E.3.c

N94-23891

VMPLOT: A VERSATILE ANALYSIS TOOL FOR MISSION OPERATIONS

Allen W. Bucher, Owen G. Short

Martin Marietta Astronautics Group, Denver, Colorado 80201, U.S.A.

ABSTRACT

VMPLOT is a versatile analysis tool designed by the Magellan Spacecraft Team to graphically display engineering data used to support mission operations. While there is nothing revolutionary or innovative about graphical data analysis tools, VMPLOT has some distinguishing features that set it apart from other custom or commercially available software packages. These features include the ability to utilize time in a Universal Time Coordinated (UTC) or Spacecraft Clock (SCLK) format as an enumerated data type, the ability to automatically scale both axes based on the data to be displayed (including time), the ability to combine data from different files, and the ability to utilize the program either interactively or in batch mode, thereby enhancing automation.

Another important feature of VMPLOT not visible to the user is the software engineering philosophies utilized. A layered approach was used to isolate program functionality to different layers. This was done to increase program portability to different platforms and to ease maintenance and enhancements due to changing requirements. This paper will describe the functionality of the unique features of VMPLOT as well as highlighting the algorithms that make these features possible. The software engineering philosophies used in the creation of the software tool will also be summarized.

Keywords: mission operations, data analysis, plotting software, Magellan.

1. INTRODUCTION

The Magellan Spacecraft is the National Aeronautics and Space Administration's planetary mission designed to obtain a high-resolution Synthetic Aperture Radar (SAR) image of the surface of Venus. Launched from the Space Shuttle Atlantis on May 4, 1989, Magellan traveled one-and-one-half times around the Sun to arrive at Venus on August 10, 1990. SAR imaging of the Venusian surface commenced on September 15, 1990 revealing the most astounding surface views ever recorded.

The Magellan Spacecraft Team, comprised of a group of specialized spacecraft engineers, is responsible for monitoring the day-to-day health of the Magellan Spacecraft. These engineers utilize a variety of software tools to assess the health and status of each spacecraft subsystem. The most widely used tool by the Magellan Spacecraft Team is VMPLOT. VMPLOT is a software program designed to produce hard copy and screen plots of user selected data.

2. VMPLOT PROGRAM OVERVIEW

VMPLOT is a data file driven program designed to allow Magellan engineers to graphically display spacecraft data on a Visual Display Terminal or Hard Copy device. The main display devices currently supported by the program include the Sun Workstation "gfxtool", X Windows, and the Tektronics 4014 display terminal or Tektronics 4014 emulator. The Hard Copy devices supported are "Laser Writer" type devices that are capable of interpreting postscript files.

VMPLOT receives user data from two required sources; a DATA file and a CONTROL file.

The DATA file(s) are typically time ordered files that contain the x/y data pairs that the user wishes to plot. These files are formatted as shown in Figure 1. The first column is reserved for the time column, either a Universal Time Coordinated (UTC) formatted "time tag" or a time tag formatted in the mission specific Spacecraft Clock (SCLK) format. The other columns are used to contain the user data; spacecraft voltages, for example. As a generic constraint the DATA file is restricted to a single time column followed by up to 36 data columns. The format of these files are fixed and documented in a formal software interface specification.

```
tBAT1       SCET 03 Bat1 Voltage Bat2 Voltage MainBus Volt
92-206/00:00:00.000 EU Vdc       EU Vdc       EU Vdc
92-206/23:59:59.999 %12.4f       %12.4f       %12.4f
00090  92-224/23:03 E-0174       E-0175       E-0180
92-206/00:03:59.424       31.1410      31.1228      31.2815
92-206/00:04:59.424       31.1410      31.1228      31.2815
92-206/00:05:59.424       31.1410      31.1228      31.2815
92-206/00:06:59.424       31.1410      31.1228      31.2815
92-206/00:07:59.424       31.1410      31.1228      31.2815
92-206/00:08:59.424       31.1410      31.1228      31.2815
92-206/00:09:59.424       31.1410      31.1228      31.2815
   .
   .
   .
   .  (or SCLK)

01600591:40               31.0002      30.9815      31.1401
01600592:40               31.0002      30.9815      31.1401
01600593:40               31.0002      30.9815      31.1401
01600594:40               31.0002      30.9815      31.1401
01600595:40               31.0002      30.9815      31.1401
01600596:40               31.0002      30.9815      31.1401
01600597:40               31.0002      30.9815      31.1401
```

Figure 1 VMPLOT DATA File

The CONTROL file instructs VMPLOT which data to plot and how to plot it. The CONTROL file, as shown in Figure 2, contains the DATA file name(s), the plot start and end times, the data columns to plot, the plot axis information, the plot line format, and the plot titles.

