO 242
    DIFF(J)=DIFF(J)-(ANS*STDEV(L+2)-1.865)*(NLH2D+DEND)
247 L=L+1
    GOTO 19000
250 CALL NORAD(PRIMER,ANS)
    IF (SQR(T(ANS**2)) GT 3.0) GOTO 250
    DEND=ANS*STDEV(L)*NL02D
    DIFF(J)=DIFF(J)+DEND*NL02V
251 CALL NORAD(PRIMER,ANS)
    IF (SQR(T(ANS**2)) GT 3.0) GOTO 251
    DIFF(J)=DIFF(J)+(ANS*STDEV(L+1))*(NL02D+DEND)
252 CALL NORAD(PRIMER,ANS)
    IF (SQR(T(ANS**2)) GT 3.0) GOTO 252
    DIFF(J)=DIFF(J)-(ANS*STDEV(L+2)-1.77)*(NL02D+DEND)
    GOTO 247
260 DIFF(J)=DIFF(J)*AFD+DAL61
    GOTO 220
265 DIFF(J)=DIFF(J)*AOD+DAL81
220 L=L+1
225 CONTINUE
    GOTO 19000
300 IF (J EQ 9) GOTO 310
    DIFF(J)=SUBSUM
    GOTO 18000
310  DIFF(J)=SUM81
      GOTO18000
400  DIFF(J)=N\*0
      GOTO18000
500  CALL NORDAD(PRIMER,ANS)
   IF(SQRT(ANS**2)*GT,300)GOTO500
   DIFF(J)*=ANS
      GOTO18999
600  NLOOP*NGP(J)
      NO62=NN=1,NLOOP
607  CALL NORDAD(PRIMER,ANS)
   IF(SQRT(ANS**2)*GT,300)GOTO603
   SVDIFF(N)=ANS*STDEV(NL)
608  IF(N*NF.R)GOTO620
   IF(MPURF*EQ.21GOTO630
   IF(SVDF(IN)*GE,-MU14(B))GOTO620
   RES=1
   SVDIFF(N)=-PUSSET*(SVDF(IN)+MU14(B))-MU14(B)
620  DIFF(J)=DIFF(J)+SVDIFF(N)
624  L=L+1
625  CONTINUE
      GOTO19000
630  GOTO1701
631  CALL NORDAD(PRIMER,R2)
   IF(ABS(R2)*GT,300)GOTO631
   R2=R2*HBPDIS
   HBP=HBP+R2-TLH2+SVDF(2)
635  CALL NORDAD(PRIMER,R3)
   IF(ABS(R3)*GT,300)GOTO635
   R3=R3*HBPDIS
   LBP=LBP+R3-TLQ2+SVDF(1)
   CPU=MU14(R)
   CMS=-(CPU-SUBIAS)
   IF(SVDF(N)*LT,CMS)GOTO680
   MRBAR=((MR*(W1DOT+W2DOT)*VLVLAG*(PUSSET-MAXSE1))/((20*LBP*(MR+10))
   )+MAXSET
   SVDF(N)=SVDF(N)+CPU-SUBIAS
   IF(SVDF(N)*GT,00)GOTO637
   SVDF(N)*HBP-(LBP/MRBAR)
   IF(SVDF(N)*GE,00)GOTO636
   SVDF(N)=-SVDF(N)*MRBAR
636  DIFF(J)=DIFF(J)+SVDF(N)
      GOTO18999
637  SVDF(N)=SVDF(N)-CPU+HBP-(LBP/MRBAR)
   IF(SVDF(N)*GE,00)GOTO636
   SVDF(N)=SVDF(N)*MRBAR
      GOTO636
640  CALL NORDAD(PRIMER,R4)
   IF(ABS(R4)*GT,300)GOTO680
   R4=R4*PUSSET*STDEV(L)
   MRBAR=((W1DOT+W2DOT)*VLVLAG*(PUSSET-MINSE1))/((MR+10)*20*LBP)
   IMINSET
   SVDF(N)=R4-(CPU-SUBIAS)*PUSSET
   IF(SVDF(N)*GT,00)GOTO681
   SVDF(N)=LBP-MRBAR*HBP

