Matrix Manipulation

by: John A. Schlack

824 Rhoads Avenue
Jenkintown, PA 19046

date: June 1990 - July 1990

current version: 1.0

created: July 19, 1990

last modified: July 19, 1990

Introduction

Matrix Manipulation was developed to provide some common Linear Algebra operations on N x N matrices up to 99 x 99 in size. Data can either be entered by hand (up to 9 x 9) or read from a file in either binary or ASCII format.

This program performs five functions on matrices: transpose, determinant, inverse, solve a linear system of equations, and obtain the eigenvalues and eigenvectors. The eigenvalues and eigenvectors can only be obtained for a real, symmetric matrix.

This program is intended to work on a Mac Plus or later. The computer must be running System 6.0 or later, and it requires lots of memory for large matrices. The minimum matrix size is 2 x 2, and the matrices must be N x N.

Special Notices

Since this is an unregistered copy, you are limited to entering a matrix up to 4 x 4 in size by hand. You can enter any size matrix up to 99 x 99 through a file. In order to be able to enter a matrix up to 9 x 9 in size by hand, please register and pay the Shareware fee (see below).

Unfortunately, the screen has a lot of flicker when scrolling. This is due to uddating the TERecord for each line. I did this instead of using TEScroll since large matrices can have large screen outputs (over 100K). The TERecord is limited to 32K. I'll try to improve this in a future upgrade.

Registration

Please don't skip this section. Even if you don't plan on registering initially, some of what I have to say might entice you to register, especially if you want the program code for this or other products that I'm developing.

This program is Shareware. I am asking for a registration fee of $10. The unregistered version allows entrance of only a 4 x 4 matrix by hand. After you register, I will send a copy which allows up to 9 x 9 matrix entered by hand. If you pay the fee, you will get the opportunity to give your feedback to me. Believe me, I won't even look at suggestions if you don't register. If you do, I will read the suggestions, and consider implementing them in the next version (which I will send you free of charge).

Ok, now, here's the real value. If you send $25 for registration, I will send you the **FULL SOURCE CODE PROGRAM** in THINK C 4.02 format for this program. This will allow you to add and modify these procedures for your own program.

CHEAPER SHAREWARE FEES:

I'll drop the Shareware fees to $5 for registration and $20 for registration and source code if you agree to send a self-addressed stamped envelope and disk each time you are entitled to an upgrade (this saves me time and money). I'll tell you about the upgrade and let you send me a disk and SASE if you want to upgrade.

Please, when you register, give me a current address and the version number of Matrix Manipulation that you are using. Also, try to send either a check or money order. I will accept cash, but it could become a real hassle, especially if it gets lost in the mail.

Send your Shareware fees to:

John Schlack
824 Rhoads Avenue
Jenkintown, PA 19046

Operating Instructions

Determine whether the data will be entered by hand or through a file. If it will be entered by hand, the size is limited to 9 x 9, otherwise, the size is limited to 99 x 99. See below for information on creating a file. Matrices that are entered by hand always have solutions displayed to screen and must be saved manually. Matrices that are entered through files are always saved to disk unless one of the File Preferences is set to display to screen. If you do not want to save a solution, just hit cancel. You can always save it later manually (as long as you did not perform a new computation). I suggest that if you load the values in through a file, that you always display to screen. If you choose to automatically dump to a file, the only way to view the matrix on screen is to choose Show Default from the Default menu.

If the matrix will be entered by hand, choose the menu option Enter Matrix. This will display a dialog box with room for 9 rows. The first value entered should be the size of the matrix (from 2 to 9) in the top box. The remaining boxes allow entry of matrix data row by row. All data for a certain row must be on the same line, and separated by at least one space.

If the data is to be read by file, first choose File Preferences. This allows you to set up read and save characteristics. The files can be loaded as either ASCII or binary. The files that you save can be either ASCII or binary. The ASCII header option includes headers and labels for the data (not just raw numbers). Otherwise, you will need to consult the Save File Templates (see below) to read the appropriate data, as only raw numbers are saved (for reading in by other programs that you might want to develop). You can choose whether to always save data or always display the data to the screen if (and only if) the file was read in through a file. Finally, you tell the computer whether the size of the matrix is the first number in the external file or if you will enter it manually.