The main structure within the CONTROL file is the $SOP (Start Of Plot) - $EOP (End Of Plot) blocks that control the production of a single plot image. Each $SOP-$EOP block must contain exactly 26 records including the $SOP-$EOP records. Each line of the $SOP-$EOP block has special meaning to VMPLOT and must be used in conjunction with other lines within the block to achieve accurate results. The selected CONTROL file may contain any number of $SOP-$EOP blocks. Each block results in the production of one plot image. This feature allows trending plots for related components to be grouped in a single CONTROL file to aid the automation of plot production.

2.1. Plot Types

The two basic plot types utilized by VMPLOT are Channel versus Time and Channel versus Channel. A channel is defined as a column of data from a DATA file. In spacecraft operations most of the data in the DATA file originates from telemetry channels, hence the name.

```
$SOP  /* PLOT #1 */
2                       ! Number of DATA files
tickmap.drf             ! DATA file 1 name
motorcur.drf            ! DATA file 2 name
1                       ! Type 1,2=CVT 3,4=CVC 5,6=REL 7,8=STRP
89-348/20:00:00.000     ! Start Time of the plot
89-348/20:25:30.000     ! End Time of the plot
00                      ! Column for x data - file1
00                      ! Column for x data - file2
1                       ! Num y-axis variables - file1
2                       ! Num y-axis variables - file2
01 00 00 00             ! Col nums of y variables - file1
02 03 00 00             ! Col nums of y variables - file2
AUTO                    ! X axis scale type AUTO or MAN
N/A                     ! X Axis Min, Max (MANual scale)
N/A                     ! # Maj Ticks, # Minor/Major Ticks
1                       ! Draw an X-Axis Grid 1=Yes, 0=No
MAN                     ! Y axis scale type AUTO, MAN, LOG
0 250                   ! Y Axis Min, Max (MAN scale), Decade
6 4                     ! MajTicks & MinorTicks, or Start Exp
1                       ! Draw an Y-Axis Grid 1=Yes, 0=No
2                       ! Line Format 1=strip,2=line,3=points
DMS A Motor Current - Radar Functional     ! Plot Title
SCET                                       ! X Axis Title
Current in Milliamps                       ! Y Axis Title
$EOP
$SOP  /* PLOT #2 */
1                       ! Number of DATA files
tBAT1.drf               ! DATA file 1 name
   .
   .
   .
$EOP
```

Figure 2 VMPLOT CONTROL File

2.1.1. Channel Versus Time

A Channel versus Time (CVT) plot, as shown in Figure 3, includes one or more data channels plotted versus time. The time axis may either be a UTC or SCLK formatted time tag. The UTC time tags are formatted YY-DDD/HH:MM:SS.FFF where YY is the last two digits of the year, DDD is the day of the year, and HH:MM:SS.FFF is hours, minutes, seconds, and fractional seconds. The SCLK time tag is formatted RIM:MOD91 where the Real-Time Image Count (RIM) is a 24 bit counter which is incremented every 60-2/3 seconds and the Mod 91 Count (MOD91) is an 8 bit counter which increments every 2/3 of a second. Both of these time formats are processed by VMPLOT to allow spacecraft analysts to view spacecraft events versus time. The majority of spacecraft analysis during Mission Operations is accomplished by viewing spacecraft performance as a function of time (i.e. spacecraft trends over time).

2.1.2. Channel Versus Channel

A Channel versus Channel (CVC) plot, Figure 4, includes a data channel plotted against another data channel. The most common use of this plot type for Magellan was the analysis of attitude phase plane information (spacecraft position versus rate) and the analysis of High Gain Antenna signal strength versus spacecraft offset. This feature was also used to plot data that is not associated with a specific event time.

