Phillippine Engineering Journal (1990) XI (1): 93-100

Technical Note

MATRIX OPERATIONS ON MICROS (MOM)

Salvador F. Reyes
Professor
Department of Civil Engineering
University of the Philippines

INTRODUCTION

Applications programs for matrix arithmetic have been available since the early days of electronic computing. SMIS (Symbolic Matrix Arithmetic System), developed for mainframes by Meek (1971), featured simple commands for elementary arithmetic operations as well as common special operations like copying of submatrices, solution of simultaneous equations, etc. MAIS (Matrix Arithmetic - Interpretive System) was developed by the writer (1968) also for mainframes and had the additional capability of evaluating arbitrary arithmetic operations automatically.

Microcomputers are evidently even more effective for such computations. Accordingly, a number of versions of IMAM (Interactive Matrix Arithmetic Module) were developed by the writer in interpreted BASIC for machines available during the early 80's, including pocket computers with at least 16k-bytes of memory and BASIC interpreter. More powerful current equipment permits the implementation of a program with a wider variety of matrix operations on micros (MOM) along with several modes of data entry, contextual help, and other user-friendly features.

The program described herein, coded in the widely used Turbo PASCAL dialect (Borland, 1988), enables the user to perform standard matrix operations or combinations of these operations by entering statements in practically the same manner they are usually written. Some common special operations like solving simultaneous equations, permutations, etc. are also provided.

These operations, along with input/output operations are made available without unneccessarily burdening the user with housekeeping details like allocation of storage for matrices in RAM or in the save file which is optionally provided as backup storage. Syntax checking of the arithmetic expressions is provided and the necessary consistency checks are made in the course of their evaluation.

The program is fairly robust in that the user is prevented as much as possible from making fatal errors, and should be able to recover from most that inevitably crop up. Most operations or tasks can be safely aborted during the initial stages.

The main features and how they were implemented are briefly described in the following sections.

BASIC FEATURES OF MOM

MOM Matrices

Matrices are named, as in IMAM, by single (case-immaterial) alphabetic characters. Thus, up to 26 matrices may be simultaneously available in random access memory (RAM) for arithmetic and other operations. Any of the matrices may be saved in a file of earlier editions (not necessarily of the same dimensions), any of which may be retrieved or returned to RAM.

The memory management system which these features entail dynamically allocates the matrices in one or more RAM pools, each of which can accommodate up to 8000 matrix elements. In turn, the areas allocated are maintained through a linked list of corresponding codewords containing, aside from pointers to the allocated areas, the matrix dimensions and labels. Allocation is on a "first available" basis.

Printable Output

Printable 80-column text can be produced in two forms - formatted or unformatted. For readability, formatted output presents the matrices with row and column labels. Keyboard entry of 40-character I.D. field is possible. "Unformatted" output presents the matrix elements in E-format (scientific format) for maximum precision, and uses no headers.

Save File

A set of matrices may be copied from a text file which has the "unformatted" form, into the save file from which they may then be selectively retrieved. This feature makes possible the use of MOM as a convenient adjunct to applications programs for performing matrix arithmetic. It can also be used to reorganize the save file should it become too fragmented from continuous usage as no "garbage collection" routine is provided for the file.

Automatic compaction is provided in the dynamic allocation scheme for RAM, however. Hence, the only practical limitations are that the size of any matrix may not exceed 8000 elements (the size of a single RAM pool). As many RAM pools or areas may be assigned for matrices as are available in the hardware being used.

Menu System

The user interacts with MOM through a two-level menu system from which the user interactively selects tasks in a sequential but otherwise completely unstructured fashion. The main menu, which is displayed after the program is initialized and upon completion or abortion of a task, consists of the following item:

1) Define, display, or update a matrix

2) Print a matrix

3) Push a matrix into the save file

4) Fetch a matrix from the save file

5) Delete an active matrix (in RAM)

6) Set or execute an arithmetic statement

7) Perform a (special) matrix operation

8) Perform a file management function

9) Quit

A second-level menu is presented when any one of items 6, 7, or 8 is selected. If item 6 (arithmetic) is selected, the following menu is presented.

1) Enter and (optionally) execute an arithmetic statement

2) Execute a stored arithmetic statement

3) Delete a stored arithmetic statement

4) Delete all stored arithmetic statements

If item 7 of the main menu (special matrix operations) is selected, the following menu is presented

1) Solve a system of equations

2) Copy a submatrix from a hypermatrix

3) Add a submatrix to a hypermatrix

4) Permute rows or columns of a matrix

5) Determine the row rank of a matrix

Finally, if item 8 (file management) is selected the following menu is presented.

1) Copy a sequential or text (ASCII) file to the save file

2) Duplicate the save file

3) Clear the save file

4) Open an ASCII (or printable output) file

5) Change the number field format specifications.

Tasks are selected from the displayed menus by pressing appropriate function (Fn) keys. Pressing the Alt-Fn key combination causes the help list corresponding to the nth item to be displayed. Details of tasks which are not self-evident are provided in these help lists.

Arithmetic Expression Evaluation

When an arithmetic statement is typed, it is automatically stored and, if the user so desires, immediately executed. As indicated in the subsidiary menu, any stored statement may subsequently be picked for execution from a display of the stored statements. Both selective and total deletion of stored statements are possible.

