LAUNCH, A COMPUTER CODE FOR DETERMINING LAUNCH VEHICLE RELIABILITY

September 1977

Final Report

Approved for public release; distribution unlimited

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base, NM 87117
This final report was prepared by the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico, Job Order 20070317. Lieutenant Marcia A. Thornton (NSQ) was the Laboratory Project Officer-in-Charge.

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

This report has been reviewed by the Office of Information (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

MARCIA A. THORNTON  
1Lt, USAF  
Project Officer

FOR THE COMMANDER

DERMOD KELLEHER  
Chief, Power Branch

HERBERT M. FERNANDEZ  
Chief, Nuclear Systems Division

DO NOT RETURN THIS COPY. RETAIN OR DESTROY.
The computer code, LAUNCH, is designed to maintain a data file on the launches of major launch vehicles. From this data file one can obtain a chronological history of any desired vehicle or group of vehicles as well as the historical reliability and/or projected reliability for any future launch. A discussion of the code and its applications are included along with a listing of the program and a sample output results.
PREFACE

The work described in this report is part of the total technical support provided by the Power Branch, Nuclear Systems Division of the Air Force Weapons Laboratory (AFWL/NSQ) to the Directorate of Nuclear Surety (AFISC/SN) on the Viking, Lincoln Experimental Satellites 8 and 9, and Mariner Jupiter/Saturn missions.

The author would like to acknowledge the assistance, in the form of a nonlinear regression curve fitting subroutine, provided by the Mathematics Section, Technical Branch, Technology Division of the AFML.
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTRODUCTION</td>
</tr>
<tr>
<td>II</td>
<td>COMPUTER CODE</td>
</tr>
<tr>
<td>III</td>
<td>SOURCES OF INFORMATION</td>
</tr>
<tr>
<td>IV</td>
<td>COMPUTER DECK STRUCTURE</td>
</tr>
<tr>
<td>V</td>
<td>DATA CARD FORMAT</td>
</tr>
<tr>
<td>VI</td>
<td>SAMPLE OUTPUT</td>
</tr>
<tr>
<td>VII</td>
<td>RESULTS</td>
</tr>
<tr>
<td></td>
<td>APPENDIX</td>
</tr>
<tr>
<td></td>
<td>VARIABLES USED IN LAUNCH</td>
</tr>
<tr>
<td></td>
<td>ABBREVIATIONS</td>
</tr>
<tr>
<td></td>
<td>REFERENCES</td>
</tr>
</tbody>
</table>
# ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Program LAUNCH</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Subroutine RENMER</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Subroutine CHANGE</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>Subroutine VEHICLE</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>Subroutine FAILRAT</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>Subroutine FAILLOC</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>Subroutine CURVIT</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>Deck Structure</td>
<td>13</td>
</tr>
<tr>
<td>9</td>
<td>Data Entry Card</td>
<td>17</td>
</tr>
<tr>
<td>10</td>
<td>Data Output Card</td>
<td>17</td>
</tr>
<tr>
<td>11</td>
<td>Output from Subroutine VEHICLE (Failures Only)</td>
<td>20</td>
</tr>
<tr>
<td>12</td>
<td>Output from Subroutine VEHICLE (All Launches)</td>
<td>21</td>
</tr>
<tr>
<td>13</td>
<td>Output from Subroutine FAILRAT (Chronological)</td>
<td>23</td>
</tr>
<tr>
<td>14</td>
<td>Output from Subroutine FAILRAT (Summary)</td>
<td>24</td>
</tr>
<tr>
<td>15</td>
<td>Output from Subroutine CURVIT</td>
<td>25</td>
</tr>
<tr>
<td>16</td>
<td>Plot of Equations Determined by Subroutine CURVIT</td>
<td>26</td>
</tr>
</tbody>
</table>
SECTION I
INTRODUCTION

The Interagency Nuclear Safety Review Panel (INSRP) is tasked with reviewing the safety analyses of space launches carrying nuclear power sources. After review of the launch safety analysis, the INSRP prepares a Safety Evaluation Report (SER) for submission to the National Security Council and ultimately to the President. The SER contains a recommendation for launch approval or disapproval.

These safety analyses are performed under Energy Research and Development Administration (ERDA) contract and include inputs from many sources including the launch vehicle contractors. For many years the INSRP has accepted the booster vehicle and upper stage failure rates provided by the contractors while wondering how these failure rates compare with the performance history of the vehicle. The Power Branch, Nuclear Systems Division of the Air Force Weapons Laboratory (AFWL) has developed a computer code, LAUNCH, to determine historical failure rates in an effort to resolve the potential differences between contractor supplied and historical failure rates. LAUNCH allows the user to obtain the launch history, historical failure rate, and projected reliability of specific launch vehicles using various reliability growth techniques.

Preliminary results of LAUNCH analysis on Viking, Lincoln Experimental Satellites 8 and 9 (LES 8/9), and Mariner Jupiter/Saturn (MJS) have been incorporated into the SERs for those launches. Similar results for other launches and launch vehicles should prove useful to the INSRP in its review of safety analyses.
SECTION II

COMPUTER CODE

The main program, LAUNCH, is a short bookkeeping program which calls subroutines as directed by data cards input to it. The main program and various subroutines will be discussed individually here. Listings of the main program and subroutines appear in the Appendix. Formats for the data cards are discussed in Section V.

A flowchart of LAUNCH appears as figure 1. The data entry cards are read into the proper arrays. An end-of-file (EOF) card terminates the data entry cards. Subroutine RENMER is called to sort the data entries chronologically and to merge them with the main data file which has been stored on tape or permanent file. The reordered and merged main data file is then written onto tape or permanent file for future use. Data cards indicating the desired output information are read next, and the appropriate subroutines are called. Another EOF card terminates the data cards. This ends the program execution.

RENMER is flowcharted in figure 2. This subroutine uses a "shell" sort to chronologically order the data entries from cards. The reordered set is then written on a scratch file. This scratch file and the main data file are then merged into one single file with any duplicate entries combined into a single entry. Duplicate entries can occur because information is obtained from a variety of sources and more than one source may provide information on a given launch. Program control is returned to LAUNCH to output the requested information using this updated data file.
START

READ DATA ENTRY CARDS

SORT AND MERGE SUBROUTINE

STORE MAIN DATA FILE

TAPE

READ DATA OUTPUT CARDS

TEST ON OUTPUT TYPE

APPROPRIATE SUBROUTINE

END

Figure 1. PROGRAM LAUNCH
Figure 2. SUBROUTINE RENMER
Subroutine CHANGE appears in figure 3 in flowchart form. CHANGE uses a dummy variable as a temporary storage location for switching the items in the data entries.

A flowchart of VEHICLE appears as figure 4. This subroutine takes the data entries which contain the desired output information from the main data file and enters them into a vehicle array. Data entries are tested to determine if they contain the desired vehicle and, if only failures are desired, if the launch result is a failure. The vehicle array is then printed for use.

Subroutine FAILRAT is flowcharted in figure 5. Using success and failure counters, FAILRAT determines the historical reliability after each launch. This can be done using all launches or for only the last NO launches, where NO is supplied by the user. A success is the successful performance of the desired vehicle; a failure of a booster vehicle is a no-test for the upper stage if the upper stage is the desired vehicle.

Figure 6 is a flowchart of FAILLOC. This subroutine determines the percentage of failures occurring during each launch phase: pad, land, ascent, orbital. These percentages can be for all launches or for only the last NO launches where NO is input by the user.

A flowchart of CURVIT appears as figure 7. CURVIT uses a least-squares nonlinear regression subroutine to determine a best fit to the historical data. Four general equations are currently employed (Y = reliability; X = launch number; A, B, C = curve fit parameters):

\[
Y = A + Be^{Cx} \quad (1)
\]

\[
Y = Ae^{Be^{Cx}} \quad (2)
\]

\[
Y = A \left(1 - \frac{B}{e^{Cx} + B}\right) \quad (3)
\]

\[
Y = Ae^{B/x} \quad (4)
\]
Figure 3. SUBROUTINE CHANGE
Figure 4. SUBROUTINE VEHICLE
Figure 5. SUBROUTINE FAILRAT
Figure 6. SUBROUTINE FAILLOC
START

FILL WORKING ARRAYS

OUTPUT RESULTANT PARAMETERS

TEST ON EQN. OF CURVE FIT

APPROPRIATE SUBROUTINE

RETURN

PRINTER

Figure 7. SUBROUTINE CURVIT
SECTION III

SOURCES OF INFORMATION

The information used to build the data file comes from a variety of sources. The original starting point was the TRW Space Log (reference 2). This provided basic information on launches including date, vehicle, project director, and mission success or failure. Another major source was NASA Pocket Statistics (reference 3) which provided the same type of information on NASA missions. Vandenberg AFB Launch Summary (reference 4) yielded additional data on many boosters which have been used as reentry vehicles rather than space launches. In instances where it was available, contractor information on launch vehicles was also included (reference 5, 6).
SECTION IV

COMPUTER DECK STRUCTURE

The program LAUNCH is written in ANSI standard FORTRAN and requires the appropriate control cards. Different parts of the card deck are separated by end-of-file (EOF) cards (also called end-of-section or end-of-partition). Figure 8 depicts the deck structure.

The control cards must request the storage device for the main data file to read and update the file. They also establish the language used and other peripheral equipment desired.

The main program and associated subroutines follow. Data entry cards are next and may be in any order.

The data output cards must be somewhat organized. First, the vehicle array must be formed (KEY = 0). Then any statistics desired may be determined using that array. If statistics on another vehicle are desired, the vehicle array for that vehicle must be formed first. Any number of vehicles can be examined sequentially during a computer run but only three booster and upper stages can be examined at one time. An EOF card terminates the computer run.
Figure 8. DECK STRUCTURE
SECTION V

DATA CARD FORMAT

Data cards are of two types: data entries and data output. Additional information on launches is entered on data entry cards. The type of output desired is coded on data output cards.

Data entry cards have the following format:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>Launch date as an integer=year, month, day</td>
<td>02=Atlas, 03=Scout, 04=Titan II, 05=Titan III, 06=Vanguard, 07=Redstone, 08=Juno II, 09=Saturn, 11=Titan III (w/solid strapons), 12=Titan I, 20=Unknown</td>
</tr>
<tr>
<td>7-8</td>
<td>Booster vehicle designator as an integer</td>
<td>01=Thor, 02=Atlas, 03=Scout, 04=Titan II, 05=Titan III, 06=Vanguard, 07=Redstone, 08=Juno II, 09=Saturn, 11=Titan III (w/solid strapons), 12=Titan I, 20=Unknown</td>
</tr>
<tr>
<td>9-10</td>
<td>Upper stage designator as an integer</td>
<td>01=Agena, 02=Centaur, 03=Able, 04=Delta, 05=Burner II, 06=Transtage, 10=None, 20=Unknown</td>
</tr>
<tr>
<td>11-14</td>
<td>Project director as an alphanumeric (e.g., USAF, NASA, USN)</td>
<td>20=Unknown</td>
</tr>
<tr>
<td>Col 15</td>
<td>Source of information as an alphanumeric</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------------</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>Booster Contractor</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>Air Force Eastern Test Range (AFETR)</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>NASA Pocket Statistics</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>Other</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>TRW SpaceLog</td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>Upper Stage Contractor</td>
<td></td>
</tr>
<tr>
<td>W</td>
<td>Western Test Range at Vandenberg Air Force Base (VAFB)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Col 16</th>
<th>Launch result as an alphanumeric</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Failure</td>
</tr>
<tr>
<td>S</td>
<td>Success</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Col 17</th>
<th>Failure phase as an alphanumeric</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Ascent</td>
</tr>
<tr>
<td>L</td>
<td>Land</td>
</tr>
<tr>
<td>O</td>
<td>Orbital</td>
</tr>
<tr>
<td>P</td>
<td>Pad</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Col 18</th>
<th>Failed stage as an alphanumeric</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Booster Vehicle</td>
</tr>
<tr>
<td>U</td>
<td>Upper Stage</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Col 19</th>
<th>Type of launch as an integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Space Launch</td>
</tr>
<tr>
<td>2</td>
<td>Training</td>
</tr>
<tr>
<td>3</td>
<td>Test</td>
</tr>
</tbody>
</table>
4=Reentry Test Vehicle
5=Suborbital
6=Unknown

Col 20  Multiple launch indicator as an integer
O=First launch of given vehicle on that date
1=One multiple launch of vehicle on that date

The sample data entry card in Figure 9 adds a launch of a Thor Delta spacecraft combination on 13 May 1960 to the main data file. The launch was a NASA space launch which failed during ascent due to an upper state failure. Information is available from Thor contractor.

