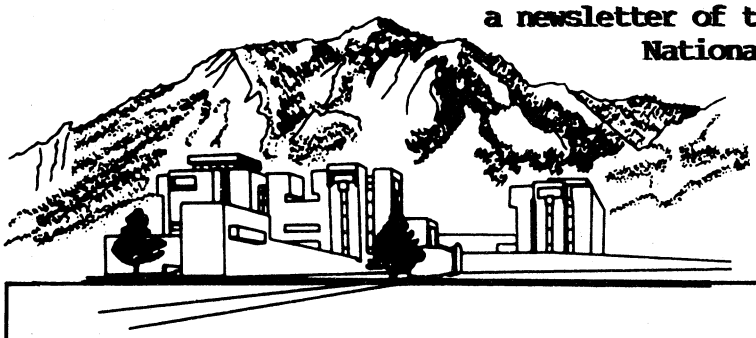
Volume 4, Number 4
April 1, 1983

The Record

a newsletter of the Scientific Computing Division,
National Center for Atmospheric Research



contents

SERVICES DIRECTORY

(303) 494-5151 - FTS prefix 322-5

		ext.	room
NEW USER INFORMATION			
Computing Resource Applications	Cicely Ridley	638	119
	John Adams	573	118
Project & User Number Assignment	Rosemary Mitchell	530	5
Document & Manual Distribution	Sal Farfan	346	17g
REMOTE USER INFORMATION			
Remote Job Entry	Don Morris	582	11d
RJE Password Assignment	Rosemary Mitchell	530	5
Visitor Information	Kathy Lucero	519	6a
OPERATIONAL INFORMATION			
Operational Procedures	Oper. Supervisor	536	29
1/2" Tape Librarian	Sue Long	505	24f
TMS-4 Tape Librarian	Mary Trembour	450	5
Software Distribution/Output Mailing	Sue Long	505	24f

CONSULTING OFFICE

The Consulting Office will be closed every Monday from 13:30-14:30 for staff meetings. If extension 579 is busy, a message may be left on extension 313.

	08:30-12:30	13:00-17:00
MONDAY	Ken Hansen	Harsh Anand Passi
TUESDAY	Ann Cowley	Dan Anderson
WEDNESDAY	Harsh Anand Passi	Dave Kennison
THURSDAY	Barb Horner	Dan Anderson
FRIDAY	Erich Thanhardt	Dave Kennison

SCHEDULE OF MACHINE UNAVAILABILITY

All machines may be down from 07:00 until 08:30 daily for Systems Checkout. In addition, some machines will be down for Preventive Maintenance as follows: 7600, 06:00-07:00 (M T W Th F); CRAY-1, 06:00-07:00 (M Th), 06:00-07:30 (T), 06:00-06:30 (F); TMS-4, 06:00-07:00 (daily); MODCOMP, 08:00-12:00 (1st Monday of month).

The Record is published monthly by the Scientific Computing Division of the National Center for Atmospheric Research. NCAR is operated by the University Corporation for Atmospheric Research and is sponsored by the National Science Foundation. Reference to a company or product name does not imply approval or recommendation of the product to the exclusion of others. David Maxey, Editor; Ann Cowley, Head, User Interfaces; Astrik Deirmendjian, Trouble/Design Reports; Sylvia Darmour, Summary of NCAR Computer Use; Mary Bartels, Computer Resources Allocated.

The Multi-User Software Group left to right--Bob Lackman, Erich Thanhardt, Gloria Williamson, Bonnie Gacnik, Dan Anderson, Herb Poppe, and Stu Henderson

The Multi-User Software Group

The primary responsibility of the Multi-User Software (MUS) Group, under the management of Bob Lackman, is to represent the scientific application needs and interests of NCAR and the university scientists. Through collaborative ventures in modeling, scientific applications, and data processing, the MUS Group is exposed to the needs and perceptions of SCD computing services as seen through the eyes of the scientific user. The MUS Group tries to use this information in two ways.

First, MUS strives to channel user viewpoints regarding facilities, software, and services back into the SCD decision making process in order to achieve a balance between user requirements and internal organizational requirements.

Second, the MUS Group takes direct advantage of the insight gained to create general software tools and facilities which are useful to a broad base of users. From a software perspective, the MUS Group tends to concentrate on specialized software tuned to the unique NCAR environment. Such software is frequently unavailable from the commercial vendor.

Approximately 4.5 FTEs (full time equivalents) are currently involved in Multi-User Software activities. Other User Services activities currently being supported by MUS include two Consulting Office staffing assignments for the User Interfaces Group, and one assignment to the Software and Libraries Group.

Some recent activities of the MUS Group include:

1. Phase out of the CDC 7600 and conversion of processes to the CRAY-1

   The MUS Group has been involved in a wide range of activities in this area including:

   a. Planning of the User Services organization for phase-out of the CDC 7600

   b. Definition of primary activities

   c. Staffing of activities

   d. Creation and chairing of a User Group for CDC 7600 phase-out

   e. Pursuit of large and special conversion problems

   f. Creation of a File Management System to replace PLIB (PSTORE)

      This system is described in the Conversion Handbook, and documentation is also available from the Consulting Office library.

      Contained in this issue of The Record is documentation of two MUS utilities now available to aid the user in moving files from PLIB to PSTORE.

   g. Reading CDC 7600-created volumes on the CRAY-1

      The MUS Group ran a comprehensive study of the feasibility of accessing CDC 7600-created volumes on the CRAY-1 after the CDC 7600 is removed from the network. This study led to the creation of a number of utilities which were needed and resulted in detailed documentation of the associated processes.

   h. Random and asynchronous I/O on the CRAY-1

      Cray Research, Inc. (CRI) is currently working with the MUS Group in defining the requirements of random I/O facilities to be standard CRI-supported software. The MUS Group is serving as a test site for CRI and local I/O products by creating and running a series of test benchmarks.