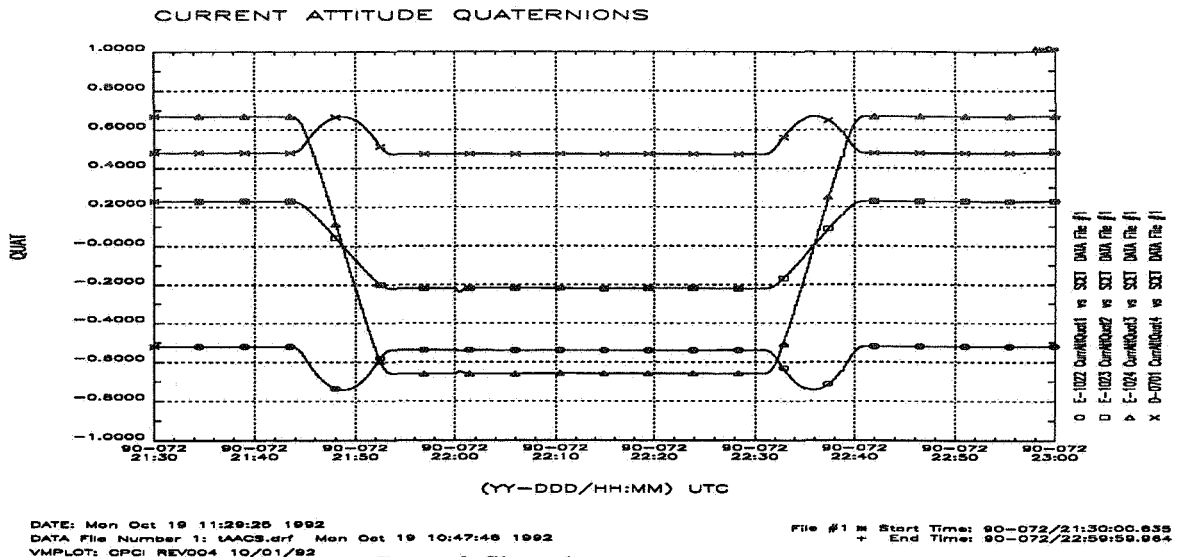

Figure 3 Channel versus Time Plot

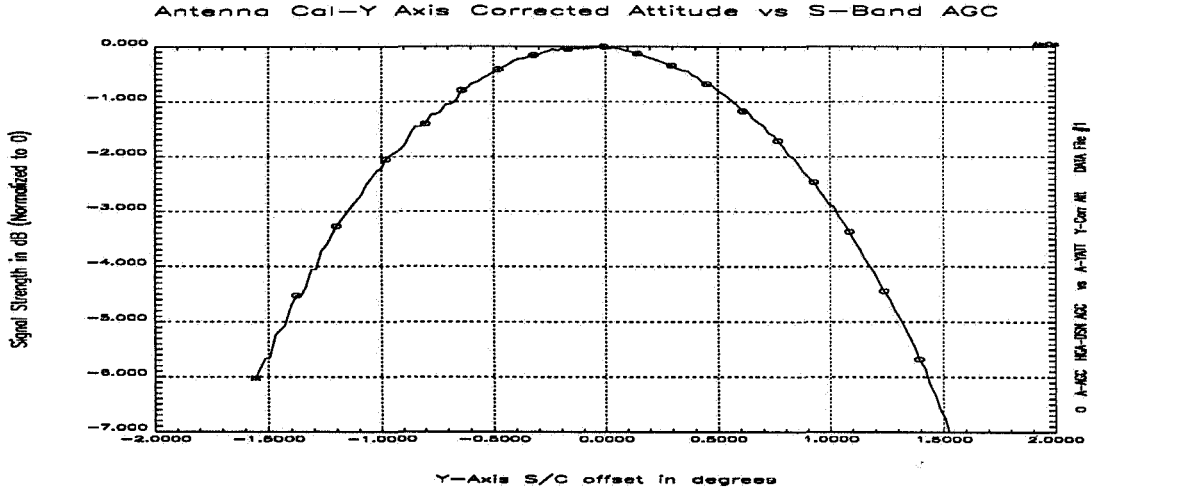

Figure 4 Channel versus Channel Plot

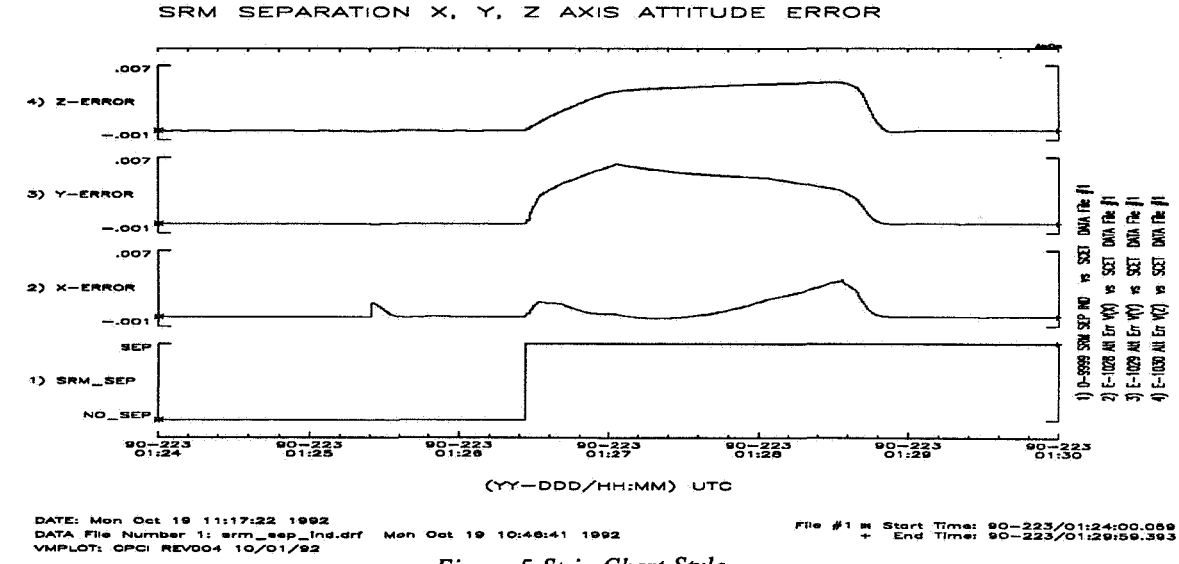

Figure 5 Strip Chart Style

2.1.3. Overplotting and Strip Charts

In combination with the two main plot types described above, CVT and CVC, VMPLOT provides three methods to view the data. The default method provided is the single channel plot, either a single channel versus time or a single channel versus another channel, Figure 4. The second method, as illustrated in Figure 3, is overplotting of up to four channels versus time or up to four channels versus a single channel. In addition to the methods described above, another method is the production of up to four strip chart type plots versus time, Figure 5.

3. VMPLOT FUNCTIONALITY

VMPLOT was designed and coded to meet the requirements of the Magellan Spacecraft Team. The Spacecraft Team levied requirements including the plot template outline, the different plot types, the scale types and options, the drawing types, the display requirements for the data, the devices that the program must support, and the program user interface.

The flow of the VMPLOT program contains many features and functions. Although all features of the software are important, the following paragraphs will highlight the unique features of the program. These features include the ability to utilize Universal Time Coordinated or Spacecraft Clock times as enumerated data types, the ability to automatically scale both axes, and the the ability to combine data from different files on the same plot image. The algorithms behind the features will also be highlighted, where appropriate.

VMPLOT uses a standard Cartesian Coordinate system for its axes and plotting region defining the y values as the ordinate values and the x values as the abscissa values. The program is designed to allow the X Axis to support either time or engineering values. The Y axis can be scaled to support linear scale or logarithmic scale types.

