

MANAGEMENT APPLICATIONS

in

DATA



MANAGEMENT APPLICATIONS in **APL**

Edited by
EPHRAIM R. McLEAN
and
JAMES W. SCHENCK



Graduate School
of Management
UCLA

MANAGEMENT APPLICATIONS

in

APL

Edited by

EPHRAIM R. McLEAN

and

JAMES W. SCHENCK



Graduate School
of Management
UCLA

© January 1981
E.R. McLean

COMPUTERS AND INFORMATION SYSTEMS
RESEARCH PROGRAM
Graduate School of Management
University of California
Los Angeles, California 90024

TABLE OF CONTENTS

<u>TITLE</u>	<u>PAGE</u>
1.0 Program Overview	1
2.0 Program Participants	7
3.0 Participants' APL Applications	9
3.1 Armitage, Howard "The Accounting Cycle"	11
3.2 Baker, Dennis "Simulation of the Birth and Death of Firms in a Competitive Industry"	21
3.3 Brown, Lawrence D. "Single Asset Depreciation Methods"	41
3.4 Case, Fred "Real Estate Valuation Using the Direct Capitalization Method"	47
3.5 Chai, John "Simulation of Random Samples and Sampling Distributions of Various Statistics"	55
3.6 Dauner, Jack "Setting the Size of the Sales Force"	67
3.7 Harpell, John L. Jr. "Simulation of a 3-Stage Order-Shipment System"	71
3.8 Hickman, Edgar "Multivariate T-Square Test".	81
3.9 Mason, R. Harold "Circulation of Disability Settlement Awards"	87
3.10 Miller, Ronald D. "Linear Programming with Mixed Constraints"	95
3.11 Park, John "Interactive Effects of the Marginal Propensity to Consume and the Investment Accelerator on the Production of Income"	107
3.12 Sanford, Donald G. "Network Analysis".	115

3.13	Schriber, Thomas "Interactive Optimization of a Production System"	127
3.14	Sprague, Ralph H. Jr. "Interactive Inventory Simulator"	155
3.15	Stricklin, Wilma "Financial Ratio Calculations"	173
3.16	Thielges, Bernard "Computation of a Price Index by the Method of the Weighted Geometric Mean"	177
4.0	UCLA APL Library Functions	185
5.0	Outline of the Course.	201
5.1	Class Schedule and Reading Assignments	203
5.2	Class Handouts	213
5.2.1	A Guide to Writing APL Programs	215
5.2.2	Signing-On to the UCLA APL System	221
5.2.3	An APL Bibliography	227
5.2.4	"Things Go Better with APL" Claire Stegmann, <u>Think</u> , May/June 1975	255
5.3	APL Programming Assignment Instructions.	261
6.0	Program Administration and Promotional Materials	265
6.1	Program Announcement Brochure.	267
6.2	Invitation Letter to Business School Deans	273
6.3	Publicity.	277
6.4	Application Questionnaire and Cover Letter	281
6.5	Letter of Acceptance	291
6.6	Program Evaluation Questionnaire	297
6.7	Post Conference Follow-Up Letter	303

1.0 PROGRAM OVERVIEW

In recognition of the growing importance of interactive computing in support of management education, the IBM Corporation agreed to sponsor a two-week program on the subject in the summer of 1975. Modelled after the highly successful program at the University of Michigan, which introduced business school educators to the use of FORTRAN, this program was aimed at accomplishing the same goal, this time with the interactive computer language APL. Approximately two dozen management school professors were to be invited to come to the Graduate School of Management (GSM) at UCLA and to be involved in an intensive learning experience.

Basically, the class had two objectives. The first was to gain some familiarity and confidence in the use of the computer and APL. The second was to write a substantial program in APL that would be useful in teaching each faculty member's own discipline (e.g., accounting, finance, production, etc.). The programs that comprise a major part of this book are the results of this second objective.

Naturally, some of these programs are more comprehensive than others; and, in fact, there were a few of the participants that set such ambitious goals for themselves that they were unable to complete their programs in the two-week time period. Then, when they returned home, the press of other activities crowded out their ability to finish their programs. This explains why a few of the participants' programs are missing from this book. However, given the fact that most of the faculty members had never written a computer program before beginning the course, the programs that do appear in the following pages are an impressive testimonial to their hard work and commitment.

Because a great deal of planning and arrangements went into the conduct of the course, it might be of interest to the reader to have a chronology of the events leading up to its offering.

Upon receiving notification of IBM's decision to support the program on Interactive Computing for Management Educators in December of 1974, plans were begun immediately for the conduct of the program. In January of 1975, announcement brochures were prepared (See Section 6.1), describing the program in detail and calling for interested parties to apply for

admission. The contents of this brochure were patterned very closely after the description of the program included in the original grant request proposal sent to IBM. This brochure was then sent to the deans of all of the member schools of the American Assembly of Collegiate Schools of Business (AACSB). It was accompanied by an individually typed cover letter (Section 6.2) which was personally signed by GSM Dean Harold Williams. This more personalized approach was taken to assure that the letter, and the brochure, received a careful reading. In all, slightly over 500 letters were mailed out, each one containing at least two brochures (so that the dean receiving them could circulate them to two groups within the school at the same time).

The response to this mailing -- and to some publicity in the Computing Newsletter for Schools of Business (Section 6.3) -- was nothing short of phenomenal. By the time of the March 15th deadline stated in the brochure, 197 requests for applications had been received for the 24 openings in the program. Several more requests were received after the deadline, particularly from abroad, which are not included in this total.

To provide a basis for selection -- and, quite frankly, to weed out the idle curious -- a detailed six-page questionnaire (Section 6.4) was prepared and mailed to each of those expressing interest in attending. From this mailing, 105 completed questionnaires were received, representing a wide cross-section of schools, geographical distribution, and academic ranks. With only one opening for every four applicants, the selection process was quite difficult, and a second or third offering could have easily been filled with qualified individuals. Of particular interest is the fact that, of those applying, over 60 came from schools that either already had APL installed and running on their campus computer or were planning its installation within the coming year -- an impressive indication of the spread of APL.

After much careful study, 24 individuals were selected and notified (See Program Participants, Section 2.0). Those who could not be accommodated were also notified at the same time, thanking them for their interest and promising to contact them personally if another offering were to be held.

These 24 faculty members were from schools large and small, public and private. Two were from Canada and two were women (one a nun). All academic ranks were represented, and most areas of specialization within the field of business (e.g., accounting, economics, finance, marketing, production, etc.). Also of interest is the fact that while many people consider APL to be a uniquely IBM product, nearly half of the schools (9 out of 21) were running APL on other than IBM equipment.

Upon arrival, the participants were housed in the Holiday Inn of Westwood which allowed them the option of either taking their evening meals at the motel or in one of the many restaurants in nearby Westwood Village. This arrangement proved to be particularly well received, as was the opening reception and dinner which served as an important get-acquainted function. Breakfasts were at the Holiday Inn and lunches at the UCLA Faculty Center. Because the first Friday of the two-week program fell on July 4th, an Independence Day barbecue was held; and, with the exception of one or two individuals who had made previous commitments, it was attended by all of the participants and by the program faculty and staff.

In terms of the conduct of the course itself, the program was split into two groups, A and B. Although it was hoped that all of the participants would be of approximately the same skill level and background, the vagaries of selection were such that some of the attendees were able to move through the course material faster than others. Thus Group B became somewhat the "fast track" and Group A, the "slow track." The reading assignments, class schedule, class handouts, and programming assignments are shown in Sections 5.1, 5.2, and 5.3.

As might be expected, a certain amount of anxiety was caused by the requirement that each person was expected to produce a working, documented program of reasonable complexity by the end of the two weeks. These programs were of their own choosing, and the finished results are published in this monograph. Most of the participants were able to accomplish this task, although a few individuals undertook projects which were really too ambitious. However, even for those who failed to complete their programs, it is felt that the exercise still contributed markedly to their understanding of APL

and interactive computing.

At the end of the program, a three-page questionnaire (Section 6.6) was passed out to each of the participants, asking them their comments and opinions on various parts of the program. So that frank reactions could be obtained, they were not required to sign the questionnaire or otherwise identify themselves; however, most chose to do so. Fourteen questionnaires were returned (and others gave their views orally) and the general reactions were highly favorable. On the key question "Your overall evaluation of the program as a whole -- (Poor) 1 2 3 4 5 6 7 (Outstanding)," the median response was 6. The following were typical comments:

"Great! I feel that I was really pressed into learning a great deal in a short time. I am enthused about APL and will begin to use it at my school."

"Outstanding."

"The program was extremely worth while to me. While the work was ample, I feel that's what we were here for so there should be no complaints in that regard. The program was well structured and very well presented."

Of particular interest are the comments (quoted in full) of Dr. Thomas Schriber of the University of Michigan, who for several years conducted the previously mentioned IBM-sponsored program on FORTRAN:

"Excellent, on a variety of counts:

1. Personal-professional (total immersion is the way to get on top of a highly detailed subject like this)
2. Personal-personal (a pleasure to work hard with intelligent, dedicated people, and to get to know these people)
3. Home curriculum (will be upgraded as a result of this course -- APL will get into our curriculum faster than otherwise)
4. Schools of Business overall (will get into the APL generation much sooner than otherwise; why not lead instead of lag?)"

Naturally, there were some specific suggestions about particular aspects of the course, the accommodations, or other administrative details, but they were mostly minor. By and large, both the written and oral comments were extremely positive.

As a final wrap-up to the program, a follow-up letter (Section 6.7) was sent to all participants, thanking them for their diligence and offering our assistance to those who had not yet completed their programs.

In the preparation of this monograph, every effort has been made to ensure that the APL programs, as submitted by the participants, are correct and "bug" free and will work as the sample output indicates. However, it is possible that some errors do still exist. If, in using the programs, such errors are encountered, it would be appreciated if the author of the program would be notified directly, with a copy of the correspondence sent to me here at UCLA.

In closing, I would like to acknowledge and thank the several individuals who made this program a success. On the instructional staff, Dr. Kenneth Siler, a fellow member of the GSM faculty, shared equally in the teaching load. James Schenck, a member of the staff of the GSM computing facility, assisted the participants in a number of ways and helped prepare the final versions of the programs for inclusion here. Dr. William Bergquist of IBM also was in attendance during much of the program, first as an observer and later as an assistant in the computer lab. Finally, Bill DeLone and Janice Durbin, graduate students in GSM helped in the preparation of the final manuscript. To all of these individuals, I extend my thanks and deep appreciation.

Ephraim R. McLean

June 1980
Los Angeles

PARTICIPANTS IN THE CONFERENCE ON
INTERACTIVE COMPUTING
FOR
MANAGEMENT EDUCATORS
June 29 - July 11, 1975
GRADUATE SCHOOL OF MANAGEMENT
UNIVERSITY OF CALIFORNIA, LOS ANGELES

Howard M. Armitage
Assistant Professor of Business
Department of Business
Wilfrid Laurier University
Waterloo, Ontario, Canada N2L 3C5

Fred E. Case
Professor of Human Land Economics
Graduate School of Management
University of California
Los Angeles, California 90024

Dennis M. Baker
Associate Professor of Accounting
and Information Systems
Faculty of Business Administration and
Commerce
University of Alberta
486 Central Academic
Edmonton, Alberta, Canada T6K 1M6

John J. Chai
Associate Professor of Quantitative
Methods
Department of Quantitative Methods
School of Management
Syracuse University
Syracuse, New York 13210

Sister Mary Seraphine Bennett
Professor of Quantitative Studies
School of Business
University of Baltimore
1420 North Charles Street
Baltimore, Maryland 21201

Jack R. Dauner
Associate Professor of Marketing
Department of Marketing
University of Akron
Akron, Ohio 44325

Lawrence D. Brown
Lecturer in Accounting and Economics
School of Management
State University of New York at
Binghamton
Binghamton, New York 13901

Richard A. Elnicki
Associate Professor of Management
College of Business Administration
University of Florida
Gainesville, Florida 32605

Elwood S. Buffa
Professor of Operations Research and
Operations Management
Graduate School of Management
University of California
Los Angeles, California 90024

John L. Harpell, Jr.
Assistant Professor of Management
College of Business and Economics
West Virginia University
Morgantown, West Virginia 26506

Kenneth J. Carey
Assistant Professor of Finance
Department of Finance
701 Business Administration Building
The Pennsylvania State University
University Park, Pennsylvania 16802

Edgar P. Hickman
Professor and Program Director
of Management Science
College of Business Administration
University of South Carolina
Columbia, South Carolina 29208

R. Harold Mason
Professor, International Business and
Business Policy
Graduate School of Management
University of California
Los Angeles, California 90403

Thomas J. Schriber
Professor of Management Science
Graduate School of Business
Administration
University of Michigan
Ann Arbor, Michigan 48104

Ronald D. Miller
Assistant Dean
College of Business
Governors State University
Park Forest South, Illinois 60466

John P. Shelton
Professor of Finance
Graduate School of Management
University of California
Los Angeles, California 90024

William H. Murphy
Chairman, Business Administration
South Campus
Salem State College
Salem, Massachusetts 01970

Barry W. Sheridan
Assistant Professor of Business
Division of Business
Ithaca College
1402 West Tower
Ithaca, New York 14850

John Park
Chairman, Department of Economics
and Business Administration
Frostburg State College
Frostburg, Maryland 21532

Ralph H. Sprague, Jr.
Associate Professor of Quantitative
Business Analysis
College of Business
University of Hawaii
2404 Maile Way
Honolulu, Hawaii 96822

Avinoam Perry
Assistant Professor of Decision Sciences
Decision Sciences Department
Graduate School of Management
Northwestern University
Evanston, Illinois 60201

Wilma D. Stricklin
Professor of Management
College of Business Administration
Northern Arizona University
Flagstaff, Arizona 86001

Donald L. Rogoff
Professor of Accounting and Finance
School of Business and Management
Pepperdine University
Malibu, California 90265

Bernard A. Thielges
Assistant Professor of Management
College of Business Administration
Marquette University
606 North 13th St.
Milwaukee, Wisconsin 53233

Donald G. Sanford
Assistant Professor and Assistant Dean
College of Business and Public Administration
Florida Atlantic University
Boca Raton, Florida 33432

3.0

PARTICIPANTS' APL APPLICATIONS

THE ACCOUNTING CYCLE:
An Interactive Approach

Howard M. Armitage
Assistant Professor in Accounting
School of Business and Economics
Wilfrid Laurier University
Waterloo, Ontario
Canada

DESCRIPTION

Introduction

In many respects, accounting is a highly repetitive process. In fact, the system that a student learns to follow at the outset of his accounting education consists mainly of the following steps:

- Journalizing
- Posting
- Trial Balance Preparation
- Adjustments
- Adjusted Trial Balance Preparation
- Statement Preparation
- Closing Entries

This sequence of steps is known as the accounting cycle and the user learns very early that the effect of any set of transactions on the final statements can be determined by following the above format.

Once this system has been illustrated and used and the student has grasped the repetitive nature of the cycle, further work with hand written accounting systems seems time consuming and only marginally useful. It is clearly advantageous to introduce a student to a system which, once the original entries have been made, can take over the repetitive tasks of posting,

ensuring equality, classifying items into revenue and capital accounts, and preparing final reports. This program has been written in response to such a need.

Instructions for Use

The user should specify his account names by using the function *CNAMES*. He must also establish a numbered set of accounts to correspond to the account titles. This should be done by using the function, *INIT*. *INIT*, when executed, requests that you enter the account numbers to be used. You should respond by typing the desired accounts separated by at least one blank. For example,

101 102 103 104 201 202 203 301 etc.

If you are unable to fit all the account numbers desired on a single line, just enter what will fit on the line and then strike the return key. The *INIT* function will then ask if you have any more accounts to enter. If you answer *YES*, *INIT* will request that you enter the remaining group. *INIT* will continue requesting more accounts until you respond *NO* to the question, *ANY MORE?* At this time, the two needed global variables will be created.

For each entry, the user will be asked to enter an account number, debit or credit, and dollar amount. After each transaction, the equality of the input can be checked by typing *NEXT*. If the debits do not equal the credits, the system will respond by printing out the last transaction, indicating where the error lies. When the transactions have been completed, a final equality check can be made by typing *OUT*.

The program will then take over and print out an UNADJUSTED TRIAL BALANCE. The user will be asked if he has any adjustments to make, following which the system will print out an ADJUSTED TRIAL BALANCE, INCOME STATEMENT, and BALANCE SHEET for the period

covered by the transactions. The revenue and expense accounts will automatically be set back to zero.

The program can be used for illustrating the accounts of a service, merchandising, or manufacturing enterprise whose inventory accounts are kept on a perpetual basis. With slight modifications the system can accommodate periodic inventory systems.

At the end of each iteration, an updated balance sheet will be printed out. Should the user wish to start anew, he can employ the function *ZERO* which will reduce the balances in all his accounts to zero. The user can also begin the cycle with any opening balance he wishes by debiting the asset accounts by the desired amount and similarly crediting the liability and equity accounts. When the command *OUT* is typed, the system will return the desired opening position.

In the following section is an example of program output which has resulted from a small number of transaction data. The amount of detail produced in these reports depends solely on the number of accounts set up and the extent to which these accounts are utilized during the accounting period.

In addition to the functions listed in the Program Listing Section, the program also utilizes a number of public functions. Specifically these are as follows: *AKI*, *AYN*, *CENTER*, *COLNAMES*, *ΔFMT*, and *IPI*.

PROGRAM USAGE

ACT
DO YOU NEED INSTRUCTIONS Y
ENTER THE ACCOUNT NUMBER, THE DEBIT OR CREDIT, AND THE AMOUNT FOR EACH
PART OF A TRANSACTION.

WHEN YOU HAVE COMPLETED THE TRANSACTION ENTER NEXT
WHEN YOU HAVE COMPLETED ALL THE TRANSACTIONS ENTER OUT
THE SYSTEM WILL RESPOND BY PRINTING OUT A TRIAL BALANCE.

YOU WILL BE ASKED IF YOU HAVE ANY ADJUSTMENTS TO MAKE.
FOLLOWING THESE ADJUSTMENTS, AN ADJUSTED TRIAL BALANCE, INCOME STATEMENT, AND
BALANCE WILL BE PRINTED.

IF YOU WISH TO STOP THE PROGRAM ENTER STOP UNDER ACCOUNT NUMBER.
ACCOUNT NUMBER?

□:

102
DEBIT OR CREDIT?

D
AMOUNT?

□:

3000
ACCOUNT NUMBER?

□:

201
DEBIT OR CREDIT?

C
AMOUNT?

□:

2000
ACCOUNT NUMBER?

□:

OUT
DEBITS DO NOT EQUAL CREDITS
ACCT DB CR

102	3000	0
201	0	2000

ACCOUNT NUMBER?

□:

201
DEBIT OR CREDIT?

C
AMOUNT?

□:

1000
ACCOUNT NUMBER?

□:

OUT
UNADJUSTED TRIAL BALANCE

UNADJUSTED TRIAL BALANCE

ACCTS	DEBITS	CREDITS
101	16900.00	0.00
102	12000.00	0.00
103	5000.00	0.00
104	10000.00	0.00
105	0.00	0.00
201	0.00	11000.00
202	0.00	400.00
203	0.00	100.00
204	0.00	8000.00
205	0.00	20000.00
206	0.00	3400.00
301	0.00	11000.00
401	1000.00	0.00
402	100.00	0.00
403	6000.00	0.00
404	500.00	0.00
405	0.00	0.00
<hr/> 0	<hr/> 51500.00	<hr/> 53900.00

DO YOU HAVE ANY ADJUSTMENTS TO MAKE? N
DO YOU WISH A PRINT OUT OF THE ADJUSTED TRIAL BALANCE? N

→

PLEASE ANSWER YES OR NO: N
DO YOU WISH A PRINT OUT OF THE ADJUSTED TRIAL BALANCE? Y

INCOME STATEMENT

REVENUES	
SALES REVENUE	11000.00
	<hr/>
	11000.00
EXPENSES	
DEPRECIATION EXPENSE	1000.00
INTEREST EXPENSE	100.00
SUPPLIES EXPENSE	6000.00
SALARIES EXPENSE	500.00
	<hr/>
	7600.00
NET INCOME	<hr/>
	3400.00
	<hr/>

BALANCE SHEET

ASSETS	
CASH	16900.00
ACCOUNTS RECEIVABLE	12000.00
SUPPLIES	5000.00
OFFICE AND EQUIPMENT	10000.00
	<hr/>
	43900.00
LIABILITIES AND OWNERS EQUITY	
ACCOUNTS PAYABLE	11000.00
SALARIES PAYABLE	400.00
INTEREST PAYABLE	100.00
LONG TERM LIABILITIES	8000.00
CONTRIBUTED CAPITAL	20000.00
RETAINED EARNINGS	3400.00
	<hr/>
	42900.00
	<hr/>

PROGRAM DESCRIPTION

```

▽ ACT
[1]  R TO INITIALIZE VARIABLES
[2]  R THIS FUNCTION REQUIRES THE FOLLOWING FUNCTIONS:
[3]  R AYN, CENTER, COLNAMES, ΔFMT, AND INC
[4]  NEXT+5*7
[5]  STOP+10*10
[6]  OUT+7*5
[7]  →LO1×1~AYN 'DO YOU NEED INSTRUCTIONS'
[8]  'ENTER THE ACCOUNT NUMBER, THE DEBIT OR CREDIT, AND THE AMOUNT FOR EACH'
[9]  'PART OF A TRANSACTION.'
[10] 'WHEN YOU HAVE COMPLETED THE TRANSACTION ENTER      NEXT'
[11] 'WHEN YOU HAVE COMPLETED ALL THE TRANSACTIONS ENTER      OUT'
[12] 'THE SYSTEM WILL RESPOND BY PRINTING OUT A TRIAL BALANCE.'
[13] ''
[14] 'YOU WILL BE ASKED IF YOU HAVE ANY ADJUSTMENTS TO MAKE.'
[15] 'FOLLOWING THESE ADJUSTMENTS, AN ADJUSTED TRIAL BALANCE, INCOME STATEMENT
      , AND'
[16] 'BALANCE WILL BE PRINTED.'
[17] 'IF YOU WISH TO STOP THE PROGRAM ENTER      STOP      UNDER ACCOUNT NUMBER.
      '
[18] LO1:
[19] UN←'UN'
[20] START:TEMPM←TEMP+10
[21] L1:'ACCOUNT NUMBER?'
[22] ACTN←□
[23] R TO CHECK IF (A)THE TRANSACTION HAS BEEN COMPLETED, (B)THE CYCLE HAS BEEN
[24] R COMPLETED, (C)THE EXIT SYMBOL HAS BEEN CALLED, AND (D)A VALID ACCOUNT
[25] R NUMBER HAS BEEN ENTERED.
[26] →AGAIN×1ACTN=NEXT
[27] →FINISH×1ACTN=OUT
[28] →0×1ACTN=STOP
[29] →L2×1(ρMATRIX[;1])>IN←MATRIX[;1]1ACTN
[30] 'INVALID ACCOUNT NUMBER'
[31] →L1
[32] L2:'DEBIT OR CREDIT?'
[33] CHG←□
[34] R TO CHECK FOR DEBIT OR CREDIT RESPONSES.
[35] →L3×1√/WW+( 'D'=1+CHG ), ( 'C'=1+CHG )
[36] →L2
[37] L3:'AMOUNT?'
[38] AMT←,□
[39] R TO ENSURE THAT THE VARIABLES NEXT, STOP, AND OUT ARE NOT EXECUTED AND THA
      T THE INPUT IS A SINGLE ENTRY.
[40] →L3×1^/NEXT=AMT
[41] →L3×1^/OUT=AMT
[42] →L3×1^/STOP=AMT
[43] →L4×11=ρAMT
[44] 'ENTER ONLY ONE NUMBER'
[45] →L3
[46] L4:TEMP←TEMP,ACTN,WW×AMT
[47] R ENTERS THE DEBITS AND CREDITS AGAINST THEIR APPROPRIATE ACCOUNT NUMBERS.
[48] MATRIX[IN; 2 3]+MATRIX[IN; 2 3]+WW×AMT
[49] →L1

```

```

[50] * CHECKS FOR EQUALITY OF DEBITS AND CREDITS ON COMPLETION OF THE TRANSACTIONS.
[51] AGAIN:TEMPM+(((PTEMP)÷3),3)PTEMP
[52] →START×\0=-/1+÷TEMPM
[53] 'DEBITS DO NOT EQUAL CREDITS'
[54] 'ACCT DB CR '
[55] TEMPM
[56] →L1
[57] * CHECKS FOR EQUALITY OF DEBITS AND CREDITS ON COMPLETION OF THE CYCLE.
[58] FINISH:TEMPM+(((PTEMP)÷3),3)PTEMP
[59] →COMPLETE×\0=-/1+÷TEMPM
[60] →AGAIN
[61] * PRINTS OUT UNADJUSTED AND ADJUSTED TRIAL BALANCES.
[62] COMPLETE:UN,'ADJUSTED TRIAL BALANCE'
[63] UN+' '
[64] FS+'I6,X3,2F10.2'
[65] FS CENTER 'MATRIX'
[66] FS COLNAMES '/ACCTS/DEBITS/CREDITS'
[67] FS ΔFMT(MATRIX)
[68] '
[69] FS ΔFMT(1 3 P0,+÷MATRIX[; 2 3])
[70] ' '
[71] ' '
[72] →L1×\1AYN 'DO YOU HAVE ANY ADJUSTMENTS TO MAKE?'
[73] →LADJ×\~AYN 'DO YOU WISH A PRINT OUT OF THE ADJUSTED TRIAL BALANCE?'
[74] ' ADJUSTED TRIAL BALANCE'
[75] FS+'I6,X3,2F10.2'
[76] FS CENTER 'MATRIX'
[77] FS COLNAMES '/ACCTS/DEBITS/CREDITS'
[78] FS ΔFMT(MATRIX)
[79] '
[80] FS ΔFMT(1 3 P0,+÷MATRIX[; 2 3])
[81] * GOES ON TO INCOME STATEMENT FUNCTION.
[82] LADJ:INC

```

▽

```

VINC[ ]
▽ INC;MA;NM;FS;TR;TE;NI
[1] * ISOLATING THE REVENUE ACCOUNTS, NETTING THEM OUT, AND PRINTING THEM.
[2] * THIS FUNCTION REQUIRES THE FOLLOWING FUNCTIONS:
[3] * BSHEET, CENTER, AND ΔFMT
[4] MA+(,(ACCTN>300)∧ACCTN<400)÷MATRIX
[5] MA[;3]÷-/MA[; 3 2]
[6] NM+((MA[;3])≠0)÷MA
[7] FS+'30A1,X10,F12.2'
[8] ' '
[9] ' '
[10] ' '
[11] ' '
[12] FS CENTER 'INCOME STATEMENT'
[13] ' '
[14] ' REVENUES'
[15] ' '
[16] FS ΔFMT(NAMES[((,ACCTN)\NM[;1]);];NM[;3])
[17] (40P' '),12P'-'

```

```

[18] 'X40,F12.2' ΔFMT(TR++/NM[;3])
[19] ' '
[20] ' EXPENSES'
[21] ' '
[22] * ISOLATING THE EXPENSE ACCOUNTS, NETTING THEM OUT, AND PRINTING THEM.
[23] MA+(,(ACCTN>400)^ACCTN<500)†MATRIX
[24] MA[;2]←-/MA[; 2 3]
[25] NM+((MA[;2])≠0)†MA
[26] FS ΔFMT(NAMES[((,ACCTN) \NM[;1]);];NM[;2])
[27] (40ρ' '),12ρ'_'
[28] 'X40,F12.2' ΔFMT(TE++/NM[;2])
[29] (40ρ' '),12ρ'_'
[30] * ARRIVING AT THE NET INCOME FIGURE AND PRINTING IT.
[31] 'X10, [NET INCOME],X30,F12.2' ΔFMT(NI+TR-TE)
[32] (52ρ' '),10ρ'_'
[33] * RETURNING THE NET INCOME FIGURE TO RETAINED EARNINGS.
[34] MATRIX[((,ACCTN) \206);3]←NI
[35] * GOES ON TO THE BALANCE SHEET FUNCTION.
[36] BSHEET
  ▽
    ▽BSHEET[ ]▽
  ▽ BSHEET;MA;NM;FS;TA;TLE
[1] * ISOLATING THE ASSET ACCOUNTS, NETTING THEM OUT, AND PRINTING THEM.
[2] * THIS FUNCTION REQUIRES THE FOLLOWING FUNCTIONS:
[3] * CENTER AND ΔFMT
[4] MA+(,(ACCTN>100)^ACCTN<200)†MATRIX
[5] MA[;2]←-/MA[; 2 3]
[6] NM+((MA[;2])≠0)†MA
[7] FS+ '30A1,X10,F12.2'
[8] ' '
[9] ' '
[10] ' '
[11] ' '
[12] FS CENTER 'BALANCE SHEET'
[13] ' '
[14] ' ASSETS'
[15] ' '
[16] FS ΔFMT(NAMES[((,ACCTN) \NM[;1]);];NM[;2])
[17] (40ρ' '),12ρ'_'
[18] 'X40,F12.2' ΔFMT(TA++/NM[;2])
[19] ' '
[20] ' LIABILITIES AND OWNERS EQUITY'
[21] ' '
[22] * ISOLATING THE LIABILITY AND EQUITY ACCOUNTS, NETTING THEM OUT AND PRINTING THEM.
[23] MA+(,(ACCTN≥200)^ACCTN<300)†MATRIX
[24] MA[;3]←-/MA[; 3 2]
[25] NM+((MA[;3])≠0)†MA
[26] FS ΔFMT(NAMES[((,ACCTN) \NM[;1]);];NM[;3])
[27] (40ρ' '),12ρ'_'
[28] 'X40,F12.2' ΔFMT(TLE++/NM[;3])
[29] (40ρ' '),12ρ'_'
  ▽

```

```

▽ C NAMES
[1]  A THIS ALLOWS THE USER TO ADD, CHANGE, OR DELETE ACCOUNTS.
[2]  A THIS FUNCTION REQUIRES THE FOLLOWING FUNCTIONS:
[3]  A AKI AND IPI
[4]  N1:→((1+1+ρNAMES)>L+(,ACCTN)\I+1+IPI 'ENTER ACCOUNT NUMBER')/GO
[5]  'NOT A VALID ACCTI, TRY AGAIN'
[6]  →0
[7]  GO:NAMES[L;]+30+AKI 'ENTER TITLE:'
[8]  →N1
▽

```

```

▽ ZERO
[1]  A THIS ALLOWS THE USER TO SET ALL HIS ACCOUNTS TO ZERO
[2]  MATRIX[;2]+MATRIX[;3]←0
▽

```

3.2

SIMULATION OF THE BIRTH AND DEATH OF FIRMS IN A COMPETITIVE INDUSTRY

Dennis M. Baker
Associate Professor of Accounting and Information Systems
Faculty of Business Administration and Commerce
University of Alberta
Edmonton, Alberta
Canada

DESCRIPTION

Introduction

The program described in this paper is an APL simulation of the birth and death of firms in a competitive industry. In summary, the model operates as follows:

1. Three firms begin production in period one.
2. Production is calculated for each firm.
3. An industry price is calculated.
4. Costs and profits are calculated for each firm.
5. Any firm with a net loss disappears from the industry.
6. If the total industry profit for the current period is equal to or greater than some minimum value, one new firm enters the industry in the next period; otherwise, a new firm does not enter.
7. When these determinations have been completed, time, the number of producing firms in the industry, and the total number that have entered the industry are updated.

8. The program then returns to 2 above until the maximum number of time periods has elapsed.

The program facilitates study of the effects of two variables, FACTR and AMULT, separately and in combination, on (1) the average industry production per time period, (2) the life of the firms that are in the industry, and (3) the industry price.

These variables are described in the subsequent pages.

In an attempt to conform to the economic theory of competition, the model is constructed under the following assumptions:

1. Competition is imperfect; that is, buyers and sellers do not have complete knowledge (i.e., if one firm obtains a technological breakthrough, others in the industry do not necessarily know of it immediately).
2. Firms are in business to maximize profits.
3. Each firm cannot individually influence input prices, but all working together can cause the prices of inputs to change.
4. No one firm can affect the industry price.
5. In the long run, the increase in the number of firms causes the industry supply curve to shift; and, with a given demand curve, the market price of the product will fall.
6. As new firms enter the industry, there is an effect on the costs of inputs.
7. The assumption of a "spaceless economy" is relaxed and factors such as location are allowed to affect a firm's cost function and possibly cause the death of a firm.

This simulation is an extension of Problem 6-5 described in Naylor, et al.¹ Thus, the output of each firm is determined by the following decision rule: each firm assumes that its competitors will not change their output in the following period. Each firm then produces at the optimum level for each period, based on its cost function and the industry demand function (in which output of competitors is held constant). Consequently, assumption 2 above is met. To meet assumption 1, parameters A and C (in Problem 6-5) are changed to exogenous variables (i.e., stochastic variates with given expected values, variances, and probability distributions).

Assumptions 3 and 6 are met by adding to B the product of B and BB, an exogenous variable. If the number of firms in the industry in period t is greater than the number in period t-1, then a "percentage-of-increase-in-input-costs" (BB) is calculated. Assumption 7 enters the model in the following manner: the environment (in which the industry operates) generates an "incident" at some point in time. The time at which the incident occurs (e.g., four time periods since the last incident) is a stochastic variate with given expected value, variance, and probability distribution. A test is made comparing the life of every firm in the industry at a given time period with a variable (VA in the program) generated for "number of years between the current incident and the immediately preceding one." The firm whose life at the point of the test exceeds "between-incident-years" is subjected to the consequences of the environment-generated event.

One way to effect a net loss for the selected firm would be to alter the revenue of that firm; and, if the net profit of the firm were negative, a "death" would result. However, this approach

1 See T. H. Naylor, J. L. Balintfy, D. S. Burdick, and Kong Chu, Computer Simulation Techniques, John Wiley and Sons, New York, 1966, pp. 197-98 and 234.

would violate assumption 4 in that if the price of one firm is affected, the price of all firms would be affected also because the industry price is the firm price.

Thus, because no one firm should be able to affect price, the shock is directed at the cost function of the firm selected. The program variable FACTR increments the costs of this firm; depending upon the "severity" of the incident (i.e., the magnitude of FACTR), the firm may or may not experience a net loss for the period and thus may or may not exit the industry. One note of caution: a new firm is allowed to enter the industry (in the model) if the total industry profit for the previous period does not fall below some arbitrarily selected amount (VALUE in the program). Therefore, it is important to note that the magnitude assigned to FACTR not only affects the costs of one firm, but also the profit of the total industry, with possible consequent effects on the entrance of a new firm. Thus, if FACTR is very high, the large loss for the firm subjected to environmental shock may be large enough to produce a total industry loss less than the amount of VALUE, in which case another firm would not enter the industry in the subsequent time period.

A second way in which one firm could influence the industry price (and therefore violate assumption 4) is created automatically with the birth of a new firm. This possibility occurs because a higher current period total production would be subtracted from the constant, D, in the industry price equation. To counter this possibility, the variable TERM is incorporated into the model. The basic model price equation for any period subtracts from D the quantity, "E x Current total production." To restore most of the original industry price (as determined in period one), TERM adds back to D (and the industry price) the quantity, "E x (total production in period t - total production in period 1)." Although two additional terms are also subtracted from D in the price

calculation, the weight of these terms, .20 and .03, does not affect price as much as E which has a value of .50.

Another variable, AMULT, was created to facilitate control of the impact of TERM. In the modified price equation, AMULT is multiplied times TERM. When AMULT = 1, the full impact of TERM flows through to the industry price. As AMULT decreases below 1, the impact of TERM diminishes; that is, PRICE in a period will be less than what it would have been because less is "added back" in the price equation. Thus, by decreasing the value of AMULT, one can move into the long run situation described in assumption 5, where the market price of the product will fall because of a supply curve shift resulting from an increase in the number of firms in the industry.

Variables, parameters, and operating characteristics for the model are as follows:

Exogenous Variables

- U = a stochastic variate with a given expected value, variance, and uniform distribution.
- V = a stochastic variate with a given expected value, variance, and uniform distribution.
- A = a stochastic variate with a given expected value, standard deviation, and normal distribution.
- BB = a stochastic variate with a given expected value, standard deviation, and normal distribution.
- C = a stochastic variate with a given expected value, standard deviation, and normal distribution.
- VA = a stochastic variate with a given expected value, standard deviation, and normal distribution.

FACTR = multiple by which a selected firm's production cost is enlarged.
AMULT = amount of impact on PRICE from the variable TERM.

Parameters

B = a technological parameter
D,E,F,G = constants
QLES2 = amount of total industry production two periods prior to period one of the simulation.
VALUE = lowest total industry loss level at which a new firm can enter the industry.
NBEGF = number of firms in the industry in period one.
PLES1 = production for first firm one period prior to period one of the simulation.
PLES2 = production for second firm one period prior to period one.
PLES3 = production for third firm one period prior to period one.

Endogenous Variables

PROD = matrix of quantities of units produced by firms in the industry (rows) in each time period (columns).
COST = matrix of the total cost of production for firms in the industry for each time period.
PRICE = vector of industry price in each time period.
PRFT = matrix of the net profits for firms in the industry for each time period.

Operating Characteristics

$$\text{COST}_{it} = (\text{PROD}_{it} - A_i)^2 + \text{BA}_i^2 + C + U_{it}$$

$$\text{PRICE}_t = D + (\text{TERM} \times \text{AMULT}) - \text{ES}_t - \text{FS}_{t-1} - \text{GS}_{t-2} + V_t$$

where

i = i^{th} firm

t = period t

S_t = quantity of industry output supplied and demanded in period t

Potential users of this simulation include undergraduate and graduate students in economics, management science, and information systems courses. For example, application of the program may include instruction in sensitivity analysis, detailed analysis of production and financial statistics for each firm, and modification and further enhancement of the simulation itself.

Instructions for Use

Entering the word *DATAENTRY* at the terminal activates the function by that name. This function requests the user to enter fourteen (14) values for the exogenous variables and eleven (11) values for the parameters listed above. The final entry is the number of periods to be simulated. In the present version of the simulation, the number of firms in the industry in period one (NBEGF) must equal three and the number of time periods (NN) must satisfy the relationship, $1 \leq \text{NN} \leq 10$. The twenty-six variables and their definitions for the function *DATAENTRY* are listed in the Explanation section. Note that values for QLES2, PLES1, PLES2, and PLES3 may be either computer or non-computer simulated or may

be assigned by the user, subject to certain initial conditions.

Setting the number of firms in the industry in the initial period equal to three introduces bias in the following manner. According to the cost equation, any firm's cost is roughly equal to $PROD^2$ (i.e., the values of A, B, C and U are small relative to the value of PROD). Therefore, assuming that each firm produces 200 units in period one, the cost of production for each firm is about 200^2 , or 40,000. Price is roughly equal to $D - ES_t - FS_{t-1}$ or 1000 (value of the constant D) - $.5(600) - .2(580)$ where $600 = 3 \times$ production of 200 for each firm and $580 =$ the input value for the first period preceding period one of the simulation. Thus price would be about $1000 - 300 - 116$, or 584.

One consequence of these initial conditions is a net profit of approximately 76,000 ($(200 \text{ units} \times 584) - 200^2$). This profit margin relates directly to the minimum level at which FACTR can be set to remove a firm from the industry and yet not produce a total industry loss below VALUE which would then block the entrance of a new firm into the industry. A net profit of 76,000 provides a buffer against "death" and indicates that FACTR would have to be about 2.9 to generate a net loss and the death of a firm (i.e., $(200 \text{ units} \times 584) - (2.9 \times 200^2) = 0$) and also not effect a reduction in industry profit from production by other firms. Therefore, when the effects of changes in FACTR on various simulation outputs are being investigated, one should recognize that the initial conditions of three firms produce a measurable "margin against death." Thus, because each simulation run is biased by its initial conditions, the experimenter should make use of his a priori information about the system under study and the purposes of the simulation to select initial conditions with a bias which will support the purposes of this study.

A second consequence of initial conditions is that PRICE is determined by the actions of the three firms, and it is this price

generated in period one that is maintained or approached in the computation of TERM. Finally, besides the number of firms in period one, another condition is the setting of $BB = 0$ (i.e., the percentage of increase in input costs in period one is zero).

The function *REPORT* provides two outputs: the production matrix PROD and the average industry production per time period. The PROD output is the set of time paths of unit quantities produced by all firms. Time of entry into, and exit from, the industry are displayed. For example, if firm *m* (row *m* of the matrix PROD) enters the industry in period 5 and exits in period 8, row *m* will contain values only in columns five through eight.

In addition to the two outputs described above, modification of the function *REPORT* would facilitate the outputting of other financial and statistical information. Other financial results available include the industry price for each period (PRICE) and the revenue, cost, and profit for each firm in all time periods (i.e., the variables REV, COST, and PRFT). Firm statistics available include the number of producing firms in the industry for each period (NFRMS) and the life of each firm (ILIFE).

Experiments designed to test the effects of changes in values of FACTR and AMULT on different variables (e.g., COST, ILIFE) include the following:

Values Selected For			
<u>Experiment No.</u>	<u>FACTR</u>	<u>AMULT</u>	<u>Purposes</u>
1	9.0	1	To test effects of decreases in FACTR
2	5.0	1	
3	2.0	1	

4	9.0	.7	To test effects of decreases in AMULT
5	9.0	.5	
6	9.0	.3	

7	5.0	.7	To test joint effect of decreases in FACTR and AMULT

The input of typical data requested by the function *DATAENTRY* and typical outputs of the function *REPORT* are shown in the Sample Use section. For purposes of illustration, experiment No. 1 (above) was used. Data values were entered to test this experiment. (See the symbol table in the following section for the definitions of all of the variables.)

Program Explanation

Public functions necessary for the successful execution of *REPORT* include *CENTER*, *RWTD*, and ΔFMT . Logical segments of the simulation appear as separate functions. Consequently, execution proceeds through the functions in the following order: *DATAENTRY*, *INITIALIZE*, *PRODUCTION*, *COSTREV*, *UPDATE*, and *REPORT*. Because two or more functions must modify the value of a given variable, all variables are global. APL code for the random number generators, *RANDU* and *RANDN*, was prepared by Dr. J. L. Politzer of the Pennsylvania State University Computation Center Staff and made available to Computing Services, University of Alberta, Edmonton.

Special features of the code are the creation of the variables *P* and *RF*. When first created in the function, *INITIALIZE*, *P* is a production indicator matrix where the $(i,j)^{th}$ cell value = 1 signifies that the i^{th} firm, unless removed from the industry, will produce in the j^{th} time period. To record the exit of a firm from the industry, *P* is multiplied in the function *UPDATE* by *RF*, a matrix of zeros and ones where a zero indicates removal of a firm and a one permits continued existence of a firm. Therefore, in the subsequent time period, *PROD* (in the function *PRODUCTION*) is multiplied by the modified *P*.

Variables first assigned in the functions *DATAENTRY*, *INITIALIZE*, *PRODUCTION*, *COSTREV*, and *UPDATE* and their definitions, in order of appearance, are as follows:

<u>Function</u>	<u>Variable</u>	<u>Definition</u>
<i>DATAENTRY</i>	B	a technological parameter.
"	D,E,F,G	constants
"	EXVA1	expected value of V.
"	VAR1	variance of V.
"	EXVA2	expected value of U.
"	VAR2	variance of U.
"	QLES2	amount of total industry production two periods prior to period one of the simulation.
"	VALUE	lowest total industry loss level at which a new firm can enter the industry.
"	NBEGF	number of firms in the industry in period one.
"	PLES1	production for first firm one period prior to period one of the simulation.
"	PLES2	production for second firm one period prior to period one of the simulation.
"	PLES3	production for third firm one period prior to period one of the simulation.
"	STDA	standard deviation of A.
"	EVA	expected value of A.
"	STDB	standard deviation of BB.
"	EVV	expected value of BB.
"	STDC	standard deviation of C
"	EVC	expected value of C.
"	FACTR	multiple by which a selected firm's production cost is enlarged.
"	AMULT	amount of impact on PRICE from the variable TERM.
"	STDV	standard deviation of VA.
"	EVV	expected value of VA.
"	NN	number of time periods to be simulated.

<u>Function</u>	<u>Variable</u>	<u>Definition</u>
<i>INITIALIZE</i>	T	current time period.
"	TT	index for columns of PROD in which production in period T will be recorded.
"	MM	maximum number of producing firms (also the number of rows the columns in PROD).
"	IN	column numbers of PROD to be assigned to P.
"	P	production indicator matrix.
"	ILIFE	vector of lives of firms (value of element i = life of firm i).
"	BB	percentage of increase in input costs.
"	NADD	vector of number of firms added to the industry in period T (value will equal zero or one).
"	NDROP	vector of number of firms dropped in period T.
"	PRICE	vector of the industry price for each time period.
"	EXP	sum of production in period T-1 weighted by F plus sum of production in period T-2 weighted by G, at any time period T.
"	V	a stochastic variate with a given expected value, variance, and uniform distribution.
"	TNFRM	total number of firms that have entered the industry up to and including a given time period.
"	NFRMS	the number of firms producing in the industry in a given time period.
"	REV	matrix of revenues for firms in the industry for each time period.
"	PRFT	matrix of the net profits for the firms in the industry for each time period.

<u>Function</u>	<u>Variable</u>	<u>Definition</u>
<i>INITIALIZE</i>	COST	matrix of the total cost of production for the firms in the industry for each time period.
"	U	a stochastic variate with a given expected value, variance, and uniform distribution.
"	VA	a stochastic variate with a given expected value, standard deviation, and normal distribution.
"	C	a stochastic variate with a given expected value, standard deviation, and normal distribution.
"	A	a stochastic variate with a given expected value, standard deviation, and normal distribution.
"	DIV	the divisor in the production equation.
"	ALPH1	lower bound of the range over which V will be distributed.
"	BETA1	upper bound of the range over which V will be distributed.
"	ALPH2	lower bound of the range over which U will be distributed.
"	BETA2	upper bound of the range over which U will be distributed.
<i>PRODUCTION</i>	VV	reshaping of V into a vector for use in calculating PROD.
"	EEXP	reshaping of EXP into a vector for use in calculating PROD.
"	PP	vector whose i^{th} element is total production in period T-1 minus production of firm i in T-1.
"	PROD	quantities produced by each firm in the industry in period T.

<u>Function</u>	<u>Variable</u>	<u>Definition</u>
<i>COSTREV</i>	LIFE1	firms at time T with age \leq VA.
"	LIFE2	firms at time T with age $>$ VA.
"	CST1	costs for firms at time T with age \leq VA.
"	CST2	costs for firms at time T with age $>$ VA.
"	COST	matrix of the total cost of production for firms in the industry for each time period.
"	TERM	$E \times$ (production in period T - production in period 1).
<i>UPDATE</i>	FIRMDROP	vector indexing firms to be dropped from the industry.
"	XX	vector of column numbers of P to be adjusted for firms exiting the industry.
"	RF	matrix of zeros and ones where a zero indicates removal of a firm and a one permits continued existence of a firm.
"	IZ	period T-1.

PROGRAM USAGE

DATAENTRY

ENTER THE VALUES FOR B, D, E, F AND G;
INSERT ONE SPACE BETWEEN EACH NUMBER.

□:

2 1000 0.5 0.2 0.03

ENTER THE VALUES FOR EXVA1, VAR1, EXVA2, VAR2,
QLES2 AND VALUE; INSERT ONE SPACE BETWEEN EACH NUMBER.

□:

0 0.5 0 1 500 3000

ENTER THE VALUES FOR NBEGF, PLES1, PLES2 AND PLES3
(LEAVE ONE SPACE BETWEEN EACH NUMBER)

□:

3 200 300 80

ENTER THE VALUES FOR STDA, EVA, STDB, EVB, STDC, EVC
(LEAVE ONE SPACE BETWEEN EACH NUMBER)

□:

0.1 2 0.005 0.04 1.333 24

ENTER THE VALUES FOR FACTR, AMULT, STDV AND EVV
(LEAVE ONE SPACE BETWEEN EACH NUMBER)

□:

9 1 0.6667 5

ENTER NN; NN MUST BE AN INTEGER WITH VALUE $1 \leq NN \leq 10$

□:

10

CALL THE SIMULATOR BY TYPING THE WORD INITIALIZE
INITIALIZE

THE PRODUCTION MATRIX FOR THE PREVIOUS 2 PERIODS AND THE
SIMULATED 10 TIME PERIODS IS:

TIME PERIODS

	-1	0	1	2	3	4	5	6	7	8	9	10
1	500.0	200.0	227.7	208.2	177.3							
2		300.0	244.4	211.0	177.8	159.3	175.4					
3		80.0	207.7	204.9	176.8	159.1	175.4					
4				170.3	171.0	158.1	175.2					
5					142.6	153.4	174.4	118.2	262.6			
6						129.6	170.5	117.6	262.5	134.6	253.4	
7							148.9	114.0	261.9	134.5	253.4	
8												
9									242.9	131.3	252.9	141.5
10												
11											231.0	137.8
12												

AVERAGE PRODUCTION FOR EACH OF THE 10 PERIODS IS:

1	679.9
2	794.4
3	845.5
4	759.6
5	1019.9
6	349.7
7	1029.9
8	400.3
9	990.7
10	279.3

PROGRAM DESCRIPTION

```

V INITIALIZE
[1]  A THIS FUNCTION INITIALIZES VARIOUS ARRAYS AND PARAMETERS
[2]  A AND SETS VALUES FOR DIV, ALPH1, BETA1, ALPH2 AND BETA2
[3]  T+SETLINKI20
[4]  T+1
[5]  TT+T+2
[6]  MM+D6+2
[7]  NN+D6
[8]  PROD+(1MM)0.51MM
[9]  IN+2+1NN
[10] P+PROD[;IN]
[11] PROD[1;1]+D2[5]
[12] PROD[1 2 3 ;2]+D3[2 3 4]
[13] B+D1[1]
[14] E+D1[3]
[15] F+D1[4]
[16] G+D1[5]
[17] ILIFE+(3p1),(MM-3)p0
[18] V+EXP+PRICE+NDROP+NADD+BB+D6p0
[19] NFKMS+TNFKM+D3[1],((NN-1)p0)
[20] A+C+VA+U+COST+PRFT+REV+(MM,NN)p0
[21] D+MMpD1[2]
[22] DIV+MMp(2+2×E)
[23] ALPH1+D2[1]-0.5*3×D2[2]
[24] BETA1+(2×D2[1])-ALPH1
[25] ALPH2+D2[3]-0.5*3×D2[4]
[26] BETA2+(2×D2[3])-ALPH2
[27] PRODUCTION
V

V PRODUCTION
[1]  A THIS FUNCTION CALCULATES THE UNIT PRODUCTION FOR
[2]  A EACH FIRM IN EXISTENCE IN THE CURRENT TIME PERIOD
[3]  VV+MMpV[T]+ALPH1+(BETA1-ALPH1)×Z+RANDU 1
[4]  EEXP+MMpEXP[T]+(F++PROD[;(TT-1)])+(G++PROD[;(TT-2)])
[5]  A[;T]+MMpZZ+D4[1] RANDN D4[2]
[6]  A EACH FIRM ASSUMES THAT THE ONLY CHANGES IN OUTPUT FOR THE
[7]  A INDUSTRY IN THE NEXT PERIOD (OVER THE PREVIOUS PERIOD)
[8]  A WILL BE ATTRIBUTED ONLY TO ITS CHANGES IN OUTPUT AND
[9]  A NO THAT OF ITS COMPETITORS. NOW, FOR EACH FIRM, CALCULATE
[10] A THE TOTAL PRODUCTION OF ALL OTHER FIRMS IN THE PREVIOUS
[11] A PERIOD BY SUBTRACTING EACH FIRM'S PRODUCTION FROM THE TOTAL
[12] A INDUSTRY PRODUCTION.
[13] PP+(MMp(+PROD[;(TT-1)]))-PROD[;(TT-1)]
[14] A DETERMINE PRODUCTION FOR THE CURRENT PERIOD; IF THE
[15] A PRODUCTION INDICATOR MATRIX P CONTAINS A CELL VALUE
[16] A OF ZERO, THIS INDICATES THE DEATH OF A FIRM IN THE
[17] A PREVIOUS TIME PERIOD AND THEREFORE NO PRODUCTION
[18] A FOR THIS FIRM IN THE CURRENT PERIOD.
[19] PROD[;TT]+(((VV+D+2×A[;T])-(E×PP)+EEXP)×P[;T]):DIV
[20] A UPDATE THE LIFE OF THE FIRMS IN EXISTENCE BY ONE TIME PERIOD
[21] ILIFE+ILIFE+(PROD[;TT]>0)
[22] COSTREV
V

```

```

V R+A CENTER B
[1] A+*/0 ^2 +RWTD A
[2] R+(1,A)ρA↑((↑0.5×0[A-ρB)↑' '),B+,B
V

```

```

V COSTREV
[1] A THIS FUNCTION DETERMINES THE COST OF PRODUCTION FOR
[2] A FIRMS WHOSE PRESENT AGE IS ≤ OR > THE RANDOM VARIABLE
[3] A VA; TOTAL COST FOR EACH FIRM, UNIT PRICE IN PERIOD T
[4] A AND REVENUE AND PROFIT FOR FIRMS IN THE CURRENT PERIOD
[5] A ARE THEN DETERMINED.
[6] U[;T]+ALPH2+(BETA2-ALPH2)×U[;T]←RANUNI 1
[7] VA[;T]+D5[3] RANDOR D5[4]
[8] C[;T]+D4[5] RANDOR D4[6]
[9] A IDENTIFY FIRMS WITH PRESENT AGE≤VA
[10] LIFE1←ILIFE≤VA[;T]
[11] A IDENTIFY FIRMS WITH PRESENT AGE>VA
[12] LIFE2←ILIFE>VA[;T]
[13] CST2←CST1+(MM,NN)ρ0
[14] CST1[;T]+P[;T]×LIFE1×C[;T]+U[;T]+(B×(A[;T]*2))+((PROD[;TT]-A[;T])*2)
[15] CST2[;T]+P[;T]×LIFE2×D5[1]×C[;T]+U[;T]+(B×(A[;T]*2))+((PROD[;TT]-A[;T])*2)
[16] COST[;T]+CST1[;T]+CST2[;T]
[17] TERM+E×((+PROD[;TT])-(+PROD[;3]))
[18] PRICE[T]←(D1[2]+(TERM×D5[2])+V[T])-(EXP[T]+(D1[3]×(+PROD[;TT])))
[19] →(PRICE[T]≥0)/P1
[20] PRICE[T]←0
[21] P1:REV[;T]+PRICE[T]×PROD[;TT]
[22] PRFT[;T]+REV[;T]-COST[;T]
[23] UPDATE
V

```

```

V DATAENTRY
[1] 'ENTER THE VALUES FOR B, D, E, F AND G;'
[2] 'INSERT ONE SPACE BETWEEN EACH NUMBER.'
[3] D1←□
[4] 'ENTER THE VALUES FOR EXVA1, VAR1, EXVA2, VAR2,'
[5] 'PLES2 AND VALUE; INSERT ONE SPACE BETWEEN EACH NUMBER.'
[6] D2←□
[7] 'ENTER THE VALUES FOR NBEGF, PLES1, PLES2 AND PLES3'
[8] '(LEAVE ONE SPACE BETWEEN EACH NUMBER)'
[9] D3←□
[10] 'ENTER THE VALUES FOR STDA, EVA, STDB, EVB, STDC, EVC'
[11] '(LEAVE ONE SPACE BETWEEN EACH NUMBER)'
[12] D4←□
[13] 'ENTER THE VALUES FOR FACTR, AMULT, STDV AND EVV'
[14] '(LEAVE ONE SPACE BETWEEN EACH NUMBER)'
[15] D5←□
[16] 'ENTER NN; NN MUST BE AN INTEGER WITH VALUE 1≤NN≤10'
[17] D6←□
[18] 'CALL THE SIMULATOR BY TYPING THE WORD INITIALIZE'
V

```

```

▽ UPDATE
[1]  A THIS FUNCTION DROPS AND/OR ADDS FIRMS TO THE
[2]  A INDUSTRY, UPDATES TIME PERIODS AND CALCULATES
[3]  A THE CHANGE, IF ANY, IN THE PARAMETER B.
[4]  A COUNT THE NUMBER OF FIRMS TO BE DROPPED.
[5]  NDROP[T]←+/(PRFT[;T]<0)
[6]  A DETERMINE THOSE FIRMS TO BE DROPPED FROM THE INDUSTRY.
[7]  →((T-NN)≥0)/UP5
[8]  A INDEX FIRMS WHICH WILL EXIT THE INDUSTRY.
[9]  FIRMDROP←(PRFT[;T]<0)^(MMp1)
[10] XX←T+1NN-T
[11] RF←Q((pXX),MM)ρFIKMDROP
[12] A RECORD ZEROS IN PRODUCTION INDICATOR MATRIX P FOR FIRMS
[13] A THAT HAVE EXITED THE INDUSTRY
[14] P[;XX]←RF×P[;XX]
[15] A IF TOTAL INDUSTRY PROFIT + VALUE(D2[6]) IS ≤ 0, DO NOT
[16] A ADD A FIRM TO THE INDUSTRY; OTHERWISE, ADD ONE FIRM
[17] UP5:→(((+PRFT[;T])+D2[6])≤0)/UP1
[18] NADD[T]←NADD[T]+1
[19] →((T-NN)≥0)/REP
[20] →NEXTPERIOD
[21] UP1:→((T-NN)≥0)/REP
[22] P[(3+T);]←0
[23] NEXTPERIOD:T←T+1
[24] TT←TT+1
[25] A IF NUMBER OF FIRMS IN NEW PERIOD IS > NUMBER IN PERIOD
[26] A T-1, ADJUST PARAMETER B.
[27] IZ←T-1
[28] NFRMS[T]←(NFRMS[IZ]+NADD[IZ])-NDROP[IZ]
[29] TNFRM[T]←TNFRM[IZ]+NADD[IZ]
[30] →((NFRMS[T]-NFRMS[IZ])≤0)/UP3
[31] B←B+B×BB[T]←D4[3] RANDN D4[4]
[32] PRODUCTION
[33] →0
[34] UP3:BB[T]←0
[35] PRODUCTION
[36] →0
[37] REP:REPORT

```

▽

```

      V ZZ←SIG RANDN MU
[1] ZZ←MU+SIG×-6+(+/-1+?12p178956970)÷178956969
      V

```

```

      V ZZ←SIG RANDOR MU
[1] X←1
[2] ZZ←MMp0
[3] ZZ[X]+MU+SIG×-6+(+/-1+?MMp178956970)÷178956969
[4] →(MM<X+X+1)/0
[5] →3
      V

```

```

      V Z←RANDU X
[1] Z←X×(-1+?2147483647)÷2147483646
      V

```

```

      V RU←KANUNI X
[1] RU←X×(-1+?MMp2147483647)÷2147483646
      V

```

```

      V REPORT
[1] FORMAT←'I4,5X1,',(RP+'0123456789'[1+ 10 10 TMM]),'BF8.1'
[2] 'THE PRODUCTION MATRIX FOR THE PREVIOUS ';MM-NN;' PERIODS AND THE'
[3] ' SIMULATED ';NN;' TIME PERIODS IS: ';3pQR
[4] FORMAT CENTER 'TIME PERIODS';4pQR
[5] (,('X9,',RP,'I8') ΔFMT(1,MM)p-2+iMM),2pQR
[6] FORMAT ΔFMT(iMM;PROD);8pQR
[7] 'AVERAGE PRODUCTION FOR EACH OF THE ';NN;' PERIODS IS: ';3pQR
[8] 'I4,5X1,F8.1' ΔFMT(iNN;((+PROD[;(2+iNN)])÷NFRMS[T]))
[9] →0
      V

```

```

      V TESTDATA
[1] D1← 2 1000 0.5 0.2 0.03
[2] D2← 0 0.5 0 1 500 3000
[3] D3← 3 200 300 80
[4] D4← 0.1 2 0.005 0.04 1.333 24
[5] D5← 9 1 0.6667 5
[6] D6←10
[7] INITIALIZE
      V

```

SINGLE ASSET DEPRECIATION METHODS

Lawrence D. Brown
Assistant Professor of Accounting and Economics
School of Management
State University of New York at Binghamton
Binghamton, New York

DESCRIPTION

The primary purpose of this program, entitled *DEPRECIATE*, is to aid management educators in teaching single asset depreciation methods to beginning and intermediate financial accounting students. The program can be used for drill and practice, homework assignments, or in the classroom. It enables the user to choose any of the following depreciation methods:

1. Straight Line (SL)
2. Sum-of-the-years Digits (SYD)
3. Declining Balance (DB)
4. Double Declining Balance (DDB)
5. Sinking Fund (SF)
6. Annuity (AN)

Usage Instructions

The user is initially asked to respond to the following four questions:

1. Historical cost of asset?
2. Expected net salvage value of asset?
3. Expected life of asset (in years)?
4. Depreciation method desired: SL, SYD, DB, DDB, SF, or AN?

