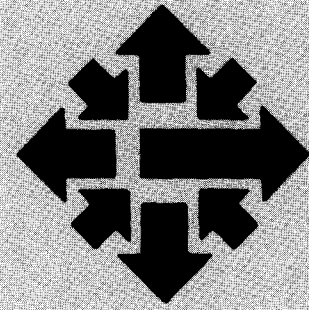


University Computer Center Newsletter

University of Minnesota
Twin Cities

September 1984
Volume 18, Number 9



Math and Statistics Roundup

SPSS VERSION 9.0

The long-awaited release of SPSS version 9.0 is at last on the CYBERS. It contains substantial improvements over the current (8.3) version:

- A new procedure, BOX-JENKINS, identifies, fits, and forecasts time series data using an ARIMA model.
- REGRESSION has been entirely redesigned. It now includes five equation-building methods; four missing value treatments; descriptive, bivariate, and other statistics; the ability to read or write related statistics and matrices to external files; and extended facilities for the analysis and graphic display of residuals. Since this new procedure is incompatible with the current REGRESSION, we call it NEW REGRESSION. The older REGRESSION of Version 8.3 remains available.
- CROSSTABS, BREAKDOWN, and REPORT run a lot faster than they did in Version 8.3: CROSSTABS is 40 to 50 percent faster; BREAKDOWN is 55 to 60 percent faster in general mode; and REPORT is 5 to 50 percent faster.
- A major update of MANOVA. We are adding an 86-page MANOVA section to the *SPSS Support Packet* that contains *only* the corrections and improvements. It is currently available as WRITEUP(SPSS=MANOVA).
- ONEWAY analysis of variance has improved multiple comparison range tests (including the graphic output of means), and allows the specification of a random effects, as well as a fixed-effects, model. It has also added several statistics (Welch's *w*, Brown-Forsythe *F**, Fisher's randomization test) that can test the equality of group means with less restrictive assumptions than the usual *F* test.

The new *spss 9.0 Update Manual* and the *spss Support Packet* will be available sometime this fall. You will be able to examine them in our Reference Room, 140 Experimental Engineering, or purchase them through the University bookstores. The *Update Manual* is currently available as WRITEUP(SPSS=MANUAL).

WRITEUP(SPSS) includes information about other on-line material.

To access the new version of SPSS, enter:

**FUTURE,SPSS.
SPSS,parameters.**

To use SPSS 9.0 interactively with SPSS/ONLINE, enter:

**FUTURE,SPSSONL.
SPSSONL.**

(Bruce Center)

LISREL VI ON THE CYBERS

A new version of LISREL, LISREL VI, is now available as FUTURE on the CYBERS. LISREL performs Joreskog Maximum Likelihood Factor Analysis (sometimes called Confirmatory Factor Analysis because a confidence interval of known probability may be computed for each factor) and analyzes linear structural relationships.

Some of the improvements in LISREL include:

- Automatic starting values for the parameters: you no longer need to provide them. LISREL's starting values are often satisfactory for final estimates as well. These initial estimates are obtained by a non-iterative four-step procedure, combining the methods of instrumental variables and ordinary least squares. Such initial estimates are consistent and usually unbiased.

- The ability to handle missing values.
- Generation of tetrachoric or polyserial correlations and analysis of a matrix of such correlations, which means LISREL can properly handle discrete and ordinal variables as well as continuous variables.
- A "modification index" for each fixed parameter that shows the extent to which these parameters, subject to an equality constraint, depart from this assumed equality. This is especially useful for improving the specification of the underlying LISREL model.
- Parameter estimates for all models by using ordinary least squares, two stage least squares, generalized least squares, instrumental variables, and/or maximum likelihood.
- Improved free format data input.
- Greatly expanded and much more readable printed output.

A version of LISREL VI will soon be available on the CRAY. The CRAY's million words of memory and faster execution will offer significant savings on medium to large problems.

The *LISREL VI User's Guide* by Joreskog and Sorbom (1984) provides LISREL VI documentation. You can examine a copy in our Reference Room, 140 Experimental Engineering. It is also available at the University bookstores.

