NASA SP-6009(01)

OFFICE OF MANNED SPACE FLIGHT

FORECASTS AND APPRAISALS FOR MANAGEMENT EVALUATION

Volume 2

NASA

Prepared by
APOLLO
PROGRAM OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546

NASA SP-6009 (01)

APOLLO PROGRAM

OFFICE OF MANNED SPACE FLIGHT

FORECASTS AND APPRAISALS FOR MANAGEMENT EVALUATION

Volume 2

Prepared by

APOLLO PROGRAM OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – Price $4.50

FOREWORD

This document, though an official release of the Apollo Program Office, is furnished for information purposes only. Its purpose is to create awareness, stimulate interest and further promote understanding in the art and science of making real-life forecasts and their subsequent utilization in the control of space vehicle weight and performance throughout the Apollo Program.

This book is primarily intended for those in the Apollo Program who are responsible for the administration, design, development, manufacture, and test of the Apollo System. New theorems have been developed, as well as application of proven techniques but more importantly, a weight/performance forecasting methodology has been developed and automated. The text emphasizes the utilization of forecasting devices as applied to space vehicle weight and performance since these two parameters are of vital interest to all levels of management as well as technical personnel. Further, weight is tangible and readily measurable and can be readily related to performance.

The text provides, to those who wish to apply the developed methodology, all details necessary to do so and includes the mathematical development, computer program user's manuals and necessary instructions and procedures.

Forecasts and Appraisals for Management Evaluation text is intended to be a constructive aid to the NASA Apollo team in assisting them in the weight and performance area.

Samuel C. Phillips
Major General, USAF
Director, Apollo Program

TABLE OF CONTENTS

Paragraph	Title	Page
	APPENDIX A — WEIGHT/PERFORMANCE DATA	
A.1	INPUT DATA FLOW	A-1
A.1.1	MEASURING INDICES	A-1
A.1.2	WEIGHT/PERFORMANCE DATA FLOW	A-1
	APPENDIX B — PROBABLE ERROR RELATIONSHIP AND USE	
B.1	SELECTION OF MODEL	B-1
B.2	ESTABLISHING CONFIDENCE LIMIT LINES	B-1
B.3	VALUES OF PROBABLE ERROR	B-1
B.4	SECONDARY USE OF PROBABLE ERRORS AND DERIVATION	B-2
	APPENDIX C — MATH MODELS	
C.1	LINEAR MAXIMUM LIKELIHOOD	C-1
C.1.1	PREDICTION AND PREDICTION INTERVAL	C-5
C.1.2	APPENDIX	C-10
C.1.3	SAMPLE CALCULATIONS	C-12
C.2	THE EXPONENTIAL MODEL	C-24
C.2.1	METHOD I	C-30
C.2.2	METHOD II	C-35
C.2.3	PREDICTION AND PREDICTION INTERVAL	C-37
C.3	THE LOGISTICS MODEL	C-42
C.3.1	ESTIMATION OF PARAMETERS	C-42
C.3.2	PREDICTION AND PREDICTION INTERVAL	C-46
C.4	ADAPTIVE (FOURIER) EXPONENTIAL MODEL	C-53
C.4.1	FOURIER SMOOTHING	C-55
C.4.2	SEPARATION OF THE EXPONENTIAL CONTENT	C-57
C.4.3	ESTIMATION OF PARAMETERS AND PREDICTION	C-60
C.4.4	FOURIER ANALYSIS OF A FUNCTION CONTAINING AN EXPONENTIAL TERM	C-62

TABLE OF CONTENTS (Cont.)

Paragraph	Title	Page
C.4.5	LEAST-SQUARES APPROXIMATION OF AN EMPIRICAL FUNCTION BY ANOTHER FUNCTION	C-63
C.5	NOMENCLATURE	C-66
	APPENDIX D — PERFORMANCE RELATIONSHIPS	
D.1	INTRODUCTION	D-1
D.2	DERIVATION OF EQUATIONS	D-1
D.3	EXAMPLES OF GENERALIZED EQUATIONS TO DETERMINE TRADE-OFF FACTORS	D-7
D.4	EXAMPLES OF USE OF EQUATIONS TO OBTAIN TRADE-OFF FACTORS	D-8
D.4.1	EXAMPLE I	D-8
D.4.2	EXAMPLE II	D-13
	APPENDIX E — USER'S GUIDE	
	INTRODUCTION	1
	HISTORICAL BACKGROUND	1
	MARK II SYSTEM PHILOSOPHY	2
	MARK II COMPUTER SYSTEM	4
	Program Control	4
	Input Data Processing	4
	Trend Prediction	6
	Output Data Processing	6
	PURPOSE OF THE CURRENT PUBLICATION	7
	REVISIONS	7

	Reference No.
EXECUTIVE CONTROL (SPACE)	54S
CONTROL PROGRAM (GOODE)	55S
MAXIMUM LIKELIHOOD LINEAR	56S
MAXIMUM LIKELIHOOD NONLINEAR	57S
ADAPTIVE (FOURIER) EXPONENTIAL	58S

TABLE OF CONTENTS (Cont.)

Paragraph	Title	Page Reference No.
	ASYMPTOTIC (LOGISTIC) EXPONENTIAL	59S
	OUTPUT PROGRAM (OUTPUT)	60S
	HISTORY PLOT PROGRAM (HIS)	61S
	TITLING SUBROUTINE (TITLE)	62S
	SUMMING PROGRAM (SUM)	63S
	DECISION RELEVANCY (Criticality)	64S
	OVERALL COST ESTIMATION	65S
	UPDATE PROGRAM	66S
	SUBROUTINE GETDAT	67S
	NORMALIZATION SUBROUTINE (NORM12)	68S
	DOUBLE PRECISION MATRIX INVERTER (DPM1)	69S
	PROBABLE ERROR PROGRAM (RSS)	70S
	PLOTTING SUBROUTINE (UMPLOT)	71S

LIST OF ILLUSTRATIONS

Figure	Title	Page
A-1	Form for Typical Control Weight Information	A-5
A-2	Typical Data Submittal Requirements	A-6
A-3	Mass Properties Summary - Form 1 - Part I	A-7
A-4	Mass Properties Summary - Form 1 - Part II	A-8
A-5	Detail Mass Properties - Form 2 - Part I	A-9
A-6	Detail Mass Properties - Form 2 - Part II	A-10
A-7	Mass Properties Change Analysis - Form 3 - Part I	A 11
A-8	Mass Properties Change Analysis - Form 3 - Part II	A-12
A-9	Weight and Balance History Log - Form 4 - Part I	A-13
A-10	Weight and Balance History Log - Form 4 - Part II	A-14
A-11	Sequenced Mass Properties Data	A-15
A-12	Current Weight Summary - Saturn IB	A-16
A-13	Current Weight Summary - Saturn V	A-17
A-14	Current Weight Summary - Apollo Spacecraft	A-18
A-15	Current Weight Status	A-19
A-16	Current Change Analysis	A-20
A-17	Pending Change Analysis	A-21
A-18	Sequenced Mass Property Data	A-22
A-19	Performance Increase Shopping List	A-23
A-20	Propulsion Summary - Saturn V	A-24
A-21	Spacecraft Performance and Propulsion Summary	A-25
B-1	The Behavior of $\sigma_{\hat{w}_k}$ with Time	B-3
B-2		B-4
B-3		B-5
C-1	Maximum-Likelihood Linear Model Showing the Two Types of Error	C-7
C-2	Sample Computer Printout Spacecraft Weight (2 sheets)	C-25
C-3	Possible Combinations of Nonlinearity	C-59

APPENDIX A

WEIGHT/PERFORMANCE DATA

A.1 INPUT DATA FLOW

A.1.1 MEASURING INDICES

The establishment and documentation of control indices was achieved in the Apollo program through publication by the NASA Office of Manned Space Flight, of the Apollo Program Specification. This document establishes control values for vehicle weights and payload capabilities. Control weights were established separately for early Saturn IB and Saturn V mission vehicles and the operational vehicles. Each set of control values for the Saturn IB and Saturn V Launch Vehicle and the Block I and Block II Spacecraft are based upon the mission objectives and requirements presented in the Manned Space Flight Program Directive, Apollo Flight Mission Assignments. They include the typical information shown in Figure A-1.

A.1.2 WEIGHT/PERFORMANCE DATA FLOW

The second requirement, a periodic flow of current weight/performance status data, was met by the development of two documents, the Office of Manned Space Flight, Apollo Program Mass Properties Standard and the Office of Manned Space Flight Weight and Performance Data Submittal Requirements. The first of these two documents, the Mass Properties Standard was prepared for the Apollo Program primarily as a guide for contractor use in controlling and reporting mass property data to the Marshall Space Flight Center and the Manned Space Flight Center. It established a system for the management of mass properties in the design and use of space vehicles. The document was designed for three objectives:

a. To permit the organization of systematized, verifiable, and controllable mass properties of vehicle systems.

b. To facilitate rapid establishment and reporting of inputs for the weight/performance relationship.

c. To enable parametric extrapolation from reported systems to newly evolving systems.

Included in this standard, in addition to the specific definition of the many terms frequently used in discussion of mass properties and control are detail description and submittal frequencies for the type of mass property reports required throughout the various program phases. Illustrated in Figure A-2 are the reports required from functional studies through flight operations. The reporting formats to be utilized are shown in Figures A-3 through A-11.

In addition, the standard provides a functional code system for a three-generation breakdown of vehicle items according to their functional use. The objectives of the functional code are:

a. To provide a basis for computing weight summaries.
b. To allow direct substantiation of weight summaries and analysis methods.
c. To provide a uniform basis for design weight comparison of vehicle system.
d. To facilitate the preparation of weight summaries for complete vehicles, in such a way that a given section, stage, or module summary may be readily included in the summary of the total vehicle.
e. To provide identifiable vehicle coordinate location data.

The functional code, along with the nomenclature employed, are considered as the basic functional breakdown of vehicle items. The code consists of first-generation items, each of which is broken down into second-generation items. The second-generation items are further broken down into third-generation items. The first 16 first-generation codes include items which are essentially fixed in location and weight. The summary of these items is the dry weight of the particular stage or module being considered. The remaining first-generation code items include items which are variable either in location or weight. The summary of these items in a given stage or module configuration, as in flight sequence, is the variable weight and this weight when added to the dry weight <u>is</u> the total weight of the particular stage or module for the particular configuration. For example, this weight would be the total weight at lift-off, at a particular point in time, or at separation.

The second document, <u>Weight and Performance Data Submittal Requirements</u>, establishes the minimum requirements for uniform weight and performance data submittals. These inputs are to be supplied by the appropriate NASA Centers to the office of Manned Space Flight support of the Apollo Program Office Weight/Performance Management System for the surveillance and assessment of the Apollo Program Status.

This document, from a data flow point of view, actually completes the data flow cycle. The standard closed the data gap between the contractors and the NASA Centers, and the Submittal Requirements closes the data gap between the NASA Centers and the Office of Manned Space Flight.

The Weight and Performance Data Submittal Requirements document, like the standard, includes, for each data submittal requirement, a detail description and submittal schedule for the type data to be reported and the formats to be utilized. Each of these data submittal requirements is discussed below.

A.1.2.1 Weight and Associated Properties

These data are submitted monthly and include Weight Status, Change Analysis, and Sequence Mass Properties for each numbered launch vehicle and spacecraft combination. Useable propellants reported are based upon the specific defined mission requirements in accordance with the Office of Manned Space Flight Program Directive Apollo Flight Mission Assignments. Control weights are in accordance with the "Apollo Program Specification and Specification Weight," and are the weights specified in the contractor's statement of work or procurement specification.

For this segment of data, formats illustrated in Figures A-12 through A-18 are utilized.

The Current Weight Summary, Figures A-12, A-13, and A-14, summarize each reported launch vehicle and spacecraft from the stage and module level to launch vehicle capability and total spacecraft weight. The Current Weight Status Figure A-15, includes a weight breakdown to the functional system level for each launch vehicle stage and total launch vehicle as reported on formats shown in Figures A-12 and A-13 and for each spacecraft module and total spacecraft as reported on the format shown in Figure A-14. Figure A-15 (Format 1D) also includes program maturity data in the form of a percentage breakdown of current weight. This information is the percent of estimated, calculated, and actual weights that comprise the current weight. Weight changes that have taken place since the last report also are shown on the format for Current Weight Status.

The Current Change Analysis and Pending Change Analysis, Figures A-16 and A-17 (Formats 2A and 2B) which are referenced in the last column of the Current Weight Summary and Current Weight Status formats, require brief concise statements explaining the weight changes that have taken place since the last report.

Discussion concerning changes planned for incorporation at some future date, in the various launch vehicle stages and spacecraft modules, are also included.

The Sequenced Mass Property Data Figure A-18 (Format 3) includes weight, center of gravity, inertia, and time of occurrence data, and provides a weight account linked to space vehicle performance and to principal events throughout the flight profile. Each discrete weight-loss event is listed and identified along with time of occurrence. All weight is accounted for. Weight losses which occur over a period of time, such as full thrust propellants are summarized. Propellants carried as part of full thrust propellants, but not given thrust credit in the performance calculation, are clearly identified.

A.1.2.2 Vehicle Performance Increase Shopping List

The Performance Increase Shopping List, Figure A-19 (Format 4) provides, on a current basis, a list of selected items which, when initiated, will result in increased vehicle performance. Following initial submittal, the list is updated and submitted to the Office of Manned Space Flight as additional information becomes available.

A.1.2.3 Performance Analysis Data

These data are submitted monthly and include propulsion and velocity capability data for each numbered launch vehicle and spacecraft combination. For this segment of data, Figures A-20 and A-21 (Formats 5B and 5C) are utilized.

The Propulsion Summary, Figure A-20 (Format B) summarizes propulsion parameters used in the determination of launch vehicle payload, as reported under segment A for the Saturn IB and Saturn V launch vehicles. The Performance and Propulsion Summary, Figure A-21 (Format 5C) summarizes both performance and propulsion parameters used in the determination of Block I and Block II spacecraft propellant loading and gives total weight as reported for the Apollo spacecraft.

Control Point	Item	Control Weight (Pound)			
		201	202	203	204
S-IB	Dry Weight				
	Propellant Tank Capacity				
	Separation Weight				
S-IB/S-IVB Interstage	Total Weight				
S-IVB	Dry Weight				
	Propellant Tank Capacity				
	Separation Weight				
Instrument Unit	Total Weight				
Launch Vehicle	Payload Capability				

Figure A-1. Form for Typical Control Weight Information

D = DETAIL
S = STATUS
P = PROCEDURAL
M = MISCELLANEOUS

Report Element	Tasks (paragraph)	Page	Report Elements (paragraph)	Page	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Program Phase					Funded Studies (any phase)		Definition		Acquisition												
Report Type					D	S	D	S	D	D	D	D	S	S	S	S	S	S	P	M	M
Schedule of Submittals					Semiannually for studies of 1 year or more: at completion of all studies	Every other week for studies of 2 months or less: at first of each month for studies of more than 2 months	Semiannually for studies of 1 year or more: at completion of all studies: with submittal of all proposals	Every other week for studies of 2 months or less: at first of each month for studies more than 2 months (to be continued through contract definition)	At PDR, or within 30 days of approved definitive contract (or Go-Ahead) (FOR APPROVAL)	Annually for contracts exceeding 2 years	At CDR, or within 30 days after 90% completion of original drawing release for contracts of less than 2 years	At FACI, or within 30 days after weighings, or flights, specified by the procuring activity	Design - first of each month after approved definitive contract (or Go-Ahead) and ending with delivery of last item	Pre-flight - 120 days prior to launch	Pre-flight - 14 days after acceptance weighing	Pre-flight - 14 days after significant mass properties change	Pre-flight - Quick Look - within 24 hours after launch	Post-flight - Final - 30 days after launch	Verification Plan - within 6 months after definitive contract (or Go-Ahead) (FOR APPROVAL)	Contract Change Proposal - with each contract change proposal	Fluid and Propellant Verification - as required by Verification Plan
Title Page			3.4.3.1	7	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Table of Contents			3.4.3.2	8	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Introduction			3.4.3.3	8	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Contractor Mass Properties Organization	3.1	3	3.4.3.4	8			x		x	x	x										
Mass Properties Summary	3.4.2.1.4	7	3.4.3.5	8	x	x	x	x	x	x	x	x	x	x	x	x		x			
Detail Mass Properties	3.4.2.1.4	7	3.4.3.6	9	x		x		x	x	x	x									
Mass Properties Change Analysis	3.3.2.1, 3.3.2.2	4	3.4.3.7.1	9	x	x	x	x	x	x	x	x	x	x	x	x		x			
Pending Mass Properties Change Analysis	3.3.2.1, 3.3.2.2, 3.3.3.4	4 6	3.4.3.7.2	10		x		x					x	x	x	x				x	
Summary of Improvement Potentials	3.3.3.1, 3.3.3.2	6	3.4.3.7.3	10		x		x					x								
Unresolved Problems	3.3.3.3	6	3.4.3.8	11		x		x					x	x	x	x	x	x			
Mass Properties History Log	3.3.2.7	5	3.4.3.9	11								x		x	x	x		x			
Sequenced Mass Properties Data	3.4.2.1.4	7	3.4.3.10	11	x	x	x	x	x	x	x	x	x	x	x	x	x	x			
Powered Flight Mass Properties	3.4.2.1.4	7	3.4.3.11	12	x		x		x	x	x	x									
Diagrams	3.4.2.1.4	7	3.4.3.12	12	x	x	x	x	x	x	x	x	x	x[1]	x[1]	x[1]		x[1]			
Mass Properties Substantiating Plan	3.2.1	3	3.4.3.13	13	x		x		x											x	
Government Furnished Equipment	3.4.2.1.5	7	3.4.3.14	13	x		x		x	x	x	x	+	+	+	+		x			
Computer Cards or Tapes	3.4.2.1.4	7	3.4.3.15	13					(x)	(x)	(x)	(x)	(x)								
Mass Distribution	3.4.2.1.4	7	3.4.3.16	13	x		x		x	x	x	x									
Mass Properties Dependent Design Information	3.4.2.1.5	7	3.4.3.17.1	13	x		x		x	x	x	x	+								
Structural Increments for Design Features	3.4.2.1.5	7	3.4.3.17.2	13	x		x		x	x		x									
Records for all Measurements	3.3.2.8	5	3.4.3.18.1	14								x			x			x			x
Records for Dry Mass Properties Measurements	3.3.2.7	5	3.4.3.18.2	14								x			x						
Records for Fluid Mass Properties Measurements	3.3.2.7, 3.3.2.9	5	3.4.3.18.3	14													x	x			x
Summary of Critical Mass Properties	3.4.2.1.4	7	3.4.3.19	14		x		x					x	x	x	x	x	x			
Evaluation of Flight	3.3.2.9	5	3.4.3.20	14													x	x			
Critical Mass Properties Uncertainties	3.3.2.3	4	3.4.3.21	15	x		x		x	x	x	x	+								
Mass Properties Limits	3.3.2.4	4	3.4.3.22	15	x		x		x	x	x	x	+								
Parameters and Inventory of Fluids and Propellants Loaded	3.3.2.1	4	3.4.3.23	15	x	x	x	x	x	x	x	x	x	x	x	x	x	x			
Capacity and Loading Information for Fluids and Propellants	3.3.2.1	4	3.4.3.24	15	x		x		x	x	x	x	+								
Mass Properties Verification Plan	3.3.2.5	4	3.4.3.25	16									+						x		

NOTES: 1 less graphs
+ update only - submit changes to the data, information, or analyses contained in the last published report of these items
(x) submitted as required by procuring activity

Figure A-2. Typical Data Submittal Requirements

Mass Properties Summary										
Code	Description	Specified Weight Base ( )	Procuring Activity and GFE Changes	Revised Specified Weight Base	Current Weight	Changes Last to Current	Percentage Breakdown of Current Weight			Note Number
							Est.	Calc.	Act	

Prepared by ____________ Identification Report No. ____________

Date ____________ Page ______ of ______

Form 1 - Part I

Figure A-3.

Mass Properties Summary

Code	Description	Current Weight ( )	Center of Gravity (__ from Referenced Datum)			Moment of Inertia (______)		
			X	Y	Z	Pitch	Roll	Yaw

Prepared by __________

Date _____________

Form 1 - Part II

Identification

Report No. ________

Page _____ of _____

Figure A-4.

Detail Mass Properties

Code	Description	Class (E, C, A)	Weight (______) Code Generation			Center of Gravity (__ from Referenced Datum)		
			First	Second	Third	X	Y	Z

Prepared by __________ Identification Report No. ________

Date __________ Page _____ of _____

Form 2 - Part I

Figure A-5.

Detail Mass Properties								
Code	Description	Current Weight ( ) Code Generation			Moment of Inertia ( )			Note Number
		First	Second	Third	Pitch	Roll	Yaw	

Prepared by ____________

Date ______________

Form 2 - Part II

Identification

Report No. __________

Page ______ of ______

Figure A-6.

Mass Properties Change Analysis

*Change Identification and/or *Note Number	Code	Effective Point	Weight Change ( )			Center of Gravity (__________ from Referenced Datum)			Moment of Inertia (__________)		
			Total	Responsibility							
				Design Activity	Procuring Activity	X	Y	Z	Pitch	Roll	Yaw

Prepared by ____________ Identification Report No. ____________

Date ____________ Page ______ of ______

Form 3 - Part I

*Cross out if not applicable.

Figure A-7.

Mass Properties Change Analysis

*Change Identification and/or *Note Number	Code	Total Weight Change ( )	Remarks

Prepared by ____________ Identification Report No. ____________

Date ____________ Page ______ of ______

Form 3 - Part II

*Cross out if not applicable.

Figure A-8.

Weight and Balance History Log

Sheet No. ____

Place an X in blanks not required for identification

Item

Space Vehicle ____________ S/C Module ____________ Engine Model ____________

Mission ____________ L/V Stage ____________ Serial No. ____________

Entry No.	Date / Time	Signature	Item Description (Drawing No., Serial No., Name, System)	Code	+ / -	Weight ( )	Center of Gravity (__ from Referenced Datum)		
							X	Y	Z

Form 4 - Part I

Figure A-9.

Weight and Balance History Log

Sheet No. ____

Work Sheet

Place an X in blanks not required for identification

Item

Space Vehicle ____________ S/C Module ____________ Engine Model ____________

Mission ____________ L/V Stage ____________ Serial No. ____________

Entry No.	+ / -	Weight ( )	Center of Gravity			Product of Inertia			Transfer Factors			Moment of Inertia ( )			Moment of Inertia ( )		
			X	Y	Z	WXY	WXZ	WYZ	X^2	Y^2	Z^2	Ioy	Iox	Ioz	Pitch	Roll	Yaw

Form 4 - Part II

Figure A-10.

Sequenced Mass Properties Data									
Code	Description	Current Weight	Center of Gravity (__ from Referenced Datum)			Moment of Inertia ( )			Note Number
		( )	X	Y	Z	Pitch	Roll	Yaw	

Prepared by ______ Identification Report No. ______

Date ______ Page ____ of ____

Form 5

Figure A-11.

Vehicle No. ________ Date ________

Mission Type ________

Saturn IB

Current Weight Summary

Control Point	Item	Control Weight	Current Specification Weight	Current Weight	Comments and/or Reference to Change Analysis
S-IB	Dry Weight				
	Separation Weight				
	Useable Propellant				
	Propellant Tank Capacity				
S-IB/S-IVB Interstage	Total Weight				
S-IVB	Dry Weight				
	Separation Weight				
	Useable Propellant				
	Propellant Tank Capacity				
Instrument Unit	Total Weight				
Launch Vehicle	Liftoff Weight				
	Payload Capability	*			

*Includes ____ pounds parking orbit loss.

Figure A-12.

Vehicle No. ________ Date ________

Mission Type ________

Saturn V

Current Weight Summary

Control Point	Item	Control Weight	Current Specification Weight	Current Weight	Comments and/or Reference to Change Analysis
S-IC	Dry Weight				
	Separation Weight				
	Useable Propellant				
	Propellant Tank Capacity				
S-IC/S-II Interstage	Total Weight				
S-II	Dry Weight				
	Separation Weight				
	Useable Propellant				
	Propellant Tank Capacity				
S-II/S-IVB Interstage	Total Weight				
S-IVB	Dry Weight				
	Separation Weight				
	Useable Propellant*				
	Propellant Tank Capacity				
Instrument Unit	Total Weight				
Launch Vehicle	Liftoff Weight				
	Payload Capability				

*Includes ____ pounds parking orbit loss.
**Guaranteed.

Figure A-13.

Vehicle No. ________ Date ________

Mission Type ________

Apollo Spacecraft

Current Weight Summary

Control Point	Item	Control Weight	Current Specification Weight	Current Weight	Comments and/or Reference to Change Analysis
Command Module	Total Weight (Including Crew)				
Service Module	Dry Weight				
	Injected Inert Weight				
	Useable Propellant				
	Propellant Tank Capacity				
LEM Ascent Stage (Exclude Crew)	Dry Weight				
	Injected Inert Weight				
	Usable Propellant				
	Propellant Tank Capacity				
LEM Descent Stage	Dry Weight				
	Injected Inert Weight				
	Useable Propellant				
	Propellant Tank Capacity				
Adapter	Total Weight				
Launch Escape System	Total Weight				
Spacecraft	Liftoff Weight				
	Injected Inert Weight				

Figure A-14.

Current Weight Status

Vehicle No. ______________ Mission Type ______________ Date ______________

Control Point	Item and Description	Current Weight	Change Last to Current	Percentage Breakdown of Current Weight			Reference to Change Analysis
				Estimated	Calculated	Actual	

Format 1D

Figure A-15.

Current Change Analysis

Vehicle No. ________ Date ________

Note Number*	Item and Description	Change Last to Current	Remarks

*Reference to formats 1A, B, C, and D as applicable.

Format 2A

Figure A-16.

Pending Change Analysis*

Vehicle No. ______________ Date ______________

Item and Description	Estimated Change	Effectivity	Remarks

*To include only those changes based on firm design decisions as applicable to specific vehicles.

Format 2B

Figure A-17.

Sequenced Mass Property Data

Vehicle No. ____________ Date ____________

Item and Description	Weight (Pounds)	Center of Gravity (Increment from Reference Datum)						Time of Occurrence	Notes
		X	Y	Z	Pitch	Roll	Yaw		

Format 3

Figure A-18.

Performance Increase Shopping List			
1.	Index Number	-	An identification numbering sequence prefixed by NASA Center initials, e.g., MSC-1, MSC-2, MSC-3, etc.
2.	Description	-	Short title and concise description. Relate to control point where possible.
3.	Payload/Weight Change	-	Predicted payload increase, weight change, and/or other vehicle property change.
4.	Status	-	Provide dates, schedules, possible or actual vehicle effectivity for the following categories of change: proposed, approved - still pending, incorporated.
5.	Cost Effect	-	Indicate the dollar value per pound of payload increase with point of effectivity.
6.	Influence on Reliability	-	Evaluate, insofar as possible, the predicted effect of change on mission success, and/or other reliability factors.
7.	Schedule Impact	-	Indicate the schedule slippage, if any, due to implementation and indicate effectivity (vehicle number) desired and expected.
8.	Originator	-	Name, code, and initiation date.
9.	Cognizant Office and Responsible Personnel	-	Office code, location, and name.
10.	Remarks	-	Explanatory notes, substantiation, and references.

Format 4

Figure A-19.

Vehicle No. ____________ Date ____________

Mission Type ____________

Saturn V

Propulsion Summary

Propulsion Data	Stage		
	S-IC	S-II	S-IVB
Engine Model	F-1	J-2	J-2
Engine Type	RP-1/LOX	LH_2/LOX	LH_2/LOX
Thrust (Pounds)/Engine	(Nominal)	(Nominal)	(Nominal)
Control Weight*			
Current Performance**			
Specific Impulse (I_{sp}, Sec.)	(Nominal)	(Nominal)	(Nominal)
Control Weight*			
Current Performance**			
Oxidizer/Fuel Ratio			
Control Weight*			
Current Performance**			

*As related to Control Weight value of Format 1B.
**As related to Current Weight.

Format 5B

Figure A-20.

Vehicle No. ____________ Date ________

Mission Type ____________

Spacecraft Performance and Propulsion Summary

Propulsion Data	Stage or Module		
	Service	LEM Descent	LEM Ascent
Engine Model			
Engine Type			
Thrust (Pounds)/Engine	(Nominal)	(Nominal)	(Nominal)
Control Weight*			
Current Performance**			
Specific Impulse (I_{sp}, Sec.)	(Nominal)	(Nominal)	(Nominal)
Control Weight*			
Current Performance**			
Oxidizer/Fuel Ratio			
Control Weight*			
Current Performance**			
Velocity Capability (Feet per second)***			
Control Weight*			
Current Performance**			

*As related to Control Weight value of Format 1C.
**As related to Current Weight.
***To be completed in accordance with the discrete velocity increments (set forth in the Apollo System Specification) used in computing the weights of Format 1C.

Format 5C

Figure A-21.

APPENDIX B

PROBABLE ERROR RELATIONSHIP AND USE

B.1 SELECTION OF MODEL

In any forecasting process there is some degree of uncertainty associated with predicted numbers. This degree of uncertainty many result from errors in selecting an inadequate model or from the tolerance expected in measurement or the displacement from the mean due to random changes up until ship date or a host of other reasons. But the fact remains that forecast analysis is based on the production of the best available quantitative values. This in turn necessitates selection of a single, most-likely weight at the specific date, and this is taken as the mean expected weight.

B.2 ESTABLISHING CONFIDENCE LIMIT LINES

For purposes of illustrating the range of the scatter of potential weights at ship date, the upper and lower 95 percent confidence limit lines are calculated. In this instance, the upper confidence limit is defined as a "one tailed" limit below which, for a large sample, 95 percent of the points would fall. That is, for a large sample, 5 percent of the data could be expected to exceed the upper confidence limit value.

Confidence limit lines are readily presented in graphic form but difficult to visualize in tabular form; hence, the term probable error is introduced as follows:

> Probable error is the numerical difference between the upper (95 percent) confidence limit and forecasted weight at the shipping date.

Probable error is then tabulated adjacent to forecasted values in the detail result tables so that the reader can readily grasp the magnitude of uncertainty of any forecasted value.

This then is the primary use of probable error – a ready index of the prediction accuracy.

B.3 VALUES OF PROBABLE ERROR

The values of probable error are obtained directly from the forecast and analysis of the functional system. For stages and modules and for over-all spacecraft and launch vehicles, the probable errors are then calculated using appropriate factors and a simple root-sum-squared process. The resultant probable errors are only approximate, but sufficient to give an indication of expected range of the final weights about the predicted weight.

B.4 SECONDARY USE OF PROBABLE ERRORS AND DERIVATION

A secondary use of probable errors is for assistance in forecasting launch vehicle capability dispersion and space weight tolerance at ship date. In functional system weight forecasted errors, there are numerous factors which contribute to these errors. In a general sense, these kinds of errors can be classed into three categories:

a. Errors represented by a variation of reported weight about the "true" weight, which are in turn a function of estimation processes, calculation accuracies, or actual measuring equipment precision. These errors can be expressed as e_m in Figure B-1 with standard deviation, σ_m.

b. Variation of true weight about its "true" expected growth line on a more or less random pattern due to the myriad of random forces at work. This variation, approximated by e_R with standard deviation σ_R, is associated with the random changes in weight being made right up until launch date.

c. Error of the forecasted line, calculated from the limited available sample, to exactly match the "true" trend line. This error is expressed as $e_{\hat{w}_k}$ in Figure B-1 with standard deviation $\sigma_{\hat{w}_k}$.

For initial mathematical approximations, all three of these errors are assumed to be normally distributed with zero mean. Further, the first two kinds of error are indistinguishable and are represented here by

$$\sigma_k^2 = \sigma_m^2 + \sigma_R^2$$

where σ_k is the assumed standard deviation of the reported data bout the "true" but unknown trend line, indicated as e_k in Figure B-1.

In the early maximum likelihood models, σ_k is assumed to decrease in some relation to program progress, as indicated by changes from estimated to calculated to actual measurements. In the larger sense, these E/C/A values may be considered as a kind of program maturity factor. The expected value of σ_k is therefore assumed to decrease in some fashion, represented schematically in Figure B-2, and approaches a final value as the discrete parts of the functional system move from estimated to calculated to actual status, and as measurement accuracies increase due to installation of better facilities.

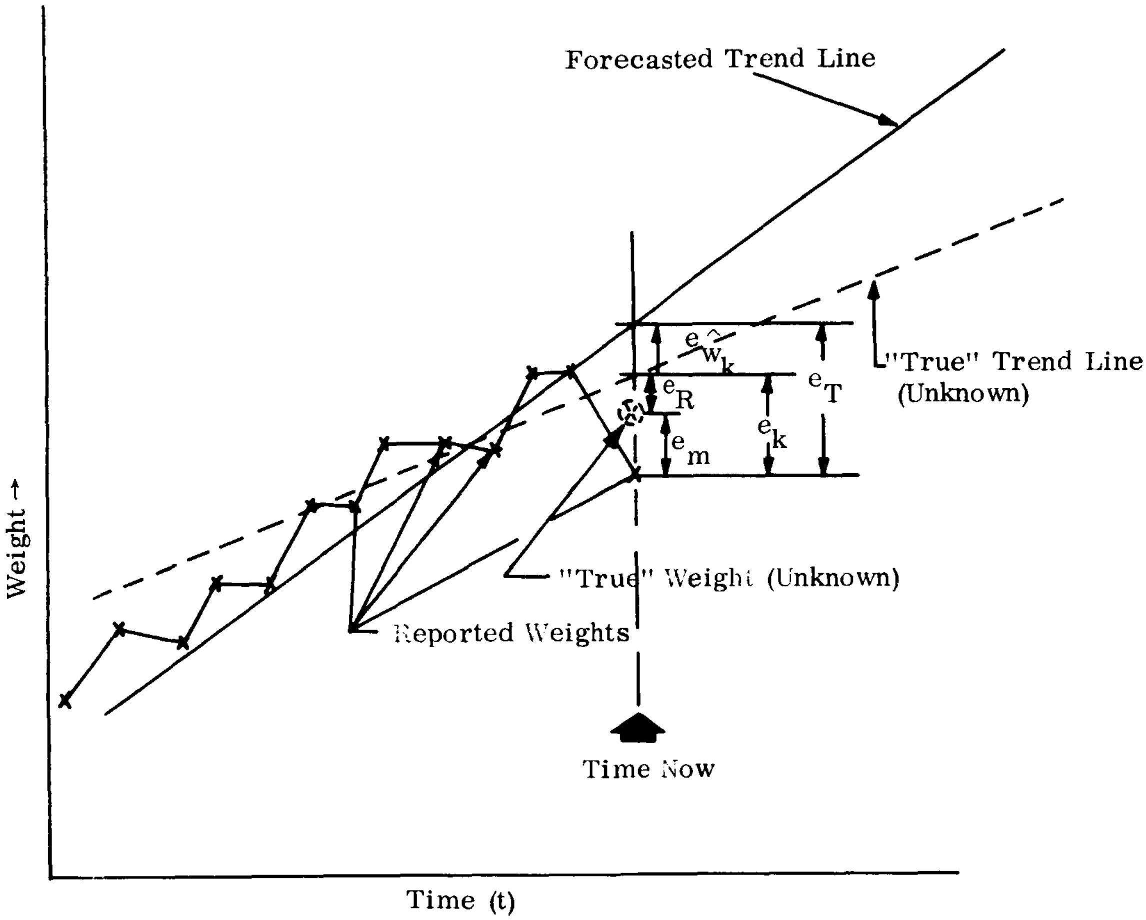


Figure B-1

The behavior of $\sigma_{\hat{w}_k}$ with time is dependent on the nature of the fundamental model assumed. For the linear case, $\sigma_{\hat{w}_k}$ increases rather linearly with time from current to ship; for the nonlinear model, increases in a nonlinear fashion, as illustrated in Figure B-2.

The total variation of the reported weight about the forecasted line is designated as e_T with standard deviation σ_T found by

$$\sigma_T^2 = \sigma_k^2 + \sigma_{\hat{w}_k}^2$$

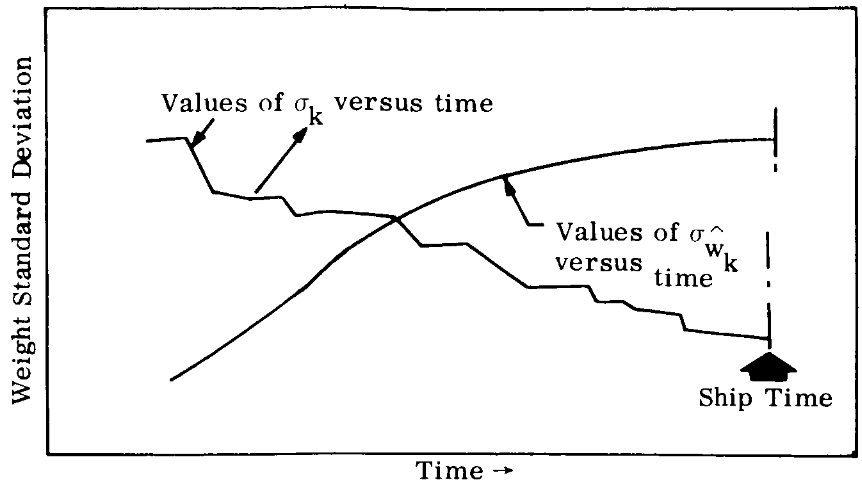


Figure B-2

The expected value of σ_T will therefore vary in a manner predetermined by the combined variations of σ_k and σ_R with time, and can readily take a shape such as indicated by any of the curves A, B, C, or D shown in Figure B-3. Curve A represents the case where σ_k is decreasing slowly and $\sigma_{\hat{w}_k}$ increasing rapidly; curve D is the opposite extreme with rapid decrease in σ_k and slow rise in $\sigma_{\hat{w}_k}$. Or, from the program control viewpoint, curve A represents the case where the variations of weight about the mean line are not expected to damp out significantly as the program matures while the expected errors in duplicating the true growth line increase markedly with time in the forecasting range. Curve D is just the opposite; program maturity effects are expected to significantly minimize the range of the weight variation about the true growth line, while accuracy remains high for prediction line matching the true growth line. Curves B and C are intermediate situations.

The question naturally arises, "What can be forecasted about the value of σ_T at the time we reach ship date?" For this discussion, it is appropriate to express the variation in terms of the probable error (PE) at the ship date, the difference between the upper confidence limit and the mean line, as

$$PE_s = C_1 \sigma_{T_s}$$

where C_1 is a factor varying from 1.645 for a large sample to 2.132 for a sample of only six observations, and σ_{T_s} is the actual expected value of σ_T at ship.

For a large sample (less than 10) a value of 2 is a reasonable approximation for C, and the equation now reads

$$PE_s \cong 2\sigma_{T_s} \cong 2\sqrt{\sigma_{k_s}^{2} + \sigma_{\hat{w}_{k_s}}^{2}}$$

It is significant to note that the probable error at ship date will decrease as the actual time approaches the ship date due to decreases in both σ_{k_s} and $\sigma_{\hat{w}_{k_s}}$. The decrease in σ_{k_s} can be predicted prior to ship and will correspond to expected decrease in weight oscillations with expected changes from estimated to calculated to actual. Likewise, $\sigma_{\hat{w}_{k_s}}$ at ship date can be estimated from known accuracies of prediction in the current observed range.

If the resultant probable error at ship, PE_s is excessively high, the over-all mission could be needlessly jeopardized by the necessity of providing excessive launch capability to handle the large contingency factor. In any event, if $\sigma_{\hat{w}_{k_s}}$ is significantly greater than σ_{k_s}, then it is possible that mass measurement accuracy requirements are excessive — a very expensive situation indeed. That is, if σ_{k_s} is low, then σ_m must also be low which may not be reasonable in view of the large variation of expected vehicle requirements as evidenced by a large $\sigma_{\hat{w}_k}$. Since σ_m is related to mass property measurement, and high accuracies of mass measurement (to minimize σ_m) are very costly, then it may be valuable to consider if potential cost reductions may be achieved by relaxing the requirements for σ_m.

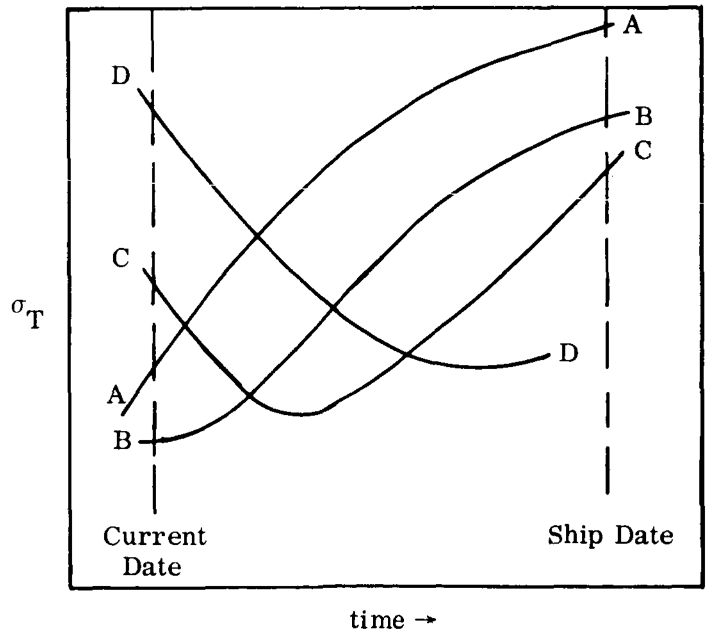


Figure B-3

APPENDIX C

MATH MODELS

C.1 LINEAR MAXIMUM LIKELIHOOD

In this discussion it is assumed that the reported weight data, represented by the random variable $\underline{w}_i$ observed at time t_i, are generated in a linear fashion. Hence, the assumed model is

$$w_i = a + b\,t_i$$

where a and b are unknown parameters to be estimated. It can be said then that this model represents the expected value of the weight or

$$E\{\underline{w}_i\} = w_i = a + b\,t_i.$$

It is also assumed that the value of the random variable $\underline{w}_i$ can be expressed as its expected value plus an unobservable error which is a random variable with zero mean and standard derivation σ_i. Thus,

$$\underline{w}_i = w_i + \underline{e}_i.$$

The observed weight w_i is reported as a summation of three values. One of these reflects that portion whose weight is totally estimated, the second is that portion which is calculated (from final design drawings) and the third is the portion which is actual or manufactured so that it can be actually weighed. The weight is therefore expressible as

$$\underline{w}_i = E_i\,\underline{w}_i + C_i\,\underline{w}_i + A_i\,\underline{w}_i$$

where

$E_i\,\underline{w}_i$ = fraction of $\underline{w}_i$ which is estimated.

$C_i\,\underline{w}_i$ = fraction of $\underline{w}_i$ which is calculated.

$A_i\,\underline{w}_i$ = fraction of $\underline{w}_i$ which is measured or actual.

Each of the three coefficients E_i, C_i, and A_i is available for every observation, or reported value of $\underline{w}_i$. The three components of $\underline{w}_i$ can now be expressed as a mean value plus an error, yielding

$$\underline{w}_i + (w_{E_i} + \underline{e}_{E_i}) + (w_{C_i} + \underline{e}_{C_i}) + (w_{A_i} + \underline{E}_{A_i})$$

where the three component means are assumed derived from the mean value of $\underline{w}_i$ as follows:

$$w_{E_i} = E_i w_i$$

$$w_{C_i} = C_i w_i$$

$$w_{A_i} = A_i w_i .$$

Since $E_i + C_i + A_i \equiv 1.0$, it follows that

$$w_i = w_{E_i} + w_{C_i} + w_{A_i}$$

$$\underline{e}_i = \underline{e}_{E_i} + \underline{e}_{C_i} + \underline{e}_{A_i} .$$

Note that the term "error" is used here in a very broad sense. Even if the total weight could be measured exactly, without any error in the conventional sense, this weight would not be expected to follow a straight line. Instead, the exact weight is considered a random phenomenon whose dispersion about some trend is caused by the interaction of many random causes. It is this dispersion or deviation which is referred to as error.

The three errors are assumed to be normally and independently distributed with zero mean and standard deviations σ_{E_i}, σ_{C_i}, and σ_{A_i}. It then follows that the variance of the error is expressible as

$$\sigma_i^2 = \sigma_{E_i}^2 + \sigma_{C_i}^2 + \sigma_{A_i}^2 .$$

It is further assumed that the ratio of standard deviation to mean of the three random weight components is constant, or specifically,

$$\frac{\sigma_{E_i}}{w_{E_i}} = s$$

$$\frac{\sigma_{C_i}}{w_{C_i}} = R_1 s$$

$$\frac{\sigma_{A_i}}{w_{A_i}} = R_2 s$$

The ratio s is an unknown parameter which will be estimated while the factors R_1 and R_2, which define the relationship among the three ratios, must be specified. The variance of the i^{th} observation can now be expressed as

$$\sigma_i^2 = s^2 m_i^2$$

where m_i^2 is a weighting factor defined as

$$m_i^2 = w_i^2 \left[E_i^2 + R_1^2 C_i^2 + R_2^2 A_i^2\right]$$

This weighting factor is actually a constant but contains the unknown value of w_i. For computational purposes one will need some estimate, say $\tilde{w}_i$, of w_i and use instead of m_i^2,

$$\tilde{m}_i^2 = \tilde{w}_i^2 \left[E_i^2 + R_1^2 C_i^2 + R_2^2 A_i^2\right]$$

As a first estimate of w_i the actual observed value can be used, i.e., $\tilde{w}_i = w_i$, or a least squares fit performed and points from the resulting function used. In the computer program developed for this linear maximum likelihood technique the latter procedure is used. The resulting $\tilde{w}_i$ is further improved in an iterative fashion by the maximum likelihood estimator for w_i.

The final assumption is that the random variables $\underline{w}_i$ and $\underline{w}_j$, where $i \neq j$, are independent. This is perhaps the most limiting assumption. For example, a project manager may decide to freeze the weight of some component for several months, the fact that the weight stays constant during that time is then not attributable to randomness, but is rather a result of strong dependence. The likelihood function is

$$L = \prod_{i=1}^{n} \left[\frac{1}{s\, m_i \sqrt{2\pi}}\, e^{-\frac{1}{2}\left(\frac{\underline{w}_i - a - b\, t_i}{s\, m_i}\right)^2} \right]$$

and the maximum of this function is

$$L_{max} = \prod_{i=1}^{n} \left[\frac{1}{\hat{s}\, m_i \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{w_i - \hat{a} - \hat{b}\, t_i}{\hat{s}\, m_i}\right)^2} \right]$$

where $\hat{a}$, $\hat{b}$, $\hat{s}$ are maximum likelihood estimators of the unknown parameters a, b, and s. Taking the partial derivatives of ln L with respect to each of the three parameters and equating them to zero produces the following normal equations:

$$\sum_{i=1}^{n} \frac{w_i - \hat{a} - \hat{b}\, t_i}{m_i^2} = 0$$

$$\sum_{i=1}^{n} \left(\frac{w_i - \hat{a} - \hat{b}\, t_i}{m_i^2}\right) t_i = 0$$

$$\frac{1}{n} \sum_{i=1}^{n} \left(\frac{w_i - \hat{a} - \hat{b}\, t_i}{m_i}\right)^2 = \hat{s}^2$$

Solving the first two equations for $\hat{a}$ and $\hat{b}$ yields

$$\hat{a} = \frac{CE - BF}{D}$$

$$\hat{b} = \frac{AF - BE}{D}$$

where

$$A = \sum_{i=1}^{n} \frac{1}{m_i^2}$$

$$B = \sum_{i=1}^{n} \frac{t_i}{m_i^2}$$

$$C = \sum_{i=1}^{n} \frac{t_i^2}{m_i^2}$$

$$D = AC - B^2$$

$$E = \sum_{i=1}^{n} \frac{\underline{w}_i}{m_i^2}$$

$$F = \sum_{i=1}^{n} \frac{\underline{w}_i t_i}{m_i}$$

It is interesting to note at this point that by setting $m_i = 1$, which says that $\sigma_i^2 = s^2$, a least squares fit is obtained. Although $\hat{\underline{a}}$ and $\hat{\underline{b}}$ are unbiased, i.e., $E(\hat{\underline{a}}) = a$ and $E(\hat{\underline{b}}) = b$, the estimator $\hat{\underline{s}}$ as defined above is biased. Its unbiased form (see C.1.2 to this section) is given by

$$\hat{\underline{s}} = \frac{1}{n-2} \sum_{i=1}^{n} \left(\frac{\underline{w}_i - \hat{\underline{a}} - \hat{\underline{b}} t_i}{m_i} \right)^2 .$$

C.1.1 PREDICTION AND PREDICTION INTERVAL

One of the benefits accruing from the maximum likelihood method is that one has enough information to establish a confidence interval, called prediction interval. Let $(t_k, \underline{w}_k)$ be pairs of future observations, which are to be predicted. The model holds in analogy to the original model, namely

$$\underline{w}_k = w_k + \underline{e}_k .$$

If a, b, and s were known we could determine $w = a + b\, t_k$ and use s to assess the confidence interval due to the error $\underline{e}_k$. What is instead available are the estimators $\hat{\underline{a}}$, $\hat{\underline{b}}$, and $\hat{\underline{s}}$. The predicted value of $\underline{w}_k$ is the estimator of w_k

$$\hat{\underline{w}}_k = \hat{\underline{a}} + \hat{\underline{b}}\, t_k$$

Note that

$$E\{\underline{w}_k\} = E\{\hat{\underline{w}}_k\} = w_k .$$

Since both $\hat{\underline{a}}$ and $\hat{\underline{b}}$ are linear combinations of the normally distributed random variables $\underline{w}_i$, $\hat{\underline{w}}_k$ is also normally distributed, hence

$$\hat{\underline{w}}_k = w_k + \underline{e}_{\hat{w}_k}$$

where $\underline{e}_{\hat{w}_k}$ is a normal error with zero mean. Let $\underline{u}_k$ be the difference between the k^{th} observation and prediction

$$\underline{u}_k = \underline{w}_k - \hat{\underline{w}}_k = \underline{e}_k - \underline{e}_{\hat{w}_k}$$

It is seen that the error $\underline{u}_k$ has two sources. One, $\underline{e}_k$, tells us that the observation $\underline{w}_k$ will deviate from its expected value, while secondly our estimate of the expected value contains the error $\underline{e}_{\hat{w}_k}$. These two errors and their relationship are illustrated in Figure C-1. In this diagram, the true trend line is provided by

$$E\{\underline{w}_i\} = w_i .$$

The error $\underline{u}_k$ is again normal with parameters

$$E\{\underline{u}_k\} = 0,$$

$$\sigma_{u_k}^2 = \text{var}\,(\underline{u}_k) = \text{var}\,(\underline{e}_k) + \text{var}\,(\underline{e}_{\hat{w}_k})$$

The last equation follows from the stipulated independence of observations. The variance of $\underline{e}_k$ is simply

$$\text{var}\,\{\underline{e}_k\} = \sigma_k^2 = s^2 \cdot m_k^2$$

In place of m_k^2 we use its estimate

$$\underline{m}_k^2 = \hat{\underline{w}}_k^2 \left[E_k^2 + R_1^2 C_k^2 + R_2^2 A_k^2 \right]$$

The variance of $\underline{e}_{\hat{w}_k}$ is obtained from the earlier expression for $\hat{\underline{w}}_k$

$$\text{var}\,(\underline{e}_{\hat{w}_k}) = \text{var}\{\hat{\underline{a}}\} + t_k^2\,\text{var}\{\hat{\underline{b}}\} + 2t_k\,\text{cov}\{\hat{\underline{a}}, \hat{\underline{b}}\}$$

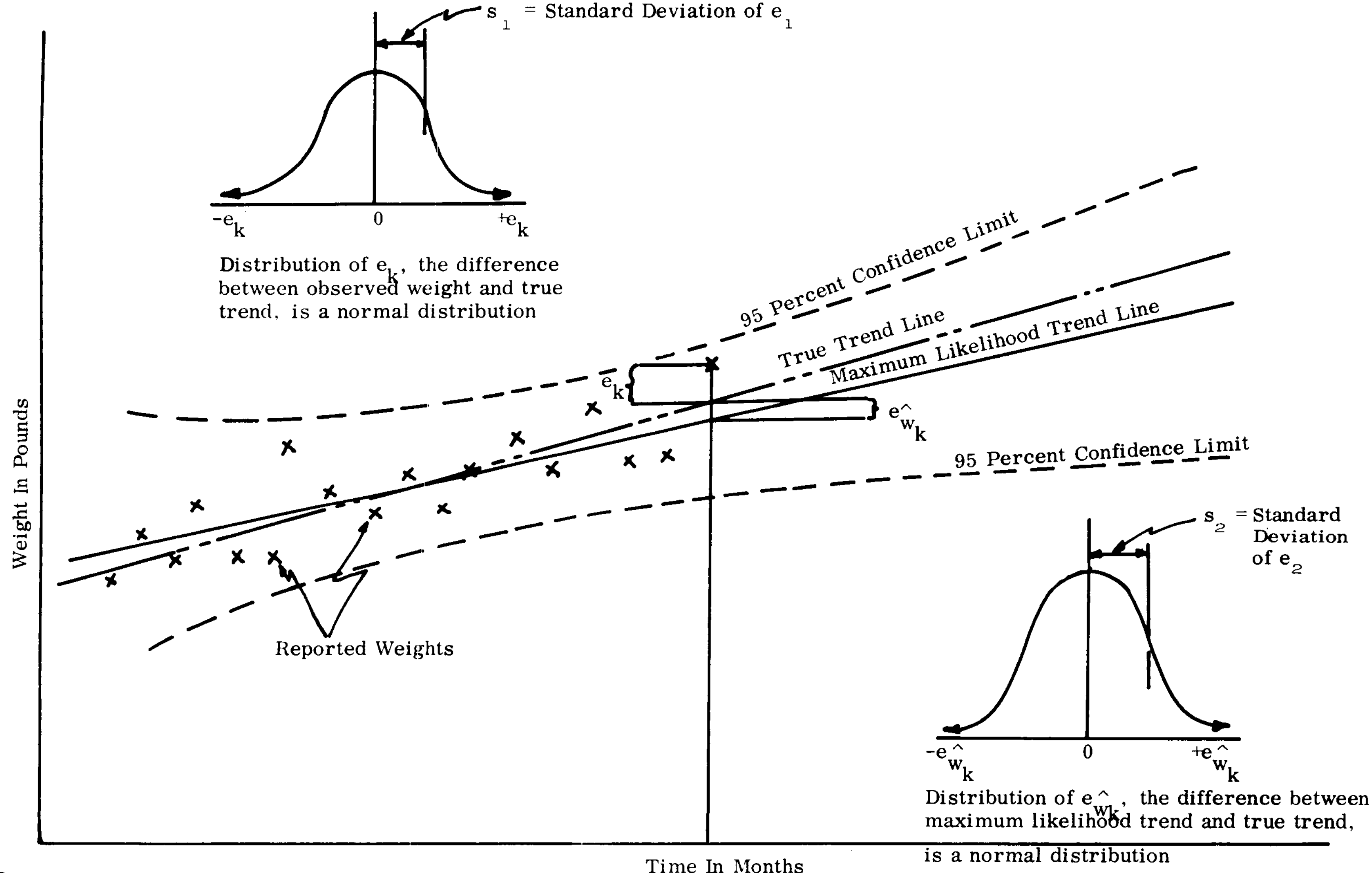


Figure C-1. Maximum-Likelihood Linear Model Showing the Two Types of Error

This equation contains the covariance of $\hat{\underline{a}}$ and $\hat{\underline{b}}$ which implies that $\hat{\underline{a}}$ and $\hat{\underline{b}}$ are dependent. There is a simpler way to find var $(\underline{e}_{\hat{w}_k})$.

Going back to the normal equations solution it is seen that $\hat{w}_k$ can be expressed as a function of the random variables $\underline{E}$ and $\underline{F}$,

$$\hat{w}_k = \alpha_k \underline{E} + \beta_k \underline{F}$$

where

$$\alpha_k = \frac{C - Bt_k}{D}$$

$$\beta_k = \frac{At_k - B}{D} .$$

This leads to

$$\text{var}\,\{\underline{e}_{\hat{w}_k}\} = \alpha_k^2 \,\text{var}\,\{\underline{E}\} + \beta_k^2 \,\text{var}\,\{\underline{F}\} + 2\alpha_k\beta_k \,\text{cov}\,\{\underline{E}, \underline{F}\}$$

Since both $\underline{E}$ and $\underline{F}$ consist of a sum of independent normal random variables, we can write down directly,

$$\text{var}\,\{\underline{E}\} = \sum_{i=1}^{n} \frac{\sigma_i^2}{m_i^4} = s^2 \sum_{i=1}^{n} \frac{1}{m_i^2} = s^2 A$$

$$\text{var}\,\{\underline{F}\} = \sum_{i=1}^{n} \frac{t_i^2}{m_i^4}\sigma_i^2 = s^2 \sum_{i=1}^{n} \frac{t_i^2}{m_i^2} = s^2 C$$

The covariance of $\underline{E}$ and $\underline{F}$ is defined as

$$\text{cov}\,\{\underline{E}, \underline{F}\} = E\left\{\left(\sum_{i=1}^{n} \frac{\underline{w}_i}{m_i^2}\right)\left(\sum_{i=1}^{n} \frac{\underline{w}_i t_i}{m_i^2}\right)\right\} - E\left\{\sum_{i=1}^{n} \frac{\underline{w}_i}{m_i^2}\right\} E\left\{\sum_{i=1}^{n} \frac{\underline{w}_i t_i}{m_i^2}\right\}$$

The equation for $\hat{\underline{w}}_k$ is simplified by the fact that, due to independence,

$$E\{\underline{w}_i \underline{w}_j\} = E\{\underline{w}_i\} \cdot E\{\underline{w}_j\}, \quad i \neq j$$

and there results

$$\text{cov}\{\underline{E}, \underline{F}\} = E\left\{\sum_{i=1}^{n} \frac{t_i^2}{m_i^4} w_i^2\right\} - \left[E\left\{\sum_{i=1}^{n} \frac{t_i}{m_i^2} w_i\right\}\right]^2$$

$$= \sum_{i=1}^{n} \frac{t_i^2}{m_i^4} \sigma_i^2 = s^2 B$$

So now the expression for var $\{\underline{e}_{w_k}\}$ becomes

$$\text{var}\{\underline{e}_{w_k}\} = s^2\left[\alpha_k^2 A + \beta_k^2 C + 2\alpha_k \beta_k B\right]$$

or by substitution for α_k and β_k

$$\text{var}\{\underline{e}_{\hat{w}_k}\} = \frac{s^2}{D}\left[A t_k^2 - 2B t_k + C\right]$$

It is now clear that

$$\text{var}\{\hat{\underline{a}}\} = s^2 \frac{C}{D}$$

$$\text{var}\{\hat{\underline{b}}\} = s^2 \frac{A}{D}$$

$$\text{cov}\{\hat{\underline{a}}, \hat{\underline{b}}\} = -s^2 \frac{B}{D}$$

and

$$\text{var}\{\underline{u}_k\} = s^2\left(m_k^2 + \frac{A t_k^2 - 2B t_k + C}{D}\right)$$

If s were known the prediction interval could be readily established since $\underline{u}_k$ is normally distributed. For example, for a 95 percent interval on $\underline{w}_k$ one would use

$$\hat{\underline{a}} + \hat{\underline{b}}_k \pm 1.96\, \sigma_{u_k} .$$

With $\hat{\underline{s}}$ available instead of s, this may still be used if the sample is large enough, say n > 30. For small samples the random variable

$$\underline{t} = \frac{\dfrac{u_k}{\sigma_{u_k}}}{\sqrt{\dfrac{\hat{\underline{s}}^2}{s^2}}}$$

has the Student T distribution with n - 2 degrees of freedom, $\hat{s}$ being the unbiased estimator for s. From this follows the small sample distribution of $\underline{u}_k$

$$\underline{u}_k = \underline{t}\,\sigma_{u_k} \frac{\hat{\underline{s}}}{s} = \underline{t} \cdot \hat{\underline{\sigma}}_{u_k}$$

where

$$\hat{\sigma}_{u_k} = \hat{\underline{s}} \left(m_k^2 + \frac{At_k^2 - 2Bt_k + C}{D} \right)^{\frac{1}{2}}$$

The confidence interval containing (i - ϵ) of all possible outcomes of $\underline{w}$ is then

$$\hat{\underline{a}} + \hat{\underline{b}}t_k \pm t_{\epsilon/2}\, \hat{\underline{\sigma}}_{u_k}$$

in the case of a sample with n = 10, and a 95 percent interval, the value of $t_{\epsilon/2}$ is 2.306 as compared to 1.96 used earlier.

C.1.2 APPENDIX

It was seen in the preceding discussion that the maximum-likelihood estimator $\hat{\underline{s}}$ of the parameter s is

$$\hat{\underline{s}}^2 = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{\underline{w}_i - \hat{\underline{a}} - \hat{\underline{b}}t_i}{m_i} \right)^2$$

This estimator is to be tested for bias, which means that its expected value has to be found.

$$E\{\hat{\underline{s}}^2\} = E\left\{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\underline{w}_i - \hat{\underline{w}}_i}{m_i}\right)^2\right\}$$

The following transformations are made:

$$E\{\hat{\underline{s}}^2\} = E\left\{\frac{1}{n}\sum_{i=1}^{n}\left[\frac{(\underline{w}_i - w_i) - (\hat{\underline{w}}_i - w_i)}{m_i}\right]^2\right\}$$

$$= E\left\{\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i^2}\left[(\underline{w}_i - w_i)^2 - 2(\underline{w}_i - w_i)(\hat{\underline{w}}_i - w_i) + (\hat{\underline{w}}_i - w_i)^2\right]\right\}$$

$$= s^2 + \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i^2}\left[-2\,\mathrm{cov}\,\{\underline{w}_i, \hat{\underline{w}}\} + \mathrm{var}\,\{\hat{\underline{w}}_i\}\right]$$

For the last term the following earlier results are used:

$$\mathrm{var}\,\{\hat{\underline{w}}\} = \mathrm{var}\,\{\underline{e}_{\hat{w}_i}\} = \frac{s^2}{D}\left(At_i^2 - 2Bt_i + C\right)$$

and

$$\sum_{i=1}^{n}\frac{1}{2}\,\mathrm{var}\,\{\hat{\underline{w}}_i\} = \frac{s^2}{D}(AC - 2B^2 + AC) = 2s^2$$

The middle term in the $E\{\hat{\underline{s}}^2\}$ expression is treated as follows:

$$\mathrm{cov}\,\{\underline{w}_i, \hat{\underline{w}}_i\} = E\{\underline{w}_i \cdot \hat{\underline{w}}_i\} - w_i^2$$

Next,

$$E\{\underline{w}_i \cdot \hat{\underline{w}}_i\} = \alpha_i E\left\{\sum_{j=1}^{n}\underline{w}_i\frac{\underline{w}_j}{m_j^2}\right\} + B_i E\left\{\sum_{j=1}^{n}\underline{w}_i\frac{\underline{w}_j t_j}{m_j^2}\right\}$$

Now, using $E\{\underline{w}_i \cdot \hat{\underline{w}}_j\} = w_i w_j$ for $(i \neq j)$

$$E\{\underline{w}_i \cdot \hat{\underline{w}}_i\} = \frac{\alpha_i + \beta_i t_i}{m_i^2} (E\{\underline{w}_i^2\} \cdot w_i^2) + w_i^2$$

and

$$\text{cov}\,\{\underline{w}_i, \hat{\underline{w}}_i\} = (\alpha + \beta_i t_i) \frac{\sigma_i^2}{m_i^2} = (\alpha + \beta_i t_i) s^2$$

Further

$$\sum_{i=1}^{n} \frac{\alpha_i + \beta_i t_i}{m_i^2} = \sum_{i=1}^{n} \frac{C - Bt_i + At_i^2 - Bt_i}{m_i^2 (AC - B^2)} = \frac{AC - B^2 + AC - B^2}{AC - B^2} = 2$$

Combining this with the expressions for $E\{\hat{\underline{s}}^2\}$ and

$$\sum_{i=1}^{n} \frac{1}{m_i^2} \text{var}\, \hat{\underline{w}}_i ,$$

$$E\{\hat{\underline{s}}^2\} = s^2 - \frac{4s^2}{n} + \frac{2s^2}{n} = \frac{n-2}{n} s^2$$

From this follows that the unbiased estimator of s should be

$$\hat{\underline{s}}^2 = \frac{1}{n-2} \sum_{i=1}^{n} \left(\frac{\underline{w}_i - \hat{\underline{a}} - \hat{\underline{b}} t_i}{m_i} \right)^2$$

C.1.3 SAMPLE CALCULATIONS

An understanding of the computational mechanics involved in exercising this model can be best obtained by examining a step-by-step numerical example. The following material results from a calculation using the appropriate expressions from the above discussion. The computer program was applied to the same set of input data and the agreement between final results was very good. The computer output is included at the end of the computations (see Figures C-2 and C-3).

The original data is as follows:

Case No. 30

Date	t_i	$\underline{w}_i$	A_i	C_i	E_i	i
8/63	1	86405	0	0.90	0.10	1
9/63	2	87916	0	0.90	0.10	2
10/63	3	-	-	-	-	-
11/63	4	86648	0.01	0.91	0.08	3
12/63	5	-	-	-	-	-
1/64	6	-	-	-	-	-
2/64	7	86992	0.02	0.90	0.08	4
3/64	8	86542	0.02	0.90	0.08	5
4/64	9	-	-	-	-	-
5/64	10	88056	0.02	0.90	0.08	6
6/64	11	88460	0.02	0.90	0.08	7
7/64	12	88791	0.02	0.92	0.06	8

For the least squares estimators, $\hat{\underline{a}}_{\ell s}$ and $\hat{\underline{b}}_{\ell s}$,

$$\hat{\underline{a}}_{\ell s} = \frac{C\underline{E} - B\underline{F}}{D}$$

$$\hat{\underline{b}}_{\ell s} = \frac{A\underline{F} - B\underline{E}}{D}$$

where:

$$A = \sum_{i=1}^{n} \frac{1}{m_i^2}$$

$$B = \sum_{i=1}^{n} \frac{t_i}{m_i^2}$$

$$C = \sum_{i=1}^{n} \frac{t_i^2}{m_i^2}$$

$$D = AC - B^2$$

$$\underline{E} = \sum_{i=1}^{n} \frac{\underline{w}_i}{m_i^2}$$

$$\underline{F} = \sum_{i=1}^{n} \frac{\underline{w}_i t_i}{m_i^2}$$

These equations give the maximum likelihood estimators, $\hat{\underline{a}}$ and $\hat{\underline{b}}$. By setting the $m_i = 1$, the least squares estimators are obtained.

i	t_i	t_i^2	$\underline{w}_i$	$w_i t_i$
1	1	1	86401	86401
2	2	4	87916	175832
3	4	16	86648	346592
4	7	49	86992	608944
5	8	64	86542	692336
6	10	100	88056	880560
7	11	121	88460	973060
8	12	144	88791	1065492
Σ	55	499	699806	4829217

$$A = \sum_{i=1}^{8} \left(\frac{1}{1}\right)_i = 8$$

$$B = \sum_{i=1}^{8} t_i = 55$$

$$C = \sum_{i=1}^{8} t_i^2 = 499$$

$$D = AC - B^2 = 8(499) - (55)^2$$

$$= 3992 - 3025 = 967$$

$$\underline{E} = \sum_{i=1}^{8} \underline{w}_i = 699806$$

$$\underline{F} = \sum_{i=1}^{8} w_i t_i = 4829217$$

$$\hat{\underline{a}}_{\ell s} = \frac{C\underline{E} - B\underline{F}}{D} = \frac{499(699806) - 55(4829217)}{967} = \frac{349203194 - 265606935}{967}$$

$$= \frac{83596259}{967} = 86449.079$$

$$\hat{\underline{b}}_{\ell s} = \frac{A\underline{F} - B\underline{E}}{D} = \frac{8(4829217) - 55(699806)}{967} = \frac{38633736 - 38489330}{967}$$

$$= \frac{144406}{967} = 149.334$$

For the least squares curve fit,

$$w_i = \hat{\underline{a}}_{\ell s} + \hat{\underline{b}}_{\ell s} t_i .$$

i	t_i	$\hat{\underline{b}}_{\ell s} t_i$	$\hat{\underline{a}}_{\ell s} + \hat{\underline{b}}_{\ell s} t_i$	w_i
1	1	149.334	86598.413	86598
2	2	298.668	86747.747	86748
3	4	597.336	87046.415	87046
4	7	1045.338	87494.417	87494
5	8	1194.672	87643.751	87644
6	10	1493.340	87942.419	87942
7	11	1642.674	88091.753	88092
8	12	1792.008	88241.087	88241
9	13	1941.342	88390.421	88390

For the first estimate of m_i,

$$\tilde{m}_i^2 = \tilde{w}_i^2(E_i^2 + R_1^2 C_i^2 + R_2^2 A_i^2) .$$

$$R_1 = 25 , R_1^2 = 625$$

$$R_2 = 50 , R_2^2 = 2500$$

i	E_i	E_i^2	C_i	C_i^2	$R_1^2C_i^2$	A_i	A_i^2	$R_1^2c_i^2$	$E_i^2+A_1^2C_i^2+R_2^2A_i^2$
1	0	0	0.90	0.8100	506.2500	0.10	0.01	25.00	531.2500
2	0	0	0.90	0.8100	506.2500	0.10	0.01	25.00	531.2500
3	0.01	0.0001	0.91	0.8281	517.5625	0.08	0.0064	16.00	533.5626
4	0.02	0.0004	0.90	0.8100	506.2500	0.08	0.0064	16.00	522.2504
5	0.02	0.0004	0.90	0.8100	506.2500	0.08	0.0064	16.00	522.2504
6	0.02	0.0004	0.90	0.8100	506.2500	0.08	0.0064	16.00	522.2504
7	0.02	0.0004	0.90	0.8100	506.2500	0.08	0.0064	16.00	522.2504
8	0.02	0.0004	0.92	0.8464	529.0000	0.06	0.0036	9.00	538.0004

i	w_i	w_i^2	$E_i^2+R_1^2C_i^2+R_2^2A_i^2$	m_i^2	$1/m_i^2$
1	86598.413	74.992851×10^8	531.2500	398.39952×10^{10}	$25.100432 \times 10^{-14}$
2	86747.747	75.751716×10^8	531.2500	399.77474×10^{10}	$25.014087 \times 10^{-14}$
3	87046.415	75.770784×10^8	533.5626	404.28457×10^{10}	$24.735052 \times 10^{-14}$
4	87494.417	76.552730×10^8	522.2504	399.79694×10^{10}	$25.012698 \times 10^{-14}$
5	87643.751	76.814271×10^8	522.2504	401.16284×10^{10}	$24.927533 \times 10^{-14}$
6	87942.419	77.338691×10^8	522.2504	403.90162×10^{10}	$24.758504 \times 10^{-14}$
7	88091.753	77.601569×10^8	522.2504	405.27450×10^{10}	$24.674634 \times 10^{-14}$
8	88241.087	77.864894×10^8	538.0004	418.91344×10^{10}	$23.871280 \times 10^{-14}$
Σ	-	-	-	-	$198.094220 \times 10^{-14}$

Further iterations of m_i using values of w_i obtained from the maximum likelihood estimators were not made in the hand calculations.

For the maximum likelihood estimators, $\hat{\underline{a}}_{m\ell}$ and $\hat{\underline{b}}_{m\ell}$,

$$\hat{\underline{a}}_{m\ell} = \frac{C\underline{E} - B\underline{F}}{D}$$

$$\hat{\underline{b}}_{m\ell} = \frac{A\underline{F} - B\underline{E}}{D}$$

where:

$$A = \sum_{i=1}^{n} \frac{1}{m_i^2}$$

$$B = \sum_{i=1}^{n} \frac{t_i}{m_i^2}$$

$$C = \sum_{i=1}^{n} \frac{t_i^2}{m_i^2}$$

$$D = AC - B^2$$

$$\underline{E} = \sum_{i=1}^{n} \frac{\underline{w}_i}{m_i^2}$$

$$\underline{F} = \sum_{i=1}^{n} \frac{\underline{w}_i t_i}{m_i^2}$$

i	t_i	$(t_i/m_i^2) \times 10^{14}$	t_i^2	$(t_i^2/m_i^2) \times 10^{14}$	$(\underline{w}_i/m_i^2) \times 10^{14}$
1	1	25.100432	1	25.100432	2168702.4
2	2	50.028174	4	100.056348	2199138.5
3	4	98.940208	16	395.760832	2143242.8
4	7	175.088886	49	1225.622202	2175904.6
5	8	199.420264	64	1595.362112	2157278.6
6	10	247.585040	100	2475.850400	2180134.8
7	11	271.420974	121	2985.630714	2182718.1
8	12	286.455360	144	3437.464320	2119554.8
Σ	-	1354.039338	-	12240.847360	17326674.6

i	$\underline{w}_i t_i$	$(t_i w_i/m_i^2) \times 10^{14}$
1	86401	2168702.4
2	175832	4398276.9
3	346592	8572971.1
4	608944	15231332.4
5	692336	17258228.5
6	880560	21801348.3
7	973060	24009899.4
8	1065492	25434657.9
Σ	-	118875416.9

$$A = \sum_{i=1}^{8} \frac{1}{m_i^2} = 198.09422 \times 10^{-14}$$

$$B = \sum_{i=1}^{8} \frac{t_i}{m_i^2} = 1354.0393 \times 10^{-14}$$

$$C = \sum_{i=1}^{8} \frac{t_i^2}{m_i^2} = 12240.847 \times 10^{-14}$$

$$D = AC - B^2 = (198.09422 \times 10^{-14})(12240.847 \times 10^{-14}) - (1354.0393 \times 10^{-14})^2$$

$$= (2424841.1 \times 10^{-28}) - (1833422.5 \times 10^{-28})$$

$$= 591418.6 \times 10^{-28}$$

$$\underline{E} = \sum_{i=1}^{8} \frac{\underline{w}_i}{m_i^2} = 17326674.5 \times 10^{-14}$$

$$\underline{F} = \sum_{i=1}^{8} \frac{\underline{w}_i t_i}{m_i^2} = 118875416.9 \times 10^{-14}$$

$$\hat{\underline{a}}_{m\ell} = \frac{C\underline{E} - B\underline{F}}{D}$$

$$= \frac{(12240.847 \times 10^{-14})(17326674.5 \times 10^{-14}) - (1354.0393 \times 10^{-14})(118875416.9 \times 10^{-14})}{591418.6 \times 10^{-28}}$$

$$= \frac{2.120931716 \times 10^{-17} - 1.609619863 \times 10^{-17}}{5.914186 \times 10^{-23}}$$

$$= \frac{0.511311853 \times 10^{-17}}{5.914186 \times 10^{-23}} = 86455.153$$

$$\hat{\underline{b}}_{m\ell} = \frac{A\underline{F} - B\underline{E}}{D}$$

$$= \frac{(198.09422 \times 10^{-14})(118875416.9 \times 10^{-14}) - (1354.0393 \times 10^{-14})(17326674.5 \times 10^{-14})}{591418.6 \times 10^{-28}}$$

$$= \frac{2.354853299 \times 10^{-18} - 2.34609982 \times 10^{-18}}{5.914186 \times 10^{-23}}$$

$$= \frac{8.753479 \times 10^{-21}}{5.914186 \times 10^{-23}} = 148.008$$

For the maximum likelihood curve fit,

$$w_i = \hat{\underline{a}}_{m\ell} + \hat{\underline{b}}_{m\ell} t_i$$

i	t_i	$\hat{\underline{b}}_{m\ell} t_i$	$\hat{\underline{a}}_{m\ell} + \hat{\underline{b}}_{m\ell} t_i$	$w_{i(m\ell)}$
1	1	148.008	86603.161	88603
2	2	296.016	86751.169	86751
3	4	592.032	87047.105	87047
4	7	1036.056	87491.209	87491
5	8	1184.064	87639.217	87639
6	10	1480.080	87935.233	87935
7	11	1628.088	88083.241	88083
8	12	1776.096	88231.249	88231
9	13	1924.104	88379.257	88379

For the unbiased maximum likelihood estimator, $\hat{\underline{s}}$,

$$\hat{\underline{s}}^2 = \frac{1}{n-2} \sum_{i=1}^{n} \left(\frac{\underline{w}_i - \hat{\underline{a}}_{m\ell} - \hat{\underline{b}}_{m\ell} t_i}{m_i} \right)^2$$

i	$\underline{w}_i$	$\hat{\underline{a}} + \hat{\underline{b}}t_i$	$\underline{w}_i - \hat{\underline{a}} - \hat{\underline{b}}t_i$	$(\underline{w}_i - \hat{\underline{a}} - \hat{\underline{b}}t_i)^2$
1	86401	86603.161	-202.161	4.08690699×10^4
2	87916	86751.169	1164.831	1.35683126×10^6
3	86648	87047.185	-399.185	1.59348664×10^5
4	86992	87491.209	-499.209	2.49209626×10^5
5	86542	87639.217	-1097.217	1.20388515×10^6
6	88056	87935.233	120.767	1.45846683×10^4
7	88460	88083.241	376.759	1.41947344×10^5
8	88791	88231.249	559.751	3.13321182×10^5

i	$1/m_i^2$	$(\underline{w}_i - \hat{\underline{a}} - \hat{\underline{b}}t_i)^2/m_i^2$
1	$25.100432 \times 10^{-14}$	1.0258313×10^{-8}
2	$25.014087 \times 10^{-14}$	$33.9398952 \times 10^{-8}$
3	$24.735052 \times 10^{-14}$	3.9414975×10^{-8}
4	$25.012698 \times 10^{-14}$	6.2334051×10^{-8}
5	$24.927533 \times 10^{-14}$	$30.0098868 \times 10^{-8}$
6	$24.758504 \times 10^{-14}$	0.3610946×10^{-8}
7	$24.674634 \times 10^{-14}$	3.5024988×10^{-8}
8	$23.871280 \times 10^{-14}$	7.4793777×10^{-8}
Σ		$86.4934870 \times 10^{-8}$

$$\hat{\underline{s}}^2 = \frac{1}{n-2} \sum_{i=1}^{8} \left(\frac{\underline{w}_i - \hat{\underline{a}} - \hat{\underline{b}}t_i}{m_i} \right)^2$$

$$= \frac{1}{8-2} \cdot 86.4934870 \times 10^{-8}$$

$$\hat{\underline{s}}^2 = 14.415581 \times 10^{-8}$$

$$\hat{\underline{s}} = 3.7967856 \times 10^{-4}$$

For the error in the observed value, s_1,

$$s_1^2 = \text{var}\left\{\underline{e}_k\right\} = s^2 \cdot m_k^2$$

Using $\hat{s}$ for s and $\hat{m}_k$ for m_k, where

$$\hat{m}_k = \hat{w}_k^2 (E_k^2 + R_1^2 C_k^2 + R_2^2 A_k^2)$$

then

$$s_1 = \hat{s} \cdot \hat{w}_k (E_k^2 + R_1^2 C_k^2 + R_2^2 A_k^2)^{\frac{1}{2}}$$

i	( ) x 10^{-2}	($)^{\frac{1}{2}}$ x 10^{-1}	$\hat{w}_k$ x 10^{-4}	$\hat{w}_k($ $)^{\frac{1}{2}}$ x 10^{-6}	s_1 x 10^{-2}
1	5.312500	2.3048861	8.6603161	1.9961042	7.5787797
2	5.312500	2.3048861	8.6751169	1.9995156	7.5917320
3	5.335626	2.3098974	8.7047185	2.0107007	7.6341995
4	5.222504	2.2852799	8.7491209	1.9994190	7.5913653
5	5.222504	2.2852799	8.7639217	2.0028014	7.6042075
6	5.222504	2.2852799	8.7935233	2.0095662	7.6298920
7	5.222504	2.2852799	8.8083241	2.0129486	7.6427343
8	5.380004	2.3194836	8.8231249	2.0465094	7.7701574

For the error in the predicted value, s_2,

$$s_2 = \text{var}\left\{ e_{\hat{w}_k} \right\} = \frac{s^2}{D} (A t_k^2 - 2Bt_k + C)$$

where:

$$A = \sum_{k=1}^{n} \frac{1}{m_k^2}$$

$$B = \sum_{k=1}^{n} \frac{t_k}{m_k^2}$$

$$C = \sum_{k=1}^{n} \frac{t_k^2}{m_k^2}$$

$$D = AC - B^2$$

Using $\hat{s}$ for s_1

$$s_2 = \hat{s} \left(\frac{At_k^2 - 2Bt_k + C}{D} \right)^{\frac{1}{2}}$$

where:

$A = 1.9809422 \times 10^{-12}$

$B = 13.540393 \times 10^{-12}$

$C = 122.40847 \times 10^{-12}$

$D = 59.141860 \times 10^{-24}$

$2B = 27.080786 \times 10^{-12}$

	(1)	(2)	(3)	(4)	(5)	(6)
		$\times 10^{12}$	$\times 10^{12}$		$\times 10^{12}$	$\times 10^{12}$
i	t_i	2B	(1) x (2)	t_i^2	A	(4) x (5)
1	1	27.080786	27.080786	1	1.9809422	1.9809422
2	2	27.080786	54.161572	4	1.9809422	7.9237688
3	4	27.080786	108.323144	16	1.9809422	31.6950752
4	7	27.080786	189.565502	49	1.9809422	97.0661678
5	8	27.080786	216.646288	64	1.9809422	126.7803008
6	10	27.080786	270.807860	100	1.9809422	198.0942200
7	11	27.080786	297.888646	121	1.9809422	239.6940062
8	12	27.080786	324.969432	144	1.9809422	285.2556768

	(7)	(8)	(9)	(10)	(11)
	$\times 10^{12}$	$\times 10^{12}$	$\times 10^{24}$	$\times 10^{-12}$	$\times 10^{-6}$
i	C	(6) + (7) - (3)	D	(8) ÷ (9)	$((10))^{\frac{1}{2}}$
1	122.40847	97.3086262	59.141860	1.645342676	1.2827091
2	122.40847	76.1706668	59.141860	1.287931539	1.13487071
3	122.40847	45.7804012	59.141860	0.774077805	0.87981692
4	122.40847	29.9091358	59.141860	0.505718551	0.7111389
5	122.40847	32.5424828	59.141860	0.550244493	0.7417847
6	122.40847	49.6948300	59.141860	0.840264915	0.9166597
7	122.40847	64.2138302	59.141860	1.085759396	1.0419978
8	122.40847	82.6947148	59.141860	1.398243389	1.1824734

	(12)	(13)
	$\times 10^{4}$	$\times 10^{-2}$
i	$\hat{\underline{s}}$	(11) x (12)
1	3.7967856	4.8701714
2	3.7967856	4.3088608
3	3.7967856	3.3404762
4	3.7967856	2.7000227
5	3.7967856	2.8163774
6	3.7967856	3.4803356
7	3.7967856	3.9562141
8	3.7967856	4.4895661

Thus the two errors, s_1 and s_2, are

i	s_2	s_1
1	487.0	757.9
2	430.9	759.2
3	334.0	763.4
4	270.0	759.1
5	281.6	760.4
6	348.0	763.0
7	395.6	764.3
8	449.0	777.0

For the 95-percent confidence interval,

$$\sigma_{uk}^2 = \text{var}\left\{\underline{u}_k\right\} = \text{var}\left\{\underline{e}_k\right\} + \text{var}\left\{\underline{e}_{w_k}\right\}$$

where $\underline{u}_k$ is the error between the k^{th} observation and the prediction. Then,

$$\underline{\hat{\sigma}}_{uk}^2 = s_1^2 + s_2^2$$

$$\underline{\hat{\sigma}}_{uk} = (s_1^2 + s_2^2)^{\frac{1}{2}}$$

and

$$\text{Confidence Interval} = \underline{\hat{a}} + \underline{\hat{b}}\,t_k \pm t_{\epsilon/2}\underline{\hat{\sigma}}_{uk}$$

and for a 95-percent interval with sample size n = 8,

$$t_{\epsilon/2} = 2.447$$

i	① $\times 10^{-2}$ s_1	② $\times 10^{-4}$ s_1^2	③ $\times 10^{-2}$ s_2	④ $\times 10^{-4}$ s_2^2	⑤ $\times 10^{-14}$ $s_1^2 + s_2^2$
1	7.5787797	57.437902	4.8701714	23.718569	81.156471
2	7.5917320	57.634395	4.3088608	18.566281	76.200676
3	7.6341995	58.281002	3.3404762	11.158781	69.439783
4	7.5913653	57.628827	2.7000227	7.290123	64.918950
5	7.6042075	57.823972	2.8163774	7.931982	65.755954
6	7.6298920	58.215252	3.4803356	12.112736	70.327988
7	7.6427343	58.411388	3.9562141	15.651630	74.063018
8	7.7701574	60.375346	4.4895661	20.156204	80.531550

i	$\times 10^{-2}$ $\hat{\sigma}_{uk}$	$t_{\epsilon/2}\hat{\sigma}_{uk}$	$\hat{a} + \hat{b}t_k$	-95% C.I.	+95% C.I.
1	9.0086886	2204.426	86603.161	84398.735	88807.587
2	8.7292999	2136.060	86751.169	84615.109	88887.229
3	8.3330536	2039.098	87047.185	85008.087	89086.283
4	8.0572297	1971.604	87491.209	85519.605	89462.813
5	8.1090045	1984.273	87639.217	85654.944	89623.490
6	8.3861784	2052.098	87935.233	85883.135	89987.331
7	8.6059873	2105.885	88083.241	85977.356	90189.126
8	8.9739373	2195.922	88231.249	86035.327	90427.171

C.2 THE EXPONENTIAL MODEL

The data used to quantize the model consist of n observations of the random variable $\underline{w}_i$ made at times t_i. The expected value of $\underline{w}_i$, or the assumed process model, is given by the following exponential function of time,

$$E\{\underline{w}_i\} = w_i = a - be^{-ct_i}, \quad c > 0$$

Two methods have been used to evaluate the parameters a, b, and c. In one, the classic, direct application of maximum likelihood estimation results in four normal equations, three of which are coupled and non-linear. Solution of this trio is accomplished by an iterative numerical procedure. Experience has shown, however, that this iterative process often fails to converge, in which event the model cannot be used. The second method, designed to overcome this problem, results in three, linear, normal equations, two of which are coupled. The linearity is achieved by assigning values to c (the parameter producing the non-linear terms in the first method) from within a known interval. This allows a straightforward solution for a and b evaluation of the likelihood function. This is repeated for other values of c within the interval and the

RESTO 28 4 1963 25. 50. 27
CONSTANT MISSION SPACECRAFT WEIGHT
PROGRAM 791—PHASE B LINEAR TREND PREDICTION CASE NO. 30

TIME IN MONTHS	*	OBSERVED WEIGHT (LBS.)	**	--- EST.	*	P E R C E N T --- CALC	*	--- ACT.	**	S1 (LBS.)		S2 (LBS.)
******************	******	******************	******	******	******	******	******	******	******	******	******	******
AUG'63	*	86401	**	10	*	90	*	0	**	757.9	*	1281.5
SEP'63	*	87916	**	10	*	90	*	0	**	759.2	*	1097.8
OCT'63	*		**		*		*		**		*	
NOV'63	*	86648	**	8	*	91	*	1	**	763.5	*	734.6
DEC'63	*		**		*		*		**		*	
JAN'64	*		**		*		*		**		*	
FEB'64	*	86992	**	8	*	90	*	2	**	759.2	*	250.7
MAR'64	*	86542	**	8	*	90	*	2	**	760.5	*	207.5
APR'64	*		**		*		*		**		*	
MAY'64	*	88056	**	8	*	90	*	2	**	763.0	*	460.5
JUN'64	*	88460	**	8	*	90	*	2	**	764.3	*	633.1
JUL'64	*	88791	**	6	*	92	*	2	**	777.1	*	811.9
BEGIN EXTRAPOLATION												
AUG'64	*		**	6	*	89	*	5	**	750.6	*	993.5
SEP'64	*		**	6	*	85	*	9	**	724.0	*	1176.7
OCT'64	*		**	5	*	82	*	12	**	697.3	*	1360.7
NOV'64	*		**	5	*	79	*	16	**	670.5	*	1545.4
DEC'64	*		**	5	*	76	*	19	**	643.7	*	1730.4
JAN'65	*		**	5	*	72	*	23	**	616.7	*	1915.7
FEB'65	*		**	4	*	69	*	26	**	589.7	*	2101.2
MAR'65	*		**	4	*	66	*	30	**	562.6	*	2286.9
APR'65	*		**	4	*	62	*	33	**	535.4	*	2472.7
MAY'65	*		**	4	*	59	*	37	**	508.1	*	2658.6
JUN'65	*		**	4	*	56	*	40	**	480.7	*	2844.5
JUL'65	*		**	3	*	53	*	44	**	453.2	*	3030.5
AUG'65	*		**	3	*	49	*	47	**	425.7	*	3216.6
SEP'65	*		**	3	*	46	*	51	**	398.0	*	3402.7
OCT'65	*		**	3	*	43	*	54	**	370.3	*	3588.9
NOV'65	*		**	3	*	39	*	58	**	342.5	*	3775.1
DEC'65	*		**	2	*	36	*	61	**	314.7	*	3961.3
JAN'66	*		**	2	*	33	*	65	**	286.8	*	4147.6
FEB'66	*		**	2	*	30	*	68	**	258.8	*	4333.8
MAR'66	*		**	2	*	26	*	72	**	230.8	*	4520.1
APR'66	*		**	2	*	23	*	75	**	202.8	*	4706.4
MAY'66	*		**	1	*	20	*	79	**	174.8	*	4892.7
JUN'66	*		**	1	*	16	*	82	**	146.9	*	5079.0
JUL'66	*		**	1	*	13	*	86	**	119.3	*	5265.3
AUG'66	*		**	1	*	10	*	89	**	92.2	*	5451.7
SEP'66	*		**	0	*	7	*	93	**	66.4	*	5638.0
OCT'66	*		**	0	*	3	*	96	**	44.5	*	5824.4
NOV'66	*		**	0	*	0	*	100	**	35.1	*	6010.8

Figure C-2. Sample Computer Printout Spacecraft Weight (Sheet 1 of 2)

****	PREDICTED WEIGHT IN POUNDS						
****	LEAST	**	- - - - - - - - - - MAXIMUM LIKELIHOOD - - - - - - -				
****	SQUARES	**	-95 P.C. CONF.	*	MEAN	*	+95P.C. CONF.
****	86598	**	82960	*	86603	*	90246
****	86748	**	83485	*	86751	*	90017
****		**		*		*	
****	87046	**	84454	*	87047	*	89640
****		**		*		*	
****		**		*		*	
****	87494	**	85535	*	87491	*	89448
****	87644	**	85710	*	87639	*	89568
****		**		*		*	
****	87942	**	85754	*	87935	*	90116
****	88092	**	85654	*	88083	*	90512
****	88241	**	85481	*	88231	*	90981
****	88390	**	85332	*	88379	*	91426
****	88540	**	85146	*	88527	*	91908
****	88689	**	84934	*	88675	*	92417
****	88838	**	84701	*	88823	*	92945
****	88988	**	84453	*	88971	*	93489
****	89137	**	84195	*	89119	*	94044
****	89286	**	83927	*	89267	*	94607
****	89435	**	83652	*	89415	*	95178
****	89585	**	83372	*	89563	*	95754
****	89734	**	83088	*	89711	*	96334
****	89883	**	82800	*	89859	*	96918
****	90033	**	82509	*	90007	*	97505
****	90182	**	82215	*	90155	*	98095
****	90331	**	81920	*	90303	*	98686
****	90481	**	81623	*	90451	*	99280
****	90630	**	81324	*	90599	*	99875
****	90779	**	81023	*	90747	*	100471
****	90929	**	80722	*	90895	*	101069
****	91078	**	80420	*	91043	*	101667
****	91227	**	80116	*	91191	*	102266
****	91377	**	79812	*	91339	*	102866
****	91526	**	79507	*	91487	*	103467
****	91675	**	79202	*	91635	*	104069
****	91825	**	78896	*	91783	*	104671
****	91974	**	78589	*	91931	*	105273
****	92123	**	78282	*	92079	*	105877
****	92273	**	77975	*	92227	*	106480
****	92422	**	77667	*	92375	*	107084

Figure C-2. Sample Computer Printout Spacecraft Weight (Sheet 2 of 2) search is directed always toward the c value providing higher values of the likelihood function. When the maximum value of this function is thus determined, the corresponding values of a, b, and c are, by definition, the maximum likelihood estimators and the model is then quantized. It should also be noted that the second method leads directly to a means of determining confidence intervals, whereas the first method would require the use of Monte Carlo methods or linear approximations.

As both of these methods have been programmed and used (the second is used exclusively at this time) the following discussion includes the details of both. The first portion is identical in both cases.

