

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

CR 151903

SIGMA CORPORATION
TECHNICAL NOTE

TN 76-115

(NASA-CR-151903) ENGINEERING DESIGN AND
INTEGRATION SIMULATION UTILIZATION MANUAL
(Sigma Corp., Houston, Tex.) 77 p
HC A05/MF A01

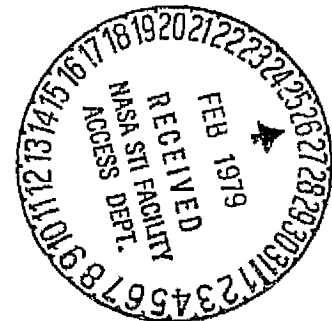
N79-17577

CSCI 09B

Unclas

G3/61 13937

ENGINEERING DESIGN AND INTEGRATION
SIMULATION UTILIZATION MANUAL.



By: Guy N. Hirsch

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Johnson Space Center
Houston, Texas 77058

July 1976

PREFACE

This report provides a description of the Engineering Design Integration (EDIN) Simulation System as it exists at Johnson Space Center. A discussion of the EDIN Simulation System capabilities and applications is presented. This report was written in support of NASA Contract NAS9-14520. The contract was monitored by Mr. Robert W. Abel. The report specifically describes a user-developed data processor which is integrated with the Univac 1100 Executive System.

TABLE OF CONTENTS

	Page
SUMMARY.....	1
INTRODUCTION.....	3
ENGINEERING ASPECTS OF THE SIMULATION EXECUTIVE.....	5
Concepts and Definitions.....	7
Language Structure and Usage.....	8
Construction and Maintenance of Data Bases.....	9
Construction of Partial Run Streams.....	13
Execution of a Sequence of Technology Programs	
Conditional Branching	
Retrieval of Design Information.....	14
Replacement Command	
Insert Command.....	16
Comment Command.....	17
Report Generation	
Technology Module Interface Package.....	18
PROGRAM UTILIZATION.....	20
DLG Usage	
Control Statement	
Option Specifications	
Syntax Definition.....	21
Description of Control Directives.....	22
Examples.....	25
UNIVAC/ADAGE DATA TRANSFER.....	29
Use of the Univac to Adage (UTA) Routine	
Use of the Adage to Univac (ATU) Routine.....	30
EDIN EXEC 8 PROGRAM LIBRARY.....	32
Geometry	
Aerodynamics, Stability and Control	
Propulsion.....	33
Mass and Volumetric Properties	
Performance.....	34
Thermodynamics	
Structures.....	35
Cost	
Environmental Protection	
General and Special Purpose Utilities	
ABLATOR: One Dimensional Analysis of the Transient	
Response of Thermal Protection Systems.....	36
ACMOTAN: Linear Aircraft Motion Analysis	
AESOP: Automated Engineering and Scientific	
Optimization Program.....	37
AFSP: Automated Flutter Solution Procedure	
AIRFOIL: A Program for Generating Geometric and	
Aerodynamic Characteristics of Airfoil	
Sections.....	38

TABLE OF CONTENTS (Cont'd)

	Page
APAS:	Automated Predesign of Aircraft Structures..38
ATOPII:	Atmospheric Trajectory Optimization
CIPHER:	Report Writing Program.....39
COAP:	Combat Optimization and Analysis Program
CONPLOT:	Aircraft Configuration Plot.....40
COT PROCESSOR	
CUSSER:	Upper Stage Sizing Program
DAPCA:	Development and Production Cost of Aircraft
DATCOM:	Configuration Design Analysis Program (TRW)
DATCOM2:	Configuration Design Analysis Program (MDAC).....41
ENCYCL:	Design Point Performance of Turbojet and Turbofan Engine
FLOWGEN:	Automatic Flowchart Program
GENENG:	A Program for Calculating Design and Off-Design Performance for Turbojet and Turbofan Engines
GENENGLI:	A Program for Calculating Design and Off-Design Performance of Two and Three Spool Turbofans with as many as Three Nozzles.....42
GEOMETRY:	Body Coordinate Generator
GTM:	Geometry Technology Module
HABACP:	Hypersonic Arbitrary Body Aerodynamic Computer Program.....43
HDG:	Heading Program
IMAGE:	Configuration Display Program
MINIVER:	Aerodynamic Heating Program.....44
PADS:	Performance Analysis and Design Synthesis Program
PANEL:	A Program for Generating Panelled Configuration Geometry.
PLTVIEW:	Program for Generating Separation Plots.....45
POST:	A Program to Optimize Simulated Trajectories
PRESTO:	Program for Rapid Earth-to-Space Trajectory Optimization
PRICE:	A Program for Improved Cost Estimation.....46
RDPRO:	POST Plot Data Generation Program
REPORT:	Report Generator
SBOOM:	Sonic Boom Prediction for Shuttle Type Vehicles
SEPARTE:	The Program to Simulate Separating Stages of Launch Vehicles.....47
SFIT:	A Surface Fitting Program
SIZER:	A Preliminary Sizing Program for Launch Vehicles
SKINF:	Turbulence Skin Friction Drag Program.....48
SSAM:	Swept Strip Aeroelastic Model
SSSP:	Space Shuttle Synthesis Program
TOLAND:	Take-Off and Landing Program.....49

TABLE OF CONTENTS (Cont'd)

	Page
TOP:	Trajectory Optimization Program.....49
TOPLOT:	Plot Generator for TOP
TREND:	Subsonic/Supersonic/Hypersonic Aero- dynamic Trade-Off Program.....50
VAMP:	Volume, Area and Mass Properties
VL70:	A Program for Reading Aerodynamic Data Tapes
VSAC:	Vehicle Synthesis for High Speed Aircraft.....51
WAATS:	Weights Analysis for Advanced Trans- portation Systems
WAB:	A Program for Computing Weights and Balances.....52
WDRAG:	Zero-Lift Wave Drag Program
WETTED:	Wetted Area in Reference Length Program
XLT:	Translator Program
EDIN ADAGE GS340	PROGRAM LIBRARY.....53
IMP:	Interactive Mass Properties
IMAGE:	Cornerpoint Geometry Display
FUSEGEN:	Fuselage Generator.....54
PARAGEOM:	Parametric Geometry Program
WINGEN:	Wing Generation
SAMPLE ANALYSIS.....	55
Definition of Analysis Tasks	
Selection of Analysis Programs.....	57
Intercommunication Data	
Generation of a Run Stream.....	58
REFERENCES.....	68

ENGINEERING DESIGN AND INTEGRATION
SIMULATION UTILIZATION MANUAL

By: G. N. Hirsch - Sigma Corporation

SUMMARY

The EDIN system is a digital computer program complex for the evaluation of aerospace vehicle preliminary designs. The system consists of a Univac 1100 series computer and peripherals using the Exec 8 operating system, a set of demand access terminals of both the alphanumeric and graphics types and a library of independent computer programs. The program library contains programs for estimating all major flight vehicle characteristics such as aerodynamics, propulsion, mass properties, trajectory and mission analysis, cost, steady state aeroelasticity, flutter and stability and control. There are also utility programs in the library for generating and controlling the flow of design data within the computer program complex.

The data base used by the EDIN system consists of dynamically constructed name addressable region of online storage which can be subdivided into user established classes of data. Each class represents a subset of the available data. A data manipulation program called the DLG processor is used for the construction and maintenance of the data base. It interrogates the data base to satisfy data interface requirements among the technology programs. In addition, the DLG processor can address all or portions of online structured data sets for insertion as part of the input stream of the technology programs.

The DLG Processor is an 1100 series Exec 8 computer program designed to read, modify, manipulate and replace symbolic images. DLG is controlled by a set of user supplied directives (or language elements) which provide the basic capabilities of the DLG Processor as follows:

1. Language elements for the construction and maintenance of a data base which is independent of any other computer program.
2. Language elements for processing information generated by other computer programs.
3. Language elements for automatically retrieving data base information as input to any other computer program.

4. A simple technique for editing and interrogating the data base and for generating summary reports of data base information.

The DLG processor can be used to form a linkage between engineering technology modules through the manipulation of common information in the data base. The use of the system for this purpose requires the prior assimilation of the following basic components:

1. A library of independent technology programs including the DLG processor.
2. The control card sequences for the execution of the technology modules.
3. The setup data for the technology modules which perform the desired analysis function.

The basic components are often available without additional program development. The program sequencing and intercommunication required to integrate the basic components into a design simulation are established using the Exec 8 run stream concept. (See Reference 1) Data base requests for common information are established by the formation of skeletonized data elements containing execution control cards and/or technology module input data. The skeleton elements containing DLG directives are processed or filled out by the DLG Processor using data base information. The result of the preprocessing is a partial run stream which is acceptable to Exec 8 for performing computerized design tasks.

The EDIN Simulation System can be operated in demand or batch environment. In the demand mode some interaction with the data base and technology module is available. Usually a combination of demand and batch operations is employed in the evaluation of preliminary designs using the system.

INTRODUCTION

The EDIN system provides a balance of data management techniques which consider the inherent capabilities of the computer operating system, past efforts in the storage and retrieval of stratified data and the recent development of some flexible paging techniques for the transfer of information between the computer core and the mass storage of the computer. The Univac Exec 8 system provides the resources for the storage of large complex data files, for the storage and retrieval of the files and for the cataloguing protection and backup of the files. The executive system has several processors with instruction sets for manipulating the data retained in mass storage. A limitation on the operating system capabilities arises in accessing the subfile level of information in the system files once the file is addressed.

The EDIN data management system is designed to subdivide the files in a manner that will allow the data which is retained in mass storage to be accessed at any level from the single parameter level to a large matrix of data. Rather than constructing an extensive single computer program that attempts to be everything to everyone, the EDIN data management system provides a three-level data management capability. This approach permits the individual designer using the system to make his own decisions with regard to the storage method and techniques. It also permits the flexibility of using existing data sources not specifically created for EDIN.

The three levels of the EDIN data management system are built upon one another as illustrated in figure 1. The lowest level deals with the interface between the data in mass storage and the computer operating system. The file level of the data management system is provided by the Exec 8 software and consists of the file utility processor FURPUR, the file administration processor SECURE and other system level processors. The system processors are accessed using Exec 8 control statements. Therefore, file level software may be used directly by the designer for transmitting large structured blocks of data or the files themselves to be accessed by the programmer who seeks economy above all else. The file level constitutes the foundation for all higher level data management components.

The second level of the EDIN data management system provides the mechanism whereby the files can be organized into blocks of data called pages. Pages of information can be organized in a number of ways and names can be given to each page. A pointer system or directory is maintained by a Fortran callable software

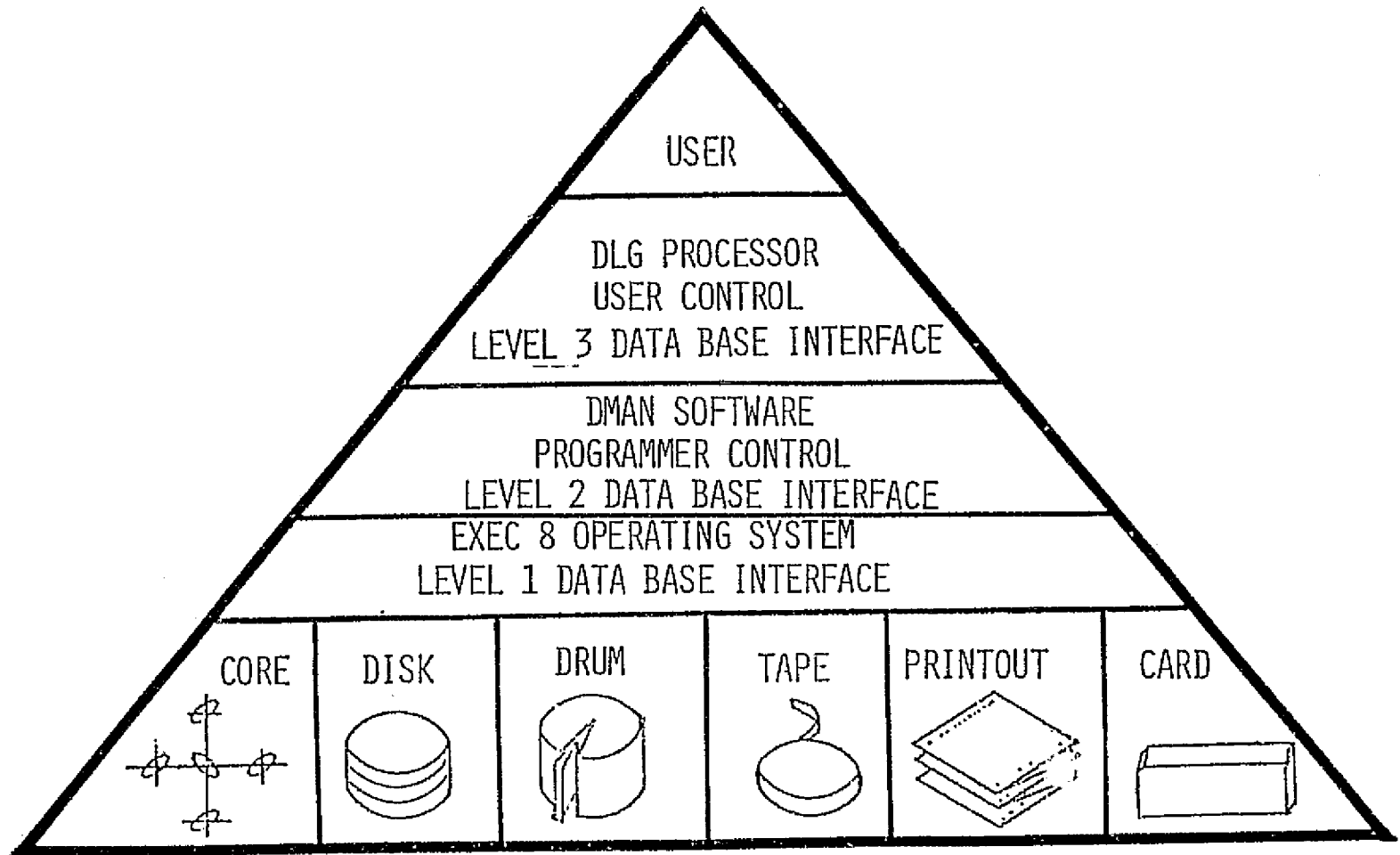


FIGURE 1 EDIN DATA MANAGEMENT SYSTEM.

package, called DMAN, a subroutine utility package maintained in the EDIN library.

The third and highest level of the data management system is provided to make the system more usable to the designer who may not be a programmer. The capability is provided in the DLG processor which is designed to maintain a data base of stratified information, the stratified data can be selectively accessed and merged with the input stream of the EDIN technology programs. This level also provides the interactive language structure which allows the designer to sit at a remote terminal and interact with the data base directly as he develops a design. The DLG processor also contains routines for processing the output from the technology programs for the storage of design information in the data base.