To access LISREL VI, enter:

**FUTURE,LISREL.
LISREL. or LISREL,SIZE=nnn.**

By default, nnn is 10,000 words. You can increase this with the SIZE parameter on the LISREL control statement as shown.

(Bruce Center)

LINDO CHANGES

On August 2, we changed the LINDO linear programming package to solve several problems that resulted when variables were not initially set to zero in the previous version.

HARWELL SUBROUTINE LIBRARY ON CRAY

On August 6, we installed the Harwell Subroutine Library on the CRAY-1. It is a collection of several hundred FORTRAN subroutines for problems in numerical mathematics. To access the library, use the LIB parameter on the LDR control statement:

LDR(LIB = HARWELL).

Documentation on the library is available in a two-volume manual on reserve in the UCC Reference Room, 140 Experimental Engineering, (612) 373-7744.

ALGORITHMS FROM ACM TRANSACTIONS OF MATHEMATICAL SOFTWARE

We have received ACM-collected algorithms (originally published in

ACM Transactions of Mathematical Software) from IMSL over the past few years. Our collection begins with algorithm 493, published in March 1975, and goes through algorithm 613, published in March 1984. Algorithms 581-584 and 601-604 are not available at this time but will be added soon.

All these algorithms are available to CYBER users on the FETCH file CALGOPL. The list of the available algorithms and control statements to obtain them are listed in WRITEUP(CALGOPL).

The *ACM Transactions of Mathematical Software* is available in our Reference Room, 140 Experimental Engineering.

CRAY News

SUBMIT VIDEOTAPE AVAILABLE

UCC has produced a videotape that explains how to submit jobs to the CRAY-1 through the CYBER front-end. This 28-minute tape features Kurt Richards of our Client Services Group as instructor.

Topics covered in the tape include:

- batch access through the CYBER 845;
- syntax of the SUBMIT command;
- structure of the SUBMIT file;
- CYBER job status inquiry commands;
- a detailed description of some CRAY Operating System (COS) commands and parameters; and
- general job flow between machines.

The tape is available to our users in a variety of video formats. The cost is \$48 for 3/4" U-matic format, and \$42 for 1/2" VHS or Beta format. For information, or to place an order, call Jerry Stearns, (612) 376-8806.

Programming Languages

PASCAL VERSION 4.0

On Monday, September 3, the current version of Pascal on our CYBER systems will be Version 4.0. This change will fix several errors in the old version (4.J).

Relative Efficiency of CRAY FORTRAN Functions

UCC's Client Services group recently investigated three CRAY FORTRAN (CFT) implementation methods in order to find out which performed most efficiently on the CRAY supercomputer. We wrote in-line code, statement functions, and external functions, and used these three methods to implement boolean and numerical algorithms in CFT.

Our results showed that in-line coding and statement functions are roughly comparable in CPU efficiency and overhead costs. External functions, however, required far more CPU time and cost more than statement functions for our small sample functions. Given these test results, we recommend that, when possible, you use statement functions rather than external functions if execution efficiency is your main concern. This technique cannot substitute, however, for a efficient algorithm. Assuming that you have selected the best algorithm for your program and taken advantage of the CRAY's vectorizing capability, choosing a statement function can also improve the execution time of your program.

Definitions

"In-line coding" refers to normal FORTRAN source code that does not transfer control to any user-written subroutines, subprograms, or functions. In other words, this code appears in the place where it was written each time it is used.