2. Uniformity on the NCAR network

   The MUS Group has taken an active interest in creating standards and uniformity for JCL, processes, files, and application software. This

viewpoint is reflected in the design of such major software products as the GENPRO Data Processor and PSTORE file system. The MUS Group also supports and recommends other User Services software products which meet these standards including IFTRAN, EDITOR, and the NCAR System Plot Package.

The ideas of standards for software interfaces, documentation, and HELP facilities were also reflected in the Standard EXEC Library created on the IBM 4341. The MUS Group defined the standards of that library and created many of the library EXECS. Before the standards were available, the MUS Group helped users with this new system by providing initial bootstrap EXECs through an experimental library facility.

3. Community Climate Model (CCM) Collaborations with the AAP Division and University Scientists

   The MUS Group is involved in a joint collaborative effort of SCD, AAP, and the UCAR universities in the creation of a new spectral model for forecasting and climate studies. This model will soon become available to the general community to support their research activities. The MUS Group will participate in all aspects of the use and maintenance of this controlled model.

4. Data Processing Collaborations with RAF and CSD

   The MUS Group, in collaboration with the Research Aviation Facility (RAF), has created a modular, portable general time series and spectral analysis processor called GENPRO. This processor currently runs on the CRAY-1, the VAX 11/780, and the CDC 7600. Both RAF and the Convective Storms Division (CSD) of NCAR have been using this software package to meet a variety of processing needs during this past year. Look for a general announcement of the availability of this utility in the May issue of The Record. (Volume 3, Number 5).

by Bob Lackman

Attention IBM 4341 Users

A virtual machine, CONSULT1, has been established to allow user contact with members of the Consulting staff. This machine has two main purposes: to be a vehicle for requesting help with your computing questions and to serve as the online form of the Suggestion Notebook.

The mail facility should be used to send messages to this machine via the command "TO CONSULT1". Users not familiar with this command should use the HELP facility, "HELP MAIL", to learn how to use it. Please use the subject field to indicate if your message is a request for help or a suggestion. Messages will be read each working day at approximately 08:30 and 13:00. They will be answered soon after they have been read.

Suggestions which are sent to CONSULT1 will be answered each month in The Record by an appropriate person. This will replace the Suggestion Notebook which was located in the Consulting Office library in the past. Users may also make suggestions by mailing in the special form for this purpose which appears each month in The Record.

by Barb Horner

NCAR Graphics Software—An Investigation of Future Options

The Graphics Project has completed a preliminary analysis of the evolving computer graphics software needs of NCAR computer users, and how best to meet those needs.

In the mid-1970s, NCAR was among a handful of pioneers of the concepts of portability and device independence in computer graphics software. While the system has served NCAR well, the portability and functional characteristics of the system have been essentially unchanged since the mid-1970s. Meanwhile the NCAR environment has changed substantially. First, more diverse computing environments and graphics hardware have become available to NCAR's users. Second, the nature and diversity of computer graphics applications at NCAR have increased.

In consequence, several inadequacies inherent in SCD's current computer graphics software system have been uncovered. These problems include functional deficiencies, system portability problems, difficulty of direct device interfaces, and limitations on both transporting NCAR programs outside of NCAR and bringing useful outside products to NCAR.

About a year ago the Graphics Project began to study and identify those enhancements to the NCAR graphics software which would be required to keep pace with changing hardware technology and user needs. At about the same time as a preliminary requirements specification was completed, it became apparent that the acceptance of the first international computer graphics standard was imminent. Since the functionality of this standard appeared to satisfy those needs which we could foresee for the next few years, the Graphics Project changed emphasis and examined the appropriateness of adopting the standard.

The standard is known as the Graphical Kernel System, or GKS. It had its origin in the common concepts of the early device-independent software systems of the mid-1970s. The effort to synthesize the best concepts of such systems into a coherent unit has taken over five years, and has finally resulted in a product acceptable to the international community. The technical content of GKS is complete and frozen, and it is expected that the procedural and editorial steps required for its formal adoption will be complete by early 1984. All national bodies of the International Standard Organization (ISO), including the American National Standards Institute (ANSI), have voted to accept GKS.

Because the possibility of adopting GKS appeared to the Graphics Project to be an attractive option, the advantages and potential problems of that course have been carefully studied. A summary of the pro and con arguments for GKS adoption are presented here.

Advantages of GKS Adoption

1. GKS would be a clean way of resolving the most serious of the current functional deficiencies of the NCAR graphics system: lack of a useable color capability; lack of filled-area capabilities; and lack of any form of graphical input facilities.

2. GKS would provide other valuable functional capabilities, including:

   * Logical pixel arrays (a mechanism for integrating raster-like data with line drawing).

   * Optional clipping as an integral part of the system. (Clipping is the capability currently provided by the separate WINDOW utility.)

   * Fully functional segmentation (segmentation is a coherent extension of the capabilities provided by NCAR's current FLASH routines).

   * A "logical pen" capability. Such a feature allows a user to select by index a number of styles for lines, text, etc, without having to worry about the precise details of appearance (intensity, dash pattern, line width, etc.) or the capabilities of the target device (e.g., is it color or monochrome). What is guaranteed on every target device is visual distinguishability of graphical entities, which is often all that is desired.

3. Conversion to GKS would provide a good opportunity to resolve software portability problems inherent in the current system. Some currently required support routines (INTT, LOC) are difficult or impossible to implement on some of the new computer systems, and there are inherent language portability problems in the current implementation of dual-mode coordinates. These problems have already forced the splitting of the software system into multiple versions (which are now starting to diverge) in order to accommodate such popular computers as the VAX and Perkin-Elmer 3000 series.

4. The structure of GKS primitives makes possible certain coding efficiencies, such as vectorization, which can not be had with the current software system.

5. Adherence to a standard has many inherent advantages:

   * Programs written at NCAR using graphics would have much wider portability to other installations.