49
IF(SVDF1F(N)*GF.0.0) GO TO 636
SVDF1F(N)=SVDF1F(N)/MRDBAR
GOTO636

681 SVDF1F(N)=SVDF1F(N)+LRP-MRDBAR*HRP
IF(SVDF1F(N)*GF.0.0) GO TO 636
SVDF1F(N)=SVDF1F(N)/MRDBAR
GOTO636

700 TR=0.0
705 CALL NORAD(Primer,ANS)
IF(SORT(ANS**2)*GT.3.0)GOTO705
DIFF(J)*TR=ANS*STDVF(L)
GOTO18999
800 IF=0.0
805 CALL NORAD(Primer,ANS)
IF(SORT(ANS**2)*GT.3.0)GOTO805
DIFF(J)*IB=ANS*STDVF(L)
GOTO18999
900 TS=0.0
905 CALL NORAD(Primer,ANS)
IF(SORT(ANS**2)*GT.3.0)GOTO905
DIFF(J)*TS=ANS*STDVF(L)
GOTO18999
1000 IS=0.0
1005 CALL NORAD(Primer,ANS)
IF(SORT(ANS**2)*GT.3.0)GOTO1005
DIFF(J)*IS=ANS*STDVF(L)
GOTO18999
1100 TV=0.0
1105 CALL NORAD(Primer,ANS)
IF(SORT(ANS**2)*GT.3.0)GOTO1105
DIFF(J)*TV=ANS*STDVF(L)
GOTO18999
1200 IV=0.0
1205 CALL NORAD(Primer,ANS)
IF(SORT(ANS**2)*GT.3.0)GOTO1205
DIFF(J)*IV=ANS*STDVF(L)
GOTO18999
1300 IF(MPURS*EQ.0)GOTO1370
1301 LHRES=MU14(1)+SVDF1F(1)+MU14(3)+SVDF1F(3)
LOXVNT=MU16(1)+DIFF(16)
IF(RES*EQ.0)GOTO1310
LOXRES=LOXRES+SVDF1F(8)
1310 LOXAVL=MU11(1)+DIFF(11)-LOXRES-LOXVNT
LH2RES=MU14(2)+SVDF1F(2)+MU14(4)+SVDF1F(4)
LH2VNT=MU15(1)+DIFF(15)
IF(RES*EQ.0)GOTO1315
LH2RES=LH2RES+SVDF1F(8)
1315 LH2AVL=MU10(1)+DIFF(10)-LH2RES-LH2VNT
MR=LOXAVL/LH2AVL

CALL TABL(MR,THNOM,TICEN(1),TICEN(4),1,1,1,1,3,1,IFBAD)
GOTO(1318,1317),IFBAD
1317 THNOM=0.0
1318 CALL TABL(MR,ISPNOM,TICEN(1),TICEN(7),1,1,1,1,3,1,IFBAD)
GOTO(1320,1319),IFBAD
1319 ISPNUM=0.0
1320 CALL NORM(PRIMFR,ANS)
1325 IF(SQRT(ANS**2) GT 3.0) GOTO 1320
1328 DT="=ANS*STDEV(L)
1330 CALL NORM(PRIMFR,ANS)
1335 IF(SQRT(ANS**2) GT 3.0) GOTO 1335
1338 DT="=ANS*SG28(1)
1340 W1DOT=THNOM+DTE1/(ISPNUM+IDE1)
1345 W2DOT=THNOM+DTE2/(ISPNUM+IDE2)
1350 IF(MPREF EQ 2) GOTO 0631
1355 DIFF(J)=DT1+DTE2+P*0*(THNOM-MU27(1))
1360 GOTO 8999
1370 DIFF(J)=DIFF(27)+2.0*MU27(1))/(W1DOT+W2DOT)-MU28(1)
1380 GOTO 8999
1390 L=L+1
1400 CONTINUE
1405 DIFF(13)=DIFF(13)+23.0
1410 IF(NITFR.NE.100) GOTO 19001
1415 WRTF(6,20024) (DIFF(N),N=1,NPARA)
1420 SUM=0.0
1425 DO 19050 K=1,NPARA
1430 IF(K*FQ.39) GOTO 19030
1435 IF(K*NF.10) GOTO 19010
1440 DIFF(K)=DIFF(K)+DIFF(K+1)
1445 M=4*(K-1)+1
1450 IF(P(M).FQ.0.0) GOTO 19050
1455 IF(DIFF(K).GT.0.0) GOTO 19015
1460 PFE=P(M+2)
1465 GOTO 19033
1470 IF(PFE.0.0) GOTO 19033
1475 CALL TABL(DIFF(39),PEE,P39(1),P39(4),1,1,1,3,IFBAD)
1480 GOTO 19033,19031,19031,19031
1485 PFE=0.0
1490 SUM=SUM+DIFF(K)*PFE
1495 IF(K*NF.10) GOTO 19050
1500 K=K+1
1505 CONTINUE
1510 IF(NITFR.NE.100) GOTO 19075
1515 WRTF(6,20024) SUM
1520 DLP=DLP+SUM
1525 DLP50=DLP50+SUM**2
1530 IF(PLT*EQ.0) GOTO 20000
1535 INDEX=SUM+300.0
1540 IF(INDEX.LE.0) GOTO 20000
1545 IF(INDEX.GT.600) GOTO 20000
1550 DPFREQ(INDEX)=DPFREQ(INDEX)+1
1560 CONTINUE
1565 FNIT=NITFR
FM=DELP/FNIT
SIGMA=SQRT((DELP/SQ/FNIT)-FM**2)
WRITE(6,20020)NPARA,NITER
20020 FORMAT(1H0,22X,23HNUMBER OF PARAMETERS = ,14,22X,23HNUMBER OF ITERATIONS = ,16)
20024 FORMAT(8(4X,1PE12.5))
WRITE(6,20026)FM,SIGMA
20026 FORMAT(1H0,20X,14HMEAN OF FPR = ,1PE12.5,30X,21HSTANDARD DEVIATION = ,1PE12.5)
IF(PLOT.EQ.0)GOTO20028
CALL EPLOT
20028 GOTO20
END
$IPFCT FOPLTN FULIST,RF
SUBROUTINE FOPLTN
COMMON /AD/NGP,NCALC,P39,TICEN,TAPE
DIMENSION NGP(300),NCALC(300),P39(6),TICEN(9),TAPE(500)
COMMON DLDP,DLDP55,FX,SIGMA,FINIT,NITER
COMMON NM,AVMR,THNOM,ISPON,M,FUKL,LXLKVL,PUS,DPFREQ(600)
COMMON SDEV(1000),MEAN(300),P(600),DIFF(300),SVDF(10),TITLE(20)
DIMENSION CUMGFX(300),CUMGY(300)
CALL CAMRAV(N3)