An arithmetic statement has the form "v = expression" in which "expression" consists of matrix variables represented by single alphabetic characters (case immaterial) combined with pairs of parentheses, "( )", and operators. All matrices to the right of the equals sign must be defined and conformable relative to the operations called for. Unlike the target matrix (v) which receives the result of the expression evaluated, the matrices in the expression itself remain unchanged.

The following comprise the set of operators and their respective ranks:

Op. Symbol	Definition	Rank	Type
+	addition	1	infix
−	subtraction	1	infix
*	multiplication	2	infix
-	negation	3	prefix
'	transposition	3	postfix
~	inversion	3	postfix

In a series of arithmetic operations involving operators of the same rank, execution is right to left. Operators of relatively higher rank take precedence during execution. For example

$$a = b + c * d$$

assigns b plus the product of c and d to a. As usual, the operator precedences may be overriden by using parentheses; thus

$$a = (b + c) * d$$

or

$$a = (b + c) d$$

multiplies the sum of b and c with d and assigns the result to a. As illustrated in the second form, the multiplication symbol may be omitted and blanks freely embedded in te expression). Scalar constants enclosed by "< >" may be included in the expression.

For example, the following hypothetical statement

$$a = - 29e6\ b'^{\sim}\ c(-d + e * f)$$

is executed as follows:

1. The scalar constant is negated.
2. B is inverted and transposed.
3. C and the result of step 2 are multiplied.
4. The result of steps 1 and 3 are multiplied.
5. D is negated.
6. E and F are multiplied.
7. The result of the last two steps are added.
8. The result of steps 7 and 4 are multiplied.
9. The result of the last step is assigned to A.

Appropriate statement syntax and operand conformability checks are made and the evaluation process is aborted if an error is detected. All matrices on the right hand side of the equals sign must of course be defined. The target matrix, if previously defined, is replaced by the result of the operations.

Special Operations

A number of special operations which are common in the applications are included in MOM. Prominent among these is the simultaneous equation solver. It determines the solution matrix, X, in the set of equations

$$A X = B$$

'in which A is the coefficient matrix and B the matrix of constants. The solution replaces B, and A is altered in the course of the operation. It may be noted that the same result may be obtained by evaluating the arithmetic expression

$$x = a^{\sim} b$$

or

$$b = a^{\sim} b$$

However, the special operations is more efficient for fairly large matrices.

Also provided under special operations are (1) extraction of a submatrix from a hypermatrix or system matrix, and (2) incrementation of a system matrix by a submatrix. These operations are useful in the study of the basic algorithm for the stiffness method of analysis, for example.

Permutation

Any permutation matrix is a permutation or rearrangement of the rows (or columns) of an identity matrix of the same size. A matrix of conformable size is row-permuted if premultiplied (or column-permuted if postmultiplied) by this matrix. Permutations, however, are actually done by direct row or column interchanges.

For present purposes, permutation function, p, is defined by a set of n pairs of integers,

$$(i, p[i]), \ i = 1, ..., n$$

in which the n distinct function values are constitute a rearrangement (permutation) of the set, 1, ..., n. In MOM, this set of integers may be specified as a row or column matrix.

The permutation operation, typified by the component, p[i] = j may mean either of the following

(a) row/column i is assigned to row/column j, or

(b) row/column i receives row/column j

These mutually inverse operations are referred to, under MOM, as "forward permutation" and "reverse permutation", respectively.

Rank Determination

The Gauss elimination process (Crandall, 1954) with partial pivoting is applied to a rectangular matrix to determine its row rank. If the matrix represents a set of system constraints, the number of columns is the number of (unconstrained) degrees of freedom. The process isolates the independent from the dependent degrees of freedom. The resulting reduced matrix is returned along with the generated permutation matrix. Permuting the reduced matrix based on the latter yields the system constraint matrix.

File Management

The set of tasks under this subsidiary menu is provided for managing the save file or backup storage (a direct-access file) as well as the print file or output file. Matrices may be imported from an ASCII file. For each matrix, the line-oriented data must consist of

1) one header line containing the matrix label ('A', 'B', columns) and, optionally, by an I.D. field of not more than 40 characters.

2) One more lines containing the elements of the matrix in row-wise order.

There are no restrictions on form except that data fields must be set apart with one or more blanks. Blanks between data lines are allowed.

CONCLUSIONS

A microcomputer-based program for conducting matrix arithmetic operations in a convenient interactive fashion. Various features like screen-oriented input, dynamic memory allocation, secondary working storage and a variety of output modes are provided to enhance the "user-friendliness" of the program. Although it is intended primarily as a learning aid for the study of elementary structural analysis, the program could well be useful in such basic courses as linear algebra, for example.

Copy of the program may be secured from the writer or from the Civil Engineering Department, University of the Philippines.

REFERENCES

BORLAND (1988). Turbo PASCAL Reference Guide, Borland International, Scotts Valley, CA.

Froberg, Carl-Erik (1979). Introduction to Numerical Analysis 2nd Edition.

Meek, John L. (1971). Matrix Structural Analysis, Mc-Graw Hill pp. 592-606.

Reyes, S. F. (1968). "MAIS - An Interpretive System for Matrix Arithmetic", Unpublished class notes.

Reyes, S. F. (1984). "An Interactive Matrix Arithmetic Module (IMAM), Philippine Engineering Journal, Vol. V, No. 2, pp. 66-79.