The Historical Cost (HC) can be any positive number less than one trillion. The Net Salvage Value (NSV) can be any positive number less than the historical cost of the asset. The Expected Asset Life (N) can be any integer between one and nine hundred ninety-nine. If any of these constraints are violated, asterisks will be printed out in the depreciation table in lieu of numbers.

One of the mnemonics representing the six depreciation methods identified above must be entered in response to question 4. If the user does not input one of these available depreciation methods, he will receive an error message and will be asked question 4 again.

After these four questions are answered satisfactorily, the user obtains a depreciation table containing Annual Depreciation Expense (DE) Accumulated Depreciation (ACC), and Book Value (BV). The user is then given the option to plot depreciation expense over time. He is next asked whether or not he desires to use the same data for a different depreciation method. If he responds *YES*, he is reasked question 4 above. If he answers *NO*, he is asked if he wants to enter new data. An affirmative answer results in question 1 being asked again. A negative response yields the message, *GOODBYE FOR NOW*, and exits the user from the program.

In addition to the function *DEPRECIATE*, the following public functions are also required: *AKI*, *AYN*, *NIP*, *PLOT*, and *VS*. In addition, Scientific Time Sharing Corporations *CENTER*, *COLNAMES*, and *ΔFMT* are also needed in order to execute the program successfully.

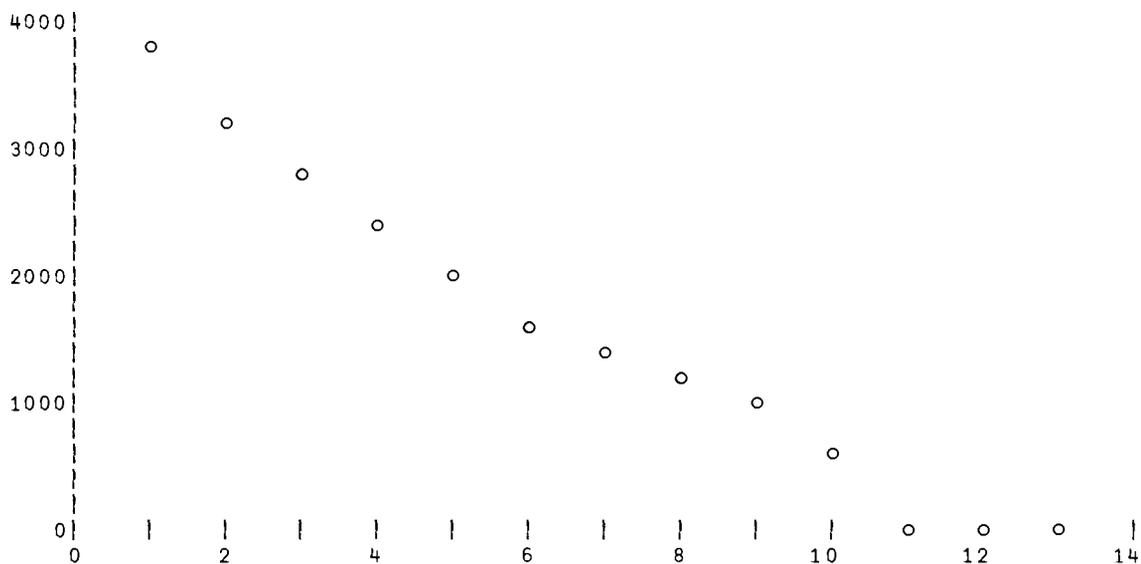
PROGRAM USAGE

DEPRECIATE
 HISTORICAL COST OF ASSET?: 25000
 EXPECTED NET SALVAGE VALUE OF ASSET?: 5000
 EXPECTED LIFE OF ASSET (IN YEARS)?: 13
 DEPRECIATION METHOD DESIRED: SL,SYD,DB,DDB,SF,OR AN?
 DDB

TABLE OF ASSET DEPRECIATION

YEAR	DEP EXP	ACC DEP	BOOK VALUE
1	3846.15	3846.15	21153.85
2	3254.44	7100.59	17899.41
3	2753.76	9854.35	15145.65
4	2330.10	12184.45	12815.55
5	1971.62	14156.07	10843.93
6	1668.30	15824.37	9175.63
7	1411.64	17236.00	7764.00
8	1194.46	18430.46	6569.54
9	1010.70	19441.16	5558.84
10	558.84	20000.00	5000.00
11	0.00	20000.00	5000.00
12	0.00	20000.00	5000.00
13	0.00	20000.00	5000.00

DO YOU WANT A PLOT OF DEPRECIATION EXPENSE VERSUS TIME?: Y
 PLEASE ANSWER YES OR NO: YES



DO YOU WANT TO USE THE SAME DATA TO CALCULATE DEPRECIATION EXPENSE,
 ACCUMULATED DEPRECIATION AND BOOK VALUE USING A DIFFERENT DEPRECIATION METHOD? YES
 DEPRECIATION METHOD DESIRED: SL,SYD,DB,DDB,SF,OR AN?

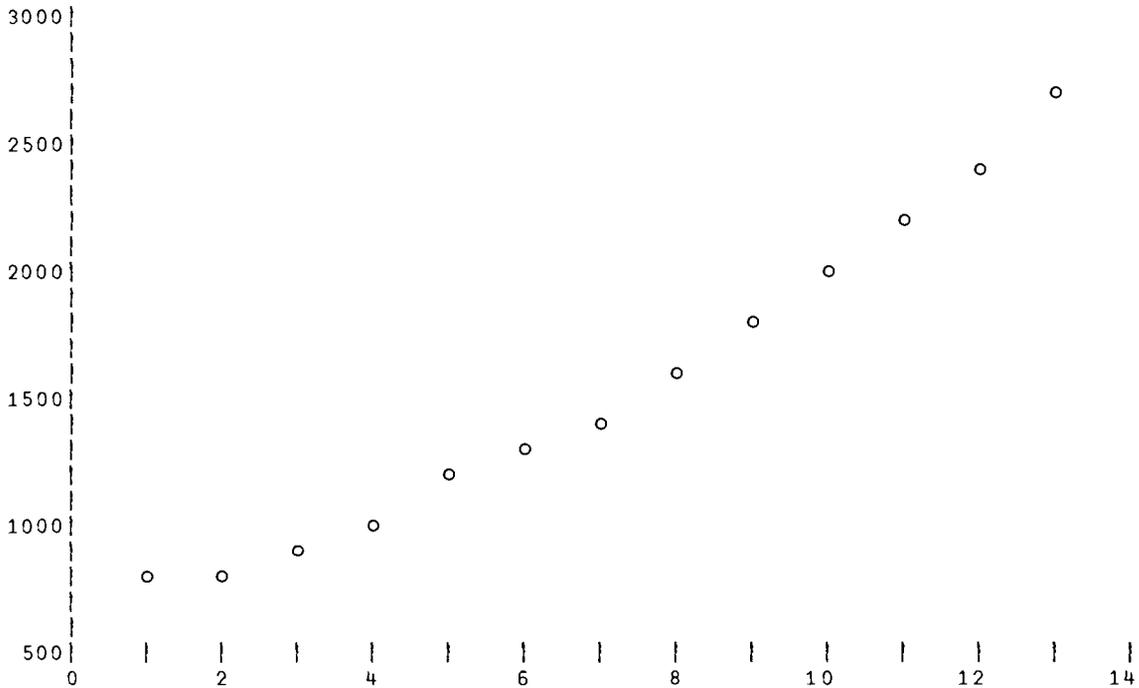
SF

WHAT INTEREST RATE DO YOU WISH TO APPLY TO THIS ASSET? (0 ≤ INT RATE ≤ 100): 11

TABLE OF ASSET DEPRECIATION

YEAR	DEP EXP	ACC DEP	BOOK VALUE
1	763.02	763.02	24236.98
2	846.95	1609.97	23390.03
3	940.12	2550.09	22449.91
4	1043.53	3593.62	21406.38
5	1158.32	4751.94	20248.06
6	1285.73	6037.67	18962.33
7	1427.16	7464.83	17535.17
8	1584.15	9048.98	15951.02
9	1758.41	10807.39	14192.61
10	1951.83	12759.22	12240.78
11	2166.53	14925.76	10074.24
12	2404.85	17330.61	7669.39
13	2669.39	20000.00	5000.00

DO YOU WANT A PLOT OF DEPRECIATION EXPENSE VERSUS TIME?: YES



DO YOU WANT TO USE THE SAME DATA TO CALCULATE DEPRECIATION EXPENSE,
 ACCUMULATED DEPRECIATION AND BOOK VALUE USING A DIFFERENT DEPRECIATION METHOD? NO
 DO YOU WANT TO ENTER NEW DATA? NO
 GOODBYE FOR NOW

PROGRAM DESCRIPTION

```

▽ DEPRECIATE
[1]  THIS FUNCTION REQUIRES THE FOLLOWING OTHER FUNCTIONS:
[2]  AAKI, AYN, IPI, NIP, OUTPUT, PLOT, AND VS
[3]  L2:→L3×10<HC+NIP 'HISTORICAL COST OF ASSET?: '
[4]  'HISTORICAL COST MUST BE GREATER THAN ZERO: ' ◇ →L2
[5]  L3:→L4×10≤NSV+NIP 'EXPECTED NET SALVAGE VALUE OF ASSET?: '
[6]  'EXPECTED NET SALVAGE VALUE OF ASSET MUST BE POSITIVE: ' ◇ →L3
[7]  L4:→(NSV<HC)/L5
[8]  'HISTORICAL COST MUST EXCEED NET SALVAGE VALUE: ' ◇ →L3
[9]  L5:→L6×11<N+IPI 'EXPECTED LIFE OF ASSET (IN YEARS)?: '
[10] 'EXPECTED LIFE OF ASSET MUST EXCEED ONE: ' ◇ →L5
[11] L6:'DEPRECIATION METHOD DESIRED: SL,SYD,DB,DDB,SF,OR AN?'
[12] →(SL,SYD,DB,DDB,SF,AN,ERR) ['LYBDFN'11+2+,]
[13] ERR:'YOU MADE AN ERROR. THE ONLY PERMISSIBLE DEPRECIATION'
[14] 'METHODS ARE SL,SYD,DB,DDB,SF,AND AN: ' ◇ →L6
[15] SL:→L9,CV+HC-ACC+(1N)×DE+Np(HC-NSV)÷N
[16] SYD:→L9,CV+HC-ACC++\DE+(Φ1N)×2×(HC-NSV)÷N×N+1
[17] DB:CV+HC-ACC++\DE+HC×((1-FRAC)×((1N)-1))×FRAC+1-(NSV÷HC)*÷N ◇ →L9
[18] DDB:CV+HC-ACC++\DE+HC×((1-FRAC)×((1N)-1))×FRAC+2÷N
[19] DE+ACC-0,1+ACC+HC-CV+CV[NSV ◇ →L9
[20] SF:→L7×11>INT+0.01×INT+NIP 'WHAT INTEREST RATE DO YOU WISH TO APPLY TO THI
S ASSET? (0 ≤ INT RATE ≤ 100): '
[21] 'INTEREST RATE CANNOT EXCEED ONE HUNDRED.' ◇ →SF
[22] L7:CV+HC-ACC++\DE+(HC-NSV)×(INT÷(((1+INT)*N)-1))×(1+INT)×(1N)-1 ◇ →L9
[23] AN:→L8×11>INT+0.01×INT+NIP 'WHAT INTEREST RATE DO YOU WISH TO APPLY TO THI
S ASSET? (0 ≤ INT RATE ≤ 100): '
[24] 'INTEREST RATE CANNOT EXCEED ONE HUNDRED.' ◇ →AN
[25] L8:DE+Np(HC-(NSV×(1÷(1+INT)*N)))÷(1-(1÷(1+INT)*N))÷INT
[26] CV+HC-ACC++\HC-NSV)×(INT÷(((1+INT)*N)-1))×(1+INT)×(1N)-1 ◇ →L9
[27] L9:OUTPUT
[28] →L11×1R+AYN 'DO YOU WANT A PLOT OF DEPRECIATION EXPENSE VERSUS TIME?: '
[29] L10:'DO YOU WANT TO USE THE SAME DATA TO CALCULATE DEPRECIATION EXPENSE,'
[30] →L6×1R+AYN 'ACCUMULATED DEPRECIATION AND BOOK VALUE USING A DIFFERENT DEP
RECIATION METHOD? '
[31] →L2×1R+AYN 'DO YOU WANT TO ENTER NEW DATA?
[32] 'GOODBYE FOR NOW' ◇ →0
[33] L11: 30 60 PLOT DE VS1N ◇ →L10

```

```

▽
▽ OUTPUT
[1]  THIS FUNCTION REQUIRES THE FOLLOWING OTHER FUNCTIONS:
[2]  ACENTER, COLNAMES, AND ΔFMT
[3]  FS+'I4,3F17.2'
[4]  ' '
[5]  FS CENTER 'TABLE OF ASSET DEPRECIATION'
[6]  ' '
[7]  FS COLNAMES '/YEAR/DEP EXP/ACC DEP/BOOK VALUE/'
[8]  R+1N
[9]  FS ΔFMT(R;DE;ACC;CV)

```


3.4

REAL ESTATE VALUATION USING THE DIRECT CAPITALIZATION METHOD

Fred E. Case
Professor of Urban Land Economics
Graduate School of Management
University of California
Los Angeles, California

DESCRIPTION

Introduction

The value of income-producing real estate may be calculated by one of two methods: the present worth of a stream of future benefits or by the direct capitalization method. The most commonly used -- and most conservative -- is the latter, and it is this method which is used in this program. The basic formula is $V = NY \div CF$, where V is value, NY is the net income, and CF is the capitalization factor. Depending upon the data available, the building residual method, the land residual method, or the property residual method can be used. The accompanying program, *DIRCAL*, is designed to perform each of these calculations.

This program can be used by anyone wanting to understand the fundamentals of the direct method of capitalization and to calculate the value for any type of income-producing property for which data are available. The program is useful in both undergraduate and graduate courses in real estate economics, valuation, finance, and investments.

Use of the Program

To use the function, the user provides the information requested and is then supplied with the answers. Basically, the user must supply data on gross scheduled income; a vacancy and

collection loss percentage; operating expenses, both fixed and reserved, expressed as a percentage of gross scheduled income; property taxes as a percent of the property market value; anticipated earnings rate; and anticipated earnings period or economic life. Then, depending upon the data which are available, estimates are required on the land value; the building value; and the expected residual value of the property at the end of the investment period.

In entering data, a few words of explanation are in order. The following are some of the input prompts, accompanied by a more detailed explanation of what they mean.

ENTER GROSS SCHEDULED INCOME (GSY)

Gross scheduled income refers only to scheduled rents, not to other forms of income.

*ENTER OPERATING EXPENSES (OE) AS A
PERCENTAGE OF GROSS SCHEDULED INCOME*

Operating expenses are not total expenses; total expenses include operating expenses (OE), reserve expenses (RE), fixed expenses (FE), and property taxes (TE).

ENTER RESERVE EXPENSE PERCENTAGE (RE)

Reserve expenses are cash outlays which occur in one year which relate to more than one year of operation. To obtain the figure for reserve expenses, divide each reserve expense by the number of years over which it is to be allocated. For example, roof repairs might cost \$1000 in one year but be expected to last for ten years. Dividing \$1000 by 10 would yield an annual cost of \$100. All these annualized expenses should then be totalled and expressed as a percentage of the gross operating income. Reserve expenses are not made for the purpose of increasing the rents but merely to maintain the scheduled rents.

ENTER FIXED EXPENSE PERCENTAGE (FE)

Fixed expenses include such things as fire insurance, fees, licenses, and similar types of costs whose amounts are determined externally and are not a function of the level of operation.

ENTER MARKET VALUE OF THE PROPERTY (MV)

The market value is either the original purchase price or the current estimated market value, whichever is most appropriate. If neither are available, use the assessed value, adjusted to approximate the market value.

ENTER LAND VALUE (LV)

ENTER BUILDING VALUE (BV)

Each of the above should be entered separately. If they are not available directly, they should be derived from the assessed values.

The remaining requests for input are fairly straightforward. When all inputs have been completed, the program will print out the resultant value estimations. These are calculated according to the building residual method, the land residual method, and the property residual method. If the user wishes to run the program again, retyping *DIRCAL* will cause it to restart.

All data must be entered each time calculations are run. However, the user can vary the data each time according to his needs. The sample session provides an illustration of the kinds of data asked for and the output produced, given the data provided.

PROGRAM USAGE

DIFCAL

ENTER ALL PERCENTAGE FIGURES AS DECIMAL FRACTIONS, E.G.,
8 PERCENT WOULD BE ENTERED AS .08
IF YOU MAKE A MISTAKE ENTERING DATA, START OVER.

ENTER GROSS SCHEDULED INCOME(GSY)

□:
30000

ENTER VACANCY AND COLLECTION LOSS (VCL) AS A PERCENTAGE OF
GROSS SCHEDULED INCOME.

□:
.06

ENTER OPEATING EXPENSES (OE), AS A PERCENTAGE OF GROSS SCHEDULED INCOME.

□:
.10

ENTER RESERVE EXPENSE PERCENTAGE(RE)

□:
.04

ENTER FIXED EXPENSE PERCENTAGE(FE)

□:
.06

ENTER MARKET VALUE OF THE PROPERTY (MV)

□:
190000

ENTER PROPERTY TAX EXPENSES (TE) AS PERCENTAGE OF MARKET VALUE

□:
.025

ENTER LAND VALUE(LV)

□:
70000

ENTER BUILDING VALUE(BV)

□:
120000

ENTER INTEREST RATE OR EARNING RATE (EGR).

□:
.08

ENTER EARNING PERIOD OF ECONOMIC LIFE, IN YEARS (ECL)

□:
40

ENTER EXPECTED RESIDUAL VALUE (ERV) AT END OF EARNING PERIOD.
IF YOU CANNOT ESTIMATE RESIDUAL VALUE, USE LAND VALUE(LV).

□:
70000

VALUE ESTIMATE CALCULATIONS USING THE BUILDING RESIDUAL METHOD

LAND INCOME (LVY) 5600
BUILDING INCOME (RBY) 12210
VALUE OF BUILDING (FBV) 116285.7143
VALUE ESTIMATE DERIVED (BPV) 186285.7143

VALUE ESTIMATE CALCULATIONS USING THE LAND RESIDUAL METHOD

BUILDING INCOME (BLY) 12600
LAND INCOME (RLY) 5210
LAND VALUE ESTIMATE (RLV) 65125
VALUE ESTIMATE (RPV) 185125

VALUE ESTIMATE CALCULATIONS, DERIVED USING THE PROPERTY RESIDUE METHOD

PRESENT VALUE OF FUTURE NET INCOME STREAM (PVY) 169619.0476
PRESENT VALUE OF ANY RESIDUAL PROPERTY AT END OF INVESTMENT(DLV)
3222.165331
VALUE ESTIMATE DERIVED (PRV) 172841.213
ANNUAL PERCENT NET INCOME DECLINE WHILE PROTECTING EQUITY EARNING RATE
0.01904761905

PROGRAM DESCRIPTION

```

▽ DIRCAL
[1]  ''
[2]  'ENTER ALL PERCENTAGE FIGURES AS DECIMAL FRACTIONS, E.G.,'
[3]  '8 PERCENT WOULD BE ENTERED AS .08'
[4]  'IF YOU MAKE A MISTAKE ENTERING DATA, START OVER.'
[5]  ''
[6]  'ENTER GROSS SCHEDULED INCOME(GSY)'
[7]  GSY+□
[8]  'ENTER VACANCY AND COLLECTION LOSS (VCL) AS A PERCENTAGE OF
      GROSS SCHEDULED INCOME.'
[9]  VCL+□
[10] 'ENTER OPERATING EXPENSES (OE), AS A PERCENTAGE OF GROSS SCHEDULED INCOME
      .'
[11] OE+□
[12] 'ENTER RESERVE EXPENSE PERCENTAGE(RE)'
[13] RE+□
[14] 'ENTER FIXED EXPENSE PERCENTAGE(FE)'
[15] FE+□
[16] 'ENTER MARKET VALUE OF THE PROPERTY (MV)'
[17] MV+□
[18] 'ENTER PROPERTY TAX EXPENSES (TE) AS PERCENTAGE OF MARKET VALUE'
[19] TE+□
[20] 'ENTER LAND VALUE(LV)'
[21] LV+□
[22] 'ENTER BUILDING VALUE(BV)'
[23] BV+□
[24] 'ENTER INTEREST RATE OR EARNING RATE (EGR).'
[25] EGR+□
[26] 'ENTER EARNING PERIOD OR ECONOMIC LIFE, IN YEARS (ECL)'
[27] ECL+□
[28] 'ENTER EXPECTED RESIDUAL VALUE (ERV) AT END OF EARNING PERIOD.'
[29] 'IF YOU CANNOT ESTIMATE RESIDUAL VALUE, USE LAND VALUE(LV).'

```

```

[47] ' LAND INCOME (LVY) ' ;LVY
[48] ' BUILDING INCOME (RBY) ' ;RBY
[49] ' VALUE OF BUILDING (RBV) ' ;RBV
[50] ' VALUE ESTIMATE DERIVED (BPV) ' ;BPV
[51] ''
[52] 'VALUE ESTIMATE CALCULATIONS USING THE LAND RESIDUAL METHOD'
[53] ' BUILDING INCOME (BLY) ' ;BLY
[54] ' LAND INCOME (RLY) ' ;RLY
[55] ' LAND VALUE ESTIMATE (RLV) ' ;RLV
[56] ' VALUE ESTIMATE (RPV) ' ;RPV
[57] ''
[58] 'VALUE ESTIMATE CALCULATIONS, DERIVED USING THE PROPERTY RESIDUE METHOD'
[59] ' PRESENT VALUE OF FUTURE NET INCOME STREAM (PVY) ' ;PVY
[60] ' PRESENT VALUE OF ANY RESIDUAL PROPERTY AT END OF INVESTMENT(DLV)
';DLV
[61] ' VALUE ESTIMATE DERIVED (PRV) ' ;PRV
[62] ' ANNUAL PERCENT NET INCOME DECLINE WHILE PROTECTING EQUITY EARNING R
ATE ' ;DEY

```

v

SIMULATION OF RANDOM SAMPLES
AND SAMPLING DISTRIBUTIONS OF VARIOUS STATISTICS

John J. Chai
Associate Professor of Quantitative Methods
School of Management
Syracuse University
Syracuse, New York

DESCRIPTION

Introduction

This program is made up of three functions: (1) *POPDIST*, (2) *SIMUSAMPLE*, and (3) *SAMPDIST*. *SIMUSAMPLE* is the main function and the other two are subfunctions. The first function inputs a finite population and formulates the frequency (or probability) distribution with equal class intervals; the second function generates, from the input, simple random samples with replacement of a specified size for a specified number of replications; and the third function formulates the sampling distribution of a specified statistic such as sample mean, median, range, variance, and standard deviation. The program is designed primarily to teach students in a beginning applied statistics course (undergraduate and graduate levels) the concepts of simple random sampling from a finite population, the sampling distribution of a statistic, the central limit theorem, and the relationship between parameters and statistics. This program can also be used for a general simulation problem. The program is intended to be used in a flexible way in that the instructor may use it in a classroom as a teaching device or students may use it independently outside class as an exercise in statistical problems related to sampling distribution theory.

Instructions for Use

It is assumed that the users know what a frequency (or probability) distribution is; what simple random sampling without replacement is; what a sampling distribution is; what the central limit theorem is; and what relationship exists between the statistic involved and the parameters of the population under consideration. The data for input is a set of a finite population. Either a string of individual values (ungrouped data) of a population or a frequency distribution (grouped data) of a population can be entered. If a string of individual values is to be entered as input, the program asks a user to specify the parameters necessary for formulating the ungrouped data into a frequency distribution (i.e., the lower class limit value of the first class, the class interval, and the number of classes required). The program outputs the maximum and minimum values of the input for the user's convenience. If a user wants to input a frequency distribution, the program asks for the class midpoints and corresponding frequencies as input. The program also asks for the population size. If a user inputs a probability distribution, the program asks the user to convert the probabilities to absolute frequencies based on the population size. Then the program outputs a table and a graph showing the details of the population. The table shows lower and upper class limits, class midpoints, absolute and relative frequencies (probabilities), and absolute and relative cumulative frequencies. The graph is in the form of a histogram.

The program also outputs the values of population mean, median, range, variance, and standard deviation. The program then asks the user to input the size of the sample and the number of replications. Next, the program outputs the names of statistics for which sampling distributions can be developed and asks the user to designate one

statistic. The program outputs the maximum and minimum values of the statistic and asks the user to input the lower class limit value of the first class, class interval, and the number of classes required to formulate the sampling distribution. The program then outputs the sampling distribution in tabular and graphic forms, which are very much like the ones given for the population data. It also outputs the various parameters of the sampling distribution.

The program gives the user the option of respecifying the three parameter values necessary for formulating the frequency distribution of the population and the sampling distributions. It also provides the user with the option of continuing the program with a new statistic or a new population.

The constraints of the program are as follows. First, the population size is set to at least 100. This is an arbitrary number. In fact, one should be aware that the population size must be considerably larger than the sample size in order that the effect of the central limit theorem can be realized. On the other hand, one should be aware of workspace size limitations if a string of ungrouped data of a large size population is to be entered. In this case, one is advised to formulate a frequency distribution first and input it instead of inputting ungrouped data. Second, the sample size and the number of replications are also limited to a maximum of 99 and 50, respectively, in view of the workspace size limitation and the high cost of replications. Third, a user must select the lower class limit value of the first class to be slightly less than the minimum value in the data for formulating a frequency distribution of a population and for developing a sampling distribution, for the lower class limit values are not counted into the classes.

To use the program, the student merely needs to type *SIMUSAMPLE*. This first program then calls the others (*POPDIST*

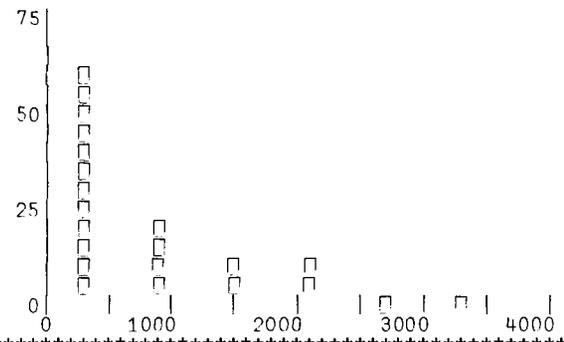
and *SAMPDIST*). In addition to these functions, the public functions *AKI*, *AYN*, *CENTER*, *COLNAMES*, *ΔFMT*, *PLOT*, and *VS* are needed.

PROGRAM USAGE

SIMUSAMPLE
 A POPULATION DATA TO BE ENTERED AS INPUT MAY BE IN THE FORM OF A VECTOR OF INDIVIDUAL VALUES (UNGROUPEDED DATA) OR IN THE FORM OF FREQUENCY (OF PROBABILITY) DISTRIBUTIONS. PLEASE FOLLOW THE INPUT INSTRUCTIONS BELOW CAREFULLY
 HAVE YOU ALREADY ENTERED THE POPULATION DATA? TYPE YES OR NO. N
 ENTER THE POPULATION SIZE. THE SIZE MUST BE AT LEAST 100:
 □: 90
 ENTER THE POPULATION SIZE. THE SIZE MUST BE AT LEAST 100:
 □: 500
 IS YOUR DATA UNGROUPEDED? TYPE 'Y' OR 'N'. Y
 ENTER UNGROUPEDED DATA. TYPE ' ' IF MORE SPACE IS NEEDED FOR ENTERING DATA.
 □: ((1200)+200p10),
 □: (1300)×2+(?300p10]
 SYNTAX ERROR
 (1300)×2+(?300p10]
 ^
 □: (1300)×2+(?300p10)

MAXIMUM VALUE IS: 3267 MINIMUM VALUE IS: 11
 IN ORDER TO SHAPE A FREQUENCY DISTRIBUTION FOR THE UNGROUPEDED DATA YOU ENTERED, YOU ARE ASKED TO SPECIFY THE LOWER CLASS LIMIT VALUE OF THE FIRST CLASS, WIDTH OF CLASS, AND NUMBER OF CLASSES, E.G., 5 20 10. ENTER THESE THREE VALUES:
 □: 6 600 6

POPULATION DISTRIBUTION TABLE							
LOWER CL LMT	UPPER CL LMT	CLASS MID PT	FREQ	REL FREQ	CUMUL FREQ	REL CUM FREQ	
6.00	606.00	306.00	299	59.80	299	59.80	
606.00	1206.00	906.00	90	18.00	389	77.80	
1206.00	1806.00	1506.00	54	10.80	443	88.60	
1806.00	2406.00	2106.00	42	8.40	485	97.00	
2406.00	3006.00	2706.00	11	2.20	496	99.20	
3006.00	3606.00	3306.00	4	0.80	500	100.00	
TOTAL			500	100.00			



PARAMETERS OF POPULATION	
MEAN	771.60
MEDIAN	507.67
RANGE	3600.00
VARIANCE	25439.80
STAND DEV	159.50

 ENTER THE SAMPLE SIZE AND THE NUMBER OF REPLICATIONS DESIRED, E.G., 20 30.
 SAMPLE SIZE MUST BE LESS THAN 100 AND NO. OF REPLICATIONS MUST BE NO GREATER THAN 50.

□: 109 56

TRY AGAIN.
 ENTER THE SAMPLE SIZE AND THE NUMBER OF REPLICATIONS DESIRED, E.G., 20 30.
 SAMPLE SIZE MUST BE LESS THAN 100 AND NO. OF REPLICATIONS MUST BE NO GREATER THAN 50.

□: 10 20
 ENTER THE VARIABLE NAME OF ONE STATISTIC OF YOUR INTEREST.
 THE VARIABLE NAMES ARE AS FOLLOWS:

STATISTICS	VARIABLE
MEAN	SMS
MEDIAN	SMDS
RANGE	SES
VARIANCE	SVS
STAND DEV	SSDS

□: SMS

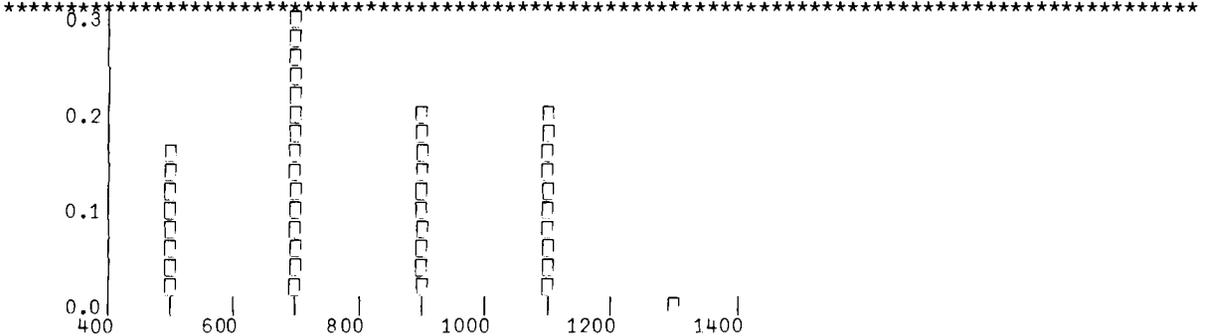
 PARAMETERS OF SAMPLING DISTRIBUTION
 MEAN 744.00
 MEDIAN 756.00
 RANGE 780.00
 VARIANCE 66276.00
 STAND DEV 257.44

SAMPLING DISTRIBUTION OF THE STATISTIC IS PRESENTED IN THE FORM
 OF PROBABILITY DISTRIBUTION AND GRAPH BELOW:
 BUT, FIRST, ENTER THE LOWER CLASS LIMIT VALUE OF THE FIRST CLASS, CLASS WIDTH, AND NO. OF
 CLASSES YOU NEED FOR THE SAMPLING DISTRIBUTION, E.G., 0 5 10:
 (MAXIMUM VALUE OF THE STATISTIC IS: 1146 THE MINIMUM IS: 366):

□: 400 200 5

 SAMPLING DISTRIBUTION

LOWER CL LMT	UPPER CL LMT	CLASS MID PT	PROB
400.00	600.00	500.00	0.15
600.00	800.00	700.00	0.30
800.00	1000.00	900.00	0.20
1000.00	1200.00	1100.00	0.20
1200.00	1400.00	1300.00	0.00
TOTAL			0.85



 WOULD YOU LIKE TO DEVELOP THE SAMPLING DISTRIBUTION FOR A NEW STATISTIC? N
 WOULD YOU LIKE TO RESPECIFY SAMPLE SIZE AND/OR NO. OF REPLICATIONS? N

DO YOU LIKE TO CONTINUE EXERCISE WITH A NEW POPULATION? N

SIMUSAMPLE
 A POPULATION DATA TO BE ENTERED AS INPUT MAY BE IN THE FORM
 OF A VECTOR OF INDIVIDUAL VALUES (UNGROUPED DATA) OR IN THE FORM
 OF FREQUENCY (OR PROBABILITY) DISTRIBUTIONS. PLEASE FOLLOW THE INPUT
 INSTRUCTIONS BELOW CAREFULLY
 HAVE YOU ALREADY ENTERED THE POPULATION DATA? TYPE YES OR NO. N
 ENTER THE POPULATION SIZE. THE SIZE MUST BE AT LEAST 100:

:

500
 IS YOUR DATA UNGROUPED? TYPE 'Y' OR 'N'. N
 ENTER THE GROUPED DATA AS FOLLOWS: IF THE DATA CONTAINS ABSOLUTE FREQUENCIES,
 ENTER THE FREQUENCIES. IF THE DATA CONTAINS RELATIVE FREQUENCIES OR PROBABILITIES,
 CONVERT THEM INTO ABSOLUTE FREQUENCIES AND ENTER THE ABSOLUTE FREQUENCIES:

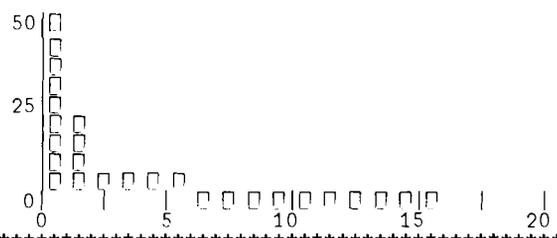
:

253 108 31 26 21 17 7 7 3 9 4 5 5 2 1 1
 ENTER THE LOWER CLASS LIMIT OF THE FIRST CLASS, CLASS INTERVAL, AND NUMBER OF CLASSES IN
 YOUR FREQUENCY DISTRIBUTION TABLE, I.E., 5 20 10.

:

0 1 16

POPULATION DISTRIBUTION TABLE							
LOWER CL LMT	UPPER CL LMT	CLASS MID PT	FREQ	REL FREQ	CUMUL FREQ	REL CUM FREQ	
0.00	1.00	0.50	253	50.60	253	50.60	
1.00	2.00	1.50	108	21.60	361	72.20	
2.00	3.00	2.50	31	6.20	392	78.40	
3.00	4.00	3.50	26	5.20	418	83.60	
4.00	5.00	4.50	21	4.20	439	87.80	
5.00	6.00	5.50	17	3.40	456	91.20	
6.00	7.00	6.50	7	1.40	463	92.60	
7.00	8.00	7.50	7	1.40	470	94.00	
8.00	9.00	8.50	3	0.60	473	94.60	
9.00	10.00	9.50	9	1.80	482	96.40	
10.00	11.00	10.50	4	0.80	486	97.20	
11.00	12.00	11.50	5	1.00	491	98.20	
12.00	13.00	12.50	5	1.00	496	99.20	
13.00	14.00	13.50	2	0.40	498	99.60	
14.00	15.00	14.50	1	0.20	499	99.80	
15.00	16.00	15.50	1	0.20	500	100.00	
TOTAL			500	100.00			



PARAMETERS OF POPULATION	
MEAN	2.15
MEDIAN	0.99
RANGE	16.00
VARIANCE	1.78
STAND DEV	1.33

 ENTER THE SAMPLE SIZE AND THE NUMBER OF REPLICATIONS DESIRED, E.G., 20 30.
 SAMPLE SIZE MUST BE LESS THAN 100 AND NO. OF REPLICATIONS MUST BE NO GREATER THAN 50.

:

30 30

ENTER THE VARIABLE NAME OF ONE STATISTIC OF YOUR INTEREST.
 THE VARIABLE NAMES ARE AS FOLLOWS:

STATISTICS	VARIABLE
MEAN	SMS
MEDIAN	SMDS
RANGE	SRS
VARIANCE	SVS
STAND DEV	SSDS

:

SMS

 PARAMETERS OF SAMPLING DISTRIBUTION

MEAN	2.06
MEDIAN	2.00
RANGE	1.83
VARIANCE	0.20
STAND DEV	0.44

 SAMPLING DISTRIBUTION OF THE STATISTIC IS PRESENTED IN THE FORM
 OF PROBABILITY DISTRIBUTION AND GRAPH BELOW:

BUT, FIRST, ENTER THE LOWER CLASS LIMIT VALUE OF THE FIRST CLASS, CLASS WIDTH, AND NO. OF
 CLASSES YOU NEED FOR THE SAMPLING DISTRIBUTION, E.G., 0 5 10:
 (MAXIMUM VALUE OF THE STATISTIC IS: 2.766666667 THE MINIMUM IS: 0.933333333):

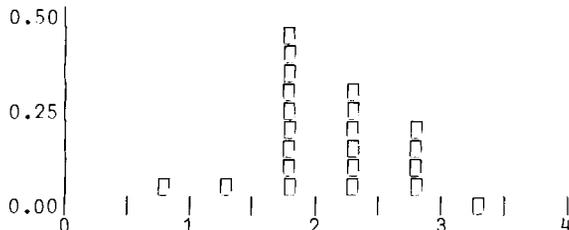
:

.5 .5 6

 SAMPLING DISTRIBUTION

LOWER CL LMT	UPPER CL LMT	CLASS MID PT	PROB
0.50	1.00	0.75	0.03
1.00	1.50	1.25	0.03
1.50	2.00	1.75	0.43
2.00	2.50	2.25	0.30
2.50	3.00	2.75	0.20
3.00	3.50	3.25	0.00

TOTAL 1.00



 WOULD YOU LIKE TO DEVELOP THE SAMPLING DISTRIBUTION FOR A NEW STATISTIC? N
 WOULD YOU LIKE TO RESPECIFY SAMPLE SIZE AND/OR NO. OF REPLICATIONS? N

DO YOU LIKE TO CONTINUE EXERCISE WITH A NEW POPULATION? N

PROGRAM DESCRIPTION

```

V POPDIST;X;CW;F;FORM;K;LC;LCM;PARAMPM;PMD;PR;PV;PSD;RCF;RF;UC
[ 1] THIS FUNCTION REQUIRES THE FOLLOWING PUBLIC FUNCTIONS:
[ 2] AKI, CENTER, COLNAMES, ΔFMT, PLOT, AND VS.
[ 3] TEST:'ENTER THE POPULATION SIZE. THE SIZE MUST BE AT LEAST 100:'
[ 4]   PN+□
[ 5]   +((PN<100),PN≥100)/TEST,GOOD
[ 6] GOOD:ANS+AKI 'IS YOUR DATA UNGROUPED? TYPE 'Y' OR 'N'.'
[ 7]   +(('Y'∈ANS),('N'∈ANS))/L1,L2
[ 8] L1:'ENTER UNGROUPED DATA. TYPE ',□ ' IF MORE SPACE IS NEEDED FOR ENTERING DA
[ 9]   X+□
[10]   ''
[11]   'MAXIMUM VALUE IS: ';/X;' MINIMUM VALUE IS: ';/X
[12]   'IN ORDER TO SHAPE A FREQUENCY DISTRIBUTION FOR THE UNGROUPED DATA'
[13]   'YOU ENTERED, YOU ARE ASKED TO SPECIFY THE LOWER CLASS LIMIT VALUE'
[14]   'OF THE FIRST CLASS, WIDTH OF CLASS, AND NUMBER OF CLASSES, E.G.:'
[15]   ' 5 20 10. ENTER THESE THREE VALUES:'
[16]   P+□
[17]   F+/(1P[3])°.=[(X-P[1])÷P[2] ◊ +L3
[18] L2:'ENTER THE GROUPED DATA AS FOLLOWS: IF THE DATA CONTAINS ABSOLUTE'
[19]   'FREQUENCIES, ENTER THE FREQUENCIES. IF THE DATA CONTAINS RELATIVE'
[20]   'FREQUENCIES OR PROBABILITIES, CONVERT THEM INTO ABSOLUTE FREQUENCIES'
[21]   'AND ENTER THE ABSOLUTE FREQUENCIES:'
[22]   F+□
[23]   'ENTER THE LOWER CLASS LIMIT OF THE FIRST CLASS,CLASS INTERVAL, AND'
[24]   'NUMBER OF CLASSES IN YOUR FREQUENCY DISTRIBUTION TABLE, I.E.,'
[25]   ' 5 20 10.'
[26]   P+□
[27] L3:Y+(P[1]+0.5×P[2])+CW+(((1P[3])-1)×P[2])
[28]   UC+P[2]+LC+P[1]+CW
[29]   RF+100×F÷PN
[30]   RCF+100×((CF+/(1P[3])°.≥1P[3])×(2P[3]-P[3])÷PN)
[31]   PR+1/UC-1/LC
[32]   PSD+(PV+(+(Y-PM+(+/Y×F)÷PN)*2)÷PN)*0.5
[33]   LCM+LC[K+1+(50≤RCF)/(1P[3])]
[34]   +((K=1),K>1)/MD1,MD2
[35] MD1:PMD+LCM+P[2]×50÷RF[K]
[36]   +FORMAT
[37] MD2:PMD+LCM+P[2]×(50-RCF[K-1])÷RF[K]
[38] FORMAT:FORM+3F14.2,I10,F14.2,I10,F14.2'
[39]   STAR+95p '* '
[40]   STAR
[41]   ''
[42]   FORM CENTER 'POPULATION DISTRIBUTION TABLE'
[43]   FORM COLNAMES '/ LOWER/ UPPER/ CLASS/ FREQ/ REL/ CUMUL/REL CUM/'
[44]   FORM COLNAMES '/CL LMT/CL LMT/MID PT/ / FREQ/ FREQ/ FREQ/' ◊ ' '
[45]   FORM ΔFMT(LC;UC;Y;F;RF;CF;RCF) ◊ ' '
[46]   'X37, □TOTAL□,I10 ,F14.2' ΔFMT(+/F;+/RF;)
[47]   ''
[48]   STAR
[49]   HS+1
[50]   EC+ '□ '
[51] 20 40 PLOT RF VS Y
[52]   STAR
[53] PNAME+' MEAN MEDIAN RANGE VARIANCE STAND DEV '
[54] PARAM+PM,PMD,PR,PV,PSD
[55] FORM CENTER 'PARAMETERS OF POPULATION '
[56] 'X20,9A1,F14.2' ΔFMT(5 9 pPNAME;PARAM)
[57]   STAR

```

V

```

V SAMPDIST[ ]V
V SAMPDIST;FORM1;PROB;R;SCW;SLC;SDM;SDMD;SDR;SDV;SDSD;SDY;SUC;XX
[1] THIS FUNCTION REQUIRES THE FOLLOWING PUBLIC FUNCTIONS:
[2] AYN, CENTER, COLNAMES, AND ΔFMT
[3] LOOP: 'ENTER THE VARIABLE NAME OF ONE STATISTIC OF YOUR INTEREST.'
[4] 'THE VARIABLE NAMES ARE AS FOLLOWS:'
[5] 'X20,9A1,X12,9A1' COLNAMES '/STATISTICS/VARIABLE/' ◇ ' '
[6] 'X20,9A1,X12,9A1' ΔFMT((5 9 ρPNAME);(5 9 ρ'SMS SMDS SRS SVS
SSDS '))
[7] XX←[ ]
[8] SDSD←(SDV+(+/(XX-SDM+(+/(XX)÷S[2]))*2)÷S[2])*0.5
[9] SDMD←0.5×+ /XX[(ΔXX)[{f 0.5 0.5 ×1+ρXX]}
[10] SDR←[ /XX-[ /XX
[11] STATS←SDM,SDMD,SDR,SDV,SDSD
[12] STAR
[13] 'X10,9A1,F20.2' CENTER 'PARAMETERS OF SAMPLING DISTRIBUTION'
[14] 'X10,9A1,F20.2' ΔFMT((5 9 ρPNAME);STATS)
[15] STAR
[16] 'SAMPLING DISTRIBUTION OF THE STATISTIC IS PRESENTED IN THE FORM'
[17] 'OF PROBABILITY DISTRIBUTION AND GRAPH BELOW:'
[18] 'BUT, FIRST, ENTER THE LOWER CLASS LIMIT VALUE OF THE FIRST CLASS, CLASS
WIDTH, AND NO. OF'
[19] 'CLASSES YOU NEED FOR THE SAMPLING DISTRIBUTION, E.G., 0 5 10:'
[20] '(MAXIMUM VALUE OF THE STATISTIC IS: ';[ /XX;' THE MINIMUM IS: ';L /XX;'
):'
[21] R←[ ]
[22] SDY←(R[1]+0.5×R[2])+SCW+(((R[3])-1)×R[2])
[23] SUC←R[2]+SLC←R[1]+SCW
[24] PROB←(+/(R[3])°.=[(XX-R[1])÷R[2])÷S[2]
[25] STAR
[26] FORM1←'X10,4F14.2'
[27] FORM1 CENTER 'SAMPLING DISTRIBUTION'
[28] FORM1 COLNAMES '/ LOWER/ UPPER/ CLASS/ PROB/'
[29] FORM1 COLNAMES '/CL LMT/CL LMT/MID PT/ /' ◇ ' '
[30] FORM1 ΔFMT(SLC;SUC;SDY;PROB) ◇ ' '
[31] 'X47, TOTAL, F14.2' ΔFMT(+ /PROB)
[32] STAR
[33] HS←1
[34] EC←'[ ]'
[35] 20 40 PLOT PROB VS SDY
[36] STAR
[37] +LOOP×ANS←AYN 'WOULD YOU LIKE TO DEVELOP THE SAMPLING DISTRIBUTION FOR A
NEW STATISTIC?'

```

V

```

▽ SIMUSAMPLE; ANS; CF; I; J; P; PN; PNAME; S; SAMPLE; SM; SMD; SR; SV; SSD; SMS; SMDs; SRS;
SV S; SSDs; STAR; SY; Y
[1] THIS FUNCTION REQUIRES THE FUNCTIONS AKI, AYN, POPDIST, AND SAMPDIST.
[2] 'A POPULATION DATA TO BE ENTERED AS INPUT MAY BE IN THE FORM
OF A VECTOR OF INDIVIDUAL VALUES (UNGROUPED DATA) OR IN THE FORM
OF FREQUENCY (OR PROBABILITY) DISTRIBUTIONS. PLEASE FOLLOW THE INPUT
INSTRUCTIONS BELOW CAREFULLY'
[3] *SAM*1ANS+AYN 'HAVE YOU ALREADY ENTERED THE POPULATION DATA? TYPE YES OR
NO.'
[4] POP: POPDIST
[5] SAM: 'ENTER THE SAMPLE SIZE AND THE NUMBER OF REPLICATIONS DESIRED, E.G., 2
0 30.'
[6] 'SAMPLE SIZE MUST BE LESS THAN 100 AND NO. OF REPLICATIONS MUST BE NO GRE
ATER THAN 50.'
[7] S+[]
[8] *DATA*1((S[1]<100)^S[2]≤50)
[9] ' TRY AGAIN.'
[10] *SAM
[11] DATA: I+0
[12] SMS+SMDs+SRS+SV S+SSDs+10
[13] REPL: I+I+1
[14] SAMPLE+10
[15] J+0
[16] SAMP: J+J+1
[17] RN+? S p PN
[18] SY+Y[G+1+(RN[J; I]≤CF)/(1P[3])]
[19] SAMPLE+ SAMPLE, SY
[20] *(J<S[1])/ SAMP
[21] SSD+(SV+(+/(SAMPLE-SM+(+ /SAMPLE)÷S[1])*2)÷S[1]-1)*
0.5
[22] SMD+0.5*+ /SAMPLE[(1SAMPLE)[ | [ ^0.5 0.5 *1+pSAMPLE]]
[23] SR+[ /SAMPLE-1 /SAMPLE
[24] SMS+ SMS, SM
[25] SMDs+ SMDs, SMD
[26] SRS+ SRS, SR
[27] SV S+ SV S, SV
[28] SSDs+ SSDs, SSD
[29] *(I<S[2])/ REPL
[30] SAMPDIST
[31] *SAM*1ANS+AYN 'WOULD YOU LIKE TO RESPECIFY SAMPLE SIZE AND/OR NO. OF REPL
ICATIONS?' ◊ 2 1 p' '
[32] ANS+AKI 'WOULD YOU LIKE TO CONTINUE EXERCISE WITH A NEW POPULATION?'
[33] *(( 'Y'∈ANS), 'N'∈ANS)/POP, 0
▽

```


3.6

SETTING THE SIZE OF THE SALES FORCE

Jack R. Dauner
Associate Professor of Marketing
College of Business Administration
The University of Akron
Akron, Ohio

DESCRIPTION

This program facilitates the analysis and determination of the number of salesmen to employ in each sales district and hence in a company's entire sales force. The original mathematical model was developed by Professor Zarrel V. Lambert and published in the book Setting the Size for the Sales Force by the Center for Research of the College of Business Administration, The Pennsylvania State University (1968). The model is based on a case study which uses a company that manufactures and markets medical and industrial X-Ray equipment, accessories, and supplies.

The program *SALESFORCE* can be used to determine the number of salesmen and the estimated change in the company's gross profits where

ECP = Estimated change in the company's gross profits less the change in personal selling costs including commissions.

R = Salesmen's commission rate based on gross profits.

MUI = Gross markup percentage on product cost in district.

PMI = Product mix ratio in district.

PI = Selling price index in district.

SI = Number of Salesmen employed by the company in district.

C = Average out-of-pocket costs per salesmen.

N = Number of salesmen by which the sales force in the district is to be increased or decreased.

$NLOW$ = Minimum number of salesmen in district.

$NMAX$ = Maximum number of salesmen in the district.

To begin the use of the program, type *SALESFORCE* and then enter the inputs as requested. The number of values entered (e.g., the number of sales districts) must be the same for each of the questions or an error condition will result. In the sample shown in the following section, there were seven districts in the company, so seven values had to be entered for each question. In the last question, the number of salesmen to be evaluated is entered.

PROGRAM USAGE

SALESFORCE
 ENTER SALESMAN COMMISSION RATE BASED ON GROSS PROFITS
 □: .05 .05 .05 .05 .05 .05 .05
 ENTER GROSS MARKUP PERCENTAGE ON PRODUCT COST IN DISTRICT
 □: 9 13 12 10 9 4 12
 ENTER PRODUCT MIX RATIO IN DISTRICT
 □: 51.29 13.40 42.03 25.72 19.91 33.48 38.79
 ENTER SELLING PRICE INDEX IN DISTRICT
 □: 109 113 112 110 109 104 112
 ENTER NUMBER OF SALESMEN EMPLOYED IN DISTRICT
 □: 6 2 5 9 4 4 1
 ENTER AVERAGE OUT-OF-POCKET COSTS PER SALESMAN
 □: 8269 8269 8269 8269 8269 8269 8269
 ENTER RANGE OF SALESMEN TO TEST (ENTER MINIMUM AND MAXIMUM)
 □: 1 10

N	ESTIMATED NET CHANGE IN PROFITS							
1	26779946.13	1581457.77	20216653.97	23562427.73	4940102.33	4564756.18	2190749.47	
2	29731047.23	1975811.33	22809048.04	25384674.28	5692961.91	5259882.28	3006285.05	
3	32603351.61	2348825.87	25322386.26	27171391.88	6418860.62	5930092.57	3763592.10	
4	35407418.77	2705654.73	27768583.72	28926155.15	7122441.86	6579678.21	4480323.57	
5	38151481.51	3049551.02	30156637.90	30651932.71	7807069.24	7211747.70	5166278.42	
6	40842123.54	3382742.07	32493563.78	32351225.83	8475282.77	7828648.36	5827642.33	
7	43484719.58	3706841.34	34784971.17	34026168.14	9129062.77	8432210.32	6468650.50	
8	46083733.21	4023067.22	37035439.74	35678599.28	9769992.88	9023897.08	7092370.91	
9	48642925.28	4332369.63	39248772.81	37310120.24	10399365.96	9604903.44	7701120.26	
10	51165503.90	4635508.22	41428174.89	38922135.84	11018255.68	10176221.62	8296704.83	

PROGRAM DESCRIPTION

```

▽ SALESFORCE;C;ECP;ECP1;ECP2;ECP3;ECP4;ECP5;MUI;N;NLOW;NMAX;PI;PMI;R;SI
[1] THIS PROGRAM REQUIRES THE PUBLIC FUNCTION ΔFMT
[2] 'ENTER SALESMAN COMMISSION RATE BASED ON GROSS PROFITS' ◇ R+□
[3] 'ENTER GROSS MARKUP PERCENTAGE ON PRODUCT COST IN DISTRICT' ◇ MUI+□
[4] 'ENTER PRODUCT MIX RATIO IN DISTRICT' ◇ PMI+□
[5] 'ENTER SELLING PRICE INDEX IN DISTRICT' ◇ PI+□
[6] 'ENTER NUMBER OF SALESMEN EMPLOYED IN DISTRICT' ◇ SI+□
[7] 'ENTER AVERAGE OUT-OF-POCKET COSTS PER SALESMAN' ◇ C+□
[8] 'ENTER RANGE OF SALESMEN TO TEST (ENTER MINIMUM AND MAXIMUM)' ◇ IN+□
[9] NMAX+IN[2]
[10] N+IN[1]
[11] ECP+10
[12] A:ECP1+409220750
[13] ECP2+PMI*1.169518
[14] ECP3+PI*-2.645349
[15] ECP4+(SI+N)*0.7847286
[16] ECP5+SI*0.7847286
[17] ECP6+((1-R)*MUI*ECP1*ECP2*ECP3*ECP4*ECP5)-(C*N)
[18] ECP+ECP,N,ECP6
[19] N+N+1
[20] +(N<(NMAX+1))/A
[21] ' N ESTIMATED NET CHANGE IN PROFITS'
[22] ECP+(NMAX,(1+ρR))ρECP
[23] ('I3' ΔFMT ECP[;1]),'F12.2' ΔFMT ECP[;1+1-1+ρECP]
▽

```

SIMULATION OF A THREE STAGE ORDER/SHIPMENT SYSTEM

John L. Harpell
 Assistant Professor of Management
 West Virginia University
 Morgantown, West Virginia

DESCRIPTION

The accompanying function *INVNTRY* is a simple simulation exercise based upon the type of format used by Jay Forrester in his book Industrial Dynamics. The principal use of this program (or function) is to show the concept of simulation and the effects of manipulating decision rules to undergraduate business students. However, it is likely to be of interest to anyone desiring an understanding of simulation. While the program was developed during an introductory course in APL at UCLA, a strong attempt was made to keep the program independent of the UCLA system. Only the plotting ability is not self-contained and it can be ignored.

The system which is to be simulated is shown in Figure 1, where lags 1-3 represent delays from order placement to order shipment and lags 4-6 represent delays from order shipment to order receipt.

Internal to the program are the following variables and relationships:

<u>VARs</u>	<u>MEANING</u>	<u>COMPUTATION</u>
N	Number (period)	Time period counter
BI	Beginning Inventory	= Ending inventory of preceding period
DMND	Demand	Input data
SALES	Amount Sold	Smaller of BI or DMND
BAL	Balance in Stock	BI-SALES (or SHIP)
SHIP	Amount Shipped	Smaller of BI or Order Placed

<u>VAR</u>	<u>MEANING</u>	<u>COMPUTATION</u>
ORD	Order Placed	*200-BAL
REC	Amount Received	SHIP of LAG4,5,or 6 periods prior
EI	Ending Inventory	BAL+REC

*A somewhat arbitrary rule.

It is suggested that the first time the program is run, the user should ask for an explanation each time it is offered. This is done by typing a "2" when asked.

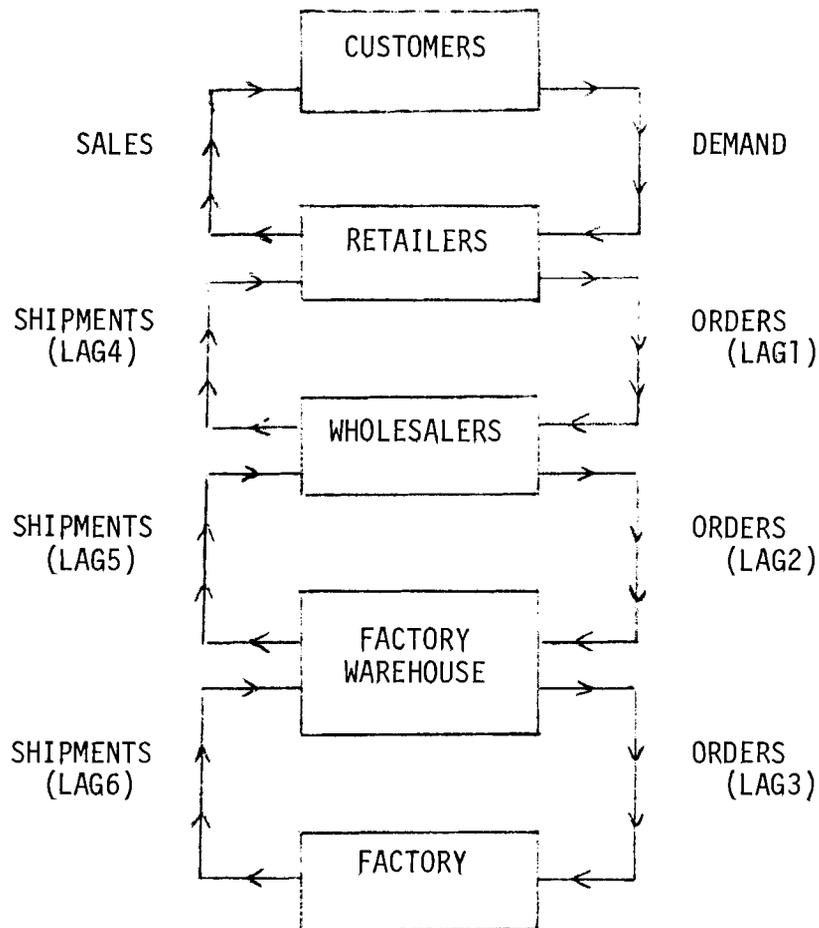


Figure 1

PROGRAM USAGE

INVNTRY

FOR PROGRAM DESCRIPTION, SEE DESCRIBE FUNCTION
SET LAGS? (ENTER 0 FOR NO, 1 FOR YES, AND 2 FOR EXPLANATION.)