Although the user may access the data base through any of the three levels, it is the lowest level maintained by the Exec 8 system which actually stores and retrieves the data. Exec 8 handles all of the underlying data management functions including file assignments, file directories and maintenance and security procedures as well as the data block transfer to and from mass storage. The Exec 8 system is discussed in the previous section and a thorough treatment of the first level data management is provided by Univac in the appropriate User Documentation. The following discussion deals with the second and third level of the EDIN management system.

Complete instructions for use of the DLG language elements in conjunction with the basic EDIN simulation components are presented herein. In addition, an interface technique is described for allowing any program in the library to update the data base. The technique does not influence the stand-alone operation of the program.

ENGINEERING ASPECTS OF THE SIMULATION EXECUTIVE

The DLG procedure for implementing the linkage of the technology modules is to read output data from one module, insert a selected subset of the resulting data into a stratified data base and then selectively extract this and other stored data for placement into the input stream for other applications programs. The linkage is illustrated in figure 2. The effect is to provide a unified analysis involving several modules operating from a single source of data. Repetition of execution sequences can be triggered by looping criteria residing in the data base.

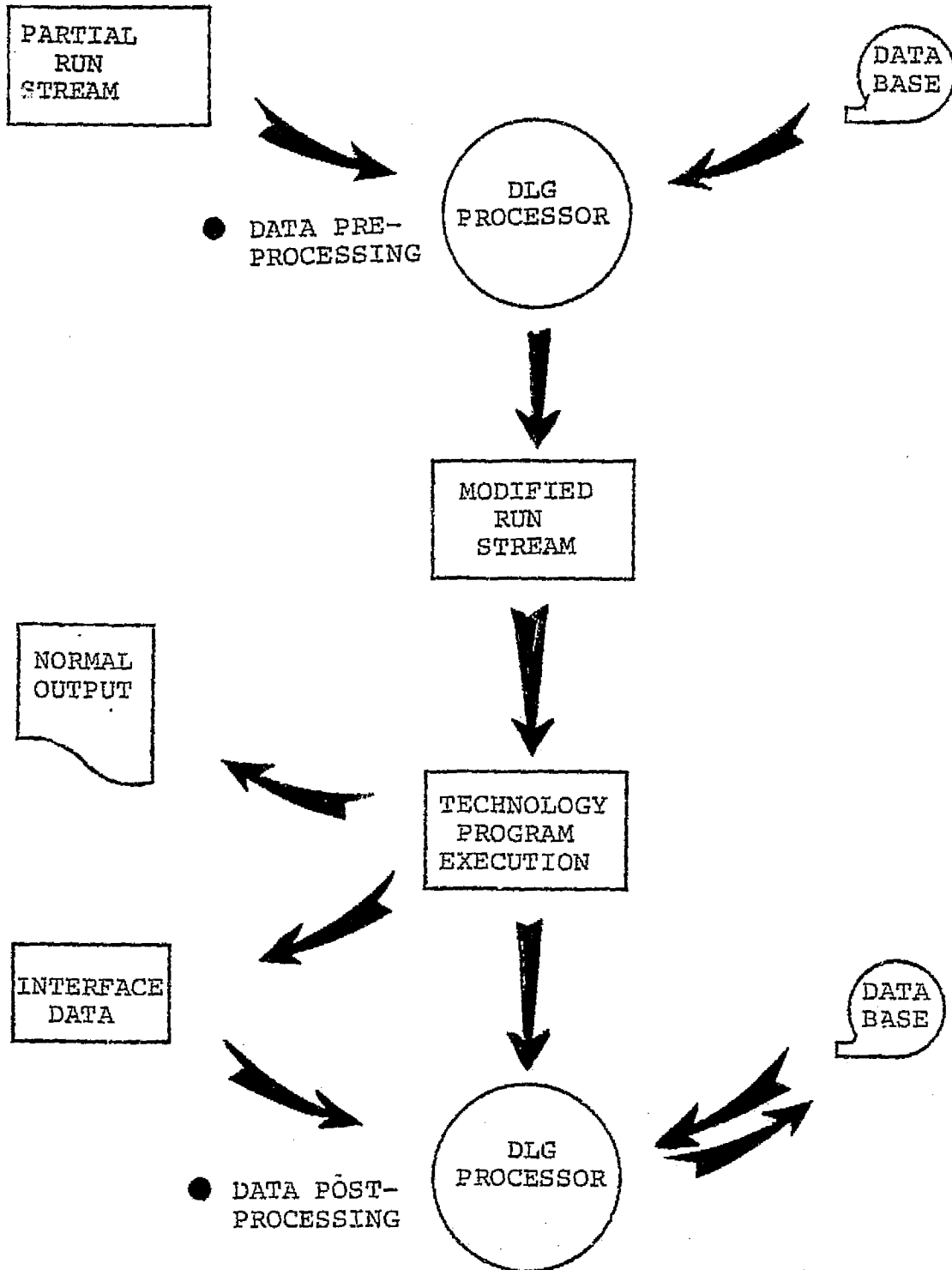


FIGURE 2 DLG PROCESSOR FUNCTIONS.

Concepts and Definitions

The following concepts and definitions which may be new to the reader will be helpful in understanding this document:

Processor	An absolute program element which is executed with a special Exec 8 processor control statement: @name elt1,elt2 and which is interfaced with the elements named on the processor control statements.
Data Base	File of information which is subdivided into named pages of data accessible by the DLG processor. Each page is further subdivided into named parameters and arrays.
Technology Module (Application Program)	An independent computer program which will receive or generate data base information.
Interrogation	The process of retrieving information from the data base. The disposition of the retrieved data is dependent upon the directive employed.
Directive (Also Command)	A language element used to specify a DLG Processor's action or function.
Data Management	A class of DLG functions which control and manipulate data base information. These functions include the creation of data base pages, the adding and defining information in the data base, printing and many others.
Data Storage	A special class of data management functions which are designed specifically to store data generated by a technology module.
Run Stream	A sequence of data images which constitute a computer run.
Partial Run Stream	A portion of a run stream which can be merged at any point in the run stream through an @ADD control statement.

Language Structure and Usage

The language for controlling the DLG Processor consists of control directives which are summarized below:

'name'	Replace name with information from the data base.
'ADD	Replace specified information in the data base.
'CHANGE	Change values in common IDLOG.
'COMMENT or '.	User description with null effect.
'CREATE	Create a new data base.
'CSF or 'ER	Submit executive control statement.
'DBLIST	Print the names of all random access data bases on the data base file.
'DEFINE	Place description in data base directory.
'FORMAT	Format free data base information in place.
'INSERT	Insert binary SDF data in place.
'ON	Mode activation.
'OFF	Mode suppression.
'PRINT	Print data base information.
'USE	Specify a circular data base search.
'UPDATE	Update a specified data base.

Each language element is delimited by a pair of apostrophes (') to distinguish DLG input from the technology module input data and execution control cards in the run stream. For example, in the language element,

'NAME.....'

NAME is the name of the command or directive (i.e. CREATE or ADD). The remaining information within the delimited region is ancillary to the specific directive. The DLG Processor scans the partial run stream for delimited information and performs the function specified therein.

The control directives provide the means by which data base storage and retrieval functions can be performed. They command the transfer of information from the analyst to the data base, from the data base to the technology programs and from the technology programs to the data base. In the transfer of information from the analyst to the data base, the ADD, DEFINE and INSERT commands are used. A comment command is also available for annotating data. In the transfer of information from the data base to the applications program, the replacement command is used. Data from the data base is transferred by name to the input stream of the technology programs. In the transfer of information from the technology programs to the data base, a special extension of the ADD command is employed. An interface file of ADD-like commands can be generated by any technology program for later processing by the DLG Processor.

Construction and Maintenance of Data Bases

Data base construction and maintenance are accomplished by the use of the three control directives, CREATE, UPDATE and USE. The CREATE directive creates a new data base which is stored on a temporary disk file by the same name. By permanently storing the data on the disk file, the user may access the information at a later time. The use of the UPDATE directive in conjunction with previously created data bases, allows the user to modify the data contained therein. The USE directive specifies a search sequence of existing data bases.

The basic construction of data base pages is similar but varies in total size and specific characteristics which can be specified by the user at creation time. Data bases consist of two distinct parts, a free storage array of packed information where the actual data is stored and a directory of names and pointers to the data in the free storage array. The directory and the data base have certain attributes which are assigned by the CREATE directive.

```
'CREATE name, attribute = value, attribute = value'  
(DATA)
```

Any or all of the attributes may be specified by the user as follows. If not specified, the default values will be used.

<u>Attributes</u>	<u>Default</u>	<u>Description</u>
LTOTAL	1024	Total number of computer words allocated to the specified data base.
NWORD	2	Number of computer words per data base entry.
DIRLEN	47	Total number of directory entries allocated to the data base.
LENDES	5	Number of words of descriptive information associated with each entry in the data base.

Each entry in the directory is the name of a data element or the name of an array of data elements. Up to DIRLEN entries may be made. Each entry may have a short description (LENDES-2) stored with it. Each data element occupies NWORD computer words. In establishing the size of the data base (LTOTAL), the user must consider the total number of elements of data desired in the data base, the directory length, the desired length of description and the number of words per data element. If NELMT is the total number of elements, then:

$$LTOTAL = DIRLEN * (LENDES + 3) + NELMT * NWORD$$

After a data base is initially established, the UPDATE directive may be used for adding or modifying the information contained therein. The form of the directive is:

```
'UPDATE name'
(DATA)
```

Attribute parameters which are established once by the CREATE directive may not be specified by the UPDATE directive. If the use of the existing data base pages without modification is desired, the directive:

```
'USE name,name,...'
```

is used. No attributes may be specified.

The basic control directive for entering information is the ADD command. It permits a variable name and value or values to be placed in the data base:

```
'ADD name=value,value,'
```


If the name is a new entry, any number of values may be added. The number of values associated with the name when it is first added is also the number of locations reserved in the data base for that information. Later modifications to the information can not create more data base space. The values may be real, integer, hollerith or logical. A single ADD command may be used for creating or updating many information sets.

```
'ADD V1 = 25., V2 = 30, V3 = ALPHA, V4 = .TRUE.,
```

```
  A = 10., 15., 20., 25., I = 4, 5, 6
```

```
  V4 = '5 * 0.,'
```

The data type is defined by the input data type. Upon retrieval, the data type is determined by the characteristics of the stored data. The format of the ADD statement is patterned after the FORTRAN NAMELIST feature and indeed has many of the same characteristics and usage rules. For example, all name/value sets are separated by commas (,); all elemental values of an array are separated by commas; the entire statement (command) is delimited. In the case of NAMELIST, the delimiter is a dollar (\$) sign; in the case of ADD commands, the delimiter is ('). The rules for entering hollerith information and multiple constants differ from NAMELIST.

The ADD command has additional capability not present in NAMELIST. The value associated with the ADD name may be a previously defined data base variable name:

```
ADD V1 = V2,
```

The effect of the above command is to transfer the information associated with V2 to the data base space assigned to V1. V1 may or may not exist prior to the ADD command. If V1 did not exist, space will be created in the data base as the information is transferred. If V1 did exist, then the information in V1 will be replaced by the information in V2. The transfer of information from one data base location to another is generally limited to scalar quantities. The ADD command may also be used for combining existing data base information with other data information or constants:

```
'ADD V1 = V2 * V3,'
```

The operation illustrated above indicates a multiplication of the two numbers on the right side of the equal (=) sign prior to

transferring the resulting information to the space allocated to V1. Any algebraic operator may be employed as follows:

- + addition
- subtraction
- * multiplication
- / division
- ** exponentiation

More than one operation may be performed on the right side of the equal (=) sign.

```
'ADD V1 = V2 + V3 * K,'
```

Up to ten operations may be performed within a single ADD command. The operations start from the equal sign and progress to the right. The first variable is combined with the second. The result of that operation is combined with the third. The result of that operation is combined with the fourth, etc. It should be noted that the hierarchy or order of the operations does not conform to FORTRAN arithmetic. For example, in the above illustration, V3 is added to V2, then the sum is multiplied by K.

If the data base were so defined, the directory may contain space for storing descriptive information. Depending on the value of CREATE parameters, LENDES space is reserved for each name entered equivalent to:

```
LENDES -2
```

The default value of 5 provides three computer words (30 characters) for descriptive information. The DEFINE command is used to store the descriptive information in the directory:

```
'DEFINE name, description,'
```

```
DEFINE name = value, description,'
```

Name is a new or existing data base entry. If the name exists, the description is added. If not, the name and description are added. If the name is a new entry, then the value may be used to reserve space in the data base for data elements to be added later. The absence of value on a new entry results in the reservation of space for a single data element.

Construction of Partial Run Streams

There are three major considerations in the construction of design partial run streams:

1. Establish a design data base and control card data base for use in the simulation.
2. Construct the desired design sequence.
3. Provide data base interfaces within the input data sets for the technology programs.

Data bases are established at the beginning of the run or transferred from mass storage on an one-time basis using the techniques discussed above. The design data base is defined first by use of a CREATE, UPDATE or USE directive.

Execution of a Sequence of Technology Programs. - Now consider the problem of sequential execution of one or more technology programs using the EDIN system.

DLG preprocesses the data associated with each program and constructs a modified run stream. The preprocessing function will be discussed later. The physical linkage between the control card sequences of the various applications programs and DLG is an Univac operation system utility program, @ADD. @ADD is executed to link the technology program sequences just preprocessed by the DLG program. The linkage is continued by Exec 8 until an @FIN control statement is encountered. Details of the operation of the executive system are contained in reference 1.

Conditional Branching. - The DLG processor can direct the transfer of control forward or backward in the simulation stream by the use of dynamic run stream modifications. This capability is achieved by the following control card directive language elements:

```
@SETC 'V2'  
@TEST OP/'V1'  
@JUMP label
```

The @JUMP control statement establishes a label name in the execution sequence where control may be skipped to. The @SETC control statement establishes a data base value in a special system register which is tested by the @TEST control statement to determine if the @JUMP control statement should be processed.

The combination of the DLG Processor functions and the Exec 8 dynamic run stream modification capabilities permit a complicated system of analysis loops to be constructed for satisfying a variety of matching constraints.

The @TEST employed provides a standard set of tests on the variable set by the @SETC control statement:

```
@TEST TG/Value      for (V1 > V2)
@TEST TLE/Value     for (V1 < V2)
@TEST TE/Value      for (V1 = V2)
@TEST TNE/Value     for (V1 ≠ V2)
```

As noted previously V1 and V2 are constants or variables constructed in the design data base through use of the ADD command or generated by any technology program in the synthesis and passed to the design data base.

Retrieval of Design Information

The retrieval of design information from the data base is accomplished by two basic control directives:

1. Replacement Directives.
2. INSERT Directives.

The control directives are strategically placed into the technology program input data sets. The augmented data sets form skeletonized input for the technology programs. Each data set is headed by UPDATE or USE control directives corresponding to the appropriate data base pages.

Upon execution of the DLG Processor, the data sets are scanned for control directives delimited by apostrophe pairs (') for identification. Each encounter with delimited information causes DLG to perform the indicated replacement or insertion action. The result of all encounters within the skeletonized input for each technology program is a modified partial run stream which can be merged with the run stream through the @ADD control statement.