It has been assumed that the following relationship holds:

$$\underline{w}_i = w_i + \underline{e}_i$$

where $\underline{e}_i$ is an unobserved normal error having zero mean and standard deviation σ_i. The observed weight consists of three parts, namely

$$\underline{w}_i = E_i\underline{w}_i + C_i\underline{w}_i + A_i\underline{w}_i$$

with

$E_i\underline{w}_i$ = fraction of $\underline{w}_i$ which is Estimated.

$C_i\underline{w}_i$ = fraction of $\underline{w}_i$ which is Calculated.

$A_i\underline{w}_i$ = fraction of $\underline{w}_i$ which is Actual or manufactured.

The coefficients E_i, C_i, and A_i are available for every observation. Each of the three parts is now broken into its mean plus an error,

$$\underline{w}_i = (w_{E_i} + \underline{e}_{E_i}) + (w_{C_i} + \underline{e}_{C_i}) + (w_{A_i} + \underline{e}_{A_i})$$

with the assumption

$$w_{E_i} = E_i\underline{w}_i$$

$$w_{C_i} = C_i\underline{w}_i$$

$$w_{A_i} = A_i\underline{w}_i$$

It follows that,

$$w_i = w_{E_i} + w_{C_i} + w_{A_i}$$

$$\underline{e}_i = \underline{e}_{E_i} + \underline{e}_{C_i} + \underline{e}_{A_i}$$

The term "error" should be understood in a very broad sense. Even if the total weight would be measured exactly, without any error in the conventional sense, this weight would still not be expected to follow the exponential function.

Instead, view the exact weight as being a random phenomenon, the deviation of which from some trend is caused by the interaction of many random causes. It is this deviation which is the "error" of interest here.

The three component errors are assumed to be normally and independently distributed with zero mean and standard deviations σ_{E_i}, σ_{C_i}, and σ_{A_i}. A fundamental result of statistics then states that

$$\sigma_i^2 = \sigma_{E_i}^2 + \sigma_{C_i}^2 + \sigma_{A_i}^2$$

It is further assumed that the ratio of standard deviation to mean of the three random weights is constant, specifically

$$\frac{\sigma_{E_i}}{w_{E_i}} = s$$

$$\frac{\sigma_{C_i}}{w_{C_i}} = R_1 s$$

$$\frac{\sigma_{A_i}}{w_{A_i}} = R_2 s$$

The ratio s is an unknown parameter which will be estimated, while the factors R_1 and R_2 must be specified.

The variance of the i^{th} observation now obtains as

$$\sigma^2 = s^2 \cdot m_i^2$$

where m_i^2 is a weighting factor

$$m_i^2 = w_i^2 \left[E_i^2 + R_1^2 C_i^2 + R_2^2 A_i^2 \right]$$

The weighting factor is actually a constant, but contains the unknown value of w_i. For computational purposes one will need some estimates $\tilde{w}_i$ of w_i, and use instead of m_i^2

$$\tilde{m}_i^2 = \tilde{w}_i^2 \left[E_i^2 + R_1^2 C_i^2 + R_2^2 C_i^2 \right]$$

As a first estimate of w_i one can use the actual observation, i.e., $\tilde{w}_i = w_i$, or else points from a linear curve fit. The latter procedure is used in the computer program developed for this model.

The final assumption is that the random variables $\underline{w}_i$ and $\underline{w}_j$, where $i \neq j$, are independent. This is perhaps the most limiting assumption. For example, a project manager may decide to freeze the weight of some component for several months. The fact that the weight stays constant during that time is then not attributable to randomness, but is rather a result of strong dependence.

The likelihood function is

$$L = \prod_{i=1}^{n} \left[\frac{1}{s m_i \sqrt{2\pi}} \, e^{-\frac{1}{2}\left(\frac{\underline{w}_i - a + b e^{-c t_i}}{s m_i} \right)^2} \right]$$

It is at this point that the treatment diverges as described earlier. In the first case, $\hat{a}$, $\hat{b}$, $\hat{c}$, and $\hat{s}$ are to be estimated, while in the second, values are assigned to c and a, b and s only are estimated.

C.2.1 METHOD I

It is required that the maximum likelihood estimators $\hat{a}$, $\hat{b}$, $\hat{c}$, and $\hat{s}$ be found which yield the maximum value of the likelihood function, i.e.,

$$L_{max} = \prod_{i=1}^{n} \left[\frac{1}{\hat{s} m_i \sqrt{2\pi}} \, e^{-\frac{1}{2}\left(\frac{w_i - \hat{a} + \hat{b} e^{-\hat{c} t_i}}{\hat{s} m_i} \right)^2} \right]$$

Taking the partial derivatives of the natural logarithm of L

$$\ln L = -\frac{1}{2} \sum_{i=1}^{n} \left(\frac{w_i - a + b e^{-c t_i}}{s m_i} \right)^2 - \frac{n}{2} \ln 2\pi - n \ln s - \sum_{i=1}^{n} \ln m_i$$

with respect to the four parameters and equating them to zero produces the following normal equations

$$\sum_{i=1}^{n} \frac{w_i - \hat{a} + \hat{b} e^{-\hat{c} t_i}}{m_i^2} = 0$$

$$\sum_{i=1}^{n} \frac{w_i - \hat{a} + \hat{b} e^{-\hat{c} t_i}}{m_i^2} \left(e^{-\hat{c} t_i} \right) = 0$$

$$\sum_{i=1}^{n} \frac{w_i - \hat{a} + \hat{b} e^{-\hat{c} t_i}}{m_i^2} \left(- t_i \hat{b} e^{-\hat{c} t_i} \right) = 0$$

$$\hat{s}^2 = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{w_i - \hat{a} + \hat{b} e^{-\hat{c} t_i}}{m_i} \right)^2$$

Of these four equations, the last one is seen to be independent, while the first three are simultaneous equations. Since they are nonlinear, there exists no closed form solution. The computer program resorts to an iterative solution, the derivation of which is given in the following section. One starts out with an initial estimate $\hat{a}_1$, $\hat{b}_1$, and $\hat{\underline{c}}_1$ of the three parameters a, b, and c. The iterative scheme provides corrections of the previous estimate. Thus, after the j^{th} iteration,

$$\begin{bmatrix} \hat{\underline{a}}_{j+1} \\ \hat{\underline{b}}_{j+1} \\ \hat{\underline{c}}_{j+1} \end{bmatrix} = \begin{bmatrix} \hat{\underline{a}} \\ \hat{\underline{b}}_j \\ \hat{\underline{c}}_j \end{bmatrix} + \begin{bmatrix} \hat{\underline{\alpha}}_j \\ \hat{\underline{\beta}}_j \\ \hat{\underline{\gamma}}_j \end{bmatrix}$$

where the corrections $\hat{\underline{\alpha}}$, $\hat{\underline{\beta}}$, and $\hat{\underline{\gamma}}$ are determined by

$$\begin{bmatrix} \hat{\underline{\alpha}}_j \\ \hat{\underline{\beta}}_j \\ \hat{\underline{\gamma}}_j \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} \frac{1}{m_{ij}^2} & \sum_{i=1}^{n} \frac{e^{-\hat{\underline{c}}_j t_i}}{m_{ij}^2} & \sum_{i=1}^{n} \frac{\hat{\underline{b}}_j t_i e^{-\hat{\underline{c}}_j t_i}}{m_{ij}^2} \\ \sum_{i=1}^{n} \frac{e^{-\hat{\underline{c}}_j t_i}}{m_{ij}^2} & \sum_{i=1}^{n} \frac{e^{-2\hat{\underline{c}}_j t_i}}{m_{ij}^2} & -\sum_{i=1}^{n} \frac{b_j t_i e^{-2\hat{\underline{c}}_j t_i}}{m_{ij}^2} \\ \sum_{i=1}^{n} \frac{\hat{\underline{b}}_j t_i e^{-\hat{\underline{c}}_j t_i}}{m_{ij}^2} & \sum_{i=1}^{n} \frac{\hat{\underline{b}}_j t_1 e^{-2\hat{\underline{c}}_j t_i}}{m_{ij}^2} & \sum_{i=1}^{n} \frac{\hat{\underline{b}}_j^2 t_i^2 e^{-2\hat{\underline{c}}_j t_i}}{m_{ij}^2} \end{bmatrix} \begin{bmatrix} \sum_{i=1}^{n} \frac{\underline{w}_i - y_i}{m_{ij}^2} \\ \sum_{i=1}^{n} \frac{\underline{w}_i - y_i e^{-2\hat{\underline{c}}_j t_i}}{m_{ij}^2} \\ \sum_{i=1}^{n} \frac{w_i - y_i}{m_{ij}^2} \hat{\underline{b}}_j t_1 e^{-\hat{\underline{c}}_j t_i} \end{bmatrix}$$

The weighting factor m_{ij}^2 (i.e., m^2 for the i^{th} data point and the j^{th} iteration) is in agreement with the m_i^2 equation given earlier.

$$m_{ij}^2 = \left[\hat{\underline{a}}_j - \hat{\underline{b}}_j e^{-\hat{\underline{c}}_j t_i}\right]^2 \left[E_i^2 + R_1^2 C_i^2 + R_2^2 A_i^2\right]$$

The iterative process is terminated when $\hat{\alpha}$, $\hat{\beta}$, and $\hat{\gamma}$ have become small enough to be insignificant.

After $\hat{\underline{a}}$, $\hat{\underline{b}}$, and $\hat{\underline{c}}$ have thus been determined, the unbiased estimator of s is computed.

$$\hat{\underline{s}} = \sqrt{\frac{1}{n-3} \sum_{i=1}^{n} \left(\frac{\underline{w}_i - \hat{\underline{a}} + \hat{\underline{b}} e^{-\hat{\underline{c}} t_i}}{m_i}\right)}.$$

C.2.1.1 Numerical Solution of Nonlinear Normal Equations

Two methods are explored for the solution of the nonlinear normal equations. The first method is described on pages 478-480 of Reference while the second method is Newton's method, to be found on pages 204-205 of the same reference.

The first method was found superior for the model at hand. It converged on the average after 10 iterations while the second method always took longer, usually at least twice as long. In some cases it did not converge at all where the first method had converged. The first method is therefore the one now being used in the program. These two methods are described in a general form in the following paragraphs.

Consider the problem of representing the mean of a stochastic process $\underline{w}(t)$ by some function g(t) containing k unknown parameters θ_1 to θ_k, namely

$$E\{\underline{w}(t)\} = g(t\ ;\ \theta_1,\ \ldots,\ \theta_k)$$

Estimation of these parameters is to be made from the observation $(\underline{w}_i, t_i)$ $i = 1, \ldots, n$. Both the least-squares and the maximum-likelihood methods then lead to the condition

$$\sum_{i=1}^{n} \left[\frac{\underline{w}_i - g(t_i\ ;\ \theta_1,\ \ldots,\ \theta_k)}{m_i}\right]^2 = \min$$

or more generally

$$\sum_{i=1}^{n} \left[f(\underline{w}_i, t_i\ ;\ \theta_1,\ \ldots,\ \theta_k)\right]^2 = \min$$

The realization of this condition occurs when θ_1 to θ_k assume simultaneously the values of the values of the estimators $\hat{\theta}_1$ to $\hat{\theta}_k$ which are determined from the k simultaneous equations.

$$\sum_{i=1}^{n} f\left(\underline{w}_i, t_i;\ \hat{\theta}_1,\ \ldots,\ \hat{\theta}_k\right) \cdot \frac{\partial}{\partial \theta_\ell} f\left(\underline{w}_i, t_i\ ;\ \hat{\theta}_1,\ \ldots,\ \hat{\theta}_k\right) = 0$$

$$\ell = 1, 2, \ldots, k$$

When the function f is linear in the parameters θ_1 to θ_k, then solution of this equation in closed form is straightforward. We are concerned now with cases where f is not linear in at least one of the coefficients θ_1 to θ_k.

C.2.1.2 First Method

This method involves a linearization of the function f in the form of a truncated Taylor series. Let us write

$$f\left(\underline{w}_i, t_i \; ; \hat{\theta}_1, \ldots, \hat{\theta}_k\right) = f_i(\overline{\theta})$$

where $\overline{\theta}$ is the vector of estimators, i.e.,

$$\overline{\theta} = \left(\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_k\right)$$

and let us define an incremental estimator vector

$$\overline{\delta} = \left(\hat{\delta}_1, \hat{\delta}_2, \ldots, \hat{\delta}_k\right)$$

The linearization is

$$f(\overline{\theta} + \overline{\delta}) = f_i(\overline{\theta}) + \sum_{\ell=1}^{k} \hat{\delta}_\ell \cdot \frac{\partial f(\overline{\theta})}{\partial \theta_\ell}$$

Now introduce the new function into the objective function

$$\sum_{i=1}^{n} \left[f_i(\overline{\theta} + \overline{\delta})\right]^2 = \min$$

and consider the value of $|\overline{\theta}|$ fixed while that of $\overline{\delta}$ is to be determined from this condition. It follows that

$$\sum_{i=1}^{n} f_i(\overline{\theta} + \overline{\delta}) \cdot \frac{\partial}{\partial \theta_\ell} f(\overline{\theta} + \overline{\delta}) = 0$$

$$\ell = 1, 2, \ldots, k$$

which after substitution of the linearization becomes explicitly

$$\sum_{i=1}^{n} \left[f_i(\bar{\theta}) + \hat{\delta}_1 \frac{\partial f_i(\bar{\theta})}{\partial \theta_1} + \ldots + \hat{\delta}_k \frac{\partial f_i(\bar{\theta})}{\partial \theta_k} \right] \frac{\partial f_i(\bar{\theta})}{\partial \theta_\ell} = 0$$

$$\ell = 1, 2, \ldots, k$$

Let $\bar{\theta}_j$ be the estimator before the j^{th} iteration. The j^{th} iteration yields an improvement such that

$$\bar{\theta}_{j+1} = \bar{\theta}_j + \bar{\delta}_j$$

where

$$\begin{bmatrix} \hat{\delta}_1 \\ \cdot \\ \cdot \\ \cdot \\ \hat{\delta}_k \end{bmatrix}_j = \left[a_{rs}(j) \right]^{-1} \left[b_r(j) \right]$$

and, from a solution of the normal equations

$$a_{rs}(j) = \sum_{i=1}^{n} \frac{\partial f_i(\bar{\theta}_j)}{\partial \theta_r} \cdot \frac{\partial f_i(\bar{\theta}_j)}{\partial \theta_s}$$

$$b_r(j) = \sum_{i=1}^{n} f_i(\bar{\theta}_j) \cdot \frac{\partial f_i(\bar{\theta}_j)}{\partial \theta_r}$$

Note that the matrix a_{rs} is symmetrical.

C.2.1.3 Second Method

Again taking the function

$$\sum_{i=1}^{n} f(\underline{w}_i, t_i; \hat{\theta}_1, \ldots, \hat{\theta}_k) \cdot \frac{\partial}{\partial \theta_\ell} f(\underline{w}_i, t_i ; \hat{\theta}_1, \ldots, \hat{\theta}_k) = 0$$

$$\ell = 1, 2, \ldots, k$$

a new function $F_{i\ell}(\overline{\theta})$ is defined

$$F_{i\ell}(\theta) = f_i(\overline{\theta}) \cdot \frac{\partial}{\partial\theta_\ell} f_i(\overline{\theta}) = 0$$

$$\ell = 1, 2, \ldots, k$$

This function involves the solution of k simultaneous nonlinear equations, which will be Newton's method. This again makes use of linearizing $F_{i\ell}(\overline{\theta})$ through a truncated Taylor series and provide iterative improvements on some initial estimator. Using the earlier expression for $\overline{\theta}_{j+1}$ and

$$\begin{bmatrix} \hat{\delta}_1 \\ \vdots \\ \hat{\delta}_k \end{bmatrix}_j$$

for the iteration we find

$$a_{rs}(j) = \sum_{i=1}^{n} \frac{\partial F_{ir}(\overline{\theta})}{\theta \partial_s} = \sum_{i=1}^{n} \frac{\partial F_i(\overline{\theta})}{\theta\partial_r} \cdot \frac{\partial F_i(\overline{\theta})}{\partial\theta_s} + F_i(\overline{\theta}) \frac{\partial^2 F_i(\overline{\theta})}{\partial\theta_r \partial\theta_s}$$

$$b_r(j) = \sum_{i=1}^{n} F_{ir}(\overline{\theta}) = \sum_{i=1}^{n} F_i(\overline{\theta}) \cdot \frac{\partial F_i(\overline{\theta})}{\partial\theta_r}$$

It is interesting to note that the $b_r(j)$ values are identical for the two methods.

C.2.2 METHOD II

In this approach, as values are assigned to c, denoted by $\tilde{c}$, the maximum likelihood function becomes

$$L_{max} = \prod_{i=1}^{n} \left[\frac{1}{\hat{\underline{s}}m_i\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{\underline{w}_i - \hat{\underline{a}} + \hat{\underline{b}} e^{-\tilde{c}t_i}}{\hat{\underline{s}}m_i}\right)^2} \right]$$

where $\underline{\hat{a}}$, $\underline{\hat{b}}$, $\underline{\hat{s}}$ are maximum likelihood estimators and $\tilde{c}$ is a value for c in its range of values. Since it is often more convenient to maximize the natural logarithm of L, the maximum of which coincides with that of its argument, we have instead,

$$\ln L = -\frac{1}{2} \sum_{i=1}^{n} \left(\frac{\underline{w}_i - w_i}{s m_i} \right)^2 - \frac{n}{2} \ln 2\pi - n \ln s - \sum_{i=1}^{n} \ln m_i$$

By setting the partial derivatives of ln L with respect to the three parameters a, b, s equal to zero, the three normal equations for the maximum likelihood estimators are obtained.

$$\left. \frac{\partial \ln L}{\partial a} \right|_{\substack{a=\hat{a} \\ b=\hat{b} \\ c=\tilde{c}}} = \sum_{i=1}^{n} \frac{\underline{w}_i - w_i}{m_i^2} = 0$$

$$\left. \frac{\partial \ln L}{\partial b} \right|_{\substack{a=\hat{a} \\ b=\hat{b} \\ c=\tilde{c}}} = \sum_{i=1}^{n} \left(\frac{\underline{w}_i - w_i}{m_i^2} \right) e^{-\tilde{c} t_i} = 0$$

$$\left. \frac{\partial \ln L}{\partial s} \right|_{\substack{a=\hat{a} \\ b=\hat{b} \\ c=\tilde{c}}} = \frac{1}{\hat{s}^3} \sum_{i=1}^{n} \left(\frac{\underline{w}_i - w_i}{m_i} \right)^2 - \frac{n}{\hat{s}} = 0$$

Solving the first two simultaneously yields

$$\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} \frac{1}{m_i^2} & -\sum_{i=1}^{n} \frac{e^{-\tilde{c} t_i}}{m_i^2} \\ \sum_{i=1}^{n} \frac{e^{-\tilde{c} t_i}}{m_i^2} & -\sum_{i=1}^{n} \frac{e^{-2\tilde{c} t_i}}{m_i^2} \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{n} \frac{\underline{w}_i}{m_i^2} \\ \sum_{i=1}^{n} \frac{\underline{w}_i \, e^{-\tilde{c} t_i}}{m_i^2} \end{bmatrix}$$

After $\hat{\underline{a}}$ and $\hat{\underline{b}}$ have been determined, the unbiased estimator of s, rather than the maximum likelihood estimator defined by the third normal equation is computed. It is

$$\hat{\underline{s}} = \sqrt{\frac{1}{n-2} \sum_{i=1}^{n} \left(\frac{\underline{w}_i - \hat{\underline{a}} + \hat{\underline{b}} e^{-ct_i}}{m_i} \right)^2}$$

Once the values for "a" and "b" are determined for a given c value, the value ln L is computed. This process is continued for each c in the range. The maximum ln L determines the parameters a, b, c to be used by the model.

C.2.3 PREDICTION AND PREDICTION INTERVAL

Consider pairs $(t_k, \underline{w}_k)$ which are to be predicted.

$$\underline{w}_k = w_k + \underline{e}_k$$

Since a, b, and s are not known, we use the estimators $\hat{\underline{a}}$, $\hat{\underline{b}}$, and $\hat{\underline{s}}$ to assess the confidence interval due to the error $\underline{e}_k$. The predicted value of $\underline{w}_k$ is the estimator of w_k,

$$\hat{\underline{w}}_k = \hat{\underline{a}} - \hat{\underline{b}} e^{-\tilde{c}t_k}$$

and

$$E\{\underline{w}_k\} = E\{\hat{\underline{w}}_k\} = w_k$$

Both $\hat{\underline{a}}$ and $\hat{\underline{b}}$ are linear combinations of the normally distributed random variables $\underline{w}_k$, hence $\hat{\underline{w}}_k$ is normally distributed and

$$\hat{\underline{w}}_k = w_k + \underline{e}_{w_k}$$

where $\underline{e}_{\hat{w}_k}$ is a normal error with zero mean. Let $\underline{u}_k$ be the difference between the k^{th} observation and prediction

$$\underline{u}_k = \underline{w}_k - \hat{\underline{w}}_k = \underline{e}_k - \underline{e}_{\hat{w}_k}$$

It is seen that the error $\underline{u}_k$ has two sources. One, $\underline{e}_k$, says that the observation $\underline{w}_k$ will deviate from its expected value, while secondly the estimate of the expected value contains the error $\underline{e}_{\hat{w}_k}$. The error $\underline{u}_k$ is again normal with parameters

$$E\{\underline{u}_k\} = 0$$

$$\sigma^2_{u_k} = \text{var}\{\underline{u}_k\} = \text{var}\{\underline{e}_k\} + \text{var}\{\underline{e}_{\hat{w}_k}\}$$

The last equation follows from the stipulated independence of observations. The variance of $\underline{e}_k$ follows simply from

$$\text{var}\{\underline{e}_k\} = \sigma^2 = s^2 \cdot m_k^2$$

In place of m_k^2 its estimate is used

$$\hat{\underline{m}}_k^2 = \hat{\underline{w}}_k^2 \left[E_k^2 + R_1^2 C_k^2 + R_2^2 + A_k^2 \right]$$

The variance of e_{w_k} is

$$\text{var}\{\underline{e}_{w_k}\} = \text{var}\{\hat{\underline{a}}\} + e^{-2ct_k} \text{var}\{\hat{\underline{b}}\} - 2e^{-ct_k} \text{cov}(\hat{a}, \hat{b})$$

This equation contains the covariance of $\hat{\underline{a}}$ and $\hat{\underline{b}}$ which implies that $\hat{\underline{a}}$ and $\hat{\underline{b}}$ are dependent. From the solution of the normal equations,

$$\hat{\underline{a}} = \frac{\underline{B}D - \underline{E}C}{AD - C^2}$$

$$\hat{\underline{b}} = \frac{\underline{B}C - A\underline{E}}{AD - C^2}$$

where

$$A = \sum_{i=1}^{n} \frac{1}{m_i^2}$$

$$\underline{B} = \sum_{i=1}^{n} \frac{\underline{w}_i}{m_i^2}$$

$$C = \sum_{i=1}^{n} \frac{e^{-\tilde{c}t_i}}{m_i^2}$$

$$D = \sum_{i=1}^{n} \frac{e^{-2\tilde{c}t_i}}{m_i^2}$$

$$\underline{E} = \sum_{i=1}^{n} \frac{\underline{w}_i \, e^{-\tilde{c}t_i}}{m_i^2}$$

Therefore $\hat{\underline{w}}_k$ can be expressed as a function of random variables $\underline{B}$ and $\underline{E}$

$$\hat{\underline{w}}_k = \alpha_k \underline{B} + \beta_k \underline{E}$$

where

$$\alpha_k = \frac{D - Ce^{-\tilde{c}t_k}}{AD - C^2}$$

$$\beta_k = \frac{Ae^{-\tilde{c}t_k} - C}{AD - C^2}$$

This leads to

$$\text{var}\{\underline{e}_{w_k}\} = \alpha_k^2 \,\text{var}\{\underline{B}\} + \beta_k^2 \,\text{var}\{E\} + 2\alpha_k\beta_k \,\text{cov}\{\underline{B}, \underline{E}\}$$

Since both $\underline{B}$ and $\underline{E}$ consist of a sum of independent normal variables we can write

$$\text{var}\{\underline{B}\} = \sum_{i=1}^{n} \frac{\sigma_i^2}{m_i^4} = s^2 \sum_{i=1}^{n} \frac{1}{m_i^2} = s^2 A$$

$$\text{var}\{\underline{E}\} = \sum_{i=1}^{n} \frac{e^{-2\tilde{c}t_i}}{m_i^4} \sigma_i^2 = s^2 \sum_{i=1}^{n} \frac{e^{-2\tilde{c}t_i}}{m_i^2} = s^2 D$$

The covariance of $\underline{B}$ and $\underline{E}$ is defined as

$$\text{cov}\{\underline{B},\underline{E}\} = E\left\{\left(\sum_{i=1}^{n} \frac{\underline{w}_i}{m_i^2}\right)\left(\sum_{i=1}^{n} \frac{\underline{w}_i e^{-\tilde{c}t_i}}{m_i^2}\right)\right\}$$

$$- E\left\{\sum_{i=1}^{n} \frac{\underline{w}_i}{m_i^2}\right\} E\left\{\sum_{i=1}^{n} \frac{\underline{w}_i e^{-\tilde{c}t_i}}{m_i^2}\right\}$$

This equation is simplified by the fact that, due to independence

$$E\{\underline{w}_i \, \underline{w}_j\} = E\{\underline{w}_i\} \cdot E\{\underline{w}_j\} , \quad i \neq j$$

and there results

$$\text{cov}\{\underline{B},\underline{E}\} = E\left\{\sum_{i=1}^{n} \frac{e^{-\tilde{c}t_i}}{m_i^4} \underline{w}_i^2\right\} - e^{-\tilde{c}t_i}\left[E\left\{\sum_{i=1}^{n} \frac{\underline{w}_i}{m_i^2}\right\}\right]$$

$$= \sum_{i=1}^{n} \frac{e^{-\tilde{c}t_i}}{m_i^4} \sigma^2 = s^2 C$$

Therefore

$$\text{var}\{\underline{e}_{\hat{w}_k}\} = s^2\left[\alpha_k^2 A + \beta_k^2 D + 2\alpha_k\beta_k C\right]$$

or by substitution

$$\text{var}\{\underline{e}_{\hat{w}_k}\} = \frac{s^2}{AD - C^2}\left[A\, e^{-2\tilde{c}t_k} - 2C\, e^{-\tilde{c}t_k} + D\right].$$

By comparison of this expression with the initial expression for var $\{\underline{e}_{\hat{w}_k}\}$, it is seen that

$$\text{var}\{\hat{\underline{a}}\} = s^2 \frac{D}{AD - C^2}$$

$$\text{var}\{\hat{\underline{b}}\} = s^2 \frac{A}{AD - C^2}$$

$$\text{cov}\{\hat{\underline{a}}, \hat{\underline{b}}\} = s^2 \frac{C}{AD - C^2}$$

now

$$\text{var}\{\underline{u}_k\} = s^2 \left[m_k^2 + \frac{A\, e^{-2\tilde{c}t_k} - 2C\, e^{-\tilde{c}t_k} + D}{AD - C^2} \right]$$

If s were known the prediction interval could be readily established since $\underline{u}_k$ is normally distributed. For example, for a 90 percent interval on $\underline{w}_k$ one would use

$$\hat{\underline{a}} - \hat{\underline{b}}\, e^{-\tilde{c}t_k} \pm 1.96\, \sigma_{u_k}$$

With $\hat{\underline{s}}$ available instead of s, this may still be used if the sample is large enough, say n > 30. For small samples we know that the random variable

$$\underline{t} = \frac{\dfrac{u_k}{\sigma_{u_k}}}{\sqrt{\dfrac{\hat{\underline{s}}^2}{s^2}}}$$

has the Student t distribution with n - 2 degrees of freedom, $\hat{s}$ being the unbiased estimator for s. From this follows the small sample distribution of $\underline{u}_k$

$$\underline{u}_k = \underline{t}\, \sigma_{u_k} \frac{\hat{\underline{s}}}{s} = \underline{t} \cdot \sigma_{u_k}$$

where

$$\hat{\sigma}_{\underline{u}_k} = \hat{\underline{s}}\left(m_k + \frac{A e^{-2\tilde{c}t_k} - 2C e^{-\tilde{c}t_k} + D}{AD - C^2} \right)^{\frac{1}{2}}$$

the confidence interval containing (1 - ϵ) of all possible outcomes of $\underline{w}_k$ is then

$$\hat{\underline{a}} - \hat{\underline{b}} e^{-\hat{c}t_k} \pm t_{\epsilon/2}\, \hat{\sigma}_{\underline{u}}$$

C.3 THE LOGISTICS MODEL

The computer program for this model fits a curve of the form

$$w_i = \frac{a}{1 + be^{-ct_i}}$$

through a set of observations $\underline{w}_i$ made at times t_i by the method of weighted least squares.

A range of values of c is established and for each c in the range, initial estimates of a and b in the above model are made. An iterative procedure is employed to obtain corrections, Δa_j and Δb_j for a and b respectively. When $|\Delta a_j/a|$ and $|\Delta b_j/b|$ are both less than $\epsilon = 0.001$, the iteration process is terminated. The weighted sum of the squares of the errors

$$S = \sum_{i=1}^{n} \rho_i(\underline{w}_i - w_i)^2$$

is computed and the program repeats the above process for the next value of c in the range. When all c values have been exhausted, the minimum S is determined through inspection and the corresponding values of a, b, and c are the desired parameters to be used in the model.

C.3.1 ESTIMATION OF PARAMETERS

The weighted least squares criteria for fitting a curve to data requires that we minimize

$$S = \sum_{i=1}^{n} \rho_i(\underline{w}_i - w_i)^2$$

where

$\underline{w}_i$ is the observation made at time t_i.

w_i is the prediction given by the model at time t_i.

This equation is nonlinear in the parameters a and b as are the normal equations. This means we must use an iterative procedure to determine the minimum value of S. To apply this technique S is written as

$$S = \sum_{i=1}^{n} [F_i(\underline{w}_i, t_i, a, b, c)]^2$$

and using a truncated Taylor series expansion we have

$$F_i(\underline{w}_i, t_i, a, b, c) = F_i(\underline{w}_i, t_i, a, b, c) + \frac{\partial F_i}{\partial a}\Bigg|_{\substack{\hat{a}_j \\ \hat{b}_j \\ c}} \Delta a + \frac{\partial F}{\partial b}\Bigg|_{\substack{\hat{a}_j \\ \hat{b}_j \\ c}} \Delta b$$

where $\hat{a}_j$, $\hat{b}_j$ are estimates of a, b; Δa, Δb improvements to the estimators $\hat{a}_j$, $\hat{b}_j$. In the computer program c ranges from 0.001 to 2 in step sizes of 0.001. S_{min} then is given by the iterative process

$$\begin{bmatrix} \hat{a}_{j+1} \\ \hat{b}_{j+1} \end{bmatrix} = \begin{bmatrix} \hat{a}_j \\ \hat{b}_j \end{bmatrix} + \begin{bmatrix} \Delta a_j \\ \Delta b_j \end{bmatrix}$$

where the corrections Δa_j and Δb_j are determined by

$$\begin{bmatrix} \Delta a_j \\ \Delta b_j \end{bmatrix} = - \begin{bmatrix} \sum_{i=1}^{n} \left(\frac{\partial F_i}{\partial a}\right)^2 & \sum_{i=1}^{n} \left(\frac{\partial F_i}{\partial a}\right)\left(\frac{\partial F_i}{\partial b}\right) \\ \sum_{i=1}^{n} \left(\frac{\partial F_i}{\partial a}\right)\left(\frac{\partial F_i}{\partial b}\right) & \sum_{i=1}^{n} \left(\frac{\partial F_i}{\partial b}\right)^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{n} F_i \left(\frac{\partial F_i}{\partial a}\right) \\ \sum_{i=1}^{n} F_i \left(\frac{\partial F_i}{\partial b}\right) \end{bmatrix}$$

and

$$\begin{cases} F_i = \rho_i^{\frac{1}{2}} (\underline{w}_i - w_i) \\[2ex] \dfrac{\partial F_i}{\partial a} = \rho_i^{\frac{1}{2}} \Big/ 1 + be^{-ct_i} \\[2ex] \dfrac{\partial F_i}{\partial b} = \rho_i^{\frac{1}{2}} ae^{-ct_i} \Big/ \left(1 + be^{-ct_i}\right)^2 \end{cases}$$

Substituting these into the above solution yields

$$\begin{bmatrix} \Delta a_j \\ \\ \Delta b_j \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} \dfrac{w_i^2 \rho_i}{\hat{a}_j^2} & - \sum_{i=1}^{n} \dfrac{w_i^2 e^{-ct_i} \rho_i}{\hat{a}_j \left(1 + \hat{b}_j e^{-ct_i}\right)} \\[3ex] \sum_{i=1}^{n} \dfrac{w_i^2 e^{-ct_i} \rho_i}{\hat{a}_j \left(1 + \hat{b}_j e^{-ct_i}\right)} & \sum_{i=1}^{n} \dfrac{\rho_i e^{-2ct_i} w_i^2}{\left(1 + \hat{b}_j e^{-ct_i}\right)^2} \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{n} \dfrac{\rho_i (\underline{w}_i - w_i) w_i}{\hat{a}_j} \\[3ex] \sum_{i=1}^{n} \dfrac{\rho_i (\underline{w}_i - w_i) a_i e^{-ct_i}}{\left(1 + \hat{b}_j e^{-ct_i}\right)} \end{bmatrix}$$

The iteration process is terminated when $|\Delta a_j / a_j| < 0.001$ and $|\Delta b_j / b_j| < 0.001$.

To initialize the problem first guesses of $\hat{a}_1$ and $\hat{b}_1$ are required. A least squares curve fit of the form

$$\tilde{w}_i = \tilde{a} + \tilde{b}\, t_i$$

is fitted to the observed weights.

$$\tilde{a} = \frac{\sum_{i=1}^{n} \underline{w}_i \rho_i \sum_{i=1}^{n} t_i^2 \rho_i - \sum_{i=1}^{n} t_i \rho_i \sum_{i=1}^{n} t_i \underline{w}_i \rho_i}{\sum_{i=1}^{n} \rho_i \sum_{i=1}^{n} t_i^2 \rho_i - \left[\sum_{i=1}^{n} t_i \rho_i\right]^2}$$

$$\tilde{b} = \frac{\sum_{i=1}^{n} \rho_i \sum_{i=1}^{n} \rho_i \underline{w}_i t_i - \sum_{i=1}^{n} \rho_i \underline{w}_i \sum_{i=1}^{n} \rho_i t_i}{\sum_{i=1}^{n} \rho_i \sum_{i=1}^{n} \rho_i t_i^2 - \left[\sum_{i=1}^{n} t_i \rho_i \right]^2}$$

This least squares curve fit is matched with the logistic model at two time points for a value of c = 0.1. (For c = 0.1 the logistic curve is approximately a straight line in the region where the two curves are to be matched.) The two time points chosen were $t_1 = n/4$ and $t_2 = 3n/4$. This leads to

$$\frac{\hat{a}_1}{1 + \hat{b}_1 e^{-0.1t_2}} = \tilde{a} + \tilde{b}t_1$$

$$\frac{\hat{a}_1}{1 + \hat{b}_1 e^{-0.1t_2}} = \tilde{a} + \tilde{b}t_2$$

Solving these two equations in the two unknowns $\hat{a}_1$ and $\hat{b}_1$ yields

$$\hat{b}_1 = \frac{\tilde{b}(t_1 - t_2)}{\left[e^{-0.1t_2} (\tilde{a} + \tilde{b}t_2) - e^{-0.1t_1} (\tilde{a} + \tilde{b}t_1) \right]}$$

$$\hat{a}_1 = (\tilde{a} + \tilde{b}t_1)\left(1 + \tilde{b}_1 e^{-0.1t_1}\right)$$

These values are used as a first approximation to a and b for the first value of c in the established range of c values. They are then improved upon by the iteration process discussed above. The final values of a and b for the first c value are then used as a first estimate for the next value of c in the range, etc.

Once a and b are determined for a given c, a value of S is computed. This is done for each c in the range. The minimum S determines the parameters a, b, and c to be used by the model.

C.3.2 PREDICTION AND PREDICTION INTERVAL

Unlike the linear and exponential models, the logistic model involves a quotient of the parameters a and b. This non-linear combination no longer allows one to say, even if a and b are normally distributed, that the combination is normally distributed. The following discussion resolves this difficulty by using the first values of a and b, with c = 0.1, to obtain the variance. An approximation is made, based on the number of observation values being more than ten, which yields a slight overestimate to the predicted value as a linear combination of a and b. This error is the estimate of the expected value combined with the error obtained by considering that the observation will deviate from its expected value yields a variance from which, for a specified level of confidence, a confidence interval or prediction interval can be obtained.

Defining the following quantities,

$$\underline{\alpha} = \sum_{i=1}^{n} \underline{w}_i \rho_i$$

$$\beta = \sum_{i=1}^{n} t_i^2 \rho_i$$

$$\gamma = \sum_{i=1}^{n} t_i \rho_i$$

$$\underline{\delta} = \sum_{i=1}^{n} t_i \underline{w}_i \rho_i$$

$$\epsilon = \sum_{i=1}^{n} \rho_i$$

and recalling the earlier expressions defining the initial estimates of the parameters a and b, a and b may be expressed as

$$\tilde{a} = \frac{\underline{\alpha}\beta - \gamma\underline{\delta}}{\epsilon\beta - \gamma^2}$$

$$\tilde{b} = \frac{\epsilon\underline{\delta} - \underline{\alpha}\gamma}{\epsilon\beta - \gamma^2}$$

Now, assuming the value c = 0.1 and recalling that

$$\hat{a}_1 = (\tilde{a} + \tilde{b}t_1)\left(1 + \hat{b}_1 e^{-0.1t_1}\right)$$

$$\hat{b}_1 = \frac{\tilde{b}(t_1 - t_2)}{\left[e^{-0.1t_2}(\tilde{a} + \tilde{b}t_2) - e^{-0.1t_1}(\tilde{a} + \tilde{b}t_1)\right]}$$

it follows that

$$\hat{a} \approx \hat{a}_1$$

$$\hat{b} \approx \hat{b}_1$$

Making the appropriate substitutions into the model

$$\hat{\underline{w}}_k = \frac{\hat{a}}{1 + \hat{b}e^{-0.1t_1}}$$

yields

$$\hat{\underline{w}}_k = \frac{(\tilde{a} + \tilde{b}t_1)\left[1 + \dfrac{\tilde{b}(t_1 - t_2)\,e^{-0.1t_1}}{e^{-0.1t_2}(\tilde{a} + \tilde{b}t_2) - e^{-0.1t_1}(\tilde{a} + \tilde{b}t_1)}\right]}{1 + \dfrac{\tilde{b}(t_1 - t_2)\,e^{-0.1t_1}}{\left[e^{-0.1t_2}(\tilde{a} + \tilde{b}t_2) - e^{-0.1t_1}(\tilde{a} + \tilde{b}t_1)\right]}}$$

This expression is highly nonlinear in the terms a and b. However, simplification followed by an approximation results in a linear expression as seen below.

$$\hat{\underline{w}}_k = (\tilde{a} + \tilde{b}t_1)\left[\frac{1 + \dfrac{\tilde{b}(t_1 - t_2)\left(e^{-0.1t_1} - e^{-\tilde{c}t_i}\right)}{e^{-0.1t_2}(\tilde{a} + \tilde{b}t_2) - e^{-0.1t}(\tilde{a} + \tilde{b}t_1)}}{1 + \dfrac{\tilde{b}(t_1 - t_2)\, e^{-\tilde{c}t_i}}{e^{-0.1t_2}(\tilde{a} + \tilde{b}t_2) - e^{-0.1t_1}(\tilde{a} + \tilde{b}t_1)}}\right]$$

$$= (\tilde{a} + \tilde{b}t_1)\left[1 + \frac{\tilde{b}(t_1 - t_2)\left(e^{-0.1t_1} - e^{-\tilde{c}t_i}\right)}{e^{-0.1t_2}(\tilde{a} + \tilde{b}t_2) - e^{-0.1t_1}(\tilde{a} + \tilde{b}t_1) + \tilde{b}(t_1 - t_2)\, e^{-\tilde{c}t_i}}\right]$$

$$= \left[1 + \frac{\tilde{b}(t_1 - t_2)\left(e^{-0.1t_1} - e^{-\tilde{c}t_i}\right)}{-e^{-0.1t_1}\tilde{b}\left[t_1 - e^{-0.1(t_2 - t_1)}t_2\right] + \tilde{a}\left(e^{-0.1t_2} - e^{-0.1t_1}\right) + \tilde{b}(t_1 - t_2)e^{-\tilde{c}t_i}}\right]$$

$$\cdot\ (\tilde{a} + \tilde{b}t_1)$$

For $n > 20 \ln 3$

$$\hat{\underline{w}}_k \approx (\tilde{a} + \tilde{b}t_1)\left[1 + \frac{e^{-0.1t_1} - e^{-\tilde{c}t_i}}{e^{-\tilde{c}t_i} - e^{-0.1t_1}\left\{1 - \dfrac{t_2\left[e^{-0.1(t_2 - t_1)} + 1\right]}{t_1 - t_2}\right\}}\right]$$

$$\approx (\tilde{a} + \tilde{b}t_1)\left[1 + \frac{1 - e^{-\tilde{c}t_i + 0.1t_1}}{e^{-\tilde{c}t_i + 0.1t_1} - 1 + \dfrac{t\left(e^{-0.1\frac{n}{2}} + 1\right)}{\frac{n}{2}}}\right]$$

$$\hat{\underline{w}}_k \approx (\tilde{a} + \tilde{b}t_1)\left[1 + \frac{1 - e^{-\tilde{c}t_i + 0.1t_1}}{e^{-\tilde{c}t_i + 0.1t_1} - 1 + \frac{3n}{4} \cdot \frac{2}{n}}\right]$$

$$\approx (\tilde{a} + \tilde{b}t_1)\left[1 + \frac{1 - e^{-\tilde{c}t_i + 0.1t_1}}{e^{-\tilde{c}t_i + 0.1t_1} - 0.5}\right]$$

and note that $\tilde{c} = 0.1$, $t_1 = n/4$, $t_2 = 3n/4$.

$$\approx (\tilde{a} + \tilde{b}t_1)\left[1 - 7.\dot{5}\dot{7}\left(e^{-\tilde{c}t_i + 0.1t_1} - 1\right)\right]$$

$$\approx \left(\tilde{a} + \frac{n\tilde{b}}{4}\right)\left[1 - 7.\dot{5}\dot{7}\left(e^{-0.1t_i + \frac{n}{40}} - 1\right)\right]$$

Now let

$$k_{t_i} = 1 - 7.\dot{5}\dot{7}\left(e^{-0.1t_i + \frac{n}{40}} - 1\right)$$

and applying the earlier expressions for $\tilde{a}$ and $\tilde{b}$

$$\tilde{a} + \frac{n\tilde{b}}{4} = \frac{\underline{\alpha}\beta - \gamma\underline{\delta}}{\epsilon\beta - \gamma^2} + \frac{\epsilon\underline{\delta} - \underline{\alpha}\gamma}{\epsilon\beta - \gamma^2} \cdot \frac{n}{4}$$

$$= \frac{\beta - \frac{n\gamma}{4}}{\epsilon\beta - \gamma^2}\,\underline{\alpha} + \frac{\frac{n\epsilon}{4} - \gamma}{\epsilon\beta - \gamma^2}\,\underline{\delta}$$

$$= A\underline{\alpha} + B\underline{\delta}$$

where

$$A = \frac{\beta - \frac{n\gamma}{4}}{\epsilon\beta - \gamma^2}$$

$$B = \frac{\frac{n\epsilon}{4} - \gamma}{\epsilon\beta - \gamma^2}$$

Consequently

$$\text{var}\{\overline{w}_i\} = E\left\{(A\underline{\alpha} + B\underline{\delta})^2 k_{t_i}^2\right\} - E^2\left\{(A\underline{\alpha} + B\underline{\delta}) k_{t_i}\right\}$$

$$= \left[A^2k_{t_i}^2 E\{\underline{\alpha}^2\} + B^2k_{t_i}^2 E\{\underline{\delta}^2\} + 2ABk_{t_i}^2 E\{\underline{\alpha}\,\underline{\delta}\}\right]$$

$$- \left[A^2k_{t_i}^2E^2\{\underline{\alpha}\} + B^2k_{t_i}^2E^2\{\underline{\delta}\} + 2ABk_{t_i}^2E\{\underline{\alpha}\}\,E\{\underline{\delta}\}\right]$$

$$= A^2k_{t_i}^2 \text{ var }\{\underline{\alpha}\} + B^2k_{t_i}^2 \text{ var }\{\underline{\delta}\} + 2ABk_{t_i}^2 \text{ cov }\{\underline{\alpha}\,\underline{\delta}\}$$

Using the following expression for the observation range weighting factor from

$$\rho_i = \alpha^{n-i}(E_i + R_1C_i + R_2A_i)$$

and also

$$S = \sum_{i=1}^{n} \rho_i(w_i - \overline{w}_i)^2$$

Let

$$\tilde{S} = \frac{S}{n - 2}$$

so that

$$\text{var}\{\underline{\alpha}\} = \tilde{S}\epsilon$$

$$\mathrm{var}\,\{\underline{\delta}\} = \tilde{S}\beta$$

$$\mathrm{cov}\,\{\underline{\alpha}, \underline{\delta}\} = \tilde{S}\gamma$$

thus

$$\mathrm{var}\,\{\underline{\hat{w}}_k\} = A^2 k_{t_i}^2 \tilde{S}\epsilon + B^2 k_{t_i}^2 \tilde{S}\beta + 2AB k_{t_i}^2 \tilde{S}\gamma$$

$$= k_{t_i}^2 \tilde{S}(A^2\epsilon + B^2\beta + 2AB\gamma)$$

$$= k_{t_i}^2 \cdot \tilde{S}\left[\left(\frac{\beta - \frac{n\gamma}{4}}{\epsilon\beta - \gamma^2}\right)^2 \epsilon + \left(\frac{\frac{n\epsilon}{4} - \gamma}{\epsilon\beta - \gamma^2}\right)^2 \beta + 2\gamma\left(\frac{\beta - \frac{n\gamma}{4}}{\epsilon\beta - \gamma^2}\right)\left(\frac{\frac{n\epsilon}{4} - \gamma}{\epsilon\beta - \gamma^2}\right)\right]$$

Using

$$A = \left(\frac{\beta - \frac{n\gamma}{4}}{\epsilon\beta - \gamma^2}\right)$$

and

$$B = \left(\frac{\frac{n\epsilon}{4} - \gamma}{\epsilon\beta - \gamma^2}\right)$$

$$\mathrm{var}\,\{\underline{\hat{w}}_k\} = k_{t_i}^2 \cdot \tilde{S}\left[\frac{\left(\beta^2 - \frac{n\beta\gamma}{2} + \frac{n^2\gamma^2}{16}\right)\epsilon + \left(\frac{n^2\epsilon^2}{16} - \frac{n\epsilon\gamma}{2} + \gamma^2\right)\beta + 2\gamma\left(\frac{n\beta\epsilon}{4} - \frac{n^2\gamma\epsilon}{16} - \gamma\beta + \frac{n\gamma^2}{4}\right)}{(\epsilon\beta - \gamma^2)^2}\right]$$

$$= k_{t_i}^2 \cdot \tilde{S}\left[\frac{\beta^2\epsilon - \frac{n\beta\gamma\epsilon}{2} + \frac{n^2\gamma^2\epsilon}{16} + \frac{n^2\beta\epsilon}{16} - \frac{n\beta\gamma\epsilon}{2} + \beta\gamma^2 + \frac{n\beta\gamma\epsilon}{2} - \frac{n^2\gamma^2\epsilon}{8} - 2\beta\gamma^2 + \frac{n\gamma^2}{2}}{(\epsilon\beta - \gamma^2)^2}\right]$$

$$= k_{t_i}^2 \cdot \tilde{S}\left[\frac{(\beta\epsilon^2 - \gamma^2\epsilon)\frac{n^2}{16} + (\gamma^3 - \beta\gamma\epsilon)\frac{n}{2} + (\beta^2\epsilon - \beta\gamma^2)}{(\epsilon\beta - \gamma^2)^2}\right]$$

$$= k_{t_i}^2 \cdot \tilde{S}\left(\frac{\frac{\epsilon n^2}{16} - \frac{\gamma n}{2} + \beta}{\epsilon\beta - \gamma^2}\right)$$

Since it is assumed that

$$\underline{w}_i = w_i + \underline{e}_i$$

where $\underline{e}_i$ is an unobservable normal error having zero mean and standard deviation σ_i

$$\underline{w}_i = E_i \underline{w}_i + C_i \underline{w}_i + A_i \underline{w}_i$$

Then each of the three parts is broken into its mean plus an error

$$\underline{w}_i = \left(w_{E_i} + \underline{e}_{E_i}\right) + \left(w_{C_i} + \underline{e}_{C_i}\right) + \left(w_{A_i} + \underline{e}_{A_i}\right)$$

and the assumption

$$w_{E_i} = E_i w_i$$

$$w_{C_i} = C_i w_i$$

$$w_{A_i} = A_i w_i$$

It follows that

$$w_i = w_{E_i} + w_{C_i} + w_{A_i}$$

$$\underline{e}_i = \underline{e}_{E_i} + \underline{e}_{C_i} + \underline{e}_{A_i}$$

These three errors are assumed normally and independently distributed with zero mean and standard deviations σ_{E_i}, σ_{C_i}, σ_{A_i}.

It is further assumed that the ratio of the standard deviation to the mean of the three random weights is the constant s. Then the variance of the i^{th} observation may be expressed as

$$\sigma_i^2 = s^2 \cdot \rho_i^2 \cdot w_i^2$$

Let $\underline{u}_k$ be the difference between the k^{th} observation and prediction

$$\underline{u}_k = \underline{w}_k - \hat{\underline{w}}_k = \underline{e}_k - \underline{e}_{\hat{w}_k}$$

where $\underline{e}_{\hat{w}_k}$ is the variance of $\hat{w}_k$ and $\underline{e}_k$ is the error related to our observation σ_i^2. Note that the error $\underline{u}_k$ is again normally distributed with mean zero and

$$\text{var}\ \{\underline{u}_k\} = \text{var}\ \{\underline{e}_k\} + \text{var}\ \{\underline{e}_{\hat{w}_k}\}$$

$$= s^2 \rho_i \hat{w}_k^2 + k_{t_i}^2 \tilde{S} \left(\frac{\frac{\epsilon n^2}{16} - \frac{\gamma n}{2} + \beta}{\epsilon\beta - \gamma^2} \right)$$

So the upper confidence limit for the i^{th} observation

$$\hat{w}_k + \frac{t_p}{2} \cdot \sqrt{s^2 \rho_i \hat{w}_k^2 + k_{t_i}^2 \cdot \tilde{S} \cdot \left(\frac{\frac{\epsilon n^2}{16} - \frac{\gamma n}{2} + \beta}{\epsilon\beta - \gamma^2} \right)}$$

for Student t variable $\underline{t}$, where p is the percent confidence limit desired and n - 2 degrees of freedom.

C.4 ADAPTIVE (FOURIER) EXPONENTIAL MODEL

The models discussed in paragraphs C.1, C.3, and C.4 are based on the assumption that the underlying process generating varying monthly weight estimates can be approximated by an analytic man value function. In this paragraph a model is discussed which, rather than assuming such a function, senses the latest tendencies of the trend and is therefore adaptive. It reflects only immediate past history, treating the weight increment as a random variable. In the simplest case the expected value of the increment is constant, that is, with w_i being the i^{th} observation we have

$$w_{i+1} - w_i = a + R_i$$

where a is the mean increment and R_i is a random residual of zero mean. Going one step further one many assume that the next increment also depends on what has happened in the past, specifically one may make it proportional to the last increment, namely,

$$w_{i+1} - w_i = a + b(w_i - w_{i-1}) + R_i$$

This equation contains a linear autoregressive term $b(w_i - w_{i-1})$, where be is a parameter; it represents a mixed autoregressive model. This equation, without the residual term, is also a difference equation and has the solution

$$w_{i+1} - w_i = cb^i + \frac{a}{1 - b}$$

where the constant c depends upon the initial condition w_1. Returning to the original process, and provided that observations are equally spaced in time, the solution becomes

$$w_i = w_1 + cb \frac{1 - b^{i-1}}{1 - b} + a \frac{i - 1}{1 - b}$$

The only acceptable case of this expression in the weight trend problem is that where $0 \leq b \leq 1$; in that case it represents a straight line with a superimposed decaying exponential.

The application of the simple and plausible argument that the forces at work in the weight evolution process, and the increments which they cause from one observation to the next, are on the average constant, but also proportional to the latest increment, has resulted in the above model. This model has some desirable properties. It is simple. Its behavior is asymptotically linear, but it allows for perturbations which change the trend. Thus it is adaptive and a suitable model for processes which behave linearly in some time interval, but whose parameters are subject to slow variations with time.

Unfortunately there is one shortcoming. Experimentation with estimation of the parameters a and b from data which were generated in accordance with the model revealed that with 20 to 30 observations available the estimators for a and b were not consistently good, and the resulting prediction was not only inferior to the conventional linear regressive prediction, but sometimes outright wrong. Two factors compound the difficulty. Firstly, using increments, which is analogous to differentiation, increases the relative magnitude of the residuals; the effect of noise becomes more pronounced. Secondly, if there is little exponential trend in the data, which means that b is close to one, then separation of linear and exponential trend becomes difficult. For with $b = 1$, the functions

$$w_{i+1} - w_i = a$$

and

$$w_{i+1} - w_i = b(w_i - w_{i-1})$$

are linearly independent and there exists then an infinity of solutions. To overcome these difficulties we decided to use a model which, without the residual, has the same time dependent behavior as this model, but which is not autoregressive in its structure. This model is

$$w_{i+1} - w_i = a + be^{-\alpha t_i} + R_i$$

Here t_i denotes the time of the i^{th} observation. When observations are equally spaced, namely $t_{i+1} = t_i + \Delta t$, there follows

$$w_{i+1} = (w_1 - c) + ia + ce^{\alpha t_i} + \sum_{j=1}^{i} R_j$$

where

$$c = b(1 - e^{\alpha \Delta t})^{-1}$$

This equation contains the constant term $(w_1 - c)$, the linear term ia, and the exponentially decaying term $ce^{-\alpha t_i}$. If straightforward maximum likelihood techniques were used to estimate the parameters a, c, and , then one would again encounter the problem mentioned before; the functions ia and $ce^{-\alpha t_i}$ may be almost linearly dependent, which means that there are infinitely many combinations of these functions which will make a good fit.

As a consequence one may obtain estimators which make little sense. For this reason it was decided to first separate the nonlinear part by a Fourier analysis and then to treat the remainder as the increment process described by the very first equation.

C.4.1 FOURIER SMOOTHING

A transformation of variable is made

$$X_i = w_i - \frac{t_n - t_i}{t_n - t_1} w_1 - \frac{t_i - t_1}{t_n - t_1} w_n$$

This transformation is such that $X_1 = 0$, $X_n = 0$. The functional behavior of X_i $(1 \le i \le n)$ is represented exactly by the following Fourier series

$$X_i = B_1 \sin \pi \frac{t_i - t_1}{t_n - t_1} + B_2 \sin 2\pi \frac{t_i - t_1}{t_n - t_1} + \ldots + B_{n-2} \sin (N-2)\pi \frac{t_i - t_1}{t_n - t_1}$$

The Fourier coefficients B_k are

$$B_k = \frac{2}{n-1} \sum_{i=2}^{n-1} X_i \sin k\pi \frac{t_i - t_1}{t_n - t_1}, \qquad k = 1, 2, \ldots, n-2$$

A smoothed function $\widetilde{X}_i$ through the data X_i is obtained by truncating the X_i series and by adjusting the coefficients B_k, namely

$$\widetilde{X}_i = \sum_{i=1}^{4} B_k' \sin k\pi \frac{t_i - t_1}{t_n - t_1}$$

The truncation serves to suppress noise. The $\widetilde{X}_i$ series contains four terms. This number of terms was arrived at empirically by experimentation. The adjustment of the coefficients serves to obtain a smoothed function which does not have any points of inflection. This adjustment is achieved by comparing the coefficients B_k to the Fourier coefficients $\overline{A}_k$ of an exponential whose decay rate α is such that

$$e^{-\alpha(t_n - t_1)} = 0.1$$

where the value 0.1 was selected empirically. The rule for adjustment is

$$B_1' = B_1$$

$$B_k' = \begin{cases} B_k & \text{if} \quad 0 \le B_k/B_1 \le \overline{A}_k/\overline{A}_1 \\ B_1\overline{A}_k/\overline{A}_1 & \text{if} \quad B_k/B_1 > \overline{A}_k/\overline{A}_1 \\ 0 & \text{if} \quad B_k/B_1 < 0 \end{cases}, \quad k = 2, 3, 4, 5$$

If the first Fourier coefficient B_1 is less than 5 percent of the increase of w_i in the interval (ℓ, n) then it is assumed that w_i contains no significant nonlinear trend and the smoothed function is

$$\tilde{X}_i = 0 \text{ , if } \left| \frac{B_1}{w_n - w_1} \right| \leq 0.05$$

The smoothed function $\tilde{w}_i$ through the normalized observations follows from reversing the variable transformation

$$\tilde{w}_i = w_1 + (w_n - w_1) \frac{t_i - t_1}{t_n - t_1} + \tilde{X}_i$$

C.4.2 SEPARATION OF THE EXPONENTIAL CONTENT

The model assumes that the nonlinearity present in the data be of exponential form. The next step is to find that exponential which best matches the nonlinearity $\tilde{X}_i$. The matching is very conveniently done in the spectral domain, using the least squares error as criterion. The Fourier spectrum of $\tilde{X}_i$ is compared with the spectra of 13 functions, having the form

$$f_\ell(t) = a + bt + e^{-\alpha_\ell (t - t_1)}$$

where the range of the decay constants α_ℓ is selected empirically such that the min $(\alpha_\ell) = \alpha_1$, max $(\alpha_\ell) = \alpha_{13}$, and

$$e^{-\alpha_1 (t_n - t_1)} = 0.7$$

$$e^{-\alpha_{13} (t_n - t_1)} = 0.1$$

$$e^{-\alpha_\ell (t_n - t_1)} = 0.75 - 0.05\ell$$

The spectrum corresponding to $f_\ell(t)$ is

$$A_k(\alpha_\ell) = \frac{2}{k\pi} \cdot \left\{ \frac{(-1)^k e^{-\alpha_\ell (t_n - t_1)} - 1}{1 + \left[\frac{k\pi}{(t_n - t_1)\alpha_\ell} \right]^2} \right\} , \quad \begin{array}{l} k = 1, 2, \ldots, 5 \\ \ell = 1, 2, \ldots, 13 \end{array}$$

We consider now four possible combinations of nonlinearity (see Figure C-3).

Case 1 - B_1 is negative and $w_n - w_1$ is positive.

Case 2 - B_1 is positive and $w_n - w_1$ is positive.

Case 3 - B_1 is positive and $w_n - w_1$ is negative.

Case 4 - B_1 is negative and $w_n - w_1$ is negative.

Cases 1 and 3 are rejected because an extrapolation of their behavior does not agree with the physical picture of weight growth. It is then assumed that no nonlinearity exists, that is, $\tilde{X}_i = 0$.

In Cases 3 and 4 the comparison of the B_k' and the $A_k(\alpha_\ell)$ is such that one wants to find the value α' where $\alpha_1 \leq \alpha' \leq \alpha_{13}$, which produces the least squares error. The criterion for the error as expressed by the two spectra is $E(\alpha_\ell)$

$$E(\alpha_\ell) = \sum_{k=1}^{5} [C_\ell A_k(\alpha_\ell)]^2 - \sum_{k=1}^{5} C_\ell A_k(\alpha_\ell) B_k', \quad \ell = 1, 2, \ldots, 13$$

where the comparative magnitude of the exponentials, C_ℓ, is determined from empirical considerations, namely

$$C_\ell = -0.8 \frac{w_n - w_1}{1 - e^{-\alpha_\ell(t_n - t_1)}} \quad \text{in Case 2}$$

$$C_\ell = - \frac{w_n - w_1}{1 - e^{\alpha_\ell(t_n - t_1)}} \quad \text{in Case 4}$$

The value α' has the property

$$E(\alpha') = \min E(\alpha) \quad \alpha_1 \leq \alpha \leq \alpha_{13}$$

The exponent with decay constant α' gives the best fit. We first find α_j such that

$$E(\alpha_j) = \min E(\alpha_\ell) \quad \ell = 1, 2, \ldots, 13$$

If α_j coincides with one of the end points, namely $\alpha_j = \alpha_1$ or $\alpha_j = \alpha_{13}$, then $\alpha' = \alpha_j$. Otherwise α' is determined by interpolation with a parabolic arc

$$\alpha' = - \frac{b}{2c} + \alpha_{j-1}$$

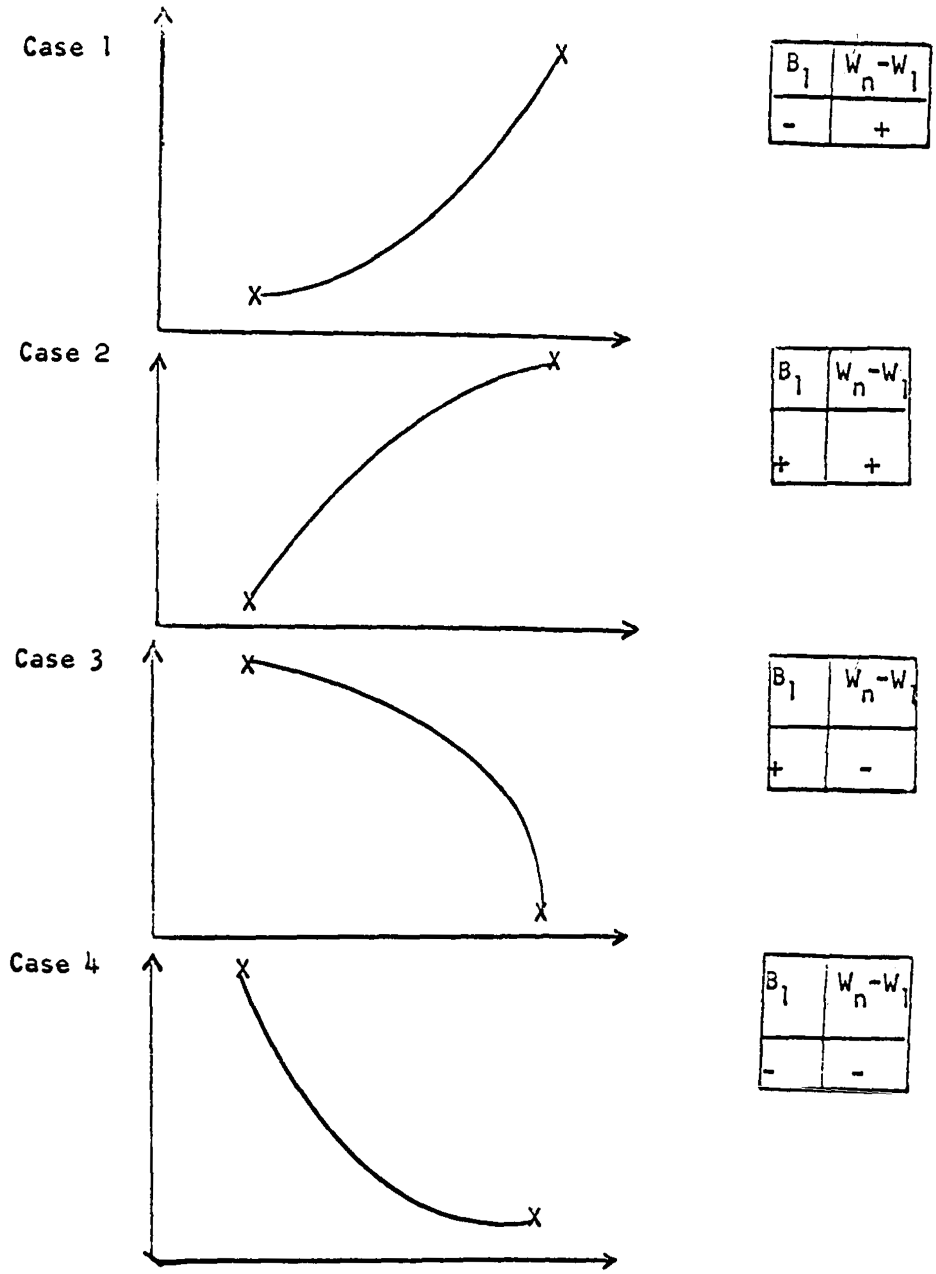


Figure C-3. Possible Combinations of Nonlinearity

where

$$\begin{bmatrix} b \\ c \end{bmatrix} = \begin{bmatrix} (\alpha_j - \alpha_{j-1}) & (\alpha_j - \alpha_{j-1})^2 \\ (\alpha_{j+1} - \alpha_{j-1}) & (\alpha_{j+1} - \alpha_{j-1})^2 \end{bmatrix}^{-1} \begin{bmatrix} E(\alpha_j) & - E(\alpha_{j-1}) \\ E(\alpha_{j+1}) & - E(\alpha_{j-1}) \end{bmatrix}$$

Having found α', the corresponding Fourier coefficients are obtained and the magnitude c' of the exponential content is

$$c' = \frac{\sum_{k=1}^{5} A_k(\alpha') B_k'}{\sum_{k=1}^{5} A_k^2(\alpha')}$$

There is one restriction in Case 4, which is to avoid crossing into negative weights in the prediction range. Any index ℓ when it has the property $C_\ell > 0.8\ w_1$ is considered a forbidden region. If there is a forbidden region at all then it will involve an index set $\ell = 1, 2, \ldots, f$. If $f = 13$ then nonlinearity must be rejected and we assume again $\tilde{X}_i = 0$. If $1 \le f < 13$, and α_j falls in the forbidden region, including $\alpha_j = \alpha_f$, then we substitute $\alpha' = \alpha f$.

The exponential trend contained in the data w_i has now been identified and is subtracted from the data. There remains the corrected data Z_i,

$$Z_i = w_i - c' e^{-\alpha'(t_i - t_1)}$$

In the cases where nonlinearity was rejected, i.e., where $\tilde{X}_i = 0$, we have $c' = 0$, and $Z_i = w_i$.

C.4.3 ESTIMATION OF PARAMETERS AND PREDICTION

A linear fit is made through the Z_i based on increments, including decreasing weighting. The increments are

$$Y_i = Z_i - Z_{i-1}, \quad i = 2, 3, \ldots, n$$

The model is

$$Y_i = a + R_i$$

where R_i is a normal residual having zero mean. The estimator $\hat{a}$ is determined from

$$\hat{a} = \frac{\sum_{i=2}^{n} \rho^{t_n - t_i} Y_i}{\sum_{i=2}^{n} \rho^{t_n - t_i}}$$

where the weighting coefficient ρ is defined by $\rho^{n-1} = k$ and k is given as input to the computer program. If k is not specified the program uses $\rho = k = 1$. The estimator $\hat{\sigma}^2$ for the variance of Y_i is

$$\hat{\sigma}^2 = \frac{1}{n - 2} \sum_{i=2}^{n} \rho^{t_n - t_i} (Y_i - \hat{a})^2$$

In making the prediction it is necessary to distinguish whether the estimator $\hat{a}$ is positive or negative. In the latter case the monthly increments are reduced exponentially in the prediction range, in order to prevent crossing into negative weights. There follows

$$w_m = Z_n + (t_m - t_n)\hat{a} + c' e^{-\alpha'(t_m - t_1)} , \quad \hat{a} > 0 , \ m > n$$

or

$$w_m = \beta^{(t_m - t_n)} Z_n + c' e^{-\alpha(t_m - t_1)} , \quad \hat{a} < 0 , \ m > n$$

where

$$\beta = \frac{Z_n + \hat{a}}{Z_n}$$

The upper 95 percent confidence limit G is determined from

$$G_m = w_m + 1.64 \sqrt{(t_m - t_n)\, \hat{\sigma}^2}$$

C.4.4 FOURIER ANALYSIS OF A FUNCTION CONTAINING AN EXPONENTIAL TERM

Consider the function

$$f(x) = a + bx + e^{-\alpha x}$$

in which a and b are arbitrary constants while α is positive. The domain of f(x) is the interval (0, L). We wish to represent the nonlinear part of f(x) by a rapidly converging Fourier series. We first subtract from f(x) a linear function such that

$$g(x) = f(x) - a' - b'x$$

and

$$g(0) = g(L) = 0$$

There follows

$$a' = 1 + a$$

$$b' = \frac{1}{L}(e^{-\alpha L} - 1) + b$$

$$g(x) = -1 - \frac{1}{L}(e^{-\alpha L} - 1)x + e^{-\alpha x}, \quad 0 \le x \le L$$

The Fourier analysis is applied to data whose first and last points go through zero by virtue of removing a linear function from the original data. In order that the first derivative be a continuous function, it is hypothesized that the smoothed curve fit to the data be a periodic sine-type function, $g(x + L) = - g(x)$ with the interval where the data fall representing a half-period. It is important that a function with a continuous first derivative is used as the terms of a Fourier series expansion decay as n^{k+1} where k is the highest order derivative that is continuous. The Fourier series representation of this sine-type function contains only sine terms, thus

$$g(x) = \sum_{k=1}^{\infty} A_k \sin \frac{k\pi}{L} x$$

with

$$A_k = \frac{2}{L} \int_0^L g(x) \sin \frac{k\pi}{L} x \, dx$$

The following integrals are involved in evaluating this expression

$$\int_0^L e^{-\alpha x} \sin \beta x dx = \frac{\beta[1 - (-1)^k e^{-\alpha L}]}{\alpha^2 + \beta^2}, \quad \beta = \frac{k\pi}{L}$$

$$\int_0^L \sin \beta x dx = \frac{1}{\beta}[1 - (-1)^k]$$

$$\int_0^L x \sin \beta x dx = \frac{L}{\beta}(-1)^k$$

The resulting Fourier coefficients are

$$A_k = \frac{2}{k\pi}\left[\frac{1}{1 + \frac{\beta}{\alpha}^2}\right] [(-1)^k e^{-\alpha L} - 1]$$

C.4.5 LEAST-SQUARES APPROXIMATION OF AN EMPIRICAL FUNCTION BY ANOTHER FUNCTION

Consider the case where we have a function f(x), defined on the interval (0, L), which was obtained from smoothing of empirical data with a truncated Fourier series

$$f(x) = \sum_{i=1}^{n} B_i \sin \frac{i\pi}{L} x, \quad 0 \le x \le L$$

To approximate f(x) by some known function $g(x;\alpha)$ which is also given in its Fourier series representation

$$g(x;\alpha) = \sum_{k=1}^{\infty} A_k(\alpha) \sin \frac{k\pi}{L} x$$

The approximation is of the form

$$f(x) \approx Cg(x;\alpha)$$

where the magnitude C and the parameter α may be varied in order to obtain the best approximation. The least-squares criterion requires that

$$E = \int_0^L [f(x) - Cg(x;\alpha)]^2 \, , \; dx = \text{minimum}$$

and there follow, from differentiation with respect to the parameters, the simultaneous equations

$$\int_0^L [f(x) - Cg(x;\alpha)] \, g(x;\alpha)dx = 0$$

and

$$\int_0^L [f(x) - Cg(x;\alpha)] \frac{\partial g(x;\alpha)}{\partial \alpha}$$

Rather than solving these two equations simultaneously the approach will be taken of first holding α fixed, solving the first equation, then varying α and locating the minimum of E in some way. The first equation is rewritten as

$$\int_0^L f(x) \, g(x;\alpha)dx = C \int_0^L g(x;\alpha) \, g(x;\alpha)dx$$

In virtue of the orthogonality properties of the Fourier series we have

$$\sum_{i=1}^{n} A_i(\alpha) \, B_i = C \sum_{k=1}^{\infty} A_k^2(\alpha)$$

and the factor C which provides the best approximation in the sense of least squares is

$$C = \frac{\sum_{i=1}^{n} A_i(\alpha) \, B_i}{\sum_{k=1}^{\infty} A_k^2(\alpha)}$$

For practical purposes the sum in the denominator cannot be carried to infinity. The question arises how many of the coefficients A_k are needed. One feels intuitively that all coefficients with $k > n$ should be omitted, since in the Fourier series terms of order greater than n cannot contribute anything to the function f(x). It will now be shown that the integrated square error E is indeed smallest if the following expression for C is used:

$$C = \frac{\sum_{i=1}^{n} A_i(\alpha)\, B_i}{\sum_{i=1}^{n} A_k^2(\alpha)}$$

Expanding E

$$E = \int_0^L \{[f(x)]^2 - 2Cf(x)\, g(x;\alpha) + [Cg(x;\alpha)]^2\}dx$$

substituting

$$E = \int_0^L [f(x)]^2 dx - C^2 \int_0^L [g(x;\alpha)]^2 dx$$

$$\frac{2E}{L} = \sum_{i=1}^{n} B_i^2 - C^2 \sum_{k=1}^{\infty} A_k^2(\alpha)$$

$$\frac{2E}{L} = \sum_{i=1}^{n} B_i^2 - \frac{\left(\sum_{i=1}^{n} A_i(\alpha)\, B_i\right)^2}{\sum_{k=1}^{\infty} A_k^2(\alpha)}$$

The error thus expressed implies that the Fourier series of the approximating function $g(x;\alpha)$ is carried to an infinity of terms.

Since

$$\sum_{k=1}^{m} A_k^{\,2}(\alpha) < \sum_{k=1}^{\infty} A_k^{\,2}(\alpha)$$

It is seen that if $g(x;\alpha)$ is represented only by the first m terms of its Fourier series, where $m < n$, then E will be smaller. The desired conclusion is now at hand. The approximating function for f(x) which produces the best approximation in the sense of least squares is

$$Cg(x;\alpha) = \sum_{k=1}^{n} A_k(\alpha) \sin \frac{k\pi}{L} x$$

where C is is determined from the second expression for C. The integrated error square is

$$\frac{2E}{L} = \sum_{i=1}^{n} B_i^{\,2} - \frac{\left(\sum_{i=1}^{n} A_i(\alpha)\, B_i \right)^2}{\sum_{k=1}^{n} A_k^{\,2}(\alpha)}$$

The optimal value of the parameter is the one which minimizes E. This value may have to be determined by searching methods rather than analytically.

C.5 NOMENCLATURE

a, b, c	Model parameters.
$\hat{a}$, $\hat{b}$, $\hat{c}$	Maximum likelihood estimators of the model parameters.
cov { }	Covariance of two random variables in the argument { }.
e	Unobservable error between observed value and mean value provided by model.
E, C, A	Proportions of observed value which are Estimated, Calculated, and Actual.
E{ }	Expected value of the argument { }.
L	Likelihood function.
m	Weighting factor based on E, C, and A.
n	Number of observations.

R_1, R_2	Weighting terms defining relationship between s values for the E, C, and A weight components.
s	Constant ratio of standard deviation to weight components (i.e., E, C, and A components).
$\hat{s}$	Maximum likelihood estimator of s.
t	Time.
$\underline{t}$	Parameter of Students t distribution.
var { }	Variance of the random variable in the argument { }.
w	Weight.
$\underline{\hat{w}}$	Predicted value of weight.
$\underline{w}$	Observed value of weight.
σ	Standard deviation.

<u>Subscripts</u>

i	The value of i denotes a specific observed data point.
k	The value of k denotes a specific predicted point.

APPENDIX D

PERFORMANCE RELATIONSHIPS

D.1 INTRODUCTION

When drag, angle of attack, and gravity losses do not change significantly as a vehicle's weight is perturbed, the impulse velocity rocket equation can be used with good accuracy to determine the effect of vehicle perturbations on stage dry weight, propellant loading, and specific impulse. This is the case with the Apollo Spacecraft during earth orbit, translunar injection, and the remainder of its space flight. When applied to the Saturn/Apollo launch vehicle this equation yields significant errors due to the changing velocity losses. In either case, the velocity increment used in the impulsive equation must include all velocity losses as well as the actual change in velocity.

Using this equation, trade-off factors can be found as the ratio of total derivatives. Partial derivatives are evaluated from known mass ratios, ideal velocity increments, and specific impulses. Control values are used to evaluate these derivatives. Appropriate assumptions must be made to eliminate or evaluate total derivatives not directly entering into the trade-off factor. These assumptions can include such constraints as constant propellant loading or a given ratio of stage propellant loadings, no change in ideal impulsive velocity or specific impulse, constant liftoff weight, etc.

The following paragraphs show the derivation of the generalized equations for the impulsive velocity changes of an N-stage rocket vehicle in terms of the changes in stage specific impulse, stage dry weight, and stage propellant loading. Expressions are presented for the velocity change of:

a. the k^{th} stage.

b. the sum of the first K stages.

c. the sum of all N stages.

D.2 DERIVATION OF EQUATIONS

Consider the impulsive velocity change rocket equation

$$\Delta V = I_{sp}\, g \ln \frac{W_1}{W_2} \tag{1}$$

where

ΔV = ideal velocity change (fps)

I_{sp} = specific impulse (sec)

g = acceleration due to gravity (ft/sec^2)

W_1 = weight at beginning of ΔV (lbs)

W_2 = weight at end of ΔV (lbs)

The weight at the start of the impulsive velocity change can be thought of as made up of "dry" weight (W_D) and propellant weight (W_P), where the "dry" weight is the total vehicle weight less the propellant weight burned to achieve the velocity change.

$$W_2 = W_D \tag{2a}$$

and

$$W_1 = W_D + W_P \tag{2b}$$

Equation (1) can then be written

$$\Delta V = I_{sp}\, g \ln\left(\frac{W_D + W_P}{W_D}\right) \tag{3}$$

The total differential of velocity change is then

$$d(\Delta V) = \frac{\partial(\Delta V)}{\partial I_{sp}}\, d\, I_{sp} + \frac{\partial(\Delta V)}{\partial W_D}\, d\, W_D + \frac{\partial(\Delta V)}{\partial W_P}\, d\, W_p \tag{4}$$

From Equation (3)

$$\frac{\partial(\Delta V)}{\partial I_{sp}} = g \ln\left(\frac{W_D + W_P}{W_D}\right) \tag{5a}$$

$$\frac{\partial(\Delta V)}{\partial W_P} = \frac{I_{sp}\, g}{(W_D + W_P)} \tag{5b}$$

and

$$\frac{\partial(\Delta V)}{\partial W_D} \quad \frac{-I_{sp}\, g\left(\frac{W_P}{W_D}\right)}{(W_D + W_P)} \tag{5c}$$

Substituting the preceding partial derivatives into Equation (4)

$$d(\Delta V) = \left[g \ln\left(\frac{W_D + W_P}{W_D}\right)\right] d\,I_{sp} + \left(\frac{g\,I_{sp}}{W_D + W_P}\right) d\,W_P$$

$$- \left[\left(\frac{g\,I_{sp}}{W_D + W_P}\right)\left(\frac{W_P}{W_D}\right)\right] d\,W_D \qquad (6)$$

For a multi-stage vehicle the above equation can be used by adding the velocity increments from each stage burned during the mission. Care must be taken to include all weight including propellants, above the stage being burned, as dry weight in evaluating the derivatives for the stage being burned. For example, for a two-stage vehicle, the second stage dry weight plus propellants plus the first stage dry weight must be included as dry weight in evaluating first stage derivatives.

Now consider the "K^{th} stage" of an "N-stage" vehicle. Using Equation 4, an incremental impulsive velocity change occuring during this stage is given by

$$d(\Delta V)_K = \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_K d\,I_{sp_K} + \left(\frac{\partial \Delta V}{\partial W_D}\right)_K d\,W_{D_K} + \left(\frac{\partial \Delta V}{\partial W_P}\right)_K d\,W_{P_K} \qquad (7)$$

where W_{D_K} includes all the vehicle weight at burnout of the K^{th} stage and is therefore given by

$$W_{D_K} = \sum_{i=K}^{N} \left(W_{D_i} + W_{P_{i+1}}\right) \qquad (8)$$

The convention used herein is that when the subscript is greater than the upper limit (N in the above equation) the value of the parameter is zero. From Equation (8),

$$d\,W_{D_K} = \sum_{i=K}^{N} \left(d\,W_{D_i} + d\,W_{P_{i+1}}\right) \qquad (9)$$

Equation (7) can then be written

$$d(\Delta V)_K = \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_K d\,I_{sp_K} + \left(\frac{\partial \Delta V}{\partial W_D}\right)_K \sum_{i=K}^{N} \left(d\,W_{D_i} + d\,W_{P_{i+1}}\right)$$

$$+ \left(\frac{\partial \Delta V}{\partial W_P}\right)_K d\,W_{P_K} \qquad (10)$$

By summing Equation (10) from 1 to K the incremental impulsive velocity change for the first K stages is obtained.

$$\sum_{j=1}^{K} d\,\Delta V_j = \sum_{j=1}^{K} \left\{ \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_j d\,I_{sp_j} + \left(\frac{\partial \Delta V}{\partial W_D}\right)_j \sum_{i=j}^{N} \left(d\,W_{D_i} + d\,W_{P_{i+1}} \right) + \left(\frac{\partial \Delta V}{\partial W_P}\right)_j d\,W_{P_j} \right\}$$

or

$$\sum_{j=1}^{K} d\,\Delta V_j = \sum_{j=1}^{K} \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_j d\,I_{sp_j} + \sum_{j=1}^{K} \sum_{i=j}^{N} \left(\frac{\partial \Delta V}{\partial W_D}\right)_j d\,W_{D_i} + \sum_{j=1}^{K} \sum_{i=j}^{N} \left(\frac{\partial \Delta V}{\partial W_D}\right)_j d\,W_{P_{i+1}} \tag{11}$$

In double summations, as in the above equation, the inner summation is performed once for each value of the outer summation index. It can be shown that

$$\sum_{j=1}^{K} \sum_{i=j}^{N} \left(\frac{\partial \Delta V}{\partial W_D}\right)_j d\,W_{D_i} = \sum_{j=1}^{N} \sum_{i=1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i d\,W_{D_j} - \sum_{j=K+1}^{N} \sum_{i=K+1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i d\,W_{D_j} \tag{12a}$$

and

$$\sum_{j=1}^{K} \sum_{i=j}^{N} \left(\frac{\partial \Delta V}{\partial W_D}\right)_j d\,W_{P_{i+1}} = \sum_{j=2}^{N} \sum_{i=1}^{j-1} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i d\,W_{P_j} - \sum_{j=K+2}^{N} \sum_{i=K+1}^{j-1} \left(\frac{\partial \Delta V}{\partial W_P}\right)_i d\,W_{P_j} \tag{12b}$$

Substituting Equations (12a and 12b) into Equation (11)

$$\sum_{j=1}^{K} d(\Delta V)_j = \sum_{j=1}^{K} \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_j dI_{sp_j} + \sum_{j=1}^{N} \sum_{i=1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i dW_{D_j}$$

$$- \sum_{j=K+1}^{N} \sum_{i=K+1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i dW_{D_j} + \sum_{j=2}^{N} \sum_{i=1}^{j-1} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i dW_{P_j}$$

$$- \sum_{j=K+2}^{N} \sum_{i=K+1}^{j-1} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i dW_{P_j} + \sum_{j=1}^{K} \left(\frac{\partial \Delta V}{\partial W_P}\right)_j dW_{P_j}$$

or factoring,

$$\sum_{j=1}^{K} d(\Delta V)_j = \sum_{j=1}^{K} \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_j dI_{sp_j} + \sum_{j=1}^{N} \left\{\left[\sum_{i=1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i\right] - \left[\sum_{i=K+1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i\right]\right\} dW_{D_j}$$

$$+ \sum_{j=1}^{N} \left\{\left[\sum_{i=1}^{j-1} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i\right] - \left[\sum_{j=K+1}^{j-1} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i\right]\right\} dW_{P_j}$$

$$+ \sum_{j=1}^{K} \left(\frac{\partial \Delta V}{\partial W_P}\right)_j dW_{P_j} \qquad (13)$$

Now introducing a unit step function, δ_{jK}, defined as

$$\delta_{jK} = 1 \quad j \leq K, \qquad \delta_{jK} = 0 \quad j > K \qquad (14)$$

Equation (13) can be further factored to yield

$$\sum_{j=1}^{K} d(\Delta V)_j = \sum_{j=1}^{K} \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_j d I_{sp_j} + \sum_{j=1}^{N} \left\{ \left[\sum_{i=1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i \right] - \left[\sum_{i=K+1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i \right] \right\} d W_{D_j}$$

$$+ \sum_{j=1}^{N} \left\{ \left[\sum_{i=1}^{j-1} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i \right] - \left[\sum_{i=K+1}^{j-1} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i \right] + \delta_{jK} \left(\frac{\partial \Delta V}{\partial W_P}\right)_j \right\} d W_{P_j} \qquad (15)$$

Equation (15) gives the incremental impulsive velocity change through the first K stages of an N-stage vehicle. By letting K equal N the total incremental impulsive velocity change through the N stages can be found.

Letting K equal N,

$$\sum_{j=1}^{N} \left[\sum_{i=K+1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i \right] d WD_j = \sum_{j=1}^{N} \left[\sum_{i=N+1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i \right] d W_{D_j} = 0$$

and

$$\sum_{j=1}^{N} \left[\sum_{i=K+1}^{j-1} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i \right] d W_{P_j} = \sum_{j=1}^{N} \left[\sum_{i=N+1}^{j-1} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i \right] d W_{P_j} = 0$$

since the lower limit of the summation index is greater than the upper limit. Equation (15) then becomes, for $K = N$,

$$\sum_{j=1}^{N} d(\Delta V)_j = \sum_{j=1}^{N} \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_j dI_{sp_j} + \sum_{j=1}^{N} \left[\sum_{i=1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i\right] dW_{D_j}$$

$$+ \sum_{j=1}^{N} \left\{\left[\sum_{i=1}^{j-1} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i\right] + \left(\frac{\partial \Delta V}{\partial W_P}\right)_j\right\} dW_{P_j} \qquad (16)$$

Equation (16) gives the total incremental impulsive velocity change for an N-stage vehicle. All partial derivatives are calculated from a nominal or reference mission using Equations (5a, 5b, and 5c).

D.3 EXAMPLES OF GENERALIZED EQUATIONS TO DETERMINE TRADE-OFF FACTORS

Equations (10), (15), or (16) can now be used to obtain trade-off factors (sensitivity factors). This is done by imposing constraints dictated by the vehicle configuration and mission and by setting to zero all differentials not of interest. The trade-off factors are then the ratio of the coefficients of the differentials being considered.

Possible constraints which might be imposed include:

1. No change in total ideal velocity. This assumes no change in velocity losses and that the vehicle can still perform the nominal mission. The nominal mission is that at which the partial derivatives are evaluated.
2. No ideal velocity change in any given stage (as opposed to constraint 1 in which the sum of the ideal velocity for all stages is constant).
3. Constant consumed propellant per stage.
4. Constant total consumed propellant.
5. Constant consumed propellant ratio among stages.
6. Constant liftoff weight.
7. Constraints imposed by stage definition (when one physical vehicle stage is broken into two or more "stages" for analysis purposes because of engine restarts or discrete weight jettisons during any phase).
8. Combinations of the above constraints.

D.4 EXAMPLES OF USE OF EQUATIONS TO OBTAIN TRADE-OFF FACTORS

Consider a three-stage launch vehicle to inject a payload into translunar or interplanetary flight. The launch mission profile is the same as the Saturn V/Apollo LOR vehicle, i.e.,

1. Liftoff to first stage jettison.
2. Second stage ignition to launch escape system jettison.
3. LES jettison to second stage jettison.
4. Third stage ignition to earth orbit.
5. Translunar or interplanetary injection (propellant boil-off occurs during earth orbit).

For analysis purposes the above five phases will be considered as "stages" (i.e., N=5).

D.4.1 EXAMPLE I

Constraints:

1. Total mission ideal velocity constant $\left[\sum_{j=1}^{5} d(\Delta V)_j = 0\right]$
2. Injection ideal velocity constant $\left[d(\Delta V)_5 = 0\right]$
3. Constant consumed propellant loading per physical stage.

Constraint 1 above implies that Equation (16) must equal zero, while constraint 2 implies that Equation (10) equals zero for K = 5. Constraint 3 implies

$$dW_{P_1} = dW_{P_2} = dW_{P_3} = 0 \tag{17a}$$

and

$$dW_{P_4} + dW_{P_5} = 0 \tag{17b}$$

From Equation (16),

$$\sum_{j=1}^{5} d(\Delta V)_j = \sum_{j=1}^{5} \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_j d I_{sp_j} + \sum_{j=1}^{5} \left[\sum_{i=1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i \right] d W_{D_j}$$

$$+ \sum_{j=1}^{5} \left\{ \left[\sum_{i=1}^{j-1} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i \right] + \left(\frac{\partial \Delta V}{\partial W_P}\right)_j \right\} d W_{P_j} \qquad (18)$$

Factoring in constraint 1 and Equations 17a and 17b

$$0 = \sum_{j=1}^{5} \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_j d I_{sp_j} + \sum_{j=1}^{5} \left[\sum_{i=1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i \right] d W_{D_j}$$

$$+ \left[\left(\frac{\partial \Delta V}{\partial W_D}\right)_1 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_2 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_3 + \left(\frac{\partial \Delta V}{\partial W_P}\right)_4 \right] d W_{D_4}$$

$$+ \left[\left(\frac{\partial \Delta V}{\partial W_D}\right)_1 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_2 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_3 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_4 + \left(\frac{\partial \Delta V}{\partial W_P}\right)_5 \right] d W_{D_5} \qquad (19)$$

Making use of Equation (17b) the above equation reduces to

$$0 = \sum_{j=1}^{5} \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_j d I_{sp_j} + \sum_{j=1}^{5} \left[\sum_{i=1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i \right] d W_{D_j}$$

$$+ \left[\left(\frac{\partial \Delta V}{\partial W_D}\right)_4 + \left(\frac{\partial \Delta V}{\partial W_P}\right)_5 - \left(\frac{\partial \Delta V}{\partial W_P}\right)_4 \right] d W_{D_5} \qquad (20)$$

From Equation (10) (constraint 2)

$$0 = \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_5 d I_{sp_5} + \left(\frac{\partial \Delta V}{\partial W_D}\right)_5 d W_{D_5} + \left(\frac{\partial \Delta V}{\partial W_P}\right)_5 d W_{P_5} \qquad (21)$$

Solving for dW_{P_5},

$$dW_{P_5} = - \frac{\left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_5}{\left(\frac{\partial \Delta V}{\partial W_P}\right)_5} dI_{sp_5} - \frac{\left(\frac{\partial \Delta V}{\partial W_D}\right)_5}{\left(\frac{\partial \Delta V}{\partial W_P}\right)_5} dW_{D_5}$$

Substituting for dW_{P_5} in Equation (20),

$$0 = \sum_{j=1}^{5} \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_j dI_{sp_j} + \sum_{j=1}^{5} \left[\sum_{i=1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i \right] dW_{D_j}$$

$$- \left[\left(\frac{\partial \Delta V}{\partial W_D}\right)_4 + \left(\frac{\partial \Delta V}{\partial W_P}\right)_5 - \left(\frac{\partial \Delta V}{\partial W_P}\right)_4 \right]$$

$$\cdot \left[\frac{\left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_5}{\left(\frac{\partial \Delta V}{\partial W_P}\right)_5} dI_{sp_5} + \frac{\left(\frac{\partial \Delta V}{\partial W_D}\right)_5}{\left(\frac{\partial \Delta V}{\partial W_P}\right)_5} dW_{D_5} \right] \tag{22}$$

All trade-off factors involving the weight or specific impulse of any stage with respect to the weight or specific impulse of the same or any other stage can now be determined from Equation (22). This is done by solving Equation (22) for the ratio of the two differentials making up the trade-off factor, assuming all other differentials to be zero. Payload is included in last stage dry weight.

Some specific examples of the calculation of trade-off factors are:

a. The ratio of first stage dry weight to last stage dry weight (that is, payload).