□:

2

ENTERING A 1 ALLOWS YOU TO CHANGE THE SIX DELAYS FOR THE ORDER
TO BE SHIPPED FROM A)RETAIL TO WHOLESALE, B)WHOLESALE TO FACTORY, AND
C)FACTORY WAREHOUSE AND WHOLESALE AND F)FACTORY AND FACTORY WAREHOUSE.
ENTER 1 FOR YES AND SIX RESPECTIVE NUMBERS (ENTER SPACES ONLY BETWEEN
THEM). IF YOU ENTER 2 FOR NO, THE DEFAULT IS 2 2 4 0 0 0
SET LAGS? (ENTER 0 FOR NO, 1 FOR YES, AND 2 FOR EXPLANATION.)

□:

1

ENTER 6 LAGS

□:

1 1 4 1 1 0

ENTER NUMBER OF PERIODS FOR PROGRAM TO RUN.(SUGGEST 10 MAX FOR FIRST TRIAL.)

□:

10

SELECT ONE OF THE FOLLOWING OPTIONS AND ENTER THE CORRESPONDING NUMBER:

- 1 I WILL ENTER MY OWN SALES DATA.
- 2 USE AN EQUALLY LIKELY RANDOM NUMBER GENERATOR TO
CREATE SALES DATA.
- 3 USE CONSTANT INCREASE/DECREASE RANDOM NUMBER GENERATOR
TO CREATE SALES DATA.
- 4 WHAT ARE YOU TALKING ABOUT? (EXPLANATION)

□:

4

PROGRAM DESCRIPTION

```

V INVNTFY;LAG1;LAG2;LAG3;LAG4;LAG5;LAG6;REAL;RBI;RDMND;REI;ROFD;RREC;LAG;
  RSALES;SLSV;TDMND;TFACSHIP;TFBAL;TFBI;TFEI;TFREC
[1] ' FOR PROGRAM DESCRIPTION, SEE DESCRIBE FUNCTION'
[2] A THIS IS THE MAIN PROGRAM.
[3] A THE FOLLOWING VARIABLES MAY BE ASSOCIATED WITH THE NUMBERS USED IN THE
[4] A MATRIX(M) COLUMNS AND WITH THE PLOTFORMAT CALLS:
[5] A FBI+1 ◊ RDMND+2 ◊ RSALES+3 ◊ REAL+4 ◊ ROFD+5 ◊ RREC+6 ◊ REI+7
[6] A WBI+8 ◊ WSHIP+9 ◊ WEAL+10 ◊ WORD+11 ◊ WFEC+12 ◊ WEI+13
[7] A FRI+14 ◊ FSHIP+15 ◊ FBAL+16 ◊ FORC+17 ◊ FREC+18 ◊ FEI+19
[8] A NN+21 ◊ FACSHIP+20
[9] A INITIALIZE LAGS(DELAYS) IN ORDEFS AND SHIPMENTS---GO TO GETLAG FUNCTION.
[10] XLAG+GETLAG
[11] LAG1+XLAG[1] ◊ LAG2+XLAG[2] ◊ LAG3+XLAG[3]
[12] LAG4+XLAG[4] ◊ LAG5+XLAG[5] ◊ LAG6+XLAG[6]
[13] A FIND MAXIMUM LAG.
[14] LAG+I/XLAG
[15] A INITIALIZE MATRIX WITH STEADY STATE CONDITIONS.
[16] VINV+(LAG,1)ρ200
[17] VO5+(LAG,5)ρ100
[18] VO4+(LAG,4)ρ100
[19] VINV2+(LAG,2)ρ200
[20] SSM+(-1*(LAG,1)ρ\LAG
[21] M+VINV,VO5,VINV2,VO4,VINV2,SSM
[22] N+0
[23] A MAIN PROGRAM
[24] 'ENTER NUMBER OF PERIODS FOR PROGRAM TO RUN.(SUGGEST 10 MAX FOR FIRST TRI
AL.)'
[25] LOOPS:K+NOPES+[]
[26] A GO TO SIMSALE FUNCTION AND RETURN WITH SLSV(SALES VECTOR).
[27] SLSV+SIMSALE NOPES
[28] Z:TDMND+1+SLSV
[29] SLSV+1+SLSV
[30] A COMPUTE NEW PERIOD VALUES.
[31] NOPES+NOPES-1
[32] TFORD+0[200-TRPAL+TFBI-SALES+TDMND\TFBI+M[1;7]
[33] TWSHIP+(((LAG1=0)*TFORD)+((LAG1≠0)*M[LAG1;5]))\TWBI+M[1;13]
[34] TFEI+TFBAL+TFREC+((LAG4=0)*TWSHIP)+((LAG4≠0)*M[(1\LAG4);9])
[35] TWORD+0[200-TWBAL+TWBI-TWSHIP
[36] TFSHIP+(((LAG2=0)*TWORD)+((LAG2≠0)*M[LAG2;11]))\TFBI+M[1;19]
[37] TWFI+TWBAL+TWREC+((LAG5=0)*TFSHIP)+((LAG5≠0)*M[(1\LAG5);15])
[38] TFORD+0[200-TFBAL+TFBI-TFSHIP
[39] TFACSHIP+((LAG3=0)*TFORD)+((LAG3≠0)*M[LAG3;17])
[40] TFEI+TFBAL+TFREC+((LAG6=0)*TFACSHIP)+((LAG6≠0)*M[(1\LAG6);20])
[41] N+N+1
[42] A PUT PERIOD VALUES IN A VECTOR AND ADD THAT VECTOR TO THE MATRIX.
[43] AVROW1+(TFBI,TDMND,SALES,TFBAL,TFORD,TFREC,TFBI,TWBI,TWSHIP,TWBAL,TWORD,
TFEC,TWEI)
[44] AVROW2+(TFBI,TFSHIP,TFBAL,TFORD,TFREC,TFEI,TFACSHIP,N)
[45] VROW+(AVROW1,AVROW2)
[46] M+VROW,[1] M
[47] +(NOPES>0)/Z
[48] A PRINT TABLE HEADING.
[49] ' | RETAILERS | WHOLESALERS |
FACTORY WAREHOUSE | FACTORY.'
[50] '
[51] ' 1 2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18 19 20'
[52] ' N | FI DMND SALES PAL OFD REC EI | FI SHIP PAL OFD REC EI |
FI SHIP PAL ORD REC EI |SHIP M'
[53] A CATENATE BOTH END COLUMNS--FLIP MATRIX OVER--PRINT RESULTS.
[54] K+iK
[55] MPRINT+ΘM[K;]
[56] MPRINT[;21],MPPINT
[57] SIMPLOT

```

V

```

VGETLAG[ ]V
V XLAG+GETLAG
[1] LLLL:LAG1+2 < LAG2+2 < LAG3+4
[2] LAG4+LAG5+LAG6+0
[3] XLAG+LAG1,LAG2,IAG3,LAG4,IAG5,IAG6
[4] SL:'SET LAGS? (ENTER 0 FOR NO, 1 FOR YES, AND 2 FOR EXPLANATION.)'
[5] JJJJ+[ ]
[6] +(0^JJJJε 0 1 2)/SL
[7] +(JJJJ=2)/EXPAIN
[8] +(JJJJ=0)/0
[9] 'ENTER 6 LAGS'
[10] GET:XLAG+[ ]
[11] +(XLAGε110)/0
[12] 'GIVE 6 NUMBERS, EACH BETWEEN 0 AND 12.'
[13] +GET
[14] E
[15] EXPAIN:'ENTERING A 1 ALLOWS YOU TO CHANGE THE SIX DELAYS FOR THE ORDER'
[16] 'TO BE SHIPPED FROM A)RETAIL TO WHOLSALE, F)WHOLESALE TO FACTORY, AND'
[17] 'C)FACTORY WAREHOUSE AND WHOLESALE AND F)FACTORY AND FACTORY WAREHOUSE.'
[18] 'ENTER 1 FOR YES AND SIX RESPECTIVE NUMBERS (ENTER SPACES ONLY BETWEEN'
[19] 'THEM). IF YOU ENTER 2 FOR NO, THE DEFFAULT IS 2 2 4 0 0 0'
[20] +SL
V
V SLSV+SIMSALE NOPES;SLS
[1] * THIS FUNCTION DETERMINES AND FEEDS SALES DATA TO MAIN FUNCTION(INVNTRY).
[2] BLOB:'SELECT ONE OF THE FOLLOWING OPTIONS AND ENTER THE CORRESPONDING NUMB
ER:'
[3] ''
[4] ' 1 I WILL ENTER MY OWN SALES DATA.'
[5] ''
[6] ' 2 USE AN EQUALLY LIKELY RANDO" NUMBER GENEFEATOR TO'
[7] ' CREATE SALES DATA.'
[8] ''
[9] ' 3 USE CONSTANT INCREASE/DECREASE FANDOM NUMBER GENERATOR'
[10] ' TO CREATE SALES DATA.'
[11] ''
[12] ' 4 WHAT ARE YOU TALKING ABOUT? (EXPLANATION)'
[13] ''
[14] TYPESLS+[ ]
[15] +((TYPESLSε 1 2 3 4)π(ppTYPESLS=ρ10))'BLOB
[16] +(TYPESLS=4)/XXPLAINN
[17] +(TYPESLS=3)/IDCRSE
[18] +(TYPESLS=2)/ELPAND
[19] BOOBOO:'ENTER SALES DATA FOR ';NOPES;' PERIODS. (E.G. 110 110 90 90 )'
[20] SLSV+[ ]
[21] * CHECK MANUALLY INPUT DATA.
[22] +((NOPES<ρSLSV),(NOPES=ρSLSV),(NOPES>ρSLSV))/L1,L2,L3
[23] L1:'TOO MUCH DATA. TRY AGAIN.....'
[24] +BOOBOO
[25] L3:'TOO LITTLE DATA. GIVE IT ANOTHER TRY*****'
[26] +BOOBOO
[27] L2:+0
[28] ELRAND:SLSV+89+?NOPESρ21
[29] +0
[30] IDCRSE:SLSV+100+(?NOPESρ11)-?NOPESρ11
[31] +0
[32] XXPLAINN:' EVERYTHING IN THIS PROGRAM IS IMPLICIT EXCEPT FOR THE'
[33] 'CUSTOMER DEMAND. IN THE PAST THIS DEMAND HAS BEEN 100 PER PERIOD,'
[34] 'RESULTING IN AN EQUILIBRIUM CONDITION (OFTEN CALLED THE STEADY STATE'
[35] 'CONDITION). TO THE EXTENT THAT YOU CHANGE THE DEMAND FROM 100, YOU WILL'

```

```

[36] 'UPSET THIS EQUILIBRIUM. YOU HAVE SEVERAL OPTIONS FOR DETERMINING THE'
[37] 'CUSTOMER DEMAND, NAMELY:'
[38] ''
[39] ' 1. IF YOU ENTER A 1, THE PROGRAM WILL REQUIRE YOU TO FEED IN'
[40] ' THE DEMAND FIGURES FOR EACH OF THE PERIODS. YOU MUST ENTER'
[41] ' A ROW OF FIGURES (SEPARATED BY SPACES) EQUAL IN NUMBER TO'
[42] ' THE NUMBER OF PERIODS YOU HAVE ALREADY ENTERED.'
[43] ' 2. IF YOU ENTER A 2, THE COMPUTER WILL GENERATE A RANDOM NUMBER'
[44] ' FROM A CONSTANT (OR EQUALLY LIKELY) PROBABILITY DISTRIBUTION FO'
[45] ' SALES BETWEEN 90 AND 110. TO CHANGE 90 AND 110 FIND THE PROGRA'
[46] ' LINE ''ELPAND'' IN THE FUNCTION SIMSALE AND CHANGE THE 89 TO'
[47] ' ''(MIN-1)'' AND THE 21 TO ''(RANGE+1)''.'
[48] ' 3. IF YOU ENTER A 3, THE COMPUTER WILL GENERATE A RANDOM NUMBER'
[49] ' FROM A CONSTANTLY INCREASING (FROM 90 TO 100) AND CONSTANTLY'
[50] ' DECREASING (FROM 100 TO 110) PROBABILITY DISTRIBUTION. TO'
[51] ' CHANGE TO 90, 100, AND 110, FIND THE LINE ''IDCFSE'' IN SIMSALE'
[52] ' AND THEN CHANGE THE 100 TO ''MID-RANGE'' AND THE 11 TO'
[53] ' ''RANGE/2 +1''.'
[54] ' 4. NOW YOU KNOW...***'
[55] ''
[56] →BLOP

```

∇

∇ SIMPLOT

```

[1] AA:'DO YOU WANT TO PLOT ANY RESULTS? ENTER 0 FOR NO; 1 FOR YES; 2 FOR EXPL'
    ANATION.'
[2] ZZ+□
[3] →(1≠ZZ∈ 0 1 2)/AA
[4] A EXPLANATION
[5] →(ZZ=2)/EXPLN
[6] A NO
[7] →(ZZ=0)/0
[8] AYES
[9] A UCLA
[10] NN←(1+(NOPES≤60))×pNOPES
[11] YES:'HOW MANY VARIABLES DO YOU WANT TO PLOT? (MAXIMUM OF 3 )'
[12] NUMB+□
[13] →(p10≠ppNUMB)/YES
[14] →(1≠NUMB∈13)/YES
[15] 'THE PERIOD NUMBERS WILL BE ON THE HORIZONTAL AXIS UNLESS YOU CHOSE ONLY'
    ONE VARIABLE.'
[16] ''
[17] ''
[18] ' USE THE COLUMN NUMBERS IN THE PRECEDING TABLE TO ENTER THE'
[19] ' VARIABLES FOR THE HORIZONTAL AXIS (VERTICAL AXIS IS ONLY ONE VARIABLE).'  

[20] '(E.G. FOR RETAIL PI, USE 1; FOR FACTORY WAREHOUSE, USE 19).'  

[21] ''
[22] ''
[23] →(NUMB=1)/ONE
[24] →(NUMB=2)/TWO

```

```

[25] THREE:'WHAT ARE THE THREE VARIABLES? (ENTER WITH SPACES ONLY BETWEEN.)'
[26] TROIS+□
[27] →(3*PTROIS)/THREE
[28] →(3*+/(TROIS€\20))/THREE
[29] G←TROIS[1] ◊ H←TROIS[2] ◊ I←TROIS[3]
[30] A UCLA-NEXT FIVE LINES.
[31] 'O REPRESENTS ';G
[32] '* REPRESENTS ';H
[33] 'O REPRESENTS ';I
[34] 'READ THESE VARIABLES ON THE LEFT AXIS.'
[35] (20,NN) PLOT MPRINT[;G] AND MPRINT[;H] AND MPRINT[;I] VS MPRINT[;
21]
[36] →0
[37] TWO:'WHAT ARE THE TWO VARIABLE NUMBERS? (LIST WITH A SPACE ONLY BETWEEN.)'
[38] DEUX+□
[39] →(2*PDEUX)/TWO
[40] →(2*+/(DEUX€\20))/TWO
[41] G←DEUX[1] ◊ H←DEUX[2]
[42] A UCLA-NEXT FOUR LINES
[43] 'O REPRESENTS ';G
[44] '* REPRESENTS ';H
[45] 'READ THESE VARIABLES ON THE LEFT AXIS.'
[46] (20,NN) PLOT MPRINT[;G] AND MPRINT[;H] VS MPRINT[;21]
[47] →0
[48] ONE:' ENTER VARIABLE NUMBER TO BE PLOTTED ON HORIZONTAL AXIS.'
[49] UNE+□
[50] →(1*PUNE)/ONE
[51] →(1*+/(UNE€\20))/ONE
[52] A UCLA
[53] ONE:PFFT+1
[54] A UCLA
[55] (20,NN) PLOT MPRINT[;UNE] AND MPRINT[;21]
[56] →0
[57] EXPLN:'THIS FUNCTION(SIMPLOT) UTILIZES THE PLOTFORMAT ROUTINE IN'
[58] 'LIBRARY 1 AT UCLA. TO USE, YOU WILL NEED TO HAVE ALREADY LOADED THE'
[59] 'PLOTFORMAT FUNCTION INTO YOUR WORKSPACE. YOU MAY BE ABLE TO USE'
[60] 'PLOTFORMATS FROM OTHER LIBRARIES BY CALLING PFOGAM LINES MARKED BY A'
[61] ''UCLA'' JUST ABOVE THEM.'
[62] →AA

```

∇

MULTIVARIATE T-SQUARE TEST

Edgar P. Hickman
Professor and Program Director of Management Science
University of South Carolina
Columbia, South Carolina 29208

DESCRIPTION

The program *MULTTEST* is designed to test a hypothesis concerning a vector of means using Hotelling's T^2 statistic. It is designed for instructional use in an intermediate or advanced level statistical methods course covering multivariate statistical methods.

The user is required to specify the hypothesis to be tested and to input the sample data. The program then computes and prints the sample variance-covariance matrix and its inverse. Finally the program computes and prints Hotelling's T^2 statistic and the associated F value.

Numerical examples of the multivariate T^2 test concerning means may be found in the following references:

Bolch, B. W. and Huang, C. J., Multivariate Statistical Methods for Business and Economics, Englewood Cliffs: Prentice-Hall, Inc., 1974.

Cooley, W. W. and Lohnes, P. R., Multivariate Data Analysis, New York: John Wiley and Sons, Inc., 1971.

Morrison, Donald F., Multivariate Statistical Methods, New York: McGraw-Hill Book Company, 1967.

Instructions for Use

In order to use *MULTTEST* the user must copy *MULTTEST* into his workspace as well as the supporting functions *COLNAMES*, *DATAMAT*, *INPUT*, *RWTD*, and *ΔFMT*. *DATAMAT* and *INPUT* are functions written especially for use with *MULTTEST* and should always be saved in the

same library.

MULTEST is a niladic function which supplies instructions to the user at each step. Once the functions have been copied into the user's workspace, the function is activated by simply typing *MULTEST* and striking the return key.

PROGRAM USAGE

MULTEST
THIS PROGRAM TESTS THE HYPOTHESIS THAT A VECTOR OF MEANS IS EQUAL TO
A VECTOR OF CONSTANTS. FOR EXAMPLE:
H₀: [U(1)=100, U(2)=75, U(3)=543, ... ,U(P)=25].
WHAT IS THE NUMBER OF VARIABLES FOR THIS PROBLEM?
□:
 3
ENTER THE VALUES FOR YOUR NULL HYPOTHESIS BELOW - ONE VALUE ON EACH LINE.
U(1)=
□:
 10
U(2)=
□:
 20
U(3)=
□:
 30
THE HYPOTHESIS YOU WISH TO TEST IS
U(1) =10
U(2) =20
U(3) =30
ENTER YOUR DATA WITH ONE OBSERVATION VECTOR ON EACH ROW. EACH ROW
SHOULD CONTAIN 3 VARIABLES. BELOW THE LAST ROW TYPE: STOP
□:
 15 16 40
□:
 14 11 38
□:
 18 9 38
□:
 17 16 37
□:
 14 12 31
□:
 9 15 37
□:
 15 16 39
□:
 13 18 33
□:
 16 12 35
□:
 18 11 38
□:
 STOP

YOUR DATA MATRIX CONTAINS 10 ROWS AND 3 COLUMNS. PLEASE CHECK TO SEE THAT IT IS CORRECT.

1	15	16	40
2	14	11	38
3	18	9	38
4	17	16	37
5	14	12	31
6	9	15	37
7	15	16	39
8	13	18	33
9	16	12	35
10	18	11	38

IF DATA MATRIX IS CORRECT TYPE: GOOD

IF DATA MATRIX IS NOT CORRECT TYPE: CHANGE CHANGE

WHAT IS THE NUMBER OF THE ROW YOU WISH TO CHANGE?

□:

6

ENTER THE CORRECT OBSERVATION FOR ROW 6

□:

10 15 38

DO YOU WISH TO CHANGE ANOTHER ROW?

NO

YOUR DATA MATRIX IS

15	16	40
14	11	38
18	9	38
17	16	37
14	12	31
10	15	38
15	16	39
13	18	33
16	12	35
18	11	38

MEAN(1)	MEAN(2)	MEAN(3)
15.00	13.60	36.70

THE SUMS OF SQUARES AND CROSS PRODUCTS ARE

2304	2010	5517
2010	1928	4990
5517	4990	13541

THE SAMPLE VARIANCE COVARIANCE MARTRIX IS

6	-3.333333333	1.333333333
-3.333333333	8.711111111	-0.133333333
1.333333333	-0.133333333	8.011111111

THE INVERSE OF THE SAMPLE VARIANCE COVARIANCE MATRIX IS

0.2212789951	0.08413081272	-0.03542844613
0.08413081272	0.1468119097	-0.01155888295
-0.03542844613	-0.01155888295	0.1305308002
T-SQUARE	F-COMPUTED	
106.4	27.6	

PROGRAM DESCRIPTION

```

▽ R←MULTEST;C;COUNT;D;F;INDEX;INVERSE;KXX;S;SUMS;TSQ
[1]  Ⓜ THIS PROGRAM REQUIRES THE USE OF FUNCTIONS: DATAMAT, COLNAMES,
[2]  Ⓜ INPUT, RWTD, AND ΔFMT.
[3]  Ⓜ THE VARIABLES N, X, MEAN, AND SS ARE GLOBAL VARIABLES SINCE THEY ARE
[4]  Ⓜ REQUIRED IN BOTH DATAMAT AND MULTEST.
[5]  'THIS PROGRAM TESTS THE HYPOTHESIS THAT A VECTOR OF MEANS IS EQUAL TO'
[6]  'A VECTOR OF CONSTANTS. FOR EXAMPLE:'
[7]  '      H0: [U(1)=100, U(2)=75, U(3)=543, ... ,U(P)=25].'
[8]  'WHAT IS THE NUMBER OF VARIABLES FOR THIS PROBLEM?'
[9]  P←□
[10] 'ENTER THE VALUES FOR YOUR NULL HYPOTHESIS BELOW - ONE VALUE ON EACH LIN
    E.'
[11] U←10
[12] COUNT←1
[13] Ⓜ THIS LOOP REQUESTS THE USER TO INPUT THE HYPOTHETICAL VALUES FOR MU(1) T
    O MU(P). ΔFMT IS REQUIRED.
[14] LOOP:U←U,INPUT 'U(□,I2,□)=□' ΔFMT(COUNT) ◊ COUNT←COUNT+1
[15]   +LOOP×1(COUNT≤P)
[16] 'THE HYPOTHESIS YOU WISH TO TEST IS'
[17] 'U(1) =' ;1+U ◊ NUMBER←2
[18] LP1:'U(';NUMBER;' ) =' ;1+(NUMBER-1)φU ◊ NUMBER←NUMBER+1
[19]   +LP1×1(NUMBER≤P)
[20] DATAMAT
[21] SUMS←(+X)
[22] C←MEANS◦.×SUMS
[23] KXX←SS-C
[24] S←KXX÷(N-1)
[25] ' '
[26] 'THE SAMPLE VARIANCE COVARIANCE MARTRIX IS'
[27] ' '
[28] S
[29] ' '
[30] 'THE INVERSE OF THE SAMPLE VARIANCE COVARIANCE MATRIX IS'
[31] ' '
[32] □←INVERSE+□S
[33] D←(MEANS-U)
[34] TSQ←N×D+.×INVERSE+.×(□D)
[35] Ⓜ THE FOLLOWING LINES FORMAT THE PRINTING OF THE VALUES FOR T-SQUARE AND F
[36] F←TSQ×(N-P)÷(N-1)×P
[37] FS+'2F12.1'
[38] ' '
[39] FS COLNAMES ' /T-SQUARE/F-COMPUTED'
[40] ' '
[41] FS ΔFMT(TSQ;F)

```

▽

```

▽ R←DATAMAT;FS;H;INDEX;V
[1] 'ENTER YOUR DATA WITH ONE OBSERVATION VECTOR ON EACH ROW. EACH ROW '
[2] 'SHOULD CONTAIN ';P;' VARIABLES. BELOW THE LAST ROW TYPE: STOP'
[3] STOP←35.34758697821
[4] X←(0,P)ρ0
[5] REPEAT:V←,□
[6] +(STOP=1↑V)/NEXT
[7] +(P=ρV)/ADD
[8] 'ERROR'
[9] →REPEAT
[10] ADD:X←X,[1](1,P)ρV
[11] →REPEAT
[12] NEXT:N←1↑(ρX)
[13] 'YOUR DATA MATRIX CONTAINS ';N;' ROWS AND ';P;' COLUMNS. PLEASE CHECK'
[14] 'TO SEE THAT IT IS CORRECT.'
[15] ' '
[16] ((N,1)ρ\N),X
[17] 'IF DATA MATRIX IS CORRECT TYPE: GOOD
      IF DATA MATRIX IS NOT CORRECT TYPE: CHANGE'
[18] CHECK←□
[19] +('GOOD'=4↑CHECK)/PRODUCT
[20] CHANGE:'WHAT IS THE NUMBER OF THE ROW YOU WISH TO CHANGE?'
[21] INDEX←□
[22] 'ENTER THE CORRECT OBSERVATION FOR ROW ';INDEX
[23] X[INDEX;]←□
[24] 'DO YOU WISH TO CHANGE ANOTHER ROW?'
[25] +('YES'=3↑□)/CHANGE
[26] 'YOUR DATA MATRIX IS'
[27] ' '
[28] X
[29] ' '
[30] PRODUCT:MEANS+(+/X)÷1↑ρX
[31] SS←(QX)+.×X
[32] H←,'□/MEAN(□,I2,□)□' ΔFMT(\P)
[33] ' '
[34] FS←,'I2,□F10.2□' ΔFMT P
[35] FS COLNAMES H
[36] FS ΔFMT((1,P)ρMEANS)
[37] ' '
[38] 'THE SUMS OF SQUARES AND CROSS PRODUCTS ARE'
[39] ' '
[40] SS

```

▽

```

▽ R←INPUT PROM
[1] I1:PROM
[2] +(1=ρR←,□)/0
[3] 'ENTER ONE VALUE ONLY'
[4] →I1

```

▽

CALCULATION OF DISABILITY SETTLEMENT AWARDS

R. Harold Mason
Graduate School of Management
University of California
Los Angeles, California

DESCRIPTION

LUMPSUM is a program designed to discount future streams of income, medical costs, retirement annuities, etc., and reduce these to a lump sum payment in liability cases. For example, a person injured in an automobile accident may have:

1. Lost income between time of injury and settlement of a liability case.
2. Incurred uninsured medical costs during the same period.
3. Been permanently disabled to some degree and therefore lose future income, promotions, and retirement benefits and incur both uninsured maintenance medical costs and periodic uninsured major medical costs.

LUMPSUM will handle this problem and can be used by students, lawyers and economists who wish to calculate the net present value of medical costs and lost income due to the disability of an injured person.

LUMPSUM is a prompted program. The inputs required are:

1. Annual income of the injured person at the time of injury.
2. Age at the time of injury.
3. Present age.

4. Risk-free current interest rate (or some agreed upon interest rate acceptable to litigants), in percentage.
5. Long-run improvement in productivity (real percentage increase in per capita income).
6. Employer contribution to retirement annuity, in percentage of income at time of injury.
7. Age of the injured at the time of future promotions that the employer could have anticipated for the injured.
8. Each promotion as a percentage of income at that age.
9. Based upon current age, the remaining expected life, taken from mortality tables.
10. Continuing uninsured maintenance medical costs as a percentage of income at time of injury due to injury. (For example, the injured may require continuing physical therapy throughout his or her life.)
11. Major future uninsured medical costs: age when they are to be incurred and the dollar amounts. (For example, physicians may suggest the need for one or two major operations to correct the more serious physical aspects of the disability. If so, they should provide current dollar estimates which are then adjusted for inflation at a rate equal to the productivity factor.)
12. Lost income between the time of injury and the time of settlement resulting from the injury.

13. Uninsured medical costs incurred between the time of injury and the settlement.
14. Interest rates for each year of lost income or uninsured medical costs incurred between the time of injury and the settlement.
15. Degree of disability in percentage. (For example, physicians may estimate that the injured has lost 25% of his or her capacity to continue earning income and pursuing a career.)

The outputs are:

1. The discounted value of future income, including promotions.
2. The discounted value of employer contributions to retirement.
3. The discounted value of uninsured medical maintenance costs, adjusted for cost escalation.
4. The value of lost income and uninsured medical costs between the time of injury and the settlement, compounded by the past annual interest rate.
5. The discounted value of uninsured major medical costs, adjusted for cost escalation.
6. The value of the award, adjusted for the degree of continuing disability.

To illustrate the use of the program, suppose you have the following case:

- | | |
|-----------------------------|----------|
| 1. Income at time of injury | \$10,000 |
| 2. Age at time of injury | 30 yrs. |
| 3. Present age | 35 yrs. |

4.	Current interest rate		6%
5.	Rate of productivity improvement		3%
6.	Employer's contribution to retirement		12% of income
7.	Anticipated promotions	Age	Percent of Income
		40 yrs.	10%
		45	8%
		50	6%
		55	5%
		60	4%
8.	Remaining life expectancy		35 yrs.
9.	Uninsured maintenance medical costs		10% of income
10.	Future major medical costs	Age	Dollars
		40 yrs.	\$3000
		45	\$2500
11.	Lost income uninsured medical costs, and interest rates since injury.		
	Age	Lost Income	Uninsured Medical Costs
			Interest Rate
	30 yrs.	\$1200	\$500
	31	\$1300	\$600
	32	\$1500	\$400
	33	\$1200	\$600
	34	\$1300	\$500
	35	\$1200	\$600
12.	Degree of disability, in percentage		25%

Using these data, the following section shows a sample run.

PROGRAM USAGE

LUMPSUM
 ENTER INCOME AT TIME OF INJURY
: 10000
 ENTER AGE AT TIME OF INJURY AND PRESENT AGE
: 30 35
 ENTER CURRENT INTEREST RATE IN PERCENTAGE
: 6
 ENTER PRODUCTIVITY FACTOR IN PERCENTAGE
: 3
 ENTER RETIREMENT ANNUITY IN PERCENTAGE OF CURRENT INCOME
: 12
 ENTER AGE AT PROMOTION, PERCENTAGE PROMOTION WOULD BE AT THEN INCOME
 ENTER ALL PROMOTIONS AND WHEN DONE JUST HIT CARRIAGE RETURN

	AGE	PERCENT
ENTRY:	40	10
ENTRY:	45	8
ENTRY:	50	6
ENTRY:	55	5
ENTRY:	60	4

ENTRY:
 BASED ON CURRENT AGE, ENTER REMAINING YEARS OF LIFE EXPECTANCY FROM MORTALITY TABLES
: 35
 ENTER BASIC UNINSURED MAINTENANCE MEDICAL COSTS AS PERCENT OF INCOME AT TIME OF INJURY
: 10
 ENTER ESTIMATED AGE AT INCURRENCE OF FUTURE MAJOR MEDICAL COSTS AND ENTER DOLLAR ANNUAL AMOUNTS OF COSTS VALUED IN PRESENT DOLLARS WHEN ALL MAJOR MEDICAL COST OCCURENCES ARE ENTERED JUST HIT RETURN

	AGE	DOLLARS
ENTRY:	40	3000
ENTRY:	45	2500

ENTRY:
 FOR AGE INDICATED SINCE INJURY ENTER ANNUAL LOST INCOME, UNINSURED MEDICAL COSTS IN DOLLAR VALUES AND INTEREST RATES IN PERCENT.

	AGE	LOST INCOME	UNINS MED COSTS	INTEREST RATE
AGE:	30	1200	500	5
AGE:	31	1300	600	4
AGE:	32	1500	400	6
AGE:	33	1200	600	7
AGE:	34	1300	500	6
AGE:	35	1200	600	6

ENTER PERCENTAGE DEGREE OF DISABILITY
: 25

THE DISCOUNT RATES ARE:

0.9708737864	0.9425959091	0.9151416594	0.8884870479	0.8626087844
0.9275363273	0.9005207061	0.8742919477	0.8488271337	0.8241040133
0.8674779087	0.8422115619	0.817681128	0.7938651728	0.7707428862
0.7945802951	0.7714371797	0.7489681356	0.7271535297	0.7059743007
0.7203819395	0.6993999413	0.6790290692	0.6592515235	0.6400500228
0.6465151746	0.6276846355	0.6094025587	0.5916529696	0.5744203589
0.5576896688				

THE DISCOUNTED VALUE OF FUTURE INCOME IS:238005.5728

THE DISCOUNTED VALUE OF THE RETIREMENT ANNUITY IS:28560.66873

THE DISCOUNTED VALUE OF EXPECTED MAINTENANCE MEDICAL COSTS IS:22487.22007

THE VALUE OF LOST INCOME AND MEDICAL COSTS SINCE INJURY PLUS INTEREST IS:
13299.63044

THE DISCOUNTED VALUE OF FUTURE MAJOR MEDICAL EXPENSES IS:4448.06114

THE AWARD IS:233425.7725

PROGRAM DESCRIPTION

```

▽ LUMPSUM
[1] THIS FUNCTION REQUIRES THEE USE OF PUBLIC FUNCTIONS INP AND IPI
[2] 'ENTER INCOME AT TIME OF INJURY' ◇ INC+INP 1
[3] 'ENTER AGE AT TIME OF INJURY AND PRESENT AGE' ◇ AGE+INP 2
[4] 'ENTER CURRENT INTEREST RATE IN PERCENTAGE' ◇ RATE+(INP 1)÷100
[5] 'ENTER PRODUCTIVITY FACTOR IN PERCENTAGE' ◇ PROD+(INP 1)÷100
[6] 'ENTER RETIREMENT ANNUITY IN PERCENTAGE OF CURRENT INCOME'
[7] RET+(INP 1)÷100
[8] PROM+ 2 0 p0
[9] 'ENTER AGE AT PROMOTIOM, PERCENTAGE PROMOTION WOULD BE AT THEN INCOME '
[10] 'ENTER ALL PROMOTIONS AND WHEN DONE JUST HIT CARRIAGE RETURN'
[11] '     AGE     PERCENT '
[12] L0:*(L0+3)*1^/0=IN+2+IPI 'ENTRY: '
[13] PROM+PROM,IN
[14] +L0
[15] 'BASED ON CURRENT AGE, ENTER REMAINING YEARS OF LIFE EXPECTANCY FROM'
[16] 'MORTALITY TABLES'
[17] RLE+INP 1
[18] 'ENTER BASIC UNINSURRED MAINTENANCE MEDICAL COSTS AS PERCENT OF INCOME'
[19] 'AT TIME OF INJURY'
[20] MMC+(INP 1)÷100
[21] MAJMED+ 2 0 p0
[22] 'ENTER ESTIMATED AGE AT INCURRENCE OF FUTURE MAJOR MEDICAL COSTS AND'
[23] 'ENTER DOLLAR ANNUAL AMOUNTS OF COSTS VALUED IN PRESENT DOLLARS'
[24] 'WHEN ALL MAJOR MEDICAL COST OCCURENCES ARE ENTERED JUST HIT RETURN'
[25] '     AGE     DOLLARS '
[26] L1:*(L1+3)*1^/0=IN+2+IPI 'ENTRY: '
[27] MAJMED+MAJMED,IN
[28] +L1
[29] MEDINC+ 3 0 p0 ◇ T+AGE[1]
[30] 'FOR AGE INDICATED SINCE INJURY ENTER ANNUAL LOST INCOME, UNINSURED'
[31] 'MEDICAL COSTS IN DOLLAR VALUES AND INTEREST RATES IN PERCENT. '
[32] '     LOST INCOME     UNINS MED COSTS     INTEREST RATE '
[33] L2: MEDINC+MEDINC,3+IPI 'AGE: ',('0123456789'[1+ 10 10 T]),' '
[34] +L2*1AGE[2]≥T+T+1
[35] 'ENTER PERCENTAGE DEGREE OF DISABILITY' ◇ DIS+(100-INP 1)÷100
[36] ELOSS+(N+66-AGE[2])p1
[37] ELOSS[YRS]+ELOSS[YRS+PROM[1;]-AGE[2]-1]-PROM[2;]÷100
[38] ELOSS+ELOSS+(RATE-PROD)
[39] EINC+INC*+ /ELOSS*+ \÷ELOSS
[40] RET+EINC*RET
[41] EMMC+(MMC*INC)*+ /÷(1+RATE-PROD)*0, \RLE
[42] EMEDINC+(MEDINC[1;]+MEDINC[2;])*+ . *φ* \÷1-MEDINC[3;]÷100
[43] EMLOSS+MAJMED[2;]+ . *÷(1+RATE-PROD)*MAJMED[1;]-AGE[2]
[44] TOTAL+EINC+RET+EMMC+EMLOSS
[45] AWARD+EMEDINC+TOTAL*DIS
[46] 'THE DISCOUNT RATES ARE: '
[47] '     ;ELOSS
[48] 'THE DISCOUNTED VALUE OF FUTURE INCOME IS: ';EINC
[49] 'THE DISCOUNTED VALUE OF THE RETIREMENT ANNUITY IS: ';RET
[50] 'THE DISCOUNTED VALUE OF EXPECTED MAINTENANCE MEDICAL COSTS IS: ';EMMC
[51] 'THE VALUE OF LOST INCOME AND MEDICAL COSTS SINCE INJURY PLUS INTEREST IS
: ';EMEDINC
[52] 'THE DISCOUNTED VALUE OF FUTURE MAJOR MEDICAL EXPENSES IS: ';EMLOSS
[53] 'THE AWARD IS: ';AWARD

```


LINEAR PROGRAMMING WITH MIXED CONSTRAINTS

Ronald D. Miller
 Assistant Dean of Systems
 College of Business and Public Service
 Governors State University
 Park Forest South, Illinois

DESCRIPTION

Linear programming is a mathematical technique for finding the best use of a firm's limited resources. The adjective "linear" is used to describe a relationship between two or more variables, a relationship which is directly and precisely proportional. For example, linear means that a 10 percent change in the number of productive hours used in some operation will cause a 10 percent change in output. "Programming" refers to the use of certain mathematical techniques to get the best solution to a problem involving limited resources.

The APL function which initiates these linear programming calculations is entitled *SIMPLEX* (after the name of the technique which is employed). A complete list of all of the functions which are required for successful operation are given.

<u>Function Name</u>	<u>Description</u>
<i>SIMPLEX</i>	This function accepts and validates user input.
<i>SIMPLEX1</i>	This function develops the LP matrix and performs the computations necessary for the solution of the LP problem.

<u>Function Name</u>	<u>Description</u>
<i>SOLI</i>	This function solves the matrix by the Barnes M Method.
<i>PRINT</i>	This function provides the printed output as well as the format features.
<i>REASON</i>	This function contains author information.
<i>DECSLP</i>	This function gives a brief verbal description of linear programming (similar to what is contained in this section)
<i>SAMPLP</i>	This function contains a sample problem, as well as explanations and examples of data input.

These functions have comprehensive built-in instructions and prompts for the user, and so there is no need to explain their detailed usage here. The user merely needs to type *SIMPLEX* at the terminal and the functions, through a series of interactive questions, will request the data necessary to execute the program. Thus the user need not have any special experience with either APL or linear programming in order to use this series of functions.

PROGRAM USAGE

SIMPLEX

TURN CARRIAGE TO POSITION PAPER AT START OF NEW PAGE, THEN PRESS RETURN.

LINEAR PROGRAMMING - MIXED CONSTRAINTS

DO YOU KNOW HOW TO USE THIS PROGRAM? YES

ENTER A PROJECT TITLE IF YOU WISH. DOCUMENTATION RUN

IS THIS A MAXIMIZATION OR MINIMIZATION PROBLEM? REPLY MAX OR MIN. MIN

ENTER THE NUMBER OF CONSTRAINTS, AND THE NUMBER OF VARIABLES: 4 3

ENTER COEFFICIENTS OF SIMPLEX PROBLEM, ONE ROW AT A TIME: 1 1 1 = 1000

ENTER NEXT CONSTRAINT: 1 0 0 ≤ 300

ENTER NEXT CONSTRAINT: 0 1 0 ≤ 150

ENTER NEXT CONSTRAINT: 0 0 1 ≥ 200

ENTER PROFIT OR COST FUNCTION COEFFICIENTS: 5 6 7

TURN CARRIAGE TO POSITION PAPER AT START OF NEW PAGE, THEN PRESS RETURN.

LINEAR PROGRAMMING WITH MIXED CONSTRAINTS - MINIMIZATION

DOCUMENTATION RUN

TABLEAU 1

VAR	Z	X1	X2	X3	X4	X5	X6	X7	X8	X9
4	1000.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00
5	300.00	1.00	0.00	0.00	0.00	1.00	-0.00	0.00	0.00	0.00
7	150.00	0.00	1.00	0.00	0.00	0.00	1.00	1.00	-0.00	0.00
9	200.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	1.00	1.00
0	135000000.00	100000.00	200000.00	200000.00	100000.00	0.00	100000.00	100000.00	-100000.00	100000.00

TABLEAU 2

VAR	Z	X1	X2	X3	X4	X5	X6	-X7	X8	X9
4	850.00	1.00	0.00	1.00	1.00	0.00	1.00	-1.00	0.00	0.00
5	300.00	1.00	0.00	0.00	0.00	1.00	-0.00	0.00	0.00	0.00
2	150.00	0.00	1.00	0.00	0.00	0.00	-1.00	1.00	-0.00	0.00
9	200.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	-1.00	1.00
0	105000900.00	100000.00	6.00	200000.00	100000.00	0.00	99994.00	-99994.00	-100000.00	100000.00

TABLEAU 3

VAR	Z	X1	X2	X3	X4	X5	X6	-X7	X8	-X9
4	650.00	1.00	0.00	0.00	1.00	0.00	1.00	-1.00	1.00	-1.00
5	300.00	1.00	0.00	0.00	0.00	1.00	-0.00	0.00	0.00	0.00
2	150.00	0.00	1.00	0.00	0.00	0.00	-1.00	1.00	-0.00	0.00
3	200.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	1.00	-1.00
0	65002300.00	100000.00	6.00	7.00	100000.00	0.00	99994.00	-99994.00	99993.00	-99993.00

TABLEAU 4

VAR	Z	X1	X2	X3	X4	-X5	X6	-X7	X8	-X9
4	350.00	0.00	0.00	0.00	1.00	-1.00	1.00	-1.00	1.00	-1.00
1	300.00	1.00	0.00	0.00	0.00	1.00	-0.00	0.00	0.00	0.00
2	150.00	0.00	1.00	0.00	0.00	0.00	-1.00	1.00	-0.00	0.00
3	200.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	-1.00	1.00
0	35003800.00	5.00	6.00	7.00	100000.00	-99995.00	99994.00	-99994.00	99993.00	-99993.00

TABLEAU 5

VAR	Z	X1	X2	X3	X4	-X5	X6	-X7	X8	-X9
6	350.00	0.00	0.00	0.00	1.00	-1.00	1.00	-1.00	1.00	-1.00
1	300.00	1.00	0.00	0.00	0.00	-1.00	0.00	0.00	0.00	-0.00
2	150.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	-1.00	-1.00
3	200.00	0.00	0.00	1.00	0.00	-0.00	0.00	0.00	-1.00	1.00
0	5900.00	5.00	6.00	7.00	6.00	-1.00	0.00	0.00	-1.00	1.00

OPTIMAL SOLUTION FOR MINIMIZATION

THE OPTIMUM VALUE OF Z IS 5900.00

VAR	VALUE
6	350.00
1	300.00
2	500.00
3	200.00

LINEAR PROGRAMMING COMPLETED

SIMPLEX

TURN CARRIAGE TO POSITION PAPER AT START OF NEW PAGE, THEN PRESS RETURN.

LINEAR PROGRAMMING - MIXED CONSTRAINTS

DO YOU KNOW HOW TO USE THIS PROGRAM? YES

ENTER A PROJECT TITLE IF YOU WISH. DOCUMENTATION RUN

IS THIS A MAXIMIZATION OR MINIMIZATION PROBLEM? REPLY MAX OR MIN. MAX

ENTER THE NUMBER OF CONSTRAINTS, AND THE NUMBER OF VARIABLES: 2 2

ENTER COEFFICIENTS OF SIMPLEX PROBLEM, ONE ROW AT A TIME: 4 2 ≤ 60

ENTER NEXT CONSTRAINT: 2 4 ≤ 48

ENTER PROFIT OR COST FUNCTION COEFFICIENTS: 8 6

TURN CARRIAGE TO POSITION PAPER AT START OF NEW PAGE, THEN PRESS RETURN.

LINEAR PROGRAMMING WITH MIXED CONSTRAINTS - MAXIMIZATION

DOCUMENTATION RUN

TABLEAU 1

VAR	Z	X1	X2	X3	X4
3	60.00	4.00	2.00	1.00	0.00
4	48.00	2.00	4.00	0.00	1.00
0	0.00	0.00	0.00	0.00	0.00

TABLEAU 2

VAR	Z	X1	X2	X3	X4
1	15.00	1.00	0.50	-0.25	0.00
4	18.00	0.00	3.00	-0.50	1.00
0	120.00	8.00	4.00	2.00	0.00

TABLEAU 3

VAR	Z	X1	X2	X3	X4
1	12.00	1.00	0.00	-0.333	-0.17
2	6.00	0.00	1.00	-0.17	0.33
0	132.00	8.00	6.00	1.67	0.67

OPTIMAL SOLUTION FOR MINIMIZATION

THE OPTIMUM VALUE OF Z IS 132.00

VAR VALUE

1 12.00
2 6.00

LINEAR PROGRAMMING COMPLETED

PROGRAM DESCRIPTION

▽ REASON

[1] '
[2] 'THE LINEAR PROGRAMMING APPLICATION THAT FOLLOWS HAS BEEN WRITTEN AS ONE
IN'
[3] 'A SERIES OF PROGRAMS RESULTANT FROM:'
[4] '
[5] ' INTERACTIVE COMPUTING FOR MANAGEMENT EDUCATORS'
[6] ' JUNE 29 TO JULY 11, 1975'
[7] ' GRADUATE SCHOOL OF MANAGEMENT'
[8] ' UNIVERSITY OF CALIFORNIA AT LOS ANGELES (UCLA)'
[9] '
[10] 'THIS PROGRAM WAS AUTHORED BY:'
[11] ' '
[12] ' RONALD D. MILLER'
[13] ' ASSISTANT DEAN OF SYSTEMS'
[14] ' COLLEGE OF BUSINESS AND PUBLIC SERVICE'
[15] ' GOVERNORS STATE UNIVERSITY'
[16] ' PARK FOREST SOUTH, ILLINOIS 60466'
[17] '
▽

▽DESCLP[]▽

PA: PLEASE SIGN OFF BY 1530 FOR A SYSTEM RESTART.

▽ DESCLP

[1] ''
[2] 'LINEAR PROGRAMMING IS A MATHEMATICAL TECHNIQUE FOR FINDING THE BEST USE
'
[3] 'OF A FIRM'S LIMITED RESOURCES. THE ADJECTIVE LINEAR IS USED TO'
[4] 'A RELATIONSHIP BETWEEN TWO OR MORE VARIABLES, A RELATIONSHIP WHICH IS'
[5] 'DIRECTLY AND PRECISELY PROPORTIONAL. FOR EXAMPLE, LINEAR MEANS'
[6] 'THAT A 10 PERCENT CHANGE IN THE NUMBER OF PRODUCTIVE HOURS USED IN SOME'
[7] 'OPERATION WILL CAUSE A 10 PERCENT CHANGE IN OUTPUT. PROGRAMMING'
[8] 'REFERS TO THE USE OF CERTAIN MATHEMATICAL TECHNIQUES TO GET THE '
[9] 'BEST SOLUTION TO A PROBLEM INVOLVING LIMITED RESOURCES.'
[10] ''
▽

```

▽ SIMPLEX
[1] THIS PROGRAM REQUIRES THE FUNCTIONS DESCLP, REASON,
[2] RSAMPL, SEPERATE, AND SIMPLEX1; AND THE PUBLIC FUNCTIONS
[3] RAKI, AYN, AND IPI.
[4] PGS←AKI '
    TURN CARRIAGE TO POSITION PAPER AT START OF NEW PAGE, THEN PRESS RETURN.'
[5] '
    ,
[6] 'LINEAR PROGRAMMING - MIXED CONSTRAINTS
    ,
[7] R      AUTHOR:      RONALD D. MILLER
[8] R      ASSISTANT DEAN OF SYSTEMS
[9] R      COLLEGE OF BUSINESS AND PUBLIC SERVICE
[10] R     GOVERNORS STATE UNIVERSITY
[11] R     PARK FOREST SOUTH, ILLINOIS 60466
[12] R
[13] PGS←AYN 'DO YOU KNOW HOW TO USE THIS PROGRAM?'
[14] +(1=PGS)/L1
[15] REASON ◇ DESCLP ◇ SAML P
[16] L1:TITLE←AKI '
    ENTER A PROJECT TITLE IF YOU WISH.'
[17] L111:''
[18] REPLY←AKI '
    IS THIS A MAXIMIZATION OR MINIMIZATION PROBLEM? REPLY MAX OR MIN. '
[19] +((('A'≠REPLY)^( 'I'≠REPLY+1+REPLY+2+REPLY))/ERR1
[20] L112:''
[21] V←,IPI '
    ENTER THE NUMBER OF CONSTRAINTS, AND THE NUMBER OF VARIABLES:'
[22] +((2=ρV),2≠ρV)/L4,ERR2
[23] L4:C+V[2] ◇ R+V[1]
[24] MAT+0ρ0 ◇ C1+C+1 ◇ OP+0ρ''
[25] ''
[26] L114:''
    ,
[27] PARM←AKI 'ENTER COEFFICIENTS OF SIMPLEX PROBLEM, ONE ROW AT A TIME:'
[28] →LP2
[29] LP:''
[30] PARM←AKI '                                ENTER NEXT CONSTRAINT:'
[31] LP2:SEPARATE PARM
[32] ALPH+,ALPH[1;1]
[33] +(1≠(ALPHε' = ≥ ≤ '))/LP14
[34] +(((C+1)=ρNUM)^(1=ρALPH))/LP1
[35] →LP14+1
[36] LP14:''          SIGN ENTERED IN ERROR,'
[37] PARM←AKI '          DATA ENTERED IN ERROR, REENTER LAST CONSTRAINT:'
[38] →LP2
[39] LP1:MAT+MAT,NUM ◇ OP+OP,ALPH
[40] +(((R×C1)>ρMAT),((R×C1)<ρMAT))/LP,ERR4
[41] CJ+0ρ0
[42] L6:PARM←AKI '
    ENTER PROFIT OR COST FUNCTION COEFFICIENTS: '
[43] SEPARATE PARM
[44] +(C=ρNUM)/L160
[45] 'INVALID ENTRY, PLEASE REENTER' ◇ →L6
[46] L160:CJ+CJ,NUM
[47] +((C>ρCJ),(C<ρCJ))/L6,ERR5
[48] CODE+'123456789012345678901234567890'
[49] SIMPLEX1
[50] 'LINEAR PROGRAMMING COMPLETED'
[51] +0
[52] ERR1:'THE REPLY MUST BE MAX OR MIN. PLEASE REENTER:' ◇ →L111
[53] ERR2:'YOU SHOULD HAVE ENTERED TWO INTEGERS. PLEASE REENTER:' ◇ →L112
[54] ERR4:'YOU HAVE ENTER AN INCORRECT NUMBER OF COEFFICIENTS OR YOU HAVE
    FORGOTTEN TO ENTER THE ARITHMETIC OPERATOR(S). PLEASE REENTER.' ◇ →L114
[55] ERR5:'YOU SHOULD HAVE ENTERED ONE PROFIT OR COST FUNCTION COEFFICIENT
    FOR EACH VARIABLE. PLEASE REENTER.' ◇ →L6
[56] UNBD:''
    THE OBJECTIVE FUNCTION IS NOT BOUNDED ABOVE X = ';COL;''
▽

```

```

V SAMP
[1] '
[2] '*****'
[3] ''
[4] 'GIVEN THE FOLLOWING SAMPLE APPLICATION'
[5] ''
[6] '*****'
[7] '

[8] 'SOLVE THE FOLLOWING LINEAR PROGRAMMING PROBLEM:'
[9] ''
[10] '          MINIMIZE          5X+6Y+7Z=COST
[11] '          SUBJECT TO      X+ Y+ Z=1000'
[12] '                          X≤300'
[13] '                          Y≥150'
[14] '                          Z≥200'
[15] '

[16] 'REFORMAT TO THIS FORM FOR COMPUTER ENTRY:'
[17] ''
[18] '          MINIMIZE          5  6  7 '
[19] ''
[20] '          SUBJECT TO      1  1  1 = 1000'
[21] '                          1  0  0 ≤ 300'
[22] '                          0  1  0 ≥ 150'
[23] '                          0  0  0 ≥ 200'
[24] '

[25] 'THE DATA WILL BE ENTERED IN THE FOLLOWING FORMAT'
[26] ''
[27] ' PROMPT: DO YOU KNOW HOW TO USE THIS PROGRAM?'
[28] ' REPLY:  ENTER YES IF YOU WISH A SAMPLE PROBLEM PLUS AN EXPANATION OF TH
E ENTRY OF THAT PROBLEM.'
[29] '        ENTER NO IF YOU ALREADY KNOW HOW TO USE THE PROGRAM. YOU MAY U
SE Y OR N.'
[30] ' EX.:   YES
[31] ' PROMPT: ENTER A PROJECT TITLE IF YOU WISH.'
[32] ' REPLY:  ENTER ANY TITLE THAT YOU WISH TO HAVE PRINTED ON THE RESULT PAGE
[33] ' EX.:   OP512 - OPERATIONS RESEARCH - JOHN JONES
[34] ' PROMPT: IS THIS MAXIMIZATION OR A MINIMIZATION PROBLEM? REPLY MAX OR MI
N.'
[35] ' REPLY:  TYPE MAX IF YOU WISH TO MAXIMIZE THE DATA YOU ENTER, OR MIN IF
YOU WISH TO MINIMIZE.'
[36] ' EX.:   MIN
[37] ' PROMPT: ENTER THE NUMBER OF CONSTRAINTS, AND THE NUMBER OF VARIABLES.'

```

[38] ' REPLY: YOU MUST ENTER THE NUMBER OF CONSTRAINTS THAT THE PROBLEM CONTAINS. YOU WILL ALSO NEED TO'

[39] ' ENTER THE NUMBER OF VARIABLES (SUCH AS X Y Z). YOU MUST ENTER TWO INTEGERS WITH A SPACE BETWEEN'

[40] ' EX.: 4 3
'

[41] ' PROMPT: ENTER COEFFICIENTS OF SIMPLEX PROBLEM, ONE ROW AT A TIME:'

[42] ' OR: ENTER NEXT CONSTRAINT:'

[43] ' REPLY: ENTER ONE ROW OF THE CONSTRAINTS AT A TIME, MAKING SURE THAT YOU ENTER THE VALUES AND THE'

[44] ' ARITHMETIC OPERATOR IN THE REFORMAT EXAMPLE GIVEN ABOVE. ONLY (= ≥ ≤) ARE VALID OPERATORS.'

[45] ' EX.: 1 1 1 = 1000 (FIRST CONSTRAINT ENTRY)
'

[46] ' PROMPT: ENTER PROFIT OR COST FUNCTION COEFFICIENTS.'

[47] ' REPLY: ENTER THE PROFIT OR COST VALUES OF THE OBJECTIVE FUNCTION.'

[48] ' EX.: 5 6 7
'

[49] '
'

[50] 'PLEASE RETAIN THIS PRINTOUT FOR FUTURE REFERENCE.'

[51] '
'

∇

```

▽ SIMPLEX1
[1] *THIS FUNCTION REQUIRES THE FUNCTION SOLI.
[2] *
    TURN CARRIAGE TO POSITION PAPER AT START OF NEW PAGE, THEN PRESS RETURN.'
[3] PGS+□
[4] M1+1+(REPLY='A')
[5] *(M1=1)/L7
[6] *
    LINEAR PROGRAMMING WITH MIXED CONSTRAINTS - MAXIMIZATION
    *
[7] OPTI+'OPTIMAL SOLUTION FOR MAXIMIZATION'
[8] *L7+1
[9] L7:'
    LINEAR PROGRAMMING WITH MIXED CONSTRAINTS - MINIMIZATION
    *
[10] OPTI+'OPTIMAL SOLUTION FOR MINIMIZATION'
[11] TITLE;'
    *
[12] MHIGH+-1*PHIGH+100*[/MAT
[13] MAT+MAT,0 ◇ RC1+R,C1 ◇ CNTR+0 ◇ MAT+RC1ρMAT
[14] ZEROS+R,(+/OPε'≥')+ρOP ◇ MAT+MAT,ZEROSρ0
[15] RC1+ρMAT ◇ C1+1+RC1 ◇ C+C+1
[16] MAT[;C1]+MAT[;C] ◇ MAT[;C]+Rρ0
[17] PC+C ◇ PR+1
[18] VIB+Rρ0 ◇ ZJ+C1ρ0 ◇ I+1 ◇ CJ+CJ,(C1-C)ρ0
[19] L13:*(OP[I]≠'≥')/L8 ◇ MAT[PR;PC]+-1 ◇ MAT[PR;PC+1]+1
[20] VIB[PR]+PC+1 ◇ *(2=M1)/L9
[21] CJ[PC+1]+PHIGH ◇ +L9+1
[22] L9:CJ[PC+1]+MHIGH
[23] PC+PC+2 ◇ +L10
[24] L8:MAT[PR;PC]+1 ◇ VIB[PR]+PC
[25] *(OP[I]≠'=')/L11+1 ◇ *(M1=1)/L11
[26] CJ[PC]+MHIGH ◇ +L11+1
[27] L11:CJ[PC]+PHIGH
[28] PC+PC+1
[29] L10:PR+PR+1 ◇ *(R≥I+1)/L13 ◇ C+C1-1
[30] VPT+VIB[;R] ◇ CJC+CJ[VPT] ◇ PRINT
[31] L12:*(M1=2)/L16
[32] *((√/(CZ<0))≠0)/L117
[33] SOLI
[34] +0
[35] L117:COL+,CZi([/CZ) ◇ +L17
[36] L16:*((√/(CZ>0))≠0)/L118
[37] SOLI
[38] +0
[39] L118:COL+,CZi([/CZ)
[40] L17:*((+/(,MAT[;COL]>0))=0)/UNBD
[41] ANDED+(ANDED>0)/;ρANDED+,ANDED+(MAT[;COL]>0)^(MAT[;C1]>0)
[42] REMAIN+,MAT[ANDED;C1]≠MAT[ANDED;COL]
[43] ROW+,REMAINi([/REMAIN) ◇ ROW+ANDED[ROW] ◇ VIB[ROW]+COL ◇ CJC[ROW]+CJ[COL]
[44] MAT[ROW;]+MAT[ROW;]≠MAT[ROW;COL] ◇ I+1
[45] L18:PIV+MAT[I;COL] ◇ *(I=ROW)/L19
[46] MAT[I;]+MAT[I;]-,(PIV×MAT[ROW;])
[47] L19:*(R≥I+1)/L18
[48] PRINT
[49] +L12
[50] UNBD:'
    THE OBJECTIVE FUNCTION IS NOT BOUNDED ABOVE X = ';COL;'
    *

```

▽

```

▽ PRINT
[1] *THIS FUNCTION USES THE FOLLOWING SCIENTIFIC
[2] *TIME SHARING FUNCTIONS: CENTER, ΔFMT, AND COLNAMES
[3] ZJ+CJC+. *MAT
[4] CZ+CJ-ZJ[;C1-1]
[5] '
'
[6] CNTR+CNTR+1
[7] FORM+'TABLEAU ',CODE[CNTR]
[8] COLNM+'/X1 /X2 /X3 /X4 /X5 /X6 /X7 /X8 /X9 /X10/X11/X12/X13/X14/X15/X16/X
17/X18'
[9] COLNM+' /VAR/Z ',((C1-1)*4)+COLNM
[10] FSTR+'I4,X1,F14.2,10F10.2'
[11] FSTR CENTER FORM
[12] '
'
[13] FSTR COLNAMES COLNM
[14] FSTR ΔFMT(VIB;MAT[;C1];MAT[;R;C1-1])
[15] CC+1,C1-1
[16] CC+CCp1(C1-1)
[17] FSTR ΔFMT(0;ZJ[C1];ZJ[CC])
▽

```


THE INTERACTIVE EFFECTS OF THE MARGINAL
PROPENSITY TO CONSUME AND THE INVESTMENT
ACCELERATOR ON THE PRODUCTION OF INCOME

John Park
Chairman, Department of Economics
and Business Administration
Frostburg State College
Frostburg, Maryland

DESCRIPTION

This is a short, simple problem designed for use by undergraduate economic and business forecasting students to get them acquainted with an APL application in economics. The program attempts to generate a trend of national income (Gross National Product or "GNP"), based upon a set of assumptions. These assumptions are described in the program. Different combinations of assumptions would, of course, produce different income trends.

