



Technical Note

No. 203

**BOUMAC
A MACRO-PROGRAMMING SYSTEM
FOR SCIENTIFIC COMPUTATION**

JOHN H. DEVENNEY AND JOHN J. SOPKA



U. S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

U. S. DEPARTMENT OF COMMERCE
Luther H. Hodges, *Secretary*

NATIONAL BUREAU OF STANDARDS
A. V. Astin, *Director*



THE NATIONAL BUREAU OF STANDARDS

Functions and Activities

The functions of the National Bureau of Standards include the development and maintenance of the national standards of measurement and the provision of means and methods for making measurements consistent with these standards; the determination of physical constants and properties of materials; the development of methods and instruments for testing materials, devices, and structures; advisory services to government agencies on scientific and technical problems; invention and development of devices to serve special needs of the Government; and the development of standard practices, codes, and specifications, including assistance to industry, business and consumers in the development and acceptance of commercial standards and simplified trade practice recommendations. The work includes basic and applied research, development, engineering, instrumentation, testing, evaluation, calibration services, and various consultation and information services. Research projects are also performed for other government agencies when the work relates to and supplements the basic program of the Bureau or when the Bureau's unique competence is required. The scope of activities is suggested by the listing of divisions and sections on the inside of the back cover.

Publications

The results of the Bureau's research are published either in the Bureau's own series of publications or in the journals of professional and scientific societies. The Bureau itself publishes three periodicals available from the Government Printing Office: The Journal of Research, published in four separate sections, presents complete scientific and technical papers; the Technical News Bulletin presents summary and preliminary reports on work in progress; and Central Radio Propagation Laboratory Ionospheric Predictions provides data for determining the best frequencies to use for radio communications throughout the world. There are also seven series of nonperiodical publications: Monographs, Applied Mathematics Series, Handbooks, Miscellaneous Publications, Technical Notes, Commercial Standards, and Simplified Practice Recommendations.

A complete listing of the Bureau's publications can be found in National Bureau of Standards Circular 460, Publications of the National Bureau of Standards, 1901 to June 1947 (\$1.25), and the Supplement to National Bureau of Standards Circular 460, July 1947 to June 1957 (\$1.50), and Miscellaneous Publication 240, July 1957 to June 1960 (includes Titles of Papers Published in Outside Journals 1950 to 1959) (\$2.25); available from the Superintendent of Documents, Government Printing Office, Washington, D.C., 20402.

NATIONAL BUREAU OF STANDARDS

Technical Note 203

Issued December 18, 1963

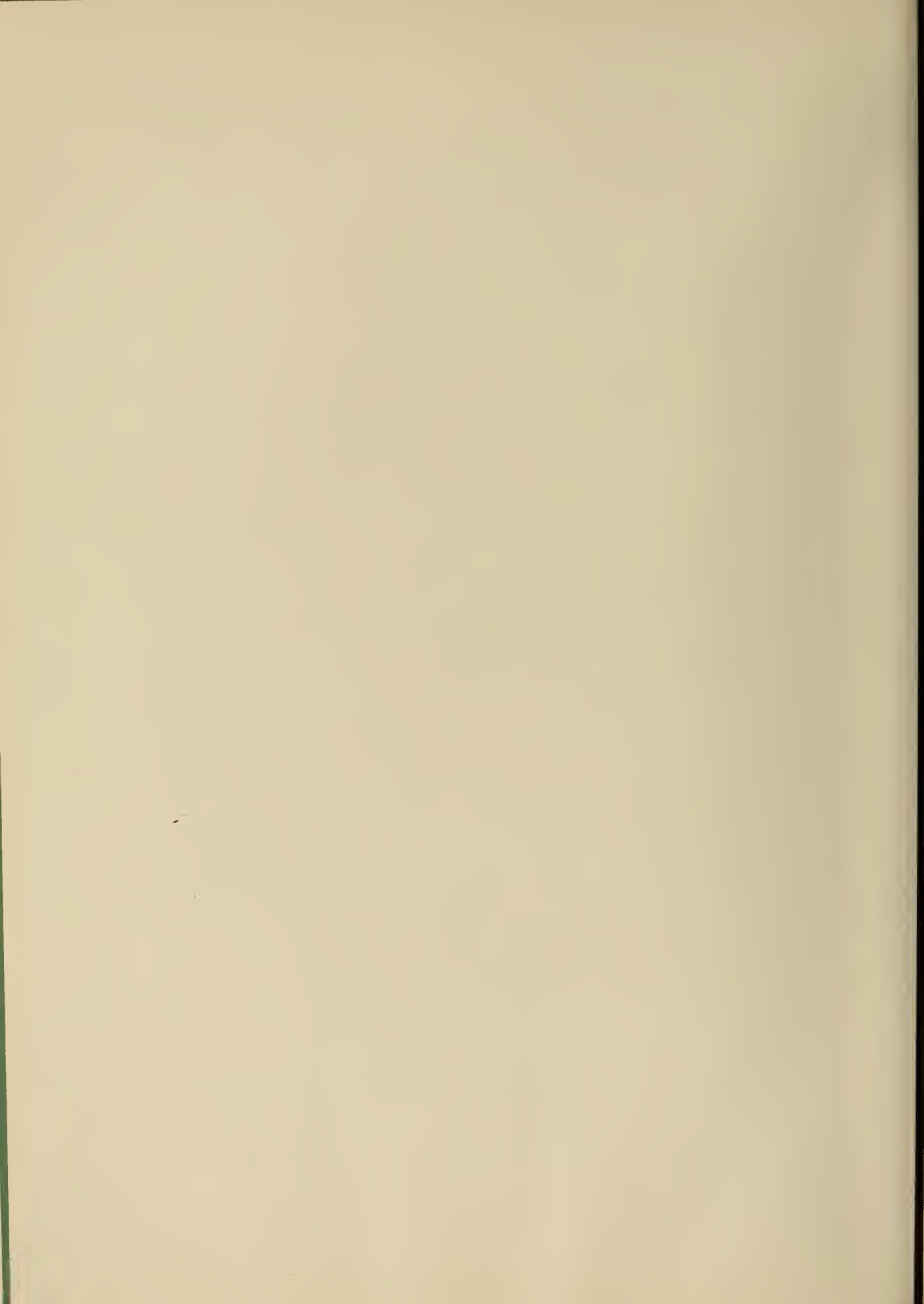
BOUMAC A MACRO-PROGRAMMING SYSTEM FOR SCIENTIFIC COMPUTATION

John H. Devenney and John J. Sopka
National Bureau of Standards
Boulder, Colorado

NBS Technical Notes are designed to supplement the Bureau's regular publications program. They provide a means for making available scientific data that are of transient or limited interest. Technical Notes may be listed or referred to in the open literature.