   * Within a few years, there will be a great proliferation of standard-based programs and products (both commercially and free in the public

domain) which would be available to NCAR computer users.

* After supplying a "starter system" based on a minimal GKS implementation, SCD staff would be able to concentrate on the more powerful high-level utilities, knowing that adequate support for customized or more advanced GKS packages would be widely available.

6. GKS provides a clean and well-defined interface to devices and device drivers. As a result, different device drivers can be connected directly to GKS and the application, which is a prerequisite for true interactive graphics applications.

7. GKS is a layered system, and this layered structure allows users and implementors on different computers (or even different users on the same computer) to select a balance of richness and efficiency appropriate to the application.

8. A minimal configuration of GKS is probably smaller and faster than the current software system.

9. Conversion to GKS would be a clean departure from the lowest levels of the NCAR system, the SPP. Such a clean departure would in many ways be less disruptive than trying to implement improvements and fixes within the current system. A patched and extended SPP would look and behave similarly to the old SPP, but would necessarily differ in many subtle and confusing ways.

Potential Problems with GKS Adoption

1. The conversion could be a significant disruption for some users.

2. Some functionality of the current SPP may be lost, for example non-linear scaling and the mixing of integer and floating coordinates.

3. Any levels above the minimal level of GKS would be bigger and slower than the current system.

4. Implementation of GKS and conversion of utilities would require a significant manpower commitment.

5. There may be possible distribution problems if we purchase the higher levels of GKS or products based on GKS.

6. Adherence to a standard can restrict flexibility.

These potential problems can be minimized by careful planning of the conversion and provision of conversion aids. Specifically:

1. User disruption would be minimized in several ways. First, the old system would remain available indefinitely (frozen in its current state) on binary and source libraries, for those users who do not need the better

facilities of GKS. Second, for those users who wish to gradually convert to the new system, conversion aids would be made available. These would include a software translator (a "hook") whose external interface is identical to the current SPP, and which passes its calls on to the appropriate GKS facilities.

Finally, the current NCAR metafile would be a subset of the extended metafile which would service GKS. Hence the output of any currently existing codes, as well as any saved metafiles, would be acceptable to extended-metafile translators. Conversely, current metafile translators would be able to gracefully accept extended metafiles, if the minimal changes were made to recognize and ignore the new opcodes.

2. The lost features of the current SPP would be available via the just mentioned "hook". Some of the valuable features of the SPP, such as graph backgrounds, non-linear scaling, etc., would be preserved by embodying them in new, middle-level utilities.

3. It is unavoidable that the richness of the higher levels of GKS would result in larger and slower codes. As mentioned before, however, the minimal level may be smaller and faster, and there would be no problem with offering different levels in different libraries on the same machine, to accommodate users with different requirements.

4. The manpower commitment and distribution problems with purchased software present the most significant difficulties. We feel we would be able to effect implementation of a minimal system, generating an expanded NCAR metafile, as well as convert the high-level utilities and produce the conversion aids. This would be in the public domain, and would offer functionality greater than the current system. With current staffing, it is doubtful that we could produce the richest possible GKS system from scratch. There is certain to be such a system in the public domain eventually, which we could then integrate.

   It should be noted that even if we do not adopt GKS, there will be significant manpower required to extend and correct the current software system.

5. The restrictiveness of a standard can be just as easily considered an advantage as a disadvantage.

On the balance, we feel the advantages significantly outweigh the disadvantages. The conclusion of the review at this functional level is that GKS should be adopted. The Graphics Project is continuing the review at a very detailed level, in order to identify exactly what must be done to implement minimal GKS, what is possible in the way of conversion aids, etc.

We would welcome any feedback on this article through the "Your Turn" page, which may be found at the end of this issue of The Record.

by Lofton Henderson

STATLIB Manuals Available

STATLIB manuals are now available from Sal Farfan (ext. 346). STATLIB is a statistical analysis software package which is available on the CRAY-1 computer at NCAR. General questions about STATLIB may be directed to the SCD Consulting Office (ext. 579), while technical questions about the statistical techniques used in STATLIB should be directed to Ginger Caldwell (ext. 640 or 519). As before, a reference copy of the STATLIB manual is available in the Consulting Office library.

STATLIB is a collection of 146 routines (nearly 20,000 lines of FORTRAN source code) released by the Statistical Engineering Laboratory of the National Bureau of Standards in Boulder, Colorado. The emphasis is on time-series analysis and includes routines for bivariate spectral analysis and ARIMA (Auto Regressive Integrated Moving Average) modeling. Other data analysis techniques covered include data plotting on a line printer; analysis of a single random sample, including histograms and tests for non-randomness; one-way analysis of variance; correlation analysis; linear and nonlinear least squares; and random number/data generation. Each method of analysis is supported by two or more routines of varying complexity. A "simple" routine minimizes the complexity of the FORTRAN CALL statement by using preset parameters; other subroutines provide user control of computations, printouts, and storage of computed results when longer argument lists are used.

On the CRAY-1, the binary version of STATLIB may be accessed by specifying the library on the loader (LDR) card. A simple STATLIB job on the CRAY-1 looks like:

```
JOB,JN=name,US=uuuupppppppp,T=t,OLM=olm.
CFT.
LDR,LIB=STATLIB.
EXIT.
\EOF
:
:  [user's program which calls STATLIB routine(s)]
:
\EOF
:
: [user's data cards]
:
```

where name is the job name, uuuu is the user number, pppppppp is the project number, t is the time, and olm is the output limit in blocks.

by Ginger Caldwell

Mass File Transfers to PSTORE

Two utilities have been written to simplify the conversion process from PLIB volumes (or any other volume containing named or unnamed character files) to the new PSTORE system.

The utility PTRANS allows the user to build structured PSTORE directories by specifying a source volume (such as a PLIB volume name), the names of directories which are to be the destinations of file transfers, and a double column input consisting of the CDC 7600 source file names and their respective destination file names under the specified PSTORE directories.