Data output cards have the following format:

Col 1-2  Output key as an integer
Negative=Automatic call sequence to output complete statistics on desired launch vehicle (this does not form vehicle array)
O=Vehicle history output (this must be done before statistics are requested because the statistics are derived from the vehicle array)
1=Failure rate statistics
2=Failure location statistics
50=Curve fit of data using forms as requested in Cols. 21-24

Col 3-4  Booster vehicle designator (see data entry card Col 7-8) - If zero, any booster may be considered.
Figure 9. DATA ENTRY CARD

Figure 10. DATA OUTPUT CARD
Col 5-6  Upper stage designator (see data entry card Col 9-10) -
If zero, any upper stage may be considered.

Col 7-10  Percentage of launches to be used in deriving statistics
(zero causes all launches to be considered, negative
number causes a summary only to be printed).

Col 11  Blank

Col 12  Launch result to be considered
F=Failures only
S=All launches

Col 13-14,
17-18  Additional booster vehicle designators

Col 15-16,
19-20  Additional upper stage designators

Col 21-24  Curve-fit forms indicators
Col 21=1  fit to $A+Be^{CX}$
Col 22=1  fit to $Ae^{Be^{CX}}$
Col 23=1  fit to $A (1 - \frac{B}{CX+B})$
Col 24=1  fit to $Ae^{B/x}$

The sample data output card in figure 10 requests the vehicle history
(failures only) for the Thor Agena spacecraft combination.
SECTION VI

SAMPLE OUTPUT

Figures 11 to 15 show the output generated by LAUNCH. A discussion of each figure will demonstrate its features and uses. The Centaur upper stage will be used as an example.

Figures 11 and 12 are representative of the output from subroutine VEHICLE. The output is labeled for the vehicle being considered and for failures only or all launches. The columns are basically self-explanatory. Abbreviations for the source of information are:

TRW - TRW Space Log
NASA - NASA Pocket Statistics
BV - Booster vehicle contractor
UV - Upper stage contractor
ETR - Air Force Eastern Test Range
WTR - Western Test Range (Vandenberg Air Force Base)
OTHER - Other sources

The launch types are:

SPACE - Orbital flight or space probe
TRNG - Training (these are boosters only in general)
TEST - Vehicle test
RTV - Reentry test vehicle
SUBORB - Suborbital flight
## Total Failures for Centaur GDC

<table>
<thead>
<tr>
<th>Date</th>
<th>Booster</th>
<th>Upper Stage</th>
<th>Project Director</th>
<th>Source of Information</th>
<th>Launch Type</th>
<th>Launch Result</th>
<th>Failure Stage</th>
<th>Failure Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/8/62</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>X X X X</td>
<td>Space</td>
<td>F</td>
<td>U</td>
<td>A</td>
</tr>
<tr>
<td>6/30/64</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>X X X X</td>
<td>Space</td>
<td>F</td>
<td>U</td>
<td>A</td>
</tr>
<tr>
<td>12/11/64</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>X X X X</td>
<td>Space</td>
<td>F</td>
<td>U</td>
<td>0</td>
</tr>
<tr>
<td>11/30/70</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>X X X X</td>
<td>Space</td>
<td>F</td>
<td>U</td>
<td>0</td>
</tr>
<tr>
<td>5/8/71</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>X X X X</td>
<td>Space</td>
<td>F</td>
<td>U</td>
<td>A</td>
</tr>
<tr>
<td>2/11/74</td>
<td>Titan  I</td>
<td>Centaur</td>
<td>US</td>
<td>X X X</td>
<td>Space</td>
<td>F</td>
<td>U</td>
<td>A</td>
</tr>
</tbody>
</table>

**Figure 11** Output from Subroutine VEHICLE (Failures Only)

20
<table>
<thead>
<tr>
<th>DATE</th>
<th>BOOSTER</th>
<th>UPPER STAGE</th>
<th>PROJECT DIRECTOR</th>
<th>SOURCE OF INFORMATION</th>
<th>LAUNCH TYPE</th>
<th>RESULT</th>
<th>FAILURE STAGE LOCATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/8/62</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>F</td>
<td>U A</td>
</tr>
<tr>
<td>11/27/63</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td>U A</td>
</tr>
<tr>
<td>6/30/64</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>F</td>
<td>U A</td>
</tr>
<tr>
<td>12/11/64</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>F</td>
<td>U O</td>
</tr>
<tr>
<td>3/2/65</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>F</td>
<td>B P</td>
</tr>
<tr>
<td>8/11/65</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>5/30/66</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>9/20/66</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>10/26/66</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>7/14/67</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>11/7/67</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>1/7/68</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>8/10/68</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>12/7/68</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>2/24/69</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>8/12/69</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>11/30/70</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>1/25/71</td>
<td>Atlas</td>
<td>Centaur</td>
<td>CSC</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>5/30/71</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>F</td>
<td>U A</td>
</tr>
<tr>
<td>12/19/71</td>
<td>Atlas</td>
<td>Centaur</td>
<td>CSC</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>1/23/72</td>
<td>Atlas</td>
<td>Centaur</td>
<td>CSC</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>6/13/72</td>
<td>Atlas</td>
<td>Centaur</td>
<td>CSC</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>8/21/72</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>4/6/73</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>8/23/73</td>
<td>Atlas</td>
<td>Centaur</td>
<td>CSC</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>11/3/73</td>
<td>Atlas</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>2/11/74</td>
<td>TIII Solid</td>
<td>Centaur</td>
<td>US</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>F</td>
<td>U A</td>
</tr>
<tr>
<td>11/21/74</td>
<td>Atlas</td>
<td>Centaur</td>
<td>CSC</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>12/10/74</td>
<td>TIII Solid</td>
<td>Centaur</td>
<td>Germ</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>20/075</td>
<td>TIII Solid</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>9/9/75</td>
<td>TIII Solid</td>
<td>Centaur</td>
<td>NASA</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>1/15/76</td>
<td>TIII Solid</td>
<td>Centaur</td>
<td>Germ</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>1/29/76</td>
<td>Atlas</td>
<td>Centaur</td>
<td>CSC</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>5/13/76</td>
<td>Atlas</td>
<td>Centaur</td>
<td>USAF</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>7/22/76</td>
<td>Atlas</td>
<td>Centaur</td>
<td>USAF</td>
<td>TRW NASA BV UV ETR</td>
<td>Space</td>
<td>S</td>
<td></td>
</tr>
</tbody>
</table>

Figure 12. Output from Subroutine VEHICLE (All Launches)

21
Failure locations used are:

P - Failure occurred on launch pad or resulted in a pad impact
L - Failure occurred over land
A - Failure occurred during ascent over water before achieving orbit
O - Failure occurred in orbital phase

Note that the failures in figure 11 are only Centaur failures; the booster failure is not listed. The launches are listed in chronological order and show the basic information at a glance. If more detailed information is desired, it can be obtained from one of the sources marked with an "X".

Figure 13 is a sample output from subroutine FAILRAT. The columns are self-explanatory. Note that the failure of the booster does not affect the success ratio of the upper stage.

The percentages of failures during the various launch phase are shown in figure 14. If the phase of failure is not known, that failure is not considered in determining the percentages.

Figure 15 shows the output of subroutine CURVIT, the reliability growth curve-fitting subprogram. The equations can be plotted as in figure 16 with the historical data to graphically show the trends for the vehicle being considered.
# CENTAUR GDC

Adjusted Cumulative Success/Failure Ratio for Last 33 Flights of a Total of 41.

<table>
<thead>
<tr>
<th>LAUNCH NUMBER</th>
<th>RESULT</th>
<th>STAGE</th>
<th>TOTALS</th>
<th>PERCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S</td>
<td></td>
<td>S</td>
<td>F</td>
</tr>
<tr>
<td>1</td>
<td>F</td>
<td>U</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>F</td>
<td>U</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>F</td>
<td>U</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>F</td>
<td>B</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>S</td>
<td></td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>S</td>
<td></td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td></td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>S</td>
<td></td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>S</td>
<td></td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td></td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>S</td>
<td></td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>S</td>
<td></td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>S</td>
<td></td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>S</td>
<td></td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>16</td>
<td>S</td>
<td></td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>17</td>
<td>S</td>
<td></td>
<td>13</td>
<td>3</td>
</tr>
<tr>
<td>18</td>
<td>S</td>
<td></td>
<td>14</td>
<td>3</td>
</tr>
<tr>
<td>19</td>
<td>S</td>
<td></td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>20</td>
<td>S</td>
<td></td>
<td>16</td>
<td>3</td>
</tr>
<tr>
<td>21</td>
<td>F</td>
<td>U</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>22</td>
<td>S</td>
<td></td>
<td>17</td>
<td>4</td>
</tr>
<tr>
<td>23</td>
<td>F</td>
<td>U</td>
<td>17</td>
<td>5</td>
</tr>
<tr>
<td>24</td>
<td>S</td>
<td></td>
<td>18</td>
<td>5</td>
</tr>
<tr>
<td>25</td>
<td>S</td>
<td></td>
<td>19</td>
<td>5</td>
</tr>
<tr>
<td>26</td>
<td>S</td>
<td></td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>27</td>
<td>S</td>
<td></td>
<td>21</td>
<td>5</td>
</tr>
<tr>
<td>28</td>
<td>S</td>
<td></td>
<td>22</td>
<td>5</td>
</tr>
<tr>
<td>29</td>
<td>S</td>
<td></td>
<td>23</td>
<td>5</td>
</tr>
<tr>
<td>30</td>
<td>S</td>
<td></td>
<td>24</td>
<td>5</td>
</tr>
<tr>
<td>31</td>
<td>S</td>
<td></td>
<td>25</td>
<td>5</td>
</tr>
<tr>
<td>32</td>
<td>S</td>
<td></td>
<td>26</td>
<td>5</td>
</tr>
<tr>
<td>33</td>
<td>F</td>
<td>U</td>
<td>26</td>
<td>6</td>
</tr>
<tr>
<td>34</td>
<td>S</td>
<td></td>
<td>27</td>
<td>5</td>
</tr>
<tr>
<td>35</td>
<td>S</td>
<td></td>
<td>27</td>
<td>5</td>
</tr>
<tr>
<td>36</td>
<td>S</td>
<td></td>
<td>28</td>
<td>4</td>
</tr>
<tr>
<td>37</td>
<td>S</td>
<td></td>
<td>29</td>
<td>3</td>
</tr>
<tr>
<td>38</td>
<td>S</td>
<td></td>
<td>30</td>
<td>3</td>
</tr>
<tr>
<td>39</td>
<td>S</td>
<td></td>
<td>30</td>
<td>3</td>
</tr>
<tr>
<td>40</td>
<td>S</td>
<td></td>
<td>30</td>
<td>3</td>
</tr>
<tr>
<td>41</td>
<td>S</td>
<td></td>
<td>30</td>
<td>3</td>
</tr>
</tbody>
</table>

Figure 13  Output from Subroutine FAILRAT (Chronological)
## CENTAUR GDC

Adjusted Failures Classed by Location for Last 33 Flights of a Total of 41.

<table>
<thead>
<tr>
<th>LOCATION</th>
<th>FAILURES</th>
<th>PERCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pad</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>Land</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>Ascent</td>
<td>2</td>
<td>66.67</td>
</tr>
<tr>
<td>Orbital</td>
<td>1</td>
<td>33.33</td>
</tr>
<tr>
<td>Total</td>
<td>3</td>
<td>100.00</td>
</tr>
</tbody>
</table>

Figure 14  Output from Subroutine FAILRAT (Summary)
The curve fit for this data is of the form:

\[ \text{Reliability} = A \times \exp(B \times \exp(C \times \text{Launch Number})) \]

With the Parameters A, B, C as follows:

\[
\begin{align*}
A &= .8099545744E+00 \\
B &= -.8207599285E+00 \\
C &= -.9104745067E+00
\end{align*}
\]

The curve fit for this data is of the form:

\[ \text{Reliability} = A \times (1 - B/(C \times \text{Launch Number} + B)) \]

With the Parameters A, B, C as follows:

\[
\begin{align*}
A &= .876411049E+00 \\
B &= .2880982693E+15 \\
C &= .2880982693E+15
\end{align*}
\]

The curve fit for this data is of the form:

\[ \text{Reliability} = A + B \times \exp(C \times \text{Launch Number}) \]

With the Parameters A, B, C as follows:

\[
\begin{align*}
A &= .8199671974E+00 \\
B &= -.2989172644E+00 \\
C &= -.4309589840E+00
\end{align*}
\]

The curve fit for this data is of the form:

\[ \text{Reliability} = A \times \exp(B/\text{Launch Number}) \]

With the Parameters A, B as follows:

\[
\begin{align*}
A &= .8328733128E+00 \\
B &= -.3642819210E+00
\end{align*}
\]

Figure 15. Output from Subroutine CURVIT
Figure 16. Plot of equations determined by subroutine CURVIT

\[ \text{RELIABILITY} = 0.81 \times \exp(-0.82 \times \exp(-0.91 \times \text{LAUNCH})) \]

\[ \text{RELIABILITY} = 0.82 - 0.38 \times \exp(-0.43 \times \text{LAUNCH}) \]

\[ \text{RELIABILITY} = 0.88 \times (1 - 0.29 \times \exp(-0.29 \times \text{LAUNCH} + 0.29 \times 0.15)) \]

\[ \text{RELIABILITY} = 0.83 \times \exp(-0.36 \times \text{LAUNCH}) \]
SECTION VII
RESULTS

LAUNCH has been used to provide preliminary results for use with the safety analyses of the Viking, LES 8/9, and MJS missions. These results give a historical reliability which is used to scale the failure probabilities originated by the analyses.