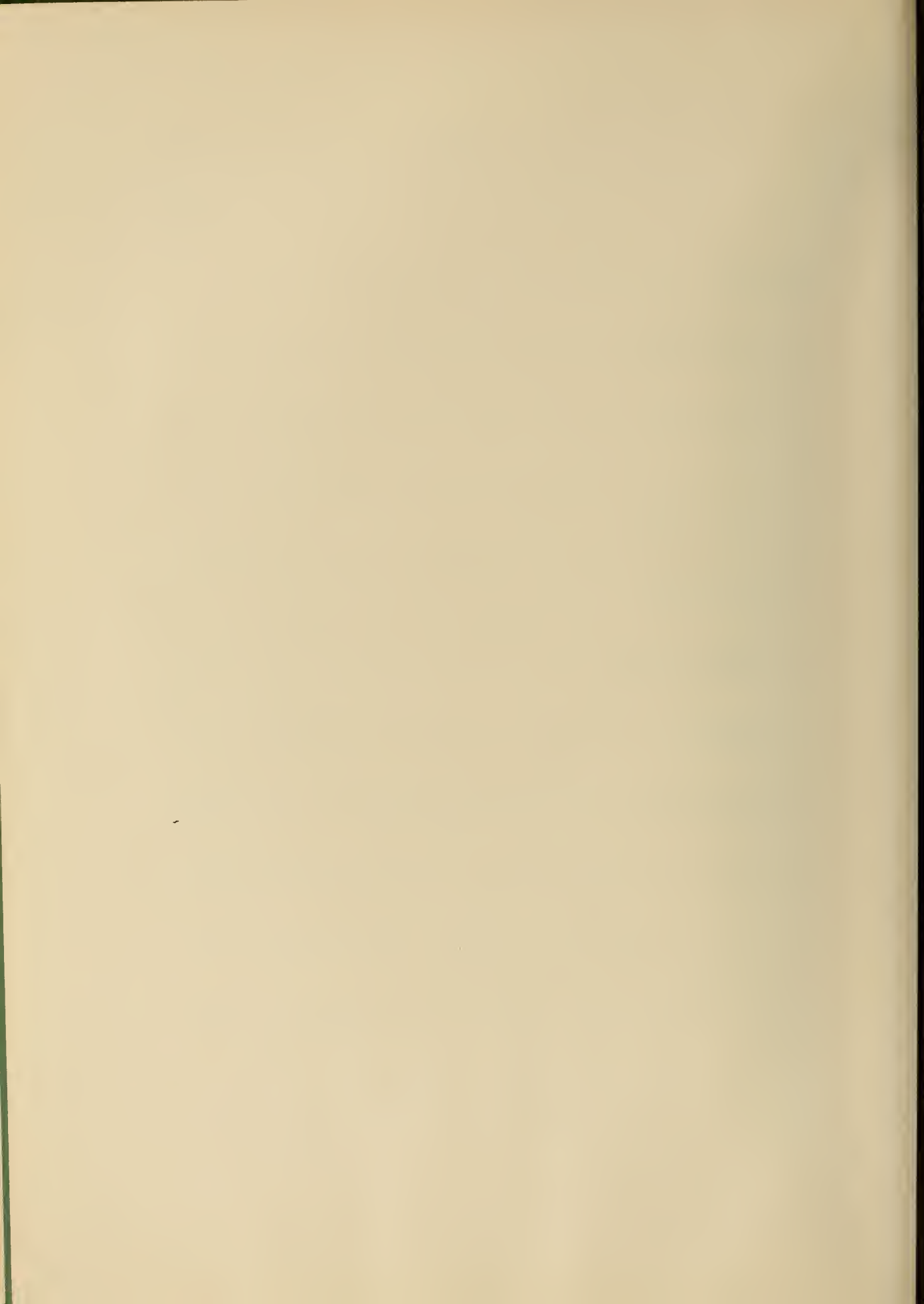
For sale by the Superintendent of Documents, U. S. Government Printing Office
Washington, D.C. 20402

Price 30¢



CONTENTS

	PAGE
ABSTRACT	1
1. INTRODUCTION	1
2. INSTRUCTIONS	3
3. DATA FORMS	10
4. SYSTEM OUTPUT	13
5. RESTRICTIONS	14
6. APPENDIX LIST 1	15
7. APPENDIX 2	16
8. APPENDIX 3	27
9. APPENDIX 4	29
10. APPENDIX 5	34
11. REFERENCES	37



BOUMAC

A Macro-Programming System for Scientific Computation

John H. Devenney
John J. Sopka

The BOUMAC computing system will allow a scientist having no programming knowledge to perform the more commonly used macro computations such as matrix operations, solution of linear equations, numerical integrations, curve fitting, statistical analyses, etc., using a high-speed electronic computer. The scientist needs only to know the various mathematical operations which are required to obtain the information he wants from the data and to learn to use a simple code with which to specify these mathematical operations.

1. INTRODUCTION

The purpose of this system is to allow a scientist having no programming knowledge to perform one, or a sequence, of standard mathematical operations on sets of data using a high speed electronic computer. This system is designed to eliminate the task of programming the computer and hence eliminate the necessity of the scientist to learn computer programming.

As an example, with two lines of simple instructions, the user can solve a set of simultaneous equations (solve the matrix equation $A\bar{x} = \bar{b}$) and then multiply the solution vector \bar{x} by the original matrix A to obtain the original constant vector \bar{b} as a check. Or with one instruction, he could multiply the original matrix by the inverse matrix to obtain the identity matrix. In short, with a very few macro-instructions, the user can perform a number of macroscopic mathematical and statistical operations, provided the operations are specified in Appendix List 1 of this paper.

In general, this macro-system, called BOUMAC, is an attempt to use a canonical format for the more commonly used macro computations such as matrix addition, multiplication, inversion, solution of

linear equations, eigenvalue calculations, etc. It is expected to include numerical integrations, determination of means and standard deviations, linear or multilinear and polynomial least squares fitting, and others to be added. Each such macro-computation requires a single one line instruction written by the user and followed by the data. Instructions and Data are written (by the user) in a standard format on BOUMAC Instruction and Data Forms, as described later.

The user need only take his completed Instruction and Data sheets to the computer facility for processing. Key punching of standard BOUMAC Instruction and Data cards and assembly of the deck will be handled by computer facility personnel. The program can then be run and results submitted to the user in standard output format. If desired, the BOUMAC cards may be taken by the user for future direct use, thus eliminating re-keypunching.