Replacement Command. - Data base information may be entered into the technology program data sets by means of delimited data base variable names entered in the precise location where the requested data is to be placed. The delimiters define the field width for simple replacement (i.e. a single data element). The

DLG processor will replace the variable name and delimiters by its data base value and rewrite the card image in the modified run stream. Input formats such as NAMELIST-like inputs and rigidly formatted input can be accommodated by the procedure. For example, in a true NAMELIST input, a data base variable would be entered as follows:

```
NAME1 = 'V1 ',
```

NAM1 is the name of the NAMELIST variable. V1 is the data base name. The delimiters specify the field width to be employed in replacing the data base name with the corresponding data base value. Similarly for a formatted input where data normally appears on a card within a specified range of columns, the data base replacement procedure is simply:

```
'V1 '
```

The delimiters are placed at the appropriate card columns defining the field for the data element. The data base name is placed within the defined field adjacent to the first delimiter.

Additional capability is available when namelist input is used by the technology program. Entire arrays may be transferred to the input stream using the following procedure:

```
NAME = 'VARRAY'
```

where VARRAY is a data base array name. If the data in VARRAY contains more data than can fit on one card record, additional 'cards' are created and placed in the modified input stream to accommodate the excess data.

Data base variables and constants may be combined much like the capability described for the ADD command above. For example, the operation:

```
'V1 * V2'
```

illustrates how the product of V1 and V2 may be used as the replacement value of data base variable. The product never resides in the data base but only in the modified input stream. V1 must be a data base variable name but V2 may be a data base variable or constant. More than one arithmetic operation may be performed within the delimiters:

```
'V1 + V2 * V3'
```

Up to ten operations may be performed within a single command. The hierarchy of operations is the same as that described for the ADD command.

In general, array elements may also be used in the replacement command:

```
'V1(n)'
```

where n is a data base variable, or

```
'V1(5) * V2(6)'
```

One exception from this capability is in dealing with the first element of an array:

```
'V(1)'
```

The above command specifies the replacement of an entire data base array and is therefore not an acceptable command for replacing the first element of an array. The dilemma may be avoided by use of the following two cards:

```
'ADD NEW = V1(1)'
```

```
'NEW'
```

The ADD command defines a new data base variable which will contain the element V1(1). The replacement command will place the variable NEW (i.e. V1(1)) in the modified input stream. In general, all ADD command capability described under data base construction is applicable when placed in the skeletonized element of a partial run stream. The 'ADD command instructions are performed but the 'card' which contains the command is not transferred to the modified input stream.

A special feature of the replacement command is the element-by-element combination of entire arrays or the combination of arrays with constants. As an illustration, consider the example:

```
'V1 * V2'
```

where the data base variables V1 and V2 are arrays. The above command specifies the element-by-element multiplication of the arrays. If one array has fewer elements than the other, the combining of elements ceases after the shorter array is exhausted and the rest of the longer array remains unchanged. The variable V2 may be a constant or data base name. Multiple operations may be performed using the hierarchical rules described above.

Insert Command. - Unlike the replacement command which extracts information from the design data base, the INSERT command transfers information sequentially from a formatted data card file. Starting and stopping positions within the file may be specified as follows:

'INSERT name1 = n/m, name2 = j/k'

The effect of the command is to search the named system files, name1 and name2, for integer card numbers n through m and j through k. If the card numbers are omitted, the entire named file is transferred to the modified input stream. The end-of-file may be specified for the named file by replacing m or k with the character string, EOF. The card containing the INSERT command is not transferred to the modified input stream.

Comment Command. - In addition to previously described commands, there exists a special "command" for identifying data. The format is:

' . comment'

No action is performed as a result of this command. It is useful only as an identifier for other data. For example, consider a technology program which uses formatted input (i.e. numbers with no identifiers or names associated with them). The comment command may be used to identify the data elements within the input data stream. The effect of processing this command through DLG is simply the replacement of the command with blanks. If the resulting card image is entirely blank, then the card is not transferred to the modified input stream.

Report Generation

A special feature of the DLG program is its ability to produce user generated report formats augmented with data base information. The report generator is applied in the following manner.

The report data is formatted by the user to provide any descriptive information desired. The report data may contain data base information through the use of the communication commands described above. One example of card image in the report might read:

WEIGHT OF THE SYSTEM IS 'WGT' POUNDS

In the above illustration, WGT is a data base variable name. The DLG Processor replaces the data base name and delimiter 'WGT' with the information stored in the data base. The report is printed through the normal computer output channels. The insertion of a report in the simulation stream does not effect the normal sequence of events for the simulation. The report may contain carriage control characters in column 1 of the report data cards.

1 - Eject a page before printing.

0 - Skip a line before print

Any number of reports may be generated during a simulation. The format of the individual reports is tailored to the needs of the particular study. Once the format is established, it can become a permanent part of the simulation stream. Any of the features of the DLG language, including scaling and adding data base information, are used in a completely free field report format.

Technology Module Interface Package

The communication of information from a technology program to the EDIN data base generally requires modification of the applications program. This modification is usually trivial and requires little programming knowledge to accomplish. The objective of the modification is to create a special file of information which contains a format suitable for reading by the DLG processor. The information is placed on the special file by the technology program. The file is later integrated by the DLG for possible placement of the information into the EDIN data base.

A series of four routines for printing the common types of data in a format readable by DLG are available. They may be called at any point in the calculation sequence for generating EDIN output. The format simulates the control directives format used in the DLG processor.

- ADDREL - For printing real variables and arrays.
- ADDINT - For printing integer variables and arrays.
- ADDHOL - For printing Hollerith variables and arrays.
- ADDLOG - For printing Logical variables and arrays.

The output is similar to the format of NAMELIST for one variable name only with any number of associated values. Each subroutine has the same calling sequence characterized as follows:

CALL ADDREL (LU, NAME, NUM, VALUE)

LU - Logical unit or special output file.

NAME - Desired name chosen by the analyst/programmer. It may be a stored name set by a Fortran data statement or can be set in the calling sequence as nHname.

NUM - Number of values in the array. For a single variable NUM=1.

VALUE - Internal variable or array name (starting location).

The subroutines for the other variable types have the same calling sequence. The primary difference among them is the format used for writing the variables and the special output file. Each output is a DLG control directive format. The name associated with the directive is set by a data statement in the individual subroutines. The data statement may be set at the time the technology program is modified. Usually it is desirable to use a name which is reminiscent of the application program name. The selected name may be precisely the same as the acronym used to execute the application program in EDIN. The reason for such a choice is that the directive name is stored in the EDIN data base. A print of the data base prints the last directive which updated each variable in the data base.

For most technology programs, the use of the software described above is adequate. However, certain programs generate data base information in a Fortran "DO LOOP." In these instances, the package (by itself) can not satisfy the EDIN requirement of separate names for different data elements and arrays.

The most convenient way to make this program and others of this type compatible with EDIN is to provide some name-generating capability with the applications program. Function subroutines which provide this capability can be called as illustrated below:

NAMGEN (NAME, K, J)

NAME = The desired root name.

I = Concatenated number occupying the first one or two BCD character positions beyond the root name.

J = Concatenated number occupying the second one or two BCD character positions beyond the root name.

An example would be:

NAM=NAMGEN (4HNAME,1,2)

In the above illustration, the name NAME would be extended by the BCD characters 1 and 2 concatenated to it and stored in NAM.

NAM=6HNAME12

A maximum of 6 characters may be generated. This limit is imposed by the word size limit for EDIN data base names.

Usually the NAMGEN function is used in conjunction with the NAMELIST simulator described above in the following manner:

```
CALL ADDREL (LU, NAMGEN (NAME, I, J), NUM, VALUE)
```

In the illustration, the name is generated within the calling sequence of the subroutine which prints the simulated namelist for the generated name.

PROGRAM UTILIZATION

DLG Usage

Control Statement.-

```
@DLG.DLG,options lfn.elt1,lfn.elt2
```

Option Specifications. -

- I Source input will follow the processor card. Source output will be placed in elt1.
- L Source input data will be listed.
- O Source output data will be listed.
- D Card cracking information will be listed.
- E Solicitation and result of directives will be printed.
- S List interrupt mode will be invoked.
- M New data base files will be generated with this execution.
- B Build option will be invoked. This option specifies that all data directives of the form:

```
'name name=value---
```

or

\$name name=value---

This will permit the addition of data to the data base regardless of the directive name. Otherwise, only those data base variable names, which were previously defined in the data base, will be updated unless the data directive name is ADD or DEFINE.

The B option may not be invoked via the "ON" command. If desired, it must be present on the processor call card.

Syntax Definition. -

name	Must be six (6) or less alphanumeric characters and begin with an alphabetical character.
' (quote or prime)	The DLG delimiter. Strings that occur between pairs of delimiters will be processed by DLG. Strings external to primes will be passed "as is" into the output element.
-	The underline on a command indicates an optional character string which may be used as a directive.
value	Indicates a data base value in real, integer or hollerith format.
i,j,k	Indicates integer constants used in the directives.
elt	Exec 8 file element name in program file format.
lfn	Exec 8 logical file name in system data format.
text	Textual information.
[]	Indicates optional items on the line.

Descriptions of Control Directives. -

'ADD name' - Specifies that information will be added to data base.

'ADD name=value'

'ADD name=name'

'ADD name=value,value,---'

'ADD name=name,name,---'

'ADD name=name op name, name op value,---'

	+	Add
	-	Subtract
where op =	/	Divide
	*	Multiply
	**	Exponentiation

'CHANGE number=value' - Using the integer number 'number' as an index into the master common block, IDILOG, the current value is replaced by 'value.'

'COMMENT _____' - This is a null card and is discarded by DLG.

'CREATE name,DIRLEN=number,LENDES=number,LTOTAL=number' - The data of name 'name' is brought into existence on the data base file. Optional parameters are DIRLEN - the directory length (This should be a prime number.).

LENDES - Length in computer words of the description.

LTOTAL - Total size, in computer words, reserved for the data base.

'CHANGE' -

Example 'CHANGE 27=3'

Location 27 of the common block IDILOG will have its value replaced by an integer 3.

'COMMENT' - A null card. The delimited field is removed from the card. If the resulting card is BLANK, the card will be removed from the run stream.

'CSF @ Control Statement' - Specifies that an execution control statement will be processed using the standard CSF\$ package. The following control statements may be used:

@ADD	@CKPT	@RSPAR
@ASG	@FREE	@RSTRT
@BRKPT	@LOG	@START
@CAT	@MODE	@SYM
@CKPAR	@QUAL	@USE

Example:

```
'CSF @USE 25, DBASE'
'CSF @ADD DUSEFIL.DLOG'
'CSF @QUAL B'
```

'DEFINE name=value,text' - Stores a textual description with the name in the data base directory. If the name is a new directory entry, the value is the number of data base entries allotted. Existing data is unaffected and new data is not added.

'DEFINE A, LETTER 1' - Stores the description, LETTER 1, with the name A.

'DEFINE B=10, BARRAY' - Stores the description, BARRAY, with the variable name B and allots 10 data base entries for B.

'FORMAT name=value/value, (Fortran compatible format statement)'
Extracts freely stored data from the data base and places into the output elements in accordance with the given format.

```
'FORMAT A=6/3, (1X, 3F15.3)'
```

The six items of A are output into the named element, 3 on a line through the (1X, 3F15.3) format.

'INSERT name=value/value' - Specifies that binary coded information the SDF file name will be placed in the source output element in 14A6 format.

'INSERT A' - Entire file of data in A will be transferred to source output element.

'INSERT B=5-13' - Insert data from B from records 5 through 23.

'INSERT C=5*EOF' - Insert records from file C records 5 to the end-of-file.

Other Examples - 'INSERT A,B=5-23, C=5*EOF'

'name' - Specifies a simple replacement of named information with data base parameters or arrays.

'REAL'	Real parameter or array.
'INTEG'	Integer parameter or array.
'HOLLITH'	Hollerith parameter or array.
'LOGICL'	Logical parameter or array.
'ARRAY(j)'	Real or integer element of an array, j must be a constant greater than 1. A value of j=1 will cause the transfer of all of j.

'ON name,name---' - Mode activation directive.

'OFF name,name---' - Mode suppression directive.

P or PAGDMP	Print card cracking information.
O or OUTDMP	List logical file 1 data.
N or INDUMP	List source output element.
C or CONTINUE	Activate continuation card option.
L or LIST	List source input information.
S or SPLIT	Interrupt mode.
E or EDIT	Edit mode (demand response to printer).

'PRINT name' - Specifies that data information will be printed.

'PRINT name=A,Z'	Print all information in name.
'PRINT name=n,m'	Print entries n through m alphabetically.
'PRINT name'	Directory and first data base entry of named data base.
'PRINT'	Directory and first data base entry of current 'USE' assigned data bases.

'USE' -

'USE A,B,C' - The data bases named will be circularly searched in the order given. The variables used in replacements. All will be searched once before a NO FIND is declared. It should be noted that this command may cause very excessive SUP changes if not carefully used.

'UPDATE name' - Specifies that the named data base will be updated with the information which follows:

'UPDATE A' - Specifies that the data base A will be updated with the data which follows.

Examples

Example 1. - Data base creation.

```
@fn.DLG,GI .RPT
```

```
'CREATE AERO,LTOTAL=4000,DIRLEN=501'
```

```
[INITIALIZATION  
DATA]
```

```
'CREATE STR,LTOTAL=6000,DIRLEN=557'
```

```
[INITIALIZATION  
DATA]
```

```
'CREATE MASSP,LTOTAL=9167,DIRLEN=487,LENDES=7'
```

```
[INITIALIZATION  
DATA]
```

```
'PRINT'
```

Example 2. - Data for DLG is in input stream.

```
@DLG,I PANDATA
```

```
$IPANEL IGEOM=5,$  
'UPDATE DBASE'  
$INLIPS  
  
THETOX='DBARA'  
  
$END
```

← Element PANDATA Constructed from this Data.

●.DBARA from Data Base.

```
@ASG,T 8.
```

```
@ASG,T 11.
```

```
@XQT PANEL.OPANEL
```

```
@ADD PANDATA
```

```
@USE 14,NMLIST
```

```
@fn.DLG,I MAIN
```

```
NAMELIST/OUT/C  
'UPDATE DBASE'  
CALL GETA(A)  
B = 'SMAXIS'  
C = A**2+B**2  
WRITE(14,OUT)  
END
```

← Element MAIN constructed from this data.

- GETA is an element of BRARY.
- SMAXIS from data base
- Data on 14 goes to data base. on next execution of @fn.DLG.

```
@FOR MAIN
```

```
@MAP,I DUM,MYPGM
```

```
IN MAIN
```

```
LIB BRARY
```

- BRARY contains GETA

```
END
```

```
@XQT MYPGM
```

Example 3. - Data for DLG stored in element.

```
@fn.DLG NAME1,NAME2
```

```
@XQT PGM
```

```
@ADD NAME2
```

```
NAME1 Element  
'UPDATE DBASE'  
'DATA'
```

← Element NAME2 constructed from this data.

```
@fn.DLG NAME3,NAME4
```

```
@ADD NAME4
```

```
NAME3 Element  
@XQT PGM  
'UPDATE DBASE'  
'DATA'
```

← Element NAME4 constructed from this data.