The object of the program is to attack a problem which is too tedious for hand calculation and solution. The project may be assigned to students as a homework assignment or for drill and practice. Outputs are printed out in a tabular form and, as an option, the trend of income (the last column in the table) can be plotted out on a chart.

To use the program, the student types *ACTION* and, when asked, enters the following values.

1. The autonomous investment (constant throughout the exercise)
2. The marginal propensity to consume (MPC)
3. The investment accelerator (alpha)
4. The number of periods the program is to run.

As an additional exercise, the instructor may wish to designate to students a particular shape (perhaps fluctuating) of income trends of the past and/or present as a starting point, and then ask them to suggest a particular combination of numerical figures for marginal propensity to consume and alpha which would bring the given initial trend of income to some planned or desired level.

PROGRAM USAGE

ACTION

DO YOU NEED INSTRUCTIONS ?

YES

HI ECONOMIST! I SEE THAT YOU ARE TRYING TO FORECAST THE NATIONAL INCOME TREND. WITH THE HELP OF APL, AND YOU MAY SUCCEED IN DOING SO IN A JIFFY. LET US ASSUME THE FOLLOWING CONDITIONS:

1. AUTONOMOUS INVESTMENT (A) IS CONSTANT THROUGHOUT THE GIVEN PERIODS WE OBSERVE.
2. INDUCED CONSUMPTION (C) OF A PERIOD IS THE NATIONAL INCOME OF THE PREVIOUS PERIOD ADJUSTED BY YOUR MARGINAL PROPENSITY TO CONSUME (MPC).
3. INDUCED INVESTMENT (I) OF A PERIOD IS THE ABSOLUTE VALUE OF THE DIFFERENCE BETWEEN THE C OF THE PERIOD AND THE C OF THE IMMEDIATELY PRECEDING PERIOD, ADJUSTED BY YOUR ALPHA (I.E., MULTIPLIED BY THE GIVEN ALPHA).
4. TOTAL NATIONAL INCOME (Y) OF A PERIOD IS THE SUMMATION OF A, C, AND I FOR THAT PERIOD.
5. THE TOTAL NUMBER OF MULTIPLIER PERIODS (T'S) WE OBSERVE IS, SAY 20 (I.E., 20 T'S).

ENTER AUTONOMOUS INVESTMENT (A)

□:

1000

ENTER MARGINAL PROPENSITY TO CONSUME (MPC)

□:

.75

ENTER ALPHA (ACCELERATOR)

□:

2.7

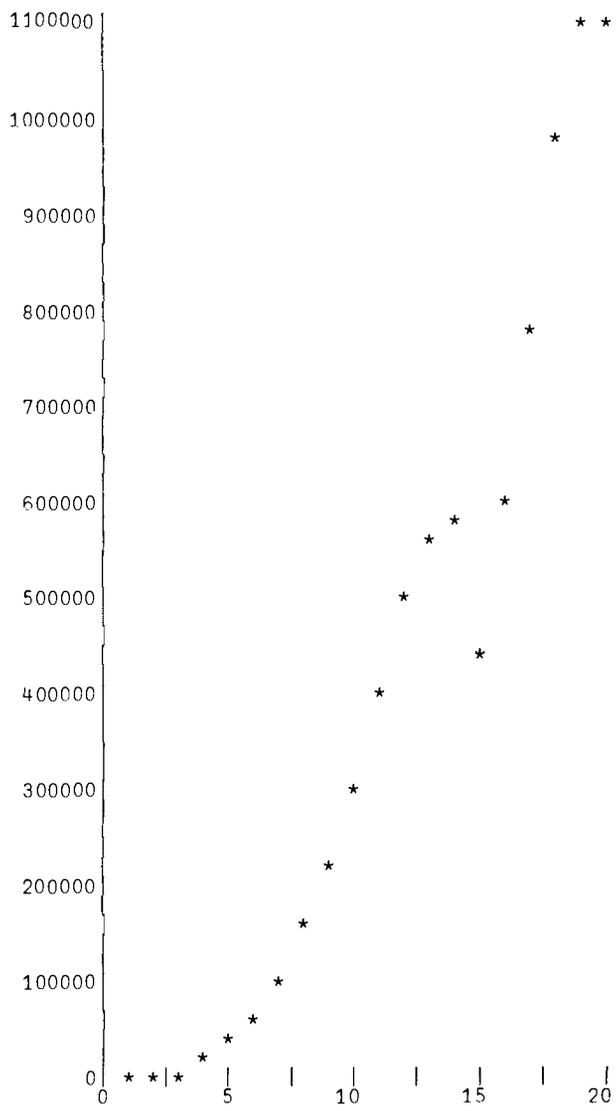
ENTER NUMBER OF PERIODS

□:

20

<i>MULTIPLIER PERIOD</i>	<i>AUTONOMOUS INVESTMENT</i>	<i>INDUCED CONSUMPTION</i>	<i>INDUCED INVESTMENT</i>	<i>TOTAL NATIONAL INCOME</i>
(T)	(A)	(C)	(I)	(Y)
1	1000	0	0	1,000
2	1000	750	2,025	3,775
3	1000	2,831	5,619	9,451
4	1000	7,088	11,493	19,581
5	1000	14,686	20,514	36,200
6	1000	27,150	33,653	61,803
7	1000	46,353	51,847	99,199
8	1000	74,400	75,727	151,126
9	1000	113,345	105,152	219,497
10	1000	164,623	138,450	304,073
11	1000	228,055	171,267	400,321
12	1000	300,241	194,903	496,144
13	1000	372,108	194,041	567,149
14	1000	425,361	143,785	570,146
15	1000	427,609	6,070	434,679
16	1000	326,009	274,320	601,330
17	1000	450,997	337,467	789,464
18	1000	592,098	380,973	974,071
19	1000	730,553	373,828	1,105,381
20	1000	829,036	265,904	1,095,940

WOULD YOU LIKE A PLOT OF THE ABOVE DATA?
YES



PROGRAM DESCRIPTION

▽ DESCRIBEACTION

[1] ' HI ECONOMIST! I SEE THAT YOU ARE TRYING TO FORECAST THE NATIONAL'
[2] ' INCOME TREND. WITH THE HELP OF APL, AND YOU MAY SUCCEED IN DOING SO'
[3] ' IN A JIFFY. LET US ASSUME THE FOLLOWING CONDITIONS:'
[4] ''
[5] ' 1. AUTONOMOUS INVESTMENT (A) IS CONSTANT THOUGHOUT THE GIVEN PERIODS'
[6] ' WE OBSERVE.'
[7] ''
[8] ' 2. INDUCED CONSUMPTION (C) OF A PERIOD IS THE NATIONAL INCOME OF THE'
[9] ' PREVIOUS PERIOD ADJUSTED BY YOUR MARGINAL PROPENSITY TO CONSUME (MP
[10] ' C).'
[11] ''
[12] ' 3. INDUCED INVESTMENT (I) OF A PERIOD IS THE ABSOLUTE VALUE OF THE'
[13] ' DIFFERENCE BETWEEN THE C OF THE PERIOD AND THE C OF THE IMMEDIATELY'
[14] ' PRECEDING PERIOD, ADJUSTED BY YOUR ALPHA (I.E., MULTIPLIED BY THE'
[15] ' GIVEN ALPHA). '
[16] ''
[17] ' 4. TOTAL NATIONAL INCOME (Y) OF A PERIOD IS THE SUMMATION OF A, C, AND
[18] ' I'
[19] ' FOR THAT PERIOD.'
[20] ''
[21] ' 5. THE TOTAL NUMBER OF MULTIPLIER PERIODS (T''S) WE OBSERVE IS, SAY 20
[22] ' (I.E.,'
[23] ' 20 T''S).'
[24] ''

▽

```

V ACTION;A;ALPHA;C;CTR;FMT;I;MPC;N;T;Y
[1] THIS FUNCTION REQUIRES THE FUNCTION DESCRIBEACTION, WHICH DESCRIBES
[2] THE ASSUMPTIONS OF THE MODEL, AND THE PUBLIC FUNCTIONS COLNAMES,
[3] ΔFMT AND PLOT
[4] 'DO YOU NEED INSTRUCTIONS ?'
[5] →L1×7\ 'N8'=1+
[6] DESCRIBEACTION
[7] L1:'ENTER AUTONOMOUS INVESTMENT (A) '
[8] Y+,A+
[9] I+C+0
[10] 'ENTER MARGINAL PROPENSITY TO CONSUME (MPC) '
[11] MPC+
[12] 'ENTER ALPHA (ACCELERATOR) '
[13] ALPHA+
[14] 'ENTER NUMBER OF PERIODS'
[15] N+
[16] CTR+2
[17] LOOP:C+C,MPC×Y[CTR-1]
[18] I+I,ALPHA×C[CTR]-C[CTR-1]
[19] Y+Y,A+C[CTR]+I[CTR]
[20] CTR+CTR+1
[21] →LOOP×\ (CTR≤N)
[22] A+NρA
[23] T+1N
[24] ''
[25] ''
[26] FMT+'2I12,3CI17'
[27] FMT COLNAMES '/MULTIPLIER/AUTONOMOUS/INDUCED /INDUCED /TOTAL NATIONAL
L'
[28] FMT COLNAMES '/PERIOD /INVESTMENT/CONSUMPTION/INVESTMENT/INCOME
'
[29] ''
[30] FMT COLNAMES '/(T)/(A)/(C)/(I)/(Y) '
[31] ''
[32] FMT ΔFMT(T;A;C;I;Y)
[33] ''
[34] ''
[35] 'WOULD YOU LIKE A PLOT OF THE ABOVE DATA?'
[36] →0×\ 'N'=1+
[37] 60 80 PLOT Y
[38] ''
[39] ''

```

∇

NETWORK ANALYSIS

Donald G. Sanford
Chairman
Department of Administration
and Systems
Florida Atlantic University
Boca Raton, Florida

DESCRIPTION

This program is used to analyze Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT) networks. Such networks are useful in planning, scheduling, and control of complex projects (such as research, development, and construction).

The program can be used to solve problems which would be too time consuming for hand calculation or to verify the results of manual calculations.

It was developed to be used in a junior year course in quantitative methods in business or by beginning graduate students in a prerequisite course in quantitative methods for an MBA program.

Instructions for Use

Input instructions are contained within the program and are printed out to the user as needed.

It is assumed that the user has a basic understanding of network theory as presented in a basic text such as Quantitative Approaches to Management by Levin and Kirkpatrick.

The user desiring a description of the workspace should execute the niladic function *NETINFO*. The user then has the option of analyzing a network after having read the workspace description. Those not desiring a description should execute the niladic function *NET1* directly in order to initiate a network

analysis. The input variables are:

PERT Networks

Beginning Event

Ending Event

Optimistic Time

Most Likely Time

Latest Allowable Date

Completion Dates for
Probability Analysis

CPM Networks

Beginning Event

Ending Event

Expected Time

Latest Allowable Date

Outputs from the program include:

- (1) List of input data
- (2) Number of activities input
- (3) Events listed in numerical order
- (4) Starting and finish events listed
- (5) Critical path
- (6) Expected completion date
- (7) Slack
- (8) Z values for probability analysis (optional) --
(PERT only)
- (9) Complete event analysis (optional)
For each event:
Event numbers
Expected completion date
Latest date
Slack
Critical path indicator

(10) Complete activity analysis (optional)

For each activity:

To-From events

Expected time

Expected start date

Expected completion date

Latest start date

Latest completion date

Standard deviation (PERT only)

Critical path indicator

Sample Use

In the next section the following uses are demonstrated:

- (1) PERT Network
- (2) CPM Network
- (3) Error checking and recovery

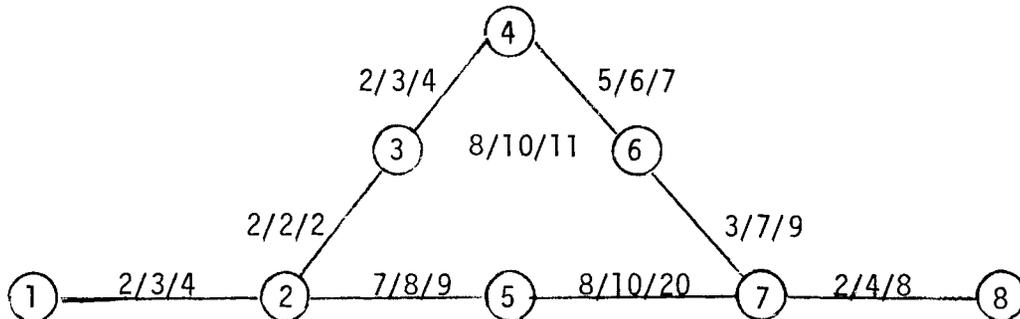


Figure 1

The network shown in Figure 1 is used for these demonstrations. These data were used to illustrate that the critical path can change as the result of consideration of pessimistic or optimistic times, as is true with PERT analysis, while CPM considers only one time estimate.

PROGRAM USAGE

NETINFO

WORKSPACE NETWORK

THIS WORKSPACE CONTAINS THREE FUNCTIONS WHICH OPERATE TOGETHER TO SOLVE PROBLEMS IN NETWORK ANALYSIS. IT IS ASSUMED THAT THE USER IS FAMILIAR WITH NETWORK THEORY.

THE USER MAY SPECIFY CPM OR PERT ANALYSIS. PROVISION IS MADE FOR THE USER TO SELECT ABBREVIATED OR COMPLETE OUTPUT RESULTS.

FEATURES INCLUDE:

- (1) SELF-CONTAINED INPUT INSTRUCTIONS
- (2) EXTENSIVE ERROR CHECKING AND RECOVERY
- (3) CAPABILITY OF CORRECTING DATA IN NETWORK AFTER IT HAS BEEN INPUT
- (4) OPTIONAL COMPUTATION OF 'Z' VALUES FOR PROBABILITY ANALYSIS WITH PERT NETWORKS

THE FUNCTIONS IN THIS WORKSPACE ARE DESCRIBED BELOW:

- NETINFO - CONTAINS A DESCRIPTION OF THE FUNCTIONS. THE USER HAS THE OPTION OF EXECUTING THE ANALYSIS OF A PERT OR CPM NETWORK FROM THIS FUNCTION BY CALLING FUNCTION NET1.
- NET1 - HANDLES INPUT OF NETWORK DATA TO BE ANALYZED, ERROR CHECKING, RECOVERY, AND NETWORK ANALYSIS. CALLS FUNCTION NET2 FOR OUTPUT.
- NET2 - PRODUCES OUTPUT OF THE PERT OR CPM NETWORK ANALYSIS. THE USER MAY SPECIFY SHORT OR LONG OUTPUT FORMAT.

DO YOU WANT TO INPUT A NETWORK TO BE ANALYZED? ANSWER YES OR NO.
YES
ENTER CPM OR PERT TO SPECIFY TYPE OF NETWORK TO BE ANALYZED
PERT

NET1
ENTER CPM OR PERT TO SPECIFY TYPE OF NETWORK TO BE ANALYZED
XYZ
ENTER CPM OR PERT TO SPECIFY TYPE OF NETWORK TO BE ANALYZED
PERT
ENTER PERT ACTIVITY DATA IN GROUPS OF FIVE: BEGINNING EVENT, ENDING
EVENT, OPTIMISTIC TIME, MOST LIKELY TIME, PESSIMISTIC TIME.
BEGINNING EVENT MUST BE A HIGHER NUMBER THAN ENDING EVENT FOR EACH ACTIVITY.
ENTER -1 TO INDICATE THAT YOU HAVE NO MORE DATA.

□:
1 2 6 7 8
□:
2 3 8 9
INPUT MUST BE 5 ITEMS.
□:
2 3 8 9 10
□:
3 4 4 5 6
□:
3 5 7 8 9
□:
-1

INPUT DATA

1	2	6	7	8
2	3	8	9	10
3	4	4	5	6
3	5	7	8	9

DO YOU WISH TO CHANGE ANY DATA? ANSWER YES OR NO.
YES

ENTER ROW NUMBER, COLUMN NUMBER, AND NEW VALUE.

□:
1
INPUT MUST BE 3 ITEMS.

ENTER ROW NUMBER, COLUMN NUMBER, AND NEW VALUE.

□:
10 10 8

ENTER ROW NUMBER, COLUMN NUMBER, AND NEW VALUE.

□:
1 5 10

REVISED DATA:

1	2	6	7	10
2	3	8	9	10
3	4	4	5	6
3	5	7	8	9

DO YOU WISH TO CHANGE ANY DATA? ANSWER YES OR NO.

NO

PROGRAM DESCRIPTION

```

V NETINFO
[1]  * THIS FUNCTION PROVIDES THE USER WITH A DESCRIPTION OF THE CONTENTS OF
[2]  * THE WORKSPACE. THE USER MAY CALL THE FUNCTION WHICH PERFORMS THE
[3]  * NETWORK ANALYSIS AFTER READING THE DESCRIPTION OF THE WORKSPACE BY
[4]  * GIVING AN AFFIRMATIVE RESPONSE TO THE QUESTION, 'DO YOU WANT TO INPUT
[5]  * A NETWORK TO BE ANALYZED?'
[6]  * '
[7]  * '
[8]  * '
[9]  * '
[10] * '                                WORKSPACE NETWORK'
[11] * '
[12] * '
[13] * THIS WORKSPACE CONTAINS THREE FUNCTIONS WHICH OPERATE'
[14] * TOGETHER TO SOLVE PROBLEMS IN NETWORK ANALYSIS. IT IS '
[15] * ASSUMED THAT THE USER IS FAMILIAR WITH NETWORK THEORY.'
[16] * '
[17] * THE USER MAY SPECIFY CPM OR PERT ANALYSIS. PROVISION'
[18] * IS MADE FOR THE USER TO SELECT ABBREVIATED OR COMPLETE'
[19] * OUTPUT RESULTS.'
[20] * '
[21] * FEATURES INCLUDE:'
[22] * (1) SELF-CONTAINED INPUT INSTRUCTIONS'
[23] * (2) EXTENSIVE ERROR CHECKING AND RECOVERY'
[24] * (3) CAPABILITY OF CORRECTING DATA IN '
[25] * NETWORK AFTER IT HAS BEEN INPUT'
[26] * (4) OPTIONAL COMPUTATION OF 'Z' VALUES'
[27] * FOR PROBABILITY ANALYSIS WITH PERT'
[28] * NETWORKS'
[29] * '
[30] * '
[31] * THE FUNCTIONS IN THIS WORKSPACE ARE DESCRIBED BELOW:'
[32] * '
[33] * NETINFO - CONTAINS A DESCRIPTION OF THE FUNCTIONS.'
[34] * THE USER HAS THE OPTION OF EXECUTING THE'
[35] * ANALYSIS OF A PERT OR CPM NETWORK FROM'
[36] * THIS FUNCTION BY CALLING FUNCTION NET1.'
[37] * '
[38] * NET1 - HANDLES INPUT OF NETWORK DATA TO BE ANALYZED,'
[39] * ERROR CHECKING, RECOVERY, AND NETWORK ANALYSIS.'
[40] * CALLS FUNCTION NET2 FOR OUTPUT.'
[41] * '
[42] * NET2 - PRODUCES OUTPUT OF THE PERT OR CPM NETWORK'
[43] * ANALYSIS. THE USER MAY SPECIFY SHORT OR'
[44] * LONG OUTPUT FORMAT.'
[45] * '
[46] * '
[47] * '
[48] * QUES5:'DO YOU WANT TO INPUT A NETWORK TO BE ANALYZED? ANSWER YES OR NO.'
[49] * +(GOTO,0,QUES5)['YN',1↑, ]
[50] * CALL FUNCTION NET1 IF THE USER WISHES TO PERFORM A NETWORK ANALYSIS
[51] * GOTO:NET1
V
```

```

∇ NET1;A;ACT;B;B1;B2;B3;C;CPAC;CPEV;CRIT;D;DATA1;D1;D2;E;FLAGS;F1;F2;G;H;J;
K;LATEST;M;N;NA;NE;STDEV;Z
[1]  R THIS FUNCTION RECEIVES CPM OR PERT NETWORK DATA FROM THE USER AND
[2]  R ANALYZES THE NETWORK TO DETERMINE THE EXPECTED TIMES, EXPECTED START
[3]  R AND COMPLETION DATES, LATEST START AND COMPLETION DATES, AND THE

[4]  R CRITICAL PATH. IN ADDITION, ACTIVITY STANDARD DEVIATIONS, NETWORK
[5]  R STANDARD DEVIATION, AND Z VALUES FOR SELECTED COMPLETION DATES ARE
[6]  R COMPUTED FOR PERT NETWORKS ONLY. IT IS ASSUMED THAT THE USER IS
[7]  R FAMILIAR WITH NETWORK THEORY. THE FUNCTION SHOULD BE USED TO ANALYZE
[8]  R PROBLEMS WHICH ARE TOO COMPLEX FOR MANUAL CALCULATION OR TO CHECK THE
[9]  R RESULTS OF MANUAL CALCULATIONS FOR CLASSROOM USE.
[10] R INPUT INSTRUCTIONS ARE CONTAINED WITHIN THE PROGRAM.
[11] START:'ENTER CPM OR PERT TO SPECIFY TYPE OF NETWORK TO BE ANALYZED'
[12] →(CPM,PERT,START)['CP'11↑,□]
[13] CPM:Z+3
[14] R INPUT DATA INSTRUCTIONS FOR CPM NETWORKS
[15] 'ENTER CPM ACTIVITY DATA IN GROUPS OF THREE:'
[16] 'BEGINNING EVENT, ENDING EVENT, EXPECTED TIME.'
[17] →GO1
[18] PERT:Z+5
[19] R INPUT DATA INSTRUCTIONS FOR PERT NETWORKS
[20] 'ENTER PERT ACTIVITY DATA IN GROUPS OF FIVE: BEGINNING EVENT, ENDING'
[21] 'EVENT, OPTIMISTIC TIME, MOST LIKELY TIME, PESSIMISTIC TIME.'
[22] GO1:'BEGINNING EVENT MUST BE A HIGHER NUMBER THAN ENDING EVENT FOR EACH AC
TIVITY.'
[23] 'ENTER -1 TO INDICATE THAT YOU HAVE NO MORE DATA.'
[24] NA←0
[25] ACT←10
[26] R NETWORK DATA READ IN HERE
[27] READIN:A+□
[28] →(NEXT,NOMOREDATA)[1+(1+A)<1]
[29] R CHECK THAT THE APPROPRIATE NUMBER OF DATA ITEMS IS ENTERED (3 FOR CPM,
5 FOR PERT)
[30] NEXT:→(OK1,ERR1)[1+(ρA)≠Z]
[31] R ERROR MESSAGE FOR INCORRECT NUMBER OF DATA ITEMS
[32] ERR1:'INPUT MUST BE ';Z;' ITEMS.'
[33] →READIN
[34] OK1:→(OK2,ERR2)[1+(A[1]≥A[2])]
[35] ERR2:'EVENTS MUST BE IN ASCENDING ORDER.'
[36] →READIN
[37] OK2:NA+NA+1
[38] ACT←ACT,A
[39] →READIN
[40] NOMOREDATA:ACT←(NA,Z)ρACT
[41] ' '
[42] ' '
[43] R INPUT DATA IS DISPLAYED TO USER FOR CORRECTNESS
[44] 'INPUT DATA'
[45] ' '
[46] ACT
[47] AGAIN:' '
[48] 'DO YOU WISH TO CHANGE ANY DATA? ANSWER YES OR NO.'
[49] →(YES,NO,AGAIN)['YN'11↑,□]

```

```

[50] YES: ' '
[51] 'ENTER ROW NUMBER, COLUMN NUMBER, AND NEW VALUE.'
[52] R DATA CORRECTIONS ENTERED HERE
[53] A+□
[54] R CHECK THAT THREE ITEMS HAVE BEEN ENTERED FOR DATA CORRECTION
[55] R (ROW NO., COL. NO., NEW VALUE)
[56] +(GOOD1,BAD1)[1+(pA)≠3]
[57] BAD1:'INPUT MUST BE 3 ITEMS.'
[58] +YES
[59] GOOD1:+((A[1]>1+pACT)v(A[2]>-1+pACT))/YES
[60] R DATA CORRECTIONS MADE HERE
[61] ACT[A[1];A[2]]+A[3]
[62] ' '
[63] 'REVISED DATA:'
[64] ' '
[65] ACT
[66] R BRANCH BACK TO SEE IF THERE ARE MORE CORRECTIONS TO BE ENTERED
[67] +AGAIN
[68] R NO (MORE) CORRECTIONS
[69] NO:NA+1+pACT
[70] ' '
[71] 'ENTER LATEST ALLOWABLE DATE FOR COMPLETION OF PROJECT.'
[72] LATEST+□
[73] ' '
[74] R NEXT TEN LINES SUMMARIZE THE INPUT DATA
[75] NA;' ACTIVITIES HAVE BEEN ENTERED.'
[76] B1+ACT[;1],ACT[;2]
[77] B1+B1[ΔB1]
[78] B2+1+((B1,0)-(0,B1))
[79] B2+(0≠B2)/B1
[80] NE+pB2
[81] 'INVOLVING ';NE;' DIFFERENT EVENTS, AS FOLLOWS:'
[82] ' '
[83] B2
[84] R CHECK FOR MULTIPLE FINISH EVENTS
[85] D1+(-1+B2)εACT[;1]
[86] +((^/D1)=1)/OK3
[87] 'MULTIPLE FINISH EVENTS IN THE NETWORK, USE DUMMY ACTIVITY.'
[88] R IF MULTIPLE FINISH EVENTS, EXIT PROGRAM
[89] +0
[90] OK3:D1+(1+B2)εACT[;2]
[91] R CHECK FOR MULTIPLE START EVENTS
[92] +((^/D1)=1)/OK4
[93] 'MULTIPLE STARTING EVENTS IN THE NETWORK, USE DUMMY ACTIVITY.'
[94] R IF MULTIPLE START EVENTS, EXIT PROGRAM
[95] +0
[96] OK4:' '
[97] 'THE STARTING EVENT FOR THE NETWORK IS '1+B2;' AND THE FINISH EVENT IS '
; -1+B2
[98] R SET UP ARRAYS FOR NETWORK ANALYSIS. E FOR EVENTS AND C FOR ACTIVITIES.
[99] E+(NE,5)p0
[100] C+(NA,7)p0
[101] +(Z=3)/OK5
[102] R CALCULATE EXPECTED TIMES AND SD'S FOR PERT NETWORKS

```

```

[103] C[;1]+(ACT[;3]+(4*ACT[;4])+ACT[;5])÷6
[104] C[;6]+(ACT[;5]-ACT[;3])÷6
[105] +G02
[106] OK5:C[;1]+ACT[;3]
[107] PUT EVENT NUMBERS IN MATRIX E
[108] G02:E[;1]+B2
[109] E[1;2]+0
[110] E[1;5]+1
[111] N IS ACTIVITY INDEX
[112] G02:N+1
[113] REFERENCE NUMBER IN E FOR BEGINNING EVENT OF ACTIVITY
[114] G03:F1+B2\ACT[N;1]
[115] +(~(E[F1;5]=1)^(C[N;7]=0))/G04
[116] C[N;2]+E[F1;2]
[117] C[N;3]+C[N;2]+C[N;1]
[118] C[N;7]+1
[119] G04:+(N=NA)/G05
[120] N+N+1
[121] +G03
[122] M IS EVENT INDEX
[123] G05:M+1
[124] G06:+(E[M;5]=1)/G07
[125] F2 IS LIST OF ACTIVITIES WHOSE ENDING EVENT IS EVENT BEING ANALYZED
[126] F2+,(ACT[;2]=E[M;1])/NA
[127] IF FLAGS FOR ALL SUCH ACTIVITIES ARE SET, UPDATE EVENT.
[128] FLAGS+,C[F2;7]
[129] +(~(^(FLAGS)))/G07
[130] EXPECTED DATE FOR EVENT IS LATEST COMPLETION DATE OF ALL ACTIVITIES
[131] LEADING TO THE EVENT
[132] E[M;2]+[C[F2;3]
[133] E[M;5]+1
[134] G07:+(M=NE)/G08
[135] M+M+1
[136] +G06
[137] IF ALL EVENT FLAGS ARE SET, ALL EXPECTED DATES ARE RESOLVED AND
[138] FORWARD PART OF ANALYSIS IS COMPLETE
[139] G08:+(~(^(E[;5])))/G002
[140] FORWARD ANALYSIS COMPLETED, REVERSE NEXT
[141] ZERO OUT FLAGS FOR RE-USE
[142] E[;5]+0
[143] C[;7]+0
[144] E[NE;3]+LATEST
[145] E[NE;5]+1
[146] N IS ACTIVITY INDEX
[147] G020:N+1
[148] F1 IS REF. NO. IN E FOR ENDING EVENT OF ACTIVITY
[149] G021:F1+B2\ACT[N;2]
[150] +(~(E[F1;5]=1)^(C[N;7]=0))/G022
[151] C[N;5]+E[F1;3]
[152] C[N;4]+C[N;5]-C[N;1]
[153] C[N;7]+1
[154] G022:+(N=NA)/G023
[155] N+N+1
[156] +G021
[157] M IS EVENT INDEX
[158] G023:M+1

```

```

[159]G024:→(E[M;5]=1)/G025
[160]F2 IS A LIST OF ACTIVITIES WHOSE BEGINNING EVENT IS EVENT BEING ANALYZED
[161]F2←,(ACT[;1]=E[M;1])/NA
[162]IF FLAGS FOR ALL SUCH ACTIVITIES ARE SET, UPDATE EVENT
[163]FLAGS←,C[F2;7]
[164]→(~/FLAGS)/G025
[165]LATEST DATE FOR EVENT IS EARLIEST START DATE OF ALL ACTIVITIES COMING FROM THE EVENT
[166]E[M;3]←L/C[F2;4]
[167]E[M;5]←1
[168]G025:→(M=NE)/G026
[169]M←M+1
[170]→G024
[171]IF ALL EVENT FLAGS ARE SET, ALL LATEST DATES ARE RESOLVED AND NETWORK ANALYSIS IS NEARLY COMPLETE
[172]G026:→(~/E[;5])/G020
[173]COMPUTE SLACK
[174]E[;4]←E[;3]-E[;2]
[175]DETERMINE CRITICAL PATH
[176]CRIT←(E[;4]=(L/E[;4]))/E[;1]
[177]CALL NET2 FOR OUTPUT REPORT
[178]NET2

```

▽

▽ NET2

```

[1] THIS FUNCTION PRODUCES OUTPUT OF CPM OR PERT NETWORK ANALYSIS. IT
[2] ACCEPTS VALUES FROM FUNCTION NET1 WHICH CALLS THIS FUNCTION AS ITS
[3] LAST STATEMENT. THIS FUNCTION SHOULD NOT BE USED ALONE. NETWORK
[4] ANALYSIS SHOULD BE INITIATED BY EXECUTING NETINFO OR NET1. THE FORMER
[5] PROVIDES A DESCRIPTION OF THE WORKSPACE PRIOR TO GIVING THE USER THE
[6] OPTION OF PERFORMING NETWORK ANALYSIS. THIS FUNCTION REFERENCES
[7] ΔFMT WHICH MAY NOT BE AVAILABLE ON SOME SYSTEMS.
[8]
[9] RESULTS OF THE NETWORK ANALYSIS ARE CONTAINED IN VECTOR CRIT WHICH IS
[10] THE CRITICAL PATH EVENTS AND IN ARRAYS E AND C. ARRAY E CONTAINS
[11] EVENT INFORMATION. ONE ROW IS ASSIGNED TO EACH EVENT.
[12] COLUMNS IN ARRAY E ARE DEFINED AS FOLLOWS:
[13] (1) EVENT NUMBER (2) EXPECTED DATE (3) LATEST DATE
[14] (4) SLACK (5) STATUS FLAG
[15] ARRAY C CONTAINS ACTIVITY INFORMATION. ONE ROW IS ASSIGNED TO EACH
[16] ACTIVITY. COLUMNS IN ARRAY C ARE DEFINED AS FOLLOWS:
[17] (1) EXPECTED TIME (2) EXPECTED START DATE (3) EXPECTED
[18] COMPLETION DATE (4) LATEST START DATE (5) LATEST COMPLETION DATE
[19] (6) STANDARD DEVIATION (PERT ONLY) (7) STATUS FLAG
[20]
[21] VECTORS 1= ON CRITICAL PATH 0= NOT ON CRITICAL PATH
[22] CPAC IS ACTIVITY VECTOR CPEV IS EVENT VECTOR
[23] CPAC←(ACT[;1]∈CRIT)^(ACT[;2]∈CRIT)
[24] CPEV←E[;1]∈CRIT
[25] ' '
[26] 'CRITICAL PATH:'
[27] ' '
[28] CRIT←(1,ρCRIT)ρCRIT
[29] 'P[ ]→[ ]I3' ΔFMT(CRIT)
[30] ' '

```

```

[31] 'EXPECTED COMPLETION DATE FOR PROJECT: ',F7.2' ΔFMT(E[NE;2])
[32] ' '
[33] 'SLACK: ',F7.2' ΔFMT(E[NE;4])
[34] ' '
[35] R CPM NETWORKS BRANCH TO LABEL GO30
[36] →(Z=3)/GO30
[37] R PROBABILITY ANALYSIS FOR PERT NETWORKS ONLY
[38] QUES1:'DO YOU WANT TO INPUT COMPLETION DATE(S) FOR COMPUTATION OF Z VALUES
'
[39] 'FOR THE NORMAL PROBABILITY DISTRIBUTION? ANSWER YES OR NO.'
[40] →(GO31,GO30,QUES1)['YN'\1+,']
[41] GO31:' '
[42] R COMPUTE STANDARD DEVIATION FOR THE NETWORK
[43] STDEV←(+/(C[;6]*2)×CPAC)*0.5
[44] →(STDEV=0)/GO30
[45] 'INPUT COMPLETION DATE(S):'
[46] A+,[]
[47] ' '
[48] 'NETWORK STANDARD DEVIATION:',F7.2' ΔFMT(STDEV)
[49] ' '
[50] R COMPUTE Z VALUES
[51] K←(A-E[NE;2])÷STDEV
[52] K←(1,ρK)ρK
[53] 'Z VALUES:'
[54] ' '
[55] 'X5,F7.3' ΔFMT K
[56] ' '
[57] →QUES1
[58] GO30:' '
[59] QUES2:'FOR COMPLETE EVENT AND ACTIVITY ANALYSIS, TYPE MORE.'
[60] 'TO EXIT PROGRAM TYPE STOP.'
[61] →(GO32,0,QUES2)['MS'\1+,']
[62] GO32:' '
[63] ' '
[64] '
EVENT ANALYSIS'
[65] ' '
[66] ' EVENT EXPECTED LATEST ON CRITICAL'
[67] ' NUMBER COMPLETION DATE DATE SLACK PATH'
[68] R H AND J ARE VECTORS CONTAINING EVENT NUMBERS, H[N] + J[N] REPRESENTS
AN ACTIVITY
[69] H←ACT[;1]
[70] J←ACT[;2]
[71] R ARRAY E WITH EVENT FLAG COLUMN REMOVED
[72] DATA1←(NE,4)+E
[73] G←NEρ' '
[74] R ARRAY G IS * FOR 'ON CRITICAL PATH' AND BLANK FOR 'NOT ON CRITICAL
[75] R PATH' USED IN EVENT ANALYSIS
[76] N←1
[77] GO34:→(CPEV[N]=0)/GO35
[78] G[N]←'*'
[79] GO35:→(N=NE)/GO36
[80] N←N+1
[81] →GO34
[82] GO36:' '
[83] 'X4,I3,X12,F6.2,X9,F6.2,X6,F6.2,X11,A1' ΔFMT(DATA1;G)
[84] G←NAρ' '

```

```

[85] * ARRAY G IS * FOR 'ON CRITICAL PATH' AND BLANK FOR 'NOT ON CRITICAL
[86] * PATH' USED IN ACTIVITY ANALYSIS.
[87] N+1
[88] GO38:→(CPAC[N]=0)/GO39
[89] G[N]←'*'
[90] GO39:→(N=NA)/GO40
[91] N+N+1
[92] →GO38
[93] GO40:' '
[94] ' '
[95] ' '
[96] * CPM NETWORKS BRANCH TO GO50
[97] →(Z=3)/GO50
[98] * PRODUCE PERT ACTIVITY REPORT
[99] '
[100] '
[101] * ARRAY C WITH ACTIVITY FLAG COLUMN REMOVED.
[102] DATA1+(NA,6)↑C
[103] '
[104] '
[105] '
[106] ' I3, □→□, I3, X7, F6.2, X9, F6.2, X9, F6.2, X8, F6.2, X7, F6.2, X7, F7.3, X13, A1' ΔFMT(H
; J; DATA1; G)
[107] →0
[108] GO50:' '
[109] * PRODUCE CPM ACTIVITY REPORT
[110] DATA1+(NA,5)↑C
[111] '
[112] '
[113] '
[114] '
[115] '
[116] ' I3, □→□, I3, X7, F6.2, X9, F6.2, X9, F6.2, X8, F6.2, X7, F6.2, X12, A1' ΔFMT(H; J; DATA1
; G)
▽

```

INTERACTIVE OPTIMIZATION OF A PRODUCTION SYSTEM

Thomas J. Schriber
Professor of Management Science
Graduate School of Business Administration
The University of Michigan
Ann Arbor, Michigan

DESCRIPTION

The APL model presented here uses Monte Carlo methods to simulate the behavior of a simplified production system. The system involves two independent random variables, one dependent random variable, and two decision variables. Given particular cost information, the problem is to find decision-variable values which minimize the average daily cost of the production process. This implies the need to experimentally investigate the nature of a cost surface, searching across the surface in an attempt to find its low point. In general, such a search can be conducted according to one or more formalized searching strategies, or it can be based solely on informal strategies which the investigator devises intuitively. No searching strategy is built directly into the model given here; it consequently falls to the user to supply either a formal or an informal strategy for searching the cost surface efficiently and successfully. The model supports the search by letting the user interactively estimate the cost consequences of various alternative values (s)he chooses for the decision variables.

Assignments involving use of this model, or models similar to it, can be appropriate for a variety of audiences and in a variety of settings. Some of the possibilities are suggested below.

- The model can be used (as one in a series of computer models) in a "computer applications

in business" course to illustrate one way the computer can be employed to support decision making in the firm.

- As a more specific alternative to the above, the model can be brought into play in a course in production and operations management to suggest one mode of computer use in a production context.
- The model can be used in an introductory course in a discrete-event simulation as an example of the type of modeling in which the students themselves will engage.
- It can be used in a second-level course in computer-implemented algorithms to motivate the student of methods for seeking optima.
- The model may be of interest in courses dealing with probability and statistics. The production system on which the model is based has the flavor of a practical problem involving a combination of random variables whose complete analytic manipulation is difficult or impossible; in this sense, the underlying model demonstrates an alternative method for determining the properties of dependent random variables. In addition, for any given choice of decision-variable values, the model can be used to estimate the sampling distribution of a statistic of interest (e.g., average daily cost). Finally, the model can be used to explore the behavior of the distribution of the average daily cost estimator as a function of sample size (that is, as a function of the duration of each simulation).

Use of the model does not require previous computer experience. The user should understand the fundamental concepts of "relative frequency," but these concepts can be motivated intuitively for the purpose at hand; as a result, no course in probability and/or statistics is prerequisite to use of the model. Within these guidelines, and subject to whatever other requirements may be implied in the above list of suggested applications, the model could be used by students at any level in the curriculum.

It might be noted here that the model is designed to protect itself against "bad data" which the user might accidentally or intentionally feed to it. That is, the model examines all information inputted to it by the user, and determines whether the information is feasible in the context of the problem. If infeasible information is detected, the model asks the user to re-enter the information in question. For example, the model will not accept negative values for relative frequencies, it requires that relative frequencies sum to one hundred percent, and so on. A later section in this paper displays a sample session in which a hypothetical user deliberately tries to get the model into trouble, and shows how the model responds to such a user.

Provided below, under section headings entitled Instructions for Use and Sample Use, is a "user's writeup" which includes the following:

- A description of the production system
- An indication of the problem to be solved
- A description of a formalized searching strategy
- An illustration of model use

In spirit, the user's writeup is designed to be a self-contained problem statement from which copies can be made for distribution to students. Of course, instructors who decide to use the

problem might want to modify the user's writeup to suit their own purposes. For example, the writeup presented here describes the univariate searching strategy as one which might be employed to explore the cost surface in methodical fashion. Those who assign the problem might prefer to let their students develop their own heuristics spontaneously as a user-conducted search proceeds; or, it might be thought preferable to have the students follow some formalized searching strategy other than that of univariate search.

This problem was originally composed by the author to meet several differing needs for such a problem-model combination in the business curriculum at The University of Michigan. The system described here has been modeled in FORTRAN¹ and in GPSS.² Variations on the problem described here have been presented in SIMULETTER, the quarterly publication of ACM's Special Interest Group for Simulation.³

Instructions for Use

Statement of the Problem

In a certain factory, a goal of management is to operate 50 machines of a particular kind 8 hours a day, 5 days a week. Management owns 50 machines of the kind in question, and employs 50 machine operators (one per machine), so that this operating objective can be met, at least in theory. In practice, however, a problem arises in that each of the machines is subject to random failure. Whenever a machine fails, it must of course be taken out of operation. It is then routed to an in-house repair shop,

-
- 1 See FORTRAN Applications in Business Administration, Vol. II, edited by Thomas J. Schriber and Laurence A. Madeo, The University of Michigan, Ann Arbor, Michigan, 1971, pp. 385-422.
 - 2 See Simulation Using GPSS, by Thomas J. Schriber, John Wiley & Sons, Inc., New York, 1974, pp. 112-120.
 - 3 See SIMULETTER, Vol. 5, No. 3, April 1974, pp. 75-76.

where it is eventually repaired. Meanwhile, unless backup machines are available to supplement the 50 owned machines, management's operating objective is not being met.

Management estimates that the profit opportunity which is foregone whenever fewer than 50 machines are in operation at all times during the 40-hour work week is \$22.50 per hour per machine. For example, if only 47 machines are being operated during a given 2-hour period, then the associated profit opportunity which is foregone is \$135 (\$22.50 times 3 times 2). For various reasons, any under-capacity operation which might come about cannot be made up by working overtime. Management does have the option, however, to supplement the 50 machines it owns by leasing (renting) machines of like kind on a long-term basis at a cost of \$30 per day per machine. (The leasing charges are independent of the extent to which the leased machines are actually used.) Management also can decide how many repairmen to employ for the purpose of repairing failed machines. For example, it might be decided to hire 2 repairmen, and to lease 9 machines. The Figure 1 diagram then shows what the overall state of affairs might be in the system at some arbitrary time. Each square in the diagram represents a machine. Squares without an X in them denote machines which are either being operated, or are in backup status, ready for use when needed. Squares with an X in them signify machines which either are being repaired, or are queued up waiting to go into repair. Of the 59 machines in this hypothesized situation, then, 2 are in repair; 2 are queued for repair; 5 are in backup status; and the other 59 are being operated.

Assignment

Your task is to determine for management that combination of repairmen-hired and machines-leased which will minimize the average daily cost of the production process. To help you in

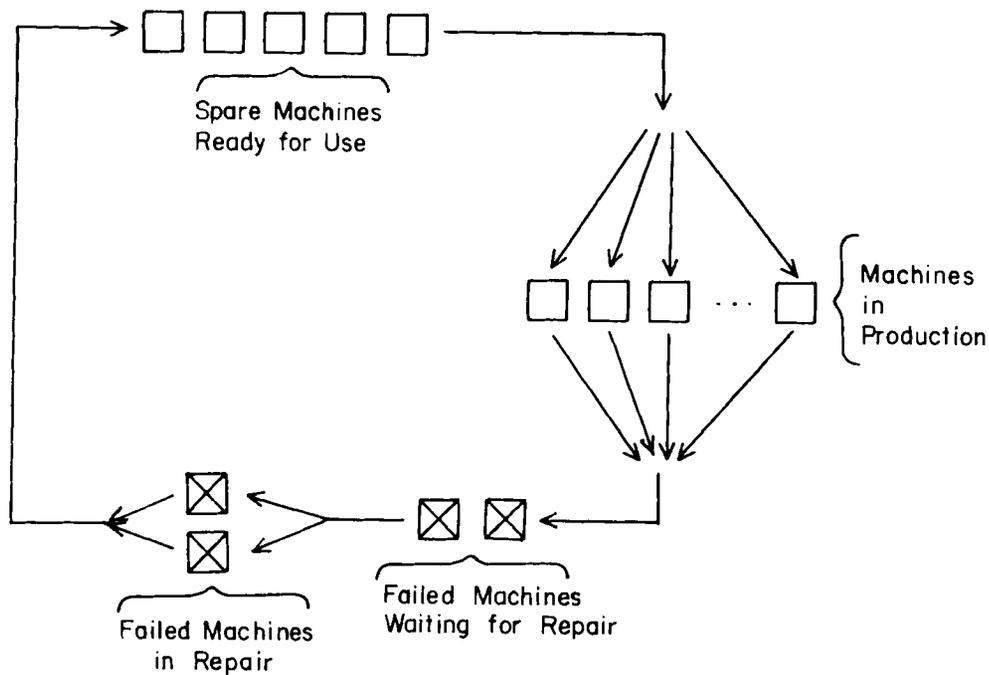


Figure 1

task, a computer model has been built to simulate the operation of the factory. You are to work with the model to conduct a search for that hire-lease combination which minimizes the average daily cost of operating the factory. Pertinent lifetime and repair time data for the machines in question are given below, and various modeling assumptions are stated. A strategy is also described for methodically searching for the minimum-cost hire-lease combination. Finally, the history of a typical session involving use of the available computer model is shown.

After scanning the sections on data and modeling assumptions, you should do the following:

- Study the section on searching strategy
- Review the specifics of the sample session

- Go to a computer terminal and use the model to perform the requested optimization.
- Write up and hand in a brief report to management, summarizing the results of your investigation and providing your final recommendations.

Data

Based on past records, the lifetime of a machine which has just been repaired is distributed as follows:

<u>Lifetime, Hours:</u>	110	120	130	140	150	160	170	180	190	200
<u>Relative Frequency, %:</u>	3	6	8	12	21	17	11	9	8	5

Records also indicate that the time required to repair a failed machine (not counting any time spent waiting to go into repair) follows the distribution given below.

<u>Repairtime, Hours:</u>	4	5	6	7	8	9	10	11	12
<u>Relative Frequency, %:</u>	7	10	16	20	15	11	10	8	3

Modeling Assumptions

The computer model which simulates operation of the factory is based on the following assumptions.

- The time required to replace a failed machine with a backup machine is negligibly small.
- Owned machines and leased machines are virtually identical to one another. They are characterized by the same lifetime and repairtime distributions. Like owned machines, when leased machines fail they are repaired in-house.
- The lifetime and repairtime distributions are independent of a machine's operating history.

- Machines are taken out of operation only when they fail.
- Repairmen always work on an individual basis, one repairman per machine being repaired. (Idle repairmen cannot team up with busy repairmen to speed up the process of completing a machine's repair.)
- Each simulation performed with the model starts with the factory at typical "steady state" conditions. For example, at startup time those machines being operated already have used up some randomly-chosen fraction of their current lifetime. (The random fraction is determined separately, machine by machine.) This means that transient phenomena do not have to be taken into account in interpreting statistics printed out by the model at the end of each simulation.

Strategy for Use of the Model

The model is to be used to estimate the hire-lease combination which minimizes the average daily cost of running the factory. Some thought should be given then, to the way the cost is expected to behave as various hire-lease combinations are considered. For a fixed number of repairmen, average daily cost will vary with "machines rented" as shown in Figure 2. With too few rented machines, costs are high because of the foregone profit opportunities resulting from under-capacity operation. With too many rented machines, costs are also high because the leasing fee is being paid to provide more supplemental machines than are necessary. Somewhere between these two extremes, the average

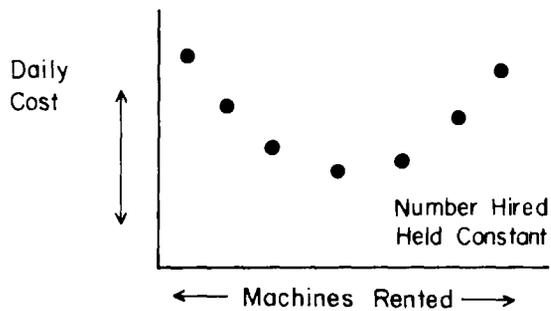


Figure 2

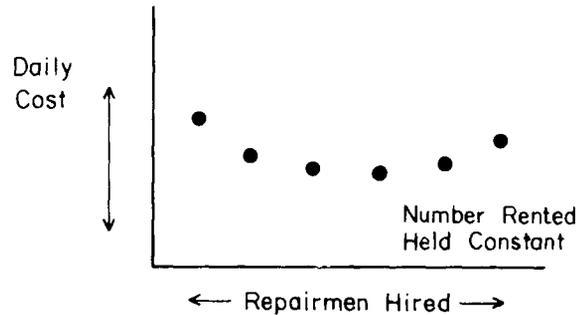


Figure 3

cost can be expected to pass through a minimum, as is suggested in Figure 2.

Similarly, for a given number of rented machines, the influence of "repairmen hired" will follow the pattern in Figure 3. With too few repairmen, high average costs are experienced because the cost component associated with under-capacity operation is large. With too many repairmen, costs are high because the wage component is larger than it need be. It is reasonable to expect that average daily cost goes through a minimum somewhere between these two extremes, as per Figure 3.

A three-dimensional perspective of the situation is suggested in Figure 4. Note that the mesh points in the grid laid down in the Figure 4 hired-rented plane represent feasible (that is, non-negative and integer-valued) choices of values for the decision-variable pair. Located above each of these mesh points, in the third dimension, is the expected value of the daily cost corresponding to that particular hire-rent alternative. (These cost points are not shown in Figure 4.) The collection of all such cost points forms a cost surface in the third

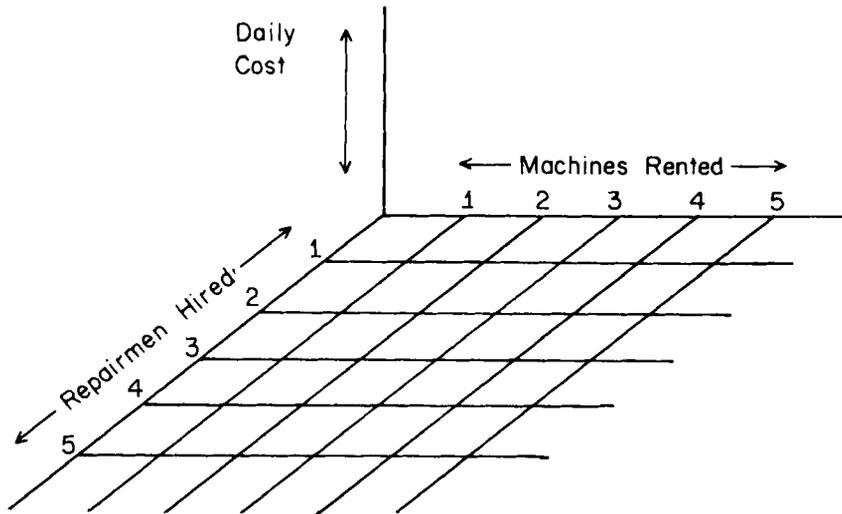


Figure 4

dimension. (The term "surface," although suggestive, is a bit misleading here. The points do not actually form a surface; instead, they simply form a collection of discrete values, with no continuum joining them.) Figures 2 and 3 are then simply contours through this cost surface.

It can be argued that the cost surface in this problem is like that of an irregularly shaped bowl. More specifically, it can be argued that the cost is a unimodal function of the two decision variables involved. A function is unimodal if the independent variables are altered in an (arbitrarily) given direction, and if the resulting values of the function are found initially to decrease down to some minimum, and then to rise again. In such functions, as the name suggests, there is only a single low point. Note the consistency between the concept of unimodality, and the shape of the Figure 2 and Figure 3 contours through the

cost surface.⁴

Given that the cost surface is unimodal, the search for the optimal hire-rent combination, then, is essentially a search for the only "lowest point" on the cost surface. This unimodality can be exploited by employing a univariate searching strategy to seek the conditions under which the cost function assumes its minimum value. Briefly, the univariate strategy involves moving through the search domain along successive contours until all more promising directions of movement have been exhausted. In the context of the problem at hand, the specifics of a univariate search are outlined below.

- (1) Perform a simulation with the model to estimate the average daily cost associated with some first choice of a hire-rent combination. (In this problem, preliminary hand computations can be carried out to estimate a reasonable hire-rent combination with which to begin.)
- (2) Holding the "number hired" constant, vary the "number rented" and perform corresponding simulations until the low point in the corresponding cost contour has been found. This corresponds to finding the low point in Figure 2.
- (3) Holding the "number rented" constant at its low-point value from (2), vary the "number hired" and perform corresponding simulations until the low point in that cost contour has been found.

4 For more information about the concept of unimodality, see Beveridge and Schechter, Optimization: Theory and Practice, McGraw-Hill, New York, 1970, and Wilde and Beightler, Foundations of Optimization, Prentice-Hall, Englewood Cliffs, N. J., 1967.

This is equivalent to finding the low point in Figure 3.

- (4) Is the final hire-rent pair in (3) the same as that in (2)? If not, return to (2). Otherwise, go to (5).
- (5) Call the best hire-rent combination found so far the tentative optimum. As a final check, simulate for the as-yet-uninvestigated hire-rent combinations (if any) surrounding this tentatively optimal combination.
- (6) Is the lowest-cost point in (5) that of the tentative optimum? If so, the search is finished. If not, let the lowest-cost point be the new tentative optimum, and return to (5).

In carrying out the steps just described, it is helpful to enter the various cost points in a grid as the search proceeds. Figure 5 provides a grid which can be used for this purpose.

It should now be clear where the univariate searching strategy gets its name. Only one decision variable is permitted to vary at a time, while the other is held constant (or, more generally, while the others are held constant). Of course, the univariate searching strategy involves a well defined, step-by-step procedure, meaning it can be programmed and need not necessarily be carried out interactively by a person sitting at a computer terminal. (For a FORTRAN subroutine which implements the generalized univariate searching strategy, see "Use of An External Optimizing Algorithm with a GPSS Model," by Robert M. Lefkowitz and Thomas J. Schriber, Proceedings of the Fifth Conference on Applications of Simulation, 1971, pp. 1-6.)

← Machines Rented →

	1	2	3	4	5	6	7	8	9	10	11
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											
11											

↑ Repairmen Hired ↓

Figure 5

Note that in a simulation context, a potential problem can arise with any searching strategy because each simulation only provides an estimate of the expected daily cost. If the variability of the estimator is high (because, for example, each simulation is of short duration), it is possible to draw a false conclusion about where the low cost point is located. (Strictly speaking, before the computer model is employed for optimization purposes, it should be used to study the behavior of the cost estimator as a function of the duration of simulation. Statistical methods should then be used either to determine point estimates of the expected cost values, or to construct interval estimates of the expected cost values at whatever is judged to be an acceptable confidence level. For simplicity here, it is simply recommended that each simulation be performed for 200 working days, and that the resulting average daily cost be taken as a point estimate of the expected cost value.)

Sample Use

Use of the computer model is illustrated in the sample sessions displayed in the following section. Lifetime and repair-time data used in these sessions are identical to those given earlier in the statement of the problem. This is also true of the cost data. The recommended simulation span of 200 days has been used for each of the two simulations performed in the sample session.

The first sample session should be largely self-explanatory. The model requests information from the user by issuing questions such as HOW MANY VALUES CAN THE LIFETIME RANDOM VARIABLE ASSUME? and commands such as ENTER THE VARIOUS LIFETIME VALUES. The user responds by typing in numbers on some occasions and by typing in words on other occasions. The dialogue between the model and the

user is easily followed simply by reading through this sample session. With the sample session in hand, even a first-time computer user can sit at a terminal and engage in a similar dialogue with the model almost without effort.

The second sample session is designed to demonstrate how the model analyzes the user inputs for feasibility. As in the first session, this second session should be self-explanatory. As this session suggests, every attempt has been made to design the model to trap infeasible data inputs, notify the user when infeasible data are encountered, and give the user an immediate opportunity to type in acceptable information.

Comments on the Program Listing

Following the sample sessions is a complete listing of the model. Except for using the public functions *IPI*, *INP*, and *ΔFMT*, the model is self-contained. (Code for the input functions *IPI* and *INP* is provided in an appendix to this monograph. *ΔFMT* is a proprietary function of the Scientific Time Sharing Corporation, and is available in their APL/Plus System. The user who does not have access to APL/Plus can substitute his own logic in place of *ΔFMT*. In this model, that would not be unduly difficult to do.)

The model consists of 15 functions (modules). The first of these functions, *MASTERΔCONTROL*, acts as a logical main program. *MASTERΔCONTROL* systematically routes control to each of the other functions which make up the overall package. The names of these other functions have been chosen to suggest their role in context (e.g., *GETΔLIFEΔDATA*, *GETΔREPAIRΔDATA*, *SIMULATE*, *PRODUCEΔREPORT*). All variables in the model are global in type, which paves the way for ease of communication among the various modules.