"Statement functions" exist in almost all high-level and machine-level computer languages; some languages refer to them as "macros" or "procs." Statement functions are templates of parameterized code that are defined once within a program and later called any number of times. When you reference statement functions, you generate code in line at the point of reference, and the compiler optimizes this code. Keep in mind, however, the following constraints on FORTRAN statement functions:

- They must be written as single statements, but they may

```
STATEM1(A) = A**2 - 1.4*SIN(A)/2.  
STATEM2(B,C) = B**2*C - STATEM1(B)
```

A typical statement function reference to STATEM2 would be:
RESULT = STATEM2(.417, 1.392)

Exhibit 1. Nested statement function definition statements.

```
C  DEFINITION OF FUNCTION FUNCT1  
FUNCTION FUNCT1 (A)  
FUNCT1 = A**2 - 1.4*SIN(A)/2.  
END
```

```
C  
C  DEFINITION OF FUNCTION FUNCT2  
FUNCTION FUNCT2 (B,C)  
FUNCT2 = B**2*C - FUNCT1(B)  
END
```

A typical reference to external function FUNCT2 would be:
RESULT = FUNCT2(-3.1412, 0.0171)

Exhibit 2. Nested external function definitions.

reference other statement functions.

- They must appear in every module in which they are used.
- You must place them after declarative statements and before executable statements.

The example in Exhibit 1 illustrates two nested statement function definition statements in CFT. The definitions are nested because the second definition statement refers to the first.

"External functions" are implemented in FORTRAN as subprograms that are jumped to when referenced by a different subroutine or subprogram. They may reside on a system library. The example in Exhibit 2 illustrates nested external function definitions that perform the computation equivalent to that in our first example.

Results

In our benchmarks, we compared triply nested function calls for statement functions and external functions, and wrote equivalent in-line code.

Our results show that statement functions and in-line coding compare in CPU efficiency and overhead within 40 percent. Any variance is due to CFT instruction and register scheduling or poor coding techniques.

We observed that external functions had from 78 percent to several times more overhead compared to statement functions. (Remember, however, the small size of our sample functions.)

High overhead costs in external function calling, both on the "called" and the "calling" sides (such as parameter processing, register saves and restores, etc.), increased considerably with each nested reference.

Cautions

Of course, you must consider more than sheer speed when you decide how to implement algorithms in FORTRAN. From the perspectives of good programming practice and minimized software support and maintenance costs (often higher in human than in machine dollars), both statement functions and

external functions are superior to in-line coding. Because you write statement function definitions only once for every module in which they are used, they are easy to locate, easy to document, and simple to check, change, or debug. Similarly, you only write an external function once. This is not true for in-line coding.

And, of course, in the overall process of creating efficient CRAY FORTRAN programs, other considerations such as vectorization of DO-loops and selection of the best algorithm come into play. Appendix H.1 of the *M77 Reference Manual* describes some of these considerations.

Although the next FORTRAN international standard (ANSI FORTRAN 88) is not yet published, we expect it to include a form of internal function very similar to today's statement function. In addition, statement functions are portable: they are accepted by almost all FORTRAN compilers today.

Additional test

We also included a CAL (Cray Assembly Language) subroutine in the benchmark tests to compare CAL overhead with CFT external function overhead. We were surprised to discover that external function overhead was 55 percent greater than CAL subroutine call overhead. (These subroutines simply retrieved and returned a passed parameter.) Evidently CAL subroutine calls, and CAL code as the ultimate CRAY language for driving the CRAY-1 toward its 140 megaflop peak, are the optimal techniques for sheer machine speed. But most of us do not have the time to write CAL routines for the 20 percent of our code that typically performs 80 percent of our processing.

So, as our benchmark tests show, statement functions, where applicable, are the next best implementation. (Tom Kovarik)

Fall Quarter Short Course Schedule

INTRODUCTORY COURSES

PRICES: U. student \$10, U. staff \$20, Non-University \$30

- 010 Introduction to Computer Terms (Jerry Larson)
Oct 1-5 (MWF) 3:15-5 p.m.
- 020 A Taste of Computing (staff)
Oct 8-12 (M-F) 3:15-5 p.m.
- 040 Introduction to VAX/VMS (Bryan Senn and Jerry Stearns)
Oct 9-23 (TTh) 2:15-4 p.m.
- 050 NOS (CYBER operating system) (Kurt Richards and Jon Jamsa)
Oct 15-31 (MWF) 3:15-5 p.m.
- 080 Introduction to Micros: CP/M (Simin Hickman)
Oct 15-18 (MTTh) 3:15-5 p.m.
- 090 Introduction to Micros: MS-DOS (Simin Hickman)
Oct 15-19 (MWF) 3:15-5 p.m.
- 130 XEDIT (section 1) (Tom Kovarik)
Nov 5-9 (MTThF) 1:15-2:30 p.m.
- 135 XEDIT (section 2) (Tom Kovarik)
Nov 5-9 (MTThF) 2:45-4 p.m.
- 150 Introduction to the CRAY-1 and COS (Kurt Richards)
Nov 12-16 (MWF) 3:15-5 p.m.