Example 4. - Linked Simulation.

```
@ASG,A DATA7          • Assign permanent data base.  
@COPY DATA7,DBASE  
@fn.DLG EDIN.SIM,MODIN  
@ADD MODIN
```

```
EDIN.SIM Element  
@XQT PGL  
  'UPDATE DBASE'  
  'DATA'  
@fn.DLG LUI.RPT,SUMRY  
@ADD SUMRY
```

← Element MODIN constructed from this data.

```
LUI.PRT Element  
SUMMARY REPORT  
  'UPDATE DBASE'  
  'DATA'  
@fn.DLG PIX,MODIN  
@T.OIMAGE  
@ADD MODIN
```

← Element SUMRY constructed from this element.

Example 5. - Generation of a data base status report.

```
BATCH MODE:  
@USE DBASE,DATA7      • Assign permanent data base.  
@fn.DLG,M EDIN.STATUS,STATUS
```

```
EDIN.STATUS Element  
STATUS REPORT  
  'UPDATE DBASE'  
  'DATA'
```

DEMAND MODE:

```
@fn.DLG,EI HISTORY
>'UPDATE DBASE'
>'NAME' • USER INPUT
value1 • DLG RESPONSE
>'ADD A=VALUE' • USER INPUT
>' B=V2' • USER INPUT
>'A ' 'B ' • USER INPUT
value2 value3 • DLG RESPONSE
> @EOF • USER INPUT
```

<u>HISTORY Element</u>
value1
value2 value3

NOTE: The character (>) is used to denote a demand read request.

Example 6. - Conditional looping by combining the capabilities of EX8 and the DLG processor.

```
@fn.DLG LOOPELT,MODIN
@ADD MODIN.
@fn.DLG LOOPELT,MODIN
@ADD MODIN
@fn.DLG LOOPELT,MODIN
@ADD MODIN
@STOP:
```

<u>LOOPELT Element</u>
'UPDATE DBASE'
@fn.DLG A,AA
@XQT PGMA
@ADD AA ←
@fn.DLG B,BB
@XQT PGMB
@ADD BB ←
@SETC 'CONVG'
@TEST TE/1
@JUMP STOP

← Element MODIN constructed from this data.

• SET CONVG

UNIVAC/ADAGE DATA TRANSFER

The Univac side of disk-to-disk operations in the form of Adage supplied AGSCRD and AGSELT leave a great deal to be desired from the viewpoints of efficiency, stability, size, ease of use, ease of understanding and generality.

Two new Univac computer programs, UTA and ATU do much towards meeting these criteria. UTA transfers unit data from the Univac computer to the Adage computer. ATU transfers units of data from the Adage to the Univac.

The absolute (executable) versions of these computer codes are found in file EX42-N00002*UR. UTA is invoked as a processor and ATU is invoked as an executable (XQT) program.

In both cases after the program is invoked on the Univac side, the escape (ESC) key is hit and the Adage returns to the AMRMX monitor in the ordinary Adage mode. At this time, either AGSCRD or AGSELT is invoked on the Adage (for UTA and ATU respectively).

Use of the Univac to Adage (UTA) Routine

The rewrite of the Univac routine AGSCRD was carefully designed to overcome major shortcomings. It is no longer necessary to modify the element being sent, nor is it necessary to type more than one Exec 8 command card. In addition, the new version will transmit images up to a length of 132 characters. Core requirements are approximately 25% of the earlier version.

To use the routine type:

```
@UR.UTA      Your-File.Your-Element
```

When the message "USER DATA MESSAGE PENDING" begins to flash on the lower left of the screen, press the escape (ESC) key. This causes the AGT to cease functioning as a terminal and returns control to the AMOS 2 monitor.

Type "AGSCRD." This will not appear on the screen while being keyed in. The Adage will then request the pack/volume (ppvv) where to place the element as it comes across. When the transfer is complete and the Adage disk is closed out, the screen will go blank. At this time, type SYM11 to return to demand mode. An example of the use of UTA is shown in figure 3-A. Note that by default, the element being sent will be displayed on the Adage screen. Transfer time may be reduced by 60% by suppressing the print. Print suppression is accomplished by the "N" option invoked as follows:

```
@UR.UTA,N    Your-File.Your Element
```

To transmit from the Univac to the Adage, an element which is more than 72 columns wide, then UTA should be invoked with the "T" (two pass) option, i.e.,

```
@UR.UTA,T      ur-file.ur-elem.
```

For an example, see figure 3-B.

Use of the Adage to Univac (ATU) Routine

The new Adage to Univac (ATU) routine has the capability of writing the image received from the Adage directly into an Univac program file as an element. Hence, no @ELT calls are necessary, and as tabbing is completed, no calls to the editor are required.

To use the routine, type:

```
@XQT UR.ATU
```

When the message "USER DATA MESSAGE PENDING" begins to flash in the lower left of the screen, press the escape (ESC) key. This causes the AGT to return control to the AMOS monitor.

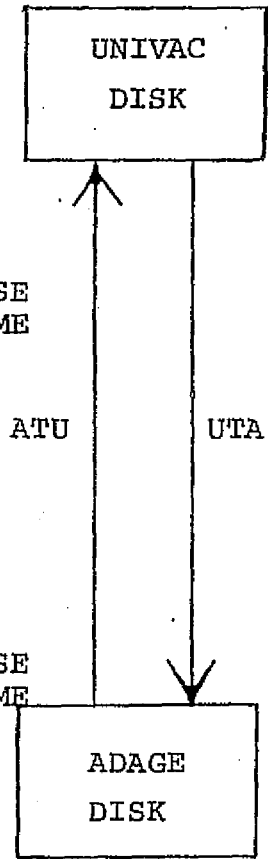
Type "AGSELT." The user should then answer the prompting question and Adage file will be transferred over to the 1110 to become an Univac element of a program file.

For an example, see figure 3-A.

DISK-DISK COMMUNICATIONS

```

@XQT UR.ATU
ESC
AGSELT
ENTER DESIRED 1100 FILE NAME (
  C/R FOR TPSS) ...
FILE NAME
**ENTER C/R TO CONTINUE ANYTHING ELSE
  TO EXIT...ENTER DESIRED PACK/VOLUME
  NO (    PVV)...
110
ENTER STARTING FILE NUMBER...
6
ENTER ENDING FILE NUMBER...
14
**ENTER C/R TO CONTINUE ANYTHING ELSE
  TO EXIT...ENTER DESIRED PACK/VOLUME
  NO (    PVV)...
A
Screen Goes Blank
SYM11
  
```



```

@UR.UTA urfile.elem
ESC
AGSCRD
ENTER DESIRED PACK/VOL NO (    PPV) ...
110
PROGRAM WILL APPEAR HERE
Screen Goes Blank
SYM11
  
```

FIGURE 3-A

FIGURE 3-B

FIGURE 3 DISK-DISK COMMUNICATIONS.

EDIN EXEC 8 PROGRAM LIBRARY

The following sections describe the EDIN Exec 8 program library.

The independent program elements, which form the library, are resident on the Exec 8 system or are readily available for incorporation should they be required. Many of the programs are nationally recognized, references 17 through 30. Others were specifically developed for the EDIN system.

The library is arranged according to the specific groups of application's software. These groups are defined in the following paragraphs. In addition, a series of abstracts are provided on the software programs available to the EDIN system.

Geometry

The geometry applications software is a collection of specialized geometry generators. Software which generates geometry or uses formatted geometry as an input is included in this classification. The following list of geometry application software and descriptions are available in the EDIN library.

- AIRFOIL: A Program for Generating Geometric and Aerodynamic Characteristics of Airfoil Sections.
- GEOMETRY: Body Coordinate Generator.
- PANEL: A Program for Generating Panelled Configuration Geometry.
- VAMP: Volume, Area and Mass Properties.
- WETED: Wetted Area and Reference Length Program.
- SPIT: A Surface Fitting Program which Generates Surfaces over Cross-Sectional Definitions. The Surfaces Generated are Defined by Cornerpoint Geometry.
- AIRCFT: Program to Generate Aircraft Type Configuration Geometry.
- GTM: Geometry Technology Module for Generation, Manipulation and Display of Cornerpoint Geometry.

Aerodynamics, Stability and Control

The Aerodynamics, Stability and Control applications software are those programs which perform configuration design analysis for predicting stability and control derivatives and aerodynamic

properties through various flight regimes. These programs and program descriptions are summarized as follows:

AIRFOIL: A Program for Generating Geometric and Aerodynamic Characteristics of Airfoil Sections.
DATCOM: Configuration Design Analysis Program (TRW).
DATCOM2: Configuration Design Analysis Program (MDAC).
HABACP: Hypersonic Arbitrary Body Aerodynamic Computer Program.
SKINF: Turbulance Skin Friction Drag Program.
TOLAND: High Lift Aerodynamics Prediction.
TREND: Subsonic/Supersonic/Hypersonic Aerodynamic Tradeoff Program.
WDRAG: Zero-Lift Wave Drag Program.
WETTED: Wetted Area and Reference Length Program.

Propulsion

The propulsion applications software would include those programs which are used for rocket engine sizing, thrust chamber design, turboramjet engine design and inlet design. These programs are summarized as follows:

ENCYCL: Design Point Performance of Turbojet and Turbofan Engine.
LREN: Liquid Rocket Engine Model.

Mass and Volumetric Properties

The Mass and Volumetric Properties applications software include those programs which perform weights trend analysis, volume and mass properties evaluations and sizing. These programs are summarized as follows:

ASPE: Weights Trend Analysis for Multiple Stage Vehicles.
CASPER: Reusable Booster/Orbiter Sizing.
CAWATA: Cost and Weight Analysis of Transport Aircraft.
ESPER: Weights Analysis for Shuttle Class Vehicle.
SSSP: Space Shuttle Synthesis Program.
VAMP: Volume, Area and Mass Properties using Harris Geometry.
VSAC: Vehicle Synthesis for High Speed Aircraft.

WAATS: Weights Analysis for Advanced Transportation Systems.

TANK: A Program for Generating Fuel and Oxidizer Tank Design Characteristics such as Volume, Wall Thickness and Weight. Cornerpoint Geometry is Generated for Display Purposes.

SIZER: A Preliminary Booster Sizing Program Based upon Ideal Velocity and Mass Ratio Relationships.

WAB: A Program that Computes Volume, Area and Mass Properties from any Combination of Gentry Geometry and Black Box Inputs.

Performance

The performance applications software includes programs which perform trajectory optimization and analysis and design synthesis. These programs are summarized as follows:

GEMASS: Three Degree/Six Degree of Freedom Motion Analysis Program.

PADS: Performance Analysis and Design Synthesis Program.

POST: A Program to Optimize Simulated Trajectories.

PRESTO: Program for Rapid Earth-to-Space Trajectory Optimization.

ROBOT: Three Degree of Freedom Launch Optimization Program.

SSFS: Space Shuttle Functional Simulator.

SVDS: Eighteen Degree of Freedom Multiple Vehicle Motion Analysis.

TOLAND: Takeoff and Landing Program.

VSAC: Vehicle Synthesis for High Speed Aircraft.

Thermodynamics

The following application software is available for thermodynamic evaluations and analysis:

HABACP: Hypersonic Arbitrary Body Aerodynamic Computer Program.

TREND: Subsonic/Supersonic/Hypersonic Aerodynamic Tradeoff Program.

Structures

The structures application software includes those programs which perform structural analysis and structural optimization of vehicles. These programs are summarized as follows:

APAS: Structural Design and Analysis Program.
AFSP: Automated Flutter Solution Procedure.
ASOP: Automated Structural Optimization Program.
BOSOR: Buckling of Shells of Revolution.
CALBAR: Buckling Loads/Cylinder Shells.
SSAM: Swept Strip Aeroelastic Model.
STAGS: Structural Analysis of General Shells.

Cost

The following application software is available for performing cost analysis:

CAWATA: Cost and Weight Analysis of Transport Aircraft.
DAPCA: Development and Production Cost of Aircraft.
PRICE: A Program for Improved Cost Estimation.

Environmental Protection

The following application software is available for predicting environmental protection requirements and environmental flight evaluations.

SBOOM: Program to Predict Ground Level Overpressure from the Over-Flight of Shuttle Type Vehicles.

General and Special Purpose Utilities

The following general and special purpose utility software are available:

PLOTTR: A General Purpose Plotter Program which Outputs to Tektronix, CALCOMP, SD4060, MOPS.
CIPHER: A Report Generation Program.
AESOP: Automated Engineering and Scientific Optimization Program.
XLT: Translator Program.

DLG: Data Processor for Linking Independent Computer Programs and Providing Data Intercommunication.

HDG: Heading Program.

MAC: Macro-Processor.

IMAGE: A Program for Displaying Three Dimensional Geometric Configurations on Tektronix, CALCOMP, SD4060, MOPS, Etc. This Program Can Perform Rotations, Translations and Zooming.

VL70: Program to Interrogate Standard Aero Tape Data.

PROFIL: Program to Interrogate Trajectory Data Tapes.

The following is a collection of abstracts on the applications and technology module software available to the EDIN Simulation System:

ABLATOR: One Dimensional Analysis of the Transient Response of Thermal Protection Systems.

The program assumes that thermal properties in the given layer of material are functions only of temperature, that all heat flow is normal to the surface, and that gases transpiring to the char layer are the same temperature as the char. The outer surface is subjected to the aerodynamic heating and the char layer provides both insulation and high temperature outer surface for reradiation. The heat passing through the layer is partially observed by pyrolysis at the interface between the char layer and the uncharred material. The remaining heat is conducted into the uncharred layer. The gases generated through the char layer are injected into the boundary layer. The gases are heated as they pass through the char and this heat removal from the char layer induces the quantity of heat conducted by the pyrolysis interface. When these gases are injected into the boundary layer, the conducted heat transfers are induced. The program accepts as input the temperature at the surface of the model and generates a time history of the thermal condition of the various layers as the vehicle enters the atmosphere.

ACMOTAN: Linear Aircraft Motion Analysis

The program is a versatile code for linear aircraft motion analysis which allows the user to supplement the standard airplane equations of motion with auxiliary equations written by the user to represent control laws or additional variables. The program prepares the system of linear differential equations

using several optional forms of input data and then carries the solution to an extent determined by the output option selected. Minimum output includes the characteristic polynomial and its roots. Additional output in the form of transfer functions, frequently responses and time histories can be selected.

AESOP: Automated Engineering and
Scientific Optimization Program.

AESOP is a multiple variable optimization program designed for the solution of a wide range of parameter optimization problems. The basic program has the ability to solve constrained optimization problems involving up to 100 parameters and up to 20 constraints. Thirteen search techniques are available for use individually or in combination to solve the desired problem. The methods include sectioning, steepest descent, quadrilateral search, Davidon's method, random ray search, pattern and several others. The program is designed to be linked with other programs to perform internal optimization or can be used as an independent program for optimizing systems of programs.