3.1. Time Scaling

The most distinguished feature of VMPLOT is its ability to utilize time as an enumerated data type. The time types currently supported for Magellan include Universal Time Coordinated (UTC) representing Spacecraft Event Time (SCET), Earth Received Time (ERT), and Monitor Sample Time (MST) and Spacecraft Clock (SCLK), see section 2.1.1. VMPLOT processes time, either UTC or SCLK, by first converting it to a common unit, decimal seconds. For UTC this is accomplished by converting the time tags to decimal seconds since 1980. For SCLK the the two main clock components (RIM and MOD91) are converted to seconds and fractions of seconds and combined to form a single decimal second representation.

3.1.1. UTC Scaling Algorithm

For UTC axis the total time delta in minutes is determined from the difference between the plot start and end times. Although the common time unit is decimal seconds, the plot scales are determined with the minimum resolution of one minute. The time delta is utilized to determine the appropriate intervals between successive "ticks" on the axis scale. The "tick" interval is selected from one of twenty-three predefined tick divisions ranging from a minimum of one minute to a maximum of 576,000 minutes (400 days) between successive ticks on the scale. Once the tick interval is determined, the minor (non-labeled) ticks are drawn. After the minor ticks are completed, the first "nice" interval is determined by finding the first whole increment of time, hours or minutes depending upon the resolution of the scale. The program then places the first axis label on the image. The remainder of the axis labels are then drawn at the predefined interval. The plot start and end times in seconds are then utilized to setup the scaling algorithms to assure the data is correctly plotted against the scale just determined. See Figure 3.