The last point to be made involves the amount of CPU time required to simulate with this model. Each 200-day simulation performed in the first sample session required about 12 CPU seconds on UCLA's IBM 360/91 computer. Of course, Monte Carlo simulations are known for the computer time they may require, and this is compounded here by the fact that APL is an interpretive language. Those who might choose to use this model for pedagogical purposes should take this rather large CPU-time requirement into account.

PROGRAM USAGE

)LOAD KUNΔPACKAGE
SAVED 16.18.21 07/11/75

HOW MANY VALUES CAN THE LIFETIME RANDOM VARIABLE ASSUME?
(ENTER AN UNSIGNED INTEGER > 0): 1

ENTER THE VARIOUS LIFETIME VALUES: 225

ENTER THE CORRESPONDING RELATIVE FREQUENCIES
(AS PERCENTAGES): 100

HOW MANY VALUES CAN THE REPAIRTIME RANDOM VARIABLE ASSUME?
(ENTER AN UNSIGNED INTEGER > 0): FIDDLESTICKS
PLEASE ENTER POSITIVE INTEGER(S): -5
PLEASE ENTER POSITIVE INTEGER(S): 0
TRY AGAIN

HOW MANY VALUES CAN THE REPAIRTIME RANDOM VARIABLE ASSUME?
(ENTER AN UNSIGNED INTEGER > 0): +4
PLEASE ENTER POSITIVE INTEGER(S): 4

ENTER THE VARIOUS REPAIRTIME VALUES: 50
ENTER 3 MORE VALUES 60 70 80 90
TOO MANY VALUES ENTERED, SHOULD HAVE ENTERED 3 VALUES
ENTER 3 MORE VALUES 60 70 80

ENTER THE CORRESPONDING RELATIVE FREQUENCIES
(AS PERCENTAGES): 15 20 25 20
YOUR RELATIVE FREQUENCIES SUM TO 80 PERCENT
TRY AGAIN

ENTER THE CORRESPONDING RELATIVE FREQUENCIES
(AS PERCENTAGES): 20 25 30 25

ENTER THE REPAIRMEN'S WAGES (\$/MAN-HOUR): -5.75
NEGATIVE VALUES ARE NOT VALID, TRY AGAIN

ENTER THE REPAIRMEN'S WAGES (\$/MAN-HOUR): THEY EARN TOO MUCH!
NONE NUMERIC INPUTS ARE INVALID, TRY AGAIN

ENTER THE REPAIRMEN'S WAGES (\$/MAN-HOUR): 7.50

ENTER THE MACHINE LEASING CHARGE (\$/MACHINE-DAY): \$25.50
NONE NUMERIC INPUTS ARE INVALID, TRY AGAIN

ENTER THE MACHINE LEASING CHARGE (\$/MACHINE-DAY): 25.50

ENTER THE PRODUCTION CAPACITY (I.E., THE NUMBER
OF MACHINES WHICH CAN BE OPERATED SIMULTANEOUSLY): 0

TRY AGAIN

ENTER THE PRODUCTION CAPACITY (I.E., THE NUMBER
OF MACHINES WHICH CAN BE OPERATED SIMULTANEOUSLY): 50

ENTER THE LOST PRODUCTION PENALTY (\$/MACHINE-HOUR): +10
NONE NUMERIC INPUTS ARE INVALID, TRY AGAIN
ENTER THE LOST PRODUCTION PENALTY (\$/MACHINE-HOUR): 10

HOW MANY REPAIRMEN SHOULD THERE BE IN TOTAL?: NONE
PLEASE ENTER POSITIVE INTEGER(S): 0
TRY AGAIN

HOW MANY REPAIRMEN SHOULD THERE BE IN TOTAL?: 5

HOW MANY LEASED MACHINES SHOULD THERE BE IN TOTAL?: 0

FOR HOW MANY 8-HOUR DAYS DO YOU WANT TO SIMULATE?: 0

TRY AGAIN

FOR HOW MANY 8-HOUR DAYS DO YOU WANT TO SIMULATE?: 500
PLEASE ENTER POSITIVE INTEGER(S): 250

---REPORT ON SIMULATION RESULTS---

NO. OF DAYS SIMULATED:	250
REPAIRMEN HIRED:	5
AVERAGE REPAIRMEN IN USE:	4.97
DAILY PAYROLL EXPENSE:	\$300.00
MACHINES LEASED:	0
DAILY LEASING EXPENSE:	\$0.00
AVERAGE DAILY UNDER-PRODUCTION LOSSES:	\$2,548.85

AVERAGE DAILY COST:	\$2,874.35

WHAT DO YOU WANT TO DO NEXT?

YOUR CHOICES ARE AS FOLLOWS...

IF YOU YOU WILL BE GIVEN A CHANCE TO
TYPE IN... SET A NEW VALUE FOR...

HIRED MEN-HIRED
LEASED MACHINES-LEASED
RUN-LENGTH DURATION OF SIMULATION

IF YOU
TYPE IN... WHAT WILL HAPPEN IS...

SIMULATE ANOTHER SIMULATION WILL BE PERFORMED
DONE YOUR USE OF THE MODEL WILL END
EXPLAIN THIS LIST OF CHOICES WILL BE TYPED OUT

(YOU ONLY NEED TYPE IN THE FIRST LETTER IN EACH CASE)

WHAT IS YOUR CHOICE?

HIRED

HOW MANY REPAIRMEN SHOULD THERE BE IN TOTAL?: 3

WHAT DO YOU WANT TO DO NEXT?

LEASED

HOW MANY LEASED MACHINES SHOULD THERE BE IN TOTAL?: 5

WHAT DO YOU WANT TO DO NEXT?

SIMULATE

-----REPORT ON SIMULATION RESULTS-----

NO. OF DAYS SIMULATED:	200
REPAIRMEN HIRED:	3
AVERAGE REPAIRMEN IN USE:	2.43
DAILY PAYROLL EXPENSE:	\$162.00
MACHINES LEASED:	5
DAILY LEASING EXPENSE:	\$150.00
AVERAGE DAILY UNDER-PRODUCTION LOSSES:	\$58.08

AVERAGE DAILY COST:	\$250.08

WHAT DO YOU WANT TO DO NEXT?

)LOAD RUNΔPACKAGE
SAVED 16.18.21 07/11/75

HOW MANY VALUES CAN THE LIFETIME RANDOM VARIABLE ASSUME?
(ENTER AN UNSIGNED INTEGER > 0): 10

ENTER THE VARIOUS LIFETIME VALUES: 110 120 130 140 150 160 170 180
ENTER 2 MORE VALUES 190 200

ENTER THE CORRESPONDING RELATIVE FREQUENCIES
(AS PERCENTAGES): 3 6 8 12 21 17 11 9 8 5

HOW MANY VALUES CAN THE REPAIRTIME RANDOM VARIABLE ASSUME?
(ENTER AN UNSIGNED INTEGER > 0): 9

ENTER THE VARIOUS REPAIRTIME VALUES: 4 5 6 7 8 9 10 11 12

ENTER THE CORRESPONDING RELATIVE FREQUENCIES
(AS PERCENTAGES): 7 10 16 20 15 11 10 8 3

ENTER THE REPAIRMEN'S WAGES (\$/MAN-HOUR): 6.75

ENTER THE MACHINE LEASING CHARGE (\$/MACHINE-DAY): 30

ENTER THE PRODUCTION CAPACITY (I.E., THE NUMBER
OF MACHINES WHICH CAN BE OPERATED SIMULTANEOUSLY): 50

ENTER THE LOST PRODUCTION PENALTY (\$/MACHINE-HOUR): 22.50

HOW MANY REPAIRMEN SHOULD THERE BE IN TOTAL?: 5

HOW MANY LEASED MACHINES SHOULD THERE BE IN TOTAL?: 3

FOR HOW MANY 8-HOUR DAYS DO YOU WANT TO SIMULATE?: 200

-----REPORT ON SIMULATION RESULTS-----

NO. OF DAYS SIMULATED:	200
REPAIRMEN HIRED:	5
AVERAGE REPAIRMEN IN USE:	2.39
DAILY PAYROLL EXPENSE:	\$270.00
MACHINES LEASED:	3
DAILY LEASING EXPENSE:	\$90.00
AVERAGE DAILY UNDER-PRODUCTION LOSSES:	\$67.12

AVERAGE DAILY COST:	\$367.12

WHAT DO YOU WANT TO DO NEXT?

LEASED

HOW MANY LEASED MACHINES SHOULD THERE BE IN TOTAL?: 6

WHAT DO YOU WANT TO DO NEXT?

SIMULATE

-----REPORT ON SIMULATION RESULTS-----

NO. OF DAYS SIMULATED:	200
REPAIRMEN HIRED:	3
AVERAGE REPAIRMEN IN USE:	2.38
DAILY PAYROLL EXPENSE:	\$162.00
MACHINES LEASED:	6
DAILY LEASING EXPENSE:	\$180.00
AVERAGE DAILY UNDER-PRODUCTION LOSSES:	\$4.82

AVERAGE DAILY COST:	\$196.82

WHAT DO YOU WANT TO DO NEXT?

DONE

PROGRAM DESCRIPTION

```
▽ PRODUCEAREPORT
[1] CR, '____REPORT ON SIMULATION RESULTS____', CR
[2] 'NO. OF DAYS SIMULATED: ', I10' ΔFMT(DAYS)
[3] 'REPAIRMEN HIRED: ', I16' ΔFMT(HIRED)
[4] 'AVERAGE REPAIRMEN IN USE: ', F10.2' ΔFMT(MENΔUSE÷CLOCK)
[5] PAY÷8×HIRED×WAGEΔRATE
[6] 'DAILY PAYROLL EXPENSE: ', CP$, F13.2' ΔFMT(PAY)
[7] 'MACHINES LEASED: ', I16' ΔFMT(LEASED)
[8] 'DAILY LEASING EXPENSE: ', CP$, F13.2' ΔFMT(LEASED×LEASEΔRATE)
[9] LPΔLOSS÷((PRODACAPACITY×CLOCK)-MACHΔUSE)×LOSSΔRATE÷DAYS
[10] 'AVERAGE DAILY'
[11] 'UNDER-PRODUCTION LOSSES: ', CP$, F11.2' ΔFMT(LPΔLOSS)
[12] (35p'-')
[13] 'AVERAGE DAILY COST: ', CP$, F16.2' ΔFMT(+ /PAY, LEASEΔRATE, LPΔLOSS)
[14] (35p'-')
```

```
▽ EXPLAINΔCHOICES
[1] CR, 'YOUR CHOICES ARE AS FOLLOWS...', CR
[2] 'IF YOU YOU WILL BE GIVEN A CHANCE TO'
[3] 'TYPE IN... SET A NEW VALUE FOR...', CR
[4] 'HIRED MEN-HIRED'
[5] 'LEASED MACHINES-LEASED'
[6] 'RUN-LENGTH DURATION OF SIMULATION', CR, CR, 'IF YOU'
[7] 'TYPE IN... WHAT WILL HAPPEN IS...', CR
[8] 'SIMULATE ANOTHER SIMULATION WILL BE PERFORMED'
[9] 'DONE YOUR USE OF THE MODEL WILL END'
[10] 'EXPLAIN THIS LIST OF CHOICES WILL BE TYPED OUT', CR
[11] '(YOU ONLY NEED TYPE IN THE FIRST LETTER IN EACH CASE)'
[12] CR, 'WHAT IS YOUR CHOICE?', CR
```

```

      ▽ GETΔWAGEΔDATA
[1]  ANOTE THE USE OF ''INP'' IN THIS FUNCTION.
[2]  WAGEΔRATE←1 INP PROMPT7
      ▽

      ▽ GETΔLEASINGΔDATA
[1]  ANOTE THE USE OF ''INP'' IN THIS FUNCTION.
[2]  LEASEΔRATE←1 INP PROMPT8
      ▽

      ▽ GETΔPRODΔCAPACITY
[1]  ANOTE THE USE OF ''IPI'' IN THIS FUNCTION.
[2]  REPEAT:→REPEAT×1ρ□+(1>PRODΔCAPACITY←1↑IPI PROMPT10)/'TRY AGAIN'
      ▽

      ▽ GETΔLOSTΔPRODΔPENALTY
[1]  ANOTE THE USE OF ''INP'' IN THIS FUNCTION.
[2]  LOSSΔRATE←1 INP PROMPT9
      ▽

      ▽ GETΔMENΔHIRED
[1]  ANOTE THE USE OF ''IPI'' IN THIS FUNCTION
[2]  REPEAT:→REPEAT×1ρ□+(1>HIRED←1↑IPI PROMPT11)/'TRY AGAIN'
      ▽

      ▽ GETΔMACHINESΔLEASED
[1]  ANOTE THE USE OF ''IPI'' IN THIS FUNCTION
[2]  REPEAT:→REPEAT×1ρ□+(0>LEASED←1↑IPI PROMPT12)/'TRY AGAIN'
      ▽

      ▽ GETΔSIMΔSPAN
[1]  ANOTE THE USE OF ''IPI'' IN THIS FUNCTION
[2]  REPEAT:→REPEAT×1ρ□+(1>DAYS←1↑IPI PROMPT13)/'TRY AGAIN'
      ▽

```

```

▽ K←CH MM MT;G
[1] K←(MT=CH)/ιρMT
[2] G←[ /K+K-1, 1+-1+R
[3] MT←(MT≠CH)/MT
[4] K←((ρK),G)ρ(,R°.≥ιG)\(MT≠CH)/MT
▽

```

```

▽ K←NIP P;O;S;I;D;E;T
[1] A NUMERIC INPUT;AUTHOR: ROY SYKES, STSC.
[2] →4,O+-?1
[3] E:P+'INCORRECT NUMBER...PLEASE REENTER',(1+(0=ρE)ρ':'),E
[4] E←(1+T+O+Pι'a')↓P+,P
[5] P←T↑P
[6] →(0 1 =ρT+AKI P)/ 6 9
[7] T←O+'0123456789.-E'ιT
[8] →(13εT+(S+((P+1)↑T)ε 11 12)↑T)φ 15 11
[9] →(11 10 0 ≤T+O+'0123456789□'ιT)/E, 10 20
[10] →0,ρK+□
[11] P←(1+D+O+Tι13)↑T
[12] →(√/10≤P+(I+11=1↑P)↓P)ρE
[13] →(0ε(ρE),ρT+D↑T)ρE
[14] P←10*(-1*I)×10ιP
[15] →((√/T≥11)√10εD+(1+I+O+Tι10)↑T)ρE
[16] →(0=ρT+(I↑T),D)ρE
[17] T←P×(-1 0.1 ×. *S,ρD)×10ιT
[18] →19+T≠|T
[19] T←|T
[20] K←T
▽

```

```

▽ DRAW←CDF SAMFCN RVVALS
[1] ATHIS FUNCTION RETURNS A VALUE DRAWN FROM THE POPULATION 'RVVALS' FOR
[2] WHICH THE CORRESPONDING CUMULATIVE DISTRIBUTION FUNCTION IS 'CDF'
[3] DRAW←(,RVVALS)[((-1+?100000)÷100000)≤,CDF)ι1]
▽

```

```

▽ SEPARATE S;B
[1] NUM←(B+ΔVI S)/ΔFI S
[2] ALPH←(∼B)†ΔMI S
▽

```

```

▽ DEFINEΔMAJORΔPROMPTS
[1]  A THESE PROMPTS ARE USED BY THE INPUT FUNCTIONS CALLED IN
[2]  A LINES 7-15 AND 27-29 IN THE MASTER CONTROL FUNCTION.
[3]  A DEFINE QR TO BE A CARRIAGE RETURN CHARACTER
[4]  QR←'
      '
[5]  PROMPT1←QR, 'HOW MANY VALUES CAN THE LIFETIME RANDOM VARIABLE ASSUME? '
[6]  PROMPT1←PROMPT1, QR, '(ENTER AN UNSIGNED INTEGER > 0): '
[7]  PROMPT2←'ENTER THE VARIOUS LIFETIME VALUES: '
[8]  PROMPT3←QR, 'ENTER THE CORRESPONDING RELATIVE FREQUENCIES'
[9]  PROMPT3←PROMPT3, QR, '(AS PERCENTAGES): '
[10] PROMPT4←QR, 'HOW MANY VALUES CAN THE REPAIRTIME RANDOM VARIABLE ASSUME?'
[11] PROMPT4←PROMPT4, QR, '(ENTER AN UNSIGNED INTEGER > 0): '
[12] PROMPT5←'ENTER THE VARIOUS REPAIRTIME VALUES: '
[13] PROMPT6←QR, 'ENTER THE CORRESPONDING RELATIVE FREQUENCIES'
[14] PROMPT6←PROMPT6, QR, '(AS PERCENTAGES): '
[15] PROMPT7←QR, 'ENTER THE REPAIRMEN'S WAGES ($/MAN-HOUR): '
[16] PROMPT8←QR, 'ENTER THE MACHINE LEASING CHARGE ($/MACHINE-DAY): '
[17] PROMPT9←'ENTER THE LOST PRODUCTION PENALTY ($/MACHINE-HOUR): '
[18] PROMPT10←QR, 'ENTER THE PRODUCTION CAPACITY (I.E., THE NUMBER '
[19] PROMPT10←PROMPT10, QR, 'OF MACHINES WHICH CAN BE OPERATED SIMULTANEOUSLY):'
[20] PROMPT11←QR, 'HOW MANY REPAIRMEN SHOULD THERE BE IN TOTAL?: '
[21] PROMPT12←'HOW MANY LEASED MACHINES SHOULD THERE BE IN TOTAL?: '
[22] PROMPT13←'FOR HOW MANY 8-HOUR DAYS DO YOU WANT TO SIMULATE?: '
▽

```

```

▽ GETΔLIFEΔDATA
[1]  A THIS FUNCTION MAKES DIRECT USE OF THE INPUT FUNCTIONS 'IPI'
[2]  A AND 'INP'. (SEE THE MONOGRAPH APPENDICES FOR DETAILS)
[3]  REPEAT:→REPEAT×10□+(1>LTIMES+1↑IPI PROMPT1)'/TRY AGAIN'
[4]  LTVALS←LTIMES INP PROMPT2
[5]  AGAIN:LTDIST←+0.01×LTDIST←LTIMES INP PROMPT3
[6]  +(((~1↑LTDIST)>0.99)^(~1↑LTDIST<1.01))/SET2ONE
[7]  'YOUR RELATIVE FREQUENCIES SUM TO ';100×~1↑LTDIST;' PERCENT'
[8]  'TRY AGAIN' ◇ →AGAIN
[9]  SET2ONE:LTDIST[LTIMES]←1 ◇ →0
▽

```

```

▽ GETΔREPAIRΔDATA
[1]  A THIS FUNCTION MAKES DIRECT USE OF THE INPUT FUNCTIONS 'IPI'
[2]  A AND 'INP'. (SEE THE MONOGRAPH APPENDICES FOR DETAILS)
[3]  REPEAT:→REPEAT×10□+(1>RTIMES+1↑IPI PROMPT4)'/TRY AGAIN'
[4]  RTVALS←RTIMES INP PROMPT5
[5]  AGAIN:RTDIST←+0.01×RTDIST←RTIMES INP PROMPT6
[6]  +(((~1↑RTDIST)>0.99)^(~1↑RTDIST<1.01))/SET2ONE
[7]  'YOUR RELATIVE FREQUENCIES SUM TO ';100×~1↑RTDIST;' PERCENT'
[8]  'TRY AGAIN' ◇ →AGAIN
[9]  SET2ONE:RTDIST[RTIMES]←1 ◇ →0
▽

```

```

V MASTERΔCONTROL
[1] ATHE MASTER CONTROL FUNCTION ROUTES CONTROL DIRECTLY TO
[2] A13 OF THE OTHER 14 FUNCTIONS WHICH CONSTGITUTE THE MODEL.
[3] A(THE 14TH FUNCTION, 'SAMFCN', IS CALLED FROM THE
[4] AFUNCTION 'SIMULATE'.) THE VARIOUS FUNCTION NAMES
[5] ARE DESIGNED TO REVEAL THE FUNCTION ROLES.
[6] DEFINEΔMAJORΔPROMPTS
[7] GETΔLIFEΔDATA
[8] GETΔREPAIRΔDATA
[9] GETΔWAGEΔDATA
[10] GETΔLEASEΔDATA
[11] GETΔPRODΔCAPACITY
[12] GETΔLOSTΔPRODΔPENALTY
[13] GETΔMENΔHIRED
[14] GETΔMACHINESΔLEASED
[15] GETΔSIMΔSPAN
[16] ASET CODE TO CONTROL PRINTING OF EXPLANATIONS
[17] A(SEE LINES 22-25)
[18] CODE←0
[19] SIM:SIMULATE
[20] PRODUCEΔREPORT
[21] MORE:CR,'WHAT DO YOU WANT TO DO NEXT?',CR
[22] →(CODE=1)/SKIP
[23] EXPLAINΔCHOICES
[24] CODE←1
[25] SKIP:→SKIP×1p□←(1≠(CHAR←1↑□)∈('HLRSDE'))/'TRY AGAIN'
[26] →(HIRE,LEASE,SPAN,SIM,DONE,EXPLAIN)['HLRSDE'\CHAR]
[27] HIRE:GETΔMENΔHIRED ◇ →MORE
[28] LEASE:GETΔMACHINESΔLEASED ◇ →MORE
[29] SPAN:GETΔSIMΔSPAN ◇ →MORE
[30] EXPLAIN:EXPLAINΔCHOICES ◇ →MORE
[31] DONE:→
V

```

```

V SIMULATE
[1]  THE SIMULATION ITSELF IS PERFORMED IN THIS MODULE.
[2]  INITIALIZE PERTINENT VARIABLES
[3]  CLOCK←FIXQ←MACHΔUSE←MENΔUSE←0 ◇ RCTIMES←BDTIMES←10
[4]  INUSE←PRODACAPACITY
[5]  TOP←BDTIMES←BDTIMES,(LTDIST SAMFCN LTVALS)×(1+?1000)÷1000
[6]  →(PRODACAPACITY>ρBDTIMES)/TOP
[7]  MENΔIDLE←HIRED ◇ READY←LEASED ◇ STOPTIME←DAYS×8
[8]  FIND IMMEDIATE EVENT TIME; UPDATE USE STATISTICS AND CLOCK
[9]  NEXTΔEVENT:ETIME←1/BDTIMES,RCTIMES,STOPTIME
[10] MENΔUSE←MENΔUSE+(HIRED-MENΔIDLE)×(JUMP+ETIME-CLOCK)
[11] MACHΔUSE←MACHΔUSE+INUSE×JUMP ◇ CLOCK+ETIME
[12] USAGE←USAGE+INUSE×(ETIME-CLOCK) ◇ CLOCK+ETIME
[13] BRANCH TO IMMEDIATE EVENT, OR BRANCH OUT WHEN DONE
[14] →((ETIME∈RCTIMES),ETIME∈BDTIMES)/FIXED,FAILED ◇ →0
[15] UPDATING IN RESPONSE TO A MACHINE BREAKDOWN FOLLOWS
[16] FAILED:BDTIMES←(1+(HOLD↑BDTIMES)),(HOLD↑BDTIMES\ETIME)↑BDTIMES
[17] BRANCH IF FAILED MACHINE MUST QUEUE UP FOR REPAIR
[18] →Q2FIX×\MENΔIDLE=0 ◇ MENΔIDLE←MENΔIDLE-1
[19] RCTIMES←RCTIMES,CLOCK←RTDIST SAMFCN RTVALS
[20] BRANCH IF NO OTHER MACHINES ARE READY FOR USE
[21] TESTΔREADYQ:→(READY=0)/NONEΔREADY ◇ READY←READY-1
[22] →NEXTΔEVENT,BDTIMES←BDTIMES,CLOCK←LTDIST SAMFCN LTVALS
[23] Q2FIX:→TESTΔREADYQ,FIXQ←FIXQ+1
[24] NONEΔREADY:→NEXTΔEVENT,INUSE←INUSE-1
[25] UPDATING IN RESPONSE TO REPAIR COMPLETION FOLLOWS
[26] FIXED:RCTIMES←(1+(HOLD↑RCTIMES)),(HOLD↑RCTIMES\ETIME)↑RCTIMES
[27] BRANCH IF THE REPAIRED MACHINE IS NEEDED NOW
[28] →NEEDED×\INUSE≠PRODACAPACITY
[29] BRANCH TO PUT REPAIRMAN BACK TO WORK (IF POSSIBLE)
[30] →QTEST,READY←READY+1
[31] NEEDED:INUSE←INUSE+1 ◇ BDTIMES←BDTIMES,CLOCK←LTDIST SAMFCN LTVALS
[32] BRANCH IF ANOTHER FAILED MACHINE IS WAITING FOR THIS REPAIRMAN
[33] QTEST:→NEXTΔREPAIR×\FIXQ≠0
[34] →NEXTΔEVENT,MENΔIDLE←MENΔIDLE+1
[35] NEXTΔREPAIR:FIXQ←FIXQ-1
[36] →NEXTΔEVENT,RCTIMES←RCTIMES,CLOCK←RTDIST SAMFCN RTVALS
V

```


INTERACTIVE INVENTORY SIMULATOR

Ralph H. Sprague, Jr.
 Professor
 College of Business Administration
 University of Hawaii
 Honolulu, Hawaii

DESCRIPTION

Introduction

This system of APL programs allows students to obtain an optimum decision for economic order quantity (Q) and reorder point (R) in a stochastic inventory system by iterative search. The iterative search procedure allows the student to experiment with alternative methods of convergence on the optimum solution. One of the functions can also be used to reinforce the concept of appropriate choice of simulation run length based on the sampling distribution of resultant cost. The major functions in the system are:

- SET* - To set initial values
- SETRL* - To experiment with simulation run length
- SIM* - To perform one simulation run
- SEARCH* - To search for optimum, Q, R
- ENTDDIST* - To enter probability distributions
- ENTLDIST*
- HPLOT* - To display sampling distributions
- SHIST*

For a detailed description of each major function and its purpose, see the listing of *DESCRIBE*.

This exercise is useful in a variety of courses, including those in operations research, production/operations management,

computer simulation, and those dealing with computer-based decision models. The flexibility of its use makes it appropriate for graduate or undergraduate courses.

It can also be used by students with a variety of backgrounds and capabilities in APL. The student who knows little or no APL will call the first function and respond to the questions and instructions given by the programs. A student with some APL knowledge may enter the required data input using the assignment operator, use calculator mode to do some preliminary calculations on EOQ, and execute each major function when needed. The fully knowledgeable APL user may write his own functions utilizing *SIM* to implement a search algorithm, or perform a sensitivity analysis.

Instructions for Use

Several types of data must be entered before running the system. See the listing of *SETHOW* for specific variable names and descriptions.

1. Cost Coefficients - (3 cost coefficients are required): order cost, carrying cost, and stockout cost. Note that the latter two are unit daily cost. Only positive numbers are acceptable.
2. Probability Distributions - (2 empirical probability distributions are required): one for daily demand and one for order lead time. These must be entered by using the function *SET* or the functions *ENTDDIST* and *ENTLDDIST* (which are called by *SET*).

3. Initial conditions: beginning inventory and initial values for Q and R. All three may be entered through *SET* or by the assignment operator directly. Only positive integers will be accepted.
4. Run length: a positive integer which can be entered using *SET* or by using the assignment operator. Only a positive integer is acceptable. The function *SETRL* allows the student to replicate simulation runs of a given length and display the sample of resultant average daily costs. This reveals the variability of response from a simulation with a short run length and allows the choice of appropriate length run. Note that the program will not accept more than 20 replications, or a run length of more than 250 days. Even these limits may be excessively expensive in CPU time on some APL installations.

After the required data are entered, the user begins the search for an optimum Q and R decision by executing the function *SEARCH*. This function accepts the value for Q and R, calls *SIM* and returns the average daily cost of that run. The student must first identify the appropriate search region and decide whether to conduct a coarse search over a large area or a fine search over a smaller area. He may use a primitive search algorithm which iteratively scans a relevant range for Q while R is fixed, then increments R and scans again over Q, etc. Alternatively, he may use a more sophisticated search algorithm such as the simplex technique described in Meier, *et al.*¹ The student may be encouraged

1. Meier, Robert C., William T. Newell, Harold L. Pazar, Simulation in Business and Economics, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1969, pp. 313 ff.

to use a Q, R grid to track the progress of his search.

If the student writes his own function to conduct the search, he may wish to use the simulation function *SIM* directly. It requires all the variables described above and calculates the average daily cost of a run. Total cost of each type is also available in global variables if needed (see below). For a further description of the logic of the simulation modeule, see a similar FORTRAN model in Schriber.²

Summary of Functions and Variables

The two following tables summarize the functions and the global variables in the *INVΔSIM* workspace.

FUNCTIONS

<u>FUNCTION NAME</u>	<u>PURPOSE</u>	<u>CALLED BY</u>	<u>CALLS</u>
<i>DESCRIBE</i>	Gives basic description of <i>INVΔSIM</i>		<i>SETHOW</i>
<i>DISTHOW</i>	Gives instructions for entering probability distributions	<i>SETHOW</i>	
<i>ENTDDIST</i>	Enters probability distribution for demand	<i>SET</i>	<i>DISTHOW</i>

2. Schriber, Thomas J., Fortran Case Studies for Business Applications, John Wiley & Sons, Inc., 1969, pp. 10-1 ff.

<u>FUNCTION NAME</u>	<u>PURPOSE</u>	<u>CALLED BY</u>	<u>CALLS</u>
<i>ENTLDIST</i>	Enters probability distribution for lead time	<i>SET</i>	<i>DISTHOW</i>
<i>FR</i>	Constructs frequency vector	<i>HPLOT</i>	
<i>HPLOT</i>	Constructs scale for histogram of sampling distribution	<i>SETRL</i>	<i>SHIST, VS</i>
<i>SEARCH</i>	Permits iterative simulations with different Q, R		<i>SIM</i>
<i>SET</i>	Accepts input data		<i>SETHOW, ENTDDIST, ENTLDIST, SETRL</i>
<i>SETHOW</i>	Gives input instructions for the use of <i>SET</i>	<i>DESCRIBE, SET</i>	<i>DISTHOW</i>
<i>SETRL</i>	Replicates <i>SIM</i> with varying run length	<i>SET</i>	<i>SIM, HPLOT</i>
<i>SHIST</i>	Prints histogram of sampling distribution	<i>HPLOT</i>	
<i>SIM</i>	Simulates the inventory system	<i>SETRL, SEARCH</i>	
<i>VS, AND</i>	Utilities used for plotting	<i>HPLOT</i>	
<i>AKI, AYN, INP, ΔFI, ΔFMT, ΔVI</i>	Input/Output utilities		

GLOBAL VARIABLE

<u>NAME</u>	<u>DESCRIPTION</u>	<u>INITIALIZED IN</u>	<u>USED IN</u>
<i>ADC</i>	<u>A</u> verage <u>D</u> aily <u>C</u> ost	<i>SIM</i>	<i>SEARCH, SETRL</i>
<i>BINV</i>	<u>B</u> eginning <u>I</u> NVENTORY	<i>SET</i>	<i>SIM</i>
<i>CAT</i>	Number of <u>C</u> ATEGORIES in histogram	<i>HPLOT</i>	<i>SHIST</i>
<i>CC</i>	<u>C</u> arrying <u>C</u> ost	<i>SET</i>	<i>SIM</i>
<i>CO</i>	<u>C</u> ost per <u>O</u> rders	<i>SET</i>	<i>SIM</i>
<i>CS</i>	Cost of Stockout	<i>SET</i>	<i>SIM</i>
<i>DDIST</i>	<u>D</u> emand <u>D</u> ISTRIBUTION as a 100 element vector	<i>ENTDDIST</i>	<i>SIM</i>
<i>FLAG</i>	A <u>F</u> LAG to fix scale of histogram during one session	<i>SETRL, HPLOT</i>	<i>HPLOT</i>
<i>INC</i>	<u>I</u> NCrement of histogram scale	<i>HPLOT</i>	
<i>LDIST</i>	<u>L</u> ead time <u>D</u> ISTRIBUTION as a 100 element vector	<i>ENTLDIST</i>	<i>SIM</i>
<i>MIN</i>	<u>M</u> INimum range of the histogram scale	<i>HPLOT</i>	
<i>Q</i>	Order <u>Q</u> uantity	<i>SET</i>	<i>SIM</i>
<i>R</i>	<u>R</u> eorder point	<i>SET</i>	<i>SIM</i>
<i>RL</i>	<u>R</u> un <u>L</u> ength in days	<i>SET, SETRL</i>	<i>SIM</i>
<i>S</i>	Vector of frequencies, <u>S</u> , for histogram	<i>HPLOT</i>	<i>SHIST</i>
<i>SAMPLE</i>	Vector of <u>S</u> AMPLE costs from replication of	<i>SETRL</i>	<i>HPLOT</i>
<i>SCALE</i>	<u>S</u> CALE for histogram	<i>HPLOT</i>	<i>SHIST</i>
<i>TCC</i>	<u>T</u> otal <u>C</u> arrying <u>C</u> ost for 1 run	<i>SIM</i>	
<i>TCO</i>	<u>T</u> otal <u>C</u> ost of <u>O</u> rders for 1 run	<i>SIM</i>	
<i>TCS</i>	<u>T</u> otal <u>C</u> ost for <u>S</u> tockout for 1 run	<i>SIM</i>	

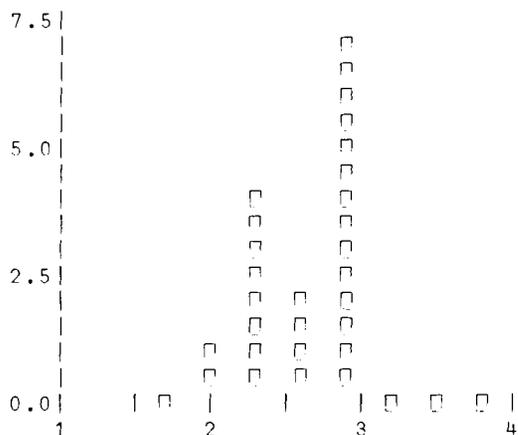
PROGRAM USAGE

```
SET
ENTRY OF PARAMETERS FOR INVENTORY SIMULATION
DO YOU NEED INSTRUCTIONS?
YES OR NO N
ENTER CARRYING COST PER UNIT PER PERIOD
? .2
ENTER COST PER ORDER PLACED
? 2.
ENTER STOCKOUT COST PER UNIT PER PERIOD
? .75
ENTER BEGINNING INVENTORY IN UNITS
? 15
ENTER PROBABILITY DISTRIBUTION FOR DEMAND
DO YOU NEED INSTRUCTIONS ON ENTERING THE PROBABILITY
DISTRIBUTION
? N
ENTER D PD
? 0 8
ENTER D PD
? 1 13
ENTER D PD
? 2 16
ENTER D PD
? 3 20
ENTER D PD
? 4 17
ENTER D PD
? 5 11
ENTER D PD
? 6 9
ENTER D PD
? 7 6
ENTER D PD
? 0 0
DO YOU WISH TO DISPLAY THE DISTRIBUTION?
YES OR NO Y
--D--|--PD--
  0 |      8
  1 |     13
  2 |     16
  3 |     20
  4 |     17
  5 |     11
  6 |      9
  7 |      6
SHALL WE GO ON?
YES OR NO Y
ENTER PROBABILITY DISTRIBUTION FOR LEAD TIME
DO YOU NEED INSTRUCTIONS ON ENTERING THE PROBABILITY
DISTRIBUTION
? N
ENTER LT P(LT)
? 3 30
ENTER LT P(LT)
? 4 50
```

```

ENTER LT P(LT)
? 5 20
ENTER LT P(LT)
? 0 0
DO YOU WISH TO DISPLAY THE DISTRIBUTION?
YES OR NO Y
  LT--|---PLT---
    3 |    30
    4 |    50
    5 |    20
SHALL WE GO ON?
YES OR NO Y
ENTER INITIAL VALUE FOR Q
? 15
ENTER INITIAL VALUE FOR R
? 13
DO YOU WANT TO EXPERIMENT IN ORDER TO DETERMINE THE
APPROPRIATE RUN LENGTH?
YES OR NO Y
ENTER TEST VALUE FOR RUN LENGTH IN PERIODS
? 20
HOW MANY RUNS OF THAT LENGTH DO YOU WISH TO TRY?
? 15
REPLICATION 1   AVG DAILY COST = 3.67
REPLICATION 2   AVG DAILY COST = 2.65
REPLICATION 3   AVG DAILY COST = 2.75
REPLICATION 4   AVG DAILY COST = 1.8375
REPLICATION 5   AVG DAILY COST = 2.0275
REPLICATION 6   AVG DAILY COST = 2.09
REPLICATION 7   AVG DAILY COST = 2.3
REPLICATION 8   AVG DAILY COST = 2.305
REPLICATION 9   AVG DAILY COST = 2.7725
REPLICATION 10  AVG DAILY COST = 2.6825
REPLICATION 11  AVG DAILY COST = 2.09
REPLICATION 12  AVG DAILY COST = 2.2075
REPLICATION 13  AVG DAILY COST = 2.7925
REPLICATION 14  AVG DAILY COST = 2.6275
REPLICATION 15  AVG DAILY COST = 2.73
SAMPLE MEAN = 2.502166667
SAMPLE STD = 0.4380295525
WOULD YOU LIKE TO SEE A PLOT OF THIS SAMPLING DISTRIBUTION? Y

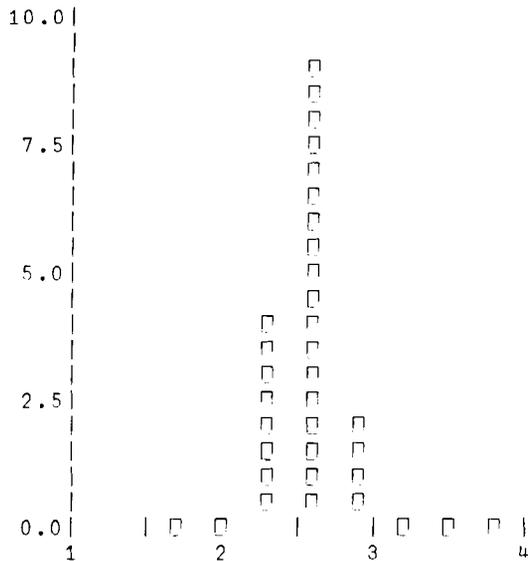
```



DO YOU WISH TO TRY ANOTHER RUN LENGTH? Y
 ENTER TEST VALUE FOR RUN LENGTH IN PERIODS
 ? 50
 HOW MANY RUNS OF THAT LENGTH DO YOU WISH TO TRY?
 ? 15

REPLICATION	AVG DAILY COST
1	2.459
2	2.428
3	2.141
4	2.644
5	2.386
6	2.354
7	2.047
8	2.3
9	2.282
10	2.302
11	2.393
12	2.379
13	2.429
14	2.781
15	2.157

SAMPLE MEAN = 2.365466667
 SAMPLE STD = 0.1786959677
 WOULD YOU LIKE TO SEE A PLOT OF THIS SAMPLING DISTRIBUTION? Y



DO YOU WISH TO TRY ANOTHER RUN LENGTH? N
 ENTER LENGTH OF EACH RUN IN PERIODS

? 40

PARAMETER ENTRY COMPLETE. WOULD YOU LIKE TO REVIEW THE VALUES?

? Y

CO = 2

CC = 0.2

CS = 0.75

BINV = 15

Q= 15

R= 13

RL= 40

DO YOU WANT TO PROCEED WITH THE SIMULATION?

IF SO, ENTER SEARCH, OTHERWISE RESTART PARAMETER SETTING PROCESS
 BY ENTERING SET

PROGRAM DESCRIPTION

```

V DESCRIBE
[1]  ' '
[2]  '          INTERACTIVE INVENTORY SIMULATION'
[3]  ' '
[4]  '   THIS SET OF APL FUNCTIONS ALLOWS THE USER TO SEARCH FOR OPTIMUM'
[5]  'VALUES OF ORDER QUANTITY (Q) AND REORDER POINT (R) IN A STOCHASTIC'
[6]  'INVENTORY SYSTEM. AFTER ENTERING COST COEFFICIENTS, PROBABILITY'
[7]  'DISTRIBUTIONS, AND SIMULATION RUN PARAMETERS, THE USER CONDUCTS AN'
[8]  'INTERACTIVE SEARCH FOR THE OPTIMUM Q R PAIR. THE PRIMARY FUNCTIONS ARE'
[9]  'DESCRIBED BELOW.'
[10] ' '
[11] 'SET'
[12] ' '
[13] '   THIS PROGRAM ALLOWS THE USER TO ENTER THE DATA FOR THE SIMULATION'
[14] 'RUN. SPECIFICALLY IT:'
[15] ' '
[16] '   1. ACCEPTS VALUES FOR UNIT CARRYING COST, UNIT STOCKOUT COST,'
[17] '   ORDER COST, AND BEGINNING INVENTORY.'
[18] '   2. CALLS TWO FUNCTIONS (ENTDDIST AND ENTLDIST) TO ENTER AND SHAPE'
[19] '   THE PROBABILITY DISTRIBUTIONS FOR DEMAND AND LEAD TIME.'
[20] '   3. ALLOWS THE USER TO CALL A SEPARATE FUNCTION (SETRL--SEE BELOW)'
[21] '   TO HELP DETERMINE THE APPROPRIATE LENGTH OF EACH SIMULATION RUN.'
[22] ' '
[23] 'SETRL'
[24] ' '
[25] '   THIS PROGRAM ALLOWS THE USER TO EXPERIMENT WITH SEVERAL VALUES FOR'
[26] 'THE RUN LENGTH OF THE SIMULATION. SPECIFICALLY, IT PERFORMS SEVERAL'
[27] 'RUNS OF A GIVEN LENGTH AND DISPLAYS THE RESULTANT AVERAGE DAILY COST'
[28] 'FOR EACH RUN. THE USER MAY DISPLAY THE COST FOR EACH SET OF'
[29] 'REPLICATIONS AS A FREQUENCY HISTOGRAM.'
[30] ' '
[31] 'SEARCH'
[32] ' '
[33] '   THIS PROGRAM ALLOWS THE USER TO ALTER THE VALUES OF Q AND R AND'
[34] 'PERFORM REPEATED SIMULATION RUNS. THE VALUE FOR AVERAGE DAILY COST IS'
[35] 'DISPLAYED AFTER EACH ITERATION.'
[36] ' '
[37] 'SIM'
[38] ' '
[39] '   THIS PROGRAM IS CALLED BY SETRL AND SEARCH TO PRODUCE THE AVERAGE'
[40] 'DAILY COST FOR A SINGLE SIMULATION RUN. ALL THE COST COEFFICIENTS,'
[41] 'PROBABILITY DISTRIBUTIONS, AND RUN PARAMETERS MUST BE DEFINED IN ADVANCE.'
[42] ' '
[43] 'ENTDDIST AND ENTLDIST'
[44] ' '
[45] '   THESE PROGRAMS ACCEPT VALUES OF AN EMPIRICAL PROBABILITY'
[46] 'DISTRIBUTION FOR DEMAND AND LEAD TIME. SEE ADDITIONAL INSTRUCTIONS BELOW'
V
```

```

▽ SETRL;REPL;COUNT;SM;STD
[1] REPLICATES THE INVENTORY SIMULATION TO DEVELOP
[2] A SAMPLING DISTRIBUTION OF AVERAGE COST. CALLS
[3] H PLOT ON USER OPTION TO DISPLAY HISTOGRAM OF
[4] A SAMPLE MEANS. REQUIRES THE PUBLIC FUNCTIONS AYN
[5] A RAND INP.
[6] FLAG←0
[7] START:'ENTER TEST VALUE FOR RUN LENGTH IN PERIODS'
[8] RL←1 INP '?'
[9] +(RL>250)/ERR1
[10] P1:'HOW MANY RUNS OF THAT LENGTH DO YOU WISH TO TRY?'
[11] REPL←1 INP '?'
[12] +(REPL>20)/ERR2
[13] COUNT←0
[14] SAMPLE←10
[15] LOOP:SIM
[16] COUNT←COUNT+1
[17] 'REPLICATION ':COUNT;' AVG DAILY COST = ';ADC
[18] SAMPLE←SAMPLE,ADC
[19] +(REPL>COUNT)/LOOP
[20] SM←(+/SAMPLE)÷ρSAMPLE
[21] 'SAMPLE MEAN = ';SM
[22] STD←((÷(ρSAMPLE-1))×(+/(SM-SAMPLE)*2))*0.5
[23] 'SAMPLE STD = ';STD
[24] YNT←AYN 'WOULD YOU LIKE TO SEE A PLOT OF THIS SAMPLING DISTRIBUTION?'
[25] +(YNT=0)/P2
[26] H PLOT
[27] P2:YNT←AYN 'DO YOU WISH TO TRY ANOTHER RUN LENGTH?'
[28] +(YNT=1)/START
[29] +0
[30] ERR1:'THAT''S TOO LONG A RUN. CHOOSE A NUMBER <250'
[31] +START
[32] ERR2:'THAT''S TOO MANY REPLICATIONS. 20 OR LESS SHOULD BE SUFFICIENT'
[33] +P1

```

```

▽ SIM;INV;ORDQ;DAYS;LT
[1] THIS FUNCTION IS CALLED BY SEARCH AND SETRL TO GENERATE
[2] A AN AVERAGE DAILY COST FOR A SINGLE SIMULATION RUN OF
[3] A THE INVENTORY SYSTEM.
[4] DAYS←TCO+TCC+TCS←0
[5] INV←BINV
[6] A INITIALIZE ORDER QUEUE
[7] ORDQ←(Γ/LDIST)ρ0
[8] CYCLE:←(RL<DAYS+DAYS+1)/END
[9] A ADD INCOMING ORDER TO INVENTORY
[10] INV←INV+(1+ORDQ)
[11] A SHIFT ORDER QUEUE
[12] ORDQ←(1+ORDQ),0
[13] A SUBTRACT DAILY DEMAND FROM INVENTORY
[14] INV←INV-DDIST[?100]
[15] A PLACE ORDER IF INVENTORY + COMING ORDERS < R
[16] +(R<+/INV,ORDQ)/ON
[17] A PLACE ORDER, UPDATE ORDER QUEUE AND TOTAL ORDER COST
[18] LT←LDIST[?100]
[19] ORDQ[LT]←ORDQ[LT]+Q
[20] TCO←TCO+CO
[21] A UPDATE CARRYING COST OR STOCKOUT COST
[22] ON:→((INV=0),(INV>0))/CYCLE,GT
[23] →CYCLE,TCS←TCS+CS×-INV
[24] GT:→CYCLE,TCC←TCC+CC×INV
[25] A CALCULATE AVERAGE DAILY COST
[26] END:ADC←+/(TCS,TCO,TCC)÷RL

```

```

▽ HPLOT;RANGE
[1] ACCEPTS SAMPLE OF COST MEANS FROM SETRL, SETS A SCALE,
[2] CONSTRUCTS A FREQUENCY DISTRIBUTION, AND CALLS SHIST
[3] TO PRINT A HISTOGRAM. FLAG IS SET TO
[4] USE THE SAME SCALE FOR ALL RUNS OF A
[5] SINGLE SESSION. REQUIRES THE USE OF THE
[6] PUBLIC FUNCTION VS.
[7] →(FLAG=1)/PL
[8] FLAG+1
[9] RANGE+(1/SAMPLE)-1/SAMPLE
[10] MIN+(1/SAMPLE)-0.25×RANGE
[11] CAT+(ρSAMPLE)÷3
[12] →(CAT<6)/FORCE
[13] →(CAT≥15)/FORCE1
[14] SETINC:INC+RANGE÷CAT
[15] SCALE+(INC×1(1.25×CAT))+MIN
[16] PL:S+(MIN,INC,CAT) FR SAMPLE
[17] S+S,((ρSCALE)-ρS)ρ0
[18] 20 40 SHIST S VS SCALE
[19] →0
[20] FORCE:CAT+6
[21] →SETINC
[22] FORCE?:CAT+15
[23] →SETINC

```

▽

```

▽ SEARCH;T
[1] ACCEPTS A VALUE FOR Q AND R, CALLS SIM
[2] TO RETURN THE AVERAGE DAILY COST. USED TO ITERATIVELY
[3] SEARCH FOR THE OPTIMUM VALUES OF Q AND R.
[4] REQUIRES THE PUBLIC FUNCTIONS AYN AND INP
[5] T←10
[6] 'HAVE YOU ENTERED ALL THE NECESSARY DATA?'
[7] YNT+AYN '?'
[8] →(YNT=1)/GO1
[9] 'PLEASE ENTER DATA. TYPE DESCRIBE FOR INSTRUCTIONS'
[10] →0
[11] GO1:'INITIAL VALUES FOR Q AND R ARE;'
[12] ' Q= ';Q;' R= ';R
[13] 'SHALL WE PROCEED?'
[14] YNT+AYN '?'
[15] →(YNT=1)/LOOP
[16] END:'DONE'
[17] →0
[18] LOOP:SIM
[19] 'AVERAGE DAILY COST = ';ADC
[20] ' '
[21] 'ENTER Q R'
[22] T←2 INP '?'
[23] →(1/T=0)/END
[24] Q←1↑T
[25] R←1↑T
[26] →LOOP
[27] →

```

▽

```

▽ SET
[1]  AIT REQUIRES THE FOLLOWING USER-DEFINED FUNCTIONS:
[2]  AENTDDIST, ENTLDIST, SETHOW, SETRL. ALSO REQUIRED ARE THE
[3]  APUBLIC FUNCTIONS AYN AND IPI.
[4]  ATHIS FUNCTION ACCEPTS USER INPUT TO SET VALUES FOR THE SIMULATION
[5]  ARUN.
[6]  'ENTRY OF PARAMETERS FOR INVENTORY SIMULATION'
[7]  'DO YOU NEED INSTRUCTIONS? '
[8]  YNT+AYN 'YES OR NO'
[9]  →(YNT=0)/PROCEED1
[10] SETHOW
[11] PROCEED1: 'ENTER CARRYING COST PER UNIT PER PERIOD'
[12] CC+1 INP '?'
[13] 'ENTER COST PER ORDER PLACED'
[14] CO+1 INP '?'
[15] 'ENTER STOCKOUT COST PER UNIT PER PERIOD'
[16] CS+1 INP '?'
[17] 'ENTER BEGINNING INVENTORY IN UNITS'
[18] BINV+1 INP '?'
[19] 'ENTER PROBABILITY DISTRIBUTION FOR DEMAND'
[20] ENTDDIST
[21] 'ENTER PROBABILITY DISTRIBUTION FOR LEAD TIME'
[22] ENTLDIST
[23] PROCEED2: 'ENTER INITIAL VALUE FOR Q '
[24] Q+1 INP '?'
[25] 'ENTER INITIAL VALUE FOR R '
[26] R+1 INP '?'
[27] 'DO YOU WANT TO EXPERIMENT IN ORDER TO DETERMINE THE '
[28] 'APPROPRIATE RUN LENGTH? '
[29] YNT+AYN 'YES OR NO'
[30] →(YNT=0)/PROCEED3
[31] SETRL
[32] PROCEED3: 'ENTER LENGTH OF EACH RUN IN PERIODS'
[33] RL+1 INP '?'
[34] 'PARAMETER ENTRY COMPLETE. WOULD YOU LIKE TO REVIEW THE VALUES?'
[35] YNT+AYN '?'
[36] →(YNT=0)/PROCEED4
[37] 'CO = ';CO
[38] 'CC = ';CC
[39] 'CS = ';CS
[40] 'BINV = ';BINV
[41] 'Q= ';Q
[42] 'R= ';R
[43] 'RL= ';RL
[44] PROCEED4: 'DO YOU WANT TO PROCEED WITH THE SIMULATION?'
[45] 'IF SO, ENTER SEARCH, OTHERWISE RESTART PARAMETER SETTING PROCESS'
[46] 'BY ENTERING SET'
▽

```

```

▽ SETHOW
[1]  ''
[2]  '          INPUT INSTRUCTIONS'
[3]  ''
[4]  'THE FOLLOWING DATA MUST BE ENTERED FOR THE INVENTORY SIMULATION.'
[5]  ''
[6]  'DATA NAME          DESCRIPTION'
[7]  '-----'
[8]  ''
[9]  '    CO              ORDER COST:  THE COST OF PLACING AN ORDER'
[10] ''
[11] '    CC              CARRYING COST: THE COST OF CARRYING ONE UNIT OF'
[12] '                   INVENTORY FOR ONE DAY'
[13] ''
[14] '    CS              STOCKOUT COST: THE COST OF BACK ORDERING ONE UNIT'
[15] '                   FOR ONE DAY'
[16] ''
[17] '    BINV            BEGINNING INVENTORY: THE NUMBER OF UNITS IN'
[18] '                   INVENTORY AT THE BEGINNING OF EACH SIMULATION RUN'
[19] ''
[20] '    Q               QUANTITY ORDERED: THE NUMBER OF UNITS TO BE'
[21] '                   ORDERED EACH TIME AN ORDER IS PLACED'
[22] ''
[23] '    R               REORDER POINT:  THE STOCK LEVEL AT WHICH ORDER '
[24] '                   PLACEMENT IS TRIGGERED'
[25] ''
[26] '    RL              RUN LENGTH:  THE NUMBER OF PERIODS IN EACH'
[27] '                   SIMULATION RUN'
[28] ''
[29] '    DDIST           THE PROBABILITY DISTRIBUTION FOR DEMAND'
[30] ''
[31] '    LDIST           THE PROBABILITY DISTRIBUTION FOR LEAD TIME'
[32] ''
[33] '    ALL THE DATA CAN BE ENTERED BY TYPING ''SET'' AND ANSWERING THE '
[34] '    QUESTIONS. ENTER ONLY NUMERIC DATA. TO INTERRUPT A RUN, HIT THE '
[35] '    ATTENTION KEY,TYPE THE RIGHT ARROW (+) OR ENTER O BACKSPACE U,'
[36] '    BACKSPACE T, AND START AGAIN.'
[37] ''
[38] '    AS AN ALTERNATIVE TO USING SET, ALL THE DATA ITEMS (EXCEPT THE'
[39] '    PROBABILITY DISTRIBUTIONS) CAN BE ENTERED OR CHANGED BY USING THE APL'
[40] '    ASSIGNMENT OPERATOR (FOR INSTANCE CO+2.25). TO ENTER THE PROBABILITY'
[41] '    DISTRIBUTIONS (WITHOUT USING THE ''SET'' FUNCION) TYPE ENTDDIST AND '
[42] '    ENTLDIST FOR DEMAND AND LEAD TIME DISTRIBUTIONS RESPECTIVELY AND '
[43] '    FOLLOW THE INSTRUCTIONS GIVEN BELOW. AFTER THE DATA AND DISTRIBUTIONS'
[44] '    HAVE BEEN ENTERED, TYPE SETRL TO DETERMINE THE APPROPRIATE RUN LENGTH. '
[45] '    FINALLY, TYPE SEARCH TO INTERACTIVELY SEARCH FOR THE OPTIMUM Q AND R.'

```

▽

```

V DISTHOW
[1] ''
[2] '          INSTRUCTIONS FOR ENTERING PROBABILITY DISTRIBUTIONS'
[3] ''
[4] '          ENTDDIST AND ENTLDIST REQUIRE ENTRY OF AN EMPRICAL PROBABILITY'
[5] 'DISTRIBUTION OF THE FOLLOWING FORM.'

```

V

```

V ENTDDIST;IN;T
[1] RREQUIRES THE PUBLIC FUNCTIONS AYN, ΔFMT,
[2] RAND INP.
[3] RACCEPTS PROBABILITY DISTRIBUTION FOR DEMAND AND
[4] RCONSTRUCTS A 100 ELEMENT VECTOR FOR RANDOMLY
[5] RGENERATING DEMAND IN SIMULATION.
[6] 'DO YOU NEED INSTRUCTIONS ON ENTERING THE PROBABILITY'
[7] 'DISTRIBUTION'
[8] YNT+AYN '?'
[9] +((YNT=1),YNT=0)/OUT,START
[10] OUT:DISTHOW
[11] START:DDIST+10
[12] IN+10
[13] LOOP:'ENTER D PD'
[14] T+2 INP '?'
[15] +(^/T=0)/TEST
[16] IN+IN,T
[17] DDIST+DDIST,(1+T)ρ1+T
[18] +LOOP
[19] TEST:+((ρDDIST)≠100)/ERR
[20] 'DO YOU WISH TO DISPLAY THE DISTRIBUTION? '
[21] YNT+AYN 'YES OR NO'
[22] +(YNT=0)/0
[23] IN+(((ρIN)÷2),2)ρIN
[24] '  D | PD '
[25] 'I4,0 | 0,I5' ΔFMT IN
[26] 'SHALL WE GO ON?'
[27] YNT+AYN 'YES OR NO'
[28] +(YNT=1)/0
[29] 'OK, START OVER'
[30] +START
[31] ERR:'PROBABILITY PERCENTAGES DO NOT ADD TO 100, START OVER'
[32] +START

```

V

```

▽ ENTLDIST; IN; T
[1]  REQUIRES THE PUBLIC FUNCTIONS AYN, ΔFMT,
[2]  RAND INP.
[3]  ACCEPTS PROBABILITY DISTRIBUTION FOR LOAD TIME AND
[4]  CONSTRUCTS A 100 ELEMENT VECTOR FOR RANDOMLY
[5]  GENERATING LOAD TIME IN THE SIMULATION.
[6]  'DO YOU NEED INSTRUCTIONS ON ENTERING THE PROBABILITY'
[7]  'DISTRIBUTION'
[8]  YNT←AYN '?'
[9]  →((YNT=1),YNT=0)/OUT,START
[10] OUT:DISTHOW
[11] START:LDIST←10
[12] IN←10
[13] LOOP:'ENTER LT P(LT)'
[14] T←2 INP '?'
[15] →(Λ/T=0)/TEST
[16] IN←IN,T
[17] LDIST←LDIST,(-11+T)ρ1+T
[18] →LOOP
[19] TEST:→((ρLDIST)≠100)/ERR
[20] 'DO YOU WISH TO DISPLAY THE DISTRIBUTION? '
[21] YNT←AYN 'YES OR NO'
[22] →(YNT=0)/0
[23] IN←(((ρIN)÷2),2)ρIN
[24] ' LT | PLT '
[25] ' I4, 0 | 0, I5' ΔFMT IN
[26] 'SHALL WE GO ON?'
[27] YNT←AYN 'YES OR NO'
[28] →(YNT=1)/0
[29] 'OK, START OVER'
[30] →START
[31] ERR:'PROBABILITY PERCENTAGES DO NOT ADD TO 100, START OVER'
[32] →START

```

▽

FINANCIAL RATIO CALCULATIONS

Wilma D. Stricklin
 Professor of Management
 College of Business Administration
 Northern Arizona University
 Flagstaff, Arizona

DESCRIPTION

This program performs a number of ratio calculations which are typically used to gauge the financial effectiveness of an organization. These ratios are:

$\frac{\text{Net Income}}{\text{Total Assets}}$	-	Return on investment as a function of assets
$\frac{\text{Net Income}}{\text{Total Common Equity}}$	-	Return on investment as a function of equity
$\frac{\text{Net Income}}{\text{Number of Shares}}$	-	Earnings per share
$\frac{\text{Net Income}}{\text{Total Sales}}$	-	Profits as a function of sales
$\frac{\text{Long Term Debt}}{\text{Total Assets}}$	-	Long term debt as a function of assets
$\frac{\text{Total Sales}}{\text{Total Assets}}$	-	Sales as a function of assets

The program is initiated by typing *EFFRATIO* and then following the instructions given. Any number of years' worth of data can be entered; but, beyond five, the data inputting can be tedious and error prone.