AFSP: Automated Flutter Solution Procedure

Program AFSP is based upon a new method of solving the flutter equation. The method is based upon the premise that the flutter analysis, due to the particular form in which the aerodynamic forces are available, essentially consists of a search for those V-values and w-values which render the flutter determinate zero. The procedure deviates essentially from the V-g analysis in so far that the eigenvalue calculation is replaced by simpler algorithm leading to the decomposition of the flutter matrix. The actual search for the flutter solution involves a single real and positive function of the two variables V and w rather than the (n) complex functions representing the eigenvalues in the V-g analysis. The flight altitude, Mach number and flight speed remain consistent throughout the search whereas the V-g analysis starts out from given values at altitude and Mach number. The flutter speed only follows as a result of the calculation. In general, the speed will not be consistent with the input data such that many runs are required to iterate toward a solution.

AIRFOIL: A Program for Generating Geometric and Aerodynamic Characteristics of Airfoil Sections.

The computer program is written to provide airfoil coordinates incompressible inviscid section characteristics and two-dimensional drag-rise Mach numbers for a large number of National Advisory Committee for Aeronautics (NACA) designated airfoils from a simple one card input. The program is actually a combination of two separate programs. One program gives the airfoil surface coordinates with only the NACA airfoil designation as input, and the other uses the surface coordinates to predict incompressible, inviscid pressure distribution from which the section characteristics and drag-rise Mach number are determined.

APAS: Automated Predesign of Aircraft Structures

APAS is a structural synthesis program for sizing structural elements of flight vehicles. The user may specify geometry, material type, static and fatigue loads, structural configuration, and minimum gage limits. Loading conditions may be input at up to twenty stations along the structure. Analysis routines are used to estimate failure modes such as buckling, crippling and net tension. Symmetry grouping allows identical design of adjacent panels. A two step optimization procedure is used in designing the structural elements. On the first pass, weight is minimized. On the second pass, weight is held constant while element dimensions are refined so as to maximize margins of safety. In this manner, positive margins of safety are always maintained.

ATOPII: Atmospheric Trajectory Optimization

The program is a generalized steepest descent computer program set up to handle the three-dimensional, point mass, vehicle flight path trajectory optimization problem. It is capable of simultaneously handling up to fifteen state variables, six control variables and ten constraints. Most of the usual functions required in flight path studies are available within the program; others may be added as desired by simple program additions, providing the function or its derivative is defined analytically. The program may be readily extended to cover steepest descent optimization problems in other fields, by the replacement of the basic differential equation subroutine by any other set of equations of the same general type. Convergence to the optimal solution is obtained automatically by means of one of two control systems which, by a series of logical decisions, obtain a reasonable perturbation magnitude at each iteration.

CIPHER: REPORT WRITING PROGRAM

The CIPHER program is a general purpose report writer and documentation program. CIPHER is designed to take an input stream of characters which is a mixture of both the text to be output and embedded control commands. The program processes the actual text according to the instructions given and produces a stylized report. The following capabilities are available:

- User Specified Formats.
- Upper and Lower Case.
- Greek Symbols.
- Indexing and Page Numbering.
- Cross Referencing.
- Superscripts and Subscripts.
- Footnotes.
- Equations.
- Math Symbols.
- Mixed Character Height.
- Automatic Tab Setting and Paragraphing.
- Automatic or User Controlled Paging.
- Carriage Control.

COAP: Combat Optimization and Analysis Program

The program is an extension of the ATOP II (Atmospheric Trajectory Optimization Program). It uses two complete three dimensional equations of motion sets to simulate a one-on-one combative encounter between two flight vehicles. The aerodynamic and propulsion representation are sufficiently general to permit the simulation of both current and proposed vehicles by input data. Generalized rotating planetary and atmospheric models permit stimulation of either aircraft, missiles or spacecraft encounters. Combat roles for each vehicle (attacker, defender, etc.) are automatically defined on the basis of vehicle relative positions, heading and velocities. Depending upon the vehicle role selected, any one of the set of tactics designed to satisfy the role requirement is executed. The tactics vary in nature from straight forward stylized maneuvers such as split-S to barrel role, to three-dimensional lag or lead pursuit path. Combat optimization capability may be introduced by repetitive simulation using parameterization of the combat guidance parameters and the application of multivariable search techniques. Alternatively, the variational calculus may be employed to define optimal

continuous control against a reacting opponent. In the parameter optimization mode, the option to determine a mini-max solution is also available.

CONPLOT: Aircraft Configuration Plot

This program generates the necessary instructions for automatically plotting of a numerical model of an aircraft configuration. Program options may be used to draw three-view and oblique orthographic projections as well as perspective drawings of an aircraft. These plots are useful in checking the accuracy of the numerical model data.

COT PROCESSOR

Due to the manner in which images are stored upon mass storage, the presence and sequence numbers in columns 73-80 significantly increases the physical amount of storage required. Blanking these columns is a difficult and enormous task via the ED processor, therefore, a small processor called COT has been developed for this purpose.

This processor and its two associated SSG run streams can do this blanking quickly and conveniently. The processor has two modes of operation of which one and only one must be specified.

CUSSER: Upper Stage Sizing Program

The computer program determines the size, weight and cost of upper stages for either specified payloads or gross weights. The program is capable of sizing stages for missions having "N" burns, with each having a different payload. It also has the capability to size the stage for use in conjunction with a specified booster or launch vehicle.

DAPCA: Development and Production Cost of Aircraft

The DAPCA Program computes development and production costs of major subsystems of fly-away aircraft (airframe, engines, etc.). The cost input data is simple and generally relates to aircraft and engine performance characteristics such as gross takeoff weight, speed engine type thrust, etc. The actual cost equations are the power law types with builtin cost coefficient and using user supplied parameters.

DATCOM: Configuration Design Analysis Program (TRW)

The program computes aerodynamic coefficients for aircraft/spacecraft configurations in the subsonic/transonic/supersonic regimes. Analytical techniques in the program are based on those of USAF stability and control handbook, DATCOM, revision September, 1970.

The program comprises four modules which compute lift, pitch, sideslip and control characteristics respectively. Modular construction enables other sets of aerodynamic characteristics to be incorporated into the program.

DATCOM2: Configuration Design Analysis Program (MDAC)

The program calculates the static stability characteristics of wings, bodies, wing-body, tail-body and wing-body-tail combinations at angle of attack and sideslip through the Mach range from subsonic to supersonic speeds. Whenever appropriate DATCOM methods are available, the program computes longitudinal derivatives C_{L_α} and C_{M_α} , longitudinal coefficient C_D , C_L , C_m and sideslip derivatives C_{Y_β} , C_{l_β} and C_{n_β} . Output for configurations of horizontal tails also include downwash and dynamic pressure values. All intermediate variable calculations are also available for output.

ENCYCL: Design Point Performance of Turbojet and Turbofan Engine

ENCYCL computes the design point performance of turbojet and turbofan engine cycles from user supplied engine characteristics and flight conditions. The program input requires the airplane Mach number, the altitude, the state conditions, turbine inlet temperature, afterburner temperature, duct burner temperature, bypass ratio, coolant flow, component efficiencies and component pressure ratios. The output yields specific thrust and specific fuel consumption, engine efficiencies and several component temperatures and pressures. The thermodynamic properties of the gas are expressed as functions of the temperature and fuel to air ratio.

FLOWGEN: Automatic Flowchart Program

FLOWGEN produces flow charts of Fortran language subroutines. Input to the program is namelist.

GENENG: A Program for Calculating Design and Off-Design Performance for Turbojet and Turbofan Engines

The program calculates steady state design and off-design performance for one and two spool turbojet engines. The original version of the GENENG program entitled, "Simulation of Turbofan Engines" was developed by the Turboengine Division of the Air Force Aero Propulsion Laboratory, Wright Patterson Air Force Base, Ohio. The program uses steady state gas dynamics to compute the engine design conditions. Off-design performance is

based on specific component performance maps which must be provided by the user.

GENENGII: A Program for Calculating Design and Off-Design Performance of Two and Three Spool Turbofans with as many as Three Nozzles.

The GENENGII Program is a derivative of GENENG (Generalized Engine Program). GENENG is capable of calculating steady state design and off-design performance of turbogfan and turbojet engines were evolved from SMOTE (Simulation of Turbofan Engine) which was developed by the Turbo Engine Division of the Air Force Aero Propulsion Laboratory of Wright Patterson Air Force Base, Ohio. GENENGII calculates design and off-design engine performance for existing or theoretical fan engines with two or three spools and with one, two or three nozzles. In addition, fan performance can also be calculated. Nine basic turbofan engines can be calculated without any programming changes. Included among the nine are three types which are likely candidates for STOL aircraft with internally blown flaps. Many other possibilities exist which are too numerous to mention, being determined by the users knowledge of the program itself.

GEOMETRY: Body Coordinate Generator

The program generates trapezoidal and elliptical body coordinates in a format suitable for use in VAMP, WETTED, DRAG and CONPLOT. The forbody coordinates are generated according to a "minimum drag" area distribution while the afterbody coordinates are constant in cross sectional area.

GTM: Geometry Technology Module

The Geometry Technology Module (GTM) is a system of computerized elements residing in the EDIN (Engineering Design and Integration) System library developed for the generation, manipulation, display, computation of mass properties and data base management of panelled geometry. The GTM is composed of computer programs and associated data for performing configuration analysis on geometric shapes. The program can be operated in batch or demand mode and is designed for interactive use. The significant features of the program are:

1. Data bases containing two and three dimensional shapes including standardized shapes generated by the GTM.
2. An executive computer program containing a user orientated language for controlling the generation, display and calculation of mass properties on selected vehicle components.

3. An auxiliary computer program and data base for the construction and storage of language elements, menus, user instructions and messages.
4. A library of independent geometry generation programs for the creation of specialized geometric panelling.

HABACP: Hypersonic Arbitrary Body
Aerodynamic Computer Program.

The program treats the vehicle surface as a collection of quadrilateral elements oriented tangential to the local vehicle surface. Each individual panel may have its local pressure coefficient specified by any of a variety of pressure calculation methods, including modified Newtonian, blunt body, Newtonian-Prandtl-Meyer, tangent-wedge, tangent-cone, boundary layer induced pressures, free molecular flow and a number of empirical relationships. Viscous forces may also be calculated, which include viscous-inviscid interaction effects. Skin friction options include the reference temperature and referenced enthalpy methods for both laminar and turbulent flow, the Spalding-Chi method and a special blunt body skin friction method. Control surface deflection pressures including separation effects that may be caused by the deflected surface are also calculated. Several other options are available including the calculation of dynamic derivatives, the generation of geometry and plotting.

HDG: Heading Program.

HEADING is a control card callable program which prints user specified heading information in large characters. The characters of the heading are formed by the pattern of characters which form the character itself. The letters of the heading are eight characters wide and ten characters high. Input to the program is placed on the heading execution card.

IMAGE: Configuration Display Program.

The IMAGE program uses a surface definition based on quadrilateral elements to describe picture-like drawings of arbitrary configurations. The program is used for visual check on geometric input data, monitoring of geometric perturbations and providing reports on geometric characteristics. Geometric characteristics may be input or taken from a data base of configuration data. The user describes the viewing angles, position and scaling factors as well

as textual information through the input procedure. The configuration drawings are generated on a plot vector file which is suitable for processing by various display devices.

MINIVER: Aerodynamic Heating Program.

The program is a "miniature version" of the McDonnell Douglas Corporation Aerodynamic Heating Program for use on the CDC 6600. It calculates radiation equilibrium temperature and provides thin-skin temperature response for input materials. (Aluminium, titanium, Rene 41 and inconel X-750 properties are built in.) Heat transfer methods are available including Fay and Rydell, Erkart Reference Enthalpy, Spaulding and Chi, Flat Plate Rho-Mu product, three swept cylinder theories and Lees-Detra and Hidalgo for hemispheric nose caps. The program also includes the capability of traversing three sequential shocks, shape-edge and cone plus oblique shock of availability, sharp-edge and cone modified Newtonian and swept cylinder stagnation line pressure solution.

PADS: Performance Analysis and Design Synthesis Program.

The Performance Analysis and Design Synthesis (PADS) computer program has a two-fold purpose. It can size launch vehicles in conjunction with calculus of variations optimal trajectories and can also be used as a general purpose branched trajectory optimization program. In the former case, it has the Space Shuttle Synthesis Program as well as a simplified stage weight module for optimally sizing manned recoverable launch vehicles. For trajectory optimization alone or with sizing, PADS has two trajectory modules. The first trajectory module uses the method of steepest descent, the second employs the method of quasilinearization, which requires a starting solution from the first trajectory module.

PANEL: A Program for Generating Panelled Configuration Geometry.

The PANEL Program is a general purpose external geometry definition program developed primarily for use in large scale simulations of the preliminary design process. It is an independently operated program which produces a sequence of quadrilateral panels defined by the corner points. The resulting data is acceptable as input to computer programs in other technical disciplines such as aerodynamics, structures and thermodynamics.

The program accepts as input a variety of formats from detailed definition of individual panels to selection of generalized two- and-three dimensional shapes. The section data includes circular

and elliptical as well as arbitrary cross sections. A cubic patch technique is included which allows broad sections of the vehicle to be described with a relatively small amount of input. The input data can be mathematically fitted with end matched cubic functions and reduced to distributed panels.

PLTVIEW: Program for Generating
Separation Plots.

The program is a companion to the SEPARTE Program which computes the separation distance and attitude of two stages of the shuttle vehicle during staging. PLTVIEW generates plots of separation distance and vehicle orientation for the separating vehicle components. The input data is obtained from the time history data generated by the SEPARTE Program. The output is a GERBER plot of the separation distance and orientation.

POST: A Program to Optimize
Simulated Trajectories.

The POST Program is a generalized point mass discrete parameter targeting an optimization program. POST provides the capability to target and optimize point mass trajectories for a powered or unpowered vehicle near an arbitrary rotating oblate planet. POST has been successfully used in solving a wide variety of atmospheric ascent and reentry problems as well as exoatmospheric orbital transfer problems. The generality of the program is evidenced by its N-phase simulation capability which features generalized planet and vehicle models. This flexible simulation capability is augmented by an efficient, discrete parameter optimization capability which includes equality and inequality constraints.

PRESTO: Program for Rapid Earth-to-
Space Trajectory Optimization.

The PRESTO Program uses a closed loop deepest descent optimization procedure to derive flight trajectories to produce maximum booster payloads for a variety of space missions. Trajectories can be computed in three degrees of freedom about a spherical rotating earth. Four powered stages and three upper stage thrust cycles can be accommodated. Coast periods are permitted between each stage. Aerodynamic lift and drag forces are included in the computation. The optimization routine simultaneously considers the launch direction and time the interstage coast durations and the upper stage thrust sequencing, the complete pitch and yaw attitude histories and terminal constraints. Immediate constraints may be introduced on angle attacks, coast orbit perigee altitude or on the product on angle attack and dynamic pressure. The closed loop procedure greatly facilitates the satisfaction of terminal constraints and reduces the number of iterations required to achieve convergence.