The booster vehicle for all three missions is the Titan III with strap-on solid rocket motors. This vehicle has had 32 flights and 32 successes. Therefore, it is considered very reliable, and the manufacturer's failure probability of 98% could be used directly.

The Viking and MJS missions use a Centaur upper stage. From Jan 1968 to Jun 1974, there were 21 launches with 19 successes which is a reliability of 90.48%. This is lower than the manufacturer's reliability, and some scaling was done to reconcile the numbers.

LES 8/9 uses a Transtage upper stage. From Jan 1968 to Jun 1974, there were 12 launches with 11 successes which is a reliability of 91.67%. This is also lower than the manufacturer's reliability, and scaling was done before the reliabilities were utilized.

Another consideration is the apportionment of failure probabilities to the launch phases. If all launch vehicles are considered, these figures are:

- Pad 5%
- Land 5%
- Ascent 63%
- Orbital 27%
All launches were considered because there have not been sufficient failures of any particular vehicle to accurately predict these percentages.

As information is gathered, more accurate results will be obtained from LAUNCH. These results should prove useful to the INSRP review of nuclear power source launches.
APPENDIX

LISTING OF LAUNCH

The following listing of LAUNCH and its subroutines is provided for the reader who is interested in the details of the program logic. Comments are included to explain the workings of the program and to separate it into logical units.
PROGRAM LAUNCH (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE7=TAPE10) FIII

C TYPE STATEMENTS FOR THE VARIABLES FIII

INTEGER JAGAT FIII
INTEGER G1, G2, G3, G4 FIII
INTEGER JUSTRP, BP1, BP2, BP3, BP4, JU, ALN FIII
INTEGER NP1, NP2, UVP1, UVP2 FIII
INTEGER SUGT, FAIL FIII
REAL LOC FIII

C THE COMMON BLOCKS ARE USED FOR SEPARATE PURPOSES FIII
C BLOCK *DATA* CONTAINS THE MAIN DATA FILE FIII
C BLOCK *DATA-JAGAT* CONTAINS THE OUTPUT SPECIFICATIONS FIII
C BLOCK *STORIA* CONTAINS THE BEGINNING INFORMATION AND VARIABLE COUNTS FIII
C BLOCK *FIT* CONTAINS THE CURVE FIT VARIABLES FIII

COMMON /DATA/ JAGAT, NEWJAT, I3AT, I200+ I2000+ IUPREA, IUPRED, IUPREV, IUPRED FIII
150M(200+), M(3400+), GM(200+), PROG1(200+), ST(300, 7), STG(200+), FIII
2FYP(200+), MULTA(200+), ALN FIII

COMMON /DATA-JAGAT/ JAGAT, NEWJAT, I3AT, I200+ I2000+ IUPREA, IUPRED, IUPREV, IUPRED FIII
150M(200+), M(3400+), GM(200+), PROG1(200+), ST(300, 7), STG(200+), FIII
1UCON(420+), UVP1, UVP2 FIII

COMMON /STORIA/ SUGT, FAIL, 150M, RAT, 150, MEAD, 121 FIII

COMMON /FIT/ X(3200+), Y(3200+), W(3200+), NOKK(25101), THEOR(200+), 4DATA FIII
1RELNR, ABERR, FLDAG, PAR101, NPAR FIII

C THE DATA BLOCKS SET CERTAIN VARIABLES TO ASSIGNED VALUES FIII
C LAUNCH TIME (#LATYP), BOOSTER VELOC. (#LATYP), UPPER STAGE #1 FIII
C BOOSTER CONTRACTOR (#UCON), AND UPPER STAGE CONTRACTOR (#UCON) FIII
C ASSIGNED FIII

DATA LATYP, SM, SPACE, SM, TRUE, EM, TEST, EM, RIV, ENSUBC, EM / FIII

DATA BUEN, ICHMATLAS, ICHSCOUT, ICHMITAN, II + FIII
11M1MITAN, II, ICHGUIAUR, ICHJUNO, II, ICHSATURN FIII
2 + ICHJUPITE, 11M1MITAN, II, SOLIO, ICHMITAN, I + ICH FIII
JUNKNOWN, / FIII

DATA UV, 1, ICHGENAUR, ICHWAVE, ICHDELTA, ICHWAVE, 11M1MITAN, II, ICHUNKNOWN, / FIII

DATA BCON, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, / FIII

DATA UCON, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, ICHMAG, / FIII

READ TH. M4, DATA ENTRY CARDS AND FORM A WORKING ARRAY FIII

TEST FIII
PROGRAM LAUNCH 7/7/74 OPT#1  FTN 4.6+433  08/25/

10(I), ST(II), LTPF(II),/MULTL(II)

C IF THE LAUNCH TYPE IS NOT KNOWN IT IS EQUALS TO THE INDICATOR
C I IS SET TO 6 WHICH GIVES A BLANK LAUNCH TYPE IN THE OUTPUT
C
C IF (LTPF(II).EQ.3) LTPF(II)=0
C
C THE DATA ENTRY CARDS ARE TERMINATED BY AN END-OF-FILE CARD
C
C IF (EOF(I)) I=2
C
C THE ARRAY AST IS BLANKED TO ELIMINATE SPURIOUS DATA WHEN THE
C DATA ENTRY CARDS ARE MERGED WITH THE MAIN DATA FILE
C
2 DO 2 I=4,7
3 STII/UI=1M
GO TO 1

C NEWDAT=I-1
C
C SUBROUTINE RENUMER IS CALLED TO SORT THE DATA ENTRY CARDS INTO
C CHRONOLOGICAL ORDER AND MERGE THEM WITH THE MAIN DATA FILE INTO A SINGLE DATA FILE
C
C CALL RENUMER
C
C WRITE THE DATA ENTRIES ONTO TAPE OR DISK
C
20 DO I=1,9003DAT
25 SQUAII=IM
5 WRITE (4,14) IDATE(I),BOOST(I),UPPER(I),PROJI(I),ST(II),IJ=1,71
15 SOUN(I),RESII,OCIII,STGII,LTIP(II),MULTL(II)

C WRITE AN END-OF-FILE MARK AND KEINO THE FILE
C
C REMIND 4
C
C DATA OUTPUT CARDS ARE READ FROM THE CARD READER
C THESE CARDS INDICATE THE TYPE OF OUTPUT DESIRED
C
6 READ 15,13 KEY,IV,UK,NO,RESU,VI1,VI2,VI3,VI4
13 AN END-OF-FILE CARD TERMINATES THE DATA OUTPUT CARDS
C IF (EOF(I)) I=1
C
C A TEST ON *KEY* DETERMINES WHICH OUTPUT SUBROUTINES ARE CALLED
C
7 IF (KEY).EQ.1
C
C WHEN *KEY=* THE VELOCITIES ARRAY IS FURNHED ACCORDING TO THE
C SPECIFICATIONS ON THE DATA OUTPUT CARD
C
C CALL VEHICLE
C 10 TO A
C
C WHEN *KEY* IS POSITIVE ITS VALUE DETERMINES WHICH SUBROUTINE IS CALLED
CALCULATE THE VALUE OF *NU* TO DETERMINE THE PERCENTAGE
OF LAUNCHES TO BE CONSIDERED BY THE SUBROUTINES

9 IF (KEY.EQ.1) CALL FAILRAT (NU)
   IF (KEY.EQ.2) CALL FAILLUC (NU)
   IF (KEY.EQ.5) CALL CURVIT (NO,CL,CL3,CLN)
   GO TO 6

WHEN *KL* IS NEGATIVE AN AUTOMATIC SEQUENCE OF SUBROUTINES IS:
CALCULATE THE VALUE OF *NO* TO DETERMINE THE PERCENTAGE
OF LAUNCHES TO BE CONSIDERED BY THE SUBROUTINES. IN ADDITION
THE SEQUENCE AUTOMATICALLY USES ALL LAUNCHES FOR CALLS TO THE
SUBROUTINES

10 CALL FAILRAT (NO)
    CALL FAILRAT (2)
    CALL FAILLUC (NO)
    CALL FAILLUC (1)
    CALL CURVIT (NO,1,1,1,1)
    GO TO 6

11 CONTINUE
THE PROGRAM IS COMPLETED

CALL EXIT

FORMAT STAT. ENDS

12 FORMAT (I6,2I2,A4,4A1,2I1)
13 FORMAT (3I2,2I2,A2,4I2,4I1)
14 FORMAT (16,2I2,A1,11A1,2I1)
END
SUBROUTINE HENNER

C

TYPE STATEMENTS

C

HEAL LOC.

INTEGER I_DATE, I_DATE (11), I_DATE (11+1), I_DATE (11+1+1)

INTEGER PREPST (12), PREPST (11), PREPST (11+1)

INTEGER BOOT-UPPER

DIMENSION CH (11), CH (11+1), CH (11+1+1)

C

COMMON BLOCKS

C

COMMON /DATEI4, ORGAT, HENNEAT, I_DATE (12-1), I_DATE (11+1), I_DATE (11+1+1)

C

PERFORM A SHELL-SORT ON THE DATA-ENTRY CARDS TO ORDER THEM

C

CHRONOLOGICALLY

C

N=HENNEAT

1 N=N/2

IF (N.EQ.2) GO TO 10

K=HENNEAT-M

1 J=1

3 I=1

4 IF (IDATE (I)-IDATE (I+M)) .GE. 0

5 IF (BOOTST (I)-BOOTST (I+M)) .GE. 0

T

6 IF (UPPERST (I)-UPPERST (I+M)) .GE. 0

7 IF (MULTRA (I)-MULTRA (I+M)) .GE. 0

SUBROUTINE *CHANGE* CHANGES ENTRY *I* WITH ENTRY *I+N*

5 CALL CHANGE (I, I+N)

6 I=I+N

7 IF (I.GE.1) GO TO 3

9 J=1

10 IF (J=M) GO TO 10

11 WRITE (I3,20) I, CH (I)

10 WRITE (I3,20) I, CH (I), CH (I+1), CH (I+1+1)

C

WRITE THE REORDERED ENTRIES ON A FILE FOR COMPARISON WITH THE

MAIN DATA FILE

C

11 WRITE (I3,30) I, CH (I), CH (I), CH (I), CH (I)

C

EMOBO 3

12 WRITE THE MAIN DATA FILE AND THE REORDERED DATA ENTRIES INTO A

SINGLE FILE WITH NO DUPLICATE ENTRIES (POSSIBLE SINCE MANY

SOURCES OF INFORMATION ARE USED)

THE MAIN DATA FILE IS ON TAPE 1.