From Equation (22) retaining only dW_D, and dW_{D_5} among the differentials,

$$\left(\frac{\partial \Delta V}{\partial W_D}\right)_1 dW_{D_1} + \left[\left(\frac{\partial \Delta V}{\partial W_D}\right)_1 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_2 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_3 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_4 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_5\right] dW_{D_5}$$

$$+ \left[\left(\frac{\partial \Delta V}{\partial W_D}\right)_4 + \left(\frac{\partial \Delta V}{\partial W_P}\right)_5 - \left(\frac{\partial \Delta V}{\partial W_P}\right)_4\right]\left[\frac{-\left(\frac{\partial \Delta V}{\partial W_D}\right)_5}{\left(\frac{\partial \Delta V}{\partial W_P}\right)_5} dW_{D_5}\right] = 0$$

or, solving for the desired ratio,

$$\frac{dW_{D_1}}{dW_{D_5}} = \frac{\left(\frac{\partial \Delta V}{\partial W_D}\right)_1 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_2 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_3 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_4 + \frac{\left(\frac{\partial \Delta V}{\partial W_D}\right)_5}{\left(\frac{\partial \Delta V}{\partial W_P}\right)_5}\left[\left(\frac{\partial \Delta V}{\partial W_P}\right)_4 - \left(\frac{\partial \Delta V}{\partial W_D}\right)_4\right]}{-\left(\frac{\partial \Delta V}{\partial W_D}\right)_1} \qquad (23)$$

b. The ratio of second stage specific impulse to weight of propellant boiloff in earth orbit (that is stage four "dry weight"). Note that the boiloff propellant must be considered as dry weight in this analysis due to constraint 3 and the fact that all dW_P's have been assumed zero in deriving Equation 22.

Also note that calculation stages two and three are both physical vehicle stage two. Therefore when considering changes in physical stage specific impulse, the change of specific impulse in all the calculation stages making up that physical stage must be considered (that is, $dI_{sp_2} = dI_{sp_3}$). This, of course, is not true for dry weight and propellant changes since these changes have been allowed for in the derivation of the basic equations by the definition of the calculation stages.

From Equation 22, retaining only dI_{sp_2}, DI_{sp_3}, and dW_{D_4} among the differentials

$$\left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_2 dI_{sp_2} + \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_3 dI_{sp_3} + \left[\left(\frac{\partial \Delta V}{\partial W_D}\right)_1 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_2 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_3 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_4\right] dW_{D_4} = 0$$

while from physical considerations

$$dI_{sp_2} = dI_{sp_3}$$

Again solving for the desired ratio,

$$\frac{dI_{sp_2}}{dW_{D_4}} = \frac{\left[\left(\frac{\partial \Delta V}{\partial W_D}\right)_1 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_2 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_3 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_4\right]}{\left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_2 + \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_3} \tag{24}$$

c. The ratio of second stage specific impulse to fourth stage specific impulse.

From Equation 22 retaining only dI_{sp_2}, dI_{sp_3}, dI_{sp_4}, and dI_{sp_5} among the differentials,

$$\left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_2 dI_{sp_2} + \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_3 dI_{sp_3} + \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_4 dI_{sp_4} + \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_5 dI_{sp_5}$$

$$+ \left[\left(\frac{\partial \Delta V}{\partial W_D}\right)_4 + \left(\frac{\partial \Delta V}{\partial W_P}\right)_5 - \left(\frac{\partial \Delta V}{\partial W_P}\right)_4\right] \left[\frac{-\left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_5}{\left(\frac{\partial \Delta V}{\partial W_P}\right)_5} dI_{sp_5}\right] = 0$$

while from physical considerations

$$dI_{sp_2} = dI_{sp_3}$$

and

$$dI_{sp_4} = dI_{sp_5}$$

Solving for the desired ratios,

$$\frac{dI_{sp_2}}{dI_{sp_4}} = \frac{\left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_4 - \left[\left(\frac{\partial \Delta V}{\partial W_D}\right)_4 - \left(\frac{\partial \Delta V}{\partial W_P}\right)_4\right]\left[\frac{\left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_5}{\left(\frac{\partial \Delta V}{\partial W_P}\right)_5}\right]}{\left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_2 + \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_3} \tag{25}$$

Other trade-off factors can be calculated in like manner by considering other differentials. This is not as laborious as might be suspected at first glance since the individual partial derivatives need only be calculated once, and their groupings, the coefficients of the differentials, also need only be calculated once.

D.4.2 EXAMPLE II

Constraints:

1. Total mission ideal velocity constant $\left[\sum_{j=1}^{5} d(\Delta V)_j = 0\right]$
2. Injection ideal velocity constant $\left[d(\Delta V)_5 = 0\right]$
3. Constant consumed propellant ratio among stages.
4. Constant second physical stage propellant consumed prior to LES jettison (i.e., constant second calculation stage propellants, $dW_{P_2} = 0$).
5. Constant liftoff weight.

From constraints 1 and 2, as in Example I,

$$\sum_{j=1}^{5} d(\Delta V)_j = \sum_{j=1}^{5} \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_j dI_{sp_j} + \sum_{j=1}^{5} \left[\sum_{i=1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i\right] dW_{D_j}$$

$$+ \sum_{j=1}^{5} \left\{\left[\sum_{i=1}^{j-1} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i\right] + \left(\frac{\partial \Delta V}{\partial W_P}\right)_j\right\} dW_{P_j} \tag{26}$$

and

$$0 = \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_5 dI_{sp_5} + \left(\frac{\partial \Delta V}{\partial W_D}\right)_5 dW_{D_5} + \left(\frac{\partial \Delta V}{\partial W_P}\right)_5 dW_{P_5} \tag{27}$$

From constraint 4,

$$dW_{P_2} = 0 \tag{28}$$

While from constraint 3

$$R_1 = \frac{W_{P_2} + W_{P_3}}{W_{P_1}} \tag{29a}$$

and

$$R_2 = \frac{W_{P_4} + W_{P_5}}{W_{P_1}} \tag{29b}$$

so that

$$R_1 dW_{P_1} = dW_{P_2} + dW_{P_3} = dW_{P_3} \tag{30a}$$

and

$$R_2 dW_{P_1} = dW_{P_4} + dW_{P_5} \tag{30b}$$

From constraint 5,

$$dW_{D_1} + dW_{D_2} + dW_{D_3} + dW_{D_4} + dW_{D_5} + dW_{P_1} + dW_{P_2} + dW_{P_3} + dW_{P_4} + dW_{P_5} = 0 \qquad (31)$$

Substituting Equations (28, 30a, and 30b) into the above equation,

$$dW_{D_1} + dW_{D_2} + dW_{D_3} + dW_{D_4} + dW_{D_5} + dW_{P_1} + 0 + R_1 dW_{P_1} + R_2 dW_{P_1} - dW_{P_5} = 0$$

Solving for dW_{P_1},

$$dW_{P_1} = \frac{-(dW_{D_1} + dW_{D_2} + dW_{D_3} + dW_{D_4} + dW_{D_5})}{1 + R_1 + R_2} \qquad (32)$$

From Equation (27)

$$dW_{P_5} = -\left[\frac{\left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_5 dI_{sp_5} + \left(\frac{\partial \Delta V}{\partial W_D}\right)_5 dW_{D_5}}{\left(\frac{\partial \Delta V}{\partial W_P}\right)_5}\right] \qquad (33)$$

Substituting the above into Equation (30b),

$$dW_{P_4} = R_2 dW_{P_1} + \left[\frac{\left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_5 dI_{sp_5} + \left(\frac{\partial \Delta V}{\partial W_D}\right)_5 dW_{D_5}}{\left(\frac{\partial \Delta V}{\partial W_P}\right)_5}\right] \qquad (34)$$

Expanding the last summation of Equation (26), with constraint 1, let

$$\text{Sum} = \sum_{j=1}^{5} \left\{ \left[\sum_{i=1}^{j-1} \left(\frac{\partial \Delta V}{\partial W_D} \right)_i \right] + \left(\frac{\partial \Delta V}{\partial W_P} \right)_j \right\} d\,W_{P_j}$$

then

$$\text{Sum} = \left[\left(\frac{\partial \Delta V}{\partial W_P} \right)_1 \right] d\,W_{P_1} + \left[\left(\frac{\partial \Delta V}{\partial W_D} \right)_1 + \left(\frac{\partial \Delta V}{\partial W_P} \right)_2 \right] d\,W_{P_2}$$

$$+ \left[\left(\frac{\partial \Delta V}{\partial W_D} \right)_1 + \left(\frac{\partial \Delta V}{\partial W_D} \right)_2 + \left(\frac{\partial \Delta V}{\partial W_P} \right)_3 \right] d\,W_{P_3}$$

$$+ \left[\left(\frac{\partial \Delta V}{\partial W_D} \right)_1 + \left(\frac{\partial \Delta V}{\partial W_D} \right)_2 + \left(\frac{\partial \Delta V}{\partial W_D} \right)_3 + \left(\frac{\partial \Delta V}{\partial W_P} \right)_4 \right] d\,W_{P_4}$$

$$+ \left[\left(\frac{\partial \Delta V}{\partial W_D} \right)_1 + \left(\frac{\partial \Delta V}{\partial W_D} \right)_2 + \left(\frac{\partial \Delta V}{\partial W_D} \right)_3 + \left(\frac{\partial \Delta V}{\partial W_D} \right)_4 + \left(\frac{\partial \Delta V}{\partial W_P} \right)_5 \right] d\,W_{P_5}$$

Substituting Equations (28, 30a, 33, and 34) into the above equation,

$$\text{Sum} = \left[\left(\frac{\partial \Delta V}{\partial W_P} \right)_1 \right] d\,W_{P_1} + \left[\left(\frac{\partial \Delta V}{\partial W_D} \right)_1 + \left(\frac{\partial \Delta V}{\partial W_D} \right)_2 + \left(\frac{\partial \Delta V}{\partial W_P} \right)_3 \right] R_1 d\,W_{P_1}$$

$$+ \left[\left(\frac{\partial \Delta V}{\partial W_D} \right)_1 + \left(\frac{\partial \Delta V}{\partial W_D} \right)_2 + \left(\frac{\partial \Delta V}{\partial W_D} \right)_3 + \left(\frac{\partial \Delta V}{\partial W_P} \right)_4 \right]$$

$$\cdot \left\{ R_2 d\,W_{P_1} + \left[\frac{\left(\frac{\partial \Delta V}{\partial I_{sp}} \right)_5 d\,I_{sp_5} + \left(\frac{\partial \Delta V}{\partial W_D} \right)_5 d\,W_{D_5}}{\left(\frac{\partial \Delta V}{\partial W_P} \right)_5} \right] \right\}$$

$$- \left[\left(\frac{\partial \Delta V}{\partial W_D} \right)_1 + \left(\frac{\partial \Delta V}{\partial W_D} \right)_2 + \left(\frac{\partial \Delta V}{\partial W_D} \right)_3 + \left(\frac{\partial \Delta V}{\partial W_D} \right)_4 + \left(\frac{\partial \Delta V}{\partial W_P} \right)_5 \right]$$

$$\cdot \left[\frac{\left(\frac{\partial \Delta V}{\partial I_{sp}} \right)_5 d\,I_{sp_5} + \left(\frac{\partial \Delta V}{\partial W_D} \right)_5 d\,W_{D_5}}{\left(\frac{\partial \Delta V}{\partial W_P} \right)_5} \right]$$

Factoring,

$$\text{Sum} = \left\{ \left(\frac{\partial \Delta V}{\partial W_P}\right)_1 + \left[\left(\frac{\partial \Delta V}{\partial W_D}\right)_1 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_2 + \left(\frac{\partial \Delta V}{\partial W_P}\right)_3\right] R_1 \right.$$

$$\left. + \left[\left(\frac{\partial \Delta V}{\partial W_D}\right)_1 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_2 + \left(\frac{\partial \Delta V}{\partial W_D}\right)_3 + \left(\frac{\partial \Delta V}{\partial W_P}\right)_4\right] R_2 \right\} d\, W_{P_1}$$

$$- \left[\left(\frac{\partial \Delta V}{\partial W_D}\right)_4 - \left(\frac{\partial \Delta V}{\partial W_P}\right)_4 + \left(\frac{\partial \Delta V}{\partial W_P}\right)_5\right]$$

$$\cdot \left[\frac{\left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_5 d\, I_{sp_5} + \left(\frac{\partial \Delta V}{\partial W_D}\right)_5 d\, W_{D_5}}{\left(\frac{\partial \Delta V}{\partial W_P}\right)_5}\right] \tag{35}$$

Substituting Equations (32, 35) into Equation (26), using constraint 1,

$$0 = \sum_{j=1}^{5} \left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_j d\, I_{sp_j} + \sum_{j=1}^{5} \left[\sum_{i=1}^{j} \left(\frac{\partial \Delta V}{\partial W_D}\right)_i - \frac{Q}{1 + R_1 + R_2}\right] d\, W_{D_j}$$

$$- \left[\left(\frac{\partial \Delta V}{\partial W_D}\right)_4 - \left(\frac{\partial \Delta V}{\partial W_P}\right)_4 + \left(\frac{\partial \Delta V}{\partial W_P}\right)_5\right] \left[\frac{\left(\frac{\partial \Delta V}{\partial I_{sp}}\right)_5 d\, I_{sp_5} + \left(\frac{\partial \Delta V}{\partial W_D}\right)_5 d\, W_{D_5}}{\left(\frac{\partial \Delta V}{\partial W_P}\right)_5}\right] \tag{36}$$

Where Q is the coefficient of $d\, W_{P_1}$ in Equation (35). Equation (36) is comparable to Equation (22) of Example I, and can be used to compute the desired trade-off factors in the same manner as was done in Example I. R_1 and R_2 are given by Equations (29a, 29b) or can be assigned arbitrarily. By letting R_2 go to zero and retaining R_1, the case of constant propellant loading in the third physical stage and a constant propellant loading ratio between the first two physical stages can be investigated using Equation (36).

MARK II

COMPUTER PROGRAM DESCRIPTIONS

(USER'S GUIDE)

23 December 1965

Apollo Support Department
General Electric Company
Daytona Beach, Florida

ABSTRACT

The Mark II Computational System is being designed and developed for the purpose of prediction analysis. This User's Guide describes and explains how to use the computer programs comprising Mark II.

TABLE OF CONTENTS

Title	Page
INTRODUCTION	1
HISTORICAL BACKGROUND	1
MARK II SYSTEM PHILOSOPHY	2
MARK II COMPUTER SYSTEM	4
Program Control	4
Input Data Processing	4
Trend Prediction	6
Output Data Processing	6
PURPOSE OF THE CURRENT PUBLICATION	7
REVISIONS	7

	Reference No.
EXECUTIVE CONTROL (SPACE)	54S
CONTROL PROGRAM (GOODE)	55S
MAXIMUM LIKELIHOOD LINEAR	56S
MAXIMUM LIKELIHOOD NONLINEAR	57S
ADAPTIVE (FOURIER) EXPONENTIAL	58S
ASYMPTOTIC (LOGISTIC) EXPONENTIAL	59S
OUTPUT PROGRAM (OUTPT)	60S
HISTORY PLOT PROGRAM (HIS)	61S
TITLING SUBROUTINE (TITEL)	62S
SUMMING PROGRAM (SUM)	63S
DECISION RELEVANCY (Criticality)	64S
OVERALL COST ESTIMATION	65S
UPDATE PROGRAM	66S
SUBROUTINE GETDAT	67S
NORMALIZATION SUBROUTINE (NORM12)	68S
DOUBLE PRECISION MATRIX INVERTER (DPM1)	69S
PROBABLE ERROR PROGRAM (RSS)	70S
PLOTTING SUBROUTINE (UMPLOT)	71S

APPENDIX

INTRODUCTION

This section describes:

a. The historical development of the Mark II System,

b. The current, general philosophy of the Mark II System,

c. The Mark II Computer System,

d. A brief description of the current document.

HISTORICAL BACKGROUND

In August of 1964, work was started on a computer program which could predict the weight of functional systems at some future date. The first model to be programmed was a linear model whose parameters were estimated by maximum-likelihood estimation. At that time, there was a small number of functional systems the weights of which had to be predicted each month. Input data for each functional system was kept on punched cards. As more data was received, the deck was simply enlarged to include the new information. As long as there were only a few functional systems and a limited amount of data for each, this mode of operation was quite satisfactory.

In December of 1964, the exponential model was introduced. About this same time the computerized removal of nonrandom changes was initiated. With the increasing number of functional systems being processed, the card input of data was becoming cumbersome. To alleviate this problem, the Weight Data File (WDF) was created in March 1965. The Weight Data File is a magnetic tape record of all the reported observed weights, nonrandom changes, and other weight data. A program was written to update the Weight Data File as additional information became available.

In order that running of the linear and/or exponential-model computer programs be as easy as possible, a combination program was written in April 1965. This program allowed the running of either or both of the existing programs, permitting a significant reduction in the necessary control cards. The combination program, along with the Update program and the Weight Data File, served as the core of the first system assembled for the purpose of trend prediction. This first system, which was no more than a loosely tied-together group of programs, was called Mark I and was completed during April 1965.

Since Mark I was first introduced, it has undergone many revisions. New auxiliary programs have been written and tied into this collection of programs. Two new trend models were developed and programmed.

The Fourier model was developed in May 1965, and the logistic model in June 1965. It soon became apparent that Mark I was becoming obsolete. Another revision was in order and an entirely new system design was initiated in June 1965. The new system was appropriately named Mark II.

The Mark II System is still in an evolutionary state, and will continue to be for quite some time to come; however, enough of the initial concept formulation and sufficient programs are completed to warrant the publishing of the first User's Guide.

MARK II SYSTEM PHILOSOPHY

Mark II is a system of computer programs written for the IBM 7044 computer. These programs operate as physically independent but functionally consistent units. Each program in the system is designed for a specific task. Each may utilize output from other programs in the system and from information stored in the Weight Data File. The system operates under a monitor system, SPACE (an acronym for Subsystem Processor for the Apollo Computing Effort), which handles communication between programs. The system also provides a tape library of Mark II programs and has binary input/output (I/O) capabilities.

Most computer programs are designed to be run under control of a supervisory program or "system. The advantages of these system attributes accrue from their ability to provide standardized I/O subroutines, as well as an automated means of executing the desired program or sequence of programs.

The particular system attributes that were sought for Mark II were:

a. The I/O subroutines must be as efficient as possible, for large quantities of binary data.

b. The I/O routines must provide a very convenient and flexible means of program intercommunication.

c. The system itself must provide the capability of a program library such that the desired programs for a production run can rapidly be called from magnetic tape or disk file in any order.

SPACE, which operates as a subset of the 7040/7044 IBJOB Processor Monitor, was chosen as the executive or administrative program for Mark II. The framework of SPACE is centered around a collection of seven versatile I/O subroutines. By using these subroutines the programmer can disassociate himself completely from such problems as the physical aspects of retrieving or creating externally stored data, blocking/unblocking logicial records, file positioning, synchronized CPU/channel overlap, and the differences in the characteristics of recording devices. Thus he is permitted to concentrate on his primary task - the internal processing of data.

A basic requirement of any executive monitor is to automate the running of a series of data processing programs by calling these programs from a library tape as they are needed. This requirement necessitates the SPACE user to create his own library by employing the chain feature of IBLDR. Enhanced by these facilities, the objectives of the monitor may be outlined as follows:

a. To create data files on a given I/O device with the ability to randomly access any of these files.
b. To enable "data-sharing" capabilities whereby files of output data from any program(s) can serve as input data to any later program(s), either within the same job or not.
c. To provide a framework around which systems of data processing programs can be developed. Once data is available in the standard SPACE file format, the whole range of previously written programs is available to process it.
d. To provide a flexible means of program intercommunication.

Since we have a monitor system specializing in a library of intercommunicating programs, we can build Mark II as a series of programs rather than just one program. This modularized concept is very desirable for many reasons three of which are:

a. Each program can be written as efficiently as possible without worrying about interfering with other program areas.
b. Many programmers can work on Mark II permitting many programs to be written in a short span of time.
c. New capabilities can be easily added to Mark II by replacing the affected programs by new ones or by adding new programs to the series.

MARK II COMPUTER SYSTEM

An examination of the weight trending problem indicates that the operations fall into one of four groups:

a. Program Control.
b. Input Data Processing.
c. Trend Prediction.
d. Output Data Processing.

These groups are functionally dependent in that any one group depends on another for instructions, input data, output data, etc. The basic logical structure of the Mark II system is illustrated in Figure 1. The four functional components are indicated together with the important data links.

Program Control

This group is the most sensitive of the four. Often referred to as the monitor or executive routine, it has the responsibility of accurately interpreting the user's instructions and of providing the required output. It establishes the proper sequence of operations for each job to be run and then monitors the resulting execution. It efficiently handles the flow of large blocks of data into and through the computer and communicates with all programs in the system.

The user, having decided on a sequence of operations, turns control over to the monitor. Thereafter, the monitor functions much as a bookkeeper. It executes programs in a specified order and keeps track of all data generated by individual programs.

When the sequence of operations has been completed, the job is terminated. Notice that the user has the flexibility of performing a long and involved analysis study or making a series of several short computer runs each with a specific goal in mind.

Input Data Processing

Input data may be classified as:

a. Functional System Weight Data.
b. Instructions.

Functional System Weight Data is a complete history of the weight data for a given functional system. This data includes time points, Estimated, Calculated, and Actual (E/C/A) weight percentages, nonrandom weight changes, shipping date, etc. The

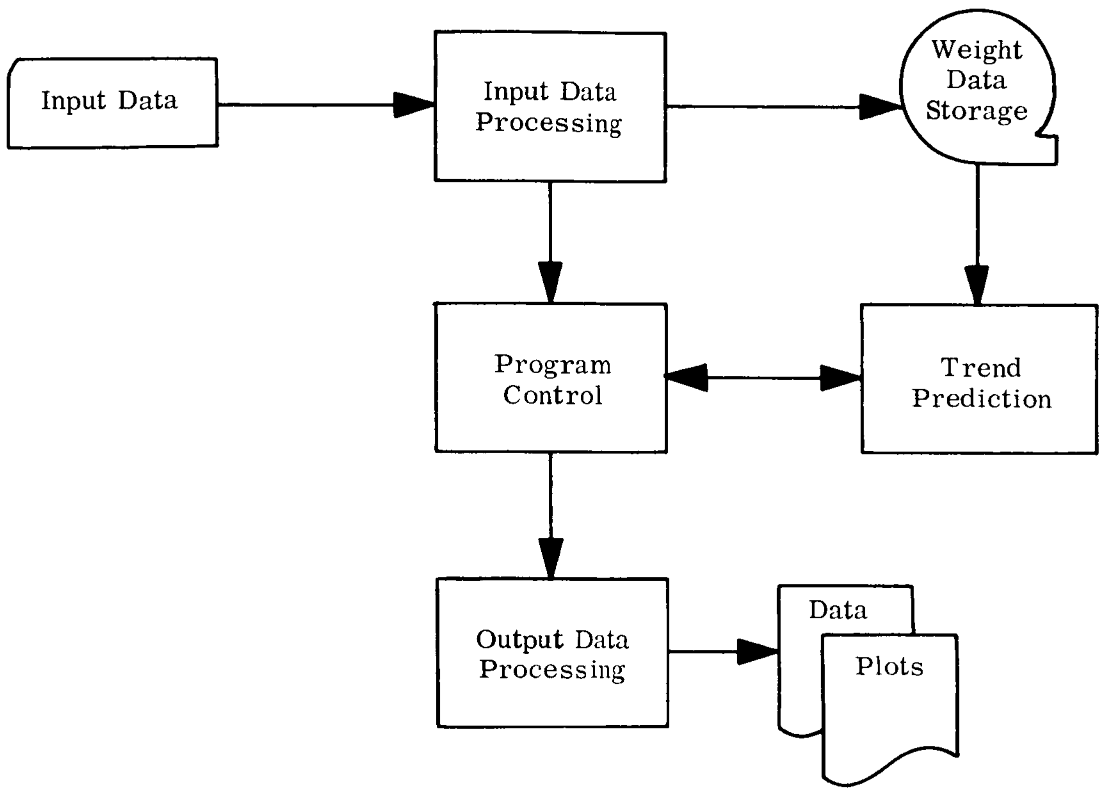


Figure 1. Mark II Basic Flow

historical weight data is stored on magnetic tape and is updated monthly or whenever a weight change is received. This involves the extraction of raw data from reports, graphs, charts, etc. and the preparation of such data for the computer. Input data cards must be punched, and when all the raw data has been reduced to data cards, the cards are processed by a computer program which performs data consistency checks and then updates the weight data file. Thus, the entire history of weight data for each functional system is stored on magnetic tape. This history data file will be called the Weight Data File (WDF).

A second type of input data is classified as input instructions. It includes operating instructions and sequencing instructions. For example, the user must specify what particular trend prediction programs are to be executed for each functional system. He must also specify which auxiliary programs, if any, are to be executed, and the order in which these auxiliary programs are to be executed.

Trend Prediction

Four prediction models are currently available in the Mark II Computer System:

a. Maximum Likelihood Linear.
b. Maximum Likelihood Nonlinear.
c. Adaptive (Fourier) Exponential.
d. Asymptotic (Logistic) Exponential.

Each of the models operates in the same general manner. The weight data for the given functional system is moved from a temporary storage location to a working storage area. Here it is prepared for the actual trending operation by removal of non-random changes and normalization of any outliers. This pretrend data preparation is dictated by the user, and the appropriate instructions are entered into the computer as input instructions. Once the data has been prepared for prediction analysis, program parameters are computed and the mean trend line is determined. Posttrend operations are now executed, and include computation of the prediction lines, calculation of confidence limits, and the introduction of E/C/A effects in the prediction range. The entire set of weight data, to include history data and predicted data, is stored on magnetic tape for future reference.

Output Data Processing

The output data processing programs perform various computations on the trended data in order to provide the most timely output for the user. Tabular listings and plots of each functional system that was trended during the computer run are presented. Another program computes the total weight of a spacecraft or launch vehicle from the individual functional systems which comprise the spacecraft or launch vehicle. Other operations in this area include:

a. Computation of probable errors.
b. Analysis of trade-offs.
c. Scheduling computations.
d. Cost data processing.

The result of this group of programs is a set of tabular data, graphs, and charts which present a complete picture of the current weight status of a launch vehicle or spacecraft.

Figure 2 presents, in elemental form, the concepts discussed above. Here, the elements for Prediction Analysis have been arranged in five groups:

a. Monitor and Executive Routine.
b. Data File Library - WDF.
c. Pretrend Data Processing.
d. Trend Prediction.
e. Prediction Output Processing.

PURPOSE OF THE CURRENT PUBLICATION

This document, as published today and as will be subsequently updated, will serve as an up-to-date description of the programs and capabilities available with the Mark II System. Each program currently incorporated in the system will be represented by a section containing the following information:

a. A brief word description of each program.
b. A pictorial representation of the output tape written by the program or typical printed output from the program.
c. A flow chart.
d. A computer program listing.

REVISIONS

As new programs are added or as existing programs are modified, revisions will be published documenting these additions and revisions.

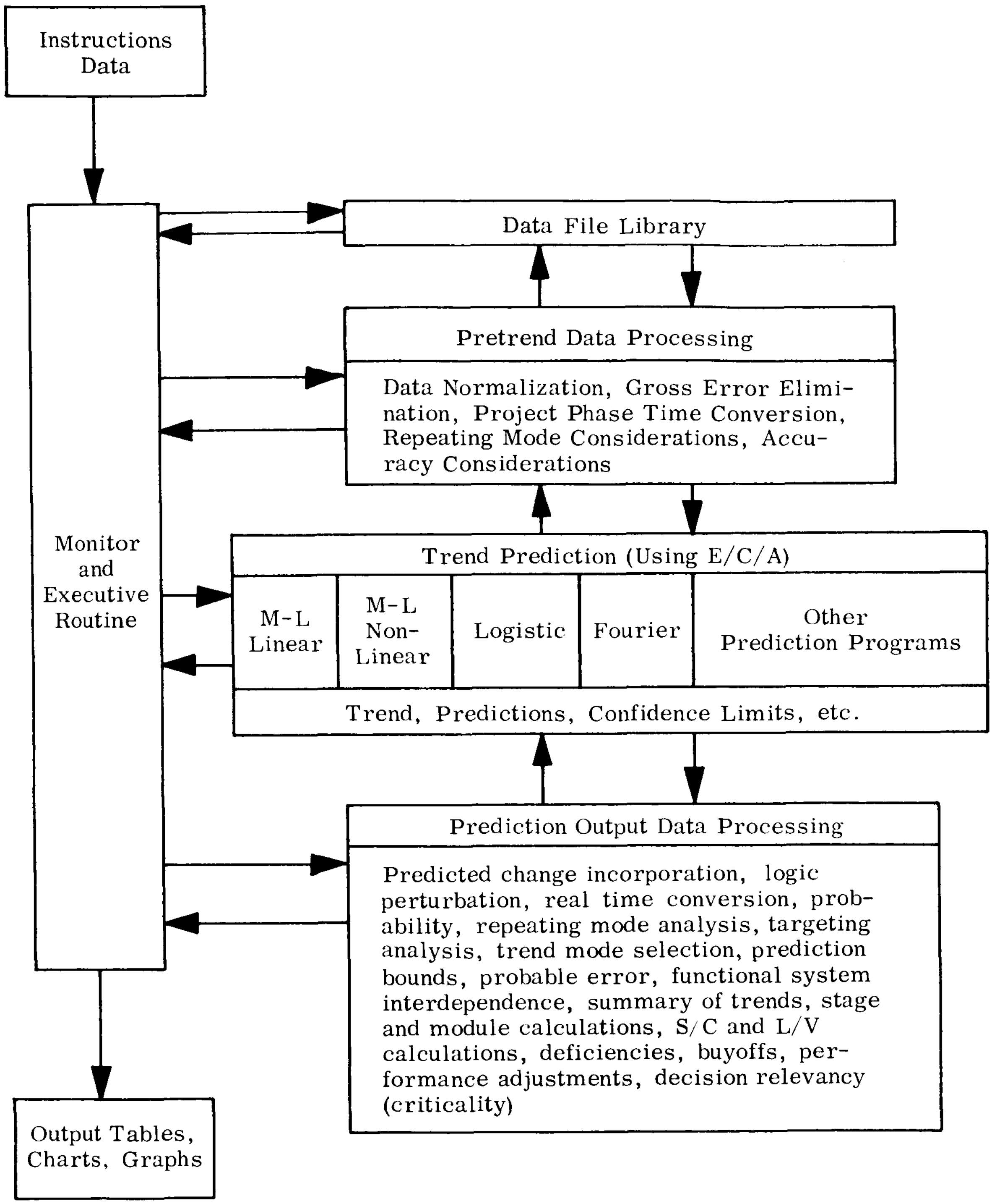


Figure 2. Mark II Prediction Analysis Computation System Elements

The decimal numbering system has been chosen to allow some flexibility in paginating revisions and/or additions. In addition to a Reference (page) No., each page will show its Issue Date and the date of the Superseded page.

EXECUTIVE CONTROL - 54S (SPACE)

Overall control of the Mark II Computational System is maintained by the executive program, SPACE, which operates as a subset of the 7040/7044 IBJOB Processor Monitor. The user is referred to the Appendix for a complete description of SPACE.

The link structure of the complete Mark II System is presented on Reference No. 54.5 Notice that the first link is the SPACE link while the very last link is the Utility link. Since the Mark II System operates by loading the specified program link into core storage from the Mark II library and by executing the specified program, the input data deck consists of a link card, followed by the input data for that program. Program data decks are stacked one behind the other in the order in which the programs are to be executed. The remainder of this section will be devoted to presenting several examples of job deck "set-ups".

EXAMPLE 1 - It is desired to run the trend prediction programs on the specified functional systems, the History Plot Program, the Summing Program, and the Probable Error Program. The complete input deck would look as is shown below:

CARD COLUMNS

```
        1------8-----------------------------------------------------72
A       REELS  R 691  NOLABEL  R 692   NOLABEL  R 693 NOLABEL
               R 84   NOLABEL   NOTYPE   *

B       MAIN     *
          1122101           X X X     X  1.00               1
          1122103  1122106    X                             1
          1234113           X                               3
          1234116  1234119  X X               X     1.00    1
          1234162              X    X    1.00               2
        END

C       MAIN     *
          1122100
          1234100
        HIS

D       SUM      *
               1122100  2    12    1965        2    *
               1122105  1.   TREND
               END *
               1234100  1    12    1965        2    *
               1234116  846.3      7.8
               1234118  4.   TREND
               END *
               END OF SUMS      *

E       RSS      *
        MISION 105   12   65        *
        PARTS  1122100  L/V FACTOR    .185   *
        PARTS  1234100  L/V FACTOR    .810   *
        ENDCSE *
               END RSS *
```

The data deck listed above will perform the operations specified in the example heading. In particular, the following items should be noted:

- Packet A, which consists of the "Reels" card, must be the first card in the input deck. For this example, the results of the linear program would be written on reel 691; the results of the nonlinear program would be written on reel 692; the results of the Fourier model would be written on reel 693; and the results of the logistic model would be written on reel 84.
- Packet B consists of those data cards which are necessary to run the trend prediction programs. It should be noted that the link card is the first card of this package and is followed by the data cards required by the Main Program (GOODE).
- Packet C consists of those cards necessary to run the History Plot Program.
- Packet D consists of those cards necessary to run the Summing Program.
- Packet E consists of those cards necessary to run the Probable Error Program.

The following points should be noted concerning the job deck setup:

- The "Reels" card is the first card in the deck.
- Each packet consists of a link card (which acts as a trigger to SPACE to call the desired link into core storage from the library and to begin execution) and the program input data. For a complete description of the program input data the user is referred to the individual program descriptions.
- The individual programs may be run in any sequence desired. It should be remembered, however, that the Summing Program (SUM) and the Probable Error Program (RSS) require the output of the Trend Prediction Programs. Thus, any of the following sequences of packets would be acceptable:

  A, B, C, E, D
  A, B, D, E, C
  A, B, E, D, C
  A, B, D, C, E
  A, B, E, C, D
  A, C, B, D, E
  A, C, B, E, D

EXAMPLE 2 - Run the Trend Prediction Programs only. The sequence A, B will perform this operation.

EXAMPLE 3 - Run the History Plot Program and the Trend Prediction Programs. The sequence A, C, B will perform this operation.

EXAMPLE 4 - Run the History Plot Program only. The input deck appears as:

CARD COLUMNS

```
1-------8---------------------------------------------------------72
REELS  *
        1122100
        1234100
HIS
```

NOTE

The "Reels" card contains no reels information since it is not necessary to mount the trending tapes; however, this card is still necessary.

EXAMPLE 5 - Run the Summing Program only. The input deck will be as:

CARD COLUMNS

```
1-------8---------------------------------------------------------72
REELS   R 691    R 692    R 693    R 84  *
SUM       3   3    2   2      *
        1122100   2   12    1965    2    *
        1122105   1.   TREND
        END *
        1234100   1   12    1965    2    *
        1234116   846.3     7.8
        1234118   4.   TREND
        END *
        END OF SUMS      *
```

NOTE

The "Reels" card contains the reels numbers of the tapes upon which the linear, nonlinear, Fourier, and logistic results have already been written. The link card (card No. 2 above) contains the integers 3 3 2 2 which represent the numbers of data files written on each of the four trending tapes.

EXAMPLE 6 - Run the Probable Error Program only.

CARD COLUMNS

```
1------8----------------------------------------------------------72
REELS  R -1 R -2    R 693    *
SUM    22  *
MISION 105  12  65    *
PARTS  1122100  L/V  FACTOR  .185  *
PARTS  1234100  L/V  FACTOR  .810  *
ENDCSE *
       END RSS  *
```

NOTE

The "Reels" card contains two "scratch" reels (see SPACE Reference Manual) and reel 693 which contains the results of the Fourier model which has previously been run. The integer 22 on the SUM card (card No. 2) is the number of data files on the Fourier tape.

Note that the Trend Prediction Programs and the History Plot Program must have the Weight Data File (WDF) advance mounted. The symbolic address should be U07 with a file identifier FTC01.

The following programs are not run under the control of SPACE but are run as separate jobs independent of SPACE:

a. Update Program (66S)
b. Decision Relevancy Program (64S)
c. Cost Program (65S)

The job deck setup and necessary control cards to run these programs are described in the individual program references.

```
$IBLDR SPCBS    04/10/65                          SPCB0001
$IBLDR GOOCH    09/25/65                          GOOC0001
$IBLDR TITEL    10/20/65                          TITE0001
$IBLDR INDIFE   10/20/65                          INDI0001
$IBLDR FLIP     09/15/65                          FLIP0001
$ENTRY
$LINK  MAIN
$IBLDR GOODE    10/14/65
$IBLDR GETDAT   10/14/65                          GETD0001
$IBLDR NORM1<   10/14/65                          NORM0001
$IBLDR DPMI     03/25/65                          DPMI0001
$IBLDR DPMI1    03/25/65                          DPMI0001
$ENTRY
$LINK  *56S     DPMI1
$IBLDR LINEAR   10/26/65                          LINE0001
$ENTRY
$LINK  57S      LINEAR
$IBLDR NONLNR   10/26/65                          NONL0001
$ENTRY
$LINK  58S      NONLNR
$IBLDR FOURNR   10/26/65                          FOUR0001
$ENTRY
$LINK  59S      FOURNR
$IBLDR LOGMOD   10/26/65                          LOGM0001
$ENTRY
$LINK  OUTPT    LOGMOD
$IBLDR OUTPUT   10/25/65                          OUTP0001
$IBLDR UMPLOT   03/04/65                          UMPL0001
```

Reference No. 54.6
Issue Date 23 December 1965
Supersedes New

```
$ENTRY
$LINK  HIS     OUTPUT
$IBLDR HISTRY  10/25/65                                   HIST0001
$IBLDR UMPLOT  03/04/65                                   UMPL0001
$ENTRY
$LINK  SUM     HISTRY
$IBLDR SUMING  11/30/65                                   SUMI0001
$ENTRY
$LINK  RSS     SUMING
$IBLDR REPORT  10/18/65                                   REPO0001
$IBLDR PROCES  10/19/65                                   PROC0001
$IBLDR PRINT   10/18/65                                   PRIN0001
$IBLDR BDATA   05/26/65                                   BDAT0001
$ENTRY
$LINK  UTLITY
$IBLDR UTLBS   04/10/65                                   UTLB0001
$ENTRY
$ENDCH
```

CONTROL PROGRAM - 55S (GOODE)

This program is the Control Program for the Trending System. It calls any or all of the four trending methods, the Output Routine, and the History Plot Routine, including any associated subroutines. The main job of this program is to set up the necessary instructions to process the functional system assemblies through the various trend prediction models. GOODE performs all of the input functions for the dependent programs. The data cards are read in one by one, analyzed, and stored in core storage. They are then sorted into an array of ascending case numbers to avoid the expense of constant winding and rewinding of the data tapes to match the cases with the files. The cards are analyzed one by one, and the arguments are prepared for the desired trending method(s). GOODE also retrieves the necessary trending information from the Weight Data File and stores this information into a working location in core.

At this time the card image is scanned and the desired trend prediction programs are called and executed. At the completion of execution of each trend prediction program, control is returned to GOODE which determines if all the required prediction programs have been called and executed. After all prediction programs for that card have been executed, the next card image is processed in the same manner. When all the input data has been processed, GOODE calls the Output Program which reads the four binary files generated by the prediction programs. At the completion of the Output Program, control is returned to SPACE.

If the History Plot Routine is to be used, the cards are stored and sorted in the same manner as in trending. The trending methods, however, are bypassed and the History Plot Program is called. The History Plot Program prepares its own output. When GOODE is through processing all of the case numbers, it simply returns control to SPACE directly, bypassing the Output Program.

Reference No. 55.1
Issue Date 23 Dec 1965
Supersedes 13 August 1965

GOODE accepts two basic types of input which must be in the following forms:

INPUT FOR TREND PREDICTION ANALYSIS

	Columns	
Program Indicator - Used Only on the last card of the deck	1-3	Punch "END"
Starting Case Number	4-10	7 digits (integer) required.
Ending Case Number (Optional)	12-18	
Programs or Trending Methods on which data is to be run		
a. Linear (56S)	20	Indicated by an "X" punched in the column.
b. Nonlinear (57S)	22	
c. Logistic Model (59S)	24	
• Option 1	26	
• Option 2	27	
• Option 3	28	
• Option 4	29	
• Program Maturity Factor α	31-34	Number with decimal point punched in Column 32.
d. Fourier Model (58S)	36	Indicated by an "X" punched in the column.
• Weighting Factor α	41-44	Number with decimal point punched in Column 42.
e. Normalization Factor	46	Integer 0, 1, 2, or 3 to indicate option.
• See NORM 12 (67S)		

INPUT FOR HISTORY - PLOTS

	Columns	
Program Indicator - Used Only on the last card of the deck	1-3	Punch "HIS"
Starting Case Number	4-10	7 digits (Integer) required.
Ending Case Number (Optional)	12-18	

Example of Trend Data Deck

CARD COLUMNS

1	4	12	20	22	24	26	27	28	29	31	36	41	46
	0611903	0611909	X		X			X		0.85			3
	0651426			X									1
	0651427				X	X				1.00	X	1.00	1
	0651436		X	X	X		X			1.00			3
	0651437	0651449	X										2
END													

Card 1 tells GOODE to trend all of the cases on the WDF from 0611903 through 0611909 by the Linear Prediction Model and also by the Logistics Model with option 3, and using an alpha of 0.85 for E/C/A effects over the observed range (see Logistics, 59S). The final piece of information is a normalization option of 3 (see NORM 12) which is used for all trend programs selected on that card.

The END card signifies the completion of the trend data deck.

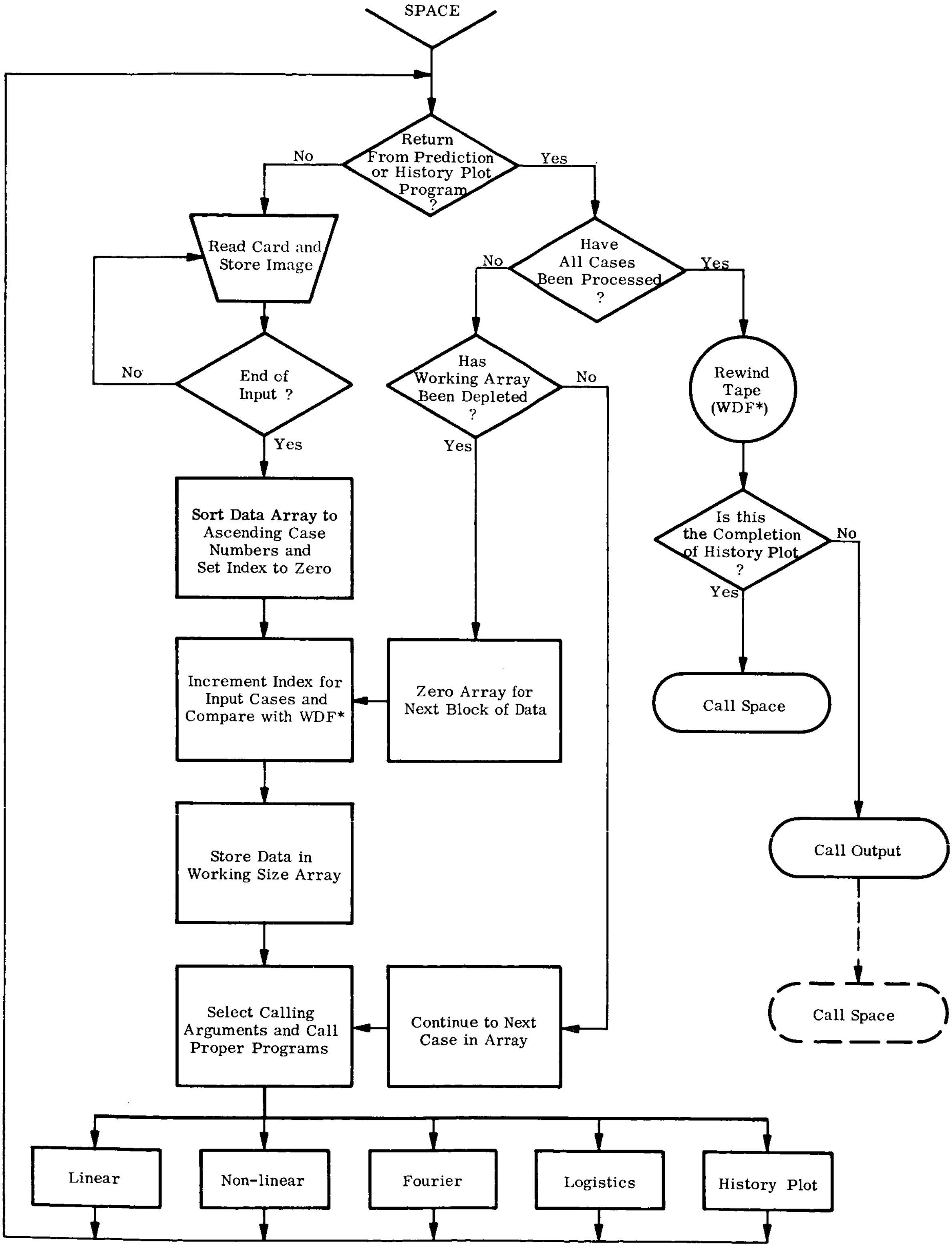
Once GOODE has processed the above cards, control is passed to the Output Routine which after its processing is completed passes control to SPACE.

Example of History Plot Data Deck

CARD COLUMNS

1	4	12
	0611900	
	0651000	
HIS		

The above data deck setup would produce History Plots for the modules 0611900 and 0651000, and upon completion, control would be returned to SPACE.



*WDF is Weight Data File.

```
$IBFTC GOODE    LIST,REF
      INTEGER REELS,WDCT
      COMMON /ACCESS/ HC(100),WDCT,ID(12),PROG
      COMMON /SYSTEM/ NTAPES,REELS(15),CNTRLS(15),FILES(15),LRS(15),
     1                POS(15),TRLPOS(15),RWCNT(15),UNITS(15)
      COMMON /PFILE/NFYLE(4,300),NFY(4)
      COMMON /PROG/ ACTUAL(100) , CALC  (100) , EST   (100) , COM  (12),
     1              LSQR  (100) , MEAN  (100) , MCONF (100) , UW2 (100),
     2              MSQR  (100) , PCONF (100) , S1    (100) , S2  (100),
     3              TIME  (100) , WEIGHT(100) , DOY   (100) , TITLE(9) ,
     4              N , NTOT
      COMMON /BLOCK/  NCASE (10) , TTITLE(90) , AAA   (150) , NANUM(10),
     1                TBLOCK(300), WBLOCK(300), EBLOCK(300) , NNUM (10),
     2                CBLOCK(300), ABLOCK(300), BBLOCK(300)
      COMMON /LOGIS/ KEEP(4),ALP
      COMMON /SIXS/LEEP(2),SALP
      COMMON /STT/N21,D1,D2,J1RUN
      COMMON /HMP/IOPT,JOPT,KOPT,NPATH
      DIMENSION  CINP(4000),NINP(4000)
      DIMENSION  AA2(20),T2(9),TIME2(100),W2(100),EST2(100),CALC2(100),
     1          ACT2(100),B2(100)
      DIMENSION  L1(5),CLIST(5)
      EQUIVALENCE (CINP,NINP)
      DATA  L/0/
      DATA  TILT/4HTILT/,END/3HEND/,BLANK/1H /
      DATA  CLIST   /3H56S,3H57S,3H59S,3H58S,3HHIS/
      DATA  CLISP /5HOUTPT/
      DATA  ININE  /9/
      INTEGER  CASENO,TILT
      IB=0
      IF( L .NE. 0 )  GO TO 412
 399  I=1
 400  K=20*(I-1)+1
      READ(5,4000) XPY2,NINP(K),NINP(K+1),CINP(K+2),CINP(K+3),CINP(K+4),
     1CINP(K+5),CINP(K+6),CINP(K+7),CINP(K+8),CINP(K+9),CINP(K+10),
     2CINP(K+11),CINP(K+12),CINP(K+13),CINP(K+14)
 4000 FORMAT(A3,I7,1XI7,1XA1,1XA1,1XA1,1X4A1,1XF4.2,1XA1,1X2A1,1XF4.2,
     1     .  1X,I1)
      IF (XPY2 .NE. CLIST(5))  GO TO 4001
      IIV = I - 1
      DO 200 LOV=1,IIV
      KL = (20 * (LOV-1)) + 18
      NINP(KL) = 5
  200 CONTINUE
      GO TO 401
 4001 IF (XPY2 .EQ. END)  GO TO 401
      I=I+1
      GO TO 400
 401  IMAX=I-1
      READ (1) N21,D1,D2
      CALL SORT(CINP,IMAX,20,1)
      L=1
  403 READ (1) CASENO,NA2,N2,WHOA2,(AA2(I),I=1,NA2),T2,(TIME2(I),I=1,N2)
```

```
     1,(W2(I),I=1,N2),(EST2(I),I=1,N2),(CALC2(I),I=1,N2),(ACT2(I),I=1,N2
     2),(B2(I),I=1,N2)
      IF (CASENO .EQ. TILT)  GO TO 408
4031  IF( CASENO .LT. NINP(L) )      GO TO 403
      IF( CASENO .EQ. NINP(L) )      GO TO 404
      IF( CASENO .LE. NINP(L+1) )    GO TO 404
      GO TO 408
404   IB=IB+1
      IF( IB .GT. 10 )    GO TO 408
      NCASE(IB)=CASENO
      NANUM(IB)=NA2
      NNUM(IB)=N2
      K1= 15*(IB-1)
      DO 405  I2=1,NA2
      K1=K1+1
405   AAA(K1)=AA2(I2)
      K1=9*(IB-1)
      DO 406  I2=1,9
      K1=K1+1
406   TTITLE(K1)=T2(I2)
      K1=30*(IB-1)
      DO 407   I2=1,N2
      K1=K1+1
      TBLOCK(K1)=TIME2(I2)
      WBLOCK(K1)=W2(I2)
      EBLOCK(K1)=EST2(I2)
      CBLOCK(K1)=CALC2(I2)
      ABLOCK(K1)=ACT2(I2)
407   BBLOCK(K1)=B2(I2)
      GO TO 403
408   NTOT=IB
      K=L
      IF( NTOT  .GT. 10 )      NTOT = 10
      DO 4081  J9=1,5
4081  L1(J9)=0
      IF (XPY2 .NE. CLIST(5))  GO TO 4082
      IF (NINP(K+17) .NE. 0)  L1(5) = 5
      GO TO 412
4082  IF (CINP(K+2) .NE. BLANK)  L1(1) = 1
      IF( CINP(K+3) .NE. BLANK )    L1(2)=2
      KOPT=NINP(K+14)
      IF( CINP(K+4) .EQ. BLANK )    GO TO  410
      L1(3)=3
      DO 409  J9=1,4
      KEEP(J9)=0
      KPOINT= K+4+J9
      IF( CINP(KPOINT)  .NE. BLANK )     KEEP(J9)=J9
409   CONTINUE
      ALP=CINP(K+9)
410   IF( CINP(K+10)  .EQ. BLANK )       GO TO 412
      L1(4)=4
      DO  411  J9=1,2
      LEEP(J9)=0
```

```
      KPOINT=K+10+J9
      IF( CINP(KPOINT) .NE. BLANK )  LEEP(J9)=J9
 411  CONTINUE
      SALP=CINP(K+13)
 412  DO 413 J9=1,5
      IF( L1(J9) .EQ. 0 )      GO TO 413
      NPATH=L1(J9)
      ARG=CLIST(NPATH)
      L1(J9)=0
      CALL SPACE(ARG)
 413  CONTINUE
      L = L + 20
      ILL =  (L-1)/20
      IF( IMAX .NE. ILL )  GO TO 4031
   58 REWIND 1
      L=0
      IF (XPY2 .EQ. CLIST(5))  CALL SPACE
      CALL SPACE (CLISP)
      END
```

MAXIMUM LIKELIHOOD LINEAR - 56S

The program fits a straight line through a set of observations W_i made at time t_i by the method of maximum-likelihood estimation. In the underlying model, each observation W_i is considered an independent and normally distributed random variable having mean Y_i and variance σ_i^2. The mean is assumed to change linearly with time

$$Y_i = a + bt_i$$

and the variance of the i-th observation is assumed to be

$$\sigma_i^2 = S^2 Y_i^2 \left[A_i^2 + r_1^2 C_i^2 + r_2^2 E_i^2\right]$$

where:

- a = intercept of mean.
- b = slope of mean.
- A_i = fraction of W_i which is measured.
- C_i = fraction of W_i which is calculated.
- E_i = fraction of W_i which is estimated.
- S = relative standard deviation of the measured weight.
- $r_1 S$ = relative standard deviation of the calculated weight.
- $r_2 S$ = relative standard deviation of the estimated weight.

It should be noted that individual observations W_i are weighted depending on the weight breakdown and depending on the ratios r_1 and r_2. The choice of these ratios is left to the user; in line with the model, they should be $r_2 \geq r_1 \geq 1$. The effect of the weighting is to consider observations with a greater amount of measured or calculated weight as more significant than others. r_1 and r_2 are functional system parameters, and at the present time, are part of the History data for each functional system. Suggested values are $r_1 = 2$, $r_2 = 5$.

The maximum-likelihood method yields estimators $\hat{a}$, $\hat{b}$, and $\hat{S}$ of the parameters a, b, and S. The estimator of the assumed mean line

$$\hat{Y}_i = \hat{a} + \hat{b}t_i$$

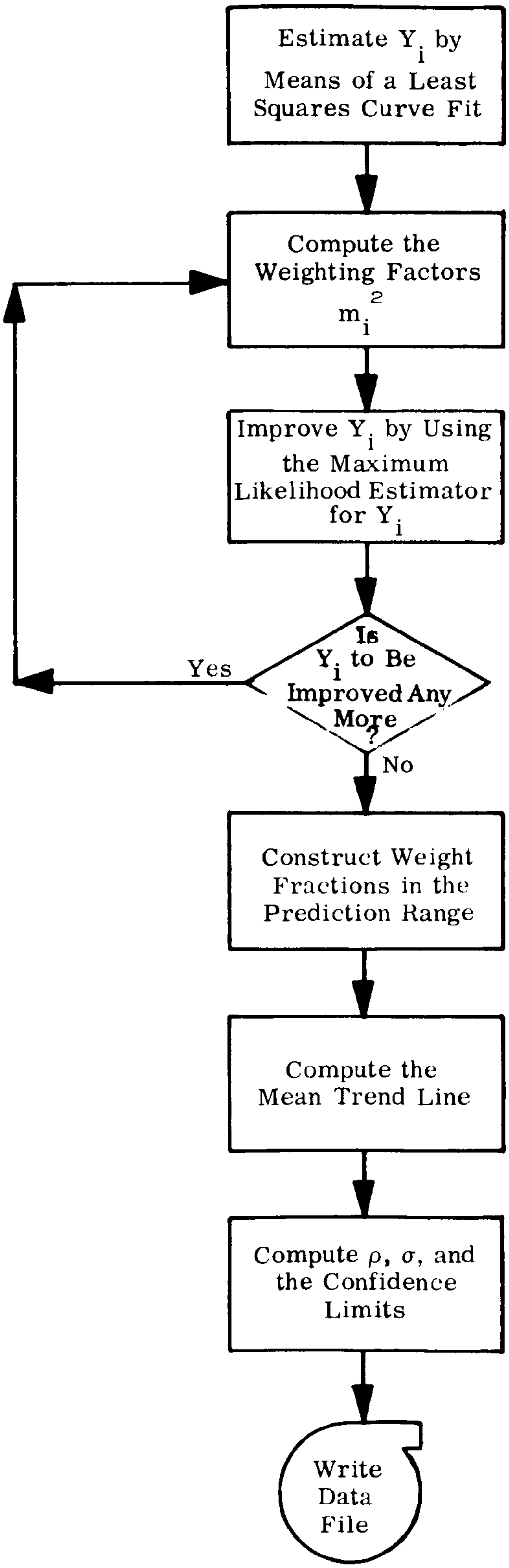
is the predicted weight or curve fit. The 95 percent confidence interval is also computed. The confidence interval is a function of S and is based on small sample methods using Student's t distribution.

Input to 56S consists of a case number and its associated weight data. This input is specified on a data card which is read by the Control Program 55S. For a detailed description of this data card see program 55S.

The output consists of a file of data as is described on the following page.

I.D. RECORD
FILEID FILENO DATE PROG LRS V CASENO MONTH YEAR N P NPROG
$TITLE_1$ . . . $TITLE_9$
$TIME_1$ . . . $TIME_{N+P}$
$WEIGHT_1$ . . . $WEIGHT_{N+P}$
EST_1 . . . EST_{N+P}
$CALC_1$ . . . $CALC_{N+P}$
$ACTUAL_1$ . . . $ACTUAL_{N+P}$
BUY_1 . . . BUY_N
$MEAN_1$ . . . $MEAN_{N+P}$
$PCONF_1$ . . . $PCONF_{N+P}$
$MCONF_1$ . . . $MCONF_{N+P}$
$UW2_1$ . . . $UW2_N$
$PRED_1$ . . . $PRED_P$

FILEID	= label identifier
FILENO	= file number
DATE	= date file was created (MMDDYY)
PROG	= BCD name of program
LRS	= size of logical records within the file
V	= control word for SPACE
CASENO	= case number (integer)
MONTH	= month in which first data point was observed (integer)
YEAR	= year of first observation (integer)
N	= number of observations (integer)
P	= number of predictions (integer)
NPROG	= code word
TITLE	= case title
TIME	= time points
WEIGHT	= normalized weights
EST	= estimated percentages
CALC	= calculated percentages
ACTUAL	= actual percentages
BUY	= nonrandom changes #1 (buyoffs)
MEAN	= mean trend line
PCONF	= +95% confidence limits
MCONF	= nonrandom changes #2 (outliers)
UW2	= observed weights
PRED	= prediction line


Estimate Y_i by Means of a Least Squares Curve Fit
Compute the Weighting Factors m_i^2
Improve Y_i by Using the Maximum Likelihood Estimator for Y_i
Is Y_i to Be Improved Any More ?
Yes
No
Construct Weight Fractions in the Prediction Range
Compute the Mean Trend Line
Compute ρ, σ, and the Confidence Limits
Write Data File

```
$IBFTC LINEAR  LIST,REF
      INTEGER REELS
      INTEGER  CASENO,P,PMONTH,PYEAR,YEAR
      REAL     LSQR,MEAN,MCONF,MSQR
      COMMON /ACCESS/ HC(100),WDCT,ID(12),PROG
      COMMON /SYSTEM/ NTAPES,REELS(15),CNTRLS(15),FILES(15),LRS(15),
     1                 POS(15),TRLPOS(15),RWCNT(15),UNITS(15)
      COMMON  /PFILE/ NFYLE(4,300),NFY(4)
      COMMON /PPOG/ ACTUAL(100) , CALC   (100) , EST    (100) , COM   (12),
     1              LSQR   (100) , MEAN   (100) , MCONF  (100) , UW2 (100),
     2              MSQR   (100) , PCONF  (100) , S1     (100) , S2   (100),
     3              TIME   (100) , WEIGHT(100) , DGY    (100) , TITLE(9) ,
     4              N , NTOT
      DIMENSION STUDNT(31)
      EQUIVALENCE  (COM(1),PMONTH), (COM(2),PYEAR), (COM(3),MONTH),
     2             (COM(4),YEAR)   , (COM(5),R1)    , (COM(6),R2)    ,
     3             (COM(7),ASUBP)  , (COM(8),BSUBP), (COM(9),CSUBP),
     4              (COM(10),NSUBP)
      DATA  STUDNT  /12.706,4.303,3.182,2.776,2.571,2.447,2.365,2.306,
     1                 2.262,2.228,2.201,2.179,2.160,2.145,2.131,2.120,
     2                 2.110,2.101,2.093,2.086,2.080,2.074,2.069,2.064,
     3                 2.060,2.056,2.052,2.048,2.045,2.042,1.960/
      DATA  NPROG/3HLIN/
      DO  1300  I1=1,NTOT
      CALL GETDAT(IPSWT,I1,CASENO)
      BPRIM = 0.0
      CPRIM = 0.0
      EPRIM = 0.0
      FPRIM = 0.0
      DO 1050 I=1,N
      BPRIM = BPRIM + TIME(I)
      CPRIM = CPRIM + TIME(I)**2
      EPRIM = EPRIM + WEIGHT(I)
 1050 FPRIM = FPRIM + WEIGHT(I) * TIME(I)
      DPRIM = FLOAT(N) * CPRIM - BPRIM**2
      ATWIST = (EPRIM*CPRIM - BPRIM*FPRIM)/DPRIM
      BTWIST = (FLOAT(N)*FPRIM - BPRIM*EPRIM)/DPRIM
      DO 1057 I=1,N
      LSQR(I) = ATWIST + BTWIST*TIME(I)
 1057 MSQR(I) = LSQR(I)**2 * (ACTUAL(I)**2 + R1**2 * CALC(I)**2 +
     1          R2**2 * EST(I)**2)
      ITERAT = 3
 1062 A = 0.0
      B = 0.0
      C = 0.0
      E = 0.0
      F = 0.0
      DO 1074 I=1,N
      A = A + 1.0 / MSQR(I)
      B = B + TIME(I) / MSQR(I)
      C = C + TIME(I)**2 / MSQR(I)
      E = E + WEIGHT(I) / MSQR(I)
```

Reference No. 56.4
Issue Date 23 Dec 1965
Supersedes 13 August 1965

```
1074 F = F + WEIGHT(I) * TIME(I) / MSQR(I)
     D = A*C - B**2
     AHAT = (E*C - F*B) / D
     BHAT = (A*F - B*E) / D
     DO 1102 I=1,N
1102 MEAN(I) = AHAT + BHAT*TIME(I)
     ITERAT = ITERAT - 1
     IF (ITERAT .EQ. 0) GO TO 1114
     DO 1111 I=1,N
1111 MSQR(I) = MEAN(I)**2 * (ACTUAL(I)**2 + R1**2 * CALC(I)**2 +
    1            R2**2 * EST(I)**2)
     GO TO 1062
1114 NPLUS1 = N + 1
     NPLUSP= 12*(PYEAR - YEAR ) + PMONTH - MONTH +1
     K85=TIME(N)
     P=NPLUSP-K85
     NPLUSP=N+P
     IF( IPSWT .GT. 6 )   GO TO 1125
     DO 1124 K=NPLUS1,NPLUSP
     EST(K) = EST(N)
     CALC(K) = CALC(N)
1124 ACTUAL(K) = ACTUAL(N)
     GO TO 1164
1125 IF( NSUBP .NE. 0 )   GO TO 1143
     DO 1140 K=NPLUS1,NPLUSP
     EST(K) = CSUBP
     CALC(K) = BSUBP
1140 ACTUAL(K) = ASUBP
     GO TO 1164
1143 DO 1146 K=NPLUS1,NPLUSP
     EST(K) = (CSUBP - EST(N)) * FLOAT(K-N) / FLOAT(P) + EST(N)
     CALC(K) = (BSUBP - CALC(N)) * FLOAT(K-N) / FLOAT(P) + CALC(N)
1146 ACTUAL(K) = (ASUBP-ACTUAL(N)) * FLOAT(K-N) / FLOAT(P) + ACTUAL(N)
1164 DO 1167 K=NPLUS1,NPLUSP
     TIME(K) = TIME(K-1) + 1.0
     LSQR(K) = ATWIST + BTWIST * TIME(K)
1167 MEAN(K) = AHAT + BHAT * TIME(K)
     SUM = 0.0
     DO 1173 I=1,N
1173 SUM = SUM + (WEIGHT(I) - MEAN(I))**2 /MSQR(I)
     RHO = SQRT(1.0/FLOAT(N-2)*SUM)
     K = N - 2
     IF (K .GT. 31)  K = 31
     DO 1215 I=1,NPLUSP
     S1(I) = RHO * MEAN(I) * SQRT(ACTUAL(I)**2 +        CALC(I)**2
    1 +  EST(I)**2)
     ALPHA = (C - B * TIME(I)) / D
     ALPHA2 = ALPHA**2
     BETA = (A * TIME(I) -B) / D
     BETA2 = BETA**2
     ALPHAA = ALPHA2 * A
     BETAC = BETA2 * C
     ALBEB2 = 2.0 * ALPHA * BETA * B
```

```
      S2(I) = RHO * SQRT(ALPHAA + BETAC + ALBEB2)
      DENO = STUDNT(K) * SQRT(S1(I)**2 + S2(I)**2)
 1215 PCONF(I) = MEAN(I) + DENO
      NFY(1) = NFY(1) + 1
      NF1 = NFY(1)
      NF11=ISIGN(NF1,REELS(1))
      NFILE=REELS(1)*1000 + NF11
      NFYLE(1,NF1) = CASENO
      DIFRC = UW2(N) - MEAN(N)
      DO 1216 IW=NPLUS1,NPLUSP
      WEIGHT(IW) = MEAN(IW)
 1216 UW2(IW) = WEIGHT(IW) + DIFRC
      ID(7)=CASENO
      ID(8)= MONTH
      ID(9)= YEAR
      ID(10)=N
      ID(11)=P
      ID(12)=NPROG
      CALL ABOUT1(NFILE,1)
      CALL ABOUT2(NFILE,TITLE,9)
      CALL ABOUT2(NFILE,TIME,NPLUSP)
      CALL ABOUT2 (NFILE,WEIGHT,NPLUSP)
      CALL ABOUT2(NFILE,EST,NPLUSP)
      CALL ABOUT2(NFILE,CALC,NPLUSP)
      CALL ABOUT2(NFILE,ACTUAL,NPLUSP)
      CALL ABOUT2(NFILE,BUY,N)
      CALL ABOUT2(NFILE,MEAN,NPLUSP)
      CALL ABOUT2(NFILE,PCONF,NPLUSP)
      CALL ABOUT2(NFILE,MCONF,NPLUSP)
 1217 CALL ABOUT2 (NFILE, UW2, NPLUSP)
 1300 CONTINUE
      CALL GOODE
      END
```

MAXIMUM LIKELIHOOD NONLINEAR - 57S

The data available for curve fitting consists of a set of observations W_i made at time t_i. The expected value of W_i is assumed to be an exponential function of time,

$$E\{W_i\} = Y_i = a - be^{-ct_i} \qquad c > 0$$

where a, b, c are parameters to be determined. So that the curve will approach the value a asymptotically, c has been restricted to positive values. A range of c values has been established, and for each c in the range, a curve of the above form is fit to the observations W_i by the method of maximum-likelihood estimation. For each curve so fitted, values of a and b are determined, and the natural logarithm $\overline{L}$ of the likelihood function is computed. When all c values in the range have been exhausted, the maximum $\overline{L}$ is determined through inspection, and the corresponding values of a, b, c are the desired parameters.

The analysis and further assumptions used in the model were identical to those in the linear model described in the previous section. The maximum-likelihood method yields estimators $\hat{a}$, $\hat{b}$, $\hat{S}$ of the parameters a, b, S while c is determined from empirical considerations. The estimator of the assumed mean line

$$\hat{Y}_i = \hat{a} - \hat{b}e^{-ct_i}$$

is the predicted weight or curve fit. The 95 percent confidence interval is also computed. The confidence interval is a function of S and is based on small sample methods using Student's t distribution.

The range of c values established within the program limits c to the following values:

$$c = k(.001) \qquad k = 1, 2, \ldots, 2000$$

Thus the greatest value that c can attain is the valuc $c = 2$, while the minimum value of c is the value $c = .001$. When $c = 2$, the exponential very quickly approaches the asymptote a. For example, if c = 2, t = 5 then

$$Y_i = a - be^{-ct_i} \qquad \text{all } i$$

$$Y_5 = a - b(.000045) \qquad i = 5$$

$$Y_i \approx a \qquad i \geq 5$$

For any value of $c > 2$, this same phenomenon occurs even more quickly and since the curve is essentially the constant $Y = a$, no loss will result by restricting $c \leq 2$. Likewise, when $c = .001$, the curve resembles a straight line with a steep slope and will exhibit no exponential trend until the value of t becomes quite large. For example if $c = .001$, then the curve will not exhibit the same tendencies as that shown in the preceding example until t = 10, 000 months. For $0 < c < .001$ this phenomenon is even more pronounced. Thus, it does not seem unreasonable to restrict c to the above values.

Input to 57S consists of a case number and its associated weight data. This input is specified on a data card which is read by the Control Program - 55S. For a detailed description of this data card see program 55S. Output consists of a file of data as is described on the following page.

I.D. RECORD:

- FILEID
- FILENO
- DATE
- PROG
- LRS
- V
- CASENO
- MONTH
- YEAR
- N
- P
- NPROG

$TITLE_1$ . . . $TITLE_9$
$TIME_1$ . . . $TIME_{N+P}$
$WEIGHT_1$ . . . $WEIGHT_{N+P}$
EST_1 . . . EST_{N+P}
$CALC_1$ . . . $CALC_{N+P}$
$ACTUAL_1$ . . . $ACTUAL_{N+P}$
BUY_1 . . . BUY_N
$MEAN_1$ . . . $MEAN_{N+P}$
$PCONF_1$ . . . $PCONF_{N+P}$
$MCONF_1$ . . . $MCONF_{N+P}$
$UW2_1$ . . . $UW2_N$
$PRED_1$ . . . $PRED_P$

FILEID = label identifier

FILENO = file number

DATE = date file was created (MMDDYY)

PROG = BCD name of program

LRS = size of logical records within the file

V = control word for SPACE

CASENO = case number (integer)

MONTH = month in which first data point was observed (integer)

YEAR = year of first observation (integer)

N = number of observations (integer)

P = number of predictions (integer)

NPROG = code word

TITLE = case title

TIME = time points

WEIGHT = normalized weights

EST = estimated percentages

CALC = calculated percentages

ACTUAL = actual percentages

BUY = nonrandom changes #1 (buyoffs)

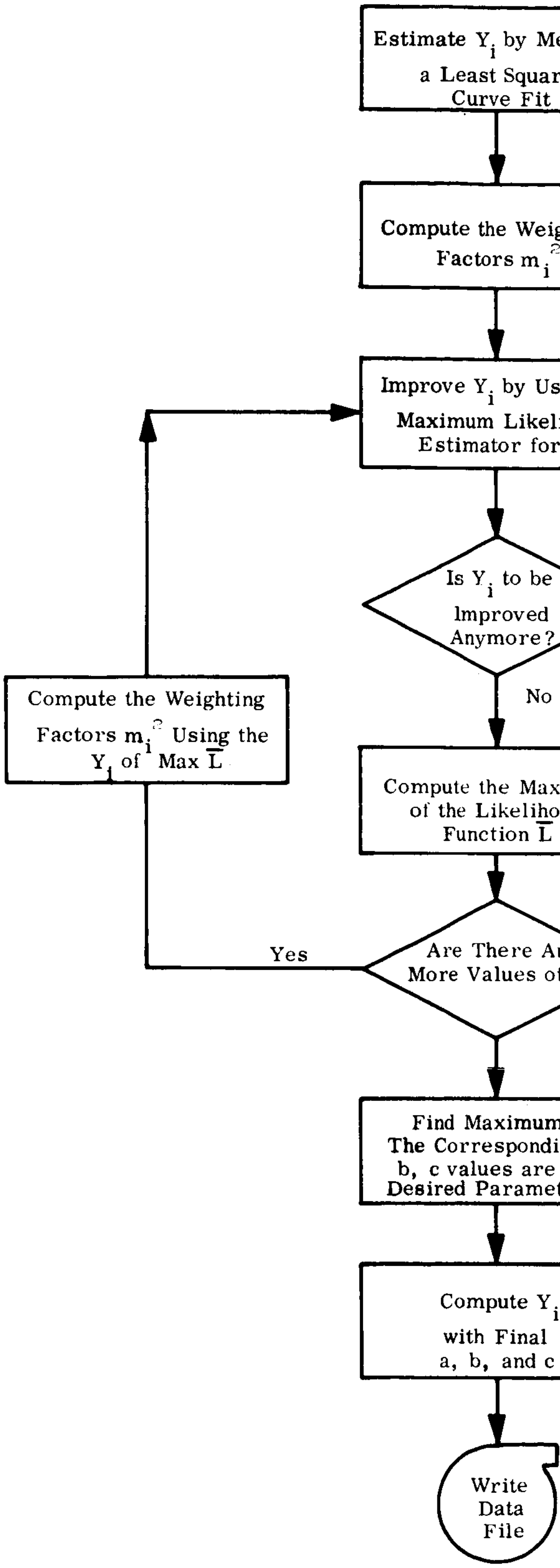
MEAN = mean trend line

PCONF = +95% confidence limits

MCONF = nonrandom changes #2 (outliers)

UW2 = observed weights

PRED = prediction line


Estimate Y_i by Means of a Least Squares Curve Fit
Compute the Weighting Factors m_i^2
Improve Y_i by Using the Maximum Likelihood Estimator for Y_i
Is Y_i to be Improved Anymore?
Yes
No
Compute the Weighting Factors m_i^2 Using the Y_i of Max L̄
Compute the Maximum of the Likelihood Function L̄
Are There Any More Values of C?
Yes
Find Maximum L̄. The Corresponding a, b, c values are the Desired Parameters.
Compute Y_i with Final a, b, and c
Write Data File

```
$IBFTC NONLNR  LIST,REF
      INTEGER REELS
      INTEGER  CASENO,P,PMONTH,PYEAR,YEAR
      COMMON /ACCESS/ HC(100),WDCT,ID(12),PROG
      COMMON /SYSTEM/ NTAPES,REELS(15),CNTRLS(15),FILES(15),LRS(15),
     1                POS(15),TRLPOS(15),RWCNT(15),UNITS(15)
      COMMON  /PROG/ A(100),B(100),D(100),COM(12),LSQR(100),YBAR(100),
     1               MCONF(100),OW2(100),MSQR(100),PCONF(100),S1(100),
     2               S2(100),T(100),Y(100),BUY(100),TITLE(9),N,NTOT
      COMMON  /PFILE/ NFYLE(4,300),NFY(4)
      DIMENSION  EM(100),YTEST(100),SQM(100),DEV(100),STUDNT(31)
      EQUIVALENCE  (COM(1),PMONTH), (COM(2),PYEAR), (COM(3),MONTH),
     2             (COM(4),YEAR)  , (COM(5),R1)   , (COM(6),R2)   ,
     3             (COM(7),ASUBP) , (COM(8),BSUBP), (COM(9),CSUBP),
     4              (COM(10),NSUBP)
      DATA STUDNT   /12.706,4.303,3.182,2.776,2.571,2.447,2.365,2.306,
     1              2.262,2.228,2.201,2.179,2.160,2.145,2.131,2.120,
     2              2.110,2.101,2.093,2.086,2.080,2.074,2.069,2.064,
     3              2.060,2.056,2.052,2.048,2.045,2.042,1.960/
      DATA   PI    /0202622077325/
      DATA NPROG/3HEXP/
      CALL TRAPOK
      DO  1300  I1=1,NTOT
      CALL GETDAT(IPSWT,I1,CASENO)
      NPLUS1 = N + 1
      NPLUSP = 12*( PYEAR - YEAR ) + PMONTH - MONTH + 1
      K85=T(N)
      P=NPLUSP-K85
      NPLUSP=N+P
      DO 450 I=NPLUS1,NPLUSP
  450 T(I)=T(I-1) + 1.
      H=0.
      TSQ=0.
      S=0.
      YT=0.
      DO 500   I=1,N
      H=H + T(I)
      TSQ=TSQ + T(I)**2
      S=S + Y(I)
  500 YT=YT + Y(I) * T(I)
      HALF=(TSQ * S - H * YT)/(FLOAT(N) * TSQ - H**2)
      HBET=(FLOAT(N) * YT - H * S)/(FLOAT(N) * TSQ - H**2)
      DO 600   J=1,N
  600 YBAR(J)=HALF + HBET * T(J)
      DELC=1.
      C=.1
      DO 1400  I2=1,3
      DELC=DELC * .1
      IF(C .EQ. 0.) C=DELC
      DO 1100  J1=1,20
      DO 810 K=1,5
      ONE=0.
```

```
      TWO=0.
      FOR=0.
      FIV=0.
      SIX=0.
      DO 700   M=1,N
      SQM(M)=YBAR(M)**2 * (A(M)**2 + R1**2*B(M)**2 + R2**2 * D(M)**2)
      ONE=ONE + 1./SQM(M)
      TWO=TWO - EXP(-C * T(M))/SQM(M)
      FOR=FOR - EXP(-2. * C * T(M))/SQM(M)
      FIV=FIV + Y(M)/SQM(M)
  700 SIX=SIX + Y(M) * EXP(-C * T(M))/SQM(M)
      TRE= -TWO
      AHAT=(FOR * FIV - TWO * SIX)/(ONE * FOR - TWO * TRE)
      BHAT=(ONE * SIX - TRE * FIV)/(ONE * FOR - TWO * TRE)
      DO 800 M=1,N
  800 YBAR(M)=AHAT - BHAT * EXP(-C * T(M))
  810 CONTINUE
      SSQ=0.
      EMMS=0.
      DO 900   II=1,N
      EM(II)=SQRT(SQM(II))
      EMMS=EMMS +ALOG(EM(II))
  900 SSQ=SSQ + (Y(II) - YBAR(II))**2/SQM(II)
      SSQ=SSQ/FLOAT(N-2)
      ELBAR= -FLOAT(N) * ALOG(2.* PI)/2. - FLOAT(N) * ALOG(SSQ)/2.-EMMS
      IF(J1 .EQ. 1) GO TO 1050
      IF(ELBAR .LT. ELMAX) GO TO 1060
 1050 CTEST=C
      ATEST=AHAT
      BTEST=BHAT
      ELMAX=ELBAR
      SAV1=ONE
      SAV3=TRE
      SAV4=FOR
      SAVS=SSQ
      DO 1055 N1=1,N
 1055 YTEST(N1)=YBAR(N1)
 1060 C=C + DELC
 1100 CONTINUE
      DO 1110 M1=1,N
 1110 YBAR(M1)=YTEST(M1)
      C=CTEST - DELC
 1400 CONTINUE
 1410 DO 1500 KK=1,NPLUSP
 1500 YBAR(KK)=ATEST - BTEST * EXP(-CTEST * T(KK))
      K=N - 2
      IF(K .GT. 31) K=31
      DO 1510 LL=NPLUS1,NPLUSP
      A(LL) = (ASUBP - A(N)) * FLOAT(LL-N)/FLOAT(P) + A(N)
      B(LL) = (BSUBP - B(N)) * FLOAT(LL-N)/FLOAT(P) + B(N)
      D(LL) = (CSUBP - D(N)) * FLOAT(LL-N)/FLOAT(P) + D(N)
      SQM(LL)=YBAR(LL)**2 * (A(LL)**2 +       B(LL)**2 +       D(LL)**2)
      DENOM=SAV1*(-SAV4)- SAV3**2
```

```
      ALPHA=(-SAV4 -(SAV3*EXP(-CTEST* T(LL))))/DENOM
      BETA=(SAV1 * EXP(-CTEST * T(LL))- SAV3)/ DENOM
      ALPHA2 = ALPHA ** 2
      BETA2  = BETA ** 2
      ALPHAA = ALPHA2 * SAV1
      BETAD  = BETA2  * (-SAV4)
      ALBEC2 = 2.* ALPHA * BETA * SAV3
      DEV(LL)=SQRT(SAV5) * SQRT(SQM(LL) + ALPHAA + BETAD + ALBEC2)
 1510 PCONF(LL)=YBAR(LL) + STUDNT(K) * DEV(LL)
      NFY(2) = NFY(2) + 1
      NF2 = NFY(2)
      NF22=ISIGN(NF2,REELS(2))
      NFILE =REELS(2)*1000 + NF22
      NFYLE(2,NF2) = CASENO
      DIFRC = UW2(N) - YBAR(N)
      DO 666 MOM = NPLUS1,NPLUSP
      Y(MOM) = YBAR(MOM)
  666 UW2(MOM) = Y(MOM) + DIFRC
      ID(7)=CASENO
      ID(8)= MONTH
      ID(9)= YEAR
      ID(10)=N
      ID(11)=P
      ID(12)=NPROG
      CALL ABOUT1(NFILE,1)
      CALL ABOUT2(NFILE,TITLE,9)
      CALL ABOUT2(NFILE,T,NPLUSP)
      CALL ABOUT2 (NFILE,Y,NPLUSP)
      CALL ABOUT2(NFILE,D,NPLUSP)
      CALL ABOUT2(NFILE,B,NPLUSP)
      CALL ABOUT2(NFILE,A,NPLUSP)
      CALL ABOUT2(NFILE,BUY,N)
      CALL ABOUT2(NFILE,YBAR,NPLUSP)
      CALL ABOUT2(NFILE,PCONF,NPLUSP)
      CALL ABOUT2(NFILE,MCONF,NPLUSP)
 1217 CALL ABOUT2 (NFILE, UW2, NPLUSP)
 1300 CONTINUE
      CALL GOODE
      END
```

ADAPTIVE (FOURIER) EXPONENTIAL - 58S

The program accepts a set of weights W_i observed at time t_i and predicts the trend of the data at a future date. The model does not assume that the observations are dispersed about a specified mean-value function (as do the other trend prediction programs). Instead, it is adaptive, meaning that it senses the latest tendencies of the trend and projects them into the prediction range. To get away from the mean value function, one must look at the weight increment from the latest observation to the next one. This increment is considered a random variable.

The model chosen was one which expressed the same time-dependent behavior desired in all the models, but which was not autoregressive in its structure. The model is,

$$W_{i+1} - W_i = a + be^{-\alpha t_i} + R_i$$

where a is the mean increment and R_i is a random residual of zero mean. When observations are equally spaced, namely $t_{i+1} = t_i + \Delta t$, the above equation becomes

$$W_{i+1} = (W_1 - C) + ia + ce^{-\alpha t_i} + \sum_{j=1}^{i} R_j$$

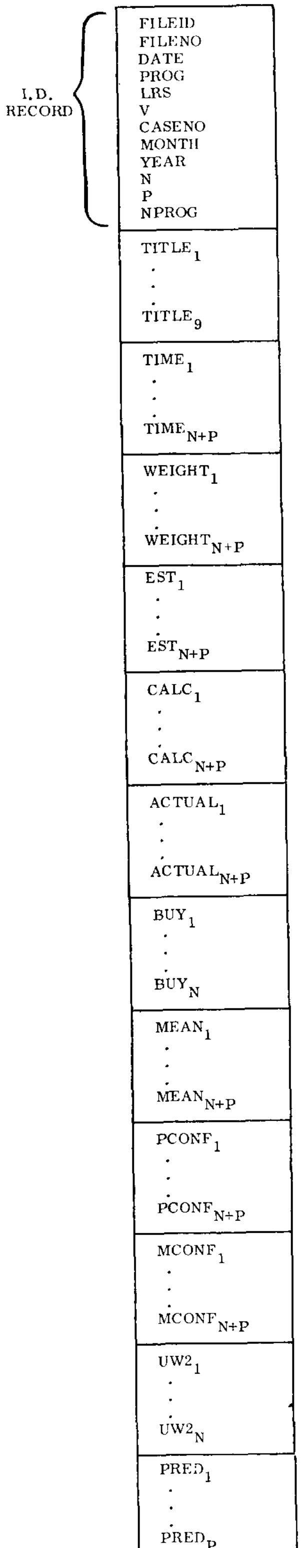
where

$$c = b(1 - e^{\alpha \Delta t})^{-1}$$

and a, c, α are parameters to be estimated. It was decided to separate the nonlinear part of the above equation by a Fourier analysis. Hence the name Fourier model.

Input to 58S consists of a case number, its associated weight data, and a weighting coefficient ρ, defined by $\rho^{n-1} = k$, $0 \le k \le 1$. The parameter k must be input to the program and is specified in columns 41-44 of the instruction card which is read by the Control Program 55S. This weighting coefficient allows the user the opportunity to weight later observations more heavily than earlier observations. If equal weighting is desired, use $\rho = k = 1$.

Output consists of a file of data as is described on the following page.



FILEID = label identifier

FILENO = file number

DATE = date file was created (MMDDYY)

PROG = BCD name of program

LRS = size of logical records within the file

V = control word for SPACE

CASENO = case number (integer)

MONTH = month in which first data point was observed (integer)

YEAR = year of first observation (integer)

N = number of observations (integer)

P = number of predictions (integer)

NPROG = code word

TITLE = case title

TIME = time points

WEIGHT = normalized weights

EST = estimated percentages

CALC = calculated percentages

ACTUAL = actual percentages

BUY = nonrandom changes #1 (buyoffs)

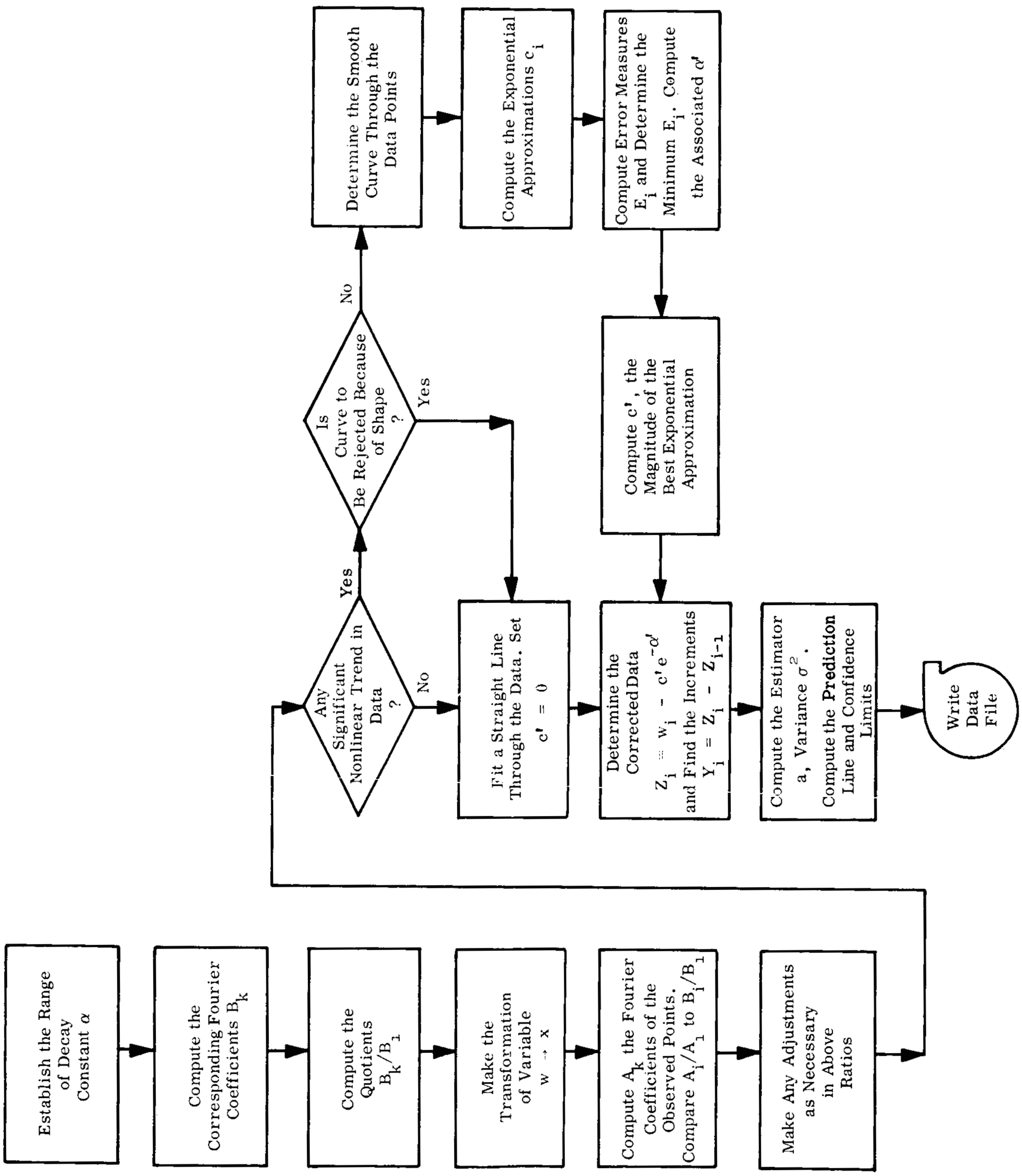
MEAN = mean trend line

PCONF = +95% confidence limits

MCONF = nonrandom changes #2 (outliers)

UW2 = observed weights

PRED = prediction line


Establish the Range of Decay Constant α
Compute the Corresponding Fourier Coefficients B_k
Compute the Quotients B_k/B_1
Make the Transformation of Variable $w \to x$
Compute A_k the Fourier Coefficients of the Observed Points. Compare A_i/A_1 to B_i/B_1
Make Any Adjustments as Necessary in Above Ratios
Any Significant Nonlinear Trend in Data ?
Yes
No
Is Curve to Be Rejected Because of Shape ?
No
Yes
Determine the Smooth Curve Through the Data Points
Compute the Exponential Approximations c_i
Compute Error Measures E_i and Determine the Minimum E_i. Compute the Associated α'
Compute c', the Magnitude of the Best Exponential Approximation
Fit a Straight Line Through the Data. Set c' = 0
Determine the Corrected Data $Z_i = w_i - c' e^{-\alpha'}$ and Find the Increments $Y_i = Z_i - Z_{i-1}$
Compute the Estimator a, Variance σ^2. Compute the Prediction Line and Confidence Limits
Write Data File

```
$IBFTC FOURNR  LIST,REF
      INTEGER REELS
      INTEGER  CASENO,P,PMONTH,PYEAR,YEAR
      REAL  MEAN,MCONF
      COMMON /ACCESS/ HC(100),WDCT,ID(12),PROG
      COMMON /SYSTEM/ NTAPES,REELS(15),CNTRLS(15),FILES(15),LRS(15),
     1                POS(15),TRLPOS(15),RWCNT(15),UNITS(15)
      COMMON  /PFILE/ NFYLE(4,300),NFY(4)
      COMMON /PPOG/ ACTUAL(100) , CALC  (100) , EST   (100) , COM  (12),
     1              LSQR  (100) , MEAN  (100) , MCONF (100) , UW2 (100),
     2              MSQR  (100) , PCONF (100) , S1    (100) , S2  (100),
     3              TIME  (100) , WEIGHT(100) , BUY   (100) , TITLE(9) ,
     4              N , NTOT
      COMMON /SIXS/LEEP(2),SALP
      DIMENSION  Q2(5,13)  ,  AS(5)    ,  F1(100)   ,  F2(100) ,  F3(100),
     1           F4(100)   , DIF(100) ,  U(100)    ,  UN(100) ,BKALP(5) ,
     2            A(5)     ,  X(100) ,   Q(5)     ,  AL(13)  ,   E(13) ,
     3            C(13)    ,  Z(100) ,   Y(100)   ,  BK(5,13),  AM(13)
      DATA NPROG/3HFOR/
      EQUIVALENCE  (COM(1),PMONTH), (COM(2),PYEAR), (COM(3),MONTH),
     2             (COM(4),YEAR)  , (COM(5),R1)   , (COM(6),R2)   ,
     3             (COM(7),ASUBP) , (COM(8),BSUBP), (COM(9),CSUBP),
     4              (COM(10),NSUBP)
      DATA PIE/0202622077325/
      DO 71 J=1,13
      EJ=J
      Z2= .7 - (EJ-1.)* .05
   71 AL(J)=-ALOG(Z2)
      DO 10 I=1,13
      DO 11  K=1,5
      EK=K
      PIEK=PIE*EK
      P1 =  1./(1. +  (PIEK/AL(I))**2 )
      P2 = ((-1.)**K)*EXP(-AL(I))-1.
      BK(K,I)= (2./PIEK)*P1*P2
   11 CONTINUE
   10 CONTINUE
      DO 102  I=1,13
      DO 102  K=1,5
  102 Q2(K,I)= BK(K,I)/BK(1,I)
      DO 103  I=1,5
  103 AS(I)=Q2(I,13)
      DO  1300 I1=1,NTOT
      CALL GETDAT(IPSWT,I1,CASENO)
    6  IF( I   .LT.  8  )  GO TO 1300
      NPLUSP=12*(PYEAR-YEAR)+PMONTH-MONTH+1
   60 NPLUS1=N+1
      K85=TIME(N)
      P=NPLUSP-K85
      NPLUSP=N+P
      L=N-1
      V=L
```

```
      RHO=SALP
      DO 117  I=1,N
  117 U(I) = WEIGHT(I)
 122  CONTINUE
      DO 4 I=1,5
      A(I)=0.
  4   CONTINUE
      DO 41 I=1,N
      X(I)= U(I)-((FLOAT(N-I))/V)*U(1) -((FLOAT(I-1))/(FLOAT(N-1)))*U(N)
  41  CONTINUE
      DO 61 K=1,5
      EK=K
      DO 5  I=2,L
      ARG= (EK*PIE*FLOAT(I-1))/V
      A(K)=A(K) + X(I)*SIN(ARG)
  5   CONTINUE
      A(K)= A(K)*(2./V)
  61  CONTINUE
      D=U(N)-U(1)
      QO= A(1)/D
      DO 7 I=1,5
  7   Q(I)= A(I)/A(1)
      DO 104  I=1,5
      IF(  Q(I)  .GE. 0. )      GO TO 1030
      A(I)=0.
      GO TO 104
 1030 IF( Q(I) .GT. AS(I) )     A(I)= AS(I)*A(1)
 104  CONTINUE
      IF( (A(1)/D )  .GE.  .05  )        GO TO  1882
 2224 CONTINUE
      DO 107  I=1,N
 107  MEAN(I)=U(1) + D*(FLOAT(I-1)/FLOAT(N-1))
      CPRIME = 0.
      GO TO 18
 2227 ALP=AL(K)/V
      GO TO 160
 1882 CONTINUE
      DO 8  I=1,N
      EK=I-1
      F1(I)=A(1)* SIN( PIE*(EK/V))
      F2(I)=F1(I)+ A(2)*SIN( 2.*PIE*(EK/V))
      F3(I)=F2(I)+ A(3)*SIN( 3.*PIE*(EK/V))
      F4(I)=F3(I)+ A(4)*SIN( 4.*PIE*(EK/V))
  8   CONTINUE
      DO 9 I=1,N
      MEAN(I)=U(1) + D*(FLOAT(I-1)/FLOAT(N-1)) + F4(I)
  9   CONTINUE
      DO 1117  I=1,13
 1117 C(I)= (-D)/(1.-EXP(-AL(I)))
      LPASS=0
      IF(  D .LT. 0. .AND.  A(1) .LT. 0. )   GO TO  2222
      DO  110  I=1,13
 110  C(I)= (-.8*D)/(1.-EXP(-AL(I)))
```

```
      LPASS=1
 2222 CONTINUE
      DO  111  I=1,13
      AONE=0.
      ATWO=0.
      DO  112  K=1,5
      AONE=AONE +(BK(K,I)*C(I))**2
 112  ATWO=ATWO +( BK(K,I)*C(I)*A(K))
      E(I)= AONE - 2.*ATWO
 111  CONTINUE
      EMIN=10.E+10
      DO 14 K=1,13
      IF( EMIN .LT. E(K) )  GO TO 14
      I=K
      EMIN=E(K)
  14  CONTINUE
      ALP=AL(I)/V
      K=1
      IF( LPASS .EQ. 1 )       GO TO  2228
      FACT= .8*U(1)
      IF( C(13) .GT. FACT)     GO TO  2224
      DO  2225  K=1,13
      IF( C(K)    .LE.   FACT )    GO TO 2226
 2225 CONTINUE
 2226 IF( I   .LT. K )      GO TO  2227
 2228 IF( I   .EQ. K   .OR. I .EQ. 13 )    GO TO 160
      DO 147 K=1,13
 147  AM(K)=AL(K)/V
      ANUM=(E(I)-E(I-1))*(AM(I+1)-AM(I-1))**2
     1    -(E(I+1)-E(I-1))*(AM(I)-AM(I-1))**2
      DNUM=(AM(I)-AM(I-1))*(E(I+1)-E(I-1))
     1    -(AM(I+1)-AM(I-1))*(E(I)-E(I-1))
      ALP=-.5*(ANUM/DNUM)+AM(I-1)
 160  ANUM=0.
      DNUM=0.
      PLA=ALP*V
      DO 17  K=1,5
      EK=K
      PIEK=PIE*EK
      BKALP(K)= (2./PIEK)*( 1./(1.+(PIEK/PLA)**2))*((-1.)**K*EXP(-PLA)
     1         -1.)
      ANUM=ANUM + BKALP(K)*A(K)
      DNUM=DNUM + BKALP(K)**2
  17  CONTINUE
      CPRIME=  ANUM/DNUM
  18  DO  19  I=1,N
  19  Z(I)=U(I)-CPRIME*EXP(-ALP*FLOAT(I-1))
      DO 191  I=2,N
 191  Y(I)=Z(I)-Z(I-1)
      RHO= (ALOG(RHO))/V
      RHO= EXP(RHO)
      ANUM=0.
      DNUM=0.
```

```
      DO 20 I=2,N
      ANUM=ANUM +  Y(I)*RHO**(N-I)
 20   DNUM=DNUM +  RHO**(N-I)
      AHAT= ANUM/DNUM
      SIGMA=0.
      DO 21  I=2,N
21    SIGMA=SIGMA + RHO**(N-I)*(Y(I)-AHAT)**2
      SIGMA=  SQRT ( SIGMA/FLOAT(N-2))
      BETA= (Z(N)+AHAT)/Z(N)
      LSET=0
      DO 22  K=NPLUS1,NPLUSP
      TIME(K)=TIME(K-1)+1.
      NT(K)=NT(K-1)+1
      IF( AHAT .LT. 0. )  GO TO 6527
      MEAN(K)=Z(N)+FLOAT(K-N)*AHAT + CPRIME*EXP(-ALP*FLOAT(K-1))
      GO TO 6529
6527   IF( LSET .EQ. 1 )  GO TO 6528
      AMUTH=BETA**(K-N)
      IF( AMUTH .GT. (.000001))   GO TO 6528
      LSET=1
      AMUTH=0.
6528  MEAN(K)=AMUTH * Z(N)+CPRIME*EXP(-ALP*FLOAT(K-1))
6529  PCONF(K)=MEAN(K)+1.64*SQRT(FLOAT(K-N)*SIGMA**2)
 22   CONTINUE
      NFY(3) = NFY(3) + 1
      NF3 = NFY(3)
      NF33=ISIGN(NF3,REELS(3))
      NFILE=REELS(3)*1000 + NF33
      NFYLE(3,NF3) = CASENO
      DIFRC = UW2(N) - MEAN(N)
      FLP = P
      DO 1216 IW=NPLUS1,NPLUSP
      FKN = IW - N
      EST(IW) = (CSUBP-EST(N)) * FKN / FLP + EST(N)
      CALC(IW) = (DSUBP-CALC(N)) * FKN / FLP + CALC(N)
      ACTUAL(IW) = (ASUBP-ACTUAL(N)) * FKN / FLP + ACTUAL(N)
      U(IW) = MEAN(IW)
1216  UW2(IW) = U(IW) + DIFRC
      ID(7)=CASENO
      ID(8)=MONTH
      ID(9)=YEAR
      ID(10)=N
      ID(11)=P
      ID(12)=NPROG
      CALL ABOUT1(NFILE,1)
      CALL ABOUT2(NFILE,TITLE,9)
      CALL ABOUT2(NFILE,TIME,NPLUSP)
      CALL ABOUT2 (NFILE,U,NPLUSP)
      CALL ABOUT2(NFILE,EST,NPLUSP)
      CALL ABOUT2(NFILE,CALC,NPLUSP)
      CALL ABOUT2(NFILE,ACTUAL,NPLUSP)
      CALL ABOUT2(NFILE,BUY,N)
      CALL ABOUT2(NFILE,MEAN,NPLUSP)
```

```
      CALL ABOUT2(NFILE,UN,N)
      CALL ABOUT2 (NFILL,PCONF(NPLUS1),P)
      CALL ABOUT2 (NFILE,MCONF,NPLUSP)
 1217 CALL ABOUT2 (NFILE, UW2, NPLUSP)
 1300 CONTINUE
      CALL GOODE
      END
```

ASYMPTOTIC (LOGISTIC) EXPONENTIAL - 59S

Unlike the first two models developed, the Maximum Likelihood Linear Model and the Maximum Likelihood Nonlinear Model, the Asymptotic (Logistic) Exponential Model is a weighted least squares model. The two earlier models, as their name indicates, are maximum likelihood models. It should be noted that the difference between a maximum likelihood model and a weighted least squares model are small. When talking about the mean line these two models are identical. The weighting factors in the weighted least squares model cause this to be true.