BOUMAC was written to operate with the IBM 7090/7090 FORTRAN System in use at the National Bureau of Standards, Boulder Laboratories, Boulder, Colorado. This FORTRAN system consists of a monitor, the FORTRAN compiler, a symbolic machine language assembly program called FAP, and is essentially the distributed IBM system. Necessary control cards (instructions) for BOUMAC are described in Appendix 4. It is hoped that installations with the same IBM system can employ the BOUMAC computing system with little modification of control instructions. Installations that do not have this system will have greater difficulty using BOUMAC, but perhaps the idea of this type of macro system will be of value.

2. INSTRUCTIONS

The general format of the Instruction Form is shown in figure 1.

The first two lines are merely control information, and the user need be concerned with only three portions. At the Boulder Laboratories, the XXXXX in line one is the project number to be charged and must be inserted by the user. Also, the user's telephone extension can be inserted in columns 42-45. Columns 50-72 are reserved for the user's name. The second line, columns 8 through 72, is for comments. The user may write any desired information up to 65 characters (including spaces), which will in turn be printed on his output.

The type of operation desired is named in columns 8 through 13 of lines 3,4, etc., using the exact name given in Appendix List 1. Starting in column 16 of each such line will be what we shall refer to as an argument string. The argument string will contain names of data (ARRAYS) to be operated on and the amount of information to be processed. The last item of each argument string is a FLAG, which will be described shortly. The last line with the word "END" in columns 8-10 can be disregarded by the user as may any information in columns 73-80 of all lines.

The user is cautioned that all information written on the instruction forms should be printed clearly and legibly to eliminate the chance of errors in processing.

At the Boulder Laboratories, National Bureau of Standards, the following conventions are adopted:

The letter O is written \bar{O}

The number 0 is written ϕ

The letter I is written I

The number 1 is written |

The letter Z is written \mathbb{Z}

The number 2 is written 2

BOUMAC INSTRUCTION FORM				
1	8	16	42 47 50	73
*	I	D(XXXXX, BOUMAC, JHD, P, 1, 1 ϕ)	EXT SSS NAME	
		COMMENTS		MACR1 $\phi\phi\phi$
		OPER1	ARRAY1, N1, ARRAY2, N2, FLAG	
		OPER3	ARRAY3, N3, ARRAY4, N4, FLAG	
		.	.	
		.	.	
		.	.	
		OPERN	ARRAYN, NN, FLAG	
		END		MACROEND

Figure 1

As an example, suppose it is desired to solve a set of simultaneous equations, i.e., solve the matrix equation $A\bar{x} = \bar{b}$ where A is a non-singular $n \times n$ matrix and \bar{b} is a constant vector of n elements. The required Instruction Form is shown in figure 2.

In this example, line 1 carries the Project Number 79000, Line 2 is the comment, line 3 is the instruction in which

SIMEQU is the operation to be performed,
ALPHA is the name of the matrix,
N is the order of the matrix,
BETA is the name of the constant vector,
FLAG is either the integral number 0 or 1.

Each item in the argument string must be separated by a comma. No punctuation should follow the flag.

Notice that "ALPHA" and "BETA" are names of the information to be processed and may be written as any combination of 1 through 6 characters. "A" and "B" could just as well have been used. "N" is an integral number. If the matrix had had dimensions 20×20 , then 20 would be written after the name "ALPHA". "FLAG" is an output option.

It is also possible to chain together a sequence of such operations on the same data. Intermediate results in a sequence of operations often need not be printed at the end of each operation.

If FLAG $\left\{ \begin{array}{l} = 0, \text{printing is not suppressed} \\ = 1, \text{printing is suppressed} \end{array} \right.$

Assume we wish to multiply matrices A and B together, i.e., $A \times B = C$. The dimensions of A are 5×8 ; B is 8×5 ; C is 5×5 . Then, suppose we wanted to add matrix C to matrix D, i.e., $C + D = E$, suppress print-out, and, finally, obtain the inverse of matrix E. The required Instruction Form is shown in figure 3. Line 1 of figure 3 carries Project Number 79646 and the Programmer's Name and Extension. A description of the Instructions is as follows:

Instruction 1.

MATMPY is the operation of matrix multiplication.

A and B are the names of the matrices to be multiplied.

C is the name of the product matrix.

5 is the number of rows of the A matrix (equals the number of rows of the C matrix).

8 is the number of rows of the B matrix (equals the number of columns of the A matrix).

5 is the number of rows of the C matrix (equals the number of columns of the B matrix).

ϕ is to print out results.

Instruction 2.

MATADD is the operation of matrix addition.

C,D, and E are the names of the matrices.

5 is the row-column dimension of the matrices.

1 is to suppress print-out.

Instruction 3.

INVERT is the operation of matrix inversion.

E is the name of the matrix to be inverted.

5 is the order (dimension) of the matrix.

ϕ is to print out results.

A word of caution at this point. Many mathematical routines destroy the original information during processing. For instance, in the INVERT Instruction of the preceding example, the original matrix E is destroyed during the matrix inversion process. Hence, if the matrix is to be used again in other Instructions, it must be preserved before using the INVERT Instruction. The system provides this function with the Instruction shown in figure 4.

Where

SAVE is the operation to preserve information,

AR1 is the name of the information to be preserved.

N is the number of rows of the array (N = 1 if the array is one dimensioned).

M is the number of columns of the array.

AR2 is the name of the preserved information.

Again, note that AR1 and AR2 are merely names and may be any combination of 1-6 alphanumeric characters. In the example above, SAVE E, 5, F, 5, ϕ inserted between MATADD and INVERT would preserve matrix E as matrix F and print F.

A complete description of the available Instructions appears in Appendix 2. The write-up for each routine that destroys original information will indicate which information is destroyed.

BOUMAC INSTRUCTION FORM						
1	8	16	42	47	50	73
*	I	D(79000,	BOUMAC, JHD, P, 1, 1 ϕ)	EXT	SSS	NAME
		THIS IS	A SAMPLE BOUMAC INSTRUCTION			
		SIMEQU	ALPHA, N, BETA, FLAG			
		END				
						MACR1 $\phi\phi\phi$
						MACR \bar{O} END

Figure 2

BOUMAC INSTRUCTION FORM						
1	8	16	42	47	50	73
*	I	D(79646,	BOUMAC, JHD, P, 1, 1 ϕ)	EXT	SSS	JOHN DOE
		ANOTHER	BOUMAC EXAMPLE			
		MATMPY	A, 5, B, 8, C, 5, ϕ			
		MATADD	C, 5, D, 5, E, 1			
		INVERT	E, 5, ϕ			
		END				
						MACR1 $\phi\phi\phi$
						MACR \bar{O} END

Figure 3

BOUMAC INSTRUCTION FORM			
1	8	16	73
	SAVE	AR1, N, AR2, M, FLAG	

Figure 4

BOUMAC DATA FORM								
1	7	11	21	31	41	51	61	71
*	DATA							

Figure 5

BOUMAC DATA FORM								
1	7	11	21	31	41	51	61	71
*	DATA							
	1.	1.	-1.	2.	2.	3.	-2.	1.
	4.	2.	3.	ϕ .	-3.	2.	4.	3.
	3.	-5.	6.	ϕ .				