THE DATA ENTRIES ARE ON TAPE 2

C

INIT

10

SET COUNTER TO ONE
SUBROUTINE RENUMER 74/74  DATE=1 FIN 4,0+433  U4/2

I=1

C SET SAME FILE INDICATOR TO ZERO

C IF AG=0

C IF NO DATA ENTRY CARDS HAVE BEEN READ, COPY DATA FILE FROM
C TAPE10 TO TAPE8
C SET SAME FILE INDICATOR TO ONE

C IF (NEQDAT+20) IF AG=1
C IF (NEQDAT+20) IN=10

C READ FILE

C READ (IN, J4) IVS1, IVS2, IVS3, IVS4, IV51, IV52, IV53, IV54, IV55(11), II = 17, IV56, IV57, IV58, IV59, I
C IV60, IV61
C CHECK FOR (N)-OF-FILE
C IF (GOF(111)) 32, 12
C READ FILE

12 READ (10, J4) I1+1, I1+12, I1+13, I1+14, I1+15 (11), II = 17, IV6, IV7, IV8, IV9, I
C IV10, IV11
C CHECK FOR ENV-OF-FILE
C IF (GOF(111)) 34, 13
C COMPAR LAUNCH DATES -
C IF (GOF(111)) 34, 13
C IF NOT EQUAL, COMPARE FURTHER
C IF NOT EQUAL, FILL DATA ARRAY WITH EARLIER LAUNCH ENTRY

13 IF (IV11 = IV51) IV = IV14
C MAIN DATA FILE HAS EARLIER LAUNCH

14 IDATE(11) = IV51
C AGO5T(11) = IV52
C UPRO(11) = IV53
C PK01(11) = IV54
C DD 15 = 1-7
C 15 STILL(11) = IV55(1)
C IF (IV56 = 12, 14) STILL(11) = 1MX
C IF (IV59 = 20, 1MX) STILL(11) = 1MX
C IF (IV56 = 20, 1MX) STILL(11) = 1MX
C IF (IV59 = 20, 1MX) STILL(11) = 1MX
C IF (IV56 = 20, 1MX) STILL(11) = 1MX
C IF (IV59 = 20, 1MX) STILL(11) = 1MX
C RES(11) = 1HS
C IF (IV57 = 20, 1HE) RES(11) = 57
C LOC(11) = 158
C SIG(11) = 63
C LTYR(11) = IV60

34
SUBROUTINE RENNER 7474 3T=1 5TN 4624 333 08/1982

MULTA(I)=IV41

INCREMENT COUNTER

I=I+1

READ FILE

READ (14,41) I(1:1,1,62,1,61,1,64,1,661,11),I,17),VL0,VL7,VL8,VL9,1

IF (EOF(I)) 42,13

DATA ENTRY HAS EARLIER LAUNCH

IF (IDATE(I))=I+1

BOOST(I)=I+1

UPPER(I)=I+1

PHASE(I)=I+1

DO 17 IF J=17

17 ST(I,J)=VLS(J)

IF (V16.EQ.1MH) ST(I,11)=1MH

IF (V17.EQ.1MH) ST(I,2)=1MH

IF (V16.EQ.1MH) ST(I,1)=1MH

IF (V16.EQ.1MH) ST(I,4)=1MH

IF (V16.EQ.1MH) ST(I,5)=1MH

IF (V16.EQ.1MH) ST(I,6)=1MH

IF (V16.EQ.1MH) ST(I,7)=1MH

RES(I)=14S

IF (V17.EQ.1MH) RES(I)=V17

LOC(I)=I+8

ST(I)=I+9

LYTP(I)=IV26

MULTA(I)=IV21

INCREMENT COUNTER

I=I+1

READ FILE

READ (14,51) IV11,IV12,IV13,IV14,IV15,II1),I,17),VL0,VL7,VL8,VL9,1

IV26,IV21

CHECK FOR END OF FILE

IF (EOF(I)) 31,13

CHECK FOR THE SAME BOOSTER VEHICLE

13 IF (IV12.EQ.52) 10,19,14

CHECK FOR THE SAME UPPER STAGE

13 IF (IV13.EQ.53) 10,20,14

35
SUBROUTINE RINNII 74-74 30F#L FTN 4,4433 38/1

C CHECK FOR THE SAME PROJECT DIRECTOR
C
20 IF (IV14=IV541) G0T21.14
C
C CHECK FOR MULTIPLE LAUNCH OF SAME VEHICLE
C
21 IF (IV21-IV81) 16,22,14
C
C COMBINE THE DUPLICATE INFORMATION INTO A SINGLE ENTRY
C
22 DATE(II)=IV17
BOOST(II)=IV12
UPPER(II)=IV16
MLOOL(II)=IV14
30 23 J=4,7
JF (IV55.II,NC,14) J=IV55.II
C
23 CONTINUE
C IF (IV16.EQ.1HF.32,IV56.EQ.1HF) SI(II)=1MX
IF (IV16.EQ.1MN,IV56.EQ.1MN) SI(II)=1MX
IF (IV16.EQ.1HU.32,IV56.EQ.1HU) SI(II)=1MX
IF (IV16.EQ.1HU.04,IV56.EQ.1HU) SI(II)=1MX
IF (IV16.EQ.1MN,IV56.EQ.1MN) SI(II)=1MX
IF (IV16.EQ.1MN,IV56.EQ.1MU) SI(II)=1MX
RES(II)=1MS
IF (IV17.EQ.1HF.32,IV57.EQ.1HF) RES(II)=1HF
LUC(II)=1M
IF (IV50,NE,1M) LUC(II)=IV4
IF (IV58,NE,1M) LUC(II)=IV8
STG(II)=IV1
IF (IV69,NE,IV1) STG(II)=IV9
IF (IV59,NE,1M) STG(II)=IV9
LTP(II)=6
IF (IV2,NE,1M) LTP(II)=IV2
IF (IV6,NE,1M) LTP(II)=IV6
MULTL(II)=0
IF (IV61,NE,1M) MULTL(II)=IV21
IF (IV61,NE,1M) MULTL(II)=IV61
C
C INCREASE COUNTERS
C
I=IV1
C
READ FILE
C
C IF (IV51,IV52,IV53,IV54,IV56111,IF=17IV5,IV57,IV58,IV59,IV61
C
C CHECK FOR END OF FILE
C
IF (E0E(1M)) 32,24
C
C CHECK FOR DIP_DATE ENTRY
C
24 IF (IDAT(II)=IV51) 16,25,10
25 IF (900:1(I-L1) 1LV52) 38.26.30 F1
26 IF (UPPER(I-L1) 1LV53) 30.27.30 F1
27 IF (PROD(I-L1) 1LV54) 30.28.30 F1
28 IF (MULT(I-L1) 1LV61) 30.29.30 F1
29 I=I-1 F1
GO TO 21 F1
C C
READ FILE F1
C C
30 READ (I0.J-L1 IVI1+1VI12+I14+115.I11 +II=LI7.I16+117.118.+119.1 F1
1V2+IV21 F1
C C
CHECK FOR END-OF-FILE F1
C C
IF (EOF(I0)) 31.33 F1
C C
CHECK DUPLICATE FILE INDICATOR TO SEE IF 36TH END-OF-FILE MARKS F1
C C
HAVE BEEN READ F1
C C
32 IF (FFLAG=6) GO TO 33 F1
C C
CHANGE NUMBER OF FILE BEING READ SO THAT ONLY ONE FILE IS BEING F1
C C
USED F1
C C
I0=3 F1
C C
SET DUPLICATE INDICATOR FOR ONE TO INDICATE THAT ONE END-OF-FILE F1
C C
HAS BEEN READ F1
C C
IFFLAG=1 F1
C C
READ FILE F1
C C
READ (I0.J-L1 IVI1+1VI12+I14+115.I11 +II=L7.+116+117+118.+119.1 F1
1V2+IV21 F1
C C
CHECK FOR END-OF-FILE F1
C C
IF (EOF(I0)) 31.33 F1
C C
CHECK DUPLICATE FILE INDICATOR TO SEE IF 36TH END-OF-FILE MARKS F1
C C
HAVE BEEN READ F1
C C
32 IF (FFLAG=6) GO TO 33 F1
C C
CHANGE NUMBER OF FILE BEING READ SO THAT ONLY ONE FILE IS BEING F1
C C
USED F1
C C
IN=16 F1
C C
SET DUPLICATE INDICATOR TO ONE TO INDICATE THAT ONE END-OF-FILE F1
C C
HAS BEEN READ F1
C C
IFLAG=1 F1
C C
READ FILE F1

37
SUBROUTINE REXNER 14/74  DF=1

READ (14,4) IV31, IV54, IV55, IV56, IV57, IV58, IV59, IV60

IF (EOF(IN)) II=13

SET ORGDAT=EQUAL TO THE NUMBER OF ACTUAL DATA ENTRIES IN THE
MAIN DATA FILE

33 ORGDAT=1-1

RETURN CONTROL TO MAIN PROGRAM

RETURN

FORMAT (120,212,214,114,214)
END
SUBROUTINE CHANGE (I,J)

TYPE STATEMENTS

REAL LOG

INTEGER ORGAT
INTEGER BOOST,UPPER

COMMON BLOCKS


ENTRY *JA

SCH=IGATE(J)
IGATE(J)=0
IGATE(J)=SCH

SCR=BOOST(J)
BOOST(J)=SCR

UPPER(J)=SCR
UPPER(J)=SCR
UPPER(J)=SCR

SCR=SOU(J)
SOU(J)=SCR
SOU(J)=SCR

RES(J)=RE5(J)
RE5(J)=SCR

LOC(J)=LOC(J)
LOC(J)=SCR

PMOD(J)=PMOD(J)
PMOD(J)=SCR

DO 1 K=1,7
SCR=ST(J,K)
ST(J,K)=ST(J,K)
ST(J,K)=SCR

SCR=SIG(J)
SIG(J)=SIG(J)
SIG(J)=SCR

SCR=TYPE(B)
TYPE(J)=TYPE(J)
TYPE(J)=SCR
SUBROUTINE CHANGE  74*74  DATE:  FTP 4*6553  83/2

C      SCR=MULTL(A(I))
C      MULTL(A(I))=MULTL(I)
C      MULTL(I)=SCR
C
C      RETURN
C      END
SUBROUTINE VEHICLE
C
C TYPE STATEMENTS
C
INTEGER SUC,FAIL
INTEGER SMOSAT
INTEGER BIOUS,UPPER,UV,UL,UV2,UV+4,AL1
REAL LUG
DIMENSION V517
C
COMMON 3LOG2S
C
COMMON /DATALT-M,LOG2A,FMTLAT,LOG2S(126d),LOG2F(206d),UPPER(206d),L
C SOLO/LOG2A,FMTLAT,LOG2S(126d),LOG2F(206d),UPPER(206d),L
C1YPE(30d),MUTLI(30d),AL1
C
COMMON /DATAGUT-AVUV1,UV1,UV2,OVEN(20),OVEN(20),CONT(20)
C1UCON(2w),MUSUL,4TYPI(1)
C
COMMON /STORIF-SUC(150d),FAIL(150d),EXT(150d),HEAD(12)
C
BLANK THE HEADING ARRAY
C DO 1 I=1,12
C 1 HEAD(I)=10M
C
FILL THE HEADING ARRAY
C IF (UV<2d0 AND UV<2q0) HEAD(I)=SMALL LAUNCH
C IF (UV=2q0 AND UV<2q0) HEAD(I)=1OMVehicles
C IF (UV.2q0) HEAD(I)=SMALL LAUNCH
C IF (UV2.2q0) HEAD(I)=1OMVehicles
C IF (IV.2q0) HEAD(I)=SMALL LAUNCH
C IF (IV2.2q0) HEAD(I)=1OMVehicles
C IF (IV.2q0 AND IV<2q0) HEAD(I)=SMALL LAUNCH
C IF (IV2.2q0 AND IV<2q0) HEAD(I)=1OMVehicles
C IF (IV.2q0 AND IV<2q0) HEAD(I)=SMALL LAUNCH
C IF (IV2.2q0 AND IV<2q0) HEAD(I)=1OMVehicles
C IF (IV.2q0 AND IV<2q0) HEAD(I)=SMALL LAUNCH
C IF (IV2.2q0 AND IV<2q0) HEAD(I)=1OMVehicles
C IF (IV.2q0 AND IV<2q0) HEAD(I)=SMALL LAUNCH
C IF (IV2.2q0 AND IV<2q0) HEAD(I)=1OMVehicles
C ZERO INDEX FOR VEHICLE ARRAY
C
J=1
C TEST EACH ENTRY IN MAIN DATA FILE TO DETERMINE IF IT MEETS
C OUTPUT SPECIFICATIONS ON VEHICLE TYPE AND LAUNCH RESULT
C DO 6 I=1,LOG2A
C READ 12,17 JV+2,Jv2+2,JL,JL+7,VL,VL+7,XD,XD+2,VL+2
C 6 TEST FOR END OF FILE
C IF (20FL581) 6,2

41
SUBROUTINE VEHICLE 74/74 OPT*1                     FFN 4,04433  68/1

C TEST FOR VEHICLE TYPE TO BE OUTPUT
C IF $V * AND $U * ARE BOTH ZERO, OUTPUT ALL VEHICLES
C IF $V * OR $U * NOT EQUAL ZERO, THEN OUTPUT ONLY DESIRED
C VEHICLES

2 IF (IV$V = 0.0 AND IV$U = 0.0) GO TO 4
   IF (IV$V = 0.0 OR IV$U = 0.0) GO TO 3
   IF (IV$V = 0.0 AND IV$U = 0.0) GO TO 4
   GO TO 6
3 IF (IV$V = 0.0) GO TO 4
   IF (IV$U = 0.0) GO TO 4
   GO TO 6
C CHECK TO SEE IF ONLY FAILURES ARE DESIRED
C 4 IF (IV$V = 0.0 AND IV$U = 0.0) GO TO 6
   IF (IV$V = 0.0 OR IV$U = 0.0) GO TO 6
   IF (IV$V = 0.0 AND IV$U = 0.0) GO TO 6
C INCREMENT VEHICLE ARRAY INDEX AND FILL VEHICLE ARRAY
J = J + 1
IODATE(J) = IV$V
PSIZE(J) = IV$U
UPPER(J) = IV$V
PRUBIN(J) = IV$U
GO TO 7
5 ST(J, K) = IV$V
4RS(J) = IV$U
LOC(J) = IV$V
STG(J) = IV$U
LTYPE(J) = IV$V
AL = J
6 CONTINUE
C RETURN THE MAIN DATA FILE
C REMIND 8
C IF NO FAILURES HAVE BEEN FOUND TO FILL THE ARRAY A MESSAGE IS
C Printed AND CONTROL IS RETURNED TO THE MAIN PROGRAM
C 7 IF (IV$V = 0.0) GO TO 7
   PRINT 10
   RETURN
C THE VEHICLE ARRAY IS PRINTED AFTER SEPARATING THE DATE INTO
C MONTH/DAY/Year
7 DO 3 I = 1, ALN
   L1 = IODATE(J) / 1000
   L2 = IODATE(J) / 100
   L3 = IODATE(J) / 10
   IF (I MOD 11 = 0) GO TO 8
   WRITE (6, 10)
   IF (I = 0) WRITE (6, 11)
   IF (I = 70) WRITE (6, 12)
   IF (I = 71) WRITE (6, 12)