The observations are fitted to a model of the form

$$\overline{W}_i = \frac{a}{1 + be^{-ct_i}} \qquad c > 0$$

where a and b are parameters to be estimated and c is restricted to positive values. The weighted least squares criteria used to estimate these parameters require that the function $S(a, b, c, Y_i)$ be minimized, where

$$S = \sum_{i=1}^{n} \rho_i (W_i - \overline{W}_i)^2$$

The parameter ρ_i is a weighting factor which is a function of the E/C/A percentages at time t_i and a program maturity factor α,

$$\rho_i = \alpha^{n-i}(E_i + R_1 C_i + R_2 A_i)$$

Proportionality factors, R_1 and R_2, are designed to give those observations having a higher estimated percentage less influence in shaping the trend line. To achieve this, the values for R_1 and R_2 must be selected in accordance with

$$R_2 \geq R_1 \geq 1.$$

The method employed is similar to that used in the Maximum Likelihood Nonlinear Model. A range of c values is established, and for each c in the range, initial estimates of a and b are made. An iterative scheme is employed to obtain corrections

Δa_j and Δb_j for a and b, respectively. When $|\Delta a_j/a|$ and $|\Delta b_j/b|$ are both less than $\epsilon = .001$, the iteration process is terminated. The weighted sum of squares of errors, S, is computed, and the program repeats the above process for the next value of c in the range. When all c values have been exhausted, the minimum S is determined, and the corresponding values of a, b, c are the desired parameters.

Certain restrictions are placed on the corrections Δa_j and Δb_j. Maximum allowable values of the ratios $|\Delta a_j/a|$ and $|\Delta b_j/b|$ have been specified. If the computed correction exceeds the allowable maximum, the allowable maximum with the sign of the computed correction becomes the correction factor. If the convergence criterion $|\Delta a_j/a| < \epsilon$ and $|\Delta b_j/b| < \epsilon$ fails to be met within 100 iterations, the minimum S computed to date will determine the parameters a, b, c. As in the nonlinear model, c is restricted to values

$$c = k(.001) \qquad k = 1, 2, \ldots, 2000$$

Input to 59S consists of a case number, its associated Weight Data File, the program maturity factor α, $0 < \alpha \leq 1$, and the program option. This input is specified on a data card which is read by the Control Program 55S. For a detailed description of this data card see program 55S.

Four program options are available to the user. Each of these options incorporates the effects of the E/C/A percentages on the prediction line. The approach was to allow the effect of E/C/A's to be introduced in one fashion in the prediction range and to be introduced in an unrelated manner in the observation range. It turns out that the models for both ranges are functionally the same, the only difference being the parameter values in the different ranges. In the following, ρ_i^o refers to weighting factors in the observation range while ρ_i^p refers to weighting factors in the prediction range, where

$$\rho_i^o = \alpha^{n-i}(E_i + R_1 C_i + R_2 A_i)$$

$$\rho_i^p = (R_3 E_i + R_4 C_i + R_5 A_i)$$

E_i, C_i, A_i are E/C/A percentages at time t_i and R_i (i = 1, 2, 3, 4, 5) are proportionality factors designed to influence the shape of the trend line.

In the observation range, ρ_i^o is a function of α and the E/C/A percentages at time t_i. Thus, besides using the E/C/A percentages to weight the observations, the user may also weight them with the α factor. If no other weighting except the E/C/A percentages is desired, choose $\alpha = 1$. If it is desired to weight later observations more heavily than had been earlier observations, choose $0 < \alpha < 1$. In general, α should be chosen such that $.9 \le \alpha \le 1$. For $\alpha < .9$, the program maturity factor becomes dominant over the E/C/A weighting and tends to discard all but the last 9 or 10 observations.

In the prediction range, the incorporation of the influence of the E/C/A percentages is accomplished by adjusting the original model, $\overline{W}_i$, to obtain

$$\overline{W}_i^A = \overline{W}_{i-1}^A + \rho_i^p (W_i - W_{i-1})$$

where the superscript A denotes an adjusted trend. The E/C/A values employed in the prediction range are obtained by linearly interpolating the E/C/A percentages at the last observation to the assumed E/C/A percentages of 0/0/1 at ship date.

In the model, $R_1 = 2$, $R_2 = 4$, $R_3 = 1$, $R_4 = .5$, $R_5 = .1$. The following options over the observation and prediction ranges are available to the user:

$$\text{OPTION 1} \begin{cases} \rho_i^o = 1 \\ \rho_i^p = 1 \end{cases}$$

$$\text{OPTION 2} \begin{cases} \rho_i^o = 1 \\ \rho_i^p = f(E/C/A) \end{cases}$$

$$\text{OPTION 3} \begin{cases} \rho_i^o = f(E/C/A, \alpha) \\ \rho_i^p = 1 \end{cases}$$

$$\text{OPTION 4} \begin{cases} \rho_i^o = f(E/C/A, \alpha) \\ \rho_i^p = f(E/C/A) \end{cases}$$

Output consists of a file of data as is described on the following page.

I.D. RECORD:

FILEID FILENO DATE PROG LRS V CASENO MONTH YEAR N P NPROG
$TITLE_1$ . . . $TITLE_9$
$TIME_1$ . . . $TIME_{N+P}$
$WEIGHT_1$ . . . $WEIGHT_{N+P}$
EST_1 . . . EST_{N+P}
$CALC_1$ . . . $CALC_{N+P}$
$ACTUAL_1$ . . . $ACTUAL_{N+P}$
BUY_1 . . . BUY_N
$MEAN_1$ . . . $MEAN_{N+P}$
$PCONF_1$ . . . $PCONF_{N+P}$
$MCONF_1$ . . . $MCONF_{N+P}$
$UW2_1$ . . . $UW2_N$
$PRED_1$ . . . $PRED_P$

FILEID = label identifier

FILENO = file number

DATE = date file was created (MMDDYY)

PROG = BCD name of program

LRS = size of logical records within the file

V = control word for SPACE

CASENO = case number (integer)

MONTH = month in which first data point was observed (integer)

YEAR = year of first observation (integer)

N = number of observations (integer)

P = number of predictions (integer)

NPROG = code word

TITLE = case title

TIME = time points

WEIGHT = normalized weights

EST = estimated percentages

CALC = calculated percentages

ACTUAL = actual percentages

BUY = nonrandom changes #1 (buyoffs)

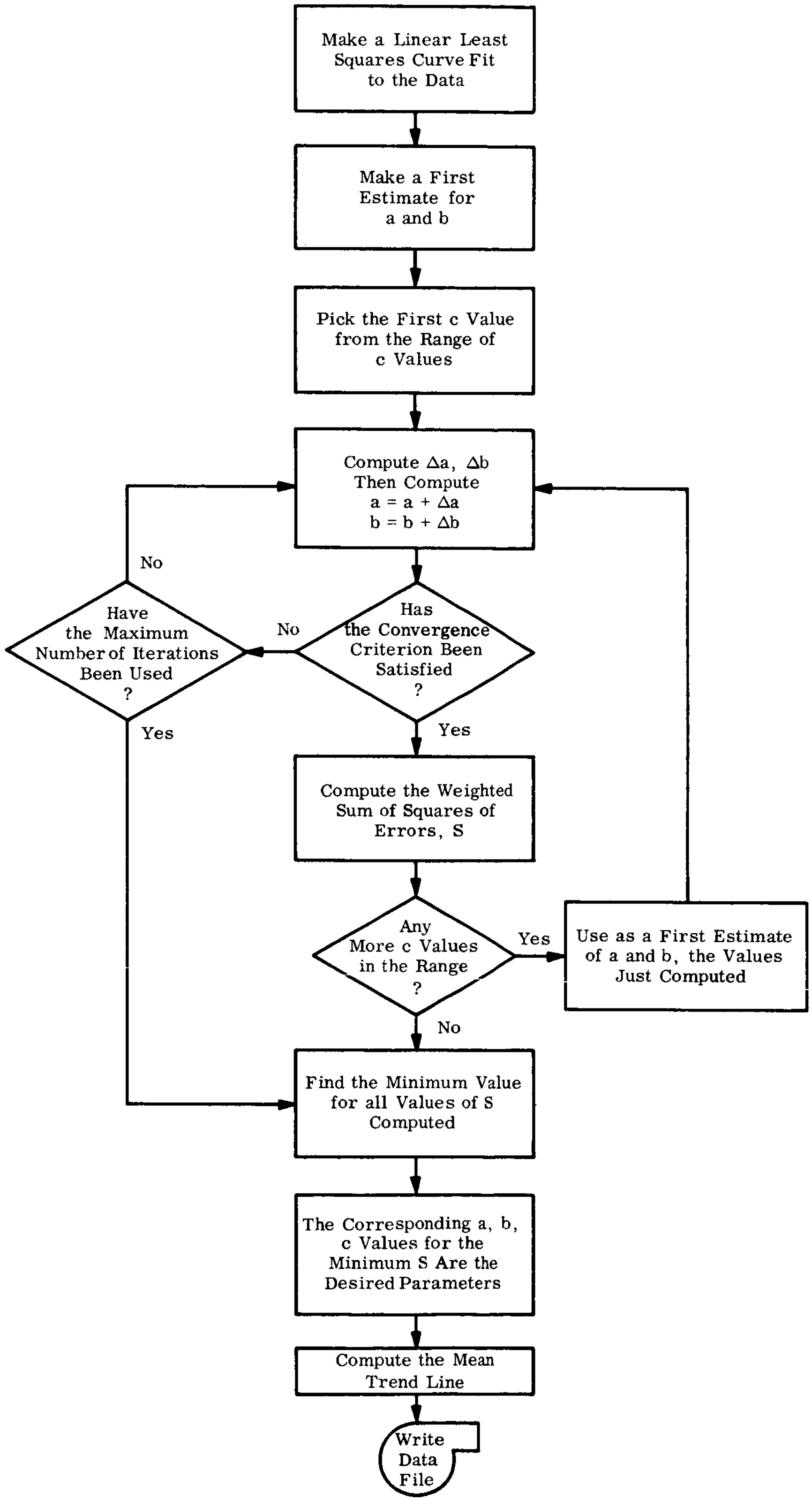
MEAN = mean trend line

PCONF = +95% confidence limits

MCONF = nonrandom changes #2 (outliers)

UW2 = observed weights

PRED = prediction line


Make a Linear Least Squares Curve Fit to the Data
Make a First Estimate for a and b
Pick the First c Value from the Range of c Values
Compute Δa, Δb Then Compute a = a + Δa b = b + Δb
No
Have the Maximum Number of Iterations Been Used ?
No
Has the Convergence Criterion Been Satisfied ?
Yes
Yes
Compute the Weighted Sum of Squares of Errors, S
Any More c Values in the Range ?
Yes
Use as a First Estimate of a and b, the Values Just Computed
No
Find the Minimum Value for all Values of S Computed
The Corresponding a, b, c Values for the Minimum S Are the Desired Parameters
Compute the Mean Trend Line
Write Data File

```
$IBFTC LOGMOD  LIST,REF
      INTEGER REELS
      INTEGER  CASENO,P,PMONTH,PYEAR,YEAR
      DOUBLE PRECISION M22,M22I
      COMMON /ACCESS/ HC(100),WDCT,ID(12),PROG
      COMMON /SYSTEM/ NTAPES,REELS(15),CNTRLS(15),FILES(15),LRS(15),
     1                POS(15),TRLPOS(15),RWCNT(15),UNITS(15)
      COMMON  /PFILE/ NFYLE(4,300),NFY(4)
      COMMON /PPOG/ ACTUAL(100) , CALC  (100) , EST   (100) , COM  (12),
     1              LSQR  (100) , MEAN  (100) , MCONF (100) , OW2 (100),
     2              MSQR  (100) , PCONF (100) , S1    (100) , S2  (100),
     3              TIME  (100) , WEIGHT(100) , BOY   (100) , TITLE(9) ,
     4              N , NTOT
      COMMON /LOGIS/ KEEP(4),ALP
      DIMENSION  CP(21),AP(21),BP(21),RHO(100),M22(2,2),M22I(2,4),G(2)
      DIMENSION YBARA(100),S(20),TOM(6),W(100),YBAR(100),ECT(100),
     1DEN(100)
      EQUIVALENCE  (COM(1),PMONTH), (COM(2),PYEAR), (COM(3),MONTH),
     2             (COM(4),YEAR)   , (COM(5),R1)    , (COM(6),R2)    ,
     3             (COM(7),ASUBP) , (COM(8),BSUBP), (COM(9),CSUBP),
     4              (COM(10),NSUBP)
      DATA TOM/1.,2.,4.,1.,.5,.1/
      DATA  NPROG /3HLOG/
      CALL TRAPOK
      DO 1300 I1=1,NTOT
      CALL GETDAT(IPSWT,I1,CASENO)
      NPLUS1 = N + 1
      NPLUSP = 12*( PYEAR - YEAR ) + PMONTH - MONTH + 1
      K85=TIME(N)
      P=NPLUSP-K85
      NPLUSP=N+P
      DO  4051   I3=1,N
 4051 W(I3)=WEIGHT(I3)
      DO  408   J8=NPLUS1,NPLUSP
      EST(J8)= -EST(N)*FLOAT(J8-N)/FLOAT(P)  + EST(N)
      CALC(J8)=-CALC(N)*FLOAT(J8-N)/FLOAT(P) + CALC(N)
  408 ACTUAL(J8)= (1.-ACTUAL(N))*FLOAT(J8-N)/FLOAT(P) + ACTUAL(N)
      DO 23 I=NPLUS1,NPLUSP
   23 TIME(I)=TIME(I-1)+1.
      DO  10000   J9=1,4
      IF( KEEP(J9)  .EQ. 0  )       GO TO  10000
      NPATH=KEEP(J9)
      GO TO (410,411,412,413),NPATH
  410 DO  4100  I=1,NPLUSP
 4100 RHO(I)=1.
      GO TO 414
  411 DO 4110  I=1,N
 4110 RHO(I)=1.
      DO 4111  I=NPLUS1,NPLUSP
      RHO(I)=  (TOM(4)*EST(I)+TOM(5)*CALC(I)+TOM(6)*ACTUAL(I))/
     1         (1.+ACTUAL(I))
 4111 CONTINUE
```

```
      GO TO 414
412   DO  4120    I=1,N
      RHO(I) =  (ALP**(N-I))*(TOM(1)*EST(I)+TOM(2)*CALC(I)+TOM(3)*ACTUAL
     1          (I))
4120  CONTINUE
      DO 4121 I=NPLUS1,NPLUSP
4121  RHO(I)=1.
      GO TO 414
413   DO 4130   I=1,N
      RHO(I) =  (ALP**(N-I))*(TOM(1)*EST(I)+TOM(2)*CALC(I)+TOM(3)*ACTUAL
     1          (I))
4130  CONTINUE
      DO 4131   I=NPLUS1,NPLUSP
      RHO(I)=   (TOM(4)*EST(I)+TOM(5)*CALC(I)+TOM(6)*ACTUAL(I))/
     1          (1.+ACTUAL(I))
4131  CONTINUE
414   CONTINUE
      KSWIT=0
      NSWIT=0
      A1=0.
      B1=0.
      C1=0.
      D1=0.
      F1=0.
      DO 6   I=1,N
      A1=A1+W(I) * RHO(I)
      B1=B1+TIME(I)*W(I)   * RHO(I)
      C1=C1+TIME(I)   * RHO(I)
      F1=F1+RHO(I)
 6    D1=D1+TIME(I)*TIME(I)   * RHO(I)
      E1=F1*D1-C1**2
      ALPHA=  (A1*D1 - C1*B1)/E1
      BETA=(F1*B1-A1*C1)/E1
      EN=N
      S1(1) = EN / 4.0
      A1 = ALPHA + S1(1) * BETA
      S2(1) = (3.0*EN) / 4.0
      B1 = ALPHA + S2(1) * BETA
      S1(1). = 0.1 * S1(1)
      S2(1) = 0.1 * S2(1)
      AP(1)= A1 + BP(1)*EXP(-S1(1))*A1
      BP(1)= (A1-B1) / ( EXP(-S2(1))*B1 - EXP(-S1(1))*A1)
      CP(1)=.1
      DELC=CP(1)
      ATEST=0.
      KOUNT=20
50    CONTINUE
      DO 15 I=1,20
      DO  13 K=1,100
      DO 9 I7=1,2
      DO  8  J=1,2
 8    M22(I7,J)=0.
 9    G(I7)=0.
```

```
      DO  10 J=1,N
      ECT(J)=EXP(-CP(I)*TIME(J))
      DEN(J)= 1.+BP(I)*ECT(J)
      YBAR(J)=  AP(I)/DEN(J)
      M22(1,1)=M22(1,1) +((YBAR(J)**2)*RHO(J))/AP(I)**2
      M22(1,2)=M22(1,2) - ((YBAR(J)**2)*RHO(J)*ECT(J))/(AP(I)*DEN(J))
      M22(2,2)=M22(2,2) +  (RHO(J)*ECT(J)*ECT(J)*(YBAR(J)**2))/DEN(J)**2
      G(1)=G(1)+ (RHO(J)*YBAR(J)*(W(J)-YBAR(J)))/ AP(I)
      G(2)=G(2) -(RHO(J)*(W(J)-YBAR(J))*AP(I)*ECT(J))/DEN(J)**2
   10 CONTINUE
      M22(2,1)=M22(1,2)
      CALL DPMI(2,M22,M22I)
      CALL SLITET(3,JFK)
      GO TO(5000,11),JFK
   11 CONTINUE
      DELA = M22I(1,1)*G(1)+M22I(1,2)*G(2)
      DELB = M22I(2,1)*G(1)+M22I(2,2)*G(2)
      IF( K .EQ. 1 ) XONE=DELB
      Q1= ABS( DELA/AP(I))
      Q2= ABS( DELB/BP(I))
      NFLAG=0
      IF( Q1 .LT. .001 .AND.  Q2  .LT. .001 )      GO TO  12
      NFLAG=1
      IF( Q1 .LE. .5     )      GO   TO  111
      DELA= SIGN(.5*AP(I),DELA)
  111 IF( Q2 .LE. 1.2 ) GO TO 1222
      DELB=SIGN(1.2*BP(I),DELB)
      GO TO 12
 1222 CONTINUE
      IF( K .EQ. 1 ) GO TO 12
      SELECT= XONE/DELB
      XONE=DELB
      IF( SELECT .GT. 0. )  GO TO 12
      SELECT=ABS(SELECT)
      IF( SELECT .GT. .75 .AND. SELECT .LT. 1.25  ) DELB=DELB/2.
      XONE=DELB
   12 AP(I)=AP(I)+DELA
      BP(I)=BP(I)+DELB
      IF( NFLAG  .EQ. 0  )          GO  TO  14
   13 CONTINUE
      IF( ATEST .EQ. 1. )  GO TO 5000
      KOUNT=I-1
      IF( KOUNT .NE. 0 )  GO TO 21
      WRITE(6,1491)
 1491 FORMAT(1H1,      42HLOGISTIC MODEL DOES NOT FIT THE DATA. EXIT)
      GO TO 10000
   14 AP(I+1)=AP(I)
      BP(I+1)=BP(I)
      CP(I+1)=CP(I)+DELC
      S(I)=0.
      DO  149  J1=1,N
  149 S(I)=S(I) + RHO(J1)*(W(J1)-( AP(I)/(1.+BP(I)*EXP(-CP(I)*TIME(J1)))
     1   ))**2
```

```
   15 CONTINUE
   21 CONTINUE
      SMIN=10.E+10
      DO 16 I=1,KOUNT
      IF( S(I) .GT. SMIN )  GO TO 16
      K=I
      SMIN=S(I)
   16 CONTINUE
      A2=AP(K)
      BB=BP(K)
      CC=CP(K)
      ATEST=1.
      IF( KSWIT  .EQ.  1)     GO    TO    20
      K=K+1
      AP(1)=AP(K)
      BP(1)=BP(K)
      CP(1)=CP(K)
      DELC=-.01
      KSWIT=1
      KOUNT=20
      GO TO 50
   20 CONTINUE
      IF( NSWIT .EQ. 1 )  GO TO 5000
      IF( CC .LE. .011)  GO TO 4999
      K=K+1
      DELC=.001
 4998 CONTINUE
      AP(1)=AP(K)
      BP(1)=BP(K)
      CP(1)=CP(K)
      NSWIT=1
      KOUNT=20
      GO TO 50
 4999 CONTINUE
      K=K-1
      DELC=-.001
      GO TO 4998
 5000 CONTINUE
      DO 22 I=1,NPLUSP
   22 YBAR(I) =    A2/(1.+BB*EXP(-CC*TIME(I)))
      YBARA(N)=YBAR(N)
      DO   415   J5=NPLUS1,NPLUSP
  415 YBARA(J5)=YBARA(J5-1)+RHO(J5)*( YBAR(J5)-YBAR(J5-1) )
      NFY(4) = NFY(4) + 1
      NF4 = NFY(4)
      NF44=ISIGN(NF4,REELS(4))
      NFILE=REELS(4)*1000 + NF44
      NFYLE(4,NF4) = CASENO
      DIFRC = UW2(N) - YBAR(N)
      DO 1216 IW=NPLUS1,NPLUSP
      W(IW) = YBARA(IW)
 1216 UW2(IW) = W(IW) + DIFRC
      ID(7)=CASENO
```

```
      ID(8)= MONTH
      ID(9)= YEAR
      ID(10)=N
      ID(11)=P
      ID(12)=NPROG
      CALL ABOUT1(NFILE,1)
      CALL ABOUT2(NFILE,TITLE,9)
      CALL ABOUT2(NFILE,TIME,NPLUSP)
      CALL ABOUT2 (NFILE,W,NPLUSP)
      CALL ABOUT2(NFILE,EST,NPLUSP)
      CALL ABOUT2(NFILE,CALC,NPLUSP)
      CALL ABOUT2(NFILE,ACTUAL,NPLUSP)
      CALL ABOUT2(NFILE,BUY,N)
      CALL ABOUT2 (NFILE,YBAR(1),N)
      CALL ABOUT2 (NFILE,YBARA(NPLUS1),P)
      CALL ABOUT2(NFILE,PCONF,NPLUSP)
      CALL ABOUT2(NFILE,MCONF,NPLUSP)
 1217 CALL ABOUT2 (NFILE, UW2, NPLUSP)
10000 CONTINUE
 1300 CONTINUE
      CALL GOODE
      END
```

OUTPUT PROGRAM - 60S (OUTPT)

This routine reads the binary output files that are generated by each of the four trending methods and produces printed tabular and printed plot output in a uniform format for each method.

No options are available; all output that is generated during trending is processed by this routine. An example appears on the following pages.

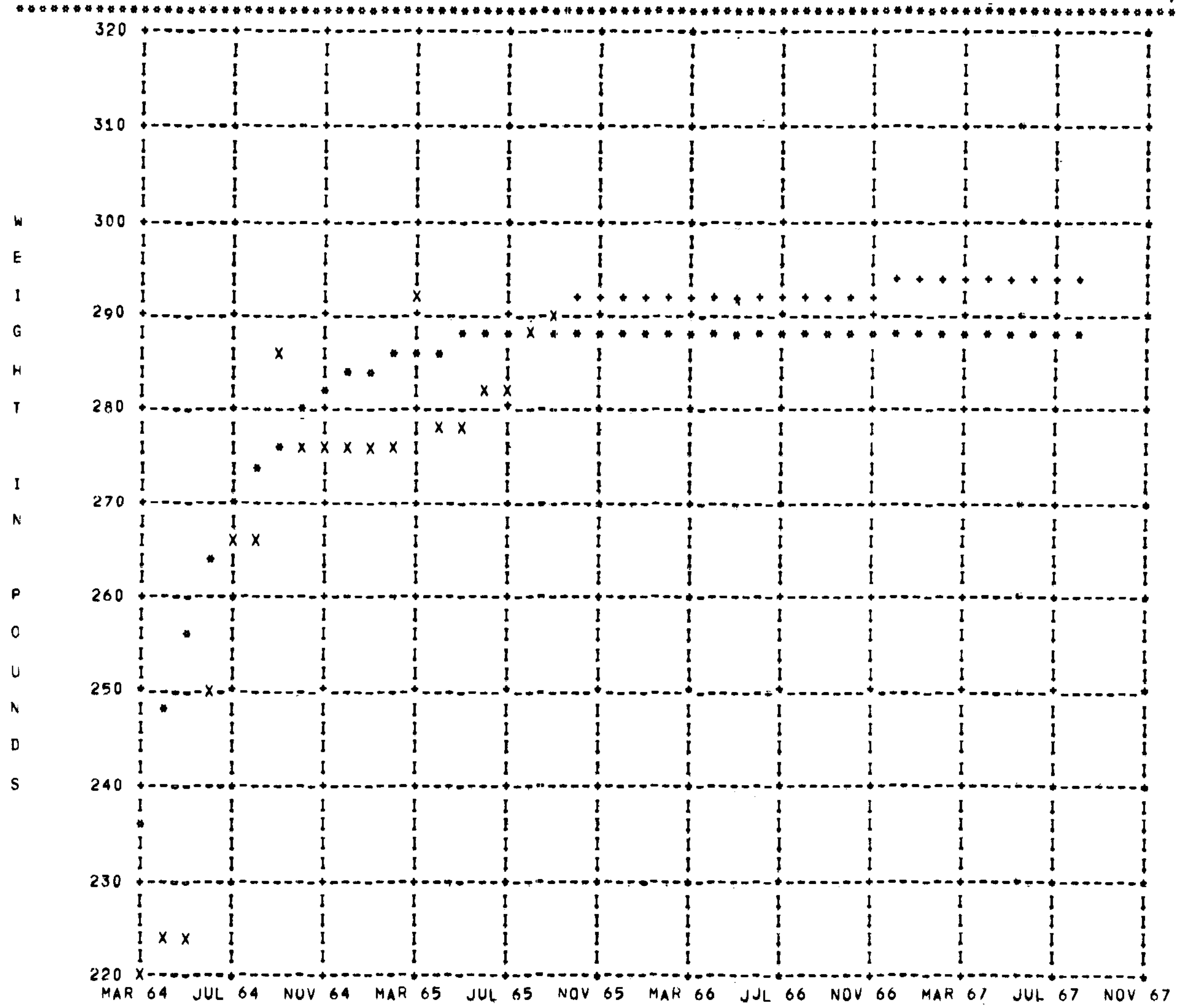
Control over this routine is maintained by the Control Program 55S.

PREDICTION ANALYSIS TECHNIQUE * BLK II SPACECRAFT MISSION 504 + SUBS * CASE NUMBER * DATE 11/21/65 * REQUIRED BUYOFF
LEM ASCENT — L3
NON-LINEAR MODEL * AT BURNOUT * 0611906 * INPUT TAPE 10/08/65 * EQUIVALENT S/C OR L/V
COMPUTER PROGRAM 57S * NAVIGATION AND GUIDANCE * FILE STATUS NO. 20 * DEFICIT L3

MONTH	.... OBSERVED DATA .... WEIGHT (LB)	PERCENT EST	PERCENT CAL	PERCENT ACT	OBSERVED NONRANDOM CHANGES (LB)	OTHER NONRANDOM CHANGES (LB)	... NORMALIZED DATA ... WEIGHT (LB)	PERCENT EST	PERCENT CAL	PERCENT ACT	MEAN TREND VALUE (LB)	UPPER 95 PCT. CONF. LIM. (LB)	AVERAGE WEIGHT GROWTH (LB/MO)
MAR 64	220	100	0	0	0	0	242				237		2
APR 64	224	100	0	0	0	0	246				248		2
MAY 64	225	100	0	0	0	0	247				257		2
JUN 64	251	100	0	0	-15	0	258				264		1
JUL 64	266	100	0	0	0	0	273				269		1
AUG 64	266	100	0	0	0	0	273				274		1
SEP 64	287	100	0	0	0	0	294				277		0
OCT 64	277	100	0	0	0	0	284				279		0
NOV 64	277	100	0	0	0	0	284				281		0
DEC 64	277	100	0	0	0	0	284				283		0
JAN 65	277	100	0	0	0	0	284				284		0
FEB 65	277	100	0	0	0	0	284				285		0
MAR 65	293	100	0	0	-22	0	278				286		-0
APR 65	278	100	0	0	15	0	278				287		0
MAY 65	278	100	0	0	0	0	278				287		1
JUN 65	282	100	0	0	0	0	282				287		0
JUL 65	282	100	0	0	0	0	282				288		0
AUG 65	288	33	0	67	0	0	288				288		0
SEP 65	291	32	0	68	0	0	291				288		0
.... PREDICTED VALUES													
OCT 65	291	31	0	69			291				288	292	0
NOV 65	291	29	0	71			291				288	292	0
DEC 65	291	28	0	72			291				289	292	0
JAN 66	291	26	0	74			291				289	293	0
FEB 66	291	25	0	75			291				289	293	0
MAR 66	291	24	0	76			291				289	293	0
APR 66	292	22	0	78			292				289	293	0
MAY 66	292	21	0	79			292				289	293	0
JUN 66	292	19	0	81			292				289	293	0
JUL 66	292	18	0	82			292				289	293	0
AUG 66	292	17	0	83			292				289	293	0
SEP 66	292	15	0	85			292				289	293	0
OCT 66	292	14	0	86			292				289	293	0
NOV 66	292	13	0	87			292				289	293	0
DEC 66	292	11	0	89			292				289	293	0
JAN 67	292	10	0	90			292				289	293	0
FEB 67	292	8	0	92			292				289	293	0
MAR 67	292	7	0	93			292				289	293	0
APR 67	292	6	0	94			292				289	293	0
MAY 67	292	4	0	96			292				289	293	0
JUN 67	292	3	0	97			292				289	293	0
JUL 67	292	1	0	99			292				289	293	0
AUG 67	292	0	0	100			292				289	293	0

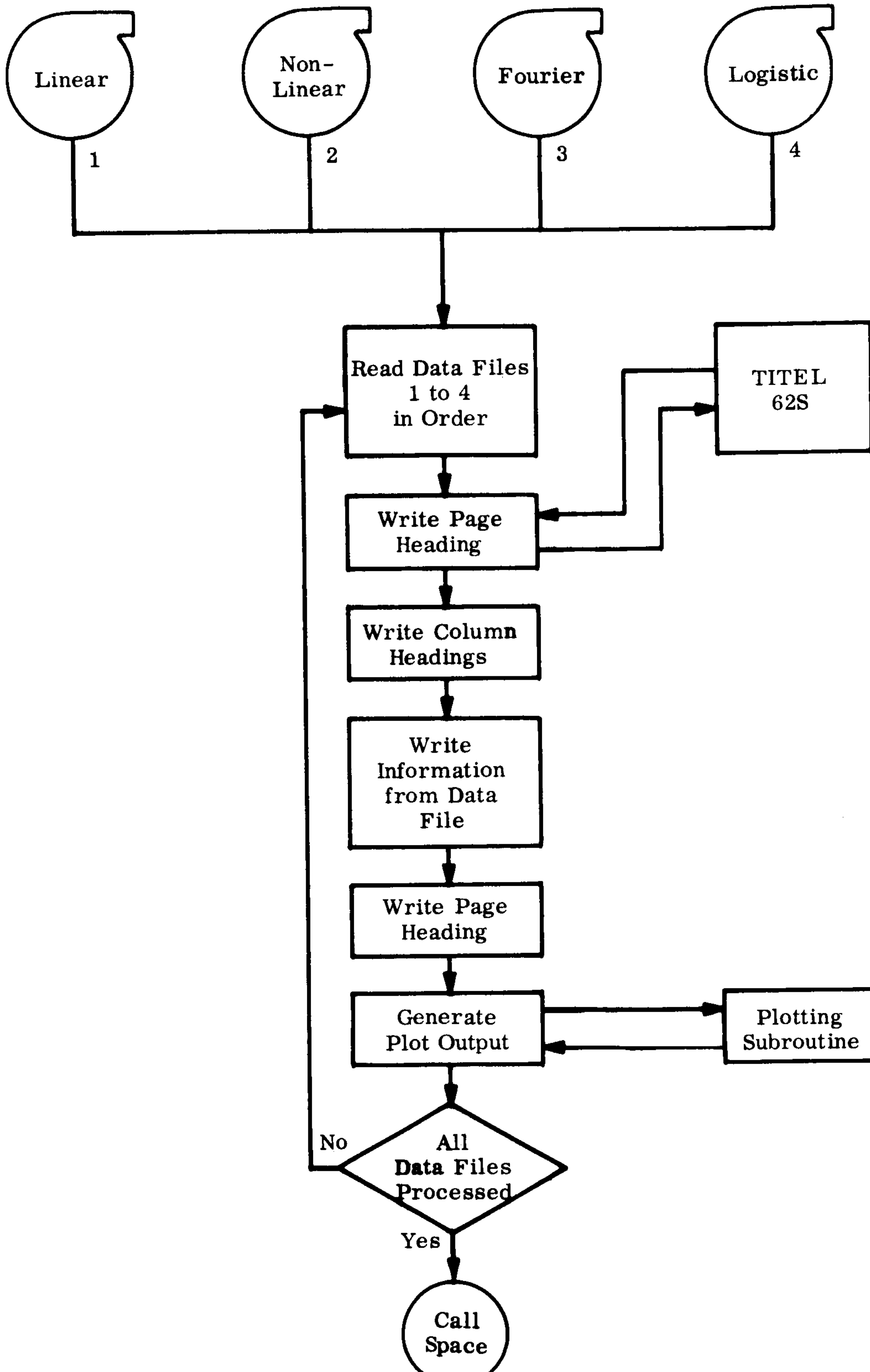
Reference No. 60.2
Issue Date 23 Dec 1965
Supersedes 13 August 1965

```
* PREDICTION ANALYSIS TECHNIQUE * BLK II SPACECRAFT MISSION 504 + SUBS * CASE NUMBER * DATE        11/21/65 * REQUIRED BUYOFF
* ***************************** * LEM ASCENT                             *              *                       *               LB
* NON-LINEAR MODEL              * AT BURNOUT                             *   0611906    * INPUT TAPE 10/08/65   * EQUIVALENT S/C OR L/V
* COMPUTER PROGRAM    57S       * NAVIGATION AND GUIDANCE                *              * FILE STATUS NO,  20   * DEFICIT       LB
```



X = OBSERVED WEIGHT   * = MEAN ESTIMATED WEIGHT   + = CONFIDENCE LIMITS

DATA FILES CREATED BY TRENDING METHODS



```
$IBFTC OUTPUT   LIST,REF
C *
C *** GENERAL OUTPUT ROUTINE FOR THE TREND SYSTEM
C *
       INTEGER   REELS,CASENO,P,PMONTH,PYEAR,YEAR
       REAL      LSQR,MEAN,MCONF,MSQR
C
       COMMON   /GGDATE/DATE(2)
       COMMON    /ACCESS/  HC(100),WDCT,ID(12),PROG
       COMMON    /SYSTEM/  NTAPES,REELS(15),CNTRLS(15),FILES(15),ERS(15),
      1                    POS(15),TRLPOS(15),RWCNT(15),UNITS(15)
       COMMON   /PFILE/ NFYLE(4,300),NFY(4)
       COMMON    /PROG/    ACTUAL(100),CALC(100),EST(100),COM(12),
      1                    LSQR(100),MEAN(100),MCONF(100),QW2(100),
      2                    MSQR(100),PCONF(100),S1(100),S2(100),
      3                    TIME(100),WEIGHT(100),BUY(100),TITLE(9),
      4                    N,NTOT
       COMMON   /STT/     N21,D1,D2,J1RUN
       COMMON   /ZEE/ ZLINE1(6),ZLINE2(6),ZLINE3(6),ZLINE4(6)
C
       DIMENSION   XMUNTH(12),PNME(4)
       DIMENSION   TNME(4,4),BLANK(7)
       DIMENSION   PTAB(4),MASK(7)
       DIMENSION   GRID(867),NSCALE(5),BLKM(12),IBLKY(12),TAB(24)
C
       EQUIVALENCE   (COM(1),PMONTH),(COM(2),PYEAR),(COM(3),MONTH),
      2              (COM(4),YEAR),(COM(5),K1),(COM(6),K2),(COM(7),ASUBP),
      3              (COM(8),BSUBP),(COM(9),CSUBP),(COM(10),NSUBP)
C
       DATA   XMUNTH   /3HJAN,3HFEB,3HMAR,3HAPR,3HMAY,3HJUN,3HJUL,3HAUG,
      /                  3HSEP,3HOCT,3HNOV,3HDEC/
       DATA   TNME    /6HLINEAR,6H MODEL,6H       ,6H       ,
      1                6HNON-LI,6HNEAR M,6HODEL   ,6H       ,
      2                6HFOURIE,6HR MODL,6HL      ,6H       ,
      3                6HLOGIST,6HIC MOD,6HEL     ,6H
       DATA   BLANK   /42H
       DATA   PTAB   /3H56S,3H57S,3H58S,3H59S/
       DATA   NSCALE   /1,0,-1,0,-1/
       DATA   TAB     /1.0,2.0,5.0,10.0,20.0,25.0,50.0,100.0,200.0,
      1                250.0,500.0,1000.0,2000.0,2500.0,5000.0,10000.0,
      2                20000.0,25000.0,50000.0,100000.0,200000.0,
      3                250000.0,500000.0,1000000.0/
       DATA   ARG   /6HSAVTAP/
       DATA   MASK   /1,1,1,1,1,1,1/
C
C   **
C
        IZO = 0
        DO 900 ICK=1,4
        IF (REELS(ICK) .EQ. 0) GO TO 900
        IF (NFY(ICK) .EQ. 0)   GO TO 900
        PNO = PTAB(ICK)
```

```
      LOOP = NFY(ICK)
      DO 120 MM=1,4
  120 PNME(MM) = TNME(MM,ICK)
C
C
      DO 600 MA=1,LOOP
      LSKIP = 0
      LINE = 0
      NZ = MA
      NZ = ISIGN (NZ,REELS(ICK))
      NZ = REELS(ICK) * 1000 + NZ
C
      CALL READB1 (NZ,1)
      NPLUSP = ID(10) + ID(11)
      NPLUS1 = ID(10)  + 1
      CALL READB2 (NZ,TITLE,9,HISTRY)
      CALL READB2 (NZ,TIME,NPLUSP,HISTRY)
      CALL READB2 (NZ,WEIGHT,NPLUSP,HISTRY)
      CALL READB2 (NZ,EST,NPLUSP,HISTRY)
      CALL READB2 (NZ,CALC,NPLUSP,HISTRY)
      CALL READB2 (NZ,ACTUAL,NPLUSP,HISTRY)
      CALL READB2 (NZ,BUY,ID(10),HISTRY)
      CALL READB2 (NZ,MEAN,NPLUSP,HISTRY)
      CALL READB2 (NZ,PCONF,NPLUSP,HISTRY)
      CALL READB2 (NZ,MCONF,NPLUSP,HISTRY)
      CALL READB2 (NZ,UW2,NPLUSP,HISTRY)
      CALL ENDF (NZ)
C
      KRA = ID(8)
      IYER = ID(9) - 1900
      CALL TITEL (ID(7),MASK)
C
      WRITE (6,9000)
      LKASE = ID(7)
      LKASE = LKASE / 1000000
      IF (LKASE .NE. 0)  GO TO 190
      WRITE (6,9004) ZLINE1,DATE,ZLINE2,PNME,ZLINE3,ID(7),
     1               D1,D2,PNO,ZLINE4,N21
      GO TO 192
  190 WRITE (6,9002) ZLINE1,DATE,ZLINE2,PNME,ZLINE3,ID(7),
     1               D1,D2,PNO,ZLINE4,N21
  192 WRITE (6,9001)
      WRITE (6,9003)
      WRITE (6,9001)
C
C
  220 DO 350 KP =1,NPLUSP
C
      IF (KP .EQ. 1)  GO TO 221
      IF (TIME(KP) - TIME(KP-1) .EQ. 1.0)  GO TO 221
      KTY = TIME(KP) - TIME(KP-1) - 0.5
      DO 1222 ITY=1,KTY
      LINE = LINE + 1
```

```
      IF (KRA .LE. 12)  GO TO 1221
      KRA = 1
      IYER = IYER + 1
 1221 XMONZ = XMUNTH(KRA)
      WRITE (6,9055)  XMONZ,IYER
      KRA = KRA + 1
 1222 CONTINUE
  221 IUW2 = UW2(KP) + 0.5
      IEST = EST(KP) * 100.0 + 0.5
      ICAL = CALC(KP) * 100.0 + 0.5
      IACT = ACTUAL(KP) * 100.0 + 0.5
      IWGT = WEIGHT(KP) + 0.5
      IMENE = MEAN(KP) + 0.5
      IPCONF = PCONF(KP) + 0.5
      IMCONF = MCONF(KP) + 0.5
      IF (BUY(KP))  224,222,222
  222 IBUY = BUY(KP) + 0.5
      GO TO 225
  224 IBUY = BUY(KP) - 0.5
  225 TAG = ((UW2(NPLUSP) - UW2(KP)) / (FLOAT(NPLUSP-KP)))
      IF (TAG)  227,226,226
  226 IAG = TAG + 0.5
      GO TO 228
  227 IAG = TAG - 0.5
  228 CONTINUE
C
      IF (KRA .LE. 12)   GO TO 230
      KRA = 1
      IYER = IYER + 1
  230 XMONZ = XMUNTH(KRA)
C
  240 IF (LSKIP .EQ. 1 .AND. KP .NE. NPLUSP) GO TO 350
      GO TO (246,250,258,254),ICK
C
  246 IF (KP .GE. NPLUS1)  GO TO 248
      WRITE (6,9010) XMONZ,IYER,IUW2,IEST,ICAL,IACT,
     1                IBUY,IMCONF,IWGT,IMENE,IAG
      GO TO 275
  248 WRITE (6,9011) XMONZ,IYER,IUW2,IEST,ICAL,IACT,
     1                IUW2,IMENE,IPCONF,IAG
      GO TO 275
C
  250 IF (KP .GE. NPLUS1)  GO TO 252
      WRITE (6,9020) XMONZ,IYER,IUW2,IEST,ICAL,IACT,
     1                IBUY,IMCONF,IWGT,IMENE,IAG
      GO TO 275
  252 WRITE (6,9021) XMONZ,IYER,IUW2,IEST,ICAL,IACT,
     1                IUW2,IMENE,IPCONF,IAG
      GO TO 275
C
  254 IF (KP .GE. NPLUS1)  GO TO 256
      WRITE (6,9030) XMONZ,IYER,IUW2,IEST,ICAL,IACT,
     1                IBUY,IMCONF,IWGT,IMENE,IAG
```

```
      GO TO 275
  256 WRITE (6,9031) XMONZ,IYER,IUW2,IEST,ICAL,IACT,
     1               IUW2,IMENE,IAG
      GO TO 275
C
  258 IF (KP .GE. NPLUS1) GO TO 260
      WRITE (6,9040) XMONZ,IYER,IUW2,IEST,ICAL,IACT,
     1               IBUY,IMCONF,IWGT,IMENE,IAG
      GO TO 275
  260 WRITE (6,9041) XMONZ,IYER,IUW2,IEST,ICAL,IACT,
     1               IUW2,IMENE,IPCONF,IAG
C
  275 LINE = LINE + 1
      IF (KP .NE. ID(10))  GO TO 276
      WRITE (6,9005)
  276 IF (LINE .LT. 40)  GO TO 350
      IF ((NPLUSP - KP) .LE. 6)  GO TO 350
      WRITE (6,9006)
      LSKIP = 1
  350 KRA = KRA + 1
C
C
      YMIN = 10.0E+10
      YMAX = 0.0
      DO 308 IPP=1,NPLUSP
      IF (MEAN(IPP) .LT. YMIN)  YMIN = MEAN(IPP)
      IF (MEAN(IPP) .GT. YMAX)  YMAX = MEAN(IPP)
      IF (IPP .GT. N) GO TO 305
      IF (UW2(IPP) .GT. YMAX)  YMAX = UW2(IPP)
      IF (UW2(IPP) .LT. YMIN)  YMIN = UW2(IPP)
  305 IF (ICK .EQ. 4)  GO TO 308
      IF (IPP .LT. NPLUS1)  GO TO 308
      IF (PCONF(IPP) .GT. YMAX)  YMAX = PCONF(IPP)
  308 CONTINUE
      DMIN = YMIN
      DMAX = YMAX
      DEL = (DMAX - DMIN) / 10.0
      DO 310 JACK=1,24
      IF (DEL .LE. TAB(JACK))  GO TO 311
  310 CONTINUE
  311 INK = TAB(JACK)
      MIN = DMIN
      YMIN = (MIN / INK) * INK
      YMAX = YMIN + 10.0 * TAB(JACK)
      IYERE = ID(9) - 1900
      MYY = ID(8)
      DO 315 MY=1,12
      BLKM(MY) = XMUNTH(MYY)
      IBLKY(MY) = IYERE
      DO 314 KY=1,4
      MYY = MYY + 1
      IF (MYY .LE. 12)  GO TO 314
      MYY = 1
```

```
      IYERE = IYERE + 1
  314 CONTINUE
  315 CONTINUE
      WRITE (6,9000)
      IF (LKASE .NE. 0)  GO TO 400
      WRITE (6,9004) ZLINE1,DATE,ZLINE2,PNME,ZLINE3,ID(7),
     1               D1,D2,PNO,ZLINE4,N21
      GO TO 402
  400 WRITE (6,9002) ZLINE1,DATE,ZLINE2,PNME,ZLINE3,ID(7),
     1               D1,D2,PNO,ZLINE4,N21
  402 WRITE (6,9001)
      CALL PLOT1 (NSCALE,10,5,11,8)
      CALL PLOT2 (GRID,45.0,1.0,YMAX,YMIN)
      CALL PLOT3 (1H*,TIME(1),MEAN(1),NPLUSP)
      IF (ICK .EQ. 4)  GO TO 425
      CALL PLOT3 (1H+,TIME(NPLUS1),PCONF(NPLUS1),ID(11))
  425 CALL PLOT3 (1HX,TIME(1),UW2(1),ID(10))
      CALL OMIT(1)
      CALL FPLOT4 (41,41H          W E I G H T   I N   P O U N D S )
      WRITE (6,9050) (BLKM(KK),IDLKY(KK),KK=1,12)
      WRITE (6,9051)
  600 CONTINUE
C
  900 CONTINUE
C
      CALL SPACE
C
 9003 FORMAT (2H *,6X,41H*  .... OBSERVED DATA ....  * OBSERVED * ,
     1          50H   OTHER  *  ... NORMALIZED DATA ...  *   MEAN  * ,
     2          33H  UPPER  * AVERAGE *              * / 9H *      *,28X,
     3          23H* NONRANDOM* NONRANDOM*,28X,21H*  TREND  * 95 PCT. *,
     4          23H  WEIGHT *              * / 21H *MONTH *  WEIGHT  * ,
     5          51H P E R C E N T  *  CHANGES *  CHANGES *  WEIGHT  * ,
     6          52H P E R C E N T  *  VALUE  *CONF.LIM.*  GROWTH *      ,
     7          8H         * / 40H *      *    (LB)   *  EST  CAL  ACT  *  ,
     8          53H (LB)    *    (LB)    *    (LB)    *  EST  CAL  ACT  *   (
     9          39HLB)  *   (LB)   * (LB/MO) *              * )
 9000 FORMAT (1H1,131(1H*))
 9001 FORMAT (1X,131(1H*))
 9002 FORMAT (1X,34H* PREDICTION ANALYSIS TECHNIQUE * ,6A6,   1X,1H*,
     1          20H CASE NUMBER * DATE ,6X,A6,A2,18H * REQUIRED BUYOFF,
     2          6X,1H*  / 3H * ,29(1H*),3H * ,6A6,2H *,13X,1H*,21X,1H*,1
     3          6X,7H LB    * / 3H * ,4A6,5X,3H * ,6A6,2H *,3X,I7,3X,
     4          13H* INPUT TAPE ,A6,A2,25H * EQUIVALENT S/C OR L/V*  /
     5          19H * COMPUTER PROGRAM,3X,A3,7X,3H * ,6A6,2H *,13X,
     6          17H* FILE STATUS NO.  ,I4,12H * DEFICIT  ,6X,7H LB    * )
 9004 FORMAT (1X,34H* PREDICTION ANALYSIS TECHNIQUE * ,6A6,   1X,1H*,
     1          20H CASE NUMBER * DATE ,6X,A6,A2,18H * REQUIRED BUYOFF,
     2          6X,1H*  / 3H * ,29(1H*),3H * ,6A6,2H *,13X,1H*,21X,1H*,1
     3          6X,7H LB    * / 3H * ,4A6,5X,3H * ,6A6,2H *,3X,1H0,I6,3X,
     4          13H* INPUT TAPE ,A6,A2,25H * EQUIVALENT S/C OR L/V*  /
     5          19H * COMPUTER PROGRAM,3X,A3,7X,3H * ,6A6,2H *,13X,
     6          17H* FILE STATUS NO.  ,I4,12H * DEFICIT  ,6X,7H LB    * )
```

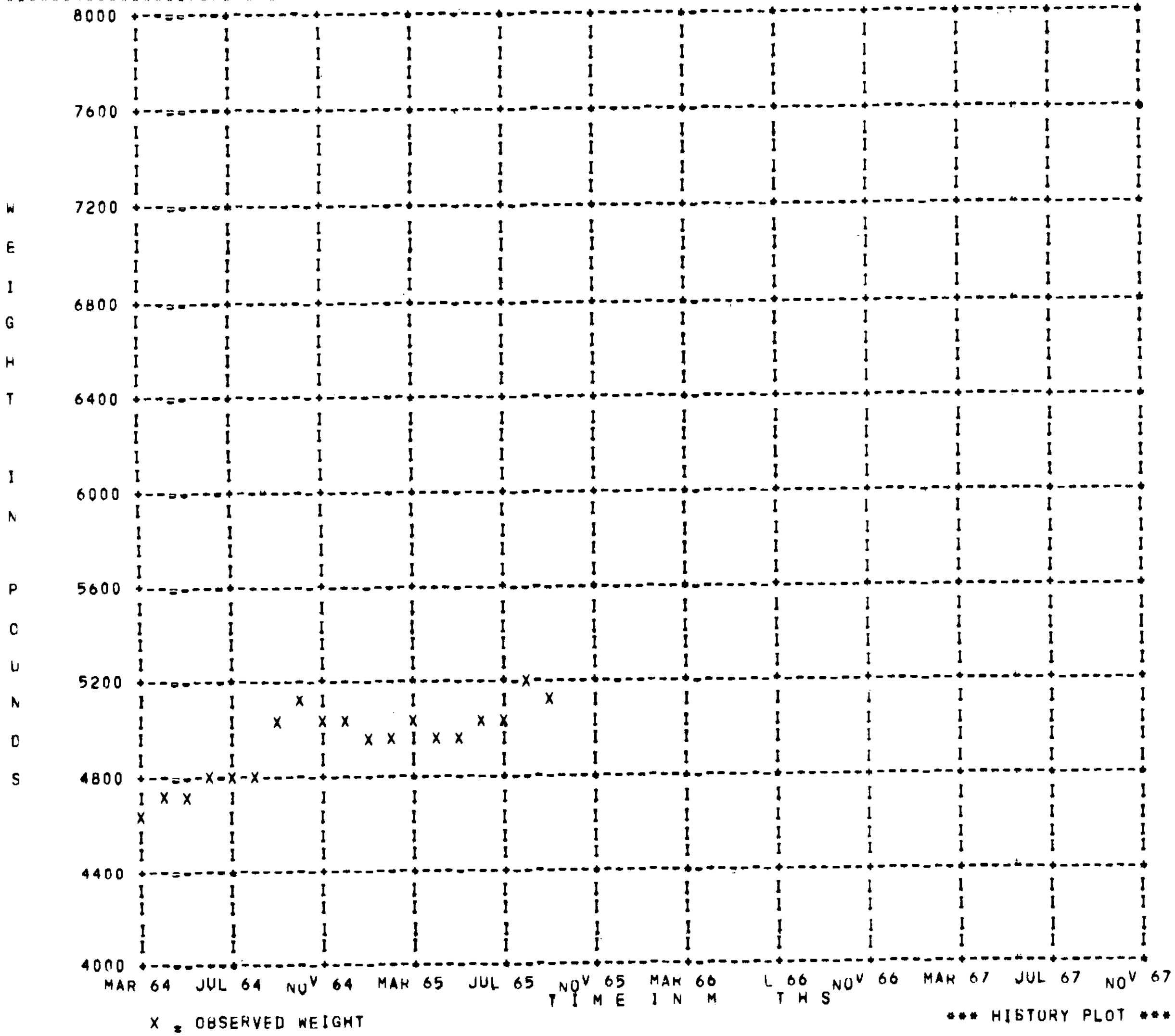
```
9005 FORMAT ( / 33H ....... PREDICTED VALUES .......  /  )
9006 FORMAT ( 3X,1H. / 3X,1H. / 33H   FINAL PREDICTED VALUES ....... ,
   1        / 3X,1H.    )
9008 FORMAT (// 13X,16HCONTROL WEIGHT   ,6X,5H LB   ,5X,
   1        19HTRADE-OFF FACTOR    ,6X,10X,
   2        32HINTERCEPT DATE OF PREDICTED LINE  /
   3        93X,19HWITH CONTROL WEIGHT,5X,5X     /
   4        13X,16HREQUIRED BUYOFF  ,6X,5H LB   ,5X,
   5        19HEQUIVALENT DEFICIT  ,6X,5H LB   ,5X,
   6        39HPROBABILITY OF EXCEEDING CONTROL WEIGHT  ,3X,5H PCT.)
9010 FORMAT (1X,A3,1X,I2,1X,1H*,I8,5H   *   ,3(I3,2X),3H*   ,
   1        I6,3H  *,2X,I6,2X,1H*   ,I8,3H   *,17X,2H*   ,I7,3H *    ,7X,
   2        3H *   , I7,2H *   )
9011 FORMAT (1X,A3,1X,I2,1X,1H*,I8,5H   *   ,3(I3,2X),3H*   ,
   1        6X,3H  *,10X,1H*   ,I8,3H   *,17X,2H*   ,I7,3H *    ,I7,
   2        3H *   , I7,2H *   )
9020 FORMAT (1X,A3,1X,I2,1X,1H*,I8,5H   *   ,3(I3,2X),3H*   ,
   1        I6,3H  *,2X,I6,2X,1H*   ,I8,3H   *,17X,2H*   ,I7,3H *    ,7X,
   2        3H *   , I7,2H *   )
9021 FORMAT (1X,A3,1X,I2,1X,1H*,I8,5H   *   ,3(I3,2X),3H*   ,
   1        6X,3H  *,10X,1H*   ,I8,3H   *,17X,2H*   ,I7,3H *    ,I7,
   2        3H *   , I7,2H *   )
9030 FORMAT (1X,A3,1X,I2,1X,1H*,I8,5H   *   ,3(I3,2X),3H*   ,
   1        I6,3H  *,2X,I6,2X,1H*   ,I8,3H   *,17X,2H*   ,I7,3H *    ,7X,
   2        3H *   , I7,2H *   )
9031 FORMAT (1X,A3,1X,I2,1X,1H*,I8,5H   *   ,3(I3,2X),3H*   ,
   1        6X,3H  *,10X,1H*   ,I8,3H   *,17X,2H*   ,I7,3H *    ,7X,
   2        3H *   , I7,2H *   )
9040 FORMAT (1X,A3,1X,I2,1X,1H*,I8,5H   *   ,3(I3,2X),3H*   ,
   1        I6,3H  *,2X,I6,2X,1H*   ,I8,3H   *,17X,2H*   ,I7,3H *    ,7X,
   2        3H *   , I7,2H *   )
9041 FORMAT (1X,A3,1X,I2,1X,1H*,I8,5H   *   ,3(I3,2X),3H*   ,
   1        6X,3H  *,10X      ,1H*   ,I8,3H   *,17X,2H*   ,I7,3H *    ,I7,
   2        3H *   , I7,2H *   )
C
9050 FORMAT (9X,12(A3,1X,I2,2X))
9051 FORMAT (48X,27HT I M E   I N   M O N T H S    / 12X,
   1        21H X = OBSERVED WEIGHT   ,15X
   2        27H * = MEAN ESTIMATED WEIGHT   ,15X,
   3        23H + = CONFIDENCE LIMITS    )
9055 FORMAT (1X,A3,1X,I2,2H *,10X,1H*,17X,1H*,3(10X,1H*),
   1        17X,1H*,3(9X,1H*) )
     END
```

Reference No. 61.0
Issue Date 23 Dec 1965
Supersedes 13 August 1965

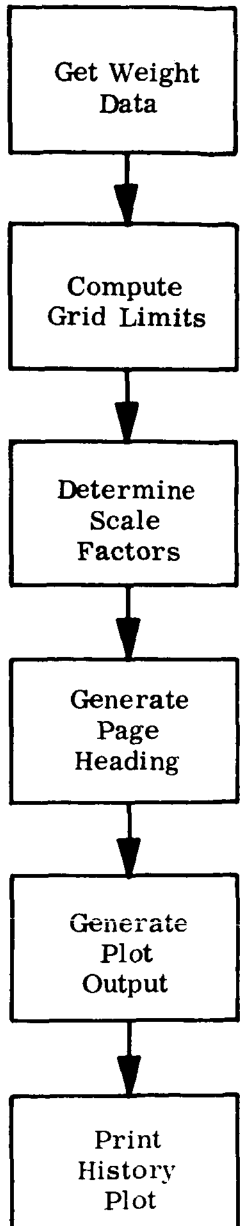
HISTORY PLOT PROGRAM - 61S (HIS)

This routine produces a printed plot of the historical weight observations. Input to 61S consists of a case number and its associated weight data. This input is specified on a data card which is read by the Control Program 55S. For a detailed description of this data card see program 55S.

PREDICTION ANALYSIS TECHNIQUE	BLK II SPACECRAFT MISSION 504 + SUBS	CASE NUMBER	DATE 11/21/65	REQUIRED BUYOFF LB
HISTORY PLOT	LEM ASCENT	0611900	INPUT TAPE 10/08/65	EQUIVALENT S/C OR L/V
COMPUTER PROGRAM 61S	AT BURNOUT		FILE STATUS NO. 20	DEFICIT LB



Reference No. 61.2
Issue Date 23 Dec 1965
Supersedes 13 August 1965



```
$IBFTC HISTRY  LIST,REF
C
C *** HISTORY PLOT ROUTINE FOR THE TREND SYSTEM
C
      INTEGER REELS,WDCT
      INTEGER  CASENO,TILT
      INTEGER  CASNO
      INTEGER  YEAR
      COMMON   /ACCESS/  HC(100),WDCT,ID(12),PROG
      COMMON  /GGDATE/DATE(2)
      COMMON   /SYSTEM/  NTAPES,REELS(15),CNTRLS(15),FILES(15),LRS(15),
     1                   POS(15),IRLPOS(15),RWCNT(15),UNITS(15)
      COMMON  /PFILE/ NFYLE(4,300),NFY(4)
      COMMON   /PPOG/    ACTUAL(100),CALC(100),EST(100),COM(12),
     1                   LSQR(100),MEAN(100),MCONF(100),UW2(100),
     2                   MSQR(100),PCONF(100),S1(100),S2(100),
     3                   TIME(100),WEIGHT(100),BUY(100),TITLE(9),
     4                   N,NTOT
      COMMON /BLOCK/  NCASE (10) , TTITLE(90) , AAA    (150) , NANUM(10),
     1                TBLOCK(300), WBLOCK(300), EBLOCK(300)  , NNUM (10),
     2                CBLOCK(300), ABLOCK(300), DBLOCK(300)
      COMMON  /STT/    N21,D1,D2,J1RUN
      COMMON  /HHP/  IOPT,JOPT,KOPT,NPATH
      COMMON   /ZEE/ ZLINE1(6),ZLINE2(6),ZLINE3(6),ZLINE4(6)
      DIMENSION  GRID (867),NSCALE(5)
      DIMENSION  XMONTH(12),PNME(4),TAB(24),BLKM(12),IBLKY(12)
      DIMENSION MASK(7)
      EQUIVALENCE  (COM(3),MONTH),(COM(4),YEAR)
      DATA  TAB  /1.0,2.0,5.0,10.0,20.0,25.0,50.0,100.0,200.0,250.0,
     1            500.0,1000.0,2000.0,2500.0,5000.0,10000.0,20000.0,
     2            25000.0,50000.0,100000.0,200000.0,250000.0,
     3            500000.0,1000000.0/
      DATA MASK/1,1,1,1,1,1,1/
      DATA  XMONTH /3HJAN,3HFEB,3HMAR,3HAPR,3HMAY,3HJUN,
     1              3HJUL,3HAUG,3HSEP,3HOCT,3HNOV,3HDEC/
      DATA  PNME  /6HHISTOR,6HY PLOT,6H      ,6H      /
      DATA  PNO   /3H61S/
      DATA  NSCALE  /1,0,-1,0,-1/
      XSTART = 1.0
      XEND = 45.0
      DO 500 MUO=1,NTOT
      CALL GETDAT(IPSWT,MUO,CASENO)
      CALL TITEL(CASENO,MASK)
      LKASE = CASENO / 1000000
      CASNO=CASENO
      IF (IPSWT .EQ. 12)  GO TO 210
      YMIN = 10.0E+10
      YMAX = 0.0
      DO 150 IP=1,N
      IF (UW2(IP) .GT. YMAX)  YMAX = UW2(IP)
      IF (UW2(IP) .LT. YMIN)  YMIN = UW2(IP)
  150 CONTINUE
```

```
      DMIN = YMIN
      DMAX = YMAX
      DEL = (DMAX - DMIN) / 10.0
      DO 200 JO=1,24
      IF (DEL .LE. TAB(JO))  GO TO 205
  200 CONTINUE
  205 INK = TAB(JO)
      MIN = DMIN
      YMIN = (MIN / INK) * INK
      YMAX = YMIN + 10.0 * TAB(JO)
      GO TO 212
  210 CONTINUE
      YMIN = COM(11)
      YMAX = COM(12)
  212 IYERE = YEAR - 1900
      MY = MONTH
      DO 220 MI=1,12
      BLKM(MI) = XMONTH(MY)
      IBLKY(MI) = IYERE
      DO 216 KI=1,4
      MY = MY + 1
      IF (MY .LE. 12)  GO TO 216
      MY = 1
      IYERE = IYERE + 1
  216 CONTINUE
  220 CONTINUE
      WRITE (6,9000)
      IF (LKASE .NE. 0)  GO TO 230
      WRITE (6,9004)  ZLINE1,DATE,ZLINE2,PNME,ZLINE3,CASNO,
     1                D1,D2,PNO,ZLINE4,N21
      GO TO 234
  230 WRITE (6,9002)  ZLINE1,DATE,ZLINE2,PNME,ZLINE3,CASNO,
     1                D1,D2,PNO,ZLINE4,N21
  234 CONTINUE
      WRITE (6,9001)
      CALL PLOT1 (NSCALE,10,5,11,8)
      CALL PLOT2 (GRID,XEND,XSTART,YMAX,YMIN)
      CALL PLOT3 (1HX,TIME(1),UW2(1),N)
      CALL OMIT (1)
      CALL FPLOT4 (41,41H          W E I G H T   I N   P O U N D S   )
      WRITE (6,9010) (BLKM(J),IBLKY(J),J=1,12)
      WRITE (6,9011)
 500  CONTINUE
      CALL GOODE
C
C
 9000 FORMAT (1H1,131(1H*))
 9001 FORMAT (1X,131(1H*))
 9002 FORMAT (1X,34H* PREDICTION ANALYSIS TECHNIQUE * ,6A6,   1X,1H*,
     1          20H CASE NUMBER * DATE ,6X,A6,A2,18H * REQUIRED BUYOFF,
     2          6X,1H*  / 3H * ,29(1H*),3H * ,6A6,2H *,13X,1H*,21X,1H*,1
     3          6X,7H LB   * / 3H * ,4A6,5X,3H * ,6A6,2H *,3X,I7,5X,
     4          13H* INPUT TAPE ,A6,A2,25H * EQUIVALENT S/C OR L/V*  /
```

```
     5          19H * COMPUTER PROGRAM,3X,A3,7X,3H * ,6A6,2H *,13X,
     6          17H*.FILE STATUS NO.  ,I4,12H * DEFICIT   ,6X,7H LB    *
  9004 FORMAT (1X,34H* PREDICTION ANALYSIS TECHNIQUE * ,6A6,   1X,1H*,
     1          20H CASE NUMBER * DATE ,6X,A6,A2,18H * REQUIRED BUYOFF,
     2          6X,1H*  / 3H * ,29(1H*),3H * ,6A6,2H *,13X,1H*,21X,1H*,1
     3          6X,7H LB   * / 3H * ,4A6,5X,3H * ,6A6,2H *,3X,1H0,I6,3X,
     4          13H* INPUT TAPE ,A6,A2,25H * EQUIVALENT S/C OR L/V*  /
     5          19H * COMPUTER PROGRAM,3X,A3,7X,3H * ,6A6,2H *,13X,
     6          17H* FILE STATUS NO.  ,I4,12H * DEFICIT   ,6X,7H LB    * )
  9010 FORMAT (9X,12(A3,1X,I2,2X))
  9011 FORMAT (48X,27HT I M E   I N   M O N T H S    /12X,
     1          21H X = OBSERVED WEIGHT   ,50X,20H*** HISTORY PLOT ***   )
C
       END
```

Reference No. 62.0
Issue Date 23 Dec 1965
Supersedes 13 August 1965

TITLING SUBROUTINE - 62S (TITEL)

This subroutine is available to each program that requires automatic title generation from the case number.

The subroutine breaks up the case number into four parts -

(06 ¦ 11 ¦ 2 ¦ 36).

Each part is used in a series of table lookups to generate one line of titling for tabular output. Upon return from the subroutine, the case title is stored in the labeled COMMON block /ZEE/Z1(6), Z2(6), Z3(6), Z4(6) with one line of titling stored in each of the arrays Z1, Z2, Z3, Z4.

Control over this routine is maintained by the calling program.

Case Identification Number and Title List

The seven-digit number used to identify specific items for which weights are tabulated is also used to provide titles for computer printouts. These titles have been revised as of September 28, 1965, and are to be used for all titling until a further revision (now in preparation) is issued. The revised title list is as follows:

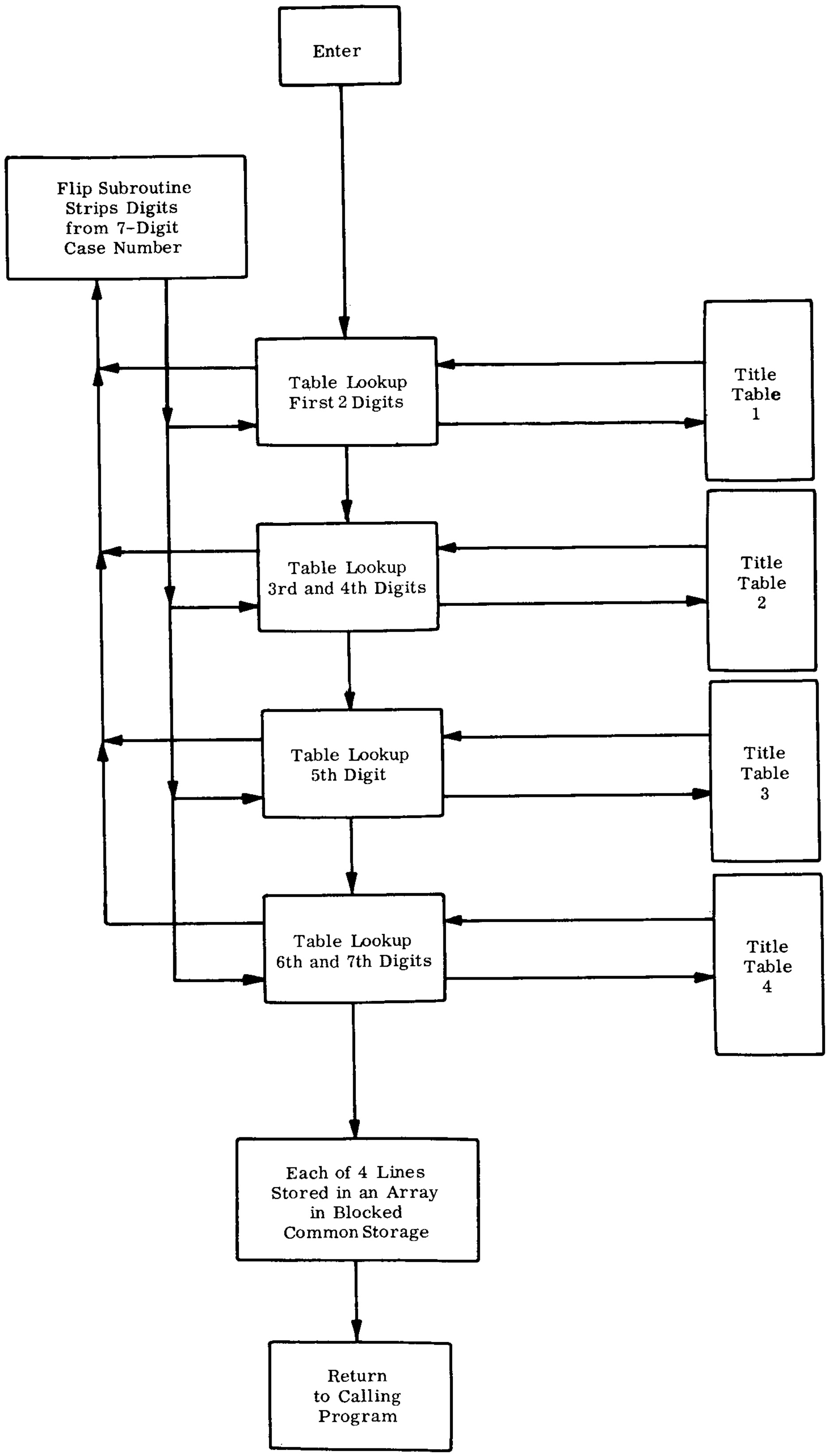
Case Identification Number and Title List

First Two Digits Class and Serial Number of Vehicles	3rd and 4th Digits Stage or Module	5th Digit Mission Phase and Time	6th and 7th Digits Functional Code
51 = Saturn V Launch Vehicle SA-501 52 = Saturn V Launch Vehicle SA-502 53 = Saturn V Launch Vehicle SA-503 56 = Saturn V Launch Vehicle SA-504 & Subs 54 = Saturn V Launch Vehicle SA-504 55 = Saturn V Launch Vehicle SA-505 57 } etc. 69 }	00 = Launch Vehicle Payload Capability 10 = S-IC Stage Dry 11 = S-IC Stage at Separation 20 = S-IC/S-II Interstage at Ground Ignition 22 = S-IC/S-II Interstage Ground Ignition (2nd Vehicle) 30 = S-II Stage Dry 31 = S-II Stage at Separation 40 = S-II/S-IVB Interstage at Ground Ignition 50 = S-IVB Stage Dry 51 = S-IVB Stage at Separation 60 = Instrument Unit 61 = Instrument Unit at Ground Ignition	0 = (Blank) 1 = (Blank) 2 = (Blank) 3 = (Blank) 4 = (Blank) 5 = (Blank) 6 = (Blank) 7 = (Blank) 8 = (Blank) 9 = (Blank)	00 = (Blank) 03 = Structure-Stage 04 = Propulsion System 06 = Equipment and Instrumentation 07 = Residual and Reserve Propellant 08 = Aux. Propellant-Power Roll 09 = Ullage Rocket Propellant
21 = Saturn IB Launch Vehicle SA-201 22 = Saturn IB Launch Vehicle SA-202 23 = Saturn IB Launch Vehicle SA-203 24 = Saturn IB Launch Vehicle SA-204 25 = Saturn IB Launch Vehicle SA-205 26 = Saturn IB Launch Vehicle SA-206 27 = Saturn IB Launch Vehicle SA-207 28 } etc. 39 }	00 = Launch Vehicle Payload Capability 10 = S-IB Stage Dry 11 = S-IB Stage at Separation 20 = S-IB/S-IVB Interstage at Ground Ignition 50 = S-IVB Stage Dry 51 = S-IVB Stage at Separation 60 = Instrument Unit 61 = Instrument Unit at Ground Ignition	Same as above	Same as above

Reference No. 62.0.2
Issue Date 23 Dec 1965
Supersedes 13 August 1965

Case Identification Number and Title List (Cont.)

First Two Digits Class and Serial Number of Vehicles	3rd and 4th Digits Stage or Module	5th Digit Mission Phase and Time	6th and 7th Digits Functional Code
01 = Block I Spacecraft Mission 501 02 = Block I Spacecraft Mission 502 03 = Block II Spacecraft Mission 503 04 = Block II Spacecraft Mission 504 05 = Block II Spacecraft Mission 505 06 = Block II Spacecraft Missiŏn 504 & Subs 07 = Block II Spacecraft Mission 507 08 } etc. 19 }	00 = Total Spacecraft 10 = LEM Ascent Inert 11 = LEM Ascent 12 = LEM Ascent Total 20 = LEM Descent Inert 21 = LEM Descent 22 = LEM Descent Total 30 = LEM Gross 40 = Spacecraft Adapter 50 = Service Module Dry 51 = Service Module Inert 52 = Service Module Gross 60 = Command Module Dry 61 = Command Module 62 = Command Module Gross 67 = Command Module Gross 70 = Launch Escape System 71 = Launch Escape System	0 = (Blank) 1 = At Liftoff 2 = (Blank) 3 = (Blank) 4 = At Injection 5 = At Injection 6 = (Blank) 7 = At Burnout 8 = (Blank) 9 = At Burnout	00 = (Blank) 01 = Crew Systems 02 = Stabilization and Control 03 = Structure 04 = Propulsion System 06 = Navigation and Guidance 08 = Propellant-Useable 11 = Earth Landing System 13 = Ballast Installation Provisions 14 = Propulsion System Jettison 16 = Environmental Control System 17 = Reaction Control (Useful Load) 21 = Controls and Displays 23 = CM Boost Protective Cover 24 = Propulsion System Jettison Motor Skirt 26 = Instrumentation 27 = Electrical Power (Useful Load) 31 = Scientific Equipment (Useful Load) 34 = Propulsion System Pitch Control 36 = Electrical Power Sys. 37 = Environmental Control (Useful Load) 41 = Landing Gear 46 = Reaction Control Sys. 47 = Main Propulsion (Useful Load) 56 = Communications 57 = Crew Systems (Useful Load) 66 = Separation System 67 = Ballast 76 = Propellant Dispersal System
71 = Block I Spacecraft Mission 201 72 = Block I Spacecraft Mission 202 73 = Block I Spacecraft Mission 203 74 = Block I Spacecraft Mission 204 75 = Block I Spacecraft Mission 205 76 = Block I Spacecraft Mission 206 77 = Block II Spacecraft Mission 207 78 } etc. 99 }	Same as above	Same as above	


Enter
Flip Subroutine Strips Digits from 7-Digit Case Number
Table Lookup First 2 Digits
Title Table 1
Table Lookup 3rd and 4th Digits
Title Table 2
Table Lookup 5th Digit
Title Table 3
Table Lookup 6th and 7th Digits
Title Table 4
Each of 4 Lines Stored in an Array in Blocked Common Storage
Return to Calling Program

Reference No. 62.2
Issue Date 23 Dec 1965
Supersedes 13 August 1965

```
$IBFTC TITEL    LIST,REF
      SUBROUTINE TITEL (LCASE,MASK)
C
      COMMON  /ZEE/ ZLINE1(6),ZLINE2(6),ZLINE3(6),ZLINE4(6)
C
      DIMENSION  MC1(46),MC2(46),MC3(46),MC4(46),MC6(10),
     1           MC8(10),MC9(46),FTD(36),TFD1(30),TFD2(42),
     2           TFD3(18),TFD4(60),TFD5(48),IDOT(10),JTAB(46),
     3           FID(60),SSD1(42),SSD2(60),SSD3(60),SSD4(54)
      DIMENSION  MASK(7)
C
C *** TABLES OF TITLES
C
      DATA FTD /36HSATURN V LAUNCH VEHICLE SA-             ,
     1          36HSATURN 1B LAUNCH VEHICLE SA-           ,
     2          36HBLOCK I SPACECRAFT MISSION             ,
     3          36HBLOCK II SPACECRAFT MISSION            ,
     4          36HSAT  V  LAUNCH VEHICLE SA-      + SUBS,
     5          36HBLK II SPACECRAFT MISSION       + SUBS/
      DATA TFD1 /36HLAUNCH VEHICLE PAYLOAD CAPABILITY     ,
     1          36HS-IVB STAGE DRY                        ,
     2          36HS-IVB STAGE AT SEPARATION              ,
     3          36HINSTRUMENT UNIT                        ,
     4          36HINSTRUMENT UNIT AT GROUND IGNITION     /
      DATA TFD2 /36HS-IC STAGE DRY                        ,
     1          36HS-IC STAGE AT SEPARATION               ,
     2          36HS-IC/S-II INTERSTAGE AT GRND IGNIT     ,
     3          36HS-IC/S-II INTERSTAGE GR IG (2ND VEH),
     4          36HS-II STAGE DRY                         ,
     5          36HS-II STAGE AT SEPARATION               ,
     6          36HS-II/S-IVB INTERSTAGE AT GRND IGNIT /
      DATA TFD3 /36HS-IB STAGE DRY                        ,
     1          36HS-IB STAGE AT SEPARATION               ,
     2          36HS-IB/S-IVB INTERSTAGE AT GRND IGNIT /
      DATA TFD4 /36HTOTAL SPACECRAFT                      ,
     1          36HLEM ASCENT INERT                       ,
     2          36HLEM ASCENT                             ,
     3          36HLEM ASCENT TOTAL                       ,
     4          36HLEM DESCENT INERT                      ,
     5          36HLEM DESCENT                            ,
     6          36HLEM DESCENT TOTAL                      ,
     7          36HLEM GROSS                              ,
     8          36HSPACECRAFT ADAPTER                     ,
     9          36HSERVICE MODULE DRY                     /
      DATA TFD5 /36HSERVICE MODULE INERT                  ,
     1          36HSERVICE MODULE GROSS                   ,
     2          36HCOMMAND MODULE DRY                     ,
     3          36HCOMMAND MODULE                         ,
     4          36HCOMMAND MODULE GROSS                   ,
     5          36HCOMMAND MODULE GROSS                   ,
     6          36HLAUNCH ESCAPE SYSTEM                   ,
     7          36HLAUNCH ESCAPE SYSTEM                   /
```

```
      DATA FID  /36H                                        ,
     1          36HAT LIFTOFF                               ,
     2          36H                                         ,
     3          36H                                         ,
     4          36HAT INJECTION                             ,
     5          36HAT INJECTION                             ,
     6          36H                                         ,
     7          36HAT BURNOUT                               ,
     8          36H                                         ,
     9          36HAT BURNOUT                               /
      DATA SSD1 /36H                                        ,
     1          36HSTRUCTURE-STAGE                          ,
     2          36HPROPULSION SYSTEM                        ,
     3          36HEQUIPMENT AND INSTRUMENTATION            ,
     4          36HRESIDUAL AND RESERVE PROPELLANT          ,
     5          36HAUX. PROPELLANT-POWER ROLL               ,
     6          36HULLAGE ROCKET PROPELLANT                 /
      DATA SSD2 /36H                                        ,
     1          36HCREW SYSTEMS                             ,
     2          36HSTABILIZATION AND CONTROL                ,
     3          36HSTRUCTURE                                ,
     4          36HPROPULSION SYSTEM                        ,
     5          36HNAVIGATION AND GUIDANCE                  ,
     6          36HPROPELLANT-USEABLE                       ,
     7          36HEARTH LANDING SYSTEM                     ,
     8          36HBALLAST INSTALLATION PROVISIONS          ,
     9          36HPROPULSION SYSTEM JETTISON               /
      DATA SSD3 /36HENVRIONMENTAL CONTROL SYSTEM            ,
     1          36HREACTION CONTROL (USEFUL LOAD)           ,
     2          36HCONTROLS AND DISPLAYS                    ,
     3          36HCM BOOST PROTECTIVE COVER                ,
     4          36HPROPULSION SYS JETTISON MOTOR SKIRT      ,
     5          36HINSTRUMENTATION                          ,
     6          36HELECTRICAL POWER (USEFUL LOAD)           ,
     7          36HSCIENTIFIC EQUIPMENT (USEFUL LOAD)       ,
     8          36HPROPULSION SYSTEM PITCH CONTROL          ,
     9          36HELECTRICAL POWER SYSTEM                  /
      DATA SSD4 /36HENVIRONMENTAL CONTROL (USEFUL LOAD)     ,
     1          36HLANDING GEAR                             ,
     2          36HREACTION CONTROL SYSTEM                  ,
     3          36HMAIN PROPULSION (USEFUL LOAD)            ,
     4          36HCOMMUNICATIONS                           ,
     5          36HCREW SYSTEMS (USEFUL LOAD)               ,
     6          36HSEPARATION SYSTEM                        ,
     7          36HBALLAST                                  ,
     8          36HPROPELLANT DISPERSAL SYSTEM              /
      DATA IDOT  /1,7,13,19,25,31,37,43,49,55/
      DATA JTAB  /0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,20,21,22,
     1           23,24,26,27,30,31,34,36,37,40,41,46,47,50,51,52,
     2           56,57,60,61,62,66,67,70,71,76/
      DATA MC1 /1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
     1         0,0,0,0,0,0,0,0,7,13,0,0,0,19,25,0,0,0,0,0,0/
      DATA MC2 /0,0,0,0,0,0,0,0,0,0,1,7,0,0,0,0,0,13,0,19,0,0,0,0,25,
```

```
     1              31,0,0,0,37,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0/
      DATA MC3 /0,0,0,0,0,0,0,0,0,0,1,7,0,0,0,0,0,13,0,0,0,0,0,0,0,
     1              0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0/
      DATA MC4 /1,0,0,0,0,0,0,0,0,0,7,13,19,0,0,0,0,25,31,37,0,0,0,0,43,
     1              0,0,0,0,49,0,0,0,55,1,7,0,0,13,19,25,0,31,37,43,0/
      DATA MC6 /1,7,13,19,25,31,37,43,49,55/
      DATA MC8 /1,0,0,7,13,0,19,25,31,37/
      DATA MC9 /1,7,13,19,25,0,31,0,37,0,0,43,0,49,55,1,7,0,
     1              13,0,19,25,31,37,0,43,49,55,1,0,7,13,19,0,0,0,
     2              25,31,0,0,0,37,43,0,0,49/
      DATA  IONE,ITWO,ITHREE,IFOUR,IFIVE,ISIX,ISEVEN  /1,2,3,4,5,6,7/
      DATA  BLANK  /6H      /
      DO 100 NN1=1,6
      ZLINE1(NN1) = BLANK
      ZLINE2(NN1) = BLANK
      ZLINE3(NN1) = BLANK
  100 ZLINE4(NN1) = BLANK
      JCODE = 0
C
C *** ROUTINE FOR FIRST AND SECOND DIGIT
C
      IA = 0
      IB = LCASE
      CALL FLIP (IA,IB,IONE)
      ISAVE = IA
      IA = IA + 1
      GO TO (221,221,222,222,500,223,223,224,224,224),IA
  221 ICODE = 4
      GO TO 225
  222 ICODE = 2
      GO TO 225
  223 ICODE = 1
      GO TO 225
  224 ICODE = 3
  225 IA = 0
      CALL FLIP (IA,IB,ITWO)
      IF (IA .EQ. 6 .AND. ISAVE .EQ. 5)  JCODE = 5
      IF (IA .EQ. 6 .AND. ISAVE .EQ. 0)  JCODE = 6
      IF (IA .GE. 7 .AND. ISAVE .GE. 7)  ICODE = 4
      IF (IA .EQ. 1 .AND. ISAVE .EQ. 0)  ICODE = 3
      IF (IA .EQ. 2 .AND. ISAVE .EQ. 0)  ICODE = 3
      GO TO (251,252,252,251),ICODE
C
  251 IBC = 500
      IF (IA .GE. 7 .AND. ISAVE .GE. 7)  IBC = 200
      GO TO 253
  252 IBC = 200
      IF (IA .EQ. 1 .AND. ISAVE .EQ. 0)  IBC = 500
      IF (IA .EQ. 2 .AND. ISAVE .EQ. 0)  IBC = 500
  253 IA1 = IA
      IF (ISAVE .EQ. 0 .AND. IA .EQ. 6)  IA1 = 4
      IF (ISAVE .EQ. 5 .AND. IA .EQ. 6)  IA1 = 4
      IBC = IBC + IA1
```

```
      IF (MASK(1) .EQ. 0 .AND. MASK(2) .EQ. 0)  GO TO 275
      GO TO (256,258,260,262),ICODE
  256 IF (JCODE .EQ. 5)  GO TO 257
      IBEG = 1
      NNC = 18
      NND = 18
      GO TO 264
  257 IBEG = 25
      NNC = 12
      NND = 18
      GO TO 264
  258 IBEG = 7
      NNC = 24
      NND = 12
      GO TO 264
  260 IBEG = 13
      NNC = 18
      NND = 18
      GO TO 264
  262 IF (JCODE .EQ. 6)  GO TO 263
      IBEG = 19
      NNC = 24
      NND = 12
      GO TO 264
  263 IBEG = 31
      NNC = 12
      NND = 18
  264 DO 265 MAM=1,6
      ZLINE1(MAM) = FTD(IBEG)
  265 IBEG = IBEG + 1
      CALL INDIFF (NNC,NND,ZLINE1(5),IBC,ZILCH)
      IF (IBEG .EQ. 13 .OR. IBEG .EQ. 25)  ZLINE1(6) = ZILCH
C
C *** ROUTINE FOR THIRD AND FOURTH DIGIT
C
C
  275 IA = 0
      CALL FLIP (IA,IB,ITHREE)
      IF (ISAVE .LT. 2 .OR. ISAVE .GT. 6)  GO TO 284
      IF (IA .LT. 1 .OR. IA .GT. 4)  GO TO 280
      KCODE = 2
      GO TO 286
  280 KCODE = 1
      GO TO 286
C
  284 KCODE = 3
  286 CONTINUE
      IC = IA * 10
      IA = 0
      CALL FLIP (IA,IB,IFOUR)
      IC = IC + IA
      IF (MASK(3) .EQ. 0 .AND. MASK(4) .EQ. 0)  GO TO 304
      DO 290 MM=1,46
```

```
      IF (IC .EQ. JTAB(MM))  GO TO 292
  290 CONTINUE
      GO TO 304
  292 GO TO (293,295,299),KCODE
  293 IF (MC1(MM) .EQ. 0)  GO TO 304
      INDEX1 = MC1(MM) - 1
      DO 294 MN=1,6
      JNDEX = INDEX1 + MN
  294 ZLINE2(MN) = TFD1(JNDEX)
      GO TO 304
  295 IF (ICODE .EQ. 2)  GO TO 297
      IF (MC2(MM) .EQ. 0)  GO TO 304
      INDEX1 = MC2(MM) - 1
      DO 296 MN=1,6
      JNDEX = INDEX1 + MN
  296 ZLINE2(MN) = TFD2(JNDEX)
      GO TO 304
  297 IF (MC3(MM) .EQ. 0)  GO TO 304
      INDEX1 = MC3(MM) - 1
      DO 298 MN=1,6
      JNDEX = INDEX1 + MN
  298 ZLINE2(MN) = TFD3(JNDEX)
      GO TO 304
  299 IF (MC4(MM) .EQ. 0)  GO TO 304
      INDEX1 = MC4(MM) - 1
      DO 300 MN=1,6
      JNDEX = INDEX1 + MN
      IF (MM .GT. 34)  GO TO 301
      ZLINE2(MN) = TFD4(JNDEX)
      GO TO 300
  301 ZLINE2(MN) = TFD5(JNDEX)
  300 CONTINUE
  304 CONTINUE
C
C *** ROUTINE FOR FIFTH DIGIT
      IA = 0
      CALL FLIP (IA,IB,IFIVE)
      IF (MASK(5) .EQ. 0)  GO TO 309
      DO 305 MO=1,10
      IF (IA .EQ. JTAB(MO))  GO TO 306
  305 CONTINUE
  306 IF (ICODE .EQ. 1 .OR. ICODE .EQ. 2)  GO TO 309
      IF (MC6(MO) .EQ. 0)  GO TO 309
      INDEX1 = MC6(MO) - 1
      DO 307 MP=1,6
      JNDEX = INDEX1 + MP
  307 ZLINE3(MP) = FID(JNDEX)
C
C
C
C *** ROUTINE FOR SIXTH AND SEVENTH DIGITS
C
  309 IA = 0
```

```
      CALL FLIP (IA,IB,ISIX)
      IE = IA * 10
      IA = 0
      CALL FLIP (IA,IB,ISEVEN)
      IE = IE + IA
C
      IF (MASK(6) .EQ. 0 .AND. MASK(7) .EQ. 0)  GO TO 500
      DO 310 MM=1,46
      IF (IE .EQ. JTAB(MM))  GO TO 312
  310 CONTINUE
      GO TO 500
  312 IF (ICODE .EQ. 3 .OR. ICODE .EQ. 4)  GO TO 320
      IF (IE .GT. 9)  GO TO 500
  315 IF (MC8(MM) .EQ. 0)  GO TO 500
      INDEX1 = MC8(MM) - 1
      DO 316 MN=1,6
      JNDEX = INDEX1 + MN
  316 ZLINE4(MN) = SSD1(JNDEX)
      GO TO 500
C
  320 IF (IE .GT. 76)  GO TO 500
  324 IF (IE .LE. 76)  IZIP = 3
      IF (IE .LE. 36)  IZIP = 2
      IF (IE .LE. 14)  IZIP = 1
      IF (MC9(MM) .EQ. 0)  GO TO 500
      INDEX1 = MC9(MM) - 1
      DO 336 MN=1,6
      JNDEX = INDEX1 + MN
      GO TO (332,333,334),IZIP
  332 ZLINE4(MN) = SSD2(JNDEX)
      GO TO 336
  333 ZLINE4(MN) = SSD3(JNDEX)
      GO TO 336
C
  334 ZLINE4(MN) = SSD4(JNDEX)
  336 CONTINUE
C
C
C
  500 RETURN
      END
```

```
$IBFTC FLIP    LIST,REF
      SUBROUTINE FLIP (I,J,K)
      IDIV = (10**(7-K))
      IC = J / IDIV
      J = MOD (J,IDIV)
      I = (I * 10) + IC
      RETURN
      END
```

```
$IBMAP INDIFE  LIST,REF
       ENTRY   INDIFE
INDIFE SAVE    1,2,4
       CLA*    3,4
       STO     GOO
       STA     SHFT1
       CLA*    4,4
       STA     SHFT2
       STA     SHFT3-1
       ADD     GOO
       STO     GOO
       CLA     =36
       SUB     GOO
       STA     SHFT3
       ZAC
       LDQ*    5,4
SHFT1  LGL     **
       SLW     SAVEA
       STQ     SAVEM
       CLA*    6,4
       LRS     2
       AXT     6,1
       VDP     CONST+6,1,6
       TIX     *-1,1,1
       RQL     18
       CAL     SAVEA
SHFT2  LGL     **
       SLW     SAVEA
       STQ     GOO
       CAL     GOO
       TZE     ZOOM
       ORA     =O006060606060
       TRA     *+2
ZOOM   CAL     =H
       SLW*    7,4
       CAL     SAVEA
       LDQ     SAVEM
       RQL     **
SHFT3  LGL     **
       SLW*    5,4
ER     RETURN  INDIFE
GOO    BSS     1
SAVEA  BSS     1
SAVEM  BSS     1
CONST  DEC     1E5B31,1E4B25,1E3B19,1E2B13,1E1B7,1B1
       END
```

Reference No.	63.0
Issue Date	23 Dec 1965
Supersedes	New

SUMMING PROGRAM - 63S (SUM)

The Summing Program performs a summation on the predicted weights of each functional system which comprise a given stage, module, spacecraft or launch vehicle. It is possible to sum any group of functional systems as long as every member of that group has a case number in which one or more of the leading digits are identical. That is, we could sum the predicted weights for all functional systems having the first four digits of 2211 (i.e., representing the Saturn IB Launch Vehicle SA-202, S-IB stage at separation) or we could sum the predicted weights for all functional systems having the first two digits of 22 (representing the Saturn IB Launch Vehicle SA-202), etc. It is assumed that the trend prediction programs have been executed and there is available as input, the four binary data files representing the linear, nonlinear, Fourier, and logistic modes.