YF=0.0
YT=0.0
DO501=1,600
40 IF(DPFREQ(I))LT,YT)GOTO50
YT=DPFREQ(I)
50 CONTINUE
VT=YF+YT*1
XL=-300
XR=300
CALL DXDYV(1,XL,XR,DX,NX,15,0,11)
CALL DXDYV(2,YF,YT,DX,M,J,NY,15,0,11)
CALL GRIDIV(1,XL,XR,YY,YT,DX,DY,MM,-I,-J,NX,NY)
DO75K=1,600
IF(DPFREQ(K))LT,F0.00)GOTO75
FX=K-300
CALL POINTV(FX,DPFREQ(K),-1)
75 CONTINUE
CALL PRINTV(-33,33,PERFORMANCE RESERVE ** LBS,380.2)
CALL PRINTV(60,TITLE(11),272,0)
CALL APRTNV(-14,-1,9HFRQUENCY,2.568)
YF=0.0
YT=1.0
CALL DXDYV(2,YF,YT,DX,M,J,NY,15,0,11)
CALL GRIDIV(1,XL,XR,YY,YT,DX,DY,MM,-I,-J,NX,NY)
K=0
DPSUM=0.0
DO100K=1,600
DPSUM=DPSUM+DPFREQ(K)
IF(DPSUM,F0,00)GOTO100
FX=K-300
SUMP=DPSUM/FINIT
IF(SUMP,LT,95)GOTO6
IF(SUMP,GT,1.0)GOTO86
100 CONTINUE
CALL PRINTV(-33,33HFLIGHT PERFORMANCE RESERVE ** LBS,380.2)
CALL PRINTV(60,TITLE(11),272,0)
CALL APRTNV(0,-14,-11,1HPROBABILITY,0,582)
CALL DXDYV(1,CUM95X(KK),CUM95X(KK),DX,NXI,NX,15,0,11)
CALL DXDYV(2,CUM95Y(KK),CUM95Y(KK),DY,M,J,15,0,11)
CALL GRIDIV(1,CUM95X(KK),CUM95X(KK),CUM95Y(KK),CUM95Y(KK),DY,DX,MM,-I,-J,NX,NY)
CALL APRTVICK,CUM95X,CUM95Y,1.1,1.4,11,1,1,11)
CALL PRINTV(-33,33HFLIGHT PERFORMANCE RESERVE ** LBS,380.2)
CALL PRINTV(60,TITLE(11),272.0)
CALL APRNTV(0,-14,-11,11PROBABILITY,0.582)
RETURN
END

$IRMAP RANDOM
ENTRY UNRAN
* CALL UNRAN (ARG1,ARG2)
* ARG1 PRIME TO BE USED
* ARG2 A RANDOM NUMBER IN FLOATING POINT
* THIS NO IS GREATER OR = ZERO AND LESS THAN ONE
UNRAN SAVE 4 SAVE INDEX 4
LDQ* 3.4 LOAD WITH PREVIOUS RANDOM NO.
MPY 0.32277244615
STO* 3.4 STORE RANDOM NUMBER
CLA 0.200 FLOAT
LRS 8 AND
YCA
FAD = 0 NORMALIZE NUMBER
STO* 4.4 STORE ANS IN ARG2
RETURN UNRAN RETURN TO CALLER
ENTRY NORD
* CALL NORD (ARG1,ARG2)
* ARG1 PRIME TO BE USED
* ARG2 A RANDOM NO. OF VARIANCES
NORD SAVE 1.4 SAVE INDEX 1 AND INDEX 4
LDQ* 3.4 LOAD WITH PREVIOUS RANDOM NO.
MPY 0.32277244615
STO* 3.4 STORE RANDOM NUMBER
MPV = 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 NORD RETURN TO CALLER
RDN
CONTROL NGP,FNDBD
NGP RSS 300
NCALC RSS 200
P30 RSS 6
TICEN RSS 0
TABLE RSS 500
ENDRND NULL END