In the sample shown in the following section, the data were taken from the Boise-Cascade Case found in Robert L. Katz, Cases and Concepts in Corporate Strategy, Englewood Cliffs, N. J.: Prentice-Hall, 1970.

PROGRAM USAGE

EFFRATIO

OBSERVE THE FOLLOWING INSTRUCTIONS IN ENTERING DATA:

- A. ROUND YOUR DATA TO THE NEAREST THOUSAND DOLLARS,
- B. ENTER 0(ZERO) IN EACH INSTANCE WHERE DATA IS UNKNOWN OR UNAVAILABLE,
- C. USE THE - SIGN (OVER THE 2-KEY) TO MARK NEGATIVE NUMBERS,
- D. DO NOT USE COMMAS TO SEPARATE NUMBERS.

ENTER YEARS FOR WHICH YOU HAVE DATA: 1962 1963 1964 1965 1966

ENTER TOTAL ASSETS FOR EACH YEAR:

: 124382 130456 293475 370522 454250

ENTER NET INCOME AFTER TAXES FOR EACH YEAR:

: 5055 7390 14947 17888 17014

ENTER TOTAL LONG TERM DEBT FOR EACH YEAR

: 71548 63815 117127 166921 193909

ENTER TOTAL COMMON EQUITY FOR EACH YEAR:

: 52834 64663

LENGTH ERROR...REENTER 5 VALUES.

: 52834 64663 166496 172251 192751

ENTER TOTAL SHARES OF COMMON STOCK FOR EACH YEAR:

: 3741 4001 4387 4376 9005

ENTER TOTAL SALES FOR EACH YEAR:

: 176015 219361 363581 422911 492466

ENTER HEADING FOR YOUR REPORT: BOISE CASCADE CORP.

EFFECTIVENESS RATIOS---BOISE CASCADE CORP.

YEAR	ROI-A	ROI-EQ	EPS	PROF/SLS	DT/ASSET	SLS/ASSET
1962	0.0406	0.0957	1.3512	0.0287	0.5752	1.4151
1963	0.0566	0.1143	1.8470	0.0337	0.4892	1.6815
1964	0.0509	0.0898	3.4071	0.0411	0.3991	1.2389
1965	0.0483	0.1038	4.0878	0.0423	0.4505	1.1414
1966	0.0375	0.0883	1.8894	0.0345	0.4269	1.0841

DO YOU WISH TO CONTINUE? NO

PROGRAM DESCRIPTION

```

▽ EFFRATIO;N;M;YRS;TA;NP;LTD;CE;CMS;SA;CO
[1]  A THE FOLLOWING PUBLIC FUNCTIONS ARE REQUIRED IN ORDER
[2]  A TO USE THIS PROGRAM: AKI, CENTER, COLNAMES, ΔFMT, INP,
[3]  A AND IPI.
[4]  A THE FOLLOWING LINES GIVE THE USER INSTRUCTIONS ON INPUT RESTRICTIONS.
[5]  A //////////////////////////////////
[6]  'OBSERVE THE FOLLOWING INSTRUCTIONS IN ENTERING DATA:'
[7]  '   A. ROUND YOUR DATA TO THE NEAREST THOUSAND DOLLARS,'
[8]  '   B. ENTER 0(ZERO) IN EACH INSTANCE WHERE DATA IS UNKNOWN OR UNAVAILABLE,'
[9]  '   C. USE THE - SIGN (OVER THE 2-KEY) TO MARK NEGATIVE NUMBERS,'
[10] '   D. DO NOT USE COMMAS TO SEPARATE NUMBERS.'
[11] ' '
[12] ' '
[13] TOP;N+0YRS+IPI 'ENTER YEARS FOR WHICH YOU HAVE DATA:'
[14] A THIS SECTION ACCEPT INPUT, TESTS EACH SET FOR ACCURACY OF NUMBER OF INPUTS, AND CYCLES FOR CORRECT DATA.
[15] 'ENTER TOTAL ASSETS FOR EACH YEAR:'
[16] TA+INP N
[17] 'ENTER NET INCOME AFTER TAXES FOR EACH YEAR:'
[18] NI+INP N
[19] 'ENTER TOTAL LONG TERM DEBT FOR EACH YEAR'
[20] LTD+INP N
[21] 'ENTER TOTAL COMMON EQUITY FOR EACH YEAR:'
[22] CE+INP N
[23] 'ENTER TOTAL SHARES OF COMMON STOCK FOR EACH YEAR:'
[24] CMS+INP N
[25] 'ENTER TOTAL SALES FOR EACH YEAR:'
[26] SA+INP N
[27] A THIS CREATES A MATRIX OF RATIOS RESULTING FROM A DIVISION OF VECTORS.
[28] CO+(6,N)0(NI,NI,NI,NI,LTD,SA)÷(TA,CE,CMS,SA,TA,TA)
[29] A THIS ESTABLISHES REPORT FORMAT.
[30] TITLE+AKI 'ENTER HEADING FOR YOUR REPORT:'
[31] FORMAT+'I5,6F10.4'
[32] FORMAT CENTER 'EFFECTIVENESS RATIOS---',TITLE
[33] FORMAT COLNAMES 'XYEARXROI-AXROI-EQXEPSXPROF/SLSXDT/ASSETXSLS/ASSETX'
[34] FORMAT ΔFMT(YRS;0CO)
[35] A THIS PERMITS THE USER TO REPEAT THE PROGRAM FOR ANOTHER SET OF DATA, FOR A COMPETITOR, FOR THE INDUSTRY, ETC.
[36] →(AYN 'DO YOU WISH TO CONTINUE?')/TOP

```

▽

COMPUTATION OF A PRICE INDEX
BY THE METHOD OF THE WEIGHTED GEOMETRIC MEAN

Bernard A. Thielges
Assistant Professor
Management Department
College of Business Administration
Marquette University
Milwaukee, Wisconsin

DESCRIPTION

Statement of the Problem

The computation of a price index can be done by various averaging techniques. The arithmetic mean, however, is the average most often used because of its simplicity and ease in calculation. Various weighting methods can also be used with the arithmetic mean, but the one most often used is the method which uses the base year values as the weights.

The method of the arithmetic mean together with base year values as weights provides a good index. There are deficiencies, however. It is known that there is an upward bias to the arithmetic mean. On the other hand, the use of base year values as weights provides a downward bias by giving too much weight to those items having the smallest change in price. At times, one bias may tend to offset the other; but in other instances, this will not be the case.

In order to compute a price index which is consistently more accurate, the geometric mean, which is an unbiased mean, might be used. And, to provide for a more appropriate weighting system, it is suggested that the average of the values computed for all years, from the base year to the current year, be used. The computation of the price index by this method is considerably longer and more sophisticated. However, the initial computation and subsequent

updating are readily facilitated through the use of the computer.

Program Description

The program *PINDEX* computes a price index for selected items. The index is computed for the final year of data submitted, using the first year as the base year. The formula used in the computation is

$$\text{Price Index} = \frac{\sum w \left(\log \frac{p_1}{p_0} \right)}{\sum w}$$

$$= \frac{\sum_{i=1}^{i=N} p_i q_i}{N}$$

where, w (Average value of all years for each item in the index)

p_i = price for the i th year of data

q_i = quantity for the i th year of data

p_1 = price for the last year of data submitted

p_0 = price for the first (base) year of data submitted

N = number of years for which data is submitted.

To begin the computation, the data must be entered in the following fashion. First, the years for which data are to be submitted are entered, i.e., 1971 1972 1973, and 1974. These data on years are needed for labeling the table in the print-out. The number of items to be included in the index is entered as a number followed by the individual item names when asked for by the program. Item names should not exceed 16 characters in length, should be only one word (compound words can be hyphenated or run together), and should be separated by blanks (do not use commas).

Quantity data is then entered by year for all items, followed by price data by year for all items. These entries constitute the quantity and price matrices. Data are entered for all items by year and appear as shown in the sample. The program puts the data in proper matrix form for calculations.

Internal computation of the index starts with the calculation of the ratio between the final year price data and the first year price data. The first year of data is therefore the base year. The logarithm to the base 10 of this ratio is then determined.

The weight to be used is calculated by first multiplying the price matrix by the quantity matrix and obtaining a value matrix. This matrix lists the value of each individual item in the index by years. The average value for each specific item is calculated by summing each year's value and then dividing by the number of years. The resultant average is used as the weight.

Next, the logarithm of the price ratio for each item is multiplied by its weight. This weighted value is summed for all items and divided by the sum of the weights, i.e., the sum of the average values for all of the years for all of the items. The final computation of the price index is made by finding the antilog of the weighted data.

The print-out of data includes a repeat of the price and quantity matrix as well as the final price index as a percent. The user should be aware that if data for many years are submitted or if the data are fairly large, then the available space might not be sufficient to print out the price and quantity matrices. In this case, an asterisk will appear wherever there is insufficient space in the print-out.

PROGRAM USAGE

PINDEX

ENTER THE YEARS FOR WHICH THE INDEX IS TO BE COMPUTED, EG, 1973 1974 1975

□:

1973 1974 1975

ENTER THE NUMBER OF ITEMS TO BE INCLUDED IN THE PRICE INDEX, EG, 6

□:

6

ENTER THE NAMES OF THE ITEMS TO BE INCLUDED IN THE PRICE INDEX, EG, LUMBER CEMENT NAILS ASPHALTSHINGLES. NOTE THAT ANY ITEM CONTAINING TWO OR MORE WORDS IS TO BE JOINED TOGETHER TO MAKE ONE WORD. NO ITEM SHOULD EXCEED SIXTEEN CHARACTERS IN LENGTH.

CORN WHEAT OATS BARLEY RYE RICE

ENTER DATA FOR THE PRODUCTION MATRIX, EG, QUANTITIES OF EACH ITEM SOLD, FOR 1973

□:

4084 1316 927 392 33 76

ENTER DATA FOR THE PRODUCTION MATRIX, EG, QUANTITIES OF EACH ITEM SOLD, FOR 1974

□:

4760 1522 789 373 24 89

ENTER DATA FOR THE PRODUCTION MATRIX, EG, QUANTITIES OF EACH ITEM SOLD, FOR 1975

□:

4375 1570 930

INCORRECT NUMBER OF ITEMS. TRY AGAIN

□:

4375 1570 930 418 23 105

ENTER DATA FOR THE PRICE MATRIX, EG, PRICES OF EACH ITEM, FOR 1973

□:

1.16 1.35 .62 1.03 .97 4.93

ENTER DATA FOR THE PRICE MATRIX, EG, PRICES OF EACH ITEM, FOR 1974

□:

1.04 1.39 .66 1.00 1.07 4.97

ENTER DATA FOR THE PRICE MATRIX, EG, PRICES OF EACH ITEM, FOR 1975

□:

1.05 1.22 .59 .88 .99 5

PRODUCTION DATA

ITEM	1973	1974	1975
CORN	4,084.00	4,760.00	4,375.00
WHEAT	1,316.00	1,522.00	1,570.00
OATS	927.00	789.00	930.00
BARLEY	392.00	373.00	418.00
RYE	33.00	24.00	23.00
RICE	76.00	89.00	105.00

PRICE DATA

ITEM	1973	1974	1975
CORN	1.16	1.04	1.05
WHEAT	1.35	1.39	1.22
OATS	0.62	0.66	0.59
BARLEY	1.03	1.00	0.88
RYE	0.97	1.07	0.99
RICE	4.93	4.97	5.00

VALUE DATA

ITEM	1973	1974	1975
CORN	4,737.44	4,950.40	4,593.75
WHEAT	1,776.60	2,115.58	1,915.40
OATS	574.74	520.74	548.70
BARLEY	403.76	373.00	367.84
RYE	32.01	25.68	22.77
RICE	374.68	442.33	525.00

PRICE INDEX (PERCENT) IS: 91.14895115

PROGRAM DESCRIPTION

```

V PINDEX;R
[1]  PREPARED BY:
[2]  BERNARD A. THIELGES
[3]  ASSISTANT PROFESSOR OF MANAGEMENT
[4]  COLLEGE OF BUSINESS ADMINISTRATION
[5]  MARQUETTE UNIVERSITY
[6]  MILWAUKEE, WISCONSIN 53233
[7]  THIS PROGRAM COMPUTES A PRICE INDEX BY THE METHOD OF THE WEIGHTED
[8]  GEOMETRIC MEAN.
[9]  INPUT: PRICE AND QUANTITY DATA FOR 'N' YEARS. NOTE THAT BOTH ITEMS
[10] ARE ENTERED AS ROW VECTORS. ITEM NAMES ARE TRANSPOSED AS NECESSARY INTO
    A
[11] COLUMN MATRIX FOR DATA PRINTOUT.
[12] THE PRICE RATIOS FOR ALL ITEMS INCLUDED IN DATA ARE COMPUTED WITH THE
[13] LAST YEAR OF DATA REFERENCED TO THE FIRST YEAR OF DATA AS BASE YEAR.
[14] THE PRICE RATIOS ARE WEIGHTED BY THE AVERAGE OF ALL OF THE YEARS VALUE
[15] OF THE ITEM FROM THE FIRST YEAR THROUGH THE LAST YEAR OF THE DATA
[16] SUBMITTED. THE GEOMETRIC MEAN OF THE WEIGHTED PRICE RATIOS IS THEN
[17] COMPUTED. THIS WEIGHTED PRICE RATIO IS THEN DIVIDED BY THE SUM OF THE
[18] WEIGHTS (TOTAL OF THE AVERAGE VALUES) TO PRODUCE THE PRICE INDEX.
[19] THIS PROGRAM REQUIRES THE USE OF ΔFMT, LIST, AND MIN
[20] CTR←1
[21] 'ENTER THE YEARS FOR WHICH THE INDEX IS TO BE COMPUTED, EG, 1973 1974 19
    75 '
[22] N←ρNN+□
[23] 'ENTER THE NUMBER OF ITEMS TO BE INCLUDED IN THE PRICE INDEX, EG, 6'
[24] M←□
[25] PRICE←QUAN←(M,N)ρ0
[26] 'ENTER THE NAMES OF THE ITEMS TO BE INCLUDED IN THE PRICE INDEX, EG, '
[27] 'LUMBER CEMENT NAILS ASPHALTSHINGLES. NOTE THAT ANY ITEM CONTAINING'
[28] 'TWO OR MORE WORDS IS TO BE JOINED TOGETHER TO MAKE ONE WORD. NO ITEM'
[29] 'SHOULD EXCEED SIXTEEN CHARACTERS IN LENGTH.'
[30] LIST←MIM ' ',□ ρCHANGES A ROW OF WORDS INTO A COLUMN MATRIX OF THE WORDS.
[31] LIST←LIST,((1+ρLIST),(16-1+ρLIST))ρ' ' ρTAKES LITERAL MATRIX CONTAINING I
    TEM NAMES
[32] ρAND PADS WITH BLANKS TO PRODUCE A LITERAL MATRIX OF THE FOLLOWING FORM: L
    IST[N;16].
[33] L1:'ENTER DATA FOR THE PRODUCTION MATRIX, EG, QUANTITIES OF EACH ITEM SOLD
    , FOR ';NN[CTR]
[34] L11:+(M=ρVV+,□)/AGAIN
[35] □+A+'INCORRECT NUMBER OF ITEMS. TRY AGAIN'
[36] →L11
[37] AGAIN:QUAN[;CTR]+((ρVV),1)ρVV
[38] CTR←CTR+1
[39] →L1×1(CTR)≤N
[40] CTR←1
[41] L2:'ENTER DATA FOR THE PRICE MATRIX,EG, PRICES OF EACH ITEM, FOR ' ;NN[CTR
    ]
[42] L111:+(M=ρVV+,□)/AGAIN1
[43] A
[44] →L111
[45] AGAIN1:PRICE[;CTR]+((ρVV),1)ρVV
[46] CTR←CTR+1
[47] →L2×1(CTR)≤N
[48] VALUE←QUAN×PRICE
[49] LOGRAT←(M,1)ρ10⊙(,PRICE[;N])÷PRICE[;1]
[50] AVEVAL←(+/VALUE)÷N

```

```

[51] WTRATIO++/(AVEVAL*.LOGRAT)
[52] LOGPINDEX+WTRATIO++/AVEVAL
[53] R+(10*LOGPINDEX)*100
[54] 'PRODUCTION DATA'
[55] '4A1,X12,10I12' ΔFMT((1 4 ρ 'ITEM'); (1,ρNN) ρNN)
[56] ''
[57] '16A1,10CF12.2' ΔFMT(LIST;QUAN)
[58] ''
[59] ''
[60] 'PRICE DATA'
[61] '4A1,X12,10I12' ΔFMT((1 4 ρ 'ITEM'); (1,ρNN) ρNN)
[62] ''
[63] '16A1,10CF12.2' ΔFMT(LIST;PRICE)
[64] ''
[65] ''
[66] 'VALUE DATA'
[67] '4A1,X12,10I12' ΔFMT((1 4 ρ 'ITEM'); (1,ρNN) ρNN)
[68] ''
[69] '16A1,10CF12.2' ΔFMT(LIST;VALUE)
[70] ''
[71] ''
[72] ''
[73] 'PRICE INDEX (PERCENT) IS: ' ;R

```

▽

4.0

UCLA APL LIBRARY FUNCTIONS

--> 1 PLOTFORMAT PLOT <--

SYNTAX: A PLOT B

THE FUNCTION PLOT WILL GRAPH ONE OR MORE FUNCTIONS SIMULTANEOUSLY, AUTOMATICALLY SCALING THE VALUES TO FIT APPROXIMATELY WITHIN SCALE DIMENSIONS SPECIFIED BY THE USER. IT WILL WORK ONLY IN 1-ORIGIN INDEXING.

THE FORM IN WHICH PLOT IS USED IS:

SCALESIZE PLOT FUNCTION

LEFT ARGUMENT: ONE OR TWO NUMBERS.

SCALESIZE[1]: HEIGHT OF PLOT ON PAPER IN LINES

SCALESIZE[2]: WIDTH OF PLOT ON PAPER IN CHARACTERS

NOTE: IF NO SCALESIZE[2] IS GIVEN, SCALESIZE[1] IS RE-USED.

THERE IS NO BUILT-IN LIMIT TO THE DIMENSIONS, AND A HORIZONTAL AXIS LARGER THAN THE WORKSPACE WIDTH WILL CAUSE SOME POINTS TO BE PRINTED ON THE NEXT LOWER LINE.

RIGHT ARGUMENT: A RECTANGULAR ARRAY WITH RANK ≤ 3 .

SCALAR: WILL BE TREATED AS A VECTOR OF LENGTH ONE.

VECTOR: WILL BE PLOTTED AS ORDINATE AGAINST ITS OWN INDICES AS ABSCISSA.

MATRIX: THE LEFTMOST COLUMN WILL BE TAKEN AS THE ABSCISSA AND ALL OTHER COLUMNS WILL BE PLOTTED AS ORDINATES. A DIFFERENT PLOTTING SYMBOL UP TO THE NUMBER OF SYMBOLS AVAILABLE WILL BE USED FOR EACH COLUMN. IN CASE TWO ORDINATES HAVE A COMMON POINT, THE SYMBOL FOR THE COLUMN FURTHEST TO THE RIGHT WILL BE USED.

3-DIMENSIONAL ARRAY: THE FIRST PLANE WILL BE PLOTTED AS A MATRIX, AND ALL OTHER PLANES WILL BE DISREGARDED.

AUXILIARY FUNCTIONS: THE FUNCTIONS AND AND VS CAN BE USED TO GENERATE THE RIGHT ARGUMENT IN THE PROPER FORM FOR PLOT. FOR EXAMPLE:

20 PLOT Z AND Y VS X

PLOT CHARACTERS: THE SYMBOLS USED ARE ASSIGNED TO THE VARIABLE PC IN LINE 1 OF PLOT. THE ALPHABET SUPPLIED IS 'O*oVΔ□'. THIS ALPHABET MAY BE EXTENDED AND MODIFIED AS DESIRED, USING THE NORMAL FUNCTION-EDITING PROCEDURES: EITHER CHANGE LINE 1 OF THE FUNCTION, OR DELETE IT AND INDEPENDENTLY SPECIFY A VALUE FOR PC.

HISTOGRAMS: PLOT CAN BE USED TO GENERATE HISTOGRAMS BY SETTING THE VARIABLE HS TO 1 IN LINE 2 OF THE FUNCTION. ALTERNATIVELY, LINE 2 CAN BE DELETED, AND HS CAN BE SET EXTERNALLY.

```

▽ A PLOT B;C;D;F;G;H;I;J;L;T;Y;HZ;NB;VT;PT;ST;ISV;U
[1] PC+'o*o▽Δ□'
[2] HS+0
[3] ST+ 1 2 5
[4] SM+ 5 10
[5] *(0=x/(2ρA),ρB), 3 2 1 <ρρB)/0,PRKERR,PLOTL2,PLOTL3
[6] +PLOTL3×ρρB+Q(2,D)ρ(1D+ρ,B),B
[7] PLOTL2:B+B[1;];]
[8] PLOTL3:Y+1+i(ρB)[2]-1
[9] C+(([/[/B[;Y])-L/L/B[;Y]),([/B[;1])-L/B[;1]
[10] F+[(2ρA)÷C+(C=0)×B[1; 2 1]+B[1; 2 1]=0
[11] F+(ST+[/(L0.0001+F×10*-G)°.>ST])×10*G+[10*F
[12] G+(SM÷F)×L((L/L/B[;Y]),L/B[;1])×F÷SM+16[SM[ 1 4
[13] B[;1]+[0.5+F[2]×B[;1]-G[2]
[14] B[;Y]+[0.5+F[1]×B[;Y]-G[1]
[15] H+SM×[([/[/B[;Y]),[/B[;1])÷SM
[16] NB+G[1]+(SM[1]÷F[1])×0,1H[1]÷SM[1]
[17] HZ+G[2]+(SM[2]÷F[2])×0,1H[2]÷SM[2]
[18] 0ρST+6ρ~ISV+1zU+9
[19] PLOTL4:VT+v/0>NB+NB×10*U-ST[6-ISV]+I+1+[L 10*[(NB≠0)/NB
[20] PT+[1+10|PT-1|PT+1E-5+(|NB)°.>10*~1+φ1U
[21] L+U+1-(φ((C+ρNB)ρ1)∧.=PT)10
[22] *(U>T+VT+[I,(L+LzI),(I≠0)×2+L-I),ST[2-ISV]+ST[2-ISV]√L≥U-VT+L>I)/
3 2 +I26
[23] *(~1+I26)×ρρST[4-ISV]+I+1
[24] +PLOTL4×ρρNB+SM[1+~ISV]×~1+1C
[25] PT+(-VT+0[1-I])φPT
[26] PT+(,PT)×J+,φ(φρPT)ρ(,φ(1zPT)√.∧(1U)°.>1VT[I-1),(C×U+1-I+VT+I[I≠0]ρ1
[27] *(2+I26)×1~VT
[28] PT[(U+/(C,U)ρJ)+U×~1+1C]+11×NB<0
[29] PT+(~(1U+J)ε(I+J),1~1+J+U-T)\(1 0 +C,U)ρPT,Uρ0
[30] PT[1C;I+J]+12
[31] PT+' 0123456789~.'[1+PT[;1U-1]]
[32] +PLOTL13×1~ISV
[33] L+1,H[2]ρ0×C+H[1]
[34] PLOTL8:L+(L×HS×Cz0)[1,H[2]ρ0
[35] L[1+(Dz0)/E[;1]]+(Dz0)/D+(C=B[;Y])[.×Y
[36] *(Cz0)/PLOTL11
[37] L+L[0=(SM[2]÷2)|0,1H[2]
[38] PLOTL11:PT((ρPT)[1],1+C÷SM[1])[1+0=SM[1]|C];],(' |',(ρY)ρPC)[1+L]
[39] *(0≤C+C-1)ρPLOTL8
[40] *(U=U+SM[2]-~ISV+~ρρNB+,HZ)ρPLOTL4
[41] PLOTL13:(SM[2]-9)φ(,0 0 ,(U-1)ρ1)\PT),' '
[42] *(ST[1 3 2 4],1)/ 1 3 5 7 19 +I26
[43] ('ORIGIN AND SCALE FACTOR FOR ORDINATE: ';G[1],÷F[1])
[44] *(2+I26)×10=ST[3]
[45] ('SCALE FACTOR FOR ORDINATE: ';10*ST[5]-1)
[46] *(2+I26)×10=ST[2]
[47] ('ORIGIN AND SCALE FACTOR FOR ABSCISSA: ';G[2],÷F[2])
[48] *0×10=ST[4]
[49] ('SCALE FACTOR FOR ABSCISSA: ';10*ST[6]-1)
[50] +0
[51] PRKERR:'THE RIGHT ARGUMENT OF PLOT MUST HAVE RANK ≤ 3.'
▽

```

--> 1 PLOTFORMAT VS <--

SYNTAX: Z+A VS B

THE <VS> FUNCTION IS ESSENTIALLY A COLUMN-CATENATOR, SIMILAR TO AND, EXCEPT THAT THE RIGHT-HAND ARGUMENT MUST BE OF RANK ≤ 1 . IT IS DESIGNED PRIMARILY TO PROVIDE CONVENIENT FORMATION OF INPUT TO PLOT FUNCTION. WHETHER USED BY ITSELF OR WITH AND, VS WILL CAUSE ITS RIGHT ARGUMENT TO APPEAR AS THE LEFTMOST COLUMN OF THE RESULTING ARRAY. (THE RESULTANT WILL BE AN ARRAY OF RANK THREE, CONSISTING OF A SINGLE PLANE).

```
∇ M←A VS B;C;D
[1] +((ρρB+,B)<ρρB), 2 1 0 <ρρA)/ 8 8 4 3
[2] A+((ρB),1)ρA
[3] A+((×/ρA),1)ρA
[4] +(^(ρB)≠1,1ρρA)/9
[5] M←(0,(1ρφρA)ρ1)\A
[6] M[;1]←B
[7] +0×ρρM+(1,ρM)ρM
[8] +0=ρ[]←'AN ARGUMENT OF VS IS OF IMPROPER RANK.'
[9] 'ARGUMENTS OF VS ARE NOT CONFORMABLE.'
```

∇

--> 1 PLOTFORMAT AND <--

SYNTAX: Z+A AND B

THE <AND> FUNCTION IS ESSENTIALLY A COLUMN-CATENATOR, WITH SOME EXTRA EFFECTS WHEN THE ARGUMENTS ARE NOT MATRICES. THE FUNCTION IS DESIGNED TO BE USED EITHER INDEPENDENTLY, OR IN CONJUNCTION WITH VS. IT PROVIDES A CONVENIENT WAY OF FORMING INPUT TO DFT AND EFT.

```
∇AND[[]]∇
∇ L←A AND B;C;D
[1] +(((2<ρρA)∨3<ρρB),0≠ρρB)/ 17 3
[2] B+,B
[3] +(((3=ρρB)∧1≠1ρρB),2=ρρA)/ 17 7
[4] A+,A
[5] +(^(ρA)≠1,D),1≠D+1ρ-2φρB)/16
[6] A+(((D×ρA)LD[ρA],1)ρA
[7] +(1≠ρρB)/9
[8] B+(((ρB)[(1=ρB)×1ρρA],1)ρB
[9] +(^(D≠1,1ρρA),1≠D+1ρ-2φρB)/ 16 11
[10] B+(((3=ρρB)ρ1),(1ρρA),1ρφρB)ρB
[11] +(3=ρρB)/14
[12] L+(((C+1ρφρA)ρ0),(1ρφρB)ρ1)\B
[13] +0×ρρL[;1C]←A
[14] L+(1,((C+1ρφρA)ρ0),(-1+1ρφρB)ρ1)\B
[15] +0×ρρL[;1+1C]←A
[16] +0=ρ[]←'ARGUMENTS OF AND ARE NOT CONFORMABLE.'
[17] 'AN ARGUMENT OF AND IS OF IMPROPER RANK.'
```

∇

--> 1 FORMAT ΔFMT <--

SYNTAX: FS ΔFMT (V1;V2;V3;VN)

AUTHOR: STSC

REFERENCE: ΔFMT - REPORT FORMATTING IN APL PLUS , STSC

ΔFMT GIVES THE APL PLUS USER A SIMPLE, ECONOMICAL, AND ADAPTABLE MEANS OF PRODUCING REPORTS INVOLVING TABULAR FORMATTING OF NUMERIC AND CHARACTER DATA. TABULAR OUTPUT PRODUCED USING ΔFMT TAKES ABOUT ONE-TWENTIETH THE CPU TIME REQUIRED USING OTHER METHODS.

ΔFMT EMPLOYS ITS LEFT ARGUMENT TO CONTROL PRINTING OF ITS RIGHT IN ONE OR MORE COLUMNS.

RIGHT ARGUMENT

V1, V2, V3, ..., AND VN ARE APL VARIABLES OR EXPRESSIONS. THEY MAY BE ANY COMBINATION OF CHARACTER OR NUMERIC SCALARS, VECTORS OR MATRICES. A MATRIX IS PRINTED IN ONE OR MORE COLUMNS; A VECTOR OR SCALAR IS TREATED AS THOUGH IT WERE A SINGLE COLUMN MATRIX.

LEFT ARGUMENT

FS IS A CHARACTER VECTOR MADE UP OF ONE OR MORE FORMAT PHRASES SEPARATED BY COMMAS. EACH PHRASE CONTROLS THE PRINTING OF ONE OR MORE COLUMNS. THE ALLOWABLE FORMAT PHRASES ARE:

<u>MAW</u>	CHARACTER EDITING
<u>MIW</u> OR <u>MQIW</u>	INTEGER EDITING
<u>MF_W.D</u> OR <u>MQF_W.D</u>	FIXED-POINT NUMERIC EDITING
<u>ME_W.S</u>	FLOATING-POINT NUMERIC EDITING
<u>MX_W</u>	INSERTION OF BLANKS
<u>MTEXT</u>	INSERTION OF TEXT

HERE M, W, D, AND S REPRESENT INTEGER CONSTANTS. Q REPRESENTS OPTIONAL QUALIFIERS AND DECORATIONS.

M, WHICH IS OPTIONAL, INDICATES HOW MANY TIMES A PHRASE IS TO APPLY. IF M IS NOT PRESENT, IT IS TAKEN TO BE 1. THUS, 3A1 IS EQUIVALENT TO A1,A1,A1 AND 2I4 IS EQUIVALENT TO I4,I4.

W IS THE FIELD WIDTH, INDICATING THE NUMBER OF PRINT POSITIONS OCCUPIED BY THE EDITED VALUE (FOR A, I, F, AND E) OR BY INSERTED BLANKS (FOR X).

FOR FIXED-POINT EDITING, D IS THE NUMBER OF DIGITS TO APPEAR TO THE RIGHT OF THE DECIMAL POINT. D MUST BE LESS THAN W. FOR FLOATING-POINT EDITING, S IS THE NUMBER OF SIGNIFICANT DIGITS TO BE PRINTED. S MUST BE LESS THAN W-4.

THE A PHRASE IS USED ONLY TO EDIT CHARACTER VALUES WHILE I, F, AND E PHRASES ARE USED ONLY TO EDIT NUMERIC VALUES.

CHARACTERS ENCLOSED BETWEEN `␣` SYMBOLS ARE INSERTED DIRECTLY INTO THE EDITED LINE. THE INSERTED TEXT IS PRINTED EXACTLY AS IT APPEARS (INCLUDING SPACES) BETWEEN THE `␣` SYMBOLS.

QUALIFIER CODES

- B LEAVES THE RESULT FIELD BLANK IF THE EDITED VALUE IS ZERO
- C INSERTS COMMAS BETWEEN EACH GROUP OF THREE DIGITS IN THE INTEGER PART OF THE EDITED VALUE.
- L LEFT-ADJUSTS THE VALUE IN THE RESULT FIELD.
- Z FILLS UNUSED LEADING POSITIONS IN THE RESULT FIELD WITH ZEROS (AND INSERTED COMMAS, IF THE C QUALIFIER IS USED) INSTEAD OF BLANKS.

DECORATION CODES

- `M␣TEXT␣` PLACES THE TEXT ON THE LEFT OF AN EDITED NEGATIVE VALUE. THE DEFAULT VALUE FOR THE M DECORATIONS IS THE NEGATIVE SIGN, '-'.
- `N␣TEXT␣` PLACES THE TEXT ON THE RIGHT OF AN EDITED NEGATIVE VALUE. THE DEFAULT VALUE IS NULL.
- `P␣TEXT␣` PLACES THE TEXT ON THE LEFT OF AN EDITED POSITIVE OR ZERO VALUE. THE DEFAULT VALUE IS NULL.
- `Q␣TEXT␣` PLACES THE TEXT ON THE RIGHT OF AN EDITED POSITIVE OR ZERO VALUE. THE DEFAULT VALUE IS NULL.
- `R␣TEXT␣` FILLS THE RESULT FIELD WITH THE TEXT, PRIOR TO EDITING THE VALUE INTO THE FIELD. THE TEXT WILL APPEAR IN ALL POSITIONS OF THE RESULT FIELD NOT FILLED BY THE EDITED VALUE. THE TEXT IS USED AS MANY TIMES (REPLICATED) AS NEEDED TO FILL THE FIELD. THE DEFAULT VALUE IS BLANK.

--> 1 FORMAT COLNAMES <--

SYNTAX: RESULT ← FS COLNAMES TEXTVECTOR

REQUIRES: RWTD

REFERENCE: ΔFMT - REPORT FORMATTING IN APL PLUS, STSC

THE FUNCTION COLNAMES PRODUCES COLUMN HEADINGS FOR A ΔFMT RESULT. FS IS THE VECTOR OF FORMAT PHRASES NORMALLY USED BY ΔFMT AND TEXTVECTOR CONTAINS THE COLUMN HEADINGS. THE FIRST CHARACTER OF TEXTVECTOR IS TAKEN AS THE SEPARATOR CHARACTER, AND THE CHARACTERS BETWEEN SUCCESSIVE SEPARATORS WILL APPEAR OVER ΔFMT FIELDS. A ΔFMT FIELD CAN BE SKIPPED BY USING TWO CONTIGUOUS SEPARATORS. THE CHARACTERS ARE RIGHT-JUSTIFIED FOR F, I, AND E FIELDS, AND ARE LEFT-JUSTIFIED FOR A FIELDS. X AND TEXT FIELDS ARE SKIPEP. IF THE NUMBER OF CHARACTERS TO BE PRINTED EXCEEDS THE FIELD WIDTH, THAT PART OF THE LINE WILL BE FILLED WITH STARS, EXCEPT THAT OVERSIZE TEXT ASSOCIATED WITH A FIELDS WILL BE PLACED IN FOLLOWING X AND TEXT POSITIONS.

```
VCOLNAMES[ ]∇
∇ R←A COLNAMES B;C;D;E;S
[1] C←R←0ρA←((E←~?1)ρ0),,RWTD A
[2] +(1≥ρA)ρ12
[3] +(A[3]≥4)ρ7
[4] C←C,A[1]ρ-A[2]
[5] 4←4+A
[6] +2
[7] C←C,(0.5×4≠A[3])+×/A[1 2]
[8] +(1≥ρA+4+A)ρ12
[9] +(A[3]< 4 5)/ 2 7
[10] C←(¯1+C),(¯1+C)+×/A[1 2]
[11] +8
[12] A←+ /|LC
[13] S←0,E+(B=¯1+B)/1ρB+1ϕ,B
[14] +(1 0.5 =(ρS),1|1+C)/ 22 19
[15] R←R,(|1+C)+(((ρD)>|1+C)/(|1+C)ρ'*'),(1+C)†D+¯1+S[1-E]+S[2-E]†B
[16] S←1+S
[17] C←1+C
[18] +14
[19] R←R,(|1+C)†' '
[20] C←1+C
[21] +14
[22] R←(1,A)ρA†R
∇
```

--> 1 FORMAT CENTER <--

SYNTAX: RESULT ← FS CENTER TEXTVECTOR

REQUIRES: RWTD

REFERENCE: ΔFMT - REPORT FORMATTING IN APL PLUS, STSC

THE FUNCTION CENTER PRODUCES A CENTERED TITLE LINE FOR USE WITH A ΔFMT DISPLAY. FS IS THE VECTOR OF FORMAT PHRASES NORMALLY USED BY ΔFMT, AND THE TEXTVECTOR IS A CHARACTER VECTOR HOLDING THE TITLE. THE RESULT IS A CHARACTER MATRIX WITH ONE ROW AND AS MANY COLUMNS AS A ΔFMT RESULT WOULD HAVE IF USED WITH FS AS THE LEFT ARGUMENT. THE CHARACTERS GIVEN IN TEXTVECTOR ARE CENTERED IN THIS MATRIX. IF TEXTVECTOR IS TOO LONG, IT IS TRUNCATED AT THE RIGHT

```
▽ R←A CENTER B
[1] A++/×/ 0 -2 +RWTD A
[2] R←(1,A)ρA↑((⌈0.5×0⌈A-ρB⌉↑' '),B←,B
▽
```

--> 1 FORMAT RWTD <--

SYNTAX: RESULT ← RWTD FS

REFERENCE: ΔFMT - REPORT FORMATTING IN APL PLUS, STSC

THE RESULT OF RWTD IS A NUMERIC MATRIX WITH AS MANY ROWS AS THERE ARE FORMAT PHRASES IN FS (A VALID ΔFMT FORMAT PHRASE (LEFT-ARGUMENT)), AND FOUR COLUMNS. THE COLUMNS HAVE THE FOLLOWING INTERPRETATION:

- 1 REPETITIONS
- 2 WIDTH OF FIELD
- 3 TYPE OF FIELD, AS FOLLOWS
 - 1 F FIXED-POINT
 - 2 I INTEGER
 - 3 E FLOATING-POINT
 - 4 A CHARACTER DATA
 - 5 X SKIP
 - 6 [TEXT] LITERAL TEXT
- 4 DECIMAL POSITIONS FOR FIXED-POINT, SIGNIFICANCE FOR FLOATING-POINT, AND ZERO OTHERWISE.

ERROR CHECKING IS NOT PERFORMED AND IT IS ASSUMED THAT FS IS A PROPER ΔFMT ARGUMENT.

--> 1 INFNS AKI <--

SYNTAX: R+AKI P

AUTHOR: ROY SYKES, STSC, L.A. OFFICE

WHEN EXECUTED, <AKI> PRINTS OUT THE PROMPT P AND AWAITS A USER RESPONSE ON THE SAME LINE. CONSECUTIVE SPACES ARE COMPRESSED OUT OF THE RESPONSE, WHICH IS THEN RETURNED AS THE EXPLICIT RESULT OF THE FUNCTION.

```
∇ R+AKI AKI;A
[1]  ▢+AKI
[2]  20I,0
[3]  +(4 7 =pA+,▢)/ 9 8
[4]  R+' '=A+' ',A
[5]  R+1+(R+1+R,1)/A
[6]  5I120L0[ 140+2*(pA)+ 1+pAKI
[7]  +0
[8]  +4*'SUSPEND'v.⌘A
[9]  +('STOP'v.⌘A)p4
[10]  +
∇
```

--> 1 INFNS AYN <--

SYNTAX: R+AYN P

REQUIRES: AKI

AUTHOR: ROY SYKES, STSC, L.A. OFFICE

THE FUNCTION <AYN> PRINTS THE PROMPT, SPECIFIED BY THE MAIN PROGRAM IN THE VARIABLE P, AND ATTEMPTS TO RECEIVE A YES OR NO REPOSE FROM THE USER. THE ONLY RESPONSES ACCEPTED ARE: YES, NO, Y, OR N. ANY OTHER RESPONSE WILL CAUSE THE <AYN> FUNCTION TO PRODUCE AN ERROR MESSAGE AND REQUEST THE USER TO TRY AGAIN. WHEN A VALID RESPONSE IS GIVEN BY THE USER, THE <AYN> FUNCTION RETURNS TO THE MAIN PROGRAM A 1 IF THE RESPONSE WAS YES AND A 0 IF THE RESPONSE WAS NO.

```
∇ R+AYN AYN;A;N;T
[1]  ▢ ACCEPT YES/NO INPUT.  REQUIRES <AKI>, AND <BS>.  AUTHOR: R. SYKES, STSC.
[2]  A+ 4 3 p'YESNO Y N '
[3]  T+AYN
[4]  +(0 3 ≥pN+AKI T,' _____',4pBS)/ 3 6
[5]  +4,pT+'PLEASE ANSWER YES OR NO:'
[6]  +(v/T+A^.=3+N)+5
[7]  R+≥/T
∇
```

--> 1 INFNS INP <--

SYNTAX: R+INP L

AUTHOR: ROY SYKES, STSC, L.A. OFFICE

THE FUNCTION <INP> ALLOWS THE USER TO SPECIFY THE ACCEPTABLE NUMBER OF INPUTS. THE RIGHT ARGUMENT IS ONE OR MORE NUMBERS WHICH REPRESENT THE ACCEPTABLE SIZE OF THE INPUT VECTOR. FOR INSTANCE, IF YOU WOULD ACCECT EITHER 1 OR 5 NUMERIC INPUTS, WHERE IN THE CASE WHERE 1 WAS GIVEN IT WOULD BE USED FOR ALL FIVE NUMBERS, THE FOLLOWING FORM WOULD BE USED:

R+INP 1 5

THE <NIP> FUNCTION WOULD ACCEPT A SERIES OF NUMBERS AND CHECK TO SEE IF THERE WAS EITHER 1 OR 5 VALUES GIVEN. IF THERE WAS, THE RESULT WOULD BE RETURNED TO THE CALLING PROGRAM. IF THERE WAS A LARGER OR SMALLER GROUP OF NUMBERS GIVEN, AN APPROPRIATE ERROR MESSAGE WOULD BE GIVEN.

```

V Z+INP X;V;EXIT
[1] * INPUT X VALUES. HAS 'EXIT' (-) OPTION. REQUIRES: <ΔFMT>. AUTHOR: C. CLAU
    SEN.
[2] EXIT← 0 0 ρ0
[3] +(1≥ρρZ+□)ρ5
[4] +
[5] +6*(^/X≠ρ,Z)^×ρ,X
[6] Z+ ' '=V+ ' LENGTH ERROR...REENTER ',(-3+, 'I3, □ OR□' ΔFMT X), ' VALUE',((v/1
    ≠X,ρ,X)ρ'S'),'. '
[7] 1+(Z*1+Z,1)/V
[8] +3
V

```

--> 1 INFNS IPI <--

SYNTAX: R+IPI P

REQUIRES: AKI MIM

AUTHOR: CHRIS CLAUSEN, PHYSIOLOGY DEPARTMENT, UCLA

THE FUNCTION <IPI> ALLOWS THE MAIN PROGRAM TO PROMPT THE USER FOR A SERIES OF POSITIVE INTEGERS. THE MAIN PROGRAM PRINTS THE RIGHT ARGUMENT, WHICH IS AN ALPHA NUMERIC LITERAL VECTOR (PROMPT) EXPLAINING TO THE USE WHAT NUMBERS HE SHOULD ENTER. THE FUNCTION THEN WILL ACCEPT THE RESPONSE FROM THE USER, TYPED ON THE SAME LINE AS THE PROMPT, AND RETURN THE RESULT TO THE MAIN PROGRAM. IF ANYTHING OTHER THEN POSITIVE INTEGERS ARE ENTERED AN APPROPRIATE ERROR MESSAGE WILL BE GIVEN. THIS PROGRAM WILL ACCEPT EITHER 1 VALUE, WHICH IS CONVERTED INTO A SCALAR; OR A SERIES OF VALUES, WHICH ARE CONVERTED INTO A VECTOR.

--> 1 INFNS LINP <--

SYNTAX: R+LINP S

AUTHOR: CHRIS CLAUSEN, PHYSIOLOGY DEPARTMENT, UCLA

THE FUNCTION <LINP> IS DESIGNED TO ACCEPT INPUT THAT WOULD NORMALLY NOT FIT ON ONE LINE. THE MAIN PROGRAM SPECIFIES THE NUMBER OF INPUTS DESIRED, S, AND THE <LINP> CONTINUES TO ASK FOR MORE DATA UNTIL IT HAS RECEIVED THE REQUIRED NUMBER. IF THE USER TYPES IN MORE NUMBERS THAN ARE NEEDED, THE FUNCTION WILL ASK THE USER TO REENTER JUST THE LAST LINE OF HIS INPUTS.

```

      ▽ Z+LINP X;EXIT;END;N;M
[1]  * LARGE INPUT OF X VALUES. HAS 'EXIT' OPTION. REQUIRES: <NOSPACE>, <ΔFMT>
      .;AUTHOR: CHRIS CLAUSEN, UCLA-PHYSIOLOGY
[2]  EXIT+ 0 0 0 ρ0
[3]  END+ 0 0 0 ρ0
[4]  M+|Γ/X
[5]  Z+1 0
[6]  *(2 3 =ρρN+□)/ 12 15
[7]  *(M<ρN+Z,N)ρ11
[8]  Z+N
[9]  *(M=ρN)ρ0
[10] *6
[11] *6,ρ□+'LENGTH ERROR...REENTER LAST LINE.'
[12] *((0=ρ,X)∨X∨.=ρZ)/0
[13] NOSPACE 'LENGTH ERROR...REENTER ',(-3+, 'I3, OR' ΔFMT X), ' VALUES.'
[14] *5
[15] *
      ▽
```

--> 1 INFENS NIP <--

SYNTAX: R←NIP P

REQUIRES: AKI LF

AUTHOR: ROY SYKES, STSC, L.A.OFFICE

THE FUNCTION <NIP> ACCEPTS A PROMPTED NUMERIC ANSWER AND RETURNS THE RESULT TO THE CALLING PROGRAM. THE RIGHT ARGUMENT, P, IS A ALPHANUMERIC LITERAL DIFINING TO THE USER WHAT INFORMATION IS DESIRED. THIS FUNCTION WILL ACCEPT JUST ONE NUMBER, BUT THAT NUMBER CAN BE ANY VALID REPRESENTATION ACKNOWLEDGEABLE IN APL.

```
∇ R←NIP P;O;S;I;D;E;T
[1]  ⍺ NUMERIC INPUT;AUTHOR: ROY SYKES, STSC.
[2]  +4, O←-?1
[3]  E←P←'INCORRECT NUMBER...PLEASE REENTER',(1+(0=ρE)ρ':'),E
[4]  E←(1+I←O+F1'⍺')+P←,P
[5]  P←T←P
[6]  +(0 1 =ρT←AKI P)/ 6 9
[7]  T←O+'0123456789.'-E'1T
[8]  +(13εT←(S←((P←1)↑T)ε 11 12)↑T)ϕ 15 11
[9]  +(11 10 0 ≤T←O+'0123456789□'1T)/E, 10 20
[10] +0, ρR←□
[11] P←(1+D←O+T113)↑T
[12] +(√/10≤P←(I←11=1↑P)↑P) ρE
[13] +(0ε(ρP), ρT←D↑T) ρE
[14] P←10*(¯1*I)×10↓P
[15] +((√/T≥11)√10εD←(1+I←O+T110)↑T) ρE
[16] +(0=ρT←(I↑T),D) ρE
[17] T←P×(¯1 0.1 ×.*S,ρD)×10↓T
[18] +19+T≠⌊T
[19] T←⌊T
[20] R←T
∇
```

--> 1 INFNS BS <--

THE BS CHARACTER CAN BE USED TO BACKSPACE THE PRINT ELEMENT. IT CAN BE STORED LIKE ANY OTHER APL CHARACTER IN A CHARACTER ARRAY. AN EXAMPLE OF ITS USE IS:

R←'THIS IS A TEST',(14ρBS),14ρ'_'

R NOW LOOKS LIKE THIS:

THIS IS A TEST

--> 1 INFNS CR <--

THE CR CHARACTER CAN BE USED TO CAUSE A CARRIAGE RETURN WITHIN A PROGRAM. IT CAN BE STORED LIKE ANY OTHER APL CHARACTER. FOR EXAMPLE IT CAN BE IMBEDDED WITH IN A CHARACTER ARRAY. FOR EXAMPLE:

R←'THIS',CR,'IS',CR,'A',CR,'TEST'

R IS NOW THIS:

THIS
IS
A
TEST

--> 1 INFNS LF <--

THE LINE FEED CHARACTER CAN BE USED TO CAUSE THE TERMINAL TO SKIP LINES WITHOUT RETURNING THE CARRIAGE. FOR INSTANCE, IF THE PRINT BALL WAS POSITIONED AT POSITION 50 ON THE PAGE, 4 LF'S WOULD CAUSE THE TERMINAL TO ADVANCE 4 LINES, BUT THE TYPE BALL WOULD STILL BE IN POSITION 50.

ENTER DOCUMENT NUMBER(S) TO PRINT:

5.0

OUTLINE OF THE COURSE

5.1

Class Schedule and Reading Assignments

INTERACTIVE COMPUTING FOR MANAGEMENT EDUCATORS

Graduate School of Management
University of California, Los Angeles

CLASS SCHEDULE

<u>Day</u>	<u>Time</u>	<u>Topic</u>	
		<u>Group A</u>	<u>Group B</u>
Monday, 30 June	9:00 - 10:00 a.m.	Course Introduction and Overview: Administrative Details of the Program (2232A GSM)	
	10:00 - 11:00 a.m.	Tour of the UCLA Campus	
	11:00 - 11:30 a.m.	Tour of UCLA Campus Computing Network (CCN)	Visit Ackerman and Kerckhoff Halls
	11:30 - 12:00 a.m.	Visit Ackerman and Kerckhoff Halls	Tour of UCLA Campus Computing Network (CCN)
	12:00 - 1:00 p.m.	Lunch, UCLA Faculty Center	
	1:00 - 2:30 p.m.	Assignment #1, Lecture (2232A GSM)	
	2:30 - 4:00 p.m.	Assign. #2, Lecture (2232A GSM)	Assign. #1, Lab. (2035C GSM)
	4:00 - 5:30 p.m.	Assign. #1, Lab. (2035C GSM)	Assign. #3, Lecture (2232A GSM)
Tuesday, 1 July	8:30 - 10:00 a.m.	Assign. #3, Lecture (2232A GSM)	Assign. #3, Lab. (2035C GSM)
	10:00 - 10:15 a.m.	Break	Break
	10:15 - 11:45 a.m.	Assign. #3, Lab. (2035C GSM)	Assign. #4, Lecture (2232A GSM)
	11:45 - 12:50 p.m.	Lunch, UCLA Faculty Center	
	12:50 - 2:20 p.m.	Assign. #4, Lecture (2232A GSM)	Assign. #4, Lab. (2035C GSM)
	2:20 - 2:25 p.m.	Break	Break

	2:25 - 3:55 p.m.	Assign. #4, Lab. (2035C GSM)	Assign. #5, Lecture (2232A GSM)
	3:55 - 4:00 p.m.	Break	Break
	4:00 - 5:30 p.m.	Assign. #5, Lecture (2232A GSM)	Assign. #5, Lab. (2035C GSM)
Wednesday, 2 July	8:30 - 10:00 a.m.	Assign. #5, Lab. (2035C GSM)	Assign. #6, Lecture (2232A GSM)
	10:00 - 10:15 a.m.	Break	Break
	10:15 - 11:45 a.m.	Assign. #6, Lecture (2232A GSM)	Assign. #6, Lab. (2035C GSM)
	11:45 - 12:50 p.m.	Lunch, UCLA Faculty Center	
	12:50 - 2:20 p.m.	Assign. #6, Lab. (2035C GSM)	Assign. #7, Lecture (2232A GSM)
	2:20 - 2:25 p.m.	Break	Break
	2:25 - 3:55 p.m.	Assign. #7, Lecture (2232A GSM)	Assign. #7, Lab. (2035C GSM)
	3:55 - 4:00 p.m.	Break	Break
	4:00 - 5:30 p.m.	Assign. #7, Lab. (2035C GSM)	Assign. #8, Lecture (2232A GSM)
Thursday, 3 July	8:30 - 10:00 a.m.	Assign. #8, Lecture (2232A GSM)	Assign. #8, Lab. (2035C GSM)
	10:00 - 10:15 a.m.	Break	Break
	10:15 - 11:45 a.m.	Assign. #8, Lab (2035C GSM)	Assign. #9, Lecture (2232A GSM)
	11:45 - 12:50 p.m.	Lunch, UCLA Faculty Center	
	12:50 - 2:20 p.m.	Assign. #9, Lecture (2232A GSM)	Assign. #9, Lab. (2035C GSM)
	2:20 - 2:25 p.m.	Break	Break
	2:25 - 3:55 p.m.	Assign. #9, Lab. (2035C GSM)	Assign. #10, Lecture (2232A GSM)

	3:55 - 4:00 p.m.	Break	Break
	4:00 - 5:30 p.m.	Assign. #10, Lecture (2232A GSM)	Assign. #10, Lab. (2035C GSM)
Friday, 4 July	8:30 - 10:00 a.m.	Assign. #10, Lab. (2035C GSM)	Assign. #11, Lecture (2232A GSM)
	10:00 - 10:15	Break	Break
	10:15 - 11:45 a.m.	Assign. #11, Lecture (2232A GSM)	Assign. #11, Lab. (2035C GSM)
	11:45 - 12:50 p.m.	Lunch , Buffet in 2355 GSM	
	12:50 - 2:20 p.m.	Assign. #11, Lab. (2035C GSM)	Assign. #12, Lecture (2232A GSM)
	2:20 - 2:25 p.m.	Break	Break
	2:25 - 3:55 p.m.	Assign. #12, Lecture (2232A GSM)	Assign. #12, Lab. (2035C GSM)
	3:55 - 4:00 p.m.	Break	Break
	4:00 - 5:30 p.m.	Assign. #12, Lab. (2035C GSM)	Assign. #13, Lecture (2232A GSM)
Monday, 7 July	8:30 - 10:00 a.m.	Assign. #13, Lecture (2232A GSM)	Assign. #13, Lab. (2035C GSM)
	10:00 - 10:15 a.m.	Break	Break
	10:15 - 11:45 a.m.	Assign. #13, Lab. (2035C GSM)	Assign. #14, Lecture (2232A GSM)
	11:45 - 12:50 p.m.	Lunch, UCLA Faculty Center	
	12:50 - 2:20 p.m.	Assign. #14, Lecture (2232A GSM)	Assign. #14, Lab. (2035C GSM)
	2:20 - 2:25 p.m.	Break	Break
	2:25 - 3:55 p.m.	Assign. #14, Lab. (2035C GSM)	Assign. #15, Lecture (2232A GSM)
	3:55 - 4:00 p.m.	Break	Break

	4:00 - 5:30 p.m.	Assign. #15, Lecture (2232A GSM)	Assign. #15, Lab. (2035C GSM)
Tuesday 8 July	8:30 - 10:00 a.m.	Assign. #15, Lab. (2035C GSM)	Assign. #16, Lecture (2232A GSM)
	10:00 - 10:15 a.m.	Break	Break
	10:15 - 11:45 a.m.	Assign. #17, Lecture (2232A GSM)	
	11:45 - 12:50 p.m.	Lunch, UCLA Faculty Center	
	12:50 - 2:20 p.m.	Assign. #18, Lecture (2232A GSM)	
	2:20 - 2:25 p.m.	Break	Break
	2:25 - 3:55 p.m.	Assign. #16, Lecture (2232A GSM)	Programming Project Lab. (2035C GSM)
	3:55 - 4:00 p.m.	Break	Break
	4:00 - 5:30 p.m.	Programming Project, Lab. (2035C GSM)	
Wednesday, 9 July	8:30 - 10:45 a.m.	Programming Project, Lab. (2035C GSM)	
	10:45 - 11:00 a.m.	Break	
	11:00 - 12:00 a.m.	Commerical Applications of APL, Scientific Time Sharing Corp., Lecture, (2232A GSM)	
	12:00 - 1:00 p.m.	Lunch, UCLA Faculty Center	
	1:00 - 5:30 p.m.	Programming Project, Lab (2035C GSM)	
Thursday, 10 July	8:30 - 10:45 a.m.	Programming Project, Lab (2035C GSM)	
	10:45 - 11:00 a.m.	Break	
	11:00 - 12:00 a.m.	Commerical Applications of APL, Proprietary Computer Systems, Inc., Lecture (2232A GSM)	
	12:00 - 1:00 p.m.	Lunch, UCLA Faculty Center	
	1:00 - 5:30 p.m.	Programming Project, Lab. (2035C GSM)	
Friday, 11 July	8:30 - 10:00 a.m.	A Review of APL Usage, Lecture (2232A GSM)	
	10:00 - 10:15 a.m.	Break	
	10:15 - 11:00 a.m.	Pedagogical Approaches to Teaching APL, Lecture (2232A GSM)	

11:00 - 12:00 a.m.	Programming Project, Lab. (2035C GSM)
12:00 - 1:00 p.m.	Lunch, UCLA Faculty Center
1:00 - 5:30 p.m.	Programming Project, Lab. (2035C GSM)

6/30/75

INTERACTIVE COMPUTING FOR MANAGEMENT EDUCATORS

Graduate School of Management
University of California, Los Angeles

Reading Assignments

Note: "Buckley et al." refers to Buckley, Nagaraj, Sharp, and Schenck, Management Problem-Solving in APL, Melville, 1974.

"G. & R." refers to Gilman & Rose, APL: An Interactive Approach (2nd Edition), John Wiley & Sons, 1974.

<u>Assign. #</u>	<u>Required Reading</u>
1.	Buckley <u>et al.</u> , Chapter 1. G. & R., Chapters 1 & 2.
2.	Bohl, <u>Information Processing</u> , SRA, 1971, pp. 1-37, 49-93, 106-171, 189-220, 231-235 and 247-282.
3.	G. & R., Chapters 3, 4, & 5.
4.	G. & R., Chapters 6 & 7.
5.	G. & R., Chapters 8 & 9.
6.	G. & R., Chapters 10 & 12.
7.	G. & R., Chapters 11 & 13.
8.	G. & R., pp. 131-137, 159-164, Chapters 22 & 23
9.	G. & R., Chapter 14 & 15.
10.	G. & R., Chapters 16, 17, & 28.
11.	G. & R., Chapters 18 & 19.
12.	G. & R., Chapters 20 & 24.
13.	G. & R., Chapters 25, 26, & 27.

14. UCLA APL Library Discription (Handout); PLOT Description (Handout); Buckley et al., Chapter 7.
15. STSC, APL Plus Δ FMT Report Formatting Instructions Manual; G. & R., Chapter 21 (lightly)
16. No reading assignment (a discussion of programming techniques for use with APL).
17. G.& R., Chapters 29, 30, & 31; STSC, APL Plus File Subsystem Instruction Manual.
18. G.& R., Chapters 32 & 33.

6/30/75

5.2

Class Handouts

5.2.1

A Guide to Writing APL Programs

A GUIDE TO WRITING APL PROGRAMS

1. Define the Problem.

Read the problem statement carefully and decide what it is that is really being asked for.

2. Identify the Steps Needed.

Break down the problem into the logical steps needed to solve the problem (not the specific APL instructions).

3. Define the Logic of Each Step.

Here the exact operations needed to accomplish the desired results are defined; e.g., "Input data from a user"--a step as defined in No. 2 above--would require (1) "Display instructions to the users of your program," (2) "Accept a literal input from the user, and (3) "Store the input in a variable," would be the No. 3 equivalent of No. 2.