There is no limit on the number of source volumes which can be specified during the course of a single run, but the user is cautioned to limit the number of files transferred on a single run to a reasonable number, and to verify that the files are being correctly translated.

The utility PAUTO performs the same function as PTRANS but in an automatic manner similar to that of PLIBCONV. The user need specify only the source volume name and the target PSTORE directory. Unlike PTRANS, only one source volume can be specified per invocation of PAUTO. Users intending to use PAUTO must first read the discussion below on PTRANS for information on the keywords used by both utilities, as well as to gain an understanding of the JCL file which PAUTO produces.

Because file names on PSTORE are limited to 9 characters, and file names on a CDC 7600 volume can have 17 characters, PAUTO will generate unique file types for any two files whose file names are the same. Also, all embedded special characters will be squeezed out of CDC 7600 file names (PSTORE names are alphanumeric only).

This article assumes rudimentary knowledge of the PSTORE system: specifically it assumes knowledge of PSTORE directory and file naming conventions.

PTRANS

Input Data to PTRANS

The PTRANS utility requires as input a file consisting of a sequence of "data segments", each of which follows the same simple format.

Volume and Read Password Specifications

The first line of each data segment specifies a source volume and its associated read password, if it has one. To specify the volume, use the keyword **PDN=** followed immediately (no blanks!) by the name of the volume. For PLIBs you must supply the VSN (tape name of the form Pxxxxx) for the volume. If the volume is read password protected you must include on the same line the keyword **R=** followed immediately by the read password of that volume. A source volume name and its read password must appear on the same line, and nothing else may be specified on the same line with them. Of course, if the volume is not password protected you will omit this keyword.

Directory Specification

The second line of a data segment consists of a PSTORE directory specification indicating the target directory to which all files in the current data segment will be sent, until a new directory specification is detected in the input. Thus, a data segment can specify any number of directories. To specify a directory, use the keyword **DIR=** followed by a valid PSTORE directory. A valid PSTORE directory must begin with /TB/USERNAME/ for the TBM, where USERNAME is your PSTORE username. All directory names must terminate with "/". A directory specification must occur alone on a single line.

Source and Destination Files

A directory specification is followed by a two-column input, where column 1 specifies a CDC 7600 source file name and column 2 specifies the destination PSTORE filename within the directory. All files will be copied to the most recent directory specified in the input file.

You can specify an unnamed file by its source sequence number prefixed with "FS=". Thus, the unnamed file with source sequence number 3 would be referenced by "FS=3" instead of by name.

The PSTORE file names listed in the right column are any valid PSTORE file names (including file types and version numbers if you wish). You may not include a '/' in any of the names however, as this implies the specification of a new directory.

The source and destination file names must be separated by at least one blank. Otherwise, positioning within the line is not important. Exactly two file names must be specified per line.

To begin a new data segment in the input file (indicating that source files are now coming from a different volume) simply include a line indicating the new source volume and repeat the above procedure.

Current Limitations

There is no limit to the number of files, source volumes, or directories that can be specified in the input stream for PTRANS other than limitations inherent in PSTORE itself. The one major restriction presently is that ALL SOURCE FILES MUST BE CHARACTER FILES.

PTRANS may be extended later to include binary files as well, depending on the desire and demand for such a feature.

The user input data file must be on $IN, and the output will be written to $OUT.

Inputs can occur anywhere within columns 1-79 of the input file.

Output

Your output will consist of the JCL file generated by PTRANS plus the normal logfile. All error messages as well as indications of the success or failure of your file transfers will be written in the logfile at the end of your job. A PSTORE directory listing is also produced both before your run and after each directory transfer has been completed so that you can verify (on a directory-by-directory basis) what has actually been written.

Examples

If, for example, you first wish to transfer files from volume P12345 (with read password XXXYYY) to your top level directory, and your PSTORE name is JSBACH, you might have the following input file:

```
PDN=P12345 R=XXXYYY
DIR=/TB/JSBACH/
76FILE1      FILE1.FOR
76FILE2      FILE2.DAT.1
76FILE3      FILE3
```

Note that the columns of file names have been justified for legibility but it is not necessary to do so.

To continue the previous example, suppose you wish to transfer two additional named files to a subdirectory called /TB/JSBACH/PROGRAMS/ and another unnamed file from volume XMODEL (no read password) with source sequence number 5 to a subdirectory called /TB/JSBACH/DATA/. The entire input data file might look like:

```
PDN=P12345        R=XXXYYY
DIR=/TB/JSBACH/
76FILE1                    FILE1.FOR
76FILE2                    FILE2.DAT.1
76FILE3                    FILE3
DIR=/TB/JSBACH/PROGRAMS/
NUMERICMODEL1              MODEL1.FOR
PROCESSOR1                 PRCSSOR1.FOR
PDN=XMODEL
DIR=/TB/JSBACH/DATA/
FS=5                       XMODEL5.DAT
```

Job File Structure

In order to access the utility PTRANS, include an ACQUIRE statement:

```
ACQUIRE,DN=PTRANS,MF=TB.
```

There are two ways to invoke PTRANS, as follows:

```
CALL,DN=PTRANS.
-PTRANS.
```

-OR-

```
CALL,DN=PTRANS.
-PTRANS,GO.
```

The first form generates the necessary JCL to perform the file transfers, but does not execute that JCL. The second form allows generation of the JCL and its immediate execution. In either case, your output will include a listing of the generated JCL.

The user is STRONGLY ADVISED to first use the first form of PTRANS and to verify for himself that the JCL being generated is what was expected.