PRICE: A Program for Improved Cost Estimation.

The PRICE computer program was developed in order to rapidly generate preliminary estimates of total program costs for mission studies of V-STOL and conventional transport aircraft, hypersonic aircraft and reusable space transportation system. The program uses cost estimating relationships based of historical cost data for conventional and advanced aircraft, spacecraft and launch vehicles. The approach is based on correlating cost with parameters such as weight and thrust, then adjusting the results with complexity factors to account for differences in material and type of construction, performance level, etc.

**RDPRO: POST Plot Data
Generation Program.**

The program interrogates the output data tape from the POST program and generates specified plot information pertaining to time histories of various performance and constraint functions available after the execution of the POST Program. The program is designed specifically as an interface program for analysis of trajectory data from the POST Program.

REPORT: Report Generator.

REPORT is an executive program (DLG) option which allows the user to interrogate the data base and format engineering reports. Descriptive information may be provided in any format with delimited data base names inserted where data base requests are desired. The report is automatically printed with descriptive information printed exactly as originally specified and delimited data base names replaced by the corresponding data base information.

**SBOOM: Sonic Boom Prediction for
Shuttle Type Vehicles.**

The SBOOM Program is based on the prediction technique of Thomas (references). The prediction technique calculates the far-field overpressure from the near-field pressure signatures measured in the wind tunnel. The wind tunnel results generally are stored in a data base and accessed by the computer program for interpolation/extrapolation based on geometric similarity of the pressure signatures. Wind tunnel data in the data base was generated for space shuttle type configurations. The approach used in determining ground overpressure is to describe the wave form of the sonic boom wave by several wave form parameters and then obtaining equations for the parameters as function time. This approach has the advantage of being simple and providing a convenient method for extrapolating experimental signatures

because the signatures are dealt with directly. The input consists of the vehicle conditions and the environment in which the vehicle is operating as a basic condition operating the program. Many flight conditions can be equated through the normal input channels or the program will accept flight condition data from trajectory generation programs stored on an auxiliary file.

SEPARTE: The Program to Simulate Separating Stages of Launch Vehicles.

The program is a twelve-degree-of-freedom separation program for studying space shuttle separation problems. Each stage of the separating vehicle is represented by six degrees of freedom. Aerodynamic data governing the motion of the separating vehicle is staged into the program from a storage file because of the large bulk of data. The program generates Gerber plots of the separating vehicle components at specified time intervals. Separating components are represented by simplified geometric shapes.

SFIT: A Surface Fitting Program

The SFIT program defines surfaces for objects that have a plane of symmetry. It was developed for the purpose of generating and/or modifying fuselage type surfaces. A parametric cubic patch method is used to define the surface over a set of cross sections that the user in the form of corner point geometry.

SIZER: A Preliminary Sizing Program for Launch Vehicles.

The SIZER program performs preliminary sizing for a launch vehicle composed of up to ten stages. User inputs of payload, specific impulse of each stage, ideal velocity of each stage and mass fraction of each stage are required. Program results are based on the ideal velocity equation,

$$V = g \cdot I_{sp} \cdot \ln(MR),$$

where MR is the ratio of the mass of the vehicle at the start of the stage to the mass of the vehicle at the end of the stage, I_{sp} is the specific impulse and g is gravitational acceleration. It should be restated that SIZER gives a preliminary sizing estimation based upon ideal velocity relationships; no drag losses or gravity losses are included. SIZER outputs a summary which presents the sizing information for each stage.

SKINF: Turbulance Skin Friction Drag Program.

The program computes the skin friction drag of a vehicle including the effects of distributed roughness and temperature of the surfaces at arbitrary combinations of Mach number and altitude. Calculations can be made using either the standard day or the +10 degrees hot day atmospheres. Input consists of flight conditions (Mach-altitude), the wetted areas, reference length and form factors for all of the components of the aircraft in the mean roughness height and emittance of the surfaces. Wetted areas in reference lengths may be obtained from the program WETED.

SSAM: Swept Strip Aeroelastic Model.

The program performs an aeroelastic evaluation of the wing spanwise flight load analysis including the complete aircraft balance for a specified set of steady state maneuvers and/or design lift conditions. Here, proper inclusion of the wing-body and the nacelle aerodynamic and weight effects are included in order to compute the required balance tail load which is reflected in the wing load calculation. The flight loads, including the aerodynamic and wing dead weight loads, are converted into structural wing box bending and torsion loads to evaluate the resulting bending and torsional stresses. If the calculated wing stress exceeds the allowable wing stresses, a new set of values of wing section stiffness values are selected to match the allowable stress distribution specified within the program data. The wing aeroelastic load solution is then repeated until the calculated and allowable wing stresses are matched. The cycling process is fast and usually requires three to five cycles to converge depending upon the error margin set within the program. The final calculation is the wing box weight based on the final set of EI and GJ values obtained. The analysis is limited to subsonic flight conditions.

SSSP: Space Shuttle Synthesis Program.

The program automates the trajectory weights and performance computation essential to predesign of the space shuttle system for earth-to-orbit operation. The two-stage space shuttle system is a completely reusable space transportation system, consisting of a booster and an orbiter element. The SSSP major subprograms are detailed weights/volume routine, a precision three dimensional trajectory simulation and the iteration and synthesis logic necessary to satisfy the hardware and trajectory constraints. Three versions of SSSP are available representing some early space shuttle concepts in the predesign stage of the shuttle project.

TOLAND: Take-Off and Landing Program.

The program was constructed by NASA's Advanced Concepts and Mission Division, OART. The program provides simplified high lift aerodynamics based on DATCOM methods, a ground roll analysis, rotation logic and climbout to clear a fifty-foot obstacle. Angle-of-attack in the ground run and rotation maneuvers are determined from the vehicle geometry which is input to the program.

TOP: Trajectory Optimization Program.

A steepest-ascent optimization program has been developed which is capable of optimizing the flight path of a wide class of vehicles. The program will optimize rockets, air-augmented rockets, ramjets, scramjets, turbojets and glide vehicles. Unique features are incorporated which allow extension of optimization procedures into the airbreathing propulsion field, particularly to supersonic transport type vehicles.

The body pitch angle is used for in-plane trajectory shaping in place of the usual angle-of-attack control variable. Additional control variables include bank angle, engine throttling and a variable geometry control variable for vehicles with variable sweep wings, drag brakes, etc. Enroute placards are available to allow optimization within realistic constraints, such as control limits, structural loads, engine operating limits, manned vehicle requirements and geopolitical limitations such as sonic boom.

The optimization is accomplished with an automatic step-size controller and with automatic control variable weighting matrices to allow problem solution in a single computer run. Automatic plotting capability is included. Multistaged vehicles and problems involving variable initial conditions may be optimized. A variable step Runge-Kutta integrator performs the derivative evaluations.

TOPLOT: Plot Generator for TOP.

This program interrogates the output file from TOP (Boeing Trajectory Optimization Program) and generates time history plots of various performance and constraints functions generated during a top run. The output is placed on temporary disk and can be transferred to a physical tape for plotting using the PLOTSV procedure.

TREND: Subsonic/Supersonic/Hypersonic
Aerodynamic Trade-Off Program.

The program provides rapid aerodynamic lift, drag and moment estimates and the subsonic, supersonic and hypersonic lift regimes. It is primarily designed to estimate high lift/drag re-entry vehicle aerodynamic characteristic, but the class of vehicles which may be analyzed by the program is of greater range than the primary class of vehicle. Some program modification may be required if the extension is too great. The program may be used for generating basic aerodynamic coefficients or it may be used for trending from known aerodynamic characteristics based on theoretical changes in the geometry. In the hypersonic flight regime, the program contains an optional aerodynamic heating computation capability. The program does not possess a transonic aerodynamic characteristic estimation capability.

VAMP: Volume, Area and Mass Properties.

The VAMP computer program calculates the mass properties, c.g. location, enclosed volume, wetted area and planform area of any closed structure that has a plane of symmetry. The vehicle is described to the computer program by ordered sets of X, Y, Z coordinates of points on its surface. The X, Y, Z coordinates are converted to quadrilateral elements for analysis. The mass properties of each quadrilateral may be computed from a thickness and density input for each quadrilateral or from a weight per unit area input at each point or from a combination of both.

The computed mass property totals may contain not only the contribution from the distributed mass on the vehicle surface wall, but additional masses may be added as "black boxes" by specifying each one's c.g. location and mass properties.

Program VAMP can also produce picture-like images of the vehicle or individual sections from the quadrilateral element data. This facilitates checking the input and visualizing and/or modified configurations. Orthographic, perspective and stereo views may be obtained.

VL70: A Program for Reading Aero-
dynamic Data Tapes.

VL70 is a general purpose utility program designed to extract and reformat specific incremental aerodynamic information from standard aerodynamic data tapes. Input to the program is name-list and the program can be executed in batch or demand. Output of the program consists of formatted data which can be easily interfaced to existing EDIN application software.

VSAC: Vehicle Synthesis for
High Speed Aircraft.

The VSAC program was developed for use in preliminary sizing studies of high speed flight vehicles. Single-stage aircraft with either takeoff and landing capability or airborne start, and two-stage vehicles consisting of an aircraft with an expendable, rocket-propelled booster can be sized. The single-stage aircraft are capable of either rocket or airbreathing propulsion and flight speeds to Mach 6. Two-stage configurations cover flight regimes to Mach 15, although no supersonic-combustion engines were considered.

Mission modeling is provided in a modular fashion to allow the program user maximum flexibility, and to allow improvement or substitution of subroutines as new studies may require. Analytic performance equations and numerical trajectory integration are provided, together with appropriate iteration routines.

The primary aerodynamics method is a component buildup procedure that requires the user to input only a geometrical description of the external vehicle characteristics and generates curve-fit expressions for the lift and drag coefficients. Stability and control are not considered.

WAATS: Weights Analysis for
Advanced Transportation Systems.

WAATS is a weights analysis program which uses the component buildup technique for weight estimation. Each component weight is based on the weight of the same component of similar vehicles that have actually been built or at least designed in great detail. The similarity law that gives the best correlation for most subsystems has been shown to be the power law formula:

$$y = a x^b.$$

The program logic assumes the propellant weight and physical characteristics are known. It performs the weight estimations with user supplied correlation parameters (a and b), estimated gross weight and estimated landing weight. An internal iteration loop cycles through the equations until convergence on weight is achieved.

WAB: A Program for Computing Weights and Balances.

The WAB program computes the volume, area and mass properties of a structure. The structure can be defined as a surface form, a set of black box components or a combination of both. Surface form definition is in the form of corner point geometry. WAB outputs include a mass property summary and EDIN data base additions.

WDRAG: Zero-Lift Wave Drag Program.

The program calculates the zero-lift wave drag at supersonic speeds for airplanes having arbitrary combinations of fuselage, wing, nacelle, horizontal fins and vertical tail. The input geometry description is in the format of CONPLOT and WETTED programs. The program may also be used to calculate (for any desired Mach number) the fuselage area distributions that are required for a minimum wave drag. Simplified fuselage constraints may be included for the above calculation. The fuselage may be cambered and may have arbitrary cross-sectional shapes. The wing may be twisted and cambered. The theoretical approach is based on the far-field linear theory that considered the relationship between the forces on the airplane and the momentum transported through the boundaries of a control surface completely surrounding the airplane.

WETTED: Wetted Area in Reference Length Program.

This program computes the surface wetted area and reference length for all components of a configuration based on corner-point geometry inputs to the program. The configuration may be arbitrarily divided into components on the fuselage and all aerodynamic surfaces. The wing may be divided into many stream-wise panels in order to approximate a strip integration for skin friction calculations. Input to the program is identical to CONPLOT. Output is suitable for use in the SKINF Program.

XLT: Translator Program

XLT is a program which translates binary data. The program operates in a conversational mode and is used primarily for translation of plot data.

EDIN ADAGE GS340 PROGRAM LIBRARY

The following sections describe the EDIN Adage GS340 Program Library. The independent program elements which form the library, are resident on the Adage or are readily available for incorporation should they be required.

The following is a collection of abstracts on the Adage applications software available to the EDIN Simulation System.

IMP: Interactive Mass Properties

The mathematical modeling of the IMP program permits the calculation of the mass properties, c.g. locations, enclosed volume, wetted area and planform area of any closed structure that has a plane of symmetry, c.g. fuselage, stiffened fuel tank, etc. The vehicle is described in a symmetrical manner and consists of sets of X, Y Z coordinates on its surface. The surface coordinates are converted to quadrilateral elements but since the four corners of the quadrilaterals are not necessarily coplanar, each quadrilateral is analyzed as two triangles. The mass properties are computed from a thickness and density input for each quadrilateral, from a weight per unit area input at each point, or from a combination of both. The weight per unit area can be a composite of the real wall including skin, insulation, ribs, stringers, standoffs, brackets, etc.

The elemental mass properties are accumulated for each section and for the total vehicle. The computed mass property totals containing the contribution from the distributed mass in the vehicle surface wall may be combined with additional masses specifying each one's c.g. location and mass properties about its c.g. The added black boxes may be inside or outside the surface and do not have to be symmetrical.

IMAGE

The IMAGE Program on the Adage 340 Graphics System generates pictorial representations of configurations defined by three-dimensional corner-point geometry. The program allows the user to view the display from variable viewing angles with variable scaling. IMAGE also provides windowing and zooming on a region of interest. Interactive options in the program allow the user to delete components from the display list, move components, duplicate components and rotate components. Interaction with the program is handled through the Adage alphanumeric keyboard and the function switch control box.

FUSEGEN

The FUSEGEN Program generates corner-point geometry for fuselage shapes. The interactive capabilities of the Adage 340 Graphics System are used to define arbitrary cross section definitions or sideview/topview silhouettes which determine the fuselage shape. FUSEGEN can also be used to modify existing fuselage corner-point geometry. The program accepts inputs from the Adage keyboard, disk, function switches and digital data tablet. Output data is stored on disk for use by other applications software.

PARAGEOM: Parametric Geometry Program.

The interactive parametric wing geometry program is now operational as part of the Engineering Design and Integration (EDIN) System on the Adage 330 graphics computer.

The primary purpose of the parametric wing geometry program is to enable the user to interactively modify or redesign a given wing planform by changing or specifying such parametric characteristics as aspect ratio, taper ratio, sweep angles, etc.

The capabilities of the program are illustrated in figure 1, which is a representation of the displays generated on the Adage screen by the program. The user interacts with this display to create and modify wing planform geometry. Information displayed on the screen includes a table of the parametric characteristics of both the original and current wings, a graphical representation of the exposed semiplanforms of both the original and current wings, a list of program control options including the retrieval of the original wing from the data base and storing the current wing in the data base. The device for altering wing data is a grid containing alpha numeric characters which can be picked by the light pen.