INPUT

For each stage, module, launch vehicle, or spacecraft to be summed, there is required the following set of cards:

CARD A

To sum a typical group of cases, you need a general case number which is a 7-digit number and a mask number which is a number between 1 and 7. The mask number tells the program the number of digits to sum on. The remaining digits on the left (7 - mask number = MN) will be the first MN digits which all of the summed functional systems will have in common. For example, an extreme general case number of 0000000 with a mask number of 7 would sum all of the functional systems on the tape. A general case number of 5650100 with a mask number of 7 would sum all of the functional systems whose case number are greater than or equal to 5650100. A more practical example would be a general case number of 5650100 with a mask number of 4. This would sum all of the functional systems starting from 5650100 to, but not including, 5660000. Another example would be a general case number of 0640000 with a mask number of 3. This would sum all of the functional systems starting from 0640000 to, but not including, 0641000.

The mask number has a second purpose, which is to create a masking index for the Titel Program which will, in turn, generate a general title for the applicable summing run.

The first item of information on Card A is the general case number. The next item of information to be included on this card is an indication of which prediction mode (linear, nonlinear, Fourier, logistic) will be used in the majority of the functional systems to be summed. Thus, if some particular module is composed of six functional systems and it has been decided to use the nonlinear mode for summing in four of the six cases, then the nonlinear mode would be so indicated on this card. The following code is used to make the above indications:

1 Linear mode
2 Nonlinear mode
3 Fourier mode
4 Logistic mode

The summing program will sum the predicted weights from the last observed weight out to the shipping date weight. Thus, if some functional system is not current, the summed weights will be in error, since the program sums from the last observation and does no checking of dates. The third and fourth pieces of information appearing on Card A are the current date of the latest observation (e.g., 9 1965).

The final piece of information is the mask number followed by an asterisk (*).

Thus, a typical Card A would appear as

```
┌──────Columns 7—72──────┐
5650100   2   9   1965   2   *
```

CARD B

For each functional system that is not to be summed in the mode specified on Card A, a type-B card is required. Two options are available:

a. The user may specify a prediction mode other than the one specified on Card A. In this case, all that is required is the case number of the functional system, the desired mode to be summed, and the word TREND.

```
┌─────Columns 7— 72────┐
5650103.     4.     TREND
```

b. The user may specify that a line with constant slope be summed, rather than one of the four trend prediction lines. In this case, it is necessary to give the case number of the functional system, the desired weight at the last observation, and the desired slope of the line. The slope may be positive, negative, or zero.

Columns 7—72

5650106.   629.   -5.

NOTE

All numbers punched on the B-Card will be punched with decimal point.

CARD C

If the word END is followed by an asterisk (punched in Columns 7 - 72), the card signifies the end of summing for the particular general case number specified on Card A.

NOTE

1. Card A and Card C are required whenever a sum is desired. Card B is optional. If Card B is not present, all functional systems will be summed in the mode specified on Card A.
2. Each set of A, B, and C Cards must appear in ascending order by general case number.
3. The last data card in the summing deck contains the words END OF SUMS followed by an asterisk. When this card is encountered, control is returned to SPACE.

EXAMPLE OF A DATA DECK

5650100	1	9	1965	2	*	Sums all functional systems (whose case numbers have the first five [7 - mask number, 7-2] digits 56501) in linear mode; except 5650103 on which the non-linear mode is used, and 5650106 on which an arbitrary line with a slope of 3 is substituted.
5650103	2.	TREND				
5650106.	29.	3.				
END	*					
7470000	3	9	1965	4	*	Sums all functional systems (whose case numbers have the first three (7- mask number, 7-4) digits 747) in the Fourier mode.
END	*					
END OF SUMS		*				

OUTPUT

Output consists of a printed tab which contains the sums of the functional system for each stage or module. An example is given on the following page.

Reference No. 63.1
Issue Date 23 Dec 1965
Supersedes New

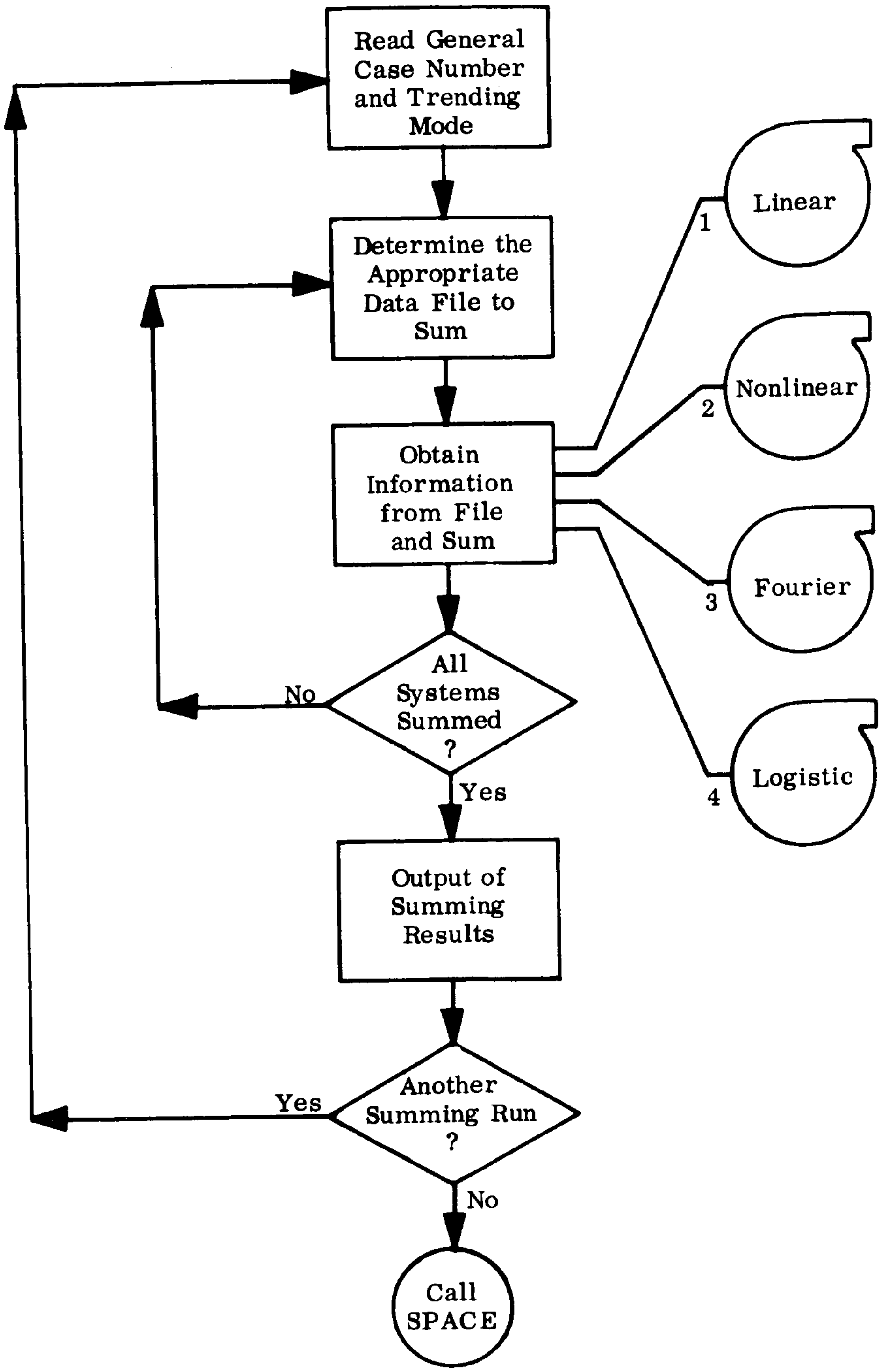
PREDICTION ANALYSIS TECHNIQUE * BLK II SPACECRAFT MISSION 504 + SUBS * CASE NUMBER (GENERAL) 0611900 * DATE 11/21/65 * REQUIRED BUYOFF LB

SUMMING PROGRAM * LEM ASCENT AT BURNOUT * EQUIVALENT S/C OR L/V DEFICIT LB

COMPUTER PROGRAM 63S

THE FOLLOWING ARBITRARY CASES AND THEIR ASSOCIATED WEIGHTS HAVE BEEN USED TO REPLACE THOSE ON RECORD.

CASE NUMBERS IN THIS SUMMATION	MODE FOR THIS CASE	DATE	PREDICTION LINE	MAXIMUM LIKELIHOOD LINE	PERCENT EST	PERCENT CAL	PERCENT ACT	CASE 611903
611901	LIN	SEP 65	5173	5189	57	34	9	1326
611902	LOG	OCT 65	5183	5199	54	32	13	1334
611903	ARB	NOV 65	5194	5209	52	31	17	1342
611904	NLN	DEC 65	5204	5220	49	30	21	1350
611906	NLN	JAN 66	5214	5230	47	28	25	1358
611916	LOG	FEB 66	5224	5240	44	27	29	1366
611921	LIN	MAR 66	5234	5250	42	25	33	1374
611926	NLN	APR 66	5244	5260	39	24	37	1382
611936	NLN	MAY 66	5254	5270	37	22	41	1390
611946	LIN	JUN 66	5264	5280	34	21	45	1398
611956	LIN	JUL 66	5274	5290	32	19	49	1406
		AUG 66	5284	5300	30	18	53	1414
		SEP 66	5294	5310	27	16	57	1422
		OCT 66	5304	5320	25	15	61	1430
		NOV 66	5314	5330	22	13	64	1438
		DEC 66	5324	5340	20	12	68	1446
		JAN 67	5334	5349	17	10	72	1454
		FEB 67	5343	5359	15	9	76	1462
		MAR 67	5353	5369	12	7	80	1470
		APR 67	5363	5379	10	6	84	1478
		MAY 67	5372	5388	7	4	88	1486
		JUN 67	5382	5398	5	3	92	1494
		JUL 67	5392	5408	2	1	96	1502
		AUG 67	5401	5417	0	0	100	1510


Read General Case Number and Trending Mode
Determine the Appropriate Data File to Sum
Obtain Information from File and Sum
All Systems Summed ?
No
Yes
Output of Summing Results
Another Summing Run ?
Yes
No
Call SPACE
1
Linear
2
Nonlinear
3
Fourier
4
Logistic

Reference No. 63.3
Issue Date 23 Dec 1965
Supersedes New

```
$IBFTC SUMING  LIST,REF
      COMMON  /GGDATE/DATE(2)
      COMMON  /ZEE/ ZLINE1(6),ZLINE2(6),ZLINE3(6),ZLINE4(6)
      COMMON   /ACCESS/  HC(100),WDCT,ID(12),PROG
      COMMON   /SYSTEM/  NTAPES,REELS(15),CNTRLS(15),FILES(15),LRS(15),
     1                   POS(15),TRLPOS(15),RWCNT(15),UNITS(15)
      COMMON  /PFILE/ NFYLE(4,300),NFY(4)
      COMMON   /PPOG/    ACTUAL(100),CALC(100),EST(100),COM(12),
     1                   LSQR(100),MEAN(100),MCONF(100),UW2(100),
     2                   MSQR(100),PCONF(100),S1(100),S2(100),
     3                   TIME(100),WEIGHT(100),BUY(100),TITLE(9),
     4                   N,NTOT
       COMMON  /STT/ N21,D1,D2,J1RUN
      REAL  MEAN
      INTEGER  RELLS
      INTEGER WDCT
      INTEGER  GENCAS,EXCASE,OLDTAP,XDCT,CASENO
      INTEGER  DOM(3,25),MOM(3,25)
      EQUIVALENCE    (TOM(1),GENCAS ),(TOM(2),ITAPE),
     1                (TOM(3),MONTH)   ,(TOM(4),MYEAR),(TOM(5),MASKNO)
      EQUIVALENCE (HC(1),IC1),(HC(2),IC2),(HC(3),IC3),(HC(4),IC4)
      EQUIVALENCE    (DOM,BOM),(XOM,MOM)
      DIMENSION  TOM(5),WT(100),XOM(3,25),BOM(3,25),NP(4)
      DIMENSION  WTA(100),WTC(100),WTE(100),WTM(100)
      DIMENSION  XMONTH(12),PNME(4)
      DIMENSION  FMT(16),SKIP1(8),SKIP2(10),FOMT(16),MASK(7)
      DIMENSION  TYPE(5),HEAD(18)
      DIMENSION IWADS(8),NBLN(18),NCAZ(8)
      DATA TYPE /3HLIN,3HNLN,3HFOR,3HLOG,3HARB/
      DATA  NCAX,NBLNK  /4HCASE,6H        /
      DATA FMT /48H(1X,59H*SUMMATION* CASE *  DATE   *   LINE   *  LIN,
     1         48HE  * EST CAL ACT *,       1(1X,I7),     63X,1H*)   /
      DATA SKIP1 /48H 1(1X, 2(1X, 3(1X, 4(1X, 5(1X, 6(1X, 7(1X, 8(1X,/
      DATA SKIP2 /48H  63X,  55X,  47X,  39X,  31X,  23X,  15X,   7X,,
     1              6H      ,6H  71X,/
      DATA  XMONTH  /3HJAN,3HFEB,3HMAR,3HAPR,3HMAY,3HJUN,3HJUL,3HAUG,
     1                3HSEP,3HOCT,3HNOV,3HDEC/
      DATA  PNME  /24HSUMMING PROGRAM         /
      DATA  PNO   /3H635/
      DATA  HEAD   /48HTHE FOLLOWING ARBITRARY CASES AND THEIR ASSOCIAT,
     1              48HED    WEIGHTS HAVE BEEN USED TO REPLACE THOSE ON,
     2              12H RECORD.    /
      DATA  TREND/5HTREND/
      DATA  IONE  /1/
      DATA END/3HEND/,NP/1,1,1,1/
      DATA OF/2HOF/,SUMS/4HSUMS/


      WRITE (6,9013)
      IF( WDCT  .EQ.     0 )  GO TO  1
      NFY(1)=IC1
      NFY(2)=IC2
```

```
      NFY(3)=IC3
      NFY(4)=IC4
C
 1    CALL READH(TOM)
      IF(TOM(1) .EQ. END .AND. TOM(2) .EQ. OF .AND. TOM(3) .EQ. SUMS )
     1      CALL SPACE
      MYEAR = MOD(MYEAR,100)
      DO  600   I = 1,18
600   NBLN(I) = NBLNK
      DO  601   I = 1,8
      NCAZ(I) = NBLNK
601   IWABS(I) = 0
      DO  2  I=1,100
      WT(I) = 0.
      WTA(I) = 0.
      WTC(I) = 0.
      WTE(I) = 0.
      WTM(I) = 0.
2     CONTINUE
      DO  250   I = 1,7
250   MASK(I) = 0
      DO  251   I = 1,16
251   FOMT(I) = FMT(I)
      DO  253   I = 1,3
      DO  254   J = 1,25
      DOM(I,J) = 0
      XOM(I,J) = 0.
254   CONTINUE
253   CONTINUE
      ICOUNT = 0
      KK   = 7 - MASKNO
      DO  3     I = 1,KK
      MASK(I) = 1
3     CONTINUE
      IF  ( GENCAS .NE. 0 ) GO TO 4
      WRITE (6,10001)
10001 FORMAT (1H0,20X,33HERROR - GENERAL CASE NUMBER = 0. )
      CALL SPACE
4     K    = MASKNO
      CALL TITEL (GENCAS,MASK)
      CALL READH(XOM,IMAX)
      IMAX = IMAX / 3 + 1
      IMIN = IMAX - 1
      IGEN = IMAX
      CALL SORT(XOM,IMIN,3,1)
      KGATE = 0
41    KGATE = KGATE + 1
      EXCASE = XOM(1,KGATE)
      NTAPE = XOM(2,KGATE)
      IF  ( XOM(3,KGATE) .EQ. TREND ) XOM(3,KGATE) = 9999999.
      IF  ( XOM(1,KGATE) .EQ. END )  EXCASE =   9999999
 38   NZL= ISIGN(NP(ITAPE),REELS(ITAPE))
```

```
      IF(NP(ITAPE) .GT. NFY(ITAPE))   GO TO 6
      NF= REELS(ITAPE)*1000 + NZL
      CALL READB1(NF,1)
      CASENO=ID(7)
      NTEST = CASENO - GENCAS
      IF( NTEST .GT. 0  .AND. NTEST .LT. (10**K) )      GO TO 5
      IF( NTEST  .GT. (10**K) )                          GO TO 9
   40 NP(ITAPE) = NP(ITAPE) + 1
      IF( NP(ITAPE).GT. NFY(ITAPE))   GO TO 6
      NZL= ISIGN(NP(ITAPE),REELS(ITAPE))
      NF= REELS(ITAPE)*1000 + NZL
      CALL READB1(NF,1)
      CASENO =    ID(7)
      IF( CASENO  .LT.  GENCAS )          GO  TO  40
    5 IF( CASENO  .LT.  EXCASE )          GO  TO 10
      IF( CASENO .EQ. EXCASE )  NP(ITAPE)=NP(ITAPE)+1
6     IF   ( XOM(3,KGATE) .EQ. 9999999. ) GO TO 7
      IF   ( XOM(1,KGATE) .EQ. END ) GO TO 9
252   ICOUNT = ICOUNT + 1
      BOM(1,ICOUNT) = EXCASE
      BOM(2,ICOUNT) = XOM(2,KGATE)
      BOM(3,ICOUNT) = XOM(3,KGATE)
      IF   ( CASENO .EQ. EXCASE ) GO TO 208
      JTAPE = 0
202   JTAPE = JTAPE + 1
      IF   ( JTAPE .EQ. ITAPE ) GO TO 202
      IF   ( JTAPE .GT. 4 )  GO TO 205
203   NZL = ISIGN(NP(JTAPE),REELS(JTAPE))
      IF   ( NP(JTAPE) .GT. NFY(JTAPE) ) GO TO 202
      NF   = REELS(JTAPE) * 1000 + NZL
      CALL READB1(NF,1)
      CASENO = ID(7)
      IF   ( CASENO - EXCASE ) 204, 207, 202
204   NP(JTAPE) = NP(JTAPE) + 1
      GO TO 203
205   NPLUSP = ID(11) + 1
      DO  206  I = 1,NPLUSP
      EST(I) = 1.
      CALC(I)    = 0.
      ACTUAL(I) = 0.
206   CONTINUE
      N  = 1
      GO TO 209
207   NP(JTAPE) = NP(JTAPE) + 1
208   NPLUSP = ID(10) + ID(11)
      NTIM = 2 * NPLUSP + 9
      N    = ID(10)
      CALL READB2 (NF,TIME,NTIM,HISTRY)
      CALL READB2 (NF,EST,NPLUSP,HISTRY)
      CALL READB2 (NF,CALC,NPLUSP,HISTRY)
      CALL READB2 (NF,ACTUAL,NPLUSP,HISTRY)
209   L    = 0
```

```
      DO  61    I=N,NPLUSP
      L    = L + 1
      FEIGHT = XOM(2,KGATE) + ( FLOAT(L-1) * XOM(3,KGATE) )
      WT(L) = WT(L) + FEIGHT
      WTA(L) = WTA(L) + ( FEIGHT * ACTUAL(I) )
      WTC(L) = WTC(L) + ( FEIGHT * CALC(I) )
      WTE(L) = WTE(L) + ( FEIGHT * EST(I) )
      WTM(L) = WTM(L) + FEIGHT
61    CONTINUE
      XOM(2,KGATE) = 5.
      GO TO 41
 69   NP(NTAPE)=NP(NTAPE)+1
 7    NZL= ISIGN(NP(NTAPE),REELS(NTAPE))
      IF( NP(NTAPE) .GT. NFY(NTAPE) )   GO TO 79
      NF= REELS(NTAPE)*1000 + NZL
      CALL READB1(NF,1)
      CASENO =  ID(7)
      IF( CASENO .LT. EXCASE )  GO TO 69
      IF( CASENO .EQ. EXCASE )     GO TO 8
 79   WRITE(6,1001) EXCASE
 1001 FORMAT(1H1,5X12HCASE NUMBER I8,1X21HIS NOT ON TAPE. OMIT.)
      GO TO 41
 8    NP(NTAPE)=NP(NTAPE)+1
      NPLUSP=ID(10)+ID(11)
      NTIM = 2 * NPLUSP + 9
      N=ID(10)
      CALL READB2 (NF,TIME,NTIM,HISTRY)
      CALL READB2 (NF,EST,NPLUSP,HISTRY)
      CALL READB2(NF,CALC,NPLUSP,HISTRY)
      CALL READB2 (NF,ACTUAL,NPLUSP,HISTRY)
      CALL READB2 (NF,BUY,N,HISTRY)
      CALL READB2 (NF,MEAN,NPLUSP,HISTRY)
      NTIM = NTIM - 9
      CALL READB2 (NF,MCONF,NTIM,HISTRY)
      CALL READB2 (NF,UW2,NPLUSP,HISTRY)
      L=0
      DO 81 I=N,NPLUSP
      L=L+1
      WT(L) = WT(L) + UW2(I)
      WTA(L) = WTA(L) + ( UW2(I) * ACTUAL(I) )
      WTC(L) = WTC(L) + ( UW2(I) * CALC(I) )
      WTE(L) = WTE(L) + ( UW2(I) * EST(I) )
      WTM(L) = WTM(L) + MEAN(I)
81    CONTINUE
      GO TO 41
  9   IF (EXCASE .NE. 9999999 )   GO TO 6
C
C                               CONDITIONS CALCULATIONS AND PRINT ROUTINE
C
      DO  260  I = 1,100
      IF  ( WT(I) .EQ. 0. ) GO TO 261
260   CONTINUE
```

```
261    ISTOP1 = I - 1
       XOM(1,IMAX) = XOM(1,IGEN)
       XOM(2,IMAX) = ITAPE
       XOM(3,IMAX) = 9999999.
       XOM(1,IGEN) = END
       ISTOP2 = IGEN - 1
       CALL SORT(XOM,ISTOP2,3,1)
       DO  262  I = 1,ISTOP2
       MOM(1,I) = XOM(1,I)
       IF  ( XOM(3,I) .NE. 9999999. )  XOM(2,I) = 5.
       MOM(2,I) = XOM(2,I)
262    CONTINUE
       DO  264  I = 1,8
       IF  ( DOM(1,I) .EQ. 0   ) GO TO 265
264    CONTINUE
265    ISTOP3 = I - 1
       WRITE (6,9000)
       LKASE = GENCAS
       LKASE = LKASE / 1000000
       IF  ( LKASE .NE. 0 ) GO TO 266
       WRITE (6,9004) ZLINE1,DATE,ZLINE2,PNME,ZLINE3,GENCAS,PNO,ZLINE4
       GO TO 267
266    WRITE (6,9002) ZLINE1,DATE,ZLINE2,PNME,ZLINE3,GENCAS,PNO,ZLINE4
267    WRITE (6,9001)
       IF  ( ISTOP3 .LE. 0 ) GO TO 269
       DO  268  I = 1,ISTOP3
268    NCAZ(I) = NCAX
       FOMT(13) = SKIP1(ISTOP3)
       FOMT(15) = SKIP2(ISTOP3)
       GO TO 270
269    FOMT(13) = SKIP2(9)
       FOMT(14) = SKIP2(9)
       FOMT(15) = SKIP2(10)
       WRITE(6,9003) (NBLN(I),I=1,9),(NBLN(I),I=10,18),(NBLN(I),I=1,8)
       WRITE(6,FOMT)
       GO TO 275
270    WRITE (6,9003) (HEAD(I),I=1,9),(HEAD(I),I=10,18),(NCAZ(I),I=1,8)
       DO  274  I = 1,ISTOP3
274    CONTINUE
       WRITE (6,FOMT) (DOM(1,I),I=1,ISTOP3)
275    WRITE (6,9001)
       MONTH = MONTH - 1
C
C                                      START PREDICTION PRINTING
C
       DO  350  I = 1,ISTOP1
       IF  ( I .GT. ISTOP2 ) GO TO 351
       CASENO = MOM(1,I)
       LOCA = MOM(2,I)
351    CONTINUE
       MONTH = MONTH + 1
       IF  ( MONTH .LE. 12 ) GO TO 352
```

```
      MONTH = 1
      MYEAR = MYEAR + 1
352   CONTINUE
      NWT=WT(I) + .5
      NACT = ( 100. * WTA(I) ) / WT(I) + .5
      NCALC = ( 100. * WTC(I) ) / WT(I) + .5
      NEST = ( 100. * WTE(I) )  / WT(I) + .5
      NWTM = WTM(I) + .5
      IF ( ABS( WT(I) - WTA(I) - WTC(I) - WTE(I) ) .GT. .1 ) GO TO 201
      IF ( ISTOP3 .EQ. 0 ) GO TO 354
      DO  353  J = 1,ISTOP3
      IWABS(J) = BOM(2,J) + ( FLOAT(I-1) * BOM(3,J) )
353   CONTINUE
354   CONTINUE
      IF ( I .GT. ISTOP2 ) GO TO 358
      IF ( ISTOP3 .EQ. 0 ) GO TO 750
      WRITE (6,9032) CASENO,TYPE(LOCA),XMONTH(MONTH),MYEAR,
     1               NWT,NWTM,NEST,NCALC,NACT,(IWABS(J),J=1,ISTOP3)
      GO TO 350
358   CONTINUE
      IF ( ISTOP3 .EQ. 0 ) GO TO 751
      WRITE (6,9037) XMONTH(MONTH),MYEAR,NWT,NWTM,NEST,NCALC,NACT,
     1               (IWABS(J),J=1,ISTOP3)
      GO TO 350
750   WRITE (6,9032) CASENO,TYPE(LOCA),XMONTH(MONTH),MYEAR,
     1               NWT,NWTM,NEST,NCALC,NACT
      GO TO 350
751   WRITE (6,9037) XMONTH(MONTH),MYEAR,NWT,NWTM,NEST,NCALC,NACT
350   CONTINUE
355   IF ( ISTOP2 .LE. ISTOP1 ) GO TO 1
370   ISTOP4 = ISTOP1 + 1
      DO  371  I = ISTOP4,ISTOP2
      CASENO = MOM(1,I)
      LOCA = MOM(2,I)
      WRITL (6,9032) CASENO,TYPE(LOCA)
371   CONTINUE
      GO TO 1
 10   NPLUSP=ID(10) + ID(11)
      IGEN = IGEN + 1
      IF ( IGEN .GT. 25 ) GO TO 220
      XOM(1,IGEN) = CASENO
      XOM(2,IGEN) = ITAPE
      XOM(3,IGEN) = 9999999.
220   CONTINUE
      NTIM = 2 * NPLUSP + 9
      N=ID(10)
      CALL READB2 (NF,TIME,NTIM,HISTRY)
      CALL READB2 (NF,EST,NPLUSP,HISTRY)
      CALL READB2(NF,CALC,NPLUSP,HISTRY)
      CALL READB2 (NF,ACTUAL,NPLUSP,HISTRY)
      CALL READB2 (NF,BUY,N,HISTRY)
      CALL READB2 (NF,MEAN,NPLUSP,HISTRY)
```

```
      NTIM = NTIM - 9
      CALL READB2 (NF,MCONF,NTIM,HISTRY)
      CALL READB2 (NF,UW2,NPLUSP,HISTRY)
      L=0
      DO 101 I=N,NPLUSP
      L=L+1
      WT(L) = WT(L) + UW2(I)
      WTA(L) = WTA(L) + ( UW2(I) * ACTUAL(I) )
      WTC(L) = WTC(L) + ( UW2(I) * CALC(I) )
      WTE(L) = WTE(L) + ( UW2(I) * EST(I) )
      WTM(L) = WTM(L) + MEAN(I)
101   CONTINUE
      NP(ITAPE)=NP(ITAPE)+1
      GO TO  38
201   WRITE (6,10000) WT(I),WTA(I),WTC(I),WTE(I)
10000 FORMAT (1H0,5X,34HSUMS OF PERCENTS DO NOT EQUAL 100.,2X,5HWT = ,
     1        F9.2,2X,6HWTA = ,F9.2,2X,6HWTC = ,F9.2,2X,6HWTE = ,F9.2)
      WRITE (6,9104)
      GO TO 1
 9000 FORMAT (1H1,131(1H*))
 9001 FORMAT (1X,131(1H*))
 9002 FORMAT (1X,34H* PREDICTION ANALYSIS TECHNIQUE * ,6A6,   1X,1H*,
     1        20H CASE NUMBER * DATE ,6X,A6,A2,18H * REQUIRED BUYOFF,
     2        6X,1H* / 3H * ,29(1H*),3H * ,6A6,16H *  (GENERAL)  *,21X,
     3        1H*,16X,7H LB   * / 3H * ,4A6,5X,3H * ,6A6,2H *,3X,I7,3X,
     4        13H*           ,6X,2X,25H * EQUIVALENT S/C OR L/V*  /
     5        19H * COMPUTER PROGRAM,3X,A3,7X,3H * ,6A6,2H *,13X,
     6        17H*                   ,4X,12H * DEFICIT   ,6X,7H LB    * )
 9003 FORMAT (1X,18H*   CASE  * MODE *,8X,1H*,8X,10H* MAXIMUM*,13X,2H* ,
     1        9A6,16X,1H* / 1X,18H* NUMBERS * FOR   *,8X,10H* PREDIC *,
     2        24H LIKELI * ..PERCENT.. * ,9A6,16X,1H* / 1X,9H* IN THIS ,
     3        9H * THIS *,8X, 19H*  TION   *  HOOD   *,13X,1H*,3X,8(A4,4X)
     4       ,4X,1H* )
 9004 FORMAT (1X,34H* PREDICTION ANALYSIS TECHNIQUE * ,6A6,   1X,1H*,
     1        20H CASE NUMBER * DATE ,6X,A6,A2,18H * REQUIRED BUYOFF,
     2        6X,1H* / 3H * ,29(1H*),3H * ,6A6,16H *  (GENERAL)  *,21X,
     3        1H*,16X,7H LB   * / 3H * ,4A6,5X,3H * ,6A6,2H *,3X,1H0,I6,
     4        3X,13H*           ,6X,2X,25H * EQUIVALENT S/C OR L/V*  /
     5        19H * COMPUTER PROGRAM,3X,A3,7X,3H * ,6A6,2H *,13X,
     6        17H*                   ,4X,12H * DEFICIT   ,6X,7H LB    * )
 9013 FORMAT (1H1)
 9032 FORMAT (1X,I9,3X,A3,4X,A3,1X,I2,2(2X,I7),3X,3(I3,1X),1X,8(1X,I7))
 9037 FORMAT (20X,           A3,1X,I2,2(2X,I7),3X,3(I3,1X),1X,8(1X,I7))
 9104 FORMAT (25H *** SUMMING RUN DELETED.  )
      END
```

DECISION RELEVANCY PROGRAM - 64S (Criticality)

The Decision Relevancy Program compares current and predicted values for each launch vehicle, spacecraft, and the corresponding stages and modules with their respective control limits. The resulting ratios are weighted against predetermined rating scales to determine the criticality index. This index identifies any problems which may exist and emphasizes the degree of seriousness of each.

INPUT

Raw data is entered on a special-purpose input form titled "Criticality Program Coding Form", a sample of which is attached. This form is designed so as to be very nearly self-explanatory. The columns which are to be utilized for input are grouped under a heading or title which indicates the specific type of information to be entered in those columns.

A separate input sheet is used for each mission. The vehicle for each mission consists of a spacecraft with its component modules and a launch vehicle with its component stages. To complete the input form for a typical mission, determine the stages and modules which comprise the vehicle for that mission. Refer to the listing at the bottom of the input form and determine the code for each component stage or module opposite the component name. Enter the code for every stage and module of the vehicle in numerical order in columns 9 and 10 of the input form, being sure that where two stages are listed opposite one code (for instance, the S-II Stage and S-II/S-IVB interstage are both listed opposite code 3) that the code is entered twice. Opposite the appropriate code in columns 9 and 10, enter the case number, the months to shipping, and the current control and predicted weights of every stage and module in the group of columns headed for these entries. On the line directly below the last stage or module, also in numerical order of code, list the code, the case number, the months to ship, and the current control and predicted values of the launch vehicle capability (code 13), and the spacecraft weight (code 14).

In the line directly below spacecraft weight, enter code 15 (for L/V-S/C interface) in columns 9 and 10 and the largest numerical value of "months to ship" appearing for any stage or module in columns 24 and 25. Enter code 16 (for total mission) on the next line in columns 9 and 10. Finally, on the next line below code 16, enter either the word RUN or END in columns 1, 2, and 3, depending on whether another mission is to be considered or the run is to be terminated.

Any coded line may be omitted from the above if it is not to be considered, without affecting the output of the others; however, the last line on any sheet should have the RUN or END entry in columns 1-3. Any identification desired may be entered in columns 73 through 80 under the heading identification.

In addition to the above, further instructions are given on the input form itself.

One other card is needed in the input deck. This card will be the first card of the input data deck (ENTIRE RUN). The first 14 columns of this card are placed in the title format. The original intent was to include the month and year of the data run (example 1); however, it may be desirable to modify this for data distinction as shown in example 2.

Example 1:

1 - 9	11 - 14
MONTH	YEAR

Example 2:

1 - 14
MONTH YEAR RUN NO. X

Other modifications are possible as long as they stay within the first fourteen columns.

Reference No. 64.0.2
Issue Date 23 December 1965
Supersedes New

CRITICALITY PROGRAM CODING FORM

NAME
CHARGE

PAY NO. UNIT

DATE

PAGE 11 OF 11

IDENTIFICATION
73 AUG DATA 80

** 1 2 3	* CODE 9 10	CASE NUMBER 14 20	MONTH TO SHIP 24 25	CURRENT WEIGHT 29 35	CONTROL WEIGHT 39 45	PREDICTED WEIGHT 49 55
	1	5461500	25	3972	4156	4[illegible]34
	2	5451500	25	27285	29691	28473
	3	5440100	25	7484	7700	7665
	3	5431300	25	87[illegible]92	92139	88748
	4	5420100	25	12783	14100	13578
	4	5411200	25	358310	369018	363976
	6	0471100	23	8200	8200	8193
	7	0462500	23	10485	11000	10929
	8	0452500	23	46535	46890	49477
	9	0412500	23	9843	9940	10389
	10	0422500	23	22215	22060	24406
	12	0440500	23	3550	3800	3665
	13	5400500	25	100227	95000	98105
	14	0400500	23	92937	94000	99175
	15		23			
	16					
END						

ALL NUMBERS RIGHT-ADJUSTED IN THEIR FIELDS. NO DECIMAL POINTS ALLOWED.

** LAST CARD MUST HAVE "END" IN COLS. 1-3 TO TERMINATE RUN OR "RUN" IN COLS. 1-3 TO MAKE AN ADDITIONAL RUN.

A 1 PUNCHED IN COL. 10 OF THIS CARD WILL SUPPRESS CONDENSED OUTPUT AND A TWO PUNCHED IN COL. 10 WILL SUPPRESS EXTENDED OUTPUT. THIS APPLIES TO THE RUN IMMEDIATELY PRECEDING THIS "END" OR "RUN" CARD.

* CODE	COMPONENT NAME	CODE	COMPONENT NAME
1	INSTRUMENTATION UNIT	9	LEM ASCENT STAGE
2	S-IVB STAGE	10	LEM DESCENT STAGE
3	S-II STAGE + S-II/S-IVB INTERSTAGE	11	LEM GROSS
4	S-IC STAGE + S-IC/S-II INTERSTAGE	12	ADAPTER
5	S-IB STAGE + S-IB/S-IVB INTERSTAGE	13	LAUNCH VEHICLE CAPABILITY
6	LAUNCH ESCAPE SYSTEM (TOTAL)	14	SPACECRAFT WEIGHT
7	COMMAND MODULE (GROSS)	15	L/V-S/C INTERFACE
8	SERVICE MODULE (GROSS)	16	TOTAL MISSION

***** CRITICALITY DETERMINATION PROGRAM *****   204 MISSION   SEPTEMBER 1965

COMPONENT	NO. TO SHIP	REPORTED (OR CALC.) POUNDS CURRENT	CONTROL	PREDICT	R A T I O CURR.	PRED.	COND. CURR.	CODE PRED.	TIME FACTOR CURR.	PRED.	TIME WEIGHTED CURR.	PRED.	COND. SUM	OVERALL RATIO	COND.
LAUNCH VEHICLE STAGES															
****** ******* ******															
*CASE NO.		2461100													
I. U.	10	4426	4650	4504	1,051	1,032	4	4	0,72	0,28	2,88	1.12	4.00	1.00	G
*CASE NO.		2451400													
S-IVB	10	25390	26077	25756	1,027	1.012	4	4	0,72	0,28	2,88	1.12	4,00	1.00	G
*CASE NO.		2420100	2411200												
S-IB/SUM	10	102723	107500	102567	1,047	1,048	4	4	0,72	0,28	2,88	1.12	4,00	1.00	G
SPACECRAFT MODULES															
********** *******															
*CASE NO.		7471100													
LES	8	8200	8200	8205	1,000	0,999	4	4	0,78	0.22	3,12	0.88	4,00	1.00	G
*CASE NO.		7461400													
CM	8	11080	11000	11361	0,993	0,968	4	3	0.78	0,22	3,12	0.66	3,78	0.95	G
*CASE NO.		7452400													
SM	8	20416	20400	20776	0,999	0,982	4	3*	0.78	0.22	3,12	0.66	3,78	0.95	G
*CASE NO.		7440400													
ADAPTER	8	3505	3900	3505	1,113	1.113	4	4	0,78	0,22	3.12	0.88	4,00	1.00	G
OVERALL MISSION															
******* *******															
*CASE NO.		2400400													
L/V CAPAB	10	36277	35300	35856	1,028	1,016	G	G	0,72	0.28	2.88	1.12	4,00	1.00	G
*CASE NO.		7400400													
S/C WGT.	8	35001	35300	35642	1,009	0,990	G	MI*	0.78	0,22	3,12	0.66	3,78	0.95	G
INTERFACE	10				1,036	1.006	4	4	0,72	0,28	2,88	1.12	4.00	1.00	4
TOTAL											8.88	2.90	11,78	0.98	G

***** CRITICALITY DETERMINATION PROGRAM ***** 204 MISSION SEPTEMBER 1965

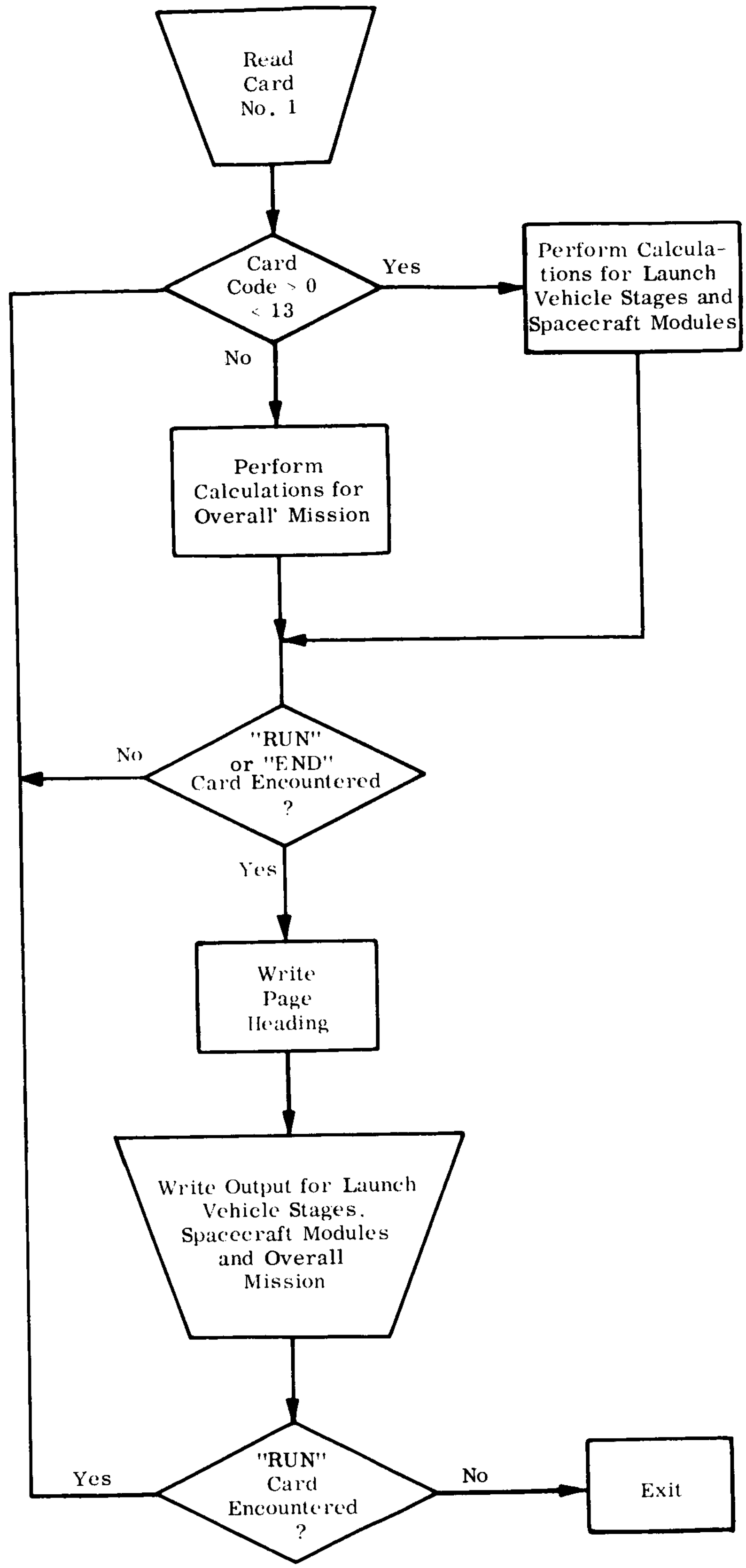
CONDENSED RATING TABLE

COMPONENT	CASE NUMBER		CRITICAL	WEAKNESS MAJOR	WEAKNESS MINOR	GOOD SHAPE
LAUNCH VEHICLE STAGES						
INSTRUMENTATION UNIT	2461100					*
S-IVB STAGE	2451400					*
S-IB STAGE + S=IB/S-IVB INTERSTAGE	2420100	2411200				*
SPACECRAFT MODULES						
LAUNCH ESCAPE SYSTEM (TOTAL)	7471100					*
COMMAND MODULE (GROSS)	7461400					*
SERVICE MODULE (GROSS)	7452400					*
ADAPTER	7440400					*
OVERALL MISSION						
LAUNCH VEHICLE CAPABILITY	2400400					*
SPACECRAFT WEIGHT	7400400					*
L/V-S/C INTERFACE						*
TOTAL MISSION						*

JOB DECK STRUCTURE - The Criticality Program is run as a separate job independent of SPACE. The job deck setup for processing the Criticality Program appears below:

CARD COLUMNS

```
1-------8-------16--------------------------------------------72
$ID                CHARGE NO., NAME, PAY NO., UNIT NO., PROGRAM, MODE
$PAUSE             (MOUNT RELOAD TAPE)
$OPEN
$IBJOB    CRITIC   MAP, DLOGIC
$RELOAD            I06, NAME = CRITIC, SRCH

                   (INPUT DATA DECK)

$IBSYS
$CLOSE
$IBSYS
```


Read
Card
No. 1
Card
Code > 0
< 13
Yes
No
Perform Calcula-
tions for Launch
Vehicle Stages and
Spacecraft Modules
Perform
Calculations for
Overall Mission
"RUN"
or "END"
Card Encountered
?
No
Yes
Write
Page
Heading
Write Output for Launch
Vehicle Stages,
Spacecraft Modules
and Overall
Mission
"RUN"
Card
Encountered
?
Yes
No
Exit

```
$IBJOB CRITIC  MAP,LOGIC
$IBFTC CRITIC  LIST,REF
C
C *** CRITICALITY DETERMINATION PROGRAM ***
C
C
      COMMON   /GGDATE/DATE(2)
C
      INTEGER  CASNO,CASNU,CASN1,CASN2,CASN3
C
      DIMENSION NXASP(2,17)
      DIMENSION  TABLEE(36),TABLEG(36),TUBLE(8),TVBLE(8),TABLE(17,17),
     1           KTUB(4),NAST(4),NOB(4),NAM(96),INDEX(16),NEM(32),
     2           JNDEX(17),TWBLE(8),TXBLE(8)
C
C *** TABLES
C
      DATA  BLANK     /060606060606O/
      DATA  END       /3HEND/
      DATA  RUN       /3HRUN/
      DATA  INDEX     /1,7,13,19,25,31,37,43,49,55,61,67,73,79,85,91/
      DATA  JNDEX     /1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33/
      DATA  KTUB      /2H C,2HMA,2HMI,2H G/
      DATA  NEM       /3HI. ,6HU.      ,3HS-I,6HVB     ,
     1                 3HS-I,6HI/SUM ,3HS-I,6HC/SUM ,
     2                 3HS-I,6HB/SUM ,3HLES,6H      ,
     3                 3HCM ,6H      ,3HSM ,6H      ,
     4                 3HLEM,6H-A    ,3HLEM,6H-D    ,
     5                 3HLEM,6H GROSS,3HADA,6HPTER  ,
     6                 3HL/V,6H CAPAB,3HS/C,6H WGT. ,
     7                 3HINT,6HERFACE,3HTOT,6HAL     /
      DATA  NAST      /6H******,3H***,3H***,1H*/
      DATA  NBLNK     /060606060606O/
      DATA  TUBLE     /0.0000,0.9200,0.9245,0.9600,0.9645,0.9800,0.9845,
     1                 100.0/
      DATA  TVBLE     /0.000,0.640,0.655,0.790,0.805,0.890,0.905,1.001/
      DATA  TABLEE    /1.00,0.99,0.98,0.96,0.94,0.89,0.82,0.78,0.75,0.72,
     1                 0.70,0.68,0.66,0.64,0.62,0.60,0.59,0.58,0.56,0.55,
     2                 0.54,0.53,0.52,0.51,0.50,0.49,0.48,0.47,0.46,0.45,
     3                 0.44,0.43,0.42,0.41,0.40,0.40/
      DATA  TABLEG    /0.00,0.01,0.02,0.04,0.06,0.11,0.18,0.22,0.25,0.28,
     1                 0.30,0.32,0.34,0.36,0.38,0.40,0.41,0.42,0.44,0.45,
     2                 0.46,0.47,0.48,0.49,0.50,0.51,0.52,0.53,0.54,0.55,
     3                 0.56,0.57,0.58,0.59,0.60,0.60/
      DATA  TWBLE     /0.0000,0.9500,0.9545,0.9800,0.9845,0.9900,0.9945,
     1                 100.0/
      DATA  TXBLE     /0.0000,0.9500,0.9545,0.9800,0.9845,0.9985,0.9995,
     1                 100.0/
      DATA  NAM       /48HINSTRUMENTATION UNIT                     S-IVB STAGE
     1  ,60H                            S-II STAGE + S-II/S-IVB INTERSTAGE
     2  ,60HS-IC STAGE + S-IC/S-II INTERSTAGE    S-IB STAGE + S-IB/S-IVB
     3  ,60HINTERSTAGE  LAUNCH ESCAPE SYSTEM (TOTAL)          COMMAND MODU
```

```
     4   ,60HLE (GROSS)                        SERVICE MODULE (GROSS)
     5   ,60HLEM ASCENT STAGE                            LEM DESCENT STAGE
     6   ,60H               LEM GROSS                              ADAPTER
     7   ,60H                            LAUNCH VEHICLE CAPABILITY
     8   ,60HSPACECRAFT WEIGHT                           L/V-S/C INTERFACE
     9   ,48H               TOTAL MISSION                          /
C
C *** TABLE A IS NAMED TUBLE
C *** TABLE B IS NAMED TWBLE
C *** TABLE C IS NAMED TXBLE
C *** TABLE D IS NAMED TVBLE
C *** TABLE E IS NAMED TABLEE AND TABLEG
C
C *** BEGIN COMPUTATION
C
      READ (5,10000) MONTH1,MONTH2,YEAR
10000 FORMAT (A6,A4,A4)
      IONE = 1
      ITWO = 2
   24 CASN1 = 0
      CASN2 = 0
      CASN3 = 0
      DO 23 II=1,17
      DO 22 JJ=1,17
      TABLE(II,JJ) = 0.0
   22 CONTINUE
   23 CONTINUE
   25 READ (5,1000) AUX,NAME,CASNO,MO,CURR,CONT,PRED
      LOOP = 0
C
      IF (AUX .EQ. RUN)  GO TO 200
      IF (AUX .EQ. END)  GO TO 200
      CONTI = CONT
      TABLE(1,NAME) = NAME
      TABLE(2,NAME) = MO
      IF (NAME .NE. INAME)  GO TO 55
      IF (NAME .NE. 3)  GO TO 42
      CASN1 = CASNO
      GO TO 46
   42 IF (NAME .NE. 4)  GO TO 44
      CASN2 = CASNO
      GO TO 46
   44 IF (NAME .NE. 5)  GO TO 55
      CASN3 = CASNO
   46 CONTINUE
      TABLE(3,NAME) = TABLE(3,NAME) + CURR
      TABLE(4,NAME) = TABLE(4,NAME) + CONT
      TABLE(5,NAME) = TABLE(5,NAME) + PRED
      CURR = TABLE(3,NAME)
      CONT = TABLE(4,NAME)
      PRED = TABLE(5,NAME)
      CONTI = CONT
      LOOP = 1
```

```
      GO TO 65
   55 TABLE(3,NAME) = CURR
      TABLE(4,NAME) = CONT
      TABLE(5,NAME) = PRED
   65 INAME = NAME
      IF (CURR .EQ. 0.0 .OR. CONT .EQ. 0.0 .OR. PRED .EQ. 0.0) GO TO 70
C
      IF (NAME .NE. 13) GO TO 70
      TEMP = CONT
      CONT = CURR
      CURR = TEMP
      TEMP = CONTI
      CONTI = PRED
      PRED = TEMP
   70 IF (NAME .NE. 15)  GO TO 75
      TABLE(6,15) =  TABLE(3,13) / TABLE(3,14)
      TABLE(7,15) =  TABLE(5,13) / TABLE(5,14)
      GO TO 80
C
   75 IF (CURR .EQ. 0.0 .OR. CONT .EQ. 0.0 .OR. PRED .EQ. 0.0) GO TO 111
      TABLE(6,NAME) = CONT / CURR
      TABLE(7,NAME) = CONTI/ PRED
C
   80 J = 6
   95 DO 100 I=1,7
      IF (NAME .EQ. 13)  GO TO 96
      IF (NAME .EQ. 14)  GO TO 96
      GO TO 97
   96 IF (TABLE(J,NAME) .GT. TWBLE(I) .AND. TABLE(J,NAME) .LT. TWBLE
     1   (I+1))  GO TO 105
      GO TO 100
   97 IF (NAME .NE. 15) GO TO 98
      IF (TABLE(J,NAME) .GT. TXBLE(I) .AND. TABLE(J,NAME) .LT. TXBLE
     1   (I+1))  GO TO 105
      GO TO 100
   98 IF (TABLE(J,NAME) .GE. TUBLE(I) .AND. TABLE(J,NAME) .LT. TUBLE
     1   (I+1)) GO TO 105
  100 CONTINUE
105   CONTINUE
      IMO = ( I + 1 ) / 2
      NXASP(J-5,NAME) = NBLNK
      IF  ( MOD(I,2) .EQ. 0 ) NXASP(J-5,NAME) = NAST(4)
      TABLE(J+2,NAME) = IMO
      IF (J .EQ. 7)  GO TO 110
      J = 7
      GO TO 95
C
  110 CONTINUE
  111 IF (IO .EQ. 0)    MO = 1
      TABLE(10,NAME) = TABLEE(MO)
      TABLE(11,NAME) = TABLEG(MO)
  112 ICBLE = TABLE(1,NAME) + 0.5
      IF (ICBLE .NE. 16)  GO TO 140
```

```
      TABLE(12,NAME) = TABLE(12,NAME-3) + TABLE(12,NAME-2) + TABLE
     1                 (12,NAME-1)
      TABLE(13,NAME) = TABLE(13,NAME-3) + TABLE(13,NAME-2) + TABLE
     1                 (13,NAME-1)
      TABLE(14,NAME) = TABLE(14,NAME-3) + TABLE(14,NAME-2) + TABLE
     1                 (14,NAME-1)
      TABLE(15,NAME) = TABLE(14,NAME) / 12.0
      GO TO 145
C
  140 IF (MO .EQ. 0)  GO TO 160
      TABLE(12,NAME) = TABLE(8,NAME) * TABLE(10,NAME)
      TABLE(13,NAME) = TABLE(9,NAME) * TABLE(11,NAME)
      TABLE(14,NAME) = TABLE(12,NAME) + TABLE(13,NAME)
      TABLE(15,NAME) = TABLE(14,NAME) / 4.0
  145 DO 150 L=1,7
      IF (TABLE(15,NAME) .GE. TVBLE(L) .AND. TABLE(15,NAME) .LT.
     1    TVBLE(L+1))  GO TO 155
  150 CONTINUE
155   IMMO = L
      IMNO = NBLNK
      IF  ( MOD(IMMO,2) .EQ. 0 ) IMNO = NAST(4)
      IMMO = ( IMMO + 1 ) / 2
      TABLE(16,NAME) = ( L + 1 ) / 2
  160 CONTINUE
      IF (LOOP .EQ. 1)  GO TO 25
      TABLE(17,NAME) = CASNO
      GO TO 25
C
  200 DO 215 LOZ=1,5
      IF (TABLE(17,LOZ) .EQ. 0.0)  GO TO 215
      IB = TABLE(17,LOZ) + 0.5
      GO TO 217
  215 CONTINUE
  217 IA = 0
      CALL FLIP (IA,IB,IONE)
      IRC = IA * 100
      IA = 0
      CALL FLIP (IA,IB,ITWO)
      IRC = IRC + IA
      IF (NAME .EQ. 2)  GO TO 450
C
C *** ROUTINE FOR FULL OUTPUT
C
      WRITE (6,1004) IRC,MONTH1,MONTH2,YEAR
      WRITE (6,1005)
      WRITE (6,1011)
      DO 300 K=1,16
      ICBLE = TABLE(1,K) + 0.5
      IF (ICBLE .EQ. 0)  GO TO 290
      IABLE = TABLE(2,K) + 0.5
      JABLE = TABLE(3,K) + 0.5
      KABLE = TABLE(4,K) + 0.5
      LABLE = TABLE(5,K) + 0.5
```

```
      IBBLE = TABLE(8,K) + 0.5
      JBBLE = TABLE(9,K) + 0.5
  246 NUT = TABLE(16,K)
      KBBLE = KTUB(NUT)
      NED = JNDEX(K)
      NOM = JNDEX(K+1) - 1
      CASNO = TABLE(17,K) + 0.5
C
C                    KIND = 1   BOTH CASNO AND CASNU HAVE 7 DIGITS
C                         = 2   CASNO HAS 6 DIGITS, CASNU HAS 7 DIGITS
C                         = 3   CASNO HAS 7 DIGITS, CASNU HAS 6 DIGITS
C                         = 4   BOTH CASNO AND CASNU HAVE 6 DIGITS
C
      KAZNO = CASNO/1000000
      IF (ICBLE .EQ. 3 .AND. CASN1 .NE. 0)  GO TO 249
      GO TO 250
  249 CASNU = CASN1
      GO TO 255
  250 IF (ICBLE .EQ. 4 .AND. CASN2 .NE. 0)  GO TO 251
      GO TO 252
  251 CASNU = CASN2
      GO TO 255
  252 IF (ICBLE .EQ. 5 .AND. CASN3 .NE. 0)  GO TO 253
      GO TO 254
  253 CASNU = CASN3
      GO TO 255
  254 IF (ICBLE .EQ. 15 .OR. ICBLE .EQ. 16)  GO TO 256
      IF (KAZNO .EQ. 0)   GO TO 1254
      WRITE (6,1003) CASNO
      GO TO 256
 1254 WRITE (6,1002) CASNO
      GO TO 256
  255 KIND = 1
      IF (KAZNO .EQ. 0)   KIND = KIND + 1
      KAZNO = CASNU/1000000
      IF (KAZNO .EQ. 0)   KIND = KIND + 2
      GO TO (1255,1256,1257,1258),KIND
 1255 WRITE (6,1003) CASNO,CASNU
      GO TO 256
 1256 WRITE (6,1001) CASNO,CASNU
      GO TO 256
 1257 WRITE (6,1007) CASNO,CASNU
      GO TO 256
 1258 WRITE (6,1002) CASNO,CASNU
  256 CONTINUE
      IF (ICBLE .EQ. 13 .OR. ICBLE .EQ. 14)  GO TO 265
      IF (ICBLE .EQ. 15 .OR. ICBLE .EQ. 16)  GO TO 258
      WRITE (6,1006) (NEM(IB),IB=NED,NOM),IABLE,JABLE,KABLE,LABLE,
     1               TABLE(6,K),TABLE(7,K),IBBLE,NXASP(1,K),JBBLE,
     2               NXASP(2,K),(TABLE(IX,K),IX=10,15),KBBLE,IMNO
      GO TO 290
  258 IF (ICBLE .EQ. 16)  GO TO 262
      WRITE (6,1027) (NEM(IB),IB=NED,NOM),IABLE,TABLE(6,K),TABLE(7,K),
```

```
    1                    IBBLE,NXASP(1,K),JBBLE,NXASP(2,K),(TABLE(IX,K),
    2                    IX=10,15),IMMO,IMNO
      GO TO 290
  262 WRITE (6,1028) (NEM(IB),IB=NED,NOM),(TABLE(IX,K),IX=12,15),KBBLE,
    1                    IMNO
      GO TO 290
  265 IBBLI = NBLNK
      JGGLE = NBLNK
      DO 270 MAW=1,4
      IBCD = TABLE(8,K) + 0.5
      JBCD = TABLE(9,K) + 0.5
      IF (IBCD .NE. MAW)  GO TO 267
      IGGLE = KTUB(MAW)
  267 IF (JBCD .NE. MAW)  GO TO 270
      JGGLE = KTUB(MAW)
  270 CONTINUE
      WRITE (6,1024) (NEM(IB),IB=NED,NOM),IABLE,JABLE,KABLE,LABLE,
    1                    TABLE(6,K),TABLE(7,K),IGGLE,NXASP(1,K),JGGLE,
    2                    NXASP(2,K),(TABLE(IX,K),IX=10,15),KBBLE,IMNO
  290 IF (K .EQ. 5)  GO TO 294
      GO TO 295
  294 WRITE (6,1012)
  295 IF (K .EQ. 12)  GO TO 298
      GO TO 300
  298 WRITE (6,1013)
  300 CONTINUE
C
C *** ROUTINE FOR CONDENSED OUTPUT
C
  450 IF (NAME .EQ. 1)  GO TO 775
      WRITE (6,1004) IRC,MONTH1,MONTH2,YEAR
      WRITE (6,1009)
      WRITE (6,1011)
      DO 750 MA=1,16
      ICBLE = TABLE(1,MA) + 0.5
      IF (ICBLE .EQ. 0)  GO TO 715
      LUM = INDEX(MA)
      LUN = LUM + 5
      NIB = TABLE(16,MA) + 0.5
      DO 495 IR=1,4
  495 NOB(IR) = NBLNK
      NOB(NIB) = NAST(NIB)
      IF (ICBLE .EQ. 3 .AND. CASN1 .NE. 0)  GO TO 509
      GO TO 510
  509 CASNU = CASN1
      GO TO 518
  510 IF (ICBLE .EQ. 4 .AND. CASN2 .NE. 0)  GO TO 511
      GO TO 512
  511 CASNU = CASN2
      GO TO 518
  512 IF (ICBLE .EQ. 5 .AND. CASN3 .NE. 0)  GO TO 513
      GO TO 514
  513 CASNU = CASN3
```

```
      GO TO 518
  514 CONTINUE
  518 CASNO = TABLE(17,MA) + 0.5
      IF (ICBLE .EQ. 3 .OR. ICBLE .EQ. 4 .OR. ICBLE .EQ. 5)  GO TO 610
      IF (ICBLE .EQ. 15 .OR. ICBLE .EQ. 16)  GO TO 620
      WRITE (6,1025)  (NAM(MG),MG=LUM,LUN),CASNO,(NOB(MH),MH=1,4)
      GO TO 625
  610 WRITE (6,1015)  (NAM(MG),MG=LUM,LUN),CASNO,CASNO,(NOB(MH),MH=1,4)
      GO TO 625
  620 WRITE (6,1026)  (NAM(MG),MG=LUM,LUN),(NOB(MH),MH=1,4)
  625 CONTINUE
  715 IF (MA .EQ. 5)  GO TO 724
      GO TO 725
  724 WRITE (6,1012)
  725 IF (MA .EQ. 12)  GO TO 740
      GO TO 750
  740 WRITE (6,1013)
  750 CONTINUE
  775 IF (/UX .EQ. RUN)  GO TO 24
      CALL EXIT
C
C
 1000 FORMAT (A3,I7,3X,I7,3X,I2,3(3X,F7.0))
 1001 FORMAT (4X,9H*CASE NO.  ,3X,1H0,I6,3X,I7)
 1002 FORMAT (4X,9H*CASE NO.  ,2(3X,1H0,I6))
 1003 FORMAT (4X,9H*CASE NO.  ,2(3X,I7))
 1004 FORMAT (1H1,46H ***** CRITICALITY DETERMINATION PROGRAM *****
     1         10X,I3,8H MISSION,20X,A6,2A4)
 1005 FORMAT (1H0,12X,3HMO. / 13X,2HTO,5X,26HREPORTED (OR CALC.) POUNDS,
     1         3X,9HR A T I O,5X,10HCOND. CODE,3X,11HTIME FACTOR,3X,
     2         19HTIME WEIGHTED COND.,4X,7HOVERALL /
     3         1X,9HCOMPONENT,2X,4HSHIP,3X,7HCURRENT,3X,7HCONTROL
     4         ,3X,7HPREDICT,2X,5HCURR.,2X,5HPRED.,2X,5HCURR.,2X,
     5         5HPRED.,2X,5HCURR.,2X,5HPRED.,2X,5HCURR.,2X,5HPRED.,4X,
     6         3HSUM,2X,5HRATIO,2X,5HCOND. /  )
1006  FORMAT (1X,A3,A6,3X,I2,1X,3(3X,I7),2(1X,F6.3),2(2X,I4,A1),
     1         6(2X,F5.2),2X,A3,A1 / )
 1007 FORMAT (4X,9H*CASE NO.  ,3X,I7,3X,1H0,I6)
 1009 FORMAT (1H0,10X,22HCONDENSED RATING TABLE // 14X,9HCOMPONENT,
     1         24X,11HCASE NUMBER,19X,11HR A T I N G // 68X,8HCRITICAL,
     2         22H  --WEAKNESS--   GOOD / 78X,19HMAJOR  MINOR  SHAPE // )
 1011 FORMAT (22H LAUNCH VEHICLE STAGES / 22H ****** ******* ****** / )
 1012 FORMAT ( / 20H SPACECRAFT MODULES / 20H ********** ******* / )
 1013 FORMAT ( / 16H OVERALL MISSION / 16H ******* ******* / )
 1015 FORMAT (1X,6A6,6X,2(I7,6X),A6,4X,A3,4X,A3,5X,A1 / )
1024  FORMAT (1X,A3,A6,3X,I2,1X,3(3X,I7),2(1X,F6.3),2(4X,A2,A1),
     1         6(2X,F5.2),2X,A3,A1 / )
 1025 FORMAT (1X,6A6,6X,I7,19X,A6,4X,A3,4X,A3,5X,A1 / )
 1026 FORMAT (1X,6A6,32X,A6,4X,A3,4X,A3,5X,A1 / )
1027  FORMAT (1X,A3,A6,3X,I2,31X,2(1X,F6.3),2(2X,I4,A1),6(2X,F5.2),2X,
     1         I2,A1 /)
 1028 FORMAT (1X,A3,A6,78X,4(1X,F6.2),2X,A3,A1 / )
      END
```

Reference No. 65.0
Issue Date 13 August 1965
Supersedes New

OVERALL COST ESTIMATION - 65S

This program calculates the overall cost required to accomplish a given buyoff and is based on the following assumptions:

ASSUMPTIONS

1. Only overall costs on the stage and module levels were considered. An analysis based on the functional-systems level will be performed in the future as a refinement of this analysis.
2. A ten percent redesign yield was assumed, i.e., 1000 pounds of inert weight must be redesigned and refabricated, where required, in order to obtain a 100-pound weight reduction.
3. R&D dollars vary exponentially with weight reduction, i.e., the first 100 pounds of weight reduction costs less than the next 100 pounds, etc. The exponential equation used is

$$D = aW^b$$

where:

D = required R&D dollars

W = weight reduction required

a,b = constants derived for each stage/module from the available cost data.

4. Production costs for a change are equal to the dollars spent to date on the affected portion of the vehicle. The affected portion of the vehicle is the 1000 pounds, for example, which is being redesigned in order to obtain a 100-pound weight reduction.
5. R&D costs are assigned to the first vehicle for which a required buyoff occurs, even though the effects will be realized on subsequent vehicles. For example, if a 100-pound buyoff is required for the S-IC-501 and a 300-pound buyoff is required for the S-IC-502, the S-IC-501 will be charged with the cost of the first 100-pound buyoff while the S-IC-502 will be charged with only a 200-pound buyoff. The costs are determined by first calculating the cost of the 100-pound buyoff using the equation in assumption 3. This cost of the buyoff is charged to the S-IC-501. Using the same equation, the cost of a 300-pound buyoff is

calculated. The cost of the 200-pound buyoff assigned to the S-IC-502 is determined by subtracting the cost of the 100-pound buyoff from the cost of the 300-pound buyoff. This type of calculation is required since we are assuming an exponential cost distribution.

6. The predicted schedule slip is equal to the absolute buyoff cost divided by the current spending rate of the given stage/module. The absolute cost is the cost of a given buyoff, not considering the cumulative effects of previous buyoffs. In the previous example, the cost of the 300-pound buyoff is the absolute cost for the S-IC-502. It is this cost that is divided by the spending rate in order to obtain the desired schedule slip.

COMPUTER PROGRAM

Based on the above assumptions, a computer program was developed to perform the required calculations. Referring to the computer output sheet columns (see Reference Number 65.3), counting from the left, the program requires the following data and performs the following calculation:

1. Stage/module.
2. Total buyoff required, in pounds (obtained from prediction analysis).
3. Cumulative required weight reduction (column 2 minus the greatest previous number in column 2, if negative, enter zero).
4. a (input constant for each stage/module, taken as the production cost per pound).
5. b (input constant, i.e., the exponent, for each stage/module, which is the slope of the Cost versus Weight Reduction line, when it is plotted on log-log graph paper).
6. Total required R&D dollars, in millions, (column 2 raised to the column 5 power and then multiplied by column 4, i.e., $D = aW^b$ from assumption 3).
7. Cumulative R&D dollars, in millions, (column 6 minus the greatest previous number in column 6, if negative, enter zero).
8. Number of production months (the number of months that a given stage/module has been in the production phase. This data is obtained from the SARP charts).
9. Production dollars/pound-vehicle-month (the production cost of one pound of a given stage/module for one month).
10. Percent yield (this is the percent weight reduction obtainable from a given redesign, assumed to be ten percent for this study).
11. Production weight requiring redesign, in pounds, (100 divided by column 10 and then multiplied by column 2).

12. Required production dollars in millions, (column 8 times column 9 times column 11).
13. Total cost (gross), in millions, (column 6 plus column 12).
14. Total cumulative cost, in millions, (column 7 plus column 12).
15. Tradeoff factor in dollars/pound (column 13 divided by column 2).
16. Current spending rate in dollars/month.
17. Schedule slip in months (column 13 divided by column 16).

Reference No.	65.2.1
Issue Date	23 Dec 1965
Supersedes	New

JOB DECK STRUCTURE - The Cost Program is run as a separate job independent of SPACE. The job deck setup for processing the Cost Program appears below:

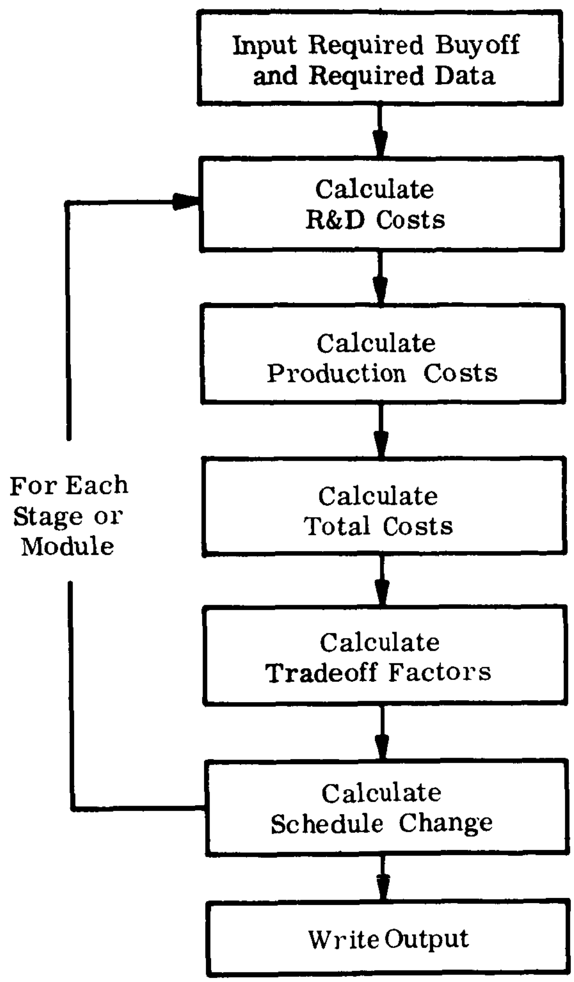
CARD COLUMNS

```
1-------8-------16--------------------------------------------------------72
$ID             CHARGE NO., NAME, PAY NO., UNIT NO., PROGRAM, MODE
$PAUSE          (MOUNT RELOAD TAPE)
$OPEN
$IBJOB  COST    MAP, DLOGIC
$RELOAD         I06, NAME = COST, SRCH

                (INPUT DATA DECK)

$IBSYS
$CLOSE
$IBSYS
```

SAMPLE CASE OF OUTPUT SHEET

OVERALL COST ESTIMATION - SATURN V LAUNCH VEHICLE

1 Stage/ Module	2 Total Reqd Buyoff, Pounds	3 Cum Reqd Wt Red, Pounds Δ2	4 a	5 b	6 Total Reqd, Million Dollars 4×2^5	7 Cum R&D Million Dollars Δ6	8 No. of Prod Months	9 Prod, Dollars/ Pounds-Veh-Month	10 Yield, Per-cent	11 Prod Wt Reqd Redesign, Pounds $\frac{(100)}{10} \times 2$	12 Reqd Prod, Million Dollars 8 x 9 x 11	13 Total Cost, Million Dollars 6 + 12	14 Total Cum Cost, Million Dollars 7 + 12	15 Tradeoff Dollars/ Pounds $\frac{13}{2}$	16 Current Spending Rate, Million Dollars/ Month	17 Sch Slip, Month $\frac{13}{16}$
SIC-501																
SIC-502																
SIC-503																
SIC-504																
SIC-505																
SIC-506																
SII-501																
SII-502																
SII-503																
SII-504																
SII-505																
SIVB-501																
SIVB-502																
SIVB-503																
SIVB-504																
SIVB-505																
SIVB-506																
IU-501																
IU-502																
IU-503																
IU-504																
IU-505																
IU-506																


Input Required Buyoff and Required Data
Calculate R&D Costs
Calculate Production Costs
Calculate Total Costs
Calculate Tradeoff Factors
Calculate Schedule Change
Write Output
For Each Stage or Module

DATE ______ **OVERALL COST ESTIMATION PROGRAM - INPUT DATA** ______ OF ______

STAGE/ MODULE- MISSION		TOTAL REQ'D B.O. (LBS.)	a	b	NO. OF PRODUCTION MONTHS	PROD. $/ LB.-VEH. MONTH	% YIELD	CURRENT SPEND. RATE MILLIONS/MO
1–8	9–24	25–32	33–40	41–48	49–56	57–64	65–72	73–80
								(Comment Line)
					(Fill in Column 7 Through 24 for Page Heading Desired)			
PAGE								
								(Cross Out if More Data Follows)
END								
1–8	9–24	25–32	33–40	41–48	49–56	57–64	65–72	73–80

```
$IBFTC 65S     LIST,REF
C COMMON DEFINITIONS
C
C DIMENSIONS
      DIMENSION  IHED1 (12)
      DIMENSION INPUT (11),FINPUT (7),FOPUT(8),INPUS(4)
C EQUATE THE FOLLOWING
      EQUIVALENCE (INPUT(5),FINPUT(1))
C
      READ  (5,5003) IHED1
 5003 FORMAT (12A6)
      AKARN = 100.
      AKON = .000001
      AKORN = 1000000.
C AT 100 READ A CARD
C COLUMNS  1 THRU 24 , BCD
C COLUMNS 25 THRU 66 , FLOATING POINT, FIELD WIDTH 6, DEC. PLACES 2
100   READ (5,1001) INPUT
1001  FORMAT  (4A6,2F8.2,F8.3,4F8.2)
C DATA STATEMENT
      DATA IEND, IPAGE,  INODAT,IMASK,IDEL,INODEL
     1   /3HEND,4HPAGE,6HNO DAT,077777000000,3HDEL,6HNO DEL/
C AT 110, DID CARD CONTAIN END
110   IF (INPUT(1).EQ.IEND) GO TO 2001
C AT 120, DID CARD CONTAIN PAGE
120   IF (INPUT(1).EQ.IPAGE) GO TO 125
C AT 122 IS NAME IN SAME CATEGORY AS PREVIOUS
122   IX = AND(INPUT(1),IMASK)
      IF (IX.EQ.IPREVN)  GO  TO  130
      WRITE (6,1007)
1007  FORMAT (15X)
      ILCNT = ILCNT -1
      FPREV1 = 0.0
      FPREV5 = 0.0
C AT 130, DO COLUMNS 19 THRU 25 STATE NO DATA
130   IF (INPUT(4).EQ.INODAT) GO TO 300
C AT 150, IS (1) GREATER THAN PREVIOUS (1)
150   IF (FINPUT(1).GT.FPREV1) GO TO 160
C AT 180, SET (2) OR FOPUT(1) = 0
180   FOPUT (1) = 0.0
C AT 190, CALCULATE (5) = (3)X(1)**(4)
190   FOPUT(2) = AKON*FINPUT(2)*FINPUT(1)**FINPUT(3)
      GO TO 200
C AT 125 SET PREVIOUS VALUES TO ZERO
125   FPREV1 = 0.0
      FPREV5 = 0.0
      INPUS(2) =INPUT(2)
      INPUS(3) =INPUT(3)
      INPUS(4) =INPUT(4)
      GO  TO  370
C AT 160 (2) =(1) - PREVIOUS(1)
160   FOPUT(1)= FINPUT(1)-FPREV1
C AT 170, SET PREVIOUS(1) = THIS (1)
170   FPREV1 = FINPUT(1)
      GO TO 190
```

```
C END FLOW CHART, PAGE 1
C
C AT 200, IS (5) GREATER THAN PREVIOUS (5)
200    IF (FOPUT(2).GT.FPREV5) GO TO 220
C AT 210, SET(6) = 0.0
210    FOPUT(3) =0.0
       GO TO 236
C AT 220, (6) = (5)-PREVIOUS(5)
220    FOPUT(3)= FOPUT(2) -FPREV5
C AT 230,SET PREVIOUS(5)= THIS ONE(5)
230    FPREV5 = FOPUT(2)
C AT 236 CALCULATE (10),(11),(12),(13),(15)
236    FOPUT(4) = AKARN*FINPUT(1)/FINPUT(6)
       FOPUT(5) = AKON*FINPUT(4)*FINPUT(5)*FOPUT(4)
       FOPUT(6) = FOPUT(2)+FOPUT(5)
       FOPUT(7) = FOPUT(3)+FOPUT(5)
       FOPUT(8) = FOPUT(6)/FINPUT(7)
C END MAIN CALCULATION BLOCK
C AT 242, IS (15) LESS THAN 0.0
242    IF (FOPUT(8).LT.0.0) GO TO 260
C AT 250, IS (15) EQUAL TO 0.0
250    IF (FOPUT(8).EQ.0.0) GO TO 260
C AT 270, SET  FMCNT = 1
270    FMCNT =1.0
C AT 278, IS (15) GREATER THAN FMCNT
278    IF(FOPUT(8).GT.FMCNT) GO TO 284
C AT 290, SET(15)=FMCNT
290    FOPUT(8) = FMCNT
295    FTRADE   = AKORN*FOPUT(6)/FINPUT(1)
       GO  TO  310
C AT 284, INCREASE FMCNT
284    FMCNT = FMCNT + 1.0
       GO TO 278
C AT 260,SET (15) = 0.0
260    FOPUT(8) = 0.0
       GO  TO  295
C
C END FLOW CHART, PAGE 2
C
C BEGIN FLOW CHART, PAGE 3
C AT 300 PRINT LINE, ALL PARAMETERS BLANK
300    WRITE (6,1002) INPUT(1),INPUT(2)
1002   FORMAT (1X,2A6)
       GO TO 320
C AT 310 PRINT LINE WITH ALL PARAMETERS
310    WRITE (6,1003) INPUT(1),INPUT(2),FINPUT(1),FOPUT(1),FINPUT(2),
      1FINPUT(3),FOPUT(2),FOPUT(3),FINPUT(4),FINPUT(5),FINPUT(6),FOPUT(4)
      2,FOPUT(5),FOPUT(6),FOPUT(7),
      3FTRADE,                          FINPUT(7),FOPUT(8)
1003   FORMAT (1X,1A6,1A2,    F7.0,F7.0    ,    F6.0,    F6.3,1X,F8.1,
      11X,F8.1,1X,F6.0,1X,F7.0,1X,F5.0,2X,F7.0,1X,F7.1,1X,F7.1,F9.1,
      2    F9.1,1X,F7.1,1X,F6.0)
C AT 320, REDUCE LINE COUNT
320    ILCNT = ILCNT -1
```

```
C AT 330 TEST LINE COUNT
330   IF (ILCNT.LT.0) GO TO 370
      GO  TO  380
 370  WRITE  (6,5007)   IHED1
5007  FORMAT  (1H1,30X,12A6)
      WRITE (6,1004) INPUS(2),INPUS(3),INPUS(4)
1005  FORMAT ( 1X, 130HSTAGE/   TOTAL    CUM.     A      B    TOTAL    CUM
     1    NO. OF PROD $/ P.C.  PROD WT. REQD.     TOTAL     TOTAL    TRADE
     2 CURRENT  SCH. /
     3          1X, 130HMODULE-  REQD    REQD.              REQD. $  R+D $
     4    PROD   LB-VEH. YIELD REQD.    PROD. $    COST     CUM.     OFF
     5 SPENDING SLIP /
     6          1X  131H         B.O.    WT.                MILLIONS MILLI
     7ONS MONTHS   MO.          REDESIGN MILLIONS MILL.     COST     ($/LB)
     8   RATE    (MO.)/10X,11H(LBS)   RED,54X,5H(LBS),20X, 8HMILLIONS,
     910X,9HMILL./MO./18X,5H(LBS)/)
373   WRITE (6,1005)
1004  FORMAT (     40X, 30HREQUIRED BUYOFF COST SUMMARY     ,3A6//)
377   ILCNT = 50
C AT 380 ,  MOVE NAME TO PREVIOUS CELL
380   IPREVN = AND(INPUT(1),IMASK)
      GO TO 100
C AT 2001, RESTORE PAGE AND EXIT
2001  WRITE (6,1006)
1006  FORMAT(1H1)
      CALL EXIT
      STOP
      END
```

CARD COUNT 138

UPDATE PROGRAM - 66S

This program reads a set of data cards and performs an update on the Weight Data File (WDF). The update may consist of:

a. Additions of entire new cases to the Weight Data File.

b. Addition of new data points.

c. Corrections to previously submitted erroneous points.

Input cards for each of the three types are described as follows:

NEW CASE ADDITIONS

The addition of entire new cases consists of the following set of cards:

Card 1

Word 1 - The word RESTO punched in columns 1-5

Word 2 - Month in which the system is to be shipped (integer)

Word 3 - Year in which the system is to be shipped (integer)

Word 4 - Month in which first data point was observed (integer)

Word 5 - Year in which first data point was observed (integer)

Word 6 - Parameter r_1 (floating point)

Word 7 - Parameter r_2 (floating point)

Word 8 - Actual percentage of weight at shipping date (floating point)

Word 9 - Calculated percentage of weight at shipping date (floating point)

Word 10 - Estimated percentage of weight at shipping date (floating point)

Word 11 - The word ITERAT

Word 12 - Minimum weight value to be plotted (floating point)

Word 13 - Maximum weight value to be plotted (floating point)

Word 14 - *

NOTE - Words 2-14 must be punched in columns 7-72 inclusive with at least one blank space between words. Words 12-13 are optional and may be omitted from the card.

Card 2

Columns 1-7 Case number

Columns 19-72 Case title

Cards 3 through n-1

Word 1 - Time, t_i, in months (integer), measured relative to words 4 and 5 on Card 1 above

Word 2 - Observed weight W_i in pounds (floating point)

Word 3 - Estimated fraction of observed weight ($0 \le e_i \le 1$)

Word 4 - Calculated fraction of observed weight ($0 \le c_i \le 1$)

Word 5 - Actual fraction of observed weight ($0 \le a_i \le 1$)

Word 6 - Nonrandom change for this month

Word 7 - *

NOTE - Words 2-7 must appear in columns 7-72 inclusive with at least one blank space between words.

Card n

The word END must appear in columns 7-9 and an * in column 12.

NEW POINT ADDITIONS

A new data point is defined as a series of values which update a given functional system from the previous month to the present month. These values appear on a data card with the following format:

Word 1 - Case number of the system to be updated

Word 2 - Observed weight (floating point)

Word 3 - Estimated percentage of observed weight (floating point)

Word 4 - Calculated percentage of observed weight (floating point)

Word 5 - Actual percentage of observed weight (floating point)

Word 6 - Nonrandom change

Word 7 - *

NOTE - As above, words 1-7 must appear in columns 7-72 inclusive with at least one blank space between words.

ERRONEOUS POINT CORRECTIONS

To correct previously submitted erroneous data points, the following corrections are permitted:

a. Corrections to the shipping date

Word 1 - S in column 1

Word 2 - Case number of the functional system which is to be changed

Word 3 - New month in which system is to be shipped (integer)

Word 4 - New year in which system is to be shipped (integer)

Word 5 - *

NOTE - Words 2-5 must appear in columns 7-72 inclusive with at least one blank space between words.

b. Corrections to the plotting scale factors

Word 1 - P in column 1

Word 2 - Case number of system which is to be corrected

Word 3 - New minimum scale factor

Word 4 - New maximum scale factor

Word 5 - *

NOTE - Words 2-4 must appear in columns 7-72 inclusive with at least one blank space between words.

c. Corrections to the ECA percentages at shipping date

Word 1 - ECA in columns 1-3

Word 2 - Actual percentage of weight at shipping date (floating point)

Word 3 - Calculated percentage of weight at shipping date (floating point)

Word 4 - Estimated percentage of weight at shipping date (floating point)

Word 5 - The word ITERAT

Word 6 - *

NOTE - Words 2-6 must appear in columns 7-72 inclusive with at least one blank space between words.

d. Corrections to the nonrandom change of some past data point

Word 1 - Case number of functional system which is to be changed

Word 2 - Corrected nonrandom change

Word 3 - Month in which the change applies

Word 4 - Year in which the change applies

Word 5 - *

NOTE - Words 1-5 must appear in columns 7-72 inclusive with at least one blank space between words.

e. Correction of a past observed data point

Word 1 - Case number of the functional system which is to be changed

Word 2 - Weight

Word 3 - Estimated percentage of weight

Word 4 - Calculated percentage of weight

Word 5 - Actual percentage of weight

Word 6 - Nonrandom change

Word 7 - Month to which the correction applies

Word 8 - Year to which the correction applies

Word 9 - *

NOTE - Words 1-9 must appear in columns 7-72 inclusive with at least one blank space between words.

f. Corrections to the entire set of control parameters (A card change)

Word 1 - Month in which the system is to be shipped (integer)

Word 2 - Year in which the system is to be shipped (integer)

Word 3 - Month in which the first data point was observed (integer)

Word 4 - Year in which the first data point was observed (integer)

Word 5 - Parameter r_1 (floating point)

Word 6 - Parameter r_2 (floating point)

Word 7 - Actual percentage of weight at shipping date (floating point)

Word 8 - Calculated percentage of weight at shipping date (floating point)

Word 9 - Estimated percentage of weight at shipping date (floating point)

Word 10 - The word ITERAT

Word 11 - Minimum weight value to be plotted (floating point)

Word 12 - Maximum weight value to be plotted (floating point)

Word 13 - *

NOTE - Words 1-13 must be punched in columns 7-72 inclusive with at least one blank space between words. Words 11 and 12 are optional and may be omitted.

METHOD OF UPDATING INPUT

So that the Weight Data File can be updated, it is necessary to mount the WDF on U04. The output file must be mounted on U03. The order of the input deck must be set up in one of two ways.

No New Cases

If no new cases are to be added the cards should be prepared as follows:

Card 1

Word 1 - The current month which is being added to the WDF (JAN, FEB, MAR, ...)

Word 2 - The current year (integer)

Word 3 - The word PRINT

Word 4 - *

Cards 2 through (K-1)

All additions or corrections as described in the preceding paragraphs. These cards may appear in any order.

Card K

Word 1 - The word END

Word 2 - The word OF

Word 3 - The word UPDATE

Word 4 - *

NOTE - Words 1-4 on both cards 1 and card K must be punched in columns 7-72 inclusive with at least one blank space between words.

New Cases

If there are one or more new cases to be added, the new cases must appear as the first data cards in the update deck. They must be manually sorted by case number and appear in ascending order. Following the END card of the last case to be added, must appear the following card:

Word 1 - END

Word 2 - OF

Word 3 - ADDS

Word 4 - *

NOTE - Words 1-4 must appear in columns 7-72 inclusive with at least one blank space between words.

Thereafter the input deck follows the format given under No New Cases.