To conclude the previous example, the following is a complete job file to generate and execute the JCL necessary for the specified file transfers:

```
JOB,...
ACQUIRE,DN=PTRANS,MF=TB.
CALL,DN=PTRANS.
-PTRANS,GO.
\EOF
PDN=P12345          R=XXXYYY
DIR=/TB/JSBACH/
76FILE1                     FILE1.FOR
76FILE2                     FILE2.DAT.1
76FILE3                     FILE3
DIR=/TB/JSBACH/PROGRAMS/
NUMERICMODEL1               MODEL1.FOR
PROCESSOR1                  PRCSSOR1.FOR
PDN=XMODEL
DIR=/TB/JSBACH/DATA/
FS=5                        XMODEL5.DAT
\EOF
```

Problems

Problems and questions may be directed to Erich Thanhardt (ext. 382) or to the Consulting Office.

PAUTO

To use PAUTO, the user need only specify one (and only one) source volume together with its read password (if it has one) on the first line of the input file, and a directory specification on the second line. See the documentation above on PTRANS for a description of how to specify volumes, read passwords and directories.

In order to access the utility PAUTO, include an ACQUIRE statement:

```
ACQUIRE,DN=PAUTO,MF=TB.
```

There are two ways to invoke PAUTO, as follows:

```
CALL,DN=PAUTO.
-PAUTO.

 -OR-

CALL,DN=PAUTO.
-PAUTO,GO.
```

The first form generates the necessary JCL to perform the file transfers, but does not execute that JCL. The second form allows generation of the JCL and its immediate execution. In either case, your output will include a listing of the generated JCL.

The user is STRONGLY ADVISED to first use the first form of PAUTO and to verify for himself that the JCL being generated is what was expected.

The output for PAUTO includes the normal output for a PTRANS run plus a table indicating how CDC 7600 file names were mapped into PSTORE file names. The user is strongly urged to check this table (using the first form of PAUTO above) to verify that unique file names are, in fact, being produced in the expected manner. If you are dissatisfied with the file names being generated, you may wish to go with PTRANS. PTRANS may cost you far less time to use than PAUTO and then having to change many of your PSTORE file names later!

PAUTO is restricted to character files. Volumes with binary files may produce garbage on PSTORE (in particular, TBMCONV may fail on them). Unnamed files will cause names of the form NCARSYSTE.xxx to be generated on PSTORE.

Example

The following example illustrates conversion of the PLIB P12345 (with no read password) where all files are written to the PSTORE directory /TB/JSBACH/.

```
JOB,...
ACQUIRE,DN=PAUTO,MF=TB.
CALL,DN=PAUTO.
-PAUTO,GO.
\EOF
PDN=P12345
DIR=/TB/JSBACH/
\EOF
```

Problems

Problems and questions may be directed to Erich Thanhardt (ext. 382) or to the Consulting Office.

by Erich Thanhardt

New Library Packages

Documentation for these new library files is available by listing the comment cards at the beginning of the source file (the GETDOC utility on the CRAY-1 will do this) or from the documentation file cabinet in the Consulting Office (ext. 579).

CODY — Modified Bessel Functions of the First Kind

Contributor: Written by William J. Cody of Argonne National Laboratory, implemented for NCAR by James H. Curry.

Purpose: Calculates Bessel Functions I SUB(N+ALPHA) (X) for non-negative argument X, and non-negative order N+ALPHA, with or without exponential scaling. The software is now available on XLIB and has been implemented for use only on the CRAY-1.

The FORTRAN subroutine is dependent on only seven machine constants and is therefore transportable. The code is based on algorithms implemented by W. Cody and is currently the fastest available on the CRAY-1, when modified Bessel functions are called for.

To obtain documentation for this routine, include the following card image in a CRAY-1 job:

```
GETDOC,LIB=XLIB,DOC=CODY
```

Software Notes Update: Bessel Functions

Introduction

This article is an update of the Software Notes column on Bessel functions which appeared in Computing Facility Notes No. 59. Since the previous article, NCAR has acquired the NAG library and new software devoted to Bessel functions. Further, this article also contains the results of extensive timing test to aid the user in choosing among the available routines.

The earliest systematic studies of the cylindrical functions now known as the Bessel functions can be traced to F. Bessel in 1824. Apparently what are known as the integer Bessel functions J_n appeared in the memoirs of L. Euler (1764) where he considered the problem of a stretched circular membrane. They were also studied by J. Lagrange while considering problems in celestial mechanics (1769), and by the Bernoulli brothers (James and Daniel) during the period 1694-1732.

Bessel functions are solutions to the second order ordinary differential equation:

$$\frac{d^2z}{dx^2} + \frac{1}{x}\frac{dz}{dx} + \frac{(1 - f^2)}{x^2} z = 0$$

which is called the Bessel function of order f, and z is a function of x.

The simplest Bessel functions are of nonnegative integer order in which f is an integer n (n=0, 1, 2,...). These are called Bessel functions of the first kind of integer order. If the sign preceeding f is changed in the equation above, the solutions are the modified Bessel functions.

Since the defining relation for Bessel functions is a second order differential equation it is not a surprise that there are solutions which are linearly independent of J_n. These functions are denoted by Y_n and are called Bessel functions of the second kind. Bessel functions of the second kind have the following defining relation:

$$Y_n(z) = \frac{J_n(z)\ \cos(n\ pi) - J_{-n}(z)}{\sin(n\ pi)}$$

and the associated modified Bessel functions of the second kind are denoted by K_n.

Bessel functions of the first and second kind satisfy the simple recurrence relationships:

$$J_{n-1}(z) + J_{n+1}(z) = \frac{2n}{z} J_n(z)$$

$$J_{n-1}(z) - J_{n+1}(z) = 2 \frac{d}{dx} J_n(z)$$

hence Bessel functions of the first kind are all expressible in terms of J_0 and J_1. Similar recurrence relations hold for the modified Bessel functions of the first kind I_0 and I_1.

NCAR Software

NCAR currently has available approximately five dozen routines in six mathematical libraries which can aid in various aspects of computing when Bessel functions are required.