WINGEN: Wing Generation

"WING" is one of a series of computer codes for use on the AGT340 that enables the engineer to create planform designs of a wing starting with nothing except the conception of how it should look. From this interactive design the wing description and 3-D picture is created by the application of only a few key strokes. After the basic planform is created, the result may be continually modified and reshaped via the interactive devices until the engineer is satisfied. Standard NACA airfoils of virtually any specification may be added at any time. Continuously during the modification, various wing parameters such as theoretical area, taper ratio, aspect ratio, span, etc. are

calculated and displayed in real time. Upon exit the parameter description is stored, a "corner-point" description is created, then displayed and finally stored for possible later inclusion in a grouping of other components to a larger drawing.

SAMPLE ANALYSIS

The use of the EDIN system for a design analysis task involves the execution of four major tasks:

1. Define the analysis tasks including the technology areas to be considered, the technological depth and the study parameters.
2. Select the technology programs from the EDIN library and establish the sequence of executions including sizing and matching loops necessary to accomplish the analysis.
3. Establish the data intercommunication requirements of the program including both the fixed input data and the intercommunication data.
4. Generate the run stream required to perform the analysis.

Definition of Analysis Tasks

Figure 4 illustrates the definition of a group of analysis tasks involving the simple sizing and performance evaluation of a parameter series of all expendable launch vehicles. The study parameters are:

Payload Weight
Number of Stages
Fuel Type
Oxidizer Type
Tank Diameter
Thrust-to-Weight Ratio
Engine Chamber Pressure
Number of Engines

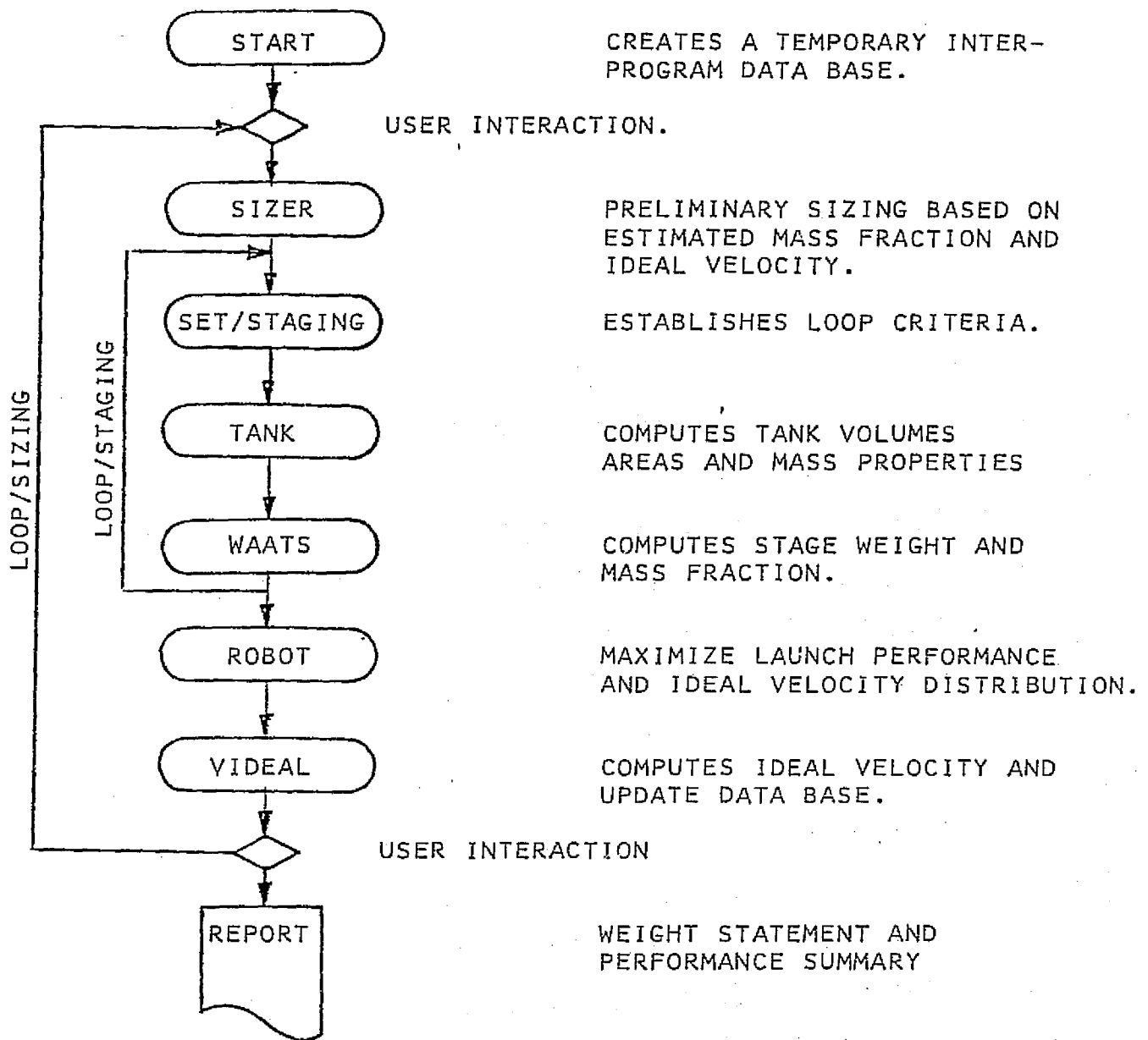


FIGURE 4 LAUNCH PERFORMANCE AND SIZING STUDY.

Selection of Analysis Programs

The analysis is to be performed using the following EDIN programs:

SIZER	Preliminary Sizing Program which Estimates the Propellant Weights and Inert Weights of all Stages based upon the Input Stage Mass Fraction and Specific Impulse.
TANK	Program to Compute the Fuel and Oxidizer Tank Weights and Geometry based upon the Propellant Quantity, Mixture Ratio, Structural Factors, etc.
WAATS	Weights Analysis Program which Computes all Sub-system Weight based upon Historical Weight Estimating Relations.
ROBOT	Launch Performance Program which Optimizes the Control History and Stage Time.

Intercommunication Data

The following table describes the program intercommunication requirements for the EDIN programs selected. Fixed input requirements for the program are discussed earlier.

<u>INPUT</u>	<u>OUTPUT</u>
<u>SIZER.</u> -	
PAYLD - Payload	WSB(I) - Start Burn Weight
NSTAGE - Number of Stages	WEB(I) - End Burn Weight
XLAMDP(I) - Mass Fraction	WPR(I) - Propellant Weight
VIDEAL(I) - Ideal Velocity	WI(I) - Inert Weight
XISP(I) - Specific Impulse	WPL(I) - Stage Payload
WPEST(I) - Propellant Estimate	
<u>TANK.</u> -	
DTANK(I) - Diameter of Tanks	Tank Geometry (File 8)
WPR(I) - Propellant Wt.	LT - Length(S)
	VOL - Volume (s)
	SA - Area(s)
	WP - Propellant Wt(s)
	WTK - Tank Weights
	WAC - Accessory Weights
	WTOT - Total Weights

INPUT

OUTPUT

WAATS. -

ENG(I) - Number of Engines
THR(I) - Thrust
WTKG - Tank Weight
SBODY - Body Area

WCOMP(75) - Dry Weight
WCOMP(72) - Gross Weight
WPARM(S) - Mass Fraction

ROBOT. -

WSBL - Stage Gross Weights
WPL(I) - Payload
XISP - Specific Impulse
THR(I) - Thrust

RBPORT - Stage Report

VIDEAL. -

RBPORT - Stage Report
XISP - Specific Impulse
NSTAGE - Number of Stages

VIDEAL - IDEAL Velocity

Generation of a Run Stream

Each box in figure 5 represents a sequence of tasks. The executive control statements and data requirements to accomplish the tasks are stored as partial run stream elements in the data base file EDIN-WAATS. The linkage of the elements to perform the analysis is shown in figure

The elements will be preprocessed by the DLG processor as the analysis proceeds to satisfy data base requests contained therein. DLG preprocessing produces executable partial run streams which are merged with the run stream using the @ADD control statement.

The sizing analysis is initiated by the control statement:

@ADD EDIN-WAATS.START

The start task sequence shown below performs the necessary file assignments and constructs a temporary intercommunication data base DBASE.

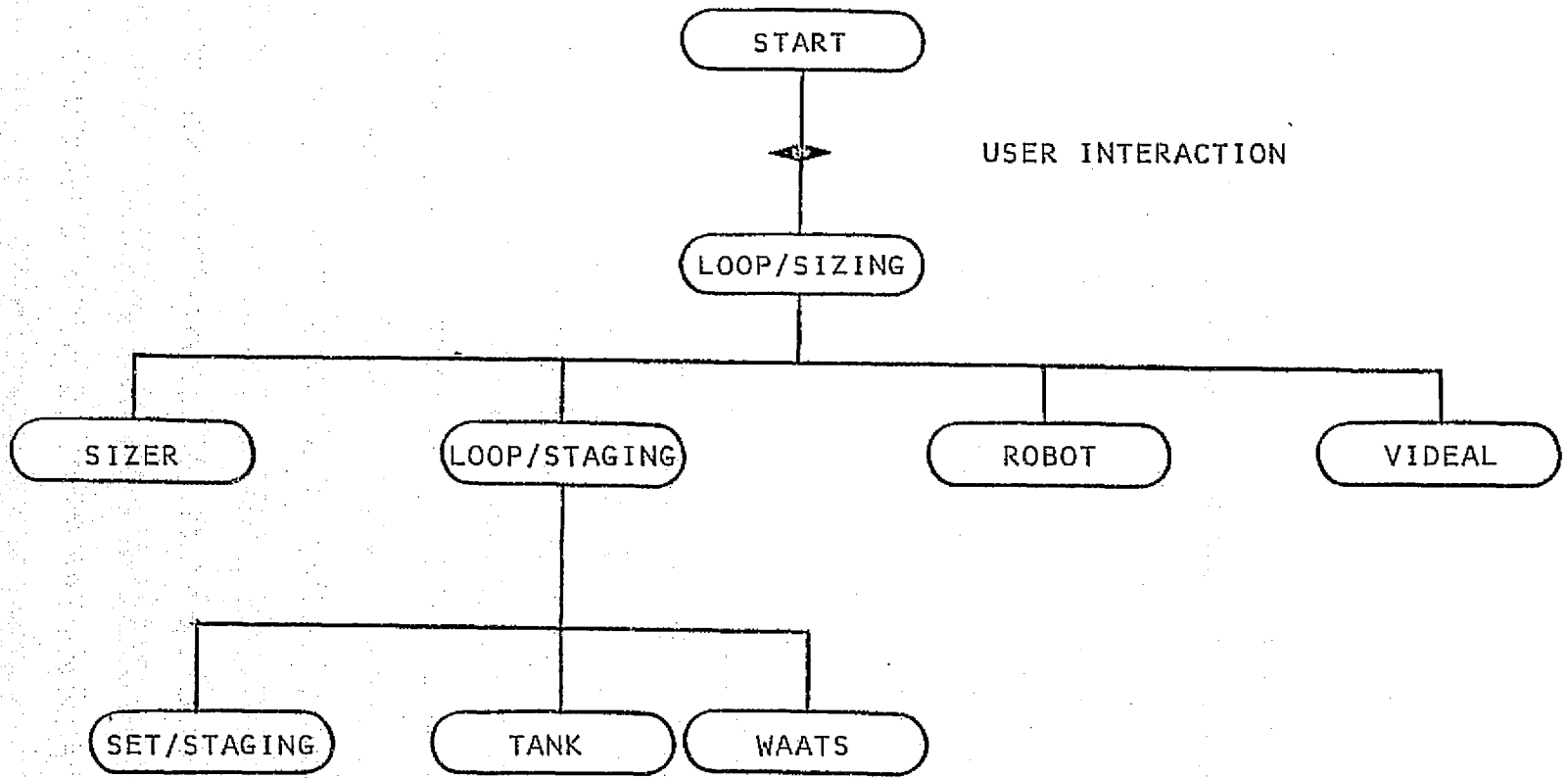


FIGURE 5 PARTIAL RUN STREAM LINKAGE.

ORIGINAL PAGE IS
OF POOR QUALITY

```
      ' . OUTPUT FROM SIZER'  
' DEFINE WSB=4, STAGE START BURN WTS,'  
' DEFINE WEB=4, STAGE END BURN WTS'  
' DEFINE WPL=4, STAGE PAYLOADS,'  
' DEFINE WPR=4, STAGE PROP WTS,'  
' DEFINE WI=4, STAGE INERT WTS,'  
' ADD WSB=0.,0.,0.,0.,  
      WEB=0.,0.,0.,0.,  
      WPL=0.,0.,0.,0.,  
      WPR=0.,0.,0.,0.,  
      WI=0.,0.,0.,0.,  
' END  
      ' . OUTPUT FROM WAATS'  
' DEFINE WCOMP=77, COMPONENT WTS,'  
' DEFINE WPARAM=5, WT PARAMETERS,'  
' ADD WCOMP=77*0.,'
```

STUDY PARAMETERS

```
' DEFINE PAYLD, LEO PAYLOAD,'  
' DEFINE NSTAGE, NO OF STAGES,'  
' DEFINE VIDEAL=4, STAGE VELOCITIES,'  
' DEFINE XLAMP=4, STAGE MASS FRACTIONS,'  
' DEFINE XISP=4, STAGE SPECIFIC IMPULSE,'  
' DEFINE WPEST=4, STAGE PROP ESTIMATE,'  
' DEFINE S, CURRENT STAGE NO,'  
' DEFINE DTANK=4, TANK DIAMETERS,'  
' DEFINE TWR=4, STAGE THRUST/WEIGHT,'  
' DEFINE PC=4, STAGE CHAMBER PRESSURE,'  
' DEFINE ENG=4, NUMBER OF ENGINES,'  
' DEFINE AREOT=4, EXIT AREA PER LB,'  
' DEFINE TAUT=15, STG BURN TIME,'  
' DEFINE TTBL=31, ATT HISTORY TIME TAB,'  
' DEFINE CPTBL=31, CHI PITCH TABLE,'  
' DEFINE CYTBL=31, CHI YAW TABLE,'  
' DEFINE CHIDT=1, TILT RATE,'  
' DEFINE ELB=4, STAGE LENGTHS,'  
      ' . INITIALIZATION DATA,'  
' DEFINE RREPORT=44, RREPORT REPORT,'  
' ADD PAYLD=60000.,  
      NSTAGE=2,  
      S=0,  
      VIDEAL=10152.,19866.,0.,0.,  
      XISP=319.,455.3,455.3,455.3,  
      WPEST=3945000.,1033300.,0.,0.,  
      XLAMP=.935,.935,.935,.935,  
      DTANK=300.,300.,300.,300.,  
      TWR=1.36,1.46,0.,0.,  
      PC=1000.,1000.,0.,0.,  
      ELB=864.,276.,0.,0.,
```

```
ENG=5.,2.,1.,1.,
AECT=.0000000001,.0000000001,.0000000001,0.0000000001,
TAUT=133.,375.15,0.,2.,11*0.,
REPORT=44*0.,
TTBL=133.,415.,531.2,28*2000.,
CPTBL=.92436,1.331,1.654,28*2.,
CYTBL=31*0.,
CHIDT=.044,
```

ORIGINAL PAGE IS
OF POOR QUALITY

'END

Following the initial creation of DBASE, there is a "user interaction" point. When operating in the demand mode, the user interaction is an interrupt in the execution sequence. The user has the option of interacting with the data base, to make parametric modifications or to construct a summary report. Requested data base information will be placed on the element, REPORT residing in the temporary program file TPF\$.