42
SUBROUTINE VEHICLE  74/74  03F#1  FHN 436433  097

IMPLICIT (I3,I4)

CALL (4,3) KEP

CALL (4,4)

8 WRITE (6,151) J=1,10,1101,K,103,1091,VEHICLESTAGE1,VEHICLEPLOT1,PROJECT1

1ST(JK),I=1,7,ATYP(L),TPH(J),RES(I),STC(I),LOC(I)

9 CONTINUE

RETURN

RETURN

FORMAT STATEMENTS

10 FORMAT (1H1)

11 FORMAT (1H1)

12 FORMAT (1H1)

13 FORMAT (1H1)

14 FORMAT (1H1)

15 FORMAT (1H1)

16 FORMAT (1H1)

17 FORMAT (1H1)

END
SUBROUTINE FAILIT 7:74  OPT=1.  FIN 4:04:13  08/1

SUBROUTINE FAILIT (NO)

C TYPE STATEMENTS

INTEGER ALN, UV, UV1
INTEGER SUCC, FAIL

COMMON BLOCK

COMMON /DATAOUT/ BV, UV, BV1, UV1, BV2, UV2, BVEH(20), IUEN(120), ACONT(20)

COMMON /STOR/(SUCC(1500), FAIL(1500), RAT(1500), HEAD(12))

ZERO THE SUCCESS AND FAILURE COUNTERS

DO 1 I=1,ALN
SUCC(I)=0
FAIL(I)=0

1 CONTINUE

DETERMINE THE NUMBER OFLaunches TO BE USED IN CALCULATIONS

NOI=NO•ALN/or 100.
IF (NOI.LE.0) GO TO 2
IF (NOI.LE.1) GO TO 5

THE NUMBER OF SuccESSES AND FailURES ARE CALCULATED

THESE MAY BE THE TOTALS TO THE LAUNCH BEING CONSIDERED OR ONLY
THE LAST NUP LAUNCHES
A FAILURE OF THE BOOSTER WHEN THE UPPER STAGE IS THE DESIRED
VELOCITY RESULTS IN A NO TEST CONDITIONS FOR THE UNDER- STAGE.

DG. 4 I=1,ALN
DG 3 J=1,NUL
IF (I+J.LE.ALN+1) GO TO 3
IF (STG(I).EQ.LH, AND BV.NE.X) GO TO 2
IF (STG(I).EQ.LH, AND UV.NE.U) GU 10 2
IF (STG(I).EQ.LH, AND BV.EQ.U AND UV.NE.X) GO TO 3

INCMENT THE SUCCESS COUNTER

SUCC(I+1)=SUCC(I)+1
GO TO 3

INCMENT THE FAILURE COUNTER

FAIL(I+1)=FAIL(I)+1
CONTINUE

AFTER EACH LAUNCH THE SUCCESS RATIO IS CALCULATED

RAI(L)=SUCC(I+1)/1.0/ SUM(SUCC(I)+FAIL(I))
SUBROUTINE FAILRT

74/74

OPI01

FTN 4266483

1.

IF (NO,LT,1) GO TO 4

1.

IF A RUNNING TALLY IS REQUESTED IT IS PRINTED AT THIS POINT

1.

IF (MOD(II-1,50).EQ.0) WRITE (55) HEAD, NO, ALN

1.

WRITE (6,61) RES(II), SIC(II), SUCCI(II), FAIL(II), RAT(II)

1.

CONTINUE

1.

IF (NO,GE,1) RETURN

1.

IF ONLY A SUMMARY IS REQUESTED IT IS PRINTED AT THIS POINT

1.

WRITE (6,7) NO, ALN, SUCCI(1N), FAIL(1N), RAT(1N)

1.

RETURN CONTROL TO THE MAIN PROGRAM

1.

RETURN

1.

FORMAT STATEMENTS

1.

5 FORMAT (4H1,4T17,E14.2K), 411.41, 41/11.41, 4H1ADJUSTED CUMU F

5.

LATIVE SUCCESS/FAILURE RATIO FOR LAST 14,254 FLIGHTS OF A TOTAL OF F

5.

2F, I4/15,14,13H, LAUNCH NUMBER, T27, 6H, RESULT, T35, 5HSIZE, T47, 6HTOTALS, F

5.

$55, 6H, PER CEN(147,140, T52,14F)

5.

6 FORMAT (T13, 14,14,14,14,14/14.14, 14,14,14,14, 14,14,14,F6.2)

5.

7 FORMAT (4H1, 6H1, 18,14, 14,14,14,14,14,14,14,14,14,14,14,14,14,14,14, F6.2, 14F)

5.

SUCCESS/FAILURE RATIO FOR LAST 14,254 FLIGHTS OF A TOTAL OF 14,719, 22 NUMBER OF SUCCESSES = 14,719, 22M F

5.

NUMBER OF FAILURES = 14,719, 22M, 22ABILITY = 14,719, 22M F

5.

END

5.

END
SUBROUTINE FAILLOC J=7/14 3PI=1

SUBROUTINE FAILLOC (NO)

INTEGER FL,FP,FA,FO,FALL,ALN,BV,W,SUCG
INTEGER BU1,BU2,UV1,UV2
REAL LRT

COMMON A-LONG
STP(2000),MULTL(2000),ALN

COMMON /NATAU/ BY,UV,BU1,BU2,UV1,UV2,STRU(2001),UV2H(2001),3CONT(2001)
COMMON /NATAU/ BY,UV,BU1,BU2,UV1,UV2,STRU(2001),UV2H(2001),3CONT(2001)

COMMON /NATAU/ BV,UV,BU1,BU2,UV1,UV2,STRU(2001),UV2H(2001),3CONT(2001)

ZERO THE FAILURE LOCATION COUNTERS

FL=0
FP=0
FA=0
FO=0

SET THE STARTING LAUNCH NUMBER

NO1=NO*ALN/LUG.3
IF (NO1.LT.1) NO1=ALM
N1=ALN=NO1+1

THE FAILURES DURING EACH PHASE ARE NOW CALCULATED
IF THE PHASE OF FAILURE IS UNKNOWN THAT LAUNCH IS NOT INLUCED

DO 2 I=1,ALN
IF (SIG11.EQ.143.AND.BV.NE.0) GO TO 1
IF (SIG11.EQ.144.AND.UV.NE.0) GO TO 1
IF (LGOAT.ALN-BV.EQ.0.AND.RE311.EQ.1M1) 10 TO 1
GO TO 2
1 IF (LGO11.EQ.14M) FP=FP+1
IF (LGO11.EQ.14M) ELF=ELF
IF (LGO11.EQ.14M) FA=FA+1
IF (LGO11.EQ.14M) FO=FO+1
2 CONTINUE

CALCULATE THE TOTAL NUMBER OF FAILURES
IF NO FAILURES ARE FOUND, RETURN TO THE MAIN PROGRAM
FAIL(1)=FP*ELF*FA*FU
IF (FAIL(1).LT.1) WRITE (6,4)
IF (FAIL(1).LE.1) RETURN

CALCULATE THE PERCENTAGE IF FAILURES OCCURRING DURING EACH PHASE
PRAT=FP*100./FAIL(1).

46
SUBROUTINE FAILLOG  74/74  CPT=1  FTN 4,5,433  09/1

          LKAT=FL*LOG.1./FAIL(1)  F
          ARAT=FA*LOG.1./FAIL(1)  F
          ORAT=FO*LOG.1./FAIL(1)  F
          PRINT THE RESULTS  F
          WRITE (6,51) LKAT, HKAT, MLN, PPK, PRAT, FL, LKAT, FA, AKAT, FO, ORAT, FAIL(1)  F
          RETURN CONTROL TO THE MAIN PROGRAM  F
          RETURN  F
          FORMAT STAT, RENFS  F
          END  F

1 FORMAT (H1, MT+1, TL+1, 2X, A10/T42, A12, 2X, A16+1/F4.47, ADJUSTED FAIL  F
LURES G3SS.0 BY LOCATION FOR LAST T4.23H FLIGHTS OF A TOTAL OF 3F,  F
2478/T12.84, LOCATION T10, 8M failures, T48, BMR-CENT/T14, SMBAL, T12, 3,  F
3149F6.2/T14, 4MLAND, T12, T14, F6.2/T14, DISCENT, T12, T14, F6.2/T1 F
4147/NOHIT/T14, T12, T14, F6.2/T14, SHOTAL, T31, T47, T14, 5M40.0  F
4 FORMAT (H4, T14, IFNO FAILURES HAVE BEEN REPORTED)  F
END  F
SUBROUTINE CURVIT (NU,C1,C2,C3,C4)

TYPE STATEMENTS

INTEGER C1,C2,C3,C4
INTEGER ALN,SUCF,FAIL

COMMON BLOCKS

COMMON /DAFIN/ NOFIN,NEWAT,LODATE,20001,ANNUAL(20001),UPPER(20001),
     1 QRUR(20001),PREJIO(20001),ST(20001),ST(20001),L
     1 STRAIGHT,MULT(20001),ALN

COMMON /STOR1/,SUC(15001),FAIL(15001),RAT(15001),HEAD(12)

COMMON /FIT/ X(14001),Y(14001),WF(14001),WGR(15001),THEOR(15001),NDATA
     1,RELERR,ABSERR,IFLAG,RA(161),R,UPPAR

EXTERNAL THEOR,DERIV1,DERIV2,DERIV3,DERIV4,DERIV5,DERIV6,DERIV7

DATA BLOCK SETS THE VALUES OF THE ERROR TEST FOR THE CURVE
FITTING SUBROUTINES

DATA RELERR=1.2-4,ABSERR=1.0-6/

DETERMINE THE NUMBER OF LAUNCHES TO BE CONSIDERED

N01=NO*ALN/158,3

FILL THE *X ARRAY WITH THE LAUNCH NUMBER [ADJUSTED TO THE NEA
START]

FILL THE *X ARRAY WITH THE ACQUIRED RELIABILITY

FILL THE *Y ARRAY WITH ONES SO ALL LAUNCHES ARE CONSIDERED

EQULALLY

DO I=1,N01
   1 CONTINUE

SET *NDATA = NUMBER OF LAUNCHES
SET *NPACK = NUMBER OF CURVE FIT PARAMETERS

NDATA=NU
NPACK=3

TEST FOR FIT ON Y = A * X + B * EXP (C * X)

48
CALL CURVE-FITTING SUBROUTINES
CALL FITIT (TH=0.4, DERIV1)
IF (IFLAG=0, IFLAG=0) GO TO 2
WRITE EQUATION WITH CALCULATED PARAMETERS
WRITE (6,6) PAR(1), PAR(2), PAR(3)
TEST FOR FIT ON Y = A * EXP(B * EXP(C * X))
2 IF (C2-NH-1) GO TO 3
SET INITIAL GUESS ON PARAMETERS
PAR(1)=1.0
PAR(2)=1.0
PAR(3)=1.0
IFLAG=0
CALL CURVE-FITTING SUBROUTINES
CALL FITIT (TH=0.4, DERIV1)
IF (IFLAG=0, IFLAG=0) GO TO 3
WRITE EQUATION WITH CALCULATED PARAMETERS
WRITE (6,6) PAR(1), PAR(2), PAR(3)
TEST FOR FIT ON Y = A * (1 - B / (C + X + 3))
3 IF (C3-NH-1) GO TO 4
SET INITIAL GUESS ON PARAMETERS
PAR(1)=1.0
PAR(2)=1.0
PAR(3)=1.0
IFLAG=0
CALL CURVE-FITTING SUBROUTINES
CALL FITIT (TH=0.4, DERIV1)
IF (IFLAG=0, IFLAG=0) GO TO 4
WRITE EQUATION WITH CALCULATED PARAMETERS
WRITE (6,6) PAR(1), PAR(2), PAR(3)
SUBROUTINE CURVE