4. Write the Code and Define the Function.

Taking the detailed logic of Step No. 3, write the actual APL code needed to accomplish each step or task. If you are unsure about a particular operator, or how a line of code will actually execute, you may wish to test it out first before going into function definition mode.

5. When You Have Defined the Function, Test It with Sample Data.

In testing your function, use "typical" data first, and then use more unusual data (e.g., vectors rather than scalars, negative numbers, zeros, etc.).

6. When the Function "Hangs," Examine the Indicators.

The error messages (e.g., *LENGTH ERROR*, *RANK ERROR*, *SYNTAX ERROR*, etc.) and the location of the caret (the "^") can be very helpful in determining what is wrong.

7. Examine the Line of Code at Which Execution Stopped.

It is possible that the problem was caused by a mistake in a preceding line of code (but not of the sort that would produce an error message), but you should first check out the line where execution stopped. In analyzing a line, the following procedure is a helpful guide:

- START AT THE RIGHT AND WORK TO THE LEFT (the same way APL executes!)
- Ask the following questions to yourself of each term encountered:
 - 1) Is it an argument or an operator?
 - 2) If an argument, is it a matrix, vector, or scalar?
 - 3) If an operator, is it monadic or dyadic? (To tell, look to see if there is a left argument.)
 4. If it is dyadic, is the left argument a matrix, vector, or scalar and is it compatible with the right argument?
- Once the above have been determined, execute (in desk calculator mode) the portion of the line of code just analyzed.
- Using the result of the preceding execution, proceed again to the left until the entire line of code has been checked out and produces the desired result. (N.B., the first error you find and correct may not be the only error in the line.)
- Two of the most common errors are to overlook the strict right to left execution (which can frequently be corrected by the use of parentheses) and the failure to recognize that vector operations are involved (e.g., thus creating *LENGTH*, *RANK*, or *INDEX* errors). This latter problem may require a variety of solutions (e.g., reduction, raveling, etc.) as determined by each situation.

8. Clear the Suspend Execution.

Although it is possible to continue the execution of the suspended program (by branching to the point in the program where the execution was suspended), it is probably easier to terminate execution and start again. You can clear out the suspended execution simply by typing a right arrow ("→"). (To check to see if all suspensions have been cleared, type ")SI". If there are none, nothing will print; if some still exist, one line will print for each suspension stored. As above, one right arrow is needed to clear each suspension.)

9. Correct the Error.

By use of function editing, go into function definition mode and change the incorrect line of code into its corrected version.

10. Retest the Function.

Quite frequently, another error will be detected, so Steps No. 6, 7, and 8 should be repeated.

11. When No Errors are Detected, "Desk Check" Your Results.

Even though no errors have been detected by the system, it does not necessarily mean that the results are what are desired. Therefore, manually calculate the answer for a few test data conditions to insure that the function is working as planned.

12. (Optional) When Correct, See If the Function Can Be Shortened.

Because APL executes in an interpretive mode, the more lines of code there are, the longer execution will take. Therefore--if you have time and/or the function is likely to be used frequently--investigate ways whereby one or more lines of code can be combined into a single line.

P.S. Don't forget to save a copy of your now-correct function in your stored workspace.

5.2.2

Signing-On to the UCLA APL System

INTERACTIVE COMPUTING FOR MANAGEMENT
EDUCATORS

NAME -----

APL NUMBER -----

PASSWORD -----

NEW PASSWORD -----

GTE TERMINALS

SIGNING-ON

1. TURN THE TERMINAL ON BY FLIPPING THE ON-OFF SWITCH ON THE LEFT OF THE KEYBOARD.
2. PICK UP THE PHONE AND DIAL ONE OF THESE NUMBERS: 50911, 50921, 50931, 50941 OR 50721. WHEN YOU HEAR A HIGH PITCHED TONE, PLACE THE HEADSET ON THE TABLE AND PRESS DOWN THE SILVER BUTTON IN THE CRADLE OF THE PHONE. THE ON-LINE LIGHT ON THE LEFT SIDE OF THE KEYBOARD SHOULD COME ON.
3. TYPE '*' (UPPER CASE P) FOLLOWED BY 'α' (UPPER CASE A) FOLLOWED BY ')' AND PRESS 'RETURN'; EXAMPLE *α) (CARRIAGE RETURN). THE COMPUTER WILL RESPOND WITH THE MESSAGE 'INCORRECT SIGN-ON' AND PRINT A SHORT LINE OF OVERSTPUCK CHARACTERS TO OBSURE THE NEXT TYPED ENTRY.
4. TYPE YOUR APL NUMBER AND YOUR PASSWORD; EXAMPLE)1234;MCLEAN (DO NOT FORGET THE RIGHT PARENTHESIS BEFORE YOUR APL NUMBER AND THE COLON BEFORE YOUR PASSWORD).

ERROR CORRECTION

1. STRIKE THE BACKSPACE KEY TO MOVE THE CARRIAGE OVER TO WHERE THE ERROR BEGINS.
2. HIT THE INDEX KEY ON THE UPPER RIDE SIDE OF THE KEYBOARD. AN INVERTED CARET (∨) WILL APPEAR UNDER THE CHARACTER AT THAT POINT. EVERYTHING ABOVE AND TO THE RIGHT OF THE CARET IS WIPED OUT FROM THE MEMORY OF THE SYSTEM.
3. TYPE THE CORRECTION AND THE REMAINDER OF THE LINE.

SIGNING-OFF

1. TYPE)OFF OR)CONTINUE AND PRESS 'RETURN'.
2. TURN OFF THE TERMINAL.
3. PLACE THE PHONE HEADSET BACK IN THE CRADLE.

ANDERSON JACOBSON TERMINAL 841

SIGNING-ON

1. TURN THE TERMINAL ON BY FLIPPING THE ON-OFF SWITCH ON THE RIGHT OF THE KEYBOARD.
2. CHECK THE SMALL SWITCH LOCATED IN THE FRONT LEFT CORNER, UNDER THE KEYBOARD, AND SWITCH IT TO THE RIGHT.
3. PICK UP THE PHONE AND DIAL ONE OF THESE NUMBERS, 50911, 50921, 50931, 50941, 50721. WHEN YOU HEAR A HIGH PITCHED TONE, INSERT THE PHONE HEADSET INTO THE COUPLER, MAKING SURE THE CORD END IS ON THE CORRECT SIDE. THE LIGHTS 'C' AND 'S' MUST BE ON IF THE CONNECTION IS MADE.
4. TYPE 'A' FOLLOWED BY '9' FOLLOWED BY ')' AND PRESS 'RETURN'; EXAMPLE: A9) (CARRIAGE RETURN). THE COMPUTER RESPONDS WITH THE MESSAGE, 'INCORRECT SIGN-ON' AND PRINTS A SHORT LINE OF OVERSTRUCK CHARACTERS TO OBSURE THE NEXT TYPED ENTRY.
5. TYPE YOUR APL NUMBER AND YOUR PASSWORD; EXAMPLE:)1234:MCLEAN (DO NOT FORGET THE RIGHT PARENTHESIS BEFORE YOUR APL NUMBER AND THE COLON BEFORE YOUR PASSWORD).

ERROR CORRECTION

1. STRIKE THE BACKSPACE KEY TO MOVE OVER TO WHERE THE ERROR BEGINS.
2. HIT THE ATTENTION KEY (ATTN) IN THE UPPER RIGHT HAND CORNER. AN INVERTED CARET (v) WILL APPEAR UNDER THE CHARACTER AT THAT POINT. EVERYTHING ABOVE AND TO THE RIGHT OF THE CARET IS WIPED OUT FROM THE MEMORY OF THE SYSTEM.
3. TYPE THE CORRECTION AND THE REMAINDER OF THE LINE.

SIGNING-OFF

1. TYPE:)OFF OR)CONTINUE AND PRESS 'RETURN'.
2. TURN OFF THE TERMINAL AND THE ACOUSTIC COUPLER.
3. REMOVE THE TELEPHONE HEADSET FROM THE COUPLER AND HANG UP.

JRM 2741 TERMINALS

SIGNING-ON

1. TURN THE TERMINAL ON BY FLIPPING THE ON-OFF SWITCH ON THE RIGHT OF THE KEYBOARD.
2. CHECK THE SWITCH ON THE LEFT SIDE OF THE TERMINAL. IT SHOULD BE SET TO COM NOT LCL.
3. PICK UP THE PHONE AND DIAL ONE OF THESE NUMBERS, 50911, 50921, 50931, 50941, 50721. WHEN YOU HEAR A HIGH PITCHED TONE, INSERT THE PHONE HEADSET INTO THE COUPLER, MAKING SURE THE CORD END IS ON THE CORRECT SIDE. THE LIGHTS 'C' AND 'S' MUST BE ON IF THE CONNECTION IS MADE.
4. TYPE 'A' FOLLOWED BY '[' FOLLOWED BY ')' AND PRESS 'RETURN'; EXAMPLE: A]) (CARRIAGE RETURN). THE COMPUTER RESPONDS WITH THE MESSAGE, 'INCORRECT SIGN-ON' AND PRINTS A SHORT LINE OF OVERSTRUCK CHARACTERS TO OBSCURE THE NEXT TYPED ENTRY.
5. TYPE YOUR APL NUMBER AND YOUR PASSWORD; EXAMPLE:)1234:MCLEAN (DO NOT FORGET THE RIGHT PARENTHESIS BEFORE YOUR APL NUMBER AND THE COLON BEFORE YOUR PASSWORD).

ERROR CORRECTION

1. STRIKE THE BACKSPACE KEY TO MOVE OVER TO WHERE THE ERROR BEGINS.
2. HIT THE ATTENTION KEY (ATTN) IN THE UPPER RIGHT HAND CORNER. AN INVERTED CARET (v) WILL APPEAR UNDER THE CHARACTER AT THAT POINT. EVERYTHING ABOVE AND TO THE RIGHT OF THE CARET IS WIPED OUT FROM THE MEMORY OF THE SYSTEM.
3. TYPE THE CORRECTION AND THE REMAINDER OF THE LINE.

SIGNING-OFF

1. TYPE:)OFF OR)CONTINUE AND PRESS 'RETURN'.
2. TURN OFF THE TERMINAL AND THE ACOUSTIC COUPLER.
3. REMOVE THE TELEPHONE HEADSET FROM THE COUPLER AND HANG UP.

5.2.3

An APL Bibliography

APL: The Language and Its Use

R.P. Polivka and S. Pakin

Raymond P. Polivka and Sandra Pakin, **THE LANGUAGE AND ITS USAGE**, © 1975, pp. 545-569. Reproduced by permission of Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

BIBLIOGRAPHY

This APL Bibliography is organized in sections as follows:

General: Articles that tell about APL or that describe it generally.

Texts: Books for learning APL.

Applications: Books and articles discussing uses of APL and applications in education, business, engineering and science, computers and computing, mathematics and statistics, and miscellaneous.

APL developments: Articles discussing extensions to APL and/or the APL terminal system.

Miscellaneous: Articles of interest to the APL programmer.

Collections: APL journals and conference proceedings.

The bibliography is rich in source materials with the following exceptions. No books or articles written prior to 1968 are included. 1968 was chosen as a cutoff date because APL/360 was not made publicly available by IBM until then. Earlier references tend to use Iverson notation (as formulated in *A Programming Language*, Wiley, New York, 1962) rather than APL. Neither articles nor manuals discussing specific implementations of APL on various machines or by various companies or universities nor how APL is used at various installations are included since such information tends to be transient. No foreign language articles are included. Finally no articles whose distribution is restricted are included.

Bibliography

General: Articles that Tell About APL—Describe or Discuss It Generally

- "APL: The Controversial, Powerful Interactive Language," *Computing Newsletter*, IV, No. 7, 1971, 1-2.
- "APL—The Language of the 4th Generation?," *Computing Newsletter*, March 1972, 1-2.
- Bairstow, J. N., "Mr. Iverson's Language and How it Grew," *Computing Decisions*, Sept. 1969, 42-45.
- Couger, J. D., "Schools, Colleges Attest to APL Growth," *Computer World*, March 29, 1972, 16.
- "Creating Plain Talk for Computers," *IBM Magazine*, Feb. 16, 1970.
- DeVries, W., "Watch Your (Computer) Language," *The Actuary*, March 1971.
- Falkoff, A., "APL/360 History," in *APL Users Conference at S.U.N.Y. Binghamton*, Binghamton, N. Y., July 1969, 8-15.
- Falkoff, A. D., and K. E. Iverson, "The Design of APL," *IBM Journal of Research and Development*, 17, No. 4, July 1973, 324-334.
- Foster, G. H., "APL—A Perspicuous Language," *Computers and Automation*, Nov. 1969, 24-29.
- Griffith, E. V., "APL—A Programming Language," *Chemical Engineering*, March 6, 1972, 99-104.
- Iverson, K. E., "The Story of APL," *Computing Report*, 6, No. 3, 1970, 14-18.
- McCracken, D. D., "Whither APL?," *Datamation*, Sept. 15, 1970, 53-57.
- McMurchie, T. D., "Applause for APL," *Computers and Automation*, March 1970, 4.
- Reeves, R. J. D., "APL, A Potential Liability," *Datamation*, Sept. 15, 1971, 71-72.
- Robertson, R. E., "A Programming Language Just for Everybody," *College Management*, April 1971, 32-34, 38.
- Rueter, D., "Array for APL," *Datamation*, Nov. 15, 1971, 17.
- Sammet, J., "APL/360 and PAT," in *Programming Languages, History and Fundamentals*, Prentice-Hall, Englewood Cliffs, N. J., 1969, pp. 247-253.
- Standish, T. A., *An Essay on APL*, Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh, March 1969.
- Stickler, G. P., "Real-World APL," *Datamation*, Dec. 1, 1971, 19.
- Taylor, A., "APL, A Complex or Simple Language," *Computerworld*, April 1, 1970, 11.

Texts: Teaching APL

- Anger, A., *The APL Language*, Wiley, New York, 1971.
- APL Audio Education Package*, Form No. SR20-9382, SR20-9383, SR20-9384, IBM Corp., 1971.
- Berry, P. C., *APL/1130 Primer*, Form No. GC-20-1967, IBM Corp., 1968.

Bibliography

- Berry, P. C., *APL/360 Primer*, 2nd ed., Form No. GH-20-0689-1, IBM Corp., Jan. 1970.
- Buckley, J. W., D. L. Sharp, M. R. Nagaraj, and J. Schenck, *Management Problem Solving with APL*, Wiley/Becker and Hayes, New York, 1974.
- Falkoff, A. D., and K. E. Iverson, *APL/360 User's Manual*, Form No. GH-20-0683, IBM Corp.
- Gilman, L., and A. J. Rose, *APL—An Interactive Approach*, 2nd ed., Wiley, New York, 1974.
- Gray, L. D., *A Course in APL/360 with Applications*, Addison-Wesley, Reading, Mass., 1973.
- Henson, J. C., and W. F. Manny, *APL—An Introduction*, 2nd ed., Atlanta Public Schools, Atlanta, April 1971.
- Katzan, H., *APL Programming and Computer Techniques*, Van Nostrand Reinhold, New York, 1970.
- , *APL User's Guide*, Van Nostrand Reinhold, New York, 1971.
- Pakin, S., *APL/360 Reference Manual*, 2nd ed., Science Research Associates, Chicago, 1971.
- Pakin, S., and Staff of Computer Innovations, *APL: A Short Course*, Prentice-Hall, Englewood Cliffs, N.J., 1973.
- Prager, W., *An Introduction to APL*, Allyn and Bacon, Boston, 1970.
- Rose, A. J., *APL for Users of Basic*, Scientific Timesharing Corp., Washington, D.C., 1969.
- Smillie, K. W., *An Introduction to APL/360 with Some Statistical Applications*, Publ. No. 19, Dept. of Computing Science, University of Alberta, Edmonton, Alberta, Canada, Jan. 1970.
- Education: Uses of APL and Applications in*
- Adams, W. S., "The Use of APL in Teaching Programming," *Programming Teaching Techniques*, W. M. Turski, ed., American Elsevier Publishing Co., New York, 1973, 1-13.
- Bartoli, G., L. Bartolo, and V. Spadavecchia, "Design of a Simulator for Open Use of an APL Computer in Science Teaching," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, APL Technical Committee, May 1973, pp. 8.1-8.9.
- Bartoli, G., L. Bartolo, P. Berry, and V. Spadavecchia, "APL Functions in Teaching Science: Tools of Analysis and Simulation," in *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 41-48.
- Berry, P. C., A. D. Falkoff, and K. E. Iverson, "Using the Computer to Compute, a Direct but Neglected Approach to Teaching Mathematics," *Report No. 320-2988*, IBM Scientific Center, New York, May 1970.
- Berry, P. C., G. Bartoli, C. Dell'aquila, and V. N. Spadavecchia, "APL and Insight, a Strategy for Teaching," *Colloque APL*, Institut de Recherche d'Informatique et Automatique, Rocquencourt, France, Sept. 1971, 251-272.
- Berry, P. C., G. Bartoli, C. Dell'aquila, and V. Spadavecchia, "APL and Insight: The Use of Programs To Represent Concepts in Teaching," *Report No. 320-3020*, IBM Philadelphia Scientific Center, Philadelphia, March 1973.

Bibliography

- Carolan, K. P., "Teaching APL," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 1.1-1.5.
- Charmonman, S. and D. K. Reed, "A Table-Driven Program to Play a Game of Logical Deduction," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 6.1-6.18.
- Clark, J., and R. Mercer, "Computers in Instruction at the Coast Community Colleges," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 5.1-5.18.
- Conklin, R. F., "Instant CAI," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, Coast Community College District, May 1974, pp. 100-105.
- Davis, L. N., J. Eskinazi, and D. J. Macero, "APL as a Teaching Tool: Two Versatile Tutorial Approaches," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, Board of Education of the City of Atlanta, June 1972, pp. 59-66.
- , "Communication, Computers, and the Classroom," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 7.1-7.11.
- Davis, L., and D. Macero, "Techniques for Computer Administered Examinations of Restricted or Confidential Material," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 124-132.
- DeBarthe, D., "Application of APL/360 in Junior High School," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 133-138.
- DeVahl Davis, G., and W. N. Holmes, "The Use of APL in Engineering Education," *Colloque APL*, Publication IRIA, Sept. 1971, 179-307.
- Dehner, T. R., and B. E. Norcross, "The Evolution of an Interactive Chemistry Laboratory Program," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972, pp. 67-72.
- Edwards, E. M., "APL, A Natural Language for Engineering Education, Part II," *IEEE Transactions on Education*, Nov. 1971, 179-180.
- Falkoff, A. D., and K. E. Iverson, "The Use of Computers in Teaching Mathematics," *Report No. 320-2986*, IBM Philadelphia Scientific Center, Philadelphia, April 1970.
- Foster, G. H., "APL, A Natural Language for Engineering Education, Part I," *IEEE Transactions on Education*, Nov. 1971, 174-185.
- , "Motivating Arrays in Teaching APL," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 3.1-3.8.
- Iverson, K. E., "The Role of the Computer in Teaching," *Queen's Papers on Pure and Applied Mathematics*, No. 13, Kingston, Ontario, Canada, 1968.
- , "Introducing APL to Teachers," *Report No. 320-3014*, IBM Philadelphia Scientific Center, Philadelphia, July 1972.
- , "An Introduction to APL for Scientists and Engineers," *Report No. 320-3019*, IBM Philadelphia Scientific Center, Philadelphia, March 1973.

Bibliography

- , "The Use of APL in Teaching," *Report No. 320-0996-0*, IBM Corp., 1969.
- Kellerman, Anne, and Jeanine Meyer, "Teaching Beyond Elementary APL to Applications Users," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 266-279.
- Konstam, A. H., and J. E. Howland, "APL as a LINGUA FRANCA in the Computer Science Curriculum," *SIGCSE Bulletin*, 6, No. 1, Feb. 1974, pp. 21-27.
- LePage, W. R., "APL, A Natural Language for Engineering Education, Part III," *IEEE Transactions on Education*, Nov. 1971, 180-183.
- Levine, R., "APL Instruction for the Class of History 330," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 292-300.
- Liknaitzky, R., "APL Functions for Use in Junior High School Mathematics," *Report CAI 3-69*, Division of Education Research, Faculty of Education, University of Alberta, Edmonton, Alberta, Canada, Nov. 1969.
- Lippert, H. T., and E. V. Harris, "APL: An Alternative to the Multi-Language Environment for Education," *CAI-Systems Memo-4*, Florida State University, Tallahassee, Aug. 1970.
- Orgass, R. J., "APL in the Teaching of Computational Complexity," in *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 339-346.
- Peelle, H. A., "Teaching Children Thinking via APL," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 0.1-0.2.
- , "A Generalized Learning Game," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 424-428.
- Penfield, P., "Use of APL in Teaching Electrical Network Theory," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972, p. 191.
- Peterson, S. R., "A CAI System Based on Modifications to York/APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 429-437.
- Plum, T. W. S., and G. M. Weinberg, "Teaching Structured Programming Attitudes, Even in APL. By Example," *SIGCSE Bulletin*, 6, No. 1, Feb. 1974, pp. 133-143.
- Raucher, S. M., "APL and Its Use in the Classroom," *AEDS Journal*, Dec. 1968, 3-11.
- Rudberg, D., "APL, A Natural Language of Engineering Education, Part IV," *IEEE Transactions on Education*, Nov. 1971, 183-185.
- Savary, J. R., "APL as an Aid to the Teaching of Undergraduate Economics," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 9.1-9.9.
- , "APL, A Natural Language of Engineering Education, Part V," *IEEE Transactions on Education*, Nov. 1971, 185.
- Schmidt, K., and M. M. Vijay, "Development of Some New Functions in APL for Computer-Aided Individualized Instruction," *SIGCUE Bulletin*, 8, No. 1, ACM, Jan. 1974, pp. 23-26.
- Searle, C. L., "Teaching of Transistor Circuit Design Using a Digital Computer," *IEEE Transactions on Education*, Sept. 1969, 216-219.

Bibliography

- Smillie, K. W., "The Use of APL in the Teaching of Probability," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 475-483.
- Spadavecchia, V. N., P. C. Berry, and G. Bartoli, "An Abstract Machine for the Introduction to Computer Science," *Colloque APL*, Institut de Recherche d'Informatique et D'Automatique, Rocquencourt, France, Sept. 1971, 273-278.
- Sprowis, R. C., "APL in Management Education," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 431-438.
- Taylor, R. W. W., "APL as a Coursewriting Language," *APL Quote-Quad*, June 1972, 3-12.
- Vaskevitch, D., "APL/CAT: APL for Computer Assisted Teaching," *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 544-557.
- Business: Uses of APL and Applications in*
- Andersen, H., O. Bjarkild, and E. Holsøe, "CBC Plan, an APL Application," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 25-32.
- Barasz, J-P., and F. Lapadu-Hargues, "A Multi-Network Pert Program," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, 275-278.
- Carlson, J. G. H., and R. Gilman, "Management Information/Decision Systems Using APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 65-78.
- Courtney, H., "A Generalized Multi-Stage Cost Minimization Model in APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 106-111.
- Courtney, H. M., and C. M. Cheney, "An Application of the Binomial Distribution to Labs Planning," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 16.1-16.17.
- Dyer, D., "The Use of APL in the Management of a Time Sharing Company," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 148-150.
- Flower, D., "The Brown Box Data Management System," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 15.1-15.17.
- Grazis, D. C., "A Computer Model for the Financial Analysis of Urban Projects," *Report No. RC-2850*, IBM Research Center, Yorktown Heights, N.Y.
- Hahn, S. G., "Minipert," *Proceedings of Share XXXVII, 1*, Aug. 1971, 371-379.
- Halmstad, D. G., "APL/Solid. A Life Insurance Management Game," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973.
- Henselyn, R. G., "Pert Project Scheduling in APL*PLUS," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 211-214.
- Hirschberg, D., "A Conversational APL Program for Decision Analysis," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 215-222.

Bibliography

- Hurtabise, R. A., and Y. Poulin, "APL and MIS: Two Compatible Concepts," in *Proceedings of The Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 201-213.
- Jarvis, W., and H. Wobbe, "APL as a Foundation for a Marketing Planning and Analysis System," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 214-217.
- Lee, J. H., "Advanced Decision-making for Private Real Estate Construction Management—An APL Program," in *Proceedings of the Southwestern IEEE Conference*, April 1970, pp. 272-276.
- Løken, S. A., "An Interactive Long Range Planning Model Written in APL," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 289-292.
- McAllister, C., "APL Based On-Line Manpower and Budget Planning," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 293-298.
- McLean, E. R., "The Use of APL in Management Education," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 324-333.
- Mezei, J. E., "Structure in the Traveling Salesman's Problem—The Shklar Algorithm," *Report No. 320-3003*, IBM Philadelphia Scientific Center, Philadelphia, Sept. 1971.
- Mock, J. T., and M. A. Vasarhelyi, *APL for Management*, Melville Publishing Co., Los Angeles, 1972.
- Montalbano, M. S., "Conversational Linear Programming—A User's Manual for LP-APL, Computers in Management Education, Report No. 1," *Report No. G320-3272*, IBM Palo Alto Scientific Center, Palo Alto, Calif., March 1970.
- Ohayon, S. and P. A. LaVallee, "APL Data Management Systems (APLDMS)," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 393-404.
- Redwood, P. H. S., "APL for Business Applications," *Datamation*, May 1972, 82-84.
- Richter, J. A., "APL for Synthesis in Corporate Dynamics," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 369-376.
- Riedy, J. E., "Dynamic Job Tracking Using APL," *APL Congress*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 385-391.
- Rough, J. W., "Xerox Financial Planning and Control System in APL," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 14.1-14.11.
- Seaberg, R. A., "Computer Assisted Forecasting—How Business Is Using APL," *Canadian Data Systems*, Jan. 1971, 30-31.
- , "APL in Financial Forecasting Is Base for Eventual MIS," *Canadian Data Systems*, Feb. 1971, 50-53.
- , "Computer Assisted Forecasting (CAF) at Xerox," *Proceedings of Share XXXVII*, 2, Aug. 1971, 756-761.
- Sharon, E., and M. S. Montalbano, "The Formal Description of Data Processing Procedures: An APL/360 Income Tax Program," *Report No. 320-3212*, IBM Scientific Center, Palo Alto, Calif., May 1968.

Bibliography

- Tava, P. A., "The Financial Planning System—The Application of APL to Financial Modeling," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 487-498.
- Taylor, R., "Budget Planning and Costing in Xerox APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 499-528.
- Vaskocvitch, D., "APL for CAL and Information Management," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 18.1-18.12.
- Wahi, P. N., "AIMS—Applied Information and Management Simulation, A General Business Simulation in APL," *Report No. G320-2066*, IBM Cambridge Scientific Center, Cambridge, Mass., April 1971.
- Wahi, P. N., "A General Management Business Simulation in APL," *IBM Systems Journal*, 1, No. 2, 1972, 169-180.
- Weaver, K. R., "Shared Information Management System," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 558-561.
- Engineering and Science: Applications and Uses of APL in*
- Beam, W. R., "An APL Implementation of Microwave Circuit Analysis," *IEEE NEREM Record*, 1970, 99-105.
- Bork, A. M., "Physics Computer Development Project," *Technical Summary Report*, University of California, Irvine, June 1972.
- Buchheit, E. A., and R. B. Roden, "APL Routines for Evaluating Functions in Mathematical Physics," *Research Report CSRR 2029*, Dept. of Applied Analysis and Computer Science, University of Waterloo, Ontario, Nov. 1970.
- Desblache, A. and V. Riso, "Simulation of Automatic Equalizers to be Used in Data Transmission Systems," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 91-98.
- Fancher, E. J., "Controlling Numerically Controlled Machines Using APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 151-160.
- Foster, G. H., "APL as a Descriptive Language for Associative Processing and Systems Design," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 133-140.
- Ghez, R., "Data Fitting Thin Film Oxidation and Tarnishing Reactions with APL," *Report No. RC-3817*, IBM Research Center, Yorktown Heights, N.Y., April 1972.
- Hansler, E., "Using APL for Network Reliability Calculations," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 223-230.
- Jensen, R. W., T. A. Higbee, and P. M. Hansen, "ECAPL: An APL Electronic Circuit Analysis Program," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972, pp. 161-190.
- Jones, A. L., "The Use of APL/360 in Mechanical Analysis," in *Proceedings of 1970 IEEE International Computer Group Conference*, June 1970, pp. 195-204.

Bibliography

- Jones, A. L., and A. J. Lavin, "Effect of Hammer Length and Nonlinear Paper-Ribbon Characteristics on Impact Printing," *IBM Journal of Research and Development*, March 1971, 108-115.
- LePage, W. R., and L. E. Linge, "Power System Load Flow in APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 287-291.
- Meldal, K. G., "APL/360 Used for Sewer Design in Danish Cities," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 319-324.
- Penfield, P., "A Set of APL Programs for Use in Network Theory," *APL Quote-Quad*, June 1971, 4.
- Posdamer, J. L., and R. Riesenfeld, "The Application of APL in Graphics Computation," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 353-360.
- Spence, R., "Patterns in the Development of Insight: An Experimental Text on Electrical Circuit Theory," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 421-430.
- Thornton, R. D., "Computer Flavored Circuit Theory," *IEEE Transactions on Education*, Sept. 1969, 219-222.
- Wende, C. D., "Modeling a Satellite Experiment in APL," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972, pp. 45-57.

Computers and Computing: Uses and Applications of APL in

- "Aaims' Utilizes Power of APL, Provides Flexible MIS Facility," *Computerworld*, Oct. 4, 1972, 11.
- Abrams, P. S., "Program Writing, Rewriting, and Style," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 1-8.
- Aguzzi, G., F. Cesarini, R. Pinzani, G. Soda, and R. Sprugnoli, "An APL Implementation of an Interpreter Writing System," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 9-16.
- Bartel, D., H. W. Homrighausen, and L. Richter, "Simulation of a Class of Computers and Their Assembly Languages," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 376-384.
- Bingham, H. W., "Use of APL in Microprogrammable Machine Modeling," *ACM Sig Plan Notices*, Oct. 1971.
- Blaauw, G. A., "The Use of APL in Computer Design," in *MC-25 Information Symposium*, Mathematical Center, Amsterdam, 1971, 2.1-2.36.
- Bouricius, W. G., W. C. Carter, K. A. Duke, J. P. Roth, and P. R. Schneider, "Interactive Design of Self-Testing Circuitry," in *Proceedings of the Purdue Symposium on Information Processing*, April 1969, pp. 73-80.
- Brame, J. L., and C. V. Ramamoorthy, "An Interactive Simulator Generating System for Small Computers," *SJCC*, 1971, 425-449.

Bibliography

- Brown, J. A., "Using the Ackerman Function to Rate Programming Languages," *APL Quote-Quad*, April 1970, 4-5.
- Burkley, R. M., "Implementation of a Numerical Control Application Language (APT) by Direct Use of APL," *Report No. G320-2652*, IBM Corp., Los Angeles, Calif., Aug. 1971.
- Compton, M. Y., "APL in PL/1," *Report No. RC4481*, IBM T. J. Watson Research Center, Yorktown Heights, N.Y., Aug. 1973.
- Correia, M., D. Cossman, F. Putzolu, and T. S. Nethen, "Minimizing the Problem of Logic Testing by the Interaction of a Design Group with User Oriented Facilities," in *Seventh Design Automation Workshop*, June 1970, pp. 100-107.
- Day, W. H. E., "Compiler Assignment of Data Items to Registers," *IBM Systems Journal*, 9, No. 4, 1970, 281-317.
- Demars, G., E. Girard, and J. C. Rault, "APL in a Two-Step Programming Technique for Developing Complex Programs," *APL Congress 73*, P. Gjerlov, H. J. Helms and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 83-90.
- Dhatt, G., and L. Robichaud, "Finite Elements, Flow-Graphs and APL," *Colloque APL*, IRIA, Recquencourt, France, Sept. 1971, 37-69.
- Duke, K. A., H. D. Schnurmann, and T. I. Wilson, "System Validation by Three-Level Modeling Synthesis," *IBM Journal of Research and Development*, March 1971, 166-174.
- Falkoff, A. D., "Criteria for a System Design Language," *Software Engineering Techniques*, ed. J. N. Buxton and B. Randell, Pub. NATO Science Committee, Brussels, Belgium, 1970. pp. 88-92.
- Falster, P., "A New Approach in System Design by Utilizing APL," *APL Congress 73*, P. Gjerlov, H. J. Helms and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 109-118.
- Fisher, R. A., "Automated Testing of Software Systems," in *Proceedings of the Third Hawaii International Conference on System Science*, Jan. 1970, part 2, 886-889.
- Foster, G. H. and W. J. Jones, "On the Efficiency of Language Processors," in *Large Scale Information Processing System, Vol. II (Systems: Theory, Advanced Concepts and Designs)*, Rome Air Development Center, Griffiss Air Force Base, N.Y., Report RADC-TR-68-401, April 1969, pp. 1-24.
- Foster, G. H., "Using APL to Investigate Sequential Machines," *IEEE NEREM Record*, 1970, 121-127.
- Foster, G. H., A. E. Spaanenburg, and W. E. Stumpf, *Computer Enhancement Through Interpretive Techniques* (NASA Grant NGR 33-022-125), Dept. of Electrical and Computer Engineering, Syracuse University, N.Y., Jan. 1972.
- Friedman, T.D., "A System of APL Functions to Study Computer Networks," *AFIPS Conference Proceedings*, 42, 1973, pp. 141-148.
- Friedman, T. D., and S. C. Yang, "Methods Used in an Automatic Logic Design Generator—(Alert)," *IEEE Transactions on Computers*, July 1969, 593-614.
- Gerhart, S., "Verification of APL Programs," *APL Congress 73*, P. Gjerlov, H. J. Helms and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 141-149.

Bibliography

- Ghandour, Z. J., "Formal Systems and Analysis," *Report No. 320-3002*, IBM Philadelphia Scientific Center, Philadelphia, June 1971.
- Goncharsky, R. S., A. Rauch, and W. W. White, "Large Scale Mathematical Programming in an APL Environment," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 161-172.
- Grant, C. A., M. L. Greenbert, and D. E. Redell, "A Computer System Providing Microcoded APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 173-179.
- Grebert, A., F. Gerbstadt, and P. Colen, "Space Programming Language Machine Architecture Study," CIRAD Report No. WS-10 300-2, Vol. 1, Claremont, Calif., April 1972.
- Haegi, H. R., "Euler Files—Simulation of a File System with APL," *APL Congress 73*, P. Gjerlov, H. J. Helms and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973.
- Hake, R. B., and D. R. Page, "Tasking in APL," *Report Tr-12.102*, IBM World Trade Corp., Hursley, England, May 1972.
- Hall, F. J., and G. R. Peterson, *Digital Systems: Hardware Organization and Design*, Wiley, New York, 1973.
- Hassitt, A., "Microprogramming and High Level Languages," *International IEEE Computer Conference*, Sept. 1971, 91-92.
- Hasterlik, R. L., "Algorithm Evaluation: A Practical Use for APL/360," *Report No. Tr-01.1182*, IBM Systems Development Division, Endicott Laboratory, Endicott, N.Y., Oct. 1969.
- Hellerman, H., "Some Principles of Time Sharing Scheduler Strategies," *IBM Systems Journal*, 8, No. 2, 1969, 94-117.
- Hellerman, H., and Y. Ron, "A Time-sharing System Simulation and Its Validation," *Report No. 320-2984*, IBM Philadelphia Scientific Center, Philadelphia, April 1970.
- Juran, W., and C. Moore, "An APL Program for Remote Job Entry," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 233-238.
- Kelly, R. A., and J. R. Walters, "APLGOL-2 A Structured Programming System for APL," *Report No. 320-3318*, IBM Scientific Center, Palo Alto, Aug. 1973.
- , "A Structured Programming Language System for APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 275-280.
- Konnerth, K. L., "Use of a Terminal System for Data Acquisition," *IBM Journal of Research and Development*, Jan. 1969, 132-138.
- Laws, B. H., "A Parallel BCH Decoder," *Report No. ERL-603-T2-0670*, Montana State University, Bozeman Electronics Research Lab, June 1970.
- Lehman, M. M., "Syscom, An APL Program for System Evaluation," *Report No. RC-3375*, IBM Research Center, Yorktown Heights, N.Y., May 1971.
- Leiner, A. L., "Supermod—An Analytic Tool for Modeling the Performance of Large Scale Systems," *Report No. RC-2796*, IBM Research Center, Yorktown Heights, N.Y., Feb. 1970.
- Levy, S., R. H. Doyle, and R. M. Heller, "APL as a Development Tool for Special Purpose Processors," in *Proceedings of the National Computer Conference*, May 1974, pp. 279-284.

Bibliography

- Lewis, P. A. W., and P. C. Yue, "Statistical Analysis of Series Events in Computer Systems," *Report No. RC-3606*, IBM Research Center, Yorktown Heights, N.Y., Nov. 1971.
- Lorie, R. A., "APL as a Language for Handling a Relational Data-Base," *Report No. G320-2067*, IBM Cambridge Scientific Center, Cambridge, Mass., March 1971.
- Lorie, R. A., and A. J. Symonds, "Use of a Relational Access Method Under APL," *Report No. G320-2071*, IBM Cambridge Scientific Center, Cambridge, Mass., May 1971.
- MacDonald, Geraldine, "Evaluating the Equality or Inequality of Symbol Strings in APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 301-306.
- Marchal, A., "Data Structure Techniques in an Implementation of APL," Unpublished Ph.D. Thesis, Dept. of Electrical Engineering, Division of Computer Science, University of Utah, Provo, June 1973.
- Maybury, W. J., "Programming Aids in Xerox APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 307-315.
- McFarland, C., "A Language-Oriented Computer Design," *AFIPS Conference Proceedings*, 37, 1970, 629-640.
- Milville-DeChene, A., and L. P. A. Robichaud, "Subtasking in APL," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972, pp. 135-141.
- Moore, C. R., "Character String Manipulation in APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 349-353.
- Nielsen, C. P., T. H. Burnett, and W. Vernon, "APL and Fortran: A Symbiotic Mix for a Small Machine," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 387-392.
- Nissen S. M., and S. J. Wallach, "The ALL Applications Digital Computer," *Proceedings of a Symposium on High-Level-Language Computer Architecture*, SIGPLAN Notices, 8, No. 11, Nov. 1973, pp. 43-51.
- Owens, J. L., "Bulk I/O and Communications with Livermore Time Sharing System," *APL Quote-Quad*, June 1971, 7-8.
- Peccoud, D. M., and G. L. Noguez, "An Array Processor Design for APL-like Data Structure," *Information Processing 71*, ed. C. V. Freiman, North-Holland, Amsterdam, 1972, pp. 717-720.
- Penfield, P., "General Purpose Electric Circuit Analyzer Imbedded in APL," *Electrodynamics Memo No. 15*, Research Lab of Electronics, M.I.T., Cambridge, Mass., Feb. 1971.
- Polivka, R., and K. Haralson, "Microprogramming Training—An APL Application," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, 1972, pp. 154-160.
- Puckett, T. H., "Improved Security in APL Applications Package," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 438-441.
- Raynaud, J., L. P. A. Robichaud, and G. Simian, "APL in the Description and Simulation of Computer Systems," *SIGMICRO Newsletter*, 3, No. 4, Jan. 1973, 5-24.
- Robichaud, L.P.A., "Lexical and Syntactic Analysis of APL, Generation of a Prefix Polish String," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 447-451.

Bibliography

- Roth, J. P., W. G. Bouricius, W. C. Carter, and P. R. Schneider, *Phase II of an Architectural Study for a Self Repairing Computer*, SAMSO-Tr-67-106, Air Force Systems Command, Los Angeles, Calif., Nov. 1967.
- Saal, H. J., "Simulating Computers in Symbolic Assembler Notation," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 20.1-20.6.
- Schaefer, M., *A Mathematical Theory of Global Program Organization*, Prentice-Hall, 1973.
- Schmidt, R. O., "Collection of Walsh Analysis Programs Information Theory," *IEEE Transactions in Electromagnetic Compatibility*, Aug. 1971, 88-94.
- Schroeder, S. C., and L. E. Vaughn, "A High Order Language Optimal Execution Processor," *Proceedings of a Symposium on High-Level-Language Computer Architecture, SIGPLAN Notices*, 8, No. 11, Nov. 1973, pp. 109-116.
- Smillie, K. W., "APLISP: A Simple List Processor in APL," *APL Quote-Quad*, Feb. 1972, 16-20.
- Surkan, A. J., "A Virtual-Variable Name-Specification Operator for High Level Languages," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 484-486.
- Thistle, W. G., and D. S. Galbraith, "APL Flowcharting Functions," *Report No. DREV M-2191/72*, Defense Research Establishment Valcartier, Valcartier, Quebec, Canada, 1972.
- Tuckerman, B., "A Study of the Vigenere-Vernam Single and Multiple Loop Enciphering Systems," *Report No. RC-2879*, IBM Research Center, Yorktown Heights, N.Y., May 1970.
- Wadia, N. C., "Theory of Improvement in System Reliability," *Report No. TR-27.089*, IBM Corp., Kingston, N.Y., Feb. 1972.
- Waldbaum, G., "Evaluating Computing System Changes by Means of Regression Models," *Report No. RC-4060*, IBM Research Center, Yorktown Heights, N.Y., Sept. 1972.
- Woodrum, L. J., "Internal Sorting with Minimal Comparing," *IBM Systems Journal*, 8, No. 3, 1969, 189-203.
- , "A Model of Floating Buffering," *IBM Systems Journal*, 9, No. 2, 1970, 118-144.
- , "Representing Negative Integers in Bit Vectors—A Short Note," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972, p. 58.
- Zaks, R., and D. Steingart, "A Language Machine," *APL Quote-Quad*, Oct. 1971, 34-39.

Mathematics and Statistics: Applications and Uses of APL in

- Alleman, G. B., and J. Richardson, "A Time Study in Numerical Methods Programming," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 6-27.
- Anscombe, F. J., "Use of Iverson's Language APL for Statistical Computing," *Technical Report No. 1*, Dept. of Statistics, Yale University, New Haven, Conn., July 1968.
- Bergquist, J. W., "Algebraic Manipulation," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 45-49.
- Bickert, T. A., "Function to Accelerate and/or Induce Sequence Convergence," *APL Quote-Quad*, April 1970, 8-9.

Bibliography

- Casarosa, V., and S. Trumpy, "Interactive Interpolation and Display of Surfaces," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 59-66.
- Chen, T. C., "The Automatic Computation of Exponentials, Logarithms, Ratios, and Square Roots," *Report No. RJ-970*, IBM Research Center, San Jose, Calif., Feb. 1972. Also *IBM Journal of Research and Development*, July 1972, 380-388.
- Cousineau, F. G., J. F. Perrot, and J. M. Rifflet, "APL Programs for Direct Computation of a Finite Semigroup," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 67-74.
- Dell'Orco, P., "A Syntactic Approach to Automatic Theorem Proving," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 75-82.
- Edwards, E. M., and W. R. Tinga, "An APL Complex Arithmetic Package," *IEEE NEREM Record*, 1970, 106-112.
- Feldbrugge, F. H. J., "An Interactive APL System (ISVAP) for the Verification of APL Programs Using Floyd's Method," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 119-126.
- Fontanella, F., "An APL Polyalgorithm for Function Approximation by Some Interpolating Functions," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 127-132.
- Girard, E., D. Bastin, and J. C. Rault, "A Collection of Graph Analysis APL Functions," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June, 1972, pp. 73-102.
- Halpern, M. M., "Permutations," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 13.1-13.15.
- , "Algebra and the Properties of Scans," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973.
- Hassitt, A., and L. E. Lyon, "Efficient Evaluation of Array Subscripts of Arrays," *IBM Journal of Research and Development*, Jan. 1972, 45-57.
- Heath, J. S., "Planayse—An APL Program for the Analysis of Planar Structures," *Report No. TR-12,103*, IBM World Trade Corp., Hursley, England, June 1972.
- Heilberger, R. M., "APL Functions for Data Analysis and Statistics," *Research Report CP-5*, Dept. of Statistics, Harvard University, March, 1971.
- Horvath, A., "A Conversational Regression Analysis Program in APL," *Report No. TR-22.1400*, IBM Corp., East Fishkill, N.Y., Dec. 1971.
- Iverson, K. E., "Use of APL in Statistics," in *Statistical Computations*, R. C. Milton and J. A. Nelder, eds., Academic Press, New York, 1969, pp. 285-299.
- , "Elementary Algebra," *Report No. 320-3001*, IBM Philadelphia Scientific Center, Philadelphia, June 1971.
- , "Algebra as a Language," *Colloque APL*, IRIA, Rocquencourt, France, Sept. 1971, 5-16.

Bibliography

- Jenkins, M. A., "The Solution of Linear Systems of Equations and Linear Least Squares Problems in APL," *Report No. 320-2989*, IBM New York Scientific Center, New York, June 1970.
- Jones, A. L., "The Fast Fourier Transform Algorithm—Applications and an APL Implementation," IBM Systems Development Divisions, Endicott, N.Y., Report No. TR-01.1560, March 1972.
- Kellerman, Anne, "An Investigation of Curve Fitting Methods Using an APL Graphics Terminal," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 249-265.
- Kellerman, E., and W. C. Rodgers, "APL Tools for Combinatorics," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 239-248.
- Korsan, R. J., "On the Design of a Function to Solve Large Linear Systems of Equations Using the APL*PLUS File Subsystem," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 251-258.
- Korsan, R. T., "A Spline Package in APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 281-286.
- Laflamme, A. K., *Symbolic Differentiation with APL*, DREV, Quebec, Canada, 1971.
- Levy, L. S., "An APL Program for the Multiple Output Two Level Minimization Problem," *Report No. RC-2323*, IBM Research Center, Yorktown Heights, N.Y., Dec. 1968.
- Linden, T. A., "The Ackerman Function in APL," *APL Quote-Quad*, Jan. 1971, 10-11.
- McAuliffe, G. K., "APL Fast Fourier Program," *Report No. RC-2832*, IBM Research Center, Yorktown Heights, N.Y., March 1970.
- McDonnell, E. E., "Complex Floor," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 299-306.
- McIntyre, D. B., "Introduction to the Study of Data Matrices," *An Introduction to Mathematical Geology*, AGI Short Course Lecture Notes, American Geological Institute, Nov. 1969.
- Michel, J. Y., "APLAMBDA: A Lambda-Calculus Machine for APL-Operators," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 325-332.
- More, T., "An Interactive Method for Algebraic Proofs," *Report No. 320-3005*, IBM Philadelphia Scientific Center, Philadelphia, Sept. 1971.
- , "Axioms and Theorems for a Theory of Arrays," *IBM Journal of Research and Development*, 17, No. 2, March 1973, 135-175.
- , "Notes on the Development of a Theory of Arrays," *Report No. 320-3016*, IBM Philadelphia Scientific Center, Philadelphia, May 1973.
- , "Notes on the Axioms for a Theory of Arrays," *Report No. 320-3017*, IBM Philadelphia Scientific Center, Philadelphia, May 1973.
- Murray, M. D., "ISP/1: A Conversational Statistical Package," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 364-372.
- Myers, H. J., and M. Y. Hsiao, "APL Algorithm for Calculating Boolean Difference," in *Proceedings of 1968 Automatic Support Symposium for Advanced Maintainability*, paper 5D, IEEE, Nov. 1968.

Bibliography

- Page, P.D., "An On-line Proof Checker Operating Under APL 360, with Educational Applications in Logic, Mathematics, and Computer Science," *APL Quote-Quad*, Oct. 1971, 30-34.
- Prins, J., "General K Factor Analysis of Variance Program Package for APL Users," Computer Center, State University of New York at New Paltz, Dec. 1972.
- Richardson, J. L., "The Newton Coefficients in APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 443-446.
- Schatzoff, M., "Interactive Statistical Data Analysis-APL Style," *Report No. G320-2079*, IBM Cambridge Scientific Center, Cambridge, Mass., April 1972.
- Schatzoff, M., P. Bryant, and A. P. Dempster, "Interactive Statistical Computation with Large Data Structures," *Report No. 320-2085*, IBM Cambridge Scientific Center, Cambridge, Mass., March 1973.
- Smillie, K. W., "An APL Algorithm for the Critical Path," *Quarterly Bulletin of the Computer Society of Canada*, 8, No. 2, 1968, 6-13.
- , "Statpack 1, An APL Statistical Package," Dept. of Computing Science, Alberta University, Edmonton, Alberta, Canada, Jan. 1968.
- , "The APL Language and Statistical Computations," *Computer Bulletin*, 13, No. 8, 1969, 896-897.
- , "Statpack 2, An APL Statistical Package," *Publication No. 17*, Dept. of Computing Science, University of Alberta, Edmonton, Alberta, Canada, Feb. 1969.
- , "Some APL Algorithms for Orthogonal Factorial Experiments," *Publication No. 18*, Dept. of Computing Science, University of Alberta, Edmonton, Alberta, Canada, June 1969.
- , "Statistical Programs in APL/360," *Computer Bulletin*, 14, No. 5, 1970, 151-152.
- , "APL and Statistics, Programs or Insight," *Colloque APL*, IRIA, Rocquencourt, France, Sept. 1971, 17-35.
- , "Lines Through a Cube and Tic-Tac-Toe," *APL Quote-Quad*, Sept. 1972, 18-28.
- , "APL as a Notation for Statistical Analysis," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972, pp. 146-149.
- Smith, R., "Interpretations of All Possible Reductions of a Boolean Vector Where the Result is 0 or 1" and "Similarities Between Reduction and Scan of Boolean Vectors," *APL Quote-Quad*, Jan. 1973, 34-35.
- Snyder, M., "Interactive Data Analysis and Non-Parametric Statistics," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 483-488.
- Stockwell, R. K., and K. E. Vanbee, "Use of APL to Implement Algorithms for Sparse Linear Systems," *IEEE NEREM Record*, 1970, 113-119.
- Surkan, A. J., "On Line APL Realization of Nonnumeric Algebraic Operations for Recursively Deriving Polynomial Coefficient Series of Orbital Motion," *Report No. RC-2048*, IBM Research Center, Yorktown Heights, N.Y., April 1968.
- , "Symbolic Polynomial Operations with APL," *IBM Journal of Research and Development*, March 1969, 209-211.

Bibliography

- , "Discrete Fast Fourier Transformation Made Simple by a Single Reliable APL Function," *Report No. RC-2591*, IBM Research Center, Yorktown Heights, N.Y., Aug. 1969.
- Tang, D. T., and C. N. Liu, "Distance Two Cyclic Chaining of Constant Weight Codes," *Report No. RC-3448*, IBM Research Center, Yorktown Heights, N.Y., July 1971.
- Vervoort, W., "Correctness Proofs of APL Programs Using the Assertion Method of Floyd," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 465-472.
- Von Maydell, U. M., "An Introduction to Probability and Statistics Using APL," *Publication No. 21*, Dept. of Computing Science, University of Alberta, Edmonton, Alberta, Canada, June 1970.
- Wilhelmi, G., "Formal Differentiation Using APL," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 11.1-11.9.
- Other Applications and Uses of APL*
- Brown, J. A., and B. Warner, "An Application of a Specialized Data Bank for Information Retrieval," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 58-64.
- Coffey, C. E., J. Eskinazi, F. H. Fraser, and D. J. Macero, "A Program for Filing and Searching Literature References," *APL Quote-Quad*, Jan 1973, 28-32.
- Dell'Orco, P., and L. Bartolo, "APL/Info: A System for Automatic Documentation," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 139-147.
- Gray, B. C., "A Military Enlisted Manpower Projection and Simulation Model," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973.
- Gregersen, O. V., "Temperature-Level in Airconditioned Room," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973.
- Gull, W. E., "The Solution of Rational Equations Using Text Processing," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973.
- Hagaman, W. D., and D. J. Linden, "Encoding Verbal Information as Unique Numbers," *IBM Systems Journal*, 11, No. 4, 1972, 278-315.
- Hopkins, M. J. D., H. G. Moria, and T. V. Narayana, "A First APL Tournament Package," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 185-200.
- Jones, A. L., "Cross-Country Scoring and Timing Using APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 218-220.
- Kellerman, E., "An Adaptive Query System," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972, pp. 150-153.
- Kolsky, H. G., "Problem Formulation Using APL," *IBM Systems Journal*, 8, No. 3, 1969, 204-219.

Bibliography

- Kvaternik, R., "An APL Terminal Approach to Computer Mapping," *Technical Report CSRG-21*, Toronto University, Canada, Dec. 1972.
- LePage, W. R., "APL Simulation of Musical Staff Notation," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 281-288.
- Lorie, R. A., and A. J. Symonds, "Interactive Problem Solving Using a Relational Data-Base in APL," in *Conference Digest of International IEEE Computer Conference*, IEEE, Boston, Sept. 1971, pp. 191-92.
- Niehoff, W. H., "Contour Plotting in Rectangular Domains," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 373-386.
- Pages, J-C., "Individualized Documentation Application," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 347-352.
- Pages, J-C., and A. Mauboassin, "Flexible Building of Specialized Retrieval Systems," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 414-423.
- Petersen, T. I., "The Ecology Decision Game Introduction," *Report No. G320-2073*, IBM Cambridge Scientific Center, Cambridge, Mass., Sept. 1971.
- Price, R., "APL Processing of Traffic Data Within a PBX (IBM 3750 System)," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 361-368.
- Raynaud, Y. G., and L. P. A. Robichaud, "Scalar Segment Processor," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, p. 442.
- Rogers, G., "Adaptive Data-Classifying Systems," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 395-400.
- Rosenkrands, B., "Graphics by APL," *Colloque APL*, IRIA, Rocquencourt, France, Sept. 1971, 91-113.
- Rueter, D. B., "Speech Synthesis Under APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 585 ff.
- Rux, W. A., and E. Baelen, "Casper," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 452-470.
- Sandi, C., "From the Free Optimum to the True Optimum: A Direct Search Method for the Transportation Problem," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 415-420.
- Schench, J., J. Peetz, and C. Caluson, "APL Catalog System," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 471-474.
- Smart, J. S., "An APL Version of JABOWA, the Northeastern Forest Growth Simulation," *Report No. RC-3985*, IBM Research Center, Yorktown Heights, N.Y., Aug. 1972.
- Thomson, N. D., "The Use of APL in Constructing School Time-Tables," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 439-446.
- Thygesen, S., and N. C. Mortensen, "APL in Local Government Economical and Physical Planning," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 447-456.

Bibliography

- Wahi, P. N., and T. I. Petersen, "The Ecology Decision Game Management Science and Gaming," *Report No. G320-2076*, IBM Cambridge Scientific Center, Cambridge, Mass., Oct. 1971.
- Weaver, K. R., "An Information Retrieval System Using One File," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 465-472.
- Wende, C. D., "A Modest Information System Implemented on APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 568-580.
- APL Developments*
- Abrams, P. S., "An APL Machine," Ph.d. dissertation, *SLAC Report No. 114*, Stanford University, Stanford, Calif., Feb. 1970.
- Adams, M. F., "The Extended File System at GSFC," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 1-5.
- Alfonseca, M., "An APL-Written APL-Subset to System/7-MSP Translator," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 17-24.
- Amram, Y., B. deCosnac, J. L. Granger, and A. Smoucovit, "An APL Interpreter for Mini-Computers. A Microprogrammed APL Machine," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 33-40.
- Ashcroft, E. A., "Towards an APL Compiler," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 28-38.
- Battarel, G., and D. Tusera, "APL Optimization: A Unified Approach," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 39-44.
- Battarel, G., M. Delbreil, and D. Tusera, "Optimized Interpretation of APL Statements," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 49-58.
- Bauer, A. M., and H. J. Saal, "Does APL Really Need Run-Time Checking?," *Dept. Memo 59-73*, Dept. of Computer Science, State University of New York at Buffalo, April 1972.
- Bernecky, B., "Speeding Up Dyadic IOTA and Dyadic EPSILON," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 479-482.
- Bonyum, D. A., "Mark-Sense APL," *APL Quote-Quad*, June 1971, 18-19.
- Breed, L. M., "Generalizing APL Scalar Extensions," *APL Quote-Quad*, March 1971, 5-7.
- . "Design of the APL*PLUS File Subsystem," *Colloque APL*, IRIA, Rocquencourt, France, Sept. 1971.
- . "APL-ASCII: An ASCII Overlay Stanford for APL Terminals," *APL Quote-Quad*, Jan. 1973, 13-19.
- Brown, J. A., *A Generalization of APL*, Ph.D. dissertation, Systems and Information Science Dept., Syracuse University, Syracuse, N.Y., 1971.

Bibliography

- Charmonman, S., "Sixty-Character Representation of APL Symbols," *APL Quote-Quad*, July 1970, 5-10.
- , "A Generalization of APL Array-Oriented Concept," *APL Quote-Quad*, Sept. 1970, 13-17.
- Clement, J. F., P. Sandery, and B. James, "Development of a Batch APL System," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 85-99.
- Clementi, J. F., and P. P. Fletcher, "Modifications to the APL/1130 System To Provide More Convenient Operating on a Fortran User's Machine," *APL Quote-Quad*, June 1971, 16-18, and Oct. 1971, 40-42.
- Colle, A., "Increasing the Flexibility of HI and PA Messages," *APL Quote-Quad*, Sept. 1972, 29-34.
- Crayton, J., "APL Unary Matrix Operators," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 112-116.
- Edwards, E. M., "Generalized Arrays (Lists) in APL," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 99-106.
- Falkoff, A. D., "Design of a Format Function with a Pictorial Argument," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 107-108.
- , "A Survey of Experimental APL File and I/O Systems in IBM," *Colloque APL*, IRIA, Roquencourt, France, Sept. 1971, 365-374.
- Falkoff, A. D., and K. E. Iverson, "Communication in APL Systems," *Report No. 320-3022*, IBM Philadelphia Scientific Center, Philadelphia, May 1973.
- Fletcher, J., "An 8-Bit ASCII Code," *APL Quote-Quad*, June 1971, 13.
- Foster, G. "File Action in APL," *Sigplan Notices*, 8, No. 4, April 1973, pp. 6-15.
- Gerhart, S. L., *Verification of APL Programs*, Ph.D. dissertation. Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh, Nov. 1972.
- Ghandour, Z., and J. Mezei, "General Arrays, Operators, and Functions," *IBM Journal of Research and Development*, July 1973, 335-352.
- Hail, J. D., and J. D. Kirscher, "A Batch Processing Version of APL/1130 for Student Use," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 180-184.
- Harbinson, M. D., "A Magnetic Tape Format for Communication of APL Functions," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 195-202.
- Harris, L. R., "A Logical Control Structure for APL," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 203-210.
- Hassitt, A., J. W. Lageshulte, and L. E. Lyon, "A Microprogrammed Implementation of an APL Machine," *APL Quote-Quad*, June 1970, 11-12.