METHOD OF UPDATING OUTPUT

Output consists of the updated Weight Data File together with a computer printout of this file. The format of the Weight Data File is presented on the following page. A copy of the computer printout is presented following the file format.

JOB DECK STRUCTURE - The Update Program is run as a separate job independent of SPACE. The job deck setup for processing the system Update Program appears below:

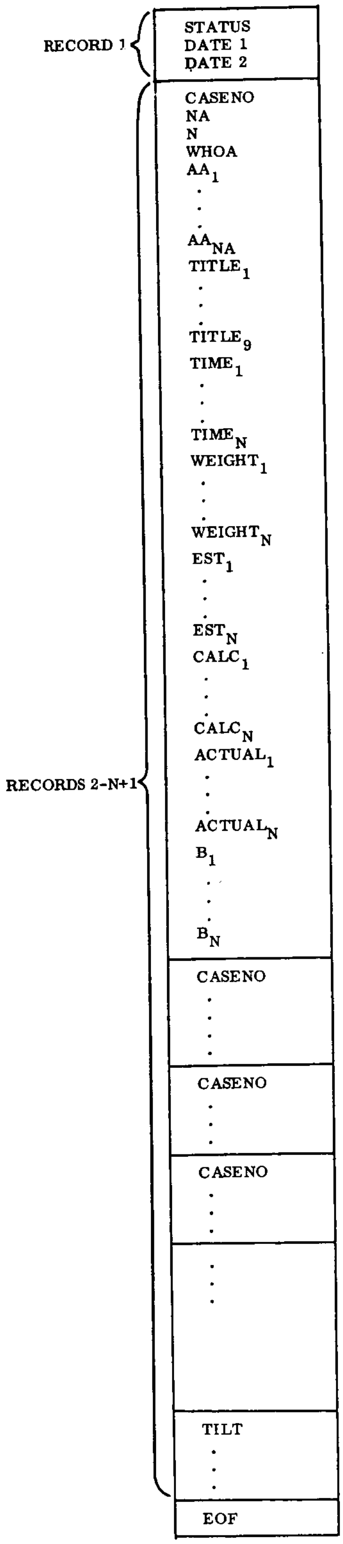
CARD COLUMNS

```
1-------8-------16-----------------------------------------------72
$ID                CHARGE NO., NAME, PAY NO., UNIT NO., PROGRAM, MODE
$PAUSE             (MOUNT RELOAD TAPE)
$OPEN
$IBJOB    SYSUPD   MAP, DLOGIC
$RELOAD            I06, NAME = SYSUPD, SRCH
       {
       {
       {           (INPUT DATA DECK)
       {
       {
$IBSYS
$CLOSE
$IBSYS
```

EXAMPLE OF INPUT DATA DECK

CARD COLUMNS

```
1------7---------------------------------------------------------72
RESTO  9   1967  6 1965   2.  5.  1.  0.  0.  ITERAT  *
5650103                TEST CASE
       1     12673.  .52  .44  .04   0.   *
       2     13411.  .48  .40  .12   0.   *
       3     13505.  .48  .40  .12  10.   *
       END     *
       END OF ADDS   *
       AUG 1965 PRINT   *
S      0662501  8    1967   *
       0652703  54000.    .53  .43  .04   6   1965     *
       END OF UPDATE     *
```



STATUS = a number associated with each weight data file (integer)

DATE 1 = MM/DD/

DATE 2 = YY

CASENO = case number (integer)

NA = number of entries in array AA

N = number of observations

WHOA = RESTO

AA = control parameters

TITLE = case title

TIME = time points

WEIGHT = observed weights

EST = estimated percentages

CALC = calculated percentages

ACTUAL = actual percentages

B = nonrandom changes

TILT = dummy word which signals the end of the WOF

NOTE - Record N+1 is a dummy record the first word of which is the word TILT.

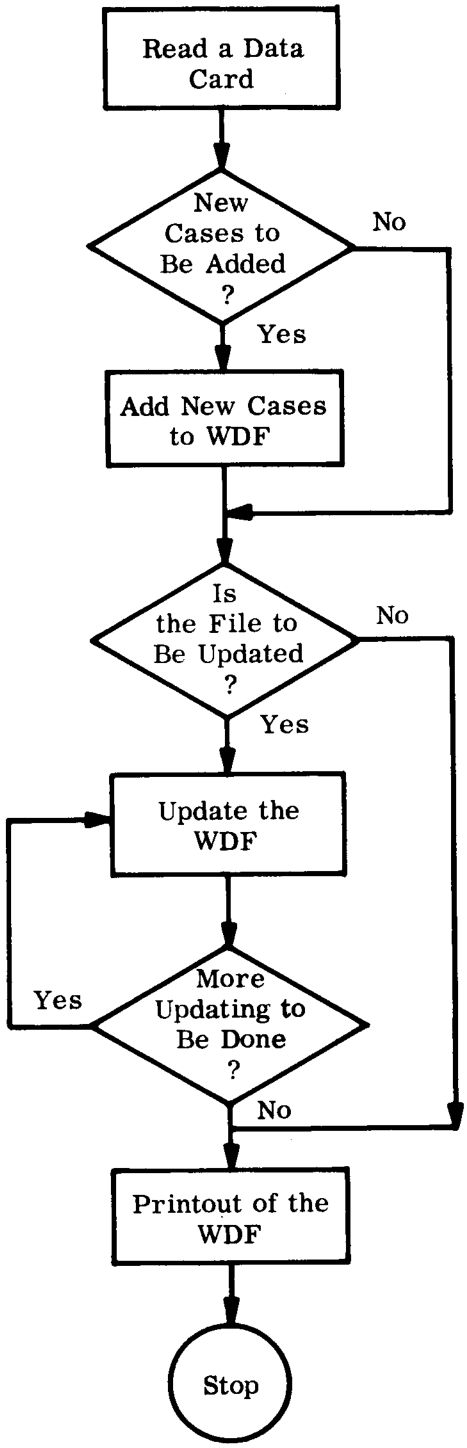
Reference No. 66.6
Issue Date 13 August 1965
Supersedes New

FILE STATUS NO 15 DATE 07/14/65

RESTO 8 1967 1 1964 2.000 5.000 1.000 0.000 0.000 ITERAT

661521 C/M GROSS CONTROLS + DISPLAYS SYSTEM WEIGHT

DATE	WEIGHT	EST	CAL	ACT	BUYOFF
JAN 64	288	92	8	0	0
FEB 64	300	32	68	0	0
MAR 64	299	20	80	0	0
APR 64	319	21	79	0	0
MAY 64	318	21	79	0	0
JUN 64	315	21	79	0	0
JUL 64	324	40	60	0	0
AUG 64	327	40	60	0	0
SEP 64	364	40	60	0	0
OCT 64	373	40	60	0	-10
NOV 64	246	70	30	0	125
DEC 64	243	70	30	0	0
JAN 65	258	70	30	0	0
FEB 65	293	100	0	0	-33
MAR 65	294	100	0	0	-1
APR 65	295	100	0	0	0
MAY 65	300	100	0	0	0
JUN 65	296	100	0	0	0


Read a Data Card
New Cases to Be Added ?
No
Yes
Add New Cases to WDF
Is the File to Be Updated ?
No
Yes
Update the WDF
More Updating to Be Done ?
Yes
No
Printout of the WDF
Stop

```
'BETC MAIN     LIST,REF
      REAL  INPUT
      INTEGER  CASENO,TILT,CASE
      INTEGER STATNO,WDCT,OLDST,STATUS,YEAR
      COMMON /GGDATE/DATE(2)
      DIMENSION            DATE2(2),YAT(12),B(100),X(15)
      DIMENSION AA(20),BB(9),          CC(6),CINP(10000)
      DIMENSION TIME(100),WEIGHT(100),EST(100),CALC(100),ACTUAL(100)
      DIMENSION U(3),AOUT(3),CH(2)
      DIMENSION NINP(10000)
      DIMENSION AAA(20),BBB(9),CCC(6),TT(100),WW(100),EE(100),CA(100),
     1AC(100),BY(100)
      DIMENSION DMOT(3)
      DIMENSION  BLK(12)
      EQUIVALENCE  (DMOT(2),KMOT)
      EQUIVALENCE  (LLTIM,CCC(1))
      EQUIVALENCE (AA(3),MONTH),(AA(4),YEAR)
      EQUIVALENCE (KTIME,CC(1)) ,(CINP,NINP)
      DATA END,OF,INPUT,UPDATE/3HEND,2HOF,5HINPUT,6HUPDATE/,FILE/4HFILE/
      DATA TILT/4HTILT/
      DATA  S,ECA,P/1HS,3HECA,1HP/
      DATA CH            /6H CHANG,1HE/,ACARD/6HA CARD/,CCARD/6HC CARD/
      DATA U             /6HMONTHL,6HY UPDA,2HTE/,BOFF/6HBUYOFF/
      DATA BLK/4HJAN ,4HFEB ,4HMAR ,4HAPR ,4HMAY ,4HJUN ,4HJUL ,4HAUG ,
     1        4HSEP ,4HOCT ,4HNOV ,4HDEC /
      DATA ADDS/4HADDS/
      DATA PRINT/5HPRINT/
      DATA AST/1H*/
C
C
C     MOUNT WEIGHT DATA FILE ON UNIT01
C
C
      READ(1)  OLDST,DATE2
      STATNO = OLDST + 1
      WRITE(2) STATNO,DATE
      WRITE(3) STATNO,DATE
      KSWIT=0
      NN=0
      L=2
      CALL READH(AAA,NAA,WHOAA)
      IF( NAA  .NE. 3 )           GO TO 7990
      DMOT(1)=AAA(1)
      DMOT(2)=AAA(2)
      L=1
      GO TO  825
  799 NN=0
      CALL READH(AAA,NAA,WHOAA)
      IF(AAA(1).EQ.END .AND. AAA(2).EQ.OF .AND. AAA(3).EQ.ADDS)GO TO 805
 7990 READ(5,798) CASE,BBB
  798 FORMAT(I7,11X9A6)
  800 CALL READH(CCC,NCC,WHOC)
      IF( NCC .EQ. 1 .AND. CCC(1) .EQ. END )   GO TO 801
      NN=NN+1
      TT(NN)=LLTIM
```

```
      WW(NN)=CCC(2)
      EE(NN)=CCC(3)
      CA(NN)=CCC(4)
      AC(NN)=CCC(5)
      IF( NCC  .EQ. 5   )  CCC(6)=0.
      BY(NN)=CCC(6)
      GO TO 800
 801  IF( KSWIT  .EQ. 1 )      GO TO 8021
 802  READ(1) CASENO,NA,N,WHOA,(AA(I),I=1,NA),BB,(TIME(I),I=1,N),(WEIGH
     1T(I),I=1,N),(EST(I),I=1,N),(CALC(I),I=1,N),(ACTUAL(I),I=1,N),(B(I)
     2,I=1,N)
 8021 IF( CASENO .EQ. TILT )  GO TO 8031
 803  IF( CASE    .GT. CASENO) GO TO 804
 8031 WRITE(2) CASE,NAA,NN,WHOAA,(AAA(I),I=1,NAA),BBB,(TT(I),I=1,NN),
     1           (WW(I),I=1,NN),(EE(I),I=1,NN),(CA(I),I=1,NN),(AC(I),I=1,
     2           NN),(BY(I),I=1,NN)
      KSWIT=1
      IF( CASE .EQ. CASENO )  KSWIT=0
      GO TO 799
 804  WRITE(2) CASENO,NA,N,WHOA,(AA(I),I=1,NA),BB,(TIME(I),I=1,N),(WEIGH
     1T(I),I=1,N),(EST(I),I=1,N),(CALC(I),I=1,N),(ACTUAL(I),I=1,N),(B(I)
     2,I=1,N)
      GO TO 802
 805  IF( KSWIT  .EQ. 0 )   GO TO 8052
 8051 WRITE(2) CASENO,NA,N,WHOA,(AA(I),I=1,NA),BB,(TIME(I),I=1,N),(WEIGH
     1T(I),I=1,N),(EST(I),I=1,N),(CALC(I),I=1,N),(ACTUAL(I),I=1,N),(B(I)
     2,I=1,N)
      IF( CASENO .EQ. TILT)    GO TO 806
 8052 READ(1) CASENO,NA,N,WHOA,(AA(I),I=1,NA),BB,(TIME(I),I=1,N),(WEIGH
     1T(I),I=1,N),(EST(I),I=1,N),(CALC(I),I=1,N),(ACTUAL(I),I=1,N),(B(I)
     2,I=1,N)
      GO TO 8051
 806  END FILE 2
      REWIND 1
      REWIND 2
      REWIND 3
C
C
      CALL READH(DMOT)
      READ(2) STATNO,DATE
      WRITE(3) STATNO,DATE
 825  LINES=0
      WRITE(6,796)  STATNO,DMOT(1),KMOT,DATE
 796  FORMAT(1H1,43X31HW E I G H T   D A T A   F I L E//
     1           42X36HRECORD OF UPDATING - FILE STATUS NO I2//
     2           45X21HLAST MONTH ON FILE - A3,1XI4//
     3           53X6HDATE  2A6///
     4           41X37HCASE     UPDATED   UPDATED   PREVIOUS/
     5           41X35HNUMBER   STATUS    WEIGHT    WEIGHT//)
      I=0
    1 CALL READH(X,WDCT,WHO)
      IF(X(1).EQ.END.AND.X(2).EQ.OF.AND.X(3).EQ.UPDATE)      GO TO 3
      I=I+1
      J=15*(I-1)
      CINP(J+1)=X(1)
```

```
      CINP(J+14)=WHO
      NINP(J+15)=WDCT-1
      K=J+1
      DO 2 IV=2,WDCT
      K=K+1
    2 CINP(K)=X(IV)
      GO TO 1
C
    3 IF( I .EQ. 0 )      GO TO 93
      CALL SORT(CINP,I,15,1)
      IMAX=I
C
COMMENCE UPDATING
C
      II=0
   30  READ(L) CASENO,NA,N,WHOA,(AA(I),I=1,NA),BB,(TIME(I),I=1,N),(WEIGH
     1T(I),I=1,N),(EST(I),I=1,N),(CALC(I),I=1,N),(ACTUAL(I),I=1,N),(B(I)
     2,I=1,N)
      IF( CASENO .EQ. IILT )  GO TO 155
      STATUS=OLDST
      IF( II .NE. 0 )     GO TO 311
   31 II=II+1
      J=15*(II-1)
  311 IF( II .GT. IMAX )  GO TO 6
  310 IF( CASENO .EQ. NINP(J+1) )  GO TO 8
      IF( CASENO .LT. NINP(J+1) )  GO TO 6
    4 WRITE(6,797)  NINP(J+1)
  797 FORMAT(62H CASENO ON UPDATE CARD IS MISPUNCHED. OFFENDING FIELDS R
     1EAD...I7)
      GO TO 31
    6 WRITE(3) CASENO,NA,N,WHOA,(AA(I),I=1,NA),BB,(TIME(I),I=1,N),(WEIGH
     1T(I),I=1,N),(EST(I),I=1,N),(CALC(I),I=1,N),(ACTUAL(I),I=1,N),(B(I)
     2,I=1,N)
      LINES=LINES+1
      IF( LINES .LE. 48 )  GO TO 795
      LINES=0
      WRITE(6,796)  STATNO,DMOT(1),KMOT,DATE
  795 NT1=TIME(N)
      KT1=MONTH+NT1-1
      LT1= (KT1-1)/12
      YEAR=YEAR+LT1
      NP= KT1-12*LT1
      IF( BLK(NP) .EQ. DMOT(1) .AND. YEAR .EQ. KMOT )  GO TO 95
      WRITE(6,794) CASENO,WEIGHT(N)
  794 FORMAT(41XI7,4X2HNO,15XF8.0)
      GO TO 30
   95 WRITE(6,7941) CASENO,WEIGHT(N),WEIGHT(N-1)
 7941 FORMAT(41XI7,4X3HYES,4XF8.0,2XF8.0)
      GO TO 30
    8 WDCT= NINP(J+15)
      IF( WDCT .EQ. 2  .AND. CINP(J+14) .EQ. S )  GO TO 880
      IF( WDCT .EQ. 2  .AND. CINP(J+14) .EQ. P )  GO TO 881
      IF( WDCT .EQ. 4  .AND. CINP(J+14) .EQ. ECA) GO TO 882
      IF( WDCT .EQ. 3 )       GO TO 13
      IF( WDCT .EQ. 5 )       GO TO 9
```

```
      IF( WDCT .EQ. 7 )        GO TO 10
      IF( WDCT.EQ.6 .OR. WDCT.EQ.10 .OR. WDCT.EQ.12 )   GO TO 79
      WRITE(6,790)  NINP(J+1)
  790 FORMAT(28H HEADER CARD ERROR. CASENO =  I7)
      GO TO 31
C
C
    9 N=N+1
      TIME(N)=TIME(N-1)+1.
   89 K=N
      STATUS=STATNO
   90 WEIGHT(K)=CINP(J+2)
      EST(K)=CINP(J+3)
      CALC(K)=CINP(J+4)
      ACTUAL(K)=CINP(J+5)
      B(K)=CINP(J+6)
      GO TO 31
   10 KMONTH=NINP(J+7)
      KYEAR=NINP(J+8)
      POINT = 12*( KYEAR - YEAR ) + KMONTH - MONTH + 1
      DO 11  I2=1,N
      IF( POINT .EQ. TIME(I2))     GO TO 12
   11 CONTINUE
      IF( POINT  .LT. TIME(N) )  GO TO  111
      N=N+1
      TIME(N)=POINT
      GO TO 89
  111 WRITE(6,789)  NINP(J+1)
  789 FORMAT(41H TIME MISPUNCHED ON HEADER CARD. CASENO =I7)
      GO TO 31
C
   12 K=I2
      GO TO 90
   13 KMONTH= NINP(J+3)
      KYEAR=  NINP(J+4)
      POINT= 12*(KYEAR-YEAR) +KMONTH -MONTH + 1
      DO 130  I3=1,N
      IF( POINT .EQ. TIME(I3))   GO TO 131
  130 CONTINUE
      GO TO 111
  131 B(I3)=CINP(J+2)
      GO TO 31
C
C     A CARD CHANGES
C
   79 KMONTH=NINP(J+4)
      KYEAR=NINP(J+5)
      POINT= 12*(KYEAR-YEAR) + KMONTH - MONTH
      IF( POINT  .EQ. 0. )     GO TO 81
      NPOINT=POINT +1.
      IP=0
   80 IP=IP+1
      TIME(IP)=TIME(NPOINT)-POINT
      WEIGHT(IP)=WEIGHT(NPOINT)
      EST(IP)=EST(NPOINT)
```

```
      CALC(IP)=CALC(NPOINT)
      ACTUAL(IP)=ACTUAL(NPOINT)
      B(IP)=B(NPOINT)
      NPOINT=NPOINT + 1
      IF( NPOINT .LE. N )      GO TO 80
      N=IP
   81 NA=WDCT
      LL=J+1
      DO 82  I3=1,NA
      LL1=LL+I3
   82 AA(I3)=CINP(LL1)
      GO TO 31
  880 AA(1)=CINP(J+2)
      AA(2)=CINP(J+3)
      GO TO 31
  881 AA(NA)=CINP(J+3)
      AA(NA-1)=CINP(J+2)
      GO TO 31
  882 AA(7)=  CINP(J+2)
      AA(8)=  CINP(J+3)
      AA(9)=  CINP(J+4)
      AA(10)=  CINP(J+5)
      GO TO 31
C
  155 WRITE(3) CASENO,NA,N,WHOA,(AA(I),I=1,NA),BB,(TIME(I),I=1,N),(WEIGH
     1T(I),I=1,N),(EST(I),I=1,N),(CALC(I),I=1,N),(ACTUAL(I),I=1,N),(B(I)
     2,I=1,N)
      END FILE 3
      REWIND 1
      REWIND 2
      REWIND 3
      LINES=0
      WRITE(6,788) STATNO,DATE
  788 FORMAT(1H1,28X61HC H R O N O L O G I C A L   R E C O R D   O F   C
     1 H A N G E S//
     242X15HFILE STATUS NO I2,4X6HDATE  2A6///
     35X42HCASE NUMBER           DESCRIPTION OF CHANGE//)
      DO  83  I2=1,IMAX
      J=15*(I2-1)
      WDCT=NINP(J+15)
      IF( WDCT .EQ. 2  .AND. CINP(J+14) .EQ. S )  GO TO 86
      IF( WDCT .EQ. 2  .AND. CINP(J+14) .EQ. P )  GO TO 86
      IF( WDCT .EQ. 4  .AND. CINP(J+14) .EQ. ECA) GO TO 86
      IF(WDCT .EQ. 3 )        GO TO  85
      IF(WDCT .EQ. 5 )        GO TO  84
      IF(WDCT .EQ. 7 )        GO TO  87
      IF( WDCT .EQ. 6 .OR. WDCT.EQ.10 .OR. WDCT.EQ.12 )  GO TO 86
      WRITE(6,787)  NINP(J+1)
  787 FORMAT(5XI7,2X17HHEADER CARD ERROR//)
      GO TO 83
   84 AOUT(1)=U(1)
      AOUT(2)=U(2)
      AOUT(3)=U(3)
      GO TO 88
   85 AOUT(1)=BOFF
```

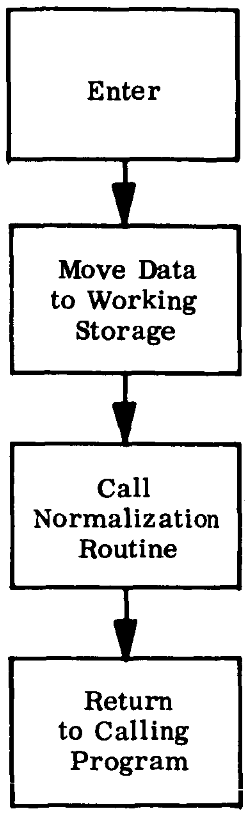
```
 850 AOUT(2)=CH(1)
     AOUT(3)=CH(2)
     GO TO 88
 86  AOUT(1)=ACARD
     GO TO 850
 87  AOUT(1)=CCARD
     GO TO 850
 88  WRITE(6,786)  NINP(J+1),AOUT
 786 FORMAT(5XI7,14X3A6)
     LINES=LINES+1
     IF( LINES .LE.  50 )        GO  TO  83
     LINES=0
     WRITE(6,788) STATNO,DATE
 83  CONTINUE
     READ(3) STATNO,DATE
 92   READ(3) CASENO,NA,N,WHOA,(AA(I),I=1,NA),BB,(TIME(I),I=1,N),(WEIGH
    1T(I),I=1,N),(EST(I),I=1,N),(CALC(I),I=1,N),(ACTUAL(I),I=1,N),(B(I)
    2,I=1,N)
     IF( CASENO .EQ. TILT )  GO TO 93
     IF( NA .EQ. 12 )    GO TO 94
     WRITE(6,785)  STATNO,DATE,WHOA,(AA(I),I=1,NA)
 785 FORMAT(1H1,41X15HFILE STATUS NO I2,4X6HDATE  2A6///1XA6,2XI2,2XI4,
    12XI2,2XI4,5(1XF7.3),2XA6//)
     GO TO 96
 94  WRITE(6,784)  STATNO,DATE,WHOA,(AA(I),I=1,NA)
 784 FORMAT(1H1,41X15HFILE STATUS NO I2,4X6HDATE  2A6///1XA6,2XI2,2XI4,
    12XI2,2XI4,5(1XF7.3),2XA6,2XF8.0,2XF8.0//)
 96  WRITE(6,783) CASENO,BB
 783 FORMAT(1XI7,11X9A6//)
     WRITE(6,200)
 200 FORMAT( /6X36HDATE     WEIGHT EST  CAL  ACT  BUYOFF//)
     NEAR=YEAR-1900
     NP=MONTH-1
     DO 91  J=1,N
     IF( J .EQ. 1 )  GO TO 7239
     IF((TIME(J) -TIME(J-1)) .EQ. 1.  )  GO TO 7239
     K3= TIME(J) -TIME(J-1) - .5
     DO 7240 I4=1,K3
     NP=NP+1
     IF (NP .LE. 12 )  GO TO 7240
     NP=1
     NEAR=NEAR+1
 7240 WRITE(6,7241) BLK(NP),NEAR
 7241 FORMAT(5XA4,I2)
 7239 NP=NP+1
     IF( NP .LE. 12 )   GO TO  7238
     NP=1
     NEAR=NEAR+1
 7238 NW=WEIGHT(J)+.5
     NEST= (EST(J)+.005 )*100.
     NCALC= (CALC(J)+.005 )*100.
     NACT= (ACTUAL(J)+.005 )*100.
     NB=B(J)
     WRITE(6,782)  BLK(NP),NEAR,NW,NEST,NCALC,NACT,NB
 782 FORMAT(5XA4,I2,1XI7,2XI3,2XI3,2XI3,1XI5)
```

```
   91 CONTINUE
      GO TO 92
   93 REWIND 3
      STOP
      END
```

SUBROUTINE GETDAT - 67S

This subroutine is available to each of the four trending programs and the History Plot Program. Its function is to move data from one storage location to another prior to the actual operations which are performed on the data. The subroutine is also responsible for calling the Normalization Routine 68S.

Control over this routine is maintained by the calling program.


Enter
Move Data to Working Storage
Call Normalization Routine
Return to Calling Program

```
$IBFTC GETDAT  LIST,REF
      SUBROUTINE GETDAT(IPSWT,I1,NCAS)
      COMMON /BLOCK/  NCASE (10) , TTITLE(90) , AAA   (150) , NANUM(10),
     1                TBLOCK(300), WBLOCK(300), EBLOCK(300) , NNUM (10),
     2                CBLOCK(300), ABLOCK(300), BBLOCK(300)
      COMMON /PPOG/ ACTUAL(100) , CALC  (100) , EST   (100) , COM  (12),
     1              LSQR  (100) , MEAN  (100) , MCONF (100) , UW2 (100),
     2              MSQR  (100) , PCONF (100) , S1    (100) , S2  (100),
     3              TIME  (100) , WEIGHT(100) , BUY   (100) , TITLE(9) ,
     4              N , NTOT
      COMMON  /HHP/  IOPT,JOPT,KOPT,NPATH
      NCAS=NCASE(I1)
      N=NNUM(I1)
      IPSWT=NANUM(I1)
      K1=15*(I1-1)
      DO  1001  I2=1,IPSWT
      K1=K1+1
 1001 COM(I2)=AAA(K1)
      K1=9*(I1-1)
      DO 1003  I2=1,9
      K1=K1+1
 1003 TITLE(I2)=TTITLE(K1)
      K1=30*(I1-1)
      DO 1005  I2=1,N
      K1=K1+1
      TIME(I2)=TBLOCK(K1)
      WEIGHT(I2)=WBLOCK(K1)
      EST(I2)=EBLOCK(K1)
      CALC(I2)=CBLOCK(K1)
      ACTUAL(I2)=ABLOCK(K1)
 1005 BUY(I2)=BBLOCK(K1)
      CALL NORM12 (KOPT)
      RETURN
      END
```

Reference No.	68.0
Issue Date	23 Dec 1965
Supersedes	New

NORMALIZATION SUBROUTINE - 68S (NORM12)

Normalization is the process of removing the effects of nonrandom changes from the data prior to trend prediction. Nonrandom weight changes are those changes not mathematically a part of normal weight growth. Two types of nonrandom weight changes are recognized by the subroutine:

a. Nonrandom changes No. 1 - These changes are, in general, buyoffs or transfer of weight between functional systems.

b. Nonrandom changes No. 2 - These changes consist of gross error eliminations or rejection of monthly changes which exceed preset criteria. The term outlier will characterize these changes.

NORMALIZATION USING NONRANDOM CHANGES NO. 1

We have n data points, or observations, $U_1, U_2, \ldots, U_n$. These points are normalized by applying all nonrandom changes, r_i, which are furnished as input, in the following manner. If r_i is the amount of change associated with the i-th observation U_i, then

$$V_k = U_k - \sum_{i=k+1}^{n} r_i, \qquad k = 1, 2, \ldots, n$$

represents the data normalized with nonrandom changes. Of course, if $r_i = 0$ for all i, then no normalization takes place and $V_i = U_i$. It should be noted that a nonrandom change is by definition positive if the weight was forced down. Consequently, a positive nonrandom change results in subtraction of weight while a negative nonrandom change results in addition of weight.

NORMALIZATION OF OUTLIERS

The points V_i undergo a second normalization if this option is specified in the input. The average monthly increment, $\overline{V}$,

$$\overline{V} = \frac{V_n - V_1}{n - 1}$$

is computed with the standard deviation

$$\sigma_V = \sqrt{\frac{\sum_{i=2}^{n} [(V_i - V_{i-1}) - \overline{V}]^2}{n - 2}}$$

Each monthly increment, $V_i - V_{i-1}$, is compared with the average $\overline{V}$. If it deviates from the average by more than $\pm 2\sigma_V$, it is assumed that the particular increment was not completely random. A nonrandom change, r_i', is therefore postulated and is:

$$r_i' = \begin{cases} -(V_i - V_{i-1} - \overline{V}) \text{ if } (V_i - V_{i-1} - \overline{V}) \geq 2\sigma_V \\ -(V_i - V_{i-1} - \overline{V}) \text{ if } (V_i - V_{i-1} - \overline{V}) \leq 2\sigma_V \\ 0 \text{ otherwise} \end{cases}$$

Normalization then is completely analogous to that above, namely

$$W_k = V_k - \sum_{i=k+1}^{n} r_i', \qquad k = 1, 2, \ldots, n$$

The subroutine allows four options:

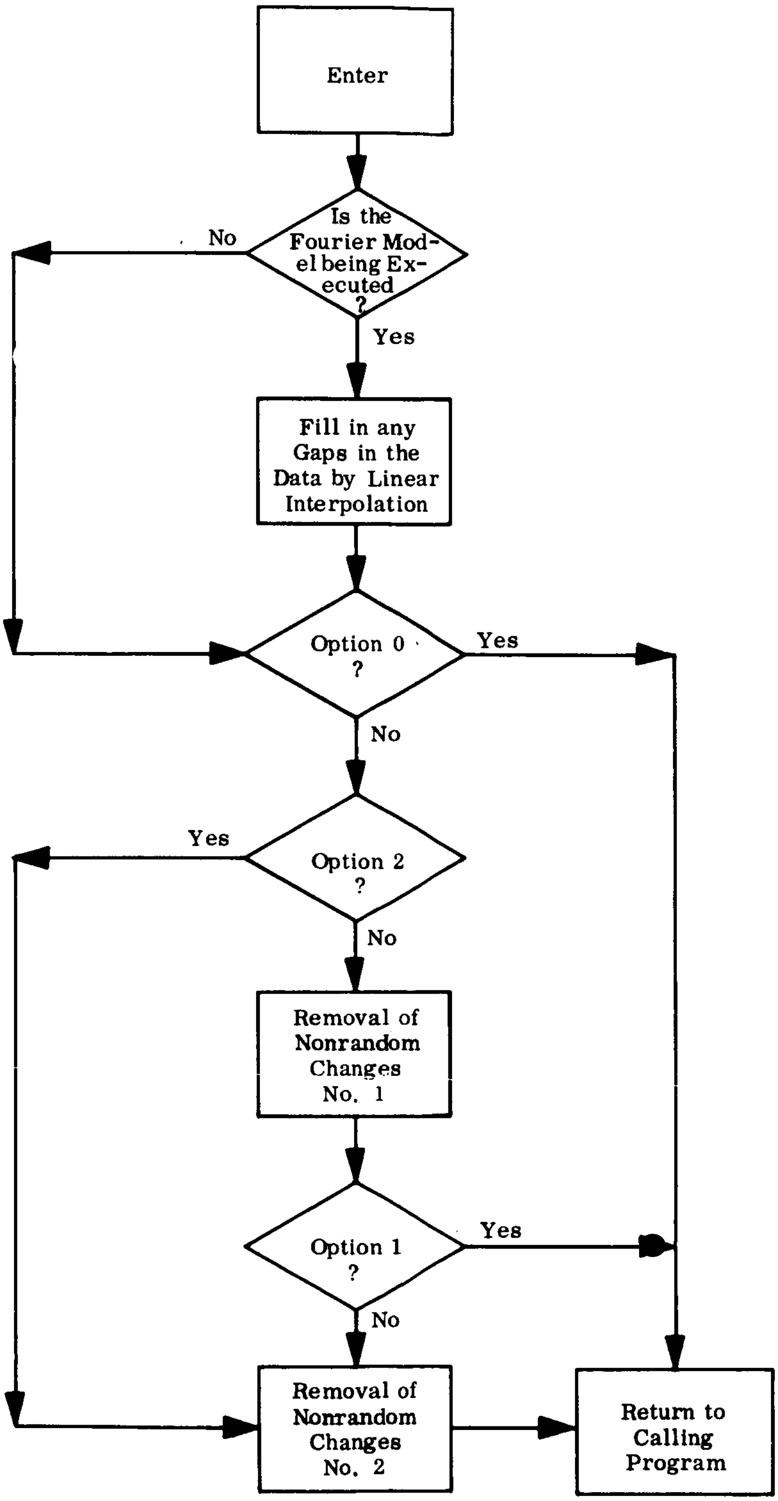
Option 0 - No normalization of data is performed.

Option 1 - Removal of nonrandom changes No. 1 only.

Option 2 - Removal of nonrandom changes No. 2 only.

Option 3 - Removal of nonrandom changes No. 1 followed by removal of nonrandom changes No. 2.

The user must specify which option he desires on the trend prediction card which is read by the Control Program 55S. For a detailed description of this data card see program 55S.


Enter
Is the Fourier Model being Executed ?
No
Yes
Fill in any Gaps in the Data by Linear Interpolation
Option 0 ?
Yes
No
Option 2 ?
Yes
No
Removal of Nonrandom Changes No. 1
Option 1 ?
Yes
No
Removal of Nonrandom Changes No. 2
Return to Calling Program

```
$IBFTC NORM12  LIST,REF
      SUBROUTINE NORM12 (IARG)
      REAL  MCONF
      COMMON /PPOG/ ACTUAL(100) , CALC  (100) , EST   (100) , COM  (12),
     1              LSQR  (100) , MEAN  (100) , MCONF (100) , UW2 (100),
     2              MSQR  (100) , PCONF (100) , S1    (100) , S2  (100),
     3              TIME  (100) , WEIGHT(100) , BUY   (100) , TITLE(9) ,
     4              N , NTOT
      COMMON /HHP/ IOPT,JOPT,KOPT,NPATH
      DIMENSION  TH(100),WH(100),EH(100),CH(100),AH(100),BH(100),
     1            NAST(100),NT(100),DIF(100)
C
      IF (NPATH .NE. 4)  GO TO 1001
      DO 3493 I=1,N
      TH(I)=TIME(I)
      WH(I) = WEIGHT(I)
      EH(I)=EST(I)
      CH(I)=CALC(I)
      AH(I)=ACTUAL(I)
 3493 BH(I)=BUY(I)
      NT(1)=1
      K4=1
      DO 3494  I=2,N
      IF(   (TH(I)-  TH(I-1)) .EQ. 1. )       GO TO 105
      K3=   TH(I)-  TH(I-1)-.5
      XWZ= (WH(I)-WH(I-1))/FLOAT(K3+1)
      DO 3495  I3=1,K3
      K4=K4+1
      TIME(K4)=TIME(K4-1)+1.
      NT(K4)=TH(I)
      BUY(K4)=0.
 3495 WEIGHT(K4) = WEIGHT(K4-1) + XWZ
  105 K4=K4+1
      TIME(K4)=TIME(K4-1)+1.
      NT(K4)=TH(I)
      WEIGHT(K4) = WH(I)
      EST(K4)=EH(I)
      CALC(K4)=CH(I)
      ACTUAL(K4)=AH(I)
 3494 BUY(K4)=BH(I)
      N=K4
 1001 CONTINUE
      DO 1004 MEM=1,N
      MCONF(MEM) = 0.0
 1004 UW2(MEM) = WEIGHT(MEM)
C
C IF IARG IS 1,  PERFORM NORMALIZATION NO. 1 ONLY
C IF IARG IS 2,  PERFORM NORMALIZATION NO. 2 ONLY
C IF IARG IS 3,  PERFORM NORMALIZATION NO. 1 AND 2
C IARG IS 1 OR 3, NORMALIZATION NO. 1 IS REQUIRED
      IF (IARG.LE.0)  GO TO 5901
      IF (IARG.GT.3)  GO TO 5901
```

```
      IF (IARG.EQ.2)  GO TO 5801
      DO 1007  I2=2,N
      IF( BUY(I2) .EQ. 0. )   GO TO 1007
      I22=I2-1
      DO 1006  I3=1,I22
 1006 WEIGHT(I3)=WEIGHT(I3)-BUY(I2)
 1007 CONTINUE
      IF (IARG.NE.3)  GO  TO  5901
C TO REACH 5801, IARG MUST BE 2 OR 3
C AT 5801, BEGIN NORMALIZATION  NO. 2
 5801 CONTINUE
      SUM=0.
      L = N - 1
      V = L
      DO 118  I=1,L
      DIF(I) = WEIGHT(I+1) - WEIGHT(I)
 118  SUM=SUM +  DIF(I)
      SUM=SUM/V
      SIGMA=0.
      DO 119  I=1,L
 119  SIGMA=SIGMA +  (DIF(I)-SUM)**2
      SIGMA=  SQRT( SIGMA/FLOAT(N-2))
      TWOSIG=2.*SIGMA
      DO 120  I=1,L
      DDIF=DIF(I)-SUM
      IF( DDIF .GE. (-TWOSIG) .AND. DDIF .LE. TWOSIG)  GO TO 120
      MCONF(I+1) = -DDIF
      DO 121  J=1,I
  121 WEIGHT(J) = WEIGHT(J) + DIF(I) - SUM
 120  CONTINUE
 122  CONTINUE
5901  RETURN
      END
```

Reference No. 69.0
Issue Date 23 Dec 1965
Supersedes New

DOUBLE PRECISION MATRIX INVERTER - 69S (DPMI)

DPMI is a double precision floating point matrix inversion subroutine employing the Gaussian elimination method with partial pivoting. The calling sequence is:

CALL DPMI (N, A, AI)

where N is a location containing the number of rows (columns) in the matrix to be inverted (address integer).

A is the first location of a block containing the matrix to be inverted.

AI is the first location of a block of at least $4N^2$ locations. Upon return, the inverse will be stored in the first $2N^2$ locations of this block.

```
        ENTRY   DPMI
DPMI    SAVE    1,4
        CLA     3,4
        STA     NAME+3
        CLA     4,4
        STA     NAME+4
        CLA     5,4
        STA     NAME+5
NAME    CALL    DPMI1(**,**,**,DET)
        CLA     1,4
        PDX     ,1
        TXL     RTRN,1,3
        CLA     6,4
        PAC     ,4
        CLA     DET
        STO     0,4
        CLA     DET+1
        STO     1,4
RTRN    RETURN  DPMI
DET     BSS     2
        EXTERN  DPMI1
        END
```

```
$IBFTC DPMI1   LIST,REF
CDPMI       - - - SUBROUTINE  DPMI - - -
C
C                DOUBLE PRECISION MATRIX INVERTER
C
C     THIS IS THE SUBROUTINE DMI1 BY M. J. SULLIVAN MODIFIED BY AM OLSON
C
C     EMPLOYS  OLD  SUBROUTINE DMI1
C     DPMI ACCEPTS DOUBLE PRECISION INPUT MATRIX, COMPUTES IN DOUBLE
C     PRECISION AND PRODUCES DOUBLE PERCISION INVERSE MATRIX
C     FLOATING INPUT OUTPUT AND COMPUTATIONS
C     STORAGE IS FROM HIGHER LOCATIONS TO LOWER LOCATIONS WITH HIGH
C     ORDER WORDS IN A BLOCK SEPARATE FROM THE LOW ORDER WORDS AS IN
C     FORTRAN II
C     MAXIMUM SIZE MATRIX IS 50 BY 50
C     SENSE LIGHT 3 IS TURNED ON FOR SINGULAR MATRIX OR FOR OVERFLOW
C     REQUIRES BLOCK OF ERASABLE STORAGE THE SIZE OF THE INVERSE BLOCK
C     (2*N**2) IMMEDIATELY BELOW THE INVERSE BLOCK
C     DETERMINANT OF INPUT MATRIX AVAILABLE
C
C     CALLING SEQUENCE         CALL DPMI(NSIZE,FINPUT,FOUTPUT,FDETERM)
C
      SUBROUTINE DPMI1(N,FINPUT,A,DETA)
      DOUBLE PRECISION FINPUT , DETA   , A      , ELMAX   , HOLD
      DOUBLE PRECISION HOLDR , DET
      DIMENSION FINPUT(1),DETA(1)
      DIMENSION A(1),ELMAX(1),HOLD(1),HOLDR(1),DET(1), ICOL(25),IROW(25)
      DATA HOLDBP/0113400000000/
      CALL SLITET(3,KOOF)
      GO TO (1,1),KOOF
 1    MN=N
      MM=MN-1
      N2=MN*MN
      MN2=MN+MN
      JFK=N2+N2
      JGK=N2+N2
C
C     (ARRANGE HI-LO ELEMENTS OF INPUT MATRIX FOR COMPUTATIONAL EASE)
 2    DO 3 I=1,N2
      K=N2+I
      A(I)=FINPUT(I)
3     A(K)=0.
      ASSIGN 11 TO ITRA
CC                                          GAUSSIAN ELIMINATION SCHEME
CC                                          ---- TRIANGULAR SYSTEM ----
4     DO 32 I=1,MM
      IC=I-1
      IJK=IC*MN
      L=IJK+I
C              PIVITOL CHOICE = MAX, ELEMENT OF SUB-MATRIX = A(M,K)
5     ELMAX=0.
      ICOL(I)=I
```

```
      IROW(I)=I
6     DO 10 M=I,MN
      J=L+M-I
      DO 9 K=I,MN
      HOLD=ABS(A(J))
      IF( ELMAX-HOLD)8,9,9
8     ELMAX=HOLD
      ICOL(I)=K
      IROW(I)=M
9     J=J+MN
10    CONTINUE
      GO TO ITRA, (11, 14)
C
C     ELIMINATE AS NOISE ALL A(I,J) BEYOND D-P RANGE OF MAXIMUM A(I,J)
C
11    HOLDR=ELMAX*HOLDBP
      DO 13  J=L,N2
      IF( ABS( A(J))-HOLDR )12,13,13
 12   A(J)=0.
13    CONTINUE
      DET=ELMAX
      ASSIGN 14 TO  ITRA
      GO TO 16
C            (ABSF (DET) OF MATRIX = PRODUCT OF PIVITOL VALUES)
14    IF (ELMAX-HOLDR) 63,15,15
15    DET=DET*ELMAX
16    CALL OVERFL(K000FX)
      GO TO (17,17,17),K000FX
17    IF(ICOL(I)-I) 18,20,18
C                               (INTERCHANGE I TH AND K TH COLUMNS OF A)
18    K=MN*(ICOL(I)-1)
      DO 19 J=1,MN
      M=IJK+J
      HOLD=A(M)
      A(M)=A(K+1)
      A(K+1)=HOLD
19    K=K+1
20    IF(IROW(I)-I) 21,23,21
C                            (INTERCHANGE I TH AND M TH ROWS OF A AND B)
21    K=IJK+IROW(I)
      M=L
      IC=MN+IC
      DO 22 J=I,IC
      HOLD=A(K)
      A(K)=A(M)
      A(M)=HOLD
      K=K+MN
22    M=M+MN
C                               DIVIDE I TH ROW OF A AND B BY A(I,I)
23    IA=N2+L
      A(IA)=1.
      IC=L+MN
```

```
      IA=N2+IJK+MN
      HOLD=A(L)
      DO 25 J=IC,IA,MN
      IF(A(J)) 24,25,24
 24   A(J)=A(J)/HOLD
 25   CONTINUE
      CALL OVERFL(K000FX)
      GO TO (63,26,26),K000FX
C                               (REDUCTION OF A AND B TO TRIANGULAR FORM)
 26   DO 31 J=I,MM
      L=L+1
      HOLD=A(L)
      IF(HOLD) 27,31,27
 27   IB=IC
      IA=IC+(J-I)
      DO 30 K=1,MN
      IF(A(IB)) 28,29,28
 28   A(IA+1)=A(IA+1)-A(IB)*HOLD
 29   IA=IA+MN
 30   IB=IB+MN
 31   CONTINUE
      CALL OVERFL(K000FX)
      GO TO (63,32,32),K000FX
 32   CONTINUE
C                                                  (REDUCE N TH ROW)
      L=JGK
      JGK=JKG-1
      J=N2+MN
      A(L)=1.
      HOLD=A(N2)
      IF( ABS(HOLD)-HOLDR)63,33,33
 33   DET=DET*HOLD
      CALL OVERFL(K000FX)
      GO TO (34,34,34),K000FX
 34   DO 35 I=J,L,MN
 35   A(I)=A(I)/HOLD
      CALL OVERFL(K000FX)
      GO TO (63,36,36),K000FX
C
C     (REDUCTION OF A TO UNITY-MATRIX, YIELDING (B) = PERMUTED INVERSE)
C
 36   DO 41 I=1,MM
      L=MN-I
      IJK=L
      K=L+1
      DO 40 M=1,IJK
      DO 39 J=1,MN
      IC=MN*(J-1)
      IA=MN*(K-1)+L
      IF( A(IA) )37,40,37
 37   IB= N2+(IC+L)
      IC= N2+(IC+K)
```

```
      IF(A(IC)) 38,39,38
38    A(IB)=A(IB)-A(IC)*A(IA)
39    CONTINUE
40    L=L-1
41    CONTINUE
      CALL OVERFL(K000FX)
      GO TO (63,42,42),K000FX
C
C(UNSCRAMBLE ROWS, COLS. OF PERMUTED MATRIX,(B) TO YIELD A-INVERSE)
C                  -ROWS-
42    IC=N2-MM
      DO 48 I=1,MM
      L=MN-I
      IF(ICOL(L)-L) 43,45,43
43    IA=N2+L
      IB=N2+ICOL(L)
      DO 44 J=1,MN
      HOLD=A(IA)
      A(IA)=A(IB)
      A(IB)=HOLD
      IA=IA+MN
44    IB=IB+MN
      M=-M
45    IF(IROW(L)-L) 46,48,46
C                  -COLUMNS-
46    IA=IC+MN*L
      IB=IC+MN*IROW(L)
      DO 47 J=1,MN
      HOLD=A(IA)
      A(IA)=A(IB)
      A(IB)=HOLD
      IA=IA+1
 47   IB=IB+1
      M=-M
48    CONTINUE
      IF(M) 49,50,50
49    DET=-DET
50    DO 53 I=1,N2
      J=N2+I
53    A(I)=A(J)
C
C     ELIMINATE AS NOISE ALL B(I,J) BEYOND D-P RANGE OF MAXIMUM B(I,J)
C
56    ELMAX=0.
      DO 58 I=1,N2
      HOLD=ABS(A(I))
      IF (HOLD-ELMAX) 58,58,57
57    ELMAX=HOLD
58    CONTINUE
      HOLDR=HOLDBP*ELMAX
      DO 60 I=1,N2
      IF( ABS( A(I) ) - HOLDR )59,60,60
```

```
59    J=N2+I
      A(I)=0.
60    CONTINUE
      DETA=DET
61    RETURN
C
C     ERRORS - (OVERFLOW OR INPUT MATRIX IS SINGULAR) -- INVERSE = INPUT
63    CALL SLITE (3)
      GO TO 61
      END
```

Reference No. 70.0
Issue Date 23 Dec 1965
Supersedes New

PROBABLE ERROR PROGRAM - 70S (RSS)

This program extracts the probable error from each functional system where probable error is defined as the difference between the +95 percent confidence limits and the mean line. These probable errors for the functional systems are then summed in a root sum square fashion to obtain the probable error for the stages or module. These numbers are then used with appropriate trade-off factors to compute a probable error for the total spacecraft or launch vehicle.

The program uses the results of the Fourier model in its computations, and the binary tape containing the Fourier output must be premounted. The following pages give examples of an input deck and typical program output.

SAMPLE OF RSS INPUT DECK

CARD COLUMNS

```
1------8-------------------------------------------------------72
MISION  207   3 67 503 5 67    506  8  67 *
PARTS   0611900   S/C FACTOR   1.0 3.026    5.9363 *
PARTS   0640500   S/C FACTOR   1.0 1.0      1.0    *
PARTS   0651500   S/C FACTOR   1.0 2.8165   2.0772 *
ENDCSE  *
MISION



ENDCSE  *
   .
   .
   .
MISION



ENDCSE  *
        END  RSS*
```

NOTE

The last card in the deck is END RSS. Upon encountering this card, the program will pass control back to SPACE.

MISSION	207		503		506	
SHIP DATE	3/1967		5/1967		8/1967	
611901	287,11	82433,78	302,64	91593,09	324,55	105332,06
611902	29,15	849,44	30,72	943,82	32,95	1085,39
611903	218,69	47825,04	230,52	53138,93	247,20	61109,77
611904	104,23	10864,69	109,87	12071,88	117,82	13882,67
611906	47,41	2247,31	49,97	2497,01	53,59	2871,56
611916	94,48	8926,46	99,59	9918,29	106,80	11406,04
611921	28,59	817,42	30,14	908,24	32,32	1044,48
611926	125,85	15838,13	132,66	17597,92	142,26	20237,61
611936	150,94	22783,12	159,11	25314,57	170,62	29111,76
611946	64,48	4157,80	67,97	4619,78	72,89	5312,75
611956	30,11	906,40	31,73	1007,11	34,03	1158,17

R.M.S. VALUES

R.M.S. VALUES	444,58	466,63	502,55

Reference No. 70.3
Issue Date 23 December 1965
Supersedes New

MISSION	207		503		506	
SHIP DATE	3/1967		5/1967		8/1967	
640503	436,64	190655,51	460,26	211839,44	493,57	243615,39
640536	38,44	1477,86	40,52	1642,07	43,46	1888,38

R,M,S, VALUES

	207	503	506
	438,33	462,04	495,48

Reference No. 70.4
Issue Date 23 December 1965
Supersedes New

MISSION	207		503		506	
SHIP DATE	3/1967		5/1967		8/1967	
651503	299,82	89893,57	316,04	99881,73	338,92	114864,02
651504	612,14	374715,36	645,25	416350,41	691,96	478802,99
651516	244,26	59660,83	257,47	66289,81	276,10	76233,28
651517	0,00	0,00	0,00	0,00	0,00	0,00
651526	62,27	3877,29	65,64	4308,10	70,39	4954,32
651527	0,00	0,00	0,00	0,00	0,00	0,00
651536	142,80	20390,83	150,52	22656,48	161,42	26054,95
651537	90,24	8143,03	95,12	9047,81	102,00	10404,99
651546	161,61	26118,83	170,36	29020,92	182,69	33374,06
651547	453,92	206039,13	478,47	228932,37	-4,68	-4,68
651556	79,75	6359,67	84,06	7060,31	90,15	8126,25
						R.M.S. VALUES
		891,74		939,98		867,65

Reference No. 70.5
Issue Date 23 December 1965
Supersedes New

MISSION	207		503		506	
SHIP DATE	3/1967		5/1967		8/1967	
611900	1,00	444,58	3,03	1418,06	5,94	2988,26
640500	1,00	438,33	1,00	462,04	1,00	495,48
651500	1,00	891,74	2,82	2647,44	2,08	1802,28
						R,M,S, VALUES
		1088,57		3038,64		3520,45

```
$IBFTC REPORT  LIST,REF
C    THIS ROUTINE WILL GENERATE ROOT MEAN SQUARE REPORTS FOR STRUCTURAL
C    DATA.
C    ALL SUBROUTINE COMMUNICATION WILL BE THROUGH THE  ('A') ARRAY.
C    THE FIRST 50 WORDS OF THE 'A' ARRAY WILL BE RESERVED FOR CONSTANTS
      INTEGER  REELS
      INTEGER  WDCT
      COMMON   /ACCESS/  HC(100),WDCT,ID(12),PROG
      COMMON   /SYSTEM/  NTAPES,REELS(15),CNTRLS(15),FILES(15),LRS(15),
     1                   POS(15),TRLPOS(15),RWCNT(15),UNITS(15)
      COMMON   /PFILE/ NFYLE(4,300),NFY(4)
      COMMON   /PPOG/    ACTUAL(100),CALC(100),EST(100),COM(12),
     1                   LSQR(100),MEAN(100),MCONF(100),UW2(100),
     2                   MSQR(100),PCONF(100),S1(100),S2(100),
     3                   TIME(100),WEIGHT(100),BUY(100),TITLE(9),
     4                   N,NTOT
      COMMON   /STT/  N21,D1,D2,J1RUN
      COMMON /ALL / A(3000)
      EQUIVALENCE (A,K)
      EQUIVALENCE (BLOCK(1,1),A(1000))
      EQUIVALENCE (A(1),WORD(1))
      EQUIVALENCE (ICASE(1),A(100 ))
      EQUIVALENCE ( WEIGH(1),A(300))
      EQUIVALENCE (IMISNO(1),K(50)) ,(IMONTH(1),K(60)) ,(IYEAR(1),K(70))
     1  ,(KRDATE(1),K(80))
      EQUIVALENCE  (K(30),MDATE) , (K(31),NUMCAS),  (K(32) ,IEQ )
      EQUIVALENCE  (K(20),NTAPE1) , (K(21), NTAPE2) , (K(22),NTAPE3) ,
     1 (K(23),NTAPE4) , (K(24),NTAPE5) , (K(25),NTAPE6)
      EQUIVALENCE (NCASE(1), A(200)    )
      EQUIVALENCE  (K(33),IXI)
      EQUIVALENCE  (K(34),MASK)
      EQUIVALENCE  (K(90),KFLAG)
      EQUIVALENCE (A(40),RMT(1))
      EQUIVALENCE  (A(500),SCF(1,1)) ,(A(2500),RMS(1,1))
      EQUIVALENCE  (HC(1),IC1)
      DIMENSION SCF(80,6),RMS(80,6)
      DIMENSION  RMT(10)
      DIMENSION  NCASE(1)
      DIMENSION K(1),WEIGH(100,2)
      DIMENSION BLOCK (100,15),WORD(10)
      DIMENSION KRDATE(1),IMISNO(1),ICASE(1),IMONTH(1),IYEAR(1)
      DATA  WORD /6H  N.A. , 6HTILT   ,6H      ,6H       ,6H        ,
     16H     ,6H       ,6H        ,6H        ,6H           /
      DATA NILT / 4HTILT  /
C    NOW, LET US START TO WORK........
      NP=0
      IF( WDCT  .EQ. 0    )        GO TO  1
      NFY(3)= IC1
    1 CALL PROCES
      DO 48 J=1,NUMCAS
      DO 48  I=1,MDATE
   48 RMS(J,I)=0.0
```

```
      KFLAG=1
C     THE (A(1000)TH) WORD STARTS THE  THE INPUT BUFFER
      IXI=1
   20 NP=NP+1
      IF (NP .GT. NFY(3) )    GO TO 220
      NX=ISIGN(NP,REELS(3))
      NF=REELS(3)*1000 + NX
      CALL READB1(NF,1)
      NOCASE=ID(7)
   21 IF( NOCASE .LT. ICASE(IXI) )  GO TO 20
C     RECORD FOUND....NOW GO TO WORK
      IF( NOCASE .LT. ( ICASE(IXI) + MASK) )  GO TO 49
C      PRINT ERROR COMMENT
      IXI=IXI+1
      GO TO 20
C     CODE FOUND.....NOW LETS DECIDE ON A DATE
   49 IEQ=0
   50 IEQ=IEQ+1
      N71=ID(10)
      N=ID(11)
      NMONTH=ID(8)
      NYEAR=ID(9)-1900
      DO 1112  I71=1,N71
      NMONTH=NMONTH + 1
      IF( NMONTH .LE. 12 )    GO TO 1112
      NMONTH=1
      NYEAR=NYEAR + 1
 1112 CONTINUE
      NPLUSP= ID(10)+ID(11)
      NUMER=5*NPLUSP + 2*ID(10) + 9
      CALL READB2(NF,ACTUAL,NUMER,HIST)
      CALL READB2(NF,WEIGH(1,1),N,HIST)
      CALL READB2(NF,ACTUAL,ID(10),HIST)
      CALL READB2(NF,WEIGH(1,2),N,HIST)
      NCASE(IEQ)=NOCASE
      KEY=(NYEAR -60)*12+NMONTH
      KEND=KEY+N
      IDT=1
   55 IF(KRDATE(IDT).GE.KEY ) GO TO 60
      IXX=IDT*2-1
      BLOCK(IEQ,IXX)=WORD(1)
      BLOCK(IEQ,IXX+1)=WORD(1)
      IDT=IDT+1
      GO TO 55
   60 CONTINUE
C     COMPUTE  THE CORRECT INDEXES FOR THE CORRECT DATES
      DO 210 I=IDT,MDATE
      KEYS=KRDATE(I)-KEY+1
      J=I*2-1
      IF(KEYS .GT.  N)  GO TO 205
      BLOCK(IEQ,J)=ABS(WEIGH(KEYS,1)-WEIGH(KEYS,2))
      BLOCK(IEQ,J+1)=BLOCK(IEQ,J)**2
      RMS(IXI,I)=RMS(IXI,I)+BLOCK(IEQ,J+1)
```

```
      GO TO 210
  205 BLOCK(IEQ,J)=WORD(1)
      BLOCK(IEQ,J+1)=WORD(1)
  210 CONTINUE
      NP=NP+1
      IF( NP .GT. NFY(3) .AND. IXI .EQ. NUMCAS  )   GO TO 214
      IF( NP .GT. NFY(3) )     GO TO 220
      NX= ISIGN(NP,REELS(3))
      NF= REELS(3)*1000   +   NX
      CALL READ31(NF,1)
      NOCASE=ID(7)
      IF( NOCASE .LT. ( ICASE(IXI) + MASK) )  GO TO 50
  214 DO 215  I=1,MDATE
  215 RMS(IXI,I)=SQRT(RMS(IXI,I))
      CALL PRINT
      IXI=IXI+1
      IF(IXI .LE. NUMCAS)  GO TO 21
C    LETS SUMMARIZE THE REPORT
      KFLAG=2
      DO 245 I=1,MDATE
  245 RMT(I)=0.0
      DO 250 I=1,NUMCAS
      DO 250 J=1,MDATE
      RMS(I,J)=RMS(I,J)*SCF(I,J)
  250 RMT(J)=RMT(J)+RMS(I,J)**2
      DO 260 I=1,MDATE
  260 RMT(I)=SQRT(RMT(I))
      CALL PRINT
      GO TO 1
  220 CONTINUE
C    SINCE THERE IS AN ERROR, LET US PROCEED TO THE NEXT CASE
      GO TO 1
      END
```

```
$IBFTC PROCES  LIST,REF
      SUBROUTINE PROCES
      COMMON /ALL / A(3000)
      EQUIVALENCE (A,K)
      EQUIVALENCE (BLOCK(1,1),A(1000))
      EQUIVALENCE (A(1),WORD(1))
      EQUIVALENCE (ICASE(1),A(100 ))
      EQUIVALENCE ( WEIGH(1),A(300))
      EQUIVALENCE (IMISNO(1),K(50)) ,(IMONTH(1),K(60)) ,(IYEAR(1),K(70))
     1  ,(KRDATE(1),K(80))
      EQUIVALENCE  (K(30),MDATE) , (K(31),NUMCAS),   (K(32) ,IEQ )
      EQUIVALENCE  (K(20),NTAPE1) , (K(21), NTAPE2) , (K(22),NTAPE3) ,
     1 (K(23),NTAPE4) , (K(24),NTAPE5) , (K(25),NTAPE6)
      EQUIVALENCE (IDATA(1),DATA(1), A(1001) )
      EQUIVALENCE  (K(34),MASK)
      EQUIVALENCE  (K(33),IXI)
      EQUIVALENCE (A(40),RMT(1))
      EQUIVALENCE  (K(90),KFLAG)
      EQUIVALENCE  (A(500),SCF(1,1)) ,(A(2500),RMS(1,1))
      DIMENSION SCF(80,6),RMS(80,6)
      DIMENSION  RMT(10)
      DIMENSION K(1),WEIGH(1,1)
      DIMENSION BLOCK (100,15),WORD(10)
      DIMENSION KRDATE(1),IMISNO(1),ICASE(1),IMONTH(1),IYEAR(1)
      DIMENSION CARDID(10)
      DIMENSION DATA(50),IDATA(50)
      DATA KFV /6HFACTOR       /
      DATA CARDID /6HMISION , 6HPARTS  ,6HMASK   ,6H       ,6H       ,
     16H      ,6H      ,6H       ,6HENDCSE,6H          /
      DATA END/3HEND/,RSS/3HRSS/
C    RESTORE COUNTERS
      MASK=100
      MFLAG =0
      NUMCAS=0
      MDATE=0
C
C
C
    1 CALL READH (DATA,NUM,XCARD)
      IF( DATA(1) .EQ. END .AND. DATA(2) .EQ. RSS )  CALL SPACE
      DO 5  I=1,10
      IF(XCARD .EQ.CARDID(I))GO TO 9
    5 CONTINUE
C    PRINT ERROR MESSAGE
      GO TO 1
    9 GO TO (10,20,30,40,50,60,70,80,90,100),I
   10 MDATE=NUM/3
      DO 11 I=1,MDATE
      J=(I-1)*3+1
C    STORE DATE AND MISSION NUMBER
      IMONTH(I)=IDATA(J+1)
      IYEAR(I) = IDATA(J+2)
```

```
      IMISNO(I)=IDATA(J)
   11 KRDATE(I)=IDATA(J+1)+(IDATA(J+2)-60)*12
      GO TO 1
   20 NUMCAS=NUMCAS+1
      ICASE(NUMCAS)=IDATA(1)
      IF(NUMCAS .NE. 2 )  GO TO 23
      MFLAG=1
   22 MASKED=MASK*10
      ICAS=ICASE(2)/MASKED
      ICAS= ICAS*MASKED
      IF(ICASE(2).NE.ICAS) GO TO 23
      MASK=MASKED
      GO TO 22
   23 IF(IDATA(3) .EQ.KFV ) GO TO 24
      SCF(NUMCAS,1)=-1.0
      IDATA(3) =0
      GO TO 1
   24 DO 26 I=1,6
   26 SCF(NUMCAS,I)=DATA(I+3)
      GO TO 1
   30 CONTINUE
      MASK=IDATA(1)
      MFLAG=1
      GO TO 1
   40 CONTINUE
      GO TO 1
   50 CONTINUE
      GO TO 1
   60 CONTINUE
      GO TO 1
   70 CONTINUE
      GO TO 1
   80 CONTINUE
      GO TO 1
   90 IF(MFLAG .EQ.  1    )     GO TO 999
C     SINCE NOT ENOUGH INFO HAS BEEN GIVEN,   SKIP THIS CASE AND WRITE
C      ERROR COMMENT
  100 GO TO 1
  999 RETURN
      END
```

```
$IBFTC PRINT    LIST,REF
      SUBROUTINE PRINT
      COMMON /ALL / A(3000)
      EQUIVALENCE (A,K)
      EQUIVALENCE (A(1),WORD(1))
      EQUIVALENCE (BLOCK(1,1),A(1000))
      EQUIVALENCE (ICASE(1),A(100 ))
      EQUIVALENCE ( WEIGH(1),A(300))
      EQUIVALENCE (IMISNO(1),K(50)) ,(IMONTH(1),K(60)) ,(IYEAR(1),K(70))
     1  ,(KRDATE(1),K(80))
      EQUIVALENCE  (K(30),MDATE) , (K(31),NUMCAS),  (K(32) ,IEQ )
      EQUIVALENCE  (K(20),NTAPE1) , (K(21), NTAPE2) , (K(22),NTAPE3) ,
     1 (K(23),NTAPE4) , (K(24),NTAPE5) , (K(25),NTAPE6)
      EQUIVALENCE (NCASE(1), A(200) )
      EQUIVALENCE  (K(33),IXI)
      EQUIVALENCE  (A(500),SCF(1,1)) ,(A(2500),RMS(1,1))
      EQUIVALENCE (A(40),RMT(1))
      EQUIVALENCE  (K(90),KFLAG)
      DIMENSION SCF(80,6),RMS(80,6)
      DIMENSION  RMT(10)
      DIMENSION  NCASE(1)
      DIMENSION K(1),WEIGH(1,1)
      DIMENSION BLOCK (100,15),WORD(10)
      DIMENSION KRDATE(1),IMISNO(1),ICASE(1),IMONTH(1),IYEAR(1)
C   THIS ROUTINE PRINTS THE REPORT
      WRITE (      6,104)
  104 FORMAT (1H1)
      WRITE (      6,100) (IMISNO(I),I=1,MDATE)
  100 FORMAT (//3X,7HMISSION, 10X,  I4, 5(16X,I4)   )
      WRITE(      6,103)
  103 FORMAT (/ )
      WRITE (      6,101)((IMONTH(I),IYEAR(I)),I=1,MDATE)
  101 FORMAT (2X,9HSHIP DATE , 8X, I2,3H/19,I2, 5(13X,I2,3H/19,I2))
      WRITE(      6,103)
      WRITE(      6,103)
      GO TO (1,200,300),KFLAG
    1 MDATED=MDATE*2
      DO 10  KK=1,IEQ
      WRITE (      6,102)    NCASE(KK),(BLOCK(KK,I),I=1,MDATED)
  102 FORMAT (2X,I7,2X,12F10.2)
   10 CONTINUE
      WRITE (      6,105)   (RMS(IXI,I),I=1,MDATE)
  105 FORMAT(// 64X,14HR.M.S. VALUES   //   11X,6F20.2)
      GO TO 999
  200 CONTINUE
      DO  210 KK=1,NUMCAS
  210 WRITE (      6,102 )    ICASE(KK),((SCF(KK,I),RMS(KK,I)),I=1,MDATE)
      WRITE (      6,105)   (RMT(I),I=1,MDATE )
      WRITE (      6,104)
  300 CONTINUE
  999 RETURN
      END
```

Reference No. 70.12
Issue Date 23 Dec 1965
Supersedes New

```
$IBFTC BDATA    LIST,REF
      BLOCK DATA
      COMMON/ALL / A(3000)
      EQUIVALENCE (A,K)
      DIMENSION K(3000)
      DATA  K(20),K(21),K(22),K(23),K(24),K(25)  / 1,2,3,4,5,6 /
      END
```

Reference No. 71.0
Issue Date 23 Dec 1965
Supersedes New

PLOTTING SUBROUTINE - 71S (UMPLOT)

GENERAL DESCRIPTION

UMPLOT is an acronym for University of Michigan Plotting Routine. The philosophy used in writing the routine was to treat a region of core storage (subsequently, called the image region or simply the image) much as a piece of graph paper when plotting data manually.

First, the image region is blanked out, and a grid, formed of I's and -'s (with +'s at the intersection points), is placed in the image region. Given the numerical limits of the abscissa and ordinate, (i.e., the maximum and minimum values of the two variables, say x and y), the routine can place any specified BCD plotting character at the appropriate position in the image for a given pair of data values (x_i, y_i).

Each point (x_i, y_i) is plotted individually and independently of any preceding point. In other words, the data need not be presorted. Any number of points (x_i, y_i) with any corresponding BCD plotting characters can be placed in the image. A character falling on a previously plotted character will replace that character. Thus, only the last one plotted of two coincident data points appears in the final image. Points falling outside the grid limits (not in the image region) are ignored.

When all desired points have been placed in the region, the image is copied onto the specified decimal output tape for subsequent off-line (or simulated off-line) printing or punching. Any number of duplicate copies of the graph can be produced.

The subroutine has four main entries which perform the following functions:

PLOT 1

This entry to the subroutine sets up the grid spacing and the total width and length of the graph image. It also determines the location of the decimal points and the multiplying factors (powers of 10) for values of the ordinate and abscissa to be printed at the grid lines.

PLOT 2

This entry to the subroutine prepares the grid, examines the maximum and minimum values of the abscissa and ordinate, and establishes internally a formula for computing the location in the image region corresponding to the point (x_i, y_i).

PLOT 3

This entry to the subroutine places a specified BCD plotting character in the appropriate position(s) corresponding to the given value(s) of (x_i, y_i).

PLOT 4

PLOT 4 (or FPLOT 4) entry writes the image of the completed graph on the output tape for subsequent printing off-line. A label for the ordinate is printed vertically (one character per line) at the left edge of the page. Values of the abscissa and ordinate are printed at the grid lines outside the bottom and left edges of the graph.

FORTRAN CALLING SEQUENCES

```
Call  PLOT 1     (NSCALE, NHL, NSBH, NVL, NSBV)
Call  PLOT 2     (IMAGE, XMAX, XMIN, YMAX, YMIN)
Call  PLOT 3     (BCD, X, Y, NDATA)
Call FPLOT 4     (NCHAR, nHABCDEF. . . . .)
```

DESCRIPTION OF ARGUMENTS

NSCALE

This is a vector (array) in the user's program having one or five locations. If the user wishes to use the standard scale factors and decimal point positions (see below), NSCALE should equal zero. To alter the standard factors, NSCALE must be any nonzero quantity. In this case, the NSCALE array must have five locations containing the following information:

FORTRAN Location	Contents	Function
NSCALE (1)	Any nonzero value	Alter standard factors.
NSCALE (2)	I	Printed values of the ordinate (y) are 10.P.I times the actual values.
NSCALE (3)	J	Printed values of the ordinate (y) have J digits following the decimal point (J.LE.8).

FORTRAN Location	Contents	Function
NSCALE (4)	K	Printed values of the abscissa (x) are 10.P.K times the actual values.
NSCALE (5)	M	Printed values of the abscissa (x) have M digits following the decimal point (M.LE.9).

STANDARD SCALE FACTORS

When NSCALE is zero, the standard scale factors are used. The effective values of I, J, K, and M are 0, 3, 0, and 3, respectively. The actual values are printed with three decimal places.

NHL	The number of horizontal grid lines in the graph image.
NSBH	The number of spaces between horizontal grid lines.
NVL	The number of vertical grid lines in the graph image.
NSBV	The number of spaces between vertical grid lines.

NOTE

In keeping with standard notation for graph paper, (e.g., 10 x 10 to the inch) NHL and NVL are really one less than the actual number of lines. It is not customary to consider the axes when counting lines in the grid.

IMAGE — An array (vector), dimensioned in the user's program consisting of N sequential locations not used between execution of PLOT 2 and PLOT 4, where

N = P*(NSBH*NHL + 1)

P = (NSBV*NVL + 1)/6, rounded up to the nearest integer

XMAX	The value of the abscissa at the rightmost grid line.
XMIN	The value of the abscissa at the leftmost grid line.
YMAX	The value of the ordinate at the uppermost grid line.
YMIN	The value of the ordinate at the lowermost grid line.
BCD	The BCD (Hollerith) plotting character, and may be any legitimate left-adjusted BCD character (letter, digit, blank, or special character * , . + etc.).
X	A single location (or array name) containing the x coordinate(s) of the point(s), (x_i, y_i).

Y	A single location (or array name) containing the y coordinate(s) of the point(s), (x_i, y_i).
NDATA	The number of data points (x_i, y_i) associated with the arrays x and y. With NDATA equal to 1, a single point will be plotted for a single execution of PLOT 3. With NDATA equal to Q, Q points (x_i, y_i) taken in sequence from vectors of length Q starting at x and y are plotted for a single execution of PLOT 3.
NCHAR	The number of BCD (Hollerith) characters (including blanks) in the label array (vector).
LABEL	The name of an array (vector) which contains the string of BCD characters to be printed at the left edge of the output page, i.e., a label for the ordinate of the graph.

LABELING THE ORDINATE - USE OF FPLOT 4

The string of characters for the ordinate label appears directly in the calling sequence as the second argument (Hollerith). The n preceding the H (specifying the Hollerith string) should be the same as the value of NCHAR.

RESTRICTIONS ON ARGUMENTS

NHL	.GT.	0	
NSBH	.GT.	0	
NVL	.GT.	0	
NSBV	.GT.	0	
(NSBV*NVL)	.LE.	101	
XMAX	.GT.	XMIN	
YMAX	.GT.	YMIN	
BCD			Must be a left-adjusted legitimate BCD (Hollerith) character, i.e., 1H-, 1H*, 1HA, 1H1, etc. (FORTRAN)

MODES OF ARGUMENTS

Those arguments which deal directly with data values (XMAX, XMIN, YMAX, YMIN, X, Y) must be in floating point mode.

Those arguments which deal with the arrangement of the image and the scale factors (NSCALE, NHL, NSBH, NVL, NSBV, NCHAR) and the number of data points can be:

a. Floating Point

b. FORTRAN type integers

The routine automatically determines which mode is being used for each argument.

LABEL and BCD must contain Hollerith information only.

```
$IBMAP UMPLOT
*                  PLOT1
         ENTRY   PLOT1
         ENTRY   PLOT2
         ENTRY   PLOT3
         ENTRY   PLOT4
         ENTRY   FPLOT4
         ENTRY   OMIT
         PZE     0
         PZE     0
A4       DEC     6
PLOT1    SAVE    1,2,4
         CLA     3,4
         STA     A141
         STA     A144
         STZ     A1723
         CLA     A1657
         STO     A1725
         CLA*    4,4
         TSX     A1072,2
         TZE     A63
         STO     A1655
         CLA*    5,4
         TSX     A1072,2
         TZE     A63
         STO     A1666
         LDQ     A1666
         MPY     A1655
         STQ     A1633
         CLA*    6,4
         TSX     A1072,2
         TZE     A63
         STO     A1656
         CLA*    7,4
         TSX     A1072,2
         TZE     A63
         STO     A1667
         LDQ     A1667
         MPY     A1656
         STQ     A1703
         LLS     35
         ADD     A1657
         STO     A1704
         SUB     A1622
         TMI     A71
A63      CLA     A1607
         STO     A1723
         AXT     -*-1,4
         TRA     A223
         PZE     A1600,,1
A70      MTH     A413
A71      CLA     A1704
```

```
        TSX     A1067,2
        FDP     A1673
        STQ     A1701
        CLA     A1701
        FAD     A1647
        TSX     A1102,2
        STO     A1716
        LDQ     A1716
        MPY     A1672
        STQ     A1705
        LXA     A1716,2
        CLA     A1705
        SUB     A1704
        PAX     ,1
        CLA     A1576,1
        STO     A1567,2
        LDQ     A1513
        STQ     A1513,2
        TIX     A116,2,1
        MTH     A122
A116    CLA     A1576
A117    STO     A1567,2
        STQ     A1513,2
        TIX     A117,2,1
A122    STZ     A1623
A123    TSX     A1123,4
        PTH     A1624
        PTH     A1623
        PTH     A1512
        TSX     A1123,4
        PTH     A1625
        PTH     A1623
        PTH     A1566
        CLA     A1623
        ADD     A1667
        STO     A1623
        SUB     A1704
        TZE     A123
        TMI     A123
A141    CLA     **
        TZE     A150
        AXT     -1,4
A144    CLA     ,4
        TSX     A1073,2
        SXA     XY,4
        LAC     XY,2
        STO     A1620,2
        TXI     *+1,4,-1
        TXH     A144,4,-5
A150    CLA     A1616
        TZE     A155
        TPL     A155
```

```
        CLS     A1657
A154    TXL     A160,,20
A155    STO     XZ
        CLA     A1577
        CAS     XZ
        TRA     A161
        TRA     *+1
A160    STO     A1616
A161    CLA     A1614
        SUB     A1702
        TPL     *-1
        ADD     A1702
        TZE     A170
        TPL     A170
        CLS     A1657
A170    STO     A1614
        CLA     A1667
        SUB     A1657
        STO     A1621
        CAS     A1614
        TRA     A200
        TRA     *+1
        SUB     A1657
        STO     A1614
A200    CLA     A1711
        ADD     A1614
        STO     A1620
        CLA     A1665
        SUB     A1704
        SUB     A1620
        TPL     A211
        ADD     A1620
        STO     A1620
A211    CLA     A1620
        CAS     A1614
        TRA     A216
        TRA     *+1
        SUB     A1657
        STO     A1614
A216    ZAC
        RETURN  PLOT1
A223    SXA     LOAD4,4
        STO     =HSAVEAC
        CLA     1,4
        ORA     =030000000000
        STO     Y1
        CALL    JOBOU(Y1-1)
LOAD4   AXT     **,4
        CLA     =HSAVEAC
        TRA     2,4
PLOT2   SAVE    1,2,4
        CLA     A507
```

```
        COM
        STD     A1065
        STZ     A1725
        CLA     A1723
        TZE     A243
        CLA     A1610
        RETURN  PLOT2
A243    LDQ     A1633
        MPY     A1716
        STQ     S1
        CLA     A1716
        SSP
        ADD     S1
        SUB     A1634
        STO     S3
        CLA     3,4
        SSP
        ADD     S3
        STA     A341
        STA     A367
        STA     A530
        STA     A744
        STZ     A1724
        CLA*    4,4
        TSX     A1056,2
        STO     A1727
        CLA*    5,4
        TSX     A1056,2
        STO     A1730
        CLA*    6,4
        TSX     A1056,2
        STO     A1731
        CLA*    7,4
        TSX     A1056,2
        STO     A1732
        CLS     A1730
        FAD     A1727
        TMI     A411
        STO     A1674
        CLS     A1732
        FAD     A1731
        TMI     A411
        STO     A1675
        CLA     A1633
        TSX     A1067,2
        FDP     A1675
        STQ     A1714
        CLA     A1703
        TSX     A1067,2
        FDP     A1674
        STQ     A1715
        CLA     A1656
```

```
         TSX    A1067,2
         STO    A1701
         CLA    A1674
         FDP    A1701
         STQ    A1567
         CLA    A1655
         TSX    A1067,2
         STO    A1701
         CLA    A1675
         FDP    A1701
         STQ    A1570
         STZ    A1623
         STZ    A1545
A333     STZ    A1630
A334     CLA    A1630
         PAX    ,2
         ADD    A1623
         PAX    ,4
         CLA    A1566,2
A341     STO    ,4
         CLA    A1630
         ADD    A1657
         STO    A1630
         SUB    A1716
         TNZ    A334
         CLA    A1623
         ADD    A1716
         STO    A1623
         CLA    A1655
         SUB    A1545
         TZE    A4161
         CLA    A1657
         STO    A1701
         SUB    A1666
         TZE    A405
A361     STZ    A1630
A362     CLA    A1630
         PAX    ,2
         ADD    A1623
         PAX    ,4
         CLA    A1512,2
A367     STO    ,4
         CLA    A1630
         ADD    A1657
         STO    A1630
         SUB    A1716
         TNZ    A362
         CLA    A1623
         ADD    A1716
         STO    A1623
         CLA    A1701
         ADD    A1657
```

```
         STO     A1701
         SUB     A1666
         TNZ     A361
A405     CLA     A1545
         ADD     A1657
         STO     A1545
         TXL     A333,,19
A411     CLA     A1610
         STO     A1724
A413     AXT     -*-1,4
         TRA     A223
         PZE     A1717,,4
A416     TXL     *+1,0,**
         RETURN  PLOT2
A4161    ZAC
         RETURN  PLOT2
A417     AXT     -*-1,4
         TRA     A223
         PZE     A1602,,3
         TXL     A416,0,
PLOT3    SAVE    1,2,4
         STZ     A1606
         CLA     3,4
         STA     A526
         CLA     4,4
         ADD*    6,4
         SUB     A1634
         STA     A500
         CLA     5,4
         ADD*    6,4
         SUB     A1634
         STA     A460
         CLA     A1723
         ORA     A1724
         TZE     A442
         CLA     A1611
         RETURN  PLOT3
A442     ORA     A1725
         TZE     A446
         CLA     A1611
         TXI     A417,4,1
A446     CLA*    6,4
         TSX     A1072,2
         TNZ     A455
         CLS     A1611
         RETURN  PLOT3
A455     STO     A1654
         STZ     A1630
A457     LXA     A1630,1
A460     CLS     ,1
         TSX     A1056,2
         FAD     A1731
```

```
        LRS     35
        FMP     A1714
        TPL     A470
        FSB     A1645
        MTH     A471
A470    FAD     A1645
A471    TSX     A1102,2
        STO     A1623
        TZE     A500
        TMI     A532
        SUB     A1633
        TZE     A500
        TPL     A532
A500    CLA     ,1
        TSX     A1056,2
        FSB     A1730
        LRS     35
        FMP     A1715
        TPL     A510
        FSB     A1645
A507    TXL     A511,,PLOT3
A510    FAD     A1645
A511    TSX     A1102,2
        STO     A1545
        TZE     A520
        TMI     A532
        SUB     A1703
        TZE     A520
        TPL     A532
A520    LDQ     A1705
        MPY     A1623
        LLS     35
        ADD     A1545
        STO     A1631
        TSX     A1123,4
A526    PTH     0
        PTH     A1631
A530    OCT     377777000000
A531    TXL     A534,,4
A532    CLS     A1611
        STO     A1606
A534    CLA     A1630
        ADD     A1657
        STO     A1630
        SUB     A1654
        TNZ     A457
        CLA     A1606
        RETURN  PLOT3
FPLOT4  SAVE    1,2,4
        CLA     A530
        TXL     A547,0,
PLOT4   CLA     A1660
```

Reference No. 71.12
Issue Date 23 Dec 1965
Supersedes New

```
A547    STD     A713
        CLA     A1723
        ORA     A1724
        TZE     A560
        CLA     A1612
        RETURN  FPLOT4
A560    ORA     A1725
        TZE     A564
        CLA     A1612
        TXI     A417,4,3
A564    CLA     4,4
        STA     A601
        STA     A714
        CLA*    3,4
        TSX     A1072,2
        ADD     A1657
        PAX     ,4
        SXD     A1212,4
        STD     A1440
        LXA     A1672,1
        SXD     A416,1
A601    LDQ     **
        STQ     A1632
        CLA     A1617
        STO     A1653
        CLA     A1616
        STO     A1651
        CLA     A1633
        STO     A1605
        CAL     A1626
        ANA     A1627
        TZE     A617
        CLA     A1633
        SUB     A1657
        STO     A1605
A617    STZ     A1623
        LXA     A1623,2
A621    CLA     A1605
        SUB     A1623
        TMI     A763
        SUB     A1634
        LXA     A1634,4
        TPL     A632
        ADD     A1634
        ADD     A1657
        PAX     ,4
A632    SXD     A1066,4
        SXD     A70,2
A634    CLA     A1514
        STO     A1542
        CLA     A1513
        STO     A1541
```

```
        PXD     128,0
        LDQ     A1623
        DVP     A1666
        STQ     A1631
        TNZ     A667
        CAL     A1626
        ANA     A1713
        TNZ     A667
        CLA     A1631
        TSX     A1067,2
        LRS     35
        FMP     A1570
        STO     A1701
        CLA     A1731
        FSB     A1701
        STO     A1471
        CLA     A1713
        STO     A1545
        CLA     A1650
        STO     A1652
        SXD     A762,4
        TSX     A1156,4
        LXD     A762,4
A667    CLA     A1542
        STO     A1541,4
        CLA     A1541
        STO     A1540,4
        CLA     A1623
        ADD     A1657
        STO     A1623
        TIX     A634,4,1
        LXD     A1066,4
A700    SXD     A762,4
        LXD     A1212,4
        CAL     A1676
        TNX     A717,4,1
        LXD     A1440,2
        LXD     A416,1
        LDQ     A1632
        LGL     6
        ALS     24
        TIX     A715,1,1
        LXA     A1672,1
A713    TXI     *+1,2,**
1
A1441   LXD     A1316,4
        TRA     1,4
A1443   LDQ     A1607
        STQ     A1662
        PAX     ,4
        TXH     A1451,4,0
        CLA     A1607
```

```
        TRA     1,2
A1451   TMI     A1457
A1452   LDQ     A1662
        FMP     A1613
        STO     A1662
        TIX     A1452,4,1
        TRA     1,2
A1457   CLA     A1662
        FDP     A1613
        STQ     A1662
        TIX     A1457,4,1
        CLA     A1662
        TRA     1,2
        OCT     77777
TEMP    PZE     0
        PZE     1
Y1      BSS     1
S7      PZE
S6      PZE
S5      PZE
S4      PZE
S3      PZE
S1      PZE
XZ      PZE
XY      PZE
A1471   PZE     0
        BCI     1,      I
        BCI     1,I
        BCI     1,
        BCI     1,   I
        BCI     1,
        BCI     1,      I
        BCI     1,I
        BCI     1,
        BCI     1,   I
        BCI     1,
        BCI     1,      I
        BCI     1,I
        BCI     1,
        BCI     1,   I
        BCI     1,
        BCI     1,      I
A1512   BCI     1,I
A1513   BCI     1,
A1514   BCI     1, 0
A1515   PZE     0
        BSS     18
A1540   BSS     1
A1541   BSS     1
A1542   PZE     0
A1543   PZE     0
A1544   BCI     1,CH0000
```

```
A1545  PZE  0
       BCI  1,----+
       BCI  1,+-----
       BCI  1,------
       BCI  1,--+---
       BCI  1,------
       BCI  1,----+-
       BCI  1,+-----
       BCI  1,------
       BCI  1,--+---
       BCI  1,------
       BCI  1,----+-
       BCI  1,+-----
       BCI  1,------
       BCI  1,--+---
       BCI  1,------
       BCI  1,----+-
A1566  BCI  1,+-----
A1567  PZE  0
A1570  PZE  0
A1571  BCI  1,-
       BCI  1,--
       BCI  1,---
       BCI  1,----
       BCI  1,-----
A1576  BCI  1,------
A1577  OCT  10
A1600  BCI  1,0PLOT1
       BCI  1,0PLOT2
A1602  BCI  1,0NO PR
       BCI  1,EVIOUS
       BCI  1, PLOT2
A1605  PZE  0
A1606  PZE  0
A1607  OCT  201400000000
A1610  BCI  1,+D0000
A1611  BCI  1,+F0000
A1612  BCI  1,+)0000
A1613  BCI  1,+N0000
A1614  OCT  3
A1615  PZE  0
A1616  OCT  3
A1617  PZE  0
A1620  OCT  16
A1621  OCT  11
A1622  BCI  1,00001P
A1623  PZE  0
A1624  BCI  1,I
A1625  BCI  1,+
A1626  PZE  0
A1627  OCT  4
A1630  PZE  0
```

```
A1631   PZE    0
A1632   PZE    0
A1633   BCI    1,00000S
A1634   OCT    1
A1635   OCT    777777777
        OCT    777777777700
        OCT    777777770077
        OCT    777777007777
        OCT    777700777777
        OCT    770077777777
A1643   OCT    7777777777
A1644   OCT    770000000000
A1645   BCI    1,+40000
A1646   BCI    1,00000N
A1647   OCT    200772702436
A1650   OCT    11
A1651   PZE    0
A1652   PZE    0
A1653   PZE    0
A1654   PZE    0
A1655   OCT    5
A1656   OCT    12
A1657   OCT    1
A1660   OCT    1000000
A1661   PZE    0
A1662   PZE    0
A1663   BCI    1,
        OCT    606060606072
A1665   BCI    1,00001Y
A1666   OCT    12
A1667   OCT    12
A1670   PZE    0
A1671   PZE    0
A1672   OCT    6
A1673   OCT    203600000000
A1674   PZE    0
A1675   PZE    0
A1676   TCOA   **
A1677   PZE    0
A1700   PZE    0
A1701   PZE    0
A1702   OCT    12
A1703   BCI    1,00001M
A1704   BCI    1,00001N
A1705   BCI    1,00001O
A1706   OCT    3
A1707   OCT    134537657770
A1710   OCT    233575360400
A1711   OCT    14
A1712   BCI    1,00000D
A1713   OCT    2
A1714   PZE    0
```

```
A1715   PZE      0
A1716   BCI      1,00000A
A1717   BCI      1,OPLOT2
        BCI      1, IMPRO
        BCI      1,PER AR
        BCI      1,GUMENT
A1723   PZE      0
A1724   PZE      0
A1725   OCT      1
        PZE      0
A1727   PZE      0
A1730   PZE      0
A1731   PZE      0
A1732   PZE      0
A1733   PZE      0
A1734   PZE      0
A1735   BSS      1
        BSS      29
A1773   PZE      0
A1774   PZE      0
        EXTERN   JOBOU
        END
A714    LDQ      ,2
A715    SXD      A1440,2
        SXD      A416,1
A717    SXD      A1212,4
        LXD      A762,4
        SLW      S4
        ORA      A1541,4
        SLW      A1541,4
        CAL      S4
        STQ      A1632
        TIX      A700,4,1
        SXA      A1773,0
        LXD      A1066,4
        LXD      A70,2
A727    CLA      A1541,4
        SXA      A1774,4
        LXA      A1773,4
        STO      A1735,4
        TXI      *+1,4,-1
        SXA      A1773,4
        LXA      A1774,4
        CLA      A1540,4
        SXA      A1774,4
        LXA      A1773,4
        STO      A1735,4
        TXI      *+1,4,-1
        LXA      A1716,1
A744    CLA      ,2
        STO      A1735,4
        TXI      *+1,4,-1
```

```
         TXI     *+1,2,1
         TIX     A744,1,1
         PXA     ,4
         PAC     ,4
         SXD     A756,4
         AXT     -*-1,4
         TRA     A223
A756     PZE     A1735
         SXA     A1773,0
         LXA     A1774,4
         TIX     A727,4,1
A762     MTH     A621
A763     CAL     A1626
         ANA     A1657
         TNZ     A1054
         LXD     A154,4
         CLA     A1513
         STO     A1543,4
         TIX     *-1,4,1
         CLA     A1657
         STO     A1545
         LDQ     A1713
         CLA     A1614
         TZE     A1001
         TPL     A1001
         STQ     A1545
A1001    CLA     A1730
         STO     A1471
         CLA     A1615
         STO     A1653
         CLA     A1620
         STO     A1652
         CLA     A1614
         STO     A1651
         TSX     A1156,4
         CLA     A1657
         STO     A1623
A1014    CLA     A1623
         TSX     A1067,2
         LRS     35
         FMP     A1567
         FAD     A1730
         STO     A1471
         CLA     A1545
         ADD     A1657
         STO     A1545
         CLA     A1621
         STO     A1652
         TSX     A1156,4
         CLA     A1623
         ADD     A1657
         STO     A1623
```

```
         SUB    A1656
         TZE    A1014
         TMI    A1014
         AXT    -*-1,4
         TRA    A223
         PZE    A1513,,1
         AXT    20,2
         LXD    A1734,1
         AXT    ,4
A1044    CLA    A1542,1
         STO    A1735,4
         TXI    *+1,4,-1
         TXI    *+1,1,1
         TIX    A1044,2,1
         AXT    -*-1,4
         TRA    A223
         PZE    A1735,,20
A1054    RETURN FPLOT4
A1056    TZE    1,2
         STO    A1701
         SSP
         SUB    A1635
         TMI    A1065
         CLA    A1701
         TRA    1,2
A1065    TXL    A532,2,**
A1066    MTH    A411
A1067    ORA    A1544
         FAD    A1544
         TRA    1,2
A1072    SSP
A1073    LRS    33
         TNZ    A1101
         LLS    15
         TNZ    1,2
         LLS    18
         TRA    1,2
A1101    LLS    33
A1102    UFA    A1544
         LRS    27
         ZAC
         LLS    27
         TRA    1,2
OMIT     SAVE   1,2,4
         CLA*   3,4
         SXA    A1114,2
         TSX    A1073,2
A1114    AXT    ,2
         TMI    A1120
         SLW    S5
         ORA    A1626
         SLW    A1626
```

```
         CAL      S5
         RETURN   OMIT
A1120    COM
         SLW      S6
         ANA      A1626
         SLW      A1626
         CAL      S6
         RETURN   OMIT
A1123    CLA      1,4
         STA      A1147
         CLA      2,4
         STA      *+1
         LDQ      **
         STQ      A1677
         ZAC
         DVP      A1672
         STQ      A1700
         LXA      A1700,2
A1135    MPY      A1672
         STQ      A1700
         CLA      A1677
         SUB      A1700
         PAX      ,1
         CLA      3,4
         STA      A1146
         STA      A1154
         CAL      A1643,1
A1146    ANA      ,2
         SLW*     *-1
A1147    CAL      **
         ANA      A1644
         TNX      A1154,1,6
         ARS      6
         TIX      *-1,1,1
A1154    ORA      ,2
         SLW*     *-1
         TRA      4,4
A1156    SXD      A1316,4
         CLA      A1653
         SSP
         SUB      A1646
         TMI      A1165
         CLA      A1607
A1164    TXI      A1167,,-4096
A1165    CLA      A1653
         TSX      A1443,2
A1167    STO      A1701
         TOV      *+1
         LDQ      A1701
         SUB      A1164
         STO      S7
         CLA      A1471
```

```
        SSP
        ADD     S7
        TOV     A1177
        SUB     A1135
        TPL     A1200
A1177   LDQ     A1607
A1200   FMP     A1471
        STO     A1733
        SSP
        CAS     A1707
        TRA     A1206
        TRA     *+1
        STZ     A1733
A1206   SUB     A1607
        TPL     A1213
        CLA     A1651
        ADD     A1706
A1212   MTH     A1215
A1213   CLA     A1651
        ADD     A1712
A1215   SUB     A1652
        TPL     A1227
        ADD     A1652
        LDQ     A1652
        STQ     A1701
        STO     A1652
        CLA     A1701
        ADD     A1545
        SUB     A1652
        STO     A1545
A1227   CLA     A1652
        TZE     A1441
        SUB     A1651
        SUB     A1713
        TSX     A1443,2
        STO     A1661
        CLA     A1513
        STO     A1671
        CLA     A1733
        TPL     A1245
        SSP
        STO     A1733
        CLA     A1571
        STO     A1671
A1245   LDQ     A1607
        CLS     A1651
        TPL     A1252
        TSX     A1443,2
        LRS     35
A1252   FMP     A1645
        FAD     A1733
        STO     A1733
```

```
        CLA     A1657
        STO     A1515
        CLA     A1652
        STO     A1670
A1261   SUB     A1651
        SUB     A1657
        TNZ     A1317
        CLA     A1663
        STO     A1543
        CLA     A1515
        TZE     A1315
        CLA     A1670
        SUB     A1652
        TPL     A1315
        CLA     A1545
        SUB     A1657
        STO     A1701
        TSX     A1123,4
        PTH     A1672
        PTH     A1701
        PTH     A1542
        CLA     A1670
        ADD     A1657
        SUB     A1652
        TPL     A1315
        CLA     A1545
        SUB     A1713
        STO     A1701
        TSX     A1123,4
A1312   PTH     A1671
A1313   PTH     A1701
A1314   PTH     A1542
A1315   STZ     A1515
A1316   MTH     A1413
A1317   CLA     A1661
        STO     A1662
        CLA     A1733
        FDP     A1661
        STQ     A1701
        CLA     A1701
        SUB     A1710
        TMI     A1350
A1327   LDQ     A1662
        FMP     A1710
        STO     A1662
        CLA     A1701
        FDP     A1710
        STQ     A1701
        CLA     A1710
        CAS     A1701
        TRA     A1340
        TRA     *+1
```

```
         MTH    A1327
A1340    CLS    A1701
         TSX    A1102,2
         TSX    A1067,2
         LRS    35
         FMP    A1662
         FAD    A1733
         STO    A1733
         MTH    A1317
A1350    CLA    A1701
         TSX    A1072,2
         STO    A1543
         TSX    A1067,2
         LRS    35
         FMP    A1661
         CHS
         FAD    A1733
         STO    A1733
         CLA    A1661
         FDP    A1613
         STQ    A1661
         CLA    A1650
         SUB    A1543
         TPL    A1374
         STZ    A1515
         ZAC
         LDQ    A1543
         DVP    A1702
         STO    A1543
A1374    CLA    A1515
         TZE    A1413
         CLA    A1543
         TZE    A1422
         CLA    A1670
         SUB    A1652
         TPL    A1412
         CLA    A1545
         SUB    A1657
         STO    A1701
         TSX    A1123,4
         PTH    A1671
         PTH    A1701
         PTH    A1542
A1412    STZ    A1515
A1413    LDQ    A1543
         RQL    30
         STQ    A1701
A1416    TSX    A1123,4
         PTH    A1701
         PTH    A1545
         PTH    A1542
A1422    CLA    A1545
```

```
        ADD     A1657
        STO     A1545
        CLS     A1657
        ADD     A1670
        STO     A1670
        TPL     A1261
        CLA     A1515
        TZE     A1441
        CLA     A1545
        SUB     A1657
        STO     A1545
        STZ     A1701
        STZ     A1515
A1440   MTH     A1416
        PXD     128,0
        LDQ     A1623
        DVP     A1666
        STQ     A1631
        TNZ     A667
        CAL     A1626
        ANA     A1713
        TNZ     A667
        CLA     A1631
        TSX     A1067,2
        LRS     35
        FMP     A1570
        STO     A1701
        CLA     A1731
        FSB     A1701
        STO     A1471
        CLA     A1713
        STO     A1545
        CLA     A1650
        STO     A1652
        SXD     A762,4
        TSX     A1156,4
        LXD     A762,4
A667    CLA     A1542
        STO     A1541,4
        CLA     A1541
        STO     A1540,4
        CLA     A1623
        ADD     A1657
        STO     A1623
        TIX     A634,4,1
        LXD     A1066,4
A700    SXD     A762,4
        LXD     A1212,4
        CAL     A1676
        TNX     A717,4,1
        LXD     A1440,2
        LXD     A416,1
```

```
        LDQ     A1632
        LGL     6
        ALS     24
        TIX     A715,1,1
        LXA     A1672,1
A713    TXI     *+1,2,**
1
A1441   LXD     A1316,4
        TRA     1,4
A1443   LDQ     A1607
        STQ     A1662
        PAX     ,4
        TXH     A1451,4,0
        CLA     A1607
        TRA     1,2
A1451   TMI     A1457
A1452   LDQ     A1662
        FMP     A1613
        STO     A1662
        TIX     A1452,4,1
        TRA     1,2
A1457   CLA     A1662
        FDP     A1613
        STQ     A1662
        TIX     A1457,4,1
        CLA     A1662
        TRA     1,2
        OCT     77777
TEMP    PZE     0
        PZE     1
Y1      BSS     1
S7      PZE
S6      PZE
S5      PZE
S4      PZE
S3      PZE
S1      PZE
XZ      PZE
XY      PZE
```

APPENDIX

REFERENCE MANUAL

(SPACE)

REFERENCE MANUAL

SUBSYSTEM PROCESSOR FOR THE APOLLO COMPUTING EFFORT

(SPACE)

1 February 1965

Prepared by:

Robert G. Hansen

Apollo Support Department
General Electric Company
Daytona Beach, Florida

SUBSYSTEM PROCESSOR FOR THE APOLLO COMPUTING EFFORT

REFERENCE MANUAL

TABLE OF CONTENTS

Paragraph		Page
	SECTION 1 - GENERAL INFORMATION	
1.1	INTRODUCTION TO THE MONITOR	1-1-01
1.2	HISTORY AND DEVELOPMENT	1-2-01
	SECTION 2 - PROGRAMMER'S MANUAL	
2.1	WRITING A PROGRAM	2-1-01
2.2	DEBUGGING FACILITIES	2-2-01
2.3	RELATED PROGRAMMER INFORMATION	2-3-01
	SECTION 3 - USER'S MANUAL	
3.1	PREPARING THE REELS CARD SERIES	3-1-01
3.2	CREATING THE LIBRARY	3-2-01
3.3	THE TYPEWRITER LISTING	3-3-01
3.4	RELATED USER INFORMATION	3-4-01
	SECTION 4 - RELATED SUBROUTINES	
4.1	GENERALIZED SORTING	4-1-01
4.2	BCD OUTPUT VIA GGOUT	4-2-01
4.3	UTILITY ROUTINES	4-3-01

SECTION 1

GENERAL INFORMATION

TABLE OF CONTENTS

Paragraph		Page
1.1	INTRODUCTION TO THE MONITOR	1-1-01
1.2	HISTORY AND DEVELOPMENT	1-2-01

SECTION 1

GENERAL INFORMATION

1.1 INTRODUCTION TO THE MONITOR

SPACE is an executive, or administrative, program operating as a subset of the 7040/7044 IBJOB Processor Monitor (#7040-SV-811). The framework of SPACE is centered around a collection of seven versatile I/O subroutines. By using these subroutines, the programmer can disassociate himself completely from problems such as the 'physical' aspects of retrieving or creating externally stored data, blocking/unblocking logical records, file positioning, synchronized CPU/channel overlap, and the differences in the characteristics of recording devices. Thus, he is permitted to concentrate on his primary task--the internal processing of data.