You can now perform desired operations on the matrix if there was no errors during data entry. The functions are under the Special menu. The program keeps a copy of the matrix and vectors (which I have not discussed yet; two vector examples are the eigenvalues of a matrix and the solution vector for a linear system of equations) just before the operation was performed and allows you to revert back to them (using the Default menu options; which work similar to an Undo). When any operation is performed (except determinant), the solution will replace the current matrix and vectors. Determinant is different from the other functions as it only returns a single value. This value is displayed to screen and cannot be saved. It does not change any stored matrix or vector values.

If you want to operate on a new matrix, you must close the current matrix (with a menu item Close) and load the new matrix. If data has not been saved and you chose Close or Quit, then a dialog box appears asking if you want to save the current data.

Most functions just require selecting the appropriate menu item. Solve Linear System is different, however. If this option is selected, a solution vector (B) must be provided for the matrix (A) to solve for X:
[A] [X] = [B]. If you entered the values by hand, then a dialog box will appear asking you to enter these values by hand. If you loaded a file, then a dialog box will appear asking you to locate the appropriate file (which must contain N double precision values).

When a file is displayed on screen, it will be shown in matrix form with headers that will be similar to those that appear in ASCII files with headers. The headers option displays titles, the original matrix (and vectors if appropriate), and the solution matrix and/or vectors. When saving a solution, if ASCII is chosen, it will be written in matrix form if N is less than or equal to 9. Otherwise, it will be in listing format.

To view the current and previous matrices and vectors, choose Show Default Matrix from the Default menu. This displays the old matrix and vector followed by the matrix and vector that the computer will use as the data. You can swap these sets by choosing Set Default Matrix from the Default menu. NOTE: by hitting Set Default Matrix, the previous data set will be destroyed. This is why you will be prompted to save data if you have not yet done so. Currently, the vector has no significance for swapping purposes. The vectors will be swapped, but if you are prompted to enter a vector (for a solution to non-linear systems of equations), you will need to enter it again. It will not use the stored vector.

Creating Files

External data files are an important aspect of this program. These files are needed for matrices larger than 9 x 9. The format is either ASCII (in C using fprintf or by use of a text editor) or binary (in C using either FSWrite or fwrite). The data is double precision. NOTE: The length of the binary double is 10 bytes (for SANE compatibility), not 12 bytes (that the 68881/2 coprocessor uses). The size must be between 2 x 2 and 99 x 99 inclusive. The first value entered must be the size (written as a double, between 2 and 99) if you are not going to enter the value manually.

The data written should start at the top row, first column. It should proceed to the end of the row, then increment the row by one, and start again at the beginning of the row. Thus, the program reads as follows:

A[1][1], A[1][2], ... , A[1][N], A[2][1], A[2][2], ... , A[2][N], etc. until A[N][N] is reached.

ASCII files must have "white" spaces between each double precision number. A white space is either a tab, space, or return. The last data value must have at least one white space after it, but two is desirable.

Save File Templates

Below is a text explanation of the set up of each type of saved solution when the headers are not included.

FOR ALL SOLUTIONS

- All values are double precision numbers 10 bytes long (for SANE compatibility).
- The first value is the function type:

    1. transpose
    2. eigenvalues and eigenvectors
    3. solution to linear system of equations
    4. inverse

- The second value is the size N of the matrix (N x N).

TRANSPOSE, INVERSE

- The data is arranged as it was when read in. A[1][1] is the first value (after the size), followed by A[1][2], until A[1][N] is reached. A[2][1] is next, then A[2][2], until A[2][N], etc. until the end of the matrix (A[N][N]) is reached.

SOLUTION TO LINEAR SYSTEM OF EQUATIONS

- This solution is a vector, not a matrix. The first value after the size is X[1] and this proceeds sequentially until X[N] is reached.

EIGENVALUES AND EIGENVECTORS

- The first value written is the set of eigenvalues in vector form. B[1] (the first eigenvalue) is written, followed by B[2], and so on until B[N] is reached.
- After the eigenvalues come the eigenvectors. The eigenvectors for the first eigenvalue are written first starting from 1 and proceeded to the Nth value of that eigenvector. Then, the next eigenvector is written (which corresponds to the second eigenvalue). This process is continues until the N eigenvectors have been written.

I have included a sample source code to read the binary file. This is not a stand alone program, but can be incorporated into a program to read the data files or changed to write the data files.

Menus:

Apple

About Matrix: This gives information about the creation of this program.