Most of the libraries or packages have a full complement of Bessel function software. In particular, most have code for computing Bessel functions of the first and second kind as well as the modified Bessel functions. Further, several of the routines implement an algorithm due to Sookne [4] while others trace their origin to the National Activity to Test Software [2],[3]. The fundamental reference on Bessel functions is [5].

It should be noted that while there are some intersections among the various package and library routines each has its own subtleties; for example, IMSL routines must be typed either real single or double precision, otherwise incorrect results may be obtained. It is therefore strongly recommended that

a potential user of any of the special function software read the documentation carefully (see Computing Facility Notes No. 59 for more details).

Recommendations

Since NCAR has such an extensive collection of Bessel function software (all of which provides at least thirteen decimal digits precision) the following suggestions are offered for choosing among them:

1. If the application only requires the computation of the Bessel functions $J_{0,1}$ or $I_{0,1}$ it is recommended that library function software be used.
2. If the user's application requires Bessel functions of order greater than one it is recommended that FORTRAN-callable subroutine versions of available routines be used.

The above suggestions are primarily based on the empirical observation that the function evaluation software is approximately five times faster than the analogous subroutine callable software when the evaluation of Bessel functions are concerned. Further, the FORTRAN-callable subroutines were designed to compute more than a single Bessel function and should be expected to require more time for such things as initialization. There is a natural division of the routines into two groups: callable subroutines and function evaluations. The following table contains the results (by library) of timing test on the CRAY-1 of the integer Bessel function software available at NCAR.

TABLE I

SUBROUTINE CALLABLE J_n			SUBROUTINE CALLABLE I_n		
NCARLB	1	1.00	NCARLB	1	1.00
AMOSLIB	2	1.23	AMOSLIB	2	1.23
PORT	3	1.43	PORT	3	1.45

FUNCTION EVALUATION $J_{0,1}$			FUNCTION EVALUATION $I_{0,1}$		
IMSL	1	1.00	NAG	1	1.00
NAG	2	1.06	IMSL	2	1.54
AMOSLIB	3	1.93	AMOSLIB	3	2.30
FUNPACK	4	2.12	FUNPACK	4	2.68

The entries in TABLE I indicate the rankings of the specified software by category and timing. For example, of all available software for computing I_0 or I_1 using function evaluations, the routines in the NAG library were the fastest, being 54% faster than the corresponding IMSL routine in the same subgroup, etc.

A concise tabulation of types and location of the various software is presented in Table II.

TABLE II

	J	I	Y	K
AMOSLIB	CIFS	CIFS	I	I
FUNPACK	I	IS	IF	IS
IMSL	I	IS	IF	I
NAG	I	I	I	I
NCARLB	CI	CIS	N	N
PORT	CI	CI	N	N
ULIB	CI	CI	N	N

Notes:

The various columns of the table indicate whether:

A library has software for the evaluation of Bessel functions of a complex argument (C), integer (I) or fractional (F) order, or exponentially scaled (S) Bessel functions of various kinds, while (N) indicates that software of a specified type is not available in a corresponding library. Finally, though not indicated in TABLE II, some libraries do provide double precision software and the user should refer to the NCAR Software Catalog for more details.

Catalog Documentation

Inclusion of the following job control statement in a CRAY-1 program will produce a short description of available Bessel function routines:

```
GETDOC(LIB=DOCLIB,DOC=B5I)
```

References

1. Abramowitz, M. & I.A. Stegun, 1968: Handbook of Mathematical Functions, Chaps. 9-10. Dover Publications.

2. Cody, W.J., March 1975: "The FUNPACK Package of Special Function Subroutines." ACM Tranctions on Mathematical Software, Vol. 1, No. 1, pp. 13-25.

3. Cody, W.J., R.M. Motley, & L.W. Fullerton, September 1977: "The Computation of real Fractional Order Bessel Functions of the Second Kind." ACM Tranctions on Mathematical Software, Vol. 3, No. 3, pp. 232, 239.

4. Sookne, D.J., July-December 1973: "Bessel functions of real arguments and integer order." Journal of Research of the National Bureau of Standards. B. Mathematical Sciences Vol. 77 A, Nos. 3 & 4, pp. 125-132.

5. Watson, G.N., A Treatise on the Theory of Bessel functions.

by James H. Curry
University of Colorado and NCAR

Documentation Update

The purpose of this column is to announce revisions, updates and new documents of interest to the user of SCD's computers.

Included at various times will be documents issued by SCD, by NCAR but outside SCD, by IBM, and by Cray Research, Inc. Directions for obtaining the documents are included.

Please note that manuals ordered from SCD will take approximately two weeks to reach you. This is because we use the library rate to mail our manuals.

SCD Documents

New documents available from the Consulting Office (Room 11A, ext. 579) are:

* "Disposing Plot and Print Files Over the Local NCAR Network"
* "STATLIB - Software Package for Statistical Analysis"
* "New CRAY-1 Utility Provides Same VSN Information as Old CDC 7600 Program"
* "UPKEEP - A Program for TBM Volume Maintenance"
* "GBYTES/GBYTE, SBYTES/SBYTE"
* Edition 01.01 of "PSTORE - A Permanent File Storage System." This version is more current than the information in the Conversion Handbook.
* Also available are copies of the article "Mass File Transfers to PSTORE" which appears in this issue of The Record.

NCAR Publications

The following reports are recent NCAR or UCAR publications which may be of interest. They are available from Roane Simkin in the Publications Office (ext. 281) unless otherwise stated.

Joint Solar Dynamics Project Data Summary, 14 June-13 August, Volume I: Chromospheric and Coronal Observations, by R.R. Fisher, C.J. Garcia, K.A. Rock, P.H. Seagraves, and E.A. Yasukawa (NCAR) and M.K. McCabe, D.A. Mickey, J.R. Najita, J.R. Lieberman, and M.G. Hardy (University of Hawaii); NCAR Technical Note No. 202, November 1982.