Figure 6

3. DATA FORMS

The general format of the Data Form is shown in figure 5. The first line will contain an asterisk in Column 1 and the word "DATA" in Columns 7-10. This is a control signal to the system, and the user need not be concerned with it. The user should start recording data on line 2. There can be up to and including eight pieces of data on each line, with a width of ten columns per piece of data. The data can be written anywhere within the ten columns. The decimal point must clearly be indicated (it will require one column).

The reader who is familiar with FORTRAN is referred to Appendix 3 for a discussion of data input format which allows even greater flexibility than described here.

Again the user is cautioned that all information written on the data forms must be printed clearly and legibly.

All data (for 2 dimensional arrays) must be written in a continuous row-sequence on the data forms, with each new array of data starting on a new line, e.g., in $A \times B = C$, data in A and B must start new lines. As an illustrative example, suppose we wish to solve a set of simultaneous equations with 4 equations and 4 unknowns.

$$\begin{pmatrix} 1 & 1 & -1 & 2 \\ 2 & 3 & -2 & 1 \\ 4 & 2 & 3 & \phi \\ -3 & 2 & 4 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 3 \\ -5 \\ 6 \\ \phi \end{pmatrix}$$

The data must be written row-wise in a continuous sequence. Figure 6 illustrates the data form.

If information required by the next Instruction to be processed is not in the computer memory, the system will initiate input from the cards punched by the Computer Facility personnel, the cards being punched from the BOUMAC Data Forms. That is, if the name of an

array is encountered for the first time in an Instruction, the data for that particular array will be read in at that time. Hence, the data must be written on the forms in the order it is to be processed. To illustrate, suppose the Instruction sequence as shown in figure 7 is executed. The Instruction "SAVE" will store the array ALPHA at GAMMA, since ALPHA is destroyed during execution of the "SIMEQU" Instruction. There are 16 elements saved (a 4 x 4 matrix). The "SIMEQU" Instruction will solve a 4 x 4 set of simultaneous equations where ALPHA is the original matrix and BETA is the original constant vector. When this Instruction is completed, the inverse matrix will be located at ALPHA and the solution vector will be located at BETA. The "MATMPY" Instruction will multiply the original matrix (GAMMA) by its inverse (ALPHA) to obtain the identity matrix (DELTA). The read in of data for the ALPHA array is initiated by the SAVE Instruction since the name ALPHA has not been previously encountered, and, by the same token, the data for BETA is read in by the "SIMEQU" Instruction. Hence, the data must be written on the data forms in that sequence.

BOUMAC INSTRUCTION FORM			
1	8	16	73
	SAVE	ALPHA,4, GAMMA, 4, FLAG	
	SIMEQU	ALPHA,4, BETA, FLAG	
	MATMPY	GAMMA,4, ALPHA,4, DELTA,4, FLAG	

Figure 7

INVERSE MATRIX (PRINTED ROWWISE)					
Row No.					
1	1 x_1	2 x_2	3 x_3	4 x_4	5 x_5
2	1 x_6	2 x_7	3 x_8	4 x_9	5 x_{10}
3	1 x_{11}	2 x_{12}	3 x_{13}	4 x_{14}	5 x_{15}
4	1 x_{16}	2 x_{17}	3 x_{18}	4 x_{19}	5 x_{20}
5	1 x_{21}	2 x_{22}	3 x_{23}	4 x_{24}	5 x_{25}

Figure 8

4. SYSTEM OUTPUT

The first sheet of output will contain a list of the Instructions written by the user. The data output of each separate operation will begin on a new page with a self-explanatory heading printed in the center of the page. To conserve space and print-out time, information will be printed across the page with ample information for proper interpretation. For example, using the INVERT Instruction to invert a 5 x 5 matrix will produce the output, as shown in figure 8, where the x's are the elements of the inverse matrix and the integer values indicate the row and column number.

5. RESTRICTIONS

The memory of the computer can be exceeded in two ways. One is that the size of the information being processed is too large (e.g., it would be impossible to invert a 200 x 200 matrix using only core storage), and the other is using too many Instructions at one time which would cause the memory of the computer to become over-filled. When the memory is exceeded, the following message is printed:

CORE STORAGE EXCEEDED

JOB FLUSHED

6. APPENDIX LIST 1

List of Available Instructions and Operation Code*

Solution of Simultaneous Equations	SIMEQU
Matrix Inversion	INVERT
Determinant Evaluation	DETERM
Matrix Multiplication	MATMPY
Matrix Transposition	MATRAN
Matrix Addition	MATADD
Matrix Subtraction	MATSUB
Scalar-Matrix Multiplication	SCMPY
Scalar-Matrix Division	SCMDIV
Data Preservation	SAVE
Variable-Input Format	READIN
Mean, Variance, Standard Deviation	STAT1
Linear Correlation	CORLAT
Polynomial Root Finder	PIROOT
Multiple Regression	MULREG
Auto-correlation	AUTOOCR

This Appendix List 1 will be continuously updated with the addition of new macro-operations. Suggestions for such additions will be appreciated.

* Refer to Appendix 2 for a detailed discussion of each Instruction.

7. APPENDIX 2

Available Operations and Instruction Description

SIMEQU - Solution of Simultaneous Equations

This Instruction solves the matrix equation $A\bar{x} = \bar{b}$, where A is a real non-singular square coefficient matrix and \bar{b} is a vector of constant elements. A and b are destroyed during the operation.

	8	16	
	SIMEQU	ALPHA,N,BETA,FLAG	73

where ALPHA is any name of the matrix A,

BETA is any name of the constant vector \bar{b} ,

N is the number of equations to be solved,

FLAG is the print out option.

NOTE: The inverse matrix A^{-1} is located at ALPHA and the solution vector \bar{x} is located at BETA upon completion.

INVERT - Matrix Inversion

The Instruction inverts a real non-singular square matrix. The matrix is destroyed during the inversion process.

	8	16	
	INVERT	GAMMA,N,FLAG	73

where GAMMA is any name of the matrix to be inverted,

N is the order of the matrix,

FLAG is the print out option.

Note: the inverse matrix is located at GAMMA upon completion.

DETERM - Determinant Evaluation.

This Instruction calculates the determinant of a given real matrix A. A is destroyed during the evaluation.

8	16	73
DETERM	AMATRIX, N, DET, FLAG	

where AMATRIX is any name of the given matrix,
DET is any name of the determinant,
N is the order of the matrix,
FLAG is the print out option .