Help: This is a full-featured help section. The small window lists the topics. When one is highlighted, information regarding that topic appears in a scrollable window.

File

Enter Matrix Values: This menu item allows for manual entry of a matrix up to 9 x 9 in size. The top box is the size of the matrix (N x N). The 9 remaining boxes represent the rows of the matrix. Include at least one space between values in each row. If you do not enter enough values, the program will display an error message and the data will have to be re-entered.

Load Matrix Values: This menu option allows entry of the matrix data through a file. Before choosing this option, make sure that the file preferences are set the way you want. I have included sample source code to help you write routines to read and write the matrix data.

File Preferences: This menu brings up a dialog box that allows you to select some file preferences that you want. These include: loading either an ASCII or binary file, saving in either ASCII or binary format, saving the headers for an ASCII file if ASCII file save is chosen and the matrix size is less than 10 x 10, a flag that tells the computer to get the matrix size from the file or get it from the user, and whether to display to screen or automatically save data if the data was read from a file.

Save: This option saves matrix data for the following types of functions: eigenvalues and eigenvectors, solving linear systems of equations, matrix inverse, and matrix transpose. The file will be saved in a format determined by the file preferences.

Close: This item disposes of the matrix data, closes the window, and resets all globals (except file preferences) to the values that they had when the program was entered.

Quit: Use this to quit the program.

Edit

This menu is not operational.

Special

Check Matrix: This menu option checks the matrix for symmetry. An error dialog box will appear if the matrix is not symmetric. If it is symmetric, no action is taken.

Eigenvalues and Vectors: This first checks the matrix for symmetry (see Check Matrix). If it is not symmetrical, the process is stopped. Otherwise, this calculates the eigenvalues and eigenvectors of the matrix.

Inverse: This calculates the inverse of the matrix.

Determinant: This menu item finds the determinant of the matrix. As opposed to the other functions, this value cannot be saved. It is simply displayed in a dialog box.

Solve Linear System: This menu item is also different from the previous items. You must enter a vector representing B in AX = B, where A is the matrix. The program will then calculate X (the solution vector for the linear system of equations). If you entered the values by hand, this displays a dialog to enter the values across a row (not a column). Otherwise, the program prompts the user for the name of a file to load (using the current file preferences). NOTE: If one or more rows are linear combinations of the other rows (and will row reduce to all zeros), the results will be incorrect. This program cannot handle such a situation. Be careful of such possible situations (this program will NOT let you know when they happen).

Transpose: This calculates the transpose of the matrix.

Default

Show Default Matrix: This menu item displays the old matrix and vector and the current matrix and vector in a window on the screen. This will allow you to swap the values if you want to do more calculations on the old matrix.

Set Default Matrix: This swaps the matrices and vectors described in Show Default Matrix. Make sure that you have saved the previous work since it will be lost by choosing this option.

Other Info

When you register, you will get a certain number of future upgrades free. If you pay $10 (or $5), you get the fully functional current version and one fully functional free upgrade. If you pay $25 (or $20), you get the above, plus the source code for the current version of the software. When a new version comes out, I will send you a letter about it (if you are entitled to a free upgrade) explaining the new features. You have the option of using the upgrade or waiting until a newer version appears to use the free upgrade.

Revision History

version 1.0: (July 19, 1990) This is the initial release.

Machines Tested

This program has only been tested on a Mac SE. Please write and tell me your experiences on other computers. If any trouble arises, please contact me with a full description of the problem, hardware, System and INITs, and version of *Matrix Manipulation.*

Other Projects

If you register, please tell me if you want details my other projects. Some are available, others will be available soon, or I might develop a program if there is enough interest in it.

Feedback

Register and give me feedback. I want to know what you think. What do you like? What do you want improved? Added? Deleted? Overhauled? Registration is $10, and you will get a fully functional current version and a future update sent to you free, along with a description of other completed works and projects that I'm working on if you request it. For $25, you get all the above, plus the THINK C 4.02 source code for the current version (if you would like the source for the updates, contact me when the update becomes available, and we'll work something out). It is only $5 and $20 if you send the disk and SASE. A great value considering the time that I put into this program. Send your comments and registration fees to:

John Schlack
824 Rhoads Avenue
Jenkintown, PA 19046

You can also reach me through E-mail. On **America Online**, I am **John40**. On **CompuServe** (which I rarely use), I am **71311,1262**.