Plan for the Development and Utilization of Ocean Surface Drifting Buoys: Drifters, by James C. McWilliams and John E. Masterson; UCAR publication, September 1982; available from Nancy Wright (ext. 365).

The National STORM Program--Scientific and Technological Bases and Major Objectives, edited by Richard Anthes; UCAR publication, January 1983, 520 pp.;

available at a cost of $11 from Nancy Wright (ext. 365).

by Linda Besen

Summary of Daily Bulletin Items

Below is a summary of some items which have appeared in the Daily Bulletin (the NEWS file for remote entry users). These items concerned systems changes, operations procedures, and other news of general interest. They may still be of interest to users and are listed below by topic and date.

CRAY-1:

February 11, 1983
Beginning February 9, the trailer information on CRAY-1 background jobs was changed. The last output line previously did not reflect the background discount:

```
TOTAL CCUS USED (HOURS) -          0.1234
```

The new format reflects the actual background charge used for accounting:

```
TOTAL CCUS USED (HOURS) -          0.0308
   (THE ABOVE IS BACKGROUND RATE CHARGE)
```

February 16, 1983
The semicolon (;) has been added as a legal control statement continuation character on the CRAY-1. It may be used in exactly the same way as the caret (^) is used.

February 24, 1983
The mainframe designator for the existing CRAY-1, Serial Number 3 is "C1". The mainframe designator for the new CRAY-1, Serial Number 14 is "CA". The Daily Bulletin will make announcements using these mainframe designators.

February 28, 1983
Machine C1 has the remainder of Bugfix 2 for COS 1.11 installed this morning. These are internal fixes to the operating system. This is COS 1.11, Version 6.

The phase 1 installation testing of the CRAY resident job queue manager begins this morning. This phase passes all jobs through to input as normal except for background jobs. Background jobs are moved from the 7600 background queue to a CRAY resident background queue and from there into the machine.

March 2, 1983
PSTORE: Change 01 of Edition 01 (E01.01) of "PSTORE - A Permanent File Storage System" is available in the Consulting Office or from Sal Farfan (ext. 346). This change documents the use of the semicolon as a continuation character on the CRAY-1 (page 13), corrects the third PCOPY example (page 18), and

adds a new PCOPY example (page 18).

March 7, 1983
A new CRAY-1 user utility program provides TBM VSN information by user and/or project number which was previously available only on the CDC 7600. The CRAY-1 program uses the same data card images and produces the same output as the 7600 program; only the JCL is different. The two versions will temporarily run in tandem. A typical deck set up to run the CRAY-1 version is shown below:

```
JOB,JN=jobname,US=userproject,OLM=100.
TBMVSN.
EXIT.
\EOF
PROJ
    43310016
    43310016/D
    43310016/T
    43510012/T
USER
    5136
    8039
    8039/D
    8039/T
END
\EOF
```

March 11, 1983
TBM "BK=1" VOLUMES: A fix has been generated on the CRAY-1 to handle volumes with BK not equal to 8 and having partial blocks. This fix was installed on Monday, March 14.

Summary of NCAR Computer Use for February 1983

7600 COMPUTER	FEBRUARY Total	FEBRUARY Day Avg.	FISCAL YTD Total	FISCAL YTD Day Avg.
Clock Hours in the Month	672.00	24.000	3624.00	24.000
less Scheduled PM	17.83	.637	99.58	.659
less Hardware Downtime	6.75	.241	54.03	.358
less Software Downtime	0.00	0.000	.30	.002
less Environmental Downtime	1.97	.070	26.09	.173
less Operations Use	9.00	.321	51.48	.341
less Other Causes	1.60	.057	23.05	.153
Clock Hours Up	634.85	22.673	3369.47	22.314
less Systems Checkout	0.00	0.000	.18	.001
Clock Hours Avail. to Users	634.85	22.673	3369.29	22.313
less Idle Time	69.90	2.496	427.55	2.831
Clock Hours in Use	564.95	20.177	2941.74	19.482
% Available Hours Used	88.99 %		87.31 %	

CRAY-1 COMPUTER	FEBRUARY Total	FEBRUARY Day Avg.	FISCAL YTD Total	FISCAL YTD Day Avg.
Clock Hours in the Month	672.00	24.000	3624.00	24.000
less Scheduled PM	16.98	.606	77.33	.512
less Hardware Downtime	1.30	.046	29.03	.192
less Software Downtime	3.23	.115	9.40	.062
less Environmental Downtime	.33	.012	22.71	.150
less Operations Use	0.00	0.000	1.42	.009
less Other Causes	.97	.035	16.35	.108
Clock Hours Up	649.19	23.185	3467.76	22.965
less Systems Checkout	6.53	.233	27.21	.180
Clock Hours Avail. to Users	642.66	22.952	3440.55	22.785
less Idle Time	.64	.023	19.45	.129
Clock Hours in Use	642.02	22.929	3421.10	22.656
% Available Hours Used	99.90 %		99.43 %	

Computer Resources Allocated in February 1983

SCIENTIST	PROJECT TITLE	CCU		KCRU	
		Request	Alloc.	Request	Alloc.
James Richman Oregon State Univ.	Wind induced oceanic fronts	3.67	3.67	0.0	0.0
David Houghton Yun-Qi Ni Univ. of Wisconsin	CCM orography sensitivity	0.0	0.0	4.5	4.5
Thomas G. Kyle Univ. of Denver	Search for minor atmospheric gases	5.0	5.0	0.0	0.0
M. Dejnakarintra Stanford Univ.	Electrical coupling between the troposphere and ionosphere	4.5	4.5	0.0	0.0
Edward Benton Univ. of Colorado	Steady flow at top of earth's core	3.0	3.0	0.0	0.0
David Hathaway Sacramento Peak Obs.	Nonlinear interactions between convection rotation and shear flows	5.0	5.0	0.0	0.0
Philip Rasch Florida State Univ.	Normal mode initialization in tropical regions	5.0	5.0	0.0	0.0
S. I. Cheng Princeton Univ.	Computational study of turbulent spot	5.0	5.0	0.0	0.0

Note: Resources requested and granted may differ for several reasons.