Note: the inverse matrix is located at AMATRIX and the determinant is located at DET upon completion.

MATMPY - Matrix Multiplication

This Instruction multiplies two real matrices, i.e., $A \cdot B = C$, where A is an $n \times m$ matrix, B is $m \times l$, and C is $n \times l$.

8	16	73
MATMPY	A, N, B, M, C, L, FLAG	

where A,B,C are any names of the matrices,
N is the number of rows of the A matrix,
M is the number of rows of the B matrix,
L is the number of columns of the C matrix,
FLAG is the print out option.

The product matrix is located at C upon completion.

MATRAN - Matrix Transposition

This Instruction computes the transpose of a real matrix A (i.e. $A^t = B$) with dimensions $n \times m$.

8	16	73
MATRAN	A, N, B, M, FLAG	

where A is any name of the matrix to be transposed,
 B is any name of the transposed matrix,
 N is the number of rows of A,
 M is the number of columns of A,
 FLAG is the print out option.

The transposed matrix, A^t , is located at B upon completion.

MATADD - Matrix Addition

This Instruction adds two $n \times m$ real matrices, i.e., $A + B = C$.

1	8	16	73
	MATADD	A,N,B,M,C,FLAG	

where A and B are any names of the matrices to be added,
 C is any name of the matrix sum,
 N is the number of rows of the matrices,
 M is the number of columns of the matrices,
 FLAG is the print-out option.

The sum is located at C upon completion.

MATSUB - Matrix Subtraction

Given two real $n \times m$ matrices, A and B, this instruction subtracts B from A, i.e., $A - B = C$.

1	8	16	73
	MATSUB	A,N,B,M,C,FLAG	

where A and B are any names of the matrices indicated in the equation above.

C is any name of the matrix difference.

N is the number of rows of the matrices.

M is the number of columns of the matrices.

FLAG is the print out option.

The difference is located at C upon completion.

SCMPY - Scalar-Matrix Multiplication

This Instruction solves the equation $B = kA$, where A and B are real $n \times m$ matrices and k is a real scalar.

1	8	16	73
	SCMPY	A,N,B,M,K,FLAG	

where A and B are any names of the matrices,
K is the scalar (written with the decimal point),
N is the number of rows of the matrices,
M is the number of columns of the matrices,
FLAG is the print out option.

The resultant matrix is located at B upon completion.

SCMDIV - Scalar-Matrix Division

This Instruction solves the equation $D = \frac{1}{c} E$, where D and E are real $n \times m$ matrices and c is a real scalar.

1	8	16	73
	SCMDIV	E,N,D,M,C,FLAG	

where D and E are any names of the matrices,
C is the scalar (written with the decimal point),
N is the number of rows of the matrices,
M is the number of columns of the matrices,
FLAG is the print out option.

The resultant matrix is located at D upon completion.

SAVE - Preservation of arrays of information

This instruction will take the information at ARRAY1 and store it at ARRAY2 for future use. This instruction is useful when inverting matrices, evaluating determinants, solving simultaneous equations, etc.

1	8		73
	SAVE	ARRAY1,N,ARRAY2,M,FLAG	

where ARRAY1 is any name of the information to be saved,

ARRAY2 is any name where the preserved information is to be located,

N is the number of rows in the array,

M is the number of columns in the array.

Note: N = 1 for one dimensional arrays.

READIN - Variable Input Format

See Appendix 3 for complete description of this Instruction. A word of caution - this Instruction is only for use by people having a good understanding of FORTRAN programming.

STAT1 - Mean, Variance, Standard Deviation

This instruction computes the mean, variance, and standard deviation of a given set of data.

1	8	16	73
	STAT1	X, N	

where X is any name of the data;

N is the number of elements in the set of data. The following formulas are used:

$$\text{Mean} = \bar{X} = \frac{\sum_{i=1}^n X_i}{n},$$

$$\text{Variance} = S^2 = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n-1},$$

$$\text{Standard Deviation} = S = \sqrt{\text{Variance}}$$

CORLAT - Linear Correlation

This Instruction computes the simple linear regression and correlation coefficients of two given sets of data.

1	8	16	73
	CORLAT	X, N, Y	

where X is any name of the independent variable,
 N is the number of observations in the two sets,
 Y is any name of the dependent variable.

The following formula are used:

$$\frac{\sum_{i=1}^n X_i Y_i - \frac{\sum_{i=1}^n X_i \sum_{i=1}^n Y_i}{n}}$$

$$b_{yx} = \text{regression coefficient} = \frac{\sum_{i=1}^n X_i Y_i - \frac{\sum_{i=1}^n X_i \sum_{i=1}^n Y_i}{n}}{\sum_{i=1}^n X_i^2 - \frac{(\sum_{i=1}^n X_i)^2}{n}},$$

r_{yx} = correlation coefficient =

$$\frac{\sum_{i=1}^n X_i Y_i - \frac{\sum_{i=1}^n X_i \sum_{i=1}^n Y_i}{n}}{\left[\left(\sum_{i=1}^n X_i^2 - \frac{(\sum_{i=1}^n X_i)^2}{n} \right) \left(\sum_{i=1}^n Y_i^2 - \frac{(\sum_{i=1}^n Y_i)^2}{n} \right) \right]^{1/2}}$$

The instruction also computes "a" in the prediction equation $Y = a + bX$ from the equation $a = \bar{Y} - b\bar{X}$, where \bar{Y} and \bar{X} are the means of the dependent and independent variable, respectively.

PIROOT - Polynomial Root Finder

This instruction finds the real and complex roots of a polynomial of arbitrary degree with real coefficients.

1	8	16	73
	PIROOT	AI,N,ROOTR, ROOTI, FLAG	

where AI is any name of the array of coefficients. The coefficients must be ordered (on the data forms) from the highest degree to the lowest degree.

N is the degree of the polynomial..

ROOTR is any name of the array where the real parts of the complex roots will be located.

ROOTI is any name of the array where the imaginary parts of the complex roots will be located.

FLAG is the print out option.

All arithmetic is performed in the complex mode. Therefore, all roots found have real and imaginary parts. Real roots will have small imaginary parts in the order of seven decimal places less than the real parts. The method of Muller is used. See D. E. Muller, "A Method for Solving Algebraic Equations Using an Automatic Computer," in Mathematical Tables and other Aids to Computation, 1956, pages 208-215.

MULREG - Multiple Regression

This instruction incorporates all normal phases of statistical regression analysis. The standard statistical coefficients, standard errors, sums of squares, and averages are computed and printed. Predictions and residual errors for each item in the data are computed and printed.