1474 OPT=1

IFN 44433

<table>
<thead>
<tr>
<th>C</th>
<th>TEST FOR FIT U cưới V = A * EXP(3/2)</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>IF (LOG,LOG1 PER) GO TO 5</td>
<td>F1</td>
</tr>
<tr>
<td>C</td>
<td>SET INITIAL GUESS ON PARAMETERS</td>
<td>F1</td>
</tr>
<tr>
<td>C</td>
<td>PARAM(1)=0.1</td>
<td>F1</td>
</tr>
<tr>
<td>C</td>
<td>PARAM(2)=0.1</td>
<td>F1</td>
</tr>
<tr>
<td>C</td>
<td>IFLAG=1</td>
<td>F1</td>
</tr>
<tr>
<td>C</td>
<td>CALL CURVE FITTING SUBROUFINES</td>
<td>F1</td>
</tr>
<tr>
<td>C</td>
<td>CALL FIT(TH,DR,DERR)</td>
<td>F1</td>
</tr>
<tr>
<td>C</td>
<td>IF (IFLAG=0.1) GO TO 5</td>
<td>F1</td>
</tr>
<tr>
<td>C</td>
<td>WRITE EQUATION WITH CALCULATED PARAMETERS</td>
<td>F1</td>
</tr>
<tr>
<td>C</td>
<td>WRITE (6,9, PARAM(1), PARAM(2))</td>
<td>F1</td>
</tr>
<tr>
<td>C</td>
<td>CONTINUE</td>
<td>F1</td>
</tr>
<tr>
<td>C</td>
<td>RETURN CONTROL TO MAIN PROGRAM</td>
<td>F1</td>
</tr>
<tr>
<td>C</td>
<td>RETURN</td>
<td>F1</td>
</tr>
<tr>
<td>C</td>
<td>FORMAT STATEMENTS</td>
<td>F1</td>
</tr>
<tr>
<td>C</td>
<td>6 FORMAT (14H4,3H4) THE CURVE FIT FOR THIS DATA IS OF THE FORM 1/(10X,4)</td>
<td>F1</td>
</tr>
<tr>
<td>1 RELIABILITY = A * 3 X EXPIC * LAUNCH NUMBER/12X,3MM WITH THE PARA</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>2 PARAMETERS A, 3 AS FOLLOWING/15X,3HA = .18.10/15X,3HA = .18.10</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>3/15X,3HA = .18.10</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>7 FORMAT (14H4,3H4) THE CURVE FIT FOR THIS DATA IS OF THE FORM 1/(10X,4)</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>1 INRELIABILITY = A * 3 X EXPIC * LAUNCH NUMBER/12X,3MM WITH THE</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>2 PARAMETERS A, 3 AS FOLLOWING/15X,3HA = .18.10/15X,3HA = .18.10</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>3/15X,3HA = .18.10</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>8 FORMAT (14H4,3H4) THE CURVE FIT FOR THIS DATA IS OF THE FORM 1/(10X,4)</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>1 UNRELIABILITY = A * 3 X EXPIC * LAUNCH NUMBER/12X,3MM WITH THE</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>2 PARAMETERS A, 3 AS FOLLOWING/15X,3HA = .18.10/15X,3HA = .18.10</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>3/15X,3HA = .18.10</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>9 FORMAT (14H4,3H4) THE CURVE FIT FOR THIS DATA IS OF THE FORM 1/(10X,39</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>1 UNRELIABILITY = A * 3 X EXPIC * LAUNCH NUMBER/12X,3MM WITH THE</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>2 PARAMETERS A, 3 AS FOLLOWING/15X,3HA = .18.10/15X,3HA = .18.10</td>
<td>F1</td>
<td></td>
</tr>
<tr>
<td>3/15X,3HA = .18.10</td>
<td>F1</td>
<td></td>
</tr>
</tbody>
</table>
SUBROUTINE THEORY 74/74  OPT=1  FIN 4.54433  61

COMMON /F1(1,2000),F(1,2000),W(1,2000),THEO(200),NDATA F
L,MLEARN,MUSE,IFLAG,PAR(I),X,YIELD
GO TO 2 I=1,NDATA F

C CHECK FOR POSSIBLE EXPONENTIAL ARGUMENT OUT OF RANGE

IF (PAR(I)*X(I) .GT. 670.4950,AND,PAR(I)*X(I) .LE. 75) GO TO 1
WRITE (1,1)
IFLAG=1
RETURN
1 THEO(I)=PAK(I)*PAR(2)*EXP(PAR(3)*X(I))
2 CONTINUE
RETURN

C FORMAT (11H1,45E7H3,30H3,4F30.16)
LH=RELIABILITY = a + b * EXP(C * LAUNCH NUMBER)/W
END
SUBROUTINE DERIV1 74/7+ DPT=1 FTN 4.8+31 36/1

SUBROUTINE DERIV1 (L,PF,PPF)
DIMENSION PF(6), PPF(6,6)
COMMON (FIT, Y(1200), K(200), HF(1200), WORK(2500), THE3(1200), TVDATA)

L,RELVAR, ABSERR, IFLAG, PAR(6), R, NPAR

PF(1)=1,
PF(2)=X*PAR(1)*Y(1)
PF(3)=X(1)+PAR(2)*PF(2)
PF(4)=1.0
PF(5)=PAR(1)+PAR(2)
PF(6)=PAR(3)+PAR(2)
PF(7)=PAR(4)+PAR(2)
PF(8)=PAR(5)+PAR(2)
PF(9)=X(1)+PAR(2)
RETURN
END
<table>
<thead>
<tr>
<th>SUBROUTINE THEOR2</th>
<th>74/74</th>
<th>OPT=1</th>
<th>FIN 4.3433</th>
<th>84/2</th>
</tr>
</thead>
</table>

SUBROUTINE THEOR2  
COMMUN X(N(1000), X(N(2000), X(N(2500)), THEOR(2000), HUATA FI  
1.RETURN.ASERA.(FLAG = PAR), A, HPAR  
30 2 E=1, HUATA  
C CHECK FOR POSSIBLE EXPONENTIAL ARGUMENT OUT OF RANGE  
C IF (EXPAR(11)*EXPAR(11)*EXPAR(11)*EXPAR(11)*EXPAR(11) > 0) GO TO 1  
WRITE (3, 1)  
IFLAG = 13  
RETURN  
1.FM, HR(11)*EXPAR(11)*EXPAR(11)*EXPAR(11)*EXPAR(11)  
2.CONTINUE  
RETURN  
C 3 FORMAT (LM1, S) EXPONENTIAL ARGUMENT TOO LARGE FOR CURVE FITS/W/4. FI  
IFMNE = 165.1 = A * EXPB - EXP(A - LAUNCH NUMBER)*N/4. FI  
END  

SUBROUTINE DERIV2  I4,74  OPT=1
FIN 4,5,433  451

DIMENSION PF(6), PPF(6,6)
COMM. (FIT, X(1:2501), X(2:1001), W(1:2048), WORK(2501, 1ME3N2 12,3041, NO DATA)

A = EXP(PAR(3)*X(1))
B = EXP(PAR(2)*X(1))

PF(1) = PF(1)*PAR(1)
PF(2) = PF(2)*PAR(1)*X(1)+PAR(2)*A
PF(3) = PF(3)*PAR(1)*X(1)+PAR(2)*A
PF(1,1) = PF(2,1)/PAR(1)
PF(1,2) = PF(2,1)/B
PF(2,2) = PF(2,2)/B
PF(3,1) = PF(3,1)/PAR(1)
PF(3,2) = PF(3,2)/PAR(1)
PF(1,3) = PF(1,3)/PAR(1)
PF(2,3) = PF(2,3)/PAR(1)
PF(3,3) = PF(3,3)/PAR(1)

RETURN
END
SUBROUTINE THEOR3 74/74  OPT=1
                     FIN 4.6443  68/4

SUBROUTINE THEOR3
COMMON /FIT/ X(2346),Y(2346),HF(2346),HORK(2346),THEOR1(2346),NGAFA F
L,RERR,ABERR,IFLAG,PAR(6),R,PAR
DO 1 I=1,NGAFA     F
THEOR1 =PAR(1)*I*PAR(21)/(PAR(3)*I*PAR(21))      F
1 CONTINUE         F
RETURN             F
END

55
SUBROUTINE DcRIV3

DIMENSION PF(n), PPF(n,n)

F(1) = PAR(1)
PF(2) = PAR(1) * X(1) / A**2.0
PF(3) = PAR(1) * X(2) * X(1) / A**2.0
PF(4) = 0.0
PF(5) = PAR(1)
PF(6) = PAR(1) / PAR(1)
PF(7) = PAR(2)
PF(8) = PAR(2) / A
PF(9) = 2.0 * PAR(2) / A
PF(10) = PAR(1) * X(1) + PAR(3) / A
PF(11) = PAR(3)
PF(12) = PAR(3)
PF(13) = 2.0 * PAR(2)
PF(14) = 3.0 * X(1) * PAR(3) / A
RETURN
END
SUBROUTINE THEOR
COMMON /FIT/ X(2500),Y(2500),WF(2500),WORK(2500),THEOR(2500),N0DATA
DO 2 I=1,N0DATA
  IFLAG=0
  WRITE(6,*) X(I),Y(I)
  DO 5 I=1,2
    IFLAG=IFLAG+1
  5 CONTINUE
  RETURN
1  THEOR(I)=THEOR(I)*EXP(PAR(2)/X(I))
2  CONTINUE
3  FORMAT (1H1,4S,6X,13H*THEO~R*F~OR~U~L*C~E*F~IT~S*~X*,1)
4  END
SUBROUTINE DERIV4 IL,PF,PPF
DIMENSION PV(6), PFIL(6,5)
COMMON IPI(1), JI, JI2, L1, L2, L3, MR(200), MOK(256, 1), R, DQA, X, F, M1, M2, M3, M4
PV(1) = EXP(PAR(2)S*XI)
PV(2) = EXP(PAR(2)/X(11))
PPF(1,1) = 0.0
PPF(2,1) = PV(1)*X(11)
PPF(3,1) = PV(2)*X(11)
RETURN
END
SUBROUTINE FITIT (THEORY, DERIV)

C

AFNL SCIENTIFIC PROGRAM LIBRARY

C

AFNL IDENTIFICATION - FITIT

C

AFNL CONTAM - TECHNOLOGY DIVISION, THEORETICAL BRANCH

C

DATE ESTABLISHED IN LIBRARY

C

DATE OF LAST MODIFICATION

C

PROGRAMMER - CAPT. JAMES M. HEAD

C

USAF ACADEMY

C

COLORADO SPRINGS

C

MODIFIED FOR AFNL USE BY LT. HENRY J. HARPE

C

DESCRIPTION OF SUBROUTINE CALLING ARGUMENTS

C

NODATA... INPUT... NUMBER OF DATA POINTS IN THE X AND Y ARRAYS TO

C

BE FIT

C

NPAR... INPUT... NUMBER OF PARAMETERs IN THE PAR ARRAY

C

THEORY... EXTERNAL... SUBROUTINE WHICH DEFINES THE THEORETICAL FIT

C

VALUES AT THE GIVEN X VALUES

C

DERIV... EXTERNAL... SUBROUTINE WHICH DEFINES THE FIRST AND SECOND

C

PARTIAL DERIVATIVES OF THE FIT EXPRESSION WITH RESPECT

C

TO THE PARAMETERS AT EACH OF THE DATA POINTs.

C

... INPUT... ARRAY OF ABSOLUT DATA VALUES

C

... INPUT... ARRAY OF GRADUATE DATA VALUES

C

... INPUT... ARRAY OF POSITIVE WEIGHTING FACTORS FOR THE DATA.

C

ALFA... INPUT... RELATIVE ERROR TOLERANCE FOR PARAMETER CONVERGENCE

C

ABSFRA... INPUT... ABSOLUTE ERROR TOLERANCE FOR PARAMETER CONVERGENCE

C

IFLAG... INPUT/OUTPUT... ON INPUT, IFLAG = 1 WILL PRINT INTERMEDIATE

C

ITERATION VALUES AT THE SUBROUTINE CONVERGES, AS WELL AS

C

STATEMENTS GIVING THE REASONS FOR NON-CONVERGENCE. ON

C

OUTPUT, IFLAG... IS AN ERROR INDICATION FLAG.

C

= 0... CONVERGENCE... NORMAL RETURN.

C

= 1... CONVERGENCE... RESIDUALS ARE ZERO. POSSIBLE LIMITING

C

PRECISION EFFECTS.

C

= 2... SLIGHT CONVERGENCE... RESIDUAL IS GREATER THAN 0.9 TIMES

C

THE PREVIOUS RESIDUAL FOR NSGS ITERATIONS.

C

= 3... MAXIMUM NUMBER OF ITERATIONS EXCEEDED...

C

= 4... POSSIBLE DIVERGENCE... RESIDUAL HAS REMAINED LARGE.