1. During the processing of a request for computer time, the applicant may decide to switch from one machine to the other.

2. The applicant may not have requested the resources on the Control Data 7600 necessary for access to the CRAY-1.

3. A request may be supported at a lower level than requested because:

   a. It exceeds the 5-hour limit above which Panel review is required; or

   b. Reviewers consider the amount of time requested to be excessive.

TROUBLE/DESIGN REPORT
March, 1983

CRAY No. 112

TROUBLE: No error message issued by the CFT 1.10 compiler when attempts are made to concatenate character variables back into themselves.

EXAMPLE:

```
      CHARACTER*16 STRNG1,STRNG2,STRNG3
      DATA         STRNG1/'ABCDEFGHIJKLMNOP'/
C
C     Case  1:  Incorrect results achieved for I>7
C               when assignment is made to the same
C               character variable.
C
      WRITE(UNIT=*,FMT='()')
      DO 10 I = 1,16
      STRNG2 = '1234567890123456'
      STRNG2 = STRNG1(1:I) // STRNG2
      WRITE(UNIT=*,FMT='('' '',I2,2X,A)')  I, STRNG2
 10   CONTINUE
C
C     Case  2:  Correct results achieved when assign-
C               ment is made to another character
C               variable.
C
      WRITE(UNIT=*,FMT='()')
      DO 20 I = 1,16
      STRNG2 = '1234567890123456'
      STRNG3 = STRNG1(1:I) // STRNG2
      WRITE(UNIT=*,FMT='('' '',I2,2X,A)')  I, STRNG3
 20   CONTINUE
      END
\EOF
```

COMMENTS: The lefthand column below resulted from concatenating "STRNG1" and "STRNG2" back into "STRNG2". It appears to work correctly for I less than 8, however when I is greater than or equal to 8, the resultant string appears to be the concatenation of "STRNG1" with itself, which is not what was intended. Although the assigning of character positions back into themselves is not standard conforming, no error message was issued indicating that an illegal assignment was being attempted. The column on the right, which is correct, is the concatenation of "STRNG1" and "STRNG2" into "STRNG3".

```
1   A123456789012345     A123456789012345
          ...                  ...
6   ABCDEF1234567890     ABCDEF1234567890
7   ABCDEFG123456789     ABCDEFG123456789
8   ABCDEFGHABCDEFGH     ABCDEFGH12345678
9   ABCDEFGHIABCDEFG     ABCDEFGHI1234567
          ...                  ...
```

TEMPORARY SOLUTION: Do not concatenate character variables back into themselves.

ORIGINATOR: Herb Poppe.

TROUBLE/DESIGN REPORT
March, 1983

CRAY No. 113

TROUBLE: CFT 1.10 may not evaluate character relational expressions as expected.

EXAMPLE:

```
      CHARACTER FORM*11
      CHARACTER ITEST*9
      DATA ITEST/'FORMATTED'/
C
      FORM = 'FORMATTED'
C
      IF(FORM .EQ. ITEST)THEN
      WRITE(UNIT=6,FMT='('' FILE IS FORMATTED.'')')
      ELSE
      WRITE(UNIT=6,FMT='('' FILE IS NOT FORMATTED.'')')
      END IF
```

COMMENTS: When character variables of unequal length are used as operands in a character relational expression, and the operator used is either .EQ. or .NE., the shorter operand is supposed to be considered as if it were extended on the right with blanks to the length of the longer operand. In the above example, "FORM" and "ITEST" were declared character variables of length 9 and 11 respectively. Even though both variables were assigned the same value, 'FORMATTED', when compared in the relational expression, they evaluate as false. If "ITEST" is changed so that it is also length 11, they evaluate correctly, true or false, depending on the values assigned to "FORM" and "ITEST".

TEMPORARY SOLUTION: When using character variables in relational expressions, be sure they are of the same length.

ORIGINATOR: Herb Poppe.

TROUBLE/DESIGN REPORT
March, 1983

CRAY No. 114

TROUBLE: Expressions used in JCL Conditional Control Statements may not execute if the expressions use literal strings.

EXAMPLE:

```
 SET(G0 = FALSE)
 IF( ('STRNG1' .EQ. 'STRNG1') .AND. (
1    ('STRNG2' .EQ. 'STRNG2') .OR.
2    ('STRNG3' .EQ. 'STRNG3') ) )
 SET(G0 = TRUE)
 ENDIF.
 PRINT (G0)
 EXIT.
```

COMMENTS: The above Conditional Block Control statement is syntactically correct, but attempts to execute it cause the control statement processor (CSP) to abort with the following message:

```
0.0132    CSP     CS050 - EXPRESSION ERROR
0.0132    CSP           - MISSING PERIOD(.)
0.0132    CSP           - NEAR    3'.E
0.0132    ABORT   AB025 - USER PROGRAM REQUESTED ABORT
```

TEMPORARY SOLUTION: Temporarily store results of the sub-expression calculations in the local J0-J7 registers, and use these J registers as operands in the final expression.

```
SET(J1 = ('STRNG1' .EQ. 'STRNG1'))
SET(J2 = ('STRNG2' .EQ. 'STRNG2'))
SET(J3 = ('STRNG2' .EQ. 'STRNG2'))
IF(J1 .AND. (J2 .OR. J3))
      ....      ...
```

ORIGINATOR: Herb Poppe.

Non-Profit Organization
U.S. POSTAGE PAID
Boulder/Colorado
Permit No. 558

The Record

Issued by the
National Center for Atmospheric Research
Scientific Computing Division
P.O. Box 3000
Boulder, Colorado 80307

PRINTED MATTER

SUMMERS,BARBARA
Mesa Lab
NCAR
Boulder, CO (MAIL ROOM)