The instruction also includes an optional feature of preparing a list of synthetic observations which are added to the list of observations used in fitting. The only requirement in setting up the synthetic list is to set the independent variables at values at which it is desired to test the variability of prediction. The predicted value and the variance of individual prediction are computed for each synthetic observation in the set.

1	8	16	73
	MULREG	X,N,M,MVP	

where

X is any name of the data ,

N is the number of independent variables,

M is the number of observations per variable,

MVP is the number of synthetic observations used. MVP = 0 if none is used.

The data are written on the data forms in the following manner. The value of each independent variable per observation is written on the data form in a row-wise sequence with the value of the dependent variable written last. For example, an equation with 3 independent variables and 5 observations would be written as shown in figure 9, where the 1st subscript on the X's indicates the observation and the 2nd subscript indicates the variable. The Y's are the value of the dependent variable of each observation.

The data for the synthetic observations (if used) are written on the data forms after the original data, and each observation set must start on a new line. Hence, if we wish to compute $Y = A_0 + A_1X_1 + A_2X_2 + A_3X_3$ for three synthetic observations of the preceding example, the data would appear after the original data as shown in figure 10, where the X^1 's are the synthetic observations.

Information printed is identified by headings which are self-explanatory except for the following abbreviations:

- DET = Determinant,
- SSR = Unadjusted regression sum of squares,
- SSE = Residual or error sum of squares,
- ATSS = Total sum of squares adjusted for the mean,
- ASSR = Regression sum of squares adjusted for the mean,
- COR = Multiple correlation coefficient,
- SDEV = Square root of residual variance,
- RSDEV = Square root of variance of mean.

Restrictions: A maximum of 500 observations with 24 or fewer independent variables may be used.

AUTOOCR - Auto-correlation analysis

Given a series of values X_1, X_2, \dots, X_n , this instruction will compute the serial correlation coefficients (auto-correlation) between successive terms, where the lag (k) goes from 0,1,2, ..., L.

1	8	16	73
	AUTOOCR	X,N,L	

where X is any name of the array of values,

N is the number of elements in the array,

L is the final lag value.

Information printed is identified by headings which are self-explanatory except the column N-K. This is the number of pairs entering into the correlation of lag K.

The only restriction imposed is that L be less than or equal to 300.

Reference

Kendall, M. G., The Advanced Theory of Statistics, Vo. II.
Ch. 30, Sec 12.

$$r_k = \frac{\text{COV}(X_j, X_{j+k})}{(\text{VAR } X_j \text{VAR } X_{j+k})^{1/2}}$$

$$\text{where COV } (X_j, X_{j+k}) = \frac{1}{n-k} \sum_{j=1}^{n-k} (X_j X_{j+k}) - \frac{1}{(n-k)^2} (\sum_{j=1}^{n-k} X_j) (\sum_{j=1}^{n-k} X_{j+k})$$

$$\text{VAR}(X_j) = \frac{1}{n-k} \sum_{j=1}^{n-k} X_j^2 - \frac{1}{(n-k)^2} (\sum_{j=1}^{n-k} X_j)^2$$

$$\text{VAR}(X_{j+k}) = \frac{1}{n-k} \sum_{j=1}^{n-k} X_{j+k}^2 - \frac{1}{(n-k)^2} (\sum_{j=1}^{n-k} X_{j+k})^2$$

and $r_0 = 1$

$$r_{-k} = r_k$$

Appendix 2 will be continuously updated with the addition of new macro-operations. Suggestions for such additions will be appreciated.

BOUMAC DATA FORM							
1	11	21	31	41	51	61	71
* DATA							
X ₁₁	X ₁₂	X ₁₃	Y ₁	X ₂₁	X ₂₂	X ₂₃	Y ₂
X ₃₁	X ₃₂	X ₃₃	Y ₃	X ₄₁	X ₄₂	X ₄₃	Y ₄
X ₅₁	X ₅₂	X ₅₃	Y ₅				

Figure 9

BOUMAC DATA FORM							
1	11	21	31	41	51	61	71
*DATA							
X ₁₁	X ₁₂	X ₁₃	Y ₁	X ₂₁	X ₂₂	X ₂₃	Y ₂
X ₃₁	X ₃₂	X ₃₃	Y ₃	X ₄₁	X ₄₂	X ₄₃	Y ₄
X ₅₁	X ₅₂	X ₅₃	Y ₅				
X ₁₁ ¹	X ₁₂ ¹	X ₁₃ ¹					
X ₂₁ ¹	X ₂₂ ¹	X ₂₃ ¹					
X ₃₁ ¹	X ₃₂ ¹	X ₃₃ ¹					

Figure 10

8. APPENDIX 3

For people who are familiar with FORTRAN programming

There is provided an optional method of data input which allows more flexibility in format specification. This method uses a system Instruction which will in turn generate instructions allowing a variable format. The general type of Instruction is shown in Figure 11.

READIN is the input operation to be performed.

ARRAY is the name of the information to be read in.

N is the integral (base 10) number of elements of information.

Using this Instruction requires that the first line written on the data form before the data must be a recognizable 709/7090 FORTRAN II format. All eighty columns of the sheet can be used to specify the format. Examples are (26F3.1), (E12.4, F10.6, 3E16.8), (3F9.2, 2F10.3, E14.5), etc. Note that parentheses must enclose the desired format. This method of input and the normal system method may be used interchangeably, e.g., one array could be read in under the standard system input (8F10.4) and the next array read in under variable format or vice versa.

BOUMAC INSTRUCTION FORM			
1	8	16	73
	READIN	ARRAY, N	

Figure 11

BOUMAC INSTRUCTION FORM							
1	7	8	16	42	41	50	73
*	I	D(XXXXXX,	BOUMAC, JHD,P,1,1 ϕ)	EXT	SSS	NAME	
		THIS LINE	IS FOR COMMENTS BY THE USER				MACR1 $\phi\phi\phi$
		OPER1	ARRAY1,N1,ARRAY2,N2,FLAG				
		OPER2	ARRAY3,N3,ARRAY4,N4,FLAG				
		.	.	.			
		.	.	.			
		.	.	.			
		END					MACROEND

Figure 12

9. APPENDIX 4

INSTRUCTIONS FOR PROCESSING BOUMAC INSTRUCTIONS

(Not for Users)

Introduction

The people using the BOUMAC system will bring two types of forms to the Computer Facility scheduler for processing. This processing consists of keypunching the information on cards and assembly of the deck which will be run with the regular Fortran stack. The procedure for the forms will now be described.

BOUMAC Instruction Forms

The general format of the Instruction Form is shown in figure 12. All information on these sheets must be punched on 709 cards in the columns indicated. After the cards are punched, the cards illustrated in figure 13 (which can be gang-punched from originals) must be inserted immediately after the ID card.