A basic requirement of any executive monitor is to automate the running of a series of data-processing programs by calling these programs from a library tape as they are needed. This requirement necessitates the SPACE user to create his own library by employing the chain feature of IBLDR. The only requisite conditions imposed upon the link structure of this library is that SPACE be the main link, with the subsidiary programs acting as dependent links. One such dependent link is a post-execution file utility processor which must be included in the library. It may, for all purposes, be considered as a part of the monitor, only in core when needed.

Enhanced by these facilities, the objectives of the monitor may be outlined, as follows:

a. To maintain semi-compatibility with programs written for the now defunct DSDPS (Defense Systems Data Processing System).

b. To create, if possible, up to 999 files of data on a given I/O device, with the ability to randomly access any of these files.

c. To enable 'data-sharing' capabilities whereby files of output data from any program(s) can serve as input data to any later program(s), either within the same job or not.

d. To provide a framework around which systems of data-processing programs can be developed. Once data is available in the standard SPACE file format, the whole range of previously written programs is available to process it.

1.2 HISTORY AND DEVELOPMENT

The history of data processing in ASD began in July of 1963 with the installation of an IBM 7094. Mr. E. E. Johnson's engineering programming group assumed the task of writing sufficient programs which, using the Monte Carlo simulation technique, enabled the calculation of the probability of success for various flight regimes of the Apollo Mission. The effort was dubbed SOAR III (Simulation of Apollo Reliability) and, headed by H. N. Lerman, grew into a complex system of approximately thirty functionally dependent programs. The apparent need of a monitoring system forced the use of DSDPS to provide a tape library of SOAR programs, and to handle communications between these programs. Although the operating philosophy of DSDPS was no less than excellent, it left much to be desired in other respects. The most stringent shortcomings were threefold--(a) programs had to be assembled in absolute mode and rely upon fixed absolute locations, thus virtually inhibiting interface capabilities with relocatable subroutine libraries, assemble-and-go facilities, compiler languages[1], etc., (b) the monitor itself had to be adopted to IBSYS and maintained by the installation, thus causing the implementation of new version software releases to be indeed 'painful', and (c) DSDPS was not suitably designed for reliability data-processing applications.

This led to the advent of SPACE, written by R. G. Hansen in the early fall of 1964, operating as a subset of 7090/7094 IBJOB under IBSYS. The new monitor was accepted with enthusiasm by most ASD programming personnel and immediate action was taken to begin converting all active DSDPS programs in the SOAR series.

Soon after the completion of 7090/7094 SPACE, a decision was made to replace the 7094 with a 7044. The structural differences in software design of the two computers necessitated a complete rewrite of the monitor. The 7040/7044 version of SPACE is a highly flexible system and features increased capabilities over the 7090/7094 version, at a reduction of core storage requirements.

---

[1] A means was found by which DSDPS could utilize the FORTRAN II language; however, the resultant programs ran quite inefficiently.

SECTION 2

PROGRAMMER'S MANUAL

TABLE OF CONTENTS

Paragraph		Page
2.1	WRITING A PROGRAM	2-1-01
2.1.1	THE SPACE COMMUNICATION NUCLEUS	2-1-01
2.1.2	JOB DECK ORGANIZATION	2-1-02
2.1.3	INTER-PROGRAM COMMUNICATION BY DATA FILES	2-1-04
2.1.4	FORMAT OF THE STANDARD DATA FILE	2-1-05
2.1.5	THE FILE-PROCESSING SUBROUTINES	2-1-06
2.1.6	PERMISSIBLE SEQUENCES OF FILE OPERATIONS	2-1-11
2.1.7	FILE PROCESSING ERROR DIAGNOSTICS	2-1-12
2.1.8	THE SYSTEM REEL/FILE TABLES	2-1-15
2.1.9	THE BCD OUTPUT EDITOR	2-1-17
2.1.10	RETURNING TO THE MONITOR	2-1-18
2.1.11	A PROGRAMMING EXAMPLE	2-1-19
2.2	DEBUGGING FACILITIES	2-2-01
2.2.1	THE DUMP PROGRAM	2-2-01
2.2.2	THE POST-EXECUTION FILE UTILITY PROCESSOR	2-2-02
2.2.3	ABSOLUTE BINARY PATCHING	2-2-03
2.2.4	USING THE LOGIC AND MEMORY MAP	2-2-05
2.3	RELATED PROGRAMMER INFORMATION	2-3-01

SECTION 2

PROGRAMMER'S MANUAL

2.1 WRITING A PROGRAM

The following paragraphs describe the organization and operating philosophy of the monitor. Although some of the material presented here is user oriented, it is included to provide a clear understanding of SPACE programming procedures.

2.1.1 THE SPACE COMMUNICATION NUCLEUS

Two individual named common blocks contained within SPACE provide all necessary communications between the monitor and the executing programs. The control section names of these blocks are 'ACCESS' and 'SYSTEM'. Individual FORTRAN coded programs must contain either, or both, of the following COMMON statements in order to reference these areas:

```
COMMON/ACCESS/HC(100),WDCT,ID(12),PROG

COMMON/SYSTEM/NTAPES,REELS(15),CNTRLS(15),FILES(15),LRS(15),
    POS(15),TRLPOS(15),RWCNT(15),UNITS(15)
```

Individual MAP coded programs must contain either, or both, of the following code sequences:

```
Sequence 1:   HC        BSS       100
              WDCT      BSS       1
              ID        BSS       12
              PROG      BSS       1
              ACCESS    CONTRL    HC,*
```

Sequence 2:

```
NTAPES   BSS      1
REELS    BSS      15
CNTRLS   BSS      15
FILES    BSS      15
LRS      BSS      15
POS      BSS      15
TRLPOS   BSS      15
RWCNT    BSS      15
UNITS    BSS      15
SYSTEM   CONTRL   NTAPES,*
```

These blocks are maintained by the monitor and reflect information related to the current program in execution, and also an up-to-date history of status information pertaining to I/O file activity within the entire job.

2.1.2 JOB DECK ORGANIZATION

The job deck for a SPACE application consists of all data cards up to, but not including, the $IBSYS card. Consider a data card series as a group of one or more cards, the last of which contains an asterisk (*) in columns 7-72, inclusive. Individual cards comprising the series, including the terminal (*) card, may contain data items (subsequently referred to as 'words') punched in these columns in accordance with the following rules:

a. A given word will be classified as and converted to a BCD, integer, or floating point quantity, as characterized by the appearance of a letter, the absence of a decimal point, and the appearance of a decimal point, respectively.
b. Words are separated from one another by one or more intervening blank card columns.
c. A word cannot be continued from one card to the next.

The job deck must always begin with a card series describing information such as each reel of magnetic tape which will be needed for the job, the device to which the reel should be assigned, and, among other things, the file numbers of certain files which are to be processed by the post-execution file utility program. This special program, in addition to performing all housekeeping functions at the end of the job, can dump selected records from generated files in a variety of formats, and can also copy a given file from one device to another. Card columns 1-5 of the first card in this series must contain the identification word 'REELS'.

The remainder of the job deck consists of program control cards interspersed with data cards. The program control cards comprise a series and are read by the monitor. Columns 1-6 of the first card in such a series must contain the name of the next program which is to be loaded and executed. The name must be left-adjusted and must be identical to the name given the dependent link to which the program belongs; i.e., the name appearing in columns 8-13 of the $LINK card. When the program control card series is read by the monitor, the following actions are taken:

a. The program (link) is located in the library and loaded into core storage.
b. The name of the program is placed in location PROG.
c. Program control parameters which were read from the series are placed into the HC block, and the number of such parameters read is placed in the address portion of location WDCT.
d. Control is given to the program via a TRA instruction.

Data cards to be read by the program may follow the control card series; however, caution must be exercised to avoid accidental reading of the next program's control cards in the job deck. This implies that the program must have a means of recognizing the end of its associated data. If, for some reason, a program fails to read all of its data before returning to the monitor, the job deck will be searched until the next control card series is found, or a card with a dollar sign ($) in column one is encountered, which automatically terminates the job.

Thus, any of the programs in the SPACE library may be run in any order, as many times as desired.

2.1.3 INTER-PROGRAM COMMUNICATION BY DATA FILES

Central to the design of SPACE is a collection of seven versatile binary I/O subroutines. These subroutines enable programs to read or write up to fifteen files of data, simultaneously. The basic unit of information within such a file is termed a logical record. Several logical records grouped together constitute a block, or physical record on the recording medium, and all such blocks grouped together constitute the data file itself. Every data file has a unique number associated with it, termed the file number, and programs communicate with the files by these numbers alone. The uniqueness is effected by specifying the reel number[1] (R) upon which the file resides, together with the position (P) of the file relative to load point. The file number is then expressed as an integer of the form, 1000R+P. For example, the file number 387002 reflects the second file of data contained on reel 387.

The discussion so far has implied the use of tape reels which have been assigned to a particular individual for his exclusive use. The reels must, of course, be mounted at the beginning of his job and must be terminally removed to prevent possible destruction by subsequent jobs. This philosophy is an excellent one provided that all data files which were created in the job were of a permanent nature; i.e., files which would subsequently be used for input at a later date. Many applications, however, require the use of mediary, or scratch files. These files are to be used only within the job and need not be saved. Since only one file per reel can be in an active state at any one time, it follows if at some point a requirement exists to read or write several mediary files simultaneously, an appropriate number of reels would have to be mounted at the outset of the job to accommodate these files. This time consuming and costly procedure can be alleviated by specifying the reel(s) as mediary in the tape assignment card series. A mediary reel number consists of a negative integer, the magnitude of which is less than 32,768. This number serves only as a reference when specifying file numbers; hence, the choice of such

[1]The reel number is an integer code assigned to a reel when it enters an installation. This number appears on the outer surface of the reel for visual identification.

is left to the discretion of the user[1]. The term 'reel' takes on a different connotation when referring to mediary reels. In this case, the 'reel' may be sequential tracks of disk or drum storage, or a reel of magnetic tape. From the programmer's and user's point of view, however, all of the devices may be treated as magnetic tape. A mediary reel, then, consists of an available device upon which data files can be recorded and read only by the programs used within a given job. Unless otherwise specified, the monitor will attempt to assign these reels to devices which are in ready status, thus eliminating the need of manual intervention to mount and remove tapes.

2.1.4 FORMAT OF THE STANDARD DATA FILE

The standard SPACE data file consists of a twelve-word identification record followed by any number of data blocks, the last block of which is followed by an end of file mark. The content of the identification record is as follows:

Word 1 - A label identifier, consisting of the BCD word 'FILEID'.

Word 2 - The file number of the file.

Word 3 - The date upon which the file was created, formated in BCD as MMDDYY.

Word 4 - The BCD name of the program which created the file.

Word 5 - An integer, right adjusted, indicating the size of logical records within the file.

Word 6 - The address of this word contains a count of all physical records constituting the previous data file. It is used by the I/O routines for file positioning purposes.

Words 7-12 of the identification record are available for use by the programmer, should he elect to place additional information about the file into the label. The contents of these six words are never altered or disturbed in any way by the file processing routines, except when a file is opened for reading.

---

[1] Some programs depend upon fixed mediary file numbers. The user should consult program descriptions to determine which mediary reel numbers, if any, must be supplied in the tape assignment card series for the job.

Physical data records, or blocks, within the file are 257 words in length, with the exception of the last such block, which may be less than 257 words. Two dummy words are provided at the beginning of each block to prevent it from being noise (containing less than three words) and, in addition, a word is appended to the end of the block which serves internal validity checking purposes. Before the block is written, its sequence number is placed in the address of this word. A check sum is then formed by computing a logical sum of the entire data in the block. The left and right halves of this sum are added together, and the resultant 'folded' check sum is placed in the left half of the word. Whenever the block is read, a folded check sum is computed and compared against the left half of the last word in the block. In addition, the current block sequence count is compared against the address of this word. An unequal compare, in either case, is considered an unrecoverable error condition. These words are handled internally by the processing routines only and will never, under any circumstances, be transmitted to the user as data.

As mentioned earlier, the last data block in a file is followed by an end of file mark. An additional file mark is placed at the end of all usable files on the reel which serves as a trailer label for the reel. If the trailer label is encountered by the processing routines while attempting to position at a file upwards on the reel, and positioning requirements have not yet been satisfied at the point of encounterment, an unrecoverable error condition occurs and the user is informed of his attempt to access a nonexistent file.

2.1.5 THE FILE-PROCESSING SUBROUTINES

Seven subroutines enable SPACE programs to handle data files by file number alone. With these facilities, programs can read, write, skip, and backspace logical records, or write an end of file mark simply by calling the appropriate routine. The following definitions apply to CALL prologue arguments which must be supplied to the various subroutines when they are used:

F = A location containing the file number of the file which is to be acted upon.

LRS = A location containing the size, or length, of logical records within the file, in terms of words. It must be a positive, right-adjusted integer less than 255.

FWA = The first word address of the array, or block, into or from which logical records are to be transmitted.

NREC = A location containing the number of logical records which are to take part in the activity requested by the CALL. It must be a positive, right-adjusted integer.

HISTRY= A location which contains a history of operation performed by subroutines READB2, SKIPR, and BSPR. Each of the routines will update this location before returning to the calling program.

2.1.5.1 SUBROUTINE READB1

Before data can be read from a given file, the file must have been previously activated, or opened, by calling READB1 as follows:

CALL READB1(F,LRS)

When this call is executed, the following events take place:

a. If any file is currently active on the reel specified by F, it is closed by calling ENDF.
b. The reel is positioned to the data file specified by F, and the identification record of the file is read into locations ID through ID+11.
c. Call arguments F and LRS are compared with C(ID+1) and C(ID+4), respectively, to verify positioning success. The label identifier contained in C(ID) is additionally verified.
d. All housekeeping functions are performed to permit subsequent use of READB2 for reading the file.

It is conceivable that the LRS specification of a given file will not be known when it is desired to open the file with READB1. Although such a situation should be avoided, the file may be opened by setting the C(LRS) to zero before calling READB1. At return, the C(ID+4) can be inspected to determine the LRS of the file.

2.1.5.2 SUBROUTINE READB2

This subroutine is used to read any number of logical records from a file which has previously been activated by READB1. Linkage to READB2 is as follows:

CALL READB2(F,FWA,NREC,HISTRY)

When this call is executed, READB2 will begin reading the next NREC logical records from file F, each of size LRS, into the transmission area specified by FWA. The first such record will be transmitted to locations FWA through FWA+LRS-1, the second to locations FWA+LRS through FWA+2*LRS-1, etc., until NREC logical records have been transmitted, or the end of file is encountered, whichever occurs first. At the end of the operation, READB2 will place a right-adjusted integer into location HISTRY which reflects the total number of words transmitted by the request. Furthermore, if the end of file was encountered before NREC logical records were transmitted, the sign of location HISTRY will be set negative and the file will become inactive. It cannot be referenced again until it is opened by READB1 (or ABOUT1).

2.1.5.3 SUBROUTINE ABOUT1

Before data can be written in a given file, the file must have been previously activated, or opened, by calling ABOUT1 as follows:

CALL ABOUT1(F,LRS)

when this call is executed, the following events take place:

a. If any file is currently active on the reel specified by F, it is closed by calling ENDF.
b. The reel is positioned as specified by F.
c. The first six words of the ID block are prepared and the identification record is written from locations ID through ID+11.
d. All housekeeping functions are performed to permit subsequent use of ABOUT2 for writing the file.

Note that words 7-12 of the identification record are reserved for use by the programmer. If he elects to use any of these words, he must prepare them in the ID block prior to calling ABOUT1.

2.1.5.4 SUBROUTINE ABOUT2

This subroutine is used when it is desired to write logical records in a file which has previously been activated by ABOUT1. Linkage to ABOUT2 is as follows:

CALL ABOUT2(F,FWA,NREC)

When this call is executed, ABOUT2 will write NREC logical records into file F, each of size LRS, from the transmission area specified by FWA. Thus, a total of NREC*LRS contiguous words will be transmitted to the file from locations FWA through FWA+NREC*LRS-1.

2.1.5.5 SUBROUTINE SKIPR

In some applications, it may be necessary to start processing a file at some logical record other than the first, or perhaps only every n^{th} record of the file is to be processed. In either case, the SKIPR routine can be employed to pass over, or ignore, the unwanted records. Linkage must be as follows:

CALL SKIPR(F,NREC,HISTRY)

The call to SKIPR causes the next NREC logical records contained in file F to be skipped over, unless the end of file is encountered before the request is satisfied. In either case, SKIPR will place a count of the number of words skipped into location HISTRY. Furthermore, if the operation was ended due to encountering the end of file, the sign of location HISTRY will be set negative. The file will not, however, be made inactive. This is an important distinction between READB2 and SKIPR. For example, if it is desired to start processing at the last logical record in a file and work backwards, the following subroutines would be used in the prescribed order:

READB1	To activate the file.
SKIPR	Until the end of file is reached.
BSPR	To backspace one logical record.
READB2	To obtain the last record.
BSPR	To backspace two logical records.
READB2	To obtain the next-to-last record.
etc.	

A check must, of course, be made after each call to BSPR to determine if the beginning of file has been reached.

2.1.5.6 SUBROUTINE BSPR

This subroutine is used to backspace over logical records in a file which has been previously activated by READB1. Linkage is as follows:

CALL BSPR(F,NREC,HISTRY)

This call results in backspacing over NREC logical records in file F which have already been processed by READB2 or SKIPR, unless the beginning of the file is encountered before the BSPR request is satisfied. In either case, the BSPR subroutine will place a count of the number of <u>words</u> backspaced into location HISTRY. If the operation was ended due to encountering the beginning of file, the sign of location HISTRY will be set negative, and a subsequent call to READB2 or SKIPR will begin processing with the first logical record of the file.

2.1.5.7 SUBROUTINE ENDF

This subroutine is used to close, or inactivate, a given file. Subsequent references cannot be made to the file until it is re-opened by READB1 or ABOUT1. Linkage is as follows:

CALL ENDF(F)

If file F is being written, data in the current buffer used by the file is written out, and followed by a file mark. Note that this may result in writing a physical data record less than 257 words, as the buffer may only be partially filled when ENDF is called. If file F is being read, ENDF will cause it to become inactive. If the file is already inactive, the call is ignored.

The monitor automatically calls ENDF between programs for any files which are active. In addition, ENDF will be called by subroutines READB1 and ABOUT1 if any file is currently active on the reel they are to use. Hence, it is frequently unnecessary for the programmer to employ ENDF. Exceptional circumstances might include:

a. The case in which there would otherwise be more files active than buffers to accommodate them. A file which is active requires the exclusive use of two buffers. When a file becomes inactive, its buffers are returned to a 'pool' so that they may be used by some other file. This pool is located near the top of core storage and normally contains twice as many buffers as there are reels for the job. If after loading a program, however, the monitor discovers that it extends into the pool, an appropriate number of buffers will be inhibited, or made unavailable for use by active files within that program. Hence, very large programs should call ENDF for files which remain unnecessarily active to free the buffers for other activity; e.g., all output files, and partially read input files.

b. When the number of buffers to be assigned to the pool has been specified in the REELS card series, and is less than the standard number.

2.1.6 PERMISSIBLE SEQUENCES OF FILE OPERATIONS

Certain combinations of reading and writing files on a given reel are illegal. In the table that follows, R_c implies that the current (or last) use of the reel was for input from file C, R_p implies that the next use of the reel is to be for input from a previous file, i.e., one physically preceding file C, and R_s implies that the next use of the reel is to be input from a subsequent file; i.e., one physically following file C. Similar definitions apply to W_c, W_p and W_s for output files.

OPERATION	POSSIBLE	REMARKS
$R_c R_p$	Yes	Always permissible.
$R_c R_s$	Yes	Provided file S exists.
$R_c R_c$	Yes	Always permissible.
$R_c W_p$	Yes	All files following P will become nonexistent.
$R_c W_s$	Yes	Provided that a file immediately preceding file S exists.
$R_c W_c$	Yes	Always permissible.
$W_c R_p$	Yes	Always permissible.
$W_c R_s$	No	When file C is written, it is considered as the last file currently contained on the reel.
$W_c R_c$	Yes	Always permissible.
$W_c W_p$	Yes	All files following P will become nonexistent.
$W_c W_s$	Yes	Only if file S immediately follows file C.
$W_c W_c$	Yes	Always permissible.

2.1.7 FILE PROCESSING ERROR DIAGNOSTICS

When any of the file processing routines are called upon, several checks are made to insure that the requested operation is legal. If for any reason an error condition is indicated, an appropriate diagnostic is given, accompanied with a terminal dump of the program area. The diagnostic will always include the name of the subroutine called, the absolute octal location of the CALL, and also its internal formula number (IFN), providing one exists.

The error comments, together with the conditions which cause the error are given below. Asterisks denote a quantity which will be supplied.

2.1.7.1 <u>CALL SPECIFIES IMPROPER NUMBER OF PARAMETERS</u>.

The number of parameters, or arguments, supplied to subroutines READB1, READB2, ABOUT1, ABOUT2, SKIPR, BSPR and ENDF must be 2, 4, 2, 3, 3, 3 and 1, respectively.

2.1.7.2 <u>CALL REFERENCES INACTIVE OR NONEXISTENT FILE ******</u>.

This diagnostic is given whenever (a) the reel number R, or the position P implied by the file number is zero, (b) R is not specified in the REELS card series, or (c) the file was inactive when a call was made to READB2, ABOUT2, SKIPR, or BSPR.

2.1.7.3 <u>FILE ****** CANNOT BE OPENED DUE TO INSUFFICIENT BUFFERS</u>.

Subroutine READB1 (or ABOUT1) has discovered that there are not two buffers available for opening the given file.

2.1.7.4 <u>CALL INDICATES LRS = *** FOR FILE ******</u>.

The LRS specification given to subroutines READB1 or ABOUT1 is greater than 254, or zero if the CALL was to ABOUT1.

2.1.7.5 <u>LRS IN FILE ID = ***</u>.

This is appended to the above diagnostic if READB1 discovers that the LRS specification given in the call differs with the LRS contained in the identification record of the file.

2.1.7.6 <u>CALL REFERENCES FILE PROTECTED REEL ******</u>.

The reel upon which ABOUT1 is to open a file has been logically file protected and hence, cannot be written on.

2.1.7.7 FILE ****** IS NONEXISTENT, OR CANNOT BE ACCESSED DUE TO POSITIONING FAILURE.

While positioning from one file to another, READB1 and ABOUT1 verify the identification records of all files passed over by checking the label identifier and the length of the record. Furthermore, READB1 will verify the file number in the identification record of the desired file by comparing it with the file number given in the CALL. A discrepancy in these compares will result in the above diagnostic and may be due to encountering the reel trailer label while positioning forward, or perhaps by a program still in the debug phase which has accidently moved the recording medium.

Before the actual positioning begins, certain criteria are inspected to determine the existency of the file. Existency can always be predetermined if (a) any file has been previously written on the reel within the job, or (b) the reel is mediary, or (c) the NOLABEL option was specified in the REELS card series for the reel.

2.1.7.8 CALL SPECIFIES ILLEGAL TRANSMISSION AREA.

Whenever a call is made to READB2 or ABOUT2, the transmission area into which data is to be read or from which data is to be written is checked. The first word address (FWA) of this area must not be less than the origin of the first dependent link within the job, and FWA+NREC*LRS must not be greater than S.SEND (normally 32767).

2.1.7.9 FILE ****** IS IN ***** STATUS.

Either a call has been made to ABOUT2 for a file which is currently being read, or a call has been made to READB2, SKIPR, or BSPR for a file which is being written.

2.1.7.10 EOT DETECTED ON REEL ******.

The physical end of the recording medium has been detected by ABOUT1, ABOUT2, or ENDF.

2.1.7.11 AN IOBS ERROR (CODE **) HAS OCCURRED ON REEL ******.

An error has been detected by the Input-Output Buffering System on the specified reel. The various error codes are as follows:

1 = Block sequence number error.
2 = Check sum error.
3 = Both block sequence and check sum errors.
4 = An unrecoverable read error.
5 = An attempt to write on an unopened output file.
6 = Buffer overflow (writing).
7 = Buffer overflow (reading).
8 = Unexpected mode change.
9 = An unrecoverable write error.
10 = Incomplete word read.

2.1.8 THE SYSTEM REEL/FILE TABLES

As mentioned in section 2.1.1, part of the SPACE communication nucleus consists of a named common block termed 'SYSTEM'. This block contains tables of information used by the file processing routines. Individual programs may also use any of this information, however, they must never alter it in any way. An entry within any of the eight tables bears a one-to-one correspondence with an entry in any other table; e.g., REELS(4) and LRS(4) are related. The format of an entry within each of the tables follow:

NTAPES - An address integer reflecting the total number of reels specified in the 'REELS' card series. Note that this also represents the number of entries contained in each of the eight tables.

REELS - Each entry contains the reel number of a reel being used in the job.

CNTRLS - Each entry reflects the options which were exercised for the reel in the 'REELS' card series, as follows:

bit S = 1, if the reel is logically file protected.

bit 1 = 1, if a reel header label had to be created for the reel.

bit 2 = 1, if mounting was deferred.

bit 3 = 1, if unit assignment was made by an IxxR specification.

bits 4-17 = The intersystem reservation code associated with the unit, or zero.

bits 18-20 = The index (1, 2, etc.) of the channel requested for the reel, or zero.

bits 21-35 = The first word address of the IOBS file control block assigned to the reel.

FILES - Contains the file number of the file currently active on the associated reel. If a file is not active on the reel, this entry will be zero.

LRS - Reflects the logical record size of the last file which was opened by READB1 or ABOUT1.

POS - Contains the position P implied by the file number of the last file opened by READB1 or ABOUT1. If an end of file was encountered by READB2, or written by ENDF, the sign of the entry is set negative.

TRLPOS - Reflects the position P implied by the file number of the last file which was written on the associated reel. If no files have been written on the reel within the job, the entry contains zero.

RWCNT - The decrement contains a count of all files opened by ABOUT1, and the address contains a count of all files opened by READB1.

UNITS - Contains the symbolic units table entry of the device to which the reel was assigned.

2.1.9 THE BCD OUTPUT EDITOR

In addition to the file processing subroutines contained in SPACE, the programmer has access to a flexible BCD output routine which is used by the monitor. With this routine, a given line can optionally be edited, typed, and/or written on S.SOU1. The editing process consists of eliminating superfluous blank characters from the given prototype line image; i.e., if two or more contiguous blanks appear in the prototype, all but one will be eliminated. The prototype itself will not be changed. The entry point to the subroutine is a control section termed 'OEDIT'. Note that it can only be accessed by MAP coded programs. Linkage is as follows:

```
TSX      OEDIT,4
PFX      PROTYP,SPCING,N
(RETURN)
```

where---PROTYP = The FWA of the prototype line image

SPCING = 0 for single space,
1 for page restore, or
2 for double space

(Note - SPCING is ignored for type requests)

N = The number of words in the prototype line image.

PFX = Output options, as follows:

PZE - Type with edit.
PON - Type without edit.
PTW - S.SOU1 and type with edit.
PTH - S.SOU1 and type without edit.
MZE - Edit only (at return, the AC contains PZE NEWFWA,,NEWN).
MON - No operation performed.
MTW - S.SOU1 with edit.
MTH - S.SOU1 without edit.

The use of 'OEDIT' is best shown by the following example. An integer of from 1-5 digits is to be converted to BCD, left adjusted with trailing blanks, and stored in line image 'PROTYP'. The line is then to be edited and written on S.SOU1 w/page restore.

```
            CLA     INTGER
            TSX     CNVRT,4        CONVERT NUMBER TO BCD AND
            STQ     PROTYP+3       PLACE IN LINE IMAGE.
            TSX     OEDIT,4
            MTW     PROTYP,1,6
             .
             .
             .
PROTYP      BCI     6, NETWORK CONTAINS ***** COMPONENTS.
```

Suppose the number had been 29. If the line had been written without editing, it would have appeared as:

bNETWORKbCONTAINSb29bbbbCOMPONENTS.

The above use of OEDIT, however, would produce:

bNETWORKbCONTAINSb29bCOMPONENTS.

The following restrictions and conventions apply to the use of OEDIT:

a. If the argument, N, is greater than 22, it will be reduced to 22.
b. If N is zero, the call is ignored.
c. The same prototype line may be used repeatedly. It is never altered or changed by OEDIT.
d. The entire subroutine requires 91 locations.

2.1.10 RETURNING TO THE MONITOR

When a given program has finished processing, it must return control to the monitor so that the next program, if any, can be loaded and executed. This must be accomplished via the following CALL:

CALL SPACE

When this call is executed, the monitor will read the next program control card series in the job deck, if any, and take actions as prescribed in paragraph 2.1.2.

In many instances, however, some sequences of program executions are fixed. For example, a very large program may be split into two or more smaller programs, which must be executed successively. To eliminate the need of preparing a control card series for each of these programs, a 'direct chain' technique may be employed. This enables a given program to dictate the next program to be executed, as follows:

CALL SPACE(PROGRM)

'PROGRM' is the location of a word containing the BCD name of the next program in line for execution. When the direct chain technique is used, the monitor will not read the next control card series, nor will it disturb the information contained in the HC block or location WDCT. Machine registers including the AC, MQ, XR1, XR2, and XR4 are always saved when the monitor receives control, and are restored prior to giving control to the next program. Thus, when using the direct chain, a program could pass control parameters to the next program through any of these registers. Note that an unrecoverable error will result if the monitor cannot find the requested program in the library.

2.1.11 A PROGRAMMING EXAMPLE

The following program could be used to delete specified logical records from a given input file; i.e., create a new file reflecting the deletions. To use it, the program control card series must be prepared as follows:

Columns 1-3 of first card = 99Z (program name)

Word 1 = File number of input file.

Word 2 = File number of output file.

Word 3 = Record number of first record to be deleted.

Word 4 = Record number of second record to be deleted.

.
.
.

Word N = Record number of $(N-2)^{th}$ record to be deleted, followed by an asterisk (*).

For example, if the 3rd, 17th and 80th records of file 91006 were to be deleted, and the desired output file number was 478001, then the series would look like this:

99Z 91006 478001 3 17 80 *

Our program must not assume that the record numbers are given in an ascending order. Therefore, they must be sorted before the editing process begins.

```
$LINK      99Z
$IBFTC     99Z
C          SPACE EDITING PROGRAM - 2/18/65
           COMMON/ACCESS/HC(100),WDCT,ID(12),PROG
           INTEGER HC, WDCT
           DIMENSION BUFFER(254)
           CALL SORT(HC(3), WDCT-2,1,1)
           CALL READB1(HC(1),0)
           CALL ABOUT1(HC(2),ID(5))
           I = 0
           J = 3
2          CALL READB2(HC(1),BUFFER(1),1,EOF)
           IF (EOF .NE. 0.) GO TO 1
           I=I-J+3
           WRITE (6,100) HC(2),I
100        FORMAT(19H099Z OUTPUT FILE IS,I9,
          $14H, AND CONTAINS,I6,9H RECORDS.)
           CALL SPACE
1          I=I+1
           IF(I .NE. HC(J)) CALL ABOUT2(HC(2),BUFFER(1),1)
           IF(I .EQ. HC(J)) J=J+1
           GO TO 2
           END
$ENTRY
```

2.2 DEBUGGING FACILITIES

2.2.1 THE DUMP PROGRAM

The dump is one of the most effective tools which can be used for programming entomology; i.e., locating bugs, or errors, within a program. The SPACE dump program differs from other programs in the sense that it is part of the monitor itself; however, it may be treated and used as any other program.

The control card series for the dump program must be prepared as follows:

Columns 1-4 of first card = DUMP

Words 1-3: A_1 B_1 F_1

Words 4-6: A_2 B_2 F_2

.
.
.

Words (N-2)-N: $A_{\frac{N}{3}}$ $B_{\frac{N}{3}}$ $F_{\frac{N}{3}}$ followed by an asterisk (*).

The parameters A_i and B_i represent decimal limits of areas to be dumped, inclusive. The relative ordering of these limits is immaterial, and either A_i or B_i may represent the upper or lower limit. The parameter F_i is an integer indicating the dump format desired, as follows:

F_i = 0, dump in octal (single line)

F_i = 1, dump as floating point decimal (single line)

F_i = 2, dump as integer (single line)

F_i = 3, dump in octal with mnemonics (double line)

The direct chain technique may also be employed to use the dump program, as follows:

```
CALL  SPACE(4HDUMP,A1,B1,F1...,AN,BN,FN)
```

Parameters A_i, B_i and F_i have the same definitions as before, except that A_i and B_i may be variable names; e.g.,

CALL SPACE(4HDUMP,ARRAY(1),ARRAY(50),2)

If no parameters are supplied to the dump program in the DUMP control card series (or the calling sequence of a direct chain), then standard parameters will be assumed, as follows:

A_1 = S.SLOC/8*8

B_1 = S.SEND

F_1 = 0

Following execution of the dump program, the monitor will resume by reading the next program control card series, if any. Note that panel information (AC, MQ, etc.) will be given for each group of three parameters; however, only that given for the first group will reflect the status of the registers when the dump program was called.

2.2.2 THE POST-EXECUTION FILE UTILITY PROCESSOR

As mentioned in the introduction, the utility processor is a program residing in the library which may be considered as part of the monitor, only in core when needed. It is executed at the end of every job and performs the file utility requests specified in the REELS card series, provides statistics related to the job, and handles all terminal housekeeping functions.

The file utility operations which may be performed are two-fold and are as follows:

a. Dumping selected logical records within any given file(s) onto S.SOU1, formatted in octal, floating point decimal, integer, or BCD.
b. Copying any given file onto another file. It is possible to move a file to another position on the same reel, provided that some other non-file protected reel is being used in the job. A file may also be copied to a reel which is logically file protected.

Any request to perform a utility operation on a nonexistent file, or one which cannot be accessed will be identified and otherwise ignored. It will never be treated as an error.

When all file utility operations have been completed, the following information will be provided:

a. The reel number of each reel used in the job.
b. The symbolic and physical device to which each reel was assigned.
c. Both the number of files read and the number of files written on each reel during the job.
d. The position of the trailer label on each reel. Note that this reflects the number of useable files now contained on the reel.
e. A tabular listing indicating the name and order of all programs currently residing in the library.

The program name of the post-execution file utility processor is 'UTLITY'. In the event a program finds itself hopelessly lost, to the point where further execution should be abandoned, the following CALL should be executed:

```
CALL SPACE(6HUTLITY)
```

A program control card series is not required at the end of the job deck to execute the utility processor--it will automatically be called by the monitor.

2.2.3 ABSOLUTE BINARY PATCHING

In order to debug a new program, the programmer must create his own library tape, incorporating his program, by employing the COPY feature of IBJOB. To eliminate the need of recreating the library each time a new bug is found, the programmer can make execution-time binary patches to the appropriate program(s). This is accomplished by prefixing the letter X to the program name in the control card series and following the series with binary cards, punched in standard absolute column binary format. The last such card must be the transfer card and is identified by having a word count of zero. If the transfer address is non-zero, this address will be used as the entry point to the program; if it is zero, the standard entry point will be used. A checksum is computed and verified for each card unless (a) the ignore-checksum bit in column one is punched, or (b) the checksum on the card is zero. If the computed checksum does not agree with the card checksum, a warning message will be typed stating the load address of the offending card. The condition will otherwise be ignored.

An attempt to load a binary card below the origin of the first dependent link will be considered an error and will cause execution to be terminated. This is also true if a word count is given which exceeds $22_{/10}$, or if a BCD card is encountered before the transfer card.

To employ the patching facility, the programmer must, of course, know the absolute location(s) into which the patch is to be placed. By exercising the MAP or LOGIC option on the $IBJOB control card at copy time, he can determine the origin, or relocation factor, of the deck or subroutine which he wishes to patch. The absolute location of the patch can then be determined by adding this relocation factor to the relative location given in the assembly listing. Note that a technique of reserving a block of, say 50 cells in a program for future patching purposes deserves some merit.

When the monitor reads a program control card series which indicates the presence of binary cards, the following actions take place:

a. If the specified program is in the library, it is loaded into core storage.
b. The binary cards are processed.
c. Control is given to the program.

∟.2.4 USING THE LOGIC OR MEMORY MAP

By employing the LOGIC or MAP option on the $IBJOB control card when creating a library, the programmer is provided with facilities for determining the absolute origin and extent of each deck or subroutine contained therein. The logic option specifies a detailed storage-allocation map, while the MAP option specifies a non-detailed storage-allocation map. If the MAP option is used, in addition with the LOGIC option, the former results only in producing redundant information.

Inspecting the logic map of the main link (SPACE), the programmer can find the absolute origin of each file processing routine. Whenever one of these subroutines is called, the location plus one of the call is stored in the address of the appropriate location referenced in the logic map. Thus, in the event of an unexpected stop or error condition (accompanied with a dump) these locations act as pointers to the last call executed for each of the routines. The monitor always initializes these locations between programs.

The logic and memory maps which follow reflect a library configuration with SPACE as the main link, and four dependent links, each constituting one of the programs 13I, 12I, 61I and UTLITY. It should be noted that the utility program is the last link in the sample library. If this were not the case, it might have to be passed over several times in the course of accessing the other programs, thus causing an unnecessary waste of tape passing time.

```
                          L O G I C   M A P

                   FOR MAIN LINK

      REAL SECTION    'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.

DECK 'SPC    ' ASSIGNED ABSOLUTE ORIGIN 12352.  ADJUSTED LENGTH IS 04427.
      VIRTUAL SECTION 'S.BSR ' - REFERS TO DECK 'IBNUC ', LOCATION   00172.
      VIRTUAL SECTION 'S.CLSE' - REFERS TO DECK 'IBNUC ', LOCATION   00170.
      VIRTUAL SECTION 'S.FBIN' - REFERS TO DECK 'INSYFB', LOCATION   12261.
      VIRTUAL SECTION 'S.FBOU' - REFERS TO DECK 'OUSYFB', LOCATION   12304.
      VIRTUAL SECTION 'S.FBPP' - REFERS TO DECK 'PPSYFB', LOCATION   12327.
      VIRTUAL SECTION 'S.GETB' - REFERS TO DECK 'IBNUC ', LOCATION   00164.
      VIRTUAL SECTION 'S.GETL' - REFERS TO DECK 'IBNUC ', LOCATION   00163.
      VIRTUAL SECTION 'S.IOCP' - REFERS TO DECK 'IBNUC ', LOCATION   00157.
      VIRTUAL SECTION 'S.JNAM' - REFERS TO DECK 'IBNUC ', LOCATION   00311.
      VIRTUAL SECTION 'S.JXIT' - REFERS TO DECK 'POSTX ', LOCATION   17040.
      VIRTUAL SECTION 'S.OPEN' - REFERS TO DECK 'IBNUC ', LOCATION   00161.
      VIRTUAL SECTION 'S.PUTB' - REFERS TO DECK 'IBNUC ', LOCATION   00166.
      VIRTUAL SECTION 'S.PUTL' - REFERS TO DECK 'IBNUC ', LOCATION   00165.
      VIRTUAL SECTION 'S.SAVE' - REFERS TO DECK 'IBNUC ', LOCATION   00250.
      VIRTUAL SECTION 'S.SCCR' - REFERS TO DECK 'IBNUC ', LOCATION   00143.
      VIRTUAL SECTION 'S.SCDI' - REFERS TO DECK 'IBNUC ', LOCATION   00266.
      VIRTUAL SECTION 'S.SDAT' - REFERS TO DECK 'IBNUC ', LOCATION   00213.
      VIRTUAL SECTION 'S.SEND' - REFERS TO DECK 'IBNUC ', LOCATION   77777.
      VIRTUAL SECTION 'S.SLCR' - REFERS TO DECK 'IBNUC ', LOCATION   00135.
      VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
      VIRTUAL SECTION 'S.SLTC' - REFERS TO DECK 'IBNUC ', LOCATION   00210.
      VIRTUAL SECTION 'S.SUOO' - REFERS TO DECK 'IBNUC ', LOCATION   00337.
      VIRTUAL SECTION 'S.SUNI' - REFERS TO DECK 'IBNUC ', LOCATION   00207.
      VIRTUAL SECTION 'S.XACT' - REFERS TO DECK 'IBNUC ', LOCATION   00145.
      VIRTUAL SECTION 'S.XDVA' - REFERS TO DECK 'IBNUC ', LOCATION   00153.
      VIRTUAL SECTION 'S.XDVD' - REFERS TO DECK 'IBNUC ', LOCATION   00154.
      VIRTUAL SECTION 'S.XOVA' - REFERS TO DECK 'IBNUC ', LOCATION   00151.
      VIRTUAL SECTION 'S.XPRT' - REFERS TO DECK 'IBNUC ', LOCATION   00147.
      VIRTUAL SECTION 'S.XPSE' - REFERS TO DECK 'IBNUC ', LOCATION   00150.
      VIRTUAL SECTION 'S.XSNS' - REFERS TO DECK 'IBNUC ', LOCATION   00300.
      VIRTUAL SECTION 'PDUMP ' - REFERS TO DECK 'DMP   ', LOCATION   20731.
      VIRTUAL SECTION 'READHP' - REFERS TO DECK 'RDH44 ', LOCATION   22606.
      VIRTUAL SECTION 'SCAN  ' - REFERS TO DECK 'RDH44 ', LOCATION   23100.
      REAL SECTION    'SIGNON' - ASSIGNED ABSOLUTE ORIGIN 12352.
      REAL SECTION    'SPACE ' - ASSIGNED ABSOLUTE ORIGIN 13533.
      REAL SECTION    'READB1' - ASSIGNED ABSOLUTE ORIGIN 14242.
      REAL SECTION    'READB2' - ASSIGNED ABSOLUTE ORIGIN 14246.
      REAL SECTION    'ABOUT1' - ASSIGNED ABSOLUTE ORIGIN 14252.
      REAL SECTION    'ABOUT2' - ASSIGNED ABSOLUTE ORIGIN 14256.
      REAL SECTION    'SKIPR ' - ASSIGNED ABSOLUTE ORIGIN 14262.
      REAL SECTION    'BSPR  ' - ASSIGNED ABSOLUTE ORIGIN 14266.
      REAL SECTION    'ENDF  ' - ASSIGNED ABSOLUTE ORIGIN 14272.
      REAL SECTION    'OEDIT ' - ASSIGNED ABSOLUTE ORIGIN 15344.
      REAL SECTION    'SYSTEM' - ASSIGNED ABSOLUTE ORIGIN 16476.
      REAL SECTION    'ACCESS' - ASSIGNED ABSOLUTE ORIGIN 16617.

DECK 'DSSCAN' ASSIGNED ABSOLUTE ORIGIN 17001.  ADJUSTED LENGTH IS 00037.
      VIRTUAL SECTION 'SCAN  ' - REFERS TO DECK 'RDH44 ', LOCATION   23100.
```

```
     VIRTUAL SECTION 'S.SLCC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     REAL SECTION    'DSSCAN' - ASSIGNED ABSOLUTE ORIGIN 17007.

DECK 'IBNUC ' ASSIGNED ABSOLUTE ORIGIN 00000.  ABSOLUTE DECK.
     REAL SECTION    'S.SUTL' - ASSIGNED ABSOLUTE ORIGIN 00032.
     REAL SECTION    'S.SLCR' - ASSIGNED ABSOLUTE ORIGIN 00135.
     REAL SECTION    'S.SRPT' - ASSIGNED ABSOLUTE ORIGIN 00136.
     REAL SECTION    'S.SDMP' - ASSIGNED ABSOLUTE ORIGIN 00137.
     REAL SECTION    'S.SRUP' - ASSIGNED ABSOLUTE ORIGIN 00140.
     REAL SECTION    'S.SRET' - ASSIGNED ABSOLUTE ORIGIN 00141.
     REAL SECTION    'S.SRST' - ASSIGNED ABSOLUTE ORIGIN 00142.
     REAL SECTION    'S.SCCR' - ASSIGNED ABSOLUTE ORIGIN 00143.
     REAL SECTION    'S.SICR' - ASSIGNED ABSOLUTE ORIGIN 00144.
     REAL SECTION    'S.XACT' - ASSIGNED ABSOLUTE ORIGIN 00145.
     REAL SECTION    'S.XDAC' - ASSIGNED ABSOLUTE ORIGIN 00146.
     REAL SECTION    'S.XPRT' - ASSIGNED ABSOLUTE ORIGIN 00147.
     REAL SECTION    'S.XPSE' - ASSIGNED ABSOLUTE ORIGIN 00150.
     REAL SECTION    'S.XUVA' - ASSIGNED ABSOLUTE ORIGIN 00151.
     REAL SECTION    'S.XUVD' - ASSIGNED ABSOLUTE ORIGIN 00152.
     REAL SECTION    'S.XDVA' - ASSIGNED ABSOLUTE ORIGIN 00153.
     REAL SECTION    'S.XDVD' - ASSIGNED ABSOLUTE ORIGIN 00154.
     REAL SECTION    'S.XUCV' - ASSIGNED ABSOLUTE ORIGIN 00155.
     REAL SECTION    'S.SCKT' - ASSIGNED ABSOLUTE ORIGIN 00156.
     REAL SECTION    'S.IOOP' - ASSIGNED ABSOLUTE ORIGIN 00157.
     REAL SECTION    'S.IOLS' - ASSIGNED ABSOLUTE ORIGIN 00160.
     REAL SECTION    'S.SCBL' - ASSIGNED ABSOLUTE ORIGIN 00160.
     REAL SECTION    'S.OPEN' - ASSIGNED ABSOLUTE ORIGIN 00161.
     REAL SECTION    'S.OPNL' - ASSIGNED ABSOLUTE ORIGIN 00162.
     REAL SECTION    'S.GETL' - ASSIGNED ABSOLUTE ORIGIN 00163.
     REAL SECTION    'S.GETB' - ASSIGNED ABSOLUTE ORIGIN 00164.
     REAL SECTION    'S.PUTL' - ASSIGNED ABSOLUTE ORIGIN 00165.
     REAL SECTION    'S.PUTB' - ASSIGNED ABSOLUTE ORIGIN 00166.
     REAL SECTION    'S.PLCC' - ASSIGNED ABSOLUTE ORIGIN 00167.
     REAL SECTION    'S.CLSE' - ASSIGNED ABSOLUTE ORIGIN 00170.
     REAL SECTION    'S.CLSL' - ASSIGNED ABSOLUTE ORIGIN 00171.
     REAL SECTION    'S.BSR ' - ASSIGNED ABSOLUTE ORIGIN 00172.
     REAL SECTION    'S.WEF ' - ASSIGNED ABSOLUTE ORIGIN 00173.
     REAL SECTION    'S.REW ' - ASSIGNED ABSOLUTE ORIGIN 00174.
     REAL SECTION    'S.FEOR' - ASSIGNED ABSOLUTE ORIGIN 00175.
     REAL SECTION    'S.CKPT' - ASSIGNED ABSOLUTE ORIGIN 00176.
     REAL SECTION    'S.SLVL' - ASSIGNED ABSOLUTE ORIGIN 00177.
     REAL SECTION    'S.SCOR' - ASSIGNED ABSOLUTE ORIGIN 00200.
     REAL SECTION    'S.SCSN' - ASSIGNED ABSOLUTE ORIGIN 00201.
     REAL SECTION    'S.SPNC' - ASSIGNED ABSOLUTE ORIGIN 00202.
     REAL SECTION    'S.SCMX' - ASSIGNED ABSOLUTE ORIGIN 00203.
     REAL SECTION    'S.SPER' - ASSIGNED ABSOLUTE ORIGIN 00204.
     REAL SECTION    'S.SUBC' - ASSIGNED ABSOLUTE ORIGIN 00205.
     REAL SECTION    'S.SSBC' - ASSIGNED ABSOLUTE ORIGIN 00206.
     REAL SECTION    'S.SUNI' - ASSIGNED ABSOLUTE ORIGIN 00207.
     REAL SECTION    'S.SLTC' - ASSIGNED ABSOLUTE ORIGIN 00210.
     REAL SECTION    'S.SRCC' - ASSIGNED ABSOLUTE ORIGIN 00211.
     REAL SECTION    'S.BSLA' - ASSIGNED ABSOLUTE ORIGIN 00212.
     REAL SECTION    'S.SDAT' - ASSIGNED ABSOLUTE ORIGIN 00213.
     REAL SECTION    'S.SCLK' - ASSIGNED ABSOLUTE ORIGIN 00215.
     REAL SECTION    'S.SCIS' - ASSIGNED ABSOLUTE ORIGIN 00216.
```

```
REAL SECTION    'S.SDEX' - ASSIGNED ABSOLUTE ORIGIN 00217.
REAL SECTION    'S.UCBL' - ASSIGNED ABSOLUTE ORIGIN 00220.
REAL SECTION    'S.SCUR' - ASSIGNED ABSOLUTE ORIGIN 00221.
REAL SECTION    'S.SFAZ' - ASSIGNED ABSOLUTE ORIGIN 00222.
REAL SECTION    'S.SSWI' - ASSIGNED ABSOLUTE ORIGIN 00223.
REAL SECTION    'S.SFLG' - ASSIGNED ABSOLUTE ORIGIN 00224.
REAL SECTION    'S.SAVE' - ASSIGNED ABSOLUTE ORIGIN 00250.
REAL SECTION    'S.SCDI' - ASSIGNED ABSOLUTE ORIGIN 00266.
REAL SECTION    'S.PGCT' - ASSIGNED ABSOLUTE ORIGIN 00267.
REAL SECTION    'S.SHCR' - ASSIGNED ABSOLUTE ORIGIN 00270.
REAL SECTION    'S.SSCH' - ASSIGNED ABSOLUTE ORIGIN 00275.
REAL SECTION    'S.SSNS' - ASSIGNED ABSOLUTE ORIGIN 00276.
REAL SECTION    'S.XTCT' - ASSIGNED ABSOLUTE ORIGIN 00277.
REAL SECTION    'S.XSNS' - ASSIGNED ABSOLUTE ORIGIN 00300.
REAL SECTION    'S.XLTP' - ASSIGNED ABSOLUTE ORIGIN 00302.
REAL SECTION    'S.XSCH' - ASSIGNED ABSOLUTE ORIGIN 00303.
REAL SECTION    'S.XTPS' - ASSIGNED ABSOLUTE ORIGIN 00304.
REAL SECTION    'S.XCPS' - ASSIGNED ABSOLUTE ORIGIN 00305.
REAL SECTION    'S.NAPT' - ASSIGNED ABSOLUTE ORIGIN 00306.
REAL SECTION    'S.SFBL' - ASSIGNED ABSOLUTE ORIGIN 00307.
REAL SECTION    'S.JNAM' - ASSIGNED ABSOLUTE ORIGIN 00311.
REAL SECTION    'S.IAUN' - ASSIGNED ABSOLUTE ORIGIN 00312.
REAL SECTION    'S.OAUN' - ASSIGNED ABSOLUTE ORIGIN 00313.
REAL SECTION    'S.LDUN' - ASSIGNED ABSOLUTE ORIGIN 00314.
REAL SECTION    'S.EDUN' - ASSIGNED ABSOLUTE ORIGIN 00315.
REAL SECTION    'S.SRUS' - ASSIGNED ABSOLUTE ORIGIN 00316.
REAL SECTION    'S.SPRP' - ASSIGNED ABSOLUTE ORIGIN 00317.
REAL SECTION    'S.SYCV' - ASSIGNED ABSOLUTE ORIGIN 00320.
REAL SECTION    'S.SYCW' - ASSIGNED ABSOLUTE ORIGIN 00321.
REAL SECTION    'S.SYCX' - ASSIGNED ABSOLUTE ORIGIN 00322.
REAL SECTION    'S.SYCY' - ASSIGNED ABSOLUTE ORIGIN 00323.
REAL SECTION    'S.SYCZ' - ASSIGNED ABSOLUTE ORIGIN 00324.
REAL SECTION    'S.SYFS' - ASSIGNED ABSOLUTE ORIGIN 00325.
REAL SECTION    'S.SLB1' - ASSIGNED ABSOLUTE ORIGIN 00326.
REAL SECTION    'S.SLB2' - ASSIGNED ABSOLUTE ORIGIN 00327.
REAL SECTION    'S.SIN1' - ASSIGNED ABSOLUTE ORIGIN 00330.
REAL SECTION    'S.SIN2' - ASSIGNED ABSOLUTE ORIGIN 00331.
REAL SECTION    'S.SOU1' - ASSIGNED ABSOLUTE ORIGIN 00332.
REAL SECTION    'S.SOU2' - ASSIGNED ABSOLUTE ORIGIN 00333.
REAL SECTION    'S.SPP1' - ASSIGNED ABSOLUTE ORIGIN 00334.
REAL SECTION    'S.SPP2' - ASSIGNED ABSOLUTE ORIGIN 00335.
REAL SECTION    'S.SCK1' - ASSIGNED ABSOLUTE ORIGIN 00336.
REAL SECTION    'S.SU00' - ASSIGNED ABSOLUTE ORIGIN 00337.
REAL SECTION    'S.SU01' - ASSIGNED ABSOLUTE ORIGIN 00340.
REAL SECTION    'S.SU02' - ASSIGNED ABSOLUTE ORIGIN 00341.
REAL SECTION    'S.SU03' - ASSIGNED ABSOLUTE ORIGIN 00342.
REAL SECTION    'S.SU04' - ASSIGNED ABSOLUTE ORIGIN 00343.
REAL SECTION    'S.SU05' - ASSIGNED ABSOLUTE ORIGIN 00344.
REAL SECTION    'S.SU06' - ASSIGNED ABSOLUTE ORIGIN 00345.
REAL SECTION    'S.SU07' - ASSIGNED ABSOLUTE ORIGIN 00346.
REAL SECTION    'S.SU08' - ASSIGNED ABSOLUTE ORIGIN 00347.
REAL SECTION    'S.SU09' - ASSIGNED ABSOLUTE ORIGIN 00350.
REAL SECTION    'S.SU10' - ASSIGNED ABSOLUTE ORIGIN 00351.
REAL SECTION    'S.SU11' - ASSIGNED ABSOLUTE ORIGIN 00352.
REAL SECTION    'S.SU12' - ASSIGNED ABSOLUTE ORIGIN 00353.
```

```
     REAL SECTION    'S.SU13' - ASSIGNED ABSOLUTE ORIGIN 00354.
     REAL SECTION    'S.SU14' - ASSIGNED ABSOLUTE ORIGIN 00355.
     REAL SECTION    'S.SU15' - ASSIGNED ABSOLUTE ORIGIN 00356.
     REAL SECTION    'S.SU16' - ASSIGNED ABSOLUTE ORIGIN 00357.
     REAL SECTION    'S.SUTT' - ASSIGNED ABSOLUTE ORIGIN 00360.
     REAL SECTION    'S.UCBT' - ASSIGNED ABSOLUTE ORIGIN 00600.
     REAL SECTION    'S.SCBT' - ASSIGNED ABSOLUTE ORIGIN 00760.
     REAL SECTION    'S.SDMA' - ASSIGNED ABSOLUTE ORIGIN 01643.
     REAL SECTION    'IBDMP ' - ASSIGNED ABSOLUTE ORIGIN 01653.
     REAL SECTION    'S.SORG' - ASSIGNED ABSOLUTE ORIGIN 03000.
     REAL SECTION    'S.LSO1' - ASSIGNED ABSOLUTE ORIGIN 06401.
     REAL SECTION    'S.LRS1' - ASSIGNED ABSOLUTE ORIGIN 06433.
     REAL SECTION    'S.LRS2' - ASSIGNED ABSOLUTE ORIGIN 06440.
     REAL SECTION    'S.LRS3' - ASSIGNED ABSOLUTE ORIGIN 06441.
     REAL SECTION    'S.LFE1' - ASSIGNED ABSOLUTE ORIGIN 06445.
     REAL SECTION    'S.LFE2' - ASSIGNED ABSOLUTE ORIGIN 06452.
     REAL SECTION    'S.LFXT' - ASSIGNED ABSOLUTE ORIGIN 06500.
     REAL SECTION    'S.LVI1' - ASSIGNED ABSOLUTE ORIGIN 06514.
     REAL SECTION    'S.LWO3' - ASSIGNED ABSOLUTE ORIGIN 06573.
     REAL SECTION    'S.LRH4' - ASSIGNED ABSOLUTE ORIGIN 06661.
     REAL SECTION    'S.LST4' - ASSIGNED ABSOLUTE ORIGIN 07050.
     REAL SECTION    'S.LVRR' - ASSIGNED ABSOLUTE ORIGIN 07343.
     REAL SECTION    'S.LERR' - ASSIGNED ABSOLUTE ORIGIN 07405.
     REAL SECTION    'S.SP1D' - ASSIGNED ABSOLUTE ORIGIN 10000.
     REAL SECTION    'S.SP2D' - ASSIGNED ABSOLUTE ORIGIN 10000.
     REAL SECTION    'S.LVLN' - ASSIGNED ABSOLUTE ORIGIN 10000.
     REAL SECTION    'S.LXRS' - ASSIGNED ABSOLUTE ORIGIN 10044.
     REAL SECTION    'S.SLND' - ASSIGNED ABSOLUTE ORIGIN 10074.
     REAL SECTION    'S.BDMP' - ASSIGNED ABSOLUTE ORIGIN 12126.
     REAL SECTION    'S.SSND' - ASSIGNED ABSOLUTE ORIGIN 12253.
     REAL SECTION    'S.SEND' - ASSIGNED ABSOLUTE ORIGIN 77777.

DECK 'INSYFB' ASSIGNED ABSOLUTE ORIGIN 00000.  ADJUSTED LENGTH IS 00000.
     FILE            'S.FBIN' - ASSIGNED ABSOLUTE ORIGIN 12261.

DECK 'OUSYFB' ASSIGNED ABSOLUTE ORIGIN 00000.  ADJUSTED LENGTH IS 00000.
     FILE            'S.FBOU' - ASSIGNED ABSOLUTE ORIGIN 12304.

DECK 'POSTX ' ASSIGNED ABSOLUTE ORIGIN 17040.  ADJUSTED LENGTH IS 00112.
     VIRTUAL SECTION 'S.SRET' - REFERS TO DECK 'IBNUC ', LOCATION   00141.
     VIRTUAL SECTION 'S.SCUR' - REFERS TO DECK 'IBNUC ', LOCATION   00221.
     VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     VIRTUAL SECTION 'S.CLSE' - REFERS TO DECK 'IBNUC ', LOCATION   00170.
     VIRTUAL SECTION 'S.SCCR' - REFERS TO DECK 'IBNUC ', LOCATION   00143.
     VIRTUAL SECTION 'S.SIN1' - REFERS TO DECK 'IBNUC ', LOCATION   00330.
     VIRTUAL SECTION 'S.SSWI' - REFERS TO DECK 'IBNUC ', LOCATION   00223.
     VIRTUAL SECTION 'S.SOU1' - REFERS TO DECK 'IBNUC ', LOCATION   00332.
     VIRTUAL SECTION 'S.SICR' - REFERS TO DECK 'IBNUC ', LOCATION   00144.
     VIRTUAL SECTION 'S.SPP1' - REFERS TO DECK 'IBNUC ', LOCATION   00334.
     REAL SECTION    'S.JXIT' - ASSIGNED ABSOLUTE ORIGIN 17040.

DECK 'PPSYFB' ASSIGNED ABSOLUTE ORIGIN 00000.  ADJUSTED LENGTH IS 00000.
     FILE            'S.FBPP' - ASSIGNED ABSOLUTE ORIGIN 12327.

DECK 'F05   ' ASSIGNED ABSOLUTE ORIGIN 17152.  ADJUSTED LENGTH IS 00001.
```

```
     VIRTUAL SECTION 'S.FBIN' - REFERS TO DECK 'INSYFB', LOCATION   12261.
     REAL SECTION    'FILO5.' - ASSIGNED ABSOLUTE ORIGIN 17152.

DECK 'F06    ' ASSIGNED ABSOLUTE ORIGIN 17153.  ADJUSTED LENGTH IS 00001.
     VIRTUAL SECTION 'S.FBCU' - REFERS TO DECK 'OUSYFB', LOCATION   12304.
     REAL SECTION    'FIL06.' - ASSIGNED ABSOLUTE ORIGIN 17153.

DECK 'IOS    ' ASSIGNED ABSOLUTE ORIGIN 17154.  ADJUSTED LENGTH IS 00276.
     VIRTUAL SECTION 'FEXEM.' - REFERS TO DECK 'XEM   ', LOCATION   20473.
     VIRTUAL SECTION 'EXIT  ' - REFERS TO DECK 'XIT   ', LOCATION   20700.
     VIRTUAL SECTION 'S.FBOU' - REFERS TO DECK 'OUSYFB', LOCATION   12304.
     VIRTUAL SECTION 'S.SINI' - REFERS TO DECK 'IBNUC ', LOCATION   00330.
     VIRTUAL SECTION 'S.SPRP' - REFERS TO DECK 'IBNUC ', LOCATION   00317.
     VIRTUAL SECTION 'S.SCCI' - REFERS TO DECK 'IBNUC ', LOCATION   00266.
     VIRTUAL SECTION 'S.PUTL' - REFERS TO DECK 'IBNUC ', LOCATION   00165.
     VIRTUAL SECTION 'S.SCCR' - REFERS TO DECK 'IBNUC ', LOCATION   00143.
     VIRTUAL SECTION 'S.CLSE' - REFERS TO DECK 'IBNUC ', LOCATION   00170.
     VIRTUAL SECTION 'S.OPEN' - REFERS TO DECK 'IBNUC ', LOCATION   00161.
     VIRTUAL SECTION 'S.SPPI' - REFERS TO DECK 'IBNUC ', LOCATION   00334.
     VIRTUAL SECTION 'S.PLOC' - REFERS TO DECK 'IBNUC ', LOCATION   00167.
     VIRTUAL SECTION 'S.SOUI' - REFERS TO DECK 'IBNUC ', LOCATION   00332.
     VIRTUAL SECTION 'S.SAVE' - REFERS TO DECK 'IBNUC ', LOCATION   00250.
     REAL SECTION    'IOSUP.' - ASSIGNED ABSOLUTE ORIGIN 17154.
     REAL SECTION    'CKEND.' - ASSIGNED ABSOLUTE ORIGIN 17257.
     REAL SECTION    'SYSOU.' - ASSIGNED ABSOLUTE ORIGIN 17301.
     REAL SECTION    'SYSCK.' - ASSIGNED ABSOLUTE ORIGIN 17334.
     REAL SECTION    'RERRX.' - ASSIGNED ABSOLUTE ORIGIN 17372.
     REAL SECTION    'REOFX.' - ASSIGNED ABSOLUTE ORIGIN 17402.
     REAL SECTION    'REORX.' - ASSIGNED ABSOLUTE ORIGIN 17410.
     REAL SECTION    'IOCEL.' - ASSIGNED ABSOLUTE ORIGIN 17430.

DECK 'RWD    ' ASSIGNED ABSOLUTE ORIGIN 17452.  ADJUSTED LENGTH IS 00531.
     VIRTUAL SECTION 'ERLOC.' - REFERS TO DECK 'XEM   ', LOCATION   20472.
     VIRTUAL SECTION 'IOSUP.' - REFERS TO DECK 'IOS   ', LOCATION   17154.
     VIRTUAL SECTION 'CKEND.' - REFERS TO DECK 'IOS   ', LOCATION   17257.
     VIRTUAL SECTION 'FEXEM.' - REFERS TO DECK 'XEM   ', LOCATION   20473.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS   ', LOCATION   17430.
     VIRTUAL SECTION 'S.GETL' - REFERS TO DECK 'IBNUC ', LOCATION   00163.
     VIRTUAL SECTION 'S.PUTL' - REFERS TO DECK 'IBNUC ', LOCATION   00165.
     VIRTUAL SECTION 'S.GETR' - REFERS TO DECK 'IBNUC ', LOCATION   00164.
     VIRTUAL SECTION 'S.PLOC' - REFERS TO DECK 'IBNUC ', LOCATION   00167.
     VIRTUAL SECTION 'S.OPEN' - REFERS TO DECK 'IBNUC ', LOCATION   00161.
     REAL SECTION    'TSHIO.' - ASSIGNED ABSOLUTE ORIGIN 17452.
     REAL SECTION    'STHIO.' - ASSIGNED ABSOLUTE ORIGIN 17472.
     REAL SECTION    'IOHCM.' - ASSIGNED ABSOLUTE ORIGIN 17537.
     REAL SECTION    'HDLIO.' - ASSIGNED ABSOLUTE ORIGIN 17570.
     REAL SECTION    'HNLIO.' - ASSIGNED ABSOLUTE ORIGIN 17573.
     REAL SECTION    'HCT.  ' - ASSIGNED ABSOLUTE ORIGIN 17614.
     REAL SECTION    'IOHCT.' - ASSIGNED ABSOLUTE ORIGIN 17615.
     REAL SECTION    'IOHLP.' - ASSIGNED ABSOLUTE ORIGIN 17625.
     REAL SECTION    'IOHRP.' - ASSIGNED ABSOLUTE ORIGIN 17646.
     REAL SECTION    'IOHEF.' - ASSIGNED ABSOLUTE ORIGIN 17665.
     REAL SECTION    'IOHSF.' - ASSIGNED ABSOLUTE ORIGIN 17677.
     REAL SECTION    'IOHIO.' - ASSIGNED ABSOLUTE ORIGIN 17702.
     REAL SECTION    'SC.   ' - ASSIGNED ABSOLUTE ORIGIN 20021.
```

```
     REAL SECTION     'CCA.  ' - ASSIGNED ABSOLUTE ORIGIN 20027.
     REAL SECTION     'CCQ.  ' - ASSIGNED ABSOLUTE ORIGIN 20031.
     REAL SECTION     'BU1.  ' - ASSIGNED ABSOLUTE ORIGIN 20037.
     REAL SECTION     'BU4.  ' - ASSIGNED ABSOLUTE ORIGIN 20044.
     REAL SECTION     'BU10. ' - ASSIGNED ABSOLUTE ORIGIN 20047.
     REAL SECTION     'BU22. ' - ASSIGNED ABSOLUTE ORIGIN 20053.
     REAL SECTION     'DE60. ' - ASSIGNED ABSOLUTE ORIGIN 20077.
     REAL SECTION     'DE70. ' - ASSIGNED ABSOLUTE ORIGIN 20100.
     REAL SECTION     'XC.   ' - ASSIGNED ABSOLUTE ORIGIN 20110.
     REAL SECTION     'FILIO.' - ASSIGNED ABSOLUTE ORIGIN 20141.
     REAL SECTION     'RTNIO.' - ASSIGNED ABSOLUTE ORIGIN 20144.
     REAL SECTION     'GETCH.' - ASSIGNED ABSOLUTE ORIGIN 20151.
     REAL SECTION     'GETC3.' - ASSIGNED ABSOLUTE ORIGIN 20152.
     REAL SECTION     'GETC. ' - ASSIGNED ABSOLUTE ORIGIN 20163.
     REAL SECTION     'CVCEL.' - ASSIGNED ABSOLUTE ORIGIN 20165.

DECK 'ACV   ' ASSIGNED ABSOLUTE ORIGIN 20203.  ADJUSTED LENGTH IS 00030.
     VIRTUAL SECTION 'BU1.  ' - REFERS TO DECK 'RWD   ', LOCATION   20037.
     VIRTUAL SECTION 'BU10. ' - REFERS TO DECK 'RWD   ', LOCATION   20047.
     VIRTUAL SECTION 'CCA.  ' - REFERS TO DECK 'RWD   ', LOCATION   20027.
     VIRTUAL SECTION 'GETCH.' - REFERS TO DECK 'RWD   ', LOCATION   20151.
     VIRTUAL SECTION 'HCT.  ' - REFERS TO DECK 'RWD   ', LOCATION   17614.
     VIRTUAL SECTION 'SC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20021.
     VIRTUAL SECTION 'XC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20110.
     VIRTUAL SECTION 'CVCEL.' - REFERS TO DECK 'RWD   ', LOCATION   20165.
     VIRTUAL SECTION 'IOHCM.' - REFERS TO DECK 'RWD   ', LOCATION   17537.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS   ', LOCATION   17430.
     REAL SECTION     'IOHAC.' - ASSIGNED ABSOLUTE ORIGIN 20203.

DECK 'XCV   ' ASSIGNED ABSOLUTE ORIGIN 20233.  ADJUSTED LENGTH IS 00016.
     VIRTUAL SECTION 'CCQ.  ' - REFERS TO DECK 'RWD   ', LOCATION   20031.
     VIRTUAL SECTION 'GETCH.' - REFERS TO DECK 'RWD   ', LOCATION   20151.
     VIRTUAL SECTION 'HCT.  ' - REFERS TO DECK 'RWD   ', LOCATION   17614.
     VIRTUAL SECTION 'SC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20021.
     VIRTUAL SECTION 'XC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20110.
     VIRTUAL SECTION 'IOHCT.' - REFERS TO DECK 'RWD   ', LOCATION   17615.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS   ', LOCATION   17430.
     VIRTUAL SECTION 'CVCEL.' - REFERS TO DECK 'RWD   ', LOCATION   20165.
     REAL SECTION     'IOHXC.' - ASSIGNED ABSOLUTE ORIGIN 20233.

DECK 'FPT   ' ASSIGNED ABSOLUTE ORIGIN 20251.  ADJUSTED LENGTH IS 00216.
     VIRTUAL SECTION 'SYSOU.' - REFERS TO DECK 'IOS   ', LOCATION   17301.
     VIRTUAL SECTION 'ERLOC.' - REFERS TO DECK 'XEM   ', LOCATION   20472.
     VIRTUAL SECTION 'FEXEM.' - REFERS TO DECK 'XEM   ', LOCATION   20473.
     VIRTUAL SECTION 'EXIT  ' - REFERS TO DECK 'XIT   ', LOCATION   20700.
     VIRTUAL SECTION 'S.XOVA' - REFERS TO DECK 'IBNUC ', LOCATION   00151.
     VIRTUAL SECTION 'S.SDAT' - REFERS TO DECK 'IBNUC ', LOCATION   00213.
     VIRTUAL SECTION 'S.SCCR' - REFERS TO DECK 'IBNUC ', LOCATION   00143.
     REAL SECTION     'SETFP.' - ASSIGNED ABSOLUTE ORIGIN 20251.
     REAL SECTION     'GGDAT1' - ASSIGNED ABSOLUTE ORIGIN 20300.
     REAL SECTION     'GGDATE' - ASSIGNED ABSOLUTE ORIGIN 20301.
     REAL SECTION     'OVERF.' - ASSIGNED ABSOLUTE ORIGIN 20453.

DECK 'XEM   ' ASSIGNED ABSOLUTE ORIGIN 20467.  ADJUSTED LENGTH IS 00211.
     VIRTUAL SECTION 'SYSOU.' - REFERS TO DECK 'IOS   ', LOCATION   17301.
```

```
     VIRTUAL SECTION 'S.XDVD' - REFERS TO DECK 'IBNUC ', LOCATION   00154.
     VIRTUAL SECTION 'S.SDMP' - REFERS TO DECK 'IBNUC ', LOCATION   00137.
     VIRTUAL SECTION 'S.XOVA' - REFERS TO DECK 'IBNUC ', LOCATION   00151.
     VIRTUAL SECTION 'S.JXIT' - REFERS TO DECK 'POSTX ', LOCATION   17040.
     VIRTUAL SECTION 'S.XDVA' - REFERS TO DECK 'IBNUC ', LOCATION   00153.
     REAL SECTION    'ERROU.' - ASSIGNED ABSOLUTE ORIGIN 20470.
     REAL SECTION    'ERLOC.' - ASSIGNED ABSOLUTE ORIGIN 20472.
     REAL SECTION    'FEXEM.' - ASSIGNED ABSOLUTE ORIGIN 20473.
     REAL SECTION    'MATOP.' - ASSIGNED ABSOLUTE ORIGIN 20674.
     REAL SECTION    'SYSOP.' - ASSIGNED ABSOLUTE ORIGIN 20675.
     REAL SECTION    'IOCEL.' - DELETED.  REFERS TO DECK 'IOS    ', LOCATION   17430.

DECK 'XIT    ' ASSIGNED ABSOLUTE ORIGIN 20700.  ADJUSTED LENGTH IS 00002.
     VIRTUAL SECTION 'S.JXIT' - REFERS TO DECK 'POSTX ', LOCATION   17040.
     REAL SECTION    'EXIT  ' - ASSIGNED ABSOLUTE ORIGIN 20700.

DECK 'DMP    ' ASSIGNED ABSOLUTE ORIGIN 20702.  ADJUSTED LENGTH IS 01670.
     VIRTUAL SECTION 'SYSOU.' - REFERS TO DECK 'IOS   ', LOCATION   17301.
     VIRTUAL SECTION 'EXIT  ' - REFERS TO DECK 'XIT   ', LOCATION   20700.
     VIRTUAL SECTION 'S.XOVA' - REFERS TO DECK 'IBNUC ', LOCATION   00151.
     VIRTUAL SECTION 'S.SCOR' - REFERS TO DECK 'IBNUC ', LOCATION   00200.
     VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     REAL SECTION    'DUMP  ' - ASSIGNED ABSOLUTE ORIGIN 20710.
     REAL SECTION    'PDUMP ' - ASSIGNED ABSOLUTE ORIGIN 20731.

DECK 'RDH44 ' ASSIGNED ABSOLUTE ORIGIN 22572.  ADJUSTED LENGTH IS 01440.
     VIRTUAL SECTION 'FIL05.' - REFERS TO DECK 'F05   ', LOCATION   17152.
     VIRTUAL SECTION 'FIL06.' - REFERS TO DECK 'F06   ', LOCATION   17153.
     VIRTUAL SECTION 'FILIO.' - REFERS TO DECK 'RWD   ', LOCATION   20141.
     VIRTUAL SECTION 'HNLIO.' - REFERS TO DECK 'RWD   ', LOCATION   17573.
     VIRTUAL SECTION 'IOHAC.' - REFERS TO DECK 'ACV   ', LOCATION   20203.
     VIRTUAL SECTION 'IOHEF.' - REFERS TO DECK 'RWD   ', LOCATION   17665.
     VIRTUAL SECTION 'IOHIO.' - REFERS TO DECK 'RWD   ', LOCATION   17702.
     VIRTUAL SECTION 'IOHXC.' - REFERS TO DECK 'XCV   ', LOCATION   20233.
     VIRTUAL SECTION 'RTNIO.' - REFERS TO DECK 'RWD   ', LOCATION   20144.
     VIRTUAL SECTION 'SETFP.' - REFERS TO DECK 'FPT   ', LOCATION   20251.
     VIRTUAL SECTION 'STHIO.' - REFERS TO DECK 'RWD   ', LOCATION   17472.
     VIRTUAL SECTION 'TSHIO.' - REFERS TO DECK 'RWD   ', LOCATION   17452.
     VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     VIRTUAL SECTION 'S.JXIT' - REFERS TO DECK 'POSTX ', LOCATION   17040.
     VIRTUAL SECTION 'S.SCCR' - REFERS TO DECK 'IBNUC ', LOCATION   00143.
     REAL SECTION    'READH ' - ASSIGNED ABSOLUTE ORIGIN 22572.
     REAL SECTION    'READHR' - ASSIGNED ABSOLUTE ORIGIN 22601.
     REAL SECTION    'READHP' - ASSIGNED ABSOLUTE ORIGIN 22606.
     REAL SECTION    'BCD   ' - ASSIGNED ABSOLUTE ORIGIN 22760.
     REAL SECTION    'BCDT  ' - ASSIGNED ABSOLUTE ORIGIN 22767.
     REAL SECTION    'BCDP  ' - ASSIGNED ABSOLUTE ORIGIN 22775.
     REAL SECTION    'SCAN  ' - ASSIGNED ABSOLUTE ORIGIN 23100.
     REAL SECTION    'RDHREE' - ASSIGNED ABSOLUTE ORIGIN 24214.
```

```
SPACE LIBRARY - VER. 2, MOD. 1                    IBLDR -- JOB   SPACE

                         M E M O R Y   M A P

                  FOR MAIN LINK

SYSTEM, INCLUDING IOCS                              00000 THRU  12252

FILE BLOCK ORIGIN                                   12261

    NUMBER OF FILES -        3

    1.   S.FBIN                       12261
    2.   S.FBOU                       12304
    3.   S.FBPP                       12327

OBJECT PROGRAM                                      12352 THRU  24231

    1.   DECK 'SPC    '               12352
    2.   DECK 'DSSCAN'                17001
    3.   SUBR 'INSYFB'                00000
    4.   SUBR 'OUSYFB'                00000
    5.   SUBR 'POSTX '                17040
    6.   SUBR 'PPSYFB'                00000
    7.   SUBR 'F05    '               17152
    8.   SUBR 'F06    '               17153
    9.   SUBR 'IOS    '               17154
   10.   SUBR 'RWD    '               17452
   11.   SUBR 'ACV    '               20203
   12.   SUBR 'XCV    '               20233
   13.   SUBR 'FPT    '               20251
   14.   SUBR 'XEM    '   *           20467
   15.   SUBR 'XIT    '               20700
   16.   SUBR 'DMP    '               20702
   17.   SUBR 'RDH44 '                22572

   (* - INSERTIONS OR DELETIONS MADE IN THIS DECK)

INPUT - OUTPUT BUFFERS                              76157 THRU  77776

UNUSED CORE                                         24232 THRU  76147
```

```
SPACE LIBRARY - VER. 2, MOD. 1                    IBLDR -- JOB   SPACE

                    L O G I C   M A P

  FOR DECKS OF LINK NUMBER  1,( 13I      )

 REAL SECTION     'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.

DECK '13I    ' ASSIGNED ABSOLUTE ORIGIN 24232.  ADJUSTED LENGTH IS 37553.
     VIRTUAL SECTION 'ABOUT1' - REFERS TO DECK 'SPC    ', LOCATION   14252.
     VIRTUAL SECTION 'ABOUT2' - REFERS TO DECK 'SPC    ', LOCATION   14256.
     VIRTUAL SECTION 'ENDF  ' - REFERS TO DECK 'SPC    ', LOCATION   14272.
     VIRTUAL SECTION 'READB1' - REFERS TO DECK 'SPC    ', LOCATION   14242.
     VIRTUAL SECTION 'READB2' - REFERS TO DECK 'SPC    ', LOCATION   14246.
     VIRTUAL SECTION 'SKIPR ' - REFERS TO DECK 'SPC    ', LOCATION   14262.
     VIRTUAL SECTION 'OEDIT ' - REFERS TO DECK 'SPC    ', LOCATION   15344.
     VIRTUAL SECTION 'SPACE ' - REFERS TO DECK 'SPC    ', LOCATION   13533.
     VIRTUAL SECTION 'S.SDAT' - REFERS TO DECK 'IBNUC ', LOCATION   00213.
     VIRTUAL SECTION 'S.XDVA' - REFERS TO DECK 'IBNUC ', LOCATION   00153.
     REAL SECTION    'ACCESS' - DELETED.  REFERS TO DECK 'SPC    ', LOCATION   16617.
```

```
                         M E M O R Y   M A P

                      OF LINK NUMBER  1,( 13I     )
                 ORIGIN OF THIS LINK AT  DECK '13I    '


SYSTEM, INCLUDING IOCS                              00000 THRU  12252

    NUMBER OF FILES -  NONE

OBJECT PROGRAM                                      12352 THRU  64004

     1.  DECK 'SPC    '            12352
     2.  DECK 'DSSCAN'             17001
     3.  SUBR 'INSYFB'             00000
     4.  SUBR 'OUSYFB'             00000
     5.  SUBR 'POSTX '             17040
     6.  SUBR 'PPSYFB'             00000
     7.  SUBR 'F05    '            17152
     8.  SUBR 'F06    '            17153
     9.  SUBR 'IOS    '            17154
    10.  SUBR 'RWD    '            17452
    11.  SUBR 'ACV    '            20203
    12.  SUBR 'XCV    '            20233
    13.  SUBR 'FPT    '            20251
    14.  SUBR 'XEM    '   *        20467
    15.  SUBR 'XIT    '            20700
    16.  SUBR 'DMP    '            20702
    17.  SUBR 'RDH44 '             22572
    18.  DECK '13I    '   *        24232

   (* - INSERTIONS OR DELETIONS MADE IN THIS DECK)

INPUT - OUTPUT BUFFERS                              76157 THRU  77776

UNUSED CORE                                         64005 THRU  76147
```

```
SPACE LIBRARY - VER. 2, MOC. 1                    IBLDR -- JOB   SPACE

                        L O G I C   M A P

     FOR DECKS OF LINK NUMBER  2,( 12I     )

    REAL SECTION    'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.

DECK '12I    ' ASSIGNED ABSOLUTE ORIGIN 24232.  ADJUSTED LENGTH IS 34271.
    VIRTUAL SECTION 'BCD    ' - REFERS TO DECK 'RCH44 ', LOCATION   22760.
    VIRTUAL SECTION 'STOR   ' - REFERS TO DECK 'STOR  ', LOCATION   60531.
    VIRTUAL SECTION 'ATHRUZ' - REFERS TO DECK 'ATHRUZ', LOCATION   61265.
    VIRTUAL SECTION 'FIL03.' - REFERS TO DECK 'FTC03 ', LOCATION   60576.
    VIRTUAL SECTION 'UNSIFT' - REFERS TO DECK 'UNSIFT', LOCATION   6C561.
    VIRTUAL SECTION 'DSSCAN' - REFERS TO DECK 'DSSCAN', LOCATION   17007.
    VIRTUAL SECTION 'SPACE ' - REFERS TO DECK 'SPC   ', LOCATION   13533.
    VIRTUAL SECTION 'ABOUT1' - REFERS TO DECK 'SPC   ', LOCATION   14252.
    VIRTUAL SECTION 'ABOUT2' - REFERS TO DECK 'SPC   ', LOCATION   14256.
    VIRTUAL SECTION 'BOOL   ' - REFERS TO DECK 'BOOL  ', LOCATION   61307.
    VIRTUAL SECTION 'SKIPR ' - REFERS TO DECK 'SPC   ', LOCATION   14262.
    VIRTUAL SECTION 'FIL06.' - REFERS TO DECK 'F06   ', LOCATION   17153.
    VIRTUAL SECTION 'READB1' - REFERS TO DECK 'SPC   ', LOCATION   14242.
    VIRTUAL SECTION 'ENDF   ' - REFERS TO DECK 'SPC   ', LOCATION   14272.
    VIRTUAL SECTION 'READB2' - REFERS TO DECK 'SPC   ', LOCATION   14246.
    VIRTUAL SECTION 'HOLOCT' - REFERS TO DECK 'HLCT  ', LOCATION   61330.
    VIRTUAL SECTION 'HNLIC.' - REFERS TO DECK 'RWD   ', LOCATION   17573.
    VIRTUAL SECTION 'FILIO.' - REFERS TO DECK 'RWD   ', LOCATION   20141.
    VIRTUAL SECTION 'STHIO.' - REFERS TO DECK 'RWD   ', LOCATION   17472.
    VIRTUAL SECTION 'IOHEF.' - REFERS TO DECK 'RWD   ', LOCATION   17665.
    VIRTUAL SECTION 'IOHIO.' - REFERS TO DECK 'RWD   ', LOCATION   17702.
    VIRTUAL SECTION 'IOHHC.' - REFERS TO DECK 'HCV   ', LOCATION   60577.
    VIRTUAL SECTION 'IOHXC.' - REFERS TO DECK 'XCV   ', LOCATION   20233.
    VIRTUAL SECTION 'IOHAC.' - REFERS TO DECK 'ACV   ', LOCATION   20203.
    VIRTUAL SECTION 'IOHIC.' - REFERS TO DECK 'ICV   ', LOCATION   60716.
    VIRTUAL SECTION 'IOHRP.' - REFERS TO DECK 'RWD   ', LOCATION   17646.
    VIRTUAL SECTION 'IOHLP.' - REFERS TO DECK 'RWD   ', LOCATION   17625.
    VIRTUAL SECTION 'ERROU.' - REFERS TO DECK 'XEM   ', LOCATION   20470.
    VIRTUAL SECTION 'S.JXIT' - REFERS TO DECK 'POSTX ', LOCATION   17040.
    VIRTUAL SECTION 'SETFP.' - REFERS TO DECK 'FPT   ', LOCATION   20251.
    REAL SECTION    'ACCESS' - DELETED.  REFERS TO DECK 'SPC   ', LOCATION   16617.

DECK 'STOR   ' ASSIGNED ABSOLUTE ORIGIN 60523.  ADJUSTED LENGTH IS C0030.
    VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
    REAL SECTION    'STOR   ' - ASSIGNED ABSOLUTE ORIGIN 60531.

DECK 'UNSIFT' ASSIGNED ABSOLUTE ORIGIN 60553.  ADJUSTED LENGTH IS CC023.
    VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
    REAL SECTION    'UNSIFT' - ASSIGNED ABSOLUTE ORIGIN 60561.

DECK 'FTC03 ' ASSIGNED ABSOLUTE ORIGIN 60576.  ADJUSTED LENGTH IS CC001.
    VIRTUAL SECTION 'S.FBCU' - REFERS TO DECK 'OUSYFB', LOCATION   12304.
    REAL SECTION    'FIL03.' - ASSIGNED ABSOLUTE ORIGIN 60576.

DEC  'HCV   ' ASSIGNED ABSOLUTE ORIGIN 60577.  ADJUSTED LENGTH IS CC117.
    VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS   ', LOCATION   17430.
    VIRTUAL SECTION 'CVCEL.' - REFERS TO DECK 'RWD   ', LOCATION   20165.
    VIRTUAL SECTION 'BU1.   ' - REFERS TO DECK 'RWD   ', LOCATION   20037.
```

```
     VIRTUAL SECTION 'BU22. ' - REFERS TO DECK 'RWD   ', LOCATION   20053.
     VIRTUAL SECTION 'SC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20021.
     VIRTUAL SECTION 'GETCH.' - REFERS TO DECK 'RWD   ', LOCATION   20151.
     REAL SECTION    'IOHHC.' - ASSIGNED ABSOLUTE ORIGIN 60577.

DECK 'ICV   ' ASSIGNED ABSOLUTE ORIGIN 60716.  ADJUSTED LENGTH IS 00020.
     VIRTUAL SECTION 'CCQ.  ' - REFERS TO DECK 'RWD   ', LOCATION   20031.
     VIRTUAL SECTION 'SC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20021.
     VIRTUAL SECTION 'AST.  ' - REFERS TO DECK 'INTJ  ', LOCATION   61106.
     VIRTUAL SECTION 'IC2.  ' - REFERS TO DECK 'INTJ  ', LOCATION   61057.
     VIRTUAL SECTION 'IC10. ' - REFERS TO DECK 'INTJ  ', LOCATION   61073.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS   ', LOCATION   17430.
     VIRTUAL SECTION 'IOHCM.' - REFERS TO DECK 'RWD   ', LOCATION   17537.
     VIRTUAL SECTION 'IOHCT.' - REFERS TO DECK 'RWD   ', LOCATION   17615.
     VIRTUAL SECTION 'IOHDB.' - REFERS TO DECK 'INTJ  ', LOCATION   61126.
     REAL SECTION    'IOHIC.' - ASSIGNED ABSOLUTE ORIGIN 60716.
     REAL SECTION    'IOHIT.' - ASSIGNED ABSOLUTE ORIGIN 60722.

DECK 'INTJ  ' ASSIGNED ABSOLUTE ORIGIN 60736.  ADJUSTED LENGTH IS 00321.
     VIRTUAL SECTION 'CVCEL.' - REFERS TO DECK 'RWD   ', LOCATION   20165.
     VIRTUAL SECTION 'BU1.  ' - REFERS TO DECK 'RWD   ', LOCATION   20037.
     VIRTUAL SECTION 'BU4.  ' - REFERS TO DECK 'RWD   ', LOCATION   20044.
     VIRTUAL SECTION 'CCQ.  ' - REFERS TO DECK 'RWD   ', LOCATION   20031.
     VIRTUAL SECTION 'DE60. ' - REFERS TO DECK 'RWD   ', LOCATION   20077.
     VIRTUAL SECTION 'DE70. ' - REFERS TO DECK 'RWD   ', LOCATION   20100.
     VIRTUAL SECTION 'XC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20110.
     VIRTUAL SECTION 'FEXEM.' - REFERS TO DECK 'XEM   ', LOCATION   20473.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS   ', LOCATION   17430.
     VIRTUAL SECTION 'GETCH.' - REFERS TO DECK 'RWD   ', LOCATION   20151.
     REAL SECTION    'J.    ' - ASSIGNED ABSOLUTE ORIGIN 60747.
     REAL SECTION    'IC2.  ' - ASSIGNED ABSOLUTE ORIGIN 61057.
     REAL SECTION    'IC10. ' - ASSIGNED ABSOLUTE ORIGIN 61073.
     REAL SECTION    'AST.  ' - ASSIGNED ABSOLUTE ORIGIN 61106.
     REAL SECTION    'AST1. ' - ASSIGNED ABSOLUTE ORIGIN 61107.
     REAL SECTION    'DE30. ' - ASSIGNED ABSOLUTE ORIGIN 61113.
     REAL SECTION    'IOHDB.' - ASSIGNED ABSOLUTE ORIGIN 61126.
     REAL SECTION    'DBC1. ' - ASSIGNED ABSOLUTE ORIGIN 61140.
     REAL SECTION    'DBC5. ' - ASSIGNED ABSOLUTE ORIGIN 61160.
     REAL SECTION    'ANACH.' - ASSIGNED ABSOLUTE ORIGIN 61207.
     REAL SECTION    'SG1.  ' - ASSIGNED ABSOLUTE ORIGIN 61226.
     REAL SECTION    'DECEL.' - ASSIGNED ABSOLUTE ORIGIN 61250.

DECK 'ATHRUZ' ASSIGNED ABSOLUTE ORIGIN 61257.  ADJUSTED LENGTH IS 00022.
     VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     REAL SECTION    'ATHRUZ' - ASSIGNED ABSOLUTE ORIGIN 61265.

DECK 'POOL  ' ASSIGNED ABSOLUTE ORIGIN 61301.  ADJUSTED LENGTH IS 00021.
     VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     REAL SECTION    'BOOL  ' - ASSIGNED ABSOLUTE ORIGIN 61307.

DECK 'HLCT  ' ASSIGNED ABSOLUTE ORIGIN 61322.  ADJUSTED LENGTH IS 00047.
     VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     REAL SECTION    'HOLOCT' - ASSIGNED ABSOLUTE ORIGIN 61330.
```

```
SPACE LIBRARY - VER. 2, MOD. 1                    IBLDR -- JOB   SPACE

                          M E M O R Y   M A P

                       OF LINK NUMBER  2,( 121     )
                   ORIGIN OF THIS LINK AT  DECK '121    '


SYSTEM, INCLUDING IOCS                                00000 THRU  12252

     NUMBER OF FILES -   NONE

OBJECT PROGRAM                                        12352 THRU  61370

      1.  DECK 'SPC    '            12352
      2.  DECK 'DSSCAN'             17001
      3.  SUBR 'INSYFB'             00000
      4.  SUBR 'OUSYFB'             00000
      5.  SUBR 'POSTX '             17040
      6.  SUBR 'PPSYFB'             00000
      7.  SUBR 'F05    '            17152
      8.  SUBR 'F06    '            17153
      9.  SUBR 'IOS    '            17154
     10.  SUBR 'RWD    '            17452
     11.  SUBR 'ACV    '            20203
     12.  SUBR 'XCV    '            20233
     13.  SUBR 'FPT    '            20251
     14.  SUBR 'XEM    '   *        20467
     15.  SUBR 'XIT    '            20700
     16.  SUBR 'DMP    '            20702
     17.  SUBR 'RDH44 '             22572
     18.  DECK '121    '   *        24232
     19.  DECK 'STOR   '            60523
     20.  DECK 'UNSIFT'             60553
     21.  DECK 'FTC03 '             60576
     22.  SUBR 'HCV    '            60577
     23.  SUBR 'ICV    '            60716
     24.  SUBR 'INTJ   '            60736
     25.  SUBR 'ATHRUZ'             61257
     26.  SUBR 'BOOL   '            61301
     27.  SUBR 'HLCT   '            61322

    (* - INSERTIONS OR DELETIONS MADE IN THIS DECK)

INPUT - OUTPUT BUFFERS                                76157 THRU  77776

UNUSED CORE                                           61371 THRU  76147
```

```
SPACE LIBRARY - VER. 2, MOD. 1                    IBLDR -- JOB   SPACE

                          L O G I C   M A P

     FOR DECKS OF LINK NUMBER   3,( 611      )

    REAL SECTION     'S.SLCC' - ASSIGNED ABSOLUTE ORIGIN 12253.

DECK '611    ' ASSIGNED ABSOLUTE ORIGIN 24232.  ADJUSTED LENGTH IS 22275.
    VIRTUAL SECTION 'BCD    ' - REFERS TO DECK 'RCH44 ', LOCATION   22760.
    VIRTUAL SECTION 'DELETE' - REFERS TO DECK 'DELETE', LOCATION   47243.
    VIRTUAL SECTION 'SORTC1' - REFERS TO DECK 'SORTC1', LOCATION   46535.
    VIRTUAL SECTION 'ATHRUZ' - REFERS TO DECK 'ATHRUZ', LOCATION   51327.
    VIRTUAL SECTION 'FILENO' - REFERS TO DECK 'FILENO', LOCATION   47015.
    VIRTUAL SECTION 'DSSCAN' - REFERS TO DECK 'DSSCAN', LOCATION   17007.
    VIRTUAL SECTION 'ALLIN1' - REFERS TO DECK 'ALLIN1', LOCATION   46704.
    VIRTUAL SECTION 'CNVRT ' - REFERS TO DECK 'CNVRT ', LOCATION   47064.
    VIRTUAL SECTION 'SPACE ' - REFERS TO DECK 'SPC    ', LOCATION   13533.
    VIRTUAL SECTION 'ABOUT1' - REFERS TO DECK 'SPC    ', LOCATION   14252.
    VIRTUAL SECTION 'ABOUT2' - REFERS TO DECK 'SPC    ', LOCATION   14256.
    VIRTUAL SECTION 'BOOL   ' - REFERS TO DECK 'BOOL   ', LOCATION   51351.
    VIRTUAL SECTION 'EXP    ' - REFERS TO DECK 'XPN    ', LOCATION   51072.
    VIRTUAL SECTION 'GSMRGE' - REFERS TO DECK 'GSMRGE', LOCATION   47332.
    VIRTUAL SECTION 'SKIPR ' - REFERS TO DECK 'SPC    ', LOCATION   14262.
    VIRTUAL SECTION 'FIL06.' - REFERS TO DECK 'F06    ', LOCATION   17153.
    VIRTUAL SECTION 'READB1' - REFERS TO DECK 'SPC    ', LOCATION   14242.
    VIRTUAL SECTION 'ENDF   ' - REFERS TO DECK 'SPC    ', LOCATION   14272.
    VIRTUAL SECTION 'READB2' - REFERS TO DECK 'SPC    ', LOCATION   14246.
    VIRTUAL SECTION 'COMPAR' - REFERS TO DECK 'COMPAR', LOCATION   46626.
    VIRTUAL SECTION 'HOLCCT' - REFERS TO DECK 'HLCT   ', LOCATION   51372.
    VIRTUAL SECTION '.EXP2.' - REFERS TO DECK 'XP2    ', LOCATION   50735.
    VIRTUAL SECTION '.EXP3.' - REFERS TO DECK 'XP3    ', LOCATION   51020.
    VIRTUAL SECTION 'HNLIO.' - REFERS TO DECK 'RWD    ', LOCATION   17573.
    VIRTUAL SECTION 'FILIO.' - REFERS TO DECK 'RWD    ', LOCATION   20141.
    VIRTUAL SECTION 'SLOIO.' - REFERS TO DECK 'SLO    ', LOCATION   50703.
    VIRTUAL SECTION 'STHIO.' - REFERS TO DECK 'RWD    ', LOCATION   17472.
    VIRTUAL SECTION 'IOHEF.' - REFERS TO DECK 'RWD    ', LOCATION   17665.
    VIRTUAL SECTION 'IOHIO.' - REFERS TO DECK 'RWD    ', LOCATION   17702.
    VIRTUAL SECTION 'IOHHC.' - REFERS TO DECK 'HCV    ', LOCATION   47636.
    VIRTUAL SECTION 'IOHXC.' - REFERS TO DECK 'XCV    ', LOCATION   20233.
    VIRTUAL SECTION 'IOHAC.' - REFERS TO DECK 'ACV    ', LOCATION   20203.
    VIRTUAL SECTION 'IOHIC.' - REFERS TO DECK 'ICV    ', LOCATION   47755.
    VIRTUAL SECTION 'IOHFC.' - REFERS TO DECK 'FCV    ', LOCATION   47545.
    VIRTUAL SECTION 'IOHEC.' - REFERS TO DECK 'ECV    ', LOCATION   47344.
    VIRTUAL SECTION 'IOHSF.' - REFERS TO DECK 'RWD    ', LOCATION   17677.
    VIRTUAL SECTION 'IOHRP.' - REFERS TO DECK 'RWD    ', LOCATION   17646.
    VIRTUAL SECTION 'IOHLP.' - REFERS TO DECK 'RWD    ', LOCATION   17625.
    VIRTUAL SECTION 'ERROU.' - REFERS TO DECK 'XEM    ', LOCATION   20470.
    VIRTUAL SECTION 'S.JXIT' - REFERS TO DECK 'POSTX ', LOCATION   17040.
    VIRTUAL SECTION 'SETFP.' - REFERS TO DECK 'FPT    ', LOCATION   20251.
    REAL SECTION     'ACCESS' - DELETED.  REFERS TO DECK 'SPC    ', LOCATION   16617.
    REAL SECTION     'ARRAYS' - ASSIGNED ABSOLUTE ORIGIN 35013.
    REAL SECTION     'CONSNT' - ASSIGNED ABSOLUTE ORIGIN 46476.

D   K 'SORTC1' ASSIGNED ABSOLUTE ORIGIN 46527.  ADJUSTED LENGTH IS 00071.
    VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
    REAL SECTION     'SORTC1' - ASSIGNED ABSOLUTE ORIGIN 46535.
```

```
DECK 'COMPAR' ASSIGNED ABSOLUTE ORIGIN 46620.  ADJUSTED LENGTH IS 00056.
     VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     REAL SECTION    'COMPAR' - ASSIGNED ABSOLUTE ORIGIN 46626.

DECK 'ALLINI' ASSIGNED ABSOLUTE ORIGIN 46676.  ADJUSTED LENGTH IS 00111.
     VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     REAL SECTION    'ALLINI' - ASSIGNED ABSOLUTE ORIGIN 46704.

DECK 'FILENO' ASSIGNED ABSOLUTE ORIGIN 47007.  ADJUSTED LENGTH IS 00047.
     VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     REAL SECTION    'FILENO' - ASSIGNED ABSOLUTE ORIGIN 47015.

DECK 'CNVRT ' ASSIGNED ABSOLUTE ORIGIN 47056.  ADJUSTED LENGTH IS 00157.
     VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     REAL SECTION    'CNVRT ' - ASSIGNED ABSOLUTE ORIGIN 47064.

DECK 'DELETE' ASSIGNED ABSOLUTE ORIGIN 47235.  ADJUSTED LENGTH IS 00067.
     VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     REAL SECTION    'DELETE' - ASSIGNED ABSOLUTE ORIGIN 47243.

DECK 'GSMRGE' ASSIGNED ABSOLUTE ORIGIN 47324.  ADJUSTED LENGTH IS 00020.
     VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     REAL SECTION    'GSMRGE' - ASSIGNED ABSOLUTE ORIGIN 47332.

DECK 'ECV   ' ASSIGNED ABSOLUTE ORIGIN 47344.  ADJUSTED LENGTH IS 00201.
     VIRTUAL SECTION 'AST.  ' - REFERS TO DECK 'INTJ  ', LOCATION   50145.
     VIRTUAL SECTION 'DE30. ' - REFERS TO DECK 'INTJ  ', LOCATION   50152.
     VIRTUAL SECTION 'IC10. ' - REFERS TO DECK 'INTJ  ', LOCATION   50132.
     VIRTUAL SECTION 'J.    ' - REFERS TO DECK 'INTJ  ', LOCATION   50006.
     VIRTUAL SECTION 'BU4.  ' - REFERS TO DECK 'RWD   ', LOCATION   20044.
     VIRTUAL SECTION 'BU22. ' - REFERS TO DECK 'RWD   ', LOCATION   20053.
     VIRTUAL SECTION 'CCQ.  ' - REFERS TO DECK 'RWD   ', LOCATION   20031.
     VIRTUAL SECTION 'DE60. ' - REFERS TO DECK 'RWD   ', LOCATION   20077.
     VIRTUAL SECTION 'DE70. ' - REFERS TO DECK 'RWD   ', LOCATION   20100.
     VIRTUAL SECTION 'SC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20021.
     VIRTUAL SECTION 'XC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20110.
     VIRTUAL SECTION 'FFC10.' - REFERS TO DECK 'FFC   ', LOCATION   50316.
     VIRTUAL SECTION 'FFC12.' - REFERS TO DECK 'FFC   ', LOCATION   50325.
     VIRTUAL SECTION 'FFC24.' - REFERS TO DECK 'FFC   ', LOCATION   50346.
     VIRTUAL SECTION 'FLTSW.' - REFERS TO DECK 'FFC   ', LOCATION   50512.
     VIRTUAL SECTION 'FFC30.' - REFERS TO DECK 'FFC   ', LOCATION   50466.
     VIRTUAL SECTION 'DE41. ' - REFERS TO DECK 'FFC   ', LOCATION   50471.
     VIRTUAL SECTION 'DE44. ' - REFERS TO DECK 'FFC   ', LOCATION   50500.
     VIRTUAL SECTION 'CVCEL.' - REFERS TO DECK 'RWD   ', LOCATION   20165.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS   ', LOCATION   17430.
     VIRTUAL SECTION 'IOHCM.' - REFERS TO DECK 'RWD   ', LOCATION   17537.
     VIRTUAL SECTION 'DECEL.' - REFERS TO DECK 'INTJ  ', LOCATION   50307.
     REAL SECTION    'IOHEC.' - ASSIGNED ABSOLUTE ORIGIN 47344.
     REAL SECTION    'IOHET.' - ASSIGNED ABSOLUTE ORIGIN 47350.
     REAL SECTION    'DC1.  ' - ASSIGNED ABSOLUTE ORIGIN 47355.
     REAL SECTION    'DE.   ' - ASSIGNED ABSOLUTE ORIGIN 47373.
     REAL SECTION    'FFC9. ' - ASSIGNED ABSOLUTE ORIGIN 47476.
     REAL SECTION    'FFC21.' - ASSIGNED ABSOLUTE ORIGIN 47544.

DECK 'FCV   ' ASSIGNED ABSOLUTE ORIGIN 47545.  ADJUSTED LENGTH IS 00071.
```

```
     VIRTUAL SECTION 'FLTSW.' - REFERS TO DECK 'FFC   ', LOCATION   50512.
     VIRTUAL SECTION 'BU22. ' - REFERS TO DECK 'RWD   ', LOCATION   20053.
     VIRTUAL SECTION 'CCQ.  ' - REFERS TO DECK 'RWD   ', LOCATION   20031.
     VIRTUAL SECTION 'SC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20021.
     VIRTUAL SECTION 'XC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20110.
     VIRTUAL SECTION 'AST1. ' - REFERS TO DECK 'INTJ  ', LOCATION   50146.
     VIRTUAL SECTION 'DE41. ' - REFERS TO DECK 'FFC   ', LOCATION   50471.
     VIRTUAL SECTION 'DE44. ' - REFERS TO DECK 'FFC   ', LOCATION   50500.
     VIRTUAL SECTION 'FFC10.' - REFERS TO DECK 'FFC   ', LOCATION   50316.
     VIRTUAL SECTION 'FFC24.' - REFERS TO DECK 'FFC   ', LOCATION   50346.
     VIRTUAL SECTION 'IOHCM.' - REFERS TO DECK 'RWD   ', LOCATION   17537.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS   ', LOCATION   17430.
     VIRTUAL SECTION 'CVCEL.' - REFERS TO DECK 'RWD   ', LOCATION   20165.
     VIRTUAL SECTION 'DECEL.' - REFERS TO DECK 'INTJ  ', LOCATION   50307.
     REAL SECTION    'IOHFC.' - ASSIGNED ABSOLUTE ORIGIN 47545.
     REAL SECTION    'IOHFT.' - ASSIGNED ABSOLUTE ORIGIN 47546.
     REAL SECTION    'FG.   ' - ASSIGNED ABSOLUTE ORIGIN 47630.

DECK 'HCV   ' ASSIGNED ABSOLUTE ORIGIN 47636.  ADJUSTED LENGTH IS 00117.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS   ', LOCATION   17430.
     VIRTUAL SECTION 'CVCEL.' - REFERS TO DECK 'RWD   ', LOCATION   20165.
     VIRTUAL SECTION 'BU1.  ' - REFERS TO DECK 'RWD   ', LOCATION   20037.
     VIRTUAL SECTION 'BU22. ' - REFERS TO DECK 'RWD   ', LOCATION   20053.
     VIRTUAL SECTION 'SC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20021.
     VIRTUAL SECTION 'GETCH.' - REFERS TO DECK 'RWD   ', LOCATION   20151.
     REAL SECTION    'IOHHC.' - ASSIGNED ABSOLUTE ORIGIN 47636.

DECK 'ICV   ' ASSIGNED ABSOLUTE ORIGIN 47755.  ADJUSTED LENGTH IS 00020.
     VIRTUAL SECTION 'CCQ.  ' - REFERS TO DECK 'RWD   ', LOCATION   20031.
     VIRTUAL SECTION 'SC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20021.
     VIRTUAL SECTION 'AST.  ' - REFERS TO DECK 'INTJ  ', LOCATION   50145.
     VIRTUAL SECTION 'IC2.  ' - REFERS TO DECK 'INTJ  ', LOCATION   50116.
     VIRTUAL SECTION 'IC10. ' - REFERS TO DECK 'INTJ  ', LOCATION   50132.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS   ', LOCATION   17430.
     VIRTUAL SECTION 'IOHCM.' - REFERS TO DECK 'RWD   ', LOCATION   17537.
     VIRTUAL SECTION 'IOHCT.' - REFERS TO DECK 'RWD   ', LOCATION   17615.
     VIRTUAL SECTION 'IOHDB.' - REFERS TO DECK 'INTJ  ', LOCATION   50165.
     REAL SECTION    'IOHIC.' - ASSIGNED ABSOLUTE ORIGIN 47755.
     REAL SECTION    'IOHIT.' - ASSIGNED ABSOLUTE ORIGIN 47761.

DECK 'INTJ  ' ASSIGNED ABSOLUTE ORIGIN 47775.  ADJUSTED LENGTH IS 00321.
     VIRTUAL SECTION 'CVCEL.' - REFERS TO DECK 'RWD   ', LOCATION   20165.
     VIRTUAL SECTION 'BU1.  ' - REFERS TO DECK 'RWD   ', LOCATION   20037.
     VIRTUAL SECTION 'BU4.  ' - REFERS TO DECK 'RWD   ', LOCATION   20044.
     VIRTUAL SECTION 'CCQ.  ' - REFERS TO DECK 'RWD   ', LOCATION   20031.
     VIRTUAL SECTION 'DE60. ' - REFERS TO DECK 'RWD   ', LOCATION   20077.
     VIRTUAL SECTION 'DE70. ' - REFERS TO DECK 'RWD   ', LOCATION   20100.
     VIRTUAL SECTION 'XC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20110.
     VIRTUAL SECTION 'FEXEM.' - REFERS TO DECK 'XEM   ', LOCATION   20473.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS   ', LOCATION   17430.
     VIRTUAL SECTION 'GETCH.' - REFERS TO DECK 'RWD   ', LOCATION   20151.
     REAL SECTION    'J.    ' - ASSIGNED ABSOLUTE ORIGIN 50006.
     REAL SECTION    'IC2.  ' - ASSIGNED ABSOLUTE ORIGIN 50116.
     REAL SECTION    'IC10. ' - ASSIGNED ABSOLUTE ORIGIN 50132.
     REAL SECTION    'AST.  ' - ASSIGNED ABSOLUTE ORIGIN 50145.
```

```
     REAL SECTION    'AST1. ' - ASSIGNED ABSOLUTE ORIGIN 50146.
     REAL SECTION    'DE30. ' - ASSIGNED ABSOLUTE ORIGIN 50152.
     REAL SECTION    'IOHDB.' - ASSIGNED ABSOLUTE ORIGIN 50165.
     REAL SECTION    'DBC1. ' - ASSIGNED ABSOLUTE ORIGIN 50177.
     REAL SECTION    'DBC5. ' - ASSIGNED ABSOLUTE ORIGIN 50217.
     REAL SECTION    'ANACH.' - ASSIGNED ABSOLUTE ORIGIN 50246.
     REAL SECTION    'SG1.  ' - ASSIGNED ABSOLUTE ORIGIN 50265.
     REAL SECTION    'DECEL.' - ASSIGNED ABSOLUTE ORIGIN 50307.

DECK 'FFC   ' ASSIGNED ABSOLUTE ORIGIN 50316.  ADJUSTED LENGTH IS 00365.
     VIRTUAL SECTION 'IC10. ' - REFERS TO DECK 'INTJ  ', LOCATION   50132.
     VIRTUAL SECTION 'BU22. ' - REFERS TO DECK 'RWD   ', LOCATION   20053.
     VIRTUAL SECTION 'DE60. ' - REFERS TO DECK 'RWD   ', LOCATION   20077.
     VIRTUAL SECTION 'DE70. ' - REFERS TO DECK 'RWD   ', LOCATION   20100.
     VIRTUAL SECTION 'IC2.  ' - REFERS TO DECK 'INTJ  ', LOCATION   50116.
     VIRTUAL SECTION 'J.    ' - REFERS TO DECK 'INTJ  ', LOCATION   50006.
     VIRTUAL SECTION 'DECEL.' - REFERS TO DECK 'INTJ  ', LOCATION   50307.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS   ', LOCATION   17430.
     VIRTUAL SECTION 'IOHCT.' - REFERS TO DECK 'RWD   ', LOCATION   17615.
     VIRTUAL SECTION 'IOHDB.' - REFERS TO DECK 'INTJ  ', LOCATION   50165.
     REAL SECTION    'FFC10.' - ASSIGNED ABSOLUTE ORIGIN 50316.
     REAL SECTION    'FFC11.' - ASSIGNED ABSOLUTE ORIGIN 50322.
     REAL SECTION    'FFC12.' - ASSIGNED ABSOLUTE ORIGIN 50325.
     REAL SECTION    'FFC23.' - ASSIGNED ABSOLUTE ORIGIN 50344.
     REAL SECTION    'FFC24.' - ASSIGNED ABSOLUTE ORIGIN 50346.
     REAL SECTION    'FFC30.' - ASSIGNED ABSOLUTE ORIGIN 50466.
     REAL SECTION    'DE41. ' - ASSIGNED ABSOLUTE ORIGIN 50471.
     REAL SECTION    'DE44. ' - ASSIGNED ABSOLUTE ORIGIN 50500.
     REAL SECTION    'FLTSW.' - ASSIGNED ABSOLUTE ORIGIN 50512.
     REAL SECTION    'BDE.  ' - ASSIGNED ABSOLUTE ORIGIN 50537.
     REAL SECTION    'CMC.  ' - ASSIGNED ABSOLUTE ORIGIN 50606.
     REAL SECTION    'IOHPW.' - ASSIGNED ABSOLUTE ORIGIN 50621.

DECK 'SLO   ' ASSIGNED ABSOLUTE ORIGIN 50703.  ADJUSTED LENGTH IS 00025.
     VIRTUAL SECTION 'ERLOC.' - REFERS TO DECK 'XEM   ', LOCATION   20472.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS   ', LOCATION   17430.
     REAL SECTION    'SLOIO.' - ASSIGNED ABSOLUTE ORIGIN 50703.

DECK 'XP2   ' ASSIGNED ABSOLUTE ORIGIN 50730.  ADJUSTED LENGTH IS 00067.
     VIRTUAL SECTION 'FEXEM.' - REFERS TO DECK 'XEM   ', LOCATION   20473.
     VIRTUAL SECTION 'ERLOC.' - REFERS TO DECK 'XEM   ', LOCATION   20472.
     REAL SECTION    '.EXP2.' - ASSIGNED ABSOLUTE ORIGIN 50735.

DECK 'XP3   ' ASSIGNED ABSOLUTE ORIGIN 51017.  ADJUSTED LENGTH IS 00051.
     VIRTUAL SECTION 'FEXEM.' - REFERS TO DECK 'XEM   ', LOCATION   20473.
     VIRTUAL SECTION 'ERLOC.' - REFERS TO DECK 'XEM   ', LOCATION   20472.
     VIRTUAL SECTION 'ALOG  ' - REFERS TO DECK 'LOG   ', LOCATION   51177.
     VIRTUAL SECTION 'EXP   ' - REFERS TO DECK 'XPN   ', LOCATION   51072.
     REAL SECTION    '.EXP3.' - ASSIGNED ABSOLUTE ORIGIN 51020.

DECK 'XPN   ' ASSIGNED ABSOLUTE ORIGIN 51070.  ADJUSTED LENGTH IS 00103.
     VIRTUAL SECTION 'FEXEM.' - REFERS TO DECK 'XEM   ', LOCATION   20473.
     VIRTUAL SECTION 'ERLOC.' - REFERS TO DECK 'XEM   ', LOCATION   20472.
     REAL SECTION    'EXP   ' - ASSIGNED ABSOLUTE ORIGIN 51072.
```

```
DECK 'LOG    ' ASSIGNED ABSOLUTE ORIGIN 51173.  ADJUSTED LENGTH IS 00126.
     VIRTUAL SECTION 'FEXEM.' - REFERS TO DECK 'XEM    ', LOCATION   20473.
     VIRTUAL SECTION 'ERLOC.' - REFERS TO DECK 'XEM    ', LOCATION   20472.
     REAL SECTION    'ALOG10' - ASSIGNED ABSOLUTE ORIGIN 51173.
     REAL SECTION    'ALOG  ' - ASSIGNED ABSOLUTE ORIGIN 51177.

DECK 'ATHRUZ' ASSIGNED ABSOLUTE ORIGIN 51321.  ADJUSTED LENGTH IS 00022.
     VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     REAL SECTION    'ATHRUZ' - ASSIGNED ABSOLUTE ORIGIN 51327.

DECK 'POOL   ' ASSIGNED ABSOLUTE ORIGIN 51343.  ADJUSTED LENGTH IS 00021.
     VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     REAL SECTION    'BOOL  ' - ASSIGNED ABSOLUTE ORIGIN 51351.

DECK 'HLCT   ' ASSIGNED ABSOLUTE ORIGIN 51364.  ADJUSTED LENGTH IS 00047.
     VIRTUAL SECTION 'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.
     REAL SECTION    'HOLOCT' - ASSIGNED ABSOLUTE ORIGIN 51372.
```

```
SPACE LIBRARY - VER. 2, MOD. 1                        IBLDR -- JOB   SPACE

                              M E M O R Y   M A P

                          OF LINK NUMBER  3,( 61I     )
                     ORIGIN OF THIS LINK AT  DECK '61I    '


SYSTEM, INCLUDING IOCS                                 00000 THRU  12252

     NUMBER OF FILES -  NONE

OBJECT PROGRAM                                         12352 THRU  51432

     1.   DECK 'SPC   '              12352
     2.   DECK 'DSSCAN'              17001
     3.   SUBR 'INSYFB'              00000
     4.   SUBR 'OUSYFB'              00000
     5.   SUBR 'POSTX '              17040
     6.   SUBR 'PPSYFB'              00000
     7.   SUBR 'F05   '              17152
     8.   SUBR 'F06   '              17153
     9.   SUBR 'IOS   '              17154
    10.   SUBR 'RWD   '              17452
    11.   SUBR 'ACV   '              20203
    12.   SUBR 'XCV   '              20233
    13.   SUBR 'FPT   '              20251
    14.   SUBR 'XEM   '  •           20467
    15.   SUBR 'XIT   '              20700
    16.   SUBR 'DMP   '              20702
    17.   SUBR 'RDH44 '              22572
    18.   DECK '61I   '  •           24232
    19.   DECK 'SORTC1'              46527
    20.   DECK 'COMPAR'              46620
    21.   DECK 'ALLIN1'              46676
    22.   DECK 'FILENO'  •           47007
    23.   DECK 'CNVRT '  •           47056
    24.   DECK 'DELETE'              47235
    25.   DECK 'GSMRGE'              47324
    26.   SUBR 'ECV   '              47344
    27.   SUBR 'FCV   '              47545
    28.   SUBR 'HCV   '              47636
    29.   SUBR 'ICV   '              47755
    30.   SUBR 'INTJ  '              47775
    31.   SUBR 'FFC   '              50316
    32.   SUBR 'SLC   '              50703
    33.   SUBR 'XP2   '              50730
    34.   SUBR 'XP3   '              51017
    35.   SUBR 'XPN   '              51070
    36.   SUBR 'LOG   '              51173
    37.   SUBR 'ATHRUZ'              51321
    38.   SUBR 'BOOL  '              51343
    39.   SUBR 'HLCT  '              51364

    • - INSERTIONS OR DELETIONS MADE IN THIS DECK)
```