3.1.2. SCLK Scaling Algorithm

Since the SCLK time in seconds is similar to other floating point numbers processed by VMPLOT, the SCLK time in decimal seconds is processed by the auto scaling routine, section 3.2, to determine the scale maximum and minimum values. This information is then passed to the routine for drawing normal engineering axes and the axis is treated as any other engineering number, except that the labels are placed and labeled according to the Magellan SCLK conventions.

3.2. Automatic Scaling Algorithm

The automatic scaling routine is designed to accomplish two goals for engineering scales; first determine the scale boundaries based on the input data to be plotted, and second determine the axis labels and intervals to guarantee that the axis labels are at even intervals (Ref. 1). Based on the maximum and minimum values from the input data set, the automatic scaling routine determines the number of label ticks to be placed on the scale, the delta between each label, and the scale maximum and

minimum. The scale maximum and minimum values are then utilized to determine the scaling algorithms to assure the data is correctly plotted.

3.3. Plotting Data

After determining the plot axis and scales, the user data x/y pairs are then "drawn" to the plot image. The world plotting coordinate bounds used by VMPLOT are fixed. The program determines scaling factors between user and world coordinates, slope and intercept, based on the input user data set x and y axis scaling information. Once the scaling factors are determined, each x/y user data pair is scaled to fit within the predetermined world coordinate bounds. The world coordinates of the x/y data pair to be plotted are determined by the equation:

$$w = \text{slope} \times \text{user\_data} + \text{intercept}$$

VMPLOT plots the user data utilizing poly line or poly marker segments of one thousand data points each. Each plot image can contain from 1 to n segments of 1000 data points. This means that there is no program limitations (other than host machine limitations) regarding the maximum number of x/y data pairs that can appear on a single plot image. Plot images with in excess of 100,000 x/y data points have been processed by VMPLOT. The plotting routine also identifies the first x/y pair plotted with a star symbol and the last x/y pair plotted with a cross symbol. The times associated with these data points are also recorded and displayed to the plot image as an indicator of the actual data start and end times.

For the overplotting and strip charts display methods, described in section 2.1.3, the data to be plotted may be selected from one DATA file or from two DATA files. This characteristic allows VMPLOT to extract one set of x/y pairs from one file and another set of x/y pairs from another file and overplot all x/y pairs on the same image. In all cases however, x/y pairs must come from the same DATA file. This feature was created to allow engineers to compare actual spacecraft events versus events predicted by ground software models.

4. SOFTWARE ENGINEERING

The Software Engineering philosophy used in developing VMPLOT was a combination of *Structured Programming* and *Layered Architecture*.

Structured Programming was formulated in the late '60's and early '70's. It is a discipline for programming which emphasizes the use of only one entry and exit point per construct. It is described in Refs. 2-4 as programming using only sequence, selection, and iteration statements in place of gotos for program control. Any program written using goto statements can be rewritten using if-then-else, while, and sequence statements. Although the main emphasis of structured programming is programming using well defined constructs, it also includes top down design and repeated subdividing of major tasks into subtasks. The subtasks must have a clear objective that is easy to design, code, and test.

The main idea of *Layered Architecture* is isolating different functionalities to different layers. This approach is similar to the underlying principles of the International Standards Organization Open Systems Interconnect (or ISO/OSI) reference model. In a layered approach, each layer is its own independent entity. It has no knowledge of the processing details of the other layers. Each layer knows only of its tasks and what it must provide to or receive from adjacent layers. By concentrating on designing VMPLOT using this approach, each different function can be isolated to its own distinct layer. This layering also allowed the software engineer to isolate the environment or host dependent functions, allowing the software products to be used in different computing environments with minimal impact on coding and documentation efforts.

By utilizing both structured programming and layered architecture, software systems can be developed which emphasize portability and flexibility. By approaching software development utilizing these techniques and carefully designing the system, software can be written and easily transported between different environments. This design process builds upon the ideas put forth by Kernighan and Ritchie in the standardization of C to provide maximum portability of code (Refs.5-6). Although code portability plays an important role in this structured and layered technique, the software engineering approach being considered treats the software development process consisting of Requirements, Design, Code, and Test as a whole. This approach also allows all products of the lifecycle to be flexible enough to entertain changes without major impacts.