Bibliography

- Hassitt, A., J. W. Lageshulte, and L. E. Lyon, "Implementation of a High Level Language Machine," *CACM*, 16, No. 4, April 1973, 199-212.
- Iverson, E. B., "APL/4004 Implementation," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 231-236.
- Jenkins, M. A., "Domino—An APL Primitive Function for Matrix Inversion—Its Implementation and Applications," *Proceedings of Share XXXVII*, 1, Aug. 1971, 379-388, and *APL Quote-Quad*, Feb. 1972, 4-15.
- , "Numerical Analysis and APL, Can They Coexist?," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 237-244.
- Juran, W., "A Fast Formatter for APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 228-232.
- Kalfon, P., "On a Bottom-Up Method for Evaluation of APL Expressions," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 245-250.
- Korsan, R. J., "A Proposed APL Extension," *APL Quote-Quad*, Oct. 1971, 21-23.
- , "APL, the FFT and Ordinary Differential Equations (Odes)," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 259-268.
- Lathwell, R. H., "System Formulation and APL Shared Variables," *IBM Journal of Research and Development*, 17, No. 4, July 1973, pp. 353-359.
- , "File Processing in Terms of Concurrent Processors with Shared Variable Interfaces," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 279-280.
- Leibovitz, C., "Suggestions for a 'Mapped' Extension of APL," in *Proceedings of the Fourth International APL User's Conference*, Atlanta, June 1972, pp. 142-145.
- Liu, Y., "Reverse Operator in APL," *Computing Center News* (Syracuse University, Syracuse, N.Y.), March 1, 1971, 9-10.
- Marchal, A., and R. A. Keir, "The Design of a Structured APL Interpreter," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 313-318.
- McDonnell, E. E., "The Caret Functions," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 316-323.
- McEwan, A., and D. Watson, "APL/360 Recurse, Part I," *APL Quote-Quad*, July 1970, 11-16.
- McMurchie, T. D., "A Limited Character APL Symbolism," *APL Quote-Quad*, Nov. 1970, 3-4.
- Mezei, J., "General Arrays Operators and Functions," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 334-348.
- Moore, C., W. Juran, C. Orndorff, and L. Rice, "The PCS APL Shared File System," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 354-363.
- Murray, R. C., "On Tree Structure Extensions to the APL Language," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 333-339.

Bibliography

- O'Dell, M. D., "APL/XAD: An Extension of APL for Abstract Data Manipulation," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 405-413.
- Penfield, P., "Proposed Notation and Implementation for Derivatives in APL," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 12.1-12.5.
- Perlis, A., "APL as a Conventional Language—What is Missing?," *APL Quote-Quad*, June 1971, 3-4.
- Pettus, C., "Indeterminate (0÷0) Check in APL," *APL Quote-Quad*, Feb. 1972, 15-16.
- Ruggiu, G., and P. Aigrain, "Description of APL Operators," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 401-406.
- Ryan, J., "Generalized Lists and Other Extensions," *APL Quote-Quad*, June 1971, 8-10.
- , "Secure Applications Within an APL Environment," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 407-414.
- Stohr, E. A., "Simulation of Some APL Operators," *Report LR-16*, Center for Research in Management Science, University of California, Berkeley, Calif., Feb. 1971.
- Thurber, K. J., and J. W. Myrna, "System Design of a Cellular APL Computer," *IEEE Transactions on Computers*, April 1970, 291-303.
- Van Hedel, H., "An APL Batch Processor," *Colloque APL*, IRIA, Rocquencourt, France, Sept. 1971, 339-364.
- Vasseur, J. P., "Extension of APL Operators to Tree-Like Data Structures," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 457-464.
- Wiedman, C., "The Case for APL Standards," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 581-584.
- Williams, J., "Conditional Branch APL Compiler," *APL Quote-Quad*, June 1971, 5-6.
- Zaks, R., D. Steingart, and J. Moore, "Firmware APL Timesharing System," *SJCC*, 1971, 179-190.
- Zaks, R., "Microprogrammed APL," *Conference Digest of International IEEE Computer Conference*, Boston, Sept. 1971, IEEE, pp. 192-194.

Miscellaneous: Of Interest to the APL-Programmer or APL-Related Information

- Alercia, R., R. Swiatek, and G. M. Weinberg, "Every Little Bit Hurts: Saving Money by Saving Space in APL," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972, pp. 118-119.
- Bork, A. M., "Graphics in APL," in *Proceedings of the Fourth International Users' Conference*, Atlanta, June 1972, pp. 33-36.
- , "APL as a Language for Interactive Computer Graphics," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 50-57.

Bibliography

- Casarosa, V., and S. Trumpy, "Conversational Building and Display of Solid Objects," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 79-84.
- Charmonman, S., "A Comparison of the Structures of APL, Fortran, Algol, and PL/I," *APL Quote-Quad*, Jan. 1970, 2-4.
- Charmonman, S., and S. E. Bell, "A PL/I Batch Processor for APL," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972, pp. 123-134.
- Creveling, C. J., "APL and Its Entry Into the World," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 117-123.
- Dayton, M., "A Plotter of APL," *APL Quote-Quad*, June 1971, 13.
- Embley, D. W., "APL Graphics," unpublished M.S. thesis, Dept. of Computer Science, University of Utah, Provo, 1971.
- Gerhart, S. L., "Verification of APL Programs," unpublished Ph.D. Thesis, Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh, University Microfilms, Ann Arbor, 1972.
- Greenberg, S. G., and C. I. Johnson, "An Interactive APL Graphics System," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972, pp. 37-44.
- Hagerty, P. E., "On the Portability of APL Packages," *APL Quote-Quad*, Jan. 1973, 22-27.
- Haralson, K. N., "Useful APL Defined Functions," *Report No. TR-00.2409*, IBM Poughkeepsie Laboratory, Poughkeepsie, N.Y., Feb. 1973.
- Higgins, J., and A. Kellerman, "Management of APL Time-sharing Activities," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972, pp. 103-114.
- Hunter, G. T., "Found in an Empty APL Workspace," *APL Quote-Quad*, Jan. 1971, 5-8.
- Iverson, K. E., "APL in Exposition," *Report No. 320-3010*, IBM Philadelphia Scientific Center, Philadelphia, 1971.
- Jones, A. L., and R. H. Katyl, "A Graphical Input Capability for APL," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 221-227.
- Jones, A. L., and W. H. Niehoff, "APL/360 Graphpak-Interactive Graphics in the Terminal Environment," *Report No. TR-01.1502*, IBM Systems Development Division, Endicott, N.Y., Aug. 1971.
- Katzan, H., "A Prose Glossary of APL," *Computers and Automation*, Aug. 1970, 39-42.
- , "Representation and Manipulation of Data Structures in APL," in *Proceedings of a Symposium on Data Structures in Programming Languages*, ACM, Feb. 1971, pp. 366-397.
- Kupka, I., and N. Wilsing, "An APL Based Syntax Form for Dialog Languages," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 269-274.
- Lathwell, R. H., and J. E. Mezei, "A Formal Description of APL," *Colloque APL*, IRIA Rocquencourt, France, Sept. 1971, 181-215.
- Macon, H. P., "A Survey of APL Compatible Terminals," *APL Quote-Quad*, Oct. 1971, 12-20.

Bibliography

- Mahanti, B. K., and M. E. Sandfelder, "Interactive Data Processing Application Implementation Using APL," *APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973, pp. 307-312.
- McDonnell, E. E., "The Variety of Alternative Definitions of a Single Function," in *Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973, pp. 4.1-4.23.
- Moruzzi, V. L., "APL/Fortran Translations," *Report No. RC-3644*, IBM Corp., Yorktown Heights, Dec. 1971.
- Niehoff, W. H., and A. L. Jones, "An APL Approach to Interactive Display Terminals," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972, pp. 23-32.
- Penfield, P., "Security of APL Application Packages," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972, pp. 119-122.
- Rault, J. C., and G. Demars, "Is APL Epidemic? or a Study of its Growth Through an Extended Bibliography," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972, pp. 1-22.
- Riedy, J. E., "Some Solved Problems in Intermediate APL," *Report No. TR-01.1544*, IBM Systems Development Division, Endicott Laboratory, Endicott, N.Y., June 1972.
- Rose, A. J., "Teaching the APL/360 Terminal System," *Proceedings of Share XXXII*, Part II, March 1969, 543-554.
- Rowley, J. R., "An APL-ASCII Graphics Terminal," *APL Quote-Quad*, Jan. 1973, 20-21.
- Ryan, J. L., "Enhanced Interaction for an APL System," in *Proceedings of the Fourth International APL Users' Conference*, Atlanta, June 1972.
- Sant, D., "The MRX 1240 Communication Terminal and 1270 Transmission Control Unit," *APL Quote-Quad*, June 1971, 13.
- Schreiber, G. P., and R. Polivka, "Experiences and Observations with a Self-Teaching Course in APL," *Colloque APL*, IRIA, Rocquencourt, France, Sept. 1971, 77-90.
- Thanhouser, N., and L. Koeningsberg, "A Graphics System for APL Users—APL/Graph-II," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 529-543.
- Weiss, C., "Communications Between Users in a Dispersed APL Network," in *Proceedings of the Sixth International APL Users' Conference*, Anaheim, May 1974, pp. 562-567.

Collections

- APL Congress 73*, P. Gjerlov, H. J. Helms, and J. Nielsen, eds., American Elsevier Publishing Co., New York, 1973.
- APL Quote-Quad*, an informal publication for APL-users, published quarterly.
- Proceedings APL Conference*, York University, Downsview, Ontario, Canada, Jan. 1972.
- Proceedings of the APL Users' Conference at S.U.N.Y.*, Binghamton, N.Y., July 1969.
- Proceedings of the Fifth International APL Users' Conference*, Toronto, Canada, May 1973.

Bibliography

Proceedings of the Fourth International APL Users' Conference, Atlanta, June 1972.

Colloque APL, IRIA (Institut de Recherche d'Informatique et d'Automatique), Rocquencourt, France, Sept. 1971.

Proceedings of the Sixth International APL Users' Conference, Anaheim, May 1974, Coast Community College District.

5.2.4

"Things Go Better with APL"

Claire Stegmann

Think

May/June 1975

Tom Ryder, a Salt Lake City businessman, picks up his weekly payroll at a computer service bureau. He stops for a Coke. Then it's back to his office in his Citroën for an appointment with an IBM salesman. He's considering a computer of his own; and he'd like to have it by the time his son, just graduating from M.I.T., joins the firm.

Ryder might be surprised to learn that everything he has just done is involved, in some way, with the computer time-sharing language, APL.

by Claire Stegmann

- The customer service bureau he uses, which offers APL service, is operated by Management Systems Corporation, a wholly owned subsidiary of the Church of Jesus Christ of Latter-day Saints. The Mormon Church, as it is more commonly known, also uses APL in many of its own operations.

- Both The Coca-Cola Company and Citroën are increasing users of APL.

- The IBM salesman, to help Ryder tailor a computer to his particular needs, will almost certainly consult a terminal in his branch office for design analysis and to find out what IBM products and services are available, and where. That terminal is tied into the Data Processing Division's HONE (Hands-on Network) system, which makes significant use of APL.

- If Ryder decides to order, the System Products Division's worldwide commodity planning system, based on APL, will enable SPD to provide the earliest possible delivery date.

Incidentally, Ryder's son, back at M.I.T., uses APL to do his homework. And he and his father share another APL-based activity: the National Football League statistics they follow so closely are compiled with the language.

Ryder and his son are fictitious. But the APL applications are real. They illustrate the versatility of *A Programming Language*, which has come into its own as an important product for the computer user.

For most of its 13 years, APL's primary habitat has been the classroom, where, as an extension of simple mathematical notation, it has helped high

school and college students learn not only algebra, but also economics and electrical engineering. In Atlanta, students use it to design and compute the cost of making cabinets.

IBM Fellow Dr. Kenneth Iverson, who invented APL as a precise way to describe data processing, explains: "Every field of human knowledge that deals in quantity—mathematics, physics, economics—has a special vocabulary that must be put into a general-purpose language for computer use. APL defines these terms by their simple mathematical formulas, using a number of symbols, some of them (for example, +, -, ×, ÷) already familiar to every school child. Thus, as a student learns APL, he or she is also reinforcing knowledge of a particular discipline."

Dr. Iverson believes that APL's versatility as a teaching aid has barely been touched. Much of his work as a Fellow is devoted to helping educators in many fields study its application to their particular specialties.

But, according to Dr. Truman Hunter, APL manager, because APL is relatively simple to learn, versatile, and interactive (the user and the computer "converse" during the processing of a job), it is becoming more and more attractive to business for management decisions that need frequent updating and quick response.

"This has been especially true," says Adin Falkoff, Dr. Iverson's longtime collaborator, "since the introduction of an APL program product called the APL Shared Variable System. This makes it efficient and convenient to work with files and data bases." Both APLSV and APL/CMS (Conversational Monitor System), recently introduced by IBM, enable the APL user to do a greater variety of jobs on a much larger scale than before.

"It is the analytic power of APL, the language," says Dr. Hunter, "plus the easy access to large stores of information, that I think is going to make it increasingly important to the businessman. Now, more than ever, he can



THINGS GO BETT



develop applications and get results while they are still timely."

APL and its conversational cousin BASIC are by far the most widely used languages specifically designed for terminal use, which allows the user to bypass such time-consuming steps as going to, and queuing up in the computer room. Of the two, BASIC is older and more widespread. But, according to Curt Bury, DPD senior product administrator, APL's inherent power is extending its effectiveness to areas beyond the scope of BASIC. An IBM operations planner agrees: "We planners have to sort out a great many complex relationships in a very short time. APL is the only game in town for this kind of thing."

IBM's accelerated use of APL since it was first introduced in the mid-Sixties, on the IBM 1130 and System/360 Model 40 and above, reflects the recognition of the language by data processing people as an effective way of getting applications written faster. "In the past several years," Data Processing Division President John F. Akers told a GUIDE user-group last year, "we have gone from 4,000 to 16,000 APL users within IBM. And the balance has shifted from scientific to commercial users."

Every IBM division relies on the language in some way. DPD's HONE system, which connects U.S. branch offices to a central data base and makes heavy use of APL programs, made a tremendous contribution to field productivity. In what is possibly the largest single use of APL, SPD's commodity planning system integrates production schedules so efficiently that decisions that used to take two months can now be made within a matter of minutes. The General Products Division depends on APL to write test cases for its new 3850 mass storage system. Many aspects of the business of IBM in Europe, according to Yves Le Borgne, advisory APL support representative in Europe, are managed from APL terminals. IBM World Trade Americas/Far East Corporation has used APL to develop applications for both its oper-

ER WITH APL

ating plans and its strategic planning.

Office Products Division engineers use APL in the development of Copiers and magnetic keyboard typewriters. OPD, incidentally, has another interest in the language. It makes the APL Selectric typing element.

In a move that brought APL into the mainstream of IBM software production, the APL group was transferred last year from OPD's Philadelphia Scientific Center into SDD. A design group, consisting of Iverson, Falkoff, Richard Lathwell, and a small staff, remain in Philadelphia. The implementation group, under L. A. Morrow, joined the SDD programming center at San Jose. "We consider APL one of the primary interactive environments for current and future systems," says R. E. Ruthrauff, SDD manager of language and data facilities advanced system design.

One strong advocate of interactive computing, and of APL in particular, is Professor Ephraim McLean of U.C.L.A.'s Graduate School of Man-

agement. He introduced APL to the business school four years ago. Today, it is a requirement for all entering M.B.A. students.

This summer, with an IBM grant, Professor McLean will offer a course on interactive computing to a limited number of faculty members from other business schools. "I have been surprised at both the number and the variety of requests for applications," he comments. "They come from all over—a Jesuit priest, a woman marketing professor, a faculty member from Howard University, several individuals from both French and Mexican universities, and a surprisingly large number of business school deans and department chairmen."

Next, McLean hopes to do a study on the increased effectiveness of APL users. "I think a study will find that APL is much more productive," he says. "For instance, I am convinced that with APL a programmer can accomplish in 10 instructions and a few hours what

might take 200 instructions and several days in COBOL or FORTRAN."

One of the beauties of APL, says Dr. Iverson, "is that you have to learn so little to get started, like a child, or a grownup for that matter, learning a language. You tell a child two or three things and he builds his vocabulary gradually. With APL, you gradually accumulate a knowledge of the symbols as you do more and more work."

Several clerks and secretaries in town halls near Paris would probably agree. They have never seen a computer but are using terminals, and APL, to do payrolls they formerly did by hand. "Here in Europe," says Yves Le Borgne, "we are confident that APL has a great deal to offer our customers—especially now, with the economic crisis, when there is a need for better management to grasp a situation and react to a changing environment quickly."

In May, Iverson receives the annual Harry H. Goode Memorial Award from the American Federation of Information Processing Societies, for the impact APL is expected to have on programming language theory and practice in the future. The setting would be the envy of many youngsters: Disneyland, in California. But it probably wouldn't impress the son and daughter, both under nine, of IBM senior programmer Corey Fair in East Fishkill. Since the Outstanding Contribution Award winner (for his part in development of the SPD commodity planning system) brought a terminal home, his kids don't watch Walt Disney on TV much anymore. Instead, they compete for time on the terminal, when allowed, to practice APL. ■

Here's how it works

One of the virtues of APL is that it is interactive, that is, a person can "converse" with the computer during the processing of a job. Here is a simple example of how such a conversation might go:

Let's say a businessman wants to examine the feasibility of taking out an \$8,000 loan at 9½ percent interest. He wants to see what his monthly payment would be each month if he repaid the loan over an eight-year period. Also, for income tax purposes, he'd like to know what his annual interest would amount to each year.

Now, let's assume that the APL program to perform this function is identified in the computer as ME. The businessman begins by typing:

M E

The computer responds:

ENTER PRINCIPAL, ANNUAL INTEREST (AS A DECIMAL), AND NO. OF YEARS

The businessman types:

8 0 0 0 . 0 9 5 8

The computer comes back:

ENTER MONTH AND YEAR THAT FIRST PAYMENT IS DUE (E.G. 8 73 FOR AUG, 1973)

The businessman types:

5 7 5

The computer prints out:

PAYMENT AMOUNT IS 119.29

END OF YEAR	ANNUAL INT.	PRINCIPAL REMAINING	END OF YEAR	ANNUAL INT.	PRINCIPAL REMAINING
1975	494.05	7539.73	1980	340.52	2984.95
1976	684.30	6792.55	1981	232.24	1785.71
1977	610.15	5971.22	1982	113.23	467.46
1978	528.63	5068.37	1983	9.28	.00
1979	439.02	4075.91			

Think

May/June 1975

Think is published by International Business Machines Corporation, Armonk, New York 10504. Frank T. Cary, Chairman of the Board. It is produced by IBM Corporate Communications, Jane P. Cahill, IBM vice president, communications. Think is distributed to all IBM employees in the U.S.A., and to a number of IBM employees abroad.

Chet Hansen, Editor; Woody Klein, Managing Editor; Benito P. Carucci, Art Director; Quentin W. Johnson, Manager, Publications Production; Angelo M. Donofrio, Special Features Editor; Ed Grimm, International Editor; Howard K. Janis, Senior Editor-at-Large; Geoffrey D. Austrian, Ernie Bauer, Editors-at-Large; Claire Stegmann, Senior Editor; Ruth S. Baker, Associate Editor; Walter H. Wszulkowski, Photo Editor; Elizabeth W. Brown, Circulation. The word Selectric and all other similarly capitalized marks identify products manufactured by the IBM Corporation and distinguish them from similar products manufactured by others.

Credits: Cover: Bob Cook—Shostal Associates. Joe Bairnell: 40, 41, H. von der Becke: 14, 16 top, 17 bottom, Henri Dauman: 22, 25, 27, 29 left, 30, 33, 39, Paul Giovanopoulos: 9, Mike Hirst: 36, IBM Photo: 35, IBM Pressfoto: 16 bottom, 17 top, Mark Joseph: 13, Bruce McCalister: 38 left, Don Miller: 38 right, Eric Schnaal: 29 right, Rene Schumacher: 46, Bob Smallman: 24, 28, 32, Bob Sirovink: 37, Murray Tinklemann: 5, Ted Treshinsky: 23, 31, Walter Wszulkowski: 12.

5.3

APL Programming Assignment Instructions

INTERACTIVE COMPUTING FOR MANAGEMENT EDUCATORS

Graduate School of Management
University of California, Los Angeles

APL PROGRAMMING ASSIGNMENT

In applying to this program, each of you described, in varying degrees of detail, the problem or application that you wished to program in APL during the second week of the course. The resulting programs are to be published as a UCLA monograph.

In order for this aspect of the course to be successful, it is vitally important that you prepare for this exercise prior to arrival. Although you of course know nothing about APL, it is reasonable to expect that you should have a good understanding of the application which you intend to program. Toward this end, the following is a list of items that, to the maximum extent possible, should be prepared prior to arrival and brought with you.

1. A concise statement of the problem, application, game, student exercise, or other use you wish to make of the computer in an interactive mode. This should include how you expect the program to be used (as drill-and-practice, gaming or simulation, problem-solving, etc.), by whom (e.g., beginning undergraduates, managers, advanced M.B.A.'s, etc.), and to what end (i.e., what is the object of the program; e.g., to attack problems too complex for hand solution, to reinforce materials already presented by traditional means, etc.). Also included in this should be the formula, algorithm, or other logic required to arrive at the solution, result, etc. This is similar to what you put in your application questionnaire, but unfortunately most of them were rather sketchy.

2. A general description of how the program works and what the user can expect from it. This statement, which will naturally be refined after you arrive, will be the kind of documentation you would like to see accompany your program in published form. It should include a specification of the inputs, parameters, and other variables that are necessary on the part of the user in order to make the program work. For example, in a simple program to calculate the terminal value of \$1.00 invested at compound interest, i , for N periods, the user must be told that he needs to supply both i and N .

3. Data which can be used to test the program. These data should be of two types: one should be sample or typical data such as would be used to illustrate the use of your program; the other should be a wide range of data designed to test all possible conditions (e.g., what if a user tries to run the program with zero — or an unreasonably large number? what happens if some input is missing? etc.) In other words, try to think of all the ways the program might be used — correctly or incorrectly — and then devise data to test these possibilities.

For some of you with little or no experience in computing, you may feel that it is extremely difficult for you to compile such information prior to the beginning of the course; but it is important that you try, for without having carefully thought through the problem that you wish to undertake, you may find that the two weeks are over before you have half begun.

6.0
PROGRAM ADMINISTRATION
AND
PROMOTIONAL MATERIALS

6.1

Program Announcement Brochure

**INTERACTIVE COMPUTING
FOR
MANAGEMENT EDUCATORS**

June 29 — July 11, 1975



**GRADUATE SCHOOL
OF MANAGEMENT
UNIVERSITY OF CALIFORNIA
LOS ANGELES**

THE PROGRAM

An intensive two-week program, supported by a grant from the IBM Corporation, will be conducted by the Graduate School of Management at UCLA from June 29 to July 11, 1975. Its purpose is to introduce faculty members from business schools and colleges throughout the country to the use of interactive computing in management education.

Unlike other programs which have been offered elsewhere and which have dealt with the use of FORTRAN and computer batch processing, this new program will focus upon the advantages offered by time-sharing languages for the teaching as well as for the practice of management. Specifically, it will cover the following four areas:

1. The use of conversational, interactive computing in business.
2. The APL programming language as a new and powerful means for providing this interactive capability.
3. The methods for teaching APL to business students.
4. The ways in which APL can be used for teaching accounting, finance, operations research, and other management subjects.

For many faculty members who have made little or no use of computing in their teaching, the mastery of a computer language represents a substantial hurdle. They feel that in order to use computers they must first learn a batch processing language, with all the accompanying intricacies of the job control language, before they can move on to a time-sharing language like APL.

Such views are unfortunate, for they preclude many teachers from taking advantage of the assistance of the computer because of this perceived high start-up cost. Therefore, it is the intention of this program to help overcome this barrier and to make the transition from non-use to informed use as easy and painless as possible.

Toward this end, the use of APL is particularly appropriate, for it is both simple to learn and yet powerful to use. Rapidly growing in popularity, this conversational, interactive computer language is being used in such dissimilar industries as manufacturing, insurance, and transportation. It is also widely available through commercial time-sharing services.

As a prerequisite to attendance, each participant will be expected to define a potential application in his or her own area of specialization prior to arrival. (This involves specifying a learning objective, structuring the problem or exercise, and gathering the data necessary to test and illustrate the use of the resultant program.) After arrival, this application will be coded into APL. In this way, not only will the participants be able to acquire a deeper understanding of APL relevant to their own discipline, but they will also have produced working programs (or series of programs) which can be put into use immediately upon return.

These programs, identified with the author's name and accompanied by a description of their use, will be published as a monograph after the completion of the program. Copies will be furnished to all participants, to the IBM Corporation, to member schools of the American Assembly of Collegiate Schools of Business (AACSB), and to other interested parties.

Participation is limited to 24 faculty members. The cost of the program, as well as all of the living expenses of the participants while they are at UCLA, will be borne by the sponsoring grant from IBM. Travel costs to and from Los Angeles are not included.

TOPICS TO BE COVERED

1. **Introduction to Computing Concepts**
A brief overview of computer terminology and technology, covering the general functioning of computer systems, especially those supporting time-sharing services.
2. **Introduction to APL**
A description of the concept and background of APL, including a demonstration of its various features.
3. **The APL Functions**
The complete set of APL primitive functions, showing their use in conjunction with scalars, vectors, and matrices.
4. **User-Defined APL Functions**
The writing of programs, or, in APL terminology, the defining of one's own functions.

5. **Systems Commands and Workspace Management**
The concepts of public and private libraries, the manipulation of workspaces and their contents, and the functioning of the various systems commands.
6. **Report Generation**
The preparation of reports, both in a tabular and a graphical mode.
7. **File Systems**
The concepts of shared variables (SV) and file systems (such as STSC's APL Plus File System).
8. **Programming Techniques**
A number of APL programming procedures and techniques, exploring the best of current practice.
9. **Application of APL to Business**
The applications of interactive computing to diverse business problems, drawing upon the experience of the instructional staff, outside speakers, and the program participants.
10. **APL as an Aid in Management Education**
The use of APL to help teach accounting, finance, operations research, and other disciplines; also, the participants' development of their own applications.
11. **Methods for Teaching APL**
The teaching strategies used at UCLA over the last four years in introducing APL to well over a thousand M.B.A. students.

THE FORMAT

The program will begin on Sunday, June 29th, and end on Friday, July 11th. A reception and welcoming dinner will be held on Sunday evening, with classes beginning the following morning.

All classes will be held at the Graduate School of Management (GSM) from 8:30 A.M. to 5:30 P.M. each day, with the evenings and middle weekend free for studying or recreation.

The participants will be split into two groups of twelve each. One group will attend a lecture while the other works in the computer laboratory. Every two hours the two groups will switch. This allows for small class sizes and maximum individual attention.

Of the three instructors associated with the program, at least two will be lecturing or assisting in the lab at all times. Over the two-week period, each attendee will have 34 hours of lecture and 44 hours of laboratory time, including the programming of his or her own application program.

WHO SHOULD ATTEND

The program is designed for those business school faculty members who wish to enhance their teaching skills by applying the power of interactive computing to their area of specialization. There is no need to have had any previous experience in computing. Therefore, those who have never used computers, as well as those who had had some programming experience, are invited to apply.

The selection of the 24 participants will be governed, at least in part, by the desire to obtain a broad cross section of disciplines, functional areas, and representation from among AACSB member schools. In the event the program is oversubscribed, all applications for attendance will be retained in case a subsequent offering is possible.

Requests for applications should be sent by March 15th to:

Dr. Ephraim R. McLean
Graduate School of Management
University of California
Los Angeles, California 90024

THE FACULTY

EPHRAIM R. McLEAN, Program Coordinator

Professor McLean is presently the Director of the Center for Information Studies, and was formerly the Chairman of the Computers and Information Systems Curriculum Committee, both at GSM. Instrumental in bringing APL onto the UCLA campus, he taught the first regularly-scheduled course in APL at UCLA and has taught it every year since. In conjunction with Professor Siler, he has taught APL to over a thousand graduate students. When the widely-used book *APL: An Interactive Approach* (which is to be the primary text for the program) was recently revised in preparation for the second edition, Professor McLean was retained by John Wiley & Sons as a consultant on the revision.

KENNETH F. SILER

Professor Siler, the other faculty member associated with the program, is the current Chairman of the Computers and Information Systems Curriculum Committee and has worked closely with APL for the last two years.

JAMES W. SCHENCK

Mr. Schenck, the third member of the instructional staff, was formerly the Coordinator of Computer Services for GSM and is currently a doctoral student in the School. He is one of the co-authors (with J. W. Buckley, M. R. Nagaraj, and D. L. Sharp) of *Management Problem-Solving with APL*, (Los Angeles: Melville Publishing Co., 1974), another text in the program.

ACCOMMODATIONS AND FACILITIES

All of the participants will be housed at the Holiday Inn in Westwood, immediately adjacent to the UCLA campus. A special bus will furnish transportation between the motel and GSM. Meals will be provided at the Holiday Inn and at the UCLA Faculty Center, or participants may choose to dine at one of the many restaurants in Westwood.

The lecture and laboratory sessions at GSM will be conducted "on-line." For lectures, the air-conditioned classroom will be equipped with a computer terminal and with closed circuit television so that everyone in the room can follow the output directly.

The computer laboratory, a carpeted, sound-deadened room equipped with 12 computer terminals, will be reserved for the exclusive use of the participants during the program. In addition, a number of other terminals located within GSM will be available on a first-come, first-served basis. A portable terminal will be placed in the Holiday Inn so that participants may work in the evening without having to return to GSM.

With regards to the terminals themselves, GSM has a wide assortment, including A.J. 841's, IBM 2741's, G.T.&E. 5741's, and Tektronix 4013 graphic terminals. The main support computer is the UCLA Campus Computer Network's IBM 360/91 with four million bytes of main storage. Two commercial companies, Proprietary Computer Systems and Scientific Time Sharing Corporation, are available as back-up, if necessary, to insure uninterrupted service.

6.2

Invitation Letter to
Business School Deans



OFFICE OF THE DEAN
GRADUATE SCHOOL OF MANAGEMENT
LOS ANGELES, CALIFORNIA 90024

15 January 1975

Dear Dean :

It is my pleasure to invite your participation in a program that I know will be of interest to you.

With the use of computers expanding on every front, it is difficult for those of us in the academic world to keep up with these developments, let alone employ them in our own educational programs. Significant among these developments is the use of conversational, interactive computing. With this capability being increasingly employed in government and industry, it is unfortunate when the students we are graduating are not familiar with this powerful aid to management. But before our students can use this technology, our faculty must first become comfortable with it. This brings me to the purpose of this letter.

Next summer, a program is being offered at the Graduate School of Management here at UCLA on the subject of Interactive Computing for Management Education. The enclosed brochure explains this program in detail.

Made possible by a generous grant by the IBM Corporation, this two-week program provides an opportunity for 24 faculty members to attend an intensive workshop in the use of APL for management education. All of the expenses connected with attending will be borne by the IBM grant, with the exception of the travel expenses to and from Los Angeles. In this way one or more of your faculty members can substantially upgrade his or her skills by participating in a program which, if taken commercially, would cost the individual several hundred dollars.

In offering this program, we believe that we bring to it a perspective and a measure of experience that is unique in the country. The program will be directed by Dr. Ephraim R. McLean who was instrumental in bringing the interactive computer language APL to the UCLA campus several years ago and is widely regarded as one of the leaders in the country in its application to management education.

Page 2

As is described in the brochure, there are only 24 openings in the program, so it is possible that there may not be room for everyone who wishes to attend. May I urge you therefore to bring the enclosed material to the attention of those of your faculty whom you feel would most benefit by attending and have them forward their application to Dr. McLean. It is not necessary that they have any previous experience with computing or that APL is currently being used on your campus.

I am excited by the promise of this program and hope that you share my enthusiasm. May we hear from you, or from your designated faculty member, in the near future?

Sincerely,

Harold M. Williams
Dean

Enclosure

6.3
Publicity

COMPUTING NEWSLETTER

FOR SCHOOLS OF BUSINESS

J. DANIEL COUGER, EDITOR

PUBLISHED BY THE COLLEGE OF BUSINESS ADMINISTRATION • UNIVERSITY OF COLORADO • COLORADO SPRINGS

Vol. VIII, No. 7

March, 1975

ADVANCES IN INTERACTIVE COMPUTING

"Today the computer is as natural an educational tool as paper and pencil for the Tuck students, and their ability to solve operational problems with quantitative computer techniques is highly attractive to prospective employers." **Tuck Today**

Seventy-five percent of AACSB schools responding to our 1974 survey have time/sharing capability. That compares to 43 percent in the 1970 survey. What are the advantages of T/S over batch processing? Each year our March issue is dedicated to that question.

We begin our comparison with an update of Dartmouth's T/S approach. During our 1966 survey, 11 schools were visited. Of those 11, Dartmouth's Amos Tuck School of Business proved to be a barometer of T/S potentialities. We've revisited that campus during each of our triennial surveys. As in previous visits, Tuck revealed some characteristics which can be emulated by other schools—even schools with very modest computing budgets.

Objective in Computer Education

When asked the principal objective in educating Tuck students about the computer, Dean John Hennessey responded, "to acquaint students with the power of the computer as an aid to decision making analysis—an aid to thought—for the practicing manager." Dean Hennessey then expanded this concept: "We want to make sure our students are not in any way frightened by the computer. . . . We hope that at least students will become familiar with the idea of computers as a natural part of the managerial environment. Also, we want students to know how computers will be directly and indirectly useful to them in their work here at Tuck."

Motivating the Faculty to Computer Use

For a decade, Tuck's faculty has been highly computer-oriented. In the early 1960's Dean Karl Hill acquired an IBM 1620 and installed it in the Tuck School basement. His approach was to make computing resources available—both equipment and advisors. Gradually the faculty began to utilize those resources. I asked Dean Hill's successor the approach to motivating new faculty to acquire a computer capability. Dean Hennessey replied, "When a new faculty member arrives on our campus he recognizes that everybody else is using the computer as a natural part of their

environment . . . almost as easily as telephones are used. I use it myself—it's required of me in order to access the management information system which President Kemeny has installed as a part of Dartmouth's administration. . . . It's a rare faculty member who wouldn't want to be a part of our computer activities."

Asked if a requirement for faculty proficiency was then implicit, rather than explicit, Dean Hennessey agreed. "There are far more persuasive and lasting ways to influence people than telling them what to do. . . . The mandate from their colleagues and their students to use the computer is much more powerful. For example, a new faculty member in finance is quickly going to find that other faculty ask the students to use the computer in several ways. So, faculty in any kind of quantitative area would take it up immediately. A new faculty member in a non-quantitative area finds fairly quickly that he would want to be acquainted enough with computers to understand what students mean when they come to him and say 'I want to do an independent study project and I'm going to use a computer the following way in your field.'"

What's New at Dartmouth?

All three AACSB surveys indicated the importance of faculty participation for achieving a viable computer curriculum. Even an extremely capable instructor in the core computer course will not be able to convince students of the value of the computer—that value must be demonstrated through assignments in subsequent courses. Dartmouth has shown the way to create such an environment. Computer proficient faculty in the functional area courses encourage students to use the computer to acquire greater depth of knowledge in each subject area.

Such an environment is not dependent upon, but is facilitated by time/sharing. Ease of access is a principal motivator—to both faculty and students. Page 2 will describe some T/S innovations at Tuck School.

SUMMER FACULTY EDUCATION PROGRAMS

UCLA

A two-week faculty education program, supported by a grant from the IBM Corporation, will be conducted by the Graduate School of Management at UCLA from June 29 to July 11, 1975. Its purpose is to introduce faculty members from business schools and colleges throughout the country to the use of interactive computing in management education.

Specifically, it will cover the following four areas:

1. The use of conversational, interactive computing in business.
2. The APL programming language as a new and powerful means for providing this interactive capability.
3. The methods for teaching APL to business students.
4. The ways in which APL can be used for teaching accounting, finance, operations research, and other management subjects.

No Cost To Participants

Participation is limited to 24 faculty members. The cost of the program, as well as all of the living expenses of the participants while they are at UCLA, will be borne by the sponsoring grant from IBM. Travel costs to and from Los Angeles are not included.

Who Should Apply

The program is designed for those business school faculty mem-

bers who wish to enhance their teaching skills by applying the power of interactive computing to their area of specialization. There is no need to have had any previous experience in computing.

Selection of the 24 participants will be governed, at least in part, by the goal to obtain a broad cross section of disciplines, functional areas, and representation from among AACSB member schools.

Although the cutoff date is March 15, if you immediately telephone Dr. Ephraim McLean, it is possible to have your application considered, (213) 825-7982.

SUNY-BINGHAMTON

Although not formulated at this time, another 3-week Summer Institute in APL will be offered at SUNY-Binghamton in 1975.

The institute is largely motivated by numerous requests from APL users and potential APL users from academia and industry for various APL techniques, APL applications, and APL curricular development topics. The three-week syllabus involves: Elementary APL, APL packages, and Advanced APL.

Sessions are suitable for anyone from academia or industry. For complete details, session outlines, fee sheets, and registration forms contact Geraldine McDonald, Computer Center, SUNY-Binghamton, Binghamton, New York 13901.

USE OF APL IN MANAGEMENT EDUCATION

Prof. Ephraim McLean (UCLA) presented a paper of this title at the Sixth International APL Users Conference in May 1974. His school now teaches APL in the introductory data processing course in the School of Management. The following excerpts from his paper support the school's decision.

During the school year 1971-72, UCLA, in collaboration with Scientific Time Sharing Corporation, installed the latter's APL Plus system. This

In reviewing use of APL with other business educators across the country, the question which most commonly arises is: "But why don't you use BASIC?" UCLA first hoped that both would be available. The campus computer center agreed to support only one. The school selected APL, for the following reasons:

1. APL is easy to learn. Although the special symbols and character set are a little strange at first, they are quickly mastered and their power and conciseness are soon appreciated.

2. APL is unique. APL's powerful operations and matrix-handling ability set it apart from other languages. But this supposed "disadvantage" arises only if students do, in fact, go on to some other language. However, most students find little reason to move on to some other language. And again, UCLA's emphasis is on developing problem-solvers and decision-makers—not journeymen programmers.

3. APL is widely available. APL has been implemented, in one way or another, by almost all computer manufacturers. It is true that the number of different machines for which BASIC interpretive compilers presently exist is greater than for APL. Also, because of APL's greater power and correspondingly greater memory requirements, it is unlikely that it will ever be implemented on any of the mini-computers. Finally, in terms of the number of service bureaus, again far more are offering BASIC than APL. However, the average user typically does not care whether a language is available on five machines or fifty; his main concern is, "Is it available to me?" The answer, in most major cities in the U.S., is yes; and the availability of APL appears to be growing.

4. APL has large libraries of programs. The number of APL programs in the public domain is quite large. It is true, of course, that they are nowhere near as numerous as those in BASIC, but the APL libraries are growing rapidly, fueled by such major users as IBM itself and by American Airlines' comprehensive APL-based management information system.

Although BASIC might have an edge at this point, APL is gaining so rapidly that soon the number of available programs, as well as the number of companies offering time-sharing services, will result in differences that are more apparent than real.

Business School Applications

1. Finance

Capital budgeting is one area in which a fair amount of work has been done. Such variables as cost of capital, differing depreciation schedules and tax treatments, variable income streams, and provisions for risk can all be taken into account and tested for their relative sensitivity.

Two examples of specific assignments which have been undertaken involve computer acquisition and property tax evaluation. In the first case, the frequently-encountered problem of whether to lease or buy computer hardware was studied. Discount rates, equipment configurations, expected operating schedules (e.g., one shift or two), and charging and pricing algorithms were all varied to arrive at the most desirable solution. In the second case, a real estate company's capital asset management was partially dependent upon local property tax rates, so the extent of this sensitivity was analyzed.

Another application in the financial area is the use of corporate financial data from such sources as Standard and Poor. Financial ratio analysis across companies or over time has been undertaken. Also, various types of financial forecasting and trends analysis have been studied through the use of regression techniques and time series analysis. Finally, in the field of investment portfolio selection, the study of risk by means of beta factor analysis has been done.

2. Statistics.

With the availability of STATPACK2 from the University of Alberta, as well as other statistical tools in the APL public library,

system, now in use by over ten other university computing centers, contains a number of APL enhancements. Among them are additional systems commands; powerful formatting functions for outputting data; a shared file system permitting APL, for the first time, to handle significant-sized files; and an APL-batch file interface allowing high speed input/output devices to be used instead of the requirement of having to enter all data through the terminal. The importance of this file handling capability will become more apparent when some of the applications are discussed.

a variety of problems can be investigated. In a course on measurement theory, analysis of variance and problems of data transformation were studied. For statistics classes, both classical statistics problems (e. g., determining sample sizes and tests of significance) as well as Bayesian statistics (e. g., decision tree analysis) have used the students' APL background to good effect. Correlation, regression, and analysis of variance applications of various types are undertaken; and, in decision theory, much of the calculation drudgery has been eliminated through the use of APL modules without sacrificing the mastery of the concepts.

3. Planning and Control.

Of the several courses within the school that focus on planning and control, one is built almost entirely around the use of APL. Student teams have developed planning models, each in a setting of their own choice. In creating these planning systems, stress is placed on the importance of the user-interface and upon such issues as data and file management, security features, and real-world feasibility.

4. Operations Research.

By using a variety of programs in the public library, students are able to attack nonlinear and integer programming problems which, because of the need to solve a linear programming subprogram for each iteration, are usually so long that only the simplest are assigned. Now, with APL routines to handle the L. P. calculations, students can devote their attention to understanding the underlying nonlinear or integer algorithm. Resource allocation decisions and transportation and shortest route problems are regularly assigned and students can perform a number of sensitivity analyses on constraint variables and coefficients of the objective functions.

Some simulation and model building is also being done, but the interpretive nature of APL makes the execution of problems of this sort fairly costly if the number of iterations is high. It is in instances such as these that the quality of the coding becomes critical. Poorly or inefficiently written programs can be intolerably expensive, while the same program carefully written to take full advantage of the power of APL can be much more cost effective.

5. Management Gaming.

Probably the most extensive use of APL is in connection with the playing of the general-purpose management game which is an integral part of the M. B. A. students' first year of study. Externally, it is quite similar to the IBM APL Business Simulation, internally, it is constructed rather differently. The availability of the APL Plus-batch file interface permits the best of both worlds.

In almost all business games which involve more than two teams, the inputs (in the form of various management decisions) must be provided by all teams before the game can be run for the next period. This is intrinsically a batch operation. On the other hand, it is very helpful if each team can prepare input, and analyze output, in an interactive mode.

There are 72 teams divided into 12 industries of 6 firms each. The teams prepare input at the APL terminals, making use of whatever tools they find helpful: resource allocation models; *pro forma* income projections; forecasts based upon regression techniques; and, in a few cases, specially-prepared simulation models of the industry and the competition (i.e., a simulation of a simulation).

Each of the teams then stores its decision in a specially-accessed—and protected—APL Plus file. When all the decisions are in, they are passed to a temporary batch file. From here they are input into a FORTRAN-coded batch program which actually executes the game for the next period of play. The results of this execution are printed on the high-speed remote printer in the School as well as being returned to each team's APL stored workspace; there they can analyze the results for the next round of competition.

6.4

Application Questionnaire
and Cover Letter



GRADUATE SCHOOL OF MANAGEMENT
LOS ANGELES, CALIFORNIA 90024

Thank you for your interest in our program, Interactive Computing for Management Educators, to be held here at UCLA during the period June 29 - July 11, 1975. The response has been such that we expect to have many more applicants than we have openings. For this reason, we will be forced to select from among those expressing interest in order to obtain the desired class size of 24. As stated in the descriptive brochures, it is our desire "to obtain a broad cross section of disciplines, functional areas, and representation from among AACSB member schools."

Toward this end, I am enclosing an application questionnaire which I would like for you to fill out and return to me by April 5th. I know this questionnaire may appear formidable, but only in this way can we obtain the broad base of information we need in order to make intelligent decisions. Therefore, I beg for your indulgence.

As of the writing of this letter, it appears that there will be approximately five applicants for every one opening. On the surface, this may discourage you from completing the questionnaire, especially when you see the questions relating to computing background. You may assume that because you have little or no experience, your chances are correspondingly lower. Actually, if anything, the reverse is true. As stated in the brochure "there is no need to have had any previous experience in computing." Also, "it is the intention of this program...to make the transition from non-use [emphasis added] to informed use as easy and painless as possible."

A second reason for completing and submitting the questionnaire — even if you are not selected — is that your application, together with others who were unable to be accepted, will form a "data base" of evidence of the widespread interest for programs such as this. Hopefully, such a large list of individuals would move IBM or some other sponsoring organization to underwrite a second or third offering. Of course, there is no guarantee of this, but I would be remiss if I did not mention it as a possibility.

Therefore, may I enlist your care and assistance in completing the application questionnaire and returning it to me here at UCLA. All applicants will be notified of the final selections by the second week in April. Thanks again for your interest and help.

Sincerely,

EPHRAIM R. McLEAN
Assistant Professor of
Information Systems
Director, Center for
Information Studies

ERM/mc
Enclosure

INTERACTIVE COMPUTING FOR MANAGEMENT EDUCATORS

Graduate School of Management
University of California
Los Angeles, California 90024

29 June - 11 July 1975

Application Questionnaire

Please fill out this questionnaire as completely as possible and mail to the address below by April 5, 1975.

Professor Ephraim R. McLean
Graduate School of Management
University of California
Los Angeles, California 90024

I. Information Concerning Applicant

1. Full Name _____

2. Present Title _____

3. Institution _____

4. Address (include ZIP Code):

a. Business _____

Telephone _____

b. Home _____

Telephone _____

5. Date of Birth _____ Birth Place _____

6. Citizenship _____ 7. Soc. Security # _____

8. Education:	Type of Degree	Date Obtained	Institution Where Obtained
a) Undergraduate:	_____	_____	_____
b) Graduate:	_____	_____	_____
	_____	_____	_____

9. Discipline or Area of Specialization: _____

10. List the three courses which you teach most regularly:

<u>Title</u>	<u>Topic (if not clear from title)</u>
a. _____	_____
b. _____	_____
c. _____	_____

In which of these courses do you anticipate making the greatest use of interactive computing? a. _____ b. _____ c. _____

If some other course, please describe: _____

II. Information Concerning Applicant's Computer Background (if any).

1. Have you ever written a computer program or taken a course in computer programming?

No: _____ If no, please go on to Question II.3. below.

Yes: _____ If yes, please state in which language(s) these programs were written _____

2. List the computer courses you have taken (do not list more than 5):

	<u>Title</u>	<u>Topic (if not clear from title)</u>
a.	_____	_____
b.	_____	_____
c.	_____	_____
d.	_____	_____
e.	_____	_____

3. Have you ever used the computer in a conversational, interactive mode (e.g., APL, BASIC, TSO, etc.)?

No: _____

Yes: _____ If yes, please describe: _____

4. Have you used the computer in any other way not described above?

No: _____

Yes: _____ If yes, please describe: _____

III. Information Concerning Applicant's Institution

1. Number of Students Enrolled:

	<u>Business School or Department</u>	<u>Total Campus</u>
a) Undergraduate:	_____	_____
b) Masters:	_____	_____
c) Doctorate:	_____	_____
d) Total:	_____	_____

2. Year in which business school or department was founded: _____

3. Total number of faculty F.T.E.'s (Full Time Equivalents): _____
4. Type of computer most frequently used by business school faculty and students. (Please give manufacturer's name and model number. E.g. IBM 360/50, IBM 370/145, Burroughs 5700, Xerox Sigma 7, etc.): _____

5. Computer available to business school faculty and students, if different from Question III.4. above. (Again, please give manufacturer's name and model number): _____

6. Does the business school currently have a terminal-based interactive computing service available to it on a regular basis?
No: _____ If no, please go on to Question III.9. below.
Yes: _____
7. What type of interactive service is available?

BASIC : _____	APL (Burroughs): _____
APL/360 : _____	APL/CMS : _____
APL * PLUS : _____	APL/SV : _____
APL (UNIVAC): _____	APL (Xerox) : _____

 Other (please describe): _____

8. Number of computer terminals available for student and faculty use: _____
9. What plans are there for the future with regards to acquiring or enhancing the interactive computing capability for the business school and/or the campus as a whole (please be as specific as you can)? _____

IV. Information Concerning Use of Interactive Computing

1. Do business school students currently take a required course in computing?
No: _____
Yes: _____ If yes, please describe this course: _____

2. Are there other courses in the curriculum which make use of the computer?

No: _____

Yes: _____ If yes, please list them (if more than 5, list only those closest to your discipline or area of specialization):

<u>Title</u>	<u>Topic (if not clear from title)</u>
a. _____	_____
b. _____	_____
c. _____	_____
d. _____	_____
e. _____	_____

3. As part of the program, Interactive Computing for Management Educators, each participant will be required to develop and program (in APL) an application relating to his or her discipline. If selected, describe what your application would be:

4. Recognizing that there may be many more applicants than there are openings in this program, please make a brief statement as to why you wish to attend and what you believe you will be able to do with the experience once you return to your own campus:

5. Although all expenses while at UCLA will be covered by the sponsoring grant from IBM, travel to and from Los Angeles will not be covered. Please indicate the source of such travel funds:

School or university sources: _____

From your own pocket: _____

Other (please describe): _____

6. If you are unable to get into the June 29th program because of the limited number of openings, would you be interested in attending this program at some later date (assuming that additional support could be obtained)?

No: _____

Yes: _____ If yes, please indicate your preference as to time:

Late July 1975 : _____

Mid - August 1975: _____

Summer 1976 : _____

Other : _____

Thank you for your care and patience in completing this application questionnaire.

6.5

Letter of Acceptance



GRADUATE SCHOOL OF MANAGEMENT
LOS ANGELES, CALIFORNIA 90024

On behalf of Ken Siler, Jim Schenck, and myself, I would like to take this opportunity to welcome you personally to the upcoming program "Interactive Computing for Management Educators" here at UCLA.

At final count, there were 105 faculty members who made application to the program and so the selection process proved to be very difficult indeed. I believe, however, that those who were selected represent a broad cross section in terms of disciplines, schools, and geographical distribution. I hope that the two weeks that you spend with them—and with us—will prove to be enjoyable as well as productive.

The program itself is described in a fair amount of detail in the announcement brochure and so I won't repeat that information here. However, a few additional details may be helpful.

The program will open with a reception-cocktail party, followed by dinner, beginning at 6:00 P.M. on Sunday, June 29th, at the Holiday Inn of Westwood, 10740 Wilshire Boulevard, in West Los Angeles. A color brochure, describing the Holiday Inn is enclosed. We have made reservations for you at this Holiday Inn for the 12 nights, June 29th through July 10th. If you wish to arrive earlier or stay later (at your expense), please let us know so that we may notify the Holiday Inn of your plans. All of the rooms will be double occupancy, but each room has two queen-sized beds and so I am sure you will be quite comfortable.

The Inn is about 12 miles from Los Angeles International Airport and there is a bus which goes directly to the Inn from either Building 2 or Building 7 at the Airport. The fare is \$1.70 and the schedule is as follows:

L.A. International Airport		
<u>Bldg. 2</u>	<u>Bldg. 7</u>	<u>Holiday Inn</u>
9:25 A.M.	9:35 A.M.	10:00 A.M.
11:50 A.M.	12:00 A.M.	12:30 P.M.
1:40 P.M.	1:50 P.M.	2:20 P.M.
3:10 P.M.	3:20 P.M.	4:00 P.M.
5:10 P.M.	5:20 P.M.	5:45 P.M.

If you miss the bus, there is of course taxi service; the fare is approximately \$6.00 - \$7.00.

Breakfasts will be at the Holiday Inn (just charge it to your room) and lunches at the UCLA Faculty Club. A cash stipend will be provided upon arrival so that you can have your choice of dinner at either the Inn or at one of the many restaurants in Westwood. Laundry dry cleaning, or other personal expenses can also be charged to your room. A special bus will provide transportation between the Inn and the Graduate School of Management. Westwood itself is within walking distance, so you will not require a car unless you wish to plan a trip on your own (e.g., a visit to Disneyland on the middle weekend). Naturally, such trips, as well as any other private entertainment, are not covered by the IBM grant.

The weather in Los Angeles is quite pleasant in June-July (mid-80's), and the dress for classes will be informal. However, the evenings can be cool and so you will probably want something warmer for evening wear.

Classes will be Monday through Friday for both weeks (including the 4th of July), 8:30 A.M. to 5:30 P.M. We plan to finish the formal instruction early in the second week and devote the balance of the time to working on your individual programs. As mentioned in the announcement, these APL programs are to be published in a monograph, and so we are most anxious that everyone have their program completed by Friday morning. Therefore, you may make plans to leave Los Angeles sometime Friday afternoon or evening. More information about the preparation for this programming assignment is covered on an enclosed sheet.

Finally, I am enclosing a list of the course materials and a copy of one of the books we will be using: Marilyn Bohl's Information Processing. Although none of the participants are knowledgeable about APL, several do have some background in computing. Therefore, in order that we all have the same vocabulary, I am sending this book in advance so that you can "come up to speed" on basic computer concepts prior to the beginning of the program. It is a well-written, straight-forward text and the enclosed book list indicates those pages

that will be most useful. If you do not have any previous background in computing, please read the assigned sections before June 30th.

I believe that I have covered most of the key points of the program; however, if you have any questions between now and its beginning, please feel free to call Miss Maddie Contraras, the administrative assistant, at (213) 825-2001. The number for the Holiday Inn is (213) 475-8711. I am looking forward to meeting you personally and to the two weeks we will be spending together.

Sincerely,

Ephraim R. McLean
Assistant Professor of
Information Systems
Director, Center for
Information Studies

ERM/kw

Enclosures (5)

INTERACTIVE COMPUTING FOR MANAGEMENT EDUCATORS

Graduate School of Management
University of California, Los Angeles

READING LIST

The following books and materials are to be used in conjunction with the UCLA program on APL for management:

*Bohl, Marilyn, Information Processing, Chicago: SRA, 1971.

Buckley, J. W., Nagaraj, M. R., Sharp, D. L., and Schenck, J. W., Management Problem-Solving with APL, Los Angeles: Melville Publishing Co., 1974.

Gilman, L., and Rose, A. J., APL: An Interactive Approach (2nd Edition), New York: John Wiley & Sons, Inc., 1974.

Scientific Time Sharing Corporation, APL Plus Δ FMT Report Formatting Instruction Manual, Bethesda, Maryland: STSC, 1972.

Scientific Time Sharing Corporation, APL Plus File Subsystem Instruction Manual, Bethesda, Maryland: STSC, 1971.

Scientific Time Sharing Corporation, APL Plus Time-Sharing Reference Data Card, Bethesda, Maryland: STSC, 1971.

* The Information Processing book is to be read prior to arrival. Although all of the book is informative and potentially useful, the most important parts are found on the following pages: 1-37, 49-93, 106-171, 189-220, 231-235, and 247-282. The other books and materials will be distributed at the beginning of the course.

6/9/75

6.6

Program Evaluation Questionnaire

INTERACTIVE COMPUTING FOR MANAGEMENT EDUCATORS

Graduate School of Management
University of California, Los Angeles

Program Questionnaire

I. Accommodations

	Poor	Outstanding
1. Rooms at the Holiday Inn	/ / / / / / / / /	/ / / / / / / / /
2. Meals		
Breakfasts at the Holiday Inn	/ / / / / / / / /	/ / / / / / / / /
Lunches at the UCLA Faculty Center	/ / / / / / / / /	/ / / / / / / / /
Arrangements for independent dinner dining	/ / / / / / / / /	/ / / / / / / / /
3. Transportation	/ / / / / / / / /	/ / / / / / / / /
4. Administrative and clerical support (Maddie Contreras)	/ / / / / / / / /	/ / / / / / / / /
5. Program support (Bob Bradspies)	/ / / / / / / / /	/ / / / / / / / /
6. Comments and/or general appraisal of housing, meals and other living arrangements:		

II. Course

	Poor	Outstanding
1. Text (Gilman & Rose)	/ / / / / / / / /	/ / / / / / / / /
2. Other reading materials and handouts	/ / / / / / / / /	/ / / / / / / / /
3. Lecture content and sequencing	/ / / / / / / / /	/ / / / / / / / /
4. Lab assignments	/ / / / / / / / /	/ / / / / / / / /

5. Presentations by outsiders

Didn't Attend

STSC	()	/	/	/	/	/	/	/	/
PCS	()	/	/	/	/	/	/	/	/
Xerox	()	/	/	/	/	/	/	/	/
Bill Bergquist	()	/	/	/	/	/	/	/	/

	Too Basic								Too Advance
--	-----------	--	--	--	--	--	--	--	-------------

6. Overall course content / / / / / / / / /

	Too Slow								Too Fast
--	----------	--	--	--	--	--	--	--	----------

7. Pace of presentation / / / / / / / / /

8. What would you like to see added or expanded?

9. What would you like to see deleted or reduced?

10. Other suggested changes to the structure or content of the program:

11. Other suggestions or comments:

III. Instruction

1. Quality and effectiveness of classroom sessions

	Poor								Outstanding
Eph McLean	/	/	/	/	/	/	/	/	/
Ken Siler	/	/	/	/	/	/	/	/	/
Jim Schenck	/	/	/	/	/	/	/	/	/

2. Quality and effectiveness of lab sessions (Jim Schenck) Poor Outstanding
 / / / / / / / / /

3. Quality and effectiveness of individual coaching and assistance

Bob Bradspies / / / / / / / / /

Eph McLean / / / / / / / / /

Jim Schenck / / / / / / / / /

Ken Siler / / / / / / / / /

4. Comments and/or general appraisal of the quality of instruction:

IV. Summary

1. Your overall evaluation of the program as a whole Poor Outstanding
 / / / / / / / / /

2. A brief statement summarizing your overall appraisal of the program:

3. Any other suggestions for improvement?

 Name (Optional)

THANK YOU FOR YOUR ASSISTANCE

6.7

Post Conference Follow-Up Letter

UNIVERSITY OF CALIFORNIA, LOS ANGELES

BERKELEY • DAVIS • IRVINE • LOS ANGELES • RIVERSIDE • SAN DIEGO • SAN FRANCISCO



SANTA BARBARA • SANTA CRUZ

23 July 1975

GRADUATE SCHOOL OF MANAGEMENT
LOS ANGELES, CALIFORNIA 90024

Now that a week or so has past since the end of the program, I hope you have had a chance to recuperate from the rigors of our two weeks together. It was a demanding experience for all concerned; but I, for one, thoroughly enjoyed myself. I hope you did too.

Jim, Ken, and I are in the process of reviewing the programs that were left with us. On the attached sheet is a checklist of the materials that we have from you. If you want--or need--to work on your program, merely go to one of the APL terminals on your campus and then dial into our system here at UCLA. The numbers again are as follows: 213-825-0911, -0921, 0931, -0941, and -0721. Naturally, you will have to identify to the system the type of terminal you are using (e.g., A1) or A9) for IBM 2741's), just as you did here at UCLA. After that, use your APL number and password and then go to town!

The numbers will be kept active until Monday, August 18th. At that time we hope to have everyone's work in hand and will then begin the detailed editing. If you have any questions or problems, don't hesitate to give one of us a call.

Eph McLean 213-825-1449
Ken Siler 213-825-1252
Jim Schenck 213-825-4196 or 213-825-2870

I hope that the balance of the summer will be pleasant for you and that you will be able to put your newly acquired APL skills to use in the Fall.

Sincerely,

EPHRAIM R. McLEAN
Associate Professor of
Information Systems

Attachment

ERM/ss

305