If the user supplies more than one sheet to the scheduler, then all of the indicated control cards need not be punched. The only ones necessary are the first two lines of the first instruction sheet, only one set of cards as shown in figure 13, and the "END" line on the last instruction sheet.

BOUMAC Data Forms

The general format of the Data Form is shown in figure 14. Here, all information written on the sheets can be punched on plain white cards or general data cards, whichever is easiest for keypunching. If more than one sheet is submitted by the user, only the first "*DATA" card need be punched.

Deck Makeup

After completion of all keypunching, the deck is assembled for processing as illustrated in figure 15.

1	7	8	16	50	73
*		XEQ			
*			MOUNT MACRO SYSTEM TAPE ON B5		
*		PAUSE			
*		FAP			
		UPDATE	1φ		

Figure 13

BOUMAC DATA FORMS								
1	7	11	21	31	41	51	61	71
* DATA								

Figure 14

* NOTE: For the present, this system uses a Fortran binary deck, which must be inserted just before the "*DATA" card. In the future, this binary deck will be replaced by a magnetic tape file. After each job is processed, this binary deck must be removed from the deck. In the future, these routines will be placed on the master tape file and the scheduler will not have to be concerned with them. The scheduler will have to fill out the white job submittal card (necessary information can be obtained from ID card) and place it at the front of the deck. Figure 16 indicates this card.

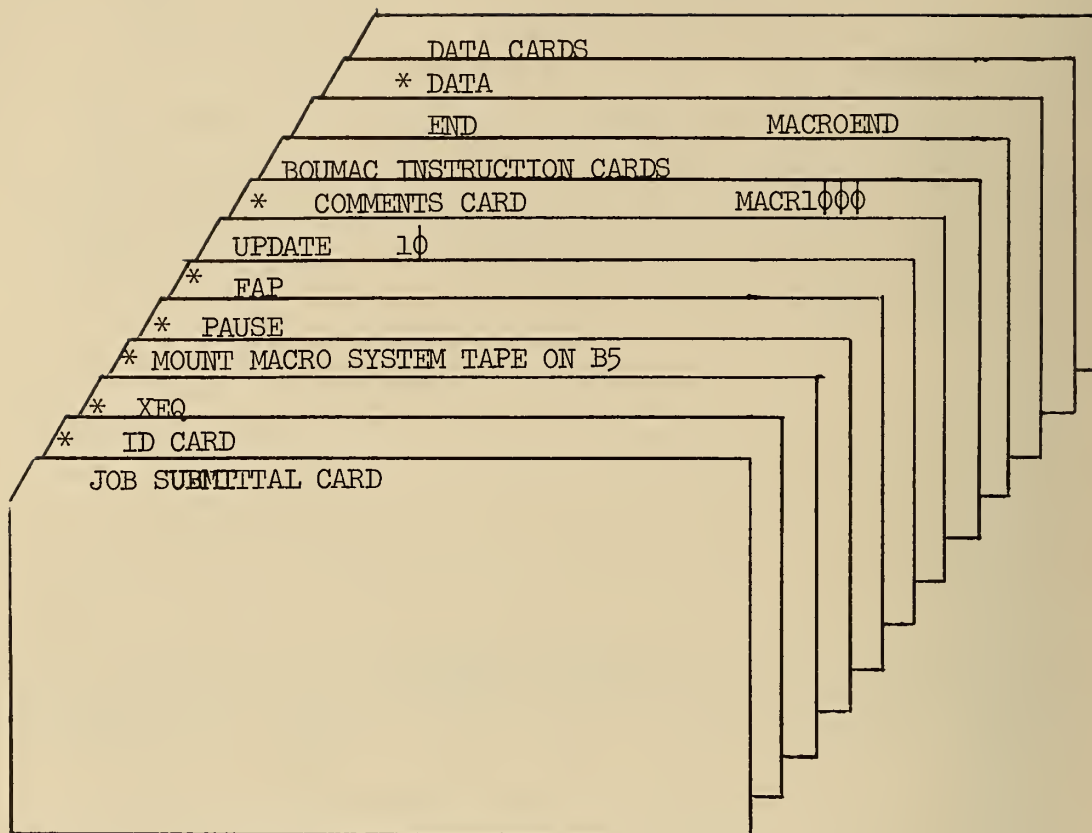


Figure 15

NAME		EXTENSION		REMARKS
<input checked="" type="checkbox"/>	Compile and/or assemble	SPECIAL MAGNETIC TAPES USED		
<input type="checkbox"/>	Execute only	Logical No. Dens	Identification	Comp. Lab. Cat. No.
<input type="checkbox"/>	Output requires special attention. (Special cards, paper, etc.)	B5 High Low	Macro System Tape	C-288
	EXPECTED OUTPUT	H L		
<input checked="" type="checkbox"/>	Printout	H L		
<input checked="" type="checkbox"/>	Binary Cards	H L		
<input type="checkbox"/>	BCD Cards	H L		

Figure 16

10. APPENDIX 5

"What Makes BOUMAC Tick"

The BOUMAC system is programmed for the IBM 7090, using both FORTRAN and FAP. The system is divided into two basic parts: the subroutines that actually perform the various operations and the part which sets up all necessary calling sequences to the subroutines.

Part 1 - The subroutines

Each subroutine is coded in FORTRAN. All two-dimensional arrays have been written in terms of one dimensional arrays to allow "packing" of data in storage. Also, whenever possible, all arrays used by a subroutine are placed in the argument list so, in essence, they are dummy variables and need not explicitly be dimensioned. An example will help clarify the situation. Consider the first three statements of the FORTRAN subroutine used in solving simultaneous equations.

```
SUBROUTINE MATINV (A,N,B)
DIMENSION IPIVOT (100), INDEX1(100), INDEX2(100),
      PIVOT(100), A(1), B(1).
COMMON IPIVOT,INDEX1,INDEX2,PIVOT.
```

In this example, the matrix A, the constant vector B, and the size of the matrix N are in the argument list. Hence, A and B need only be dimensioned with the subscript 1. The arrays IPIVOT, INDEX1, INDEX2, and PIVOT are used internally by the subroutine and hence must be dimensioned. Notice that they are placed in upper storage (COMMON) above the starting location of the system. In this way, all data transmitted to the subroutine will be sequentially located in storage. Hence, we can make optimum use of the computer memory. The binary deck of the Fortran subroutines is placed on the master tape and will be called as needed. This deck could be inserted before the "DATA" card and used in this manner.