The Magellan Project Team realized cost savings from these philosophies when the program was ported to other host machines with minimal effort in support of Assembly, Test, and Launch Operations and System Verification Laboratory efforts. This program has also been ported for use to support other programs including the Transfer Orbit Stage and Mars Observer (xvmplot).

4.1. VMPLOT Implementation

The structured and layered architecture of VMPLOT evolved during the design phase. The program was divided into three layers, illustrated in Figure 7, consisting of Data Manipulation Routines (Layer 2), Graphics Package Interface Routines (Layer 1), and Vendor Graphics or Device Interface Routines (Layer 0). The layer 2 routines control all user and data interaction to the program. The layer 1 routines control all graphics calls to the graphics package being utilized. The layer 0 routines contain the actual graphics primitives being utilized, either a commercial package or user defined routines, to produce the graphic effect.

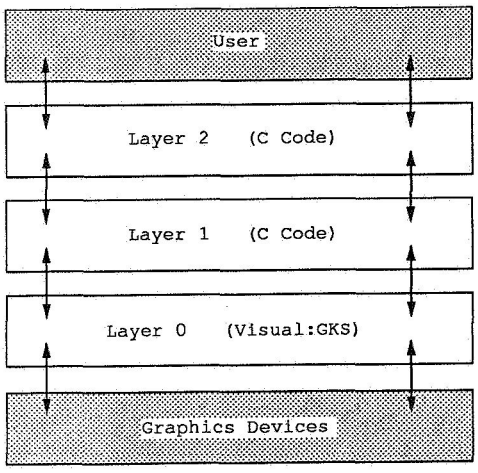

Figure 7 VMPLOT Functional Layers

VMPLOT was coded using the C programming language. The system code was developed using a top-down approach, adhering to the structured and layered constructs introduced during the design phase. The layer 2 and layer 1 routines were custom routines written specifically to satisfy the unique VMPLOT software and data manipulation requirements. The layer 0 routines were provided by purchasing the commercial software package Visual:GKS, developed by Visual Engineering.

Visual:GKS was selected because of its ability to easily satisfy the three main graphics primitives required by VMPLOT; the ability to move the current position within the plotting surface, the ability to draw a line, and the ability to place text on the plot image. Another key feature of Visual:GKS is the ability to produce graphic output for any number of devices utilizing the GRAPHCAP technology. The GRAPHCAP technology, similar to the Unix TERMCAP technology, allows VMPLOT to drive many different output devices with only minimal code changes for device identification. This concept allows the layer 2 and layer 1 routines to be virtually device independent.

5. SUMMARY

The VMPLOT program was designed specifically to support spacecraft data analysis in a Mission Operations Environment. The majority of spacecraft data analysis is accomplished by analyzing or trending events over time. Another characteristic of Mission Operations is that the daily tasks tend to be repetitive in nature. Knowing these facts, VMPLOT was designed to support the time oriented analysis of spacecraft engineering data and provide ease of operation to simplify the daily burden of data analysis. VMPLOT is a generic versatile data analysis tool capable of supporting spacecraft Mission Operations.

6. REFERENCES

1. Lewart, Cass R., "Subroutine plots data on graph automatically", *Electronics*, July 14, 1983, pages 148-151.

2. Mills, Harlan D., and Linger, Richard C., "Data Structured Programming: Program Design without Arrays and Pointers", *IEEE Transactions on Software Engineering*, vol. SE-12, no. 2, February 1986, pages 192-197.

3. Linger, R.C., Mills, H.D., Witt, B.I., *Structured Programming: Theory and Practice*, Reading, MA: Addison-Wesley, 1979.

4. Dahl, O.J., Dijkstra, E.W., Hoare, C.A.R., *Structured Programming*, New York, NY: Academic Press INC., 1972.

5. Kernighan, B.W. and Ritchie, D.M., *The C Programming Language*, Englewood Cliffs, N.J.: Prentice-Hall, 1978.

6. Johnson, S.C. and Ritchie, D.M., "Portability of C Programs and the UNIX System", *The Bell System Technical Journal*, Vol. 57, no. 6, July-August 1978, pages 2021-2048.

7. ACKNOWLEDGEMENT

The work described in this paper was accomplished at Martin Marietta under contract to the Jet Propulsion Laboratory, California Institute of Technology and sponsored by the National Aeronautics and Space Administration.