C

= 5... LESS THAN 10 TIMES THE SMALLEST RESIDUAL FOR NSGS

C

ITERATIONS.

C

= 6... POSSIBLE DIVERGENCE... LARGEST SOLUTION INCREMENT

C

HAS INCREASED BY A FACTOR OF 10 FOR THE LAST

C

NSGS ITERATIONS.

C

= 7... POSSIBLE LOCAL MINIMUM. MAXIMUM NUMBER OF CUT STEP

C

ITERATIONS TAKEN.

C

= 8... MATRICE IS SINGULAR. THE FORMULATION OF THE THEORY

C

MIGHT DERIV SUBROUTES MAY BE INCORRECT.

C

PAR... INPUT/OUTPUT... ARRAY OF PARAMETER VALUES... ON INPUT, PAR

C

CONTAINS AN INITIAL ESTIMATE OF THE PARAMETER VALUES... ON

C

OUTPUT, PAR CONTAINS THE ACTUAL... BY THE

C

SUBROUTINE AS IT ITERATED.

C

RES... OUTPUT... RESIDUAL VALUE

C

WORK... SCRATCH... WORK ARRAY. WORK MUST BE DIMENSIONED AT LEAST

C

59

The fit is actually accomplished by subroutine FITIT. FITIT allocates virtual storage in the array WORK, and calls FITIT. This eliminates the need for a long call list and allows the number of data points and parameters to remain arbitrary.

FITIT accomplishes the fit by truncating the Taylor series for each fitting parameter about the initial approximation. After the quadratic term, and using this new value to replace the initial value, iterations this procedure until the number of significant digits desired is obtained, or until the maximum number of iterations allowed (INITMAX) is reached.

The basic code is explained in "FITIT, A PROGRAM TO LEAST-SQUARES FIT NON-LINEAR THEORIES" by James M. HECO. LIBRARY OF CONGRESS CATALOG NUMBER 4804.7052.

The user must supply two (2) subroutines named THEOR and DERIV. Theory must take the following form:

THEORY (NOATA, PAR, X, THEOR)

WHERE NOATA IS THE NUMBER OF DATA POINTS, PAR IS THE CURRENT VECTOR OF FITTING PARAMETERS, X IS THE ARRAY OF DATA ABSCISSAE, AND THEOR IS THE VECTOR OF PREDICTED THEORETICAL VALUES, THAT IS,

THEOR(I) = THEORY(X(I), PAR), I = 1, ..., NOATA

Deriv must take the following form:

DERIV (I, NPAR, X, PAR, PP, PPF)

WHERE I IS THE INDEX OF THE DATA POINT, NPAR IS THE NUMBER OF FITTING PARAMETERS, X AND PAR ARE AS IN THEORY, PPF IS THE MATRIX OF FIRST PARTIAL DERIVATIVES, AND PPF IS THE MATRIX OF SECOND PARTIAL DERIVATIVES.

PF(I,J) = THEORY(X(I), PAR(J)), J = 1, ..., NPAR

PF(I,J,K) = J - THEORY(X(I), PAR(K)), J = 1, ..., NPAR, K = 1, ..., NPAR

The scratch array WORK allocates virtual storage and must be dimensioned:

MATA(NPAR**2 + NPAR + NOATA)
COMMON /FIL/ X(L200), Z(L200), NF(200), NOA(K(200)), THEO(200), NOAF
C
1, HELEN = ABSK, IFLAG = PAR(1), R, HEPAR
C
EQUIV, NOC = (A, RES)
C
EXTERNAL THEORY, SERIV
C
SET MAXIMUMS
C
DATA NITMAX, 90, 14
DATA NCSIP, 14
DATA NCSIS, 14
DATA NSCS, 90
DATA NCGS/14/
DATA NGCS/14/
C
SET UP ACCOUNTING PROCEDURE
C
DATA KKKKKK/3/
IF (KKKKKK = EQ. 3) CALL REMARK (L14E = * FITI)
C
KKKKKK
C
CHECK INPUT PARAMETERS
C
IF (HAPAR.LT.1) GO TO 11
IF (HAPAR.GT.1000) GO TO 12
DO 1 J = 1, HAPAR
IF (MFI = EQ. 3) GO TO 13
DO 2 K = 1, HAPAR
IF (LJ = EQ. 1) GO TO 1
IF (LJ.LT.Q . K(LJ)) GO TO 14
1 CONTINUE
C
2 CONTINUE
IF (ABS KAPAR = EQ. 3) GO TO 15
IF (RECPAR.LT.1) GO TO 16
IF (ABS RAR = RECPAR.GE.3.0) GO TO 16
C
SET INDICES FOR VIRTUAL STORAGE ALLOCATION
C
NPAR2 = 30
IF (RF = 1) IF (DF = 6)
IF (LHWR = 1, PFH = NPAR2
IF (LHEC = 1, HC = NQDATA
IF (LOA = IA = NPAR2
G
SET IRR
C
IPMIN = SIGN(1, IFLAG)
C
FIT THE DATA
CALL FITI (TH, DAY, DERIV, NITMAX, NCSIP, NSCS, NGCS, NCGS, IPK, IH)
C
CHECK PRINT AND ERROR FLAGS
C
SUBROUTINE FINIT 74/74 OPT1 FTN 433 84/8

IF (IFLAG .EQ. 1) RETURN
IF (IFLAG .EQ. 7) GO TO 16
IF (IRPAP .NE. 1) GO TO 10
IF (IFLAG) GO TO 16,45,5,6,7,8,9, IFL

C
3 PRINT 18, IFLAG
GO TO 16
4 PRINT 17, IFLAG
GO TO 16
5 PRINT 26, NGST, IFLAG
GO TO 16
6 PRINT 27, NFMXK, IFLAG
GO TO 16
7 PRINT 28, NGST, IFLAG
GO TO 16
8 PRINT 29, NGST, IFLAG
GO TO 16
9 PRINT 26, NGST, IFLAG

C
10 CONTINUE
RETURN

C
11 PRINT 27, NPAR
GO TO 17
12 PRINT 28, NODATA, NPAR
GO TO 17
13 PRINT 29, I, AF(I)
GO TO 17
14 PRINT 27, l, l(l(I)
GO TO 17
15 PRINT 28, ABS, 44, RELERR
GO TO 17
16 PRINT 29
RETURN

C
17 IFLAG = 10

C
18 FORMAT (17,6X,H4M
CONGRATULATIONS. THE SUBROUTINE CONVERGED NORMALLY.
1 WITH IFLAG = [1L,1H] F
19 FORMAT (17,15H THE SUBROUTINE CONVERGED SUCH THAT ALL RESIDUALS ARE
1.0X10 .46. 80.0. THERE ARE POSSIBLE LIMITING PRECISION EFFECTS. IFLAG = 1F
2,1H] F
20 FORMAT (17,12H SUBROUTINE IS CONVERGING VERY SLOWLY. RESIDUAL IS
1.0X10 .46. 80.0. IFLAG = [1L,1H] F
21 FORMAT (17,17H SUBROUTINE HAS UNABLE TO CONVERGE IN 14.75M ITERATIONS.
1.0X10 .46. 80.0. RELAX. PERHAPS LESS STRINGENT ERROR TOLERANCES ARE REQUIRED. IFLAG = 1F
22,1L,1H] F
22 FORMAT (17,13A4
POSSIBLE DIVERGENCE. LARGEST SOLUTION INCREMENT IS
1.0X10 .46. 80.0. IFLAG = [1L,1H] F
23 FORMAT (17,15H LOCAL MINIMUM, 15,65H CUT STEP IERATIONS
1.0X10 .46. 80.0. IFLAG = [1L,1H] F
24 FORMAT (17,19A,16A,1H
TAKEN WITH NO SOLUTION IMPROVEMENT. IFLAG = [1L,1H] F
25 FORMAT (17,19A,16A,1H
COMPUTE A NONLINEAR LEAST SQUARES FIT, THE NUMBER
1.0X10 .46. 80.0. IFLAG = [1L,1H] F
15F DATA POINTS MUST EXCEED THE NUMBER OF PARAMETERS. IFLAG = 1H. AS INPUT F

62
SUBROUTINE FITIT  74/74 OPF=1  FITN 4.6+433  08/2

2. NDATA = (5.4LH AMI NPAR = (14.1LH)  
26 FORMAT (7SM) THE WEIGHTING FACTORS FOR THE DATA VALUES MUST BE POSI F) 
21.5 MSE. AS INPUT, W/(X.14+, 1H)  
27 FORMAT (51HNO TWO ABSISSA VALUES MAY BE EQUAL AS INPUT, Xi.14+, 1H) 
21.5 LN = Xi.14+, 1H) = (1PLG.1LH)  
28 FORMAT (150SM ERROR TOLERANCES RELERR AND ABSErr MUST BE NO;EHE F) 
21.5 QATIVE. IN ADDITION, AT LEAST ONE MUST BE POSITIVE.//1NH AS INPUT F) 
21.5 ABSErr = (1PLG.1LH) AND NRELERR = (1PLG.1LH)  
29 FORMAT (4SM THE DATA MATRIX IN SUBROUTINE FITIT IS SINGULAR.//124H F) 
21.5 POSSIBLE CAUSES ARE AN INCORRECT FORMULATION OF THE THEORY AND/OR F) 
21.5 ERROR SUBROUTINES OR A POOR INITIAL ESTIMATE FOR THE PARAMETERS.) F)

ENU
SUBROUTINE FIT - THEORY, DERIV, NMAX, NCSIP, NCSO, NCPS, NCSD, NCDS+1,FI

IPRINT

***************************************************************************

FITIT MERELY ALLOCATES VIRTUAL STORAGE FOR SUBROUTINE FITIT, FITIT

PERFORMS THE ACTUAL WORK.

EXPLANATION OF SUBROUTINE ARGUMENTS

NOAFT..... INPUT... NUMBER OF DATA POINTS IN THE X AND Y ARRAYS TO A FI

NPAR..... INPUT... NUMBER OF PARAMETERS IN THE PAR ARRAY.

THEORY..... EXTERNAL SUBROUTINE WHICH DEFINES THE THEORETICAL FIT.

VALUES OF THE GIVEN ABSCISSA (X) VALUES.

DERIV..... EXTERNAL SUBROUTINE WHICH DEFINES THE FIRST AND SECOND FI

PARTIAL DERIVATIVES OF THE FIT EXPRESSION WITH RESPECT TO...* FI

THE PARAMETERS AT EACH OF THE DATA POINTS.

X..... INPUT... ARRAY OF ABSISSA DATA VALUES.

F..... INPUT... ARRAY OF ORDIINATE DATA VALUES.

#F...... INPUT... ARRAY OF POSITIVE WEIGHTING FACTORS FOR THE DATA.

#ERRR..... INPUT... RELATIVE ERROR TOLERANCE FOR PARAMETER CONVERGENCE.

#ERRA..... INPUT... ABSOLUTE ERROR TOLERANCE FOR PARAMETER CONVERGENCE.

NITMX..... INPUT... MAXIMUM NUMBER OF ITERATIONS ALLOWED.

NCSIP..... INPUT... MAXIMUM NUMBER OF CONSECUTIVE CUT STEPS ALLOWED.

GUT-STEP ITERATION.

NCSO..... INPUT... MAXIMUM NUMBER OF CUT-STEP ITERATIONS ALLOWED PER... F.

ITERATION.

NCSO..... INPUT... MAXIMUM ALLOWABLE STEPS WHERE THE ITERATION IS SLOWLY

CONVERGENT.

NCSO..... INPUT... MAXIMUM ALLOWABLE CONSECUTIVE DIVERGENT STEPS.

NCSO..... INPUT... MAXIMUM NUMBER OF STEPS WHERE SOLUTION INCREMENT

MAY INCREASE BY A FACTOR OF 10 OVER THE PREVIOUS.

F.

F.

ITERATION.

IPRINT..... INPUT... PRINT FLAG. INTERMEDIATE RESULTS ARE PRINTED IF F.

IPRINT = 1

IPRINT = 0.

PAR..... INPUT... ARRAY OF PARAMETER VALUES. ON INPUT, PAR.

CONTAINS THE INITIAL ESTIMATE. ON OUTPUT, PAR CONTAINS THE... F.

LATEST MACHINE ITERATION RESULT.

R=SUM..... OUTPUT... NORM OF THE RESIDUALS AFTER COMPUTATION.

IFLAG=1, ERROR INDICATION FLAG.

CONVERGENCE, NORMAL RETURN.

CONVERGENCE, RESIDUALS ARE ZERO. POSSIBLE LIMITING PRECISION EFFECTS.

SLOW CONVERGENCE, RESIDUAL IS GREATER THAN 0.1 TIMES THE PREVIOUS RESIDUAL FOR NCDS ITERATIONS.

MAXIMUM NUMBER OF ITERATIONS EXCEEDED.