SPACE LIBRARY - VER. 2, MOD. 1 IBLDR -- JOB SPACE

INPUT - OUTPUT BUFFERS	76157 THRU 77776
UNUSED CORE	51433 THRU 76147

```
                     L O G I C   M A P

      FOR DECKS OF LINK NUMBER  4,( UTLITY )

     REAL SECTION    'S.SLOC' - ASSIGNED ABSOLUTE ORIGIN 12253.

DECK 'UTL   ' ASSIGNED ABSOLUTE ORIGIN 24232.  ADJUSTED LENGTH IS 02347.
     VIRTUAL SECTION 'S.JNAM' - REFERS TO DECK 'IBNUC ', LOCATION   00311.
     VIRTUAL SECTION 'S.SDAT' - REFERS TO DECK 'IBNUC ', LOCATION   00213.
     VIRTUAL SECTION 'S.SLTC' - REFERS TO DECK 'IBNUC ', LOCATION   00210.
     VIRTUAL SECTION 'S.XDVA' - REFERS TO DECK 'IBNUC ', LOCATION   00153.
     VIRTUAL SECTION 'FILIO.' - REFERS TO DECK 'RWD   ', LOCATION   20141.
     VIRTUAL SECTION 'FIL06.' - REFERS TO DECK 'F06   ', LOCATION   17153.
     VIRTUAL SECTION 'HNLIO.' - REFERS TO DECK 'RWD   ', LOCATION   17573.
     VIRTUAL SECTION 'SLOIO.' - REFERS TO DECK 'SLO   ', LOCATION   30127.
     VIRTUAL SECTION 'STHIO.' - REFERS TO DECK 'RWD   ', LOCATION   17472.
     VIRTUAL SECTION 'IOHAC.' - REFERS TO DECK 'ACV   ', LOCATION   20203.
     VIRTUAL SECTION 'IOHEC.' - REFERS TO DECK 'ECV   ', LOCATION   26601.
     VIRTUAL SECTION 'IOHEF.' - REFERS TO DECK 'RWD   ', LOCATION   17665.
     VIRTUAL SECTION 'IOHHC.' - REFERS TO DECK 'HCV   ', LOCATION   27002.
     VIRTUAL SECTION 'IOHIC.' - REFERS TO DECK 'ICV   ', LOCATION   27121.
     VIRTUAL SECTION 'IOHIO.' - REFERS TO DECK 'RWD   ', LOCATION   17702.
     VIRTUAL SECTION 'IOHLP.' - REFERS TO DECK 'RWD   ', LOCATION   17625.
     VIRTUAL SECTION 'IOHOC.' - REFERS TO DECK 'OCV   ', LOCATION   27141.
     VIRTUAL SECTION 'IOHRP.' - REFERS TO DECK 'RWD   ', LOCATION   17646.
     VIRTUAL SECTION 'IOHXC.' - REFERS TO DECK 'XCV   ', LOCATION   20233.
     VIRTUAL SECTION 'ABOUT1' - REFERS TO DECK 'SPC   ', LOCATION   14252.
     VIRTUAL SECTION 'ABOUT2' - REFERS TO DECK 'SPC   ', LOCATION   14256.
     VIRTUAL SECTION 'ENDF  ' - REFERS TO DECK 'SPC   ', LOCATION   14272.
     VIRTUAL SECTION 'OEDIT ' - REFERS TO DECK 'SPC   ', LOCATION   15344.
     VIRTUAL SECTION 'READB1' - REFERS TO DECK 'SPC   ', LOCATION   14242.
     VIRTUAL SECTION 'READB2' - REFERS TO DECK 'SPC   ', LOCATION   14246.
     VIRTUAL SECTION 'SIGNON' - REFERS TO DECK 'SPC   ', LOCATION   12352.
     VIRTUAL SECTION 'SPACE ' - REFERS TO DECK 'SPC   ', LOCATION   13533.
     REAL SECTION    'ACCESS' - DELETED.  REFERS TO DECK 'SPC   ', LOCATION   16617.
     REAL SECTION    'SYSTEM' - DELETED.  REFERS TO DECK 'SPC   ', LOCATION   16476.

DECK 'ECV   ' ASSIGNED ABSOLUTE ORIGIN 26601.  ADJUSTED LENGTH IS 00201.
     VIRTUAL SECTION 'AST.  ' - REFERS TO DECK 'INTJ  ', LOCATION   27370.
     VIRTUAL SECTION 'DE30. ' - REFERS TO DECK 'INTJ  ', LOCATION   27375.
     VIRTUAL SECTION 'IC10. ' - REFERS TO DECK 'INTJ  ', LOCATION   27355.
     VIRTUAL SECTION 'J.    ' - REFERS TO DECK 'INTJ  ', LOCATION   27231.
     VIRTUAL SECTION 'BU4.  ' - REFERS TO DECK 'RWD   ', LOCATION   20044.
     VIRTUAL SECTION 'BU22. ' - REFERS TO DECK 'RWD   ', LOCATION   20053.
     VIRTUAL SECTION 'CCQ.  ' - REFERS TO DECK 'RWD   ', LOCATION   20031.
     VIRTUAL SECTION 'DE60. ' - REFERS TO DECK 'RWD   ', LOCATION   20077.
     VIRTUAL SECTION 'DE70. ' - REFERS TO DECK 'RWD   ', LOCATION   20100.
     VIRTUAL SECTION 'SC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20021.
     VIRTUAL SECTION 'XC.   ' - REFERS TO DECK 'RWD   ', LOCATION   20110.
     VIRTUAL SECTION 'FFC10.' - REFERS TO DECK 'FFC   ', LOCATION   27541.
     VIRTUAL SECTION 'FFC12.' - REFERS TO DECK 'FFC   ', LOCATION   27550.
     VIRTUAL SECTION 'FFC24.' - REFERS TO DECK 'FFC   ', LOCATION   27572.
     VIRTUAL SECTION 'FLTSW.' - REFERS TO DECK 'FFC   ', LOCATION   27736.
     VIRTUAL SECTION 'FFC30.' - REFERS TO DECK 'FFC   ', LOCATION   27712.
     VIRTUAL SECTION 'DE41. ' - REFERS TO DECK 'FFC   ', LOCATION   27715.
```

```
     VIRTUAL SECTION 'DE44. ' - REFERS TO DECK 'FFC    ', LOCATION   27724.
     VIRTUAL SECTION 'CVCEL.' - REFERS TO DECK 'RWD    ', LOCATION   20165.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS    ', LOCATION   17430.
     VIRTUAL SECTION 'IOHCM.' - REFERS TO DECK 'RWD    ', LOCATION   17537.
     VIRTUAL SECTION 'DECEL.' - REFERS TO DECK 'INTJ   ', LOCATION   27532.
     REAL SECTION    'IOHEC.' - ASSIGNED ABSOLUTE ORIGIN 26601.
     REAL SECTION    'IOHET.' - ASSIGNED ABSOLUTE ORIGIN 26605.
     REAL SECTION    'DC1.  ' - ASSIGNED ABSOLUTE ORIGIN 26612.
     REAL SECTION    'DE.   ' - ASSIGNED ABSOLUTE ORIGIN 26630.
     REAL SECTION    'FFC9. ' - ASSIGNED ABSOLUTE ORIGIN 26733.
     REAL SECTION    'FFC21.' - ASSIGNED ABSOLUTE ORIGIN 27001.

DECK 'HCV   ' ASSIGNED ABSOLUTE ORIGIN 27002.  ADJUSTED LENGTH IS 00117.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS    ', LOCATION   17430.
     VIRTUAL SECTION 'CVCEL.' - REFERS TO DECK 'RWD    ', LOCATION   20165.
     VIRTUAL SECTION 'BU1.  ' - REFERS TO DECK 'RWD    ', LOCATION   20037.
     VIRTUAL SECTION 'BU22. ' - REFERS TO DECK 'RWD    ', LOCATION   20053.
     VIRTUAL SECTION 'SC.   ' - REFERS TO DECK 'RWD    ', LOCATION   20021.
     VIRTUAL SECTION 'GETCH.' - REFERS TO DECK 'RWD    ', LOCATION   20151.
     REAL SECTION    'IOHHC.' - ASSIGNED ABSOLUTE ORIGIN 27002.

DECK 'ICV   ' ASSIGNED ABSOLUTE ORIGIN 27121.  ADJUSTED LENGTH IS 00020.
     VIRTUAL SECTION 'CCQ.  ' - REFERS TO DECK 'RWD    ', LOCATION   20031.
     VIRTUAL SECTION 'SC.   ' - REFERS TO DECK 'RWD    ', LOCATION   20021.
     VIRTUAL SECTION 'AST.  ' - REFERS TO DECK 'INTJ   ', LOCATION   27370.
     VIRTUAL SECTION 'IC2.  ' - REFERS TO DECK 'INTJ   ', LOCATION   27341.
     VIRTUAL SECTION 'IC10. ' - REFERS TO DECK 'INTJ   ', LOCATION   27355.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS    ', LOCATION   17430.
     VIRTUAL SECTION 'IOHCM.' - REFERS TO DECK 'RWD    ', LOCATION   17537.
     VIRTUAL SECTION 'IOHCT.' - REFERS TO DECK 'RWD    ', LOCATION   17615.
     VIRTUAL SECTION 'IOHDB.' - REFERS TO DECK 'INTJ   ', LOCATION   27410.
     REAL SECTION    'IOHIC.' - ASSIGNED ABSOLUTE ORIGIN 27121.
     REAL SECTION    'IOHIT.' - ASSIGNED ABSOLUTE ORIGIN 27125.

DECK 'OCV   ' ASSIGNED ABSOLUTE ORIGIN 27141.  ADJUSTED LENGTH IS 00057.
     VIRTUAL SECTION 'BU1.  ' - REFERS TO DECK 'RWD    ', LOCATION   20037.
     VIRTUAL SECTION 'CCA.  ' - REFERS TO DECK 'RWD    ', LOCATION   20027.
     VIRTUAL SECTION 'GETCH.' - REFERS TO DECK 'RWD    ', LOCATION   20151.
     VIRTUAL SECTION 'HCT.  ' - REFERS TO DECK 'RWD    ', LOCATION   17614.
     VIRTUAL SECTION 'SC.   ' - REFERS TO DECK 'RWD    ', LOCATION   20021.
     VIRTUAL SECTION 'XC.   ' - REFERS TO DECK 'RWD    ', LOCATION   20110.
     VIRTUAL SECTION 'CVCEL.' - REFERS TO DECK 'RWD    ', LOCATION   20165.
     VIRTUAL SECTION 'FEXEM.' - REFERS TO DECK 'XEM    ', LOCATION   20473.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS    ', LOCATION   17430.
     VIRTUAL SECTION 'IOHCM.' - REFERS TO DECK 'RWD    ', LOCATION   17537.
     REAL SECTION    'IOHOC.' - ASSIGNED ABSOLUTE ORIGIN 27141.

DECK 'INTJ  ' ASSIGNED ABSOLUTE ORIGIN 27220.  ADJUSTED LENGTH IS 00321.
     VIRTUAL SECTION 'CVCEL.' - REFERS TO DECK 'RWD    ', LOCATION   20165.
     VIRTUAL SECTION 'BU1.  ' - REFERS TO DECK 'RWD    ', LOCATION   20037.
     VIRTUAL SECTION 'BU4.  ' - REFERS TO DECK 'RWD    ', LOCATION   20044.
     VIRTUAL SECTION 'CCQ.  ' - REFERS TO DECK 'RWD    ', LOCATION   20031.
     VIRTUAL SECTION 'DE60. ' - REFERS TO DECK 'RWD    ', LOCATION   20077.
     VIRTUAL SECTION 'DE70. ' - REFERS TO DECK 'RWD    ', LOCATION   20100.
     VIRTUAL SECTION 'XC.   ' - REFERS TO DECK 'RWD    ', LOCATION   20110.
```

```
     VIRTUAL SECTION 'FEXFM.' - REFERS TO DECK 'XEM    ', LOCATION   20473.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS    ', LOCATION   17430.
     VIRTUAL SECTION 'GETCH.' - REFERS TO DECK 'RWD    ', LOCATION   20151.
     REAL SECTION    'J.    ' - ASSIGNED ABSOLUTE ORIGIN 27231.
     REAL SECTION    'IC2.  ' - ASSIGNED ABSOLUTE ORIGIN 27341.
     REAL SECTION    'IC10. ' - ASSIGNED ABSOLUTE ORIGIN 27355.
     REAL SECTION    'AST.  ' - ASSIGNED ABSOLUTE ORIGIN 27370.
     REAL SECTION    'AST1. ' - ASSIGNED ABSOLUTE ORIGIN 27371.
     REAL SECTION    'DE30. ' - ASSIGNED ABSOLUTE ORIGIN 27375.
     REAL SECTION    'IOHDB.' - ASSIGNED ABSOLUTE ORIGIN 27410.
     REAL SECTION    'DBC1. ' - ASSIGNED ABSOLUTE ORIGIN 27422.
     REAL SECTION    'DBC5. ' - ASSIGNED ABSOLUTE ORIGIN 27442.
     REAL SECTION    'ANACH.' - ASSIGNED ABSOLUTE ORIGIN 27471.
     REAL SECTION    'SG1.  ' - ASSIGNED ABSOLUTE ORIGIN 27510.
     REAL SECTION    'DECEL.' - ASSIGNED ABSOLUTE ORIGIN 27532.

DECK 'FFC    ' ASSIGNED ABSOLUTE ORIGIN 27541.  ADJUSTED LENGTH IS C0366.
     VIRTUAL SECTION 'IC10. ' - REFERS TO DECK 'INTJ   ', LOCATION   27355.
     VIRTUAL SECTION 'BU22. ' - REFERS TO DECK 'RWD    ', LOCATION   20053.
     VIRTUAL SECTION 'DE60. ' - REFERS TO DECK 'RWD    ', LOCATION   20077.
     VIRTUAL SECTION 'DE70. ' - REFERS TO DECK 'RWD    ', LOCATION   20100.
     VIRTUAL SECTION 'IC2.  ' - REFERS TO DECK 'INTJ   ', LOCATION   27341.
     VIRTUAL SECTION 'J.    ' - REFERS TO DECK 'INTJ   ', LOCATION   27231.
     VIRTUAL SECTION 'DECEL.' - REFERS TO DECK 'INTJ   ', LOCATION   27532.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS    ', LOCATION   17430.
     VIRTUAL SECTION 'IOHCT.' - REFERS TO DECK 'RWD    ', LOCATION   17615.
     VIRTUAL SECTION 'IOHDB.' - REFERS TO DECK 'INTJ   ', LOCATION   27410.
     REAL SECTION    'FFC10.' - ASSIGNED ABSOLUTE ORIGIN 27541.
     REAL SECTION    'FFC11.' - ASSIGNED ABSOLUTE ORIGIN 27545.
     REAL SECTION    'FFC12.' - ASSIGNED ABSOLUTE ORIGIN 27550.
     REAL SECTION    'FFC23.' - ASSIGNED ABSOLUTE ORIGIN 27570.
     REAL SECTION    'FFC24.' - ASSIGNED ABSOLUTE ORIGIN 27572.
     REAL SECTION    'FFC30.' - ASSIGNED ABSOLUTE ORIGIN 27712.
     REAL SECTION    'DE41. ' - ASSIGNED ABSOLUTE ORIGIN 27715.
     REAL SECTION    'DE44. ' - ASSIGNED ABSOLUTE ORIGIN 27724.
     REAL SECTION    'FLTSW.' - ASSIGNED ABSOLUTE ORIGIN 27736.
     REAL SECTION    'BDE.  ' - ASSIGNED ABSOLUTE ORIGIN 27763.
     REAL SECTION    'CMC.  ' - ASSIGNED ABSOLUTE ORIGIN 30032.
     REAL SECTION    'IOHPW.' - ASSIGNED ABSOLUTE ORIGIN 30045.

DECK 'SLO    ' ASSIGNED ABSOLUTE ORIGIN 30127.  ADJUSTED LENGTH IS C0025.
     VIRTUAL SECTION 'ERLOC.' - REFERS TO DECK 'XEM    ', LOCATION   20472.
     VIRTUAL SECTION 'IOCEL.' - REFERS TO DECK 'IOS    ', LOCATION   17430.
     REAL SECTION    'SLOIC.' - ASSIGNED ABSOLUTE ORIGIN 30127.
```

```
SPACE LIBRARY - VER. 2, MOD. 1                 IBLDR -- JOB   SPACE

                        M E M O R Y   M A P

                  OF LINK NUMBER  4,( UTLITY )
              ORIGIN OF THIS LINK AT  DECK 'UTL    '


SYSTEM, INCLUDING IOCS                              000C0 THRU  12252

     NUMBER OF FILES -  NONE

OBJECT PROGRAM                                      12352 THRU  30153

     1.  DECK 'SPC    '          12352
     2.  DECK 'DSSCAN'           17001
     3.  SUBR 'INSYFB'           00000
     4.  SUBR 'OUSYFB'           00000
     5.  SUBR 'POSTX '           17040
     6.  SUBR 'PPSYFB'           00000
     7.  SUBR 'F05    '          17152
     8.  SUBR 'F06    '          17153
     9.  SUBR 'IOS    '          17154
    10.  SUBR 'RWD    '          17452
    11.  SUBR 'ACV    '          20203
    12.  SUBR 'XCV    '          20233
    13.  SUBR 'FPT    '          20251
    14.  SUBR 'XEM    '  *       20467
    15.  SUBR 'XIT    '          20700
    16.  SUBR 'DMP    '          20702
    17.  SUBR 'RDH44 '           22572
    18.  DECK 'UTL    '  *       24232
    19.  SUBR 'ECV    '          26601
    20.  SUBR 'HCV    '          27002
    21.  SUBR 'ICV    '          27121
    22.  SUBR 'OCV    '          27141
    23.  SUBR 'INTJ   '          27220
    24.  SUBR 'FFC    '  *       27541
    25.  SUBR 'SLC    '          30127

    (* - INSERTIONS OR DELETIONS MADE IN THIS DECK)

INPUT - OUTPUT BUFFERS                              76157 THRU  77776

UNUSED CORE                                         30154 THRU  76147

 OBJECT PROGRAM IS BEING ENTERED INTO STORAGE.

SPACE - VER. 2, MOD. 1 HAS CONTROL
                   REELS R 91  B  DEFER    F 91001 LIST F 91002 LIST F 91003 LIST
                         R -3 F -3001 LIST F -3002 LIST
                          REEL -4    C    FILE -4001 LIST
                         R  -5  F -5001 COPYTO 91004 LIST F -5002 COPYTO 91005 LIST
                         F 91001 COPYTO -5003   F -5001 COPYTO 91006   F 91004 LISTI 4
                         •
```

2.3 RELATED PROGRAMMER INFORMATION

a. The monitor itself occupies approximately 2325/10 locations of core storage; however, it overlays more than the first quarter of itself with the following:

   1. Post-execution file utility operations table.
   2. IOCS file control blocks.
   3. IOCS buffer pool list.
   4. Buffers.

   Depending upon the storage required by items 1-3, one or two buffers will occupy this area. Remaining buffers, if needed, will be constructed in upper core. Note that the number of such buffers which can be used for a given program is a function of its size; i.e., if the program is very large, some buffers may have to be released to serve as program area. The pool will, however, be restored to its original state for the next program, unless a similar situation exists. In every case, the upper memory break contained in the address portion of S.SLOC+3 is adjusted by the monitor to reflect buffer residence. This adjustment is made for each program when it is loaded.

b. The READH subroutine is used by the monitor for BCD input and hence, must be included in the main link.

c. Although the library must be prepared in chain link format, the chain subroutine in IBLIB is not used for loading links. Unless referenced by some program, it will never be in core.

d. Two stringent requirements are imposed upon the main link. First, it must contain the definitions of all IOCS files used in the entire library, and second, it must contain the definition of the largest block of blank common. Thus, if any programs require the use of non-SPACE files, the appropriate $FILE cards must be included in the main link when the library is created. The DEFER or READY mounting option should be specified for all such files. Use of blank common can usually be avoided by employing named common; however, the user may elect to include a deck in the main link specifying the largest definition. For example:

```
1       8       16
$IBMAP  DEFINE
        CONTRL  //
        USE     //
        BSS     500
        END
```

e. Variable unit designations should never be used in FORTRAN I/O statements, as all logical units defined in the IOU table will be considered unavailable to the monitor.

SECTION 3

USER'S MANUAL

TABLE OF CONTENTS

Paragraph		Page
3.1	PREPARING THE REELS CARD SERIES	3-1-01
3.1.1	CATEGORY KEYWORDS	3-1-01
3.1.2	OPTIONS PERTAINING TO INDIVIDUAL REELS	3-1-02
3.1.3	ORDER OF UNIT ASSIGNMENT	3-1-04
3.1.4	POST-EXECUTION FILE UTILITY OPTIONS	3-1-05
3.1.5	SERIES PREPARATION EXAMPLES	3-1-06
3.1.6	ASSOCIATED ERROR DIAGNOSTICS	3-1-07
3.2	CREATING THE LIBRARY	3-2-01
3.2.1	PLANNING THE LINK STRUCTURE	3-2-01
3.2.2	LIBRARY PREPARATION EXAMPLES	3-2-02
3.3	THE TYPEWRITER LISTING	3-3-01
3.4	RELATED USER INFORMATION	3-4-01

SECTION 3

USER'S MANUAL

3.1 PREPARING THE REELS CARD SERIES

Every reel, permanent or mediary, to be used in a given job must be specified in the REELS card series. In addition, all files which are to take part in post-execution utility operations must be identified in the series. The user should familiarize himself with the organization of the job deck as prescribed in section 2.1.2 of this manual.

3.1.1 CATEGORY KEYWORDS

Information appearing in the REELS card series must be in the form of groups, of which there are three possible categories. The beginning of each group is identified by a keyword classifying the group and ends when either the keyword of the next group or the end of the series is encountered. The category keywords are as follows:

a. REEL - This keyword indicates the beginning of a group of information related to some reel which is to be used for the job. It must always be followed immediately with a positive permanent, or negative mediary reel number. Options pertaining to the reel may follow the reel number in any order.

b. FILE - This keyword indicates the beginning of a group of information related to some file which is to be processed by the post-execution utility program. The keyword must always be followed immediately with the file number of the file which is to be referenced. Either, or both, of the two utility options may follow the reference file number in any order.

c. BUFFERS - If it is used, this keyword must be immediately followed with a single integer which establishes the number of buffers to be assigned for the job. If this keyword is omitted, or if the specification is zero, two buffers will be assigned for each reel (unless two or more reels share the same physical device by virtue of the DEFER option).

To permit economy in punching, the keywords REEL, FILE, and BUFFERS may be punched as R, F, and BUF, respectively.

3.1.2 OPTIONS PERTAINING TO INDIVIDUAL REELS

In some instances, it is necessary to supply additional information related to a certain reel, or the user may elect to exercise one or more options available for the reel. This information must be part of the R group to which it applies and can follow the reel number in any order. The various options/specifications recognized include:

a. The unit to which the reel is to be assigned, as follows:

Uxx = The reel is to be assigned to the symbolic utility unit, S.SUxx.

Iyy = The reel is to be assigned to a unit having the intersystem reservation code yy.

IyyR = Same as Iyy, except that after the assignment is made, the unit's reservation code is to be canceled (set to 00).

Uxx Iyy = Same as Uxx, except that the reservation code yy ($\leq$ 20) is to be assigned to the symbolic unit specified by Uxx. Note that these are two separate words.

B = The reel should be assigned to any tape drive on channel B.

C = The reel should be assigned to any tape drive on channel C.

b. The specification, NOLABEL, which indicates that a reel header label does not exist, and hence must be created. Unless NOLABEL has been specified for a given permanent reel, the monitor will attempt to read and verify its label to insure that the correct reel has been mounted. If the label cannot be read, or cannot be identified, the condition will be treated as an error and will terminate execution. The NOLABEL specification is not required for mediary reels and should only be employed with the first SPACE application of a permanent reel.

c. The FP option may be exercised for permanent reels which are to be logically file protected by the monitor. With this option in effect, the associated reel cannot be written upon. An exception occurs when a utility operation requests a file to be copied onto the reel. In this case, and this case only, the file protect option is ignored.

d. The DEFER option may be used for a reel which will not be needed until later in the job, or perhaps will not be needed at all. When this option is employed, the operator mounting message is deferred until a file on the reel is opened by READB1 or ABOUT1. Observe that this supplies the user with the ability to assign two or more reels to the same physical unit. With such an application, the monitor will unload the reel currently mounted and request the deferred reel to be mounted in its place. The removed reel will then be placed in a deferred status, even though the DEFER option was not exercised with it. The user must, therefore, be aware of the order in which files will be used on reels sharing the same unit. Consider, for example, reels A and B which have both been assigned to the same unit by virtue of the DEFER option for reel B. If files on these reels starting with A, were referenced in an alternate fashion, the operator would be in continuous state of repeating the following four steps:

1. Remove A
2. Mount B
3. Remove B
4. Mount A

3.1.3 ORDER OF UNIT ASSIGNMENT

The monitor will consider a device available and hence, permit assignment only if the following requirements are met:

a. The device is symbolically attached.

b. Its reservation code is 00, except for Iyy or IyyR specifications.

c. It is either a tape transport on channel B or C, or disk/drum storage accessed sequentially in full track with addresses mode.

When more than one technique of unit assignment has been specified for reels appearing in the REELS card series, the order of assignment is as follows:

1. All Uxx specifications are processed first.

2. All Uxx Iyy, Iyy, and IyyR specifications are processed second.

3. All reels having a channel preference are assigned next. If an available unit cannot be found on the desired channel, the channel preference will be ignored and the reel placed in the next category.

4. All reels which do not have an assignment specification are assigned units last. When making such assignments, the monitor will attempt to satisfy as many of the following criteria as possible:

   a. Disk/drum utility units for mediary reels.
   b. Tape transports in ready status for all mediary reels which cannot be assigned disk/drum.
   c. Tape transports not in ready status for all permanent reels.

3.1.4 POST-EXECUTION FILE UTILITY OPTIONS

The user may elect to employ either or both of the two utility options for a given file. These options must be part of the F group to which they apply, and can follow the reference file number in any order.

a. The LISTx option is used to dump selected logical records from the reference file onto S.SOU1 in the format prescribed by the character x appended to the word. The significance of this character is as follows:

x	=	blank or 8	- List in octal.
x	=	I	- List as integer.
x	=	F	- List as floating point decimal in the format $\pm$X.XXXXXXE$\pm$YY.
x	=	B	- List as BCD.

The LISTx option may assume any one of the following forms:

1.	LISTx	- All records in the reference file are to be listed.
2.	LISTx i	- The first i records of the reference file are to be listed.
3.	LISTx i,j	- Records starting with record i up to and including record j are to be listed. The integers i and j may be given in any order.

Note that option (1) is a single word, while options (2) and (3) constitute two words.

b. The COPYTO option provides a means by which the reference file can be copied to another file. The word, COPYTO, must always be followed by the file number of the file to which the copy is desired. With the exception of the file number, the contents of the identification record of the new file will remain the same as that of the reference file. The user will always be notified of the success or failure of a requested copy.

NOTE: Utility operations are performed in the order which they are given in the REELS card series. Hence, the user must plan the sequence of activities to be performed.

3.1.5 SERIES PREPARATION EXAMPLES

Example 1:

Col. 1-5	Col. 7-72
REELS	R 91 NOLABEL F 91001 LIST F 91002 LIST
	R -1 R -2 B FILE -1004 COPYTO 91003
	LISTF 20 *

In the example above, a total of three reels have been specified. Reel 91 is permanent and does not have a label. Two files on this reel, 91001 and 91002 should be fully listed in octal at the end of the job. Two mediary reels having reference numbers -1 and -2 are also required in the job. If possible, reel -2 should be assigned to a tape unit on channel B. At the end of the job, file -1004 is to be copied to file 91003 and the first twenty logical records of file -1004 are to be listed in floating point decimal.

Example 2:

Col. 1-5	
REELS	R -5 R -10 R -15 R 478 DEFER
	NOLABEL U04 R 1396 U04 R 673 I16R
	F -10002 LIST8 50,100 COPYTO 478001 *

This example illustrates the use of six reels. Three of these are mediary and have reference numbers -5, -10 and -15. The remaining three reels are permanent and bear identification numbers 478, 1396 and 673. Assuming the configuration of the system consists of only five symbolic utility units, reels 478 and 1396 have been assigned to the same device (S.SU04) by virtue of the DEFER option for reel 478. This reel is not needed during processing; however, a file which will be created during the job must be saved on the reel. Hence, when the utility processor opens file 478001 for the requested copy, reel 1396 will be unloaded and reel 478 mounted in its place. The appearance of the NOLABEL specification for reel 478 indicates that it must be labeled when it is mounted. Reel 673 is to be assigned to a symbolic unit having the intersystem reservation code 16 and, as soon as the assignment is made, the reservation code is to be cancelled. At the end of the job, logical records 50-100 of file -10002 are to be listed in octal and the file is to be copied to file 478001.

3.1.6 ASSOCIATED ERROR DIAGNOSTICS

The following errors can occur while processing the REELS card series. Each will have the effect of terminating execution without calling the DUMP program.

3.1.6.1 'REELS' CARD MISSING, OR IMPROPERLY PUNCHED.

The monitor has not found the BCD word REELS punched in columns 1-5 of the first card in the series.

3.1.6.2 UNRECOGNIZED OPTION ON 'REELS' CARD -- WORD ** = ******

A word has been punched in the series which the monitor cannot recognize.

3.1.6.3 FILE INFO ON 'REELS' CARD EXCEEDS CAPACITY OF ALLOTTED STORAGE BY *** ENTRIES.

The table into which the post-execution file utility operation requests are placed has overflowed. This table is initially constructed in the area occupied by the first dependent link, and later moved to overlay code which is no longer needed at the origin of the monitor. This overlay area constitutes approximately $230_{/10}$ cells and can accommodate many utility requests. Within the table, three entries are required for each F group specifying both utility operations, and two entries are required for each F group specifying only one operation. The number given in the diagnostic reflects the amount of excess in terms of cells.

3.1.6.4 TOO MANY REELS SPECIFIED - LIMIT is 15.

More than fifteen R groups have been specified in the series. This limit is nominal and may be changed according to user requirements.

3.1.6.5 SYMBOLIC UNIT S.SU** REQUESTED FOR REEL ***** IS UNAVAILABLE.

A Uxx specification has been given for the reel; however, the monitor has discovered that the requested symbolic unit is not available. The fact that it is unavailable may be attributed to either of the following conditions:

a. The unit is not symbolically attached.
b. Its reservation code is not 00.

3.1.6.6 RESERVATION CODE ** (REEL *****) IS NONEXISTENT.

An Iyy or IyyR specification has been given for the reel; however, a unit with a matching reservation code was not found.

3.1.6.7 UNIT ASSIGNMENT CANNOT BE MADE FOR REEL *****

When the monitor has allegedly finished processing the REELS card series, a verification of all unit assignments is made. This diagnostic indicates that the given reel could not be assigned to a unit, or the assignment would otherwise be illegal. The condition was probably caused by one of the following:

a. Sufficient utility devices were not available to accommodate all of the reels.
b. A utility device other than magnetic tape on channel B or C was specified (or implied) for the given permanent reel.
c. A utility device other than magnetic tape on channel B or C, or disk/drum storage which is not accessed sequentially in full track w/addresses was specified (or implied) for the given mediary reel.

3.1.6.8 REELS ***** AND ***** HAVE DUPLICATE UNIT ASSIGNMENT SPECIFICATIONS.

The two reels were assigned by the user to the same physical device; however, the DEFER option was not specified for either one.

3.2 CREATING THE LIBRARY

The SPACE library must always consist of an overlay structure created by employing the chain feature of IBJOB. The configuration must be such that the monitor is contained in the main link, with subsidiary programs constituting the dependent links. By exercising the copy feature, the library can be placed on the user's personal reel of tape. Subsequent SPACE applications will then require the library reel to be mounted, and the reload feature to be employed. The user should read the sections entitled Loader Chain Feature and the Reload Program, both contained in the 7040/44 Programmer's Guide.

3.2.1 PLANNING THE LINK STRUCTURE

A considerable amount of tape spacing time can be saved by planning the link structure of the library in an efficient manner, as prescribed by the following:

a. A program which is executed via a direct chain from another program should immediately follow the latter in the library.

b. Subroutines common to two or more programs should be included in the main link along with the monitor. This otherwise prevents the same routine(s) from being loaded with each program in the course of execution.

c. The post-execution file utility processor should always be the last link in the library. Otherwise, it might have to be passed over several times in the course of accessing the other programs, thus causing an unnecessary waste of tape passing time.

In general, a certain amount of finesse is needed to create an efficient library. The user should be aware of the frequency and order in which programs will normally be executed, and try to plan the library so that a minimum of tape passing time is needed to access the programs.

3.2.2 LIBRARY PREPARATION EXAMPLES

In the examples which follow, the library will be created on reel 527 and will contain programs 71Z, 81Z and 91Z, in addition to the required utility processor, UTLITY. Normally, program 71Z will be executed via a direct chain from 91Z, after which 81Z will be executed. The structure of the library, therefore, should be as follows:

```
Main link            =  SPACE
1st dependent link   =  91Z
2nd dependent link   =  71Z
3rd dependent link   =  81Z
4th dependent link   =  UTLITY
```

Assume that the decks comprising the main link and the four dependent links above exist in binary, FORTRAN source, binary, MAP symbolic and binary, respectively.

Example 1: Create the library, but do not execute.

CARD NUMBER \ CARD COLUMN	1	8	16
1.	$JOB		CREATE LIBE - NO EXECUTION
2.	$PAUSE		READY REEL 527 ON S.SU04
3.	$IBJOB	LIBE	NOGO,LOGIC,COPY=U04
4.	$CHAIN	SPACE	
5.	$IBLDR	SPC	
6.	(SPACE binary deck)		
7.	$DKEND	SPC	
8.	$ENTRY		
9.	$LINK	91Z	
10.	$IBFTC	91Z	LIST,REF
11.	(91Z FORTRAN source deck terminating with END card)		
12.	$ENTRY		
13.	$LINK	71Z	
14.	$IBLDR	71Z	
15.	(71Z binary deck)		
16.	$DKEND	71Z	
17.	$ENTRY		
18.	$LINK	81Z	
19.	$IBMAP	81Z	
20.	(81Z MAP symbolic deck terminating with END card)		
21.	$ENTRY		
22.	$LINK	UTLITY	
23.	$IBLDR	UTLITY	
24.	(UTLITY binary deck)		
25.	$DKEND	UTLITY	
26.	$ENTRY		
27.	$ENDCH		
28.	$IBSYS		

Card functions for example 1:

1. Define job.

2. Pause, after directing operator to mount reel 527 on the physical unit (tape transport) assigned S.SU04.

3. Define processor application, name of such, and options which are to be applied.

4. Must immediately follow the $IBJOB card and specifies the name of the main link. For purposes of convention and identification, this name should always be SPACE.

5. Specifies loader application and name of deck.

6. Constitutes the binary deck of SPACE.

7. Signifies end of deck SPC. Should the user elect to place any subroutine decks in the main link, they should be inserted between cards 7 and 8.

8. Signifies end of main link.

9. Defines beginning of first dependent link. The link name beginning in column 8 is that which must be used in columns 1-6 in the program control card series, or the CALL argument in the case of a direct chain. Note that it may be the same as the deck name appearing on any processor control card.

10. Specifies a FORTRAN compilation for deck 91Z. Note that the LIST and REF options have been exercised to obtain a MAP listing and cross reference index. This is necessary, should the programmer or user desire to make an execution-time binary patch in the program.

11. Constitutes the entire FORTRAN source deck for program 91Z. If 91Z requires subroutine decks, these decks should immediately follow. They will become part of the first dependent link and will be loaded along with 91Z at execution time.

12. Signifies end of first dependent link.

13. Defines beginning of second dependent link.

14-16. Entire binary deck of program 71Z.

17. Signifies end of second dependent link.

27. Signifies end of all links. Since a symbolic unit specification (U04) was given for the COPY option, reel 527 will be unloaded.

28. Return control to IBSYS.

Example 2: An application using the library.

CARD NUMBER / CARD COLUMN	1	8	16
1.	$JOB		SPACE APPLICATION
2.	$PAUSE		READY REEL 527 ON S.SU04
3.	$IBJOB	RELOAD	NOSOURCE
4.	$RELOAD		U04,NAME=SPACE,SRCH
5.	REELS	R -1 R 91 F -1002 COPYTO 91003	
6.		LISTB 100 *	
7.	91Z	INPT 91001 OTPT -1001 *	
8.	(Data cards to be read by program 91Z)		
9.	DUMP	*	
10.	81Z	-1001 *	
11.	DUMP	10428	10498 2 *
12.	$IBSYS		
13.	$CLOSE		S.SU04,REMOVE
14.	$IBSYS		

Card functions for example 2:

1. Define job.
2. Pause for operator to mount library tape.
3. Define processor application. The NOSOURCE option should always be specified to expedite pre-processing.
4. Request reload program.

5-6. Constitutes the REELS card series.

7. Program control card requesting execution of program 91Z. Note that four control parameters are being supplied to the program.

8. Data cards which will be read by 91Z. The program must have some means of recognizing the last card it is to read.

9. Program control card requesting execution of the dump program with standard parameters. Note that program 71Z was the last program executed up to this point, due to the direct chain from 91Z.

10. Program control card requesting execution of program 81Z. One control parameter must be supplied to the program.

11. Program control card requesting execution of the dump program. The area of core storage from decimal locations 10428 through 10498 will be dumped in integer format.

12. Signifies end of job deck and indicates that SPACE is to return to IBJOB, after which control will be given to IBSYS.

13. Unloads reel 527 to prevent destruction by subsequent jobs.

14. Card is redundant; however, must be included at the end of every job.

Example 3: Create the library and test it:

CARD NUMBER / CARD COLUMN	1	8	16
1.	$JOB		CREATE LIBE AND TEST
2.	$PAUSE		READY REEL 527 ON S.SU04
3.	$IBJOB	LIBE	LOGIC,COPY=U04=I04
(Insert cards 4-28, inclusive, of example 1)			
4.	$IBJOB	TST527	NOSOURCE
5.	$RELOAD		I04,NAME=SPACE,SRCH
(Insert cards 5-12, inclusive, of example 2)			
6.	$CLOSE		I04R,REMOVE
7.	$IBSYS		

Card functions for example 3:

1. Define job.
2. Pause for operator to mount reel 527.
3. Define processor application. Note that a reservation code is assigned to the unit to prevent the reel from being unloaded at the end of the first application.
4. Define next processor application.
5. Request reload program.
6. Cancel reservation code and remove reel 527.
7. Required at end of job.

3.3 THE TYPEWRITER LISTING

The purpose of the typewriter listing is to provide the operator with a list of mounting instructions, and the user with a record of all programs executed. Use of the typewriter is a relatively expensive operation, and an effort has been made to minimize its useage.

The monitor always identifies itself at the outset of an application by typing the following message:

SPACE - VER. x, MOD. y HAS CONTROL

Appropriate version and mod levels will be supplied to the message. Following this, a list of mounting instructions, if applicable, will be issued to the operator, followed by a halt. For example:

READY REEL 91 ON B4.
READY REEL 1395 ON C3.
OPER. ACTION PAUSE

The remainder of the typewriter listing will consist of the order and name of each program as it is executed, interspersed with error or unusual condition diagnostics, if any. The last two examples given in section 3.2.2 might have produced the following:

1. 91Z
2. 71Z
3. DUMP
4. 81Z
5. DUMP
6. UTLITY

3.4 RELATED USER INFORMATION

a. The monitor will not issue mounting instructions for a reel which is assigned to a unit via an Iyy(R) or Uxx Iyy option, unless the given unit is not in ready status.

b. Permanent reels will always be unloaded at the end of the job, except those which have been assigned to units via the Iyy or Uxx Iyy option. All other reels will be rewound.

c. The reel header label created or verified by the monitor for every reel used in a job consists of a five word physical record, as follows:

   Word 1 - SPACE (BCD)
   Word 2 - Reel number (integer)
   Word 3 - Creation date (BCD, MMDDYY)
   Word 4 - Job name on $CHAIN card (BCD)
   Word 5 - A checksum of words 1-4.

   Only words 1, 2 and 5 are verified when a label is read.

d. All labels and data files are in high density binary mode.

e. If an interval timer is available (and operative), the monitor will indicate the total number of both green and red light seconds which have expired in the job before returning to IBJOB.

f. The REELS card series is always written on S.SOU1 in card image form.

g. The user should always specify a SPACE application on the machine request card.

h. To permit adaptability according to user requirements, the following specifications exist as assembly parameters and may readily be changed, should the need arise. The nominal assignment in the distributed version appears to the right of the associated specification:

1.	Buffer size	257
2.	Length of HC block	100
3.	Length of reel header labels	5
4.	Length of file labels	12
5.	Max. number of reels permitted	15
6.	Single or double buffering	Double
7.	Block sequence checking	Yes
8.	Block check sum verification	Yes

SECTION 4

RELATED SUBROUTINES

TABLE OF CONTENTS

Paragraph		Page
4.1	GENERALIZED SORTING	4-1-01
4.1.1	THE INTERNAL SORT ROUTINE	4-1-01
4.1.2	FILE SORTING WITH GSMRGE	4-1-04
4.2	BCD OUTPUT VIA GGOUT	4-2-01
4.2.1	CONVERSION SPECIFICATIONS	4-2-01
4.2.2	PSEUDO-OPERATION DESCRIPTIONS	4-2-03
4.3	UTILITY ROUTINES	4-3-01
4.3.1	FILE NUMBER ACQUISITION WITH GETF	4-3-01
4.3.2	OBTAINING THE CURRENT DATE	4-3-01

SECTION 4

RELATED SUBROUTINES

4.1 GENERALIZED SORTING

Two flexible subroutines are available for the SPACE programmer enabling him to sort either an internal block of logical records, or a given data file. The sorting process, in both cases, consists of arranging the logical records in an ascending ordered sequence, with respect to a certain word (or words) within each of the records. These words constitute the 'sort keys', and the order in which they are specified establishes how the records are to be sorted. The first sort key given is termed the major key. Sort keys following the major key, if any, are termed the first minor key, second minor key, etc. Whenever two records are compared during the sorting process, the major key in one record is compared with the major key in the other record to determine which record is the smallest. Should the major keys compare equal, the first minor keys, if specified, are compared. This process continues through the minor keys until either an unequal compare occurs, or no more keys remain for comparison.

Each of the routines has the ability of comparing individual keys via an arithmetic compare (CLA-CAS), or a logical compare (CAL-LAS). No provision has been made for specifying collating sequences, nor preserving the sequence of presorted strings. If sequence preservation is necessary, the programmer can append a record sequence word to each logical record and specify this word as the last minor sort key.

4.1.1 THE INTERNAL SORT ROUTINE

The internal sort routine may be employed to sort a given block of contiguous logical records. Linkage must be as follows:

CALL SORT(FWA,NREC,LRS,KEY1,KEY2···,KEYN)

where,

FWA = The first word address of the block of records to be sorted.

NREC = The address of a location containing the number of records in the block.

LRS = The address of a location containing the logical record size; i.e., the length, in words, of each record.

KEY1 = The address of a location containing the relative position of the major sort key; e.g., if the sixth word of each record is to be the major key, then C(KEY1) = 6.

KEY2 = The address of a location containing the relative position of the first minor sort key.

KEYN = The address of a location containing the relative position of the $(N-1)^{th}$ minor sort key.

The following restrictions/conventions apply to the use of the internal sort routine:

a. Parameters NREC, LRS and KEY1 through KEYN must be expressed as right-adjusted integers.

b. The sign of each individual key specified establishes the type of compare to be used for the key. If the sign is plus, a CLA-CAS instruction combination will be used; if minus, a CAL-LAS combination will be used. Any number of keys may be specified.

c. No validity checks are made on the arguments given to the routine.

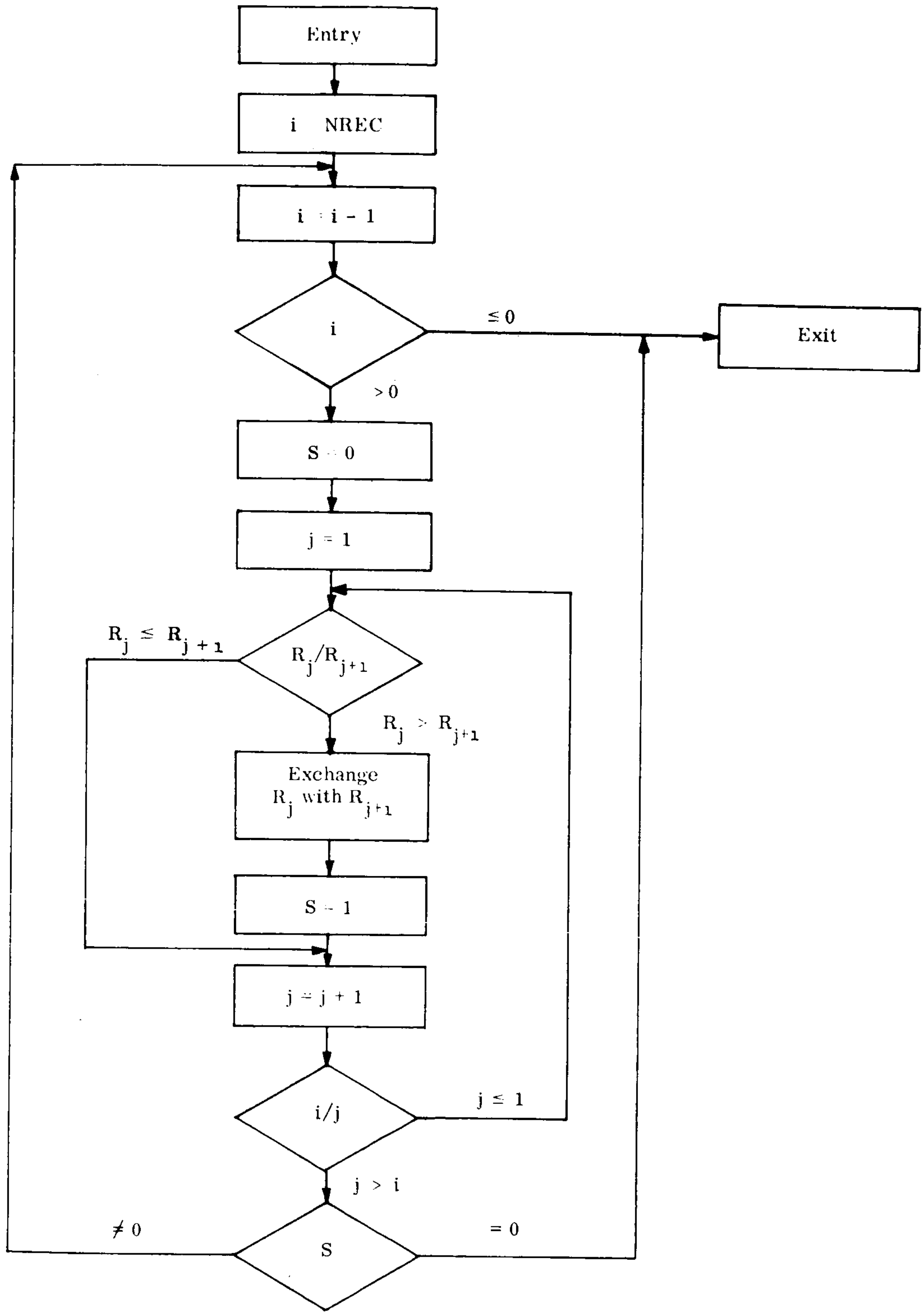
d. The most time consuming sort application will occur when all records are presorted in reverse order with respect only to the last minor key. For such an application, the exact number of machine instructions (I) required to sort the block is given by the following equation:

$$I = \tfrac{1}{2}(NREC-1)\ \{NREC[6(LRS) + 14(NKEYS) + 3] + 20\} + 38$$

This equation also holds for the special cases in which NREC = 0, or NREC = 1.

e. The entire routine occupies 73 decimal locations.

f. The sorting technique employed is depicted by the following flow chart. Here, R_j represents the j^{th} record of the block.


Entry
i NREC
i = i - 1
i
≤ 0
Exit
> 0
S = 0
j = 1
R_j ≤ R_j+1
R_j/R_j+1
R_j > R_j+1
Exchange R_j with R_j+1
S = 1
j = j + 1
i/j
j ≤ 1
j > i
≠ 0
S
= 0

1.2 FILE SORTING WITH GSMRGE

With many sorting applications, the logical records constitute a data file on an external recording medium. By employing the following subroutine, the data file can be sorted, thus creating a new file reflecting the logical records in ordered sequence. Linkage must be as follows:

CALL GSMRGE(IFILE,OFILE,KEY1,KEY2,•••,KEYN)

where, IFILE = The address of a location containing the file number of the file to be sorted (input file).

OFILE = The address of a location containing either the desired file number of the sorted file to be created by GSMRGE (output file), or zero if the user has no preference concerning the output file number. In the latter case, GSMRGE will store the output file number in location OFILE before returning to the caller.

The sort keys, KEY1 through KEYN, are defined and must be used in the same manner as with the internal sort routine.

The technique which is employed to sort a given file consists of four logical phases--each phase performing a specific function of the sorting operation. A brief description of the four phases, in order of execution, follows:

Phase 1: The Edit Phase.

The edit phase is primarily devoted towards analyzing and verifying call arguments, defining record storage areas for subsequent phases, and compiling a calling sequence to the internal sort routine for use by phase two. The input file is opened during this phase, and an available work file (WORK1) is selected to accommodate the first sorted string which will be developed in phase two. If OFILE has been specified, and, if it is not on the same reel as IFILE, it will be chosen as WORK1.

Phase 2: The Internal Sort Phase.

The function of the internal sort phase is to develop strings of sorted records read from the input file, and place these strings alternately on two output work files. The second such work file (WORK2) will not be located until the need for it becomes apparent; i.e., when the second string is developed. If only one string is generated during this phase, control passes to phase four.

Phase 3: The Merge Phase.

The function of the merge phase is to elongate the strings produced by phase two via a second order of merge until only one string exists, which constitutes the sorted file. To accomplish this, the record storage area defined during the edit phase is divided into two read buffers, and each of these buffers are filled with records read from files WORK1 and WORK2, respectively. An output work file (WORK3) is located to accommodate the first elongated string developed during the first merge pass. A fourth work file (WORK4) will be located when the need for it becomes apparent; i.e., if a second elongated string is developed. The merge process begins by comparing the first record in one buffer with the first record in the other buffer. The lowest of the two records is then written in file WORK3. From this point on, the process continues by comparing the next unused record of one buffer with the next unused record of the other buffer. The lowest of these two records is then compared with the last record which was written, and, if the former is higher than the latter, it is added to the file. Whenever one of the buffers is emptied, it is refilled from the appropriate input file. Eventually, stepdowns will occur in both buffers; i.e., a point will be reached where the last record added to the output file is higher than either of the records compared in the two buffers. At this time, the current output file is suspended, and the process resumes using the other output file. Thus, the elongated strings will be alternately written on the two output files, and, the length of any such string will be at least twice that of any string read from either of the input files. The end of the merge pass occurs when both input files have been emptied. If only one output file was used during the pass, control passes to phase four; otherwise, input and output files are switched and the next merge pass is started.

Phase 4: The Posting Phase

The posting phase is executed when a single string has been developed on an output file. If the user did not indicate a preference of output files, the file number of this file is placed in location OFILE, and control is returned to the calling program. If an output file preference was indicated, a file copy, if necessary, is performed to produce the sorted output on the specified file.

4.1.2.1 <u>RELATED INFORMATION</u>

a. The record storage area defined in the edit phase consists of all locations between the upper and lower memory breaks, these of which are contained in the address of S.SLOC+3 and decrement of S.SLOC+4, respectively.

b. The search for available working files always begins with the first entry in the REELS Table. A given reel will be judged available only if the following criteria are satisfied:

1. No file is currently active on the reel.
2. The reel is not logically file protected.
3. The reel is not currently in a deferred state.
4. The position of the trailer label is known. Note that the position of the trailer will be known if (a) the reel is mediary, or (b) any file has previously been written on the reel during the job, or (c) the NOLABEL option was exercised in the REELS card series.

When an available reel is found, the file number of the work file is computed by multiplying the reel number by 1000 and adding one plus the corresponding entry in the TRLPOS table. Thus, no data file can ever be inadvertently destroyed by GSMRGE.

c. All files used by the subroutine, including the final output file, are closed before returning to the calling program.

d. The entire subroutine occupies 597 storage locations. Virtual control sections include READB1, READB2, ABOUT1, ABOUT2, ENDF, OEDIT, SPACE, SORT, S.SLOC and S.XOVA.

4.1.2.2 ASSOCIATED ERROR DIAGNOSTICS

If an error is detected by the subroutine, an appropriate diagnostic will be written on S.SOU1, followed with a direct chain to the dump program. The diagnostic will always include the BCD name of the program currently in execution, the absolute octal location of the call to GSMRGE, and its associated internal formula number (IFN), if one exists. The error comments, together with the condition(s) which can cause the error are as follows:

1. NO SORT KEYS.

   At least one sort key must be supplied to the routine.

2. BAD SORT KEY SPECIFIED.

   The magnitude of a given sort key either exceeds the LRS of the file, or is zero.

3. **INSUFFICIENT INTERNAL SORT AREA.**

The record storage area defined during the edit phase is not large enough to accommodate at least three logical records.

4. **INSUFFICIENT MERGE TAPES.**

An available work file cannot be located. Note that this condition could occur in any of the first three phases.

4.2 BCD OUTPUT VIA GGOUT

This is a highly flexible, interpretive output subroutine, controlled by the programmer via pseudo-operations placed in the calling sequence. It will write a given line on S.SOU1 and/or type it, convert floating point to fixed- or floating-point decimal numbers, fixed-point binary numbers to decimal, any 36-bit word to octal; and will print words already set in memory in binary-coded decimal form. Note that the routine can only be accessed via a TSX linkage and hence, is applicable only with MAP coded programs or subroutines.

4.2.1 CONVERSION SPECIFICATIONS

Use of the GGOUT subroutine will involve at least three instructions, the first of which is the entry instruction, followed by one of the conversion pseudo-instructions listed in Table 4-1, followed by a line-spacing instruction (PON or MON). These pseudo-operations will accomplish the appropriate conversion and store the word in the line image.

Table 4-1

GGOUT Pseudo-Operations

Code	Comment	Explanation
PZE	Plus Zero	Binary-to-Octal (Logical) Conversion.
MZE	Minus Zero	Integer-to-Integer Conversion.
PON	Plus One	S.SOU1 Line-Spacing Instruction.
MON	Minus One	Indicator for a type request.
PTW	Plus Two	Fixed-to-Fixed Conversion.
MTW	Minus Two	Floating-to-Floating Conversion.
PTH	Plus Three	BCD-to-BCD Conversion.
MTH	Minus Three	Floating-to-Fixed Conversion.

Note that in Table 4-2, the symbols D, PPP; and NN appear in the decrement. PPP, which appears in all the decrements, denotes the end print-wheel position and thus determines the column of the last symbol of a word. NN, used only with the PTH operation, indicates the number of BCD words (six characters each) to be printed, starting with the one in the address K. The normal range of PPP is 001 to 132, hence the range of NN is 01 to 22. Normally, then, the largest possible decrement is 22132. The D in the decrement indicates the number of decimal places to be used in the output, and has the range of 0 to 8. If, however, D, or NN is used, the PPP must always consist of a three-digit number (e.g., if DD is 8, and PPP is 15, then DDPPP must be 8015).

The tag, shown, in Table 4-2, may be used to modify the address of the conversion instructions. The tag may be a 0, 1, or 2. If a 1 or 2 is used, the modified address will be K minus the contents of the corresponding index register.

Table 4-2

Pseudo-Operation Instructions

Code	Address	Tag	Decrement
PZE	K	T	PPP
MZE	K	T	PPP
PTW	K	I	DPPP
MTW	K	T	DPPP
PTH	K	T	NNPPP
MTH	K	T	DPPP

4.2.2 PSEUDO-OPERATION DESCRIPTIONS

A brief description of each of the pseudo-operations associated with the GGOUT subroutine is presented here. Because address modification is permitted with the conversion operations, the effective address will be shown as L.

4.2.2.1 THE PTH PSEUDO-OPERATION

This instruction is used to write or type, from left to right, BCD words located in address positions L to L + NN. The last character of word NN is printed in print position PPP. In most cases, this character will be a blank space, since BCI listings usually do not contain exactly six characters in the last word. Also, 22 words may be printed with one PTH instruction, even though only ten can be entered into memory with a single BCI instruction.

If just the print position is stated (i.e., NN is omitted from the decrement), the word occupying position L will be printed with its last character in position PPP. However, if NN is correctly indicated, but PPP is not big enough to contain NN words, the PPP will automatically be increased to print the line in the first NN + 1 word location.

Greater speed may be achieved using the PTH conversion by employing multiples of 6 for PPP, since a 6-character word is transferred to the image as a whole. If PPP/6 leaves a remainder the transfer proceeds character-by-character.

4.2.2.2 THE MTH PSEUDO-OPERATION

This instruction converts the floating binary number in location L to a D-place fixed decimal number, where $0 \leq D \leq 8$. For $D = 0$, a rounded integer without the decimal point is the output. For a negative number, the minus sign is printed to the right of the number; this position is determined by the print-wheel position. Any number (N), where $N \geq 2^{35}$, can be converted to a floating-point decimal number by MTH.

2.2.3 THE PTW PSEUDO-OPERATION

This operation converts a fixed-point binary number to a D-place fixed decimal number, where $0 \leq D \leq 8$. The location of the binary point is indicated in the address portion of a PZE instruction, which must follow the PTW. If the binary point is outside the 0 to 35 range, the program will go to the next pseudo-operation. The binary point in the PZE address is determined by counting from the left-hand end of the word in location L.

4.2.2.4 THE MTW PSEUDO-OPERATION

This operation converts the binary floating number of location L to a D-place decimal number, where $1 \leq D \leq 8$. The answer format is:

X.XX..........XX± YY±,

where the positive signs and leading zero of YY, if any, are actually indicated by a blank space. A floating-point number whose binary exponent is zero is printed out as an integer by MZE.

4.2.2.5 THE PZE PSEUDO-OPERATION

This instruction converts the logical word in location L to a 12-digit octal number with no sign. A one-space separation is inserted between digits 6 and 7.

This conversion is accomplished by inserting three binary zeros before each set of three binary digits of the number in location L, thus transforming any three binary bits to the appropriate octal number expressed in BCD code. The PTH operation is used internally by PZE to transfer this information to the output.

4.2.2.6 THE MZE PSEUDO-OPERATION

This operation converts the contents of location L to a decimal integer with no decimal point. The sign is located immediately to the right of the integer.

?.2.7 THE PON PSEUDO-OPERATION

This operation is a line-spacing instruction for writing on S.SOU1.

The address portion of the instruction may be used for regular spacing, double spacing, or to restore a page before printing by specifying an address of 0, 3, or 1, respectively. If the address of a PON is -1, GGOUT will not alter the first character in the line to be put on tape.

4.2.2.8 THE MON PSEUDO-OPERATION

This operation indicates that the line image is to be typed. The decrement, tag and address of this pseudo-operation are ignored. A line may be written on S.SOU1 and typed by following the PON with a MON.

4.3 UTILITY ROUTINES

4.3.1 FILE NUMBER ACQUISITION WITH GETF

In many instances, a program will require the use of one or more working files for temporary use. To eliminate the need of depending upon fixed file numbers, the programmer can employ GETF to locate an available working file. The criteria used to judge availability is identical to that described in part b of paragraph 4.1.2.1. Linkage must be as follows:

```
CALL  GETF(AVAILF)
```

The argument, AVAILF, is the location into which GETF will store the file number. Note that a reel with an active file on it will be judged unavailable by GETF; hence, several working files could be located at one time with the following sequence:

```
CALL  GETF(WORK1)
CALL  ABOUT1(WORK1,LRS1)
CALL  GETF(WORK2)
CALL  ABOUT1(WORK2,LRS2)
CALL  GETF(WORK3)
        .
        .
        .
```

If GETF is unable to locate a working file, the condition will be treated as an error and a diagnostic will be given indicating the current file activity status of all reels. Execution of the calling program will then be terminated by returning to the monitor.

4.3.2 OBTAINING THE CURRENT DATE

A subroutine is available to the SPACE programmer enabling him to obtain the current date (including the day of the week) for page headings, etc. Linkage to the routine must be as follows:

```
CALL  MTWTF(ARRAY(1))
```

The argument, ARRAY, is the location of the first word of a five cell array into which the BCD date is to be stored by MTWTF. The date format will be as follows:

ARRAY(1) = W E D N E S
ARRAY(2) = D A Y b A P
ARRAY(3) = R I L b 7 ,
ARRAY(4) = b 1 9 6 5 b
ARRAY(5) = b b b b b b