The analysis is restarted with the executive control statement:

```
@ADD E.LOOP/SIZING
```

The element LOOP/SIZING shown below contains the control statements and data required to perform the sizing loop shown in figure 4.

```
@D.DLG E.SIZER,SIZER
@ADD SIZER
@BRKPT PRINT$/BK
@BRKPT PRINT$
@D.DLG,I REPORT
  'UPDATE DBASE'
  'PROCESS NMLIST'
  'WSB'
  'WEB'
  'WPR'
  'WI'
@ADD E.LOOP/STAGING
@ADD E.LOOP/STAGING
@ADD E.LOOP/STAGING
@ADD E.LOOP/STAGING
@ENDSTG:
@D.DLG E.ROBOT,ROBOT
@ADD ROBOT
@D.DLG E.VIDEAL,VIDEAL
@ADD VIDEAL
```

```

@D.DLG,10 REPORT
'UPDATE DBASE'
'PROCESS NMLIST'
'ADD S=0'
'UIDEAL'
'XLAMP'
@EOF

```

ORIGINAL PAGE IS
OF POOR QUALITY

The sizing loop contains instructions for executing the preliminary sizer, the actual weighing of the individual stages, the determination of optimal performance and the calculation of ideal velocity. Preliminary sizing data and control statements are stored in E.SIZER.

```

'UPDATE DBASE'
@XQT WAATS2.0SIZER
#IN PAYLD='PAYLD'
NSTAGE='NSTAGE'
VIDEAL='VIDEAL'
XLAMP='XLAMP'
XISP='XISP'

$END

```

Performance input data and control statements are stored in E.ROBOT.

```

@ASG,T 7.,F///500
@ASG:1 11
@ASG:T 12
@ASG:T 13
@ASG:T 15
@USE T,ODIN-PETE
@T.ROBOT
GO
#INPUT
'UPDATE DBASE'
'ADD ENG1=ENG(1)'
TNE='ENG(1)',0,0,0,'ENG(2)',0,0,0,'ENG(3)',0,0,0,'ENG(4)',0,0,0,
FLBS='TWR*WSB/ENG',
WDOT='TWR*WSB/XISP/ENG',
WTJET='NI',
RE='AECT*TWR*WSB/ENG',
'ADD SREF=DTI(1)/12.*2.*.7854'
S='SREF'
TAUT='TAUT'
NOEUNT='NSTAGE'*1,
'ADD WSB1=WSB(1)'
W01='WSB1'
KCDPHI=1,2,3,4,

```

```
PSIREQ=23859.,0.,21233700.,
NDB(1)=0,1,
NDB(16)=0,1,
NBGCT=2, NENDCT=3, NP=29,
TTBL='TTBL ',
CPTBL='CPTBL ',
CAN=.4,.4,.8,.8,
PNM=0.,.9,1.0,100.,
K4010T=1,
TLIFT=6., TTILT=16., CHIDT='CHIDT ',
TDRAG=4.,
TCHFRZ=1000.,
QY=.001,
DP2=.5,
```

```
MSMCH=
FUELG=1.00,1.00,
MSMCH(13)=2,
IMDCHI=1,
MOTRAC=0,
KIND=2,
NMAX=20,
STEP=2.,3.,8.,8.,
TZERO=0.,
DTZ=15.,
JOB=1,
MOUT=6,
LAST=0,
HOWD=2,
GLING=5,5,5,5,
KNTA=2,
JTHR=0,
K4010T=11,
$
@EOF
```

ORIGINAL PAGE IS
OF POOR QUALITY

The requirements for determining ideal velocity is contained
in the element E.VIDEAL.

```
@XQT WAATS2.OVIDEAL
'UPDATE DBASE'
'PROCESS HMLIST'
$IN RBPORF='RBPORF'
XISP='XISP'
MSTAGE='NSTAGE'
```

The staging loop shown below contains the control statements and data required to generate weights data for the number of stages specified. The three major elements are the establishment of the looping criteria, the calculation of tank weight and the calculation of subsystem weights.

```
@HDG STAGING-LOOP
@D.DLG E.SET/STAGING,SET
@ADD SET
@D.DLG E.TANK,TANK
@ADD TANK
@D.DLG,I REPORT
'UPDATE DBASE'
'PROCESS NMLIST'
'REPORT FROM TANK
  LT='LT'
  VOL='VOL'
```

```
END OF REPORT
@D.DLG E.WAATS,WAATS
@ADD WAATS
@D.DLG,I REPORT
'UPDATE DBASE'
'PROCESS NMLIST'
'ADD WSB(S)=WCOMP(72)'
'ADD NEB(S)=WCOMP(75)'
'ADD XLAMP(S)=MPARM(3)'
```

The looping criteria is contained in the element E.SET/STAGING.

```
'UPDATE DBASE'
'ADD S=S+1'
@SETC 'S'
@TEST ILE/'NSTAGE'
@JUMP ENDSTG
```

When the above element is processed by DLG, the @SETC control statement will contain the stage counter 'S' from the data base. The @TEST control statement will contain the study parameter 'NSTAGE' from the data base. The above control statements, together with the @JUMP control statement, will cause the Exec 8 control to pass to the statement label @ENDSTG when the stage counter 'S' reaches a value of 'NSTAGE.' Since @ENDSTG is in the element E.LOOP/SIZING, the staging loop will be terminated and the launch performance sequence will be started.

The control statements and data for tank sizing are contained in E.TANK.

```
'UPDATE DBASE
@XQT WAAT32.OTANK
6.
'ADD MPI=MFR(3)
'MPI
2
.04131
.029167
3.
3.
33.
40.
'ADD DTI=DTANK(3)
'DTI
```

```
1.4
1.4
33.
1
1
0.
0.
1.
3.
24.
```

The control statements and data for the calculation of subsystem weights are contained in the element E.WAATS.

```
'UPDATE DBASE'
@XQT WAAT32.OWAATS
#INMAP
IPRINT=0,
IENG=1,
CREW=0,
'ADD DTI=DTANK(3)
'ADD ELBODY=LT(1)+LT(2)+DTANK/12.
ELBODY='ELBODY
'ADD TWRI=TWRI(3)
'ADD W3BI=W3B(3)
'ADD ENGI=ENG(3)
ENGIN='ENGI',
HBODY='DTI/12.
ISHAPE=1,
OF=6,
OFACS=0,
```

```

'ADD PCI=PC(8)'
PCHAN='PCI'
'ADD SA1=SA(1)'
'ADD STPS=SA1+SA(2)/144.'
STPS='STPS'
'ADD SBODY=STPS*1.25'
SBODY='SBODY'
SFUTK='SA(2)/144.'
SOMTK='SA1/144.'
TANKS=1;
'ADD TENG=TWRI*WSBI/ENGI'
THRUST='TENG'
UFUTK=30000;
UOMTK=20000;
'ADD WEBI=WEB(8)'
WLANDI='WEBI'
'ADD WPLI=WPL(8)'
WPAYLD='WPL1

```

```

'ADD WPRI=WPR IN'
WPMIN='WPR'
WFOIN='WSBI'
WLF=4;
AC(14)=.13;
AC(19)=.0003;
AC(21)=.15;.10;
AC(28)=.00755;.00033;.5;130.;
AC(37)='WTK(2)'
'ADD WTK1=WTK(1)'
AC(39)='WTK1'
AC(40)=.4;
AC(42)=.15;
AC(44)=.0025;.5;
AC(62)=.001;
AC(55)=.021; AC(110)=.700;
AC(126)=.25; AC(127)=1.0;
AC(68)=2.22; AC(113)=1.;
AC(70)=.0003;
AC(98)=.01
$END

```

ORIGINAL PAGE IS
OF POOR QUALITY

There are many techniques which can be employed to meet the analysis requirements. The Sample Analysis presented above is an illustration only. The EDIN system offers the potential user the capability of performing design analysis ranging from the simplest design sequence to the most complex network of tasks.

REFERENCES

1. Gregory, T. J., Peterson, R. J. and Wyss, J. A.: Performance Tradeoffs and Research Problems for Hypersonic Transonic Transports. AIAA Journal of Aircraft. July-August 1965.
2. Gold, R. and Ross, S.: Automated Mission Analysis Using a Parametric Sensitivity Executive Program. AAS Paper 68-146, presented at the AAS/AIAA Astronautics Specialist Conference. September 1968.
3. Wennegal, G. J., Mason, P. W. and Rosenbaum, J. D.: IDEAS, Integrated Design and Analysis System. SAE Paper 68-0728, Presented to SAE Aeronautics and Space Engineering Meeting. October 1968.
4. Adams, J. D.: Vehicle Synthesis of High Speed Aircraft, VSAC, Volume 1. USAF AFFDL-TR-71-40. 1971.
5. Oman, B.: Vehicle Synthesis for High Speed Aircraft, VSAC, Volume II. USAF AFFDL-TR-71-40. 1971.
6. Lee, V. A., Ball, H. G., Wadsworth, E. A., Moran, W. J. and McLead, J. D.: Computerized Aircraft Synthesis. AIAA Journal of Aircraft. September-October 1967.
7. Herbst, W. B. and Ross, H.: Application of Computer Aided Design Programs for the Management of Fighter Development Projects. AIAA Paper 70-364, Presented to AIAA Fighter Aircraft Conference. March 1970.
8. Wallace, R. E.: A Computerized System for the Preliminary Design of Commercial Airplanes. AIAA Paper 72-793. Los Angeles, California. 1972.
9. Hague, D. S. and Glatt, C. R.: Optimal Design Integration of Military Flight Vehicles - ODIN/MFV. AFFDL-TR-72-132. 1973.
10. Glatt, C. R., Hague, D. S. and Watson, D. A.: ODINEX: An Executive Computer Program for Linking Independent Programs. NASA CR-2296. National Aeronautics and Space Administration. Washington D. C. September 1973.
11. Fulton, R. E., Sobieszczanski, J. and Landrum, E. A.: An Integrated Computer System for Preliminary Design of Advanced Aircraft. AIAA Paper 72-796. Los Angeles, California. 1972.

12. Rhodes, T. R.: The Computer Aided Design Environment (COM-RADE) Project. Presented at the 1973 National Computer Conference New York. June 1973.
13. Rau, Timothy R. and Decker, John P.: ODIN: Optimal Design Integration System for Synthesis of Aerospace Vehicles. AIAA Paper No. 74-72. AIAA 12th. Aerospace Sciences Meeting. 1974.
14. Love, Eugene S.: Advanced Technology and the Space Shuttle. Astronautics and Aeronautics, Volume II, No. 2. February 1973.
15. Henderson, Arthur, Jr.: Aerothermodynamic Technology for Space Shuttle and Beyond. AIAA Paper No. 73-59, presented at the AIAA 9th. Annual Meeting and Technical Display. January 1973.
16. Phillips, W. Pelham, Decker, John P., Rau, Timothy R. and Glatt, C. R.: Computer Aided Space Shuttle Orbiter Wing Design Study. NASA TN D-7478. 1973.
17. Harris, R. V., Jr.: An Analysis and Correlation of Aircraft Wave Drag. NASA TM X-947. 1974.
18. Norton, P. and Glatt, C. R.: VAMP: A Computer Program for Calculating the Volume, Area and Mass Properties of Aerospace Vehicles. NASA CR-2419. 1974.
19. Vanco, Michael: Computer Program for Design Point Performance of Turbojet and Turbofan Engine Cycles. NASA TM X-1340. 1967.
20. Fishbach, Laurence H. and Koenig, Robert W.: GENENG II - A Program for Calculating Design and Off-Design Performance of Two and Three Spool Turbofans with as Many as Three Nozzles. NASA TN D-6553. 1972.
21. Wilwerth, R. E., Rosenbaum, R. C. and Chuck, Wong: PRESTO: Program for Rapid Earth to Space Trajectory Optimization. NASA CR-158. 1965.
22. Stein, L. H., Matthews, M. L. and Frenk, J. W.: STOP - A Computer Program for Supersonic Transport Trajectory Optimization. NASA CR-793. 1967.

23. Kinsey, Don W. and Bowers, Douglas L.: A Computerized Procedure to Obtain the Coordinates and Section Characteristics of NACA Designated Airfoils. Technical Report AFFDL-TR-71-87. Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio. 1971.
24. Hague, D. S. and Glatt, C. R.: An Introduction to Multi-Variable Search Techniques for Parameter Optimization. NASA CR-73200. 1968.
25. Swann, R. T., et. al.: One-Dimensional Numerical Analysis of the Transient Response of Thermal Protection Systems. NASA TN D-2976. 1965.
26. Boren, H. E.: DAPCA: A Computer Program for Determining Aircraft Development and Production Costs. Rand Corporation Report RM-5221-PR. 1967.
27. Glatt, C. R.: IMAGE: A Computer Code for Generating Picture-Like Images of Aerospace Vehicles. NASA CR-2430. 1974.
28. Glatt, C. R., Hague, D. S. and Reiners, S. J.: Prediction of Sonic Boom from Experimental Near-Field Overpressure Data, Method and Results. NASA CR-2441. 1974.
29. Glatt, C. R., Reiners, S. J. and Hague, D. S.: Prediction of Sonic Boom from Experimental Near-Field Overpressure Data, Data Base Construction. NASA CR-2442. 1974.
30. Glatt, C. R.: WAATS: A Computer Program for Weights Analysis of Advanced Transportation Systems. NASA CR-2420. 1974.
31. Gentry, A.: Hypersonic Arbitrary Body Aerodynamic Program. Douglas Report. DAC56080. June 1967.
32. Brauer, G. L., Cornick, D. E., Steinhoff, R. T. and Stevenson, R.: Program to Optimize Simulated Trajectories (POST), Martin Marietta. NASA MCR-73-206. October 1973.
33. Reiners, S. J., Glatt, C. R., Hirsch, G. N. and Alford, G.: GTM: Geometry Technology Module. Aerophysics Research Corporation. JTN-09. December 1974.
34. Glatt, C. R. and Hirsch, G. N.: PLOTTR: An Independent Computer Program for the Generation of Graphical Displays. Aerophysics Research Corporation. JTN-07. December 1974.

35. Glatt, C. R. and Colquitt, W. N.: The DLG Processor - A Data Management Executive for the Engineering Design Integration System (EDIN). Aerophysics Research Corporation. JTN-10. December 1974.
36. Braley, Dennis M.: Users Guide for the Automatic Flow Chart Generator Program (FLOGEN). JSC Internal Note No. 73-FM-62. April 1973.
37. Glatt, C. R., Hirsch, G. N., Alford, G. E. Colquitt, W. N. and Reiners, S. J.: The Engineering Design Integrattion (EDIN) System. Aerophysics Research Corporation. JTN-11. 1974.