POSSIBLE DIVERGENCE, RESIDUAL HAS REMAINED LARGER THAN 10 TIMES THE SMALLEST RESIDUAL FOR NCDS ITERATIONS.

POSSIBLE DIVERGENCE, LARGEST SOLUTION INCREMENT NORM HAS INCREASED BY A FACTOR OF 10 FOR THE LAST NCDS ITERATIONS.

POSSIBLE LOCAL MINIMUM, MAXIMUM NUMBER OF CUT STEP ITERATIONS EXCEEDED.

A MATRIX IS SINGULAR. THE FORMULATION OF THE THEORY IS INCORRECT.

END
THE REST OF THE ARGUMENT ARE WORK ARRAYS.

** DIMENSION P(E), RRF(I, E), C(E), D04(E), A(E, E) **


** EQUIVALENCE (R, RES) **

** DATA URO 7 = 15 **

** DATA XMAX = 777777777777777777 **

** COMPUTE UNIT ROUND-OFF VALUE **

** LARGEST POSITIVE FLOATING-POINT NUMBER **

** SET INITIAL VALUES **

** R0 = ANAXL (RELAX, URO) **

** PRE = XMAX **

** IN = 0.0 **

** NCUT = 0 **

** E0 = 1 **

** E03 = 0 **

** M1 = COS * NCSIP **

** R0 = 0.0 **

** DO 1 = 1, NDATA **

** R = R0 * W(I) **

** 1 CONTINUE **

** R = 1.0/R **

** IF (IPRINT <= 1) PRINT 31 **

** **** BEGIN ITERATION **** **

** DO 23 = 1, NMAX **

** KON = 1 **

** M = M - 1 **

** BEGIN CUT STEP LOOP **

** JC = 0 **

** NCUA1 = NCSIP **

** RES = 0.0 **

** CALL THEORY **

** IF (IEFLAG = 0) RETURN **

** JR 2 = 1, NO_DATA **

** THEOR(I) = THEOR(I + 1) **

** RES = RES * THEOR(I) * THEOR(I + 1) **

** 2 CONTINUE **

** RES = RES * R **

** PRINT INTERMEDIATE SOLUTION VALUES; IF DESIRED **

** IF (IPRINT <= 1) GO TO 3 **

** NMAX = NCUA1 **

** PRINT 32, IMAX, VAL, RES (PARII), I = 1, NPAR **
SUBROUTINE FITI  7=7w  OPT=1  FTN 4.64+33  64/2

C   CHECK RESIDUAL AND ITERATION PARAMETERS
F1
C   3 CONTINUE
F1
IF (RES.EQ.0.0) GO TO 24
F1
IF (RES.LT.PREV) GO TO 8
F1
C   CHECK FOR POSSIBLE DIVERGENCE
F1
IF (RES.LT.11.*PRED) GO TO 4
F1
IF (IDJ.LT.N) GO TO 5
F1
GO TO 26
F1
C   4 CONTINUE
F1
IDJ:
F1
C   NEW PARAMETER VALUES DID NOT IMPROVE FIT. TAKE AVERAGE OF OLD AND
C   NEW PARAMETERS AND TRY AGAIN.
F1
C   5 CONTINUE
F1
GO = I-1.0*PREV
F1
PAR(I)=OLD(I)+PAR(I)*0.5
F1
C   END OF CUT STEP LOOP
F1
C   7 CONTINUE
F1
C   MAXIMUM NUMBER OF CONSECUTIVE CUT STEPS EXCEEDED. INCREMENT AND
F1
C   CHECK CUTOFF ITERATION COUNTER.
F1
C   8 CONTINUE
F1
IF (IDJ.LT.NCSS) GO TO 12
F1
GO TO 28
F1
C   TEST FOR SLOW CONVERGENCE
F1
C   9 CONTINUE
F1
IF (RES.LT.0.2*PREV) GO TO 3
F1
ISU=ISU+1
F1
IF (ISU.LT.NCSS) GO TO 10
F1
GO TO 25
F1
C   NEW VALUES IMPROVED FIT. RESET ABNORMAL TERMINATION COUNTERS AND
C   SAVE NEW VALUES
F1
C   10 CONTINUE
F1
ISU=0
F1
C   INITIALIZE WORK VARIABLES
F1

66
SUBROUTINE FITI   74/74   JPT=1   FTN 4,6443   08/2
C
12 CONTINUE
  DO 14 I=1,NPAR
  DO 14 J=1,NPAR
  A(I,J)=E*0
14 CONTINUE
C(I)=J
14 CONTINUE
C
SET UP VARIABLES TO CALCULATE TAYLOR SERIES EXPANSION
C
  DO 17 K=1,NOATA
  CALL BEFIV(K,PF,PRF)
  T=THOR(K)
  U=NF(K)
  G=T+U
  DO 15 J=1,NPAR
  F=PF(J)
  C(J)=G+C+G
  DO 15 I=1,NPAR
  C(I)=-RATN(J,RATN(J)+RE(J))
15 CONTINUE
17 CONTINUE

INVERT THE A MATRIX, CHECK FOR SINGULARITY
C
C
CALL INVAT(A,PF)
  IF (IFLAG EQ 7) 50 TO 19
C
CALCULATE NEW PARAMETER VALUES
C
  DO 19 I=1,NPAR
  Z=0.
  ZNORM=0.
  DO 19 J=1,NPAR
  C(J)=C(J)+Z
19 CONTINUE
  ZNORM=AMAX1(ZNORM,ABS(Z))
  RAN(I)=PAR(I)+Z
C
CHECK FOR CONVERGENCE
C
  IF (ABS(Z) GT .4*ABS*ABSLOLO(I1)*ABS*ABSRII) 20 TO 30
19 CONTINUE
  IF (KONV EQ 0) 30 TO 30
C
PROCEDURE (NVT=250) TO A SOLUTION
C
  IFLAG=0
  GO TO 30
C
NO CONVERGENCE, TEST FOR DIVERGENCE
C
20 CONTINUE
  IF (AML EQ 0) 30 TO 30
  IF (ZNOR I+11+245) 30 TO 30
  67
SUBROUTINE FIFI

102 = 102 - 1
IF (102 .LE. NO2S) GO TO 22
GO TO 27
C
21 CONTINUE
102 = 0
C
SET VARIABLES TO BE SAVED AND Iterate AGAIN
C
22 CONTINUE
PREVX = (PREVX - PREVX)
INC = INCH
C
23 CONTINUE
C
** END OF IterATION **
C
24 IF(FLAG < 0)
GO TO 34
C
SLOW CONVERGENCE, RESIDUAL IS GREATER THAN 9 TIMES THE PREVIOUS
RESIDUAL FOR NO2S Iterations
25 IF(FLAG = 2)
GO TO 30
C
POSSIBLE DIVERGENCE, RESIDUAL HAS REMAINED LARGER THAN 10 TIMES
THE SMALLEST RESIDUAL FOR NO2S Iterations
26 IF(FLAG = 6)
GO TO 36
C
POSSIBLE DIVERGENCE, LARGEST SOLUTION INCREMENT NORMALIZED
INCREASED BY A FACTOR OF 10 FOR THE LAST NO2S Iterations
27 IF(FLAG = 5)
GO TO 30
C
MAXIMUM NUMBER OF CONSECUTIVE OUT STEP Iterations TAKEN, POSSIBLE
LOCAL MINIMUM ENCOUNTERED
28 IF(FLAG = 6)
GO TO 36
C
MATRIX IS SINGULAR, THE FORMULATION OF THE THEORY AND/OR DERIV
SUBROUTINES MAY BE INCORRECT
29 IF(FLAG = 7)
C
30 CONTINUE
GO TO 42
C
31 CONTINUE
GO TO 41
C
RETURN
SUBKUTLINE FIT1

74/74  GAF=1

FTN 4,6+433  9.5.7

C

C

J2 FORMAT (.25, 1PE11.5, 1E11.10/12E11.10)

J3 FORMAT (.46NGET_RATE_CUT_RESIDUAL_PARAMETERS)

END
SUBROUTINE INVT (A, X1)
DIMENSION A(I, J), X1(I)
COMMON X1(1, J), X2, X1(1, J)
I = 0
DATA X1(I, J)
EQUIVALENCE (I, J)
INVERSE = I
DO 1 I = 1, NPAR
1 CONTINUE
K(I) = 0
IF (ABS(A(I, J)) .LT. 1.E-03) GO TO 5
DO 2 J = 1, NPAR
K(J) = K(I) + 1
2 CONTINUE
DO 3 J = 1, NPAR
TEMP = A(I, J)
4 K(J) = 1
3 CONTINUE
DO 4 K = 1, NPAR
RETURN
END
VARIABLES USED IN LAUNCH

ABSERR - Absolute error tolerance for parameter convergence
ALN - Number of data entries loaded in the vehicle array after processing
   by VEHICLE
ARAT - Percentage of failures occurring during ascent
BCONT - Name of booster vehicle contractor
BOOST - Booster designator
BV, BV1, BV2 - Booster designators for requesting output information
BVEH - Name of booster vehicle
C1, C2, C3, C4 - Designators for curve fit equations
DERIV1, DERIV2, DERIV3, DERIV4 - Subroutines defining first and second
   partial derivatives of the fit expression
   with respect to the parameters at each of
   the data points.

DERIV - Dummy name in FITIT and FITI to call desired partial derivative
   subroutine
FA - Ascent failure counter
FAIL - Total failure counter
FL - Land failure counter
FO - Orbital failure counter
FP - Pad failure counter
HEAD - Page heading array
ID1, ID2, ID3 - Launch date separated into month, day, year for printing
   output
IDATE - Launch date
IFLAG - Duplicate file indicator in RENMER; Error indication flag in CURVIT
IN - File designator
IO - File designator
KEY - Output request parameter indicating desired output subroutines
LATYP - Launch type nomenclature
LOC - Phase of failure
LRAT - Percentage of failures occurring over land
LTYP - Launch type designator
MULTLA - Multiple launch indicator
NDATA - Number of data points in the X and Y arrays to be fit
NEWDAT - Number of data entry cards read
NO - Percentage of launches to be considered in output subroutines
NO1 - Number of launches to be considered in output subroutines
NPAR - Number of parameters in the PAR array
ORAT - Percentage of failures occurring during orbital phase
ORGDAT - Number of data entries in updated main data file
PAR - Parameter values
PF - First partial derivative array
PPF - Second partial derivative array
PRAT - Percentage of failures occurring on the launch pad
PRODIR - Project director name
R - Residual value
RAT - Success ratio
RELERR - Relative error tolerance for parameter convergence
RES - Launch result
RESU - Launch result designator for use in output subroutines
SCR - Dummy variable used for switching data entries
SOUR - Source of information designator as indicated on data entry cards
ST - Source of information designator as stored in main data file
STG - Failed stage designator
SUCC - Success Counter
THEOR1, THEOR2, THEOR3, THEOR4 - Subroutines defining the theoretical fit values at given X values
THEORY - Dummy name in FITIT and FITI to call desired fit subroutine
UCONT - Name of upper stage contractor
UPPER - Upper stage designator
UV, UV1, UV2 - Upper stage designators for requesting output information
UVEH - Name of upper stage
WF - Weighting factors for curve fit data
WORK - Dummy work array
X - Abscissa data variables (launch number)
Y - Ordinate data variables (historical reliability)

All other variables are dummy variables.
### ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFETR</td>
<td>Air Force Eastern Test Range</td>
</tr>
<tr>
<td>AFWL</td>
<td>Air Force Weapons Laboratory</td>
</tr>
<tr>
<td>ANSI</td>
<td>American National Standards Institute, Inc.</td>
</tr>
<tr>
<td>ARPA</td>
<td>Advanced Research Projects Agency</td>
</tr>
<tr>
<td>DOD</td>
<td>Department of Defense</td>
</tr>
<tr>
<td>EOF</td>
<td>End-of-File</td>
</tr>
<tr>
<td>ERDA</td>
<td>Energy Research and Development Administration</td>
</tr>
<tr>
<td>INSRP</td>
<td>Interagency Nuclear Safety Review Panel</td>
</tr>
<tr>
<td>LES 8/9</td>
<td>Lincoln Experimental Satellites 8 and 9</td>
</tr>
<tr>
<td>MJS</td>
<td>Mariner Jupiter/Saturn</td>
</tr>
<tr>
<td>NASA</td>
<td>National Aeronautics and Space Administration</td>
</tr>
<tr>
<td>SER</td>
<td>Safety Evaluation Report</td>
</tr>
<tr>
<td>VAFB</td>
<td>Vandenberg Air Force Base</td>
</tr>
</tbody>
</table>
REFERENCES


2. Schefter, J., etc., TRW Space Log, TRW Systems Group, Redondo Beach, CA, various editions.


4. Vandenberg AFB Launch Summary, 1st Strategic Aerospace Division, Vandenberg Air Force Base, CA, 1 July 1975.