Part 2 - The main program

As the name implies, the system utilizes the macro-operation processor. The main program, consisting of FAP instructions, is stored on the MACRO SYSTEM TAPE (magnetic) in the computer facility tape storage area. The first location for storage starts at 66000(8). All elements in COMMON are stored above this location. If this is not enough COMMON storage, the location can be lowered by changing the LOCORG card in the main program. However, this will necessitate either updating the BOUMAC system tape or changing the FAP cards and rewriting the tape on the IBM 1401 (or other similar card-to-tape processor).

The main program begins with the FAP instruction START AXT J,1 and initializes tables. It also sets up the floating point trap routine to indicate either an overflow or underflow occurred. The program sets up three tables called NAME, LENGTH, and BEGIN. The NAME table contains the BCD names of arrays used; the LENGTH table contains the integer number of elements in each array; and the BEGIN table contains the starting address of the first element of each array. As an example, suppose we referenced the array ALPHA which consisted of twenty elements. Then, the three tables in memory might appear as

NAME	LENGTH	BEGIN
452146256060	000024000000	000000051310

The system operates on this principle: the user inserts MACRO Instructions (although he is not aware of this) at the appropriate place in the program. The method will be explained shortly. As each array is encountered, the program searches the NAME table to see if it is there. If it isn't, the program places it in the table, sets up necessary addresses, and continues with the processing of the

next array. If the name is found, then the other information is obtained from the LENGTH and BEGIN Tables and utilized in the processing. In either case, after the name has been found or placed in the table, the necessary instructions are initiated to set up calling sequences for the FORTRAN subroutines. This is accomplished by the macro-generated sequence of instructions.

As previously mentioned, the main program is on magnetic tape. Each card is serialized in columns 73-80. The MACRO Instructions are placed in the proper sequence by the UPDATE feature of the FAP system. They will be placed after the J EQU 100 card. The serialization is used to make sure they are sequenced properly. The FAP program is then assembled and executed.

Note that the J EQU 100 card will allow for 100 locations in each table. Hence, 100 different arrays may be in storage at the same time provided storage is not exceeded. If this is not sufficient, the size may be increased by changing this card. Again note that there are two ways to change this card: by updating the tape or re-writing it. In the same way, new instructions may be added to the tape to allow for future expansion of the system.

11. REFERENCES

1. IBM 7090/7094 Programming Systems Manual -- Fortran II Programming -- File N . 7090-20 Form C28-6054-3
2. IBM 709/7090 Programming Systems: FORTRAN Assembly Program(FAP) Form C28-6235



THE NATIONAL BUREAU OF STANDARDS

The scope of activities of the National Bureau of Standards at its major laboratories in Washington, D.C., and Boulder, Colorado, is suggested in the following listing of the divisions and sections engaged in technical work. In general, each section carries out specialized research, development, and engineering in the field indicated by its title. A brief description of the activities, and of the resultant publications, appears on the inside of the front cover.

WASHINGTON, D. C.

Electricity. Resistance and Reactance. Electrochemistry. Electrical Instruments. Magnetic Measurements. Dielectrics. High Voltage. Absolute Electrical Measurements.

Metrology. Photometry and Colorimetry. Refractometry. Photographic Research. Length. Engineering Metrology. Mass and Volume.

Heat. Temperature Physics. Heat Measurements. Cryogenic Physics. Equation of State. Statistical Physics.

Radiation Physics. X-ray. Radioactivity. Radiation Theory. High Energy Radiation. Radiological Equipment. Nucleonic Instrumentation. Neutron Physics.

Analytical and Inorganic Chemistry. Pure Substances. Spectrochemistry. Solution Chemistry. Standard Reference Materials. Applied Analytical Research. Crystal Chemistry.

Mechanics. Sound. Pressure and Vacuum. Fluid Mechanics. Engineering Mechanics. Rheology. Combustion Controls.

Polymers. Macromolecules: Synthesis and Structure. Polymer Chemistry. Polymer Physics. Polymer Characterization. Polymer Evaluation and Testing. Applied Polymer Standards and Research. Dental Research.

Metallurgy. Engineering Metallurgy. Metal Reactions. Metal Physics. Electrolysis and Metal Deposition.

Inorganic Solids. Engineering Ceramics. Glass. Solid State Chemistry. Crystal Growth. Physical Properties. Crystallography.

Building Research. Structural Engineering. Fire Research. Mechanical Systems. Organic Building Materials. Codes and Safety Standards. Heat Transfer. Inorganic Building Materials. Metallic Building Materials.

Applied Mathematics. Numerical Analysis. Computation. Statistical Engineering. Mathematical Physics. Operations Research.

Data Processing Systems. Components and Techniques. Computer Technology. Measurements Automation. Engineering Applications. Systems Analysis.

Atomic Physics. Spectroscopy. Infrared Spectroscopy. Far Ultraviolet Physics. Solid State Physics. Electron Physics. Atomic Physics. Plasma Spectroscopy.

Instrumentation. Engineering Electronics. Electron Devices. Electronic Instrumentation. Mechanical Instruments. Basic Instrumentation.

Physical Chemistry. Thermochemistry. Surface Chemistry. Organic Chemistry. Molecular Spectroscopy. Elementary Processes. Mass Spectrometry. Photochemistry and Radiation Chemistry.

Office of Weights and Measures.

BOULDER, COLO.

CRYOGENIC ENGINEERING LABORATORY

Cryogenic Processes. Cryogenic Properties of Solids. Cryogenic Technical Services. Properties of Cryogenic Fluids.

CENTRAL RADIO PROPAGATION LABORATORY

Ionosphere Research and Propagation. Low Frequency and Very Low Frequency Research. Ionosphere Research. Prediction Services. Sun-Earth Relationships. Field Engineering. Radio Warning Services. Vertical Soundings Research.

Troposphere and Space Telecommunications. Data Reduction Instrumentation. Radio Noise. Tropospheric Measurements. Tropospheric Analysis. Spectrum Utilization Research. Radio-Meteorology. Lower Atmosphere Physics.

Radio Systems. Applied Electromagnetic Theory. High Frequency and Very High Frequency Research. Frequency Utilization. Modulation Research. Antenna Research. Radiodetermination.

Upper Atmosphere and Space Physics. Upper Atmosphere and Plasma Physics. High Latitude Ionosphere Physics. Ionosphere and Exosphere Scatter. Airglow and Aurora. Ionospheric Radio Astronomy.

RADIO STANDARDS LABORATORY

Radio Standards Physics. Frequency and Time Disseminations. Radio and Microwave Materials. Atomic Frequency and Time-Interval Standards. Radio Plasma. Microwave Physics.

Radio Standards Engineering. High Frequency Electrical Standards. High Frequency Calibration Services. High Frequency Impedance Standards. Microwave Calibration Services. Microwave Circuit Standards. Low Frequency Calibration Services.

Joint Institute for Laboratory Astrophysics-NBS Group (Univ. of Colo.).