ADVANCED COURSES

PRICES: U. student \$20-30, U. staff \$30-50, Non-University \$50-100

- 670 CRAY FORTRAN Vectorizing (Tom Kovarik)
Oct 1-12 (MWF) 2:15-4 p.m. \$25-\$35-\$60
- 520 Lotus 1-2-3 (section 1) (Joanne Bergman)
Oct 2-4 (TTh) 2:15-4 p.m. \$25-\$35-\$60
- 525 Lotus 1-2-3 (section 2) (Joanne Bergman)
Nov 13-15 (TTh) 2:15-4 p.m. \$25-\$35-\$60
- 650 COMPASS (assembly) (John Larsen and Tom Kovarik)
Oct 17-26 (TWTh) 2:15-4 p.m. \$30-\$45-\$70
- 580 Scribe (formatting program) (Elaine Collins)
Oct 29-Nov 9 (MWF) 3:15-5 p.m. \$25-\$35-\$60
- 640 Beginning Pascal (Peter Oberg)
Nov 5-16 (MWF) 3:15-5 p.m. \$25-\$40-\$60
- 530 SPSS (statistics package) (Bruce Center)
Nov 5-9 (MWF) 2:15-4 p.m. \$20-\$30-\$55
- 570 TELL-A-GRAF Graphics (Bryan Senn)
Nov 12-21 (MW) 2:15-5 p.m. \$30-\$45-\$75

REGISTRATION: You can register at the UCC Reference Room, 140 Experimental Engineering (hours 8 a.m. to 4:30 p.m., Monday through Friday). A self-service terminal for registration is located inside the Reference Room. We accept mail registrations for an additional \$1 fee per class. The deadline for registration is Reference Room closing on the last working day *before* the class begins. You may pay course fees with cash, check, University journal vouchers, or you can charge them to your non-instructional UCC user account.

REFUND POLICY: No refunds are made after the class begins. Refunds are made in the same form as the fee was paid, i.e., check, journal voucher, UCC account credit.

If you have questions about short courses or about registration, call Jerry Stearns, 376-8806, or see WRITEUP(CLASSES) on the MERITSS or CYBER CA systems.

PHONE NUMBERS

Access:	Engineering Services	376-1023, 376-8153
CYBER(CA)—10, 30 cps	Equipment Purchase/Information	376-8153
—120 cps	Experimental Engineering I/O	373-4596
MERITSS(ME)—10, 30 cps	Graphics Software	376-5592
—120 cps	HELP-line	376-5592
VAX/VMS(VA)—(autobaud)	7 a.m.-7 p.m., Monday-Friday	
Budgets	HOURS-line (recorded message)	373-4927
Computer-Aided Instruction	Information, Experimental Engineering	373-4360
Computer Hours (recorded message)	Information, Lauderdale	373-4912
Consulting	Instructional Labs	376-2703
HELP-line	Instructional Services	373-7745
7 a.m.-7 p.m., Monday-Friday	Lauderdale Computer Room	373-4940
Business Data Products	Lauderdale Services	373-4995
1-3 p.m., Monday-Friday	Lauderdale Services Manager	373-7538
Statistics Packages	Lauderdale Users' Room	373-4921
376-1761	MECC Liaison	373-7745
1-2 p.m., Monday-Friday	Newsletter Subscription	376-1491
Data Bases	Permanent File Restoration	376-5605
376-1761	Professional Services Division (PSD)	376-1764
10-11 a.m., Monday-Friday	Project Assistance	376-1764
Microcomputers	Reference Room	373-7744
376-4276	Remote Batch (RJE) Services	376-2703
9:30 a.m.-noon and 1:30-4 p.m., Monday-Friday	Short Courses	376-8806
Humanities Computing	Shuttle Bus Service	376-3068
376-2944	System Status (recorded message)	373-4927
2-3 p.m., Tuesday & Wednesday	Tape Librarian: see Lauderdale Services	
Text Processing	Text Processing Services	376-2943
376-2944	User Accounts	373-4548
1-4 p.m., Monday-Friday		
Contract Programming		
376-1764		
Data Base Applications		
376-1764		
EDUNET Liaison		
373-5780		

OPERATING HOURS

	CYBER (CA)	Low rate	CRAY (CR)	MERITSS (ME)	VAX (VA)
M-F	7 a.m. - 4 a.m.	8 p.m. - 4 a.m.	7 a.m. - midnight	7:45 a.m. - 3:30 a.m.	8 a.m. - 6 a.m.
Sat	4 a.m. - 5:15 p.m.	4 a.m. - 5:15 p.m.	7 a.m. - 5 p.m.	7:45 a.m. - 3:30 a.m.	24 hours
Sun	4 p.m. - 1 a.m.	4 p.m. - 1 a.m.	4 p.m. - midnight	4 p.m. - 3:30 a.m.	24 hours

PUBLIC LABS—TWIN CITIES CAMPUS

Location	Batch	Interactive	Micro	Location	Batch	Interactive	Micro
<i>East Bank</i>				<i>West Bank</i>			
Arch 160		X	X	BlegH 25		*	
CentH		X		BlegH 90	X		
ComH		X		BlegH 140		X	
DiehlH 270, 207		X		MdbH		X	
EltH 121, 125		X		OMWL 2		X	
EltH N640	X			SocSci 167			X
Exp Eng 130		*					
FolH 14, 14a	X	X*	X	<i>St. Paul</i>			
LindH 26	X	X		BaH		X	
MechE 308		X		ClaOff 125	X	X	
Physics 69		*					
SanfH		X					
TerrH		X					
Vinch 4		X					
WaLib B9		X					

* Research cluster; access to CYBER CA and VAX/VMS
X in interactive column indicates access to MERITSS

Contents

Math and Statistics Roundup

SPSS VERSION 9.0	71
LISREL VI ON THE CYBERS	71
LINDO CHANGES	72
HARWELL SUBROUTINE LIBRARY ON CRAY ..	72
ALGORITHMS FROM ACM TRANSACTIONS OF MATHEMATICAL SOFTWARE	72

CRAY News

SUBMIT VIDEOTAPE AVAILABLE	72
Programming Languages	
PASCAL VERSION 4.0	72
Relative Efficiency of CRAY FORTRAN Functions	73
Fall Quarter Short Course Schedule	74

Michael M. Skow, Acting Director

The *UCC Newsletter* is published monthly by the University Computer Center. Deadline for articles is the 10th of the month preceding publication; deadline for short announcements is the 15th. The *Newsletter* is edited and coded for typesetting at the Computer Center, then typeset on a Linotron 202 and printed at the University of Minnesota's Printing and Graphics Arts Department.

Comments, suggestions, articles, and announcements should be directed to the editor, 227 Experimental Engineering, (612) 376-1491.

The University of Minnesota adheres to the principle that all persons should have equal opportunity and access to facilities in any phase of University activity without regard to race, creed, color, sex, national origin, or handicap.

Copyright 1984 University of Minnesota. Permission to copy is hereby granted, provided that proper acknowledgement is given.

Nonprofit Org.
U.S. Postage
PAID
Minneapolis, Mn.
Permit No. 155

University Computer Center Newsletter

Technical Publications
227 Experimental Engineering
University of Minnesota
208 Union Street SE
Minneapolis, Minnesota 55455

UNIVERSITY ARCHIVES
10 WALTER LIBRARY
UNIVERSITY OF MINNESOTA
EAST BANK
117 PLEASANT STREET SE
MINNEAPOLIS MN 55455